

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

Кафедра систем телекоммуникаций

**П. А. Капуно**

## **ЦИФРОВЫЕ ФУНКЦИОНАЛЬНЫЕ УСТРОЙСТВА В ТЕЛЕКОММУНИКАЦИЯХ**

В 2-х частях

Часть 1

### **БАЗОВЫЕ ЦИФРОВЫЕ ФУНКЦИОНАЛЬНЫЕ УСТРОЙСТВА**

*Рекомендовано УМО по образованию в области информатики  
и радиоэлектроники в качестве учебно-методического пособия  
для специальностей 1-45 01 01 «Многоканальные системы телекоммуникаций»,  
1-45 01 02 «Системы радиосвязи, радиовещания и телевидения»  
и направлений специальности 1-45 01 01-01 «Инфокоммуникационные  
технологии (системы телекоммуникаций)»,  
1-45 01 01-03 «Инфокоммуникационные технологии (системы  
телекоммуникаций специального назначения)»,  
1-45 01 01-04 «Инфокоммуникационные технологии  
(цифровое теле- и радиовещание)»*

Минск БГУИР 2014

УДК 004.312:621.391(076)  
ББК 32.973я73+32.811я73  
К20

Рецензенты:

кафедра связи учреждения образования  
«Военная академия Республики Беларусь»  
(протокол №89 от 31.03.2014);

заведующий кафедрой телекоммуникационных систем  
учреждения образования «Высший государственный колледж связи»,  
кандидат технических наук, доцент К. И. Пирогов

**Капуρο, П. А.**

К20

Цифровые функциональные устройства в телекоммуникациях.  
В 2 ч. Ч. 1 : Базовые цифровые функциональные устройства :  
учеб.-метод. пособие / П. А. Капуρο. – Минск : БГУИР, 2014. – 64 с. : ил.

ISBN 978-985-543-022-4.

Изложены основы теории алгебры логики, логических функций и цифровых автоматов, позволяющие простейшими методами решать задачи аналитического описания работы цифровых устройств и их проектирования.

Рассматриваются варианты схемной реализации типовых цифровых функциональных устройств. Приведены методики проектирования схем комбинационного и последовательностного типов.

**УДК 004.312:621.391(076)**  
**ББК 32.973я73+32.811я73**

**ISBN 978-985-543-022-4 (ч. 1)**  
**ISBN 978-985-543-021-7**

© Капуρο П. А., 2014  
© УО «Белорусский государственный университет информатики и радиоэлектроники», 2014

# 1 ОСНОВЫ АЛГЕБРЫ ЛОГИКИ

## 1.1 Аксиомы, законы и правила алгебры логики

Основы алгебры логики были заложены в середине XIX века трудами английского математика Дж. Буля, по имени которого она называется также булевой алгеброй. Понимание принципов, лежащих в ее основе, важно для овладения методами описания и проектирования цифровых устройств. Начало использованию алгебры логики для синтеза переключательных (релейных) схем было положено в 1938 г. работами американского ученого К. Шеннона.

В алгебре логики рассматриваются переменные, которые могут принимать только два значения – 0 и 1. В дальнейшем переменные будем обозначать латинскими буквами (при необходимости с цифровыми индексами):  $x, y, B, A_2$ .

В алгебре логики определены отношение эквивалентности ( $=$ ) и три операции: *дизъюнкция* (операция ИЛИ, логическое сложение), обозначаемая знаком  $\vee$  (например  $X \vee Y$ ); *конъюнкция* (операция И, логическое умножение), обозначаемая знаками  $\wedge$ ,  $\&$  или точкой (которую можно опускать, например,  $X \cdot A_3 = XA_3$ , но если это не приводит к неоднозначности определения места операции или переменных); *отрицание* (операция НЕ, инверсия), обозначаемое чертой над переменными или над элементами 0 и 1 (например  $\overline{A_3}, \overline{0}$ ).

Отношение эквивалентности удовлетворяет следующим свойствам:

$X = X$  – *рефлексивность*;

если  $X = Y$ , то  $Y = X$  – *симметричность*;

если  $X = Y$  и  $Y = Z$ , то  $X = Z$  – *транзитивность*.

Из отношения эквивалентности следует принцип подстановки: если  $X = Y$ , то в любой формуле, содержащей  $X$ , вместо  $X$  можно подставить  $Y$ , и в результате будет получена эквивалентная формула.

Алгебра логики определяется следующей системой аксиом:

$$\left. \begin{array}{l} X = 0, \text{ если } X \neq 1 \\ X = 1, \text{ если } X \neq 0 \end{array} \right\} \quad (1.1)$$

$$\left. \begin{array}{l} 0 \vee 0 = 0 \\ 0 \vee 1 = 1 \\ 1 \vee 0 = 1 \\ 1 \vee 1 = 1 \end{array} \right\} (1.2) \qquad \left. \begin{array}{l} 0 \cdot 0 = 0 \\ 0 \cdot 1 = 0 \\ 1 \cdot 0 = 0 \\ 1 \cdot 1 = 1 \end{array} \right\} (1.3) \qquad \left. \begin{array}{l} \bar{0} = 1 \\ \bar{1} = 0 \end{array} \right\} (1.4)$$

Аксиома (1.1) является утверждением того, что в алгебре логики рассматриваются только двоичные переменные, аксиомы (1.2), (1.3) определяют свойства операций дизъюнкции и конъюнкции, а аксиома (1.4) – операции отрицания.

С помощью аксиом алгебры логики можно доказать целый ряд тождеств и теорем. Одним из эффективных методов доказательства теорем является *метод перебора* всех значений переменных: если теорема истинна, то с учетом (1.2)...(1.4) уравнение, формулирующее утверждение теоремы, должно быть истинно при подстановке любых значений переменных в обе его части. Метод перебора не слишком трудоемок, т. к. переменные могут иметь только два значения: 0 и 1.

Если в логическое выражение входят операции дизъюнкции и конъюнкции, то следует соблюдать следующий порядок выполнения операций: сначала выполняется операция конъюнкции (она имеет приоритет), затем операция дизъюнкции. В сложных логических выражениях для задания иного порядка выполнения операций используются скобки.

Операции логического сложения (дизъюнкции) и логического умножения (конъюнкции) для случая одной переменной  $x$  обладают следующими свойствами:

$$\left. \begin{array}{l} x \vee 0 = x \\ x \vee 1 = 1 \\ x \vee x = x \\ x \vee \bar{x} = 1 \end{array} \right\} (1.5) \qquad \left. \begin{array}{l} x \cdot 0 = 0 \\ x \cdot 1 = x \\ x \cdot x = x \\ x \cdot \bar{x} = 0 \end{array} \right\} (1.6)$$

Для случая двух и более переменных справедливы следующие законы:

- переместительные (коммутативные):

$$\left. \begin{array}{l} x \cdot y = y \cdot x \\ x \vee y = y \vee x \end{array} \right\}; \qquad (1.7)$$

- сочетательные (ассоциативные):

$$\left. \begin{aligned} x \cdot (y \cdot z) &= (x \cdot y) \cdot z = (x \cdot z) \cdot y \\ x \vee (y \vee z) &= (x \vee y) \vee z = (x \vee z) \vee y \end{aligned} \right\}; \quad (1.8)$$

- распределительные (дистрибутивные):

$$\left. \begin{aligned} x \cdot (y \vee z) &= x \cdot y \vee x \cdot z \\ x \vee y \cdot z &= (x \vee y)(x \vee z) \end{aligned} \right\}; \quad (1.9)$$

- двойственности (теоремы де Моргана):

$$\left. \begin{aligned} \overline{x \cdot y} &= \bar{x} \vee \bar{y} \\ \overline{x \vee y} &= \bar{x} \cdot \bar{y} \end{aligned} \right\}; \quad (1.10)$$

- двойного отрицания:

$$\overline{(\bar{x})} = \bar{\bar{x}} = x; \quad (1.11)$$

- поглощения:

$$\left. \begin{aligned} x \vee x \cdot y &= x \\ x \cdot (x \vee y) &= x \end{aligned} \right\}; \quad (1.12)$$

- операции склеивания:

$$\left. \begin{aligned} x \cdot y \vee x \cdot \bar{y} &= x \\ (x \vee y) \cdot (x \vee \bar{y}) &= x \end{aligned} \right\}; \quad (1.13)$$

- операции обобщенного склеивания:

$$\left. \begin{aligned} x \cdot y \vee \bar{x} \cdot z \vee y \cdot z &= x \cdot y \vee \bar{x} \cdot z \\ (x \vee y) \cdot (\bar{x} \vee z) \cdot (y \vee z) &= (x \vee y) \cdot (\bar{x} \vee z) \end{aligned} \right\}; \quad (1.14)$$

$$\left. \begin{aligned} x \vee \bar{x} \cdot y &= x \vee y \\ x \cdot (\bar{x} \vee y) &= x \cdot y \end{aligned} \right\}. \quad (1.15)$$

Некоторые законы и тождества алгебры логики имеют особое значение, т. к. позволяют упрощать логические выражения. Например, в выражениях (1.5), (1.6), (1.11)...(1.15) правая часть выражения проще левой, поэтому, проведя в логических выражениях соответствующие преобразования, можно добиться существенного их упрощения. Наиболее часто для преобразования логических выражений с целью их упрощения используются тождества (1.12)...(1.15).

Операция сложения по модулю 2 (исключающее ИЛИ, логическая неравнозначность) обозначается символом  $\oplus$  и определяется соотношением

$$x \oplus y = \bar{x} \cdot y \vee x \cdot \bar{y} = (\bar{x} \vee \bar{y}) \cdot (x \vee y). \quad (1.16)$$

Используя аксиомы алгебры логики, легко убедиться, что

$$0 \oplus 0 = 1 \oplus 1 = 0; \quad 1 \oplus 0 = 0 \oplus 1 = 1. \quad (1.17)$$

Из этих соотношений следует, что значение  $x \oplus y$ :

- совпадает со значением младшего разряда суммы двух одноразрядных двоичных чисел  $x$  и  $y$ ;
- совпадает с результатом дизъюнкции переменных  $x$  и  $y$ , за исключением случая, когда две переменные равны 1;
- равно 1, когда значения переменных не равны друг другу.

Перечисленные свойства операции  $x \oplus y$  и определяют ее названия. Операция суммирования по модулю 2 коммутативна, ассоциативна и дистрибутивна относительно операции конъюнкции:

$$\left. \begin{aligned} x \oplus y &= y \oplus x \\ x \oplus (y \oplus z) &= (x \oplus y) \oplus z = (x \oplus z) \oplus y \\ x \cdot (y \oplus z) &= x \cdot y \oplus x \cdot z \end{aligned} \right\}. \quad (1.18)$$

Для операции сложения по модулю 2 также справедливы следующие тождества:

$$\left. \begin{aligned} x \oplus 0 &= x \\ x \oplus 1 &= \bar{x} \\ x \oplus x &= 0 \\ x \oplus \bar{x} &= 1 \\ \overline{x \oplus y} &= \bar{x} \cdot \bar{y} \vee x \cdot y = (\bar{x} \vee y) \cdot (x \vee \bar{y}) = \bar{x} \oplus y = x \oplus \bar{y} \end{aligned} \right\}. \quad (1.19)$$

## 1.2 Логические функции

### 1.2.1 Общие понятия

Любое логическое выражение, составленное из  $n$  переменных  $x_n, \dots, x_1$  с помощью конечного числа операций алгебры логики, можно рассматривать как некоторую функцию  $n$  переменных. В соответствии с аксиомами (1.1)...(1.4) функция может принимать в зависимости от значений переменных только два значения: 0 и 1. Такие функции являются весьма удобным инструментом для описания, анализа и синтеза логических схем, выходные сигналы которых характеризуются лишь двумя уровнями напряжения: высоким (1) и низким (0). В связи с этим логические функции называются переключательными.

Для функций  $n$  переменных  $x_n, \dots, x_1$  используется общее обозначение  $F(v) = F(x_n, \dots, x_1)$ , где  $v = (x_n, \dots, x_1)$ , т. е. совокупность переменных  $x_n, \dots, x_1$  можно рассматривать как  $n$ -мерный вектор. Так как каждая переменная  $x_p$  ( $p = 1, 2, \dots, n$ ) может принимать только два значения (0 и 1), то число всех возможных комбинаций значений  $x_n, \dots, x_1$  конечно. В общем виде конкретное значение переменной  $x_p$  (0 или 1) будем обозначать через  $e_p$ .

Областью определения функции  $n$  переменных  $x_n, \dots, x_1$  является совокупность точек  $n$ -мерного пространства, причем каждая из точек задается определенной комбинацией значений этих переменных:

$$x_n = e_n, \dots, x_p = e_p, \dots, x_1 = e_1,$$

где  $e_p = 0$  или  $1$  ( $p = 1, 2, \dots, n$ ).

Обозначим точки, задающие область определения функции  $F(v)$ , через

$$v_i = (e_n, \dots, e_p, \dots, e_1).$$

Если представить эту комбинацию значений переменных  $(e_n, \dots, e_p, \dots, e_1)$  как число, записанное в двоичном коде, то оказывается возможным пронумеровать все точки области определения логической функции с помощью  $n$ -разрядных двоичных чисел  $(e_n \dots e_p \dots e_1)_2$  или с помощью эквивалентных им десятичных чисел  $i$ . Так как имеется  $2^n$  различных  $n$ -разрядных двоичных чисел, то область определения функции  $n$  переменных будет содержать  $2^n$  точек:

$$v \in \{v_0, v_1, \dots, v_{2^n - 1}\}.$$

Для задания функции  $F(v)$  следует указать ее значения во всех точках области определения, т. е. следует задать значения  $F(v_i) = 0$  или  $1$ , где  $i = 0, 1, \dots, 2^n - 1$ . Каждой конкретной функции  $n$  переменных можно поставить в соответствие  $2^n$ -разрядное число, составленное из значений  $F(v_i) = 0$  или  $1$ , которые она принимает в  $2^n$  точках области определения. Так как имеется всего  $2^{2^n}$  различных  $2^n$ -разрядных двоичных чисел, то и число различных функций  $n$  переменных равно  $2^{2^n}$ .

Функции  $n$  переменных могут зависеть не от всех переменных  $x_n, \dots, x_1$ . Такие функции называются вырожденными. В частности, функция  $F_0(v)$ , равная нулю, и функция  $F_1(v)$ , равная единице во всех точках  $v_i$  ( $i = 0, 1, \dots, 2^n - 1$ ) области определения, не зависят ни от одной переменной. Эти функции называются *константой ноль* и *константой единица* соответственно.

Функция  $n$  переменных  $F(v)$  называется *полностью определенной*, если ее значения  $F(v_i) = a_i = 0$  или  $1$  заданы во всех  $2^n$  точках  $v_i$  области определения. Если же значение функции не задано хотя бы в одной точке  $v_i$ , то она называется *не полностью определенной*. Не определенное в точке  $v_i$  значение функции описывается символом  $c_i = *$  или  $\Phi$  ( $\Phi$  – совмещенные символы  $0$  и  $1$ , что указывает на неопределенность значения функции), т. е., если в точке  $v_i$  значение функции не задано, то  $F(v_i) = c_i$ .

Не полностью определенные функции можно доопределять произвольным способом, полагая  $c_i = 0$  или  $1$ . Если значения функции не заданы в  $m$  точках, то функцию можно доопределить  $2^m$  способами, т. к. имеется  $2^m$  различных  $m$ -разрядных двоичных чисел, соответствующих различным способам доопределения функции в  $m$  точках. Таким образом, не определенной в  $m$  точках функции соответствует группа из  $2^m$  полностью определенных функций.

Так как область определения любой функции  $n$  переменных конечна ( $2^n$  точек), она может быть задана таблицей значений  $F(v_i) = a_i = 0$  или  $1$  (или  $c_i$ ), которые она принимает в точках  $v_i$ , где  $i = 0, 1, \dots, 2^n - 1$ . Такие таблицы называются *таблицами истинности*.

Второй способ задания логической функции – аналитический, при котором функция задается в виде логического выражения, получаемого путем логических преобразований с помощью законов и правил булевой алгебры.



### 1.2.2 Функции двух переменных $F(x_2, x_1)$

Область определения таких функций состоит из четырех точек ( $2^n = 2^2 = 4$ ):  $v_0 = (0, 0)$ ,  $v_1 = (0, 1)$ ,  $v_2 = (1, 0)$ ,  $v_3 = (1, 1)$ . Общее количество функций двух переменных равно 16 ( $2^{2^n} = 2^{2^2} = 2^4 = 16$ ). Значения всех возможных функций  $F(x_2, x_1)$  представлены в таблице 1.1, которая является совмещенной таблицей истинности функций  $F_0 \dots F_{15}$  двух переменных. В левой части таблицы перечислены все возможные сочетания значений переменных  $x_1$  и  $x_2$ , в правой части приведены значения функций  $F_0 \dots F_{15}$  при этих сочетаниях переменных.

Таблица 1.1 – Таблицы истинности функций двух переменных

$i$	$x_2$	$x_1$	$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$	$F_9$	$F_{10}$	$F_{11}$	$F_{12}$	$F_{13}$	$F_{14}$	$F_{15}$
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	0	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

Каждая функция  $F_0 \dots F_{15}$  обозначает одну из 16 возможных логических операций над двумя переменными  $x_1$  и  $x_2$ , имеет собственное название и аналитическое описание:

$$F_0 = 0 \quad (F_{15} = 1) \text{ – константа ноль (единица);}$$

$$F_1 = x_1 \cdot x_2 \text{ – конъюнкция, логическое умножение, И;}$$

$$F_2 = \overline{x_1} \cdot x_2 \quad (F_4 = x_1 \cdot \overline{x_2}) \text{ – запрет } x_1 \text{ (} x_2 \text{);}$$

$$F_3 = x_2 \quad (F_5 = x_1) \text{ – повторение } x_2 \text{ (} x_1 \text{);}$$

$$F_6 = \overline{x_1} \cdot x_2 \vee x_1 \cdot \overline{x_2} = x_1 \oplus x_2 \text{ – неэквивалентность, сложение по модулю}$$

2, исключающее ИЛИ;

$$F_7 = x_1 \vee x_2 \text{ – дизъюнкция, логическое сложение, ИЛИ;}$$

$$F_8 = \overline{x_1 \vee x_2} \text{ – функция Пирса, ИЛИ-НЕ;}$$

$$F_9 = \overline{x_1} \cdot \overline{x_2} \vee x_1 \cdot x_2 = \overline{x_1 \oplus x_2} \text{ – эквивалентность, исключающее ИЛИ-НЕ;}$$

$$F_{10} = \overline{x_1} \quad (F_{12} = \overline{x_2}) \text{ – отрицание } x_1 \text{ (} x_2 \text{), инверсия } x_1 \text{ (} x_2 \text{), НЕ;}$$

$$F_{11} = \overline{x_1} \vee x_2 \quad (F_{13} = \overline{x_2} \vee x_1) \text{ – импликация } x_1 \text{ (} x_2 \text{);}$$

$$F_{14} = \overline{x_1 \cdot x_2} \text{ – функция Шеффера, И-НЕ.}$$

Функции двух переменных исключительно важны в силу того, что любая функция  $n$  переменных может быть получена из них методом суперпозиции – подстановкой этих функций вместо переменных в другие функции. Такая подстановка возможна на основании того, что области значений функций и переменных совпадают (0 и 1). Наибольшее применение получили следующие невырожденные функции двух переменных  $x_2$  и  $x_1$ :  $F_1$  (конъюнкция, И),  $F_6$  (сумма по модулю два, исключающее ИЛИ),  $F_7$  (дизъюнкция, ИЛИ),  $F_8$  (ИЛИ-НЕ) и  $F_{14}$  (И-НЕ).

### 1.2.3 Принцип и закон двойственности

Алгебра логики обладает замечательным свойством, которое называется принципом двойственности: если имеет место тождество  $F(v, 0, 1/\ \&, \vee) = G(v, 0, 1/\ \&, \vee)$ , где  $v = (x_n, \dots, x_1)$ , то справедливо также тождество  $F(v, 1, 0/\ \vee, \&) = G(v, 1, 0/\ \vee, \&)$ , т. е., если в каком-либо тождестве произвести взаимную замену символов 0 и 1 (при их наличии) и операций дизъюнкции и конъюнкции, то в результате также будет получено тождество. Два тождества, связанные между собой таким образом, называются *двойственными*. Истинность самого принципа двойственности не доказывается, т. к. данный принцип является внутренним свойством алгебры логики (заключен в ее аксиомах).

Законы двойственности (теоремы де Моргана (1.10)) определяют способ отыскания инверсных функций, представляющих собой дизъюнкцию и конъюнкцию двух переменных. К. Шеннон предложил обобщение этих теорем, позволяющее отыскивать инверсию любой функции  $F(v)$ . Закон двойственности имеет вид

$$\overline{F(v/\ \&, \vee)} = F(\bar{v}/\ \vee, \&), \quad (1.20)$$

где  $v = (x_n, \dots, x_1)$ ,  $\bar{v} = (\bar{x}_n, \dots, \bar{x}_1)$ , т. е. инверсию любой функции  $F(v)$  можно получить взаимной заменой переменных  $x_p$  и их инверсий  $\bar{x}_p$  ( $p = 1, 2, \dots, n$ ) и операций дизъюнкции и конъюнкции.

### 1.2.4 Теоремы разложения

В теории логических функций важное значение имеет теорема разложения: любую функцию  $F(v)$  можно разложить по переменной  $x_p$  в форме

$$F(x_n, \dots, x_p, \dots, x_1) = x_p \cdot F(x_n, \dots, 1, \dots, x_1) \vee \bar{x}_p \cdot F(x_n, \dots, 0, \dots, x_1). \quad (1.21)$$

Теорема легко доказывается методом перебора.

Используя принцип двойственности, можно также записать

$$F(x_n, \dots, x_p, \dots, x_1) = (x_p \vee F(x_n, \dots, 0, \dots, x_1)) \cdot (\bar{x}_p \vee F(x_n, \dots, 1, \dots, x_1)). \quad (1.22)$$

С теоремами разложения связаны тождества:

$$\left. \begin{aligned} x_p \cdot F(x_n, \dots, x_p, \dots, x_1) &= x_p \cdot F(x_n, \dots, 1, \dots, x_1) \\ \bar{x}_p \cdot F(x_n, \dots, x_p, \dots, x_1) &= \bar{x}_p \cdot F(x_n, \dots, 0, \dots, x_1) \end{aligned} \right\}, \quad (1.23)$$

$$\left. \begin{aligned} x_p \vee F(x_n, \dots, x_p, \dots, x_1) &= x_p \vee F(x_n, \dots, 0, \dots, x_1) \\ \bar{x}_p \vee F(x_n, \dots, x_p, \dots, x_1) &= \bar{x}_p \vee F(x_n, \dots, 1, \dots, x_1) \end{aligned} \right\}. \quad (1.24)$$

Теоремы разложения и связанные с ними тождества являются удобным инструментом для преобразования и упрощения логических выражений. В частности, разложив логическую функцию, выполняемую каким-либо цифровым устройством, по переменной, описывающей состояние какого-либо сигнала управления, можно определить реакции этого устройства на наличие и отсутствие этого сигнала.

### 1.2.5 Первичные термы, минтермы и макстермы

Переменные  $x_p$  и их инверсии  $\bar{x}_p$  в теории алгебры логики называются *первичными термами*, для которых используется символическое обозначение:

$$x_p^{e_p} = \overline{e_p} \cdot \overline{x_p} \vee e_p x_p = \overline{e_p} \oplus x_p, \quad (1.25)$$

где  $e_p = 0$  или  $1$ . Данное символическое обозначение объединяет в одном символе  $x_p^{e_p}$  оба первичных терма  $x_p$  и  $\bar{x}_p$ . Действительно, при подстановке в (1.25) значений  $e_p = 0$  и  $1$  получаем

$$x_p^{e_p} = \begin{cases} \bar{x}_p, & \text{если } e_p = 0, \\ x_p, & \text{если } e_p = 1. \end{cases}$$

*Минимальным термом (минтермом, конституентой единицы)* называется функция  $n$  переменных (конъюнкция первичных термов):

$$K_i(v) = x_n^{e_n} \cdot x_{n-1}^{e_{n-1}} \cdot \dots \cdot x_1^{e_1} = \prod_{p=1}^n x_p^{e_p}, \quad (1.26)$$

где  $v = (x_n, \dots, x_1)$ ,  $e_p = 0$  или  $1$ ,  $i = (e_n \dots e_1)_2$ . Из данного определения следует, что существует  $2^n$  различных минтермов  $n$  переменных, т. к. имеется  $2^n$  различных  $n$ -разрядных двоичных чисел  $i = 0, 1, \dots, 2^n - 1$ .

Минтермы обладают следующими свойствами:

$$K_i(v) = \begin{cases} 1, & \text{если } v = v_i, \\ 0, & \text{если } v = v_j \neq v_i; \end{cases} \quad (1.27)$$

$$K_i(v) \cdot K_j(v) \equiv 0, \text{ если } i \neq j; \quad (1.28)$$

$$\sum_{i=0}^{2^n-1} K_i(v) \equiv 1. \quad (1.29)$$

Свойство (1.27) соответствует тому, что любой минтерм  $K_i(v)$  равен 1 только в одной точке  $v_i$  области определения.

Все минтермы двух переменных  $x_1$  и  $x_2$ :

$$K_0(v) = x_2^0 \cdot x_1^0 = \bar{x}_2 \cdot \bar{x}_1;$$

$$K_1(v) = x_2^0 \cdot x_1^1 = \bar{x}_2 \cdot x_1;$$

$$K_2(v) = x_2^1 \cdot x_1^0 = x_2 \cdot \bar{x}_1;$$

$$K_3(v) = x_2^1 \cdot x_1^1 = x_2 \cdot x_1.$$

Аналогично можно записать минтерм большего числа переменных. Например, пусть  $n = 4$ ,  $i = 5 = (0101)_2$ , тогда

$$K_5(v) = x_4^0 \cdot x_3^1 \cdot x_2^0 \cdot x_1^1 = \bar{x}_4 \cdot x_3 \cdot \bar{x}_2 \cdot x_1.$$

*Максимальным термом (макстермом, конституентой ноля)* называется функция  $n$  переменных (дизъюнкция инверсий первичных термов):

$$M_i(v) = \overline{K_i(v)} = \overline{\prod_{p=1}^n x_p^{e_p}} = \sum_{p=1}^n \overline{x_p^{e_p}} = \overline{x_n^{e_n}} \vee \overline{x_{n-1}^{e_{n-1}}} \vee \dots \vee \overline{x_1^{e_1}}, \quad (1.30)$$

где  $v = (x_n, \dots, x_1)$ ,  $e_p = 0$  или  $1$ ,  $i = (e_n \dots e_1)_2$ .

Макстермы обладают следующими свойствами:

$$M_i(v) = \begin{cases} 0, & \text{если } v = v_i, \\ 1, & \text{если } v = v_j \neq v_i; \end{cases} \quad (1.31)$$

$$M_i(v) \vee M_j(v) \equiv 1, \text{ если } i \neq j; \quad (1.32)$$

$$\prod_{i=0}^{2^n-1} M_i(v) \equiv 0. \quad (1.33)$$

Свойства макстермов могут быть получены из свойств минтермов на основании (1.30). Макстермы – функции, равные нулю только в одной точке  $v_i$  области определения, состоящей из  $2^n$  точек.

Все макстермы двух переменных  $x_1$  и  $x_2$ :

$$M_0(v) = \overline{x_2^0} \vee \overline{x_1^0} = \overline{x_2} \vee \overline{x_1} = x_2 \vee x_1;$$

$$M_1(v) = \overline{x_2^0} \vee x_1^1 = \overline{x_2} \vee x_1 = x_2 \vee \overline{x_1};$$

$$M_2(v) = \overline{x_2^1} \vee \overline{x_1^0} = \overline{x_2} \vee \overline{x_1} = \overline{x_2} \vee x_1;$$

$$M_3(v) = \overline{x_2^1} \vee x_1^1 = \overline{x_2} \vee x_1.$$

Запишем макстерм  $M_5(v)$  для  $n = 4$ :

$$M_5(v) = \overline{x_4^0} \vee \overline{x_3^1} \vee \overline{x_2^0} \vee \overline{x_1^1} = \overline{x_4} \vee \overline{x_3} \vee \overline{x_2} \vee \overline{x_1} = x_4 \vee \overline{x_3} \vee x_2 \vee x_1.$$

Минтермы (макстермы) представляют собой функции, принимающие минимальное (максимальное) значение из значений своих первичных термов  $x_p^{e_p}$ , т. е. если хотя бы один из первичных термов  $x_p^{e_p}$  равен 0 (1), то и минтерм (макстерм) равен 0 (1).

### 1.2.6 Формы аналитического представления логических функций

Логические функции, которые считаются полностью определенными, при аналитической записи могут быть представлены различными формами.

*СДНФ – совершенная дизъюнктивная нормальная форма.* Теорему разложения (1.20) для функции  $n$  переменных можно использовать  $n$  раз, т. е. функцию можно разложить по всем  $n$  переменным  $x_p$ , где  $p = 1, 2, \dots, n$ . В качестве примера разложим функцию  $F(v) = F(x_2, x_1)$  двух переменных  $x_2$  и  $x_1$ . По теореме разложения получим

$$\begin{aligned}
F(x_2, x_1) &= x_2 \cdot F(1, x_1) \vee \bar{x}_2 \cdot F(0, x_1) = \\
&= x_1 [x_2 \cdot F(1, 1) \vee \bar{x}_2 \cdot F(0, 1)] \vee \bar{x}_1 [x_2 \cdot F(1, 0) \vee \bar{x}_2 \cdot F(0, 0)] = \\
&= \bar{x}_2 \cdot \bar{x}_1 \cdot F(0, 0) \vee \bar{x}_2 \cdot x_1 \cdot F(0, 1) \vee x_2 \cdot \bar{x}_1 \cdot F(1, 0) \vee x_2 \cdot x_1 \cdot F(1, 1) = \\
&= x_2^0 \cdot x_1^0 \cdot F(0, 0) \vee x_2^0 \cdot x_1^1 \cdot F(0, 1) \vee x_2^1 \cdot x_1^0 \cdot F(1, 0) \vee x_2^1 \cdot x_1^1 \cdot F(1, 1) = \\
&= \sum_{i=0}^3 x_2^{e_2} \cdot x_1^{e_1} \cdot F(e_2, e_1) = \sum_{i=0}^3 K_i(v) \cdot F(v_i) = \sum_{i=0}^3 a_i \cdot K_i(v),
\end{aligned}$$

где  $v_i = (e_2, e_1)$  –  $i$ -я точка области определения,  $i = (e_2 e_1)_2$ ;

$K_i(v) = x_2^{e_2} \cdot x_1^{e_1}$  – минтерм двух переменных  $x_2$  и  $x_1$ ;

$F(v_i) = a_i = 0$  или  $1$  – значение функции в  $i$ -й точке области определения.

Такая форма представления функции двух переменных и называется СДНФ. Разложение функции  $n$  переменных будет представлять собой дизъюнкцию  $2^n$  членов вида  $x_n^{e_n} \cdot \dots \cdot x_1^{e_1} \cdot F(e_n, \dots, e_1) = K_i(v) \cdot F(v_i) = a_i \cdot K_i(v)_i$ :

$$F(v) = \sum_{i=0}^{2^n-1} a_i \cdot K_i(v). \quad (1.34)$$

Выражение (1.34) представляет собой СДНФ функции  $n$  переменных.

Так как значения функции  $a_i = 0$  или  $1$ , то  $a_i \cdot K_i(v) = 0$ , если  $a_i = 0$ , и  $a_i \cdot K_i(v) = K_i(v)$ , если  $a_i = 1$ . Поэтому СДНФ можно представить в виде

$$F(v) = \sum_{i1} K_{i1}(v), \quad (1.35)$$

где  $i1$  – номера точек области определения, в которых функция  $F(v)$  равна 1.

СДНФ записи функции легко получить по ее таблице истинности. В качестве примера рассмотрим функцию трех переменных  $x_3$ ,  $x_2$  и  $x_1$ , заданную таблицей истинности (таблица 1.2). Из нее следует, что  $a_0 = a_1 = a_4 = a_6 = 0$ ,  $a_2 = a_3 = a_5 = a_7 = 1$ , поэтому

$$\begin{aligned}
F(v) &= K_2(v) \vee K_3(v) \vee K_5(v) \vee K_7(v) = \\
&= \bar{x}_3 \cdot x_2 \cdot \bar{x}_1 \vee \bar{x}_3 \cdot x_2 \cdot x_1 \vee x_3 \cdot \bar{x}_2 \cdot x_1 \vee x_3 \cdot x_2 \cdot x_1.
\end{aligned}$$

Таблица 1.2 – Таблица истинности функции трех переменных

$i$	$x_3$	$x_2$	$x_1$	$F(v)$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

*СКНФ* – совершенная конъюнктивная нормальная форма. Такую форму записи функции  $n$  переменных  $F(v)$  можно получить на основании двойственной теоремы разложения (1.22). Однако СКНФ можно получить более простым способом – записав СДНФ инверсии функции  $\overline{F(v)}$ . Инверсная функция в каждой точке  $v_i$  области определения имеет инверсные значения  $\overline{a_i}$  по отношению к значениям  $a_i$  самой функции, т. е. если  $F(v_i) = a_i$ , то  $\overline{F(v_i)} = \overline{a_i}$ .

Тогда на основании (1.34) СДНФ инверсной функции

$$\overline{F(v)} = \sum_{i=0}^{2^n-1} \overline{a_i} \cdot K_i(v). \quad (1.36)$$

Отсюда на основании закона двойственности (1.20) следует

$$F(v) = \overline{\sum_{i=0}^{2^n-1} \overline{a_i} \cdot K_i(v)} = \prod_{i=0}^{2^n-1} (\overline{\overline{a_i} \cdot K_i(v)}) = \prod_{i=0}^{2^n-1} (a_i \vee \overline{K_i(v)}). \quad (1.37)$$

Используя определение макстермов (1.30), получаем СКНФ представления функции  $n$  переменных:

$$F(v) = \prod_{i=0}^{2^n-1} (a_i \vee M_i(v)). \quad (1.38)$$

Так как значения функции  $a_i = 0$  или  $1$ , то  $a_i \vee M_i(v) = 1$ , если  $a_i = 1$ , и  $a_i \vee M_i(v) = M_i(v)$ , если  $a_i = 0$ . Поэтому СКНФ можно представить в виде

$$F(v) = \prod_{i \in I} M_{i0}(v), \quad (1.39)$$

где  $i \in I$  – номера точек области определения, в которых функция  $F(v)$  равна 0.

Для примера рассмотрим функцию трех переменных  $x_3, x_2$  и  $x_1$ , заданную таблицей истинности (см. таблицу 1.2). Так как  $a_0 = a_1 = a_4 = a_6 = 0$ , то на основании (1.39) СКНФ функции имеет вид

$$\begin{aligned} F(v) &= M_0(v) \cdot M_1(v) \cdot M_4(v) \cdot M_6(v) = \\ &= (x_3 \vee x_2 \vee x_1) \cdot (x_3 \vee x_2 \vee \bar{x}_1) \cdot (\bar{x}_3 \vee x_2 \vee x_1) \cdot (\bar{x}_3 \vee \bar{x}_2 \vee x_1). \end{aligned}$$

*Совершенные нормальные формы в базисах И-НЕ и ИЛИ-НЕ.* Совокупность элементарных функций, с помощью которых можно записать любую функцию  $F(v)$ , называется *функционально полной системой функций* или *базисом*. Из выражений (1.35) и (1.39) следует, что для представления любой функции  $F(v)$  в СДНФ и СКНФ достаточно использовать только операции (функции) И, ИЛИ и НЕ (операция НЕ нужна для получения первичных термов  $\bar{x}_p$ , входящих в минтермы и макстермы), т. е. совокупность этих операций является базисом.

Преобразуем СДНФ функции (1.34) с использованием закона двойного отрицания и закона двойственности:

$$F(v) = \overline{\sum_{i=0}^{2^n-1} a_i \cdot K_i(v)} = \prod_{i=0}^{2^n-1} \overline{a_i \cdot K_i(v)}. \quad (1.40)$$

Данная форма представления функции называется *совершенной нормальной формой (СНФ) в базисе И-НЕ*, т. к. она требует использования только функций И-НЕ.

Преобразуем теперь СКНФ функции (1.38) с использованием закона двойного отрицания и закона двойственности:

$$F(v) = \overline{\prod_{i=0}^{2^n-1} (a_i \vee M_i(v))} = \sum_{i=0}^{2^n-1} \overline{(a_i \vee M_i(v))}. \quad (1.41)$$

Данная форма представления функции называется *совершенной нормальной формой в базисе ИЛИ-НЕ*, т. к. она требует использования только функций ИЛИ-НЕ.



СНФ функции, рассматриваемой в качестве примера, в базисах И-НЕ и ИЛИ-НЕ будут записаны в виде

$$F(v) = \overline{\overline{\overline{x_3 \cdot x_2 \cdot \bar{x}_1 \cdot \bar{x}_3 \cdot x_2 \cdot x_1 \cdot x_3 \cdot \bar{x}_2 \cdot x_1 \cdot x_3 \cdot x_2 \cdot x_1}}}$$

$$F(v) = \overline{\overline{(x_3 \vee x_2 \vee x_1) \vee (x_3 \vee x_2 \vee \bar{x}_1) \vee (\bar{x}_3 \vee x_2 \vee x_1) \vee (\bar{x}_3 \vee \bar{x}_2 \vee x_1)}}$$

Таким образом, термин «дизъюнктивная (конъюнктивная)» форма представления функции подразумевает, что в логическом выражении, задающем функцию, последняя выполняемая операция – дизъюнкция (конъюнкция), термин «нормальная» форма – в логическом выражении, задающем функцию, последовательно выполняется не более двух операций из совокупности операций И, ИЛИ, И-НЕ, ИЛИ-НЕ, «совершенная» – все минтермы (макстермы), входящие в логическое выражение, имеют одинаковую размерность, т. е. содержат одинаковый набор переменных.

## 2 МИНИМИЗАЦИЯ ЛОГИЧЕСКИХ ФУНКЦИЙ

Одной из основных задач, возникающих при синтезе схем цифровых устройств, является минимизация логических функций, которые она реализует. Чем проще логические выражения, описывающие функции, тем проще и дешевле реализующая их цифровая схема.

Существует два основных «ручных» метода минимизации логических функций:

- 1 Аналитический метод с использованием законов и правил алгебры логики.
- 2 Табличный метод, при котором функция представляется в виде карты Карно (диаграммы Вейча).

*Аналитический метод* минимизации базируется на отыскании соседних элементов в записи функции и их объединении на основании законов алгебры логики.

Два минтерма  $K_i(v)$  и  $K_j(v)$  (макстерма  $M_i(v)$  и  $M_j(v)$ ) называются *соседними*, если они различаются только одним первичным термом  $x_p$  (т. е., если один из минтермов содержит  $x_p$ , а другой  $\bar{x}_p$ , все же остальные первичные

термы одинаковые). Так, например, минтермы  $K_2(v) = \bar{x}_3 x_2 \bar{x}_1$  и  $K_6(v) = x_3 x_2 \bar{x}_1$  являются соседними, т. к. они различаются только одним первичным термом  $x_3$ . Для минтерма  $K_2(v)$  соседними будут также минтермы  $K_0(v) = \bar{x}_3 \bar{x}_2 \bar{x}_1$ ,  $K_3(v) = \bar{x}_3 x_2 x_1$ . Очевидно, что каждый минтерм (макстерм)  $n$  переменных  $K_i(v)$  имеет по  $n$  соседних минтермов (макстермов) из общего числа  $2^n$ . На основании свойств дизъюнкции и конъюнкции один и тот же минтерм (макстерм), входящий в СДНФ (СКНФ) может использоваться несколько раз для объединения с соседними минтермами (макстермами).

Пусть задана СКНФ функции трех переменных  $F(v) = K_0(v) \vee K_2(v) \vee K_3(v) \vee K_6(v) = \bar{x}_3 \bar{x}_2 \bar{x}_1 \vee \bar{x}_3 x_2 \bar{x}_1 \vee \bar{x}_3 x_2 x_1 \vee x_3 x_2 \bar{x}_1$ . Здесь для минимизации минтерм  $K_2(v) = \bar{x}_3 x_2 \bar{x}_1$  необходимо использовать три раза:

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

Уже из этого элементарного примера видно, что аналитический метод достаточно сложен в силу трудоемкости поиска соседних минтермов.

*Табличный метод* минимизации логической функции базируется на использовании диаграммы Вейча (карты Карно), которая является одним из способов табличного задания функции и состоит из клеток, каждая из которых соответствует определенной точке  $v_i$  области определения функций, т. е. диаграммы Вейча для функции  $n$  переменных состоят из  $2^n$  клеток, которые можно пронумеровать числами  $i = 0, 1, \dots, 2^n - 1$ . Чтобы с помощью диаграммы Вейча задать функцию  $F(v)$ , необходимо в каждую клетку с номером  $i$  занести значение функции  $F(v_i) = a_i = 0$  или  $1$ , которое она принимает в точке  $i$ . Таким образом, одна строка таблицы истинности функции в диаграмме Вейча представляется одной ячейкой.

Для составления диаграммы Вейча (карты Карно) рисуется таблица, ее строки и столбцы нумеруются в двоичном коде и им присваиваются переменные. Вид диаграммы Вейча и карты Карно для функции четырех переменных представлен на рисунке 2.1. Здесь для наглядности координаты клеток указаны в трех формах: в виде соответствующего минтерма; в виде двоичного числа, соответствующего порядковому номеру наборов аргументов; в десятичном эквиваленте номеров наборов аргументов.

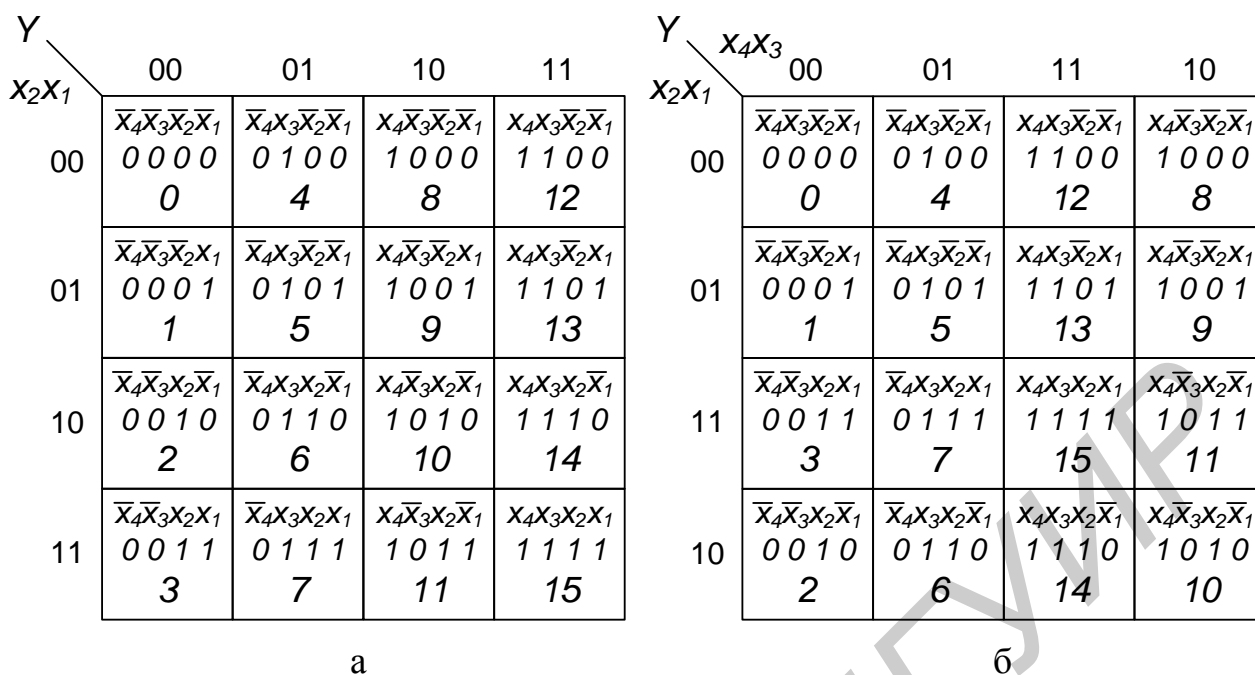


Рисунок 2.1 – Вид диаграммы Вейча (а) и карты Карно для функции четырех аргументов (б)

Следует отметить, что диаграмма Вейча отличается от карты Карно только в способе обозначения строк и столбцов. В диаграмме Вейча строки и столбцы пронумерованы в порядке нарастания двоичных чисел (00, 01, 10, 11), а в карте Карно – последовательностью чисел в коде Грея (00, 01, 11, 10), что делает **все** клетки карты Карно соседними. Это свойство определяет более широкое применение при минимизации функции именно карты Карно.

Для представления логической функции картой Карно необходимо последнюю заполнить единицами и нулями согласно значениям функции, определяемым ее таблицей истинности либо логическим выражением.

Для минимизации функции единицы в карте объединяются замкнутыми контурами в группы по правилам:

1 Число единиц в группе должно быть  $2^k$ , где  $k = 0, 1, 2, 3, \dots$ , следовательно, допустимое число клеток в контуре 1, 2, 4, 8,  $\dots$ .

2 Внутри контура должны быть только клетки, заполненные единицами, одна и та же клетка может входить в несколько контуров.

3 Крайние столбцы и строки карты Карно являются соседними и их клетки могут входить в общий контур.

4 Число контуров (групп) должно быть минимальным, охватывая все единичные клетки.

5 При записи минтермов в минимизированной функции исключаются переменные, имеющие взаимоинверсные значения (0 и 1 либо  $x$  и  $\bar{x}$ ).

В качестве примера минимизации рассмотрим функцию четырех аргументов, заданную таблицей истинности (таблица 2.1)

Таблица 2.1 – Таблица истинности логической функции

$N$	$x_1$	$x_2$	$x_3$	$x_4$	$F(x_1, x_2, x_3, x_4)$
0	0	0	0	0	$F_0(0,0,0,0) = 0$
1	0	0	0	1	$F_1(0,0,0,1) = 1$
2	0	0	1	0	$F_2(0,0,1,0) = 0$
3	0	0	1	1	$F_3(0,0,1,1) = 0$
4	0	1	0	0	$F_4(0,1,0,0) = 1$
5	0	1	0	1	$F_5(0,1,0,1) = 0$
6	0	1	1	0	$F_6(0,1,1,0) = 1$
7	0	1	1	1	$F_7(0,1,1,1) = 0$
8	1	0	0	0	$F_8(1,0,0,0) = 1$
9	1	0	0	1	$F_9(1,0,0,1) = 1$
10	1	0	1	0	$F_{10}(1,0,1,0) = 1$
11	1	0	1	1	$F_{11}(1,0,1,1) = 1$
12	1	1	0	0	$F_{12}(1,1,0,0) = 0$
13	1	1	0	1	$F_{13}(1,1,0,1) = 0$
14	1	1	1	0	$F_{14}(1,1,1,0) = 1$
15	1	1	1	1	$F_{15}(1,1,1,1) = 1$

Карта Карно, соответствующая данной функции, представлена на рисунке 2.2.

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

Рисунок 2.2 – Карта Карно для функции  $F(x_1, x_2, x_3, x_4)$

Объединяем единичные клетки в группы и записываем результат минимизации в минимальной ДНФ:

$$F(x) = x_1 \cdot \overline{x_2} \vee \overline{x_2} \cdot x_4 \vee x_1 \cdot x_3 \vee \overline{x_1} \cdot x_2 \cdot \overline{x_4}.$$

Основная задача минимизации не полностью определенных функций заключается в отыскании оптимального варианта ее доопределения, позволяющего получить минимальную из всех возможных ДНФ или КНФ. Если значения функции не заданы в  $m$  точках, то ее можно доопределить  $2^m$  способами. Поэтому минимизация неполностью определенной функции состоит в оптимальном выборе одной из  $2^m$  полностью определенных функций (понятно, что, как и при минимизации полностью определенных функций, может быть получено несколько равноценных МДНФ и МКНФ). Следует иметь в виду, что в результате минимизации неполностью определенных функций всегда получаются полностью определенные функции.

### 3 ТИПОВЫЕ КОМБИНАЦИОННЫЕ СХЕМЫ

#### 3.1 Базовые логические элементы

Физическое устройство, реализующее одну из основных операций алгебры логики или простейшую переключательную функцию, называется логическим элементом (ЛЭ). Схема, составленная из конечного числа ЛЭ путем соединения выходов одних ЛЭ со входами других по определенным правилам, называется логической схемой (ЛС). В общем случае считаем, что построение ЛС основано на следующих правилах:

- выход ЛЭ можно подсоединять ко входам нескольких ЛЭ;
- на входы ЛЭ можно подавать сигналы, представляющие собой константы 0 и 1;
- выходы ЛЭ нельзя соединять вместе;
- выходы ЛЭ нельзя подключать к собственным входам;
- ЛС может иметь любое число обратных связей, по которым выходные сигналы некоторых ЛЭ возвращаются на собственные входы, предварительно пройдя через некоторое число ЛЭ.

Если ЛС полностью описывается переключательными функциями  $y_i = F_i(x_n, \dots, x_1)$  (одной или несколькими), то она называется комбинационной схемой (КС). Таким образом, логическое состояние выходного сигнала КС однозначно определяется только текущим логическим состоянием входных сигналов. В таких схемах нет обратных связей, по которым сигнал с выхода схемы может пройти на ее вход. При наличии цепей обратной связи схема становится последовательностной.

Проектирование КС осуществляется по этапам:

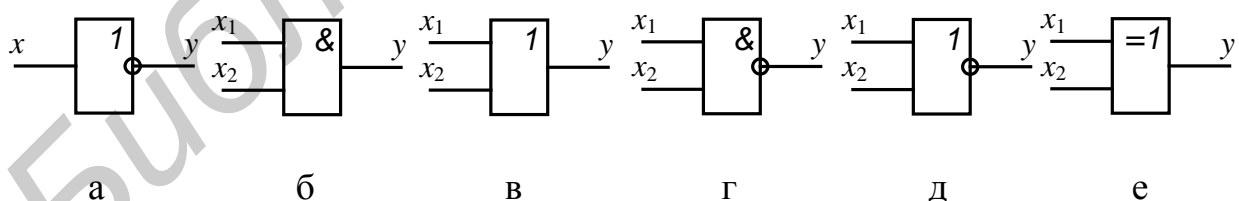
1 На основании словесной формулировки задачи составляется таблица истинности, задающая значения функции на всех наборах аргументов.

2 При использовании аналитического или табличного метода производится минимизация логического выражения.

3 При использовании полученного выражения в качестве структурной формулы, при необходимости, осуществляется переход к заданному логическому базису.

4 Синтезируется схема с использованием базовых логических элементов.

В качестве базовых используются ЛЭ, получившие название исходя из реализуемой логической функции: элемент (схема) НЕ, инвертор; элемент (схема) И, конъюнктор; элемент (схема) ИЛИ, дизъюнктор; элемент (схема) И-НЕ; элемент (схема) ИЛИ-НЕ; элемент (схема) исключающее ИЛИ, сумматор по модулю два. Их условные графические изображения представлены на рисунке 3.1. ЛЭ И, ИЛИ, И-НЕ, ИЛИ-НЕ могут иметь более двух входов, реализуя соответствующие логические функции большего числа переменных.



а – НЕ; б – И; в – ИЛИ; г – И-НЕ; д – ИЛИ-НЕ; е – исключающее ИЛИ

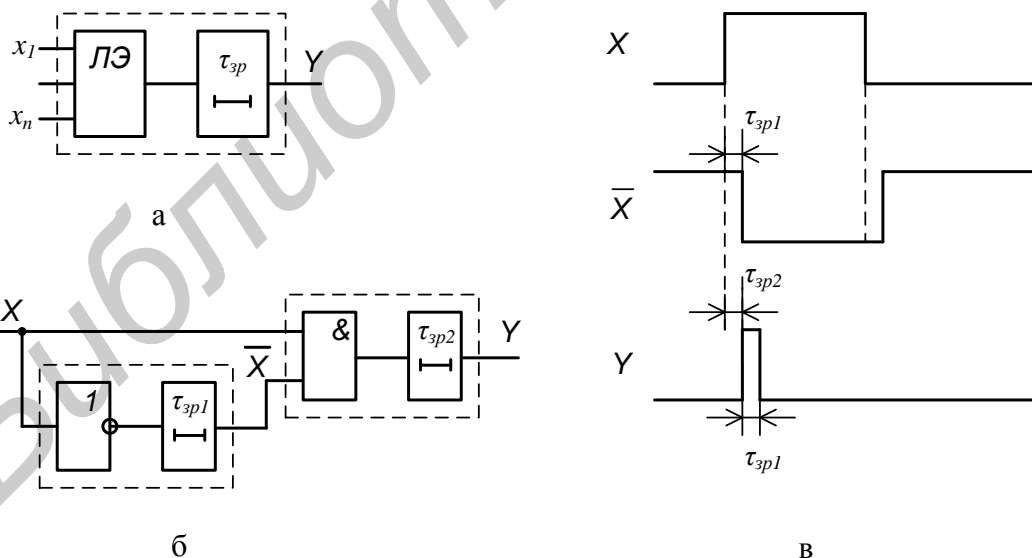
Рисунок 3.1 – Условные графические обозначения базовых ЛЭ

В любом реальном ЛЭ имеется некоторая паразитная задержка между моментом времени, в который на его входы поступают новые значения сигналов, и моментом времени, когда выходной сигнал принимает значение, опреде-

ляемое функцией, которую выполняет ЛЭ. Эта функция представляет собой статическую модель ЛЭ, т. к. она не учитывает поведение ЛЭ при изменении входных сигналов. Аналогично этому логическая функция или система функций, описывающая работу КС без обратных связей, является ее статической моделью.

Для учета переходных процессов, вызываемых в ЛС изменениями входных сигналов, необходимо использовать динамические модели ЛЭ, учитывающие паразитные задержки. Тогда динамическая модель ЛС в целом будет определяться динамической моделью ЛЭ и статической моделью ЛС.

В простейшем случае реальный ЛЭ можно представить в виде последовательного включения идеального ЛЭ, статическая модель которого соответствует выполняемой логической функции и линии задержки на время задержки переключения (рисунок 3.2, а). Учет конечных времен задержек в реальных ЛЭ показывает, что существуют интервалы времени, в пределах которых логическое состояние выхода схемы будет отличаться от тех, что описываются статической моделью. Так, для простейшей схемы (рисунок 3.2, б) статическая модель определяет постоянное нулевое состояние выхода ( $Y = X \cdot \bar{X} = 0$ ), а динамическая – позволяет объяснить появление короткого единичного импульса (рисунок 3.2, в) за счет различных времен задержек в цепях, предшествующих входам ЛЭ.



а – динамическая модель логического элемента;

б – модель логической схемы;

в – временные диаграммы работы

Рисунок 3.2 – Моделирование логической схемы

Максимальное число последовательно выполняемых логических операций для реализации функции  $F(x_n, \dots, x_1)$  называется *порядком переключательной функции*. Функции, представленные в любой нормальной форме, имеют порядок не выше второго. Порядком КС называется максимальное число последовательно включенных ЛЭ. Порядки КС и соответствующих им функций совпадают. Очевидно, чем выше порядок КС, тем меньше ее быстродействие.

Для представления логических функций можно использовать не только нормальные формы. Некоторые тождественные преобразования МНФ могут привести к уменьшению числа первичных термов в аналитическом представлении функции, а значит и к уменьшению стоимости ее реализации на ЛЭ.

Так, преобразование МДНФ с помощью первого дистрибутивного закона (1.9) приводит к скобочным формам представления функций и сокращению числа первичных термов.

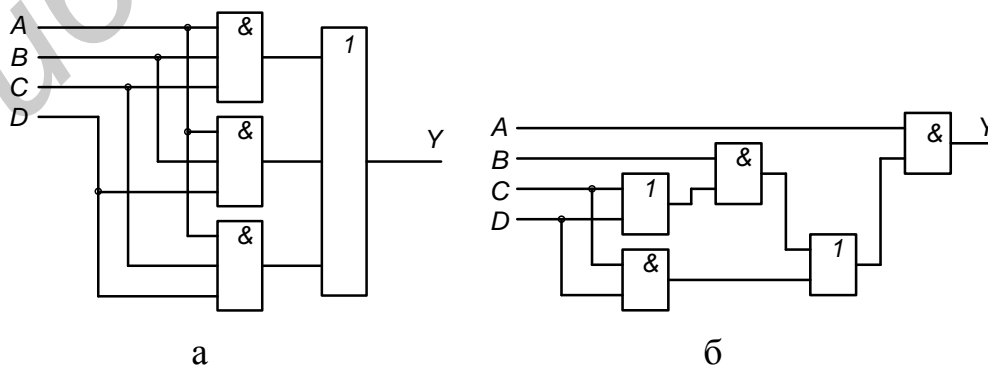
Например, в результате минимизации некоей логической функции получено выражение

$$Y = A \cdot B \cdot C \vee A \cdot B \cdot D \vee A \cdot C \cdot D. \quad (3.1)$$

Этой функции соответствует схема второго порядка, представленная на рисунке 3.3, а. На основании дистрибутивных законов функцию (3.1) можно представить в скобочной форме

$$Y = A \cdot (B \cdot (C \vee D) \vee C \cdot D), \quad (3.2)$$

которой соответствует схема на рисунке 3.3, б. В этой КС максимальное число последовательно включенных ЛЭ равно четырем, т. е. КС имеет четвертый порядок.



а – на основе МДНФ (3.1); б – на основе скобочной формы (3.2)

Рисунок 3.3 – Варианты схемной реализации логической функции



Наряду с базовыми ЛЭ широкое применение находят типовые КС: мультиплексоры, демультимплексоры, шифраторы, дешифраторы, арифметические устройства и др.

### 3.2 Мультиплексоры и демультимплексоры

*Мультиплексоры* (multiplexer) – это функциональный узел, который по заданному адресному двоичному коду осуществляет выбор одного из информационных каналов и подключает его к своему выходу.

Условное графическое обозначение мультиплексора приведено на рисунке 3.4, а.

В общем случае на входы мультиплексора  $D_0 \dots D_{n-1}$  поступают по информационным каналам  $n$ -разрядные сигналы. Число информационных каналов, коммутируемых на выход  $Q$ , составляет  $m = 2, 4, 8, 16$ . При  $m = 4$  и  $Q = 1$  мультиплексор имеет размерность  $4 - 1$ . Это 4-канальный одноразрядный мультиплексор, на выход которого передается один из четырех входных сигналов.

На входы  $A_0 \dots A_{n-1}$  подается  $n$ -разрядный адресный код, при этом каждому информационному каналу присваивается свой адрес (номер), а общее число комбинаций адресных сигналов  $N = 2^n$ .

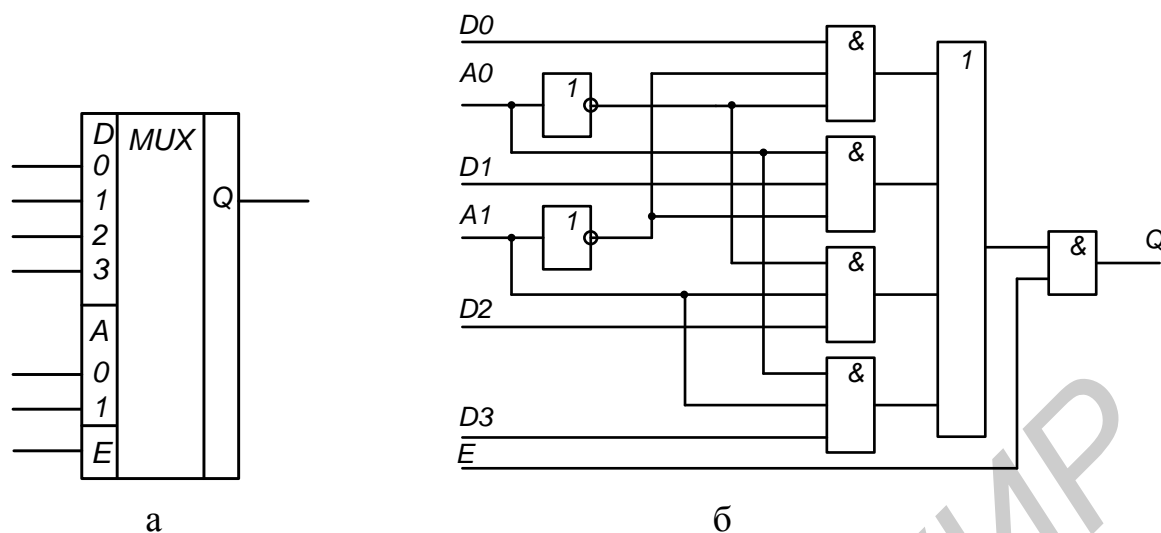
Вход  $E$  (enable) разрешает (либо запрещает) работу мультиплексора при подаче на него сигналов управления.

Таблица 3.1 – Таблица истинности мультиплексора  $4 - 1$

Адресные входы		Вход разрешения	Выход
$A_1$	$A_0$	$E$	$Q$
*	*	0	0
0	0	1	$D_0$
0	1	1	$D_1$
1	0	1	$D_2$
1	1	1	$D_3$

Согласно данным таблицы 3.1 логическая функция  $Q$ , реализуемая мультиплексором  $4 - 1$ , описывается выражением

$$Q = E \cdot (D_0 \cdot \bar{A}_1 \cdot \bar{A}_0 \vee D_1 \cdot \bar{A}_1 \cdot A_0 \vee D_2 \cdot A_1 \cdot \bar{A}_0 \vee D_3 \cdot A_1 \cdot A_0). \quad (3.3)$$



а – условное графическое обозначение; б – логическая схема  
 Рисунок 3.4 – Мультиплексор 4 – 1

Если  $A_0 = 0$ ,  $A_1 = 1$ ,  $E = 1$ , то  $Q = (D_2 \cdot A_1 \cdot \bar{A}_0) \cdot E = D_2$ , следовательно, при  $E = 1$  по адресному коду 10 на выход  $Q$  подключается сигнал, действующий на информационном входе  $D_2$ .

Построенная по выражению (3.3) схема мультиплексора приведена на рисунке 3.4, б.

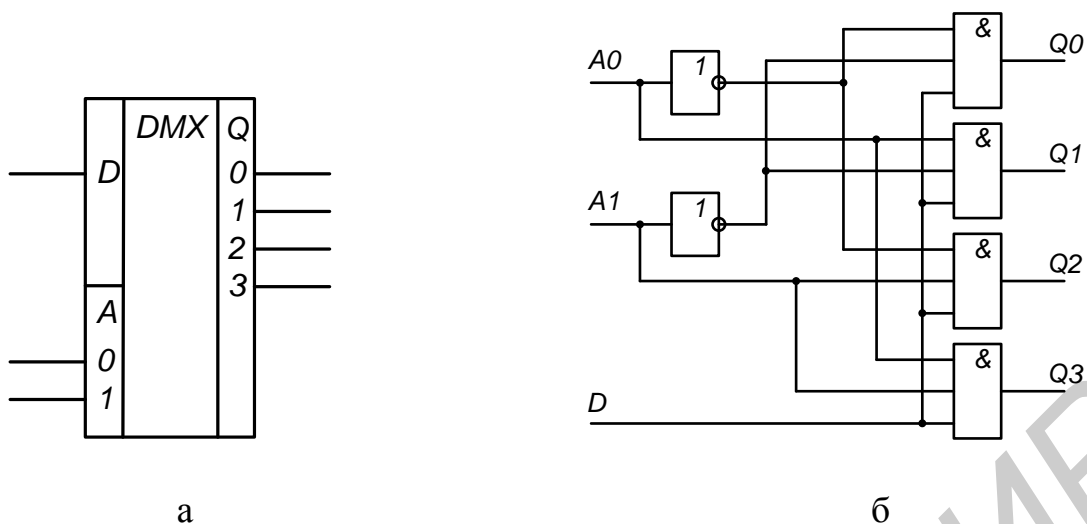
С помощью мультиплексоров можно решать широкий круг задач, используя их как универсальные логические модули.

*Демультимплексор* осуществляет передачу сигналов с одного информационного входа  $D$  на  $N$  выходов, имеющих заданный адрес. В общем случае число выходов  $N = 2^n$  и определяется количеством адресных входов  $n$ .

На рисунке 3.5, а показано условное обозначение демультимплексора 1–4, функционирование которого описывается таблицей 3.2 истинности.

Таблица 3.2 – Таблица истинности мультиплексора 4 – 1

Адресные входы		Выходы			
$A_1$	$A_0$	$Q_0$	$Q_1$	$Q_2$	$Q_3$
0	0	$D$	0	0	0
0	1	0	$D$	0	0
1	0	0	0	$D$	0
1	1	0	0	0	$D$



а – условное графическое обозначение; б – логическая схема

Рисунок 3.5 – Демультимплексор 1 – 4

Согласно данным таблицы истинности запишем логическое выражение для каждого выхода демультимплексора и по ним составим логическую схему (рисунок 3.5, б).

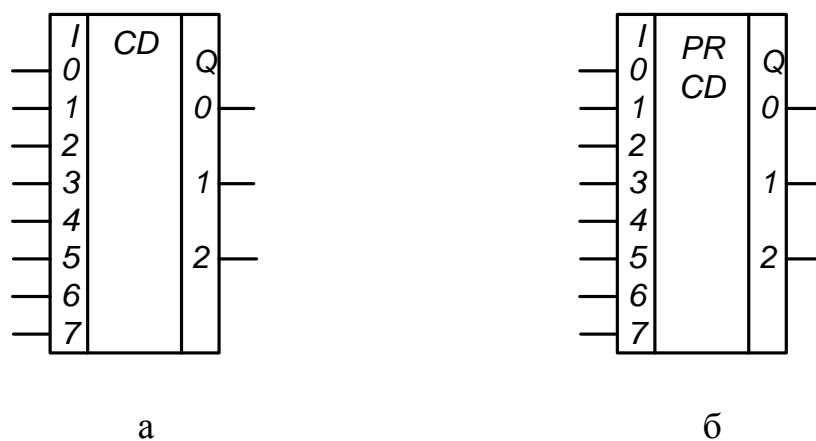
$$\begin{aligned}
 Q_0 &= \overline{A_0} \cdot \overline{A_1} \cdot D, \\
 Q_1 &= A_0 \cdot \overline{A_1} \cdot D, \\
 Q_2 &= \overline{A_0} \cdot A_1 \cdot D, \\
 Q_3 &= A_0 \cdot A_1 \cdot D.
 \end{aligned}
 \tag{3.4}$$

### 3.3 Шифраторы и дешифраторы

*Шифратор* (coder) предназначен для преобразования входных сигналов в виде кода «1 из  $m$ » (унитарного) в выходной  $n$ -разрядный двоичный код и применяется в устройстве ввода информации в цифровые системы. Таким образом, классический шифратор преобразует номер возбуждаемого входа (на который подается логическая 1) в двоичный код на своем выходе.

Шифратор называется полным, если число входов  $m = 2^n$ , а число выходов равно  $n$  ( $n$  – разрядность двоичного кода).

На рисунке 3.6, а показано условное графическое обозначение полного шифратора 8 – 3, а его таблица истинности приведена в таблице 3.3.



а – полный; б – приоритетный

Рисунок 3.6 – Условные графические обозначения шифраторов 8 – 3

Таблица 3.3 – Таблица истинности полного шифратора 8 – 3

Номер входа	Входной код								Выходной код		
	$I_7$	$I_6$	$I_5$	$I_4$	$I_3$	$I_2$	$I_1$	$I_0$	$Q_2$	$Q_1$	$Q_0$
0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0	0	1
2	0	0	0	0	0	1	0	0	0	1	0
3	0	0	0	0	1	0	0	0	0	1	1
4	0	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	0	0	0	0	1	0	1
6	0	1	0	0	0	0	0	0	1	1	0
7	1	0	0	0	0	0	0	0	1	1	1

Согласно данным таблицы 3.3 значения выходных переменных шифратора определяются следующими логическими выражениями:

$$\begin{aligned}
 Q_2 &= I_4 \vee I_5 \vee I_6 \vee I_7; \\
 Q_1 &= I_2 \vee I_3 \vee I_6 \vee I_7; \\
 Q_0 &= I_1 \vee I_3 \vee I_5 \vee I_7.
 \end{aligned}
 \tag{3.5}$$

Необходимо отметить, что шифратор, реализованный по выражениям (3.5), будет правильно работать только в случае подачи на его входы унитарного кода, т. е. должно выполняться условие  $I_j \cdot I_k = 0$  при  $j \neq k$ . Для обнаружения нулевого состояния всех входов шифратора можно предусмотреть в его схеме

дополнительный выход  $GS$ . Очевидно, что логическая функция, реализуемая по этому выходу, описывается выражением

$$GS = \overline{I_0 \vee I_1 \vee I_2 \vee I_3 \vee I_4 \vee I_5 \vee I_6 \vee I_7}. \quad (3.6)$$

Для обработки входного кода, в котором присутствует более одной единицы, т. е. при возбуждении нескольких входов устройства, применяют *приоритетный шифратор*. В нем выходной код соответствует номеру того возбуждаемого входа, приоритет которого задан более высоким. На схемах приоритетный шифратор обозначается буквами  $PRCD$  (рисунок 3.6, б).

Таблица истинности шифратора, приоритет входов которого возрастает с увеличением номера входа, представлена таблицей 3.4.

Таблица 3.4 – Таблица истинности приоритетного шифратора 8 – 3

Номер входа	Входной код								Выходной код		
	$I_7$	$I_6$	$I_5$	$I_4$	$I_3$	$I_2$	$I_1$	$I_0$	$Q_2$	$Q_1$	$Q_0$
0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	*	0	0	1
2	0	0	0	0	0	1	*	*	0	1	0
3	0	0	0	0	1	*	*	*	0	1	1
4	0	0	0	1	*	*	*	*	1	0	0
5	0	0	1	*	*	*	*	*	1	0	1
6	0	1	*	*	*	*	*	*	1	1	0
7	1	*	*	*	*	*	*	*	1	1	1

Состояния выходов такого приоритетного шифратора определяются следующими логическими функциями:

$$\begin{aligned}
 Q_2 &= I_4 \cdot \overline{I_5} \cdot \overline{I_6} \cdot \overline{I_7} \vee I_5 \cdot \overline{I_6} \cdot \overline{I_7} \vee I_6 \cdot \overline{I_7} \vee I_7 = I_4 \vee I_5 \vee I_6 \vee I_7; \\
 Q_1 &= I_2 \cdot \overline{I_3} \cdot \overline{I_4} \cdot \overline{I_5} \cdot \overline{I_6} \cdot \overline{I_7} \vee I_3 \cdot \overline{I_4} \cdot \overline{I_5} \cdot \overline{I_6} \cdot \overline{I_7} \vee I_6 \cdot \overline{I_7} \vee I_7 = \\
 &= I_2 \cdot \overline{I_4} \cdot \overline{I_5} \vee I_3 \cdot \overline{I_4} \cdot \overline{I_5} \vee I_6 \vee I_7; \\
 Q_0 &= I_1 \cdot \overline{I_2} \cdot \overline{I_3} \cdot \overline{I_4} \cdot \overline{I_5} \cdot \overline{I_6} \cdot \overline{I_7} \vee I_3 \cdot \overline{I_4} \cdot \overline{I_5} \cdot \overline{I_6} \cdot \overline{I_7} \vee I_5 \cdot \overline{I_6} \cdot \overline{I_7} \vee I_7 = \\
 &= I_1 \cdot \overline{I_2} \cdot \overline{I_4} \cdot \overline{I_6} \vee I_3 \cdot \overline{I_4} \cdot \overline{I_6} \vee I_5 \cdot \overline{I_6} \vee I_7.
 \end{aligned} \quad (3.7)$$

Дешифратор (decoder) выполняет функцию, обратную шифратору, т. е. преобразует входной  $n$ -разрядный двоичный код в унитарный («1 из  $m$ »). Таким образом, сигнал логической единицы появится только на том выходе дешифратора, номер которого соответствует десятичному эквиваленту входного двоичного кода. Дешифратор называется полным, если при  $n$  входах имеется  $m = 2^n$  выходов. На схемах дешифраторы обозначаются буквами  $DC$  (рисунок 3.7).

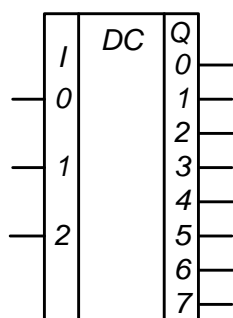


Рисунок 3.7 – Условное графическое обозначение дешифратора 3 – 8

Рассмотрим пример синтеза полного дешифратора 3 – 8, имеющего на входе 3-разрядный двоичный код, а на восьми выходах – унитарный код.

Таблица 3.5 – Таблица истинности полного дешифратора 3 – 8

$I_2$	$I_1$	$I_0$	$Q_7$	$Q_6$	$Q_5$	$Q_4$	$Q_3$	$Q_2$	$Q_1$	$Q_0$
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Анализ таблицы истинности (таблица 3.5) такого дешифратора показывает, где каждой выходной функции  $Q_0 \dots Q_7$  соответствует только один минтерм. Так как рассматриваемый дешифратор полный, то он реализует полный набор минтермов входных переменных  $I_0, I_1, I_2$ :

$$Q_0 = \overline{I_0} \cdot \overline{I_1} \cdot \overline{I_2};$$

$$Q_1 = I_0 \cdot \overline{I_1} \cdot \overline{I_2};$$

$$\begin{aligned}
 Q_2 &= \overline{I_0} \cdot I_1 \cdot \overline{I_2}; \\
 Q_3 &= I_0 \cdot I_1 \cdot \overline{I_2}; \\
 Q_4 &= \overline{I_0} \cdot \overline{I_1} \cdot I_2; \\
 Q_5 &= I_0 \cdot \overline{I_1} \cdot I_2; \\
 Q_6 &= \overline{I_0} \cdot I_1 \cdot I_2; \\
 Q_7 &= I_0 \cdot I_1 \cdot I_2.
 \end{aligned}
 \tag{3.8}$$

Если комбинационная схема реализует только один минтерм, то ее часто называют дешифратором состояния (кода).

### 3.4 Сумматоры и вычитатели

*Сумматор* (sum) – цифровой функциональный узел, выполняющий арифметическое сложение двоичных чисел.

Простейшим сумматором является полусумматор *HS*, осуществляющий арифметическое сложение двух одноразрядных чисел *A* и *B* (рисунок 3.8, а). Он имеет два выхода – выход *S* бита суммы и выход *P* бита переноса в старший разряд. Логика работы полусумматора описывается таблицей 3.6.



а – полусумматор; б – одноразрядный полный сумматор

Рисунок 3.8 – Условные графические обозначения

Таблица 3.6 – Таблица истинности полусумматора

<i>A</i>	<i>B</i>	<i>S</i>	<i>P</i>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Из таблицы истинности следует, что арифметическая сумма *S* двух одноразрядных чисел *A* и *B* совпадает с суммированием их по модулю два:

$$S = \overline{A}B + A\overline{B} = A \oplus B. \quad (3.9)$$

Сигнал переноса в старший разряд

$$P = A \cdot B \quad (3.10)$$

появляется на выходе только тогда, когда на входы  $A$  и  $B$  поданы единицы, реализуется логическим элементом И.

Одноразрядный полный сумматор  $SM$  (рисунок 3.8, б) отличается от полусумматора  $HS$  тем, что на его дополнительный вход  $C$  поступает сигнал переноса от предыдущего разряда сумматора (таблица 3.7).

Таблица 3.7 – Таблица истинности полного сумматора и вычитателя

Входные сигналы			Выходные сигналы сумматора		Выходные сигналы вычитателя	
$A$	$B$	$C$	$S$	$P$	$D$	$Z$
0	0	0	0	0	0	0
0	1	0	1	0	1	1
1	0	0	1	0	1	0
1	1	0	0	1	0	0
0	0	1	1	0	1	1
0	1	1	0	1	0	1
1	0	1	0	1	0	0
1	1	1	1	1	1	1

Для определения бита суммы  $S$  и бита переноса  $P$  в старший разряд по таблице истинности составляем логические выражения в СДНФ и преобразуем их:

$$\begin{aligned} S &= \overline{A} \cdot B \cdot \overline{C} + A \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C + A \cdot B \cdot C = \\ &= \overline{C}(\overline{A} \cdot B + A \cdot \overline{B}) + C(\overline{A} \cdot \overline{B} + A \cdot B) = \\ &= \overline{C}(A \oplus B) + C(\overline{A \oplus B}) = A \oplus B \oplus C, \end{aligned} \quad (3.11)$$

$$P = A \cdot B \cdot \overline{C} + \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot C + A \cdot B \cdot C = A \cdot B + B \cdot C + A \cdot C. \quad (3.12)$$

По выражениям (3.11) и (3.12) составляем логическую схему двоичного одноразрядного сумматора  $SM$  (рисунок 3.9).



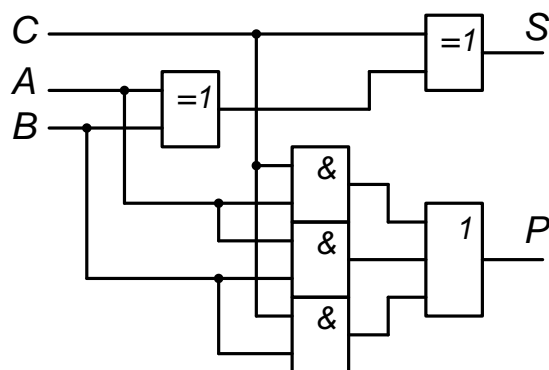


Рисунок 3.9 – Логическая схема одноразрядного полного сумматора

Рассмотрим реализацию универсального блока, который может вычислять не только сумму, но и разность двух одноразрядных чисел  $A$  и  $B$ , т. е. будет работать либо как сумматор  $SM$ , либо как *вычитатель*  $SUB$  (*subtraktor*).

Управление режимом работы блока осуществляется сигналом  $M$  (mode): при  $M = 0$  схема работает как сумматор, т. е. выполняет арифметическую операцию  $A + B + C$ , где  $C$  – бит переноса из младшего разряда, с формированием бита  $S$  суммы и бита  $P$  переноса в старший разряд. При  $M = 1$  схема работает как вычитатель, выполняя операцию  $A - B - C$  (при этом  $C$  трактуется как бит заема из текущего разряда), с формированием на выходе бита  $D$  разности и бита  $Z$  заема из старшего разряда (см. таблицу 3.7).

На основании совпадения данных столбцов  $S$  и  $D$  таблицы 3.7 истинности делаем вывод, что логическое выражение для определения бита  $D$  разности будет совпадать с (3.11), т. е.

$$D = A \oplus B \oplus C, \quad (3.13)$$

и, следовательно, сигналы  $S$  и  $D$  могут быть сформированы одной схемой.

Формирование бита  $Z$  заема осуществляется по выражению

$$Z = \bar{A} \cdot B \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C + A \cdot B \cdot C = \bar{A} \cdot B + B \cdot C + \bar{A} \cdot C, \quad (3.14)$$

которое отличается от (3.12) заменой переменной  $A$  на инверсную  $\bar{A}$ .

Схемная реализация такой замены переменной может быть осуществлена на базе элемента Иключающее ИЛИ (сложения по модулю 2), входными сигналами которой выступают  $A$  и  $M$ . На основании (1.19) можно сделать вывод,

что при  $M = 0$  (режим сумматора) схема сложения по модулю 2 выступает в качестве повторителя сигнала  $A$ , а при  $M = 1$  (режим вычитателя) – его инвертора.

С учетом этого по выражениям (3.11)...(3.14) составляем логическую схему одноразрядного универсального сумматора-вычитателя (рисунок 3.10).

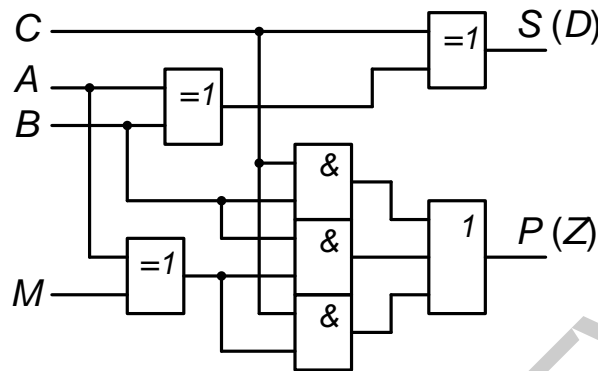


Рисунок 3.10 – Логическая схема сумматора-вычитателя

В многоразрядных сумматорах увеличение разрядности достигается путем каскадирования, при этом выход  $P$  переноса младшего разряда соединяют с входом  $C$  переноса старшего разряда. Однако последовательное формирование бита переноса снижает быстродействие многоразрядных сумматоров. Для его повышения применяют параллельное формирование переносов в специальных блоках ускоренного переноса.

### 3.5 Цифровые компараторы

*Компаратор* (comparator) предназначен для сравнения двух двоичных чисел. С помощью компаратора можно производить проверку равенства чисел или определить какое из них больше. На схемах компаратор обозначается буквами  $СМР$  либо (= =).

Цифровые компараторы подразделяются на одноразрядные и многоразрядные. Простейшим одноразрядным компаратором является схема Исключающее ИЛИ-НЕ, с помощью которой реализуется функция равнозначности.

На рисунке 3.11 показана реализация четырехразрядного компаратора, выполненного на логических элементах Исключающее ИЛИ и ИЛИ-НЕ.

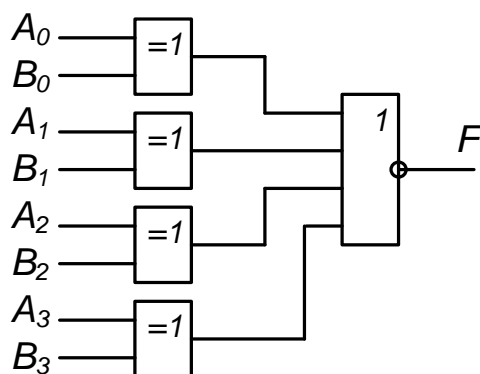


Рисунок 3.11 – Логическая схема четырехразрядного цифрового компаратора

При равенстве входных сигналов во всех разрядах выходной сигнал  $F = 1$ , но если хотя бы в одном разряде сигналы различны, то выходной сигнал  $F = 0$ .

Составим таблицу истинности для одноразрядного универсального компаратора, который кроме констатации равенства двух чисел может определить, какое из них больше.

Таблица 3.8 – Таблица истинности универсального одноразрядного цифрового компаратора

$A$	$B$	$F_{A>B}$	$F_{A=B}$	$F_{A<B}$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

Согласно таблице 3.8 запишем логические функции, характеризующие признаки равенства или неравенства между входными числами  $A$  и  $B$ :

$$F_{A>B} = A\bar{B}; \quad (3.15)$$

$$F_{A<B} = \bar{A}B; \quad (3.16)$$

$$F_{A=B} = \bar{A} \cdot \bar{B} \vee A \cdot B = \overline{A \oplus B} = \overline{\bar{A} \cdot B \vee A \cdot \bar{B}} = \overline{F_{A>B} \vee F_{A<B}}. \quad (3.17)$$

По полученным выражениям (3.15)...(3.17) построим схему одноразрядного универсального компаратора (рисунок 3.12).

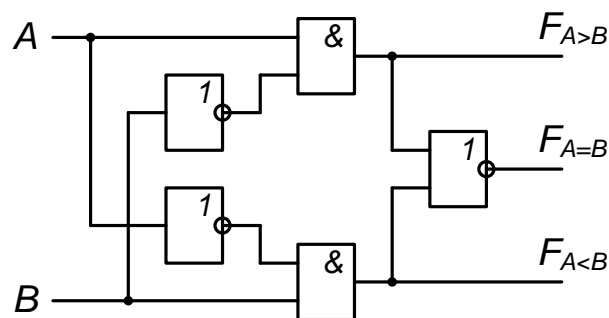


Рисунок 3.12 – Схема универсального одноразрядного компаратора

С целью повышения разрядности компараторы соединяют последовательно, как показано на рисунке 3.13 на примере двух четырехразрядных *СМР*.

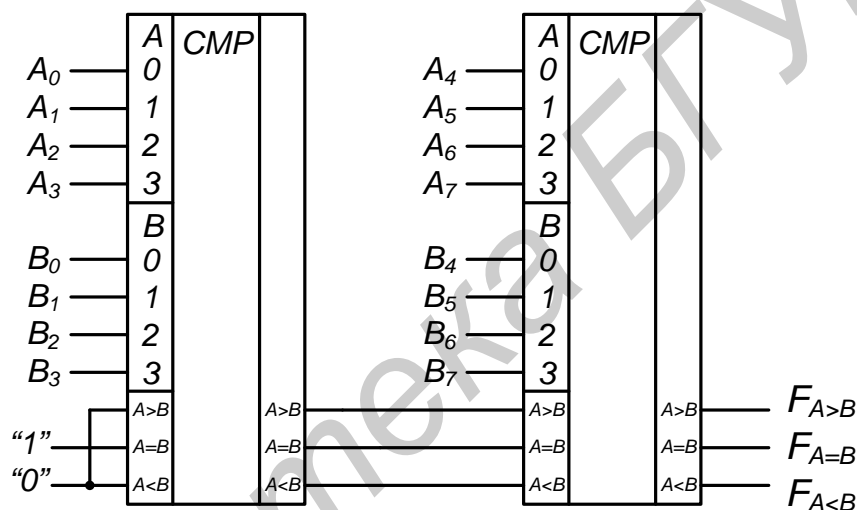


Рисунок 3.13 – Вариант наращивания разрядности компараторов

### 3.6 Преобразователи кодов

*Преобразователи кодов* (converter) предназначены для преобразования кода кодирования цифровой информации. На схемах преобразователи обозначаются буквами  $X/Y$ .

Преобразователи кодов отличаются достаточным многообразием, в частности, к ним можно отнести схемы шифратора и дешифратора, выполняющие взаимное преобразование двоичного и унитарного кодов.

Рассмотрим синтез преобразователя двоично-десятичного кода в специальный семиразрядный код индикатора. Подсветка всех сегментов цифрового

индикатора, образующих цифры 0...9 (рисунок 3.14), производится подачей на них единичного сигнала, вызывая свечение сегментов в определенных комбинациях.

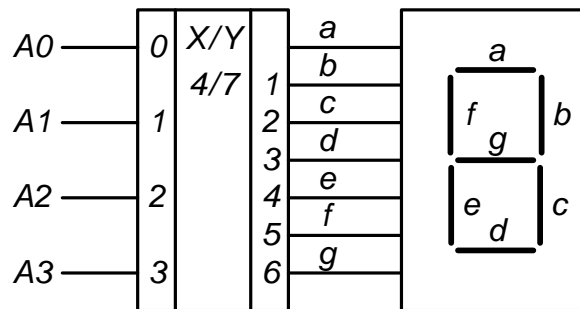


Рисунок 3.14 – Схема управления семисегментным индикатором

Из рисунка 3.15 следует, что для цифры индикатора, например 6, которая задана двоично-десятичным кодом 0110, необходимо подсветить все сегменты, за исключением сегмента *b*.

Выходные переменные *a*, ..., *g* являются функциями аргументов *A3*, ..., *A0*. Определяем для функций  $F_a \dots F_g$  таблицу истинности (таблица 3.9), по которой отдельно для каждой функции составляем карты Карно, при этом «лишние» ячейки карт, где функции не определены, помечаем знаком \* и доопределяем единицами.

Таблица 3.9 – Таблица истинности преобразователя кодов

Цифра индикатора	Входы				Выходы						
	$A_3$	$A_2$	$A_1$	$A_0$	$F_a$	$F_b$	$F_c$	$F_d$	$F_e$	$F_f$	$F_g$
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

Минимизировав логические функции  $F_a \dots F_g$ , получаем выражения для них в минимальной дизъюнктивной нормальной форме:

$$\begin{aligned}
F_a &= A_3 \cdot A_0 \vee \overline{A_3} \cdot \overline{A_0} \vee A_3 \vee A_1; \\
F_b &= \overline{A_1} \cdot \overline{A_0} \vee A_1 \cdot A_0 \vee \overline{A_2}; \\
F_c &= A_2 \vee \overline{A_1} \vee A_0; \\
F_d &= A_2 \cdot \overline{A_1} \cdot A_0 \vee A_3 \vee \overline{A_2} \cdot \overline{A_0} \vee \overline{A_2} \cdot A_1 \vee A_1 \cdot \overline{A_0}; \\
F_e &= \overline{A_2} \cdot \overline{A_0} \vee A_1 \cdot \overline{A_0}; \\
F_f &= A_2 \cdot \overline{A_1} \vee \overline{A_1} \cdot \overline{A_0} \vee A_2 \cdot \overline{A_0} \vee A_3; \\
F_g &= A_2 \cdot \overline{A_1} \vee A_1 \cdot \overline{A_0} \vee \overline{A_2} \cdot A_1 \vee A_3.
\end{aligned}
\tag{3.18}$$

## 4 ТИПОВЫЕ ПОСЛЕДОВАТЕЛЬНОСТНЫЕ СХЕМЫ

### 4.1 Потенциальные и импульсные сигналы

В последовательностных схемах, в отличие от комбинационных, значения выходных сигналов зависят не только от значений входных сигналов, но и от последовательности их изменений. В таких схемах (последовательных автоматах) зачастую необходимо введение в рассмотрение в явном виде времени для описания изменений сигналов. Аналитически это осуществляется путем использования операторов переходов  $d$  и  $\nabla$ .

Сигнал называется *потенциальным*, если интервал времени  $T_i$  между соседними изменениями сигнала значительно больше времени установления переходных процессов в схеме, в которой он используется. Сигнал называется *импульсным*, если длительность его активного уровня того же порядка, что и время реакции схемы (схема должна отреагировать на воздействие импульсного сигнала, а он должен закончиться сразу же после окончания в схеме переходного процесса).

При аналитическом описании схем, на которые воздействуют импульсные сигналы, используется понятие абстрактного импульсного сигнала, длительность которого бесконечно мала. Такие сигналы ( $dx$ ,  $d\bar{x}$  и  $\nabla x$ ) показаны на рисунке 4.1 и соответствуют изменениям потенциального сигнала  $x$  с 0 на 1 или (и) с 1 на 0. Реальные импульсные сигналы всегда имеют конечную длительность, которая определяется временем реакции схемы. В зависимости от быст-

родействия ЛЭ, из которых построена схема, время реакции может изменяться в широких пределах. Понятие абстрактного импульсного сигнала позволяет абстрагироваться от физических параметров конкретных схем.

Оператор переходов  $d$  определяется соотношением

$$dx = x(t) \cdot \overline{x(t - \Delta t)}, \quad (4.1)$$

где  $dx$  – импульсный сигнал, принимающий единичное значение при изменении потенциального сигнала с 0 на 1;

$x(t)$  – значение потенциального сигнала в текущий момент времени;

$x(t - \Delta t)$  – значение потенциального сигнала в предыдущий момент времени.

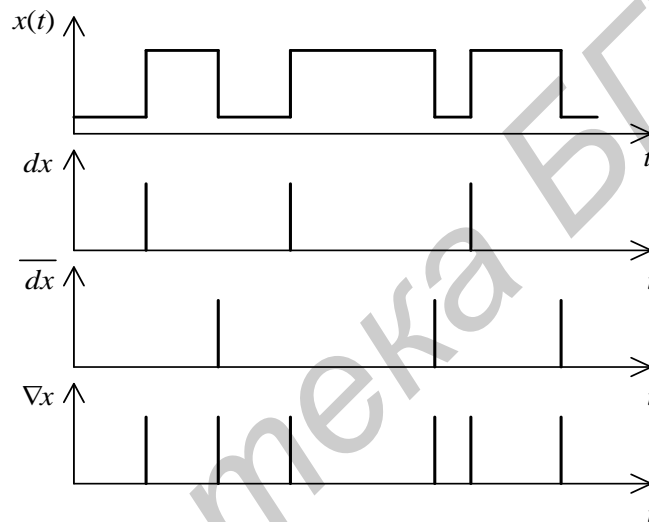


Рисунок 4.1 – Разновидности импульсных сигналов

Импульсный сигнал, соответствующий изменениям потенциального сигнала с 1 на 0:

$$d\bar{x} = \overline{x(t) \cdot x(t - \Delta t)}. \quad (4.2)$$

Оператор переходов  $\nabla$  определяется соотношением

$$\nabla x = dx \vee d\bar{x} = x(t) \oplus x(t - \Delta t), \quad (4.3)$$

где  $\nabla x = 1$  как при изменении потенциального сигнала с 0 на 1, так и с 1 на 0.

На рисунке 4.2 показана схема формирования импульсного сигнала  $dx$  и временные диаграммы ее работы в предположении, что используемые ЛЭ безынерционны.

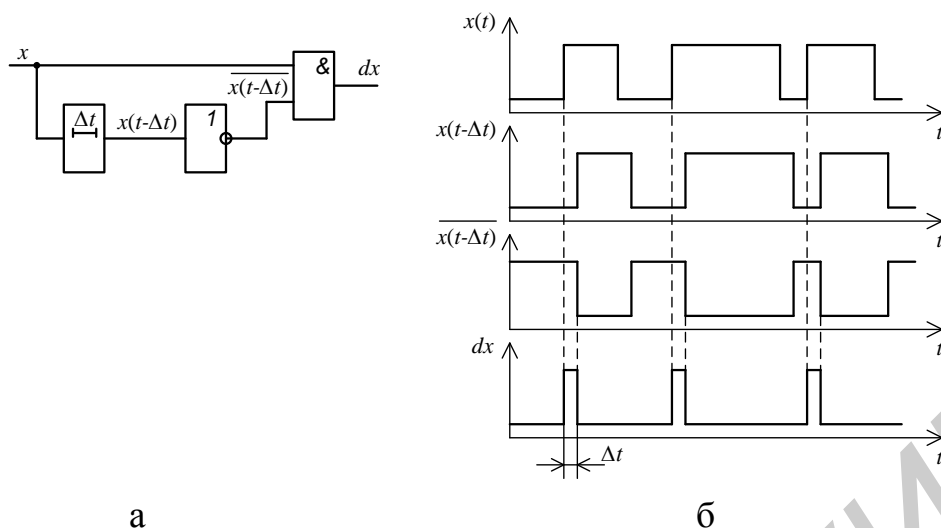


Рисунок 4.2 – Схема формирования импульсного сигнала  $dx$  (а) и временные диаграммы ее работы (б)

#### 4.2 Модель последовательного автомата

Эта модель состоит из комбинационной схемы (КС) и элементов задержки входных сигналов  $Q_r^+$  на время  $\Delta t$ , включенных в цепях обратных связей (рисунок 4.3). Элементы задержки производят запоминание внутренних сигналов КС, т. е. они являются элементами памяти (ЭП). Эти сигналы появляются на входах КС через время  $\Delta t$  и могут вызвать изменение ее выходных сигналов. Понятно, что если сигнал  $Q_r = Q_r(t)$ , то сигнал  $Q_r^+ = Q_r(t + \Delta t)$ .

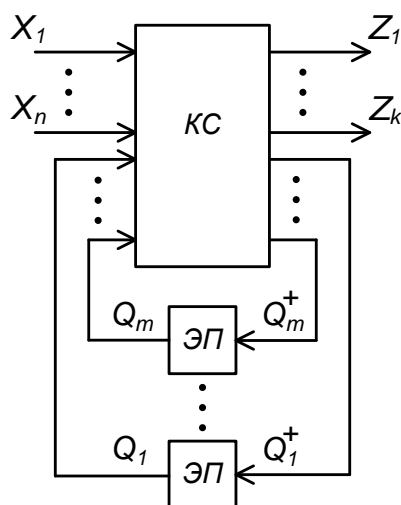


Рисунок 4.3 – Модель последовательного автомата



В общем случае автомат содержит  $m$  ЭП входных сигналов  $Q_m, \dots, Q_r, \dots, Q_1$ , имеет  $n$  физических входов, на которые подаются сигналы  $x_n, \dots, x_p, \dots, x_1$ , и  $k$  физических выходов, с которых снимаются сигналы  $z_k, \dots, z_q, \dots, z_1$ .

Совокупность входных сигналов автомата  $\nu = (x_n, \dots, x_1)$  называется *состоянием входа* автомата, совокупность выходных сигналов автомата  $\lambda = (z_k, \dots, z_1)$  – *состоянием выхода* автомата, а совокупность выходных сигналов  $\mu = (Q_m, \dots, Q_1)$  элементов памяти – *внутренним состоянием* автомата.

При фиксированных значениях внутренних сигналов  $Q_r$  автомат ведет себя подобно некоторой КС, т. е. реализует однозначное соответствие между значениями входных и выходных сигналов. Однако при изменении входных сигналов его реакция может выразиться в изменении внутренних сигналов. Если затем подать прежние значения входных сигналов, то соответствие между значениями входных и выходных сигналов может оказаться совсем другим.

Последовательностная схема полностью описывается двумя функциями:  
- *функцией переходов* автомата

$$Q_r^+ = F_r(\nu, \mu), \quad (4.4)$$

показывающей зависимость состояния внутреннего сигнала  $Q_r$  в последующий момент времени от состояния входа и внутреннего состояния автомата в текущий момент времени;

- *функцией выхода* автомата

$$z_q = G_q(\nu, \mu), \quad (4.5)$$

показывающей зависимость состояния выходного сигнала  $z_q$  от состояния входа и внутреннего состояния автомата.

Если для каждой упорядоченной пары внутренних состояний  $(\mu_j, \mu_k)$  имеется хотя бы одно состояние входа  $\nu_i$ , которое переводит автомат из состояния  $\mu_j$  в состояние  $\mu_k$ , то говорят, что автомат обладает полной системой переходов. Данное условие должно выполняться как при  $j \neq k$ , так и при  $j = k$ .

Цифровые автоматы (последовательностные схемы) по способу воздействия на них входных информационных сигналов подразделяются на три основных класса: асинхронные потенциальные автоматы, синхронные автоматы и асинхронные импульсные автоматы. Каждый из классов автоматов можно разделить на несколько типов по другим признакам.

На асинхронный потенциальный автомат воздействия производят непосредственно его входные информационные сигналы до тех пор, пока сохраняется их активный уровень. В синхронном автомате используется единый для всего автомата тактовый сигнал, который осуществляет на него импульсное воздействие в соответствии с уровнями входных информационных сигналов, т. е. в синхронном автомате последние только управляют воздействиями, а момент времени воздействия полностью определяется тактовым сигналом. Асинхронный импульсный автомат отличается от потенциального тем, что входные потенциальные сигналы производят на него импульсные воздействия в момент их изменения. Такие воздействия в синхронных автоматах производит только один сигнал – тактовый, а в асинхронных импульсных автоматах любой из входных сигналов может управлять воздействиями и производить их. Классы автоматов различаются только типом используемых в них ЭП: с потенциальным воздействием входных сигналов, импульсным воздействием только одного тактового сигнала или с импульсным воздействием нескольких входных сигналов (может быть и одного).

Для проектирования цифровых автоматов используются не только ЛЭ, но и элементы памяти, в качестве которых чаще всего используются триггеры. Элементы памяти и триггеры являются элементарными автоматами, поэтому они, как и цифровые автоматы, делятся на те же три класса.

### 4.3 Триггеры

Триггер – это цифровой автомат, имеющий не более двух устойчивых состояний выхода для каждого допустимого состояния входа и обладающий полной системой переходов, внутренние состояния которого отождествляются со значениями выходного сигнала  $Q$ . Таким образом, состояние триггера полностью характеризуется значением его выходного сигнала  $Q$ .

Так как любой автомат задается функциями переходов (4.4) и выхода (4.5), то из данного выше определения следует, что для триггеров  $r = 1$ ,  $q = 1$ ,  $z = Q$ . Ввиду этого закон функционирования триггеров полностью описывается одной только функцией переходов  $Q^+ = F(x_n, \dots, x_1, Q) = F(v, Q)$ .

Триггеры обычно имеют два выхода: прямой (основной)  $Q$  и инверсный  $\bar{Q}$ . Состояние триггера определяется значением логического уровня на

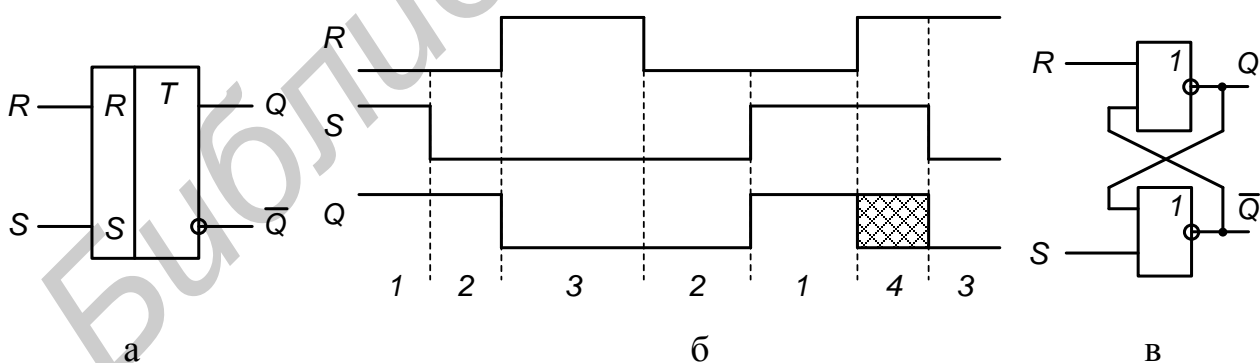
прямом выходе. Если, например, на прямом выходе имеется уровень, соответствующий логической «1», то триггер находится в единичном состоянии (при этом потенциал на инверсном выходе соответствует логическому «0»). В противном случае триггер находится в нулевом состоянии. Число входов схемы управления зависит от структуры и функций, выполняемых триггером.

Триггеры подразделяются:

- по способу управления: на асинхронные и синхронные;
- по виду входных сигналов: на потенциальные и импульсные;
- по функциональным возможностям: *RS*-, *D*-, *JK*-, *T*- триггеры и комбинированные;
- по внутренней структуре: на одноступенчатые (однотактные) и двухступенчатые (двухтактные) – типа *M-S*.

### 4.3.1 *RS*-триггеры

*Асинхронный* потенциальный *RS*-триггер – это триггер с отдельной установкой состояний логического «0» и «1» (с отдельным запуском). Он имеет два информационных входа *S* и *R* (рисунок 4.4, а). По входу *S* (*Set* – установка) триггер устанавливается в состояние  $Q = 1$ , а по входу *R* (*Reset* – сброс) – в состояние  $Q = 0$ . Функционирование *RS*-триггера (рисунок 4.4, б) может быть описано в виде таблицы переключений, представляемой в полном (таблица 4.1) или сокращенном виде (таблица 4.2).



а – условное графическое обозначение;

б – временные диаграммы работы (1 – режим установки «1»,

2 – хранение, 3 – установка «0», 4 – неопределенность);

в – схемная реализация на элементах ИЛИ-НЕ

Рисунок 4.4 – Асинхронный *RS*-триггер

Таблица 4.1 – Полная таблица переходов асинхронного  $RS$ -триггера

$R$	$S$	$Q$	$Q^+$	Режим
0	0	0	0	Хранение
0	0	1	1	
0	1	0	1	Установка 1
0	1	1	1	
1	0	0	0	Установка 0
1	0	1	0	
1	1	0	*	Запрещенный
1	1	1	*	

Таблица 4.2 – Таблица переходов асинхронного  $RS$ -триггера в сокращенном виде

$R$	$S$	$Q^+$	Режим
0	0	$Q$	Хранение
0	1	1	Установка 1
1	0	0	Установка 0
1	1	*	Запрещенный

Функция переходов асинхронного  $RS$ -триггера может быть представлена в МКФ (с последующим переходом к базису ИЛИ-НЕ) или МДФ (к базису И-НЕ), получаемым в результате минимизации функции  $Q^+$ :

$$Q^+ = \bar{R} \cdot (S \vee Q) = \overline{\overline{\bar{R}} \cdot \overline{(S \vee Q)}} = \overline{\overline{\bar{R}} \vee \overline{(S \vee Q)}}, \quad (4.6)$$

$$Q^+ = S \vee QR = \overline{\overline{S \vee QR}} = \overline{\overline{S} \cdot \overline{QR}}, \quad (4.7)$$

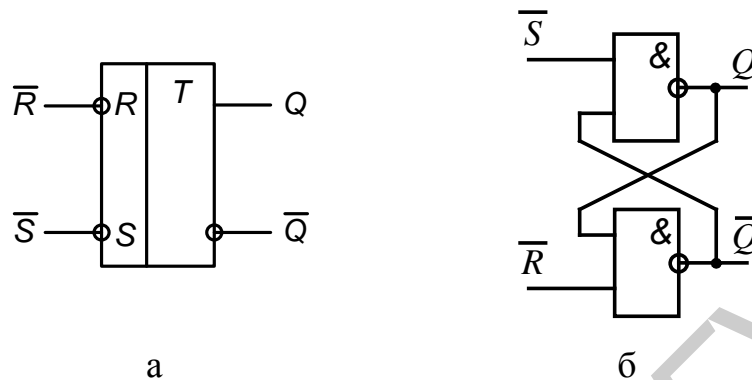
$$S \cdot R = 0. \quad (4.8)$$

Выражение (4.8) определяет условие реализации незапрещенных режимов работы  $RS$ -триггера.

Схема  $RS$ -триггера, синтезированная в соответствии с (4.6) на базе элементов ИЛИ-НЕ, представлена на рисунке 4.4, в.

Анализируя работу этой схемы для запрещенного режима (логические «1» на входах  $R$  и  $S$ ), получаем, что на прямом и инверсном выходах триггера будут присутствовать одинаковые логические уровни «0», что нарушает аксиому (1.4) алгебры логики.

При реализации схемы триггера по выражению (4.7) с использованием элементов И-НЕ (рисунок 4.5, б), отметим, что входными сигналами для триггера будут выступать  $\bar{R}$  и  $\bar{S}$ , т. е. активным уровнем для входов будет логический «0». Для этого варианта исполнения схемы условное графическое обозначение триггера представлено на рисунке 4.5, а.



а – условное графическое обозначение;

б – схемная реализация на элементах И-НЕ

Рисунок 4.5 – Асинхронный  $RS$ -триггер с инверсными входами  $R$  и  $S$

В этой схеме для запрещенного режима (логические «0» на входах  $R$  и  $S$ ) также наблюдается нарушение аксиомы (1.4), т. к. на прямом и инверсном выходах триггера будут присутствовать одинаковые логические уровни «1». Эта особенность схем триггера, выполненных как на элементах ИЛИ-НЕ, так и И-НЕ, и определяет то, что одновременная активация входов  $R$  и  $S$  запрещена. Также необходимо отметить, что углубленный анализ работы схем триггера, учитывающий задержку переключения в реальных логических элементах (см. подраздел 3.1), показывает, что кратковременное нарушение логики работы (возникновение одинаковых логических уровней на прямом и инверсном выходах триггера) может наблюдаться и для разрешенных сочетаний сигналов  $R$  и  $S$  при изменении одного из них с «0» на «1».

Таблица 4.3 переключений асинхронного  $RS$ -триггера, описывающая, какие уровни необходимо подавать на входы триггера (т. е. значения функций возбуждения входов) для обеспечения требуемого перехода на его выходе, получается путем преобразования данных таблицы 4.1 (4.2). Отметим, что для перехода выхода  $0 \rightarrow 0$  состояние функции возбуждения  $R$  не определено, т. к. этот переход может быть реализован как за счет использования режима хране-

ния ( $R = 0, S = 0$ ), так и за счет режима установки «0» ( $R = 1, S = 0$ ). Аналогичная ситуация возникает для функции возбуждения  $S$  в случае перехода  $1 \rightarrow 1$ , который можно реализовать, используя режим хранения ( $R = 0, S = 0$ ) или установки «1» ( $R = 0, S = 1$ ).

Таблица 4.3 – Таблица переключений асинхронного  $RS$ -триггера

$Q$	$Q^+$	$R$	$S$
0	0	*	0
0	1	0	1
1	0	1	0
1	1	0	*

Таблицу переключений можно представить графически – в виде графа переходов (рисунок 4.6).

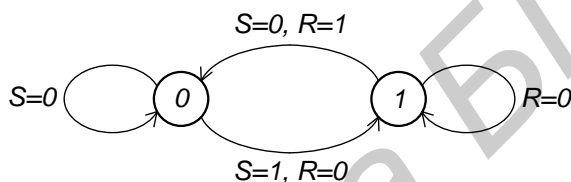


Рисунок 4.6 – Граф переходов  $RS$ -триггера

Схему асинхронного  $RS$ -триггера можно дополнить входом  $L$  (*Load* – загрузка) разрешения работы (рисунок 4.7, а). При наличии активного уровня (логической «1») на этом входе триггер работает как асинхронный  $RS$ -триггер, в противном случае – состояние триггера не меняется ни при каких значениях сигналов  $R$  и  $S$ , т. е. триггер находится в режиме хранения (таблица 4.4, рисунок 4.7, б).

Таблица 4.4 – Таблица переходов  $RS-L$ -триггера

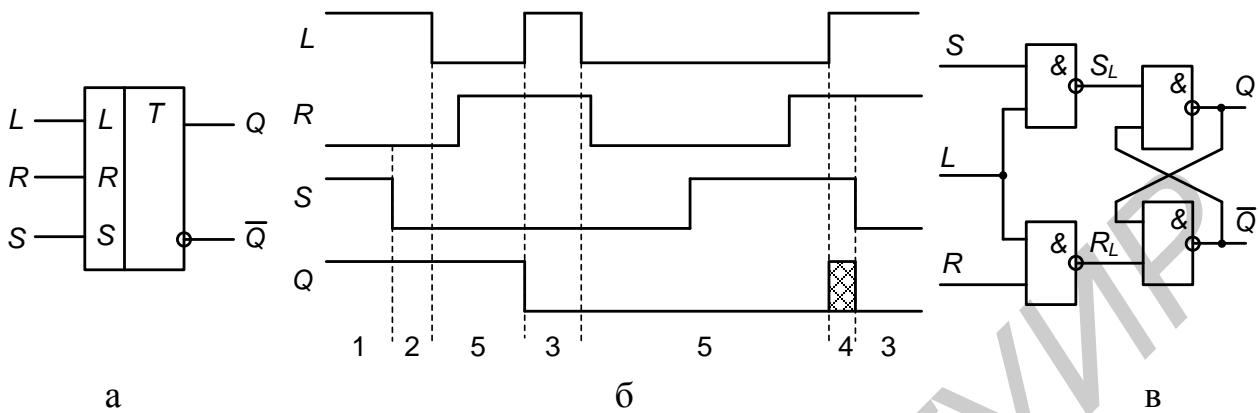
$L$	$R$	$S$	$Q^+$	Режим
0	*	*	$Q$	Хранение (по $L$ )
1	0	0	$Q$	Хранение (по $RS$ )
1	0	1	1	Установка 1
1	1	0	0	Установка 0
1	1	1	*	Запрещенный

Функция переходов  $RS-L$ -триггера описывается выражениями:

$$Q^+ = Q\bar{L} \vee L(S \vee Q\bar{R}) = SL \vee Q\bar{R} \vee Q\bar{L} = SL \vee Q\bar{R}\bar{L} = S_L \vee Q\bar{R}_L, \quad (4.9)$$

где  $S_L = SL$  и  $R_L = RL$ ;

$$L \cdot S \cdot R = 0. \quad (4.10)$$



а – условное графическое обозначение;

б – временные диаграммы работы (1 – режим установки «1»,

2 – хранение (по  $RS$ ), 3 – установка «0»,

4 – неопределенность, 5 – хранение (по  $L$ );

в – вариант схемной реализации

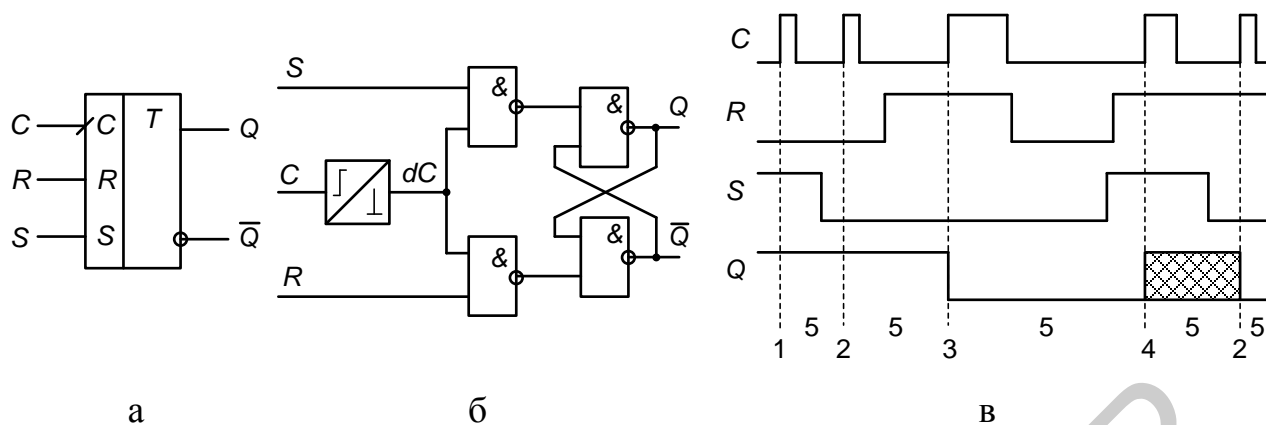
Рисунок 4.7 –  $RS-L$ -триггер

Выражению (4.9) соответствует схема, приведенная на рисунке 4.7, в, в которой для выходного  $RS$ -триггера дополнительно формируются сигналы возбуждения  $\overline{S}_L$  и  $\overline{R}_L$ . Выполнение выражения (4.10) позволяет избежать запрещенного режима  $L = S = R = 1$ .

Синхронный импульсный  $RS-C$ -триггер имеет импульсный вход синхронизации (рисунок 4.8, а). Его схему можно получить из схемы  $RS-L$ -триггера, используя по входу  $C$  формирователь короткого импульса по фронту входного сигнала, что соответствует  $dC = 1$  (рисунок 4.8, б).

Таблица переключений и функция переходов  $RS-C$ -триггера получается из таблицы 4.4 и выражений (4.9), (4.10) путем замены переменной  $L$  на  $dC$ . Таким образом, состояние выхода триггера может меняться только в момент тактирования (наличия переднего фронта в импульсе на входе  $C$ , рисунок 4.8, в).

$RS$ -триггеры применяются в качестве «защелок», а также служат основой построения схем других типов триггеров.



а – условное графическое обозначение;  
 б – вариант формальной схемной реализации;  
 в – временные диаграммы работы (1 – установка «1»,  
 2 – хранение (по  $RS$ ), 3 – установка «0», 4 – неопределенность,  
 5 – интервалы хранения (по  $C$ ))

Рисунок 4.8 –  $RS$ - $C$ -триггер

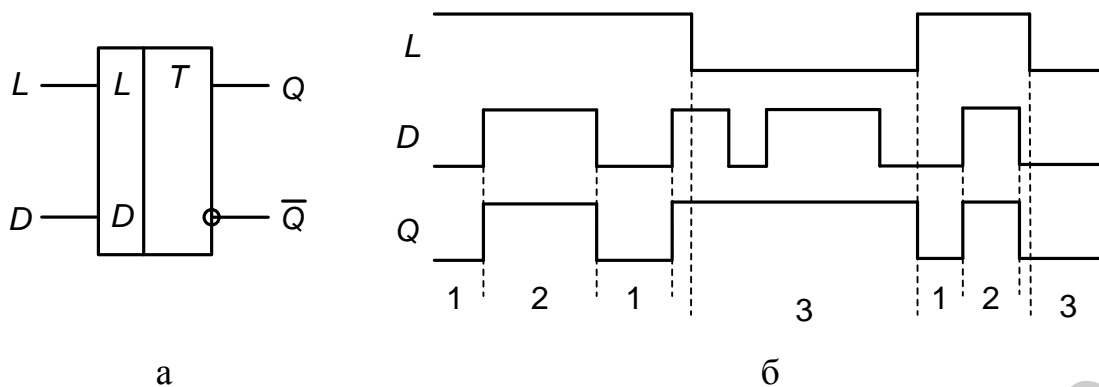
### 4.3.2 $D$ -триггеры

Асинхронный потенциальный  $D$ - $L$ -триггер имеет два входа – информационный  $D$  (*Data* – данные) и управления  $L$  (рисунок 4.9, а). При наличии логической «1» на этом входе  $D$ - $L$ -триггер пропускает данные со входа  $D$  на выход  $Q$  с задержкой, обусловленной конечным временем распространения. Это определило название  $D$ -триггеров как триггеров задержки (*Delay* – задержка) и использование в качестве элементов памяти (ЭП, см. подраздел 4.2). При наличии логического «0» на входе  $L$  – состояние триггера не меняется, т. е. триггер находится в режиме хранения (таблица 4.5, рисунок 4.9, б).

Таблица 4.5 – Таблица переходов  $D$ - $L$ -триггера

$L$	$D$	$Q^+$	Режим	
0	*	$Q$	Хранение (по $L$ )	
1	0	1	Установка 1	Загрузка данных
1	1	0	Установка 0	





а – условное графическое обозначение;  
 б – временные диаграммы работы (1 – установка «0»,  
 2 – установка «1», 3 – хранение (по  $L$ ))

Рисунок 4.9 –  $D-L$ -триггер

Функция переходов  $D-L$ -триггера описывается выражением

$$Q^+ = Q\bar{L} \vee LD \quad (4.11)$$

или в виде, свободном от состязаний,

$$Q^+ = Q\bar{L} \vee LD \vee QD. \quad (4.12)$$

Представив данную функцию в базисе И-НЕ

$$Q^+ = \overline{\overline{Q\bar{L} \vee LD \vee QD}} = \overline{\overline{QD} \cdot \overline{LD} \cdot \overline{Q\bar{L}}}, \quad (4.13)$$

получим схему  $D-L$ -триггера, называемого триггером Эрла (рисунок 4.10, а). Этот триггер имеет наибольшее быстродействие из всех триггеров, используемых для конвейерной обработки информации, т. к. комбинационная схема (при разрыве обратной связи) имеет второй порядок. Некоторым недостатком схемы можно считать отсутствие инверсного выхода.

Для синтеза схемы  $D-L$ -триггера на базе асинхронного  $RS$ -триггера применим методику, позволяющую осуществить реализацию любого триггера на базе другого типа триггеров. Для этого составляем полную таблицу переходов  $D-L$ -триггера и, используя таблицу 4.3 переключений  $RS$ -триггера, определяем значения функций возбуждения его входов  $R$  и  $S$  как триггера-прототипа, необ-

ходимые для обеспечения требуемых переключений проектируемого триггера (таблица 4.6).

Таблица 4.6 – Таблица переходов  $D$ - $L$ -триггера и функции возбуждения входов

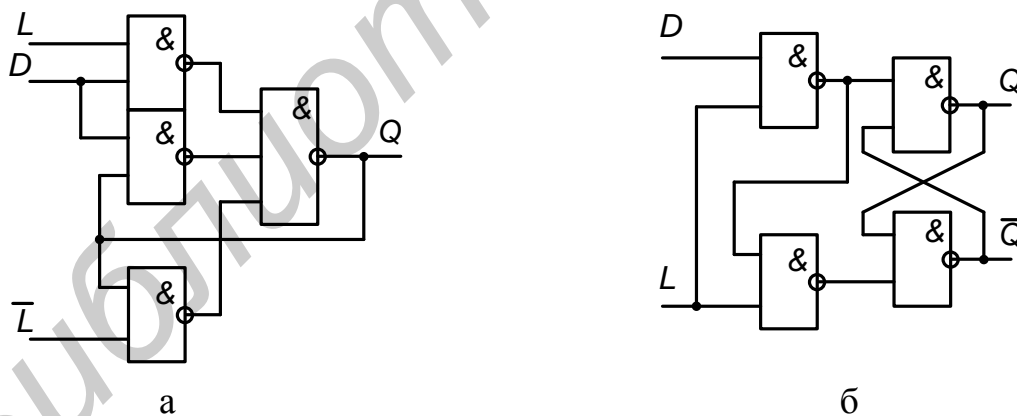
$L$	$D$	$Q$	$Q^+$	$R$	$S$
0	*	0	0	*	0
0	*	1	1	0	*
1	0	0	0	*	0
1	0	1	0	1	0
1	1	0	1	0	1
1	1	1	1	0	*

Используя эти данные для заполнения карт Карно и проведя минимизацию, получаем:

$$\bar{S} = F_1(L, D, Q) = \overline{LD}, \quad (4.14)$$

$$\bar{R} = F_2(L, D, Q) = \overline{LD} = \overline{L(\bar{D} \vee \bar{L})} = \overline{L(DL)} = \overline{LS}. \quad (4.15)$$

Схема  $D$ - $L$ -триггера, синтезированная по выражениям (4.14) и (4.15) на базе элементов И-НЕ, представлена на рисунке 4.10, б.



а – триггер Эрла;

б – на базе  $RS$ -триггера

Рисунок 4.10 – Схемная реализация  $D$ - $L$ -триггера

Граф переходов  $D$ - $L$ -триггера представлен на рисунке 4.11.

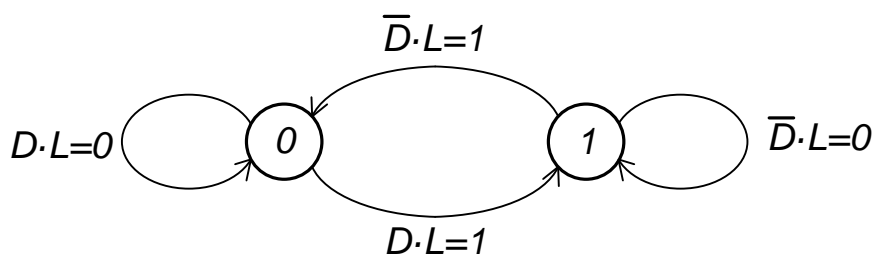
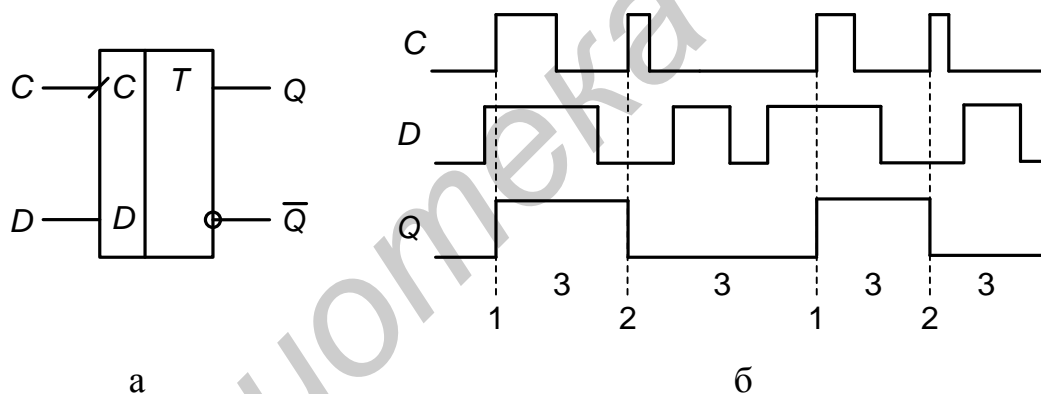


Рисунок 4.11 – Граф переходов  $D$ - $L$ -триггера

Второй тип  $D$ -триггера – синхронный импульсный. Наряду со входом  $D$  данных он имеет импульсный (динамический) вход  $C$  синхронизации (рисунок 4.12, а). В триггерах с динамическим входом  $C$  информация записывается только в течение перепада сигнала (перехода от логического «0» к логической «1») на входе  $C$ , т. е. при  $dC = 1$ . Таблицу переходов  $D$ - $C$ -триггера отображает таблица 4.7, таблицу переключений – 4.8, а временные диаграммы, поясняющие принцип функционирования, представлены на рисунке 4.12, б.



а – условное графическое обозначение;

б – временные диаграммы работы (1 – установка «1»,  
2 – установка «0», 3 – интервалы хранения данных (по  $C$ ))

Рисунок 4.12 –  $D$ - $C$ -триггер

Таблица 4.7 – Таблица переходов синхронного импульсного  $D$ - $C$ -триггера

$C$	$dC$	$D$	$Q^+$	Режим	
0, 1, ↓	0	*	$Q$	Хранение данных (по $C$ )	
↑	1	0	1	Установка 1	Загрузка данных
↑	1	1	0	Установка 0	

Функция переходов  $D$ - $C$ -триггера описывается выражением

$$Q^+ = D \cdot dC \vee Q \cdot \overline{dC}. \quad (4.16)$$

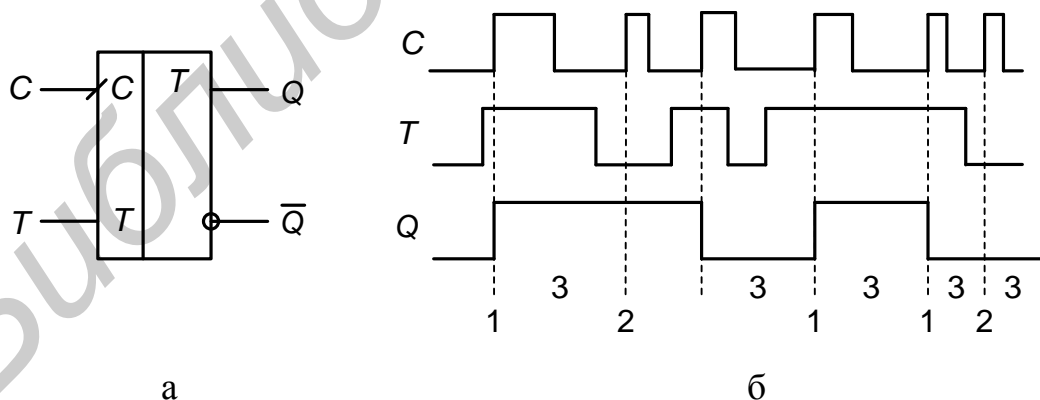
Таблица 4.8 – Таблица переключений синхронного импульсного  $D$ - $C$ -триггера (для моментов тактирования  $dC = 1$ )

$Q$	$Q^+$	$D$
0	0	0
0	1	1
1	0	0
1	1	1

$D$ -триггеры широко применяются при построении более сложных последовательностных устройств – счетчиков, регистров и т. п.

### 4.3.3 $T$ -триггеры

Синхронный импульсный  $T$ -триггер имеет информационный вход  $T$  и динамический вход  $C$  синхронизации (рисунок 4.13, а). При подаче на вход  $T$  уровня логического «0» триггер находится в режиме хранения, при активации входа – в режиме инверсии: по каждому такту на входе  $C$  состояние выхода триггера меняется на противоположное (таблица 4.9, рисунок 4.13, б).



а – условное графическое обозначение;  
 б – временные диаграммы работы (1 – инверсия, 2 – хранение,  
 3 – интервалы хранения данных (по  $C$ ))

Рисунок 4.13 – Синхронный импульсный  $T$ -триггер

Таблица 4.9 – Таблица переходов синхронного импульсного  $T$ -триггера

$C$	$dC$	$T$	$Q^+$	Режим
0, 1, ↓	0	*	$Q$	Хранение данных (по $C$ )
↑	1	0	$Q$	Хранение данных (по $T$ )
↑	1	1	$\bar{Q}$	Инверсия (переключение)

Функция переходов синхронного импульсного  $T$ -триггера описывается выражением

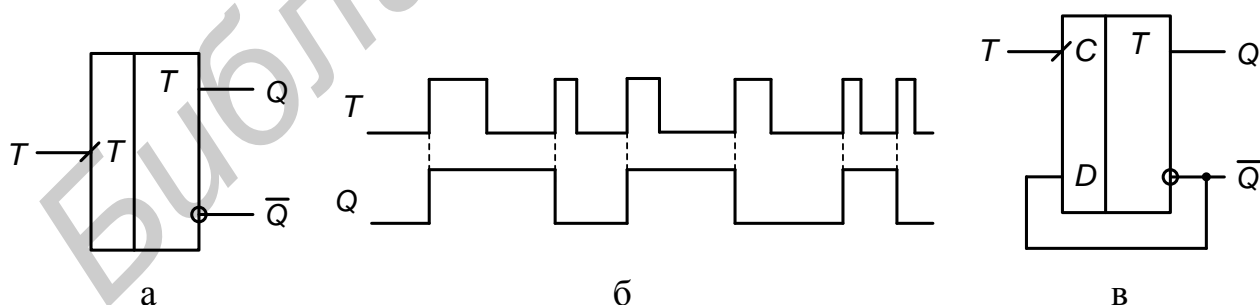
$$Q^+ = \bar{dC} \cdot Q \vee dC \cdot (T \cdot \bar{Q} \vee Q \cdot \bar{T}) = \bar{dC} \cdot Q \vee dC \cdot (T \oplus Q). \quad (4.17)$$

Таблица переключений синхронного импульсного  $T$ -триггера приведена в таблице 4.10.

Таблица 4.10 – Таблица переключений синхронного импульсного  $T$ -триггера (для моментов тактирования  $dC = 1$ )

$Q$	$Q^+$	$T$
0	0	0
0	1	1
1	0	1
1	1	0

Применяется также асинхронный импульсный  $T$ -триггер, имеющий только один вход (рисунок 4.14, а) и у которого инверсия состояния выхода осуществляется по переднему фронту сигнала на входе  $T$ , т. е. при  $dT = 1$  (рисунок 4.14, б).



а – условное графическое обозначение;

б – временные диаграммы работы;

в – схемная реализация на базе  $D$ - $C$ -триггера

Рисунок 4.14 – Асинхронный импульсный  $T$ -триггер

Функция переходов асинхронного импульсного Т-триггера описывается выражением

$$Q^+ = \overline{dT} \cdot Q \vee dT \cdot \overline{Q} = dT \oplus Q. \quad (4.18)$$

Такая схема может быть реализована на базе синхронного D-триггера (рисунок 4.14, в), если в выражении (4.16) принять  $dC = dT$  и  $D = \overline{Q}$ .

T-триггеры широко применяются при построении переключательных схем, в частности счетчиков.

#### 4.3.4 JK-триггеры

Синхронный импульсный JK-триггер имеет два информационных входа J и K и динамический вход C синхронизации (рисунок 4.15). По входу J триггер устанавливается в состояние  $Q = 1$ , а по входу K – в состояние  $Q = 0$ . Функционирование JK-триггера описывается данными таблицы 4.11. Этот триггер отличается от RS-триггера прежде всего тем, что в нем устранена неопределенность, которая возникает в RS-триггере при одновременной активации информационных входов.

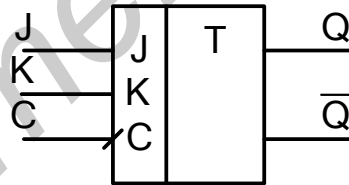


Рисунок 4.15 – Условное графическое обозначение JK-триггера

Таблица 4.11 – Таблица переходов синхронного импульсного JK-триггера

C	dC	J	K	Q <sup>+</sup>	Режим
0, 1, ↓	0	*	*	Q	Хранение
↑	1	0	0	Q	Хранение (по JK)
↑	1	0	1	0	Установка в 0
↑	1	1	0	1	Установка в 1
↑	1	1	1	Q-bar	Инверсия

Функция переходов JK-триггера описывается выражением

$$Q^+ = \overline{Q} \cdot J \cdot dC \vee Q \cdot \overline{K} \cdot dC. \quad (4.19)$$

Функциональные возможности  $JK$ -триггера определяют его как универсальный. Используя таблицу переключений (таблица 4.12) синхронного  $JK$ -триггера, легко получить схемные реализации синхронных  $D$ - $C$ - и  $T$ -триггеров на его основе (рисунок 4.16).

Таблица 4.12 – Таблица переключений синхронного  $JK$ -триггера (для моментов тактирования  $dC = 1$ )

$Q$	$Q^+$	$J$	$K$
0	0	*	0
0	1	0	1
1	0	1	0
1	1	0	*

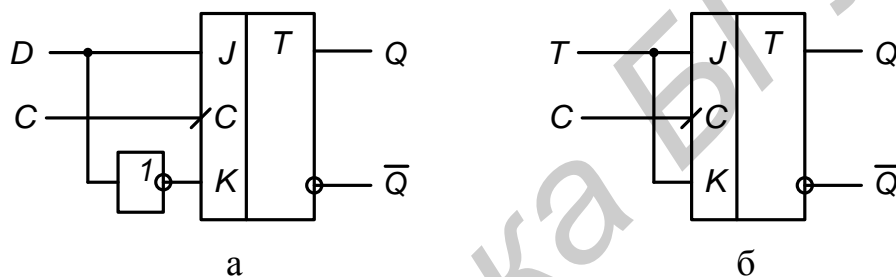


Рисунок 4.16 – Схемная реализация  $D$ - $C$ -триггера (а) и  $T$ -триггера (б) на базе синхронного  $JK$ -триггера

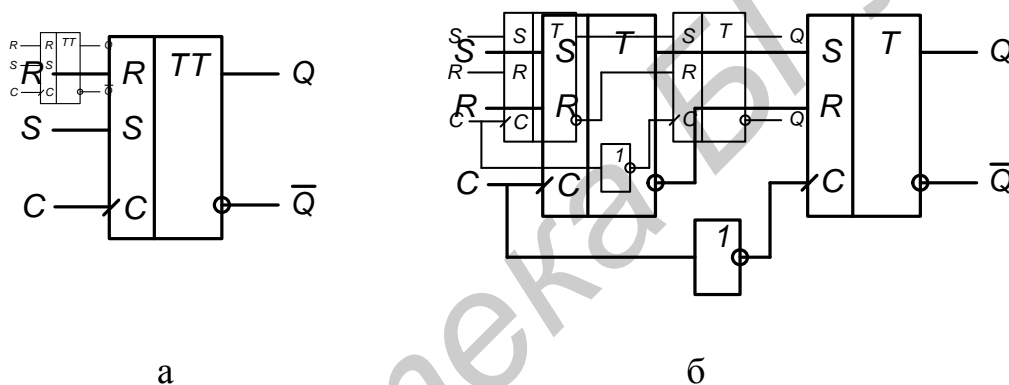
#### 4.3.5 Двухступенчатые $M$ - $S$ -триггеры

Одним из способов, позволяющих повысить помехоустойчивость (устранить «прозрачность») синхронных триггеров во время тактового импульса, является применение двухступенчатых схем, работающих по принципу «ведущий – ведомый».

$RS$ - $C$ -триггер типа  $M$ - $S$  (рисунок 4.17) состоит из двух последовательно включенных синхронных  $RS$ -триггеров. Эти триггеры синхронизируются одним тактовым сигналом, но на вход второго триггера этот сигнал поступает через инвертор. Когда в тактовом сигнале  $C$  присутствует передний фронт (переход с «0» в «1»), разрешается запись информации в ведущий ( $M$  – *Master*) триггер. При этом ведомый триггер ( $S$  – *Slave*) находится в режиме хранения предыдущего состояния, т. к. на его входе синхронизации  $dC2 = 0$ . Когда тактовый им-

пульс закончится и  $C$  станет равным нулю, триггер  $M$  перейдет в режим хранения, а в триггер  $S$  по заднему фронту тактового импульса будет записано состояние триггера  $M$ . После этого никакие изменения на выходе всей схемы произойти не смогут, т. к. ведущий триггер  $M$  заблокирован нулевым уровнем  $C$ . Следовательно, изменение состояния выходов триггера типа  $M-S$  возможно только по отрицательному перепаду (срезу) тактовых импульсов. При этом на выход передается та информация, которую несут сигналы  $S$  и  $R$  непосредственно перед фронтом тактового импульса.

Таким образом, двухступенчатые триггеры задерживают выходную информацию на время, равное длительности синхронизирующего импульса. Такие триггеры называют еще триггерами с внутренней задержкой.



а – условное графическое обозначение;

б – внутренняя структура

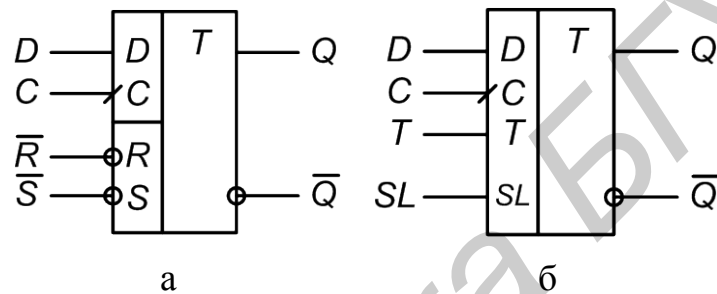
Рисунок 4.17 – Двухступенчатый синхронный  $RS-C$ -триггер

#### 4.3.6 Комбинированные триггеры

Комбинированные триггеры чаще всего представляют собой сочетание какого-либо типа синхронного ( $D-C$ -,  $T$ -,  $JK$ -) триггера с элементами асинхронного  $RS$ -триггера (рисунок 4.18, а). Такие триггеры отличаются наличием дополнительных входов – асинхронных входов  $S$  и  $R$  для предварительной установки триггера в определенное состояние (логической «1» или «0»). Названия типов таких триггеров составляются из названий их синхронных и асинхронных входов, например,  $D-C/RS$ -триггер означает синхронный  $D-C$ -триггер с асинхронными потенциальными входами  $R$  и  $S$ .



Другой тип комбинированных триггеров предполагает объединение в одной схеме функциональных возможностей двух различных триггеров (оба синхронные или оба асинхронные). Триггер, условное графическое обозначение которого приведено на рисунке 4.18, б, реализует возможности двух синхронных триггеров ( $D$ - $C$ - и  $T$ -). Для изменения режима работы используется дополнительный вход  $SL$  со следующей логикой: если  $SL = 0$ , то триггер работает как  $D$ -триггер, а если  $SL = 1$ , то как  $T$ -триггер. Возможны также реализации, в которых одному из информационных входов отдается приоритет, например, при  $T = 1$  триггер работает в режиме инверсии как  $T$ -триггер, а при  $T = 0$  – как  $D$ -триггер. В этом случае дополнительные входы управления не нужны.



а –  $D$ - $C$ / $RS$ -триггер;

б –  $D$ - $C$ - $T$ -триггер со входом  $SL$  управления режимами

Рисунок 4.18 – Условное графическое обозначение комбинированных триггеров

#### 4.4 Регистры

Регистр – цифровое устройство, предназначенное для кратковременного хранения и (или) преобразования многоразрядных двоичных кодов. Синтез схем регистров выполняется с использованием триггеров.

Регистры можно классифицировать по ряду признаков.

1 По функциональному назначению различают регистры:

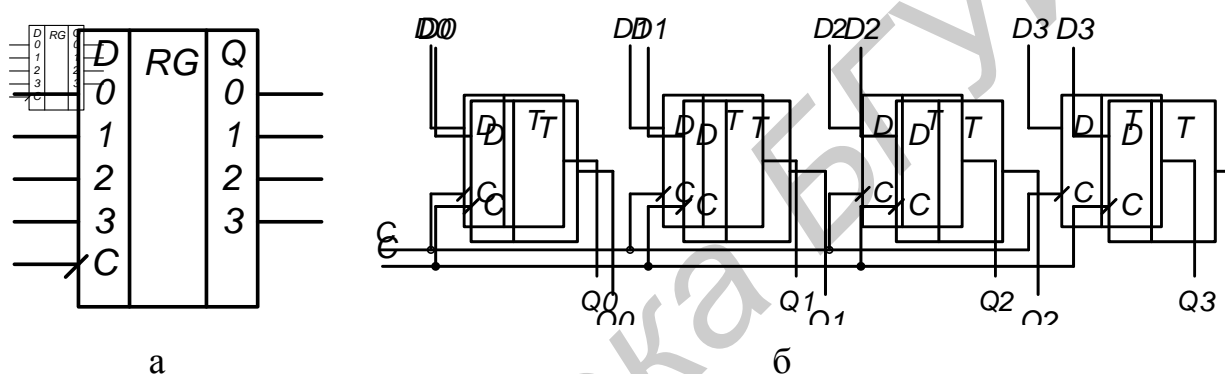
- хранения;
- сдвига;
- универсальные;
- специализированные.

2 По способу ввода и вывода цифровой информации:

- с параллельным вводом – выводом;
- с последовательным вводом – выводом;
- с параллельным вводом – последовательным выводом;
- с последовательным вводом – параллельным выводом.

#### 4.4.1 Регистр хранения (памяти)

Регистр памяти представляет собой систему синхронных триггеров (чаще всего  $D$ -триггеров) с общим тактовым входом и независимыми параллельными входами и выходами (рисунок 4.19).



а – условное графическое обозначение;

б – внутренняя структура

Рисунок 4.19 – Четырехразрядный регистр хранения

Ввод и вывод цифровых данных производится одновременно во всех разрядах. Данные в параллельном коде устанавливаются на входах  $D0...D3$  регистра, и по фронту тактового импульса осуществляется их запись в триггеры регистра. В отсутствие тактовых импульсов (при  $dC = 0$ ) регистр работает в режиме хранения. С приходом очередного фронта тактового импульса происходит обновление записанных ранее данных.

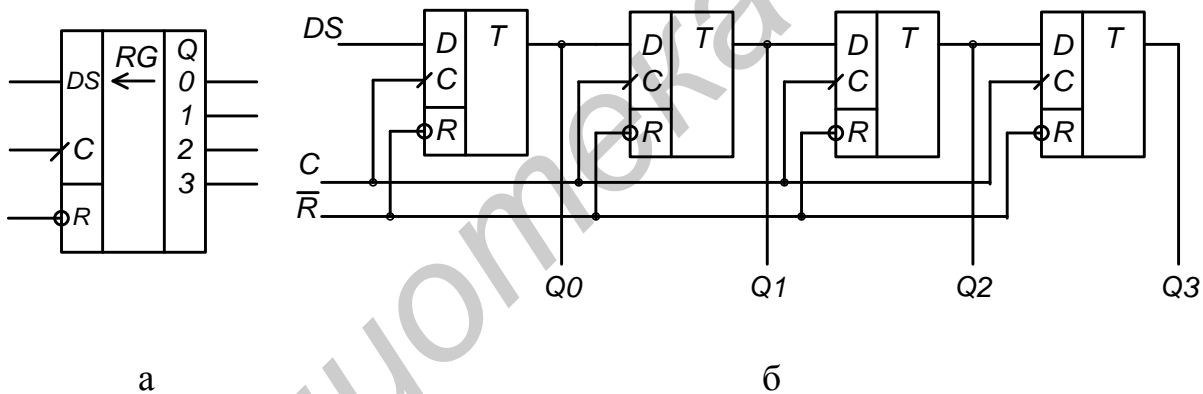
Для построения регистров памяти можно использовать триггеры как с динамическим, так и со статическим управлением, например  $D-L$ -триггеры. В этом случае, при  $L = 1$  регистр «прозрачен» для входных данных, а при  $L = 0$  осуществляется их «защелкивание».

Наращивание разрядности регистра достигается увеличением числа триггеров и подключением их тактовых входов к общей линии синхронизации.

#### 4.4.2 Регистры сдвига

Регистры этого типа предназначены для сдвига цифровых данных влево или вправо по разрядной сетке на определенное число разрядов. Различают однонаправленные и двунаправленные регистры сдвига. Ввод данных осуществляется в последовательном коде, а вывод – в последовательном или параллельном виде.

В простейшем однонаправленном регистре сдвига (рисунок 4.20) триггеры соединены последовательно, т. е. выходы предыдущего триггера передают информацию на входы последующего. Тактовые входы  $C$  триггеров соединены параллельно. Такой регистр имеет один последовательный вход данных  $DS$  (*Data Serial*) и один или несколько параллельных выходов. В первом случае используется выход  $Q3$  последнего триггера линейки, а сигналы с выходов  $Q0...Q2$  остальных триггеров для внешних схем не используются. Вход управления – тактовый вход  $C$ . Вход  $R$  асинхронного сброса используется для принудительного обнуления регистра.



а – условное графическое обозначение;  
б – внутренняя структура

Рисунок 4.20 – Четырехразрядный однонаправленный регистр сдвига

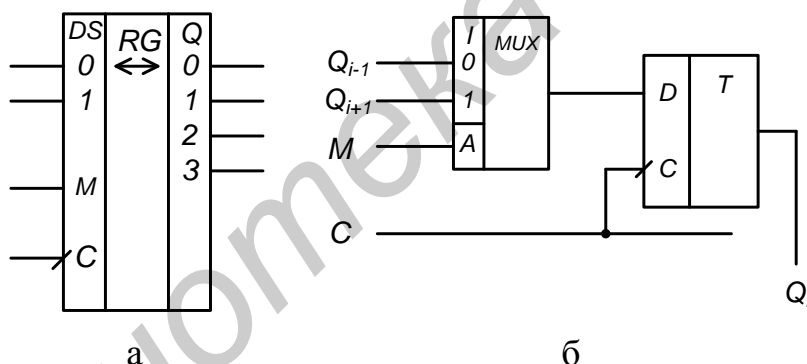
Приведенная на рисунке 4.20, б, структура соответствует так называемому регистру сдвига влево. Направление сдвига определяется не переносом данных по линейке триггеров на условном рисунке, а направлением сдвига данных по разрядам регистра – от младшего (выход  $Q0$ ) к старшему (выход  $Q3$ ), т. е. по разрядной сетке. Реализация сдвига вправо по разрядной сетке достигается простым переопределением назначений выходов (принятием соглашения, что выход  $Q0$  соответствует старшему разряду записанного числа).

Для реализации двунаправленного регистра сдвига с последовательным вводом (рисунок 4.21, а) необходимо к информационному входу каждого триггера добавить разрешающую логику, управляемую дополнительным сигналом  $M$  выбора направления сдвига.

Функции возбуждения входов  $D$  триггеров линейки описываются выражениями в предположении, что при  $M = 0$  в регистре осуществляется сдвиг данных влево, а при  $M = 1$  – вправо:

$$\begin{aligned} D_0 &= DS_0 \cdot \overline{M} \vee Q_1 \cdot M ; \\ D_i &= Q_{i-1} \cdot \overline{M} \vee Q_{i+1} \cdot M \quad \text{для } i = 1, 2; \\ D_3 &= Q_2 \cdot \overline{M} \vee DS_1 \cdot M . \end{aligned} \quad (4.20)$$

Реализовать данные функции можно как с использованием простейших логических элементов И, ИЛИ, НЕ, так и с использованием мультиплексов  $2 \rightarrow 1$  (рисунок 4.21, б), адресный вход которых управляется сигналом  $M$ .



а – условное графическое обозначение;

б – внутренняя структура для одного сегмента схемы

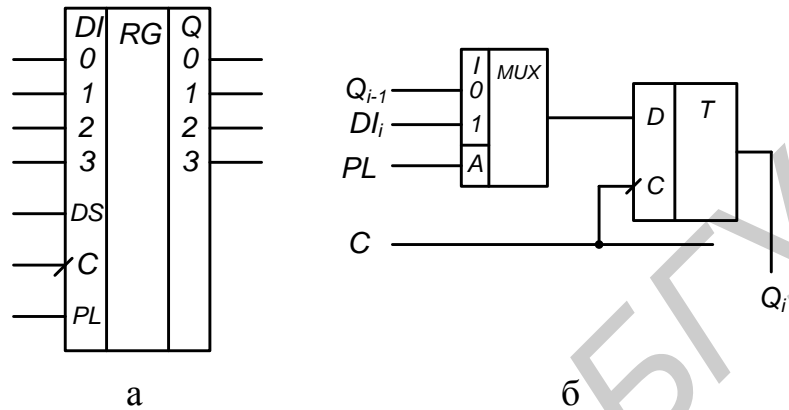
Рисунок 4.21 – Четырехразрядный двунаправленный регистр сдвига

Аналогичным образом решается задача реализации параллельного ввода данных в регистр сдвига. Такой регистр (рисунок 4.22, а) должен работать под управлением сигнала  $PL$  (*Parallel Load*) разрешения параллельной загрузки данных: при  $PL = 1$  данные со входов  $DI_0 \dots DI_3$  параллельно записываются в триггеры регистра, а при  $PL = 0$  осуществляется их сдвиг влево.

Функции возбуждения входов  $D$  триггеров линейки в данном случае описываются выражениями

$$\begin{aligned}
 D_0 &= DS \cdot \overline{PL} \vee DI_0 \cdot PL; \\
 D_i &= Q_{i-1} \cdot \overline{PL} \vee DI_i \cdot PL \text{ для } i = 1, 2; \\
 D_3 &= Q_2 \cdot \overline{PL} \vee DI_3 \cdot PL,
 \end{aligned}
 \tag{4.21}$$

схемная реализация которых также возможна с использованием мультиплексов  $2 \rightarrow 1$  (рисунок 4.22, б).



а – условное графическое обозначение;  
 б – внутренняя структура для одного сегмента схемы

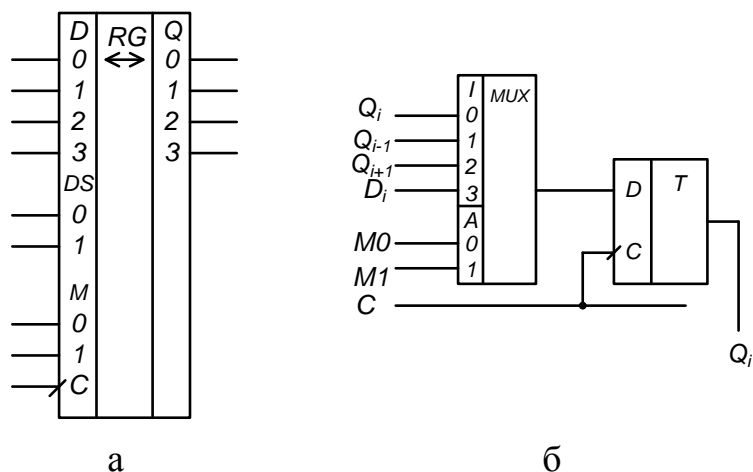
Рисунок 4.22 – Четырехразрядный однонаправленный регистр сдвига с параллельной загрузкой

Универсальные регистры сочетают в себе возможности регистров хранения и сдвига с возможностью параллельной или последовательной загрузки данных. Они требуют уже двух сигналов управления (рисунок 4.23, а) для реализации четырех режимов работы: хранение –  $M_0 = 0, M_1 = 0$ ; сдвиг влево –  $M_0 = 1, M_1 = 0$ ; сдвиг вправо –  $M_0 = 0, M_1 = 1$ ; параллельная загрузка –  $M_0 = 1, M_1 = 1$ .

Для такого универсального регистра функции возбуждения входов  $D$  триггеров описываются выражениями

$$\begin{aligned}
 D_0 &= Q_0 \cdot \overline{M_0} \cdot \overline{M_1} \vee DS_0 \cdot M_0 \cdot \overline{M_1} \vee Q_1 \cdot \overline{M_0} \cdot M_1 \vee DI_0 \cdot M_0 \cdot M_1; \\
 D_i &= Q_i \cdot \overline{M_0} \cdot \overline{M_1} \vee Q_{i-1} \cdot M_0 \cdot \overline{M_1} \vee Q_{i+1} \cdot \overline{M_0} \cdot M_1 \vee DI_i \cdot M_0 \cdot M_1; \\
 D_3 &= Q_3 \cdot \overline{M_0} \cdot \overline{M_1} \vee Q_2 \cdot M_0 \cdot \overline{M_1} \vee DS_1 \cdot \overline{M_0} \cdot M_1 \vee DI_3 \cdot M_0 \cdot M_1.
 \end{aligned}
 \tag{4.22}$$

Схемная реализация сегмента регистра приведена на рисунке 4.23, б.



а – условное графическое обозначение;  
 б – внутренняя структура для одного сегмента схемы

Рисунок 4.23 – Четырехразрядный универсальный регистр

Примером специализированного регистра может служить регистр последовательных приближений, применяемый при построении некоторых типов АЦП.

## ЛИТЕРАТУРА

- 1 Угрюмов, Е. П. Цифровая схемотехника : учеб. пособие / Е. П. Угрюмов. – 2-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2007. – 800 с.
- 2 Пухальский, Г. И. Цифровые устройства : учеб. пособие / Г. И. Пухальский. – СПб. : Политехника, 1996. – 885 с.
- 3 Брамер, Ю. А. Цифровые устройства : учеб. пособие / Ю. А. Брамер. – М. : Высш. шк., 2004. – 229 с.
- 4 Новиков, Ю. В. Основы цифровой схемотехники : базовые элементы и схемы. Методы проектирования / Ю. В. Новиков. – М. : Мир, 2001. – 379 с.
- 5 Точи, Р. Дж. Цифровые системы : теория и практика / Р. Дж. Точи, Н. С. Уидмер ; пер. с англ. – 8-е изд. – М. : Издат. дом «Вильямс», 2004. – 1024 с.
- 6 Бойт, К. Цифровая электроника / К. Бойт. – М. : Техносфера, 2007. – 472 с.
- 7 Амосов, В. В. Схемотехника и средства проектирования цифровых устройств : учеб. пособие / В. В. Амосов. – СПб. : БХВ-Петербург, 2007. – 560 с.
- 8 Бабич, Н. П. Основы цифровой схемотехники / Н. П. Бабич, И. А. Жуков. – М. ; Киев : Додэка-XXI ; МК-Пресс, 2007. – 480 с.

## СОДЕРЖАНИЕ

1	ОСНОВЫ АЛГЕБРЫ ЛОГИКИ .....	3
1.1	Аксиомы, законы и правила алгебры логики .....	3
1.2	Логические функции .....	7
1.2.1	Общие понятия .....	7
1.2.2	Функции двух переменных $F(x_2, x_1)$ .....	9
1.2.3	Принцип и закон двойственности.....	10
1.2.4	Теоремы разложения .....	10
1.2.5	Первичные термы, минтермы и макстермы .....	11
1.2.6	Формы аналитического представления логических функций.....	13
2	МИНИМИЗАЦИЯ ЛОГИЧЕСКИХ ФУНКЦИЙ.....	17
3	ТИПОВЫЕ КОМБИНАЦИОННЫЕ СХЕМЫ.....	21
3.1	Базовые логические элементы.....	21
3.2	Мультиплексоры и демультимплексоры .....	25
3.3	Шифраторы и дешифраторы .....	27
3.4	Сумматоры и вычитатели .....	31
3.5	Цифровые компараторы.....	34
3.6	Преобразователи кодов .....	36
4	ТИПОВЫЕ ПОСЛЕДОВАТЕЛЬНОСТНЫЕ СХЕМЫ.....	38
4.1	Потенциальные и импульсные сигналы.....	38
4.2	Модель последовательного автомата .....	40
4.3	Триггеры .....	42
4.3.1	<i>RS</i> -триггеры.....	43
4.3.2	<i>D</i> -триггеры .....	48
4.3.3	<i>T</i> -триггеры .....	52
4.3.4	<i>JK</i> -триггеры .....	54
4.3.5	Двухступенчатые <i>M-S</i> -триггеры .....	55
4.3.6	Комбинированные триггеры .....	56
4.4	Регистры .....	57
4.4.1	Регистры хранения (памяти) .....	58
4.4.2	Регистры сдвига.....	59
	ЛИТЕРАТУРА .....	63



*Учебное издание*

**Капуро Павел Александрович**

**ЦИФРОВЫЕ ФУНКЦИОНАЛЬНЫЕ УСТРОЙСТВА  
В ТЕЛЕКОММУНИКАЦИЯХ**

В 2-х частях

Часть 1

**БАЗОВЫЕ ЦИФРОВЫЕ ФУНКЦИОНАЛЬНЫЕ  
УСТРОЙСТВА**

**УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ**

Редактор *М. А. Зайцева*

Корректор *Е. Н. Батурчик*

Компьютерная правка, оригинал-макет *А. А. Лысеня*

Подписано в печать 10.07.2014. Формат 60×84 1/16. Бумага офсетная. Гарнитура «Таймс».  
Отпечатано на ризографе. Усл. печ. л. 3,95. Уч.-изд. л. 4,0. Тираж 100 экз. Заказ 267.

Издатель и полиграфическое исполнение: учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники».  
Свидетельство о государственной регистрации издателя, изготовителя,  
распространителя печатных изданий №1/238 от 24.03.2014,  
№2/113 от 07.04.2014, №3/615 от 07.04.2014.  
ЛП №02330/264 от 14.04.2014.  
2200013, Минск, П. Бровки, 6