

ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА НЕСКОЛЬКИХ СИГНАЛОВ С КВАДРАТУРНО-АМПЛИТУДНОЙ МАНИПУЛЯЦИЕЙ, ПЕРЕДАВАЕМЫХ ПО ТЕЛЕФОННОМУ КАНАЛУ

Кобзев В.Н.

Институт Точной Механики и Вычислительной Техники им. С.А. Лебедева РАН

При работе многоканальных факс-серверов или модемных пулов в серверах доступа самой ресурсоёмкой задачей является преобразование аналогового сигнала, передаваемого по телефонным каналам, в цифровую последовательность. Преобразование осуществляется путём демодуляции КАМ-сигнала с использованием методов цифровой обработки сигналов. Для этих целей достаточно широко применяются процессоры ЦОС.

Нами рассматривался вопрос достижения максимальной производительности при решении задач параллельной обработки нескольких сигналов с КАМ на цифровых сигнальных процессорах семейства TMS320C6x.

В [1], [2] описаны алгоритмы обработки сигналов с квадратурно-амплитудной манипуляцией. Нами был выбран алгоритм демодуляции, который включает следующий набор операций над отсчётами мгновенных значений сигналов, взятыми с частотой дискретизации 8кГц:

- преобразование Гильберта;

- передискретизация (формирование интерполированных отсчётов) на частоте $2f_s$, с которой работает генератор тактовой частоты, частота тактового генератора в 2 раза выше символьной скорости;

- подстройка частоты $2f_s$ «слепым методом»;

- коррекция МСИ пропуском компонент аналитического сигнала через адаптивный цифровой фильтр;

- комплексное умножение на фазовый множитель $e^{-j2\pi f_0}$, где f_0 - частота генератора несущей частоты;

- декодирование символа, коррекция сигналом ошибки генератора несущей частоты и адаптивного фильтра-корректора.

Анализ структуры алгоритма показывает, что для каждого поступающего отсчёта набор операций не является постоянным, а зависит от фазы внутреннего генератора тактовой частоты. Например, если после приёма символа фаза генератора тактовой частоты не пересекла нулевое значение, то операции декодирования символа не имеют смысла. Такая структура алгоритма не позволяет оптимально использовать вычислительную мощность процессора и режим последовательной обработки отсчётов является невыгодным.

В целях повышения производительности обработки по данному алгоритму было принято решение производить обработку данных в пакетном режиме.

В этом случае входной информационной единицей является не один отсчёт, а целый набор отсчётов. В результате работа алгоритма разбивается на последовательные этапы:

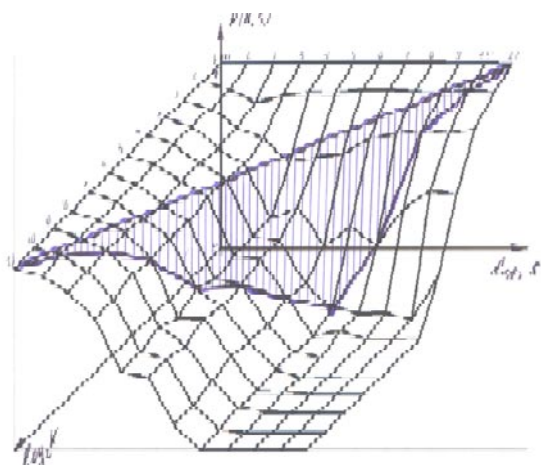
1. Сначала производится преобразование Гильберта для набора отсчётов и формируется аналитический сигнал – набор комплексных значений. Полученные отсчёты аналитического сигнала загружаются в буфер.
2. Передискретизация компонент аналитического сигнала на двойную тактовую частоту и подстройка генератора частоты $2f_s$. Передискретизация осуществляется путём интерполяции по 8 отсчётам. Полученные интерполированные значения загружаются в буфер.
3. Коррекция МСИ путём пропуска компонент интерполированных отсчётов через цифровой трансверсальный фильтр (т.н. дробный эквалайзер). Результаты помещаются в буфер.
4. Формирование выходных символов – комплексный смеситель, решающее устройство, декодер для символов. Полученные символы склеиваются в выходной кадр.

Параллельная обработка нескольких сигналов производится следующим образом – входные отсчёты группируются в сегменты в памяти, после того как какой-либо из сегментов готов к обработке, запускается код с реализацией вышеописанного алгоритма. Кроме описанных выше 4 этапов обработки, код содержит также операции загрузки и сохранения контекста. Операции с контекстом необходимы для того, чтобы обеспечить независимость обработки одного потока данных от других – в этом случае переход от одного пакета к другому происходит прозрачно, как будто обработка велась над непрерывным потоком отсчётов.

При использовании пакетного режима помимо разработки программного кода, максимально использующего ресурсы процессора, очень важную роль играет выбор размера пакета. Использование коротких пакетов (в пределе – переход к режиму последовательной обработки отсчётов) невыгодно, так как велика доля накладных расходов на операции с контекстом. Использование слишком больших пакетов приводит к тому, что требуется большой объём буферной памяти. При этом возникает ситуация, когда объём требуемой памяти превышает ёмкость внутренней памяти данных. В этом случае скорость исполнения программного кода существенно снижается.

Задача определения оптимального размера пакета для обработки тесно связана с вопросом организации буферной памяти. Буферная память используется для накопления входных отсчётов и содержит K сегментов по S отсчётов в каждом. Сегменты динамически распределяются между каналами, при этом устанавливаются теги, указывающие на принадлежность к тому или иному каналу. Тег необходим для правильной загрузки контекста при передаче сегмента в обработку.

Для определения оптимального размера пакета было применено моделирование. В модели учитывался характер поступающих данных (N каналов типа включено/выключено), характер зависимости скорости обработки от размера пакетов и накладные расходы на операции с контекстом и управление записью отсчётов в сегменты буферной памяти. Оптимизация проходила по критерию достижения при фиксированном объёме памяти минимальной вероятности переполнения буферов.



На рисунке приведены результаты моделирования для $N=30$ каналов с аппроксимацией каждого канала в виде источника сигнала типа включено/выключено, интенсивность описывается пуассоновским распределением, средняя длительность распределена по экспоненциальному закону. По горизонтальной оси в логарифмическом масштабе отложены величины K (число сегментов) и S (размер сегмента). Вертикальная ось отражает вероятность переполнения буферной памяти и потерь данных. Сечение поверхности $P(K, S)$ плоскостью $\log_2 K + \log_2 S = const$ позволяет определить оптимальную организацию буфера для имеющегося объёма памяти.

Таким образом, в процессе работы над реализацией алгоритмов обработки КАМ-сигналов были получены следующие результаты:

- применён пакетный режим обработки, что дало увеличение производительности приблизительно на 30 % по сравнению со стандартным режимом последовательной обработки отсчётов,

- разработаны средства для увеличения скорости обработки путём оптимального распределения буферной памяти.

В дальнейшем планируется применить описанные в [3] способы оптимизации программного кода и в итоге добиться производительности, достаточной для обработки 40 сигналов одновременно (или до 20 сигналов с дополнительным декодированием свёрточного кода по алгоритму мягких решений Витерби).

Литература

1. V.34 Transmitter and Receiver Implementation on the TMS320C50 DSP. Digital Signal Processing Solutions, 1997.
2. Tim Massey, Ramesh Iyer. DSP Solutions for Telephony and Data/Facsimile Modems. Digital Signal Processing Solutions, 1997.
3. Ghassan Farah. Optimizing modems using Code Composer Studio and TI resources. Embedded Edge, June 2001.
4. Прокис Дж. Цифровая связь. Радио и Связь. 2000.

PARALLEL PROCESSING OF SEVERAL QAM SIGNALS TRANSMITTED THROUGH TELEPHONE LINE

Kobzev V.

The Institute of Precise Mechanics and Calculation Equipment

The efficient way of QAM demodulation is to use DSP processors. The performance of modern DSP processors is enough for processing up to 30 QAM-modulated signals with symbol freq 2400Hz and sample freq 8000Hz.

The one method of performance increasing on task of QAM-demodulation is packet processing. The algorithm we use consist of 3 studies:

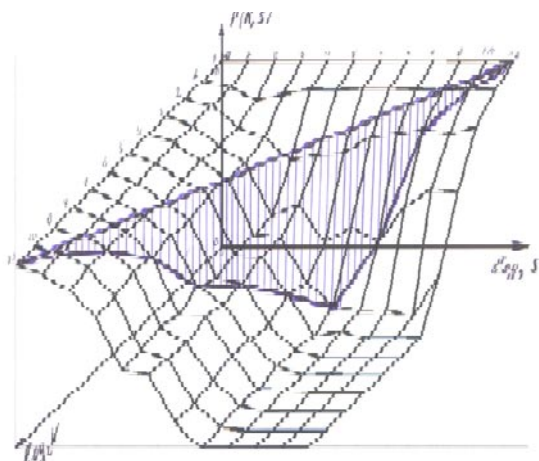
- Hilbert transformation and quadratures oversampling, results are written into buffer;
- Data taken from buffer and filtering by the adaptive equalizer, and data corresponding every tact moment written into buffer;
- Symbol decision and correction of carrier clock generator errors and adaptive equalizer coefficients correction.

Because of the several signals are processing simultaneously, the code must be reentrant and the processor time must be distributed between this signals.

Using the packet method of processing allow do this. Indeed, if we are have two memory regions – one to store of the context and another to buffering the input samples, we may to process packets of samples from input samples buffer (ISB) , pointing to memory region with stored context. The benefit of using packet method is dramatically decrease of context store/load operations and in program structure optimization. The second reason is especially important if using DSP processors with VLIW and pipeline.

The large amount of memory needed for ISB with packet method of processing used. The organization of ISB is K segments of size S. Every segment has a tag with signal identifier (SID). When the amount of samples in segment is equal to S, this packet of samples will be processed and so on.

We are simulated the process of processing with the aim of definition the optimal value of S. The results are shown on figure. The horizontal axis correspond to logarithm K and logarithm S, and vertical axis correspond to value of buffer overflow probability.

**Refs**

1. V.34 Transmitter and Receiver Implementation on the TMS320C50 DSP. Digital Signal Processing Solutions, 1997.
2. Ghassan Farah. Optimizing modems using Code Composer Studio and TI resources. Embedded Edge, june 2001.