

## РЕАЛИЗАЦИЯ МНОГОКАНАЛЬНЫХ РЕЧЕВЫХ CELP КОДЕКОВ НА DSP TMS320C548.

*Андреев И. В., Бабкин В. В., Знамеровский А. Е.*

Центр Цифровой Обработки Сигналов,  
Санкт-Петербургский государственный университет телекоммуникаций им. проф.  
М.А. Бонч-Бруевича,  
193232, Санкт-Петербург, пр. Большевиков д.24, тел. 589-52-23, 589-82-43  
E-mail: barabashka@fnmail.com, aznamer@nevsky.net

**Реферат.** В докладе представлена реализация многоканального речевого CELP кодека, ориентированного на применение в системах с пакетной передачей данных, а так же многоканальные варианты стандартных алгоритмов G723.1, G.729a/b на DSP семейства TMS320c54x. Обсуждаются способы перевода алгоритмов в арифметику с фиксированной точкой и особенности их ассемблерной реализации. Излагаются методы создания многоканальных вариантов вокодеров. Приводятся результаты.

### 1. Введение.

Для практической реализации новых алгоритмов компрессии речи (вокодеров) на цифровых процессорах обработки сигналов (DSP), использующих арифметику с фиксированной точкой (ФТ), необходимо решить следующие задачи:

Во-первых, моделирование алгоритмов вокодеров ведется как правило на персональном компьютере (ПК) на языках высокого уровня (ФОРТРАН или Си) в арифметике с плавающей точкой (ПТ), поэтому переход к реализации на DSP требует предварительной разработки на ПК алгоритма, функционирующего в арифметике с ФТ с сохранением качественных показателей.

Во-вторых, высокая вычислительная сложность алгоритмов и условие работы в реальном масштабе времени требуют эффективного использования специфической архитектуры DSP и приводят к необходимости написания программ на уровне ассемблера.

В-третьих, при построении многоканальных систем необходимо рационально реализовать несколько независимых речевых каналов на одном DSP.

Статья построена следующим образом. В разделе 2 рассматриваются задачи, возникающие при переводе модели алгоритма из арифметики с ПТ в арифметику с ФТ. В разделе 3 кратко рассмотрены методы ассемблерной реализации на DSP. В разделе 4 излагаются методы создания многоканальных вариантов вокодеров и в разделе 5 приводятся полученные результаты.

### 2. Подготовка целочисленной модели.

В нашем центре был разработан CELP кодек на скорость 4.6 кбит/с, ориентированный на применение в системах с пакетной передачей данных [1]. Кодек имеет эффективный механизм интерполяции потерянных пакетов, быстрый поиск в алгебраической кодовой книге и встроенный детектор речевой активности VAD, позволяющий уменьшать выходной поток речевых данных в паузах до 200 бит/с.

Исходная модель вокодера была написана на языке Си в арифметике с ПТ, так как это позволяло сосредоточиться на проработке функциональных блоков, не отвлекаясь на проблемы, связанные с точностью вычислений. В качестве конечной платформы для реализации был выбран 16 разрядный DSP TMS320c549, имеющий эффективную внутреннюю архитектуру, большую внутреннюю память 32 К слов, малое потребление энергии и производительность 100 MIPS. Поэтому, перед реализацией на DSP, было необходимо адекватно перевести алгоритм вокодера в арифметику с ФТ. При этом необходимо было найти баланс между сохранением качества кодирования речи и требованием работы в реальном масштабе времени при дальнейшей реализации на DSP.

Для организации вычислений в ФТ следует обратить внимание на следующие моменты:

- Оценка динамического диапазона внутренних переменных
- Оценка необходимой точности представления внутренних переменных

- Исключение переполнения разрядной сетки
- Применение набора базовых операций, моделирующих инструкции DSP

При переводе модели вокодера нами использовалось масштабирование внутренних переменных с применением различных форматов представления чисел от Q0 до Q15. Диапазон возможных значений параметров определялся либо аналитически, либо с использованием тестовых сигналов. При выборе формата выбирался баланс между точностью представления переменной и вероятностью переполнения разрядной сетки. Для избежания переполнения все вычисления производились с насыщением.

При динамическом диапазоне значений переменных большем 16 бит или при их высокой чувствительности к погрешностям, использовалась с двойная точность (32 бита). Например, так хранятся коэффициенты автокорреляционной функции и значения энергии, а так же промежуточные результаты умножений и свертки. Другим вариантом (удобным в случае массива параметров) является введение блоковой экспоненты.

Следует учитывать, что детали реализации арифметики целых чисел для DSP отличаются от арифметики целых чисел языка Си. Это связано с аппаратной поддержкой в DSP операций насыщения, округления и умножения чисел в формате Q.15. Поэтому, для адекватного перевода алгоритма речевого кодека в арифметику с фиксированной точкой обычно используется библиотека базовых операций, например предоставляемая вместе с рекомендациями G.729, G.723.1 [5,6]. Большинство операций из этого базового набора соответствует типовым инструкциям ассемблера для сигнальных процессоров. Этот набор стандартизован, но не все базовые операции прямо совпадают с инструкциями конкретных типов DSP. Поэтому, при реализации стандартов, необходимость соответствия тестовым векторам приводит к записи таких операций в виде комбинации ассемблерных команд DSP.

При реализации собственных алгоритмов вокодеров удобно оптимизировать стандартный набор базовых операций с учетом архитектуры и набора команд DSP, на котором осуществляется реализация. Поэтому нами была изменена операция деления, добавлены функции логарифмирования и возведения в степень (в зависимости от ограничений, выполнение этих действий производится либо алгоритмически, либо по таблицам).

### 3. Ассемблерная реализация.

Набор команд процессора для организации максимально быстро исполняемого кода частично закладывается в выборе базовых операций при переводе модели в арифметику с ФТ. При использовании стандартной библиотеки базовых операций перевод языка базовых операций на язык ассемблерных команд процессоров семейств TMS320C54x, ADSP-218x требует дополнительных вычислительных затрат, так как некоторые базовые операции не имеют прямых аналогов в ассемблерных командах и требуют записи в несколько действий. Это касается умножений с повышенной точностью (типа 32 на 16 бит) для процессоров семейства TMS320C54x или сдвига влево с насыщением для процессоров семейства ADSP-218x [3].

Кроме этого, различные семейства процессоров имеют особенности по организации конвейерных и параллельных вычислений, которые не могут быть учтены на этапе моделирования в базовых операциях, но могут дать существенный выигрыш в быстродействии, например, расширенный набор регистров и инструкций или аппаратная поддержка стека.

Основную трудность отладки при реализации стандартов G.723.1, G.729 представляло требование побитового соответствия тестовым векторам (последовательностям), которые гарантируют как совместимость различных реализаций одного стандарта между собою, так и соответствие ассемблерной реализации и модели в базовых операциях. На практике это приводит к требованию побитового соответствия входов и выходов всех внутренних функций.

При реализации собственного алгоритма вокодера тестовые вектора были выбраны самостоятельно на этапе создания и проверки целочисленной модели вокодера в базовых операциях. В качестве тестовых были использованы как нормальные входные речевые сигналы различных дикторов, так и сигналы, записанные с перегрузками (например, с ограничением по входу), с фоновыми шумами, помехами или даже чисто стохастические колебания. Основными целями при выборе тестовых векторов были проверка качества речевого кодирования и мак-

симально широкий охват всех возможных внутренних состояний вокодера. Это позволило выявить большинство несоответствий между моделью в ФТ и её реализацией на ассемблере.

#### 4. Реализация многоканальности.

При реализации многоканальных систем возникают проблемы рационального использования ресурсов времени и памяти DSP. Основная проблема реализации многоканальности – влияние повторных вызовов процедур кодека друг на друга. Влияние происходит из-за того, что вокодер проводит долговременный анализ и некоторые массивы данных (например, речевой буфер, на котором проводится линейное предсказание) хранят информацию о предыдущих кадрах. Поэтому, если делать последовательные вызовы процедур кодека, то данные будут накладываться друг на друга.

Решить проблему можно разными способами, но все они требуют разделения массивов данных, используемых кодеками, на те, которые задействованы в кратковременном анализе (локальные переменные, временные массивы и пр.) - их сохранять не надо - и те, что участвуют в долговременном анализе и требуют хранения между вызовами, т.е. контекстно-зависимые данные (контекст). К ним относятся: память речевого буфера линейного предсказания, квантованные LSP от предыдущих кадров, адаптивная кодовая книга и пр. Рассмотрим возможные варианты сохранения контекста.

##### 4.1 Метод отдельного программного кода.

Это простейший метод разделения контекстов (рис. 1). Кодеки для разных каналов компилируются с привязкой к разным адресам массивов контекстно-зависимых данных.

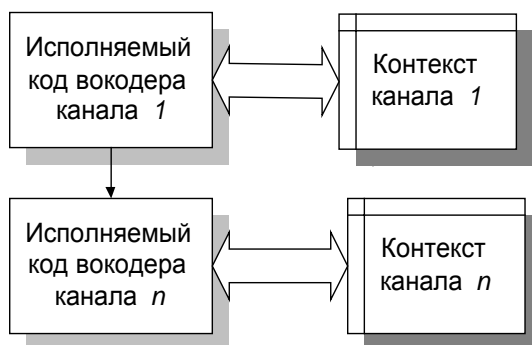


Рис.1. Метод отдельного программного кода

Выигрыш по времени максимальный, но тратятся дополнительные ресурсы памяти из-за нескольких копий одного и того же программного кода и данных.

##### 4.2 Метод переключения страниц памяти.

В данном методе (рис. 2) код программы один, а контексты данных располагаются в разных страницах памяти, причем находятся по одинаковым адресам

При вызове кодека определенного канала происходит подключение соответствующей ему страницы памяти. Достигается выигрыш по времени переключения между каналами и по памяти, но этот метод часто невозможно реализовать на сигнальном процессоре без специальной аппаратной поддержки (подключение внешней памяти с механизмом, допускающим переключение страниц) и такую реализацию труднее встроить в общую систему речевой обработки, которая, конечно же, может не ограничиваться несколькими вокодерами. Кроме того, располагая данные на внешних страницах памяти, мы проигрываем во времени обращения к ним по сравнению с внутренней памятью DSP из-за дополнительных тактов ожидания, а это может ухудшить итоговые характеристики по скорости в два или более раза.

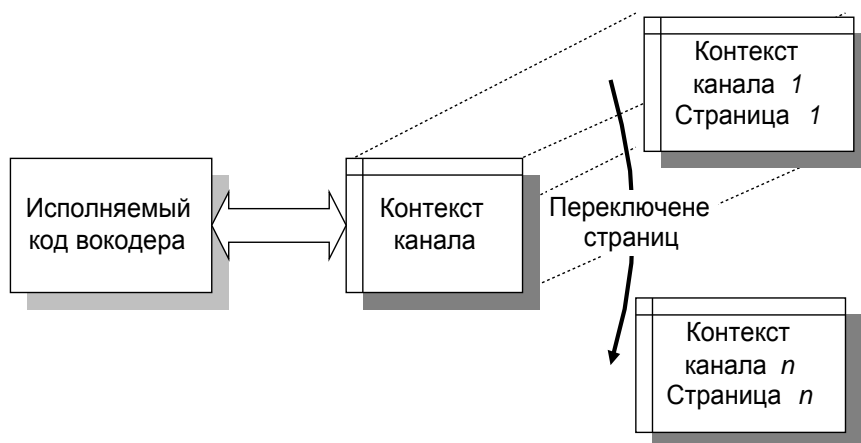


Рис.2. Метод переключения страниц памяти

#### 4.3 Метод копирования контекстов.

В данном варианте (рис. 3) контексты хранятся во внешних массивах и при вызове кодека копируются в рабочие массивы, которые могут быть расположены во внутренней памяти DSP.

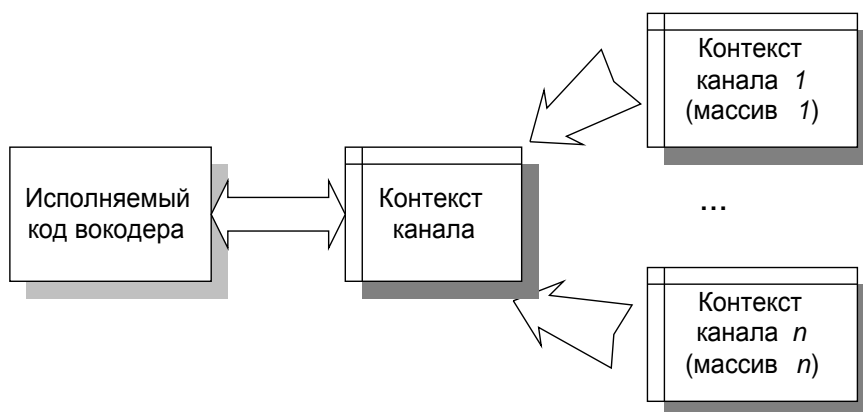


Рис. 3. Метод копирования контекстов

После вызова кода для первого канала внешние массивы обновляются и загружаются данные следующего канала. Метод достаточно прост, но его недостаток в том, что растут накладные расходы в виде дополнительного времени на перенос массивов и выделения дополнительной памяти под внешние массивы.

#### 4.4 Метод переключения дескрипторов.

Самый сложный в реализации метод (рис. 4), но он позволяет рационально использовать ресурсы времени и памяти. Контексты данных хранятся в разных массивах, но перед вызовом кодека копируются не сами контексты, а только таблицы ссылок на них.

Существует специальный массив, который называется глобальным дескриптором. Кодек использует хранящиеся в нем ссылки для доступа к памяти, выделенной под соответствующий канал контекст. Перед вызовом соответствующего канала в него копируются только локальные дескрипторы, содержащие таблицы ссылок соответствующего контекста, а для доступа к данным используется косвенная адресация.

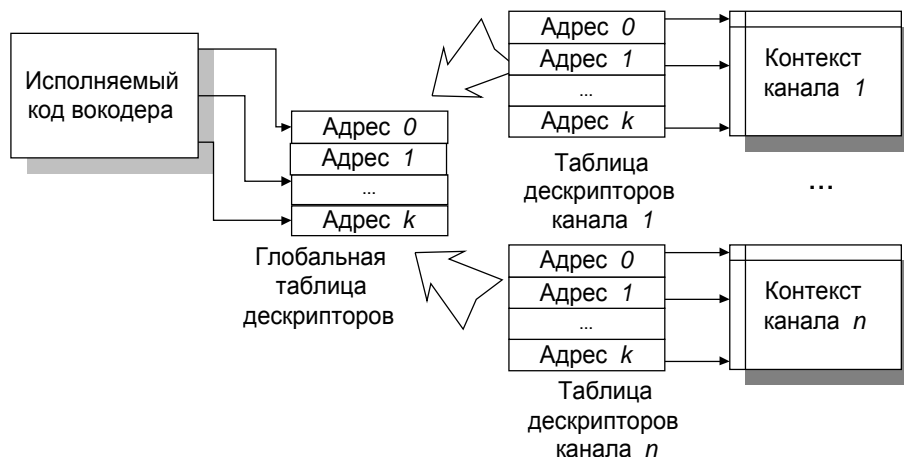


Рис.4. Метод переключения дескрипторов

## 5. Полученные результаты.

Изложенный подход перехода от моделей к реализации на DSP был использован при создании вокодера, описанного в [1]. Была создана целочисленная модель в базовых операциях, реализованная затем на DSP TMS320c54x в 2-канальном варианте [2]. Для реализации многоканальности был использован способ переключения дескрипторов. Тот же способ был применен при реализации многоканальных вариантов стандартов G.729 [4], G.729A, G.729B, G.723.1 на DSP TMS320c54x. Параметры реализаций приведены в таблице 1.

Таблица 1. Параметры реализации алгоритмов на TMS320c548.

Алгоритм	Загрузка DSP (MIPS)	Память программ (К слов)	Память данных (К слов)
CELP 4.6 кбит/с 1 канал:	27	8	16
2 канала:			+1.6
G.729ab 1 канал:	15.5	9.65	8.13
2 канала:			+1.85
G.723.1a 1 канал:	21	8	11
2 канала:			+1
G.729 1 канал:	26	9	6.5
2 канала:			+1.5

## Библиография

1. Babkin V., Ivanov V., Lanne A., Pozdnov I. "Internet Telephony Vocoders", Proc. The Second European DSP Educational and Research Conference 1998 p.83-87.
2. Знамеровский А.Е. Реализация CELP-вокодера с переменной скоростью на сигнальном процессоре TMS320C548. 52-я НТК: тез. докл./ СПб ГУТ, -СПб, 1999.
3. Бабкин В.В. Реализация двухскоростного CELP вокодера на цифровом сигнальном процессоре ADSP-2181. 52-я НТК: тез. докл./ СПб ГУТ, -СПб, 1999.
4. Андреев И.В. Реализация 8 кбит/с CS-ACELP-вокодера на сигнальном процессоре семейства TMS320C54x. 52-я НТК: тез. докл./ СПб ГУТ, -СПб, 1999.
5. ITU-T Recommendation G.729 Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP) (03/96)
6. ITU-T Recommendation G.723.1 Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s (03/96)