Кировское областное государственное профессиональное
образовательное бюджетное учреждение
профессионального образования
«Кировский авиационный техникум»
КОГПОБУ «Кировский авиационный техникум»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ВЫПОЛНЕНИЯ КОНТРОЛЬНОЙ РАБОТЫ

По дисциплине:

Электронная преобразовательная техника

для специальности 13.02.10
4 курса з/о

2019

РАССМОТРЕНА
Предметной цикловой комиссией электротехнических дисциплин
Протокол №_____ от
«____»_____2019 г.

Составлена на основе ФГОС СПО по специальности 13.02.10 «Электрические машины», рабочей программы по «Электронная преобразовательная техника» от 2019 г

Составитель: Ланских С.П. преподаватель выс. категории КОГПОБУ «Кировского авиационного техникума»

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Настоящие методические указания предназначены для облегчения самостоятельного изучения первого и второго разделов дисциплины «Вычислительная техника» студентами заочной формы обучения. Методические указания состоят из трех частей и приложений.

В первой и второй частях приведены краткие сведения из теории, достаточные для самостоятельного изучения.

В третьей части методических указаний приведены варианты для выполнения заданий контрольной работы. Каждое задание содержит восемь задач, для решения которых студент должен освоить законы и правила алгебры логики и методы синтеза типовых функциональных узлов ЭВМ. Исходные данные заданий приведены в приложениях.

В результате изучения первого и второго разделов дисциплины **студент должен:**

знать:

- элементную базу;
- системы счисления;
- правила и законы алгебры логики;
- типы и назначении типовых средств вычислительной техники;

уметь:

- выполняет синтез простейших схем управления электрическим и электромеханическим оборудованием;
- применять полученные знания для практического использования;

Завершает методические указания список литературы, необходимой для изучения всего курса.

1. АРИФМЕТИЧЕСКИЕ И ЛОГИЧЕСКИЕ ОСНОВЫ ЦИФРОВОЙ ТЕХНИКЕ

1.1. Позиционные системы счисления

Системой счисления называется способ представления любых чисел с помощью ограниченного алфавита символов, называемых цифрами.

Количество цифр, входящих в алфавит системы, принято называть основанием системы. Название системы происходит от ее основания.

Позиционной называется такая система счисления, в которой одна и та же цифра в последовательности цифр, изображающей число, имеет разное значение - вес в зависимости от ее позиции в этой последовательности. Вес цифры в записи числа не изображается, а подразумевается.

$(3243)_{10}$

Таким образом, в полной записи число представляется в виде полинома, каждый член которого представляет собой произведение коэффициента, в качестве которого могут использоваться только цифры алфавита системы, на соответствующую степень основания системы.

$3*10^3 + 2*10^2 + 4*10^1 + 3*10^0$

Обобщая с десятичной системы счисления на любую **а**-ичную позиционную систему счисления, любое число в этой системе может быть записано как

$$x = b_r a^r + b_{r-1} a^{r-1} + ... + b_0 a^0 + b_{-1} a^{-1} + ... + b_{-m} a^{-m}$$

Для краткости записи это число представляется в виде последовательности цифр - коэффициентов полинома

$(b_r\ b_{r-1} ...\ b_0, b_{-1} ...b_{-m})_a$

Позиции цифр в этой последовательности называются разрядами. В любой позиционной системе вес каждого разряда отличается от веса соседнего разряда в число раз, равное основанию системы.

Для внутреннего представления информации в ЭВМ используется двоичная система счисления.

Из названия системы следует, что она имеет основание, равное двум и в ее алфавит входит две цифры : 0 и 1.

Согласно общему правилу представления чисел в позиционных системах счисления, любое число в двоичной системе счисления представляется последовательностью нулей и единиц.

$(101001,01)_2 = (41,25)_{10}$

Согласно общему правилу такая запись является сокращенной записью полиномиального представления числа

$1*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0 + 0*2^{-1} + 1*2^{-2}$

Приведенный пример демонстрирует методику преобразования числа из двоичной системы счисления в десятичную.

В задаче обратного преобразования основной трудностью является нахождение максимальной степени основания (2) , которая еще содержится в преобразуемом десятичном числе. После ее нахождения определяется вхождение в преобразуемое число всех степеней меньших максимальной.

$(57)_{10} = 0*2^6 + 1*2^5 + 1*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0 = (0111001)_2$

При этом способе первым определяется значение старшего разряда искомого двоичного числа.

Можно использовать и другой способ, при котором искомое число определяется, начиная с младшего разряда. Суть метода состоит в последовательном делении преобразуемого десятичного числа на основание системы т.е. на 2 с выделением остатка.

57/2	28	**1**	**младший разряд**
28/2	14	0	
14/2	7	0	
7/2	3	1	
3/2	1	1	
1/2	0	**1**	**старший разряд**

Описанными методами осуществляется преобразование целого десятичного числа в двоичное.

Для преобразования десятичной дроби используется следующий алгоритм:

Преобразуемая десятичная дробь умножается на основание системы т.е. на 2.

Если результат умножения меньше 1,то старшему значащему разряду искомой двоичной дроби присваивается значение 0, если больше - значение 1. В последнем случае целая часть произведения отбрасывается, а дробная используется для получения следующего разряда двоичной дроби.

Описанная процедура повторяется до тех пор, пока:

а) в результате очередного умножения не получится произведение, равное точно 1.Тогда очередному разряду двоичной дроби присваивается значение 1 и процесс заканчивается (пример):

0.375 * 2	0.75	**0**	**старший разряд**
0.75* 2	1.5	1	
0.5* 2	1	**1**	**младший разряд**

т.е. $(0.375)_{10} = (0.011)_2$

б) не будет достигнута возможная точность преобразования, определя-

емая числом двоичных разрядов, отведенных для представления двоичной дроби (пример):

0.3*2	0.6	**0**	**старший разряд**
0.6*2	1.2	1	
0.2*2	0.4	0	
0.4*2	0.8	0	
0.8*2	1.6	1	
0.6*2	1.2	1	
0.2*2	0.4	0	
0.4*2	0.8	**0**	**младший разряд,**

т.е. $(0.3)_{10} = (0.01001100)_2 = (0.296875)_{10}$.

Иногда **при вводе и выводе** или просто при изображении двоичного числа на бумаге для более компактной его записи **используются восьмеричная и шестнадцатеричная системы счисления.**

Алфавит восьмеричной системы счисления содержит восемь символов: 0,1,2,3,4,5,6,7.

Алфавит шестнадцатеричной системы счисления содержит 16 символов: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.

Обе системы являются позиционными и им присущи те же закономерности, что и двоичной системе. Алгоритмы взаимных преобразований с десятичной системой принципиально не отличаются от описанных для двоичной системы. Используя их необходимо только помнить об изменении основания системы.

Взаимные преобразования между этими системами и двоичной системой более просты, поскольку основания этих систем сами представляют собой целые степени двойки.

$(142)_8 = (001100010)_2$, $(10101001101)_2 = (2515)_8$

$(A13E)_{16} = (1010000100111110)_2$, $(11001101110101110)_2 = (19BAE)_{16}$.

1.2. Кодирование отрицательных чисел

Правила арифметики во всех позиционных системах аналогичны.

В соответствии с этим очевидны правила сложения одноразрядных двоичных чисел (пример).

0 + 0 = 0, 1 + 0 = 1, 0 + 1 = 1, 1 + 1 = 0 и 1 переноса в следующий разряд.

Многоразрядные двоичные числа складываются с учетом этих очевидных правил. (пример).

```
 101010100011
+001101101101
-------------
 111000010000
```

Очевидны и правила вычитания одноразрядных двоичных чисел:

0 - 0 = 0, 1 - 0 = 1, 1 - 1 = 0, 0 - 1 = 1 и 1 заема из старшего разряда.

По этим же правилам с организацией заемов могут вычитаться и Многоразрядные двоичные числа (пример).

```
 10110011
-01001010
---------
 01101001
```

Организация заемов является операцией технически более сложной, чем организация переносов. В связи с этим желательно заменить операцию вычитания операцией сложения, а это возможно лишь при введении представления для отрицательных чисел.

Для этого необходимо, во-первых, ввести обозначения для знака числа. Принято обозначать + нулем, а - единицей. Для размещения символа знака обычно отводится старший, т.е. самый левый из отведенных для представления числа разряд. (пример).

$(+18)_{10} = (0\ 0010010)_{2\ пр.}$

$(-18)_{10} = (1\ 0010010)_{2\ пр.}$

Такое представление чисел со знаком называется прямым кодом.

Поскольку прямой код ничем, кроме введения знака, не отличается от представления двоичного числа без знака, он не позволяет достичь желаемого результата - замены вычитания сложением. Прямой код практически не используется для представления чисел в ЭВМ.

Для достижения желаемого результата необходимо ввести особое представление и для модуля отрицательного числа. В качестве такого особого представления используются обратный и дополнительный коды.

Прежде всего, необходимо отметить, что положительные числа во всех трех названных кодах (прямом, обратном, дополнительном) изображаются одинаково.

Обратный код отрицательного числа получается путем инвертирования всех разрядов, включая знаковый, прямого кода положительного числа, имеющего тот же модуль.

Пример1: $(+14)_{10} + (-3)_{10} = (+11)_{10}$, число разрядов n=5

$(+14)_{10}$=- (0 1 1 1 0)$_{2\ обр.}$
$(-3)_{10}$= (1 1 1 0 0)$_{2\ обр.}$
1 0 1 0 1 0
1 Круговой перенос
$(+11)_{10}$= (0 1 0 1 1)$_{2\ обр.}$

Пример2: $(+9)_{10}+(-14)_{10}=(-5)$

$(-14)_{10}$= (1 0 0 0 1)$_{2\ обр.}$
$(+9)_{10}$= (0 1 0 0 1)$_{2\ обр.}$
$(-5)_{10}$= (1 1 0 1 0)$_{2\ обр.}$

Из анализа примеров можно сделать выводы о недостатках обратного кода:

Единица переноса из знакового разряда при выполнении операций в обратном коде должна быть арифметически подсуммирована к младшему разряду результата, т.е. должен быть выполнен т.н. круговой перенос - лишняя операция.

Неоднозначность представления нуля.

От этих недостатков свободен дополнительный код.

Дополнительный код отрицательного числа получается из обратного кода этого же числа путем подсуммирования единицы к младшему разряду.

Пример1: $(+14)_{10} + (-3)_{10} =(+11)_{10}$, число разрядов n=5

$(+14)_{10}$=- (0 1 1 1 0)$_{2\ доп.}$
$(-3)_{10}$= (1 1 1 0 1)$_{2\ доп.}$
1 0 1 0 1 1
$(+11)_{10}$= (0 1 0 1 1)$_{2\ доп.}$

Пример2: $(+9)_{10}+(-14)_{10}=(-5)$

$(-14)_{10}$= (1 0 0 1 0)$_{2\ доп.}$
$(+9)_{10}$= (0 1 0 0 1)$_{2\ доп.}$
$(-5)_{10}$= (1 1 0 1 1)$_{2\ доп.}$

1.3. Двоично-десятичные коды

В некоторых цифровых устройствах используются двоично-десятичные коды.

При их использовании каждая десятичная цифра представляется четырмя двоичными разрядами, т.е. тетрадой или декадой.

Многоразрядные десятичные числа в двоично-десятичном коде образуются путем конкатенации (пристыковки) тетрад, кодирующих отдельные

цифры десятичного числа.

Существует большое количество различных двоично-десятичных кодов. (с избытком 3, 2421, 5421, 8421 и другие).

Наибольшее распространение получил код 8421.

Десятичная цифра	Алфавит кода 8421
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Его название отражает веса разрядов тетрады. Этот **код аддитивен**, т.е. сумма представлений двух цифр есть код их суммы. Недостатком кода является **отсутствие у него свойства самодополняемости**, т.е. из двоичного представления десятичной цифры путем простой инверсии разрядов тетрады нельзя получить представление цифры, дополняющей первую до девяти. Это свойство важно при операциях над двоично-десятичным числами со знаком.

Над числами, представленными в коде 8421, арифметические операции выполняются точно также, как и над многоразрядными двоичными числами, но с учетом того, что это все-таки не двоичные числа, а особое представление десятичных чисел. Учет этих особенностей выражается в необходимости коррекции результата

Пример1:

$(27)_{10}$= (0010 0111)$_{8421}$
+
$(36)_{10}$= (0011 0110)$_{8421}$

(0101 1101)$_{8421}$
+
0110 Коррекция

$(63)_{10}$= (0110 0011)$_{8421}$

Пример2:

$(28)_{10}$= (0010 1000)$_{8421}$
+
$(59)_{10}$= (0101 1001)$_{8421}$

(1000 0001)$_{8421}$
+
0110 Коррекция

$(87)_{10}$= (1000 0111)$_{8421}$

Таким образом коррекция осуществляется под суммированием корректирующей тетрады $(6)_{10} = (0110)_2$ к тетраде : а) которой получился код, не входящий в алфавит кода 8421, или б)из которой произошел перенос. Следует заметить, что перенос из тетрады в тетраду, возникающий при коррекции, не является основанием для повторной коррекции.

Для выполнения операций над числами со знаками, представленными в коде 8421, используют обратный и дополнительный коды 8421. Поскольку код 8421 не является самодополняющимся, то для получения обратного кода необходимо произвести обращение к десятичной системе , после чего полученное обращение представить в коде 8421 и выполнить требуемую опера-

цию.

1.4. Логические основы цифровой техники

Проектирование цифровых устройств и выбор наиболее оптимального варианта их построения производится с помощью алгебры логики. Алгебра логики (Булева алгебра) оперирует с такими понятиями как суждение, утверждение и высказывание. При этом используются двоичные переменные, удовлетворяющие следующим условиям:

- *переменная X=1, если она не равна нулю (X≠0);*
- *переменная X=0, если она не равна единице (X≠1);*

Выделяют три основных логических операции: И, ИЛИ, НЕ.

Любая из операций может быть задана тремя способами: словесно, с помощью таблицы истинности и с помощью формулы (аналитически).

Операция **НЕ - отрицание** (инверсия)

X	Y
0	1
1	0

$Y = \overline{X}$

Функция истинна, когда переменная ложна

Операция **ИЛИ – логическое сложение** (дизъюнкция)

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	1

$Y=X1+X2$

Функция *Y* истинна, если хотя бы одна переменная истинна

Операция **И – логическое умножение** (конъюнкция)

X1	X2	Y
0	0	0
0	1	0
1	0	0
1	1	1

$Y=X1 \bullet X2$

Функция *Y* истинна, если обе переменные истинны

Используя основные логические операции, можно реализовать любую логическую функцию (любую формулу). В табл.1.1 приведены наиболее распространенные логические функции. Для преобразования формул в алгебре логики существуют следующие правила и законы:

$$x \cdot 0 = 0 \qquad x + 0 = x$$

$$x \cdot 1 = x \qquad x + 1 = 1$$

$$x \cdot x \cdot x \cdot \ldots \cdot x = x \qquad x + x + x + \ldots + x = x$$

$$x \cdot \overline{x} = 0 \qquad x + \overline{x} = 1$$

$$\overline{\overline{x}} = x$$

Переместительный закон:

$$x + y = y + x$$

$$x \cdot y = y \cdot x$$

Сочетательный закон:

$$x + y + z = x + (y + z) = (x + y) + z$$

$$x \cdot y \cdot z = x \cdot (y \cdot z) = (x \cdot z) \cdot y$$

Распределительный закон:

$$x \cdot (y + z) = x \cdot y + x \cdot z$$

$$x + y \cdot z = (x + y) \cdot (x + z)$$

Закон двойственности:

$$\overline{x + y} = \overline{x} \cdot \overline{y}$$

$$\overline{x \cdot y} = \overline{x} + \overline{y}$$

Таблица 1.1

Обозначения логических операций		**Таблица истинности**					**Как читается**	**Название операции**
Основные	**Дополнительные**	x_1	**0**	**0**	**1**	**1**		
		x_2	**0**	**1**	**0**	**1**		
$x_1 \cdot x_2$	$x_1 x_2$; $x_1 \wedge x_2$; $x_1 \,\&\, x_2$	$x_1 \cdot x_2$	0	0	0	1	x_1 и x_2	Конъюнкция, логическое И, логическое произведение
$x_1 \vee x_2$	$x_1 + x_2$	$x_1 \vee x_2$	0	1	1	1	x_1 или x_2	Дизъюнкция, логическое ИЛИ, логическое сложение
$x_1 \to x_2$	$x_1 \supset x_2$	$x_1 \to x_2$	1	1	0	1	x_1 влечет x_2	Импликация
$x_1 \sim x_2$	$x_1 \equiv x_2$ $x_1 \leftrightarrow x_2$	$x_1 \sim x_2$	1	0	0	1	x_1 эквивалентно x_2	Эквивалентность, функция равнозначности
$x_1 \oplus x_2$	$x_1 \forall x_2$	$x_1 \oplus x_2$	0	1	1	0	x_1 неэквивалентно x_2	Сумма по модулю 2; функция неравнозначности; исключающее ИЛИ
$x_1 \downarrow x_2$	—	$x_1 \downarrow x_2$	1	0	0	0	ни x_1, ни x_2	Логическое ИЛИ- НЕ; Стрелка Пирса
$x_1 \mid x_2$	—	$x_1 \mid x_2$	1	1	1	0	x_1 и x_2 несовместимы	Логическое И-НЕ; штрих Шеффера

Синтез логического устройства распадается на несколько этапов. На первом этапе требуется функцию, заданную в словесной, табличной или других формах, представить в виде логического выражения с использованием некоторого

базиса. Дальнейшие этапы сводятся к получению минимальных форм функций, обеспечивающих при синтезе наименьшее количество электронного оборудования и рациональное построение функциональной схемы устройства.

Исходными, из соображений удобства последующих преобразований, приняты две канонические формы представления функций: совершенная дизъюнктивная нормальная форма (СДНФ) и совершенная конъюнктивная нормальная форма (СКНФ).

СДНФ.

Дизъюнктивной нормальной формой (ДНФ) называется такая форма представления функции, при которой логическое выражение функции строится в виде дизъюнкции ряда членов, каждый из которых является простой конъюнкцией аргументов или их инверсий.

Примером ДНФ может служить выражение

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

Приведем форму представления функций, не являющейся ДНФ. Например, функция $f(x_1, x_2, x_3) = x_1 \vee x_2 \cdot \overline{x_3} \vee \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \vee \overline{x_2 \cdot x_3}$ представлена не в ДНФ, так как последний член не является простой конъюнкцией аргументов.

Также не является ДНФ следующая форма представления функции:

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

Если в каждом члене ДНФ представлены все аргументы (или их инверсии) функции, то такая форма носит название *совершенной дизъюнктивной нормальной формы* (СДНФ). Можно сформулировать следующее правило записи СДНФ функции, заданной таблицей истинности.

Необходимо записать столько членов в виде конъюнкции всех аргументов, сколько единиц содержит функция в таблице. Каждая конъюнкция должна соответствовать определенному набору значений аргументов, обращающему функцию в единицу, и если в этом наборе значение аргумента равно нулю, то в конъюнкцию входит инверсия данного аргумента.

Следует отметить, что любая функция имеет единственную СДНФ.

СКНФ.

Конъюнктивной нормальной формой (КНФ) называется такая форма представления функции, при которой логическое выражение функции строится в виде конъюнкции ряда членов, каждый из которых является простой дизъюнкцией аргументов или их инверсий.

Примером КНФ может служить выражение

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

Приведем форму представления функций, не являющейся КНФ:

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

(здесь третий член не является простой дизъюнкцией аргументов или их инверсий);

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

(эта форма также не является КНФ, так как в ней первый член не связан с остальными операцией конъюнкции).

В *совершенной конъюнктивной нормальной форме* (СКНФ) в каждом члене КНФ должны быть представлены все аргументы.

Правило записи СКНФ функции, заданной таблицей истинности: следует записать столько конъюнктивных членов, представляющих собой дизъюнкции всех аргументов, при скольких наборах значений аргументов функция равна нулю, и если в наборе значение аргумента равно единице, то в дизъюнкцию входит инверсия данного аргумента. Любая функция имеет единственную СКНФ.

Структурная схема логического устройства может быть построена по канонической форме (СКНФ и СДНФ) реализуемой функции. Недостаток такого метода построения структурных схем состоит в том, что получающиеся схемы, как правило, оказываются неоправданно сложными, требуют использования большого числа логических элементов и, следовательно, имеют низкие экономичность и надежность.

Во многих случаях удается так упростить логическое выражение, не нарушая функции, что соответствующая структурная схема оказывается существенно более простой.

Методы такого упрощения функции, называются *методами минимизации функций.* В простых случаях минимизацию можно осуществить, непосредственно используя основные законы алгебры. В качестве примера упростим выражение:

$$y = x_3 x_2 \overline{x_1} + x_3 \overline{x_2} x_1 + x_3 \overline{x_2}\,\overline{x_1} + \overline{x_3}\,\overline{x_2} x_1 = x_3 \overline{x_1}(x_2 + \overline{x_2}) + \overline{x_2} x_1 (x_3 + \overline{x_3}) = x_3 \overline{x_1} + \overline{x_2} x_1$$

Полученное выражение равносильно исходному, но намного проще его.

Пусть имеется логическая функция

$$y = x_3 x_2 \overline{x_1} + x_3 \overline{x_2} x_1 + \overline{x_3} x_2 x_1 + x_3 x_2 x_1$$

Добавим дважды к ее правой части уже имеющийся член $x_3 x_2 x_1$ (отчего функция не изменится); тогда

$$y = x_3 x_2 \overline{x_1} + x_3 \overline{x_2} x_1 + \overline{x_3} x_2 x_1 + x_3 x_2 x_1 + x_3 x_2 x_1 + x_3 x_2 x_1 = x_3 x_2 (x_1 + \overline{x_1}) + x_2 x_1 (x_3 + \overline{x_3})$$
$$+ x_3 x_1 (x_2 + \overline{x_2}) = x_3 x_2 + x_2 x_1 + x_3 x_1.$$

И это выражение значительно проще исходного.

Следует отметить, что такие элементарные приемы минимизации удается использовать не часто- при малом количестве членов функции и небольшом числе переменных. В других случаях применяются специальные методы ми-

нимизации, облегчающие поиск склеивающихся членов. К ним относится метод минимизации с помощью карт Вейча.

Карта Вейча построена так, что в ее соседние клетки попадают смежные члены функции – члены, отличающиеся значением одной переменной: в один член эта переменная входит в прямой форме, а в другой – в инверсной. В карте содержится 2^n клеток, где ***n***-число переменных. Так, при ***n***=2 карта содержит четыре клетки (рис. 1,а), а при ***n***=3- восемь клеток (рис. 1,б), при ***n***=4-16 клеток (рис. 1,в).

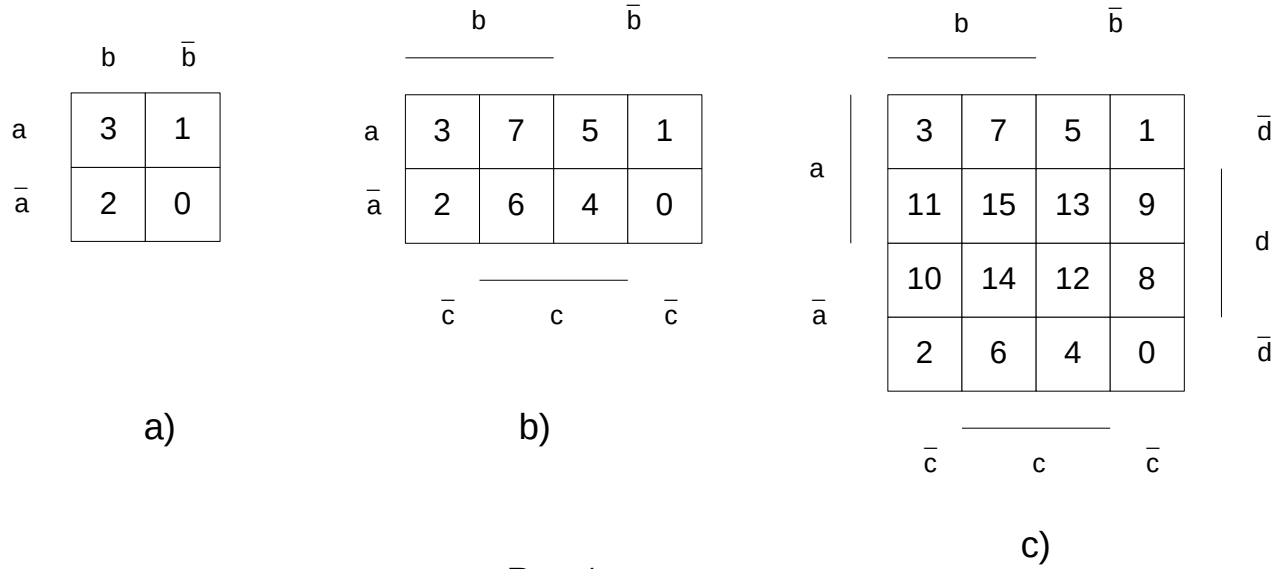


Рис.1

Каждая клетка соответствует определенному набору значений аргументов. Например, клетка с числом **0** (Рис.1,с) соответствует набору $\bar{a}\cdot\bar{b}\cdot\bar{c}\cdot\bar{d}$. Число клеток карты равно числу всех возможных наборов значений аргументов 2^n (***n***- число аргументов функции). В каждую из клеток карты записывается значение функции на соответствующем этой клетке наборе значений аргументов. Пусть функция задана таблицей истинности:

d	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
c	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**
b	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**
A	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
F_1(a,b,c,d)	0	0	0	1	1	1	0	1	1	0	0	0	1	1	0	1

Таблица истинности этой функции в форме карты Вейча представлена на Рис.2

Как видим, карта Вейча определяет значения функции на всех возможных наборах значений аргументов и ,таким образом, является таблицей истинности.

Сформулируем правила получения МДНФ функции с помощью карт 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,…Таким образом, допустимое число клеток в области – 1,2,4,8,..Области могут пересекаться и одни и те же клетки могут входить в разные области.

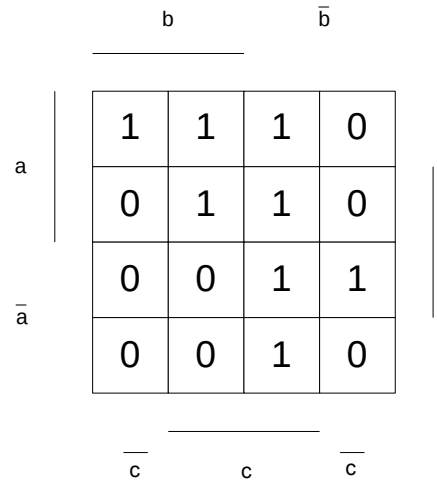


Рис. 2

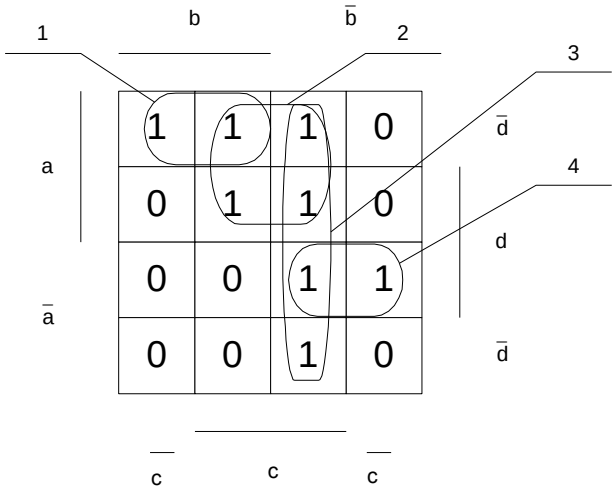


Рис. 3

Затем проводится запись выражения МДНФ функции. Каждая из областей в МДНФ представляется членом, число букв в котором на **k** меньше общего числа аргументов функции **n** (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, чтобы число областей было минимальным (при этом минимальным будет число членов в МДНФ функции), а каждая область содержала возможно большее число клеток.

Найдём МДНФ для функции изображённой на Рис.3

Первая и четвёртая области имеют по две клетки, для них **n-k=4-1=3**. Эти области будут в МДНФ представлены членами, содержащими по три буквы.

Вторая и третья области содержат по четыре клетки и в МДНФ выражаются членами, содержащими по две буквы (**n-k=4-2=2**).

МДНФ будет иметь вид

$$F_1 = a \cdot b \cdot \bar{d} \vee a \cdot c \vee \bar{b} \cdot c \vee \bar{a} \cdot \bar{b} \cdot d$$

При построении замкнутых областей допускается сворачивание карты в цилиндр с объединением противоположных граней. В силу этого крайние клетки строки или столбца таблицы рассматриваются как соседние и могут быть объединены в общую область(Рис.4).

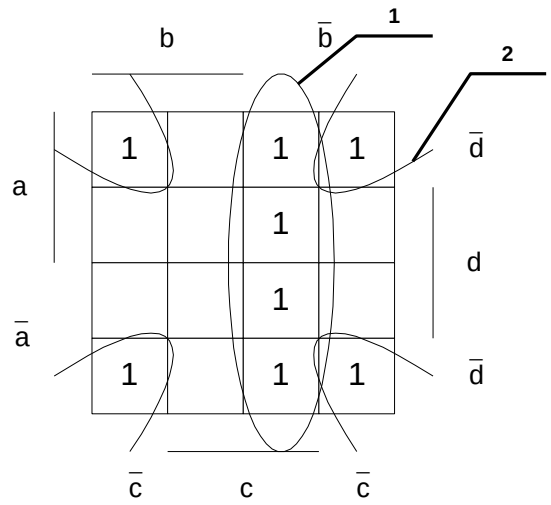


Рис.4

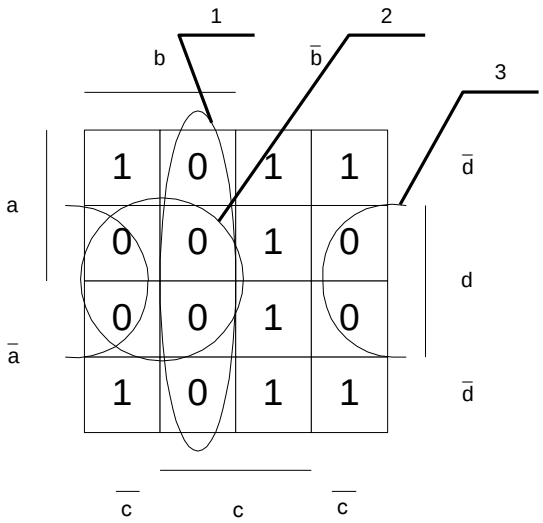


Рис. 5

Минимальная ДНФ функции

$$F_2 = \bar{c} \cdot \bar{d} \vee \bar{b} \cdot c$$

Для получения МКНФ функции замкнутыми областями охватываются клетки с нулевыми значениями функции, и при записи членов логического выражения берутся инверсии аргументов, на пересечении которых находятся области. Так, для функции, приведённой на Рис.5,МКНФ

$$F = (\bar{b} \vee \bar{c}) \cdot (\bar{b} \vee \bar{d}) \cdot (c \vee \bar{d})$$

Для минимизации функций с числом аргументов, большим пяти, карты Вейча оказываются неудобными. Минимизация таких функций может быть выполнена методом Квайна [Л5 стр.128-137].

1.5. Формы и форматы представления информации в ЭВМ

С учетом сказанного ранее можно сделать вывод о том, что числа в ЭВМ отображаются в виде последовательности нулей и единиц.

Формы этого отображения могут быть различными.

Во всех рассмотренных ранее примерах использовалась форма, называемая формой с фиксированной точкой, причем та ее разновидность, когда точка фиксируется после младшего разряда. В этой форме могут быть представлены только целые числа со знаком.

Используется и другая разновидность формы с фиксированной точкой,

когда точка фиксируется перед старшим разрядом. В этой форме могут быть представлены числа по модулю меньше единицы.

Использование обеих этих форм приводит к необходимости заботиться о масштабировании исходных данных, промежуточных и конечных результатов решения задачи.

Помимо этих форм в необходимых случаях используется форма с плавающей точкой, причем та ее разновидность, которая называется нормализованной. Поскольку диапазон представления чисел в форме с плавающей точкой значительно шире, а относительная погрешность представления меньше, то эта форма является значительно менее критичной к необходимости масштабирования.

До сих пор рассматривался только один тип данных - числа. Но в ЭВМ могут отображаться с помощью последовательности нулей и единиц и другие типы данных.

Процессор способен обрабатывать большой набор различных типов данных:

Данные без знака.

Данные со знаком. Представление таких данных и выполнение операций производится в дополнительном коде. Под знак отводится старший разряд формата.

Двоично-десятичные данные. Процессор поддерживает двоично-десятичное представление чисел в формате 8-разрядного неупакованного, 8-разрядного упакованного и 80-разрядного упакованного представления. Первые два представления поддерживаются процессором, третье - сопроцессором.

Данные типа строка. Строка представляет собой непрерывную последовательность бит, байт, слов или двойных слов. Строка бит может быть длиной до 1 Гбит, длина остальных строк - от 1 байта до 4 Гбайт.

Символьные данные. Поддерживаются строки символов в коде ASCII и арифметические операции (сложение и умножение) над ними.

Основной информационной единицей , которой оперируют устройства ЭВМ является машинное слово, представляющее собой фиксированное количество разрядов, каждый из которых может иметь значение 0 или 1, т.е. содержать один бит информации. Промежуточной информационной единицей между битом и словом является байт, равный 8 битам. Слово может содержать от одного до нескольких байт.

Количество разрядов в машинном слове называется иногда разрядной сеткой.

Форматом называется разрядная сетка с разметкой функционального назначения отдельных разрядов или их групп.

Процессор оперирует с данными, представленными в одном из следующих 7 форматов :

16-разрядные целые двоичные числа.
32-разрядные целые двоичные числа.
64-разрядные целые двоичные числа.
80-разрядные целые двоично-десятичные числа.
32-разрядные вещественные числа.
64-разрядные вещественные числа.
80-разрядные вещественные числа.

В первых трех форматах двоичные числа представляются в дополнительном коде. В качестве знакового разряда используется старший бит числа. Комбинация, когда в старшем разряде 1, а в остальных разрядах нули, используется для обозначения неопределенности. Это значение устанавливается в качестве результата выполнения недопустимой информации.

Двоично-десятичные числа представляются в упакованной форме и содержат 18 тетрад, каждая из которых соответствует одному десятичному разряду. Для представления знака используется старший разряд формата. Неопределенность в этом формате представляется комбинацией, в которой два старших байта содержат во всех разрядах 1, а все остальные разряды формата могут иметь произвольное значение.

Вещественные числа в каждом из трех отведенных для них форматах (5, 6, 7) имеют три поля :

- поле знака мантиссы - старший разряд формата,
- поле порядка,
- поле мантиссы.

В пятом формате под мантиссу отведено 23 разряда, а под порядок - 8 разрядов. В шестом формате под мантиссу отведено 52 разряда, а под порядок - 11 разрядов. В седьмом формате под мантиссу отведено 64 разряда, а под порядок - 15 разрядов.

Мантисса записывается в нормализованном виде.

Двоичное число без знака в поле порядка указывает смещенный порядок. Истинный порядок числа равен E - P, где E - содержимое поля порядка, а P зависит от формата. В пятом формате P=127, в шестом формате P=1024, в седьмом формате P=16383.

Пятый и шестой форматы достаточны для решения практически любых задач.

Однако внутри сопроцессора используется только седьмой формат, позволяющий избежать потери точности результата при выполнении многократных операций над вещественными числами. Для этого над всеми числами, хранящимися в памяти во всех остальных форматах, при вводе в сопроцессор осуществляется преобразование в седьмой формат. При записи результата в память в случае необходимости осуществляется обратное преобразование.

1.6 Серии логических элементов

Цифровые микросхемы предназначены для обработки, преобразования и хранения цифровой информации. Выпускаются они сериями. Микросхемы, входящие в состав каждой серии, имеют единое конструктивно-технологическое исполнение, единый номинал напряжения питания, одинаковые уровни сигналов логического 0 и логической 1. Все это делает микросхемы одной серии совместимыми, т.е. обеспечивает возможность построения из них сколь угодно сложных устройств.

Основой каждой серии является базовая схема. Как правило, в качестве базовой выбирают схему, выполняющую логическую операцию И-НЕ, либо ИЛИ-НЕ. По принципу построения базовых схем цифровые микросхемы делятся на следующие основные типы: элементы диодно-транзисторной логики (ДТЛ), резистивно-транзисторной логики (РТЛ), транзисторно-транзисторной логики (ТТЛ), эмиттерно-связанной логики (ЭСЛ) и микросхемы на основе МОП и комплиментарных МОП (КМОП)-транзисторов.

Наиболее широкое применение находят микросхемы, изготавливаемые по ТТЛ (ТТЛШ)-, n-МОП и КМОП - технологиям. Каждая технология непрерывно совершенствуется с целью увеличения быстродействия, уменьшения потребляемой мощности и увеличения степени интеграции, т.е. числа активных элементов, размещаемых на кристалле заданной площади.

Наибольший интерес с точки зрения изучения представляют универсальные серии микросхем. Типичным примером является серия ТТЛ К155. На рис.6 приведена базовая схема этой серии, реализующая функцию И-НЕ.

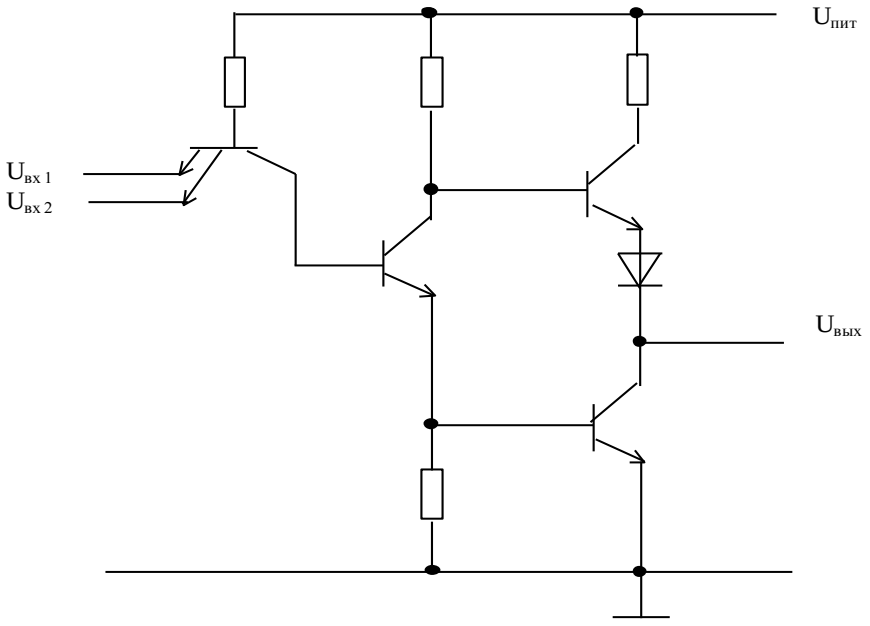


Рис.6

Основными параметрами любой серии являются напряжение источника питания, уровни напряжений логического 0 и логической 1, нагрузочная способность, помехоустойчивость, быстродействие и потребляемая мощность.

Микросхемы ТТЛ рассчитаны на напряжение питания +5В $\pm$ 10%. Различают пороговое напряжение логической единицы $U^1_{пор}$–наименьшее напряжение высокого уровня на входе микросхемы, которое воспринимается ею как логическая единица, и пороговое напряжение логического нуля $U^0_{пор}$– наибольшее напряжение низкого уровня на входе микросхемы, которое воспринимается ею как логический нуль. Для микросхем серий ТТЛ $U^1_{пор}$=2,4 В, $U^0_{пор}$=0,4 В. Напряжения высокого и низкого уровней на выходах микросхем ТТЛ $U^1_{пор} \geq 2{,}4$ В и $U^0_{пор} \leq 0{,}4$ В.

Нагрузочная способность микросхем определяется числом входов других микросхем, которое без дополнительных согласующих устройств может быть подключено к одному выходу микросхемы. Справочным параметром, характеризующим нагрузочную способность, является коэффициент разветвления по выходу, который численно равен количеству единичных нагрузок, которые можно одновременно подключить к выходу микросхемы. Единичной нагрузкой является вход базовой схемы данной серии. Коэффициент разветвления по выходу большинства микросхем серии К155 равен 10, серии 555–20, серии 1531 – 33.

Помехоустойчивость микросхем оценивают в статическом и динамическом режимах. Статическая помехоустойчивость определяется уровнем напряжения, подаваемого на вход элемента, относительно уровня логического нуля и логической единицы, при котором состояние на выходе не меняется. Для микросхем ТТЛ статическая помехоустойчивость составляет не менее 0,4 В. Динамическая помехоустойчивость зависит от формы и амплитуды помехи, а также скорости переключения схемы и ее статической помехоустойчивости.

Одним из основных динамических параметров микросхем является быстродействие. Быстродействие обычно оценивается средним временем задержки распространения сигнала

$$t_{зд.ср} = 0{,}5(t^{1,0}_{зд.} + t^{0,1}_{зд.}),$$

где $t^{1,0}_{зд.}$ и $t^{0,1}_{зд}$ - времена задержки распространения сигнала при включении и выключении, определяемые в соответствии с рис.7 Для микросхем серии К155 $t_{зд.ср} \approx 20нс$.

Потребляемая микросхемой мощность в статическом режиме различна при уровнях логического нуля (P^0) и логической единицы (P^1) на выходе. В связи с этим для характеристики потребления используют среднюю мощность потребления $P_{ср} = 0{,}5(P^0 + P^1)$. Эта мощность для микросхем серии К155 составляет несколько десятков мВт. Мощность, потребляемая при работе в динамическом режиме, возрастает. Поэтому, помимо $P_{ср}$, в справочниках задается также $P_{дин}$, измеряемая на максимальной частоте переключения.

Микросхемы серии 155 являются изделиями массового применения. В составе этой и родственных ей серий (133, 555, 531, 1531, 1533) имеется свыше 100 типономиналов микросхем, которые можно разделить на отдельные функциональные группы: логические элементы, более сложные комбинационные устройства (автоматы без памяти), к которым относятся дешифраторы, шифраторы, мультиплексоры, сумматоры и другие типовые схемы, последовательностые устройства (автоматы с памятью), к которым относятся триггеры, регистры, счетчики. В дальнейшем при описании конкретных микросхем будем придерживаться указанного функционального разбиения, ориентируясь, главным образом на серию К155.

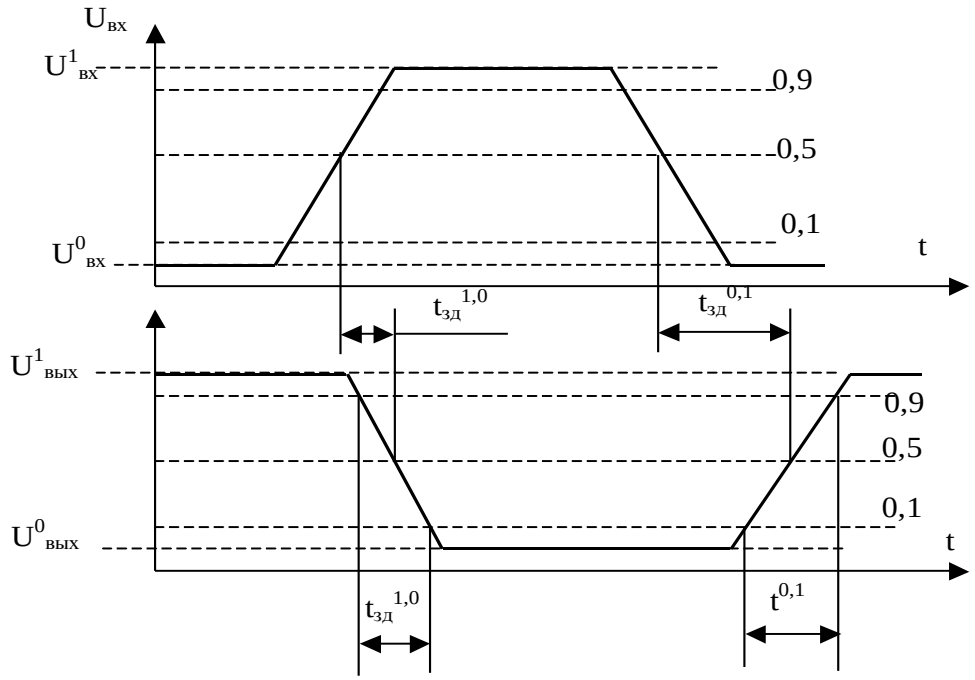


Рис.7

Одной из названных ранее функциональных групп интегральных микросхем (ИС) серии К155 является группа логических элементов. Она включает в себя ИС, реализующие основные элементарные логические функции: НЕ, И, ИЛИ, И-НЕ, ИЛИ-НЕ, И-ИЛИ-НЕ, ИСКЛЮЧАЮЩЕЕ ИЛИ.

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

Логическая ИС, как конструктивный узел (корпус), может содержать несколько логических элементов, каждый из которых выполняет одну из упомянутых выше логических операций.

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 используются 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.

2 ТИПОВЫЕ И ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ И УЗЛЫ ЭВМ

2.1. Типовые комбинационные устройства

2.1.1. Общие положения

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 главе 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, демультиплексоры, сумматоры, компараторы, схемы формирования и контроля разряда паритета и множество других схем.

2.1.2. Дешифраторы

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 $\mathbf{2^n}$ 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 $\mathbf{2^n} = 8$. Обозначим входы $\mathbf{x_0, x_1, x_2}$, а выходы $\mathbf{y_0 \div y_7}$.

Тогда аналитическое задание функции дешифратора такой размерности будет выглядеть следующим образом :

$$y_0 = \bar{x}_2 \cdot \bar{x}_1 \cdot \bar{x}_0, y_1 = \bar{x}_2 \cdot \bar{x}_1 \cdot x_0, y_2 = \bar{x}_2 \cdot x_1 \cdot \bar{x}_0, y_3 = \bar{x}_2 \cdot x_1 \cdot x_0,$$

$$y_4 = x_2 \cdot \bar{x}_1 \cdot \bar{x}_0, y_5 = x_2 \cdot \bar{x}_1 \cdot x_0, y_6 = x_2 \cdot x_1 \cdot \bar{x}_0, y_7 = x_2 \cdot x_1 \cdot x_0$$

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. В соответствии с этим аналитическим описанием может быть построена схема дешифратора размерностью 3x8 (рис.8). Его УГО выглядит следующим образом (рис.9)

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

2.1.3. Шифраторы

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 $\mathbf{2^n}$. При этом выходной код представляет собой двоичный эквивалент номера входа, на котором активный уровень сигнала. Сказанное можно считать словесным заданием функции шифратора. На основании этого словесного задания, положив число входов шифратора равным $8 = 2^3$ и обозначив их **I**0 - **I**7, составим таблицу истинности шифратора (табл.2.1), в которой число выходов шифратора равно 3 и обозначены они **A**0 - **A**2.

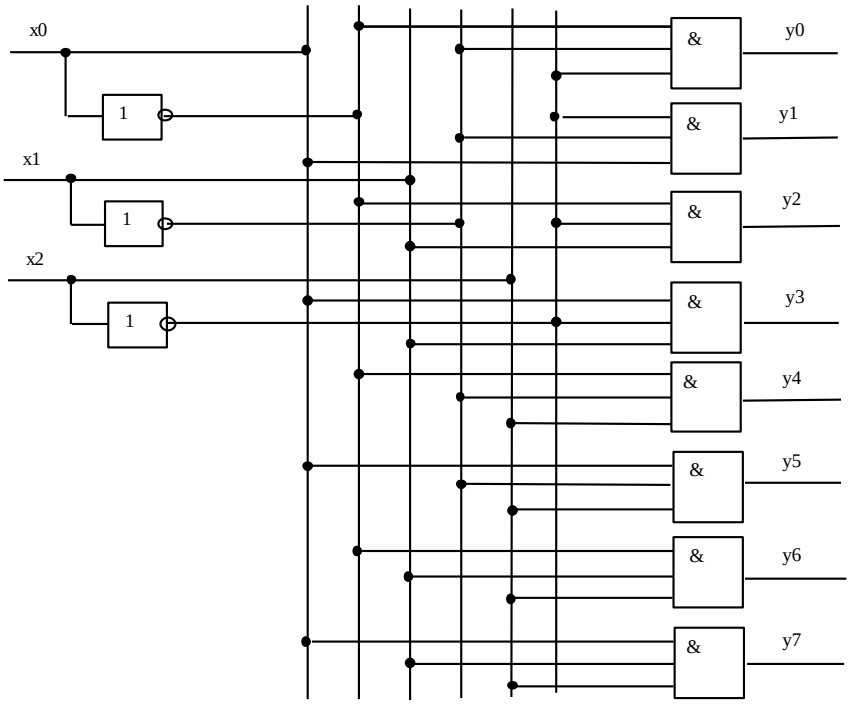


Рис.8

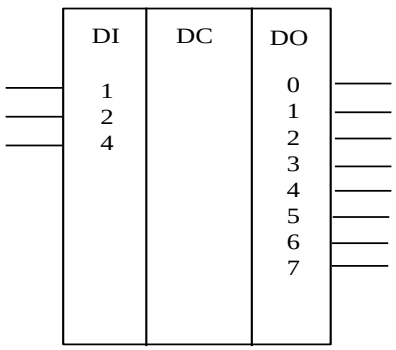


Рис.9

Таблица 2.1

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.10)

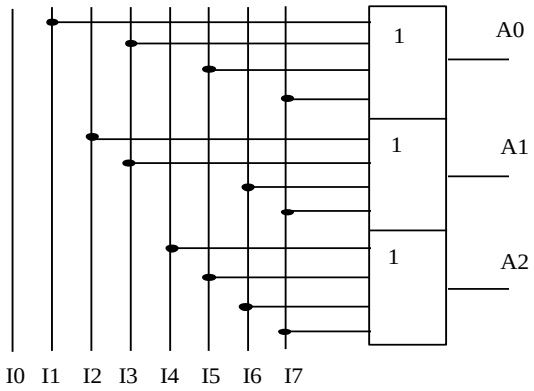


Рис.10

УГО приведенной схемы шифратора выглядит следующим образом (рис.11).

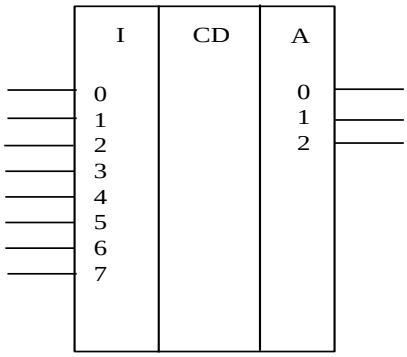


Рис. 11

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 единицы преобразует его в унитарный, выделив лишь самую старшую единицу в поданной на входы комбинации и проигнорировав все остальные единицы.

2.1.4. Мультиплексоры

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 равным **n**=3 и обозначим их **A1 - A3**. Тогда число информационных входов равно $\mathbf{2^n}$ = 8. Обозначим их **D0 - D7**. С учетом этого аналитическое выражение для функции, реализуемой на выходе F мультиплексора, можно записать следующим образом :

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

В соответствии с этим выражением схему мультиплексора можно представить следующим образом (рис. 12).

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.

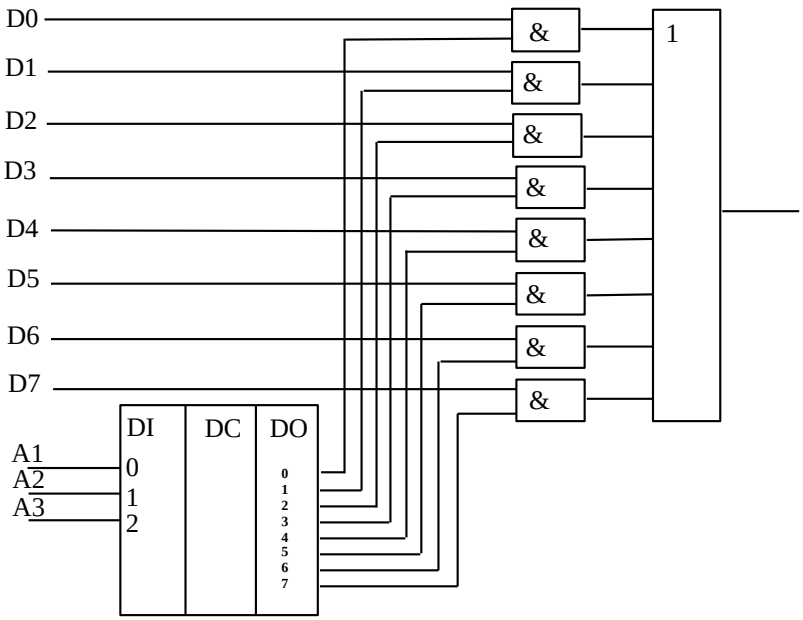


Рис.12

УГО этой схемы выглядит следующим образом (рис13).

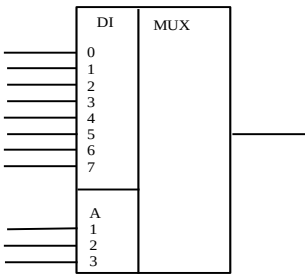


Рис.13

2.1.5. Демультиплексоры

Исходя из названия в функциональном отношении демультиплексор реализует функцию, обратную функции мультиплексора. В соответсвии 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 $\mathbf{2^n}$ выходов. Положим n=3 и обозначим адресные входы $\mathbf{X_0, X_1, X_2}$, а $2^n = 8$ выходов обозначим $\mathbf{Y_0 \div Y_7}$. Обозначим информационный вход демультиплексора **E**. Тогда аналитическое задание функции демультиплексора такой размерности будет выглядеть следующим образом :

$$Y_0 = E \cdot \overline{X}_0 \cdot \overline{X}_1 \cdot \overline{X}_2, Y_1 = E \cdot X_0 \cdot \overline{X}_1 \cdot \overline{X}_2, Y_2 = E \cdot \overline{X}_0 \cdot X_1 \cdot \overline{X}_2, Y_3 = E \cdot X_0 \cdot X_1 \cdot \overline{X}_2,$$

$$Y_4 = E \cdot \overline{X}_0 \cdot \overline{X}_1 \cdot X_2, Y_5 = E \cdot X_0 \cdot \overline{X}_1 \cdot X_2, Y_6 = E \cdot \overline{X}_0 \cdot X_1 \cdot X_2, Y_7 = E \cdot X_0 \cdot X_1 \cdot X_2.$$

Но эта система описывает дешифратор с тремя входами $\mathbf{X_0, X_1, X_2}$, входом стробирования **E** и восемью выходами $\mathbf{Y_0 \div Y_7}$. С учетом сказанного можно рассматривать дешифратор со входом стробирования как обращенный по

входам демультиплексор, у которого селектирующие входы стали адресными входами дешифратора, а информационный вход - входом стробирования дешифратора.

2.1.6.Сумматоры

Сумматоры представляют собой функциональные цифровые устройства, выполняющие сложение чисел. Суммирование осуществляется в двоичном или, реже, двоично-десятичном коде. По характеру действия сумматоры подразделяются на комбинационные и накапливающие, т.е. сохраняющие результат в специальном регистре. Каждый из многоразрядных сумматоров может

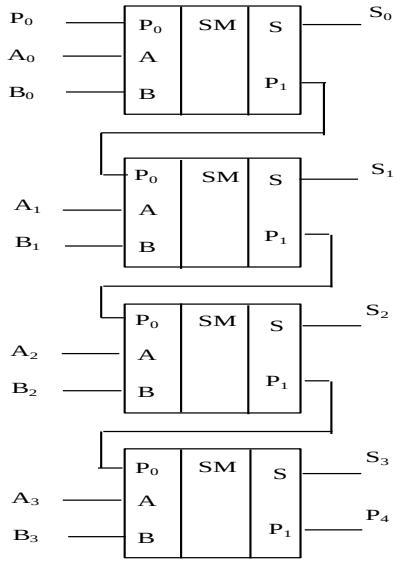


Рис.14

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 отдельно, но с учетом результата сложения в предыдущем разряде, т.е. с учетом переноса. Таким образом, каждый одноразрядный полный сумматор должен иметь один вход переноса $\mathbf{P_0}$, два входа слагаемых **A** и **B**, выход суммы **S** и выход переноса $\mathbf{P_1}$. Таблица истинности одноразрядного полного сумматора строится с учетом правил сложения одноразрядных двоичных чисел. Схема одноразрядного полного сумматора может быть реализована на основании этой таблицы. Для построения многоразрядного параллельного сумматора с последовательным переносом необходимо соответствующее количество одноразрядных полных сумматоров соединить последовательно по цепи переноса (рис.14).

Аналогично осуществляется наращивание разрядности при использовании микросхем многоразрядных сумматоров.

2.1.7. Компараторы

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.15).

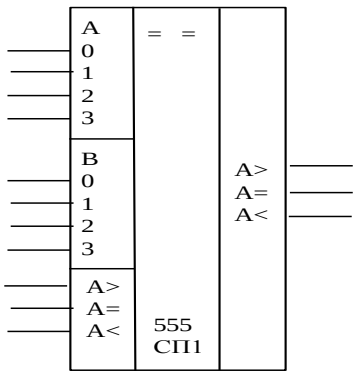


Рис.15

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

2.1.8. Схемы формирования и контроля разряда паритета

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.

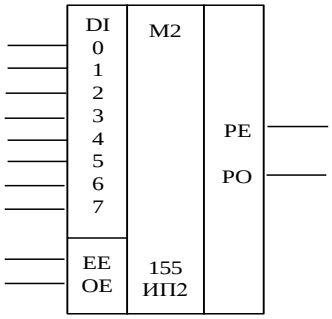


Рис.16

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 рисунке (рис.16).

2.1.9. Арифметическо - логическое устройство (АЛУ)

АЛУ является одной из составных частей операционного устройства процессора. АЛУ представляет собой комбинационное устройство, способное выполнять определенный набор арифметических и логических операций над многоразрядными двоичными словами. Вид операции, выполняемой АЛУ, задается двоичным кодом операции, подаваемым на специальные входы АЛУ.

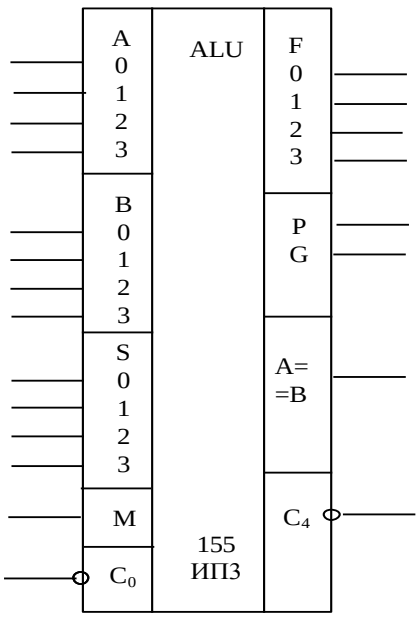


Рис.17

Наиболее простым примером АЛУ в интегральном исполнении является микросхема К155ИП3, УГО которой приведено на рисунке (рис.17).

2.2. Последовательностные устройства

2.2.1. Общие положения

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.

К основным типам последовательностных устройств относятся триггеры и реализуемые на их основе регистры и счетчики.

2.2.2. Триггеры

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 на его прямом выходе уровень логической единицы, а на инверсном - уровень логического нуля.

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. Как правило, триггер имеет два взаимно инверсных выхода Q и Q и один или два информационных входа, обозначаемых T или D для одновходовых триггеров и R и S или K и J для двухвходовых триггеров. Функциональная классификация является наиболее общей и характеризует состояние входов и выходов триггера в момент времени до его срабатывания (t-1) и после его срабатывания (t). Практическое применение из множества возможных типов триггеров нашли T-, D-,RS- и JK-триггеры, принципы работы которых иллюстрируются таблицами переходов (табл.3.1 и табл.4.1)

Таблица 3.1

Информационные входы	Тип триггера и значения сигналов на его выходе после срабатывания Q(t)	
D или T	D	T
0	0	Q(t-1)
1	1	$\overline{Q}(t-1)$

Таблица 4.1

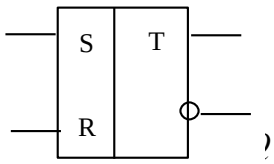
Ин- входы форм.		Тип триггера и значения сигналов на его выходе после срабатывания Q(t)	
R или K	S или J	RS - триггер	JK - триггер
0	0	Q(t-1)	Q(t-1)
0	1	1	1
1	0	0	0
1	1	неопределенное сост.	$\overline{Q}(t-1)$

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 по поступлении сигналов на информационные входы. Синхронные триггеры помимо информационных имеют тактирующий или синхронизирующий вход С, и переключение такого триггера происходит только при наличии соответствующего сигнала на этом входе.

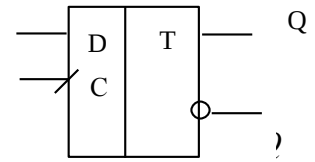
В зависимости от того, по какому параметру этого сигнала происходит переключение триггера, они подразделяются на триггеры со статическим управлением и триггеры с динамическим управлением. В триггерах со статическим управлением управляющим параметром является уровень на входе С, а в триггерах с динамическим управлением - фронт (передний или задний) сигнала на входе С. Недостатком триггеров со статическим управлением является то, что при активном уровне на входе С он превращается в асинхронный, поскольку переключается непосредственно по изменению сигналов на информационных входах.

УГО триггеров различных типов из рассмотренной классификации приведены на рис.18- 21.



асинхронный RS-триггер

Рис.18



синхронный D-триггер с прямым динамическим управлением

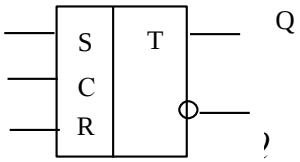
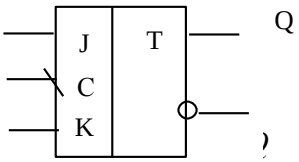


Рис.20

синхронный RS-триггер со статическим управлением

Рис.19



синхронный JK-триггер с инверсным динамическим управлением

Рис.21

2.2.3. Регистры

Упорядоченная совокупность триггеров, предназначенная для хранения многоразрядных двоичных слов или их частей, называется регистром. 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 (рис.22).

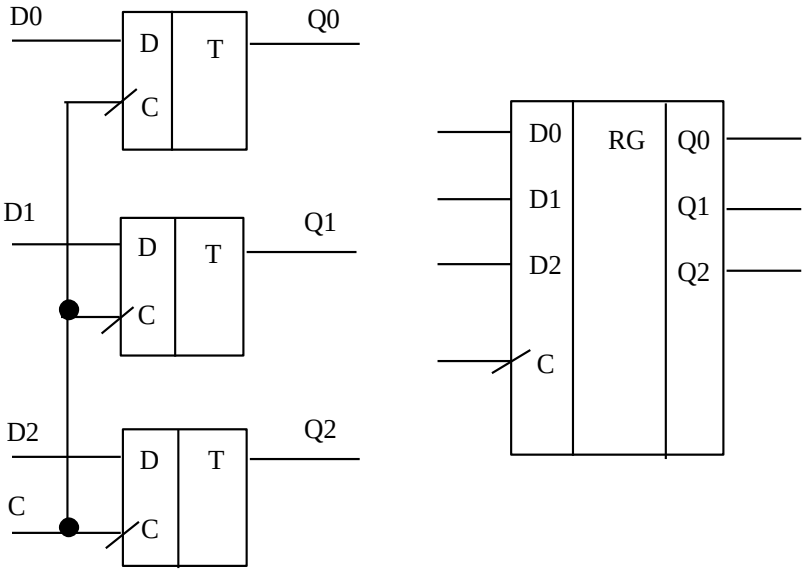


Рис.22

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 (рис.23).

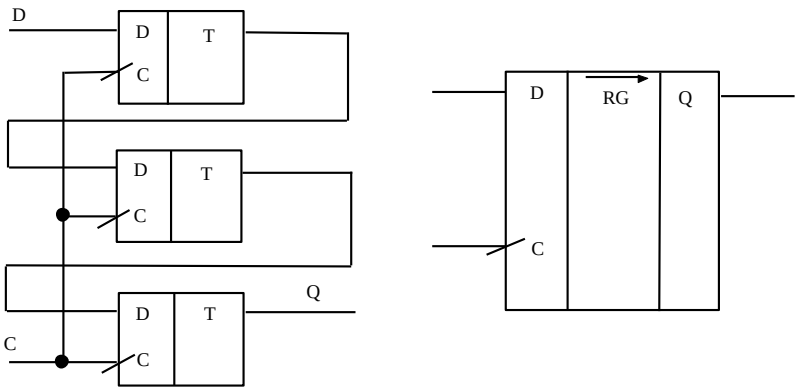


Рис.23

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 для обеспечения сдвига влево, либо с выходом последующего триггера для обеспечения сдвига вправо. 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 коммутировать вход каждого триггера с выходами соседних слева и справа триггеров.

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

2.2.4.Счетчики

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 $\mathbf{2^m}$ импульсов. Каждый из триггеров такой цепочки называют разрядом счетчика. Число **m** определяет количество разрядов двоичного числа, которое может быть записано в счетчике. Число $\mathbf{К_{сч}=2^m}$ 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 $\mathbf{К_{сч}}$, 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.

Цифровые счетчики классифицируются следующим образом :

*по коэффициенту счета - двоичные, двоично-десятичные или с другим

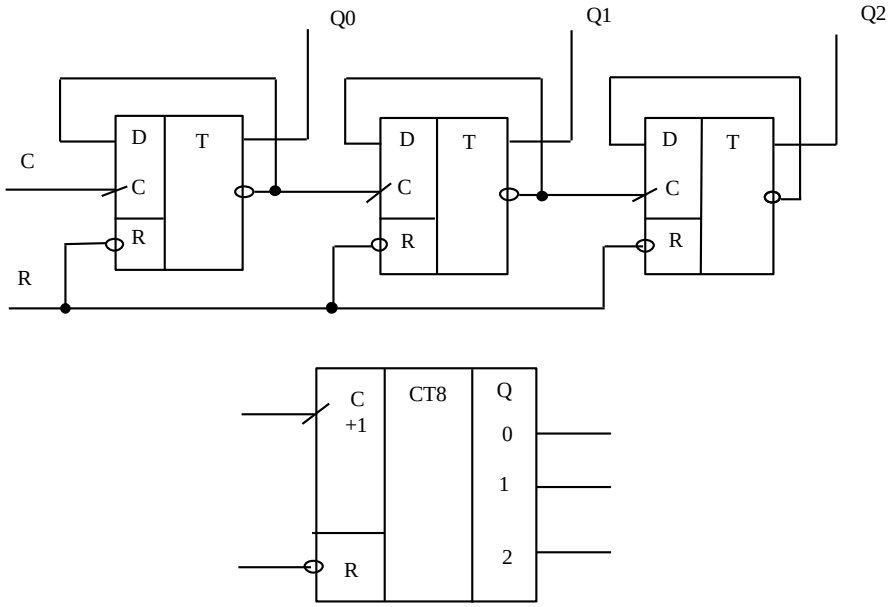


Рис.24

коэффициентом счета, с произвольным постоянным коэффициентом счета, с переменным коэффициентом счета;

*по направлению счета - суммирующие, вычитающие, реверсивные;

*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 $\mathbf{К_{сч}=2^m}$. Пример двоичного суммирующего счетчика с $К_{сч}$=8,реализованного на синхронных **D**-триггерах с прямым динамическим управлением, которые работают в режиме счетных триггеров, приведен на рис.24. Рядом приведено его УГО.

Для синтеза двоичного счетчика используется $\mathbf{m = \log_2 K_{сч}} = \log_2 8 = 3$

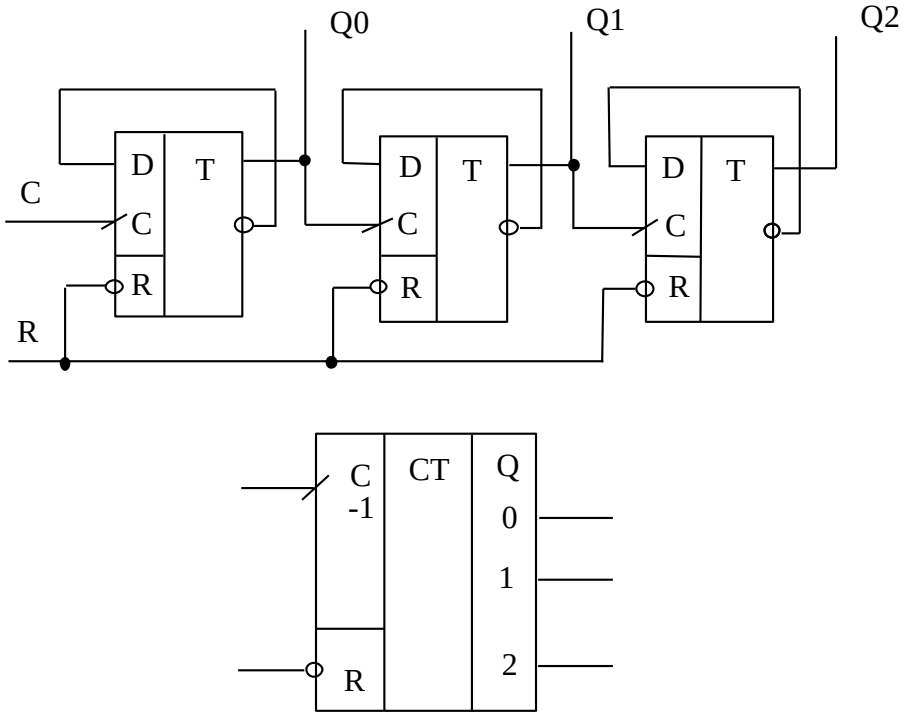


Рис.25

счетных триггеров. Поскольку для синтеза предложены синхронные **D**-триггеры, то следует их сделать счетными путем соединения инверсного выхода каждого триггера с его же входом **D**. Поскольку перенос из младшего разряда в следующий 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, то счетный вход каждого последующего триггера следует соединить с инверсным выходом предыдущего.

В суммирующем счетчике каждый входной импульс увеличивает число, записанное в счетчике, на единицу. Вычитающий счетчик действует обратным образом: двоичное число, хранящееся в счетчике, с каждым поступающим импульсом уменьшается на единицу. Переполнение вычитающего счетчика происходит после достижения им нулевого (исходного) состояния. Для примера на рис.25 приведен вычитающий счетчик на тех же триггерах с $\mathbf{K_{сч}}$=8.

Реверсивный счетчик может работать в качестве суммирующего и вычитающего. Такой счетчик имеет дополнительные управляющие входы для задания направления счета.

Введением дополнительных логических связей - обратных и прямых - двоичные счетчики могут быть обращены в недвоичные, для которых коэффициент пересчета не равен целой степени двойки. В качестве примера на рис.26. приведена схема суммирующего счетчика с $\mathbf{K_{сч}}$=5.

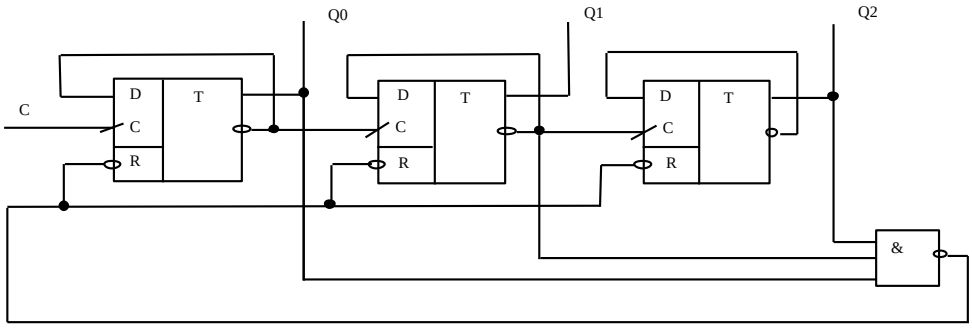


Рис.26

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. Основной недостаток 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 $К_{сч}=8$, реализованного на синхронных JK - триггерах приве-

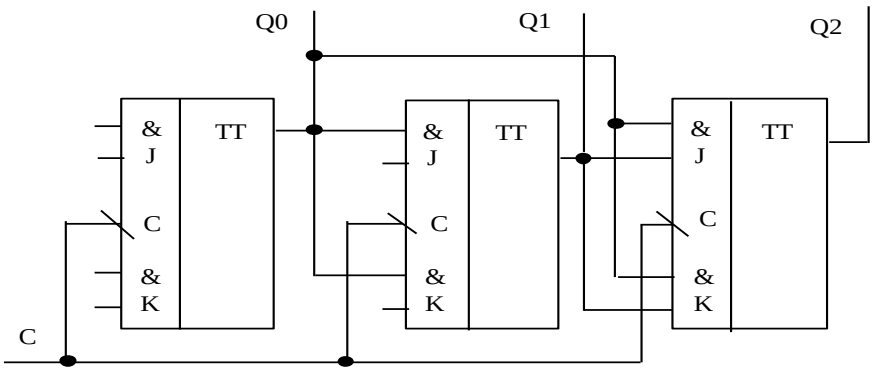


Рис.27

дена на рис.27.

Пример построения синхронного недвоичного счетчика с $К_{сч}$=3 приведен на рис.34.

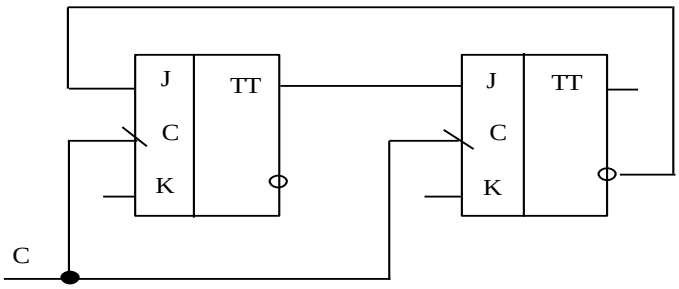


Рис.34

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 практических задач, возникающих при разработке цифровой аппаратуры.

3 ВАРИАНТЫ ЗАДАНИЙ

Тема 1. Позиционные системы счисления

1.1. Задание №1

Представить заданное в табл. 3.1 ПРИЛОЖЕНИЯ А:

а) десятичное число в двоичной, восьмеричной и шестнадцатеричной системах счисления;

б) двоичное число в десятичной, восьмеричной и шестнадцатеричной системах счисления;

в) восьмеричное число в двоичной, десятичной и шестнадцатеричной системах счисления;

г) шестнадцатеричное число в двоичной, десятичной и восьмеричной системах счисления.

Тема 2. Кодирование отрицательных чисел

2.1. Задание №2

а) представить заданные в табл. 3.2 ПРИЛОЖЕНИЯ А два десятичных числа в двоичном восьмиразрядном прямом, обратном и дополнительном кодах (старший разряд - знаковый);

б) выполнить сложение чисел из п. а), используя их представление в прямом коде.;

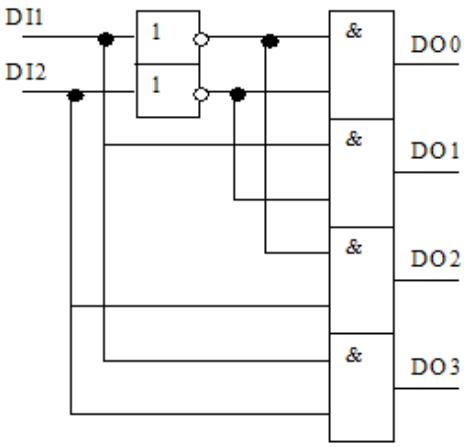
в) выполнить вычитание второго числа из первого, используя их представление в обратном коде. Повторить операцию, используя представление чисел в дополнительном коде.

г) выполнить вычитание первого числа из второго, используя их представление в обратном коде. Повторить операцию, используя представление чисел в дополнительном коде.

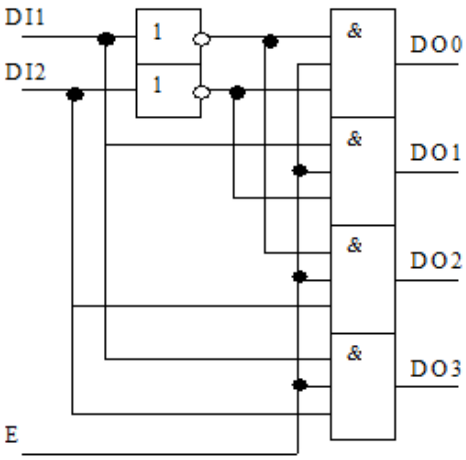
Тема 3. Логические функции и логические элементы

3.1. Задание №3

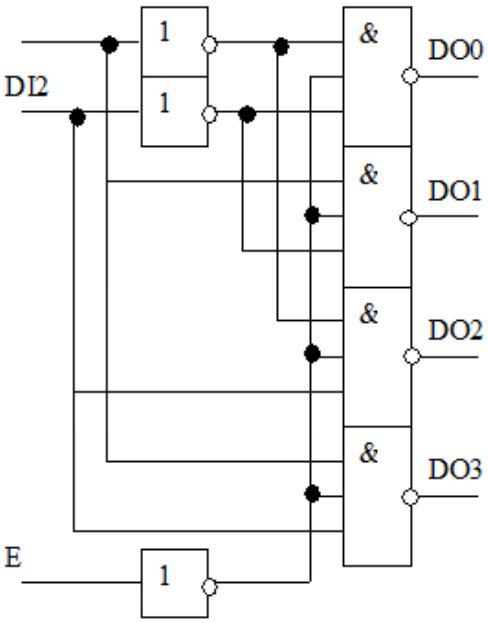
В данном разделе предлагается 21 вариант задач, представляющих собой схемы, построенные с использованием ИС логических элементов серии К155. Решение задачи состоит в составлении таблицы истинности и аналитических выражений, описывающих схему.



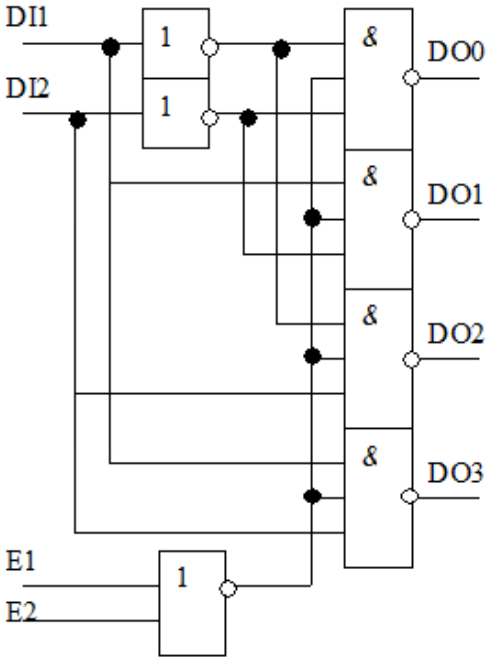
Вариант 1.



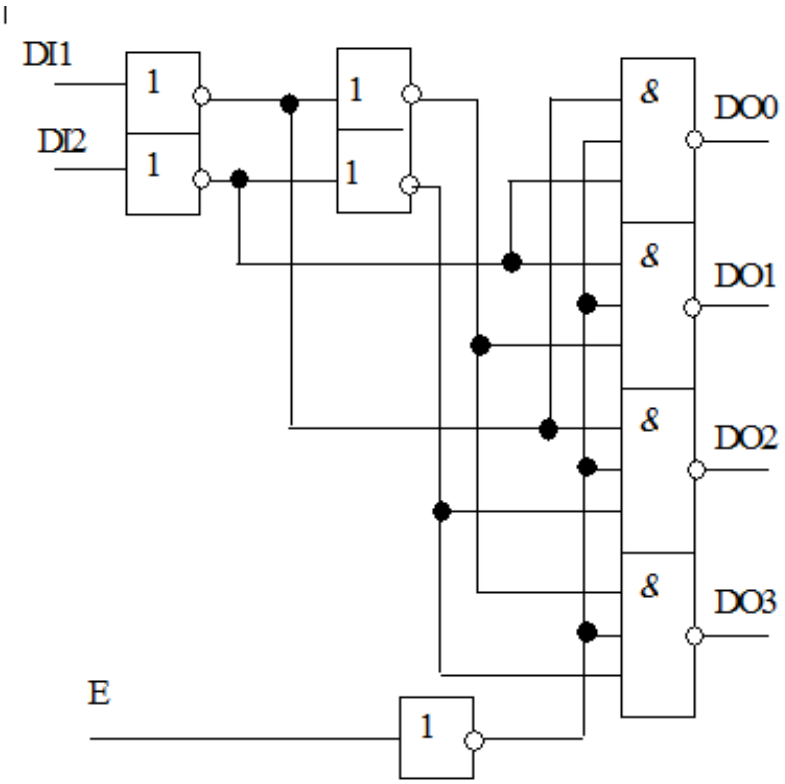
Вариант 2.



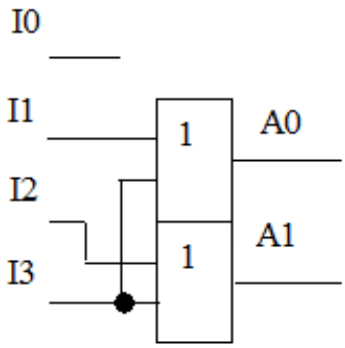
Вариант 3.



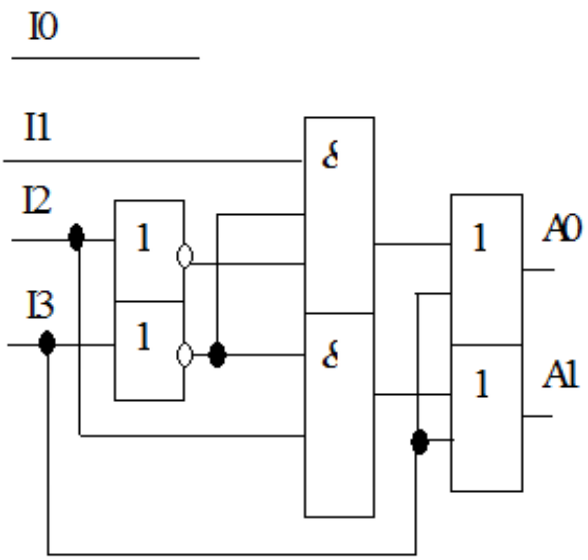
Вариант 4



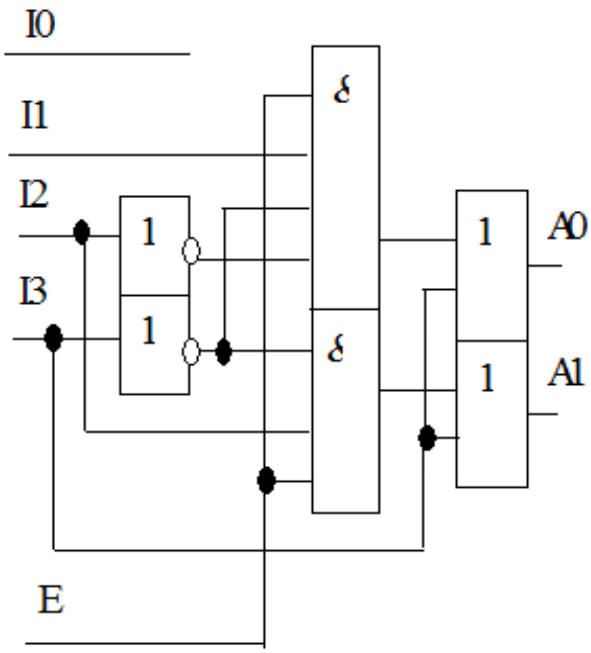
Вариант 5.



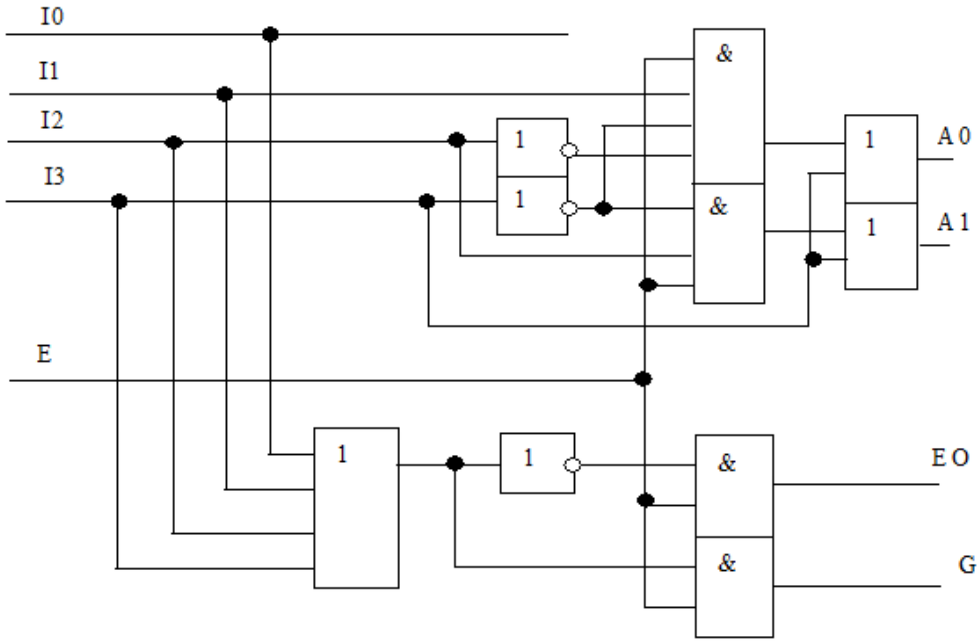
Вариант 6.



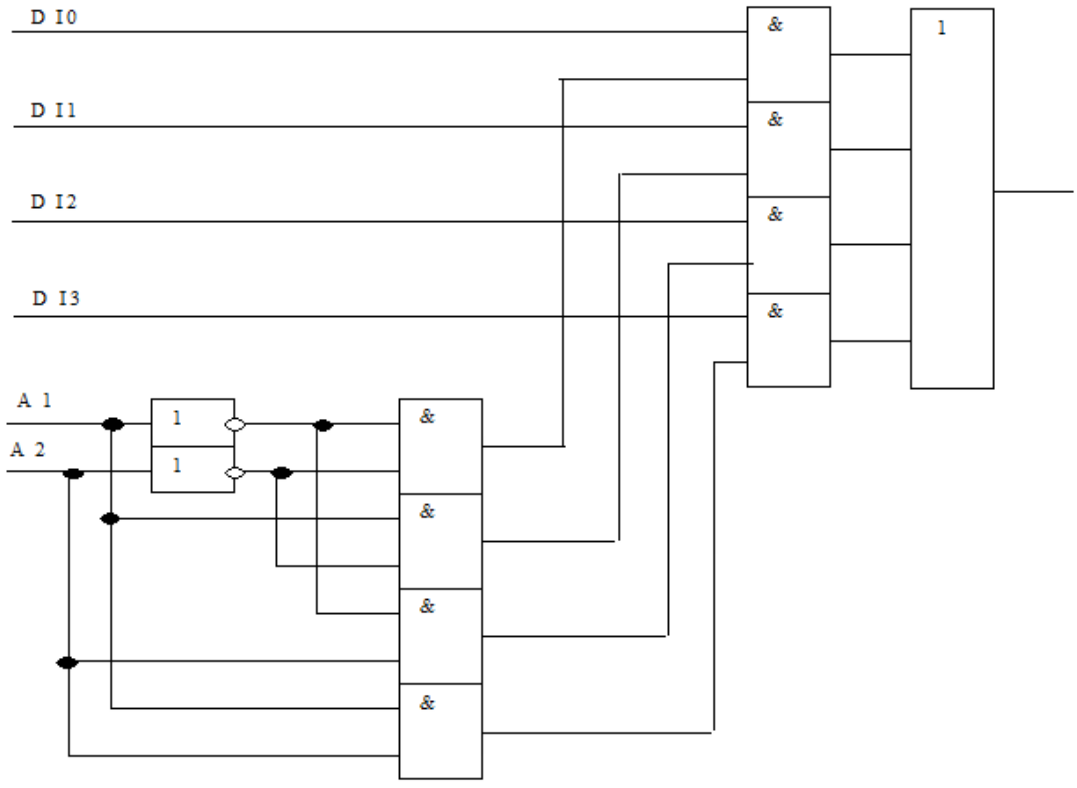
Вариант 7.



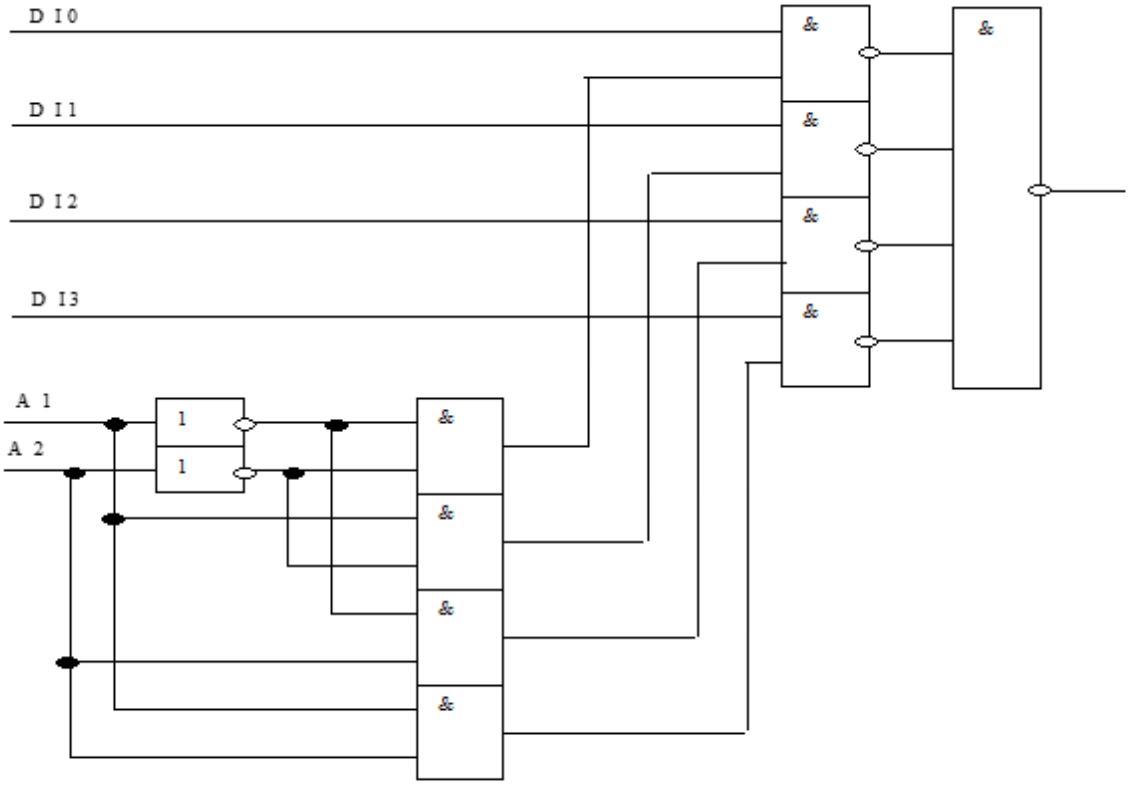
Вариант 8.



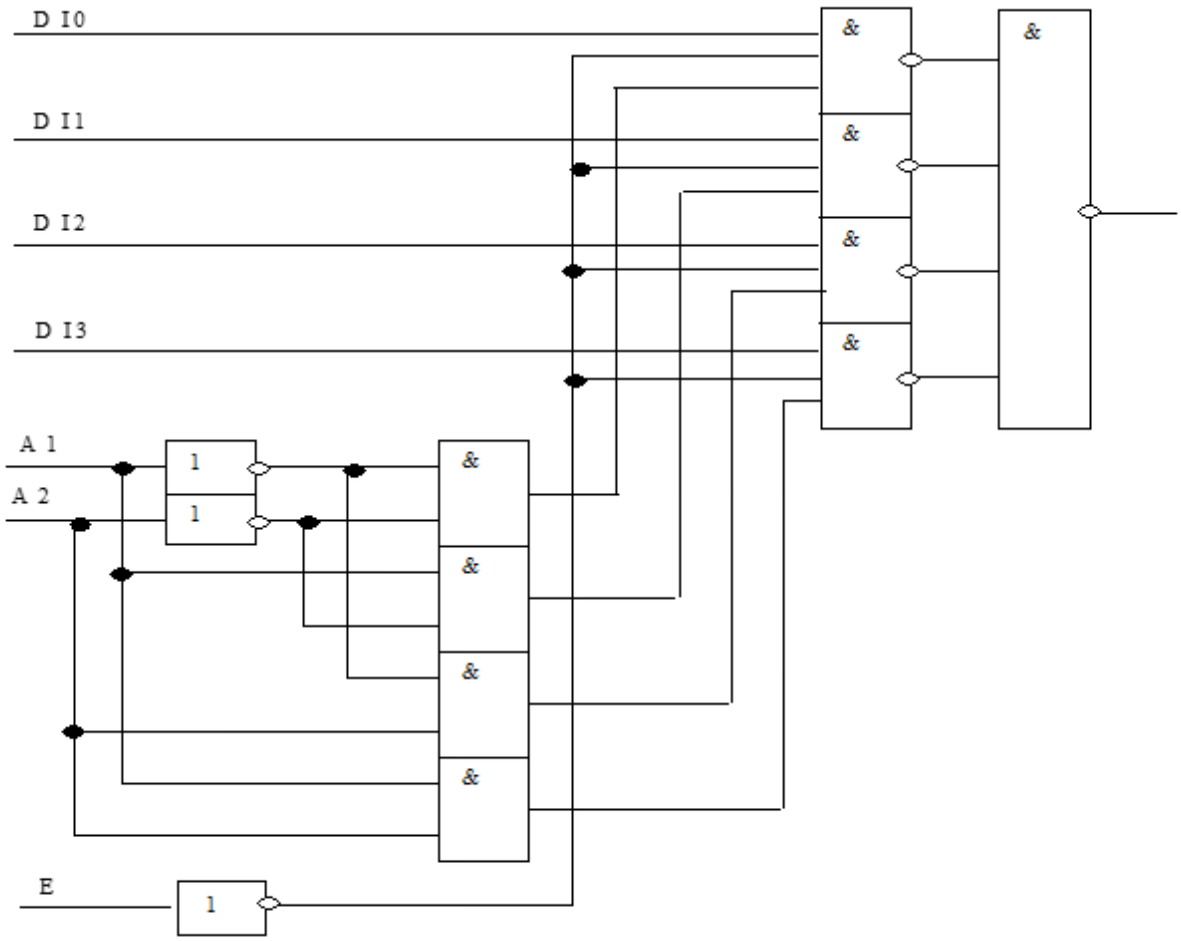
Вариант 9.



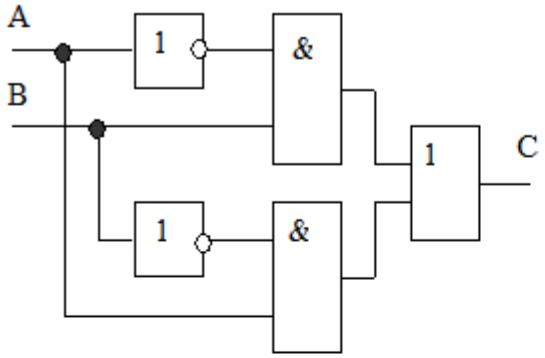
Вариант 10



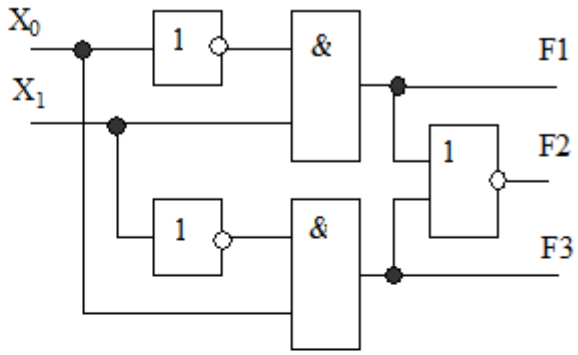
Вариант 11



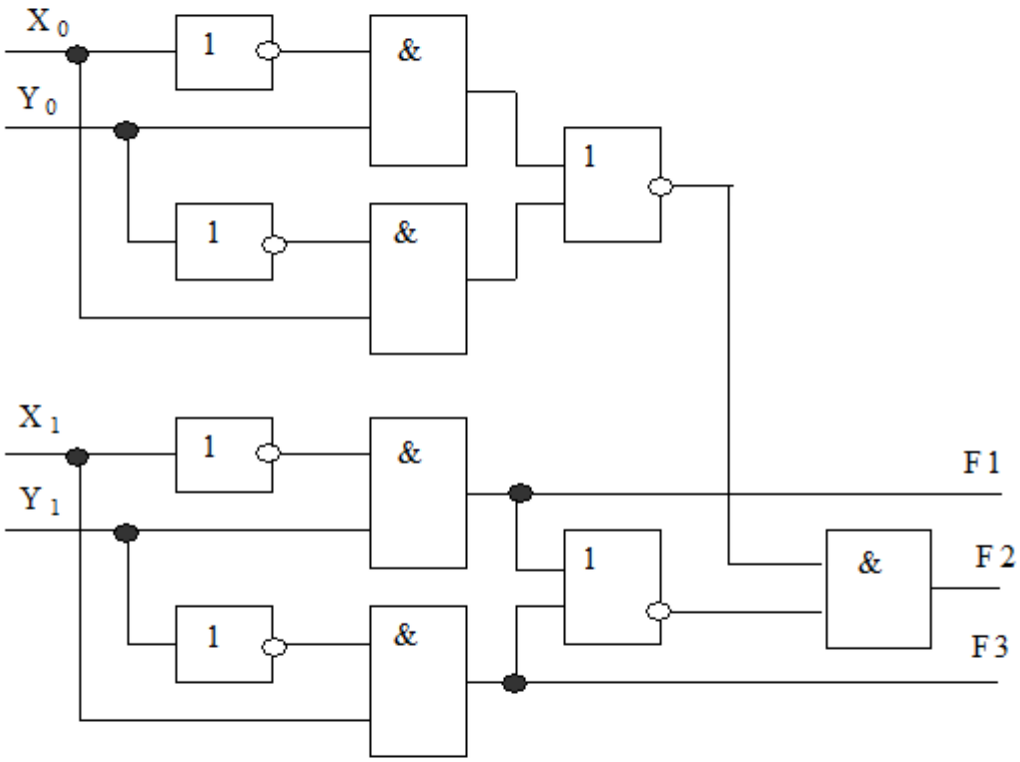
Вариант 12



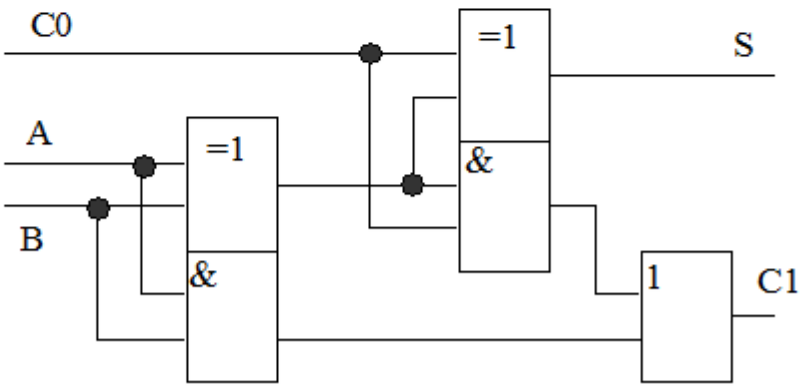
Вариант 13



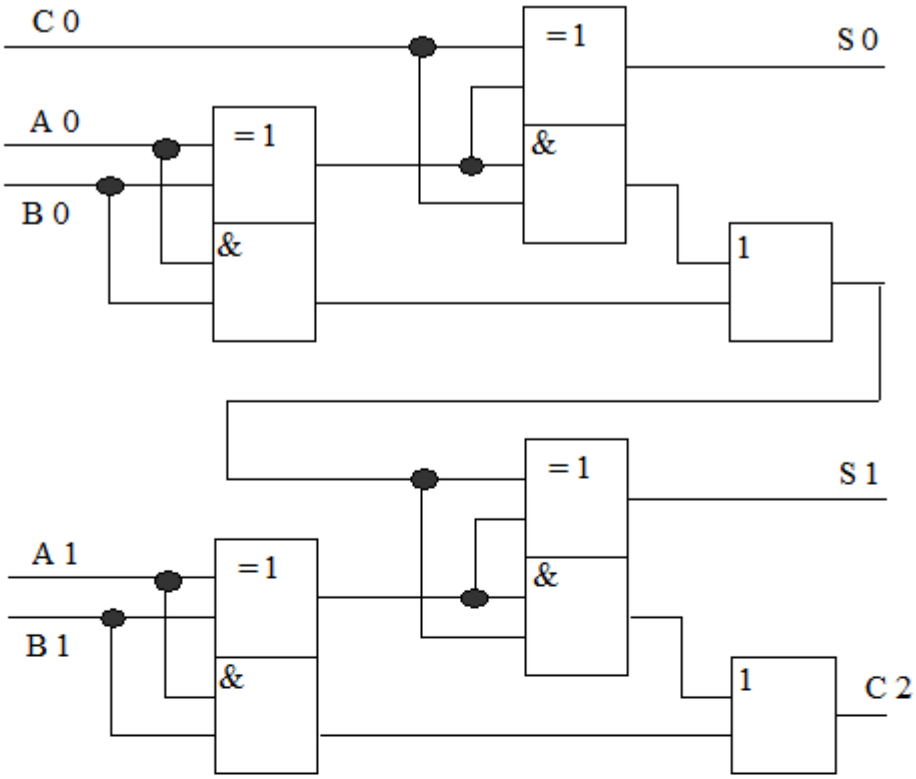
Вариант 14



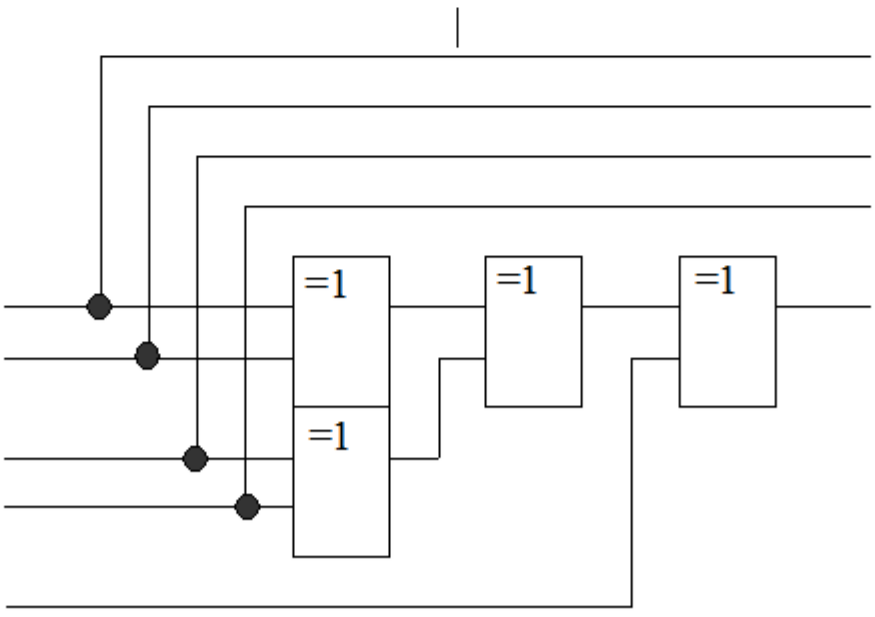
Вариант 15



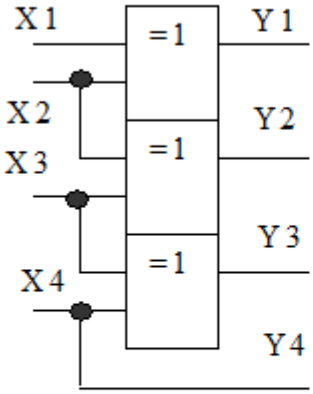
Вариант 16



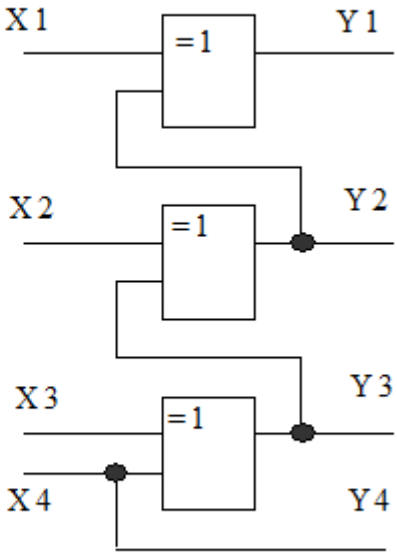
Вариант 17.



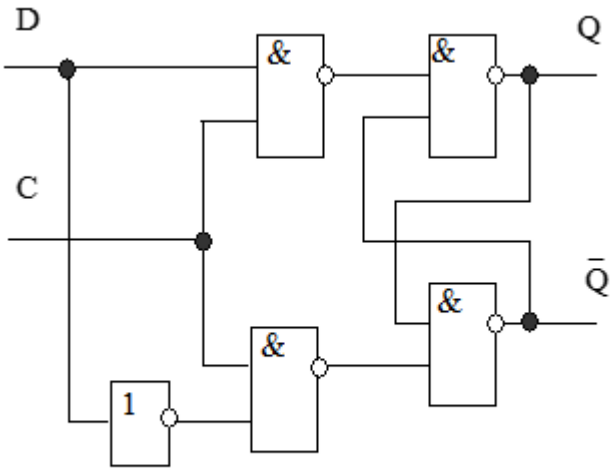
Вариант 18



Вариант 19



Вариант 20



Вариант 21

Тема 4. Логические основы цифровой техники

4.1. Задание №4

Произвести расчет функции (ПРИЛОЖЕНИЕ В) в двоичной системе счисления (число разрядов восемь).

4.2. Задание №5

Используя законы и правила алгебры логики, упростить функцию (ПРИЛОЖЕНИЕ Е).

Тема 5. Типовые последовательностные узлы. Триггеры

5.1. Задание №6

Изобразить временную диаграмму работы триггера при заданном изменении входных сигналов. Исходные данные (принципиальная и функциональная схемы триггера, и временная диаграмма входных сигналов) приведены на рис. 4.1 – 4.29 (ПРИЛОЖЕНИЕ F), а их соответствие вариантам содержится в ПРИЛОЖЕНИИ F.

Тема 6. Типовые комбинационные узлы. Мультиплексор, демультиплексор

6.1. Задание №7

Используя мультиплексоры и демультиплексоры серии К155 (ПРИЛОЖЕНИЕ G) синтезировать принципиальную схему согласно варианту.

Вариант 1. Используя 8 корпусов микросхем К155ИД3, 1 корпус микросхем К155ИД4 и необходимые логические элементы синтезировать принципиальную схему дешифратора размерностью 7*128 без стробирования.

Вариант 2. Используя 8 корпусов микросхем К155ИД3, 1 корпус микросхем К155ИД4 и необходимые логические элементы, синтезировать принципиальную схему дешифратора размерностью 7*128 со стробированием.

Вариант 3. Используя 8 корпусов микросхем К155КП1, 1 корпус микросхем К155КП5 и необходимые логические элементы, синтезировать принципиальную схему мультиплексора размерностью 128*1 со стробированием.

Вариант 4. Используя 17 корпусов микросхем К155ИД3, синтезировать принципиальную схему дешифратора размерностью 8*256 без стробирования.

Вариант 5. Используя 2 корпуса микросхем К155ИД3 и необходимые логические элементы, синтезировать принципиальную схему дешифратора размерностью 5*32 без стробирования.

Вариант 6. Используя 2 корпуса микросхемы К155ИД4 и необходимые логические элементы, синтезировать принципиальную схему дешифратора размерностью 4*16 без стробирования.

Вариант 7. Используя 17 корпусов микросхем К155ИД3, синтезировать принципиальную схему дешифратора размерностью 8*256 со стробированием.

Вариант 8. Используя 2 корпуса микросхем К155ИД3 и необходимые логические элементы, синтезировать принципиальную схему дешифратора размерностью 5*32 со стробированием.

Вариант 9. Используя 4 корпуса микросхем К155ИД3 и необходимые логические элементы, синтезировать принципиальную схему дешифратора размерностью 6*64 без стробирования.

Вариант 10. Используя 1 корпус микросхемы К155ИД4, синтезировать принципиальную схему дешифратора размерностью 3*8 со стробированием.

Вариант 11. Используя 2 корпуса микросхем К155КП1 и необходимые логические элементы, синтезировать принципиальную схему мультиплексора размерностью 32*1 без стробирования.

Вариант 12. Используя 5 корпусов микросхем К155КП2 и необходимые логические элементы, синтезировать принципиальную схему мультиплексора размерностью 32*1 без стробирования.

Вариант 13. Используя 4 корпуса микросхем К155КП5, 1 корпус микросхем К155КП2 и необходимые логические элементы, синтезировать принципиальную схему мультиплексора размерностью 32*1 без стробирования.

Вариант 14. Используя 4 корпуса микросхем К155КП7 и необходимые логические элементы, синтезировать принципиальную схему мультиплексора размерностью 32*1 без стробирования.

Вариант 15. Используя 1 корпус микросхем К155КП2 и необходимые логические элементы, синтезировать принципиальную схему мультиплексора размерностью 8*1 со стробированием.

Вариант 16. Используя 2 корпуса микросхем К155КП5 и необходимые логические элементы, синтезировать принципиальную схему мультиплексора размерностью 16*1 со стробированием.

Вариант 17. Используя 2 корпуса микросхем К155КП7 и необходимые логические элементы синтезировать принципиальную схему мультиплексора размерностью 16*1 со стробированием.

Вариант 18. Используя 8 корпусов микросхем К155КП5, 1 корпус микросхем К155КП2 и необходимые логические элементы, синтезировать принципиальную схему мультиплексора размерностью 64*1 со стробированием.

Вариант 19. Используя 9 корпусов микросхем К155КП7 и необходимые логические элементы, синтезировать принципиальную схему мультиплексора размерностью 64*1 со стробированием.

Вариант 20. Используя 17 корпусов микросхем К155КП1 и необходимые логические элементы, синтезировать принципиальную схему мультиплексора размерностью 256*1 со стробированием.

Вариант 21. Используя 1 корпус микросхемы К155ИД4 и необходимые логические элементы, синтезировать принципиальную схему дешифратора размерностью 3*8 без стробирования.

ПРИЛОЖЕНИЕ А

Таблица 3.1

№ вар	10-чное число	2-чное число	8-чное число	16-чное число
1	789,375	11010100	160	98,C
2	117	100010,01	137	48,E
3	56	10111,11	312,6	9D,5
4	101	1111110,01	110	86,101
5	56,4375	1001000,111	235,34	A2
6	95,2	10101100,1	206	75,D
7	126,125	1000110,1	126,53	38,F
8	72,75	10100010,001	165	65,2A
9	432,875	1110101,011	70	D5,12
10	134,0625	1011101,11	145	5F,3A
11	162,5	1100101,01	2651,2	7E,6
12	5,375	10000001,1	101	56, D9
13	54,375	100,011	134	B8
14	120,8	1110001,111	162	1C2
15	142,4	110000,11	326,54	1FF
16	81,41	1001001,1	72	56,83
17	45,5	1010101	416	3A2
18	420,5	11001010,11	100	2E,C
19	511,25	101110,010	325,126	99
20	127,39	1011100,01	200	101,E7
21	75,03125	1110100,0001	51,5	37

Таблица 3.2

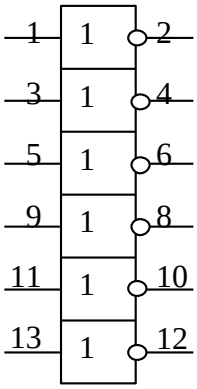
№ вар	1-е чис-ло	2-е чис-ло
1	21	45
2	32	17
3	44	62
4	-35	61
5	67	20
6	52	47
7	74	15
8	17	72
9	49	53
10	23	81
11	48	24
12	59	40
13	81	14
14	76	85
15	33	66
16	77	36
17	89	58
18	97	42
19	56	38
20	88	19
21	43	48

ПРИЛОЖЕНИЕ В

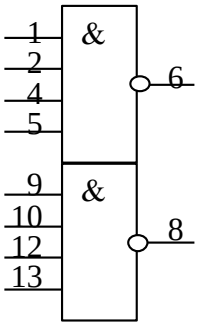
Вариант	Логическая функция F(A,B,C)	A	B	C
1	$(\overline{A} \oplus B) \wedge (\overline{A + B}) \vee (C - A)$	14	11	15
2		4	2	5
3		5	6	13
4		2	8	9
5		8	7	7
6	$\overline{((A + B) \oplus C)} \cdot (A \vee (B - C)) \wedge A$	7	11	4
7		2	11	6
8		3	14	9
9		8	10	7
10		13	15	8
11	$(\overline{\overline{A} \wedge \overline{B}}) \oplus \overline{C} + ((A \wedge B) - 1)$	13	9	4
12		4	9	7
13		5	8	11
14		12	7	8
15		6	10	9
16	$\overline{((A \wedge \overline{B}) + A)} \oplus \overline{\overline{C}} - ((\overline{A \vee B} - 1) - 2)$	6	13	9
17		5	12	8
18		7	4	11
19		3	13	5
20		4	12	9
21	$(\overline{A} \vee (B \oplus \overline{C}) \wedge (C - 1))$	10	5	4

ПРИЛОЖЕНИЕ D

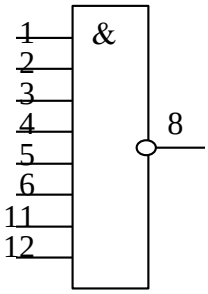
УГО ИМС серии К155



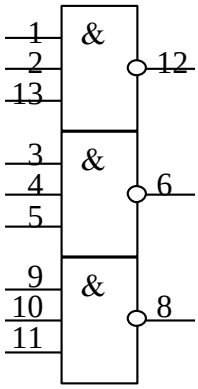
К155ЛН1



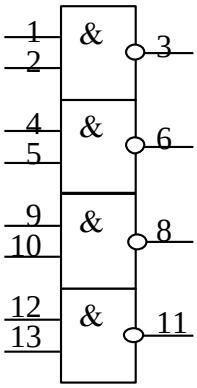
К155ЛА1



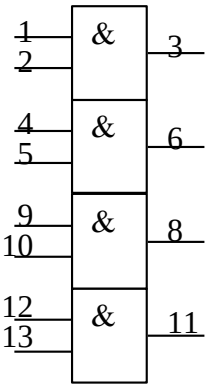
К155ЛА2



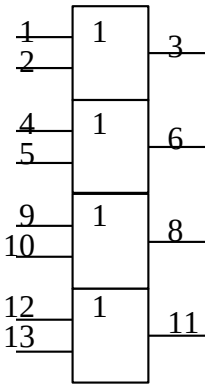
К155ЛА4



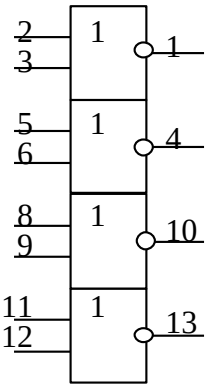
К155ЛА3



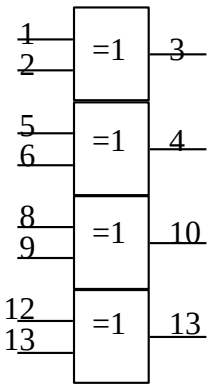
К155ЛИ1



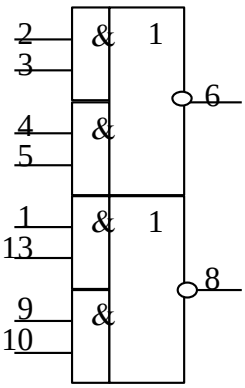
К155ЛЛ1



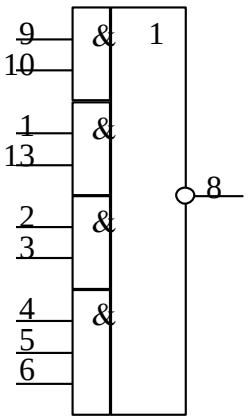
К155ЛЕ1



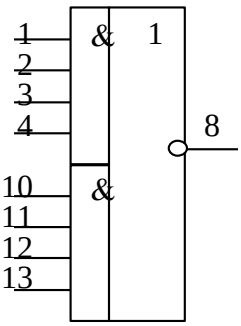
К155ЛП5



К155ЛР1



К155ЛР3



6

К155ЛР4

ПРИЛОЖЕНИЕ E

Вариант	Функция
1	$F(a,b,c)=(\overline{(\bar{a}b\bar{c}+ca\bar{b})\cdot(ab+c)}+a\bar{b})\cdot ca$
2	$F(a,b,c)=(\overline{(\bar{a}\bar{b}\bar{c}+ab\bar{c})\cdot(a+bc)}+\bar{b}\bar{c})\cdot ab$
3	$F(a,b,c)=\bar{a}\bar{b}+\bar{a}\bar{c}\cdot a\bar{c}$
4	$F(a,b,c)=\overline{((ab+\bar{c})\cdot(abc+\bar{b}c)+ca)}\cdot a\bar{b}$
5	$F(a,b,c)=\overline{c\cdot\bar{a}\cdot(b\bar{c}\bar{a}+\bar{b}c\bar{a})}\cdot\bar{a}\bar{b}$
6	$F(a,b,c)=\overline{(ab+\bar{a}c+c\bar{b})}+\overline{abc}$
7	$F(a,b,c)=\overline{(\bar{c}\bar{a}+\bar{c}\bar{b}\cdot c\bar{b})\cdot b}$
8	$F(a,b,c)=\overline{(ab\bar{c}+\bar{a}\bar{b}\bar{c})\cdot(b+a\bar{c})}+\overline{acb}$
9	$F(a,b)=\bar{a}\bar{b}+a\bar{b}+\bar{a}b$
10	$F(a,b)=\overline{\bar{a}\bar{b}+a\bar{b}+\bar{a}b}$
11	$F(a,b,c)=\overline{(ab\bar{c}+\bar{a}\bar{b}\bar{c})\cdot(b+ac)}+a\bar{c}\cdot ab$
12	$F(a,b,c)=\overline{\bar{a}\cdot\bar{b}\cdot(abc+\bar{b}\bar{c})}+\overline{ac\cdot(\bar{a}\bar{b}+\bar{c}b+a)}$
13	$F(a,b,c)=\overline{(\overline{ab\bar{c}}\cdot\overline{\bar{a}\bar{b}})\cdot(ab+\bar{c})}$
14	$F(a,b,c)=\overline{(\bar{b}+\bar{c})}\cdot\overline{(ab+c)(a\bar{b}\bar{c}+bc)}$
15	$F(a,b)=(a+\bar{b})\cdot(\bar{a}+b)\cdot(\bar{a}+\bar{b})$
16	$F(a,b)=(a+b)\cdot\overline{(\bar{a}\cdot b)}\cdot\overline{(a\cdot\bar{b})}$
17	$F(a,b,c)=\overline{(ac+b)}\cdot(\bar{a}\bar{b}\bar{c}+a\bar{c})$
18	$F(a,b,c)=(a\bar{c}+\bar{b})\cdot\overline{(\bar{a}\bar{b}\bar{c}+a\bar{c})}$
19	$F(a,b,c)=\overline{(\bar{a}+bc)\cdot(\overline{a\bar{b}}\cdot\overline{\bar{b}c})}$
20	$F(a,b,c)=(\bar{a}+\overline{\bar{b}c})\cdot(ab+b\bar{c})$
21	$F(a,b,c)=\bar{a}\bar{b}\bar{c}\cdot\overline{(ac+b)\cdot(\bar{b}+c+a)}$

ПРИЛОЖЕНИЕ F

№ вар.	Схема	Врем. диагр.	№ вар.	Схема	Врем. диагр.	№ вар.	Схема	Врем. диагр.
1	Рис. 4.6	Рис. 4.19	**8**	Рис. 4.8	Рис. 4.17	**15**	Рис. 4.3	Рис. 4.16
2	Рис. 4.8	Рис. 4.25	**9**	Рис. 4.3	Рис. 4.23	**16**	Рис. 4.12	Рис. 4.25
3	Рис. 4.4	Рис. 4.24	**10**	Рис. 4.12	Рис. 4.24	**17**	Рис. 4.4	Рис. 4.17
4	Рис. 4.9	Рис. 4.20	**11**	Рис. 4.1	Рис. 4.14	**18**	Рис. 4.11	Рис. 4.20
5	Рис. 4.6	Рис. 4.26	**12**	Рис. 4.10	Рис. 4.20	**19**	Рис. 4.1	Рис. 4.21
6	Рис. 4.2	Рис. 4.22	**13**	Рис. 4.7	Рис. 4.18	**20**	Рис. 4.5	Рис. 4.23
7	Рис. 4.12	Рис. 4.18	**14**	Рис. 4.5	Рис. 4.16	**21**	Рис. 4.7	Рис. 4.25

Продолжение **ПРИЛОЖЕНИЯ F**

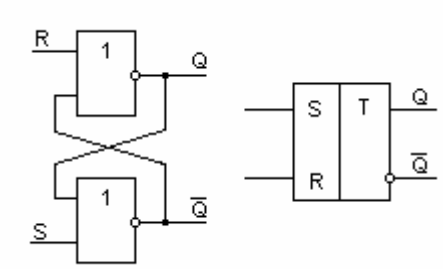


Рис. 4.1

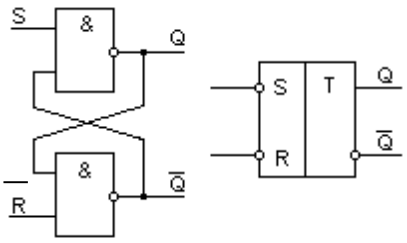


Рис. 4.2

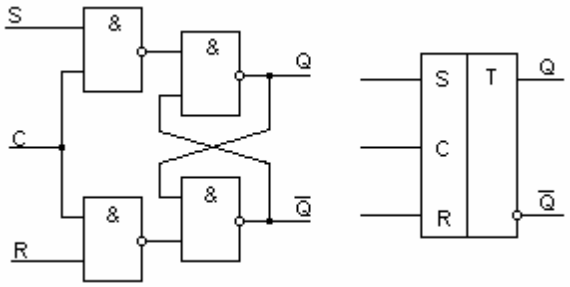


Рис. 4.3

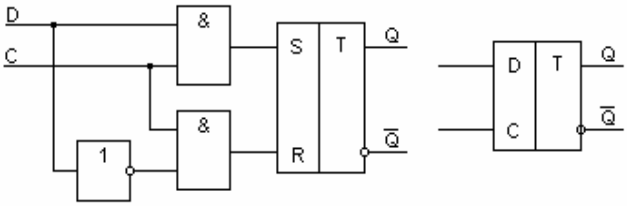


Рис. 4.4

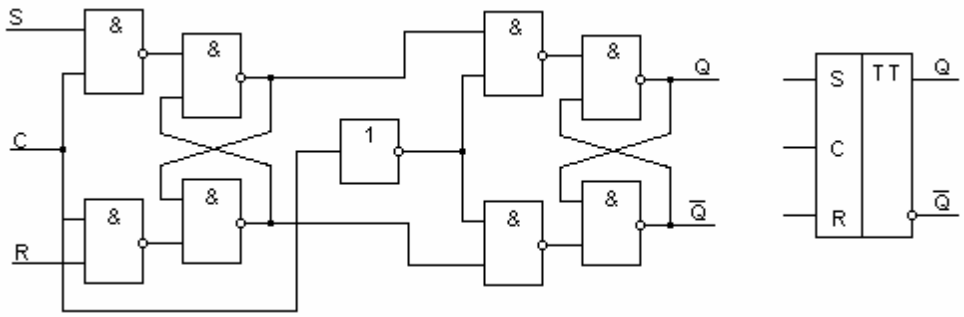


Рис. 4.5

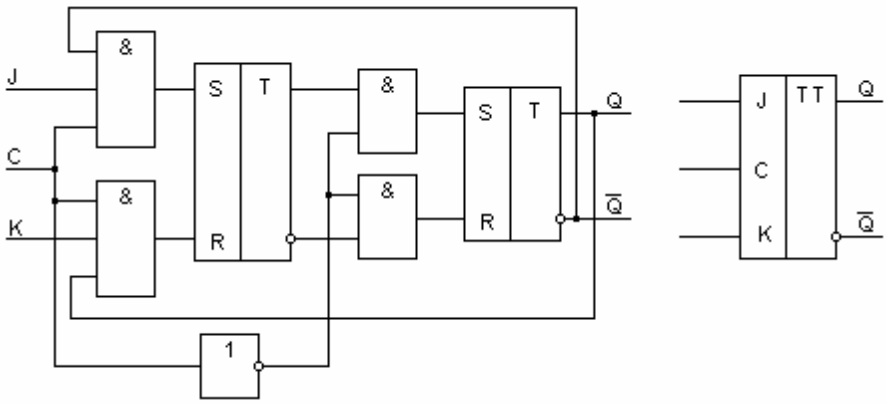


Рис. 4.6

Продолжение **ПРИЛОЖЕНИЯ F**

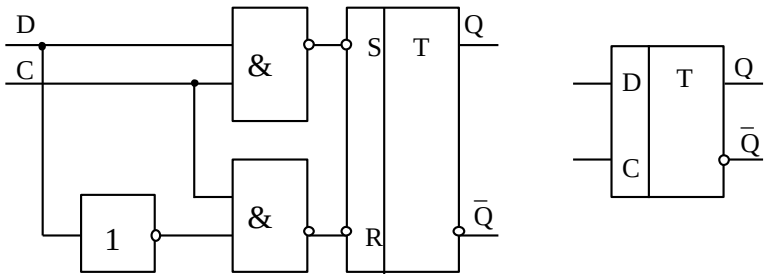


Рис. 4.7

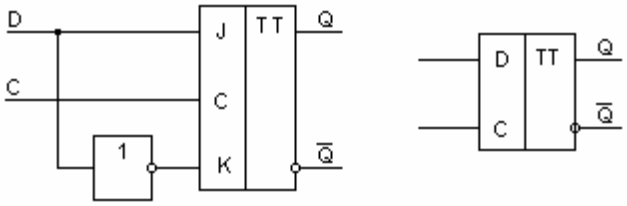


Рис. 4.8

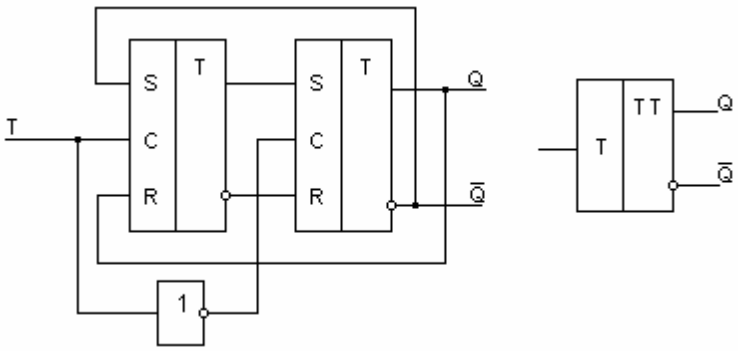


Рис. 4.9

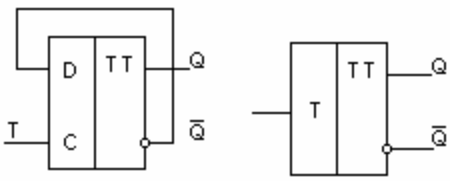


Рис 4.10

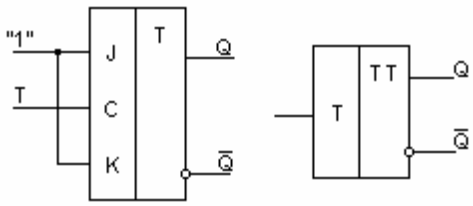


Рис. 4.11

Продолжение **ПРИЛОЖЕНИЯ F**

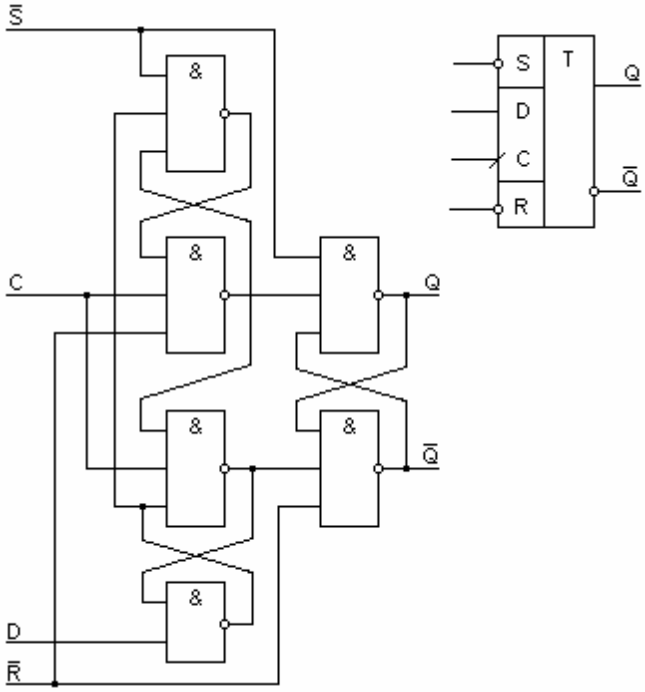


Рис. 4.12

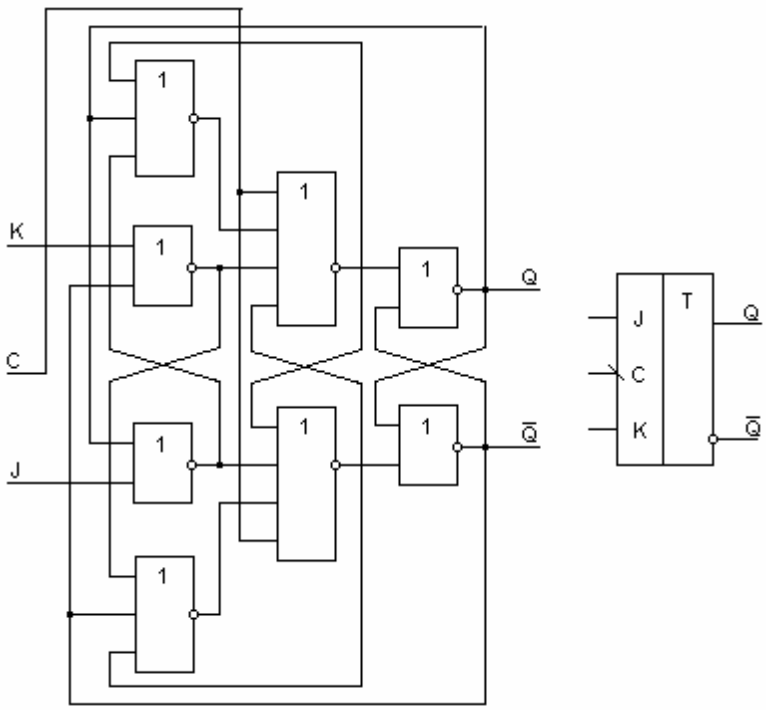


Рис. 4.13

Продолжение **ПРИЛОЖЕНИЯ F**

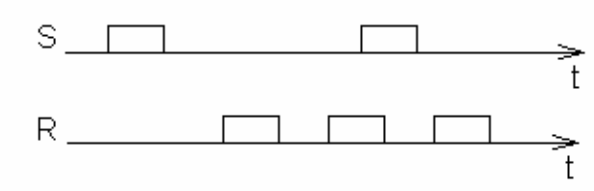


Рис. 4.14

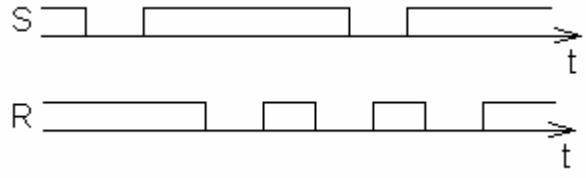


Рис. 4.15

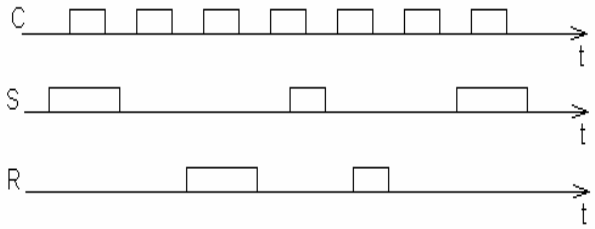


Рис. 4.16

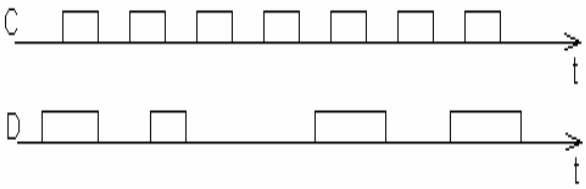


Рис. 4.17

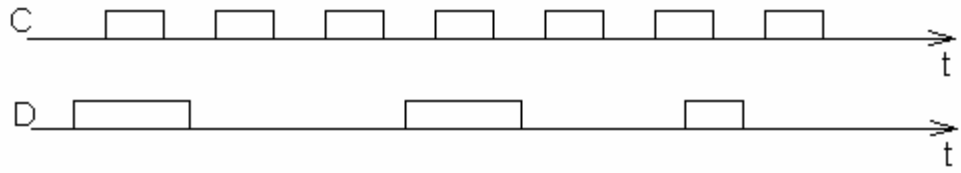


Рис. 4.18

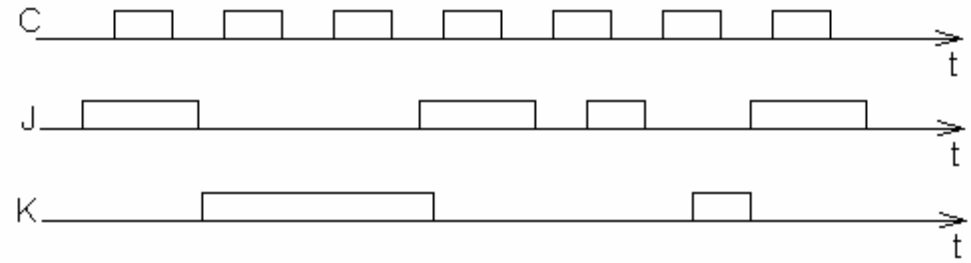


Рис. 4.19

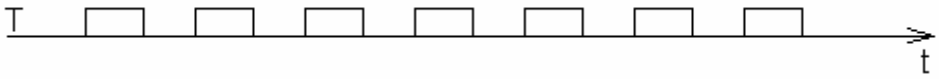


Рис. 4.20

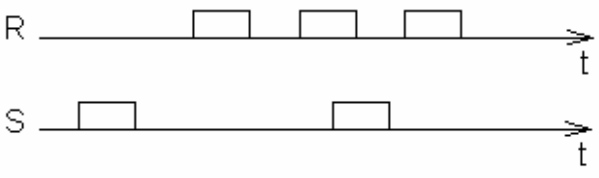


Рис. 4.21

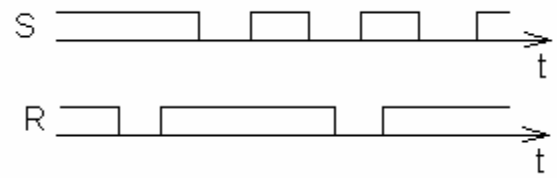


Рис. 4.22

Продолжение **ПРИЛОЖЕНИЯ F**

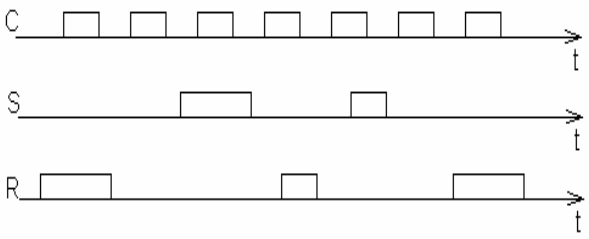


Рис. 4.23

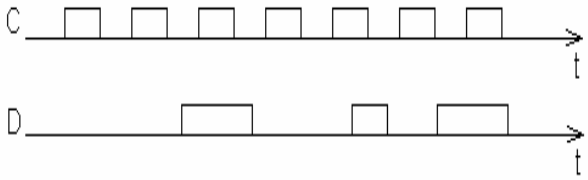


Рис. 4.24

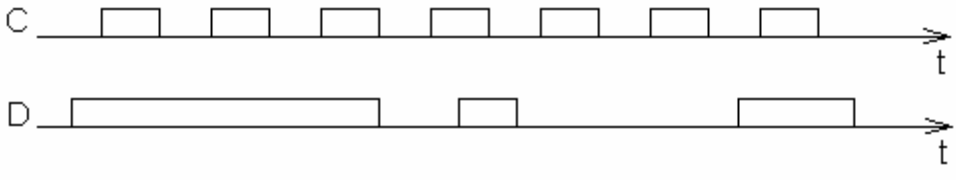


Рис. 4.25

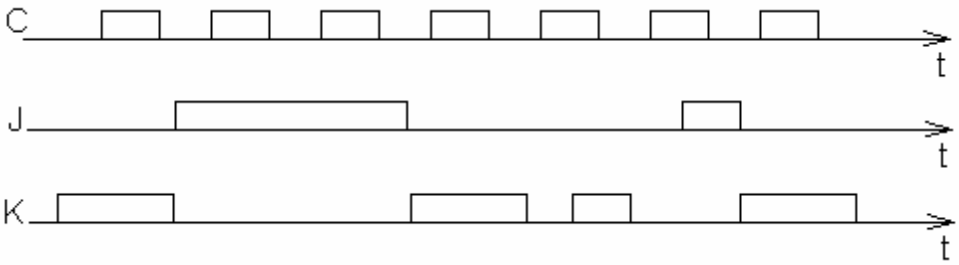


Рис. 4.26

ПРИЛОЖЕНИЕ G

УГО мультиплексоров серии К155

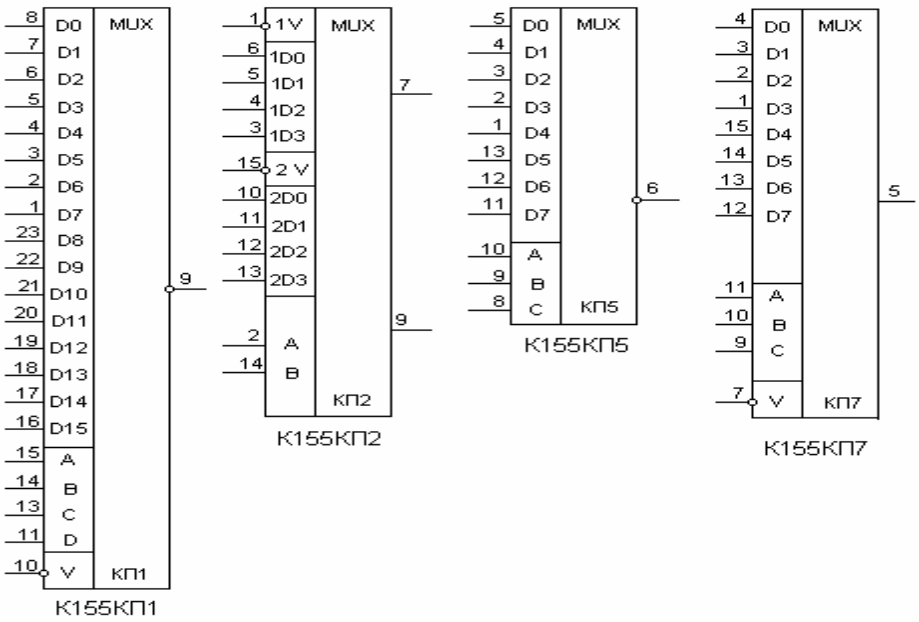


Таблица истинности К155КП1

Входы																					Вы-ход
V	D	C	B	A	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	
0	0	0	0	0	⅟₀	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0⁄1
0	0	0	0	1	X	⅟₀	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0⁄1
0	0	0	1	0	X	X	⅟₀	X	X	X	X	X	X	X	X	X	X	X	X	X	0⁄1
0	0	0	1	1	X	X	X	⅟₀	X	X	X	X	X	X	X	X	X	X	X	X	0⁄1
0	0	1	0	0	X	X	X	X	⅟₀	X	X	X	X	X	X	X	X	X	X	X	0⁄1
		-	-	-					-	-	-						-	-	-		
0	1	0	1	1	X	X	X	X	X	X	X	X	X	X	X	⅟₀	X	X	X	X	0⁄1
0	1	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X	⅟₀	X	X	X	0⁄1
0	1	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	⅟₀	X	X	0⁄1
0	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	⅟₀	X	0⁄1
0	1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	⅟₀	0⁄1
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1

Таблица истинности К155КП5

Входы											Вы-ход
C	B	A	D0	D1	D2	D3	D4	D5	D6	D7	
0	0	0	⅟₀	X	X	X	X	X	X	X	0⁄1
0	0	1	X	⅟₀	X	X	X	X	X	X	0⁄1
0	1	0	X	X	⅟₀	X	X	X	X	X	0⁄1
0	1	1	X	X	X	⅟₀	X	X	X	X	0⁄1
1	0	0	X	X	X	X	⅟₀	X	X	X	0⁄1
1	0	1	X	X	X	X	X	⅟₀	X	X	0⁄1
1	1	0	X	X	X	X	X	X	⅟₀	X	0⁄1
1	1	1	X	X	X	X	X	X	X	⅟₀	0⁄1

Таблица истинности К155КП2

Входы							Вы-ходы
V	B	A	D0	D1	D2	D3	
0	0	0	⅟₀	X	X	X	⅟₀
0	0	1	X	⅟₀	X	X	⅟₀
0	1	0	X	X	⅟₀	X	⅟₀
0	1	1	X	X	X	⅟₀	⅟₀
1	X	X	X	X	X	X	0

Таблица истинности К155КП7

Входы											Выходы	
V	C	B	A	D0	D1	D2	D3	D4	D5	D6	D7	5	6
0	0	0	0	⅟₀	X	X	X	X	X	X	X	⅟₀	0⁄1
0	0	0	1	X	⅟₀	X	X	X	X	X	X	⅟₀	0⁄1
0	0	1	0	X	X	⅟₀	X	X	X	X	X	⅟₀	0⁄1
0	0	1	1	X	X	X	⅟₀	X	X	X	X	⅟₀	0⁄1
0	1	0	0	X	X	X	X	⅟₀	X	X	X	⅟₀	0⁄1
0	1	0	1	X	X	X	X	X	⅟₀	X	X	⅟₀	0⁄1
0	1	1	0	X	X	X	X	X	X	⅟₀	X	⅟₀	0⁄1
0	1	1	1	X	X	X	X	X	X	X	⅟₀	⅟₀	0⁄1
1	X	X	X	X	X	X	X	X	X	X	X	0	1

Продолжение **ПРИЛОЖЕНИЯ G**

УГО ИМС серии К155

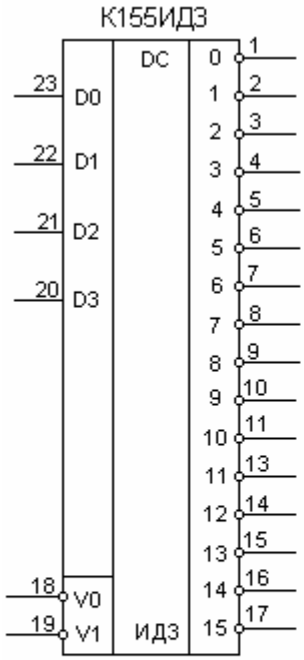


Таблица истинности К155ИД3

Входы						Выходы															
V0	V1	D3	D2	D1	D0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

К155ИД4

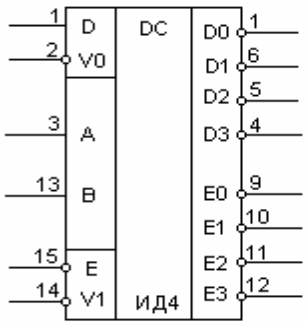


Таблица истинности К155ИД4

Адр. вх.		Секция D						Секция E					
		Вх.		Вых.				Вх.		Вых.			
B	A	D	V1	D0	D1	D2	D3	E	V2	E0	E1	E2	E3
0	0	1	0	0	1	1	1	0	0	0	1	1	1
1	0	1	0	1	0	1	1	0	0	1	0	1	1
0	1	1	0	1	1	0	1	0	0	1	1	0	1
1	1	1	0	1	1	1	0	0	0	1	1	1	0
X	X	0	X	1	1	1	1	1	X	1	1	1	1
X	X	X	1	1	1	1	1	X	1	1	1	1	1

ЛИТЕРАТУРА

Основная

1. Браммер Ю.А. Импульсные и цифровые устройства: Учеб. для студентов электрорадиоприборостроительных средн. спец. Учеб. заведений. / Ю.А.Браммер, И.М.Пащук. – 6-е изд., перераб. и доп. – М.: Высш. шк., 2002

2. Калиш Г.Г. Основы вычислительной техники. Учеб. пособ. для средн. проф. уч. заведений. – М.: Высш. шк., 2000.

3. Цифровая и вычислительная техника: Учебник для вузов»/ Э.В. Евреинов и др.; Под ред. Э.В. Евреинова.- М.: Радио и связь, 1991

4. Напрасник М.В. Микропроцессоры и микроЭВМ- М.: Высш. шк., 1989

5. Калабеков Б.А., Мамзелев И.А. Цифровые устройства и МПС: Учебник для техникумов связи.- М.: Радио и связь

6. Браммер Ю.А., Пащук И.Н. Цифровые устройства: Учеб.пособие для вузов.- М.: Высш.шк.,2004г.

Дополнительная

7. .Полупроводниковые БИС запоминающих устройств: Справочник/ В.В.Баранов, Н.В.Бекин, А.Ю.Гордонов и др.; Под ред. А.Ю.Гордонова, Ю.Н.Дъякова. - .- М.: Радио и связь, 1986.

8. Аванесян Г.Р., Левшин В.П. Интегральные микросхемы ТТЛ, ТТЛШ : Справочник. - .- М.: Машиностроение, 1993.

9. Лебедев О.Н. и др. Изделия электронной техники. Цифровые микросхемы. Микросхемы памяти. ЦАП и АЦП. : Справочник. - .- М.: Радио и связь, 1994.

10. БИС запоминающих устройств.: Справочник/ под ред. А.Ю.Гордонова. - .- М.: Радио и связь, 1990.