Министерство путей сообщения РФ

Департамент кадров и учебных заведений

САМАРСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ ПУТЕЙ СООБЩЕНИЯ

Кафедра «Информатика»

«Булева алгебра и логические элементы»

Методические указания

по дисциплине «Дискретная математика» для студентов заочной формы обучения специальностей 230201 «Информационные системы и технологии» и 230102 «Автоматизированные системы обработки информации и управления»

Составители: Никищенков С.А.

Смышляев В.А.

Припутников А.П.

Самара 2004

УДК 519.1

Булева алгебра и логические элементы: Методические указания по дисциплине «Дискретная математика» для студентов заочной формы обучения специальностей 230201 «Информационные системы и технологии» и 230102 «Автоматизированные системы обработки информации и управления» / Составители: Никищенков С.А., Смышляев В.А., Припутников А.П. – Самара: СамГАПС, 2004. – 20 с.

Утверждено на заседании кафедры информатики 03.10.2003 г., протокол №8.
Печатается по решению редакционно-издательского совета академии.

Данные методические указания предназначены для изучения теоретического материала по дисциплине «Дискретная математика» для студентов специальностей «Информационные системы в технике и технологиях» и «Автоматизированные системы обработки информации и управления». Рассмотрены основы булевой алгебры и логические элементы, на которых возможна реализация булевых функций от нескольких переменных. Показана минимизация логических функций. Приведены примеры решения задач синтеза и анализа функций и логических схем. Рассматриваются примеры комбинационных схем и ошибки при их проектировании.

Составители: Никищенков Сергей Алексеевич
Смышляев Валерий Анатольевич
Припутников Алексей Петрович

Рецензенты: д.ф.-м.н., профессор Сараев Леонид Александрович, зав. кафедрой «Высшая математика и информатика» СамГУ,
к.т.н., профессор Гуменников Валерий Борисович, зав. кафедрой «Автоматика, телемеханика и связь на ж.д. транспорте» СамГАПС.

Редактор: И.М. Егорова
Компьютерная верстка: Н.В. Чертыковцева
Подписано в печать 07.05.04 Формат 60×90 1/16
Бумага писчая. Печать оперативная. Усл. п.л. 1,2
Тираж 100 экз. Заказ №69

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Данные методические указания являются частью курса лекций по дисциплине «Дискретная математика», читаемым студентам 1 курса СамГАПС специальностей «Информационные системы и технологии» и «Автоматизированные системы обработки информации и управления».

Цифровые устройства в настоящее время широко используются во всех отраслях: связь, телевидение, управление промышленным и бытовым оборудованием, медицина, системы защиты и сигнализации и т.п. Характерной особенностью является использование цифровой техники специалистами различного профиля.

Теоретической базой цифровой техники являются алгебра логики, двоичная арифметика и теория конечных автоматов. Основные функциональные узлы, разработанные на основе этой базы, представлены широкой номенклатурой изделий микроэлектронной техники от простейшего вентиля до микропроцессора. Все эти узлы универсальны и многофункциональны, что позволяет использовать их по разному назначению.

Булева алгебра, созданная в середине 18 века Дж. Булем, оперирует с логическими переменными. Основополагающим законом булевой алгебры является закон исключения третьего, согласно которому логические переменные, в отличие от переменных обычной алгебры, могут принимать только два значения {«да», «нет»}, {«истинно», «ложно»} и т.д. Переменные обычно обозначаются, как и двоичные цифры, символами 0 и 1.

Методические указания имеют следующие цели:

1) изучение символики и математического языка описания булевых функций,
2) ознакомление студентов с основами булевой алгебры,
3) изучение основных логических элементов,
4) решение задач анализа и синтеза логических функций и схем.

1. БУЛЕВА АЛГЕБРА

Булева алгебра – алгебра, образованная множеством B={0, 1} вместе со всеми возможными логическими операциями на нём.

Булева (*логическая*) *функция* – это функция, принимающая значения 0 или 1 в результате логических операций над логическими переменными. Операции над переменными записываются с помощью символов: &, $\vee$, –, $\oplus$, $\rightarrow$ и т.д.

Булева функция может быть задана:

1) словесным описанием (назначением, определением),
2) таблицей истинности,
3) формулой, состоящей из букв, знаков логических операций и скобок,
4) комбинационной схемой, составленной из логических элементов,
5) координатным способом (картой Карно),
6) переключательной схемой,
7) диаграммой Венна,
8) геометрическим способом (гиперкубами),
9) диаграммой двоичного решения и т.д.

1.1. Таблица истинности

Любая логическая функция нескольких переменных однозначно задается в виде *таблицы истинности*, в левой части которой выписаны все возможные наборы значений её аргументов $x_1, \ldots, x_n$, а правая часть представляет собой столбец значений функций, соответствующих этим наборам. Набор значений переменных, на котором функция принимает значение $f = 1$, называется *единичным набором функции f*; множество всех единичных наборов – *единичным множеством функции f*. Аналогично набор значений, на котором $f = 0$, называется *нулевым набором функции f*, а множество нулевых наборов – *нулевым множеством*. В общем случае таблица истинности для функции от n переменных должна иметь 2^n строк.

Пример. Составить таблицу истинности для функции f трех переменных x_1, x_2, x_3, которая равна единице в случае, если только одна из входных переменных равна 1.

Таблица 1

Входные переменные			Выходная переменная
x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Переменная x в функции f называется *фиктивной* (несущественной), если значение переменной x не влияет на значение булевой функции. Фиктивные переменные могут быть удалены или введены в набор переменных функций. Количество булевых функций определяется числом n переменных $|P_2(n)| = 2^{2^n}$.

1.2. Булевы функции одной переменной

Множество всех логических функций одной переменной – *унарные логические операции* – представлено в таблице 2. Число функций $|P_2(1)| = 2^{2^1} = 4$.

Таблица 2

x	φ_0	φ_1	φ_2	φ_3
0	0	0	1	1
1	0	1	0	1
Обозначения	0	x	$\overline{x}$	1

В таблице 2 представлены следующие функции:

φ_0, φ_3 – функции констант 0 и 1, значения которых не зависят от переменной (x – фиктивная переменная);

$\varphi_1(x)=x$ – повторяет значение x;

$\varphi_2(x)=\overline{x}$ – инверсия (отрицание переменной, функция НЕ).

1.3. Булевы функции двух переменных

Множество всех логических функций двух переменных – *бинарные логические операции* – представлено в таблице 3. Число функций $|P_2(2)| = 2^{2^2} = 16$.

Таблица 3

x_1	x_2	ψ_0	ψ_1	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6	ψ_7	ψ_8	ψ_9	ψ_{10}	ψ_{11}	ψ_{12}	ψ_{13}	ψ_{14}	ψ_{15}
0	0	0	0	0	1	1	0	1	0	1	0	1	1	0	1	0	1
0	1	0	0	1	1	0	1	0	0	1	1	0	1	0	0	1	1
1	0	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1
1	1	0	1	1	0	0	1	0	1	0	0	1	1	0	1	0	1
Обозначения		0	x_1	x_2	$\overline{x_1}$	$\overline{x_2}$	$\vee$	$\downarrow$	&	$\mid$	$\oplus$	$\equiv$	$\rightarrow$	$\overline{\rightarrow}$	$\leftarrow$	$\overline{\leftarrow}$	1

В таблице 3 представлены следующие функции:

ψ_0, ψ_{15} – функции констант 0 и 1;

ψ_1 – повторение x_1 (x_2 – фиктивная переменная);

ψ_2 – повторение x_2 (x_1 – фиктивная переменная);

ψ_3 – $\overline{x_1}$ (x_2 – фиктивная переменная);

ψ_4 – $\overline{x_2}$ (x_1 – фиктивная переменная);

ψ_5 – x_1+x_2; $x_1 \vee x_2$ – дизъюнкция (логическая функция ИЛИ);

ψ_6 – $\overline{x_1 + x_2}$; $x_1 \downarrow x_2$ – стрелка Пирса (логическая функция ИЛИ-НЕ);

ψ_7 – $x_1 \& x_2$; $x_1 \wedge x_2$; $x_1 \cdot x_2$ – конъюнкция (логическая функция И);

ψ_8 – $\overline{x_1 \cdot x_2}$; $x_1 \mid x_2$ – штрих Шеффера (логическая функция И-НЕ);

ψ_9 – $x_1 \oplus x_2$; $x_1 \not\equiv x_2$ – сложение по модулю 2 (функция неравнозначности);

ψ_{10} – $\overline{x_1 \oplus x_2}$; $x_1 \sim x_2$; $x_1 \equiv x_2$ – эквивалентность, равнозначность, отрицание сложения по модулю 2;

ψ_{11} – $x_1 \rightarrow x_2$; $x_1 \supset x_2$ – импликация (x_1 влечёт за собой x_2);

ψ_{12} – $\overline{x_1 \rightarrow x_2}$ – отрицание импликант (функция запрета);

ψ_{13} – $x_2 \rightarrow x_1$; $x_2 \supset x_1$ – импликация (x_2 влечёт за собой x_1);

ψ_{14} – $\overline{x_2 \rightarrow x_1}$ – отрицание импликант (функция запрета).

Логические функции трех и более переменных обычно задаются (наряду с таблицами истинности) также формулами, состоящими из символов переменных и знаков унарных и бинарных операций. Значение любой логической формулы, содержащей знаки логических операций, можно вычислить для любого набора значений переменных, используя таблицы 2 и 3.

Пример. Составить таблицу истинности функции трех переменных, заданной формулой: $f(x_1, x_2, x_3) = (\overline{x_1} \vee x_2) \rightarrow (x_1 \& x_3)$.

Для построения таблицы истинности f вычислим ее значения на каждом из восьми наборов значений (табл. 4).

Таблица 4

x_1	x_2	x_3	$\overline{x_1}$	$\overline{x_1} \vee x_2$	$x_1\&x_3$	$(\overline{x_1} \vee x_2)\to(x_1\&x_3)$
0	0	0	1	1	0	0
0	0	1	1	1	0	0
0	1	0	1	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	1
1	1	0	0	1	0	0
1	1	1	0	1	1	1

1.4. Законы и теоремы булевой алгебры

Формулы, представляющие одну и ту же функцию называются *эквивалентными* или *равносильными* (обозначаются =).

Основные законы булевой алгебры.

1. Ассоциативность конъюнкции и дизъюнкции (сочетательный закон):

а) $x_1\cdot(x_2\cdot x_3)=(x_1\cdot x_2)\cdot x_3=x_1\cdot x_2\cdot x_3$, б) $x_1\vee(x_2\vee x_3)=(x_1\vee x_2)\vee x_3=x_1\vee x_2\vee x_3$.

2. Коммутативность конъюнкции и дизъюнкции (переместительный закон):

а) $x_1\cdot x_2=x_2\cdot x_1$, б) $x_1\vee x_2=x_2\vee x_1$.

3. Дистрибутивность (распределительный закон):

а) $x_1\cdot(x_2\vee x_3)=x_1\cdot x_2\vee x_1\cdot x_3$, б) $x_1\vee(x_2\cdot x_3)=(x_1\vee x_2)\cdot(x_1\vee x_3)$.

4. Идемпотентность (правило повторения):

а) $x\cdot x=x$, б) $x\vee x=x$.

5. Закон двойного отрицания: $\overline{\overline{x}} = x$.

6. Свойства констант 0 и 1:

а) $x\cdot 1=x$, б) $x\cdot 0=0$, в) $x\vee 1=1$,

г) $x\vee 0=x$, д) $\overline{0}=1$, е) $\overline{1}=0$.

7. Теорема двойственности (правила де Моргана):

а) $\overline{x_1 \cdot x_2} = \overline{x_1} \vee \overline{x_2}$, б) $\overline{x_1 \vee x_2} = \overline{x_1} \cdot \overline{x_2}$.

8. Закон противоречия: $x\cdot \overline{x}=0$.

9. Закон исключённого третьего: $x\vee \overline{x}=1$.

Все эти равенства остаются справедливыми при подстановке вместо переменных любых логических функций и, следовательно, любых формул, представляющих эти функции. Наряду с основными соотношениями для упрощения формул часто используются следующие правила:

1. Правила поглощения:

а) $x_1\vee x_1\cdot x_2=x_1$, б) $x_1\cdot(x_1\vee x_2)=x_1$.

2. Правила склеивания:

а) $x_1\cdot x_2\vee x_1\cdot \overline{x_2}=x_1$ б) $x_1\vee \overline{x_1}\cdot x_2=x_1\vee x_2$.

3. Правило обобщенного склеивания:

$x_1\cdot x_3\vee x_2\cdot \overline{x_3}\vee x_1\cdot x_2=x_1\cdot x_3\vee x_2\cdot \overline{x_3}$.

Пример. Упростить булевы формулы:

а) $f(x_1, x_2, x_3)=x_1\vee x_1x_3\vee \overline{x_1}\, x_2x_3\vee x_2\overline{x_3}=x_1\vee \overline{x_1}\, x_2x_3\vee x_2\overline{x_3} = x_1\vee x_2x_3\vee x_2\overline{x_3} =$

$= x_1 \vee x_2(x_3 \vee \overline{x_3}) = x_1 \vee x_2 \cdot 1 = x_1 \vee x_2$;

б) $f(x, y, z) = x(\overline{y} \vee z)(\overline{x} \vee y \vee z) = x\overline{x}\,\overline{y} \vee x\overline{x}z \vee xy\overline{y} \vee xyz \vee x\overline{y}z \vee xzz =$
$= 0 \cdot \overline{y} \vee 0 \cdot z \vee x \cdot 0 \vee xyz \vee x\overline{y}z \vee xzz = xyz \vee x\overline{y}z \vee xzz = xyz \vee x\overline{y}z \vee xz = xz$.

2. ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ

Логический элемент в электронных схемах – это устройство, реализующее ту или иную логическую функцию. При этом логические сигналы 0 и 1 задаются разными уровнями напряжения. Сигнал логического нуля обычно представляется низким уровнем напряжения U^0, логической единицы – высоким U^1. Такая логика получила название положительной. В ряде случаев используют отрицательную логику, где логический нуль представляется высоким уровнем напряжения, а логическая единица – низким.

Логические схемы состоят из логических элементов, осуществляющих логические операции. Для изображения логических схем всегда используются условные графические обозначения элементов, описывающие только выполняемую элементами функцию и не зависящие от его схемы. В настоящее время в мире существует несколько общепринятых стандартов условных обозначений. Наиболее распространенными являются американский стандарт milspec 806B и стандарт МЭК 117-15 A, созданный Международной Электротехнической Комиссией. Часто в литературе используются также обозначения в европейской системе DIN 4070. В отечественной литературе условные обозначения элементов в основном соответствуют ГОСТ 2.743-82.

2.1. Основные логические операции и логические элементы

Конъюнкция (логическое умножение, операция И, AND): функция f принимает единичное значение только тогда, когда равны единице абсолютно все входные переменные. Следовательно, функция f принимает значение «0» только в том случае, если хотя бы один из входных переменных будет равен «0». В частности, для двух переменных x_1 и x_2 существует четыре различных сочетания, но только одному из них ($x_1=x_2=1$) соответствует единичное значение функции (табл. 5).

Таблица 5

x_1	x_2	f
0	0	0
0	1	0
1	0	0
1	1	1

Условные графические обозначения элементов логического умножения по различным стандартам показаны на рис. 1.

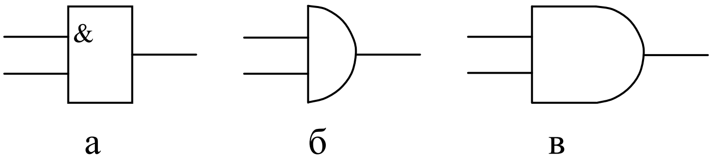


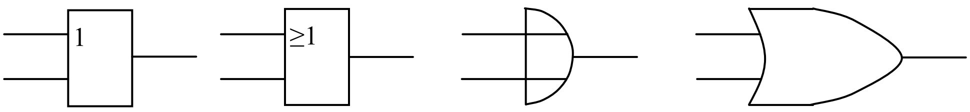
Рис. 1. Условные обозначения логического элемента И:

а) по ГОСТ и стандарту МЭК, б) по стандарту DIN, в) по стандарту milspec

Дизъюнкция (логическое сложение, операция ИЛИ, OR): функция f принимает единичное значение, если единице равна хотя бы одна из входных переменных (табл. 6). Следовательно, функция f принимает нулевое значение только в том случае, если все входные переменные будут равны «0».

Таблица 6

x_1	x_2	f
0	0	0
0	1	1
1	0	1
1	1	1



а б в г

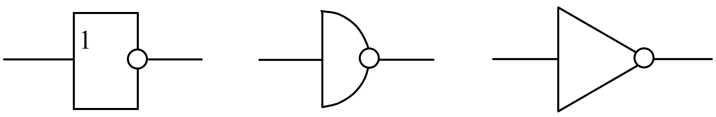
Рис. 2. Условные обозначения логического элемента ИЛИ:

а) по ГОСТ, б) по стандарту МЭК, в) по стандарту DIN, г) по стандарту milspec

Инверсия (отрицание, операция НЕ, NOT): функция одной переменной, принимает единичное значение, если входная переменная равна «0». Функция равна «0», если входная переменная равна «1» (табл. 7).

Таблица 7

x	f
0	1
1	0



а б в

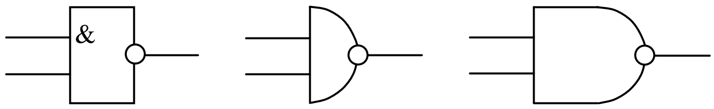
Рис. 3. Условные обозначения логического элемента НЕ:

а) по ГОСТ и стандарту МЭК, б) по стандарту DIN, в) по стандарту milspec

2.2. Распространённые логические операции и логические элементы

Наряду с простейшими распространены и более сложные логические элементы, сочетающие в себе несколько простейших операций. Такими являются логические элементы И-НЕ, ИЛИ-НЕ, ЭКВИВАЛЕНТНОСТЬ, ИСКЛЮЧАЮЩЕЕ ИЛИ и т.п.

Штрих Шеффера. Элемент И-НЕ реализует функцию «штрих Шеффера» двух переменных $f = \overline{x_1 \cdot x_2}$ и имеет соответствующую таблицу истинности (табл. 3). Условное обозначение логического элемента И-НЕ в любом стандарте объединяет в себе обозначение элемента И и кружок, являющийся признаком элемента НЕ (рис. 4).

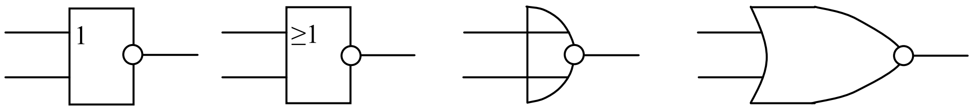


а б в

Рис. 4. Условные обозначения логического элемента И-НЕ:

а) по ГОСТ и стандарту МЭК, б) по стандарту DIN, в) по стандарту milspec

Стрелка Пирса. Элемент ИЛИ-НЕ реализует функцию «стрелка Пирса»: $f = \overline{x_1 \vee x_2}$, описанную в табл. 3. Условные обозначения объединяют в себе обозначение элемента ИЛИ и кружок – символ операции отрицания НЕ (рис. 5).



а б в г

Рис. 5. Условные обозначения логического элемента ИЛИ-НЕ:

а) по ГОСТ, б) по стандарту МЭК, в) по стандарту DIN, г) по стандарту milspec

Эквивалентность. Элемент ЭКВИВАЛЕНТНОСТЬ (РАВНОЗНАЧНОСТЬ) описывается функцией ψ_{10} из табл. 3. В ней выходная переменная f принимает единичные значения только при равенстве входных переменных $x_1{=}x_2$. Функция имеет собственное обозначение и может быть выражена через простейшие логические операции: $f = \overline{x_1} \cdot \overline{x_2} \vee x_1 \cdot x_2$.

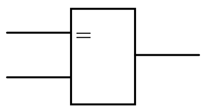


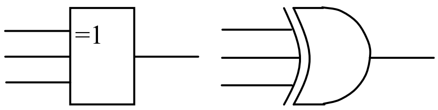
Рис. 6. Условное обозначение логического элемента ЭКВИВАЛЕНТНОСТЬ

Исключающее ИЛИ. Элемент ИСКЛЮЧАЮЩЕЕ ИЛИ, «1 и только 1», XOR (Exclusive OR) работает в соответствии с табл. 8.

Таблица 8

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

В таблице единичные значения функции соответствуют строкам, содержащим только одну единицу. Функция сравнительно просто выражается с помощью элементарных логических операций: $f = \overline{x}_1 \cdot \overline{x}_2 \cdot x_3 \vee \overline{x}_1 \cdot x_2 \cdot \overline{x}_3 \vee x_1 \cdot \overline{x}_2 \cdot \overline{x}_3$.



а б

Рис. 7. Условные обозначения логического элемента ИСКЛЮЧАЮЩЕЕ ИЛИ:

а) по ГОСТ и по стандарту МЭК, б) по стандарту milspec

Сумма по модулю 2. Элемент СУММА ПО МОДУЛЮ 2 (MOD2) работает в соответствии с табл. 9.

Таблица 9

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

В таблице единичные значения функции соответствуют строкам, в которых младший разряд арифметической суммы входных переменных равен 1: $f = x_1 \oplus x_2 \oplus x_3$.

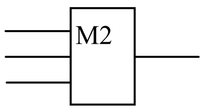


Рис. 8. Условное обозначение логического элемента СУММА ПО МОДУЛЮ 2

Неэквивалентность. Элемент НЕЭКВИВАЛЕНТНОСТЬ, (НЕРАВНОЗНАЧНОСТЬ) работает в соответствии с табл. 3. В таблице единичные значения функции соответствуют неравенству входных переменных.

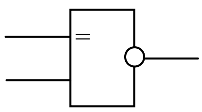


Рис. 9. Условное обозначение логического элемента НЕЭКВИВАЛЕНТНОСТЬ

Мажоритарность. Мажоритарный элемент имеет много входов. Выходная переменная элемента принимает единичное значение, если большая часть её входных переменных равна единице. Так, переменная f на выходе трехвходового мажоритарного элемента принимает единичное значение, если два или три его входа имеют единичное значение (табл. 10).

Таблица 10

x_1	x_2	x_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Логическая функция элемента может быть выражена через элементарные логические операции: $f=x_1x_2\vee x_1x_3\vee x_2x_3$.

Условное обозначение элемента по ГОСТ и стандарту МЭК приведено на рис. 10.

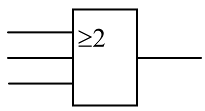


Рис. 10. Условное обозначение трехвходового мажоритарного элемента

2.3. Функционально-полный набор логических элементов

Набор логических элементов, достаточный для построения любой сколь угодно сложной логической схемы, называется функционально полным.

Функционально полным является набор элементов И, ИЛИ, НЕ. Из этого набора можно исключить некоторые элементы без нарушения функциональной полноты. В частности, функционально-полным считается набор из двух элементов И и НЕ. В этом случае для выполнения операции ИЛИ двух переменных x_1 и x_2 просто по уравнению $x_1\vee x_2=\overline{\overline{x_1}\cdot\overline{x_2}}$ строится схема на трех элементах НЕ и одном элементе И.

Аналогично, функционально полным является набор из элементов ИЛИ и элементов НЕ. На основании формулы де Моргана элемент И реализуется по уравнению $x_1\cdot x_2=\overline{\overline{x_1}\vee\overline{x_2}}$.

Свойство функциональной полноты используется при разработке и реализации многовходовых логических схем, так как реальные элементы интегральных схем выполняют или функцию И-НЕ (схемы транзисторно-транзисторной логики и КМОП технологии), или функцию ИЛИ-НЕ (схемы и-МОП и КМОП технологии). С помощью логических элементов ИЛИ-НЕ или И-НЕ можно собрать любую логическую схему. На таких элементах собран микропроцессор компьютера и другие логические устройства. Переход от логического умножения к логическому сложению (и обратно) и позволяет строить различные логические схемы, используя ограниченный функциональный набор логических элементов.

Для разработки структуры многовходовой логической схемы в заданном базисе применяют следующий алгоритм:

– составляется таблица истинности;

– по таблице истинности составляется булева функция в виде совершенной дизъюнктивной нормальной форме;

– используя правила булевой алгебры, карты Карно или другие способы минимизации, функция (если возможно) упрощается;

– полученное выражение приводят к заданному базису, применяя правило де Моргана. Если в качестве базы заданы элементы И-НЕ, правило де Моргана применяется к дизъюнкции, если ИЛИ-НЕ – к конъюнкции.

2.4. Карты Карно

Компактной и очень удобной формой записи логической функции, используемой наряду с таблицей истинности, является карта Карно. *Карта Карно* является специальной компактной формой таблицы истинности, которая позволяет не только представить функцию, но и минимизировать ее. Количество клеток в карте Карно равно количеству строк в таблице истинности. Каждая клетка соответствует одной строке таблицы. Комбинации входных переменных распределяются по двум сторонам прямоугольника, а соответствующие значения функции в клетках таблицы, находящихся на пересечении строк и столбцов, соответствующих выбранным состояниям переменных.

Карта Карно для функции двух переменных содержит четыре клетки и имеет форму квадрата (табл. 11). Два возможных значения первой переменной x_1 отражаются обычно на верхней стороне квадрата, значения второй переменной x_2 – на левой стороне.

Пример. Составить карту Карно для функции ЭКВИВАЛЕНТНОСТЬ, соответствующей значениям ψ_{10} из табл. 3.

Соседними считаются клетки карты, отличающиеся значениями только одной входной переменной. В карте Карно двух переменных (табл. 11) каждая клетка имеет две соседние.

Карта Карно для функции трех переменных состоит из 8 клеток и имеет обычно 2 строки и четыре столбца (табл. 12).

Таблица 11

	x_1	0	1
x_2	0	1	0
	1	0	1

Таблица 12

	$x_1\,x_2$	00	01	11	10
x_3	0	0	0	1	0
	1	0	1	1	1

На верхней стороне прямоугольника каждому столбцу ставится в соответствие одна комбинация входных переменных x_1 и x_2. Причем, при переходе от каждого столбца к соседнему имеет право измениться только одна переменная, а первый и последний столбцы карты также считаются соседними. В карте трех переменных каждая клетка имеет три соседние.

Пример. В карту, приведённую в таблице 12, занесём функцию из таблицы 10.

В карте Карно для функции четырех переменных 16 клеток, размещенных в четырех столбцах и четырех строках. Две переменные x_1 и x_2 располагаются наверху квадрата, а две другие x_3 и x_4 – слева (табл. 13). В отличие от предыдущего случая здесь каждой строчке таблицы соответствует определенная комбинация двух переменных x_3 и x_4. При переходе от каждой строки к соседней меняется только одна переменная, а первая и последняя строки карты, так же как и крайние столбцы, считаются соседними. Каждая клетка карты имеет четыре соседние клетки.

Пример. Занесём в карту Карно для четырёх переменных следующую логическую функцию: $f = \overline{x_1} \cdot \overline{x_2} \vee x_1 \cdot x_3 \cdot \overline{x_4} \vee \overline{x_2} \cdot \overline{x_4}$.

Таблица 13

x_1x_2		00	01	11	10
x_3x_4	00	1	0	0	1
	01	1	0	0	0
	11	1	0	0	0
	10	1	0	1	1

2.5. Минимизация логических функций

Законы и теоремы булевой алгебры позволяют минимизировать (упростить) логическое выражение. При небольшом количестве переменных минимизацию удобно осуществлять непосредственно по карте Карно. Если в карте Карно встречаются группы из 2-х, 4-х, 8-ми и т.д. соседних ячеек, содержащих единицы, которые можно выделить контуром в виде квадрата или прямоугольника, то такая группа может быть описана одним логическим произведением. В это произведение входят только неизменные для всех ячеек данной группы переменные. Например, в карте Карно четырех переменных табл. 13 можно выделить группу из четырех клеток в первом столбце, группу из четырех угловых клеток и группу из двух соседних клеток в нижней строке (табл. 14).

Таблица 14

x_1x_2		00	01	11	10
x_3x_4	00	1	0	0	1
	01	1	0	0	0
	11	1	0	0	0
	10	1	0	1	1

В результате минимизированная функция представляет собой сумму трех произведений, соответствующих отдельным группам:

$$f = \overline{x_1} \cdot \overline{x_2} \vee x_1 \cdot x_3 \cdot \overline{x_4} \vee \overline{x_2} \cdot \overline{x_4} .$$

3. СИНТЕЗ И АНАЛИЗ ЛОГИЧЕСКИХ ФУНКЦИЙ И СХЕМ

В цифровой электронике известны основные логические функции И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ, сложение по модулю 2 (исключающее ИЛИ), которые находят наиболее широкое применение при реализации цифровых устройств различного назначения. При представлении логической функции математическим выражением используют два вида ее представления: дизъюнктивная нормальная форма и конъюнктивная нормальная форма.

3.1. Дизъюнктивная и конъюнктивная нормальные формы

Элементарной конъюнкцией называется конъюнкция переменных или их отрицаний, в которой каждая переменная встречается не более одного раза. *Дизъюнктивной нормальной формой* (ДНФ) называется формула, имеющая вид дизъюнкции элементарных конъюнкций. *Совершенная ДНФ* – это ДНФ, каждая элементарная конъюнкция которой включает все переменные с отрицанием или без.

Для получения СДНФ по таблице истинности, в ней выделяют строки, в которых функция принимает единичные значения. Для каждой выделенной строки составляется конъюнкция всех входных переменных, причем сомножитель записывают со знаком инверсии, если переменная принимает в этой строке нулевое значение. Записывается логическая сумма всех составленных логических произведений, которые носят названия конституент единицы, или минтермов. Таким образом, СДНФ функции f содержит ровно столько конъюнкций, сколько единиц в таблице истинности, и все полученные конъюнкции соединяются знаками дизъюнкции. Для каждой функции СДНФ единственна (с точностью до перестановок переменных или конъюнкций).

Пример. Пусть задана логическая функция трёх переменных, которая равна единице в случае, если хотя бы две из входных переменных равны «1» (см. табл. 10). Требуется записать СДНФ этой функции.

Для данной логической функции СДНФ имеет вид:

$$f(x_1, x_2, x_3) = \overline{x_1} x_2 x_3 \vee x_1 \overline{x_2} x_3 \vee x_1 x_2 \overline{x_3} \vee x_1 x_2 x_3 .$$

Конъюнктивной нормальной формой (КНФ) называется логическое произведение элементарных сумм, в каждую из которых аргумент или его отрицание входят один раз.

КНФ может быть получена из таблицы истинности: для каждого набора аргументов на котором функция равна «0» составляют элементарную сумму, причем переменные, значение которых равно «1», записываются с отрицанием. Полученные суммы, которые носят название конституент нуля, или макстермов, объединяют операцией логического умножения.

Пример. КНФ для функции из табл. 10 имеет вид:

$$f(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3)(x_1 \vee x_2 \vee \overline{x_3})(x_1 \vee \overline{x_2} \vee x_3)(\overline{x_1} \vee x_2 \vee x_3) .$$

КНФ также называется *совершенной*, если каждая элементарная сумма содержит все переменные с инверсией или без.

Иногда удобнее пользоваться не самой логической функцией, а ее инверсией. В этом случае при использовании вышеописанных методик для записи СДНФ надо использовать нулевые, а для записи СКНФ – единичные значения функции.

Пример. Для логической функции из табл. 10 СДНФ и СКНФ инверсной функции имеют вид:

СДНФ: $\overline{f}(x) = \overline{x_1}\,\overline{x_2}\,\overline{x_3} + \overline{x_1}\,\overline{x_2} x_3 + \overline{x_1} x_2 \overline{x_3} + x_1 \overline{x_2}\,\overline{x_3}$,

СКНФ: $\overline{f}(x) = (x_1 + \overline{x_2} + \overline{x_3})(\overline{x_1} + x_2 + \overline{x_3})(\overline{x_1} + \overline{x_2} + x_3)(\overline{x_1} + \overline{x_2} + \overline{x_3})$.

3.2. Переход от логической функции к логической схеме

Задача синтеза. По заданной функции f требуется построить схему, реализующую данную функцию. Задача синтеза решается неоднозначно. Можно поставить в соответствие заданной функции f целое множество схем.

Для построения логической схемы необходимо элементы, предназначенные для выполнения логических операций, указанных в логической функции, располагать в порядке, указанном в булевом выражении.

Пример. Построить логическую схему устройства, реализующего логическую функцию $f = \overline{x}_1 x_2 x_3 \vee x_1 \overline{x}_2 x_3 \vee x_1 x_2 \overline{x}_3 \vee x_1 x_2 x_3$.

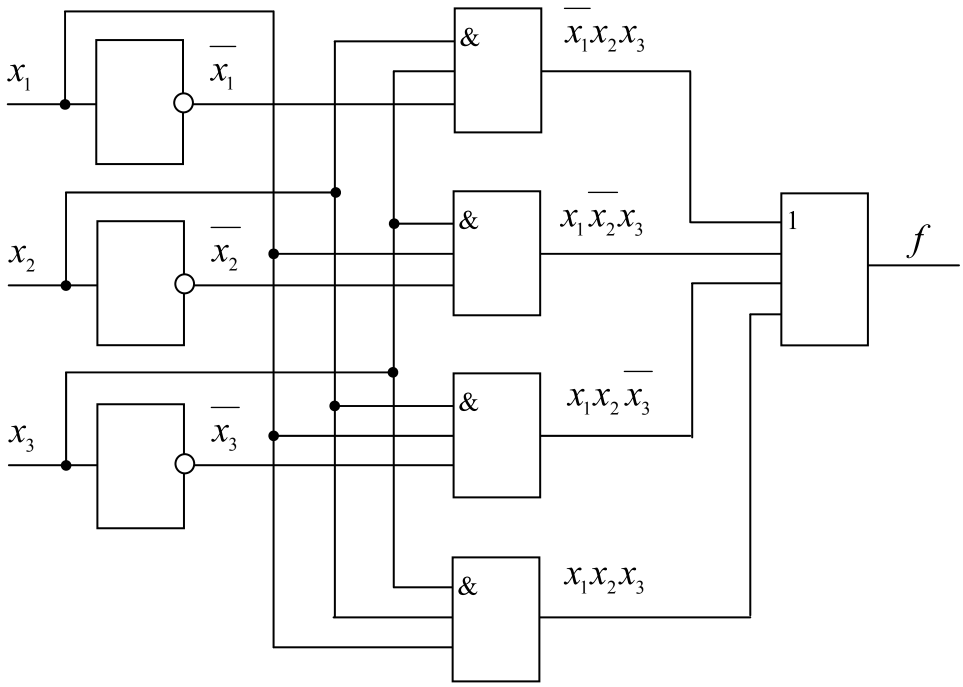


Рис. 11. Пример логической схемы устройства

3.3. Синтез логических устройств в заданном базисе

С целью уменьшения номенклатуры используемых микросхем, часто пользуются функционально полной системой в составе двух логических элементов выполняющих операции И-НЕ, ИЛИ-НЕ. Любую логическую функцию можно записать в заданном базисе логических элементов. Если задан базис И-НЕ, то путем двойного инвертирования исходного выражения или его части и применения теорем де Моргана логическая функция приводится к виду, содержащему только операции логического умножения и инвертирования. Если же задан базис ИЛИ-НЕ, исходную логическую функцию теми же приемами приводят к виду, содержащему только операции логического сложения и инверсии. Далее логическое выражение записывается через условные обозначения выбранных операций.

Пример. Заданную функцию *f* перевести в базисы И-НЕ и ИЛИ-НЕ.

Исходная ДНФ в базисе И-НЕ имеет вид:

$$f = x_2 x_4 + \overline{x}_1 x_3 \overline{x}_4 + x_1 \overline{x}_2 \overline{x}_3 = \overline{\overline{x_2 x_4 + \overline{x}_1 x_3 \overline{x}_4 + x_1 \overline{x}_2 \overline{x}_3}} = \overline{(\overline{x_2 x_4})(\overline{\overline{x}_1 x_3 \overline{x}_4})(\overline{x_1 \overline{x}_2 \overline{x}_3})} =$$
$$= (x_2 | x_4)(\overline{x_1} | x_3 | \overline{x_4})(x_1 | \overline{x_2} | \overline{x_3}) .$$

Аналогично, КНФ в базисе ИЛИ-НЕ имеет вид:

$$f = \overline{\overline{(x_1 + x_4)(\overline{x_1} + x_2 + x_3)(\overline{x_2} + \overline{x_3} + \overline{x_4})}} = \overline{\overline{(x_1 + x_4)} + \overline{(\overline{x_1} + x_2 + x_3)} + \overline{(\overline{x_2} + \overline{x_3} + \overline{x_4})}} =$$
$$= (x_1 \downarrow x_4) \downarrow (\overline{x_1} \downarrow x_2 \downarrow x_3) \downarrow (\overline{x_2} \downarrow \overline{x_3} \downarrow \overline{x_4}).$$

Пример. Пусть логическая функция задана выражением

$$f = x_1 x_4 + \overline{(\overline{x_1} + x_3 + \overline{x_4})}(x_2 + \overline{x_3})$$

Привести логическую функцию в базис И-НЕ, ИЛИ-НЕ.

а) приводим функцию к базису И-НЕ

$$f = f_1 + \overline{f_2}f_3 = f_1 + \overline{\overline{\overline{f_2}f_3}} = f_1 + \overline{(\overline{f_2}|f_3)} = \overline{\overline{f_1 + \overline{(\overline{f_2}|f_3)}}} = \overline{\overline{f_1}\,\overline{\overline{(\overline{f_2}|f_3)}}} = \overline{f_1}|(\overline{f_2}|f_3)\,;$$

$$\overline{f_1} = \overline{x_1x_4} = x_1|x_4\,;$$

$$\overline{f_2} = \overline{\overline{x_1} + x_3 + \overline{x_4}} = x_1\overline{x_3}x_4 = \overline{\overline{x_1\overline{x_3}x_4}} = \overline{x_1|\overline{x_3}|x_4}\,;$$

$$f_3 = x_2 + \overline{x_3} = \overline{\overline{x_2 + \overline{x_3}}} = \overline{\overline{x_2}x_3} = \overline{x_2}|x_3\,;$$

$$f = (x_1|x_4)|\overline{((x_1|\overline{x_3}|x_4)|(\overline{x_2}|x_3))}\,.$$

б) приводим функцию к базису ИЛИ-НЕ

$$f = f_1 + \overline{f_2}f_3 = f_1 + \overline{\overline{\overline{f_2}f_3}} = f_1 + \overline{(f_2 + \overline{f_3})} = \overline{\overline{f_1 + \overline{(f_2 + \overline{f_3})}}} = \overline{f_1 \downarrow (f_2 \downarrow \overline{f_3})}\,;$$

$$f_1 = x_1x_4 = \overline{\overline{x_1x_4}} = \overline{\overline{x_1} + \overline{x_4}} = \overline{x_1} \downarrow \overline{x_4}\,;$$

$$f_2 = \overline{x_1} + x_3 + \overline{x_4} = \overline{\overline{\overline{x_1} + x_3 + \overline{x_4}}} = \overline{\overline{x_1} \downarrow x_3 \downarrow \overline{x_4}}\,;$$

$$f_3 = x_2 + \overline{x_3} = \overline{\overline{x_2 + \overline{x_3}}} = \overline{x_2 \downarrow \overline{x_3}} \Rightarrow \overline{f_3} = x_2 \downarrow \overline{x_3}\,;$$

$$f = \overline{(x_1 \downarrow x_4) \downarrow ((\overline{x_1} \downarrow x_3 \downarrow \overline{x_4}) \downarrow (x_2 \downarrow \overline{x_3}))}\,.$$

3.4. Переход от логической схемы к логической функции

Задача анализа. По заданной схеме требуется определить функцию *f*, реализующуюся данной схемой.

При решении задачи анализа следует придерживаться следующей последовательности действий:

1) Заданная схема разбивается по ярусам.
2) Начиная с последнего, выходы каждого элемента обозначаются проиндексированными функциями в зависимости от яруса, к которому относится элемент.
3) Записываются выходные функции каждого элемента в виде формул в соответствии с введенными обозначениями.
4) Производится подстановка одних выходных функций через другие, используя входные переменные.
5) Записывается получившаяся булева функция через входные переменные.

Пример. По заданной логической схеме (рис. 12) составить булеву функцию.

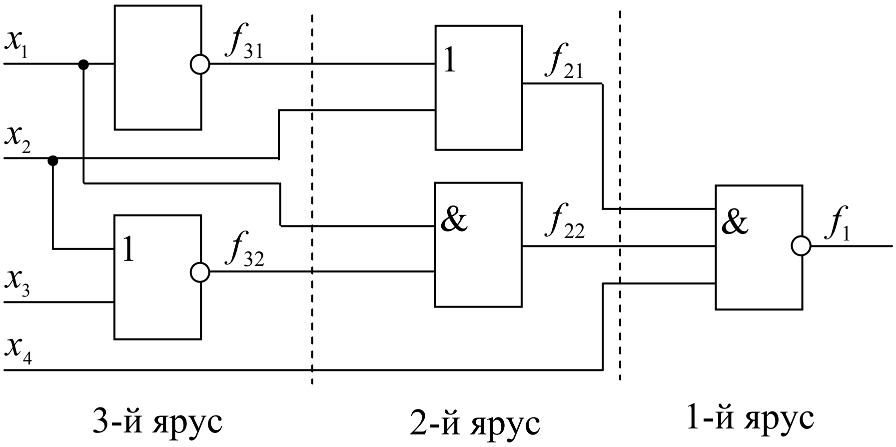


Рис. 12. Пример логической схемы устройства

Согласно приведённой выше последовательности действий, произведём разбиение схемы на ярусы. Пронумеровав получившиеся ярусы, введём обозначения для каждой выходной функции (рис. 12). Запишем все функции, начиная с 1-го яруса:

1) $f_1 = \overline{f_{21} \cdot f_{22} \cdot x_4}$

2) а) $f_{21} = f_{31} \vee x_2$, б) $f_{22} = f_{32} \cdot x_1$

3) а) $f_{31} = \overline{x_1}$, б) $f_{32} = \overline{x_2 \vee x_3}$

Теперь запишем все функции, подставляя входные переменные $x_1, ..., x_4$:

а) $f_{21} = \overline{x_1} \vee x_2$, б) $f_{22} = x_1 \cdot (\overline{x_2 \vee x_3})$

В итоге, получим выходную функцию:

$$f = f_1 = \overline{x_1 \cdot (\overline{x_1} \vee x_2) \cdot (\overline{x_2 \vee x_3}) \cdot x_4}$$

4. КОМБИНАЦИОННЫЕ СХЕМЫ

4.1. Основные сведения

Логическая схема, которая полностью описывается булевыми выражениями или таблицами истинности, называется *комбинационной схемой*. Таким образом, комбинационная схема – схема, в которой значения входных переменных в текущий момент времени полностью определяют значения выходных переменных.

Все рассмотренные выше схемы являются комбинационными схемами.

Другой класс схем – *последовательностные схемы*. Это схемы с внутренней памятью. В них значения выходных переменных определяются не только значениями входных переменных в текущий момент времени, но и их значениями в предыдущие моменты времени.

Комбинационные схемы рассматриваются как совокупность следующих групп: базовые логические элементы; коммутационные узлы (ключи, мультиплексоры, селекторы, в том числе многофункциональные, матричные коммутаторы и др.); преобразователи кодов (дешифраторы, шифраторы, приоритетные шифраторы, специализированные преобразователи двоичных кодов); постоянные запоминающие устройства (ПЗУ); арифметические узлы (четвертьсумматоры, полусумматоры, сумматоры, схемы ускоренного переноса, инкременторы, декременторы, цифровые матричные умножители, цифровые компараторы, арифметико-логические устройства (АЛУ), схемы контроля на четность/нечетность и др.).

Пример. Рассмотрим мультиплексор, представленный на рис. 13.

Назначение мультиплексора (multiplex – многократный) – коммутировать в желаемом порядке информацию, поступающую с нескольких входных шин на одну выходную. С помощью мультиплексора осуществляется временное разделение информации, поступающей по разным каналам.

Рис. 13 отражает структуру мультиплексора «четыре линии к одной» (4:1) – половину микросхемы ТТЛ К155КП2.

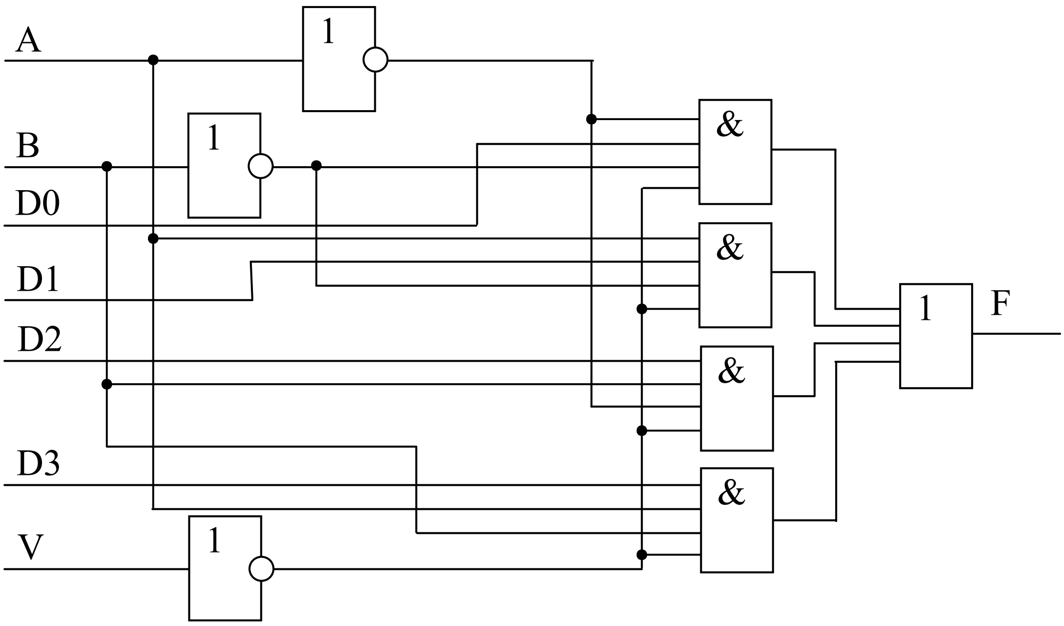


Рис. 13. Логическая схема мультиплексора 4:1

4.2. Корректность логических функций и комбинационных схем

К ошибкам, совершаемым при записи логических функций, относятся следующие:

1) подмена индексов у переменных,

2) наличие функции в части равенства, где записаны переменные и логические операции, совершаемые с этими переменными,

3) потеря инверсии переменных, и т.п.

Проведённый анализ типичных ошибок, совершаемых при самостоятельном проектировании комбинационных схем, показывает, что к наиболее встречающимся следует отнести:

1) наличие в комбинационной схеме обратных связей,

2) отсутствие соединительных узлов в линиях сигналов,

3) отсутствие в элементах символов логических операций,

4) использование различных масштабов для изображения элементов, и т.п.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Майоров С.А., Новиков Г.И. Принципы организации цифровых машин. – Л.: «Машиностроение», Ленингр. отд-ние, 1974. – 432 с.

2. Москинова Г.И. Дискретная математика. Математика для менеджера в примерах и упражнениях: Учебное пособие. – М.: Логос, 2000. – 240 с.

3. Дискретная математика для программистов / Ф.А. Новиков. – СПб: Питер, 2001. – 304 с.

4. Архангельский С.В., Шафоростов Ю.И. Методические указания к выполнению контрольной работы по дисциплине «Теория дискретных устройств железнодорожной автоматики, телемеханики и связи». – Куйбышев: КИИТ, 1987. – 25 с.

5. Никищенков С.А., Смышляев В.А., Юшков С.А. Задания к выполнению контрольной работы по курсу «Математическая логика и теория алгоритмов» для студентов заочного обучения по специальности 071900 «Информационные системы в технике и технологиях». – Самара: СамИИТ, 2002. – 7 с.