8.9. Заключение

Настоящая глава по своему содержанию несколько отличается от всех предшествующих глав книги. До сих пор основное внимание уделялось изложению основ теории и методов проектирования цифровых фильтров и анализаторов спектра. Кроме того, предыдущие главы имеют достаточно высокий научный уровень, тогда как данная глава носит вводный характер. Возможно, что инженерам, хорошо знакомым с цифровой техникой, материал этой главы покажется достаточно элементарным и далеко не полным введением в эту область. Тем не менее мы убеждены, что такая глава необходима в качестве связующего звена между предыдущими главами теоретического характера и последующими главами, имеющими практическую направленность. Наш опыт свидетельствует о том, что разработчик должен быть достаточно сведущ в вопросах, связанных с достижением высокого быстродействия, однако найти соответствующую публикацию, в которой были бы изложены все эти вопросы, мы не смогли. Это объясняет то внимание, которое было уделено здесь интегральным схемам ЭСЛ и матричным умножителям. Другие методы умножения более подробно изложены во многих книгах.

ЛИТЕРАТУРА

1. Garret L. S., Integrated Circuit Digital Logic Families, Parts I—III, *IEEE Spectrum* (Oct.— Dec. 1970).
2. Hnatek E. R., A User's Handbook of Integrated Circuits, Wiley, N.Y., 1973.
3. MECL Integrated Circuits Data Book, Motorola, Nov. 1972.
4. MECL System Design Handbook, Motorola, Oct. 1971.
5. Pezaris S., A 40 Nanosecond 17 × 17 Array Multiplier, *IEEE Trans. on Computers*, C-20, 442—447, No. 4 (April 1971).
6. Blankenship P., Gold B., McHugh P., Weinstein C., Design Study of the Advanced Signal Processor, Lincoln Lab. Technical Note, April 1972.

Глава 9

СПЕЦИАЛИЗИРОВАННЫЕ УСТРОЙСТВА ДЛЯ ЦИФРОВОЙ ФИЛЬТРАЦИИ И ГЕНЕРАЦИИ СИГНАЛОВ

9.1. Введение

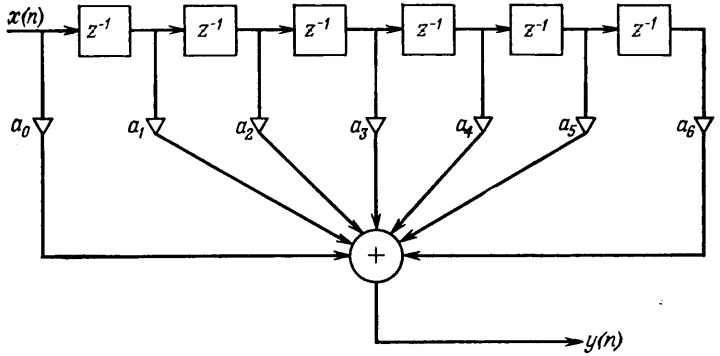
В предыдущих главах основное внимание уделялось программной реализации алгоритмов цифровой обработки сигналов. В данной, а также в двух последующих главах будут рассмотрены специализированные цифровые устройства, предназначенные для ускоренного выполнения алгоритмов обработки и для построения систем обработки в реальном масштабе времени.

Выше уже были рассмотрены основные составные части цифровых устройств, включая цепи задержки, сумматоры, умножители, универсальные элементы памяти, а также различные типы серийно выпускаемых логических интегральных схем; проанализировано взаимоотношение между их быстродействием, стоимостью и потребляемой мощностью. Важную роль в достижении максимума отношения качества системы к ее стоимости играют методы распараллелирования, временно́го разделения и поточной обработки. Прежде всего в данной главе будут рассмотрены различные способы построения цифровых фильтров, в том числе прямая и каскадная формы КИХ-фильтров, прямая, каскадная и параллельная формы БИХ-фильтров. После этого будет приведено несколько практических примеров построения цифровых специализированных систем, включая реализацию цифровых фильтров. В заключение будут описаны цифровой синтезатор частот и цифровой генератор шума.

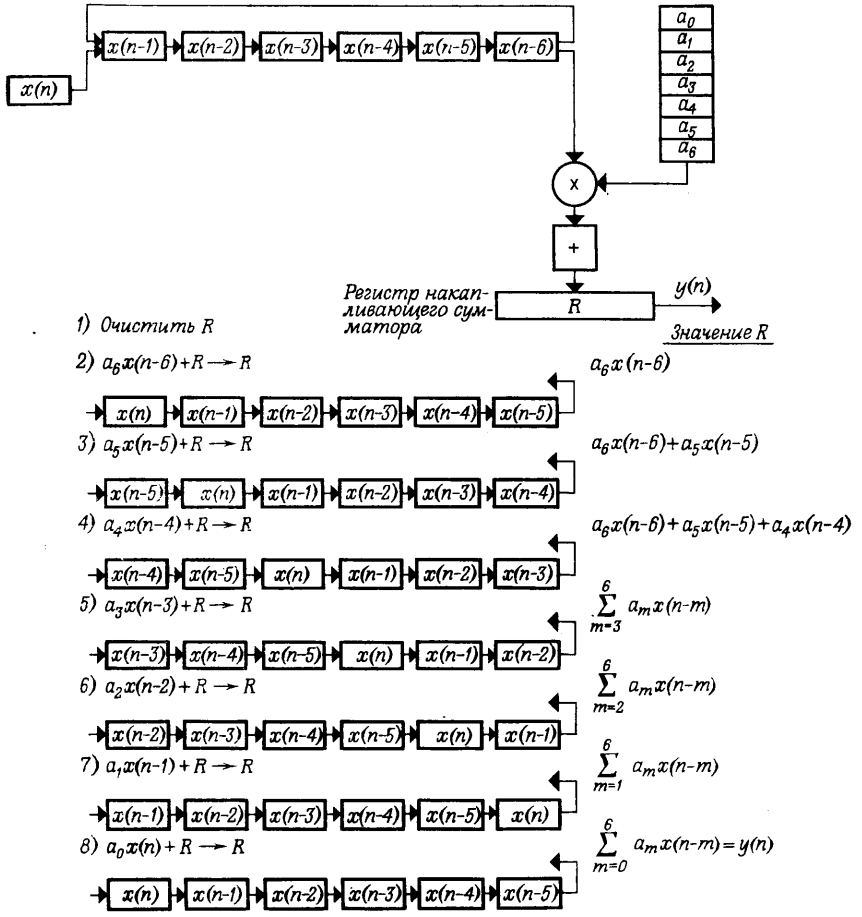
9 2. Аппаратурное построение КИХ-фильтра прямой формы

Рассмотрим способы построения КИХ-фильтра прямой формы, схематично изображенного на фиг. 9.1. Найдем прежде всего основную схему управления, пригодную для построения фильтров с различным уровнем параллелизма, причем параллелизм будет заключаться не только в увеличении количества арифметических устройств, но и в параллельной работе блоков памяти.

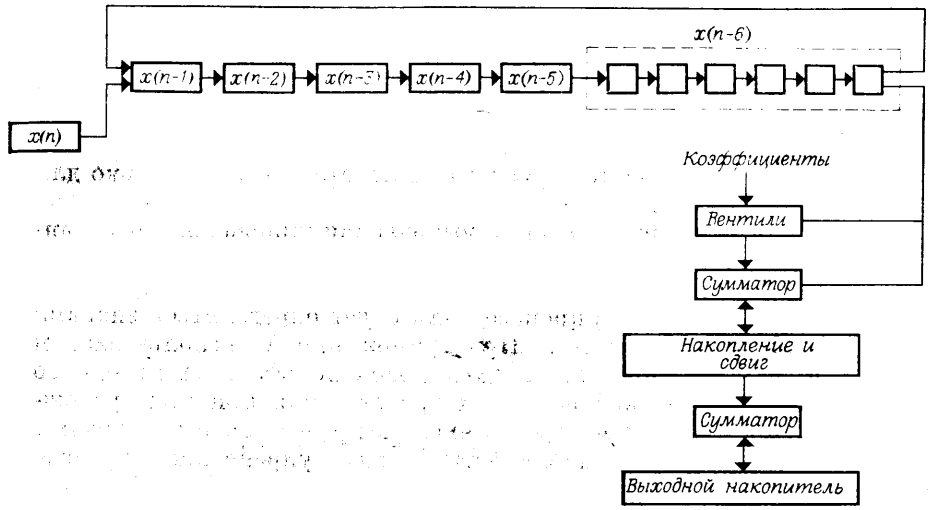
На фиг. 9.2 изображена простая блок-схема построения КИХ-фильтра прямой формы (см. фиг. 9.1) из единственного



Фиг. 9.1. Прямая форма КИХ-фильтра.



Фиг. 9.2. Блок-схема, последовательность обработки и промежуточные результаты для прямой формы КИХ-фильтра.



Фиг. 9.3. Построение КИХ-фильтра с использованием алгоритма Бута.

арифметического устройства (содержащего умножитель и сумматор), регистра сдвига для хранения промежуточных результатов и ПЗУ коэффициентов. Важным моментом является характер управления регистром сдвига. Каждый выходной отсчет вычисляется путем накопления последовательных произведений, образующихся на выходе умножителя в процессе кругового сдвига регистра. Новый входной отсчет $x\,(n)$ вводится в регистр одновременно с выталкиванием с противоположного конца регистра отсчета $x\,(n-6)$, сопровождающимся вычислением произведения $a_6 \cdot x\,(n-6)$. При каждой итерации, как это видно из схемы вычислений на фиг. 9.2, производится круговой сдвиг регистра на один отсчет. После вычисления значения $y\,(n)$ оно выводится; одновременно с этим очищается накопитель, после чего начинается следующий главный цикл.

Заметим, что структура памяти на сдвиговом регистре, используемой в схеме на фиг. 9.2, еще не была определена. Прежде всего необходимо выяснить, на каких микросхемах ее можно собирать. Оказывается, что для этой цели хорошо подходят большие интегральные схемы с МДП-регистрами сдвига. В настоящее время на одном кристалле могут размещаться несколько тысяч последовательно соединенных разрядов памяти. Поэтому для многих КИХ-фильтров вся память нужного объема размещается в одном корпусе, но при условии, что данные вводятся и подаются на арифметическое устройство последовательно. Конечно, было бы интересно построить КИХ-фильтр прямой формы

с использованием именно такой памяти, состоящей из малого числа элементов. Простейшая схема подобного типа приведена на фиг. 9.3. В этой схеме каждое произведение получается с помощью последовательности логических умножений коэффициента на разряды перемножаемого отсчета [в данном случае $x(n-6)$] и суммирования этих произведений. Все отсчеты в схеме на фиг. 9.3 хранятся в последовательном виде, хотя это показано только для отсчета $x(n-6)$.

Следует напомнить, что при выполнении умножения, основанного на сдвигах и сложениях, вид алгоритма зависит от знаков сомножителей. Так, при использовании прямого кода сложение будет обычным, а знак произведения будет определяться знаками отсчета и коэффициента. При умножении с использованием сдвигов и сложений этот подход довольно эффективен; однако заключительное накопление всех произведений при этом усложняется, так как для сложения чисел в прямом коде необходимы сумматор-вычитатель и соответствующее управление, которое обеспечило бы вычитание меньшего числа из большего. Накопление упрощается при использовании вместо прямого кода дополнительного.

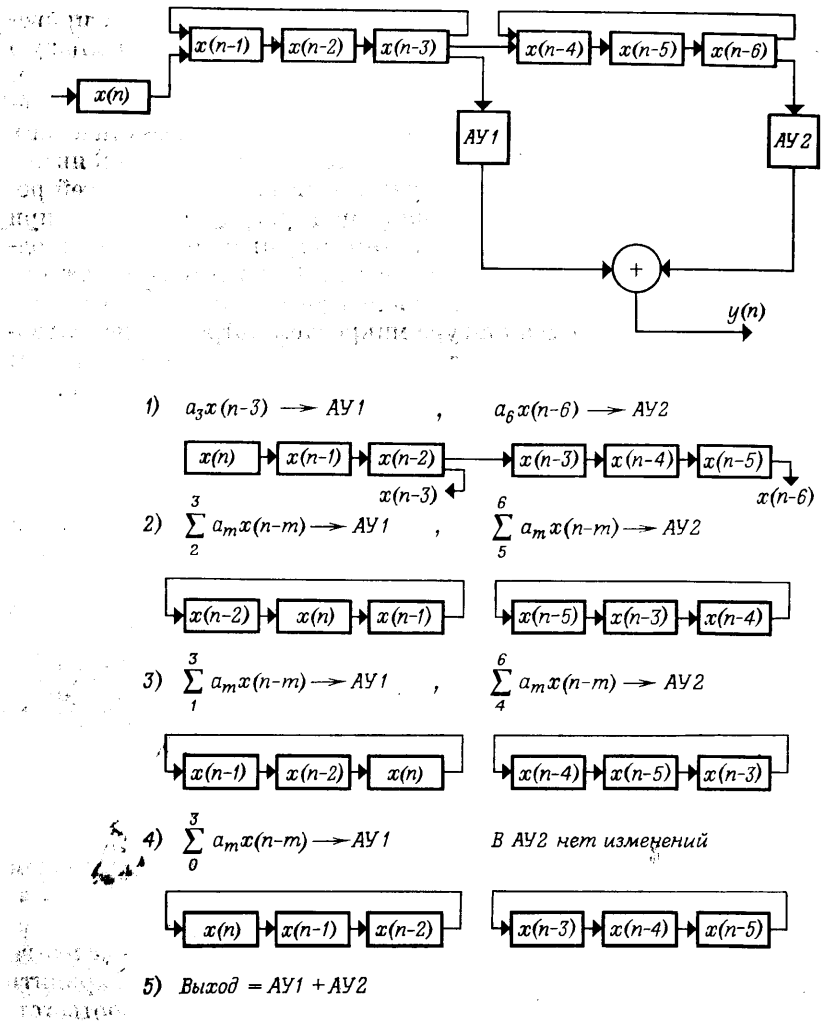
Отсчеты и коэффициенты также могут храниться в дополнительном, а не в прямом коде. Это означает, что необходимо специальное управление первым сумматором (фиг. 9.3). Удобнее всего для этого использовать алгоритм Бута (он был рассмотрен в гл. 8), в котором анализ двух соседних разрядов позволяет решить, что следует выполнить — сложение или вычитание.

9.3. Параллелизм при построении КИХ-фильтров прямой формы

Если входные отсчеты представлены b-разрядными числами, для вычисления выходного отсчета приходится выполнить N умножений и предполагается, что время сдвига регистра на один разряд τ равно времени сложения, то максимальная скорость обработки данных R, которая может быть достигнута с помощью схемы, рассмотренной в разд. 9.2 (см., например, фиг. 9.3), равна $1/(Nb\tau)$. Так, если $\tau = 100$ нс, $N = 16$ и $b = 16$, то

$$R = \frac{10^9}{256 \times 100} = 39\,062{,}5 \text{ Гц}.$$

Для увеличения R можно использовать различные способы параллелизма, касающиеся как работы памяти, так и выполнения арифметических операций. Используя два или более регистра сдвига, можно пропорционально увеличить скорость обработки; аналогичного результата можно достичь, используя два или более арифметических устройства. Рассмотрим все эти варианты, чтобы выяснить, какое дополнительное управление необходимо и какими должны быть арифметические устройства.



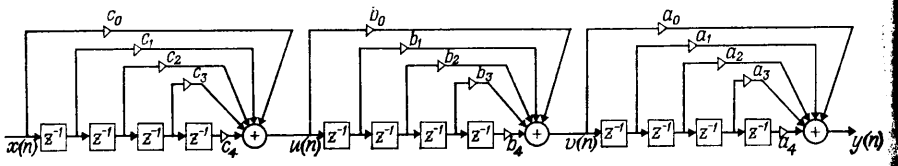
Фиг. 9.4. Блок-схема, последовательность обработки и промежуточные результаты для КИХ-фильтра с двумя параллельно работающими АУ.

На фиг. 9.4 представлены все необходимые элементы и связи между ними для системы, содержащей два параллельно работающих арифметических устройства. Там же показана последовательность вычислений и размещение данных в регистре сдвига после выполнения очередной операции умножение — накопление.

По сравнению со схемой, содержащей одно АУ (она была приведена на фиг. 9.3), данная схема обеспечивает вдвое большую скорость обработки при условии, что схемы АУ1 и АУ2 такие же, как у арифметического устройства на фиг. 9.3. Рассмотренный метод параллелизма легко обобщить на любое число арифметических устройств, однако не следует забывать, что высокий уровень интеграции сохранится только в том случае, если каждая из частей регистра сдвига является достаточно многоразрядной. Так, при построении фильтра 20-го порядка с 16-разрядными отсчетами и одним АУ требуется 320-разрядный кристалл. При построении такого же фильтра с двумя АУ необходимо иметь регистр сдвига с отводом, что при современной номенклатуре микросхем потребует использования двух корпусов, хотя не исключено, что такие регистры сдвига с отводами вполне могут быть изготовлены специально.

9.4. Каскадная форма КИХ-фильтра

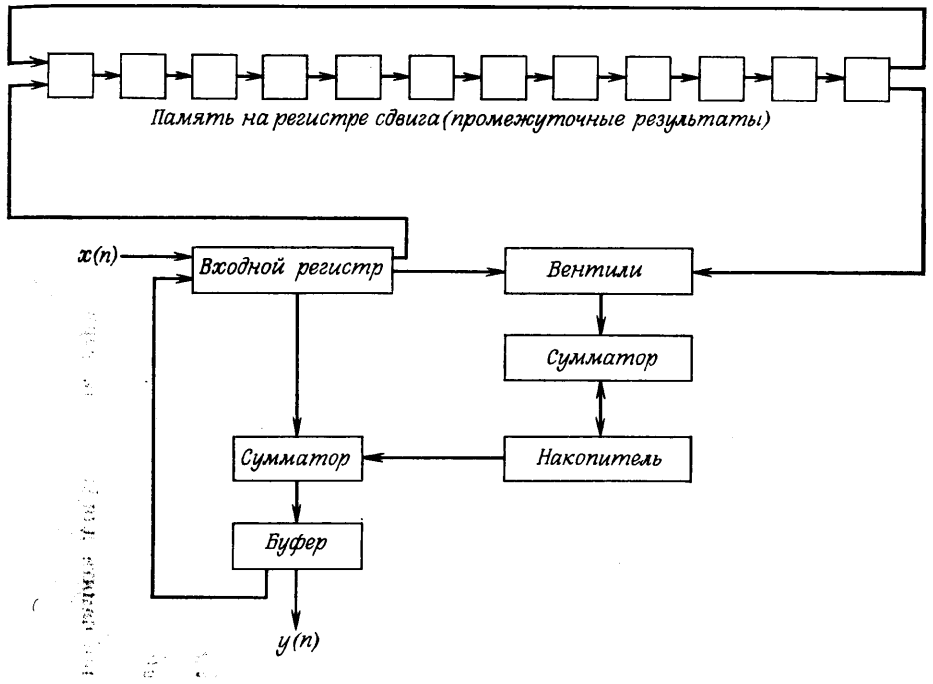
Схема, приведенная на фиг. 9.4, может быть использована и для построения КИХ-фильтров каскадной формы. На фиг. 9.5



Фиг. 9.5. Каскадная форма КИХ-фильтра.

в качестве примера изображен трехкаскадный фильтр, причем каждый из блоков представляет собой фильтр четвертого порядка. На фиг. 9.6 показано, как строится такой фильтр с использованием одного АУ и одного последовательного регистра сдвига в качестве памяти промежуточных результатов. Здесь требуется сохранять выходные отсчеты каждого из блоков и вводить их в соответствующие моменты времени в регистр сдвига. Характер размещения промежуточных результатов в регистре сдвига для 12 последовательных тактов обработки показан на фиг. 9.7а, а программа, по которой работают все устройства фильтра, — на фиг. 9.7б. Видно, что отсчеты $x(n-4)$, $u(n-4)$ и $v(n-4)$ выводятся из памяти, а вместо них вводятся три новых отсчета.

Фильтр, изображенный на фиг. 9.5, легко может быть построен с использованием параллельно работающих арифметических устройств, причем здесь возможны два простых подхода. Первый из них состоит в том, что для каждого из блоков четвертого порядка



Фиг. 9.6. Построение каскадной формы КИХ-фильтра с использованием единственного АУ.

используется свое АУ, так что весь фильтр будет содержать три АУ, как показано на фиг. 9.8. Из приведенной там же программы работы фильтра видно, что во всех трех АУ расчет последовательных частичных сумм произведений производится до тех пор, пока в каждом из них не будет сформирована сумма всех четырех произведений. На пятом такте в первом АУ заканчивается вычисление $u(n)$, на вход поступает новый отсчет $x(n)$ и все регистры сдвигаются. Получаемое значение $u(n)$ используется в качестве входного для следующего блока и может быть добавлено к результату второго АУ и т. д. Таким образом, при использовании трех последовательно включенных арифметических устройств для завершения вычислений необходимы три дополнительных такта.

В качестве упражнения читателю предлагается разработать схему каскадного КИХ-фильтра с четырьмя АУ и уровнем параллелизма, достаточным для завершения вычислений в каждом из блоков за один цикл.

	v(n-1)	v(n-2)	v(n-3)	v(n-4)	u(n-1)	u(n-2)	u(n-3)	u(n-4)	x(n-1)	x(n-2)	x(n-3)	x(n-4)
1)	——	v(n-1)	v(n-2)	v(n-3)	v(n-4)	u(n-1)	u(n-2)	u(n-3)	u(n-4)	x(n-1)	x(n-2)	x(n-3)
2)	x(n-3)	——	v(n-1)	v(n-2)	v(n-3)	v(n-4)	u(n-1)	u(n-2)	u(n-3)	u(n-4)	x(n-1)	x(n-2)
3)	x(n-2)	x(n-3)	——	v(n-1)	v(n-2)	v(n-3)	v(n-4)	u(n-1)	u(n-2)	u(n-3)	u(n-4)	x(n-1)
4)	x(n-1)	x(n-2)	x(n-3)	——	v(n-1)	v(n-2)	v(n-3)	v(n-4)	u(n-1)	u(n-2)	u(n-3)	u(n-4)
5)	x(n)	x(n-1)	x(n-2)	x(n-3)	——	v(n-1)	v(n-2)	v(n-3)	v(n-4)	u(n-1)	u(n-2)	u(n-3)
6)	u(n-3)	x(n)	x(n-1)	x(n-2)	x(n-3)	——	v(n-1)	v(n-2)	v(n-3)	v(n-4)	u(n-1)	u(n-2)
7)	u(n-2)	u(n-3)	x(n)	x(n-1)	x(n-2)	x(n-3)	——	v(n-1)	v(n-2)	v(n-3)	v(n-4)	u(n-1)
8)	u(n-1)	u(n-2)	u(n-3)	x(n)	x(n-1)	x(n-2)	x(n-3)	——	v(n-1)	v(n-2)	v(n-3)	v(n-4)
9)	u(n)	u(n-1)	u(n-2)	u(n-3)	x(n)	x(n-1)	x(n-2)	x(n-3)	——	v(n-1)	v(n-2)	v(n-3)
10)	v(n-3)	u(n)	u(n-1)	u(n-2)	u(n-3)	x(n)	x(n-1)	x(n-2)	x(n-3)	——	v(n-1)	v(n-2)
11)	v(n-2)	v(n-3)	u(n)	u(n-1)	u(n-2)	u(n-3)	x(n)	x(n-1)	x(n-2)	x(n-3)	——	v(n-1)
12)	v(n-1)	v(n-2)	v(n-3)	u(n)	u(n-1)	u(n-2)	u(n-3)	x(n)	x(n-1)	x(n-2)	x(n-3)	——
13)	v(n)	v(n-1)	v(n-2)	v(n-3)	u(n)	u(n-1)	u(n-2)	u(n-3)	x(n)	x(n-1)	x(n-2)	x(n-3)

Фиг. 9.7а. Последовательные состояния памяти для каскадной формы КИХ-фильтра.

1) $c_4x(n-4)$ → *Накопитель*

2) $c_4x(n-4) + c_3x(n-3)$ → *Накопитель*

3) $\sum_{m=2}^{4} c_m x(n-m)$ → *Накопитель*

4) $\sum_{m=1}^{4} c_m x(n-m)$ → *Накопитель*

5) $u(n) = \sum_{m=0}^{4} c_m x(n-m)$: *Ввести* $x(n)$ *в ЗУ:*

Ввести $u(n)$ *во входной регистр:* $b_4u(n-4)$ → *Накопитель*

6) $b_4u(n-4) + b_3u(n-3)$ → *Накопитель*

7) $\sum_{m=2}^{4} b_m u(n-m)$ → *Накопитель*

8) $\sum_{m=1}^{4} b_m u(n-m)$ → *Накопитель*

9) $v(n) = \sum_{m=0}^{4} b_m u(n-m)$: *Ввести* $u(n)$ *в ЗУ:*

Ввести $v(n)$ *во входной регистр* : $a_4v(n-4)$ → *Накопитель*

10) $a_4v(n-4) + a_3v(n-3)$ → *Накопитель*

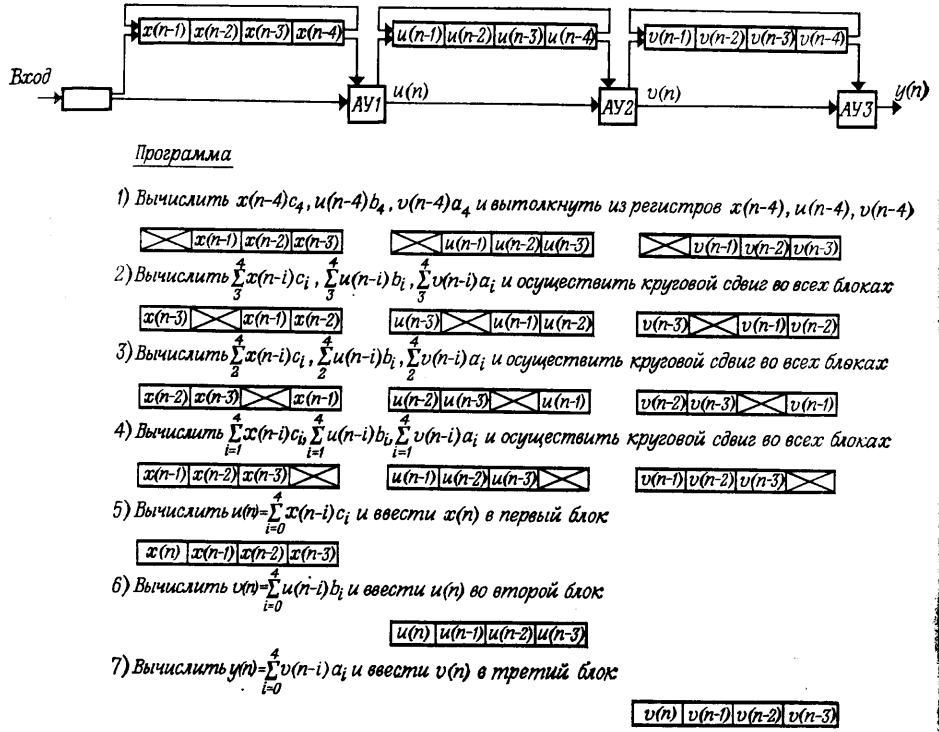
11) $\sum_{m=2}^{4} a_m v(n-m)$ → *Накопитель*

12) $\sum_{m=1}^{4} a_m v(n-m)$ → *Накопитель*

13) $y(n) = \sum_{m=0}^{4} a_m v(n-m)$: *Выход* $y(n)$

Ввести $v(n)$ *в ЗУ: Ввести* $x(n)$ *во входной регистр*

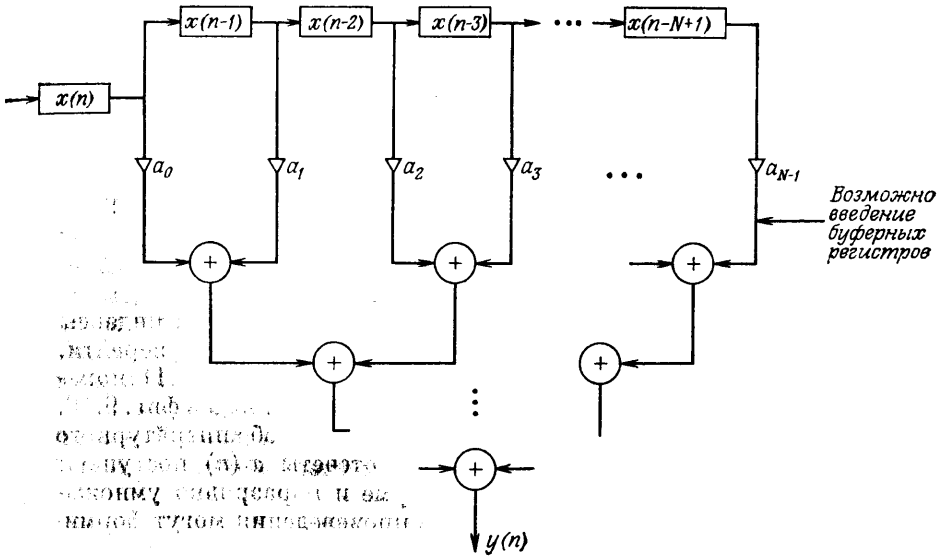
Фиг. 9.7б. Программа работы каскадной формы КИХ-фильтра.



Фиг. 9.8. Построение каскадного КИХ-фильтра с использованием трех АУ.

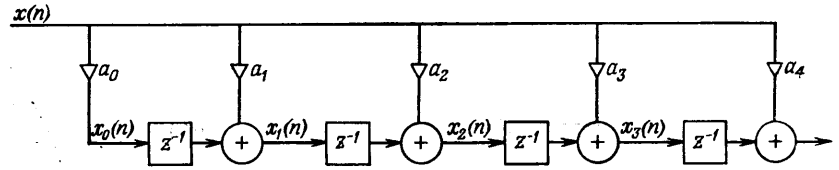
9.5. Прямая форма КИХ-фильтра с высоким уровнем параллелизма

Как уже было отмечено, основная структура фильтра с регистром сдвига в качестве ЗУ позволяет использовать параллелизм и при выполнении арифметических операций, и при работе памяти. Возникает вопрос, какого предельного уровня параллелизма можно достичь в схемах подобного типа. Прежде чем ответить на этот вопрос, рассмотрим систему с достаточно очевидным параллелизмом, а именно систему N-го порядка с N арифметическими устройствами. В такой системе, изображенной на фиг. 9.9, для каждого из входных отсчетов используется отдельный умножитель. Заметим, что в этом случае не требуется кругового сдвига памяти, так как регистр используется лишь как обычная линия задержки с отводами. Отметим также, что необходимость иметь отводы сводит на нет преимущества современной БИС-технологии. Как и во всех предыдущих примерах, отсчеты в регистре сдвига могут храниться как в последовательном, так и в параллельном виде.



Фиг. 9.9. Блок-схема КИХ-фильтра N-го порядка с N параллельно работающими АУ.

Далее, кроме N умножителей, необходимо иметь столько же сумматоров. Считая, что время умножения согласовано со временем сдвига регистра из одного состояния в другое, приходим к выводу, что необходимо иметь дополнительное время для накопления произведений. Это дополнительное время может не потребоваться, если в точках схемы, отмеченных на фиг. 9.9, ввести буферные регистры, что приводит к организации отработки в поточной форме. Если предположить, что общее время накопления в точности равно времени умножения, то выходная последовательность будет дополнительно задержана на один отсчет. Приведенная на фиг. 9.9 схема — не единственная, позволяющая для построения параллельного КИХ-фильтра N-го порядка использовать N арифметических устройств. Другая возможная схема показана на фиг. 9.10.



Фиг. 9.10. Еще одна схема построения КИХ-фильтра.

Эта схема описывается следующими уравнениями:

$$x_1(n) = a_0 x(n-1) + a_1 x(n),$$

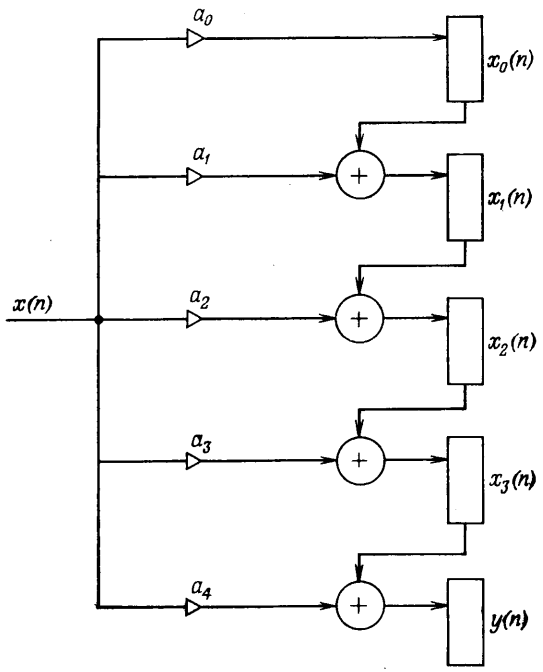
$$x_2(n) = x_1(n-1) + a_2 x(n) = a_0 x(n-2) + a_1 x(n-1) + a_2 x(n),$$

$$x_3(n) = x_2(n-1) + a_3 x(n) = \sum_{i=0}^{3} a_i x(n-3+i),$$

$$y(n) = x_3(n-1) + a_4 x(n) = \sum_{i=0}^{4} a_i x(n-4+i).$$

Видно, что, как и раньше, результат равен свертке, хотя индексы и оказались переставленными. К обычной форме можно перейти, положив $b_0 = a_3$, $b_1 = a_2$, $b_2 = a_1$, $b_3 = a_0$. На фиг. 9.11 показана простая схема построения фильтра, приведенного на фиг. 9.10.

Из схемы на фиг. 9.11 вытекает простой способ аппаратурного построения фильтра при условии, что отсчеты $x(n)$ поступают на умножители в последовательной форме и поразрядно умножаются на коэффициенты. В этом случае произведения могут формироваться последовательно, что позволяет применить последовательные сумматоры. Таким образом, по мере того как отсчеты $x_i(n)$ поразрядно выталкиваются из регистра в сумматор, в этот же регистр в последовательной форме вводится результат суммирования значения $x_{i-1}(n)$ с произведением $a_i x(n)$.



Фиг. 9.11. Блок-схема КИХ-фильтра.

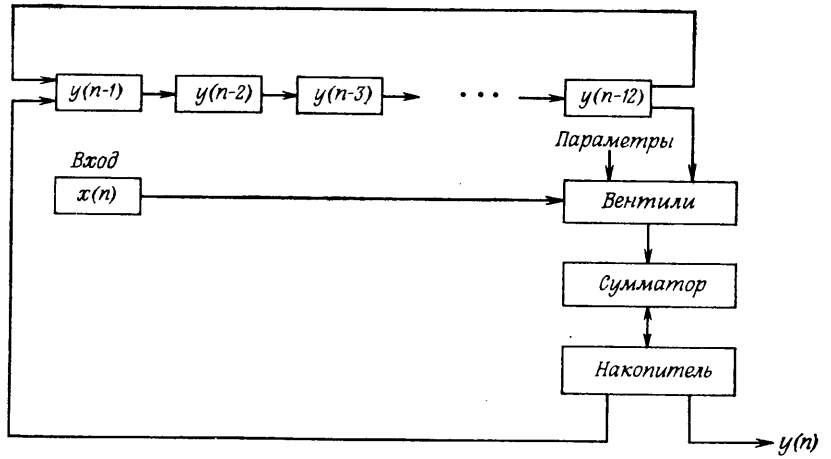
9.6. Прямая форма построения БИХ-фильтров

До сих пор рассматривались структуры, соответствующие КИХ-фильтрам. Для перехода к БИХ-фильтрам в эти структуры необходимо ввести лишь незначительные изменения. В качестве примера рассмотрим построение фильтра, содержащего только полюсы, используя форму БИХ-фильтра, приведенную на фиг. 9.12. Считается, что коэффициенты используются при вычислениях в параллельной форме, тогда как промежуточные результаты хранятся в памяти, как и в случае КИХ-фильтров, в последовательной форме. Разностное уравнение изображенного на фиг. 9.12 БИХ-фильтра, содержащего только полюсы, имеет вид

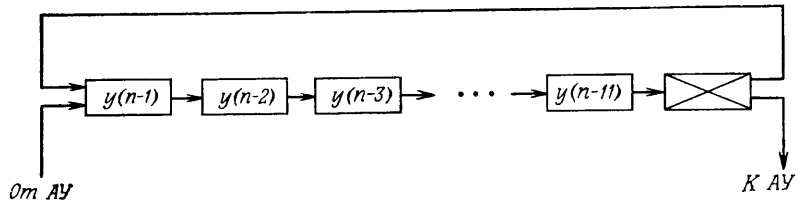
$$y(n) = \sum_{j=1}^{12} y(n-j)\, a_j + bx(n),$$

причем коэффициенты a_j и b считаются известными.

Промежуточные результаты $y(n-j)$ и входные отсчеты $x(n)$ поступают на вентили в последовательной форме. При умножении $y(n-12)$ на a_{12} значение $y(n-12)$ поразрядно выдается с выхода регистра сдвига. При выполнении оставшихся 11 умножений, начиная с $y(n-11)$ a_{11} и кончая $y(n-1)$ a_1, содержимое регистра сдвигается по кругу. Таким образом, перед поступлением на



Фиг. 9.12. Прямая форма построения БИХ-фильтра.



Фиг. 9.13. Состояние памяти БИХ-фильтра непосредственно перед вводом входного отсчета $x(n)$.

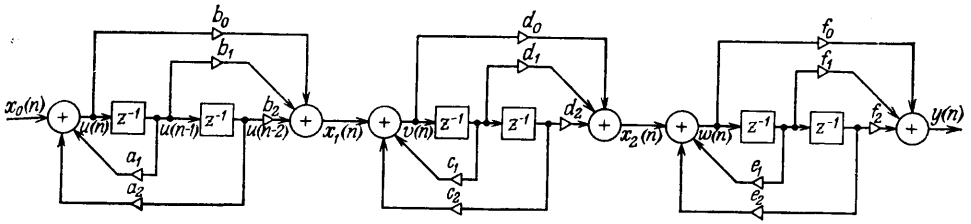
вход регистра $x(n)$ его состояние имеет вид, представленный на фиг. 9.13.

Пока вычисляется произведение $b \cdot x(n)$, отсчет $y(n)$ на вход регистра не подается, поэтому круговой сдвиг регистра не производится. Заключительный цикл включает ввод накопленного значения $y(n)$ в регистр сдвига, ввод нового значения $x(n)$, а также очистку накопителя непосредственно перед началом следующего цикла.

Типичное значение частоты дискретизации равно 8 кГц, что соответствует интервалу между отсчетами, равному 125 мкс. Если считать, что промежуточные результаты представляются 16-разрядными числами, то общее количество сдвигов регистра на один разряд оказывается равным $16 \times 14 = 224$, т. е. период следования тактовых импульсов должен быть равен 125/224 мкс (0,55 мкс). Итак, информация в регистре сдвига должна циркулировать с частотой 2 Мбит/с; с такой же частотой должно производиться сложение в сумматоре. Такая частота вполне достижима при использовании микросхем с умеренным быстродействием, так что подобные фильтры вполне можно разработать в виде большой интегральной схемы.

9.7. Каскадная форма БИХ-фильтров

На фиг. 9.14 изображен каскадный БИХ-фильтр, построенный из трех последовательно соединенных блоков второго порядка, каждый из которых имеет по два полюса и по два нуля. Все блоки второго порядка построены на основе прямой канонической формы, т. е. с использованием минимума элементов задержки. Простая блок-схема построения такого фильтра с использованием двух арифметических устройств, а также последовательность выполнения вычислений (т. е. управление) представлены на фиг. 9.15. Эта блок-схема аналогична рассмотренным выше в том смысле, что, как и ранее, память строится на последовательных регистрах сдвига, а умножитель использует сложения и сдвиги, обрабатывая последовательно поступающие разряды промежуточных результатов.



$$u(n)=a_1u(n-1)+a_2u(n-2)+x_0(n)$$
$$x_1(n)=b_0u(n)+b_1u(n-1)+b_2u(n-2)$$

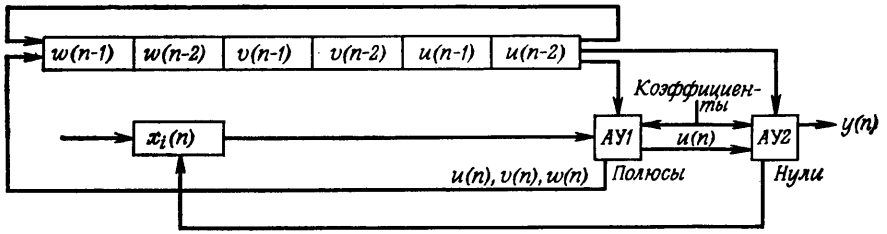
Фиг. 9.14. Каскадная форма БИХ-фильтра, состоящая из трех последовательно соединенных блоков второго порядка.

Удобство системы из двух АУ заключается в независимом вычислении для каждого из блоков второго порядка частичных сумм в цепях прямой и обратной связи. Поскольку вычисления в цепи обратной связи должны быть завершены до того, как величины $u(n)$, $v(n)$ и $w(n)$ потребуются для вычислений в цепи прямой связи, то вслед за двумя циклами сдвига промежуточных результатов в обоих АУ последовательно выполняются две вспомогательные операции: одна для формирования недостающего отсчета для цепи прямой связи, а другая для формирования входного отсчета следующего каскада.

В приведенной на фиг. 9.15 последовательности вычислений отмечено, когда регистры сдвигаются и когда сдвигов не должно быть. Видно, что для фильтра шестого порядка в общей сложности требуется семь сдвигов, а также шесть интервалов времени, на которых сдвиги не производятся. Масштабирование (т. е. простой сдвиг отсчетов) $x_0(n)$, $x_1(n)$ и $x_2(n)$ также может потребоваться, когда эти отсчеты вводятся в процессе вычислений в АУ1.

9.8. Мультиплексирование

Как правило, реальные системы оказываются сложнее рассмотренных в предыдущих разделах. Сравнительно часто используется набор КИХ-фильтров или БИХ-фильтров, причем каждый из фильтров может быть построен в параллельной или каскадной форме. Чем большее количество фильтров приходится мультиплексировать, тем, по-видимому, выгоднее использовать цифровые методы фильтрации, так как благодаря достижениям в области создания последовательных ЗУ большой интеграции эти ЗУ легко можно приспособить для запоминания промежуточных результатов. Следует, однако, с осторожностью относиться к созданию многофункциональной системы путем использования общих запоминающих и арифметических устройств, поскольку при этом будет заметно усложняться управление.



1) $a_2u(n-2), b_2u(n-2)$

	w(n-1)	w(n-2)	v(n-1)	v(n-2)	u(n-1)

2) $a_1u(n-1)+a_2u(n-2), b_1u(n-1)+b_2u(n-2)$

u(n-1)		w(n-1)	w(n-2)	v(n-1)	v(n-2)

3) $u(n)=a_1u(n-1)+a_2u(n-2)+x_0(n)$ — *Не вводить сдвигом из АУ1*

4) $x_1(n)=\sum_{i=0}^{2} b_i u(n-i)$ — *Не вводить сдвигом из АУ2*

5) $c_2v(n-2), d_2v(n-2)$. *Ввести* $x_1(n)$ *в буфер* x, *ввести сдвигом* $u(n)$ *из АУ1*

u(n)	u(n-1)		w(n-1)	w(n-2)	v(n-1)

6) $\sum_{i=1}^{2} c_i v(n-i), \sum_{i=1}^{2} d_i v(n-i)$

v(n-1)	u(n)	u(n-1)		w(n-1)	w(n-2)

7) $v(n)=\sum_{i=0}^{2} c_i u(n-1)+x_1(n)$ — *Не сдвигать* — *АУ1*

8) $x_2(n)=\sum_{i=0}^{2} d_i v(n-i)$ — *АУ2*

9) $e_2w(n-2)$, $f_2w(n-2)$. *Ввести* x_2 *в буфер, ввести сдвигом* $v(n)$ *из АУ1*

v(n)	v(n-1)	u(n)	u(n-1)		w(n-1)

10) $e_1w(n-1)+e_2w(n-2), f_1w(n-1)+f_2w(n-2)$

w(n-1)	v(n)	v(n-1)	u(n)	u(n-1)	

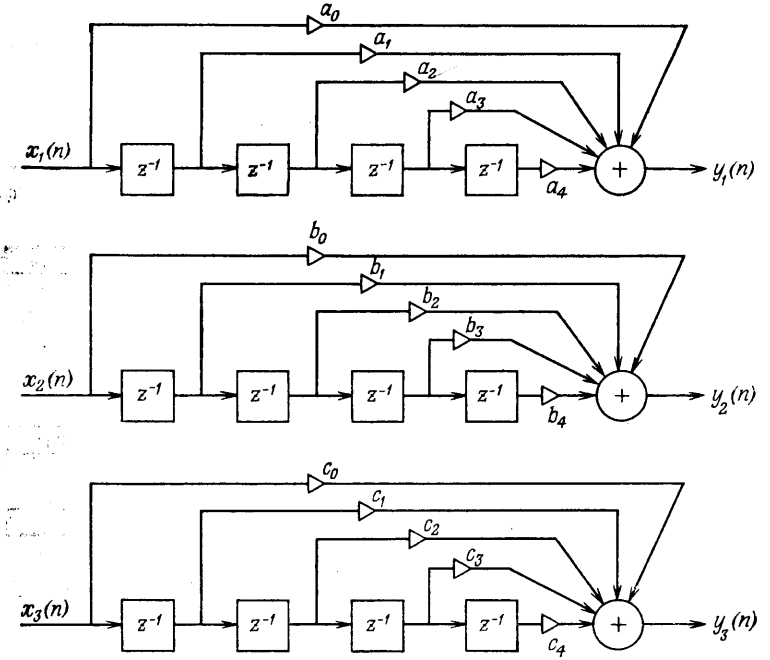
11) *Вычислить* $w(n)=e_1w(n-1)+e_2w(n-2)+x_2(n)$

12) *Вычислить* $y(n)=\sum_{i=0}^{2} f_i w(n-i)$

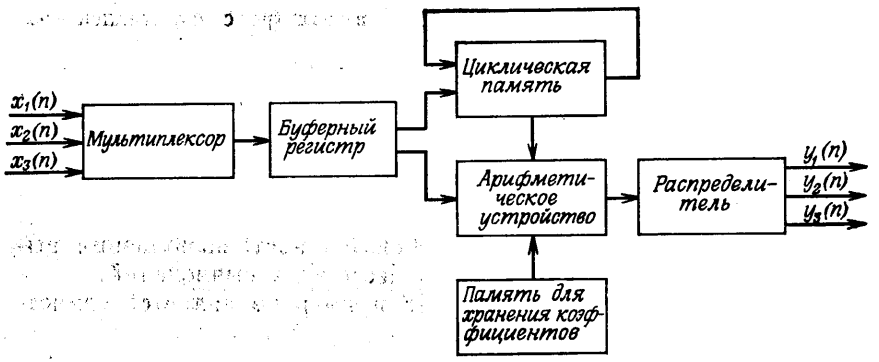
w(n)	w(n-1)	v(n)	v(n-1)	u(n)	u(n-1)

Заключительный сдвиг

Фиг. 9.15. Построение каскадного БИХ-фильтра с использованием двух АУ.

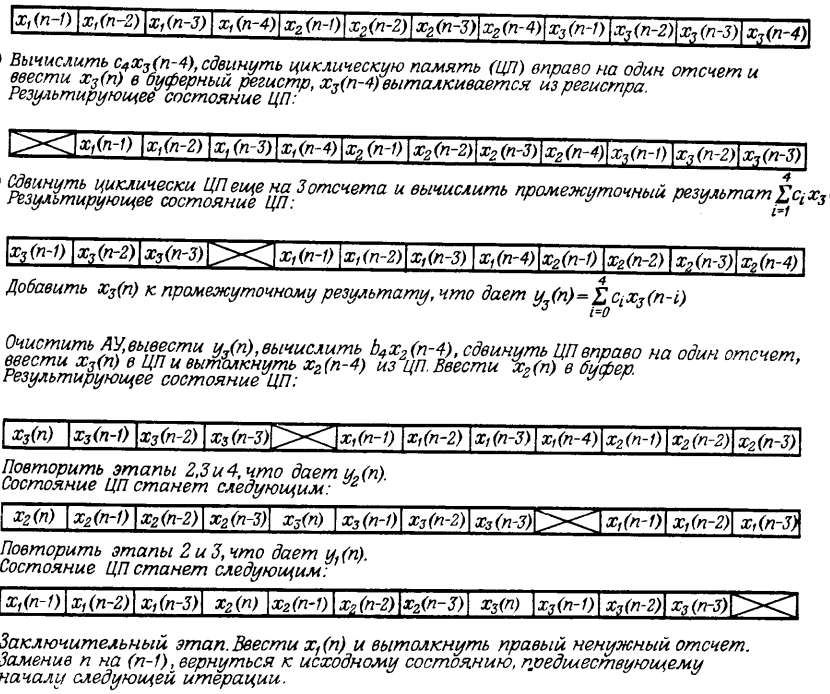


Фиг. 9.16. К иллюстрации мультиплексирования.



Фиг. 9.17. Блок-схема КИХ-фильтра с мультиплексированием.

Пример того, как с помощью мультиплексирования можно построить три независимых КИХ-фильтра, используя только одно арифметическое устройство, приведен на фиг. 9.16—9.18. На фиг. 9.16 показаны три независимых входа $x_1(n)$, $x_2(n)$ и $x_3(n)$ и три независимых выхода $y_1(n)$, $y_2'(n)$ и $y_3(n)$. Там же приведены все коэффициенты фильтров. На фиг. 9.17 изображена



Фиг. 9.18. Последовательность вычислений в фильтре с мультиплексированием.

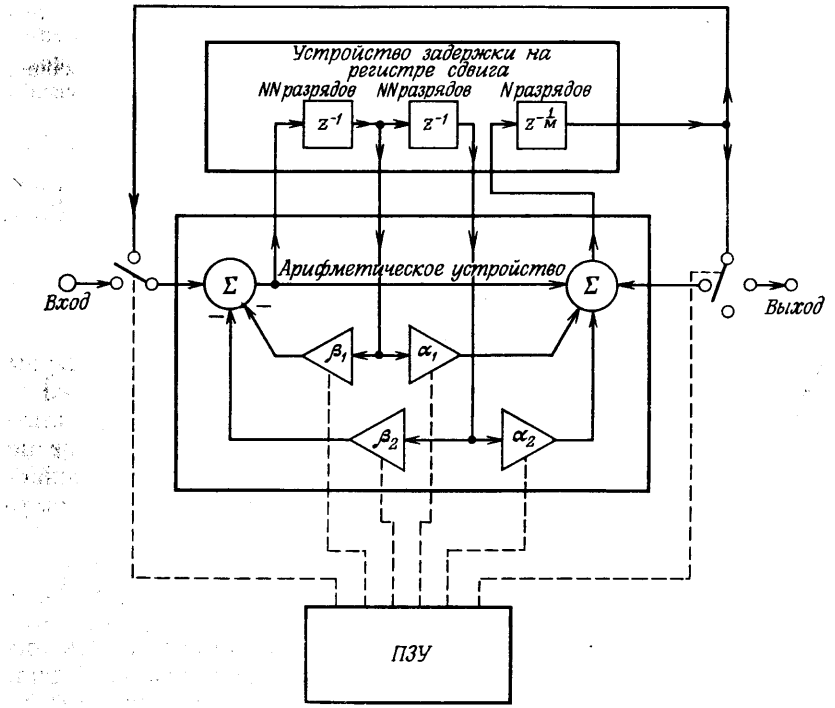
упрощенная блок-схема реализации всех трех фильтров, основанная на переключении входных сигналов и их независимой обработке в арифметическом устройстве. Распределитель обеспечивает выдачу каждого выходного отсчета в соответствующий канал. На фиг. 9.18 представлены последовательность выполнения вычислений и состояния памяти в процессе этих вычислений.

Приведенный на фиг. 9.16—9.18 пример не является единственно возможным типом мультиплексирования для цифровых устройств. Второй тип мультиплексирования имеет место при параллельном спектральном анализе, когда один и тот же входной сигнал одновременно обрабатывается с помощью набора различных фильтров. Основное внимание должно быть уделено суммарной скорости обработки данных, определяемой общим количеством фильтров, реализуемых путем мультиплексирования. До тех пор пока скорость обработки согласуется с быстродействием применяемых микросхем, система может быть построена с использованием мультиплексирования.

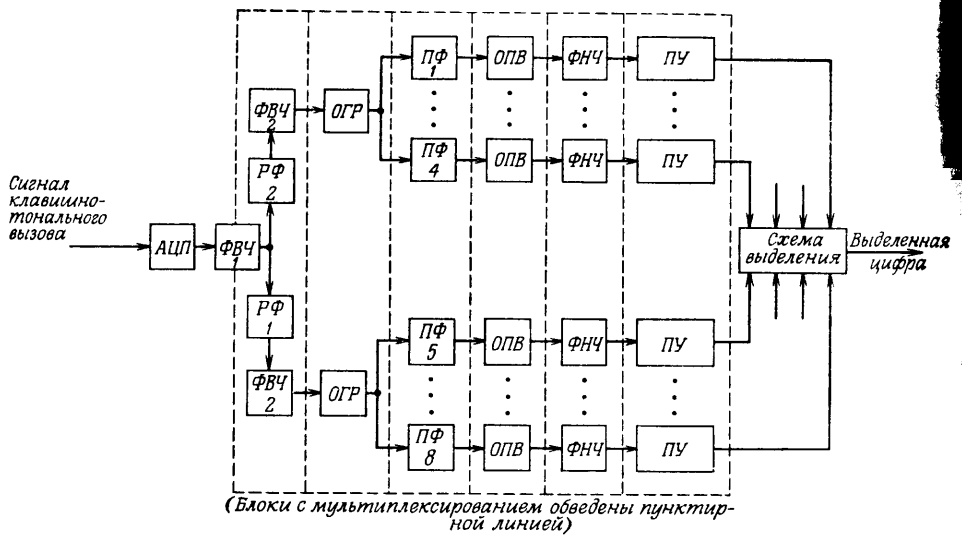
В последующих разделах будут рассмотрены несколько практических цифровых систем, построенных на основе мультиплексирования, что обеспечило преимущество цифровых методов обработки.

9.9. Цифровой приемник клавишно-тонального вызова

В качестве хорошей иллюстрации использования на практике цифровых устройств можно привести полностью цифровой приемник клавишно-тонального вызова, разработанный и сконструированный в фирме Bell, который был описан Джексоном, Кайзером и Макдональдом. На фиг. 9.19 приведена блок-схема части приемника, в которой используется мультиплексирование, а на фиг. 9.20 изображена полная схема цифрового приемника клавишно-тонального вызова, включающая как фильтры с мультиплек-



Фиг. 9.19. Блок-схема части цифрового приемника клавишно-тонального вызова, в которой используется мультиплексирование (по Джексону, Кайзеру и Макдональду).



Фиг. 9.20. Цифровой приемник клавишно-тонального вызова, содержащий фильтры с мультиплексированием и нелинейные элементы (по Джексону, Кайзеру и Макдональду).

сированием, так и элементы нелинейной обработки. Базовое арифметическое устройство, на основе которого строится каскадный БИХ-фильтр, состоит из четырех умножителей и двух трехвходовых сумматоров. Коэффициенты фильтра хранятся в ПЗУ, а промежуточные результаты — в линии задержки на регистре сдвига. Блок-схема приемника (фиг. 9.20) включает фильтры верхних частот (ФВЧ) третьего порядка, режекторные фильтры (РФ) шестого порядка, полосовые фильтры (ПФ) второго порядка и фильтры нижних частот (ФНЧ) первого порядка. Все эти фильтры построены на основе мультиплексирования, как показано на фиг. 9.19. Для нелинейной обработки в приемнике клавишно-тонального вызова используются ограничители (ОГР), однополупериодные выпрямители (ОПВ) и пороговые устройства (ПУ). Все нелинейные элементы также являются цифровыми.

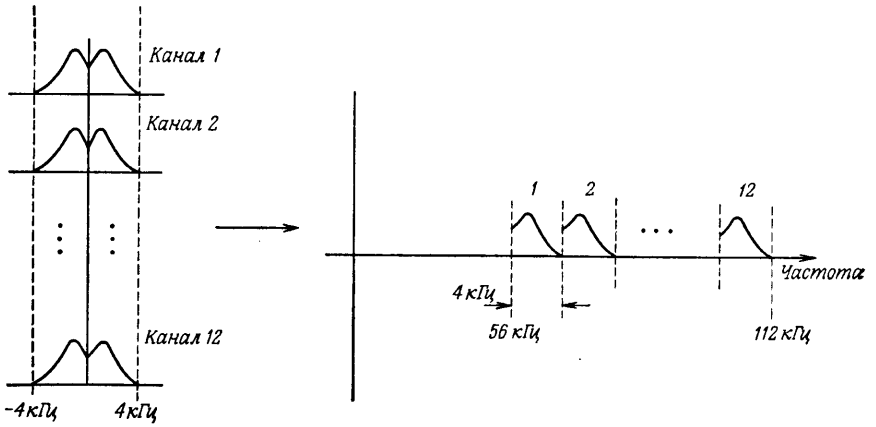
Группы по восемь однотипных элементов, обведенных пунктирными прямоугольниками, реализуются совместно с помощью мультиплексирования, так что коэффициент мультиплексирования равен восьми. Итак, для выполнения операций линейной фильтрации требуются две схемы с мультиплексированием блоков второго порядка и одна схема с мультиплексированием блоков первого порядка. Перечислим основные параметры приемника клавишно-тонального вызова:

Частота дискретизации — 10 кГц
Количество разрядов АЦП — 7
Разрядность слов — 10
Разрядность коэффициентов фильтров — 6 (правильные дроби)
Коэффициент мультиплексирования — 8

Таким образом, результирующая скорость обработки данных, которая определяется произведением частоты дискретизации, числа разрядов промежуточных результатов и коэффициента мультиплексирования, оказывается равной 800 кбит/с. В общей сложности для построения приемника клавишно-тонального вызова потребовались 40 последовательных сумматоров и память на регистрах сдвига емкостью 400 бит.

9.10. Цифровой преобразователь временного разделения каналов в частотное разделение каналов

Другой наглядный пример практического использования устройства цифровой фильтрации в крупной системе — преобразование временно́го разделения каналов в частотное разделение каналов,— описан Фрини и др. В телефонии при передаче сообщений используется чаще всего либо частотное, либо временное разделение каналов, поэтому существует необходимость преобразования одного вида разделения в другой. На фиг. 9.21 в качестве примера показано преобразование набора из 12 речевых сообщений, передаваемых с временны́м разделением, каждое из которых дискретизуется с частотой 8 кГц; преобразование заключается в размещении спектров каналов в отведенные для них участки

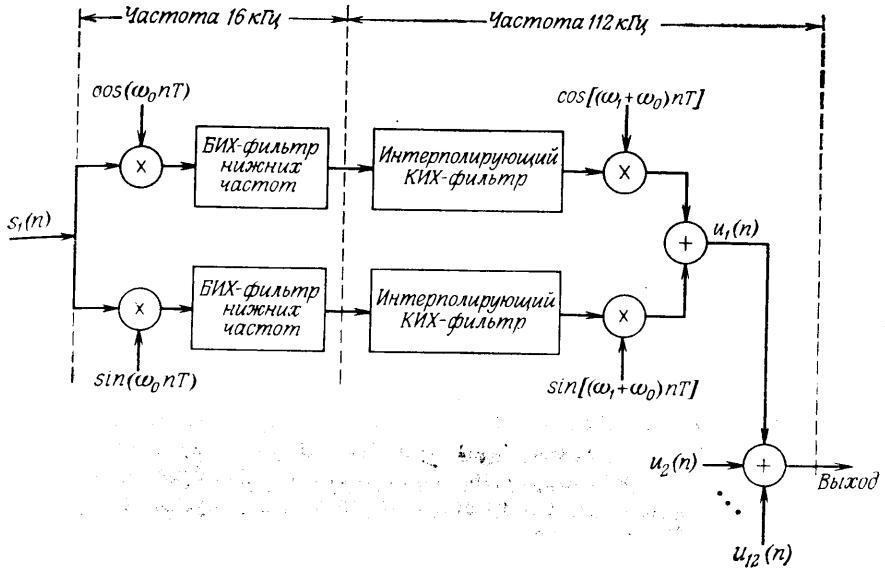


Фиг. 9.21. Преобразование 12 звуковых каналов в один канал с частотным разделением.

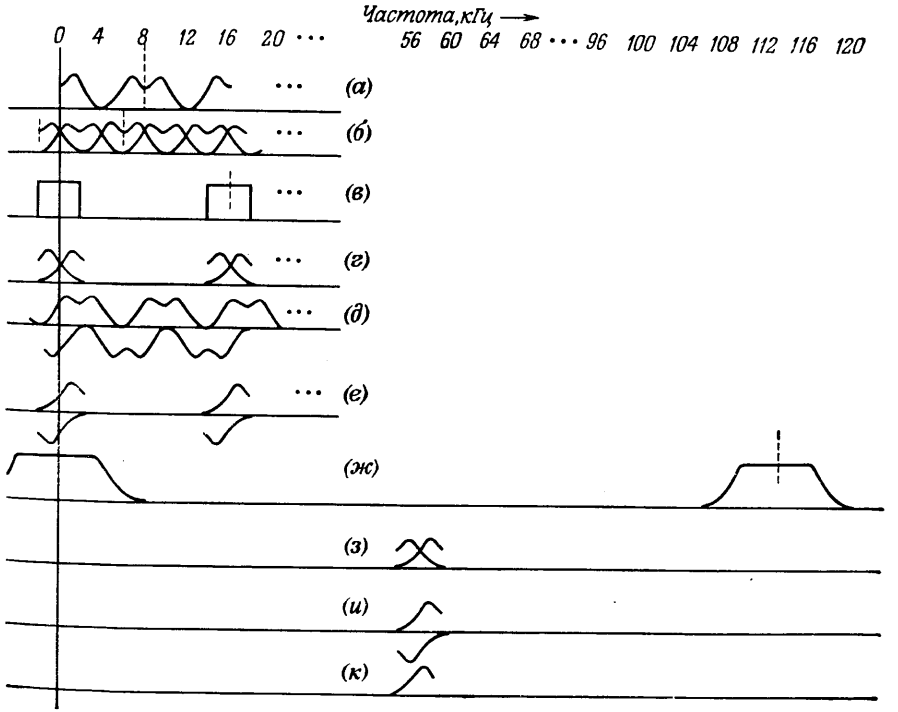
общей полосы и передаче сообщения в режиме частотного разделения каналов. В расматриваемом случае для этого выбран диапазон от 56 до 112 кГц. Каждому из 12 речевых сигналов в системе с частотным разделением отведен свой участок спектра шириной в одну боковую полосу. На каждый канал отводится всего 4 кГц, так что для передачи всех 12 каналов требуется полоса 48 кГц. Смещение частоты на 56 кГц введено для того, чтобы устранить перекрестные искажения между каналами. Таким образом, преобразование временного разделения каналов в частотное заключается в однополосном гетеродинировании, цель которого — перенести спектр на соответствующую частоту.

Чтобы избежать перекрестных искажений (т. е. помех) между соседними каналами, каждый сигнал должен быть отфильтрован в полосе 4 кГц с сильным подавлением вне этой полосы. При построении полностью цифрового устройства встречаются специфические трудности, связанные с тем, что после дискретизации спектр речевого сообщения становится периодическим и располагается на всей частотной оси. Вследствие этого обычные методы гетеродинирования здесь непригодны. Еще одна трудность возникает при попытке снизить быстродействие системы, чтобы по возможности не использовать слишком быстродействующие арифметические устройства. Если бы не эти трудности, построение преобразователя было бы достаточно простым, поскольку оно сводилось бы к пропусканию каждого дискретизованного речевого колебания через соответствующий полосовой фильтр. При таком подходе частота дискретизации должна превышать 100 кГц (точнее, она должна быть равна 112 кГц), что требует использования дорогостоящего оборудования. Чтобы преодолеть перечисленные трудности, Фрини и др. предложили схему, в которой большая часть вычислений выполняется с низкой частотой (16 кГц) и лишь для оставшихся вычислений требуется частота 112 кГц.

Блок-схема преобразования временно́го разделения каналов в частотное, предложенная Фрини и др., приведена на фиг. 9.22, а вид спектров сигналов в различных точках схемы показан на фиг. 9.23. Спектр исходного сигнала $s_1(n)$, изображенный на фиг. 9.23, *а*, является периодическим с периодом 8 кГц. После квадратурной модуляции сигнала частотой 2 кГц ($\omega_0 = 4000\,\pi$) косинусная и синусная компоненты будут иметь спектры, изображенные на фиг. 9.23, *б* и *д* соответственно. (Отметим, что все сигналы могут быть комплексными, а изображение их спектров является чисто иллюстративным.) Частотная характеристика БИХ-фильтров нижних частот с полосой 2 кГц, работающих с частотой 16 кГц, показана на фиг. 9.23, *в*. Спектры на выходах этих фильтров имеют вид кривых, изображенных на фиг. 9.23, *г* и *е*. В последующих каскадах системы частота дискретизации увеличивается в семь раз, т. е. берется равной 112 кГц, поэтому на каждый из от-



Фиг. 9.22. Блок-схема преобразования временно́го разделения каналов в частотное (по Фрини, Кибурцу, Майну и Тьюксбери).



Фиг. 9.23. Спектральные преобразования при переходе от временно́го разделения каналов к частотному.

счетов, следующих с частотой 16 кГц, приходится шесть нулевых; дополненная нулями последовательность пропускается через интерполирующий КИХ-фильтр нижних частот, характеристика которого изображена на фиг. 9.23, *ж*. Следует отметить, что так как каждые шесть из семи входных отсчетов КИХ-фильтра равны нулю, то общее количество умножений на выходной отсчет может быть уменьшено в семь раз. Таким образом, для реализации фильтра с импульсной характеристикой, содержащей 21 выборку ($N = 21$), требуется выполнить лишь три умножения на входной отсчет. Заключительная квадратурная модуляция используется для сдвига спектра сигнала в нужную полосу в сторону верхних частот, как показано на фиг. 9.23, *з* и *и*. Сложение квадратурных сигналов в выходном сумматоре позволяет выделить одну боковую спектральную полосу, изображенную на фиг. 9.23, *к*. Требуемый сигнал с частотным разделением каналов получается в результате суммирования всех 12 сигналов.

Необходимо отметить, что качество работы данной схемы зависит от того, насколько точно удается устранить путем вычитания сигналов нежелательные наложения спектров. Операции такого рода особенно успешно выполняются с помощью цифровых устройств. Другим удачным приемом, используемым в этой системе, является совместное использование БИХ-фильтра нижних частот с тактом 16 кГц и интерполирующего КИХ-фильтра нижних частот с тактом 112 кГц. Ясно, что всю фильтрацию можно было бы выполнить с тактовой частотой 112 кГц, но такой подход нецелесообразен из-за необходимости выполнять дополнительно большой объем вычислений.

Перечислим некоторые особенности построения фильтров. Порядок БИХ-фильтров нижних частот равен 9, разрядность коэффициентов равна 8, а разрядность промежуточных результатов — 22. Интерполирующие КИХ-фильтры нижних частот имеют 12-й порядок, 10-разрядные коэффициенты и 18-разрядные промежуточные результаты. Более детальные данные по этим фильтрам можно найти в статье Фрини и др.

Арифметические устройства, использованные в данной системе, аналогичны устройствам, описанным Джексоном и др. применительно к БИХ-фильтру приемника клавишно-тонального вызова, рассмотренного в разд. 9.9; модуляторы, сумматоры и КИХ-фильтры построены по сравнительно простым схемам.

9.11. Расчленение цифровых фильтров на составные части при построении их на интегральных микросхемах

При построении системы, в которой используются специализированные цифровые устройства, важную роль обычно играет ее стоимость, которая зависит от уровня интеграции применяемых интегральных микросхем и от их быстродействия. Хайтли рассмотрел методику расчленения схем применительно к построению на существующих микросхемах гребенки из 24 цифровых БИХ-фильтров четвертого порядка. Под расчленением подразумевается такое разделение памяти и арифметических устройств всей цифровой системы на небольшие функционально законченные блоки, которое делает возможным построение их с применением больших интегральных схем.

Рассмотренная Хайтли система содержала 24 канала, каждый из которых представлял собой рекурсивный БИХ-фильтр четвертого порядка (все фильтры были составлены из последовательно соединенных блоков второго порядка). Частота следования отсчетов на входах всех 24 каналов была взята равной 32 кГц, хотя входной речевой сигнал был дискретизован с частотой 8 кГц. Переход с помощью интерполяции от 8 кГц к 32 кГц был осуществлен для того, чтобы в качестве преобразователя код — аналог можно было использовать недорогой двоичный масштабный умножитель (дельта-модулятор). Разрядность промежуточных результатов фильтров была равна 16, а коэффициентов фильтра — 12. Таким образом, для запоминания всех промежуточных результатов потребовалась память объемом $16 \times 24 \times 4 = 1536$ разрядов. В каждом из мультиплексируемых блоков второго порядка используются четыре умножителя, так что общее число разрядов каждого из таких блоков равно $4 \times 12 = 48$.

Используемые интегральные микросхемы имели быстродействие 25 мГц. Хайтли расчленил систему следующим образом:

Память: 128 разрядов на корпус	→ 12 корпусов
Умножители: 4 разряда на корпус	→ 12 корпусов
Трехвходовые сумматоры: всего 2	→ 2 корпуса
Детекторы переполнения: всего 4	→ 2 корпуса
Преобразователи в дополнительный код: всего 4	→ 2 корпуса
Устройство управления: всего 1	→ 10 корпусов

Отсюда получаем, что при современном уровне технологии БИС для построения гребенки из 24 цифровых фильтров требуются 40 корпусов. По оценке Хайтли, к концу 70-х годов продолжающийся прогресс БИС-технологии позволит создать такую же систему всего на 12—14 корпусах, причем ее быстродействие будет вдвое больше, а потребляемая мощность — вдвое меньше.

9.12. Специализированный цифровой синтезатор частот

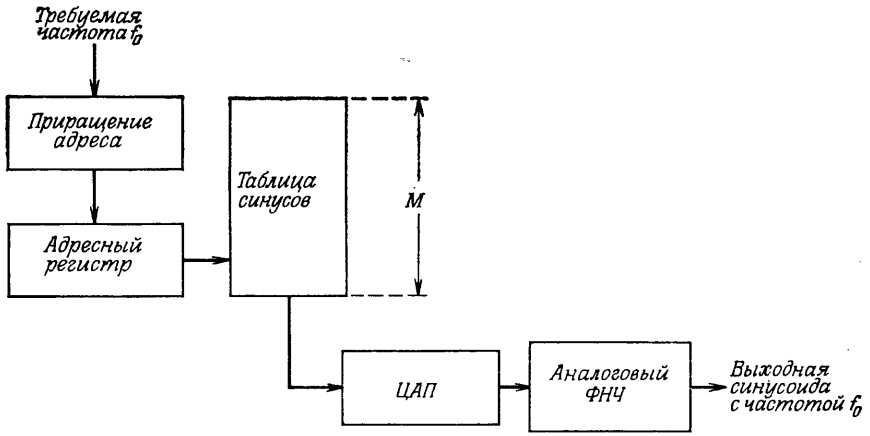
Большое внимание уделяется решению задачи синтеза синусоидальных колебаний с частотами, задаваемыми с высокой точностью. Разработано множество различных приборов, однако к моменту написания данной книги все серийно выпускаемые приборы

(кроме одного) были построены с использованием традиционных аналоговых методов, когда сигнал опорного кварцевого генератора подается на систему смесителей и фильтров для получения большого количества частот. В начале данного раздела будут рассмотрены принципы прямого цифрового синтеза синусоидальных колебаний, а затем будет описана одна из конструкций синтезатора.

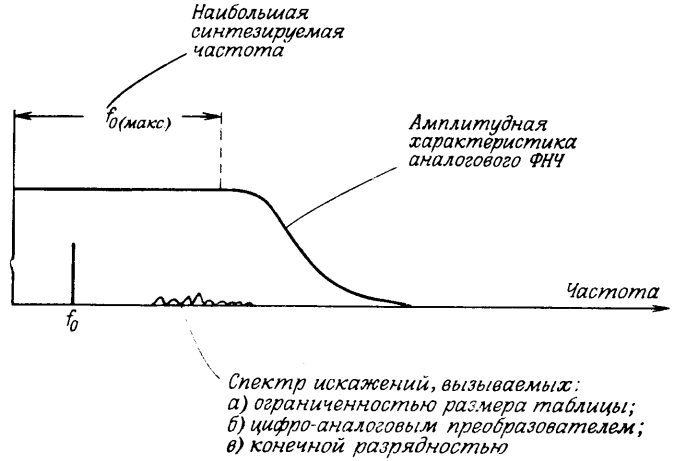
Возможны три цифровых метода получения синусоид: с использованием таблицы, путем рекурсии, а также путем сочетания обращения к таблице с вычислениями. Рассмотрим сначала проиллюстрированный на фиг. 9.24 чисто табличный метод.

Количество разрядов адресного регистра обращения к таблице синусов может превышать величину $\log_2 M$ (здесь M — размер таблицы синусов), которая необходима для вызова любого отсчета таблицы. Дело в том, что наименьшее приращение адреса определяет минимально возможное значение синтезируемой частоты. Например, таблица синусов может содержать $M = 1024$ отсчета, тогда как регистр адреса может иметь 20 разрядов. Это означает, что если приращение адреса равно единице, то 1024 раза подряд будет выбираться один и тот же отсчет синуса, после чего произойдет переход к следующему отсчету и т. д. При таких малых приращениях адреса получаемая цифровая синусоида будет очень неточной, а возникающие при этом искажения в спектре трудно устранить с помощью фильтра нижних частот.

Описанная ситуация представлена на фиг. 9.25. Для получения идеальной синусоиды необходимо, чтобы спектр искажений, обусловленных цифровым методом формирования синусоиды, располагался выше частоты среза f_c аналогового фильтра нижних частот.



Фиг. 9.24. Цифровой синтезатор частот.

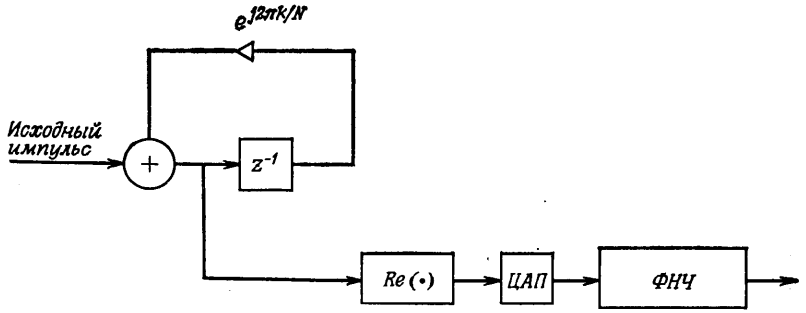


Фиг. 9.25. Шумы цифрового синтезатора частот.

Вместо выбора отсчетов синусоиды из таблицы их можно рассчитывать с помощью простой рекурсивной формулы. Действительно, пусть $x(n)$ — комплексная экспонента вида $\exp[j(2\pi nk/NT)]$. Тогда устройство, работающее согласно формуле

$$x(n) = \exp\left(j\frac{2\pi k}{NT}\right) x(n-1), \qquad (9.1)$$

будет генерировать требуемую комплексную экспоненту, причем ее действительная часть будет косинусоидой, а мнимая — синусоидой частоты $f = k/NT$. При таком подходе, если не принимать во внимание эффекты квантования, можно получить идеальную цифровую синусоиду без обращения к таблице, как показано на фиг. 9.26. Система начинает работу при поступлении внешнего единичного импульса. Изменение частоты достигается путем изменения значения k в показателе степени коэффициента умножителя,



Фиг. 9.26. Цифровой рекурсивный синтез частот.

причем предусматривается также восстановление фазы при приходе внешнего импульса или использование последнего выходного отсчета в качестве нового начального условия.

К настоящему времени синтезаторы частот рассматриваемого типа еще не построены, поскольку существует опасение, что в такой системе будут накапливаться нежелательные шумы квантования. С другой стороны, из теории предельных циклов следует, что устойчивые колебания всегда будут иметь место, однако неясно, будут ли они пригодны для получения чисто синусоидальных аналоговых колебаний. Еще одной причиной, препятствовавшей созданию устройства рассматриваемого типа, является неравномерность сетки частот, связанная с квантованием коэффициентов.

Единственный метод, на основе которого было построено цифровое устройство[1]), заключается в использовании гибридной схемы, содержащей как опрашиваемую таблицу, так и умножитель. Перечислим основные характеристики этого устройства:

количество различных частот — 2^{15};

диапазон частот — 409,6 кГц;

шаг изменения частоты (и, следовательно, минимальная частота) — 12,5 Гц;

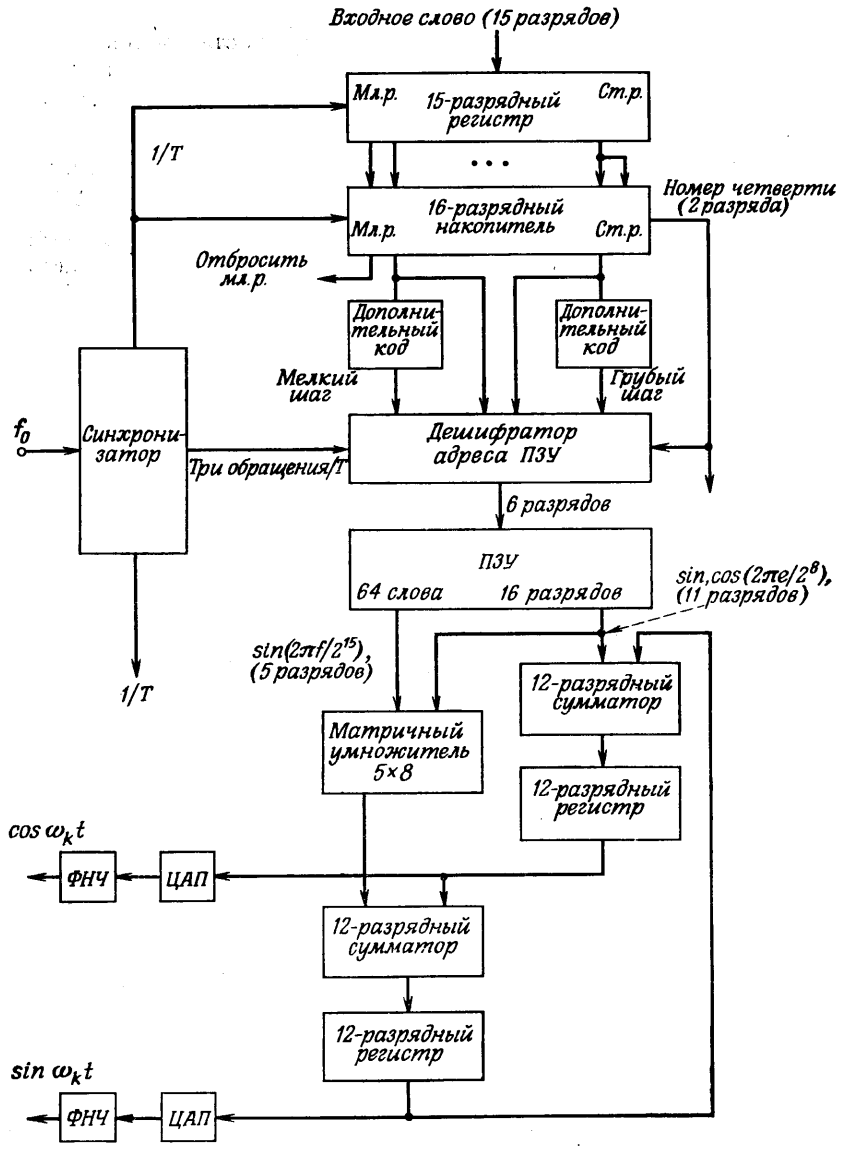
уровень чистоты спектра сигнала (определяемый как отношение мощности на заданной частоте к мощности в полосе 100 Гц на любом другом участке диапазона) равен 70 дБ.

Алгоритм работы устройства основан на простых тригонометрических формулах

$$\begin{aligned} \sin(x+y) &= \sin x \cos y + \cos x \sin y, \\ \cos(x+y) &= \cos x \cos y - \sin x \sin y' \end{aligned} \qquad (9.2)$$

что соответствует рекурсии комплексной экспоненты, описанной ранее. Отличие состоит в том, что значения x и y выбираются из таблиц, поэтому в такой системе будет отсутствовать рекурсивный шум квантования (или эффекты предельного цикла). Чтобы обеспечить перечисленные выше характеристики устройства, было принято, что x меняется с грубым шагом (это дает 2^8 различных частот), а y — с мелким шагом (что дает 2^{15} различных частот), причем значения y используются для расчета путем интерполяции промежуточных значений в соответствии с формулой (9.2). Дополнительная возможность состоит в том, что достаточно запоминать значения синуса или косинуса только на четверти периода, поэтому можно обойтись двумя блоками памяти по 64 слова каждый. Полная блок-схема цифрового синтезатора частот приведена на фиг. 9.27.

[1]) Речь идет о лабораторном макете. Первый серийный цифровой синтезатор модели 5100, выпускаемый фирмой ROCKLAND с 1975 г., основан на первом методе, см. патент США 3735269.— *Прим. перев.*



Фиг. 9.27. Блок-схема цифрового синтезатора частот (по Тирни).

По 11 разрядов в каждом из 64 16-разрядных слов отведено в памяти под отсчеты, следующие с грубым шагом. Значения $\cos y$, необходимые для перехода к мелкому шагу, с точностью до 14

бит могут быть заменены единицей, а для представления значений sin y при таких малых значениях аргумента y достаточно пяти разрядов. Таким образом, для вычисления выходного отсчета достаточно выполнить два умножения 5-разрядного числа на 8-разрядные и два сложения 12-разрядных чисел.

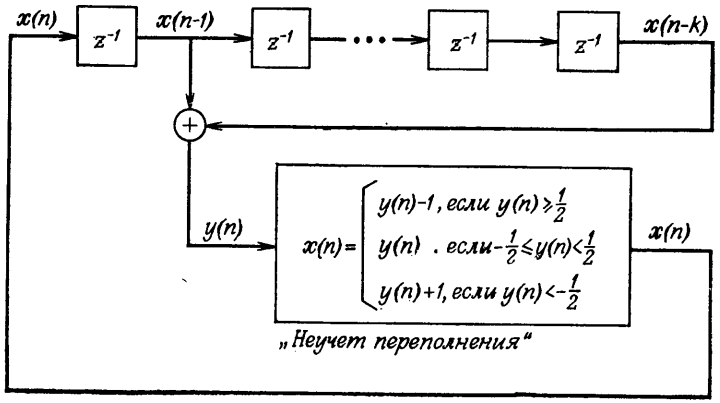
9.13. Методы генерации псевдослучайных чисел

Хотя в литературе описано много различных методов генерации последовательностей чисел с равномерным законом распределения, ниже будут коротко рассмотрены только три из них. Один из наиболее старых и распространенных методов, так называемый конгруэнтный метод, состоит в том, что очередное случайное число $x(n)$ получают из предыдущего числа $x(n-1)$ с использованием следующего правила:

$$x(n) = [A \cdot x(n-1)] \pmod{p}, \qquad (9.3)$$

где p — большое простое число, а A — соответствующим образом выбранная константа. При определенных значениях A по этому правилу в случайном порядке генерируются целые числа в интервале от 1 до $(p-1)$, причем их последовательность периодически повторяется. Преимуществами данного метода являются его простота и возможность использования в процессе вычислений памяти малого объема. Однако ему свойственна низкая скорость генерации, так как при каждой итерации приходится выполнить одно умножение (и обычно деление); кроме того, для этого метода характерна большая чувствительность к значениям A и p.

На фиг. 9.28 иллюстрируется еще один распространенный метод генерации случайных последовательностей. Предполагается,

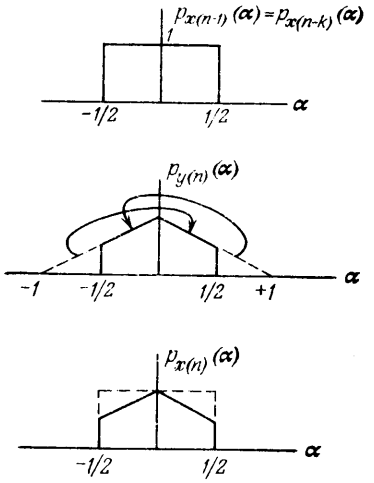


Фиг. 9.28. Метод генерации равномерно распределенных случайных чисел.

что в k регистрах памяти содержатся предварительно записанные случайные числа в диапазоне от $-1/2$ до $1/2$ (их можно взять из таблицы случайных чисел). Новое случайное число $x(n)$ генерируется по правилу

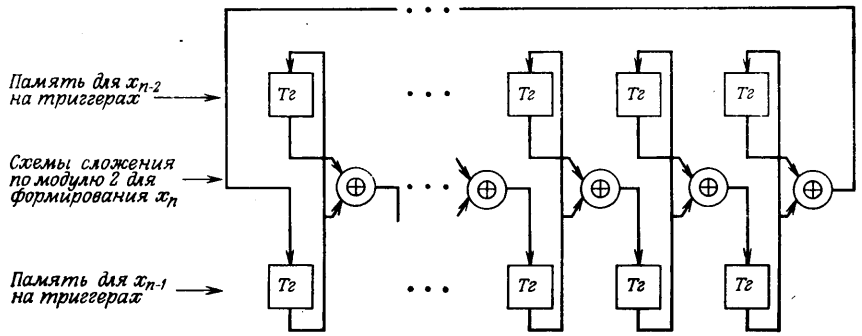
$$x(n) = [x(n-1) + x(n-k)] \left(\mathrm{mod}\ \frac{1}{2}\right), \qquad (9.4)$$

причем выполнение операции по модулю $1/2$ пояснено на фиг. 9.28 внизу: если результат сложения больше $1/2$, из него вычитается единица; если же результат меньше $-1/2$, к нему добавляется единица. Поэтому $x(n)$ будет всегда находиться в пределах от $-1/2$ до $1/2$. Из фиг. 9.29 видно, что если, как и предполагалось, случайные величины $x(n-1)$ и $x(n-k)$ равномерно распределены на одном и том же интервале $(-1/2, 1/2)$, то и случайная величина $x(n)$ также будет распределена равномерно на том же интервале, так что именно в результате взятия значения суммы по модулю $1/2$ треугольное распределение, получающееся после суммирования двух равномерно распределенных величин, опять преобразуется к равномерному. Это означает, что если использовать представление чисел в дополнительном коде, то при условии, что максимально возможное машинное число равно $1/2$, вычисления с использованием формулы (9.4) можно выполнить, просто складывая числа в дополнительном коде и не учитывая переполнений. Итак, преимущество рассматриваемого метода состоит в том, что все вычисления в пределах итерации сводятся к одному сложению. Кроме того, для хранения чисел от $x(n-1)$ до $x(n-k)$ обычно требуется порядка 50 регистров. Измерения статистических свойств образуемых таким методом псевдослучайных последовательностей показывают, что их распределения близки к равномерному, причем их энергетический спектр тоже почти равномерный, т. е. генерируемый шум близок к белому. Таким образом, рассмотренный генератор можно с успехом использовать для цифровой обработки сигналов.

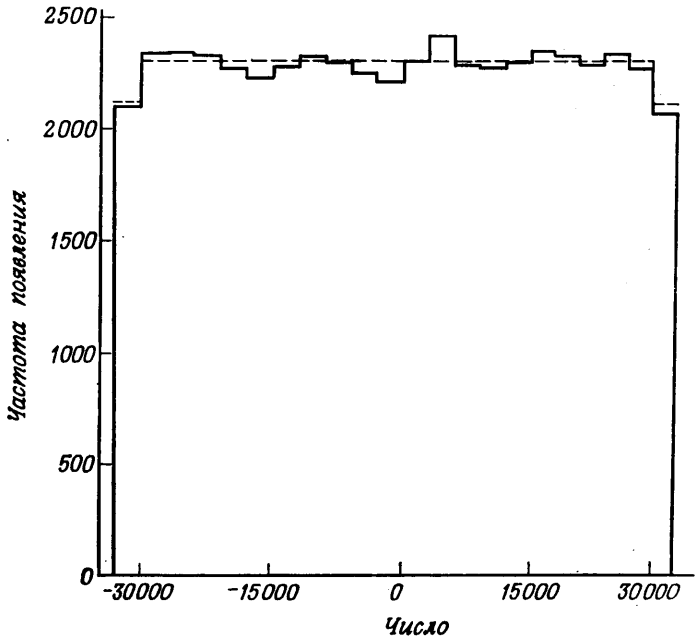


Фиг. 9.29. Пояснение заворачивания при переполнениях, используемого для генерации равномерно распределенных случайных чисел.

Третий метод получения случайных чисел иллюстрируется на фиг. 9.30. Формирование очередного L-разрядного случайного



Фиг. 9.30. Блок-схема генератора псевдослучайных чисел с равномерным распределением.



Фиг. 9.31. Гистограмма генератора случайных чисел с равномерным распределением. (Количество слагаемых = 1, размер выборки = 49 984, среднее = 5,16, дисперсия = 18 909,97.)

числа $x(n)$ осуществляется на основе двух предшествующих отсчетов $x(n-1)$ и $x(n-2)$ по следующему правилу:

$$x(n) = T_p[x(n-1) + x(n-2)], \tag{9.5}$$

причем символ $T_p[\cdot]$ обозначает циклический сдвиг вправо на P разрядов, а сложение является поразрядным и выполняется по модулю 2.

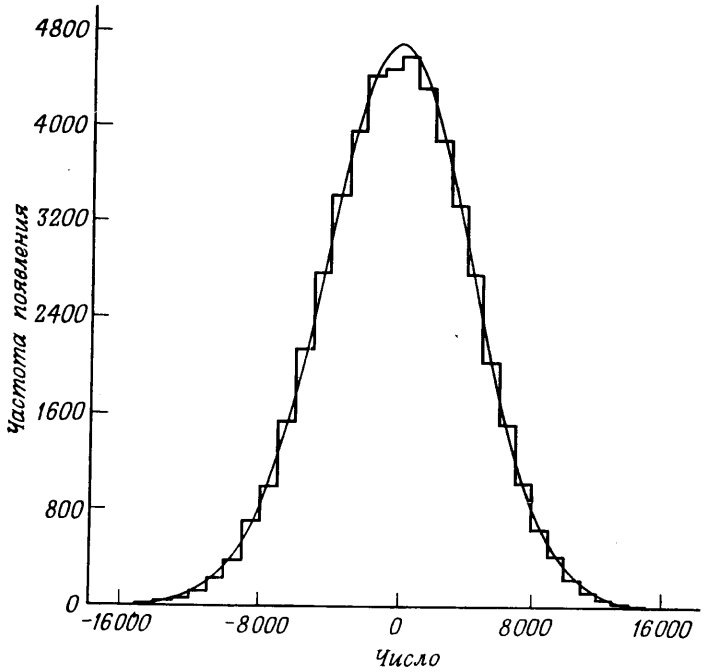
В схеме на фиг. 9.30 принято, что $P = 1$. Данному алгоритму свойственны высокие быстродействие и эффективность. Единственная выполняемая здесь операция — L-разрядное сложение по модулю 2, а объем памяти ограничен двумя L-разрядными словами для хранения $x(n-1)$ и $x(n-2)$. Показано, что при определенных значениях L распределение генерируемых чисел приблизительно равномерное, а спектральная плотность соответствует белому шуму. Например, при $L = 19$ период последовательности будет равен 14942265, т. е. примерно 1500 с при частоте дискретизации 10 кГц. Благодаря своей простоте данный генератор был построен в виде специализированного цифрового устройства с $L = 19$. На фиг. 9.31 приведена гистограмма, полученная приблизительно по 50 000 выходным отсчетам, которые были пронормированы таким образом, что оказались лежащими в диапазоне (—32 768, 32 767). Пунктиром показано теоретическое значение частоты повторения, соответствующее равномерному распределению (ее значения на краях диапазона несколько меньше, чем в остальной его части, что свидетельствует о меньшей вероятности генерации чисел вблизи краев диапазона). В целом равномерное распределение аппроксимируется достаточно хорошо.

9.14. Методы генерации гауссовых случайных чисел

Методы генерации случайных чисел, описанные в предыдущем разделе, позволяют получить последовательности только с равномерным распределением. Однако довольно часто требуются последовательности псевдослучайных чисел с гауссовым распределением. По-видимому, проще всего для их генерации воспользоваться центральной предельной теоремой, согласно которой распределение суммы N одинаково распределенных независимых случайных величин стремится к нормальному, когда N стремится к бесконечности. Поэтому последовательность независимых равномерно распределенных чисел $\{x(n)\}$ можно преобразовать в последовательность чисел с гауссовым распределением $\{y(n)\}$, используя следующее правило:

$$y(n) = \frac{1}{N} \sum_{i=0}^{N-1} x(nN-i). \tag{9.6}$$

Здесь N должно быть достаточно большим. Во всех встречающихся на практике случаях достаточно хорошее приближение к гауссову распределению обеспечивается при N порядка 10. Это видно из фиг. 9.32, где представлена измеренная гистограмма, причем



Фиг. 9.32. Гистограмма генератора чисел с нормальным распределением (Количество слагаемых = 12, размер выборки =49 984, среднее = —33,69, дисперсия = 4262,36.)

случайные числа формировались по схеме фиг. 9.30 ($L = 19$) при $N = 12$. Видно, что экспериментальные результаты хорошо согласуются с теоретической кривой, соответствующей нормальному распределению.

Одна из разновидностей рассматриваемого метода, предложенная Рэйдером, состоит в том, что последовательность из L независимых равномерно распределенных случайных величин преобразуется с помощью матрицы Адамара в новую последовательность из L некоррелированных случайных величин с гауссовым распределением. Каждая из L гауссовых величин была получена путем суммирования (вычитания) L чисел с равномерным распределением. Поэтому при $L > 16$ наблюдалось достаточно хорошее приближение к нормальному распределению. Такой подход весьма эффективен, поскольку по L величинам с равномерным распределением сразу получаются L нормальных случайных величин, а не L/N, как при использовании формулы (9.6). Правда, выходные отсчеты уже не будут независимыми, хотя с помощью простой модификации можно добиться того, что они станут почти независимыми.

Существует прямой метод преобразования пары равномерно распределенных случайных величин в пару нормальных случайных величин. Если обозначить последовательность равномерно распределенных на интервале (0, 1) случайных величин через $\{x(n)\}$ и определить $y(n)$ как

$$y(n) = \sqrt{2\sigma^2 \ln [1/x(n)]}, \tag{9.7}$$

то $y(n)$ будет иметь релеевское распределение, т. е.

$$P_y(y_0) = \frac{y_0}{\sigma^2} \exp\left(\frac{-y_0^2}{2\sigma^2}\right). \tag{9.8}$$

Если затем сформировать две новые случайные величины $w(n)$ и $w(n+1)$ согласно формулам

$$w(n) = y(n) \cos [2\pi x (n+1)], \tag{9.9}$$

$$w(n+1) = y(n) \sin [2\pi x (n+1)], \tag{9.10}$$

то обе новые величины будут иметь нормальное распределение с нулевым средним и дисперсией, равной σ^2. Более того, $w(n)$ и $w(n+1)$ оказываются при этом некоррелированными, что для нормальных величин эквивалентно их независимости. Хотя описанный метод и дает на практике хорошие результаты, он связан с довольно большими затратами времени, поскольку приходится вычислять логарифмы, квадратные корни, синусы и косинусы, так что для генерации большого массива нормальных случайных величин его стараются не использовать.

ЛИТЕРАТУРА

1. Jackson L. B., Kaiser J. F., McDonald H. S., An Approach to the Implementation of Digital Filters, *IEEE Trans. on Audio and Electroacoustics*, **16**, No. 3, 413—421 (Sept. 1968).
2. Hightley J. D., Partitioning of Digital Filters for Integrated-Circuit Realization, *IEEE Trans. on Communication Tech.*, **COM-19**, 1059—1063 (Dec. 1971).
3. Freeny S. L., Kieburtz R. B., Mina K. V., Tewksbury S. K., Design of Digital Filters for an All Digital Frequency Division Multiplex-Time Division Multiplex Translator, *IEEE Trans. Circuit Theory*, **CT-18**, 702—711 (Nov. 1971).
4. Freeny S. L., Kieburtz R. B., Mina K. V., Tewksbury S. K., Systems Analysis of a TDM-FDM Translator/Digital A-Type Channel Bank, *IEEE Trans. on Communication Tech.*, **COM-19**, 1050—1059 (Dec. 1971).
5. Kurth C. F., SSB/FDM Utilizing TDM Digital Filters, *IEEE Trans. on Communication Tech.*, **COM-19**, 1, 63—70 (Feb. 1971).
6. Tierney J., Rader C. M., Gold B., A Digital Frequency Synthesizer, *IEEE*

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.

Глава 10

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, засылаются 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.