

Фиг. 8.4. Последовательность рассмотрения элементов систем в процессе создания алгоритмов обработки сигналов.

моделирования на ЦВМ. Как производительность, так и сложность систем зависят от характеристик арифметического устройства (представление чисел с фиксированной или плавающей запятой, контроль переполнения, основание системы счисления). С помощью программы моделирования можно подобрать большое количество параметров системы как до ее разработки и конструирования, так и во время конструирования, но в меньшей степени.

Помимо моделирования, целесообразно также разработать и собрать некоторые блоки. Это даст возможность оценить быстродействие, количество корпусов интегральных схем, стоимость и затраты времени. Сведения о компонентах берутся из справочников, а логическое проектирование выполняется с использованием булевой алгебры. Многие неопытные разработчики придают слишком большое значение логическим уравнениям, подобно тому как начинающие программисты уделяют слишком большое внимание отдельным командам программы. По нашему мнению, эти этапы не являются решающими при проектировании и создании системы цифровой обработки сигналов.

На фиг. 8.4 схематично показана последовательность анализа элементов схем, блоков и систем (вместе с указанием источников

используемой информации), предшествующего разработке алгоритмов. Эта схема в принципе справедлива применительно к любой цифровой системе, поэтому имеет смысл кратко рассмотреть основные особенности алгоритмов цифровой обработки сигналов.

При разработке этих алгоритмов очень важным фактором является скорость вычислений. При достаточной скорости можно работать в реальном времени, уменьшить параллелизм системы (и, как следствие, упростить ее), сэкономить машинное время универсальных ЦВМ, повысить степень мультиплексирования. Из-за стремления к увеличению быстродействия транзисторно-транзисторная логика (ТТЛ) и эмиттерно-связанная логика (ЭСЛ) представляют больший интерес, чем резистивно-транзисторная логика (РТЛ).

Основными арифметическими операциями при цифровой обработке сигналов являются умножение и сложение. Для многих аппаратурных или программных реализаций алгоритмов обработки сигналов особенно важна разработка быстродействующих компактных и недорогих умножителей.

### 8.3. Булева алгебра.

#### Примеры простых логических цепей

При разработке цифровых устройств универсальным языком для их описания служит двоичная арифметика, использующая булеву алгебру. Таким образом, любой алгоритм можно представить системой булевых уравнений. Справочные данные для логических интегральных схем всегда содержат логические соотношения, связывающие вход микросхемы с ее выходом. Системы обозначений не во всех справочниках одинаковые, поэтому здесь произвольно выбрана одна из самых распространенных.

Любая двоичная функция может быть выражена через элементарные функции И, ИЛИ и НЕ. Пусть  $x$  и  $y$  — две двоичные переменные, а  $u$  — двоичная функция от  $x$  и  $y$ . Тогда если  $u = 1$ , только когда  $x = 1$  и  $y = 1$ , то  $u$  определяется как функция И от  $x$  и  $y$  и записывается в виде

$$u = xy. \quad (8.1)$$

Полную картину функциональной зависимости между  $x$ ,  $y$  и  $u$  дает также следующая таблица истинности:

$x$	$y$	$u$
0	0	0
0	1	0
1	0	0
1	1	1

(Т. И. 1)

Таблица истинности показывает, чему равно  $u$  для каждой из возможных комбинаций  $x$  и  $y$ . Формула (8.1) и таблица (Т.И.1) полностью эквивалентны. Очень часто алгебраические выражения типа (8.1) получают из таблицы истинности, а затем упрощают их, используя правила алгебры логики.

Если  $u = 1$ , когда либо  $x = 1$  либо  $y = 1$ , то  $u$  называется функцией ИЛИ от  $x$  и  $y$  и записывается следующим образом:

$$u = x + y. \quad (8.2)$$

Можно образовать и более сложные функции. Пусть, например,  $x_1, x_2$  и  $x_3$  — двоичные переменные и

$$u = x_1 x_2 + x_3. \quad (8.3)$$

Из этой формулы следует, что если  $x_1 = 1$  и  $x_2 = 1$  или  $x_3 = 1$ , то  $u = 1$ . Попробуйте в качестве упражнения составить таблицы истинности, соответствующие соотношениям (8.2) и (8.3).

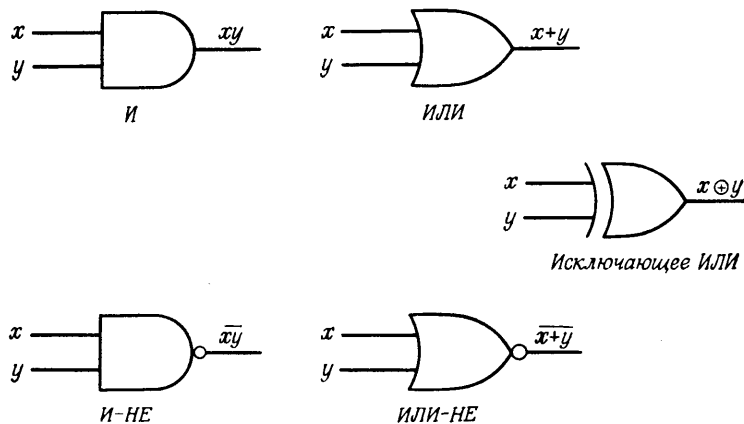
Введение отрицаний придает действиям с булевыми функциями большую гибкость. Отрицание  $x$ , обозначаемое как  $\bar{x}$ , равно единице, если  $x = 0$ , и наоборот. Уравнения

$$u = \overline{xy}, \quad u = \overline{x+y} \quad (8.4)$$

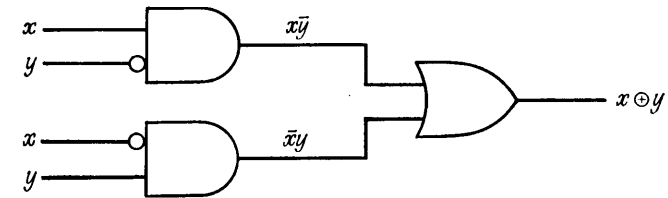
описывают функции, называемые И—НЕ и ИЛИ—НЕ. Операция неравнозначности (называемая иначе сложением по модулю 2) определяется как

$$u = \overline{xy} + \overline{x+y} = x \oplus y. \quad (8.5)$$

Пять булевых функций ИЛИ, ИЛИ—НЕ, И, И—НЕ и неравнозначность составляют основу более сложных логических выражений. На фиг. 8.5 представлены условные обозначения, которые



Фиг. 8.5. Примеры булевых функций



Фиг. 8.6. Функция неравнозначности, выраженная через функции И и ИЛИ.

будут использоваться в дальнейшем при составлении логических блок-схем, соответствующих или булевым уравнениям, или таблицам истинности (маленькие кружки здесь и ниже обозначают отрицание).

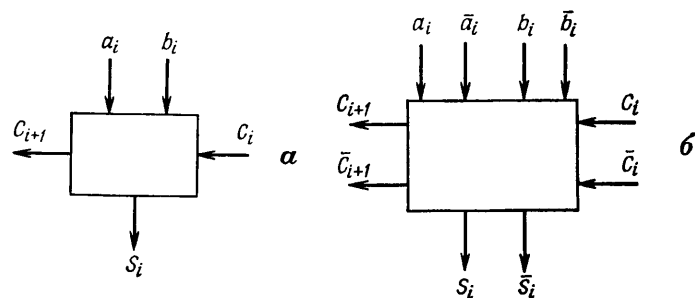
На фиг. 8.6 показан пример реализации функции неравнозначности на основе вентилях И и ИЛИ. Существенно, что для этого требуются два уровня вентилях, поэтому следует ожидать, что время переключения в этом случае будет больше, чем для четырех элементарных функций, показанных на фиг. 8.5. Время переключения является определяющей характеристикой логического элемента. Для серийно выпускаемых интегральных схем (ИС) различают два вида задержек — номинальную и максимальную. Эти задержки зависят также от температуры и нагрузки (количества входов ИС, подключенных к выходу схемы). Увеличение нагрузки усложняет работу схемы и увеличивает задержку распространения. Различают два вида нагрузочной способности — по постоянному и переменному току. Ограничение на нагрузочную способность по переменному току связано с тем, что каждый из входов интегральной схемы имеет некоторую паразитную емкость, поэтому при подключении к данной схеме большого количества входов емкостная нагрузка вызовет замедление процесса переключения. С другой стороны, нагрузка по постоянному току вызывает изменение переходной характеристики транзисторов, уменьшая логический перепад и ухудшая таким образом помехоустойчивость схемы.

В качестве упражнения предлагаем читателю следующие задания.

1. Составьте пирамидальную схему контроля четности восьми двоичных разрядов, вырабатывающую логическую единицу в четном случае и нуль в нечетном.

2. Составьте схему восьмиканального селектора, который в зависимости от входного трехразрядного кода выбирает одну из восьми входных линий.

3. Определите наибольшее из двух трехразрядных чисел. Считается, что устройство оперирует только с положительными числами, такими, например, как  $010 = 2$  и  $111 = 7$ .



Фиг. 8.7. Входные и выходные сигналы одноразрядного сумматора.

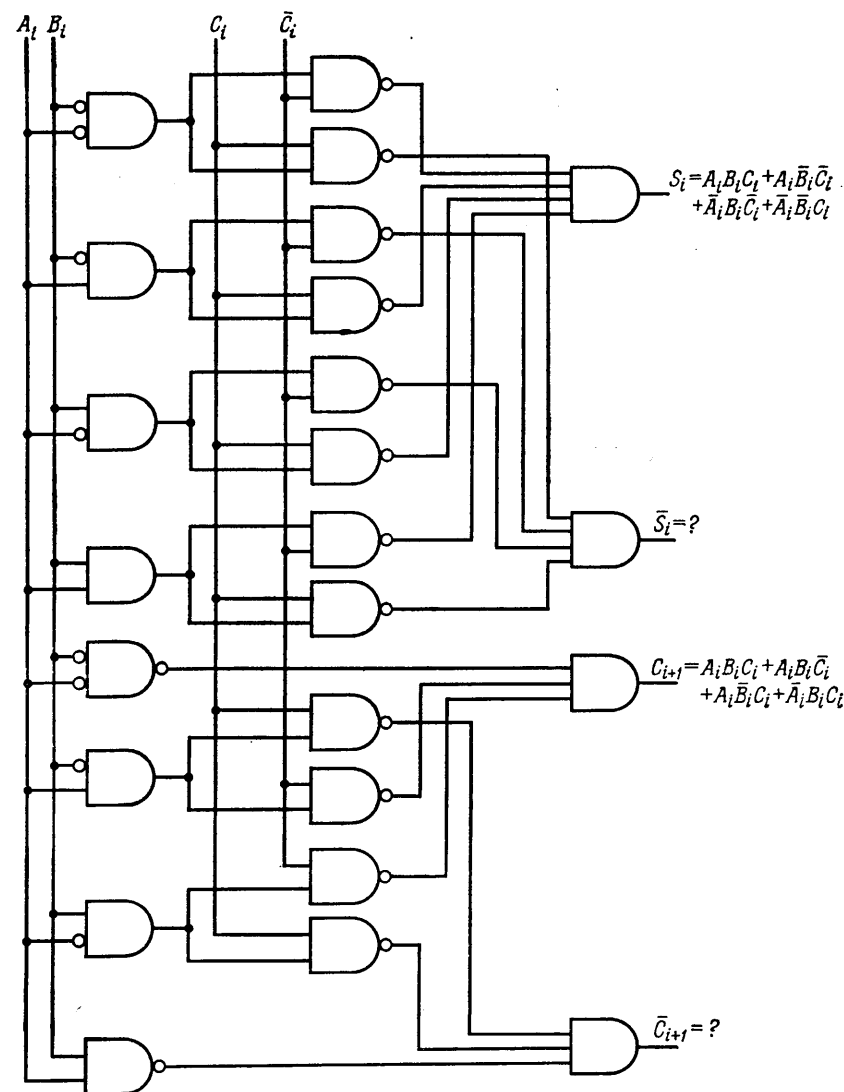
Одна из наиболее часто используемых булевых функций при цифровой обработке сигналов формируется с помощью одноразрядного сумматора (фиг. 8.7), имеющего три входа (два для слагаемых и один для переноса) и два выхода — суммы и переноса (в более совершенных схемах и их дополнений). Взаимосвязь между входами и выходами может быть задана следующей таблицей истинности:

$a$	$b$	$c_{вх}$	Арифметическая сумма	$c_{вых}$	$s$	
0	0	0	0	0	0	
0	0	1	1	0	1	
0	1	0	1	0	1	
0	1	1	2	1	0	(Т.И.2)
1	0	0	1	0	1	
1	0	1	2	1	0	
1	1	0	2	1	0	
1	1	1	3	1	1	

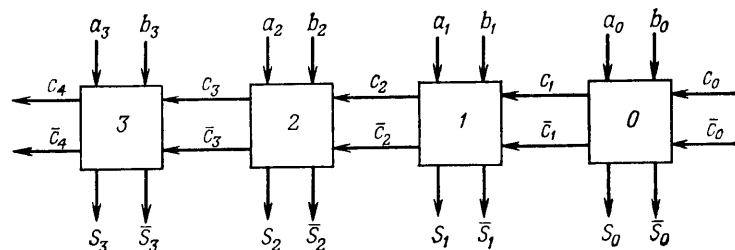
Логические уравнения полностью определены этой таблицей, поэтому непосредственно по ней можно составить булевы выражения для  $c_{вых}$  и  $s$  в функции  $a$ ,  $b$  и  $c_{вх}$ . Конкретный пример серийно выпускаемой интегральной схемы, выполняющей рассматриваемую функцию, показан на фиг. 8.8. Предлагаем читателю следующие упражнения:

1. Составьте таблицу истинности и запишите логические уравнения для одноразрядного вычитателя, работающего в дополнительном коде.

2. На фиг. 8.9 показан четырехразрядный сумматор. Необходимо оценить время «установления» сумматора, т. е. в предположении, что в нулевой момент времени на все входы подаются логические нули.



Фиг. 8.8. Логическая схема одноразрядного сумматора фирмы Motorola.



Фиг. 8.9. Четырехразрядный сумматор.

гические уровни, определить, когда логические уровни всех сумм и переносов станут достаточно стабильными, чтобы их можно было использовать в качестве входных сигналов для других логических элементов системы. Чтобы решить эту задачу, необходимо знать время распространения от каждого входа до любого выхода. Пусть  $\tau(a_i \rightarrow s_i) = \tau(b_i \rightarrow s_i) = 10$  нс,  $\tau(c_i \rightarrow s_i) = 15$  нс,  $\tau(a_i \rightarrow c_{i+1}) = \tau(b_i \rightarrow c_{i+1}) = 20$  нс и  $\tau(c_i \rightarrow c_{i+1}) = 5$  нс. Определите для этих значений наибольшее время распространения.

3. Параллельный сумматор, состоящий из восьми одноразрядных сумматоров, параметры которого приведены в предыдущем примере, имеет недостаточное быстродействие. Рассмотрите несколько схем ускорения переноса, попытайтесь составить их логические уравнения и обоснуйте выбор наилучшей из них. Предполагается, что задержка распространения базового логического элемента равна 3 нс.

4. Для схемы, приведенной на фиг. 8.8, напишите логические уравнения для  $\bar{S}_i$  и  $\bar{C}_{i+1}$ ; покажите, что  $S_i$  и  $C_{i+1}$  также можно представить в виде

$$S_i = A_i \oplus B_i \oplus C_i,$$

$$C_{i+1} = A_i B_i + (A_i \oplus B_i) C_i.$$

#### 8.4. Основные типы интегральных логических схем

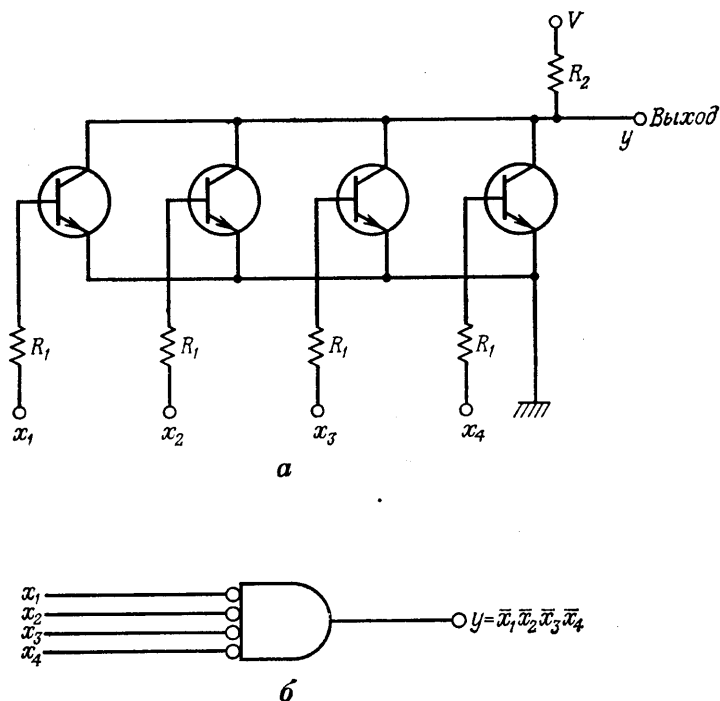
Как уже отмечалось, проектирование аппаратуры в некотором смысле напоминает программирование. Программист пытается реализовать алгоритм, используя набор команд данной универсальной ЦВМ. Аналогично разработчик обычно проектирует аппаратуру, реализующую тот или иной алгоритм, используя булевы уравнения с учетом специфики выбранной серии логических элементов. Основное различие между этими двумя направлениями состоит в следующем: программист может считать свою систему команд «идеальной» в том смысле, что каждая из них выполняется точно; логическая схема также будет «идеальной»

при условии, что конструирование проведено грамотно, помехи малы, температурная проблема решена и т. д. Поэтому разработчик аппаратуры вынужден иметь гораздо больше дополнительных сведений о логических схемах, чем программист об освоенной им системе команд. В этом разделе рассмотрены основные существующие в настоящее время типы логических схем и их классификация; приведены некоторые спецификации выпускаемых промышленностью логических схем и показано, насколько важны для разработчика эти сведения.

Прежде всего, логические интегральные схемы подразделяются на *биполярные* и *униполярные*. Биполярные ИС представляют собой обычные транзисторные схемы, использующие оба типа проводимости (электронную и дырочную), тогда как униполярные основаны на полевом эффекте, заключающемся в том, что напряжение затвора управляет протекающим в любом направлении между истоком и стоком потоком зарядов одного знака. Внутри каждого из основных классов ИС проводится более детальная классификация. Она будет рассмотрена ниже, а сейчас лишь отметим, что униполярные устройства до сих пор представляются наиболее подходящими для схем с высоким уровнем интеграции, низкой потребляемой мощностью и умеренным быстродействием, тогда как большинство биполярных схем позволяет достичь очень высокого быстродействия, но при большей рассеиваемой мощности и меньшем уровне интеграции. Например, при существующем уровне технологии МДП (металл—диэлектрик—полупроводник) на одном кристалле можно разместить довольно сложный цифровой фильтр; плотность размещения компонентов униполярных схем на один-два порядка выше, чем биполярных. Биполярные схемы классифицируют путем сравнения режимов транзисторных цепей: транзисторы в процессе переключения либо заходят в ту область переходной характеристики, где наблюдается насыщение (насыщенная логика), либо они всегда работают на линейном участке (ненасыщенная логика). Рассмотрим сначала два типа насыщенных ИС, а затем один тип ненасыщенных ИС. На фиг. 8.10 показан вентиль с четырьмя входами резистивно-транзисторной логики (РТЛ). Если четыре входных уровня  $x_1, x_2, x_3$  и  $x_4$  низкие (нулевые), то все транзисторы закрыты, через  $R_2$  ток не течет и поэтому напряжение на выходе  $y$  равно  $V$ . Если же на любом из входов напряжение возрастает до достаточно высокого уровня, начинает течь ток и напряжение на выходе падает. Таким образом, переключающая схема, показанная на фиг. 8.10, имеет булеву функцию вида

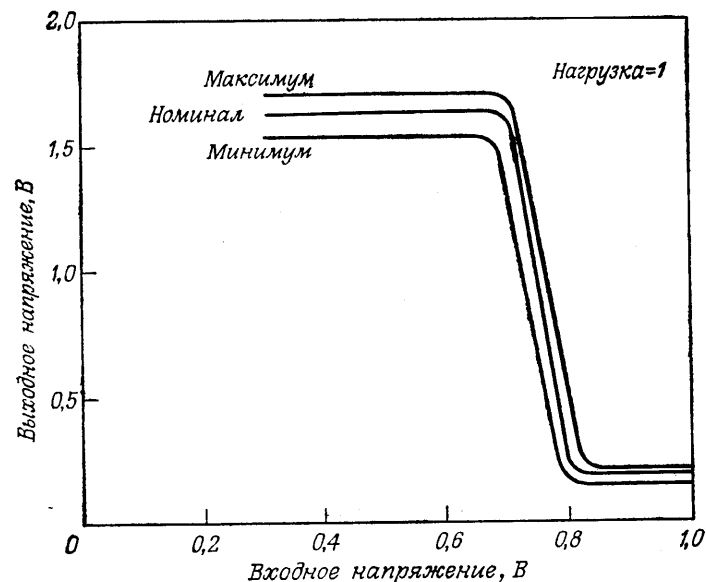
$$y = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4, \quad (8.6)$$

т. е.  $y = 1$  тогда и только тогда, когда все  $x_i = 0$ .

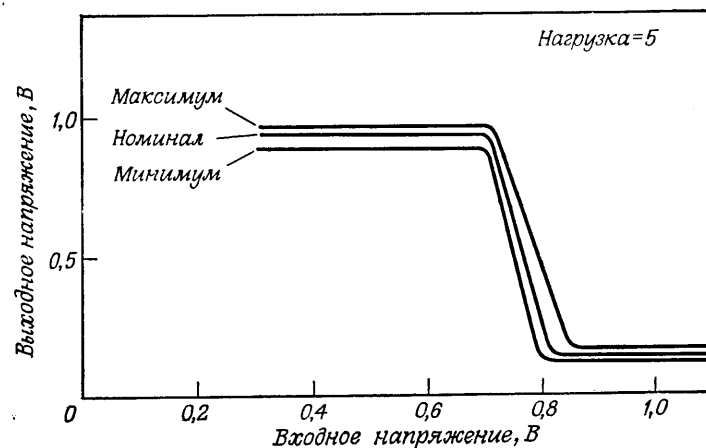


Фиг. 8.10. Принципиальная схема и условное обозначение вентиля НЕ—И с четырьмя входами (по Гаррету).  
а — принципиальная схема; б — условное обозначение.

В этом примере предполагалось, что высокому уровню напряжения соответствует логическая единица, а низкому — логический нуль. Но чему равны эти напряжения, каковы значения потребляемой мощности, какие типы транзисторов использовать и как выбрать номиналы резисторов? Все эти характеристики, а также возможности технологии по обеспечению достаточно точных значений параметров будут определять быстродействие, надежность, помехоустойчивость и т. д. Сведения такого рода можно получить из каталогов. Например, на фиг. 8.11 показана передаточная характеристика схемы РТЛ. В этих схемах применяются транзисторы с большим коэффициентом усиления  $\beta$ , поэтому относительно небольшое изменение входного уровня приводит к насыщению. Из фиг. 8.11 отчетливо видно, почему эта транзисторная схема ведет себя как переключатель, и становится понятным, какой смысл имеет термин «насыщенная логика». Следует также обратить внимание на характер изменения передаточной характеристики этой схемы по постоянному току при увеличении нагрузки

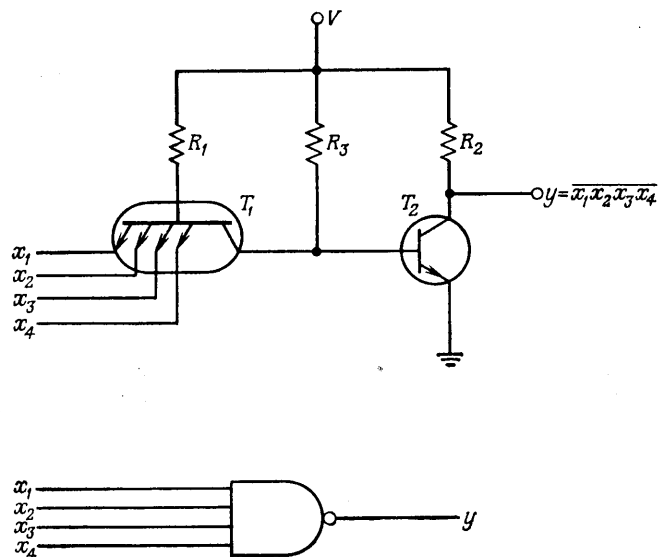


Фиг. 8.11. Передаточная характеристика по напряжению для схемы РТЛ (по Гаррету).



Фиг. 8.12. Передаточная характеристика схемы РТЛ с нагрузкой, равной пяти (по Гаррету).

от единицы (фиг. 8.11) до пяти (фиг. 8.12). Видно, что вентиль, нагруженный на пять других вентилях, имеет несколько худший режим переключения по сравнению с менее нагруженным вентилем из-за уменьшения перепада напряжения примерно на 30%.



Фиг. 8.13. Упрощенная схема ТТЛ и ее условное обозначение (по Гаррету).

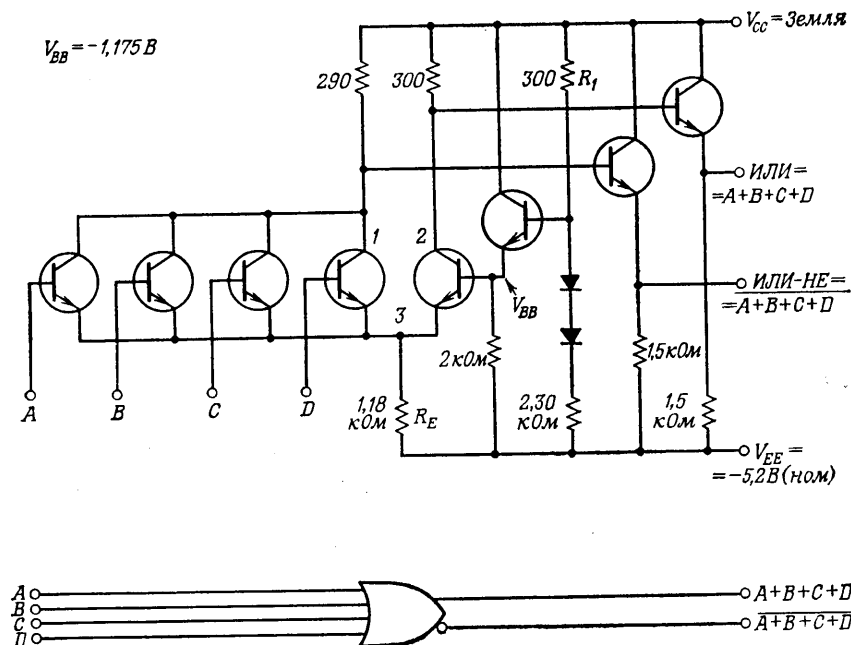
В конце концов слишком большое увеличение нагрузки настолько ухудшит характеристику, что работоспособность нарушится.

На фиг. 8.13 показан упрощенный вариант схемы транзисторно-транзисторной логики (ТТЛ). Если  $x_1$ ,  $x_2$ ,  $x_3$  или  $x_4$  имеют низкий (нулевой) уровень, то ток течет через  $R_1$  и транзистор  $T_1$ , поэтому  $T_2$  закрыт и на выходе образуется высокое (единичное) напряжение. Следовательно, логическое уравнение имеет вид

$$\bar{y} = x_1 x_2 x_3 x_4 \quad \text{или} \quad y = \overline{x_1 x_2 x_3 x_4}. \quad (8.7)$$

Отметим, что транзистор  $T_1$  многоэмиттерный. Приведенная схема отличается от промышленных, но с ее помощью легче понять процессы переключения и объяснить номенклатуру ТТЛ. Входы в схеме ТТЛ объединяются через транзисторы в отличие от РТЛ, где для этого служат резисторы. В еще одной разновидности схем — диодно-транзисторной логики (ДТЛ) — входы объединяются через диоды. По многим причинам, слишком специфичным, чтобы обсуждать их здесь, схема ТТЛ — самая быстродействующая и наиболее универсальная из трех перечисленных. Номинальное значение времени переключения обычной схемы ТТЛ равно 10—12 нс. Быстродействующие ТТЛ срабатывают за 6 нс, а ТТЛ с диодами Шоттки (ТТЛ ДШ) — за 4 нс, и их применение также экономически выгодно.

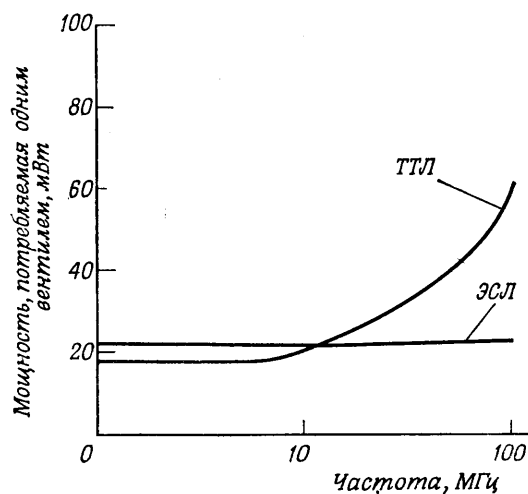
Схемы ТТЛ, ДТЛ и РТЛ относятся к семейству насыщенной логики. Для увеличения быстродействия необходимо некоторые



Фиг. 8.14. Схема ЭСЛ и ее обозначение (по Гаррету).

транзисторы удерживать в линейной части их передаточных характеристик. Это устраняет емкостные эффекты, обусловленные зарядом обедненного слоя в режиме насыщения. Ненасыщенный режим соответствует режиму класса А; ток в цепи течет все время, и поэтому рассеивается повышенная мощность. Интегральные схемы, обладающие таким свойством, относятся к эмиттерно-связанной логике (ЭСЛ), так как эмиттеры их входных транзисторов объединены (фиг. 8.14).

Схема на фиг. 8.14 состоит из четырех частей: входной цепи, дифференциального усилителя, цепи смещения и эмиттерного повторителя. Входная цепь имеет высокое входное сопротивление, а эмиттерный повторитель обеспечивает низкое выходное сопротивление; благодаря этому достигается большая нагрузочная способность по постоянному току. С помощью цепи смещения в дифференциальном усилителе устанавливается порог переключения тока. Два диода в цепи смещения компенсируют изменения уровня напряжения, вызванные вариациями температуры эмиттерных переходов транзисторов дифференциального усилителя. Кроме того, дифференциальный усилитель эффективно подавляет высшие гармоники переходного процесса. Таким образом, основное вни-



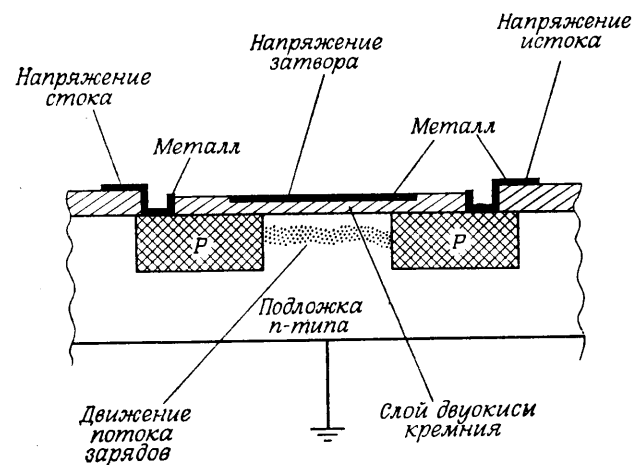
Фиг. 8.15. Зависимость потребляемой мощности от частоты переключения для насыщенной (ТТЛ) и ненасыщенной (ЭСЛ) логических схем.

вание при разработке схем ЭСЛ было направлено на обеспечение устойчивой работы на высоких частотах.

Выше уже отмечалось, что поскольку схема ЭСЛ работает на линейном участке характеристики транзистора, то она рассеивает большую мощность (по сравнению со схемами типа ТТЛ). Отметим еще одну особенность схемы. Поскольку ЭСЛ линейна, рассеиваемая мощность не зависит от скорости переключения. Что касается ТТЛ, то ток в ней течет только во время переключения. Поэтому, чем выше скорость переключения, тем большую часть составляет это время, так что рассеиваемая мощность возрастает при увеличении скорости переключения. Это отчетливо видно из кривых на фиг. 8.15, где показаны зависимости рассеиваемой мощности от скорости переключения для ТТЛ- и ЭСЛ-вентилей.

Хотя в настоящее время более популярны ТТЛ ИС, в процессе создания высокопроизводительных сложных систем стали постепенно выявляться преимущества ЭСЛ ИС. Можно ожидать, что в дальнейшем при цифровой обработке сигналов, где важную роль играет быстродействие, ЭСЛ будут успешно конкурировать с ТТЛ.

Для быстродействующих логических схем большое значение имеет выбор конструкции. При этом приходится учитывать временные задержки в соединительных проводах, искажения формы импульса, вызываемые эффектом длинных линий даже в срав-



Фиг. 8.16. Упрощенный эскиз МДП-транзистора.

нительно коротких проводниках на высоких частотах, увеличение перекрестных искажений между соседними проводами и температурные ограничения, связанные с большой рассеиваемой мощностью. Особое внимание следует обратить на системы распределения синхронизации и питающих напряжений, а также на тепловой режим (теплоотвод и грамотный монтаж, обеспечивающий равномерное распределение потока воздуха). Хорошие результаты дает применение многослойных печатных плат, которые позволяют создать надежные земляные шины, уменьшить трудности, связанные с разводкой, достичь высокой плотности монтажа и минимизировать задержки.

Перейдем к рассмотрению униполярных приборов. Они также построены на транзисторах, но основаны на другом принципе. В униполярных приборах используется полевой эффект, который состоит в том, что поток зарядов между двумя полупроводниками управляется изменением напряжения на металлическом «затворе». На фиг. 8.16 приведен эскиз прибора. Видно, что затвор изолирован от полупроводников слоем двуокиси кремния. Из-за наличия металлического затвора, слоя изолирующего окисла и (кремниевых) полупроводников эти приборы были названы МОП (металл—окисел—полупроводник)<sup>1</sup>).

На фиг. 8.16 показан один из вариантов построения такого прибора; на кремниевой подложке n-типа путем диффузии создаются две области p-типа, соответствующие истоку и стоку. За-

<sup>1</sup>) В отечественной литературе принято название МДП (металл — диэлектрик — полупроводник), которое и будет использоваться в дальнейшем. — Прим. перев.

	МДП	Биполярные ТТЛ
Память	1000-4000 бит 200-400 нс	64-1000 бит 30-100 нс
Логика	2000 вентиляей 1 мкс	50 вентиляей 2 нс

Фиг. 8.17. Современные возможности интегральных схем (по Раффилу).

тем наносится изолирующий слой окиси кремния, на котором располагается затвор—металлический управляющий элемент (обычно алюминиевый). К двум *p*-областям подводятся металлические контакты. Достаточно большое отрицательное напряжение, приложенное к затвору, вызовет инверсию в кремнии *n*-типа за счет притягивания дырок к поверхности затвора; в результате этого эффекта весь образец будет вести себя как простой полупроводник *p*-типа, и между двумя областями *p*-типа будет течь ток. Прибор имеет ярко выраженные ключевые свойства благодаря узкому порогу между состояниями наличия полного тока и отсутствия его.

В настоящее время серийно выпускаются уже несколько типов МДП-приборов, однако дальнейшие исследования обещают невиданные результаты. Поэтому на фиг. 8.17, где сделана попытка сопоставить преимущества и недостатки различных технологий, учитывается лишь уровень современного производства. Сравним униполярную (МДП) и биполярную технологии. Таблица соответствует уровню технологии 1973 г. Схемы МДП достигли намного большей степени интеграции, тогда как биполярные — много большего быстродействия. Несомненно, что МДП-схемы более предпочтительны для построения больших интегральных схем (БИС), что объясняется простотой технологического процесса, меньшими размерами и более низкой мощностью рассеивания МДП-транзисторов. Использование БИС в свою очередь сильно упрощает проблему монтажа.

Из таблицы на фиг. 8.17 видно, что при построении быстродействующих устройств цифровой обработки лучший компромисс между быстродействием и сложностью достигается для биполярных схем. Заманчиво, однако, сознавать, что существующий уровень МДП-технологии позволяет построить достаточно сложный цифровой фильтр целиком на одном кристалле! По-видимому, в настоящее время основным применением МДП-технологии в цифровой обработке сигналов является память на регистрах сдвига; для четырехфазных устройств достигнуто быстродействие 20 МГц.

Таблица 8.1

Сравнение основных типов логических интегральных схем (из статьи Гаррета)

Типы логических ИС	РТЛ	ДТЛ	ТТЛ	ЭСЛ	МДП	МДП ДТ
Выходное сопротивление	Килоомы	Килоомы	10—17 Ом	6—15 Ом	Килоомы	Килоомы
Нагрузочная способность	5	8	40	40—25	20	50
Рассеиваемая мощность на вентилятор, мВт	2,5—12	8—12	12—22	40—55	0,2—10	1 (на 1 МГц)
Помехоустойчивость	Средняя	Хорошая	Очень хорошая	Хорошая	Средняя	Очень хорошая
Уровень собственных помех	Низкий—средний	Средний	Высокий—средний	Низкий—средний	Средний	Низкий—средний
Задержка переключения вентиля, нс	12—25	30	6—12	1—4	300	70
Тактовая частота триггера, МГц	2,5—8	12—30	15—60	60—400	2	5
Универсальность	Хорошая	Средняя	Очень хорошая	Хорошая	Низкая, но улучшается	Низкая, но улучшается

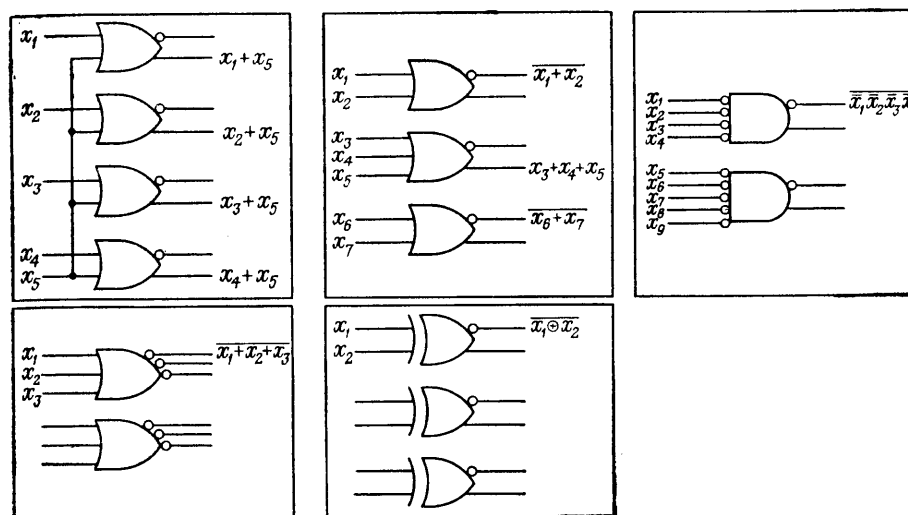


Как станет ясно в гл. 9, цифровой фильтр можно построить, используя в качестве памяти регистры сдвига.

В заключение приведем отредактированную табл. 8.4, взятую из статьи Гаррета, посвященной основным типам логических схем. Пользуясь ею, подведем итоги раздела. Во-первых, только схемы ТТЛ, ЭСЛ и МДП представляют в настоящее время интерес для цифровой обработки сигналов. Хотя схемы МДП ДТ (дополняющие МДП-транзисторы, см. последний столбец таблицы) не рассматривались, из таблицы следует, что они имеют определенные преимущества перед обычными схемами МДП. В настоящее время наиболее популярны схемы ТТЛ, которые обладают большей универсальностью, достаточным быстродействием и хорошо освоены разработчиками аппаратуры. С другой стороны, быстро завоевывают признание ЭСЛ-схемы. По-видимому, именно они лучше всего подходят для применений в области цифровой обработки сигналов. Несомненно, что в процессе ее развития будут широко использоваться логические схемы трех указанных типов.

### 8.5. Серийные логические ИС: вентили, коммутаторы и дешифраторы, триггеры, арифметические и запоминающие устройства

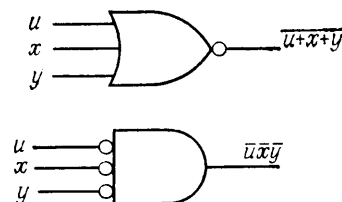
После создания очередной серии логических схем важно, чтобы она нашла широкое применение в цифровых устройствах. Даже когда новая серия намного превосходит существующие, это может произойти через несколько лет. По этой причине целесообразно начать выпуск с сокращенной номенклатуры, а затем, если схемы станут популярными и окажутся перспективными, постепенно расширять ее. К тому времени, когда серия станет общепризнанной, она должна быть самостоятельной, т. е. формировать все многообразие логических функций. Что касается биполярных устройств, то в настоящее время здесь имеется большой выбор различных вентилях и триггерах, существуют полезные с высоким уровнем интеграции арифметические устройства и элементы памяти, совместимые с существующей логикой. Для осуществления управления более высокого уровня целесообразно иметь также дешифраторы и коммутаторы в интегральном исполнении, примеры которых будут даны в этом разделе. Возможно, читателя удивит кажущийся произвол в выборе примеров. Следует не забывать специфики конструкции корпуса ИС. Чаще всего используются пластмассовый корпус с 16 выводами или плоский металлокерамический корпус (также с ограниченным числом выводов). Для схем с высоким уровнем интеграции, когда 16 выводов не хватает, могут использоваться корпуса с 24 выводами. Так как 3 вывода предназначены для питания схемы, то на сигналы остается (для корпуса с 16 выводами) 13 выводов. Легко убедиться в том, что для некото-



Фиг. 8.18. Примеры различных логических функций промышленных интегральных схем.

рых типичных логических схем, показанных на фиг. 8.18, суммарное число входов и выходов всегда равно 12 или 13.

К числу важных правил булевой алгебры относится следующее: используя элементы ИЛИ—НЕ и И, а также И—НЕ и ИЛИ, можно получить одни и те же логические функции. Например, на фиг. 8.19 сверху показан вентиль ИЛИ—НЕ с тремя входами, который может быть заменен вентилями И с тремя инверсными входами (на фиг. 8.19 внизу). В справочниках первый из этих вентилях (ИЛИ—НЕ) называется схемой *положительной логики*, а второй (И) — схемой *отрицательной логики*, так как в последнем случае логическая функция на самом деле выполняется над отрицаниями входных сигналов. Составим для этих схем таблицы истинности [см. (Т. И. 3)] и запишем соответствующие им булевы уравнения, которые иллюстрируют различные варианты построения вентилях:



Фиг. 8.19. Схема ИЛИ — НЕ с тремя входами и эквивалентная ей трехвходовая схема И — НЕ.

Положительная логика					Отрицательная логика						
		и и-НЕ		или		или-НЕ					
$u$	$x$	$y$	$xy$	$\overline{xy}$	$u+x+y$	$\overline{u+x+y}$	$\overline{xy}$	$\overline{xy}$	$\overline{u+x+y}$	$\overline{u+x+y}$	$\overline{u+x+y}$
0	0	0	0	1	0	1	1	0	1	0	0
0	0	1	0	1	1	0	0	1	1	1	0
0	1	0	0	1	1	0	0	1	1	1	0
0	1	1	0	1	1	0	0	1	1	1	0
1	0	0	0	1	1	0	0	1	1	1	0
1	0	1	0	1	1	0	0	1	1	1	0
1	1	0	0	1	1	0	0	1	1	1	0
1	1	1	1	0	1	0	0	1	0	1	1

(Т. И. 3)

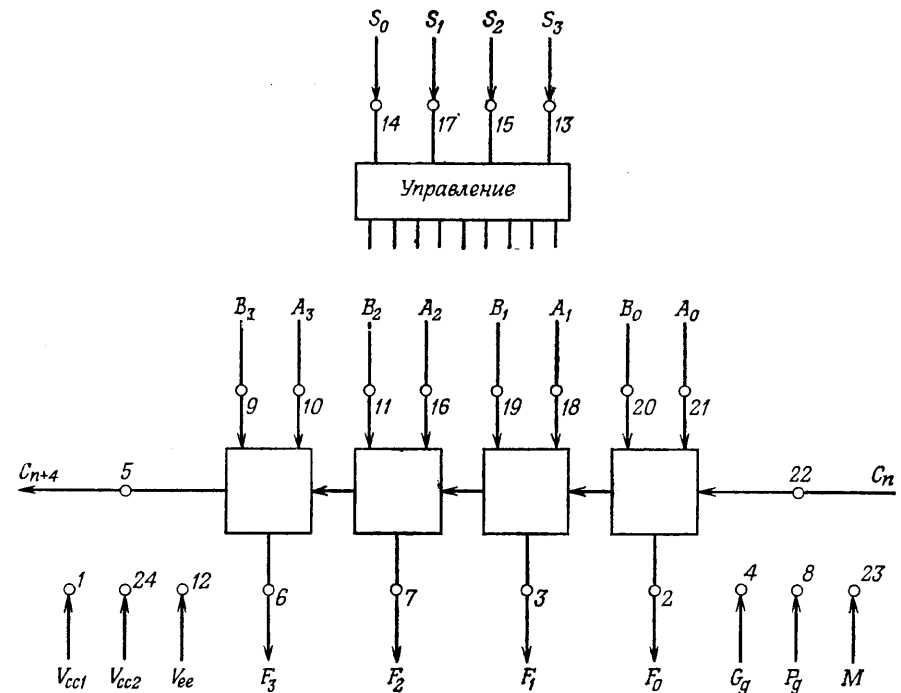
Из (Т. И. 3) имеем

$$\begin{aligned} \overline{xy} &= \overline{u+x+y}, \\ \overline{xy} &= \overline{u+x+y}, \\ xy &= \overline{\overline{u+x+y}}, \\ \overline{xy} &= \overline{u+x+y}. \end{aligned} \quad (8.8)$$

Таким образом, функция И от заданных переменных равна отрицанию функции ИЛИ от отрицаний этих переменных, и наоборот.

Существует множество вариантов реализации заданной системы с помощью основных логических вентилей, показанных на фиг. 8.18 и 8.19. Любую систему можно спроектировать, располагая сравнительно небольшой номенклатурой ИС. Однако лишь при наличии многофункциональных схем, схем с большой нагрузочной способностью и схем, реализующих специальные функции в одном корпусе (например, сумматора), можно рассчитывать на уменьшение размеров и увеличение быстродействия системы. В качестве упражнения попробуйте, используя логические цепи, приведенные на фиг. 8.18, составить логическую схему обнаружения нулевого состояния 16-разрядного параллельного регистра.

В качестве примера интегральной схемы с арифметическими функциями рассмотрим быстродействующее четырехразрядное арифметико-логическое устройство (АЛУ) MC10181, разработанное фирмой Motorola. Схема содержит 62 вентиля, т. е. несколько больше, чем говорилось выше, но она размещена в пластмассовом корпусе с 24 выводами, который занимает значительно большую площадь по сравнению с корпусом с 16 выводами. MC10181 ре-



Фиг. 8.20. Входные и выходные сигналы четырехразрядного АЛУ MC10181.

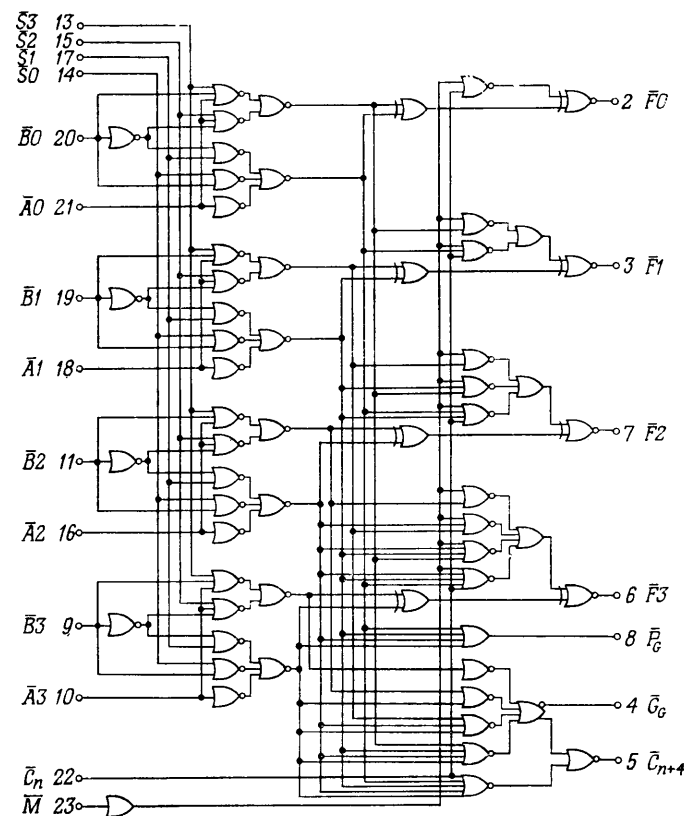
ализует функцию  $(F_0, F_1, F_2, F_3)$  от двух четырехразрядных переменных  $(A_0, A_1, A_2, A_3)$  и  $(B_0, B_1, B_2, B_3)$ . В дополнение к 12 перечисленным 3 вывода отведены под питание (фиг. 8.20). Кроме того, вводится четырехразрядный управляющий код  $(S_0, S_1, S_2, S_3)$ , определяющий вид функции; это означает, что данное АЛУ может формировать до 16 различных функций от двух входных четырехразрядных слов. Выводы  $G_g, P_g$  и  $C_{n+4}$  связаны с операцией переноса. При обычном сложении с последовательным переносом на вход  $C_n$  подается сигнал переноса в младший разряд, а с выхода  $C_{n+4}$  снимается перенос из старшего четвертого разряда. Выводы  $G_g$  и  $P_g$  предназначены для подключения дополнительной схемы формирования ускоренного переноса. Еще один управляющий вход  $M$  определяет, работает схема как арифметическое или логическое устройство. Список функций, реализуемых микросхемой MC10181, содержится в таблице на фиг. 8.21. Так как  $M$  относится к управляющим разрядам, то всего возможны 32 функции: 16 логических и 16 арифметических (они приведены во втором и третьем столбцах соответственно).

S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	Логическая функция	Арифметическая операция
0	0	0	0	$F = \bar{A}$	$F = A$ минус 1
0	0	0	1	$F = \overline{A+B}$	$F = A$ плюс $(A + \bar{B})$
0	0	1	0	$F = \bar{A} \cdot B$	$F = A$ плюс $(A + B)$
0	0	1	1	$F = 0$	$F = 2A$
0	1	0	0	$F = \overline{A \cdot B}$	$F = (A \cdot B)$ минус 1
0	1	0	1	$F = \bar{B}$	$F = (A \cdot B)$ плюс $(A + \bar{B})$
0	1	1	0	$F = A \oplus B$	$F = A$ плюс $B$
0	1	1	1	$F = A \cdot \bar{B}$	$F = A$ плюс $(A \cdot B)$
1	0	0	0	$F = \bar{A} + B$	$F = (A \cdot \bar{B})$ минус 1
1	0	0	1	$F = A \odot B$	$F = A$ минус $B$ минус 1
1	0	1	0	$F = \bar{B}$	$F = (A \cdot \bar{B})$ плюс $(A + B)$
1	0	1	1	$F = A \cdot B$	$F = (A \cdot \bar{B})$ плюс $A$
1	1	0	0	$F = 1$	$F =$ минус 1 (дополнительный код)
1	1	0	1	$F = A + \bar{B}$	$F = (A + \bar{B})$ плюс 0
1	1	1	0	$F = A + B$	$F = (A + B)$ плюс 0
1	1	1	1	$F = A$	$F = A$ плюс 0

Фиг. 8.21. Управляющие функции арифметического элемента MC10181 фирмы Motorola.

Логическое ИЛИ обозначено знаком  $+$ , а слова *плюс* и *минус* относятся к арифметическим операциям. Таким образом, видно, что ИС MC10181 может выполнять много булевых операций (по-разному) типа И, ИЛИ и неравнозначность, а также 16 вариантов обычного сложения. Поэтому при использовании MC10181 для построения ЦВМ не нужно предусматривать отдельные схемы, реализующие, например, операции И и арифметическое сложение, достаточно лишь подать на MC10181 соответствующий управляющий код. По-видимому, этот подход, заключающийся в использовании одной многофункциональной интегральной схемы, весьма полезен при создании универсальных ЦВМ. С другой стороны, применение ИС MC10181 в специализированных устройствах не целесообразно, поскольку многие из выполняемых ею функций могут не потребоваться.

Серийно выпускаемые ИС снабжаются, помимо функциональной схемы (фиг. 8.22), таблицами условий испытания ИС по постоянному току и задержек распространения. Таблицы задержек характеризуют возможности микросхемы и помогают оценить быст-

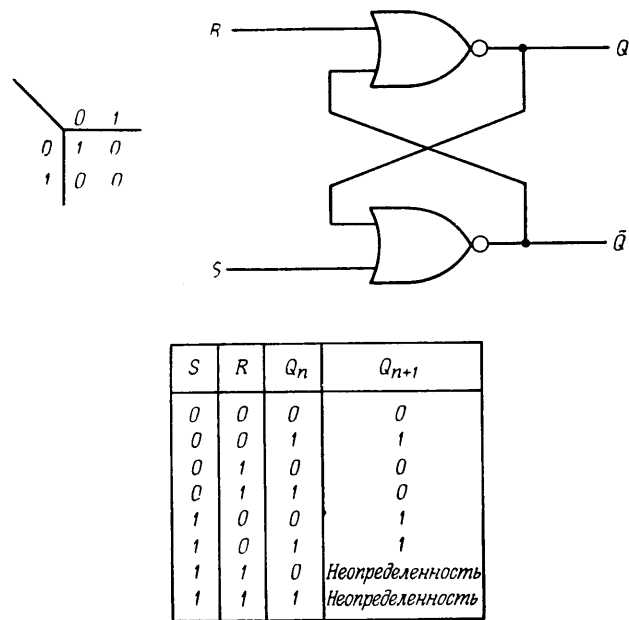


Фиг. 8.22. Функциональная схема четырехразрядного АЛУ MC10181 фирмы Motorola.

родействие устройств типа многоразрядных сумматоров или умножителей, собранных на ИС MC10181.

Перейдем к рассмотрению триггеров. Прежде чем приступить к описанию серийно выпускаемых триггеров, введем определение триггера, исходя из булевой алгебры. На фиг. 8.23 показаны два вентиля ИЛИ—НЕ с перекрестным соединением выходов со входами. Таблица слева описывает функцию ИЛИ—НЕ, а таблица внизу — работу всей схемы<sup>1</sup>). При подаче логической единицы на вход  $R$  (Reset — установка нуля) на выходе  $Q$  устанавливается нуль независимо от первоначального состояния схемы, а при подаче единицы на вход  $S$  (Set — установка единицы) устанавливается  $Q = 1$ . При подаче  $R = S = 0$  триггер сохраняет перво-

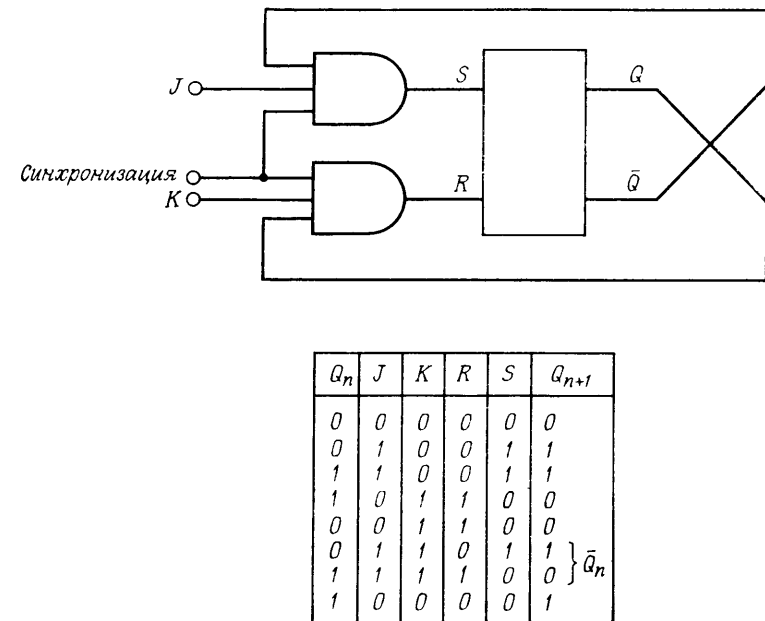
<sup>1</sup>) Здесь и ниже  $Q_n$  обозначает состояние выхода  $Q$  до подачи сигналов на входы схемы, а  $Q_{n+1}$  — после подачи сигналов. — Прим. ред.



Фиг. 8.23. Схема соединения двух вентилей ИЛИ — НЕ для получения триггера.

начальное состояние; таким образом, его можно рассматривать как одноразрядный элемент памяти. Одновременная подача на входы  $R$  и  $S$  единицы запрещена, так как логический результат этой операции не определен.

Важным применением триггеров является использование их в двоичных счетчиках. При этом триггеры постоянно должны быть готовы к *переключению*, т. е. к изменению своего состояния при каждом поступлении сигнала на синхронизирующий вход. Триггер, обладающий таким свойством, вместе с соответствующей таблицей состояний представлен на фиг. 8.24. Серийно выпускаемые триггеры такого типа называются  $JK$ -триггерами. Два входных вентиля — трехходовые схемы И, два других — аналогичны описанным выше (см. фиг. 8.23). Учтем, что  $R = 1$  и  $S = 1$  — сигналы установки выхода  $Q$  в нулевое и единичное состояния соответственно. Тогда если при подаче, например, сигнала  $J = 1$  окажется, что  $Q = 1$ , то на выходе схемы И образуется  $S = 1$  (третий вход схем И — синхронизирующий), что приведет к переключению триггера в состояние  $Q = 1$ . С другой стороны, если триггер уже находился в этом состоянии, оба вентиля будут закрыты, и значение  $Q = 1$  будет сохраняться. Таким образом,



Фиг. 8.24. Работа  $JK$ -триггера.

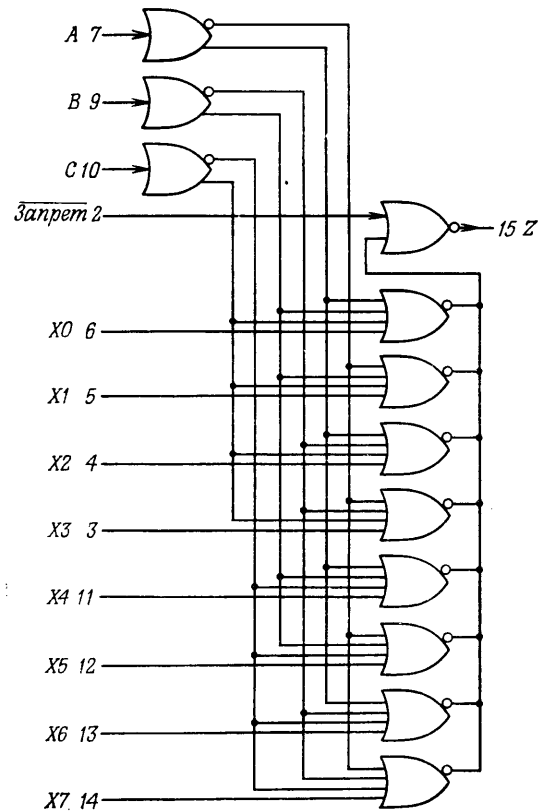
входы  $J$  и  $K$  действуют как установочные, подобно входам  $R$  и  $S$  в предыдущей схеме. Однако при подаче единичных сигналов на входы  $J$  и  $K$  одновременно триггер перебросится из одного состояния в другое, как это следует из двух нижних строк таблицы. Отметим, что в отличие от входных уровней  $R$ ,  $S$ ,  $J$  и  $K$  сигнал синхронизации может быть относительно коротким. Поэтому при необходимости  $JK$ -триггер можно использовать для синхронизации поступающих данных.

Существуют и другие типы триггеров, например триггер-защелка, в который при подаче сигнала синхронизации заносится внешняя информация. В одном корпусе могут размещаться один, два, четыре и даже восемь триггеров. Представляет интерес способ компоновки интегральной схемы из нескольких триггеров, заключающийся в том, что предусматривается параллельная запись и последовательное считывание (или наоборот). С расширением области применения ИС возникла необходимость разработки дополнительных схем, реализующих в одном корпусе некоторые достаточно сложные логические функции. К их числу относится представленная на фиг. 8.25 схема восьмиканального коммутатора данных МС10164, с помощью которого к выходу подключается любой из восьми входных сигналов  $X_0 \div X_7$  в соответствии с трех-

Таблица истинности

Разре- шение	Адресные входы			Z
	C	B	A	
0	0	0	0	X0
0	0	0	1	X1
0	0	1	0	X2
0	0	1	1	X3
0	1	0	0	X4
0	1	0	1	X5
0	1	1	0	X6
0	1	1	1	X7
1	φ	φ	φ	0

φ-любое



Фиг. 8.25. Восьмиканальный коммутатор фирмы Motorola.  
 $t_{\text{рас}} = 3,5 \text{ нс}$  (ном) (с входов на выход).

разрядным кодом *ABC*. Особенно целесообразно применение таких схем в устройствах многоканальной обработки, где придется поочередно подключать к сумматору или к какому-либо

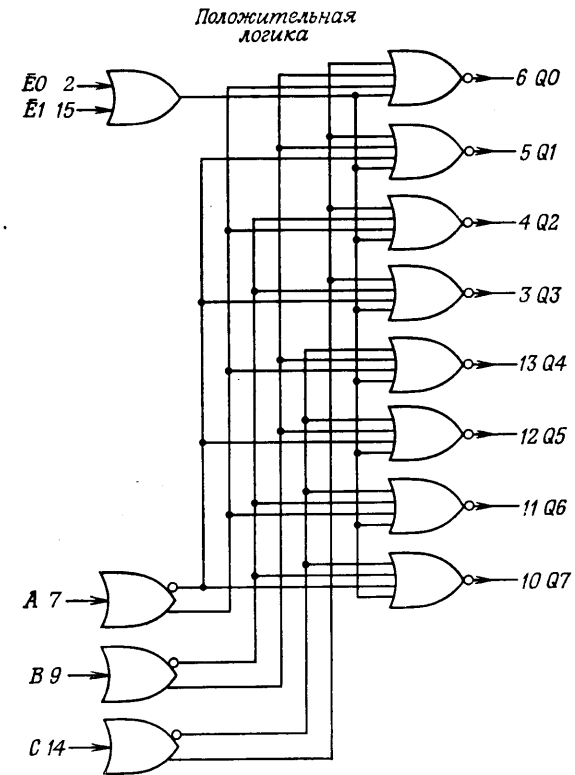


Таблица истинности

Входы			Выходы									
E0	E1	C	B	A	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	0	0
0	0	0	1	1	0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0	0	1	0	0	0
0	0	1	0	1	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	0	1	0
0	0	1	1	1	0	0	0	0	0	0	0	1
1	φ	φ	φ	φ	0	0	0	0	0	0	0	0
φ	1	φ	φ	φ	0	0	0	0	0	0	0	0

φ-любое

Фиг. 8.26. Трехразрядный дешифратор адреса.  
 $t_{\text{рас}} = 4,0 \text{ нс}$  (ном).

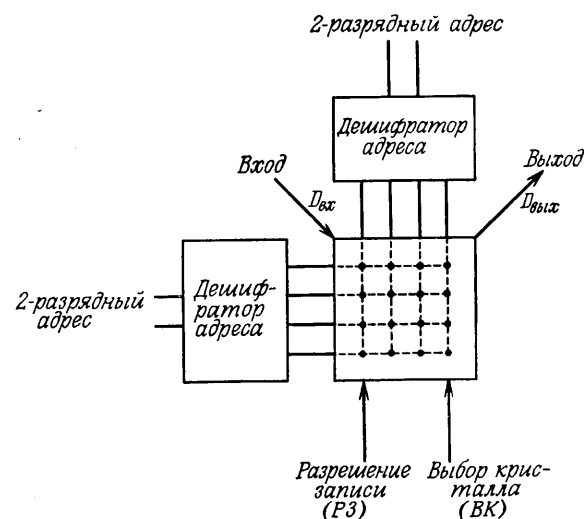
другому арифметическому устройству один или два из многих регистров. В качестве упражнения предлагаем читателю построить на базе двух ИС MC10164 коммутатор на 16 каналов.

Другой весьма полезный модуль МС10162, представляющий собой дешифратор трехразрядного кода АВС, показан на фиг. 8.26. В зависимости от входного кода только на каком-либо одном из восьми выходов возникает логическая единица. Как будет видно ниже, схема применяется для адресации запоминающего устройства с произвольной выборкой.

Запоминающие устройства являются одними из самых дорогих элементов цифровой аппаратуры, поэтому они имеют для разработчиков особенно важное значение. Опыт показывает, что устройства цифровой обработки сигналов не являются в этом смысле исключением. В настоящее время происходят серьезные изменения в технологии ЗУ, которые приводят к постепенной замене в новых ЦВМ оперативной памяти на ферритовых сердечниках интегральной полупроводниковой памятью. Топология ИС памяти отличается высокой геометрической регулярностью, что позволяет достичь большего, чем у других логических элементов, уровня интеграции. Уже сейчас выпускаются быстродействующие ЗУ с произвольной выборкой на биполярных транзисторах емкостью 256 бит в корпусе, а в ближайшее время ожидается появление ЗУ емкостью 1024 бит. В то же время емкость менее быстродействующих МДП ЗУ на регистрах сдвига скоро достигнет 4096 бит в корпусе.

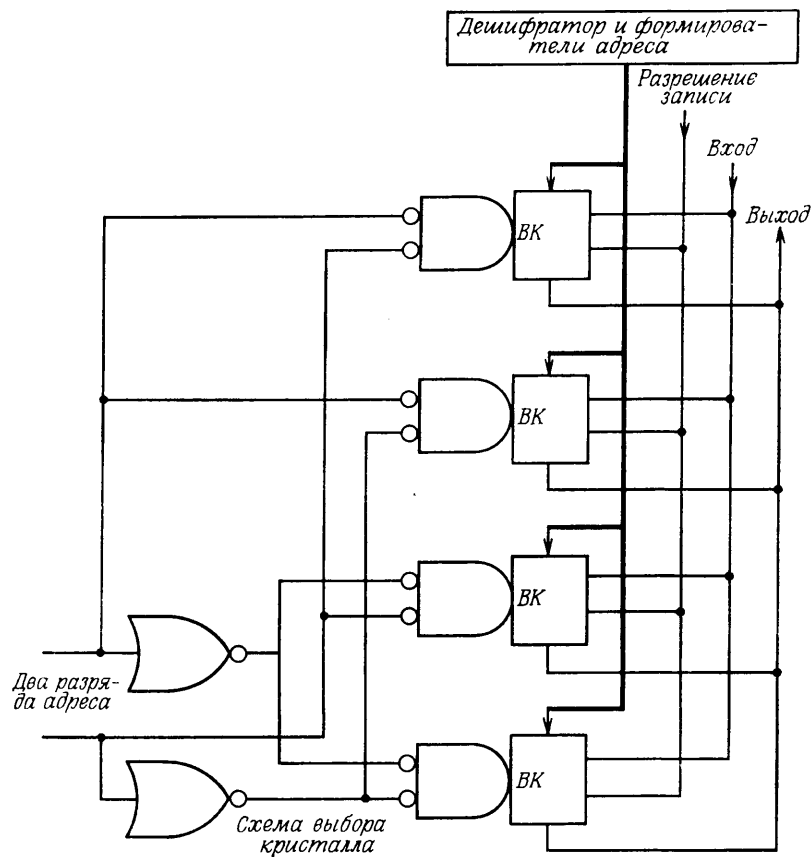
Запоминающие устройства подразделяются на *статические* и *динамические*. Информация в статических ЗУ сохраняется все время, пока есть питание, в динамических ЗУ ее требуется периодически обновлять. Биполярные ЗУ относятся к статическим, память на МДП-транзисторах бывает как статической, так и динамической. Кроме того, запоминающие устройства могут быть классифицированы по *возможности записи* (т. е. по возможности изменения информации ЗУ). Различают постоянное запоминающее устройство (ПЗУ), из которого информацию можно только считывать, но нельзя записывать; программируемое ПЗУ (ППЗУ), в которое запись производится заказчиком только один раз; полупостоянное ЗУ, у которого скорость записи значительно ниже скорости считывания, и обычное ЗУ с примерно одинаковой продолжительностью записи и считывания. Организация доступа к ячейкам памяти может быть *случайной* или *последовательной*. В первом случае следующие друг за другом адреса могут быть произвольными (поэтому память со случайным доступом правильнее называть памятью с произвольным доступом). ЗУ с последовательным доступом по своей организации аналогичны дискам, барабанам или линиям задержки, у которых соседние ячейки памяти опрашиваются последовательно одна за другой.

Чтобы понять работу ЗУ с произвольной выборкой, рассмотрим схему на 16 разрядов, приведенную на фиг. 8.27. Память организована в виде матрицы  $4 \times 4$  (каждый разряд представляет собой



Фиг. 8.27. Блок-схема элемента памяти на 16 бит.

триггер на транзисторах). Чтобы выбрать любой из 16 разрядов, требуется четырехразрядный адрес. Адресное слово разделено на два полуслова: одно для строк, а другое для столбцов. Для дешифровки полуслов используется схема, подобная приведенной на фиг. 8.26, в результате чего возбуждается одна из четырех выходных шин каждого дешифратора. Выбираемый разряд находится в месте пересечения возбужденных шин по строке и столбцу. Кроме четырех адресных разрядов, имеются входная и выходная шины ( $D_{вх}$ ,  $D_{вых}$ ), по которым в зависимости от режима, определяемого сигналом на шине разрешения записи (PЗ), осуществляется либо запись новых данных, либо чтение информации из выбранного разряда. Наконец, имеется шина управления выбором кристалла (ВК), позволяющая объединять кристаллы для построения памяти большей емкости. В рассматриваемом примере кристалл предназначен для запоминания 16 одноразрядных слов. Для увеличения длины слова до  $n$  разрядов требуется  $n$  таких кристаллов, управляемых одними и теми же шинами адреса и сигналами PЗ и ВК. Увеличение емкости памяти требует новой схемы, пример которой дан на фиг. 8.28. Она представляет собой ЗУ на 64 одноразрядных слова. Шины  $D_{вх}$ ,  $D_{вых}$  и PЗ всех кристаллов соединены между собой. Теперь разряды адреса разбиваются на три группы, соответствующие шинам строк, столбцов и ВК каждого из кристаллов (см. фиг. 8.28). Таким образом осуществляется выбор единственного разряда одного из четырех кристаллов. Добавление по одному кристаллу по горизонтали приводит к увели-

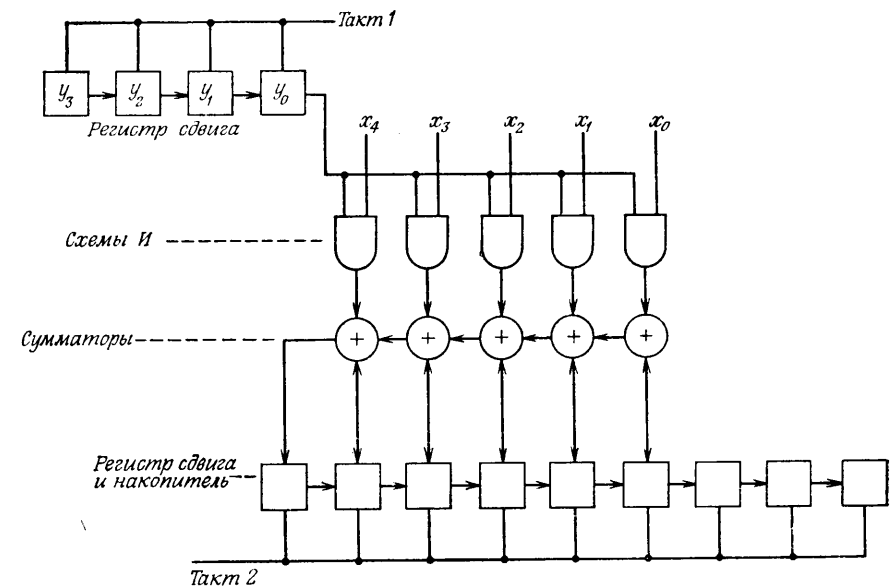


Фиг. 8.28. Память емкостью 64 одноразрядных слова на основе 16-разрядных кристаллов памяти.

чению длины слова на 1 разряд, а по вертикали — к увеличению количества слов (в рассматриваемом примере до 16 слов).

### 8.6. Умножители

Настоящий раздел посвящен умножителям, играющим особенно важную роль в устройствах цифровой обработки сигналов. С точки зрения принципа действия умножители можно разделить на многотактные и матричные. В обоих случаях произведение является результатом последовательных сложений с той лишь разницей, что достаточный параллелизм матричных умножителей позволяет обойтись без запоминания промежуточных результатов. Сначала кратко рассмотрим применяемые в умножителях системы счисле-



Фиг. 8.29. Блок-схема умножения на основе сложения и сдвига.

ния, а затем опишем различные алгоритмы многотактного и матричного умножения. Более полный анализ систем счисления содержится в пятой главе.

При анализе систем счисления было описано представление чисел в прямом (с использованием знака и величины), обратном и дополнительном кодах. Сравнение двух последних систем счисления показало, что дополнительный код больше подходит для выполнения высокоскоростных операций, поэтому в дальнейшем будут рассматриваться только прямой и дополнительный коды. Перемножение чисел в прямом коде несколько проще, чем в дополнительном, поэтому сначала будут рассмотрены способы умножения положительных чисел. Однако больше внимания будет уделено методам умножения чисел в дополнительном коде, так как сложение двух чисел в прямом коде менее удобно, чем в дополнительном.

На фиг. 8.29 изображена схема простого умножителя пятиразрядного числа на четырехразрядное, использующего сложения и сдвига. Если соответствующее значение  $y$  из правого столбца равно 1, то в этом устройстве последовательно накапливаются следующие слагаемые:

$$\begin{array}{rcccccc}
 & x_4 & x_3 & x_2 & x_1 & x_0 & y_0 \\
 & & x_4 & x_3 & x_2 & x_1 & x_0 & y_1 \\
 & & & x_4 & x_3 & x_2 & x_1 & x_0 & y_2 \\
 x_4 & x_3 & x_2 & x_1 & x_0 & & & & y_3
 \end{array}$$

При  $y = 0$  на сумматор с выхода вентиля поступают нули. После логического умножения и накопления очередной строки сумма сдвигается на один разряд вправо, чтобы обеспечить в соответствии с требованием алгоритма добавление следующей строки в более старшие разряды. По команде тактового импульса 1 (см. наверху) разряды  $y$  последовательно поступают на вентили И и обеспечивают подачу на вход сумматора либо числа  $x$  либо нуля. После окончания цикла сложения по команде тактового импульса 2 результат суммирования заносится в триггерный регистр накопителя и затем после завершения процесса установления сдвигается на один разряд вправо. Этот умножитель достаточно простой, но и медленный. Поскольку в сумматоре отсутствует схема ускорения переноса, операция накопления завершается лишь после того, как сигнал переноса пройдет через весь сумматор. Наибольшее значение времени установления для  $n$ -разрядного сумматора примерно равно  $(n - 1) \tau_{\Pi} + \tau_c + \tau_p$ , где  $\tau_{\Pi}$  и  $\tau_c$  — соответственно время распространения переноса и время образования суммы в одноразрядном сумматоре,  $\tau_p$  — время установления регистра сдвига  $y$ . Если  $n$ -разрядное число умножается на  $m$ -разрядное, то общее время умножения будет приблизительно равно  $m(n - 1) \tau_{\Pi} + m\tau_c + \tau_p$ .

Приведенная на фиг. 8.29 схема отражает тот важный с аппаратной точки зрения факт, что в умножителях, основанных на сложении и сдвиге, разряды  $y$  хранятся в последовательном сдвиговом регистре. В гл. 9 будет показано, что представление данных в виде последовательного потока положительно влияет на структуру цифрового фильтра и приводит к снижению стоимости его памяти.

Рассмотрим теперь, каким образом следует модифицировать этот умножитель для работы с числами разного знака, заданными в дополнительном коде. Напомним сначала формулы для представления и перемножения чисел в дополнительном коде. Значение числа  $X$ , представленного в дополнительном коде, равно

$$X = -2^n x_n + \sum_{m=0}^{n-1} 2^m x_m = -2^n x_n + 2^{n-1} x_{n-1} + \dots + 2^0 x_0, \quad (8.9)$$

где  $x_m$  — разряды дополнительного кода, а  $x_n$  — знаковый разряд, причем сумма понимается в арифметическом, а не логическом смысле. В качестве упражнения предлагаем читателю, используя уравнение (8.9), составить алгоритм перевода числа в дополнительном коде из положительного в отрицательное.

Из формулы (8.9) можно получить арифметическое выражение для произведения  $XU$  двух чисел в дополнительном коде, когда

$X$  и  $U$  соответственно  $(n + 1)$ - и  $(m + 1)$ -разрядные числа:

$$XU = 2^{n+m} x_n y_m - 2^n x_n \sum_{j=0}^{m-1} 2^j y_j - 2^m y_m \sum_{k=0}^{n-1} 2^k x_k + \sum_{j=0}^{m-1} \sum_{k=0}^{n-1} 2^j 2^k y_j x_k. \quad (8.10)$$

Первый и четвертый члены в правой части (8.10) — положительные числа, второй и третий — отрицательные. Четвертый член равен окончательному результату в случае перемножения положительных чисел, который получается при последовательных сложениях в схеме на фиг. 8.29. Из равенства (8.10) видно, что если  $x_n$  или  $y_m$  равны единице, то суммы должны как складываться, так и вычитаться. Модификация логической структуры умножителя, основанного на сложении и сдвиге, учитывающая знаковые разряды, известна под названием *алгоритма Бута*. Чтобы понять этот алгоритм, представим число  $U$  в следующем виде:

$$U = (y_{m-1} - y_m) 2^m + \sum_{j=0}^{m-1} 2^j (y_{j-1} - y_j), \quad y_{-1} = 0. \quad (8.11)$$

Итак, при последовательном умножении множимого ( $X$ ) на разряды множителя ( $U$ ) операции будут производиться над последовательными строками, составленными из разрядов  $X$ , по следующим правилам:

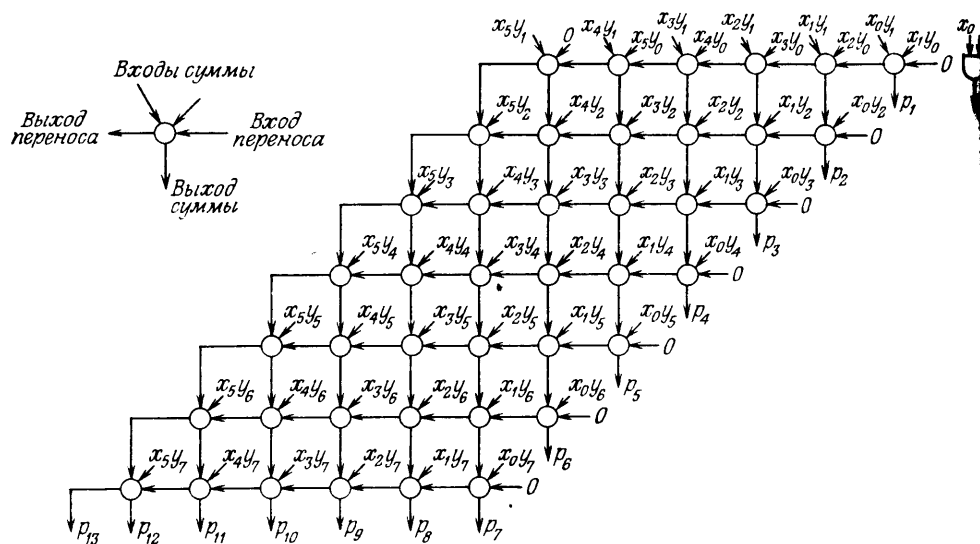
1. Если  $y_{j-1} = y_j$ , то накапливается не множимое, а нулевая строка.
2. Если  $y_{j-1} = 1$ , а  $y_j = 0$ , то множимое добавляется в накопитель.
3. Если  $y_{j-1} = 0$ , а  $y_j = 1$ , то множимое *вычитается* из накопителя.

В качестве примера читателю предлагается, используя алгоритм Бута, умножить  $X = -6$  на  $U = -27$  и, кроме того, модифицировать схему на фиг. 8.29 для реализации этого алгоритма.

Скорость выполнения алгоритма Бута можно увеличить в два раза, сдвигая множитель сразу на два разряда и анализируя одновременно три соседних разряда, чтобы определить характер накопления множимого; этот вариант также рекомендуется для самостоятельной проработки.

Самые быстрые умножители состоят из двумерной матрицы одноразрядных сумматоров и называются матричными умножителями. В отличие от многотактного умножителя (фиг. 8.29) матричный умножитель представляет собой законченную логическую схему без элементов памяти, поэтому результат умножения образуется после подачи сомножителей за время, равное времени установления схемы. Существует много различных вариантов построения матрицы, причем они могут быть достаточно хорошо классифи-



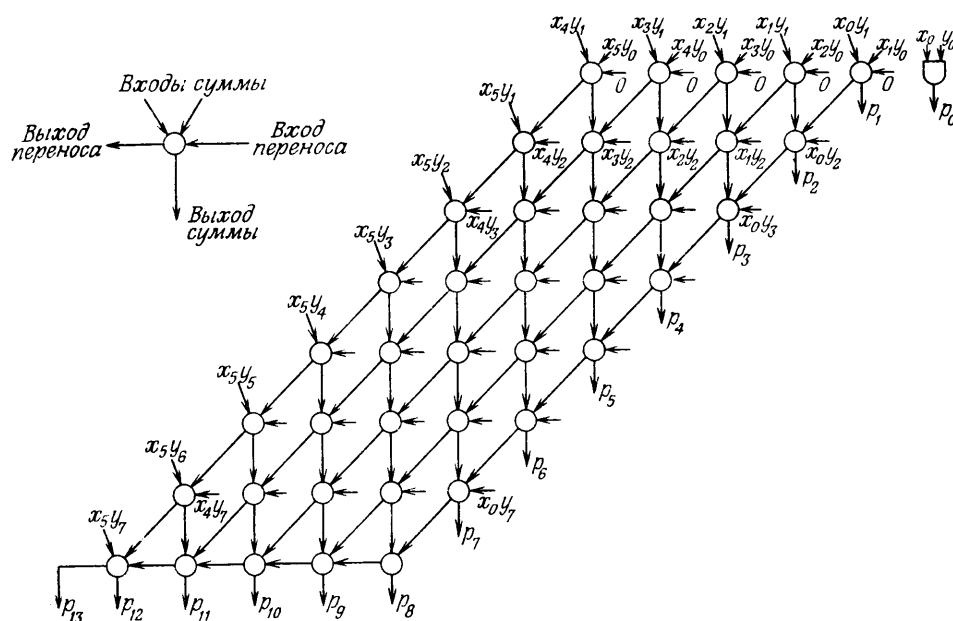


Фиг. 8.30. Матричный умножитель положительных чисел  $(x_5x_4x_3x_2x_1x_0)$  и  $(y_7y_6y_5y_4y_3y_2y_1y_0)$ .

цированы по методам соединения сумматоров и обработки отрицательных чисел. На фиг. 8.30 показан пример не слишком быстродействующего матричного умножителя, оперирующего с положительными числами. Каждый кружок обозначает одноразрядный сумматор. Видно, что каждая строка сумматоров представляет собой многоразрядный сумматор с распространением переноса. Она формирует частичную сумму и передает ее последующей строке. Чтобы оценить время установления в таком умножителе, учтем, что разряд результата  $p_7$  образуется спустя время, равное  $m$  задержкам суммирования и  $n$  задержкам переноса, а разряд  $p_{13}$  — на  $n$  задержек переноса позже. Таким образом, общее время установления  $\tau_0$  равно

$$\tau_0 = 2n\tau_{\Sigma} + m\tau_c. \quad (8.12)$$

Эта величина значительно меньше, чем для схемы на фиг. 8.29. Если обеспечить распространение сигналов переноса по диагонали (фиг. 8.31), то суммарную задержку можно уменьшить. В этом случае безразлично, на какую строку постукает перенос от предыдущего столбца. Схемы, показанные на фиг. 8.30 и 8.31, дают одно и то же произведение, однако последняя имеет преимущество, поскольку сигналы переноса и сумм, распространяющиеся от сумматоров в младших разрядах, одновременно поступают на сумматоры в более старших разрядах. Это так называемая схема с сохранением

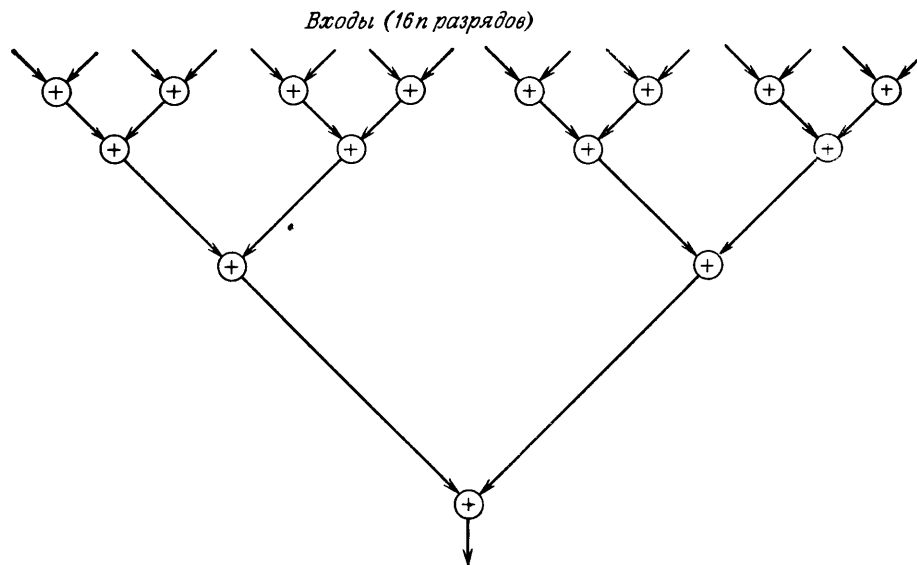


Фиг. 8.31. Структура матричного умножителя, более быстродействующего по сравнению со схемой, изображенной на фиг. 8.30.

переноса, которая применяется не только для матричных умножителей. Например, сигнал на выходе  $p_8$  в столбце 8 появится спустя время, равное пяти задержкам суммирования или восьми задержкам переноса, в зависимости от того, какой отрезок времени длиннее. Если они равны, то временем суммирования можно пренебречь. Поэтому для последней строки сумматоров нужно учитывать суммарное время переноса. Таким образом, если  $8\tau_{\Sigma} \leq 5\tau_c$ , то в наихудших условиях находится выход  $p_{13}$ , задержка образования которого равна  $12\tau_{\Sigma}$ . В противном случае время умножения будет равно  $5\tau_c + 5\tau_{\Sigma}$ . Итак,

$$\tau_0 = \begin{cases} [(m-1) + (n-1)]\tau_{\Sigma} & \text{при } m\tau_{\Sigma} > (n-1)\tau_c, \\ (n-1)(\tau_{\Sigma} + \tau_c) & \text{при } m\tau_{\Sigma} < (n-1)\tau_c. \end{cases} \quad (8.13)$$

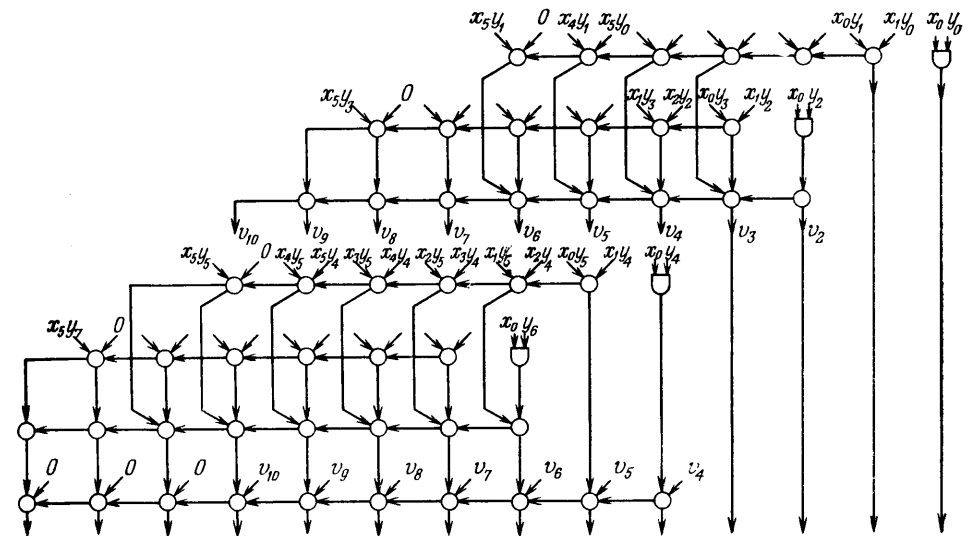
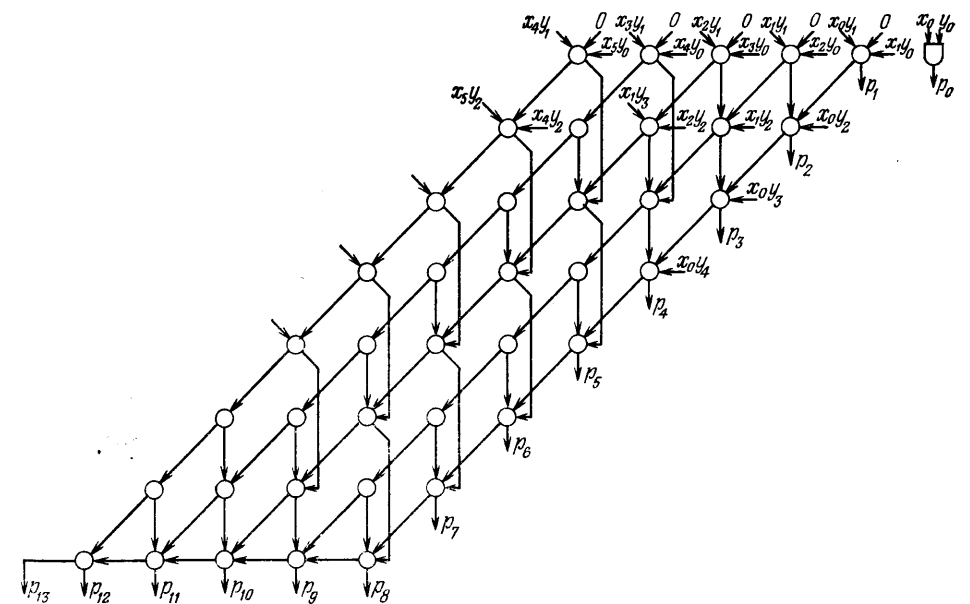
При использовании одноразрядных сумматоров, у которых  $\tau_c$  значительно больше, чем  $\tau_{\Sigma}$ , нужна схема умножителя с минимальным числом последовательно выполняемых сложений. Идея построения такого умножителя с матрицей сумматоров, организованной в виде «дерева», иллюстрируется на фиг. 8.32. Он предназначен для умножения  $n$ -разрядного множимого на 16-разрядный множитель. Каждый из кружков обозначает  $n$ -разрядный сумматор. При такой организации время суммирования будет лишь учетве-



Фиг. 8.32. Древоидная структура матричного умножителя.

ряться. Общее количество сумматоров остается прежним, но больше строк сумматоров работает теперь параллельно. Блок-схема умножителя такого типа с разрядностью, равной, как и раньше,  $(6 \times 8)$ , в котором сигналы переноса распространяются вдоль строк, показана на фиг. 8.33. Суммирование в каждой строке сумматоров происходит за время формирования переноса, равное  $(n - 1) \tau_{\text{п}}$ . Поскольку таких уровней суммирования четыре, общее время умножения равно  $4(n - 1) \tau_{\text{п}}$ . Таким образом, если  $\tau_{\text{с}}$  не очень велико, эффективность данной схемы ниже, чем приведенной на фиг. 8.31.

Возможно, что наибольшего быстродействия можно достичь, сочетая идеи древоидного соединения и диагонального распространения переносов. Соответствующая структурная схема приведена на фиг. 8.34. Так как эта структура обладает, по-видимому, наибольшим быстродействием из рассмотренных выше матричных схем, возникает вопрос о целесообразности рассмотрения других матричных структур. Ответ на этот вопрос может быть получен только в процессе проектирования системы на основе конкретных интегральных схем. Обычно ИС разрабатываются и выпускаются в расчете на самые разнообразные применения. Поэтому при создании матричного умножителя разработчик должен выбрать наиболее подходящие ИС из числа имеющихся. В качестве примера рассмотрим выпущенную в 1971 г. интересную ИС МС10181, которая пред-

Фиг. 8.33. Древоидный матричный умножитель размером  $(6 \times 8)$ .

Фиг. 8.34. Матричный умножитель с диагональным распространением переноса и древоидной организацией суммирования.

ставляет собой четырехразрядный сумматор с ускоренным переносом. На ее основе можно построить диагональную (как на фиг. 8.31) или горизонтальную (как на фиг. 8.30 и 8.33) структуры. Хотя схема, приведенная на фиг. 8.31, и оказывается более быстрой, для формирования частичных произведений  $x_j \cdot y_k$  требуются дополнительные корпуса. В схеме с горизонтальным распространением переносов эти корпуса не нужны, поскольку каждый управляющий разряд можно логически умножить сразу на четыре входных разряда. Итак, выбор «наилучшей» схемы определяется характеристиками используемых микросхем и компромиссом между быстродействием и стоимостью умножителя.

Рассмотрим умножение чисел, представленных в дополнительном коде, пользуясь основной формулой (8.9). Сущность этой формулы состоит в следующем: каждый разряд, за исключением знакового, принимает арифметические значения 0 или 1, а знаковый — значения 0 или -1. При таком подходе, выполняя суммирование согласно формуле (8.9), всегда будем получать правильный результат.

Одноразрядный сумматор может быть описан подобно тому, как это делалось ранее, с помощью логических уравнений или таблицы истинности. Если каждый из трех входных разрядов принимает (арифметические) значения 0 или 1, то двухразрядный выход может иметь значения 0, 1, 2 или 3. При реализации этой функции значения 0 или 2 приписываются разряду переноса, а значения 0 или 1 — разряду суммы; оба эти разряда описывают выход сумматора. Таблица истинности (Т. И. 4) дает состояния выходных разрядов сумматора — суммы  $D$  (младший разряд) и переноса  $E$  — в зависимости от состояния входов  $A$ ,  $B$  и  $C$ . Ясно, что совершенно необязательно обозначать два возможных состояния выходных разрядов либо нулем, либо единицей; для этого можно использовать + или -, 0 или 2, 0 или 100 и т. д. Система обозначений, интуитивно принятая в таблице истинности (Т. И. 4), фактически соответствует двоичному представлению выхода сумматора:

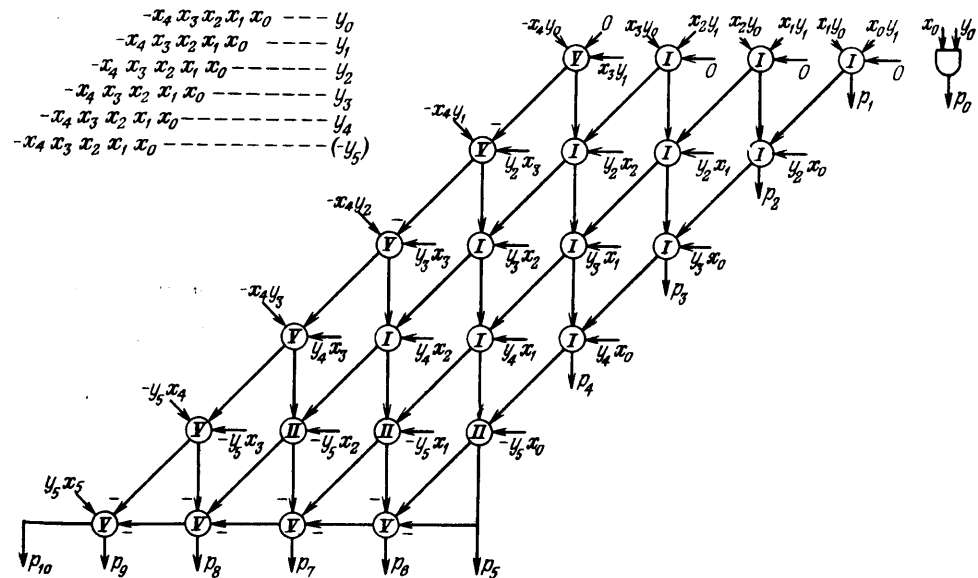
A	B	C	D	E	Арифметическая сумма
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	0	2	2
1	0	0	1	0	1
1	0	1	0	2	2
1	1	0	0	2	2
1	1	1	1	2	3

I (Т.И.4)  
 [(IV) получается из (I) изменением знака]

$A$ ,  $B$ ,  $C$  — входы;  $D$  — сумма (булева);  $E$  — перенос (арифметическая сумма минус булева сумма). Эту систему обозначений можно обобщить на случай арифметического суммирования и положительных и отрицательных чисел. Рассмотрим, например, трехвходовую схему, у которой один из входных сигналов принимает значения -1 или 0, а два других — значения 0 или 1. В этом случае сумма по-прежнему принимает четыре возможных значения -1, 0, +1, +2 и может быть представлена двухразрядным числом с младшим разрядом, равным 0 или -1, и старшим, равным 0 или 2.

Снова из таблицы истинности такой логической схемы (Т. И. 5) следует, что результат сложения в сумматоре, рассматриваемый как арифметическая сумма двух выходов, соответствует также арифметической сумме всех трех входов. Аналогично можно рассмотреть еще два случая с таблицами истинности (Т. И. 6) и (Т. И. 7).

Воспользовавшись введенной системой обозначений и располагая интегральными схемами, способными суммировать отрицательные числа, можно построить матричный умножитель чисел в дополнительном коде, основанный на последовательных сложениях. Пример такого умножителя, в котором использованы три типа интегральных схем, — обычные сумматоры, описываемые таблицей истинности (Т. И. 4), а также сумматоры, соответствующие (Т. И. 6) и (Т. И. 7), — показан на фиг. 8.35.



Фиг. 8.35. Матричный умножитель чисел в дополнительном коде размером (6 × 5).

A	B	C	D	E	Арифметическая сумма	
0	0	0	0	0	0	
0	0	1	-1	2	1	
0	1	0	-1	2	1	
0	1	1	0	2	2	(II) (Т.И.5)
-1	0	0	-1	0	-1	
-1	0	1	0	0	0	
-1	1	0	0	0	0	
-1	1	1	-1	2	1	

A	B	C	D	E	Арифметическая сумма	
0	0	0	0	0	0	
0	0	1	1	0	1	
0	-1	0	1	-2	-1	
0	-1	1	0	0	0	(III) (Т.И.6)
-1	0	0	1	-2	-1	
-1	0	1	0	0	0	
-1	-1	0	0	-2	-2	
-1	-1	1	1	-2	-1	

## Сравнение различных

Структура	Тип основной интегральной схемы
Быстрая трапецеидальная (фиг. 8.35)	Специальный двухразрядный сумматор
Древовидная	Серийный четырехразрядный сумматор
»	То же
»	Серийная матрица (2×4)
Многотактная с одновременным сдвигом на два разряда	Специальный двухразрядный сумматор
То же	Серийный двухразрядный сумматор
» »	То же

Авторы благодарны П. Бленкеншипу и А. Хантуну за разрешение использовать эту

A	B	C	D	E	Арифметическая сумма	
0	0	0	0	0	0	
0	0	-1	1	-2	-1	
0	-1	0	1	-2	-1	
0	-1	-1	0	-2	-2	(V) (Т.И.7)
1	0	0	1	0	1	
1	0	-1	0	0	0	
1	-1	0	0	0	0	
1	-1	-1	1	-2	-1	

В табл. 8.2 даны некоторые оценки, рассчитанные для различных вариантов построения умножителей. Здесь приведены оценки времени умножения и требуемого числа корпусов интегральных схем с 16 выводами для трех умножителей чисел в дополнительном коде. Поскольку серийно выпускаемый четырехразрядный сумматор имеет 24 вывода, рассчитывалось эквивалентное количество ИС с 16 выводами; во всех остальных случаях использовались только корпуса с 16 выводами. Анализ этой таблицы позволяет сделать несколько интересных замечаний. Из первой строки видно, что специально разрабатываемые ИС (они будут описаны в разд. 8.8) имеют заметные преимущества перед серийными. Из сопоставления строк 2 и 3 (а также 6 и 7) можно

Таблица 8.2

## умножителей

Тип логики	Время умножения, нс			Количество корпусов		
	16×12	16×8	9×9	16×12	16×8	9×9
ЭСЛ	32	28	22	112	82	45
ЭСЛ	62	48	40	160	120	85
ТТЛ	155	100	85	160	120	85
ТТЛ	261	195	135	80	58	30
ЭСЛ	130	120	100	40	36	25
ЭСЛ	210	140	116	40	36	25
ТТЛ	450	300	250	40	36	25

таблицу, основанную на результатах их исследований.

оценить увеличение быстродействия умножителя при замене ИС ТТЛ на более быстродействующие ИС ЭЛС; оно возросло более чем в два раза. Влияние структуры умножителя как на его быстродействие, так и на количество корпусов можно проследить, обратившись к строкам 1 и 5; несколько неожиданно, что наилучший компромисс между быстродействием и числом корпусов достигается при построении матричных умножителей. Одной из причин является чрезвычайно простое управление матричным умножителем по сравнению с многотактным (строки 5, 6, 7). Интересно также отметить, что с точки зрения общего количества корпусов двухразрядный сумматор с 16 выводами (строка 1) оказывается выгоднее использовать, чем четырехразрядный сумматор с 24 выводами (строка 2 или 3). Таким образом, в некоторых структурах, когда кристалл размещается в корпусе больших размеров, увеличение уровня интеграции может оказаться даже *вредным*.

### 8.7. Делители и устройства с плавающей запятой

В системах цифровой обработки сигналов, как и в других вычислительных системах, основной упор делается на достижение быстродействия одних и тех же узлов. Самыми важными из них, несомненно, являются сумматоры и умножители, рассмотренные в предыдущих разделах. Перейдем теперь к изложению методов построения быстродействующих делителей, а также сумматоров с плавающей запятой. Дело в том, что алгоритмы цифровой фильтрации и БПФ используются в задачах все большей сложности, поэтому возникает необходимость представлять себе, как производятся операции, отличные от сложения и умножения. Оказывается, что для построения всех устройств, рассматриваемых в настоящем разделе, можно с успехом использовать идею нетактируемых матриц, достигая при этом большого быстродействия при умеренных размерах устройства. Можно ожидать, что уже в ближайшем будущем прогресс технологии производства интегральных схем лишь подтвердит справедливость этих утверждений.

#### 1. Четырехквadrанный матричный делитель

Рассмотрим четырехквadrанный делитель, составленный из комбинационной матрицы суммирующих и вычитающих логических элементов. Будем считать, что на вход схемы поступают 24-разрядное делимое (числитель) и 12-разрядный делитель (знаменатель), оба в дополнительном коде, а на выходе образуются 12-разрядные частное и остаток. Можно рассматривать случаи, когда делитель и делимое являются произвольными целыми или дробными числами или константами. Будем считать, что входные числа — правильные дроби с запятой, расположенной непосредственно

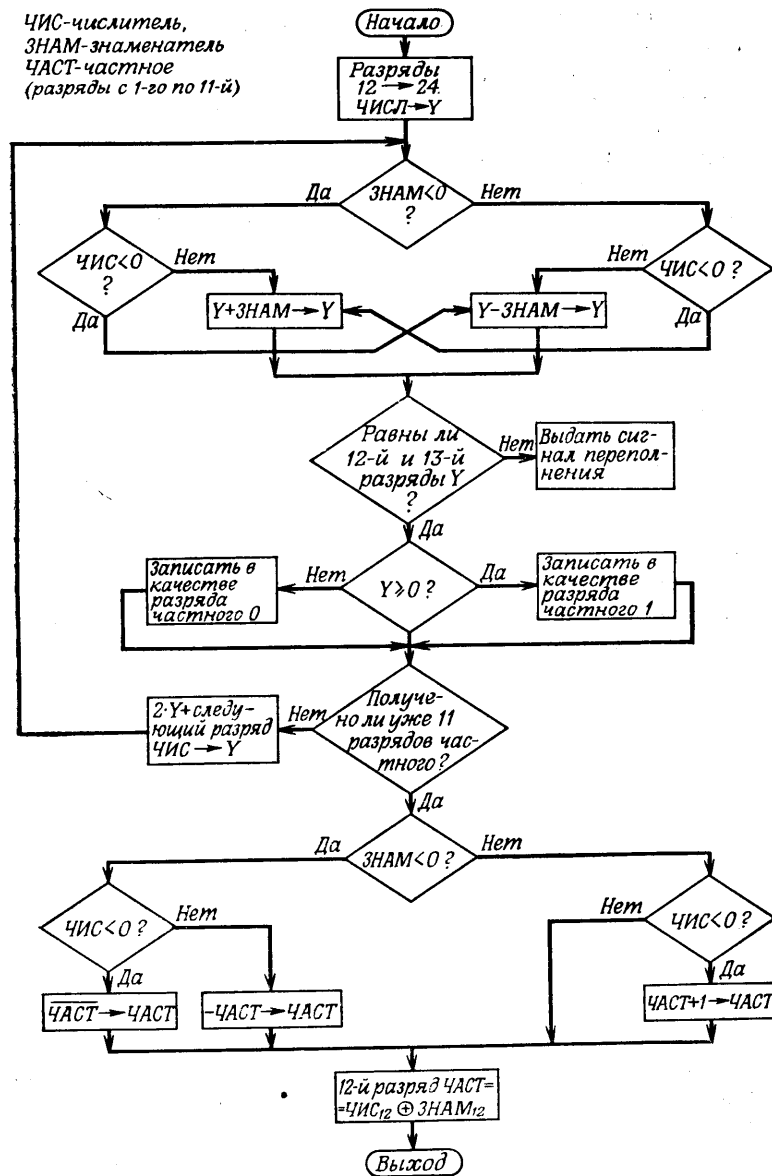
ственно правее знакового разряда; это будет учтено при построении логики индикации переполнения.

В основу работы матричной схемы положен метод деления без восстановления остатка. С целью упрощения описания этого метода примем сначала, что делимое и делитель — положительные дроби, причем делитель больше делимого или равен ему. При этих условиях частное также будет, очевидно, положительной правильной дробью. Для получения первого значащего разряда частного вычисляется частичное делимое как разность между делимым и пробным делителем, равным половине фактического. Если частичное делимое положительно, то в качестве разряда частного принимается единица. Если же оно отрицательно, т. е. если пробный делитель «не умещается» в делимом, то искомый разряд равен нулю. При обычном делении (с восстановлением остатка) в этом случае нужно сложить пробный делитель с частичным делимым для восстановления исходного делимого и попытаться вычесть новый пробный делитель. При делении без восстановления остатка используется тот факт, что каждый последующий пробный делитель в два раза меньше предыдущего. Поэтому сложение пробного делителя (т. е. операция восстановления) с последующим вычитанием его половины всегда будет совпадать с простым *добавлением* половины пробного делителя. Таким образом, если первый значащий разряд частного равен 1, то предыдущее значение пробного делителя делится пополам и затем вычитается из частичного делимого для получения следующего разряда частного. Если же первый разряд частного равен нулю, то пробный делитель делится пополам и затем добавляется к частичному делимому для получения следующего разряда частного. Эта процедура продолжится до тех пор, пока не будет получено требуемое количество разрядов частного. Преимущества этого алгоритма отчетливо выявляются при реализации его в виде нетактируемой матрицы. В нем нет обратных связей и шагов «назад», процесс протекает без всяких условий от начала и до конца.

Практическая реализация этого алгоритма для четырехквadrантного случая (когда приходится учитывать все возможные комбинации знаков делителя и делимого) требует некоторых дополнительных манипуляций. Основой матричного деления является последовательность этапов сложения и вычитания. На каждом этапе в зависимости от знаков частичного делимого и самого делителя производится либо сложение, либо вычитание соответствующего делителя из соответствующего частичного делимого. На матрицу может поступать любая комбинация знаков делителя и делимого, но в конце при определенных комбинациях необходима коррекция разрядов частного.

На верхней большей части блок-схемы рассматриваемого алгоритма (фиг. 8.36) представлена процедура получения разрядов

ЧИС-числитель,  
ЗНАМ-знаменатель  
ЧАСТ-частное  
(разряды с 1-го по 11-й)



Фиг. 8.36. Последовательность получения четырех разрядов частного с использованием алгоритма без восстановления остатка.

частного, а на нижней части показан процесс заключительной коррекции. Вычисление разрядов частного на каждом этапе производится по следующим правилам:

1. Если текущее частичное делимое положительно, в соответствующий разряд частного записывается 1. В противном случае записывается 0.

2. Если делитель и текущее частичное делимое имеют одинаковые знаки, следующий пробный делитель нужно *вычесть*.

3. Если делитель и текущее частичное делимое имеют разные знаки, следующий пробный делитель следует *добавить*.

Из приведенных выше правил вытекают два интересных следствия. Во-первых, если делимое положительно, то независимо от знака делителя частное будет правильным. Во-вторых, если делимое отрицательно, то независимо от знака делителя будет получен *обратный* код частного. Правильное значение частного в дополнительном коде получается после дополнительной коррекции, основанной на учете знаков операндов. Предположим, например, что делитель положительный, а делимое отрицательное. В этом случае отрицательное частное будет получено в обратном коде. Поэтому для перехода к дополнительному коду к результату необходимо добавить 1. Предположим теперь, что и делитель и делимое отрицательны. Ясно, что частное должно быть положительным. Но ответ будет получен в *обратном* коде, поэтому необходима простая инверсия разрядов частного. Во всех случаях коррекцию результата можно производить с помощью дополнительного этапа сложения (вычитания) на выходе матрицы. Лишь в одном случае коррекция не нужна — когда и делитель и делимое положительные. Коррекция осуществляется по следующим правилам.

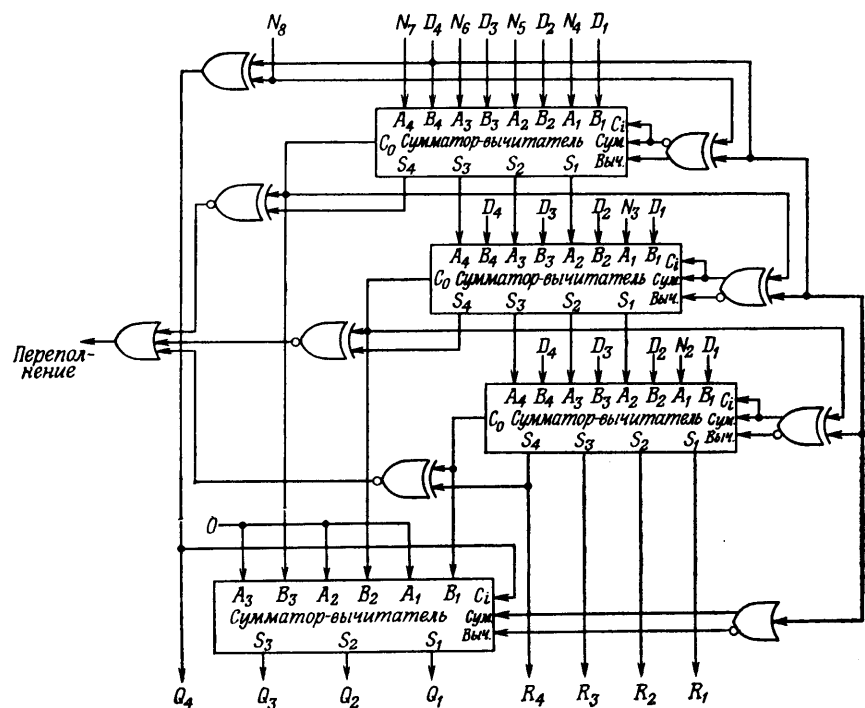
1. Если и делитель и делимое положительные, присвоить знаковому разряду частного значение 0 и *ничего не делать* со значащими разрядами.

2. Если и делитель и делимое отрицательные, присвоить знаковому разряду частного значение 0 и *проинвертировать* значащие разряды.

3. Если делитель положительный, а делимое отрицательное, присвоить знаковому разряду значение 1 и *добавить 1* в младший разряд частного.

4. Если делитель отрицательный, а делимое положительное, присвоить знаковому разряду значение 1, *проинвертировать* значащие разряды частного и *добавить* к нему 1.

В принципе при  $N$ -разрядном делителе (включая знаковый разряд) для представления всех сумм и разностей в матричной схеме достаточно  $(N + 1)$  разрядов. Разность (или сумма) между любым частичным делимым и его пробным делителем также может быть представлена *не более* чем  $N$  разрядами (включая знак). Таким образом, для рассматриваемого случая (фиг. 8.36) оба разряда,



Фиг. 8.37. Матричный делитель размером  $(4 \times 8)$  без сохранения остатка с индикацией переполнения и заключительной знаковой коррекцией (по Бланкеншину).

Числитель — 8 разрядов, знаменатель — 4 разряда, частное — 4 разряда, остаток — 4 разряда.

12-й и 13-й, отведенные во всех частичных делимых под знаковые, обязательно будут одинаковыми. В противном случае выдается сигнал переполнения. С точки зрения дробных операндов этот сигнал означает, что делимое по величине больше делителя. Такой режим запрещен, так как в этом случае частное превысит 1 и его, следовательно, нельзя представить правильной дробью, у которой все значащие разряды находятся справа от запятой. Это обстоятельство и фиксируется сигналом переполнения.

Частичное делимое, определяющее младший разряд частного [т. е. результат последнего этапа сложения (вычитания)], следует рассматривать как остаток. Он представляет собой 12-разрядное число (11 разрядов и знак) и следующим образом связан с другими операндами:

$$\text{Делимое} = (\text{Частное}) \times (\text{Делитель}) + \text{Остаток.}$$

Используя это соотношение, можно с большей точностью вычислить частное, взяв больше разрядов делимого. В действитель-

ности получение уточненного частного — довольно сложная операция, для выполнения которой необходимо знание величины остатка и его знака.

На фиг. 8.37 показан пример построения устройства для деления восьмиразрядного делимого на четырехразрядный делитель с четырехразрядными частным и остатком на выходе. Простым наращиванием его можно довести до 24/12 разрядов. Представленные на схеме сумматоры-вычитатели могут быть построены на основе двухнаносекундной четырехразрядной ИС ЭСЛ типа МС10181 (арифметико-логическое устройство). Эта ИС в зависимости от управляющего сигнала может быть запрограммирована или на сложение, или на вычитание. Вычитание осуществляется путем изменения знака числа на входе  $B$  с последующим суммированием. Фактически для выполнения этой операции требуется проинвертировать разряды  $B$  и добавить единицу. Инверсия внутри ИС предусмотрена, но для добавления единицы необходимо во время вычитания подать сигнал на вход  $C_{\text{вх}}$  (вход переноса в младший разряд).

В соответствии со сделанными ранее выводами следует ожидать, что для выполнения всех сложений (вычитаний) потребуется пять разрядов, так как делитель имеет четыре разряда. Однако можно показать, что  $(N + 1)$ -й разряд легко получить, используя выход переноса  $N$ -го разряда и немного изменив правила. Сформулируем коротко эти упрощенные правила:

1. *Первый этап.* Если знаки делителя и делимого не совпадают, следует производить сложение, если совпадают — вычитание.

2. *Все последующие этапы.* Если выходной сигнал переноса ( $C_{\text{вых}}$ )  $N$ -го разряда предыдущего сумматора-вычитателя равен 1, нужно записать 1 в разряд частного. Кроме того, если выходной сигнал переноса отличается от знакового разряда делителя, то на данном этапе следует произвести вычитание, а в противном случае — сложение.

3. *Переполнение.* Если на любом этапе выходной сигнал переноса  $N$ -го разряда совпадает с выходным сигналом суммы этого же  $N$ -го разряда, сформировать сигнал переполнения.

4. *Заключительная коррекция.* Это правило удобнее всего сформулировать в виде следующей таблицы:

Знак делителя ( $S_D$ )	Знак делимого ( $S_N$ )	Коррекция частного	Знак частного ( $S_Q$ )	Сложение	Вычитание	Вход переноса ( $C_{\text{вх}}$ )
+	+	Отсутствует	+	✓		
+	-	$Q + 1$	-	✓		✓
-	+	$\bar{Q} + 1$	-		✓	✓
-	-	$\bar{Q}$	+		✓	

Логические уравнения имеют вид

$$S_Q = C_{\text{вх}} = S_N \oplus S_D,$$

$$\text{ВЫЧ} = S_D,$$

$$\text{СЛОЖ} = \bar{S}_D.$$

В соответствии с этими уравнениями в рассматриваемой схеме осуществляется управление на этапе заключительной коррекции. Отметим, что входное число  $A$  в схеме коррекции всегда равно нулю.

Схема 24/12-разрядного делителя состоит из 12 12-разрядных каскадов. На первых 11 формируются разряды частного, на последнем выполняется коррекция. Каждый каскад состоит из трех ИС МС10181, включенных по схеме образования ускоренного переноса на 9 разрядов. Таким образом, всего требуется 36 корпусов МС10181. На каждом каскаде все необходимые разряды частичного делимого образуются за 13 нс, поэтому вся операция деления занимает  $12 \times 13 = 156$  нс. Количество корпусов, необходимых для управления и индикации переполнения, а также для распределения данных, незначительно. Следовательно, этот делитель требует меньшего количества корпусов, чем умножитель двенадцатиразрядных чисел, построенный на той же логической серии.

## 2. Цифровые устройства с плавающей запятой

Весьма часто динамический диапазон разрабатываемых систем имеет важное значение. Его можно значительно расширить, используя представление чисел с плавающей запятой. Это представление обычно применяется в задачах обработки больших числовых массивов, например при обращении матриц большого размера и при решении задач линейного программирования, т. е. когда значения чисел, получающихся в процессе вычислений, могут меняться в широких пределах.

Есть все основания для того, чтобы использовать представление чисел с плавающей запятой при создании цифровых фильтров и блоков БПФ. Рекурсивные цифровые фильтры с полюсами, расположенными вблизи единичной окружности, обладают большим усилением на частотах вблизи полюсов, поэтому уровень выходного сигнала может сильно меняться в зависимости от входной частоты. На каждом этапе БПФ уровень сигнала может увеличиться вдвое, поэтому при  $N$ -точечном БПФ возможно усиление в  $2^{10} \approx 2^N = N$  раз.

Использование плавающей запятой при фильтрации и БПФ, по-видимому, не создает дополнительных трудностей, связанных с округлением, но часто улучшает работу устройств. Однако при этом приходится использовать больший объем оборудования.

Во многих универсальных ЦВМ операции с плавающей запятой реализуются не аппаратными, а программными средствами, что приводит к удлинению программ и увеличению времени их выполнения. Алгоритмы работы устройств умножения и сложения с плавающей запятой сложнее, чем аналогичные алгоритмы с фиксированной запятой.

При умножении с плавающей запятой производится суммирование порядков сомножителей и умножение с фиксированной запятой их мантисс. Поскольку величина мантиссы (в десятичной системе) больше или равна 0,5, но меньше 1, то произведение двух мантисс заключено между 0,25 и 1. Это означает, что произведение будет либо выровнено по левому разряду<sup>1)</sup>, либо смещено на один разряд. В качестве упражнения составьте схему четырехразрядного устройства сложения порядков для умножителя с плавающей запятой.

При сложении с плавающей запятой необходимо выполнить следующие операции:

1. Определить, какой из двух порядков больше.
2. Из большего порядка вычесть меньший.
3. Полученную разность использовать для сдвига вправо мантиссы с меньшим порядком.
4. Сложить обе мантиссы.
5. Определить количество старших разрядов суммы, состоящих из одних нулей (или единиц).
6. Произвести выравнивание суммы по левому разряду и вычесть из большего порядка найденное количество одинаковых старших разрядов.

Очевидно, что устройство, выполняющее перечисленные операции, значительно сложнее сумматора с фиксированной запятой. Кроме того, данный алгоритм является довольно медленным, поскольку приходится выполнять последовательность операций, требующих принятия решения. По этой причине, несмотря на преимущества вычислений с плавающей запятой по сравнению с вычислениями с фиксированной запятой, выбор между ними часто неочевиден.

Основным узлом сумматора, в котором выполняются операции 3 и 6, является устройство сдвига на произвольное число разрядов. Проще всего эти операции выполнить с помощью последовательного регистра сдвига и счетчика числа сдвигов, однако затрачиваемое при этом время довольно значительно. Другой

<sup>1)</sup> При выравнивании по левому разряду мантиссу сдвигают влево настолько, насколько это возможно без переполнения. Синонимом термина «выравнивание по левому разряду» является *нормализация*. Если считать, что мантисса представляет собой правильную десятичную дробь со знаком, то после выравнивания по левому разряду десятичная запятая оказывается непосредственно справа от знакового разряда.





Для выполнения пятой операции при сложении с плавающей запятой необходимо иметь логическую схему подсчета количества одинаковых старших разрядов. На фиг. 8.39 и 8.40 показаны схемы, предназначенные для решения этой задачи для случая 12-разрядного числа. Первая схема (фиг. 8.39) преобразует 11 входных разрядов в 10 выходных. Количество единичных выходов определяется количеством сдвигов влево, необходимых для нормализации входного числа. Вторая схема (фиг. 8.40) составлена из полных сумматоров и выполняет операцию подсчета количества единичных выходов первой схемы.

### 8.8. Пример: проектирование быстродействующего матричного умножителя

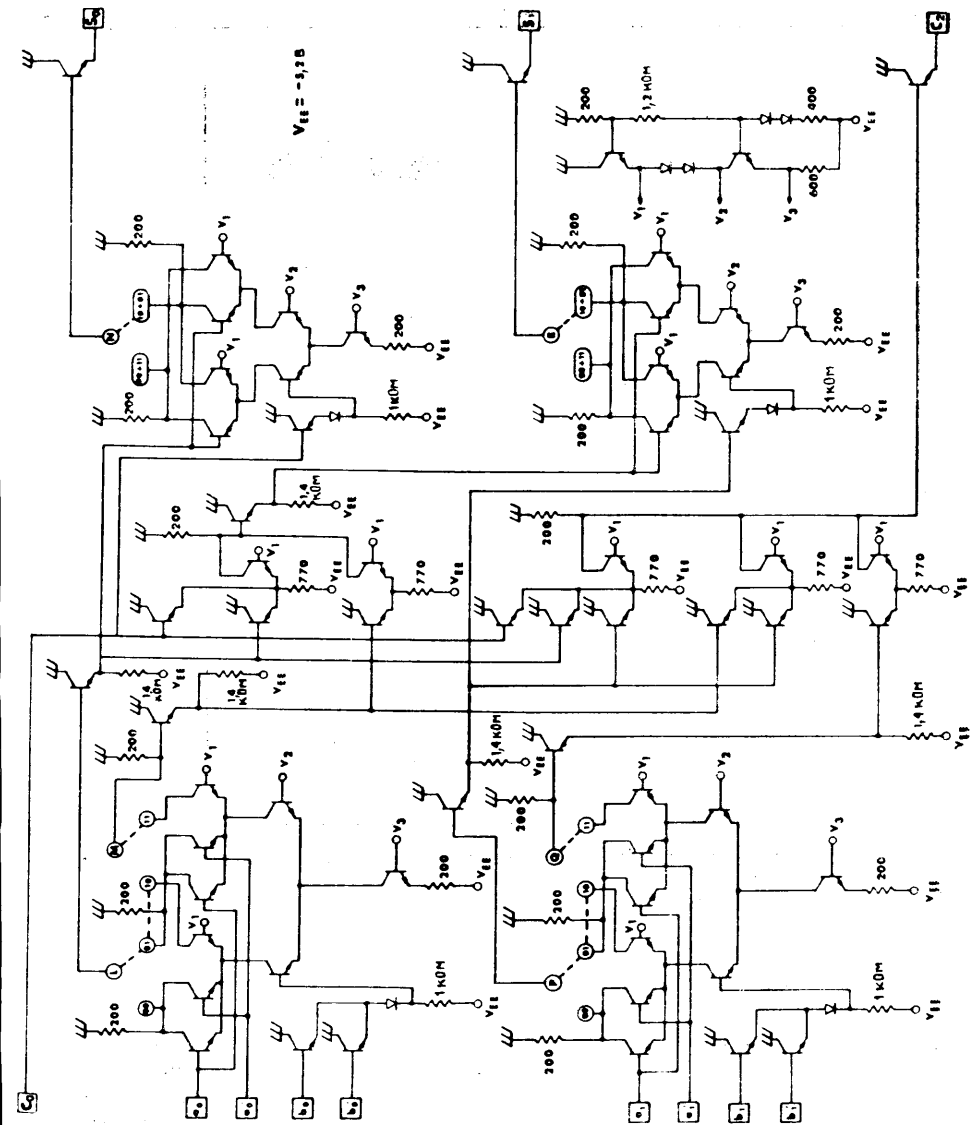
Серийно выпускаемые интегральные схемы рассчитаны на многоцелевое применение, поэтому построение на их основе цифровых систем, как правило, малоэффективно. В связи с этим целесообразно рассмотреть практический пример разработки специализированных микросхем, в которых были бы учтены конкретные требования проектируемой системы. Это позволит также проследить доказанную на фиг. 8.4 иерархию принципиальных схем, логических функций, микросхем и систем.

Рассмотрим построение быстродействующего умножителя. В разд. 8.6 было показано, что из известных схем наибольшее быстродействие обеспечивают матричные структуры, причем среди них самой быстродействующей является схема, изображенная на фиг. 8.35. Отметим, однако, что она содержит три различные логические ячейки, так что приходится разрабатывать три различные микросхемы в разном оформлении. Эту трудность удалось преодолеть, разработав одну микросхему, из которой все три модификации получаются путем незначительного изменения межсоединений. Кроме того, в кристалле были использованы два соединительных металлизированных слоя, причем для изменения межсоединений модифицировался только верхний слой.

В качестве основной микросхемы был принят двухразрядный сумматор. Наиболее важным параметром сумматора в системах рассматриваемого типа является задержка распространения переноса через два разряда. Поэтому при разработке микросхемы особое внимание было уделено минимизации этого параметра.

После того как сформулированы требования к интегральной схеме, ее разработка производится в следующей последовательности:

1. Составление принципиальной схемы.
2. Разработка трафаретов.
3. Изготовление и испытание ИС.



Фиг. 8.41. Схема двухразрядного сумматора L101 (по Пезарису).