

**ЦИФРОВЫЕ СИСТЕМЫ СБОРА И ПЕРВИЧНОЙ
ОБРАБОТКИ ДАННЫХ И ИХ ЭЛЕМЕНТЫ**

УДК 681.323

В. М. ЕФАНОВ, И. И. КОРШЕВЕР, В. М. ЛОБАСТОВ, Г. Г. МАТУШКИН

(Невинномысск, Новосибирск)

**СТРУКТУРА МНОГОФУНКЦИОНАЛЬНОГО ПРОЦЕССОРА
НА ОСНОВЕ АЛГОРИТМА
БЫСТРОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ**

Появление алгоритма быстрого преобразования Фурье (БПФ) в 1965 году [1] привлекло внимание большого количества исследователей, связанных с задачами, которые могут быть эффективно решены с использованием преобразования Фурье. Необходимость ускорения решения таких задач потребовала создания специализированных вычислителей на основе БПФ, способных почти на два порядка уменьшить время вычисления по сравнению с решением этих же задач на универсальных ЭВМ с той же элементной базой.

В литературе упомянут целый ряд разработок многофункциональных специализированных процессоров БПФ (см., например, [2]), но известна лишь одна работа, в которой описана структура такого вычислителя [3]. Однако описанный в этой статье процессор не может служить в качестве автономного прибора для оперативного анализа сигналов при проведении научных экспериментов, поскольку он, во-первых, может работать только совместно с ЭВМ IBM 1800 и не имеет независимого управления, а во-вторых, не имеет каналов непосредственной связи с экспериментом. Кроме того, организация процесса вычислений при осуществлении БПФ, принятая в этом процессоре, требует увеличения на 50% объема памяти по сравнению с необходимым, которое во многих случаях не оправдывается некоторым упрощением индексного устройства.

В настоящей статье рассмотрен один из вариантов организации многофункционального специализированного процессора со специфичным для повышения быстродействия расчленением оперативного запоминающего устройства (ОЗУ), который, кроме осуществления непосредственного прямого и обратного преобразований Фурье, позволяет выполнять целый ряд операций по обработке сигналов.

Известно несколько видов структурных схем, реализующих алгоритм БПФ [4] и отличающихся степенью параллелизма при выполнении вычислений. Дело в том, что специфика алгоритма БПФ позволяет вести вычисления параллельно как внутри одной итерации (ступени вычисления), так и по всем итерациям. Рассматривая эти схемы, можно сделать вывод, что при существующем уровне развития элементной базы наиболее целесообразно (по крайней мере, с точки зрения аппаратной сложности) строить процессор по последовательной схеме, когда при помощи одного арифметического устройства (АУ) производится последовательное выполнение всех операций алгоритма.

При построении процессоров БПФ могут быть использованы модификации алгоритма БПФ для различных оснований [5, 6], т. е. когда число анализируемых выборок сигнала может быть представлено как $N=r^m$. Практически наиболее удобны алгоритмы по основаниям $r=2^p$. Увеличение основания приводит к некоторому выигрышу в числе требуемых операций (так, применение основания 8 сокращает число действительных умножений примерно на 30%, число действительных сложений — на 10%), однако это ведет к существенному усложнению АУ процессора, поэтому наиболее часто применяемым является алгоритм по основанию 2, который и был принят при реализации описываемого процессора.

Другой класс разновидностей алгоритмов БПФ учитывает специфику обрабатываемых массивов данных. В [7—9] описаны алгоритмы, использующие тот факт, что обрабатываются действительные массивы данных и алгоритмы, учитывающие симметрию данных. При обработке действительных массивов данных, используя комплексную сопряженность как промежуточных, так и окончательных результатов, возможно сокращение вдвое числа операций и емкости ОЗУ.

Обработка комплексных сигналов в процессоре производится по известному алгоритму [1]:

$$A_p(r_0 \dots r_{p-1} k_{m-p-1} \dots k_0) = A_{p-1}(r_0 \dots r_{p-2} 0 k_{m-p-1} \dots k_0) + \\ + (-1)^{r_{p-1}} A_{p-1}(r_0 \dots r_{p-2} 1 k_{m-p-1} \dots k_0) W_m(\alpha),$$

где

$$W_m(\alpha) = e^{\frac{-j2\pi\alpha}{2^m}}; \quad \alpha = 2^{m-2} k_{m-2} + \dots + 2^0 k_0;$$

r — частотные индексы; k — временные индексы; p — номер итерации в алгоритме БПФ; $m = \log_2 N$; N — число выборок в обрабатываемом массиве данных; $A_0(k_{m-1} \dots k_0) = f(k_{m-1} \dots k_0)$ — временные выборки исходной последовательности; $A_m(r_0 \dots r_{m-1}) = F(r_{m-1} \dots r_0)$ — выборки частоты — результат Фурье-преобразования, двоичные индексы которых инвертированы.

Для обработки действительных массивов используется алгоритм [8], позволяющий сэкономить вдвое по сравнению с комплексным алгоритмом память и время обработки.

$$A_p(r_0 \dots r_{p-1} k_{(m-1)-p-1} \dots k_0) = \{ A_{p-1}(r_0 \dots r_{p-2} 0 k_{(m-1)-p-1} \dots k_0) + \\ + (-1)^{r_{p-1}} A_{p-1}(r_0 \dots r_{p-2} 1 k_{(m-1)-p-1} \dots k_0) W_m(\beta) \}^{(*)r_{p-1}},$$

где $\beta = 2^{m-3} s_{m-3} + \dots + 2^i s_i + \dots + 2^{i+1} s_{i-1} + 2^i$; $s_i = r_i \oplus s_{i-1} = \bigoplus_{t=m-3}^i r_t$;

i — порядковый номер первой старшей значащей единицы частотно-временного индекса $\{r_i k_i\}$; \tilde{r}_i — двоиочно-инвертированный частотный индекс.

$$A_0(k_{m-2} \dots k_0) = f(0 k_{m-2} \dots k_0) + j f(1 k_{m-2} \dots k_0) = f({}^0 k_{m-2} \dots k_0); \\ A_{m-1}(0 \dots 0 1 r_{i+1} \dots r_{m-2}) = \begin{cases} F(0 s_{m-3} \dots s_{i+1} 1 0 \dots 0) & \text{при } r_{m-2} = 0; \\ F(1 \bar{s}_{m-3} \dots \bar{s}_{i+1} 1 0 \dots 0) & \text{при } r_{m-2} = 1; \end{cases} \\ F(0) = A_{m-1}(0 \dots 0)_{\text{Re}} + A_{m-1}(0 \dots 0)_{\text{Im}}; \quad F(N/2) = A_{m-1}(0 \dots 0)_{\text{Re}} - \\ - A_{m-1}(0 \dots 0)_{\text{Im}};$$

(*) r_{p-1} — символ, означающий наличие комплексного сопряжения при $r_{p-1} = 1$, и, кроме того, при $r_0 = \dots = r_{p-2} = 0$ существует перестановка:

$$\left. \begin{aligned} A_p(0 \dots 0 0 k_{m-p-2} \dots k_0) \\ A_p(0 \dots 0 1 k_{m-p-2} \dots k_0) \end{aligned} \right\} \rightarrow \left. \begin{aligned} A_p(0 \dots 0 r_{p-1} 0 k_{m-p-3} \dots k_0) \\ A_p(0 \dots 0 r_{p-1} 1 k_{m-p-3} \dots k_0) \end{aligned} \right\}$$

Кроме этих основных операций, АУ может производить умножение и сложение комплексных чисел, нормирование результатов вычисления (деление на 2^n) и трансляцию чисел с входа на выход.

Устройство формирования весовых коэффициентов (УФВК) вырабатывает тригонометрические коэффициенты, необходимые для преобразования Фурье, значения взвешивающих функций, применяемых для повышения точности вычисления спектральных характеристик [10], участвует в вычислении амплитудного и фазового спектров.

ОЗУ выполняет функцию хранения исходных данных, промежуточных и окончательных результатов вычислений. Оно состоит из двух модулей, что позволяет ускорить вычислительный процесс, так как при этом обеспечивается одновременная запись и считывание пары комплексных чисел, необходимых для каждого такта вычислений.

Индексное устройство (ИУ) вырабатывает коды адресов для обращения к ОЗУ и код для получения в УФВК значений тригонометрических коэффициентов и взвешивающих функций.

Блок пересылок (БП) производит рассылку чисел из выходных регистров АУ в буферные регистры ОЗУ в соответствии с выполняемым алгоритмом.

Устройство сопряжения (УС) осуществляет обмен информацией между процессором, с одной стороны, и аналого-цифровым преобразователем (АЦП) и внешними устройствами — с другой.

АЦП производит квантование входного аналогового сигнала по уровню и времени для ввода в процессор.

Стоящие перед АЦП **фильтры (Ф)** служат для ограничения полосы пропускания входного тракта АЦП в соответствии с частотой выборок и позволяют уменьшить дополнительную погрешность получаемых результатов, появляющуюся вследствие явления маскировки (aliasing) [10].

Входные делители (ВД) выполняют функцию нормирования и позволяют расширить динамический диапазон анализируемых сигналов.

Электроннолучевой индикатор (ЭЛИ) позволяет визуализировать получаемые результаты на экране электроннолучевой трубки. Для удобства восприятия предусмотрено несколько режимов вывода информации на экран индикатора: в линейном и логарифмическом масштабах, в декартовых и полярных координатах (для случая вычисления амплитудного и фазового спектров).

Устройство управления (УУ) осуществляет взаимосвязь отдельных узлов процессора и вырабатывает все необходимые для их функционирования управляющие сигналы. Следует отметить, что для расширения функциональных возможностей процессора при его проектировании предусмотрены два режима работы: а) с заданием алгоритма, начальных и текущих условий вычисления с пульта управления, что позволяет реализовать те функции, которые были перечислены выше, и б) с заданием этих условий с ЭВМ, что позволяет решать с помощью процессора и задачи, отличные от задаваемых с пульта управления.

Рассмотрим как пример организацию функционирования процессора при вычислении свертки двух действительных дискретных последовательностей.

Выполнение любого алгоритма разбивается на три этапа: 1) ввод исходной информации в процессор, 2) вычисление, 3) вывод результатов вычисления на внешние устройства или индикатор процессора. Порядок выполнения алгоритма по этапам и внутри их для случая вычисления свертки двух действительных дискретных последовательностей представлен в табл. 1. Как видно из таблицы, каждый этап разбивается на подпрограммы, которые представляют собой при вычислении одно из функциональных преобразований над результатами вы-

Таблица 1

Номер подпрограммы	Операция	Этап
1	Ввод первой последовательности с умножением (или без) на взвешивающую функцию (задается с пульта управления)	Ввод
2	Ввод второй последовательности с умножением (или без) на взвешивающую функцию (задается с пульта управления)	Ввод
1	Вычисление прямого БПФ (алгоритм обработки действительных сигналов) первой последовательности	Вычисление
2	Вычисление прямого БПФ (алгоритм обработки действительных сигналов) второй последовательности	Вычисление
3	Перемножение между собой результатов первой и второй подпрограмм	Вычисление
4	Вычисление обратного БПФ (алгоритм обработки действительных сигналов) результатов перемножения во время третьей подпрограммы	Вычисление
1	Вывод полученных результатов	Вывод

полнения предыдущей подпрограммы (например, прямое преобразование Фурье, обратное преобразование Фурье, вычисление модуля и фазы по квадратурным составляющим и т. д.), при вводе — занесение первой и второй последовательностей, при выводе — однократный вывод полученных результатов. Очередность и содержание подпрограмм на этапе вычисления для различных алгоритмов сведены в табл. 2.

Перед началом выполнения алгоритма на пульте управления задаются начальные условия (алгоритм вычисления, размерность заносимого массива, вид взвешивающей функции, частота выборок, вид внешнего устройства при вводе и выводе информации и т. д.), которые по команде с пульта управления заносятся затем в регистр начальных условий (РНУ). Код алгоритма из регистра начальных условий и код номера этапа с регистра-счетчика этапов поступают на формирователь кода числа подпрограмм (ФКЧП), который определяет и запоминает количество подпрограмм, участвующих в соответствующих этапах выполнения заданного алгоритма.

Регистр-счетчик подпрограмм (РСП) меняет свое состояние на единицу после выполнения каждой подпрограммы. После выполнения всех подпрограмм, входящих в данный этап, регистр-счетчик выдает сигнал на регистр-счетчик этапов (РСЭ), изменяя его состояние на единицу. Новое состояние регистра-счетчика этапов приводит к формированию нового кода числа подпрограмм, соответствующего следующему этапу выполнения алгоритма.

В соответствии с поступающими кодами номера подпрограммы, номера этапа и кода алгоритма формирователь кода подпрограмм (ФКП) выдает код, соответствующий данной подпрограмме. Группы разрядов кода подпрограммы представляют собой коды микропрограмм, которые из формирователя кода подпрограмм поступают параллельно в устройство местного управления (МУ) отдельных узлов процессора (АУ, ИУ, УФВК, БП) и определяют их функционирование при выполнении заданного алгоритма.

Алгоритмы	Номер и содержа-		
	1	2	3
Прямое БПФ (результат в квадратурах)	$S_p(r) = F[A_p(k)]$		
Обратное БПФ	$A_p(k) = F^{-1}[S_p(r)]$		
Прямое БПФ (результат — модуль и фаза)	$S_p(r) = F[A_p(k)]$	$ S_p(r) \arg[S_p(r)]$	
Спектр свертки двух последовательностей	$S_p(r) = F[A_p(k)]$	$R_p(r) = F[B_p(k)]$	$S_p(r) R_p(r)$
Свертка двух последовательностей	$S_p(r) = F[A_p(k)]$	$R_p(r) = F[B_p(k)]$	$S_p(r) R_p(r)$
Автокорреляционная функция (I вариант)	$S_p(r) = F[A_p(k)]$	$ S_p(r) ^2$	$C(r) = \sum_{p=0}^n S_p(r) ^2$
Автокорреляционная функция (II вариант)	$S_p(r) = F[A_p(k)]$	$ S_p(r) ^2$	$D_p(j) = F^{-1}[S_p(r) ^2]$
Спектр мощности (I вариант)	$S_p(r) = F[A_p(k)]$	$ S_p(r) ^2$	$C(r) = \sum_{p=0}^n S_p(r) ^2$
Спектр мощности (II вариант)	$S_p(r) = F[A_p(k)]$	$ S_p(r) ^2$	$D_p(j) = F^{-1}[S_p(r) ^2]$
Взаимный спектр	$S_p(r) = F[A_p(k)]$	$R_p(r) = F[B_p(k)]$	$S_p(r) R_p^*(r)$
Взаимно-корреляционная функция	$S_p(r) = F[A_p(k)]$	$R_p(r) = F[B_p(k)]$	$S_p(r) B_p^*(r)$

Примечание. $A_p(k)$ — p -я реализация первого сигнала; $B_p(k)$ — p -я реализация второго сигнала; $h(j)$ — взвешивающая функция; оператор F — прямое преобразование Фурье; опера

В случае вычисления статистических характеристик, требующих усредненного результата по ряду дискретных последовательностей, с пульта управления необходимо число усреднений записывается в счетчик числа усреднений. Если в исходном нулевом состоянии счетчик не оказывает влияния на порядок выдачи подпрограмм в процессе выполнения алгоритма вычисления, то при записи в него числа он блокирует выдачу номера подпрограммы усреднений из регистра-счетчика подпрограмм и после выполнения подпрограммы, предшествующей подпрограмме усреднений, возвращает в исходное состояние как регистр-счетчик подпрограмм, так и регистр-счетчик этапов. Одновременно с счетчика числа усреднений считывается единица.

Исходная информация с АЦП или с одного из внешних устройств через устройство сопряжения и коммутатор, открывающий канал связи с выбранным внешним устройством или АЦП, поступает на входные регистры АУ. Оттуда она транслируется в буферные регистры ОЗУ прямо или с умножением на взвешивающую функцию, одновременно ком-

Таблица 2

ние подпрограммы			
4	5	6	7
$F^{-1}[S_p(r) R_p(r)]$			
$\frac{1}{n} C(r)$	$F^{-1}\left[\frac{1}{n} C(r)\right]$		
$\varepsilon(j) = \sum_{p=0}^n D_p(j)$	$\frac{1}{n} \varepsilon(j)$	$H(j) = h_j \left[\frac{1}{n} \varepsilon(j) \right]$	
$\frac{1}{n} C(r)$			
$\varepsilon(j) = \sum_{p=0}^n D_p(j)$	$\frac{1}{n} \varepsilon(j)$	$H(i) = h(j) \left[\frac{1}{n} \varepsilon(j) \right]$	$F[H(j)]$
$G(r) = \sum_{p=0}^n S_p(r) R_p^*(r)$	$\frac{1}{n} G(r)$		
$G(r) = \sum_{p=0}^n S_p(r) R_p^*(r)$	$\frac{1}{n} G(r)$	$F^{-1}\left[\frac{1}{n} G(r)\right]$	

сигнала; $S_p(r)$ — спектр p -й реализации первого сигнала; $R_p(r)$ — спектр p -й реализации второго сигнала; F^{-1} — обратное преобразование Фурье; символ * обозначает комплексную сопряженность.

понуясь блоком пересылок в соответствии с выполняемым алгоритмом. С буферных регистров информация поступает непосредственно в один из секторов ОЗУ, определяемых логическим блоком регистра секторов, состояние которого зависит от кода алгоритма и номера подпрограммы. Под сектором в данном случае понимается группа ячеек ОЗУ, имеющих один (в случае разбиения ОЗУ на два сектора) или два (в случае разбиения ОЗУ на четыре сектора) одинаковых старших разряда адреса. Например, при вычислении свертки первая последовательность заносится в сектор, код старших разрядов адреса которого 00 во время выполнения первой подпрограммы ввода. Во время второй подпрограммы ввода производится ввод второй последовательности в сектор с кодом старших разрядов адреса 10.

Синхронизатор устройства управления производит временную увязку работы всех узлов процессора в пределах подпрограммы, выдавая все необходимые для этого управляющие и синхронизирующие импульсы.

По описанной структуре в Институте автоматки и электрометрии СО АН СССР разработан процессор на интегральных микросхемах серии 155, АУ которого будет затрачивать 2,5 мкс для получения каждого двух результатов на любой итерации при вычислении по алгоритму БПФ. В процессоре используется ОЗУ с временем обращения 7 мкс, поэтому вычисление прямого или обратного преобразований Фурье над массивами объемом 2048 чисел осуществляется за время порядка 100 мс. Естественно, что это время можно существенно уменьшить, применяя более быстродействующую память.

В процессоре используется память емкостью 2048 16-разрядных слов, хотя УФВК и ИУ процессора позволяют работать с массивами объемом до 8192 слов, что при использовании внешней памяти значительно расширяет возможности процессора.

АЦП, стоящий на входе процессора, преобразует аналоговые входные сигналы в 9-разрядный двоичный код (с учетом знака). АУ оперирует 16-разрядными словами при выполнении алгоритмов без накопления и 32-разрядными словами при накоплении сумм в случае вычисления статистических характеристик. Частота выборок сигнала ограничена быстродействием примененной памяти и не превышает 100 кГц, хотя быстродействие входного АЦП 500 тысяч преобразований в секунду.

Количество микросхем серии 155, требующееся для создания подобного процессора, примерно равно 1500 шт.

В заключение следует отметить, что описанный процессор, обладающий подобной многофункциональностью, может найти широкое применение при решении различного рода задач оперативного анализа сигналов.

ЛИТЕРАТУРА

1. J. W. Cooley and J. W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series.— Math. Comput., 1965, v. 19.
2. G. D. Bergland. Fast Fourier Transform Hardware Implementations.— A Survey.— IEEE Trans. on Audio and Electroacoustics, 1969, v. AU-17, № 2.
3. W. Y. Dere, D. J. Sahrison. Berkeley Array Processor.— IEEE Trans. on Computers, 1970, v. C-19, № 5.
4. G. D. Bergland. Fast Fourier Transform hardware Implementations — An Overview.— IEEE Trans. on Audio and Elektroacoustics, 1969, v. AU-17, № 2.
5. R. C. Singleton. An Algorithm for Computing the Mixed Radix Fast Fourier Transform.— IEEE Frans. on Audio and Electroacoustics, 1969, v. AU-17, № 2.
6. G. D. Bergland. A Fast Fourier Transform Algorithm, Using Base 8 Iterations.— Math. Comput., 1968, v. 22.
7. J. W. Cooley, T. A. W. Lewis and P. D. Welch. The Fast Fourier Transform Algorithm Programming Considerations in the Calculation of Sine, Cosine and Laplace Transforms.— J. Sound Vib., 1970, v. 12.
8. G. D. Bergland. A Fast Fourier Transform Algorithm for Real Valued Series.— Commun. ACM, 1968, v. 11.
9. Верне. Быстрое преобразование Фурье для действительных сигналов. Уменьшение необходимой емкости памяти и числа шагов за счет применения нечетного дискретного преобразования Фурье.— ТИИЭР, 1971, т. 59, № 10.
10. Г. Д. Бергланд. Руководство к быстрому преобразованию Фурье.— Зарубежная радиоэлектроника, 1971, № 3.

Поступила в редакцию 23 октября 1972 г.