

# Программирование сигнальных процессоров компании Analog Devices в среде VisualDSP++ Спектральный анализ

**Юрий Юрченко,  
Михаил Петров,  
Вадим Шатилов**

vadim@eltech.spb.ru

Одной из интересных и наиболее часто востребованных задач цифровой обработки сигналов является спектральный анализ на основе дискретного преобразования Фурье (ДПФ), который находит самое широкое применение в анализаторах спектра, устройствах обработки речи, изображениях, сжатия информации и системах распознавания. ДПФ является частью программ проектирования фильтров для расчета импульсной характеристики фильтра по его частотной характеристике и обратно. Отдельной и достаточно обширной областью применения являются задачи распознавания характеристик объектов по их спектральным портретам. Это используется, в частности, при обработке радиолокационных сигналов. Однако в рамках данной статьи мы ограничимся примерами, работающими с сигналами звукового диапазона, в частности, рассмотрим пример реализации устройства спектрального анализа с использованием отладочного комплекта ADSP2181-EZ-KIT-LITE и отображением полученного спектра в среде VisualDSP++. Этому вопросу посвящено одно из последних занятий, проводимых преподавателями центра ЦСП при ЛЭТИ по программе подготовки специалистов по работе с сигнальными процессорами компании Analog Devices.

## Особенности алгоритма БПФ

Для расчета спектра сигнала по конечной выборке из  $N$  отсчетов используется алгоритм дискретного преобразования Фурье (ДПФ), который преобразует  $N$  комплексных отсчетов сигнала во временной области в  $N$  комплексных отсчетов спектра в частотной области. Если частота входного сигнала равна  $f$ , то в частотной области отображается спектр сигнала в пределах  $(-f/2, f/2)$ . При исследовании звуковых или видеосигналов формируются только действительные отсчеты, а мнимые приравниваются к 0. В этом случае действительная часть спектра обладает четной симметрией, а мнимая — нечетной, и при вычислении модуля частотной характеристики можно ограничиться областью частот  $(0, f/2)$ .

При обработке радиосигналов формируются квадратурные составляющие сигнала ( $I$  — синфазная,  $Q$  — квадратурная составляющие), образующие комплексный сигнал. Тогда  $I$  используется в качестве

действительной, а  $Q$  — мнимой части отсчета. Спектр сигнала в области положительных и отрицательных частот получается различным.

Ниже приведено основное уравнение для  $N$ -точечного ДПФ.

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N} = \frac{1}{N} \sum_{n=0}^{N-1} x(n) [\cos(2\pi nk/N) - j \sin(2\pi nk/N)]$$

где:  $X(k)$  — является  $k$  отсчетом спектра ДПФ;  $x(n)$  является  $n$  отсчетом во временной области;  $N$  представляет собой число отсчетов при вычислении ДПФ.

Быстрое преобразование Фурье (БПФ) является на самом деле одним из алгоритмов вычисления ДПФ за счет сокращения количества операций умножения и сложения. Вычисления при использовании БПФ сильно упрощаются за счет использования свойств симметрии и периодичности коэффициентов, представляющих собой базовые гармонические функции. При вычислении ДПФ требуется  $N^2$  операций умножения комплексных чисел, при использовании БПФ количество операций сокращается до  $(N/2)\log_2(N)$ . Таким образом, эффективность БПФ по сравнению с ДПФ становится весьма существенной, когда количество точек увеличивается до нескольких тысяч. Другой особенностью БПФ является то, что вычисляются все компоненты выходного спектра, откуда следует, что, если необходимо рассчитать только несколько точек спектра, ДПФ может оказаться эффективнее. В таблице 1 приведено соотношение объема вычислительных затрат при различном объеме входных данных.

Таблица 1

N точек	Количество операций умножения		Эффект БПФ
	ДПФ	БПФ	
256	65536	1024	64:1
512	262144	2304	114:1
1024	1048576	5120	205:1
2048	4194304	11264	372:1
4096	16772216	24576	683:1

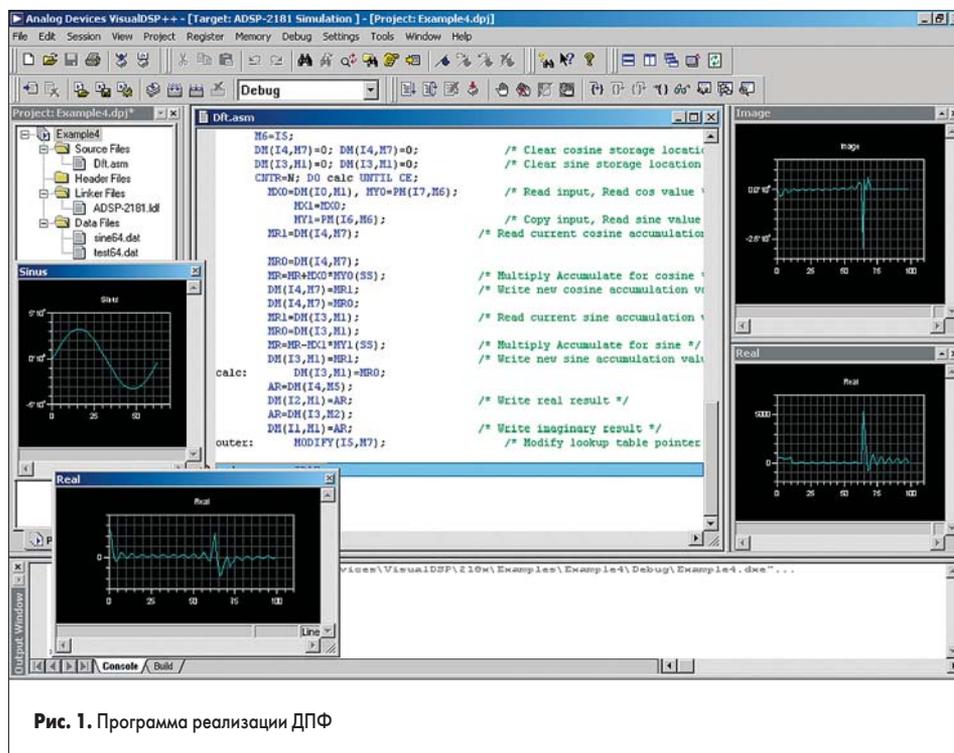


Рис. 1. Программа реализации ДПФ

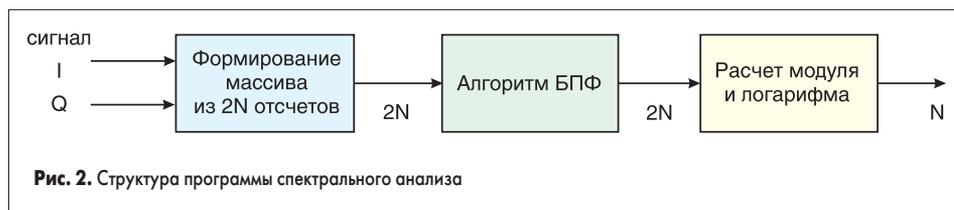


Рис. 2. Структура программы спектрального анализа

Для вычисления БПФ используют несколько различных подходов. Алгоритм по основанию 2 (Radix 2) разделяет полное вычисление ДПФ на комбинацию двухточечных ДПФ. Каждое двухточечное ДПФ использует базовую операцию умножения с накоплением (так называемую «бабочку»). При этом число точек в БПФ должно быть степенью числа 2, если количество точек является степенью числа 4, то можно использовать алгоритм по основанию 4 (Radix 4).

Скорость работы алгоритма БПФ по основанию 4 при условии максимального использования дополнительных возможностей процессоров Analog Devices может быть более чем в два раза выше скорости работы алгоритма Radix 2. Это возможно, поскольку сигнальные процессоры компании Analog Devices содержат дополнительные внутренние функции, такие как например, наличие двух генераторов адреса с возможностью бит-реверсивной адресации, и других аппаратных возможностей, широко используемых для оптимизации вычислений.

Более подробные разъяснения о структуре различных алгоритмов вычисления БПФ можно найти в книге «Теория и применение ЦОС» (Рабинер Л., Голд Б. М. «Мир». 1978).

**Примеры реализации программ для вычисления спектра сигнала**

Рассмотрим подробнее программы для вычисления спектра сигнала с использованием процессоров серии ADSP-218x. После установки стандартного пакета VisualDSP++ с ис-

пользованием тестовой 30-дневной лицензии, которая дает возможность использования симулятора, в каталоге C:\Program Files\Analog Devices\VisualDSP\218x\Examples\Example4 можно найти пример программы, реализующей дискретное преобразование Фурье (ДПФ) по вышеприведенной формуле. На рис. 1 приведен внешний вид среды разработки VisualDSP++ в процессе отладки указанного примера.

В этом примере показана прямая реализация дискретного преобразования Фурье без использования дополнительных возможностей процессора по оптимизации. На практике используются более сложные програм-

мы, например расчет БПФ по основанию 4. В рамках практических занятий со специалистами преподаватели центра ЦСП реализовали подобную программу, которую мы рассмотрим подробнее.

Программа спектрального анализатора состоит из нескольких подпрограмм (рис. 2). Подпрограмма формирования входных данных записывает массив из 2N значений, соответствующих N комплексным отсчетам сигнала. Если входной сигнал действительный, то мнимые значения обнуляются. Затем 2N входных значений во временной области преобразуются алгоритмом БПФ в массив из 2N значений спектра. В примере программы используется БПФ на 256 точек и, соответственно, массивы имеют размер 512. (В процессорах ADSP218X 256-точечное преобразование выполняется за 733 цикла процессора.) Далее из 2N отсчетов спектра вычисляется N значений модуля спектра или логарифма модуля. Если сигнал действительный, то вычисляются только N/2 значений модуля, так как отрицательная часть спектра модуля повторяет положительную.

Для наблюдения спектра в реальном масштабе времени необходимо добавить дополнительную подпрограмму вывода спектра на осциллограф через цифро-аналоговый преобразователь (ЦАП). В среде программирования VisualDSP++ наблюдение спектра можно выполнить с помощью встроенных графических средств. Для этого необходимо войти в меню View->Debug Windows->Plot->New. Появится окно задания конфигурации графика. В нем задается область памяти, в которой хранятся данные, адрес массива и его размер. Дополнительные настройки графика (шрифт, тип линий и т. п.) задаются при нажатии на кнопку «Setting». На рис. 3 показан входной сигнал.

Если БПФ используется в составе анализатора спектра, то для улучшения спектрального разрешения при формировании массива входных данных используется функция окна. При умножении входных данных на функцию окна уровни боковых лепестков в спектре уменьшаются. На рис. 4 представлен внешний вид спектра сигнала при использовании

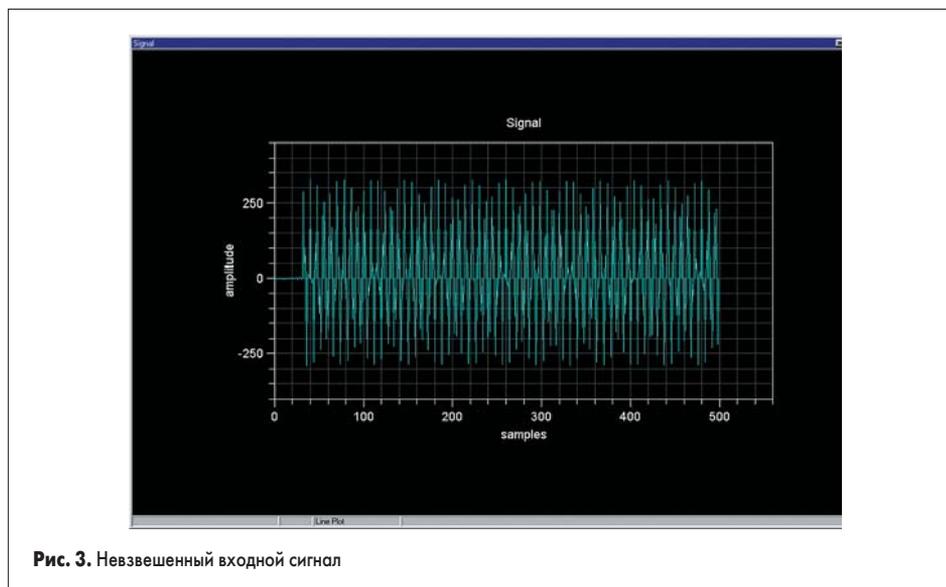


Рис. 3. Невзвешенный входной сигнал

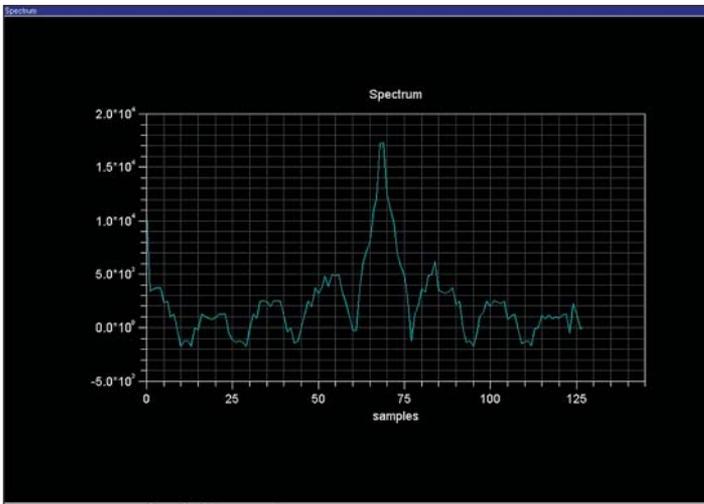


Рис. 4. «Грязный» спектр

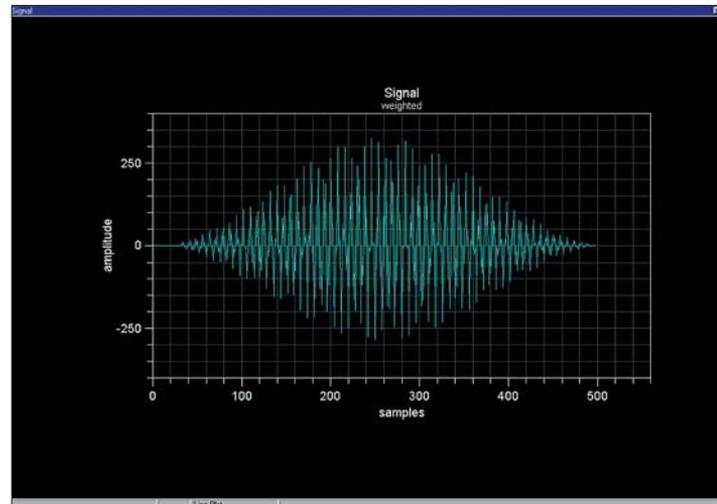


Рис. 5. Обработанный входной сигнал

невзвешенного входного сигнала. На графике наглядно видно, что уровни боковых лепестков составляют до 30% уровня основного, что для задач спектрального распознавания недопустимо много.

Выбор функции окна определяет достижимый уровень боковых лепестков спектра. Наиболее распространенные окна Хэннинга, Блэкмана и Хэмминга имеют уровни наибольших лепестков спектра -32, -43 и -58 дБ соответственно. Понижение уровня боковых лепестков связано с расширени-

ем основных линий спектра. Перечисленные функции имеют ширину на уровне -3 дБ 1,44, 1,3 и 1,68 соответственно. На рис. 5 показан сигнал, обработанный оконной функцией. Умножение на функцию окна выполняется в подпрограмме обработки прерывания при формировании массива входного сигнала.

Рассмотрим фрагменты программы, реализующей БПФ. При ее создании были максимально полно задействованы такие аппаратные возможности процессора, как:

- использование бит-реверсивной адресации;
- использование при вычислениях «теневых» регистров;
- аппаратная поддержка реализации циклических буферов.

В программе вычисления спектра формируются 3 циклических буфера, которые необходимы для подпрограммы БПФ. Буферы inplace и output длиной 512 содержат входные и выходные данные. Кроме того, имеется таблица косинусов cos\_table.

При запуске задача выполняет ввод данных функцией collect\_data, алгоритм БПФ, а затем вычисление модуля спектра, после чего останавливается. Для наблюдения массива входных данных необходимо выполнить остановку перед БПФ и вывести график массива входных данных. При работе БПФ этот массив используется для промежуточных вычислений и разрушается. Результат вычисления модуля наблюдается в массиве results.

### Подпрограмма ввода данных

В подпрограмме ввода данных организуется цикл ожидания прерываний приема последовательного порта и контролируется изменение адреса буфера входных данных.

В начале подпрограммы производится сдвиг адреса входного буфера inplace на 2 и выполняется команда ожидания idle. После обработки прерывания принятый сигнал записывается во входной буфер и происходит модификация адреса. После наполнения буфера выполняется переход в начальный адрес и остановка ввода данных путем маскирования прерывания приема последовательного порта.

```

/*
    ADSP2181 EZ-LAB FFT
    В программе содержится 256-точечный алгоритм БПФ radix-4,
    который размещается во внутренней памяти ADSP-2181. Следующие
    вторичные регистры используются в алгоритме БПФ:

    mx1'-----> M3_space
    my1'-----> bfy_count */

#define    N 256
#define    Nx2 512

.GLOBAL sub;
.GLOBAL in_out;
.SECTION/PM pm_da; VAR/CIRC cos_table[N]=«cos256.dat»;
/*таблица косинусов*/

.SECTION/DM buf_var2;
.VAR/circ output[Nx2];
.SECTION/DM buf_var3;
.VAR/circ inplace[Nx2]; /*буфер входных данных БПФ*/
.SECTION/PM seg_code;

sub:
    i0 = inplace; m0 = 1; l0 = LENGTH(inplace);
    toggle fl1;
    imask=b#0000100001; /*разрешение прерывания
    приема порта и таймера*/

    ena timer;
/*очистка входного буфера inplace*/
    ax0=0;
    cntr = N;
    do zloop until ce;
    zloop:
    dm(i0,m0) = ax0;

    call collect_data; /* вызов подпрограммы
    формирования входного массива
    call fft; /* подпрограмма БПФ
    call magnitude_squared; /* Расчет модуля и логарифма

stop:
    idle; /*остановка решения*/
    jump stop
    
```

Листинг 1. Основная программа вычисления спектра сигнала

```

/*
    Подпрограмма ввода данных Collect Data
*/

.SECTION/PM pm_da;
.var/circ window[N]=«hanning.dat»; /*функция окна Хэннинга*/
.SECTION/PM seg_code;
collect_data:
    i7=window;m7=1;l7=LENGTH(window);
    i0=inplace;
    modify(i0,m0);
    modify(i0,m0); /*сдвиг адреса i0 на 2*/
still_collecting:
    ax1=inplace; /*запись адреса начала буфера inplace*/
    ay1=i0; /*запись текущего значения адреса*/
    idle; /*ожидание прерывания приема порта*/
    ar=ax1-ay1; /*проверка возвращения адреса в начало буфера*/
    if ne jump still_collecting;

imask=b#0000000001; /*маскирование прерывания порта*/
    rts;
    
```

Листинг 2. Подпрограмма ввода данных Collect Data

### Подпрограмма обработки прерывания приема

Подпрограмма обработки прерывания приема запускается после приема информации, поступающей через последовательный порт из кодека. Отсчет сигнала извлекается из ячейки приемного буфера, соответствующей левому каналу. Для контроля этот сигнал отправляется обратно на передачу. Далее принятый отсчет умножается на значение функции Хэннинга, масштабируется путем сдвига и записывается в буфер входных данных. Вместо мнимой части отсчета в буфер записывается 0.

```

/* Обработка прерывания приема SPORTa */
in_out:
    mx0=DM(rx_buf+1); /*считывание значения из буфера приема*/
    dm(tx_buf+1)=mx0; /*передача значения в буфер передачи для
    контроля*/

    ay0=0;
    my0=pm(l7,m7); /*считывание значения функции Хэннинга*/
    mr=mx0*my0(ss); /*умножение на функцию окна*/
    sr= ashif mr 1 by -3 (hi); /*масштабирование*/
    dm(i0,m0)=sr1; /*запись действительной части в буфер inplace*/
    dm(i0,m0)=ay0; /*запись нуля на место мнимой части отсчета*/
    rts;
    
```

Листинг 3. Обработка прерывания приема SPORTa



Рис. 6. Поэтапное вычисление БПФ с прореживанием по времени по основанию 4

### Подпрограмма вычисления быстрого преобразования Фурье (Radix 4)

В рамках статьи сложно и нецелесообразно приводить листинг целиком, поскольку он имеет достаточно большой объем. Поэтому будут приведены основные, наиболее ключевые моменты подпрограммы. В полном объеме эту программу можно найти на сервере компании Analog Devices или в центре ЦСП при ЛЭТИ. Процесс преобразования проходит несколько стадий. На рис. 6 показан поэтапный процесс обработки.

```
stage1:
// настройка генераторов адреса для адресации массивов данных
I0=inplace; /*in ->Xa,Xc*/
I1=inplace+Nov2; /*in+N/2 ->Xb,Xd*/
I2=inplace+1; /*in+1 ->Ya,Yc*/
I3=inplace+Nov2+1; /*in+N/2+1 ->Yb,Yd*/
I5=cos_table; I6=cos_table; I7=cos_table;

/* Настройка модификаторов указателей для выборки данных */
M0=N; /*N,*
M1=-N; /*-N,*
..... пропущено.....
M6=Nov4+2; /*N/4 + stage*2, Cc Sc смещение */
M7=Nov4+3; /*N/4 + stage*3, Cd Sc смещение*/

AX0=DM(I0,M0); /*Первое значение Xa*/
AY0=DM(I0,M1); /*Первое значение Xc*/
AR=AX0-AY0, AX1=DM(I2,M0); /*Xa-Xc, Первое значение Ya*/
SR=LSHIFT AR(LO), AY1=DM(I2,M1); /*SR1=Xa-Xc,
Первое значение Yc*/
CNTR=Nov4; /*Bfly*/
```

Листинг 4. Стадия первая: подготовка

Как можно заметить, основная особенность связана с настройками регистров модификаторов в генераторах адреса процессора, их использование позволяет осуществлять выборку данных из входного массива в необходимой последовательности.

Далее начинается этап так называемой «бабочки» Radix 4. Каких-либо особенностей здесь нет, достаточно подробно алгоритм описан в соответствующей литературе. Можно лишь отметить использование многофункциональных команд процессора для сложения двух операндов и выборки следующего, а также использование косвенной адресации с модификатором для обращения к массиву данных.

```
stg1bfy:
AR=AX0+AY0, AX0=DM(I1,M0); /*AR=xa+xc, AX0=xb */
MR0=AR, AR=AX1+AY1; /*MR0=xa+xc, AR=ya+yc */
.....
// далее около 80 строк кода программы
.....
MR=MR-MX0*MY1(RND), AY1=DM(I2,M1); /*MR=(ya+yb-yc-
yd)(Cd)-(xa-xc-
yb+yd)Sd*/
/* y'd=(ya+xb-yc-xd)
Cd-(xa-xc-
yb+yd)Sd*/

DM(I3,M2)=MR1, SR=LSHIFT AR(LO);

if not ce jump stg1bfy; // завершение.
```

Листинг 5. Вычисление «бабочки» Radix 4

После завершения всех операций вычисления «бабочек» начинается стадия объединения полученных промежуточных результатов. На этом этапе проводится повторная настройка указателей генераторов адреса, кроме того, на этом этапе используются так называемые «теневые» регистры (это вторая копия всех основных вычислительных регистров процессора, на которые можно переключиться за один такт процессора специальной командой). Кроме того, при объединении результатов широко используются возможности бит-реверсивной адресации.

```
Stage_end:
// обновление указателей
I0=inplace; /*in -> Xa,Xc*/
I1=inplace+32; /*in+N/8 -> Xb,Xd*/
I2=inplace+1; /*in+1 -> Ya,Yc*/
I3=inplace+33; /*in+N/8+1 -> Yb,Yd*/
M0=Nov4; /*N/4, обновление модификаторов*/
M1=-Nov4; /*-N/4, обновление модификаторов */
M2=-Nov4+2; /*-N/4+2, обновление модификаторов */

M3=96; /*N*3/8, обновление модификаторов */

ena sec_reg; /* переключение на теневые регистры */
mx1=M3;
dis sec_reg; /* переключение обратно на основные */
// далее пропущено

AX0=DM(I4,M4); /*first Xa*/
AY0=DM(I4,M4); /*first Xc*/
CNTR=Nov4; /*groups/stage*/
ENA BIT_REV; /*режим бит-реверсии адреса для данных на
/* которые настроены указатели I0..I3 генераторов адреса*/
// далее пропущено
DM(I1,M0)=AR, AR=SR1-AY1; /* x'd=xa-xc-(yb-yd)*/
/*AR=ya-yc+(xb-xd)*/
DM(I3,M0)=AR; /*y'd=ya-yc-(xb-xd)*/

if not ce jump laststg1bfy;
DIS BIT_REV; /*отключение бит-реверсивной адресации*/
RTS; /*выход из подпрограммы FFT */
```

Листинг 6. Объединение результатов

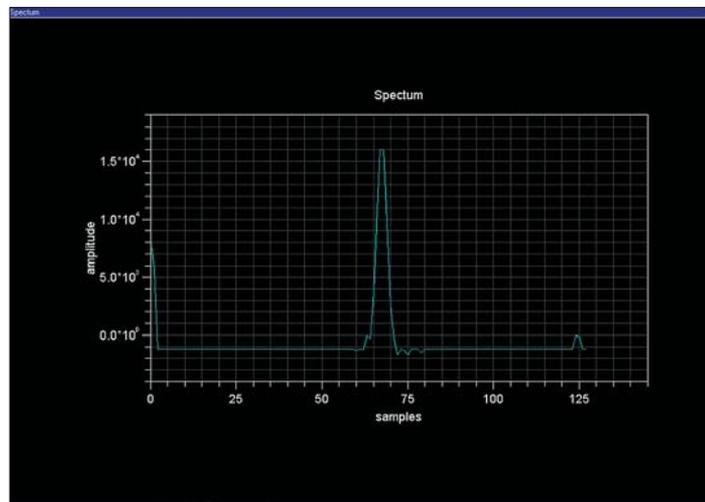


Рис. 7. Спектр обработанного входного сигнала

После окончания работы подпрограммы БПФ на последнем этапе вызывается подпрограмма вычисления модуля.

В подпрограмме используются данные, содержащиеся в буфере output, для вычисления логарифма модуля и размещения его в буфере results объемом 128 значений (только положительная часть спектра). Результат работы программы представлен на рис. 7.

### Заключение

Возможности сигнальных процессоров компании Analog Devices, а также средств разработки VisualDSP++ не исчерпываются приведенными в рамках статей примерами. Непрерывно развиваются как процессоры, так и программные средства. Появление процессора BlackFin, архитектура которого оптимизирована для программ, написанных на языке Си, позволяет существенно сократить время, необходимое для разработки программы. Кроме того, последняя, третья версия VisualDSP++ имеет в своем составе два важных дополнения — это ядро операционной системы реального времени VDK-Kernel и новая разработка компании Analog Devices VSCE — VisualDSP++ Component Software Engineering. Это набор программных компонентов (объектов), созданных с применением концепций, реализованных в COM-объектах фирмы Microsoft. Ключевым элементом этого подхода является наличие функций, называемых методами, посредством которых реализован интерфейс доступа к объекту и которые являются частью самого объекта.

Таким образом, средства разработки сегодня находятся в стадии непрерывного развития их возможностей, и компания Analog Devices в лице технических консультантов фирм-дистрибьюторов, а также специалистов центра ЦСП, делает все, чтобы российские разработчики могли воспользоваться этими возможностями в полном объеме. Этой статьей заканчивается цикл публикаций, посвященных среде разработки VisualDSP++, однако специалисты центра ЦСП при СПбГЭТУ (ЛЭТИ) продолжают свою работу, и, обратившись в центр, вы всегда сможете получить необходимую консультацию или провести переподготовку своих специалистов.