

- Trans. on Audio and Electroacoustics*, 19, No. 1, 48—58 (1971); есть русский перевод: Тирней Дж., Рэйдер Ч., Голд Б., Цифровой синтезатор частот, *Зарубежная радиоэлектроника*, № 3, 57—73 (1972).
7. Perry J. L., Schafer R. W., Rabiner L. R., A Digital Hardware Realization of a Random Number Generator, *IEEE Trans. on Audio and Electroacoustics*, AU-20, No. 4, 236—240 (Oct. 1972).
  8. Green B. F., Smith J. E., Klem L., Empirical Tests of an Additive Random Number Generator, *J. Assn. Computer Machinery*, 6, No. 4, 527—537 (Oct. 1959).
  9. MacLaren M. D., Marsaglia G., Uniform Random Number Generators, *J. Assn. Computer Machinery*, 12, 83—89 (1965).
  10. Rader C. M., Rabiner L. R., Schafer R. W., A Fast Method of Generating Digital Random Numbers, *Bell. Syst. Tech. J.*, 49, 2303—2310 (Nov. 1970).
  11. Rader C. M., A New Method of Generating Gaussian Random Variables by Computer, Lincoln Laboratory Technical Note, 1969-49, 1969.

## СПЕЦИАЛИЗИРОВАННЫЕ УСТРОЙСТВА ДЛЯ ВЫПОЛНЕНИЯ БПФ

### 10.1. Введение

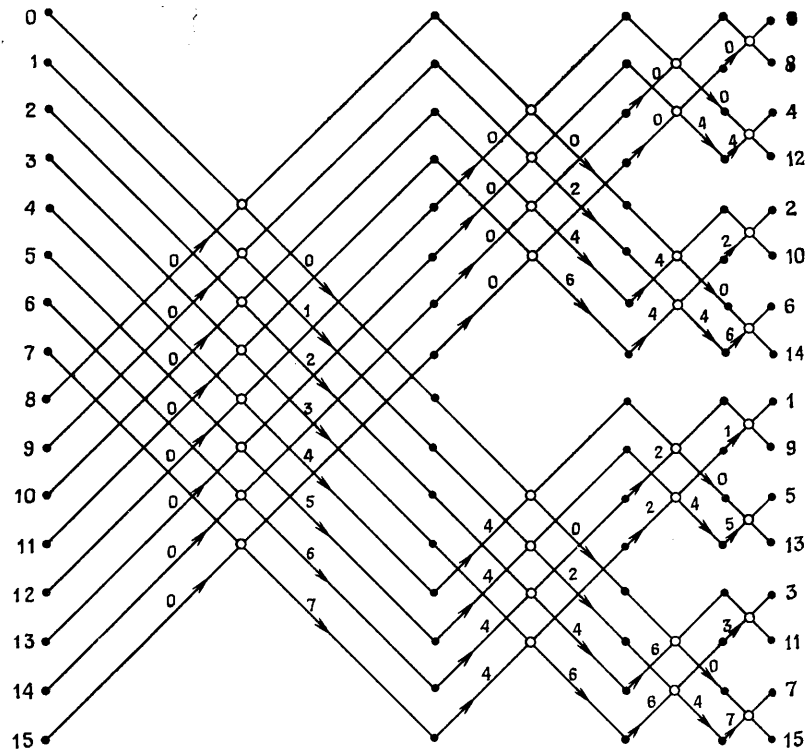
В гл. 6 было приведено весьма подробное описание алгоритмов БПФ как необходимая предпосылка к изложению методов цифрового спектрального анализа. Данная глава посвящена способам практического осуществления алгоритмов БПФ. Чтобы обеспечить возможность выбора наиболее экономичной схемы, в отдельных случаях будет рассмотрено много различных структур БПФ. В связи с этим в начале настоящей главы дан еще более полный обзор алгоритмов БПФ, чем в гл. 6.

В ряде специальных приложений более предпочтительными могут оказаться алгоритмы БПФ с основанием, отличным от 2. Поэтому значительное внимание уделяется, в частности, алгоритмам с основанием 4, позволяющим уменьшить объем оборудования (по сравнению с алгоритмами по основанию 2). Системы с основанием 4 будут рассмотрены довольно подробно. Они будут также сопоставлены с системами, использующими основание 2.

В системах с очень высоким быстродействием (например, в высокоточных радиолокаторах) приходится использовать поточную схему БПФ. В этой главе будут рассмотрены системы такого типа, а также системы, использующие при выполнении БПФ другие виды параллелизма.

### 10.2. Обзор теоретических основ БПФ

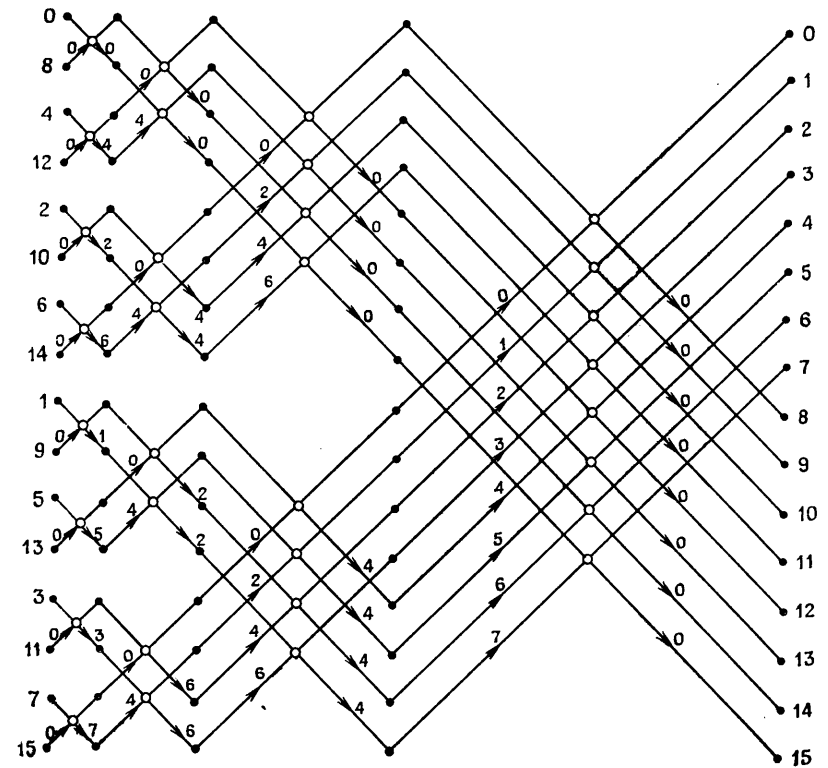
Как уже отмечалось в гл. 6, БПФ можно выполнять по схемам с замещением и без замещения. В первом случае результаты всех ДПФ, которые используются при выполнении БПФ, зачисляются в те же регистры, откуда были взяты исходные числа. Например, в 16-точечном БПФ (фиг. 10.1) при выполнении на первом этапе двухточечного ДПФ, обозначенного верхним незачерненным кружком, используется содержимое нулевого и восьмого регистров. Обозначим эти числа через  $f_0 [= x(0)]$  и  $f_1 [= x(8)]$ . Результатами двухточечного ДПФ являются  $F_0$  и  $F_1$ , причем  $F_0$  заносится в регистр на место  $f_0$ , а  $F_1$  замещает  $f_1$ .



Фиг. 10.1. Алгоритм 16-точечного БПФ с замещением, прямым порядком отсчетов на входе и двоично-инверсным на выходе.

В алгоритме с замещением выходные гармоники всегда оказываются переставленными. Для БПФ с основанием 2 (фиг. 10.1) эта перестановка соответствует двоичной инверсии номеров гармоник. В разд. 10.3 будет рассмотрен характер перестановки гармоник при использовании более высоких оснований.

Как было показано в гл. 6, в алгоритмах БПФ могут использоваться прореживание по времени и прореживание по частоте. При прореживании по времени умножение на поворачивающие множители предшествует выполнению в вершинах графа двухточечных ДПФ, а при прореживании по частоте оно выполняется вслед за ДПФ. На фиг. 10.1 показаны два набора поворачивающих множителей, один из которых используется при прореживании по времени, а другой — по частоте. Стрелки с числами возле них соответствуют умножению на  $W^k$ , где  $k$  — число, записанное около стрелки. Данная структура может быть использована также в случае, когда исходные отсчеты расположены в двоично-инверс-



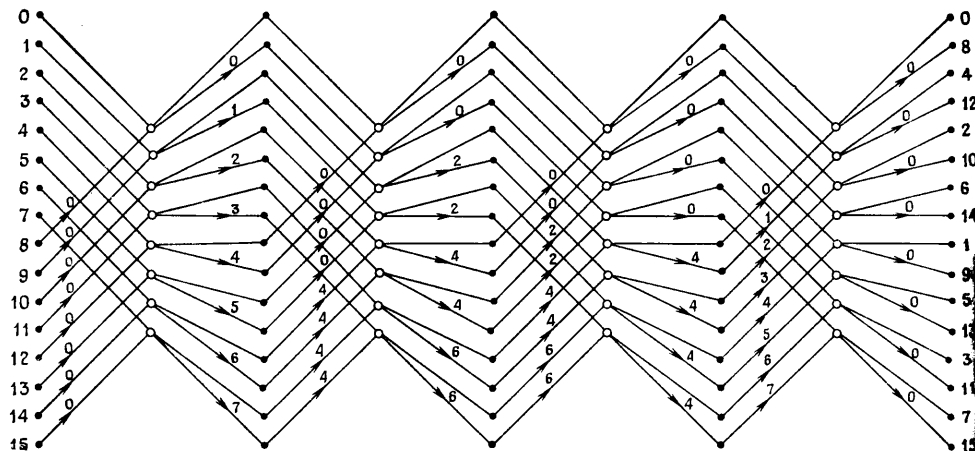
Фиг. 10.2. Алгоритм БПФ с замещением, двоично-инверсным порядком отсчетов на входе и прямым на выходе (для прореживания по времени множители расположены слева от вершин графа, для прореживания по частоте — справа).

ном порядке. При этом выходные гармоники будут размещаться в нормальном порядке.

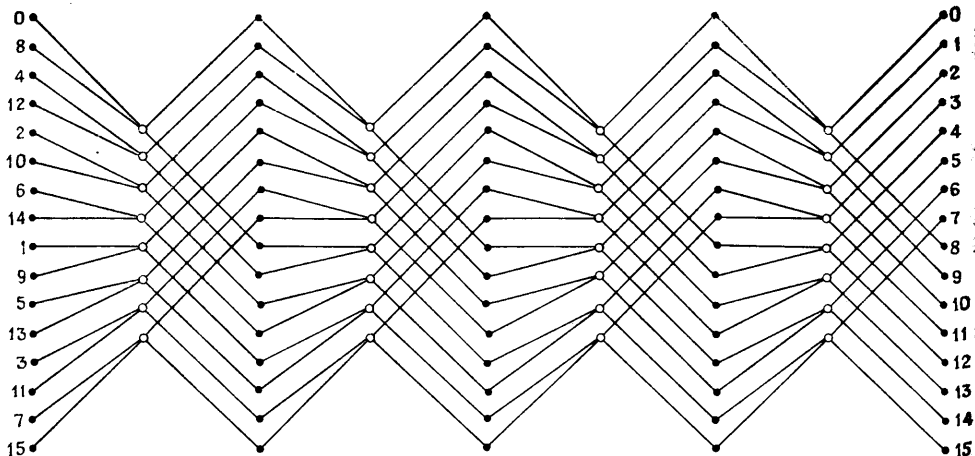
На фиг. 10.2 изображена другая схема алгоритма с теми же свойствами. Здесь также показаны варианты прореживания по частоте и по времени.

Если не будет оговорено особо, то будем считать, что регистры на схемах нумеруются сверху вниз и их номера на схемах не приводятся. Числа, указанные на фиг. 10.1 и 10.2 на входах и выходах, представляют собой номера входных отсчетов и выходных гармоник.

Алгоритм 16-точечного БПФ с основанием 2 и постоянной структурой на всех этапах показан на фиг. 10.3. Здесь результаты базовой операции не возвращаются в те регистры, откуда были взяты исходные числа, поэтому этот алгоритм относится к алгоритмам без замещения. На всех этапах характер нумерации не меняется,

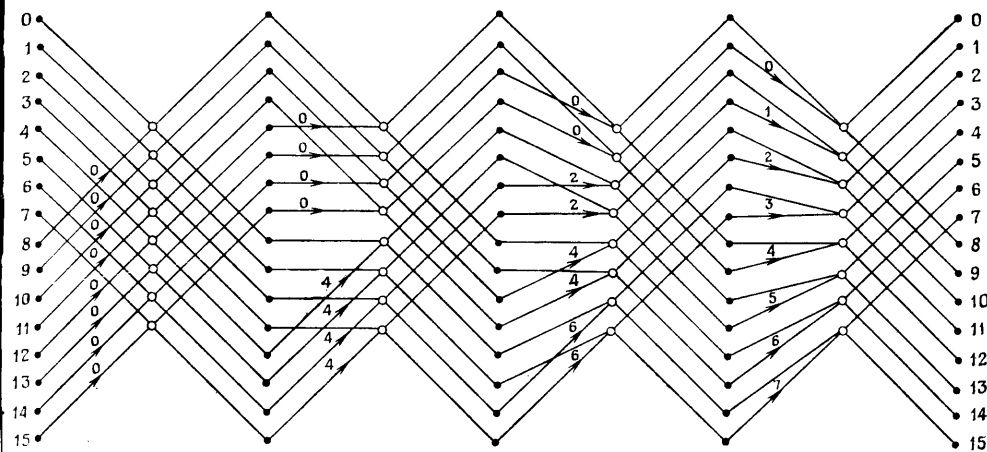


Фиг. 10.3. Алгоритм 16-точечного БПФ по основанию 2 с постоянной структурой, без замещения, с нормальным порядком отсчетов на входе и двоично-инверсным на выходе (показаны множители для прореживания по времени и по частоте).



Фиг. 10.4. Алгоритм 16-точечного БПФ по основанию 2 с постоянной структурой, без замещения, с двоичной инверсией на входе и прямым порядком на выходе.

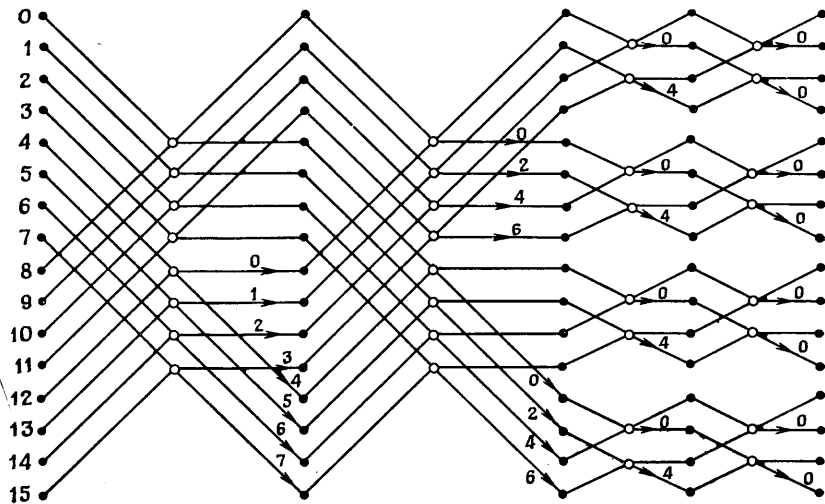
что позволяет в некоторых случаях упростить программы или аппаратуру. Входные отсчеты располагаются в нормальном порядке, а выходные гармоники — в двоично-инверсном. Вообще справедливо следующее правило: при работе с замещением необходимо  $N$



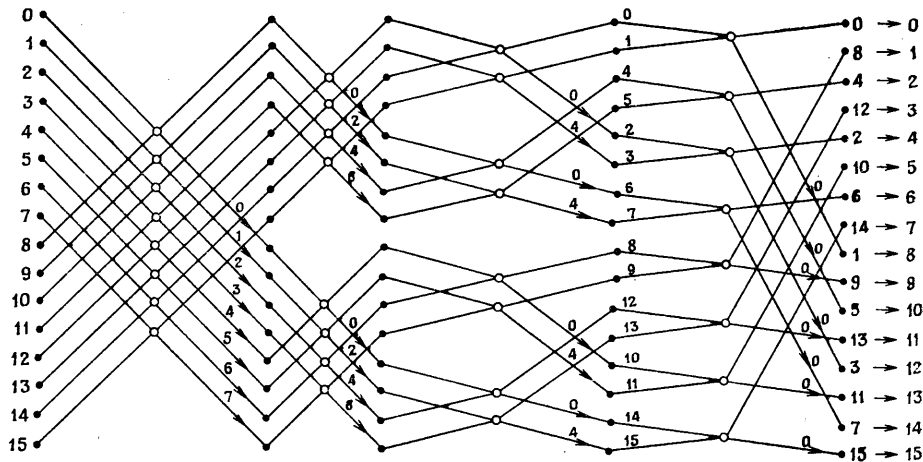
Фиг. 10.5. Алгоритм 16-точечного БПФ по основанию 2 с прореживанием по времени, без замещения, с нормальным порядком на входе и выходе.

комплексных регистров, при работе без замещения —  $2N$  комплексных регистров. На фиг. 10.4 представлен алгоритм с постоянной структурой, в котором входные отсчеты располагаются в двоично-инверсном порядке, а выходные гармоники — в прямом. Поворачивающие множители для вариантов прореживания по времени и по частоте можно найти из схемы алгоритма фиг. 10.3, рассматривая ее от выхода к входу и поменяв направления всех стрелок. На фиг. 10.5 представлен алгоритм, в котором за счет работы без замещения удается избежать двоичной инверсии выходных гармоник. Здесь показан только вариант прореживания по времени. Вариант прореживания по частоте нетрудно получить, переместив стрелки за вершины графа.

На фиг. 10.6 представлена разновидность алгоритма БПФ, позволяющая использовать преимущества сверхоперативного запоминающего устройства. Особенности аппаратной реализации этой структуры будут рассмотрены позднее; отметим лишь, что вершины графа можно объединить в пары и выполнять сразу по две базовые операции над четырьмя отсчетами, так что для этих отсчетов будут проведены сразу два этапа БПФ, и только после этого будут аналогично обрабатываться следующие четыре отсчета. Допустим, например, что отсчеты 0 и 8 поступают в нулевую вершину нулевого этапа БПФ, а отсчеты 4 и 12 — в четвертую вершину того же этапа. После выполнения этих двух базовых операций перейдем к вершинам 0 и 4 первого этапа и убедимся, что для них все исходные числа снова размещаются в тех же четырех регистрах с номерами 0, 4, 8 и 12. Содержимое регистров 1, 5, 9 и 13 можно также пропустить сразу через два этапа БПФ. Таким образом, если арифмети-

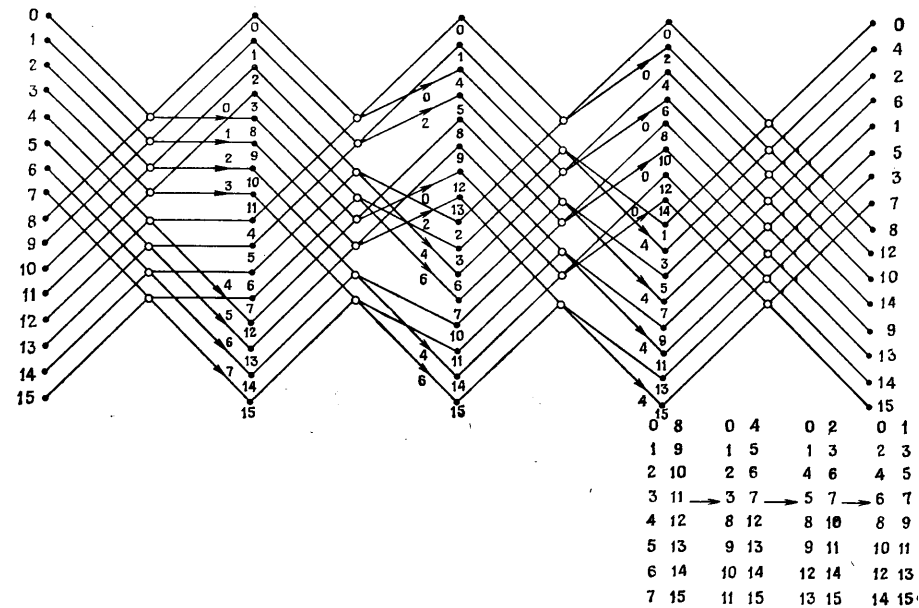


Фиг. 10.6. Алгоритм 16-точечного БПФ по основанию 2 с совместным выполнением двух этапов над четырьмя отсчетами без промежуточных обращений к ЗУ. Входные отсчеты — в прямом порядке, выходные — переставлены (но не в двоично-инверсионном порядке).



Фиг. 10.7. Алгоритм БПФ с прореживанием по частоте и совместным выполнением двух этапов, устраняющий двоичную инверсию на выходе.

численное устройство способно обрабатывать не два, а сразу четыре отсчета, то необходимое число циклов обращения к памяти может быть сокращено в два раза. Данную структуру можно рассматривать как алгоритм с замещением, в котором выполняются сразу два этапа, поэтому порядок результатов будет двоично-инверсным.



Фиг. 10.8. Алгоритм с совместным выполнением двух этапов, допускающий двукратное распараллеливание.

На фиг. 10.7 показано, как можно избежать двоичной инверсии. В этом алгоритме 16-точечного БПФ первые два этапа выполняются так же, как и в алгоритме на фиг. 10.1, т. е. как обычные операции с замещением. Затем четверки отсчетов вводятся в арифметическое устройство, где выполняются две базовые операции, после чего их результаты переставляются, как показано для двух последних этапов алгоритма. Заметим, что в данном случае регистры пронумерованы. Это сделано для того, чтобы проиллюстрировать характер нумерации. Так как нумерация регистров на выходе является двоично-инверсионной, то выходные гармоники будут располагаться в нормальном порядке. (Действительно, в обычном алгоритме с замещением нумерация регистров не изменяется и порядок следования результатов является двоично-инверсным.)

На фиг. 10.8 представлен еще один алгоритм с одновременным выполнением двух базовых операций, который отличается от алгоритма фиг. 10.7. Здесь регистры ЗУ организованы так, чтобы можно было считывать (или записывать) одновременно по два комплексных слова. Так, например, отсчеты 0 и 8 считываются для выполнения базовой операции параллельно, за счет чего удается сэкономить один цикл обращения к памяти. В расположенной справа таблице показана желательная группировка отсчетов

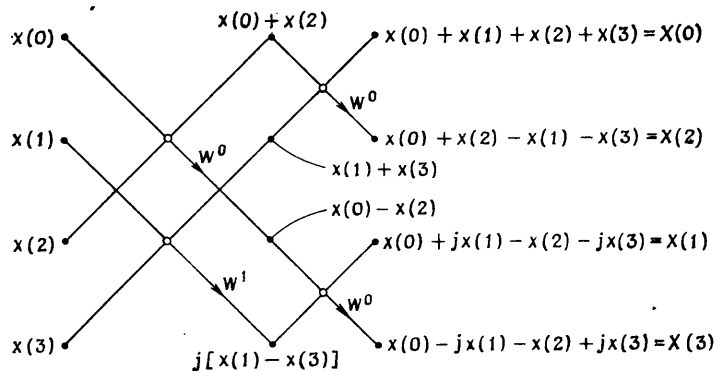
на различных этапах БПФ. Чтобы осуществить ее и обеспечить параллельную работу, необходимо одновременно поменять местами четыре совместно обрабатываемых отсчета. Например, отсчеты 0, 8 и 4, 12 переставляются, а затем обрабатываются пары отсчетов 0, 4 и 8, 12. Снова напомним, что регистры на схеме пронумерованы. Конечные результаты (правая колонка чисел) показывают, что выходные гармоники переставлены, причем перестановка является результатом инверсии и матричной перестановки, получающейся при заполнении матрицы размером  $(8 \times 2)$  по строкам с последующим чтением ее по столбцам.

**10.3. Нумерация при БПФ. Двоичная инверсия и разрядная инверсия для алгоритмов БПФ с постоянным основанием**

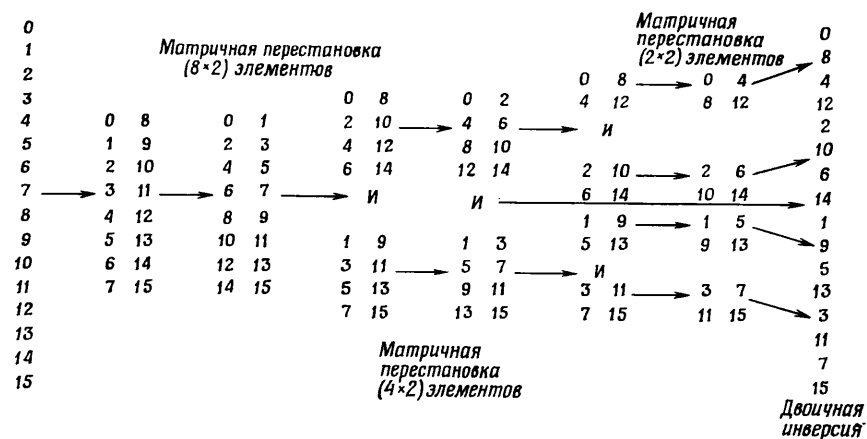
Начнем с очень простого примера, иллюстрирующего, каким образом выходные гармоники БПФ с замещением оказываются переставленными, в результате чего  $X(k)$  размещается, вообще говоря, не в  $k$ -м регистре, хотя отсчет  $x(k)$  находился в нем. Пусть задана четырехточечная последовательность  $x(n)$ . Ее ДПФ равно

$$\begin{aligned}
 X(0) &= x(0) + x(1) + x(2) + x(3) = x(0) + x(1) + x(2) + x(3), \\
 X(1) &= x(0) + x(1)W^1 + x(2)W^2 + x(3)W^3 = \\
 &= x(0) + jx(1) - x(2) - jx(3), \\
 X(2) &= x(0) + x(1)W^2 + x(2)W^4 + x(3)W^6 = x(0) - x(1) + x(2) - x(3), \\
 X(3) &= x(0) + x(1)W^3 + x(2)W^6 + x(3)W^9 = \\
 &= x(0) - jx(1) - x(2) + jx(3).
 \end{aligned}$$

Выполним эти вычисления с помощью четырехточечного БПФ по схеме, изображенной на фиг. 10.9. Заметим, что  $X(1)$  и  $X(2)$



Фиг. 10.9. Алгоритм четырехточечного БПФ по основанию 2 с прореживанием по частоте и с замещением.



Фиг. 10.10. Иллюстрация двоичной инверсии для 16-точечного БПФ с помощью последовательности матричных перестановок.

поменялись местами, т. е. гармоники оказались переставленными. Такую перестановку, характерную для БПФ, можно объяснить, используя описанную в гл. 6 возможность выполнения одномерного ДПФ по схеме двумерного ДПФ. Существенно, что результат преобразования имеет матричную перестановку. Если рассматривать алгоритм БПФ как последовательность прореживаний (путем представления одномерных массивов двумерными), то перестановку гармоник можно рассматривать как результат выполнения последовательности матричных перестановок в массивах уменьшающихся размеров, как показано на фиг. 10.10 для 16-точечного БПФ.

Оказывается, что в алгоритмах БПФ с постоянным основанием порядок следования гармоник достаточно просто определяется порядком следования входных отсчетов. Правило определения порядка на выходе, которое будет сначала просто сформулировано, а затем и доказано, известно под названием *двоичной инверсии* для БПФ с основанием 2 или (в общем случае) *разрядной инверсии* для основания  $r$ , причем номера отсчетов в последнем случае записываются в  $r$ -ичной системе счисления. Так, если при БПФ с основанием 2 отсчет  $f(n)$  находится в  $n$ -м регистре, причем  $n$  представляется двоичным числом вида  $x_5x_4x_3x_2x_1x_0$  (т. е. 6-разрядным адресом, в котором  $x_j$  равны нулю или единице), то выходная гармоника  $F(n)$  окажется в регистре с адресом  $x_0x_1x_2x_3x_4x_5$ , для которого расположение двоичных разрядов оказывается *инверсным*. Все сказанное справедливо и для БПФ с основанием 4, когда  $x_j$  представляют собой четверичные цифры, принимающие

значения 0, 1, 2 или 3; для алгоритмов с основанием 8  $x_j$  будут *восьмеричными* цифрами.

Для доказательства разрядной инверсии используем простое численное соотношение, описывающее связь между положением элементов двумерного массива до и после матричной перестановки. Рассмотрим матрицу

$$\begin{array}{ccccccc} 0 & 1 & 2 & \dots & m-1 & & \\ m & m+1 & m+2 & \dots & 2m-1 & & \\ 2m & & & & & & \\ \vdots & & & & & & \\ \vdots & & & & & & \\ (p-1)m & & & & & & N-1, \end{array}$$

состоящую из  $p$  строк и  $m$  столбцов и содержащую  $N = mp$  элементов. Если каждый элемент матрицы умножить на  $p$  по модулю  $(N-1)$ , то результат будет представлять уже упоминавшуюся выше матричную перестановку. Таким образом, преобразованная матрица имеет вид

$$\begin{array}{ccccccc} 0 & p & 2p & \dots & (m-1)p & & \\ 1 & p+1 & 2p+1 & \dots & (m-1)p+1 & & \\ 2 & & & & & & \\ \vdots & & & & & & \\ \vdots & & & & & & \\ p-1 & & & & & & N-1. \end{array}$$

Заметим далее, что если  $p = 2$ , то умножение двоичного числа на 2 по модулю  $(N-1)$  эквивалентно круговому сдвигу разрядов по часовой стрелке на одну позицию. Аналогично умножение на 4 по модулю  $(N-1)$  эквивалентно круговому сдвигу по часовой стрелке на *две позиции* или на *одну* четверичный разряд. В общем случае умножение на  $r$  по модулю  $(N-1)$  соответствует круговому сдвигу по часовой стрелке на *одну*  $r$ -ичный разряд.

Рассмотрим, что произойдет при выполнении в  $r$ -ичном числе определенной последовательности таких круговых сдвигов на  $r$ -ичный разряд. Пусть исходное число имеет вид

$$x_5x_4x_3x_2x_1x_0.$$

1. Сдвинем по кругу все шесть разрядов:

$$x_4x_3x_2x_1x_0x_5.$$

2. Сдвинем по кругу пять старших разрядов:

$$x_3x_2x_1x_0x_4x_5.$$

3. Сдвинем по кругу четыре старших разряда:

$$x_2x_1x_0x_3x_4x_5.$$

4. Сдвинем по кругу три старших разряда:

$$x_1x_0x_2x_3x_4x_5.$$

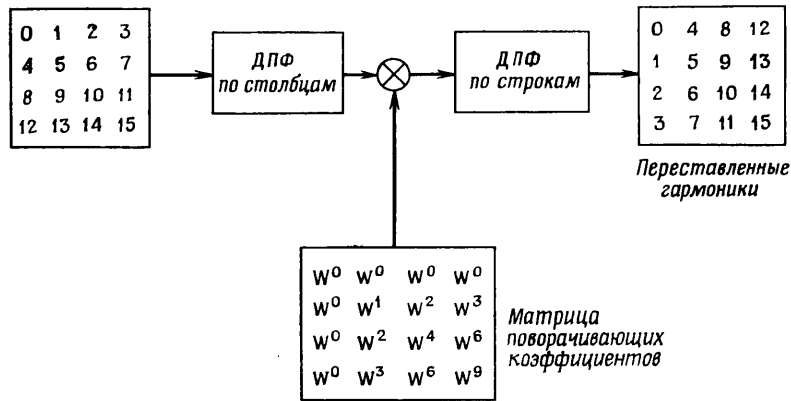
5. Сдвинем по кругу два старших разряда:

$$x_0x_1x_2x_3x_4x_5.$$

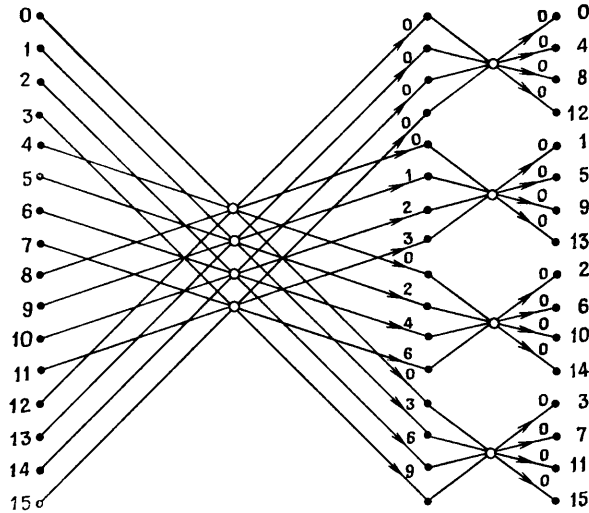
Теперь остается доказать, что при БПФ с основанием  $r$  перестановку гармоник можно описать инверсией разрядов. Начнем с представления исходного массива в виде матрицы размером  $[(N/r) \times r]$ . Этапу 1 соответствует матричная перестановка всего массива. После нее каждый из  $r$  столбцов (размером по  $N/r$ ) также следует представить матрицей размером  $[(N/r^2) \times r]$ . Преобразование каждой из этих матриц связано с круговым сдвигом всех разрядов, за исключением младшего. Аналогично при переходе от этапа к этапу будет сдвигаться все меньшее число старших разрядов. Поясним сказанное на примерах. Рассмотрим 16-точечное ДПФ по основанию 2 с прямым порядком следования исходных отсчетов. Выберем алгоритм с прореживанием по частоте и будем выполнять по строкам двухточечные ДПФ. Порядок размещения результатов будет описываться круговым смещением разрядов на этапе 1 и соответствовать матричной перестановке исходного массива. Подчеркнем, что адрес каждого регистра заменяется на новый, получаемый на этапе 1 описанного выше процесса разрядной инверсии. Но каждый из столбцов, используемых на этапе 1 разрядной инверсии, представляется матрицей из  $(4 \times 2)$  элементов, так что их адреса изменяются в соответствии с этапом 2 процесса разрядной инверсии. Рассматривая, наконец, каждый столбец этапа 2 инверсии как матрицу из  $(2 \times 2)$  элементов, получаем перестановку выходных результатов, действительно являющуюся двоичной инверсией. Важно не забывать, что адреса регистров соответствуют *геометрическому* положению чисел и считаются в процессе преобразования неизменными. Так, например, после выполнения всего алгоритма в регистре 14 находится седьмая гармоника, в регистре 5 — десятая и т. д.

#### Алгоритм с основанием 4

Если число  $N$  является степенью 4, то его можно записать как  $(N/4) \times 4$ ; аналогично  $N/4 = (N/16) \times 4$  и т. д. В результате элементы исходного одномерного массива можно расположить таким образом, чтобы элементарные операции представляли собой четырехточечные ДПФ. Простой пример для  $N = 16$  показан

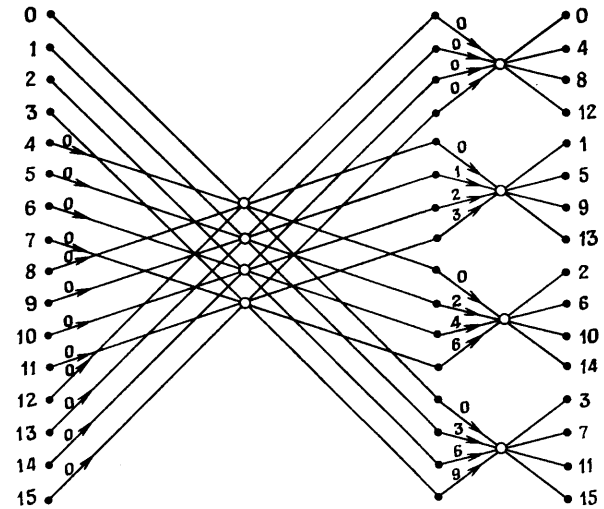


Фиг. 10.11. 16-точечное БПФ с основанием 4.



Фиг. 10.12. 16-точечное БПФ по основанию 4 с прореживанием по частоте, прямым порядком на входе и четверично-инверсным на выходе.

на фиг. 10.11. Для этого случая достаточно одного прореживания, осуществляемого путем представления одномерного массива матрицей размером  $(4 \times 4)$ . Выполняемые операции можно представить схемой (фиг. 10.12), аналогичной схемам БПФ с основанием 2. На фиг. 10.12 использованы обозначения, уже описанные в гл. 6, т. е. вершины (незачерненные кружки) представляют  $K$ -точечные ДПФ, где  $K$  равно числу линий, входящих в вершину и выходящих



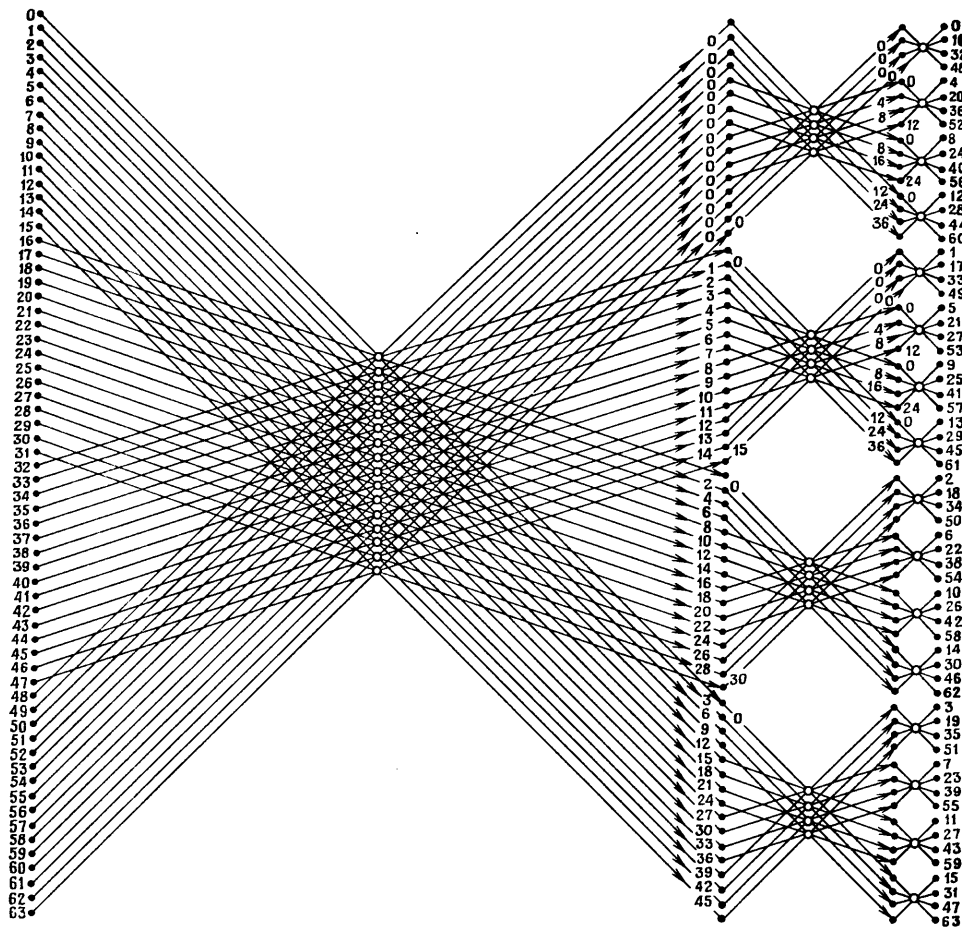
Фиг. 10.13. 16-точечное БПФ по основанию 4 с прореживанием по времени и с замещением, нормальным порядком на входе и четверично-инверсным на выходе.

из нее. Таким образом, на фиг. 10.12 эти кружки представляют четырехточечные ДПФ, а умножения на поворачивающие множители изображены, как и на схемах БПФ с основанием 2, стрелками. Ячейки памяти также представлены точками и пронумерованы сверху вниз, причем номера регистров не приводятся.

По аналогии с БПФ по основанию 2 возможны варианты БПФ по основанию 4 с прореживанием по частоте и по времени. Первый вариант представлен на фиг. 10.12, а второй — на фиг. 10.13.

Интересно сравнить количество арифметических операций, выполняемых в схемах фиг. 10.12 и 10.13 и в любой из 16-точечных структур с основанием 2, приведенных на фиг. 10.1—10.8. Во втором случае используются 10 нетривиальных комплексных умножений (тривиальными являются умножения на  $\pm 1$  или на  $\pm j$ ). Аналогичный подсчет для БПФ с основанием 4 дает восемь таких умножений. Таким образом, уже отсюда видно, что эффективность различных структур БПФ с одинаковыми конечными результатами неодинакова. Но решение вопроса о том, выгодно ли применение алгоритма БПФ с основанием 4, зависит в каждом конкретном случае от большого числа факторов и здесь не рассматривается.

На фиг. 10.14 изображена схема алгоритма БПФ с основанием 4 для массива из 64 отсчетов. Чтобы понять, каким образом в этом алгоритме осуществляется прореживание, будем считать, что первые 16 отсчетов образуют первую строку матрицы из 16 столбцов и 4 строк, следующие 16 отсчетов составляют вторую строку и



Фиг. 10.14. 64-точечное БПФ по основанию 4 с прореживанием по частоте, нормальным порядком на входе и четверично-инверсным на выходе.

т. д. На первом этапе алгоритма фиг. 10.14 выполняется 16 четырехточечных ДПФ от элементов каждого столбца. Затем производятся повороты, причем показатели экспонент поворачивающих множителей указаны перед обозначением ячеек памяти второго этапа. Затем все четыре строки прореживаются путем формирования из них матриц размером  $(4 \times 4)$  и преобразуются по схеме 16-точечного БПФ с основанием 4, но с другими поворачивающими множителями.

Алгоритм фиг. 10.14 можно рассматривать как вариант БПФ с прореживанием по частоте, так как повороты производятся

после ДПФ. Можно построить и алгоритм с прореживанием по времени, но для этого придется переходить от матриц меньшего размера к более крупным матрицам. Читателю, вероятно, интересно самому найти значения поворачивающих множителей для варианта алгоритма фиг. 10.14 с прореживанием по времени. Как получить нужный результат, исходя из повторяющейся процедуры сведения матрицы к четырехточечным ДПФ?

#### 10.4. Сопоставление объема вычислений при использовании оснований 2, 4 и 8

Алгоритмы БПФ с основаниями 4 и 8 представляют и теоретический, и практический интерес, так как они позволяют получить выигрыш (по сравнению с алгоритмом по основанию 2) как при аппаратной, так и при программной реализации БПФ. Покажем это качественно, выбрав  $N = 64$  и подсчитав число комплексных умножений, необходимых для выполнения БПФ с основаниями 2, 4 и 8.

1. Для БПФ с основанием 2 и прореживанием по времени на первом этапе используются умножения только на  $W^0 = 1$ , т. е. фактически умножений нет. На втором этапе фигурируют поворачивающие множители  $W^0$  и  $W^{16} = j$ , т. е. умножений снова нет. На третьем этапе в половине из общего числа операций, равного  $N/2$ , выполняются комплексные умножения и т. д. В результате число комплексных умножений, выполняемых при  $N$ -точечном БПФ с основанием 2, можно записать в виде

$$M_N = \frac{N}{2} \sum_{n=1}^{\log_2 N - 2} \frac{2^n - 1}{2^n} = \frac{N}{2} (\log_2 N - 2) - \frac{N}{2} \left[ \frac{1 - 2^{-(\log_2 N - 1)}}{1 - 2^{-1}} \right] + 1. \quad (10.1)$$

Упрощая формулу (10.1), получим

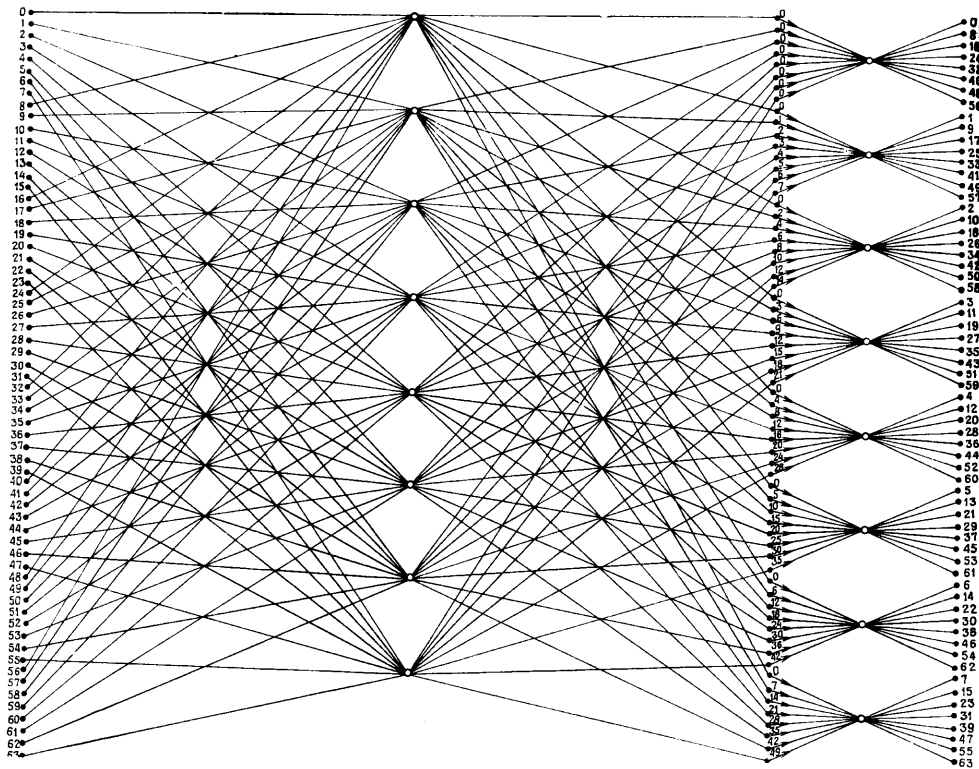
$$M_N = \frac{N}{2} \log_2 N - \frac{3}{2} N + 2. \quad (10.2)$$

При  $N = 64$  из формулы (10.2) следует, что  $M_N = 98$ . Заметим, что эта формула верна также при  $N = 2$  и  $N = 4$ , когда  $M_N = 0$ . Формула (10.2) справедлива, если  $N$  равно степени двойки.

2. Формулу, аналогичную (10.2), можно вывести и для БПФ с основанием 4, но в данном случае проще подсчитать количество умножений, используемых в алгоритме фиг. 10.14. На первом этапе при выполнении поворотов требуются 44 умножения, на втором — 32, а на третьем они не используются. Всего получается 76 умножений — заметно меньше по сравнению с предыдущим случаем.

3. В алгоритмах с основаниями 2 и 4 операция вычисления ДПФ не содержит умножений, т. е. все умножения являются





Фиг. 10.15. 64-точечное БПФ по основанию 8 с прореживанием по времени и с замещением.

поворачивающими. При более высоких основаниях ситуация меняется. В частности, для восьмиточечного ДПФ, выполняемого методом БПФ, в соответствии с формулой (10.2) требуются два умножения (на числа вида  $\pm a$  и  $\pm ja$ ). Таким образом, в алгоритме БПФ с основанием 8 умножения фигурируют при выполнении и ДПФ, и поворотов. На фиг. 10.15 представлен алгоритм 64-точечного БПФ с основанием 8. Он включает 16 восьмиточечных ДПФ (незачерненные кружки на фиг. 10.15), требующих 32 умножений, а также 48 нетривиальных умножений на поворачивающие множители, так что всего выполняется  $32 + 48 = 80$  умножений<sup>1)</sup>.

<sup>1)</sup> Можно показать, что так как в восьмиточечном ДПФ в обеих операциях умножения используется один и тот же коэффициент, то обе они эквивалентны скорее одному, а не двум комплексным умножениям. Тогда на БПФ с основанием 8 приходится  $16 + 48 = 64$  умножений, что делает основание 8 «лучшим» по сравнению с основанием 4.

Поэтому представляется, что основание 4 является в некотором смысле «оптимальным» по крайней мере для 64-точечных БПФ. Эту «оптимальность», конечно, не следует понимать буквально и использовать всегда только основание 4. Скорее она означает, что при разработке любой достаточно большой системы необходимо проанализировать возможность применения всех трех оснований.

### 10.5. Эффекты квантования в алгоритмах БПФ

При составлении программ (или при аппаратной реализации БПФ) необходимо учитывать эффекты, связанные с представлением обрабатываемых чисел и коэффициентов с ограниченной точностью. К таким эффектам относятся шумы округления при усечении результатов умножений, погрешности масштабирования промежуточных результатов с целью устранения возможных переполнений, а также погрешности преобразования, связанные с конечной точностью представления коэффициентов  $W^k$ . Уэлч и ряд других авторов подробно проанализировали влияние шума округления и масштабирования для алгоритмов БПФ по основанию 2 с прореживанием по времени и по частоте при условии, что все арифметические операции выполняются с фиксированной или плавающей запятой. Вопрос, связанный с точностью представления коэффициентов, не рассматривался столь подробно, поэтому ниже будут приведены лишь качественные результаты. Сначала будут рассмотрены вопросы реализации алгоритмов БПФ с основанием 2 при выполнении арифметических операций с фиксированной запятой.

Если последовательность  $\{x(n)\}$  из  $N$  отсчетов имеет коэффициенты ДПФ  $\{X(k)\}$ , то, согласно теореме Парсевала,

$$\sum_{n=0}^{N-1} x^2(n) = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2, \quad (10.3)$$

т. е. средняя мощность выходных гармоник в  $N$  раз превышает среднюю мощность исходной последовательности. Следовательно, значения ДПФ последовательности будут, вообще говоря, существенно превышать значения самой последовательности. Поэтому при использовании арифметики с фиксированной запятой необходимо во избежание переполнений ввести масштабирование. Чтобы показать, как могут возникать переполнения, рассмотрим базовую операцию на  $m$ -м этапе БПФ с прореживанием по времени. Пусть  $f_m(i)$  и  $f_m(j)$  — входные числа базовой операции, а  $f_{m+1}(i)$  и  $f_{m+1}(j)$  — выходные числа, причем  $W^p$  — поворачи-

вающий множитель. Тогда

$$\begin{aligned} f_{m+1}(i) &= f_m(i) + W^p f_m(j), \\ f_{m+1}(j) &= f_m(i) - W^p f_m(j). \end{aligned} \quad (10.4)$$

Из формулы (10.4) видно, что при переходе от этапа к этапу модули чисел, вообще говоря, увеличиваются, поэтому их нужно масштабировать сдвигом вправо. Нетрудно показать, что от этапа к этапу максимальное значение модуля комплексных чисел не убывает и, как видно из соотношения (10.4), удовлетворяет неравенству

$$\begin{aligned} \max\{|f_m(i)|, |f_m(j)|\} &\leq \max\{|f_{m+1}(i)|, |f_{m+1}(j)|\} \leq \\ &\leq 2 \max\{|f_m(i)|, |f_m(j)|\}. \end{aligned} \quad (10.5)$$

Следовательно, уровень сигнала на каждом этапе увеличивается не быстрее, чем на один двоичный разряд.

Учитывая вышесказанное, можно предложить три метода масштабирования:

1. Сдвиг вправо на один разряд на каждом этапе. Если  $|f_0(i)| \leq 1/2$  для любых  $i$  и числа сдвигаются на разряд вправо после каждого этапа итерации (кроме последнего), то переполнения не произойдет.

2. Контроль последовательности, гарантирующий выполнение условия  $|f_m(i)| \leq 1/2$  для всех  $i$ . На каждом этапе БПФ вычисляется массив  $f_m(i)$ , и если хотя бы один отсчет массива превысит по модулю  $1/2$ , весь массив масштабируется сдвигом вправо на разряд.

3. Проверка на переполнение. При этом исходная последовательность масштабируется так, чтобы  $|\operatorname{Re}\{f_0(i)\}|$  и  $|\operatorname{Im}\{f_0(i)\}|$  были меньше единицы (а не  $1/2$ , как в двух предыдущих методах). Если в ходе базовой операции фиксируется переполнение, то все числа последовательности, включая результаты уже выполненных на данном этапе (без переполнения) базовых операций, сдвигаются на разряд вправо, после чего итерация продолжается с той базовой операции, где произошло переполнение. На каждом этапе может произойти более чем одно переполнение, но не больше двух.

Первый метод связан с наименьшими затратами времени и наиболее прост для программирования. Однако он дает наименьшую точность, так как масштабирование на каждом этапе приводит к ненужному ухудшению точности в случаях, когда его можно было не производить. Второй метод требует больших затрат времени (поскольку на каждом этапе приходится вычислять модули всех чисел массива), и в то же время он не слишком точный, так как весь массив всегда масштабируется таким образом, чтобы модули всех его элементов не превышали  $1/2$ , поэтому один разряд памяти

данных не используется. Третий метод является самым точным, но здесь приходится повторно обрабатывать всю последовательность всякий раз, когда обнаруживается переполнение.

Уэлч разработал довольно простую методику расчета граничных значений уровня шума округления, возникающего при БПФ. Наличие шума округления обусловлено двумя факторами:

1. Появлением погрешности при округлении произведения  $W^p f_m(j)$  в формулах (10.4). Если  $W^p$  и  $f_m(j)$  представлены  $b$ -разрядными числами (т. е. действительная и мнимая части  $W^p$  и  $f_m(j)$  содержат по  $b$  разрядов), то после округления каждого из четырех действительных произведений, входящих в  $W^p f_m(j)$ , до  $b$ -разрядного числа получается ошибка, равномерно распределенная на интервале  $(-2^{-b}/2, 2^{-b}/2)$  и имеющая нулевое среднее и дисперсию, равную  $2^{-2b}/12$ .

2. Сдвигом суммы на один разряд вправо, если при сложении происходит переполнение. Если младший разряд, при сдвиге выходящий за пределы регистра, равен нулю, то ошибки не будет. Если же он равен единице, то в зависимости от знака числа возникает ошибка величиной  $\pm 2^{-b}$ . Дисперсия этой ошибки равна  $2^{-2b}/2$ .

Верхнюю границу<sup>1)</sup> отношения среднеквадратического значения шума округления к среднеквадратическому значению выходных гармоник можно получить, предположив, что переполнение происходит на каждом этапе, так что числа каждый раз приходится масштабировать. Если  $f_k(j)$  — отсчеты, получаемые на  $k$ -м этапе, то  $D(f_k)$  — их дисперсия, определяемая как

$$D(f_k) = \frac{1}{N} \sum_{j=0}^{N-1} D\{f_k(j)\}, \quad (10.6)$$

причем

$$D(f_0) = \frac{2^{-2b}}{2} = 6\Delta^2, \quad (10.7)$$

где  $\Delta^2 = 2^{-2b}/12$ , так что  $6\Delta^2 = 2^{-2b}/2$ . Равенство (10.7) получается в предположении, что на первом же этапе может произойти переполнение, поэтому в числах нулевого этапа (т. е. в исходных отсчетах) один младший разряд отбрасывается. На первом этапе базовая операция содержит умножения только на  $\pm 1$ , поэтому шум округления произведений отсутствует.

На втором этапе, согласно предположению, также может произойти переполнение, поэтому массив  $f_1(j)$  нужно предварительно

<sup>1)</sup> Следует отметить, что определяемая ниже верхняя граница представляет собой математическое ожидание искомого отношения, поэтому в экспериментальных измерениях она может быть превышена.

промасштабировать. Следовательно, дисперсия  $f_1$  будет равна

$$D(f_1) = 2D(f_0) + (4 \times 6\Delta^2) = 36\Delta^2, \quad (10.8)$$

причем множитель 4 отражает тот факт, что ошибка при усечении выходных чисел на первом этапе будет вдвое больше ошибки, обусловленной усечением выходных чисел на нулевом этапе, поэтому дисперсия возрастает в четыре раза.

На втором этапе алгоритма БПФ с прореживанием по времени выполняются умножения только на  $\pm 1$  и  $\pm j$ , поэтому ошибка, связанная с округлением произведений, опять будет равна нулю. Итак, в предположении, что числа  $f_2$  также масштабируются, получим следующую формулу для дисперсии ошибки на втором этапе БПФ:

$$D(f_2) = 2D(f_1) + (4^2 \times 6\Delta^2) = 168\Delta^2, \quad (10.9)$$

причем множитель  $4^2$  указывает на удвоение ошибки усечения при переходе от первого этапа ко второму.

На всех последующих этапах в большинстве базовых операций выполняются нетривиальные умножения. Так, на третьем этапе половина базовых операций содержит такие умножения, на четвертом этапе — три четверти и т. д. Согласно формулам (10.4), общее выражение для действительной части результата базовой операции, выполняемой на  $m$ -м этапе, имеет вид

$$\operatorname{Re}[f_{m+1}(i)] = \operatorname{Re}[f_m(i)] + \operatorname{Re}[f_m(j)] \operatorname{Re}(W^p) - \operatorname{Im}[f_m(j)] \operatorname{Im}(W^p). \quad (10.10)$$

Аналогичное выражение можно записать и для мнимой части. Для тех базовых операций, где умножение на  $W^p$  не является тривиальным, дисперсия результата  $f_{m+1}$  (включая шум округления всех произведений и шум усечения при масштабировании) имеет вид

$$D(f_{m+1}) = D(f_m) + D(W^p) \{ \overline{\operatorname{Re}^2[f_m(j)]} + \overline{\operatorname{Im}^2[f_m(j)]} \} + \{ \operatorname{Re}^2(W^p) + \operatorname{Im}^2(W^p) \} D(f_m) + D(e_{m+1}^{\text{ок}}) + D(e_{m+1}^{\text{ш}}), \quad (10.11)$$

где слагаемое  $D(e_{m+1}^{\text{ок}})$  учитывает дисперсию шума округления на  $m$ -м этапе, а член  $D(e_{m+1}^{\text{ш}})$  — дисперсию шума, связанную с масштабированием на этом же этапе. Как уже отмечалось, оба эти члена равны

$$\begin{aligned} D(e_{m+1}^{\text{ок}}) &= 4^m \Delta^2 \\ D(e_{m+1}^{\text{ш}}) &= 4^m \times 6\Delta^2. \end{aligned} \quad (10.12)$$

Третье слагаемое суммы (10.11) содержит произведение квадрата модуля  $W^p$  и  $D(f_m)$ . Поскольку  $|W^p| = 1$ , это слагаемое совпадает с первым членом суммы. Второе слагаемое равно произведению

$D(W^p)$  на среднее значение квадрата модуля переменных, преобразуемых на  $m$ -м этапе. Дисперсия  $W^p$  равна  $\Delta^2$  (поскольку  $W^p$  представляется  $b$ -разрядным числом с погрешностью). Чтобы определить средний квадрат модуля переменных, обрабатываемых на  $m$ -м этапе, нужно знать величину  $K$ , представляющую собой средний квадрат модуля исходного массива чисел, равный

$$K = \overline{f_0^2(j)} = \frac{1}{N} \sum_{j=0}^{N-1} |f_0(j)|^2. \quad (10.13)$$

Учитывая, что среднее значение квадрата модуля увеличивается от этапа к этапу вдвое, получим, что на выходе  $m$ -го этапа оно будет равно  $2^m K$ . Тогда дисперсия  $f_{m+1}$  становится равной

$$D(f_{m+1}) = 2D(f_m) + 2^m K \Delta^2 + 4^{m+1} \Delta^2 + (4^{m+1} \times 6\Delta^2), \quad (10.14)$$

если во всех базовых операциях  $m$ -го этапа выполняются нетривиальные умножения. Если же ввести  $\alpha$  — долю базовых операций с нетривиальными умножениями ( $0 \leq \alpha \leq 1$ ), — то второе и третье слагаемые суммы (10.14) следует умножить на  $\alpha$ . Так, если  $\alpha = 0$  (как это имеет место на первом и втором этапах), то равенство (10.14) переходит в (10.8) (при  $m = 0$ ) или в (10.9) (при  $m = 1$ ). Если принять, что  $\alpha = 1/2$  на третьем этапе и  $\alpha = 1$  на всех последующих этапах, причем номер последнего этапа равен  $M$ , то, согласно формулам (10.7)–(10.14), получим

$$\begin{aligned} D(f_m) &= 2^M (6\Delta^2) + 2^{M-1} (4 \times 6\Delta^2) + \dots + 2 (4^{M-1} \times 6\Delta^2) + \\ &+ 2^{M-2} K \Delta^2 + (M-3) 2^{M-1} K \Delta^2 + \\ &+ 2^{M-4} (4^3 \Delta^2) + 2^{M-4} (4^4 \Delta^2) + \dots + (4^M \Delta^2) = \\ &= (1,5) 2^{M+2} \Delta^2 (1+2+\dots+2^{M-1}) + \\ &+ (M-2,5) 2^{M-1} K \Delta^2 + 2^{M+2} \Delta^2 + \\ &+ 2^{M+4} (1+2+\dots+2^{M-4}) \approx \\ &\approx 2^{2M+3} \Delta^2 + (M-2,5) 2^{M-1} K \Delta^2 + 2^{M+2} \Delta^2. \end{aligned} \quad (10.15)$$

Поскольку среднее значение квадратов модулей элементов выходного массива  $\{f_m(j)\}$  равно  $2^m K$ , то средний квадрат их действительных (или мнимых) частей равен  $2^m K/2$ . Таким образом, оценкой отношения среднеквадратических значений (СКЗ) шума и сигнала на выходе (в пределе при больших  $M$ ) является величина

$$\begin{aligned} \frac{\text{СКЗ ошибки}}{\text{СКЗ сигнала}} &\approx \frac{2^{(M+3)/2} \Delta}{\sqrt{K/2}} = \frac{2^{(M+3)/2-2-b} (0,3)}{\text{СКЗ входного сигнала}} = \\ &= \frac{0,3 \sqrt{N} 2^{-b} \sqrt{8}}{\text{СКЗ входного сигнала}}. \end{aligned} \quad (10.16)$$

Итак, верхняя граница отношения ошибки к выходному сигналу возрастает как  $\sqrt{N}$ , т. е. по  $1/2$  разряда на этап.

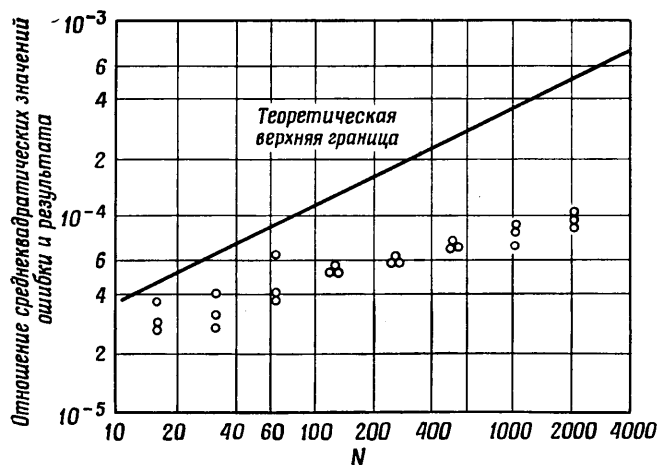
Формулу, аналогичную (10.16), можно получить и для нижней границы отношения ошибки к сигналу, если предположить, что переполнений вообще не происходит, так что ошибки усечения, связанные с масштабированием, отсутствуют. Следовательно, вплоть до третьего этапа ошибка вообще не возникает. Проведя аналогичный анализ, получим следующую формулу для дисперсии выходных отсчетов  $f_M$ :

$$D(f_M) = 2^{M-2} K \Delta^2 + (M-3) 2^{M-1} K \Delta^2 + 2^{M-3} \Delta^2 + \\ + 2^{M-5} \Delta^2 + 2^{M-6} \Delta^2 + \dots + \Delta^2 \approx (M-2,5) 2^{M-1} K \Delta^2 + \\ + 2^{M-3} \Delta^2 + 2^{M-4} \Delta^2. \quad (10.17)$$

Таким образом, при большом числе этапов  $M$  нижняя граница отношения среднеквадратических значений ошибки и сигнала на выходе имеет вид

$$\frac{\text{СКЗ ошибки}}{\text{СКЗ выходного сигнала}} \approx 0,3 \sqrt{M-2,5} 2^{-b}. \quad (10.18)$$

Рассмотренная модель возникновения ошибок была экспериментально подтверждена Уэлчем на многочисленных примерах. На фиг. 10.16 показано, как изменяется отношение среднеквадратических значений ошибки и сигнала на выходе, когда на вход



Фиг. 10.16. Сравнение расчетной верхней границы уровня шума при БПФ и экспериментальных результатов (по Уэлчу).

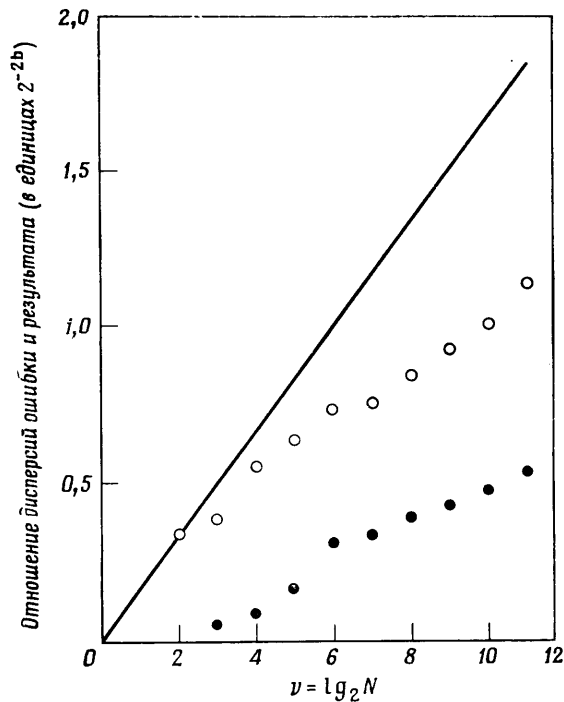
поступают случайные числа с нулевым средним, равномерно распределенные на интервале  $(-1, 1)$ , причем разрядность чисел равна 17. В этом случае  $\sqrt{K/2} = 0,58$ . Как видно из графика, теоретическая верхняя граница хорошо согласуется с экспериментальными данными, хотя ее наклон несколько больше, чем в эксперименте. Нижняя граница [см. формулу (10.18)] не приведена, так как она является слишком оптимистичной и малопримодна для практики.

Хотя выше был рассмотрен вариант алгоритма БПФ с прореживанием по времени, переход к другим вариантам несложен, и оценки верхней и нижней границ отношения среднеквадратических значений ошибки и сигнала, описываемые формулами (10.16) и (10.18), справедливы и для них. Таким образом, при разработке систем БПФ с фиксированной запятой можно руководствоваться формулой (10.16). Зная максимальный размер преобразуемого массива  $N$  и требуемое отношение среднеквадратических значений ошибки и выходного сигнала, можно выбрать длину слова  $b$  так, чтобы обеспечить требуемую точность.

В анализе, проведенном выше, рассматривались ошибки, связанные с округлением произведений и масштабированием, которое используется, чтобы избежать переполнений. При вычислении БПФ существует еще один источник погрешностей, обусловленный неточным представлением значений поворачивающих множителей  $W^p$ . Хотя полный анализ этого эффекта еще не проводился, Вайнштейн, используя предположение о случайном характере погрешностей представления коэффициентов, получил простое выражение для отношения среднеквадратических значений ошибки и сигнала на выходе

$$\frac{\sigma_{\text{ош}}^2}{\left[ \frac{1}{N} \sum_{n=0}^{N-1} |f(n)|^2 \right]} = \left( \frac{\log_2 N}{6} \right) 2^{-2b}. \quad (10.19)$$

Здесь  $b$  — число разрядов, используемых для представления коэффициентов. Из формулы (10.19) видно, что дисперсия ошибки с ростом  $N$  увеличивается весьма медленно. И хотя эта формула весьма приближенная, экспериментальные измерения подтверждают основной вывод о том, что дисперсия ошибки действительно с ростом  $N$  увеличивается очень медленно. На фиг. 10.17 приведены результаты, полученные Вайнштейном для случаев выполнения арифметических операций с фиксированной и плавающей запятой. Основное следствие из приведенных результатов заключается в том, что фиксированная длина слова (достаточно большая, чтобы обеспечить заданную точность представления коэффициентов) может использоваться для широкого диапазона значений  $N$ . Поэтому аппа-



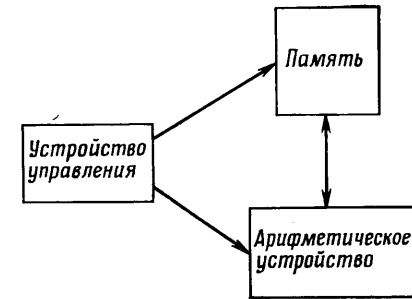
Фиг. 10.17. Сравнение предельного значения расчетной ошибки, связанной с квантованием коэффициентов БПФ, и экспериментальных результатов (по Вайнштейну).

— теория (фиксированная запятая); ○ эксперимент (фиксированная запятая); ● эксперимент (плавающая запятая).

ратуру или программы БПФ можно применять при решении широкого круга задач, для которых значения  $N$  могут отличаться на несколько порядков.

### 10.6. Особенности аппаратной реализации БПФ с основанием 2

Перейдем теперь к вопросам аппаратной реализации БПФ. Следует отметить, что между быстродействием и гибкостью устройства существует известное противоречие. То, что вычислительная машина  $A$  может выполнить БПФ в 10 раз быстрее вычислительной машины  $B$ , вовсе не означает, что она лучше, так как повышение скорости выполнения БПФ путем специализации структуры машины может привести к уменьшению скорости выполнения других алгоритмов, а общие характеристики машины при этом



Фиг. 10.18. Упрощенная блок-схема устройства БПФ.

значительно ухудшаются. Рассмотрим в качестве примера блок-схему устройства, предназначенного для выполнения алгоритмов с основанием 2, и будем пока считать, что оно содержит не более одного ЗУ с произвольным доступом и одного арифметического устройства (АУ), а также устройство управления (фиг. 10.18). Арифметические действия, выполняемые в ходе базовой операции, включая умножение на поворачивающий множитель  $W^p$ , можно записать в комплексной или действительной форме следующим образом:

#### 1. Комплексная форма записи

$$\left. \begin{aligned} f_{m+1}(i) &= f_m(i) + W^p f_m(j), \\ f_{m+1}(j) &= f_m(i) - W^p f_m(j), \end{aligned} \right\} \text{ прореживание по времени; (10.20)}$$

$$\left. \begin{aligned} f_{m+1}(i) &= f_m(i) + f_m(j), \\ f_{m+1}(j) &= [f_m(i) - f_m(j)] W^p, \end{aligned} \right\} \text{ прореживание по частоте. (10.21)}$$

Отметим, что формулы (10.20) совпадают с (10.4) и приведены здесь для удобства.

#### 2. Упрощенная комплексная форма

Чтобы показать последовательность выполнения алгоритма БПФ на вычислительной машине, избавимся в формулах (10.20) и (10.21) от различных индексов. Введем обозначения

$$\begin{aligned} f_{m+1}(i) &= A', & f_m(i) &= A, \\ f_{m+1}(j) &= C', & f_m(j) &= C. \end{aligned}$$

Тогда соотношения (10.20) и (10.21) можно записать в виде

$$\left. \begin{aligned} A' &= A + W^p C, \\ C' &= A - W^p C, \end{aligned} \right\} \text{ прореживание по времени; (10.22)}$$

$$\left. \begin{aligned} A' &= A + C, \\ C' &= (A - C) W^p, \end{aligned} \right\} \text{ прореживание по частоте.} \quad (10.23)$$

### 3. Действительная форма

Числа  $A$ ,  $A'$ ,  $C$  и  $C'$  в общем случае являются комплексными. Выразим их через действительные и мнимые части. Пусть  $A = a + jb$ ,  $A' = a' + jb'$ ,  $C = c + jd$ ,  $C' = c' + jd'$ . Тогда

$$\left. \begin{aligned} a' &= a + c \cos \frac{2\pi p}{N} - d \sin \frac{2\pi p}{N}, \\ c' &= a - c \cos \frac{2\pi p}{N} + d \sin \frac{2\pi p}{N}, \\ b' &= b + c \sin \frac{2\pi p}{N} + d \cos \frac{2\pi p}{N}, \\ d' &= b - c \sin \frac{2\pi p}{N} - d \cos \frac{2\pi p}{N}, \end{aligned} \right\} \text{ прореживание по времени;} \quad (10.24)$$

$$\left. \begin{aligned} a' &= a + c, \\ b' &= b + d, \\ c' &= (a - c) \cos \frac{2\pi p}{N} + (b - d) \sin \frac{2\pi p}{N}, \\ d' &= (a - c) \sin \frac{2\pi p}{N} - (b - d) \cos \frac{2\pi p}{N}, \end{aligned} \right\} \text{ прореживание по частоте.} \quad (10.25)$$

Исходя из блок-схемы фиг. 10.18 и приведенных формул, например из формулы (10.24), можно определить число машинных циклов, необходимых для выполнения базовой операции БПФ с прореживанием по времени. Предположим, что одно слово, хранящееся в ЗУ, содержит только действительную (или мнимую) часть комплексного числа, а в АУ имеется только один умножитель и один сумматор действительных чисел. Тогда для расчетов по формулам (10.24) потребуются шесть циклов обращения к ЗУ для вызова  $a$ ,  $b$ ,  $c$ ,  $d$ ,  $\cos(2\pi p/N)$  и  $\sin(2\pi p/N)$  и еще четыре цикла для записи результатов  $a'$ ,  $b'$ ,  $c'$ ,  $d'$ . Кроме того, нужно будет выполнить четыре умножения и шесть сложений (или вычитаний). В число дополнительных операций, не фигурирующих в формулах типа (10.24), входят в основном вспомогательные циклы обращения к ЗУ и действия над адресами. Так как эти операции тесно связаны со структурой ЦВМ, то в данном разделе они рассматриваться не будут. Отметим, между прочим, что с вычислительной точки зрения нет принципиального различия между вариантами БПФ с прореживанием по частоте и по времени.

Теперь можно перейти к более сложной модификации основной блок-схемы. Представим себе ЦВМ, в которой каждое слово со-

стоит из двух частей, причем одна из них используется для хранения действительной части комплексного числа, а другая — для хранения мнимой части того же числа. Представим также, что в АУ имеются два умножителя и два сумматора. Тогда при соответствующем управлении число машинных циклов можно сократить вдвое.

Если ЦВМ позволяет за один машинный цикл обрабатывать комплексное число, то полное число циклов можно сократить еще больше. Из формул (10.22), (10.23) видно, что нужно выполнить всего два комплексных сложения и одно комплексное умножение. Три возможных варианта блок-схемы, приведенной на фиг. 10.18, сопоставлены в табл. 10.1. В последнем столбце приводится сум-

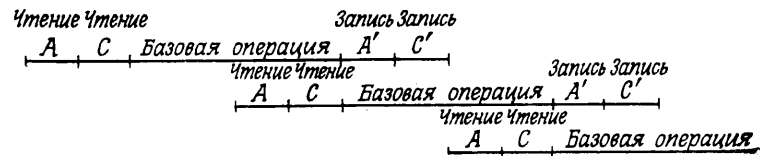
Таблица 10.1

	Количество циклов			
	Обращение к памяти	Сложение	Умножение	Всего
ЗУ с одиночными словами, сумматор и умножитель действительных чисел	10	6	4 (× 3)	28
ЗУ с двойными словами, два сумматора и два умножителя действительных чисел	5	3	2 (× 3)	14
ЗУ с комплексными словами, сумматор и умножитель комплексных чисел	5	2	1 (× 3)	10

марное число циклов, причем циклам обращения к ЗУ и циклам сложения придается единичный вес, а циклам умножения — тройной.

### 10.7. Оптимальная аппаратная реализация алгоритма БПФ с основанием 2

Теперь логично перейти к рассмотрению возможности *перекрытия* циклов обращения к памяти и циклов выполнения арифметических операций, так как эти действия выполняются различными устройствами. Отметим сначала, что при правильной адресации коэффициенты  $\cos(2\pi p/N)$  и  $\sin(2\pi p/N)$  не нужно извлекать из памяти в АУ для каждой базовой операции, поскольку они могут быть одинаковыми на протяжении нескольких циклов АУ. Следовательно, в верхней строке табл. 10.1 можно убрать два цикла обращения к памяти, а в двух нижних — по одному. Для систем, указанных во второй и третьей строках таблицы, нужно



Фиг. 10.19. Временные диаграммы для «оптимальной» системы с основанием 2.

обеспечить временное перекрытие четырех циклов обращения к ЗУ с работой АУ. «Оптимальная» временная диаграмма для такого режима работы приведена на фиг. 10.19.

Комплексные слова  $A$  и  $C$  вводятся в АУ; после выполнения половины вычислений в АУ вводятся числа для следующей базовой операции (предполагается, что для этого в АУ имеются специальные буферные регистры). После окончания первой базовой операции ее результаты возвращаются в ЗУ и сразу же начинается следующая базовая операция. Из диаграммы видно, что в такой «согласованной» системе длительность базовой операции должна быть равна четырем циклам обращения к ЗУ. Поскольку и ЗУ и АУ работают без пауз, данную структурную схему можно считать «оптимальной». То, что принятый способ синхронизации наилучшим образом обеспечивает согласование ЗУ и АУ, следует из того, что при уменьшении быстродействия одного из этих устройств общее быстродействие системы понизится, тогда как увеличение быстродействия только одного из них не приведет к повышению быстродействия всей системы.

Для оценки возможного быстродействия данной оптимальной схемы рассмотрим случай, когда  $N = 1024$ , а цикл обращения к ЗУ составляет  $\tau = 100$  нс. Полное время, затрачиваемое на выполнение БПФ по 1024 отсчетам, будет равно  $4\tau (N/2)\log_2 N \approx 2$  мс. Для обеспечения согласования между АУ и ЗУ необходимо, чтобы время выполнения базовой операции составляло  $4\tau = 400$  нс. Современный уровень технологии ИС вполне обеспечивает такое быстродействие.

Сравнение оптимальной схемы с другими схемами, приведенными в табл. 10.1, показывает, что при переходе от простейшего варианта одной и той же схемы к наиболее совершенному число циклов удается сократить от 28 до 4. Напомним, что время, затрачиваемое на вызов команд, не учитывалось. Если программа будет храниться в том же ЗУ, что и числа, то время выполнения БПФ значительно возрастет. Однако во многих современных ЦВМ имеется несколько блоков памяти с многоканальным доступом, что позволяет совмещать во времени обработку данных с подготовкой команд. Кроме того, не рассматривались дополнительные опера-

ции, используемые во всех алгоритмах БПФ. Они обеспечивают изменение коэффициентов и переход от этапа к этапу. Опыт показывает, что на все эти дополнительные операции (включая еще не упоминавшиеся операции ввода—вывода) тратится 20—70% от общего времени работы.

## 10.8. Ускорение выполнения БПФ методами параллельной обработки

Параллельная обработка может ускорить выполнение БПФ, как и других машинных алгоритмов. Вся последующая часть главы будет посвящена рассмотрению методов параллельной обработки, которые можно использовать для повышения эффективности алгоритмов БПФ. В настоящем разделе сделана попытка выделить некоторые общие методы, однако необходимо иметь в виду, что в любом алгоритме БПФ могут встретиться самые разнообразные сочетания нескольких видов параллелизма. Перечислим их:

1. Одновременное выполнение арифметических операций, обращений к ЗУ и операций над командами. На фиг. 10.19 (уже представляющей пример данного вида параллелизма) в явной форме показано временное перекрытие обращений к памяти и арифметических вычислений. Поскольку здесь не отведено время для адресации ЗУ и для выполнения команд управления, то эти операции также нужно выполнять параллельно с основными вычислениями.

2. Добавление ЗУ небольшого объема с большим быстродействием. В системе с одним арифметическим устройством можно без чрезмерных затрат использовать весьма быстродействующее АУ, однако увеличение быстродействия ЗУ является дорогостоящей операцией. Ниже будет показано, что сочетание двух ЗУ с разным быстродействием может оказаться недорогим способом увеличения быстродействия всей системы.

3. Применение алгоритмов с более высокими основаниями. При основании 2 базовая операция включает одно комплексное умножение и два комплексных сложения, а при основании 4 — три комплексных умножения и восемь комплексных сложений. Несложно построить аппаратуру, реализующую алгоритм с основанием 4 таким образом, чтобы все три умножения выполнялись одновременно. При этом быстродействие системы возрастет в четыре раза. Этот подход справедлив и для алгоритмов с более высокими основаниями, хотя сомнительно, что при основаниях, превышающих 8, затраты на дополнительное распараллеливание будут оправданы.

4. Поточное выполнение БПФ. Эта интересная специализированная структура, содержащая  $\log_2 N$  арифметических устройств (где  $r$  — основание алгоритма) и пригодная для построения си-

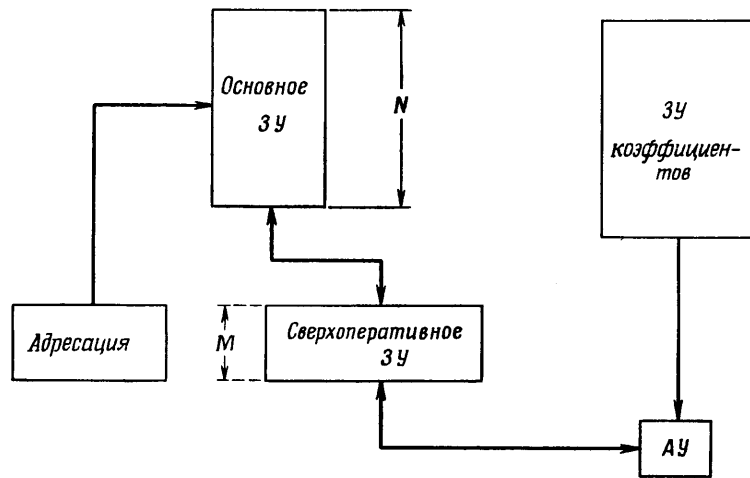
стем с очень большим быстродействием, будет детально рассмотрена в разд. 10.12 и 10.13.

5. Сверхпараллелизм. Для систем со сверхбольшим быстродействием эффективная структурная схема может содержать  $(N/2) \log_2 N$  параллельно работающих арифметических устройств. Это означает, что все преобразование реализуется в одной матричной схеме. Даже при умеренных размерах преобразования  $N$  для этого потребуется очень большой объем оборудования, однако, по крайней мере в радиолокации, такие устройства рассматриваются вполне серьезно.

Таким образом, замечательное свойство симметрии алгоритма БПФ дает возможность разработать множество структурных схем с самыми разнообразными уровнями параллелизма. Изучению этих схем и посвящены последующие разделы главы.

### 10.9. Выполнение БПФ с использованием сверхоперативной памяти

Во многих случаях применение ЗУ большого объема, по быстродействию согласующегося с АУ, обходится слишком дорого. Так, в разд. 10.6 (см. фиг. 10.19) было показано, что для ЗУ, оперирующего с комплексными числами, наилучшее согласование получается, если длительность цикла обращения к памяти  $\tau$  и время выполнения базовой операции относятся как 1:4. В настоящее время вполне реально создание АУ, в котором базовая операция



Фиг. 10.20. Применение сверхоперативного ЗУ для ускорения БПФ.

выполнялась бы за 400 нс, однако ЗУ с циклом обращения, равным 100 нс, будет стоить весьма дорого, особенно при большом объеме ЗУ (скажем, порядка 16 000 слов). В то же время, добавив к вычислительному устройству небольшое сверхоперативное ЗУ, можно без больших затрат существенно повысить скорость обработки. Одна из возможных схем такого типа приведена на фиг. 10.20.

Чтобы несколько упростить анализ, будем считать, что  $M = 2\sqrt{N}$  (фиг. 10.20). Если объем сверхоперативного ЗУ равен  $\sqrt{N}$ , то половину всех этапов БПФ, число которых равно  $\log_2 N$ , можно выполнить, вообще не обращаясь к основному ЗУ. Так, если  $N = 16$ , то из основного ЗУ можно взять пары отсчетов 0 и 8, 4 и 12, выполнить над этими парами отсчетов базовые операции и, по-прежнему оперируя только с числами, находящимися в сверхоперативном ЗУ, перейти к выполнению базовых операций второго этапа над парами отсчетов 0 и 4, 8 и 12. Затем можно выбрать отсчеты 1 и 9, 5 и 13 и повторить вычисления. Таким образом, для выполнения двух этапов БПФ над 16 отсчетами требуется одно считывание из основного ЗУ и одна запись в него. Без использования сверхоперативного ЗУ понадобилось бы проделать то же самое дважды. В табл. 10.2 сравнивается число обра-

Таблица 10.2

$N$	Со сверхоперативным ЗУ	Без сверхоперативного ЗУ
16	32	128
64	256	768
256	1 024	4 096
1024	6 144	20 480
4096	24 576	98 304

щений к основному ЗУ для различных  $N$  при отсутствии и наличии сверхоперативного ЗУ объемом в 16 слов (что соответствует  $M = 32$ , если нужно, чтобы операции, связанные с вычислениями и вводом-выводом, перекрывались во времени). Читателю предлагается следующее интересное задание. Пусть  $\tau = 100$  нс. Предложите комбинацию магнитного ЗУ и ЗУ на интегральных микросхемах, обеспечивающую эффективное выполнение ДПФ массивов размером от 256 до 16 384 точек. Для некоторых типичных значений быстродействия магнитного ЗУ (например, 1 мкс и 0,5 мкс) оцените зависимость характеристик устройства от объема сверхоперативного ЗУ.



На фиг. 10.6 приведен пример структуры алгоритма БПФ (для  $N = 16$  и  $M = 8$ ), соответствующего аппаратной реализации с помощью системы фиг. 10.20. Использование сверхоперативного ЗУ является полезным компромиссным решением задачи увеличения скорости выполнения БПФ без использования дорогостоящих быстродействующих ЗУ большого объема.

#### 10.10. Параллельные структуры для алгоритмов БПФ с основаниями 2 и 4, использующие ЗУ с произвольным доступом

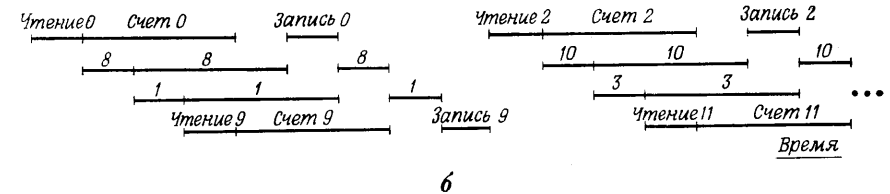
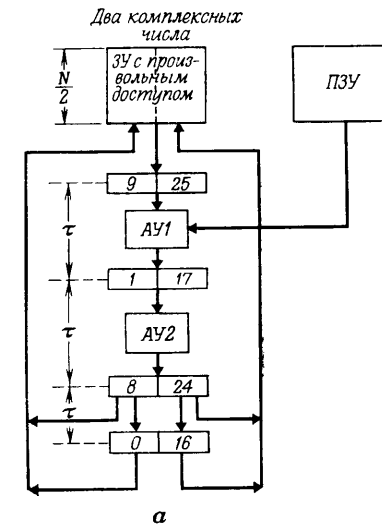
В данном разделе рассматриваются вычислительные структуры, в которых используется параллелизм со следующими отличительными особенностями:

1. Обращения к памяти и вычислительные операции полностью совмещаются во времени. Такая структура является согласованной в том смысле, что при ее проектировании и на ЗУ, и на АУ накладываются одинаковые ограничения.

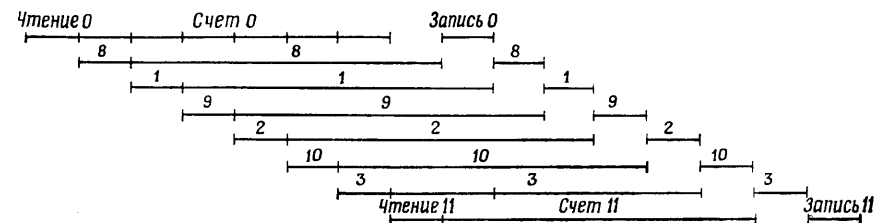
2. Числа хранятся в ЗУ с произвольным доступом, слова которого состоят из  $r$  комплексных чисел, где  $r$  — основание алгоритма БПФ, не изменяющееся в процессе счета.

Алгоритм с основанием 2, соответствующий такой структуре, иллюстрируется на фиг. 10.8. Блок-схема аппаратной реализации этого алгоритма и соответствующие временные диаграммы представлены на фиг. 10.21. ЗУ с произвольным доступом состоит из  $N/2$  регистров, каждый из которых рассчитан на два комплексных числа. Номера регистров ЗУ соответствуют их адресам. Чтобы система была согласованной, нужно, чтобы циклы считывания следовали друг за другом без пауз. Поэтому последовательные циклы считывания должны перекрываться с циклами вычислительных операций. Более того, поскольку запись результатов возможна лишь после завершения по крайней мере двух циклов вычислений, из временной диаграммы (фиг. 10.21) видно, что записи должны предшествовать четыре последовательных цикла считывания. Отсюда следует, что АУ должно иметь поточную схему с тем, чтобы выполнение второй базовой операции начиналось до завершения первой.

Если АУ не успевает выполнить свои функции за два цикла обращения к ЗУ, можно применить схему с более продолжительной поточной обработкой, занимающей шесть циклов обращения к ЗУ, причем схема по-прежнему остается согласованной. Временные диаграммы ее работы приведены на фиг. 10.22. Такая схема представляет собой расширенный вариант схемы фиг. 10.21, а, содержащий шесть буферных регистров, включенных между шестью элементарными арифметическими устройствами. Если вычисления могут выполняться быстрее, некоторые из элементарных АУ фактически будут отсутствовать. Они будут обеспечивать лишь соединение соседних буферных регистров.

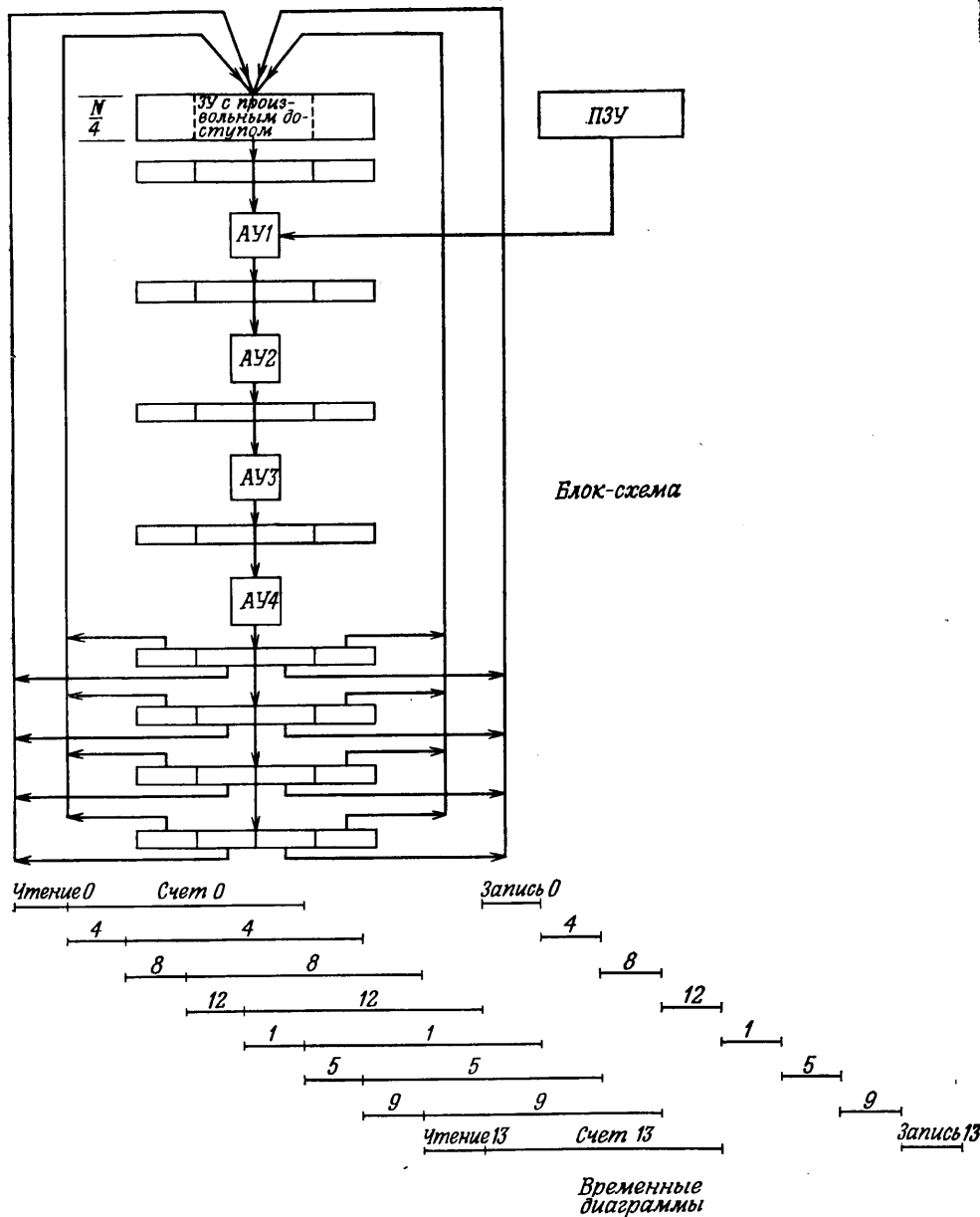


Фиг. 10.21. Блок-схема и временные диаграммы устройства БПФ с основанием 2 и регистрами двойной длины.



Фиг. 10.22. Другой вариант временных диаграмм для схемы фиг. 10.21.

Предлагаем читателю самому разработать структурную схему и построить временные диаграммы для случая, когда одно АУ работает совместно с двумя ЗУ с произвольным доступом.



Фиг. 10.23. Блок-схема параллельного устройства с основанием 4 и соответствующие временные диаграммы.

На фиг. 10.23 изображены блок-схема реализации алгоритма БПФ с основанием 4 и ее временные диаграммы. Время счета здесь равно четырем циклам обращения к памяти; сначала выполняются подряд восемь циклов чтения из ЗУ, а за ними следуют восемь циклов записи. Вычисления, проводимые поточным методом, завершаются матричной перестановкой ( $4 \times 4$ ) результатов, выполняемой с помощью четырех регистров, показанных на фиг. 10.23, причем каждый из регистров содержит результат базовой операции алгоритма БПФ с основанием 4.

Метод использования параллелизма, представленный в данном разделе, основан главным образом на идее согласования длительности обращения к памяти и продолжительности выполнения базовой операции. Предполагалось, что используется алгоритм с постоянным основанием и что при основании  $r$  создается  $r$ -кратный параллелизм. Алгоритм с основанием  $r$  состоит из  $\log_r N$  этапов. На каждом из них ко всем  $N/r$  регистрам приходится обращаться дважды — для считывания входных чисел базовой операции и для записи ее результатов. Таким образом, количество циклов обращения к памяти, затрачиваемых на выполнение БПФ, составляет

$$C_r = \frac{2N}{r} \log_r N, \quad (10.26)$$

так что на один входной отсчет приходится

$$\frac{C_r}{N} = \frac{2}{r} \log_r N \quad (10.27)$$

циклов. Соотношение (10.27) определяет наибольшую частоту дискретизации, при которой возможна обработка в реальном времени, если известна длительность выполнения базовой операции или цикла обращения к памяти. Например, если  $N = 1024$  и  $r = 2$ , то  $C_r/N = 10$ ; таким образом, при длительности выполнения базовой операции в 100 нс можно обрабатывать сигнал, дискретизуемый с частотой 1 МГц.

### 10.11. Общие вопросы построения поточных схем БПФ

Если еще раз обратиться к схемам фиг. 10.1—10.8, то можно заметить, что, хотя они и описывают многие свойства алгоритма, последовательность выполнения базовых операций во времени все еще остается неопределенной. В самом деле, существует много различных последовательностей, которые приводят к одинаковым результатам. Так, на первом этапе алгоритма фиг. 10.1 пары отсчетов 0 и 8, 1 и 9 и т. д. можно обрабатывать в любом порядке. Это же справедливо и для остальных этапов. Достоинством некоторых вариантов последовательностей выполнения операций может оказаться легкость аппаратной реализации или программирова-

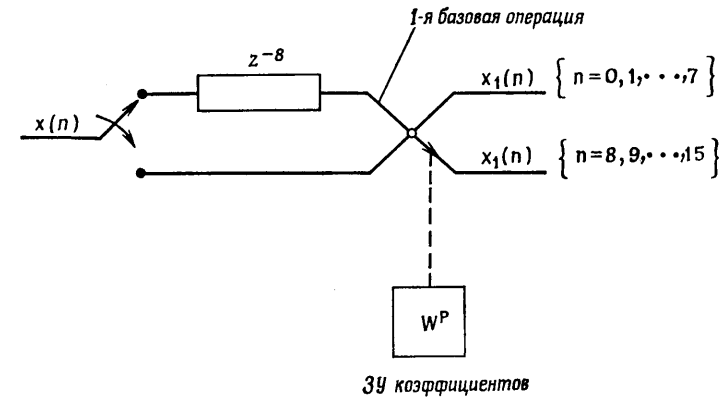
ния, но сама структура алгоритма принципиальных ограничений не вносит. Так, для перехода ко второму этапу вовсе не обязательно, чтобы первый этап был завершен. Действительно, если в начале первого этапа обрабатываются отсчеты 0 и 8, а затем 4 и 12, то после этого уже можно начать выполнение второго этапа.

Отметим также, что сами алгоритмы не накладывают никаких ограничений на уровень параллелизма при аппаратурной реализации БПФ. Однако при аппаратурной реализации БПФ уровень параллелизма накладывает определенные ограничения на последовательность выполнений базовых операций. В последующих разделах будет рассмотрен особый класс схем с параллельным выполнением операций, называемых *поточными схемами БПФ*, для которых уровень параллелизма равен  $\log_r N$ . При построении поточной схемы БПФ с основанием  $r$  используются  $\log_r N$  арифметических устройств, параллельно выполняющих базовые операции.

Чтобы проиллюстрировать уровень параллелизма в поточной схеме БПФ, рассмотрим в качестве примера БПФ с основанием 2 массива из 1024 отсчетов, которое выполняется за 10 этапов. В большинстве универсальных ЦВМ имеется только одно устройство умножения. В поточной схеме БПФ для выполнения базовых операций может быть использовано 10 независимых арифметических устройств, что соответствует 40 умножителям действительных чисел (так как каждая базовая операция включает комплексное умножение, состоящее из четырех действительных умножений). Таким образом, если предположить, что эффективность поточной схемы БПФ совпадает с эффективностью универсальной ЦВМ, запрограммированной на выполнение алгоритма БПФ, то скорость выполнения БПФ в поточном устройстве будет в 40 раз больше, чем в универсальной ЦВМ. Оказалось, что в поточной схеме БПФ аппаратура используется в 2—20 раз эффективнее, чем при выполнении БПФ на любой из известных универсальных ЦВМ. Таким образом, в поточной схеме алгоритм БПФ выполняется на 2—3 порядка быстрее. Из-за большой эффективности и относительной простоты управления поточные схемы БПФ представляются сейчас наиболее перспективными при построении специализированных высокопроизводительных процессоров БПФ.

### 10.12. Поточные схемы БПФ с основанием 2

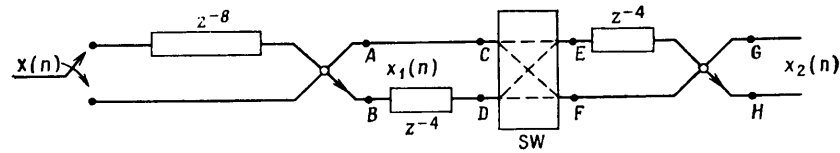
Рассмотрим сначала, как, имея  $\log_2 N$  параллельно работающих арифметических устройств, наиболее эффективно реализовать алгоритмы БПФ, аналогичные приведенному на фиг. 10.1. Количественной мерой *эффективности* может служить доля времени, в течение которого АУ заняты выполнением базовых операций.



Фиг. 10.24. Поточная схема выполнения первого этапа БПФ.

Предположим пока, что отсчеты сигнала  $x(0)$ ,  $x(1)$  и т. д. поступают на вход последовательно. Для этого случая на фиг. 10.24 приведена очень простая блок-схема выполнения первого этапа алгоритма БПФ, соответствующего, например, схеме на фиг. 10.1. Первые восемь отсчетов  $x(0) - x(7)$  вводятся в восьмикаскадную схему задержки  $z^{-8}$ . Следующие восемь отсчетов подаются на второй вход системы. Если длительность базовой операции *в точности* равна интервалу дискретизации, то весь первый этап БПФ будет выполнен за восемь таких интервалов, следующих после момента переключения. Результаты первого этапа [обозначим их через  $x_1(n)$ ] будут появляться на выходе блока базовой операции парами. Поскольку коэффициенты  $W^p$  изменяются от отсчета к отсчету, 3У коэффициентов должно выдавать их с частотой поступления отсчетов сигнала. Из фиг. 10.1 видно, что на втором этапе структурная форма первого этапа повторяется дважды. Следовательно, на втором этапе необходимо использовать устройство, обрабатывающее отсчеты  $x_1(n)\{n=0, 1, \dots, 7\}$  и  $x_1(n)\{n=8, 9, \dots, 15\}$ , аналогично тому, как обрабатывались отсчеты  $x(n)\{n=0, 1, \dots, 15\}$ . Соответствующая блок-схема изображена на фиг. 10.25. С помощью задержек и переключений частично обработанные отсчеты выстраиваются так, как это требуется в алгоритме фиг. 10.1. Временное разнесение между отсчетами, обрабатываемыми совместно, составляет восемь интервалов дискретизации на первом этапе и четыре — на втором. Полная поточная схема 16-точечного БПФ изображена на фиг. 10.26. Анализируя эту схему, можно заметить определенную симметрию и, экстраполируя, получить поточные схемы БПФ для больших  $N$ . Сделаем несколько замечаний относительно схемы фиг. 10.26.

1. От этапа к этапу задержка отсчетов уменьшается вдвое.



A	0 1 2 3 4 5 6 7	...	$x_1(n)$
B	8 9 10 11 12 13 14 15		
C	0 1 2 3 4 5 6 7		
D	8 9 10 11 12 13 14 15		
SW	$C \rightarrow E, D \rightarrow F$ $C \rightarrow E, D \rightarrow F$ <i>Прямо</i> $C \rightarrow F, D \rightarrow E$ $C \rightarrow F, D \rightarrow E$ <i>Накрест</i>		
G	0 1 2 3 8 9 10 11	...	$x_2(n)$
H	4 5 6 7 12 13 14 15		

Фиг. 10.25. Поточная схема выполнения двух первых этапов 16-точечного БПФ по основанию 2 с прореживанием по частоте.

2. Арифметические устройства на всех приведенных схемах заняты только половину времени.

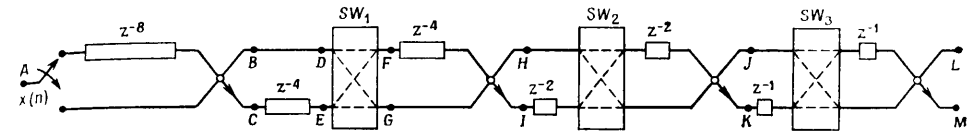
3. Частота переключений от этапа к этапу увеличивается вдвое.

4. Период синхронизации системы равен, очевидно, интервалу дискретизации.

5. Отсчеты на выходе появляются во времени в двоично-инверсном порядке.

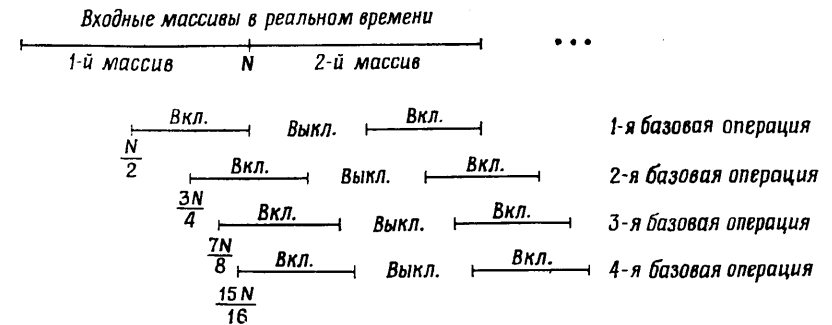
Для доказательства последнего положения отметим, что номера на фиг. 10.26 совпадают с (необозначенными) номерами регистров в схеме на фиг. 10.1. Так как в схеме фиг. 10.1 выходная последовательность расположена в двоично-инверсном порядке, то и в схеме фиг. 10.26 будет такой же порядок. Короче говоря, данная схема представляет собой один из возможных вариантов аппаратной реализации алгоритма фиг. 10.1. Поэтому преобразование, выполняемое в этой схеме, обладает всеми свойствами указанного алгоритма и, кроме того, имеет специфические временные характеристики, не указанные на фиг. 10.1. Последнее замечание можно подтвердить тем, что в поточной схеме БПФ имеются два выхода, на которых одновременно появляются две гармоники спектра. Важно отметить, что числа, приведенные в последних двух строках фиг. 10.26, фактически представляют собой двоично-инверсные номера выходных гармоник.

Доказательство положения 2 является довольно сложным, так как время работы арифметических устройств зависит от метода ввода данных в процессор. Потребуем, например, чтобы данные

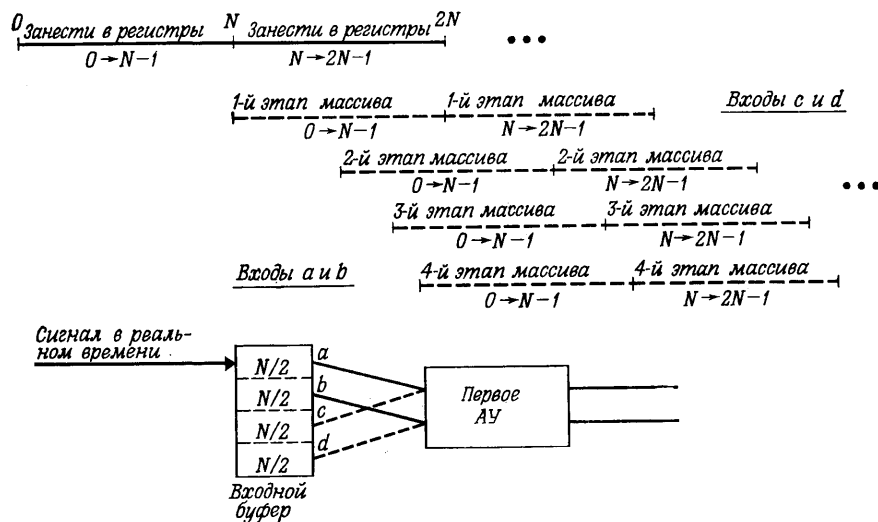


A	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	...	$x_1(n)$
B	0 1 2 3 4 5 6 7	...	$x_1(n)$
C	8 9 10 11 12 13 14 15		
D	0 1 2 3 4 5 6 7		
E	8 9 10 11 12 13 14 15		
SW <sub>1</sub>	$D \rightarrow F, E \rightarrow G$ $D \rightarrow F, E \rightarrow G$ <i>Прямо</i> $D \rightarrow G, E \rightarrow F$ $D \rightarrow G, E \rightarrow F$ <i>Накрест</i>		
F	0 1 2 3 8 9 10 11		
G	4 5 6 7 12 13 14 15		
H	0 1 2 3 8 9 10 11	...	$x_2(n)$
I	4 5 6 7 12 13 14 15		
SW <sub>2</sub>	 <i>Прямо</i> <i>Накрест</i>		
J	0 1 4 5 8 9 12 13	...	$x_3(n)$
K	2 3 6 7 10 11 14 15		
SW <sub>3</sub>	 <i>Прямо</i> <i>Накрест</i>		
L	0 2 4 6 8 10 12 14	...	$x_4(n) = x(k)$
M	1 3 5 7 9 11 13 15		

Фиг. 10.26. Полная поточная схема выполнения 16-точечного БПФ по основанию 2 с прореживанием по частоте.

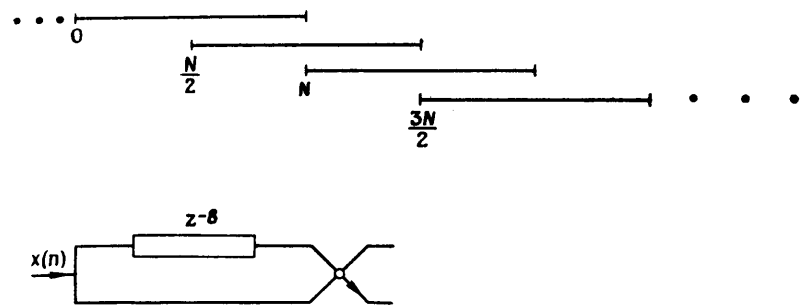


Фиг. 10.27. Временные диаграммы работы арифметических устройств при обработке последовательных массивов.



Фиг. 10.28. Включение входного буферного накопителя, позволяющего обрабатывать последовательные массивы в реальном времени со 100%-ной эффективностью.

обрабатывались в виде смежных массивов в реальном времени (фиг. 10.27). Согласно схемам, представленным на фиг. 10.24 — 10.26, обработка не может быть начата до тех пор, пока в процессор не поступит половина всех обрабатываемых отсчетов. После этого первый этап завершается за последующие  $N/2$  циклов. Затем выполнение базовых операций первого этапа прекращается до тех пор, пока первые  $N/2$  отсчетов следующего массива не заполнят схему задержки  $z^{-8}$ . В остальных АУ происходит то же самое, но с некоторым запаздыванием. Следовательно, эффективность поточного процессора равна 50%, так как каждое из АУ простаивает точно половину времени. На фиг. 10.28 показано, каким образом с помощью входной буферной памяти эффективность процессора может быть доведена до 100%. После полного накопления первого массива отсчеты с выходов  $a$  и  $b$  начинают одновременно поступать в процессор. Из-за наличия двух параллельных линий ввода тактовая частота процессора может быть в два раза меньше частоты дискретизации входного сигнала. Таким образом, первый этап БПФ завершается как раз к тому моменту, когда следующий массив уже поступил в буфер и полностью готов к обработке. Остальные этапы БПФ выполняются аналогично, но с запаздыванием, характерным для поточных схем. Преимуществом этого эффективного метода является возможность понижения тактовой частоты процессора в два раза или же возможность об-



Фиг. 10.29. Входная часть устройства обработки в реальном времени перекрывающихся массивов.

работки данных, следующих с удвоенной частотой, если используется система фиг. 10.26. Все это достигается введением входного буферного накопителя и соответствующей коммутации.

В интересном частном случае обработки в реальном времени, когда обрабатываемые массивы наполовину перекрываются (фиг. 10.29), входная линия просто подключается к устройству задержки  $z^{-8}$ , и к первому АУ. Как и в предыдущем случае (фиг. 10.28), эффективность системы будет равна 100%, так как все АУ будут работать без перерывов. Данный частный случай хорошо согласуется с алгоритмом вычисления свертки методом БПФ и поэтому является весьма полезным.

В заключение несколько уточним замечания, сделанные ранее относительно схемы фиг. 10.26. Замечание 1 в целом справедливо, но изменения, связанные с накоплением входных отсчетов, влияют на способ создания задержки на первом этапе. Например, в схеме фиг. 10.28 эта задержка создается с помощью входного буферного накопителя. Замечание 2 может не выполняться, так как существуют схемы с полной загрузкой АУ (см. фиг. 10.28, 10.29). Замечание 3 справедливо для всех этапов, за исключением, возможно, первого, причем тактовая частота процессора может быть понижена (см. схему на фиг. 10.28) по сравнению с частотой дискретизации. Во всех рассмотренных выше схемах выходные гармоники следуют в двоично-инверсном порядке и рассчитываются согласно алгоритму фиг. 10.1. Существуют и другие варианты алгоритма БПФ с основанием 2, для которых также можно построить поточные схемы БПФ, но к настоящему времени они представляют не столь изящными и более громоздкими.

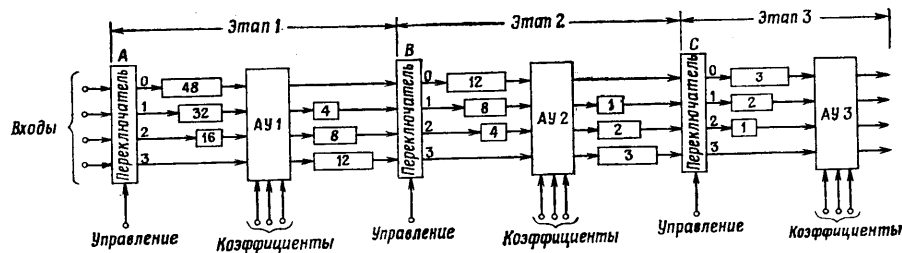
Отметим также, что при анализе схемы фиг. 10.26 специального времени для выполнения арифметических операций отведено не было. Учет этого времени никак не повлияет на общую структуру схемы, но приведет к замедлению работы процессора на число так-

тов, необходимых для выполнения базовой операции. Если это число больше 1, то и сами АУ должны строиться по поточной схеме.

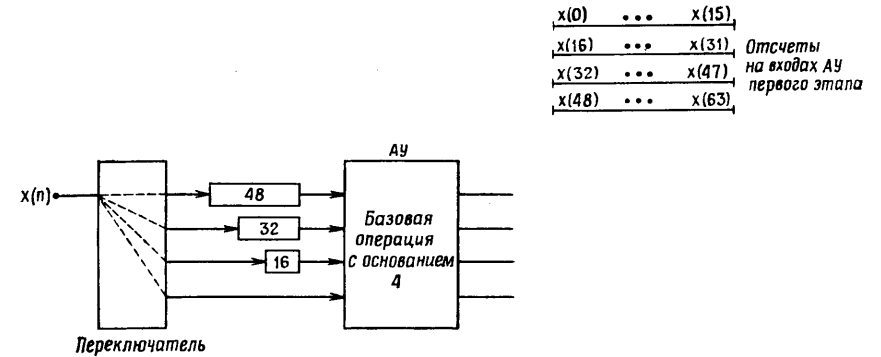
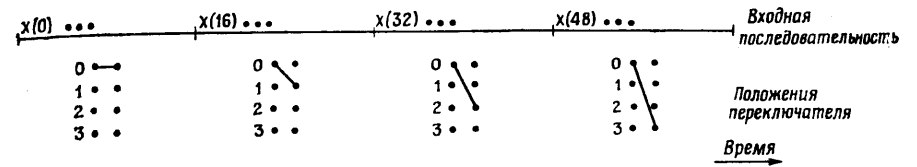
### 10.13. Поточная схема БПФ с основанием 4

Основываясь на алгоритме фиг. 10.14, можно разработать поточную схему для выполнения 64-точечного БПФ с основанием 4. Рассмотрим сначала случай обработки единственного массива из 64 отсчетов, следующих в нормальном порядке. Оказывается, что поточная схема с основанием 4 является в этом случае весьма неэффективной, так как арифметические устройства будут заняты только четверть всего времени. Тем не менее данный пример позволяет проанализировать всю интересующую нас структуру, так как многие выводы будут справедливы и для систем со 100%-ной эффективностью. Достижение такой эффективности фактически связано с решением задачи накопления входных отсчетов в буферном ЗУ, рассматриваемой ниже применительно к различным вариантам ввода обрабатываемых данных. На фиг. 10.30 представлена блок-схема поточного процессора БПФ с основанием 4. В целом она аналогична схеме, соответствующей основанию 2, но все основные блоки (АУ, схемы задержки и коммутаторы) рассчитаны на выполнение алгоритма БПФ с основанием 4. Так, в АУ выполняются не два комплексных сложения и одно комплексное умножение, как при основании 2, а восемь комплексных сложений и три комплексных умножения. Коммутатор имеет четыре входа и четыре выхода, причем в трех из четырех линий передачи данных имеются элементы задержки.

На фиг. 10.31 показано, каким образом входная последовательность  $x(0), x(1), \dots, x(63)$  распределяется между четырьмя входами первого АУ. Исходные отсчеты изображаются отрезками вверху. Через каждые 16 тактов коммутатор переходит в следующее положение, поэтому последовательные группы из 16 отсчетов поступают в АУ с неодинаковыми задержками. Начиная с 48-го такта, отсчеты  $x(0), x(16), x(32)$  и  $x(48)$  одновременно появляются на



Фиг. 10.30. Поточная схема выполнения 64-точечного БПФ с основанием 4.

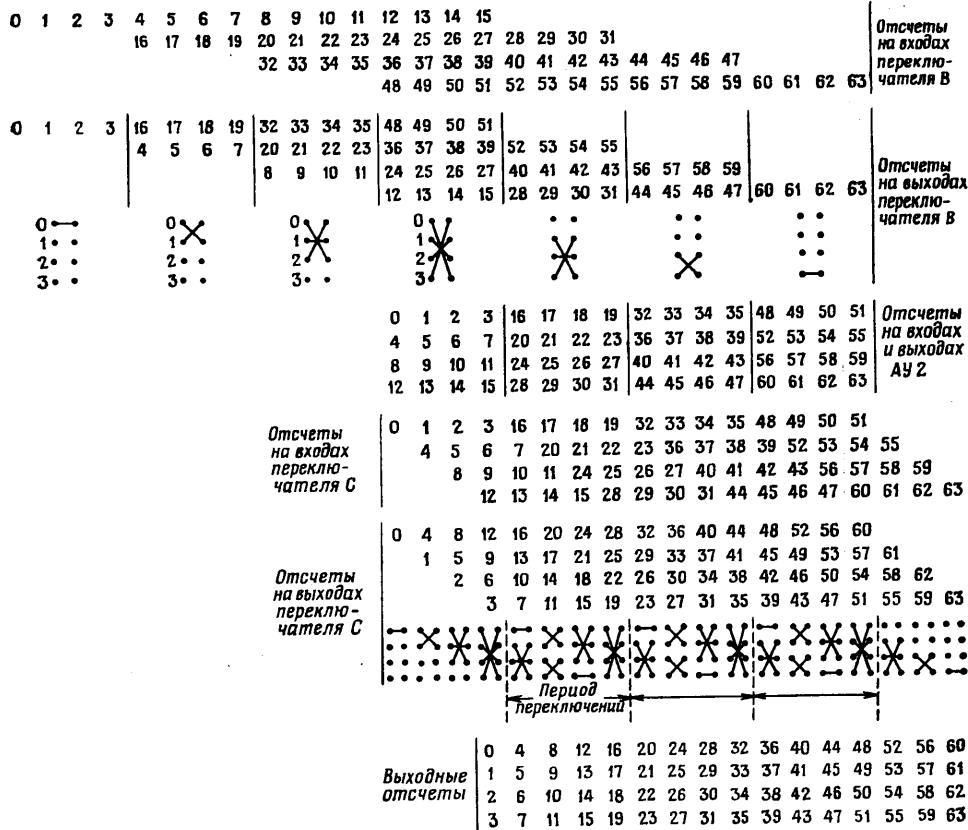


Фиг. 10.31. Разделение массива из 64 отсчетов на 4 подмассива по 16 отсчетов при выполнении базовой операции с основанием 4.

входе первого АУ, а после 63-го такта операции первого этапа алгоритма фиг. 10.14 заканчиваются. Отметим, что АУ было занято только в течение 16 тактов из 64.

Выполнение остальных операций алгоритма можно проследить с помощью фиг. 10.32. Каждая из четырех параллельно обрабатываемых последовательностей проходит через все АУ. Отметим, что во всех коммутаторах переключения производятся по одинаковому графику. Проще всего это проследить на примере коммутатора С, для которого показаны положения всех переключателей. «Цикл переключения» фактически представляет собой последовательность из четырех различных состояний коммутатора. Эта последовательность периодична. Начальный и конечный участки графика легко отождествить с основной последовательностью из четырех состояний, заметив, что из-за краевых эффектов некоторые состояния коммутатора не используются. Работа коммутатора В в действительности описывается тем же графиком, только в данном частном случае краевые эффекты преобладают. Отсюда можно заключить, что в установленном режиме, когда АУ работают без пауз, все коммутаторы повторяют цикл переключений, изображенный для коммутатора С, но каждый последующий коммутатор переключается в четыре раза чаще предыдущего.

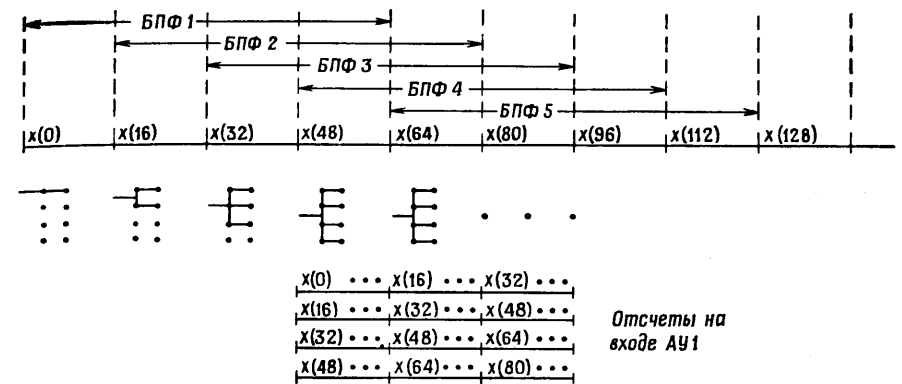
Отметим также, что отсчеты на выходе появляются во времени в том же порядке, в каком они поступали на вход, а это означает,



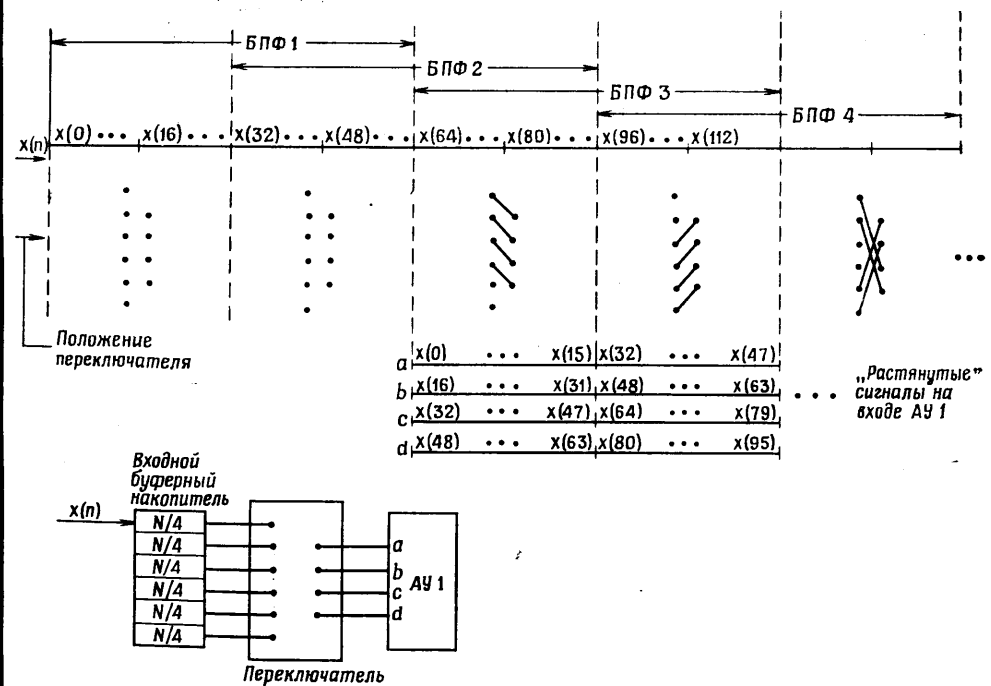
Фиг. 10.32. Временные диаграммы и работа переключателей в поточном блоке 64-точечного БПФ с основанием 4 (см. фиг. 10.14).

что по частоте гармоники будут расположены в четверично-инверсном порядке (см. разд. 10.3).

Порядок переключений при выполнении БПФ четырех перекрывающихся массивов показан на фиг. 10.33. В установившемся режиме одна и та же последовательность отсчетов, поступающая в реальном времени, всегда подается на все четыре линии системы. Чтобы избежать краевых эффектов, первые три подпоследовательности подключаются постепенно. Но по достижении установившегося режима коммутатор А фактически становится не нужен, так как входная последовательность постоянно подается на все четыре линии системы. Если же для удобства потребовать, чтобы коммутатор А переключался стандартным образом (согласно



Фиг. 10.33. Одновременное выполнение четырех БПФ в поточном блоке БПФ с основанием 4.



Фиг. 10.34. Вариант входной части устройства, позволяющий получить 100%-ную эффективность при двукратном перекрытии сигналов.

фиг. 10.32), то обработка с перекрытием будет обеспечиваться, если входную последовательность подавать сразу на все четыре входа коммутатора А. Представляется естественным, что система с основанием 4 имеет 100%-ную эффективность при выполнении БПФ массивов с четырехкратным перекрытием, однако неясно, можно ли сохранить эту эффективность при обработке в реальном времени массивов с двукратным перекрытием. На этот вопрос можно ответить утвердительно, но входные буферный накопитель и коммутатор должны быть при этом несколько видоизменены. Соответствующая схема приведена на фиг. 10.34. ЗУ, использовавшееся при выполнении первого этапа в схеме фиг. 10.30, заменено здесь блоком памяти объемом  $3N/2$  с одним входом и шестью выходами. В течение первых 64 тактов первые  $N$  отсчетов последовательно поступают во входной буферный накопитель, но не обрабатываются. Начиная с отсчета с номером 64, коммутатор устанавливается так, что верхние четыре выхода буферного накопителя подключаются к АУ1, и с них с частотой, в два раза меньшей частоты следования входных отсчетов, начинают поступать данные для обработки. Таким образом, пока на 32-тактном интервале времени в буферный накопитель вводятся два новых подмассива, четыре уже накопленных подмассива считываются в АУ1. Сле-

довательно, 100%-ная эффективность поддерживается за счет двукратного «растягивания» сигнала. Вообще следует ожидать, что по сравнению со случаем четырехкратного перекрытия здесь за единицу времени можно осуществить вдвое больше операций по обработке сигнала. В качестве упражнения предлагаем составить схему буферного накопителя для случая, когда обрабатываемые массивы переключаются только на  $N/4$  отсчетах, т. е. первый массив содержит отсчеты с 0-го по  $(N-1)$ -й, второй — отсчеты с  $3N/4$ -го по  $(7N/4-1)$ -й и т. д., так что выполнение каждого нового БПФ по  $N$  отсчетам начинается через  $3N/4$  тактов.

С помощью поточной схемы БПФ с основанием 4 можно найти БПФ четырех различных сигналов:  $x_1(n)$ ,  $x_2(n)$ ,  $x_3(n)$  и  $x_4(n)$ . Для этого последовательность переключений коммутатора на первом этапе должна быть такой же, как у коммутатора на следующем этапе. Соответствующая блок-схема приведена на фиг. 10.35. Заметим, что входные последовательности задерживаются относительно друг друга на  $N/4$  отсчетов. Начиная с четвертого интервала длиной в  $N/4$  отсчетов, режим переключений устанавливается и повторяется через каждые  $N$  отсчетов. До известной степени обработка четырех независимых сигналов является для поточной системы с основанием 4 наиболее естественным режимом, потому что графики переключений во всех коммутаторах (включая и первый) становятся одинаковыми с той лишь разницей, что каждый последующий коммутатор переключается в четыре раза чаще.

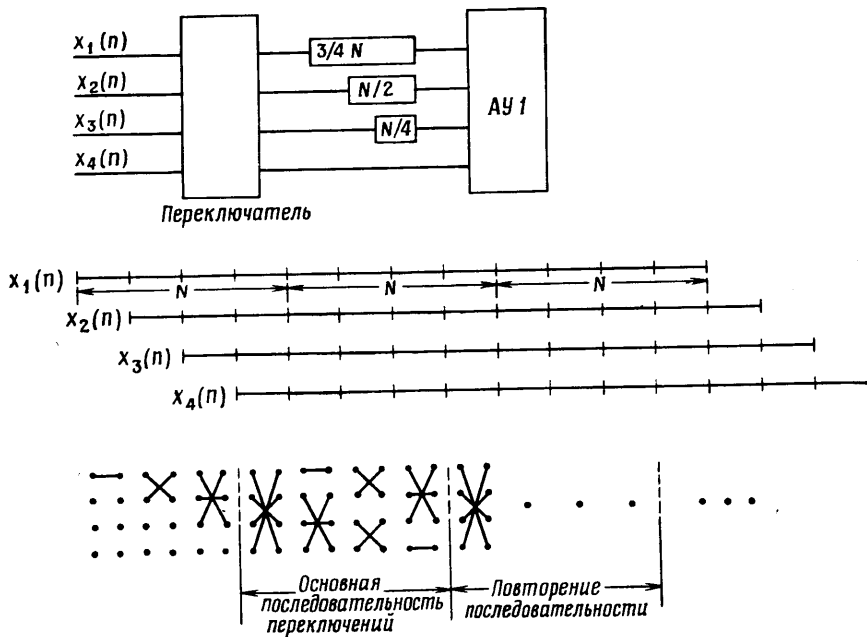
#### 10.14. Сравнение поточных схем БПФ с основаниями 2 и 4

Выше уже было показано, что для выполнения БПФ одинакового размера при основании 4 требуется меньше умножителей, чем при основании 2. Исследуем этот вопрос для частного случая поточной схемы БПФ. При данном  $N$  в схеме с основанием 2 проводится  $\log_2 N$  этапов, а в схеме с основанием 4 —  $\log_4 N$  этапов. В первом случае этапов вдвое больше, но на каждом из них нужно использовать по два комплексных сумматора и по одному комплексному умножителю, тогда как для каждого этапа второй схемы требуются восемь комплексных сумматоров и три умножителя. Таким образом,

$$r_u = \frac{\text{Число умножителей при основании 4}}{\text{Число умножителей при основании 2}} = \frac{3}{2},$$

$$r_c = \frac{\text{Число сумматоров при основании 4}}{\text{Число сумматоров при основании 2}} = 2.$$

Из вышеизложенного может показаться, что схема с основанием 2 лучше, однако проведенное сравнение некорректно, так как при указанных отношениях схема с основанием 4 позволяет обработать вдвое больше данных. Если пронормировать  $r_u$  и  $r_c$  так, что-



Фиг. 10.35. Входная часть поточной схемы БПФ с основанием 4 для обработки четырех независимых сигналов.



бы они относились к одинаковой пропускной способности, то получим

$$\hat{r}_y \left( \text{нормированное } r_y \right) = \frac{3}{4},$$

$$\hat{r}_c = 1.$$

Отсюда видно, что в схеме с основанием 4 для обработки такого же потока данных требуется меньше умножителей. Отметим, однако, что здесь не сравнивались числа элементов, необходимых для управления работой обеих схем.

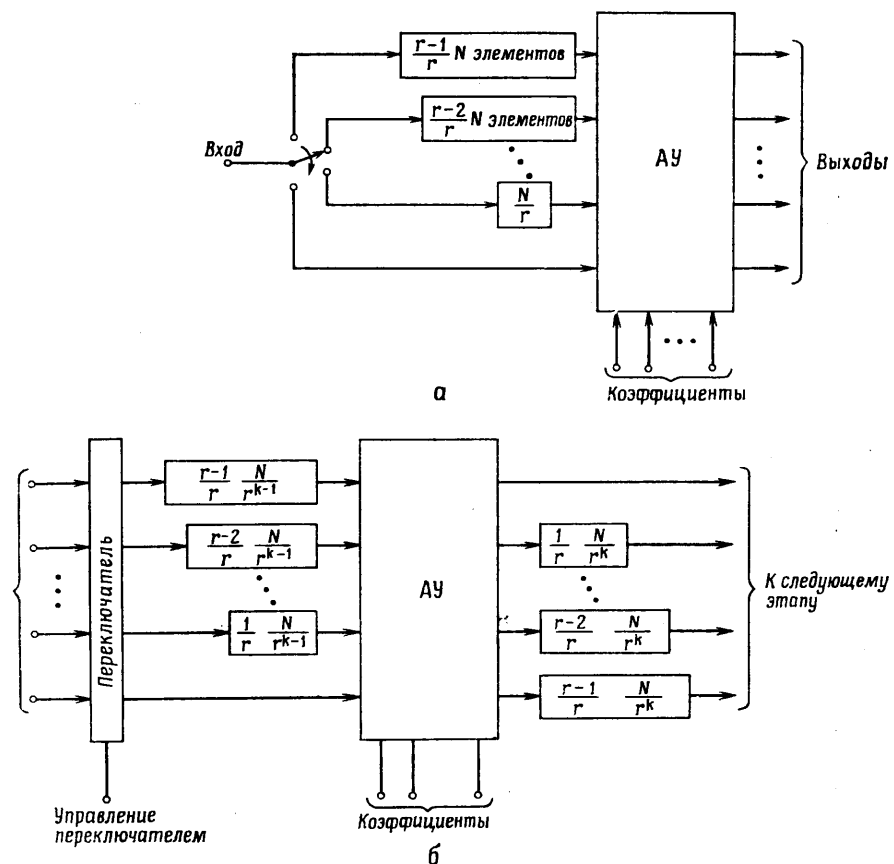
### 10.15. Структуры БПФ с повышенным уровнем параллелизма

Выше было принято, что эффективность схемы БПФ равна 100%, если все ее арифметические устройства выполняют базовые операции без перерывов. В рассмотренных поточных структурах эффективность можно довести до 100% с помощью входного буферного накопителя, согласованного с конкретными обрабатываемыми сигналами. В отличие от этих схем универсальные ЦВМ, работа которых основана на последовательном выполнении операций, будут обычно иметь существенно меньшую эффективность. В данном разделе будут описаны схемы БПФ, также обеспечивающие 100%-ную эффективность, но при более высоком уровне параллелизма, чем в рассмотренных ранее схемах.

1. Естественным обобщением поточных схем БПФ с основаниями 2 и 4 являются схемы с основанием  $r$ . Для данного  $r$  (например,  $r = 8$ ) всегда можно построить поточную схему, как показано на фиг. 10.36. Схема имеет  $r$  линий передачи данных, в которые по обе стороны от АУ включено по  $(r - 1)$  схем задержки [см. фиг. 10.36, б]. Арифметическое устройство предназначено для выполнения  $r$ -точечного ДПФ с последующим умножением на  $(r - 1)$  поворачивающих множителей. Сравним теперь три поточных схемы БПФ с основаниями 2, 4 и 8. Если тактовая частота во всех случаях одинакова, то схема с более высоким основанием сможет обработать больше данных, так как в поточной схеме

Таблица 10.3

N	Основание 2		Основание 4		Основание 8	
	число умножений	число сложений	число умножений	число сложений	число умножений	число сложений
64	16	32	24	60	44	118
4096	40	68	60	126	116	144



Фиг. 10.36. Поточная блок-схема выполнения 1-го и  $k$ -го этапов БПФ с основанием  $r$ .

БПФ с основанием  $r$  данные обрабатываются сразу по  $r$  линиям. В табл. 10.3 для некоторых частных случаев сопоставлены уровни параллелизма в схемах с основаниями 2, 4 и 8. Здесь указано фактическое количество умножителей и сумматоров действительных чисел. Комплексное умножение, например, эквивалентно четырем умножениям и двум сложениям действительных чисел. Напомним также, что при основании 4 схема в принципе может обеспечить вдвое большее быстродействие, чем при основании 2, а при основании 8 — вдвое большее, чем при основании 4 (если только тактовая частота во всех случаях одинакова). (Это означает, что АУ схемы с основанием 8 должно обеспечить выполнение восьмиточечного ДПФ и семи умножений на поворачивающие множи-

тели за то же время, за которое АУ схемы с основанием 4 выполняют четырехточечное ДПФ и три поворота, а АУ схемы с основанием 2 — двухточечное ДПФ и один поворот.)

2. Для расширения возможностей системы было бы желательно увеличить уровень параллелизма, введя в нее не  $\log_2 N$ , и даже не  $\log_2 N$ , а  $N/2$  параллельно работающих АУ. Эффективный алгоритм для такой схемы, представляющий 16-точечное БПФ с постоянной структурой, иллюстрируется на фиг. 10.3. Преимуществом данной схемы является малое число переключений, необходимое при пересылке чисел из АУ в регистры памяти и обратно. Каждое АУ принимает данные из двух (комплексных) регистров и засылает результаты в два других регистра. Эти взаимосвязи на всех этапах БПФ остаются постоянными. Результаты обработки будут получены в двоично-инверсном порядке, но за счет введения двух дополнительных соединений на каждое АУ перестановку можно устранить.

Вероятно, наиболее быстрый и экономичный метод введения коэффициентов заключается в том, чтобы хранить их в определенной последовательности в каждом из АУ. Так, для алгоритма фиг. 10.3 в АУ7 используются коэффициенты  $W^7$ ,  $W^6$ ,  $W^4$  и  $W^0$ . В любом АУ для хранения коэффициентов требуется не более  $\log_2 N$  регистров, поэтому для всех  $N/2$  АУ потребуется  $(N/2)\log_2 N$  или меньше регистров. Хотя эта величина существенно превышает необходимое число коэффициентов, равное  $N/2$ , важно исключить операцию распределения коэффициентов из центрального ЗУ по арифметическим устройствам.

Чтобы оценить быстродействие рассматриваемой системы, представим, что базовая операция выполняется за  $\tau$  секунд. Тогда для выполнения всего БПФ потребуется  $\tau \log_2 N$  секунд.

3. Имея полный набор из  $(N/2)\log_2 N$  АУ, можно увеличить быстродействие еще в  $\log_2 N$  раз, если использовать поточную схему для реализации алгоритма фиг. 10.3. Тогда каждый из незачерненных кружков будет соответствовать целому АУ, а каждая точка — регистру. Такая структура является предельной, так как матрица АУ и поточная организация обработки обеспечивают получение результатов БПФ с тактовой частотой.

В такой схеме в каждом из АУ хранится *только один* коэффициент, причем часто он равен  $\pm 1$  или  $\pm j$ , так что умножитель в таком АУ просто не нужен. С помощью формулы (10.2) можно рассчитать, сколько умножителей фактически требуется для получения  $N$ -точечного БПФ. В табл. 10.4 приведено это количество умножителей  $M_N$  при разных  $N$  для матричной схемы выполнения БПФ по основанию 2 с поточной обработкой и предельным уровнем параллелизма.

4. Если необходимо выполнять БПФ при больших  $N$  (например, при  $N = 4096$ ), то использование схем, описанных в пп. 2 и 3,

Таблица 10.4

$N$	2	4	8	16	32	64
$M_N$	0	0	2	10	34	98

потребуется огромного количества оборудования, а именно 2048 арифметических устройств в первом случае и 21 502 — во втором. С другой стороны, в поточной схеме с основанием 2 используются 12 АУ. По количеству используемой аппаратуры схема поточного БПФ с основанием 8 ( $N = 4096$ ) эквивалентна 48 АУ схемы с основанием 2. Существует еще одна структура, занимающая промежуточное положение между системой с основанием 2 и предельной системой. В ней используется специально организованный блок памяти, с помощью которого входной массив представляется в виде двумерной матрицы и обрабатывается сначала по строкам, а затем по столбцам. При  $N = 4096$ , например, матрица имеет размер  $(64 \times 64)$ , так что в системе параллельно работают 32 АУ и уровень параллелизма почти втрое больше, чем в поточной системе с основанием 2.

Требуемое двумерное ЗУ можно построить на серийно выпускаемых микросхемах памяти. Принцип его построения иллюстрируется на фиг. 10.37 для матрицы из  $(4 \times 4)$  элементов. Если входные отсчеты имеют номера 0—15, как показано на фиг. 10.37, а, то весь массив можно разместить в четырех микросхемах ЗУ (каждая из которых содержит по четыре слова) согласно фиг. 10.37, б. При необходимости прочитать (или записать)  $j$ -ю строку или  $j$ -й столбец все четыре требуемые для этого адреса можно найти в приведенной на фиг. 10.37, б специальной таблице; цифры в скобках указывают номера считываемых отсчетов, пронумерованных согласно фиг. 10.37, а. Заметим, что при считывании строк для последовательных микросхем ЗУ адреса увеличиваются на единицу, а при считывании столбцов они одинаковы. Отметим также, что последовательным строкам соответствуют одни и те же наборы адресов, отличающиеся только круговой перестановкой.

Строка (или столбец) считывается параллельно и через пакетный переключатель поступает в блок обработки, состоящий из четырех АУ. Пакетный переключатель коммутируется по схеме кругового переключателя и служит для того, чтобы режим работы АУ не зависел от номера обрабатываемой строки или столбца. Например, при считывании нулевой строки [см. фиг. 10.37, б] отсчет 0 поступает в АУ0, отсчет 1 — в АУ1, отсчет 2 — в АУ2, отсчет 3 — в АУ3. При считывании первой строки отсчет 4 (считываемый из микросхемы 1) поступает в АУ0, отсчет 5 (микросхе-

	0	1	2	3	Столбцы
Строки					
0	0	1	2	3	
1	4	5	6	7	
2	8	9	10	11	
3	12	13	14	15	

а

	Микросхема 0	Микросхема I	Микросхема II	Микросхема III
	0 13 10 7	4 1 14 11	8 5 2 15	12 9 6 3
Строка 0	0 (0)	1 (1)	2 (2)	3 (3)
1	3 (7)	0 (4)	1 (5)	2 (6)
2	2 (10)	3 (11)	0 (8)	1 (9)
3	1 (13)	2 (14)	3 (15)	0 (12)
Столбец 0	0 (0)	0 (4)	0 (8)	0 (12)
1	1 (13)	1 (1)	1 (5)	1 (9)
2	2 (10)	2 (14)	2 (2)	2 (6)
3	3 (7)	3 (11)	3 (15)	3 (3)

б

Фиг. 10.37. Размещение отсчетов в микросхемах памяти для матрицы размером  $(4 \times 4)$ .

ма II) — в АУ1, отчет 6 (микросхема III) — в АУ2, отчет 7 (микросхема IV) — в АУ3. На фиг. 10.38 приведена таблица необходимых соединений. Число, стоящее в  $j$ -м столбце, указывает, какая микросхема подключается к  $j$ -му АУ. Переход от одной строки ЗУ к другой (или от одного столбца к другому) осуществляется с помощью пакетного переключателя. Отметим, что соединения в пакетном переключателе одинаковы для строки и столбца ЗУ с одинаковыми номерами.

Конкретные схемы построения процессора могут быть самыми различными, начиная со схем реализации любого из алгоритмов БПФ и кончая матричными поточными схемами.

#### 10.16. Общие принципы построения специализированных процессоров БПФ

В настоящем разделе рассматриваются вопросы, связанные с выбором исходных параметров и разработкой специализированного устройства для выполнения БПФ (процессора БПФ). Су-

Строка	0	I	II	III
1	I	II	III	0
2	II	III	0	I
3	III	0	I	II
Столбец	0	I	II	III
1	I	II	III	0
2	II	III	0	I
3	III	0	I	II

Фиг. 10.38. Взаимосвязи арифметических устройств и микросхем памяти.

ществует много различных вариантов алгоритма БПФ, поэтому важнейшая задача проектирования заключается в выборе основания алгоритма, варианта прореживания (по времени или по частоте), способа размещения промежуточных результатов (с замещением или без него) и т. д. Однако правильное решение может быть принято лишь после более глубокого анализа, связанного с выбором микросхем, степени параллелизма, структуры памяти, т. е. с такими характеристиками, которые в свою очередь зависят от быстродействия, размеров и гибкости проектируемого устройства, — от параметров, определяемых в конечном счете назначением устройства.

Невозможно перечислить все системы, в которых может быть использован процессор БПФ, поэтому ограничимся одним типичным примером. В радиолокаторах часто используются согласованные фильтры. К наиболее важным характеристикам таких фильтров относятся: 1) гибкость, выражающаяся в возможности изменения передаточной функции фильтра; 2) быстродействие, так как полоса обрабатываемого сигнала на практике может составлять несколько мегагерц. Кроме того, важную роль играет величина произведения полосы сигнала на его длительность, в конечном счете определяющая размер БПФ. В таких случаях можно применять КИХ-фильтр на основе алгоритма БПФ, предусмотрев возможность ввода-вывода в реальном времени.

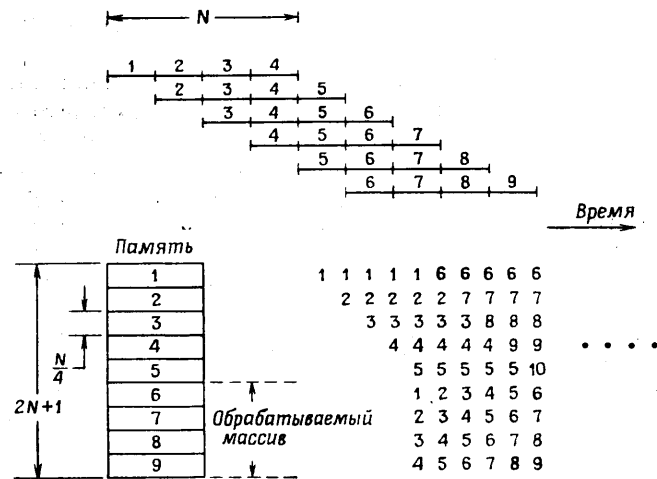
В приведенном примере, а также в аналогичных случаях быстродействие процессора БПФ определяется требованиями к быстродействию всей системы. Так, при использовании БПФ для вычисления свертки процессор БПФ должен работать в четыре раза быстрее, чем такой же процессор в системе спектрального анализа последовательных массивов. Аналогично при обработке перекрывающихся массивов быстродействие процессора БПФ должно быть пропорционально как степени перекрытия, так и полосе сигнала.

Другое часто встречающееся применение процессора БПФ — введение его в состав периферийного оборудования универсальной ЦВМ. Существуют по меньшей мере две причины, оправдывающие подключение таких приставок:

1. Если вычислительная система проводит главным образом спектральный анализ, то введение приставки позволяет сэкономить машинное время и, следовательно, денежные средства.

2. Существуют системы, в которых обработку желательно выполнять в реальном времени, причем основным препятствием этому является недостаточно быстрое вычисление БПФ.

В первом случае обеспечение определенного быстродействия не столь важно, как легкость программирования, умеренная стоимость приставки и согласующих устройств, а также гибкость процессора БПФ.



Фиг. 10.39. Структура памяти для БПФ, выполняемого в реальном времени с четырехкратным перекрытием. (Каждый столбец показывает расположение входных подмассивов в блоках ЗУ на последовательных интервалах времени.)

### 10.17. БПФ перекрывающихся массивов на основе ЗУ с произвольным доступом

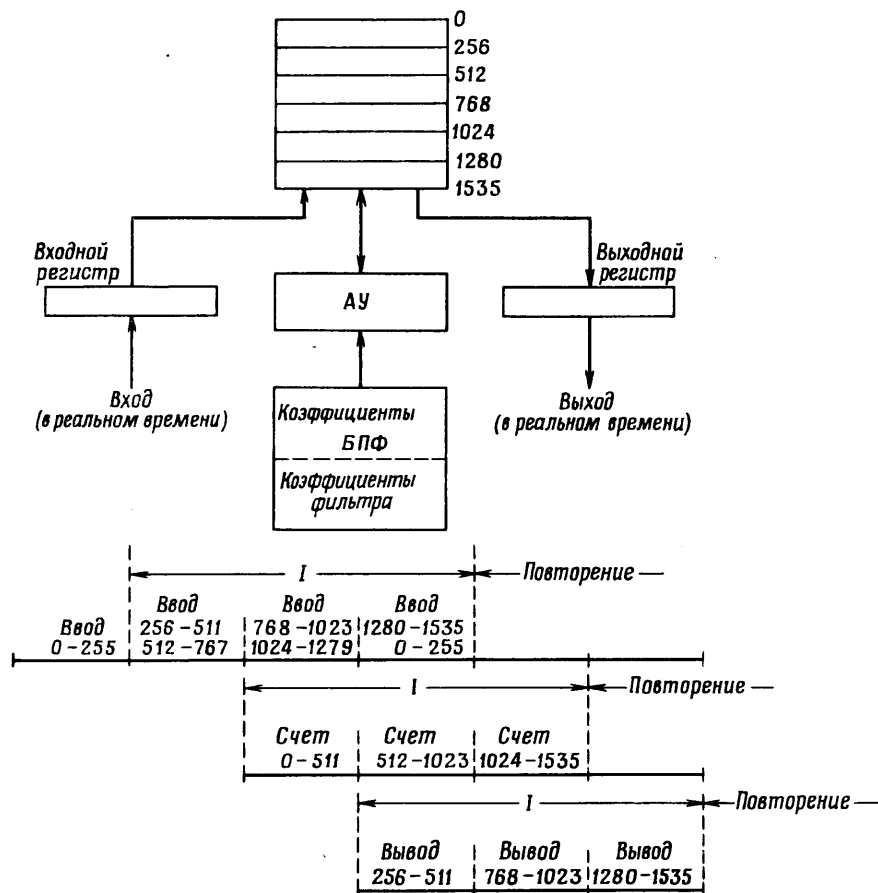
Во многих приложениях желательно выполнять БПФ перекрывающихся массивов. Выше было показано, как эта задача решается в поточной схеме БПФ. Если разделить массив, для которого выполняется БПФ, скажем, на четыре подмассива, то время выполнения БПФ должно быть равно времени поступления очередных  $N/4$  отсчетов. Алгоритм обработки перекрывающихся массивов с использованием ЗУ с произвольным доступом иллюстрируется на фиг. 10.39. Предположим, что ЗУ разделено на девять равных блоков объемом по  $N/4$  слов каждый. Пока пятый подмассив вводится в ЗУ, четыре предыдущих подмассива пересылаются в четыре нижних блока ЗУ (с 6-го по 9-й), откуда и поступают на выполнение БПФ. Для работы в реальном времени необходимо, чтобы общее время пересылки и вычисления БПФ равнялось времени ввода подмассива из  $N/4$  отсчетов. Пока шестой подмассив вводится в блок 1 ЗУ (при этом уже ненужный первый подмассив стирается), подмассивы 2 — 5 пересылаются в блоки 6—9 ЗУ для выполнения следующего БПФ. Далее обработка проводится аналогично; расположение последовательных подмассивов в ЗУ указано в соответствующих столбцах на фиг. 10.39.

### 10.18. Свертка в реальном времени методом БПФ с использованием одного АУ и ЗУ с произвольным доступом

Предположим, что для исследования влияния фильтрации на речевые сигналы нужно создать макет КИХ-фильтра, работающего в реальном времени. В этом случае речевой сигнал следует дискретизовать с частотой 20 кГц, а отсчеты представлять 9—10-разрядными числами. Используя критерии расчета фильтров, рассмотренные в гл. 3, можно оценить длину импульсной характеристики; допустим, что достаточно характеристика из 128 отсчетов. Эта величина соответствует 442 базовым операциям на каждое БПФ или  $442/128 \approx 3,45$  базовым операциям на интервал дискретизации. Как будет показано ниже, при вычислении свертки с применением БПФ потребуется в четыре раза большее быстродействие, поэтому за один 50-микросекундный интервал дискретизации нужно будет выполнять 13,8 базовых операций (т. е. на каждую базовую операцию отводится 3,62 мкс).

Из приведенных оценок видно, что в данном случае не потребуется устройство с параллельным выполнением базовых операций, так как величина 3,62 мкс представляется достаточно большой даже для логических серий с умеренным быстродействием. Тем не менее разработчик может рассмотреть много различных вариантов. Если, например, он хочет использовать микросхемы с низким быстродействием, то ему, возможно, придется строить комбинационную матрицу из четырех умножителей и шести сумматоров. При использовании более быстродействующих микросхем (например, серии ТТЛ) можно ограничиться только одним матричным умножителем (со временем умножения 0,5 мкс), а базовая операция будет выполняться путем последовательных обращений к этому умножителю. При переходе к микросхемам с еще большим быстродействием становится возможным применение умножителя, использующего сложения и сдвиги. При этом экономятся микросхемы, но зато потребляется большая мощность и усложняется работа проектировщика, возможно не имевшего ранее дела со столь быстродействующими сериями.

Допустим, что для фильтрации методом БПФ подходит система, в которой коэффициенты фильтра и отсчеты представлены числами в дополнительном коде с фиксированной запятой, имеющими соответственно 8 и 16 разрядов. Допустим также, что система обеспечивает быстродействие, оцененное выше. Далее необходимо определить структуру всего устройства: 1) выбрать тип АУ, 2) выбрать тип ЗУ, 3) разработать алгоритм обращения к ЗУ, 4) разработать методику прерываний для обеспечения ввода и вывода, 5) построить детальные временные диаграммы для всевозможных операций. Указанные этапы проектирования системы должны быть проработаны достаточно подробно, чтобы на их основе



Фиг. 10.40. Блок-схема и временная диаграмма свертывателя, использующего БПФ.

(с учетом выбранных серий микросхем) можно было создать полную принципиальную схему устройства. Опираясь на эту схему, можно перейти к разработке печатных плат и стоек, сформулировать требования к сигналам и решать множество других конструкторских задач.

Работа системы описывается временными диаграммами двух типов. Диаграммы первого типа задают распределение времени для процесса вычисления свертки в целом (т. е. между двумя БПФ и умножением на частотную характеристику фильтра), а временные диаграммы второго типа более детально представляют взаимосвязи между отдельными тактами работы АУ и ЗУ. Для составле-

ния диаграмм первого типа можно не конкретизировать структуру АУ или ЗУ и ограничиться упрощенной блок-схемой (фиг. 10.40).

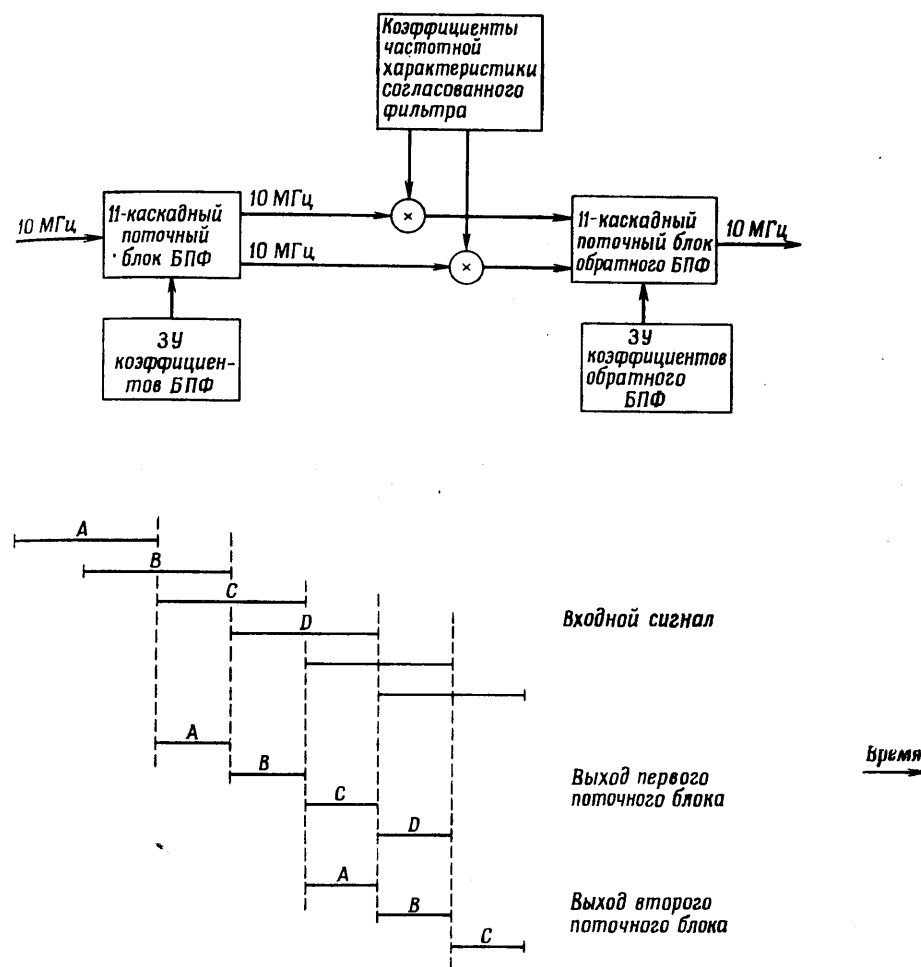
Рассмотрим для определенности проектирование КИХ-фильтра с импульсной характеристикой длиной в 256 отсчетов. Для вычисления свертки методом БПФ необходимо выполнить последовательно два 512-точечных БПФ и взять половину выходных отсчетов. Общая временная диаграмма начинается с ввода 256 комплексных отсчетов в ячейки ЗУ с адресами от 0 до 255. Длительность этой операции, представляющая собой базовый интервал работы устройства, используется в качестве единицы отсчета для всех временных диаграмм (фиг. 10.40). На следующем интервале отсчеты 256—511 вводятся не только в ячейки ЗУ с адресами 256—511, но и в ячейки 512—768. После этого можно приступить к вычислению свертки, включающему 512-точечное БПФ, умножение всех гармоник на отсчеты частотной характеристики фильтра и обратное БПФ. Для работы в реальном времени необходимо, чтобы все эти вычисления длились не более одного базового интервала. Начиная с четвертого интервала, ввод, вывод и вычисления производятся параллельно, причем каждой из этих операций соответствует временной цикл продолжительностью в три базовых интервала (на фиг. 10.40 эти циклы помечены римскими цифрами). ЗУ коэффициентов работает параллельно с АУ и ЗУ данных. Если коэффициенты БПФ и фильтра постоянны, их можно хранить в постоянном ЗУ (ПЗУ). Однако если коэффициенты желательны изменять, то нужно, чтобы ЗУ коэффициентов обеспечивало возможность записи (из ЦВМ, с перфоленки, перфокарт, с машинки и т. д.), причем частота записи в принципе может быть невысокой.

Интересно отметить, что при свертке 512-точечных реализаций (не в реальном времени) методом БПФ с замещением в обычной универсальной ЦВМ используется всего 512 (комплексных) регистров, но при переходе к работе в реальном времени это число утраивается. Кроме того, возникает вопрос, можно ли, не изменяя блок-схемы фиг. 10.40, использовать это же устройство в качестве фильтра, импульсная характеристика которого короче 256 отсчетов. Ответить на этот вопрос можно утвердительно. Действительно, выбрав соответствующие значения отсчетов частотной характеристики фильтра, можно получить импульсную характеристику любой длины, не превышающей 256. Следует, однако, учесть, что это утверждение справедливо лишь в случае, когда можно пренебречь влиянием конечной длины регистров ЗУ коэффициентов.

### 10.19. Поточный свертыватель с тактовой частотой 10 МГц

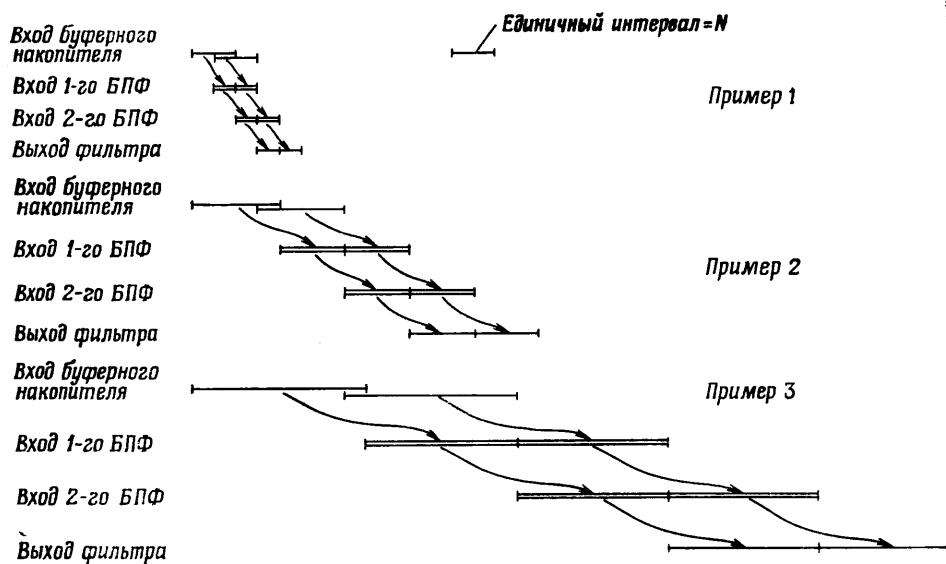
Рассмотрим радиолокатор, в котором эхо-сигналы необходимо обрабатывать в полосе 10 МГц с помощью фильтра, согласованного с разнообразными зондирующими сигналами (типа ЛЧМ-сигналов, равномерных или неравномерных пачек импульсов и т. д.). Для обеспечения требуемой гибкости фильтра нужно предусмотреть возможность простого изменения его частотной характеристики. Допустим, что максимальная длительность сигнала составляет около 100 мкс, так что произведение полосы сигнала на его длительность равно 1000. Это означает, что если фильтр построен, например, на основе многоотводной цифровой линии задержки, то число отводов должно быть равно 1000, причем каждые 100 нс нужно выполнять 1000 комплексных умножений. Решить эту задачу при использовании современных микросхем вряд ли удастся. По видимому, это тот случай, когда построение цифровой системы можно осуществить, только вычисляя свертку методом БПФ. С учетом предыдущего примера можно сделать вывод, что при вычислении свертки с использованием БПФ нужно за 102,4 мкс выполнить два 2048-точечных преобразования. Общее число базовых операций, выполняемых за это время, равно  $2048 \times 11$ , т. е. 22 528, так что каждые 100 нс приходится выполнять 22 базовые операции. Только недавно появились устройства, способные выполнить комплексное умножение за 100 нс, поэтому в данном случае потребуется 22 таких устройства, т. е. ровно столько, сколько необходимо при построении поточной схемы БПФ, описанной в разд. 10.10. Блок-схема свертывателя и его временные диаграммы приведены на фиг. 10.41. Свертыватель содержит два поточных блока БПФ и блок формирования частотной характеристики согласованного фильтра. Все три блока работают параллельно. Поскольку при поточной обработке можно одновременно выполнять два БПФ, целесообразно ввести двукратное перекрытие сигналов, что полностью согласуется с режимом работы поточной схемы и позволяет довести эффективность системы до 100%. Для каждого массива из 2048 отсчетов на двух выходах первого поточного блока БПФ появляются две последовательности сигналов. На двух выходах второго поточного блока БПФ появляются «правильная» половина отфильтрованного массива и «неправильная» (первая) половина следующего массива свертки. Таким образом, отфильтрованный сигнал формируется в реальном времени по методу перекрытия с накоплением из половины выходных отсчетов.

Для обеспечения необходимого быстродействия в поточном блоке БПФ эффективно используются  $2 \log_2 N$  параллельно работающих арифметических устройств. В предыдущем примере использовалось одно только АУ. Целесообразно попытаться найти



Фиг. 10.41. Блок-схема свертывателя, использующего БПФ.

наилучшие виды параллелизма для промежуточных случаев. Один из подходов состоит в сохранении поточного способа обработки, но при сокращении числа АУ. Уменьшим, например, полосу входного сигнала от 10 до 0,8 МГц. Заменим в обоих поточных блоках БПФ 11 АУ буферными регистрами, которые будут поочередно подключаться к одному быстродействующему АУ. Таким образом, при использовании только двух АУ (по одному на каждый поточный блок БПФ) работоспособность всего устройства сохранится, но теперь базовые операции всех этапов поточного блока будут



Фиг. 10.42. Относительные эффективности при различных размерах БПФ.

выполняться с помощью одного и того же АУ. Если же необходимо обрабатывать сигнал с полосой 3,2 МГц, для каждого из точных блоков БПФ придется использовать по 4 АУ, три из которых будут обслуживать по три этапа БПФ и одно — два этапа. Таким образом, поточная схема обработки сохраняет высокую эффективность в широком диапазоне тактовых частот.

После выбора основных характеристик системы можно перейти к более детальному проектированию. В системе фиг. 10.41, например, могут использоваться до 5 ЗУ. К ним относятся:

- 1) ЗУ, входящие в состав поточных блоков БПФ;
- 2) ЗУ для хранения коэффициентов, используемых в поточных схемах;
- 3) ЗУ частотной характеристики фильтра;
- 4) входной буферный накопитель;
- 5) выходной буферный накопитель.

Кроме того, нужно выбрать систему счисления, используемую в АУ, разрядность ячеек памяти, а также методику масштабирования и предотвращения переполнений.

Свертка методами БПФ не обязательно выполняется при двукратном перекрытии. Действительно, увеличивая размер БПФ, можно повысить эффективность свертывателя. Это обусловлено тем, что при вычислении свертки методом БПФ объем бесполезной информации зависит как от длины свертываемых последовательностей, так и от размера БПФ.

Данное положение проиллюстрировано тремя примерами (фиг. 10.42). Предположим, что во всех случаях длина импульсной характеристики равна  $N$  (что соответствует единичному отрезку на фиг. 10.42), но размер БПФ в первом примере равен  $2N$ , во втором —  $4N$ , а в третьем —  $8N$ .

Как уже было показано, в первом случае для получения  $N$  полезных отсчетов после выполнения каждой пары преобразований (размером по  $2N$  отсчетов) приходится обрасывать половину конечных результатов (см. четвертую строку первого примера). Во втором примере (при размере преобразований по  $4N$ ) остается  $3N$  полезных отсчетов. Это означает, что для выполнения  $4N$ -точечного БПФ можно отвести время, пропорциональное  $3N$ . Аналогично в третьем примере для  $8N$ -точечного преобразования можно отвести время, пропорциональное  $7N$ . Таким образом, в примерах 1, 2 и 3 относительные эффективности вычислений равны 1:2, 3:4 и 7:8. В системах с очень большим быстродействием увеличение размера преобразования с  $2N$  до  $8N$  может привести к существенному упрощению аппаратуры.

#### ЛИТЕРАТУРА

1. Bergland G. D., Fast Fourier Transform Hardware Implementations — An Overview, *IEEE Trans. on Audio and Electroacoustics*, AU-17, 104—108 (June 1969).
2. Groginsky H. L., Works G. A., A Pipeline Fast Fourier Transform, *IEEE Trans. on Computers*, C-19, 1015—1019 (Nov. 1970).
3. Klahn R., Shively R. R., Gomez E., Gilmartin M. J., The Time-Saver: FFT Hardware, *Electronics*, 92—97 (June 1968); есть русский перевод: Клан, Шайвли, Гомец, Гилмартин, Специализированный процессор для быстрого решения задач гармонического анализа, *Электроника*, т. 41, № 13 (1968).
4. Pease M. C., An Adaptation of the Fast Fourier Transform for Parallel Processing, *JACM*, 15, 252—264 (April 1968).
5. Shively R. R., A Digital Processor to Generate Spectra in Real Time, *IEEE Trans. on Computers*, C-17, 485—491 (May 1968).
6. Bergland G. D., Wilson D. E., An FFT Algorithm for a Global Highly-Parallel Processor, *IEEE Trans. on Audio and Electroacoustics*, AU-17, 125—127 (June 1969).
7. Bergland G. D., Fast Fourier Transform Hardware Implementations — A Survey, *IEEE Trans. on Audio and Electroacoustics*, AU-17, 109—119 (June 1969).
8. Theilheimer F., A Matrix Version of the Fast Fourier Transform, *IEEE Trans. on Audio and Electroacoustics*, AU-17, No. 2, 158—161 (June 1969).
9. O'Leary G. C., Nonrecursive Digital Filtering Using Cascade Fast Fourier Transformers, *IEEE Trans. on Audio and Electroacoustics*, AU-18, No. 2, 177—183 (June 1970).
10. Veenkant R. L., A Serial Minded FFT, *IEEE Trans. on Audio and Electroacoustics*, AU-20, No. 3, 180—184 (Aug 1972).
11. Gold B., Bially T., Parallelism in Fast Fourier Transform Hardware, *IEEE Trans. on Audio and Electroacoustics*, AU-21, No. 1, 5—16 (Feb. 1973).

## Эффекты, возникающие при БПФ вследствие конечной разрядности чисел

1. Oppenheim A. V., Weinstein C. J., Effects of Finite Register Length in Digital Filtering and the Fast Fourier Transform, *Proc. IEEE*, 60, No. 8, 957—976 (Aug. 1972); есть русский перевод: Оппенгейм, Вайнштейн, Влияние конечной длины регистра при цифровой фильтрации и быстром преобразовании Фурье, *ТИИЭР*, т. 60, № 8, стр. 41—65 (1969).
2. Welch P. D., A Fixed-Point Fast Fourier Transform Error Analysis, *IEEE Trans. Audio and Electroacoustics*, AU-17, No. 2, 151—157 (June 1969).
3. Weinstein C. J., Roundoff Noise in Floating Point Fast Fourier Transform Computation, *IEEE Trans. on Audio and Electroacoustics*, AU-17, No. 3, 209—215 (Sept. 1969).
4. Kaneko T., Liu B., Accumulation of Round-Off Errors in Fast Fourier Transforms, *J. Assn. Comp. Mach.*, 17, No. 4, 637—654 (Oct. 1970).

УНИВЕРСАЛЬНЫЕ УСТРОЙСТВА  
В СИСТЕМАХ ОБРАБОТКИ СИГНАЛОВ

## 11.1. Введение

При решении любой конкретной задачи специализированная аппаратура всегда эффективнее универсальных ЦВМ. Однако требуемая в очень многих случаях гибкость вычислительных средств проще всего обеспечивается с помощью универсальных ЦВМ. Именно таким специальным приложениям и посвящена настоящая глава. В ней рассматриваются вопросы проектирования универсальных устройств, предназначенных для исследования методов обработки сигналов при анализе речи, в радиолокации, гидролокации, сейсмологии и технике связи. Дело в том, что в перечисленных областях такие устройства оказываются эффективнее больших вычислительных машин, обслуживающих в режиме разделения времени все увеличивающееся число экспериментаторов. Основным назначением этих устройств является создание соответствующих алгоритмов путем моделирования разрабатываемых специализированных устройств.

В настоящее время вычислительная техника все еще продолжает стремительно развиваться. Целесообразность той или иной структуры вычислительной машины в значительной степени зависит от характеристик существующих компонентов. Так, например, когда основным видом запоминающих устройств ЦВМ были кубы памяти на магнитных сердечниках, полупроводники только начинали заменять радиолампы в арифметических устройствах, так что даже обыкновенный триггер был громоздким и дорогим. При этом внешняя (относительно куба) память была весьма дорогостоящей, и проектировщики старались уменьшить ее объем. Однако позднее, когда память на сердечниках все еще доминировала, но триггеры и арифметические устройства стали недорогими устройствами с большими функциональными возможностями, начали широко применяться общие регистры и сверхоперативные ЗУ. Появление микросхем со средним уровнем интеграции привело к быстрому внедрению принципов параллельной обработки, многие из которых и в настоящее время не являются бесспорными. Быстрый прогресс в области создания новых компонентов стимулировал