

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ  
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ  
КАФЕДРА АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ

В.Г. Ланских

**ЭЛЕМЕНТАРНАЯ  
ЦИФРОВАЯ  
СХЕМОТЕХНИКА**

Учебное пособие

Дисциплина «Микросхемотехника»  
Специальность 220201, II курс, д/о, з/о  
Дисциплина «Схемотехника»  
Специальность 230201, II курс, д/о

Киров 2009

Печатается по решению редакционно-издательского совета Вятского государственного университета

УДК 621.382(07)  
Л 228

Рецензент: кандидат технических наук, доцент кафедры РЭС  
А.Г. Корепанов

Ланских В.Г. Элементарная цифровая схемотехника: Учебное пособие. – Киров: Изд-во ВятГУ, 2009. – 82 с.

Авторская редакция

Подписано в печать

Бумага офсетная

Заказ №

Текст напечатан с оригинал - макета, представленного автором

Усл. печ. л.

Печать копир. Aficio 1022

Бесплатно

---

610 000 , г. Киров, ул. Московская, 36

Оформление обложки, изготовление – ПРИП ВятГУ

© В.Г. Ланских, 2009

© Вятский государственный университет, 2009

## СОДЕРЖАНИЕ

ЛЕКЦИЯ 1. ВВЕДЕНИЕ.....	4
ЛЕКЦИЯ 2. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ЦИФРОВЫХ УСТРОЙСТВ .....	8
Контрольная работа №1 .....	15
Контрольная работа №2 .....	29
ЛЕКЦИЯ 3. СЕРИИ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ .....	30
ЛЕКЦИЯ 4. ТИПОВЫЕ КОМБИНАЦИОННЫЕ УСТРОЙСТВА.....	43
Контрольная работа №3 .....	52
Контрольная работа №4 .....	58
ЛЕКЦИЯ 5. ТИПОВЫЕ ПОСЛЕДОВАТЕЛЬНОСТНЫЕ УСТРОЙСТВА ....	61
Контрольная работа №5 .....	69
Контрольная работа №6 .....	81
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	82

## ЛЕКЦИЯ 1. ВВЕДЕНИЕ

Преобладающей элементной базой цифровой техники в настоящее время являются интегральные микросхемы.

Интегральная микросхема (ИС) - микроэлектронное изделие, выполняющее определенную функцию по преобразованию электрических сигналов и представляющее собой совокупность компонентов (транзисторов, резисторов, диодов и т.п.), изготавливаемых в одном технологическом цикле на общей диэлектрической или полупроводниковой основе.

Основными активными компонентами ИС являются биполярные или униполярные транзисторы. Параметром, определяющим уровень сложности ИС, является степень интеграции. Количественно она характеризуется числом активных компонентов, располагаемых на кристалле одной ИС. По этому критерию ИС условно подразделяются на:

- малые (ИС) - до 100 активных компонентов на кристалле;
- средние (СИС) - до 10000;
- большие (БИС) - до 100000;
- сверхбольшие (СБИС) - свыше 100000.

Современные микросхемы характеризуются широким диапазоном выполняемых функций: от простейших усилителей и инверторов до сложнейших вычислительных устройств. Так что микросхема как функциональный узел, входящий в структуру РЭА, в зависимости от функциональной сложности может принадлежать к разным уровням в иерархии структурных составных частей аппаратуры. Поэтому и существенно различен объем сведений о микросхемах, который необходим для их практического применения: для одних достаточно знать несколько параметров, для других необходимо знание десятков параметров, временных диаграмм, определяющих форму и последовательность управляющих сигналов, а для цифровых микросхем повышенной функциональной сложности - нередко и системы команд управления. Такое разнообразие характеристик микросхем объясняет широко распространенный подход к описанию их свойств, основанный на разделении микросхем на функциональные подгруппы.

Основные подгруппы аналоговых микросхем включают: генераторы сигналов, детекторы, усилители, фильтры, преобразователи сигналов, схемы сравнения, источники вторичного электропитания и др. Особенностью большинства аналоговых микросхем является их функциональная незавершенность, обуславливающая необходимость разработки электронного обрамления. В свою очередь, наличие дополнительных навесных деталей приводит к усложнению и увеличению размеров реальных функциональных узлов, снижению их надежности и в целом к уменьшению эффективности применения микросхем. Другая особенность аналоговых микросхем заключается в том, что многие из них требуют для обеспечения рабочих режимов нескольких источников питания, что также усложняет реальные устройства на их основе.

Для аналоговых микросхем характерны повышенные требования к стабильности напряжения питания, параметрам нагрузочных элементов, линиям передачи сигналов и др.

Изучение аналоговых ИС на всех уровнях представления и базовых принципиальных схем цифровых ИС является одним из объектов изучения в курсе "Электротехника и электроника".

Развитие микроэлектронной технологии и разработка принципов цифровой обработки аналоговых сигналов обусловили появление группы аналого-цифровых микросхем, номенклатурный ряд которых быстро увеличивается, пополняясь новыми функциональными устройствами. Для микросхем аналого-цифровых и цифро-аналоговых преобразователей характерны сложность внутренней структуры, наличие прецизионных элементов, например резистивных делителей, и другие особенности. Изучение принципов построения и использования АЦП и ЦАП отнесены к предметам других дисциплин, изучаемых на последующих курсах.

Цифровые ИС не критичны к разбросу параметров своих элементов, изменениям в широких пределах уровней сигналов на входах, не содержат в своих структурах емкостей, индуктивностей. Эти свойства цифровых ИС позволяют изготавливать их по полупроводниковой технологии, что и определило высокий уровень их развития.

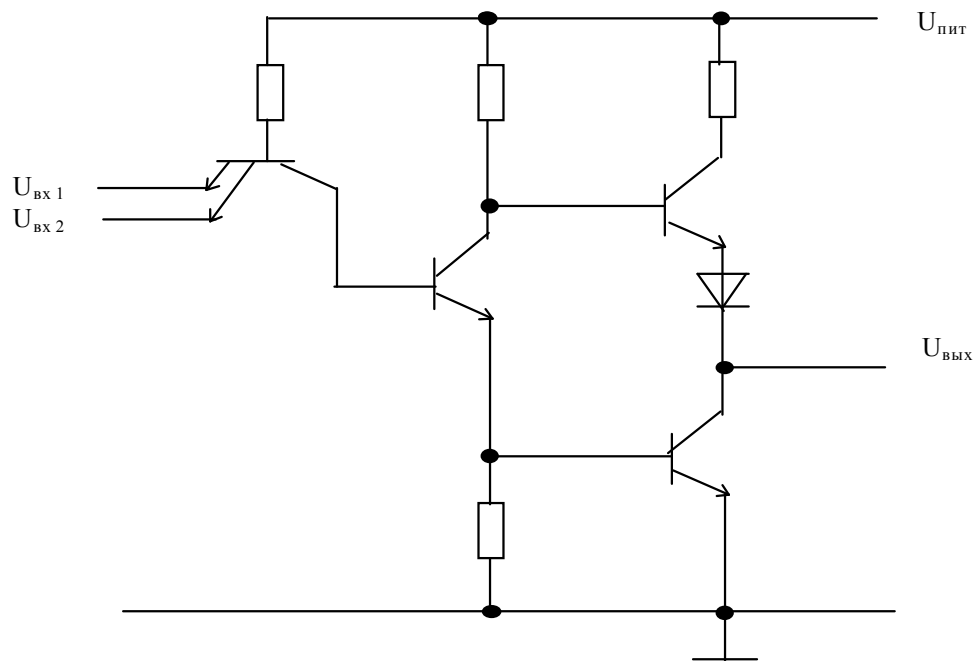


Рис. 1.1

Используются различные уровни схемотехнического представления ИС. Наиболее детальный из них - электрическая принципиальная схема ИС в виде соединения отдельных электронных компонентов (рис. 1.1).

Следующий более общий уровень - функциональная схема. На этом уровне приведенная выше схема представляется следующим образом (рис. 1.2).

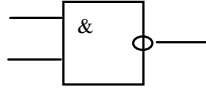


Рис. 1.2

Но если из этих функциональных схем отдельных логических элементов путем соответствующего их соединения построить более сложное устройство, например (рис. 1.3),

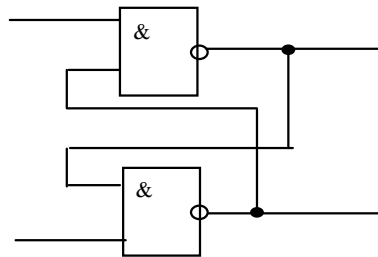


Рис. 1.3

то полученную схему следует называть принципиальной схемой в данном случае RS-триггера, функциональная схема которого, в свою очередь, выглядит следующим образом (рис. 1.4).

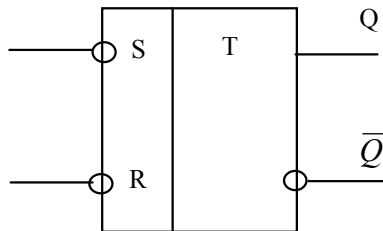


Рис. 1.4

С другой стороны, такая функциональная схема совсем не обязательно предполагает однозначно такую же принципиальную схему триггера. Триггер с такими же функциями (функциональная схема) может быть реализован с помощью множества других принципиальных схем.

Еще более высокий уровень обобщения используется для представления сложнофункциональных БИС и СБИС (микропроцессоров и микроЭВМ). В настоящее время уровень достижений в направлении создания СБИС характеризуется следующими данными: цифровые базовые матричные кристаллы содержат более 200 тыс. логических элементов на кристалле, динамическое запоминающее устройство имеет информационную емкость до 200 млн. бит на кристалле, статическое запоминающее устройство имеет информа-

ную емкость до 4 млн. бит на кристалле. Структура таких микросхем представляется и изучается в виде соединения отдельных функциональных узлов и блоков, рассмотренных на предыдущей ступени иерархии. Такое представление называется структурной схемой СБИС. Принципы построения и использования отдельных СБИС и их комплектов будут рассматриваться в курсах "ЭВМ и ВС", "Микропроцессоры", "Микропроцессорные устройства систем управления" и др.

С учетом сказанного предметом изучения курса "Микросхемотехника" является изучение на функциональном и структурном уровнях методов построения цифровых ИС малой и средней степени интеграции.

## ЛЕКЦИЯ 2. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ЦИФРОВЫХ УСТРОЙСТВ

Одним из главных понятий цифровой схемотехники является понятие цифрового автомата. Термин «автомат» в настоящем пособии используется в двух аспектах: 1) автомат как устройство, выполняющее все действия без участия человека; 2) автомат как математическая модель, описывающая реальные технические автоматы.

Информация во внешнем по отношению к автомату мире представляется в непрерывном или дискретном (цифровом) виде. Внутри цифрового автомата информация всегда представляется в виде чисел, записанных в той или иной системе счисления. Вопрос о выборе системы счисления для цифрового автомата – один из важнейших вопросов проектирования алгоритмов функционирования отдельных устройств автомата и расчета его технических характеристик.

Система изображения любых чисел с помощью ограниченного набора символов называется **системой счисления**. Набор символов, применяемый в той или иной системе счисления, называется ее **алфавитом**. Каждый из символов алфавита системы счисления называется **цифрой**.

Существуют различные системы счисления. От их особенностей зависит наглядность представления числа при помощи цифр и сложность выполнения арифметических операций над числами.

Если в системе счисления каждой цифре, располагаемой в любом месте числа, соответствует одно и то же значение – количественный эквивалент, то такая система называется **непозиционной**. Общим недостатком непозиционных систем счисления являются трудности записи в таких системах больших чисел и трудности выполнения в них арифметических операций, поскольку для этого используются громоздкие правила.

Преимущественное применение в цифровой технике нашли позиционные системы счисления. Система счисления называется **позиционной**, если одна и та же цифра имеет различное значение, которое определяется ее позицией в последовательности цифр, изображающей число. Количество цифр в алфавите называется **основанием** системы счисления.

Позиционной является десятичная система счисления. Ее алфавит – арабские цифры 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Любое число в десятичной системе счисления представляется в виде соответствующей последовательности указанных цифр. Например, 735012. А количественный эквивалент этого числа определяется с помощью следующего полинома:  $7 \times 10^5 + 3 \times 10^4 + 5 \times 10^3 + 0 \times 10^2 + 1 \times 10^1 + 2 \times 10^0$ . В этой записи 10 (т.е. основание системы) в соответствующей степени называется **весом**, т.е. количественным значением каждой единицы, определяемым местом соответствующего символа в изображении числа. При изображении разных чисел коэффициенты перед степенями могут менять свое цифровое выражение, но веса единиц, расположенных на одних и тех же позициях во всех числах, сохраняют одно и то же заранее обусловленное



значение. Именно это обстоятельство и позволяет изображать число в виде последовательности цифр, в которой веса не отображаются, а подразумеваются. Номер позиции, который определяет вес единицы, расположенной на этой позиции, называется **разрядом**. В десятичной системе счисления значение каждого разряда больше значения соседнего справа разряда в 10 раз.

Сказанное о десятичной системе можно обобщить на любую позиционную систему счисления с основанием  $p$ . Любое целое число  $x$  может быть представлено в виде полинома

$$x = a_n \times p^n + a_{n-1} \times p^{n-1} + a_{n-2} \times p^{n-2} + \dots + a_1 \times p^1 + a_0 \times p^0,$$

где  $a_n, a_{n-1}, a_{n-2}, \dots, a_1, a_0$  – символы (цифры) из алфавита системы, в качестве которых чаще всего используются числа натурального ряда  $0, 1, 2, \dots, p-1$ . Тогда число в данной системе счисления может быть изображено в виде последовательности цифр  $a_n a_{n-1} a_{n-2} \dots a_1 a_0$ . В  $p$ -ичной системе счисления значение каждого разряда больше значения соседнего справа разряда в  $p$  раз.

Выбор основания системы счисления зависит от физических элементов, на основе которых строится то или иное устройство. В цифровой технике широко используются простые и надежные элементы с **двумя** устойчивыми состояниями. Выполнение элемента с десятью четко различимыми устойчивыми состояниями, которым можно было бы сопоставить цифры десятичной системы счисления, является сложной технической задачей. Указанное обстоятельство является одной из главных причин применения в цифровой технике позиционных систем счисления с недесятичным основанием: в первую очередь двоичной системы счисления, а в качестве вспомогательных – восьмеричной и шестнадцатеричной систем счисления.

В **двоичной** системе счисления  $p=2$ , следовательно, ее алфавит состоит из двух символов: 0 и 1. Любое число  $x$  может быть представлено в двоичной системе счисления в виде полинома

$$x = a_n \times 2^n + a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \dots + a_1 \times 2^1 + a_0 \times 2^0 + a_{-1} \times 2^{-1} + a_{-2} \times 2^{-2} + a_{-3} \times 2^{-3} \dots,$$

где  $a_n, a_{n-1}, a_{n-2}, \dots, a_1, a_0, a_{-1}, a_{-2}, a_{-3}, \dots$  могут принимать значения только 0 или 1. Тогда число в двоичной системе счисления может быть изображено в виде последовательности нулей и единиц  $a_n a_{n-1} a_{n-2} \dots a_1 a_0, a_{-1} a_{-2} a_{-3} \dots$ . Например, число, представленное в двоичной системе счисления  $(11010,11)_2$  в полиномиальном представлении будет выглядеть следующим образом:

$(11010,11)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$ . Полиномиальное представление позволяет найти десятичный эквивалент данного двоичного числа:  $1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 16 + 8 + 2 + 0,5 + 0,25 = 26,75$ . Таким образом,  $(11010,11)_2 = (26,75)_{10}$ .

В качестве промежуточных или вспомогательных систем счисления, используемых только при вводе и выводе информации для более компактного представления двоичных чисел применяются восьмеричная и шестнадцатеричная системы счисления.

В **восьмеричной** системе счисления  $p=8$ , ее алфавит состоит из восьми символов: 0, 1, 2, 3, 4, 5, 6, 7. Любое число  $x$  может быть представлено в восьмеричной системе счисления в виде полинома

$$x = b_n \times 8^n + b_{n-1} \times 8^{n-1} + b_{n-2} \times 8^{n-2} + \dots + b_1 \times 8^1 + b_0 \times 8^0 + b_{-1} \times 8^{-1} + b_{-2} \times 8^{-2} + b_{-3} \times 8^{-3} \dots,$$

где  $b_n, b_{n-1}, b_{n-2}, \dots, b_1, b_0, b_{-1}, b_{-2}, b_{-3}, \dots$  могут принимать значения от 0 до 7. Тогда число в восьмеричной системе счисления может быть изображено в виде последовательности цифр  $b_n b_{n-1} b_{n-2} \dots b_1 b_0, b_{-1} b_{-2} b_{-3} \dots$ . Например, число, представленное в восьмеричной системе счисления  $(756,25)_8$  в полиномиальном представлении будет выглядеть следующим образом:

$(756,25)_8 = 7 \times 8^2 + 5 \times 8^1 + 6 \times 8^0 + 2 \times 8^{-1} + 5 \times 8^{-2}$ . Полиномиальное представление позволяет найти десятичный эквивалент данного восьмеричного числа:  $7 \times 8^2 + 5 \times 8^1 + 6 \times 8^0 + 2 \times 8^{-1} + 5 \times 8^{-2} = 448 + 40 + 6 + 0,25 + 0,078125 = 494,328125$ . Таким образом,  $(756,25)_8 = (494,328125)_{10}$ .

В **шестнадцатеричной** системе счисления  $p=16$ , ее алфавит должен содержать 16 символов. Первые десять символов представляются десятью арабскими цифрами: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Для остальных шести, чтобы не изображать одну шестнадцатеричную цифру двумя арабскими цифрами, используются первые шесть букв латинского алфавита  $A, B, C, D, E, F$ , которым в десятичной системе счисления соответствуют числа 10, 11, 12, 13, 14, 15. Любое число  $x$  может быть представлено в шестнадцатеричной системе счисления в виде полинома

$$x = d_n \times 16^n + d_{n-1} \times 16^{n-1} + \dots + d_1 \times 16^1 + d_0 \times 16^0 + d_{-1} \times 16^{-1} + d_{-2} \times 16^{-2} + d_{-3} \times 16^{-3} \dots,$$

где  $d_n, d_{n-1}, \dots, d_1, d_0, d_{-1}, d_{-2}, d_{-3}, \dots$  могут принимать значения 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,  $A, B, C, D, E, F$ . Тогда число в шестнадцатеричной системе счисления может быть изображено в виде последовательности цифр  $d_n d_{n-1} d_{n-2} \dots d_1 d_0, d_{-1} d_{-2} d_{-3} \dots$ . Например, число, представленное в шестнадцатеричной системе счисления  $(A7B,C8)_{16}$  в полиномиальном представлении будет выглядеть следующим образом:

$(A7B,C8)_{16} = 10 \times 16^2 + 7 \times 16^1 + 11 \times 16^0 + 12 \times 16^{-1} + 8 \times 16^{-2}$ . Полиномиальное представление позволяет найти десятичный эквивалент данного восьмеричного числа:  $10 \times 16^2 + 7 \times 16^1 + 11 \times 16^0 + 12 \times 16^{-1} + 8 \times 16^{-2} = 2560 + 112 + 11 + 0,75 + 0,03125 = 2683,78125$ . Таким образом,  $(A7B,C8)_{16} = (2683,78125)_{10}$ .

Из сказанного ранее ясно, что для перевода числа из двоичной, восьмеричной или шестнадцатеричной системы счисления в десятичную проще всего воспользоваться полиномиальным представлением и подсчитать полученную сумму.

Правила перевода чисел из двоичной системы в восьмеричную, шестнадцатеричную и обратно достаточно просты, поскольку основания восьмеричной и шестнадцатеричной систем выражаются целыми степенями двойки ( $8=2^3, 16=2^4$ ).

Для перевода чисел из восьмеричной системы в двоичную достаточно каждую цифру восьмеричного числа представить трехразрядным двоичным числом – триадой.

Восьмеричная цифра	0	1	2	3	4	5	6	7
Двоичная триада	000	001	010	011	100	101	110	111

Например, требуется найти двоичный эквивалент восьмеричного числа  $(762,35)_8$ .

Преобразуемое восьмеричное число	7	6	2,	3	5
Соответствующее двоичное число	111	110	010,	011	101

Таким образом,  $(762,35)_8 = (111110010,011101)_2$ .

Для перевода чисел из шестнадцатеричной системы в двоичную достаточно каждую цифру шестнадцатеричного числа представить четырехразрядным двоичным числом – тетрадой.

Шестнадцатеричная цифра	Двоичная тетрада
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Например, требуется найти двоичный эквивалент шестнадцатеричного числа  $(A7B,C7)_{16}$ .

Преобразуемое шестнадцатеричное число	A	7	B,	C	7
Соответствующее двоичное число	1010	0111	1011,	1100	0111

Таким образом,  $(A7B,C7)_{16} = (101001111011,11000111)_2$ .

При обратном переводе из двоичной системы в восьмеричную систему необходимо разряды двоичного числа, отсчитывая от запятой влево и вправо разбить на триады. Неполные крайние группы дополняются нулями, после чего каждая триада заменяется соответствующей восьмеричной цифрой. Например, требуется найти восьмеричный эквивалент двоичного числа  $(1101111001,1101)_2$ .

Преобразуемое двоичное число	001	101	111	001,	110	100
Соответствующее восьмеричное число	1	5	7	1,	6	4

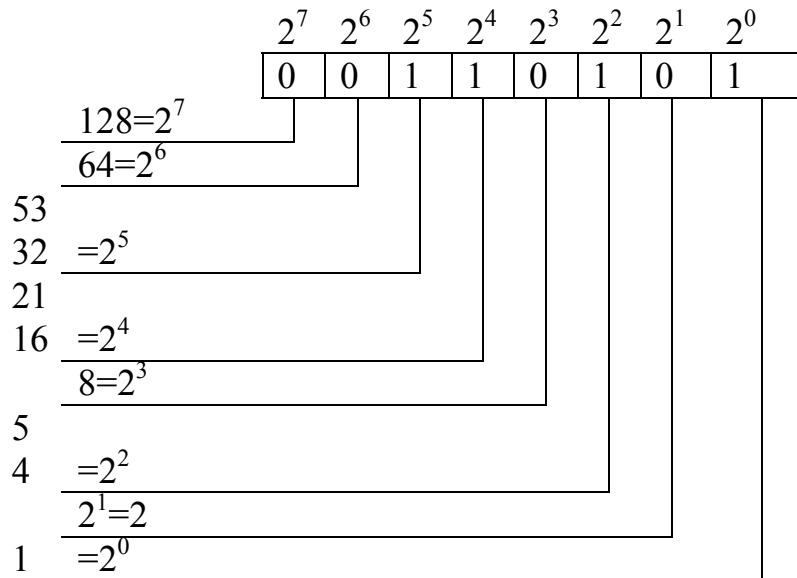
Таким образом,  $(1101111001,1101)_2 = (1571,64)_8$ .

При обратном переводе из двоичной системы в шестнадцатеричную систему необходимо разряды двоичного числа, отсчитывая от запятой влево и вправо разбить на тетрады. Неполные крайние группы дополняются нулями, после чего каждая тетрада заменяется соответствующей шестнадцатеричной цифрой. Например, требуется найти шестнадцатеричный эквивалент двоичного числа  $(1111111011,100111)_2$ .

Преобразуемое двоичное число	0111	1111	1011,	1001	1100
Соответствующее шестнадцатеричное число	7	F	B,	9	C

Таким образом,  $(1111111011,100111)_2 = (7FB,9C)_{16}$ .

Для преобразования целого десятичного числа в двоичное число можно предложить способ, являющийся разложением данного числа по степеням двойки. Для этого из преобразуемого десятичного числа вычитается наибольшее значение степени двойки, не превышающее преобразуемого числа. Затем эта операция повторяется для получаемых разностей. Процесс заканчивается, как только полученная разность станет равной нулю. Искомое двоичное выражение можно скомпоновать из единиц в позициях, соответствующих имеющимся в разложении степеням двойки, и нулей во всех остальных позициях. Следует обратить внимание на то, что при использовании этого способа, в отличие от предыдущего, при котором первым определяется значение младшего разряда искомого двоичного числа, первым формируется значение старшего разряда. Сказанное можно проиллюстрировать примером преобразования числа  $(53)_{10}$  в двоичное.



В общем случае, при некратных основаниях, перевод числа, содержащего целую и дробную части, из системы с основанием  $p_1$  в систему с основанием  $p_2$  можно выполнить по универсальному алгоритму. Согласно этому алгоритму перевод числа состоит из вычислительных процессов двух видов: 1) последовательного деления целой части числа, представленного в системе с основанием  $p_1$ , и образующихся при этом делении целых частных на основание системы  $p_2$ , записанное цифрами системы с основанием  $p_1$ ; 2) последовательного умножения дробной части числа, представленного в системе с основанием  $p_1$ , и дробных частей получающихся произведений на основание системы  $p_2$ , записанное цифрами системы с основанием  $p_1$ . При переводе целой части числа остатки, получающиеся в процессе последовательного деления, представляют собой цифры целой части числа в системе счисления с основанием  $p_2$ , записанные цифрами системы счисления с основанием  $p_1$ . Первый остаток является младшим разрядом, а последний остаток – старшим разрядом числа в системе счисления с основанием  $p_2$ . При переводе дробной части числа целые части произведений, получающихся при каждом умножении, не участвуют в последующих умножениях. Они представляют собой цифры дробной части числа в системе с основанием  $p_2$ , записанные цифрами системы счисления с основанием  $p_1$ . Значение целой части первого произведения является первой цифрой после запятой дробной части числа в системе с основанием  $p_2$ .

Описанный алгоритм используется для преобразования чисел из десятичной ( $p_1=10$ ) системы счисления в двоичную ( $p_2=2$ ) систему счисления. Например, процесс преобразования десятичного числа  $(30,6)_{10}$  в двоичное число распадается на два процесса: преобразование целой части  $(30)_{10}$  и преобразование дробной части  $(0,6)_{10}$ .

Процесс преобразования целой части можно отобразить следующим образом.

		30		2					
		30	15		2				
$2^0$	младший разряд	0	14	7	2				
$2^1$			1	6	3	2			
$2^2$				1	2	1	2		
$2^3$					1	0	0		
$2^4$	старший разряд						1		

Таким образом,  $(30)_{10} = (11110)_2$ .

Процесс преобразования дробной части можно отобразить следующим образом.

$$\begin{array}{r}
 0,6 \times 2 = 1,2 \\
 \boxed{1} \quad 0,2 \times 2 = 0,4 \\
 \quad \boxed{0} \quad 0,4 \times 2 = 0,8 \\
 \quad \quad \boxed{0} \quad 0,8 \times 2 = 1,6 \\
 \quad \quad \quad \boxed{1} \quad 0,6 \times 2 = 1,2
 \end{array}$$

и т.д.

Таким образом,  $(0,6)_{10} = (1001\dots)_2$ .

Если при переводе дробной части получается периодическая дробь (как в данном случае), то производится округление, исходя из заданной точности вычисления или из отведенного для представления дробной части количества разрядов.

Таким образом,  $(30,6)_{10} \approx (11110,1001)_2 = (30,5625)_{10}$ .

Пользуясь изложенным стандартным алгоритмом, можно переводить числа из десятичной системы в восьмеричную и шестнадцатеричную системы. При этом целая часть числа в десятичной системе последовательно делится на 8 или 16, а дробная умножается соответственно на 8 или 16. Ниже приведен пример преобразования десятичного числа  $(39519)_{10}$  в шестнадцатеричное число.

		39519		16					
		39504	2469		16				
младший разряд		15=F	2464	154	16				
			5	144	9	16			
				10=A	0	0			
старший разряд						9			

Таким образом,  $(39519)_{10} = (9A5F)_{16}$ .

Арифметическое суммирование над двоичными, восьмеричными и шестнадцатеричными числами осуществляется по тем же правилам, что и над десятичными числами, за исключением того, что переносы в следующие разряды производятся по 2, 8 и 16 соответственно, а не 10 как в десятичной системе.

### *Контрольная работа №1*

Заданы четыре целых числа:  $(\alpha)_{10}$  в десятичной системе,  $(\beta)_2$  в двоичной системе,  $(\gamma)_8$  в восьмеричной системе,  $(\delta)_{16}$  в шестнадцатеричной системе.

Найти:

- а)  $(\alpha)_2$ ;  $(\alpha)_8$ ;  $(\alpha)_{16}$ ;  
 $(\beta)_{10}$ ;  $(\beta)_8$ ;  $(\beta)_{16}$ ;  
 $(\gamma)_{10}$ ;  $(\gamma)_2$ ;  $(\gamma)_{16}$ ;  
 $(\delta)_{10}$ ;  $(\delta)_2$ ;  $(\delta)_8$ .
- б)  $(\Sigma)_{10} = (\alpha)_{10} + (\beta)_{10} + (\gamma)_{10} + (\delta)_{10}$ ;  
 $(\Sigma)_2 = (\alpha)_2 + (\beta)_2 + (\gamma)_2 + (\delta)_2$ ;  
 $(\Sigma)_8 = (\alpha)_8 + (\beta)_8 + (\gamma)_8 + (\delta)_8$ ;  
 $(\Sigma)_{16} = (\alpha)_{16} + (\beta)_{16} + (\gamma)_{16} + (\delta)_{16}$ .

в) выполнить преобразования

$$[(\Sigma)_2 \rightarrow (\Sigma)_{10}],$$

$$[(\Sigma)_8 \rightarrow (\Sigma)_{10}],$$

$$[(\Sigma)_{16} \rightarrow (\Sigma)_{10}].$$

г) сопоставить  $(\Sigma)_{10}$ ,  $[(\Sigma)_2 \rightarrow (\Sigma)_{10}]$ ,  $[(\Sigma)_8 \rightarrow (\Sigma)_{10}]$ ,  $[(\Sigma)_{16} \rightarrow (\Sigma)_{10}]$ . При отсутствии равенства искать ошибки.

Математический аппарат, описывающий действия дискретных устройств, к которым относятся и цифровые ИС, базируется на алгебре логики или булевой алгебре. Основное понятие алгебры логики – высказывание. Высказывание – некоторое предложение, о котором можно утверждать, что оно истинно или ложно. Любое высказывание можно обозначить символом  $x$  и считать, что  $x=1$ , если высказывание истинно, и  $x=0$ , если высказывание ложно. Логическая переменная – такая величина  $x$ , которая может принимать только два значения  $x=\{1,0\}$ . Логическая функция – функция  $f(x_1, x_2, \dots, x_n)$ , принимающая значение, равное 0 или 1, на наборе логических переменных  $x_1, x_2, \dots, x_n$ .

Для задания функции используются три способа: словесный, аналитический и табличный. При использовании табличного способа строится так называемая таблица истинности, в которой приводятся все возможные сочетания значений аргументов и соответствующие им значения логической функции. Если число аргументов функции равно  $n$ , то число различных сочетаний (наборов) значений аргументов равно  $2^n$ , а число различных функций  $n$  аргу-

ментов равно  $2^{2^n}$  Таблица истинности 16 функций двух аргументов представлена табл. 2.1.

Возможен и аналитический способ записи логических функций. Перечень обозначений логических операций, используемых при записи логических выражений, приведен в табл. 2.1. В дальнейшем функции одного и двух аргументов будем называть элементарными логическими функциями, имея в виду, что логические выражения этих функций, содержащие не более одной логической операции, элементарны. Сложные логические функции могут быть построены последовательным выполнением функциональных зависимостей, связывающих пары переменных. Следовательно, используя только элементарные операции  $f_0, \dots, f_{15}$  можно выполнить любую сложную логическую операцию. Такую систему функций принято называть полной системой или базисом.

Условие наличия 16 элементарных функций  $f_0, \dots, f_{15}$  является достаточным для синтеза логической функции любой сложности, но оно не является необходимым, то есть при синтезе можно ограничиться меньшим набором элементарных функций, взятых из  $f_0, \dots, f_{15}$ . Последовательно исключая из базиса функции, можно получить так называемый минимальный базис. Под минимальным базисом понимают такой набор функций, исключение из которого любой функции превращает полную систему функций в неполную.

Возможны различные базисы и минимальные базисы, отличающиеся друг от друга числом входящих в них функций и видом этих функций. Например, с помощью логических операций конъюнкции (И), дизъюнкции (ИЛИ) и инверсии (НЕ) можно выразить любую из элементарных операций  $f_0, \dots, f_{15}$ . Следовательно, эта совокупность из трех логических функций образует базис. Это означает, что любая логическая функция, как бы сложна она ни была, может быть представлена через логические операции И, ИЛИ, НЕ. Базис И, ИЛИ, НЕ не является минимальным. Из этой совокупности можно исключить функцию И либо функцию ИЛИ, и оставшийся набор функций будет удовлетворять свойствам минимального базиса. Таким образом, минимальный базис образует функция Шеффера (И-НЕ). Другой минимальный базис может быть представлен функцией Пирса (ИЛИ-НЕ).

Для сложного логического выражения установлен определенный порядок выполнения операций: вначале выполняются операции инверсии, затем операции конъюнкции и в последнюю очередь операции дизъюнкции. Если требуется нарушить это правило, используются скобки. В этом случае вначале выполняются операции в скобках. Дизъюнкция и конъюнкция обладают рядом свойств, аналогичных свойствам обычных арифметических операций сложения и умножения:



Таблица 2.1

Функ- ция	Аргументы $x_1$ и $x_2$								Название операции	Обозначение	
	0	0	0	1	1	0	1	1		основное	допол- нит.
$f_1$	0	0	0	0	0	0	0	0	Константа 0		
$f_2$	0	0	0	0	1	0	0	0	Конъюнкция, логическое И, логическое умножение	$x_1 \cdot x_2$	$x_1 \wedge x_2$
$f_3$	0	0	0	1	0	0	0	0	Запрет $x_2$ , отрицание импликации	$x_1 \cdot \bar{x}_2$	$x_1 \wedge \bar{x}_2$
$f_4$	0	0	0	1	1	0	0	0	$x_1$		
$f_5$	0	1	0	0	0	0	0	0	Запрет $x_1$ , отрицание импликации	$\bar{x}_1 \cdot x_2$	$\bar{x}_1 \wedge x_2$
$f_6$	0	1	0	0	1	0	0	0	$x_2$		
$f_7$	0	1	1	1	0	0	0	0	Сумма по модулю 2, неравнозначность, исключающее ИЛИ	$x_1 \oplus x_2$	$x_1 \vee x_2$
$f_8$	0	1	1	1	1	0	0	0	Дизъюнкция, логическое ИЛИ, логическая сумма	$x_1 + x_2$	$x_1 \vee x_2$
$f_9$	1	0	0	0	0	0	0	0	Логическое ИЛИ-НЕ, стрелка Пирса, функция Вебба	$x_1 \downarrow x_2$	
$f_{10}$	1	0	0	0	1	0	0	0	Равнозначность, эквивалентность	$x_1 \circ x_2$	
$f_{11}$	1	0	0	1	0	0	0	0	Отрицание $x_2$ , инверсия $x_2$	$\bar{x}_2$	
$f_{12}$	1	0	0	1	1	0	0	0	Импликация $x_1$	$x_1 \rightarrow x_2$	$x_1 \supset x_2$
$f_{13}$	1	1	0	0	0	0	0	0	Отрицание $x_1$ , инверсия $x_1$	$\bar{x}_1$	
$f_{14}$	1	1	0	0	1	0	0	0	Импликация $x_2$	$x_2 \rightarrow x_1$	$x_2 \supset x_1$
$f_{15}$	1	1	1	1	0	0	0	0	Логическое И-НЕ, штрих Шеффера	$x_1   x_2$	
$f_{16}$	1	1	1	1	1	0	0	0	Константа 1		

1) свойство ассоциативности или сочетательный закон:

$$x_1 \cdot (x_2 \cdot x_3) = (x_1 \cdot x_2) \cdot x_3,$$

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

2) свойство коммутативности или переместительный закон:

$$x_1 \cdot x_2 = x_2 \cdot x_1$$

$$x_1 + x_2 = x_2 + x_1;$$

3) свойство дистрибутивности или распределительный закон:

$$x_1 \cdot (x_2 + x_3) = x_1 \cdot x_2 + x_1 \cdot x_3,$$

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

Справедливость указанных свойств доказывается с помощью следующих аксиом: 1)  $\bar{\bar{x}} = x$ ; 2)  $x + x = x$ ; 3)  $x \cdot x = x$ ; 4)  $x + 1 = 1$ ; 5)  $x + 0 = x$ ; 6)  $x \cdot 1 = x$ ; 7)  $x \cdot 0 = 0$ ; 8)  $x + \bar{x} = 1$ ; 9)  $x \cdot \bar{x} = 0$ .

Несложно установить правильность соотношений, известных как законы де Моргана:

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

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

Можно сформулировать следующее правило применения формул де Моргана к сложным логическим выражениям. Инверсия любого сложного выражения, в котором аргументы связаны операциями конъюнкции и дизъюнкции, может быть представлена тем же выражением без инверсии с изменением всех знаков конъюнкции на знаки дизъюнкции, знаков дизъюнкции на знаки конъюнкции и инверсией всех аргументов. Например,

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

Исходными, из соображений удобства последующих преобразований, приняты следующие две канонические формы представления логических функций: совершенная дизъюнктивная нормальная форма (СДНФ) и совершенная конъюнктивная нормальная форма (СКНФ). Дизъюнктивной нормальной формой (ДНФ) называется такая форма представления логической функции, при которой логическое выражение функции строится в виде дизъюнкции ряда членов, каждый из которых является простой конъюнкцией аргументов или их инверсий. Например, функция  $f(x_1, x_2, x_3) = x_1 + x_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 + x_2 \cdot x_3$  является ДНФ, а функция  $f(x_1, x_2, x_3) = x_1 + x_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 + \overline{x_2 \cdot x_3}$  таковой не является, поскольку последний член дизъюнкции не является простой конъюнкцией аргументов. Если в каждом члене ДНФ представлены все аргументы (или их инверсии) функции, то такая форма и носит название СДНФ. Любая логическая функция имеет единственную СДНФ. Можно сформулировать следующее правило записи СДНФ функции по ее таблице истинности. Следует записать столько членов в виде конъюнкций всех аргументов, сколько единиц содержит функция в таблице. Каждая конъюнкция должна соответствовать определенному набору значений аргументов, обращающему функцию в 1, и если в этом наборе значение аргумента равно 0, то в конъюнкцию входит инверсия данного аргумента. Пусть, например, логическая функция  $f(x_1, x_2, x_3)$  задана следующей таблицей истинности (табл. 2.2)

Таблица 2.2

$x_3$	$x_2$	$x_1$	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Тогда для этой функции СДНФ будет иметь следующий вид:

$$f(x_1, x_2, x_3) = \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 + x_1 \cdot \bar{x}_2 \cdot x_3 + \bar{x}_1 \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot x_3. \quad (2.1)$$

Конъюнктивной нормальной формой (КНФ) называется форма представления функции в виде конъюнкции ряда членов, каждый из которых является простой дизъюнкцией аргументов или их инверсий. Примером КНФ может служить следующее выражение:

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

В СКНФ в каждом члене должны быть представлены все аргументы в прямом или инверсном виде. СКНФ строится по таблице истинности в соответствии со следующим правилом. Следует записать столько конъюнктивных членов, представляющих собой дизъюнкции всех аргументов, при скольких наборах значений аргументов функция равна 0, и если в наборе значение аргумента равно 1, то в дизъюнкцию входит инверсия этого аргумента. СКНФ функции, таблица истинности которой приведена в табл. 2.2, будет иметь следующий вид:

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

Любая логическая функция имеет единственную СКНФ.

Функциональная схема логического устройства может быть построена непосредственно по канонической форме (СДНФ или СКНФ) реализуемой функции. Так, например схема устройства, реализующего СДНФ (2.1), приведена на рис. 2.1.

Недостаток такого метода построения схем, обеспечивающего правильное функционирование устройства, состоит в том, что получающиеся схемы оказываются неоправданно сложными, требующими использования излишнего числа логических элементов. Во многих случаях удается так упростить логическое выражение, что соответствующая структурная схема оказывается существенно более простой. Методы такого упрощения называются методами минимизации логических функций.

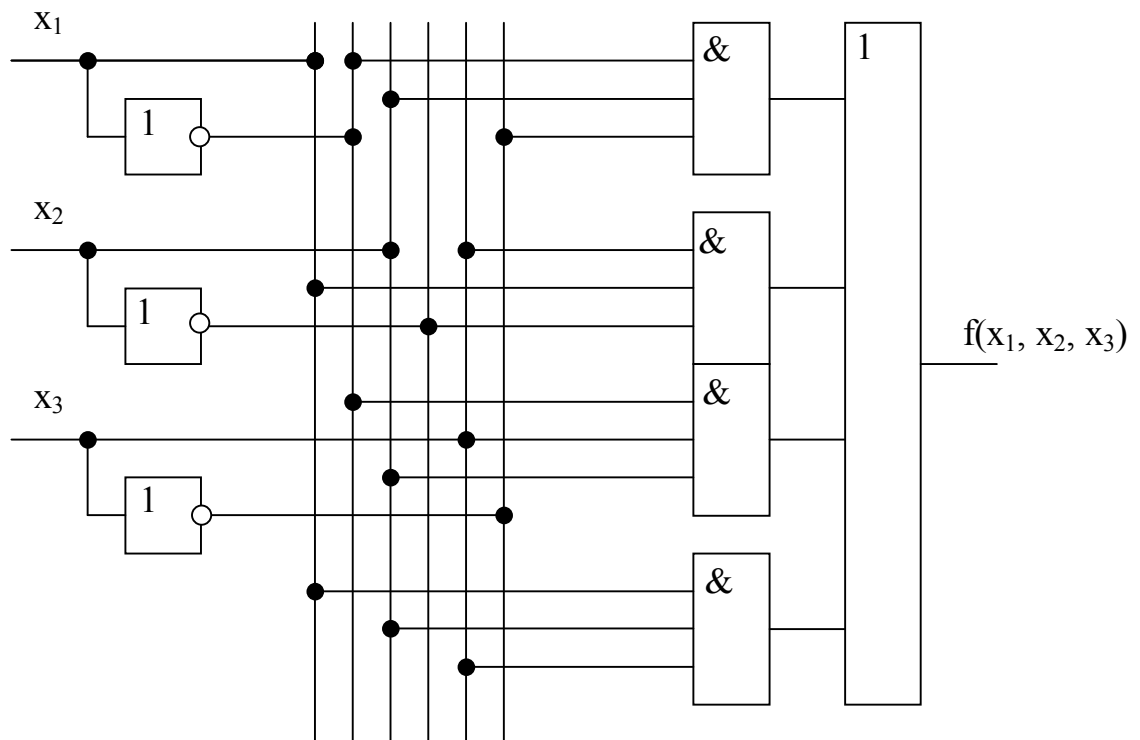


Рис. 2.1

Для дальнейшего изложения необходимо ввести ряд понятий. Под длиной ДНФ понимается число образующих ее элементарных конъюнкций. ДНФ, имеющая наименьшую длину по сравнению со всеми другими ДНФ данной функции, называется кратчайшей ДНФ (КДНФ). ДНФ, содержащая наименьшее число букв по сравнению со всеми другими ДНФ данной функции, называется минимальной ДНФ (МДНФ). Для упрощения булевых выражений могут быть использованы следующие тождества:

- 1)  $x_1 + \bar{x}_1 \cdot x_2 = x_1 + x_2$ ;
- 2)  $x_1 \cdot (\bar{x}_1 + x_2) = x_1 \cdot x_2$ ;
- 3)  $x_1 \cdot x_3 + \bar{x}_1 \cdot x_2 + x_2 \cdot x_3 = x_1 \cdot x_3 + \bar{x}_1 \cdot x_2$ ;
- 4)  $x_1 \cdot x_2 + x_1 \cdot \bar{x}_2 = x_1$ ;  $(x_1 + x_2) \cdot (x_1 + \bar{x}_2) = x_1$ , называемые правилом склеивания;
- 5)  $x_1 + x_1 \cdot x_2 = x_1$ ;  $x_1 \cdot (x_1 + x_2) = x_1$ , называемые правилом поглощения.

На их использовании основан один из аналитических методов минимизации булевых функций, называемый методом Квайна.

Для минимизации относительно несложных функций (с числом аргументов до пяти) может быть использован графический метод с помощью карт Карно-Вейча. Карта Карно-Вейча представляет собой определенную форму таблицы истинности функции. Число клеток карты равно числу возможных наборов значений аргументов  $2^n$ , где  $n$  - число аргументов функции. Каждая клетка карты соответствует определенному набору значений аргументов. Этот набор аргументов определяется присвоением значения логической 1 буквам, на пересечении строк и столбцов которых расположена клетка.

Формы карт Карно-Вейча для функций соответственно двух (а), трех (б) и четырех (в) аргументов приведены на рис.2.2.

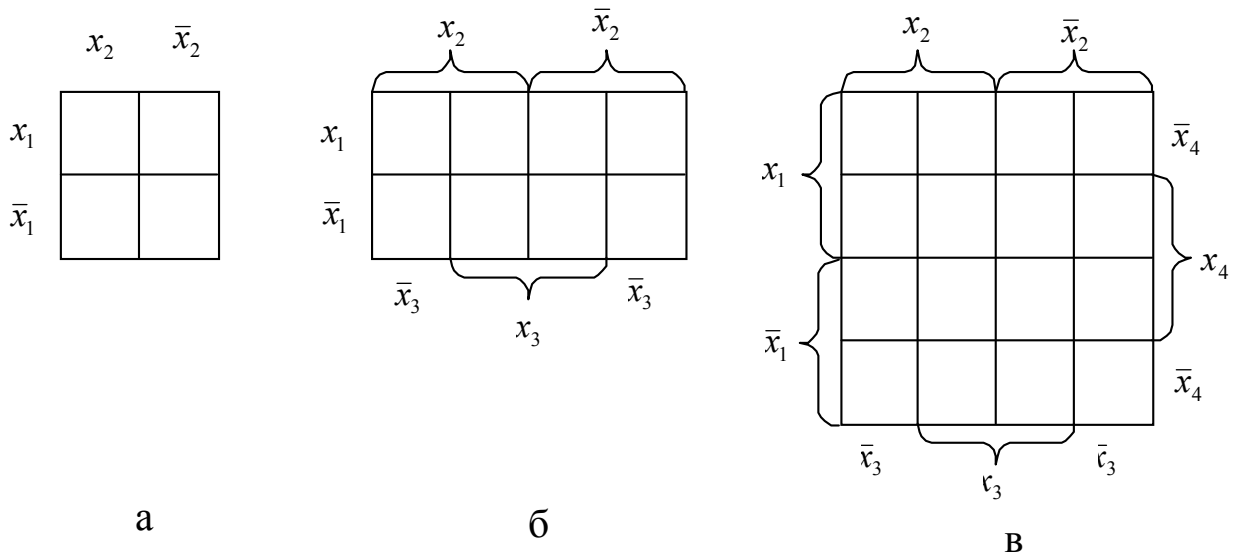


Рис. 2.2

При заполнении карты для каждой конкретной функции в каждую из клеток карты записывается значение функции на соответствующем этой клетке наборе значений аргументов. Так, например, если функция трех переменных задана таблицей истинности (табл. 2.3), то ее представление в виде карты Карно-Вейча будет выглядеть так, как представлено на рис. 2.3.

Главное достоинство карт Карно-Вейча состоит в том, что при всяком переходе из одной клетки в соседнюю вдоль столбца или строки изменяется значение лишь одного аргумента функции. Следовательно, если в паре соседних клеток содержится 1, то над соответствующими им членами канонической формы может быть произведена операция склеивания. Таким образом, использование карт Карно-Вейча облегчает поиск склеиваемых членов.

Таблица 2.3

$x_3$	$x_2$	$x_1$	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

	$x_2$		$\bar{x}_2$	
$x_1$	1	1	0	0
$\bar{x}_1$	0	1	1	0
	$\bar{x}_3$		$x_3$	

Рис. 2.3

Можно сформулировать следующие правила получения МДНФ функции с помощью карт Карно-Вейча. Все клетки, содержащие 1, объединяются в замкнутые области. При этом каждая область должна представлять собой прямоугольник с числом клеток  $2^k$ , где  $k=0, 1, 2, 3, \dots$ . Таким образом, допустимое число клеток в каждой области равно 1, 2, 4, 8,  $\dots$ . Области могут пересекаться, и одни и те же клетки могут входить в разные области. После выделения указанных областей производится запись МДНФ функции. Каждая из областей представляется в МДНФ членом, число букв в котором на  $k$  меньше общего числа аргументов  $n$  функции, т.е. равно  $n-k$ . Каждый член МДНФ составляется лишь из тех аргументов, которые для клеток соответствующей области имеют одинаковое значение (без инверсии либо с инверсией). Таким образом, при охвате клеток замкнутыми областями следует стремиться к тому, чтобы число областей было минимальным (при этом минимальным будет число членов в МДНФ функции), а каждая область содержала бы возможно большее число клеток (при этом минимальным будет число букв в членах МДНФ функции).

Пусть СДНФ функции  $f(x_1, x_2, x_3, x_4)$  имеет вид

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

Функциональная схема устройства, реализующего эту функцию при представлении ее в таком виде, приведена на рис. 2.4.

Минимизация этой функции с помощью карты Карно-Вейча приведена на рис. 2.5. Полученная в результате этих действий МДНФ функции имеет вид

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

Функциональная схема устройства, реализующего эту функцию при представлении ее в виде МДНФ, приведена на рис. 2.6. Сопоставление схем, представленных на рис. 2.4 и рис. 2.6, доказывает эффективность минимизации.

При построении замкнутых областей допускается сворачивание карты в цилиндр с объединением ее противоположных граней. В силу этого крайние

клетки строки или столбца карты рассматриваются как соседние и могут быть объединены в общую область.

Часто возникает необходимость в переходе от базиса И, ИЛИ, НЕ, в котором представлено логическое выражение, к базису И-НЕ. Для построения логического устройства на элементах И-НЕ сначала выполняется минимизация заданной функции с получением МДНФ, после чего осуществляется запись полученного логического выражения через операции И-НЕ. Методику синтеза устройства в базисе И-НЕ целесообразно рассмотреть на примере построения устройства, реализующего функцию, заданную таблицей истинности (табл. 2.4).

Таблица 2.4

$x_4$	$x_3$	$x_2$	$x_1$	$f(x_1, x_2, x_3, x_4)$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

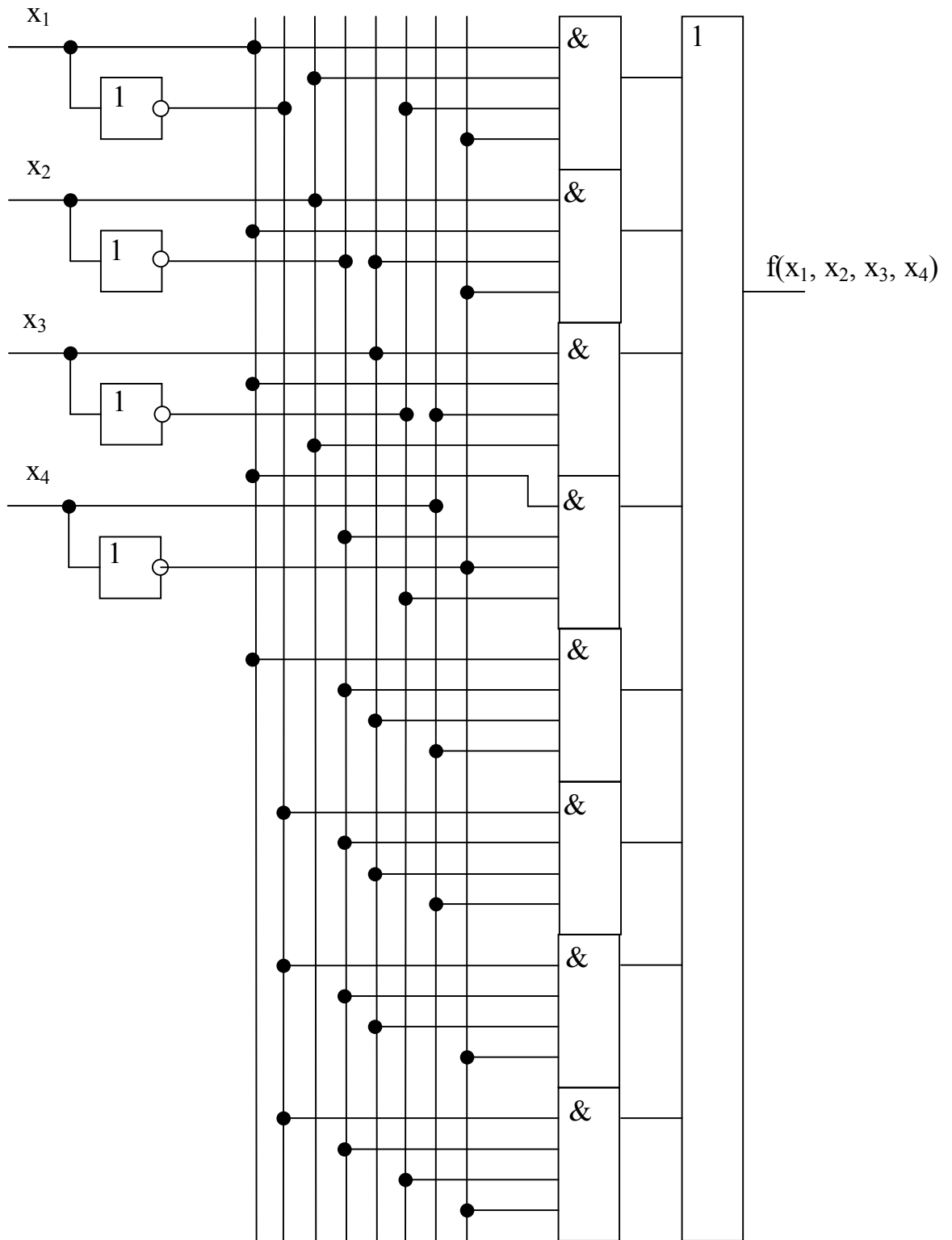


Рис. 2.4



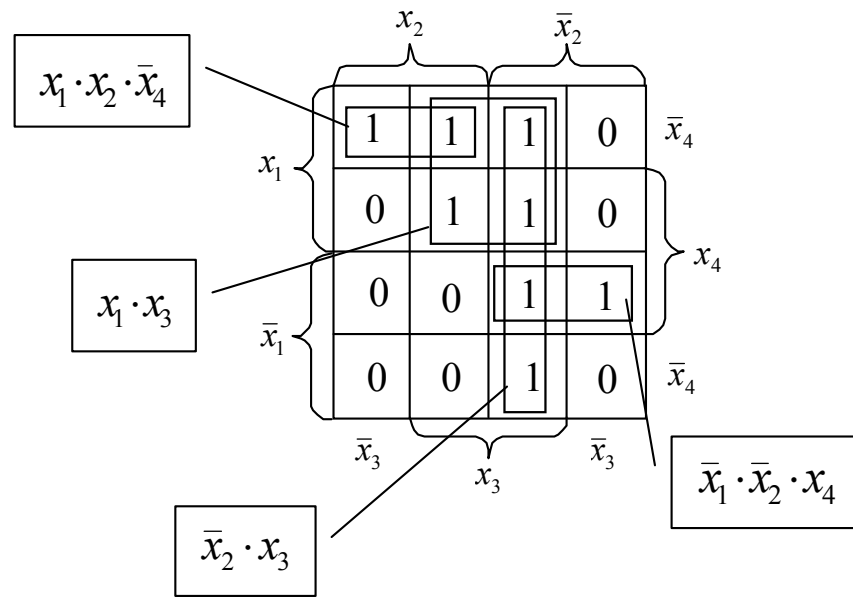


Рис. 2.5

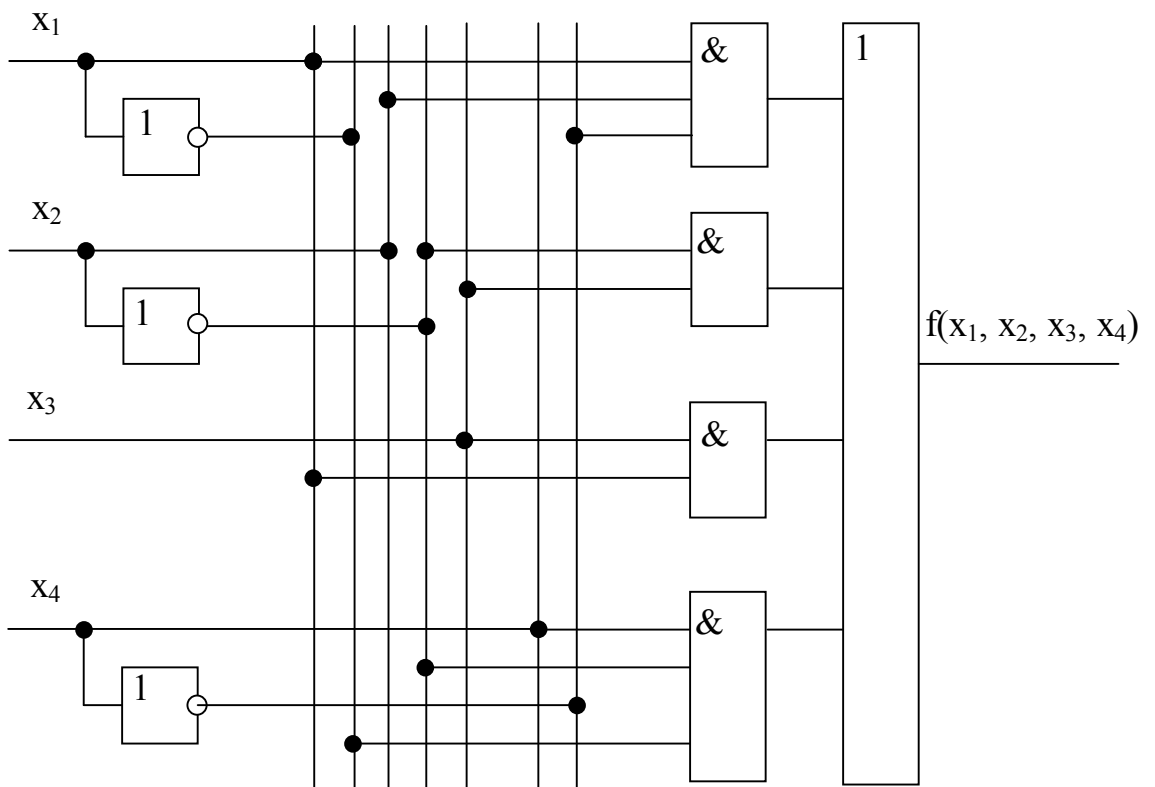


Рис. 2.6

Минимизация заданной функции выполняется с помощью карты Карно-Вейча (рис. 2.7).

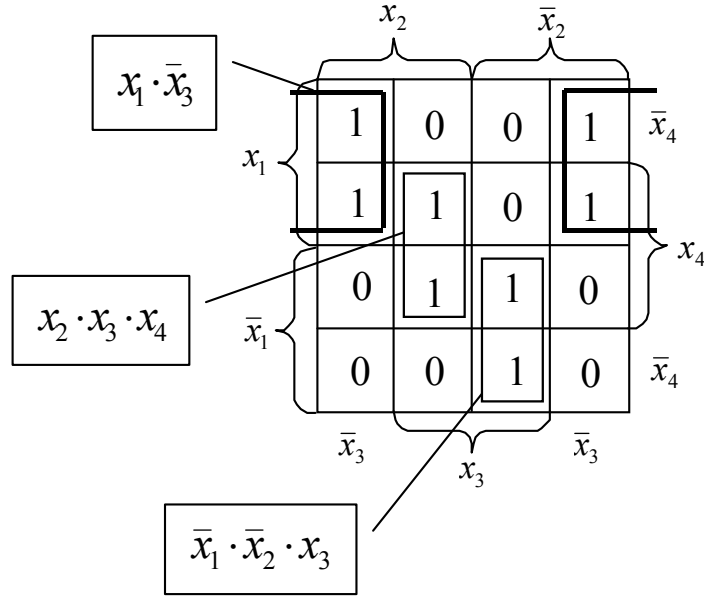


Рис. 2.7

В результате получается МДНФ вида

$$f(x_1, x_2, x_3, x_4) = x_1 \cdot \bar{x}_3 + x_2 \cdot x_3 \cdot x_4 + \bar{x}_1 \cdot \bar{x}_2 \cdot x_3. \quad (2.2)$$

Если дважды проинвертировать правую часть выражения

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

и произвести над ней преобразования в соответствии с правилом де Моргана, то получится выражение

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

в котором можно произвести замену обозначения операций И-НЕ на штрих Шеффера  $f(x_1, x_2, x_3, x_4) = (x_1 | \bar{x}_3) | (x_2 | x_3 | x_4) | (\bar{x}_1 | \bar{x}_2 | x_3)$ .

Функциональная схема устройства, соответствующего этому выражению, приведена на рис. 2.8.

Схема логического устройства может содержать элементы с разным числом входов. В схеме на рис. 2.8 используются, кроме инверторов, элементы И-НЕ с двумя и тремя входами. В выпускаемых промышленностью сериях элементов обычно предусматриваются элементы с разным числом входов. Поэтому для построения устройств в большинстве случаев могут быть использованы элементы с тем числом входов, какое требуется в отдельных элементах функциональной схемы. Иногда по различным соображениям приходится использовать элементы, число входов которых больше или меньше чем то, которое требуется при рассмотренных методах синтеза функциональных схем.

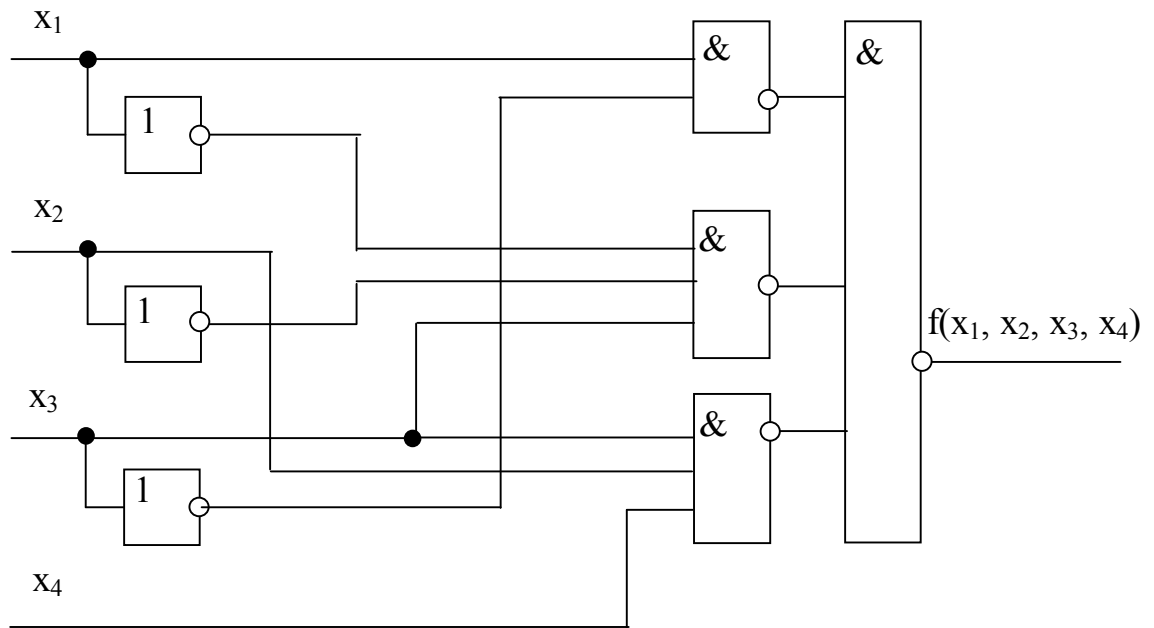


Рис. 2.8

Избыточный вход (входы) может быть оставлен свободным, т.е. не подключенным к каким-либо цепям, если это не нарушает логику работы элемента. Кроме того, неиспользуемый вход нежелательно оставлять свободным из соображений уменьшения влияния наводимых на него помех. Неиспользуемый вход может быть подключен к любому из используемых входов. Однако объединение входов приводит к тому, что к выходу предыдущего элемента, с которого сигнал подается на вход данного элемента, оказывается подключенным большее число входов. Такое возрастание нагрузки вызывает увеличение задержки распространения сигнала, снижение быстродействия элемента. Наиболее удачным следует считать способ, при котором на неиспользуемый вход подается потенциал, соответствующий логической константе 0 или логической константе 1. Причем на свободные входы элементов ИЛИ и ИЛИ-НЕ подается постоянный потенциал уровня, соответствующего логическому 0, а элементов И и И-НЕ – потенциал уровня, соответствующего логической 1. Элементы ИЛИ-НЕ или И-НЕ, в которых используется лишь один вход, а остальные соединены способом, описанным выше, выполняют операцию НЕ.

В логическом выражении могут оказаться члены с числом букв, превышающем число входов элементов. В этом случае для уменьшения числа используемых элементов следует провести соответствующие преобразования групп членов. Пусть, например, требуется построить устройство, реализующее функцию (2.2), используя только двухвходовые элементы И-НЕ. После группировки двух последних членов с вынесением за скобки  $x_3$ , двукратного

инвертирования всего выражения и применения к нему формулы де Моргана оно будет выглядеть следующим образом:

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

Применение формулы де Моргана к выражению  $x_2 \cdot x_4 + \bar{x}_1 \cdot \bar{x}_2$  обеспечивает его преобразование к виду  $x_2 \cdot x_4 + \bar{x}_1 \cdot \bar{x}_2 = \overline{\overline{(x_2 \cdot x_4)} \cdot \overline{(\bar{x}_1 \cdot \bar{x}_2)}}$ , после подстановки которого в предыдущее выражение получается

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

или с использованием обозначения операции штрих Шеффера

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

Построенная в соответствии с этим выражением функциональная схема приведена на рис. 2.9.

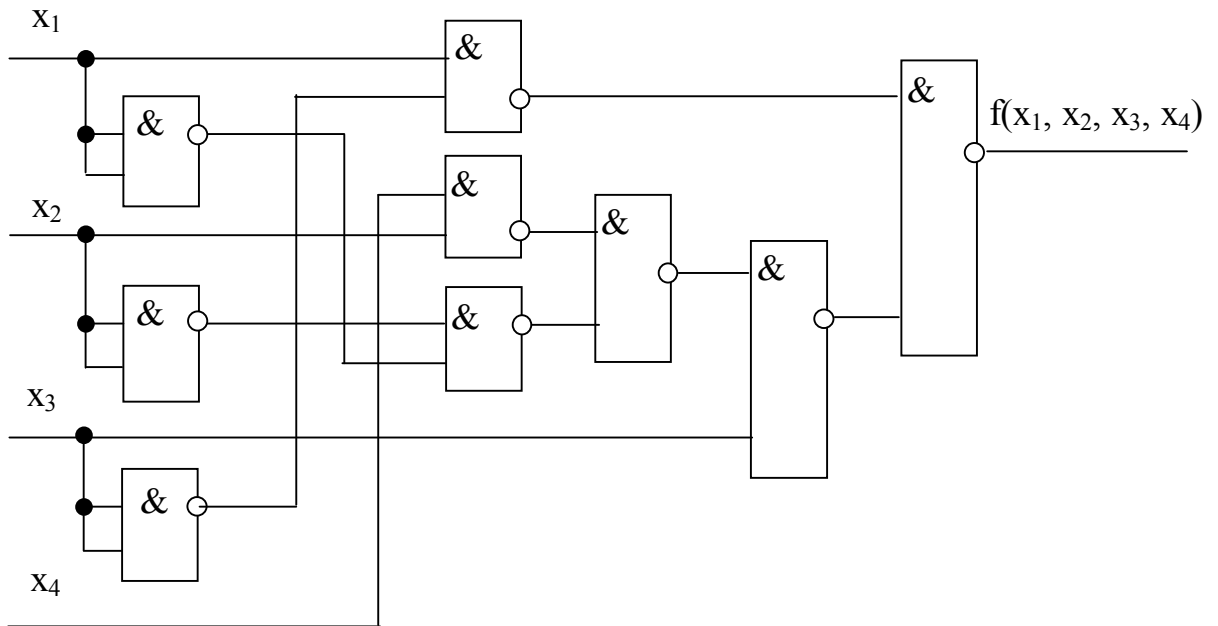


Рис. 2.9

Изложенным приемом нельзя воспользоваться в тех случаях, когда члены МДНФ не содержат общих букв. При этом необходимое преобразование логического выражения достигается с использованием тождественного соотношения  $a | b | c = \overline{a \cdot b \cdot c} = \overline{a \cdot (b \cdot c)} = \overline{a \cdot (b \cdot c)} = a | (b | c)$ .

Пусть, например, требуется синтезировать с использованием двухвходовых элементов И-НЕ логическую функцию, МДНФ которой представляется выражением  $f(x_1, x_2, x_3, x_4) = x_1 \cdot x_2 + x_3 \cdot x_4 + \bar{x}_1 \cdot \bar{x}_2$ . После применения к нему описанных ранее преобразований выражение примет вид

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

$$= (x_1 | x_2) | (x_3 | x_4) | (\bar{x}_1 | \bar{x}_2) = (x_1 | x_2) | ((x_3 | x_4) | (\bar{x}_1 | \bar{x}_2)).$$

Построенная в соответствии с данным выражением функциональная схема логического устройства приведена на рис. 2.10.

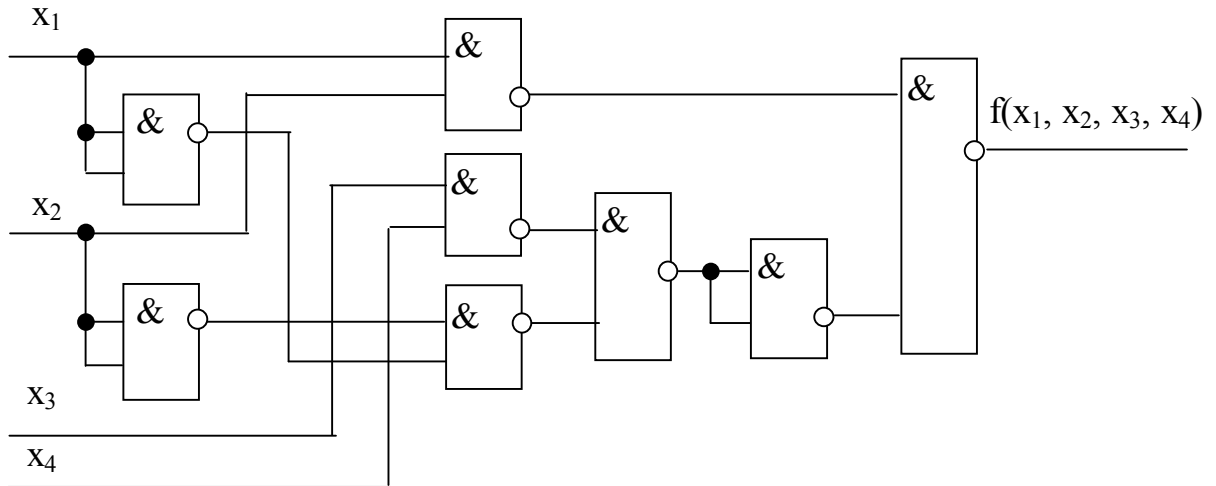


Рис. 2.10

### Контрольная работа №2

Логическая функция задана аналитически в виде, например, следующего выражения  $f(x) = \bar{x}_4 x_3 x_2 \bar{x}_1 + \bar{x}_4 x_3 \bar{x}_2 x_1 + \bar{x}_4 \bar{x}_3 x_2 \bar{x}_1 + \bar{x}_4 \bar{x}_3 \bar{x}_2 x_1$ . Необходимо:

- составить таблицу истинности заданной функции;
- синтезировать функциональную схему, реализующую заданную функцию;
- выполнить минимизацию заданной функции;
- синтезировать функциональную схему, реализующую минимизированную функцию.

### ЛЕКЦИЯ 3. СЕРИИ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ

Физический аналог булевой функции - схема, выполняющая соответствующее этой функции преобразование информации. Элементарные логические операции над двоичными переменными реализуются электронными схемами, называемыми логическими элементами. Физическими аналогами символов двоичного алфавита, т.е. 0 и 1, являются сигналы на входах и выходах логических элементов. Каждый такой элемент может находиться в одном из двух устойчивых состояний. Одно из этих состояний создает высокий уровень выходного напряжения, а другое - низкий. Первое обычно принимается за единицу, второе - за нуль, хотя возможно и иногда используется и обратное кодирование. Выходной сигнал логического элемента, например ИС серии 155, при изменении его под воздействием соответствующего входного сигнала выглядит следующим образом (рис. 3.1).

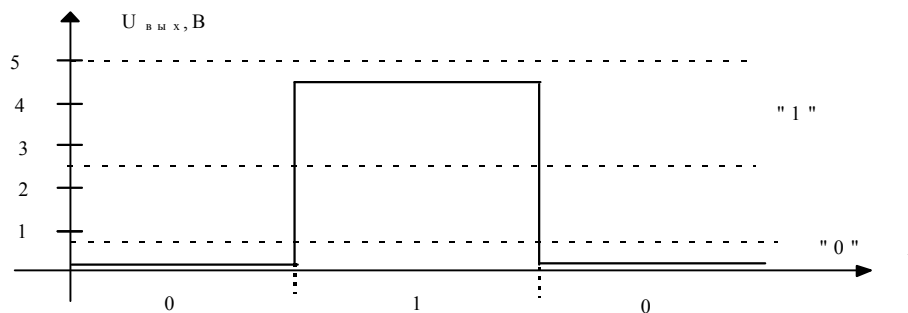


Рис. 3.1

В схемах цифровых устройств сигналы, соответствующие 0 и 1, могут изменяться не в любые, а во вполне определенные дискретные моменты времени. Интервал времени между двумя соседними такими моментами называется тактом. Для формирования тактов цифровые устройства обычно содержат генераторы тактовых импульсов и схемы формирования, вырабатывающие сигналы, отмечающие границы тактов и называемые сигналами синхронизации. Эти сигналы необходимы для синхронизации процессов передачи информации между различными цифровыми устройствами. В цифровых устройствах применяются потенциальный или импульсный способы представления двоичной информации. При использовании потенциального способа значениям двоичной переменной соответствуют разные уровни напряжений, причем потенциальный сигнал сохраняет уровень в течение такта (рис. 3.2).

При использовании импульсного способа единичное и нулевое значения двоичной переменной отображается наличием и отсутствием импульса на тактовом интервале (рис. 3.3а) или импульсами разной полярности (рис. 3.3б).

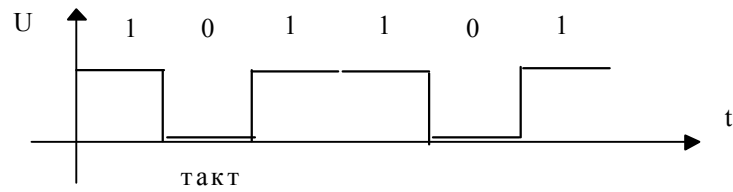
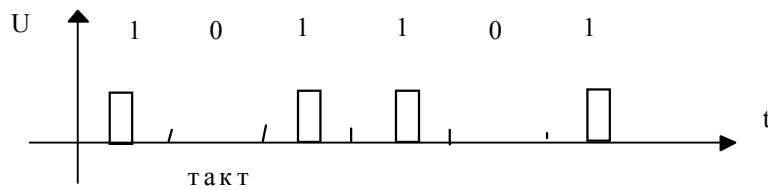
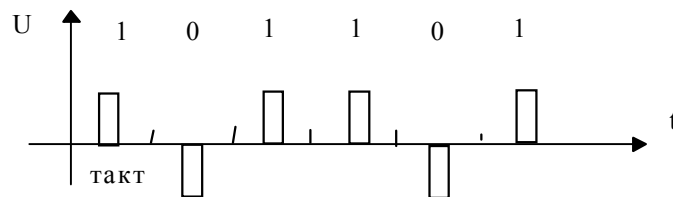


Рис. 3.2



а



б

Рис. 3.3

Двоичное слово, как совокупность 0 и 1, в цифровых устройствах может представляться в последовательной или параллельной формах или кодах. При последовательном коде для отображения каждого разряда двоичного слова отводится свой тактовый интервал. В этом случае при прохождении слова через какой-либо элемент каждый разряд слова последовательно во времени поступает на вход элемента, фиксируется им и поступает на выход элемента (рис. 3.4).

При параллельном коде для отображения всех разрядов двоичного слова отводится один тактовый интервал. Следовательно, для одновременной фиксации всех разрядов слова необходимо число элементов, равное числу разрядов слова. Входы синхронизации всех элементов следует объединить для обеспечения одновременности их срабатывания (рис. 3.5)

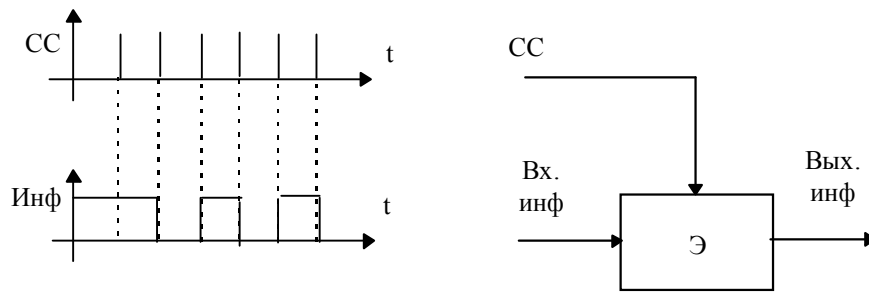


Рис. 3.4

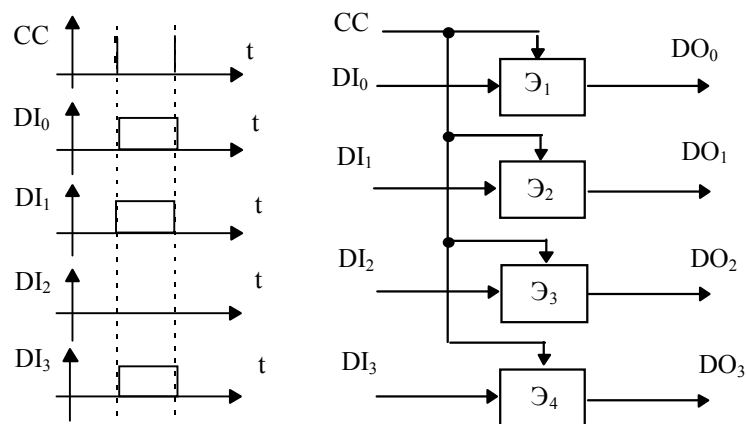


Рис. 3.5

Таким образом, при использовании последовательного кода двоичное слово разворачивается во времени, а при использовании параллельного кода - в пространстве, т.е. на аппаратуре. Использование параллельного кода обеспечивает большее быстродействие, но и требует больших аппаратных затрат. При использовании той или иной формы представления всегда приходится разрешать противоречие между обеспечением требуемого быстродействия и допустимыми аппаратными затратами.

Серией логических элементов называется предназначенный для построения цифровых устройств функционально полный набор логических элементов, объединяемый общими электрическими, конструктивными и технологическими параметрами.

Особенностью логических элементов одной серии является общая базовая схема - схема, выбранная в качестве основной при проектировании всех логических элементов данной серии. Базовая схема и определяет электрические, технологические, а иногда и конструктивные параметры серии.

Основными электрическими параметрами серии являются:

- уровни питающих напряжений  $U_{\text{пит}}$ ;
- минимальное выходное напряжение при логической 1 на выходе  $U_{\text{вых}}^1$ ;



- максимальное выходное напряжение при логическом 0 на выходе  $U_{\text{вых}}^0$ ;
- предельный входной ток при логическом 0 на входе  $I_{\text{вх}}^0$ ;
- предельный входной ток при логической 1 на входе  $I_{\text{вх}}^1$ ;
- коэффициент разветвления, численно равный количеству входов ИС этой же серии, которое может быть подключено к выходу одного элемента, характеризует нагрузочную способность элемента  $K_{\text{раз}}$ ;
- коэффициент объединения, численно равный количеству входов элемента, по которым реализуется функция элемента  $K_{\text{об}}$ ;
- мощность, потребляемая от источника питания  $P_{\text{пот}}$ ;
- параметры, характеризующие быстродействие, в качестве которых часто используются время задержки переключения из 1 в 0 и наоборот  $t_{\text{зд}}^{1,0}$  и  $t_{\text{зд}}^{0,1}$ .

Основные серии в зависимости от технологии используемых в них базовых схем можно подразделить на биполярные (ТТЛ, ЭСЛ, И<sup>2</sup>Л) и униполярные (n-МОП, p-МОП, КМОП) (табл. 3.1).

Таблица 3.1

Тип	Технология	$P_{\text{пот}}$ , мВт/ЛЭ	$t_{\text{зад}}$ , нс/ЛЭ
Биполярные	ТТЛ, ТТЛШ	1 - 20	2 - 10
	ЭСЛ	25 - 40	0,75 - 2
	И <sup>2</sup> Л	0,001 - 0,1	5
Униполярные	n - МОП	0,5	5
	p - МОП	0,5	100
	КМОП	$0,3 \times 10^{-3}$	10 - 45

Каждый из типов цифровых ИС имеет свои преимущества и недостатки, определяющие область их применения. Биполярные транзисторы пригодны для коммутации сравнительно больших токов, благодаря чему ИС на таких транзисторах характеризуются высоким быстродействием. Наиболее широкое применение из них находят ТТЛ и ТТЛШ ИС. Сверхскоростные устройства строятся на ЭСЛ ИС.

Транзисторы МОП-типа позволяют получить значительно более высокую плотность размещения их на кристалле ИС. Это определяет их широкое применение в БИС ЗУ и микропроцессоров.

Конструктивно ИС выполняются в пластмассовом, керамическом или металлокерамическом корпусе с различным количеством выводов, которые кроме электрического соединения обеспечивают и механическое крепление ИС на плате.

Для изображения элементов цифровой техники на принципиальных и функциональных схемах используется условно-графическое обозначение (УГО) в соответствии с ГОСТ 2.743 - 82.

УГО элемента цифровой техники имеет вид прямоугольника, к которому слева подводят линии входов, а справа - линии выходов.

УГО может содержать три поля - основное и два дополнительных. Дополнительные поля вводятся по необходимости.

В верхней части основного поля указывается функциональное обозначение элемента, в дополнительных полях - функциональные обозначения входов и выходов.

Размеры УГО по высоте определяются количеством входов и выходов, расстояние между линиями которых не должно быть менее 5 мм.

Размеры УГО по ширине определяются количеством полей, минимальная ширина одного дополнительного поля - 5 мм, основного - 10 мм.

Входы и выходы ИС могут быть прямыми и инверсными. Для инверсных используется специальное обозначение - кружок на линии или знак инверсии на функциональном обозначении входа или выхода в дополнительном поле.

Входы ИС могут быть статическими и динамическими. Для динамических входов используется специальное обозначение - наклонная черта.

Для обозначения ИС используются буквенно-цифровые сочетания, состоящие из следующих частей: трех- или четырехзначного числа, обозначающего номер серии; двухбуквенного индекса, обозначающего функциональное назначение элемента или устройства; числа, обозначающего порядковый номер разработки элемента в рамках данной серии; буквенного индекса, указывающего на разновидность микросхемы по какому-либо функциональному показателю (этот индекс может отсутствовать). В начале условного обозначения располагается одно- или двухбуквенный префикс: первая буква К обозначает микросхемы широкого применения, вторая - материал и тип корпуса : А - пластмассовый типа 4, Б - бескорпусное исполнение, Е - металлополимерный типа 2, И - стеклокерамический типа 4, М - металлокерамический типа 2, Н - керамический микрокорпус, Р - пластмассовый типа 2, С - стеклокерамический типа 2, Ф - пластмассовый микрокорпус. Вторая буква в префиксе может отсутствовать.

Примеры условных обозначений микросхем:

КР1820ИД1 - полупроводниковая микросхема широкого применения из серии 1820 микропроцессорных БИС, относящаяся к виду дешифраторов, имеющая номер разработки среди микросхем этого вида в указанной серии 1, конструктивно оформлена в пластмассовом корпусе типа 2;

К1121СА1 - полупроводниковая аналоговая микросхема широкого применения серии 1121, относящаяся к виду компараторов напряжения, имеющая номер разработки среди компараторов указанной серии 1.

Соответствие между двухбуквенным индексом в обозначении и функциональным назначением микросхемы приведено в табл. 3.2.

Таблица 3.2

Индекс	Назначение	Индекс	Назначение
	<b>Генераторы сигналов</b>		<b>Схемы сравнения</b>
ГС	гармонических	СК	амплитудные
ГГ	прямоугольных	СВ	временные
ГЛ	линейно изменяющихся	СС	частотные
ГФ	специальной формы	СА	компараторы напряжения
ГМ	шума	СП	прочие
ГП	прочие		<b>Многофункциональные схемы</b>
	<b>Детекторы</b>	ХА	аналоговые
ДА	амплитудные	ХЛ	цифровые
ДИ	импульсные	ХК	комбинированные
ДС	частотные	ХМ	цифровые матрицы
ДФ	фазовые	ХН	аналоговые матрицы
ДП	прочие	ХТ	комбинированные матрицы
	<b>Коммутаторы и ключи</b>	ХП	прочие
КТ	тока		<b>Логические элементы</b>
КН	напряжения	ЛИ	И
КП	прочие	ЛН	НЕ
	<b>Наборы элементов</b>	ЛЛ	ИЛИ
НД	диодов	ЛА	И - НЕ
НТ	транзисторов	ЛЕ	ИЛИ - НЕ
НР	резисторов	ЛС	И - ИЛИ
НЕ	конденсаторов	ЛБ	И - НЕ/ИЛИ - НЕ
НК	комбинированные	ЛР	И - ИЛИ - НЕ
НФ	функциональные	ЛК	И - ИЛИ - НЕ/И - ИЛИ
НП	прочие	ЛМ	ИЛИ - НЕ/ИЛИ
	<b>Усилители</b>	ЛД	расширители
УВ	высокой частоты	ЛП	прочие
УР	промежуточной частоты		<b>Триггеры</b>
УН	низкой частоты	ТВ	универсальные (JK)
УК	широкополосные	ТР	с отдельным запуском (RS)
УИ	импульсных сигналов	ТМ	с задержкой (D)
УЕ	повторители	ТТ	счетные (T)
УЛ	считывания и воспроизв.	ТД	динамические
УМ	индикации	ТЛ	Шмитта
УТ	постоянного тока	ТК	комбинированные
УД	операционные	ТП	прочие
УП	прочие		<b>Цифровые устройства</b>

Продолжение табл. 3.2

	<b>Фильтры</b>	ИР	регистры
ФВ	верхних частот	ИМ	сумматоры
ФН	нижних частот	ИЛ	полусумматоры
ФЕ	полосовые	ИЕ	счетчики
ФР	режекторные	ИВ	шифраторы
ФП	прочие	ИД	дешифраторы
	<b>Формирователи</b>	ИК	комбинированные
АГ	импульсов прямоуг. формы	ИА	АЛУ
АФ	импульсов спец. формы	ИП	прочие
АА	адресных токов		<b>Запоминающие устройства</b>
АР	разрядных токов	РМ	матрицы ОЗУ
АП	прочие	РВ	матрицы ПЗУ
	<b>Преобразователи сигналов</b>	РУ	ОЗУ
ПС	частоты	РТ	программируемые ПЗУ
ПД	длительности	РЕ	масочные ПЗУ
ПН	напряжения (тока)	РР	репрограммируемые ПЗУ с электрическим стиранием
ПМ	мощности	РФ	репрограммируемые ПЗУ с ультрафиолетовым стиранием
ПУ	уровня	РА	ассоциативные ОЗУ
ПВ	аналого-цифровые	РЦ	ЗУ на ЦМД
ПА	цифро-аналоговые	РП	прочие
ПР	код - код		<b>Вычислительные средства</b>
ПЛ	синтезаторы частоты	ВЕ	микроЭВМ
ПИ	делители частоты (аналог.)	ВМ	микропроцессоры
ПЦ	делители частоты (цифр.)	ВС	микропроцессорные секции
ПЕ	умножители частоты (анал.)	ВУ	микропрограммного управлен.
ПП	прочие	ВР	функциональные расширители
	<b>Модуляторы</b>	ВБ	синхронизации
МА	амплитуда	ВН	управления прерыванием
МС	частотные	ВВ	управления вводом-выводом
МФ	фазовые	ВТ	управления памятью
МИ	импульсные	ВФ	функциональные преобразователи информации
МП	прочие	ВА	сопряжения с магистралью
	<b>Схемы задержки</b>	ВИ	вреязадающие
БМ	пассивные	ВХ	микрокалькуляторы
БР	активные	ВГ	контроллеры
БП	прочие	ВК	комбинированные

Окончание табл. 3.2

	<b>Источники вторичного электропитания</b>	ВЖ	специализированные
ЕВ	выпрямители	ВП	прочие
ЕМ	преобразователи		
ЕН	стабилизаторы напряжения непрерывные		<b>Фоточувствительные схемы с зарядовой связью</b>
ЕК	стабилизаторы напряжения импульсные	ЦМ	матричные
ЕТ	стабилизаторы тока	ЦЛ	линейные
ЕУ	схемы управления импульсными стабилизаторами напряжения	ЦП	прочие
ЕС	системы источников вторичного электропитания		
ЕП	прочие		

Логические элементы, входящие в состав одной ИС, в функциональном отношении являются автономными и могут применяться независимо друг от друга. Однако следует иметь в виду, что все элементы одной ИС связаны по цепям питания, так что если даже какой-то элемент не используется, он все равно потребляет мощность от источника питания. Поэтому для удобства конструирования в составе серии предусматривается несколько типов ИС, выполняющих одну и ту же функцию, но отличающихся числом входящих в них логических элементов и числом входов этих элементов.

УГО некоторых ИС рассматриваемой функциональной группы приведены на рис. 3.6.

При построении схем различных устройств на ИС часто используются не все входы ИС. На эти входы, исходя из логики работы схемы, следует подать либо уровень логического 0, либо уровень логической 1. Уровень логического 0 в ИС ТТЛ подается подключением неиспользуемого входов непосредственно к общему проводу (0 В). Уровень логической 1 подается на неиспользуемые входы подключением их к источнику питания (+5 В) через токоограничивающий резистор (1-2 кОм) для защиты от скачков напряжения, возникающих, например, при включении питания. Логическую 1 можно подать также с выхода логического элемента НЕ, вход которого подключен к общему проводу. У многовходовых логических элементов неиспользуемые входы можно подключать к используемым, помня, что это приведет к соответствующему увеличению нагрузки на выход ИС, подключенный к этой группе входов.

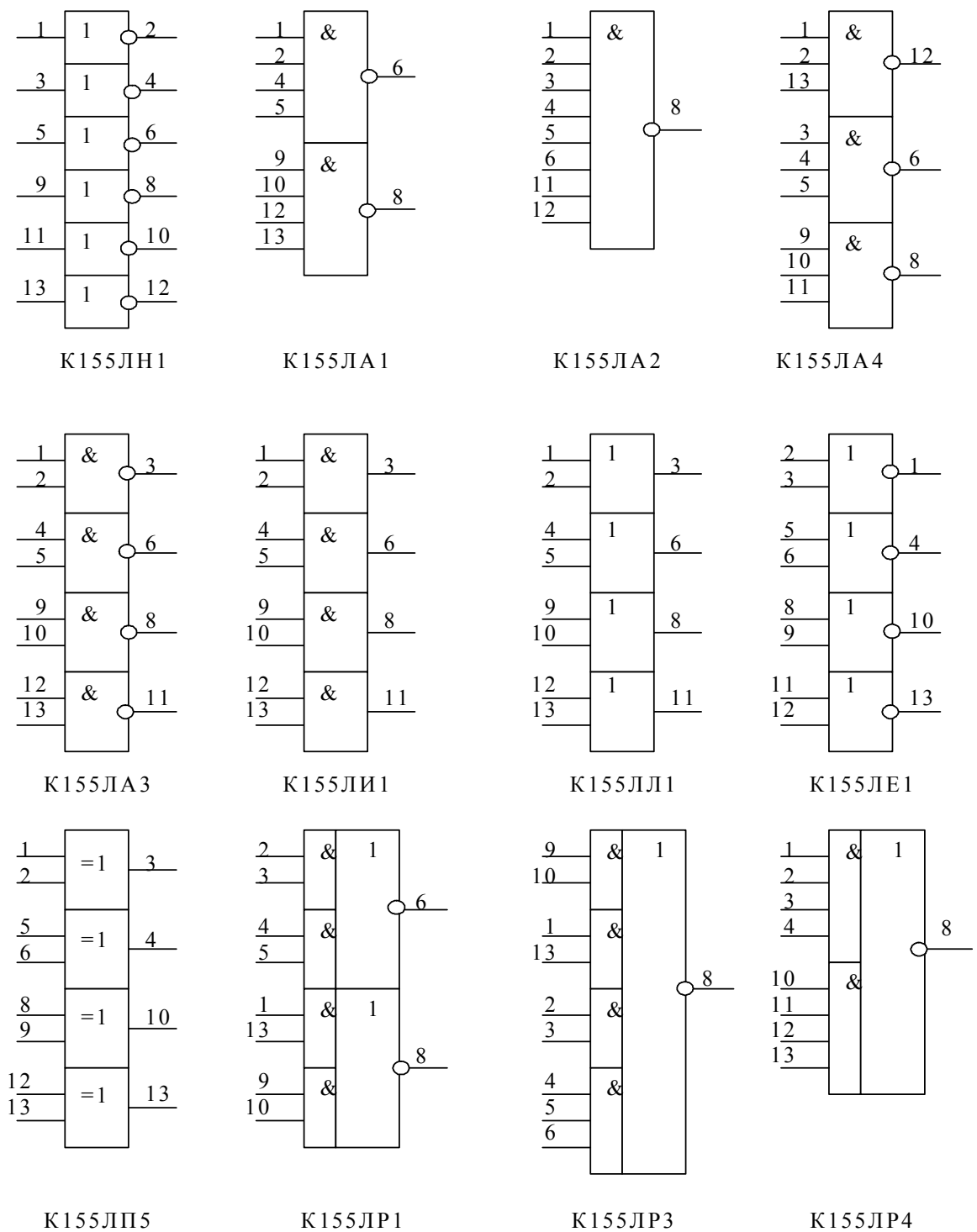


Рис. 3.6

Если вход ИС ТТЛ не подключен ни к общему проводу, ни к источнику питания («висит в воздухе»), то логический элемент будет работать так, как будто на этот вход подан уровень логической единицы, но надежность работы будет низкой из-за импульсных помех, вызванных переключением соседних входов и выходов, которые могут привести к непредусмотренному срабатыванию ИС.

Функциональные возможности и особенности применения ИС зависят от выполнения выходного каскада ИС. Наиболее часто используются четыре типа схем выходных каскадов.

1. Схема ИС со стандартным выходным каскадом приведена на рис. 3.7, которая отличается от схемы на рис. 1.1 тем, что на ней не показана вся остальная часть схемы, кроме выходного каскада.

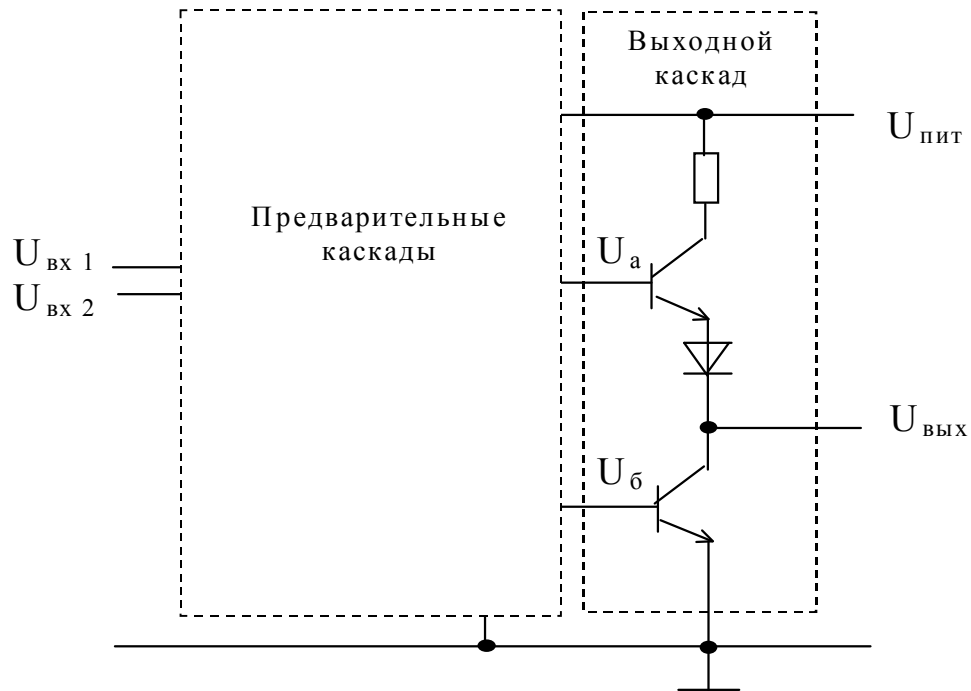


Рис. 3.7

В этой схеме напряжения  $U_a$  и  $U_b$  всегда изменяются в противофазе. Это означает, что в паре транзисторов выходного каскада один закрыт, другой открыт. Такой выходной каскад обеспечивает большой выходной (втекающий) ток при логическом 0 на выходе  $I_{\text{вых}}^0$ , когда открыт нижний транзистор, и значительно меньший выходной (вытекающий) ток при логической 1 на выходе  $I_{\text{вых}}^1$ , когда открыт верхний транзистор. Поскольку такая схема выходного каскада является стандартной, то на УГО ИС это никак не обозначается.

2. Схему ИС с открытым коллектором можно упрощенно представить так, как изображено на рис. 3.8, где в прямоугольнике скрыта вся остальная часть схемы, а выделен только выходной каскад.

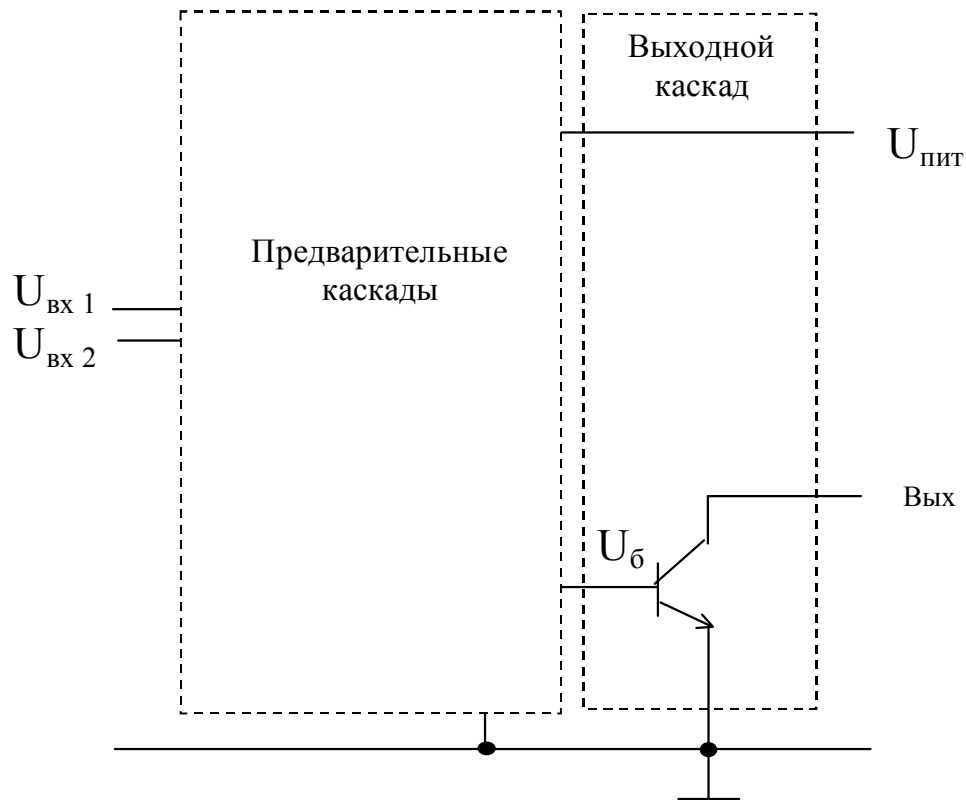


Рис. 3.8

Из рисунка видно, что в качестве выходного каскада используется транзистор, коллектор которого не подключен к источнику питания, а подключен только к внешнему выводу ИС. Выход такой ИС должен быть подключен с помощью внешнего резистора к соответствующему источнику питания. Транзисторы выходного каскада могут быть рассчитаны на разное допустимое напряжение питания (+5В, +15В, +30В, +35В), отличное от напряжения питания остальной части схемы. Выходной каскад с открытым коллектором обеспечивает большой выходной (втекающий) ток  $I_{\text{вых}}^0$ . На УГО ИС выход с открытым коллектором обозначается значком  $\diamond$ .

3. Выходные каскады с открытым эмиттером отличаются от выходных каскадов с открытым коллектором тем, что у них коллектор подключен внутри ИС обычным образом, а эмиттер внутри ИС не подключен к общему проводу, а соединен только с одним из внешних выводов ИС. Нагрузка к таким ИС подключается между выводами эмиттера и общим проводом, т.е. выходной каскад будет представлять собой эмиттерный повторитель, обеспечивающий большой выходной (вытекающий) ток  $I_{\text{вых}}^1$ . На УГО ИС выход с открытым эмиттером обозначается  $\bar{\diamond}$ .

4. На рис. 3.9 показана упрощенная схема ИС с тремя состояниями на выходе.



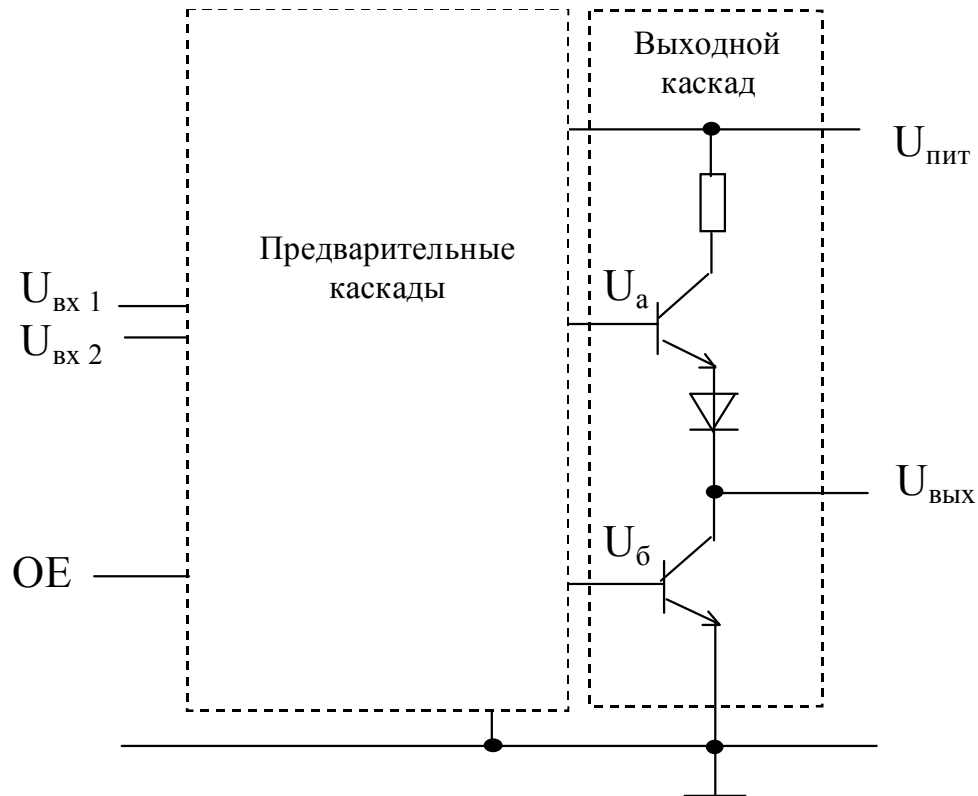


Рис. 3.9

Она отличается от схемы со стандартным выходом тем, что в ней напряжения  $U_a$  и  $U_b$  могут быть как в противофазе, и тогда она работает как схема со стандартным выходом, т.е. имеет два состояния на выходе, так и в фазе, когда оба напряжения  $U_a$  и  $U_b$  принимают низкий (закрывающий) уровень, что делает невозможным протекание выходных токов через транзисторы. Фактически это означает, что вывод выхода ИС отключен от остальной части схемы ИС. В этом случае говорят, что он находится в третьем состоянии (высокоимпедансном, высокоомном,  $z$ -состоянии). Управление переходом выходного каскада из стандартного режима в третье состояние осуществляется специальным управляющим сигналом OE (Output Enable – разрешение выхода). На УГО ИС эта особенность выходного каскада отмечается значком  $\diamond$ , реже – Z.

При построении цифровых устройств на базе ИС часто возникает необходимость объединения выходов нескольких ИС с целью перехода на общую выходную цепь. Эта задача в зависимости от особенностей выходных каскадов ИС, объединяемых по выходу, решается разными способами. Если выходы объединяемых ИС являются стандартными, то их объединение следует выполнять с помощью дополнительного логического элемента ИЛИ (рис. 3.10). При этом приходится мириться с дополнительными затратами и увеличением задержки, поскольку непосредственное соединение выходов ИС ТТЛ со стандартным выходным каскадом **недопустимо!**

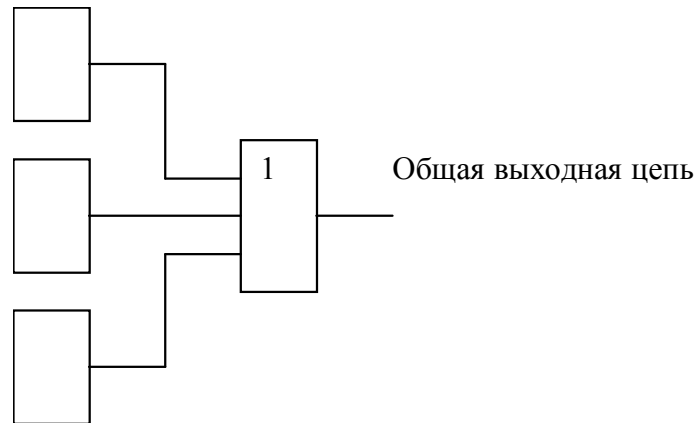


Рис. 3.10

ИС с открытым коллектором могут быть соединены друг с другом и через общий внешний резистор подключены к источнику питания (рис. 3.11). Такое включение эквивалентно использованию дополнительного логического элемента ИЛИ, поэтому его иногда называют «проводным» или «монтажным» ИЛИ. Аналогично могут быть организовано объединение выходных цепей с открытым эмиттером.

Выходы ИС с тремя состояниями выхода также могут быть объединены по схеме «монтажного» ИЛИ (рис. 3.12).

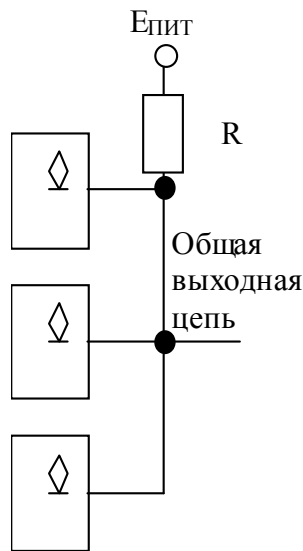


Рис. 3.11

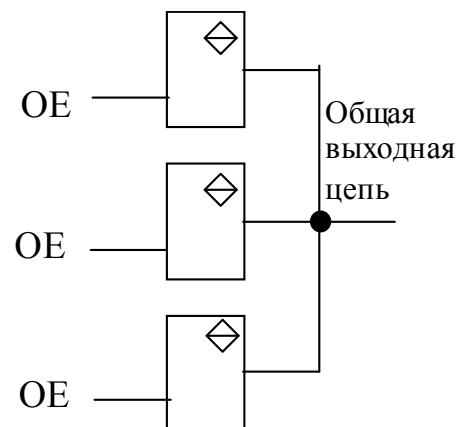


Рис. 3.12

Однако, в отличие от реализации этой функции на ИС с открытым коллектором, здесь следует учитывать существенное ограничение: в любой момент времени к общей выходной цепи должен быть подключен выход только одной ИС, а выходы всех остальных ИС должны находиться в третьем состоянии. Это достигается соответствующей организацией подачи сигналов на входы OE каждой из ИС.

## ЛЕКЦИЯ 4. ТИПОВЫЕ КОМБИНАЦИОННЫЕ УСТРОЙСТВА

Устройства, оперирующие с двоичной информацией, подразделяются на два класса: комбинационные (или автоматы без памяти) и последовательностные (или автоматы с памятью).

Комбинационные устройства не обладают памятью в том смысле, что сигналы на их выходах в любой момент времени однозначно определяются сочетанием сигналов на их входах в этот же момент времени и не зависят от предыдущих состояний.

Рассмотренные в предыдущей лекции логические элементы являются простейшими комбинационными устройствами.

Комбинационные устройства отличаются большим разнообразием, однако среди них можно выделить ряд типовых, наиболее часто встречающихся на практике и реализуемых в виде отдельных ИС в различных сериях микросхем.

К таким комбинационным устройствам можно отнести: дешифраторы, шифраторы, преобразователи кодов, мультиплексоры, демультимплексоры, сумматоры, компараторы, схемы формирования и контроля разряда паритета и множество других схем.

**Дешифратор** (декодер) - это комбинационное устройство с несколькими входами и несколькими выходами, у которого каждой комбинации входных сигналов соответствует активный уровень на одном из выходов. Классический дешифратор преобразует  $n$ -разрядный двоичный код в унитарный код, т.е. код, содержащий  $2^n$  разрядов, только один из которых равен 1, при этом номер этого разряда является десятичным эквивалентом двоичной комбинации, поданной на  $n$  входов дешифратора. Сказанное можно считать словесным заданием функции дешифратора.

Для аналитического задания положим, что число входов дешифратора  $n=3$ , тогда число выходов  $2^n = 8$ . Обозначим входы  $x_0, x_1, x_2$ , а выходы -  $y_0 - y_7$ . Тогда аналитическое задание функции дешифратора такой размерности будет выглядеть следующим образом:

$$y_0 = \bar{x}_2 \cdot \bar{x}_1 \cdot \bar{x}_0, \quad y_1 = \bar{x}_2 \cdot \bar{x}_1 \cdot x_0, \quad y_2 = \bar{x}_2 \cdot x_1 \cdot \bar{x}_0, \quad y_3 = \bar{x}_2 \cdot x_1 \cdot x_0, \\ y_4 = x_2 \cdot \bar{x}_1 \cdot \bar{x}_0, \quad y_5 = x_2 \cdot \bar{x}_1 \cdot x_0, \quad y_6 = x_2 \cdot x_1 \cdot \bar{x}_0, \quad y_7 = x_2 \cdot x_1 \cdot x_0.$$

В соответствии с этим аналитическим описанием может быть построена схема дешифратора размерностью  $3 \times 8$  (рис. 4.1). Его УГО представлено на рис. 4.2.

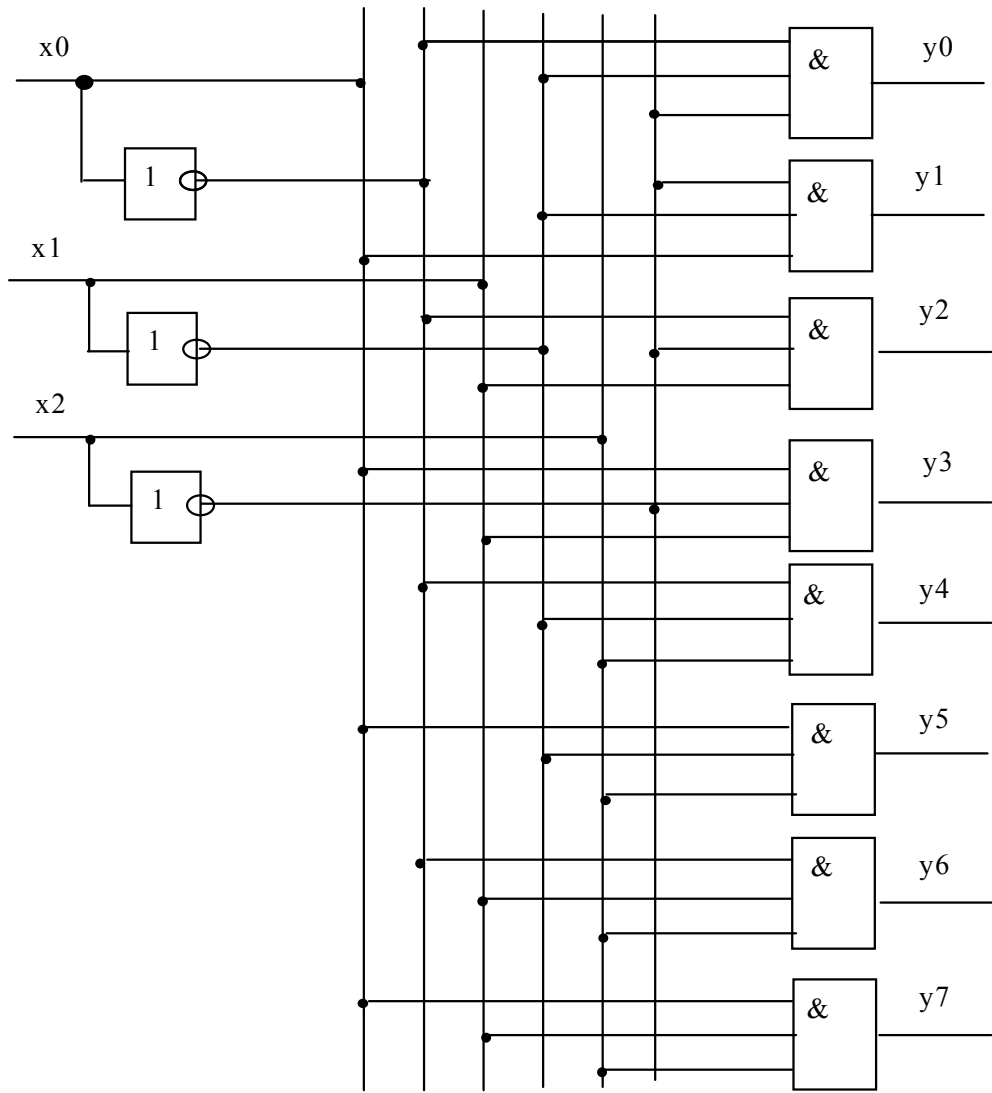


Рис. 4.1

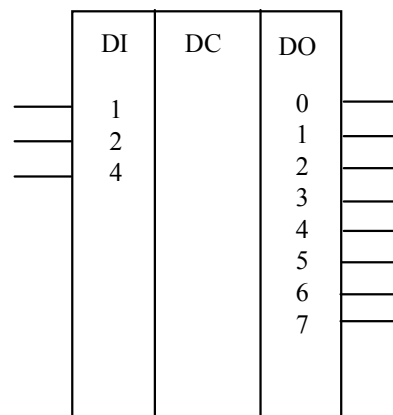


Рис. 4.2

Дешифраторы как самостоятельные функциональные узлы в виде отдельных микросхем входят в состав многих серий ИС.

Кроме информационных или адресных входов  $x$ , большинство дешифраторов снабжаются одним или несколькими разрешающими или стробирующими входами  $E$ . Их наличие может быть отображено и в аналитическом описании путем включения в каждую конъюнкцию четвертого элемента  $E$ . Это означает, что при разрешающем сигнале на входе  $E$  дешифратор выполняет свою функцию, а при запрещающем - дешифратор блокирован, т.е. на всех его выходах пассивные уровни вне зависимости от комбинации на адресных входах.

Основное назначение входа стробирования состоит в обеспечении возможности с его помощью синхронизировать работу дешифратора с работой остальных функциональных узлов, входящих в какое-либо устройство. Так, например, при асинхронном изменении значений отдельных разрядов при переходе от одной входной кодовой комбинации к другой могут возникнуть короткие (время переходного процесса) импульсы на тех выходах дешифратора, на которых они не должны появляться ни при предыдущей, ни при последующей входной комбинации. Избежать появления этих ложных импульсов на выходах позволяет подача стробирующего сигнала только после завершения переходного процесса на входах дешифратора.

Кроме того, вход стробирования может быть использован при наращивании разрядности дешифраторов. Дешифраторы как самостоятельные ИС имеют не более 16 выходов. При необходимости построения дешифратора с большим числом выходов используется пирамидальный принцип наращивания разрядности, когда в основании пирамиды устанавливается столько дешифраторов, сколько необходимо, чтобы получить требуемое число выходов. Одноименные адресные входы этих дешифраторов объединяются и образуют группу младших адресных входов синтезируемого дешифратора. В вершине пирамиды устанавливается дешифратор со столькими выходами, сколько дешифраторов установлено в основании пирамиды. Выходы этого дешифратора соединяются с входами стробирования соответствующих дешифраторов основания пирамиды, а его входы образуют группу старших адресных входов синтезируемого дешифратора.

Дешифраторы могут быть использованы для реализации логических функций, число переменных которых равно числу адресных входов дешифратора. Поскольку система уравнений, описывающих работу дешифратора, представляет собой все возможные конъюнкции от входных переменных в их прямом или инверсном виде, то для реализации логической функции, представленной в СДНФ, достаточно выполнить дизъюнкцию над соответствующими выходами дешифратора.

Дешифраторы, выполненные в виде самостоятельных микросхем различаются числом информационных входов и наличием или отсутствием входа (входов) стробирования. Выходы у большинства ИС дешифраторов выпол-

няться инверсными. Микросхема К155ИД3 (рис.4.3) служит для преобразования четырехразрядного двоичного кода в код «1 из 16», т.е. размерность дешифратора 4×16. Микросхема имеет четыре информационных входа DI1, DI2, DI4, DI8, два инверсных входа стробирования E1 и E2, объединенных по логическому И, и 16 инверсных выходов DO0 – DO15. В зависимости от способа включения ИС может работать как дешифратор и как демультиплексор.

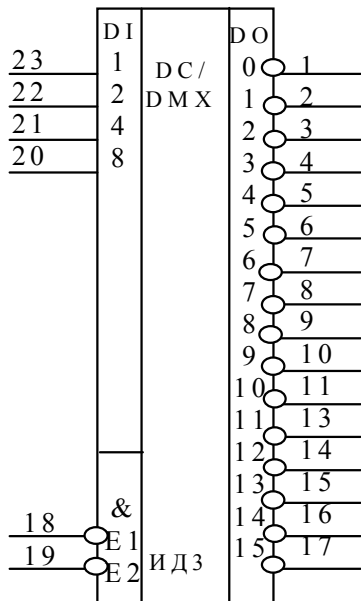


Рис. 4.3

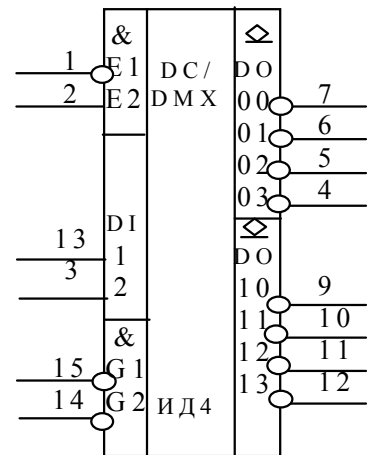


Рис. 4.4

Микросхема К155ИД4 (рис. 4.4) содержит в одном корпусе два дешифратора–демультиплексора каждый размерностью 2×4. Каждая секция имеет по два объединенных по логическому И входа стробирования и по четыре инверсных выхода DO0 – DO3. Информационные входы DI1 и DI2 – общие для обеих секций, т.е. объединены внутри микросхемы. Различие между секциями состоит в том, что у одной из них оба стробирующих входа являются инверсными (G1,G2), а у второй – один прямой (E2), а другой инверсный (E1). В зависимости от схемы включения ИС может быть использована в следующих режимах: два дешифратора 2×4, один дешифратор 3×8, два демультиплексора 1:4, один демультиплексор 1:8. При использовании К155ИД4 следует обращать внимание на то, что выходы у этой ИС с открытым коллектором.

**Шифратором** (кодером) называется комбинационное устройство, преобразующее унитарный код, подаваемый на входы, в соответствующий двоичный или двоично-десятичный код на выходах. Таким образом, шифратор реализует функцию, обратную функции дешифратора. Если с выходов шифратора снимается  $n$ -разрядный двоичный код, то максимальное число входов должно быть равно  $2^n$ . При этом выходной код представляет собой двоичный эквивалент номера входа, на котором активный уровень сигнала. Сказанное можно считать словесным заданием функции шифратора. На основании это-

го словесного задания, положив число входов шифратора равным  $8 = 2^3$  и обозначив их I0 - I7, составим таблицу истинности (табл. 4.1) шифратора, в которой число выходов шифратора равно 3 и обозначены они A0 - A2.

Таблица 4.1

I0	I1	I2	I3	I4	I5	I6	I7	A0	A1	A2
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	0	1	1
0	0	0	0	0	0	0	1	1	1	1

На основании таблицы истинности может быть построена схема шифратора (рис. 4.5). УГО приведенной схемы шифратора выглядит следующим образом (рис. 4.6).

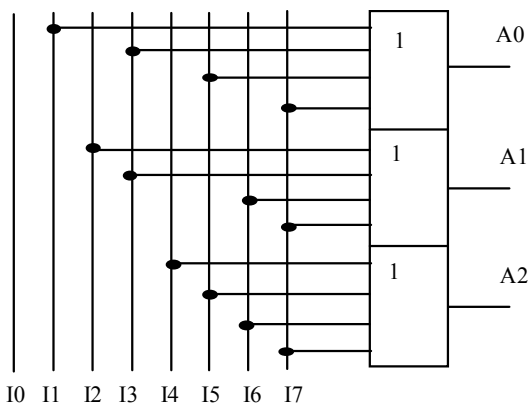


Рис. 4.5

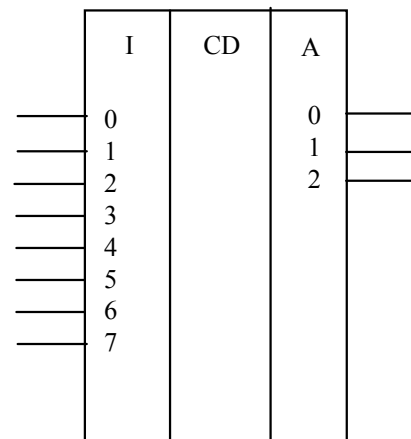


Рис. 4.6

Неполнота приведенной таблицы истинности обусловлена требованием подачи на входы шифратора только комбинаций унитарного кода. Для того чтобы это условие выполнялось всегда при любых источниках входного кода, большинство шифраторов, реализуемых в виде ИС, делают приоритетными. Это означает, что на входы такого шифратора может быть подан любой код. Встроенная схема приоритета или схема выявления старшей 1 преобразует его в унитарный, выделив лишь самую старшую 1 в поданной на входы комбинации и проигнорировав все остальные 1.

Шифраторы, выполненные в виде самостоятельных микросхем, чаще всего бывают приоритетными, отличаясь числом информационных входов, входов стробирования, наличием дополнительных служебных выходов, обеспе-

чивающих возможность наращивания разрядности, а также характером сигналов (прямые или инверсные) на всех названных входах и выходах.

Микросхема К155ИВ1 (рис. 4.7) представляет собой приоритетный шифратор 8:3, имеющий восемь инверсных информационных входов I0-I7; инверсный вход стробирования E; три инверсных информационных выхода A0-A2; инверсный выход G, сигнал на котором (0) свидетельствует о наличии хотя бы на одном из информационных входов активного уровня (0) при наличии разрешающего уровня (0) на входе стробирования E; инверсный выход E0, сигнал на котором (0) свидетельствует об отсутствии хотя бы на одном из информационных входов активного уровня (0) при наличии разрешающего уровня (0) на входе стробирования E. Выход G может быть использован для разрешения работы последующих узлов, для которых выходной код шифратора является входным. Выход E0 используется при наращивании разрядности шифраторов, разрешая или запрещая работу младших микросхем и сохраняя тем самым приоритетность схемы.

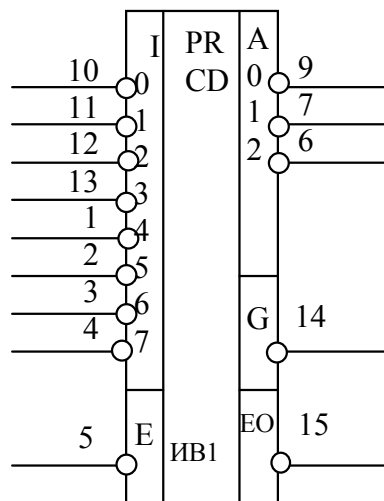


Рис. 4.7

Эти свойства ИС К155ИВ1 отражены в таблице истинности (табл. 4.2).

Таблица 4.2

E	I7	I6	I5	I4	I3	I2	I1	I0	A2	A1	A0	G	E0
1	*	*	*	*	*	*	*	*	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0	1	1	1	0	1
0	1	1	1	1	1	0	*	*	1	0	1	0	1
0	1	1	1	1	0	*	*	*	1	0	0	0	1
0	1	1	1	0	*	*	*	*	0	1	1	0	1
0	1	1	0	*	*	*	*	*	0	1	0	0	1
0	1	0	*	*	*	*	*	*	0	0	1	0	1
0	0	*	*	*	*	*	*	*	0	0	0	0	1



**Мультиплексором** называется комбинационное устройство, предназначенное для коммутации в желаемом порядке сигналов с нескольких входов на единственный выход. В этом смысле можно уподобить мультиплексор многопозиционному переключателю.

Входы мультиплексора по функциональному назначению делятся на информационные, адресные (или селектирующие) и разрешающие (или стробирующие). Подачей на селектирующие входы кодовой комбинации осуществляется выбор соответствующего этой комбинации информационного входа, подключаемого к выходу мультиплексора, а сигнал на стробирующем входе разрешает это подключение. У большинства мультиплексоров реализуется следующее правило выбора: к выходу подключается тот информационный вход, номер которого в двоичном коде подан на селектирующие входы.

Сказанное можно считать словесным заданием функции мультиплексора. Для аналитического задания положим число селектирующих входов  $n=3$  и обозначим их  $A1 - A3$ . Тогда число информационных входов равно  $2^n = 8$ . Обозначим их  $D0 - D7$ . С учетом этого аналитическое выражение для функции, реализуемой на выходе  $F$  мультиплексора, можно записать следующим образом:

$$F = D0 \cdot \bar{A1} \cdot \bar{A2} \cdot \bar{A3} + D1 \cdot A1 \cdot \bar{A2} \cdot \bar{A3} + D2 \cdot \bar{A1} \cdot A2 \cdot \bar{A3} + D3 \cdot A1 \cdot A2 \cdot \bar{A3} + \\ + D4 \cdot \bar{A1} \cdot \bar{A2} \cdot A3 + D5 \cdot A1 \cdot \bar{A2} \cdot A3 + D6 \cdot \bar{A1} \cdot A2 \cdot A3 + D7 \cdot A1 \cdot A2 \cdot A3.$$

В соответствии с этим выражением схему мультиплексора можно представить следующим образом (рис. 4.8). УГО этой схемы приведено на рис. 4.9.

Помимо указанных ранее входов, мультиплексоры, как и дешифраторы, могут иметь входы стробирования, имеющие то же функциональное назначение – обеспечение синхронизации работы мультиплексора с другими функциональными узлами, входящими в состав устройства.

Так же как и у дешифраторов, входы стробирования могут использоваться для наращивания разрядности мультиплексоров с использованием пирамидального принципа построения.

Мультиплексоры, будучи предельно универсальными логическими элементами, могут использоваться для реализации логических функций. С помощью мультиплексора может быть реализована любая логическая функция от  $n+1$  переменной, где  $n$  - число селектирующих входов мультиплексора. Для этого  $n$  переменных подается на селектирующие входы мультиплексора, а на его информационные входы подаются или константа 0, или константа 1, или прямое значение  $(n+1)$ -й переменной, или ее инверсное значение в соответствии с таблицей истинности реализуемой функции.

Мультиплексоры, выполненные в виде самостоятельных микросхем, различаются числом информационных и селектирующих входов, наличием или отсутствием входа (входов) стробирования, а также характером выходных

сигналов, которые могут быть прямыми или (и) инверсными относительно входных информационных.

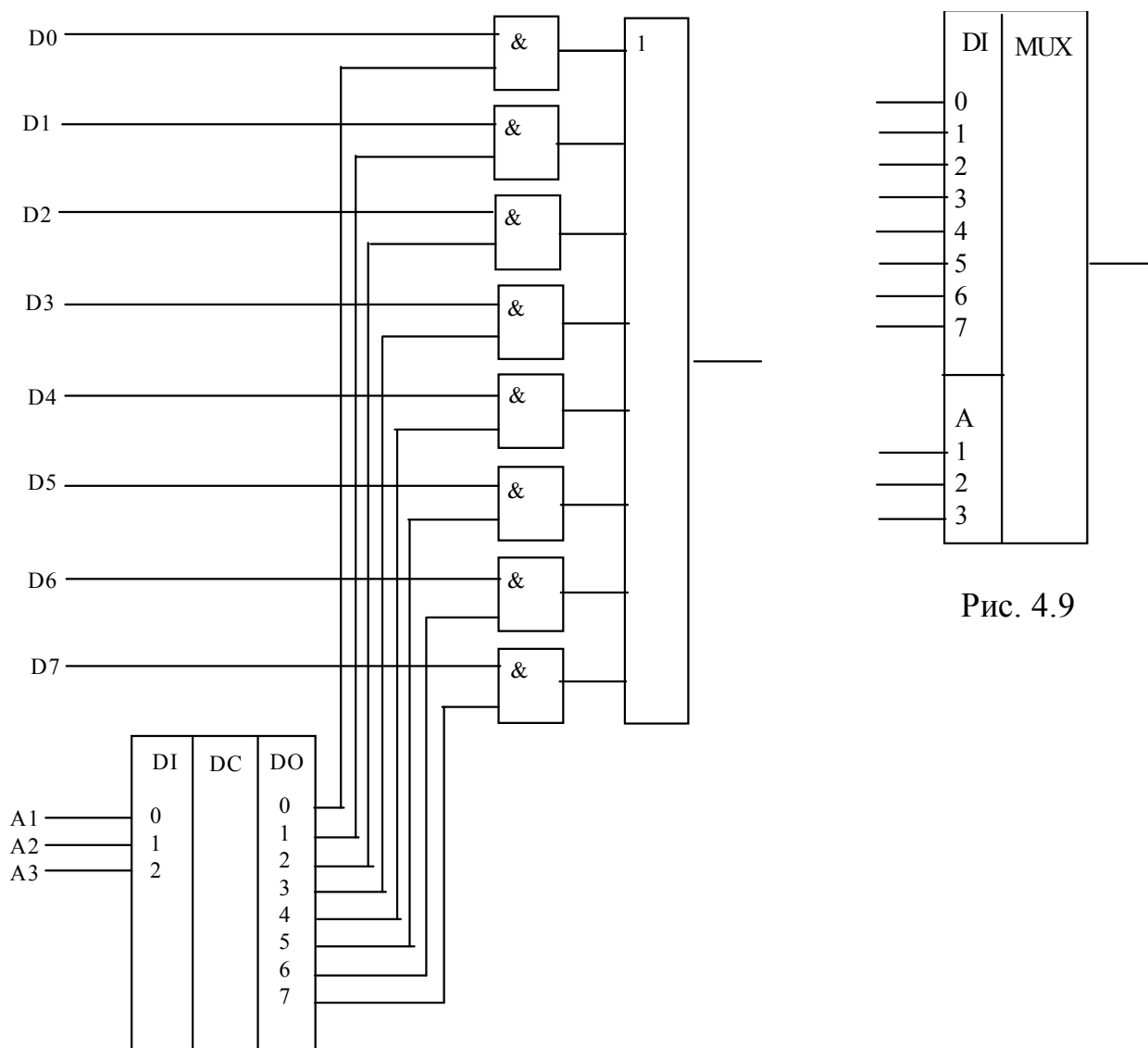


Рис. 4.9

Рис. 4.8

Микросхема К155КП1 (рис. 4.10) имеет 16 информационных входов (DI0-DI15), четыре селектирующих входа (A1, A2, A4, A8), один инверсный вход стробирования E и один инверсный выход.

Микросхема К155КП2 (рис. 4.11) содержит в одном корпусе два мультиплексора вида 4:1. Каждая секция имеет четыре информационных входа DI0 - DI3, инверсный вход стробирования E и прямой выход. Два селектирующих входа A1 и A2 – общие для обеих секций.

Микросхемы К155КП7 (рис. 4.12) и К155КП5 (рис. 4.13) представляют собой мультиплексоры вида 8:1. Они близки по логической структуре и принципу действия. Различие состоит в том, что К155КП7 имеет инверсный

вход стробирования E и два выхода – прямой и инверсный, тогда как в К155КП5 вход стробирования и прямой выход отсутствуют.

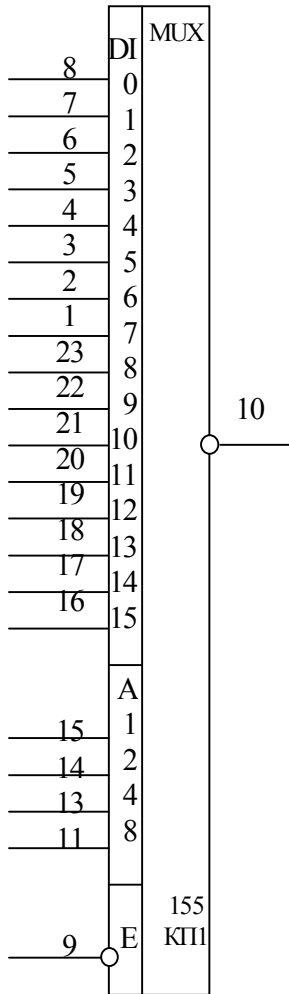


Рис. 4.10

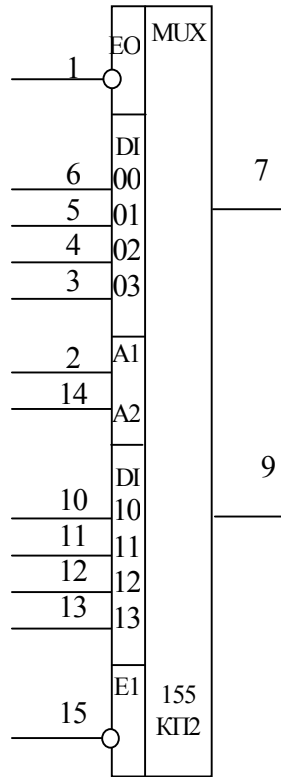


Рис. 4.11

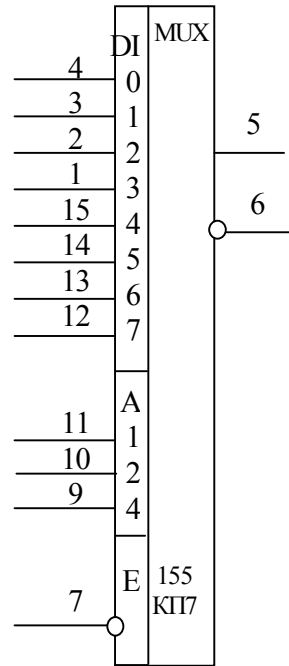


Рис. 4.12

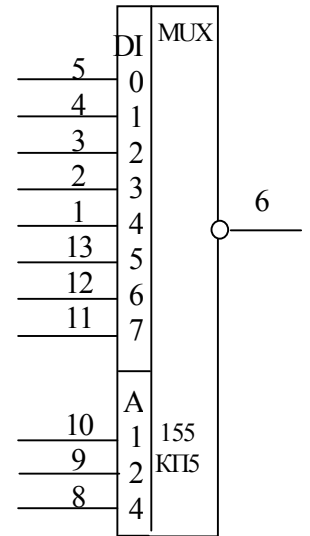


Рис. 4.13

Исходя из названия, в функциональном отношении **демультиплексор** реализует функцию, обратную функции мультиплексора. В соответствии с этим можно выполнить словесное задание функции демультиплексора. В нем сигнал с единственного информационного входа передается на один из нескольких выходов в зависимости от кода, поданного на адресные или селектирующие входы. При  $n$  адресных входах демультиплексор может иметь до  $2^n$  выходов. Положим  $n=3$  и обозначим адресные входы  $X_0, X_1, X_2$ , а  $2^n = 8$  выходов обозначим  $Y_0 - Y_7$ . Обозначим информационный вход демультиплексора E. Тогда аналитическое задание функции демультиплексора такой размерности будет выглядеть следующим образом :

$$Y_0 = E \cdot \bar{X}_0 \cdot \bar{X}_1 \cdot \bar{X}_2, Y_1 = E \cdot X_0 \cdot \bar{X}_1 \cdot \bar{X}_2, Y_2 = E \cdot \bar{X}_0 \cdot X_1 \cdot \bar{X}_2, Y_3 = E \cdot X_0 \cdot X_1 \cdot \bar{X}_2, \\ Y_4 = E \cdot \bar{X}_0 \cdot \bar{X}_1 \cdot X_2, Y_5 = E \cdot X_0 \cdot \bar{X}_1 \cdot X_2, Y_6 = E \cdot \bar{X}_0 \cdot X_1 \cdot X_2, Y_7 = E \cdot X_0 \cdot X_1 \cdot X_2.$$

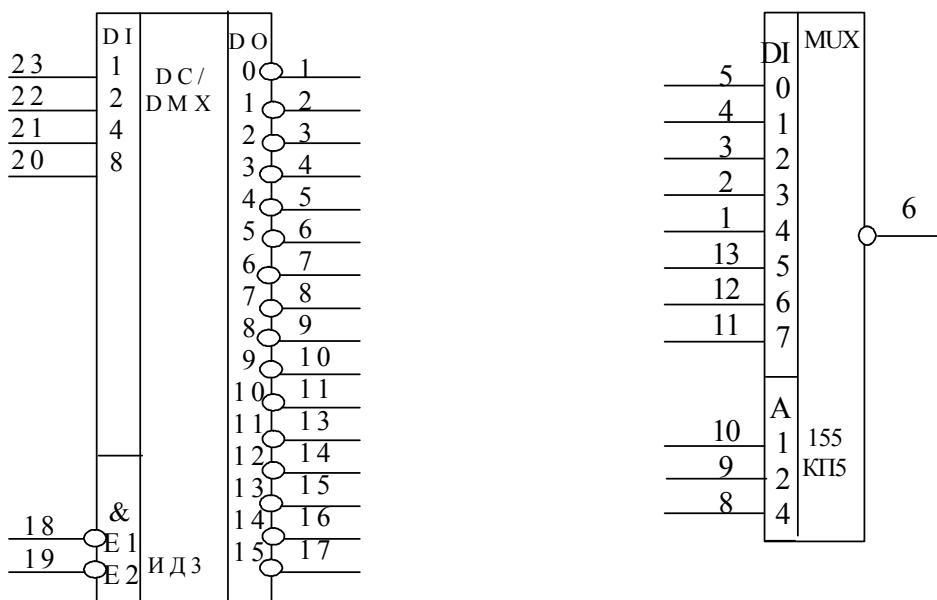
Но эта система описывает дешифратор с тремя входами  $X_0, X_1, X_2$ , входом стробирования E и восемью выходами  $Y_0 - Y_7$ . С учетом сказанного мож-

но рассматривать дешифратор с входом стробирования как обращенный по входам демультиплексор, у которого селектирующие входы стали адресными входами дешифратора, а информационный вход - входом стробирования дешифратора.

### Контрольная работа №3

Логическая функция задана аналитически в виде, например, следующего выражения  $f(x) = \bar{x}_4x_3x_2\bar{x}_1 + \bar{x}_4x_3\bar{x}_2x_1 + \bar{x}_4\bar{x}_3x_2\bar{x}_1 + \bar{x}_4\bar{x}_3\bar{x}_2x_1$ . Реализовать заданную функцию с помощью:

- дешифратора К155ИД3 и необходимых логических элементов;
- мультиплексора К155КП5 и необходимых логических элементов.



**Сумматоры** представляют собой функциональные цифровые устройства, выполняющие сложение чисел. Суммирование осуществляется в двоичном коде. По характеру действия сумматоры подразделяются на комбинационные и накапливающие, т.е. сохраняющие результат в специальном регистре. Каждый из многоразрядных сумматоров может быть отнесен в зависимости от способов сложения к параллельному или последовательному типу. В последовательных сумматорах сложение выполняется поразрядно и последовательно во времени, а в сумматорах параллельного типа все разряды суммируются одновременно. И те, и другие сумматоры строятся на основе одноразрядных полных сумматоров. Сложение выполняется в каждом разряде от-

дельно, но с учетом результата сложения в предыдущем разряде, т.е. с учетом переноса. Таким образом, каждый одноразрядный полный сумматор должен иметь один вход переноса  $P_0$ , два входа слагаемых  $A$  и  $B$ , выход суммы  $S$  и выход переноса  $P_1$ . Таблица истинности одноразрядного полного сумматора (табл. 4.3) строится с учетом правил сложения одноразрядных двоичных чисел.

Таблица 4.3

A	B	$P_0$	S	$P_1$
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

На основании этой таблицы схема одноразрядного полного сумматора может быть синтезирована описанными ранее методами с использованием различных логических элементов. Один из возможных вариантов такой реализации приведен на рис. 4.14а. На рис. 4.14б представлено условное графическое обозначение одноразрядного полного сумматора.

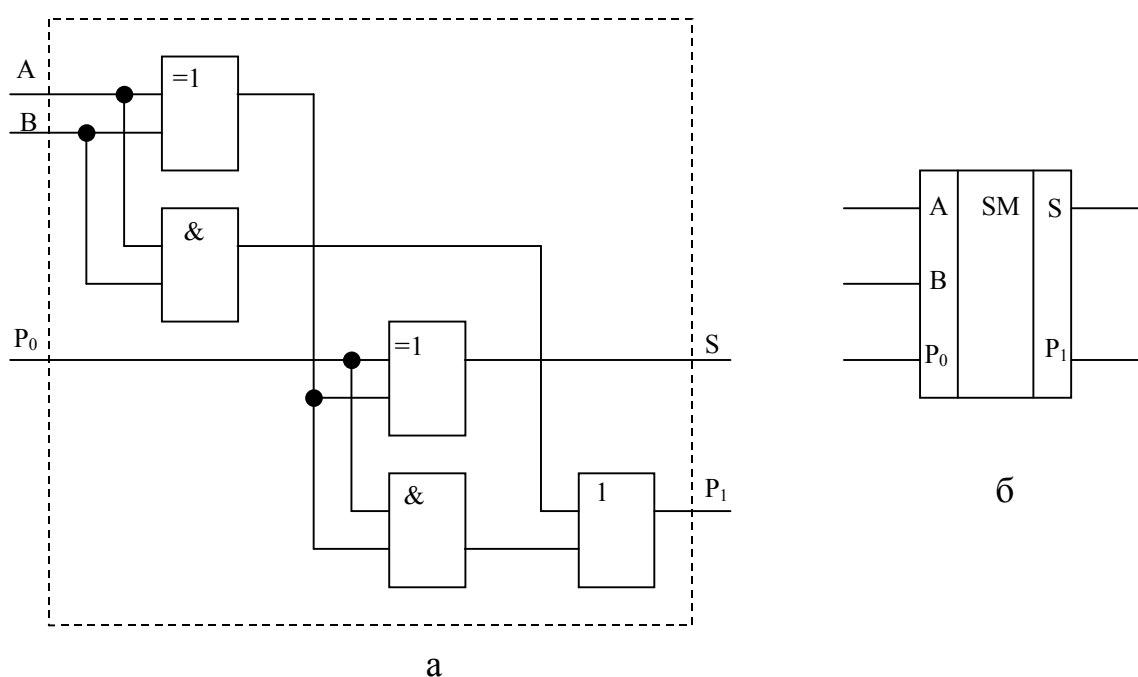


Рис. 4.14

Для построения многоразрядного параллельного сумматора с последовательным переносом необходимо соответствующее количество одноразряд-

ных полных сумматоров соединить последовательно по цепи переноса (рис. 4.15).

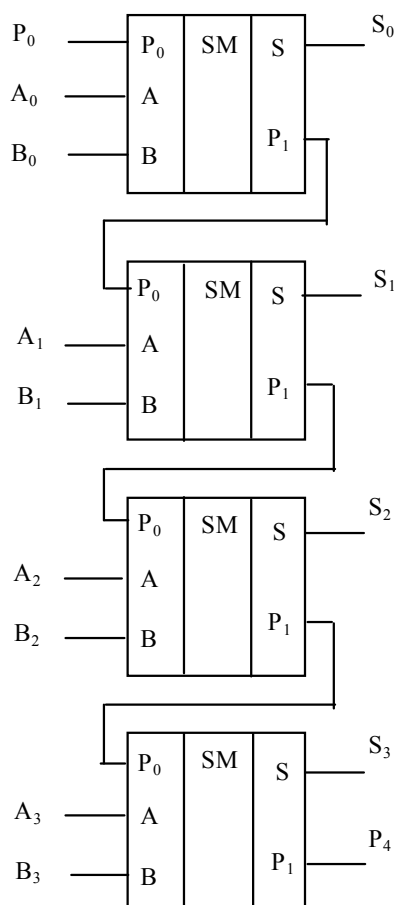


Рис. 4.15

Аналогично осуществляется наращивание разрядности при использовании микросхем многоразрядных сумматоров.

Сумматоры, выполненные в виде самостоятельных микросхем, различаются в основном разрядностью суммируемых двоичных чисел.

Микросхема К155ИМ2 (рис. 4.16) представляет собой полный двухразрядный сумматор, а микросхема К155ИМ3 (рис. 4.17) – полный четырехразрядный сумматор. Принцип их действия одинаков и основан на параллельном суммировании данных в соответствующих разрядах операндов А и В с помощью одноразрядного полного сумматора, построенного по схеме, аналогичной рис. 4.14, при последовательном переносе из разряда в разряд (рис. 4.15). У названных микросхем вход переноса есть только у младшего разряда (С0), а выход – только у старшего (С2 – у ИМ2 и С4 – у ИМ3), что обеспечивает возможность наращивания разрядности сумматоров. Для этого выход переноса микросхемы, на которую подаются младшие разряды операндов А и В, непосредственно соединяется со входом переноса микросхемы, обрабатывающей старшие разряды операндов.

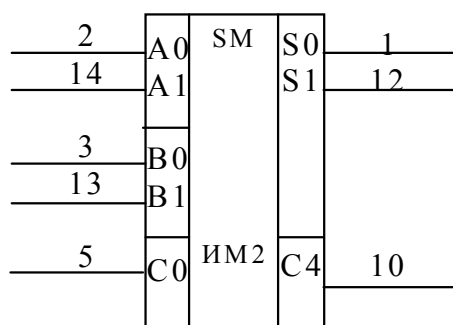


Рис. 4.16

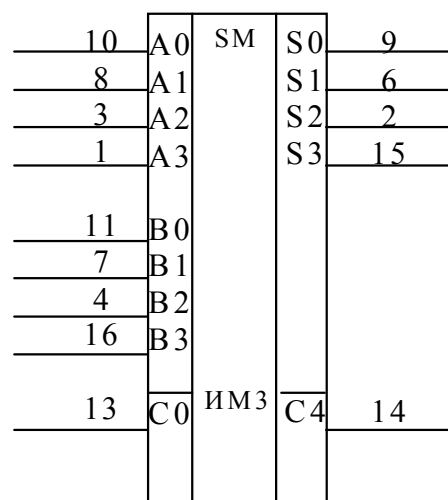


Рис. 4.17

В цифровой технике операция вычитания с использованием заема практически не применяется. Одним из способов выполнения операции вычитания с помощью двоичного сумматора является замена знака вычитаемого на противоположный и прибавление его к уменьшаемому  $A-B=A+(-B)$  с применением специальных кодов для представления отрицательных чисел. Для представления чисел со знаком используются прямой, обратный и дополнительный коды. Во всех этих кодах предусматривается дополнительный разряд (старший, т.е. крайний слева) для представления знака числа, причем знак «+» кодируется символом 0, а знак «-» - символом 1.

Прямой код как положительного, так и отрицательного числа отличается от кода этого числа без знака только наличием знакового разряда с соответствующим его значением. Прямой код не обеспечивает возможности замены операции вычитания операцией алгебраического сложения и поэтому не находит практического применения.

Обратный код положительных двоичных чисел совпадает с прямым кодом. Обратный код отрицательных двоичных чисел формируется по следующему правилу: цифры всех разрядов прямого кода числа, кроме знакового, инвертируются. Обратное преобразование из обратного кода в прямой производится по тому же правилу. Примеры выполнения операций алгебраического суммирования с использованием обратного кода приведены в табл. 4.4, где символом  $P_{\text{зн}}$  обозначен перенос из знакового разряда.

Недостатком обратного кода является наличие кругового переноса, состоящего в том, что перенос из знакового разряда должен быть подсуммирован к младшему разряду результата, что снижает быстродействие сумматора. Кроме того, число «нуль» в обратном коде имеет два представления – «положительный нуль» -  $(0\ 0\ \dots\ 0)$  и «отрицательный нуль» -  $(1\ 1\ \dots\ 1)$ .

Таблица 4.4

	Пример 1						Пример 2					
	$\{X\}_{10}$	$\{X\}_2$ ОБР					$\{X\}_{10}$	$\{X\}_2$ ОБР				
		$P_{3н}$	знак	3	2	1		$P_{3н}$	знак	3	2	1
1-е слагаемое	+5		0	1	0	1	+3		0	0	1	1
2-е слагаемое	-3		1	1	0	0	-5		1	0	1	0
Промежуточная сумма		1	0	0	0	1		0	1	1	0	1
Круговой перенос						1						0
Сумма	+2		0	0	1	0	-2		1	1	0	1

Дополнительный код свободен от обоих указанных недостатков обратного кода, поскольку 1 переноса из знакового разряда отбрасывается, т.е. круговой перенос отсутствует, а число «ноль» представляется однозначно в виде «положительного нуля» - (0 0 . . . 0). Дополнительный код положительных двоичных чисел совпадает с прямым кодом. Дополнительный код отрицательных двоичных чисел формируется по следующему правилу: цифры всех разрядов прямого кода числа, кроме знакового разряда, инвертируются, а к младшему разряду подсуммируется 1. Обратное преобразование из дополнительного кода в прямой код производится по тому же правилу. Примеры выполнения операций алгебраического суммирования с использованием дополнительного кода приведены в табл. 4.5.

Таблица 4.5

	Пример 1						Пример 2					
	$\{X\}_{10}$	$\{X\}_2$ ДОП					$\{X\}_{10}$	$\{X\}_2$ ДОП				
		$P_{3н}$	знак	3	2	1		$P_{3н}$	знак	3	2	1
1-е слагаемое	+5		0	1	0	1	+3		0	0	1	1
2-е слагаемое	-3		1	1	0	1	-5		1	0	1	1
Промежуточная сумма		1	0	0	1	0		0	1	1	1	0
Сумма	+2		0	0	1	0	-2		1	1	1	0

Из сказанного следует, что преимущественное применение для представления чисел со знаком в цифровых устройствах находит дополнительный код.

Наряду с общей системой кодирования алфавитно-цифровой информации в цифровых устройствах используется также отдельная система кодирования только десятичных цифр. Десятичные цифры кодируются двоичными цифрами с помощью различных кодов, называемых двоично-десятичными. При использовании двоично-десятичного кода каждая цифра (разряд, декада) десятичного числа представляется в двоичной форме и изображается соответствующим четырехразрядным (тетрада) числом. Количество различных двоично-десятичных кодов определяется количеством возможных сочетаний по



10 из 16 комбинаций, которые допускает тетрада. Кодирование десятичных цифр некоторыми двоично-десятичными кодами представлено в табл. 4.6.

Таблица 4.6

Десятичная цифра	8421	С избытком 3	5421	2421
0	0000	0011	0000	0000
1	0001	0100	0001	0001
2	0010	0101	0010	0010
3	0011	0110	0011	0011
4	0100	0111	0100	0100
5	0101	1000	1000	1011
6	0110	1001	1001	1100
7	0111	1010	1010	1101
8	1000	1011	1011	1110
9	1001	1100	1100	1111

Для построения одноразрядного двоично-десятичного сумматора (т.е. для сложения тетрад), использующего любой из этих кодов, могут быть применены четырехразрядные двоичные сумматоры, рассмотренные выше. Однако при этом необходимо учитывать особенности, вызванные тем, что суммируются не четырехразрядные двоичные числа, а особое представление десятичных чисел:

1. Наличие в каждом из двоично-десятичных кодов разрешенных и запрещенных комбинаций. Появление запрещенной комбинации при выполнении каких-либо действий над числами свидетельствует о возникновении ошибки или о необходимости произвести коррекцию результата.

2. При сложении тетрад возникает потетрадный перенос, если полученная сумма больше 15, вместо подекадного переноса при получении суммы больше 9. Это приводит к необходимости коррекции результата.

Рассмотрим правила сложения применительно к коду 8421. При сложении чисел в этом коде могут возникнуть следующие случаи:

1. Если действия над разрядами тетрады производятся по правилам двоичной арифметики и полученная сумма меньше 10, а потетрадный перенос отсутствует, то правильный результат получается без коррекции.

2. Свидетельством того, что результат неправильный, является либо появление запрещенной для кода 8421 комбинации при отсутствии потетрадного переноса (сумма больше 9 и меньше 15), либо появление разрешенной комбинации при наличии потетрадного переноса (сумма больше 15), который превышает значение подекадного переноса на 6. В обоих случаях требуется коррекция результата в данной тетраде путем подсуммирования к ней корректирующей тетрады 0110.

На основании этих выводов может быть построена схема одноразрядного двоично-десятичного сумматора в коде 8421, состоящая из двух двоичных четырехразрядных сумматоров и логической схемы, выявляющей наличие указанных выше причин необходимости коррекции и формирующей в случае их выявления корректирующую тетраду. Первый сумматор используется для суммирования тетрад операндов, а второй служит для подсуммирования в случае необходимости корректирующей тетрады к результату, получаемому с выходов первого сумматора. На выходах второго сумматора получается правильный результат в коде 8421.

Схемы сумматоров, работающих в других двоично-десятичных кодах, приведенных в табл. 4.6, могут быть построены на основе аналогичных рассуждений с учетом особенностей каждого кода.

#### *Контрольная работа №4*

Заданы два целых числа:  $(\alpha)_{10}$  в десятичной системе,  $(\beta)_2$  в двоичной системе

Выполнить:

- а) преобразования  $(\alpha)_{10} \Rightarrow (\alpha)_2$ ;  $(\beta)_2 \Rightarrow (\beta)_{10}$ .
- б) суммирование  $(\alpha)_2 + (\beta)_2 = (\Sigma)_2$ ;  
преобразование  $(\Sigma)_2 \Rightarrow (\Sigma)_{10}$ ;  
суммирование  $(\alpha)_{10} + (\beta)_{10} = (\Sigma)_{10}$ ;  
сопоставление  $(\Sigma)_{10}$ .
- в) вычитание в обратном коде  $(\alpha)_{2обр} - (\beta)_{2обр} = (\Delta)_{2обр}$ ;  
преобразование  $(\Delta)_{2обр} \Rightarrow (\Delta)_{10}$ ;  
вычитание  $(\alpha)_{10} - (\beta)_{10} = (\Delta)_{10}$ ;  
сопоставление  $(\Delta)_{10}$ .
- г) вычитание в дополнительном коде  $(\alpha)_{2доп} - (\beta)_{2доп} = (\Delta)_{2доп}$ ;  
преобразование  $(\Delta)_{2доп} \Rightarrow (\Delta)_{10}$ ;  
вычитание  $(\alpha)_{10} - (\beta)_{10} = (\Delta)_{10}$ ;  
сопоставление  $(\Delta)_{10}$ .
- д) вычитание в обратном коде  $(\beta)_{2обр} - (\alpha)_{2обр} = (\rho)_{2обр}$ ;  
преобразование  $(\rho)_{2обр} \Rightarrow (\rho)_{10}$ ;  
вычитание  $(\beta)_{10} - (\alpha)_{10} = (\rho)_{10}$ ;  
сопоставление  $(\rho)_{10}$ .
- е) вычитание в дополнительном коде  $(\beta)_{2доп} - (\alpha)_{2доп} = (\rho)_{2доп}$ ;  
преобразование  $(\rho)_{2доп} \Rightarrow (\rho)_{10}$ ;  
вычитание  $(\beta)_{10} - (\alpha)_{10} = (\rho)_{10}$ ;  
сопоставление  $(\rho)_{10}$ .

Цифровыми **компараторами** или схемами сравнения называются комбинационные устройства, реализующие функцию сравнения двух двоичных

чисел одинаковой разрядности. Как правило, в функцию компаратора входит не только определение равенства двух чисел  $A$  и  $B$ , но и неравенств  $A > B$  и  $A < B$ . Результаты сравнения отображаются соответствующими уровнями на соответствующих выходах компаратора.

Примером цифрового компаратора, выполненного в виде ИС, может служить микросхема К555СП1, УГО которой приведено на рис. 4.18.

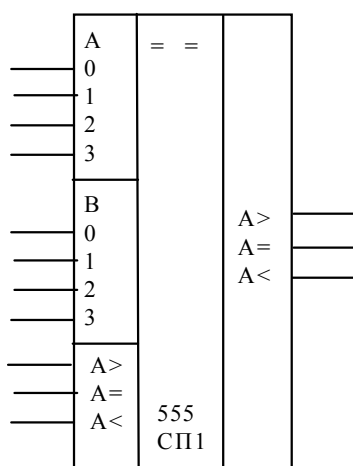


Рис. 4.18

Входы  $A <$ ,  $A =$ ,  $A >$  обеспечивают возможность наращивания разрядности компараторов путем их каскадирования без дополнительных логических элементов.

**Схемами формирования и контроля разряда паритета** называются комбинационные устройства, предназначенные для формирования на передающей стороне и контроля на приемной стороне дополнительного разряда, называемого разрядом паритета, позволяющего обнаруживать наличие ошибки в принятом двоичном слове.

Назначение разряда паритета состоит в том, чтобы доводить число единиц в каждом передаваемом слове до четного или нечетного в зависимости от принятой системы паритета. Чаще используется нечетный паритет. В этом случае содержимое разряда паритета на передающей стороне формируется таким образом, чтобы число единиц в передаваемом сообщении (слово + разряд паритета) было нечетным. На приемной стороне осуществляется контроль нечетности принятого сообщения и если она нарушена, т.е. число единиц в сообщении четно, то считается, что слово принято с ошибкой. Для построения схем формирования и контроля разряда паритета используются сумматоры по модулю 2. Примером такого устройства в интегральном исполнении может служить микросхема К155ИП2, УГО которой приведено на рис. 4.19.

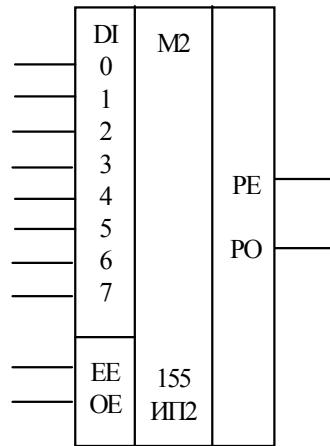


Рис. 4.19

## ЛЕКЦИЯ 5. ТИПОВЫЕ ПОСЛЕДОВАТЕЛЬНОСТНЫЕ УСТРОЙСТВА

К классу последовательностных устройств относятся цифровые устройства с памятью, в которых значения выходных сигналов определяются как значениями входных сигналов в данный момент времени, так и предысторией изменения входных сигналов. Для этих устройств характерным является то, что при одних и тех же значениях входных сигналов выходные сигналы могут иметь различное значение в зависимости от состояния устройства.

Основными типами последовательностных устройств являются триггеры и реализуемые на их основе регистры и счетчики.

К **триггерам** относится большой класс устройств, общим свойством которых является способность сколь угодно долго оставаться в одном из двух возможных устойчивых состояний и скачком переходить в другое под воздействием внешних сигналов, оставаясь в этом состоянии и после снятия сигналов, установивших его в это состояние.

Таким образом, основное назначение триггера - запоминание значения одной логической переменной или одного разряда двоичного слова или числа. Под запоминанием понимается указанная выше способность триггера оставаться в заданном устойчивом состоянии и после снятия сигнала, установившего его в это состояние.

Состояние триггера распознается по уровню на его выходе. Триггеры обычно имеют два выхода - прямой и инверсный. Принято говорить, что триггер находится в единичном состоянии, если не его прямом выходе уровень логической 1, а на инверсном - уровень логического 0.

Триггеры отличаются большим разнообразием типов и схемных решений, определяемых их функциональным назначением и способами записи в них информации.

Функциональное назначение триггеров определяется зависимостью значений на их выходах от значений входных сигналов. Практическое применение из множества возможных по функциональному назначению типов триггеров нашли RS-, D-, JK- и T-триггеры.

Их рассмотрение целесообразно начать с простейшего RS-триггера, одна из возможных схем которого приведена на рис. 5.1, здесь же представлена временная диаграмма его работы и УГО.

Классификация триггеров по способу записи информации характеризует ход процесса переключения триггера. По этому классификационному признаку триггеры подразделяются на асинхронные и синхронные.

Запись информации в асинхронные триггеры осуществляется непосредственно при поступлении сигналов на информационные входы, как в рассмотренном RS-триггере (рис. 5.1). Таким образом, по этому классификационному признаку он относится к асинхронным триггерам.

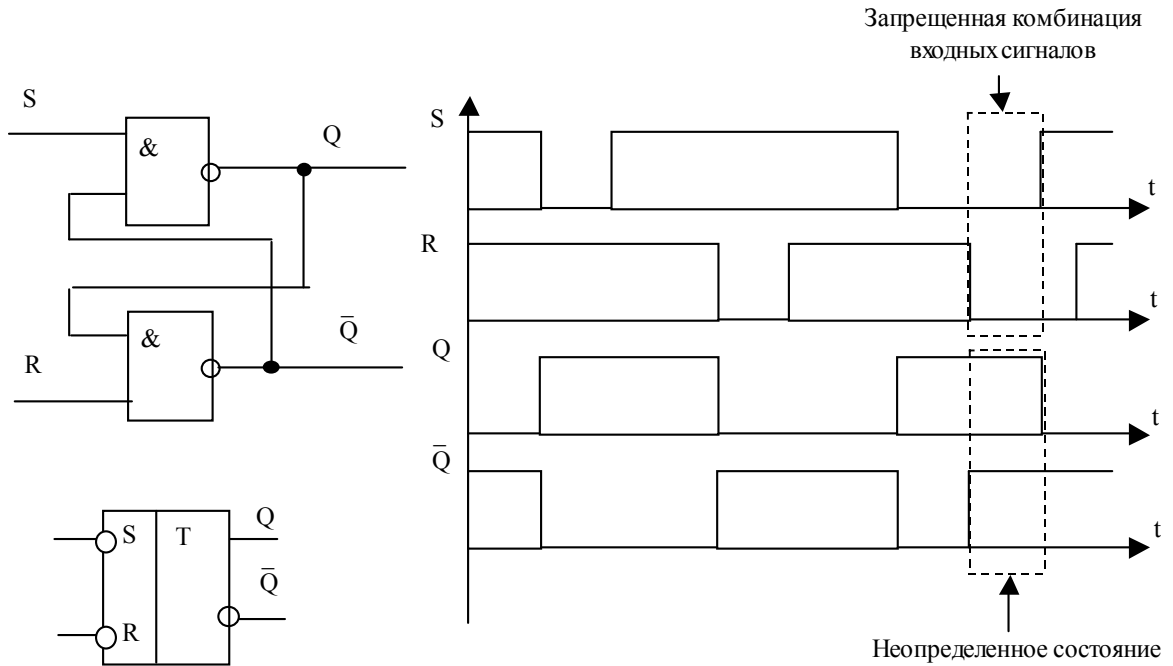


Рис. 5.1

Синхронные триггеры помимо информационных входов имеют тактирующий или синхронизирующий вход С, и переключение такого триггера в состояние, определяемое сигналами, поданными на информационные входы, происходит только при наличии соответствующего сигнала на входе синхронизации. Схема простейшего синхронного RS-триггера временная диаграмма его работы и УГО выглядят следующим образом (рис. 5.2).

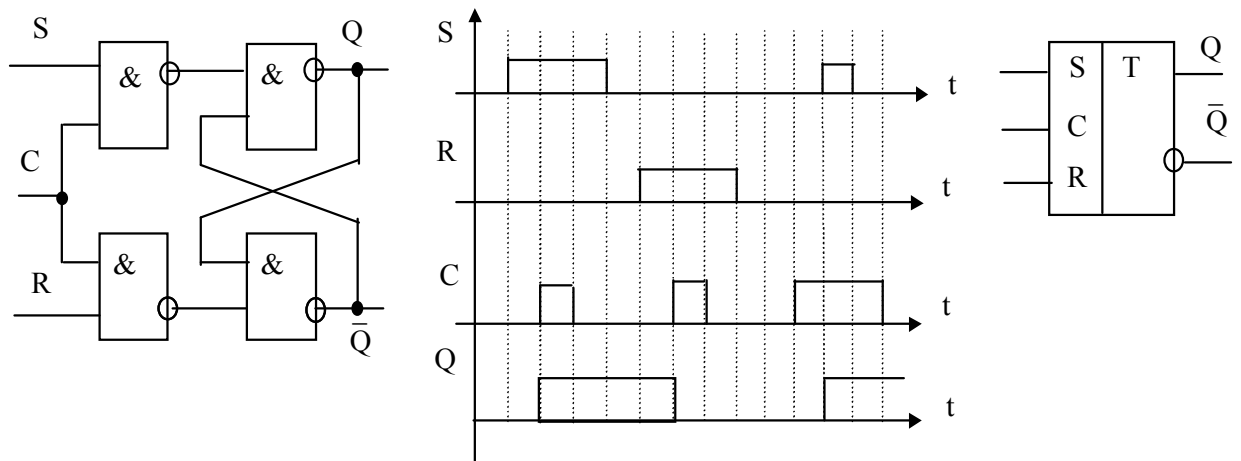


Рис. 5.2

Иногда желательным может быть наличие лишь одного информационного входа, в зависимости от значений сигнала на котором (0 или 1) триггер устанавливается в соответствующее состояние (0 или 1). Такой триггер называется D-триггером. Схема, временная диаграмма работы и УГО простейшего синхронного D-триггера выглядят следующим образом (рис. 5.3).

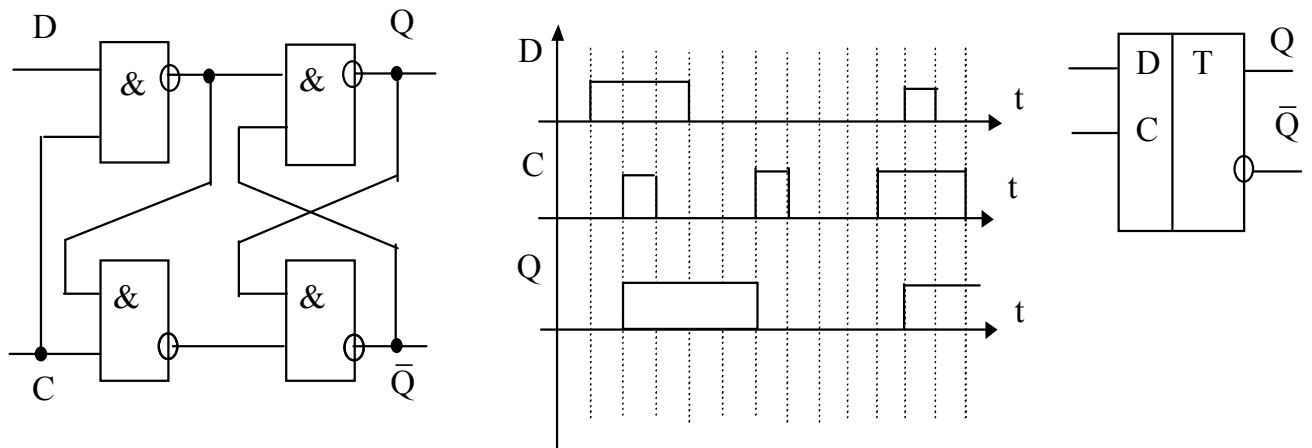


Рис. 5.3

Наличие у RS-триггера запрещенной комбинации входных сигналов и неопределенного состояния привело к появлению JK-триггера, у которого отсутствует этот недостаток. Схема простейшего синхронного двухступенчатого JK-триггера, временная диаграмма его работы и УГО приведены на рис. 5.4.

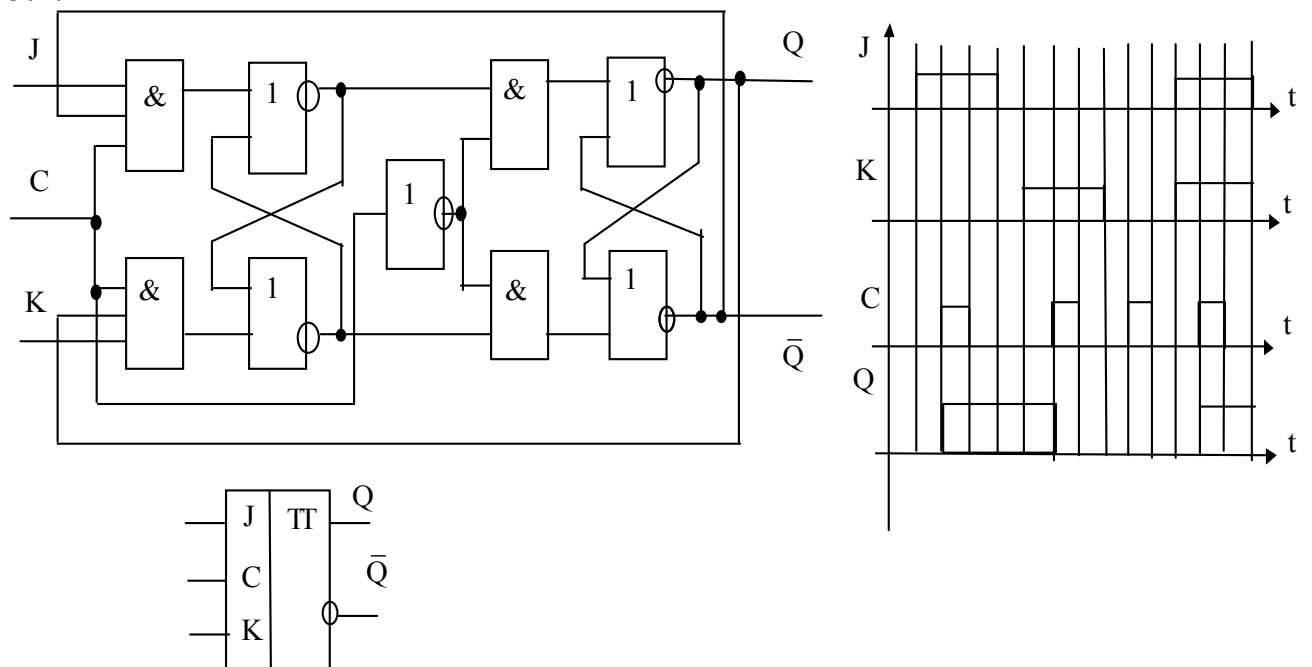


Рис. 5.4

В зависимости от того, по какому параметру сигнала на входе синхронизации происходит переключение триггера, они подразделяются на триггеры со статическим управлением и триггеры с динамическим управлением. В триггерах со статическим управлением управляющим параметром является уровень на входе C (логическая 1 – прямое статическое управление, логический 0 – инверсное статическое управление), а в триггерах с динамическим управлением – фронт сигнала на входе C (передний или задний). Если тригг-

гер переключается по переднему (перепад из 0 в 1) фронту, то он называется триггером с прямым динамическим управлением, а если по заднему (перепад из 1 в 0) – с инверсным динамическим управлением.

Все рассмотренные ранее синхронные триггеры являются триггерами с прямым статическим управлением. Их недостатком является то, что при активном уровне на входе С он превращается в асинхронный, поскольку переключается непосредственно по изменению сигналов на информационных входах. Путем некоторого усложнения их схем они могут быть преобразованы в триггеры с динамическим управлением, УГО которых приведены на рис. 5.5.

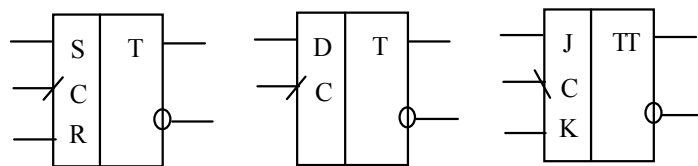


Рис. 5.5

Последним из четырех функционально различных типов триггеров является Т-триггер (или счетный триггер). Этот триггер имеет единственный вход Т и изменяет свое состояние на противоположное при подаче каждого сигнала на вход Т. В соответствии с таким определением функций Т-триггер может быть реализован на D- или JK-триггерах в соответствии со схемами, приведенными на рис. 5.6.

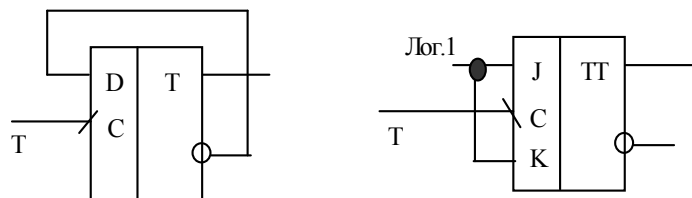


Рис. 5.6

Микросхема К155ТВ1 (рис. 5.7) представляет собой синхронный двухступенчатый JK-триггер со статическим управлением и асинхронными инверсными установочными входами R и S.

Триггер имеет по три входа J и K, связанных операцией И, т.е. J (или K)=1 только тогда когда на все три входа J (или K) одновременно поданы логические 1. Смена состояний триггера, определяемая сигналам на входах J и K, происходит в момент начала паузы между тактовыми импульсами на входе С, что объясняется двухступенчатой структурой триггера. По этой же причине смену состояний на входах J и K следует производить в паузах между тактовыми импульсами. По входам S и R выполняется асинхронная, т.е. независимая от входов С, J, K установка триггера в состояния 1 и 0 соответственно. В целом работа триггера К155ТВ1 иллюстрируется временной диаграммой, приведенной на рис. 5.8



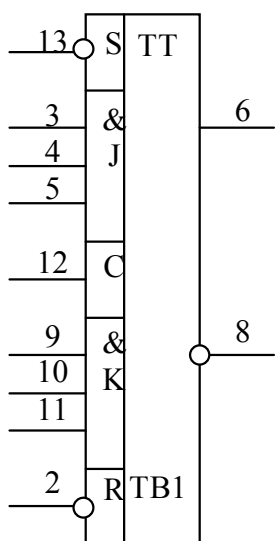


Рис. 5.7

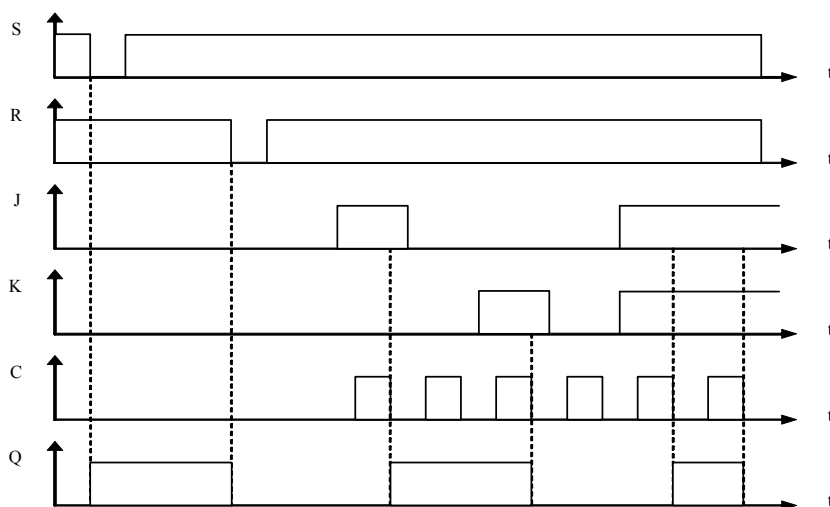


Рис. 5.8

Микросхема К155ТМ2 (рис. 5.9) содержит два независимых синхронных D-триггера с прямым динамическим управлением и асинхронными инверсными входами S и R. Работа триггера иллюстрируется временной диаграммой на рис. 5.10.

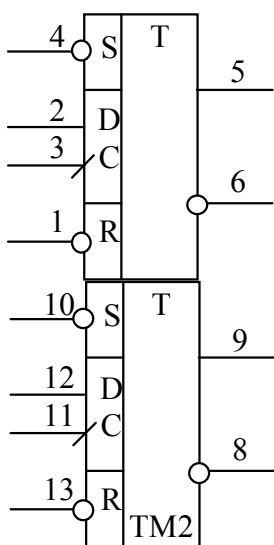


Рис. 5.9

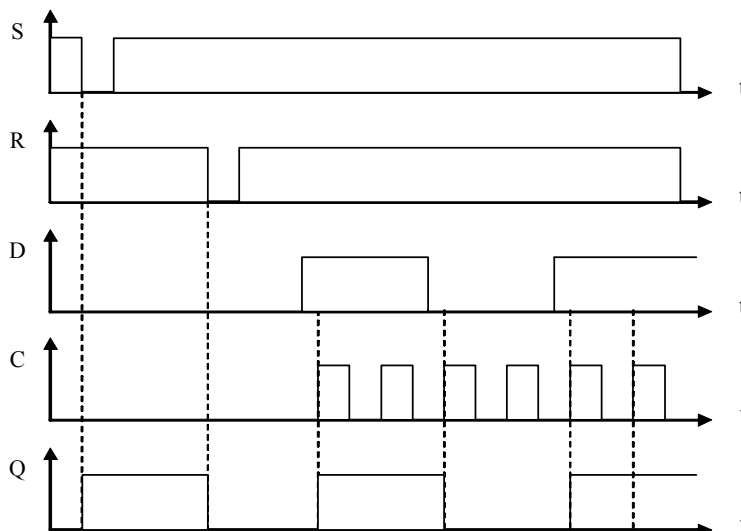


Рис. 5.10

Микросхема К155ТМ5 (рис. 5.11) представляет собой две пары синхронных D-триггеров с прямым статическим управлением, у которых входы синхронизации попарно объединены.

Микросхема К155ТМ7 (рис. 5.12) отличается от ТМ5 лишь наличием дополнительных инверсных выходов у каждого триггера.

Микросхема К155ТМ8 (рис. 5.13) содержит четыре синхронных D- триггера с прямым динамическим управлением по общему входу синхронизации С. Асинхронные инверсные входы R всех триггеров объединены.

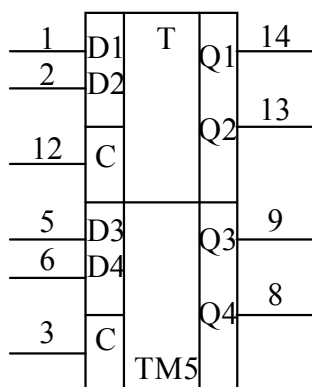


Рис. 5.11

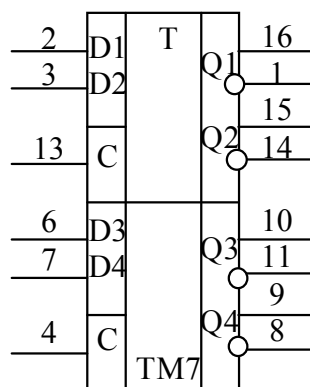


Рис. 5.12

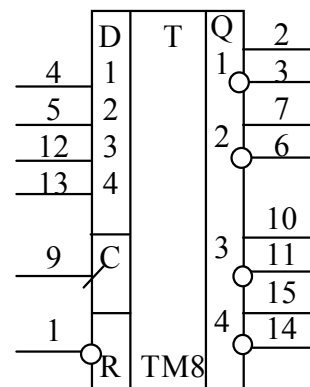


Рис. 5.13

**Регистром** называется упорядоченная совокупность триггеров, предназначенная для хранения многоразрядных двоичных слов или их частей. Элементами структуры регистра могут служить асинхронные и синхронные D -, RS - или JK-триггеры с динамическим или статическим управлением и вспомогательные логические элементы.

По способу ввода и вывода информации регистры подразделяются на параллельные, последовательные и универсальные.

Параллельным называется регистр с параллельной записью и параллельным считыванием всех разрядов записываемого или считываемого слова. Для его реализации необходимо столько триггеров, какова должна быть разрядность регистра. Информационный вход каждого триггера становится одним из информационных входов регистра, а для обеспечения параллельной, т.е. одновременной, записи каждого разряда слова в соответствующий ему триггер входы синхронизации всех триггеров следует объединить, сделав полученный вход входом синхронизации параллельной записи в регистр. Схема трехразрядного параллельного регистра, реализованного на D-триггерах с прямым динамическим управлением, и его УГО приведены на рис. 5.14.

Последовательным регистром или регистром сдвига называется регистр, который осуществляет запись и считывание многоразрядного двоичного слова, представленного в последовательном коде. Это означает, что слово вводится в регистр последовательно по одному разряду в каждом такте синхронизации, причем ввод осуществляется через единственный информационный вход регистра, в качестве которого выступает информационный вход первого триггера регистра. Отсюда следует, что в схеме регистра сдвига выход каждого триггера должен быть связан с информационным входом следующего

триггера для обеспечения перезаписи из разряда в разряд (сдвига), а синхронизирующие входы всех триггеров должны быть объединены для обеспечения одновременности перезаписи. В результате этого объединения формируется вход синхронизации сдвига регистра. Схема трехразрядного регистра сдвига, реализованного на D-триггерах с прямым динамическим управлением, и его УГО приведены на (рис. 5.15).

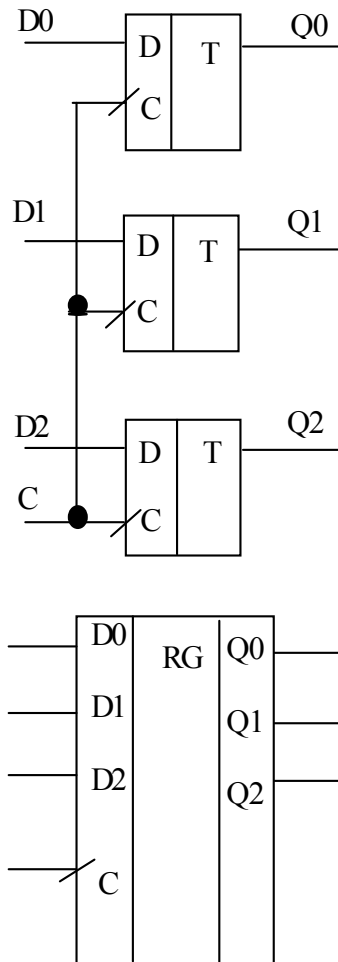


Рис. 5.14

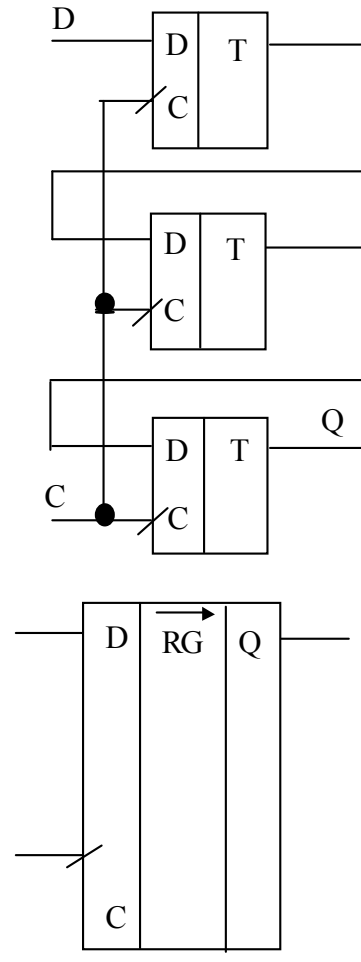


Рис. 5.15

Регистры сдвига, которые могут обеспечивать сдвиг в обоих направлениях, называются реверсивными. Для реализации реверсивного регистра информационный вход каждого триггера должен быть скоммутирован либо с выходом предыдущего триггера, либо с выходом последующего триггера. Управление коммутацией осуществляется с помощью специальных сигналов реверса. Из этих соображений следует, что на информационном входе каждого триггера реверсивного регистра должен быть установлен логический элемент, управляя которым с помощью сигналов реверса, можно было бы ком-

мутировать вход каждого триггера с выходами соседних слева и справа триггеров.

Регистры, сочетающие в себе свойства параллельных и последовательных, называются универсальными.

Микросхема К155ИР1 (рис. 5.16) представляет собой четырехразрядный универсальный регистр. Режим работы регистра задается уровнем на входе L. При L=1 регистр работает в параллельном режиме, записывая информацию с входов D, которая по заднему фронту импульса на входе С1 появляется на выходах Q. Состояния входов DR и С2 при этом могут быть произвольными. При L=0 регистр работает в последовательном режиме, записывая информацию с входа DR со сдвигом вправо (от Q0 к Q3) по заднему фронту импульса на входе С2, аналогично схеме, приведенной на рис. 4.2.1. Состояния входов D и С1 при этом могут быть произвольными, поскольку 0 на входе L блокирует прохождение сигналов с этих входов.

Микросхема К155ИР13 (рис. 5.17) представляет собой восьмиразрядный универсальный регистр. Он обеспечивает синхронное функционирование в следующих режимах: параллельный ввод, последовательный ввод со сдвигом вправо и последовательный ввод со сдвигом влево. Режим выбирается подачей соответствующих уровней на входы S0 и S1 (табл. 5.1)

Таблица 5.1

S0	S1	Режим
0	0	Хранение
0	1	Сдвиг влево
1	0	Сдвиг вправо
1	1	Параллельный ввод

В режиме параллельного ввода информация, представленная в параллельном коде на входах D, записывается в регистр по переднему фронту тактового импульса на входе С. Для последовательного ввода и сдвига в одну из сторон информация подается поразрядно на выбранный вход DR (сдвиг вправо) или DL (сдвиг влево) и по переднему фронту тактовых импульсов на входе С сдвигается в соответствующую сторону. Асинхронное обнуление регистра осуществляется подачей 0 на вход R.

Микросхема К155ИР15 (рис. 5.18) представляет собой четырехразрядный параллельный регистр с тремя состояниями выхода. Регистр обеспечивает синхронную запись параллельного кода с входов D при L1=L2=0 по переднему фронту тактового импульса на входе С. Для перевода выходов Q регистра в третье состояние достаточно на один из входов OE подать логическую 1. Асинхронное обнуление регистра осуществляется подачей единицы на вход R.

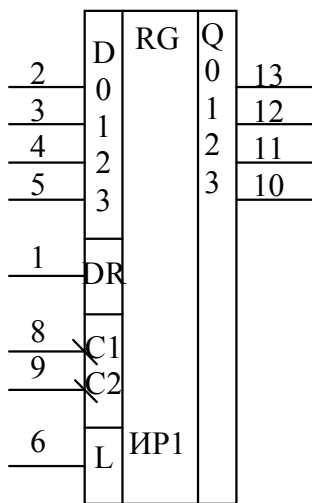


Рис. 5.16

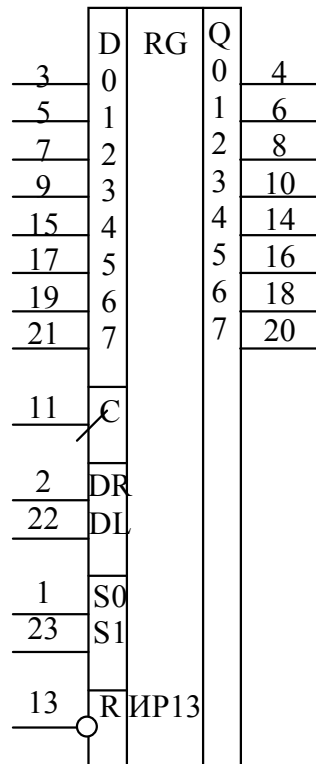


Рис. 5.17

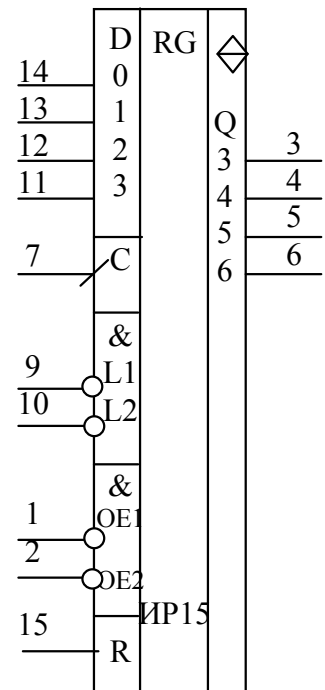


Рис. 5.18

### Контрольная работа №5

1. Осуществить классификацию предлагаемой в вашем варианте схемы триггера по следующим классификационным признакам:

<input type="checkbox"/>	RS-триггер
<input type="checkbox"/>	D-триггер
<input type="checkbox"/>	JK-триггер
<input type="checkbox"/>	T-триггер
<input type="checkbox"/>	с прямыми информационными входами (входом)
<input type="checkbox"/>	с инверсными информационными входами (входом)
<input type="checkbox"/>	асинхронный
<input type="checkbox"/>	синхронный
<input type="checkbox"/>	с прямыми установочными входами
<input type="checkbox"/>	с инверсными установочными входами
<input type="checkbox"/>	без установочных входов
<input type="checkbox"/>	со статическим управлением
<input type="checkbox"/>	двухступенчатый
<input type="checkbox"/>	с прямым динамическим управлением
<input type="checkbox"/>	с инверсным динамическим управлением

2. Осуществить классификацию предлагаемой в вашем варианте схемы регистра по следующим классификационным признакам:

<input type="checkbox"/>	с последовательным вводом
<input type="checkbox"/>	с последовательным выводом
<input type="checkbox"/>	реверсивный
<input type="checkbox"/>	с параллельным вводом
<input type="checkbox"/>	с параллельным выводом
<input type="checkbox"/>	универсальный
<input type="checkbox"/>	со статическим управлением
<input type="checkbox"/>	с прямым динамическим управлением
<input type="checkbox"/>	с инверсным динамическим управлением
<input type="checkbox"/>	с тремя состояниями выхода
<input type="checkbox"/>	с двунаправленной шиной ввода-вывода
<input type="checkbox"/>	количество разрядов

**Счетчиком** называется последовательностное устройство, представляющее собой организованную совокупность счетных или Т-триггеров, сигналы на выходах которых, взятые в совокупности, в определенном коде отображают число импульсов, поступивших на вход счетчика, называемый счетным. Счетчик, образованный цепочкой из  $m$  последовательно соединенных Т-триггеров, сможет подсчитать в двоичном коде  $2^m$  импульсов. Каждый из триггеров такой цепочки называют разрядом счетчика. Число  $m$  определяет количество разрядов двоичного числа, которое может быть записано в счетчике. Число  $K_{сч}=2^m$  называют коэффициентом счета или модулем счета.

Код, соответствующий числу подсчитанных на данный момент импульсов, снимается с выходов триггеров счетчика. В паузах между входными импульсами триггеры сохраняют свое состояние, т.е. счетчик запоминает число подсчитанных импульсов.

Некоторое состояние счетчика (часто - нулевое, но не обязательно) принимается за исходное. Остальные состояния нумеруются по числу поступивших входных импульсов. Когда число входных импульсов достигнет величины  $K_{сч}$ , происходит переполнение счетчика, характеризуемое возвратом счетчика в исходное состояние, после чего счетчик способен повторить цикл работы. Таким образом, коэффициент счета характеризует число входных импульсов, необходимое для выполнения одного цикла и возвращения в исходное состояние. В соответствии со сказанным, работу счетчика можно представить в виде графа (рис. 5.19), вершины которого отображают состояния счетчика, а дуги – переход из одного состояния в другое под действием очередного импульса на счетном входе.

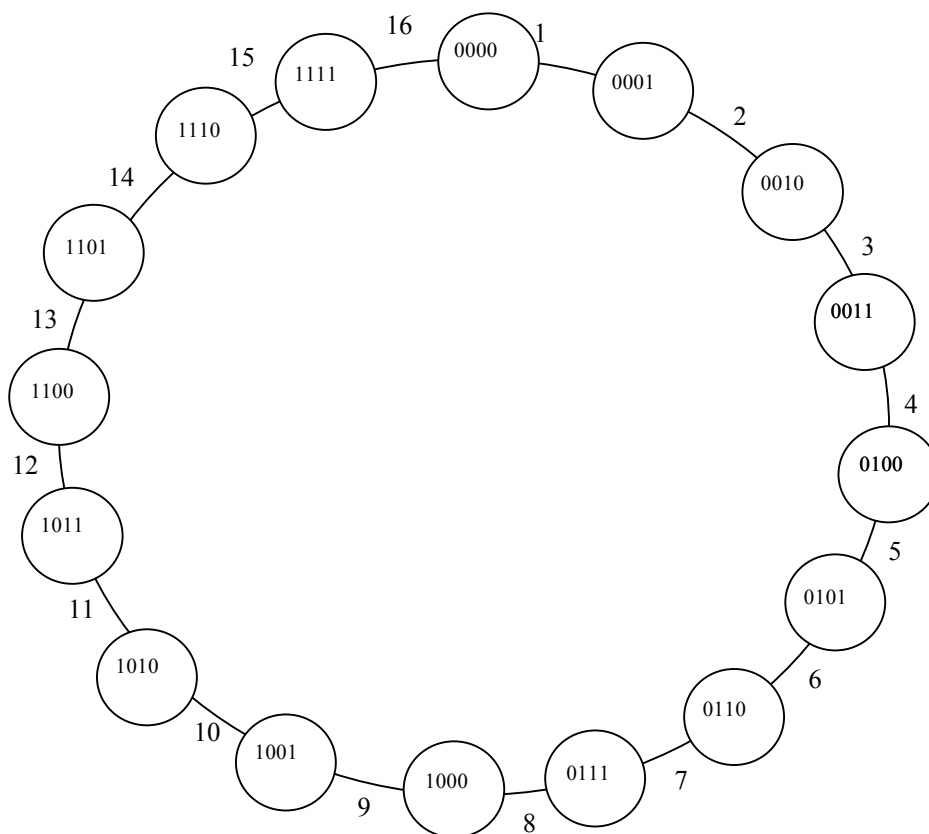


Рис. 5.19

Цифровые счетчики классифицируются следующим образом:

- по направлению счета - суммирующие, вычитающие, реверсивные;
- по коэффициенту счета - двоичные, недвоичные, с постоянным коэффициентом счета, с переменным коэффициентом счета;
- по способу организации внутренних связей - с последовательным переносом, с параллельным переносом, с комбинированным переносом.

Классификационные признаки независимы и могут встречаться в схемах реальных счетчиков в различных сочетаниях: например, суммирующие счетчики бывают как с последовательным, так и с параллельным переносом и могут иметь двоичный или любой другой коэффициент счета.

Двоичными счетчиками называются счетчики с  $K_{сч}=2^m$ . Пусть требуется построить двоичный суммирующий счетчик с  $K_{сч}=16$ . Для его реализации предложено использовать синхронные D-триггеры с прямым динамическим управлением. Для синтеза двоичного счетчика с заданным  $K_{сч}=16$  потребуется  $m = \log_2 K_{сч} = \log_2 16 = 4$  счетных триггера. Поскольку для синтеза предложены синхронные D-триггеры, то следует их сделать счетными путем соединения инверсного выхода каждого триггера с его же входом D. Обозначим полученные счетные триггеры  $Q_0, Q_1, Q_2, Q_3$ . Пусть  $Q_0$  будет младшим разрядом счетчика, а  $Q_3$  – старшим. Тогда таблица изменения состояний суммирующего счетчика будет выглядеть следующим образом (табл. 5.2).

Таблица 5.2

№ имп.	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	№ имп.	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
Исх. сост	0	0	0	0	9	1	0	0	1
1	0	0	0	1	10	1	0	1	0
2	0	0	1	0	11	1	0	1	1
3	0	0	1	1	12	1	1	0	0
4	0	1	0	0	13	1	1	0	1
5	0	1	0	1	14	1	1	1	0
6	0	1	1	0	15	1	1	1	1
7	0	1	1	1	16	0	0	0	0
8	1	0	0	0					

Из табл. 5.2 видно, что перенос из младшего разряда в следующий, выражающийся в изменении состояния этого следующего разряда, в суммирующем счетчике должен происходить при переключении младшего триггера из 1 в 0, а предложенные для синтеза триггеры с прямым динамическим управлением переключаются перепадом из 0 в 1, в связи с чем счетный вход каждого последующего триггера следует соединить с инверсным выходом предыдущего. С учетом сказанного схема суммирующего двоичного счетчика  $K_{сч}=16$  будет выглядеть так, как показано на рис. 5.20, где кроме схемы приведена временная диаграмма его работы и УГО.

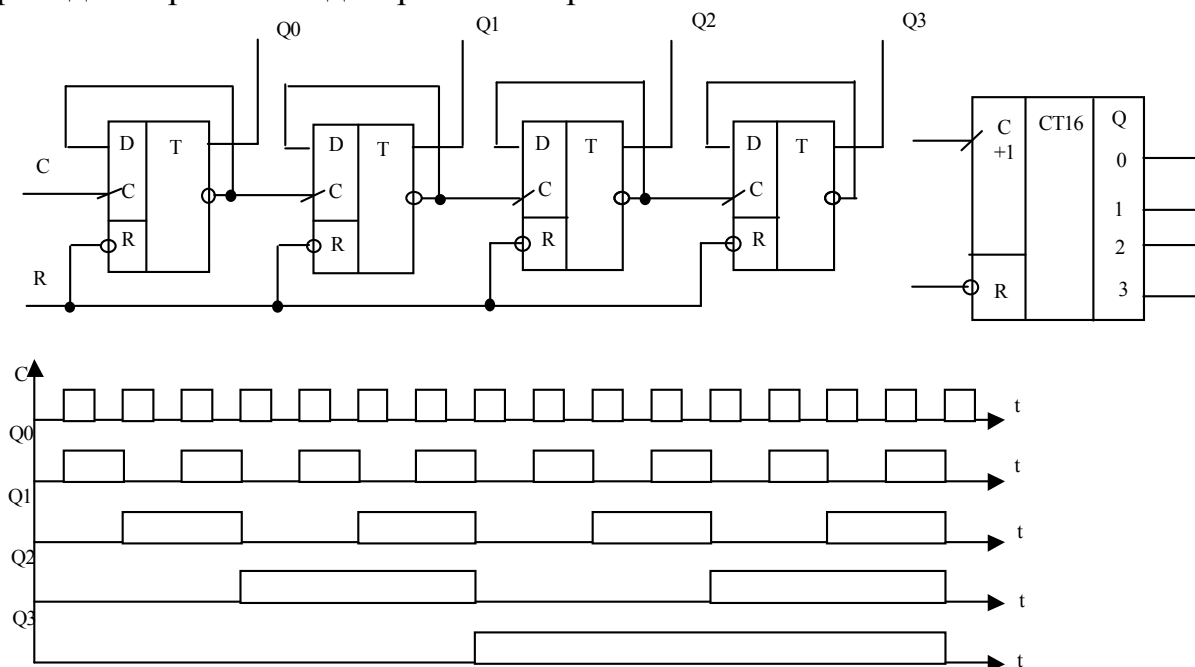


Рис. 5.20

В суммирующем счетчике каждый входной импульс увеличивает число, записанное в счетчике, на 1. Вычитающий счетчик действует обратным образом: двоичное число, хранящееся в счетчике, с каждым поступающим импульсом уменьшается на 1. Граф такого счетчика будет выглядеть аналогич-



но графу суммирующего счетчика за исключением того, что дуги будут иметь противоположное направление. Переполнение вычитающего счетчика происходит после достижения им нулевого состояния. Аналогично суммирующему счетчику в соответствии с графом может быть построена таблица изменения состояний вычитающего счетчика, из которой видно, что при использовании тех же триггеров для получения вычитающего счетчика необходимо соединить вход каждого последующего триггера с прямым выходом предыдущего. Для примера на рис. 5.21 приведен вычитающий счетчик на тех же триггерах с  $K_{сч}=16$ .

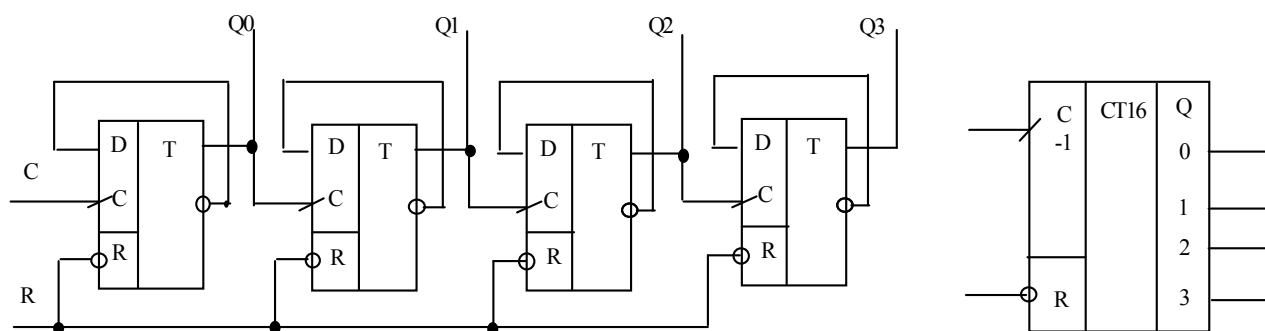


Рис. 5.21

Очевидно, что если для синтеза счетчика будут предложены триггеры с инверсным динамическим управлением, то для получения суммирующего счетчика необходимо со счетным входом последующего триггера соединять прямой выход предыдущего, а для получения вычитающего – инверсный выход.

Реверсивный счетчик может работать в качестве суммирующего и вычитающего. Такой счетчик имеет дополнительные управляющие входы для задания направления счета. С учетом рассмотренных ранее схем в качестве варианта построения реверсивного счетчика можно предложить схему, в которой между триггерами установлены логические элементы, которые под управлением сигнала реверса осуществляют коммутацию входа каждого последующего триггера либо с инверсным, либо с прямым выходом предыдущего триггера.

Введением дополнительных логических связей - обратных и прямых - двоичные счетчики могут быть обращены в недвоичные, для которых коэффициент пересчета не равен целой степени двойки. Для построения недвоичного счетчика с  $K_{счн}$  строится двоичный счетчик с ближайшим большим  $K_{сч}$ , после чего в него вводится схема, выявляющая состояние счетчика, в которое он переходит под действием импульса, номер которого равен требуемому  $K_{счн}$ . Сигналом с выхода этой схемы о выявлении этого состояния счетчик принудительно переводится в исходное состояние. В качестве примера на рис. 5.22 приведена схема суммирующего счетчика с  $K_{сч}=5$ .

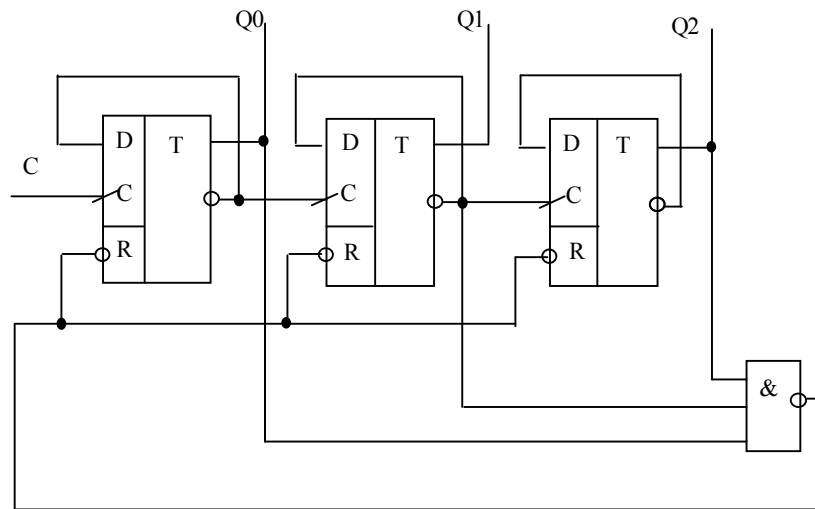


Рис. 5.22

Можно предложить еще один вариант реализации недвоичного счетчика, когда выявляется состояние, на 1 меньше требуемого состояния, сигналом о его выявлении счетчик принудительно переводится в состояние, предшествующее исходному, после чего следующим импульсом в естественном порядке переходит в исходное состояние.

Все рассмотренные до сих пор счетчики относятся к классу асинхронных или счетчиков с последовательным переносом. Их отличительной особенностью является то, что импульсы, подлежащие счету, поступают на вход первого триггера, а сигнал переноса передается последовательно от предыдущего триггера к последующему. Таким образом, каждый предыдущий триггер является источником счетных импульсов для последующего триггера. Поскольку каждый триггер обладает определенной задержкой переключения, то основным недостатком таких счетчиков - сравнительно низкое быстродействие, уменьшающееся с увеличением числа разрядов счетчика.

В синхронных счетчиках (или счетчиках с параллельным переносом) счетные импульсы подаются на тактовые входы всех триггеров счетчика одновременно, а каждый из триггеров служит по отношению к последующим триггерам лишь источником информационных сигналов о своем состоянии до поступления очередного счетного импульса. Срабатывание триггеров такого счетчика происходит синхронно, и задержка переключения всего счетчика в новое состояние равна задержке одного триггера. Для синтеза таких счетчиков используются синхронные JK- и D-триггеры с дополнительными логическими элементами на информационных входах для организации указанных информационных связей между триггерами. Схема синхронного суммирующего счетчика с  $K_{сч}=8$ , реализованного на синхронных JK-триггерах, приведена на рис. 5.23.

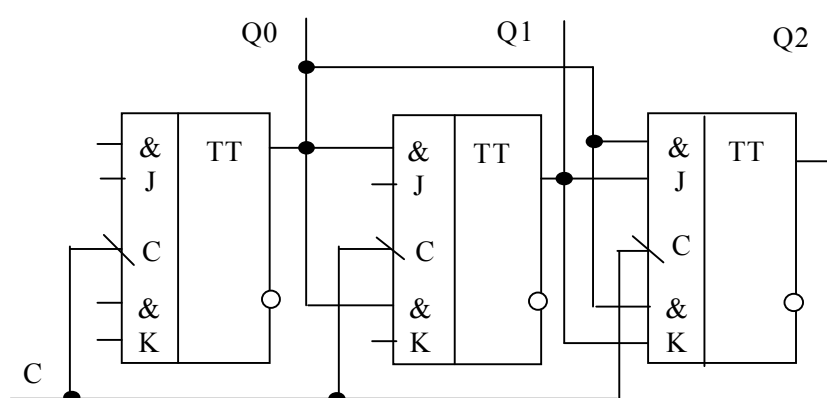


Рис. 5.23

Для построения вычитающего синхронного счетчика с теми же параметрами и на тех же триггерах все информационные связи необходимо осуществлять не с прямыми, а с инверсными выходами триггеров. Отсюда становится понятной и методика построения реверсивного синхронного счетчика.

Принцип построения недвоичных синхронных счетчиков так же, как и асинхронных, состоит в исключении некоторых состояний двоичного счетчика, которые являются избыточными для недвоичного счетчика. Однако исключение этих избыточных состояний осуществляется с помощью введения в схему счетчика информационных обратных связей, образованных дополнительными логическими цепями, соединяющими входы и выходы соответствующих триггеров. Таким образом, задача синтеза недвоичного синхронного счетчика сводится к определению необходимых обратных связей и минимизации их количества.

Количество триггеров в таком недвоичном счетчике, как и ранее, определяется из выражения  $m = \lceil \log_2 K_{сч} \rceil$ , где  $\lceil \quad \rceil$  – округление до ближайшего большего целого числа. Число исключаемых избыточных состояний равно  $N = 2^m - K_{сч}$ . Поскольку можно исключить любые состояния в любых сочетаниях, то общее число схем недвоичных счетчиков с одним и тем же  $K_{сч}$  и всеми вариантами изменения порядка счета определяется выражением

$$\frac{(K_{сч} - 1)! \times K_{сч}}{N! \times (K_{сч} - N)}$$

В общем случае выбор исключаемых состояний определяется назначением счетчика.

Продemonстрируем методику синтеза синхронного недвоичного счетчика на примере счетчика с  $K_{сч} = 3$ , для реализации которого используются синхронные JK-триггеры. Для построения счетчика требуется два триггера. Обозначим их Q1 и Q2. В качестве избыточного состояния выберем состояние счетчика Q1 = 1 и Q2 = 1. Составим таблицу изменения состояний счетчика (табл. 5.3).

Таблица 5.3

№ вх.имп.	Q2	Q1
Исх.сост	0	0
1	0	1
2	1	0
3	0	0

Задача синтеза счетчика будет состоять в отыскании логических функций

$$J1 = \varphi_1(Q1, Q2),$$

$$K1 = \varphi_2(Q1, Q2),$$

$$J2 = \varphi_3(Q1, Q2),$$

$$K2 = \varphi_4(Q1, Q2),$$

обеспечивающих реализацию табл. 5.3. Для решения этой задачи на основании таблицы переключений JK-триггера (табл. 5.4) составляется так называемая характеристическая таблица JK-триггера (табл. 5.5).

Таблица 5.4

J	K	Q(t+1)
0	0	Q(t)
1	0	1
0	1	0
1	1	$\bar{Q}(t)$

Таблица 5.5

Q(t)→Q(t+1)	J	K
0→0	0	*
0→1	1	*
1→0	*	1
1→1	*	0

На основании табл. 5.3 и табл. 5.5 составляются таблицы истинности искомых функций (табл. 5.6).

Таблица 5.6

Q2	Q1	J1	K1	J2	K2
0	0	1	*	0	*
0	1	*	1	1	*
1	0	0	*	*	1

Из этой таблицы могут быть получены искомые функции:

$$J1 = \bar{Q}2,$$

$$K1 = 1,$$

$$J2 = Q1,$$

$$K2 = 1.$$

Построенная в соответствии с этими уравнениями схема синхронного не двоичного счетчика с  $K_{сч}=3$  приведена на рис. 5.24.

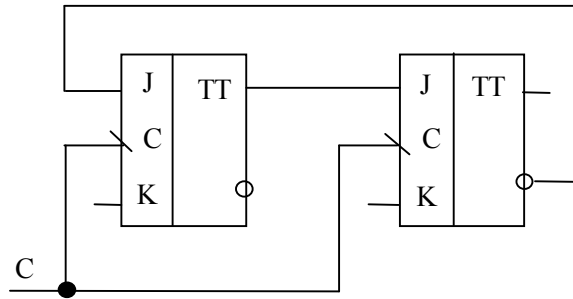


Рис. 5.24

По этой методике может быть построен как любой не двоичный синхронный счетчик, так и счетчик с произвольным порядком счета, т.е. с требуемым порядком изменения состояний счетчика, задаваемым исходной для синтеза таблицей, подобной табл. 5.3.

Рассмотренная структура синхронного счетчика с  $K_{СЧ}=3$  обладает еще одним интересным свойством. Если между триггерами этой структуры вставить счетчик с коэффициентом счета, равным  $K$ , то общий коэффициент счета полученного счетчика будет равен  $K_{СЧ}=2K+1$ .

С увеличением числа разрядов синхронных счетчиков быстро растет число внутренних логических связей и дополнительных логических элементов. Компромиссным решением является счетчик с комбинированным переносом, представляющий собой совокупность групп, причем каждая группа представляет собой счетчик с параллельным переносом, а перенос между группами осуществляется последовательно.

Счетчики, оформленные как самостоятельные изделия, имеются в составе многих серии ИС. Номенклатуру счетчиков отличает большое разнообразие. Многие из них обладают универсальными свойствами и позволяют управлять коэффициентом и направлением счета, вводить до начала счета исходное число (предустановка), прекращать по команде счет, наращивать число разрядов и т.п. С помощью таких счетчиков можно решить большинство задач, возникающих при разработке цифровой аппаратуры.

Суммирующие счетчики с последовательным переносом типа К155ИЕ2, К155ИЕ4, К155ИЕ5 близки по логической структуре и принципу действия. Они состоят из четырех одинаковых JK-триггеров, используемых либо как JK-триггеры, либо как T-триггеры. Внутри микросхемы триггеры соединены таким образом, что образуют две секции. Три триггера соединены в последовательную цепочку, четвертый – выполнен самостоятельным. Такая структура позволяет использовать секции отдельно (независимо), а также по-разному соединять их между собой. Помимо триггеров в микросхему входят логические элементы, с помощью которых осуществляется одновременная установка всех триггеров в определенное состояние.

В микросхеме К155ИЕ5 (рис. 5.25) цепочка из трех триггеров образует счетчик-делитель частоты с  $K_{СЧ} = 8$ . При внешнем соединении выхода  $Q_0$  с

входом С2 получается счетчик-делитель частоты с  $K_{Сч} = 16$ , функционирующий в соответствии с табл. 5.7. Переход счетчика из одного состояния в другое происходит по заднему фронту импульса на входе С1. Только наличие одновременно двух единиц на входах R1 и R2 обеспечивает установку всех триггеров в нулевое состояние и прекращение счета.

Таблица 5.7

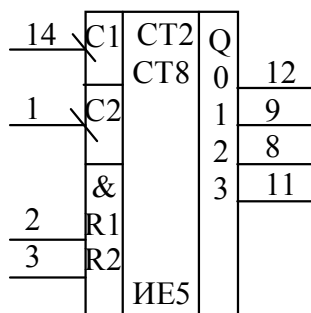


Рис. 5.25

№ вх.имп.	Выходы				№ вх.имп.	Выходы			
	Q3	Q2	Q1	Q0		Q3	Q2	Q1	Q0
Исх.сост.	0	0	0	0	8	1	0	0	0
1	0	0	0	1	9	1	0	0	1
2	0	0	1	0	10	1	0	1	0
3	0	0	1	1	11	1	0	1	1
4	0	1	0	0	12	1	1	0	0
5	0	1	0	1	13	1	1	0	1
6	0	1	1	0	14	1	1	1	0
7	0	1	1	1	15	1	1	1	1

Микросхема K155IE2 (рис. 5.26) представляет собой двоично-десятичный счетчик. Секция из трех триггеров с входом С2 и выходами Q1, Q2, Q3 представляет собой счетчик-делитель частоты с  $K_{Сч} = 5$ . При последовательном соединении обеих секций образуется счетчик с  $K_{Сч} = 10$ . Еще одно отличие K155IE2 состоит в наличии входов S1 и S2, при одновременной подаче единиц на которые счетчик устанавливается в состояние  $Q3=Q0=1$ ,  $Q1 = Q2 = 0$ . Двоично-десятичный код, в котором работает счетчик, зависит от способа соединения секций. Если автономный триггер используется в качестве младшего разряда счетчика (т.е. С1- вход счетчика, Q0 соединено с С2), то счетчик считает в коде 8421. Если этот триггер используется в качестве старшего разряда (т.е. С2 – вход счетчика, Q3 соединено с С1), то счетчик работает в коде 5421. Изменение состояний счетчика по каждому счетному импульсу при обоих включениях приведено в табл. 5.8.

Таблица 5.8

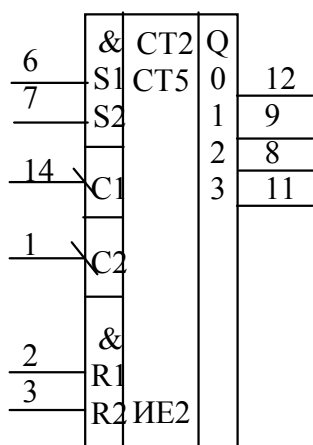


Рис. 5.26

№ вх.имп.	Выходы							
	Q3	Q2	Q1	Q0	Q0	Q3	Q2	Q1
Исх.сост.	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	0
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

Микросхема К155ИЕ4 (рис. 5.27) отличается тем, что два из трех триггеров второй секции охвачены цепью обратной связи, за счет чего коэффициент счета этой пары равен трем (см. схему на рис. 5.25), а третий триггер работает в счетном режиме. В результате общий коэффициент счета второй секции равен шести. При последовательном соединении секций образуется счетчик с  $K_{сч} = 12$ . Последовательность комбинаций, формируемых на выходах счетчика, как и в предыдущем случае, зависит от способа включения секции, образованной автономным триггером. Сказанное иллюстрируется табл. 5.9.

Таблица 5.9

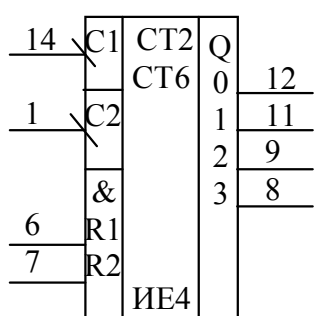


Рис. 5.27

№ вх.имп.	Выходы							
	Q3	Q2	Q1	Q0	Q0	Q3	Q2	Q1
Исх.сост.	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	1	0	0
4	0	1	0	0	0	1	0	1
5	0	1	0	1	0	1	1	0
6	1	0	0	0	1	0	0	0
7	1	0	0	1	1	0	0	1
8	1	0	1	0	1	0	1	0
9	1	0	1	1	1	1	0	0
10	1	1	0	0	1	1	0	1
11	1	1	0	1	1	1	1	0

Микросхемы К155ИЕ6 (рис. 5.28) и К155ИЕ7 (рис. 5.29) – четырехрядные реверсивные синхронные счетчики.

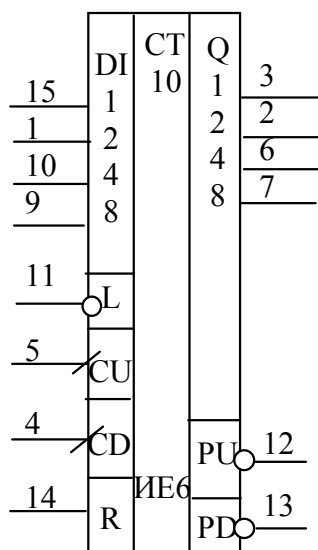


Рис. 5.28

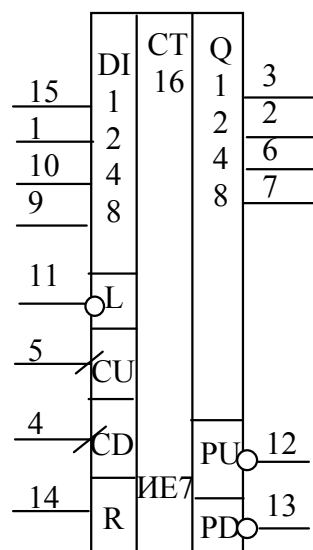


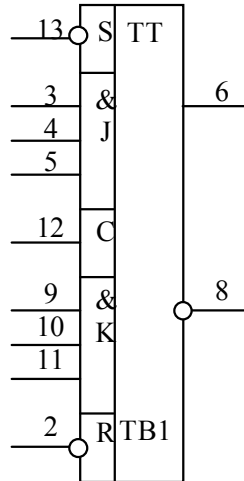
Рис. 5.29

Различие между ними состоит в величине коэффициента счета: для К155ИЕ6 он равен  $K_{\text{сч}} = 10$ , а для К155ИЕ7 он равен  $K_{\text{сч}} = 16$ . Входы СD и СU – счетные. Последовательность входных импульсов подается на один из этих входов в зависимости от требуемого режима счета (СU – суммирование, СD – вычитание). Входы D11, D12, D14, D18 предназначены для параллельного ввода в счетчик исходного числа (предустановка), от которого требуется начать счет. Ввод этого числа осуществляется подачей логической 0 на вход L. Подачей на вход R логической 1 выполняется установка в нуль всех триггеров счетчика. Этот вход имеет приоритет над всеми остальными. Помимо выходов триггеров Q1, Q2, Q4, Q8 счетчики имеют по два инверсных выхода, называемых выходами переноса (PU) и заема (PD). В режиме суммирования сигнал переноса возникает при переходе из состояния 1111 (для ИЕ7) либо 1001 (для ИЕ6) в состояние 0000 и имеет длительность паузы между соответствующими счетными импульсами. В режиме вычитания сигнал заема возникает при переходе из состояния 0000 в состояние 1111 (для ИЕ7) либо 1001 (для ИЕ6) и имеет те же параметры, что и сигнал переноса. Эти сигналы используются при каскадировании микросхем для наращивания разрядности путем соединения выходов переноса и заема младшего счетчика с входами суммирования и вычитания соответственно старшего счетчика. Кроме того, сигналы с этих выходов могут использоваться для циклической предустановки счетчика в состояние, определяемое уровнями на входах D11, D12, D14, D18. Для этого достаточно соединить вход L с выходом переноса или заема в зависимости от режима работы. Если счетчики используются в качестве делителей частоты, то при соединении входа L с выходом переноса коэффициент деления будет равен 15-М (для ИЕ7) или 9-М (для ИЕ6), где М – десятичный эквивалент двоичного числа, поданного на входы D11, D12, D14, D18. При работе в одном из режимов (суммирования или вычитания) на счетном входе другого режима должен поддерживаться уровень логической 1.

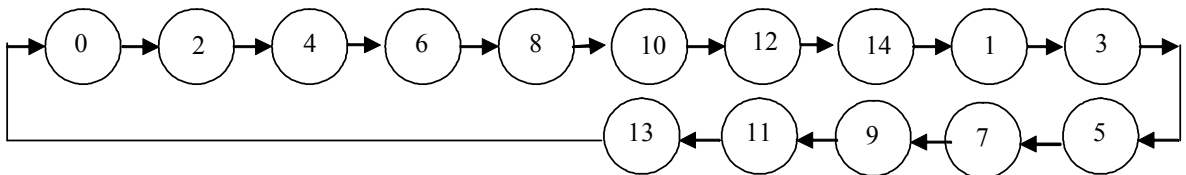


### Контрольная работа №6

Синтезировать на основе необходимого количества JK–триггеров с логикой 3И на входах J и K



схему синхронного счетчика с произвольным порядком счета, который определяется графом в соответствии с выданным вариантом, например таким, который приведен ниже.



**БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. Ланских В.Г. Основы цифровой схемотехники: Сборник задач для практических занятий. – Киров: Изд-во ВятГУ, 2009. – 45 с.
2. Ланских В.Г. Микросхемотехника: Лабораторный практикум. – Киров: Изд-во ВятГУ, 2009. – 25 с.