Глава 4

ЛОГИКА, МАТЕМАТИКА И ЦИФРОВЫЕ СХЕМЫ

Логика — наука древняя. Ее основоположником считают Аристотеля, жившего в 384—322 годах до н. э.

Ко времени зарождения логики математика уже прошла значительный путь развития. Достаточно вспомнить, что в VI веке до н. э. жил древнегреческий мыслитель Пифагор Самосский. И хотя легенды о нем трудно отделить от действительности, но достоверно известно, что с его именем связано: введение доказательств в геометрию, создание учения о подобии, построение некоторых правильных многоугольников и др. Пифагор доказал в общем виде теорему о сторонах прямоугольного треугольника, носящую его имя, хотя, возможно, она была известна еще ранее.

Логика и математика, достигшие расцвета еще в глубокой древности, имеют принципиально неограниченную область применения: все виды доказательств основаны на законах логики, все количественные соотношения подвластны математике.

В средневековье развитие математики, и особенно логики, замедлилось, потому что новые логические идеи нередко входили в противоречие с формами мышления, основанными на преклонении перед авторитетом церкви. Оживление началось в XVI веке, а уже в XVII—XVIII веках появились предложения о «математизации логики».

Французский философ и математик Рене Декарт (1596—1650) считал, что человеческий разум может постигнуть истину, если будет исходить из достоверных положений, сводить сложные идеи к простым, переходить от известного и доказанного к неизвестному, избегая каких-либо пропусков в логических звеньях исследований. Фактически Декарт рекомендовал науке о мышлении — логике руководствоваться общепринятыми в математике принципами.

Немецкий философ и математик Готфрид Вильгельм Лейбниц предлагает использовать в логике математическую символику и впервые высказывает мысль о возможности применения двоичной системы счисления в вычислительной математике. Но этим идеям Лейбница суждено было получить дальнейшее развитие лишь в середине XIX века в трудах Джоржа Буля (1815—1864), именем которого назван раздел математики — булева алгебра, а также в трудах Огастена де Моргана (1806—1871), Уильяма Стенли Джевонса (1835—1882), Платона Сергеевича Порецкого (1846—1907), Чарлза Сандерса Пирса (1839—1914) и др.

Так зародилась новая наука — математическая логика.

4.1. Логика формальная и математическая

Появлению и развитию математической логики способствовало стремление найти строгие правила обоснования и доказательства новых положений в науке. По мнению некоторых ее создателей математическая логика должна была стать математическим аппаратом той части обычной логики, которую принято называть формальной.

Как и математика, формальная логика следует строгим правилам и не вникает в сущность анализируемых суждений. Одна из ее задач — установление формальных правил получения новых суждений из исходных, истинность которых не подвергается сомнению.

Логические значения высказываний обозначим следующим образом: истинность — цифрой 1, ложность — цифрой 0.

Рассмотрим один из видов сложных высказываний, название которых происходит от латинского *conjunctio* — соединяю. По определению *конъюнктивным* называется сложное высказывание, которое истинно тогда и только тогда, когда истинны все входящие в него высказывания. Пусть число исходных высказываний будет минимальным, т. е. равным двум. Обозначим эти высказывания буквами B и A. Все возможные наборы их значений и итогового сложного высказывания Y сведены в таблицу 4.1 или равнозначную ей таблицу 4.2.

Таблица 4.1

B	A	Y
ложно	ложно	ложно
ложно	истинно	ложно
истинно	ложно	ложно
истинно	истинно	истинно

Таблица 4.2

B	A	Y
0	0	0
0	1	0
1	0	0
1	1	1

На первый взгляд вторая таблица не имеет никаких преимуществ перед первой. Но давайте посмотрим, какую из таблиц легче запомнить. Первую нужно просто-напросто «зазубрить». Вторую же можно запомнить, руководствуясь простыми правилами.

Какие же в ней закономерности? Во-первых, комбинации 0 и 1 в первых двух колонках (B, A) образуют натуральный ряд 2-разрядных двоичных чисел: 00, 01, 10, 11, т. е. в десятичной системе 0, 1, 2, 3. Во-вторых, каждая цифра в колонке Y есть результат обычного математического действия умножения значений B и A в соответствующей этой цифре строке.

Таким образом, чтобы восстановить в памяти правила определения истинности сложного конъюнктивного суждения, достаточно уметь умножать нули и единицы во всех возможных сочетаниях или, что то же самое, знать таблицу умножения одноразрядных двоичных чисел:

$$0 \cdot 0 = 0$$
$$0 \cdot 1 = 0$$
$$1 \cdot 0 = 0$$
$$1 \cdot 1 = 1$$

Еще один вид сложных суждений обязан своим названием латинскому *disjunctio* — разобщение, различие. В формальной логике неисключающим дизъюнктивным суждением называется сложное суждение, которое истинно тогда и только тогда, когда истинно хотя бы одно из входящих в него суждений. Как и в предыдущем случае, составим две таблицы, соответствующие сформулированному определению.

Таблица 4.3

B	A	Y
ложно	ложно	ложно
ложно	истинно	истинно
истинно	ложно	истинно
истинно	истинно	истинно

Таблица 4.4

B	A	Y
0	0	0
0	1	1
1	0	1
1	1	1

Математическую операцию, которая может быть пояснена таблицей 4.3, называют в *булевой алгебре* дизъюнкцией, или логическим суммированием. Применяя для ее обозначения знак «+», можем записать:

$$Y = B + A.$$

Еще раз напомним, что это не обычное, а логическое суммирование, поэтому

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 1$$

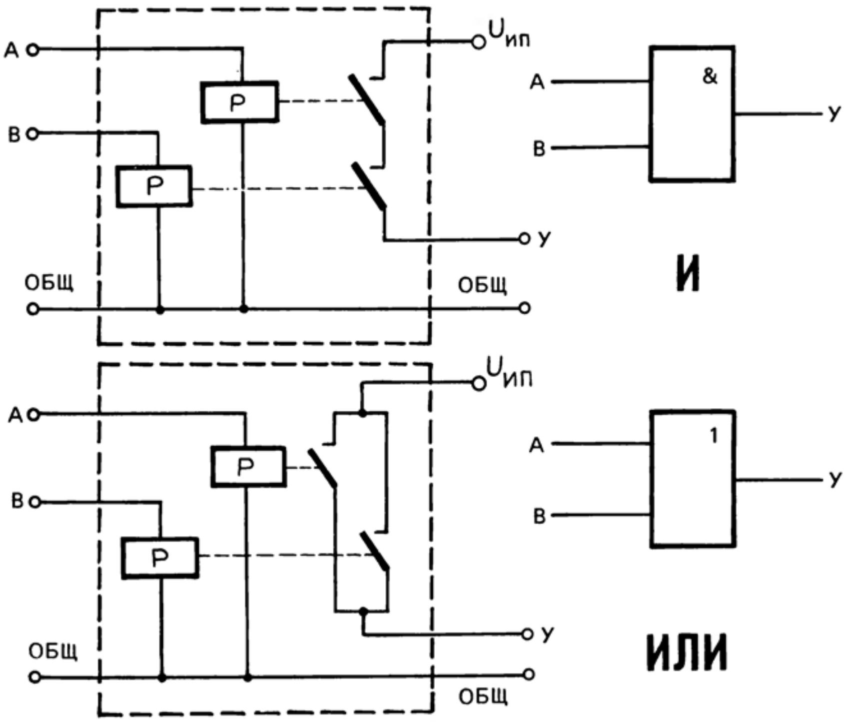
4.2. Схемы И, ИЛИ и НЕ

Начнем с бытовых аналогий. Если дверь закрывается на два (или три) замка, она сконструирована как своего рода схема И. В помещении с такой дверью можно войти, открыв **и** первый, **и** второй (**и** третий, если он есть) замок.

Но не торопитесь с выводами. Эту же дверь можно считать и схемой ИЛИ, так как она будет закрыта в случаях, когда закрыт **или** первый, **или** второй, **или** третий, **или** любая пара замков, **или** все три.

Чтобы избежать путаницы, условимся, как это принято в вычислительной технике, считать схемами И, ИЛИ только такие устройства, у которых сигналы на входах и выходах имеют одинаковую физическую природу. Обычно за 0 принимается низкий, а за единицу высокий уровень напряжения. В принципе схемы И и ИЛИ могут быть электрическими, механическими, пневматическими и иными, важно лишь, повторяем, чтобы 0 и 1 были идентичными как на входе, так и на выходе.

Рис. 4.1. Схемы И и ИЛИ на основе реле. В общем случае каждый вход и выход представляет собой пару проводов. Внешние соединения упрощаются, если один из проводов каждой такой пары — общий. Условные обозначения логических схем одинаковы независимо от того, из каких элементов они построены. Символ & обозначает «и».



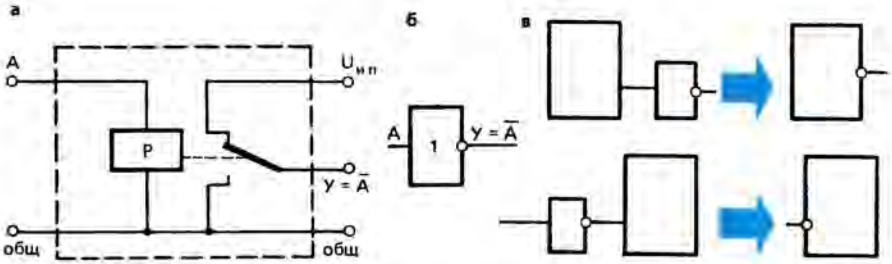


Рис. 4.2. В схеме НЕ на основе реле *(а)* при замыкании входа *А* с общим проводом выход *У* отключается от этого провода. Условное обозначение элемента НЕ *(б)*. Обозначения упрощаются, если инвертирование (отрицание) обозначается кружочком на выходе или входе логической схемы *(в)*.

Теперь позволительно перейти и к строгим определениям.

1. **Схема И** — устройство, имеющее два или несколько входов и один выход. Сигнал 1 появляется на выходе только в том случае, когда сигналы на всех входах 1.

2. **Схема ИЛИ** — устройство с одним выходом и двумя или несколькими входами. Логическая 1 появляется на выходе, когда хотя бы на одном входе есть 1.

Однотипность сигналов на входах и выходах позволяет подключать выходы одних схем ко входам других, причем к одному выходу, если это необходимо, могут быть присоединены входы нескольких схем.

На рисунке 4.1 показано, как можно сделать схемы И и ИЛИ. Здесь же приведены условные обозначения схем, которыми удобно пользоваться независимо от того, каково их конкретное устройство.

Обратите внимание на то, что работа схемы И поясняется операцией логического умножения и таблицей 4.2, а схемы ИЛИ — операцией логического суммирования и таблицей 4.4.

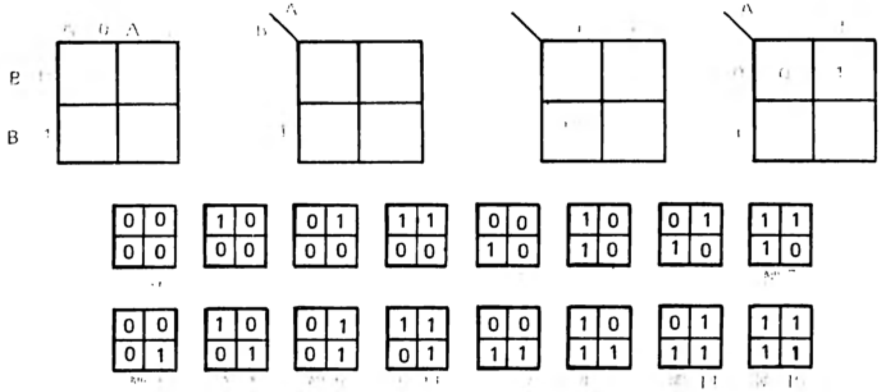
3. Рассмотрим **схему НЕ.** Она имеет один вход и один выход и превращает 0 в 1, а 1 в 0. Такое преобразование, называемое отрицанием, или инверсией, обозначается черточкой над преобразуемой переменой: $Y = \bar{A}$. В отличие от схем И и ИЛИ схему НЕ невозможно построить без активного элемента, т. е. без устройства, способного увеличивать мощность электрического сигнала — транзистора, электронной лампы или электромагнитного реле.

На рисунке 4.2 показано, как можно сделать схему НЕ на основе реле, если, как и ранее, за 0 принят низкий уровень напряжения, а за 1 — высокий.

4.3. Поиграем в «крестики-нолики»

Предлагаемое занятие действительно напоминает игру в крестики-нолики, только вместо крестика будет ставиться единица.

Поле для игры или диаграмма — четыре клеточки, образующие квадрат. Правила игры — в каждую клеточку нужно обяза-



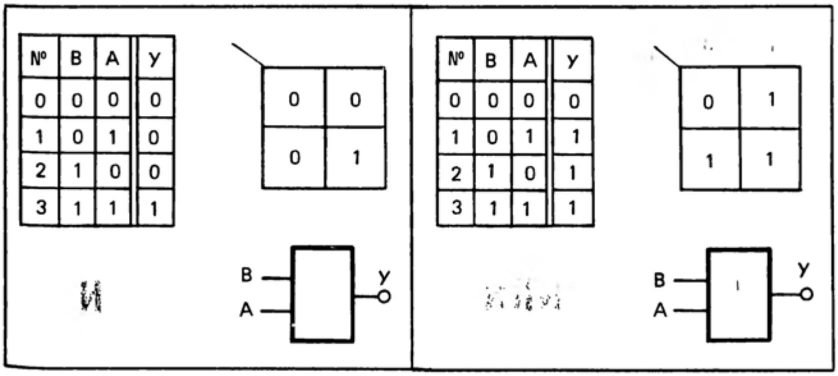
Четырехклеточные диаграммы—одна из форм описания работы двухвходовых цифровых схем. Сигналам на входе B соответствуют строки, а сигналам на входе A — столбцы диаграмм. В клеточках, каждая из которых соответствует строго определенному набору сигналов BA, записывают сигналы на выходе системы (№ № 0...15).

тельно вписать 0 или 1. Выигрывает тот, кто предложит больше отличающихся друг от друга вариантов заполнения диаграммы.

Если играющему известна двоичная система счисления, то он никогда не проиграет. Более того, он с уверенностью может заявить, что существует ровно 16 вариантов различного заполнения такой четырехклеточной диаграммы, так как предложенная задача равносильна вопросу: «Сколько существует различных четырехразрядных двоичных чисел?»

Все возможные варианты заполнения четырехклеточной диаграммы показаны на рисунке 4.3. Они имеют непосредственное отношение к логическим цифровым схемам. Чтобы убедиться в этом, пронумеруем столбцы и строки цифрами 0 и 1.

Различные формы описания работы схем И и ИЛИ.



На рисунке 4.4 воспроизведены парами таблица 4.2 и диаграмма № 8 с рисунка 4.3, а также таблица 4.4 и диаграмма № 14. Нетрудно заметить, что эти пары полностью равнозначны, т. е. схему И можно описать диаграммой № 8, а схему ИЛИ — диаграммой № 14.

Действительно, каждой клеточке диаграммы (как и каждой строке таблицы) соответствует вполне определенный набор значений B и A, а цифра в этой же клеточке указывает, какой сигнал получается на выходе схемы при таком наборе.

Напрашивается вывод, что рассмотренные схемы И и ИЛИ — лишь отдельные представители семейства из 16 возможных двухвходовых схем. Не все эти схемы в полном смысле двухвходовые. На устройства с № 0 и № 15 вообще не оказывает влияния ни сигнал A, ни сигнал B, устройства с № 3, 5, 10, 12 реагируют на изменение сигнала только на одном из входов. Подробно об этом расскажем позже. Теперь же ознакомимся с основами той алгебры, которая описывает эти устройства.

4.4. Вся алгебра в одной функции!

В алгебре буквенные обозначения заменяют произвольные числа — положительные, отрицательные, дробные, мнимые и др. В булевой алгебре буквами обозначены переменные, которые могут принимать только два значения — 0 или 1. Функции от любого количества таких переменных, называемые *переключательными*, или *булевыми*, также могут принимать значение только 0 или 1. Поэтому существует лишь строго определенное количество функций с заданным числом переменных. Его нетрудно определить.

Сначала установим, сколько может быть наборов переменных. Если переменная одна, т. е. функция имеет вид $\Phi(A)$, то таких наборов два: $A=0$ и $A=1$. При двух переменных, т. е. для функции $\Phi(B, A)$, таких наборов значений BA четыре: 00, 01, 10, 11. У функции трех переменных $\Phi(C,B,A)$ наборов CBA восемь: 000, 001, 010, 011, 100, 101, 110, 111. При числе переменных n количество наборов $m=2^n$, и они представляют собой натуральный ряд n-разрядных двоичных чисел от 0 до 2^n-1.

Переключательная функция при каждом наборе переменных принимает значение 0 или 1, поэтому отличающихся друг от друга функций может быть ровно столько, сколько существует различных комбинаций из 2^n единиц и нолей. Таких комбинаций 2^m, и они представляют собой последовательность m-разрядных двоичных чисел от 0 до 2^m-1. Таким образом, общее количество функций n переменных равно

$$2^m=(2)^{2n}$$

Не стоит расстраиваться из-за такого быстрого роста числа функций с увеличением n: знакомиться с ними нет никакой необходимости хотя бы потому, что **функция любого количества**

переменных может быть выражена через функции только двух переменных. Делается это при помощи очень простых приемов. Во-первых, путем подстановки на место переменных других булевых функций (а это можно делать, так как и функция и переменные принимают только два значения — 0 и 1). Во-вторых, перестановкой переменных местами («перенумерацией»). Оба эти приема объединяются общим названием — *суперпозиция.*

В таблицы 4.5 и 4.6 сведены соответственно все функции одной и двух переменных. Однако многие функции таблицы никак не назовешь зависящими от двух переменных. Некоторые из них вообще не зависят ни от одной переменной (Φ_0, Φ_{15}), так как при любых значениях переменных их значения остаются неизменными.

Таблица 4.5

Функция	Значения функции		Название	Обозначение
	при $A = 1$	при $A = 0$		
$\Phi_0(A)$	0	0	константа 0	0
$\Phi_1(A)$	0	1	отрицание (инверсия) A	$\bar{A}$
$\Phi_2(A)$	1	0	переменная A	A
$\Phi_3(A)$	1	1	константа 1	1

Таблица 4.6

Функция	Значения при		Название
	$B=$	1 1 0 0	
	$A=$	1 0 1 0	
$\Phi_0(B, A)$		0 0 0 0	константа 0
$\Phi_1(B, A)$		0 0 0 1	отрицание дизъюнкции
$\Phi_2(B, A)$		0 0 1 0	
$\Phi_3(B, A)$		0 0 1 1	отрицание B (инверсия)
$\Phi_4(B, A)$		0 1 0 0	
$\Phi_5(B, A)$		0 1 0 1	отрицание A (инверсия)
$\Phi_6(B, A)$		0 1 1 0	неравнозначность (сумма по модулю 2)

Функция	Значения при		Название
	$B=$	1 1 0 0	
	$A=$	1 0 1 0	
$\Phi_7(B, A)$		0 1 1 1	отрицание конъюнкции
$\Phi_8(B, A)$		1 0 0 0	конъюнкция
$\Phi_9(B, A)$		1 0 0 1	равнозначность
$\Phi_{10}(B, A)$		1 0 1 0	переменная A
$\Phi_{11}(B, A)$		1 0 1 1	
$\Phi_{12}(B, A)$		1 1 0 0	переменная B
$\Phi_{13}(B, A)$		1 1 0 1	
$\Phi_{14}(B, A)$		1 1 1 0	дизъюнкция
$\Phi_{15}(B, A)$		1 1 1 1	константа 1

Нельзя ли, применяя метод суперпозиции, выразить одни из этих функций через другие, а если можно, то каково минимальное количество функций двух переменных, через которые этим методом выражаются все остальные функции?

Оказывается, что *функционально полный набор булевых функций,* т. е. такой набор, через функции которого методом суперпозиции выражается любая булева функция двух переменных, может содержать всего одну функцию! При желании к ней можно свести всю алгебру логики.

Несколько примеров функционально полных наборов:

1) функции Φ_5 и Φ_8;
2) функции Φ_3 и Φ_{14};
3) одна функция Φ_1;
4) одна функция Φ_7.

Сравнив таблицу 4.6 с рисунком 4.3, легко заметить, что каждая строка таблицы и диаграмма с таким же порядковым номером несут одну и ту же информацию. И та и другая отвечает на вопрос, чему равна функция (сигнал на выходе) при каждом конкретном наборе значений переменных (входных сигналов). Таблица 4.6 в целом эквивалентна набору из 16 диаграмм на рисунке и содержит все сведения о работе любой двухвходовой цифровой схемы. Поэтому логические схемы, для описания которых применяются функции Φ_1 и Φ_7, т. е. *схемы ИЛИ-НЕ* и *И-НЕ,* также универсальны, как и сами функции. Любое логическое и арифметическое устройство любой ЭВМ может быть сделано из логических схем только одного типа, например только двухвходовых схем И-НЕ.

4.5. Как доказать теоремы булевой алгебры

Любое доказательство основано на положениях, считающихся справедливыми. Такие исходные положения не доказываются и называются аксиомами. *Аксиомы булевой алгебры* — это краткая математическая форма записи того, что переменные принимают только два значения — 0 и 1; что операция инверсии (отрицания) превращает 0 в 1 и наоборот и, наконец, что логическое умножение и суммирование выполняют так, как это предписано таблицами 4.2 и 4.4. Вот эти аксиомы:

$$A=0,\ \text{если}\ A\neq 1; \qquad A=1,\ \text{если}\ A\neq 0; \tag{4.1}$$
$$A=0;\ \bar{A}=1; \qquad A=1;\ \bar{A}=0; \tag{4.2}$$
$$A+0=A; \qquad A\cdot 1=A; \tag{4.3}$$
$$A+1=1; \qquad A\cdot 0=0; \tag{4.4}$$
$$A+A=A; \qquad A\cdot A=A; \tag{4.5}$$
$$A+\bar{A}=1; \qquad A\cdot\bar{A}=0. \tag{4.6}$$

На них основано доказательство *теорем булевой алгебры,* называемых по традиции законами. С помощью этих законов можно преобразовывать и упрощать выражения. Убедиться в справедливости законов, т. е. фактически доказать теоремы, можно при помощи подстановки на место переменных всех возможных сочетаний 0 и 1.

Закон коммутативности утверждает, что

$$A+B=B+A\ \text{и}\ A\cdot B=B\cdot A \tag{4.7}$$

Его доказательство предлагаем сделать самому читателю. Мы же докажем первый **закон поглощения** переменной:

$$A+A\cdot B=A\ \text{и}\ A(A+B)=A. \tag{4.8}$$

Проверку выполнения этих равенств при всех наборах A и B представим в виде таблицы 4.7.

Таблица 4.7

Наборы		Закон поглощения	
A	B	$A+A\cdot B=A$	$A(A+B)=A$
0	0	$0+0\cdot 0=0$	$0(0+0)=0$
0	1	$0+0\cdot 1=0$	$0(0+1)=0$
1	0	$1+1\cdot 0=1$	$1(1+0)=1$
1	1	$1+1\cdot 1=1$	$1(1+1)=1$

Все равенства выполняются, поэтому первый закон поглощения доказан.

Аналогично доказываются и остальные законы.

Второй закон поглощения переменной:

$$A + \bar{A} \cdot B = A + B; \; A(\bar{A} + B) = A \cdot B. \tag{4.8a}$$

Закон ассоциативности:

$$(A + B) + C = A + (B + C) = A + B + C. \tag{4.9}$$
$$(A \cdot B) \cdot C = A \cdot (B \cdot C) = A \cdot B \cdot C.$$

Закон дистрибутивности:

$$A + B + C = (A + B) \cdot (A + C). \tag{4.10}$$
$$A(B + C) = A \cdot B + A \cdot C.$$

Закон инверсии:

$$\bar{A} + \bar{B} = \overline{A \cdot B}; \qquad \bar{A} \cdot \bar{B} = \overline{A + B}. \tag{4.11}$$

Поскольку закон 4.11 не столь очевиден, как предыдущие, проверим его, сведя результаты проверки в таблицу 4.8.

Таблица 4.8

Наборы		Закон инверсии	
A	B	$\overline{A + B} = \bar{A} \cdot \bar{B}$	$\overline{A \cdot B} = \bar{A} + \bar{B}$
0	0	$\overline{0 + 0} = \bar{0} \cdot \bar{0}$, т. е. $1 = 1$	$\overline{0 \cdot 0} = \bar{0} + \bar{0}$, т. е. $1 = 1$
0	1	$\overline{0 + 1} = \bar{0} \cdot \bar{1}$, т. е. $0 = 0$	$\overline{0 \cdot 1} = \bar{0} + \bar{1}$, т. е. $1 = 1$
1	0	$\overline{1 + 0} = \bar{1} \cdot \bar{0}$, т. е. $0 = 0$	$\overline{1 \cdot 0} = \bar{1} + \bar{0}$, т. е. $1 = 1$
1	1	$\overline{1 + 1} = \bar{1} \cdot \bar{1}$, т. е. $0 = 0$	$\overline{1 \cdot 1} = \bar{1} + \bar{1}$, т. е. $0 = 0$

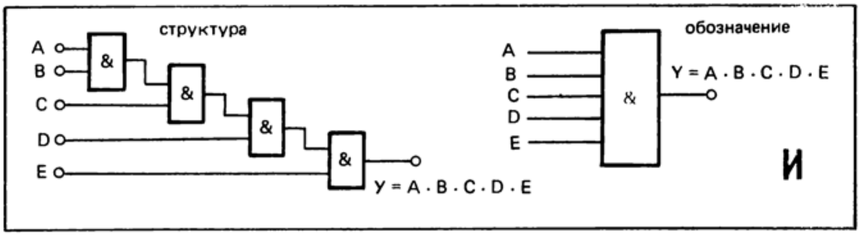
4.6. Как синтезируют комбинационные схемы

Если алфавит, применяемый на входе и выходе КС (см. 2.2 и 2.3), чисто двоичный, то каждому двоичному коду на входе КС ставит в соответствие вполне определенный код на выходе или, что то же самое, каждой комбинации нулей и единиц на входе соответствует строго определенная комбинация нулей и единиц на выходе (отсюда и название — комбинационная).

Любую КС, имеющую m выходов, можно заменить набором из m комбинационных схем, каждая из которых имеет только один выход и такое же количество входов, как и исходная КС. Очевидно, для постройки любой КС достаточно научиться составлять (синтезировать) произвольную КС с одним выходом. Именно они будут называться далее просто КС.

Итак, комбинационной будем называть схему, сигнал на выходе которой однозначно определяется комбинацией сигналов на входах. К ним относятся, в частности, схемы И и ИЛИ.

Из двухвходовых схем И и из двухвходовых схем ИЛИ можно создать схемы И и ИЛИ с любым количеством входов. Как это делается — показано на рисунке 4.5. Зная свойства двухвходовых



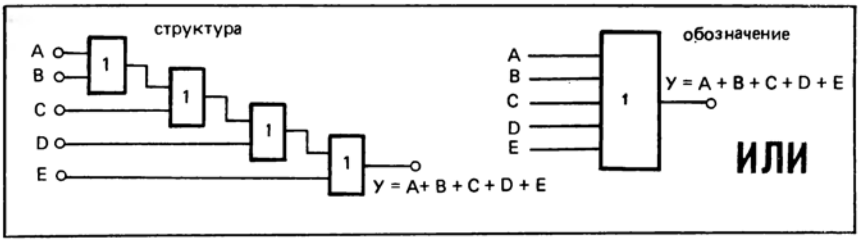


Рис. 4.5. Из двухвходовых схем можно построить многовходовые схемы И и ИЛИ.

схем, легко проверить, что многовходовые схемы И и ИЛИ работают в соответствии с определением, данным в 4.3.

Любую другую комбинационную схему также можно синтезировать путем соединения между собой схем И, ИЛИ и НЕ.

Свойства комбинационной схемы полностью определяет *таблица значений* соответствующей логической функции, в которой указывается, какой сигнал должен быть на выходе при каждом возможном наборе сигналов на входах.

Синтез произвольной КС по заданной таблице значений заключается в получении выражения булевой алгебры, описывающего работу схемы, в упрощении этого выражения *(минимизации)* и, наконец, в составлении самой искомой схемы по минимизированному выражению.

Один из способов такого синтеза основывается на использовании так называемых *простых конъюнкций*. При числе переменных n (т. е. при синтезе n-входовой схемы) число простых конъюнкций равно 2^n.

В таблицах 4.9 показано, как образуются простые конъюнкции для $n=2$ и $n=3$. При большем n принцип остается тем же. Данному набору переменных соответствует простая конъюнкция, т. е. произведение всех без исключения переменных, в котором переменные, равные 0, берутся со знаком инверсии, а равные 1 — без него.

Так, например, простая конъюнкция для набора $E=0$, $D=1$, $C=0$, $B=0$, $A=1$ есть $Y_9=\bar{E}\cdot D\cdot\bar{C}\cdot\bar{B}\cdot A$.

Выражение булевой алгебры для описания синтезируемой схемы получают на основании теоремы: **любая булева функция может быть представлена в виде логической суммы простых**

Таблица 4.9

№	B	A	простая конъюнкция
0	0	0	$\bar{B}\cdot\bar{A}$
1	0	1	$\bar{B}\cdot A$
2	1	0	$B\cdot\bar{A}$
3	1	1	$B\cdot A$

$n=2$

№	C	B	A	простая конъюнкция
0	0	0	0	$\bar{C}\cdot\bar{B}\cdot\bar{A}$
1	0	0	1	$\bar{C}\cdot\bar{B}\cdot A$
2	0	1	0	$\bar{C}\cdot B\cdot\bar{A}$
3	0	1	1	$\bar{C}\cdot B\cdot A$
4	1	0	0	$C\cdot\bar{B}\cdot\bar{A}$
5	1	0	1	$C\cdot\bar{B}\cdot A$
6	1	1	0	$C\cdot B\cdot\bar{A}$
7	1	1	1	$C\cdot B\cdot A$

$n=3$

конъюнкций, соответствующих тем наборам переменных, при которых эта функция принимает значение 1.

Воспользуемся теоремой и получим в качестве примера булевы выражения, описывающие некоторые двухвходовые схемы в таблице 4.3. Для схемы № 6 (ИСКЛЮЧАЮЩЕЕ ИЛИ) единица на выходе появляется при наборах: 1) $B=0$, $A=1$ и 2) $B=1$, $A=0$, поэтому

$$Y_6=\bar{B}\cdot A+B\cdot\bar{A}. \tag{4.12}$$

Для схемы И-НЕ (№ 7) имеем:

$$Y_7=\bar{B}\cdot\bar{A}+\bar{B}\cdot A+B\cdot\bar{A}. \tag{4.13}$$

4.7. **Минимизация**

Некоторые выражения, составленные по только что сформулированной теореме, можно упростить, применяя аксиомы и законы булевой алгебры 4.1—4.11. Дело это несложное, но требует аккуратности и внимания, а допущенная при процедуре упрощения неточность приводит к ошибочному результату. Чтобы исключить подобного рода неточности и избавиться от однообразной и малоинтересной работы, упрощения выражений производят при помощи диаграмм минимизации.

Диаграммы минимизации содержат 2^n клеточек, каждая из которых предназначена для одной из простых конъюнкций или, что то же самое, для одного набора значений переменных. В случае $n=2$ диаграмма для минимизации совпадает с принятой в 4.4 (рис. 4.3 и 4.6, *а*) формой представления таблицы значений функции двух переменных. В этой диаграмме любые соседние клетки отличаются лишь появлением или исчезновением знака инверсии над одной переменной, а соответствующие им наборы переменных отличаются лишь одной цифрой. Этот же принцип сохраняется и при большем количестве переменных.

Для $n=3$ диаграмма строится путем пририсовки к диаграмме для двух переменных ее зеркального изображения (рис. 4.6, *б*). Всем клеточкам «старой» части ставится в соответствие новая

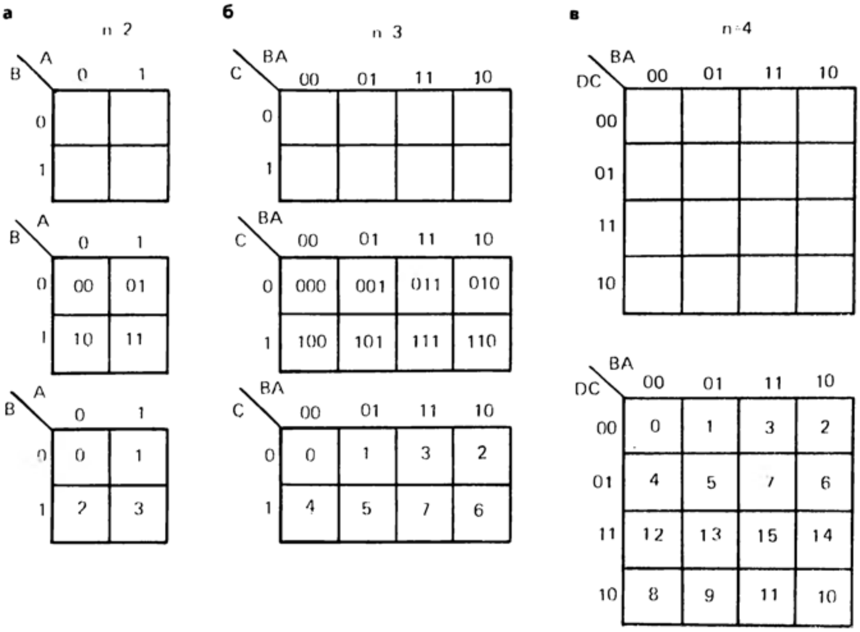


Рис. 4.6. Диаграммы минимизации для 2 (*а*), 3 (*б*) и 4 (*в*) переменных.

переменная, равная 0, «пририсованной» — равная 1. Таким же способом строят диаграммы для любого количества переменных, например для $n=4$ (рис. 4.6, *в*).

Используя диаграммы, нет необходимости записывать булево выражение искомой комбинационной схемы. Достаточно лишь занести 1 в клеточки, соответствующие наборам переменных, при которых на выходе комбинационной схемы должен быть сигнал 1 (вспомните «крестики-нолики»!).

Если занесенные единицы образуют прямоугольник из 2^n клеточек, то логическая сумма соответствующих им простых конъюнкций может быть заменена укороченной конъюнкцией, содержащей лишь те переменные, значения которых одинаковы для всех клеточек данного прямоугольника. Для этого такие массивы из 2, 4, 8 и т. д. клеточек обводятся контурами. Искомое минимизированное булево выражение есть сумма конъюнкций контуров, охватывающих все без исключения единицы. При этом крайние правый и левый столбцы диаграммы и верхняя и нижняя ее строки считаются соседними, а контуры могут перекрываться. Чем объясняется такая процедура минимизации, поясним на примере выражения (4.13). Сначала упростим его обычным методом, используя формулы (4.5), (4.7), (4.8) и (4.10):

$$\begin{aligned} Y_7 &= \bar{B}\cdot\bar{A} + \bar{B}\cdot A + B\cdot\bar{A} = \bar{B}\cdot\bar{A} + \bar{B}\cdot A + \bar{B}\cdot\bar{A} + B\cdot\bar{A} = \\ &= (\bar{B}\cdot\bar{A} + \bar{B}\cdot A) + (\bar{B}\cdot\bar{A} + B\cdot\bar{A}) = \\ &= \bar{B}(\bar{A} + A) + \bar{A}(\bar{B} + B) = \bar{B} + \bar{A}. \end{aligned} \tag{4.14}$$

Теперь проделаем то же самое, используя диаграмму минимизации для двух переменных (табл. 4.10). Укороченная конъюнкция, получающаяся в результате оконтуривания клеточек $\overline{B} \cdot A$ и $\overline{B} \cdot \overline{A}$, есть просто $\overline{B}$, а укороченная конъюнкция от контура с клеточками $\overline{B} \cdot \overline{A}$ и $B \cdot \overline{A}$ есть просто $\overline{A}$. Таким образом, сразу же можно записать

$$Y_7 = \overline{B} + \overline{A},$$

что совпадает с (4.14). Приведем еще два примера составления и минимизации выражений, описывающих действие схем, заданных таблицами значений (табл. 4.11, 4.12).

Таблица 4.10

B \ A	0	1
0	1	1
1	1	

или

B \ A	$\overline{A}$	A
$\overline{B}$	1	1
B	1	

Таблица 4.11

№	C B A	Y
0	0 0 0	1
1	0 0 1	0
2	0 1 0	1
3	0 1 1	0
4	1 0 0	1
5	1 0 1	0
6	1 1 0	1
7	1 1 1	1

$Y = \overline{A} + C \cdot B$

C \ BA	00	01	11	10
0	1			1
1	1		1	1

Таблица 4.12

№	D C B A	Y
0	0 0 0 0	0
1	0 0 0 1	1
2	0 0 1 0	0
3	0 0 1 1	1
4	0 1 0 0	1
5	0 1 0 1	1
6	0 1 1 0	1
7	0 1 1 1	1
8	1 0 0 0	0
9	1 0 0 1	1
10	1 0 1 0	1
11	1 0 1 1	1
12	1 1 0 0	0
13	1 1 0 1	1
14	1 1 1 0	0
15	1 1 1 1	1

$Y = A + \overline{D} \cdot C + D \cdot \overline{C} \cdot B$

DC \ BA	00	01	11	10
00		1	1	
01	1	1	1	1
11		1	1	
10		1	1	1

4.8. А как же с универсальной схемой и одной функцией?

Заявив в 4.4 о том, что любое устройство цифровой электроники можно построить из схем одного типа, мы почему-то забыли о такой возможности и синтезировали комбинационные схемы из схем И, ИЛИ, НЕ.

Все объясняется очень просто. Рассмотрен только один из возможных способов синтеза и минимизации. Он более прост и нагляден, чем другие, так как действие схем И, ИЛИ и НЕ легко пояснить при помощи простых аналогий (см. 4.2). Можно было бы рассмотреть способы синтеза и минимизации на основе схем одного типа, таких, как ИЛИ-НЕ, но мы поступим иначе. Покажем, что только из схем каждого из этих типов могут быть получены схемы И, ИЛИ и НЕ. Тем самым мы подтвердим, что, в принципе, из этих схем могут быть получены и любые комбинационные схемы. Как это может быть сделано — ясно из рисунка 4.7.

Рис. 4.7. С помощью законов инверсии (4.11) еще раз можно убедиться в универсальности двухвходовых схем И-НЕ и ИЛИ-НЕ.