ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ, СИСТЕМЫ И СЕТИ

ИЗДАТЕЛЬСТВО ТГТУ

УДК 004.3
ББК З973.202я73
В393

Рекомендовано Редакционно-издательским советом университета

Р е ц е н з е н т ы :
Кандидат технических наук, профессор ТГТУ
В.М. Иванов

Кандидат технических наук, доцент, заместитель начальника кафедры «Импульсной техники и электронных приборов»
ТВВАИУРЭ (ВИ)
В.М. Строев

С о с т а в и т е л ь
А.Н. Ветров

В393 Вычислительные машины, системы и сети : методические рекомендации / сост. А.Н. Ветров. – Тамбов : Изд-во Тамб. гос. техн. ун-та, 2008. – 36 с. – 100 экз.

Изложены арифметические и логические основы построения электронных вычислительных устройств.

Предназначены для студентов 3 курса специальности 200402 «Инженерное дело в медико-биологической практике» дневной и заочной форм обучения и могут быть использованы при проведении такой формы самостоятельной работы студентов, как работа с книгой.

УДК 004.3
ББК З973.202я73

Министерство образования и науки Российской Федерации

ГОУ ВПО «Тамбовский государственный технический университет»

ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ, СИСТЕМЫ И СЕТИ

Часть I

Методические рекомендации
по выполнению курсовой работы
для студентов 3 курса специальности 200402
«Инженерное дело в медико-биологической практике»
дневной и заочной форм обучения

Тамбов
Издательство ТГТУ
2008

Учебное издание

ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ, СИСТЕМЫ И СЕТИ

Часть I

Методические рекомендации

Составитель
ВЕТРОВ Александр Николаевич

Редактор Ю.В. Шиманова
Инженер по компьютерному макетированию М.А. Филатова

Подписано в печать 7.11.2008.
Формат 60 × 84/16. 2,09 усл. печ. л. Тираж 100 экз. Заказ № 484.

Издательско-полиграфический центр
Тамбовского государственного технического университета
392000, Тамбов, Советская, 106, к. 14

ВВЕДЕНИЕ

Вычислительная техника приобретает всё большее применение в различных отраслях техники и даже в быту. Широкое применение вычислительная техника получила в аппаратуре средств автоматизации и в системах автоматического управления. Автоматические и автоматизированные системы управления осуществляют сбор, хранение, передачу и переработку информации, отражающей состояние регулируемых объектов. Информация, выработанная системой, используется для оперативного воздействия на управляемый объект (процесс) с целью поддержания нужного состояния. Основу подобных систем управления составляют вычислительные машины.

Нет ни одной отрасли, в которой не использовались бы электронные вычислительные машины (ЭВМ). Внедряется, как принято говорить, всеобщая компьютеризация, вплоть до использования ЭВМ в домашнем хозяйстве. Решается задача всеобщей компьютерной грамотности.

Таким образом, современные специалисты должны иметь представление об информационных, арифметических и логических основах построения ЭВМ, о методах анализа и синтеза устройств ЭВМ. Изложенный материал закладывает необходимую базу знаний, позволяющую студентам самостоятельно изучить теоретические основы построения вычислительной техники.

1. АРИФМЕТИЧЕСКИЕ ОСНОВЫ ПОСТРОЕНИЯ ЭВМ

1.1. СИСТЕМЫ СЧИСЛЕНИЯ

Системами счисления (СС) называют совокупность правил изображения чисел с помощью цифровых знаков.

Все системы счисления делятся на 2 группы: позиционные и непозиционные.

В непозиционных системах счисления значимость цифры в любом месте числа одинакова (например, римская система счисления).

В позиционных системах счисления значимость цифры зависит от того места (позиции), которое данная цифра в числе занимает.

Позиционные СС характеризуются основанием (базисом) – количество знаков или символов, используемых в разрядах для изображения числа в данной СС.

Общепринятая система счисления десятичная, а это значит, что каждый разряд числа отличается от соседнего в 10 раз.

Основанием системы счисления может быть любое целое число. При любом основании $\boldsymbol{q}$ произвольное число $\boldsymbol{X_q}$ можно записать в следующем виде:

$$X_q = a_{n-1}q^{n-1} + a_{n-2}q^{n-2} + \ldots\ldots + a_1q^1 + a_0q^0 + a_{-1}q^{-1} + \ldots\ldots + a_{-m}q^{-m},$$

где $\boldsymbol{q}$ – основание системы счисления; $\boldsymbol{a_i}$ – весовой коэффициент разряда (цифра); $\boldsymbol{n}$, $\boldsymbol{m}$ – количество целых и дробных разрядов.

Самая младшая цифра целой части числа имеет нулевую степень, поэтому самый младший разряд целой части числа называется нулевым разрядом.

Пример: $1053{,}25_{10} = 1\cdot10^3 + 0\cdot10^2 + 5\cdot10^1 + 3\cdot10^0 + 2\cdot10^{-1} + 5\cdot10^{-2}$.

В технике десятичная система счисления не используется, а в вычислительной технике используется двоичная система счисления (0 и 1). Это обусловлено тем, что систему технически легко реализовать: например, 0 – напряжение отсутствует; 1 – какой-то заданный уровень напряжения.

1.2. ПЕРЕВОД ЧИСЕЛ ИЗ ОДНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГУЮ

При переводе числа из какой-либо СС в десятичную необходимо представить это число в виде полинома по степеням основания, возвести в степень и просуммировать.

$$X_n = a_{n-1}q^{n-1} + a_{n-2}q^{n-2} + \ldots\ldots + a_1q^1 + a_0q^0 + a_{-1}q^{-1} + \ldots\ldots + a_{-m}q^{-m},$$

$10110110{,}11_2 = 182{,}75_{10}$.

Для того чтобы сопоставить это число с десятичной системой счисления, т.е. перевести в десятичную систему, воспользуемся приведённой формулой.

8 разрядов, т.е. $n = 8$.

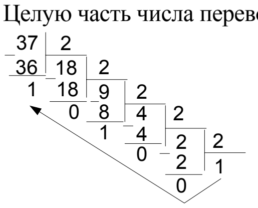
$$1\cdot2^7 + 0\cdot2^6 + 1\cdot2^5 + 1\cdot2^4 + 0\cdot2^3 + 1\cdot2^2 + 1\cdot2^1 + 0\cdot2^0 + 1\cdot2^{-1} + 1\cdot2^{-2} =$$
$$= 128 + 32 + 16 + 4 + 2 + \frac{1}{2} + \frac{1}{4} = 182{,}75_{10}.$$

Для перевода числа из десятичной системы счисления в любую другую существует правило для целой и дробной частей числа.

При переводе целой части числа необходимо это число последовательно делить на основание новой СС до тех пор, пока не получится частное меньшее основания СС. Число в новой СС записывается в виде остатков от деления, начиная с последнего.

Правило для дробной части числа: при переводе дробной части числа её необходимо последовательно умножать на основание новой СС до тех пор, пока в дробной части не получится ноль, или же до требуемого количества знаков после запятой. Перемножаются каждый раз только дробные части. Дробь в новой СС состоит из целых частей получающихся произведений, начиная с первой.

П р и м е р.
Число $37{,}85_{10}$ перевести в двоичную СС с точностью до четырёх знаков.
Целую часть числа переводим делением его на 2 (основание двоичной СС).



Дробную часть 0,85 переводим последовательным умножением на число 2.
$0{,}85 \cdot 2 = \mathbf{1}{,}7$
$0{,}7 \cdot 2 = \mathbf{1}{,}4$
$0{,}4 \cdot 2 = \mathbf{0}{,}8$
$0{,}8 \cdot 2 = \mathbf{1}{,}6$
$\mathbf{0{,}85_{10} = 0{,}1101_2}$

Для перевода числа из двоичной СС в восьмеричную и шестнадцатеричную и наоборот можно пользоваться упрощённым способом перевода, суть которого состоит в следующем: двоичное число при переводе в восьмеричную СС делится на триады справа налево в целой части числа и слева направо в его дробной части, каждая триада заменяется восьмеричным числом.

П р и м е р.
Число $1110011{,}\ 11_2$ перевести в восьмеричную СС.

00**1 110 011, 11**0_2 = $163{,}6_8$
1 6 3 6

Двоичное число при переводе в шестнадцатеричную СС делится на тетрады справа налево в целой части числа и слева направо в его дробной части, и каждая тетрада заменяется шестнадцатеричным числом.

П р и м е р.
Число $1110011{,}\ 11_2$ перевести в шестнадцатеричную СС.

0**111 0011, 11**00_2 = $73{,}C_{16}$
7 3 C

Для перевода восьмеричного числа в двоичную СС следует каждое восьмеричное число заменить эквивалентным трёхразрядным двоичным числом, а при переводе шестнадцатеричного числа – каждое шестнадцатеричное число заменить эквивалентным четырёхразрядным двоичным числом.

П р и м е р.
Число $157{,}\ 34_8$ перевести в двоичную СС.
1 5 7 , 3 4
001 101 111 011 100
$157{,}\ 34_8$ = 00**1101111,01110**0_2
или $157{,}\ 34_8 = 1101111{,}0111_2$

Число $5A3B{,}\ 6C_{16}$ перевести в двоичную СС.
5 A 3 B , 6 C
0101 1010 0011 1011 0110 1100
$5A3B{,}\ 6C_{16}$ = 0**101101000111011, 011011**00_2
или $5A3B{,}\ 6C_{16}$ = $\mathbf{101101000111011{,}\ 011011_2}$

В цифровых системах все алгебраические операции: вычитание, умножение и деление, сводятся к одной операции, к сложению.

В реальных цифровых системах приходится складывать многоразрядные числа. Необходимо учитывать следующее:

– при сложении двух единиц в данном разряде получится ноль с переносом единицы в старший разряд;
– при сложении трёх единиц в данном разряде получается единица с переносом единицы в старший разряд;
– при сложении четырёх единиц в данном разряде получается 0 с переносом единицы через разряд.

Некоторые числа в различных системах счисления приведены в приложении А.

1.3. КОДИРОВАНИЕ АЛФАВИТОВ ВХОДНОЙ ИНФОРМАЦИИ В ЭВМ

Любая информация (числа, алфавитно-цифровые записи) представляется в ЭВМ в виде двоичных кодов. Отдельные элементы двоичного кода, 0 или 1, называют разрядами или битами. Минимальной единицей информации, обрабатываемой в ЭВМ, является байт, состоящий из 8 бит.

В каждой ЭВМ имеются узлы, рассчитанные на определённое количество разрядов. Совокупность таких разрядов, в которые записывается двоичное слово, образует разрядную сетку устройства.

Байты могут обрабатываться отдельно или группами из последовательно расположенных байтов. Группы байтов образуют двоичные слова, которые могут быть фиксированной и переменной длины. Структура размещения байтов по длине разрядной сетки определяется форматами данных в ЭВМ.

Различают:

- форматы данных фиксированной длины (полуслово, слово и двойное слово) состоят соответственно из двух, четырёх и восьми последовательно расположенных байтов;
- форматы данных переменной длины состоят из группы последовательно расположенных байтов от 1 до 256.

В формате данных фиксированной длины обычно представляются двоичные числа, команды и некоторые логические данные, а в формате данных переменной длины – десятичные числа, алфавитно-цифровая и некоторая другая логическая информация.

Формы представления чисел

Для представления информации в ЭВМ можно использовать два способа, или две формы, представления чисел.

С фиксированной точкой (естественная форма) представления чисел заключается в том, что числа вводятся в виде целой и дробной частей, т.е. положение точки фиксируется в строго определённом месте относительно разрядов числа. Наибольшее распространение получило представление чисел в виде правильной дроби. Естественная форма представления чисел используется в машинах, где не требуется высокая скорость работы (быстродействие), так как организация выполнения арифметических операций в этом случае проще. Для ввода чисел в ЭВМ с фиксированной запятой используются масштабные коэффициенты.

С плавающей точкой (нормальная форма) представления чисел заключается в том, что числа в ЭВМ вводятся в полулогарифмическом виде и состоят из двух частей: мантиссы числа и порядка числа. Мантисса числа обозначается m, по абсолютному значению она меньше 1: $|\boldsymbol{m}| < 1$. Порядок числа обозначается буквой p и является целым числом.

Число X в системе счисления q в полулогарифмическом виде записывается как:

$$\boldsymbol{X = mq^p}.$$

Разрядная сетка ЭВМ для нормальной формы представления чисел содержит две группы цифр: для мантиссы и для порядка.

Положение запятой в числе зависит от порядка p, поэтому ЭВМ, использующие этот вид представления чисел, получили название машин с плавающей запятой (плавающей точкой), перемещающейся по записи числа.

Пример. Двоичное число $X = -101{,}11_2$ в нормальной, или полулогарифмической, форме будет записано $\boldsymbol{X = -0{,}10111 \cdot 10^{11}}$, а в машине

$$\mathbf{1\ \ 10111\ \ \ 0\ \ \ \ 11},$$

где первая цифра кода **1** – знак числа. В данном примере число отрицательное; **10111** – мантисса числа. Представляет собой значение цифры числа; **0** – знак порядка. В данном примере порядок положительный; **11** – порядок числа.

Знак порядка соответствует направлению смещения запятой по отношению к старшему разряду мантиссы. Если запятая смещается влево, то знак порядка отрицательный, если вправо – то положительный.

Порядок числа определяет, на сколько разрядов должна быть перенесена запятая вправо или влево в зависимости от знака порядка. Перенесение запятой вправо на n разрядов означает умножение на 2^n, а влево – умножение на 2^{-n}.

Если в записи числа старшая цифра отлична от нуля, число считается *нормализованным* (например: $x = 0{,}11001 \cdot 10^{10}$); если старшая цифра 0 – число *ненормализованное* (например: $x = 0{,}011001 \cdot 10^{11}$). При проведении нормализации чисел следует осуществить сдвиг мантиссы числа влево, а порядок числа уменьшить на единицу.

Вся обработка чисел в ЭВМ производится автоматически, но так как для выполнения действий требуются операции отдельно с мантиссами чисел и отдельно с порядками, то это вызывает усложнение цифровых устройств и замедляет выпол нение операций. В то же время этот способ при заданной точности обеспечивает больший диапазон представления чисел, чем естественный.

Кодирование чисел в ЭВМ

Для изображения знака числа в ЭВМ принято минус обозначать двоичной единицей, а плюс двоичным нулём. Поэтому число $X = -0{,}01\ldots10$, имеющее отрицательный знак, в машине с фиксированной запятой изображается в следующем виде:

$$\boldsymbol{X = 1{,}01\ldots10}.$$

Прямой код числа позволяет дать изображение числа с учётом его знака. Поэтому прямой код положительного числа совпадает с его записью, а прямой код отрицательного числа отличается от обычной записи знаковым разрядом, в который заносится единица.

Пример.
Для чисел $X = 0{,}1101_2$ и $Y = -0{,}1001_2$
$[X]_{\text{пр}} = 0{,}1101_2$, $[Y]_{\text{пр}} = 1{,}1001_2$.

Для осуществления операции вычитания используются специальные коды отрицательных чисел: обратный, дополнительный, модифицированный обратный и модифицированный дополнительный, позволяющие операцию вычитания заменить операцией сложения.

Обратный код положительного числа совпадает с его прямым кодом.

$$[X]_{\text{пр}} = [X]_{\text{пр}} = 0{,}1101_2$$

Обратный код отрицательного числа образуется по следующему правилу: в знаковый разряд числа пишется единица, а значащие разряды числа изменяются на обратные величины, т.е. 0 на 1, а 1 на 0:

$$[Y]_{обр} = 1,0110_2.$$

Дополнительный код положительного числа совпадает с его прямым кодом:

$$[X]_{доп} = [X]_{пр} = 0,1101_2.$$

Дополнительный код отрицательного числа образуется по следующему правилу: в знаковый разряд числа пишется единица, в значащих разрядах единицы меняются на нули, а нули – на единицы и к младшему разряду числа прибавляется единица.

Иначе можно было бы сказать, что образуется обратный код числа, а затем к младшему разряду прибавляется единица.

Пример.

$$\begin{array}{r} [Y]_{доп} = 1,0110 \\ + \underline{\quad\quad 1} \\ 1,0111, \end{array}$$

т.е. $[Y]_{доп} = 1,0111$.

Модифицированные коды отличаются тем, что для изображения знака числа в них отводится два разряда. Так, положительный знак задаётся двумя нулями, а отрицательный – двумя единицами.

Число $X = -0,11001$ в обратном модифицированном коде изображается в следующем виде: $[X]^{м}_{обр} = 11,00110$. Положительное число $Y = 0,01101$ в модифицированном обратном коде: $[Y]^{м}_{обр} = = 00,01101$.

Пример.

Для чисел $X = +0,10010$, $Y = -0,01100$ модифицированный дополнительный код

$$[X]^{м}_{доп} = 00,10010;\ [Y]^{м}_{доп} = \begin{array}{r} 11,10011 \\ + \underline{\quad\quad 1} \\ 11,10100 \end{array}$$

Модифицированные коды чисел используются для выявления переполнения разрядной сетки машины, которое возникает при сложении чисел с одинаковыми знаками. Наибольшее распространение получили в ЭВМ обратный и дополнительный модифицированные коды.

Изменение значения каждого разряда числа на обратную величину в ЭВМ производится с помощью инверторов.

1.4. ВЫПОЛНЕНИЕ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ В МАШИНАХ С ФИКСИРОВАННОЙ И ПЛАВАЮЩЕЙ ЗАПЯТОЙ

Для осуществления действий над числовой информацией в ЭВМ можно использовать любое кодирование отрицательных чисел, что приводит к замене операций вычитания операциями сложения и к выполнению ряда элементарных действий и преобразований. Вначале рассмотрим порядок выполнения операций сложения чисел (только мантисс), представленных в виде правильной двоичной дроби, в машинах с фиксированной и плавающей запятой.

При сложении чисел в обратном коде все значащие разряды чисел складываются поразрядно справа налево, а знаковые разряды складываются как разряды целых чисел; образующаяся при этом в знаковом разряде единица переполнения прибавляется к младшему разряду суммы.

Последнее действие носит название циклического переноса.

Пример.

Сложение двух чисел в обратном коде

Числа $X = -0,10101$ и $Y = -0,00101$.

$$\begin{array}{rr} [X]_{обр} = & 1,01010 \\ & + \\ [Y]_{обр} = & \underline{1,11010} \\ & 11,00100 \\ & \underline{\quad\quad 1} \\ & 1,00101 \end{array}$$

$[X + Y]_{пр} = 1,11010$; $X + Y = -0,11010$.

Следовательно, в машине осуществлены следующие основные микрооперации:

1) образован обратный код первого слагаемого;
2) образован обратный код второго слагаемого;
3) произведено поразрядное сложение;
4) учтена единица переполнения – циклический перенос;
5) полученная сумма преобразована в прямой код.

Таким образом, потребовалось пять элементарных действий и преобразований для сложения двух отрицательных чисел.

При сложении чисел в дополнительном коде значащие разряды складываются поразрядно, знаковые разряды – как разряды целых чисел, а образующаяся в знаковом разряде единица переполнения теряется (*не учитывается*).

Пример.

Сложение двух чисел в дополнительном коде.

Числа $X = -0,10101$ и $Y = -0,00101$;

$$\begin{array}{rr} [X]_{доп} = & 1,01011 \\ & + \\ [Y]_{доп} = & \underline{1,11011} \\ & \underline{11,00110} \end{array}$$

$$\begin{array}{r} \_1{,}00110 \\ \underline{1} \end{array}$$

$[X+Y]_{обр} = 1{,}00101$; $[X+Y]_{пр} = 1{,}11010$; $X + Y = -0{,}11010$.

Итак, выполняется следующий набор микроопераций:

1) образуется дополнительный код первого слагаемого;
2) преобразуется второе слагаемое;
3) осуществляется поразрядное суммирование;
4) вычитается единица из младшего разряда суммы (в случае отрицательного результата);
5) производится обратное преобразо вание суммы чисел в прямой код.

Не всегда используется в машине весь набор микроопераций, как было показано на примерах; кроме того, перечисленные действия не отражают специфику работы узловых схем и устройств управления операций сложения.

В машине с плавающей запятой действие сложения и вычитания над числами в полулогарифмической форме сводится к следующим операциям и преобразованиям:

1) производится сравнение порядков чисел, если необходимо их выравнивание, а следовательно, и сдвиги мантисс;
2) далее выполняются операции сложения и вычитания над мантиссами чисел, как было показано выше, в обратном или дополнительном коде;
3) сумма, полученная в результате, нормализуется.

П р и м е р : $X = -0{,}010011 \cdot 10^{100}$, $Y = 0{,}001011 \cdot 10^{110}$.

Начинать сложение чисел X и Y сразу же нельзя, так как порядки у чисел разные, а следовательно, положения запятой у чисел неодинаковы. В нашем примере следует уменьшить на два порядок числа Y, а следовательно, сдвинуть мантиссу числа Y влево на два разряда, тем самым увеличив её значение:

$$Y = 0{,}101100 \cdot 10^{100};\ X = -0{,}010011 \cdot 10^{100}.$$

Суммируем мантиссы чисел в дополнительном коде.

Мантиссы	Дополнительный код	Сложение
$m_x = 1{,}010011$ + $m_y = \underline{0{,}101100}$ $m_x + m_y = 0{,}011001$ Результат $X + Y = 0{,}011001 \cdot 10^{100}$ Сдвиг $X + Y = 0{,}11001 \cdot 10^{11}$	1,101101 0,101100	1,101101 + $\underline{0{,}101100}$ 10,011001

Сдвигом влево мантиссы суммы на один разряд проведена нормализация, следовательно, порядок уменьшен на единицу.

Умножение чисел в ЭВМ с фиксированной запятой представляет собой многократное суммирование и сдвиг промежуточной суммы влево или вправо в зависимости от того, рассматривается множитель со старших или с младших разрядов.

Рассмотрим пример, где множитель рассматривается:

а) с младших разрядов;
б) со старших разрядов:

а) $x = 1011_2 = 11_{10}$ $y = 110_2 = 6_{10}$ $\sum_1 = 0000$ $\sum_2 = \frac{1011}{10110}$ $\sum_3 = \frac{1011}{1000010_2} = 66_{10}$	б) 1011 $\sum_1 = \frac{110}{1011}$ $\sum_2 = \frac{1011}{100001}$ $\sum_3 = \frac{0000}{1000010_2} = 66_{10}$

Независимо от способа организации умножения полученная промежуточная сумма каждый раз суммируется со сдвинутым на один разряд множимым.

В машине с плавающей запятой произведение двух чисел будет: $\Pi = m_1 m_2 q_1^p q_2^p$, так как $X_1 = m_1 q_1^p$, $Y_2 = m_2 q_2^p$.

Процесс умножения начинается в машине с определения знака произведения путём суммирования по модулю двух двоичных чисел, изображающих знак:

(+)·(+) = (+); 0 + 0 = 0;
(+)·(–) = (–); 0 + 1 = 1;
(–)·(+) = (–); 1 + 0 = 1;
(–)·(–) = (+); 1 + 1 = 1.

Затем находится **порядок произведения** путём суммирования порядков сомножителей и **мантисса произведения** – перемножением мантисс сомножителей. Рассмотрим четыре микрооперации на примере умножения чисел $A_1 = -0{,}1011 \cdot 10^{100}$, $A_2 = 0{,}110 \cdot 10^{11}$:

1) порядок произведения $P(П) = 100+11=111_2=7_{10}$;

2) знак произведения: $1+0 = 1$;

3) мантисса произведения $m(П)$:

$$
\begin{array}{r}
1011_2 = 11_{10} \\
\underline{110_2 = 6_{10}} \\
\Sigma_1 = 0000 \\
\underline{1011} \\
\Sigma_2 = 10110 \\
\underline{1011} \\
\Sigma_3 = 1000010_2 = 66_{10} \quad m(П) = 1{,}1000010;
\end{array}
$$

4) запись в машине

$A_1A_2 = П = 0 \;\; 111 \;\; 1 \;\; 1000010.$

Нормальная форма записи $A_1A_2 = П = -0{,}1000010 \cdot 10^{111}$.

Деление в машине с фиксированной запятой сводится к многократным вычитаниям и сдвигам, но так как вычитание в ЭВМ заменяется сложением в дополнительном или обратном коде, то процесс организации деления состоит из операций:

1) сложения делителя в обратном или дополнительном коде с делимым;

2) сдвига;

3) сложения делителя с остатками, предварительно сдвигаемыми на каждом шаге деления влево на один разряд.

Процесс деления может быть реализован двумя методами: с восстановлением остатка и без восстановления остатка. Второй метод требует меньших преобразований и элементарных действий, поэтому рассмотрим пример организации деления без восстановления остатка с использованием дополнительного кода. Знак частного определяется по такому же принципу, как и знак произведения.

Разряды частного находятся по правилу: если разность между делимым (*или очередным остатком*) *и делителем положительна, то в соответствующий разряд заносится 1*, *а если отрицательна – 0.*

Кроме того, для определения очередного разряда частного отрицательный остаток сдвигается на один разряд влево и к нему прибавляется делитель в прямом коде. Если остаток положительный, то осуществляется сдвиг, а затем к остатку прибавляется делитель в дополнительном коде.

Пример.

Деление чисел в дополнительном коде: $A_1 = 0{,}1100$ – делимое; $A_2 = 1{,}1110$ – делитель.

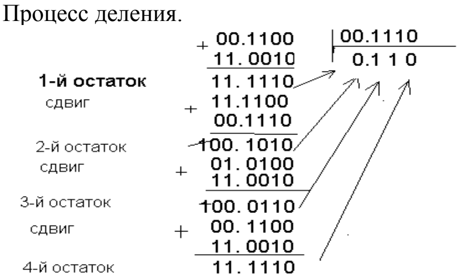
Знак частного отрицательный, т.е. $0 + 1 = 1$.

Прямой код делимого $[A_1]^{м}_{пр} = 00{,}1100$.

Прямой код делителя $[A_1]^{м}_{пр} = 00{,}1110$.

Дополнительный код делителя $[A_1]^{м}_{пр} = 11{,}0010$.

Процесс деления.



$A_1/A_2 = 0{,}110_2$.

Если продолжить деление до десяти значащих цифр, то проверка даст правильный результат.

В машине с плавающей запятой деление состоит из следующих действий и преобразований:

– определяется знак частного, находится порядок частного вычитанием порядка делителя из порядка делимого;
– производится деление мантисс;
– нормализуется частное.

В ряде ЭВМ процесс деления, как самый длительный, заменяется умножением на обратную величину, которая отыскивается в таблицах постоянной памяти.

2. ЛОГИЧЕСКИЕ ОСНОВЫ ПОСТРОЕНИЯ ЭВМ

2.1. ОСНОВЫ АЛГЕБРЫ ЛОГИКИ

Алгебра логики является математическим аппаратом, на основе которого строятся цифровые схемы или устройства, оперирующие с двоичной системой счисления.

Основой алгебры логики является высказывание, которое может быть истинным или ложным, но тем и другим одновременно быть не может, т.е. не предполагает неопределённостей. Любому высказыванию может быть присвоена некоторая переменная, которая может принимать одно из двух значений. Применительно к двоичным системам счисления это может быть 1 либо 0. Если высказывание истинно, то ему присваивается значение 1, если ложно – 0.

Поскольку есть переменные, то существуют некие функции (логические), которые в качестве аргументов имеют логические переменные (x, y, z,… или же x_1, x_2, x_3,…).

Логические функции могут быть как функциями одной переменой $F(x)$, так и функциями многих переменных $F(x_1, x_2, x_3, \ldots)$.

Логические функции ещё называются переключательными функциями.

2.1.1. Основные операции алгебры логики

В алгебре логики существует три операции:

1. Логическое сложение.
2. Логическое умножение.
3. Отрицание.

Логическое сложение

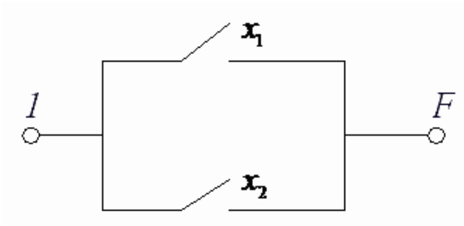
Дизъюнкция или операция ИЛИ.

Математическая запись: $F = x_1 + x_2$; $F = x_1 \vee x_2$.

Словесно звучит так: $F= x_1$ или x_2.

Еще эту операцию называют операцией разделения.

Электрическая модель этой операции:



Это значит, что логическая функция будет принимать значение 1, если любая из переменных равна 1.

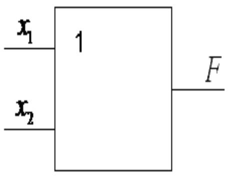
Операция ИЛИ может быть для любого количества переменных, но как минимум для двух.

Всевозможные состояния функции и возможные значения переменных сводятся в таблицу истинности.

Для двух переменных:

x_1	x_2	F
0	0	0
0	1	1
1	0	1
1	1	1

Условное обозначение логического элемента, реализующего операцию ИЛИ, для двух элементов:

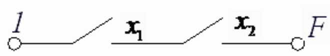


Сколько переменных, столько и входов.

Логическое умножение

Конъюнкция или операция И. При этой операции переключательная функция принимает значение 1 только тогда, когда все переменные будут равны 1.

Электрическая модель для двух переменных следующая:



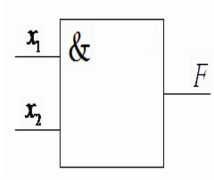
В операции конъюнкции может участвовать сколько угодно переменных.

Таблица истинности для двух переменных:

x_1	x_2	F
0	0	0
0	1	0
1	0	0

1	1	1

Условное обозначение логического элемента И для двух переменных:



Математическая запись: $F = x_1 x_2$; $F = x_1 \wedge x_2$.

Словесно звучит: $F = x_1$ и x_2.

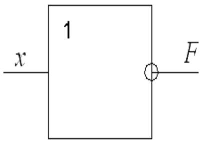
Логическое отрицание

Инверсия производится над одной переменой или над группой переменных.

Для одной переменой $F = \bar{x}$ («НЕ x»).

Для группы переменных $F = \overline{x_1 + x_2}$

Условное обозначение инвертора:



2.1.2. Основные теоремы алгебры логики

Для одной переменой и констант (0 или 1) существует девять теорем:

1. $x + 1 = 1$ 5. $x \cdot 1 = x$
2. $x + 0 = x$ 6. $x \cdot 0 = 0$
3. $x + x = x$ 7. $x \cdot x = x$
4. $x + \bar{x} = 1$ 8. $x \cdot \bar{x} = 0$

9. $\bar{\bar{x}} = x$ – закон двойной инверсии

Для группы переменных существуют следующие законы:

1. *Закон склеивания*

$$F = x_1 x_2 + \bar{x}_1 x_2 = x_2 (x_1 + \bar{x}_1) = x_2 \cdot 1 = x_2 .$$

Этот закон очень мощно работает для приведения переключательных функций к более простому виду.

Упрощение переключательных функций называется минимизацией, при этом логика переключательной функции не должна нарушаться.

2. *Закон инверсии де Моргана*

Этот закон позволяет перейти от операции логического умножения к операции логического сложения и наоборот. Он выражает принцип двойственности в алгебре логики. Закон справедлив для любого количества переменных, но не менее двух.

$$F = \overline{x_1 + x_2} = \bar{x}_1 \cdot \bar{x}_2$$

$$F = \overline{x_1 \cdot x_2} = \bar{x}_1 + \bar{x}_2$$

Применять этот закон можно всегда, если предварительно использовать закон двойной инверсии.

Пример: $F = x_1 + \bar{x}_2 + x_3$

$$F = \overline{\overline{x_1 + \bar{x}_2 + x_3}} = \overline{\bar{x}_1 \bar{\bar{x}}_2 \bar{x}_3} = \overline{\bar{x}_1 x_2 \bar{x}_3}$$

Примечание: последовательность операций в алгебре логики такая же, как и в обычной алгебре: сначала выполняется операция в скобках, затем выполняется операция логического умножения, затем логического сложения, затем инверсия над совокупностью переменных.

2.1.3. Способы задания переключательных функций

Логические функции могут быть заданы в трёх видах:

1. Словесно.
2. Таблично.
3. Алгебраически.

При задании переключательной функции необходимо описать все возможные состояния переменных и самой функции.

Пример. Пусть переключательная функция трёх переменных принимает значение 1, если хотя бы одна из переменных принимает значение 0. По словесному заданию делается табличное задание, т.е. составляется таблица истинности:

№ набора	x_1	x_2	x_3	F
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Переключательная функция имеет 8 наборов переменных.

Количество комбинаций (наборов) переменных определяется как 2^n, где *n* – количество переменных.
Каждая конкретная комбинация переменных называется **набором**.

В алгебре логики строго доказано, что функция от *n переменных будет иметь значения* 2^{2^n} .

Алгебраическое задание переключательной функции

По табличному заданию составляется алгебраический вид переключательной функции.
Алгебраический вид имеет две формы:
Первая форма – совершенная дизъюнктивная нормальная форма (СДНФ).

В этой форме переключательная функция представлена дизъюнкцией конъюнкций переменных каждого набора. Иначе говоря, переключательной функцией этой формы производится сложение каждого набора переменных, а каждый набор переменных – это их конъюнкция. Причём переменные будут представлены в прямом или инверсном виде. Если для данного набора переменных какая-либо переменная имеет значение 0, то она берется в инверсном виде, а если значение 1, то в прямом. Поэтому получается, что каждое слагаемое имеет значение 1. По-другому, каждое слагаемое называется конституентой единицы.

Из приведённой выше таблицы переключательная функция в СДНФ будет иметь вид:

$$F(x_3,x_2,x_1)=\bar{x}_3\bar{x}_2\bar{x}_1+\bar{x}_3\bar{x}_2x_1+\bar{x}_3x_2\bar{x}_1+\bar{x}_3x_2x_1+x_3\bar{x}_2\bar{x}_1+x_3\bar{x}_2x_1+x_3x_2\bar{x}_1\ .$$

В общем случае после упрощения логического выражения может быть получено логическое выражение, по которому составляется схема логического устройства (рис. 1).

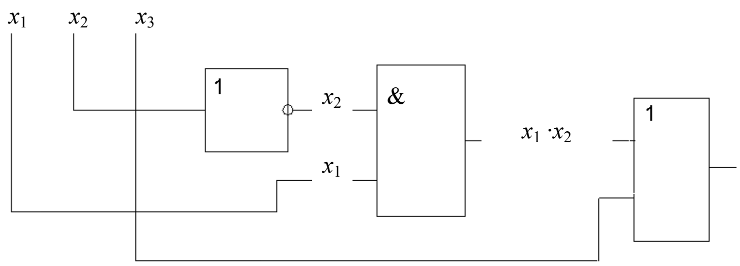


Рис. 1. Схема логического устройства, составленная по ДНФ

Пример:

$$F(x_3,x_2,x_1)=\bar{x}_2x_1+x_3\ .$$

Переключательная функция в СДНФ выписывается только для тех значений, на каком наборе переключательная функция должна принимать значения 1, поэтому в приведенной переключательной функции будет не 8 слагаемых, а 7.

Вторая форма – эта совершенная конъюнктивная нормальная форма (СКНФ).

В этой форме переключательная функция представлена конъюнкцией дизъюнкций переменных, т.е. каждый набор, который эквивалентен нулю, представлен в виде дизъюнкции слагаемых, а все нулевые наборы представлены конъюнкцией. Необходимо помнить о том, что те переменные, которые в наборе имеют значение 1, необходимо брать с инверсией. Внешний вид переключательной функции в СКНФ может быть таким:

$$F=(\bar{x}_3+x_2+\bar{x}_1)(x_3+x_2+\bar{x}_1)\ .$$

Структурная схема устройства приведена на рис. 2.

Примечание: в каждом конкретном случае форма переключательной функции выбирается из соображения максимальной простоты.

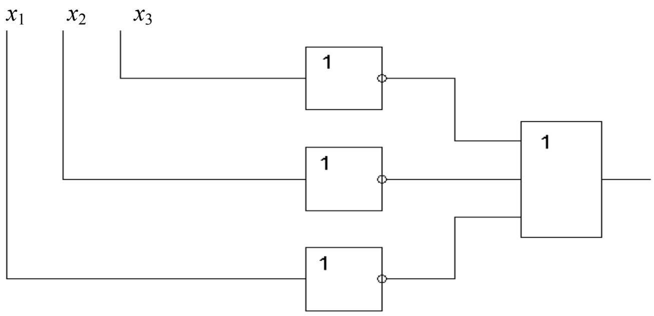


Рис. 2. Схема ЛУ, составленная в соответствии с СКНФ

Для ранее приведённого примера удобнее переключательную функцию брать в СКНФ.

$$F = \bar{x}_3 + \bar{x}_2 + \bar{x}_1 .$$

2.1.4. Минимизация переключательных функций

Если переключательную функцию реализовать в изначальном виде, то схема получается очень громоздкой, что ведёт к большим аппаратурным затратам, низкой надёжности. Поэтому всегда стремятся переключательную функцию минимизировать, т.е. привести её к минимальному виду (упростить), при этом логика работы не должна нарушаться.

Существует два способа минимизации:

– путём алгебраических преобразований;

– путём применения карт Карно (диаграмм Вейча).

Минимизация с помощью алгебраических преобразований. Здесь для приведения переключательной функции к минимальному виду применяется в основном теорема склеивания.

При минимизации случайной функции методом склеивания одно и то же слагаемое может участвовать несколько раз, так как согласно алгебре логики $x + x + x + ... + x = x$.

Пример:

Минимизировать методом непосредственных преобразований логическое выражение

$$\begin{aligned} F(x_3, x_2, x_1) &= \bar{x}_3\bar{x}_2\bar{x}_1 + \bar{x}_3\bar{x}_2 x_1 + \bar{x}_3 x_2\bar{x}_1 + \bar{x}_3 x_2 x_1 + x_3\bar{x}_2\bar{x}_1 + x_3\bar{x}_2 x_1 + \\ &+ x_3 x_2\bar{x}_1 = \bar{x}_3\bar{x}_2(\bar{x}_1 + x_1) + \bar{x}_3 x_2(\bar{x}_1 + x_1) + x_3\bar{x}_2(\bar{x}_1 + x_1) + \\ &+ x_3\bar{x}_1(\bar{x}_2 + x_2) = \bar{x}_3(\bar{x}_2 + x_2) + \bar{x}_2(\bar{x}_3 + x_3) + x_3\bar{x}_1 = \bar{x}_3 + \bar{x}_2 + \\ &+ x_3\bar{x}_1 = (\bar{x}_3 + x_3)(\bar{x}_3 + \bar{x}_1) + \bar{x}_2 = \bar{x}_3 + \bar{x}_2 + \bar{x}_1. \end{aligned}$$

Минимизация с помощью карт Карно. Карты Карно применяются, если 4, 5 и более переменных. Карты Карно представляют собой ТИ, разделённые на клетки. Число клеток в карте Карно определяется как 2^n, где n – число переменных. Каждой стороне карты присваивается своя переменная в прямом и инверсном виде. Переменные прямого и инверсного вида разделены линией. Каждая клетка карты Карно соответствует одному строго определённому набору переменных (рис. 3). При каждом переходе из одной клетки в другую вдоль строки или столбца изменяется значение только одной переменной. Следовательно, если в соседних клетках карты Карно будут стоять **0** или **1,** то над соответствующими членами канонической формы может быть проведена операция склеивания. В каждую клетку вносятся значения переключательной функции для данного набора.

Минимизация проводится путём склеивания так называемых соседних квадратов (клеток). Склеивание можно производить по 2 квадрата, по 4 и 8.

Если по 2 квадрата, то соседними будут являться рядом стоящие по горизонтали и по вертикали, или разделённые двумя квадратами по вертикали или по горизонтали.

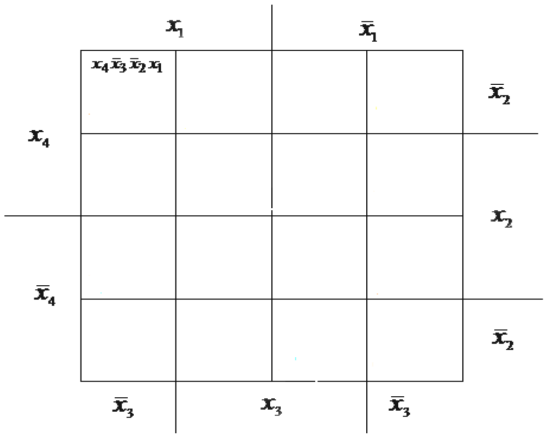


Рис. 3. Карта Карно для функции четырех переменных

По 4 – это 4 квадрата в строку или столбца; 4 квадрата, образующие квадрат; 4 квадрата по углам.

По 8 – 2 рядом лежащие строки или 2 рядом стоящие столбца, или 2 строки или 2 столбца, разделённые двумя строками или двумя столбцами.

В результате минимизации, т.е. в окончательную переключательную функцию выносятся переменные, которые были общими для склеиваемых квадратов.

Минимизация с использованием факультативных условий. В ряде случаев бывает так, что для каких-либо наборов переключательная функция неопределенна, иначе говоря, на этих наборах безразлично какое значение примет переключательная функция. Тогда при минимизации можно допустить, что на этих наборах переключательная функция примет значение 1 (если переключательная функция в СДНФ). Однако необходимо помнить, что это не всегда выгодно и в каждом конкретном случае факультативные условия можно принять индивидуально.

Рассмотрим это на примере.

Задача: построить логическое устройство, которое будет регистрировать 6 старших цифр десятичной системы счисления (4, 5, 6, 7, 8, 9).

Регистрировать – это когда на выходе должна фиксироваться 1 при поступлении любой из этих цифр.

Поскольку для регистрации самой старшей цифры 9 необходимо 4 разряда, то тогда ЛУ должно иметь 4 входа (x_3, x_2, x_1, x_0): каждый вход для своего разряда (рис. 4).

Полная таблица должна содержать 16 наборов, но 10 – 15 не нужны.

Если минимизировать переключательную функцию по карте Карно, то саму функцию в алгебраической форме представлять не обязательно (рис. 5).

Структурная схема установки приведена на рис. 6.

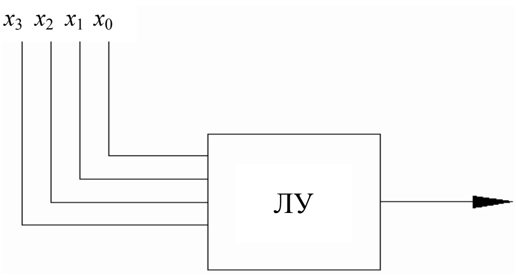


Рис. 4. Структура ЛУ

№ набора	x_3	x_2	x_1	x_0	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
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	1
9	1	0	0	1	1

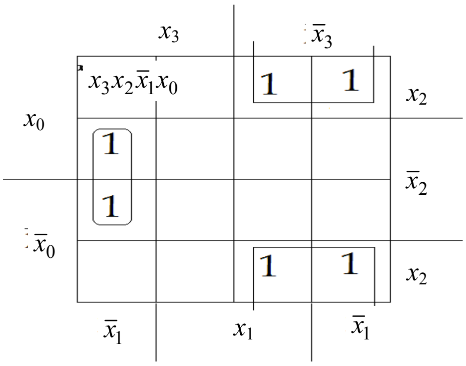


Рис. 5. Минимизация логической функции при помощи карты Карно

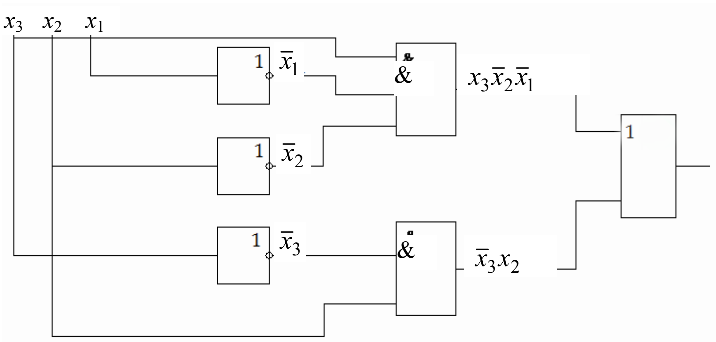


Рис. 6. Схема логического устройства, составленная по МДНФ

$F = \bar{x}_3x_2 + x_3\bar{x}_2\bar{x}_1$ – МДНФ логической функции.

Полученная схема реализует заданную логику, но её можно упростить, если применить факультативные условия: допустить, что на 10, 11, …, 15 наборах переключательная функция принимает значение 1. Тогда их надо внести в карты Карно (рис. 7).

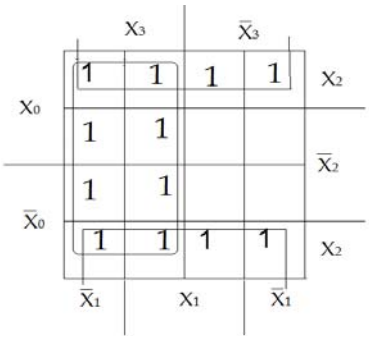


Рис. 7. Минимизация с применением факультативных условий

$F = x_3 + x_2$ – МДНФ логической функции, полученная по карте Карно.

Окончательная структурная схема устройства приведена на рис. 8.

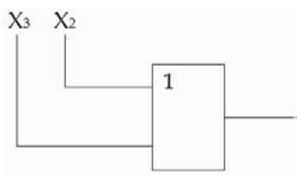


Рис. 8. Схема ЛУ

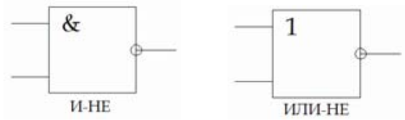
2.1.5. Приведение переключательной функции к единому базису

В цифровой технике в основном используются логические элементы вида: И-НЕ (отрицание конъюнкции), ИЛИ-НЕ (отрицание дизъюнкции).

Число входов может быть от 2 до 12. В соответствии с количеством входов эти элементы обозначаются так:

на 2 входа: 2И-НЕ, 2ИЛИ-НЕ.

Условно эти логические элементы на схемах обозначаются так:



Любую переключательную функцию можно реализовать в том или ином базисе логических элементов с любым количеством входов. Необходимо переключательную функцию определённым образом преобразовать. Преобразование сводится к применению закона двойного отрицания и закона инверсии де Моргана.

Пример: $F = \bar{x}_3 x_2 + x_3 \bar{x}_2 \bar{x}_1$.

*Реализация в базисе логических элементов **2И-НЕ**.*

1. Необходимо избавиться от логического сложения. Для этого над всей функцией применяется закон двойной инверсии, а далее закон де Моргана:

$$F = \overline{\overline{\bar{x}_3 x_2 + x_3 \bar{x}_2 \bar{x}_1}} = \overline{\overline{\bar{x}_3 x_2} \cdot \overline{x_3 \bar{x}_2 \bar{x}_1}} = \to$$

2. Необходимо объединить переменные, чтобы можно было применить двухвходовые логические элементы.

Пример: $\overline{x_3 \bar{x}_2 x_1} = \overline{x_3 \overline{\overline{\bar{x}_2 x_1}}}$.

Для инверсии одной переменой необходимо на один вход логического элемента подать переменную, а на остальные свободные входы подавать уровень логической единицы.

$$F = \overline{\overline{\bar{x}_3 x_2}\; \overline{x_3 \overline{\overline{\bar{x}_2 \bar{x}_1}}}}$$

Структурная схема приведена на рис. 9.

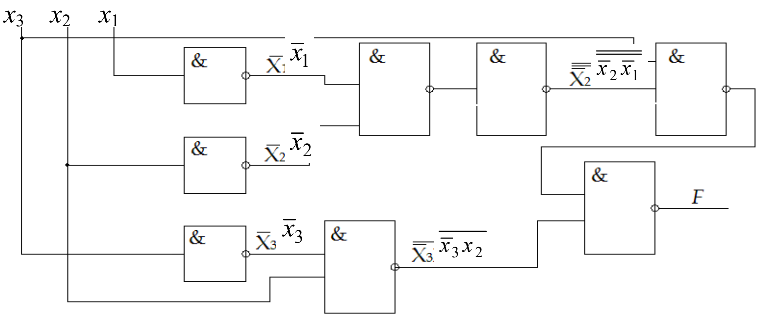


Рис. 9. Схема ЛУ в базисе 2И-НЕ

*Реализация в базисе логических элементов **2ИЛИ-НЕ.***

1. Необходимо в переключательной функции избавиться от операции логического сложения. Для этого над отдельными слагаемыми, где действует конъюнкция, необходимо применить закон двойной инверсии и закон де Моргана.

$$F = \bar{x}_3 x_2 + x_3 \bar{x}_2 \bar{x}_1 = \overline{\overline{\bar{x}_3 x_2}} + \overline{\overline{x_3 \bar{x}_2 \bar{x}_1}} = \overline{\bar{\bar{x}}_3 + \bar{x}_2} + \overline{\bar{x}_3 + \bar{\bar{x}}_2 + \bar{\bar{x}}_1} =$$
$$= \overline{x_3 + \bar{x}_2} + \overline{\bar{x}_3 + x_2 + x_1} = \to$$

2. Проделываем операцию для объединения переменных по две, чтобы реализовать в базисе логических элементов 2ИЛИ-НЕ.

$$\to = \overline{\overline{\overline{x_3 + \bar{x}_2} + \overline{\bar{x}_3 + \overline{\overline{x_2 + x_1}}}}}.$$

Структурная схема приведена на рис. 10.

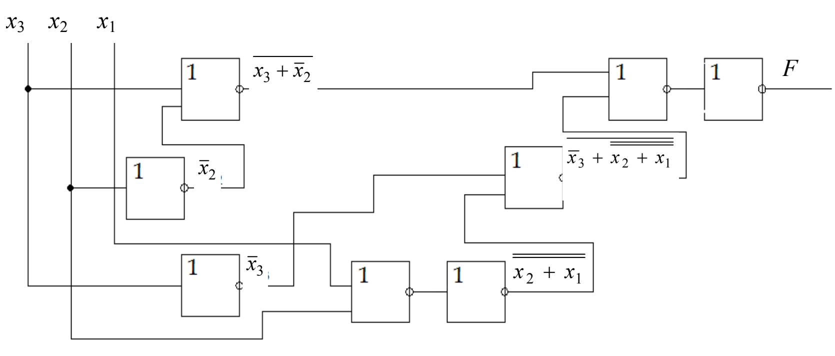


Рис. 10. Схема ЛУ в базисе 2ИЛИ-НЕ

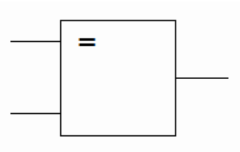
2.1.6. Простейшие комбинационные логические схемы

Рассмотрим логические функции двух переменных.

Равнозначность

Операция равнозначность реализует на выходе логического элемента 1 при равенстве переменных на входах.

x_1	x_2	F
0	0	1
0	1	0
1	0	0
1	1	1



равнозначность

$F = \bar{x}_1\bar{x}_2 + x_1x_2 = \rightarrow$

К первому слагаемому применим закон де Моргана:

$\rightarrow = \overline{x_1 + x_2} + x_1x_2$.

Структурная схема приведена на рис. 11.

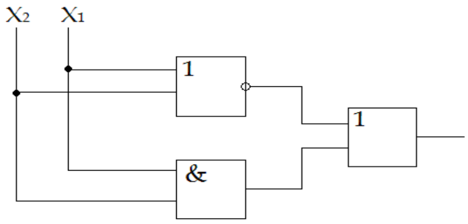


Рис. 11. Структурная схема логики «Равнозначность»

Неравнозначность (отрицание равнозначности)

Операция неравнозначность противоположна операции равнозначности, т.е. переключательная функция примет значение 1 только тогда, когда значения переменных на входе неравны.

x_1	x_2	F
0	0	0
0	1	1
1	0	1
1	1	0

$$F = \bar{x}_1x_2 + x_1\bar{x}_2 .$$

Структурная схема приведена на рис. 12.

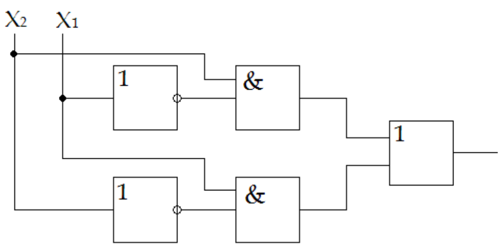
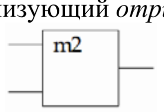
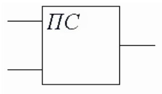


Рис. 12. Схема, реализующая логическую неравнозначность

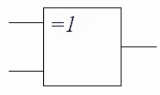
Логический элемент, реализующий *отрицание равнозначности*, носит несколько названий и обозначений:

1. Сумматор по модулю 2 

2. Полусумматор



3. Исключающее ИЛИ



И м п л и к а т о р

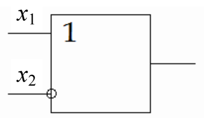
Логический элемент, реализующий операцию импликации для двух входов, имеет следующую логику работы: на выходе импликатора логический 0 будет только тогда, когда информационный вход невозбужден, т.е. у импликатора 2 различных входа: информационный (прямой) и неинформационный (инверсный).

Возбуждён вход – действует 1, а не возбуждён – действует 0.

Здесь возможно два варианта: информационный вход для x_1 и информационный вход для x_2. Выберем один из них, например, информационный x_1, а не информационный x_2, тогда таблица истинности будет иметь следующий вид:

x_1	x_2	F
0	0	1
0	1	0
1	0	1
1	1	1

Условное обозначение:



СКНФ: $F = x_1 + \bar{x}_2$.

Структурная схема, реализующая операцию импликации, приведена на рис. 13.

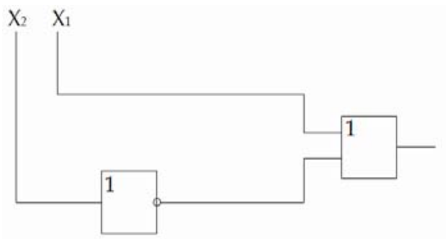


Рис. 13. Схема, реализующая импликацию

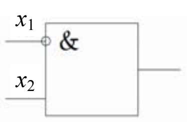
О т р и ц а н и е   и м п л и к а ц и и   (з а п р е т)

Операция отрицания импликации противоположна импликации, еще называется операция запрет.

Также имеет информационный и запрещающий вход, тогда таблица истинности будет иметь следующий вид:

x_1	x_2	F
0	0	0
0	1	1
1	0	0
1	1	0

Условное обозначение:



СДНФ: $F = \bar{x}_1 x_2$.

Структурная схема, реализующая операцию отрицания импликации, приведена на рис. 14.

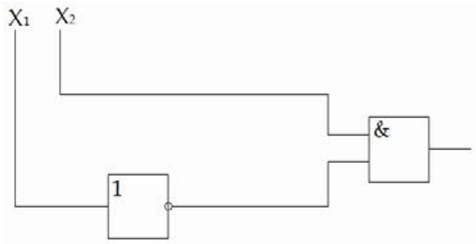


Рис. 14. Схема, реализующая отрицание импликации

Наименования и обозначение булевых функций приведено в приложении Б.

Приложение А
(справочное)

А.1. Числа в различных системах счисления

Десятичная СС	Двоичная	Восьмеричная	Шестнадцатеричная	Двоично-кодированная десятичная
00	00000	00	00	0000. 0000
01	00001	01	01	0000. 0001
02	00010	02	02	0000. 0010
03	00011	03	03	0000. 0011
04	00100	04	04	0000. 0100
05	00101	05	05	0000. 0101
06	00110	06	06	0000. 0110
07	00111	07	07	0000. 0111
08	01000	10	08	0000. 1000
09	01001	11	09	0000. 1001
10	01010	12	0*A*	0001. 0000
11	01011	13	0B	0001. 0001
12	01100	14	0C	0001. 0010
13	01101	15	0D	0001. 0011
14	01110	16	0E	0001. 0100
15	01111	17	0F	0001. 0101
16	10000	20	10	0001. 0110
17	10001	21	11	0001. 0111
18	10010	22	12	0001. 1000
19	10011	23	13	0001. 1001
20	10100	24	14	0010. 0000
21	10101	25	15	0010. 0001
22	10110	26	16	0010. 0010
23	10111	27	17	0010. 0011
24	11000	30	18	0010. 0100
25	11001	31	19	0010. 0101
26	11010	32	1*A*	0010. 0110
27	11011	33	1B	0010. 0111
28	11100	34	1C	0010. 1000
29	11101	35	1D	0010. 1001
30	11110	36	1E	0011. 0000
31	11111	37	1F	0011. 0001

П р и м е ч а н и е . Точка означает разделение тетрад.

Приложение Б
(справочное)

Б.1. Наименование и обозначение булевых функций

Функция	Значение функции	Наименование функции	Название или обозначение схемы логического элемента
$F_0(x, y)$	0	Константа нуля	Генератор нуля
$F_1(x, y)$	$X \wedge Y$	Конъюнкция, логическое умножение, И	Конъюнктор, И, &
$F_2(x, y)$	$\overline{X} \wedge Y$	Запрет по x, отрицание импликации	Схема запрета
$F_3(x, y)$	X	Переменная x	Повторитель x
$F_4(x, y)$	$X \wedge \overline{Y}$	Запрет по y, отрицание импликации	Схема запрета
$F_5(x, y)$	Y	Переменная y	Повторитель y
$F_6(x, y)$	$X \vee Y$	Сумма по модулю 2, логическая неравнозначность	Сложение по модулю 2, М2
$F_7(x, y)$	$X \vee Y$	Дизъюнкция, логическое сложение, ИЛИ	Дизъюнктор, ИЛИ
$F_8(x, y)$	$\overline{X \vee Y}$ $X \downarrow Y$	Стрелка Пирса, отрицание дизъюнкции	Элемент Пирса, ИЛИ-НЕ
$F_9(x, y)$	$X \equiv Y$	Эквивалентность	Равнозначность
$F_{10}(x, y)$	$\overline{Y}$	Отрицание, инверсия y	Инвертор, НЕ
$F_{11}(x, y)$	$Y \to X$	Импликация от y к x	Элемент импликации
$F_{12}(x, y)$	$\overline{X}$	Отрицание, инверсия x	Инвертор, НЕ
$F_{13}(x, y)$	$X \to Y$	Импликация от x к y	Элемент импликации
$F_{14}(x, y)$	X/Y $\overline{X \wedge Y}$	Штрих Шеффера, отрицание конъюнкции	Элемент Шеффера, И-НЕ
$F_{15}(x, y)$	1	Константа единицы	Генератор единицы

Б.2. Значения булевых функций

Значение аргументов		Значение Булевых функций							
x	Y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

Продолжение табл. Б.2

Значение аргументов		Значение Булевых функций							
x	Y	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Агаханян, Т.М. Интегральные микросхемы / Т.М. Агаханян. – М. : Эергоатомиздат, 1983. – 464 с.
2. Алексенко, А.Г. Микросхемотехника / А.Г. Алексенко, А.А. Шагурин ; под ред. И.П. Степаненко. – М. : Радио и связь, 1982. – 416 с.
3. Аналоговые и цифровые интегральные микросхемы : справочное пособие / С.В. Якубовский, Н.А. Барканов, Л.И. Ниссельсон и др. ; под ред. СВ. Якубовского. – 2-е изд., перераб. и доп. – М. : Радио и связь, 1984. – 432 с.
4. Банк, М.У. Аналоговые интегральные схемы в радиоаппаратуре / М.У. Банк. – М. : Радио и связь, 1981. – 136 с.
5. Вениаминов, В.И. Микросхемы и их применение : справочное пособие / В.И. Вениаминов, О.Н. Лебедев, А.М. Мирошниченко. – 3-е изд., перераб. и доп. – М. : Радио и связь, 1989. – 240 с.
6. Волгин, Л.И. Аналоговые операционные преобразователи для измерительных приборов и систем / Л.И. Волгин. – М. : Энергоатомиздат, 1983. – 208 с.
7. Воробьёв, Н.И. Проектирование электронных устройств / Н.И. Воробьёв. – М. : Высшая школа, 1989. – 223 с.
8. Вуколов, Н.И. Знакосинтезирующие индикаторы : справочник / Н.И. Вуколов, А.Н. Михайлов ; под ред. В.П. Балашова. – М. : Радио и связь, 1987. – 576 с.
9. Гусев, В.Г. Электроника / В.Г. Гусев, Ю.М. Гусев. – 2-е изд., перераб. – М. : Высшая школа, 1991. – 622 с.
10. Гутников, В.С. Интегральная электроника в измерительных устройствах / В.С. Гутников. – 2-е изд., перераб. и доп. – Л. : Энергоатомиздат, 1988.– 304 с.
11. Зельдин, Е.А. Цифровые интегральные микросхемы в информационно-измерительной аппаратуре / Е.А. Зельдин. – Л. : Энергоатомиздат, 1986. – 280 с.
12. Интегральные микросхемы : справочник / Б.В. Тарабрин, Л.Ф. Лушин, Ю.Н. Смирнов и др. ; под ред. Б.В. Тарабрина. – М. : Радио и связь, 1984. – 528 с.
13. Левинзон, С.В. Защита в источниках питания РЭА / С.В. Левинзон. – М. : Радио и связь, 1990. – 140 с.
14. Миловзоров, В.П. Элементы информационных систем / В.П. Миловзоров. – М. : Высшая школа, 1989. – 440 с.
15. Моин, В.С. Стабилизированные транзисторные преобразователи / В.С. Моин. – М. : Энергия, 1986. – 156 с.
16. Носов, Ю.Р. Оптроны и их применение / Ю.Р. Носов, А.С. Сидоров. – М. : Радио и связь, 1981. – 280 с.
17. Потёмкин, И.С. Функциональные узлы цифровой автоматики / И.С. Потёмкин. – М. : Энергоатомиздат, 1988. – 320 с.
18. Применение интегральных микросхем в электронной вычислительной технике : справочник / Р.В. Данилов, С.А. Ельцов, Ю.П. Иванов и др.; Под ред. Б.Н. Файзулаева, Б.В. Тарабрина. – М. : Радио и связь, 1987. – 384 с.
19. Применение прецизионных аналоговых микросхем / А.Г. Алексенко, Е.А. Коломбет, Г.И. Стародуб. – 2-е изд., перераб. и доп. – М. : Радио и связь, 1985. – 256 с.
20. Расчёт электронных схем : учеб. пособие для вузов / Г.И. Изъюрова, Г.В. Королёв, В.А. Терехов и др. – М. : Высшая школа, 1987. – 335 с.
21. Секлоф, С. Аналоговые интегральные схемы / С. Секлоф ; пер. с англ. – М. : Мир, 1988. – 583 с.
22. Степаненко, И.П. Основы теории транзисторов и транзисторных схем / И.П. Степаненко. – 2-е изд., перераб. и доп. – М. : Энергия, 1973. – 608 с.
23. Тимонтеев, В.Н. Аналоговые перемножители сигналов в радиоэлектронной аппаратуре / В.Н. Тимонтеев, Л.М. Величко, В.А. Ткаченко. – М. : Радио и связь, 1982. – 112 с.
24. Титце, У. Полупроводниковая схемотехника : справочное руководство ; пер. с нем. / У. Титце, К. Шенк. – М. : Мир, 1982. – 512 с.
25. Токкейм, Р. Основы цифровой электроники : пер. с англ. / Р. Токкейм. – М. : Мир, 1988. – 392 с ; ил.
26. Фолкенберри, Л. Применение операционных усилителей и линейных ИС / Л. Фолкенберри ; пер. с англ. – М. : Мир, 1985. – 572 с.
27. Цифровые и аналоговые интегральные микросхемы : справочник / С.В. Якубович, Л.И. Ниссельсон, В.И. Кулешова и др. – М. : Радио и связь, 1990. – 496 с.
28. Цытович, Л.И. Элементы аналоговой и цифровой электроники в автоматизированном электроприводе : учебник для вузов / Л.И. Цытович. – Челябинск : Южно-уральский го сударственный университет, 2001. – 480 с.
29. Шило, В.Л. Популярные цифровые микросхемы / В.Л. Шило. – М. : Металлургия, 1988. – 352 с.
30. Шкритек, П. Справочное руководство по звуковой схемотехнике / П. Шкритек ; пер. с нем. – М. : Мир, 1991. – 446 с.
31. Электронные приборы для отображения информации / Ю.А. Быстрое, И.И. Литвак, Г.М. Перминов. – М. : Радио и связь, 1985. – 240 с.
32. Электропитание устройств связи : учебник для вузов / под ред. Ю.Д. Козляева. – М. : Радио и связь, 1998. – 328 с.
33. Лобанов, В.И. Азбука разработчика цифровых устройств / В.И. Лобанов. – М. : Горячая линия-Телеком, 2001.