Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра программного обеспечения информационных технологий

А.Т. Пешков

Организация и функционирование ЭВМ

Методическое пособие для студентов специальности
«Программное обеспечение информационных технологий»
дневной формы обучения

В 3-х частях

Часть 2

Логические основы ЭВМ

Минск 2005

УДК 004(075.8)
ББК 32.973 я73
П 31

Р е ц е н з е н т:
доцент кафедры ИИТ БГУИР, канд. техн. наук В.Н. Мухаметов

Пешков А.Т.

П 31 Организация и функционирование ЭВМ: Метод. пособие для студентов спец. «Программное обеспечение информационных технологий» дневной формы обуч. В 3ч.Ч. 2: Логические основы ЭВМ/ А.Т. Пешков – Мн.: БГУИР, 2005. – 36 с.: ил.

ISBN 985-444-857-6 (ч.2)

В пособии излагается материал, связанный с арифметическими, логическими и схемотехническими вопросами построения ЭВМ. Приведено большое количество иллюстраций, таблиц и примеров с решениями, что способствует успешному усвоению излагаемого материала.

УДК 004(075.8)
ББК 32.973 я 73

Часть 1 издана в БГУИР в 2004 г.

ISBN 985-444-857-6 (ч.2)
ISBN 985-444-601-8

Содержание

1. Логические основы ЭВМ

1.1. Основные понятия алгебры логики

Алгебра логики находит широкое применение при синтезе и анализе схем ЭВМ. Это объясняется, с одной стороны, соответствием представления переменных и функций алгебры логики, с другой стороны, двоичным представлением информации и характером работы отдельных компонентов вычислительной техники, которые могут пропускать или не пропускать ток, иметь на выходе высокий или низкий уровень сигнала (напряжения или тока).

Основные понятия алгебры логики:

–*логическая переменная* - это такая переменная, которая может принимать одно из двух значений: истинно или ложно (да или нет, единица или ноль);

–*логическая константа* - это такая постоянная величина, значением которой может быть одно из двух значений: истинно или ложно (да или нет, единица или ноль);

–*логическая функция* - это такая функция, которая может принимать одно из двух значений (истинно или ложно, да или нет, единица или ноль), в зависимости от текущего значений её аргументов, в качестве которых используются логические переменные.

Логическая функция может быть одного (n=1) или нескольких (n >1) аргументов. Значение логической функции определяется комбинацией конкретных значений переменных, от которых она зависит. Комбинация конкретных значений переменных (аргументов функции) называется набором. Проведя аналогию с двоичным кодом, легко убедиться, что количество различных наборов N для n переменных определяется как $N= 2^n$.

Зависимость логической функции от переменных может задаваться поразному. Это может быть задание функции в виде таблицы истинности, или словесное описание, или задание её в виде логического выражения.

Словесное описание, как правило, может использоваться в случае сравнительно несложной логической функции.

Таблица истинности является универсальным средством задания логической функции. Она включает все наборы для заданного количества переменных, определяющих значение логической функции, с указанием значений, которые принимает функция для каждого набора. В одной таблице истинности может задаваться несколько логических функций, зависящих от одних и тех же переменных. Таблица истинности для нескольких функций y_i трех переменных x_1, x_2, x_3 может быть задана следующим образом, табл.1.1.

Таблица 1.1

№ п.п.	x_1	x_1	x_3	y_1	y_2	y_3	.....	y_n
0	0	0	0	0	1	1		0
1	0	0	1	1	1	0		1
2	0	1	0	1	1	1		0
3	0	1	1	0	1	0		-
4	1	0	0	1	0	1		0
5	1	0	1	0	0	0		1
6	1	1	0	0	0	1		-
7	1	1	1	1	1	0		1

В приведенной таблице истинности во второй, третей и четвертой колонках, помеченных соответственно x_1, x_2, x_3, даны все возможные наборы этих переменных. В следующих колонках приводятся значения функций y_1, y_2, y_n для каждого набора.

Логическая функция называется «*полностью определенная*», если для неё заданы значения по всем возможным наборам. Функция называется «*частично определенная*», если для некоторых наборов значения функции не заданы. В приведенной таблице истинности функции y_1, y_2, y_3 являются полностью определенными, а функция y_n – частично определенная (знак «-» означает неопределенность значения функции).

Максимальное количество полностью определенных функций от n переменных определяется как $M = (2^2)^n$.

Логическим выражением называется комбинация логических переменных и констант, связанных элементарными базовыми логическими функциями (или логическими операциями), которые могут разделяться скобками.

Например, логическую функцию y_1, определенную в вышеприведенной таблице истинности, можно представить в виде логического выражения

$y1 = \overline{(x1 \bullet x2 + x1 \bullet x3 + x2 \bullet x3)} \bullet (x1 + x2 + x3) + x1 \bullet x2 \bullet x3,$ где, “+”, “$\bullet$”, а также верхнее подчеркивание - знаки базовых логических функций.

Набор элементарных логических операций, с помощью которых можно задать любую, сколь угодно сложную логическую функцию, называется «функционально полная система логических функций». Иногда такую систему называют базисом.

В качестве элементарных логических функций функционально полных систем логических функций используются функции одной или двух логических переменных.

Все возможные функции одной переменой приведены в табл.1.2.

Таблица 1.2

№ п.п.	y_0	y_1	y_2	y_3
0	0	0	1	1
1	0	1	0	1

Из таблицы видно, что

y_0=0 – константа;

y_1=x – равна значению переменной;

y_2 – равна значению, обратному значению переменной x;

y_3=1 – константа.

С точки зрения базовых функций интерес представляет только функция y_2, она называется функцией отрицания, читается как «не x » и обозначается как "$\bar{x}$", т.е. можно записать

$y2 =$"$\bar{x}$".

Все возможные функции двух переменных приведены в табл.1.3.

Таблица 1.3

N стр.	x_1	x_2	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	1	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	1
2	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
3	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Информация о функциях двух переменных приведена в табл.1.4.

Таблица 1.4

y_i	Название функции	Чтение функции	Запись в виде булева выражения
1	2	3	4
y_0	Const «0»		0
y_1	Конъюнкция	И x_1, и x_2	$x_1 \cdot x_2$; x_1x_2; $x_1 \vee x_2$
y_2	Запрет по x_2	Неверно, что если x_1, то x_2	$x_1 \bar{x}_2$
y_3	$f(x_1)$	Функция одной переменной	x_1
y_4	Запрет по x_1	Неверно, что если x_2 , то x_1	$\bar{x}_1 x_2$
y_5	$f(x_2)$	Функция одной переменной	x_2
y_6	Неравнозначности	x_1 не равно x_2	$\bar{x}_1 x_2 + x_1 \bar{x}2$
y_7	Дизъюнкция	Или x_1, или x_2	$x_1 + x_2$
y_8	Функция Пирса	Ни x_1, ни x_2	$\overline{x_1 + x_2}$

Окончание табл. 1.4

1	2	3	4
y_i	Название функции	Чтение функции	Запись в виде булева выражения
y_9	Равнозначности	x_1 равно x_2	$\bar{x}_1\bar{x}_2 + x_1x_2$
y_{10}	$f(x_2)$	Функция одной переменной	$\bar{x}_2$
y_{11}	Импликация	Если x_2 , то x_1	$\bar{x}_1\bar{x}_2 + x_1$
y_{12}	$f(x_2)$	Функция одной переменной	$\bar{x}_1$
y_{13}	Импликация	Если x_1 , то x_2	$\bar{x}_1\bar{x}_2 + x_2$
y_{14}	Шеффера	Неверно, что и x_1, и x_2	$\overline{x_1x_2}$
y_{15}	Const (=1)		1

Наиболее распространенной в алгебре логики является функционально полная система логических функций, которая в качестве базовых логических функций использует функцию одной переменной НЕ (функция отрицания), и две функции двух переменных - И (конъюнкция, или логическое умножение) и ИЛИ (дизъюнкция, или логическое сложение). Эта система получила название «система *булевых* функций», или *булевый базис*. В алгебре логики имеется целый раздел «*Алгебра Буля*»), посвященный этому базису.

В вышеприведенной таблице, описывающей функции от двух переменных, в последней колонке даны варианты записи этих функций в булевом базисе.

1.2. Элементы алгебры Буля

1.2.1. Законы и правила алгебры Буля

В алгебре Буля логические выражения включают логические операции И, ИЛИ, НЕ, которые могут быть использованы в самых различных сочетаниях. При оценке значения такого выражения необходимо решить это выражение для конкретного набора переменных. В алгебре Буля используется следующая приоритетность выполнения операций: сначала рассчитываются значения имеющих место отрицаний и скобок, затем выполняется операция И (логическое умножение); самый низший приоритет имеет операция ИЛИ (логическая сумма).

При работе с булевыми логическим выражениями используются следующие законы и правила.

Переместительный (коммутативный) закон. Закон справедлив как для конъюнкции, так и для дизъюнкции.

$x_1 + x_2 + x_3 + x_4 = x_4 + x_3 + x_2 + x_1$ – от перемены мест логических слагаемых сумма не меняется.

$x_1\ x_2\ x_3\ x_4 = x_4\ x_3\ x_2\ x_1$ – от перемены мест логических сомножителей их произведение не меняется.

Этот закон справедлив для любого количества логических операндов.

Сочетательный (ассоциативный) закон. Закон справедлив как для конъюнкции, так и для дизъюнкции.

$x_1 + x_2 + x_3 + x_4 = (x_2 + x_3) + x_1 + x_4 = (x_1 + x_4) + (x_2 + x_3)$ – при логическом сложении отдельные слагаемые можно заменить их суммой.

$x_1\ x_2\ x_3\ x_4 = (x_2\ x_3)\ x_1 x_4 = (x_1\ x_4)\ (x_2\ x_3)$ – при логическом умножении отдельные логические сомножители можно заменить их произведением.

Распределительный (дистрибутивный) закон.

$(x_1 + x_2)\ x_3 = x_1\ x_3 + x_2\ x_3$.

$(x_1 + x_2)\ (x_1 + x_3) = x_1 + x_2\ x_3$.

Правило Де Моргана:

$\overline{x1 + x2} = \bar{x}1\bar{x}2$ –отрицание суммы равно произведению отрицаний,

$\overline{x1x2} = \bar{x}1 + \bar{x}2$ –отрицание произведения равно сумме отрицаний.

Операция склеивания:

$\bar{x}iA + xiA = A$ - операция *склеивания* для конъюнкций, где А - переменная или любое логическое выражение.

$(\bar{x}i + A)(xi + A) = A$ – операция склеивания для дизъюнкций.

Если в качестве А используется простая конъюнкция, т.е. конъюнкция, представляющая собой логическое произведение переменных и их отрицаний, то имеет место

$\bar{x}1x2\bar{x}3\bar{x}4x5\bar{x}6 + \bar{x}1x2\bar{x}3\bar{x}4x5\bar{x}6 = \bar{x}1x3\bar{x}4x5\bar{x}6.$

Как видно, в результирующем выражении количество переменных на единицу меньше, чем в склеенных конъюнкциях. Количество переменных в простой конъюнкции называется *рангом конъюнкции*, т.е. операция склеивания, примененная к простым конъюнкциям, дает результат с рангом, на единицу меньшим ранга исходных конъюнкций.

Операции с отрицаниями:

$\bar{\bar{x}} = x$ – двойное отрицание равносильно отсутствию отрицания.

$\bar{x} \bullet x = 0$

$\bar{x} + x = 1$

Операции с константами:

$x_1 + 1 = 1,\quad x_1 + 0 = x_1,$

$x_1\ 1 = x_1,\ x_1\ 0 = 0.$

Операции с одинаковыми операндами:

$x_1 + x_1 + x_1 + x_1 + \ldots + x_1 = x_1$,

$x_1\ x_1\ x_1 \ldots x_1 = x_1$ при любом числе повторений.

Законы и правила алгебры *Буля* могут быть доказаны путем логического рассуждения, однако такое доказательство применимо только для простейших случаев. Доказать справедливость того или иного правила можно, если с помощью различных преобразований привести правую часть правила к выражению в левой части (или наоборот). Универсальным приемом доказательства является использование таблицы истинности. Это основано на том утверждении, что

два выражения (правая и левая части правила или закона) *эквивалентны*, если они принимают одинаковые значения на всех наборах логических переменных.

Например, правило двойного отрицания, которое справедливо не только относительно одной переменной, но и любого логического выражения, можно доказать следующим рассуждением: если неверно утверждение, что выражение ложно, то очевидно утверждение, что это выражение истинно.

Доказать справедливость распределительного закона в интерпретации выражением

$(x_1 + x_2)(x_1 + x_3)= x_1 + x_2 x_3$

можно за счет приведения левой части к выражению правой части, раскрыв скобки:

$(x_1 + x_2 *(x_1 + x_3)= x_1 x_2 + x_1 x_1 + x_1 x_3 . + x_2 x_3 = x_1 x_2 + x_1 + x_1 x_3 . + x_2 x_3 =$

$= x_1 (x_2 + 1 + x_3) + x_2 x_3 =$

помня, что логическая сумма с одним слагаемым, равным константе 1, равна 1, можно записать

$= x_1 + x_2 x_3$.

Используем таблицу истинности для доказательства правила Де Моргана в варианте

$\overline{x1 + x2} = \bar{x}1\bar{x}2$ - отрицание суммы равно произведению отрицаний.

Составим таблицу истинности для правой и левой частей и составляющих их функций, табл.1.5.

Таблица 1.5

x_1	x_2	$x_1 + x_2$	$\overline{x_1 + x_2}$	$\overline{x_1}$	$\overline{x_2}$	$\overline{x_1 x_2}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Из таблицы истинности видно, что правая и левая части доказываемого правила принимают одинаковые значения на всех наборах, следовательно, они эквивалентны.

1.2.2. Формы представления логических функций

Одну и ту же логическую функцию можно представить различными логическими выражениями. Среди множества выражений, которыми представляется логическая функция, особое место занимают две канонические формы:

совершенная конъюнктивная нормальная форма (СКНФ),

совершенная дизъюнктивная нормальная форма (СДНФ).

Совершенная дизъюнктивная нормальная форма представляет собой дизъюнкцию простых конъюнкций, где под термином простая *конъюнкция* имеется в виду конъюнкция переменных или их отрицаний. В СДНФ простые конъюнк-

ции содержат все переменные в своей прямой или инверсной форме и отражают собой наборы, на которых представляемая функция имеет единичное значение. Такие конъюнкции называются конституентами единицы рассматриваемой функции. Поэтому СДНФ представляет собой дизъюнкцию (логическую сумму), слагаемыми которой являются *конституенты единицы*. Общая запись СДНФ функции y имеет вид

$$y = \vee\ x_1^{\delta 1} x_2^{\delta 2} x_3^{\delta 3} \ldots x_{(n-1)}^{\delta(n-1)} x_n^{\delta n},$$

где $$x_i^{\delta\iota} = \begin{cases} x_i, \text{ если } \delta\iota = 1, \\ \overline{x_i}, \text{ если } \delta\iota = 0. \end{cases}$$

СДНФ легко сформировать на основе таблицы истинности. Например, если функции задаются в виде табл.1.6, то СДНФ для них будет иметь следующий вид:

$$y1 = \bar{x}1\bar{x}2\bar{x}3 + \bar{x}1\bar{x}2x3 + \bar{x}1x2x3 + x1x2\bar{x}3 + x1x2x3;$$

$$y2 = \bar{x}1\bar{x}2\bar{x}3 + \bar{x}1x2x3;$$

$$y3 = \bar{x}1\bar{x}2\bar{x}3 + \bar{x}1\bar{x}2x3 + \bar{x}1x2x3 + x1\bar{x}2\bar{x}3 + x1x2\bar{x}3 + x1x2x3.$$

Таблица 1.6

N	x_1	x_2	x_3	y_1	y_2	y_3
0	0	0	0	1	1	1
1	0	0	1	1	0	1
2	0	1	0	0	0	0
3	0	1	1	1	1	1
4	1	0	0	0	0	1
5	1	0	1	0	0	0
6	1	1	0	1	0	1
7	1	1	1	1	0	1

Совершенная конъюнктивная нормальная форма – это конъюнкция простых дизъюнкций, где под термином простая дизъюнкция имеется в виду дизъюнкция переменных или их отрицаний. В СКНФ простые дизъюнкции содержат все переменные в своей прямой или инверсной форме и представляют собой отрицание конституент нуля. Общая запись СКНФ функции y имеет вид

$$y = \wedge(x_1^{\delta 1} + x_2^{\delta 2} + x_3^{\delta 3} \ldots + x_{(n-1)}^{\delta(n-1)} + x_n^{\delta n}),$$

где $$x_i^{\delta\iota} = \begin{cases} x_i, \text{ если } \delta\iota = 1, \\ \overline{x_i}, \text{ если } \delta\iota = 0. \end{cases}$$

СКНФ легко сформировать на основе таблицы истинности. Например, для функций из предыдущей таблицы (табл.1.6) имеем:

$$y_1 = (x1 + \bar{x}2 + x3)(\bar{x}1 + x2 + x3)(\bar{x}1 + x2 + \bar{x}3);$$

$$y_2 = (x1 + x2 + \bar{x}3)(x1 + \bar{x}2 + x3)(\bar{x}1 + x2 + x3)(\bar{x}1 + x2 + \bar{x}3)(\bar{x}1 + \bar{x}2 + x3)(\bar{x}1 + \bar{x}2 + \bar{x}3);$$

$$y_3 = (x1 + \bar{x}2 + x3)(\bar{x}1 + x2 + x3).$$

СКНФ строится на основе конституент нуля. Конституента нуля представляет набор логических переменных, на котором логическая функция принимает значение 0. Каждая скобка в приведенных выражениях представляет собой отрицание конституенты нуля соответствующей функции, а запись функции в виде конъюнкция таких скобок представляет собой условие, при котором отсутствуют все конституенты нуля определяемой функции, при выполнении которого функция имеет единичное значение.

Например, для y_1 выражение первой скобки представляет собой отрицание набора значений переменных второй строки, на котором функция y_1 имеет нулевое значение, выражение второй скобки представляет собой отрицание набора значений переменных четвертой строки, на котором функция y_1 также имеет нулевое значение, выражение третьей скобки представляет собой отрицание набора значений переменных пятой строки, на котором функция y_1 имеет нулевое значение.

Из вышеизложенного следует, что любую функцию можно представить или в СДНФ, или в СКНФ, а так как эти формы представлены в базисе Буля, то значит, что этот базис (базис И, ИЛИ, НЕ) является функционально полным.

Если функция задана в СДНФ и требуется найти ее СКНФ, то такой переход можно выполнить, составив по заданной СДНФ таблицу истинности для этой функции, а на основе полученной таблицы составить СКНФ заданной функции.

Однако в некоторых случаях может оказаться более удобным подход, который поясняется следующим примером.

Пример

По заданной СДНФ функции

$y_3 = x1x2\bar{x}3 + \bar{x}1\bar{x}2x3 + x1\bar{x}2\bar{x}3 + x1x2x3 + \bar{x}1\bar{x}2\bar{x}3 + \bar{x}1x2x3$

найти запись этой функции в СКНФ.

Решение

Запишем логические выражение отрицания заданной функции, т.е. найдем логическое условие, при котором эта функция имеет нулевое значение. В качестве такого выражения можно взять дизъюнкцию конъюнкций, где каждая конъюнкция представляет собой конституенту нуля заданной функции. Очевидно, что конституенты нуля это те наборы, которые не являются наборами, соответствующими конституентам единицы, которые использованы в СДНФ. Таким образом, можно записать

$$\bar{y}3 = \bar{x}1x2\bar{x}3 + x1\bar{x}2x3.$$

Эту запись можно интерпретировать как словесное описание функции:

– функция y равна нулю, если имеет место хотя бы одна из конституент нуля.

В этой записи представлена дизъюнкция тех наборов, которые не использовались в записи функции y_3. Возьмем отрицание правой и левой частей полученного уравнения и применим к правой части правило Де Моргана.

$$\bar{\bar{y}}3 = \overline{\bar{x}1x2\bar{x}3 + x1\bar{x}2x3} = \overline{\bar{x}1x2\bar{x}3} \bullet \overline{x1\bar{x}2x3}.$$

Применим *правило Де Моргана* к отрицаниям конъюнкций, полученным в правой части:

$$\bar{\bar{y}}3 = y3 = (x1 + \bar{x}2 + x3)(\bar{x}1 + x2 + \bar{x}3).$$

Полученная запись для y является искомой СКНФ.

1.2.3. Синтез логических схем по логическим выражениям

Логические схемы строятся на основе логических элементов, набор которых определяется заданным логическим базисом.

Для базиса Буля в качестве логических элементов используются элементы, реализующие базовые логические функции И, ИЛИ, НЕ, которые имеют обозначения, приведенные на рис. 1.1.

При синтезе схемы по логическому выражению, составляющие логические операции представляются в виде соответствующих логических элементов, связи между которыми определяются последовательностью выполнения логических операций в заданном выражении.

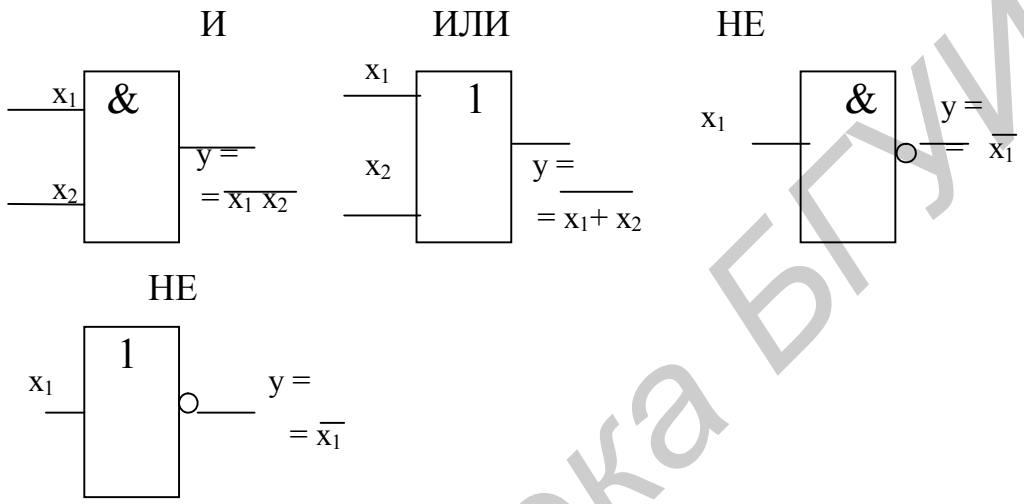


Рис. 1.1

Пример

Синтезировать логическую схему в базисе И, ИЛИ, НЕ, реализующую логическое выражение

$$y_1 = \overline{(x_1x_2 + x_1x_3 + x_2 x_3)}(x_1 + x_2 + x_3) + x_1 x_2 x_3.$$

Решение

Входными сигналами синтезируемой схемы являются x_1, x_2, x_3, а выходным - y_1.

Реализацию заданного выражения в виде логической схемы можно начать или с последней операции, или с первой.

Последней операцией в заданном выражении является операция логического сложения двух операндов:

$$\overline{(x_1x_2 + x_1x_3 + x_2 x_3)}(x_1 + x_2 + x_3) \quad \text{и} \quad x_1 x_2 x_3,$$

поэтому для её реализации требуется элемент ИЛИ с двумя входами, на выходе которого будет сформирован сигнал, соответствующий y_1, если на его входы будут поданы эти два слагаемые (например, первое слагаемое на второй вход, а второе слагаемое на первый вход).

На первый вход ИЛИ подается логическое произведение $x_1\, x_2\, x_3$,
для реализации которого необходимо использовать логический элемент И с тремя входами, на которые подаются входные переменные x_1, x_2, x_3. Аналогичным образом рассматривается последовательность формирования выражения

$$\overline{(x_1x_2 + x_1x_3 + x_2\, x_3)}(x_1 + x_2 + x_3),$$

которое соответствует сигналу, подаваемому на второй вход элемента ИЛИ. В результате синтезируется схема для заданного выражения, приведенная на рис.1.2 .

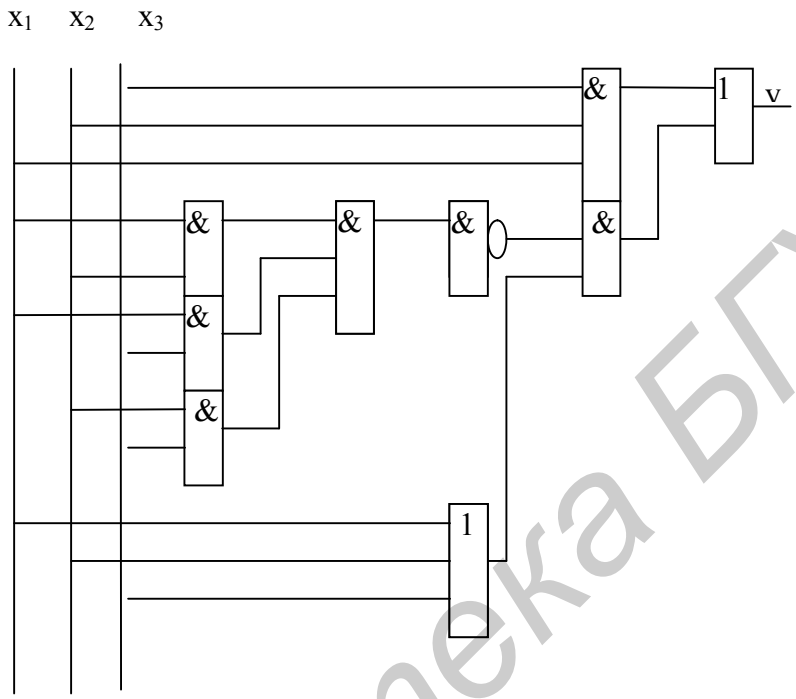


Рис. 1.2

1.2.4. Минимизация логических выражений

Учитывая то, что одну и ту же логическую функцию можно представить различными выражениями, перед реализацией функции в виде логической схемы весьма важным является выбор из всех возможных выражений, соответствующих данной функции, самого простого. Решить эту проблему можно за счет использования процедуры минимизации логического выражения. Из множества методов минимизации наиболее часто используются:

– *минимизация* методом Квайна;

– *минимизация* с использованием диаграмм Вейча (или карт Карно).

1.2.4.1. Минимизация методом Квайна

Метод минимизации *Квайна* предполагает следующее.

В качестве исходной формы представления логического выражения используется СДНФ.

Если подлежащее минимизации выражение имеет другую форму, то приведение к СДНФ осуществляется за счет открытия скобок, избавления от отрица-

ний логических выражений, более сложных, чем отрицание переменной (используется правило Де Моргана).

В результате таких действий будет получена дизъюнкция простых конъюнкций (т.е. конъюнкций, логическими сомножителями которой являются переменные или их отрицания). Далее для получения СДНФ, у которой используемые простые конъюнкции включают все переменные или их отрицания, в конъюнкции, где представлены не все переменные, вводят недостающие переменные x_i за счет умножения расширяемой конъюнкции на

$$(\bar{x}_i + x_i)$$

и раскрытия скобок.

Метод Квайна выполняется в два этапа.

Первый этап имеет своей целью получение тупиковой формы, представляющей собой дизъюнкцию, в качестве слагаемых которой используются конъюнкции, каждая из которых не склеивается ни с одной другой конъюнкцией, входящей в это выражение. Такие конъюнкции называются простыми импликантами.

Данный этап выполняется за счет реализации отдельных шагов. На каждом шаге на основании выражения, полученного на предыдущем шаге, выполняются все возможные операции склеивания для пар имеющихся конъюнкций. Каждый шаг понижает ранг исходных конъюнкций на единицу. Шаги повторяются до получения тупиковой формы.

Второй этап имеет своей целью устранения из тупиковой формы всех избыточных простых импликант, что дает в результате минимальное логическое выражение.

Пример

Найти методом Квайна минимальное выражение для функции y:

$$y = \bar{x}1\bar{x}2x3x4 + x1x2\bar{x}3x4 + \bar{x}1x2\bar{x}3x4 + \bar{x}1x2x3\bar{x}4 + \bar{x}1x2\bar{x}3\bar{x}4 + x1x2\bar{x}3x4 +$$
$$+ x1\bar{x}2x3\bar{x}4 + x1x2x3\bar{x}4 + x1\bar{x}2x3x4 + \bar{x}1\bar{x}2x3\bar{x}4.$$

Решение

1-й этап

$$y = \underline{x_1 x_2 x_3 x_4}^{1} + \underline{x_1 x_2 x_3 x_4}^{2} + \underline{x_1 x_2 x_3 x_4}^{3} + \underline{x_1 x_2 x_3 x_4}^{4} + \underline{x_1 x_2 x_3 x_4}^{5} +$$
$$+ \underline{x_1 x_2 x_3 x_4}^{6} + \underline{x_1 x_2 x_3 x_4}^{7} + \underline{x_1 x_2 x_3 x_4}^{8} + \underline{x_1 x_2 x_3 x_4}^{9} + \underline{x_1 x_2 x_3 x_4}^{10} =$$

(проставлены номера конъюнкций; подчеркнуты те конъюнкции, которые склеивались на очередном шаге)

$$= \underset{1\text{-}9}{\underline{x_2 x_3 x_4}^{1}} + \underset{1\text{-}10}{\underline{x_1 x_2 x_3}^{2}} + \underset{2\text{-}5}{\underline{x_2 x_3 x_4}^{3}} + \underset{2\text{-}6}{x_1 x_2 x_3}^{4} + \underset{2\text{-}8}{\underline{x_1 x_2 x_4}^{5}} + \underset{3\text{-}5}{x_1 x_2 x_3}^{6} + \underset{3\text{-}6}{\underline{x_2 x_3 x_4}^{7}} +$$
$$+ \underset{4\text{-}5}{\underline{x_1 x_2 x_4}^{8}} + \underset{4\text{-}8}{\underline{x_2 x_3 x_4}^{9}} + \underset{4\text{-}10}{\underline{x_1 x_3 x_4}^{10}} + \underset{7\text{-}8}{\underline{x_1 x_3 x_4}^{11}} + \underset{7\text{-}9}{\underline{x_1 x_2 x_3}^{12}} + \underset{7\text{-}10}{\underline{x_2 x_3 x_4}^{13}} =$$

(выражение представлено только результатами склеивания; если бы в исходном выражении не были подчеркнутые конъюнкции, то они должны были бы через знак «+» дописаны к сумме результатов склеивания; в скобке под конъюнкцией (i-j) указывают, что данная конъюнкция является результатом склеивания i-й и j-й конъюнкций исходного выражения):

$$= \underset{1\text{-}12}{\bar{x}_2 x_3} + \underset{2\text{-}11}{x_2 \bar{x}_3} + \underset{3\text{-}6}{\bar{x}_2 x_3} + \underset{3\text{-}8}{\bar{x}_2 x_4} + \underset{4\text{-}7}{x_2 \bar{x}_4} + \underset{8\text{-}12}{\bar{x}_3 x_4} + \underset{9\text{-}10}{x_3 \bar{x}_4} + \underset{5}{\bar{x}_1 x_2 x_3} =$$

$$= \underset{1\text{-}13}{\bar{x}_2 x_3} + \underset{2\text{-}12}{x_2 \bar{x}_3} + \underset{3\text{-}7}{\bar{x}_2 x_3} + \underset{}{\bar{x}_2 x_4} + \underset{3\text{-}9}{x_2 \bar{x}_4} + \underset{5\text{-}8}{\bar{x}_2 x_4} + \underset{9\text{-}13}{x_2 \bar{x}_4} + \underset{10\text{-}11}{\bar{x}_3 x_4} + \underset{6}{x_1 \bar{x}_2 x_3} =$$

(к результатам склеивания логически добавлен ни с чем не склеенный пятый член исходного выражения)

$$= \underset{}{\bar{x}_2 x_3} + \underset{}{x_2 \bar{x}_3} + \underset{}{\bar{x}_2 x_4} + \underset{}{\bar{x}_3 x_4} + \underset{}{x_1 \bar{x}_2 x_3} = y_T$$ - тупиковая форма.

Последнее выражение получено из предыдущего посредством удаления повторяющихся членов

2-й этап

На основании исходного выражения и полученной тупиковой формы составляется и заполняется импликантная табл.1.7.

Колонки приведенной таблицы помечены конституентами единицы, имеющимися в исходном логическом выражении.

Строки таблицы помечены простыми импликантами полученной тупиковой формы.

Таблица 1.7

	$\bar{x}_1\bar{x}_2x_3x_4$	$\bar{x}_1x_2\bar{x}_3x_4$	$\bar{x}_1x_2x_3\bar{x}_4$	$x_1\bar{x}_2\bar{x}_3x_4$	$x_1\bar{x}_2x_3\bar{x}_4$	$x_1x_2\bar{x}_3\bar{x}_4$	$x_1\bar{x}_2\bar{x}_3x_4$	$x_1\bar{x}_2x_3x_4$	$x_1x_2\bar{x}_3x_4$	$x_1x_2x_3\bar{x}_4$
	1	2	3	4	5	6	7	8	9	10
$\bar{x}_2x_3$	*						*		*	*
$x_2\bar{x}_3$		*	*		*	*				
$x_2\bar{x}_4$		*		*	*			*		
$\bar{x}_3x_4$				*			*	*		*
$\bar{x}_1\bar{x}_2x_3$			*		*					

Звездочками в каждой строке отмечены те конституенты единицы, которые покрываются соответствующей простой импликантой (практически отмечаются

те конституенты единицы, которые включают простую импликанту как свою составную часть).

Анализируя покрытия простыми импликантами конституент единицы заданной функции, составляем её минимальное выражение:

$$y_{min} = \bar{x}_2 x_3 + x_2 \bar{x}_3 + x_2 x_4.$$

Минимальное выражение y_{min} формируется за счет последовательного включения простых импликант. При этом используется следующая приоритетность включения импликант в формируемое минимальное выражение:

- простая импликанта является единственной, покрывающей одну из колонок;
- если импликант вышеуказанного типа нет, то выбирается импликанта, покрывающая большее количество еще не покрытых колонок.

Последовательность включения простых импликант в приведенное минимальное выражение:

$\bar{x}2x3$ – единственная импликанта, покрывающая колонку 1, при этом из дальнейшего рассмотрения исключаются все колонки, покрываемые этой импликантой, т.е. колонки 1,7,9 10;

$x2\bar{x}3$ – покрывает максимальное число колонок, оставшихся для рассмотрения (колонки 2,3,5,6) - эти колонки из дальнейшего рассмотрения исключаются;

$\bar{x}2x4$ – покрывает оставшиеся две колонки 4, 8; после выбрасывания этих двух колонок для рассмотрения не останется ни одной колонки, не покрытой уже включенными в формируемое выражение простыми импликантами. Поэтому простые импликанты $x3\bar{x}4$ и $\bar{x}1x2\bar{x}3$ найденной тупиковой формы являются избыточными и в минимальном логическом выражении для заданной функции не присутствуют.

1.2.4.2. Минимизация с диаграммами Вейча

Минимизация этим методом предполагает использование специальных форм – диаграмм Вейча (или карт *Карно)*.

Карта Карно для n логических переменных представляет собой множество квадратов (клеток), объединённых в близкую к квадрату прямоугольную форму. Каждая такая клетка соответствует одному набору логических переменных, причем наборы двух соседних клеток должны отличаться на значение одной переменной (их наборы образуют склеивающиеся конъюнкции).

На рис.1.3 приведены карты Карно для n = 1, 2, 3. На. рис.1.3, а, б показана разметка колонок и строк, а также указан для каждой составляющей клетки соответствующий ей набор. Разметка колонок (строк) указывает, какие значения данная переменная имеет в клетках, находящихся в данной колонке (строке). На рис.1.3,в приведен пример компактной разметки карты, соответствующей карте на рис.1.3, б. Здесь помечаются колонки (строки), в которых соответствующая переменная имеет прямое значение. На рис.1.3, г приведена карта Карно для n = 3, сформированная посредством зеркального отображения карты Карно для n=2 (рис.1.3, в) относительно правой границы. Этот прием универ-

сальный; его можно использовать для построения карты для заданного n на основании имеющейся карты Карно для n -1 переменной. Клетка, отмеченная знаком «*», соответствует набору $\bar{x}1\bar{x}2\bar{x}3$.

Карты Карно используются для представления и минимизации логических функций.

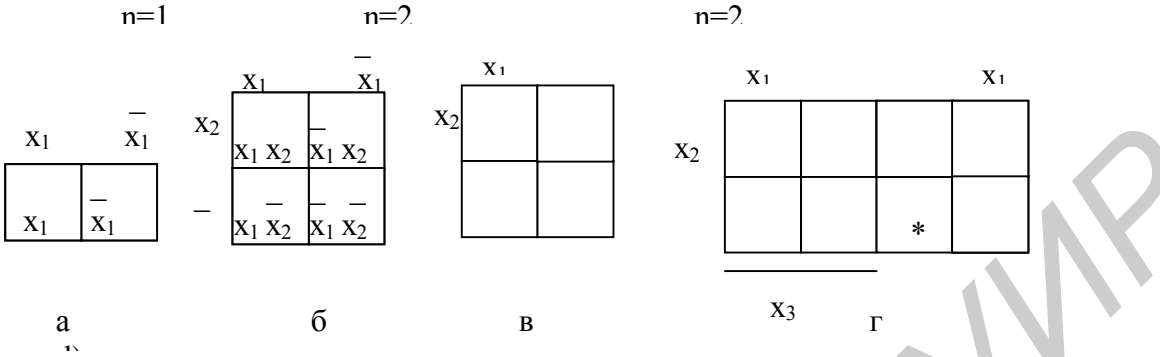


Рис. 1.3

Записываемая функция должна быть представлена в СДНФ. Запись функции в карту осуществляется за счет установки 1 в те клетки карты, которые соответствуют конституентам единиц записываемой функции.

Например, если задана логическая функция у трех переменных в виде выражения

$$y = x1x2\bar{x}3 + x1\bar{x}2x3 + \bar{x}1x2x3 + x1x2x3,$$

то её запись в карту Карно будет иметь вид, приведенный на рис.1.4.

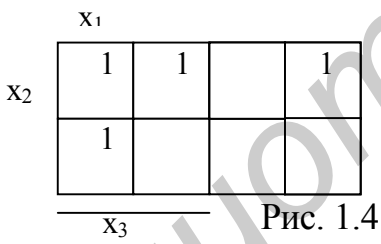


Рис. 1.4

Для выполнения минимизации представленной в карте Карно функции необходимо выполнить два этапа:

– охватить множество клеток карты Карно контурами;

– записать минимальное выражение для заданной функции в виде дизъюнкции конъюнкций, где каждая конъюнкция соответствует одному из введенных на карте контуров.

Охват клеток карты контурами выполняется с соблюдением следующих правил:

– контур должен иметь прямоугольную форму;

– в контур может входить такое количество клеток, которое равно целой степени числа 2;

– в контур могут входить клетки, являющиеся логическими соседями;

– в контур необходимо включить максимальное количество клеток с учетом вышеприведенных требований;

– контурами необходимо охватить все клетки с единичными значениями;

– контуров должно быть минимальное количество;

– количество клеток в контуре должно быть равно $2^{\Delta R}$, где Δ_R – разность ранга (дельта ранга) конституент единицы заданной функции и ранга конъюнкции, соответствующей контуру.

Логическими соседями являются такие две клетки, наборы которых отличаются только одной переменной - в одном эта переменная должна иметь прямое, в другом - обратное значение.

Для того чтобы быть логическими соседями, клеткам достаточно быть геометрическими соседями. Имея в виду, что карта является пространственным объектом и заворачивается по горизонтали и вертикали, сливаясь своими крайними горизонтальными и крайними вертикальными границами, можно считать, что соответствующие крайние горизонтальные и вертикальные клетки являются геометрическими соседями. Логическими соседями могут быть клетки, которые не являются геометрическими соседями. К числу таких клеток относятся клетки, которые по горизонтали или вертикали симметричны относительно линий зеркального отображения, которые были использованы при переходе от n к n+1 переменным.

Запись минимального выражения заданной функции имеет вид дизъюнкции простых конъюнкций, соотвествующих контурам на карте, и формируется следующим образом:

– конъюнкция, соответствующая контуру, должна включать только те переменные, которые имеют постоянное значение во всех клетках, охваченных рассматриваемым контуром,

– или по-другому: в конъюнкцию, соответствующую контуру, не должны входить переменные, которые имеют разные значения для клеток, охваченных рассматриваемым контуром.

Для функции, заданной в карте Карно, приведенной на рис.1.4, контуры имеют вид, приведенный на рис.1.5.

Для примера, контур 1 представлен на рисунке в виде двух клеток: клетки, соответствующей набору $x_1x_2x_3$, и клетки, соответствующей набору $x1x2\bar{x}3$, поэтому данному контуру будет соответствовать конъюнкция $x_1\ x_2$.

Минимальное логическое выражение для функции имеет вид

$$y = x_1\ x_2 + x_1\ x_3 + x_2\ x_3.$$

1 2 3

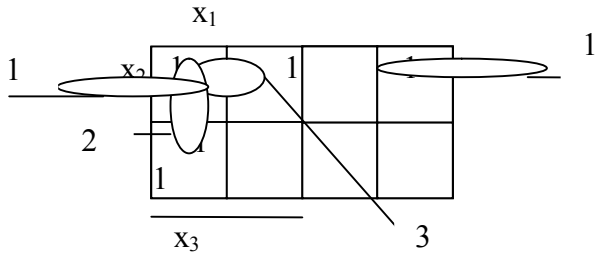


Рис. 1.5

Конъюнкции минимального выражения помечены внизу цифрами, соответствующими номерам контуров, которые они представляют.

На рис.1.6 приведены карты Карно для четырех и пяти переменных.

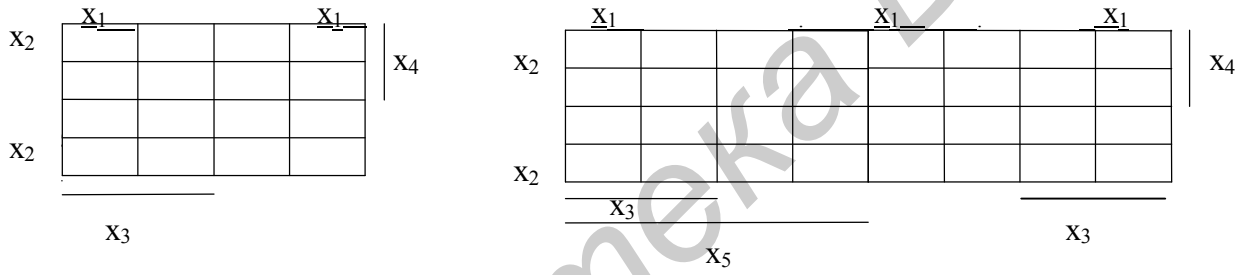
На рис.1.7 приведена карта Карно для шести переменных. В этой карте приведен пример расположения четырех геометрически соседних клеток с единичными значениями, которые нельзя объединить единым контуром 1. Эти рядом лежащие клетки необходимо охватить двумя контурами 2, 3. Действительно, конъюнкция для неправильного контура 1 имеет вид $x2\bar{x}4x6$ и дельта ранга Δ_R при ранге шесть конституент единицы исходного выражения составляет значение 3, отсюда количество клеток, входящих в контур, должно быть равно третьей степени двойки. Это требование не выполняется (контур охватывает только четыре клетки), следовательно, контур 1 введен неправильно. В приведенной ситуации необходимо для охвата рассматриваемых клеток использовать два контура, соответственно контур 2 и контур 3.

Пример

Минимизировать функцию y, заданную в карте Карно, приведенной на рис.1.8.

Решение

Карта Карно с введенными контурами приведена на рис.1.9.



а n=4 б n=5

Рис. 1.6

Минимальное выражение для y, составленное по введенным контурам, имеет вид

$$y = \underset{1}{\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4} + \underset{2}{x_3 x_4 x_5 x_6} + \underset{3}{x_1 x_2 x_3 x_4} + \underset{4}{x_1 x_2 \bar{x}_3 x_4 x_5} +$$

$$+ \underset{5}{\bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 x_5} + \underset{6}{x_1 x_2 \bar{x}_3 x_5 x_6} + \underset{7}{x_1 x_3 x_4 x_6}.$$

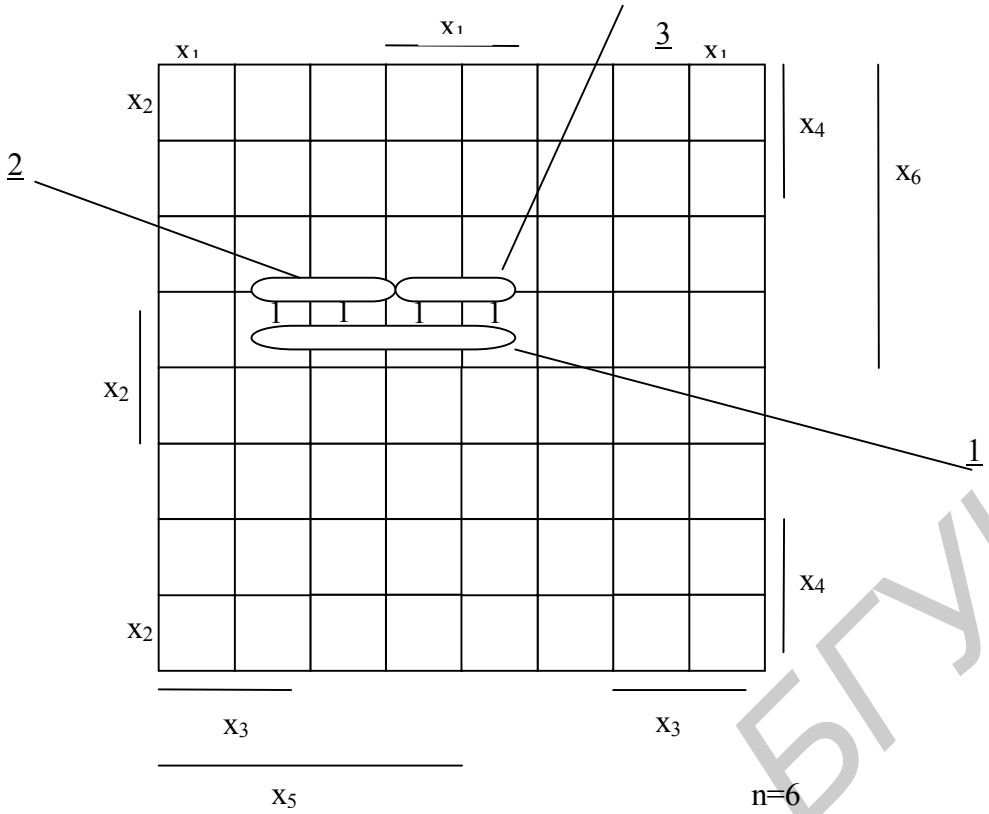


Рис.1.7

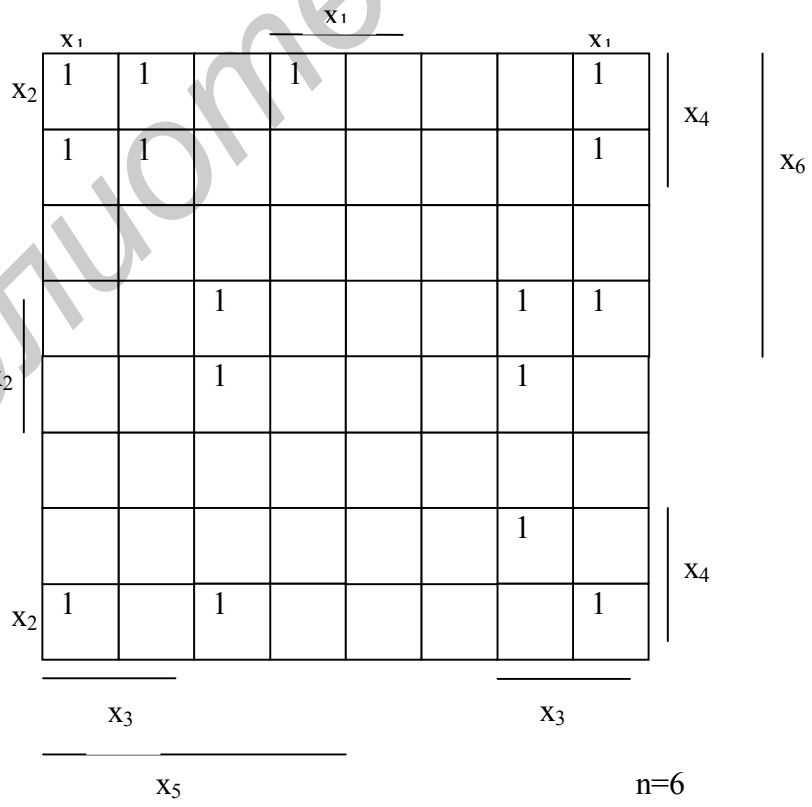


Рис. 1.8

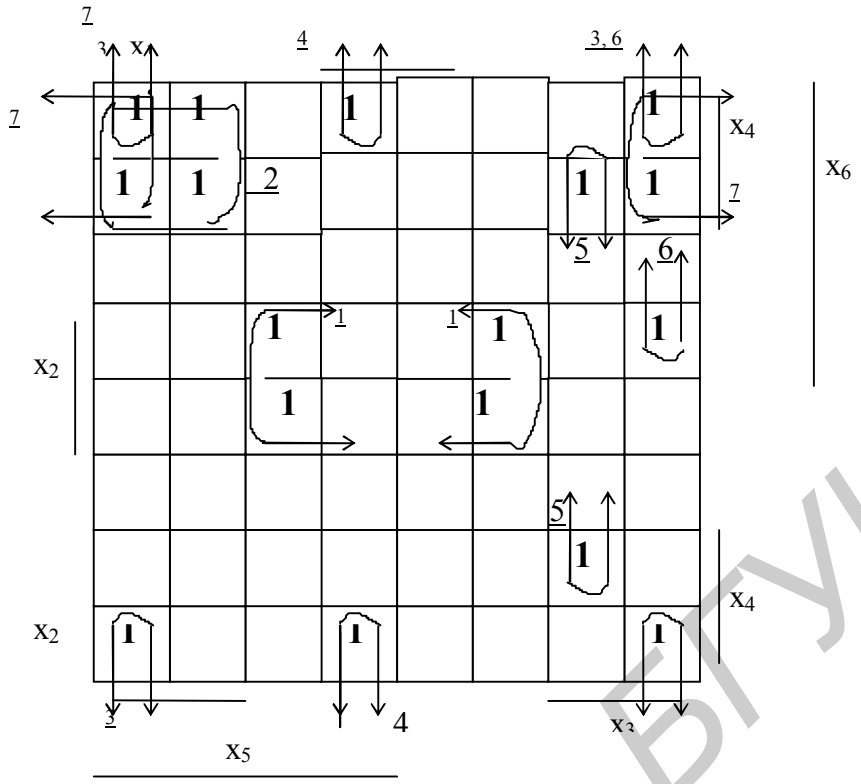


Рис. 1.9

Под каждой конъюнкцией указан номер контура, которому она соответствует.

1.2.5. Логические базисы И-НЕ, ИЛИ-НЕ

Булевый базис не является единственной функционально полной системой логических функций. Среди других наибольшее распространение получили базис И-НЕ и базис ИЛИ -НЕ.

Чтобы доказать логическую полноту любого базиса, достаточно показать, что в этом базисе можно реализовать базовые функции И, ИЛИ, НЕ.

Для базиса И-НЕ в качестве базового элемента используется элемент, приведенный на рис.1.10,а.

а б

Рис. 1.10

Реализация с помощью функции И-НЕ базовых функций алгебры Буля осуществляется следующим образом.

ИЛИ: $x1 + x2 = \overline{\overline{x1 + x2}} = \bar{x}1\bar{x}2$

И: $x1x2 = \overline{\overline{x1x2}}.$

Функция НЕ реализуется с помощью схемы И -НЕ с одним входом.

На рис.1.11 приведена схемная реализация функций И, ИЛИ, НЕ в базисе И - НЕ.

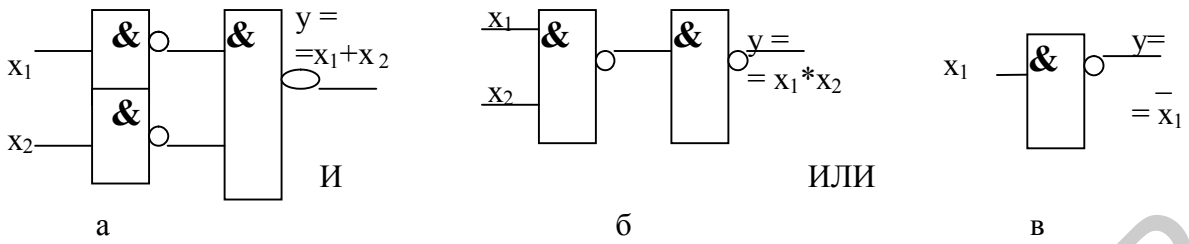


Рис. 1.11

Реализация с помощью логической функции ИЛИ-НЕ базовых функций алгебры Буля осуществляется следующим образом.

ИЛИ: $x1 + x2 = \overline{\overline{x1 + x2}}.$

И: $x1x2 = \overline{\overline{x1x2}} = \overline{\overline{x1} + \overline{x2}}$

Функция НЕ реализуется с помощью схемы ИЛИ -НЕ с одним входом.

На рис.1.12 приведена схемная реализация операции И, ИЛИ, НЕ в базисе ИЛИ – НЕ.

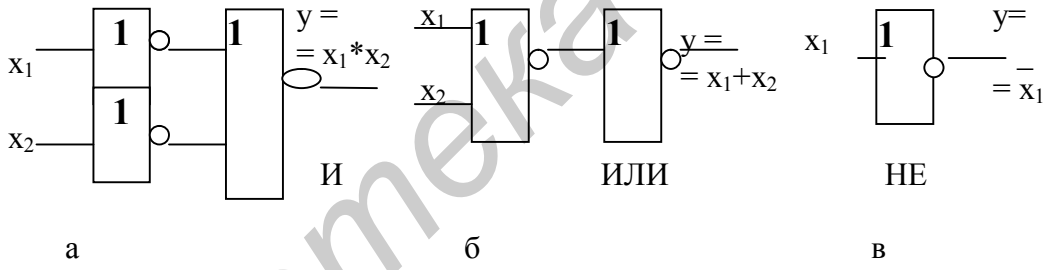


Рис. 1.12

При синтезе логических схем в заданном базисе логических элементов (например, в базисе И-НЕ или ИЛИ - НЕ) целесообразно предварительно исходное выражение привести к форме, в которой в выражении будут использованы только логические операции, соответствующие используемым логическим элементам в заданном базисе.

Пример

Синтезировать логическую схему в базисе И-НЕ, соответствующую выражению

$$y = \overline{(x1x2 + x1x3 + x2x3)}(x1 + x2 + x3) + x1x2x3 .$$

Решение

Используя правило Де Моргана, преобразуем исходное выражение таким образом, чтобы последней операцией было отрицание и в выражение были бы только операции И.

$$\overline{(x1x2 + x1x3 + x2x3)}(x1 + x3 + x3) + x1x2x3 = \overline{\overline{(x1x2 + x1x3 + x2x3)}(x1 + x2 + x3)} + x1x2x3 =$$

$$= \overline{\overline{(x1x2 + x1x3 + x2x3)}(x1 + x2 + x3)}\,\overline{x1x2x3} = \overline{\overline{(x1x2x1x3x2x3)}(x1 + x2 + x3)}\,\overline{x1x2x3} =$$

$$= \overline{\overline{x1x2x1x3x2x3}\,\overline{x1x2x3}}\,\overline{x1x2x3}.$$

Полученное выражение, представленное в виде вложенных операции И-НЕ, позволяет синтезировать соответствующую логическую схему в заданном базисе, которая приведена на рис.1.13.

Пример

Синтезировать логическую схему в *базисе ИЛИ - НЕ*, соответствующую выражению

$$y = \overline{(x1x2 + x1x3 + x2x3)}(x1 + x2 + x3) + x1x2x3\,.$$

Решение

Используя правило Де Моргана, преобразуем исходное выражение таким образом, чтобы последней операцией было отрицание и в выражение были бы только операции ИЛИ.

$$y = \overline{\overline{(x1x2+x1x3+x2x3)}(x1+x2+x3)}+x1x2x3 = \overline{\overline{(x1x2+x1x3+x2x3)}(x1+x2+x3)}+\overline{x1x2x3} =$$

$$= (x1x2+x1x3+x2x3)+\overline{(x1+x2+x3)}+\overline{x1}+\overline{x2}+\overline{x3} = \overline{(x1x2+x1x3+x2x3)+(x1+x2+x3)}+\overline{x1}+\overline{x2}+\overline{x3} =$$

$$= \overline{\overline{x1}+\overline{x2}+\overline{x1}+\overline{x3}+\overline{x2}+\overline{x3}+\overline{x1}+\overline{x2}+\overline{x3}+\overline{x1}+\overline{x2}+\overline{x3}}.$$

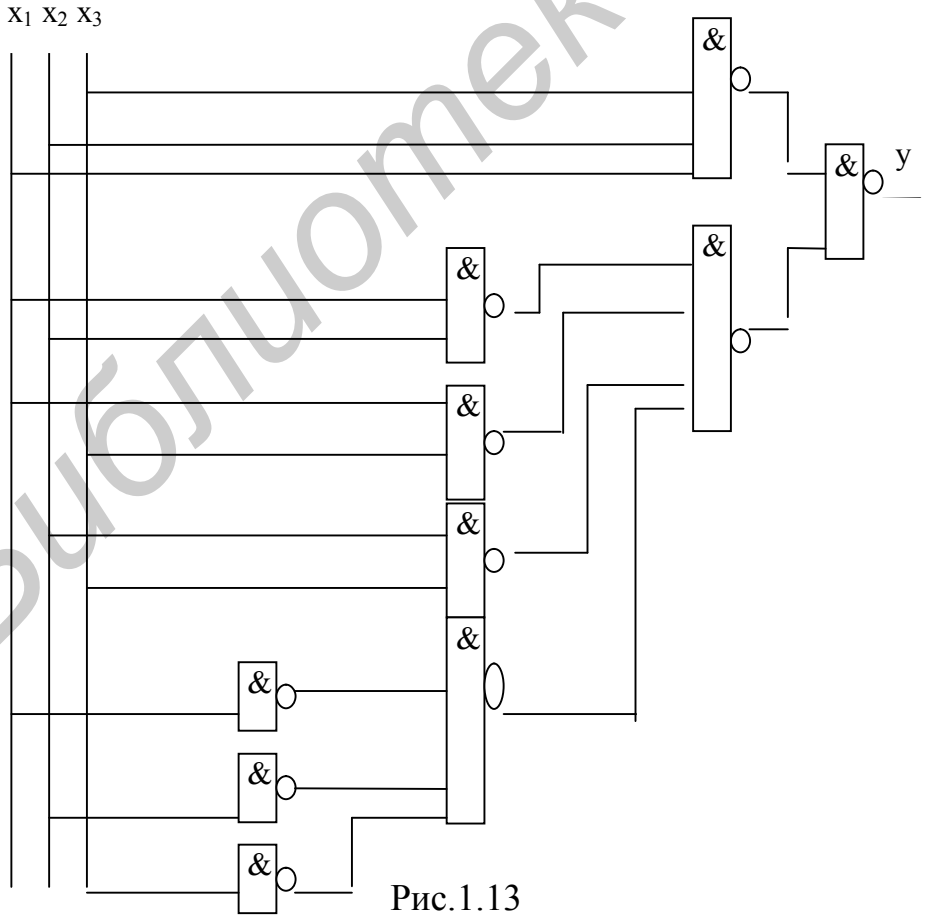


Рис.1.13

Полученное выражение, представленное в виде вложенных операций ИЛИ-НЕ, позволяет легко синтезировать соответствующую логическую схему в заданном базисе, которая приведена на рис.1.14.

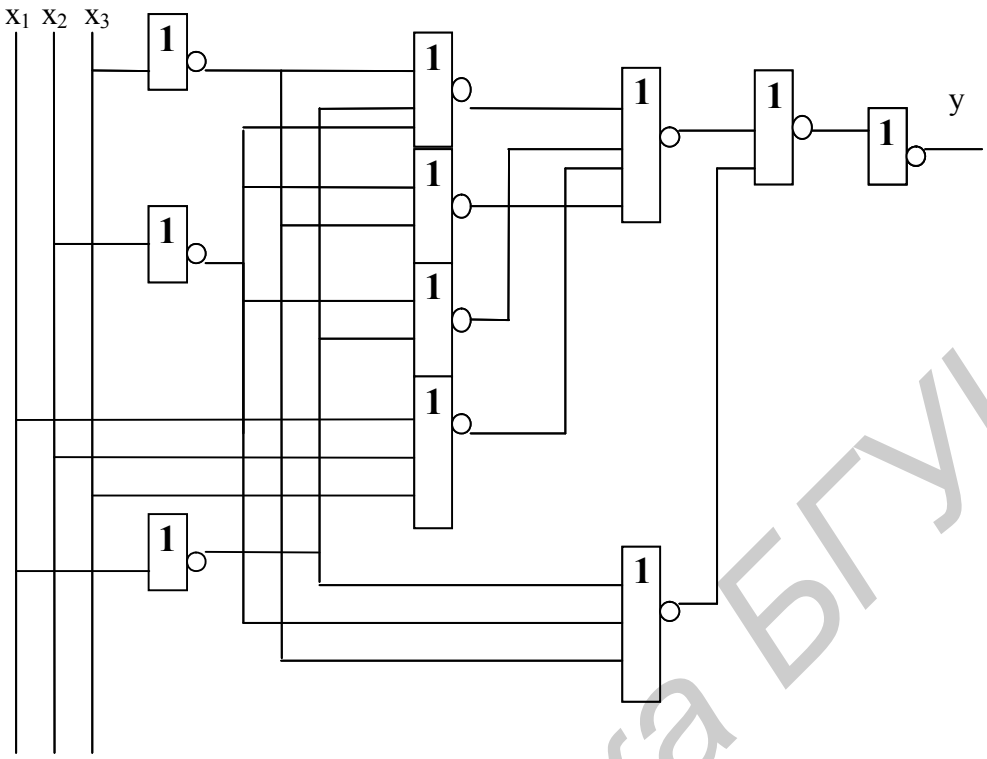


Рис. 1.14

2. Схемотехнические основы ЭВМ

2.1. Элементы ЭВМ

Элементы ЭВМ, реализованные на радиотехнических деталях, представляют собой мельчайшие компоненты, на основе которых строятся более крупные составляющие вычислительной машины. Общее обозначение элемента ЭВМ приведено на рис.2.1. Входы у элемента находятся слева, выходы – справа. На обозначении могут присутствовать три поля:

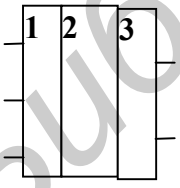


Рис. 2.1

- поле 1 – используется для обозначения входов;
- поле 2 – несет информацию о типе элемента и выполняемых функций; внизу этого поля может находиться номер элемента;
- поле 3 – содержит информацию о выходах.

Можно выделить три основные разновидности элементов:

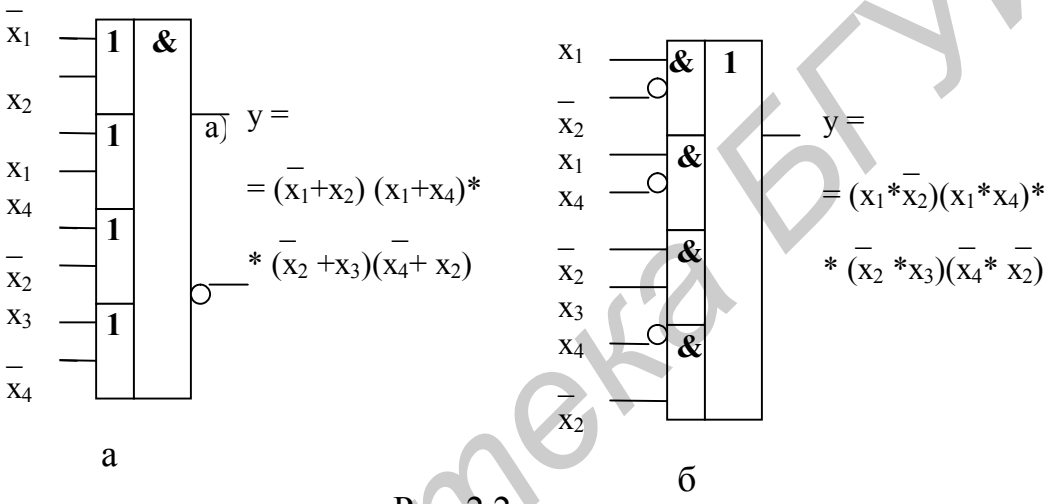
- логические элементы;
- запоминающие элементы;
- специальные элементы.

Специальные элементы не выполняют, как правило, логических функций и обеспечивают усиление сигнала, формирование выходных сигналов с заданными параметрами и т.д.

2.1.1. Логические элементы

Логические элементы, так же как и элементы алгебры логики, реализуют логические функции, но эти функции, оставаясь сравнительно простыми, все же сложней, чем базовые функции в алгебре логики. В одном логическом элементе может быть реализовано несколько простых функций. Кроме того, логические элементы характеризуются дополнительными параметрами, такими, как количество входов, нагрузочная способность (количество входов других элементов, к которым можно подключать выход данного элемента).

На рис.2.2. приведены примеры рассматриваемых логических элементов.

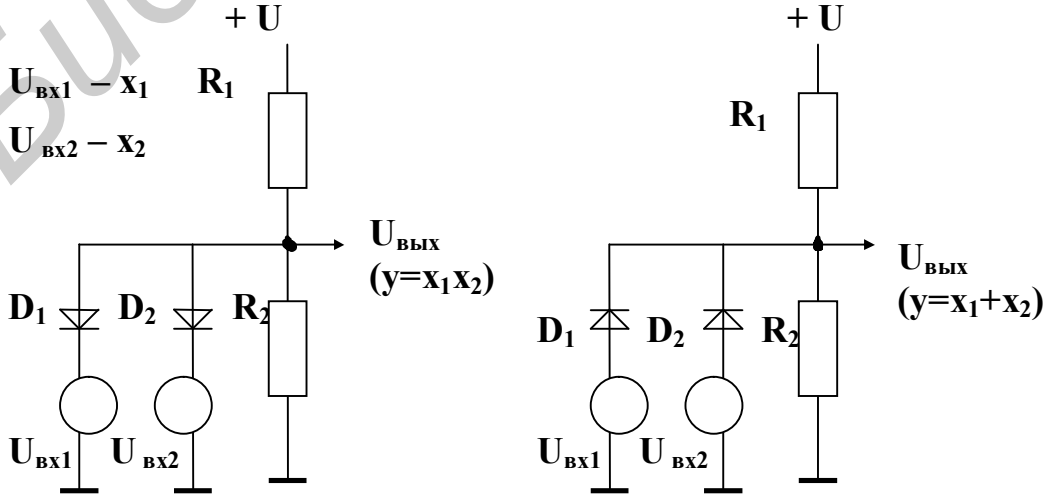


а б

Рис. 2.2

На выходах элементов указаны логические выражения для выходных сигналов в соответствии с приведенными входными сигналами. На рис. 2.2, б приведен логический элемент с инверсным входом (в логическом выражении сигнал по такому входу используется в обратном значении).

Примеры реализации простейших логических элементов с помощью диодно-резисторной схемы приведены на рис.2.3.



а Рис.2.3 б

На рис.2.3,а приведена реализация логических элементов И. Реализация элемента ИЛИ приведена на рис.2.3,б. Схемы логических элементов построены с условием, что логическая «1» соответствует высокому уровню («+»), а логический «0» - низкому уровню напряжения, близкому «земле». Это соответствие используется и в других реализациях. На рис.2.3,а соотношение сопротивления резисторов R1 и R2 при заданном напряжении «+U» выбирается таким образом, что без учета шунтирующего действия диодных цепочек напряжение на выходе имеет значение высокого уровня (уровня, соответствующего логической «1»). Источники входных сигналов $U_{вх1}$ и $U_{вх2}$ имеют малое внутреннее сопротивления. Поэтому, если один или оба источника подают низкий уровень (логический «0»), то из-за шунтирующего воздействия диодных цепочек на резисторе R2 на выходе будет иметь место низкий уровень напряжения, соответствующий логическому «0». Высокий уровень на выходе (логическая «1») будет иметь место только тогда, когда на оба входа подаются единицы, так как соответствующие им высокие уровни напряжения закрывают оба диода. Таким образом, единица на выходе будет иметь место только тогда, когда и x_1, и x_2 имеют единичные значения. Это означает, что рассматриваемая схема реализует логику И.

Для схемы на рис.2.3,б соотношение сопротивления резисторов R1 и R2 при заданном напряжении «+U» выбирается таким образом, что без учета воздействия диодных цепочек напряжение на выходе имеет значение низкого уровня (уровня, соответствующего логическому «0»). Если хотя бы один или оба источника входных сигналов подают высокий уровень (логическая «1»), то этот высокий уровень проходит через открытый диод и появляется на выходе. Низкий уровень, т.е. логический «0», будет иметь место только тогда, когда оба входных сигнала имеют низкий уровень. Это означает, что рассматриваемая схема реализует логику ИЛИ.

На рис.2.4. приведены примеры реализации логических функций НЕ (рис.2.4, а) и ИЛИ-НЕ (рис.2.4,б) на транзисторах. Транзисторы обозначены символом «Т».

На рис. 2.4,а транзистор открыт, следовательно, на его коллекторе напряжение, близкое к нулевому уровню, тогда, когда на его базе высокий уровень логической «1», и наоборот, если транзистор закрыт, на его коллекторе высокий уровень тогда, когда входной сигнал соответствует низкому уровню нуля. Таким образом, на выходе схемы, которым является коллектор транзистора, реализуется логическая функция НЕ.

На выходе у схемы рис.2.4,б будет низкий уровень (логический «0») тогда, когда открыт хотя бы один транзистор T_1, T_2, T_3, т.е. тогда, когда хотя бы одна из входных переменных x_1, x_2, x_3, имеет значение логической «1». Это означает, что выходной сигнал у зависит от входных сигналов по логике ИЛИ- НЕ.

На рис 2.5 приведены примеры реализации логических функций И-НЕ и функции И на транзисторах.

На выходе у схемы рис.2.5,а будет низкий уровень (логический «0») только тогда, когда открыты оба транзистора T_1, T_2, т.е. тогда, когда обе входные переменные x_1, x_2 имеют значение логической «1».

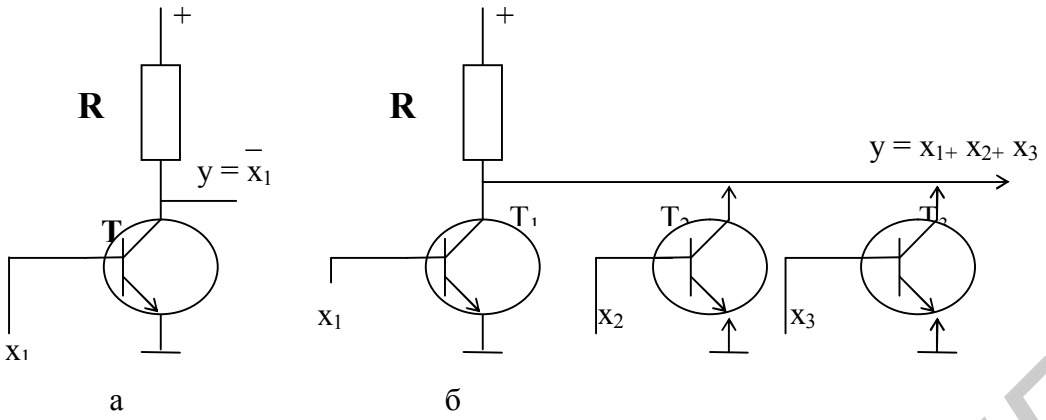


Рис.2.4

Это означает, что выходной сигнал у зависит от входных сигналов по логике И-НЕ.

На рис. 2.5,б приведена схема, использующая многоэмиттерный транзистор T_3. Транзистор такого типа пропускает ток только тогда, когда имеет место высокий уровень на его базе и низкий уровень хотя бы на одном из его эмиттеров. В приведенной схеме на базу T_3 подается постоянный высокий уровень (логическая константа, равная «1»). В этом случае на выходе схемы у будет низкий уровень (логический «0») тогда, когда есть условия протекания тока хотя бы по одному из его эмиттеров, т.е. хотя бы одна из входных переменных x_1, x_2, x_3 имеет значение логического «0». Если на все эмиттеры подается логическая «1», то T_3 закрыт, а на выходе схеме имеет место высокий уровень, т.е. логическая «1». Это означает, что выходной сигнал у зависит от входных сигналов по логике И.

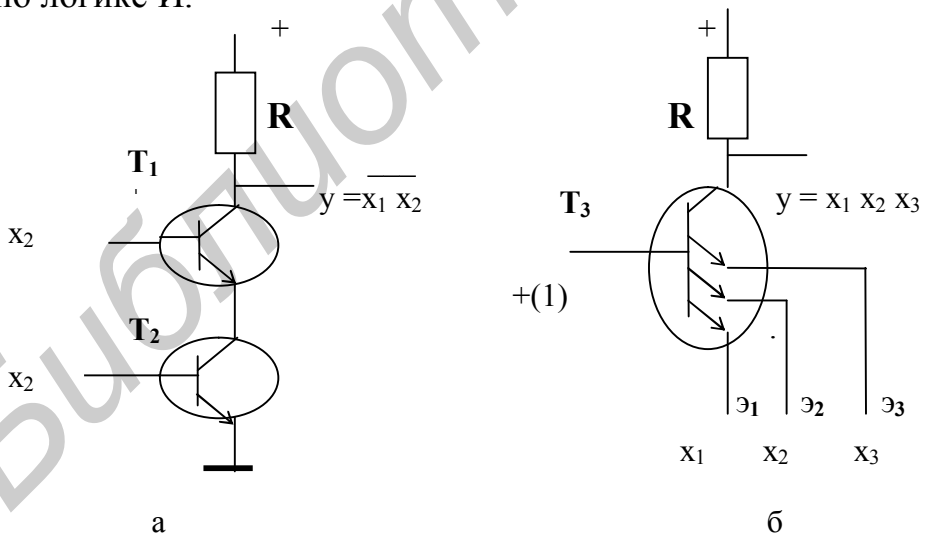


Рис.2.5

2.1.2. Запоминающие элементы

В качестве запоминающего элемента в вычислительной технике используется триггер.

Триггер в общем случае обладает следующими свойствами:

– триггер имеет два устойчивых состояния, которые называются состоянием «0» и состоянием «1»;

– триггер имеет парафазный выход, который представляется двумя выходами, всегда имеющими противоположные значения, при этом один выход имеет название «выход единицы», а другой - «выход нуля»;

– триггер имеет управляющий вход (или входы), подавая сигналы на который можно менять состояние триггера.

Используются следующие типы триггеров:

– RS- триггер;
– D-триггер;
– T-триггер;
– JK- триггер.

2.1.2.1. R S-т р и г г е р

Отличительной особенностью RS-триггера является наличие у него входа установки значения единицы (вход S) и входа установки значения ноль (вход R). Своё название этот триггер получил по названию входов.

RS-триггер может быть реализован или в логическом базисе ИЛИ-НЕ, или в логическом базисе И-НЕ.

При реализации в базисе ИЛИ-НЕ триггер представляется в виде логической схемы, приведенной на рис.2.6,а. Наличие в логической схеме обратных связей приводит к тому, что значения выходных переменных приведенной схемы зависят не только от входных переменных, но и от начального состояния схемы, т.е. от состояния триггера на момент поступления входных сигналов.

Доказательство того, что приведенная схема является триггером, имеющим вход для установки единицы и вход для установки нуля, приведено в табл.2.1.

В таблице используются обозначения Q(t) и Q(t+1), которые отражают соответственно начальное и конечное значения на выходе 1 триггера перед началом и после воздействия входных сигналов.

Обозначения $\bar{Q}(t)$ и $\bar{Q}(t+1)$ отражают соответственно начальное и конечное значения на выходе 0 триггера перед началом и после воздействия входных сигналов.

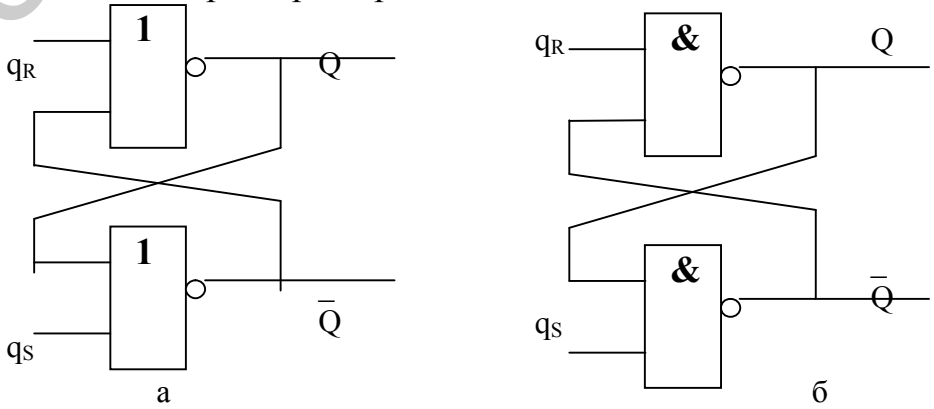


Рис.2.6

Каждая колонка имеет цифровое обозначение, соответствующее номеру набора входных переменных и начальному значению выходной переменной.

На наборе «1» рассматривается ситуация, когда на схему поданы нулевые значения на оба входа, а Q(t) равно 1. В этой ситуации $\overline{Q}$ (t+1) будет равен 0, что не изменит начального состояния на выходе Q, т. е. Q(t+1) будет 1.

Таблица 2.1

	1	2	3	4	5	6	7	8
q_s	0	0	0	1	1	0	1	1
q_R	0	0	1	0	0	1	1	1
Q(t)	1	0	0	0	1	1	0	1
$\overline{Q(t)}$	0	1	1	1	0	0	1	0
Q(t+1)	1	0	0	1	1	0	1?	1?
$\overline{Q(t+1)}$	0	1	1	0	0	1	1?	1?

На наборе 2 рассматривается ситуация, когда на схему поданы нулевые значения обеих входных переменных, а Q(t) равно 0). В этой ситуации на $\overline{Q}$(t+1) будет 1, что не изменит начального состояния на выходе Q, т.е. на Q(t+1) будет 0.

Работа схемы на наборах 1 и 2 свидетельствует о том, что при нулевых значениях входных переменных рассматриваемая схема сохраняет исходное состояние.

Формируемые выходные переменные на наборах 3 и 6 свидетельствуют о том, что рассматриваемая схема при воздействии входных переменных $q_s = 0$ и $q_R = 1$, независимо от исходного состояния (от значения Q(t)), в конечном состоянии будет иметь на своих выходах Q(t+1)=0, Q(t+1)=1, т.е. данная комбинация сигналов переводит схему в 0.

Формируемые выходные переменные на наборах 4 и 5 свидетельствуют о том, что рассматриваемая схема при воздействии значений входных переменных q_s, = 1 и q_R =0, независимо от исходного состояния (от значения Q(t)), в конечном состоянии будет иметь Q(t+1)=1, Q(t+1)=0, т.е. данная комбинация сигналов переводит схему в состояние 1.

Наборы 7 и 8 являются недопустимыми, так, при $q_S = 1$ и $q_R = 1$ состояние рассматриваемой схемы не определено, потому что на двух составляющих парафазного выхода схемы появляются одинаковые нулевые значения.

Таким образом, на основании приведенной таблицы можно заключить, что при различных комбинациях входных переменных q_S,q_R рассматриваемая схема может:

– хранить предыдущее состояние ($q_s = 0$, $q_R = 0$);
– устанавливаться в ноль ($q_s = 0$, $q_R = 1$);
– устанавливаться в единицу ($q_s = 1$, $q_R = 0$).

На основании изложенного можно заключить, что приведенная схема является триггером со входами установки единицы S и нуля R. Условное обозначение этой реализации триггера имеет вид, приведенный на рис.2.7,а.

Работу данного триггера отражает таблица истинности, табл.2.2.

Таблица 2.2

N строки	q_s	q_R	Q(t)	Q(t+1)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	Не определено
7	1	1	1	Не определено

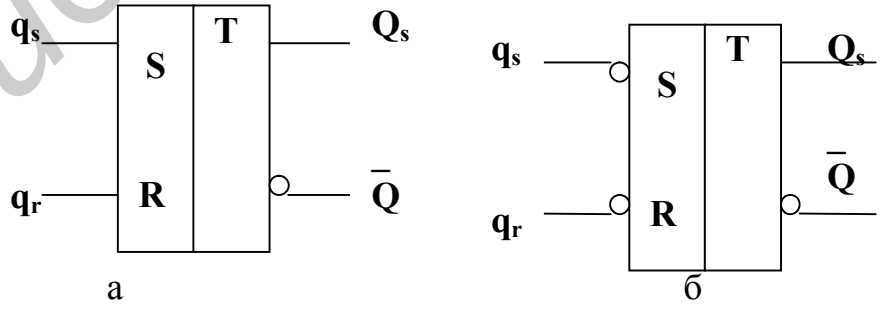
Таблицу истинности можно представить в более компактном виде, если определять конечное состояние через начальное, как это показано в табл.2.3.

Таблица 2.3

N строки	q_s	q_R	Q(t+1)
0	0	0	Q(t)
1	0	1	1
2	1	0	0
3	1	1	Не определено

На рис.2.6,б представлена реализация триггера в базисе И-НЕ, а на рис.2.7,б – его условное обозначение.

Доказательство того, что приведенная схема является триггером, имеющим вход для установки единицы и вход для установки нуля, приведено в табл.2.4. В таблице используются обозначения, аналогичные обозначениям в табл.2.1.



а б

Рис. 2.7

На наборе 1 рассматривается ситуация, когда на оба входа схемы поданы единичные значения, а Q(t) равно 1. В этой ситуации $\overline{Q}(t+1)$ будет равно 0, что не изменит начального состояния на выходе Q,
т. е. Q(t+1) будет равно 1.

На наборе 2 рассматривается ситуация, когда на схему поданы единичные значения обеих входных переменных, а Q(t) равно 0. В этой ситуации на $\overline{Q}(t+1)$ будет 1, что не изменит начального состояния на выходе Q, т.е. на Q(t+1) будет 0.

Работа схемы на наборах 1 и 2 свидетельствует о том, что при нулевых значениях входных переменных рассматриваемая схема сохраняет исходное состояние.

Формируемые выходные переменные на наборах 3 и 6 свидетельствуют о том, что рассматриваемая схема при воздействии входных переменных q_s, = 1 и q_R =0, независимо от исходного состояния (от значения Q(t)), в конечном состоянии будет иметь на своих выходах

Q(t+1)=0, $\overline{Q}(t+1)=1$,
т.е. данная комбинация сигналов переводит схему в состояние 0.

Формируемые выходные переменные на наборах 4 и 5 свидетельствуют о том, что рассматриваемая схема при воздействии значений входных переменных q_s, = 0 и q_R =1, независимо от исходного состояния (от значения Q(t)), в конечном состоянии будет иметь

Q(t+1)=1, $\overline{Q}(t+1)=0$
т.е. данная комбинация сигналов переводит схему в состояние 1.

Таблица 2.4

	1	2	3	4	5	6	7	8
q_s	1	1	1	0	0	1	0	0
q_K	1	1	0	1	1	0	0	0
Q(t)	1	0	0	0	1	1	0	1
$\overline{Q}(t)$	0	1	1	1	0	0	1	0
Q(t+1)	1	0	0	1	1	0	1?	1?
$\overline{Q}(t+1)$	0	1	1	0	0	1	1?	1?

Наборы 7 и 8 являются недопустимыми, так, при q_S = 0 и q_R =0 состояние схемы не определено, потому что на двух составляющих

Q(t+1), $\overline{Q}(t+1)$

парафазного выхода схемы появляются одинаковые единичные значения.

Таким образом, на основании приведенной таблицы можно заключить, что при различных комбинациях входных переменных q_S,q_R рассматриваемая схема может:

– хранить предыдущее состояние ($q_s = 1$, $q_R = 1$);
– устанавливаться в ноль ($q_s = 1$, $q_R = 0$);
– устанавливаться в единицу ($q_s = 0$, $q_R = 1$).

На основании изложенного можно заключить, что рассматриваемая схема является триггером. Отличием данной схемы от реализации триггера на элементах ИЛИ - НЕ является то, что по входам данный триггер воспринимает обратные значения входных переменных, что отражено на условном обозначении этой реализации триггера (см. рис.2.7,б)

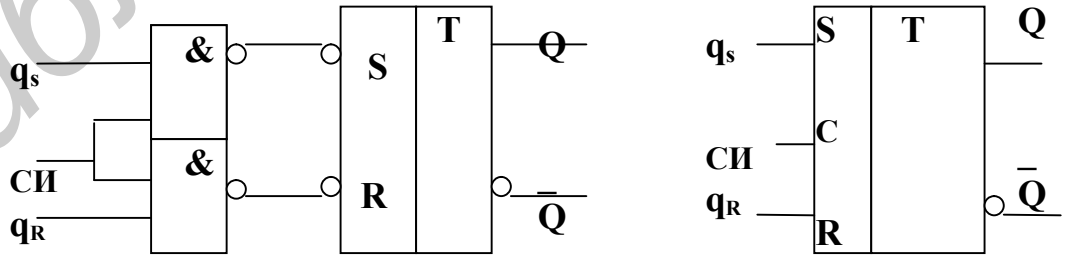
Работу данного триггера отражает таблица истинности, табл.2.5.

Таблица 2.5

N строки	q_s	q_s	Q(t+1)
0	1	1	Q(t)
1	1	0	1
2	0	1	0
3	0	0	Не определено

Синхронный триггер

Синхронный RS-триггер, помимо основных входов установки нуля и единицы, имеет дополнительный вход синхронизации С. При отсутствии сигнала на входе С триггер не реагирует на появления на его входах S и R сигналов, стремящихся изменить состояние триггера. С помощью сигнала синхронизации задается момент реакции схемы на входные сигналы. На рис.2.8,а приведена схема построения синхронного RS-триггера на базе обычного не синхронного (асинхронного) RS-триггера для случая реализации триггера на элементах И-НЕ. На рис.2.8,б приведено условное обозначение этого типа триггера. На рис.2.9 приведена временная диаграмма работы синхронного RS-триггера. Из диаграммы видно, что триггер реагирует на единичные сигналы по своим входам S и R только тогда, когда есть высокий уровень (логическая «1») синхросигнала СИ (синхроимпульс), поступающего на вход С синхронного триггера.



а б

Рис.2.8

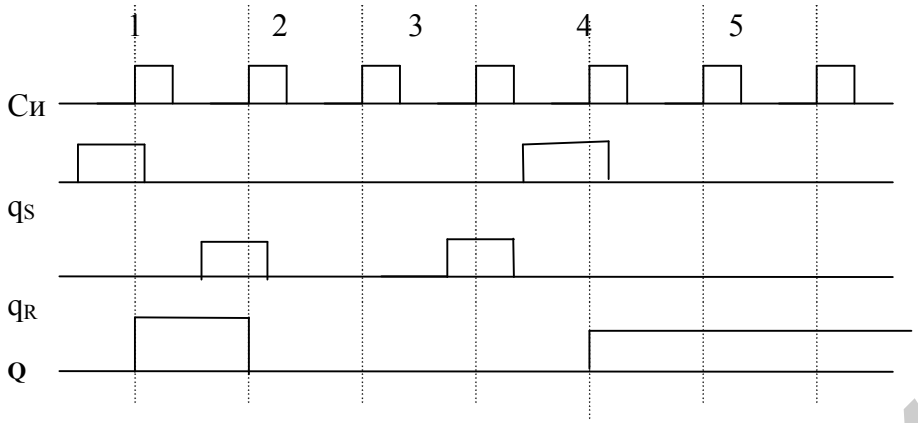


Рис. 2.9

Двухтактный RS- триггер

Двухтактный RS-триггер характеризуется тем, что у него разделены момент восприятия входных сигналов и момент изменения выходного сигнала в соответствии с комбинацией действующих входных сигналов. Построение такого триггера на базе обычного (однотактного) синхронного RS-триггер приведено на рис. 2.10, на рис. 2.11 приведено условное обозначение двухтактного RS-триггера.

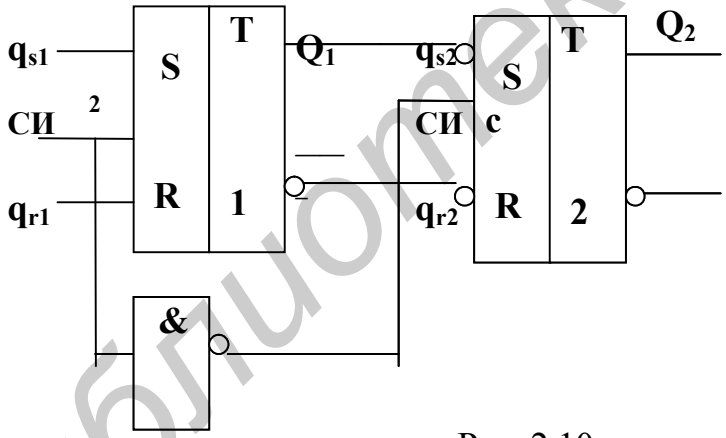


Рис. 2.10

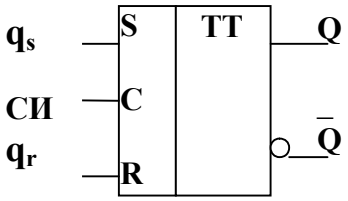


Рис. 2.11

При наличии сигнала СИ имеется разрешение по входу синхронизации первого триггера реагировать на входные сигналы q_s и q_R . В то же самое время, пока есть 1 на СИ, запрещена реакция второго триггера на сигналы, действующие

на его входах R и S. Входными сигналами для второго триггера являются сигналы на выходах первого триггера. Поэтому, несмотря на то, что при наличии сигнала СИ первый триггер может, реагируя на действующую в это время комбинацию сигналов по своим входам, изменить свое состояние, т.е. изменить свои выходные сигналы, для второго триггер, пока есть сигнал на СИ, имеет место запрет реакции на его входные сигналы. Как только на СИ появляется низкий уровень (логический «0»), разрешается восприятие вторым триггером своих входных сигналов и запрещается реакция на входы для первого триггера. В результате второй триггер установится в состояние, соответствующее состоянию первого триггера. На рис. 2.12 приведена временная диаграмма, поясняющая работу двухтактного RS-триггера. Из диаграммы видно, что второй триггер изменяет свое состояние (если комбинация входных сигналов диктует такое изменение) только тогда, когда снимается сигнал СИ. Выходные сигналы первого триггера стремятся установить во втором триггере то же состояние, что и в первом триггере.

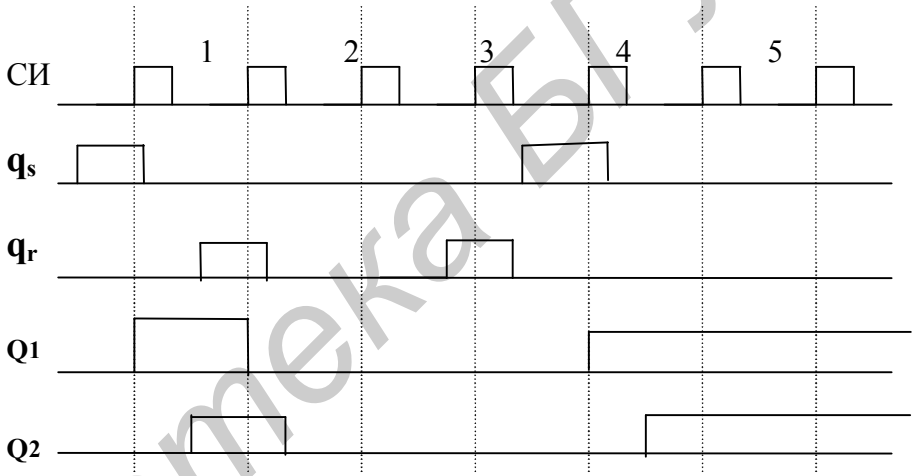


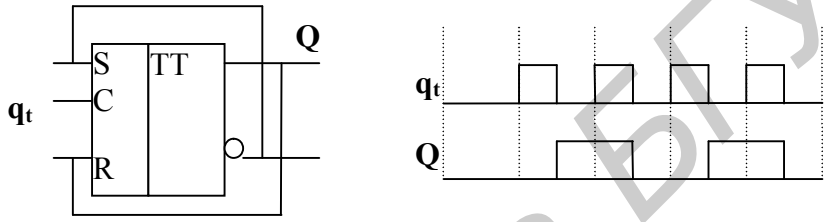
Рис.2.12

Если состояние второго триггера отличается от состояния первого, то смена состояния второго триггера происходит только в момент исчезновения 1 на СИ. Таким образом, можно заключить, что изменение состояния первого триггера может произойти только по переднему фронту сигнала СИ, а изменение второго триггера – только по заднему фронту этого сигнала. Так как выход второго триггера представляет собой выход двухтактного RS - триггера, то это означает, что изменение выходных сигналов двухтактного триггера может происходить только по заднему фронту сигнала синхронизации, поступающего на вход С этого триггера.

2.1.2.2. T-, JK-, D-триггер

Т- триггер

Т-триггер представляет собой триггер, имеющий один вход Т, поступление единичного сигнала на который переводит Т-триггер в состояние, противоположное его исходному состоянию (фигурально говоря, по каждому входному сигналу триггер «кувыркается», меняя своё состояние на противоположное). На рис.2.13 приведена реализация Т-триггера на базе двухтактного RS-триггера (а) и временная диаграмма его работы (б). Имеющиеся на схеме обратные связи создают ситуацию, при которой сигналы на входах R и S стремятся перевести триггер в состояние, противоположное текущему. Поэтому при приходе очередного сигнала q_T триггер воспринимает имеющиеся сигналы на его входах. Выходные сигналы триггера изменяются после снятия единичного сигнала на его входе q_T, так как триггер двухтактный.



а б

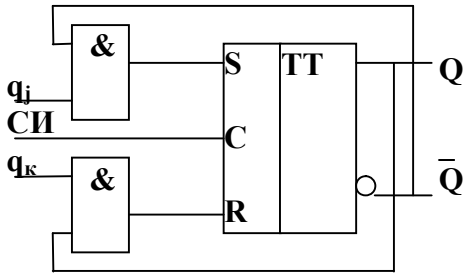
Рис. 2.13

Т-триггер можно рассматривать как счетчик, считающий по модулю 2 количество импульсов, поступающих на его вход. Действительно, если в исходном состоянии триггер находится в 0, то при поступлении на его вход нечетного количества импульсов триггер будет находиться в 1, а при четном - в 0, что соответствует суммированию по модулю 2 количества поступающих импульсов.

JK -триггер

Реализация JK-триггера и соответствующая таблица истинности приведены на рис. 2.14. Вход J - это вход установки 1, вход K - вход установки 0.

Из приведенной схемы видно, что сигнал q_J или q_K, стремящийся установить триггер соответственно в 1 или 0, поступает на соответствующий вход S или R базового RS - триггера только тогда, когда его исходное состояние противоположно тому, в которое стремится перевести JK-триггер комбинацию входных сигналов. В противном случае сигнал q_J или q_K на соответствующий вход S или R базового триггера не поступает. В связи с этим комбинация входных сигналов 1,1 не является запретной, так как в этом случае на соответствующий вход базового RS-триггера поступит только тот сигнал, который стремится установить триггер в состояние, противоположное его исходному состоянию. Этот момент отражен в таблице истинности - при комбинации входных сигналов 1,1 триггер меняет исходное состояние, то есть работает как Т-триггер.



а

	q_j	q_r	Q(t+1)
0	0	0	0
1	0	1	0
2	1	0	1
3	1	1	0
2	1	1	— Q(t)

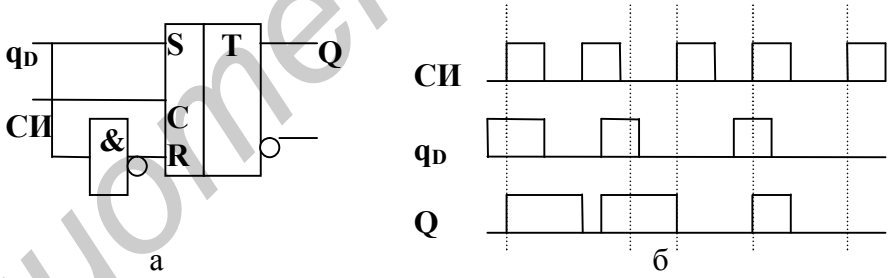
б

Рис.2.14

Таким образом, JK-триггер представляет собой универсальный триггер, объединяющий в себе свойства и RS -триггера, и Т-триггера.

D- триггер

D-триггер по-другому называют элементом задержки. Схема его реализации на базе RS-триггера и временная диаграмма его работы приведены на рис. 2.15. Использование подачи сигнала установки 1 через логику НЕ на вход установки 0 приводит к тому, что на входы R и S базового RS-триггера подаются сигналы, имеющие противоположные значения.



а б

Рис.2.15

Это означает, что когда есть 1 на входе q_D, на входах базового триггера будут сигналы: 1 - на входе S и 0 - на входе R. Поэтому по переднему фронту сигнала СИ в триггере устанавливается 1, если есть 1 на входе q_D, в противном случае в триггере будет установлен 0. Состояние, в котором находится триггер в момент заднего фронта сигнала СИ, будет сохраняться («задерживаться») до поступления очередного сигнала синхронизации СИ.

Св. план 2005, поз. 48

Учебное издание

Пешков Анатолий Тимофеевич

Организация и функционирование ЭВМ

Методическое пособие для студентов специальности
«Программное обеспечение информационных технологий»
дневной формы обучения

В 3 частях

Часть 2

Логические основы ЭВМ

Редактор Е.Н. Батурчик

Подписано в печать 23.06.2005.	Формат 60x84 1/16.	Бумага офсетная.
Гарнитура «Таймс».	Печать ризографическая.	Усл. печ. л. 2,33.
Уч.-изд. л. 2,0.	Тираж 100 экз.	Заказ 71.

Издатель и полиграфическое исполнение: Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
Лицензия на осуществление издательской деятельности №02330/0056964 от 01.04.2004.
Лицензия на осуществление полиграфической деятельности №02330/0131518 от 30.04.2004.
220013, Минск, П. Бровки, 6