

# ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ

4/2003

НАУЧНО - ТЕХНИЧЕСКИЙ ЖУРНАЛ

РАСПОЗНАВАНИЕ РЕЧИ

ЦОС В РАДИОЛОКАЦИИ

МОДУЛИ ЦОС  
И ИХ ПРИМЕНЕНИЕ

ISSN 1684-2634



9 771684 263005 >

# ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ

Научно-технический журнал  
№4 (10)/2003

Издается с 1999 года.  
Выходит четыре раза в год.

**ГЛАВНЫЙ РЕДАКТОР**  
Ю.Б. ЗУБАРЕВ

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

## РЕДАКЦИОННАЯ КОЛЛЕГИЯ:

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

**Адрес редакции:**  
107241 Москва, Щелковское ш., 23А  
Тел.: (095) 290-9088  
Факс: (095) 290-9085  
E-mail: editor@dspa.ru  
http://www.dspa.ru

**Для писем:**  
129090 Москва, а/я 48.

**Издатель:**  
КБ волоконно-оптических приборов  
Генеральный директор: С.А. Задворнов  
Контрольный редактор: С.Х. Мамиева  
Дизайн и верстка: И.Н. Романовская

Подписной индекс по каталогу  
ОАО «Роспечать» – **82185**

Подписано в печать 30.12.2003.  
Формат 60x90/8.

Гарнитура «Pragmatica C». Печать офсетная.  
Бумага мелованная. Печ.л. 7,0. Тираж 1000 экз.  
Заказ № 512.

Отпечатано в ООО «Типография Принтхаус»  
Москва, ул. Вавилова, вл. 9а, стр. 7,  
тел.: 132-7218, 105-0798.

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

Свидетельство о регистрации ПИ №77-1488  
от 14.01.2000

## В НОМЕРЕ:

С.Ю. Иконин, Д.В. Сарана

**Система автоматического распознавания речи  
SPIRIT ASR Engine.....2**

А.К. Пяткин, М.В. Никитин

**Цифровая фильтрация частотно-модулированных импуль-  
сов в многофункциональных РЛС..... 13**

М.Н. Быканов, С.М. Седунов, В.А. Толмачев

**Обработка и анализ радиосигналов на основе  
многофункционального модуля цифровой обработки  
сигналов XDSP-ЗМС РЛС.....20**

А.А. Рыболовлев, В.Г. Трубицын

**Технология eXpressDSP проектирования и отладки  
систем цифровой обработки сигналов при работе  
с TMS320VC5416DSK. Интегрированная среда  
разработки Code Composer Studio v.2 для DSK.....24**

Н.И. Гутовец

**Семейство модулей цифровой обработки сигналов  
на основе процессоров TMS320C64xx и ПЛИС XILINX.....33**

**Выставка «Электроника. Компоненты.**

**Оборудование. Технологии» или  
«Праздник электроники прошел как по нотам...» .....40**

А.А. Ланнэ

**Рецензия на книгу**

**А.Б. Сергиенко «Цифровая обработка сигналов».....44**

Подписной индекс  
по каталогу «Роспечать»

# 82185

# Система автоматического распознавания речи

## SPIRIT ASR Engine

С. Ю. Иконин, Д. В. Сарана

### Введение

**С**истема распознавания речи SPIRIT ASR Engine разработана для широкого ряда практических задач. К числу таких задач можно отнести, например, организацию автоматического call-центра (головное управление системой меню, набор PIN-кода и телефонного номера), системы безопасности, системы речевого управления и т.д. Система способна в режиме реального времени производить дикторонезависимое распознавание цепочек слитно произнесенных слов и отдельных речевых команд, в том числе и в шумовых условиях при соотношении сигнал-шум вплоть до +5дБ.

Качество работы системы было протестировано на речевой базе TIDigits, содержащей цепочки английских цифр от «0» до «9» плюс «Oh». Общее количество высказываний – 8700 (56 мужчин и 57 женщин) при длине строки от 1 до 7 цифр. Точность распознавания изолированных команд (длина строки – 1) составила 99,9%. Точность распознавания цепочек цифр – 97,9%. При использовании информации о длине цепочки качество распознавания повышается до 98,8%. Такой режим используется, например, при вводе PIN-кода или телефонного номера, когда количество цифр в высказывании известно заранее.

Следует отметить, что алгоритмы системы никак не привязаны к конкретному языку и составу словаря. При наличии соответствующего речевого материала система может быть переобучена на любой другой набор слов и речевых команд, без каких либо затруднений.

В системе SPIRIT ASR Engine были реализованы как известные решения, такие как Скрытые Марковские модели (СММ), так и нестандартные подходы, позволившие значительно повысить надежность распознавания в реальных акустических условиях.

*Статья открывает серию публикаций, посвященных технологии распознавания связной речи и речевых команд и различным аспектам ее практического применения на примере системы SPIRIT ASR Engine, разработанной в компании SPIRIT Corp. Особое внимание при разработке было уделено вопросам устойчивой работы системы в «агрессивной» шумооой обстановке. В данной работе рассматриваются основные принципы построения и применения скрытых марковских моделей (СММ) в системах распознавания речи с ограниченным словарем, а также некоторые практические аспекты реализации системы распознавания слитно произнесенных цепочек слов в режиме реального времени.*

Использование СММ является на сегодняшний день наиболее популярным и успешно применяемым подходом к проблеме распознавания речи. В данной статье представлены основные принципы СММ.

В следующей статье будут рассмотрены алгоритмы распознавания слитно произнесенных цепочек слов, примененные в SPIRIT ASR Engine.

### Скрытые Марковские модели

Рассмотрим основные принципы СММ. Более детальное обсуждение теории СММ и вопросов их применения в распознавании речи можно найти в [1,2].

### Определение Скрытых Марковских моделей

Использование СММ для распознавания речи базируется на следующих предположениях:

1. Речь может быть разбита на сегменты (состояния), внутри которых речевой сигнал может рассматриваться как стационарный. Переход между этими состояниями осуществляется мгновенно.
2. Вероятность символа наблюдения, порождаемого моделью, зависит только от текущего состояния модели и не зависит от предыдущих порожденных символов.

По сути, ни одно из этих двух предположений не является справедливым для речевого сигнала. Боль-

шое количество исследований было посвящено тому, чтобы сгладить недостатки этих предположений [3]. Тем не менее, стандартные СММ являются основой для большинства современных систем распознавания речи.

Существует несколько типов СММ, различающихся по своей топологии (эргодические, лево-правые и др.), с дискретными или непрерывными символами наблюдения. Для построения данной ASR-engine использовались лево-правые СММ без пропусков состояний с непрерывной плотностью наблюдений, именно такие модели и будут рассмотрены. На рис. 1 представлена топология подобной СММ с тремя состояниями.

Скрытая Марковская модель представляет собой конечный автомат, изменяющий свое состояние в каждый дискретный момент времени  $t$ . Переход из состояния  $S_i$  в состояние  $S_j$  осуществляется случайным образом с вероятностью  $a_{ij}$ . В каждый дискретный момент времени модель порождает вектор  $o_t$  наблюдений с вероятностью  $b_j(o_t)$ .

Для определения СММ необходимо задать следующие элементы:

1.  $N$  – число состояний модели. В каждый момент времени модель может находиться в одном из  $N$  различных состояний  $S_1, S_2, \dots, S_N$ . В дискретные моменты времени  $t$  модель претерпевает изменение состояния (возможно, переходя при этом опять в то же состояние). Состояния модели в каждый момент времени будем обозначать как  $q_t$ .
2. Распределение вероятностей переходов между состояниями  $A = \{a_{ij}\}$ , где

$$a_{ij} = P[q_{t-1} = S_i | q_t = S_j], \quad 1 \leq i, j \leq N. \quad (1.1)$$

3.  $V = \{b_j(o_t)\}$  – распределение плотностей вероятности наблюдений для каждого состояния  $S_j$ , где  $o_t$  – вектор наблюдений в момент времени  $t$  и

$$b_j(o_t) = P(o_t | q_t = S_j), \quad 1 \leq t \leq T, \quad 1 \leq j \leq N \quad (1.2)$$

В непрерывных СММ величина  $b_j(o_t)$  моделируется конечной гауссовской смесью вида

$$b_j(o_t) = \sum_{k=1}^M c_{jk} N(o_t | \mu_{jk}, U_{jk}), \quad (1.3)$$

где  $c_{jk}$  – весовой коэффициент  $k$ -й компоненты смеси в состоянии  $j$ ,  $M$  – количество компонент смеси и  $N$  – Гауссовская плотность вероятности. Коэффициенты  $c_{jk}$  удовлетворяют стохастическим ограничениям

$$\sum_{k=1}^M c_{jk} = 1, \quad 1 \leq j \leq N \quad (1.4)$$

$$c_{jk} \geq 0, \quad 1 \leq j \leq N, \quad 1 \leq k \leq M.$$

Плотность  $N$  характеризуется вектором средних значений  $\mu_{jk}$  и ковариационной матрицей  $U_{jk}$  для  $k$ -ой компоненты смеси в состоянии  $S_j$ :

$$N(o_t | \mu_{jk}, U_{jk}) = \frac{1}{\sqrt{(2\pi)^n |U_{jk}|}} \exp[-\frac{1}{2}(o_t - \mu_{jk})^T U_{jk}^{-1} (o_t - \mu_{jk})], \quad (1.5)$$

где  $n$  – размерность вектора наблюдений  $o_t$ .

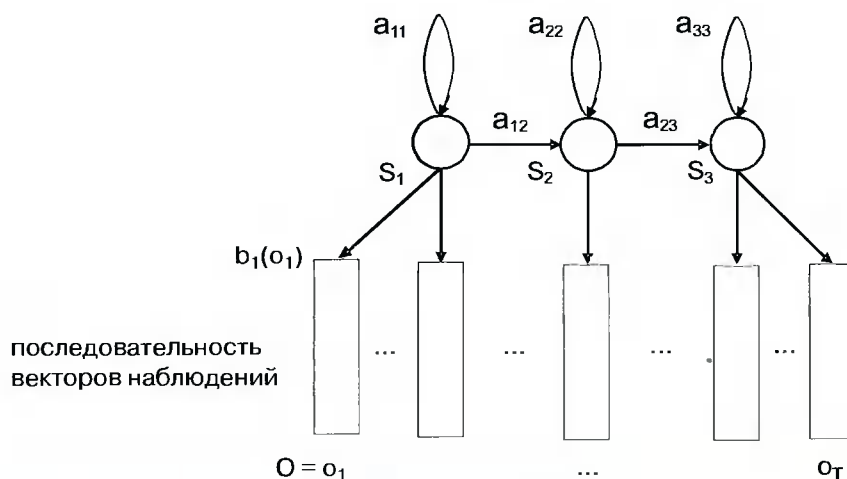


Рис. 1. Лево-правая СММ без пропусков состояний

4. Начальное распределение вероятностей состояний  $\pi = \{\pi_i\}$

$$\pi_i = P[q_1 = S_i], \quad 1 \leq i \leq N. \quad (1.6)$$

Будем обозначать набор параметров, полностью определяющий СММ, как  $\lambda = (A, B, \pi)$ .

Существует три основные проблемы, которые необходимо решить, для того, чтобы модель, описанная выше, могла бы использоваться в практических задачах.

**Проблема 1.** Пусть заданы последовательность наблюдений  $O = O_1, O_2, \dots, O_T$  и модель  $\lambda = (A, B, \pi)$ . Как эффективно вычислить вероятность  $P(O|\lambda)$  появления этой последовательности наблюдений для данной модели?

**Проблема 2.** Пусть заданы последовательность наблюдений  $O = O_1, O_2, \dots, O_T$  и модель  $\lambda = (A, B, \pi)$ . Как выбрать последовательность состояний  $Q = q_1, q_2, \dots, q_T$ , которая с наибольшей вероятностью для данной модели  $P(O, Q|\lambda)$  порождает заданную последовательность наблюдений?

**Проблема 3.** Каким образом нужно подстроить параметры модели  $\lambda = (A, B, \pi)$ , для того, чтобы максимизировать  $P(O|\lambda)$ ?

Далее последовательно будут рассмотрены эти три проблемы и алгоритмы, приводящие к их решению.

### Алгоритм прямого-обратного хода (решение проблемы 1)

Наиболее прямой путь для вычисления вероятности  $P(O|\lambda)$  – перечислить все возможные последовательности состояний заданной длины  $T$  и рассчитать соответствующие вероятности. Так, для фиксированной последовательности  $Q = q_1, q_2, \dots, q_T$ , вероятность её появления для данной модели  $\lambda$  равна

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T}. \quad (1.7)$$

Вероятность появления заданной последовательности наблюдений для этой фиксированной последовательности состояний при условии независимости наблюдений определяется как

$$P(O|Q, \lambda) = b_{q_1}(O_1) b_{q_2}(O_2) \dots b_{q_T}(O_T). \quad (1.8)$$

Совместная вероятность последовательностей  $O$  и  $Q$  – это произведение вероятностей:

$$P(O, Q|\lambda) = P(O|Q, \lambda) P(Q|\lambda). \quad (1.9)$$

Вероятность появления последовательности наблюдений  $O$  для данной модели вычисляется как сум-

ма всех этих совместных вероятностей для всех возможных последовательностей состояний  $Q$ :

$$\begin{aligned} P(O|\lambda) &= \sum_Q P(O|Q, \lambda) P(Q|\lambda) = \\ &= \sum_Q \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \dots a_{q_{T-1} q_T} b_{q_T}(O_T). \end{aligned} \quad (1.10)$$

Из последнего выражения следует, что при прямом подсчете вероятности  $P(O|\lambda)$  требуется провести порядка  $2TN^T$  вычислительных операций. Даже для небольших чисел  $N=10$  и  $T=5$  необходимо порядка  $10^6$  операций умножения. Ясно, что для практического решения проблемы 1 требуется более эффективная процедура. Такая процедура существует и называется процедурой прямого-обратного хода (the forward-backward procedure).

Существуют две модификации алгоритма, равноценные по вычислительным затратам – алгоритм прямого хода и алгоритм обратного хода. Эти алгоритмы различаются выбором ведущей переменной, прямой или обратной, которая предпочтительней в каждом конкретном случае.

### Алгоритм прямого хода

Введем так называемую прямую переменную  $\alpha_t(i)$ , определяемую выражением

$$\alpha_t(i) = P(O_1, O_2, \dots, O_t, q_t = S_i | \lambda), \quad (1.11)$$

которая представляет собой вероятность появления для данной модели частичной последовательности наблюдений  $O_1, O_2, \dots, O_t$  до момента времени  $t$  и состояния  $S_i$  в этот момент времени. Значения переменной  $\alpha_t(i)$  вычисляются по индукции следующим образом:

1. Инициализация:

$$\alpha_t(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N. \quad (1.12)$$

2. Индуктивный переход:

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}) \quad (1.13)$$

$$1 \leq t \leq T-1,$$

$$1 \leq i \leq N$$

3. Окончание:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i). \quad (1.14)$$

Для вычисления вероятности  $P(O|\lambda)$  таким образом требуется уже порядка  $N^T$  [1] вычислительных операций. Для взятых в качестве примера чисел  $N=10$  и  $T=5$  это составляет около 500 операций умножения, что в 2000 раз меньше, чем для прямых вычислений.

### Алгоритм обратного хода

Аналогичным образом можно ввести обратную переменную  $\beta_t(i)$ , определяемую выражением

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \dots, O_T | q_t = S_i, \lambda), \quad (1.15)$$

которая для заданной модели  $\lambda$  представляет собой совместную вероятность появления частичной последовательности наблюдений от момента времени  $t+1$  до  $T$  и состояния  $S_i$  в момент времени  $t$ . Значения обратной переменной также можно вычислить по индукции:

1. Инициализация:

$$\beta_T(i) = 1, \quad 1 \leq i \leq N. \quad (1.16)$$

2. Индукция:

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad (1.17)$$

для всех  $t = T-1, T-2, \dots, 1, 1 \leq i \leq N$ .

3. Окончание:

$$P(O|\lambda) = \sum_{i=1}^N \pi_i b_i(O_1) \beta_1(i). \quad (1.18)$$

### Алгоритм Витерби (решение проблемы 2)

В отличие от проблемы 1, для которой можно дать точное решение, для проблемы 2, а именно поиска «оптимальной» последовательности состояний, вообще говоря, существует несколько возможных решений. Это обусловлено существованием нескольких приемлемых критериев оптимальности. Например, один из возможных критериев состоит в том, чтобы выбрать такие состояния  $q_t$ , каждое из которых, *взятое отдельно*, является наиболее вероятным. Можно также максимизировать вероятность появления пар состояний  $(q_t, q_{t+1})$  или троек состояний  $(q_t, q_{t+1}, q_{t+2})$  и т.д. Однако, наиболее широко используется критерий, основанный на поиске единственной наилучшей последовательности состояний, максимизирующий  $P(Q|O, \lambda)$ . Метод определения такой наилучшей последовательности состояний, основанный на динамическом программировании, называется алгоритмом Витерби.

Итак, для отыскания наилучшей последовательности состояний  $Q = q_1, q_2, \dots, q_T$  по заданной последовательности наблюдений  $O_1, O_2, \dots, O_T$  определим следующую величину:

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1, q_2, \dots, q_t = i, O_1, O_2, \dots, O_T | \lambda], \quad (1.19)$$

которая представляет собой максимальную вероятность того, что при заданных  $t$  первых наблюдениях последовательность состояний заканчивается (в момент  $t$ ) в состоянии  $S_i$ . Для того, чтобы затем восстановить последовательность состояний для всех значений  $t$  будем хранить значения аргументов, максимизирующих вероятность в массиве  $\Psi_t(j)$ . Полная процедура определения последовательности состояний будет выглядеть следующим образом:

1. Инициализация:

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N; \quad (1.20a)$$

$$\Psi_1(i) = 0. \quad (1.20b)$$

2. Рекурсия:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij} b_j(O_t)], \quad (1.21a)$$

$$2 \leq t \leq T, \quad 1 \leq j \leq N;$$

$$\Psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \quad (1.21b)$$

$$2 \leq t \leq T, \quad 1 \leq j \leq N.$$

3. Окончание:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)], \quad (1.22a)$$

$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)]. \quad (1.22b)$$

4. Восстановление пути (последовательности состояний):

$$q_t^* = \Psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots, 1. \quad (1.23)$$

Реализация алгоритма Витерби аналогична (за исключением шага восстановления) процедуре прямого хода. Основное отличие – использование вместо процедуры суммирования процедуры максимизации. Кроме того, алгоритм Витерби может быть применен для решения проблемы 1 (определения вероятности появления заданной последовательности наблюдений для данной модели), поскольку на окончательном шаге алгоритма мы получаем вероятность для всей предшествующей последовательности наблюдений (выражение 1.22a).

### Алгоритм Баума-Уелша (решение проблемы 3)

Проблема переоценки параметров модели (A,B,π) по заданной последовательности наблюдений – наиболее трудоемкая (в вычислительном плане) проблема СММ. Не существует известного аналитического выражения для параметров такой модели. Тем не менее, используя итеративные процедуры, можно локально максимизировать вероятность  $P(O|\lambda)$ . Одна из таких процедур – метод переоценки Баума-Уелша (*Baum-Welch method*) или, что эквивалентно, EM-метод (*expectation-maximization*).

Введем переменную:

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda), \quad (1.24)$$

которая определяет вероятность того, что при заданной последовательности наблюдений в моменты времени  $t$  и  $t+1$  система будет соответственно находиться в состояниях  $S_i$  и  $S_j$ .

Используя определения прямой и обратной переменных (выражения 1.11 и 1.15), можно записать:

$$\begin{aligned} \xi_t(i,j) &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) b_{t+1}(j)}{P(O|\lambda)} = \\ &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) b_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) b_{t+1}(j)}. \end{aligned} \quad (1.25)$$

Введем также переменную  $\gamma_t(i)$ , являющуюся апостериорной вероятностью того, что при заданной последовательности наблюдений  $O$  система в момент времени  $t$  будет находиться в состоянии  $S_i$ :

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i,j). \quad (1.26)$$

Если величину  $\gamma_t(i)$  просуммировать по всем  $t$ , то результат можно рассматривать как ожидаемое время пребывания системы в состоянии  $S_i$ . Аналогичным образом результат суммирования  $\xi_t(i,j)$  по всем  $t$ , можно рассматривать как ожидаемое число переходов из состояния  $S_i$  в  $S_j$ .

$$\sum_{t=1}^{T-1} \gamma_t(i) \text{ – ожидаемое число} \quad (1.27a)$$

переходов из  $S_i$ .

$$\sum_{t=1}^{T-1} \xi_t(i,j) \text{ – ожидаемое число} \quad (1.27b)$$

переходов из  $S_i$  в  $S_j$ .

Используя перечисленные выше формулы можно получить процедуру переоценки параметров СММ:

$$\bar{\pi}_i = \gamma_1(i), \quad (1.28a)$$

$$\bar{a}_{ji} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad (1.28b)$$

$$\bar{b}_j(o_t) = \sum_{k=1}^M \bar{c}_{jk} N(o_t, \bar{\mu}_{jk}, \bar{U}_{jk}). \quad (1.28c)$$

Переоценка компонент  $\bar{c}_{jk}$ ,  $\bar{\mu}_{jk}$ ,  $\bar{U}_{jk}$  в выражении 1.28c выполняется по следующим формулам:

$$\bar{c}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j,k)}{\sum_{t=1}^T \sum_{k=1}^M \gamma_t(j,k)}, \quad (1.29a)$$

$$\bar{\mu}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j,k) \cdot o_t}{\sum_{t=1}^T \gamma_t(j,k)}, \quad (1.29b)$$

$$\bar{U}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j,k) \cdot (o_t - \bar{\mu}_{jk})(o_t - \bar{\mu}_{jk})^T}{\sum_{t=1}^T \gamma_t(j,k)}, \quad (1.29c)$$

где  $\gamma_t(j,k)$  – вероятность того, что при заданной последовательности наблюдений в момент времени  $t$  модель находится в состоянии  $j$ , причем наблюдаемый в это момент вектор  $o_t$  порожден  $k$ -й компонентой смеси, т.е.

$$(1.30)$$

$$\gamma_t(j,k) = \left[ \frac{\alpha_t(j) \beta_t(i)}{\sum_{i=1}^N \alpha_t(i) \beta_t(i)} \right] \left[ \frac{c_{jk} N(o_t, \mu_{jk}, U_{jk})}{\sum_{k=1}^M c_{jk} N(o_t, \mu_{jk}, U_{jk})} \right]$$

Переоценка параметров модели  $\lambda$  по приведенным формулам приводит к возрастанию функции правдоподобия, то есть:

$$P(O|\bar{\lambda}) \geq P(O|\lambda). \quad (1.31)$$

### Распознавание речи на базе СММ

Существует множество различных способов использования СММ в целях распознавания речи. Выбор определенного метода определяется конкретной задачей, которую необходимо решить. Например, задачу распознавания изолированных слов (речевых команд) при словаре небольшого размера (порядка сотен

слов) можно эффективно решить, используя отдельную СММ для каждого слова. При таком подходе входная последовательность наблюдений пропускается через каждую СММ. Используя алгоритм прямого или обратного хода, или же алгоритм Витерби, вычисляется вероятность правдоподобия для каждого слова. В качестве результата выбирается слово, вероятность правдоподобия которого наибольшая, то есть :

$$V^* = \arg \max_{1 \leq v \leq V} (P(O | \lambda_v)), \quad (1.32)$$

где  $V$  – размер словаря распознавания. Схематически этот процесс представлен на *рис. 2*.

Более сложной задачей является распознавание слитного высказывания, содержащего слова из заданного словаря. Одним из вариантов такой задачи является задача распознавания последовательности произносимых цифр. При этом базовой единицей остается слово, а слитная речь распознается как сочлененная последовательность из моделей слов. Задачи такого типа называются задачами распознавания связанных (конкатенированных) слов. Методы ее решения будут приведены в следующем разделе.

### Алгоритмы распознавания слитно произнесенных речевых команд, применяемые в SPIRIT ASR Engine

В данном разделе описаны основные алгоритмы, использованные при построении системы распознавания слитно произнесенных речевых команд.

Перед тем как перейти к рассмотрению собственно алгоритмов, мы коснемся некоторых аспектов, связанных с корректным моделированием длительности состояний СММ.

### Моделирование длительности состояний

Можно показать (см. [1]), что плотность вероятности  $p_j(d)$  пребывания в состоянии  $S_j$  с переходной вероятностью  $a_{ij}$  имеет вид:

$$p_j(d) = (a_{ij})^{d-1} (1 - a_{ij}). \quad (2.1)$$

Однако для большинства физических сигналов эта плотность вероятности для длительности пребывания в состоянии является неприемлемой. Более предпочтительным вариантом является моделирование длительности состояния явным образом. На *рис. 3* в качестве примера такого моделирования представлена гистограмма вероятности  $p_j(d)$  для первого состояния слова «six», соответствующего начальному звуку /s/. Гистограмма получена эвристически на основе разметки на состоянии. Отметим, что подобные распределения хорошо аппроксимируются Гамма функцией (см. *рис. 3*).

Введение явным образом информации о длительности, накладывает дополнительное ограничение на длительность пребывания модели в одном состоянии  $d_{min} \leq d \leq d_{max}$ . При наложении такого ограничения использование информации только от предыдущего состояния приводит к некорректной работе алгоритма Витерби. Это будет ясно из следующего примера.

Рассмотрим участок декодирования, представленный на *рис. 4*. Цифрами условно обозначены вероятности появления символов наблюдения  $b_j(o_t)$ .

Для состояния  $S_j$  в момент времени  $t$ , следуя правилу максимизации, мы должны выбрать состояние  $S_{j-1}$ , как наиболее вероятное предыдущее состояние. Предположим, что на длительность состояния  $S_j$  наложено ограничение  $d_j \geq 2$ . Тогда единственно возможным переходом из состояния  $S_j$  в момент времени  $t+1$  будет переход в то же состояние. Вследствие этого к точке к моменту  $t+2$  бу-

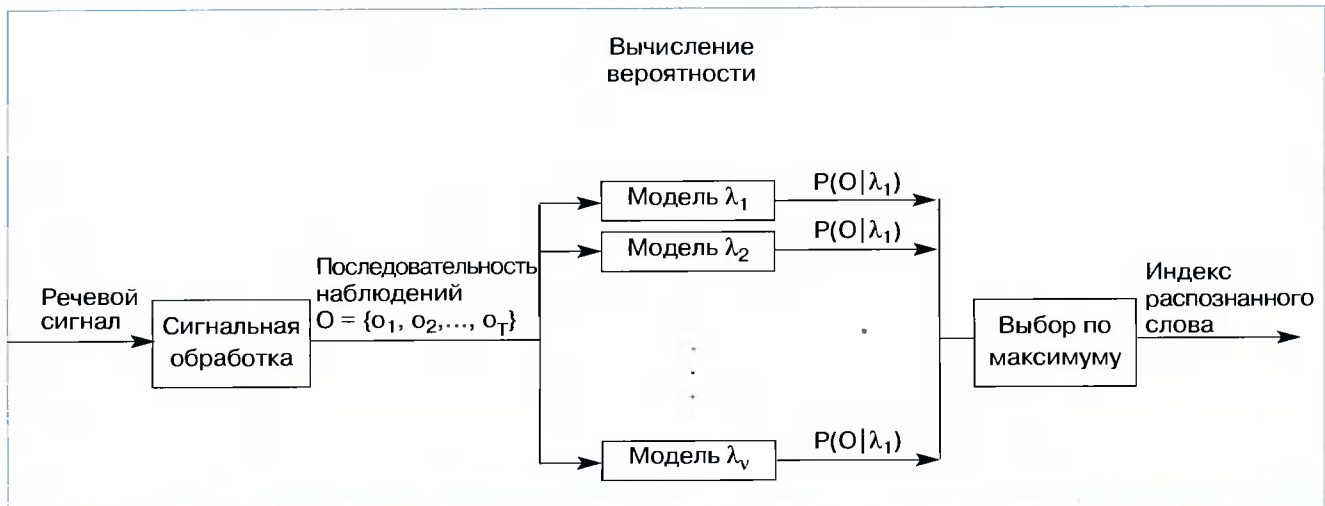


Рис. 2. Блок-схема распознавателя изолированных слов на основе СММ



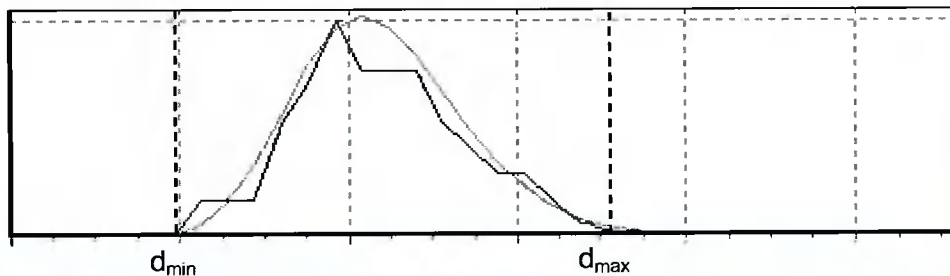


Рис. 3. Гистограмма плотности распределения длительности первого состояния слова «six»

дет пройдена траектория *ACDFG* и накоплена вероятность  $p = 0,6 \cdot 0,6 \cdot 0,6 \cdot 0,1 \cdot 0,6 = 0,01296$ . Тогда как наиболее оптимальной траекторией в этом случае будет траектория *ABDEG* с вероятностью  $p = 0,6 \cdot 0,3 \cdot 0,6 \cdot 0,6 \cdot 0,6 = 0,03888$ . Для устранения этой проблемы был разработан «модифицированный» алгоритм Витерби.

### Модифицированный алгоритм Витерби

В предлагаемой модификации алгоритма Витерби мы будем максимизировать не предыдущее состояние, а длительность текущего. Для этого определим величину  $\varphi_j^t(d)$ , которая представляет собой вероятность того, что состояние  $j$  окончилось в момент времени  $t$  и началось в момент времени  $t-d$ , т.е. имело продолжительность  $d$ :

$$\varphi_j^t(d) = \begin{cases} \prod_{t'=t-d}^t b_j(O_{t'}) \cdot (p_j(d))^\alpha \cdot \delta_{j-1}(t-d), & 1 < j \leq N, \\ \prod_{t'=t-d}^t b_j(O_{t'}) \cdot (p_j(d))^\alpha & j = 1, \end{cases} \quad (2.2)$$

где  $p_j(d)$  – плотность вероятности длительности состояния,

$\alpha$  – масштабный коэффициент для длительностей состояний,

$\delta_{j-1}(t-d)$  – вероятность, накопленная к моменту  $t-d$ .

Введем так же дополнительную переменную  $D_j(t)$ , обозначающую наиболее вероятную длительность состояния  $S_j$ , при условии его окончания в момент времени  $t$ .

Полную процедуру, требуемую для определения последовательности состояний, можно сформулировать следующим образом:

- 1) Инициализация:  
Для всех состояний  $j = 1 \dots N$ :

$$\delta_j(1) = \begin{cases} b_j(O_1), & j=1, \\ 0, & j>1. \end{cases} \quad (2.3a)$$

$$D_j(1) = 0. \quad (2.3b)$$

- 2) Рекурсия:  
Для всех состояний  $j = 1 \dots N$ :  
если  $j > 1$ :

$$dj(t) = \max_{t_s \leq d \leq t_e} \varphi_j^t(d), \quad (2.4a)$$

где  $t_s = \max(1, t - d_{j \max}), t_e = \max(1, t - d_{j \min})$ ;

$$D_j(t) = \arg \max_{t_s \leq d \leq t_e} \varphi_j^t(d); \quad (2.4b)$$

если  $j = 1$ :  
 $\delta_j(t) = \varphi_j^t(t), \quad (2.4c)$

$$D_j(t) = t \quad (2.4d)$$

- 3) Окончание:  
Итоговая вероятность:  
 $P(O) = \max_j \delta_j(T). \quad (2.5a)$

Последнее состояние:  
 $q(T) = \arg \max_j \delta_j(T). \quad (2.5b)$

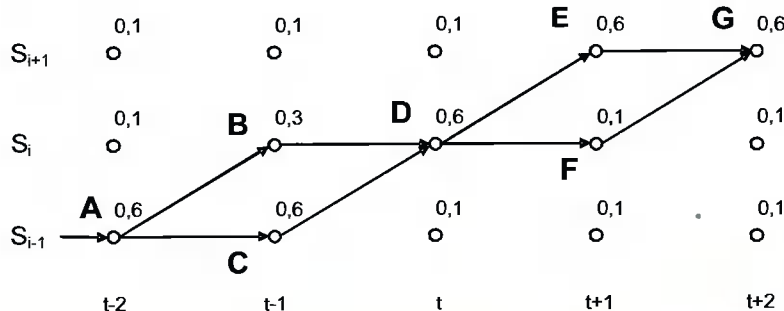


Рис. 4. Пример ошибочного декорирования вследствие ограничения на длительность состояния

4) Восстановление пути:

$$t = T, \quad (2.6a)$$

$$q = q(t), \quad (2.6b)$$

$$d = D_q(t), \quad (2.6c)$$

Пока  $d \neq 0$

$$q_{t-d} \dots q_t = q, \quad (2.6d)$$

$$q = q - 1, \quad (2.6e)$$

$$d = D_q(t). \quad (2.6f)$$

Шаг 1 устанавливает значение вероятности для первого состояния равной вероятности наблюдения символа  $O_j$ . Для остальных состояний вероятности устанавливаются в ноль. Индуктивный переход, в котором заключается основное содержание алгоритма, проиллюстрирован на рис. 5. Вероятность окончания состояния в данной точке складывается из совместной вероятности наблюдения символов  $O_{t-d}, \dots, O_t$  на интервале  $\{t - d, t\}$ , вероятности длительности состояния  $p(d)$  и вероятности окончания предыдущего состояния в точке  $t - d$ .

На заключительном этапе производится максимизация по наиболее вероятному последнему состоянию и от него восстанавливается вся последовательность состояний.

Далее в тексте, если это не оговорено особо, под алгоритмом Витерби будет пониматься именно модифицированный вариант.

**Алгоритм Витерби для цепочки связанных слов**

Поиск оптимальной последовательности состояний для цепочек связанных слов производится на всем пространстве состояний моделей. Помимо моделей слов, для повышения надежности распознавания и устранения неточностей определения граничных точек, добавляется модель паузы. Для модели паузы из вычислений исключаются плотности вероятностей длительности состояний  $p(d)$ . Таким образом из моделей отдельных слов формируется одна связанная СММ для всех слов, включая паузу (рис. 6).

Для организации переходов между словами вводится дополнительная переменная – вероятность конца слова в данной точке  $P_w(t)$ , равная вероятности конца последнего состояния в этой же точке. Переход в новое слово происходит из наиболее вероятного предыдущего слова. Для того, чтобы затем восстановить последовательность состояний, будем использовать массив  $\Psi_t(j)$ , в котором сохраняются ин-

декс состояния предшествующего данному. Кроме того, в массиве  $Pt_w(t)$  отдельно сохраняются индекс последнего состояния наиболее вероятного слова, завершившегося в точке  $t$ .

Для того, чтобы обеспечить связь моделей при переходе в новое слово, необходимо изменить процедуру вычисления величины  $\Phi_j^t(d)$  для первого состояния следующим образом:

$$\Phi_j^t(d) = \prod_{i=t-d}^t b_j(O_i) \cdot (p_j(d)) \alpha P_w(t-d), \quad j = 1. \quad (2.7)$$

Теперь алгоритм Витерби для связанных цепочек будет выглядеть следующим образом:

1) Инициализация:

$base = 0$  – индекс первого состояния слова,

Для всех слов  $W_k$ :

Для всех состояний  $j = 1 \dots N_k$ :

$$\delta_{j+base}(1) = \begin{cases} b_{j+base}(O_1), & j=1 \\ 0, & j>1 \end{cases}, \quad (2.8a)$$

$$D_{j+base}(1) = 0. \quad (2.8b)$$

$$base = base + N_k \quad (2.8c)$$

2) Рекурсия:

Вычисление вероятностей по состояниям:

$base = 0$ ,

Для всех слов  $W_k$ :

Для всех состояний  $j = 1 \dots N_k$ :

если  $j > 1$ :

$$\delta_{j+base}(t) = \max_{t_s \leq d \leq t_e} \Phi_{j+base}^t(d), \quad (2.9a)$$

$$D_{j+base}(t) = \arg \max_{t_s \leq d \leq t_e} \Phi_{j+base}^t(d), \quad (2.9b)$$

$$\Psi_t(j + base) = j + base - 1, \quad (2.9c)$$

если  $j = 1$ :

$$\delta_{j+base}(t) = \max_{t_s \leq d \leq t_e} \Phi_{j+base}^t(d), \quad (2.9d)$$

$$D_{j+base}(t) = \arg \max_{t_s \leq d \leq t_e} \Phi_{j+base}^t(d), \quad (2.9e)$$

$$\Psi_t(j + base) = St_w(t - D_{j+base}(t)), \quad (2.9f)$$

$$base = base + N_k \quad (2.9g)$$

Сохранение вероятности для наиболее вероятного слова  $W_{best}$ , окончившегося в данной точке:

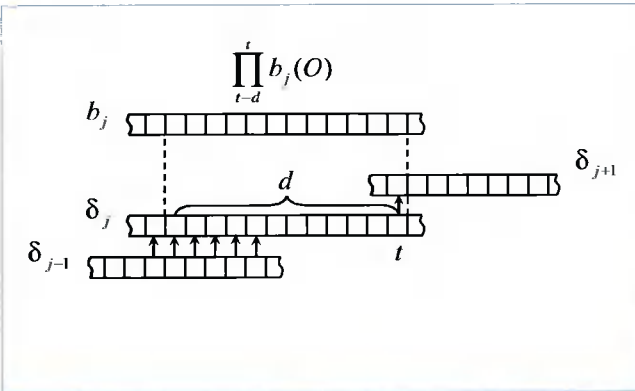


Рис. 5. Пример последовательности операций, необходимых для вычисления вероятности окончания состояния  $\varphi_j^t(d)$  в момент времени  $t$ , при условии его начала в момент  $t-d$

$$base = 0,$$

$$P_{best} = 0,$$

Для всех слов  $W_k$ :

если  $\delta_{base+N_k}(t) > P_{best}$

$$P_{best} = \delta_{base+N_k}(t), \quad (2.10a)$$

$$St_{best} = base + N_k, \quad (2.10b)$$

$$base = base + N_k; \quad (2.10c)$$

$$P_w(t) = P_{best}, \quad (2.10d)$$

$$St_w(t) = St_{best}. \quad (2.10e)$$

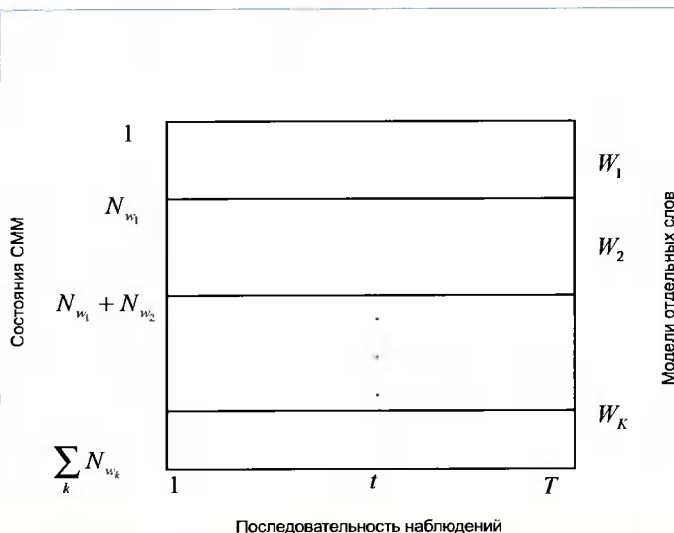


Рис. 6. Объединенная СММ для цепочки связанных слов

3) Окончание:

Итоговая вероятность:

$$P(O) = \max_j \delta_j(T). \quad (2.11a)$$

Последнее состояние:

$$q(T) = \arg \max_j \delta_j(T). \quad (2.11b)$$

4) Восстановление пути:

$$t = T, \quad (2.12a)$$

$$q = q(t), \quad (2.12b)$$

$$d = D_q(t), \quad (2.12c)$$

Пока  $d \neq 0$

$$q_{t-d} \dots q_t = q, \quad (2.12d)$$

$$t = t - d, \quad (2.12e)$$

$$d = D_q(t), \quad (2.12f)$$

$$q = \Psi_t(q). \quad (2.12g)$$

Из восстановленной последовательности состояний путем несложных вычислений восстанавливается последовательность слов.

### Вычислительные аспекты применения алгоритма Витерби

Поскольку для вычисления вероятности продолжительности состояния  $\varphi$  используется произведение вероятностей наблюдения символа  $b(O_t)$ , каждая из которых, как правило, существенно меньше единицы, то с ростом  $t$  вероятность  $\varphi$  начинает экспоненциально стремиться к нулю. Для устранения возможного переполнения необходимо либо ввести процедуру масштабирования, либо переходить к логарифмическим значениям вероятностей. Второй вариант более предпочтителен, поскольку позволяет для вычисления условных вероятностей заменить операцию умножение на сложение.

В логарифмическом виде формула вычисления вероятности длительности состояния будет иметь следующий вид:

$$\varphi_j^t(d) = \sum_{t'=t-d}^t \log(b_j(O_{t'})) + \alpha \cdot \log(p_j(d)) + \delta_j(t-d). \quad (2.13)$$

Логарифм априорной вероятности длительности состояния  $p_j(d)$  может быть рассчитан заранее на этапе подготовки моделей. Логарифмические

величины вероятностей наблюдения символа  $b(O_t)$  удобно вычислить предварительно и сохранить в массиве следующим образом:

$$b^{log}(1) = \log(b(O_1)), \text{ для } t = 1, \quad (2.14a)$$

и далее рекуррентно

$$b^{log}(t+1) = b^{log}(t) + \log(b(O_{t+1})). \quad (2.14b)$$

В таком случае вычисление  $\varphi$  становится еще более удобным:

$$\varphi_j^t(d) = b^{log}(t) - b^{log}(t-d) + \alpha \cdot p_j^{log}(d) + \delta_j(t-d). \quad (2.15)$$

### Алгоритм построения уровней

В рассмотренном выше методе предполагается декодирование цепочки команд произвольной длины. Достоинством этого метода является относительная простота реализации и универсальность применения.

Рассмотрим возможные типы ошибок, возникающие при таком подходе.

1. Ошибка типа замена (одна команда заменяется другой).
2. В случае быстрого слитного произнесения команд возможно объединение двух команд в одну (ошибка типа удаление).
3. При медленном произнесении возможно разделение команды на две (ошибка типа вставка).
4. В условиях сильных нестационарных шумов возможны вставки на месте паузы в районе высокой энергии речеподобного шума.

Существует широкий класс приложений, где можно существенно уменьшить количество ошибок типа 2 – 4. Это возможно в случае, когда заранее известна некая информация о количестве слов в строке. Длина строки может быть жестко фиксирована, (например, в случае голосового ввода PIN-кода), иметь верхние и нижние ограничения или некий фиксированный набор разрешенных значений (например, в случае голосового ввода телефонного номера).

Как можно корректно использовать эту информацию? Можно заметить, что в рассматриваемом выше подходе узлами промежуточной максимизации вероятностей, являются точки перехода из команд в паузу

или следующую команду. Таким образом при прямом проходе распознавания ошибки типа 2 – 4 неустраняемо вносятся в путь декодирования. Для исправления этой ситуации предлагается разделить узлы максимизации для каждого уровня.

Под уровнем  $L$  здесь и далее будем понимать состояние системы, содержащее  $L$  завершенных команд, в момент от начала сигнала до текущего момента времени  $t$ . Вероятность нахождения системы в момент времени  $t$  на уровне  $L$  обозначим как  $P_L(t)$ .

В данном подходе целесообразно рассматривать модель для каждой команды как независимый «черный ящик», на вход которого в каждый момент времени  $t$  подается вектор речевых параметров, а на выходе – массив абсолютных вероятностей  $P_w(d)$ , соответствующих всем возможным длительностям  $d$  команды  $w$ .  $P_w(d)$  вычисляется стандартным (пункт 1.3) или модифицированным (пункт 2.2) алгоритмом Витерби при условии, что вероятность входа в первое состояние модели не зависит от текущего состояния системы и всегда равна 1. Все модели команд просчитываются в режиме реального времени независимо. При этом не происходит дублирования процедуры декодирования команды для каждого уровня. В общем случае, внутренняя структура моделей значения не имеет и запоминание путей декодирования внутри модели может не вестись.

Введем вероятность выхода из модели  $w$  в момент времени  $t$  на уровне  $L$ :

$$P_{wL}(t) = \max_d (P_w(d) \cdot P_{L-1}(t-d)). \quad (2.16)$$

Тогда вероятность нахождения системы в момент времени  $t$  на уровне  $L$  будет вычисляться как:

$$P_L(t) = \max_w \left[ \max(P_{wL}(t)), P_L(t-1) \cdot B_{\text{pause}}(t) \right], \quad (2.17)$$

где  $B_{\text{pause}}(t)$  – вероятность паузы в момент времени  $t$ , а член  $P_L(t-1) \cdot B_{\text{pause}}(t)$  соответствует вероятности наблюдения паузы на уровне  $L$  (уровень при переходе в паузу не меняется). Здесь для упрощения не учитываются (полагаются равными 1) переходные вероятности на уровне команд. В реальном применении учет этих вероятностей также можно опустить. Схема вычисления нового значения  $P_L(t)$  для одного уровня представлена на рис. 7.

Двигаясь по шкале времени, мы получаем значения  $P_L(t)$  для всех уровней от 0 до максимального значения  $L_{\text{max}}$ . При этом мы можем определить наиболее вероятное количество команд в строке как

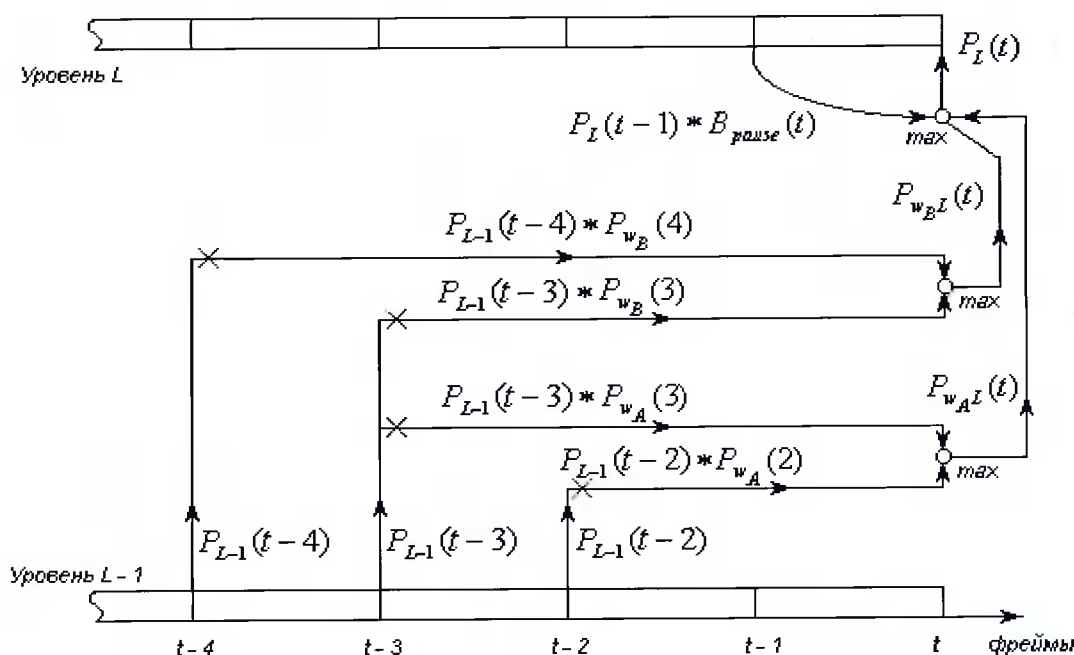


Рис. 7. Пример вычисления вероятностей по схеме построения уровней. Здесь предполагается, что существуют только 2 команды:  $w_A$  и  $w_B$  с разрешенными длительностями 2..3 и 3..4 соответственно

$$L_{best}(t) = \arg \max_L P_L(t) \quad (3.18)$$

В случае режима реального времени ввод сигнала и процесс распознавания прекращаются при выполнении условия равенства  $L_{best}(t)$  желаемому количеству команд. В случае мягких ограничений на длину строки необходимо использовать более сложные алгоритмы остановки процесса распознавания, рассмотрение которых выходит за рамки этой статьи.

После остановки процесса распознавания (при выполнении указанного выше условия или по достижению конца файла) на этапе обратного декодирования все уровни, кроме желаемого, игнорируются.

Следует отметить возможность использования в данном подходе маски ввода, т.е. когда часть команд известна. В частности, этот алгоритм использовался для автоматической разметки в большей части тренировочной базы данных.

## Заключение

В работе рассмотрены основные принципы распознавания речи на базе Скрытых Марковских моделей и методы их использования при построении SPIRIT ASR-engine. Описаны основные алгоритмы, при-

меняемые в разработанной системе распознавания речевых команд и связной речи, которые используют как классические методы, так и нестандартные подходы при декодировании и оптимизации.

Результаты работы SPIRIT ASR-engine показывают на примере распознавания английских цифр высокую эффективность изложенных подходов в решении задач распознавания изолированных и слитно произнесенных команд в дикторнезависимом режиме.

В следующей статье мы планируем затронуть проблему устойчивой работы систем распознавания в условиях зашумленной акустической обстановки. Будут рассмотрены методы и практические результаты решения этой проблемы.

## Литература

1. Л.Р. Рабинер, «Скрытые марковские модели и их применение в избранных приложениях при распознавании речи», ТИИЭР, т. 77, №2, февраль 1989.
2. L.R. Rabiner, B-H Young. Fundamentals of the speech recognition. Prentice Hall, Englewood Cliffs, NJ, 1993.
3. M. Gales, «The Theory of Segmental Hidden Markov Models», Cambridge University, 1993.

УДК 621.396.96

## Цифровая фильтрация частотно - модулированных импульсов в многофункциональных РЛС

А.К. Пяткин, М.В. Никитин

### Введение

Как известно [1], идея использования модулированных зондирующих сигналов появилась еще на этапе становления радиолокации в середине прошлого века. Использование модуляции позволяет одновременно получать широкий спектр сигнала, необходимый для обеспечения высокого разрешения радиолокационных объектов по дальности и большую длительность сигнала, необходимую для увеличения энергетического потенциала РЛС при ограниченной импульсной мощности передатчика.

В силу специфики построения передатчиков для импульсных когерентных РЛС, заключающейся в работе выходного и промежуточных каскадов в нелинейном режиме с отсечкой, а также для сохранения постоянного уровня мощности зондирующего сигнала за время его излучения, используется частотная модуляция импульса, простейший случай которой – линейная частотная модуляция (ЛЧМ).

Согласованный с ЛЧМ-импульсом фильтр производит сжатие сигнала до значения, определяемого шириной его спектральной полосы (девиации ЛЧМ – импульса) [2]. До настоящего времени, для согласованной обработки ЛЧМ-импульсов широко использовались аналоговые дисперсионные линии задержки и ПАВ-фильтры [3].

Тенденция развития радиолокационной техники [4] учитывает известные успехи в создании элементной базы для реализации алгоритмов цифровой обработки. В частности, это программируемые логические интегральные схемы (ПЛИС) [5], и аналого-ци-

*Рассматриваются вопросы, связанные с аппаратной реализацией цифрового сжатия частотно-модулированных импульсов. Предложен алгоритм адаптивного синтеза коэффициентов квазисогласованного фильтра для ЛЧМ - импульса с низким уровнем боковых лепестков сигнала с учетом искажений сигнала в аналоговых трактах. Рассматриваются перспективы использования нелинейных законов модуляции с учетом возможностей цифрового синтеза сигналов.*

фровые преобразователи (АЦП) [6], что позволяет сегодня ставить вопрос об использовании цифровой фильтрации ЛЧМ-импульсов (Таблица 1). Это дает возможность реализации преимуществ, которые дают цифровые методы фильтрации над аналоговыми методами [7].

### Аппаратная реализация

Использование ПЛИС в качестве цифровой элементной базы для реализации фильтра для сжатия ЛЧМ-импульсов с девиацией порядка единиц и десятков мегагерц на данный момент времени, представляется наиболее целесообразным. Использование ПЛИС дает возможность построения параллельных вычислительных систем в составе одного кристалла, что с одной стороны, необходимо для обработки высокоскоростных потоков данных с относительно высокой девиацией ЛЧМ-импульсов (Таблица 1), использующихся в современных многофункциональных РЛС (МРЛС). С другой стороны, применение ПЛИС существенно удешевляет и уменьшает габариты и потребляемую мощность устройства цифровой обработки РЛС по сравнению с эквивалентной по производительности системы на основе процессоров ЦОС.

**Таблица 1. Аппаратные характеристики квазисогласованного фильтра для ЛЧМ-импульса**

Длительность ЛЧМ-импульса, мкс	15
Девияция ЛЧМ-импульса, МГц	10
Разрядность входного сигнала, бит	14
Тип кристалла ПЛИС	FPGA «Xilinx» XCV600tE
Используемая часть ПЛИС	Менее 30%

При разработке фильтра для сжатия ЛЧМ-импульса для тракта сигнальной обработки МРЛС необходимо минимизировать как аппаратные затраты, так и энергетические потери в отношении сигнал/шум, возникающие при проведении вычислений. В [8] показано, что оптимальным с точки зрения вычислительных затрат, является алгоритм согласованной фильтрации ЛЧМ-импульса в частотной области с применением высокоэффективного алгоритма быстрого преобразования Фурье (БПФ) [7]. При этом производится фильтрация аналитического (комплексного) видеосигнала (сигнала с нулевой несущей частотой) [4]. Это позволяет согласовать сигнал с вычислителем БПФ, производящим операции с комплексными числами, что в несколько раз сокращает вычислительные затраты по сравнению с фильтрацией действительного сигнала на промежуточной частоте. В [8] также рассматриваются вопросы выбора оптимальной, с точки зрения вычислительных затрат, базы БПФ –  $N$  в зависимости от базы ЛЧМ-импульса.

В [9] разработан принцип построения ядра вычислителя БПФ – «бабочки» на ПЛИС и приведены формулы, которые позволяют оценить оптимальную разрядность ядра и коэффициентов БПФ. Благодаря этому, можно получить уровень энергетических потерь в отношении сигнал/шум при построении фильтра в частотной области, обусловленный усечением данных на выходах арифметических элементов «бабочки» и конечной разрядностью коэффициентов БПФ, не более 0,2 дБ.

### Использование весовых функций

Для подавления боковых лепестков сигнала на выходе оптимального фильтра, что необходимо для разрешения близкорасположенных целей в широком динамическом диапазоне, может использоваться весовая функция (окно) в частотной области [3]. В [8] для этого предлагается использовать окно Хемминга, позволяющее, в идеальном случае, получить уровень боковых лепестков на уровне – 42 дБ.

Возможно использование функции из семейства параметрических окон Кайзера, которые наиболее эффективны с точки зрения наименьшего уровня боковых лепестков при фиксированной ширине главного лепестка [7]. Изменение параметра в формуле для окна Кайзера позволяет регулировать желаемый уровень боковых лепестков сигнала на выходе фильтра.

Недостатком весовых функций Хемминга и Кайзера является относительно низкая – 6 дБ/октава (Таблица 2), скорость спада боковых лепестков в их спектре. При этом для задач разрешения сигналов, оптимальной, по некоторым критериям [2], является оконная функция, приводящая форму сигнала на выходе фильтра к форме немодулированного сигнала на выходе его оптимального фильтра, т.е. вида

$$\left(\frac{\sin(x)}{x}\right)^2$$

в частотной области. Таким образом, для ЛЧМ-импульса – это весовая функция вида:

$$W(i) = \frac{\left(\sin\left(\frac{2\pi}{N}\left(i - \frac{N}{2}\right)\right)\right)^2}{\frac{2\pi}{N}\left(i - \frac{N}{2}\right)}, \quad i = 0 \dots N - 1. \quad (1)$$

Весовая функция (1) дает уровень первого бокового лепестка около -40 дБ и скорость спада боковых лепестков 12 дБ/октава. На рис. 1 показаны спектры окон Хемминга, Кайзера и окна (1), которые, если принять, что амплитудная характеристика ЛЧМ-импульса – идеально прямоугольная, точно совпадают с формой сигнала на выходе фильтра с соответствующей весовой функцией.

Очевидно, что выбор оптимальной весовой функции зависит от динамических характеристик всего тракта обработки, требований на характеристики РЛС и т.д. Поэтому решение данного вопроса в каждом индивидуальном случае требует отдельного подхода.

Применение весовой функции приводит к расширению главного лепестка сигнала на выходе филь-

**Таблица 2. Характеристики квазисогласованной фильтрации ЛЧМ-импульса при различных весовых окнах**

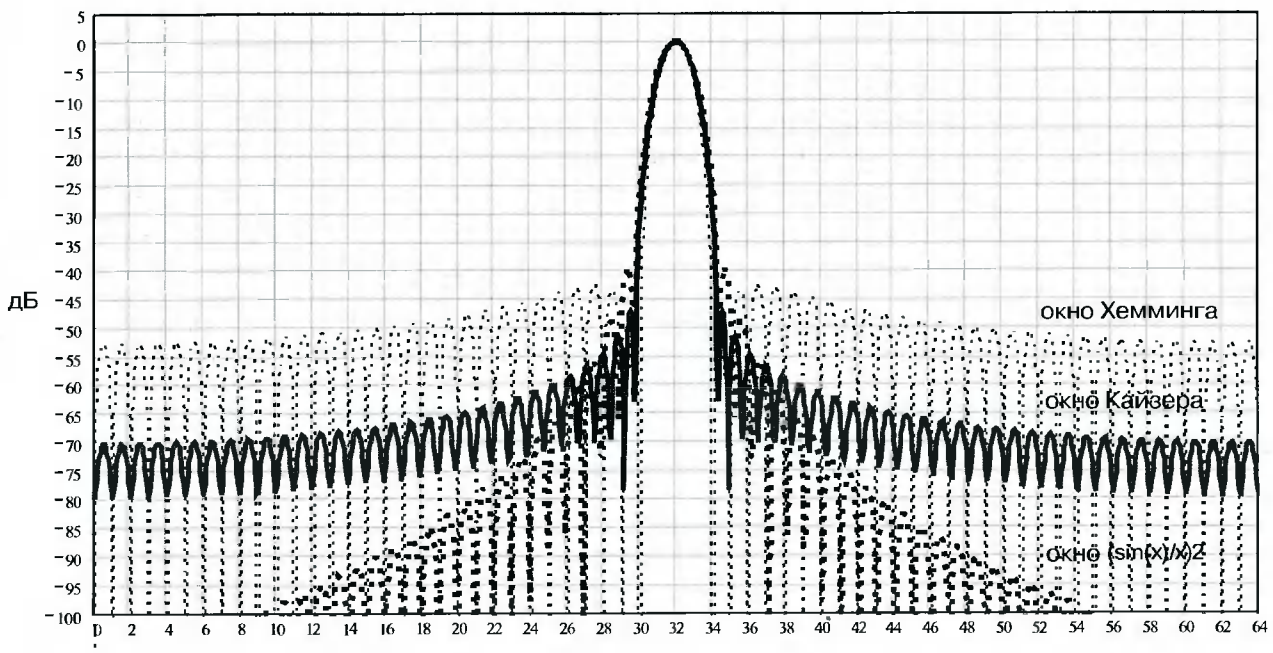
Окно	Уровень первого бокового лепестка, дБ	Энергетические потери, дБ	Расширение главного лепестка (по уровню -3 дБ)	Скорость спада боковых лепестков, дБ/октава
прямоугольное	-13	0	1	6
Хемминга	-42	1,3	1,47	6
Кайзера	$\beta = 4$	0,9	1,27	6
	$\beta = 4,8$	1,2	1,39	6
	$\beta = 5,4$	1,5	1,46	6
	$\beta = 6,0$	1,7	1,53	6
	$\beta = 6,8$	1,9	1,60	6
$\beta = 7,5$	2,1	1,67	1,67	6
$(\sin(x)/x)^2$	-40	2,1	1,67	12

тра, что с одной стороны ухудшает разрешение РЛС по дальности и точность измерения дальности, с другой ведет к энергетическим потерям в отношении сигнал/шум [2] по сравнению с согласованной обработкой. Поэтому такой алгоритм фильтрации называется квазисогласованным. В таблице 2 приведены характеристики сжатых сигналов (спектров весовых функций) в зависимости от выбора той или иной весовой функции.

#### Алгоритмы синтеза коэффициентов фильтра в частотной области

Простейший алгоритм синтеза коэффициентов фильтра  $A(i)$  в частотной области [8] осуществляется по формуле:

$$A(i) = S^*(i) \cdot W(i), \quad (2)$$



**Рис. 1. Спектры некоторых весовых функций**



где  $S^*(i)$  – комплексно-сопряженный спектр ЛЧМ - импульса,

$W(i)$  – весовая функция.

Однако, при этом может наблюдаться значительное увеличение уровня боковых лепестков сигнала по сравнению с боковыми лепестками весовой функции, из-за искажений модуляционной характеристики при прохождении сигналом приемо-передающих аналоговых трактов [8]. Линейные искажения (искажения амплитудной и фазовой характеристик сигнала) в реальных аналоговых трактах, а также неравномерность амплитудной характеристики ограниченного по длительности ЛЧМ-импульса (рис.3), на практике, дают минимально достижимый уровень боковых лепестков сигнала на уровне -30...-35 дБ во всем диапазоне доплеровских частот [2,4].

Принимая во внимание возможности цифровой фильтрации по коррекции амплитудных и фазовых искажений сигналов, представляется целесообразным синтез коэффициентов с учетом искажений по формуле:

$$A(i) = \frac{e^{-j\Psi(i)}}{|S(i)|} \cdot W(i), \quad (3)$$

где  $\Psi(i)$  – фазовая характеристика зондирующего ЛЧМ-импульса на выходе аналогового тракта,  $|S(i)|$  – амплитудная характеристика ЛЧМ-импульса на выходе аналогового тракта.

Проиллюстрируем «работу формулы» (3) следующими рассуждениями. Согласованный фильтр должен компенсировать фазовый спектр сигнала [2]. Это осуществляет множитель  $e^{-j\Psi(i)}$ , при этом происходит как компенсация (выпрямление) квадратичной фазовой характеристики ЛЧМ-импульса, так и фазовых искажений. Делитель  $|S(i)|$  производит компенсацию амплитудных искажений сигнала и неравномерность амплитудной характеристики ЛЧМ-импульса (рис.3). При этом форма сигнала на выходе квазисогласованного фильтра (без учета аддитивного шума) с коэффициентами, подсчитанными по формуле (3), в точности совпадает со спектром весовой функции  $W(i)$  (рис. 1).

С помощью формулы (3) можно производить адаптивный синтез коэффициентов фильтра, позволяющий «подстроиться» под конкретный аналоговый тракт. Интересно, что подобная коррекция позволяет также бороться с отражениями сигнала в аналоговом формирователе полосы сигнала – ПАВ-фильтре [4]. При этом происходит добавление отраженной энергии в главный лепесток сигнала на выходе цифрового фильтра. Однако это влечет за

собой существенное увеличение (в 3-5 раз) базы фильтра, что негативно отражается на затрачиваемых вычислительных ресурсах и не всегда является оправданным. Тем не менее, представляется целесообразным отметить данную возможность, предоставляемую цифровым фильтром.

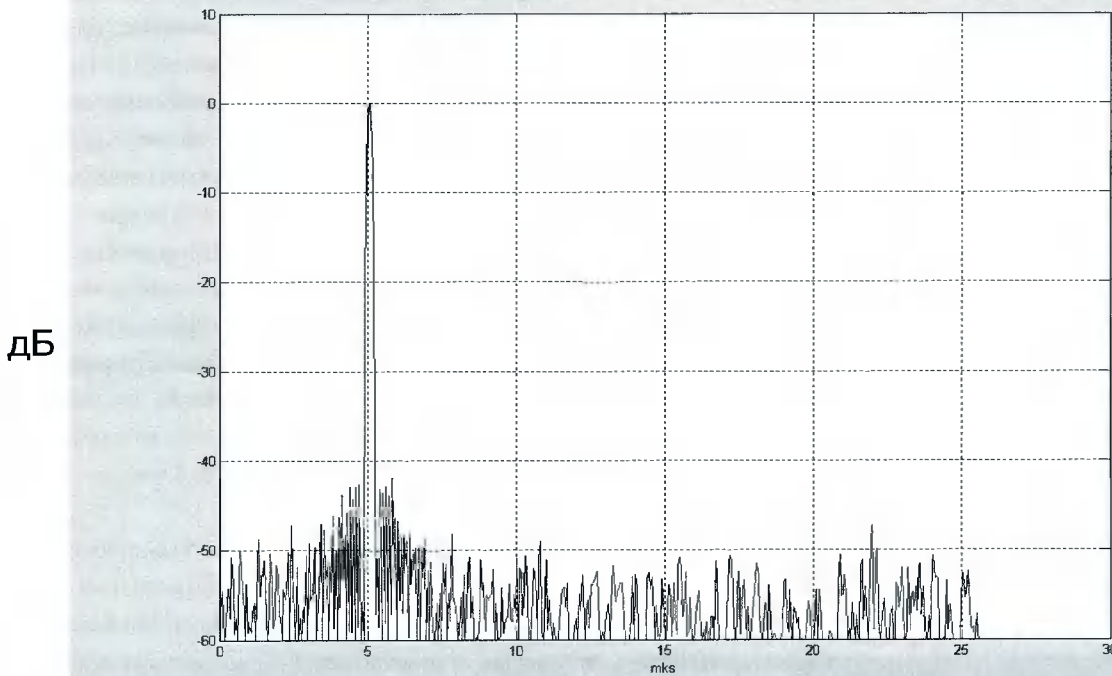
Опыт практического применения формулы (3) показывает, что при нулевой (или фиксированной) доплеровской частоте сигнала можно добиваться практически любого уровня боковых лепестков сигнала на выходе цифрового фильтра, причем форма сигнала, а, следовательно, и уровень боковых лепестков определяется только выбором  $W(i)$ .

Синтез коэффициентов фильтра по формуле (3), по-видимому, полностью оправдывает себя при работе с низкоскоростными (наземными или надводными) целями, дающими несущественный, по сравнению с девиацией сигнала, доплеровский сдвиг. Однако значительный доплеровский сдвиг сигнала приводит к «размыванию» корректирующих свойств формулы (3) и повышению уровня боковых лепестков. Тем не менее, таким способом удалось получить уровень боковых лепестков -40 дБ (рис.2) во всем диапазоне доплеровских частот при максимальной доплеровской частоте эхо-сигнала, не превышающей 2% от девиации сигнала. При этом уровень боковых лепестков мощных сигналов, отраженных от местных предметов, которым соответствует нулевая доплеровская частота (на которой производилась коррекция) составил менее -50 дБ.

Коррекция искажений сигнала по формуле (3), производимая с целью получения низкого уровня боковых лепестков, ведет к дополнительному увеличению энергетических потерь (Таблица 3). Однако, как показали измерения, проведенные с помощью натуральных экспериментов, уменьшение энергии полезного сигнала на выходе фильтра, в нашем случае, составило не более 0,2 дБ. Это можно объяснить тем фактом, что цифровая коррекция затрагивает только «тонкую структуру» сигнала, практически не влияющую на энергию сигнала, но определяющую уровень боковых лепестков.

### Использование нелинейной модуляции

Существенные энергетические потери (до 2 дБ) в отношении сигнал/шум из-за применения весовой функции (Таблица 3) при квазисогласованной фильтрации ЛЧМ-импульса и ярко выраженная зависимость уровня боковых лепестков сигнала от доплеровского смещения, делают актуальным вопрос, связанный с использованием нелинейной частотной мо-



**Рис. 2. Результат сжатия ЛЧМ-импульса с девиацией 10 МГц, длительностью 15 мкс и доплеровским смещением 100 кГц, проведенного на кристалле «Хіііпх» XCV600E**

дуляции (НЧМ) зондирующего импульса. Актуальность использования НЧМ-импульса подкрепляется также фактом развития технологий цифро-аналоговых преобразователей (ЦАП) и микросхем прямого цифрового синтеза сигналов [6].

Цифровой синтез позволяет получать аналоговый сигнал с большой точностью (до 10 – 14 бит), что дает принципиально новые возможности в его обработке.

Рассмотрим требования к параметрам синтезируемого НЧМ-импульса и проблемы, которые возникают на этапе синтеза, а также возможные пути решения задачи синтеза. Синтезируемый НЧМ-импульс дол-

жен обладать характеристиками, позволяющими, с одной стороны, производить согласованную фильтрацию с низким уровнем боковых лепестков (<-40 дБ) без применения весовых функций, с другой стороны, получать распределение боковых лепестков на выходе согласованного фильтра, слабо зависящее от доплеровского смещения сигнала.

Для решения задачи синтеза необходимо было бы оптимизировать НЧМ-импульс по минимуму изменения уровня боковых лепестков в его двумерной функции неопределенности от доплеровского смещения. Однако результаты моделирования показывают, что уровень боковых лепестков сигнала на выходе филь-

**Таблица 3. Суммарные энергетические потери при цифровой квазисогласованной фильтрации ЛЧМ-импульса**

Источник потерь	Энергетические потери, дБ
Конечная разрядность	0,2
Весовая функция (окно Кайзера)	2,0
Коррекция искажений	0,2
Всего	2,4

тра во всем диапазоне доплеровских частот, без учета параметров весовой функции, определяется «гладкостью» амплитудной характеристики сигнала (рис.3) и не зависит от фазовой характеристики сигнала [2]. Это условие существенно облегчает задачу синтеза.

Доплеровский сдвиг частоты сигнала привносит дополнительный линейный фазовый множитель в спектр отраженного сигнала [1], что не мешает компенсировать (выпрямлять) фазовую характеристику сигнала, производимую согласованным фильтром, и приводит только к известному эффекту «перехода скорости в дальность» [1] при доплеровских сдвигах частотно-модулированных сигналов.

Требование на «гладкость» амплитудной характеристики можно объяснить следующим образом: если предположить, что достигнута амплитудная характеристика сигнала, равная  $\sqrt{W(i)}$ , то это автоматически дает на выходе согласованного фильтра форму сигнала совпадающего с  $W(i)$  во временной области (рис. 1). При этом условии, как показывает моделирование, при доплеровской расстройке до 10% от девиации сигнала, рост боковых лепестков составляет 3...5 дБ, чем можно практически пренебречь.

Становится понятным, что именно неравномерность (осцилляции) (рис.3) амплитудной характеристики сигнала (известная в зарубежной литературе как «frequency ripple» – частотная рябь [3]), в основном, «ответственна» за рост бо-

ковых лепестков сигнала при доплеровском смещении сигнала.

Таким образом, достаточное условие для синтеза НЧМ-импульса, удовлетворяющего вышеизложенные требования, сводится к обеспечению модуля его спектра, равного  $\sqrt{W(i)}$  (скругленный спектр [2]), с минимальным уровнем неравномерности, требования на которую должны исходить из требования на необходимый уровень боковых лепестков. При этом фазовая характеристика может быть произвольной, что и дает необходимую степень свободы при синтезе сигнала.

Учитывая, что для сохранения энергетического режима передатчика МРЛС, амплитуда сигнала за время излучения не должна изменяться, уравнение НЧМ-импульса  $s(t)$  с длительностью  $T$ :

$$s(t) = \sin(2\pi\omega(t)t), \quad 0 \leq t < T, \quad (4)$$

где  $\omega(t)$  – модуляционная характеристика сигнала.

При этом задача синтеза НЧМ-импульса сводится к поиску оптимальной функции  $\omega(t)$ , приводящей амплитудную характеристику сигнала (4) к  $\sqrt{W(i)}$ . Для ЛЧМ-импульса функция  $\omega(t)$  – линейна (рис.4).

Разработаны алгоритмы, например с использованием принципа стационарной фазы для получения модуляционных характеристик НЧМ-импульсов с некоторой точностью, зависящей от базы сигнала [3]. Однако современные возможности цифрового синте-

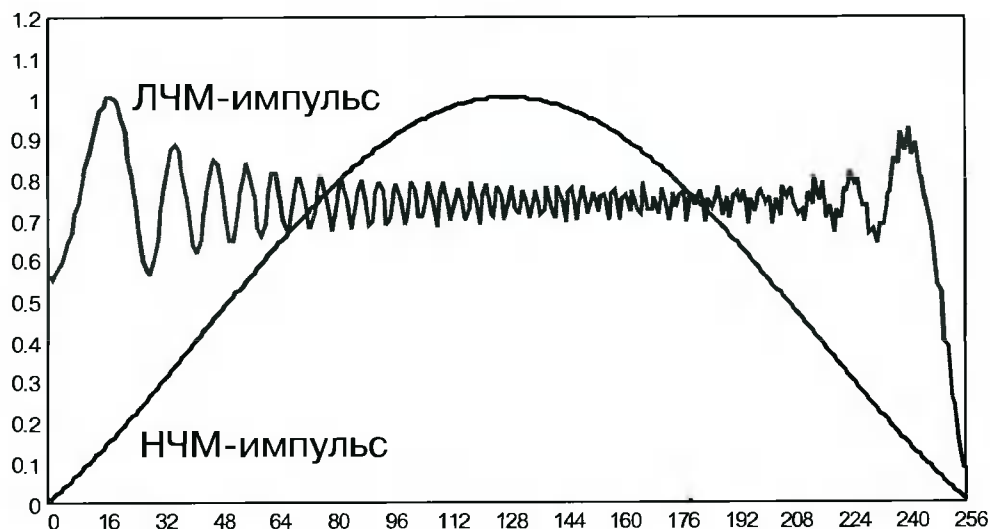


Рис. 3. Амплитудные характеристики ЛЧМ - и НЧМ-импульсов

за позволяют предъявить более жесткие требования к точности характеристик сигнала, соблюдение которых дает возможность получения более низкого уровня боковых лепестков, чем получали ранее при использовании аналоговых фильтров [3].

Учитывая также нелинейные преобразования сигнала при изменении его несущей частоты и усилении, проявляющиеся в виде появления интермодуляционных составляющих в полосе сигнала, представляется целесообразным поиск оптимальной функции  $\omega(t)$  (рис.4), путем перебора, с использованием вычислительных средств, возможных модуляционных характеристик в некоторой заданной окрестности. Далее, производится выбор модуляционной характеристики по критерию минимума среднеквадратичного отклонения (СКО) амплитудной характеристики полученного сигнала от  $\sqrt{W(i)}$ .

### Заключение

Развитие технологий цифрового синтеза и цифрового сжатия частотно-модулированных импульсов позволяют сегодня реализовать в аппаратуре МРЛС известные преимущества цифровых методов фильтрации над аналоговыми методами, например, адаптивность, повторяемость, точность и т.д.

При этом комбинация данных свойств согласованной цифровой обработки дает новый качественный уровень в характеристиках сигнала на выходе устройства сжатия. И если ранее, при использовании аналоговых фильтров уровень боковых лепестков -40 дБ считался «слишком академичным» [3],

то сегодня, используя цифровые фильтры сжатия и цифровые алгоритмы синтеза, достижение уровня -50 дБ и ниже при минимальных энергетических потерях, представляет собой реальную возможность (рис.2).

### Литература

1. Я.Д. Ширман «Теоретические основы радиолокации», М.: «Советское радио», 1970.
2. Я.Д. Ширман «Разрешение и сжатие сигналов», М.: «Советское радио», 1974.
3. Ч.Кук, М.Бернфельд «Радиолокационные сигналы. Теория и применение», М.: «Советское радио», 1971.
4. Д.Ю. Бобров, А.П. Доброжанский, Г.В. Зайцев, Ю.В. Маликов, И.Б. Цыпин «Цифровая обработка сигналов в многофункциональных РЛС», М.: журнал «Цифровая Обработка Сигналов», №4/2001.
5. <http://www.xilinx.com>
6. <http://www.analogdevices.com>
7. Л. Рабинер, Б. Гоулд «Теория и применение цифровой обработки сигналов», М.: «Мир», 1978.
8. Д.Ю. Бобров, А.П. Доброжанский, Г.В. Зайцев, Ю.В. Маликов, И.Б. Цыпин «Цифровая обработка сигналов в многофункциональных РЛС: ч. 2 – Алгоритмы обработки радиолокационных сигналов», М.: журнал «Цифровая Обработка Сигналов» №1/2002.
9. А.К. Пяткин, М.В. Никитин «Реализация быстрого преобразования Фурье (БПФ) для алгоритмов ЦОС в многофункциональных РЛС», М.: журнал «Цифровая Обработка Сигналов» №3/2003.

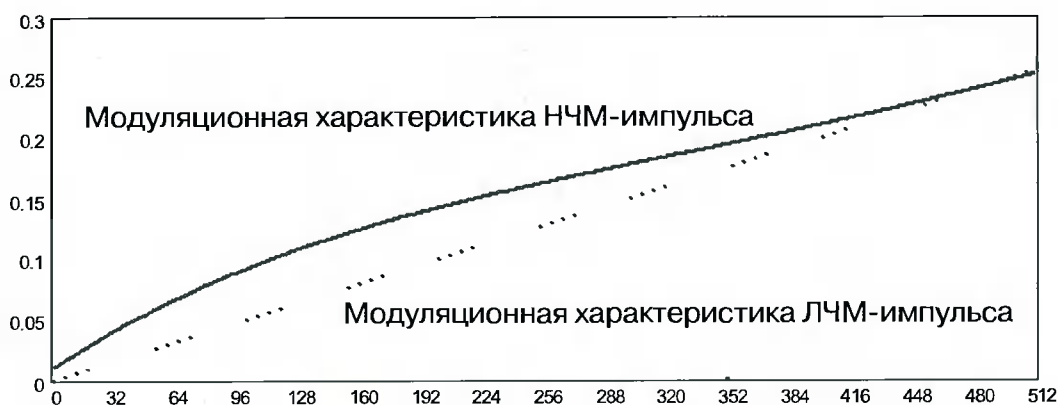


Рис. 4. Модуляционные характеристики ЛЧМ - и НЧМ-импульсов

УДК 681.3.06

## Обработка и анализ радиосигналов на основе многофункционального модуля цифровой обработки сигналов XDSP-3МС

М.Н. Быканов, С.М. Седунов, В.А. Толмачев

Одной из основных задач, решаемых при обработке радиосигналов в системах радиоэлектронного мониторинга, является измерение их частотно-временных параметров в реальном времени с требуемыми характеристиками по диапазону и точности.

Для обнаружения сигналов и получения оценок их параметров, в настоящее время широко используется прямое и обратное дискретное преобразование Фурье (ДПФ) [1]. Выполнение ДПФ даже в случае быстрых алгоритмов (БПФ) характеризуется большими вычислительными затратами, что является одним из наиболее существенных факторов, ограничивающих возможности создания цифровой аппаратуры анализа с высокой пропускной способностью ( $10^8 - 10^{10}$  бит/с) и широкой полосой анализируемых частот (десятки МГц). В связи с этим, при создании высокопроизводительных вычислительных систем, обеспечивающих обработку радиосигналов в реальном времени, необходимо реализовывать комплексный подход к выбору архитектуры построения аппаратно-программной части, учитывающий требования по: полосе анализируемых частот, виду и параметрам анализируемых сигналов, результатам обработки.

Наиболее перспективным направлением построения цифровых анализаторов сигналов является комплексирование в одном устройстве компонент, специально предназначенных для решения задач конвейерной беспрерывной обработки, к которым относятся: элементы тракта

Рассмотрен подход к построению цифрового анализатора радиосигналов на основе модуля XDSP-3МС. Приведены результаты исследования вычислительной эффективности различных алгоритмов быстрого преобразования Фурье на ЦСП модуля обработки сигналов.

аналого-цифрового преобразования, массивы буферной памяти, среды обработки данных на основе программируемых интегральных схем (ПЛИС) и цифровых сигнальных процессоров (ЦСП), а также элементы высокоскоростных интерфейсов. При этом на основе ПЛИС целесообразно выполнять алгоритмы потоковой обработки данных в беспрерывном режиме (обнаружение сигнала на фоне шума, фильтрация, корреляционная обработка, сжатие), а на основе ЦСП процедуры измерения, уточнения параметров сигналов и дополнительного анализа (демодуляция, анализ внутриимпульсной структуры, селекция/режекция сигналов).

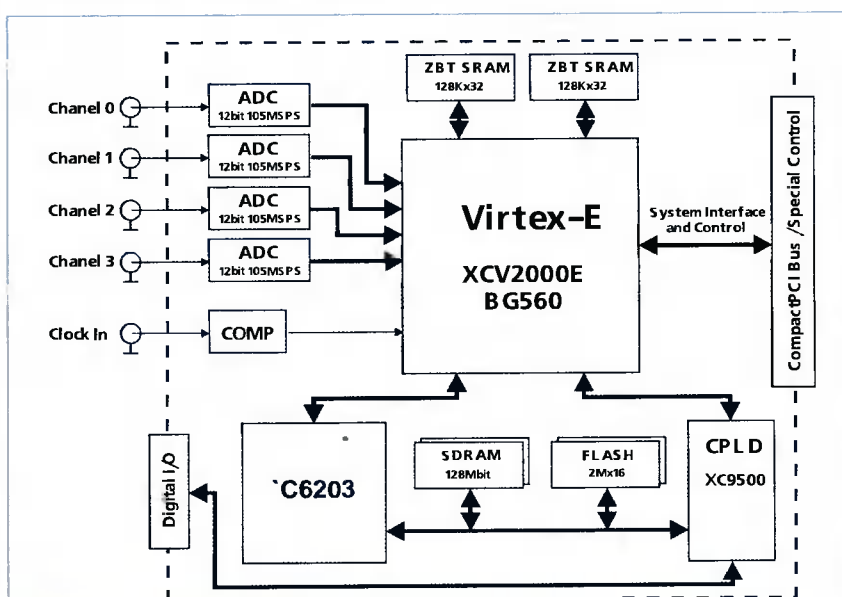


Рис. 1. Структурная схема модуля XDSP-3МС

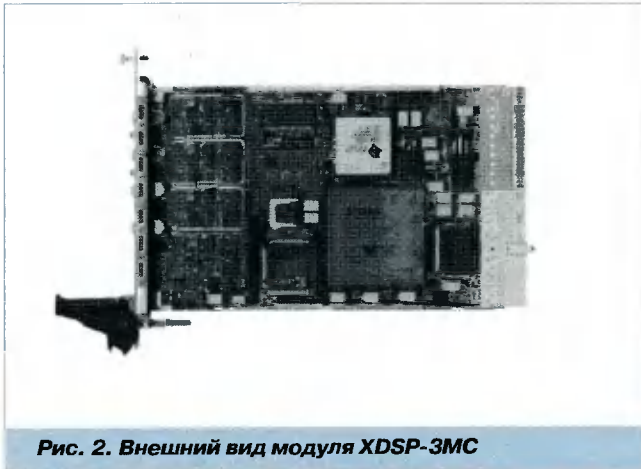


Рис. 2. Внешний вид модуля XDSP-3MC

Указанная архитектура реализована в многофункциональном модуле аналого-цифрового преобразования, обработки и анализа сигналов XDSP-3MC [2,3] разработанном ЗАО «СКАН Инжиниринг Телеком».

На рис. 1 и 2 представлены структурная схема и внешний вид модуля XDSP-3MC.

Основные технические характеристики модуля XDSP-3MC:

1. Количество каналов аналого-цифрового преобразования - 4;
2. Тактовая частота и разрядность АЦП - 105 МГц, 12 бит;
3. Тип, объем ПЛИС – Vertex E, до 2 миллионов логических вентиляей;
4. Тип, тактовая частота ЦСП – TMS 320C6202, 200 МГц;
5. Тип, объем памяти – четыре банка ZBT RAM 512Kx32, 128Мбит SDRAM;
6. Системная шина – CompactPCI 33/32, 33/64.

В состав модуля входят следующие основные элементы:

- ◆ четыре канала АЦП типа AD9432 фирмы Analog Device;
- ◆ ПЛИС типа Virtex (VirtexE) фирмы Xilinx;
- ◆ ЦСП типа 320C6203 фирмы Texas Instruments;
- ◆ динамическая память типа MT48LC фирмы Micron;
- ◆ статическая память типа MT55L2 фирмы Micron;
- ◆ энергонезависимая память типа AM29LV фирмы AMD.

Целью данной статьи является рассмотрение возможности построения на основе модуля XDSP-3MC цифрового анализатора, обрабатывающего радиосигналы на промежуточной частоте радиоприемного устройства средств радиомониторинга, с мгновенной полосой анализа до 50 МГц, а также исследование различных алгоритмов БПФ по времени их выполнения на основе ЦСП модуля.

Цифровой анализатор выполняет следующие основные процедуры: обнаружение сигнала на фоне шума, хранение предварительных и результирующих данных в буферной памяти, измерение частотно-временных параметров сигналов, вычисление амплитуды сигнала, передачу результирующих данных на следующие этапы обработки, прием/передачу команд управления.

Расчетные значения заполнения ПЛИС модуля (2 млн вентиляей) логическими блоками, реализующими обнаружение сигнала на потоке по одному каналу АЦП, а также системные интерфейсы, составили – 3858 slice (логическая ячейка), 19 BRAM (элемент блочной памяти), что от общего объема ресурсов ПЛИС составляет 20% объема логических вентиляей и 11% блочной памяти, из них:

- ◆ обнаружитель сигналов (2375 slice, 1 BRAM) 12% объема логических вентиляей и 1% блочной памяти;
- ◆ PCI контроллер (653 slice, 2 BRAM) 3% объема логических вентиляей и 1% блочной памяти;
- ◆ контроллер памяти (203 slice, 2 BRAM) 1% объема логических вентиляей и 1% блочной памяти;
- ◆ интерфейс с ЦСП, буферы промежуточных данных (627 slice, 14 BRAM) 4% объема логических вентиляей и 8% блочной памяти.

При оценке быстродействия ПЛИС модуля по операциям умножения с накоплением (MAC) независимых операндов с фиксированной запятой, получены следующие результаты: слова 8 бит – 65 млрд MAC, слова 12 бит – 25 млрд MAC, слова 16 бит – 14 млрд MAC, слово 32 бит – 1,7 млрд MAC. Это позволяет реализовывать, например, БПФ в скользящем окне длиной до 32 точек. Спектральный анализ большей выборки в реальном времени на ПЛИС такого объема реализовать нельзя. Использование ее в качестве сопроцессора для обчета длинных выборок (1024 и более точек) нецелесообразно, поскольку получаемое быстродействие сравнимо с полученным на процессоре, при значительно более сложной реализации.

На основе ЦСП целесообразно выполнять алгоритмически сложные процедуры измерения параметров сигналов и их статистической обработки. Одной из наиболее ресурсозатратных процедур являются спектральные преобразования. Для оценки быстродействия базовых процедур спектрального преобразования в статье рассмотрено выполнение следующих алгоритмов БПФ: Кули-Тьюки классический; Барруса-Парка классический по основанию 4; оптимизированный Барруса-Парка по основанию 2; опти-



мизированный Барруса-Парка по основанию 4; на основе действительных чисел; одновременного вычисления спектров двух сигналов одной процедурой БПФ; преобразования Хартли [1,4].

Указанные алгоритмы базируются на дискретном преобразовании Фурье и его свойствах.

Дискретное преобразование Фурье [1] определяется как

$$F(k) = \frac{1}{N} \sum_{i=0}^{N-1} f(i) e^{-\frac{2\pi k i j}{N}}, \quad \text{где } k=0, 1 \dots N-1; j^2 = -1 \quad (1)$$

Обратное преобразование Фурье:

$$f(k) = \sum_{i=0}^{N-1} F(i) e^{\frac{2\pi k i j}{N}}. \quad (2)$$

При реализации ДПФ количество комплексных умножений и сложений имеют порядок  $N^2$ , где  $N$  – размер входной выборки, например для  $N=1024$  требуется более  $10^6$  умножений.

Для сокращения количества выполняемых операций применяется алгоритм БПФ, основанный на свойствах коэффициентов при ряде Фурье, которые являются значениями тригонометрических функций (периодичность, четность, нечетность и т.д.). Преобразование массива данных в процедурах БПФ представляется ступенями преобразования, при этом входной массив данных каждой ступени служит входным массивом следующей ступени. Количество ступеней зависит от применяемого алгоритма БПФ, например для БПФ по основанию 4, количество ступеней равняется  $\log_4 N$ , для БПФ по основанию 2, количество ступеней равняется  $\log_2 N$ . На каждой ступени над элементами входного массива выполняется базовая операция БПФ – «бабочка», включая умножения и сложения с реализацией соответствующих операций для комплексных чисел.

Рассмотрим особенности каждого из представленных выше алгоритмов спектрального преобразования. При реализации спектрального преобразования по классическому алгоритму Кули-Тьюки размер входных данных выбирается из ряда  $2^N$ , при этом для выполнения алгоритма требуется порядка  $N \cdot \log_2 N$  комплексных умножений. Основная особенность данного алгоритма – возможность осуществлять преобразования с одним буфером памяти, что приводит к сокращению используемой буферной памяти.

При использовании классического алгоритма Барруса-Парка по основанию 4, размер входных данных выбирается из ряда  $4^N$ . В этом случае требуется на 25% комплексных умножений меньше, чем в алгоритме

БПФ по основанию 2. Для алгоритма этого типа необходимы два отдельных буфера – входной и выходной.

Оптимизированные версии алгоритмов БПФ Барруса и Парка по основанию 2 и по основанию 4 разработаны на основе классического алгоритма Барруса-Парка с использованием специальной организации процедуры вычисления, позволяющей минимизировать вычислительные затраты.

В случае, когда преобразуемый массив данных – последовательность действительных чисел, можно воспользоваться модификациями алгоритма БПФ. Особенностью одной из модификаций является проведение процедуры БПФ на количестве отсчетов в два раза меньшем количества отсчетов сигнала, что достигается записью в реальную часть четных отсчетов сигнала, а в мнимую – нечетных отсчетов сигнала. К достоинствам алгоритма относятся сокращение длины преобразования в два раза, что приводит к сокращению временных затрат и использованию реальных отсчетов входных данных. Недостатком алгоритма является необходимость введения дополнительной процедуры выделения спектра.

В данном алгоритме вспомогательные операции занимают около 40% от общего времени преобразования, но в связи с тем что, используется БПФ на количество точек в два раза меньшее, чем количество точек в выборке, повышение быстродействия составляет ~43%.

Другой модификацией, направленной на сокращение времени выполнения спектральных преобразований является алгоритм, реализующий возможность обработки двух сигналов одинаковой длительности одной процедурой БПФ. Для этого в реальную часть массива исходных данных записывается представление первого сигнала, в мнимую часть второго сигнала, после чего выполняется процедура БПФ и из полученного результирующего спектра путем преобразования выделяются спектры каждого из сигналов. Сигналы могут иметь различную длительность, в связи с чем длина преобразования будет равна длительности импульса с наибольшим значением, а часть массива, соответствующая меньшей длительности, дополняется нулями. Максимальный выигрыш от рассматриваемого алгоритма получается при равной длительности сигналов.

Общее время спектрального преобразования двух сигналов равно времени, затраченному на преобразование одного наибольшего по длительности сигнала, при этом потери времени происходят за счет выполнения операции по разделению спектров. В данном алгоритме вспомогательные операции занимают

14% от общего времени преобразования, повышение быстродействия составляет порядка 56%.

Дискретное преобразование Хартли (ДПХ) представляет собой сумму косинусного и синусного преобразований [4]:

$$H(k) = \frac{1}{N} \sum_{i=0}^{N-1} f(i) \text{cas} \left( \frac{2\pi k i j}{N} \right); \quad (3)$$

$$k=0, 1 \dots N-1,$$

$$\text{где cas}(t) = \cos(t) + \sin(t);$$

Обратное преобразование Хартли имеет вид:

$$f(k) = \sum_{i=0}^{N-1} H(i) \text{cas} \left( \frac{2\pi k i j}{N} \right); \quad (4)$$

Преобразования Хартли и Фурье связаны следующими выражениями:

$$R(k) = (H(k) + H(N-k))/2; \quad (5)$$

$$I(k) = -(H(k) - H(N-k))/2; \quad (6)$$

$$H(k) = R(k) - I(k) \quad (7)$$

ДПХ производится над действительными значениями, что приводит к меньшим вычислительным затратам по сравнению с ДПФ при их реализации на ЦСП с фиксированной точкой. Сокращения времени затрачиваемого на ДПХ достигается с помощью алгоритмов быстрого преобразования Хартли (БПХ), а также возможностью использования одного алгоритма, как для прямого, так и для обратного преобразования.

В таблице приведены результаты практических исследований различных алгоритмов БПФ на платформе ЦСП XDSP-3МC при разработке программного обеспечения в среде Code Composer.

### Выводы

1. Архитектурные решения заложенные в модуль XDSP-3МC: многоканальный высокоскоростной прецизионный тракт АЦП, специализированные распределенные вычислительные структуры (ПЛИС и ЦСП),

дополняющие друг друга по возможности обработки сигналов в реальном масштабе времени и реализации алгоритмов детальной обработки, элементы буферной памяти, системные стандартные интерфейсы обеспечивают выполнение базовых алгоритмов спектрального преобразования сигналов с требуемым быстродействием и качеством, что позволяет создавать на его основе специализированные цифровые анализаторы радиосигналов.

2. В результате проведенных исследований различных алгоритмов БПФ и разработки на их основе специального программного обеспечения на языке высокого уровня и ассемблере, удалось сократить время выполнения спектрального преобразования в 5 раз, (оптимизированный алгоритм Барруса-Парка по основанию 4 для действительных последовательностей с быстродействием ~53 мкс для 1024 точек).

3. Экспериментально определено, что для ЦСП модуля XDSP-3МC дальнейшая доработка рассмотренных алгоритмов может дать некоторое повышение быстродействия по отношению к достигнутому. Перспективным направлением повышения быстродействия алгоритмов спектрального преобразования является использование алгоритмов на основе БПХ, переход на новую элементную базу (TMS320C6416), применение новых методов построения вычислительных алгоритмов, (вычисления в полях Галуа, алгоритмов на основе факторизации матриц).

### Литература

1. *Оппенгейм А., Шафер Р.* Цифровая обработка сигналов. – М.: Связь, 1979.
2. Web-сайт ЗАО «СКАН инжиниринг телеком» [www.setdsp.ru](http://www.setdsp.ru).
3. *Мистюков В.Г.* Модуль цифровой обработки сигналов XDSP-3PC компании Scap Engineering Telecom. – Цифровая обработка сигналов, 2003, №3, с.37–43.
4. *Солонина А., Улахов Д., Яковлев Л.* Алгоритмы и процессоры цифровой обработки сигналов. Санкт-Петербург: BHV, 2000.

**Таблица 1. Быстродействия алгоритмов комплексных БПФ для данных на 1024 точки**

Тип реализации	Си-реализация	Ассемблер
Кули (По основанию 2)	600 мкс	106 мкс
Барруса-Парка (По основанию 4)	480 мкс	65 мкс
БПФ на основе алгоритма Барруса-Парка (По основанию 2)	200 мкс	76 мкс
БПФ на основе алгоритма Барруса-Парка (По основанию 4)	170 мкс	64 мкс
БПФ на БПХ (По основанию 2)	450 мкс	37 мкс (расчетное значение)



УДК 621.395

## Технология eXpressDSP проектирования и отладки систем цифровой обработки сигналов при работе с TMS320VC5416DSK. Интегрированная среда разработки Code Composer Studio v.2 для DSK

*А.А. Рыболовлев, В.Г. Трубицын*

### Введение

Потребительский спрос на продукцию компаний, производящих микропроцессорную элементную базу, не в последнюю очередь формируется уровнем совершенства предлагаемых технологий разработки систем цифровой обработки сигналов (ЦОС) и степенью их ориентированности на обеспечение производительного и комфортного труда проектировщиков. Данное соотношение объясняет тенденцию возрастания материальных и интеллектуальных затрат, вкладываемых в создание инфраструктуры конструкторской деятельности. Принципы построения аппаратных и программных средств разработки и отладки цифровых систем на цифровых процессорах обработки сигналов (ЦПОС) лидеров мирового рынка в данном секторе микроэлектроники показаны в [1]. Общая характеристика унифицированных средств проектирования и отладки систем ЦОС на основе сигнальных процессоров компании «Texas Instruments» (TI) изложена в [2]. В [3] представлено описание интегрированной среды Code Composer Studio (CCS), являющейся основным программным инструментарием отладки цифровых систем на процессорах TI. Предлагаемая статья продолжает цикл материалов [4, 5], посвященных применению стартового набора TMS320VC5416 DSK (далее – DSK) в подготовке специалистов в области ЦОС. Статья содержит характеристику eXpressDSP-технологии проектирования и отладки систем ЦОС на основе процессоров TI и среды CCS с акцентом на их особенности для работы с DSK.

*Статья продолжает публикацию цикла работ, посвященных применению стартового набора TMS320VC5416 DSK в подготовке специалистов в области ЦОС. Содержит краткое описание технологии eXpressDSP проектирования и отладки систем ЦОС и интегрированной среды CCS с акцентом на их особенности при работе со стартовым набором DSK.*

В тексте приведены Интернет-адреса страниц сайта компании «Texas Instruments», содержащих более полную информацию об элементах eXpressDSP.

### **Назначение и основные элементы технологии eXpressDSP**

Технология eXpressDSP ([www.ti.com/software](http://www.ti.com/software)), созданная компанией «Texas Instruments», представляет собой технологию проектирования и отладки систем цифровой обработки сигналов на основе процессоров TMS320xxxx. Организационные, программные и аппаратные решения, на которых базируется eXpressDSP, обеспечивают разработчику максимум удобств в процессе создания готового продукта. Знание и полное использование разработчиком системы ЦОС возможностей eXpressDSP является необходимым условием повышения эффективности конструкторской деятельности. Основные элементы технологии показаны на *рис. 1*.

Интегрированная среда разработки Code Composer Studio ([www.ti.com/ccstudio](http://www.ti.com/ccstudio)) является ключевым, максимально приближенным к разработчику компонентом eXpressDSP, структура и особенности ее сокращенной версии v.2, адаптированной к работе со стартовым набором DSK, изложены ниже.



Рис. 1. Основные элементы технологии eXpressDSP

Программное обеспечение eXpressDSP формируется тремя составными частями.

Масштабируемое ядро реального времени DSP/BIOS разработано «Texas Instruments» для процессоров семейства TMS320C28, платформ TMS320C5000 и TMS320C6000. Полная информация о DSP/BIOS размещена по Интернет-адресу [www.ti.com/dspbios](http://www.ti.com/dspbios). Ядро позволяет вести разработку цифровых систем в условиях взаимодействия хост-компьютера и целевого процессора (например, процессора стартового набора DSK) в режиме реального времени с обеспечением синхронизации и возможностью конфигурирования системы. Особенности программных решений, использованных при построении ядра, и интеграция в среду CCS специального BIOS/DSP инструментария (средств анализа BIOS/DSP Analysis Tools) позволяют реализовать в практике разработки систем ЦОС ряд возможностей, принципиально важных для проектировщика. К последним относятся:

- ◆ быстрота разработки многофункциональных систем и переноса готовых приложений на новые аппаратные средства;
- ◆ реализация режима категоризованной многозадачности;
- ◆ разработка систем ЦОС в условиях неполных знаний особенностей используемых аппаратных средств;
- ◆ контроль функционирования системы в режиме реального времени с одновременным отображе-

нием объектов ядра и загрузки целевого процессора, трассировкой и определением максимального и среднего времени исполнения задачи;

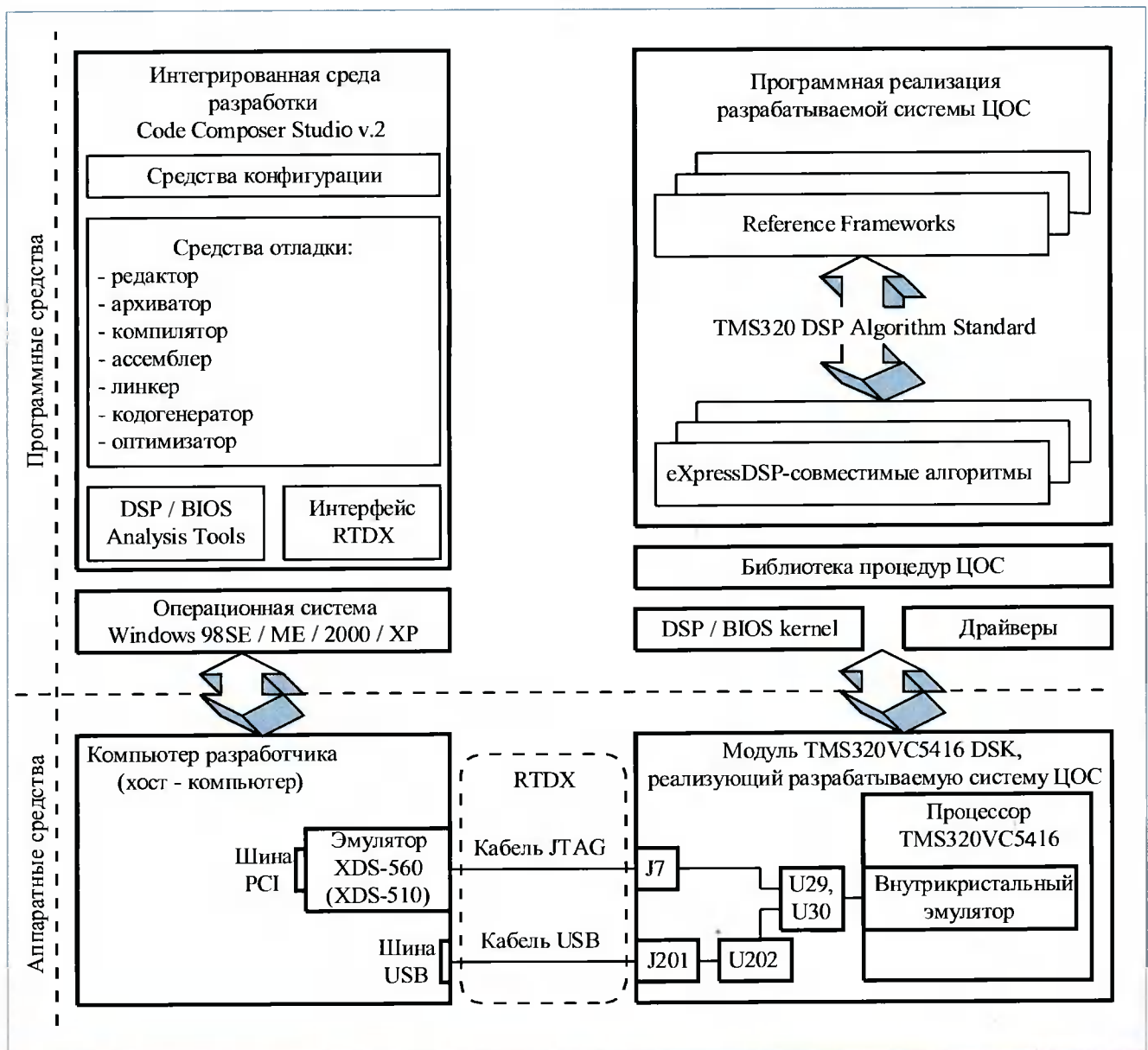
- ◆ упрощение процедур программирования и конфигурирования отдельных устройств и системы в целом и т. д.

Стандарт разработки программных продуктов TMS320 DSP Algorithm Standard детализирует требования к работе программистов в процессе разработки цифровых систем на процессорах TI платформ TMS320C2000, TMS320C5000 и TMS320C6000. Стандарт ([www.ti.com/algostanddevkit](http://www.ti.com/algostanddevkit)) представляет собой открытый для совершенствования комплекс требований, обязательных для обеспечения eXpressDSP-совместимости разрабатываемых алгоритмов, и дополнительных рекомендаций, необходимых для унификации программных реализаций конструируемых систем ЦОС. Особое внимание стандарт уделяет формированию всех видов программных интерфейсов (application programming interface, API). Четырехуровневая структура TMS320 DSP Algorithm Standard показана в *Таблице 1*.

Достаточно удобным элементом eXpressDSP представляется набор программных элементов Reference Frameworks ([www.ti.com/rfinfo](http://www.ti.com/rfinfo)), содержащий широкий перечень готовых кодовых конструкций, реализующих частные процедуры, наиболее часто используемые при разработке систем ЦОС различного назначения. Применение Reference Frameworks позволяет значительно сократить время разработки конкретного приложения.

**Таблица 1. Структура TMS320 DSP Algorithm Standart**

Уровень	Содержание требований и рекомендаций
Первый уровень	Общие принципы создания программных продуктов (порядок использования языка Си, порядок формирования и использования сегментов памяти данных и памяти программ, порядок обращения к периферийным устройствам и т. д.)
Второй уровень	Порядок структурирования алгоритмов. Обобщенная модель алгоритма
Третий уровень	Особенности использования процессоров различных семейств (порядок использования прерываний, внутрикристалльной памяти, регистров процессоров и т. д.)
Четвертый уровень	Особенности разработки систем ЦОС различного назначения (телефония, обработка изображений, управление двигателями, шифрование, модуляция и т. д.)



**Рис. 2. Структура взаимодействия основных элементов eXpressDSP в процессе разработки системы ЦОС на модуле стартового набора TMS320VC5416 DSK**

Важную роль в технологическом процессе разработки систем ЦОС играет предоставляемая конструктору возможность использования готовых eXpressDSP-совместимых алгоритмов ([www.ti.com/thirdpartycatalog](http://www.ti.com/thirdpartycatalog)), созданных компаниями-партнерами «Texas Instruments» ([www.ti.com/3p](http://www.ti.com/3p)) в полном соответствии с требованиями стандарта TMS320 DSP Algorithm Standard.

Системное представление разработчиком порядка взаимодействия основных элементов eXpressDSP в процессе разработки системы ЦОС является обязательным условием реализации преимуществ, предоставляемых технологией. Структура такого взаимодействия при работе с модулем DSK, показана на *рис. 2*. При этом отображена возможность двух режимов эмуляции с обеспечением обмена данными в режиме реального времени RTDX (Real – Time Data Exchange).

### **Интегрированная среда разработки Code Composer Studio v.2 для DSK**

CCS представляет собой отладочную среду разработчика, объединяющую в себе необходимые средства для проведения полного цикла разработки: от конфигурирования системы ЦОС, написания и компиляции программы до отладки и анализа поведения алгоритма. Первым шагом работы с CCS является задание конфигурации аппаратной части отлаживаемой системы ЦОС.

#### **Конфигурирование устройств ЦОС**

Для разработки приложений ЦОС необходимо определить аппаратную платформу компании «Texas Instruments». CCS позволяет выбирать платы «Texas Instruments» и симуляторы, используя Code Composer Studio Setup. CCS Setup автоматически запускается в тех случаях, когда CCS используется без определения конкретной платформы. Приложение CCS Setup может быть запущено из меню File главного меню CCS путем выбора опции Launch Setup, диалоговое окно которого представлено на *рис. 3*. В результате запуска CCS Setup появляется диалоговое окно Import Configuration (импорт конфигурации). Центральная часть окна содержит список доступных плат и симуляторов. Любые из них используются при запуске приложений ЦОС. Для подключения платы к конфигурируемой системе ЦОС необходимо перенести изображение платы из центральной части окна в окно System Configuration (конфигурация системы). При установке платы необходимо установить драй-

вер устройства. Это осуществляется использованием опции Install a Device Driver (установить драйвер устройства) правой части диалогового окна Code Composer Studio Setup. В появившемся окне диалога следует выбрать предлагаемый драйвер TI-платы, расположенный внутри поддиректории drivers, где установлена программа CCS. Эти файлы имеют расширение .dvr. После установки драйвера в центральной части диалогового окна Code Composer Studio Setup отображается выбранная плата. При выборе драйвера в правой части диалогового окна CCS Setup появляется информация о поддерживаемом эмуляторе, типе поддерживаемого процессора и поддерживаемом наборе функциональных возможностей. Если в состав JTAG-пути входит неподдерживаемое устройство, для него возможно задание режима пропуска (BYPASS).

### **Создание нового проекта, просмотр кода программы, редактирование, компиляция и запуск программы**

После конфигурирования системы для достижения поставленных перед разработчиком целей по формированию цифрового устройства, необходимо воспользоваться средствами разработки программ с помощью CCS для проектирования приложений ЦОС. Для написания и компиляции программы используются средства менеджмента проектов, оптимизированные компиляторы, встроенный редактор интегрированной среды программирования CCS.

В меню Project создается проект с указанием имени, пути к создаваемой рабочей папке, а также выбранного устройства, которое будет конфигурироваться с помощью CCS. Информация для каждого проекта записана в отдельном проектном файле с расширением .pj1, в котором хранятся настройки проекта и ссылки на различные файлы, используемые создаваемым проектом. К создаваемому проекту с помощью опции Add Files to Project добавляются файлы, различные по типу (расширению). Файл проекта используется для построения приложения и может включать :

- ◆ файлы с расширением .c – исходные файлы, которые могут быть скомпилированы или связаны;
- ◆ файлы **vectors.asm** и **load.asm** содержат инструкции на ассемблере, необходимые для установки прерываний во внутренних точках программы, написанной на языке Си (для сложных программ можно использовать DSP/BIOS для определения всех векторов прерываний автоматически);

- ◆ файлы с расширением **.o** или **.lib** – объектные или библиотечные файлы (они могут быть только связанными; в папке библиотек компилятора выбирается файл с расширением **.lib** для конфигурируемого устройства; библиотечные файлы обеспечивают поддержку устройств ЦОС в режиме реального времени);
- ◆ файл с расширением **.cmd** – командный файл компоновщика, содержащий карту памяти;
- ◆ файл с расширением **.h** – включенный файл, автоматически добавляемый к проектному списку;
- ◆ файл с расширением **.paf** – информационный файл для каждого проектного файла (информация в этом файле используется для определения файлов, которые должны быть восстановлены);
- ◆ файл с расширением **.out** – символьный файл, созданный при построении программы, получивший название COFF-файл (он должен быть обязательно загружен в реальную плату или модель до запуска; программный (объектный) код и данные загружаются в место назначения по адресам, определенным в COFF-файле).

Создание нового исходного файла осуществляется при помощи встроенного редактора активизации окна, выбор которого осуществляется выбором опции Source File меню New главного меню File. По умолчанию, встроенный редактор помечает новые файлы как Untitled N, где N – уникальный номер (например, Untitled 1, Untitled 2, и т. д.). Чтобы не испытывать неудобства во время открытия уже существующих файлов, не следует сохранять файлы с именами, начинающимися с префикса Untitled. При формировании исходного файла необходимо указывать тип файла, по умолчанию встроенным редактором выбрано «C Source Files (\*.c)». Существует

возможность открытия в CCS приложений Microsoft Word и Microsoft Excel. Такая процедура выполняется из окна ActiveX, при этом имеется возможность создания нового документа или открытия уже существующего.

Для компиляции и запуска программы необходимо выбрать Rebuild All из опции Project главного меню CCS. Среда CCS перекомпилирует, переассемблирует, перекомпоновывает все файлы проекта. В результате этих действий формируется файл типа .out, который записывается в директории отладки, расположенной в папке текущего проекта. После формирования символьного файла CCS загружает программу в TI-плату, либо в имитирующее устройство. Отладчик автоматически открывает окно дизассемблера (Disassembly window), которое показывает дизассемблируемые команды, выполняемые программой. Кроме того, окно дизассемблера показывает демонтированные инструкции и символьную информацию, необходимую для отладки. Дизассемблирование полностью изменяет процесс обработки и позволяет показать содержание памяти в машинном коде. Символьная информация состоит из символов и линий алфавитно-цифровых характеристик, которые представляют адреса и величины. Для каждой инструкции языка ассемблера дизассемблерное окно показывает демонтированную инструкцию и адрес, в котором расположена инструкция. Для внесения изменений в список дизассемблирования, отладчик читает код, демонтирует его и добавляет символьную информацию, начинающуюся в местоположении, обозначенном программным счетчиком. Текущее местоположение программного счетчика в окне обозначает желтая стрелка. Выполнение программы начинается с этого места.



Рис. 3. Окно приложения Code Composer Studio Setup

В меню View CCS выбор опции Mixed Source/ASM позволяет просматривать исходный код на языке Си и результирующий код на ассемблере одновременно.

При формировании программы часто приходится изменять опции компилятора и исправлять синтаксические ошибки, произошедшие в результате компиляции. Ошибки отображаются в нижней части экрана в отчете компиляции. Для установления опций компилятора (Compiler), компоновщика (Linker) и архиватора (Link Order) необходимо определение набора проектных опций, которые будут применяться ко всем файлам проекта. Установка проектных опций осуществляется в диалоговом окне Build Options выбором соответствующей вкладки. Результаты создания проекта, его редактирования, компиляции, компоновки и запуска программы представлены на рис. 4.

### Отладка программы

В состав CCS входит встроенный отладчик, интерфейс которого, как и у редактора, аналогичен MC Visual C, но имеет целый ряд ЦОС-приложений. Окно

отладчика выполняет функции окна редактирования, в котором можно производить отладочные действия. Для проведения отладки поведения системы используются различные инструменты отладки, к которым можно отнести: точки останова, точки исследования, различные окна просмотра и файловый ввод/вывод. Встроенный отладчик CCS имеет три типа точек останова, вернее, точек контроля и воздействия на выполнение программы:

- ◆ точки останова (Break Points);
- ◆ точки исследования (Probe Points);
- ◆ профильные точки (Profile Points).

Точки останова служат для остановки программы в требуемом месте. В CCS различаются два типа точек останова: программные – по достижению определенного оператора при выполнении программы и аппаратные – по выполнению определенной операции на шине, что позволяет отслеживать последовательность выполнения программы, быстро локализовать неприятные ситуации (такие, как порча по непонятным причинам данных в памяти или выход индекса массива за его пределы). При тестировании программы часто необходимо проверять значения переменных в ходе выполнения программы. Для этого

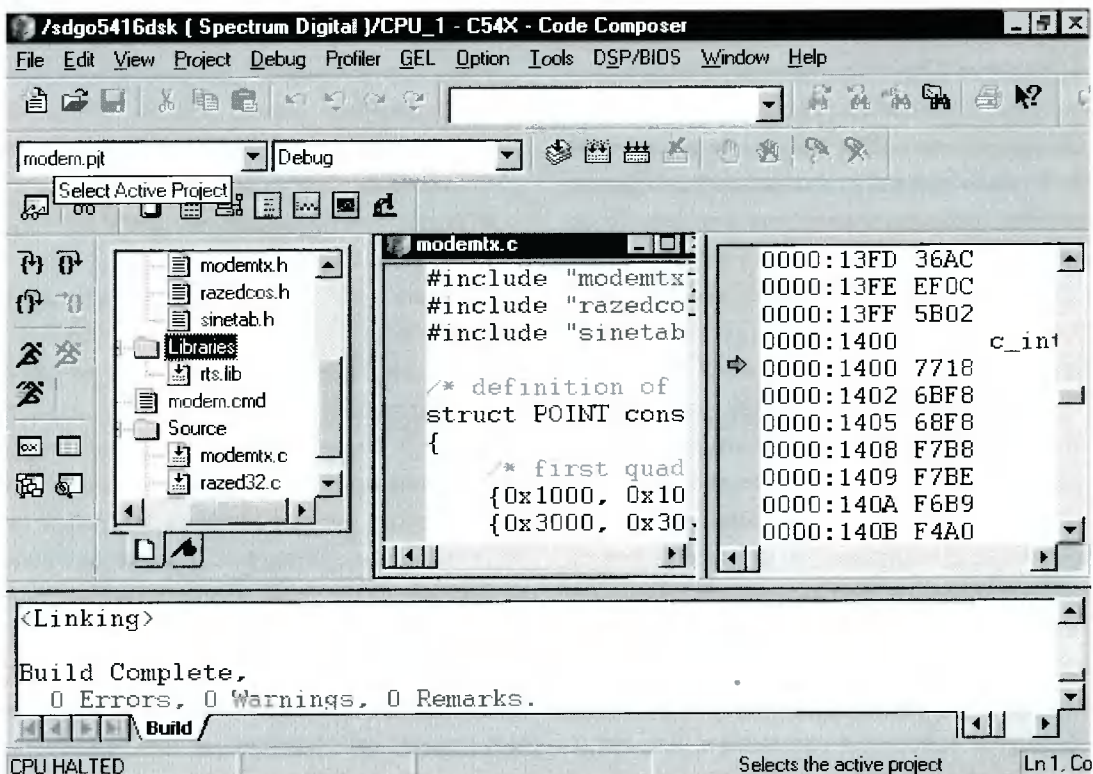


Рис. 4. Главное диалоговое окно Code Composer Studio



необходимо использовать точки останова и окна слежения для просмотра значений каждой переменной. В окне Project View необходимо выбрать исходный файл, определить строку в теле программы для установки Break Points и выбрать на панели инструментов знак Toggle Break Point. В выбранном поле появится красная иконка, сигнализирующая об установке точки останова. Окна отслеживания переменных для выбранных точек останова вызываются из меню View опцией Watch Window. По умолчанию выбрана вкладка Watch Locals, в которой отображаются локальные переменные выполняемой функции. В дополнение к отслеживанию простых переменных существует возможность отслеживать элементы структуры. С этой целью во вкладке Watch 1 окна отслеживания в колонке Name необходимо ввести имя отслеживаемой переменной. Выбранная структура определяется глобально и инициализируется в исходном файле. Тип структуры описан в файле с расширением *.h*.

Следующим типом точек контроля являются точки исследования (Probe Points), которые позволяют читать данные из файла. Точка исследования – это специальная возможность CCS, ориентированная на отладку приложений ЦОС. Фактически – это программный аналог щупа осциллографа или логического анализатора, подключаемого к схеме. Точка исследования представляет собой подключаемый к определенной точке программы канал передачи или снятия данных. Она используется при передаче входных данных из файла в буфер имитируемого устройства, а также при передаче данных из буфера имитируемого устройства в файл для анализа. Точки исследования позволяют автоматически вводить и выводить данные в файлы. С этой целью в окне Project View выбирается исходный файл, определяется строка в теле программы для установки Probe Points и активизируется на панели инструментов знак Toggle Probe Point, после чего в окне исходного кода появляется синий значок, сигнализирующий о том, что точка исследования установлена. Обмен данными в этих точках происходит прозрачно для исполняемой программы при помощи внутрисхемной эмуляции средств отладчика.

Для определения файлов с входными и выходными данными необходим выбор опции File I/O меню File CCS. Отладчик позволяет направлять или пересылать данные между файлом на компьютере и реальной (имитационной) платой. Это наилучший способ воспроизводить программу, зная проверочные значения. Средства ввода/вывода используют концепцию проверки в точках исследования (Probe Po-

int), которая позволяет вставлять/доставать примеры или отражать данные памяти в используемых точках исследования. При достижении выполняемой программой точки исследования, присоединенный к ней объект (файл, граф, или окно памяти) обновляется. Как только объект обновится, выполнение программы продолжится.

Точки исследования подобны точкам останова в том, что обе останавливают выполнение программы. Однако эти точки имеют следующие различия:

- ◆ точки исследования останавливают выполнение программы на мгновение для представления результатов, а затем выполнение программы возобновляется;
- ◆ точки останова останавливают выполнение программы до тех пор, пока выполнение не будет возобновлено вручную;
- ◆ точки останова обновляют все открытые окна;
- ◆ точки исследования позволяют осуществить автоматический ввод/вывод файла, в то время как точки останова такую возможность не предоставляют.

Символьное окно просмотра CCS используется для исследования создаваемого проекта. С помощью символьного окна просмотра можно исследовать проект, выбирая следующие табуляции:

- ◆ Files – анализ всех исходных файлов, содержащихся в пределах проекта.
- ◆ Functions – выбор функций в пределах выбранного файла.
- ◆ Globals – выбор переменных, определяемых как глобальные.
- ◆ Types – использование типов данных, определенных пользователем.
- ◆ Labels – обозначения ассемблера и директив.

Символьное окно просмотра становится активным при выборе опции Symbol Browser меню Tools CCS. При отладке программы часто полезно отображение изменения величин в ходе выполнения программы. Кроме символьного окна просмотра CCS позволяет использовать смотровые окна. Открытие смотрового окна осуществляется выбором опции Watch Window меню View CCS. Окно Watch содержит два раздела: локальное окно просмотра (Watch Locals) и общее окно просмотра (Watch). В окне Watch Locals отладчик автоматически отображает название, величину и тип локальных переменных для выполняемых в настоящее время функций. В окне Watch отладчик показывает название, величину и тип локальных и глобальных переменных и выражений, которые были определены. Поле Radix позволяет изменить формат представления переменной.

## Анализ

Средства анализа разработанных приложений ЦОС включают инструменты визуализации данных и средства анализа в реальном времени. Визуализация данных представляет собой самый эффективный метод анализа приложений ЦОС. Разнообразие графических форматов представления данных заставляет использовать большое число алгоритмов. Разработке кода для приложений ЦОС помогают различные схемы визуализации данных. Преобразование Фурье позволяет определять формат и оценить точность операции цифровой фильтрации. Сложные математические формулы, формирующие основу спектральных графиков, скрыты для пользователя. Однако, рассматривая полученный сигнал, используя возможности визуализации

данных и сравнивая его с ожидаемым результатом, можно быстро оценить алгоритм цифровой фильтрации. Для облегчения визуализации данных необходимо использование точек исследования (Probe Point) и файлового ввода/вывода (File I/O).

Процесс визуализации данных IDE CCS располагает четырьмя типами графических изображений. К ним относятся: амплитудно-временная диаграмма (Single Time), диаграмма быстрого преобразования Фурье (FFT), диаграмма просмотра (Eye Diagram) и диаграмма совокупности (Constellation). Для создания диаграмм используется опция Graph-Time/Frequency меню View. Для каждого типа диаграмм вводятся параметры свойств выбранного представления. Использование опции Time/Frequency позволяет анализировать сигнал во временной и частотной областях. Для анализа в частотной области

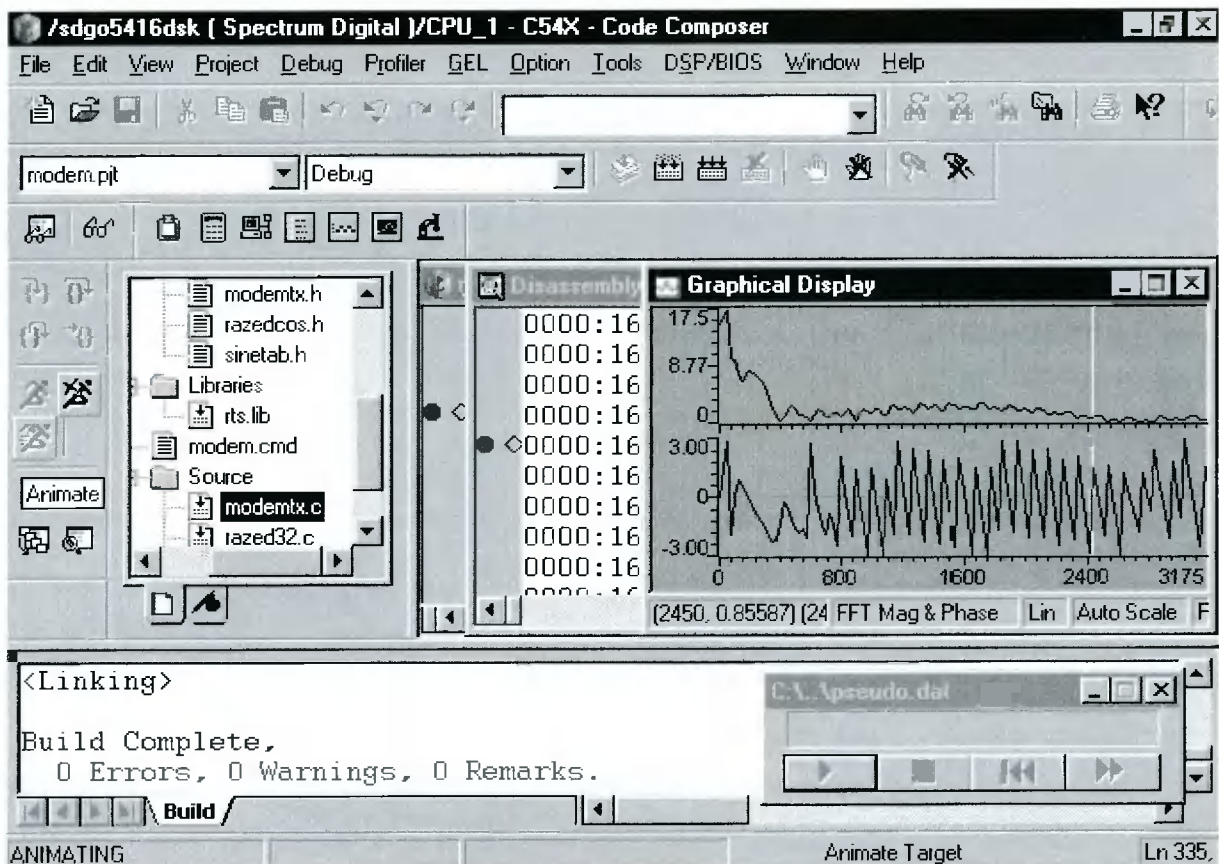


Рис. 5. Визуализация данных с использованием точек контроля и файлового ввода/вывода





диаграмма FFT включает: FFT Waterfall, Complex FFT, FFT Magnitude and Phase.

Таким образом, IDE CCS позволяет редактировать исходные тексты на Си и ассемблере. Используя проект, среда CCS может создавать полностью или частично проработанный вариант и компилировать отдельные файлы. Средства проектирования CCS могут быть использованы в качестве альтернативы традиционным средствам, однако при необходимости CCS позволяет использовать и традиционные средства проектирования.

### **Заключение**

Представленная в статье технология eXpressDSP является примером завершенной, но открытой к совершенствованию совокупности процедур и средств, нацеленных на обеспечение доступности быстрого и достаточно малозатратного решения конструкторских задач в высокотехнологичной области создания систем цифровой обработки сигналов. Авторы статьи выражают признательность Грибачеву С. А. и Чернову В. Е. (компания «Сканти – Рус») за предоставление материалов, использованных в работе.

### **Литература**

1. Солонина А., Улахович Д., Яковлев Л. Алгоритмы и процессоры цифровой обработки сигналов. С.-Пб.: БХВ-Петербург, 2001. 464 с.
2. Чернов В. Е., Грибачев С. А. Концепция цифровых сигнальных процессоров трех платформ компании «Texas Instruments. Унифицированные средства проектирования и отладки // Цифровая Обработка Сигналов. 2001. № 2. С. 41 – 44.
3. Воробьев В. И., Домрачев В. В., Бойкова О. В. Интегрированная среда Code Composer Studio. С.-Пб.: Военный университет связи, 2003. 110 с.
4. Рыболовлев А. А., Трубицын В. Г. Стартовый набор TMS320VC5416 DSK. Состав, основные характеристики и возможности. Организация памяти // Цифровая Обработка Сигналов. 2003. № 2. С. 44 – 49.
5. Рыболовлев А. А., Трубицын В. Г. Логическое устройство стартового набора TMS320VC5416 DSK. Использование элементов памяти стартового набора и внешних элементов памяти для расширения общего пространства памяти // Цифровая Обработка Сигналов. 2003. № 3. С. 46 – 52.

## **Уважаемые авторы!**

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

### **Требования к текстовым материалам и сопроводительным документам:**

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

### **Требования к иллюстрациям:**

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

**Справки по телефонам: (095) 290-9088, 290-9085  
или по электронной почте [info@dspa.ru](mailto:info@dspa.ru)**

УДК 681.323

## Семейство модулей цифровой обработки сигналов на основе процессоров TMS320C64xx и ПЛИС XILINX

Н.И. Гутовец

При разработке нового семейства модулей цифровой обработки сигналов на основе процессоров **TMS320C6414/15/16** фирмы **Texas Instruments** и ПЛИС

фирмы **Xilinx**, ставилась задача обеспечить предельно высокую производительность плат этого семейства, а также иметь возможность получить относительно менее производительные, но существенно более дешевые изделия. Решение этой во многом противоречивой задачи обеспечивалось с одной стороны, модульным подходом при разработке рассматриваемого семейства, обеспечившим возможность оптимизации стоимости конкретного экземпляра любого модуля за счет выбора элементной базы (оптимальной для решения именно данной задачи). С другой стороны, при разработке наиболее дешевого процессорного модуля (**ADP64Z1PCI**) количество компонентов в системе уменьшено всего за счет использования встроенного в процессор ЦОС интерфейса PCI. Кроме того, при разработке семейства учитывалась и возможность, а во многих случаях и необходимость использования ПЛИС для обработки сигналов.

Ввод аналоговых сигналов и их преобразование в цифровой вид в рассматриваемом семействе, как и в других изделиях фирмы «Инструментальные Системы», обеспечивается установкой на любой из несущих модулей ЦОС унифицированных submodule аналого-цифрового преобразования или цифрового радиоприема. Кроме упомянутых типов сменных submodule, возможна установка на несущие платы submodule цифроаналогового преобразования, а также submodule телефонного интерфейса. В на-

*Представлено краткое описание состава и основных технических характеристик семейства модулей ЦОС ADP64Zexx и ADPZxPCI фирмы «Инструментальные системы». Семейство модулей реализовано на основе сигнальных процессоров TMS320C64xx фирмы Texas Instruments и ПЛИС фирмы Xilinx и предназначено для эффективного решения широкого спектра приложений, от связанных задач (ADP64Z1PCI) до создания высокопроизводительных систем обработки радиолокационных сигналов и изображений.*

стоящее время фирмой «Инструментальные Системы» наработана линейка из более чем 30 сменных submodule, устанавливаемых как на беспроцессорные несущие платы, так и на несущие платы, выполненные на различных типах сигнальных процессоров и ПЛИС. Рассмотренные подходы позволяют быстро подобрать необходимую для решения конкретной задачи платформу и использовать наиболее выгодные пути ее решения, как с точки зрения производительности, так и с точки зрения стоимости системы.

В настоящее время для цифровой обработки сигналов, кроме сигнальных процессоров, часто используются программируемые логические микросхемы – ПЛИС. Во многих случаях их применение позволяет решить задачу, которую невозможно или весьма сложно решить с помощью сигнальных процессоров. Так, мы столкнулись с необходимостью обеспечить фильтрацию сигнала с помощью медианного фильтра размерностью 30-100 отсчетов (в данном случае медиана – это средняя точка рассортированной по возрастанию или убыванию выборки входных данных). Решение этой задачи сигнальным процессором любого типа или группой таких процессоров, потребовало бы недопустимо высоких временных затрат. В то же время медианный фильтр на 60 отсчетов был реализован в небольшой, по современным меркам, ПЛИС XCV600E. Тем не менее рассматривать ПЛИС как «панацею», пригодную для решения любых задач,

я бы не стал. Некоторые приложения вообще не пригодны для реализаций в ПЛИС, многие алгоритмы не поддаются распараллеливанию или содержат комплекс функций, не являющихся функциями цифровой обработки сигналов, другие, – вообще не используют функции умножения с накоплением – MAC. Кроме того, необходимо учитывать тот факт, что процессоры ЦОС лучше обеспечены программами разработки прикладного ПО, особенно большой сложности, имеют меньшее время разработки сравнимых по сложности задач, да и существенно больший круг разработчиков освоили работу с этими процессорами.

По мнению многих специалистов, как отечественных, так и зарубежных (в частности, специалистов известной фирмы BDTI), в настоящее время для решения задач цифровой обработки сигналов, наиболее эффективно использование гетерогенной комбинации вычислительных узлов – процессоров цифровой обработки сигналов и ПЛИС одновременно.

Именно этот подход реализован в семействе модулей обработки сигналов на основе процессоров **TMS320C64xx** фирмы **Texas Instruments** и ПЛИС фирмы **Xilinx**. В настоящее это семейство состоит из 3-х модулей, предназначенных для решения различного круга задач:

- **модуль ЦОС ADP64Z3E**
- **модуль ЦОС ADP64Z2PCI**
- **модуль ЦОС ADP64Z1PCI**

Эти модули позиционированы для решения нескольких различных классов задач. Так, модули ЦОС **ADP64Z3E** и **ADP64Z2PCI** предназначены для си-

стем, требующих предельно высокой вычислительной мощности – это радиолокационные станции систем управления воздушным движением, профессиональные радиоприемники, системы обработки и анализа изображений, медицинские системы и другие системы с аналогичными требованиями по быстродействию. Модуль **ADP64Z3E** используется автономно, а также в применениях, где достаточно связи с ведущей ПЭВМ по каналам IEEE1394 или USB. Элементная база этого модуля выбиралась с учетом возможности работы в расширенном температурном диапазоне. Модуль ЦОС **ADP64Z2PCI** предназначен для решения тех же задач, но в составе ПЭВМ с шиной PCI, в том числе в промышленных ПЭВМ. Самый дешевый модуль ЦОС **ADP64Z1PCI** целесообразно использовать в связанных задачах, в частности при работе с протоколами **E1/T1** или при аудиообработке и обработке речи. Тем не менее вышесказанное совершенно не означает, что эти модули можно использовать только в указанных применениях – унифицированный интерфейс сменных submodule позволяет установить на любой модуль ЦОС любой аналоговый submodule и, соответственно, использовать их для решения самых разных задач.

На каждом из модулей установлен процессор ЦОС **TMS320C6415** или **TMS320C6416** с тактовой частотой 600 или 720 МГц, а также ПЛИС, обеспечивающие как ввод/вывод данных, так и решение задач цифровой обработки сигналов. Некоторые характеристики этих модулей приведены в *таблицах 1 и 2*.

**Таблица 1. Основные ресурсы модулей обработки сигналов ADP64Zxx**

Процессорный модуль	Интерфейс	Реализация интерфейса	SDRAM A, MB	SDRAM B, MB	ASRAM B, MB	FLASH, MB	ПЛИС ADM	ПЛИС ЦОС
ADP64Z3E	ПЭВМ IEEE1394 RS232, RS485, USB1, USB2	ПЭВМ Контроллер ввода/ вывода	256	64	2	16	XC2V2000 до XC2V8000	Обработка сигналов реализовывается в ПЛИС ADM
ADP64Z2PCI	PCI2.2 32 Bit, 66 MHz	Внешний PCI контроллер	512	32	Нет	16	XC2V500 до XC2V1000	XC2V2000 до XC2V8000
ADP64Z1PCI	PCI 2.2 32 Bit, 33 MHz	Встроенный PCI контроллер	128	32	2	128/ 512KB	ACEX1K	XC2S200

Таблица 2. Скоростные характеристики ввода/вывода модулей обработки сигналов ADP64Zxx

Процессорный модуль	Скорость обменов интерфейса ПЭВМ	Тактовая частота шин процессора ЦОС, МГц	Скорость обмена данными с submodule ADMxx, LVTTTL MB/C	Скорость обмена данными с submodule ADMxx, LVDS MB/C	EMIF A MB/c	EMIF B MB/c
ADP64Z3EI	До 400 Мбит/с	120	200	800	960	240
ADP64Z2PCI	До 250 МБ/с	120	200	800	960	240
ADP64Z1PCI	50 МБ/с	133	200	нет	1000	250

### Базовая структура модулей обработки сигналов ADP64Exx

На рис. 1 представлена базовая структура каналов передачи данных рассматриваемого семейства модулей. Как видно из рисунка, при работе с submodule аналого-цифрового преобразования (АЦП) или DDC (digital down converter) входной поток данных поступает в ПЛИС ADM, где или подвергается предварительной обработке, или через реализованный в ПЛИС узел FIFO передается на обработку в сигнальный процессор или же передается в ведущую ПЭВМ. Основные обрабатывающие узлы модуля – процессор ЦОС и ПЛИС, симметрично охвачены двумя процессорными шинами – шиной EMIF A шириной 64 бита и шиной EMIF B шириной 16 бит. Обе шины работают на тактовой частоте 120 МГц и обеспечивают, таким образом пропускную способность **960 МБ/сек** на **EMIF A** и **240 МБ/сек** на **EMIF B**. Для накопления больших объемов данных, предусмотрено использование режима «Peripheral Data Transfer» (PDT), реализованного в семействе сигнальных процессоров TMS320C64xx. Этот режим позволяет передавать данные из устройства ввода непосредственно в **SDRAM** или в обратном направлении.

Основным каналом передачи данных в процессор ЦОС является шина EMIF A как обеспечивающая максимальную пропускную способность. При необходимости для обмена данными процессора ЦОС и submodule можно использовать и шину EMIF B. В стандартных реализациях EMIF B используется прежде всего для управления узлами аналогового интерфейса и вывода данных в ЦАПы и для обмена данными с ведущей ПЭВМ. Данные в ЦАПах семейства модулей **ADP64Zxx** выводятся также через реализованные в ПЛИС FIFO. Предельная скорость шины EMIF B составляет 240 – 260 МБ/сек, что соответствует частоте дискретизации около **60 МГц** в каждом канале ЦАП.

Более высокие скорости вывода данных в ЦАПах, если они требуются, могут быть достигнуты путем изменения прошивки ПЛИС и подключения FIFO каналов ЦАП к шине EMIF A.

Наличие связей ПЛИС с сигнальным процессором по обеим его шинам позволяет во многих случаях избавить разработчика прикладной системы от необходимости организовывать несколько потоков данных на одной шине. Самым очевидным применением такой архитектуры является ввод данных в процессор по одной шине, а вывод обработанных в процессоре данных в ведущую ПЭВМ – по другой. Возможна и итерационная обработка данных совместно в ПЛИС и процессоре ЦОС. Такие нетривиальные задачи иногда встречаются. Конечным итогом описанной архитектуры является повышение общей производительности системы за счет разделения потоков ввода/вывода и потоков работы с памятью, например, при накоплении большого массива входных отсчетов или при интенсивных обменах с памятью в процессе вычислений.

Шина интерфейса **HPI** обеспечивает ведущей ПЭВМ доступ ко всем адресуемым ресурсам процессора ЦОС.

### Модуль обработки сигналов ADP64Z3E

Рассмотренная выше базовая структурная схема практически полностью реализована в модуле **ADP64Z3E** (рис.2). Этот процессорный модуль выполнен в формате 6U и предназначен для автономных применений. Внешние связи модуля обеспечиваются интерфейсами **IEEE1394**, **RS232**, **RS485**, **USB 1.1** и **USB2**, а также каналом дискретного ввода/вывода (до 240 линий), подключенным к ПЛИС аналогового интерфейса. Канал дискретного ввода/вывода предназначен для организации обменов с нестандартными внешними устройствами (например, специализирован-

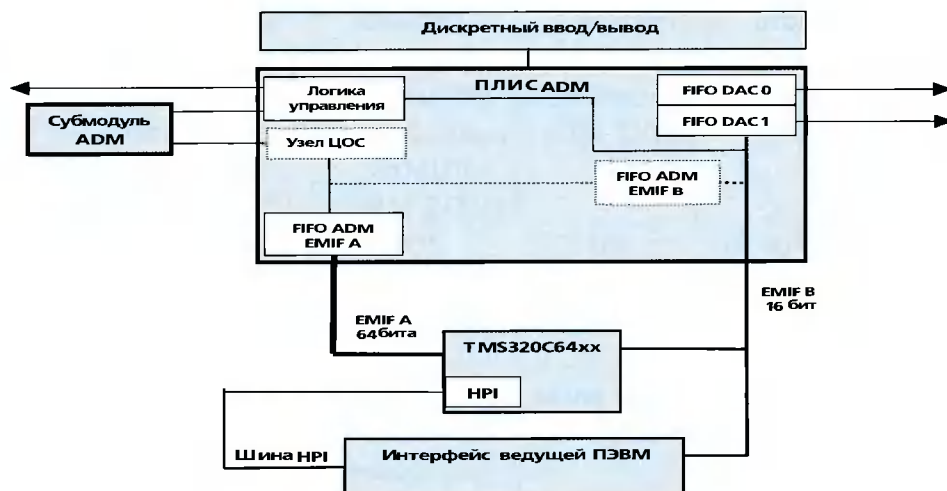


Рис. 1. Базовая структурная схема каналов передачи данных

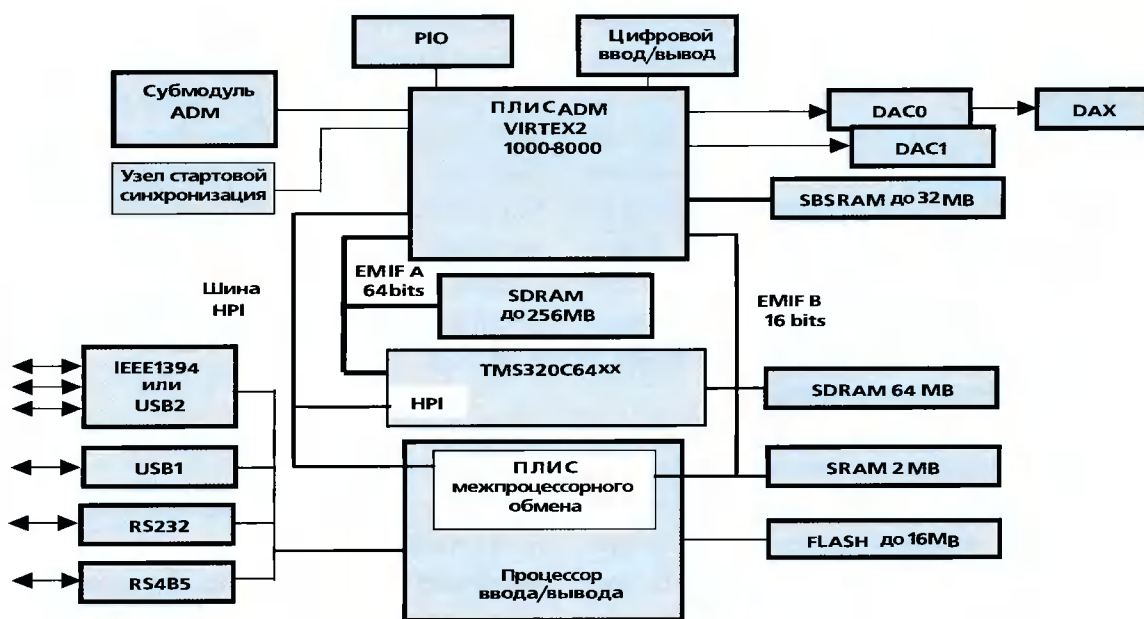


Рис. 2. Структурная схема модуля цифровой обработки сигналов ADP64Z3E

ные АЦП) или с устройствами, менее распространенными в изделиях этого класса – SCSI, EIDE и им подобные. Кроме дискретного ввода/вывода в стандарте **LVTTL**, выводы этого канала могут быть использованы для организации обменов в стандарте **LVDS**.

Набор интерфейсов связи с ведущей ПЭВМ подключен к отдельному процессору ввода/вывода, обеспечивающему все обмены между сигнальным процессором и этой ПЭВМ. Для программиста прикладной задачи этот процессор ввода/вывода является «черным ящиком». Таким образом, он может сосредоточиться только на собственной задаче, не отвлекаясь на особенности реализации различных стандартов обмена, таких как IEEE1394 или USB. Сигнальный процессор является ведомым по отношению к процессору ввода/вывода и работает под его управлением. При включении питания происходит инициализация модуля: сначала из памяти **FLASH** загружается программа работы процессора ввода/вывода, затем инициализируется ПЛИС этого процессора и далее через эту ПЛИС загружается программа работы процессора ЦОС и прошивка ПЛИС ADM.

Ресурсы памяти процессора ЦОС обеспечивают ввод и обработку достаточно больших объемов ин-

формации. Асинхронная статическая память объемом до 2 Мбайт предназначена прежде всего для ускорения выполнения задач, связанных с произвольной адресацией внешней памяти, так как случайная выборка данных из синхронной динамической памяти резко снижает общее быстродействие системы. Обработка в **ПЛИС** поддерживается памятью **SBSRAM** объемом до 16 Мбайт и быстродействием до 250 МГц. Эта память доступна только ПЛИС. Шина **HP1** данного модуля, кроме собственно межпроцессорного обмена, позволяет также вести обмен данными между интерфейсом ведущей ПЭВМ и ПЛИС ADM, не загружая этими обменами шины процессора ЦОС.

### Модуль обработки сигналов **ADP64Z2PCI**

На рис.3 приведена структурная схема модуля ЦОС **ADP64Z2PCI**. Этот модуль предназначен для установки в компьютеры с шиной **PCI** со скоростями шины до **264 МБ/сек**. Как видно из рисунка, модуль отличается от рассмотренного выше возможностью установки **второй ПЛИС**, предназначенной только для реализации алгоритмов обработки сигналов. Па-

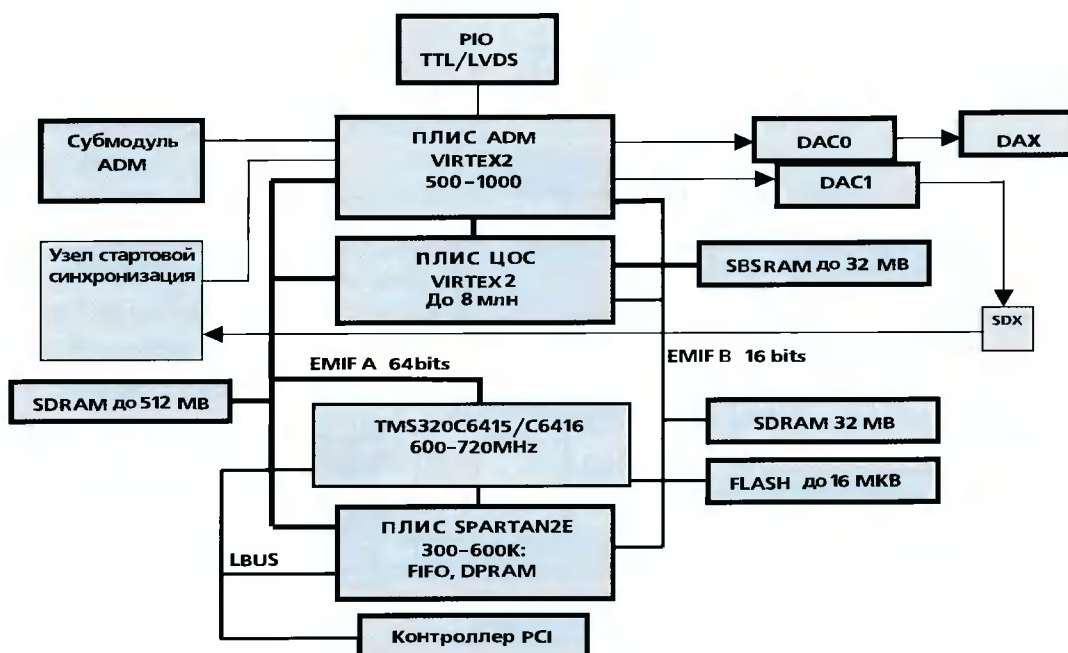


Рис. 3. Структурная схема модуля цифровой обработки сигналов **ADP64Z2PCI**

мать SBSRAM в этом модуле также подключена ко 2-ой ПЛИС. Без этой ПЛИС ЦОС структура каналов передачи данных практически идентична структуре каналов модуля ADP64Z3E. При установке ПЛИС ЦОС поток данных, принятый или уже обработанный в 1-ой ПЛИС, ответвляется на обработку во 2-ую ПЛИС и уже из нее поступает в процессор ЦОС или передается в ведущую ПЭВМ. Данная структура весьма гибка и позволяет вводить в процессор данные из любой ПЛИС, как уже подвергшиеся обработке, так и необработанные.

Используемые в качестве ПЛИС ADM микросхемы **Virtex2 XC2V500** и, тем более, **XC2V1000** позволяют, кроме управления вводом/выводом сигналов, реализовывать достаточно сложные узлы цифровой обработки сигналов. В частности, быстрое преобразование Фурье на 1024 точки занимает, по данным фирмы Xilinx, около 60%, а DDC (Digital Down Converter) – менее 50% ИМС XC2V500. Очевидно, что установка дополнительной ПЛИС ЦОС позволяет решать еще более сложные задачи.

#### Модуль обработки сигналов ADP64Z1PCI

На рис.4 приведена структурная схема модуля ЦОС **ADP64Z1PCI**. Этот модуль предназначен для

установки в компьютеры с шиной **PCI** со скоростями шины до **132 МБ/сек**. Как уже говорилось, при разработке этого изделия основной упор делался на разработку недорогого модуля обработки сигналов, имеющего тем не менее достаточно высокую производительность. В связи с этим, интерфейс шины PCI в данном модуле был реализован на встроенном в процессоры **TMS320C6415/C6416** контроллере этой шины и, соответственно, характеристики этого контроллера полностью определяют возможности обменов между процессором ЦОС и ведущей ПЭВМ. В качестве основной ПЛИС интерфейса ADM использована ПЛИС **ACEX** фирмы **Altera**, но параллельно к ней или вместо нее может быть установлена ПЛИС **Spartan2E** фирмы **Xilinx**. Функции цифровой обработки сигналов могут быть реализованы или в свободном пространстве ПЛИС ACEX, или в ПЛИС Spartan2E. Кроме того, для уменьшения количества занятых в ПЛИС контактов, обмен данными между каждой из ПЛИС и процессором ЦОС осуществляется только по шине **EMIF A** в **32-х разрядном режиме**. Шина EMIF B используется только для обменов с памятью этой шины, а также для программирования ПЛИС и управления индикаторными светодиодами. Этот модуль ЦОС располагает достаточно большими объемами памяти **SDRAM** (до 128 МБ на шине EMIF A, до

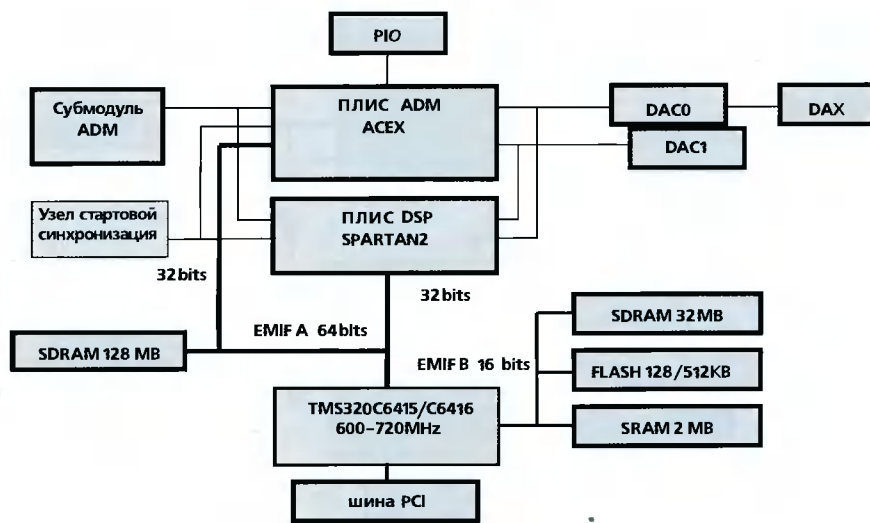


Рис. 4. Структурная схема модуля цифровой обработки сигналов ADP64Z1PCI

32 МБ на шине EMIF В и SRAM до 2-х МБ на шине EMIF В). На модуль ЦОС может устанавливаться тот же набор субмодулей ввода/вывода, но скорости обмена данными между ПЛИС и процессором ЦОС ограничены величиной **480-500 МБ/сек**, что, впрочем, достаточно для большинства вариантов его применения.

### Средства разработки и отладки прикладных задач

Все модули ЦОС данного семейства поддерживаются следующими средствами разработки и отладки программного обеспечения и прошивок ПЛИС:

1. Внешний **эмулятор** процессоров TMS320C6414/15/16 подключается к плате через внешний разъем и обеспечивает работу процессора ЦОС под управлением Code Composer Studio.

2. **Пакет Deasy Tools** предназначен для разработки программного обеспечения, исполняемого под управлением ОС **Windows 98/ME/2000/XP**.

Пакет состоит из двух основных частей: управляющей библиотеки **BARDY** и сервера **DEASY**. Библиотека BARDY позволяет разрабатывать приложения для ОС Windows, управляющие работой процессорного модуля. Она включает в себя функции инициализации, загрузки и запуска процессора ЦОС, операции обращения к памяти модуля, операции быстрого обмена данными между модулем и ПЭВМ и другие полезные возможности.

Сервер DEASY предназначен для разработки прикладных программ ЦОС на языке Си для процессора ЦОС. Он позволяет прикладным программам ЦОС включать в себя стандартные функции языка Си обращения к файловой системе и консоли ввода/вывода, такие как `printf()`, `fprintf()`, `getch()` и др. Сервер DEASY загружает такие прикладные программы на модуль, запускает их на выполнения, а затем переходит в режим терминала, обслуживая выполнение указанных выше функций.

3. **Драйверы субмодулей DAQ-DRV/DSP** представляют собой библиотеки в исходных текстах на языке Си. Их назначение – управление работой субмодуля, установленного на процессорный модуль со стороны процессора ЦОС. Драйвер позволяет выполнять начальную инициализацию субмодуля, запуск, ввод/вывод данных, а также тонкое управление субмодулем в процессе его работы.

4. Комплект разработчика ПЛИС ADM **HDK-ADP64Zxx /ADMxxx**. Этот комплект предназначен для ускорения разработки прошивок ПЛИС при реализации прикладных задач и содержит следующую информацию:

- **Проект реализации** в ПЛИС на языке VHDL интерфейса конкретного субмодуля, входящего в комплект поставки.
- **Модель модуля ЦОС** на языке VHDL. Модель включает в себя модели процессора ЦОС, ПЛИС ADM, цифрового порта и узла межмодульной синхронизации. Эта модель входит в состав проектов ПЛИС и позволяет эффективно отлаживать взаимодействие разрабатываемой ПЛИС с внешними устройствами модулей ЦОС.
- **Фрагмент принципиальной электрической схемы** модуля обработки сигналов. Этот фрагмент содержит схемы включения ПЛИС ADM и схемы узлов, связанных с интерфейсом субмодулей: узел ЦАП, узел стартовой синхронизации, узлы стартовой и межмодульной синхронизации, цифровой порт ввода/вывода, разъем ADM для установки сменных субмодулей.
- **Проект тестовой прошивки ПЛИС ADM**.
- **Описание** интерфейса конкретного субмодуля, в том числе описание сигналов и временные диаграммы работы субмодуля.

### Литература

1. Berkeley Design Technology, Inc., «DSP Benchmarks Results for the Latest Processors.» <http://www.BDTI.com>
2. Berkeley Design Technology, Inc., «DSPS ADAPT TO NEW CHALLENGES». February 2003, <http://www.BDTI.com>
3. Berkeley Design Technology, Inc., «Alternatives to DSPs», 2003, <http://www.BDTI.com>
4. J.R. Wilson, «New interconnects give DSPs a whole new level of flexibility», Military & Aerospace Electronics May, 2002.
5. Berkeley Design Technology, Inc., «Comparing FPGAs and DSPs for Embedded Signal Processing», 2002, <http://www.BDTI.com>
6. «XILINX IP SELECTION GUIDE», <http://www.partner.xilinx.com/common/coresolutions/ip/reference-guide>.



# Выставка «Электроника. Компоненты. Оборудование. Технологии» или «Праздник электроники прошел как по нотам...»

## Из официального каталога

**У**важаемые господа!

Приветствую Вас на выставке «Электроника. Компоненты. Оборудование. Технологии.»

Это первая выставка в области электроники, которая проводится силами российских компаний-производителей электроники и дистрибьюторов электронных компонентов.

Это первая выставка, которая организована в интересах России, и ее организаторами являются российские специалисты в области выставочного менеджмента.

Отраднa мысль, что появилась выставка, все средства от которой не только останутся в России, но и будут реализованы в проектах, способствующих научно и экономическому процветанию нашей страны. Также хочется отметить, что ее организаторы – дипломированные специалисты по выставочному делу, хорошо разбирающиеся в вопросах электроники, что, к сожалению, сегодня является редкостью.

Мы все понимаем необходимость развития высоких технологий. Без микроэлектроники не будет России как мощного мирового «компонента». Наша страна должна активно развиваться в этой области. Я со-

гласен с В.В. Путиным, который сказал в одном из своих выступлений, что без микроэлектроники нет страны, нет России. Это действительно так.

Наша общая задача – обеспечить среду, в которой эта отрасль сможет расти быстрее и развиваться более эффективно. Поэтому, чем больше будет таких выставок, тем успешнее будет двигаться вперед российская микроэлектроника.

Выставка – это возможность привлечения инвестиций и нормализации внутренней среды. Очень жаль, что в России сегодня часто выгодней применять импортные компоненты, а не отечественные. В этом плане выставка, которая сосредоточена на решении российских проблем, открывает прекрасные перспективы для отечественных предприятий. Именно на выставке они смогут найти ту сильную компанию, которая поможет преодолеть им сложные проблемы и создать материальную базу для развития отечественных разработок.

Хочу отметить, что электроника – это не только оборудование, станки и технологии, это не только персональные компьютеры и мобильные телефоны. Электроника – это, прежде всего, люди. Поэтому проект «Новая специальность», который осуществляется в рамках этой выставки и благодаря ее организаторам – компании ChipEXPO, необходим сегодня для решения многих задач российской электроники. Эти люди добьются результатов, благодаря которым российская электроника, я надеюсь, выйдет на мировой уровень.

Повышение квалификации и соединение научных знаний с пониманием менеджмента и маркетинга – это важнейшая задача кадровой политики, которую предстоит решить многим компаниям.

Микроэлектроника очень выгодна для всей страны, потому что с ее помощью можно построить самую передовую экономику. От всей души желаю Участникам выставки удачи, новых прекрасных перспектив и реализации поставленных задач.

**Юрий Иванович Котов,**  
Президент Московской  
торгово-промышленной  
палаты



*Почетные гости выставки: Заместитель генерального директора РАСУ Муравьев С.А., Директор московской торгово-промышленной палаты Бутов В.С., начальник отдела оборонно-промышленной политики Министерства науки, промышленности и технологий Макаренко Г.Г., Заместитель руководителя федеральной службы занятости населения по г. Москве Титов А.А.*

### Церемония открытия

До открытия выставки оставалось несколько минут. Музыканты настраивали инструменты, промоутеры растянули ленточку, журналисты достали блокноты, фоторепортеры занимали выигрышные позиции.

Церемония открытия выставки была художественно оформлена флористическими композициями в стиле «Золотая осень» и сопровождалась прекрасной музыкой в исполнении популярного джазового ансамбля солистов оркестра О. Лундстрема.

И вот – праздничные фанфары, вспышки камер, приветственные слова представителей госучреждений и московского Правительства. В своих выступлениях почетные гости говорили о необходимости и актуальности данной выставки. Они отмечали, что выставка стала достойным событием «Дня города» и ее значение трудно переоценить. Выставка такого уровня положительно влияет на решение проблем российской электроники и способствует формированию положительного имиджа страны.

Под аплодисменты празднично одетой публики и торжественную музыку атласная лента была перерезана, и поток посетителей хлынул, наконец, в выставочные залы.

### Деловая программа

Выставка напоминает работу оркестра, где все должно быть «как по нотам» – от беджей посетителей до вечернего фуршета, от охраны экспонатов до системы указателей. Все три дня работы выставки, «выставочный оркестр», состоящий из тридцати сотрудников компании-организатора, семи тысяч посетителей и ста пятидесяти компаний-участников звучал слаженно и ритмично. Аудитория собралась исключительно профессиональная. Эк-



споненты отмечали, что большая часть посетителей – специалисты, которых и ожидали увидеть на своих стендах участники. Это не было случайностью. Чтобы достичь этого результата, компания ChipEXPO вела продуманную PR-компанию, осознанно выбрав «закрытую» площадку (в ЦДХ посетители приходят целенаправленно) и заранее позаботилась о том, чтобы приглашенные билеты вовремя попали на предприятия, производства и в НИИ.

Деловая программа была насыщена разнообразными семинарами и совещаниями, на которых рассматривались наиболее актуальные вопросы отрасли. Многие посетители заметили, что благодаря этим внутренним проектам, выставка стала более значимой: она вышла за рамки коммерческого мероприятия. Теперь это уже слаженный ансамбль людей, которые хотят делать общее дело – решать проблемы отрасли. Одна из них – кадры.

Представители департамента федеральной службы занятости населения по г. Москве были удивлены активному интересу, проявленному у стенду «Ярмарка вакансий в сфере высоких технологий». По их мнению, ни на одной выставке не было так много технических

специалистов, заинтересованных в получении работы. Поскольку «кадровая» проблема является одной из центральных в российской электронике, то проведенное на выставке совещание в рамках проекта «Новая





специальность» было не только актуально, но и звучно «Ярмарке вакансий». В совещании принимали участие представители Министерства образования РФ, Департамента федеральной службы занятости населения по г. Москве, Российского Совета Ректоров, Совета ректоров вузов г. Москвы и Московской области, преподаватели ВУЗов Москвы, С.-Петербурга и Екатеринбурга и, конечно те, кто заинтересован в квалифицированных специалистах – руководители ведущих компаний-дистрибьюторов электронных компонентов. В ходе совещания отмечалось, что выпускники технических ВУЗов обладают знаниями «прошлого века», актуальными для российской электроники 90-х годов, и что учебная программа за минувшее десятилетие практически не изменилась. Поэтому, развитие российского рынка электронных компонентов требует не только создания новых рабочих мест (вакансий очень много), сколько пересмотра учебных программ и повышения квалификации выпускников. Мнения относительно объема знаний будущих специалистов разделились. Москвичи, в основном, говорили о том, что технические знания должны соответствовать современному состоянию электроники. Представители Санкт-Петербургских компаний настаивали на том, что больше внимания должно уделяться развитию «коммуникативно-маркетинговых» навыков. Конечно, решение этой проблемы требует и выпуска новых учебных пособий, и создания специализированных лабораторий, и введения новых предметов, и ознакомления с зарубежным опытом. Было высказано много идей, предложений и даже готовых решений. Обсуждался также вопрос финансирования проекта «Новая специальность», и все участники согласились с тем, что решение могло бы стать общим как для государства, заинтересованного в создании рабочих мест, так и для компаний-дистрибьюторов электронных компонентов, которым нужны специалисты. На совещании было принято ре-

шение направить все материалы в Министерство образования РФ и в течение года планомерно решать поставленные задачи. Хочется надеяться, что проект «Новая специальность» будет воплощен, и что на второй выставке в 2004 году, на аналогичном совещании будут представлены его прекрасные результаты.

Не менее интересными были и другие совещания. У всех, кто участвовал в выставке и кто на нее пришел в качестве гостя, была возможность посетить и обсудить финансовые вопросы, проблемы таможи и импорта электроники, вопросы сертификации электронных компонентов, статус «второго поставщика» и многие другие. Всего во время выставки было проведено семнадцать научно-технических семинаров и пять совещаний. Для обсуждения проблем развития дистрибуции, Автономный Регистр Дистрибьюторов Электронных Компонентов (АРДЭК) провел «встречу без галстуков», на которой были перерегистрированы члены ассоциации, а также проведена регистрация новых. О своем намерении вступить в АРДЭК заявило более 50 организаций.

### **Нестандартные проекты**

Помимо деловой программы, на выставке были представлены нестандартные внутривыставочные проекты.

Куратор проекта «Музей электроники» – сотрудник Политехнического Музея М.Э. Смолевицкая отметила в своем интервью большое количество посетителей «выездного музея» и подчеркнула, что молодежь проявляет к истории электроники и вычислительной техники гораздо больший интерес нежели старшее поколение. Интересно, что иностранцев, посетивших «Музей электроники», более других привлекли два экспоната – персональный компьютер «Истра» и один из первых отечественных диктофонов.

К нестандартным проектам можно отнести и «сервис-центр». Со стороны организаторов это был эксперимент: бесплатная копировальная техника, принтер, телефонная связь, ноутбуки, Интернет и Post-office. Спросом пользовались все услуги, кроме курьерских. Вероятно, это связано с тем, что они предоставлялись за плату, а может быть так произошло потому, что российским посетителям выставки доставляет удовольствие ходить с тяжелыми пакетами, набитыми печатной рекламой. Однако организаторы уверены, что в следующем году ситуация изменится.

К сожалению, в силу непредвиденных технических проблем не все выставочное время был предоставлен доступ в интернет.

Аудитория моложе 35 лет особенно отметила такие проекты, как «ярмарка вакансий», «музей электроники» и выставку фрактальной графики. В «Овальной зале» были представлены работы доктора ф.-м. наук П. Николаева, и многим посетителям было любопытно наблюдать творческое применение возможностей электроники.

Посетители от 32 до 50 лет говорили о конкретных запланированных встречах на выставке и благодарили организаторов за идею создания «тематических зон», систему навигации и наличие в выставочных залах сотрудников ChipEXPO «в красным майках». По мнению большинства из них, благодаря этим возможностям, ориентироваться на выставке было удобно и что для деловых людей особенно важно – такой подход к организации пространства позволял сэкономить время.

Также посетителям очень импонировало то, что на территории выставки можно было бесплатно выпить кофе и соки от компании Cafe-Montere. Особенно это было важно для студентов и служащих, которые провели здесь много часов.

### **Виртуальные посетители**

Выставку можно было посетить, не выходя из дома. В сотрудничестве с компанией «ChipInfo» на протяжении трех дней организаторы проводили прямую

Интернет-трансляцию выставки для виртуальных посетителей и заочных участников.

### **Электроника. Компоненты. Оборудование. Технологии в 2004 году**

В следующем году выставка будет проходить 5-7 октября в том же выставочном комплексе Центральный Дом художника.

Около 50 участников выставки уже подтвердило свое желание участвовать во второй выставке. Организаторы планируют сделать деловую программу следующей выставки более разнообразной, а к участию в ней привлечь экспонентов из Западной Европы и Америки. Надеемся, что в следующем году выставка сможет порадовать всех научно-техническими семинарами и симпозиумами с участием известных специалистов международного уровня, новыми интересными проектами, а также станет достойным местом для заключения международных контрактов.

Для достижения этих целей, «оркестрик» компании ЧипЭкспо постарается правильно использовать полученный опыт и в будущем октябре «сыграть» прекрасную «золотую осеннюю симфонию» выставочного профессионализма.

**Материал подготовила  
М. Флетчер**

## **Уважаемые читатели!**

Со II полугодия 2004 г. вы можете подписаться на наш журнал в любом отделении связи. Подписной индекс издания по каталогу ОАО «Роспечать» – **82185**.

Кроме того, сохраняется возможность оформить подписку непосредственно в редакции. Для этого необходимо заполнить подписной лист и отправить его по факсу (095) 290-9085 или по электронной почте [info@dspa.ru](mailto:info@dspa.ru)

### **Подписной лист**

Наименование организации \_\_\_\_\_  
ИНН \_\_\_\_\_  
Почтовый адрес \_\_\_\_\_  
Контактное лицо \_\_\_\_\_  
Телефон, факс, электронная почта \_\_\_\_\_

### **Банковские реквизиты**

Наименование банка \_\_\_\_\_  
Р/с \_\_\_\_\_ Город \_\_\_\_\_  
К/с \_\_\_\_\_ БИК \_\_\_\_\_

**Справки по телефонам: (095) 290-9088, 290-9085  
или по электронной почте [info@dspa.ru](mailto:info@dspa.ru)**

## Рецензия на книгу

### А.Б. Сергиенко «Цифровая обработка сигналов»

СПб.: ПИТЕР, 2002 .

/Учебник для вузов/

**А. А. Ланнэ**

Издательство ПИТЕР в 2002г выпустило в свет книгу доцента С.Петербургского Электротехнического Университета А.Б. Сергиенко «Цифровая обработка сигналов» под грифом «Допущено Министерством образования Российской Федерации в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению подготовки дипломированных специалистов «Информатика и вычислительная техника». Рецензенты – специалисты Московского Государственного технического университета (МГТУ) им. Н.Э. Баумана.

В аннотации сказано, что учебник представляет собой базовый курс по цифровой обработке сигналов. Теоретические сведения сопровождаются примерами реализации обсуждаемых алгоритмов с помощью системы MATLAB.

Сразу же следует отметить, что издательство ПИТЕР проявило полезную инициативу, поддержав направление «Цифровая Обработка Сигналов» (ЦОС). Именно в этом направлении высоких технологий наблюдается серьезное отставание реальных технических разработок и подготовки кадров в нашей стране от событий, происходящих в мировой науке и технике. Вместе с тем ЦОС по существу стала основой таких дисциплин как инструментальная информатика, теория и практика измерений физических величин, цифровое вещание и телевидение, речевые технологии, распознавание звуковых и видео образов, модемостроение, интернет-технологии, автомобильная и медицинская электроника, техника высококачественного воспроизведения звука, радиолокация и радионавигация, разнообразные элементы телекоммуникационных систем и многое другое. Образно говоря, можно утверждать, что ЦОС стало новой ветвью теоретической радиотехники, теории цепей и сигналов со своими специальными проблемами, самостоятельными для определения ЦОС в качестве самостоятельного научного и научно-технического направлений.

В международном обществе IEEE существует «SIGNAL PROCESSING SOCIETY» со своим журналом и множеством иных предпочтений. Деятельность этого сообщества в существенном посвящена проблемам ЦОС. Задачи, методы и средства ЦОС изучаются практически во всех университетах развитых стран, а DSP(ЦОС) – инженеры

составляют едва ли не самую дефицитную популяцию выпускников-инженеров и аспирантов.

Все это сказано с единственной целью: поддержать издание книг по ЦОС, поддержать создание базовых дисциплин ЦОС в институтах, университетах и академиях, обосновать необходимость подготовки и переподготовки кадров по ЦОС и ее приложениям.

Вернемся к анализу книги. При ее подготовке, на мой взгляд, допущено две группы погрешностей (или просчетов): стратегические и тактические.

К стратегическим просчетам следует отнести отбор материала. По этому параметру книга устарела на несколько лет, если не сказать более. В ней не отражены, либо затронуты поверхностно, вопросы, которые для разработчиков, исследователей, а, следовательно, и студентов являются базовыми. К их числу нужно отнести адаптивную фильтрацию, многоскоростную дискретизацию и банки фильтров, частотно-временной анализ и нелинейную цифровую обработку сигналов, основы обработки многомерных сигналов. Все эти разделы выпали из книги или рассмотрены недостаточно. Книга посвящена важным, но все-таки, архаичным темам: цифровой фильтрации и спектральному анализу. Новым и редко встречаемым в учебниках по ЦОС является раздел, посвященный модуляции и демодуляции и это несомненное достоинство книги. Таким образом, базовый учебник по ЦОС лишен многих основных атрибутов дисциплины. Следует отметить, что опущенные упомянутые разделы востребованы квалифицированными инженерами-разработчиками и инженерами по эксплуатации аппаратуры. Так, без знания адаптивных фильтров невозможно построение обнаружителей, компенсаторов эхо, устройств компрессии и распознавания речи. Нелинейная ЦОС требуется в задачах построения обнаружителей, демодуляторов, устройств распознавания образов и пр. Частотно-временной анализ совершенно необходим радиотехникам, занятым проектированием устройств радиолокации и навигации, специалистам в области телевидения и видеоконференц-связи, использующих компрессию изображений. Многоскоростная дискретизация (интерполяция и децимация) применяется в модемостроении, в задачах обработки звука, при узкополосной фильтрации

и др. Основы теории цифровой обработки многомерных сигналов необходимы специалистам в области визуальной техники, телевидения, техники измерений и др.

Сказанное, на самом деле, хорошо известно специалистам в области ЦОС. (см. Дискуссию в обществе «SIGNAL PROCESSING SOCIETY» ( журнал «SIGNAL PROCESSING magazine» 1999г. Том16, N1, с.16-31).

Таким образом, современный учебник по ЦОС должен содержать все базовые атрибуты данной науки. Способ изложения, доступность и уровень детализации – проблемы автора.

К сожалению, в своей основе рецензируемый учебник отражает содержание ЦОС, мало отличающееся от того, что было сделано в книге Б.Голд, Ч.Рабинер «Цифровая обработка сигналов», М. Советское радио, 1973, 367стр. Номенклатура, рассмотренных в цитированной выше книге вопросов, шире. Исключение составляет раздел, о котором говорилось ранее и примеры, которые по своему содержанию и способу представления богаче.

Вся прикладная часть – программное обеспечение, примеры расчета и моделирования, выполнена на базе известного в мире пакета MATLAB. В принципе – это несомненное достоинство издания, – даны полезные приложения по программному обеспечению. Беда здесь одна. MATLAB – дорогой пакет. И очень мало вузов, обладающих даже учебной (существенно усеченной) версией пакета ( конечно же, речь идет о лицензированных версиях продуктов). В такой обстановке возникает естественный вопрос: «Как пользоваться программным обеспечением в учебном процессе?» Не случайно многие вузы в РФ, да и за рубежом разрабатывают оригинальные пакеты для решения задач ЦОС, естественно менее универсальные и мощные, зато и менее дорогие. (По нашей информации легальную учебную версию пакета MATLAB получили 40 вузов России, из них 4 в С.Петербурге. Сегодня не один вуз связи не обладает таким пакетом.). Кроме того, методически часто бывает полезно сделать простую учебную программу на данную тему. Таким образом, наличие пакета MATLAB является полезным, но недостаточным условием подготовки инженеров DSP. Это следует, на наш взгляд, иметь в виду при постановке дисциплины ЦОС в учебных заведениях.

Что касается тактических недостатков, то их много. Они касаются терминологии, определений, вариантов изложения различных разделов, фактических ошибок, важных недоговоренностей и т.д. Нашей целью не является детальный анализ всего перечисленного. Ограничимся несколькими характерными примерами.

Терминология и определения. Во многих местах книги встречаются неудачные и неправильные термины и определения. Импульсная характеристика (стр.88) опре-

делена без учета начальных условий. Это принципиальная ошибка. Такая же ошибка допущена на стр. 89 при определении переходной характеристики. При описании пространства состояний (стр.96) переменные состояния, уравнения состояния и наблюдения не определены.

В словаре Ожегова С.И. читаем: «Кoeffициент – это числовой множитель ... или величина, определяющая какое либо свойство..., или число, на которое нужно и т.д.»


В тексте книги коэффициентом (коэффициентом передачи, комплексным коэффициентом передачи) называют комплексную функцию. Этот же коэффициент в других местах книги, (см., например, стр. 89,90,97), при замене мнимого аргумента комплексным называют функцией передачи. Таким образом, допускается сразу же несколько погрешностей: функцию называют числовым множителем, что справедливо только на заданной частоте, вводится расхождение между классической теорией цепей (см., например, учебники А.Ф.Белецкого и П.Н.Матханова), в неявном виде говорится, что, заменяя одну переменную на другую, можно от функции перейти к числовому множителю.

Не введено понятие, связанное с описанием динамики дискретной системы – уравнение «вход-выход». В отечественной литературе практически не используется термин «все пропускающие фильтры» (стр.196 и далее). Общепринятым является термин «фазовые звенья». Введение новых терминов на наш взгляд нецелесообразно без жесткой необходимости.

В теории управления, теоретической радиотехнике, теории обработки сигналов при рассмотрении структурных схем, схем фильтров и др., используется терминология теории многополюсников, в частности, четырехполюсников. В рецензируемой книге, как и во многих иных, вместо «каскадного соединения» используется термин «последовательное соединение». Но в теории четырехполюсников это разные вещи. Студент, прослушавший дисциплину ТЭЦ или ТОЭ, испытывает недоумение, когда каскадное соединение звеньев назовут последовательным, а последовательное соединение (см. нашу книгу) вообще не будет использовано.

Начальные условия. Начальные условия в книге практически не рассматриваются. Не определена связь начальных условий с Z-образами сигналов, решением разностных уравнений. В такой ситуации затруднено, если не сказать исключено, изложение вопросов устойчивости. Многие прикладные задачи связаны с учетом начальных условий.

Недоговоренности и фактические ошибки. Неудачно на наш взгляд изложен важный раздел, посвященный устойчивости. Здесь неверно представлена логическая последовательность изложения. Она должна быть такой:



физическое определение устойчивости, критерии устойчивости, следствия. Действительно, физически устойчивость может быть определена по-разному. Устойчивость по Ляпунову, устойчивость по соотношению «вход-выход», абсолютная устойчивость относительно изменения всех, либо части параметров и т.д. По этому поводу можно обратиться к книгам Лешаца С. по проблемам устойчивости систем управления. Физически речь идет о том, какую систему считать устойчивой. Ту, что практически не реагирует на малое «шевеление» начальных условий (устойчивость по Ляпунову), либо ту, что не реагирует на «шевеление» параметров (абсолютная устойчивость к изменению параметров), либо, наконец, устойчивость по соотношению «вход-выход», когда реакция системы при нулевых начальных условиях на произвольное ограниченное воздействие ограничена. Вот здесь-то и происходит путаница, и возникают ошибки, которых не избежал А.Б. Сергиенко. Введя на стр. 95 определение устойчивости по Ляпунову (по сути дела это так, хотя явно об этом по непонятным причинам не сказано), автор книги далее утверждает, что требование устойчивости (по Ляпунову) «равносильно затуханию импульсной характеристики», т.е. стремлению импульсной характеристики к нулю, когда время неограниченно возрастает. Ошибка! На самом деле это необходимое условие, но не достаточное. Легко привести пример системы, в частности, дискретной цепи, которая устойчива по соотношению «вход-выход» (ее импульсная характеристика затухающая), но неустойчива по Ляпунову. Для этого нужно построить систему с передаточной функцией, имеющей сокращающийся нуль и полюс, причем полюс располагается вне единичного круга.

В теории ЦОС широко используются фильтры с комплексными коэффициентами. По этой причине утверждение, что «полюсы функции передачи могут быть вещественными, либо составлять комплексно-сопряженные пары», вообще говоря, ошибочно (см. стр.94, 95). Здесь необходимы дополнительные комментарии.

Много замечаний вызывает раздел, посвященный расчету фильтров. Не сказано, по какой причине используется набор передаточных функций различного типа, а не передаточные функции только Чебышева 1 или, скажем, Баттерворта. А дело здесь в критериях оптимальности. Практически все использованные передаточные функции оптимальны в том или ином смысле. Этот важный момент упущен в книге. По непонятным причинам фильтры с изохрематическими характеристиками затухания (АЧХ) определены только как эллиптические фильтры или фильтры Кауэра. На самом деле для построения передаточных функций таких фильтров использовано решение 3-ей задачи Золотарева, что и применил В.Кауэр в своих работах по синтезу ЛС фильтров. По этой причине справедливо отме-

ченные фильтры называть эллиптическими (решение задачи Золотарева выражается через эллиптические функции) и фильтрами Золотарева-Кауэра, как это и принято в отечественной литературе по синтезу фильтров.

При синтезе фильтров неудачно рассмотрен вопрос о нормировании характеристик по частоте. Студент должен понимать, что это обстоятельство не догма, а предмет соглашения. Так, правая граница полосы пропускания ФНЧ в шкале нормированных частот может быть принята равной 1, может быть и другой. Далее, на границе (ФНЧ, правой) у разных типов фильтров затухание может принимать различные значения. Например, у фильтров с равноволновыми характеристиками затухание принимается равным максимально возможному значению в полосе пропускания. У фильтров Баттерворта, которые имеют максимально плоскую характеристику затухания, на границе полосы затухание принимается равным 1 дБ, а вовсе не 3 дБ, как об этом говорится в рецензируемом учебнике. Затухание 3дБ является широко распространенной цифрой, особенно в среде радистов. Но в кругу инженеров-связистов, инженеров в области измерений, да и в общетеоретической литературе по фильтрам, на границе полосы пропускания у фильтров Баттерворта может быть задано любое желаемое (разумное) значение затухания. Соответствующие расчетные соотношения легко найти в книгах по синтезу аналоговых фильтров.

Завершая эту часть, заметим, что при изложении учебного материала на наш взгляд необходимо говорить об общих принципах нормирования и создания (конструирования) прототипных передаточных функций, а уже потом обсуждать, как это сделано в системе MATLAB.

Здесь мы переходим еще к одному важному тактическому недостатку книги – излишне обширной части изложения материала в справочном стиле, да к тому же без ссылок на литературу. Приведем лишь некоторые примеры.

Формулы перехода от прототипных передаточных функций аналоговых фильтров к передаточным функциям дискретных цепей не обоснованы. Непонятно, как и почему будут пересчитываться характеристики, когда это возможно и когда нет, допустим ли подобный пересчет для других цепей (амплитудных и фазовых корректоров, фазовращателей и т.д.).

Многие факты в книге постулируются без должного основания и в ущерб подготовке студентов. Основопологающе факты стр.130, формула (3.1.) стр. 131 – даны без обоснования и ссылок. По этой причине вывод фундаментального соотношения (3.7), связывающего спектры дискретного и порождающего аналогового сигналов, не убедителен. Здесь уместен иной вывод, широко используемый в учебной и монографической литературе.

Неудовлетворительно, на наш взгляд, изложен вопрос о нерекурсивных фильтрах с линейными ФЧХ (стр.204). Симметрия и антисимметрия коэффициентов передаточной функции (отсчетов импульсной функции) должны быть доказаны, а не постулированы. Этот факт имеет важное научно-психологическое значение. Студент, воспитанный в «аналоговой» среде (ТЭЦ, ТОЭ, ТОР) практически впервые обнаруживает объект, в котором строго может быть сформирована линейная ФЧХ и при этом реализованы произвольные АЧХ, в частности, фильтры различных типов. Симметрия (антисимметрия) коэффициентов – следствие линейности ФЧХ, равно как и строгая линейность ФЧХ порождает симметрию (антисимметрию) коэффициентов.

Много неточностей и небрежностей в изложении встречается и в других местах книги. В параграфе «Фильтры Чебышева первого рода» без обоснования и без ссылок на литературу утверждается, что при частоте, стремящейся к бесконечности, АЧХ фильтра Чебышева первого рода является максимально плоской (стр.108). Рубрика «Внимание» стр.110, вероятно, относится к особенностям пакета MATLAB, но никак не связана с общей теорией цифровых и аналоговых фильтров. На стр.112 при описании АЧХ эллиптического фильтра функция  $R(\omega)$  названа рациональной функцией Чебышева  $n$ -ого порядка. На самом деле эта функция как раз и является решением 3-й задачи Золотарева, хотя формально она и может быть получена из дробей Чебышева, если известны полюсы. Но они-то как раз в рамках дробей Чебышева неизвестны и для изокстремального случая их обнаружил Золотарев, поставив и решив свою 3-ю задачу.

В параграфе «Оптимальные методы» справедливо вводится понятие нормы и буквально двумя абзацами ниже это понятие искажается, когда для ошибки аппроксимации вводится соотношение (6.5) стр.318 или соотношение (6.6) на этой же странице. Дело в том, что для оценивания ошибки упомянутые формулы могут быть использованы, но они не являются нормами, т.к. не удовлетворяют соответствующим условиям.

По отечественному ГОСТу термины «групповая задержка» или «групповое время задержки», используемые в учебнике, не применяются. Вместо них используется термин ГВП – групповое время прохождения. Это замечание относится ко многим разделам книги.

В разделе «Теорема Котельникова», во-первых, ничего не говорится о других международных именах этой теоремы (теорема отсчетов, теорема Шенона). Во-вторых, когда речь идет о восстанавливающем фильтре ничего не говорится о ФЧХ фильтра. Это существенное упущение, так как, если ФЧХ отличается от линейной зависимости, то восстановления сигнала не произойдет. Рис. 3.11 спра-

ведлив только при условии нулевой ФЧХ. Студент обязан понимать, что импульсная характеристика линейной системы определяется АЧХ и ФЧХ и знать соответствующее соотношение.

В книге отсутствует четкая классификация сигналов по характеристикам «состояние-время», что крайне важно для всего содержания книги и формирования научного мировоззрения студента (следует, однако, заметить, что некая неявная и неполная классификация сигналов в книге имеется). Несмотря на заявления типа «Классификация систем», – классификация даже рассматриваемых систем не приведена. Какие линейные дискретные стационарные причинные системы (ЛДС) обсуждаются? Четкого ответа на этот вопрос в учебнике нет. И только искусственный читатель может догадаться, что речь идет о ЛДС, описываемых линейными разностными уравнениями с вещественными и постоянными коэффициентами. Такие ЛДС не единственно возможные.

В учебнике практически не рассмотрен важный для ЦОС вопрос о шумах дискретизации, о шумах арифметики и о проблемах, связанных с минимизацией этих шумов и максимизацией динамического диапазона.

На наш взгляд, эти проблемы, хотя бы в постановочной части, должны быть рассмотрены в учебном пособии по ЦОС. К ним очень тесно примыкает вопрос об эквивалентных дискретных цепях. В неявном виде он затрагивается при рассмотрении различных методов реализации. Однако явная постановка задачи о поиске рациональных, либо наилучших схем (алгоритмов) реализации дискретных цепей отсутствует.

Во всех затронутых выше задачах очень важен оптимизационный аспект. Как и в вопросе конструирования передаточных функций этот аспект в книге, по существу, отсутствует. Некоторые объявления типа параграфа «Оптимальные методы» не меняют существа дела. Что же касается субоптимальных методов, то здесь вопрос и вовсе не ясен. Что это такое? Как объяснить студенту или аспиранту, что называть субоптимальным методом? На наш взгляд речь может идти о неоптимальных методах, применение их заметно проще и которые приводят к неоптимальным конструкциям.

Каковы же общие выводы. Они не однозначны. Для искусственного читателя, способного отделить «зерна от плевел», книга может быть полезной как справочное пособие по применению системы MATLAB в задачах ЦОС. Как учебник по ЦОС эту книгу рекомендовать нельзя из-за большого числа просчетов, допущенных при ее подготовке. Ряд погрешностей было отмечено в тексте заметки. Многие другие подготовленный читатель без труда найдет, ознакомившись в деталях с содержанием рецензируемой книги.