Титульный лист методических рекомендаций и указаний, методических рекомендаций, методических указаний

Форма
Ф СО ПГУ 7.18.3/40

Министерство образования и науки Республики Казахстан

Павлодарский государственный университет им. С. Торайгырова

Кафедра Вычислительная техника и программирование

МЕТОДИЧЕСКИЕ УКАЗАНИЯ И РЕКОМЕНДАЦИИ

к лабораторным работам

по дисциплине Вычислительные системы и сети

для студентов специальности 050702 Автоматизация и управление

Павлодар

Лист утверждения методических рекомендаций и указаний, методических рекомендаций, методических указаний

Форма
Ф СО ПГУ 7.18.3/41

УТВЕРЖДАЮ
Проректор по УР
__________ Пфейфер Н.Э.
(подпись) (Ф.И.О.)
«___»___________201_г.

Составитель: ст. преподаватель _____________Балгабаева Г.С.

Кафедра Вычислительная техника и программирование

Методические указания и рекомендации

к лабораторным работам

по дисциплине Вычислительные системы и сети

для студентов специальности 050702 Автоматизация и управление

Рекомендовано на заседании кафедры
«_____»______________201__г., протокол №__

Заведующий кафедрой __________ Потапенко О.Г. «___» _______201__г
(подпись) (Ф.И.О.)

Одобрено УМС Физики, математики и информационных технологий
(наименование факультета)
«____»______________201__г., протокол №____

Председатель УМС ___________ Муканова Ж.Г. «___» _______201__г
(подпись) (Ф.И.О.)

ОДОБРЕНО ОПиМОУП:
Начальник ОПиМОУП ___________ Варакута А.А. «___» _______201__г
(подпись) (Ф.И.О.)

Одобрена учебно-методическим советом университета
«_____»______________201_г. Протокол №____

Лабораторная работа №1

Тема: Системы счисления.

Цель: Изучить системы счисления и перевод чисел из одной системы счисления в другую.

1. Теоретические сведения

Системой счисления называется совокупность правил записи чисел.

Системы счисления подразделяются на **позиционные** и **непозиционные**. Как позиционные, так и непозиционные системы счисления используют определенный набор символов — **цифр**, последовательное сочетание которых образует число. Непозиционные системы счисления характеризуются тем, что в них символы, обозначающие то или иное число, не меняют своего значения в зависимости от местоположения в записи этого числа.

В позиционной системе счисления количество символов в наборе равно **основанию** системы счисления. Место каждой цифры в числе называется **позицией**. Номер позиции символа (за вычетом единицы) в числе называется **разрядом**. Разряд 0 называется **младшим** разрядом. Каждой цифре соответствует определенный количественный эквивалент. Введем обозначение — запись $A_{(p)}$ будет означать количественный эквивалент числа A, состоящего из **n** цифр $\mathbf{a_k}$ (где k - 0,..., n-1) в системе счисления с основанием **p**. Это число можно представить в виде последовательности цифр:

$$\mathbf{A = a_{n-1}a_{n-2} \ldots a_1a_0}.$$

При этом всегда выполняется неравенство $\mathbf{a_k < p}$.

В общем случае количественный эквивалент некоторого положительного числа A в позиционной системе счисления можно представить выражением:

$$\mathbf{A_{(p)} = a_{n-1} * p^{n-1} + a_{n-2} * p^{n-2} + \ldots + a_1 * p^1 + a_0 * p^0}, \qquad (1)$$

где: **p** — основание системы счисления (некоторое целое положительное число);

a — цифра данной системы счисления; **n** — номер старшего разряда числа.

Для получения количественного эквивалента числа в некоторой позиционной системе счисления необходимо сложить произведения количественных значений цифр на степени основания, показатели которых равны номерам разрядов (обратите внимание, что нумерация разрядов начинается с нуля).

1.1 Двоичная система счисления

Набор цифр для двоичной системы счисления:

{0, 1}, основание степени p = 2.

Количественный эквивалент некоторого целого n-значного двоичного числа вычисляется согласно формуле (1):

$$A_{(2)} = a_{n-1} *2^{n-1} + a_{n-2} *2^{n-2} + \ldots + a_1 *2^{1} + a_0 *2^{0} \qquad (2)$$

Наличие этой системы счисления обусловлено тем, что компьютер построен на логических схемах, имеющих в своем элементарном виде только два состояния — включено и выключено. Производить счет в двоичной системе просто для компьютера, но сложно для человека. Например, рассмотрим двоичное число 10100111.

Вычислим количественный эквивалент этого двоичного числа. Согласно формуле (2), это будет величина, равная следующей сумме:

$1*2^7 + 0*2^6 + 1*2^5 + 0*2^4 + 0*2^3 + 1*2^2 + 1*2' + 1*2^{\circ}$.

Сложение и вычитание двоичных чисел (рисунок 1) выполняется так же, как и для других позиционных систем счисления, например десятичной. Точно так же выполняются заем и перенос единицы из (в) старший разряд. Например:

```
 11  11111   перенос      1 1  1      заем
  110011011             _11010010011
+ 110010101              00111011011
 ----------             ------------
 1100110000              10010111000
```

Рисунок 1 - Сложение и вычитание двоичных чисел

Степени двойки приведем в таблице 1, а соответствие двоичных чисел и их десятичных и шестнадцатеричных эквивалентов в таблице 2.

Таблица 1 - Степени двойки

k	2^k
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096

1.2 Шестнадцатеричная система счисления

Данная система счисления имеет следующий набор цифр:

{0, 1, 2, ..., 9, A, B, C, D, E, F}, основание системы p = 16.

Количественный эквивалент некоторого целого n-значного шестнадцатеричного числа вычисляется согласно формуле (1):

$$\mathbf{A_{(16)} = a_{n-1} * 16^{n-1} + a_{n-2} * 16^{n-2} + \ldots + a_1 * 16^1 + a_0 * 16^0}$$

Например, количественный эквивалент шестнадцатеричного числа f45ed23c равен:

$15*16^7+4*16^6+5*16^5+14*16^4+13*16^3+2*16^2+3*16'+12*16^{\circ}$.

Таблица 2 - Шестнадцатеричные цифры

Десятичное число	Двоичная тетрада	Шестнадцатеричное число
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A, a
И	1011	B, b
12	1100	C, c
13	1101	D, d
14	1110	E, e
15	1111	F,f
16	10000	10

Шестнадцатеричная система счисления при производстве вычислений несколько сложнее, чем двоичная, в частности, в том, что касается правил переносов в старшие разряды (заемов из старших разрядов). Главное здесь — помнить следующее равенство:

$(1 + F = 10)_{16}$

Эти переходы очень важны при выполнении сложения и вычитания шестнадцатеричных чисел. Пример приведен на рисунке 2:

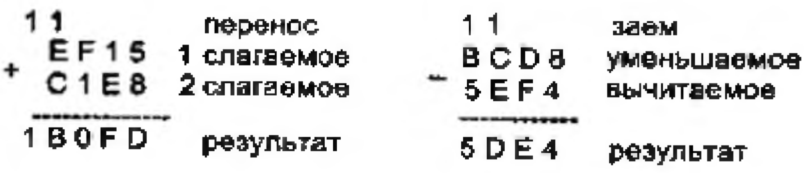


Рисунок 2 – Сложение и вычитание шестнадцатеричных чисел

1.3 Десятичная система счисления

Это наиболее известная система счисления, так как она постоянно используется нами в повседневной жизни. Данная система счисления имеет следующий набор цифр:

{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, основание степени p = 10.

Количественный эквивалент некоторого целого n-значного десятичного числа вычисляется согласно формуле (1):

$$\mathbf{A_{(10)} = a_{n-1} * 10^{n-1} + a_{n-2} * 10^{n-2} + \ldots + a_1 * 10^1 + a_0 * 10^0}$$

Например, значение числа $A_{(10)}$ - 4523 равно: $4*10^3+5*10^2+ 2*10^1 +3*10^0$.

1.4 Перевод чисел из одной системы счисления в другую

1.4.1 Перевод в десятичную систему счисления

Этот тип перевода наиболее прост. Обычно его производят с помощью так называемого **алгоритма замещения**, суть которого заключается в следующем: сначала в десятичную систему счисления переводится основание степени **p**, а затем — цифры исходного числа. Результаты подставляются в формулу (1). Полученная сумма и будет искомым результатом.

1.4.2 Перевод в двоичную систему счисления

Перевод из десятичной системы счисления

1. Разделить десятичное число **A** на 2. Запомнить частное **q** и остаток **a**.
2. Если в результате шага 1 частное **q** не равно 0, то принять его за новое делимое и отметить остаток **a**, который будет очередной значащей цифрой числа, вернуться к шагу 1, на котором в качестве делимого (десятичного числа) участвует полученное на шаге 2 частное.
3. Если в результате шага 1 частное **q** = 0, алгоритм прекращается. Выписать остатки в порядке, обратном их получению. Получится двоичный эквивалент исходного числа.

Например, требуется перевести число 247_{10} в двоичную систему счисления (рисунок 3):

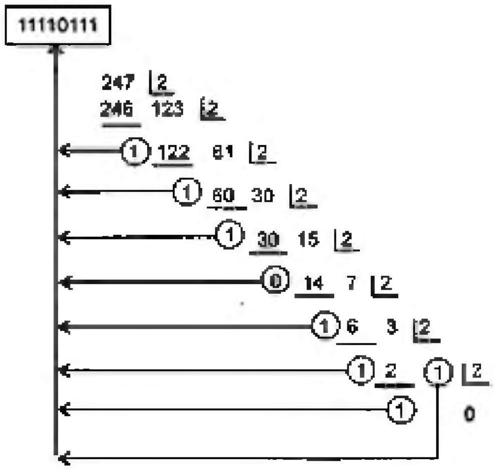


Рисунок 3 – Перевод в двоичную систему счисления

Порядок обхода остатков для получения результата показан стрелками, и результат преобразования равен 11110111_2.

1.4.3 Перевод из шестнадцатеричной системы счисления

Суть этого перехода заключается в последовательной замене шестнадцатеричных цифр соответствующими двоичными тетрадами согласно таблице 2.

Например, число $e4d5_{16}$ - $1110\ 0100\ 1101\ 0101_2$.

1.4.4 Перевод в шестнадцатеричную систему счисления

1.4.4.1 Перевод из десятичной системы счисления

Алгоритм перевода в двоичную систему счисления из десятичной.

1. Разделить десятичное число А на 16. Запомнить частное **q** и остаток **a**.
2. Если в результате шага 1 частное **q** не равно 0, то принять его за новое делимое, записать остаток и вернуться к шагу 1.
3. Если частное **q** = 0, прекратить работу алгоритма. Выписать остатки в порядке, обратном их получению. Получится шестнадцатеричный эквивалент исходного десятичного числа.

Например, требуется преобразовать $32\ 767_{10}$ в шестнадцатеричную систему счисления (рисунок 4).

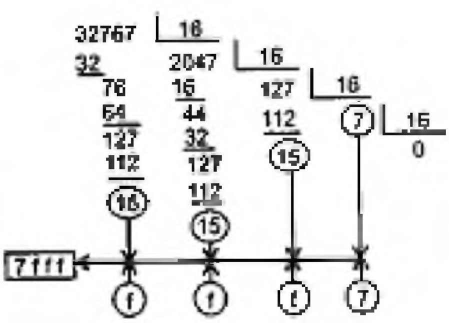


Рисунок 4 – Перевод в шестнадцатеричную систему счисления

Порядок обхода остатков для получения результата показан на рис. 4 стрелками. Результат преобразования равен $7fff_{16}$.

1.4.4.2 Перевод из двоичной системы счисления

Идея алгоритма аналогична идее перевода из двоичной системы счисления в шестнадцатеричную. Суть в том, что двоичное число разбивается на тетрады, начиная с младшего разряда. Далее каждая тетрада приводится к соответствующему шестнадцатеричному числу согласно табл. 2.

Например, требуется перевести число

$111001011010111101011000110110001111010101011101_2$

в шестнадцатеричную систему счисления.

Разобьем его на тетрады:

0111 0010 1101 0111 1010 1100 0110 1100 0111 1010 1010 1101.

По тетрадам приводим последовательности нулей и единиц к шестнадцатеричному представлению: 7 2 d 7 a c 6 c 7 a a d.

В итоге мы получили результат преобразования:

$111001011010111101011000110110001111010101011101_2$ - $72d7ac6c7aad_{16}$.

1.4.5 Перевод дробных чисел

Рассмотрим наиболее часто используемыми на практике способами перевода дробных чисел. Для этого формулу (1) преобразуем к следующему виду:

$\mathbf{A_{(p)} = a_{n-1} * p^{n-1} + a_{n-2} * p^{n-2} + \ldots + a_1 * p^1 + a_0 * p^0 + a_{-1} * p^{-1} + a_{-2} * p^{-2} + \ldots + a_{-m} * p^{-m}}$ (3)

Рассмотрим операции перевода чисел на примерах.

Пример 1:

Перевести в десятичное представление дробь в двоичной системе счисления $110100{,}01001011_2$.

Для перевода используем формулу (3):

$110100{,}01001011_2 = 1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 0*2^0 + 0*2^{-1} + 1*2^{-2} + 0*2^{-3} + 0*2^{-4} + 1*2^{-5} +$
$+ 0*2^{-6} + 1*2^{-7} + 1*2^{-8}$.

Вычислить целую часть десятичной дроби $1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 0*2' + 0*2^\circ$ для вас труда не составит. Для вычисления остальной части выражения удобно использовать таблицу 3.

Таблица 3 - Значения отрицательных степеней по основанию числа 2

m	2^m
1	0,5

2	0,25
3	0,125
4	0,0625
5	0,03125
6	0,015625
7	0,0078125 и т. д.

Подсчитаем результат перевода значения $110100,01001011_2$ в десятичное представление.

Пример 2:

Перевести в десятичное представление дробь в шестнадцатеричной системе счисления $1df2,a1e4_{16}$.

Вновь используем формулу (3) и получим:

$1df2,a1e4_{16} = 1*16^3 + 13*16^2 + 15*2^1+2*16^0 + 10*16^{-1}+1*16^{-2} +14*16^{-3} + 4*16^{-4}$

Значения отрицательных степеней числа 16 приведены в табл. 4.

Таблица 4 - Значения отрицательных степеней по основанию числа 16

m	16^m
1	0,0625
2	0,00390625
3	0,000244140625
4	0,0000152587890625
5	0,00000095367431640625
6	0,000000059604644775390625
7	0,0000000037252902984619140625

Перевод из двоичной системы счисления в шестнадцатеричную систему и обратно выполняется на основе тетрад.

Рассмотрим представление десятичных дробей в двоичной и шестнадцатеричной системах счисления.

Общий алгоритм перевода десятичной дроби в другую систему счисления можно представить следующей последовательностью шагов:

1. Выделить целую часть десятичной дроби и выполнить ее перевод в выбранную систему счисления по алгоритмам, рассмотренным выше.
2. Выделить дробную часть и умножить ее на основание выбранной новой системы счисления.
3. В полученной после умножения дробной части десятичной дроби выделить целую часть и принять ее в качестве значения первого после запятой разряда числа в новой системе счисления.
4. Если дробная часть значения, полученного после умножения, равна нулю, то прекратить процесс перевода. Процесс перевода можно прекратить

также в случае, если достигнута необходимая точность вычисления. В противном случае перейти к шагу 3.

Пример.3:

Перевести в двоичную систему счисления десятичную дробь $108{,}406_{10}$

1. Переведем целую часть десятичной дроби 108.406_{10} в двоичную систему счисления (рисунок 5).

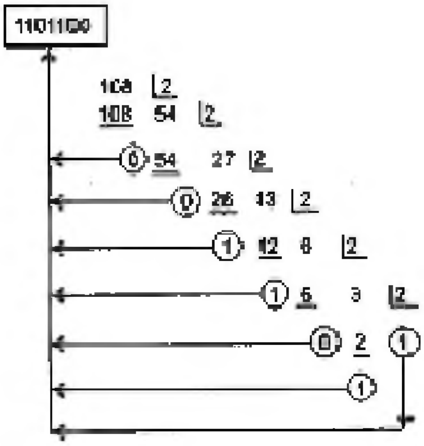


Рисунок 5 – Перевод целой части десятичного числа $108{,}406_{10}$ в двоичную систему счисления

2. Переведем дробную часть десятичного числа $108{,}406_{10}$ по алгоритму, приведенному выше (рисунок 6).

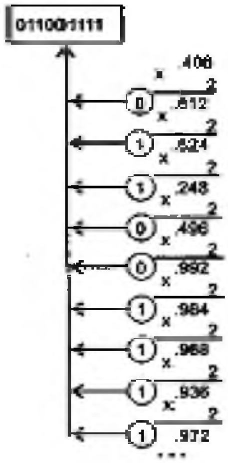


Рисунок 6 – Перевод дробной части числа108.406_{10} в двоичную систему счисления

Результат перевода следующий: $108{,}406_{10}$=1101100,011001111.

При переводе дробного числа из десятичной системы счисления в шестнадцатеричную, целесообразно предварительно перевести число в двоичную систему. Затем двоичное представление разбить на тетрады отдельно до запятой и после запятой. Разбиение на тетрады двоичных разрядов целой части производят от запятой в сторону старших разрядов. Неполную старшую тетраду дополняют слева ведущими нулями. Разряды дробной части, напротив, разбивают на тетрады от запятой вправо к младшим разрядам. Если последняя тетрада неполная, то ее дополняют справа нулями. На рис. 7 показан процесс перевода дробного десятичного числа (пример 3) в эквивалентное шестнадцатеричное представление.

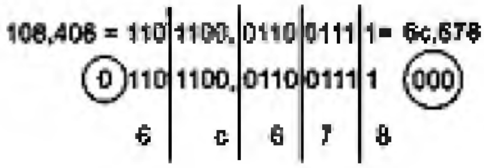


Рисунок 7 – Перевод десятичного числа в шестнадцатеричную систему счисления

1.5 Числа со знаком

Положительные целые числа со знаком — это 0 и все положительные числа.

Отрицательные целые числа со знаком — это все числа, меньшие 0. Отличительным признаком числа со знаком является особая трактовка старшего бита поля, представляющего число. В качестве поля могут выступать байт, слово или двойное слово. Физически этот бит ничем не отличается от других, — все зависит от команды, работающей с данным полем. Если в ее алгоритме заложена работа с целыми числами со знаком, то она будет по-особому трактовать старший бит поля. В случае если бит равен 0, число считается положительным и его значение вычисляется по правилам, которые мы рассмотрели выше. В случае если этот бит равен 1, число считается отрицательным и предполагается, что оно записано в так называемом **дополнительном коде**

Дополнительный код некоторого отрицательного числа представляет собой результат инвертирования (замены 1 на 0 и наоборот) каждого бита двоичного числа, равного модулю исходного отрицательного числа плюс единица. Например, рассмотрим десятичное число -185_{10}. Модуль данного числа в двоичном представлении равен 10111001_2. Сначала нужно дополнить это значение слева нулями до нужной размерности — байта, слова и т. д. В нашем случае дополнить нужно до слова, так как диапазон представления знаковых чисел в байте составляет -128.. .127. Следующее действие — получить **двоичное дополнение**. Для этого все разряды двоичного числа нужно инвертировать:

0000000010111001_2 - 11111111010001102

Теперь прибавляем единицу:

$1111111101000110_2 + 0000000000000001_2 - 1111111101000110_2$.

Результат преобразования равен 1111111101000111_2.

Именно так представляется число -185_{10} в компьютере.

При работе с числами со знаком необходимо уметь выполнять обратное действие - имея двоичное дополнение числа, определять значение его модуля. Для этого необходимо выполнить два действия:

1. Выполнить инвертирование битов двоичного дополнения.
2. К полученному двоичному числу прибавить двоичную единицу.

Например, определим модуль двоичного представления числа

$-185_{10} = 1111111101000111_2$ - инвертируем биты - 0000000010111000_2.

Добавляем двоичную единицу:

$0000000010111000_2 + 0000000000000001_2$ - 0000000010111001_2 - |-185|.

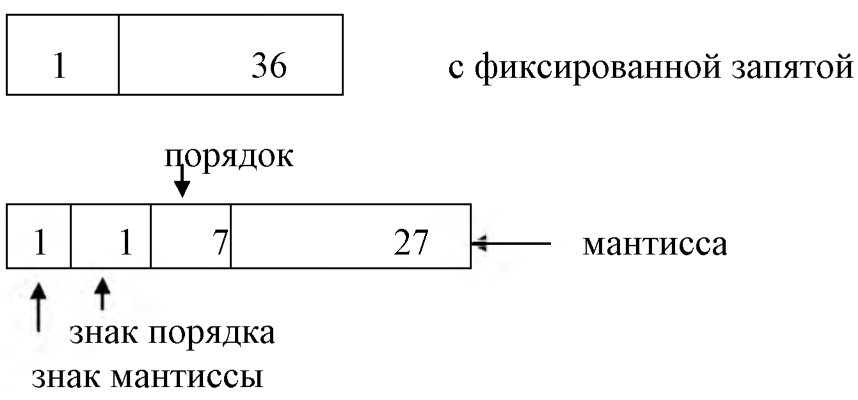
1.6 Числа с плавающей запятой

При записи в форме с плавающей запятой число делится на две части, называемые мантиссой (цифровая часть) и порядком (по некоторому основанию). В десятичной системе счисления, например, число 15 можно записать следующими способами:

Мантисса	***Порядок***
0,15	$*10^2$
1,5	10^1
15,0	10^0
150,0	10^{-1}
1500,0	10^{-2}

Т.к. ЭВМ работает с двоичной информацией, то и мантиссу, и порядок представляют как двоичные числа. Т.к. мантисса, и порядок м.б. положительными, или отрицательными числами, то два разряда отводятся для знаков представления чисел с фиксированной и плавающей запятой для 36 разрядного слова:

Знак число



2 Содержание отчёта

Отчет по лабораторной работе должен содержать:

1. Название лабораторной работы
2. Цель
3. Задание к работе;
4. Результаты расчётов;
5. Вывод.

3 Задание

Упражнение №1

Выполнить сложение чисел

Таблица 5 - Задание

№ варианта	Двоичные числа	Шестнадцатеричные числа
1	1111+101+1000= 11111+1011+10101=	ED45C+4F56= 32C+AF12=
2	100011+1101= 1011011+1011+10001=	1C4D+24F= 23DF+EF15=
3	110011001+1100001= 1010+110001+1011=	24CA+5B3A= 7B3F+1CFD=
4	10110100+1110011= 11101000+1100+111=	7B3F+5B3A= 1C4D+EF15=
5	101011+101101 11011011+11001101+11011=	ED45C+AF12= 24CA+24CA=
6	1001001+101= 111111+111111+111111=	1B0FD+C1E8= BCD8+5DE4=
7	1011011+111= 1000001+1000001+1000001=	ACD6+F5C7= EF15+24CA=
8	11010001+101010= 100010001+111+10101=	F5C7+1C4D= 9CFD+6F3F=
9	11101101+1110110= 1011+1001001+111101=	EF15+6DA7= 3EF9+ECFA=

Упражнение №2

Найдите обратный и дополнительный коды двоичных чисел, указанных в таблице 6:

Таблица 6 - Задание

№ варианта	Двоичные числа
1	011100110010

2	010111011111
3	000000000001
4	111111111111
5	111111111110
6	000000000111
7	100000000000
8	100000000001
9	000000000000
10	000100100100

Упражнение №3

Перемножьте двоичные числа из упражнения №1

Упражнение №4

Перевести в двоичную и шестнадцатеричную системы счисления десятичные дроби с точностью вычислений $e=10^6$.

Таблица 7 - Задание

№ варианта	Десятичные дроби
1	108,406 ; 54,26 ;103,54
2	96,102 ; 301,123 ; 231,563
3	210,3201 ; 432,521 ; 36,231
4	78,561 ; 69,204 ; 67,621
5	105,402 ; 104,627 ; 55,236
6	76,123 ; 123,701 ; 305,58
7	203,103 ; 100,256 ;203,156
8	235,201 ; 56,36 ; 105,78
9	301,56 ; 201,35 ; 54,126
10	236,56 ; 512,65 ;128,34

Упражнение №5

Выполнить деление в двоичной системе счисления

Таблица 7 - Задание

№ варианта	Десятичные числа
1	32:4=8 ; 18:9=2
2	25:5=5 ; 15:3=5
3	24:6=4 ; 28:2=14
4	14:7=2 ; 9:3=3
5	48:12=4 ; 52:2=26
6	27:3=9 ; 12:4=3

7	64:2=32 ; 35:5=7
8	34:2=17 ; 60:3=20
9	26:13=2 ; 42:7=6
10	48:6=8 ; 39:3=13

Контрольные вопросы:

1. Какие системы счисления называются позиционными?
2. Алгоритм перевода десятичной дроби в шестнадцатеричную систему счисления.
3. Как образуются обратный и дополнительный коды числа?
4. Что такое мантисса числа?

Лабораторная работа №2

Тема: Логические функции и формы их представления.

Цель: Изучить логические функции, формы их представления, научиться составлять таблицы истинности логических функций.

1 Теоретические сведения

В ЭВМ информация подвергается не только арифметической, но и логической обработке. Основу работы логически схем и устройств ЭВМ составляет специальный математический аппарат, называемый **алгеброй логики** или **исчислением высказываний**. При этом под высказыванием понимается любое утверждение, о котором можно сказать, что оно истинно или ложно. В логике высказываний интересуются не содержанием высказываний, а только их истинностью или ложностью; никакие другие признаки высказываний в алгебре логики не рассматриваются. Одно и то же высказывание не может быть одновременно истинным и ложным или не истинным и не ложным.

Если высказывание истинно, то считают, что его значение равно единице; если высказывание ложно, то считают, что его значение равно нулю. Таким образом, значения высказываний можно рассматривать как переменную величину, принимающую только два дискретных значения: 0 или 1. Это приводит к полному соответствию между логическими высказываниями математической логикой и двоичными цифрами в двоичной системе счисления, что позволяет описывать работу логических схем ЭВМ, проводить их анализ и синтез с помощью математического аппарата алгебры логики.

Всякое устройство ЭВМ, выполняющее арифметические или логические операции, можно рассматривать как функциональный преобразователь, входными переменными (аргументами) которого являются исходные двоичные числа, а выходной функцией от них — новое двоичное число, образованное в результате выполнения данной операции. При этом как входные переменные, так и выходные функции могут принимать лишь одно из двух возможных значений: 0 или 1.

В каждом конкретном случае количество входных переменных бывает различным. В простейшем случае это одна переменная $\mathbf{x}$, принимающая значение 0 или 1. В общем случае таких переменных может быть $\mathbf{n}$, т. е. $\mathbf{x_1, x_2, ..., x_n}$. Так как каждая переменная $\mathbf{xi}$ при этом равна 0 или 1, то для $\mathbf{n}$ переменных образуется множество разнообразных сочетаний или наборов входных переменных.

В алгебре логики строго доказывается, что для $\mathbf{n}$ переменных количество различных наборов равно 2^n. Так, для одной переменной $\mathbf{x}$ существует только два набора: (0) или (1), так как $2^1 = 2$; для двух переменных $\mathbf{x_1, x_2}$ — четыре различных набора: (0,0) (0,1), (1,0), (1,1), так как $2^2 = 4$; для трех переменных $\mathbf{x_1, x_2, x_3}$ — восемь различных наборов, так как $2^3 = 8$, и т. д.

Функция **f(x_1, x_2, ..., x_n)**, определяемая на наборах входных двоичных переменных **x_1, x_2, ..., x_n** и принимающая в качестве возможных значений 0 или 1 называется **логической функцией** или **функцией алгебры логики**

Значения функций для каждого набора входных переменных можно указать с помощью таблицы. Например, для четырех наборов переменных **x_1, x_2** в табл. 1.1 приведены значения 16 различных логических функций **f_1, f_2, ...,f_{16}**. Такими таблицами удобно описывать функционирование различных логических элементов и узлов ЭВМ. Эти таблицы также называют **таблицами истинности.**

Рассмотрим различные логические функции **f_1** двух переменных, представленные в табл. 1.1, не в порядке их нумерации, а в той последовательности, которая позволит выявить их общие и наиболее характерные свойства.

Таблица 1.1

Аргумент		**Функция**															
x_1	x_2	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	I	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

1. Функция **f_9 (x_1, x_2)** — конъюнкция (логическое умножение, или операция И) переменных **x_1, x_2**, которая обращается в единицу только в том случае, если аргументы **x_1** и **x_2** одновременно равны единице, и в нуль — во всех остальных случаях, т. е. если хотя бы один аргумент равен нулю. Иными словами, функция конъюнкции равна **min (x_1,x_2).** Обозначается конъюнкция знаком Λ, который читается как И. Значение конъюнкции для заданных аргументов находится по правилам логического умножения:

$$
\begin{array}{ll}
0 \Lambda 0 = 0 & \qquad x \Lambda 0 = 0 \\
1 \Lambda 1 = 0 & \qquad x \Lambda 1 = x \\
1 \Lambda 0 = 0 & \qquad x \Lambda x = x \\
1 \Lambda 1 = 1 & \qquad x \Lambda x = 0
\end{array}
$$

где $\bar{x}$ обозначает отрицание или инверсию переменной **x**, т. е. $\bar{x}$ =0, если **x** = 1, и $\bar{x}$ = 1, если x = 0.

В общем случае функцию конъюнкции можно определить для любого числа аргументов, т. е. $x_1 \Lambda x_2 \Lambda x_3$ .... Знак Λ может быть опущен или заменен точкой, т. е. **$x_1x_2x_3$** ... или **$x_1 \bullet x_2 \bullet x_3$**

2. Функция $\mathbf{f_8(x_1,x_2)}$ – отрицание конъюнкции (операция И-НЕ), т. е. $\overline{x1 \wedge x2}$. Данная функция обращается в нуль только в том случае, если аргументы $\mathbf{x_1}$ и $\mathbf{x_2}$ одновременно равны единице, и в единицу, если хотя бы один из аргументов равен нулю.

3. Функция $\mathbf{f_{15}(x_1, x_2)}$ – дизъюнкция (логическое сложение, или операция ИЛИ) переменных $\mathbf{x_1}$ и $\mathbf{x_2}$ которая обращается в нуль только в том случае, если аргументы $\mathbf{x_1}$ и $\mathbf{x_2}$ одновременно равны нулю, и в единицу, если хотя бы один аргумент равен единице. Иными словами, функция дизъюнкции равна max $\mathbf{(x_1, x_2)}$. Обозначается дизъюнкция знаком V, который читается как ИЛИ. Значение дизъюнкции для заданных аргументов находится по правилам логического сложения:

0 V 0 = 0	x V 0 = x
0 V 1 =1	x V 1 = 1
1 V 0 = 1	x V x = x
1 V 1 = 1	x V x = 1

В общем случае функцию дизъюнкции можно определить для любого числа аргументов, т. е. $\mathbf{x_1 \ V \ x_2 \ V \ x_3 \ V \ ...}$

4. Функция $\mathbf{f_2\ (x_1,x_2)}$ – отрицание дизъюнкции (операция ИЛИ-НЕ), т.е. $\overline{x1 \vee x2}$. Данная функции обращается в единицу только в том случае, если аргументы $\mathbf{x_1}$ и $\mathbf{x_2}$ одновременно равны нулю, во всех остальных случаях она равна нулю.

5. Функция $\mathbf{f_{10}\ (x_1,x_2)}$ — эквивалентность (или равнозначность) переменных $\mathbf{x_1}$ и $\mathbf{x_2}$. Данная функция обращается в единицу в том случае, если совпадают значения аргументов $\mathbf{x_1}$ и $\mathbf{x_2}$; в остальных случаях она равна нулю. Обозначается эквивалентность знаком ~, который читается «равнозначно».

6. Функция $\mathbf{f_7\ (x_1,x_2)}$ — отрицание эквивалентности (или неравнозначность) переменных $\mathbf{x_1}$и $\mathbf{x_2}$. Запись $\mathbf{x_1 \sim x_2}$ читается как «$\mathbf{x_1}$ неравнозначно $\mathbf{x_2}$». Можно убедиться в том, что значение функции неравнозначности получается поразрядным сложением двоичных переменных $\mathbf{x_1}$ и $\mathbf{x_2}$ по модулю 2, т. е. без учета переносов в старший разряд.

7. Функция $\mathbf{f_{12}\ (x_1,x_2)}$ — импликация $\mathbf{x_1}$ в $\mathbf{x_2}$, которая обращается в нуль только в том случае, если переменная $\mathbf{x_1}$ равна единице, а $\mathbf{x_2}$ — нулю; в остальных случаях функция импликации $\mathbf{x_1}$ в $\mathbf{x_2}$ равна единице. Данная функция обозначается $x1 \to x2$ и читается как «если $\mathbf{x_1}$, то $\mathbf{x_2}$».

8. Функция $\mathbf{f_{14}\ (x_1,x_2)}$ — импликация $\mathbf{x_2}$ в $\mathbf{x_1}$, которая обращается в нуль только в том случае, если переменная $\mathbf{x_2}$ равна единице, а $\mathbf{x_1}$ — нулю; в остальных случаях функция импликации $\mathbf{x_2}$ в $\mathbf{x_1}$ равна единице. Данная функция обозначается $x2 \to x1$ и читается как «если $\mathbf{x_2}$, то $\mathbf{x_1}$».

9. Функция $\mathbf{f_{14}}$ $\mathbf{(x_1,x_2)}$ — отрицание импликации $\mathbf{x_1}$ в $\mathbf{x_2}$, т. е. $\overline{x1 \to x2}$. Данную функцию можно рассматривать как функцию запрета со стороны входной переменной $\mathbf{x_2}$. Это означает, что выходная функция обращается в нуль в том случае, если входная переменная $\mathbf{x_2}$ равна единице; в остальных случаях она повторяет переменную $\mathbf{x_1}$.

10. Функция $\mathbf{f_3}$ $\mathbf{(x_1,x_2)}$ —отрицание импликации $\mathbf{x_2}$ в $\mathbf{x_1}$, т. е. $\overline{x2 \to x1}$. Данную функцию можно рассматривать как функцию запрета со стороны входной переменной $\mathbf{x_1}$. Это означает, что выходная функция обращается в нуль в том случае, если входная переменная $\mathbf{x_1}$ равна единице; в остальных случаях она повторяет переменную $\mathbf{x_2}$.

Оставшиеся шесть логических функций $\mathbf{f_1}$, $\mathbf{f_4}$ $\mathbf{f_6}$ $\mathbf{f_{11}}$ $\mathbf{f_{13}}$ $\mathbf{f_{16}}$ являются или константами, или функциями, зависящими только от одной из переменных $\mathbf{x_1}$, $\mathbf{x_2}$. Так, функции $\mathbf{f_1}$ и $\mathbf{f_{16}}$ являются соответственно тождественно равными нулю и единице:

$\mathbf{f_1}$ $\mathbf{(x_1,x_2)} \equiv 0$; $\mathbf{f_{16}}$ $\mathbf{(x_1,x_2)} \equiv 1$.

Функции $\mathbf{f_{11}}$ и $\mathbf{f_{13}}$ повторяют соответственно переменные $\mathbf{x_2}$ и $\mathbf{x_1}$:

$\mathbf{f_{11}}$ $\mathbf{(x_1,x_2)} = x_2$, $\mathbf{f_{13}}$ $\mathbf{(x_1,x_2)} = x_1$

Функции $\mathbf{f_4}$и $\mathbf{f_6}$ являются соответственно отрицаниями переменных $\mathbf{x_1}$ и $\mathbf{x_2}$:

$\mathbf{f_4}$ $\mathbf{(x_1,x_2)} = \overline{x1}$, $\mathbf{f_6}$ $\mathbf{(x_1,x_2)} = \overline{x2}$.

Рассмотренное табличное представление логических функций существенно усложняется с увеличением числа аргументов, например для трех аргументов будет 2^2=256 логических функций. В этом случае более удобно аналитическое представление логических функций в виде соответствующих формул.

При составлении для выражений таблиц истинности необходимо соблюдать следующий приоритет операций:

1 отрицание
2 выражение в скобках
3 конъюнкция
4 дизъюнкция
5 импликация
6 равнозначность

2 Задание

Упражнение№1

Составить таблицы истинности для следующих выражений:

№ варианта	Выражения
1.	$(\overline{x_1} \vee x_2) \to (x_1 \sim \overline{x_2}) \wedge \overline{x_1}$ $(\overline{x_2} \to x_1) \sim (x_1 \vee \overline{x_2}) \to x_1$
2.	$(x_1 \sim \overline{x_2}) \to (\overline{x_2} \to x_1) \vee x_2$ $(x_2 \to \overline{x_1}) \sim (x_1 \wedge \overline{x_2}) \to x_2$
3.	$(x_1 \vee \overline{x_2}) \to (\overline{x_1} \sim x_2) \to \overline{x_2}$ $(\overline{x_2} \wedge x_1) \sim (x_1 \to \overline{x_2}) \vee \overline{x_1}$
4.	$(\overline{x_1} \sim x_2) \to (\overline{x_2} \sim \overline{x_1}) \vee x_2$ $(x_2 \vee \overline{x_1}) \to (\overline{x_2} \to x_1) \sim \overline{x_1}$
5.	$(x_1 \wedge \overline{x_2}) \sim (x_2 \to \overline{x_1}) \vee \overline{x_2}$ $(x_2 \wedge \overline{x_1}) \to (\overline{x_2} \to x_1) \sim \overline{x_1}$
6.	$x_1 \to x_2 \vee x_1 \to \overline{x_2} \sim x_1$ $x_2 \to [(x_1 \sim \overline{x_2}) \wedge \overline{x_1}]$
7.	$\overline{x_1} \to \overline{x_2} \vee \overline{x_1} \to \overline{x_2} \sim \overline{x_1}$ $x_2 \to \overline{x_1} \vee x_1 \sim \overline{x_2} \to x_1$
8.	$x_1 \to x_2 \vee x_1 \to \overline{x_2} \sim x_1$ $x_1 \to (x_1 \vee \overline{x_1} \to \overline{x_2} \sim x_1)$

№ варианта	Выражения
9.	$(x_1 \to x_2 \vee x_1) \to \overline{x}_1 \sim x_2$ $x_2 \to \overline{x}_2 \vee x_1 \sim \overline{x}_2 \to x_2$
10.	$x_2 \to \left[\left(\overline{x}_1 \vee \overline{x_2}\right) \wedge \overline{x_1}\right]$ $\overline{x}_1 \wedge \overline{x}_2 \vee \overline{x}_1 \sim \overline{x}_2 \to \overline{x}_2$

Упражнение№2

С помощью таблиц истинности проверьте, равносильны ли следующие выражения:

№ варианта	Выражение
1.	$(x_1 \to x_2 \vee x_1) \to \overline{x}_1 \sim x_2$ $x_2 \to \left[\left(\overline{x}_1 \vee \overline{x_2}\right) \wedge \overline{x_1}\right]$
2.	$\overline{x}_1 \wedge \overline{x}_2 \vee \overline{x}_1 \sim \overline{x}_2 \to \overline{x}_2$ $x_2 \to \overline{x}_2 \vee x_1 \sim \overline{x}_2 \to x_2$
3.	$\overline{x}_1 \wedge \overline{x}_2 \vee \overline{x}_1 \sim \overline{x}_2 \to \overline{x}_2$ $x_2 \to \overline{x}_2 \vee x_1 \sim \overline{x}_2 \to x_2$
4.	$x_2 \to \overline{x}_2 \vee x_1 \sim \overline{x}_2 \to x_2$ $x_2 \to \left[\left(x_1 \sim \overline{x_2}\right) \wedge \overline{x_1}\right]$
5.	$x_1 \to x_2 \vee x_1 \to \overline{x}_2 \sim x_1$ $\left(x_1 \wedge \overline{x}_2\right) \sim \left(x_2 \to \overline{x_1}\right) \vee \overline{x_2}$
6.	$\left(\overline{x}_1 \sim x_2\right) \to \left(\overline{x}_2 \sim \overline{x_1}\right) \vee x_2$

№ варианта	Выражение
	$(\overline{x_2} \wedge x_1) \sim (x_1 \to \overline{x_2}) \vee \overline{x_1}$
7.	$(\overline{x_1} \vee x_2) \to (x_1 \sim \overline{x_2}) \wedge \overline{x_1}$ $(x_1 \wedge \overline{x_2}) \sim (x_2 \to \overline{x_1}) \vee \overline{x_2}$
8.	$(\overline{x_2} \to x_1) \sim (x_1 \vee \overline{x_2}) \to x_1$ $(x_1 \sim \overline{x_2}) \to (\overline{x_2} \to x_1) \vee x_2$
9.	$(x_2 \to \overline{x_1}) \sim (x_1 \wedge \overline{x_2}) \to x_2$ $(x_1 \vee \overline{x_2}) \to (\overline{x_1} \sim x_2) \to \overline{x_2}$
10.	$(x_1 \vee \overline{x_2}) \to (\overline{x_1} \sim x_2) \to \overline{x_2}$ $(\overline{x_2} \to x_1) \sim (x_1 \vee \overline{x_2}) \to x_1$

Контрольные вопросы:

1 Охарактеризуйте логические функции конъюнкции и дизъюнкции , приведите правила логического умножения и логического сложения.

2 Определите понятия логической переменной и логической функции. Сколько логических функций существует для двух и трех переменных?

3 Докажите справедливость распределительного закона для логического умножения.

Лабораторная работа №3

Тема: Булева алгебра. Основные законы булевой алгебры. Формы представления булевых функций. Минимизация булевых функций.

Цель: Научиться минимизировать булевы функции методом Квайна и с помощью диаграмм Вейча (Карт Карно).

1Теоретические сведения

1.1Булева алгебра. Основные законы.

В булевой алгебре для представления любой функции в виде формулы кроме символов основных операций отрицания, конъюнкции и дизъюнкции используются следующие символы: малые буквы типа x, y, $z\ldots$,для обозначения переменных (включая буквы с индексами), константы 0 и 1, пара символов (), которые называются скобками.

Формулы булевой алгебры представляются конечными последовательностями символов, указанных выше, записанных в виде строчек и удовлетворяющих требованиям определения формулы. Формулами булевой алгебры являются: булевы переменные x, y, z,...; константы 0 и 1; выражения вида (AB), (A\/B), $\overline{A}$, где A и B являются формулами.

В булевой алгебре при отсутствии в выражении скобок вводится следующий порядок действий: первыми выполняются операции отрицания, далее — конъюнкции, а затем — дизъюнкции. Наличие в выражении скобок изменяет обычный порядок действий: в первую очередь должны выполняться операции внутри скобок.

Две формулы булевой алгебры будут равносильны (равны, эквивалентны), если равны сопоставляемые им функции, т. е. они принимают одинаковые значения на всех наборах значений аргументов.

Укажем **основные законы булевой алгебры**, позволяющие производить различные тождественные преобразования формул булевой алгебры.

1. Закон двойного отрицания $\overline{\overline{x}} = x$.
2. Закон коммутативности
   $x_1 V x_2 = x_2 V x_1$
   $x_1 x_2 = x_2 x_1$
3. Закон ассоциативности
   $x_1 \vee (x_2 \vee x_3) = (x_1 \vee x_2) \vee_3$
   $x_1 (x_2 x_3) = (x_1 x_2)$
4. Законы дистрибутивности

$$x_1(x_2 \vee x_3) = x_1x_2 \vee x_1x_3$$
$$x_1 \vee x_2x_3 = (x_1 \vee x_2)(x_1 \vee x_3)$$

5. Правила де-Моргана
$$\overline{x_1 \vee x_2} = \overline{x_1}\,\overline{x_2}$$
$$\overline{x_1x_2} = \overline{x_1} \vee \overline{x_2}$$

6. Правила операций с константами 0 и 1
$$\overline{0} = 1; \qquad \overline{1} = 0;$$
$$x1 = 1; \qquad x0 = 0;$$
$$x \vee 0 = x; \qquad x \vee 1 = 1.$$

7. Правила операций с переменной и ее инверсией
$$x \vee \overline{x} = 1;$$
$$x\overline{x} = 0.$$

Справедливость основных законов (тождеств) булевой алгебры может быть доказана перебором всех значений переменных, входящих в проверяемые соотношения.

Из основных законов можно легко получить следующие важные соотношения.

1. Законы поглощения
$$x_1 \vee x_1x_2 = x_1$$
$$x_1(x_1 \vee x_2) = x_1$$

2. Закон идемпотентности дизъюнкции и конъюнкции
$$x \vee x \vee \ldots x = x$$
$$xx\ldots x = x$$

3. На основании правила де-Моргана, пользуясь методом математической индукции, отрицание любого выражения алгебры логики, построенного с использованием операций конъюнкции, дизъюнкции и отрицания, можно получить заменой в исходном выражении аргументов их отрицаниями и обменом местами символов конъюнкции и дизъюнкции.
4. Используя второй закон дистрибутивности и выполняя тождественные преобразования, можно получить
$$x_1 \vee \overline{x_1}x_2 = x_1 \vee x_2$$

1.2 Формы представления булевых функций

В алгебре логики доказывается, что любую функцию алгебры логики, кроме функции f =0, можно представить в виде формулы через конъюнкцию, дизъюнкцию и отрицание в виде

$$z = f(x_1, x_2, \ldots, x_k, \ldots, x_n) = \vee x_1^{\sigma^1} x_2^{\sigma^2} \ldots x_k^{\sigma^k} \ldots x_n^{\sigma^n} \quad (1)$$

где $x_k^{\sigma^k}$ — общее обозначение для аргумента Xk и его отрицания $\overline{x_k}$ причем $x_k^{\sigma^k} = \begin{cases} x_k \text{ при } \sigma\text{k} = 1; \\ \overline{x_k} \text{ при } \sigma\text{k} = 0. \end{cases}$

Логическое суммирование в (1) ведется для тех наборов $\sigma 1, \sigma 2, ...\sigma k, ...\sigma n,$, для которых $f\,(\text{x}_1, \text{x}_2, ..., \text{x}_\text{k}, ...\text{x}_n)$=1. Представление функции алгебры логики в форме (1) называют **совершенной дизъюнктивной нормальной формой (СДНФ)**. Члены, входящие в СДНФ, называют **дизъюнктивными членами** или **конституентами единицы**.

Для таблично заданной булевой функции СДНФ строится так: выписываем из таблицы те наборы, для которых функция равна единице, для каждого выписанного набора составляем конъюнкцию $x_1^{\sigma^1} x_2^{\sigma^2} ... x_n^{\sigma^n}$, соединяя полученные конъюнкции знаком дизъюнкции, получим СДНФ искомой функции.

Пример1

Составим СДНФ для таблично заданной функции (табл. 1). Воспользовавшись правилом построения СДНФ, получим: $z = f(x_1, x_2, x_3)$ $= \overline{x_1}\,\overline{x_2} x_3 \vee \overline{x_1} x_2 x_3 \vee x_1 \overline{x_2} x_3 \vee x_1 x_2 \overline{x_3}$

Таблица 1

x_1	x_2	x_3	z	x_1	x_2	x_3	z
0	0	0	0	1	0	0	0
0	0	1	1	1	0	1	1
0	1	0	0	1	1	0	1
0	1	1	1	1	1	1	0

Возможно также иное представление функций алгебры логики, называемое **совершенной конъюнктивной нормальной формой (СКНФ)**. В этом случае функция составляется из дизъюнкций, называемых **конъюнктивными членами** СКНФ или **конституентами нуля**, объединенных знаком конъюнкции. Для рассмотренного примера СКНФ функции будет иметь вид: z= $(x1 \vee x_2 \vee x3)$ $(x_1 \vee \overline{x_2} \vee x_3)(\overline{x_1} \vee \overline{x_2} \vee x_3)(\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$

Переход от СДНФ к СКНФ представления функций алгебры логики осуществляется так: выписывается логическая сумма дизъюнктивных членов, не вошедших в СДНФ, т. е. отрицание функции; от полученной логической суммы дизъюнктивных членов берется отрицание.

Пример 2

Построим СКНФ булевой функции по ее СДНФ: z=$f(x_1, x_2) = \overline{x_1} x_2 \vee x_1 \overline{x_2}$. Для этой функции, ее отрицание имеет вид: $\overline{z} = x_1 x_2 \vee \overline{x_1}\,\overline{x_2}$.

Откуда z=$\overline{\overline{z}} = (\overline{x_1} \vee \overline{x_2})(x_1 \vee x_2)$.

Аналогично осуществляется переход от СКНФ к СДНФ представления функции алгебры логики.

1.3 Понятие о полноте системы функций алгебры логики

Система элементарных булевых функций $\varphi_1, \varphi_2, ..., \varphi_m$ называется функционально полной, если любую функцию алгебры логики можно представить в виде суперпозиции функций $\varphi_1, \varphi_2, ..., \varphi_m$. Примером функционально полной системы элементарных булевых функций служит система трех функций: $\varphi_1 = \overline{x}, \varphi_2 = x_1 x_2, \varphi_3 = x_1 \vee x_2$. Это следует из того, что любую функцию алгебры логики можно представить в виде формулы с помощью конъюнкции, дизъюнкции и отрицания (1). Примерами функционально полных систем элементарных функций также являются:

$$\varphi_1 = \overline{x},\ \varphi_2 = x_1 x_2; \varphi_1 = \overline{x}, \varphi_2 = x_1 \vee x_2$$

$$\varphi = \overline{x_1} \vee \overline{x_2}; \varphi = \overline{x_1 x_2}; \varphi_1 = x_1 \oplus x_2, \varphi_2 = x_1 x_2; \varphi_3 = 1;$$

$$\varphi_1 = x_1 \overline{x_2}, \varphi_2 = 1.$$

Доказательством полноты приведенных выше систем элементарных функций может служить возможность сведения любой из них к функционально полной системе из трех элементарных функций $\overline{x_1}, x_1 x_2, x_1 \vee x_2$. Например, для первой системы элементарных функций отсутствует дизъюнкция. Ее можно получить из выражения $x_1 \vee x_2 = \overline{\overline{x_1}\,\overline{x_2}}$. Для второй системы отсутствует конъюнкция, которая может быть получена из выражения $x_1 x_2 = \overline{\overline{x_1} \vee \overline{x_2}}$. Таким об разом, первая и вторая системы элементарных функций сводятся к функционально полной системе, поэтому они являются функционально полными. Аналогично доказывается полнота и других приведенных выше систем.

3.4 Минимизация булевых функций

Минимизация — процесс приведения булевых функций к такому виду, который допускает наиболее простую, с наименьшим числом элементов, физическую реализацию функции. Частная задача минимизации булевой функции сводится к такому представлению заданной функции, которое содержит наименьшее возможное число букв и наименьшее возможное число операций над ними.

Оценить различные представления одной и той же булевой функции, например ДНФ, можно по количеству входов логических элементов, реализующих заданную функцию. Такую оценку реализации булевой функции называют **ценой реализации булевой функции по Квайну** (или ценой покрытия булевой функции системой логических элементов).

Для изложения методов минимизации введем некоторые определения. Конъюнкция $p = x_1^{\sigma_1} x_2^{\sigma_2} \ldots x_k^{\sigma_k}$ называется элементарной, если число ее членов меньше некоторого множества переменных **n**, причем любая переменная $x_i^{\sigma_i}$ входит в конъюнкцию только один раз. Число членов элементарной конъюнкции определяет ее ранг.

Под **импликантой булевой функции** $f(x_1, x_2 \ldots, x_n)$ понимается такая булева функция $f_1(x_1, x_2 \ldots, x_n)$, если на любом наборе значений переменных $x_1, x_2 \ldots x_n$, на котором значение функции f_1 равно единице, значение функции f также равно единице.

Под **простой импликантой булевой функции** $f(x_1, x_2 \ldots, x_n)$ понимается всякое элементарное произведение $p = x_1^{\sigma_1} x_2^{\sigma_2} \ldots x_k^{\sigma_k}$, которое является импликантой функции f и никакая собственная часть этого произведения в функцию f не входит, т. е. простые импликанты — элементарные конъюнкции наименьшего ранга, входящие в данную булеву функцию.

Сокращенной дизъюнктивной нормальной формой булевой функции называется такая функция, которая равна дизъюнкции всех своих простых импликант. Несмотря на то что сокращенная ДНФ булевой функции содержит меньшее число букв, чем СДНФ этой же функции, она в большинстве случаев допускает дальнейшее упрощение за счет поглощения некоторых простых импликант дизъюнкцией других простых импликант. В случае, если в дизъюнкции простых импликант, представляющих заданную булеву функцию, ни одну из импликант исключить нельзя, то такую дизъюнкцию называют **тупиковой дизъюнктивной нормальной формой заданной функции.**

И некоторые булевы функции имеют несколько тупиковых ДНФ. Тупиковая ДНФ булевой функции, содержащая наименьшее число букв, будет минимальной ДНФ.

Таким образом, общую задачу минимизации булевых функций можно решать в такой последовательности. Для заданной функции находят сокращенную ДНФ, т. е. все простые импликанты. Затем определяют тупиковые ДНФ заданной функции, среди которых выбирают минимальную ДНФ.

Рассмотрим один из наиболее распространенных методов нахождения сокращенных ДНФ — **метод Квайна**. Для этого метода исходная булева функция должна быть представлена в СДНФ. Если эта функция представлена в произвольной ДНФ, то ее с помощью операции развертывания, заключающейся в умножении некоторых членов на выражение вида $x \vee \bar{x} = 1$, приводят к СДНФ. Метод Квайна основан на последовательном применении к парам дизъюнктивных членов операций склеивания и элементарного поглощения. Операция склеивания основана на справедливости тождества $x_1 x_2 \vee x_1 \bar{x}_2 = x_1(x_2 \vee \bar{x}_2) = x_1$.

Вначале в СДНФ исходной булевой функции проводят все возможные операции склеивания дизъюнктивных членов — конъюнкций ранга **n**, где **n** — число аргументов функции. В результате склеивания получим конъюнкции **n-1** ранга. После выполнения операции поглощения с конъюнкциями **n-1** ранга осуществляют все возможные операции склеивания конъюнкций **n-1** ранга. Затем проводят операции поглощения с конъюнкциями **n-2** ранга и вновь выполняют операции склеивания и.т. д.

Пример3

Найдем сокращенную ДНФ булевой функции

$$z = f(x_1, x_2, x_3, x_4) = \bar{x}_1 x_4 \vee x_1 x_3 \vee \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 .$$

Используя операцию развертывания, представим исходную функцию в СДНФ:

$$z = \overset{1}{\bar{x}_1 x_2 \bar{x}_3 x_4} \vee \overset{2}{\bar{x}_1 \bar{x}_2 \bar{x}_3 x_4} \vee \overset{3}{\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4} \vee \overset{4}{\bar{x}_1 x_2 \bar{x}_3 \bar{x}_4} \vee \overset{5}{\bar{x}_1 \bar{x}_2 x_3 x_4} \vee \overset{6}{\bar{x}_1 x_2 x_3 x_4} \vee$$
$$\vee \overset{7}{x_1 \bar{x}_2 x_3 \bar{x}_4} \vee \overset{8}{x_1 \bar{x}_2 x_3 x_4} \vee \overset{9}{x_1 x_2 x_3 \bar{x}_4} \vee \overset{10}{x_1 x_2 x_3 x_4} \vee \overset{11}{x_1 x_2 \bar{x}_3 \bar{x}_4} \vee \overset{12}{x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4} .$$

Пронумеруем все дизъюнктивные члены.

Выполним все возможные операции склеивания дизъюнктивных членов в такой последовательности: 1) первый член со всеми остальными, 2) второй член с остальными, кроме первого, 3) третий член с остальными, кроме первого и второго, и т. д.В результате проведенных операций склеивания и поглощения получим ДНФ заданной функции в форме

$$z = \overset{1}{\bar{x}_1 \bar{x}_3 x_4} \vee \overset{2}{\bar{x}_1 \bar{x}_3 \bar{x}_4} \vee \overset{3}{\bar{x}_1 x_3 x_4} \vee \overset{4}{x_1 x_3 \bar{x}_4} \vee \overset{5}{x_1 x_3 x_4} \vee \overset{6}{x_1 \bar{x}_3 \bar{x}_4} \vee$$
$$\vee \overset{7}{\bar{x}_2 x_3 x_4} \vee \overset{8}{x_2 x_3 x_4} \vee \overset{9}{x_2 \bar{x}_3 \bar{x}_4} \vee \overset{10}{\bar{x}_2 \bar{x}_3 \bar{x}_4} \vee \overset{11}{\bar{x}_1 x_2 \bar{x}_3} \vee \overset{12}{\bar{x}_1 \bar{x}_2 \bar{x}_3} \vee$$
$$\vee \overset{13}{x_1 \bar{x}_2 x_3} \vee \overset{14}{x_1 x_2 x_3} \vee \overset{15}{\bar{x}_1 x_2 x_4} \vee \overset{16}{\bar{x}_1 \bar{x}_2 x_4} \vee \overset{17}{x_1 \bar{x}_2 \bar{x}_4} \vee \overset{18}{x_1 x_2 \bar{x}_4} .$$

Пронумеровав все члены выражения, полученного на предыдущем этапе, выполним все возможные операции склеивания конъюнкций третьего ранга в той же последовательности, что и ранее. В результате проведенных операций склеивания и поглощения исходная функция примет вид

$$z = \bar{x}_1 x_3 \vee \bar{x}_1 x_4 \vee x_1 x_3 \vee x_1 \bar{x}_4 \vee x_3 x_4 \vee \bar{x}_3 \bar{x}_4 .$$

Метод импликантных таблиц является одним из методов нахождения тупиковых и минимальных ДНФ по сокращенным ДНФ. Импликантная таблица представляет собой прямоугольную таблицу,

Таблица 3.9

Номер строки	Простые импликанты	Конституенты единицы											
		$\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4$	$\bar{x}_1\bar{x}_2\bar{x}_3x_4$	$\bar{x}_1\bar{x}_2x_3x_4$	$\bar{x}_1x_2\bar{x}_3\bar{x}_4$	$\bar{x}_1x_2\bar{x}_3x_4$	$\bar{x}_1x_2x_3x_4$	$x_1\bar{x}_2\bar{x}_3\bar{x}_4$	$x_1\bar{x}_2x_3\bar{x}_4$	$x_1\bar{x}_2x_3x_4$	$x_1\bar{x}_2\bar{x}_3\bar{x}_4$	$x_1x_2x_3\bar{x}_4$	$x_1x_2x_3x_4$
1	$x_1\bar{x}_4$							X	X		X	X	
2	$\bar{x}_1x_4$		X	X		X	X						
3	x_1x_3								X	X		X	X
4	$\bar{x}_1\bar{x}_3$	X	X		X	X							
5	x_3x_4			X			X			X			X
6	$\bar{x}_3\bar{x}_4$	X			X			X			X		

столбцы которой соответствуют конституентам единицы исходной булевой функции, а строки — простым импликантам. В случае если простую импликанту можно получить из конституенты единицы вычеркиванием некоторых букв, то говорят, что импликанта накрывает (покрывает) конституенту. В этом случае клетка импликантной таблицы, соответствующая импликанте и конституенте, отмечается специальным знаком (например, звездочкой).

Заполним в соответствии с указанным выше правилом импликантную таблицу (табл. 2) для сокращенной ДНФ булевой функции, полученной в рассмотренном выше примере.

Минимальной ДНФ заданной функции будет соответствовать такая система минимального числа строк таблицы, импликанты которых совместно накрывают крестиками все колонки таблицы.

Для получения тупиковых ДНФ можно пользоваться следующими рекомендациями:

1. Выделяются все существенные (или обязательные) импликанты. Существенной импликанте соответствуют столбцы таблицы, в которых имеется только одна метка (звездочка). Она обязательно входит в минимальное покрытие, так как не используя ее, невозможно покрыть все конституенты единицы.

2. Вычеркиваются столбцы таблицы, которым соответствуют выделенные существенные импликанты.

3. Выбирается такая совокупность оставшихся простых импликант после выделения существенных импликант, которая обеспечивает покрытие всех оставшихся столбцов таблицы с минимальной ценой. При большом числе переменных для оставшихся простых импликант и конституент единицы строится новая импликантная таблица.

Для примера (табл.2) возможны два варианта минимальных ДНФ:

$$z = \bar{x}_1\bar{x}_3 \vee x_1\bar{x}_4 \vee x_3x_4;$$

$$z = x_1x_3 \vee \bar{x}_1x_4 \vee \bar{x}_3\bar{x}_4.$$

Первый вариант соответствует системе строк, состоящей из 1, 4 и 5 строк; второй - из 2, 3 и 6 строк.

Диаграммы Вейча также являются методом минимизации булевых функций. Диаграмма Вейча представляет собой своеобразный способ задания булевых функций с помощью специальной таблицы, число клеток которых равно числу всех возможных наборов аргументов булевой функции. Таким образом, каждой клетке диаграммы Вейча можно поставить в соответствие конституенту единицы.

Для записи конкретной булевой функции в те клетки, для которых конституента единицы входит в исходную булевую функцию, записываются единицы, в оставшиеся — нули.

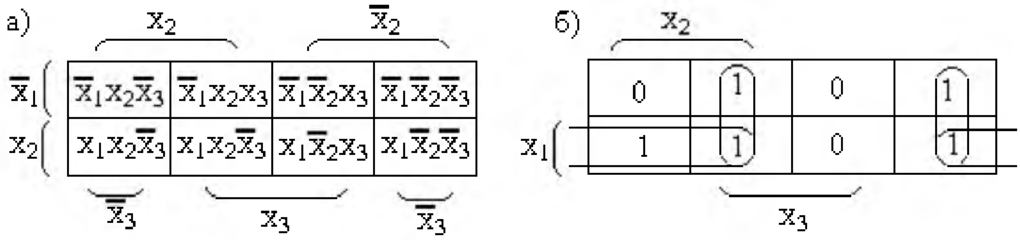


Рис 3.3 Диаграммы Вейча булевой функции трех переменных

Минимизация булевых функций с использованием диаграмм Вейча основывается на отыскании склеивающихся конституент единицы. Для диаграммы Вейча склеивающиеся конституенты единицы располагаются в соседних, вертикально или горизонтально расположенных клетках. Для диаграммы трех переменных (рис. 3.3, а) соседними клетками являются также клетки левого и правого столбцов для одноименных строк.

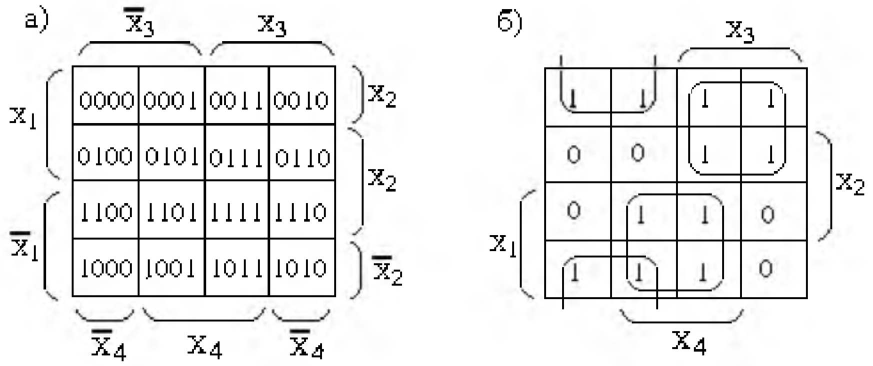


Рис 3.4 Диаграмма Вейча булевой функции четырех переменных

Для диаграммы Вейча четырех переменных (рис. 3.4, а, б) соседними клетками будут также клетки, расположенные в верхних и нижних строках для одноименных столбцов.

Диаграммы Вейча для большего числа переменных могут быть составлены из диаграмм меньшего числа переменных. Например, диаграмму Вейча для пяти переменных можно составить, соединив две диаграммы для четырех переменных. При этом удобно подсоединять их друг к другу одноименными столбцами (рис. 3.5).

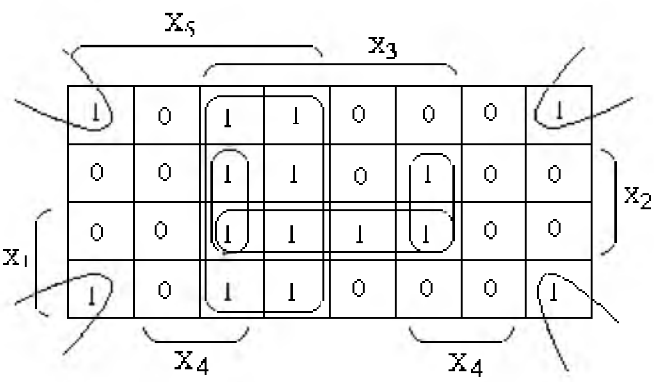


Рис 3.5 Диаграмма Вейча булевой функции пяти переменных

Тогда соседними клетками будут также клетки столбцов, симметрично расположенных относительно линии присоединения диаграмм Вейча четырех переменных для одноименных строк.

Процесс отыскания минимальной ДНФ заключается в том, чтобы всю совокупность единиц диаграммы Вейча накрыть наименьшим числом наиболее коротких произведений. Для этого соседние клетки диаграммы, содержащие единицы, объединяют в группы. Каждой такой группе будет соответствовать группа склеивающихся конституент единицы. Причем количество клеток, входящих в одну группу, равно 2^k (где $k = 1,2,3,...$), а каждая клетка, входящая в группу, должна иметь **k** соседних клеток.

Приведем некоторые рекомендации по минимизации булевых функций с использованием диаграмм Вейча.

1. Рассматриваются поочередно клетки, содержащие единицы, и анализируются всевозможные варианты склеивания. При этом сначала склеивание выполняется только для тех клеток (единиц), для которых вариант склеивания единственный. В результате будут выделены обязательные (или существенные) простые импликанты.

2. Оставшиеся не склеенные клетки (единицы) необходимо оклеивать таким образом, чтобы образовать минимальное число групп с максимальным числом клеток в каждой группе.

3. Каждой группе объединенных клеток в минимальной ДНФ будет соответствовать простая импликанта, определяемая как конъюнкция только тех переменных, значения которых постоянны для всех наборов, задающих клетки данной группы. Группе, состоящей из 2^k клеток, соответствует конъюнкция из **n-k** букв, где **n -** число переменных в исходной булевой функции.

Пример 4

Найдем минимальную ДНФ для булевой функции трех переменных:

$$z = \bar{x}_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 \bar{x}_3.$$

Диаграмма Вейча для этой функции представлена на рис. 3.3, б.

Для этой функции возможны два варианта объединения совокупности единиц в группы. Каждому варианту соответствует своя минимальная ДНФ:

$$z = x_1 \bar{x}_3 \vee x_2 x_3 \vee \bar{x}_2 \bar{x}_3; \qquad z = x_1 x_2 \vee x_2 x_3 \vee \bar{x}_2 \bar{x}_3.$$

Пример 5

Найдем минимальную ДНФ для булевой функции четырех переменных:

$$z = \bar{\mathrm{x}}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{\mathrm{x}}_1 \bar{x}_2 \bar{x}_3 x_4 \vee \bar{\mathrm{x}}_1 \bar{x}_2 x_3 x_4 \vee \bar{\mathrm{x}}_1 \bar{x}_2 x_3 \bar{x}_4 \vee \bar{x}_1 x_2 x_3 x_4 \vee \bar{x}_1 x_2 x_3 \bar{x}_4 \vee$$

$$\vee x_1 x_2 \bar{x}_3 x_4 \vee x_1 x_2 x_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_3 x_4.$$

Диагамма Вейча для этой функции представлена на рис. 3.4,б, а минимальная ДНФ будет иметь вид: $z = \bar{x}_1 x_3 \vee x_1 x_4 \vee \bar{x}_2 \bar{x}_3$.

Задание

Упражнение №1

Найти минимальную ДНФ для булевой функции четырех переменных, заданную диаграммой Вейча (карта Карно).

ри				1	ри		1	1		ри			1	1

1	1	1	
	1	1	
		1	1

		1	1
1	1		
	1	1	

	1	1	
1	1	1	
		1	1

4 вариант

1	1		
	1	1	
	1	1	
		1	1

5 вариант

1	1		
	1	1	
	1	1	
1	1		

6 вариант

1	1		
	1	1	1
		1	1
1	1		

7 вариант

	1	1	
	1	1	1
		1	
		1	1

8 вариант

	1	1	
1			1
1			1
	1	1	

9 вариант

	1	1	
	1	1	
	1	1	
	1	1	

10 вариант

1	1	1	1
1	1	1	1

Упражнение №2

Найти сокращенную ДНФ булевой функции методом Квайна.

№ варианта	Задание
1	$Z = f(x_1, x_2, x_3, x_4) = \bar{x}_1 x_4 \vee x_1 x_2 \vee \bar{x}_3 x_4 \vee x_1 x_2 \bar{x}_3$
2	$Z = f(x_1, x_2, x_3, x_4) = \bar{x}_1 x_4 \vee x_1 \bar{x}_2 \vee \bar{x}_3 x_4 \vee \bar{x}_1 x_2$
3	$Z = f(x_1, x_2, x_3, x_4) = \bar{x}_2 x_3 \vee x_1 x_2 \vee x_3 \bar{x}_4 \vee x_1 \bar{x}_2 x_3$
4	$Z = f(x_1, x_2, x_3, x_4) = x_1 \bar{x}_2 \vee x_2 \bar{x}_3 \vee \bar{x}_2 \bar{x}_3 x_4 \vee x_1 x_2 \bar{x}_3$
5	$Z = f(x_1, x_2, x_3, x_4) = \bar{x}_1 x_3 \vee x_1 \bar{x}_2 x_4 \vee x_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3$
6	$Z = f(x_1, x_2, x_3, x_4) = \bar{x}_1 x_3 \vee \bar{x}_1 x_2 x_3 \vee \bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_4$
7	$Z = f(x_1, x_2, x_3, x_4) = x_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_3 \bar{x}_4 \vee x_1 x_2 \bar{x}_4$

8	$Z = f(x_1, x_2, x_3, x_4) = \overline{x_1} x_2 \vee x_1 \overline{x_2} x_3 \vee \overline{x_3} x_4 \vee \overline{x_1}\,\overline{x_2} x_4$
9	$Z = f(x_1, x_2, x_3, x_4) = \overline{x_1} x_2 x_3 \vee \overline{x_1} x_2 \vee x_2 \overline{x_3} \vee x_3 \overline{x_4}$
10	$Z = f(x_1, x_2, x_3, x_4) = x_1 \overline{x_2} \vee x_1 x_2 \overline{x_3} \vee x_1 x_2 \overline{x_4} \vee \overline{x_2} x_3 \overline{x_4}$

Контрольные вопросы:

1. Что такое СДНФ, СКНФ ?
2. Что понимается под конституентой нуля, конституентой единицы?
3. Дайте определение понятия полнота системы БФ.

Лабораторная работа №4

Тема: RS – триггер.

Цель: Изучить структурную схему и работу RS-триггера. Научиться составлять таблицы состояний и строить временные диаграммы.

1 Теоретические сведения

Важным методом, используемым для описания функционирования RS-триггера, является метод таблиц состояний (таблиц переходов). Таблица состояний (рис. 1.а) RS-триггера в сокращенной форме (эту таблицу называют также управляющей таблицей, таблицей функционирования) содержит два входных сигнала (сигналы R и S) и один выходной сигнал Q (функция). Хотя триггеры имеют два выхода - один прямой Q, а другой - инверсный $\bar{Q}$, в описании триггера и в таблице состояний указывают лишь состояние прямого выхода Q.

Из таблицы состояний триггера видно, что при подаче на вход R уровня лог. «1» триггер принимает состояние логического «0», а при подаче управляющего сигнала «1» на вход S - состояние «1». Следует отметить также, что если до подачи управляющего сигнала, например, на вход R, триггер находился в состоянии логического «0», его состояние не изменится и после подачи сигнала «1» на вход R. Если на обоих входах триггера имеются уровни логического «0»- это состояние соответствует режиму хранения и триггер сохраняет предыдущее состояние. В таблице это состояние обозначено условно Q^0. При подаче на входы R и S одновременно уровня «1» триггер будет находиться в неопределенном (или неправильном) состоянии, поэтому такое сочетание сигналов R и S называется запрещенной комбинацией управляющих сигналов и в таблице состояний обозначается буквой α.

Сокращенная таблица состояний триггера отражает лишь динамику изменения состояния триггера и не учитывает свойство триггера запоминать единицу информации. Полная таблица состояний триггера должна учитывать влияние (на процесс управления) значения предыдущего состояния триггера Q^0. Причем Q^0 представляется как входная переменная. Полная таблица состояний RS -триггера приведена на рис. 1, б.

Таблицу состояний строят так же, как и таблицу истинности.

Анализ таблицы показывает, что только в ситуациях, описываемых строками 4 и 5, происходит изменение состояния триггера.

S	R	Q
0	0	Q^0
0	1	0
1	0	1
1	1	α

а)

N ст	S	R	Q^0	Q
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	0
5	1	0	0	1
6	1	0	1	1
7	1	1	0	α
8	1	1	1	α

б)

Здесь Q^0 - состояние триггера до поступления управляющего сигнала; Q - состояние триггера после поступления сигнала управления; α - неопределенное состояние триггера, соответствующее запрещенным комбинациям входных (управляющих) переменных

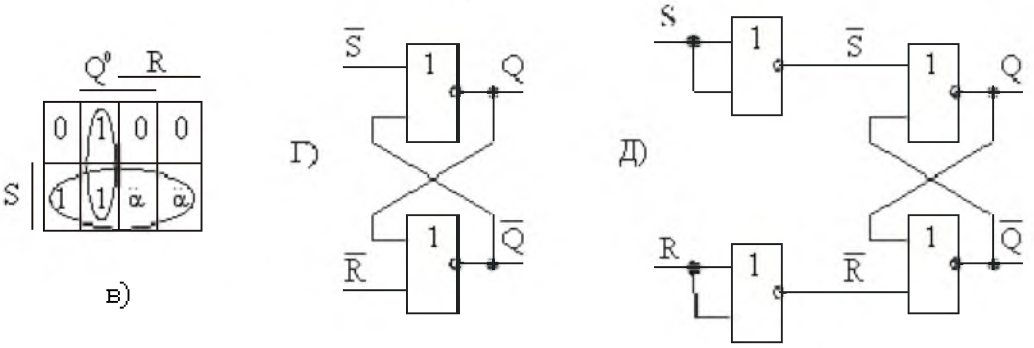


Рис. 1. RS - триггер: а) - упрощенная таблица состояний; б) полная таблица

переходов; в) Карта Карно; г) RS - триггер, управляемый сигналом низкого

уровня ($\overline{R}\,\overline{S}$ - триггер) ; д) RS - триггер на элементах базиса ИЛИ-НЕ

Рассмотрим строку 4. После того, как подается сигнал на вход R, триггер сбрасывается, т.е. переходит из состояния “1” в состояние “0”.

Рассмотрим строку 5. Триггер устанавливается, т.е. переходит из состояния “0” в состояние “1”, в результате подачи сигнала “1” на вход S. Для строк 1 и 2 сигналы S =0 и R=0, и, следовательно, никаких изменений в состоянии триггера не происходит. Для строки 3 сигнал R=1, и этот сигнал в нормальных условиях должен сбросить триггер, но так как триггер уже “сброшен” и Q = 0, то сигнал R = 1 не изменяет его состояние.

Аналогично для строки 6 сигнал S = 1, и этот сигнал в обычных условиях будет устанавливать триггер в “1”, но Q = 1, и, следовательно, состояние триггера останется без изменений до поступления следующего сигнала R.

Особенность RS-триггера заключается в том, что при подаче одновременно на входы R и S сигнала, соответствующего логической 1, состояние триггера становится неопределенным: на обоих выходах Q и $\overline{Q}$ установится уровень “1”, а после снятия со входов управляющих сигналов, в силу случайных причин, триггер может установиться в состояние “0” либо “1”. Очевидно, что **для нормальной работы триггера необходимо исключить указанное сочетание входных сигналов, приводящее к неопределен-**

ному состоянию, что можно осуществить, предусмотрев выполнения запрещающего условия $R \cdot S=0$.

Из таблицы состояний может быть получено уравнение, описывающее поведение триггера. Это уравнение носит название **характеристического уравнения** триггера. Оно показывает, как меняется состояние триггера в зависимости от текущих значений состояния и входов.

Для получения упрощенного аналитического выражения, описывающего поведение RS-триггера, построим карту Карно и проведем соответствующие контуры (рис.1, в). Полученное характеристическое уравнение триггера имеет вид:

$Q = S = Q^0\ \overline{R}$.

Применив закон де Моргана преобразуем полученное выражение в базис И-НЕ:

$$Q = \overline{\overline{S + Q^0\overline{R}}} = \overline{\overline{S}\,\overline{(Q^0\overline{R})}}$$

Схема RS- триггера, реализованного в выбранном базисе, приведена на рис. 1, г.

Из формулы RS - триггера видно, что при реализации его в базисе И-НЕ, триггер управляется сигналами низкого уровня, т.е. уровня лог. "0" (если не предусмотрены инверторы). Для приведения поведения триггера, выполненного на элементах И-НЕ, в соответствие с таблицей состояний сигналы S и R необходимо инвертировать.

Из анализа схемы рис. 1, г очевидно, что простой RS триггер можно сконструировать, соединив “крест-накрест” два элемента И-НЕ.

Входные линии триггера обозначены как $\overline{S}$ и $\overline{R}$, поскольку триггер устанавливается при $\overline{S}$=0 и сбрасывается при $\overline{R}$=0. Такой триггер иногда называют RS-триггер с инверсными входами или конъюнктивной бистабильной ячейкой.

Схема RS-триггера, реализованная в базисе И-НЕ в соответствии с таблицей состояний, приведена на рис. 1, д. Для построения RS -триггера на элементах ИЛИ-НЕ приведем формулу триггера в базис ИЛИ-НЕ

$$\overline{Q} = \overline{S + \overline{\overline{Q^0\overline{R}}}} = \overline{S + \overline{\overline{Q} + R}}$$

Схема RS -триггера, выполненная на элементах базиса ИЛИ-НЕ, приведена на рис. 2, а. Временные диаграммы, поясняющие работу RS-триггера, приведены на рис. 2, б.

Из временных диаграмм (рис. 2, б) следует, что рассмотренные выше RS-триггеры опрокидываются, т.е. управляются сигналами R и S, в любой момент времени. В тех случаях, когда длительности управляющих сигналов не синхронизированы (не согласованы), триггер может находиться в неопределенном состоянии (интервалы времени t_4, t_5), и поэтому такие триггеры называют асинхронными.

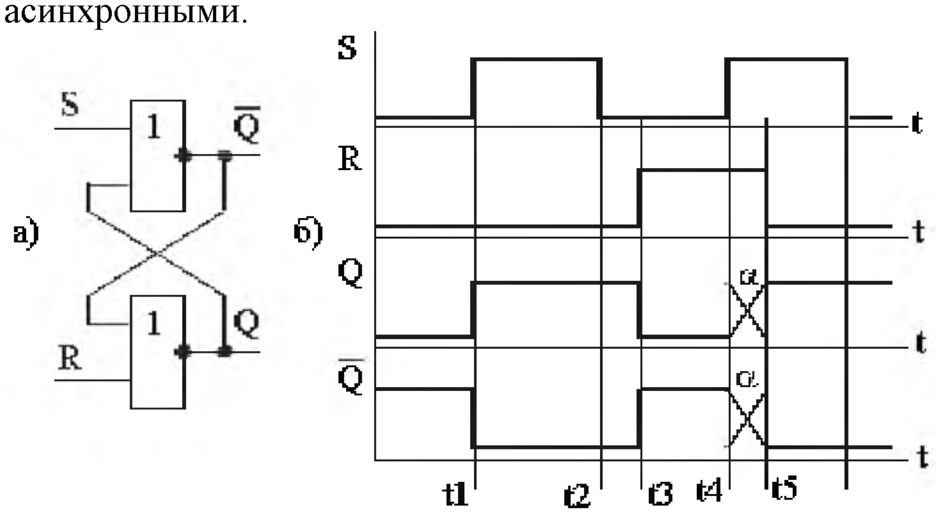


Рис. 2 Схема RS - триггера, выполненного на элементах ИЛИ-НЕ (а) и его временные диаграммы (б)

Триггер, построенный на базе элементов ИЛИ-НЕ, называют также дизъюнктивной бистабильной ячейкой. Бистабильные ячейки, помимо самостоятельного применения, входят в качестве составного узла в триггеры других типов.

Синхронный RS -триггер. Синхронные триггеры снабжаются дополнительным входом, по которому поступает синхронизирующий (тактирующий) сигнал. При этом изменение состояния триггера происходит (при наличии управляющего сигнала) только в те моменты времени, когда на специальный синхровход триггера поступает тактирующий импульс (рис 3, а). Синхронный RS-триггер строится в соответствии с рис. 3, б, а его условное изображение на принципиальных и функциональных схемах приведено на рис. 3, в. Синхронизирующий вход обозначается буквой **С**.

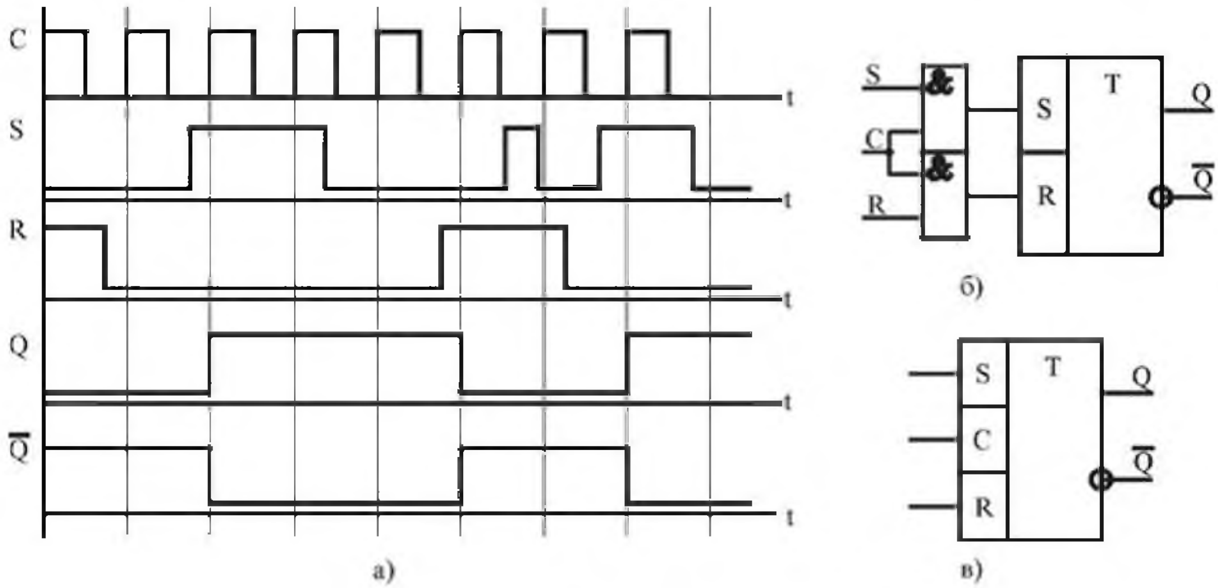


Рис. 3 Временные диаграммы, поясняющие работу синхронного триггера (а), схема реализации (б) и условное обозначение (в)

Входные сигналы S и R являются информационными, а на входе **C** - синхронизирующими, по ним происходит переключение триггера. Следует отметить, что для надежной работы триггера необходимо, чтобы длительность переключающего сигнала (синхронизирующего сигнала) на входе **C** была не меньше времени переключения триггера. Временем переключения (срабатывания, установки) триггера называется время, которое проходит от момента изменения входных сигналов до соответствующего изменения состояния выходов и определяющееся задержками распространения сигнала логическими элементами, входящими в состав триггера.

Двухступенчатый RS - триггер. Рассмотренные схемы RS-триггеров являются одноступенчатыми. Применение одноступенчатых RS-триггеров в качестве самостоятельных запоминающих элементов ограничено. Это связано с неустойчивой работой последовательностной схемы (цифрового автомата), память которой выполнена на одноступенчатых RS-триггерах. Сигналы переключения триггера S(t), R(t) формируются в цифровом автомате комбинационной схемой, в их формировании участвуют, наряду с внешними логическими сигналами, сигналы Q(t) и $\overline{Q}$(t). Переключение одноступенчатого триггера под действием сигналов S(t) и R(t) вызывает изменение значений сигналов Q(t) и $\overline{Q}$(t), а их изменение может привести к изменениям сигналов S(t) или R(t) в том же такте времени t и, как следствие, к ложному срабатыванию триггера. Для устойчивой работы триггера необходимо, чтобы сигналы Q(t) и $\overline{Q}$(t) изменялись только после прекращения действия входного сигнала S(t) или R(t). Это требование выполняется в двухступенчатых триггерах

(MS-триггерах). Базовыми схемами для построения двухступенчатых триггеров являются одноступенчатые RS-триггеры.

Двухступенчатый триггер состоит из двух секций (ступеней), соединенных каскадно, как показано на рис. 3.6 а, причем, каждая секция содержит по синхронному RS-триггеру. Первая секция, ведущая или М-секция (М происходит от английского MASTER) принимает информацию со входных линий S и R. Состояние выходов ведущей секции подается на вторую секцию, ведомую, или S-секцию (S происходит от английского SLAVE).

Для ведущего триггера используется обычная синхронизация, в то время как для ведомого триггера импульс синхронизации инвертируется. Изменение состояния выхода ведущего триггера будет происходить в момент появления положительного импульса синхронизации, и эти изменения будут переданы на входы ведомого триггера. Однако, никакие изменения на выходе ведомого триггера не будут происходить до тех пор, пока не появится положительный сигнал инвертированного импульса синхронизации, т.е. отрицательный (задний фронт) фронт исходного синхроимпульса. Следовательно, изменения на выходах Q и $\overline{Q}$ не произойдет до тех пор, пока не завершится импульс синхронизации. На рис.4, б показаны временные диаграммы работы триггера.

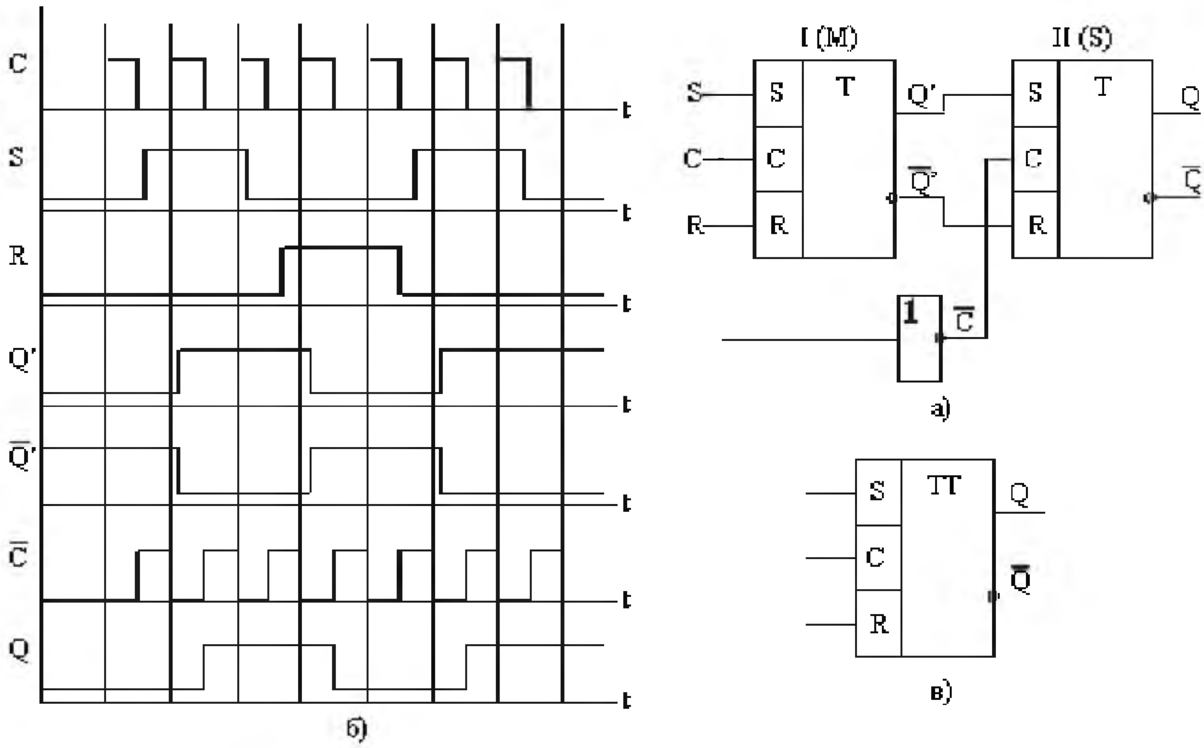


Рис. 4 Двухступенчатый RS - триггер (а), его временные диаграммы (б) и обозначение (в)

На функциональных схемах двухступенчатый триггер изображается в соответствии с рис. 4, в. Символ ТТ в поле условного обозначения означает , что триггер двухступенчатый.

Триггеры с динамическим управлением

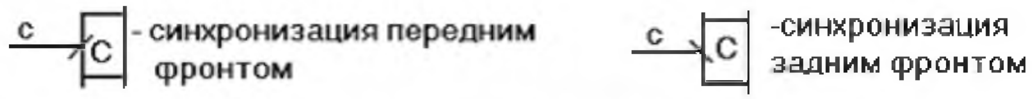


Рис. 5 Фрагменты схемного обозначения линии синхронизации

Динамические триггеры могут опрокидываться как передним, так и задним фронтом тактирующих импульсов. Фрагменты схемного обозначения приведены на рис.5 .

2 Задание

Упражнение №1:

Используя схемы 1 и 2, реализовать схемы асинхронного и синхронного RS-триггера с помощью программы Electronic Work Bench

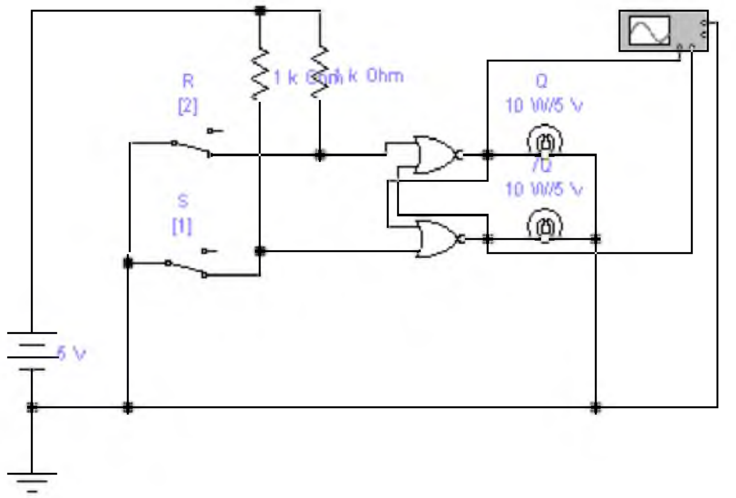


Схема 1

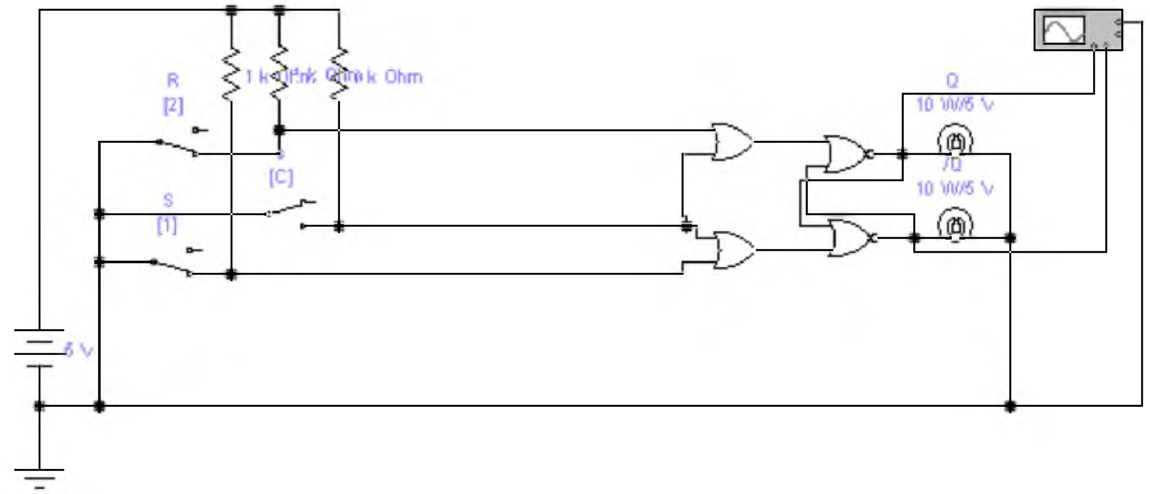


Схема 2

Упражнение №2:

1. Составить схему двухступенчатого RS-триггера на элементах базиса И-НЕ и ИЛИ-НЕ.
2. Нарисовать временную диаграмму двухступенчатого RS-триггера.

Контрольные вопросы:

1. Что такое триггер?
2. Составьте таблицу состояний RS-триггера.
3. Составить временную диаграмму для пояснения работы RS-триггера.
4. Составить схему RS-триггера на элементах базиса И-НЕ и его временную диаграмму.
5. Составить схему RS-триггера на элементах базиса ИЛИ-НЕ и его временную диаграмму.
6. В каком случае состояние RS-триггера становится неопределенным?

Лабораторная работа №5

Тема: **D-триггер.**

Цель: Изучить структурную схему и работу D-триггера. Научиться составлять таблицы состояний и строить временные диаграммы.

1 Теоретические сведения

D-триггер (от английского DELAY) называют информационным триггером, также триггером задержки. D - триггер бывает только синхронным. Он может управляться (переключаться) как уровнем тактирующего импульса, так и его фронтом. Для триггера типа D, состояние в интервале времени между сигналом на входной линии и следующим состоянием триггера формируется проще, чем для любого другого типа.

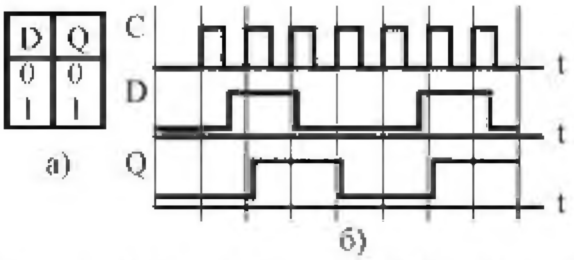


Рис. 1 Таблица управления (а) и временные диаграммы (б) D - триггера

По синхроимпульсу D-триггер принимает то состояние, которое имеет входная линия, согласно управляющей таблице состояний, приведенной на рис. 1, а. На рис. 1, б приведены временные диаграммы, поясняющие его работу.

Как следует из управляющей таблицы, D-триггер имеет как минимум две входные линии: одна - для подачи синхроимпульсов; другая- информационных сигналов. Схемное обозначение D - триггера приведено на рис. 2.

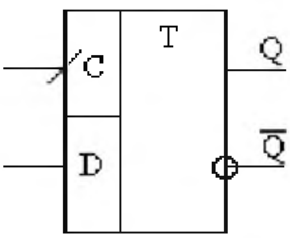


Рис. 2 Схемное обозначение D - триггера, синхронизируемого передним фронтом тактирующего импульса

Для получения характеристической формулы воспользуемся полной таблицей состояния (рис. 3.).

C	D	Q'	Q
0	0	0	0
0	0	1	~
0	1	0	0
0	1	1	~
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Рис. 3 Таблица состояний D - триггера; знаком безразличного состояния (~) обозначены значения Q для двух наборов переменных, т.к. изменение состоянија триггера при нулевом значении сигнала"C" не происходит

Для минимизации логического выражения (характеристической формулы триггера) можно воспользоваться картой Карно (рис. 4, а).

Из рис. 4, а следует, что характеристическое уравнение D-триггера содержит всего одну конъюнкцию, т.е. Q = CD.

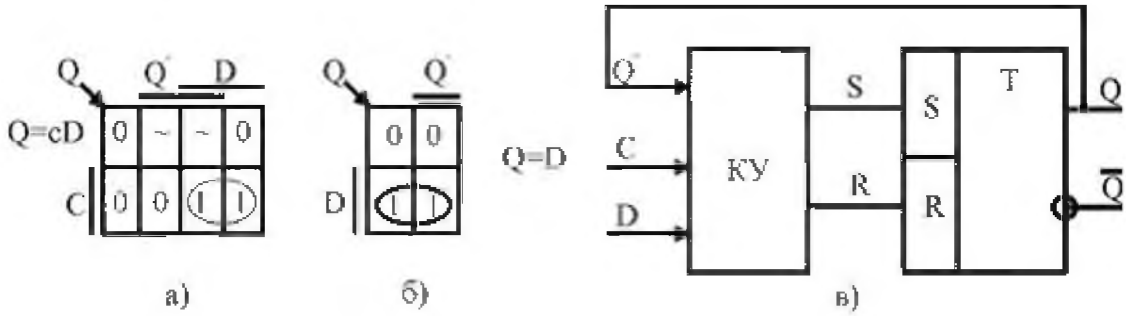


Рис. 4 Карты Карно, составленные по таблице состояний (а), управления (б) и функциональная схемы (в) D-триггера

Если учитывать, что при отсутствии синхроимпульсов состояние D-триггера не меняется, то можно упростить таблицу состояния, оставив в ней только переменные D и Q. Тогда карта Карно будет выглядеть как на рис. 4, б. Сокращенное уравнение D-триггера имеет всего один сигнал – сигнал D.

Это выражение подразумевает наличие сигнала C, т.к. в его отсутствии переключение состояния D-триггера не происходит.

Отметим, что D-триггеры могут переключаться как уровнем синхроимпульса, так и его фронтом. В технической литературе D-триггер, управляемый уровнем синхроимпульса, известен также как триггер-защелка.

Пример синтеза D-триггера, управляемого уровнем синхроимпульса из асинхронного одноступенчатого RS-триггера. Для этого D-триггер представляют как совокупность RS-триггера и комбинационного входного устройства, т.е. представляется так, что входными линиями RS-триггера управляет комбинационное устройство (КУ), согласно характеристической формуле D-триггера (см. рис. 4, в). Входными переменными КУ являются сигналы Q^0 , C, D, а выходными (функциями) - S и R. Если учесть, что сигналы R и S являются для RS-триггера управляющими сигналами, то таблица состояний синтезируемого триггера будет содержать пять столбцов: два столбца - для переменных D-триггера - D и Q^0, один - для функции Q (выходной сигнал синтезируемого триггера, он же является выходным сигналом базового RS-триггера) и два столбца - для переменных R и S RS-триггера

1	2	3	4	5
D	Q	Q	S	R
0	0	0	0	~
0	1	0	0	1
1	0	1	1	0
1	1	1	~	0

а)

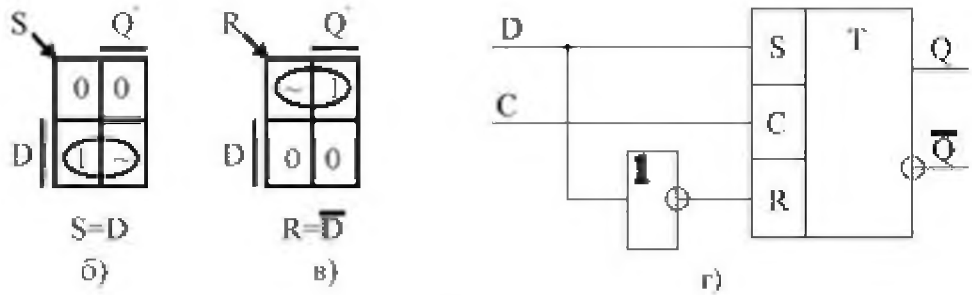


Рис. 5 Комбинированная таблица состояний а), карты Карно б) и в), и функциональная схема, синтезируемого D-триггера

Столбцы 1, 2, 3 соответствуют таблице состояний D-триггера, а в столбцы 4, 5 записываются значения сигналов R и S, при подаче которых на входы RS-триггера, последний должен принимать такие же состояния, что и D-триггер. Это обусловлено тем, что RS-триггер является выходным узлом D-триггера.

Из таблицы (рис.5, а, первая строка) следует: если триггер находился в состоянии “0” (Q^0 =0), чтобы он сохранил это состояние и после поступления очередного синхроимпульса (отметим, что переключение состояния триггера происходит только при наличии синхроимпульса, а каждая строка таблицы состояний соответствует новому синхроимпульсу) на входе S (RS-триггера) необходимо поддерживать уровень “0”, а на входе R - любой уровень, т.к. когда триггер находится в состоянии”0”, он сохраняет это состояние независимо от состояния сигнала R.

Для второй строки таблицы состояний Q^0=1, а новое состояние триггера “0” (Q=0), следовательно, необходимо подать на вход R - уровень логической 1 и т.д. Для каждой строки, где Q^0=1, это состояние триггера сохранится независимо от значения сигнала S (т.к. при S=0 - режим хранения, а при S=1 - запись единицы).

После заполнения таблицы состояний, используя карты Карно (рис. 5, б и в), записывают логические выражения для функций комбинационного устройства S и R (следует помнить, что эти сигналы являются функциями аргументов Q^0 , D и входными переменными для RS-триггера.

По полученным логическим выражениям (см. рис. 5, б и в) можно построить схему D-триггера (рис. 5, г).

Рассмотренный выше D-триггер синтезирован на базе синхронного RS-триггера. Его можно синтезировать и на базе двухступенчатого, а также - простого, асинхронного RS-триггера. Как уже было отмечено выше, переключение D-триггера происходит только при наличии (поступлении) синхроимпульса. С учетом этого, логические функции S и R можно записать в виде

$S = C\ D;$

Схема, реализующая эти функции, содержит два элемента конъюнкции и один инвертор. На рис. 6 приведена схема D-триггера, построенного на базе асинхронного RS-триггера.

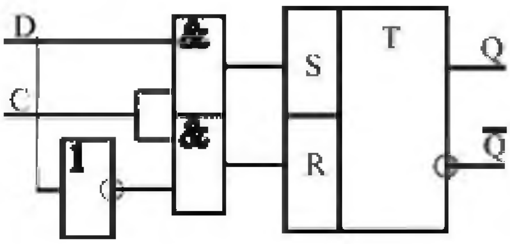


Рис. 6 D-триггер, управляемый уровнем синхроимпульса, построенный на базе асинхронного RS-триггера

На рис. 7 приведено обозначение D-триггера К1533ТМ2, выпускаемого промышленностью в виде интегральной микросхемы (ИМС).

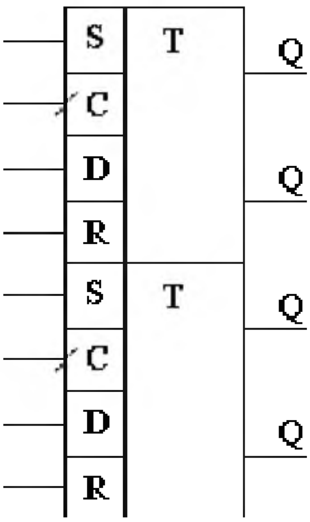


Рис. 7 Схемное обозначение ИМС К1533ТМ2; входы S и R-асинхронные (входы предустановки)

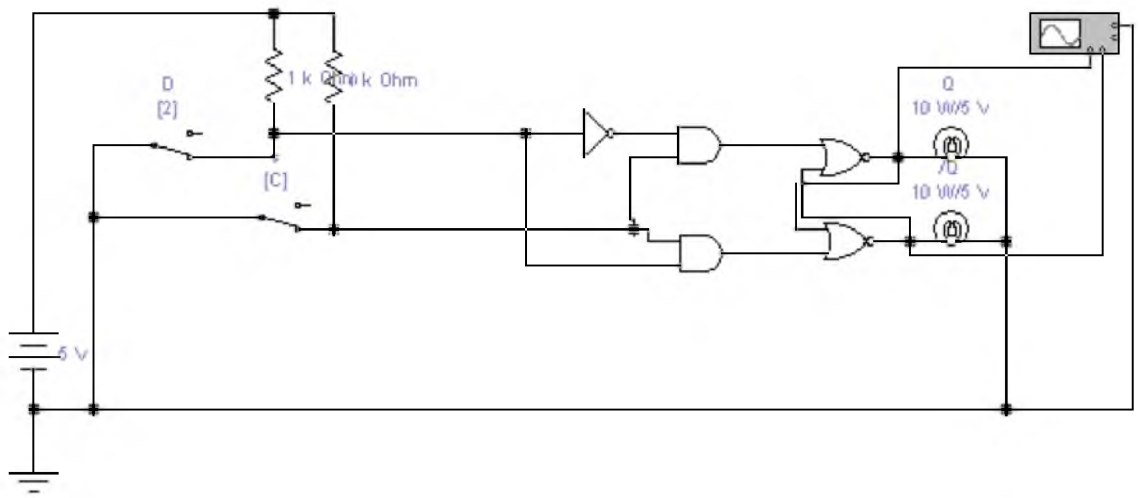
Обычно, в одном корпусе ИМС содержится два D-триггера, управляемых фронтом. D-триггеры в интегральном исполнении имеют также дополнительные асинхронные входы управления S и R. Функции асинхронных входов не зависят от сигналов синхронизации. Отметим, что асинхронные входы имеют и другие типы триггеров. Поскольку дополнительные входы “предустановка” и “очистка”, с помощью которых триггер может быть установлен в нужное состояние независимо от сигналов на других входах, включая синхронизирующий, работают независимо от синхронизации, их называют асинхронными.

Входы “предустановки” и “очистки” напоминают соответствующие входы S и R обычного несинхронизируемого RS-триггера. При подаче “1” на вход R и “0” на вход S (рис. 3.14) триггер устанавливается в состояние “0”. При подаче “1” на оба эти входа поведение триггера не определено, т.е. комбинация S=1, R=1 является запрещенной. При подаче на эти входы “1” поведение триггера полностью определяется другими входными сигналами и синхросигналом.

2 Задание

Упражнение №1

Используя схему 1, реализовать D-триггер с помощью программы Electronic Work Bench



Упражнение №2

1. Реализовать D-триггер, управляемый уровнем синхроимпульса, на базе асинхронного RS-триггера на элементах И-НЕ.
2. Нарисовать временную диаграмму D-триггера.

Контрольные вопросы:

3. Чем управляется D-триггер?
4. Что такое триггер?
5. Составить таблицу состояний D-триггера.
6. Для чего используются входы “предустановки” и “очистки”?

Лабораторная работа № 6

Тема: Дешифраторы.

Цель: Изучить схему и работу дешифратора. Научиться составлять таблицы состояний и строить временные диаграммы.

1 Теоретические сведения

1.1 Линейный или одноступенчатый дешифратор

Дешифратор - это комбинационное устройство, предназначенное для преобразования параллельного двоичного кода в унитарный, т.е. позиционный код. Обычно, указанный в схеме номер вывода дешифратора соответствует десятичному эквиваленту двоичного кода, подаваемого на вход дешифратора в качестве входных переменных, вернее сказать, что при подаче на вход устройства параллельного двоичного кода на выходе дешифратора появится сигнал на том выходе, номер которого соответствует десятичному эквиваленту двоичного кода. Отсюда следует то, что в любой момент времени выходной сигнал будет иметь место только на одном выходе дешифратора. В зависимости от типа дешифратора, этот сигнал может иметь как уровень логической единицы (при этом на всех остальных выходах уровень логического 0), так и уровень логического 0 (при этом на всех остальных выходах уровень логической 1). В дешифраторах каждой выходной функции соответствует только один минтерм, а количество функций определяется количеством разрядов двоичного числа. Если дешифратор реализует все минтермы входных переменных, то он называется полным дешифратором (в качестве примера неполного дешифратора можно привести дешифратор двоично-десятичных чисел).

Рассмотрим пример синтеза дешифратора (полного) 3 $\rightarrow$ 8, следовательно, количество разрядов двоичного числа - 3, количество выходов - 8.

Таблица состояний дешифратора

X_3	X_2	X_1	Z_0	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Как следует из таблицы состояния, каждой функции соответствует только один минтерм, следовательно, не требуется минимизировать эти функции (рис. 1).

Из полученных уравнений и схемы дешифратора следует, что для реализации полного дешифратора на m входов (переменных) потребуются $n = 2^m$ элементов конъюнкции (количество входов каждого элемента “И” равно m)и m элементов отрицания.

1.2 Пирамидальные дешифраторы

Пирамидальные дешифраторы позволяют реализовать схему на базе только двухвходовых элементов логического умножения (конъюнкции). Рассмотрим пример реализации дешифратора 3→8

$Z0=\overline{X3}\,\overline{X2}\,\overline{X1}$
$Z1=\overline{X3}\,\overline{X2}\,X1$
$Z2=X3\,X2\,X1$
$Z3=\overline{X3}\,X2\,X1$
$Z4=X3\,X2\,X1$
$Z5=X3\,\overline{X2}\,X1$
$Z6=X3\,X2\,\overline{X1}$
$Z7=X3\,X2\,X1$

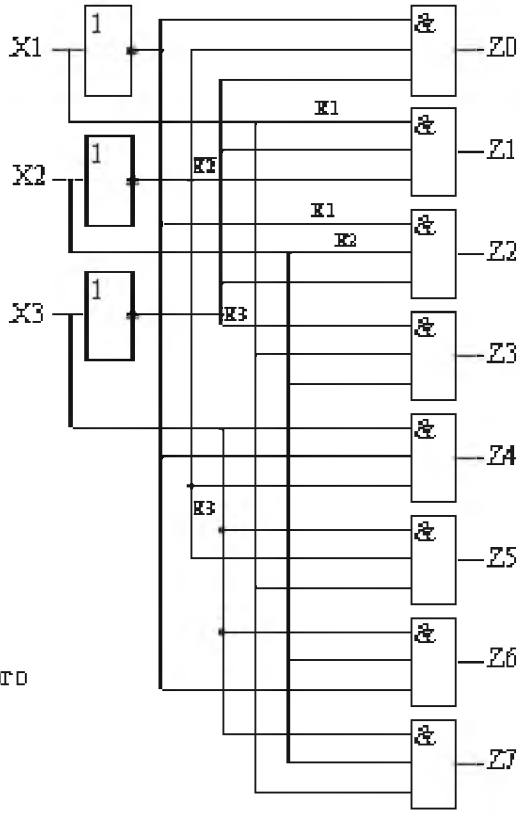


Рис. 1 Схема полного дешифратора 3→8

Для построения такого дешифратора потребуется 12 двухвходовых элементов 2И и три инвертора. Пирамидальные дешифраторы при больших количествах входных переменных позволяют несколько упростить конструкцию устройства, т.е. уменьшить количество интегральных микросхем.

Промышленностью стран СНГ, выпускаются различные модификации дешифраторов в интегральном исполнении. Обозначение дешифраторов на принципиальных схемах показано на рис. 2.

а) б)

Рис. 2. Обозначения дешифраторов:
а – дешифратор с прямыми выходами;
б – дешифратор с инверсными выходами

1.3 Двухступенчатые дешифраторы на интегральных микросхемах. Пример дешифратора для пятиразрядного двоичного кода.

Каждый дешифратор выполнен с управляющими входами, объединенными конъюнктивно. При выполнении условия конъюнкции на выходе, номер которого соответствует десятичному эквиваленту двоичного кода, появится уровень логического “0”. В противном случае все выходы находятся в состоянии логической единицы (рис.3). Как следует из рис. 3, пятиразрядный дешифратор, имеющий 32 выхода, выполнен на базе четырех дешифраторов с использованием лишь одного дополнительного инвертора, что достигнуто благодаря наличию входной управляющей логики каждой интегральной микросхемы. Нетрудно заметить, что входная логика дешифраторов КР1533ИД7 позволяет реализовать функцию дешифратора 2→3 без дополнительных элементов, а полного дешифратора 2→4 с использованием одного инвертора.

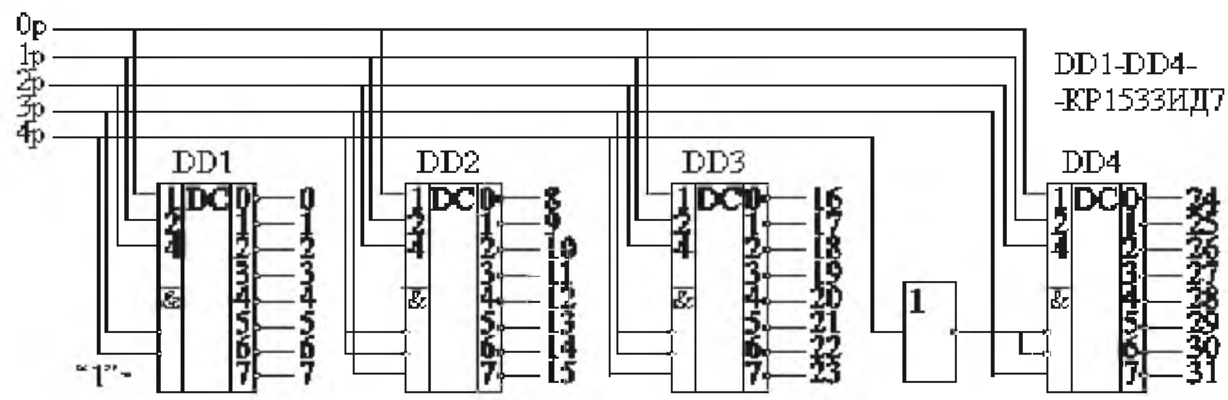
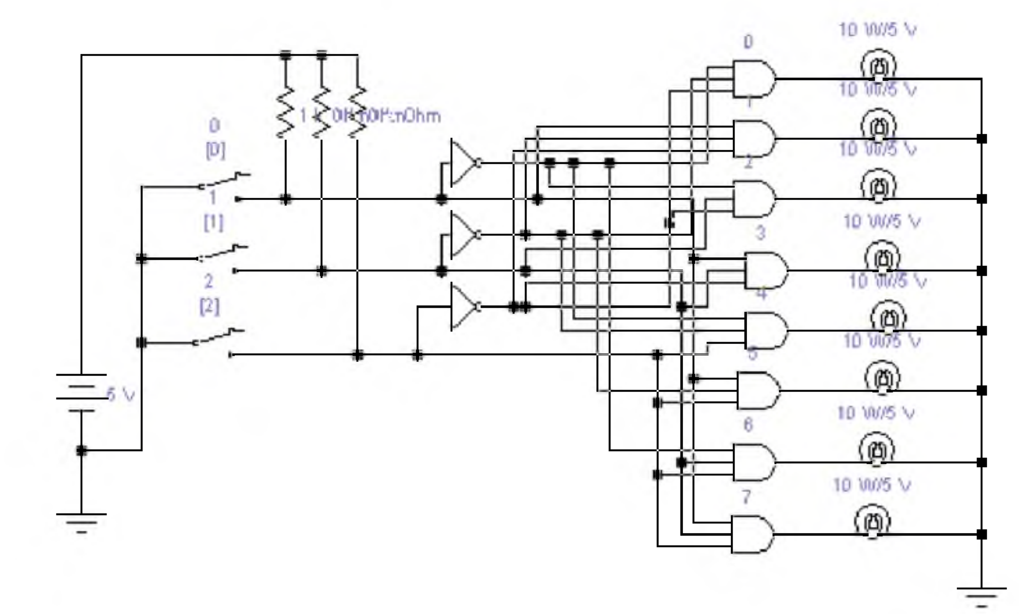


Рис. 3 Схема полного пятиразрядного дешифратора на 32 выхода

2 Задание

Упражнение №1:

Используя схему 1, реализовать дешифратор с помощью программы Electronic Work Bench



Упражнение №2:

1. Используя программу Electronic Work Bench, составить функциональную схему двухступенчатого дешифратора на три входа.
2. Нарисовать временную диаграмму двухступенчатого дешифратора на три входа.

Контрольные вопросы:

1. Что такое дешифратор?
2. Составьте таблицу состояний дешифратора (Выходов дешифратора – 8, количество разрядов двоичного числа - 3).

Лабораторная работа № 7

Тема: **Электронные счетчики и делители частоты.**

Цель: Изучить структурную схему и работу счетчиков. Научиться составлять таблицы состояний и строить временные диаграммы.

1 Теоретические сведения

На базе счетных триггеров можно построить цифровое устройство, получившее название **электронного счетчика**. Электронные счетчики позволяют вести подсчет электрических импульсов, количество которых (поступивших на вход счетчика) представляется, обычно, в параллельном коде. Счетчики могут отличаться модулем счета и типом счетной последовательности, которая, в частности, может быть двоичной, двоично-десятичной, в коде Грея и т.п. Цифровые последовательностные устройства, выполненные по схеме счетчика, но имеющие один счетный вход и один выход называются **делителями** частоты. Таким образом, любой счетчик может служить в качестве делителя частоты, если используется информация только одного из его выходов. Так как счетчики и делители имеют единую структуру, основное внимание будет уделено синтезу счетчиков.

Счетчики и делители подразделяются на **асинхронные** и **синхронные**. У синхронных счетчиков все разрядные триггеры синхронизируются параллельно одними и теми же синхроимпульсами, поступающими из источника этих импульсов. Асинхронные счетчики имеют последовательную синхронизацию, т.е. каждый последующий разрядный триггер синхронизируется выходными импульсами триггера предыдущего разряда. Асинхронные счетчики иногда называют **последовательными**, а синхронные счетчики - **параллельными**.

Синхронные счетчики, в свою очередь, подразделяются на параллельно-синхронные и последовательно-синхронные. Параллельные счетчики имеют более высокую скорость счета, чем асинхронные.

Счетчики, независимо от способа синхронизации, подразделяются на счетчики прямого счета (суммирующие) и на счетчики обратного счета (вычитающие). В интегральном исполнении выпускаются также реверсивные счетчики, в которых имеется специальный вход для переключения режима работы, т.е. направления счета. Многие типы счетчиков, выпускаемые промышленностью в интегральном исполнении, имеют дополнительные входы предустановки, позволяющие использовать эти счетчики в режиме регистра памяти.

В качестве разрядных триггеров счетчиков и делителей могут быть использованы двухступенчатые D-триггеры, T-триггеры и JK-триггеры.

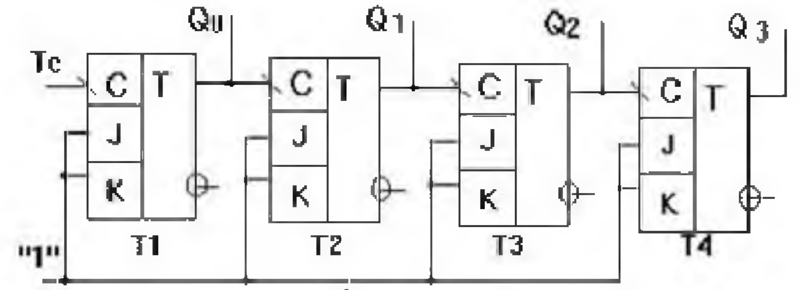
Счетчики относятся к последовательностным устройствам с циклически повторяющейся последовательностью состояний. Число, соответствующее количеству импульсов (поступивших на вход счетчика), при котором счетчик "возвращается" в исходное состояние, называется **модулем** или **коэффициентом** счета. Модуль счета, обычно, обозначают буквой **М** (или $К_{сч}$). Например, максимальный модуль счета счетчика из двух триггеров равен $М = 2^2 = 4$, трех триггеров - $М = 2^3 = 8$ и т.д. В общем случае для n-разрядного счетчика - $М = 2^n$. Модуль счета счетчика численно совпадает с модулем деления делителя частоты. Счетчик по модулю 8 позволяет реализовать (без дополнительных схемных затрат) делитель частоты на 8. Это значит, что данный делитель делит частоту входной импульсной последовательности на 8.

Асинхронный двоичный счетчик. Асинхронный двоичный счетчик представляет собой совокупность последовательно соединенных триггеров (D - или JK), каждый из которых ассоциируется с битом в двоичном представлении числа. Если в счетчике **m** триггеров, то число возможных состояний счетчика равно 2^m, и, следовательно, модуль счета М также равен 2^m. Счетная последовательность в двоичном суммирующем счетчике начинается с нуля и доходит до максимального числа $2^m - 1$, после чего снова проходит через нуль и повторяется. В вычитающем двоичном счетчике последовательные двоичные числа перебираются в обратном порядке, и при повторении последовательности максимальное число следует за нулем.

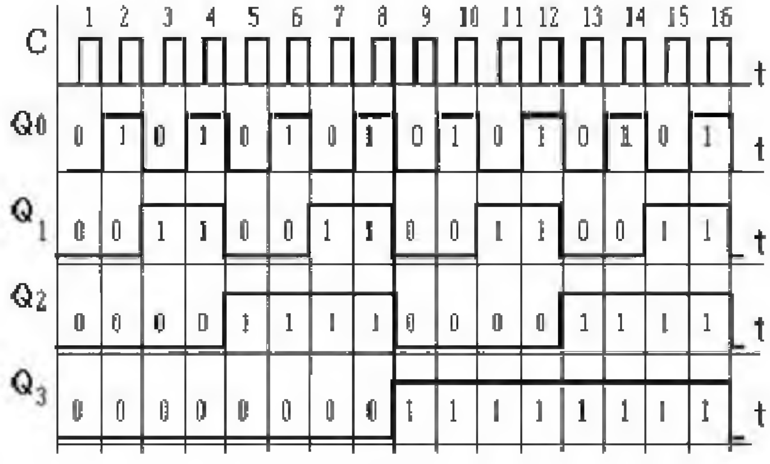
Рассмотрим устройство двоичного суммирующего счетчика по модулю М=16, выполненного на базе JK-триггеров (рис. 1, а).

Как видно из рис. 1, (а), синхронизирующие входы всех триггеров, кроме крайнего левого (Т1), соединены с выходами предыдущих триггеров. Поэтому состояние триггера меняется в ответ на изменение состояния предыдущего триггера.

Из таблицы состояния счетчика (рис. 1, б) легко заметить, что значение разряда в выбранной позиции меняется тогда, когда в соседней справа позиции состояние переходит из "1" в "0", управление триггерами осуществляется задним фронтом синхроимпульсов (отрицательным перепадом напряжения импульса синхронизации).



а)



в)

N	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

б)

Рис. 1. Схема а), таблица состояний триггеров б) и временные диаграммы, поясняющие работу в) последовательного четырехразрядного счетчика на

JK - триггерах

Временные диаграммы, поясняющие работу асинхронного суммирующего счетчика приведены на рис. 1, в.

Счетчики обратного счета (вычитающие счетчики). На рис. 2 приведена схема асинхронного трехразрядного двоичного вычитающего счетчика, построенного на базе D-триггеров. Отметим, что условия для изменения состояний триггеров вычитающих счетчиков аналогичны условиям для суммирующих счетчиков с той лишь разницей, что они должны “опираться” на значения инверсных, а не прямых выходов триггеров. Следовательно, рассмотренный выше счетчик можно превратить в вычитающий, просто переключив входы “C” триггеров с выходов Q на выходы $\overline{Q}$. Когда в качестве разрядных триггеров используются D-триггеры, синхронизируемые передним фронтом синхроимпульсов, для получения вычитающего счетчика (асинхронного) входы “C” последующих тригеров соединяются с прямыми выходами предыдущих, также как в счетчике прямого счета, построенного на JK-триггерах.

Работа вычитающего счетчика на D-триггерах наглядно иллюстрирована на рис. 2, (б). Из рис. 2 следует, что после нулевого состояния всех триггеров, с приходом первого синхроимпульса они устанавливаются в состояние "1". Поступление второго синхроимпульса приводит к уменьшению этого числа на одну единицу и т.д. После поступления восьмого импульса, снова, все триггеры обнуляются и цикл счета повторяется, что соответствует модулю M=8.

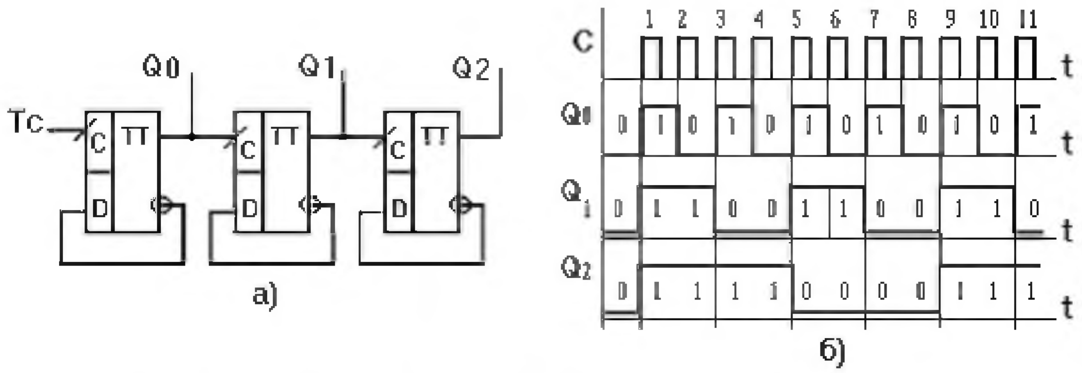


Рис. 2 Схема а) и временные диаграммы вычитающего трехразрядного счетчика на D - триггерах

В некоторых случаях необходимо, чтобы счетчик мог работать как в прямом, так и в обратном направлении счета. Такие **счетчики** называются **реверсивными**. Реверсивные счетчики могут быть как асинхронного, так и синхронного типа. Они строятся путем применения логических коммутаторов (мультиплексоров) в цепях связи между триггерами. Так, например, асинхронный реверсивный двоичный счетчик можно построить, если обеспечить подачу сигналов с прямого (при суммировании) или с инверсного (при вычитании) выхода предыдущего JK- или T-триггера на счетный вход последующего. В случае, когда реверсивный счетчик строится на базе D-триггеров, управляемых передним фронтом, для получения режима прямого счета следует соединить инверсный выход предыдущего с счетным входом последующего триггера.

Параллельные счетчики (**синхронные счетчики**). Параллельные счетчики бывают двух типов: **синхронные параллельные** и **синхронные последовательные**.

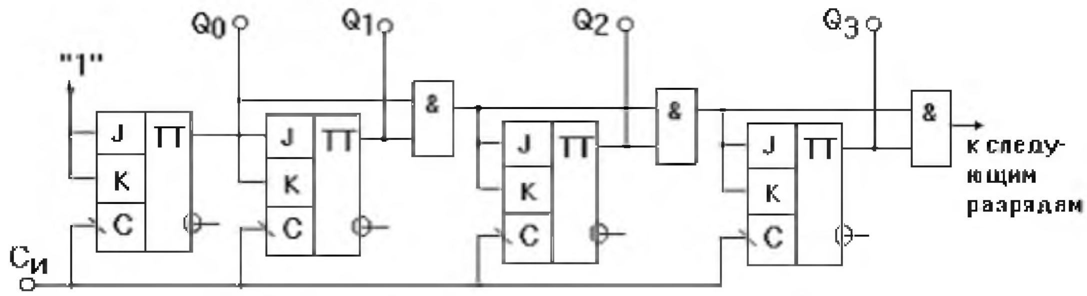


Рис. 3. Синхронный последовательный суммирующий счетчик на JK - триггерах

Синхронный последовательный счетчик. По способу подачи синхроимпульсов такие счетчики параллельные, т.е. синхроимпульсы поступают на все триггеры счетчика параллельно, а по способу управления (подачи управляющих импульсов) - последовательные. Схема синхронного последовательного счетчика, реализованного на JK-триг-герах, приведена на рис. 3.

Синхронный последовательный счетчик обладает повышенным быстродействием, однако, за счет последовательного формирования управляющих уровней, на входы “J” и “K” счетных триггеров, быстродействие несколько уменьшается. От этого недостатка лишены параллельные синхронные счетчики, в которых формирование управляющих уровней и их подача на соответствующие входы триггеров счетчика осуществляется одновременно, т.е. параллельно. Пример реализации параллельного синхронного счетчика иллюстрирован на рис. 4.

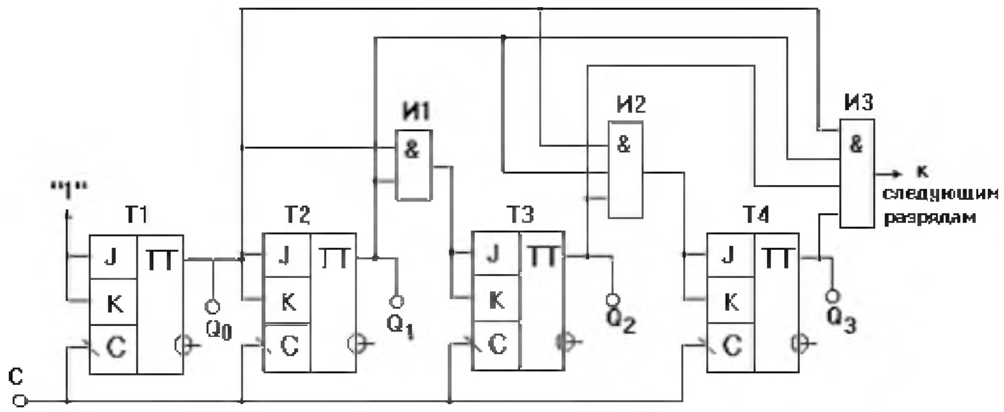


Рис. 4. Параллельный синхронный счетчик на JK – триггерах

Поскольку счетчик имеет одну общую линию синхронизации, состояние триггеров меняется синхронно, т.е. те триггеры, которые по синхроимпульсу должны изменить свое состояние, делают это одновременно, что существенно повышает быстродействие синхронных счетчиков.

Счетчики с произвольным коэффициентом счета. Принцип построения подобного класса счетных устройств состоит в исключении нескольких состояний обычного двоичного счетчика, являющихся избыточными для счетчиков с коэффициентом пересчета, отличающимися от двоичных. При этом избыточные состояния исключаются с помощью обратных связей внутри счетчика.

Число избыточных состояний для любого счетчика определяется из следующего выражения:

$M = 2^m - K_{сч}$,

где M - число запрещенных состояний, $K_{сч}$ - требуемый коэффициент счета; 2^m - число устойчивых состояний двоичного счетчика.

Задача синтеза счетчика с произвольным коэффициентом счета заключается в определении необходимых обратных связей и минимизации их числа. Требуемое количество триггеров определяется из выражения

$n= [\log_2 K_{сч}]$,

где $[\log_2 K_{сч}]$ - двоичный логарифм заданного коэффициента пересчета $K_{сч}$, округленный до ближайшего целого числа.

В каждом отдельном случае приходится применять какие-то конкретные методы получения требуемого коэффициента пересчета. Существует несколько методов получения счетчиков с заданным коэффициентом пересчета $K_{сч}$. Один их этих методов заключается в немедленном сбросе в “0” счетчика, установившегося в комбинацию, соответствующему числу $K_{сч}$. Его называют также **методом автосброса**. Рассмотрим пример реализации счетчика с $K_{сч}$=10 методом автосброса. Очевидно, что “сбрасывая” двоичный четырехразрядный счетчик на нуль каждый раз, когда он будет принимать состояние 1010, можно обеспечить”возврат” счетчика в исходное состояние после каждых десяти импульсов. Подобный прием удобно применять при использовании счетчиков в интегральном исполнении, имеющих ячейки конъюнкции (**И**) на входах установки в нуль, как это сделано в микросхеме К1533ИЕ5. В данном примере (рис. 5.) организованы соединения, обеспечивающие коэффициент пересчета $K_{сч}$ =10.

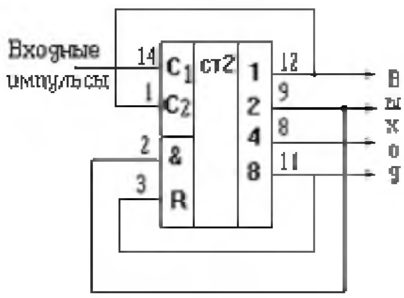


Рис. 5 . Пример реализации счетчика с $K_{сч}$=10

Таблица 1

К1533ИЕ5	Коэффициенты пересчета					
	3	5	6	9	10	12
Вход	14	14	14	14	14	14
Выход	9,12	8,9,12	8,9,12	все	все	все
Соединения выводов	1 - 12 2 - 12 3 - 9	1 - 12 2 -12 3 - 8	1 - 12 2 - 9 3 - 8	1 - 12 2 - 12 3 - 11	1 - 12 2 - 9 3 - 11	1 - 12 2 - 8 3 - 11

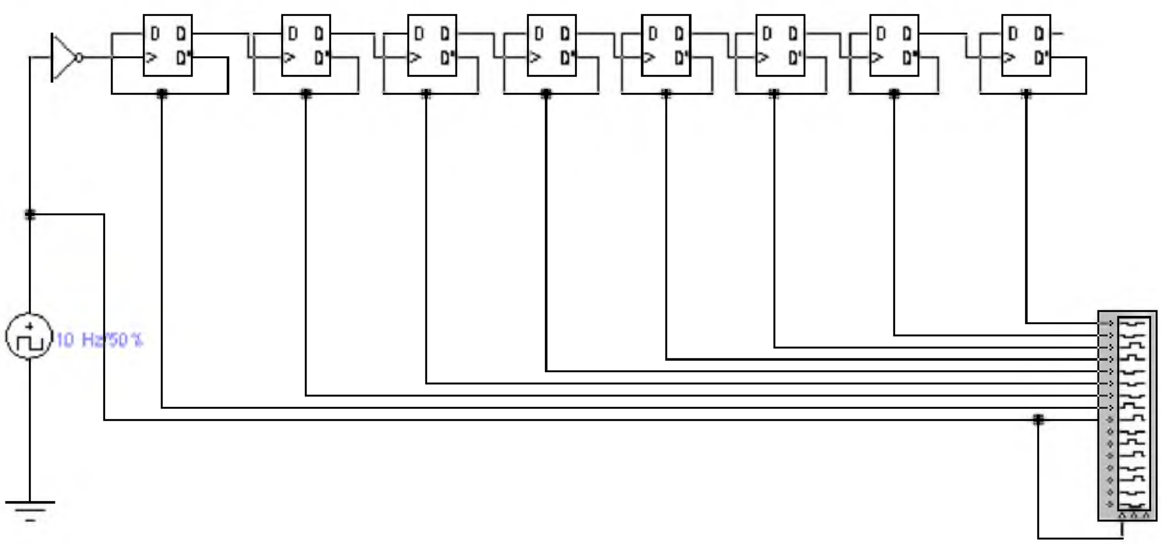
Как следует из рис. 5, роль ячейки, выявляющей факт достижения кодовой комбинации 1010 на выходах счетчика, играет ячейка **И**, уже имеющаяся на входе сброса ИМС К1533ИЕ5.

В таблице 1 поясняются конфигурации соединений для получения различных коэффициентов пересчета с помощью счетчика К1533ИЕ5. Наиболее очевидные варианты получения коэффициентов (2, 4, 8, 16) в таблице не указаны. В графе “Соединения” таблицы указано, какие выводы микросхемы должны быть соединены между собой: например, указание 1-12 означает, что нужно соединить вывод 1 с выводом 12. В строках “Ввод” и “Выход” таблицы указаны номера выводов микросхемы, на которые следует подавать входные импульсы и с которых надлежит снимать выходные, соответственно. Следует отметить, что ИМС К1533ИЕ5 состоит из четырех счетных триггеров, один из которых имеет раздельные выводы входа и выхода, а остальные три триггера соединены последовательно по схеме асинхронного счетчика.

2 Задание

Упражнение №1

С помощью программы Electronic Work Bench, реализовать схему счетчика.



Упражнение №2

С помощью программы Electronic Work Bench, реализовать схему синхронного последовательного суммирующего счетчик на ЈК – триггерах.

Контрольные вопросы:

7. Что такое счетчик?
8. Что такое асинхронный двоичный счетчик? Нарисовать временную диаграмму и составить функциональную схему.
9. Что такое счетчики обратного счета (вычитающие счетчики)?
10. Как можно определить число избыточных состояний любого счетчика?

Список рекомендуемой литературы

Основная

1 Соломенчук В.Г. Аппаратные средства ПК. - М., 2003. - 512с.

2 Мураховский В.И. Евсеев Г.А. Железо персонального компьютера: Практическое руководство, 7-е изд.. 2003. - 688с.

3 Гук М. Аппаратные средства IBM PC: Энциклопедия. 4-е изд., 2005.

4 В.А. Прянишников. Электроника. Курс лекций -Санкт-Петербург: «Корона Пресс», 2000.

Дополнительная:

5 5- Смирнов А.Д. Архитектура вычислительных систем - М.: «Наука», 1990 г.

6 Колесниченко О.В. Шишигин И.В. Аппаратные средства PC: Энциклопедия аппаратных ресурсов ПК, 2000.

7 Ветров С.И. Компьютерное «железо». 2001. - 560 с.