

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА  
(РОСАВИАЦИЯ)

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ» (МГТУ ГА)

---

Кафедра технической эксплуатации  
радиоэлектронного оборудования воздушного транспорта

С.Б. Стукалов, В.А. Костенков,  
Э.А. Болелов, В.И. Петров

# СХЕМОТЕХНИКА

## ЦИФРОВАЯ СХЕМОТЕХНИКА

**Учебное пособие**

*Утверждено редакционно-  
издательским советом МГТУ ГА  
в качестве учебного пособия*

Москва  
ИД Академии Жуковского  
2021

УДК 621.3.049.77

ББК 6Ф2.12

С88

Печатается по решению редакционно-издательского совета  
Московского государственного технического университета ГА

Рецензенты:

*Яманов Д.Н.* (МГТУ ГА) – канд. техн. наук, доцент;

*Небусев С.В.* (АО НПО «Прибор» им. С.С. Голембиовского) – канд. техн. наук, доцент

### **Стукалов С.Б.**

С88

Схемотехника. Цифровая схемотехника [Текст] : учебное пособие /  
С.Б. Стукалов, В.А. Костенков, Э.А. Болелов, В.И. Петров. – М. : ИД Ака-  
демии Жуковского, 2021. – 148 с.

ISBN 978-5-907275-99-7

Учебное пособие подготовлено в соответствии с программой по курсу дисциплины  
«Схемотехника» и требованиями государственного образовательного стандарта высше-  
го образования. Учебное пособие предназначено для студентов специальности 25.05.03  
«Техническая эксплуатация транспортного радиооборудования всех форм обучения.  
Часть 2. Цифровые устройства».

В учебном пособии излагаются современные подходы построения логических  
элементов, комбинационных цифровых устройств, последовательностных цифровых  
устройств. Представлены схемотехнические решения современных бортовых и назем-  
ных цифровых устройств. В центре внимания учебного пособия находится применение  
анализ типовых устройств, используемых в оборудовании воздушных судов и аэропор-  
тов гражданской авиации. Показаны принципы их работы, технические характеристики.  
Особенностью учебного пособия является новизна учебного материала и актуальность,  
связанная с активным внедрением в гражданскую авиацию и новых технологий цифровой  
схемотехники.

Материал учебного пособия может быть также полезен студентам, обучающихся по  
другим направлениям специальностей связанных с радиотехникой.

Рассмотрено и одобрено на заседаниях кафедры 20.02.2021 г. и методического со-  
вета 20.02.2021 г.

**УДК 621.3.049.77**

**ББК 6Ф2.12**

Св. тем. план 2021 г.  
поз. 25

СТУКАЛОВ Сергей Борисович, КОСТЕНКОВ Владимир Александрович,

БОЛЕЛОВ Эдуард Анатольевич, ПЕТРОВ Виктор Иванович

СХЕМОТЕХНИКА. ЦИФРОВАЯ СХЕМОТЕХНИКА

Учебное пособие

*В авторской редакции*

Подписано в печать 26.05.2021 г.

Формат 60x84/16 Печ. л. 9,25 Усл. печ. л. 8,6

Заказ № 773/0429-УП08 Тираж 40 экз.

Московский государственный технический университет ГА

125993, Москва, Кронштадтский бульвар, д. 20

Издательский дом Академии имени Н. Е. Жуковского

125167, Москва, 8-го Марта 4-я ул., д. 6А

Тел.: (495) 973-45-68 E-mail: zakaz@itsbook.ru

**ISBN 978-5-907275-99-7**

© Московский государственный технический  
университет гражданской авиации, 2021

## Список принятых сокращений

АЛУ –	арифметико-логического устройства;
АЦП –	аналогово-цифровой преобразователь;
БИС –	большая интегральная схема;
ДТЛ –	диодно-транзисторная логика;
ИМС –	интегральная микросхема;
ИС –	интегральная схема;
КМОП –	логика на комплектарных полевых транзисторах структуры метал-оксид-полупроводник;
КС –	комбинационная схема;
ЛФ –	логическая функция;
ЛЭ –	логический элемент;
ПНЧ –	преобразователь напряжения в частоту;
РЛС –	радиолокационная станция;
СДНФ –	совершенная дизъюнктивная нормальная форма;
СКНФ –	совершенная конъюнктивная нормальная форма;
ТТЛ –	транзисторно-транзисторная логика;
ТТЛШ –	транзисторно-транзисторная логика с диодами Шоттки;
УГО –	условное графическое обозначение;
ФАЛ –	функции алгебры логики;
ЦАП –	цифро-аналоговый преобразователь;
ЦУ –	цифровое устройство;
ЭВМ –	электронно-вычислительная машина;
ЭСЛ –	эмиттерно-связанная логика.

## Введение

Цифровые устройства схемотехники в настоящее время занимают важное положение в технике гражданской авиации. Такие устройства выполняют функции, необходимые для работы почти всех бортовых и наземных технических устройств.

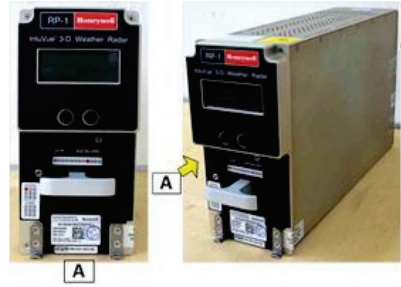
Отечественным ученым принадлежит приоритет в разработке ряда разделов теории схемотехники: М.А. Гаврилов опубликовал первую монографию по теории контактных схем, П.П. Пархоменко З.Н. Рогинский разработали первые машины автоматического анализа и синтеза схем; фундаментальные исследования проведены Э.А. Якубайтисом по синтезу асинхронных автоматов, В.Г. Лазаревым - по синтезу программных автоматов, В.Н. Рогинским – в области динамических автоматов. Первые работы по применению теории помехоустойчивого кодирования для повышения надежности схемотехнических устройств были опубликованы М.А. Гавриловым и А.Д. Закревским.

В качестве примеров областей внедрения цифровых устройств схемотехники можно отнести: спутниковую навигацию воздушных судов, спутниковую связь, цифровое телевидение, сотовую связь, мобильный интернет и т.д. Цифровые устройства схемотехники позволяют осуществить передачу информации с высокой достоверностью. Цифровые способы хранения данных обеспечивают возможность долгосрочного сохранения информации без потерь, обеспечивают более высокие характеристики. Методы цифровой обработки сигналов позволяют реализовать такие алгоритмы, которые сложно реализовать с использованием аналоговых устройств (например, специальные характеристики фильтров). Использование данных подходов позволяет существенно повысить характеристики радиотехнической аппаратуры. Большинство операций может быть реализовано программным способом. Ввиду этого достигается значительная функциональность всего комплекса используемой аппаратуры. Цифровые устройства схемотехники позволяют выполнить смену выполняемых функций, стандартов, видов модуляции. Можно изменить алгоритмы обработки. Подобного типа операции осуществляются простой перезагрузкой новой программы. Цифровые устройства схемотехники успешно решают проблему перегруженности радиочастотного диапазона и позволяют более эффективно использовать его. За счет цифровых методов обработки объемы передаваемой информации существенно увеличиваются.

Цифровые устройства схемотехники активно применяются в составе бортового оборудования современных воздушных судов (рис.1.1).



а).



б).

Рис.1.1 – Самолет MC 21(а), вычислитель метеолокатора MC 21 (б)

Наличие цифровых устройств схемотехники позволяет обеспечить эффективную работу наземного оборудования обеспечения полетов (рис.1.2).



а).



б).

Рис.1.2 – Аэродромный обзорный радиолокатор (а), пульт управления аэродромного радиолокатора (б)

Цифровые устройства схемотехники радиооборудования воздушного транспорта работают в условиях сложной эксплуатационной обстановки, связанной с полетом. Большинство устройств имеют длительный срок службы, непрерывный характер работы и выполняют ответственные функции по обеспечению безопасности полета. Вследствие этого цифровые схемотехнические устройства радиооборудования воздушных судов и аэропортов имеют большие перспективы построения и применения в новой технике.

## Системы счисления и арифметические операции

### 1.1. Общие сведения о системах счисления

Система изображения любых чисел с помощью ограниченного числа символов называется системой счисления. Используемые в системе счисления символы называются цифрами.

Исторически сложились две системы счисления: непозиционная и позиционная.

Примером непозиционной системы счисления является римская система, в которой используются римские цифры I, V, X, L, C, M. Каждой цифре в любом месте числа соответствует одно и то же значение – количественный эквивалент. Общим недостатком непозиционных систем счисления являются трудности записи больших чисел и выполнения арифметических операций. Поэтому в цифровой технике непозиционные системы не нашли применения.

В цифровых устройствах используются позиционные системы счисления. Система счисления называется изолированной, если одна и та же цифра имеет различное значение, которое определяется ее позицией в последовательности цифр, изображающей число.

Рассмотрим принцип, используемый при записи чисел в привычной для нас десятичной системе счисления.

Число представляется в виде последовательности цифр, разделенных запятой на две группы: одна группа (левее запятой) образует целую часть, другая (правее запятой) – дробную часть. Каждая цифра числа занимает определенную позицию, называемую разрядом. Разрядом приписываются различные весовые коэффициенты. Эти коэффициенты для разрядов влево от запятой равны соответственно  $10^2$ ,  $10^1$ ,  $10^0, \dots$ ; вправо –  $10^{-1}$ ,  $10^{-2}$ ,  $10^{-3} \dots$ . Таким образом, запись 547,359 в десятичной системе счисления означает следующее количество:

$$547,359 = 5 \cdot 10^2 + 4 \cdot 10^1 + 7 \cdot 10^0 + 3 \cdot 10^{-1} + 5 \cdot 10^{-2} + 9 \cdot 10^{-3}.$$

В общем случае изображение некоторого числа  $N$  будет иметь вид:

$$N = a_n \cdot p^n + a_{n-1} \cdot p^{n-1} + \dots + a_1 \cdot p^1 + a_0 \cdot p^0 + a_{-1} \cdot p^{-1} + a_{-2} \cdot p^{-2} + \dots + a_{-m} \cdot p^{-m}. \quad (1.1)$$

Здесь  $p$  – основание системы счисления, которое определяет число используемых в ней символов – цифр  $a_n$ .

Используя такой принцип можно строить разнообразные позиционные системы счисления. Основным преимуществом таких систем является удобство выполнения арифметических операций. Поэтому в цифровой технике, как правило, применяются позиционные системы счисления.

## 1.2. Системы счисления применяемые в дискретных устройствах

Выбор основания системы счисления зависит от физических элементов, на основе которых строится то или иное устройство. В цифровой технике широко используются элементы с двумя устойчивыми состояниями – включено, выключено. Поэтому в технике наибольшее распространение имеет двоичная система счисления.

Основание двоичной системы  $p=2$ . Для представления цифр разрядов используются лишь два символа: 0 и 1.

Например, число  $11010,11_2$  соответствует следующему количеству:

$$11010,11_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} = 26,7510 ,$$

где индексы при числах показывают основание системы.

Наряду с двоичной системой счисления в цифровой технике применяется восьмеричная и шестнадцатеричная системы счисления.

Восьмеричная система счисления ( $p=8$ ). Для представления цифр разрядов используется восемь символов: 0,1,2...7.

Например, восьмеричное число  $756,25_8$

$$756,25_8 = 7 \cdot 8^2 + 5 \cdot 8^1 + 6 \cdot 8^0 + 2 \cdot 8^{-2}$$

равно десятичному числу  $494,328125_{10}$ .

Шестнадцатеричная система счисления ( $p=16$ ). Для представления цифр разрядов употребляются 16 символов (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F). Буквам латинского алфавита A,B,C,D,E,F соответствуют цифры десятичной системы 10,11,12,13,14,15.

Запись числа  $A7B,C8_{16}$  соответствует следующему количеству:

$$A7B,C8_{16} = 10 \cdot 16^2 + 7 \cdot 16^1 + 11 \cdot 16^0 + 12 \cdot 16^{-1} + 8 \cdot 16^{-2} = 2683,78125_{10} .$$

Кроме перечисленных систем счисления используются двоично-кодированные десятичные системы. В этих системах каждая цифра десятичного числа предварительно представляется в двоичной форме. Наиболее применяемые коды приведены в табл. 1.1.

Десятичные цифры	Двоично-кодированные десятичные системы (Веса)					
	8421	2421	2 из 5	С избытком 3	3a+2	7421
0	0000	0000	11000	0011	00010	0000
1	0001	0001	01100	0100	0010	0001
2	0010	0010	00110	0101	01000	0010
3	0011	0011	00011	0110	01011	0011
4	0100	0100	10001	0111	01110	0100
5	0101	1011	10100	1000	10001	0101
6	0110	1100	01010	1001	10100	0110
7	0111	1101	00101	1010	10111	1000
8	1000	1110	10010	1011	11010	1001
9	1001	1111	01001	1100	11101	1010

Следует заметить, что несмотря на внешнее сходство двоично-кодированного десятичного числа с двоичным числом, оно не является двоичным. Эти коды используются для обнаружения ошибочных комбинаций, для выполнения арифметических операций над десятичными числами.

### 1.3. Формы представления чисел в дискретных устройствах

Естественная форма представления числа (фиксированная точка)

$$+0,75_{10} = +0,11_2; -7 = -111_2.$$

В разрядной сетке ЭВМ для записи числа с фиксированной точкой отводится несколько разрядов (обычно 8, 16, 32), при этом в первом разряде кодируется знак числа: 0, если число положительное и 1, если число отрицательное. Если число по абсолютной величине больше единицы, то точка фиксируется после младшего разряда; если меньше единицы, то перед старшим разрядом.



Пример:

0	1	1	0	0	0	0	0
---	---	---	---	---	---	---	---

– соответствует числу  $+0,11_2$ ;

1	0	0	0	0	1	1	1
---	---	---	---	---	---	---	---

– соответствует числу  $-111_2$ .

Нормальная форма представления числа (плавающая точка)

$$+75_{10} = +0,75 \cdot 10^2_{10} = +0,11 \cdot 10^{+10}_2;$$

где  $0,75_{10} = 0,11_2$  – мантисса числа;

$10_{10} = 10_2$  – основание числа;

$2_{10} = 10_2$  – порядок числа.

Аналогично производится запись в разрядной сетке

1	1	1	0	0	0	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---	---

ЗН: Мантисса

ЗН: Порядок

**М**

**П**

#### 1.4. Перевод чисел из одной системы счисления в другую

Основания восьмеричной и шестнадцатеричной систем счисления выражаются целой степенью двух ( $8=2^3$ ,  $16=2^4$ ). Для перевода чисел из восьмеричной системы в двоичную достаточно каждую цифру восьмеричного числа представить трехразрядным двоичным числом – триадой.

Например:

$$762,358 = \underset{7}{111} \underset{6}{110} \underset{2}{010}, \underset{3}{011} \underset{5}{101}_2$$

Перевод шестнадцатеричных чисел в двоичную систему счисления осуществляется представлением цифр четырехразрядными двоичными числами – тетрадами. Например:

$$A7B,C7_{16} = \underset{A}{1010} \underset{7}{0111} \underset{B}{1011}, \underset{1100}{1100} \underset{0111_2}{0111}_2$$

Перевод чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную производится в обратном порядке. Например:

$$001\ 111, 101\ 010_2 = 17,52_8; 0101\ 1100, 1011_2 = 5C,D_{16}$$

1

7

5

2

5

C

D

При переводе чисел с некратными основаниями, содержащих целую и дробную части пользуются универсальным алгоритмом. Согласно этому алгоритму, перевод чисел состоит из вычислительных процессов двух видов:

- последовательного деления целой части и образующихся целых частных на основание новой системы счисления;

- последовательного умножения дробей части и дробных частей, получающихся произведений на то же новое основание, записанное, как в первом случае, цифрами исходной системы счисления.

*Например:* требуется перевести число  $30,6_{10}$  в двоичную систему счисления. Перевод целой части – деление на основание 2.

$$\begin{array}{r|l}
 30_{10} = 1 & \left| \begin{array}{l} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{array} \right. \\
 & \left| \begin{array}{l} 0_2 \quad 30 \text{L}^2 \\ \quad \underline{30} \quad 15 \text{L}^2 \\ \quad \quad 0 \quad \underline{14} \quad 7 \text{L}^2 \\ \quad \quad \quad 1 \quad \underline{6} \quad 3 \text{L}^2 \\ \quad \quad \quad \quad 1 \quad \underline{2} \quad 1 \\ \quad \quad \quad \quad \quad 1 \end{array} \right.
 \end{array}$$

Перевод дробной части – умножение на основание 2

$$\begin{array}{r|l}
 0,1 & \left| \begin{array}{l} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{array} \right. \\
 & \left| \begin{array}{l} \times 0,6 \\ \quad \underline{2} \\ \times 1,2 \\ \quad \underline{2} \\ \times 0,4 \\ \quad \underline{2} \\ \times 0,8 \\ \quad \underline{2} \\ \quad \quad 1,6 \end{array} \right.
 \end{array}$$

В результате получим двоичное число

$$11110,1001\dots_2 = 30,6$$

Существенное сокращение количества операций деления можно получить, используя промежуточный переход к шестнадцатеричной или восьмеричной системам счисления, а затем к двоичной, например:

$$64_{10} = 80_8 = 1,000,000_2$$

$$64_{10} = 40_{16} = 100,0000_2$$

## 1.5. Арифметические операции в двоичной системе счисления

Основной операцией, которая используется в цифровых устройствах при выполнении различных арифметических действий, является операция *алгебраического сложения чисел*, т. е. сложения, в котором могут участвовать как положительные, так и отрицательные числа. Вычитание легко сводится к сложению путем изменения на обратный знак вычитаемого, а операции умножения и деления также сводятся к алгебраическому сложению и некоторым логическим действиям.

Сложение двух чисел в двоичной системе счисления выполняется на основе таблицы двоичного сложения:

$$\begin{aligned} 0 + 0 &= 0, \\ 0 + 1 &= 1, \\ 1 + 0 &= 1, \\ 1 + 1 &= 10. \end{aligned}$$

Двузначная сумма в последнем случае означает, что при сложении двух двоичных цифр, равных 1, в каком-либо разряде двоичного числа возникает перенос в соседний старший разряд. Этот перенос должен быть приведен к сумме цифр, образовавшейся в соседнем разряде.

При сложении двух многоразрядных двоичных чисел цифры разрядов суммы формируются последовательно, начиная с младшего разряда. Цифра младшего разряда суммы образуется суммированием цифр младших разрядов слагаемых. При этом кроме цифр суммы формируется цифра переноса в следующий, более старший разряд, если оба младших разряда единицы. Таким образом, в разрядах, начиная со второго, могут суммироваться три цифры: цифры соответствующего разряда слагаемых и перенос, поступающий в данный разряд из предыдущего. Этот перенос равен 1 во всех случаях, когда результат суммирования цифр в разряде равен или больше 2, поскольку 2 является основанием системы счисления. При этом в разряд суммы записывается цифра, на две единицы меньшая результата суммирования.

*Пример сложения двух многоразрядных двоичных чисел:*

$$\begin{array}{r} 1101101 \text{ – первое слагаемое} \\ + \\ \underline{1001111} \text{ – второе слагаемое} \\ 0100010 \text{ – поразрядная сумма без учета переносов} \\ + \\ \underline{1 \ 11 \ 1} \text{ . – переносы} \\ 10111100 \text{ – окончательная сумма.} \end{array}$$

В цифровой технике операция вычитания реализуется как алгебраическое сложение с применением специальных кодов.

При этом операция вычитания сводится к операции простого арифметического сложения при помощи обратного и дополнительного кодов, используемых для представления отрицательных чисел.

*Обратный код* отрицательных двоичных чисел может быть сформирован по следующему правилу: цифры всех разрядов, кроме знакового, заменяются на обратные (инвертируются) – единицы заменяются нулями, а нули единицами. В знаковый разряд ставится единица. Обратное преобразование из обратного кода в прямой производится по тому же правилу. При использовании обратного кода операция вычитания реализуется как арифметическое сложение положительного числа, представленного в прямом коде, с отрицательным числом, представленным в обратном коде.

Например, при вычитании из числа 10110 числа 01101 уменьшаемое представляется как положительное число в прямом коде **0** 10110, а вычитаемое – как отрицательное число в обратном коде **1** 10010. В представлении чисел знаковые разряды выделены полужирным шрифтом. При выполнении операции арифметического сложения над этими числами получаем алгебраическую сумму:

$$\begin{array}{r}
 \mathbf{0} \ 10110 \text{ – первое слагаемое в прямом коде} \\
 + \\
 \mathbf{1} \ 10010 \text{ – второе слагаемое в обратном коде} \\
 \mathbf{1} \ 01000 \\
 \hline
 \begin{array}{l}
 \left| \xrightarrow{+} \right. \\
 \mathbf{0} \ 01001 \text{ – сумма в прямом коде}
 \end{array}
 \end{array}$$

Перенос, возникающий из знакового разряда, при использовании обратного кода должен прибавляться в младший разряд суммы. В данном примере уменьшаемое по модулю больше вычитаемого, поэтому алгебраическая сумма положительная и представлена в прямом коде. При изменении знаков слагаемых в приведенном примере на обратные

$$\begin{array}{r}
 \mathbf{1} \ 01001 \text{ – первое слагаемое в обратном коде} \\
 + \\
 \mathbf{0} \ 01101 \text{ – второе слагаемое в прямом коде} \\
 \mathbf{1} \ 10110 \text{ – сумма в обратном коде}
 \end{array}$$

результатом сложения будет отрицательное число, и оно будет представлено в обратном коде.

*Дополнительный код* отрицательных двоичных чисел может быть сформирован по следующему правилу: цифры всех разрядов, кроме знакового, инвертируются, и в младший разряд прибавляется единица. Дополнительный код может быть получен и из обратного путем прибавления единицы к младшему разряду обратного кода. При этом в знаковый разряд отрицательного числа в

дополнительном коде ставится единица. Обратное преобразование из дополнительного кода в прямой производится по тому же правилу.

При использовании дополнительного кода для вычитания двоичных чисел из предыдущего примера получим

$$\begin{array}{r} 0 \quad 10110 \text{ – первое слагаемое в прямом коде} \\ + \\ \underline{1 \quad 10011} \text{ – второе слагаемое в дополнительном коде} \\ 0 \quad 01001 \text{ – сумма в прямом коде} \end{array}$$

При сложении складываются цифры знаковых разрядов с отбрасыванием возникающего из этого разряда переноса. Алгебраическая сумма, полученная в результате сложения, является положительным числом и поэтому представлена в прямом коде. Если знаки слагаемых меняются на обратные:

$$\begin{array}{r} 0 \quad 01010 \text{ – первое слагаемое в дополнительном коде} \\ + \\ \underline{0 \quad 01101} \text{ – второе слагаемое в прямом коде} \\ \underline{1 \quad 10111} \text{ – сумма в дополнительном коде} \end{array}$$

то результат сложения есть отрицательное число, и оно оказывается представленным в дополнительном коде.

При алгебраическом сложении двоичных чисел в образовавшейся сумме возможно переполнение разрядной сетки, которое заключается в том, что результат операции – сумма содержит большее число разрядов, чем число разрядов в устройстве, предназначенном для их хранения. Для выявления переполнения разрядной сетки используется *модифицированный код*. В нем два знаковых разряда и в обоих разрядах положительные числа содержат нули, а отрицательные числа – единицы. Выполнение операций суммирования с использованием модифицированного дополнительного или модифицированного обратного кода производится по сформулированным выше правилам. Если результат суммирования содержит в знаковых разрядах комбинации 01 или 10, то это служит признаком переполнения разрядной сетки.

*Например*, при сложении чисел:

$$\begin{array}{r} 00 \quad 11011 \text{ – первое слагаемое в прямом модифицированном коде} \\ + \\ 11 \quad 01011 \text{ – второе слагаемое в дополнительном модифицированном} \\ \text{коде} \\ \hline 00 \quad 00110 \text{ – сумма в прямом модифицированном коде} \end{array}$$

Переполнения разрядной сетки не возникает. Перенос из старшего знакового разряда отбрасывается.

При сложении чисел **00 10110** и **00 11011**

$$\begin{array}{r}
 00\ 10110 \\
 + \\
 00\ \underline{11011} \\
 01\ 10001
 \end{array}$$

в знаковых разрядах результата суммирования возникает комбинация 01, что свидетельствует о переполнении разрядной сетки и ошибочности зафиксированного результата. Возникновение ошибки связано с тем, что при суммировании положительных чисел перенос из старшего разряда оказался зафиксированным во втором из знаковых разрядов. Для регистрации результата суммирования в данном примере требуется шесть разрядов (кроме знаковых). При суммировании отрицательных чисел также возможно переполнение разрядной сетки:

$$\begin{array}{r}
 010011 \\
 + \\
 \underline{11\ 100011} \\
 10\ 110110
 \end{array}$$

В этом случае комбинация **10** в знаковых разрядах указывает на переполнение разрядной сетки.

*Умножение двоичных многоразрядных чисел* включает в себя операции – определение знака произведения и определения его абсолютной величины. Знаковый разряд может быть получен суммированием цифр знаковых разрядов сомножителей без формирования переноса:

$$\begin{array}{l}
 0 + 0 = 0, \\
 0 + 1 = 1, \\
 1 + 0 = 1, \\
 1 + 1 = 0 \text{ без формирования переноса.}
 \end{array}$$

При несовпадении цифр получается 1, что соответствует знаку произведения двух сомножителей с разными знаками.

Абсолютная величина значения произведения определяется путем перемножения чисел без учета их знаков. Перемножение многоразрядных двоичных чисел производится на основе таблицы двоичного умножения

$$\begin{array}{l}
 0 \times 0 = 0, \\
 0 \times 1 = 0, \\
 1 \times 0 = 0, \\
 1 \times 1 = 1.
 \end{array}$$

При умножении двух двоичных чисел множимое последовательно умножается на каждую цифру множителя, начиная либо с младшей, либо со старшей, и

для учета веса соответствующей цифры множителя сдвигается либо влево, если умножение производится, начиная с младшего разряда множителя, либо вправо, если умножение производится, начиная со старшего разряда множителя, на такое число разрядов, на которое соответствующий разряд множителя сдвинут относительно младшего или старшего разряда.

Получающиеся в результате умножения и сдвига частичные произведения после суммирования дают полное произведение.

*Пример 1:*

$$\begin{array}{r}
 111 \text{ – множимое} \\
 \times 101 \text{ – множитель} \\
 \hline
 111 \text{ – первое частичное произведение} \\
 000 \text{ – второе частичное произведение} \\
 111 \text{ – третье частичное произведение} \\
 \hline
 100011 \text{ – произведение.}
 \end{array}$$

При суммировании в разрядах 2, 3, 4 возникло переполнение. Причем переполнение возникло и в старшем разряде, поэтому записываем 1 впереди полученного числа, и получаем: 100011.

*Пример 2:*

$$\begin{array}{r}
 10111 \text{ – множимое} \\
 \times 1101 \text{ – множитель} \\
 \hline
 10111 \text{ – первое частичное произведение} \\
 00000 \text{ – второе частичное произведение} \\
 10111 \text{ – третье частичное произведение} \\
 10111 \text{ – четвертое частичное произведение} \\
 \hline
 100101011 \text{ – произведение}
 \end{array}$$

Тот же результат можно получить при умножении, начиная со старших разрядов множителя:

$$\begin{array}{r}
 10111 \\
 \times 1101 \\
 \hline
 10111 \\
 10111 \\
 00000 \\
 10111 \\
 \hline
 100101011
 \end{array}$$

В цифровых устройствах процессу суммирования частичных произведений придают последовательный характер: формируется одно из частичных произведений, к нему с соответствующим сдвигом прибавляется следующее частичное произведение, к полученной сумме прибавляется с соответствующим сдви-

гом очередное частичное произведение, и т. д., пока не окажутся просуммированными все частичные произведения и не будет получено полное произведение.

Пример умножения чисел этим методом:

$$\begin{array}{r}
 10111 \text{ – четвертое частичное произведение} \\
 101110 \text{ – сдвиг на разряд влево} \\
 \underline{10111} \text{ – третье частичное произведение} \\
 1000101 \text{ – прибавление третьего частичного произведения} \\
 10001010 \text{ – сдвиг на разряд влево} \\
 \underline{00000} \text{ – второе частичное произведение} \\
 10001010 \text{ – прибавление второго частичного произведения} \\
 100010100 \text{ – сдвиг на разряд влево} \\
 \underline{10111} \text{ – первое частичное произведение} \\
 100101011 \text{ – прибавление первого частичного произведения}
 \end{array}$$

При таком методе все частичные произведения суммируются с требуемыми сдвигами друг относительно друга, благодаря чему образуется ранее приведенный результат умножения этих чисел.

При умножении дробных чисел меньше единицы умножение удобнее начинать с младшего разряда множителя. Так, при перемножении дробных чисел  $0,10111$  и  $0,1101$  получим

$$\begin{array}{r}
 0,10111: \quad \text{– первое частичное произведение} \\
 0,01011:1 \quad \text{– сдвиг на разряд вправо} \\
 + \underline{00000:} \quad \text{– второе частичное произведение} \\
 0,01011:1 \quad \text{– прибавление второго частичного произведения} \\
 0,00101:11 \quad \text{– сдвиг на разряд вправо} \\
 + \underline{10111:} \quad \text{– третье частичное произведение} \\
 0,11100:11 \quad \text{– прибавление третьего частичного произведения} \\
 0,01110:011 \quad \text{– сдвиг на разряд вправо} \\
 + \underline{10111:} \quad \text{– четвертое частичное произведение} \\
 1,00101:011 \quad \text{– прибавление четвертого частичного произведения} \\
 0,10010:1011 \text{– сдвиг на разряд вправо}
 \end{array}$$

Если требуется сохранить все разряды в произведении, то в разрядной сетке устройства должно быть предусмотрено число разрядов, равное сумме числа разрядов множимого и множителя. Однако при умножении дробных чисел часто в произведении требуется иметь то же число разрядов, что и в множимом. В таком приближенном представлении результата не фиксируются цифры разрядов при сдвигах, выдвигаемые правее вертикальной штриховой линии, показанной в произведенном выше примере. таким образом, цифры младших разрядов окажутся потерянными и будет получен приближенный результат  $0,100101$ . Далее отбрасывается последний из разрядов, и если этот разряд содержит 1, то



1 прибавляется к следующему разряду для округления результата. Следовательно, полученный результат 0,10011.

Если множимое или множитель, или оба вместе содержат и целую и дробную часть, то запятые в множимом и множителе не учитываются, они умножаются как два целых числа и от полученного произведения справа отделяются запятой  $m + n$  разрядов, где  $n$  – число дробных разрядов множимого, а  $m$  – число дробных разрядов множителя.

*Деление двоичных чисел* включает в себя две операции – определение знака частного и определение его абсолютной величины.

Знаковый разряд частного может быть получен, как и знаковый разряд произведения, суммированием цифр знаковых разрядов делимого и делителя без формирования переноса. Абсолютная величина частного определяется делением чисел без учета их знаков.

Деление начинается с того, что от делимого слева отделяется группа разрядов, причем количество разрядов в этой группе должно либо равняться количеству разрядов в делителе, либо быть на один разряд больше. Если отделение такой группы возможно, в старший разряд частного записывается 1, в противном случае в разряд единиц частного записывается 0. Если выясилось, что частное содержит целую часть, то образуется новая группа разрядов путем вычитания из выделенной группы делителя и приписывания к разности очередной цифры делимого. Если в результате получилось число, превышающее делитель, то в частное записывается 1, в противном случае следующая цифра будет равна 0.

В дальнейшем выполняется ряд одинаковых циклов. Если последняя цифра частного была равна 1, то новая группа образуется вычитанием делителя из предыдущей группы и приписыванием очередной цифры делимого. Если последняя цифра частного 0, то для образования новой группы достаточно приписать к предыдущей группе очередную цифру делимого. Последняя цифра целой части частного получается тогда, когда после определения очередной цифры частного 1 или 0 в делимом не останется больше цифр для того, чтобы приписывать их к разности между предыдущей группой и делителем или к самой предыдущей группе. После этого начинается выделение дробных членов частного. Оно отличается от вычисления целых членов только тем, что вместо очередных цифр делимого к предыдущим группам приписываются нули.

Рассмотрим примеры, в которых делимое больше и меньше делителя:

$$\begin{array}{r}
 110010 \quad |1010 \underline{\hspace{1cm}} \\
 - 1010 \quad 101 \\
 \hline
 101 \\
 1010 \\
 - 1010 \\
 \hline
 \text{Остаток} \quad 0
 \end{array}$$

$$\begin{array}{r}
 11010111 \quad |10110 \underline{\hspace{1cm}} \\
 - 10110 \quad 1001, 110 \\
 \hline
 100111 \\
 - 10110 \\
 \hline
 100010 \\
 - 10110 \\
 \hline
 11000 \\
 - 10110 \\
 \hline
 \text{Остаток} \quad 1000
 \end{array}$$

$$\begin{array}{r}
 1011001 \quad |11011010 \\
 10 \quad 1100100 \quad 0,0110 \\
 - 11011010 \\
 \hline
 100010100 \\
 - 11011010 \\
 \hline
 11101000 \\
 - 11011010 \\
 \hline
 \text{Остаток} \quad 111000
 \end{array}$$

В цифровых устройствах при выполнении операции деления так же, как и при выполнении операции алгебраического сложения, используется дополнительный и модифицированный коды. Например, при делении числа 0,11011 на 0,11101 представляем делитель в дополнительном коде 00011:

$  \begin{array}{r}  00 \ 11011 \\  + 11 \ 00011 \\  \hline  11 \ 11110 \\  00 \ 11101 \\  00 \ 11011 \\  11 \ 10010 \\  00 \ 01101 \\  11 \ 11001 \\  00 \ 00110 \\  11 \ 11101 \\  00 \ 00011 \\  11 \ 11111 \\  00 \ 00010  \end{array}  $	$  \begin{array}{r}  11101 \\  0,1111 \\  \uparrow \uparrow \uparrow \uparrow  \end{array}  $	<p><i>исходный делитель 0,11101</i></p> <p><i>исходный делитель 0,11101</i></p> <p><i>сдвинутый делитель 0,01110</i></p> <p><i>сдвинутый делитель 0,00111</i></p> <p><i>сдвинутый делитель 0,00011</i></p> <p><i>сдвинутый делитель 0,00001</i></p>
---	---	---

При вычитании сдвинутые делители представляются в *дополнительном* коде.

## Теория логических функций

### 2.1. Функции алгебры логики. Логические элементы

Алгебра логики – это формальный аппарат описания логической стороны процессов в цифровых устройствах.

Устройства, предназначенные для формирования функции алгебры логики (ФАЛ), будем называть дискретными (цифровыми) или логическими устройствами.

Функция вида  $F(x_1, x_2, \dots, x_n)$  или  $f(x_1, x_2, \dots, x_n)$  называется булевой (логической или переключательной), если она, так же как и ее аргументы (логические переменные  $x_i$ ), может принимать только два значения – 0 или 1.

Логическая переменная – простое высказывание или событие истинность (ложность), которого не зависит от других высказываний (событий).

Логическая функция (ЛФ) – сложное высказывание или событие истинность (ложность), которого зависит от других простых высказываний (событий).

Логические переменные хорошо описывают состояния устройств, которые могут находиться в двух четко различных положениях: включено-выключено. Такими устройствами являются реле, тумблеры, кнопки, а также полупроводниковые логические элементы, на выходе которых может присутствовать лишь один из двух четко различимых уровней напряжения.

Информация, которая передается между отдельными узлами (блоками) сложного цифрового устройства, представляется в виде кодовых слов. Выходное слово зависит от того, какие слова поступают на входы узла.

Следовательно, выходное слово есть функция, для которой аргументами являются входные слова.

Если длина кодовых слов составляет  $n$  разрядов ( $n$  – переменных), то можно построить  $2^n$  различных комбинаций. Например, при  $n = 2$  можно построить 8 слов:

000, 001, 010, 011, 100, 101, 110, 111.

Булева алгебра строится на базе трех основных операций, которые выражают логические функции НЕ, ИЛИ, И, которые получили название *базис*.

Базисом называют полную систему функций алгебры логики.

Минимальный базис (полный) содержит функции; НЕ, ИЛИ, И (инверсию, дизъюнкцию, конъюнкцию).

Под минимальным базисом понимают такой набор функций, исключение из которого любой функции превращает полную систему функций в неполную.

В соответствии с перечнем логических операций различают три основных логических элемента (ЛЭ): **И**, **ИЛИ**, **НЕ**.

**Функция НЕ – логическое отрицание**, определяется и обозначается

$$F(x) = \bar{x}.$$

Отрицание (инверсия) обозначается чертой, которая ставится над переменной. Логический элемент, выполняющий операцию НЕ, называется инвертором, на рис. 2.1 представлено условное графическое обозначение (УГО) логического элемента инвертор.

### Элемент – НЕ

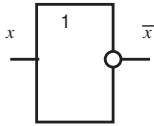


Рис. 2.1 – УГО логического элемента инвертор

Инвертор может иметь только один вход.

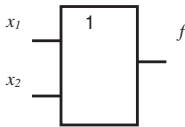
**Функция ИЛИ – логическое сложение**, дизъюнкция – обозначается с помощью знака «V» или «+», и эта операция записывается в виде

$$F(x) = x_1 + x_2 + \dots + x_n,$$

которая читается «F есть (равно)  $x_1$  или  $x_2$ , или ...  $x_n$ », т.е. знак «V» или «+» читается как «или».

Логический элемент, выполняющий операцию ИЛИ, называется дизъюнктом либо элементом ИЛИ. На рис. 2.2 представлено УГО элемента ИЛИ и его таблица истинности.

### Элемент – ИЛИ



№ набора	$x_1$	$x_2$	$f$
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

Рис. 2.2 – УГО и таблица истинности ЛЭ ИЛИ

Дизъюнктом может иметь несколько входов, как правило в интегральных схемах (ИС) их число 2, 4, 8.

**Функция И – логическое умножение**, конъюнкция – обозначается с помощью знака «&» или «∧» «×», «•». Операция записывается в виде

$$F(x) = x_1 \cdot x_2 \dots x_n$$

и читается « $F$  есть (равно)  $x_1$  и  $x_2$ , и ...  $x_n$ », т.е. знак «&» или « $\times$ », « $\bullet$ » читается как «и». На рис. 2.3 представлено УГО элемента И и его таблица истинности.

### Элемент – И

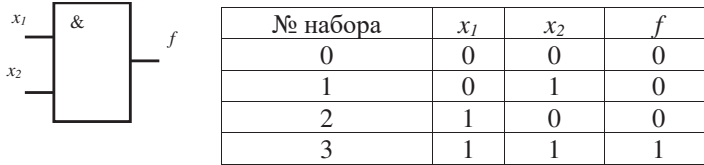


Рис. 2.3 – УГО и таблица истинности ЛЭ И

Конъюнктор может иметь несколько входов, как правило в интегральных схемах (ИС) их число 2, 4, 8.

Основные операции булевой алгебры – инверсия, конъюнкция и дизъюнкция – позволяют построить любую, сколько угодно сложную логическую функцию.

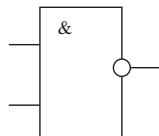
Группу элементов И, ИЛИ, НЕ называют «функционально полными элементами», или элементами, обладающими функциональной полнотой.

Базис И, ИЛИ, НЕ не является минимальным. Из этой совокупности функций можно исключить функцию И либо функцию ИЛИ, и оставшийся набор будет удовлетворять свойствам базиса.

Таким образом, функция Шеффера (И–НЕ) (рис. 2.4, а) образует полный базис, точно так же, как функция Вебба (элемент Пирса) (ИЛИ–НЕ) рис. 2.4, б.

**И–НЕ**  
элемент Шифера

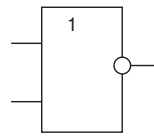
$$f = \overline{x \cdot y} = x \mid y$$



а

**ИЛИ–НЕ**  
элемент Пирса

$$f = \overline{x + y} = x \downarrow y$$



б

Рис. 2.4 – УГО логических элементов: а – И–НЕ; б – ИЛИ–НЕ

В настоящее время базис И, ИЛИ, НЕ обычно используется при начальной стадии проектирования логических устройств. Для реализации устройств обычно используют базисы И–НЕ либо ИЛИ–НЕ. ЛЭ этих базисов широко выпускаются промышленностью в интегральном исполнении.

## 2.2. Формы представления логических функций

При анализе и синтезе цифровых устройств с целью определения логики их функционирования используют следующие способы задания булевых функций:

1. *Словесное описание.*
2. *Табличное описание.*
3. *Аналитическое описание.*
4. *Числовое задание (компактная форма записи ФАЛ).*
5. *Описание ФАЛ в виде карт Карно.*

1. **Словесное описание** используется на начальном этапе синтеза логического устройства.

*Например*, рассмотрим дискретное устройство содержащие: два параллельно соединенных выключателя (разомкнутых), электрическая лампа, источник питания (рис. 2.5).

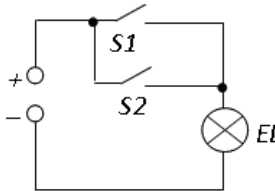
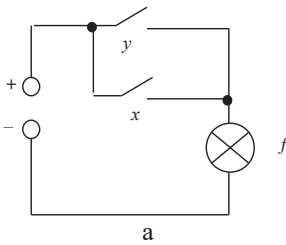


Рис. 2.5 – Схема по словесному описанию ЦУ

2. При **табличном способе** задания функция представляется в виде таблицы, в которую вписываются все возможные наборы значений аргументов в порядке возрастания их номеров на основе двоичной системы счисления и на каждом наборе устанавливается значение функции 0 или 1.

Такие таблицы называют таблицами истинности, или таблицами работы. В качестве примера составим таблицу истинности дискретного устройства, приняв выключатели за переменные «х» и «у», лампочку за функцию  $f$  (рис. 2.6, а).



Набор №	y	x	f
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

б

Рис. 2.6 – Табличное описание ЦУ: а – схема ЦУ; б – таблица истинности

Составим таблицу истинности исходя из условий, если выключатель замкнут, то переменная  $x$  ( $y$ ) равна 1 – разомкнут 0 и соответственно, если лампочка горит то функция  $f=1$ , а если нет –  $f=0$  рис. 2.6, б.

Из таблицы истинности (рис.2.6,б) видно, что для двух переменных существует 4 набора (0, 1, 2, 3). В общем случае количество наборов будет определяться величиной  $2^n$ , где  $n$  – количество переменных.

3. **Аналитическое описание.** От табличной формы записи легко перейти к аналитической форме в виде алгебраического представления функции.

В такой форме удобно проводить различные преобразования, которые делаются с целью минимизации (упрощения) логического выражения при синтезе (разработке) дискретного устройства.

В алгебраическом виде булевы функции представляются в двух совершенно нормальных формах – дизъюнктивной или конъюнктивной.

Термин дизъюнкция обозначает логическое сложение, а термин конъюнкция логическое умножение, более подробно данные функции будут рассмотрены ниже.

Совершенная дизъюнктивная нормальная форма (СДНФ) представляет собой логическую сумму (дизъюнкцию), конъюнкт единицы из тех наборов, на которых функция принимает значение 1.

Конъюнкцией единицы называется логическое произведение (конъюнкция) переменных на каждом единичном наборе, при этом если переменная принимает значение 0, то она берется с инверсией, если 1 – без инверсии.

Совершенная конъюнктивная нормальная форма (СКНФ) представляет собой логическую произведение (конъюнкцию), конъюнкт нуля из тех наборов, на которых функция принимает значение 0.

Конъюнкцией нуля называется логическая сумма (дизъюнкция) переменных на каждом нулевом наборе, при этом если переменная принимает значение 1, то она берется с инверсией, если 0 – без инверсии.

**На практике чаще используется совершенная дизъюнктивная нормальная форма – СДНФ.**

Например, запишем функцию  $f$  в СДНФ, используя таблицу истинности (рис. 2.6, б). Для записи выбираются наборы, где функция  $f = 1$ , т.е. №1, 2, 3.

При это в данных наборах нулевое значение переменной  $x$  или  $y$  записывают с инверсией. В следствие такого подхода набор №1 будет иметь вид  $\bar{y}x$ , а вся функция запишется, как

$$f_{\text{СДНФ}} = \bar{y}x + y\bar{x} + ux.$$

4. При числовом способе задания булевых функций каждый набор отмечается номером, выраженным десятичным числом. Так, в соответствии с таблицей истинности числовая запись будет выглядеть следующим образом

Набор №	$y$	$x$	$f$
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

$$f = \sum(1, 2, 3).$$

От числовой формы легко перейти обратно к аналитической записи для этого, сумма записывается в двоичном коде  $F = \sum(01, 10, 11) = 01+10+11$  и

представляется СДНФ  $f_{\text{СДНФ}} = \bar{y}x + y\bar{x} + ux.$

5. Карты Карно. Данная форма записи, как и аналитическая запись, используется, для минимизации функции в процессе синтеза.

На картах Карно (рис. 2.7) конституента изображается в виде клетки. На рис. 2.7 представлены карты Карно для двух (а), трех (б) и четырех (в) переменных.

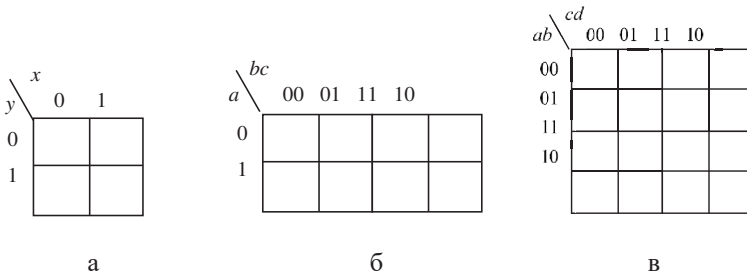


Рис. 2.7 – Карты Карно для двух (а), трех (б) и четырех (в) переменных

Символы, соответствующие значениям переменных, располагаются в порядке последовательности чисел циклического кода.



Наименование переменных написано по обе стороны диагональной черты в левом верхнем углу карты. Значения переменных обозначаются с внешней стороны карты посредством двоичных цифр: 0 соответствует инверсному значению переменной, а 1 – прямому. Данный шифр называют кодом Грея.

Чтобы нанести на карту конститuentу  $\bar{y}x$ , следует поставить **1** в клетку карты значения переменных  $01$  (рис. 2.8).

В соответствие с данным положением аналитическая запись функции в виде карты Карно будет иметь вид

$$f_{\text{СДНФ}} = \bar{y}x + y\bar{x} + yx.$$

	$x$	
	0	1
$y$		
0		<b>1</b>
1	<b>1</b>	<b>1</b>

Рис. 2.8 – Заполнение карты Карно для двух переменных

Принцип нанесения определенной функции на карту Карно рассмотрим на примерах.

*Пример 1.* Нанести на карту Карно функцию, заданную аналитически.

$$f(a, b, c) = 011 + 101 + 110 + 111,$$

$$f(a, b, c) = \bar{a}bc + a\bar{b}c + a\bar{b}\bar{c} + abc.$$

Чтобы нанести на карту конститuentу  $\bar{a}bc$ , следует поставить **1** в клетку карты значения переменных  $011$ .

Этому соответствуют клетки 0 по вертикали и 11 по горизонтали. Аналогично наносятся остальные конститuentы единицы. В результате получим карту, изображенную на рис. 2.9.

	$bc$			
	00	01	11	10
$a$				
0			<b>1</b>	
1		<b>1</b>	<b>1</b>	<b>1</b>

	$bc$			
	$\bar{b}\bar{c}$	$\bar{b}c$	$bc$	$b\bar{c}$
$a$				
$\bar{a}$			<b>1</b>	
$a$		<b>1</b>	<b>1</b>	<b>1</b>

Рис. 2.9 – Заполнение карты Карно для трех переменных

*Пример 2.* Нанести на карту Карно функцию четырех переменных, заданную аналитически.

$$f(a,b,c,d) = \bar{a}\bar{b}cd + \bar{a}b\bar{c}d + \bar{a}bc\bar{d} + \bar{a}bcd + ab\bar{c}\bar{d} + abc\bar{d}.$$

В результате преобразований карта Карно для четырех переменных будет иметь вид (рис. 2.10).

	cd			
	00	01	11	10
00			1	
01	1	1		
11	1	1		
10			1	

Рис. 2.10 – Заполнение карты Карно для четырех переменных

### 2.3. Законы алгебры логики

Применение булевой алгебры для математического исследования (анализа и синтеза) цифровых устройств основывается на системе аксиом, теорем, выражающих основные законы алгебры логики.

Законы алгебры логики для одной переменной называют тождественными, или равносильными, которые записываются следующим образом:

$$x + 0 = 0,$$

$$x + 1 = x,$$

$$x + x = x,$$

$$x + \bar{x} = 1,$$

$$\bar{\bar{x}} = x,$$

$$x \cdot 0 = 0,$$

$$x \cdot 1 = x,$$

$$x \cdot x = x,$$

$$x \cdot \bar{x} = 0,$$

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

#### Законы алгебры логики для двух и более переменных

1. Переместительный (коммутативный) закон:

$$x + y = y + x; \quad x \cdot y = y \cdot x.$$

2. Сочетательный (ассоциативный) закон:

$$x + y + z = x + (y + z); \quad x \cdot y \cdot z = x \cdot (y \cdot z).$$

3. Распределительный (дистрибутивный) закон:

$$x \cdot (y + z) = x \cdot y + x \cdot z; \quad x + y \cdot z = (x + y) \cdot (x + z).$$

4. Закон поглощения:

$$\begin{aligned} x + x \cdot y &= x; & (x + \bar{y}) \cdot y &= x \cdot y; \\ x \cdot (x + y) &= x; & x \cdot \bar{y} + y &= x + y. \end{aligned}$$

5. Закон склеивания (исключения):

$$x \cdot y + x \cdot \bar{y} = x; \quad (x + y) \cdot (x + \bar{y}) = x.$$

6. Закон инверсии (закон де Моргана):

$$\overline{x + y} = \bar{x} \cdot \bar{y}; \quad \overline{x \cdot y} = \bar{x} + \bar{y}.$$

или

$$x + y = \overline{\bar{x} \cdot \bar{y}}; \quad x \cdot y = \overline{\bar{x} + \bar{y}}.$$

Законы поглощения, склеивания и инверсии позволяют выполнить алгебраические преобразования в сложных булевых выражениях с целью приведения их к виду, удобному для реализации на основе логических элементов.

*Например,* дана функция в СДНФ следующего вида

$$f(a, b, c) = \bar{a}bc + a\bar{b}c + a\bar{b}\bar{c} + abc.$$

Применив закон склеивания

$$x \cdot y + x \cdot \bar{y} = x; \quad (x + y) \cdot (x + \bar{y}) = x.$$

функция может быть упрощена

$$f(a, b, c) = bc(\bar{a} + a) + ac(\bar{b} + b) + ab(\bar{c} + c) = bc + ac + ab.$$

Закон де Моргана, используется для перевода ФАЛ из базиса И, ИЛИ, НЕ в базис И-НЕ или в базис ИЛИ-НЕ.

## 2.4. Минимизация логических функций

При проектировании аппаратуры стремятся к реализации структурных схем, обеспечивающих минимальный расход оборудования, минимальную стоимость при условии обеспечения заданного уровня надежности, что достигается за счет минимизации.

Под минимизацией булевой функции чаще всего понимают нахождение наиболее простого ее представления с минимальным числом входящих в нее переменных.

Однако при проектировании аппаратуры с применением БИС и СБИС требование уменьшения числа корпусов ИС и их межсоединений по-прежнему остается весьма важным.

Требование уменьшения числа элементарных логических элементов (ЛЭ), входящих в разрабатываемое устройство, в настоящее время также не потеряло своей актуальности. Объясняется это все более широким использованием при проектировании электронных средств программируемых логических СБИС широкого применения и ползуказных СБИС на основе базовых матричных кристаллов. Эти СБИС и БИС, как правило, содержат отдельные некоммутированные между собой элементарные ЛЭ, например 2И–НЕ или 2ИЛИ–НЕ, или просто наборы транзисторов, резисторов и диодов, которые могут быть соединены между собой в соответствии с заданным алгоритмом обработки логических сигналов. Поскольку число элементов в одной СБИС задано из технологических соображений, то минимизация ФАЛ по критерию уменьшения числа используемых элементов позволяет на одном кристалле решать более сложные задачи логической обработки сигналов, т. е. в конечном счете уменьшать число требуемых ИС и связей между ними. Это снижает стоимость и повышает надежность электронной аппаратуры.

Рассмотрим ряд методов, позволяющих провести минимизацию ФАЛ по критерию уменьшения числа элементарных ЛЭ.

После минимизации возможно выполнение других преобразований функций, имеющих целью сокращение числа типов элементов, приведение функции к виду, удобному для реализации на заданных элементах, например, на элементах И–НЕ или ИЛИ–НЕ.

При минимизации логических функций наиболее часто пользуются алгебраическим методом и картографическим методом с помощью карт Вейча и Карно.

При алгебраическом методе минимизации выполняются следующие операции:

- булева функция представляется в СДНФ;
- проводится склеивание конституент единицы;
- отыскивают тупиковую форму, т.е. форму при которой дальнейшая минимизация не возможна.

Затем приводят функцию к базису И–НЕ или ИЛИ–НЕ на основании закона де Моргана.

*Пример 1.* Найти тупиковую форму функции, определенной табл. 2.1.

Таблица. 2.1

№ набора	$x_1$	$x_2$	$f$
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

1. Представляем функцию в СДНФ

$$F_{\text{СДНФ}} = \bar{x}_1 \bar{x}_2 + \bar{x}_1 x_2 + x_1 x_2. \quad (2.1)$$

2. Проводим склеивание конституент единицы, используя закон склеивания, по принципу «каждого с каждым» и получаем

$$F = x_2 (\bar{x}_1 + x_1) + x_1 (\bar{x}_2 + x_2) = x_1 + x_2. \quad (2.2)$$

Эта функция (2.2) является тупиковой. Построим логическую схему выражения (2.1) используя базис НЕ, ИЛИ, И. При построении будем использовать правило, что в схеме сначала записывают переменные (для инверсии используют схему НЕ), затем конъюнкторы (схема И) и потом уже дизъюнкторы (схема ИЛИ). Построенная по этому правилу комбинационная схема для ФАЛ выражение (2.1) приведена на рис. 2.11.

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

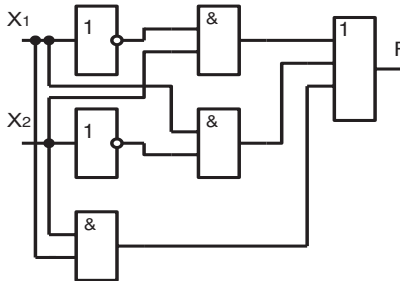


Рис. 2.11 – Комбинационная схема для ФАЛ выражение (2.1)

После склеивания было получено выражение (2.2), т.е. элемент ИЛИ (рис. 2.12).

$$F = x_1 + x_2.$$

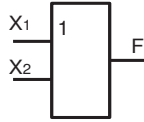


Рис. 2.12 – Логический элемент ИЛИ

Приведем функцию (2.2) в базис функции И–НЕ, для чего подвергнем двойному отрицанию (закон де Моргана  $\overline{\overline{x + y}} = \overline{\overline{x} \cdot \overline{y}}$ ) правую и левую части

$$F = x_1 + x_2 = \overline{\overline{x_1} \cdot \overline{x_2}}. \quad (2.3)$$

Логическая схема функции (2.3) в базисе И–НЕ представлена на рис. 2.13.

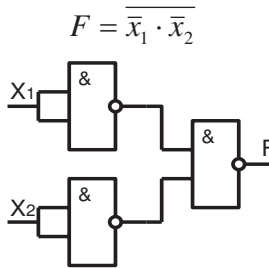


Рис. 2.13 – Логический элемент ИЛИ в базисе И-НЕ

*Пример 2.* Найти тупиковые формы функции, заданной числовым способом

$$F(a, b, c) = \Sigma(3, 5, 6, 7).$$

Переведем десятичные цифры наборов, на которых функция принимает единичное значение, в двоичный код и получим

$$F(a, b, c) = \Sigma(011, 101, 110, 111).$$

Представим эту функцию в СДНФ

$$F(a, b, c) = \overline{a}bc + a\overline{b}c + a\overline{b}\overline{c} + abc. \quad (2.4)$$

Применим закон склеивания

$$F(a, b, c) = bc(\overline{a} + a) + ac(\overline{b} + b) + ab(\overline{c} + c) = bc + ac + ab. \quad (2.5)$$

Логическая схема для выражения (2.5) будет иметь вид рис. 2.14.

$$F(a,b,c) = bc + ac + ab$$

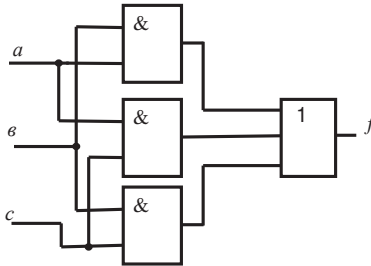


Рис. 2.14 – Логическая схема для выражения (2.5)

Если вынести одну из переменных за скобки, можно получить несколько тупиковых форм функции:

$$F(a,b,c) = a(c+b) + bc; \quad (2.6)$$

$$F(a,b,c) = b(a+c) + ac; \quad (2.7)$$

$$F(a,b,c) = c(a+b) + ab. \quad (2.8)$$

При построении схемы любой из трех равнозначных функций (2.6), (2.7), 2.8) сначала реализуют функцию находящуюся в скобках. Построим логическую схему для функции (2.7) (рис. 2.15).

$$F(a,b,c) = a(c+b) + bc$$

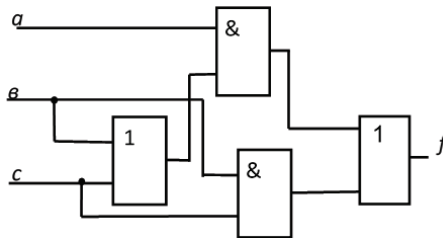


Рис. 2.15 – Логическая схема для функции (2.7)

### Минимизация логических функций методом карт Карно

Существуют методы, позволяющие упростить поиск тупиковых форм булевых функций – метод карт Карно или диаграмм Вейча. Процессы минимизации функций на этих картах совершенно одинаковы. Для рассмотрения минимизации

ции с помощью карт Карно используем ФАЛ из трех переменных, заданную аналитически записанную СДНФ.

$$f(a, b, c) = 011 + 101 + 110 + 111,$$

$$f(a, b, c) = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc.$$

Чтобы нанести на карту конститuentу  $\bar{a}bc$ , следует поставить 1 в клетку карты значения переменных 011. Этому соответствуют клетки 0 по вертикали и 11 по горизонтали. Аналогично наносятся остальные конститuenty единицы. В результате получим карту, изображенную на рис. 2.16.

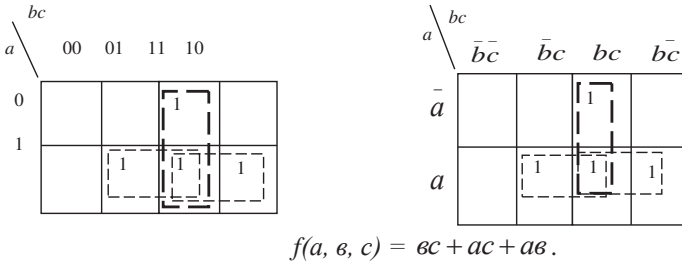


Рис. 2.16 – Минимизация логических функций методом карт Карно

Конститuenty, стоящи рядом и обведенные пунктирными линиями подвержены склеиванию. Если переменная в пределах склеивания меняется (на инверсную или на оборот), то она исключается.

*Например*, рассмотрим склеивание единиц обведенных жирным пунктиром. Склеивание производим по вертикали. В это случае переменные  $bc$  не изменяются, а переменная  $a$  изменяется на инверсную, следовательно, она исключается и в результате получаем  $bc$ . В данном примере склеиванию подвержено три пары единиц. После склеивания тупиковая функция будет иметь вид

$$f(a, b, c) = bc + ac + ab. \quad (2.9)$$

В результате получена тупиковая функция (2.9), которая идентична функции полученной ранее аналитическим путем. Логическая комбинационная схема для выражения (2.9) будет иметь вид (рис. 2.17).

$$f(a, b, c) = bc + ac + ab$$



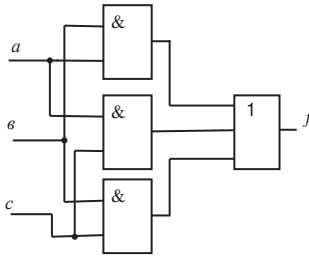


Рис. 2.17 – Логическая комбинационная схема для выражения (2.9)

Рассмотрим минимизацию функции  $f(a, e, c, d)$ , (рис. 2.18)

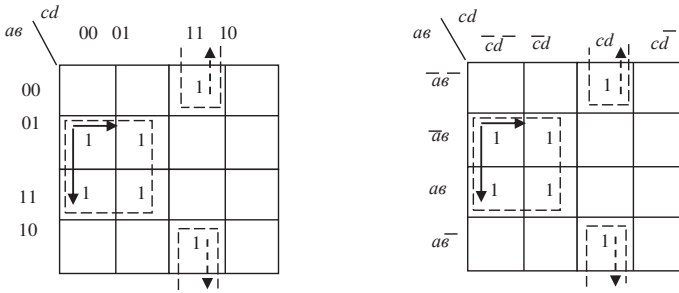


Рис. 2.18 – Минимизация ЛФ четырех переменных

При минимизации с помощью карт Карно склеиванию подлежат рядом стоящие 2, 4, 8 и т. д. единиц. Кроме это склеиваются единицы, располагающиеся на концах столбцов и строк. На рис. 2.18 показано пунктирными стрелками склеивание единиц находящихся на конце столбца, для этого случая получаем  $\bar{v}cd$ . При минимизации четырех рядом стоящих единиц анализ производится по горизонтали и вертикали (или наоборот) см. рис. 2.18. При этом не имеет значение направление анализа (склеивания). В результате склеивания четырех единиц (рис. 2.18) получаем  $\bar{v}\bar{c}$ . В результате склеивания получим тупиковую функцию вида

$$f(a, e, c, d) = \bar{v}cd + \bar{v}\bar{c}. \quad (2.10)$$

На рис. 2.19. представлены варианты склеивания восьми единиц

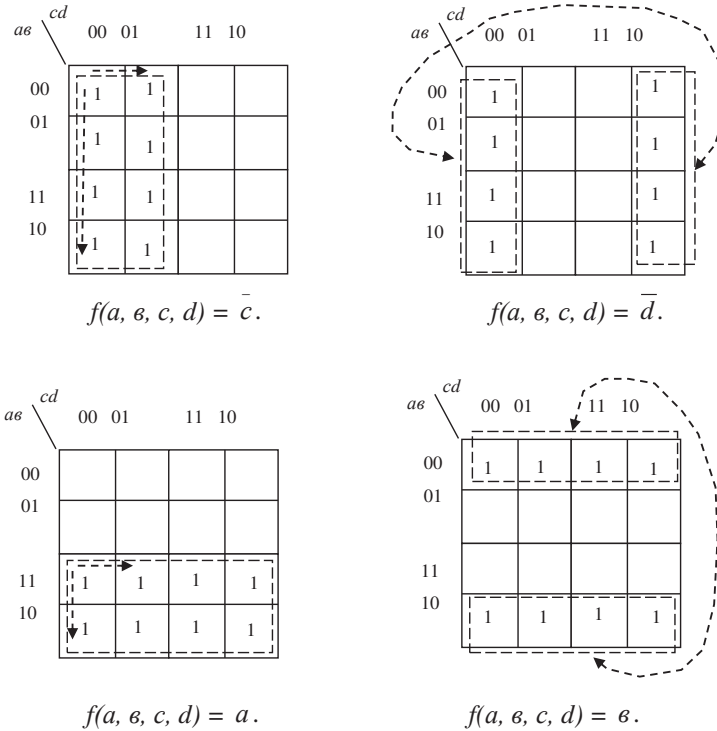


Рис. 2.19 – Варианты склеивания восьми единиц

## 2.5. Анализ и синтез комбинационных схем

Рассмотрим комбинационное устройство, имеющего четыре входа ( $x_1, x_2, x_3, x_4$ ) и один выход  $f$ , заданы таблицей истинности (табл. 2.2).

Требуется синтезировать функциональную логическую схему устройства в базисе И-НЕ (ИЛИ-НЕ), применяя методы минимизации заданной логической функции с помощью алгебраических преобразований и с использованием карт Карно.

### 1. Минимизация алгебраическим методом

Для минимизации ФАЛ запишем ее в СДНФ и проведем склеивание, используя законы алгебры логики (члены функции подлежащие склеиванию показаны пунктиром).

$$\begin{aligned}
 f(x_1, x_2, x_3, x_4) = & \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 + \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 + \\
 & + \bar{x}_1 \bar{x}_2 x_3 x_4 + \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 + \bar{x}_1 x_2 \bar{x}_3 x_4 + \bar{x}_1 x_2 x_3 \bar{x}_4 + \\
 & + \bar{x}_1 x_2 x_3 x_4
 \end{aligned}$$

В результате склеивания получаем тупиковую функцию.

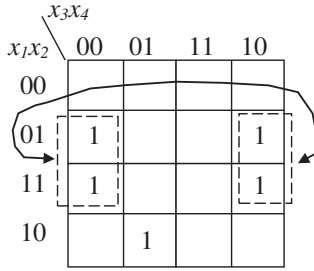
$$\begin{aligned}
 f(x_1, x_2, x_3, x_4) &= x_2 x_3 \bar{x}_4 (\bar{x}_1 + x_1) + x_2 \bar{x}_3 \bar{x}_4 (x_4 + \bar{x}_4) + \\
 &+ x_1 x_2 \bar{x}_3 x_4 = x_2 x_3 x_4 + x_2 \bar{x}_3 \bar{x}_4 + x_1 x_2 \bar{x}_3 x_4 = \\
 &= x_2 x_4 (x_3 + \bar{x}_3) + x_1 x_2 \bar{x}_3 x_4 = x_2 x_4 + x_1 x_2 \bar{x}_3 x_4.
 \end{aligned}
 \tag{2.11}$$

Таблица 2.2

№	$x_1$	$x_2$	$x_3$	$x_4$	$f$
0.	0	0	0	0	0
1.	1	0	0	0	0
2.	0	1	0	0	1
3.	1	1	0	0	1
4.	0	0	1	0	0
5.	1	0	1	0	0
6.	0	1	1	0	1
7.	1	1	1	0	1
8.	0	0	0	1	0
9.	1	0	0	1	1
10.	0	1	0	1	0
11.	1	1	0	1	0
12.	0	0	1	1	0
13.	1	0	1	1	0
14.	0	1	1	1	0
15.	1	1	1	1	0

## 2. Минимизация методом карт Карно

Для минимизации заполним карту Карно, используя таблицу истинности (табл. 2.2) и проведем склеивание 4-х единиц. В результате получаем



$$f(x_1, x_2, x_3, x_4) = x_2 \bar{x}_4 + x_1 \bar{x}_2 \bar{x}_3 x_4. \quad (2.12)$$

Полученное выражение (2.12) идентично выражению (2.11) полученному в ходе минимизации алгебраическим методом, что подтверждает правильность преобразований. Для перевода полученной в ходе минимизации ФАЛ в базис И-НЕ (ИЛИ-НЕ) воспользуемся законом де Моргана.

$$x + y = \overline{\bar{x} \cdot \bar{y}}; \quad x \cdot y = \overline{\bar{x} + \bar{y}}.$$

Используя закон де Моргана запишем ФАЛ выражение для базиса И-НЕ.

$$f(x_1, x_2, x_3, x_4) = \overline{\overline{(x_2 \bar{x}_4)} \cdot \overline{(x_1 \bar{x}_2 \bar{x}_3 x_4)}}.$$

### 3. Построение комбинационных схем (базис И-НЕ)

1. Запишем входные переменные в прямом и инверсном виде;
2. Проведем логическое умножение переменных в скобках;
3. Проведем логическое умножение скобок.

В результате получаем комбинационное устройство рис. 2.20.

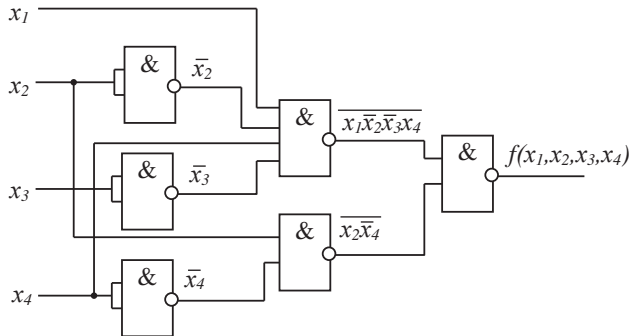


Рис. 2.20 – Комбинационное устройство в базисе И-НЕ для табл. 2.2

### 3. Построение комбинационных схем (базис ИЛИ-НЕ)

Аналогично строим комбинационную схему в базисе ИЛИ-НЕ. В этом случае ФАЛ будет иметь вид

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

Схема комбинационного устройства в базисе ИЛИ-НЕ строится по аналогичному алгоритму и представлена на рис. 2.21

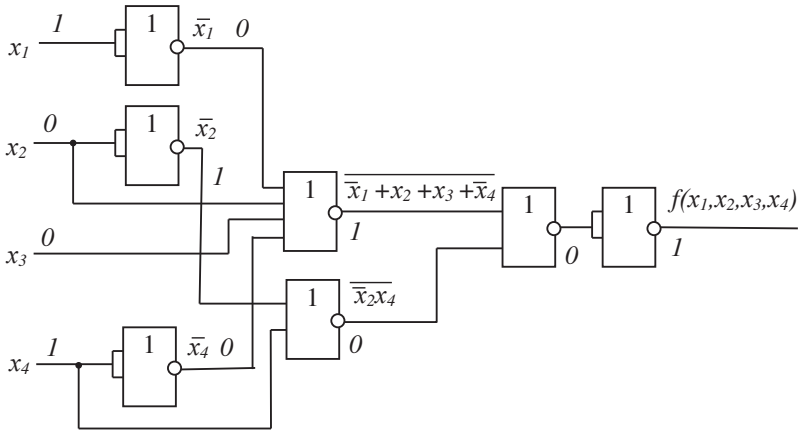


Рис. 2.21 – Схема комбинационного устройства в базисе ИЛИ-НЕ

Для проверки правильности построенных схем необходимо из табл. 1 дать на вход схемы любую строку переменных  $(x_1, x_2, x_3, x_4)$  и проверить значение функции  $f$ , на выходе комбинационного устройства.

Например, из табл. 2.2 возьмем значение переменных  $x_1, x_2, x_3, x_4$  из 9-ой строки и подадим на вход схемы (рис. 2.21), в результате на выходе получим функцию  $f=1$ , а это значит, что комбинационное устройство синтезировано правильно. Аналогичную проверку можно провести и для комбинационного устройства представленного на рис. 1.

#### Анализ комбинационного устройства

Анализ комбинационного устройства это обратная операция синтезу и заключается в построении таблицы истинности, т. е. в анализе работы комбинационного устройства.

Например, дана схема комбинационного устройства (рис. 2.22) проанализировать ее работу и построить таблицу истинности (табл. 2.3).

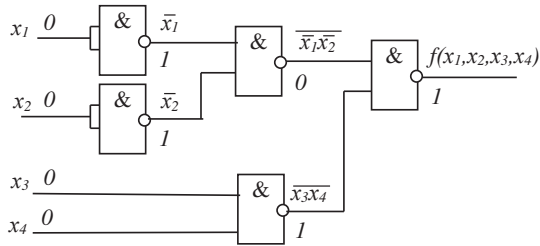


Рис. 2.22 – Схема комбинационного устройства

1. Проводим анализ функционирования каждого элемента схемы комбинационного устройства и записываем ФАЛ, которая будет иметь вид

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

2. Далее начиная с первой строки, заполняем таблицу истинности (табл. 2.3), путем анализа работы каждого элемента схемы.

Таблица 2.3

№	$x_1$	$x_2$	$x_3$	$x_4$	$f$
0.	0	0	0	0	1
1.	1	0	0	0	0
2.	0	1	0	0	0
3.	1	1	0	0	0
4.	0	0	1	0	1
5.	1	0	1	0	0
6.	0	1	1	0	0
7.	1	1	1	0	0
8.	0	0	0	1	1
9.	1	0	0	1	0
10.	0	1	0	1	0
11.	1	1	0	1	0
12.	0	0	1	1	1
13.	1	0	1	1	1
14.	0	1	1	1	1
15.	1	1	1	1	1

## Логические элементы

### 3.1. Общие сведения об ИМС логических элементов

В настоящее время каждая ИМС относится к тому или иному *логическому семейству*, именуемому *серией*. Серией микросхем называют группу микросхем, выполненных по одинаковой или близкой технологии, имеющих сходные технические характеристики и предназначенных для совместной работы в составе цифровой аппаратуры. Именно технология изготовления определяет такие важнейшие характеристики микросхемы, как напряжение питания, рассеиваемая мощность, скорость переключения и помехоустойчивость.

В настоящее время при разработке логических схем наибольшее распространение получили следующие базовые логические элементы (БЛЭ), следующих серий:

1. *Транзисторно-транзисторная логика (ТТЛ) и ТТЛ с диодами Шотки (ТТЛШ);*
2. *Эмиттерно-связанная логика (ЭСЛ);*
3. *Интегрально-инжекционная логика (И<sup>2</sup>Л);*
4. *Логика на комбинированных полевых транзисторах (КМОП).*

Первые три типа используют биполярные транзисторы, последний – полевые.

Распространение нескольких типов логики, выполняющих одни и те же логические функции, объясняется различием их основных характеристик, что в зависимости от технических требований и условий эксплуатации позволяет строить электронные средства с требуемыми параметрами.

Самыми распространенными на сегодняшний день являются ИС, реализующие ТТЛ и ее разновидности. Этот тип ИС, судя по прогнозам специалистов, останется наиболее массовым и в ближайшем будущем.

Интегральные схемы данного типа обладают средним, быстродействием ( $F_{\max} = 20 \dots 50$  МГц) и средней потребляемой мощностью.

Интегральные схемы, реализующие ЭСЛ, являются наиболее быстродействующими, но мощность, потребляемая этими элементами, превышает мощность элементов ТТЛ. Однако гибкость ее схемотехнических решений позволяет достаточно просто реализовывать сложные логические функции.

Базовые элементы И<sup>2</sup>Л были разработаны специально для использования в БИС. Их отличает высокая, недоступная для ТТЛ и ЭСЛ, степень интеграции, пониженное напряжение питания, простота согласования с элементами ТТЛ и возможность регулировать в широких пределах быстродействие путем изменения потребляемой мощности.

Особенностью ИС, выполненных на полевых транзисторах, является малая потребляемая мощность. Однако по быстродействию эти элементы пока уступают логике на биполярных транзисторах. Данный тип логики также обладает высокой помехоустойчивостью. В сравнении с биполярными, полевые транзи-

сторы имеют меньшие размеры и проще в изготовлении, что позволяет разместить на единице площади кристалла больше элементов. В соответствии с этими особенностями область их применения ограничивается функционально сложными устройствами невысокого быстродействия и малого токопотребления.

Прогнозы показывают, что по мере совершенствования технологии, направленной на повышение их быстродействия, этот класс ИС постепенно станет самым массовым. В первую очередь это касается ИС КМОП.

Рассмотрим основные параметры интегральных микросхем ЛЭ (цифровых микросхем логических элементов ЦИС).

*Коэффициент объединения по входу* определяет число входов элемента, предназначенных для подачи логических переменных. Элемент с большим коэффициентом объединения по входу имеет более широкие логические возможности.

*Нагрузочная способность* (или *коэффициент разветвления по выходу*) определяет число входов аналогичных элементов, которое может быть подключено к выходу данного элемента. Чем выше нагрузочная способность элементов, тем меньшее число элементов может потребоваться при построении цифрового устройства.

Для повышения нагрузочной способности в элементах ДТЛ и ТТЛ применяют усложненную схему инвертирующей части.

*Быстродействие* логических элементов является одним из важнейших параметров логических элементов, оно оценивается задержкой распространения сигнала от входа к выходу элемента.

На рис. 3.1 приведена форма входного и выходного сигналов логического элемента (инвертора):  $t_{1,0}^z$  – время задержки переключения выхода элемента из состояния логической 1 в состояние логического 0;  $t_{0,1}^z$  – задержка переключения из состояния логического 0 в состояние логической 1.

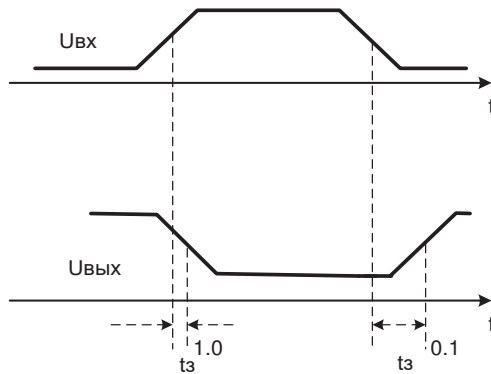


Рис. 3.1 – Диаграмма напряжений на входе и выходе инвертора



Как видно из рисунка, время задержки измеряется на уровне, среднем между уровнями логического 0 и логической 1. *Средняя задержка распространения сигнала*

$$t_{3\text{cp}} = 0,5(t_{3}^{0,1} + t_{3}^{1,0}).$$

Этот параметр используется при расчете задержки распространения сигналов в сложных логических схемах.

Рассмотрим факторы, влияющие на быстродействие логического элемента, и методы повышения быстродействия.

Для повышения скорости переключения транзисторов в элементе необходимо использовать более высокочастотные транзисторы и переключение транзисторов проводить большими управляющими токами в цепи базы; существенное уменьшение времени задержки достигается благодаря использованию ненасыщенного режима работы транзисторов (в этом случае исключается время, необходимое на рассасывание неосновных носителей в базе при выключении транзисторов).

*Помехоустойчивость* определяется максимальным значением помехи, не вызывающей нарушение работы элемента.

Для количественной оценки помехоустойчивости воспользуемся так называемой *передаточной характеристикой* логического элемента (инвертора). На рис. 3.2 приведена типичная форма этой характеристики.

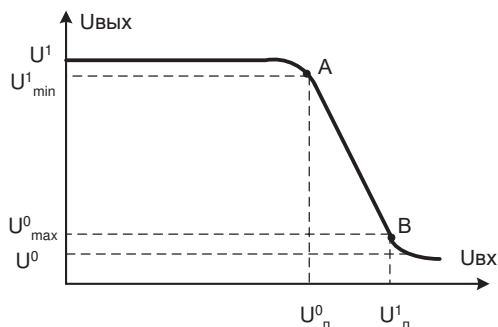


Рис. 3.2 – Передаточная характеристика инвертора

Передаточная характеристика представляет собой зависимость выходного напряжения от входного. Для ее получения необходимо соединить все входы логического элемента и, изменяя напряжение на входе, отмечать соответствующие значения напряжения на выходе. При увеличении входного напряжения от нуля до порогового уровня лог. 0  $U_n^0$  напряжение на выходе уменьшается от уровня логической 1 до некоторого минимально допустимого уровня логической 1  $U_{\min}^1$ . Дальнейшее увеличение входного напряжения приводит к резкому снижению выходного. При больших значениях входного напряжения, превышающих пороговый уровень логической 1  $U_n^1$ , на выходе устанавливается

напряжение, не превышающее максимально допустимого уровня логического 0  $U_{\max}^0$ . Таким образом, при нормальной работе элемента в статическом (установившемся) режиме недопустимы входные напряжения  $U_n^0 < u_{вх} < U_n^1$ .

Допустимыми считаются такие помехи, которые, сложившись с входным напряжением, не выведут его в область недопустимых значений  $U_n^0 < u_{вх} < U_n^1$ .

Логические элементы могут быть классифицированы по различным признакам. Вариант классификации представлен на рис 3.3.

### Классификация логических элементов

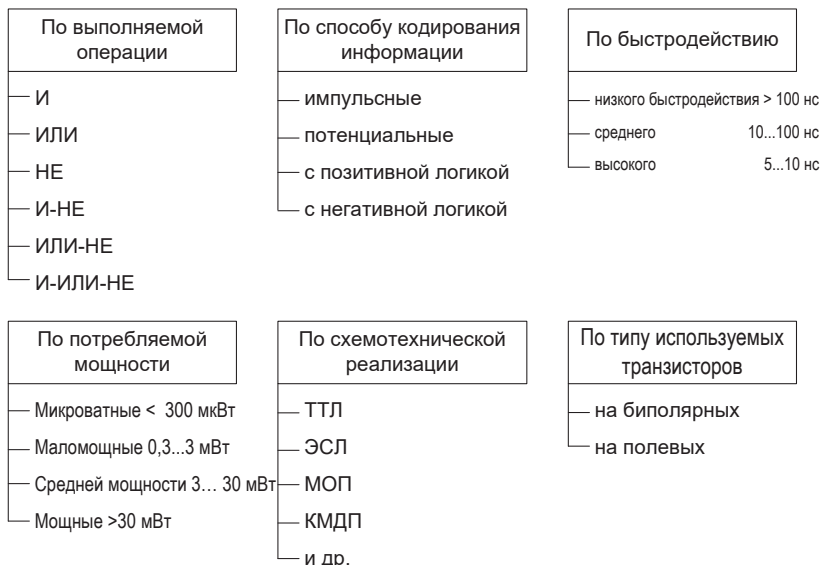


Рис. 3.3 – Классификация логических элементов

Для успешного применения ИМС в области цифровой схемотехники совсем не обязательно знать их подробное внутреннее устройство. Нужно знать основополагающие правила, касающиеся напряжений питания и требований к входным и выходным сигналам. В цифровой схемотехнике первоочередной интерес для нас представляют логические функции компонентов, а не их электрические характеристики.

Выше отмечалось, что логические функции и их аргументы принимают значение лог. 0 (*уровень лог. 0*) и лог. 1 (*уровень лог. 1*). При этом следует иметь в виду, что в устройствах лог. 0 и лог. 1 соответствует напряжение определенного уровня (либо формы). Наиболее часто используется два способа физического представления лог. 0 и лог. 1: *потенциальный* и *импульсный*.

При потенциальной форме (рис. 3.4, а и 3.4, б) для представления лог. 0 и лог. 1 используется напряжение двух уровней: высокий уровень соответствует

логической 1 (уровень лог. 1) и низкий уровень соответствует логического 0 (уровень лог. 0). Такой способ представления логических величин называется *положительной логикой*. Относительно редко используют так называемую *отрицательную логику*, при которой лог. 1 ставят в соответствие низкий уровень напряжения, а логического 0 – высокий уровень. В дальнейшем, если это не оговаривается особо, будем пользоваться только положительной логикой.

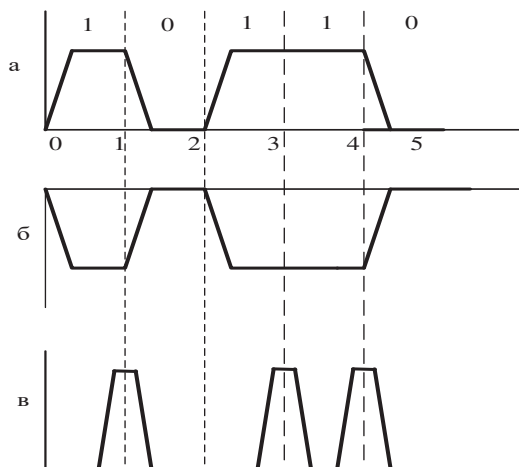


Рис. 3.4 – Схема представления логических 0 и 1

При импульсной форме лог. 1 соответствует наличие импульса, лог. 0 – отсутствие импульса (рис. 3.4, в).

При потенциальной форме соответствующая сигналу информация (лог. 1 либо лог. 0) может быть определена практически в любой момент времени. При импульсной форме соответствие между уровнем напряжения и значением логической величины устанавливается в определенные дискретные моменты времени (*такты времени*), обозначенные на рис. 3.4, в целыми числами  $t = 0, 1, 2, \dots$

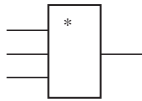
### ***Логические элементы базиса И, ИЛИ, НЕ на дискретных компонентах***

***Диодный элемент ИЛИ (сборка)***. Логический элемент ИЛИ, выполняемый на диодах, имеет два и более входа и один выход. Элемент может работать как при потенциальном, так и при импульсном представлении логических величин.

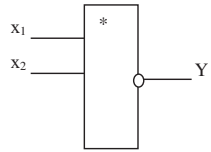
На рис. 3.5, а представлена схема диодного элемента для работы с потенциалами и импульсами положительной полярности. При использовании отрицательной логики и отрицательных потенциалов либо импульсов отрицательной полярности необходимо изменить полярность включения диодов, как показано на рис. 3.5, б.

Общие обозначения логических элементов

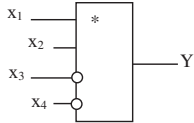
Логический элемент:  
\* – указатель функции



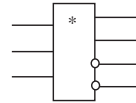
Элемент с инверсным выходом



Элемент с несколькими прямыми и инверсными входами  
 $Y = \varphi(x_1, x_2, x_3, x_4)$

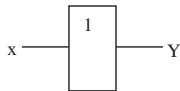


Элемент с несколькими прямыми и инверсными выходами

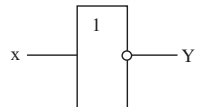


Элементы, реализующие логические функции

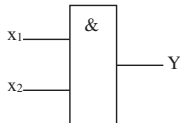
Повторитель  
 $Y = x$



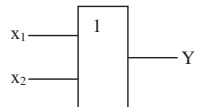
Инвертор (НЕ)  
 $Y = \bar{x}$



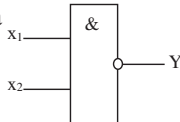
Конъюнктор (И) (ИЛИ)  
 $Y = x_1 \cdot x_2$



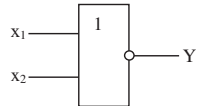
Дизъюнктор  
 $Y = x_1 \vee x_2$



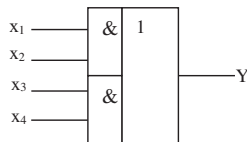
Элемент Шеффера (И-НЕ)  
 $Y = \overline{(x_1 \cdot x_2)}$



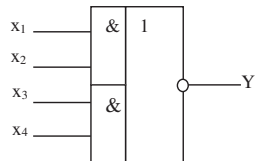
Элемент Пирса (ИЛИ-НЕ)  
 $Y = \overline{(x_1 \vee x_2)}$



И-ИЛИ  
 $Y = x_1 \cdot x_2 \vee x_3 \cdot x_4$



И-ИЛИ-НЕ  
 $Y = \overline{(x_1 \cdot x_2 \vee x_3 \cdot x_4)}$



Рассмотрим работу схемы на рис. 3.5, а. Если импульс (либо высокий потенциал) действует лишь на одном выходе, то открывается подключенный к этому входу диод и импульс (либо высокий потенциал) передается через открытый диод на резистор R. При этом на резисторе R образуется напряжение такой полярности, при которой диоды в цепях остальных входов оказываются под действием запирающего напряжения. Если сигналы, соответствующие лог. 1, одновременно поступают на несколько входов, то при строгом равенстве уровней этих сигналов откроются все диоды, подключенные к этим входам. Если сопротивление открытого диода мало по сравнению с сопротивлением резистора R, уровень выходного напряжения будет близок к уровню входного сигнала независимо от того, на скольких входах одновременно действует сигнал лог. 1.

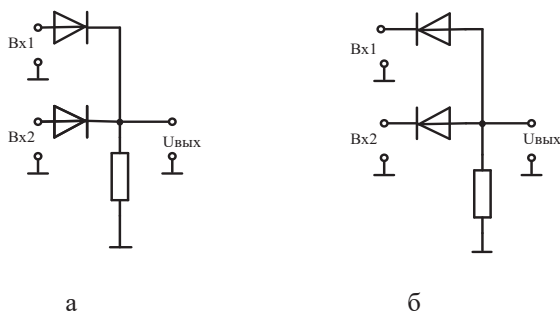


Рис. 3.5 – Диодные логические элементы:  
а – положительной логики; б – отрицательной логики

Заметим, что если уровни входных сигналов разнятся, то открывается лишь диод того из входов, уровень сигнала на котором имеет наибольшее значение. На резисторе R образуется напряжение, близкое к наибольшему из напряжений, действующих на входах. Все остальные диоды закрываются, отключая от выхода источник с малым уровнем сигнала.

Таким образом, на выходе элемента образуется сигнал, соответствующий лог. 1, если хотя бы на одном из входов действует лог. 1. Следовательно, элемент реализует операцию дизъюнкции (операцию ИЛИ).

Рассмотрим факторы, влияющие на форму выходного импульса. Пусть элемент имеет  $n$  входов и на один из них подан прямоугольный импульс напряжения от источника с выходным сопротивлением  $R_{\text{вых}}$ . Подключенный к этому входу диод открыт и представляет собой малое сопротивление. Остальные диоды закрыты, емкости  $C_d$  их  $p$ - $n$ -переходов через выходное сопротивление подключенных ко входам источников оказываются включенными параллельно выходу элемента. Вместе с емкостью нагрузки и монтажа  $C_n$  образуется эквивалентная емкость  $C_{\text{эк}} = C_n + (n-1) C_d$ , подключенная параллельно R (рис. 3.6, а).

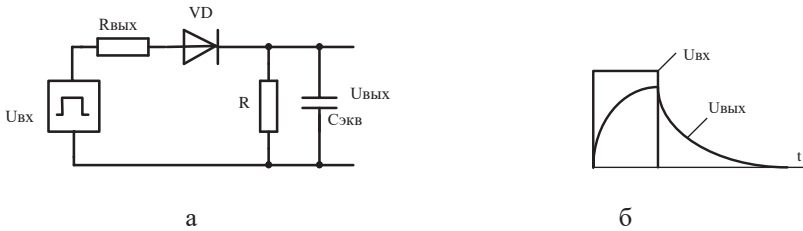


Рис. 3.6 – Эквивалентная схема логического элемента – а;  
диаграмма работы логического элемента – б

В момент подачи импульса на вход, из-за емкости  $C_{\text{ЭК}}$ , напряжение на выходе не может возрасти скачком. Оно растёт по экспоненциальному закону с постоянной времени

$$\tau_{\text{нар}} = RR_{\text{ВЫХ}}C_{\text{ЭК}}/(R+R_{\text{ВЫХ}}) \approx R_{\text{ВЫХ}}C_{\text{ЭК}}$$

(так как  $R_{\text{ВЫХ}} < R$ ), стремясь к значению  $U_{\text{ВЫХ}}R/(R+R_{\text{ВЫХ}})$ .

В момент окончания входного импульса напряжение на заряженном конденсаторе  $C_{\text{ЭК}}$  не может упасть скачком; оно снижается по экспоненциальному закону с постоянной времени  $\tau_{\text{СП}} = RC_{\text{ЭК}}$  (в это время все диоды оказываются закрытыми), так как  $\tau_{\text{СП}} \gg \tau_{\text{нар}}$ , длительность среза выходного импульса больше длительности его фронта (рис. 3.6, б). Подача следующего импульса на вход элемента допускается лишь после того, как остаточное напряжение на выходе от действия предыдущего импульса снизится до определенного малого значения. Поэтому медленный спад выходного напряжения вызывает необходимость увеличения тактового интервала и, следовательно, является причиной снижения быстродействия.

*Диодный элемент И (схема совпадения).* Логический элемент **И** имеет один выход и два или более входов. Диодный элемент **И** может работать с информацией, представленной как в потенциальной, так и в импульсной форме.

На рис. 3.7, а приведена схема, используемая при положительных значениях входных напряжений. При использовании отрицательной логики и отрицательных входных напряжений либо импульсов отрицательной полярности необходимо изменить полярность напряжения источника питания и полярность включения диодов (рис. 3.7, б).

Пусть на одном из входов цепи (рис. 2.9, а) действует низкий уровень напряжения, соответствующий уровню лог. 0. Ток будет замыкаться в цепи от источника  $E$  через резистор  $R$ , открытый диод и источник низкого входного напряжения. Так как сопротивление открытого диода мало, то низкий потенциал со входа через открытый диод будет передаваться на выход. Диоды, подключенные к остальным входам, на которые действует высокий уровень напряжения, оказываются закрытыми. Действующее на диоде напряжение можно определить суммированием напряжений при обходе внешней, по

отношению к диоду, цепи от его анода к катоду. При таком обходе напряжение на диоде оказывается равным  $U_d = u_{\text{вых}} - u_{\text{вх}}$ . Таким образом, выходное напряжение, прикладываемое к анодам диодов, является для них положительным, стремящимся открыть диоды; входное напряжение, прикладываемое к катоду, – отрицательным, стремящимся закрыть диод. И если  $u_{\text{вых}} < u_{\text{вз}}$ , то  $U_d$  отрицательно и диод закрыт.

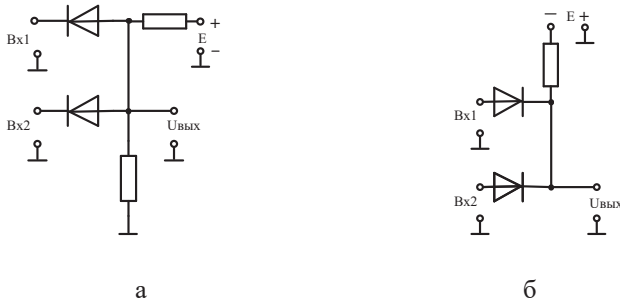


Рис. 3.7 – Диодный логический элемент И:

а – положительная логика; б – отрицательная логика

Именно поэтому, когда на выходе элемента низкий потенциал (уровень лог. 0), а на входе высокий потенциал (уровень лог. 1), подключенный к этому входу диод оказывается закрытым.

Таким образом, если хотя бы на одном из входов действует напряжение низкого уровня (лог. 0), то на выходе элемента образуется напряжение низкого уровня (лог. 0).

Пусть на всех входах действуют напряжения высокого уровня (лог. 1). Они могут несколько отличаться по значению. При этом будет открыт тот диод, который подключен ко входу с более низким напряжением. Это напряжение через диод будет передаваться на выход. Остальные диоды будут практически закрыты. На выходе установится напряжение высокого уровня (лог. 1).

Следовательно, на выходе элемента устанавливается напряжение уровня лог. 1 в том и только в том случае, когда на всех входах действует напряжение уровня лог. 1. Таким образом, убеждаемся в том, что элемент выполняет логическую операцию **И**.

Рассмотрим форму выходного импульса (рис. 3.8). Будем считать, что к выходу подключен некоторый эквивалентный емкостный элемент  $C_{\text{эк}}$ , емкость которого включает емкости нагрузки, монтажа и закрытых диодов. В момент подачи импульса напряжения одновременно на все входы напряжение на  $C_{\text{эк}}$  (на выходе элемента) не может возрасти скачком. Все диоды вначале оказываются закрытыми входными напряжениями, являющимися для диодов отрицательными. Поэтому источники входных сигналов будут отключены от  $C_{\text{эк}}$ . Конденсатор  $C_{\text{эк}}$  заряжается от источника питания через резистор R. Напряжение на

конденсаторе (а значит, и на выходе элемента) растет по экспоненциальному закону с постоянной времени  $\tau = RC_{\text{ЭК}}$  (рис. 3.8, б). В момент времени, когда  $u_{\text{ВЫХ}}$  превысит минимальное из входных напряжений, откроется соответствующий диод и рост  $u_{\text{ВЫХ}}$  прекратится. Ток от источника  $E$ , ранее замыкавшийся через  $C_{\text{ЭК}}$ , переключается в цепь открытого диода.

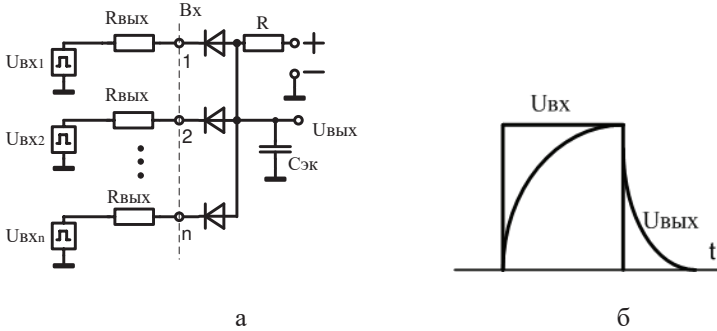


Рис. 3.8 – Эквивалентная схема – а; диаграмма работы – б

В момент окончания входных импульсов все диоды открываются положительным для них напряжением  $u_{\text{ВЫХ}}$ . Происходит относительно быстрый разряд  $C_{\text{ЭК}}$  через открытые диоды и малые выходные сопротивления источников входных сигналов. Напряжение на выходе снижается по экспоненциальному закону с малой постоянной времени  $\tau_{\text{СП}} = R_{\text{ВЫХ}}C_{\text{ЭК}}/n$ .

Сравнение форм выходных импульсов диодных элементов **ИЛИ** и **И** показывает, что в элементе **ИЛИ** оказывается более растянутым срез импульса, в элементе **И** – его фронт.

**Транзисторный элемент НЕ (инвертор).** Операция **НЕ** может быть реализована ключевым элементом, представленным на рис. 3.9, а. При низком уровне сигнала, соответствующем лог. 0, транзистор закрыт, на его выходе устанавливается напряжение высокого уровня  $E$  (лог. 1).

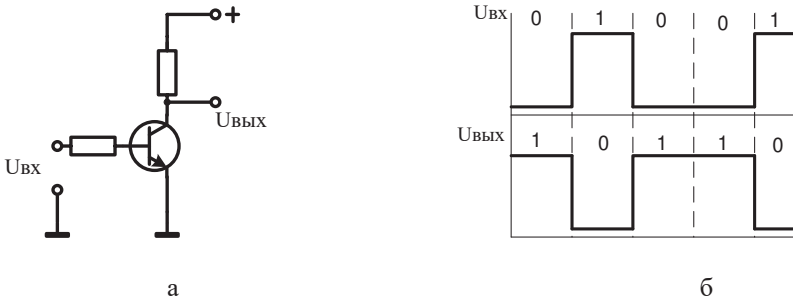


Рис. 3.9 – Схема НЕ – а; диаграмма работы схемы – б



И наоборот, при высоком уровне входного напряжения (уровне лог. 1) транзистор насыщен, на его выходе устанавливается напряжение, близкое к нулю (уровня лог. 0). Графики входных и выходных напряжений представлены на рис. 3.9, б.

### 3.2. Интегральные логические элементы ДТЛ, ТТЛ и ТТЛШ логики базиса И–НЕ и их параметры

Интегральные логические элементы используются при потенциальной форме представления логических величин.

*ДТЛ-серия.*

*Элемент И–НЕ диодно-транзисторной логики (ДТЛ).* Схема интегрального элемента И–НЕ типа ДТЛ показана на рис. 3.10.

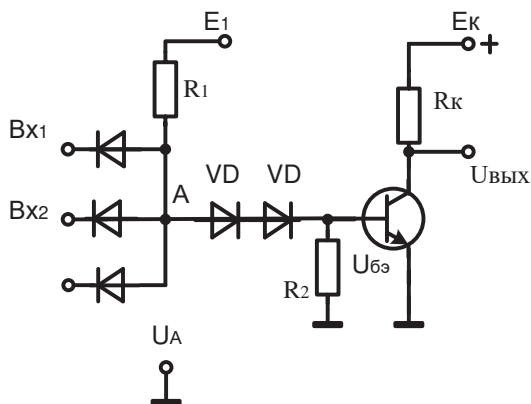


Рис. 3.10 – Схема логического элемента И–НЕ

Элемент может быть разбит на две последовательно включенные функциональные части. Входные величины подаются на часть, представляющую собой диодный логический элемент И. Вторая часть элемента, выполненная на транзисторе, представляет собой инвертор (выполняющий операцию НЕ). Таким образом, в элементе последовательно выполняются логические операции И и НЕ и, следовательно, в целом он реализует логическую операцию И–НЕ.

Если на всех входах элемента действует напряжение высокого уровня (лог. 1), то на выходе первой части схемы (в точке А) образуется напряжение высокого уровня. Это напряжение через диоды VD передается на вход транзистора, который оказывается в режиме насыщения, на выходе элемента напряжение низкого уровня (лог. 0).

Если же хотя бы на одном из входов будет действовать напряжение низкого уровня (лог. 0), то в точке А образуется напряжение низкого уровня (близкого к нулю), транзистор закрыт и на выходе элемента напряжение высокого уровня

(лог. 1). Работа диодного элемента И в интегральном исполнении отличается от работы рассмотренного выше такого же элемента на дискретных компонентах тем, что при одновременной подаче лог. 1 на все входы все диоды оказываются закрытыми. Благодаря этому уменьшается до весьма малого значения потребление тока от источника, подающего на вход напряжение лог. 1.

Рассмотрим подробнее работу инверторной части элемента. Вначале отметим некоторые особенности транзисторов интегральных микросхем. В микросхемах используются кремниевые транзисторы типа  $n-p-n$  (при этом напряжение коллекторного питания имеет положительную полярность и транзистор открывается при положительном напряжении между базой и эмиттером). На рис. 3.11 показана типичная зависимость тока коллектора от напряжения между базой и эмиттером в активном режиме.

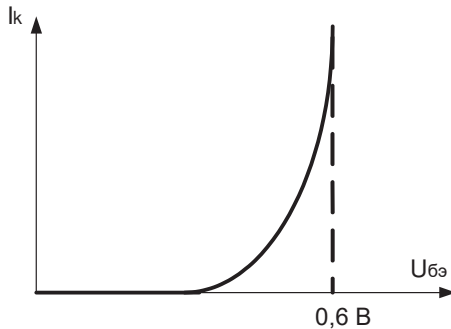


Рис. 3.11 – Напряжение между коллектором и эмиттером

Особенность этой характеристики в том, что практически транзистор начинает открываться при относительно высоких значениях базового напряжения (обычно превышающих 0,6 В). Эта особенность позволяет обходиться без источников базового смещения, так как и при положительных напряжениях на базе в десятые доли вольта транзистор оказывается практически закрытым. Наконец, еще одна особенность транзистора микросхем состоит в том, что напряжение между коллектором и эмиттером в режиме насыщения сравнительно велико (оно может быть 0,4 В и выше).

Пусть сигналы на входы логического элемента подаются с выходов аналоговых элементов. Примем напряжение лог. 1 равным 2,6 В, напряжение лог. 0 равным 0,6 В, напряжения на открытых диодах и напряжение база – эмиттер насыщенного транзистора равным 0,8 В.

При подаче на все входы напряжения 2,6 В (уровень лог. 1) закрываются диоды на входах, ток от источника  $E_1$  через резистор  $R_1$ , диоды  $VD$  проходит в базу транзистора, устанавливая транзистор в режим насыщения. На выходе элемента образуется напряжение низкого уровня 0,6 В (уровень лог. 0). Напряжение  $U_A$  равно сумме напряжений на диодах  $VD$  и напряжения  $U_{БЭ}$ :  $3 \cdot 0,8 =$

2,4 В. Таким образом, входные диоды оказываются под обратным напряжением 0,2 В.

Если хотя бы на один из входов подается напряжение низкого уровня 0,6 В (уровень лог. 0), то ток от источника  $E_1$  замыкается через  $R_1$ , открытый входной диод и источник входного сигнала. При этом  $U_A = 0,8 + 0,6 = 1,4$  В. При таком напряжении транзистор оказывается закрытым благодаря смещению, обеспечиваемому диодами VD (эти диоды называются *смещающими диодами*). Ток от источника  $E_1$ , протекая через резистор  $R_1$ , диоды VD и резистор  $R_2$ , создает на смещающих диодах падение напряжения, близкое к  $U_A$ . Напряжение  $U_{БЭ}$  положительно, но значительно меньше 0,6 В, и транзистор закрыт.

Выпускались серии 104, 121, 156, 215, 511 диодно-транзисторной логики, на смену которым пришли более совершенные микросхемы транзисторно-транзисторной логики.

### ТТЛ-серия

Базовым логическим элементом этой серии является элемент И-НЕ (рис. 3.12). Схема логического элемента ТТЛ, состоит из двух последовательно включенных функциональных частей: схемы, выполняющей операцию И, и схемы инвертора (операция НЕ).

Отличительная особенность построения схемы И в элементе ТТЛ состоит в том, что в ней использован один многоэмиттерный транзистор МТ.

Эмиттерные переходы МТ играют роль входных диодов, а коллекторный переход – роль смещающего диода в цепи базы транзистора инвертирующей части схемы элемента.

При рассмотрении принципа работы МТ его можно представить состоящим из отдельных транзисторов с объединенными базами и коллекторами.

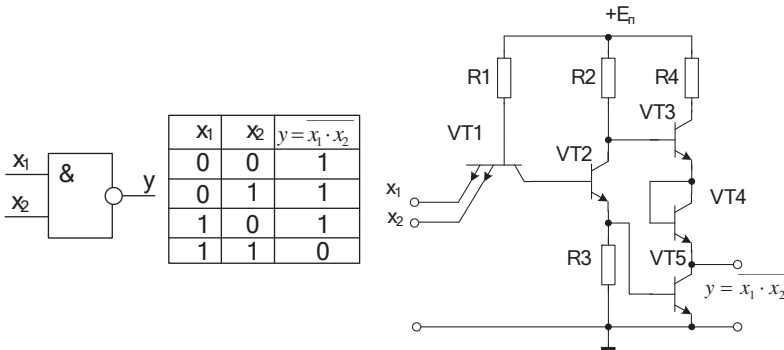


Рис. 3.12 – Схема логического элемента И-НЕ ТТЛ

Промежуточный каскад на  $VT2$  формирует противофазные управляющие сигналы на транзисторы  $VT3$  и  $VT4$  выходного каскада. Транзистор  $VT5$  применяется в диодном включении.

*Работа электрической схемы.* При подаче на любой из входов, или одновременно на два входа  $x_1$  и  $x_2$  напряжения, соответствующего логическому нулю, а это возможно путем соединения эмиттера  $VT1$  с корпусом, транзистор  $VT1$  открывается, а транзистора  $VT2$  закрывается, т. к. его база через транзистор  $VT1$  будет подключена к корпусу, т. е. ее потенциал будет равен нулю.

В следствии, этого транзистор  $VT3$  открывается положительным напряжением подаваемым через  $R2$ , а  $VT5$  закрывается, т. к. через резистор  $R3$  его база будет подключена к корпусу.

На выходе  $y$  (эмиттер  $VT4$ ) формируется высокий потенциал, т. е. логическая единица.

Если на оба входа подается высокий потенциал (логическая единица), то транзистор  $VT1$  закроется, в следствии, чего  $VT2$  откроется, но напряжение на его коллекторе будет мало, и транзистор  $VT3$  закроется.

Однако падение напряжения на  $R3$ , в следствии открытого транзистора  $VT2$ , откроет транзистор  $VT5$  и напряжение на его коллекторе будет близко к нулю, т. е. на выходе формируется логический нуль.

Рассмотренная схема обладает хорошей нагрузочной способностью, повышенной помехоустойчивостью и обеспечивает достаточно высокое быстродействие.

Такие элементы применяют в микросхемах малой и средней степени интеграции, а также в выходных каскадах БИС.

Основные параметры ТТЛ-серий приведены в табл. 3.1.

Таблица 3.1

Серия	$K_{раз}$	$t_{зад},$ нс	$P_{пот},$ мВт	$F,$ МГц	$U_{лог0},$ В	$U_{лог1},$ В	$E_{л},$ В
133,155	10	15 - 22	22	10	0,4	2,4	5
130,131	10	7 - 10	44	30	0,4	2,4	5
134	10	100	2	3	0,4	2,4	5

### ТТЛШ-серия

Принцип построения этой серии ЛЭ аналогичен ТТЛ, однако вместо обычных транзисторов используются транзисторы с диодами Шоттки, которые включены параллельно коллекторному переходу (рис. 3.13).

В результате этого транзисторы не переходят в режим насыщения и даже при воздействии насыщающего входного напряжения.

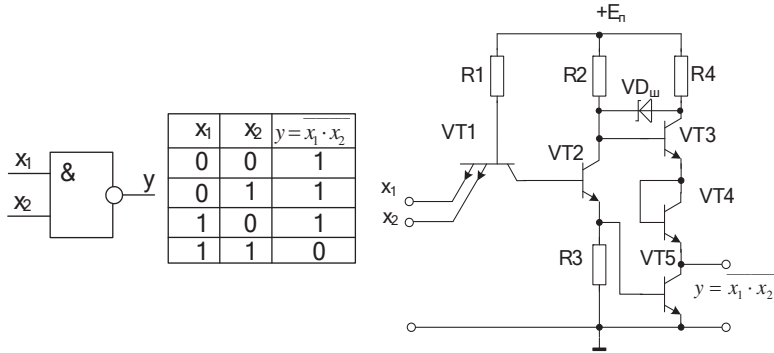


Рис. 3.13 – Схема логического элемента И-НЕ ТТЛШ

Поэтому ЛЭ могут работать на более высоких частотах (табл. 3.2).

Таблица 3.2

Серия	$K_{раз}$	$t_{зад},$ нс	$P_{пот},$ мВт	$F,$ МГц	$U_{лог0},$ В	$U_{лог1},$ В	$E_n,$ В
531,530	10	4,5 - 5	19	50	0,4	2,4	5
555,533	10	20	2	10	0,4	2,4	5

Кроме того, эта серия характеризуется умеренным потреблением энергии, что позволяет увеличить в 5 раз количество ЛЭ на одном кристалле.

К этой серии относятся микропроцессорные БИС 589, 1802, 1804 серий, а также интегральные схемы 1530, 1531, 1533 серий.

Серия ТТЛШ имеет зарубежный аналог – LS-TTL. В зарубежных микросхемах широко представлена серия 74. Маркировка микросхем этой серии начинается с цифр 74, например 7400, 7408, 7432, 74121; чаще называют *стандартной TTL серией*.

Разновидности аналогичных микросхем с диодами Шоттки имеют в середине буквы LS, например 74LS00, 74LS08, 74LS32, 74LS121. КМОП-микросхемы образуют часть серий 4000, и их номера начинаются с цифры 4, например 4001, 4174, 4501, 4574.

Элементы ТТЛ, а тем более их усовершенствованная модификация – ТТЛШ, обладают хорошим быстродействием, имеют удовлетворительные электрические и эксплуатационные характеристики, хорошо отработаны технически, следствием чего является достаточно высокая надежность.

### 3.3. Логические элементы эмиттерно-связанной логики

#### ЭСЛ-серия

Схемотехнической основой этой серии являются переключатели тока (рис. 3.14), транзисторы которого работают в активном режиме.

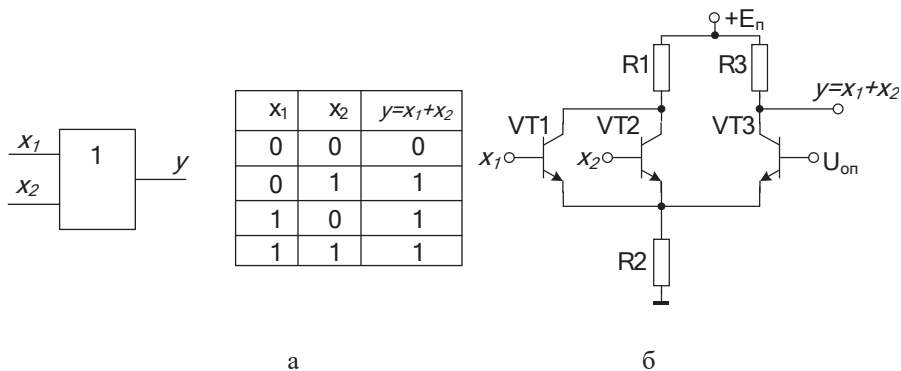


Рис. 3.14 – Схема логического элемента ИЛИ ЭСЛ

Базовым логическим элементом этой серии является элемент ИЛИ.

Схема состоит из дифференциального усилителя, собранного на транзисторах  $VT1$ – $VT3$ .

#### Работа электрической схемы.

В исходном состоянии при подключении питания транзисторы  $VT1$ ,  $VT2$  закрыты (потенциал их баз равен нулю), а транзистор  $VT3$  приоткрыт, за счет опорного напряжения  $U_{on}$  и на его выходе « $y$ » формируется напряжение, меньшее логического нуля ( $0,88 B$ ), т. е. схема находится в режиме ожидания.

В при подаче на вход  $x_1$  и  $x_2$ , или на оба сразу логического нуля ( $0,96 B$ ) транзисторы  $VT1$ ,  $VT2$  приоткрываются, а вследствие работы дифференциального каскада транзистор  $VT3$  приоткрывается и на его выходе « $y$ » сформируется логический ноль ( $0,96 B$ ).

Если на вход  $x_1$  и  $x_2$ , или на один из них подается напряжение логической единицы ( $1,65 B$ ), то транзисторы  $VT1$ ,  $VT2$  откроются, а это транзистор  $VT3$  еще сильнее закроется и на выходе формируется логическая единица ( $1,65 B$ ).

Для данной серии отличительным является то, что логический ноль и единица отличаются меньшим перепадом напряжения чем, для серии ТТЛ. Это обусловлено тем, что транзисторы при формировании лог. 0 и лог. 1 находятся в активном режиме и как следствие не входят в режим насыщения и поэтому быстродействие серии ЭСЛ значительно выше, чем у серии ТТЛ.

На практике схема ЛЭ серии ЭСЛ подключается к отрицательной шине источника напряжения  $E_n = -5,2 \text{ В}$ , коллекторные шины заземляются.

Такое включение обеспечивает меньшую зависимость выходного напряжения от наводок по цепи питания и лучшую помехоустойчивость.

На рис. 3.15 приведена трехходовая схема, выполняющая операцию ИЛИ и ИЛИ-НЕ.

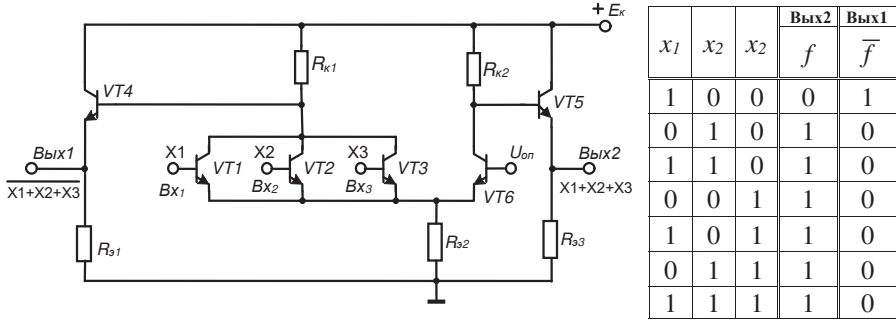


Рис. 3.15 – Схема логического элемента ИЛИ-НЕ ЭСЛ

Исключение режима насыщения и связанной с ним задержки рассасывания обеспечивает более высокое быстродействие элементов ЭСЛ по сравнению с элементами ТТЛ, кроме того, ЭСЛ-серия характеризуется значительным потреблением энергии (табл. 3.3).

Таблица 3.3

Серия	$K_{раз}$	$t$ , нс	$P_{пот}$ , мВт	$F$ , МГц	$U_{лог 1}$ , В	$U_{лог 0}$ , В	$E_n$ , В
100,500	15	4	35	100	1,65	0,96	5,2
1500, 138	25	2	35	300	1,65	0,96	5,2

### 3.4. Логические элементы интегральной инжекционной логики

#### И<sup>2</sup>Л-серия

Тенденция постоянного роста уровня интеграции БИС связана с решением следующих основных задач: обеспечение низкой мощности потребления, повышение быстродействия БИС, сокращение числа операций технологического цикла их изготовления. Требования по быстродействию и мощности рассеяния принципиально несовместимы.

Данная серия используется для создания элементов ИЛИ-НЕ, И-НЕ.

Для повышения технологичности изготовления желательнее при разработке ИС применять схемотехнические решения, использующие только однотипные элементы, например транзисторы.

Этот путь, как было показано ранее, реализован в ИС МДП, что наряду с другими достоинствами является причиной их широкого распространения.

Однако, как уже отмечалось, ключ на биполярных транзисторах на сегодняшний день обладает лучшими как ключевыми, так и частотными свойствами. Это является предпосылкой к постоянному поиску новых схемотехнических решений для реализации биполярных ИС.

Такой поиск привел к почти одновременной разработке фирмами Philips и IBM элемента интегральной инжекционной логики (И<sup>2</sup>Л)

Особенностью элементов И<sup>2</sup>Л является:

1) отсутствие резисторов, что резко упрощает технологию производства ИС;

2) использование токового принципа питания, при котором в ИС задается не напряжение, а ток, который непосредственно инжектируется в область полупроводника, образующего структуру одного из транзисторов;

3) пространственное совмещение в кристалле полупроводника областей, функционально принадлежащих различным транзисторам. При этом структура располагается как по горизонтали (планарно), так и по вертикали (рис. 3.16). Такое решение позволяет отказаться от применения специальных решений для отделения областей, принадлежащих различным элементам, как это необходимо делать в элементах ТТЛ и ЭСЛ. Для создания такой структуры требуется две фазы диффузии в кремнии с проводимостью *n*-типа: в процессе первой фазы образуются области  $p_1$  и  $p_2$ , второй фазы – области  $n_2$ .

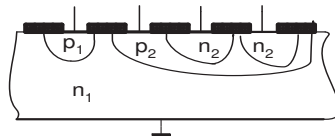


Рис. 3.16 – Топология логического элемента интегральной инжекционной логики

4) малое значение логического перепада  $0,4...0,6 V$  (потенциал лог. 0 ( $0,1...0,2 V$ ), а лог.1 ( $0,6...0,7 V$ ), что позволяет максимально увеличить быстродействие элемента до  $10 ns$ .

Однако малый перепад между лог. 0 и лог. 1 обуславливает малую помехоустойчивость.

Элемент имеет структуру  $p_1-n_1-p_2-n_2$ . Такую четырехслойную структуру удобно рассматривать, представив ее соединением двух обычных трехслойных транзисторных структур:

$$p_1-n_1-p_2.$$



Соответствующая такому представлению схема показана на рис. 3.17, а. Рассмотрим работу элемента по этой схеме.

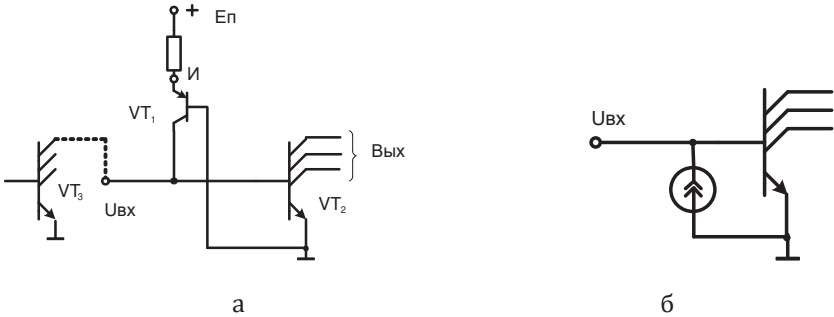


Рис. 3.17 – Схема элемента трехслойной структуры – а;  
эквивалентная схема – б

Транзистор  $VT_2$  со структурой типа  $n_1-p_2-n_2$  выполняет функции инвертора, имеющего несколько выходов (каждый коллектор образует отдельный выход элемента по схеме с открытым коллектором).

Транзистор  $VT_1$ , называемый *инжектором*, имеет структуру типа  $p_1-n_1-p_2$ . Так как область  $n_1$  у этих транзисторов общая, эмиттер транзистора  $VT_2$  должен быть соединен с базой транзистора  $VT_1$ ; наличие общей области  $p_2$  приводит к необходимости соединения базы транзистора  $VT_2$  с коллектором транзистора  $VT_1$ . Так образуется соединение транзисторов  $VT_1$  и  $VT_2$ , показанное на рис. 3.17, а.

Так как на эмиттере транзистора  $VT_1$  действует положительный потенциал, а база находится под нулевым потенциалом, эмиттерный переход оказывается смещенным в прямом направлении и транзистор открыт.

Коллекторный ток транзистора может замкнуться либо через транзистор  $VT_3$  (инвертор предыдущего элемента), либо через эмиттерный переход транзистора  $VT_2$ .

Если предыдущий логический элемент находится в открытом состоянии (открыт транзистор  $VT_3$ ), то на входе данного элемента низкий уровень, который, действуя на базе  $VT_2$ , удерживает этот транзистор в закрытом состоянии. Ток инжектора  $VT_1$  замыкается через транзистор  $VT_3$ . При закрытом состоянии предыдущего логического элемента (закрыт транзистор  $VT_3$ ) коллекторный ток инжектора  $VT_1$  втекает в базу транзистора  $VT_2$ , и этот транзистор устанавливается в открытое состояние.

Таким образом, при закрытом  $VT_3$  транзистор  $VT_2$  открыт и, наоборот, при открытом  $VT_3$  транзистор  $VT_2$  закрыт. Открытое состояние элемента соответствует лог. 0, закрытое – состоянию лог. 1.

Инжектор является источником постоянного тока (который может быть общим для группы элементов). Часто пользуются условным графическим обозначением элемента, представленным на рис. 3.17, б.

На рис. 3.18, а показана схема, реализующая операцию ИЛИ–НЕ. Соединение коллектора элементов соответствует выполнению операции так называемого *монтажного И*. Действительно, достаточно, чтобы, по крайней мере, один из элементов находился в открытом состоянии (состоянии лог. 0), тогда ток инжектора следующего элемента будет замыкаться через открытый инвертор и на объединенном выходе элементов установится низкий уровень лог. 0. Следовательно, на этом выходе формируется величина, соответствующая логическому выражению  $x_1 \cdot x_2$ . Применение к нему преобразования де Моргана приводит к выражению  $x_1 \cdot x_2 = (x_1 \vee x_2)$ . Следовательно, данное соединение элементов действительно реализует операцию ИЛИ–НЕ.

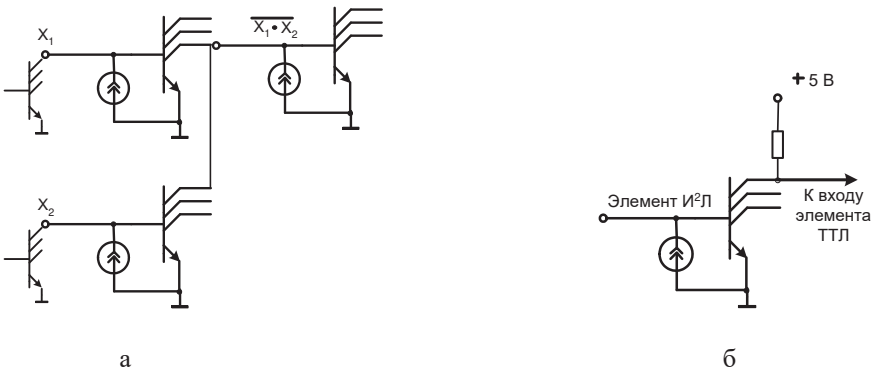


Рис. 3.18 – Логическая схема ИЛИ–НЕ – а;  
схема перехода от элемента И<sup>2</sup>Л к элементу ТТЛ – б

Логические элементы И<sup>2</sup>Л имеют следующие достоинства:

- обеспечивают высокую степень интеграции; при изготовлении схем И<sup>2</sup>Л используются те же технологические процессы, что и при производстве интегральных схем на биполярных транзисторах, но оказывается меньшим число технологических операций и необходимых фотошаблонов;

- используется пониженное напряжение ( $\approx 1$  В);

- обеспечивают возможность обмена в широких пределах мощности на быстродействие (можно изменять на несколько порядков потребляемую мощность, что соответственно приведет к изменению быстродействия);

- хорошо согласуется с элементами ТТЛ.

На рис. 3.18, б показана схема перехода от элемента И<sup>2</sup>Л к элементу ТТЛ. Характеристики ИМС И<sup>2</sup>Л серии показаны в табл. 3.4.

Серия	$K_{раз}$	$t$ , нс	$P$ , мВт	$F$ , МГц	$U_{вх0}$ , В	$U_{вх1}$ , В	$E_n$ В
531, 530, 584	-5	50	0,1	15	0,02	0,7	1-2

### 3.5. Логические элементы на МОП-транзисторах

#### МОП-серия

В серии МОП используются МОП (МДП)-транзисторы.

В этой серии выпускаются элементы НЕ, И-НЕ, ИЛИ-НЕ и другие ЛЭ, в частности по этой технологии выполнены современные микропроцессорные комплекты К580, К581, К1801, К1810, К1811.

В элементах НЕ (рис. 3.19), за счет применения однотипных структур (транзистор VT1 выполняет функцию резистора нагрузки в цепи стока  $R_c$ ) в этой серии достигается высокая плотность установки элементов.

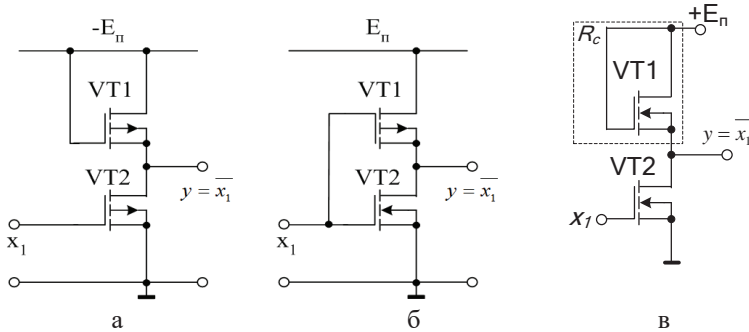
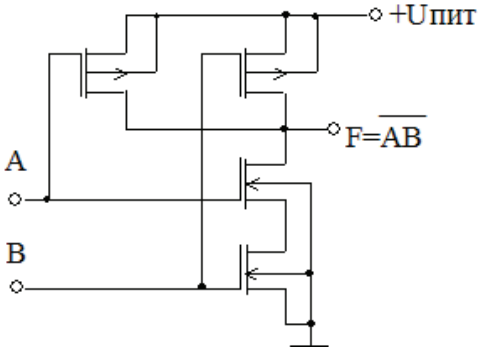


Рис. 3.19 – Элемент НЕ на полевых транзисторах:  
а – nМОП; б – pМОП; в – КМОП

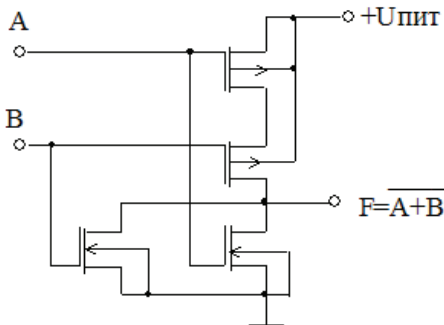
На рис. 3.20 представлен элемент И-НЕ. В данной схеме, чтобы открыть оба нижних транзистора надо, чтобы на входе А и входе В была логическая единица, при этом оба верхних транзистора закроются и на выходе F будет логический ноль. Если хотя бы на одном или на обоих входах будет логический ноль, то хотя бы один из нижних транзисторов закроется, один из верхних откроется, при этом на выходе будет логическая единица.



A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

Рис. 3.20 – Элемент И-НЕ КМОП

На рис. 3.21 представлен элемент ИЛИ-НЕ. В данной схеме, если на входах А или В (или на обоих) логическая единица, то откроется один или оба из нижних транзисторов, при этом закроется один или оба верхних транзистора, тогда на выходе логический ноль. Если на обоих входах логический ноль, то закроются оба нижних и откроются оба верхних транзистора, при этом на выходе логическая единица.



A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

Рис. 3.21 – Элемент ИЛИ-НЕ КМОП

Сери ИМС на полевых МОП транзисторах характеризуется очень малой мощностью потребления только в процессе переключения схемы (табл. 3.5), но низким быстродействием.

Таблица 3.5

Серия	$K_{раз}$	$t, нс$	$P, мВт$	$F, МГц$	$U_{вх0}, В$	$U_{вх1}, В$	$E_{п}, В$
172, МОПр	15	600	40	0,2	-2	-7,5	-27
178, МОПн	15	1200	4	0,2	0,5	9	27
176, КМОП 561, 564, КМОП	50	250	0,01	1	0,3	8,2	3-15

Однако в настоящее время удалось существенно повысить быстродействие серии КМОП о чем свидетельствуют параметры серия 1554 (табл. 3.6).

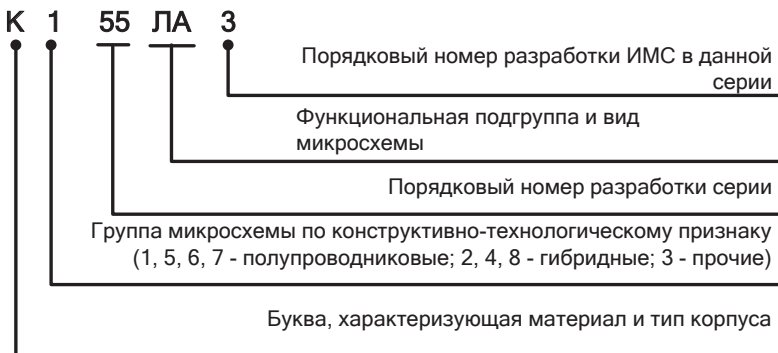
Таблица 3.6

Серия	$K_{раз}$	$t, нс$	$P_{пот}, мВт$	$F, МГц$	$U_{лог0}, В$	$U_{лог1}, В$	$E_{п}, В$
1554 КМОП	50	4,5	0,025	150	–	–	2-6

### 3.6. Условные обозначения ИМС

Промышленность выпускает в интегральном исполнении широкий ассортимент логических элементов. Ограничимся рассмотрением интегральных микросхем (ИМС) логических элементов, выполненных по технологии ТТЛ и нашедших наиболее широкое применение в ЭВМ общего и специального назначения. При этом отметим, что по технологии ТТЛ отечественной промышленностью выпускаются микросхемы серий 133, 155, 531, 555, 1533 и другие.

Порядок маркировки интегральных микросхем.



1 – буква, характеризующая условия применения, материал и тип корпуса.

- 2 – цифра, указывающая группу микросхемы по конструктивно-технологическому признаку.
- 3 – две-три цифры, указывающие номер разработки данной серии.
- 4 – две буквы, обозначающие функциональную подгруппу и вид микросхемы.
- 5 – порядковый номер разработки микросхемы в серии среди микросхем одного вида.

В зависимости от буквенного обозначения ИМС выполняют следующие функции: ЛА – И-НЕ, ЛБ – И-НЕ/ИЛИ-НЕ, ЛД – расширители, ЛЕ – ИЛИ-НЕ, ЛИ – И, ЛК – И-ИЛИ-НЕ/И-ИЛИ, ЛЛ – ИЛИ, ЛМ – ИЛИ-НЕ/ИЛИ, ЛН – НЕ, ЛП – прочие, ЛР – И-ИЛИ-НЕ, ЛС – И-ИЛИ.

Микросхемы заключены в стандартные корпуса, в основном с двумя типами выводов:

1) перпендикулярными плоскости корпуса, с шагом 2,5 мм. Такие корпуса называют корпусами типа ДИП (от английской аббревиатуры DIP – в переводе – корпус с двумя рядами выводов). В корпуса ДИП чаще всего заключаются микросхемы широкого применения, имеющие перед номером серии буквы К или КМ. Обычный температурный диапазон микросхем с буквой К – 10–70 °С. В микросхемах широкого применения более новых выпусков пластмассовый корпус часто обозначают буквами КР\* (рис. 3.22, а);

2) плоскими (планарными), шаг выводов 1,25 мм.

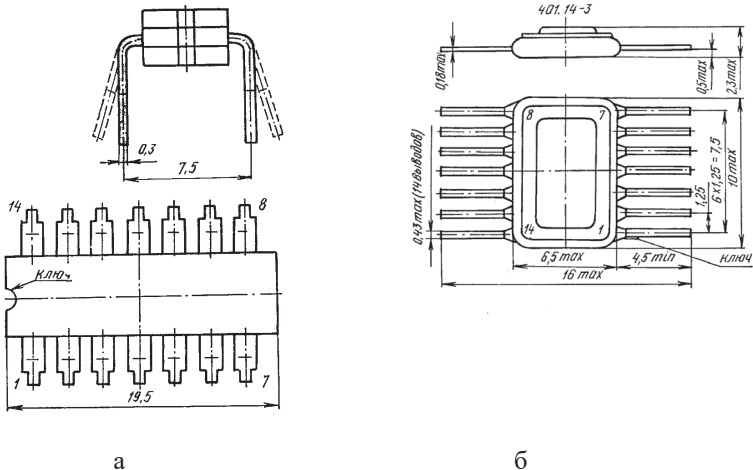


Рис. 3.22. Корпуса микросхем:  
 а – микросхемы широкого применения;  
 б – микросхемы специального назначения

В таких корпусах обычно выпускаются серии без буквы перед номером. Типичный диапазон их рабочих температур – 60...+ 125 °С (рис. 3.22, б).

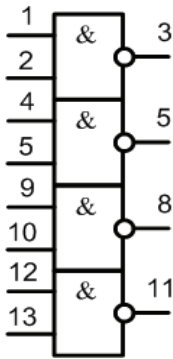
Подробные сведения об обозначениях микросхем, составе различных серий, условиях их эксплуатации и т. п. приводятся в справочниках или технических описаниях на цифровые устройства.

Габариты микросхемы определяют не кристалл кремния, а выводы из корпуса. Если элементы простые, то, чтобы использовать по возможности все выводы стандартного корпуса, в нем размещают несколько элементов. Простые логические элементы обычно размещаются в корпусах с 14 выводами, из которых один вывод – это питание и один общий. Остальные 12 выводов – логические. Пример: 4х(2И) – четыре двухвходовых элемента И (заняты все 12 логических выводов).

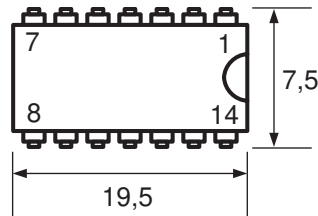
\* Этот тип серии в технике специального назначения не используется, т. к. имеет малый температурный диапазон.

Более сложные логические узлы размещаются в корпусах с 16, 24 и большим числом выводов.

Пример: микросхема К155ЛА3 – широкого применения, полупроводниковая, 155-й серии (ТТЛ), 4 элемента И-НЕ, порядковый номер в серии - 3.



7 вывод общий;  
14 – ЕП.



Корпус: 201.14-1 (DIP14).  
Нумерация выводов начинается от ключа на корпусе против часовой стрелки.

Пример: микросхема К561ЛА7 – широкого применения, полупроводниковая, 561-й серии (КМОП), 4 элемента И-НЕ, порядковый номер в серии – 7 (корпус и расположение выводов аналогичны К155ЛА3).

## Комбинационные цифровые устройства

### 4.1. Общие сведения о цифровых устройствах. Классификация цифровых устройств

Цифровые, логические, запоминающие и вспомогательные элементы могут объединяться в функциональные цифровые узлы. Цифровым узлом будем называть часть оборудования, с помощью которого может выполняться одна элементарная операция над двоичным словом или несколько.

В зависимости от типа составных элементов все узлы подразделяются на комбинационные и последовательностные. В комбинационных узлах выходное слово зависит только от комбинации входного слова и не зависит от внутреннего состояния схем, т. е. эти узлы не имеют элементов памяти. Схемы последовательностных узлов содержат элементы памяти, роль которых выполняют обычно триггеры.

К комбинационным узлам относятся преобразователи кодов, комбинационные сумматоры, схемы сравнения кодов, коммутаторы, индикаторные устройства. К узлам с памятью относятся регистры, счетчики, накапливающие сумматоры, программные датчики, цифровые фазовращатели, синтезаторы частот.

В большинстве случаев при разработке и проектировании каких-либо цифровых систем используют готовые узлы на базе интегральных микросхем. На практике часто приходится заниматься синтезом цифровых узлов в случаях принятия нестандартных решений.

### 4.2. Преобразователи кодов. Шифраторы и дешифраторы

В цифровой технике применяются различные виды кодирования информации. Так, при выполнении операций в ЭВМ обычно применяется несколько разновидностей двоичного кода (прямой, обратный, дополнительный, двоично-десятичный и т. д.). При передаче информации по линиям связи удобнее использовать другие виды кодов, позволяющие уменьшить вероятность появления ошибки или даже исправлять ее в дальнейшем. Примерами таких кодов являются коды построения по принципу 2 из 5 (в этих кодах из пяти символов два всегда имеют единичное значение), коды с проверкой четности или нечетности, коды Хемминга.

Кроме того, преобразователи кода используются для преобразования двоичного кода в код управления различными световыми индикаторами.

В связи с этим всегда стоит задача преобразования информации из одного кода в другой. Эту задачу на аппаратном уровне решают комбинационные устройства – *преобразователи кодов*.

*Преобразователем кода* называется комбинационная схема, предназначенная для изменения вида кодирования информации. Работа преобразователя кода



характеризуется таблицей истинности, ставящей в соответствие входным кодам – выходным.

В общем случае число разрядов входного и выходного кодов может не совпадать. Главное – это однозначное соответствие различных кодов. Условное графическое изображение преобразователей кодов на принципиальных электрических схемах приведено на рис. 4.1.

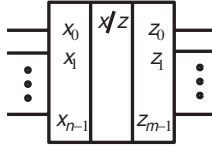


Рис. 4.1 – Условное графическое изображение преобразователя кодов

Частным случаем преобразователей кода являются шифраторы и дешифраторы.

**Шифраторы и дешифраторы.** Шифратором, или кодером, называется комбинационное логическое устройство для преобразования десятичного кода в двоичный код. Классический шифратор имеет  $m$  входов и  $n$  выходов, и при подаче на один из входов (обязательно на один, и не более) на выходе узла появляется двоичный код номера возбужденного входа. Число входов и выходов такого шифратора связано соотношением  $m=2^n$ . Такой шифратор называется полным. Если число входов шифратора меньше  $2^n$ , он называется неполным. Условное изображение шифратора показано на рис. 4.2, б. Рассмотрим работу шифратора на примере схемы 8-3 (8 входов и 3 выхода). Табличный способ задания схемы неприемлем из-за своей громоздкости. В данном случае существенно, что единица присутствует всегда только на одном из входов. Поэтому в создаваемой схеме не нужны конъюнкторы, выделяющие определенные комбинации нулей и единиц.

Схема может начинаться прямо с элементов ИЛИ – по одному на каждый выход. Ко входу элементов ИЛИ каждого выходного разряда должны быть подключены те входы шифратора, в двоичном представлении номера которых есть единица в данном разряде. Можно записать систему функций алгебры логики, описывающих работу шифратора

$$\begin{aligned} Q_0 &= x_1 + x_3 + x_5 + x_7, \\ Q_1 &= x_2 + x_3 + x_6 + x_7, \\ Q_2 &= x_4 + x_5 + x_6 + x_7. \end{aligned} \quad (4.1)$$

Логическая схема устройства, соответствующая системе (4.1), приведена на рис. 4.2, а.

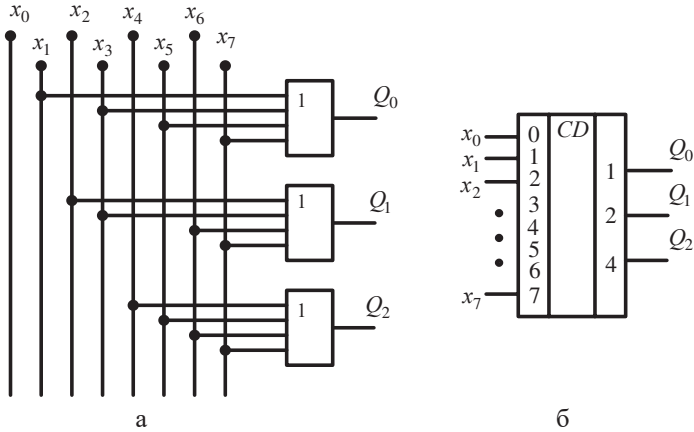


Рис. 4.2 – Функциональная схема – а; условное изображение шифратора – б

Совместно с шифратором в состав кодирующих узлов может входить схема выделения старшей единицы. Эта схема преобразует  $m$ -разрядное слово следующим образом: все старшие нули и самая старшая единица входного кода пропускаются на выход без изменения; все разряды, более младшие, чем старшая единица, заменяются нулями.

Вариант схемы выделения старшей единицы показан на рис. 4.3. Если к выходу схемы выделения старшей единицы подключить шифратор, то в сумме получится функциональный узел приоритетного шифратора, который формирует в двоичном коде номер самой старшей единицы из всех, присутствующих во входном слове.

На рис. 4.4 показано условное обозначение приоритетного 8-входового шифратора K155ИВ1. Вход разрешения  $EI = 0$  запирает не только выходы адреса  $A0, A1, A2$ , но и выходы разрешения  $E0$  и группового сигнала  $GS$ . Вход  $EI$  и выходы  $E0$  и  $GS$  служат для наращивания разрядов шифратора.

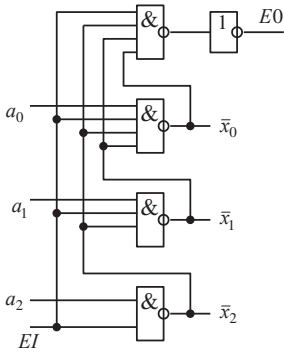


Рис. 4.3 – Схема выделения старшей единицы

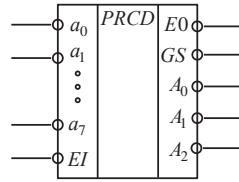


Рис. 4.4 – Микросхема приоритетного шифратора ИВ1

На рис. 4.5 показаны примеры использования групповых сигналов при объединении микросхем для наращивания разрядности.

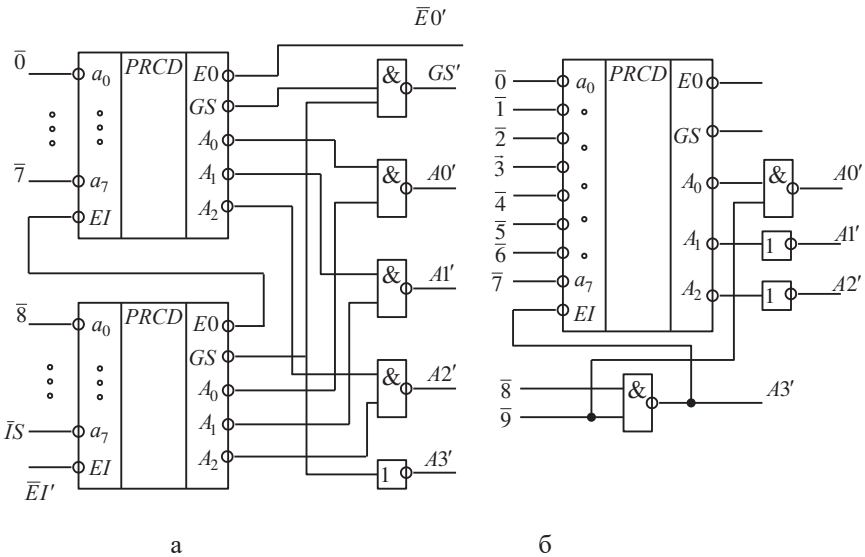


Рис. 4.5 – Наращивание разрядности микросхем приоритетных шифраторов: а – до шестнадцати входов; б – до десяти входов

Кроме кодирования состояний переключателей и номеров нажатых клавиш, приоритетные шифраторы используются для определения номера устройства, подавшего сигнал запроса на обслуживание в микропроцессорных системах, входя в состав микросхем контроллеров прерываний.

**Дешифратором** (декодером) называют кодирующее устройство, преобразующее двоичный код в унитарный. Из всех  $m$  выходов дешифратора активный уровень имеется только на одном, на том, номер которого равен поданному на вход двоичному числу. На всех остальных выходах уровни напряжения неактивные. Условное изображение дешифратора на схемах показано на рис. 4.6, а.

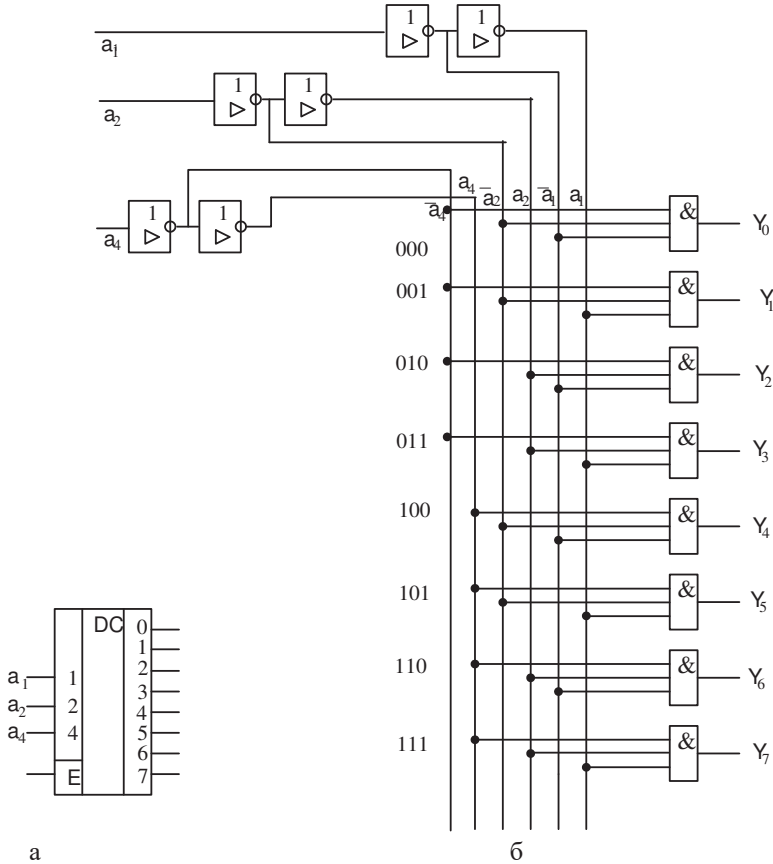


Рис. 4.6 – Дешифратор 3-8: а – условное обозначение, б – структурная схема линейного дешифратора

Если декодер имеет  $n$  входов и  $m$  выходов, то  $m=2^n$ . Такой декодер называют полным. У неполного декодера используется лишь часть возможных наборов и имеется меньшее число выходов. Декодер используют, когда нужно обращаться к различным цифровым устройствам, и при этом номер устройства – его адрес – представляется двоичным кодом. Входы декодера (их иногда называют адресными входами) часто нумеруют не порядковыми номерами, а в соот-

ветствии с весами двоичных разрядов, т. е. не 1, 2, 3, 4..., а 1, 2, 4, 8, 16... Число входов декодера указывают таким образом: декодер 3-8 (три в восемь).

Описать работу дешифратора можно с помощью системы функций алгебры логики (ФАЛ):  $Y_0 = a_4 a_2 a_1$ ;  $Y_1 = a_4 a_2 a_1$ ;  $Y_2 = a_4 a_2 a_1$ ; ...  $Y_6 = a_4 a_2 a_1$ ;  $Y_7 = a_4 a_2 a_1$ . Реализация этих восьми выражений с помощью восьми трехвходовых элементов дает простой по структуре дешифратор, называемый *линейным* (рис. 4.6, б). На входе дешифратора стоят  $2n$  буферных усилителей, обычно инвертирующих, что характерно для интегральной технологии. Назначение буферных усилителей – свести к единице кратность нагрузки, которую представляет дешифратор для источника сигнала. В противном случае каждый источник сигнала, как следует из рис. 4.6, б, будет нагружен на  $m/2$  входов элементов И. Если дешифратор состоит из элементов И–НЕ, то на его выходах будут не сами функции  $y_i$ , а их инверсии, т. е. активный уровень будет низким. Дешифраторы часто имеют разрешающий (управляющий) вход  $E$  (enable – давать возможность).

При  $E = 1$  дешифратор работает как обычно, при  $E = 0$  на всех выходах устанавливаются неактивные уровни независимо от поступившего кода адреса. Вход  $E$  часто выполняют инверсным. Дешифратор, имеющий разрешающий вход, иногда называют *декодер-демультиплексор* и на условном обозначении вместо символа DC используют символ DX.

Для наращивания разрядности дешифратора стандартные ИМС объединяются в группы, соединенные в два каскада (рис. 4.7).

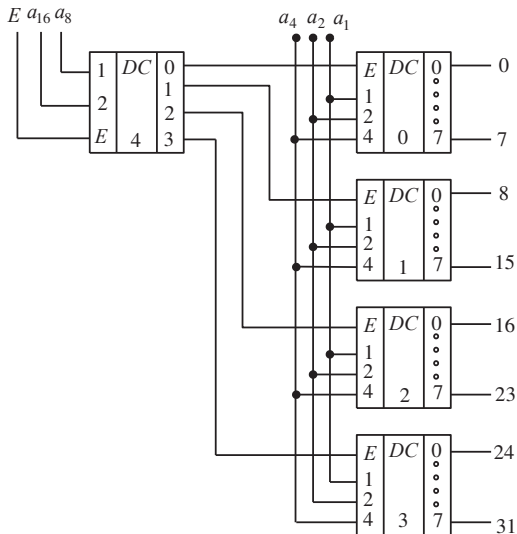


Рис. 4.7 – Каскадное соединение дешифраторов

Вся группа дешифраторов работает как дешифратор 5-32. Два старших разряда адреса  $a_{16}$  и  $a_8$  расшифровываются дешифратором 2-4 DC4, который по входу  $E$  управляет четырьмя дешифраторами 3-8 второго каскада.

### 4.3. Комбинационные сумматоры

Сумматором называется комбинационное логическое устройство, предназначенное для выполнения операции арифметического сложения чисел, представленных в виде двоичных кодов.

Сумматоры являются одним из основных узлов арифметико-логического устройства (АЛУ). Термин «сумматор» охватывает широкий спектр устройств, начиная с простейших логических схем до сложнейших цифровых узлов. Общим для всех этих устройств является арифметическое сложение чисел, представленных в двоичной форме.

#### *Классификация сумматоров*

По числу входов и выходов различают: полусумматоры, одноразрядные сумматоры, многоразрядные сумматоры.

*Полусумматором* называется устройство, предназначенное для сложения двух одноразрядных кодов, имеющее два входа и два выхода и формирующее из сигналов входных слагаемых сигналы суммы и переноса в старший разряд.

*Одноразрядным сумматором* называется устройство, предназначенное для сложения двух одноразрядных кодов, имеющее три входа и два выхода и формирующее из сигналов входных слагаемых и сигнала переноса из младших разрядов сигналы суммы и переноса в старший разряд.

*Многоразрядным сумматором* называется устройство, предназначенное для сложения двух многоразрядных кодов, формирующее на выходе код суммы и сигнал переноса в случае, если результат сложения не может быть представлен кодом, разрядности которого совпадают с разрядностью кодов слагаемых. Многоразрядные сумматоры подразделяются на последовательные и параллельные. В последовательных сумматорах операция сложения выполняется последовательно, разряд за разрядом, начиная с младшего. В параллельных все разряды входных кодов суммируются одновременно.

По способу тактирования различают синхронные сумматоры. В синхронных сумматорах время выполнения операции арифметического суммирования всегда постоянно и не зависит от вида кодов чисел. В асинхронных сумматорах время выполнения операции зависит от вида слагаемых. По завершении выполнения суммирования необходимо вырабатывать специальный сигнал завершения операции. В зависимости от используемой системы счисления различают двоичные, двоично-десятичные и другие типы сумматоров.

#### *Двоичный полусумматор*

Алгоритм выполнения арифметического сложения двух одноразрядных двоичных кодов  $x_1$  и  $x_0$  поясняется таблицей истинности (табл. 4.1).

Таблица 4.1

$X_1$	$X_0$	S	P
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Используя приведенную таблицу, легко записать систему функций алгебраической логики, описывающих алгоритм операции арифметического сложения

$$S = \overline{x_1 x_0} + x_1 \overline{x_0}, \quad (4.2)$$

$$P = x_1 x_0, \quad (4.3)$$

где S – результат сложения;

P – значение переноса в старший разряд.

С целью упрощения записи выражение (4.2) записывают следующим образом:

$$S = x_1 \oplus x_0. \quad (4.4)$$

Эту функцию называют *исключающее ИЛИ*, или *суммой по модулю два*. Условное обозначение этой функции приведено на рис. 4.8.

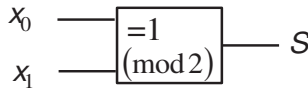


Рис. 4.8. Условное обозначение функции *исключающее ИЛИ*, или *сумма по модулю два*

Реализация функции 4.2 в базисе И, ИЛИ, НЕ показана на рис. 4.9, а. Для перехода в базис И–НЕ преобразованием выражение (4.2):

$$S = \overline{x_1 x_0} + x_1 \overline{x_0} = \overline{\overline{\overline{x_1 x_0} + x_1 \overline{x_0}}} = \overline{\overline{\overline{x_1 x_0}} \cdot \overline{\overline{x_1 \overline{x_0}}}} = \overline{\overline{x_1 x_0} \cdot x_1 \cdot x_0} = x_0 \cdot x_0 \cdot x_1 \cdot x_1 \cdot x_0 \cdot x_1. \quad (4.5)$$

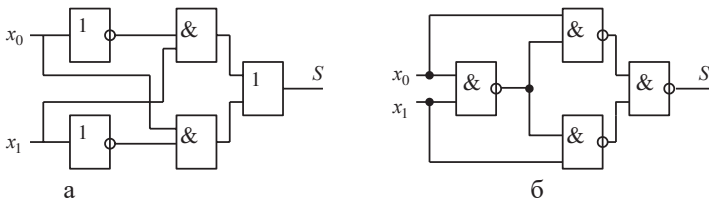


Рис. 4.9 – Структурная схема реализации функции *исключающее ИЛИ*: а – в базисе И, ИЛИ, НЕ; б – в базисе И–НЕ

Реализация полученного выражения (4.5) приведена на рис 4.10, а.

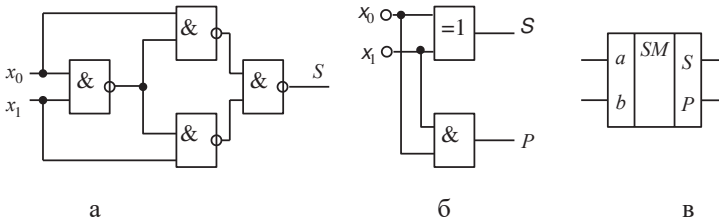


Рис. 4.10 – Структурная схема исключающее ИЛИ – а; полусумматор – б; условное обозначение полусумматора – в

Время выполнения операции исключающее ИЛИ

$$t_{BZ} = 3t_{зр},$$

где  $t_{зр}$  – время задержки распространения для элемента И–НЕ.

### Одноразрядный сумматор

Функционирование одноразрядного сумматора определяется системой ФАЛ, которая может быть получена из таблицы истинности.

Для сложения старших разрядов многоразрядных двоичных слов табл. 4.1 должна быть дополнена переменной возможного переноса из более младшего разряда  $P_{-1}$  (табл. 4.2).

Таблица 4.2

$X_1$	$X_0$	$P_{-1}$	$S$	$P$
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

ФАЛ, описывающие результаты сложения, будут иметь вид

$$S = \bar{P}_{-1}\bar{x}_0\bar{x}_1 + \bar{P}_{-1}\bar{x}_0x_1 + P_{-1}\bar{x}_0\bar{x}_1 + P_{-1}x_0x_1;$$

$$P = x_1x_0 + (x_1 + x_0)P_{-1}.$$



Техническая реализация полученных ФАЛ может быть выполнена на ЛЭ любого типа.

Рассмотрим построение одноразрядного сумматора с использованием схем двоичных полусумматоров. Для этой цели необходимо иметь два полусумматора и элемент ИЛИ (рис. 4.11, а).

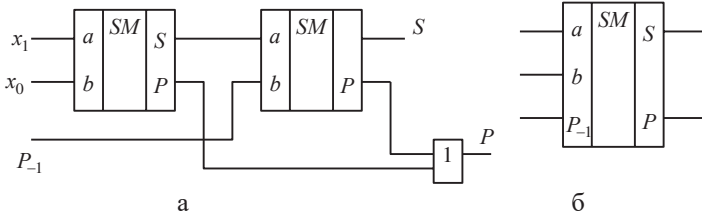


Рис. 4.11 – Одноразрядный сумматор – а; условное обозначение – б

Условное графическое обозначение одноразрядного сумматора приведено на рис 4.11, б.

### *Многоразрядный сумматор параллельного действия*

В этом сумматоре операции суммирования должны вычисляться одновременно по всем разрядам исходных двоичных чисел.

Типовая структура 4-разрядного сумматора, выполненного с использованием трех одноразрядных сумматоров и одного полусумматора, показана на рис. 4.12.

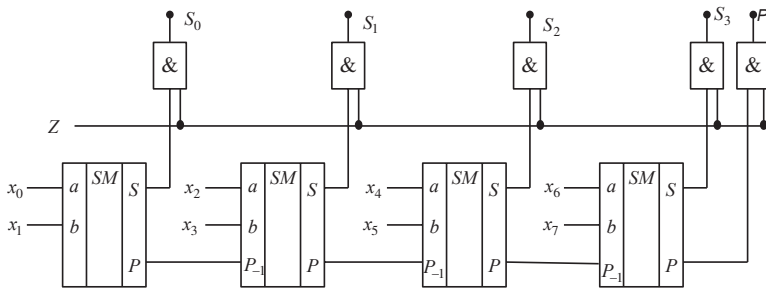


Рис. 4.12 – Структурная схема параллельного многоразрядного сумматора с последовательным переносом

Разряды кодов слагаемых подаются на соответствующие входы сумматоров, выходы суммы которых подсоединяются к первым входам логических элементов

тов И, используемых в качестве выходных ключей. На вторые входы схем И подается сигнал Z, определяющий момент считывания результата. Сигнал переноса передается последовательно, от разряда к разряду. Следовательно, независимо от того, что для суммирования в каждом разряде используется определенный сумматор, реальное время выполнения операции в схеме определяется последовательным переносом P из разряда в разряд.

Сигнал Z на входах логических элементов И должен появиться не ранее, чем после последовательной передачи сигнала переноса по всем разрядам сумматоров.

### *Сумматор на элементах И–ИЛИ–НЕ*

Сумматоры стремятся разрабатывать таким образом, чтобы задержка тракта, вход переноса  $P_{-1}$  – выход переноса P, была минимальной и при этом аппаратные затраты были бы возможно меньше.

Запишем СДНФ для S и P, воспользовавшись табл. 4.2:

$$S = \bar{P}_{-1}x_0\bar{x}_1 + \bar{P}_{-1}\bar{x}_0x_1 + P_{-1}\bar{x}_0\bar{x}_1 + P_{-1}x_0x_1; \quad (4.6)$$

$$P = \bar{P}_{-1}x_0x_1 + P_{-1}\bar{x}_0\bar{x}_1 + P_{-1}\bar{x}_0x_1 + P_{-1}x_0x_1; \quad (4.7)$$

Функция S в базисе И, ИЛИ, НЕ не минимизируется, а функцию переноса P можно представить в виде

$$P = x_1x_0 + x_0P_{-1} + x_1P_{-1}. \quad (4.8)$$

Для построения сумматора из аргументов  $P_{-1}$ ,  $x_1$ ,  $x_0$  строится функция P, которая затем используется в качестве четвертого аргумента для построения функции S:

$$\bar{P} = \overline{P_{-1}x_0 + P_{-1}x_1 + x_1x_0};$$

$$\bar{S} = \overline{P_{-1}\bar{P} + x_0\bar{P} + x_1\bar{P} + x_1x_0}. \quad (4.9)$$

Схема, реализующая (4.9), показана на рис. 4.13. Она хорошо приспособлена к реализации по ТТЛ технологии (элемента И–ИЛИ–НЕ), очень экономична по аппаратным затратам и имеет минимальную задержку в тракте переноса.

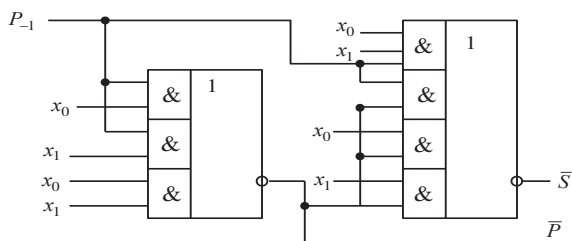


Рис. 4.13 – Одноразрядный сумматор на элементах И–ИЛИ–НЕ

Схема (рис. 5.13) применяется в спецвычислителе 9С483М1; использована в микросхемах ИМ1, ИМ2, ИМ3 серии К155, представляющих собой одноразрядные, двух- и четырехразрядные схемы сумматоров. Входы и выходы ИМ2 и ИМ3 – прямые.

Для ускорения переноса в сумматорах с большим числом разрядов применяют принцип группового переноса. Сумматор разбивают на группы, представляющие собой небольшие сумматоры с разрядностью обычно от 2 до 8. Каждый такой сумматор имеет свой штатный вход переноса  $c_i$ . Тракт группового переноса удастся построить так, что время распространения переноса в нем между группами оказывается меньше, чем если бы этот перенос распространялся по цепям внутригрупповых трактов.

Если требуется выполнить над числами не только операцию суммирования, но и другие операции, то целесообразно применять микросхемы универсальных арифметико-логических устройств (АЛУ). Они выпускаются в составе многих серий, содержат обычно по 4 разряда и хорошо приспособлены для наращивания разрядности. Подавая на управляющие входы микросхемы 5-разрядный код, можно задать одну из 32 арифметических или логических операций. В комплекте с АЛУ обычно выпускают микросхему, содержащую тракт группового переноса, что позволяет ускорить операцию суммирования при большом числе разрядов. Сами микросхемы АЛУ при этом играют роль 4-разрядных групп. Примерами микросхем АЛУ являются К155ИП3 и к ней – схема ускоренного переноса К155ИП4, а также 564ИП3 и 564ИП4. При необходимости выполнять над числами цепочки последовательных операций нужно переходить на использование микропроцессорных БИС.

#### 4.4. Компараторы

Компаратором называют функциональный узел сравнения двух двоичных чисел. Простейший компаратор формирует на выходе однобитовый сигнал равенства – 1 или неравенства – 0. Равенство двух одноразрядных чисел можно установить путем выполнения операции равнозначности, т. е.  $R = y \cdot x + \bar{x} \cdot \bar{y}$ . Такие схемы иногда называют R-схемами (рис. 4.14, а, б)

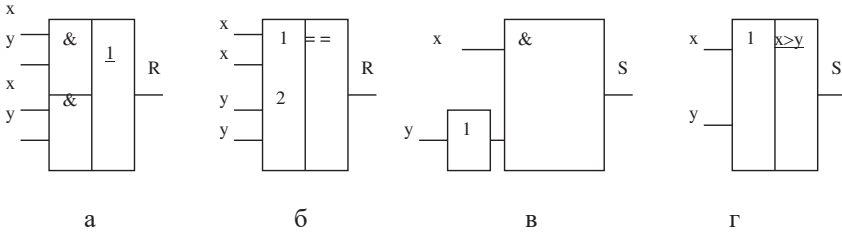


Рис. 4.14 – Схемы сравнения кодов: R-схемы – а, б; S-схемы – в, г

Если  $x > y$ , то работает так называемая S-схема, т. е.  $S = \overline{xy}$  (запрет по  $y$ ) (рис.4.14, в, г).

При равенстве многоразрядных чисел должны быть равны все их одноименные разряды. Многоразрядные R-схемы реализуются путём параллельного включения одноразрядных R-схем с объединением их выходов связью И.

Построение многоразрядных S-схем несколько сложнее, так как, кроме одноразрядных S-схем, она должна содержать дополнительные логические элементы.

S-схема сравнения кодов, при фиксированном значении одного из чисел (например,  $y$ ), выполняет операцию, называемую логическим порогом. На внешние входы такой схемы подается только одно число  $x_i$  и сравнивается с числом  $y\text{-const}$ , «заложенным» в схему при ее построении. Пороговые схемы применяются, например, в накопителях при цифровой обработке радиолокационных сигналов для повышения вероятности правильного обнаружения.

Одним из примеров повышения достоверности данных на выходе цифровых устройств является троирование аппаратуры. При этом устанавливают три одинаковых цифровых блока, на каждый из которых заводятся все входные величины. Входные сигналы всех трех блоков поступают на специальный узел мажорирования (мажоритарный элемент, кворум-элемент), который формирует из них выходной сигнал по принципу голосования «два из трех». Очевидно, что в случае отказа какого-либо одного из трех блоков сигнал на выходе мажоритарного элемента все равно остается верным. Если блок имеет несколько выходов, то мажорируется каждый из них.

Кроме простейших компараторов, которые выявляют факт равенства двух поданных на вход  $n$ -разрядных чисел  $A$  и  $B$ , существует компаратор величин, который сравнивает числа по значению. Он имеет три выхода: « $A > B$ », « $A = B$ », « $A < B$ », и в зависимости от соотношения величин  $A$  и  $B$  активный уровень появляется на одном из этих выходов.

Построить компаратор величин можно на основе сумматора, выполнив на нем операцию вычитания  $A - B$  и проанализировав результат. Для этого на сумматор нужно подать число  $B$  в инверсной форме, а на вход переноса  $CR$  сумматора подать единицу. Тогда выходной перенос будет равен 0 лишь в том случае, когда  $A$  строго меньше  $B$ , равенство суммы 0 будет признаком того, что

$A=B$ . Единица переноса при нулевой сумме говорит о том, что  $A$  строго больше  $B$ . Сказанное иллюстрируют примеры:

$$\begin{array}{r} A > B \\ -\frac{A}{B} + \frac{13}{12} + \frac{1101}{0011} \\ \hline + \quad 1 \\ \hline 1.0001 \\ \underbrace{\quad\quad\quad}_{CR=1} \quad \underbrace{\quad\quad\quad}_{S \neq 0} \end{array}$$

$$\begin{array}{r} A = B \\ -\frac{A}{B} + \frac{12}{12} + \frac{1100}{0011} \\ \hline + \quad 1 \\ \hline 1.0000 \\ \underbrace{\quad\quad\quad}_{S=0} \end{array}$$

$$\begin{array}{r} A < B \\ -\frac{A}{B} + \frac{11}{12} + \frac{1011}{0011} \\ \hline + \quad 1 \\ \hline 0.1111 \\ \underbrace{\quad\quad\quad}_{CR=0} \end{array}$$

Правила справедливы, если числа  $A$  и  $B$  рассматриваются как положительные величины, без знака. Если же их старшие разряды трактуются как знаки, то правила будут несколько иные, т. е. необходимо обращаться с обратными и дополнительными кодами. Схема, реализующая описанный алгоритм, показана на рис. 4.15, а. Ее можно перевести в любой базис и дополнить инверторами на входах числа  $B$ .

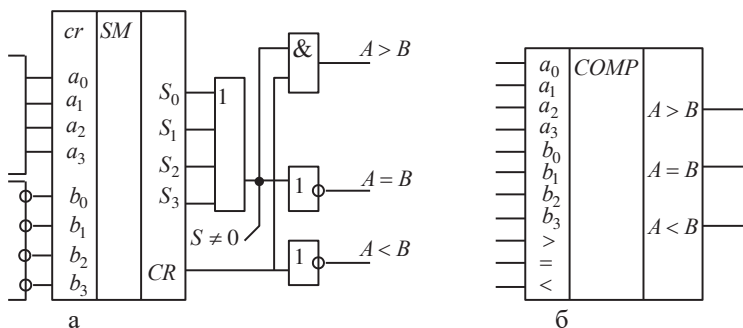


Рис. 4.15 – Компаратор величин: простейший вариант схемы – а; выводы корпуса микросхемы компаратора величин – б

В интегральном исполнении примером компараторов величин могут служить 4-разрядные микросхемы К561ИП2 и К555СП1. Они имеют встроенные инверторы – числа  $B$  и еще три входа –  $\langle > \rangle$ ,  $\langle = \rangle$ ,  $\langle < \rangle$  (рис. 4.15, б) для наращивания разрядности. Схемы наращивания К555СП1 отличаются некоторыми деталями в подключении выводов, а главное тем, что при пирамидальном наращивании, подавая разряды сравниваемых чисел не только на входы  $a_i$  и  $b_i$ , но и на входы  $\langle > \rangle$ ,  $\langle < \rangle$ , можно разрядность сравниваемых чисел увеличить с 20 до 24.

В устройствах автоматики компараторы используются для сигнализации о выходе величин за пределы допуска, в приводах следящих систем – для определения направления управляющего воздействия, ликвидирующего рассогласование, при построении счетчиков и сумматоров по произвольному основанию.

## 4.5. Мультиплексоры и демультиплексоры

Мультиплексором называется комбинированное логическое устройство, предназначенное для управляемой передачи данных от нескольких источников информации в один выходной канал.

Типовое применение мультиплексора – это передача информации от нескольких разнесенных в пространстве источников (датчиков) информации на вход одного приемника. Предположим, что измеряется температура окружающей среды в нескольких блоках РЛС, и результаты этих измерений должны быть введены в одно регистрирующее устройство, например ЭВМ. При этом, так как температура изменяется медленно, для получения достаточной точности совсем не обязательно измерять ее постоянно. Достаточно иметь информацию через некоторые фиксированные промежутки времени. Главное при этом, чтобы промежуток между двумя измерениями был существенно меньше постоянной времени, характеризующей изменение температуры в контролируемом блоке. Именно эту функцию, т. е. подключение различных источников информации к одному приемнику по заданной команде, и выполняет мультиплексор. Информацию, разнесенную в пространстве, он преобразует к виду с разделением во времени.

Согласно определению, мультиплексор должен иметь один выход и две группы входов: информационные и адресные. Код, подаваемый на адресные входы, определяет, какой из информационных входов в данный момент подключен к выходному выводу. Поскольку  $n$ -разрядный двоичный код может принимать  $2^n$  значений, то, если число адресных входов мультиплексора равно  $n$ , число его информационных входов должно равняться  $2^n$ .

Таблица истинности (табл. 4.3) отображает работу мультиплексора с двумя адресными входами.

Таблица 4.3

E	A <sub>1</sub>	A <sub>0</sub>	Q	$\bar{Q}$
1	X	X	0	1
0	0	0	D <sub>0</sub>	$\bar{D}_0$
0	0	1	D <sub>1</sub>	$\bar{D}_1$
0	1	0	D <sub>2</sub>	$\bar{D}_2$
0	1	1	D <sub>3</sub>	$\bar{D}_3$

В данной таблице учтено, что мультиплексор обычно снабжается дополнительными инверсным выходом Q и входом разрешения работы E. Если на вход разрешения работы E подан активный логический сигнал (E=1), выходной сигнал мультиплексора постоянен и не зависит от его входных сигналов. Функция алгебры логики, описывающая работу мультиплексора, имеет вид

$$Q = D_0 \bar{A}_1 \bar{A}_0 \bar{E} + D_1 \bar{A}_1 A_0 \bar{E} + D_2 A_1 \bar{A}_0 \bar{E}. \quad (4.10)$$

Логическая схема мультиплексора, соответствующая этой ФАЛ, и условное графическое обозначение мультиплексора на примере ИС типа 555КП7 показаны на рис. 4.16, а, б.

В тех случаях, когда требуется передавать на выходы многоразрядные входные данные в параллельной форме, используется параллельное включение мультиплексоров по числу разрядов передаваемых данных.

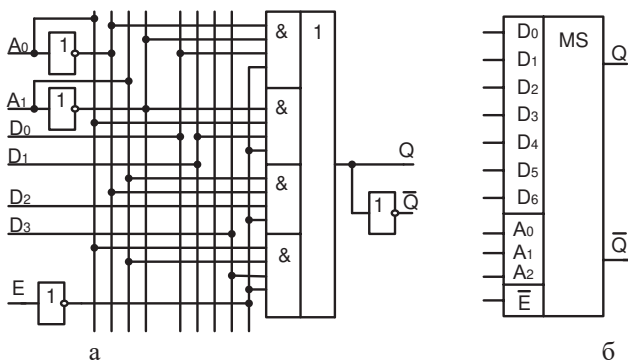


Рис. 4.16 – Логическая схема мультиплексора – а;  
условное графическое обозначение – б

Мультиплексоры могут быть использованы для синтеза логических функций. При этом число используемых в схеме элементов (корпусов интегральных микросхем) может быть уменьшено.

Логическое выражение мультиплексора (4.10) содержит члены со всеми комбинациями адресных переменных. Следовательно, если требуется синтезировать функцию трех переменных  $f(x_1, x_2, x_3)$ , то две из этих переменных (например  $x_1, x_2$ ) могут быть поданы на адресные входы  $A_1$  и  $A_0$  и третья  $x_3$  – на информационный вход.

Например, пусть требуется синтезировать функцию, заданную табл. 4.4. Логическое выражение функции

$$f(x_1, x_2, x_3) = x_1 \cdot x_2 \cdot \bar{x}_3 \vee x_1 \cdot \bar{x}_2 \cdot x_3 \vee \bar{x}_1 x_2 x_3 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3.$$

Таблица 4.4

$x_2$		$x_1$	
1	0	1	0
0	1	0	1
$x_3$			

Рассматривая переменные  $x_1$ ,  $x_2$  в качестве адресных переменных, получим табл. 4.5, из которых видно, что мультиплексор на выходе Q реализует заданную логическую функцию.

Таблица 4.5

Адресные переменные		Информационные входы	Выход
$x_1$	$x_2$		
0	0	$D_0=x_3$	$\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}$
0	1	$D_1=x_3$	$\overline{x_1} \cdot x_2 \cdot x_3$
1	0	$D_2=x_3$	$x_1 \cdot \overline{x_2} \cdot x_3$
1	1	$D_3=x_3$	$x_1 \cdot x_2 \cdot \overline{x_3}$

Принципиальная схема показана на рис. 4.17.

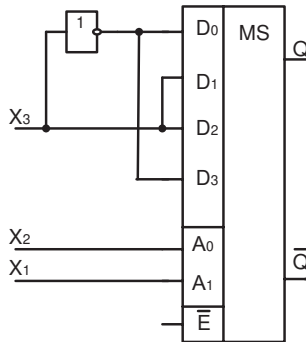


Рис. 4.17 – Реализация функции трех переменных

Очевидно, на четырехходовых мультиплексорах может быть синтезирована любая функция трех переменных, на восьмивходовых мультиплексорах – любая функция четырех переменных и т. д.

При синтезе комбинационных схем мультиплексоры могут быть использованы совместно с элементами некоторого базиса. Пусть общее число переменных функций  $n$ . Тогда, если мультиплексор имеет  $n_{\text{адр}}$  адресных входов, то на них подаются  $n_{\text{адр}}$  переменных, а на его информационные входы подаются функции  $n - n_{\text{адр}}$  переменных.

Пусть, например, требуется синтезировать логическую функцию четырех переменных с использованием четырехходового мультиплексора. Если адресными переменными являются  $x_1$ ,  $x_2$ , то на информационные входы мультиплексора должны подаваться функции переменных  $x_3$  и  $x_4$ , определяемые показанными в табл. 4.6 областями таблицы Вейча. Внутри каждой очередной для информационных входов области таблицы Вейча проводится минимизация обычными методами, после чего строятся схемы, формирующие подаваемые на ин-



формационные входы мультиплексора функции. Покажем этот прием на реализации функции, заданной табл. 4.7.

Таблица 4.6

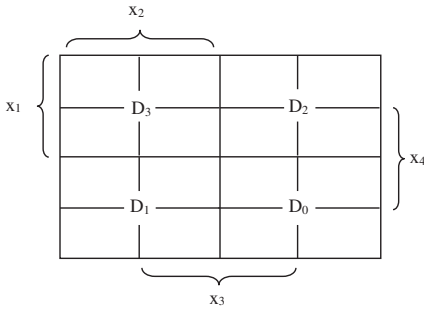


Таблица 4.7

$x_1$		$x_2$				$x_4$
		1	1	1	0	
0	0	0	0	0	$x_3$	
	0	0	1	1		
	0	0	1	1		

При подаче переменных  $x_1$  и  $x_2$  на адресные входы мультиплексора на его информационные входы должны подаваться  $D_0 = 1$ ;  $D_1 = 0$ ;  $D_2 = x_3 \cdot x_4$ ;  $D_3 = x_4$ . Реализующая заданную функцию схема показана на рис. 4.18. Синтезируя логическое устройство с использованием мультиплексора, необходимо также построить вариант схемы без использования мультиплексора. Затем сравнением полученных вариантов определить, какой из вариантов оказывается лучшим по числу используемых в схеме корпусов интегральных схем.

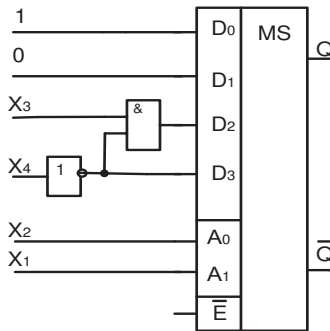


Рис. 4.18 – Схема реализации функции, заданной табл. 4.7

Максимальное число входов мультиплексоров, выполненных в виде интегральных схем, равно восьми. Если требуется построить мультиплексорное устройство с большим числом входов, можно объединить мультиплексоры в схему так называемого дерева. Такое мультиплексорное дерево, построенное на четырехвходовых мультиплексорах, показано на рис. 4.19. Схема состоит из четырех мультиплексоров первого уровня с адресными переменными  $x_1$ ,  $x_2$  и мультиплексора второго уровня с адресными переменными  $x_3$ ,  $x_4$ . Мультиплексорное устройство имеет 16 входов, разбитых на четверки, которые подключе-

ны к отдельным мультиплексорам первого уровня. Мультиплексор второго уровня, подключая к общему выходу устройства выходы отдельных мультиплексоров первого уровня, переключает четверки входов. Внутри же четверки требуемый вход выбирают мультиплексором первого уровня. По такой схеме, используя восьмивходовые мультиплексоры, можно построить мультиплексорное устройство, имеющее 64 входа.

В первом и втором уровнях мультиплексорного дерева можно использовать мультиплексоры с разным числом входов. Если в первом уровне такого дерева используются мультиплексоры с числом адресных переменных  $n_{\text{адр}1}$ , а во втором – с числом переменных  $n_{\text{адр}2}$ , то общее число входов мультиплексорного дерева будет равно  $n_{\text{инф}} = 2^{n_{\text{адр}1} + n_{\text{адр}2}}$ , а число мультиплексоров в схеме составит  $2^{n_{\text{адр}2} + 1}$ .

Мультиплексорные деревья могут использоваться не только для переключения каналов, но и для синтеза логических функций.

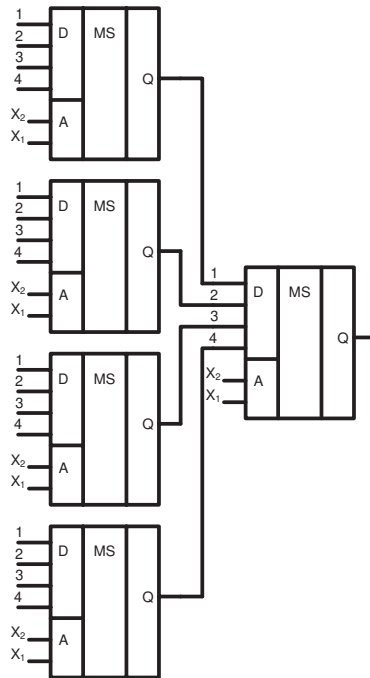


Рис. 4.19 – Мультиплексорное дерево на четырехвходовых мультиплексорах

*Демультимплексоры.* Демультимплексор имеет один вход и несколько выходов. Он представляет собой устройство, которое осуществляет коммутацию к одному из выходов, имеющему заданный адрес (номер). На рис. 4.20 показано

символическое изображение демультиплектора с четырьмя выходами. Функционирование этого демультиплектора определяется табл. 4.8.

Таблица 4.8

Адресные входы		Выходы			
$A_1$	$A_0$	$Y_0$	$Y_1$	$Y_2$	$Y_3$
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

Объединяя мультиплексор с демультиплексором, можно построить устройство, в котором по заданным адресам один из входов подключается к одному из выходов (рис. 4.21). Таким образом, может быть выполнена любая комбинация соединений входов с выходами.

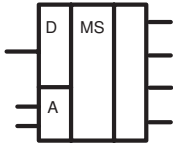


Рис. 4.20 – Демультиплексор с четырьмя выходами

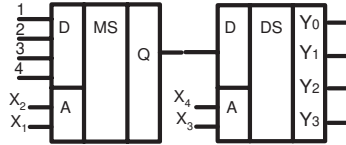


Рис. 4.21 – Объединение мультиплектора с демультиплексором

Например, при комбинации значений адресных переменных  $x_1=1$ ,  $x_2=0$ ,  $x_3=0$ ,  $x_4=0$  вход  $D_2$  окажется подключенным к выходу  $Y_0$ . Использование демультиплектора может существенно упростить построение логического устройства, имеющего несколько выходов, на которых формируются различные логические функции одних и тех же переменных.

Если на вход демультиплектора подавать константу  $D=1$ , то на выбранном в соответствии с заданным адресом выходе будет лог. 1, на остальных выходах – лог. 0. При этом демультиплексор превращается в дешифратор. При необходимости иметь большое число выходов может быть построено демультиплексорное дерево. На рис. 4.22 показано такое дерево, построенное на демультиплекторах с четырьмя выходами. Демультиплексор первого уровня подключает вход  $D$  к определенному демультиплексору второго уровня, демультиплекторы второго уровня выбирают нужный выход, куда и передается сигнал с выхода  $D$ .

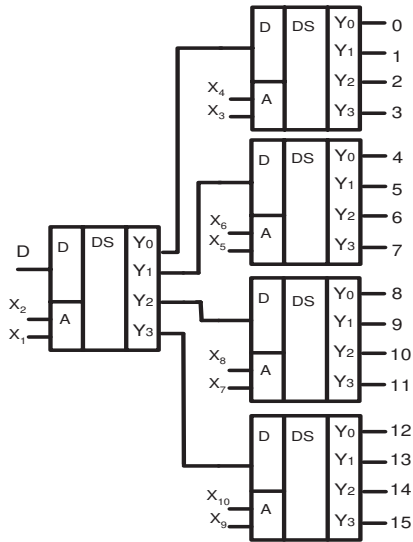


Рис. 4.22 – Демультимплексорное дерево

## Последовательностные цифровые устройства

### 5.1. Цифровые автоматы

Цифровыми автоматами, или последовательностными схемами, называют схемы, выходные сигналы которых зависят не только от значений входов в данный момент, но и от комбинации значений входных сигналов в определенные прошлые моменты времени.

В отличие от комбинационной схемы (КС), выходные сигналы которой определяются значениями входных сигналов только в настоящее время, автомат в некотором смысле помнит прошлые взаимодействия. Цифровой автомат имеет память, КС памяти не имеет. Содержимое памяти автомата определяет его внутреннее состояние. Внешнее проявление различных состояний – это различные реакции автомата на одни и те же воздействия. Обобщенная структурная схема цифрового автомата показана на рис. 5.1.

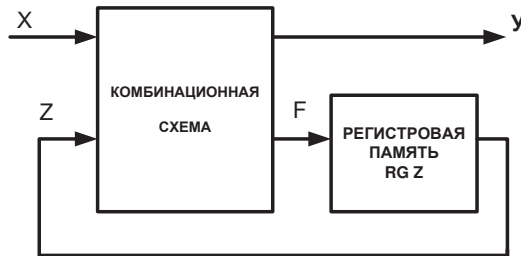


Рис. 5.1 – Структура цифрового автомата

Выходной код  $Y$ , вырабатываемый КС, есть функция не только входного кода  $X$ , но кода состояний автомата  $RGZ$ . Этот регистр и есть память автомата. Чем больше его емкость, тем богаче и разнообразнее его реакции на одни и те же входные воздействия, полнее учет прошлого опыта.

Новое состояние  $Z_{i+1}$ , в которое переходит автомат, т. е. новое содержимое  $RGZ$ , задается кодом перехода (или функцией перехода)  $F$ . Код  $F$ , так же как и выходной код  $Y$ , есть функция входных сигналов и состояния автомата непосредственно перед его переходом в новое.

Формально можно записать:

$$\begin{aligned} Z_{i+1} &= F\{x_i; z_i\}, \\ Y_i &= f\{x_i; z_i\}. \end{aligned} \quad (5.1)$$

Функция  $F$  называется функцией переходов, а  $f$  – функцией выходов.

Работа цифровых автоматов описывается с помощью таблиц переходов и выходов или графов. Задачу синтеза автомата средней сложности удобно разбить на три части: формализация задания автомата, кодирование состояний, синтез комбинационной схемы. Эти части не полностью автономны, и в процессе работы над одной из них требуется коррекция других.

**Формализация задания автомата.** Автоматы обычно строят для управления работой различных технических объектов: исполнительных механизмов, приводов, электронных устройств и вплоть до целых ЭВМ. В этой роли автоматы называют программными датчиками, блоками управления. Управляющим воздействием на объект являются выходы автомата  $Y$ . На входы автомата поступают команды оператора (вышестоящего автомата), сигналы, описывающие окружающую обстановку, сигналы о ситуациях, возникающих в управляемом объекте. Задание на автомат сначала дается в виде словесного описания закона управления. Как правило, первый вариант задания оказывается очень неполным, допускающим неоднозначное толкование. Поэтому значения сначала нужно формализовать. Чтобы проиллюстрировать характер возникающих при синтезе автомата вопросов, рассмотрим пример.

Пример. Спроектировать схему автоматического обнаружителя радиолокационных сигналов, который должен выдавать сигнал «есть цель», если на его вход поступают три отраженных от цели импульса.

Это и есть первичная, словесная формулировка задания.

Затем приступают к процессу формализации задания, который удобно разбить на три этапа.

1-й этап формализации – формирование списка (или алфавита) входных и выходных сигналов.

Вход:  $x_1$  – отсутствие сигнала;

$x_2$  – наличие входного сигнала.

Выход:  $y_1$  – отсутствие единичного сигнала;

$y_2$  – наличие единичного сигнала.

2-й этап формализации – определение требуемой задачей числа состояний автомата.

$Z_0$  – начальное состояние, ожидание первого правильного сигнала;

$Z_1$  – первый правильный сигнал получен, ожидание второго сигнала;

$Z_2$  – второй сигнал получен, ожидание третьего;

$Z_3$  – третий сигнал получен, выдается сигнал «есть цель» и автомат переводится в исходное состояние  $Z_0$ , т. е.  $Z_3 = Z_0$ .

3-й этап формализации – построение предварительного, упрощенного графика или таблицы выходов и переходов. При графовом представлении функционирования автомата состояния автомата отображаются вершинами графа, а возможные переходы из одного состояния в другое – ветвями графа, при этом направления переходов обозначаются стрелками. Ветви помечаются входными сигналами – условиями, при которых эти переходы выполняются.

Выходные сигналы, если они связаны только с определенным состоянием, изображаются внутри кружка (состояния, порождающего этот сигнал). Если кроме состояния выходной сигнал зависит еще и от входных сигналов, то он

изображается выходящей из соответствующего состояния стрелкой, помеченной поражающим его сигналом. В рассматриваемом примере граф показан на рис. 5.2.

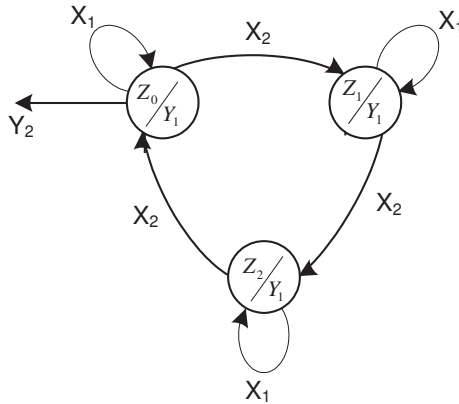


Рис. 5.2 – Граф переходов автомата

Наряду с графом строят таблицу выходов и переходов, которая показана на рис. 5.3.

	Z	Z <sub>0</sub>	Z <sub>1</sub>	Z <sub>2</sub>
X				
X <sub>1</sub>	Z <sub>0</sub> / Y <sub>1</sub>	Z <sub>0</sub> / Y <sub>1</sub>	Z <sub>1</sub> / Y <sub>1</sub>	Z <sub>2</sub> / Y <sub>1</sub>
X <sub>2</sub>	Z <sub>1</sub> / Y <sub>1</sub>	Z <sub>1</sub> / Y <sub>1</sub>	Z <sub>2</sub> / Y <sub>1</sub>	Z <sub>0</sub> / Y <sub>2</sub>

Рис. 5.3 – Таблица выходов и переходов автомата

Граф и таблица – два взаимно эквивалентных способа формального описания поведения автомата.

Описанный способ построения графа или таблицы называют этапом абстрактного синтеза.

4-й этап формализации – построение полного графа (полной) таблицы автомата. На этом этапе рассматриваются вопрос о неиспользуемых состояниях автомата, начальная установка после включения, возможность совпадения входных сигналов. Для данного примера полная таблица работы автомата приведена в табл. 5.1.

Таблица 5.1

№ п/п	Аргументы		Функции	
	Состояние в данный момент $Z_t$	Вход автом. X	Следующее состояние $Z_{t+1}$	Выход Y
1	$Z_0$	0	$Z_0$	0
2	$Z_1$	0	$Z_1$	0
3	$Z_2$	0	$Z_2$	0
4	$Z_0$	1	$Z_1$	0
5	$Z_1$	1	$Z_2$	0
6	$Z_2$	1	$Z_0$	1

**Кодирование состояний автомата.** К вопросам разработки регистровой памяти автомата принято относить выбор числа триггеров регистра и способа кодирования состояний. Минимально возможное число триггеров равно ближайшему сверху целому от двоичного числа состояний. Максимальное число триггеров равно числу состояний автомата, а каждое число состояний кодируется единицей в одном из триггеров.

Число триггеров принимаем равным двум. Тогда кодирование состояний автомата будет выглядеть следующим образом:

$$Z_0=00, Z_1=01, Z_2=10.$$

Выбираем Т-триггеры для регистра памяти.

**Синтез комбинационной части автомата.** Так как количество и тип триггеров уже выбраны, то можно составить упрощенную структурную схему автомата (рис. 5.4, табл.5.2).

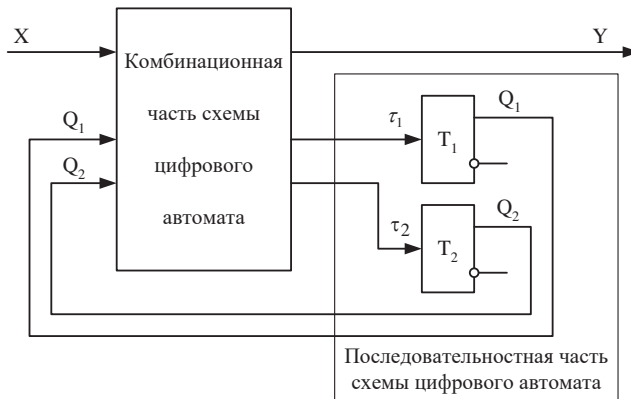


Рис. 5.4 – Упрощенная структурная схема автомата



Таблица 5.2

№ п/п	Выходы КС				Состояние $Z_{t+1}$	Выходы КС		
	Состояние $Z_t$			Входной сигнал $X$		Код перехода в следующее состояние (F)		Выход $Y$
	Имя	Код				$T_2$	$T_1$	
		$Q_2^t$	$Q_1^t$					
1	$Z_0$	0	0	0	$Z_0$	0	0	0
2	$Z_1$	0	1	0	$Z_1$	0	0	0
3	$Z_2$	1	0	0	$Z_2$	0	0	0
4	–	1	1	–	–	–	–	–
5	$Z_0$	0	0	1	$Z_1$	0	1	0
6	$Z_1$	0	1	1	$Z_2$	1	1	0
7	$Z_2$	1	0	1	$Z_0$	1	0	1
8	–	1	1	–	–	–	–	–

Из карты Карно для  $T_2$  видно (рис.5.5), что функцию можно доопределить (квадрант 111).

	$Q_2Q_1$		00	01	11	10
$X$	0					
	1		1	1	~	

а

	$Q_2Q_1$		00	01	11	10
$X$	0					
	1			1	1	1

б

Рис. 5.5 – Карты Карно для входов триггеров:  $T_1$ – а и  $T_2$ – б

Учитывая это, получим

$$T_1 = x\bar{Q}_2,$$

$$T_2 = xQ_1 + xQ_2,$$

$$Y = \bar{Q}_1 Q_2 x.$$

Построение функциональной схемы (рис. 5.6) проведем с учетом функциональных связей, показанных на рис. 5.4, и полученных выражений.

**Синтез простейшего цифрового автомата.** В теории конечных автоматов показано, что простейшим функционально полным автоматом является автомат Мура. Используя такие наборы функционально полных простейших автоматов, можно построить сколько угодно сложный автомат. Граф состояний и таблица переходов простейшего автомата Мура приведены на рис. 5.7.

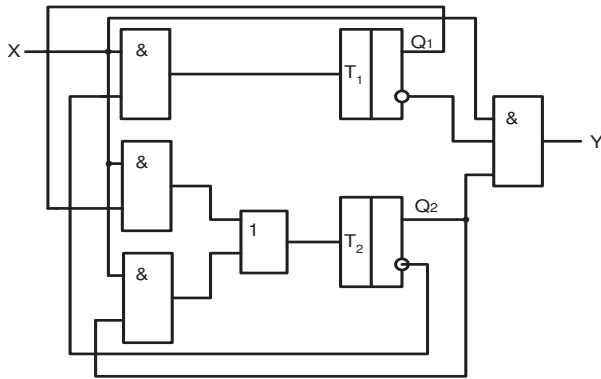
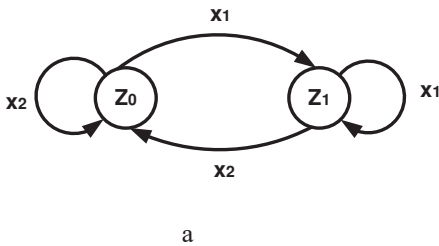


Рис. 5.6 – Функциональная схема цифрового обнаружителя



	$\frac{y}{z}$	$\frac{y_1}{z_0}$	$\frac{y_2}{z_1}$
$x$			
$x_1$		$z_1$	$z_1$
$x_2$		$z_0$	$z_0$

а

б

Рис. 5.7 – Граф состояний простейшего автомата Мура – а и таблица переходов – б

Из рис. 5.7 видно, что автомат имеет два устойчивых состояния:  $z_1$  и  $z_0$ , а выходной сигнал  $y$  определяется состоянием автомата.

Для построения простейших автоматов используются логические элементы, которые обладают функциональной полнотой (И, ИЛИ, НЕ, И–НЕ, ИЛИ–НЕ). Автоматы бывают асинхронными и синхронными.

Синтезируем простейший автомат Мура. По описанию этот автомат имеет два устойчивых состояния, два входа и два выхода. Выходы обозначают буквами  $R$ – reset и  $S$ –set, а выходы –  $Q_i$  и  $\bar{Q}_i$ . В соответствии с выражением (5.1) можно записать:

$$Q_{i+1} = F\{S, R, Q_i\}. \quad (5.2)$$

Таблица переходов и граф состояний представлены на рис. 5.8.

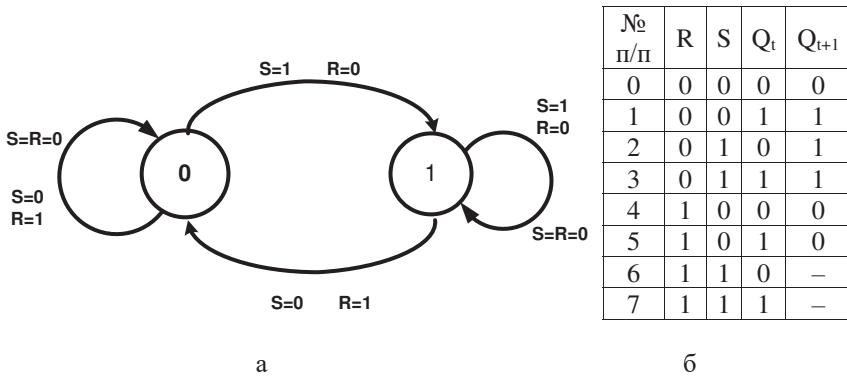


Рис. 5.8 – Граф состояний простейшего автомата Мура – а  
и таблица переходов – б

Для получения математического выражения, описывающего работу автомата, воспользуемся картами Карно.

R \ SQ <sub>t</sub>	SQ <sub>t</sub>			
	00	01	11	10
0	0	1	1	1
1	0	0	–	–

Согласно карте Карно, получим следующую СДНФ:

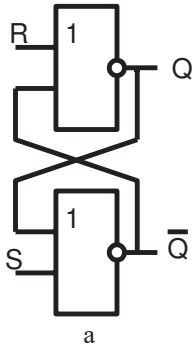
$$Q_{t+1} = \overline{R}(Q' + S). \quad (5.3)$$

Приведем полученную функцию в базис элементов Пирса (ИЛИ–НЕ), для чего воспользуемся правилом двойного отрицания

$$Q_{t+1} = \overline{\overline{\overline{Q_{t+1}}}} = \overline{\overline{\overline{R}(Q' + S)}} = \overline{\overline{R + (Q' + S)}}. \quad (5.4)$$

По выражению (5.4) составим схему элементарного цифрового автомата Мура (рис. 5.3).

Простейший автомат Мура называется асинхронным триггером с прямыми входами.



R	S	$Q_{t+1}$
0	0	$Q_t$
0	1	1
1	0	0
1	1	x

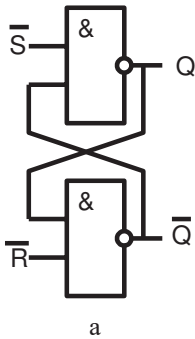
б

Рис. 5.9 – Элементарный автомат Мура на элементах Пирса – а, таблица работы – б

Аналогично можно синтезировать асинхронный триггер с инверсными входами, работа которого описывается выражением

$$Q_{t+1} = \overline{\overline{S}Q + R} = \overline{\overline{S} \cdot Q \cdot \overline{R}}. \quad (5.5)$$

Схема триггера и таблица работы приведены на рис. 5.10.



$\overline{S}$	$\overline{R}$	$Q_{t+1}$
0	0	–
0	1	1
1	0	0
1	1	$Q_t$

б

Рис. 5.10 – Схема асинхронного RS-триггера с инверсными входами – а и таблица работы – б

## 5.2. Триггеры

Триггером называется устройство, способное формировать два устойчивых состояния выходного сигнала и скачкообразно изменять эти значения под действием внешнего управляющего воздействия (сигнала). Именно эта особенность позволяет применять триггер в качестве элемента памяти.

При описании работы триггера приняты следующие соглашения:

если  $Q = 1$ , а  $\bar{Q} = 0$ , то триггер находится в состоянии установки – установлен;

если  $Q = 0$ , а  $\bar{Q} = 1$ , то триггер сброшен.

Триггеры могут быть классифицированы по различным признакам. По типу информационных входов различают RS-, D-, T-, JK-, VD-, VT-триггеры. Здесь введены следующие обозначения информационных входов:

R – (reset) сброс триггера;

S – (set) установка триггера;

K – (keep) сброс универсального триггера;

J – (Jump) установка универсального триггера.

D – (Data input) – информационный вход для записи в триггер;

T – (Toggle – переключатель) – счетный вход;

C – (clock input) – вход синхронизации.

Некоторые триггеры, кроме этих входов, могут иметь вход V (valve – клапан). Этот вход блокирует работу триггера, и он сколько угодно долго может хранить информацию.

По моменту срабатывания (реакции на входной сигнал) триггеры подразделяются на асинхронные и синхронные.

Асинхронный триггер изменяет свое состояние непосредственно в момент изменения сигнала на его информационных входах.

Синхронный триггер изменяет свое состояние только в определенные (такты) моменты времени, которые соответствуют активному сигналу на входе синхронизации C.

По виду активного логического сигнала триггеры подразделяют на статические, управляемые уровнем, и динамические, управляемые перепадом входного сигнала. Сами входы могут быть прямыми и инверсными. На рис. 5.11 показаны условные изображения прямых, инверсных, статических и динамических входов.

По принципу построения триггеры могут быть одноступенчатые и двухступенчатые. В двухступенчатых триггерах информация вначале записывается в первую ступень, а затем переписывается во вторую, и она появляется на выходе.

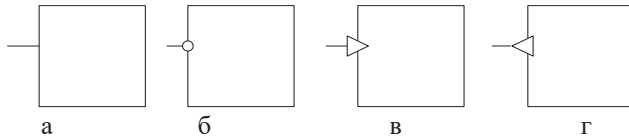


Рис. 5.11 – Условное обозначение статических входов: прямого – (а) и инверсного – (б); динамических входов: прямого – (в) и инверсного – (г)

Наиболее часто для описания работы триггера используют таблицы переходов.

### Однотупенчатые триггеры

Асинхронные RS-триггеры с прямыми и инверсными входами синтезированы и рассмотрены таблицы работы триггеров. Условное обозначение триггера с прямыми входами и временные диаграммы приведены на рис. 5.12.

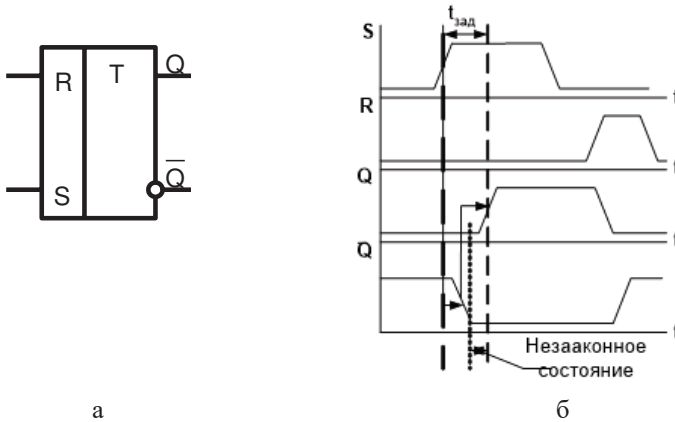


Рис. 5.12 – Условное обозначение триггера с прямыми входами – а; временные диаграммы переключения триггера – б

Из рис. 5.12, б видно, что оба элемента триггера переключаются последовательно, друг за другом.

Как видно из временной диаграммы, существуют моменты времени, когда на выходе  $Q$  и  $\bar{Q}$  уровни сигналов одинаковы. Такая комбинация сигналов является недопустимой, так как она может сформировать на выходе состояние, не предусмотренные алгоритмом работы. Для предотвращения этого явления необходимо знать время задержки распространения  $t_{зд,р}$  и минимальную дли-

тельность входных сигналов ( $R$ ,  $S$ ), при наличии которых обратная связь триггера еще не успеет замкнуться и в результате выходов триггеры вернуться в исходное состояние. Значения  $t_{зд.р}$  и минимальная длительность входных сигналов указываются в паспорте.

В RS-триггере имеется еще одна особенность, а именно: его выходы одновременно являются его входами. Поэтому если на триггер, подключенный к линии связи выходом  $Q$ , находящимся в нулевом состоянии, воздействует кратковременная помеха по выходу, то это приведет к переключению триггера, как от входного сигнала. Поэтому если триггер работает на линию, в которой возможны помехи, то ее подключают через буферные элементы, которые соединяют по схеме триггера, как показано на рис. 5.13.

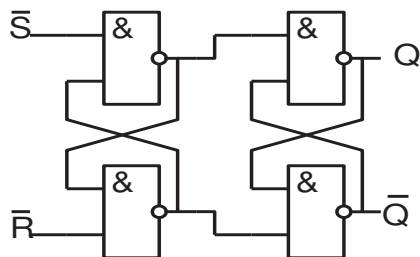


Рис. 5.13 – Схема подключения триггера Т1 к линии, в которой возможны помехи

Поскольку на выходах буферного триггера Т2 постоянно присутствует сигнал ( $\bar{R}$  или  $\bar{S}$ ), то он не сможет запомнить помеху и вернется в правильное состояние.

Триггеры можно строить на инвертирующих элементах различных типов. Подобные схемы используются тогда, когда нужно устанавливать и гасить триггер сигналами от различных источников.

Для отсечения еще не установившихся, искаженных переходными процессами, результатов между входом логической схемы и входом триггера включают конъюнктор со входом синхронизации  $C$  (рис. 5.14).

Синхронные триггеры переключаются в состояние, предписываемое управляющими входами, лишь по сигналу синхронизации, поступающему на вход синхронизации. Синхросигнал называют также синхроимпульсом,  $C$ -сигналом,  $C$ -импульсом, а синхровход –  $C$ -входом. При неактивном уровне  $C$ -сигнала синхронный триггер находится в режиме хранения и не реагирует ни на какие управляющие сигналы.





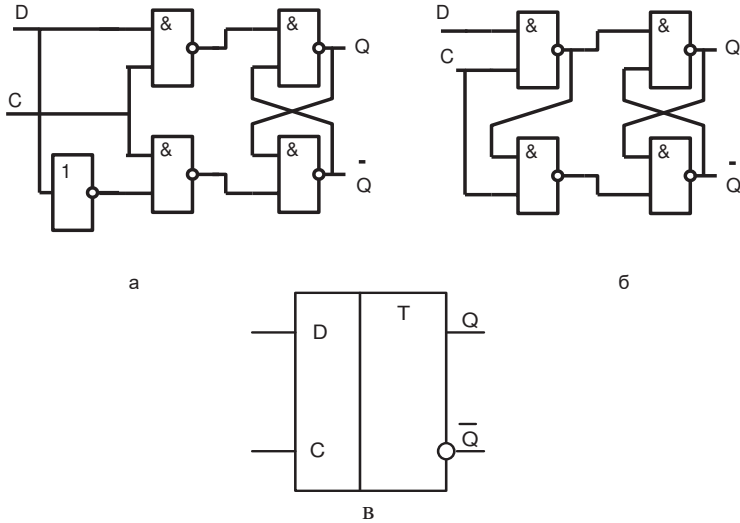


Рис. 5.15 – D-триггер типа «прозрачная защелка»: универсальный способ построения D-триггера – а; экономичный способ построения D-триггера – б; условное обозначение D-триггера – в

Таблица функционирования D-триггера приведена на рис. 5.16.

$C$	$D$	$Q_{t+1}$	$\bar{Q}_{t+1}$
1	0	0	1
1	1	1	0
0	$X$	$Q_t$	$\bar{Q}_t$

Рис. 5.16 – Таблица функционирования D-триггера

Примерами выпускаемых промышленностью D-триггеров-защелок, которые используются в ВВТ, могут служить микросхемы 136ТМ2, 133ТМ2, 133ТМ5, которые содержат по два и четыре триггера с объединенными С-входами.

### *Двухступенчатые триггеры*

На рис. 5.17 показана схема, состоящая из двух, включенных последовательно, RS-триггеров. Первый называется ведущим, или М-триггером (master), а второй – ведомым, или S-триггером (slave). Благодаря общему синхровходу С

вся схема функционирует как единое целое и называется *двухступенчатым*, или *MS-триггером* (master-slave flip-flop).

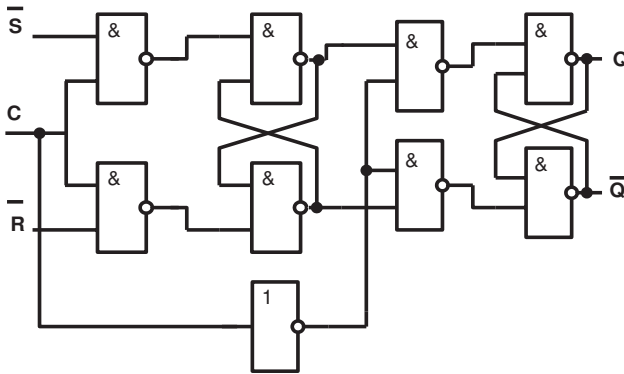


Рис. 5.17 – Двухступенчатый RS-триггер

MS-триггер, собранный по схеме (рис. 5.17), непрозрачен по управляющим сигналам R и S ни при C=0, ни при C=1. Каждая ступень его сама по себе прозрачна, но включены ступени последовательно, и какая-нибудь одна из них оказывается запертой или синхросигналом, или его отсутствием. Таким образом, в этом MS-триггере при C=1 (и тем более при C=0) никакое изменение на управляющем входе не поможет само по себе, без переключения C-сигнала, проникнуть на выход. Триггер может изменить состояние выхода только по срезу C-сигнала. В зарубежной литературе непрозрачные триггеры называют flip-flop, в отличие от прозрачных D-триггеров, за которыми укоренился термин latch.

### *JK-триггеры*

Свойство непрозрачности MS-триггера использовано для построения интересного и широко применяемого JK-триггера, схема которого показана на рис. 5.18. JK-триггер – это непрозрачный триггер, выходы которого петлями инвертирующих обратных связей (накрест) заведены на входные конъюнкторы. Внешние входы самого триггера при этом принято называть уже не R и S, а K и J.

При разноименных уровнях на J- и K-входах JK-триггер ведет себя как синхронный непрозрачный RS-триггер.

Существенным отличием от RS-триггера является поведение JK-триггера при J=K=1. Для RS-триггера такое состояние входов запрещено, а JK-триггер по срезу каждого C-сигнала меняет состояние выходов на противоположное состояние. Это так называемый счетный режим, или T-режим (от toggle – кувыраться).

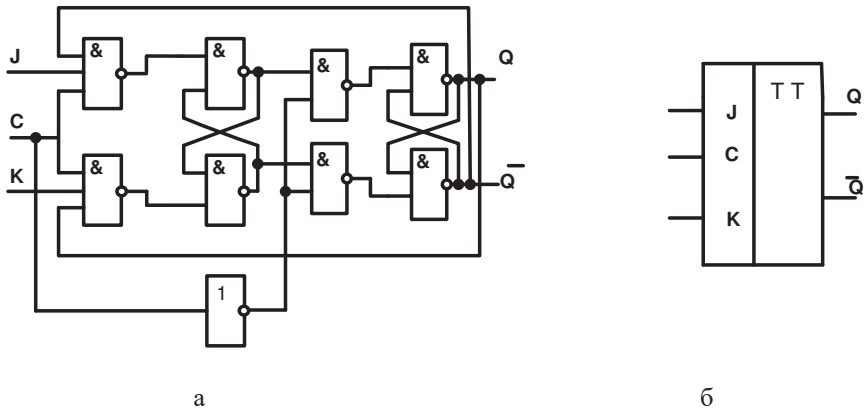


Рис. 5.18 – Двухступенчатый JK-триггер с инвертором в цепи синхронизации: а – схема; б – условное обозначение

Кратко функционирование JK-триггера описывается в табл. 5.3.

Таблица 5.3

Режим	Входы			Выходы	
	C	J	K	$Q_{t+1}$	$\overline{Q}_{t+1}$
Хранение	X	0	0	$Q_t$	$Q_t$
Сброс	1	0	1	0	1
Установка	1	1	0	1	0
Счетный (Т)	1	1	1	$\overline{Q}_t$	$Q_t$

Таблица отражает тот факт, что для JK-триггера переключающей сущностью синхроимпульса является не уровень, а перепад уровня.

Схема, близкая к показанной на рис. 5.18, а, лежит в основе триггера К155ТВ1, 133ТВ1. Эта микросхема имеет тройные конъюнктивные входы J и K. Условное обозначение показано на рис. 5.18, б. Две буквы (ТТ) указывают на наличие двух ступеней. Вход С – динамический, отрицательный перепад. Вход, реагирующий именно на перепад, иногда называют *динамическим*.

В триггерах, показанных на рис. 5.17 и 5.18, есть параллельные пути распространения сигнала: С-сигнал проходит и через М-триггер, и через инвертор, а затем оба сигнала встречаются на входах элементов 5 или 6. Это может привести к ошибочному срабатыванию схемы. Такие явления в цифровых схемах называются гонками. По этой причине двухступенчатые триггеры строят таким образом, чтобы исключить параллельный логический тракт. На рис. 5.19 показана противогоночная схема, в которой возможность ложного проникновения сигнала М-ступени в S-ступень на фронте С-импульса блокируется нулевым уровнем выхода того из входных конъюнкторов 1 или 2, который срабатывает в

данный момент. Такую схему триггера называют схемой с запрещающими связями.

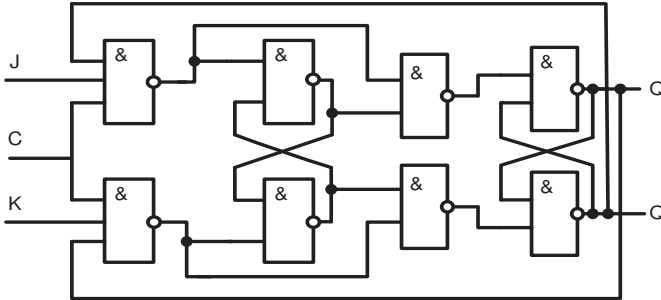


Рис. 5.19 – Двухступенчатый JK-триггер

### *Асинхронные входы триггеров*

Непрозрачные триггеры, кроме штатных входов синхровхода С и управляющих входов D, J, K, часто дополняют независимыми от них R- и S-входами. При этом входы R и S имеют приоритет в своем воздействии на триггер по отношению к штатным входам, в том числе и ко входу С. Поэтому такие входы называют асинхронными.

Типовые схемы организации асинхронных R- и S-входов показаны на рис. 5.20. Как правило, эти входы имеют активный низкий уровень.

Каждый асинхронный вход воздействует сразу на триггеры обеих ступеней и на один из входных конъюнкторов (1 или 2), пресекая их возможное противодействие при активном уровне С-сигнала.

Наличие асинхронных входов существенно расширяет возможности применения триггеров.

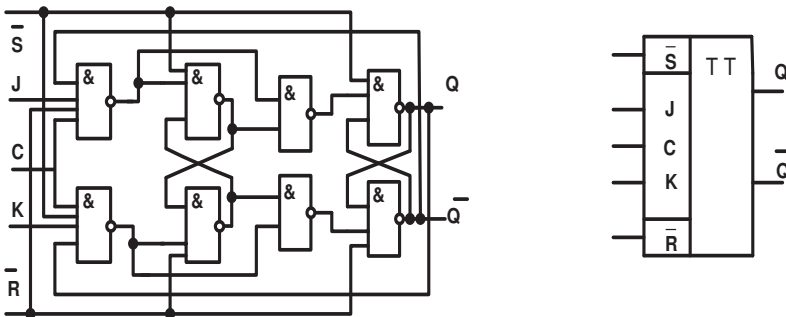


Рис. 5.20 – Асинхронные входы непрозрачного двухступенчатого JK-триггера

В качестве триггера в табл. 5.4 отражены взаимоотношения между штатными и асинхронными входами JK-триггера К155ТВ1.

Таблица 5.4

Режим	Входы					Выходы	
	S	R	C	J	K	Q	$\bar{Q}$
Асинхронная установка	Н	В	х	х	х	В	Н
Асинхронное гашение	В	Н	х	х	х	Н	В
Не определен	Н	Н	х	х	х	В	В
Загрузка 1	В	В	↓	В	Н	В	Н
Загрузка 0	В	В	↓	Н	В	Н	В
Счетный	В	В	↓	В	В	q	q
Хранение	В	В	х	Н	Н	Q	$\bar{Q}$

На основе двухступенчатого JK-триггера можно построить D-триггер (рис. 5.21, а), условное обозначение которого показано на рис. 5.21, б.

В отличие от защелки такой триггер будет непрозрачен по D-входу.

Счетным, или Т-триггером, называют JK-триггер, у которого на J- и K-выходы поданы не управляющие сигналы, а постоянные единичные уровни. В этом случае триггер будет работать в счетном режиме. Вход, вызывающий переключение триггера, называют счетным входом.

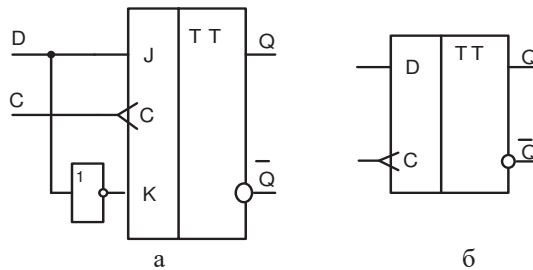


Рис. 5.21 – Использование JK-триггера в роли D-триггера – а; условное обозначение – б

Временные диаграммы работы и условное обозначение Т-триггера представлены на рис. 5.22, а, б.

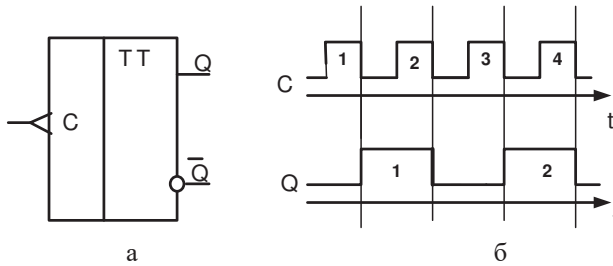


Рис. 5.22 – Использование JK-триггера в качестве Т-триггера:  
а – условное обозначение; б – временные диаграммы работы

Изменяя по срезу каждого входного сигнала состояние своего выхода, Т-триггер пересчитывает выходные сигналы на два, или, как говорят, делит частоту на два. Другая интерпретация его работы: Т-триггер суммирует по модулю два поступивший сигнал на Т-вход со своим состоянием:  

$$Q^{t+1} = Q^t \cdot \bar{T} + \bar{Q}^t \cdot T = Q^t \oplus T.$$

### 5.3. Регистры

Регистром называется последовательностное устройство, предназначенное для записи, хранения и (или) сдвига информации, представленной в виде многоразрядного двоичного кода.

В общем случае регистр может выполнять следующие микрооперации над кодовыми словами:

- 1) установка в исходное состояние;
- 2) запись информации в последовательной форме;
- 3) запись входной информации в параллельной форме;
- 4) хранение информации;
- 5) сдвиг хранимой информации вправо или влево;
- 6) выдача хранимой информации в последовательной форме;
- 7) выдача хранимой информации в параллельной форме.

Любой  $N$ -разрядный регистр состоит из  $N$  однотипных ячеек – разрядных схем, выходной сигнал каждой из которых соответствует весовым коэффициентам двоичного числа. При этом каждая разрядная схема состоит из триггера (элемента памяти) и комбинационной схемы, преобразующей входные воздействия и состояния триггера в выходные сигналы регистра. Регистры могут быть классифицированы по различным признакам. Рассмотрим основные из них. По способу приема информации регистры могут быть:

- параллельными (статическими), в которые информация записывается и считывается из них только в параллельной форме;
- последовательными (сдвигающими), в которые информация записывается и считывается из них только в последовательной форме;

- последовательно-параллельными, в которые информация записывается и считывается как в параллельной, так и в последовательной формах.

По числу каналов передачи информации регистры подразделяются на:

- парафазные, в которых информация записывается и считывается в прямом ( $Q$ ) и обратном ( $\bar{Q}$ ) кодах;

- однофазные, в которых информация записывается и считывается либо в прямом ( $Q$ ), либо в обратном ( $\bar{Q}$ ) коде.

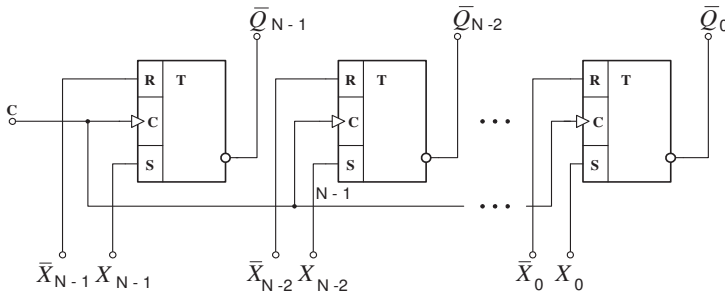
По способу тактирования регистры подразделяются на следующие:

- однотактные, управляемые одной последовательностью тактовых импульсов;

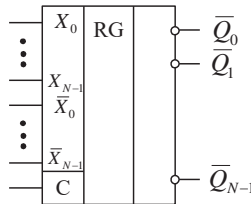
- многотактные, управляемые несколькими последовательностями импульсов.

**Параллельный регистр.** Параллельный регистр выполняет только первую, третью, четвертую и седьмую из перечисленных микроопераций над кодовыми словами, т. е. обрабатывает информацию только в параллельной форме. Поэтому образующие его разрядные схемы не связаны между собой.

Простейший параллельный регистр может быть выполнен на  $RS$ -триггерах (рис. 5.23).



а



б

Рис. 5.23 – Схема параллельного регистра – а;  
условное обозначение регистра – б

Он содержит  $N$  триггеров, входы синхронизации которых объединены между собой. На его входы  $S$  и  $R$  информация подается в прямом  $x_{N-1}, \dots, x_0$  и обратном  $\bar{x}_{N-1}, \dots, \bar{x}_0$  кодах, а с выходов  $\bar{Q}$  снимается только в обратном коде.

Согласно приведенной выше классификации, это – одноктактный регистр с парафазными входами и однофазным инверсным выходом. Запись информации выполняется за один такт синхронизации.

Если в рассмотренной структуре регистра  $RS$ -триггеры заменить  $D$ -триггерами, то получится одноктактный регистр с однофазным входом. Очевидно, что быстродействие такого регистра будет в два раза выше, чем на  $RS$ -триггерах.

Более подробно структуру параллельного регистра рассмотрим на примере ИМС 555ИР15 (рис. 5.24).

Это четырехразрядный одноктактный параллельный регистр с однофазными входами и выходами. В каждую разрядную схему входят логические элементы  $2 \times 2$  И-ИЛИ,  $D$ -триггер и выходной элемент с тремя выходными состояниями (0, 1, Z). Z-состояние – высокоимпедансное состояние, при котором выход триггера отключается от вывода  $Q$ -разрядной схемы. Этот режим необходим для объединения выходов нескольких регистров и управляется сигналами  $E_{Z1}$  и  $E_{Z2}$  (при  $E_{Z1} + E_{Z2} = 1$ , Z-элемент отключает триггер от выхода ИМС. Элемент  $2 \times 2$  И-ИЛИ является мультиплексором, управляемым одноразрядным адресным словом и обеспечивающим подключение входа  $D$ -триггера либо к входу  $D$  (запись), либо к собственному выходу  $Q$  (хранение).

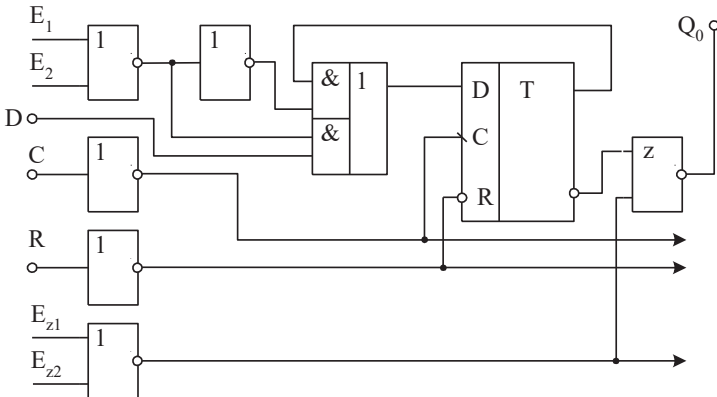


Рис. 5.24 – Фрагмент схемы параллельного регистра 555ИР15

Вход  $R$  используется для одновременного сброса триггеров всех разрядных ячеек. Условное графическое изображение регистра приведено на рис. 5.25.



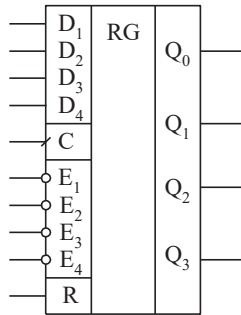


Рис. 5.25 – Условное графическое обозначение параллельного регистра 555ИР15

Используя различные входные комбинационные схемы, можно разработать разрядные схемы, реализующие необходимые алгоритмы работы параллельных регистров.

Существуют микросхемы, в которых регистр объединен с входным мультиплексором, позволяющим принимать входные данные с двух и более направлений, выбираемых сигналами на адресных входах микросхемы. Объединение регистра с выходным демультиплексором позволяет передавать содержимое регистра на различные направления.

**Сдвигающий регистр.** Схема сдвигающего четырехразрядного регистра показана на рис. 5.26.

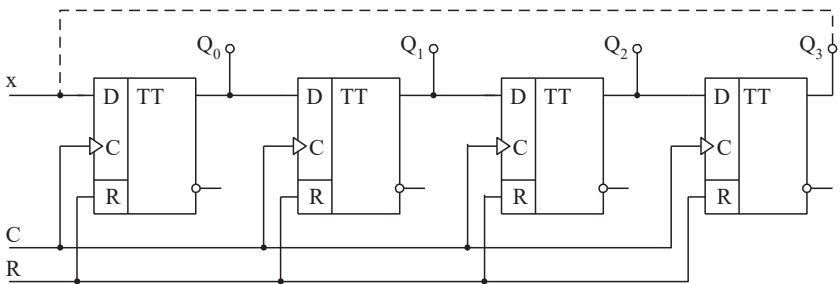


Рис. 5.26 – Схема сдвигающего регистра

Регистр имеет параллельные выходы  $Q_0$ – $Q_3$  и вход асинхронного потенциального сброса триггеров  $R$ . Значение входного сигнала  $X(t)$  появляется на выходе  $Q_3$  через четыре такта. Для последовательного ввода в  $m$ -разрядный регистр  $m$ -разрядного слова требуется  $m$  тактов синхросигнала. Такие регистры могут использоваться для преобразования последовательного кода в параллельный код. Выход  $Q_3$  используется для последовательного вывода информа-

ции из сдвигающего регистра и может быть использован для цифровой задержки информации на четыре такта. На сдвигающем регистре с последовательным выходом может быть построено запоминающее устройство с последовательным доступом к информации.

Если соединить  $Q_3$  с входом регистра, то можно получить рециркуляцию информации (режим хранения). Такой регистр называют «кольцевым».

Сдвигающие регистры с параллельной записью информации являются, как правило, универсальными и могут выполнять все доступные для регистров микрооперации. Рассмотрим построение такого регистра на примере одноктактного сдвигающего регистра с возможностью параллельной записи информации.

На рис. 5.27 показана схема универсального сдвигового регистра 133ИР1, 134ИР1. Эта схема используется в цифровых устройствах для создания местной памяти.

Микросхема представляет собой четырехразрядный сдвиговый регистр. Он имеет последовательный вход данных  $SI$ , четыре параллельных входа  $D_0 - D_3$  и четыре выхода  $Q_0 - Q_3$ . Регистр имеет два тактовых входа  $\bar{C}1, \bar{C}2$ .

Вход разрешения параллельной загрузки  $PE$  служит для выбора режима работы регистра. Отрицательные фронты от последовательности тактовых импульсов сдвигают данные, поступающие от последовательного входа  $SI$  на выход  $Q_0$ , затем на выходы  $Q_1, Q_2, Q_3$ . Таким образом, осуществляется сдвиг числа вправо. Для сдвига влево необходимо осуществлять внешнюю коммутацию, при этом перевести режим работы регистра в параллельный.

Условное графическое изображение четырехразрядного сдвигового регистра показано на рис. 5.28. Если сдвиг числа проводится влево, то стрелку ставят в обратном направлении.

Следует отметить, что в сдвигающих регистрах используются только двухступенчатые триггеры или триггеры с динамическим управлением. Это гарантирует сдвиг информации строго на один разряд по каждому импульсу синхронизации. При использовании других триггеров процесс сдвига становится неуправляемым, и за один импульс синхронизации информация может быть сдвинута на несколько разрядов.

### ***Реверсивные сдвигающие регистры***

Такие регистры предназначены для сдвига данных как в сторону младших, так и в сторону старших разрядов. Эти регистры используются, кроме ЭВМ, в цифровых устройствах слежения (цифровых дальномерах, цифровых угловых дискриминаторах и т. д.).

Примером такого универсального реверсивного регистра является ИМС К155ИР13 (133ИР13), условное графическое обозначение которого показано на рис. 5.29.

Регистр ИР13 является универсальным регистром, обеспечивающим прием, хранение, а также сдвиги восьмиразрядного слова на один разряд вправо или

влево. Режим работы регистра определяется кодом на управляющих входах  $v_0, v_1$ .

При  $v_0 = v_1 = 0$ , вход  $C$  блокируется, и регистр сохраняет свое текущее положение (хранение информации).

При  $v_0 = v_1 = 1$ , по фронту синхросигнала  $C$ , содержимое регистра сдвигается на разряд вправо, а при  $v_0 = 0, v_1 = 1$  – на один разряд влево.

Первый и последний разряды имеют дополнительные  $D$ -входы. При сдвиге влево в освобождающийся младший разряд  $Q_0$  записывается значение переменной с входа  $DSL$ , а при сдвиге вправо – с входа  $DSR$ .

Если  $v_0 = 1, v_1 = 1$ , то по фронту синхросигнала  $C$  проводится параллельная запись с входов  $D_0 - D_7$ . Асинхронный вход  $R$  служит для обнуления регистра.

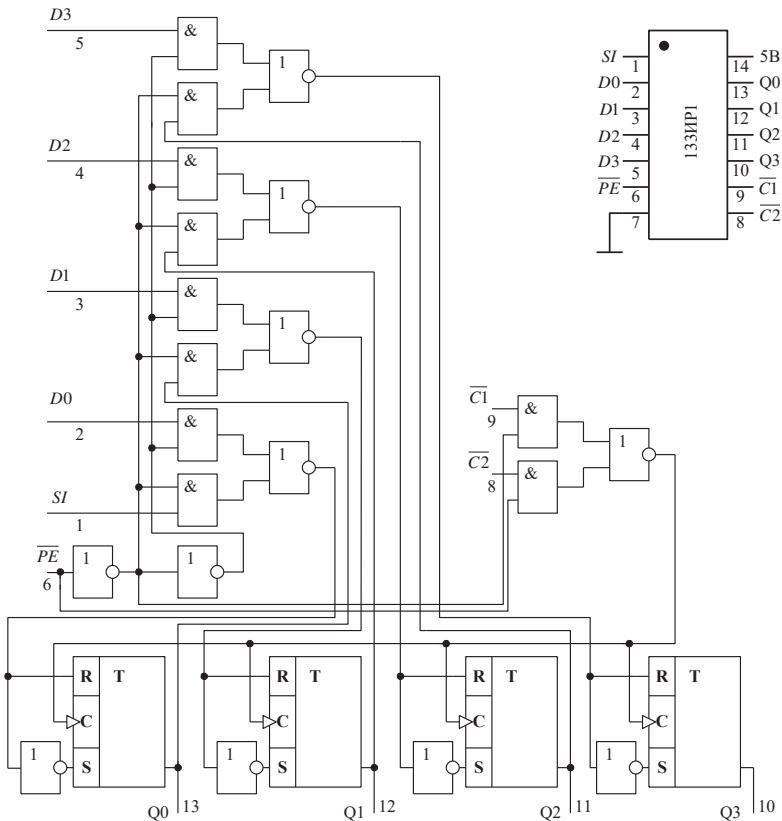


Рис. 5.27 – Схема универсального регистра 133IP1

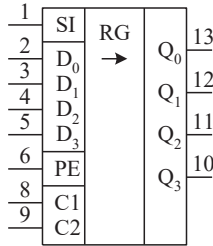


Рис. 5.28 – Сдвиговый регистр 133IP1 (134, K155IP1)

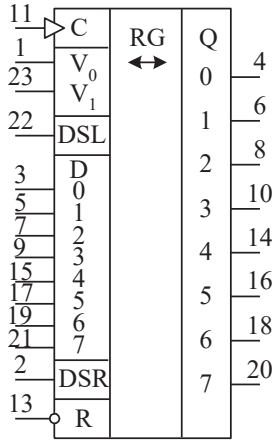


Рис. 5.29 – Условное графическое обозначение ИМС IP13

### *Организация межрегистровых связей*

В реальных цифровых устройствах информация, подлежащая обработке, хранится в регистрах. Обработка информации предполагает передачу ее из регистра в регистр. Эта передача может быть выполнена различными способами. В простейшем случае между всеми регистрами может быть организована индивидуальная связь. Однако при большом числе регистров реализация данного способа наталкивается на большие технические трудности ввиду громоздкости линий связи. В микропроцессорных системах обмен информацией между несколькими регистрами осуществляется по последовательному принципу. При этом используется единственная шина (DB – Data Bus) (рис. 5.30).

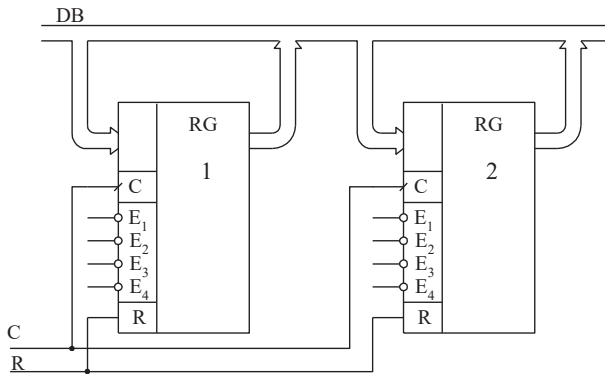


Рис. 5.30 – Обмен информацией между регистрами по общей шине

К этой шине подключены входы и выходы всех регистров, и в каждый момент времени обмен информацией возможен только между двумя регистрами. Путь передачи информации в каждый конкретный момент времени определяется значениями управляющих сигналов.

## 5.4. Счетчики

Счетчиком называется последовательное устройство, предназначенное для счета входных импульсов и фиксации их числа в двоичном коде.

Счетчики, так же как и сдвиговые регистры, строятся на основе  $N$  однотипных связанных между собой разрядных схем, каждая из которых в общем случае состоит из триггера и некоторой комбинационной схемы, предназначенной для формирования сигналов управления триггером.

В цифровых схемах счетчики могут выполнять следующие микрооперации над кодовыми словами:

установка и исходное состояние (запись нулевого кода);

запись входной информации в параллельной форме;

3) хранение информации;

4) выдача хранимой информации в параллельной форме;

5) инкремент – увеличение хранящегося кодового слова на единицу;

6) декремент – уменьшение хранящегося кодового слова на единицу.

**Основные параметры и классификация счетчиков.** Основным статическим параметром счетчика является *модуль счета*  $M$ , который характеризует максимальное число импульсов, после прихода которых счетчик устанавливается в исходное состояние.

Основным динамическим параметром, определяющим быстродействие счетчика, является *время установления* выходного кода  $t_k$ , характеризующее временной интервал между моментом подачи входного сигнала и моментом установления нового кода на выходе.

Счетчики могут классифицироваться по многим параметрам. Рассмотрим основные из них.

По значению модуля счета счетчики могут быть:

*двоичные*, модуль счета у которых равен степени числа  $2(M=2^n)$ ;

*двоично-кодированные*, у которых модуль счета может принимать любое, не равное целой степени числа 2, значение.

По направлению счета счетчики подразделяют на следующие:

*суммирующие*, выполняющие микрооперацию инкремента над хранящимся кодовым словом;

*вычитающие*, выполняющие микрооперацию декремента над хранящимся кодовым словом;

*реверсивные*, выполняющие, в зависимости от значения управляющего сигнала, над хранящимся кодовым словом либо микрооперацию декремента, либо инкремента.

По способу организации межразрядных связей счетчики могут быть с:

*последовательным переносом*, у которых переключение триггеров разрядных схем осуществляется последовательно, один за другим;

*параллельным переносом*, у которых переключение всех триггеров разрядных схем осуществляется одновременно, по сигналу синхронизации;

*комбинированным, последовательно-параллельным переносом*, у которых используются различные комбинации способов переноса.

### *Суммирующие двоичные счетчики*

В суммирующем счетчике поступление на вход очередного уровня лог. 1 (очередного импульса) вызывает увеличение на одну единицу хранимого в счетчике числа. Таким образом, в счетчике устанавливается число, которое получается путем суммирования предыдущего значения с единицей. Это суммирование проводится по обычным правилам выполнения операций сложения в двоичной системе счисления. Например,

Переносы		111
Исходное число	10110	10111
	+ <u>1</u>	+ <u>1</u>
Результат	10111	11000

Заметим, что в процессе такого суммирования имеют место следующие особенности:

1) если цифра некоторого разряда остается неизменной либо изменяется с 0 на 1, то при этом цифры старших разрядов не изменяются;

2) если цифра некоторого разряда изменяется с 1 на 0, то происходит инвертирование следующего за ним старшего разряда.

Этот принцип использован при построении схемы счетчика, представленной на рис. 5.31, а. В построении схемы имеются следующие особенности:

а) входы J и K в каждом триггере объединены, и на эти входы подан уровень лог. 1; таким образом, в каждом триггере синхронизирующий вход С является счетным входом триггера;

б) сигнал с прямого выхода триггера каждого разряда поступает на счетный вход С-триггера следующего старшего разряда, а на счетный вход триггера 1-го разряда подаются входные счетные импульсы.

Если на счетном входе С-триггера действует импульс, то его положительным фронтом переключается ведущая часть триггера, на отрицательном фронте – ведомая его часть. Итак, при каждом изменении сигнала на счетном входе с уровня лог. 1 на уровень лог. 0 изменяется на противоположное состояние выхода триггера. Таким образом, на отрицательном фронте сигнала на выходе триггера происходит переключение следующего за ним триггера старшего разряда. На рис. 5.31, б показана временная диаграмма работы данного счетчика.

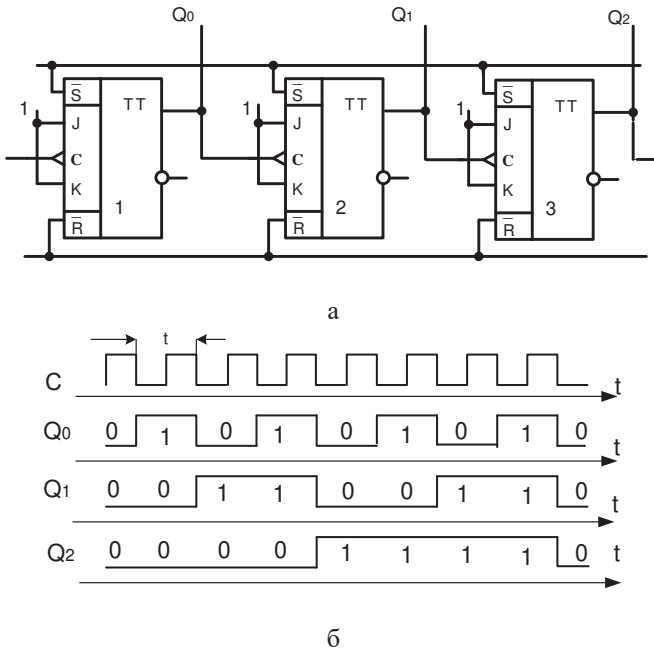


Рис. 5.31 – Схема суммирующего двоичного счетчика – а; диаграмма работы счетчика – б

С каждым входным импульсом число в счетчике увеличивается на единицу. Такое нарастание числа происходит до тех пор, пока после  $(2^n - 1)$ -го входного импульса ( $n$  – число разрядов в счетчике) не установится в счетчике двоичное число 11...1. Далее с приходом  $2^n$ -го импульса в счетчике устанавливается ис-

ходное состояние  $00\dots 0$ , после чего счет ведется сначала. Таким образом, при непрерывной подаче на вход импульсов счетчик циклически с периодом в  $2^n$  входных импульсов сбрасывается в исходное состояние.

Рассмотренная схема счетчика имеет следующие недостатки. Пусть первые  $k$  триггеров младших разрядов счетчика установлены в состояние лог. 1, и на вход счетчика поступает очередной импульс. При этом будет происходить процесс последовательного переключения триггеров. Если  $\tau$  – время переключения триггера, то относительно отрицательного фронта входного импульса первый триггер переключится с задержкой  $\tau$ ; второй триггер переключается с задержкой  $\tau$  относительно отрицательного фронта сигнала на выходе первого триггера и с задержкой  $2\tau$  относительно отрицательного фронта входного импульса и т. д. Следовательно, задержка в переключении  $k$ -го триггера составит  $k\tau$ . При большом числе разрядов задержка в переключении триггера старшего разряда может оказаться недопустимо большой. Ниже рассматриваются способы уменьшения этой задержки и, таким образом, увеличения быстродействия счетчика. Для дальнейшего рассмотрения удобнее описывать процесс суммирования с помощью логических выражений.

Пусть  $a_1, \dots, a_n$  – цифры разрядов числа до суммирования;  $c_1, \dots, c_n$  – цифры разрядов числа, получаемого в результате суммирования с единицей;  $p_i$  – значение переноса, формируемого в  $(i-1)$ -м разряде счетчика и передаваемого в  $i$ -й разряд. Функционирование  $i$ -го разряда счетчика можно описать таблицей истинности (табл. 5.5).

Таблица 5.5

$a_i$	$p_i$	$c_i$	$p_{i+1}$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Рассмотренная схема счетчика имеет следующие недостатки. Пусть первые  $k$  триггеров младших разрядов счетчика установлены в состояние лог. 1, и на вход счетчика поступает очередной импульс. При этом будет происходить процесс последовательного переключения триггеров. Если  $\tau$  – время переключения триггера, то относительно отрицательного фронта входного импульса первый триггер переключится с задержкой  $\tau$ ; второй триггер переключается с задержкой  $\tau$  относительно отрицательного фронта сигнала на выходе первого триггера и с задержкой  $2\tau$  относительно отрицательного фронта входного импульса и т. д.

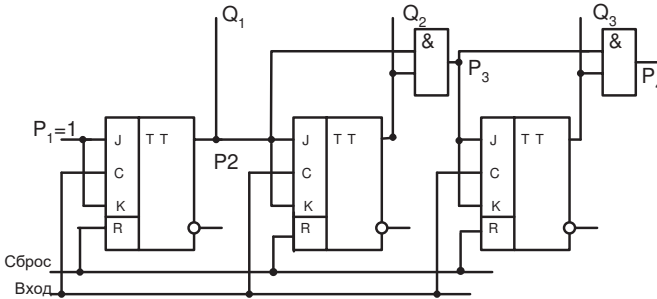
Поступление лог. 1 на вход счетчика можно представить переносом  $p_i=1$ , подаваемым на первый разряд счетчика. Из таблицы истинности следует:

а) устанавливающееся в  $i$ -м разряде значение  $c_i = a_i p_i \vee a_i p_i$  может быть получено путем подачи переноса  $p_i$  на счетный вход триггера  $i$ -го разряда, хранившего значение  $a_i$ ;

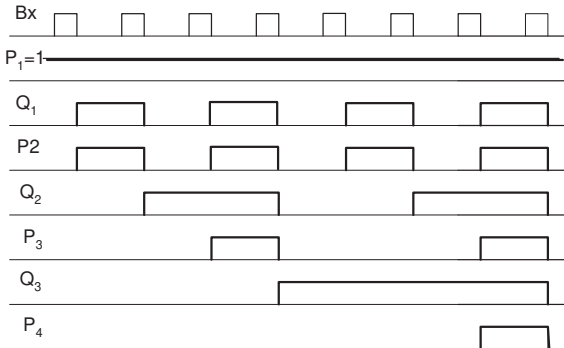


б) перенос, передаваемый в  $(i+1)$ -й разряд,  $p_{i+1} = a_i p_i$ . Так как  $p_i = 1$ , то  $p_2 = a_1$ .

Схема счетчика, в разрядах которого реализуются приведенные логические выражения, дана на рис. 5.32, а



а



б

Рис. 5.32 – Схема счетчика со сквозным переносом – а; диаграмма работы счетчика – б

. Объединенные информационные входы  $J$  и  $K$  в триггерах разрядов счетчика образуют счетные входы, на которые подаются поступающие в разряды переносы. Подлежащие счету импульсы подаются на входы синхронизации триггеров. При этом если на счетный вход триггера поступает перенос, равный лог. 1, то входной импульс переводит триггер в новое состояние. В противном случае в триггере сохраняется прежнее состояние. Для формирования переносов использованы элементы И. Цепь установки 0 используется для начальной установки в состояние 0 триггеров всех разрядов счетчика.

На рис. 5.32, б представлены временные диаграммы работы счетчика. На переднем фронте входного импульса триггер принимает поданное на его инфор-

мационные входы значение переноса, затем при спаде входного импульса на выходе триггера устанавливается новое значение. С каждым входным импульсом число в счетчике увеличивается на единицу.

В данной схеме устранен недостаток предыдущей схемы счетчика. Отрицательным фронтом входного импульса одновременно переключаются те триггеры, на входы которых в данный момент поступает сигнал переноса  $p_i = 1$ . Таким образом, может быть обеспечено более высокое быстродействие, чем в предыдущей схеме.

Фактором, ограничивающим быстродействие данной схемы счетчика, является последовательное формирование переносов.

Формирование переноса в  $i$ -м разряде начинается с момента завершения формирования переноса в предыдущем  $(i-1)$ -м разряде и поступления сформированного переноса из  $(i-1)$ -го в  $i$ -й разряд. При этом  $i$ -й разряд выдает перенос в следующий  $(i+1)$ -й разряд с задержкой относительно момента поступления переноса в  $i$ -й разряд. Эта задержка связана с задержкой распространения сигнала в элементе И. Таким образом, процесс продвижения переносов вдоль разрядов в счетчике последовательный (из разряда в разряд), и в каждом разряде имеет место задержка. Общее время задержки распространения переносов  $\tau_{\text{пер}}$  определяется задержкой в одном разряде  $\tau_{\text{пер}1}$  и числом разрядов, через которое проходят переносы.

В наихудшем случае, когда переносы последовательно проходят через все разряды счетчика (начиная со второго), общая задержка  $\tau_{\text{пер}} = \tau_{\text{пер}1}(n - 1)$ .

После переключения триггеров под действием предыдущего импульса, поданного на вход счетчика, только через время  $\tau_{\text{пер}}$  на входах триггеров оказываются сформированными переносы и может быть подан очередной импульс на вход счетчика. Таким образом, временем  $\tau_{\text{пер}}$  ограничивается минимальный временной интервал между входными импульсами и  $\tau_{\text{пер}}$  ограничивает максимальную частоту подачи импульсов на вход, т. е. быстродействие счетчика.

В тех случаях, когда необходимо обеспечить высокое быстродействие, применяют меры к уменьшению  $\tau_{\text{пер}}$ . Уменьшение  $\tau_{\text{пер}}$  может быть обеспечено применением в цепи передачи переносов элементов И с малым временем задержки распространения сигнала, а также использованием в счетчике последовательно-параллельной цепи передачи переносов. Рассмотрим принцип организации такой последовательно-параллельной передачи переносов.

Запишем логические выражения, по которым формируются переносы в разрядах счетчика

$$p_3 = a_2 \cdot p_2; \quad p_4 = a_3 \cdot p_3; \quad p_5 = a_4 \cdot p_4; \quad p_6 = a_5 \cdot p_5; \quad \dots$$

Подставляя выражения  $p_3, p_5, \dots$  в выражения  $p_4, p_6, \dots$ , получаем

$$p_3 = a_2 \cdot p_2; \quad p_4 = a_3 \cdot a_2 \cdot p_2; \quad p_5 = a_4 \cdot p_4; \quad p_6 = a_5 \cdot a_4 \cdot p_4; \quad \dots$$

При использовании этих выражений перенос  $p_2$  с выхода первого разряда одновременно поступает во второй и третий разряды, участвуя в формировании  $p_3$  и  $p_4$ . Аналогично  $p_4$  передается одновременно в четвертый и пятый разряды и участвует в формировании  $p_5$  и  $p_6$  и т. д.

Очевидно, при этом примере вдвое уменьшается число разрядов, через которые переносы передаются последовательно, уменьшается  $\tau_{пер}$ . На рис. 5.33 приведена схема счетчика, в которой цепь передачи переносов построена в соответствии с приведенными выше логическими выражениями.

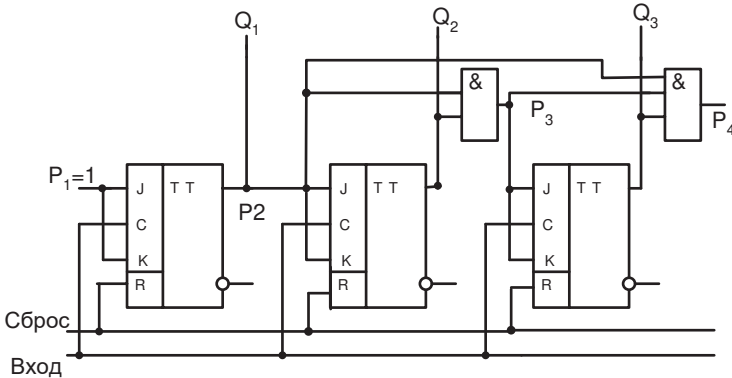


Рис. 5.33 – Схема с групповым переносом

Следует иметь в виду, что использование рассмотренного принципа повышения быстродействия вызывает усложнение схемы счетчика, связанное с использованием в цепи передачи переносов элементов И с большим числом входов.

### ***Вычитающий и реверсивный счетчики***

В вычитающем счетчике поступление на вход очередной лог. 1 (очередного импульса) вызывает уменьшение хранившегося в счетчике числа на единицу.

Примеры такого вычитания единицы.

Переносы	111	111
Исходное число	_ 11001	_ 11000
	1	1
Результат	11000	10111

Из первого примера видно, что если в младшем разряде числа содержится 1, то получающееся в результате вычитания 1 число отличается от исходного лишь в младшем разряде.

Если в младшем разряде числа содержится 0, то процесс вычитания сопровождается возникновением переносов. В отличие от операции суммирования, в которой перенос прибавляется в разряд, в который он поступает, в операции вычитания перенос имеет смысл заема из следующего, более старшего разряда

и вычитается из этого разряда. Последовательная передача таких заемов из разряда в разряд продолжается до тех пор, пока в очередном разряде, в который передается заем, не обнаруживается 1. Так, во втором из приведенных выше примеров такая 1 обнаруживается в четвертом разряде. В результате заема этой 1 в четвертом разряде образуется 0, а занятая из этого разряда 1 передается в третий разряд, где она имеет уже значение 1. Из этих двух единиц в третьем разряде остается одна, а другая передается во второй разряд, где она также приобретает значение 2 и т. д.

В результате вычитания часть числа левее первого из разрядов, содержащих 1, остается неизменной, цифры остальных разрядов инвертируются.

Функционирование  $i$ -го разряда счетчика при выполнении операции вычитания единицы представлено в табл. 5.6.

Таблица 5.6

$a_i$	$p_i$	$c_i$	$p_{i+1}$
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Из этой таблицы истинности следуют логические выражения

$$c_i = \overline{a_i} \cdot p_i \vee a_i \cdot \overline{p_i}; \quad p_{i+1} = \overline{a_i} \cdot p_i.$$

Цифры разрядов  $c_i$  определяются тем же логическим выражением, что и в суммирующем счетчике. Следовательно, как и в суммирующем счетчике, перенос должен подаваться на счетный вход, образованный соединением информационных входов J и K триггера. Отличие выражения  $p_{i+1}$  (по сравнению с соответствующим выражением суммирующего счетчика) состоит в том, что вместо  $a_i$  использовано  $\overline{a_i}$ . Таким образом, в вычитающем счетчике на элементы И, формирующие переносы, подаются сигналы с инверсных выходов триггеров.

На рис. 5.34 показана схема вычитающего счетчика с последовательной передачей переносов.

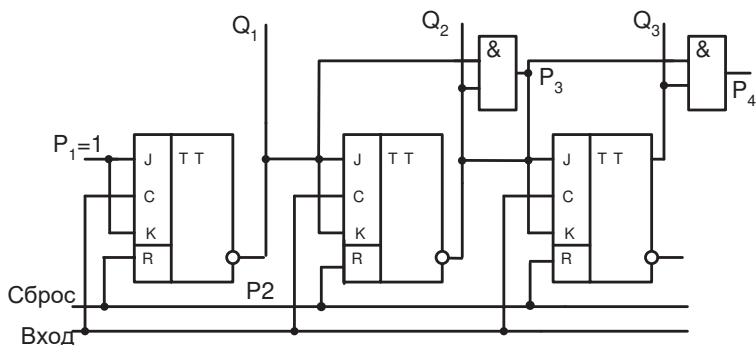


Рис. 5.34 – Схема вычитающего счетчика с последовательной передачей переносов

Для повышения скорости работы счетчика могут быть использованы последовательно-параллельные цепи передачи переносов. Вычитающий счетчик, как и суммирующий, имеет период циклической работы, равный  $2^n$  импульсов.

*Реверсивный* счетчик – счетчик, допускающий в процессе работы переключение из режима суммирования в режим вычитания, и наоборот. На рис. 5.35 приведена схема такого счетчика. В ней предусмотрены две цепи передачи переносов, одна из которых соответствует схеме суммирующего счетчика, другая – схеме вычитающего счетчика. Управляющие сигналы  $I_1$  и  $I_2$  включают в работу одну или другую цепь.

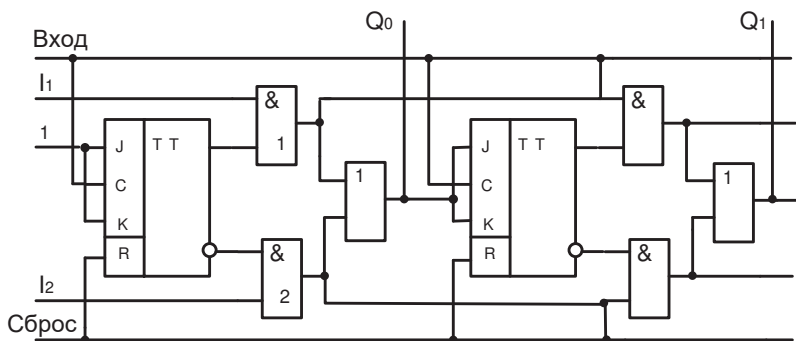


Рис. 5.35 – Схема реверсивного счетчика

При  $I_1 = 1$  и  $I_2 = 0$  оказывается закрытым элемент  $I_2$  и, следовательно, отключена цепь передачи переносов режима вычитания. Счетчик работает в режиме суммирования. При  $I_1 = 0$  и  $I_2 = 1$  закрыт элемент  $I_1$  и отключена, таким

образом, цепь передачи переносов суммирования, счетчик работает в режиме вычитания.

### *Счетчик с периодом циклической работы, не выраженным целой степенью двух*

Пусть счетчик должен иметь период циклической работы, равный  $M$ , причем  $M$  не представляется целой степенью двух. Необходимое число триггеров определяется как минимальное  $n$ , удовлетворяющее неравенству  $2^n \geq M$ .

Счетчик с таким числом триггеров может иметь период  $2^n$ , больший требуемого числа  $M$ . Поэтому после установления в счетчике числа  $M-1$  необходимо в следующем такте работы обеспечивать сброс счетчика в нулевое состояние.

Покажем метод синтеза такого счетчика. Пусть требуется синтезировать счетчик с периодом  $M = 3$ . Число триггеров  $n = 2$  (это минимальное значение, удовлетворяющее неравенству  $2^n \geq M$ ). На рис. 5.36, а представлена незаконченная схема счетчика без указания способа включения информационных входов триггеров  $J_1, K_1$  и  $J_2, K_2$ .

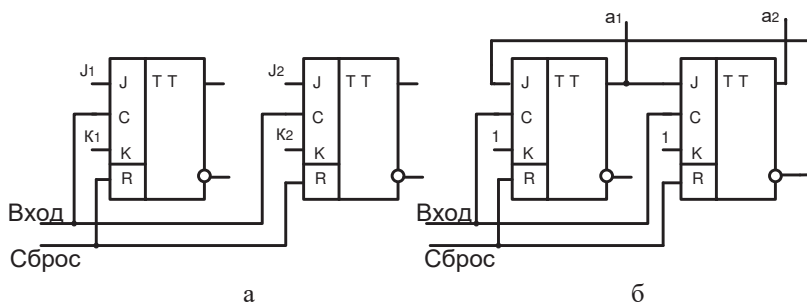


Рис. 5.36 – Схемы счетчиков с произвольным модулем счета:  
а – незаконченная схема; б – полная схема

Рассмотрим метод, позволяющий определить, каким образом должны включаться информационные входы триггеров. Под действием входных импульсов счетчик переходит из одного состояния (с одной комбинацией состояний триггеров) в другое (с другой комбинацией состояний триггеров). Комбинация состояний триггеров определяет двоичное число, значение которого при переходе счетчика в новое состояние увеличивается на единицу или устанавливается равным нулю после достижения максимального значения  $M-1$ . Такие переходы счетчика с периодом цикла  $M = 3$  показаны в табл. 5.7. Переход счетчика в новое состояние связан с переключением триггеров. Для перевода триггеров в требуемое состояние необходимы на его входах определенные логические уровни. В табл. 5.8 показаны все возможные переходы состояния триггера и требуемые для этих переходов логические уровни на входах J и K. Знак « $\leftrightarrow$ »

означает, что логический уровень на входе может быть произвольным (лог. 0 или 1).

Таблица 5.7

Номер выходного импульса	Состояние триггеров			
	текущее		следующее	
	a <sub>2</sub>	a <sub>1</sub>	a <sub>2</sub>	a <sub>1</sub>
1	0	0	0	1
2	0	1	1	0
3	1	0	0	0
4	0	0	0	1
...	...	...	...	...

Таблица 5.8

Вид пере- хода триггера	Логические уровни на входах	
	J	K
0→0	0	–
0→1	1	–
1→0	–	1
1→1	–	0

Пользуясь этой таблицей, легко построить таблицы истинности для входов J и K всех триггеров счетчика. При этом логические уровни на входах J и K являются функциями текущего состояния и на картах Вейча под a<sub>2</sub> и a<sub>1</sub> понимается состояние триггеров перед поступлением на вход счетчика очередного импульса. По картам Вейча могут быть получены следующие минимальные логические выражения для информационных входов триггеров: J<sub>1</sub> = a<sub>2</sub>; K<sub>1</sub> = 1; J<sub>2</sub> = a<sub>1</sub>; K<sub>2</sub> = 1. Полученные логические выражения определяют способ включения входов J и K триггеров, показанный на рис. 5.36, б.

Построим счетчик с периодом цикла M = 5. В таком счетчике используется три триггера. Изменение состояния триггеров под действием входных импульсов показано в табл. 5.9.

Таблица 5.9

Номер входного им- пульса	Состояние триггеров					
	текущее			следующее		
	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>
1	0	0	0	0	0	1
2	0	0	1	0	1	0
3	0	1	0	0	1	1
4	0	1	1	1	0	0
5	1	0	0	0	0	0
6	0	0	0	0	0	1
...	...	...	...	...	...	...

Используя карты Вейча, получим значения для информационных входов триггеров

$$J_1 = a_3; \quad K_1 = 1; \quad J_2 = a_1; \quad K_2 = a_1; \quad J_3 = a_1 \square a_2; \quad K_3 = 1.$$

На рис. 5.37 показана схема счетчика.

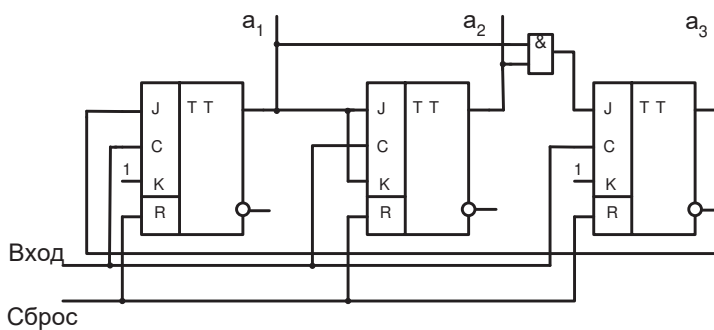


Рис. 5.37 – Схема счетчика с модулем счета, равным 5.

### *Десятичный счетчик*

Десятичные счетчики находят широкое применение в тех случаях, когда число поступающих импульсов необходимо представлять в привычной для человека десятичной системе счисления.

На рис. 5.38 показана структура десятичного счетчика. Каждый десятичный разряд такого счетчика – декада – представляет собой двоичный счетчик с периодом цикла, равным  $N = 10$ .

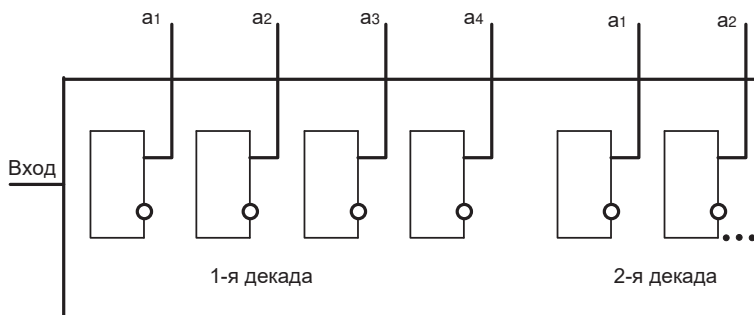


Рис. 5.38. Структура десятичного счетчика

Работа счетчика представлена таблицей переходов (табл. 5.10). Схема десятичного счетчика показана на рис. 5.39.



Таблица 5.10

Номер входного им- пульса	Состояние триггера							
	текущее				следующее			
	a <sub>4</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>4</sub>	a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>
1	0	0	0	0	0	0	0	1
2	0	0	0	1	0	0	1	0
3	0	0	1	0	0	0	1	1
4	0	0	1	1	0	1	0	0
5	0	1	0	0	0	1	0	1
6	0	1	0	1	0	1	1	0
7	0	1	1	0	0	1	1	1
8	0	1	1	1	1	0	0	0
9	1	0	0	0	1	0	0	1
10	1	0	0	1	0	0	0	0
11	0	0	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...

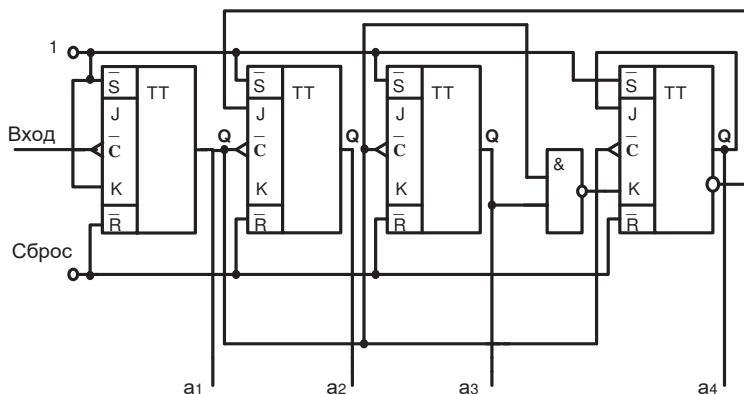


Рис. 5.39 – Схема одной декады десятичного счетчика

По такой схеме выполнена микросхема счетчика ИЕ-2.

## 5.5. Накапливающие сумматоры

При построении ЭВМ (АЛУ в частности), наряду с комбинационными сумматорами, широко применяются сумматоры накапливающего типа. По сравнению с комбинационными сумматорами они обладают рядом особенностей. К основным особенностям накапливающих сумматоров можно отнести следующие.

1. Накапливающие сумматоры строятся на основе запоминающих элементов, как правило, триггеров.

2. Слагаемые вводятся поочередно (чаще всего параллельным кодом), что позволяет суммировать не два, а любое число слагаемых. Сумма накапливается в разрядах сумматора.

3. Накапливающие сумматоры обладают меньшим быстродействием по сравнению с комбинационными, но их объединение в общую схему позволяет сочетать положительные свойства каждой разновидности.

Простейшим накапливающим сумматором является Т-триггер, который можно рассматривать как одноразрядный сумматор с одним входом ОС-1. В нем выполняется логическая операция суммирования по mod 2 для слагаемых, поступающих на один вход поочередно.

Практические схемы накапливающих сумматоров, кроме триггеров, содержат комбинационную часть. Многоразрядный накапливающий сумматор состоит из одноразрядных, связанных между собой цепями переноса и управления.

В качестве примера рассмотрим одноразрядный накапливающий сумматор ОС-3.

В качестве элемента памяти используется Т-триггер. Перед началом работы триггер устанавливается в нулевое состояние сигналом «СБРОС».

Слагаемые X и Y и сигнал переноса с предыдущего разряда (Z) подаются на входы поочередно: сначала X (момент t), затем Y (момент t+1) и, наконец, Z (t+2). При этом триггер устанавливается в состояние, при котором

$$S = Q = X \oplus Y \oplus Z.$$

Это же логическое выражение, но в канонической форме, можно выразить (получить) из таблицы работы комбинационного сумматора ОС-3 (колонка 5, табл. 5.11). Объединение X, Y, Z на счетном входе триггера осуществляется с помощью элемента 3 ИЛИ.

Таблица 5.11

X	Y	Z	P	S	T
1	2	3	4	5	6
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	1	0	1
1	0	0	0	1	1
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	1	1

Это можно увидеть, составив в таблице дополнительную колонку (6) для сигнала возбуждения Т-триггера и записать в СКНФ:  $T = X + Y + Z$ . Схема формирования сигнала суммы  $S$  будет выглядеть, как показано на рис. 5.40.

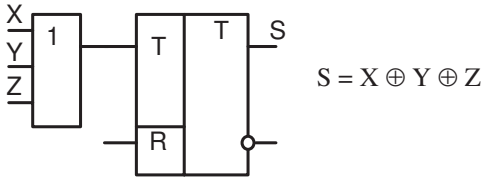


Рис. 5.40 – Схема формирования сигнала суммы

Для построения комбинационной части автомата, формирующей сигнал переноса  $P$  (табл. 5.11, колонка 4), запишем СДНФ:

$$P = \overline{X}YZ + X\overline{Y}Z + XY\overline{Z} + XYZ.$$

Обозначим:  $P = P1 + P2 = (\overline{X}YZ + X\overline{Y}Z) + (XY\overline{Z} + XYZ)$ .

Преобразуем:  $P1 = Z \cdot (\overline{X}Y + X\overline{Y}) = Z \cdot (X + Y) = Z \cdot S_{t+1}$ .

$$P2 = X_t \cdot Y_{t+1}.$$

Тогда схема переноса может быть реализована так, как показано на рис. 5.41, а, а общая схема ОС-3 – на рис. 5.41, б.

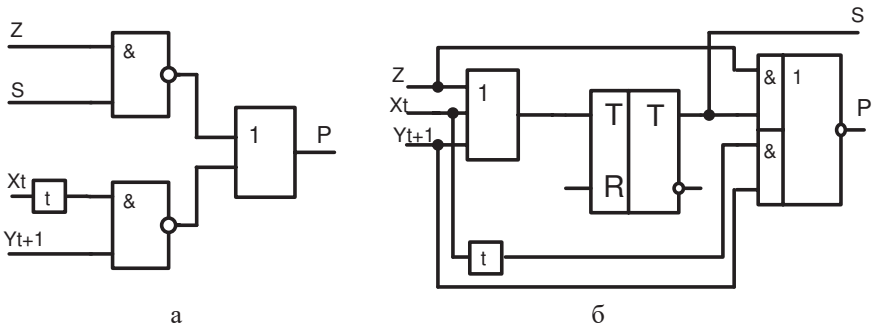


Рис. 5.41 – Схема переноса – а; сумматор со схемой переноса – б

Элемент  $t$  обеспечивает задержку первого слагаемого (сигнала  $X$ ) на время интервала между сигналами  $X$  и  $Y$  для совпадения на элементе  $\&_2$ .

## 5.6. Программные датчики

Одной из разновидностей узлов, входящих в состав процессоров ЭВМ общего и специального назначения, являются узлы синхронизации. Основным представителем узлов синхронизации современных ЭВМ является так называемый программный датчик.

Программный датчик – это цифровой узел, вырабатывающий последовательность управляющих сигналов (импульсов) для обеспечения рабочего цикла ЭВМ или другого устройства, например, большинства систем РЛС.

Датчик состоит из двух основных частей: распределителя сигналов (импульсов) и формирователя выходной последовательности сигналов (импульсов).

Распределитель имеет конечное число состояний и под воздействием тактовых импульсов вырабатывает серию сигналов, распределенных в пространстве (то есть действующих на разных выходах) и во времени (на каждом выходе – в свое время).

Формирователь из этих сигналов создает последовательности выходных сигналов заданного вида.

Простейшим распределителем является кольцевой регистр (сдвигающий регистр, выход старшего разряда которого соединен с входом младшего разряда), в котором под воздействием тактовых импульсов циркулирует единица, записанная в один из его разрядов (рис. 5.42). Единичный сигнал в такой схеме поочередно появляется на выходах каждого разряда, «передвигаясь» от младшего разряда к старшему разряду (рис. 5.43).

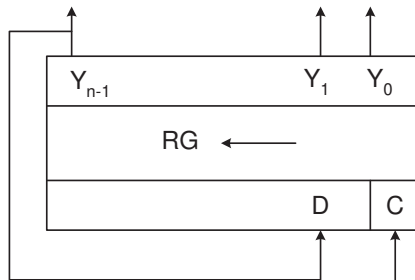


Рис. 5.42 – Схема кольцевого регистра

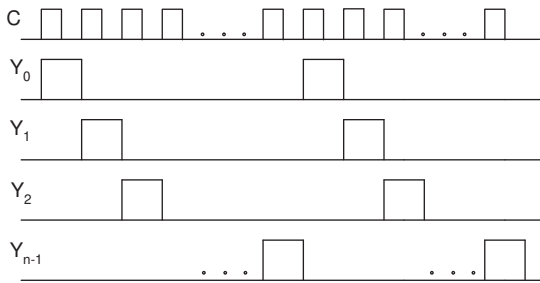


Рис. 5.43 – Диаграмма работы кольцевого регистра

Схема, обеспечивающая циркуляцию в регистре единственной единицы (рис. 5.44) в цепи обратной связи, имеет элемент ИЛИ–НЕ, на выходе которого единичный сигнал появляется только тогда, когда регистр полностью обнулен. Единица по очередной команде «СДВИГ» вписывается в младший разряд регистра. При этом сигнал обратной связи становится равным 0 и остается нулевым, пока в процессе дальнейшего сдвига единица не выйдет за пределы старшего разряда. После этого цикл будет повторяться.

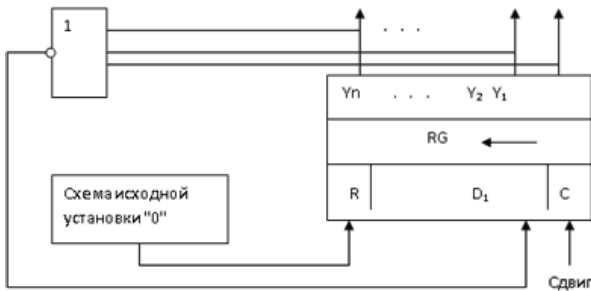


Рис. 5.44 – Схема простейшего программного датчика

Если не принято специальных мер, то при подаче на схему питания регистр может установиться в произвольное состояние (в нескольких разрядах единицы). Это приведет к тому, что первый цикл (до полного обнуления регистра) окажется нерабочим. Когда подобная потеря времени недопустима, схема дополняется элементами, обеспечивающими автоматическое обнуление регистра в момент включения.

Распределители могут строиться и по другим схемам. В качестве распределителей часто используются счетчики, выходные сигналы которых также являются распределенными в пространстве и времени.

Формирователь представляет собой комбинационную схему (дешифратор), структура которой зависит от вида требуемой последовательности.. Существует

множество схем формирователей, выполненных на двоичных счетчиках. В этом случае возможно одновременное формирование последовательностей с различными по своей длительности рабочими циклами.

Большая часть схем программных датчиков может быть реализована на программируемых делителях в интегральном исполнении, которые обычно создаются на основе микросхем двоичных счетчиков. Особенность делителя в том, что он имеет один выход. Коэффициент деления  $K_{\text{дел}}$  может иметь постоянное или переменное, то есть программируемое, значение. Делители с переменным  $K_{\text{дел}}$  (ДПКД) могут быть построены по различным схемотехническим вариантам. Чаще всего применяется вариант с предварительной установкой исходного состояния, начиная от которого счет ведется до переполнения счетчика. Возможен также вариант с установкой заданного промежуточного состояния, до которого, начиная от нулевого, ведется счет входных импульсов, а затем результат сбрасывается в нуль и начинается новый счетный цикл.

Пример ДПКД, построенного по первому варианту, приведен на рис. 5.45, а.

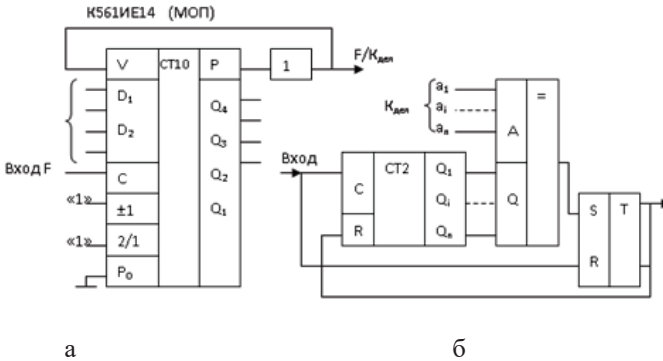


Рис. 5.45 – Программные датчики на основе счетчиков:  
а – с предварительной установкой; б – с промежуточной установкой

На входы предварительной установки микросхемы счетчика K561IE14 подается код, соответствующий числу «лишних» состояний, равных величине  $2^n - K_{\text{дел}}$ . Выход сигнала переноса P соединен через инвертор со входом V. Под воздействием входных импульсов счетчик последовательно проходит состояния от исходного, предварительно установленного, до конечного, при котором он заполняется единицами и формирует на выходе P сигнал переноса. Этим сигналом разрешается запись в счетчик исходного кода и цикл повторяется.

Вариант ДПКД (рис. 5.45, б) предполагает использование компаратора, на входы которого поданы выходной код счетчика и опорный код, соответствующий  $K_{\text{дел}}$ . При достижении счетчиком состояния, код которого равен опорному, компаратор формирует сигнал высокого уровня, переводит RS-триггер в

состояние «1», что обеспечивает установку счетчика в нулевое состояние. Дополнительный триггер в ДПКД необходим для исключения возможности сбоя из-за разброса временных параметров разрядов счетчика. Если такой опасности нет, то триггер можно исключить.

Известно, что некоторые микросхемы счетчиков имеют встроенные элементы И: К155ИЕ1, К155ИЕ2, К155ИЕ4, К155ИЕ5, К555ИЕ5 и другие. Как уже было ранее отмечено, на их основе можно строить делители без дополнительных элементов только с помощью обратных связей.

Многие делители, входящие в серии К176, К561, 564, строятся на основе регистра сдвига с перекрестной обратной связью (рис. 5.46).

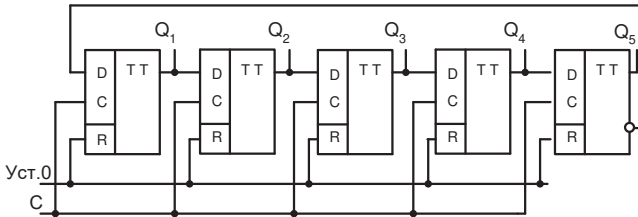
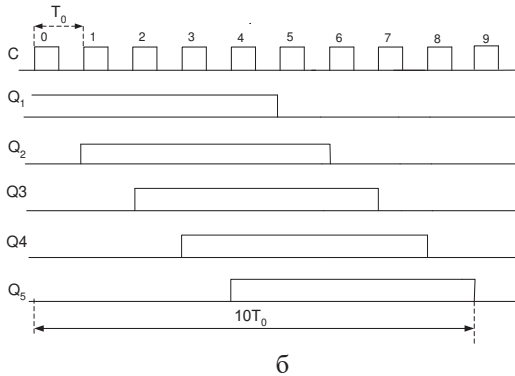


Рис. 5.46 – Счетчик Джонсона

Временные диаграммы для пятиразрядного регистра (рис. 5.47) иллюстрируют особенности работы такого делителя: первые пять тактов он заполняется единицами, в последующие пять – нулями, то есть цикл смены возможных состояний равен десяти тактам. Следовательно,  $K_{дел}$  равен удвоенному числу разрядов регистра. Такие счетчики часто называют счетчиками Джонсона.

Из временных диаграмм видно, что сигналы на всех выходах имеют одинаковый период повторения, равный  $10T_0$ , где  $T_0$  – период повторения входных сигналов. Однако они сдвинуты относительно друг друга на  $T_0$ .



б

Рис. 5.47 – Диаграмма работы счетчика

По рассмотренной схеме реализованы микросхемы К561ИЕ19, 564ИЕ19 (рис. 5.48).

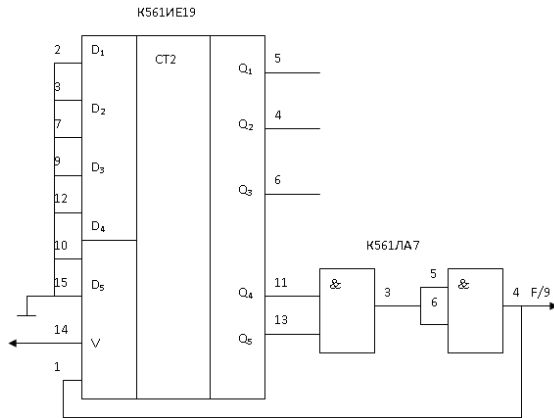


Рис. 5.48 – Программный датчик на микросхеме К561ИЕ19

Перекрестная обратная связь должна быть выполнена внешним соединением выходов с входом DI по различным вариантам, указанным в табл. 5.12.

Таблица 5.12

Способ соединения выводов ИС	Соединение входа DI с выходом счетчика без использования элемента И					Соединение входа DI с выходом элемента И, входы которого соединены с выходами счетчика			
	5	4	6	11	13	4, 5	4, 6	6, 1	11, 1
Номер выхода									
$K_{дел}$	2	4	6	18	10	3	5	1 7	3 9

На указанных микросхемах можно построить делитель и по способу предварительной установки исходного состояния.



## Аналого-цифровые и цифро-аналоговые преобразователи

### 6.1. Аналого-цифровые преобразователи

Аналого-цифровыми преобразователями (АЦП) называют устройства, преобразующие аналоговые величины в цифровые (т. е. непрерывный сигнал в импульсный).

В общем случае процедура аналого-цифрового преобразования непрерывных сигналов представляет собой превращение непрерывной функции  $f(t)$ , описывающей аналоговый сигнал, в последовательность чисел  $N(\Delta t k)$ , где  $k=0, 1, \dots, K$ ;  $\Delta t$  – интервал времени, определяемый при дискретизации сигнала (рис. 6.1).

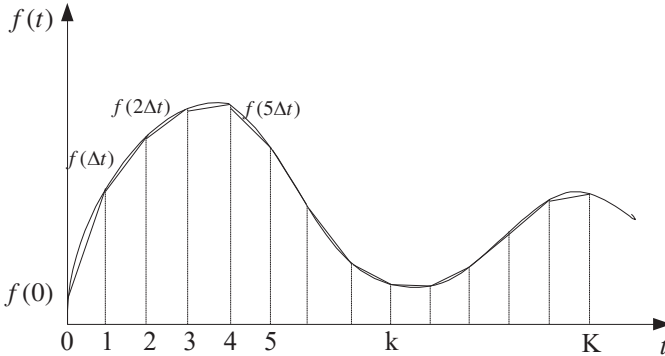


Рис. 6.1 – Преобразование непрерывного сигнала в последовательность чисел

При преобразовании сигналов из аналоговой формы в цифровую можно выделить 3 основных операции: *дискретизацию, квантование и кодирование.*

*Дискретизация непрерывных сигналов.* Из непрерывного сигнала выбирают отдельные значения, соответствующие моментам времени, следующим через некоторый интервал времени  $\Delta t = T$ , называемый тактовым интервалом времени. Если сигнал представляет собой меняющееся во времени напряжение, то  $u(t) = u(kT)$ , где  $k$  – номер моментов времени, в которые берутся отсчеты, а интервал дискретизации выбирается по теореме Котельникова:

$$\Delta t = \frac{1}{2F_{\max}}, \quad (6.1)$$

где  $F_{\max}$  – максимальная частота спектра сигнала.

Сущность дискретизации по Котельникову заключается в том, что отсчеты выбираются такими, чтобы можно было, из дискретного сигнала, восстановить исходный сигнал.

*Квантование и кодирование.*

Сущность этих операций заключается в следующем. Создается сетка так называемых уровней квантования, сдвинутых друг относительно друга на величину шага квантования  $\Delta U$  (рис. 6.2). Шаг квантования выбирается из условия минимально допустимой ошибки преобразования

$$\delta_{\text{пр}} \leq \left| \frac{\Delta U}{2} \right|. \quad (6.2)$$

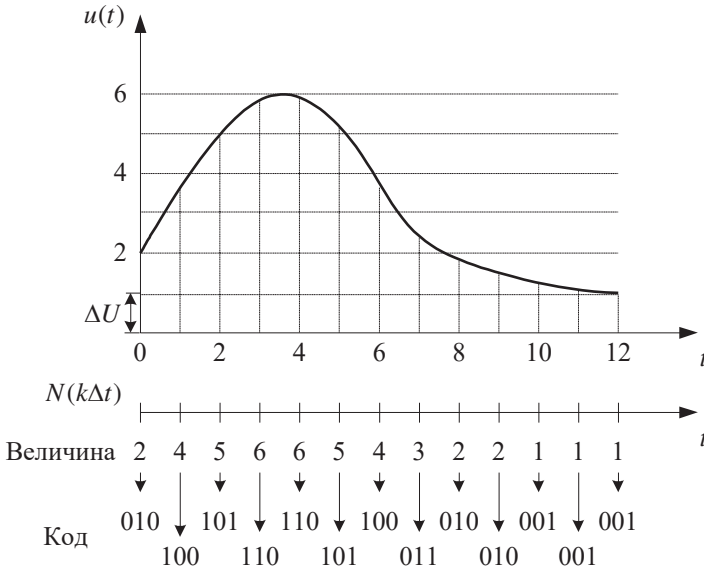


Рис. 6.2 – Квантование и кодирование

Затем переходят к кодированию – округляют значение напряжения в дискретные моменты времени и представляют эти значения номерами соответствующих уровней квантования, т.е. числами, которые представляются двоичным кодом (рис. 6.2). Дискретизация по времени и квантование по уровню может выполняться как раздельно, так и совместно.

Уменьшение шага квантования приводит к уменьшению погрешности, однако ведет к увеличению числа уровней квантования  $L$ , а это увеличивает число разрядов двоичного кода.

Если уровни квантования выражаются восьмиразрядным двоичным числом,

то количество уровней квантования оказывается равным  $L=2^8=256$ , в общем случае  $L=2^n$ , где  $n$  – количество разрядов двоичного числа.

Наряду с погрешностями квантования, возникают аппаратные погрешности, связанные с неточностью работы отдельных узлов АЦП и ЦАП. Общая погрешность может быть рассчитана следующим образом.

Считая, что ошибка в пределах  $-\frac{\Delta U}{2} \leq \delta_{\text{КВ}} \leq \frac{\Delta U}{2}$  распределена равномерно, то среднеквадратическая ошибка будет равна

$$\sigma_{\delta_{\text{КВ}}} = \frac{\Delta U}{3\sqrt{2}}, \quad (6.3)$$

а общая ошибка

$$\sigma_{\text{ПР}} = \sqrt{\sigma_{\delta_{\text{КВ}}}^2 + \sigma_{\text{А}}^2},$$

где  $\sigma_{\text{А}}$  – аппаратная СКО.

Существуют различные типы АЦП. Классификация устройств по принципу построения и методу преобразования сигналов показана на рис. 6.3, кроме того АЦП бывают прямого и косвенного действия.

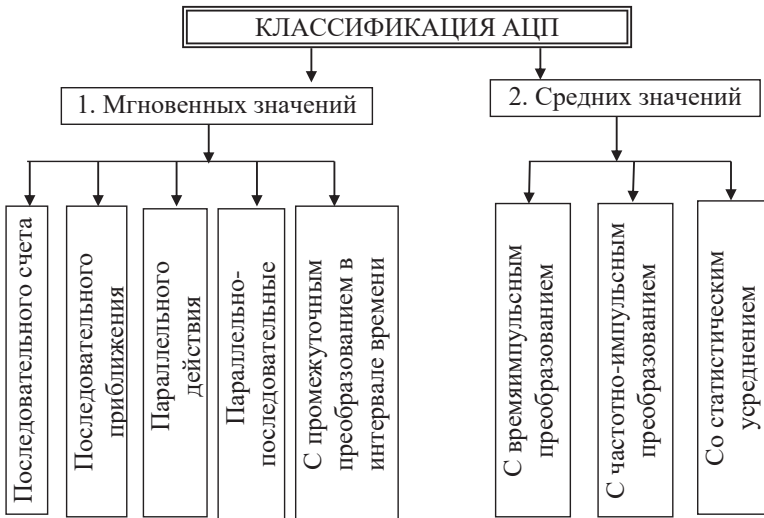


Рис. 6.3 – Классификация АЦП

АЦП мгновенных значений. АЦП последовательного счета показан на рис. 6.4.

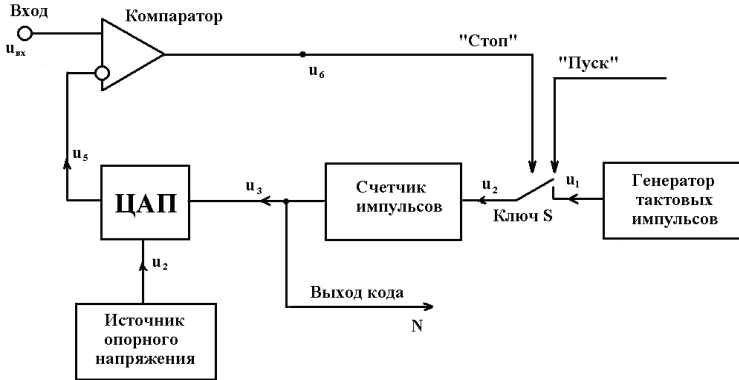


Рис. 6.4 – АЦП последовательного счета

Графики, иллюстрирующие процесс преобразования напряжения в цифровой код, приведены на рис. 6.5.

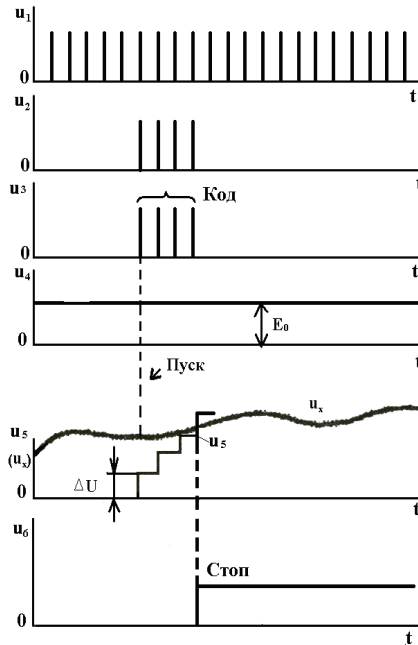


Рис. 6.5 – Графики преобразования напряжения в цифровой код

АЦП содержит компаратор, при помощи которого выполняется сравнение входного напряжения с напряжением обратной связи. На прямой вход компа-

ратора поступает входной сигнал  $U_{вх}$ , а на инвертирующий - напряжение обратной связи.

Работа преобразователя начинается с приходом импульса «ПУСК» от схемы управления, который замыкает ключ  $S$ . Через замкнутый ключ  $S$  импульсы  $U_1$  от генератора тактовых импульсов поступают на счетчик, который управляет работой цифро-аналогового преобразователя (ЦАП).

В результате последовательного увеличения выходного кода счетчика  $N$  происходит последовательно-ступенчатое увеличение выходного напряжения  $U_5$  ЦАП. Когда выходное напряжение ЦАП сравняется с входным напряжением, произойдет переключение компаратора, и по его выходному сигналу «СТОП» разомкнется ключ  $S$ . В результате импульсы от генератора перестанут поступать на вход счетчика. Выходной код, соответствующий равенству  $U_{вх} = U_5$ , снимается с выходного регистра счетчика.

Из этих графиков следует, что время преобразования зависит от уровня входного напряжения.

При числе двоичных разрядов счетчика, равном  $n$ , и периоде следования счётных импульсов  $T$  максимальное время преобразования можно определить по формуле

$$T_{пр} = (2^n - 1)T.$$

Уравнение преобразования АЦП последовательного счета можно записать в виде

$$K\Delta U = U_{вх},$$

где  $0 \leq K \leq n$  – число ступеней до момента сравнения;

$\Delta U$  – шаг квантования.

АЦП последовательного приближения (рис. 6.6) ИМС АЦП: К572ПВ1А.

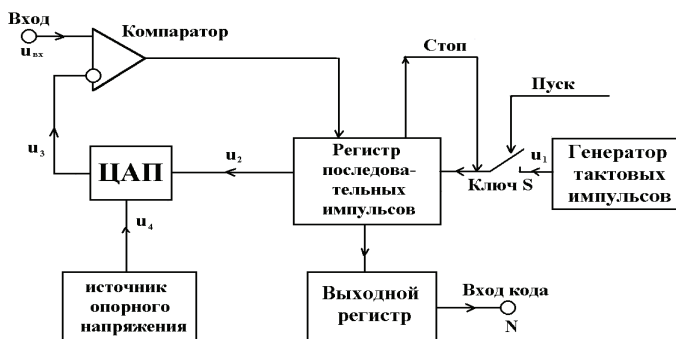


Рис. 6.6 – АЦП последовательного приближения

Данный АЦП отличается от структурной схемы последовательного счета тем, что вместо счетчика импульсов включен регистр последовательных приближений (РПП). В основе РПП лежит принцип дихотомии, т.е. последовательного сравнения преобразуемого напряжения  $U_{вх}$  с  $\frac{1}{2}$ ,  $\frac{1}{4}$ ,  $\frac{1}{8}$  и т.д. возможного максимального его значения  $U_m$ . Это позволяет для  $n$  - разрядного АЦП выполнить весь процесс преобразования за  $n$  последовательных шагов (итераций) вместо  $(2^{n-1})$  при использовании последовательного счета и получить существенный выигрыш по быстродействию.

На рис. 6 приведена структурная схема, реализующая метод последовательных приближений. На каждом шаге производится определение одного разряда, начиная со старшего. При первом сравнении определяется, больше или меньше напряжение  $U_{вх}$ , чем  $U_m/2$ .

На следующем шаге определяется, в какой четверти диапазона находится  $U_{вх}$ . Каждый последующий шаг сужает область возможного результата.

При каждом сравнении компаратор формирует импульсы, соответствующие состоянию «больше - меньше» (1 или 0), управляющие регистром последовательных приближений. График процесса последовательного приближения приведен на рис. 6.7.

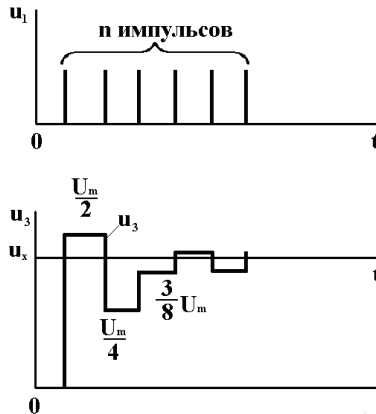


Рис. 6.7 – График процесса последовательного приближения

АЦП параллельного действия показан на рис. 6.8. Преобразователь (выполненный на компараторах) осуществляет одновременное квантование входного сигнала  $U_{вх}$  с помощью компараторов, включенных параллельно входному сигналу.

Пороговые уровни компараторов установлены с помощью резистивного деления в соответствии с используемой шкалой квантования. На выходах компараторов получаем квантованный сигнал, представленный в унитарном (десятичном) коде.

Для преобразования унитарного кода в двоичный или двоично-десятичный используется кодирующий преобразователь.

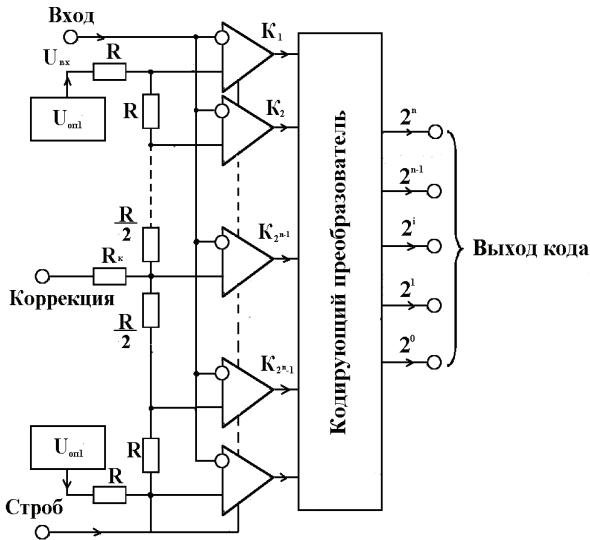


Рис. 6.8 – АЦП параллельного действия

При работе в двоичном коде все резисторы имеют одинаковые сопротивления  $R$ . Время преобразования такого преобразователя составляет один такт, т. е.  $T_{np} = T$ .

Преобразователь (выполненный на компараторах) осуществляет одновременное квантование входного сигнала  $U_{вх}$  с помощью компараторов, включенных параллельно входному сигналу.

Пороговые уровни компараторов установлены с помощью резистивного деления в соответствии с используемой шкалой квантования. На выходах компараторов получаем квантованный сигнал, представленный в унитарном (десятичном) коде.

Для преобразования унитарного кода в двоичный или двоично-десятичный используется кодирующий преобразователь.

При работе в двоичном коде все резисторы имеют одинаковые сопротивления  $R$ . Время преобразования такого преобразователя составляет один такт, т. е.  $T_{np} = T$ .

Максимальная частота дискретизации для данного типа преобразователя составляет  $\approx 100$  МГц. Для получения более широкой полосы пропускания компараторы можно выполнить сторнируемыми.

Делитель опорного напряжения представляет собой набор низкоомных прецизионных сопротивлений.

По выводу «Коррекция» возможна корректировка напряжения смещения нулевого уровня на входе, а по выводу  $U_{оп2}$  – абсолютной погрешности преобразования в конечной точке шкалы.

Номинальные значения опорных напряжений  $U_{оп1} = -0,075 \dots 0$  В,  $U_{оп2} = -2,1 \dots -1,9$  В. Типовая задержка срабатывания компараторов 7 нс.

Параллельно последовательный АЦП показан на рис. 6.9.

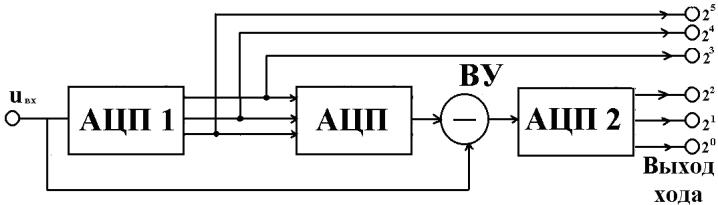


Рис. 6.9 – Параллельно последовательный АЦП

Данный АЦП работает в несколько тактов.

В первом такте АЦП1 преобразует старшие разряды входного напряжения  $U_{вх}$  в цифровой код (разряды  $2^3 \dots 2^5$ ).

Во втором такте, они преобразуются с помощью ЦАП в напряжение, которое вычитается из входного сигнала в вычитающем устройстве ВУ.

В третьем такте АЦП2 преобразует полученную разрядов код младших разрядов входного напряжения  $U_{вх}$ .

Такие преобразователи характеризуются меньшим быстродействием по сравнению с параллельными, но имеют меньше компараторов.

Количество каскадов в таких АЦП может быть увеличено, поэтому они часто называются конвейерными (многоступенчатыми).

АЦП с промежуточным преобразованием в интервале времени показан на рис. 6.10.

АЦП состоит из генератора линейно изменяющегося напряжения ГЛИН, двух компараторов К1 и К2, формирователя длительности импульса  $T_{и}$ , генератора тактовых импульсов и счетчика, с выхода которого снимается код преобразованного напряжения.

В основу работы этого преобразователя положен метод преобразования входного напряжения во временной интервал (рис. 6.11).



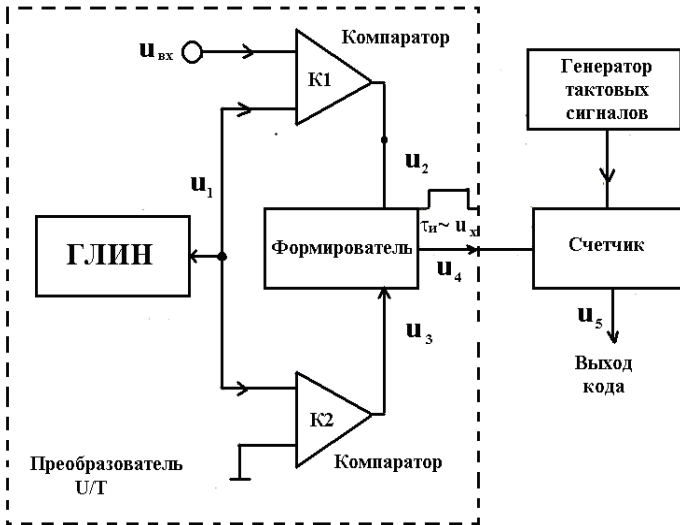


Рис. 6.10 – АЦП с промежуточным преобразованием в интервале времени

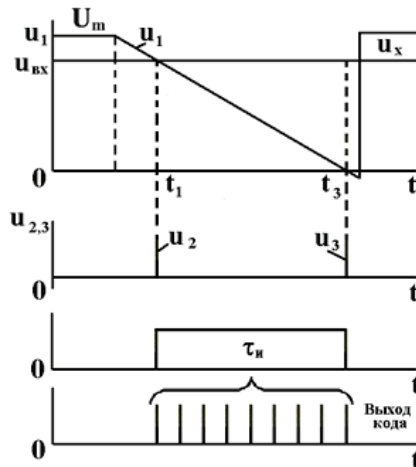


Рис. 6.11 – Метод преобразования входного напряжения во временной интервал

Первый импульс  $U_2$  (рис. 11) формируется при сравнении напряжения  $U_{вх}$  с напряжением  $U_1$ , второй импульс  $U_3$  формируется при достижении напряжени-

ем  $U_1$  нулевого уровня. Время преобразования таких АЦП в лучшем случае составляет 20 ... 50 мкс.

Уравнение АЦП определяется следующим образом. Напряжение  $U_1$  вырабатывается ГЛИН

$$U_1 = U_m - K \cdot t,$$

где  $K$  – крутизна пилообразного напряжения.

Моменты времени срабатывания компараторов  $K1$  и  $K2$ :

$$t_2 = \frac{U_m - U_{ex}}{K};$$

$$t_3 = \frac{U_m}{K}.$$

Длительность импульса определяется как разность

$$T_H = t_3 - t_2 = U_{ex}/K.$$

Количество импульсов, подсчитанных счетчиком, равно

$$N = f_0 \cdot t_u,$$

где  $f_0$  – частота тактового генератора.

АЦП средних значений (интегрирующие АЦП) разделяются на следующие виды: с времяимпульсным преобразованием; с частотно-импульсным преобразованием; со статистическим усреднением.

Наибольшее распространение получили АЦП с время импульсным преобразованием и АЦП с частотно-импульсным преобразованием.

АЦП с времяимпульсным преобразованием показан на рис. 6.12.

Работа АЦП с времяимпульсным преобразованием разделяется на три такта.

В первом такте производится заряд интегратора, во втором – его разряд, в третьем – коррекция нулевого уровня интегратора.

В первом такте, имеющем фиксированную, длительность  $T_0$  замкнут ключ  $S_1$  (остальные разомкнуты). В этом случае входное напряжение  $U_{вх}$  через замкнутый ключ  $S_{1и}$  резистор  $R_1$  заряжает емкость  $C_1$  интегратора; и входное напряжение растет линейно во времени. К концу интервала  $T_0$  напряжение на выходе интегратора будет равно

$$U_1(T_0) = K \int_0^{T_0} U_{ex} dt = KT_0 U_{ex}.$$

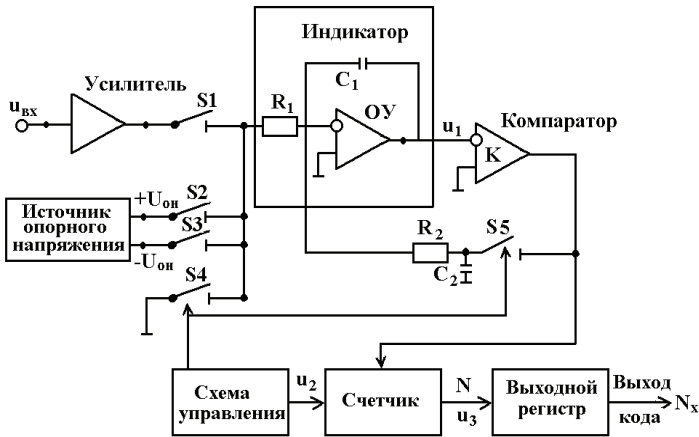


Рис. 6.12 – АЦП с времяимпульсным преобразованием

Во втором такте происходит разряд интегратора. В зависимости от требуемой полярности замыкается один из ключей  $S_2$  или  $S_3$ . Разряд интегратора происходит с постоянной скоростью, которая не зависит от накопленного в интеграторе заряда, поэтому с увеличением накопленного заряда время увеличивается. График процесса приведен на рис. 6.13.

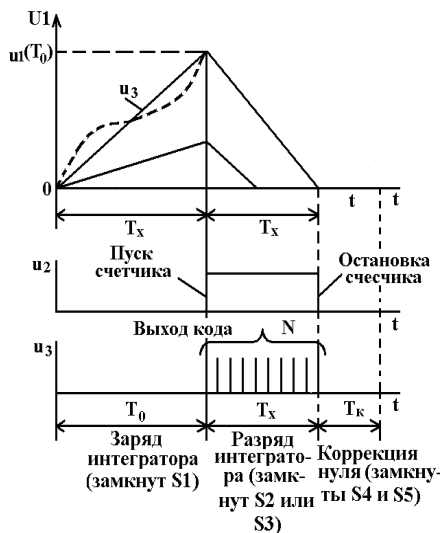


Рис. 6.13 – График работы АЦП с времяимпульсным преобразованием

Конец разряда интегратора фиксируется компаратором К, после чего ключ  $S_2$  (или  $S_3$ ) размыкается. Заполнение интервала  $T_x$  счетными импульсами, поступающими от схемы управления, позволяет найти числовой код  $N_x = T_x f_0$ .

В третьем такте производится коррекция нулевого уровня, для чего замыкаются ключи  $S_4$  и  $S_3$ , а остальные ключи размыкаются.

Так как вход интегратора через сопротивление  $R_1$  соединен с общей шиной, то конденсатор  $C_2$  через замкнутый ключ  $S_5$  заряжается до напряжения ошибки, которое после размыкания ключей  $S_4$  и  $S_5$  вычитается из входного сигнала.

АЦП с частотно-импульсным преобразованием показан на рис. 6.14.

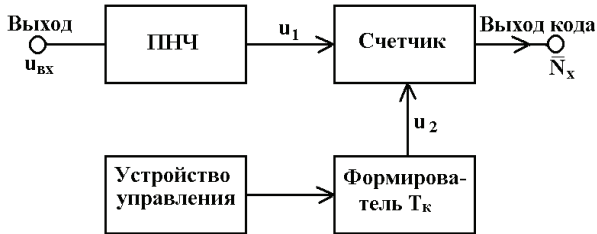


Рис. 6.14 – АЦП с частотно-импульсным преобразованием

Принцип работы АЦП основан на предварительном преобразовании входного напряжения в пропорциональную ему частоту следования импульсов, которая затем измеряется за фиксированный интервал времени.

Подсчитанное количество импульсов является цифровым эквивалентом входного напряжения (рис. 6.15).

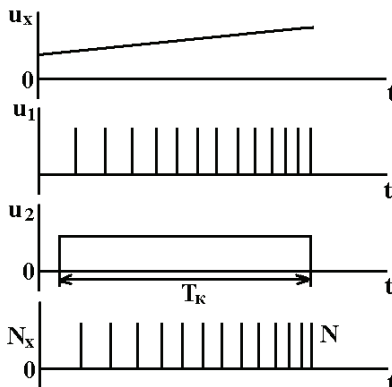


Рис. 6.15 – График работы АЦП с частотно-импульсным преобразованием

Основным звеном в этой схеме является частотно-импульсный преобразователь напряжения в частоту (ПНЧ).

При помощи ПНЧ входное напряжение преобразуется в частоту импульсов, при этом

$$f = KU_{ex}.$$

Число импульсов, подсчитанных счетчиком за выбранный интервал времени  $T_u$ , определяется формулой

$$N_x = \int_0^{T_u} f dt = \int_0^{T_u} U_{ex} dt = KT_u \bar{U}_{ex},$$

где  $U_{ex} = T_u^{-1} \int_0^{T_u} U_{ex} dt$  – среднее значение напряжения на интервале  $T_u$ .

Так как погрешность ПНЧ практически входит в погрешность АЦП, то для минимизации наиболее часто в качестве ПНЧ используется преобразователь с импульсной обратной связью, т. е. АЦП со статистическим усреднением.

АЦП со статистическим усреднением показан на рис. 6.16.

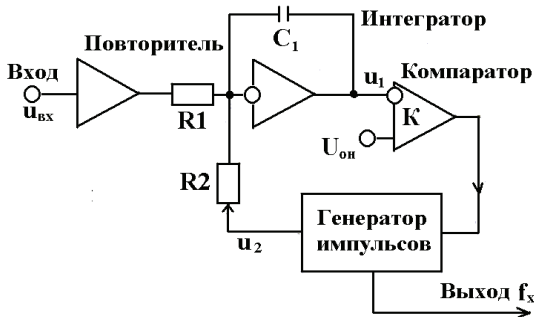


Рис. 6.16 – АЦП со статистическим усреднением

АЦП ПНЧ с импульсной обратной связью состоит из входного повторителя напряжения, интегратора и компаратора, управляющего генератора импульсов в цепи обратной связи интегратора.

Заряд конденсатора  $C_1$  интегратора осуществляется входным напряжением  $U_{вх}$ , а разряд производится импульсом с постоянной вольт-секундной площадью. Если входное напряжение имеет отрицательную полярность, то импульсы генератора должны быть положительными и наоборот. График работы преобразователя приведен на рис. 6.17.

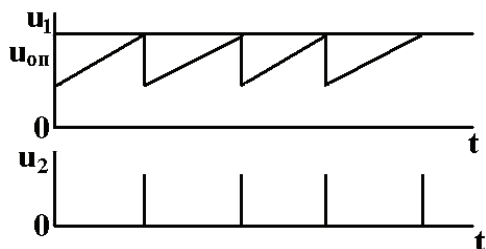
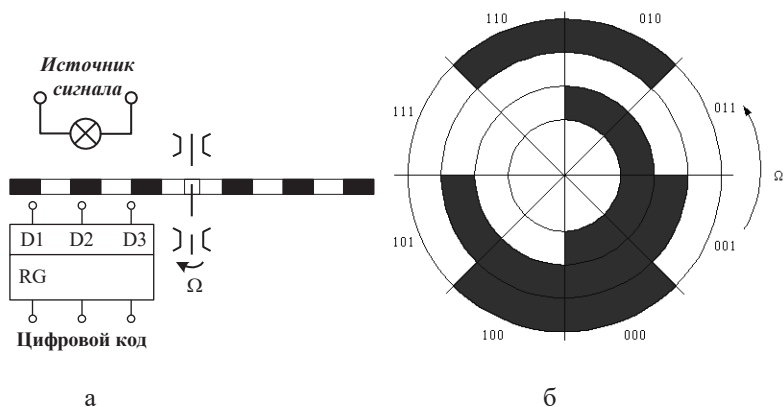


Рис. 6.17 – График работы преобразователя

АЦП прямого действия. Характерным представителем АЦП прямого действия является преобразователь «Вал – код» (рис. 6.18).



Десятич. число	0	1	2	3	4	5	6	7
Двоичный код	000	001	010	011	100	101	110	111
Код Грея	000	001	011	010	110	111	101	100

в

Рис. 6.18 – АЦП прямого действия преобразователь «Вал – код»  
а – неподвижная часть; б – подвижная часть; в – таблица кодирования

В его состав входят неподвижная часть, состоящая из источника оптического сигнала и системы считывания, и подвижная часть – кодировочное устройство, а также диск с прозрачными и непрозрачными участками. При движении вала происходит считывание угла его разворота (прозрачный участок диска – логическая единица, непрозрачный – логический ноль). Сигна-

лы кодировочного устройства передаются в регистр последовательным кодом.

Если маска кодирована двоичным кодом, то при считывании возникает ошибка, т.к. изменяется одновременно несколько двоичных разрядов. Поэтому кодирование осуществляется кодом Грея, который позволяет при каждом новом значении числа получить изменение 0 на 1 лишь в одном разряде, а это повышает точность считывания.

АЦП косвенного действия. Принцип работы АЦП косвенного действия с последовательным счетом по схеме с обратной связью, показан на рис. 6.19.

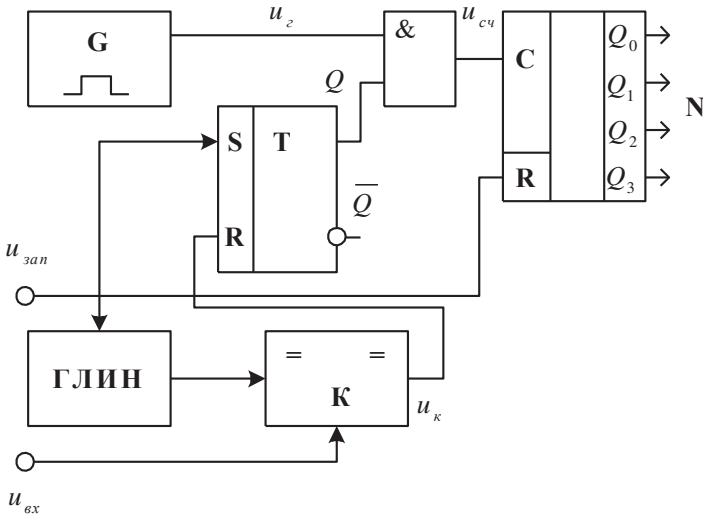


Рис. 6.19 – Схема АЦП косвенного действия с последовательным счетом

Эта схема с промежуточным преобразованием напряжения во временной интервал и последующим преобразованием его в цифровой код. Аналогичный способ реализован при измерениях напряжений постоянного и переменного токов в цифровых вольтметрах В7-16.

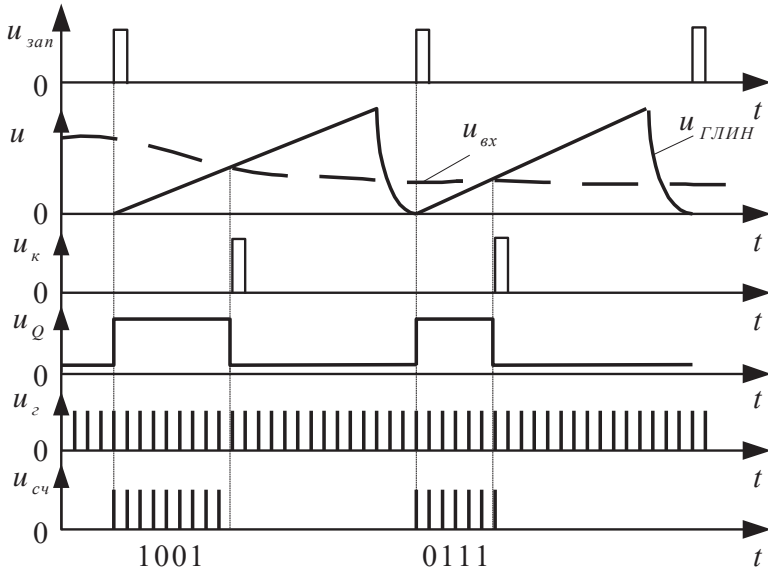


Рис. 6.20 – Временные диаграммы работы АЦП косвенного действия с последовательным счетом

Число разрядов счетчика определяется его максимальной величине счета  $N_{\max}=1/n$ , где  $n$  – относительная погрешность АЦП. Число разрядов счетчика определяется из неравенства  $2^n \geq N_{\max}$ . Процесс преобразования значения  $U_{\text{вх}}$  в число занимает время  $T \sim U_{\text{вх}}$ ,  $T_{\max}$  – время преобразования:  $T_{\max} = N_{\max} T_{\text{сч}}$ , где  $T_{\text{сч}}$  – период счетных импульсов.

## 6.2. Цифро-аналоговые преобразователи

Цифро-аналоговыми преобразователями (ЦАП) называют устройства, преобразующие цифровые величины в аналоговые (т. е. импульсный сигнал в непрерывный).

ЦАП вырабатывает напряжение пропорциональное коду на его выходах. Рассмотрим преобразователь параллельного двоичного кода в напряжение, получивший наибольшее распространение. Одна из таких схем с суммированием напряжений на ОУ приведена на рис. 6.21.



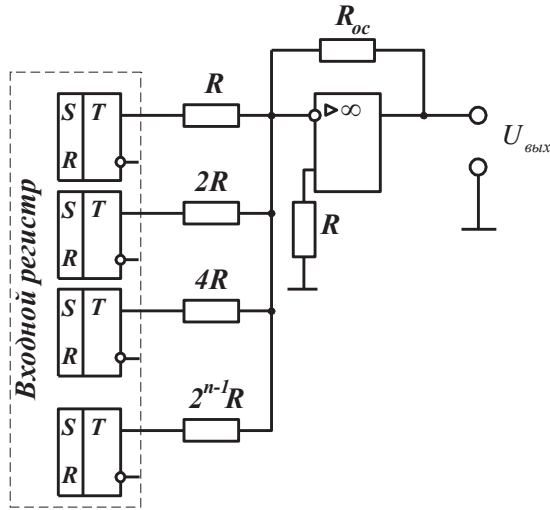


Рис. 6.21 – Схема ЦАП с суммированием напряжений на ОУ

Веса единиц в соседних разрядах регистра, записанного в двоичном коде, отличается вдвое. Поэтому, если единице в первом разряде соответствует напряжение на выходе ЦАП с уровнем  $\Delta U$ , то единице во втором разряде  $2\Delta U$ , в третьем -  $4\Delta U$  и т.д.

В схеме применен УПТ с параллельной отрицательной обратной связью, имеющей большое входное сопротивление и большой коэффициент усиления. Поэтому входной ток и напряжение на его входе практически равны 0.

Токи, обусловленные 1 в разрядах кода суммируются на резисторе  $R_{oc}$  и создают напряжение

$$U_{\text{вых}} = U_3 \frac{R_{oc}}{R} a_n + U_3 \frac{R_{oc}}{R} a_{n-1} 2^{-1} + \dots + U_3 \frac{R_{oc}}{R} a_1 2^{-(n-1)} = U_3 \frac{R_{oc}}{R} 2^{-(n-1)} (2^{n-1} a_n + 2^{n-2} a_{n-1} + \dots + a_1) = U_3 \frac{R_{oc}}{R} 2^{-(n-1)} N,$$

где  $\Delta U = U_3 \frac{R_{oc}}{R} 2^{-(n-1)}$ ,  $N$  – десятичное значение двоичного числа, введенного в регистр, т.е. напряжение на выходе ЦАП пропорционально значению числа в регистре.

Относительная погрешность  $\eta \sim 2^{-n}$ ,  $n=10$ ,  $\eta=0,1\%$ .

## Литература

1. Брамер Ю.А. Цифровые устройства. Учеб. пособие для вузов. – М.: Высш. шк., 2004. 229 с.
2. Калабеков, Б.А. Цифровые устройства и микропроцессорные системы. – Горячая линия – Телеком, 2003. 336 с.
3. Опадчий Ю.Ф. и др. Аналоговая и цифровая электроника. Учебник для вузов. М.: Горячая линия – Телеком, 2007. 768 с.
4. Гусев В.Г., Гусев Ю.М. Электроника и микропроцессорная техника: учеб. для вузов. – 4-е изд., доп. – Высшая школа, 2006. 797 с.
5. Партала О. Н. Цифровая электроника. – Санкт-Петербург.: Наука и техника, 2000. 208 с.
6. Потехин В. А. Цифровые устройства и микропроцессоры: Учеб. пособие - Томск: – Ч. 1 - Томск: ТМЦДО, 2002. 263 с.
7. Угрюмов Е. П. Цифровая схемотехника: Учебное пособие для вузов / Е. П. Угрюмов. – 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2004. 782 с.
8. Хоровиц П., Хилл У. Искусство схемотехники: Пер. с англ.– Изд.шестое. – М.: Мир, 2001. 704 с.
9. Аванесян Г.Р., Левшин В.П. Интегральные микросхемы ТТЛ, ТТЛШ: Справочник. – М.: Машиностроение, 1993. 256 с.
10. Преснухин Л.Н. и др. Расчет элементов цифровых устройств: Учеб. пособие. - М.: Высш. шк., 1991. 526 с.
11. Интегральные микросхемы и их зарубежные аналоги. Справочник. Т. 1. – М.: ИП РадноСофт, 2000. 512 с.
12. Стукалов С.Б. Радиотехнические цепи и сигналы. Ч.1. Теория радиотехнических сигналов. Детерминированные сигналы. М.: МГТУ ГА, 2017. 69 с.
13. Волков Л.Н, Немировский М.С., Шинаков Ю.С. Системы цифровой радиосвязи. Учеб. пособие. М.: Эко-Трендз, 2005. 392 с.
14. Феер К.: Беспроводная цифровая связь. М.: Радио и связь, 2000. 520 с.
15. Галкин В.А. Цифровая мобильная радиосвязь. М.: Горячая линия – Телеком, 2007. 592 с.
16. Прокис Дж.: Цифровая связь. М.: Радио и связь, 2000. 800с.
17. Шахнович И. Современные технологии беспроводной связи. М., Техносфера, 2004. 166 с.
18. Скляр Б. Цифровая связь. М.: Издательский дом Вильямс, 2003. 1104 с.
19. Приказ Министерства транспорта РФ от 20 октября 2014 г. N 297 "Об утверждении Федеральных авиационных правил "Радиотехническое обеспечение полетов воздушных судов и авиационная электросвязь в гражданской авиации».
20. Регламент радиосвязи (Международный союз электросвязи).

## Оглавление

<b>Список принятых сокращений.....</b>	<b>3</b>
<b>Введение.....</b>	<b>4</b>
<b>Глава 1. Системы счисления и арифметические операции.....</b>	<b>6</b>
1.1. Общие сведения о системах счисления .....	6
1.2. Системы счисления применяемые в дискретных устройствах..	7
1.3. Формы представления чисел в дискретных устройствах .....	8
1.4. Перевод чисел из одной системы счисления в другую.....	9
1.5. Арифметические операции в двоичной системе счисления...	11
<b>Глава 2. Теория логических функций .....</b>	<b>19</b>
2.1. Функции алгебры логики. Логические элементы .....	19
2.2. Формы представления логических функций.....	22
2.3. Законы алгебры логики.....	26
2.4. Минимизация логических функций.....	27
2.5. Анализ и синтез комбинационных схем.....	34
<b>Глава 3. Логические элементы .....</b>	<b>39</b>
3.1. Общие сведения об ИМС логических элементов .....	39
3.2. Интегральные логические элементы ДТЛ, ТТЛ и ТТЛШ логики базиса И–НЕ и их парам.....	49
3.3. Логические элементы эмиттерно-связанной логики .....	54
3.4. Логические элементы интегральной инжекционной логики...	55
3.5. Логические элементы на МОП-транзисторах.....	59
3.6. Условные обозначения ИМС.....	61
<b>Глава 4. Комбинационные цифровые устройства .....</b>	<b>64</b>
4.1. Общие сведения о цифровых устройствах. Классификация цифровых устройств .....	64
4.2. Преобразователи кодов. Шифраторы и дешифраторы.....	64
4.3. Комбинационные сумматоры.....	70
4.4. Компараторы.....	75
4.5. Мультиплексоры и демультиплексоры.....	78
<b>Глава 5. Последовательностные цифровые устройства.....</b>	<b>85</b>

5.1. . Цифровые автоматы .....	85
5.2. Триггеры .....	93
5.3. Регистры .....	102
5.4. Счетчики .....	109
5.5. Накапливающие сумматоры .....	121
5.6. Программные датчики.....	124
<b>Глава 6. Сопряжение аналоговых и цифровых устройств.....</b>	<b>129</b>
6.1. Аналого-цифровые преобразователи .....	129
6.2 Цифро-аналоговые преобразователи.....	144
<b>Литература .....</b>	<b>146</b>