

Министерство образования Красноярского края
КРАЕВОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«КРАСНОЯРСКИЙ ИНДУСТРИАЛЬНО-МЕТАЛЛУРГИЧЕСКИЙ ТЕХНИКУМ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ДЛЯ ВЫПОЛНЕНИЯ КОНТРОЛЬНОЙ РАБОТЫ
для студентов заочного отделения
специальности 22.02.02 «Металлургия цветных металлов»
по учебной дисциплине ЕН.02. «Информатика»

Рассмотрено на заседании цикловой
комиссии 22.00.00
протокол № __ от «__» __ 20__ г.
Председатель ЦК
_____/Ф.И.О.

Красноярск

Пояснительная записка

Данное методическое указание разработано для заочного отделения по специальности 22.02.02 при изучении **ЕН.02. «Информатика»** в соответствии с ФГОС.

В результате освоения учебной дисциплины обучающийся должен уметь:

- использовать изученные прикладные программы;

В результате освоения учебной дисциплины обучающийся должен знать:

- основные понятия автоматизированной обработки информации;
- общий состав и структуру персональных электронно-вычислительных машин и вычислительных систем;
- базовые системные программные продукты и пакеты прикладных программ

Формируемые компетенции:

ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.

ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

ПК 1.5. Выполнять необходимые типовые расчеты.

ПК 3.5. Выполнять необходимые типовые расчеты.

В данной методической разработке представлены: теоретический материал для выполнения контрольной работы и подготовке к аттестации, варианты контрольных работ, список использованных источников для подготовки к промежуточной аттестации для студентов заочников по специальности «Металлургия цветных металлов»

Тематический план и содержание учебной дисциплины ИНФОРМА

Наименование разделов и тем	Содержание учебного материала, лабораторные работы, самостоятельная работа обучающихся, курсовая работа (проект)
1	2
Раздел 1. Информация и информационные процессы	
Тема 1.1. Основные понятия информатики	Понятия информатики и информации. Свойства и носители информации. Кодирование. Измерение информации
	Самостоятельная работа обучающихся: Конспект. Е.В. Михеева Информатика, гл. 1, стр. 8-28
Тема 1.2. Информационные процессы	Системы кодирования данных. Информационные процессы. Информационные ресурсы вычислительной техники.
	Самостоятельная работа обучающихся: Презентация. Тема: «История развития вычислительной техники»
Тема 1.3. Технологии обработки информации	Компьютер, как основа информационных технологий. Основные стадии технологических решений обработки информации. Телекоммуникации
	Самостоятельная работа обучающихся: Конспект. Е.В. Михеева Информатика, гл. 2, стр. 33-39
Тема 1.4. Обработка, хранение и защита информации	Обработка информации центральным процессором. Организация хранения и передачи информации. Размещение информации на дисках. Защита информации от несанкционированного доступа. Антивирусные средства защиты информации
	Самостоятельная работа обучающихся: Презентация. Темы: 1)«Виды вирусов. Способы защиты от них»; 2)«Программы и их виды»
Раздел 2. Состав и структура персональных ЭВМ и вычислительных систем	

Тема 2.1. Технические средства ПК	Архитектура ПК. Основные и дополнительные устройства Архитектура ПК. Процессор. Память. Электронные устройства. Видеокарта. Клавиатура и мышь. Средства хранения и переноса информации. Требования эргономики при работе на компьютере.
	Самостоятельная работа обучающихся: Конспект Е.В. Михеева Информатика, гл. 3, стр. 42-65 Реферат. Тема: «Устройства, образующие внешнюю память ПК» Презентация. Темы: 1)«Архитектура ПК. Основные и дополнительные устройства» 2)«Клавиатура и мышь, История развития»
Тема 2.2. Программное обеспечение ПК	Классификация программного обеспечения. Системное программное обеспечение. Прикладное программное обеспечение
Раздел 3. Компьютерные сети. 1. 2. 3.	Назначение компьютерной сети. Типы сетей. Технические средства работы сети. Сетевые протоколы. Глобальная сеть Интернет. Сервисы в Интернете. Современные технологии создания Web-сайтов.
	Практические занятия Поиск информации в глобальной сети Интернет. Электронная почта Создание презентации по материалам, взятым из Интернет-ресурсов
	Самостоятельная работа обучающихся: Реферат. Тема: «История глобальной сети. Современный Интернет»
Раздел 4. Базовые системные программные продукты	
Тема 4.1. Операционные системы. Файловые менеджеры.	Файловая система организации данных. Операционные системы. Ресурсы. Архиваторы.
	Самостоятельная работа обучающихся: Конспект Е.В. Михеева Информатика, гл. 5, стр. 79-97, гл. 6, стр. 100-105
Тема 4.2. Антивирусные средства защиты информации	Виды вирусов и способы защиты от них. Назначение антивирусных средств. Действия пользователя при наличии признаков заражения ПК. Профилактика заражения.
	Самостоятельная работа обучающихся: Конспект Е.В. Михеева Информатика, гл. 8, стр. 146-156

Тема 4.3. Прикладное программное обеспечение	Текстовые процессоры. Электронные таблицы. Системы управления презентация. Графические редакторы. Структура и классификация информационных систем.
	<p>Практические занятия:</p> <ol style="list-style-type: none"> 4. Создание деловых документов в текстовом редакторе 5. Оформление текстовых документов, содержащих таблицы 6. Создание комплексных документов в текстовом редакторе 7. Контрольная работа в текстовом редакторе 8. Создание электронной книги. Абсолютная и относительная 9. Связанные таблицы. Расчет промежуточных итогов в электр 10. Подбор параметра. Организация обратного расчета в таблиц 11. Задачи оптимизации (поиск решения) в электронных таблиц 12. Знакомство с универсальными математическими системами 13. Решение упражнений, примеров и задач (УМС) 14. Построение графиков в различных системах координат (УМ 15. Решение примеров и задач профессиональной направленнос 16. Создание рисунка в графическом редакторе 17. Создание рисунка профессиональной направленности 18. Создание таблиц базы данных в системе управления БД 19. Создание форм и отчетов в СУБД 20. Знакомство с флэш-технологиями 21. Создание клипа по образцу 22. Создание учебного клипа с применением эффектов 23. Создание учебного клипа с применением эффектов 24. Создание клипа профессиональной направленности 25. Зачет (защита контрольной работы)

Единицы измерения количества информации

За единицу измерения информации принимается такое количество информации, которое содержит сообщение, уменьшающее неопределенность (неполноту знаний) в два раза.

Единица измерения информации называется *бит* (bit) – сокращение от английских слов binary digit («двоичная цифра»).

Рассмотрим на примере, что означает 1 бит информации.

Если положить в мешок два шарика разного цвета, то, загадывая какой шар будет вынут, будем иметь два варианта. Причем, ни один из этих вариантов не имеет преимущества перед другим. Таким образом, неопределенность знаний о цвете вынутого шара перед тем, как шар будет вынут, равна двум. После того как шар вынули, произошло одно из двух возможных событий. Неопределенность знаний уменьшилась в два раза: было два варианта, остался один. Значит, узнав цвет вынутого шара, получили 1 бит информации.

В компьютерной технике отдают предпочтение двоичной системе счисления. Это связано с тем, что в техническом устройстве наиболее просто реализовать два противоположных физических состояния: намагничено – не намагничено, есть сигнал – нет сигнала, заряжен конденсатор – не заряжен и т.п. При этом одно состояние принято обозначать цифрой 0, а другое – цифрой 1.

В качестве единицы измерения информации в технике принят бит, соответствующий одному двоичному разряду со значением 0 или 1. Бит является наименьшей возможной единицей измерения информации. Объем информации, записанной двоичными знаками в памяти компьютера или на внешнем носителе, подсчитывается просто по количеству требуемых для такой записи двоичных разрядов. При этом невозможно нецелое число битов.

Для удобства использования введены и более крупные, чем бит, единицы измерения. Восемь бит (двоичных разрядов) называются *байтом*. Также используются приставки «кило», «мега», «гига» и т.д. для обозначения крупных объемов информации:

1 Килобайт = 1024 байт = 2^{10} байт = 2^{13} бит.

1 Мегабайт = 1024 Килобайт = 2^{10} Килобайт = 2^{20} байт = 2^{23} бит. 1 Гигабайт = 1024

Мегабайт = 2^{10} Мегабайт = 2^{30} байт = 2^{33} бит. 1 Терабайт = 1024 Гигабайт = 2^{10} Гигабайт = 2^{40} байт = 2^{43} бит.

1 Петабайт = 1024 Терабайт = 2^{10} Терабайт = 2^{50} байт = 2^{53} бит.

1 Эксабайт = 1024 Петабайт = 2^{10} Петабайт = 2^{60} байт = 2^{63} бит.

СИСТЕМЫ СЧИСЛЕНИЯ

Любые данные для обработки компьютером представляются последовательностями двух чисел – единицы и нуля. Такая форма представления получила название двоичной. Важным понятием при представлении данных в компьютере является система счисления.

Система счисления – это совокупность приемов и правил представления чисел с помощью символов, имеющих определенное количественное значение.

Различают позиционные системы счисления и непозиционные.

Непозиционные системы счисления – система, в которой символы, обозначающие то или иное количество, не меняют своего значения в зависимости от местоположения (позиции) в изображении числа.

Запись числа A в непозиционной системе счисления может быть представлена выражением:

где D_1, D_2, \dots, D_n – символы системы

Непозиционной системой счисления является самая простая система с одним символом (палочкой). Для изображения какого-либо числа в этой системе надо записать количество палочек, равное данному числу. Это система самая неэффективная, так как форма записи очень громоздка.

К непозиционной системе относится и римская, табл. 1.

Таблица 1

Римские цифры	Значение (обозначаемое количество)	Римские цифры	Значение (обозначаемое количество)
I	1	C	100
V	5	D	500
X	10	M	1000
L	50		

Так, например, в римской системе счисления в числе XXXII (тридцать два) значение цифры X в любой позиции равно десяти.

Запись чисел в данной системе счисления осуществляется по правилам:

1) если цифра слева меньше, чем цифра справа, то левая цифра вычитается из правой (IX: $1 < 10$, следовательно, $10 - 1 = 9$; XC: $10 < 100$, следовательно, $100 - 10 = 90$); 2) если цифра справа меньше или равна цифре слева, то эти цифры складываются (VII: $5 + 1 + 1 = 7$; XXXV: $10 + 10 + 10 + 5 = 35$).

Так, число 1984 в римской системе счисления имеет вид MCMLXXXIV (M – 1000, CM – 900, LXXX – 80, IV – 4).

В римской системе нельзя записывать подряд 4 одинаковых цифр.

В общем случае непозиционные системы счисления характеризуются сложными способами записи чисел и правилами выполнения арифметических операций.

Позиционная система счисления – это система счисления, в которой значение цифры определяется ее местоположением (позицией) в изображении числа.

Алфавит позиционной системы счисления – упорядоченный набор символов (цифр) $\{a_0, a_1, \dots, a_n\}$, используемый для представления чисел в данной системе счисления.

Основание позиционной системы счисления – количество символов (цифр) алфавита $q = n + 1$, используемых для изображения чисел в данной системе счисления.

Примером позиционной системы счисления является десятичная система счисления. Ее алфавит $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Основание $q = 10$.

Например, в десятичной системе счисления число 333 записывается с помощью одной цифры 3, но значение каждой цифры определяется ее местоположением в числе: первая тройка – число сотен в числе, вторая тройка – число десятков, последняя – число

единиц.

За основание системы счисления можно принять любое натуральное число – два, три, четыре и т. д.

Обычно в качестве алфавита берутся последовательные целые числа от 0 до $(q - 1)$ включительно. В тех случаях, когда общепринятых (арабских) цифр не хватает для обозначения всех символов алфавита системы счисления с основанием $q > 10$, используются буквенные обозначения цифр. В табл. 2 приведены алфавиты некоторых систем счисления.

Таблица 2

Система счисления	Основание	Алфавит системы счисления
Двоичная	2	0, 1
Троичная	3	0, 1, 2
Четверичная	4	0, 1, 2, 3
Пятеричная	5	0, 1, 2, 3, 4
Восьмеричная	8	0, 1, 2, 3, 4, 5, 6, 7
Десятичная	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Двенадцатеричная	12	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B
Шестнадцатеричная	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Пример

Записать числа $386,154_{10}$, $101,11_2$, $561,42_8$, $6BF, A_{16}$ в развернутой форме.

Согласно равенству (9) имеем:

$$\begin{aligned}
 386,15_{10} &= 3 \cdot 10^2 + 8 \cdot 10^1 + 6 \cdot 10^0 + 1 \cdot 10^{-1} + 5 \cdot 10^{-2} \\
 101,11_2 &= 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} \\
 561,423_8 &= 5 \cdot 8^2 + 6 \cdot 8^1 + 1 \cdot 8^0 + 4 \cdot 8^{-1} + 2 \cdot 8^{-2} + 3 \cdot 8^{-3} \\
 6BF, A_{16} &= 6 \cdot 16^2 + B \cdot 16^1 + F \cdot 16^0 + A \cdot 16^{-1}
 \end{aligned}$$

В вычислительной технике наибольшее распространение получили двоичная, восьмеричная, шестнадцатеричная системы счисления.

Перевод чисел в позиционных системах счисления

Приведем таблицу для перевода первых 16 чисел в различные системы счисления (табл. 3).

Таблица 3

Десятичные числа q = 10	Двоичные числа q = 2	Восьмеричные числа q = 8	Шестнадцатеричные числа q = 16
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Примеры

1. Перевести число $7A5F_{16}$ в десятичную систему. $q = 16, n = 3$.

$$7A5F_{16} = 7 \cdot 16^3 + A \cdot 16^2 + 5 \cdot 16^1 + F \cdot 16^0 = 7 \cdot 16^3 + 10 \cdot 16^2 + 5 \cdot 16 + 15 = 28672 + 2560 + 80 + 15 = 31327_{10}$$

2. Перевести число $1001,1101_2$ в десятичную систему.

$$q = 2, \quad n = 3, \quad m = 4.$$

$$1001,1101_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} = 8 + 0 + 0 + 1 + 0,5 + 0,25 + 0 + 0,0625 = 9,8125_{10}.$$

3. Перевести число $125,03_8$ в десятичную систему. $q = 8, n = 2, m = 2$.

$$1 \cdot 8^2 + 2 \cdot 8^1 + 5 \cdot 8^0 + 0 \cdot 8^{-1} + 3 \cdot 8^{-2} = 64 + 16 + 5 + 0 + 0,046875 = 85,046875_{10}.$$

Перевод чисел из десятичной системы счисления в систему счисления с основанием q

Перевод вещественного числа из десятичной системы счисления в систему счисления с основанием q осуществляется в два этапа. Переводится отдельно целая и дробная часть числа, а затем при записи числа в новой системе счисления целая часть запятой (точкой) отделяется от дробной.

Перевод целых чисел из десятичной системы счисления в систему счисления с основанием q

Правило

Для перевода целого числа A из десятичной системы счисления в систему с основанием q необходимо A разделить с остатком (нацело) на число q , записанное в десятичной системе. Затем неполное частное, полученное от деления, нужно снова разделить с остатком на q и т. д., пока последнее полученное неполное частное не станет равным нулю. Представлением числа A в новой системе счисления будет последовательность остатков деления, изображенных q -ичной цифрой и записанных в порядке, обратном порядку их получения.

Пример

1. Перевести число 20959_{10} в шестнадцатеричную систему счисления.

Число	Частное	Остаток	
$20959:16 = 1309$		15	↑
$1309:16 = 81$		13	
$81:16 =$	5	1	
$5:16 =$	0	5	

Ответ: $20959_{10} = 51DF_{16}$.

2. Перевести число 405_{10} в двоичную систему счисления.

Число	Частное	Остаток	
$405:2 = 202$		1	↑
$202:2 = 101$		0	
$101:2 = 50$		1	
$50:2 = 25$		0	
$25:2 = 12$		1	
$12:2 = 6$		0	
$6:2 = 3$		0	
$3:2 = 1$		1	
$1:2 = 0$		1	

Ответ: $405_{10} = 110010101_2$.

Перевод правильных дробей из десятичной системы счисления в систему счисления с основанием q

Правило

Для перевода дроби из десятичной системы счисления в систему с основанием q необходимо последовательно выполнять умножение исходной дроби и получаемых дробных произведений на основание системы счисления q до тех пор, пока не получится нулевая дробная часть или не будет достигнута требуемая точность вычислений. Представлением дроби в новой системе счисления будет последовательность полученных целых частей произведения, записанных в порядке их получения.

Примеры

1. Перевести число $A=0,125_{10}$ в двоичную систему счисления.

$$\begin{array}{l} 0,125 \times 2 = 0,25 = 0 \text{ } \textcircled{+0}25 \\ 0,25 \times 2 = 0,5 = 0 \text{ } \textcircled{+0}5 \\ 0,5 \times 2 = 1 \text{ } \textcircled{} \end{array} \quad \downarrow \quad \begin{array}{l} \text{Останавливаемся, т. к. получили нулевую} \\ \text{дробную часть} \end{array}$$

Ответ: $0,125_{10} = 0,001_2$.

2. Перевести число $74,67_{10}$ в восьмеричную систему счисления с точностью до пятого знака.

Переведем сначала в восьмеричную систему счисления целую часть числа, затем дробную часть.

Число	Частное	Остаток
	$74:8 = 9$	2 \uparrow
	$9:8 = 1$	1
	$1:8 = 0$	1

$$74_{10} = 112_8$$

$$\begin{array}{l} 0,67 \times 8 = 5,36 = \textcircled{5} + 0,36 \\ 0,36 \times 8 = 2,88 = \textcircled{2} + 0,88 \\ 0,88 \times 8 = 7,04 = \textcircled{7} + 0,04 \\ 0,04 \times 8 = 0,32 = \textcircled{0} + 0,32 \\ 0,32 \times 8 = 2,56 = \textcircled{}2 + 0,56 \end{array} \quad \downarrow \quad \begin{array}{l} \text{Останавливаемся, т.к. получили 5-й знак} \\ \text{после запятой} \end{array}$$

$$0,67_{10} = 0,52702_8$$

Ответ: $72,67_{10} = 112,52702_8$.

Перевод чисел из двоичной системы счисления в системы с основанием $q = 2^n$

Перевод чисел из двоичной системы в системы с основанием, равным степени двойки, выполняется по более простым правилам, чем с другим основанием.

Правило

Для перевода двоичного числа в систему с основанием $q = 2^n$ нужно число разбить влево и вправо от запятой на группы по n цифр в каждой. Если в первой левой или последней правой группах окажется менее n цифр, то их необходимо дополнить слева и справа нулями. Затем для каждой группы, состоящей из n двоичных цифр, записать соответствующее число в системе счисления $q = 2^n$.

Примеры

1. Число 1011111100000011_2 перевести в восьмеричную систему счисления.

$$q = 8 = 2^3 \quad n = 3.$$

Заданное число разобьем справа налево на группы по 3 цифры (триады) и запишем соответствующие им числа в восьмеричной системе:

$$1011111100000011_2 = 010\ 111\ 111\ 100\ 000\ 011 = 277403_8$$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
2 7 7 4 0 3

2. Число $11011011100011,11011_2$ перевести в шестнадцатеричную систему счисления.

$$q = 16 = 2^4, \quad n = 4.$$

Целую часть числа разобьем справа налево, а дробную – слева направо группы по 4 цифры (тетрады), недостающие группы дополним нулями и запишем соответствующие им числа в шестнадцатеричной системе:

$$11011011100011,11011_2 = 0011\ 0110\ 1110\ 0011,1101\ 1000 = 36E3,D8_{16}$$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
3 6 E 3 D 8

Перевод чисел из систем счисления с основанием $q = 2^n$ в двоичную систему

Правило

Для перевода числа из системы счисления с основанием $q = 2^n$ в двоичную систему нужно каждую цифру числа заменить эквивалентным двоичным числом длиной n разрядов.

Примеры 1. Число $537,45_8$ перевести в двоичную систему счисления.

$$q = 8 = 2^3 \quad n = 3.$$

Заменим каждую цифру числа $537,45_8$ двоичным числом длиной три разряда ($n = 3$)
 $536,45_8 = 101011110,100101_2$

($5 \rightarrow 101$, $3 \rightarrow 011$, $6 \rightarrow 110$, $4 \rightarrow 100$, $5 \rightarrow 101$)

2. Число $5F7,A23_{16}$ перевести в двоичную систему счисления. $q = 16 = 2^4$ $n = 4$.

Заменим каждую цифру числа $5F7,A23_{16}$ двоичным числом длиной четыре разряда ($n = 4$)

$$5F7, A23_{16} = 010111110111, 101000100011_2$$

$$(5 \rightarrow 0101, F \rightarrow 1111, 7 \rightarrow 0111, A \rightarrow 1010, 2 \rightarrow 0010, 3 \rightarrow 0011)$$

Сложение

Если результат сложения двух цифр в системе счисления с основанием q больше q (т. е. полученное число двузначное), то старшая цифра результата равна 1. Таким образом, при сложении в следующий разряд может переходить только единица, а результат сложения в любом разряде будет меньше, чем q . Результат сложения двух положительных чисел имеет столько же значащих цифр, что и максимальное из двух слагаемых, либо на одну цифру больше, но этой цифрой может быть только единица.

Примеры

Сложить числа:

$$1. \quad 100110011_2 + 1101001_2 = 110011100_2$$

$$2. \quad 723,3_8 + 467,53_8 = 1413,03_8$$

$$3. \quad 3B9,6_{16} + 78C,8_{16} = B45,E_{16}$$

$$\begin{array}{r} 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \\ + \quad \quad \quad 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \\ \hline 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \end{array}$$

$$\begin{array}{r} 7 \ 2 \ 3, \ 3 \\ + \quad \quad \quad 4 \ 6 \ 7, \ 5 \ 3 \\ \hline 1 \ 4 \ 1 \ 3, \ 0 \ 3 \end{array}$$

$$\begin{array}{r} 3 \ B \ 9, \ 6 \\ + \quad \quad \quad 7 \ 8 \ C, \ 8 \\ \hline B \ 4 \ 5, \ E \end{array}$$

Если необходимо вычесть из цифры a цифру b и $a \geq b$, то в столбце b таблицы сложения ищем значение числа a . Самая левая цифра в строке, в которой найдено значение числа a , и будет результатом вычитания. Если же $a < b$, то нужно заимствовать единицу из левого разряда, поэтому в столбце ищем число $1a$, и левая цифра в соответствующей строке будет результатом вычитания.

Примеры

Выполнить вычитание чисел:

$$1. \quad 1100000,001_2 - 101101,1_2 = 110010,101_2$$

$$2. \quad 1510,2_8 - 1430,73_8 = 57,25_8$$

$$3. \quad 25E,D8_{16} - 171,6_{16} = ED,78_{16}$$

$$\begin{array}{r} 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0, \ 0 \ 0 \ 1 \\ - \quad \quad \quad 1 \ 0 \ 1 \ 1 \ 0 \ 1, \ 1 \\ \hline 1 \ 1 \ 0 \ 0 \ 1 \ 0, \ 1 \ 0 \ 1 \end{array}$$

$$\begin{array}{r} 1 \ 5 \ 1 \ 0, \ 2 \\ - \quad \quad \quad 1 \ 4 \ 3 \ 0, \ 7 \ 3 \\ \hline 5 \ 7, \ 2 \ 5 \end{array}$$

$$\begin{array}{r} 2 \ 5E, \ D \ 8 \\ - \quad \quad \quad 1 \ 71, \ 6 \\ \hline E \ D, \ 7 \ 8 \end{array}$$

Умножение

Умножение выполняется столбиком с использованием соответствующих таблиц умножения и сложения.

Отметим, что во всех позиционных системах счисления с любым основанием q умножение на числа вида q^m , где m – целое число, сводится просто к перенесению запятой умножаемого на m разрядов вправо или влево (в зависимости от знака m), так же, как и в десятичной системе счисления.

Примеры

Выполнить умножение чисел:

- $10011_2 \cdot 100101_2 = 101011111_2$
- $1176,4_8 \cdot 45,3_8 = 56467,74_8$
- $62, B_{16} \cdot 70, D_{16} = 2B7D, 2F_{16}$

$ \begin{array}{r} 10011 \\ 100101 \\ \hline 10011 \\ 100110 \\ 1001100 \\ 10011000 \\ 100110000 \\ \hline 1001101011 \end{array} $	$ \begin{array}{r} 1176,4 \\ 45,3 \\ \hline 35734 \\ 61704 \\ 47720 \\ \hline 56467,74 \end{array} $	$ \begin{array}{r} 62, B \\ 70, D \\ \hline 502F \\ 2B2D \\ \hline 2B7D, 2F \end{array} $
--	---	---

Деление

Как для умножения, так и для деления нужны обе таблицы – умножения и сложения в соответствующей системе счисления. Само деление выполняется уголком с последующим вычитанием сомножителей.

Выполнить деление:

- $11010111_2 : 101011_2 = 101_2$
- $46230_8 : 53_8 = 710_8$
- $4C98_{16} : 2B_{16} = 1C8_{16}$

$ \begin{array}{r} 11010111 \mid 101011 \\ 101 \\ \hline 10101011 \\ 10101011 \\ \hline 0 \end{array} $	$ \begin{array}{r} 46230 \mid 53 \\ 455 \\ \hline 53 \\ 53 \\ \hline 0 \end{array} $	$ \begin{array}{r} 4C98 \mid 2B \\ 2B \\ \hline 219 \\ 204 \\ \hline 158 \\ 158 \\ \hline 0 \end{array} $
--	---	---

Двоично-десятичная система счисления

Двоично-десятичная система счисления широко используется в цифровых устройствах, когда основная часть операций связана не с обработкой и хранением вводимой информации, а с ее вводом и выводом на какие-либо индикаторы с десятичным представлением полученных результатов (микрокалькуляторы, кассовые аппараты и т. п.).

В двоично-десятичной системе счисления цифры от 0 до 9 представляют четырехразрядными двоичными комбинациями от 0001 до 1001, т.е. двоичными эквивалентами десяти первых шестнадцатеричных чисел (см. табл. 2).

Преобразования из двоично-десятичной системы в десятичную систему и обратные преобразования выполняются путем прямой замены четырех двоичных цифр одной десятичной цифрой или обратной замены

П р и м е р

Преобразовать число 00110111

В десятичную систему.

0011 0111₂₋₁₀
↓ ↓
3 7

Ответ: 00110111₂₋₁₀ = 37₁₀.

Две двоично-десятичные цифры составляют 1 байт. Таким образом, с помощью 1 байта можно представить значения от 0 до 99, а не от 0 до 255, как при использовании 8-разрядного двоичного кода. Используя 1 байт для представления каждой двух десятичных цифр, можно формировать двоично - десятичные числа с любым требуемым числом десятичных разрядов.

Так, если число 1000 0011 0010 0111 рассматривать как двоичное, то его десятичный эквивалент 1000 0011 0010 0111₂ = 33575₁₀ в несколько раз

больше десятичного эквивалента двоично-десятичного числа

1000 0011 0010 0111₂₋₁₀ = 8327₁₀.

ОСНОВЫ ЦИФРОВОЙ ТЕХНИКИ.

Общие сведения о цифровых сигналах и цифровых устройствах.

Под информацией в широком смысле слова принято понимать различные сведения о событиях в общественной жизни, явлениях природы, о процессах в технических устройствах. Она содержится в нашей речи, в текстах книг и газет.

Информацию, воплощённую и зафиксированную в некоторой материальной форме – называют сообщением и передают с помощью сигналов на расстоянии. Непрерывный сигнал изменяется аналогично исходной информации, его обычно называют аналоговым, а устройства, в которых действуют такие сигналы – называют аналоговыми. Существуют такие дискретные сообщения или сигналы, которые содержат фиксированный набор значений.

На практике непрерывное сообщение можно представить в дискретной форме. К непрерывным сигналам можно применять квантование по уровню и по времени (рис. 1).

«Квант» - «quantum» - сколько (количество выражений)
«Tu» - условное обозначение тактового интервала

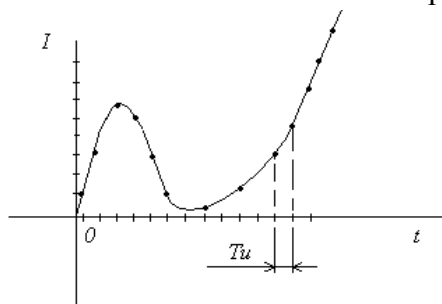


Рис. 1. – квантование непрерывного сигнала

При квантовании по уровню совокупность возможных значений напряжения и тока заменяют конечным набором дискретных значений.

При квантовании по времени непрерывный сигнал заменяется последовательностью импульсов следующих друг за другом за одинаковый промежуток времени.

При квантовании по времени амплитуда каждой выборки (кванта) будет принимать ближайшее разрешённое значение из выбранного конечного набора значений.

Цифровые сигналы в импульсной и потенциальной форме. Для обозначения различных предметов, действий пользуются словами. Слова строятся из букв, которые берутся из алфавита. В цифровой технике для тех же целей пользуются кодовыми словами, для их построения берутся простейший алфавит двоичной системы счисления (0 и 1).

На рис. 2 (а) сигнал сохраняет низкий или высокий уровень в течении одного или нескольких тактов. На рис. 2 (б) двоичной переменной соответствует наличие или отсутствие импульса конечной длительности.

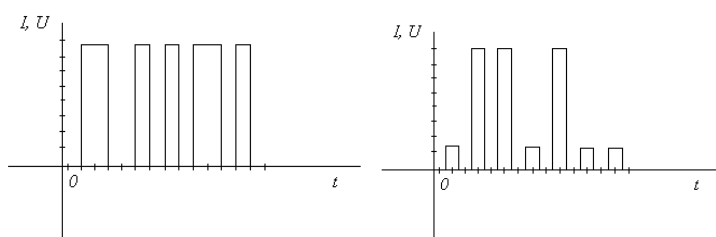


Рис. 2 (а) – потенциальная форма Рис. 2 (б) – импульсная форма

Понятие цифрового устройства (ЦУ) и их разновидности. Функция, описывающая работу логического устройства, называется логической. Устройство, предназначенное для формирования функций алгебры логики, называется цифровыми устройствами.

ЦУ по характеру информации на входах и выходах подразделяются на устройства последовательного (рис. 3а), параллельного (рис. 3б) и смешенного действия.

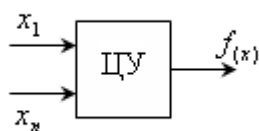


Рис. 3а – последовательное ЦУ

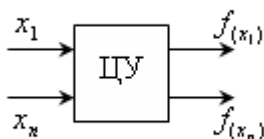


Рис. 3б – параллельное ЦУ

«Комбинационные устройства» - устройства без памяти

По способу функционирования ЦУ делятся на два класса: комбинационные и последовательные.

Комбинационные устройства – автоматы без памяти. Каждый символ на выходе определяется лишь символами, действующими на входе в данный момент времени и не зависит от того, какие символы были раньше.

Последовательные устройства - автоматы с памятью. Выходной сигнал определяется не только набором символов на входе в данный момент времени, но и внутренним состоянием устройства. Внутреннее состояние устройства зависит от того, какие наборы символов действовали во все предыдущие моменты.

Таблица истинности комбинационных цифровых устройств (КЦУ).

Анализ КЦУ и цифровых автоматов проще всего проводить с помощью алгебры логики (алгебры Буля), т.е. оперирующие только с двумя понятиями 0 и 1.

Наиболее наглядно функция преобразования характеризуются таблицей, её называют таблицей истинности. Существует различные способы представления таблицы истинности (ТИ), одни из них представлены ниже (рис. 4).

x_1	x_2	x_3	y
1	0	1	1
0	0	0	0
1	1	1	1
1	0	0	0

f	x_1
x_2	1 0
	0 1

x_1	1	0	1	1
x_2	0	0	0	0
x_3	1	1	1	1
y	1	0	0	0

Рис. 4 – Различные способы представления ТИ

Основы алгебры логики (алгебры Буля). Все сложные ЦУ состоят из простейших логических схем.

Перечень функций двух аргументов.

Функция одного или двух аргументов называются элементарными функциями. Наиболее распространенные правила в алгебре логики (Рис. 5).

$A \cdot 0 = 0$ $A \cdot 1 = A$ $A \cdot A = A$ $A \cdot \bar{A} = 0$	$A + 0 = A$ $A + 1 = 1$ $A + A = A$ $A + \bar{A} = 1$	$\overline{\overline{A}} = A$ $A \cdot B = B \cdot A$ $A + B = B + A$ $(A \cdot B) \cdot C = A \cdot (B \cdot C)$
$(A + B) + C = A + (B + C)$ $A \cdot (B + C) = AB + AC$ $A + (B \cdot C) = (A + B)(A + C)$ $A + AB = A$	$A + \overline{AB} = A + B$ $\overline{AB} = \bar{A} + \bar{B}$ $A + B = \overline{\bar{A} \cdot \bar{B}}$ $\overline{BC + AB} = \overline{BC} \cdot \overline{AB}$	

Рис.5 – Правила в алгебре логики (алгебры Буля)

В последнем блоке рисунка 5 в середине указаны два закона инверсии (де Моргана), которые другими словами можно выразить следующим образом:

Для дизъюнкции. Отрицание дизъюнкции логических переменных эквивалентно конъюнкции отрицаний этих переменных.

Для конъюнкции. Отрицание конъюнкции переменных эквивалентно дизъюнкции отрицаний этих переменных.

Для реализации сложных логических функций их сначала следует представить элементарными, которые последовательно выполняются с помощью простейших логических элементов. Структурно ход выполнения упрощения функции можно представить в виде схемы, называемой графом (рис. 6).

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

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

Итак, имея элементы, выполняющие элементарные функции можно выполнить любую сложную функцию. Такую систему элементарных функций называют базисом. Путём последовательных преобразований функций можно получить так называемый минимальный базис.

Граф – графическое представление последовательных действий упрощения функции.

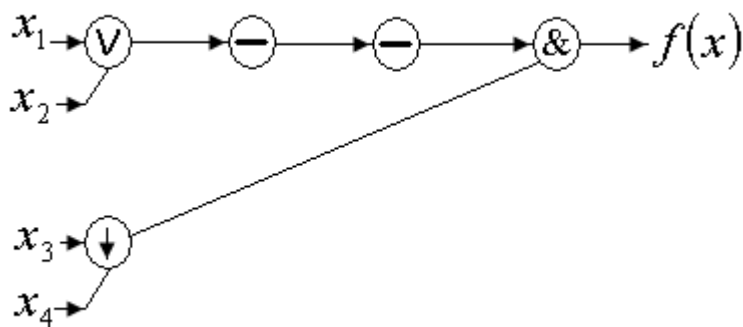


Рис. 6 - Граф

Как видно из рис. 6 упрощение функции сводится к последовательному выполнению действий: суммированию первых двух элементов (переменных), затем выполнение двух операций инвертирования, а затем умножение полученной функции на результат, полученный в результате последовательных действий сложения и инвертирования последних двух переменных (стрелка Пирса). Более подробно все эти действия будут рассмотрены ниже.

Обозначение логических элементов в схемах.

& - знак амперсанд (логическое умножения, лог. «И»).

Амперсанд является графическим сокращением латинского союза *et* (и). Найти упоминания о знаке в литературе докомпьютерной эпохи практически невозможно по причине крайне редкого употребления его в кириллическом наборе. Авторство амперсанда приписывают Марку Туллию Тиرونу, преданному рабу и секретарю Цицерона. Даже после того как Тирон стал вольноотпущенником, он продолжал записывать цicerоновские тексты. И к 63 году до н. э. изобрел свою систему сокращений для ускорения письма, называемую «тироновскими знаками» или «тироновыми нотами» (*Notæ Tironianæ*, оригиналов не сохранилось), которыми пользовались до XI века (так что заодно Тирона считают еще и основоположником римской стенографии).

Раньше даже сокращенную форму *et cetera* (лат. и так далее) писали и печатали как &c. вместо привычного сегодня *etc.*

Первоначально существовало несколько различных видов амперсанды . При произношении алфавита перед буквами, которые кроме звуков являлись еще и словами, произносили *per se* (лат. само по себе). Говорили: *and, per se and* (и, сама по себе „и“). Подобная конструкция просто обязана была стать более адаптированной к частому и быстрому произношению, и уже в 1837 году в словарях было зафиксировано слово *ampersand*.

Ниже приведены обозначения основных элементов цифровых устройств (рис. 7), более подробное их описание показано на рис. 8.

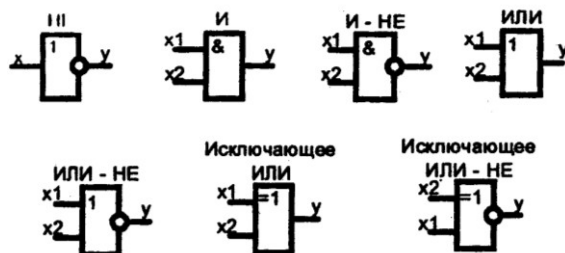


Рис. 7. Обозначение основных узлов цифровых устройств.

Рис. 8 - Обозначение основных логических элементов в схемах.

№	Аргументы	Обозначение	Наименование функции	Комментарии
0	0 0 0 0	0	Константа 0	0=Const
1	0 0 0 1	$\&, \wedge, \cdot$	Конъюнкция (лог. "И", лог. Умножение)	Аналогично «И»
2	0 0 1 0	$X \Delta Y$	Запрет по Y	$X \Delta Y = X \cdot \bar{Y}$
3	0 0 1 1	X	Переменная X	-
4	0 1 0 0	$Y \Delta X$	Запрет по X	$Y \Delta X = Y \cdot \bar{X}$
5	0 1 0 1	Y	Переменная Y	-
6	0 1 1 0	$X \oplus Y$	Сумма по модулю 2	$X \oplus Y = (X \vee Y) \cdot (\bar{X} \wedge \bar{Y})$
7	0 1 1 1	\vee	Дизъюнкция (лог. "ИЛИ", лог. Сложение)	Аналогично «ИЛИ»
8	1 0 0 0	\downarrow	Стрелка Пирса или отрицание дизъюнкции	$\bar{X}_1 \vee \bar{X}_2 = X_1 \downarrow X_2$
9	1 0 0 1	$X=Y$	Эквивалентность	-
10	1 0 1 0	\bar{Y}	Отрицание Y (лог. функция "НЕ")	Обратная величина Y
11	1 0 1 1	$Y \rightarrow X$	Импликация от Y к X	$Y \rightarrow X = \bar{Y} \vee X$
12	1 1 0 0	\bar{X}	Отрицание X (лог. "НЕ")	Обратная величина X
13	1 1 0 1	$X \rightarrow Y$	Импликация от X к Y	$X \rightarrow Y = \bar{X} \vee Y$
14	1 1 1 0	X/Y	Штрих Шеффера (отрицание Конъюнкции)	$\overline{X_1 \wedge X_2} = X_1 / X_2$
15	1 1 1 1	1	Константа 1	1=Const

Инверсия любого сложного выражения, в котором аргументы, либо их инверсии, связаны операциями конъюнкции и дизъюнкции, может быть представлена тем же выражением без инверсии с изменением всех знаков конъюнкции на знаки дизъюнкции, а знаков дизъюнкции на знаки конъюнкции и инверсии всех аргументов.

Пример 1.

Преобразовать следующие функции:

- 1) x/x ;
- 2) $(x_1/x_2)/(x_1/x_2)$

Решение: 1) $x/x = \overline{xx} = \bar{x}$;

$$2) (x_1/x_2)/(x_1/x_2) = \overline{(x_1 \cdot x_2)(x_1 \cdot x_2)} = \overline{x_1 \cdot x_2}.$$

Ответ: 1) \bar{x} ; 2) $\overline{x_1 x_2}$.

Задание 1.

Преобразовать следующие функции:

- 1) $\overline{x_1 \cdot x_2 \cdot x_3 \vee x_1 \cdot x_2 \vee x_3 x_4 x_5}$;
- 2) $x_1 \Delta x_2$;
- 3) $x_1 \rightarrow x_2$;
- 4) $x_1 \downarrow x_2$;
- 5) $\overline{x_1 \Delta x_2}$.

Задание 2.

Доказать справедливость следующего выражения:

$$(x_1 \oplus x_2) = (x_1 \downarrow x_2) \downarrow (\overline{x_1} \downarrow \overline{x_2}).$$

Задание 3.

Пользуясь правилами де Моргана преобразовать выражение:

$$\overline{(x_1 x_2) \cdot (x_3 x_4) \cdot (\overline{x_1} \cdot \overline{x_2})}$$

Задание 4.

Укажите устройства и их количество, входящих в схему, работа которой описано следующей функцией:

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

Задание 5.

Построить цифровое устройство, работа которого описывается следующей функцией:

$$f(x_1 x_2) = (x_1 / x_2) (x_1 \downarrow x_2) (x_1 \vee x_2)$$

Пример 2.

Построить цифровое устройство, работа которого описывается следующей функцией:

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

1. Упростим функцию согласно правилу

де Моргана. При упрощении функции необходимо помнить, что чем меньше количество действий, тем меньше количество простейших устройств. Чем меньше количество простейших устройств, тем экономичнее будет производство данного цифрового устройства:

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

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

2. Прежде чем построить устройство, необходимо на основе функции определить количество простейших элементов: конъюнкции, дизъюнкции, инверсии. При этом нужно учитывать тот факт, что переменные, находящиеся в одной скобке и выполняющие одно действие описывают работу одного устройства, причём количество входов равно количеству переменных (рис. 9).

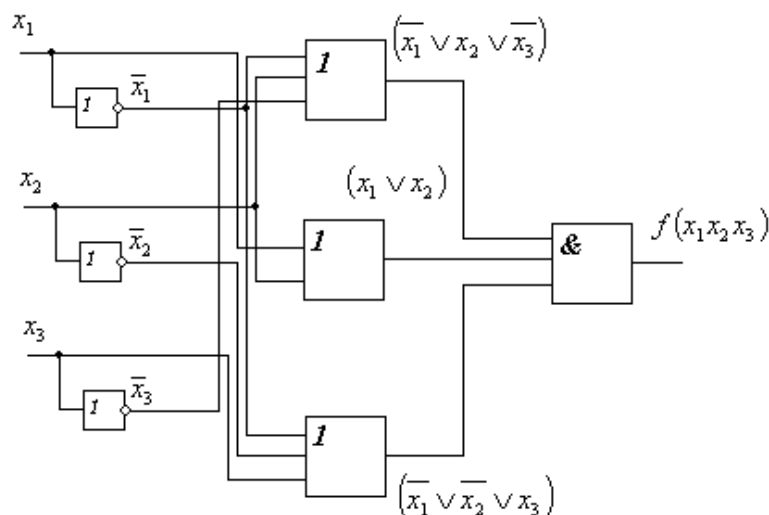


Рис. 9. Построенное цифровое устройство на основе заданной функции.

СДНФ, СКНФ (Совершенная дизъюнктивная нормальная форма, совершенная конъюнктивная нормальная форма).

Синтез логического устройства распадается на несколько этапов. На первом этапе требуется функцию, заданную словесной таблицей или в другой форме представить в виде логического выражения с использованием некоторого базиса (И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ).

При синтезе можно обойтись меньшим набором функций. Последовательно, исключая из базиса функции можно получить минимальный базис.

Под минимальным базисом понимают такой базис, исключение из которого любой функции превращает полную систему в неполную. Дальнейшие этапы сводятся к получению минимальных форм функций, обеспечивающих при синтезе наименьшее количество элементов.

Исходными принято считать две канонические формы представления функций.

ДНФ (дизъюнктивная нормальная форма) – такая форма представления функции, при которой логическое выражение функций строится в виде дизъюнкций ряда членов, каждый из которых является простой конъюнкцией аргументов и их инверсий.

$$\text{ДНФ} = (x_1 \cdot x_2) \vee (\overline{x_1} \cdot x_3 \cdot x_4) \vee (x \cdot \overline{x_2} \cdot x_4) \vee (\overline{x_1} \cdot \overline{x_4});$$

Если в каждом члене ДНФ представлены все аргументы функции или их инверсии, то такая форма называется совершенной дизъюнктивной нормальной формой (СДНФ). Таким же образом представляется и совершенная конъюнктивная нормальная форма, различием лишь является то, что СКНФ выражается в виде конъюнкции аргументов функции.

Пример 4.

$$\text{СДНФ} \rightarrow f(x_1, x_2, x_3) = (x_1 x_2 \overline{x_3}) \vee (\overline{x_1} \overline{x_2} \overline{x_3}) \vee (x_1 \overline{x_2} x_3);$$

$$\text{СКНФ} \rightarrow f(x_1, x_2, x_3) = (x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee x_3);$$

Подбор микросхем, построение и анализ работы схем КЦУ (комбинационных цифровых устройств) в базисе И, ИЛИ, НЕ.

Пример 5.

Получить СДНФ, составить карту Вейча, получить МДНФ (минимизированную дизъюнктивную нормальную форму), перейти к базису И, НЕ, начертить схему, выполнить анализ, дать спецификацию. Расчёты произвести на основе таблицы истинности (Рис.10).

Рис. 10. Таблица истинности.

x_1	x_2	x_3	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

1. Составим СДНФ. Необходимо записать, сколько членов конъюнкции всех аргументов, сколько единиц содержит функция в таблице истинности. Каждая конъюнкция должна соответствовать определённому набору аргументов. Для составления функций используются только те значения выходного аргумента, которые равны 1.

| Любая функция имеет единственную СДНФ

Прежде всего, при решении любой задачи, необходимо знать двоичное кодирование десятичных чисел (Рис. 11).

Рис. 11. Двоичное кодирование десятичных чисел.

№	Код 8421	Код 2421	Код 2 из 5	Код с избытком 3	Код 3а+2	Код 7421
0	0 0 0 0	0 0 0 0	1 1 0 0 0	0 0 1 1	0 0 0 1 0	0 0 0 0
1	0 0 0 1	0 0 0 1	0 1 1 0 0	0 1 0 0	0 0 1 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0	0 0 1 1 0	0 1 0 1	0 1 0 0 0	0 0 1 0
3	0 0 1 1	0 0 1 1	0 0 0 1 1	0 1 1 0	0 1 0 1 1	0 0 1 1
4	0 1 0 0	0 1 0 0	1 0 0 0 1	0 1 1 1	0 1 1 1 0	0 1 0 0
5	0 1 0 1	1 0 1 1	1 0 1 0 0	1 0 0 0	1 0 0 0 1	0 1 0 1
6	0 1 1 0	1 1 0 0	0 1 0 1 0	1 0 0 1	1 0 1 0 0	0 1 1 0
7	0 1 1 1	1 1 0 1	0 0 1 0 1	1 0 1 0	1 0 1 1 1	1 0 0 0
8	1 0 0 0	1 1 1 0	1 0 0 1 0	1 0 1 1	1 1 0 1 0	1 0 0 1
9	1 0 0 1	1 1 1 1	0 1 0 0 1	1 1 0 0	1 1 1 0 1	1 0 1 0

При составлении СДНФ, учитываем лишь те значение выходной переменной (в данном задании y), которые равны 1, все остальные значения игнорируются. Каждая выходная переменная выражаются через значения переменных x . Соответственно в описании СДНФ, те переменные x , которые будут иметь значение 0, обозначаются через инверсию этой переменной.

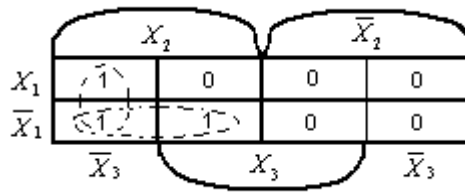
То есть СДНФ в данном случае будет иметь следующий вид:

$$\text{СДНФ: } f(x_1, x_2, x_3) = (\overline{x_1} \overline{x_2} \overline{x_3}) \vee (\overline{x_1} x_2 x_3) \vee (x_1 x_2 \overline{x_3});$$

Таблицы истинности могут быть записаны в других формах, которые могут оказаться более удобными в различных случаях: карты Вейча, карты Карно и т. д.

Рассмотрим карты Вейча (Рис. 12). Данная карта заполняется на основе полученной СДНФ. Значение переменных вписываются в таблицу согласно пересечению их по горизонтали и по вертикали. К примеру, заполним карту Вейча на основе таблицы истинности, изображённой на Рис. 11. При заполнении учитываются лишь значения y равное логической единице, остальные же ячейки, после выставления всех единиц, заполняются нулями.

Рис. 12. Карты Вейча для функции трех переменных



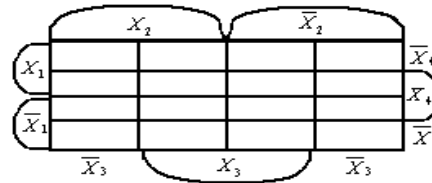
2. Далее объединяем единицы так, как показано пунктирными линиями на Рис. 12. При этом надо учитывать, что количество объединяемых единиц кратно двум, т.е. 2^n , где $n=1, 2, 3 \dots$ и т.д. Также можно объединять одну единицу, если нет соседних.

$$\text{МДНФ: } f(x_1, x_2, x_3) = \overline{x_1}x_2 \vee x_2\overline{x_3} = x_2(\overline{x_1} \vee \overline{x_3});$$

Нахождение МДНФ сводится к тому, что записываются адреса не самих единиц, а объединений, в которые входят единицы.

Аналогично заполняется карта Вейча для четырёх элементов (рис. 13).

Рис. 13. Карты Вейча для функции четырёх переменных



3. Построим устройство в базисе И, И-НЕ.

Первоначально согласно МДНФ, устройство должно быть построено только с использованием элементов И, ИЛИ, НЕ (рис. 14). Поэтому преобразуем полученное МДНФ таким образом, чтобы устройство соответствовало заданному базису. Для этого воспользуемся законами Де-Моргана. Исходя из найденного МДНФ, наше устройство будет содержать один конъюнктор и устройство «Штрих Шеффера», т.е. конъюнктор с инверсией (рис.15).

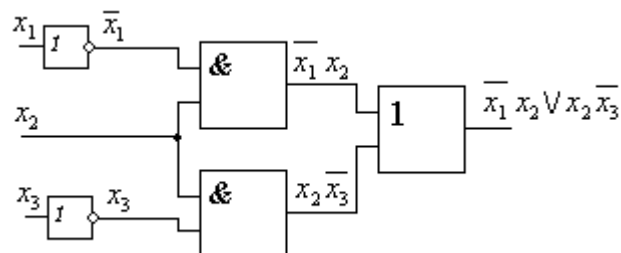


Рис. 14. Цифровое устройство

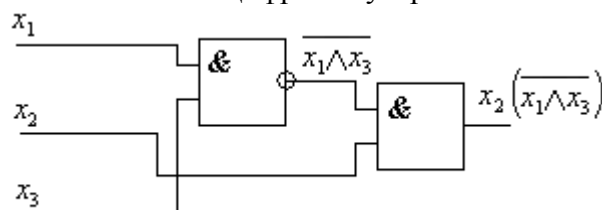


Рис. 15. Цифровое устройство с заданным базисом

Проверка устройства осуществляется путём подачи комбинаций 0 и 1 на вход, результат должен соответствовать выходному значению переменной y в первоначальной таблице истинности.

Задание 7.

Для функции заданной таблицей истинности найдите МДНФ. Постройте схему.

Рис. 16. Таблица истинности к заданию №8.

Задание 8.

Для функции заданной таблицей истинности найдите МДНФ методом карт Вейча. Построить схему в базисе И, ИЛИ, НЕ.

Рис. 17. Таблица истинности к заданию №7.

Задание №8					
x_1	x_2	x_3	x_4	y	
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	1	0	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	1	1	1
1	1	1	0	0	0
1	1	1	1	1	1

Задание №7				
x_1	x_2	x_3	y	
0	0	0	0	0
1	0	0	0	0
0	1	0	0	1
1	1	0	0	1
0	0	1	1	1
1	0	1	1	1
0	1	1	1	1
1	1	1	1	1

КОМБИНАЦИОННЫЕ ЦИФРОВЫЕ УСТРОЙСТВА.

Кодирующие цифровые устройства. Шифраторы.

Шифратор – служит для преобразования десятичного числа в двоичную систему счисления, в упрощённой форме его можно назвать преобразователем кодов (кодер). Он служит для перевода одной формы числа в другую, т.е. формирует соответствующий двоичный код при появлении сигналов на одном из входов. Шифратор наибольшее применение находит в устройствах ввода информации (пультах управления).

Предположим на пульте 10 клавиш (от 0 до 9). При нажатии любой из них на вход шифратора подаётся сигнал, на выходе шифратора должен появиться двоичный код, соответствующий нажатой клавиши.

Условное графическое изображение шифратора изображено на рисунке 18.

Рис. 18. Условное графическое изображение шифратора.

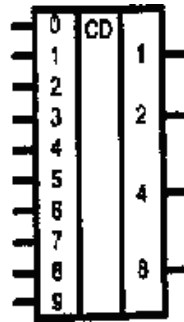


Таблица истинности шифратора изображена на Рис. 19

Рис. 19. Таблица истинности шифратора.

y_1	y_2	y_3	y_4	x
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

При записи функций работы шифратора учитываются только те значение x , которые равны 1, все остальные значение игнорируются.

$$y_4 = x_1 \vee x_3 \vee x_5 \vee x_7 \vee x_9;$$

$$y_3 = x_2 \vee x_3 \vee x_6 \vee x_7;$$

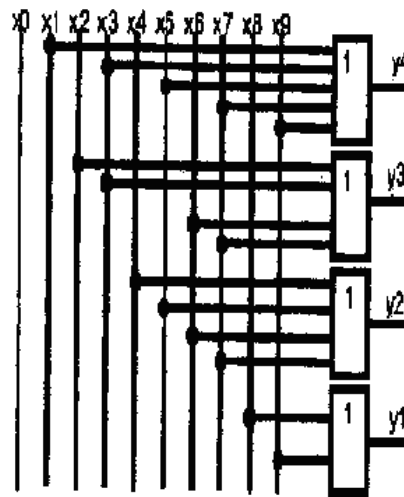
$$y_2 = x_4 \vee x_5 \vee x_6 \vee x_7;$$

$$y_1 = x_8 \vee x_9.$$

Шифратор начинают строить поэтапно. Для начала строится шина (входные контакты устройства), количество контактов в шине напрямую зависит от количества выходных переменных (в данном случае их 10). Количество выходов зависит от количества входных переменных (в данном случае их 4).

На рисунке 20 изображена схема шифратора, построенная в базисе ИЛИ.

Рис. 20. Структурная схема шифратора, построенного в базисе ИЛИ.



Задание 9.

Построить шифратор в базисе И-НЕ на основе таблицы истинности изображённой на рисунке 18.

При построении шифратора в заданном базисе, имеющего инверсию (И-НЕ), необходимо исходную функцию преобразовать, пользуясь правилами Моргана и согласно алгебры Буля.

Например:

$$\begin{aligned}
 y &= x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5; \\
 &= \overline{\overline{x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5}}; \\
 y &= \overline{\overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \wedge \overline{x_4} \wedge \overline{x_5}}; \\
 y &= \overline{\overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \wedge \overline{x_4} \wedge \overline{x_5}}.
 \end{aligned}$$

И уже по полученной функции строим схему.

Дешифраторы.

Дешифратор – устройство для автоматической расшифровки (декодирования) сообщения и перевода содержащейся в нём информации на язык (код) воспринимающей системы.

Дешифратор (декодер) преобразует код, поступающий на его входы в сигнал на один из его выходов.

Чтобы представить схему дешифратора, необходимо составить логические функции, при этом рассуждаем: $y_n=1$, если на входах: $x_0=1, x_1=1, x_2=1, x_3=1$.

Рис. 21. Условное графическое изображение дешифратора.

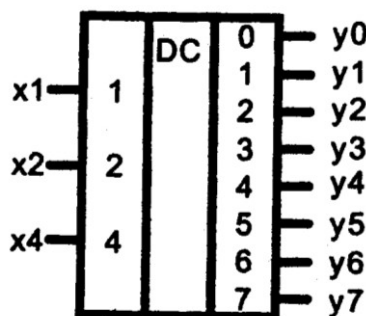


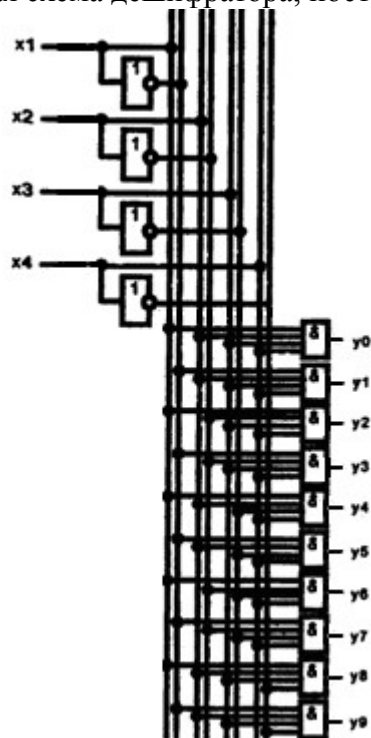
Рис. 22. Таблица истинности работы дешифратора.

x_4	x_3	x_2	x_1	y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

Построим дешифратор в базисе И.

$$\begin{aligned}
 y_0 &= \overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4}; & y_5 &= \overline{x_1} x_2 x_3 \overline{x_4}; \\
 y_1 &= \overline{x_1} x_2 x_3 x_4; & y_6 &= x_1 \overline{x_2} x_3 \overline{x_4}; \\
 y_2 &= x_1 \overline{x_2} x_3 x_4; & y_7 &= x_1 x_2 \overline{x_3} \overline{x_4}; \\
 y_3 &= x_1 x_2 x_3 \overline{x_4}; & y_8 &= x_1 x_2 x_3 x_4; \\
 y_4 &= \overline{x_1} x_2 x_3 x_4; & y_9 &= \overline{x_1} \overline{x_2} x_3 x_4.
 \end{aligned}$$

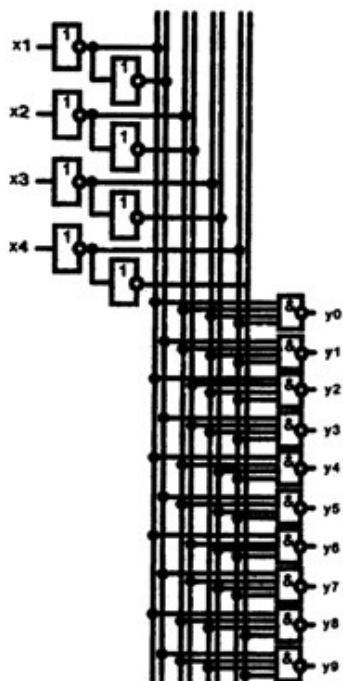
Рис. 23. Структурная схема дешифратора, построенного в базисе И.



Вывод: Если схема дешифратора строится на элементах И, то для любых комбинаций входных сигналов, только на одном выходе присутствует 1, а на остальных 0.

Построим дешифратор в базисе И-НЕ (рис. 24). Такой дешифратор будет иметь прямые входы и инверсные выходы. В полученных логических функциях до и после знака равенства поставим инверсию.

Рис. 24. Структурная схема дешифратора, построенного в базисе И-НЕ.



Вывод: Если схема дешифратора построена на элементах И-НЕ, то для любой комбинации входящих сигналов только на одном выходе будет 0, а на остальных 1.

Задание 10.

Построить дешифратор в базисе 2И-НЕ на основе таблицы истинности изображённой на рисунке 25.

Рис. 25. Таблица истинности работы дешифратора к заданию №10.

x_5	x_4	x_3	x_2	x_1	y
1	1	0	0	0	0
0	1	1	0	0	1
0	0	1	1	0	2
0	0	0	1	1	3
1	0	0	0	1	4
1	0	1	0	0	5
0	1	0	1	0	6
0	0	1	0	1	7
1	0	0	1	0	8
0	1	0	0	1	9

Задание 11.

Построить дешифратор с кодом 7421 на 10 выходов в базисе 2И-НЕ, ИЛИ.

Преобразователь двоично-десятичного кода в код семисегментарного индикатора.

В цифровых устройствах часто возникает необходимость преобразования информации из одной двоичной системы в другую (из одного двоичного кода в другой).

На аппаратном уровне задачу преобразования информации из одного кода в другой выполняют комбинационные устройства - преобразователи кодов.

Преобразователь кода - комбинационное устройство, предназначенное для изменения вида кодирования информации (английское - *converter*).

На принципиальных схемах преобразователи кодов обозначаются X/Y .

Числа на табло и пультах индицируются, как правило, в десятичном коде.

Преобразование двоично-десятичного кода в код семисегментарного индикатора показано на рисунке 26.

При проектировании и конструировании преобразователей кодов можно выделить два подхода:

1) метод, основанный на преобразовании исходного двоичного кода в десятичный и последующем преобразовании десятичного представления в требуемый код;

2) метод, основанный на использовании логического устройства комбинационного типа, непосредственно реализующего данное преобразование.

В первом методе каскадно соединяют дешифратор и шифратор. Сами шифраторы и дешифраторы являются частным случаем преобразователей кодов.

Во втором случае, как для любого комбинационного устройства, составляют таблицу истинности и устанавливают однозначное соответствие между подаваемыми на входы и снимаемыми на выходах комбинациями. Далее проводят синтез логического комбинационного устройства в заданном базисе.

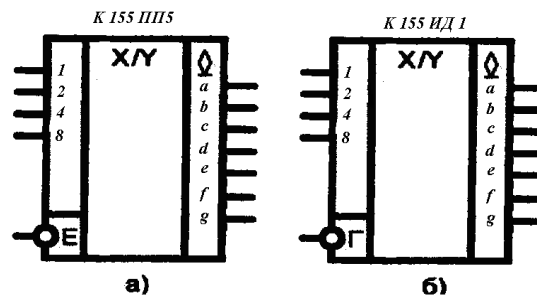
Рис. 26. Таблица преобразования десятичного кода в семисегментный код.

Десятичная цифра	Код 8421				Состояние сегментов						
	X_4	X_3	X_2	X_1	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0



Для формирования цифр и знаков на семисегментных и матричных индикаторах и запуска шкальных индикаторов используют различные преобразователи кодов, иногда неправильно называемые дешифраторами. Существует также микросхема для преобразования двоичного кода в код двоично-десятичный и наоборот.

Рис. 27. Условное графическое изображение преобразователей кодов.



Вход Г (К 155 ИД 1) – служит для гашения сигнала низкого уровня.

Вход Е (К 155 ПП 5) – также служит для гашения индикации при подаче на него логической 1. Индикация осуществляется при подаче логического нуля.

Задание 12.

Построить преобразователь двоично-десятичного кода в код семисегментного индикатора, основываясь на первом подходе к построению преобразователя кодов.

Синтезируемое комбинационное устройство имеет четыре входа и семь выходов, поэтому для каждого из семи выходов получаем:

$$a = (\bar{x}_1 | \bar{x}_2 | \bar{x}_3) | (x_1 | \bar{x}_2 | \bar{x}_3 | x_4),$$

$$b = (\bar{x}_1 | x_2 | x_3) | (x_1 | \bar{x}_2 | x_3), \quad c = \overline{\bar{x}_1 | x_2 | \bar{x}_3};$$

$$d = (\bar{x}_1 | \bar{x}_2 | x_3) | (x_1 | \bar{x}_2 | \bar{x}_3 | \bar{x}_4) | (x_1 | x_2 | x_3),$$

$$e = (x_1 | \bar{x}_2 | x_3) | \bar{x}_1; \quad f = (x_1 | \bar{x}_2 | \bar{x}_3 | \bar{x}_4) | (\bar{x}_1 | x_2 | x_3) | (x_1 | x_3),$$

$$g = (\bar{x}_1 | \bar{x}_2 | x_3) | (\bar{x}_2 | \bar{x}_3 | \bar{x}_4).$$

Полученные формулы позволяют без большого труда построить схему преобразователя кода двоичного в семисегментный.

Сумматоры.

Дискретное сообщение состоит из набора чисел и символов, каждое число состоит из цифр. Способ записи чисел цифровыми знаками, называется системой счисления. В цифровой технике используют позиционные системы счисления. Значение каждой входящей в число цифры зависит от его положения в записи числа. Количество различных цифр, применяемых в позиционной системе счисления, называется основанием системы.

Сумматоры выполняют арифметическую операцию сложения двух чисел.

Они имеют как самостоятельное значение, так и являются составной частью арифметическо-логического устройства (АЛУ). При организации различных вычислительных процессов суммированию отводится главная роль, оно является основной операцией.

Сумматор предназначен для арифметического сложения двоичных чисел. Из принципа сложения многоразрядных двоичных чисел следует, что в каждом i -ом разряде находится сумма трёх чисел по модулю два (слагаемых A_i , B_i и переноса, поступившего из младшего разряда P_i), и формируется сигнал переноса в старший разряд P_{i+1} .

Правила цифрового логического сложения.

1+0=1
0+1=1
0+0=0
1+1=10

10	0	1	0	—	переносы
1	1	0	1	—	первое слагаемое
+	1	0	0	1	— второе слагаемое
<hr/>					
1	0	1	1	0	— сумма

Рис. 28. Условное графическое изображение сумматора.

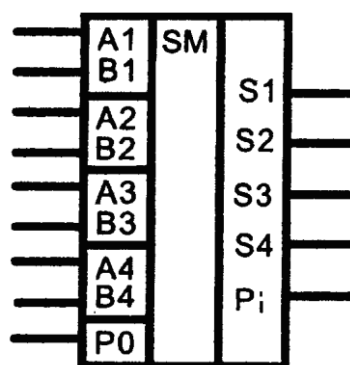


Рис. 29. Таблица истинности одноразрядного сумматора.

Входы			Выходы	
слагаемые		перенос	сумма	перенос
a_i	b_i	p_i	S_i	P_{i+1}
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

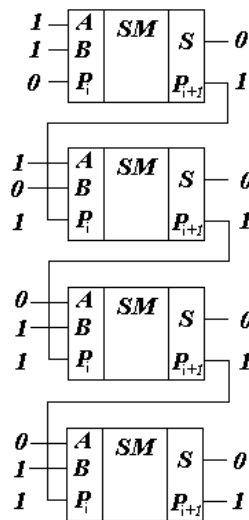
Совершенная дизъюнктивная нормальная форма одноразрядного сумматора.

$$S_i = \bar{a}_i \bar{b}_i \bar{p}_i \vee a_i \bar{b}_i \bar{p}_i \vee \bar{a}_i \bar{b}_i p_i \vee a_i b_i p_i,$$

$$P_{i+1} = a_i b_i \bar{p}_i \vee \bar{a}_i b_i p_i \vee a_i \bar{b}_i p_i \vee a_i b_i p_i.$$

Полные одноразрядные сумматоры являются основами, из которых получают различные схемы многоразрядных сумматоров.

Рис. 30. Структурная схема четырёхразрядного сумматора.



Задание 13.

- 1) Сложите два числа 101 и 110. Постройте соответствующий сумматор, выполняющий сложения этих чисел.
- 2) Полученную сумму сложите с числом 111 и постройте соответствующий сумматор. Расставьте правильно переносы.

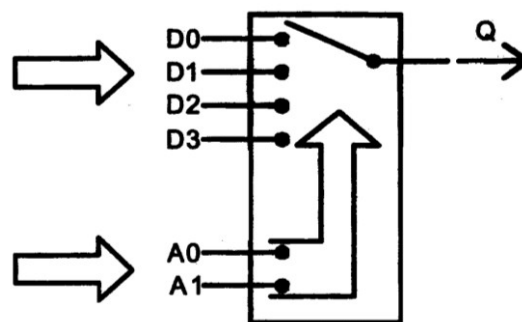
Коммутационные цифровые устройства. Мультиплексоры.

Мультиплексор – это узел, осуществляющий преобразование параллельных цифровых кодов в последовательные. Его применяют для последовательного опроса заданного числа информационных сигналов и передачи их на один выход.

Мультиплексор имеет информационные входы (D_0, D_1, \dots), адресные входы (A_0, A_1, \dots), вход C для подачи стробирующего (синхронного) сигнала и один выход Q .

Символически мультиплексор можно представить многоканальным коммутатором, имеющим одностороннюю передачу данных.

Рис. 31. Символическое изображение мультиплексора.



Каждому информационному входу D_i мультиплексор присваивается номер A , называемый адресом. При подаче стробирующего сигнала на вход C мультиплексор выбирает один из входов D_i , адрес которого, задается двоичным кодом на адресных входах A , и подключает его к выходу Q . Число информационных входов $n_{\text{инф}}$ и число адресных входов $n_{\text{адр}}$ связаны соотношением $n_{\text{инф}} = 2^{n_{\text{адр}}}$.

Рис. 32. Условное графическое изображение мультиплексора.

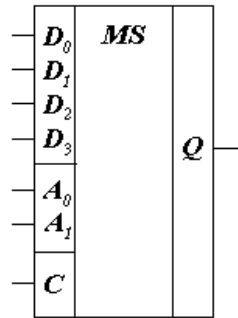


Рис. 33. Структурная схема мультиплексора.

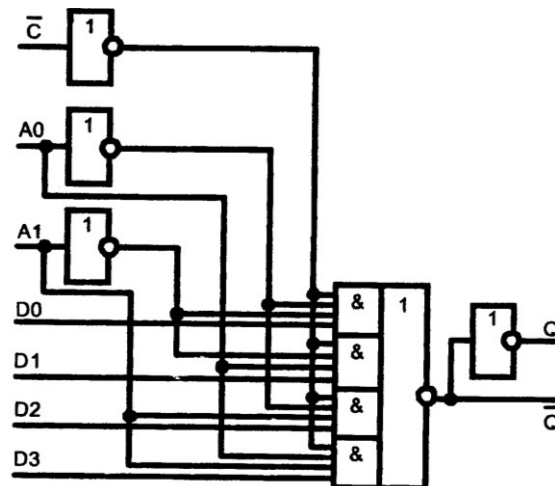


Рис. 34. Таблица истинности мультиплексора

Адресные входы		Стробирующий сигнал	Выход
A_1	A_0	C	Q
X	X	0	0
0	0	1	D_0
0	1	1	D_1
1	0	1	D_2
1	1	1	D_3

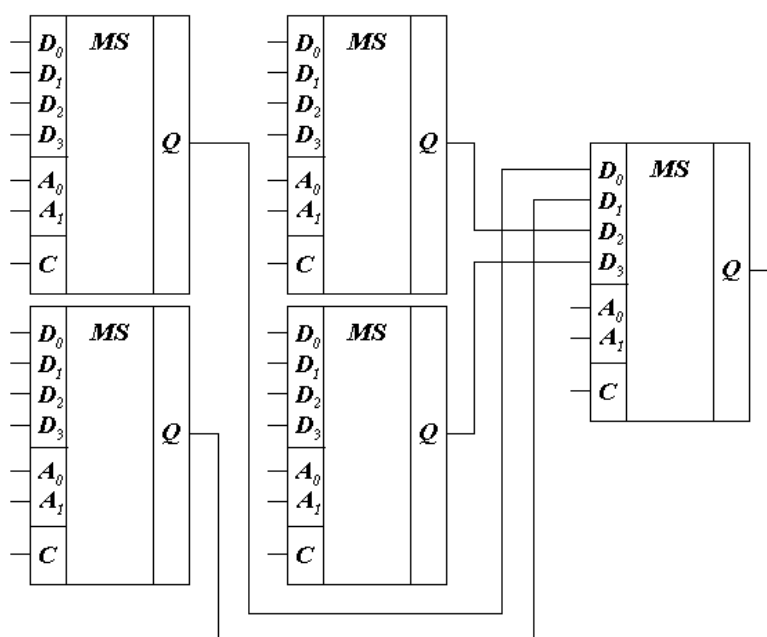
Совершенная дизъюнктивная нормальная форма мультиплексора:

$$Q = C(D_0 \bar{A}_1 \bar{A}_0 \vee D_1 \bar{A}_1 A_0 \vee D_2 A_1 \bar{A}_0 \vee D_3 A_1 A_0)$$

Мультиплексорное дерево. Максимальное число информационных входов в мультиплексорах, выполненных в виде интегральных схем равно 16 (D_0 - D_{15}).

Если требуется построить мультиплексорное устройство с большим числом входов можно объединить мультиплексоры в схему так называемого мультиплексорного дерева. Такое мультиплексорное дерево, построенное на четырёх входовых мультиплексорах, изображено на рисунке.

Рис. 35. Мультиплексорное дерево.



Демультимплексоры.

Демультимплексоры выполняют операцию, обратную операции мультиплексоров - передают данные из одного входного канала в один из нескольких каналов приемников. Демультимплексор имеет один информационный вход и несколько выходов и осуществляет коммутацию входа; к одному из выходов, имеющему заданный адрес (номер).

На рис. 36 показана структурная схема демультимплексора. Она включает в себя дешифратор, выходы которого управляют ключами. В зависимости от поданной на адресные входы кодовой комбинации, определяющей номер выходной цепи, дешифратор открывает соответствующий ключ, и вход демультимплексора подключается к определенному выходу.

Рис. 36. Структурная схема демультимплексора.

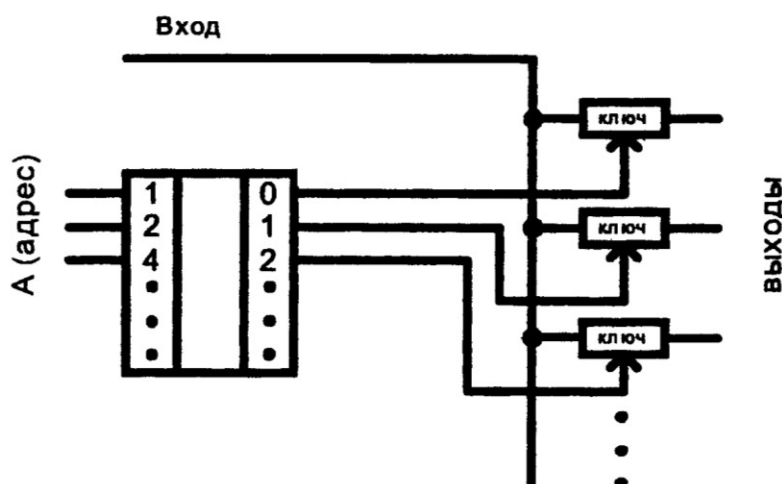
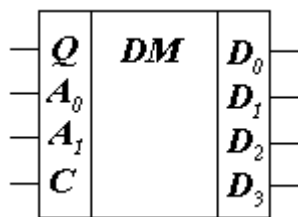


Рис. 37. Условное обозначение демультимплексора.



**Устройства сравнения чисел.
Цифровые компараторы АЛУ**

Цифровой компаратор - комбинационное устройство, предназначенное для сравнения двоичных слов.

Компаратор выполняет следующие действия над двумя двоичными словами:

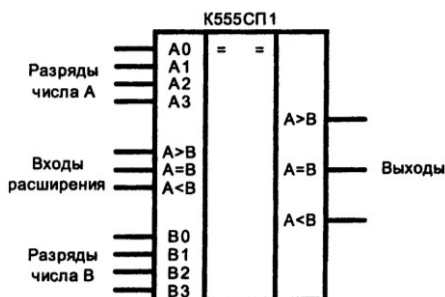
$F(A = B)$ - равенство двоичных слов A и B ;

$F(A > B)$ - слово A больше слова B ;

$F(A < B)$ - слово A меньше слова B

Критерием равенства двух двоичных чисел является совпадение их по всем разрядам.

Рис. 38. Условное обозначение компаратора.



Микросхема K555СП1 имеет четыре сравниваемых входа чисел A и B ($A0, B0, A1, B1, A2, B2, A3, B3$) и три дополнительных входа переноса $A < B, A = B, A > B$ для сравнения чисел большей разрядности путем последовательного соединения компараторов в каскад. Возможно построение многоразрядных компараторов в двоичном коде.

Устройства сравнения на равенство строятся на основе поразрядных операций над одноименными разрядами обоих слов. Слова равны, если равны все одноименные разряды, т. е. если в обоих нули или единицы.

Рис. 39. Таблица истинности компаратора
одноразрядных слов

A	B	$F(A=B)$	$F(A>B)$	$F(A<B)$
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

ПОСЛЕДОВАТЕЛЬНЫЕ ЦИФРОВЫЕ УСТРОЙСТВА.

Интегральные триггеры.

Интегральные триггеры обычно реализуются на логических элементах И-НЕ, ИЛИ-НЕ.

Легко убедиться, что каждый из этих элементов характеризуется некоторым логическим уровнем, который определяет полностью логический уровень на выходе.

Такие логические уровни, действующие на одном из входов элемента, однозначно задают логический уровень на выходе, называют активным логическим уровнем.

Уровни обратные активным логическим уровням, называют пассивными логическими уровнями.

Назначение триггера.

Триггер – это устройство, предназначенное для хранения значения одной логической переменной. При хранении многоразрядных двоичных чисел для запоминания значения каждого разряда числа, используется отдельный триггер (пяти разрядное число – пять триггеров). Триггер может находиться в одном из двух состояний (сост. 1 и сост. 0).

Основные обозначения.

Триггер имеет два выхода: прямой и инверсный (Q, \bar{Q}).

Уровнями напряжения на этих выхода определяется состояние триггера (0,1).

Триггер – это простейший цифровой автомат с памятью, способный хранить 1 бит (binary digit – двоичный разряд) информации.

Триггеры имеют различные типы входов:

S – раздельный вход установки триггера в состояние 1,

R – раздельный вход установки триггера в состояние 0,

J – вход установки универсального триггера в сост. 1,

K – вход установки универсального триггера в сост. 0,

D – информационный вход установки триггера в состояние, соответствующее логическому уровню на этом входе.

C – управляющий синхронизирующий вход.

По характеру реакций на входах, сигналы триггера делятся на два типа: синхронные и асинхронные.

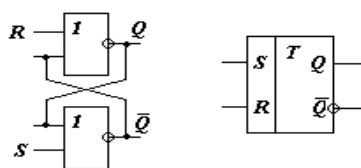
Асинхронные триггеры характеризуются тем, что входные сигналы действуют на состояние триггера непосредственно с момента подачи их на входы. В синхронных триггерах только при подачи синхронизирующего сигнала на управляющий вход C.

RS-Триггер с прямым входом

RS-триггер – это триггер с раздельной установкой состояний логического нуля и логической единицы (с раздельным запуском). Он имеет два информационных входа S и R. По входу S триггер устанавливается в состояние $Q = 1$ ($\bar{Q} = 0$), по входу R – в состояние $Q = 0$ ($\bar{Q} = 1$).

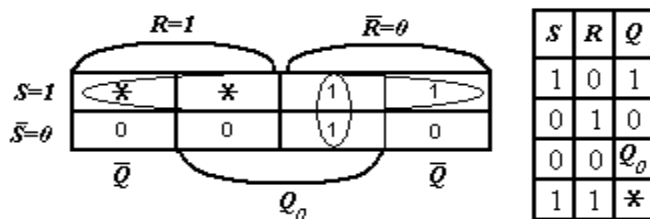
Асинхронные RS-триггеры. В асинхронных триггерах срабатывание происходит непосредственно в момент изменения сигнала на информационных входах. Асинхронные RS-триггеры являются наиболее простыми. В качестве самостоятельного устройства используется редко, но являются основой для построения более сложных систем.

Рис. 40. Условное обозначение и структурная схема RS-Триггера с прямыми входами.



В зависимости от логической структуры асинхронные RS-триггеры бывают с прямыми либо инверсными входами.

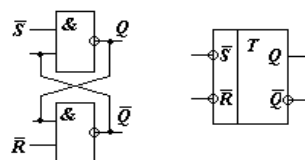
Рис. 41. Карта Вейча и таблица истинности работы RS-Триггера



Переключение триггера из одного активного состояния в другое происходит при подаче активных сигналов на входы.

RS-Триггер с инверсным входом.

Рис. 42 Условное обозначение и структурная схема



RS-Триггера с инверсными входами.

Здесь используются элементы «И-НЕ». Активным логическим уровнем на входах является уровень логического нуля, пассивным логическим уровнем – единица. При $S = 1$ ($\bar{S} = 0$) на выходах будет действовать активный логический уровень – ноль.

Синхронные триггеры со статическим управлением.

Отличие синхронного триггера от асинхронного состояния в том, что синхронный триггер снабжён дополнительным входом, называемым синхронизирующим.

Назначение синхронного входа в том, что сигналом на этом входе разрешается приём сигналов с информационных входов в заданные временные интервалы.

Синхронные триггеры в свою очередь делятся на два типа: синхронные со статическим управлением и синхронные триггеры с динамическим управлением. Первый триггер реагирует на изменение сигнала на информационных входах, происходящие во время действия в сигналы на синхронном входе. В триггере с динамическим управлением приём сигналов с информационных входов происходит в течении малой длительности фронта (+, -), (\uparrow , \downarrow).

Рис. 43 Диаграмма синхронного триггера с динамическим управлением (положительный фронт).

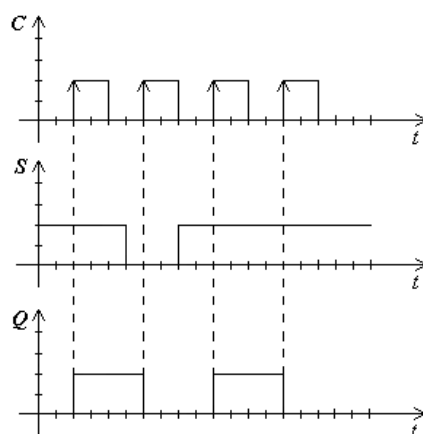
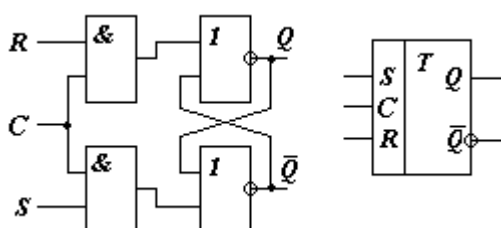


Рис. 44 Логическая структура и условное графическое изображение синхронного RS-триггера.



Задание 14.

1) Составить СДНФ, карту Вейча. Построить схему согласно следующей таблице истинности:

x_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
f	0	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0

Д-Триггеры.

Этот тип триггера имеет один информационный вход D и управляющий вход C, который служит для подачи синхроимпульса. Функционирование D-триггера описывается таблицей истинности. На рисунке 43 представлена логическая структура D-триггера и его таблица истинности. Как видно при $C=1$ D-триггер устанавливается в состояние, определяемое логическим уровнем на входе триггера. При $C=0$, он сохраняет ранее установленное состояние Q_0 .

Поскольку смена информации в D-триггере происходит по синхросигналу, то этот триггер может быть только синхронным. Его также называют тактируемым. Для устойчивой работы D-триггера необходимо, чтобы в течении синхроимпульса информация на входе была неизменной, т.е. необходим защитный интервал времени, равный длительности синхроимпульса.

Рис. 45 Логическая структура и таблица истинности D-триггера.

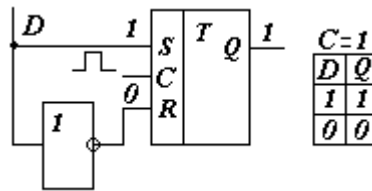


Рис. 46 Условное графическое обозначение D-триггера.

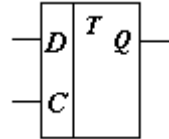


Рис. 47 Временная диаграмма D-триггера с синхронным динамическим входом.

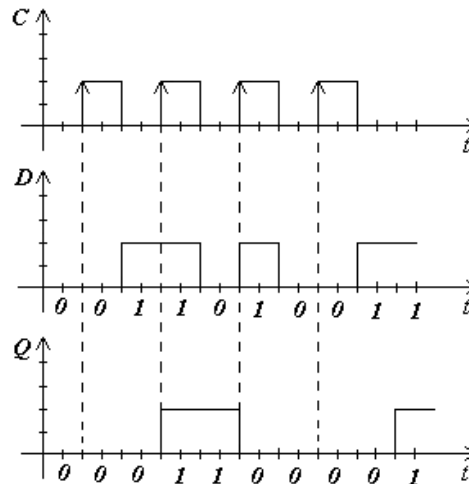
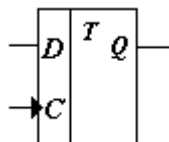


Рис. 48 Условное графическое обозначение D-триггера с синхронным динамическим управлением (при возрастающем импульсе).



Счётный Т-триггер.

Т-триггер – это триггер со счётным входом. Он имеет один информационный вход. При приходе активного сигнала Т-триггер меняет своё состояние на противоположное и сохраняет предыдущее значение при отсутствии сигнала на входе.

Для обеспечения такого режима функционирования необходимо обеспечить обратную связь с выхода на вход. Соответственно, Т-триггер можно построить по схемам RS и D-триггеров.

Выходной элемент памяти одновременно является и приёмником новой информации, и источником (памятью) предыдущей.

Триггер принимает новую информацию и сразу же передаёт её по цепи обратной связи на вход.

Т-триггер широко используется в различных цифровых устройствах в качестве делителей частоты и двоичных счётчиков.

Рис. 49 Структурная схема Т-триггера.

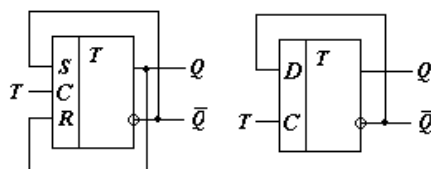
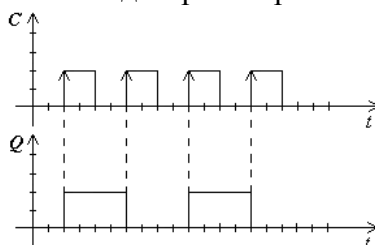


Рис. 50 Временная диаграмма работы Т-триггера.



Задание 15.

Дано: На вход D-триггера подано двоичное слово:
001100110011101

- 1) Построить временную диаграмму D-триггера при возрастающем импульсе с динамическим управлением.
- 2) Построить временную диаграмму D-триггера при возрастающем импульсе с статическим управлением.

Примечание: период импульса в два бита (две тетрадные клетки).

Триггеры с двухступенчатой структурой (JK-Триггеры).

Особенность JK-триггеров с двухступенчатой структурой состоит в том, что они содержат два триггера: ведущий и ведомый. Оба триггера функционируют как синхронные триггеры со статическим управлением. Если на синхронизирующем входе $C=1$, ведущий триггер устанавливается в состояние, соответствующее сигналам, поступающим на его вход. Он продолжает находиться в состоянии, в которое ранее был установлен. При изменении значения синхроимпульса с $C=1$ в $C=0$, ведущий триггер отключается от информационных входов и перестаёт реагировать на изменения значения сигнала на этих входах. Ведомый триггер устанавливается в состояние, в котором находится до этого ведущий триггер. Таким образом, управление процессами в триггере с двухступенчатым запоминанием информации за время тактового периода осуществляется двумя фронтами сигнала на C -ходе. На положительном фронте происходит установка ведущего триггера на отрицательном - ведомого.

□ Q_0 – предыдущее состояние

Рис. 51 Таблица истинности JK-триггера.

J	K	C	Q
1	0	1	1
0	1	1	0
0	0	1	Q_0
1	1	1	\bar{Q}_0

Рис. 52 Логическая структура JK-триггера.

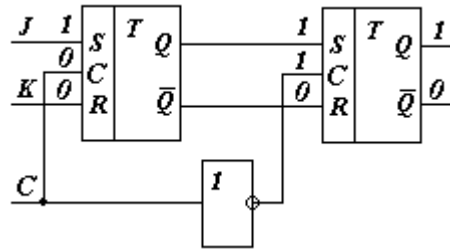
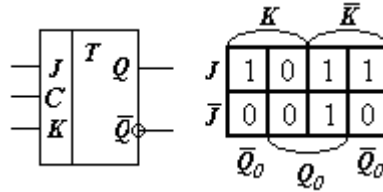
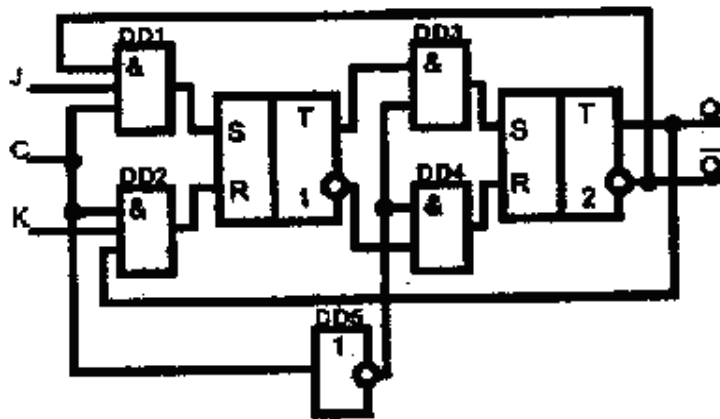


Рис. 53 Условное графическое изображение и Карта Вейча JK-триггера.



МДНФ: $Q = \bar{K}Q_0 \vee J\bar{Q}_0$

Рис. 54 Логическая структура универсального JK-триггера.



Построение D-триггера с помощью JK-триггера.

Рис. 55 Структурная схема D-триггера, построенного с помощью JK-триггера

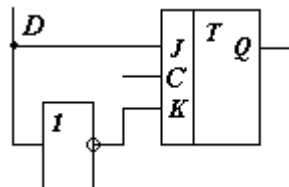
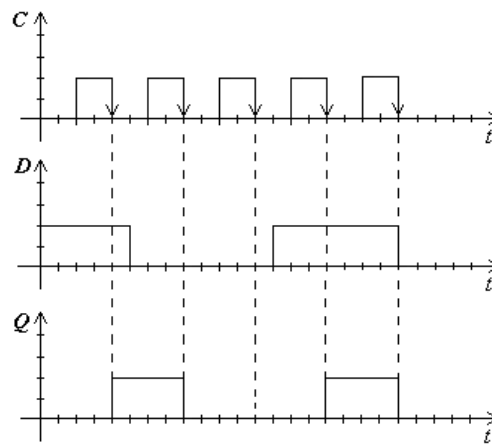
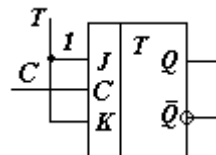


Рис. 56 Временная диаграмма D-триггера, построенного с помощью JK-триггера



Построение Т-триггера с помощью JK-триггера

Рис. 57 Структурная схема Т-триггера, построенного с помощью JK-триггера



Построение Т-триггера с помощью D-триггера

Рис. 58 Структурная схема Т-триггера, построенного с помощью D-триггера

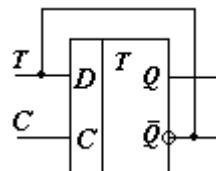
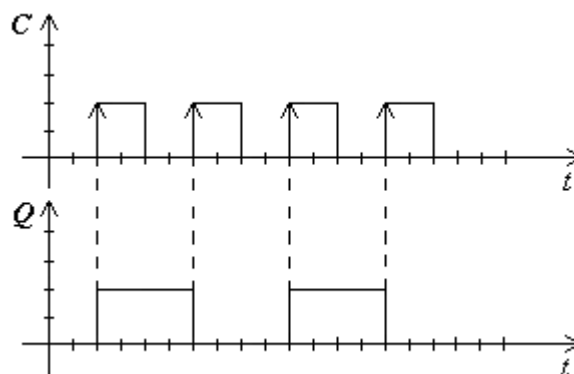


Рис. 59 Временная диаграмма Т-триггера, построенного с помощью D-триггера



Интегральные регистры.

Основная функция регистров – хранение одного многоразрядного числа, при этом число должно быть представлено в двоичной системе счисления. Регистр, для хранения n -разрядного двоичного числа должен содержать n количество триггеров. Регистры могут использоваться для выполнения некоторых других функций: сдвиг хранимого в регистре числа на определённое число разрядов в лево или в право, преобразование из последовательной формы числа в параллельную форму и наоборот. В зависимости от формы представления число вводимого в регистр различают два типа регистров: параллельные и последовательные.

В регистрах выполняются следующие операции:

- ввод и вывод хранимой информации;
- хранение информации;
- сдвиг хранимой информации вправо либо влево на определенное число разрядов;
- преобразование кода числа из последовательного в параллельный и наоборот.

По способу приема информации регистры подразделяются:

- параллельные - информация записывается и считывается только в параллельной форме;
- последовательные, которые также называются сдвигающие (сдвиговые) - запись и считывание информации происходит только в последовательной форме;
- последовательно-параллельные, такие универсальные регистры можно использовать для преобразования кода из параллельного в последовательный и наоборот.

По числу каналов передачи информации регистры подразделяются:

- однофазные - информация вводится либо в прямом, либо в обратном коде;
- парафазные - одновременно и в прямом, и в обратном кодах.

По способу тактирования (синхронизации) регистра подразделяют:

- одноктактные, синхронизируемые одной управляющей последовательностью;
- двуктактные;
- многотактные, управляемые несколькими последовательностями импульсов.

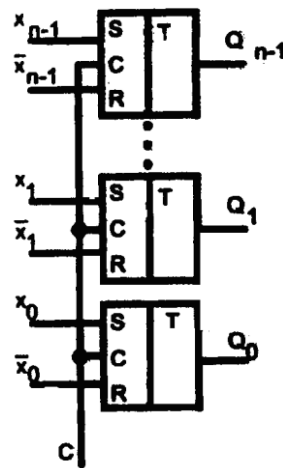
Сдвиговые регистры могут быть однонаправленные, когда записанную информацию сдвигают только в одном направлении, и двунаправленные, в которых информация сдвигается как вправо, так и влево. Для таких регистров предусматривается специальный вход включения режима направления сдвига.

Параллельный регистр.

В параллельных регистрах прием и выдача двоичных слов осуществляется по всем разрядам одновременно. Поэтому триггеры, соответствующие разным разрядам не связаны между собой. Каждый триггер в параллельном регистре имеет свои независимые входы и выходы. Тактовые входы всех триггеров соединены между собой.

На рис. 61 показан параллельный регистр, построенный на D -триггерах: одноктактный регистр с однофазным входом. В таком регистре при уровне логической 1 на C -синхровходе все триггеры устанавливаются в состояния, определяемые действующим на D -входах входными сигналами. Этот триггер имеет четыре однофазных прямых входа $D0, D1, D2, D3$, четыре соответствующих выхода $Q0, Q1, Q2, Q3$ и вход синхронизации C .

Рис. 60. Параллельный регистр, построенный на RS-триггерах



Параллельные регистры могут иметь дополнительно вспомогательные комбинационные схемы, которые расширяют функциональные возможности. Они используются для установки начального состояния, режима приема, хранения либо выдачи информации, реализуют запись с двух направлений и т.д. В этом случае регистр имеет соответствующие входы.

Рис. 61 Условное обозначение параллельного регистра на D-триггерах

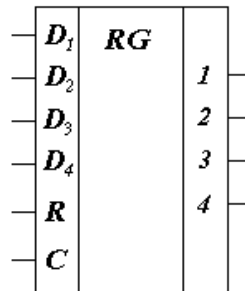


Рис. 62 Временная диаграмма работы параллельного регистра на D-триггерах.

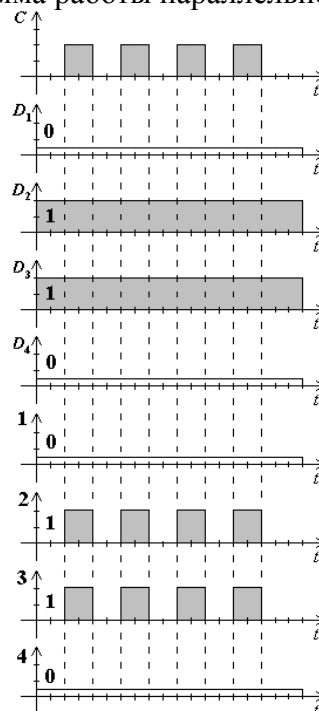
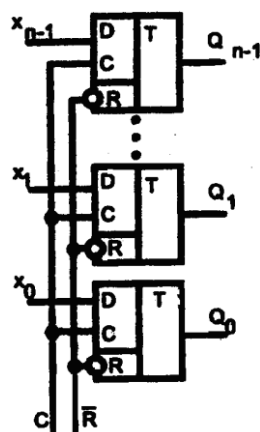


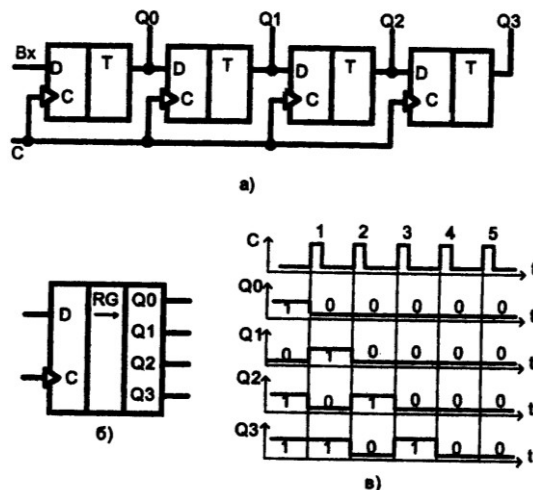
Рис. 63 Структурная схема параллельного регистра на D-триггерах.



3.2.2. Сдвиговые регистры

В сдвиговом (последовательном) регистре триггеры соединены последовательно, т.е. выходы предыдущего триггера передают информацию на входы последующего. Простейший однофазный сдвиговой регистр на D -триггерах изображен на рис. 64.

Рис. 64. Простейший однофазный сдвиговой регистр на D -триггерах (а – структурная схема, б – условное графическое изображение, в – временная диаграмма).



Суть сдвига состоит в том, что логическая переменная (1 либо 0), записанная в i -разряде, передается в соседний справа ($t+1$) разряд. В крайний левый разряд заносится число, подаваемое в него входа. Из крайнего правого разряда двоичный код последовательно поступает из регистра во внешнюю цепь.

Помимо основного предназначения (хранения двоичного кода, преобразования, сдвига и передачи на выход) регистры также используются для выполнения и других функций. На их основе строятся такие устройства: линии задержки на заданное число тактов, накапливающие сумматоры, формирователи импульсов большей длительности, генераторы псевдослучайных последовательностей. Регистры применяют в арифметическо-логических устройствах в качестве узлов, выполняющих различные логические операции.

Интегральные счётчики.

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

Счет импульсов является одной из наиболее распространенных операций в устройствах дискретной обработки информации.

Основное функциональное назначение счетчиков:

- счет импульсов, поступивших на вход;
- деление частоты.

Счетчики относятся к устройствам с памятью. Строятся счетчики, как и регистры, на основе одноступенчатых связанных между собой триггеров. Наиболее часто используются *T*- и *JK*-триггеры.

Комбинационные элементы в счетчиках используются для управления работой триггеров. Число триггеров определяет максимальное количество импульсов, которое может быть подсчитано счетчиком.

В счетчиках выполняются следующие логические операции:

- установка в нулевое состояние (сброс);
- запись входной информации в параллельной форме - начального кода, с которого начинается счет;
- хранение записанной информации;
- выдача хранимой информации в параллельной форме;
- инкремент - увеличение хранящегося числа на единицу;
- декремент - уменьшение хранящегося числа на единицу.

Основным параметром счетчика является коэффициент счета $K_{сч}$, равный максимальному числу импульсов, после которых счетчик устанавливается в исходное состояние (обнуляется) и начинается новый цикл работы счетчика.

Если $K_{сч}=2^n$ – то счётчик является двоичным, если $K_{сч} \neq 2^n$ – то счётчик не двоичный, если $K_{сч}=10^n$ – то счётчик является десятичным, где n – количество триггеров в счётчике.

Простейший счётчик – это *T* триггер, который считает от 0 до 1. Любой счётчик есть делитель частоты. В наших часах стоит кварц с коэффициентом счёта 2^{25} Гц.

В счётчике поступление на вход очередного уровня вызывает увеличение на единицу, хранящегося в счётчике числа. Таким образом, в счётчике устанавливается число, которое получается путём суммирования предыдущего значения с 1. Это суммирование производится по обычным правилам сложения двоичных чисел.

$$\begin{array}{r}
 1110 \\
 + \quad 1 \\
 \hline
 1111
 \end{array}
 \quad
 \begin{array}{r}
 1111 \\
 + \quad 1 \\
 \hline
 10000
 \end{array}
 \quad
 \begin{array}{r}
 1110 \\
 - \quad 1 \\
 \hline
 1101
 \end{array}
 \quad
 \begin{array}{r}
 1111 \\
 - \quad 1 \\
 \hline
 1110
 \end{array}$$

В процессе суммирования имеет место следующие особенности:

1. Если цифра некоторого разряда остаётся неизменными, либо изменяется с 0 на 1, т.е. при этом цифры более старших разрядов меняются.

2. Если цифра некоторого разряда изменяется с 1 на 0, то происходит инвертирование цифры, следующие за ним более старшего разряда.

В вычитающем счётчике поступление на вход очередного уровня (лог. 1) вызывает уменьшение хранившегося в счётчике числа на 1. Если в младшем разряде числа содержится 1, то получается в результате вычитание ед. число, отличается от исходного лишь в младшем разряде.

Рис. 65. Условное графическое изображение счётчика

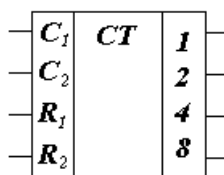


Рис. 66. Двоичный счётчик (а – условное графическое изображение, б – структурная схема).

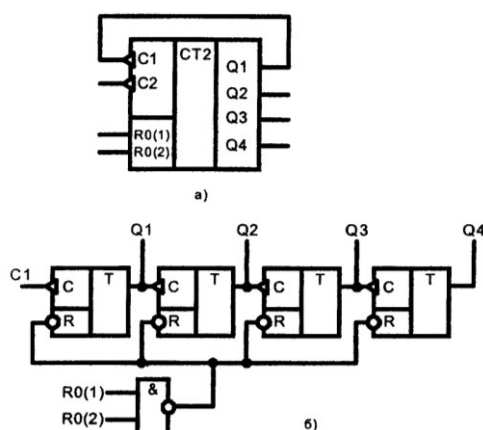


Рис. 67. Временная диаграмма двоичного суммирующего счётчика

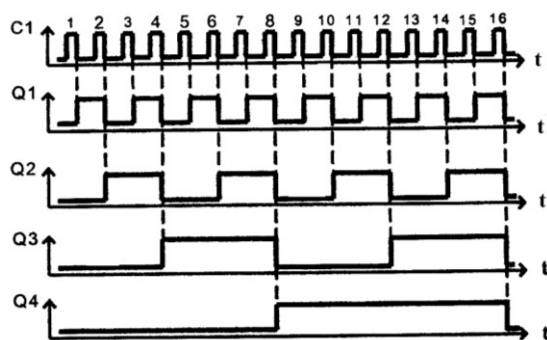


Рис. 68. Таблица истинности четырехразрядного счетчика

Число поступивших импульсов		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Выходы	Q1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	Q2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
	Q3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
	Q4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0

Пример 6.

Выполнить синтез вычитающего счетчика с циклом счета $K_4=6$, построенного на ЖТ:

Решение:

В процессе решения необходимо выполнить следующие пункты:

1) Записать таблицу выходных значений ЖТ-триггеров, входящих в счётчик на основе таблицы переходов счетчика. Существуют следующие таблицы переходов:

Рис. 69. Таблица переходов счетчика (а - D-триггера, б – ЖТ-триггера).

D	
0	0 → 0
1	0 → 1
1	1 → 0
0	1 → 1

а

J	K	
0	1	0 → 0
1	-	0 → 1
-	1	1 → 0
-	0	1 → 1

б

- 2) Исходя из вышеуказанных таблиц и таблицы истинности работы трёхразрядного счётчика можно без особого труда записать таблицу выходных значений JK-триггеров.

Рис. 70 Таблица истинности (исходная).

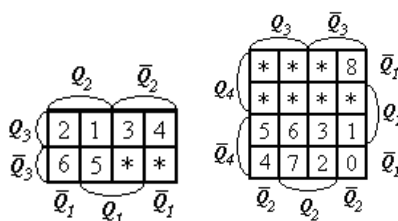
N	Q3	Q2	Q1
0	1	1	1
1	1	1	0
2	1	0	1
3	1	0	0
4	0	1	1
5	0	1	0

- 3) Исходя из вышесказанного получаем следующую таблицу истинности выходных значений счетчика:

Рис. 70 Таблица истинности выходных значений триггеров.

J1	K1	J2	K2	J3	K3
-	1	-	0	-	0
1	-	-	1	-	0
-	1	0	-	-	0
1	-	1	-	-	1
-	1	-	0	0	-
1	-	-	0	1	-

- 4) Далее проводим синтез устройства и минимизацию с использованием Карт Вейча (так как у нас $K_{сч}=6$, то триггеров будет три). Заполняем карты Вейча по следующей схеме, где цифра обозначает номер порядка (строки) таблицы истинности рисунка 70.



- 5) В результате получаем 4 карты Вейча, параметры которых будут: $J_1=K_1=1$, $J_2=J_3=\neg Q_1$, $K_2=Q_3 \neg Q_1$, $K_3=\neg Q_1 \neg Q_2$.
- 6) На основе полученных параметров строим схему счетчика, при этом учитываем следующий порядок: соединение синхронных входов, далее параметры 1 и далее,

выходы с каждого триггера должны выходить вверх. Если входной параметр описывается двумя или несколькими выходными параметрами, то строятся обратные связи в виде конъюнкторов.

Задание 16.

- 1) Выполнить синтез суммирующего счетчика с $K_{сч}=9$, построенного на D-триггерах.

Задание 17.

- 2) Выполнить синтез суммирующего четырёхразрядного двоичного счетчика с $K_{сч}=9$, построенного на JK-триггерах.

Классификация запоминающих цифровых устройств.

Для хранения небольших массивов кодовых слов могут использоваться регистры. Для хранения информации в 1000 и более бит, применение регистров нерационально. В этом случае используют запоминающие устройства (ЗУ). Различают следующие типы ЗУ:

- 1) Операционное запоминающее устройство (ОЗУ)
- 2) Постоянное запоминающее устройство (ПЗУ)
- 3) Перепрограммируемое запоминающее устройство (ППЗУ)

ОЗУ используется в условиях, когда необходимо выбирать и обновлять информацию в высоком темпе работы процессора цифрового устройства. ОЗУ предусматривает 3 режима работы:

- 1) Режим чтения
- 2) Режим хранения
- 3) Режим записи.

Обычно время чтения и записи слова в ОЗУ составляет доли микросекунд. ОЗУ используется для хранения исходных данных промежуточных и конечных результатов обработки данных и программ.

ПЗУ предназначена для хранения однажды записанной информации, не нарушаемой при отключении источников питания. В ПЗУ предусматривается 2 режима работы:

- 1) Режим чтения
- 2) Режим хранения

ПЗУ используется для хранения программ в таких специализированных цифровых устройствах, которые функционируют длительное время, многократно выполняют действия по одному и тому же алгоритму.

ППЗУ в процессе функционирования цифровых устройств, используются как постоянное запоминающее устройство. Оно отличается от ПЗУ тем, что допускает обновление однажды записанной в него информации. Однако в отличие от ОЗУ запись требует отключения ППЗУ от цифрового устройства. Запись производится с помощью специального устройства – программатора. ЗУ содержит некоторое количество информации, которое хранится в ячейках, каждая ячейка n -разрядная. Ячейки нумеруются двоичными числами. Номер ячейки называется адресом. Если для представления адреса используется комбинация n -разрядного двоичного кода, то число ячеек в ЗУ равно 2^n . Количество информации, которое может храниться в ЗУ, определяет его ёмкость. Ёмкость измеряется в битах, в байтах и т.д.

Быстродействие ЗУ характеризуется двумя параметрами: временем выборки (f_b), представляющим собой интервал времени между моментом подачи сигнала выборки и появлением считанных данных на выходе, а также циклом записи (f_3), определяемым минимально допустимым временем между моментом подачи сигнала выборки при записи и моментом. Когда допустимо следующие обращение к памяти.

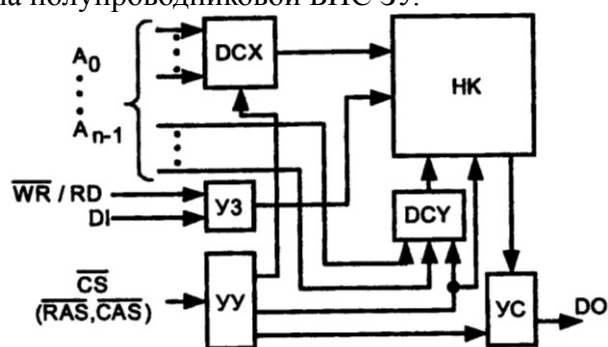
В цифровых системах ЗУ подразделяются на внешние и внутренние. Внешние ЗУ служат для хранения больших объемов информации: запасов данных и программного обеспечения системы, но работают со сравнительно низкой скоростью. Внешние ЗУ характерны неопределенно длительным сохранением информации при отсутствии питания. Такие ЗУ могут быть реализованы на основе самых разнообразных физических принципов и элементов.

Из внешних ЗУ отдельные массивы информации, подлежащие текущей обработке, поступают во внутренние ЗУ, которые служат для хранения данных, используемых при реализации решения задачи или ее части. Это полупроводниковые ЗУ. Чтобы обеспечить высокую производительность системы, быстродействие внутренних ЗУ должно быть близким к быстродействию операционного и управляющего устройств.

Полупроводниковые ЗУ являются одним из основных типов в силу следующих особенностей.

1. Широкий диапазон быстродействия и функционального назначения.
2. Конструктивная, электрическая и технологическая совместимость с другими микросхемами.
3. Отсутствие необходимости применения мощных формирователей токов для записи и считывания информации, а также сложных высокочувствительных усилителей воспроизведения.
4. Высокая надежность и возможность обеспечения высокой информативной плотности.
5. Технологичность.

Рис. 71 Типовая схема полупроводниковой БИС ЗУ.



Она состоит из следующих узлов: матрицы - накопителя (НК), дешифраторов строк и столбцов (DCX , DCY), устройства записи ($УЗ$), устройства считывания ($УС$), устройства управления ($УУ$). В зависимости от типа ЗУ те или иные типовые узлы могут в схеме отсутствовать. Основным узлом БИС ЗУ является матрица памяти - накопитель, представляющий собой совокупность элементов памяти. Элемент памяти может хранить один разряд числа, т.е. бит информации. Элементы памяти размещаются по строкам m и столбцам n , так что их общее число равно произведению $N=mn$. Местоположение элемента памяти (ЭП) определяется двумя координатами X и Y . Для обращения к нужному элементу памяти (выборки ЭП) сигналами единичного уровня возбуждаются адресные линии строки и столбца, на пересечении которых находится данный ЭП. На всех остальных адресных шинах должны быть сигналы нулевого уровня. Такая система адресации информации (выборки ЭП) при обращении к накопителю получила название матричной. Кроме матричной организации существует словарная и комбинированная. Формирование сигналов выборки производится дешифратором кода адреса. После того, как ЭП выбран, можно либо записать в него информацию, либо считать.

Входные информационные сигналы DI поступают в устройство записи $УЗ$, которое служит для записи информации в ЭП. Выходные информационные сигналы DO считываются из БИС ЗУ через устройство считывания $УС$. Управляющие сигналы

$\overline{CS}, \overline{RAS}, \overline{CAS}, \overline{WR}/RD, \overline{CS}$ поступают в УУ и УЗ и определяют режим работы БИС ЗУ (запись, хранение, считывание информации).

Микросхемы статических ОЗУ имеют, как правило, матричную структуру с двухкоординатной системой адресации (выборки). Матричная структура накопителя и двухкоординатная система выборки обеспечивают возможность доступа к каждому элементу памяти.

Всякое ОЗУ состоит из двух основных частей: накопителя и схемы управления или, как говорят, периферии. Периферия предназначена для ввода и вывода данных, в нее входят дешифраторы, усилители, регистры, разного рода ключи, коммутаторы и другие схемы общего назначения.

Накопитель - основная часть ОЗУ, где хранятся данные (двоичные коды), он состоит из элементов памяти (запоминающих ячеек), каждый из которых хранит один бит информации (0 или 1). Элементы памяти являются бистабильными запоминающими ячейками, основным свойством которых является наличие двух устойчивых состояний $Q = 1$ или $Q = 0$.

На рис. 72 показана типичная матричная организация ОЗУ, где отдельный элемент памяти ЭП расположен в узлах решетки, образованной адресными шинами X и Y . Количество ячеек равно произведению количества горизонтальных шин на количество вертикальных (например, $4 \times 4 = 16$ ячеек).

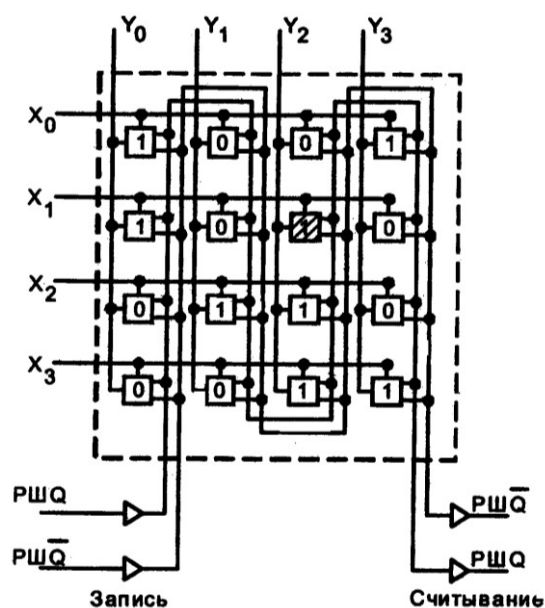
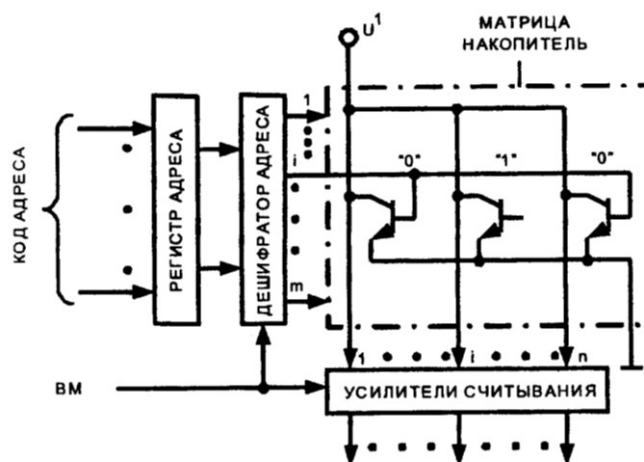


Рис. 72. типичная матричная организация ОЗУ

Рис. 73 Структурная схема ПЗУ на биполярных транзисторах.



СИСТЕМЫ УПРАВЛЕНИЯ

Программируемые логические матрицы.

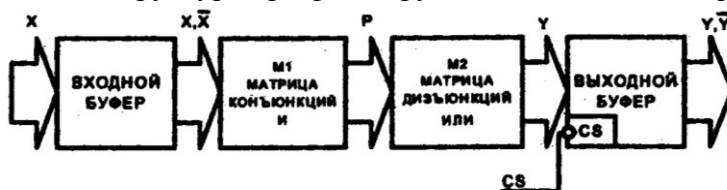
Развитие интегральной схемотехники от микросхем малой и средней степени интеграции к БИС/СБИС привело к противоречию. Специализированные БИС обладают повышенной надежностью, значительно снижается масса и габариты, возрастает быстродействие. Но проектирование специализированных БИС/СБИС требует больших затрат средств и времени.

Разрешением этого противоречия явились полузаказные БИС/СБИС, которые выпускаются универсальными, имеют общую универсальную структуру. Дальнейшее программирование путем изменения внутренних физических свойств делает их специализированными, способными выполнять специфические логические операции по заказу потребителя.

Представителями данного направления являются программируемые логические матрицы (ПЛМ), программируемая матричная логика (ПМЛ) и базовые матричные кристаллы (БМК).

Основой ПЛМ служат последовательно соединенная матрица конъюнкций М1 (матрица элементов И) и матрица дизъюнкций М2 (матрица элементов ИЛИ) (рис. 74).

Рис. 74. Структура программируемой логической матрицы



Для реализации системы функций алгебры логики на ПЛМ необходимо проводить минимизацию. Поиск кратчайших дизъюнктивных форм целесообразно проводить до тех пор, пока число конъюнкций (термов) становится равным l - параметру имеющихся в ПЛМ конъюнкций. При этом учитывается наличие одинаковых конъюнкций в связанных логических функциях. В результате уменьшения числа конъюнкций интегральные схемы ПЛМ и ПМЛ по сравнению с ППЗУ при одинаковой площади кристалла способны реализовать логические функции большого числа переменных.

После минимизации и получения кратчайшей дизъюнктивной формы системы логических функций составляется таблица и осуществляется программирование. Программирование выполняется пользователем на специальном оборудовании - программаторах.

a11	a21	a31	a41
a12	a22	a32	a42
a13	a23	a33	a43
a14	a24	a34	a44

i – номер столбца

j – номер строки

a_{ij} – ячейка матрицы

Матрица – это поименованный массив информации.

Общие сведения о микропроцессорах и микропроцессорных системах.

Команды процессора. В процессоре предусматривается выполнение большого количества операций. Любая из этих операций представляет собой простейшие арифметическое или логическое устройство (сложение, вычитание, дизъюнкция и др.). При этом в каждой операции участвует не более двух операндов, поэтому предварительно решаемая задача должна быть представлена последовательностью выполняемых операций. На каждую выполняемую операцию должна быть составлена команда. Совокупность команд называется программой. Команда должна содержать все необходимые для выполнения операций указания: вид операции, место, где хранятся операнды данной операции и куда должен быть помещён результат. В любой ЭВМ предусматривается, определённа система кодирования операций. Например, 0,1 – сложение, 10-вычитание и т.д.

KO	A1	A2	A3
----	----	----	----

Трёхадресный формат команды

KO – код операции, A – адрес ячейки.

В ЭВМ могут также использоваться двухадресные команды. В этом случае результат операции помещается в ячейку одного из операндов или остаётся в АЛУ.

KO	A1	A2
----	----	----

Двухадресный формат команды

Существуют также и одноадресные команды. При таком формате для выполнения одного арифметического действия над двумя числами от машины может потребоваться выполнения нескольких одноадресных команд.

KO	A1
----	----

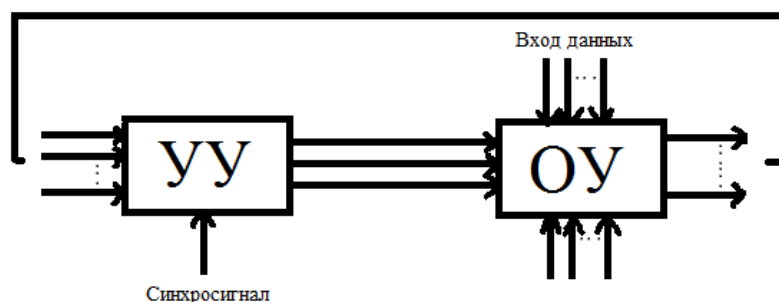
Одноадресный формат команды

Например, для сложения двух чисел необходимо будет выполнить три команды:

1. Ввод в АЛУ число, хранящиеся в ОЗУ или ПЗУ по приведенной в команде адреса.
2. Прибавить к принятому числу число, хранящиеся по адресу, указанному в команде.
3. Поместить полученный в АЛУ результат в память по адресу, указанному в команде.

Структура процессора. Процессор осуществляет непосредственную обработку данных и программное управление обработки данных. Он синтезируется в виде соединения двух устройств: операционного устройства (ОУ) и управляющего устройства (УУ).

Рис. 75. Структура процессора



ОУ устройство, в котором выполняются операции, оно включает в себя в качестве узлов регистры, сумматоры, шифраторы, мультиплексоры и демультиплексоры для коммутации каналов.

УУ координирует действие узлов ОУ, оно вырабатывает в некоторой временной последовательности управляющие сигналы, под действием которых в узлах ОУ выполняются действия.

Процесс функционирования ОУ распадается на последовательность элементарных действий в его узлах.

- 1) Установка программы в некоторое состояние
- 2) Инвертирование содержимого разрядов программы
- 3) Пересылка содержимого одного узла в другой
- 4) Сдвиг содержимого узла влево или вправо
- 5) Счёт, при котором число в счетчике возрастает или убывает на 1
- 6) Сложение
- 7) Сравнение содержимого программы на равенство с некоторым числом, результат сравнения.
- 8) Некоторые дизъюнкции, конъюнкции и др.

Каждое такое элементарное действие называется микрооперациями (выполняются в течении одного тактового периода). Совокупность микроопераций называется микрокомандой.

Набор микрокоманд, предназначенных для выполнения некоторой задачи. Называется микропрограммой.

Два подхода к построению процессора. Существует два подхода к проектированию микропрограммного автомата:

1. Использование схемной логики.

В процессе проектирования подбирается некоторый набор цифровых микросхем и определяется такая схема соединения их выводов, которая обеспечивает требуемое функционирование.

2. Использование программируемой логики.

Построение с использованием одной или некоторой БИС некоторого универсального устройства, в котором требуемое функционирование обеспечивается занесением в памяти устройства определённой программы. Процессор, построенный на одной или нескольких БИС, называется микропроцессором.

Контрольная работа

Вариант №1

1.Перевести

А) Из двоичной системы счисления в десятичную

100110100

1000111101

1111000100

Б) Из десятичной в двоичную

36

14

0,28

В) Из двоичной в восьмеричную

111100110110111011000100111

Г) Из двоичной в шестнадцатеричную

10111110110001101011110101100

Д) Из восьмеричной в десятичную

217

2.Выполнить операцию сложения в двоичном коде

А)-12+2

Б)1+3

В)-7-5

3.Нормализовать числа и найти их сумму

А) A1=3871,8

A2=0,00428

Б) A1+(A2+A3)

A1=6,981

A2=45,6

A3=25,51

4.Синтезировать функцию

$$F = \overline{(X1 * X2 * X3)} + X4 + \overline{X4} + X1$$

5.Минимизировать функцию

X1	X2	X3	X4	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

6.Память компьютера

Вариант №2

1. Перевести

А) из двоичной системы счисления в десятичную

10110100

11010110

10000000

Б) Из десятичной в двоичную

10

100

0,56

В) из двоичной в восьмиричную

110000111010101000011101

Г) из двоичной в шестнадцатеричную

1000011111101011110000110

Д) из восьмиричной в десятичную

123

2. Выполните операцию сложения в двоичном коде

А) 11-2

Б) 2-11

В) 2+11

Г) -2-11

3. Нормализовать числа и найти их сумму

А) $A_1=125$

$A_2=0,00125$

Б) $A_1+(A_2+A_3)$

$A_1=1,536 \cdot 10^2$

$A_2=526,4$

$A_3=0,03$

4. Синтезировать функцию

$$F = \overline{X_1} + (X_2 + \overline{X_3}) \cdot X_4 \cdot \overline{X_4}$$

5. Минимизировать функцию

X1	X2	X3	X4	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

6. Программное обеспечение

Вариант №3

1. Перевести:

А) из двоичной системы счисления в десятичную

11011000

11100101

10000011

Б) из десятичной в двоичную

37

21

0,27

В) из двоичной в восьмеричную

1111100011110001001

Г) из двоичной в шестнадцатеричную

101011010101100011100101010

Д) из восьмеричной в десятичную

547

2. Выполнить операцию сложения в двоичном коде

А) 7-5

Б) -10-2

В) 15+6

3. Нормализовать числа и найти их сумму

А) $A_1 = 378$

$A_2 = 0,0076$

Б) $A_1 + (A_2 + A_3)$

$A_1 = 2,738 \cdot 10^3$

$A_2 = 125,8$

$A_3 = 0,678$

4. Синтезировать функцию

$$F = (\overline{X_1 + X_2}) * X_3 * X_4 * \overline{X_4}$$

5. Минимизировать функцию

X1	X2	X3	X4	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

7. Вирусы. Виды вирусов, способы борьбы с ними

Вариант №4

1.Перевести:

А) из двоичной системы счисления в десятичную

100011110

1110010100

1000001111

Б) из десятичной в двоичную

59

27

0,361

В) из двоичной в восьмеричную

110010000011110001001

Г) из двоичной в шестнадцатеричную

10101101010111100011100101010

Д) из восьмеричной в десятичную

247

2.Выполнить операцию сложения в двоичном коде

А) -7-5

Б) 4+9

В) -1+6

3.Нормализовать числа и найти их сумму

А) $A_1 = 0,0378$

$A_2 = 478 \cdot 10^4$

Б) $A_1 + (A_2 + A_3)$

$A_1 = 0,698$

$A_2 = 125,8$

$A_3 = 678 \cdot 10^{-7}$

4.Синтезировать функцию

$$F = X_1 \cdot (X_2 \cdot X_3 + \overline{X_4}) \cdot X_4$$

5. Минимизировать функцию

X1	X2	X3	X4	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

7.Локальные и глобальные сети ЭВМ

Вариант №5

1. Перевести:
 - А) из двоичной системы счисления в десятичную
 110101100
 1111101010
 1001000111
 - Б) из десятичной в двоичную
 71
 23
 0,254
 - В) из двоичной в восьмеричную
 101011100101000101001110
 - Г) из двоичной в шестнадцатеричную
 111101110101011110111101010
 - Д) из восьмеричной в десятичную
 351
2. Выполнить операцию сложения в двоичном коде
 - А) -6-3
 - Б) 3+5
 - В) -2+7
3. Нормализовать числа и найти их сумму
 - А) $A1 = 0,00562$
 $A2 = 821 \cdot 10^5$
 - Б) $A1 + A2 + A3$
 $A1 = 0,00562$
 $A2 = 821 \cdot 10^5$
 $A3 = 369 \cdot 10^{-8}$
4. Синтезировать функцию

$$F = X1 \cdot (X2 \cdot X3) + X4 \cdot X4$$

5. Минимизировать функцию

1	X 2	X 3	X 4	X	F
	0	0	0	0	0
	0	0	0	1	0
	0	0	1	0	0
	0	0	1	1	0
	0	1	0	0	1
	0	1	0	1	1
	0	1	1	0	1
	0	1	1	1	1
	1	0	0	0	1
	1	0	0	1	1
	1	0	1	0	0
	1	0	1	1	0
	1	1	0	0	1
	1	1	0	1	1
	1	1	1	0	0
	1	1	1	1	0

6. Компьютерные обучающие системы, типы обучающих программ

Вариант №6

1.Перевести:

А) из двоичной системы счисления в десятичную

100001101

111010011

100100110

Б) из десятичной в двоичную

95

48

0,628

В) из двоичной в восьмеричную

1100000110110000110001001

Г) из двоичной в шестнадцатеричную

010110111010110001101011101010

Д) из восьмеричной в десятичную

323

2.Выполнить операцию сложения в двоичном коде

А) -2-5

Б) 4+13

В) -1-10

3.Нормализовать числа и найти их сумму

А) $A1 = 0,3458 \cdot 10^{-5}$

$A2 = 47,8 \cdot 10^4$

Б) $A1 + (A2 + A3)$

$A1 = 698,1 \cdot 10^0$

$A2 = 0,7 \cdot 10^{-2}$

$A3 = 25,85$

4.Синтезировать функцию

$$F = (\overline{X1} * X2 * \overline{X3}) + X4 * \overline{X4} * X1$$

5. Минимизировать функцию

X1	X2	X3	X4	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

6.Основные внешние устройства, подключаемые к ЭВМ. Монитор.

Вариант №7

1.Перевести:

А) из двоичной системы счисления в десятичную

10011011101

11101101001

10010011111

Б) из десятичной в двоичную

93

42

0,248

В) из двоичной в восьмеричную

1100110110000110001001

Г) из двоичной в шестнадцатеричную

01011010101100011010111101010

Д) из восьмеричной в десятичную

512

2.Выполнить операцию сложения в двоичном коде

А) 2+5

Б) -4-3

В) -1+14

3.Нормализовать числа и найти их сумму

А) $A1 = 32,458 \cdot 10^{-5}$

$A2 = 478 \cdot 10^4$

Б) $A1 + (A2 + A3)$

$A1 = 69,8 \cdot 10^0$

$A2 = ,00125 \cdot 10^{-2}$

$A3 = 25,8$

4.Синтезировать функцию

$$F = (\overline{X1} * \overline{X2}) * X3 + (X4 * \overline{X4}) * X1$$

5. Минимизировать функцию

X1	X2	X3	X4	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

6.Основные внешние устройства, подключаемые к ЭВМ. Принтеры.

Вариант №8

1.Перевести:

А) из двоичной системы счисления в десятичную

111001101

1100100111

11100100110

Б) из десятичной в двоичную

30

55

0,228

В) из двоичной в восьмеричную

111100110110000110001001

Г) из двоичной в шестнадцатеричную

010111110110001101011101010

Д) из восьмеричной в десятичную

187

2.Выполнить операцию сложения в двоичном коде

А) -2-5

Б) 4+13

В) 1+11

3.Нормализовать числа и найти их сумму

А) $A1 = 0,38718 \cdot 10^{-3}$

$A2 = 0,000478 \cdot 10^6$

Б) $A1 + (A2 + A3)$

$A1 = 698,1 \cdot 10^0$

$A2 = 456 \cdot 10^{-2}$

$A3 = 255,1$

4.Синтезировать функцию

$$F = X1 * (\overline{X2} * X3) + X4 * \overline{X4} * X1$$

5. Минимизировать функцию

X1	X2	X3	X4	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

6.Основные внешние устройства, подключаемые к ЭВМ. Клавиатура.

Вариант №9

1.Перевести:

А) из двоичной системы счисления в десятичную

111110100

1100111101

111000110

Б) из десятичной в двоичную

38

51

0,928

В) из двоичной в восьмеричную

1111001101101110110001001

Г) из двоичной в шестнадцатеричную

0101111101100011010111101011

Д) из восьмеричной в десятичную

258

2.Выполнить операцию сложения в двоичном коде

А) -12-2

Б) 1+13

В) 10-11

3.Нормализовать числа и найти их сумму

А) $A1 = 3871,8 \cdot 10^{-3}$

$A2 = 0,00428 \cdot 10^6$

Б) $A1 + (A2 + A3)$

$A1 = 6,981$

$A2 = 45,6 \cdot 10^{-2}$

$A3 = 2551$

4.Синтезировать функцию

$$F = (X1 * (\overline{X2 * X3}) + X4 * \overline{X4} * X1)$$

5. Минимизировать функцию

X1	X2	X3	X4	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

6. Микропроцессор. Виды микропроцессоров

Вариант №10

1. Перевести:

А) из двоичной системы счисления в десятичную

100110100

1000111101

1111000110

Б) из десятичной в двоичную

36

14

0,28

В) из двоичной в восьмеричную

111100110110111011000100111

Г) из двоичной в шестнадцатеричную

010111110110001101011110101100

Д) из восьмеричной в десятичную

217

2. Выполнить операцию сложения в двоичном коде

А) $-12+2$

Б) $1+3$

В) $-7-5$

3. Нормализовать числа и найти их сумму

А) $A1=3871,8$

$A2=0,00428$

Б) $A1+(A2+A3)$

$A1=6,981$

$A2=45,6$

$A3=2551$

4. Синтезировать функцию

$$F = (\overline{X1} * X2 * \overline{X3}) + X4 * \overline{X4} * X1$$

5. Минимизировать функцию

X1	X2	X3	X4	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

6. Память компьютера. Виды памяти. Распределение адресного пространства

Вариант №11

1. Перевести:
 - А) из двоичной системы счисления в десятичную
11100011
10001010
11011001
 - Б) из десятичной в двоичную
75
25
0,56
 - В) из двоичной в восьмеричную
1111100010101001
 - Г) из двоичной в шестнадцатеричную
101011011100011100101010
 - Д) из восьмеричной в десятичную
345
2. Выполнить операцию сложения в двоичном коде
 - А) 5+6
 - Б) -4+7
 - В) -8-2
3. Нормализовать числа и найти их сумму
 - А) $A_1 = 3,45$
 $A_2 = 0,001$
 - Б) $A_1 + (A_2 + A_3)$
 $A_1 = 2,567$
 $A_2 = 0,123$
 $A_3 = 678$
4. Синтезировать функцию
 $F = \overline{X_1} + X_2 * (X_3 * X_4)$
5. Минимизировать функцию

X1	X2	X3	X4	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

6. Интернет. Особенности работы в интернет

Вариант №12

1. Перевести:

А) из двоичной системы счисления в десятичную

11001010

11100000

10111001

Б) из десятичной в двоичную

26

41

0,31

В) из двоичной в восьмеричную

1111100010101001

Г) из двоичной в шестнадцатеричную

101011011100011100101010

Д) из восьмеричной в десятичную

341

2. Выполнить операцию сложения в двоичном коде

А) 8-3

Б) 10+2

В) -15-7

3. Нормализовать числа и найти их сумму

А) $A_1 = 3,78$

$A_2 = 0,76$

Б) $A_1 + (A_2 + A_3)$

$A_1 = 27,38 \cdot 10^3$

$A_2 = 1258$

$A_3 = 0,0678$

4. Синтезировать функцию

$$F = (X_1 + X_2) * (X_3 * X_4)$$

5. Минимизировать функцию

X1	X2	X3	X4	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

6. Операционная система WINDOWS. Ее особенности.

Вариант №13

1. Перевести:

А) из двоичной системы счисления в десятичную

11011000

11100101

10000011

Б) из десятичной в двоичную

37

21

0,27

В) из двоичной в восьмеричную

1111100011110001001

Г) из двоичной в шестнадцатеричную

101011010101100011100101010

Д) из восьмеричной в десятичную

547

2. Выполнить операцию сложения в двоичном коде

А) 7-5

Б) -10-2

В) 15+6

3. Нормализовать числа и найти их сумму

А) $A_1 = 378$

$A_2 = 0,0076$

Б) $A_1 + (A_2 + A_3)$

$A_1 = 2,738 \cdot 10^3$

$A_2 = 125,8$

$A_3 = 0,678$

4. Синтезировать функцию

$$F = (\overline{X_1 + X_2}) \cdot X_3 \cdot \overline{X_4} \cdot X_4$$

5. Минимизировать функцию

X1	X2	X3	X4	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

6. Операционная система DOS. Версии. История развития. Особенности

Вариант №14

1. Перевести:

А) из двоичной системы счисления в десятичную

100011110

1110010100

1000001111

Б) из десятичной в двоичную

59

27

0,361

В) из двоичной в восьмеричную

110010000011110001001

Г) из двоичной в шестнадцатеричную

10101101010111100011100101010

Д) из восьмеричной в десятичную

247

2. Выполнить операцию сложения в двоичном коде

А) -7-5

Б) 4+9

В) -1+6

3. Нормализовать числа и найти их сумму

А) $A_1 = 0,0378$

$A_2 = 478 \cdot 10^4$

Б) $A_1 + (A_2 + A_3)$

$A_1 = 0,698$

$A_2 = 125,8$

$A_3 = 678 \cdot 10^{-7}$

4. Синтезировать функцию

$$F = X_1 * (\overline{X_2} * X_3 + X_4) * X_4$$

5. Минимизировать функцию

X1	X2	X3	X4	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

7. Области применения ЭВМ. Особенности использования ЭВМ на производстве

Список использованных источников

Обязательная литература:

- Михеева Е.В., Титова О.И. Информатика: учебник для студентов среднего профессионального образования – 2-е изд., испр. – М. : Издательский центр «Академия», 2008. – 325 с.
- Михеева Е.В. Практикум по информатике: учебное пособие для студентов среднего профессионального образования – 6-е изд., стер. – М. : Издательский центр «Академия», 2008. – 192 с.
- Михеева Е.В. Практикум по информационным технологиям в профессиональной деятельности: учебное пособие для студентов среднего профессионального образования – 2-е изд., стер. – М.: Издательский центр «Академия», 2004. – 256 с.

Дополнительная литература:

- Антонова Г.М., Байков А.Ю. Современные средства ЭВМ и телекоммуникаций: учебное пособие для студентов среднего профессионального образования – 2-е изд., стер. – М.: Издательский центр «Академия», 2010. – 144 с.
- Мезенцев К.Н. Автоматизированные информационные системы: учебник для студентов среднего профессионального образования – 1-е изд. – М. : Издательский центр «Академия», 2010. – 176 с.
- Киселев С.В. Flash-технологии: учебник для студентов среднего профессионального образования – М. : Издательский центр «Академия», 2009. – 64 с.
- Левин В.И. Информационные технологии в машиностроении: учебник для студентов среднего профессионального образования – М. : Издательский центр «Академия», 2010. – 240 с.

Интернет-ресурсы:

- www.5byte.ru
- www.klyaksa.net