Высшее профессиональное образование

БАКАЛАВРИАТ

Т. Ф. ЩЕРБАКОВА, С. В. КОЗЛОВ, А. А. КОРОБКОВ

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Рекомендовано
Учебно-методическим объединением по образованию
в области инфокоммуникационных технологий и систем связи
в качестве учебного пособия для студентов
высших учебных заведений, обучающихся по направлению
подготовки «Инфокоммуникационные технологии и системы связи»
квалификации (степени) «бакалавр» и квалификации (степени) «магистр»

ACADEMIA

Москва
Издательский центр «Академия»
2012

УДК 681.3(075.8)
ББК 32.81я73
Щ612

Рецензенты:

д-р техн. наук, проф. *Г. И. Ильин*; заместитель генерального директора ООО «АМФИТЕЛ ПЛЮС», канд. техн. наук *А. Л. Овчинников*

Щербакова Т. Ф.

Щ612 Вычислительная техника и информационные технологии : учеб. пособие для студ. учреждений высш. проф. образования / Т. Ф. Щербакова , С. В. Козлов, А. А. Коробков. — М. : Издательский центр «Академия», 2012. — 304 с. — (Сер. Бакалавриат).

ISBN 978-5-7695-8413-8

Учебное пособие создано в соответствии с Федеральным государственным образовательным стандартом по направлению подготовки 210700 «Инфокоммуникационные технологии и системы связи» (квалификации «бакалавр» и «магистр»).

Рассмотрены логические основы цифровой техники, методы синтеза комбинационных и последовательных узлов, узлы цифровых устройств, общие принципы построения компьютеров. Дано описание микропроцессоров общего назначения, сигнальных процессоров, микроконтроллеров. Изложены вопросы применения вычислительной техники в современных инфокоммуникационных технологиях. Приведены алгоритмы и структурные схемы устройств, реализующих задачи обработки информации и управления объектами в телекоммуникационных системах.

Для студентов учреждений высшего профессионального образования.

УДК 681.3(075.8)
ББК 32.81я73

ISBN 978-5-7695-8413-8

ПРЕДИСЛОВИЕ

Еще каких-то 20 лет назад мы не знали о сотовой телефонии, цифровом телевидении, Интернете и электронной почте. Мы даже не предполагали того, что мир кардинально изменится за этот малый по историческим меркам период времени. Пережив многие перемены в нашей стране, мы вошли в эпоху цифровой техники и информационных технологий. Согласно статистике проникновение сотовой связи вот уже несколько лет как переступило стопроцентный порог. Телефонная связь сейчас доступна практически в любом месте земного шара: там, где ее нет, можно воспользоваться спутниковой связью.

Развитие Интернета практически стерло границы между странами. Люди могут общаться, обмениваться информацией, и в этот процесс включается все больше и больше пользователей. Информационные технологии стали неотъемлемой частью любого предприятия или организации независимо от их рода деятельности и размера, активно внедряется электронный документооборот в государственных учреждениях. Развитие систем передачи информации позволяет реализовывать новые услуги: цифровое интерактивное телевидение, электронную коммерцию и др. Дальнейшее развитие информационных технологий и систем требует подготовки специалистов в области телекоммуникаций и радиоэлектронных систем.

На кафедре радиоэлектронных и телекоммуникационных систем (РТС) (ранее кафедра радиоуправления (РУ)) Казанского государственного технического университета (КГТУ) им. А. Н. Туполева с 1985 г. существует и успешно развивается направление по применению микропроцессорных систем обработки информации в радиоэлектронных и телекоммуникационных системах различного назначения. В то время микропроцессорные системы стали применяться в радиосистемах, причем в основном оборонного назначения. Кафедра одной из первых в институте начала обучение студентов основам микропроцессорной техники, а техническую базу для этого дали проводимые на кафедре хоздоговорные работы по разработке систем сбора и первичной обработки телеметрической информации с использованием микропроцессоров.

Одним из направлений работ на кафедре было создание систем приема и кодирования-декодирования сигналов радиотехнических систем различного назначения. На базе однокристальных и секционированных микропроцессоров были решены многие задачи по обработке радиотехнических сигналов.

Другим направлением работ является разработка систем обработки биоэлектрических сигналов и передачи медико-биологической информации по различным каналам связи. С развитием новых направлений обучения студентов активно прорабатываются вопросы применения микропроцессорных систем обработки информации для телекоммуникационных систем и систем подвижной радиосвязи.

Логическим продолжением применения микропроцессоров в разработках коллектива кафедры и преподавания основ микропроцессорной техники стало данное пособие. Оно включает в себя, по мнению авторов, наиболее важные и необходимые основы вычислительной техники и информационных технологий.

В первой главе описываются логические основы цифровой техники: алгебра логики, логические функции, системы счисления, правила выполнения арифметических операций, погрешности при их выполнении и др. Во второй главе представлено описание основных элементов цифровых устройств: дешифраторов, мультиплексоров, триггеров, регистров и др. В третьей главе описываются основные принципы построения и функционирования компьютеров и приводится достаточно полное описание микропроцессора общего назначения К580ВМ80. В четвертой главе рассмотрены отечественный сигнальный процессор КМ1813ВЕ1 и сигнальный процессор фирмы *Analog Device* серии ADSP-21XX. В пятой главе приводится описание микроконтроллера 1886ВЕ2У. В третьей, четвёртой и пятой главах рассмотрены примеры программ, реализующие некоторые функции обработки данных в телекоммуникационных системах. Шестая глава посвящена применению вычислительной техники в современных инфокоммуникационных технологиях. В ней рассматриваются принципы построения локальных вычислительных систем, принципы функционирования глобальной сети Интернет и электронной почты, компьютерной телефонии и видеоконференцсвязи.

ГЛАВА 1
ЛОГИЧЕСКИЕ ОСНОВЫ ЦИФРОВОЙ ТЕХНИКИ

Основой цифровой техники является представление любых, сколь угодно сложных, задач совокупностью логических операций над содержащимися в них данными. Набор таких операций и правила их выполнения задаются алгеброй логики или булевой алгеброй, названной по имени ее основоположника Д. Буля.

1.1. Основы алгебры логики, логические функции, логические элементы

В алгебре логики различные логические выражения (высказывания) могут иметь только два значения — «истинно» или «ложно». Для обозначения истинности или ложности высказываний пользуются символами 1 или 0.

В общем случае логические выражения являются функциями логических переменных x_1, x_2, ..., x_n, каждая из которых может иметь значения 0 или 1:

$$y = f(x_1, x_2, ..., x_n).$$

Если имеется n логических переменных, то они образуют $L = 2n$ возможных логических наборов из 0 и 1. При $n = 1$: $x = 0$ и $x = 1$; при $n = 2$: $x_1x_2 = 00/01/10/11$ и т. д. Для каждого набора переменных логическая функция y может принимать значение 0 или 1. Поэтому для n переменных можно образовать $N = 2^L$ различных логических функций. Таким образом, при $n = 2$ можно получить $N_2 = 16$ функций и далее при увеличении n число N_n растет чрезвычайно быстро: $N_3 = 256$, $N_4 = 65\,536$ и т. д.

Все возможные логические функции n переменных можно образовать с помощью трех основных операций: *логическое отрицание* (инверсия, операция НЕ), обозначаемое символом «–» над соответствующей переменной; *логическое сложение* (дизъюнкция, операция ИЛИ), обозначаемая символом «+» или «$\vee$», *логическое умножение* (конъюнкция, операция И), обозначаемая «·» или «$\wedge$». Для обозначения эквивалентности логических выражений используется знак равенства «=». Представления логических операций отрицания, сложения, умножения для переменных x_1, x_2 даны в табл. 1.1 и 1.2.

Таблица 1.1

x	$y = x$
0	1
1	0

Таблица 1.2

x_1	x_2	$y = x_1 \vee x_2$	$y = x_1 \wedge x_2$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Условные графические обозначения логических элементов, реализующих функции НЕ, ИЛИ, И, приведены на рис. 1.1. Элемент ИЛИ и элемент И изображены с двумя входами. На самом деле они могут иметь и больше входов, теоретически до бесконечности.

Рассмотрим логику работы каждого элемента.

Элемент НЕ, или инвертор. У него всегда один вход и один выход. Когда на входе у инвертора сигнал логического нуля, на выходе логическая единица. И наоборот, когда на входе логическая единица, на выходе логический нуль.

Элемент ИЛИ. На выходе этого элемента появится логическая единица тогда, когда хотя бы на одном из входов появится единица (или на первом, или на втором, или на любом другом из входов). Логический нуль на выходе будет только тогда, когда на всех входах будет сигнал логического нуля.

Элемент И. На выходе этого элемента сигнал логической единицы появляется только тогда, когда на всех входах будет присутствовать логическая единица (и на первом, и на втором, и на всех других входах). Если хотя бы на одном входе будет нуль, то и на выходе будет нуль.

Для рассмотренных логических операций справедливы аксиомы (тождества) и законы, основные из которых приведены далее.

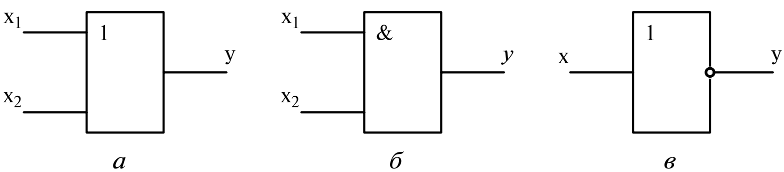


Рис. 1.1. Условные графические обозначения логических элементов:
а — ИЛИ (дизъюнктор); *б* — И (конъюнктор); *в* — НЕ (инвертор)

Аксиомы (*тождества*):

$$\begin{aligned} 1 \vee x = 1,\\ 0 \wedge x = 0; \end{aligned} \qquad (1.1)$$

$$\begin{aligned} 0 \vee x = x,\\ 1 \wedge x = x; \end{aligned} \qquad (1.2)$$

$$\begin{aligned} x \vee x = x,\\ x \wedge x = x; \end{aligned} \qquad (1.3)$$

$$\begin{aligned} x \vee \bar{x} = 1,\\ x \wedge \bar{x} = 0; \end{aligned} \qquad (1.4)$$

$$\bar{\bar{x}} = x. \qquad (1.5)$$

Законы коммутативности (*переместительности*):

$$\begin{aligned} x_1 \wedge x_2 = x_2 \wedge x_1;\\ x_1 \vee x_2 = x_2 \vee x_1. \end{aligned} \qquad (1.6)$$

Законы ассоциативности (*сочетательности*):

$$\begin{aligned} x_1 \wedge (x_2 \wedge x_3) = (x_1 \wedge x_2) \wedge x_3;\\ x_1 \vee (x_2 \vee x_3) = (x_1 \vee x_2) \vee x_3. \end{aligned} \qquad (1.7)$$

Законы дистрибутивности (*распределительности*)

$$\begin{aligned} x_1 \wedge (x_2 \wedge x_3) = (x_1 \wedge x_2) x_3;\\ x_1 \vee (x_2 \wedge x_3) = (x_1 \vee x_2) \wedge (x_1 \vee x_3). \end{aligned} \qquad (1.8)$$

Законы дуальности (*теоремы де Моргана*):

$$\begin{aligned} \overline{x_1 \vee x_2} = \bar{x}_1 \wedge x_2;\\ \overline{x_1 \wedge x_2} = \bar{x}_1 \vee \bar{x}_2. \end{aligned} \qquad (1.9)$$

Законы поглощения:

$$\begin{aligned} x_1 \vee x_1 \wedge x_2 = x_1;\\ x_1 \wedge (x_1 \vee x_2) = x_1. \end{aligned} \qquad (1.10)$$

Используя данные тождества и законы, можно получать новые логические выражения, а также доказывать справедливость тех или иных законов на основании других. Например, с помощью второго закона дистрибутивности (1.8), тождества (1.1) и тождества (1.4) получаем следующее соотношение:

$$x_1 \vee \left(\bar{x}_1 \wedge x_2\right) = \left(x_1 \vee \bar{x}_1\right) \wedge \left(x_1 \vee x_2\right) = x_1 \vee x_2. \qquad (1.11)$$

Используя первый закон дистрибутивности (1.8), тождества (1.1), (1.3) и закон ассоциативности (1.7), получаем доказательство справедливости закона поглощения (1.10):

$$x_1 \wedge (x_1 \vee x_2) = x_1 \wedge x_1 \vee x_1 \wedge x_2 = x_1 \vee x_1 \wedge x_2 = x_1 \wedge (1 + x_2) = x_1. \quad (1.12)$$

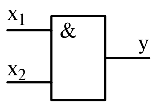
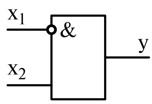
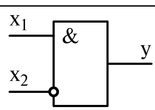
Используя законы ассоциативности, любую логическую функцию многих переменных ($n > 2$) можно представить в виде комбинаций функций двух переменных. Полный набор 16-логических функций для двух переменных приведен в табл. 1.3.

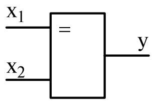
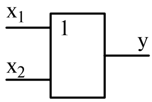
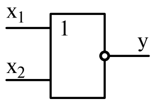
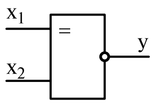
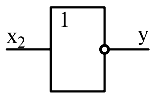
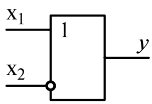
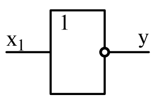
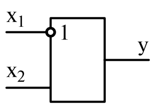
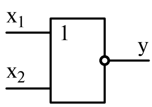
Логическое сложение ИЛИ — это функция, значение которой равно 0 только тогда, когда все аргументы равны 0. Во всех остальных ситуациях хотя бы один аргумент равен 1, функция равна 1. Инверсия логического сложения называется функцией Пирса (ИЛИ-НЕ), а также стрелкой Пирса.

Логическое умножение И — это функция, равная 1, если все аргументы равны 1. Во всех остальных ситуациях функция равна 0. Инверсию логического умножения называют функцией Шеффера (И-НЕ), но чаще штрихом Шеффера.

Логическая равнозначность — это функция, равная 1, если значения обоих аргументов одинаковы. Функция равна 0, если значения аргументов различны. Инверсия логической равнозначности — функция «Исключающее ИЛИ» (логическая неравнозначность). Эта функция соответствует сложению в двоичной системе счисления.

Таблица 1.3

Переменная					Алгебраическое выражение	Название функции	Логический элемент
x_1	0	0	1	1			
x_2	0	1	0	1			
y_0	0	0	0	0	$y_0 = 0$	Постоянный нуль	—
y_1	0	0	0	1	$y_1 = x_1 \wedge x_2$	Конъюнкция	
y_2	0	0	1	0	$y_2 = x_1 \wedge \bar{x}_2$	Запрет $\left(\overline{x_1 \to x_2}\right)$	
y_3	0	0	1	1	$y_3 = x_1$	Тождественность x_1	—
y_4	0	1	0	0	$y_4 = \bar{x}_1 \wedge x_2$	Запрет $\left(\overline{x_1 \leftarrow x_2}\right)$	

Переменная					Алгебраическое выражение	Название функции	Логический элемент
y_5	0	1	0	1	$y_5 = x_2$	Тождественность	—
y_6	0	1	1	0	$y_6 = (x_1 \wedge \bar{x}_2) \vee (\bar{x}_1 \wedge x_2)$	Исключающее ИЛИ $(x_1 \forall x_2) = (x_1 \oplus x_2)$	
y_7	0	1	1	1	$y_7 = x_1 \vee x_2$	Дизъюнкция	
y_8	1	0	0	0	$y_8 = \overline{x_1 \vee x_2}$	Стрелка Пирса $(x_1 \downarrow x_2)$ (ИЛИ-НЕ)	
y_9	1	0	0	1	$y_9 = (x_1 \wedge x_2) \vee (\bar{x}_1 \vee \bar{x}_2)$	Равнозначность $(x_1 = x_2)$ (эквивалентность)	
y_{10}	1	0	1	0	$y_{10} = \bar{x}_2$	Инверсия x_2	
y_{11}	1	0	1	1	$y_{11} = x_1 \vee \bar{x}_2$	Импликация от x_2 к x_1	
y_{12}	1	1	0	0	$y_{12} = \bar{x}_1$	Инверсия x_1	
y_{13}	1	1	0	1	$y_{13} = \bar{x}_1 \vee x_2$	Импликация от x_1 к x_2	
y_{14}	1	1	1	0	$y_{14} = \overline{x_1 \wedge x_2}$	Штрих Шеффера $(x_1 \vert x_2)$ (И-НЕ)	
y_{15}	1	1	1	1	$y_3 = 1$	Постоянная единица	—

Импликация — это функция, образуемая по схеме «если x_1, то x_2». Если x_1 ложно (0), то импликация истинна ($y = 1$) при произвольном x_2. Если $x_1 = 1$, то $y = 1$ только при $x_2 = 1$. Если $x_1 = 1$, а $x_2 = 0$, то $y = 0$. Аналогичная функция получается при замене аргументов местами. Инверсия импликации дает логическую функцию — запрет.

Логические функции сложения, умножения, эквивалентности и их инверсии симметричны. Значения функций не меняются при перестановке переменных. Остальные функции — импликация и запрет — несимметричны.

Число возможных функций при увеличении числа аргументов растет очень быстро. Практически логические функции, зависящие от большого числа аргументов, выражаются суперпозициями функций, зависящих от меньшего числа аргументов. При суперпозиции функций одни функции подставляются вместо аргументов других. Например, суперпозиция функций двух аргументов является функцией четырех аргументов:

$$y = f_1[f_2(x_1, x_2), f_3(x_3, x_4)]. \tag{1.13}$$

Суперпозиция функций — это представление сложной функции структурной формулой, в которую входят логические аргументы, связанные между собой простыми функциями. Набор простейших функций, через которые можно выразить любые другие логические функции, называют функционально полным или *логическим базисом*. Одним из распространенных логических базисов является набор И, ИЛИ, НЕ.

Логический базис минимальный, если удаление хотя бы одной функции превращает этот базис в неполный. Базис И, ИЛИ, НЕ не является минимальным, так как И либо ИЛИ можно выразить через две остальные функции. Так, функция И заменяется суперпозицией функций НЕ, ИЛИ, НЕ, а функция ИЛИ — функциями НЕ, И, НЕ. Минимальными базисами являются И, НЕ и ИЛИ, НЕ. Имеются минимальные логические базисы, содержащие только одну функцию: функцию Шеффера И-НЕ или функцию Пирса ИЛИ-НЕ.

Электронные цепи, выполняющие простейшие логические функции, называют логическими элементами. Реализация любой логической функции возможна, если имеются логические элементы, реализующие функции одного из минимальных базисов. Чаще всего используют элементы И—НЕ либо ИЛИ—НЕ. Наряду с элементами минимального базиса используются и другие элементы И—ИЛИ—НЕ, И, ИЛИ и др., применение которых позволяет улучшить технические характеристики устройств.

По способу функционирования логические устройства (и их схемы) делятся на два класса: комбинационные и последовательностные.

В *комбинационном устройстве* (называемом также *автоматом без памяти*) каждый символ на выходе (Лог. 0 или Лог. 1) определяется лишь символами на выходе (Лог. 0 или Лог. 1), действующими в данный момент времени на входах устройства, и не зависит от того, какие ранее действовали на этих входах. В этом смысле комбинационные устройства лишены памяти (они не хранят сведений о прошлом работы устройства).

В *последовательностных устройствах* (или *автоматах с памятью*) выходной сигнал определяется не только набором символов, действующих на входах в данный момент времени, но и внутренним состоянием устройства, а последнее зависит от того, какие наборы символов действовали на входах во все предшествующие моменты времени в процессе работы устройства. Поэтому можно говорить, что последовательностные устройства обладают памятью (они хранят сведения о прошлом работы устройства).

1.2. Методы синтеза комбинационных схем

Синтез логического устройства распадается на несколько этапов. На первом этапе функцию, заданную в словесной, табличной или других формах, требуется представить в виде логического выражения с использованием некоторого базиса. Дальнейшие этапы сводятся к получению минимальных форм функций, обеспечивающих при синтезе наименьшее количество электронного оборудования и рациональное построение функциональной схемы устройства. Для первого этапа обычно используется базис И, ИЛИ, НЕ независимо от базиса, который будет использован для построения логического устройства.

Сложные логические функции чаще всего задаются таблицей значений (истинности). Структурные формулы по таблице можно составлять двумя способами.

В ходе применения **первого способа** (по единицам) наборы аргументов, при которых функция равна 1, записывают в виде логических произведений аргументов и их инверсий. В произведении пишут инверсии тех аргументов, значения которых равны 0. Полученные логические произведения объединяют функцией логического сложения ИЛИ. Такую структурную формулу называют *дизъюнктивной нормальной формой* (ДНФ).

Примером ДНФ может служить выражение

$$y(x_1,x_2,x_3)=x_1\vee(x_2\wedge\bar{x}_3)\vee(\bar{x}_1\wedge\bar{x}_2\wedge\bar{x}_3)\vee(x_2\wedge x_3). \qquad (1.14)$$

Второй способ (по нулям) предусматривает, что наборы аргументов, при которых функция равна 0, суммируются в прямом виде логических сумм. Те аргументы, которые равны 0, суммируются в прямом виде, а аргументы, равные 1, суммируются инвертированными. Полученные суммы объединяют функцией логического умножения И. Такую структурную формулу называют *конъюнктивной нормальной формой* (КНФ).

Примером КНФ может служить следующее выражение:

$$y(x_1, x_2, x_3) = x_1 \wedge (x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (x_2 \vee x). \qquad (1.15)$$

Если в каждом члене ДНФ представлены все аргументы функции или их инверсии, то такая форма называется *совершенной дизъюнктивной формой* (СДНФ). Выражение (1.14) не является СДНФ, так как в нем лишь третий член содержит все аргументы.

Для перехода от ДНФ к СДНФ необходимо в каждый из членов, в которых представлены не все аргументы, ввести выражение вида $x_i \vee \bar{x}_i$, где x_i — отсутствующий в члене аргумент. Так как $x_i \vee \bar{x}_i = 1$, то такая операция не может изменить значение функции. Каждая логическая функция имеет единственную СДНФ.

В *совершенной конъюнктивной форме* (СКНФ) в каждом члене КНФ должны быть представлены все аргументы. Для перехода от КНФ к СКНФ необходимо добавить каждому члену, не содержащему всех аргументов, члены вида $x_i \wedge \bar{x}_i$, где x_i — аргумент, не представленный в члене. Так как $x_i \wedge \bar{x}_i = 0$, то такая операция не может повлиять на значение функции. Каждая логическая функция имеет единственную СКНФ.

Для примера запишем логические функции эквивалентности, импликации и запрета.

Совершенные дизъюнктивные формы эквивалентности, импликации и запрета соответственно:

$$y = x_1 \sim x_2 = (\bar{x}_1 \wedge \bar{x}_2) \vee (x_1 \wedge x_2);$$
$$y = x_1 \to x_2 = (\bar{x}_1 \wedge \bar{x}_2) \vee (\bar{x}_1 \wedge x_2) \vee (x_1 \wedge x_2);$$
$$y = x_1 \overline{\to} x_2 = x_1 \wedge \bar{x}_2.$$

Совершенные конъюнктивные формы эквивалентности, импликации и запрета соответственно:

$$y = x_1 \sim x_2 = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2);$$
$$y = x_1 \to x_2 = \bar{x}_1 \vee x_2;$$
$$y = x_1 \overline{\to} x_2 = (x_1 \vee \bar{x}_2) \wedge (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_2).$$

Приведенные примеры наглядно показывают, что в зависимости от способа составления получаются различные структурные формулы. Перед реализацией полученные структурные формулы подвергаются минимизации, цель которой — минимизация числа логических функций и, следовательно, минимизация числа логических элементов. При минимизации структурных формул пользуются основными законами алгебры Буля: переместительности (коммутативности), сочетательности (ассоциативности) и распределительности (дистрибутивности), которые совпадают с законами классической алгебры. Кроме того, пользуются законами, справедливыми только для алгебры логики: законами тождественности и законами Моргана.

Рассмотрим пример представления логической функции трех аргументов.

Аргументы: x_1— 000111, x_2 — 0011011, x_3 — 0101001. Функция y — 0001011.

Совершенная дизъюнктивная форма этой функции имеет вид

$$y = (\bar{x}_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \bar{x}_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge \bar{x}_3) \vee (x_1 \wedge x_2 \wedge x_3), \quad (1.16)$$

а СКНФ

$$y = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee x_3). \quad (1.17)$$

Для минимизации функции (1.16) воспользуемся законом тождественности и к структурной формуле добавим еще одно произведение, уже имеющееся в выражении (1.16):

$$y = (\bar{x}_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \bar{x}_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge \bar{x}_3) \vee (x_1 \wedge x_2 \wedge x_3) \vee$$
$$\vee (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_3).$$

Сгруппируем слагаемые в соответствии с законом дистрибутивности:

$$y = [x_2 \wedge x_3 \wedge (\bar{x}_1 \vee x_1)] \vee [x_1 \wedge x_2 \wedge (\bar{x}_2 \vee x_2)] \vee [x_1 \wedge x_2 \wedge (\bar{x}_3 \vee x_3)].$$

Учитывая, что $\bar{x}_i \wedge x_i = 1$, получаем следующее выражение:

$$y = (x_2 \wedge x_3) \vee (x_1 \wedge x_3) \vee (x_1 \wedge x_2). \quad (1.18)$$

Структурная формула (1.17) минимизируется аналогично. Сначала ее умножают на $(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$. Затем, в соответствии с законом распределительности имеем выражение

$$y = [(x_1 \vee x_2) \wedge (x_1 \vee x_2) \vee (x_1 \vee x_2) \wedge (x_3 \vee \bar{x}_3) \vee x_3 \wedge \bar{x}_3] \wedge [(x_1 \vee x_3) \wedge$$
$$\wedge (x_1 \vee x_3) \vee (x_1 \vee x_3) \wedge (x_2 \vee \bar{x}_2) \vee (x_2 \wedge x_2)] \wedge [(x_2 \vee x_3) \wedge (x_2 \vee x_3) \vee$$
$$\vee (x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_1) \vee (x_1 \wedge \bar{x}_1),$$

которое с учетом формул (1.3) и (1.4) принимает следующий вид:

$$y = (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_2 \vee x_3). \quad (1.19)$$

Структурные формулы (1.18) и (1.19) эквивалентны. Схема, реализующая функцию (1.18), показана на рис. 1.2, *а*, а реализующая функцию (1.19) — на рис. 1.2, *б*.

Обе схемы содержат одинаковое число логических элементов.

Когда число аргументов функции больше трех, минимизация логических функций достаточно сложна и выполняется с помощью специальных методов.

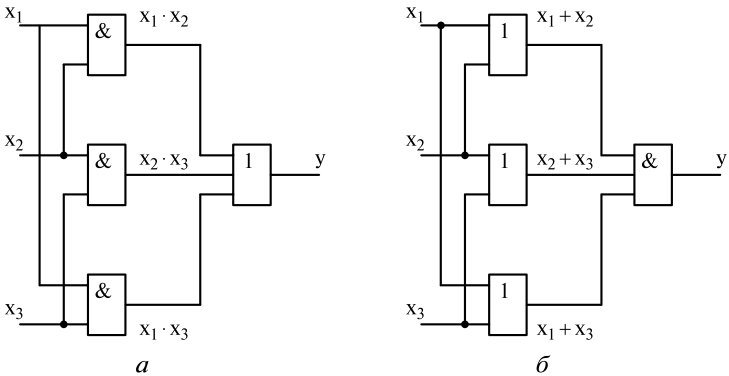


Рис. 1.2. Комбинационная схема:

а — реализующая функцию (1.18); *б* — реализующая функцию (1.19)

Минимизация логических функций может проводиться с помощью следующих методов: метод Квайна, метод Квайна — Мак-Класки, метод Патрика.

Каждый из методов имеет свои достоинства и недостатки. Минимизация функций алгебраическими методами требует известного навыка. Метод Квайна может быть применен для минимизации функций с использованием ЭВМ в тех случаях, когда минимизируемая функция достаточно сложна. Однако для минимизации функции ручным способом (без ЭВМ) этот метод оказывается весьма трудоемким.

При относительно небольшом числе переменных (не более шести) весьма удобным и наглядным является графическое представление логических функций в виде карт Вейча и карт Карно.

Минимизация логических функций методом карт Вейча. Метод минимизации с помощью карт Вейча обеспечивает простоту получения результатов. Он используется при минимизации относительно несложных функций ручным способом. Карта Вейча представляет собой определенную форму таблицы истинности. Карты Вейча для двух, трех и четырех аргументов соответственно показаны в табл. 1.4 — 1.6.

Каждая клетка карты соответствует некоторому набору значений аргументов.

Этот набор аргументов определяется присвоением значения Лог. 1 буквам, на пересечении строк и столбцов которых расположена клетка. Так, в карте функций четырех аргументов (см. табл. 1.6) клетки первой строки соответствуют следующим комбинациям значений аргументов:

- 1-я клетка: $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 0$;
- 2-я клетка: $x_1 = 1$, $x_2 = 1$, $x_3 = 1$, $x_4 = 0$;
- 3-я клетка: $x_1 = 1$, $x_2 = 0$, $x_3 = 1$, $x_4 = 0$;
- 4-я клетка: $x_1 = 1$, $x_2 = 0$, $x_3 = 0$, $x_4 = 0$.

Таблица 1.4

	x_2	$\bar{x}_2$
x_1		
$\bar{x}_1$		

Таблица 1.5

	x_2		$\bar{x}_2$	
x_1				
$\bar{x}_1$				
	$\bar{x}_3$	x^3		$\bar{x}_3$

Таблица 1.6

	x_2		$\bar{x}_2$		
x_1					$\bar{x}_4$
					x_4
$\bar{x}_1$					
					$\bar{x}_4$
	$\bar{x}_3$	x^3		$\bar{x}_3$	

Число клеток карты равно числу всех возможных наборов значений аргументов 2^n (n — число аргументов функций). В каждую из клеток карты записывается значение функции на соответствующем этой клетке наборе значений аргументов. Пусть функция задана таблицей истинности в форме, приведенной в табл. 1.7. Таблица истинности этой функции в форме карты Вейча представлена табл. 1.8.

Как видим, карта Вейча определяет значения функций на всех возможных наборах значений аргументов и является таблицей истинности. Карты Вейча компактны, но главное их достоинство состоит в следующем. При любом переходе из одной клетки в соседнюю вдоль столбца или строки изменяется значение лишь одного аргумента функции. Следовательно, если в паре соседних клеток содержится единица, то над соответствующими им членами может быть проведена операция склеивания. Таким образом, облегчается поиск склеиваемых членов.

Сформулируем правила получения минимальных ДНФ (МДНФ)-функций с помощью карт Вейча.

Все клетки, содержащие единицы, объединяются в замкнутые области. При этом каждая область должна представлять собой прямоугольник с числом клеток 2^k, где $k = 0, 1, 2, \ldots$ Значит, допустимое число клеток в области 1, 2, 4, 8, ... Области могут пересекаться и

Таблица 1.7

Показатель	Значение							
x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
$f(x_1, x_2, x_3)$	0	1	0	1	0	0	1	1

Таблица 1.8

	x_2	x_2	$\bar{x}_2$	$\bar{x}_2$
x_1	1	1	0	0
$\bar{x}_1$	0	1	1	0
	$\bar{x}_3$	x^3	x^3	$\bar{x}_3$

одни и те же клетки могут входить в разные области. Затем производится запись выражения МДНФ функции. Каждая из областей в МДНФ представляется членом, число букв в котором на k меньше общего числа аргументов функции n (т. е. равно $n - k$). Каждый член МДНФ составляется лишь из тех аргументов, которые для клеток соответствующей области имеют одинаковое значение (без инверсии либо с инверсией).

Таким образом, при охвате клеток замкнутыми областями следует стремиться, чтобы число областей было минимальным (при этом минимальным будет число членов в МДНФ функции), а каждая область содержала возможно большее число клеток (при этом минимальным будет число букв в членах МДНФ функции).

Рассмотрим минимизацию с помощью карты Вейча функции трех аргументов, представленной на табл. 1.9.

Таблица 1.9

I	x_2	x_2	$\bar{x}_2$	$\bar{x}_2$
x_1	1	1	0	0
$\bar{x}_1$	0	1	1	0
	$\bar{x}_3$	x^3	x^3 II	$\bar{x}_3$