

Департамент внутренней и кадровой политики Белгородской области
Областное государственное автономное профессиональное образовательное
учреждение
«Белгородский индустриальный колледж»

Рассмотрено
ЦК «Информатики и ПОВТ»
Протокол заседания № 1
от «31» августа 2020 г.
Председатель цикловой комиссии
_____ Третьяк И.Ю.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ЛАБОРАТОРНЫХ РАБОТ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

ОП.02 Архитектура аппаратных средств

по специальности

09.02.07 Информационные системы и программирование

Квалификация: программист

Разработчик: Сергеев Павел Евгеньевич,
преподаватель ОГАПОУ «Белгородский
индустриальный колледж»

Белгород 2020

**Тематический план лабораторных работ по дисциплине
ОП.02 Архитектура аппаратных средств**

№ п/п	Тема лабораторной работы	Объем часов
1	Перевод чисел из одной системы счисления в другую	2
2	Выполнение операций над числами в естественной и нормальной формах	2
3	Исследование работы триггеров, регистров, счетчиков	2
4	Исследование работы шифраторов и дешифраторов	2
5	Исследование работы счетчиков	2
6	Построение последовательности машинных операций для реализации простых вычислений	2
7	Архитектура системной платы	2
8	Внутренние интерфейсы системной платы	2
9	Интерфейсы периферийных устройств IDE и SCSI	2
10	Параллельные порты и их особенности работы	2
11	Последовательные порты и их особенности работы	2
12	Программирование арифметических и логических команд	2
13	Программирование переходов	2
14	Программирование ввода-вывода	2
15	Программирование и отладка программ	2
16	Идентификация и установка процессора	2
17	Проектирование геоинформационной системы	2
18	Изучение принципа работы клавиатуры и мыши	2
19	Исследование работы дисплея ПК	2
20	Исследование работы печатающего устройства	2
21	Устройства ввода и вывода звуковой и видеоинформации	2
	Итого	42

Лабораторная работа № 1

Тема: «Перевод чисел из одной системы счисления в другую»

Цель работы: Получить навыки перевода чисел из одной системы счисления в другую, научиться выполнять математические действия с числами в двоичной форме счисления.

Ход работы:

Для проведения операций выбирается число состоящее из следующих разрядов: первые два разряда – число рождения студента, вторые два разряда – месяц рождения, третьи два разряда – год рождения. В итоге к операциям выбирается 6-значное число. С этим числом предстоит сделать следующие операции:

1. Перевод из десятичной системы в двоичную и обратно
2. Перевод из десятичной системы в шестнадцатичную и обратно
3. Перевод из десятичной системы в восьмиричную и обратно
4. Произвести сложение трех двоичных чисел
5. Произвести умножение двоичных чисел

Порядок вычислений и результаты заносятся в отчет

Лабораторная работа №2

Тема: «Выполнение операций над числами в различных формах»

Методические указания

При решении задач по этой теме необходимо хорошо знать степени числа 2:

$2^1=2$	$2^6=64$
$2^2=4$	$2^7=128$
$2^3=8$	$2^8=256$
$2^4=16$	$2^9=512$
$2^5=32$	$2^{10}=1024$

Задание 1. Считая, что каждый символ кодируется одним байтом, оцените информационный объем следующего высказывания Жан-Жака Руссо:

Тысячи путей ведут к заблуждению, к истине - только один.

1) 92 бита; 2) 220 бит; 3) 456 бит; 4) 512 бит.

Решение.

Определим количество символов, включая пробелы и знаки препинания - 57. Так как каждый символ кодируется одним байтом, то информационный объем равен $57 \cdot 1 \text{ байт} = 57 \text{ байт} = 57 \cdot 8 \text{ бит} = 456 \text{ бит}$.

Правильный ответ 3) 456 бит

Задание 2. Сколько различных последовательностей длиной в 7 символов можно составить из цифр 0 или 1.

1) 32; 2) 64; 3) 100; 4) 128

Решение.

Так как алфавит 2-х символьный, то информационный объем 1 символа равен 1 бит. Так как длина сообщения 7 символов, то все сообщение 7 бит. Количество последовательностей можно определить по формуле: $2^1=N$, $2^7=128$.

Правильный ответ 4) 128

Задание 3. Книга состоит из 64 страниц. На каждой странице 256 символов. Какой объем информации содержится в книге, если используемый алфавит состоит из 32 символов?

- 1) 81 920 байт; 2) 40 Кбайт; 3) 16 Кбайт; 4) 10 Кбайт.

Решение.

По условию задачи мощность алфавита равна 32 символам. Найдем информационную емкость одного символа $I = \log_2 N = \log_2 32 = 5$ (бит). Определим информационную емкость одной страницы: поскольку на странице 256 символов, то имеем $256 * 5 = 1280$ (бит). Определим информационную емкость всей книги: $64 * 1280 = 81920$ (бит).

$$81920 \text{ бит} = 81920 / 8 \text{ байт} = 10240 \text{ байт} = 10240 / 1024 \text{ килобайт} = 10 \text{ килобайт}$$

Правильный ответ 4) 10 Кбайт.

Задание 4. Световое табло состоит из лампочек, каждая из которых может находиться в двух состояниях («включено» или «выключено»). Какое наименьшее количество лампочек должно находиться на табло, чтобы с его помощью можно было передать 50 различных сигналов?

- 1) 5; 2) 6; 3) 25; 4) 50

Решение.

Так как одна лампочка может находиться в 2-х состояниях, то количество информации, которая передает одна лампочка равно 1 биту. Воспользуемся формулой $N = 2^I$. Найдём I. При $I = 5$ $N = 32$, т.е. можно закодировать 32 различных сигнала - этого недостаточно, а если 6, $2^6 = 64$, т.е. максимально можно получить 64 состояния, а 50 - тем более.

Правильный ответ 2) 6.

Задание 5. Метеорологическая станция ведет наблюдение за влажностью воздуха. Результатом одного измерения является целое число от 0 до 100 процентов, которое записывается при помощи минимально возможного количества бит. Станция сделала 80 измерений. Определите информационный объем результатов наблюдений.

- 1) 80 бит; 2) 70 байт; 3) 80 байт; 4) 560 байт

Решение.

При $I = 6$ можно измерить только $2^6 = 64$ состояния, поэтому возьмем $I=7$, $2^7=128$, тем более хватит и для 100.

Найдём объём результатов наблюдений $80 * 7 = 560$ бит = 70 байт.

Правильный ответ 2) 70 байт

Задание 6. Два текста содержат одинаковое количество символов. Первый текст составлен в алфавите мощностью 16 символов. Второй текст в алфавите мощностью 256 символов. Во сколько раз количество информации во втором тексте больше, чем в первом?

1) 2; 2) 4; 3) 8; 4) 4.

Решение.

Используя формулу $2^I=N$, найдем информационный объем одного символа для каждого текста.

Для первого текста $N = 16$, отсюда $2^I=16$, $I = \log_2 N = \log_2 16 = 4$ (бит).

Для второго текста $N = 256$, отсюда $I=\log_2 N = \log_2 256 = 8$ (бит).

Поскольку оба текста содержат одинаковое количество символов, то во втором тексте количество информации больше, чем в первом в $8/4 = 2$ раза.

Правильный ответ 1) 2.

Задание 7. Азбука Морзе позволяет кодировать символы для радиосвязи, задавая комбинацию из точек и тире. Сколько различных символов (цифр, букв, знаков пунктуации и т.д.) можно закодировать, используя код Морзе длиной не менее пяти и не более шести сигналов (точек и тире)?

1) 80; 2) 120; 3) 112; 4) 96.

Решение.

Если длина кода 5, то максимальное количество символов, которое можно закодировать определяется по формуле $2^I=N$, $2^5 = 32$.

Если длина- 6, то $2^6=64$, таким образом, общее количество символов $64+32=96$.

Правильный ответ 4) 96.

Задание 8. Скорость передачи данных через модемное соединение 28 Кбит/сек.

Передача текстового файла заняла 10 с. Определите, сколько символов содержал переданный текст, если известно, что он был представлен в кодировке Unicode.

Решение.

1 Кбит - 1024 бит, поэтому за 1 сек будет передано $28 \cdot 1024 = 28672$ бита, за 10 секунд - $28672 \cdot 10 = 286720$. Так как переданный текст представлен в кодировке Unicode, то на 1 символ отводится 16 бит, поэтому количество символов $286720 : 16 = 17920$ бит.

Задание 9. В корзине лежат шары. Все разного цвета. Сообщение о том, что достали синий шар, несет 5 битов информации. Сколько всего шаров в корзине?

1) 5; 2) 10; 3) 16; 4) 32.

Решение.

Воспользуемся формулой $I = \log_2 N$, $I = 5$ бит. Найдем N - то есть количество шаров в корзине. $N = 2^I = 2^5 = 32$ бит.

Правильный ответ 4)32.

Задание 10. В лотерее разыгрывается 64 шара. Выигрышная комбинация состоит из X шаров, и сообщение о ней несет 42 бита информации. Чему равно X ?

1) 7; 2) 2; 3) 42; 4) 64.

Решение.

В данной задаче идет речь о лотерее, в которой из 64 шаров вытягивается какое-то количество шаров, которые являются выигрышной комбинацией. Известно, что сообщение о выигрышной комбинации шаров несет 42 бита. Необходимо определить количество шаров в выигрышной комбинации.

Воспользуемся формулой Хартли $2^I = N$, $2^I = 64$, отсюда $I = 6$.- информационный объем сообщения о вытягивании 1 шара, а информационный объем о вытягивании X шаров - 42 бит. Количество шаров определяется как $42/6 = 7$ шаров

Правильный ответ 1) 7.

Задание 11. Сколько информации несет сообщение о том, что было угадано число в диапазоне целых чисел от 684 до 811?

1) 6 битов; 2) 7 битов; 3) 127 битов; 4) 128 битов.

Решение.

Определим диапазон: $811-684+1 = 128$.

Воспользуемся формулой $I = \log_2 N$, где I - информационный объем одного символа, N - мощность алфавита. Мощность алфавита равна 128 различным значениям. Определим информационный объем $I = \log_2 N = I = \log_2 128 = 7$ бит.

Правильный ответ 2) 7 битов.

Задания для самостоятельной работы.

Пояснение: решение задач в самостоятельной работе записать подробно!
Обязательно указать ответ для каждой решенной задачи!

Задание 1. Считая, что каждый символ кодируется одним байтом, оцените в битах информационный объем следующего высказывания Рене Декарта:

Я мыслю, следовательно, существую.

1) 28 бит; 2) 272 бита; 3) 32 Кбайта; 4) 34 байта.

Задание 2. Обычный дорожный светофор без дополнительных секций подает шесть видов сигналов (непрерывные красный, желтый и зеленый, мигающий желтый, мигающий зеленый, мигающие красный и желтый одновременно). Электронное устройство управления светофором последовательно воспроизводит записанные сигналы. Подряд записано 100 сигналов светофора. Сколько составляет в байтах данный информационный объем?

1) 37; 2) 38; 3) 50; 4) 100.

Задание 3. Сколько существует различных последовательностей из символов «плюс» и «минус», длиной ровно в 5 символов?

1) 64; 2) 50; 3) 32; 4) 20.

Задание 4. Какое минимальное количество битов нужно для кодирования 5 цветов палитры?

1) 8 бит; 2) 4 бита; 3) 1 бит; 4) 3 бита

Задание 5. В велокроссе участвуют 119 спортсменов. Специальное устройство регистрирует прохождение каждым из участников промежуточного финиша, записывая его номер с использованием минимально возможного количества бит,

одинакового для каждого спортсмена. Каков информационный объем сообщения, записанного устройством, после того как промежуточный финиш прошли 70 велосипедистов?

1) 70 бит 2) 70 байт 3) 490 бит 4) 119 байт

Задание 6. Какое количество информации будет получено вторым игроком в игре «Крестики-нолики» на поле 4x4 после первого хода первого игрока?

1) 1 бит; 2) 2 бита; 3) 3 бита; 4) 4 бита.

Задание 7. Для передачи секретного сообщения используется код, состоящий из десятичных цифр. При этом все цифры кодируются одним и тем же (минимально возможным) количеством бит. Определите информационный объем сообщения длиной в 150 символов.

1) 600 бит; 2) 750 бит; 3) 1200 бит; 4) 60 байт.

Задание 8. Средняя скорость передачи данных с помощью модема равна 36 Кбит/сек. Сколько секунд понадобится модему, чтобы передать 4 страницы текста в кодировке КОИ8, если считать, что на каждой странице в среднем 2304 символа?

Задание 9. Сколько бит информации несет сообщение о том, что тетраэдр, у которого все грани окрашены в разные цвета, после подбрасывания упал на синюю грань?

1) 1; 2) 2; 3) 8; 4) 4.

Задание 10. Сколько бит информации несет в себе известие о том, что монета упала гербом вверх?

1) 1; 2) 8; 3) 3; 4) 4.

Задание 11. Сколько единиц содержится в двоичной записи числа десятичного числа 173?

1) 4 2) 5 3) 6 4) 7

Лабораторная работа № 3

Тема: Исследование работы интегральных триггеров.

Цель работы: Исследовать работу синхронных и асинхронных триггеров в статическом и динамическом режимах.

Приборы: Лабораторный макет, осциллограф.

Краткие теоретические сведения

Интегральные триггеры.

Триггер- это устройство, имеющее два устойчивых состояния и способное под действием управляющего сигнала скачком переходить из одного состояния в другое.

Триггер имеет два выхода: прямой Q и инверсный \bar{Q} . Состояние триггера определяется по логическому уровню на прямом выходе. Если триггер в единичном состоянии, то $Q=1$, $\bar{Q}=0$. Если триггер в нулевом состоянии, то $Q=0$, $\bar{Q}=1$. Триггер может иметь различные информационные входы: S – отдельный вход установки триггера в единичное состояние, R – отдельный вход установки триггера в нулевое состояние, J – вход установки универсального триггера в единичное состояние, K – вход установки универсального триггера в нулевое состояние, D – вход установки триггера в состояние, соответствующее логическому уровню действующему на нем, T – счетный вход. В зависимости от того, какие входы имеет триггер, различают: RS- триггер, JK- триггер, D- триггер, T- триггер. Синхронные триггеры, кроме информационных, имеют дополнительный синхронизирующий (управляющий) вход C . Срабатывание синхронного триггера происходит в том случае, если на входе C действует определенный сигнал. Срабатывание триггеров со статическим синхронизирующим входом C происходит в том случае, если на нем действует логическая единица. Срабатывание синхронного триггера с прямым динамическим входом C может произойти в момент положительного перепада напряжения на нем, а срабатывание синхронного триггера с инверсным динамическим синхронизирующим входом происходит в момент отрицательного перепада напряжения на нем.

Асинхронный RS триггер с прямыми входами.

Может быть построен на элементах «или-не», для которых активным уровнем является логическая единица.

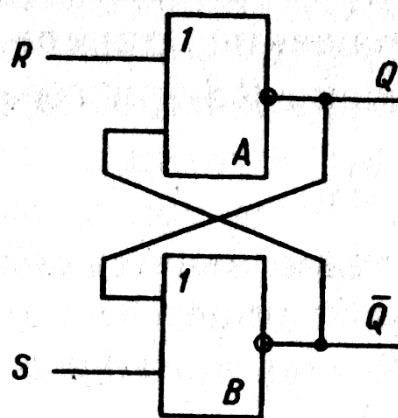
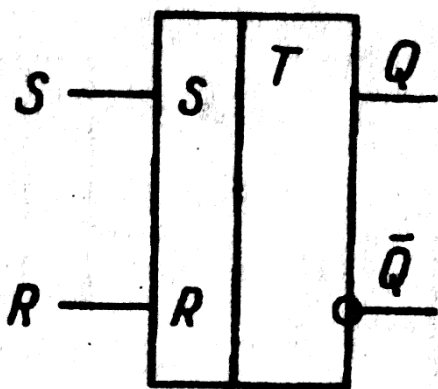


Таблица функционирования.

Вход	Выход
------	-------

S	R	Q
0	0	Q0
0	1	0
1	0	1
1	1	X

При $S=0$ и $R=0$ на триггер действуют пассивные логические уровни, срабатывание триггера не происходит и он остается в исходном состоянии $Q0$.

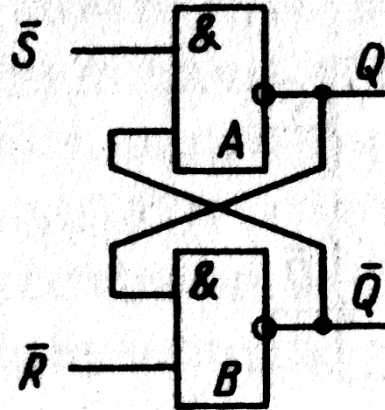
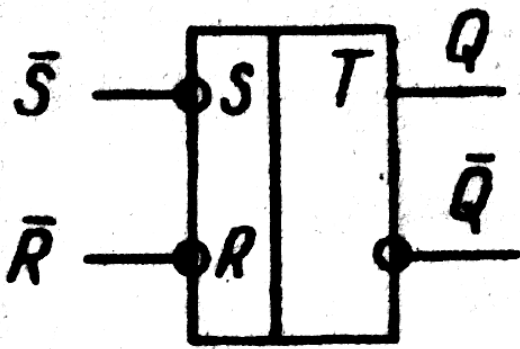
При $S=0$ и $R=1$ активный уровень логической единицы действуют на вход установки в нулевое состояние R , поэтому триггер переходит в нулевое состояние.

При $S=1$ и $R=0$ активный уровень логической единицы действуют на вход установки в единичное состояние S , поэтому триггер переходит в единичное состояние.

При $S=1$ и $R=1$ состояние триггера будет неопределенным, поэтому такая комбинация входных сигналов называется запрещенной.

Асинхронный RS триггер с инверсными входами.

Он может быть построен на элементах «и-не», для которых активным уровнем является логический ноль.

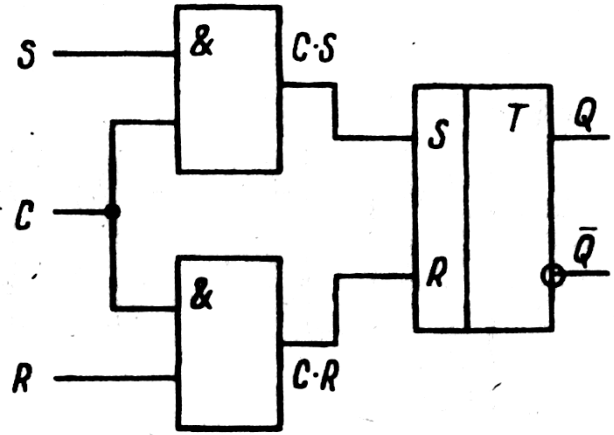
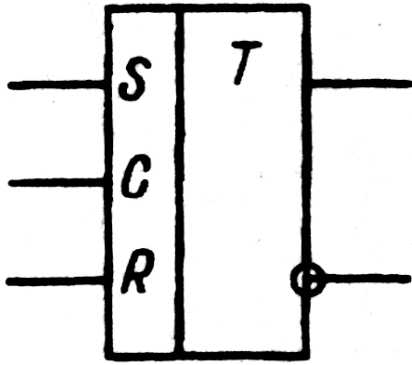


Вход		Выход
S	R	Q
0	0	X
0	1	1
1	0	0
1	1	Q0

Таблица функционирования для триггера с инверсными входами инверсна таблице функционирования для триггера с прямыми входами.

Синхронный RS – триггер.

Он может быть получен из асинхронного, если на его входы включить элементы «и».

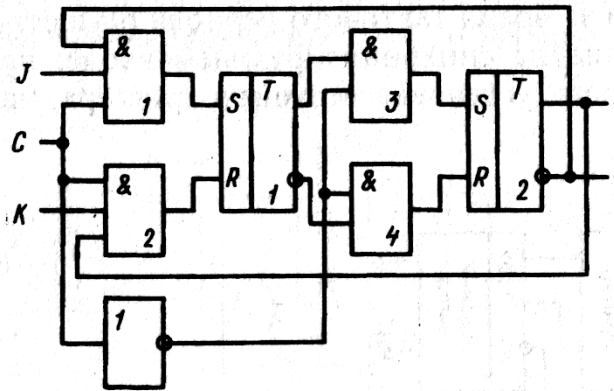
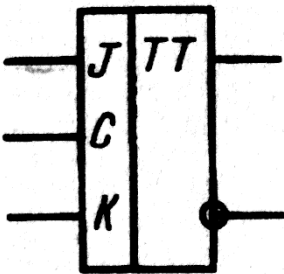


При $C=1$.

Вход		Выход
S	R	Q
0	0	Q_0
0	1	0
1	0	1
1	1	X

При $C=0$ независимо от того, что действует на входы S и R синхронного триггера, на входах S и R асинхронного триггера, входящего в состав синхронного действует логический ноль, и срабатывание не происходит. При $C=1$ логические уровни со входов синхронного триггера передаются на входы асинхронного триггера и вызывает его соответствующее срабатывание.

Универсальный JK-триггер с двухступенчатым запоминающим информации



Вход		Выход
J	K	Q
0	0	Q_0
0	1	0
1	0	1
1	1	\bar{Q}_0

При $J=0$ и $K=0$ срабатывание не происходит и триггер остается в исходном состоянии Q_0 .

При $J=0$ и $K=1$ логическая единица действует на вход установки в нулевое состояние K и триггер переходит в нулевое состояние.

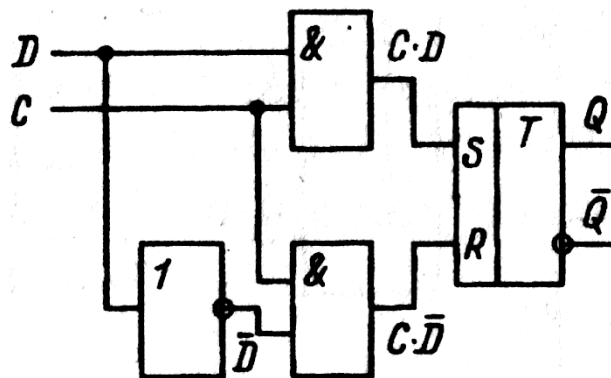
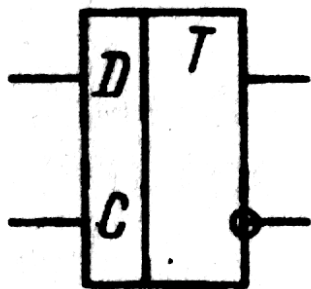
При $J=1$ и $K=0$ логическая единица действует на вход установки в единичное

состояние J и триггер переходит в единичное состояние.

Универсальный триггер не имеет запрещенной комбинации входных сигналов и при $J=1$ и $K=1$ состояние триггера меняется на противоположное.

Срабатывание триггера с двухступенчатым запоминанием информации происходит в два этапа: 1) при положительном перепаде ($C=1$) новое состояние формируется в первом RS-триггере, а логические уровни на выходах JK триггера не изменяются; 2) при отрицательном перепаде ($C=0$) новое состояние, сформированное в первом RS-триггере передается во второй RS- триггер и меняются логические уровни на выходах JK-триггера.

Синхронный D – триггер.

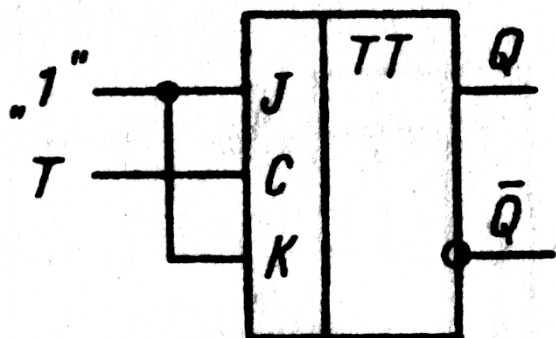


Вход	Выход
D	Q
0	0
1	1

При $C=0$ срабатывание триггера не происходит.

При $C=1$ и $D=0$ триггер устанавливается в нулевое состояние. При $C=1$ и $D=1$ триггер устанавливается в единичное состояние.

T-триггер



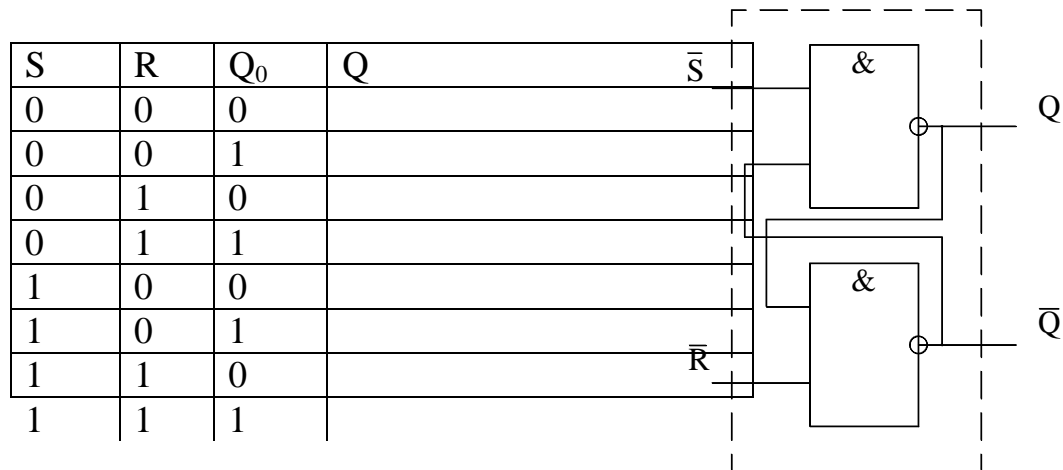
Состояние триггера меняется на противоположное, если на вход поступает прямоугольный импульс, во всех остальных случаях срабатывание триггера не происходит.

Порядок выполнения работы:

1. Изучить внутреннюю структуру, функционирование и параметры ИС К155ТВ9, К1555ТМ2.

2. Исследовать работу асинхронного **RS** – триггера. Для этого входные сигналы подавать с помощью внешних проводников, состояния триггера определять по элементам индикации.

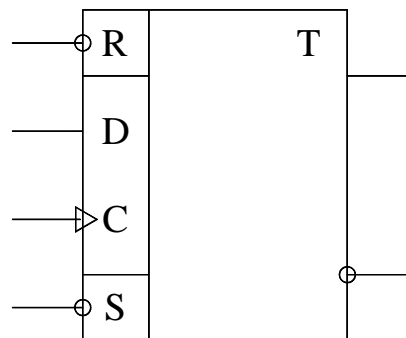
Результаты занести в таблицу



3. Исследовать работу **D** – триггера в статическом режиме. Для этого входной сигнал подавать на информационный вход **D**. Управляющий сигнал на вход “С” подать кнопкой SA1. Состояние триггера определять по элементам индикации.

Результаты занести в таблицу.

D	C	Q ₀	Q
1		1	
1		0	
0		1	
0		0	

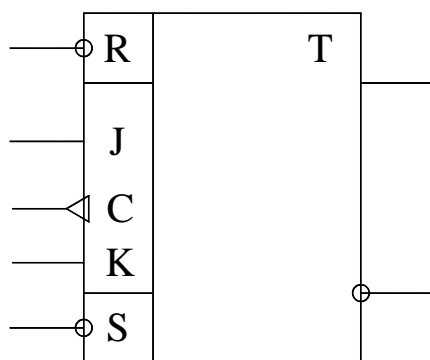


4. Исследовать работу **D** – триггера в динамическом режиме. На входы **S** и **R** подавать импульсные сигналы с частотой **F₂** через линии задержки, на вход **C** – импульсные с частотой **F₁**, на вход **D** – с частотой **F₃**. Зарисовать осциллограммы для **S, R, C, D, Q**. Построить временные диаграммы работы триггера.

5. Исследовать работу **JK** – триггера в статическом режиме. Для этого сигналы на информационные входы **J** и **K** подать внешними проводниками, управляющий сигнал на вход **C** подавать с помощью кнопки **SA1**, состояния триггера определять по элементам индикации.

Результаты записать в таблицу.

J	K	C	Q₀	Q
0	0		0	
0	0		1	
1	0		0	
1	0		1	
0	1		0	
0	1		1	
1	1		0	
1	1		1	



6. Исследовать работу **JK – триггера** в динамическом режиме. Для этого на входы **S** и **R** подать импульсные сигналы с частотой F_2 , через линии задержки, на вход **C** – импульсные сигналы с частотой F_1 , на входы **J** и **K** - сигналы с частотами $F_2/2$ и $F_2/4$. Зарисовать осциллограммы для **S, R, C, J, K, Q**.

Построить диаграммы работы триггера.

7. Сделать выводы по работе.

8. Оформить отчет.

Контрольные вопросы.

1. Привести функциональное обозначение и таблицу функционирования асинхронного **RS – триггера** с прямыми входами.

2. Привести функциональное обозначение и таблицу функционирования асинхронного **RS – триггера** с инверсными входами.

3. Привести функциональное обозначение и таблицу функционирования синхронного **RS - триггера**.

4. Привести функциональное обозначение и таблицу функционирования синхронного **D - триггера**.

5. Привести функциональное обозначение и таблицу функционирования универсального **JK - триггера**.

6. Привести функциональное обозначение и таблицу функционирования **T – триггера**.

7. Пояснить назначение выводов и принцип работы всех этих триггеров.

Лабораторная работа № 4

Тема: Исследование работы шифратора и дешифратора

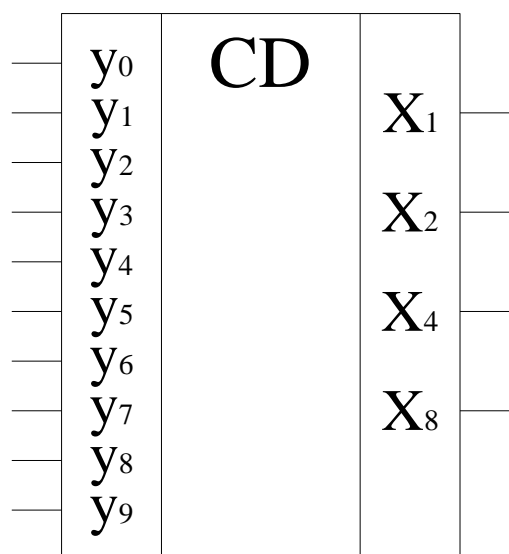
Цель: Исследовать работу комбинационных цифровых устройств, построенных на шифраторах и дешифраторах.

Приборы: лабораторный макет, осциллограф С1-112.

Краткие теоретические сведения.

Шифратор (кодер)

Служит для преобразования десятичного числа в двоичный код. Шифратор имеет входы Y пронумерованные в десятичной системе счисления и выходы X , соответствующие отдельным разрядам двоичного числа. Чтобы преобразовать десятичное число в двоичное, необходимо активный логический уровень («0» или «1») подать на вход, номер которого в десятичной форме соответствует преобразуемому числу. При этом на выходах X формируется соответствующий двоичный код.



Входы	Выходы			
	X8	X4	X2	X1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Запишем логические выражения для выходов X .

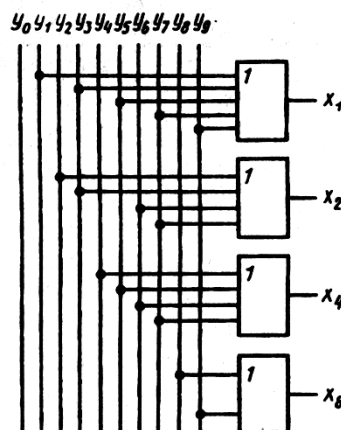
$$X_1 = Y_1 \vee Y_3 \vee Y_5 \vee Y_7 \vee Y_9$$

$$X_2 = Y_2 \vee Y_3 \vee Y_6 \vee Y_7$$

$$X_4 = Y_4 \vee Y_5 \vee Y_6 \vee Y_7$$

$$X_8 = Y_8 \vee Y_9$$

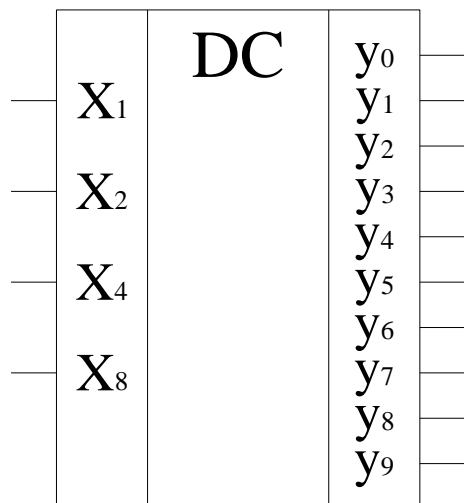
В соответствии с полученными выражениями построим схему шифратора.



Дешифратор (декодер)

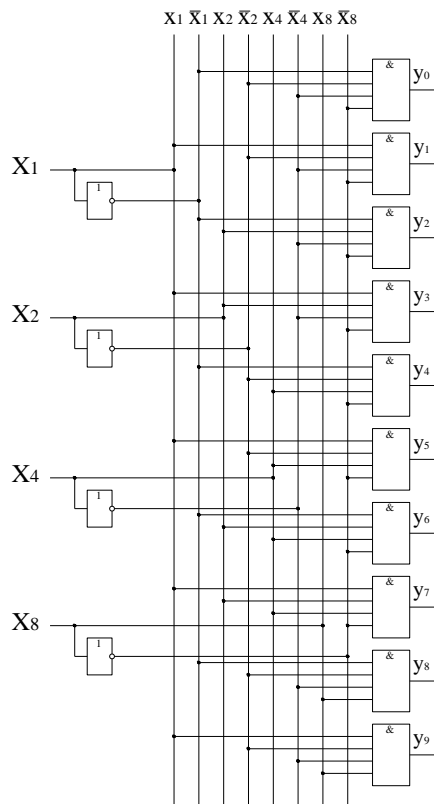
Выполняет обратную шифратору функцию, т.е. преобразует двоичный код в десятичное число. Дешифратор имеет входы X , соответствующие отдельным разрядам двоичного числа и выходы Y , пронумерованные в десятичной системе счисления. Чтобы двоичное число преобразовать в десятичное, необходимо разряды двоичного числа подать на соответствующие входы X . При этом активный логический уровень («0» или «1») формируется на одном из выходов, номер которого в десятичной форме соответствует преобразуемому коду.

Логические выражения для выходов Y



$$\begin{aligned}
 Y_0 &= \overline{X_8} \wedge \overline{X_4} \wedge \overline{X_2} \wedge \overline{X_1} \\
 Y_1 &= \overline{X_8} \wedge \overline{X_4} \wedge \overline{X_2} \wedge X_1 \\
 Y_2 &= \overline{X_8} \wedge \overline{X_4} \wedge X_2 \wedge \overline{X_1} \\
 Y_3 &= \overline{X_8} \wedge \overline{X_4} \wedge X_2 \wedge X_1 \\
 Y_4 &= \overline{X_8} \wedge X_4 \wedge \overline{X_2} \wedge \overline{X_1} \\
 Y_5 &= \overline{X_8} \wedge X_4 \wedge \overline{X_2} \wedge X_1 \\
 Y_6 &= \overline{X_8} \wedge X_4 \wedge X_2 \wedge \overline{X_1} \\
 Y_7 &= \overline{X_8} \wedge X_4 \wedge X_2 \wedge X_1 \\
 Y_8 &= X_8 \wedge \overline{X_4} \wedge \overline{X_2} \wedge \overline{X_1} \\
 Y_9 &= X_8 \wedge \overline{X_4} \wedge \overline{X_2} \wedge X_1
 \end{aligned}$$

В соответствии с логическими выражениями построим схему дешифратора.



4. Проверить работу дешифратора в динамическом режиме. Для этого кодовые комбинации на входы дешифратора подавать с выхода счетчика. Зарисовать осциллограммы входных кодов и выходных сигналов.

5. Проверить работу дешифратора К514ИД1.

6. Сделать выводы по работе.

7. Оформить отчет

Контрольные вопросы.

1. Какую функцию выполняет шифратор?
2. Привести функциональное обозначение шифратора?
3. Пояснить назначение выводов шифратора?
4. Пояснить принцип работы шифратора?
5. Какую функцию выполняет дешифратор?
6. Привести функциональное обозначение дешифратора?
7. Пояснить назначение выводов дешифратора?
8. Пояснить принцип работы дешифратора?

Лабораторная работа №5

Тема: Исследование работы счетчиков и сумматора.

Цель: Исследовать работу суммирующих и вычитающих двоичных счетчиков. Уметь строить диаграммы состояний счетчиков и осуществлять контроль работы счетчиков в статистическом и динамическом режимах.

Приборы: Макет, осциллограф.

Краткие теоретические сведения.

Счетчики.

Счетчик – устройство, выполняющее в двоичной системе счет импульсов поступивших на его вход. Счетчик состоит из триггеров, у которых организуется счетный вход, и логических элементов для организации различных режимов работы и передачи переносов от младших разрядов к старшим. Счетчики бывают суммирующие, вычитающие и реверсивные.

Реверсивный двоичный счетчик.

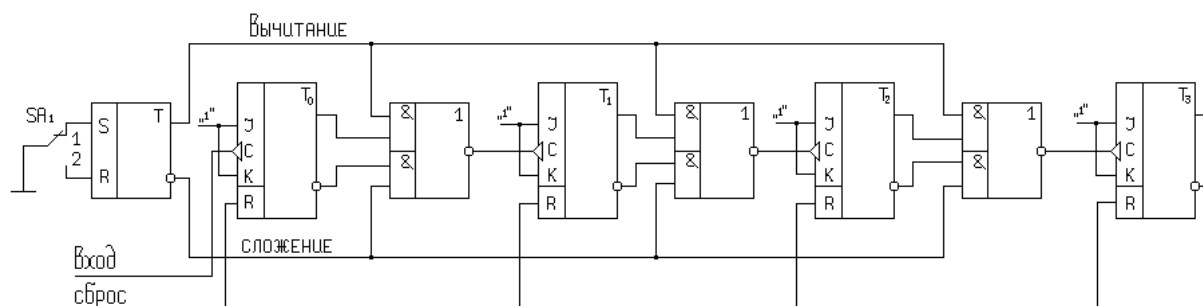


Схема состоит из RS – триггера, который служит для выбора режима счета и JK – триггеров, которые служат для запоминания отдельных разрядов двоичного числа.

В JK-триггерах входы J и K объединены и на них действует логическая «1», а вход С используется как счетный вход. Так же как в регистре входы R образуют цепь сброса, а входные импульсы поступают на вход С триггера T0.

Если переключатель SA в положении «1», то RS – триггер в нулевом состоянии и в схеме выполняется сложение. Если SA в положении «2», то RS – триггер в единичном состоянии и в схеме выполняется вычитание.

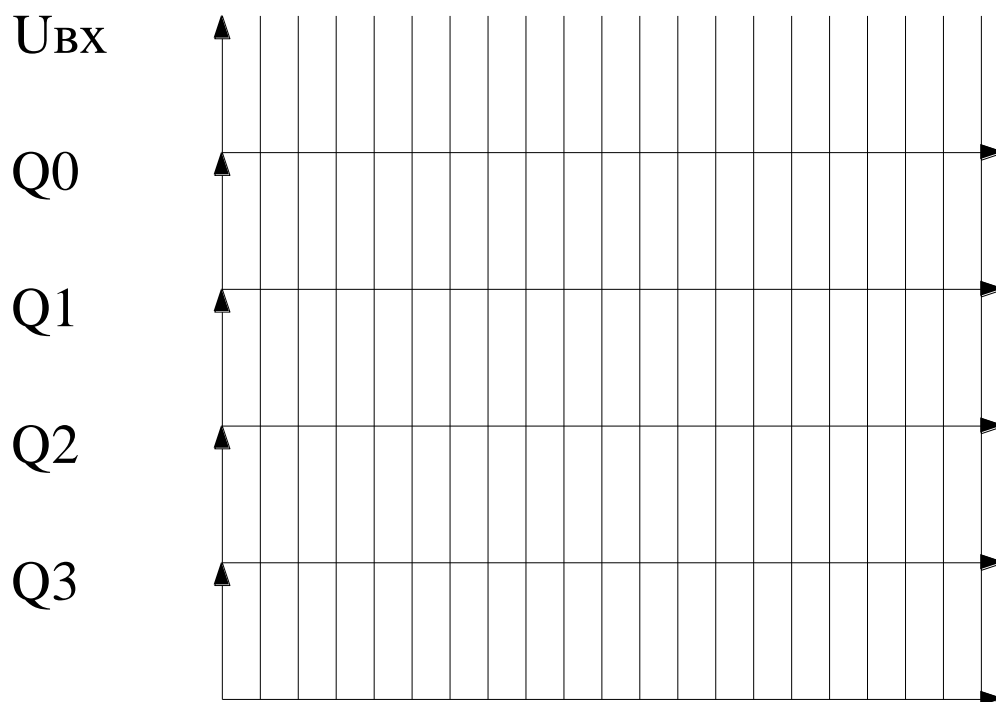
Выходы схем, «2и-или-не» соединены со входами С триггеров T1, T2, T3. Прямой выход RS – триггера и прямые выходы JK – триггеров соединены с верхними элементами «и» схем «2и-или-не», инверсный выход RS – триггера и инверсные выходы JK – триггеров соединены с нижними элементами «и» схем «2и-или-не».

Чтобы определить в какое состояние перейдет счетчик после поступления импульса на его вход необходимо:

1. Исходное состояние счетчика показать на прямых и инверсных выходах JK – триггеров.
2. Задать режим счета с помощью RS – триггера.
3. Указать логические уровни на всех входах схем «2и-или-не».
4. Определить логические уровни на выходах схем «2и-или-не».
5. При подаче импульса на вход счетчика триггер T0 всегда меняет свое состояние на противоположное, T1, T2, T3 меняют свое состояние на противоположное, если на входе С произойдет отрицательный перепад напряжения, в остальных случаях состояния триггеров не изменятся.

4. Проверить работу счетчика в динамическом режиме. Для этого входные импульсы подавать на счетчик от генератора. Зарисовать осциллограммы на выходах Q_3, Q_2, Q_1, Q_0 .

Построить временные диаграммы работы счетчика.



5. Сделать выводы по работе.

6. Оформить отчет.

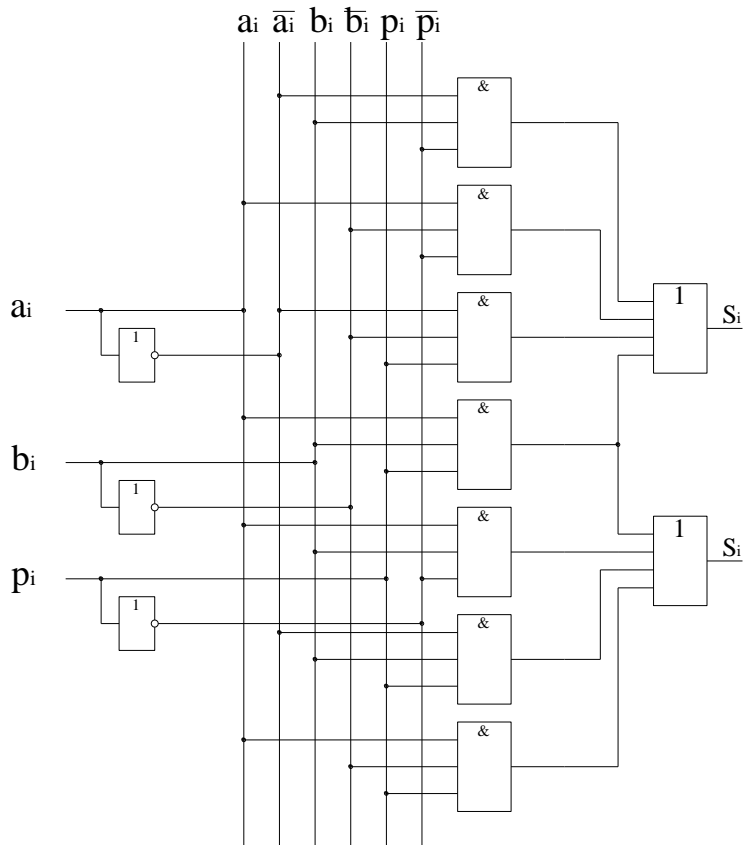
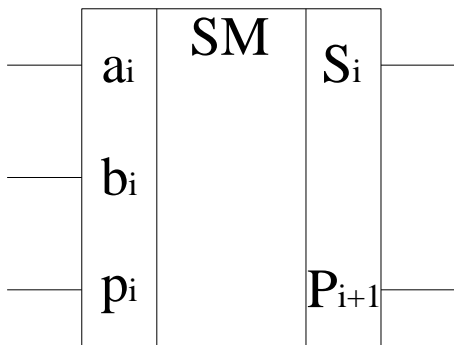
Сумматор

При сложении многоразрядных двоичных чисел в каждом разряде выполняются однотипные действия: складываются соответствующие разряды слагаемых и перенос из предыдущего разряда, при этом формируется цифра суммы данного разряда и перенос в следующий разряд. Поэтому можно построить схему одноразрядного сумматора, а для сложения многоразрядных чисел - объединить соответствующее количество одноразрядных сумматоров.

a_i	b_i	p_i	S_i	P_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

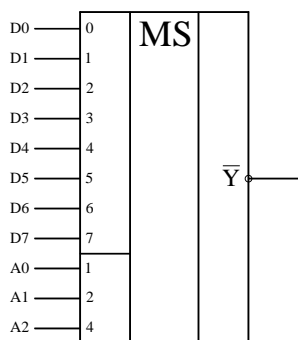
$$S_i = \bar{a}_i \bar{b}_i p_i \vee \bar{a}_i b_i \bar{p}_i \vee a_i \bar{b}_i \bar{p}_i \vee a_i b_i p_i$$

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



Порядок выполнения работы:

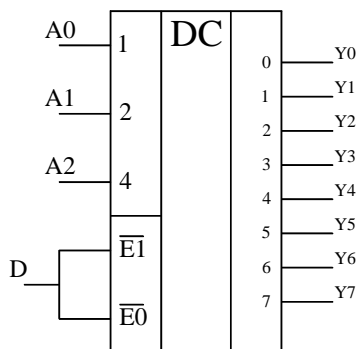
1. Изучить внутреннюю структуру, функционирование и параметры ИС К155ИД3, К155КП1.
2. Проверить работу мультиплексора, для чего импульсы от генератора подать на информационные входы D0-D7, а на адресные входы **1,2,4** подавать кодовые комбинации.



Адресные коды			Информационные коды								Выход
4	2	1	0	1	2	3	4	5	6	7	Y
0	0	0									
0	0	1									
0	1	0									
0	1	1									
1	0	0									
1	0	1									
1	1	0									
1	1	1									

Результаты занести в таблицу.

3. Проверить работу демультиплексора, для этого импульсы от генератора подать на входы E_1 E_0 , а на адресные входы **1,2,4** подавать кодовые комбинации.



Результаты занести в таблицу.

E1	E0	Входы			Выходы								
		4	2	1	0	1	2	3	4	5	6	7	
		0	0	0									
		0	0	1									
		0	1	0									
		0	1	1									
		1	0	0									
		1	0	1									
		1	1	0									
		1	1	1									

4. Объединяя мультиплексор и демультиплексор построить коммутатор каналов. Начертить схему. Проверить его функционирование.

Результат занести в таблицу.

Мультиплексор		Демультиплексор	
Вход	Адрес	Адрес	Выход

5. Проверить работу сумматора. Выполнить сложение и вычитание 3 различных двоичных чисел (по заданию преподавателя).

Для сложения необходимо SA1 перевести в положение «+», SA2 в любом положении.

Для вычитания необходимо SA1 перевести в положение «-», SA2, перевести в положение “A>B”, если A>B, и в положение “A<B”, если A<B.

Результаты записать в таблицу:

	A ₁₀	A ₃	A ₂	A ₁	A ₀	B ₁₀	B ₃	B ₂	B ₁	B ₀
1										
2										
3										

	(A+B) ₁₀	Знак	G	Σ ₃	Σ ₂	Σ ₁	Σ ₀
1							

2							
3							

	$(A-B)_{10}$	Знак	G	Σ_3	Σ_2	Σ_1	Σ_0
1							
2							
3							

Контрольные вопросы.

1. Пояснить назначение и виды счетчиков.
2. Привести функциональное обозначение счетчика K155ИЕ7. Пояснить назначение выводов.
3. Как определить период циклической работы счетчика и количество импульсов, которое он может сосчитать.
4. Пояснить режимы работы счетчика (на конкретных примерах).
5. Начертить схему реверсивного счетчика. Пояснить назначение элементов и цепей.
6. Пояснить принцип работы реверсивного счетчика (на конкретном примере).

Лабораторная работа № 6.

Тема: «Построение последовательности машинных операций для реализации простых вычислений»

Цель работы: знакомство с правилами построения последовательности машинных операций для реализации простых вычислений.

Методические указания.

Для кратковременного хранения и обработки данных в МП предусмотрены специальные ячейки памяти, конструктивно расположенные внутри МП и доступные из программ. Такие ячейки называются регистрами. Доступ к регистрам осуществляется гораздо быстрее, чем к ОП. В 8086 процессоре все регистры имеют разряд слова (16) и каждый имеет имя. В 386 -32. По назначению и способу использования регистры можно разделить на следующие группы.

1. РОН (регистры общего назначения): AX, BX, CX, DX, SI, DI, BP, SP;
2. сегментные регистры: CS, DS, SS, ES;
3. регистр флагов;
4. регистр IP.

1. Регистры общего назначения.

К ним относятся: AX: AH/AL, BX: BH/BL, CX: CH/CL, DX: DH/DL, SI, DI, BP, SP.

AX - (accumulator) аккумулятор,

BX - (base) база,

CX - (counter) счетчик,

DX (data) данные, extended,

SI - (source index) индекс источника,

DI (destination index) индекс приемника

BP (base pointer) указатель базы,

SP (stack pointer).

РОН можно использовать в любых арифметических, логических операциях. Однако каждый РОН имеет определенную спецификацию: некоторые команды требуют, чтобы их операнд или операнды находились в определенных регистрах. Делимое - ax, dx.

2. Сегментные регистры.

Сегментные регистры CS, DS, SS и ES (Code, data, stack, extra segments) не могут быть операндами никаких команд, кроме команд пересылки и стековых команд. Эти регистры используются только для сегментирования адресов

3. Регистр флагов.

В ПК имеется особый регистр флагов. Флаг - это бит, принимающий значение 1 ("флаг установлен"), если выполнено некоторое условие, и значение 0 ("флаг сброшен") в противном случае. В ПК используется 9 флагов, каждому из них присвоено определенное имя (ZF, CF и т.д.). Все они собраны в регистре флагов (каждый флаг - это один из разрядов регистра, часть его разрядов не используется):

Flags

x	x	x	x	x	OF	DF	IF	TF	SF	ZF	AF	x	PF	x	CF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Некоторые флаги принято называть флагами условий; они автоматически меняются при выполнении команд и фиксируют те или иные свойства их результата (например, равен ли он нулю). Другие флаги называются флагами состояний; они меняются из программы и оказывают влияние на дальнейшее поведение процессора (например, блокируют прерывания).

Флаги условий:

CF (carry flag) - флаг переноса. Принимает значение 1, если при сложении целых чисел появилась единица переноса, не "влезавшая" в разрядную сетку, или если при вычитании чисел без знака первое из них было меньше второго. В командах сдвига в CF заносится бит, вышедший за разрядную сетку. CF фиксирует также особенности команды умножения.

OF (overflow flag) - флаг переполнения. Устанавливается в 1, если при сложении или вычитании целых чисел со знаком получился результат, по модулю превосходящий допустимую величину (произошло переполнение мантиссы и она "залезла" в знаковый разряд).

ZF (zero flag) - флаг нуля. Устанавливается в 1, если результат команды оказался равным 0.

SF (sign flag) - флаг знака. Устанавливается в 1, если в операции над знаковыми числами получился отрицательный результат.

PF (parity flag) - флаг четности. Равен 1, если результат очередной команды содержит четное количество двоичных единиц. Учитывается обычно только при

операциях ввода-вывода.

AF (auxiliary carry flag) - флаг дополнительного переноса. Фиксирует особенности выполнения операций над двоично-десятичными числами.

Флаги состояний:

DF (direction flag) - флаг направления. Устанавливает направление просмотра строк в строковых командах: при DF=0 строки просматриваются "вперед" (от начала к концу), при DF=1 - в обратном направлении.

IF (interrupt flag) - флаг прерываний. При IF=0 процессор перестает реагировать на поступающие к нему прерывания, при IF=1 блокировка прерываний снимается.

TF (trap flag) - флаг трассировки. При TF=1 после выполнения каждой команды процессор делает прерывание (с номером 1), чем можно воспользоваться при отладке программы для ее трассировки.

4. Регистр IP.

Счетчик команд IP всегда содержит адрес (смещение от начала программы) той команды, которая должна быть выполнена следующей (начало программы хранится в регистре CS). Содержимое регистра IP можно изменить только командами перехода.

Команды процессора.

Простейшей, чаще всего используемой командой процессора, является команда

MOV Operand_1, Operand_2

Данная команда копирует содержимое правого операнда в левый операнд.

Эта команда, равно как и практически все остальные команды, оперирующие парой операндов, удовлетворяет двум важным условиям:

Результат операции сохраняется в левом операнде.

Ёмкости левого и правого операндов равны:

Operand_1: 1 байт Operand_2: 1 байт

или

Operand_1: 2 байта Operand_2: 2 байта

Для обозначения содержимого ячеек оперативной памяти при записи команд используются квадратные скобки. В квадратных скобках указывается не содержимое ячейки оперативной памяти, а адрес ячейки. Например:

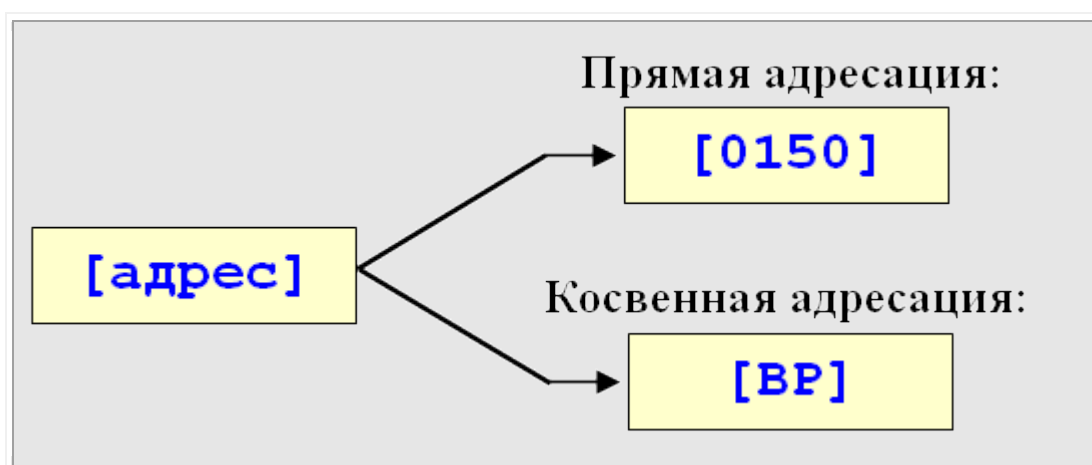
[0150] – **содержимое** 1-байтовой ячейки оперативной памяти.

0150 – **адрес** 1-байтовой ячейки оперативной памяти.

[BP] – **содержимое** 1-байтовой ячейки оперативной памяти.

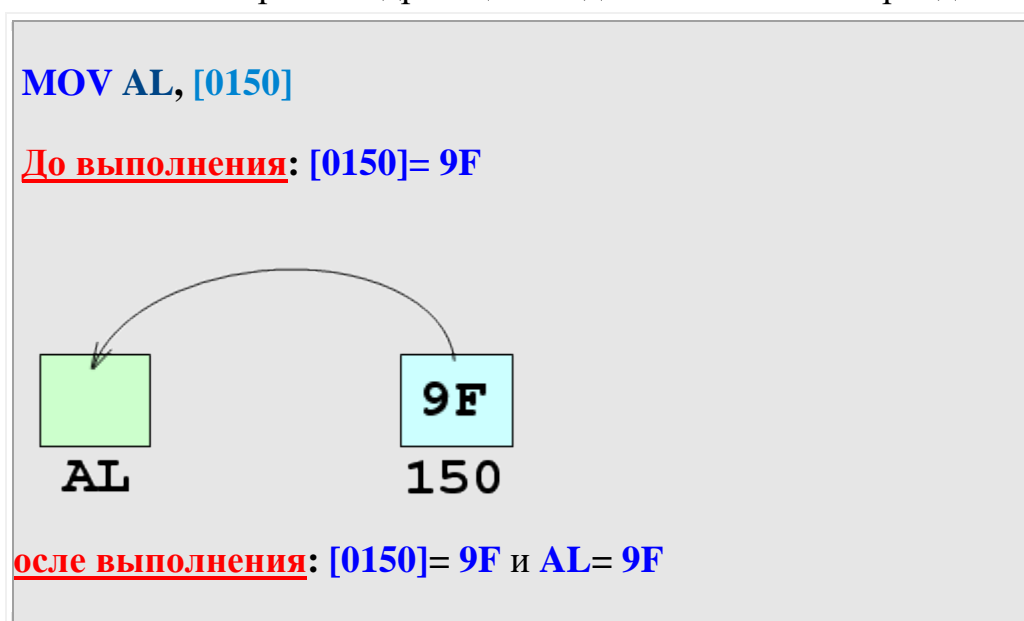
BP – регистр, содержащий **адрес** 1-байтовой ячейки памяти.

Адрес в квадратных скобках может задаваться двумя различными способами:



Прямая адресация.

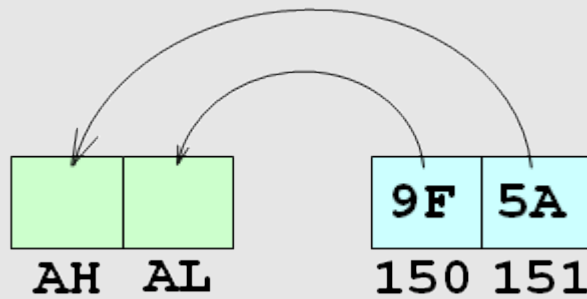
Пример использования прямой адресации с однобайтовыми операндами:



Пример использования прямой адресации с двухбайтовыми операндами:

MOV AX, [0150]

До выполнения: [0150]= 9F [0151]= 5A



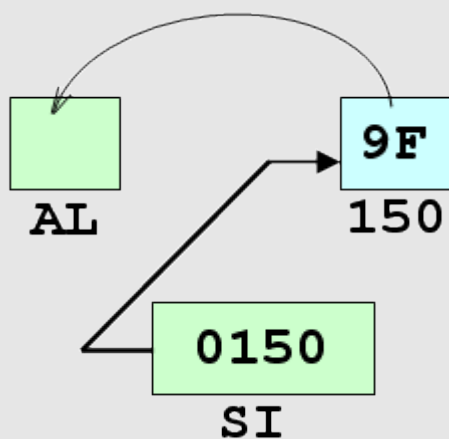
После выполнения: [0150]= 9F [0151]= 5A и AX= 5A9F, то есть AL= 9F AH= 5A

Косвенная адресация.

Пример использования косвенной адресации с однобайтовыми операндами:

MOV AL, [SI]

До выполнения: [0150]= 9F SI= 0150

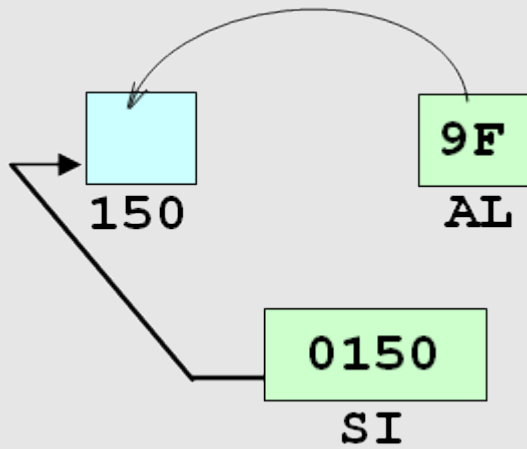


После выполнения: [0150]= 9F SI= 0150 и AL= 9F

Изменение порядка операндов отражается на результате следующим образом:

MOV [SI], AL

До выполнения: AL= 9F SI= 0150

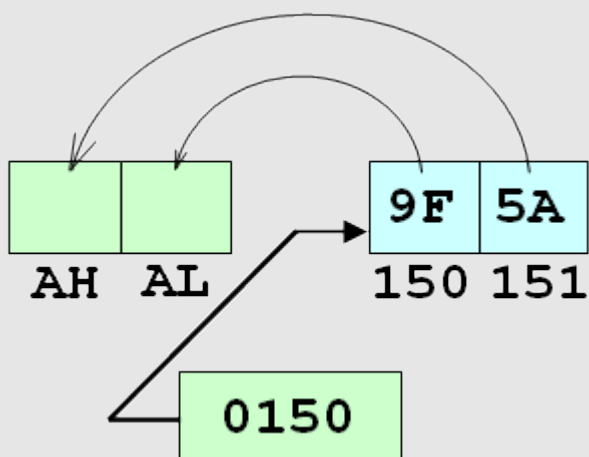


После выполнения: AL= 9F SI= 0150 и [0150]= 9F

Пример использования косвенной адресации с двухбайтовыми операндами:

MOV AX, [SI]

До выполнения: [0150]= 9F [0151]= 5A SI= 0150



После выполнения: [0150]= 9F [0151]= 5A SI= 0150 и AX= 5A9F, то есть AL= 9F и AH= 5A

Практическое использование способов адресации (на примере команды MOV):

Адрес начала команды	Команда, писанная на языке машинных слов	Та же самая команда, записанная на языке ассемблер	Комментарий, поясняющий, как полняется команда
100	B8 34 12	MOV AX,1234	<u>НЕПОСРЕДСТВЕННАЯ АДРЕСАЦИЯ:</u> Занесение 2-х байтового числа 1234 (12 – старший байт числа, 34 – младший байт числа) в 2-х байтовый регистр AX
103	A3 20 01	MOV [0120],AX	<u>ПРЯМАЯ АДРЕСАЦИЯ:</u> Копирование 2-х байтового числа, находящегося в AX , в однобайтовые ячейки оперативной памяти 0120 и 0121 (в 0120 копируется младший байт AX , в 0121 копируется старший байт AX)
106	8B 1E 20 01	MOV BX,[0120]	<u>ПРЯМАЯ АДРЕСАЦИЯ:</u> Копирование содержимого ячеек оперативной памяти 0120 и 0121 , в 2-х байтовый регистр BX (в младший байт BX , регистр BL , копируется содержимое ячейки 0120 , в старший байт BX , регистр BH , копируется содержимое ячейки 0121)
10A	BD 38 01	MOV BP,0138	<u>НЕПОСРЕДСТВЕННАЯ АДРЕСАЦИЯ:</u> Занесение 2-х байтового числа 0138 в 2-х байтовый регистр BP
10D	88 46 00	MOV [BP+00],AL	<u>КОСВЕННАЯ АДРЕСАЦИЯ:</u> Копирование однобайтового числа, находящегося в AL , в однобайтовую ячейку оперативной памяти, адрес которой содержится в регистре BP , т.е. в ячейку с вычисляемым адресом 0138 (0138+00=0138)

110	89 46 07	MOV [BP+07],AX	<p><u>КОСВЕННАЯ АДРЕСАЦИЯ:</u></p> <p>Копирование 2-х байтового числа, находящегося в AX, в ячейки оперативной памяти, на адреса которых указывает регистр BP:</p> <p>младший байт регистра AX, именуемый AL, копируется в ячейку с вычисляемым адресом 013F (0138 + 07 + 1 = 013F)</p> <p>старший байт регистра AX, именуемый H, копируется в ячейку с вычисляемым адресом 0140 (0138 + 07 + 1 = 0140)</p>
113	CD 20	INT 20	Команда завершения работы программы

Практическое использование способов адресации (на примере команды ADD):

Адрес начала команды	Команда, написанная на языке машинных кодов	Та же самая команда, написанная на языке ассемблера	Комментарий, поясняющий, как выполняется команда
100	05 34 12	ADD AX,1234	<p><u>НЕПОСРЕДСТВЕННАЯ АДРЕСАЦИЯ:</u></p> <p>Сложение 2-х байтового числа 1234 (2 – старший байт числа, 34 – младший байт числа) и содержимого 2-х байтового регистра AX. Результат складывается в регистре AX.</p>
103	01 06 20 01	ADD [0120],AX	<p><u>ПРЯМАЯ АДРЕСАЦИЯ:</u></p> <p>Сложение 2-х байтового числа, находящегося в AX, с 2-х байтовым числом, расположенном в ячейках оперативной памяти 0120 и 0121 (с содержимым ячейки 0120 складывается содержимое младшего байта AX, а с содержимым ячейки 0121 складывается содержимое</p>

			<p>аршего байта AX).</p> <p>Результат размещается в ячейках 20 и 0121.</p>
107	03 1E 20 01	ADD BX,[0120]	<p><u>ПРЯМАЯ АДРЕСАЦИЯ:</u></p> <p>Сложение содержимого ячеек оперативной памяти 0120 и 0121 с содержимым 2-х байтового регистра X (к младшему байту регистра BX, регистру BL, добавляется содержимое ячейки 0120, а к содержимому старшего байта регистра BX, регистра H, добавляется содержимое ячейки 21).</p> <p>Результат размещается в регистре BX.</p>
10B	BD 38 01	MOV BP,0138	<p><u>НЕПОСРЕДСТВЕННАЯ АДРЕСАЦИЯ:</u></p> <p>Занесение 2-х байтового числа 0138 в 2-х байтовый регистр BP</p>
10E	00 46 20	ADD [BP+20],AL	<p><u>КОСВЕННАЯ АДРЕСАЦИЯ:</u></p> <p>Сложение однобайтового числа, находящегося в AL, с содержимым 1-байтовой ячейки оперативной памяти, адрес которой определяется как сумма содержимого регистра BP и значения 20, т.е. с содержимым ячейки с вычисляемым адресом 0158 (138+20=0158).</p> <p>Результат размещается в ячейке 58.</p>
111	01 46 E0	ADD [BP-20],AX	<p><u>КОСВЕННАЯ АДРЕСАЦИЯ:</u></p> <p>Сложение 2-байтового числа, находящегося в AX с содержимым ячеек оперативной памяти, адреса которых вычисляются как BP-20: младший байт регистра AX,</p>

			<p>менюемый AL, складывается с держимым ячейки, вычисляемый рес которой определяется как едующая сумма: 0118 (0138 - 20 + 0 0118)</p> <p>старший байт регистра AX, менуемый AH, складывается с держимым ячейки с вычисляемым ресом 0119 (0138 - 20 + 1 = 0119)</p> <p>Результат размещается в ячейках 18 и 0119.</p>
114	CD 20	INT 20	Команда завершения работы программы

Трассировка представленного выше кода :

Позволяет проследить, как выполняется операция сложения.

```

-u 100 113
0CC0:0100 053412      ADD     AX,1234
0CC0:0103 01062001     ADD     [0120],AX
0CC0:0107 031E2001     ADD     BX,[0120]
0CC0:010B BD3801         MOV     BP,0138
0CC0:010E 004620      ADD     [BP+20],AL
0CC0:0111 0146E0      ADD     [BP-20],AX
-d 118 150
0CC0:0110                00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0CC0:0120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0CC0:0130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0CC0:0140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0CC0:0150 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0CC0:0160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
-t=100 3
AX=1234 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0CC0 ES=0CC0 SS=0CC0 CS=0CC0 IP=0103  NU UP EI PL NZ NA PO NC
0CC0:0103 01062001     ADD     [0120],AX                DS:0120=0000
AX=1234 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0CC0 ES=0CC0 SS=0CC0 CS=0CC0 IP=0107  NU UP EI PL NZ NA PO NC
0CC0:0107 031E2001     ADD     BX,[0120]                DS:0120=1234
AX=1234 BX=1234 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0CC0 ES=0CC0 SS=0CC0 CS=0CC0 IP=010B  NU UP EI PL NZ NA PO NC
0CC0:010B BD3801         MOV     BP,0138
-D 120 110
0CC0:0120 34 12 00 00 00 00 00 00 00 00 00 00 00 00 00 00  4.....

```

ервая команда **ADD AX,1234** приводит к тому, что **AX = 0+1234 = 1234**

Вторая команда **ADD [0120], AX** приводит к результату: **[0120] = [0120] + AX = + 1234 = 1234 ([0120]= 34 [0121]= 12)**

Третья команда **ADD BX,[0120]** приводит к результату: **BX= BX + 1234= 1234**

Трассировка второй группы команд даёт следующие результаты:

```
-t=10B 3
AX=1234 BX=1234 CX=0000 DX=0000 SP=FFEE BP=0138 SI=0000 DI=0000
DS=0CC0 ES=0CC0 SS=0CC0 CS=0CC0 IP=010E NU UP EI PL NZ NA PO NC
0CC0:010E 004620 ADD [BP+20],AL SS:0158=00

AX=1234 BX=1234 CX=0000 DX=0000 SP=FFEE BP=0138 SI=0000 DI=0000
DS=0CC0 ES=0CC0 SS=0CC0 CS=0CC0 IP=0111 NU UP EI PL NZ NA PO NC
0CC0:0111 0146E0 ADD [BP-20],AX SS:0118=0000

AX=1234 BX=1234 CX=0000 DX=0000 SP=FFEE BP=0138 SI=0000 DI=0000
DS=0CC0 ES=0CC0 SS=0CC0 CS=0CC0 IP=0114 NU UP EI PL NZ NA PO NC
0CC0:0114 06 PUSH ES
-d 118 15f
0CC0:0110 34 12 00 00 00 00 00 00 00 00 00 00 00 4.....
0CC0:0120 34 12 00 00 00 00 00 00 00 00 00 00 00 4.....
0CC0:0130 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0CC0:0140 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0CC0:0150 00 00 00 00 00 00 00 00 00 00 00 00 00 .....4.....
```

Первая команда **MOV BP,138** приводит к результату: **BP= 0138**

Вторая команда **ADD [BP+20],AL** приводит к результату: адрес= **0138 + 20 = 0158**. Далее, **[0158]= [0158] + AL = 0 + 34= 34**

Третья команда **ADD [BP-20],AX** приводит к результату: адрес= **0138 - 20 = 0118**. Далее, **[0118]= [0118] + AL = 0 + 34= 34** и **[0119]= [0119] + AH = 0 + 12= 12**

Задания к практической работе:

Выполнить следующие задания:

1. Изучить содержание практической работы.
2. Рассмотреть выполнение разобранных в работе примеров.

1 Чем завершится выполнение следующей группы команд:

```

MOV     BX,80A0
MOV     AX,90E0
DD      BX,BX
DD      AX,BX
MOV     [BX],AX
    
```

Результат выполнения группы команд записать в виде таблицы:

	H	L
AX	?	?
BX	?	?

2 Чем завершится выполнение следующей группы команд:

```

MOV     DX,     E0F0
MOV     CX,     A0C0
DD      CX,     DX
DD      DX,     CX
MOV     [128], DX
    
```

Результат выполнения группы команд записать в виде таблицы:

	H	L
CX	?	?
DX	?	?
0128	?	
0129	?	

3 Чем завершится выполнение следующей группы команд:

```

MOV     DX,     E0B0
MOV     AX,     F0D0
DD      AX,     DX
DD      AH,     AL
DD      AL,     AH
MOV     [132], AX
    
```

Результат выполнения группы команд записать в виде таблицы:

	H	L
AX	?	?
DX	?	?
0132	?	
0133	?	

4 Чем завершится выполнение следующей группы команд:

```

MOV     CX,     7090
MOV     BX,     9070
DD      BX,     BX
DD      CH,     BL
DD      CL,     BH
MOV     [BX], CH
    
```

Результат выполнения группы команд записать в виде таблицы:

	H	L
AX	?	?
BX	?	?
CX	?	?

5

Чем завершится выполнение следующей группы команд:

```
MOV     BP,     190
MOV     CX,     9070
DD      CX,     BP
DD      CH,     CL
DD      CL,     CH
MOV     [BP], CX
```

Результат выполнения группы команд записать в виде таблицы:

	H	L
BX	?	?
CX	?	?
BP	?	

Лабораторная работа № 7

Тема: «Архитектура системной платы»

Цель работы: изучение архитектуры системной платы.

Методические указания:

Системная, или *материнская* (motherboard — МП), плата — это важнейшая часть компьютера, содержащая основные электронные компоненты машины. С помощью материнской платы осуществляется взаимодействие между большинством устройств машины.

Конструктивно системная плата представляет собой печатную плату площадью 100-150 кв. см, на которой размещается большое число различных микросхем, разъемов и других элементов. Существует две основные разновидности конструкции системной платы (СП):

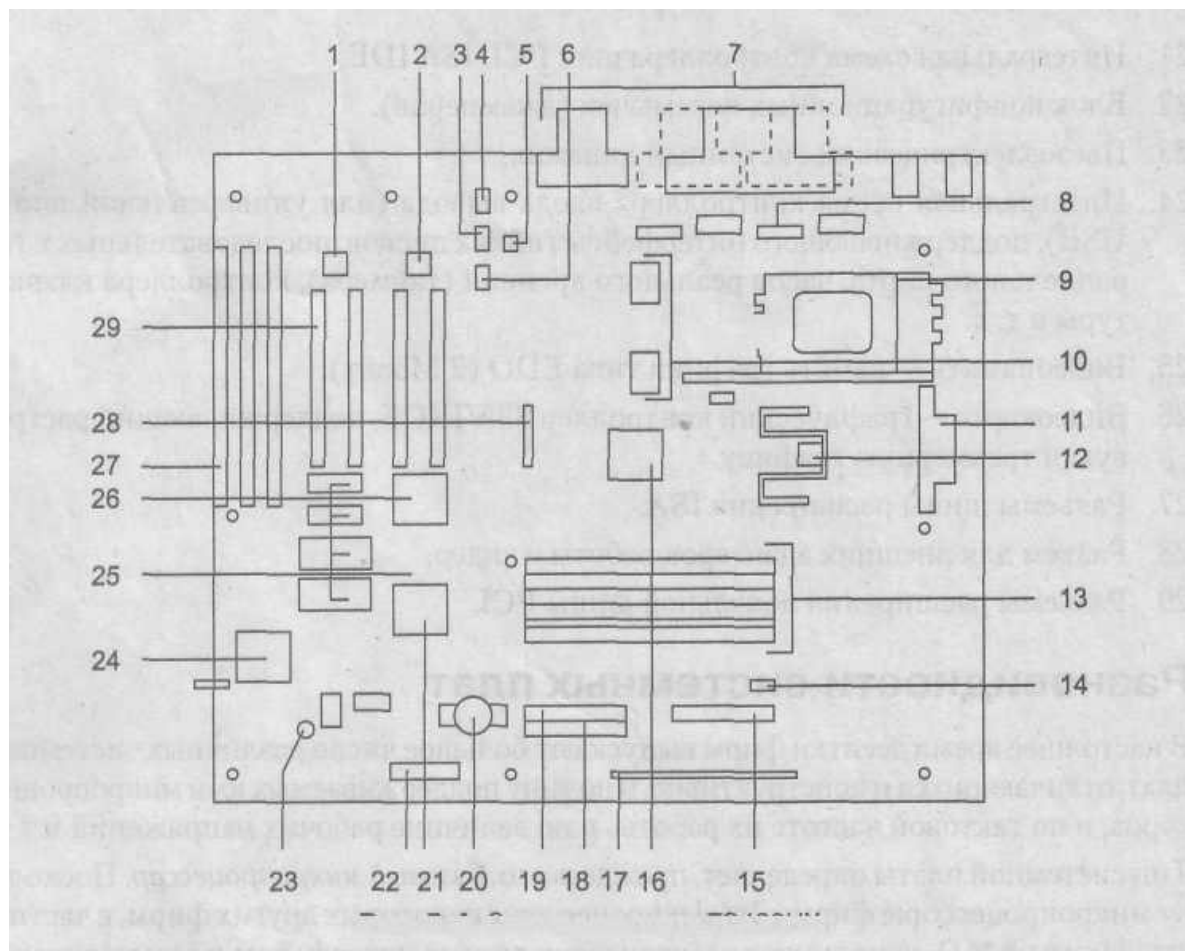
1. на плате жестко закреплены все необходимые для работы микросхемы — сейчас такие платы используются лишь в простейших домашних компьютерах, называемых одноплатными;
2. непосредственно на системной плате размещается лишь минимальное количество микросхем, а все остальные компоненты объединяются при помощи системной шины и конструктивно устанавливаются на дополнительных платах (*платах расширения*), устанавливаемых в специальные разъемы (слоты), имеющиеся на материнской плате; компьютеры, использующие такую технологию, относятся к вычислительным системам с шинной архитектурой.

Современные профессиональные персональные компьютеры имеют именно *шинную архитектуру*. На системной плате непосредственно расположены:

- разъем для подключения микропроцессора;
- набор системных микросхем (чипсет, chipset), обеспечивающих работу микропроцессора и других узлов машины;
- микросхема постоянного запоминающего устройства, содержащего программы базовой системы ввода-вывода (basic input-output system — BIOS);
- микросхема энергонезависимой памяти (питается от автономного, расположенного на МП аккумулятора), по технологии изготовления называемая CMOS;
- микросхемы кэш-памяти 2-го уровня (если они отсутствуют на плате микропроцессора);

- разъемы (слоты) для подключения модулей оперативной памяти;
- наборы микросхем и разъемы для системных, локальных и периферийных интерфейсов;
- микросхемы мультимедийных устройств и т. д.

В качестве примера на рис. 1 показано размещение основных компонентов на мультимедийной системной плате TC430NH.



На рисунке показаны:

1. Интегральная схема звукового адаптера Yamaha OPL4-ML — звуковая карта с поддержкой табличного синтеза звука Wave Table.
2. Интегральная схема звукового адаптера Yamaha OPL3-SA, звуковая карта с поддержкой цифрового частотно-модулированного синтеза звука.
3. Выход CD-ROM audio.
4. Разъем для подключения внешнего звукового адаптера.
5. Разъем для подключения телефонной линии.

6. Интегральная схема стереоадаптера audio.
7. Разъемы ввода-вывода на задней панели ПК.
8. Последовательный порт COM 2.
9. Разъем типа Socket 7 для микропроцессора.
10. Микросхемы кэш-памяти 2-го уровня.
11. Разъем питания материнской платы (два отдельных напряжения питания 2,8 и 3,3 В).
12. Регулятор напряжения.
13. Разъемы (слоты) для микросхем оперативной памяти.
14. Разъем для подключения вентилятора микропроцессора.
15. Разъем для подключения дисководов гибких дисков.
16. Набор системных микросхем — чипсет 1430HX.
17. Разъемы передней панели.
18. Разъем первичного канала дискового интерфейса IDE.
19. Разъем вторичного канала дискового интерфейса IDE.
20. Аккумулятор для системы CMOS (в том числе и для внутреннего таймера).
21. Интегральная схема контроллера шин PCI/ISA IDE.
22. Блок конфигурационных перемычек (джамперов).
23. Пьезоэлектрический системный динамик.
24. Интегральная схема контроллера ввода-вывода (для универсальной шины USB), поддерживающего интерфейсы гибких дисков, последовательных и параллельного порта, часов реального времени (таймера), контроллера клавиатуры и т. д.
25. Видеопамять — память графики типа EDO (2 Мбайт).

26. Видеокарта — графический контроллер S3 VIRGE, поддерживающий растровую и трехмерную графику.

27. Разъемы шины расширения ISA.

28. Разъем для внешних адаптеров работы с видео.

29. Разъемы расширения локальной шины PCI.

При **выборе системной платы** следует учитывать:

- Микропроцессор, который может быть установлен на плате;
- Типоразмер системной платы (должен быть согласован с возможностями системного блока);
- Тактовую частоту, на которой работает системная плата;
- Набор основных и вспомогательных микросхем (чипсет), обеспечивающих эффективную работу ПК;
- Основную, локальную и периферийные шины, с которыми плата может работать, и количество слотов для них;
- Наличие и возможность установки кэш-памяти;
- Наличие разъемов для подсоединения микросхем (разъем для процессора, слоты микросхем памяти и т.д.)
- Электронные схемы, управляющие различными устройствами компьютера, называются контроллерами.
- В настоящее время выпускается большое число различных системных плат, отличающихся конструктивно.
- Существуют различные базовые типоразмеры (форм-факторы) плат. Форм-фактор определяет не только внешние размеры системных плат, но и ряд специфических параметров, характеризующих функциональные и эксплуатационные свойства системной платы.

Задания на лабораторную работу:

Задание 1. Идентифицируйте элементы системной платы ASRock M3A770DE. Установите тип платы или форм-фактор (поясните).



Задание 2. Нарисуйте схему системной платы, указав на ней основные элементы:

1. центральный процессор;
2. разъемы задней панели;
3. разъемы для оперативной памяти (количество);
4. разъемы PCI (количество);
5. разъемы IDE;
6. разъем для подключения к блоку питания;
7. дисковые контролеры (тип разъема);
8. микросхема BIOS;
9. северный мост;
10. южный мост;
11. подключение USB.

Задание 3. Опишите назначение каждого элемента материнской платы, которые вы указали на схеме.

Вопросы для зачета

1. Что такое системная плата и зачем она нужна?
2. Что такое чипсет?
3. Что такое системная шина?

4. Назовите основные устройства, расположенные на системной плате.

Лабораторная работа № 8

Тема: «*Внутренние интерфейсы системной платы*»

Цель работы: изучение внутренних интерфейсов системной платы.

Методические указания

Интерфейс – это совокупность средств сопряжения и связи, обеспечивающая эффективное взаимодействие систем или их частей. В интерфейсе обычно предусмотрены вопросы сопряжения на механическом (число проводов, элементы связи, типы соединений, разъемы, номера контактов и т.п.) и логическом (сигналы, их длительность, полярность, частоты и амплитуда, протоколы взаимодействия) уровнях.

Внутренний интерфейс – это система связи и сопряжения узлов и блоков компьютера между собой. Представляет собой совокупность электрических линий связи, схем сопряжения с компонентами компьютера, протоколов (алгоритмов) передачи и преобразования сигналов.

В современных компьютерах в качестве системного интерфейса обычно используется системная шина.

Шина (bus) – это совокупность линий связи, по которым информация передается одновременно. Под основной или системной шиной понимается шина между процессором и подсистемой памяти. Шины характеризуются разрядностью и частотой.

Разрядность или ширина шины (bus width) – количество линий связи в шине, т.е. количество битов, которое может быть передано по шине одновременно.

Тактовая частота шины (bus frequency) – частота с которой передаются последовательные биты информации по линиям связи.

В качестве системной шины в ПК могут использоваться шины расширений и локальные шины.

Шины расширений – шины общего назначения, позволяющие подключать большое количество самых разнообразных устройств.

Локальные шины специализируются на обслуживании небольшого количества устройств определенного класса.

Шины расширений

Шина расширения ISA (Industry Standard Architecture) – основная шина на устаревших материнских платах. Служит для подключения видеокарт, модемов, звуковых карт и т.д. Конструктивно представляет собой разъем состоящий из двух частей – 62-контактного и примыкающего к нему 36-контактного сегментов. Допускает подключение до 6 устройств. Пропускная способность шины до 16 Мбайт/с. Рабочая частота до 8 МГц. Представлена в двух версиях PC/XT и PC/AT.

Шина PC/XT - 8-разрядная шина данных и 20-разрядная шина адреса, имеет 4 линии аппаратных прерываний и 4 канала для прямого доступа в память. Шина адреса ограничивает адресное пространство микропроцессора величиной 1 Мбайт. Тактовая частота 4,77 МГц.

Шина PC/AT – 16-разрядная шина данных и 24-разрядная шина адреса. Имеет 7 линий для аппаратных прерываний и 4 канала прямого доступа в память. Тактовая частота до 16 МГц. Адресное пространство шины до 16 Мбайт.

Шина EISA (Extended Industry Standard Architecture) – 32-разрядная адресная шина данных и 32-разрядная шина адреса. Адресное пространство шины 4 Гбайт. Тактовая частота 8-33 МГц. Пропускная способность 33 Мбайт/с. Теоретически может подключаться до 15 устройств. Шина поддерживает многопроцессорную архитектуру. Шина весьма дорога и применяется в скоростных ПК, сетевых серверах и рабочих станциях. Внешне слоты шины имеют такой же вид, как и ISA, и

в них могут вставляться платы ISA , но в глубине разъема находятся дополнительные ряды контактов EISA, а платы EISA имеют более высокую ножевую часть разъема с дополнительными рядами контактов.

Локальные шины.

В настоящее время существует три основных стандарта универсальных локальных шин: VLB, PCI, AGP.

Шина VLB (VL-bus, VESA Local Bus) является расширением внутренней шины микропроцессора для связи с видеоадаптером или жестким диском, платами мультимедиа, сетевым адаптером. Разрядность шины для данных 32 бита, для адреса 30 бит.

Шина не адаптирована для процессоров класса Pentium. Имеется жесткая зависимость от тактовой частоты процессора. Шина позволяет только 4 устройства, при этом отсутствует арбитраж шины, т.е. могут быть конфликты между подключаемыми устройствами.

Шина PCI (Peripheral Component Interconnect, соединение внешних компонентов) – с помощью этого интерфейса к материнской плате подключаются видеокарты, звуковые карты, модемы, контроллеры SCSI и др. устройства. Конструктивно разъем шины на системной плате состоит из двух следующих подряд секций по 64 контакта (каждая со своим ключом). Разрядность шины – 32 разряда данных и 32 разряда адреса с возможностью расширения до 64 бит. Тактовая частота шины 33 МГц. Допускает подключение до 10 устройств. При наличии шины PCI шины расширения подключаются не непосредственно к микропроцессору, а к самой шине PCI. Благодаря такому решению шина является независимой от процессора и может работать параллельно с шиной процессора, не обращаясь к ней с запросами. Таким образом, загрузка шины процессора существенно снижается.

Таблица. Спецификации PCI

Спецификация PCI	Дата выпуска	Основные изменения
PCI 1.0	Июнь 1992 г.	Оригинальная 32/64-разрядная спецификация
PCI 2.0	Апрель 1993 г.	Определенные соединители и платы расширения
PCI 2.1	Июнь 1995 г.	Рабочая частота 66 МГц, порядок групповых операций, изменение времени задержек
PCI 2.2	Январь 1999 г.	Управление режимом электропитания, механические изменения
PCI-X 1.0	Сентябрь 1999	Рабочая частота 133 МГц, дополнение к спецификации 2.2
Mini-PCI	Ноябрь 1999 г.	Уменьшенный форм-фактор плат, дополнение к спецификации 2.2
PCI 2.3	Март 2002 г.	Напряжение 3,3 В, предназначена для низкопрофильных плат расширения
PCI-X 2.0	Июль 2002 г.	Рабочая частота 266 или 533 МГц, подразделение 32-разрядной шины данных на 32- и 16-разрядные сегменты для использования с различными устройствами, имеющими напряжение 3,3/1,5 В
PCI-Express	Июль 2002 г.	Общее быстродействие 2,5 Гбайт/с, рабочее напряжение 0,8 В, 250 Мбайт/с на каждую опускную полосу. Предназначена для замены шины PCI 2.x в ПК

Тактовая частота стандартной шины PCI равна 33 МГц, а разрядность соответствует разрядности данных процессора. Для 32-разрядного процессора пропускная способность составляет 132 Мбайт/с:

$$33 \text{ МГц} \times 32 \text{ бит} = 1056 \text{ Мбит/с}; 1056 \text{ Мбит/с} : 8 = 132 \text{ Мбайт/с}.$$

Шина AGP (Accelerated Graphics Port – ускоренный графический порт) – интерфейс для подключения видеоадаптера к отдельной магистрали, имеющей выход непосредственно на системную память. Шина может работать с частотой до 133 МГц и обеспечивает высочайшую скорость передачи графических данных. По сравнению с шиной PCI, в шине AGP устранена мультиплексность линий адреса и данных (в PCI для удешевления конструкции адрес и данные передаются по одним и тем же линиям) и усилена конвейеризация операций чтения-записи, что позволяет устранить влияние задержек в модулях памяти на скорость выполнения этих

операций.

К шинам подключаются электронные платы (контроллеры). Каждый контроллер может быть подключен только к той шине, на которую он рассчитан. Поэтому разъемы разных шин сделаны разными.

Задания на лабораторную работу:

Задание 1. С помощью программы **AIDA64 Extreme Edition** идентифицируйте внутренние интерфейсы системной платы целевого компьютера.

Заполните отчетные таблицы:

	Изготовитель	Модель
Процессор		
Чипсет		
Система BIOS		

Количество разъемов модулей оперативной памяти (<i>укажите для каждого</i> <i>икость</i>)				
SIMM	DIMM	DDR	DDR2	DDR3

Тип шины	Количество портов для установки плат расширения	Разрядность бит	Частота шины, МГц	Скорость передачи данных, Мбайт/с	Устройства подключения
ISA					
PCI					
AGP					
COM1					
COM2					
LTP1					

Задание 2. Дайте сравнительную характеристику внутренних интерфейсов целевой системной платы.

Вопросы для зачета:

1. Какие типы внутренних интерфейсов вам известны?
2. Дайте сравнительную характеристику шины ISA

3. Дайте сравнительную характеристику шины РСІ

4. Дайте сравнительную характеристику шины АGR

Лабораторная работа № 9

Тема: «Интерфейсы периферийных устройств IDE и SCSI»

Цель работы: изучение интерфейсов периферийных устройств.

Методические указания:

Периферийные шины используются в основном для внешних запоминающих устройств.

Интерфейс IDE (Integrated Drive Electronics) – интерфейс устройств со встроенным контроллером. Поддерживает несколько способов обмена. Первый способ производит обмен данными через регистры процессора под его непосредственным управлением. Следствием этого является высокая загрузка процессора при операциях ввода/вывода. Вторым способом является использование режима прямого доступа к памяти, при котором контроллер интерфейса IDE и контроллер прямого доступа к памяти материнской платы пересылают данные между диском и оперативной памятью, не загружая центральный процессор. В целях развития возможностей интерфейса IDE была предложена его расширенная спецификация EIDE (синонимы ATA, ATA-2). Она поддерживает накопители емкостью свыше 504 Мбайт, поддерживает несколько накопителей IDE и позволяет подключать к одному контроллеру до четырех устройств, а также поддерживает периферийные устройства, отличные от жестких дисков. Расширение спецификации IDE для поддержки иных типов накопителей с интерфейсом IDE называют также ATAPI.

Интерфейс SCSI (Small Computer System Interface) - является стандартным интерфейсом для подключения приводов компакт-дисков, звуковых плат и внешних устройств массовой памяти. Спецификацией SCSI предусматривается параллельная передача данных по 8, 16 или 32 линиям данных. Структура SCSI, по существу, является магистральной, хотя устройства включаются в нее по принципу последовательной цепочки. Каждое SCSI-устройство имеет два разъема – один входной, а другой выходной. Все устройства объединяются в последовательную цепочку, один конец которой подключается к контроллеру интерфейса. Все

устройства работают независимо и могут обмениваться данными, как с компьютером, так и друг с другом. К шине SCSI можно подключить до 8 устройств, включая основной контроллер SCSI (хост-адаптер). Контроллер SCSI является, по сути, самостоятельным процессором и имеет свою собственную BIOS. К шине Wide SCSI подключается до 15 устройств.

ACPI (Advanced Configuration Power Interface – расширенный интерфейс конфигурирования и питания) – интерфейс, представляющий собой единую систему управления питанием для всех компонентов компьютера.

Задания на лабораторную работу:

Задание 1

Подключить жесткий диск к системной плате.

Задание 2

Подключить CD-ROM к системной плате.

Задание 3

Дать сравнительную характеристику периферийных устройств целевого компьютера. Определить их достоинства и недостатки.

Контрольные вопросы:

1. Перечислите интерфейсы накопителей и дайте их краткую характеристику.
2. Дайте сравнительную характеристику интерфейса IDE
3. Дайте сравнительную характеристику шины SCSI

Лабораторные работы № 10, 11

«Параллельные и последовательные порты и особенности их работы»

Цель работы: изучение особенностей работы параллельных и последовательных портов.

Методические указания:

Принтеры, модемы и другое периферийное оборудование подключаются к компьютеру через стандартизированные интерфейсы, называемые портами. В зависимости от способа передачи информации между сопряженными устройствами различают параллельные и последовательные интерфейсы.

Последовательный порт стандарта RS-232-C. Является стандартом для соединения ЭВМ с различными последовательными внешними устройствами. В операционных системах каждому порту RS-232 присваивается логическое имя COM1-COM4.

Параллельный порт используется для одновременной передачи 8 битов информации. В компьютерах этот порт используется главным образом для подключения принтера, графопостроителей и других устройств. Параллельные порты обозначаются LPT1-LPT4.

Интерфейс USB (Universal Serial Bus) – универсальная последовательная шина призвана заменить устаревший последовательный (COM-порт) и параллельный (LTP-порт) порты. Шина USB допускает подключение новых устройств без выключения компьютера. Шина сама определяет, что именно подключили к компьютеру, какой драйвер и ресурсы понадобятся устройству, после чего выделяет их без вмешательства пользователя. Шина USB позволяет подключить до 127 устройств.

IEEE 1394 (Institute of Electrical and Electronic Engineers 1394 – стандарт Института инженеров по электротехнике и электронике 1394) - последовательный интерфейс, предназначенный для подключения внутренних компонентов и внешних устройств. IEEE 1394 известен также под именем **FireWire** «огненный провод». Цифровой последовательный интерфейс FireWire характеризуется высокой

надежностью и качеством передачи данных, его протокол поддерживает гарантированную передачу критичной по времени информации, обеспечивая прохождение видео- и аудиосигналов в реальном масштабе времени без заметных искажений. При помощи шины FireWire можно подключить до 63 устройств и практически в любой конфигурации, чем она выгодно отличается от трудноконфигурируемых шин SCSI. Этот интерфейс используется для подключения жестких дисков, дисководов CD-ROM и DVD-ROM, а также высокоскоростных внешних устройств, таких как видеокамеры, видеомагнитофоны и т.д.

Задания на лабораторную работу:

Задание 1.

Определить внешние интерфейсы целевого компьютера.

Задание 2.

Подключить к целевому компьютеру принтеру.

Задание 3

Подключить к целевому компьютеру монитору

Задание 4

Подключить к целевому компьютеру сканеру.

Контрольные вопросы:

1. Какие типы внешних интерфейсов вы знаете?
2. Дайте сравнительную характеристику интерфейсов USB и IEEE 1384 (FireWire).
3. Дайте сравнительную характеристику параллельного и последовательного порта.
4. Что такое порты устройств?
5. Охарактеризуйте основные виды портов

Лабораторная работа № 12

Тема: «Программирование арифметических и логических команд»

Цель работы: знакомство с правилами построения последовательности машинных операций для реализации арифметических вычислений.

Методические указания.

Для кратковременного хранения и обработки данных в МП предусмотрены специальные ячейки памяти, конструктивно расположенные внутри МП и доступные из программ. Такие ячейки называются регистрами. Доступ к регистрам осуществляется гораздо быстрее, чем к ОП. В 8086 процессоре все регистры имеют разряд слова (16) и каждый имеет имя. В 386 -32. По назначению и способу использования регистры можно разделить на следующие группы.

5. РОИ (регистры общего назначения): AX, BX, CX, DX, SI, DI, BP, SP;
6. сегментные регистры: CS, DS, SS, ES;
7. регистр флагов;
8. регистр IP.

1. Регистры общего назначения.

К ним относятся: AX: AH/AL, BX: BH/BL, CX: CH/CL, DX: DH/DL, SI, DI, BP, SP.

AX - (accumulator) аккумулятор,

BX - (base) база,

CX - (counter) счетчик,

DX (data) данные, extended,

SI - (source index) индекс источника,

DI (destination index) индекс приемника

BP (base pointer) указатель базы,

SP (stack pointer).

ax		bx		cx		dx	
ah	al	bh	bl	ch	cl	dh	dl
Аккумулятор		База		Счетчик		Регистр данных	

В верхнем ряду таблицы находятся шестнадцатиразрядные регистры, которые могут хранить числа от 0 до 65 535 или от 0h до FFFFh в шестнадцатеричной системе.

В следующей строке расположен ряд восьмиразрядных регистров ah, al, bh, bl, ch, cl, dh, dl. В эти регистры можно загружать максимальное число 255 (FFh). Это так называемые половинки (старшая и младшая) шестнадцатиразрядных регистров.

Если шестнадцатеричное число начинается не с цифры (12h), а с буквы (C5h), то перед таким числом ставится ноль: 0C5h. Это необходимо для того, чтобы программа-ассемблер могла отличить, где шестнадцатеричное число, а где название переменной или метки.

РОН можно использовать в любых арифметических, логических операциях. Однако каждый РОН имеет определенную спецификацию: некоторые команды требуют, чтобы их операнд или операнды находились в определенных регистрах. Делимое - ah, dx.

2. Сегментные регистры.

Сегментные регистры CS, DS, SS и ES (Code, data, stack, extra segments) не могут быть операндами никаких команд, кроме команд пересылки и стековых команд. Эти регистры используются только для сегментирования адресов

CS	DS	ES	SS
Регистр кода	Регистр данных	Дополнительный регистр	Регистр стека

3. Регистр флагов.

В ПК имеется особый регистр флагов. Флаг - это бит, принимающий значение 1 ("флаг установлен"), если выполнено некоторое условие, и значение 0 ("флаг сброшен") в противном случае. В ПК используется 9 флагов, каждому из них присвоено определенное имя (ZF, CF и т.д.). Все они собраны в регистре флагов (каждый флаг - это один из разрядов регистра, часть его разрядов не используется):

Flags

x	x	x	x	x	OF	DF	IF	TF	SF	ZF	AF	x	PF	x	CF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Некоторые флаги принято называть флагами условий; они автоматически меняются при выполнении команд и фиксируют те или иные свойства их результата (например, равен ли он нулю). Другие флаги называются флагами состояний; они меняются из программы и оказывают влияние на дальнейшее поведение процессора

(например, блокируют прерывания).

Флаги условий:

CF (carry flag) - флаг переноса. Принимает значение 1, если при сложении целых чисел появилась единица переноса, не "влезаящая" в разрядную сетку, или если при вычитании чисел без знака первое из них было меньше второго. В командах сдвига в CF заносится бит, вышедший за разрядную сетку. CF фиксирует также особенности команды умножения.

OF (overflow flag) - флаг переполнения. Устанавливается в 1, если при сложении или вычитании целых чисел со знаком получился результат, по модулю превосходящий допустимую величину (произошло переполнение мантиссы и она "залезла" в знаковый разряд).

ZF (zero flag) - флаг нуля. Устанавливается в 1, если результат команды оказался равным 0.

SF (sign flag) - флаг знака. Устанавливается в 1, если в операции над знаковыми числами получился отрицательный результат.

PF (parity flag) - флаг четности. Равен 1, если результат очередной команды содержит четное количество двоичных единиц. Учитывается обычно только при операциях ввода-вывода.

AF (auxiliary carry flag) - флаг дополнительного переноса. Фиксирует особенности выполнения операций над двоично-десятичными числами.

Флаги состояний:

DF (direction flag) - флаг направления. Устанавливает направление просмотра строк в строковых командах: при DF=0 строки просматриваются "вперед" (от начала к концу), при DF=1 - в обратном направлении.

IF (interrupt flag) - флаг прерываний. При IF=0 процессор перестает реагировать на поступающие к нему прерывания, при IF=1 блокировка прерываний снимается.

TF (trap flag) - флаг трассировки. При TF=1 после выполнения каждой команды процессор делает прерывание (с номером 1), чем можно воспользоваться при отладке программы для ее трассировки.

4. Регистр IP.

Счетчик команд IP всегда содержит адрес (смещение от начала программы) той команды, которая должна быть выполнена следующей (начало программы хранится в регистре CS). Содержимое регистра IP можно изменить только командами

перехода.

5. Регистры-указатели.

si	di	bp	sp
Индекс источника	Индекс приемника	Регистры для работы со стеком	

Регистры Si и Di используются в строковых операциях. Регистры bp и sp задействуются при работе со стеком.

Команды процессора.

Простейшей, чаще всего используемой командой процессора, является команда

MOV Operand_1, Operand_2

Данная команда копирует содержимое правого операнда в левый операнд.

Эта команда, равно как и практически все остальные команды, оперирующие парой операндов, удовлетворяет двум важным условиям:

Результат операции сохраняется в левом операнде.

Ёмкости левого и правого операндов равны:

Operand_1: 1 байт Operand_2: 1 байт

или

Operand_1: 2 байта Operand_2: 2 байта

Для обозначения содержимого ячеек оперативной памяти при записи команд используются квадратные скобки. В квадратных скобках указывается не содержимое ячейки оперативной памяти, а адрес ячейки. Например:

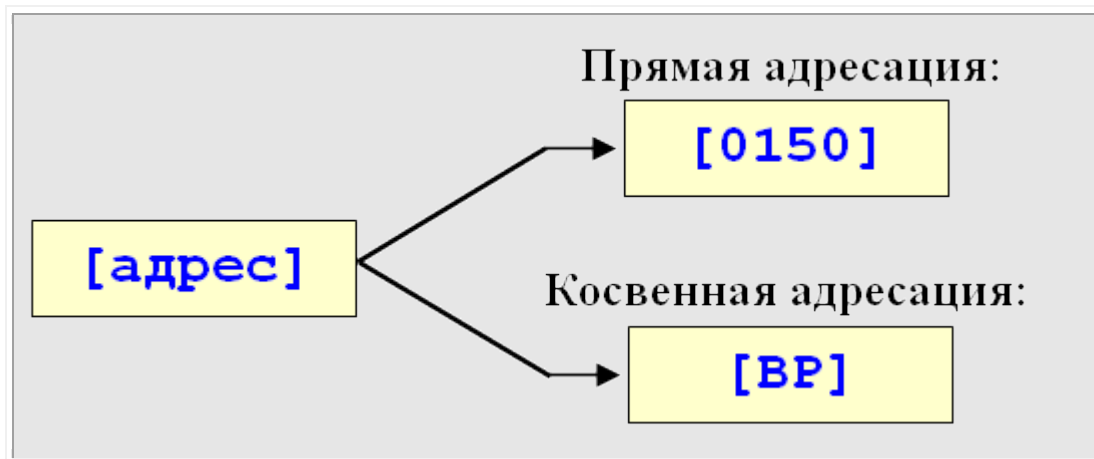
[0150] – **содержимое** 1-байтовой ячейки оперативной памяти.

0150 – **адрес** 1-байтовой ячейки оперативной памяти.

[BP] – **содержимое** 1-байтовой ячейки оперативной памяти.

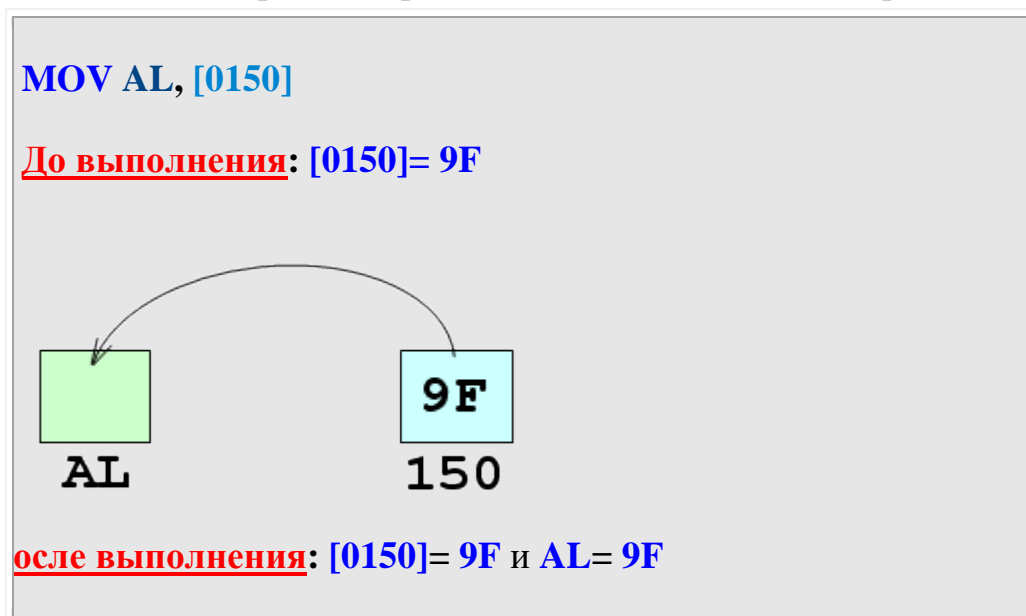
BP – регистр, содержащий **адрес** 1-байтовой ячейки памяти.

Адрес в квадратных скобках может задаваться двумя различными способами:

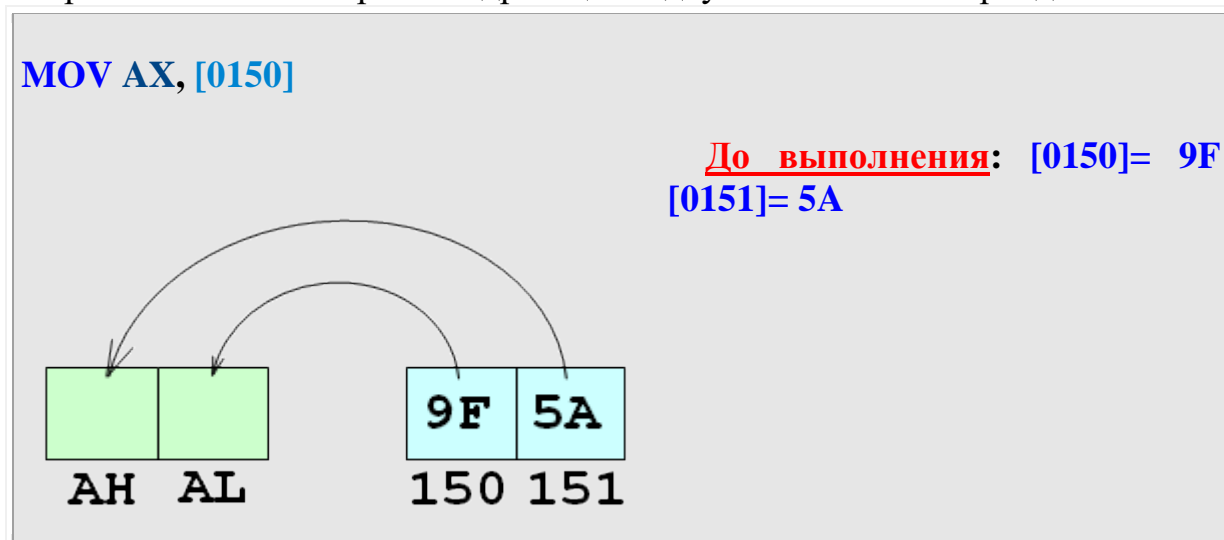


Прямая адресация.

Пример использования прямой адресации с однобайтовыми операндами:



Пример использования прямой адресации с двухбайтовыми операндами:



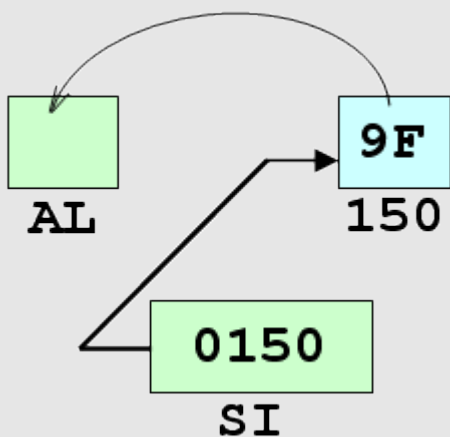
После выполнения: [0150]= 9F [0151]= 5A и AX= 5A9F, то есть AL= 9F
AH= 5A

Косвенная адресация.

Пример использования косвенной адресации с однобайтовыми операндами:

MOV AL, [SI]

До выполнения: [0150]= 9F SI= 0150

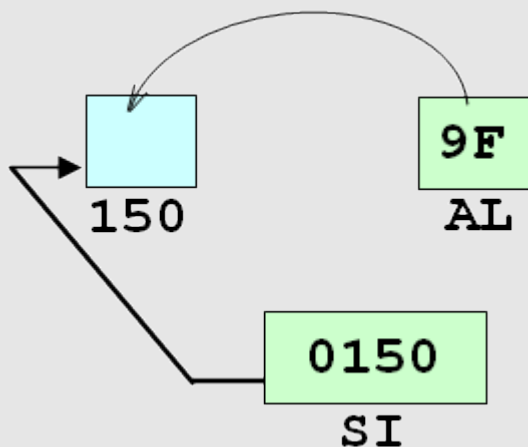


После выполнения: [0150]= 9F SI= 0150 и AL= 9F

Изменение порядка операндов отражается на результате следующим образом:

MOV [SI], AL

До выполнения: AL= 9F SI= 0150

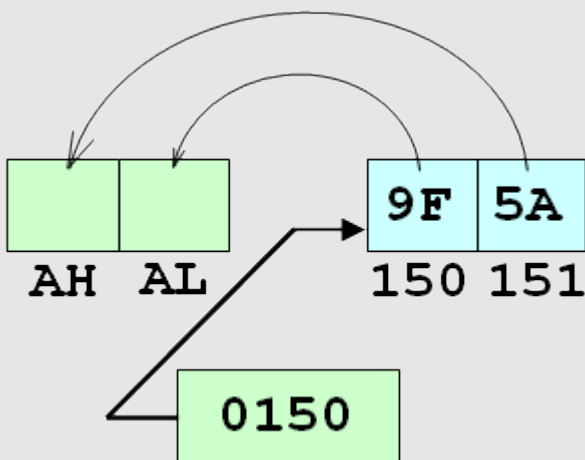


После выполнения: AL= 9F SI= 0150 и [0150]= 9F

Пример использования косвенной адресации с двухбайтовыми операндами:

MOV AX, [SI]

До выполнения: [0150]= 9F [0151]= 5A SI= 0150



После выполнения: [0150]= 9F [0151]= 5A SI= 0150 и AX= 5A9F, то есть AL= 9F и AH= 5A

Практическое использование способов адресации (на примере команды MOV):

Адрес начала команды	Команда, написанная на языке машинных кодов	Та же самая команда, записанная на языке ассемблера	Комментарий, поясняющий, как выполняется команда
100	B8 34 12	MOV AX,1234	НЕПОСРЕДСТВЕННАЯ АДРЕСАЦИЯ: Занесение 2-х байтового числа 1234 (12

			старший байт числа, 34 – младший байт (сла) в 2-х байтовый регистр AX
103	A3 20 01	MOV [0120],AX	<u>ПРЯМАЯ АДРЕСАЦИЯ:</u> Копирование 2-х байтового числа, ходящегося в AX , в однобайтовые ячейки оперативной памяти 0120 и 0121 (в 0120 копируется младший байт AX , в 21 копируется старший байт AX)
106	8B 1E 20 01	MOV BX,[0120]	<u>ПРЯМАЯ АДРЕСАЦИЯ:</u> Копирование содержимого ячеек оперативной памяти 0120 и 0121 , в 2-х байтовый регистр BX (в младший байт X , регистр BL , копируется содержимое ячейки 0120 , в старший байт BX , регистр CH , копируется содержимое ячейки 0121)
10A	BD 38 01	MOV BP,0138	<u>НЕПОСРЕДСТВЕННАЯ АДРЕСАЦИЯ:</u> Занесение 2-х байтового числа 0138 в 2-х байтовый регистр BP
10D	88 46 00	MOV [BP+00],AL	<u>КОСВЕННАЯ АДРЕСАЦИЯ:</u> Копирование однобайтового числа, ходящегося в AL , в однобайтовую ячейку оперативной памяти, адрес которой содержится в регистре BP , т.е. в ячейку с вычисляемым адресом 0138 (138+00=0138)
110	89 46 07	MOV [BP+07],AX	<u>КОСВЕННАЯ АДРЕСАЦИЯ:</u> Копирование 2-х байтового числа, ходящегося в AX , в ячейки оперативной памяти, на адреса которых указывает регистр BP : младший байт регистра AX , вычисляемый AL , копируется в ячейку с вычисляемым адресом 013F (0138 + 07 + 01 = 013F)

			старший байт регистра AX , именуемый H , копируется в ячейку с вычисляемым адресом 0140 (0138 + 07 + 1 = 0140)
113	CD 20	INT 20	Команда завершения работы программы

Практическое использование способов адресации (на примере команды ADD):

Адрес начала команды	Команда, написанная на языке машинных кодов	Та же самая команда, написанная на языке ассемблера	Комментарий, поясняющий, как выполняется команда
100	05 34 12	ADD AX,1234	<u>НЕПОСРЕДСТВЕННАЯ АДРЕСАЦИЯ:</u> Сложение 2-х байтового числа 1234 (12 – старший байт числа, 34 – младший байт числа) и содержимого 2-х байтового регистра AX . Результат размещается в регистре AX .
103	01 06 20 01	ADD [0120],AX	<u>ПРЯМАЯ АДРЕСАЦИЯ:</u> Сложение 2-х байтового числа, находящегося в AX , с 2-х байтовым числом, расположенном в ячейках оперативной памяти 0120 и 0121 (с содержимым ячейки 0120 складывается содержимое младшего байта AX , а с содержимым ячейки 0121 складывается содержимое старшего байта AX). Результат размещается в ячейках 0120 и 0121 .
107	03 1E 20 01	ADD BX,[0120]	<u>ПРЯМАЯ АДРЕСАЦИЯ:</u> Сложение содержимого ячеек оперативной памяти 0120 и 0121 с содержимым 2-х байтового регистра AX (к младшему байту регистра AX , к младшему байту регистра BX , добавляется содержимое младшего байта ячейки 0120 , а к старшему байту регистра BX , к старшему байту регистра BX , добавляется содержимое старшего байта ячейки 0121).

			<p>H, добавляется содержимое ячейки 21).</p> <p>Результат размещается в регистре BX.</p>
10B	BD 38 01	MOV BP,0138	<p><u>НЕПОСРЕДСТВЕННАЯ АДРЕСАЦИЯ:</u></p> <p>Занесение 2-х байтового числа 0138 в 2-х байтовый регистр BP</p>
10E	00 46 20	ADD [BP+20],AL	<p><u>КОСВЕННАЯ АДРЕСАЦИЯ:</u></p> <p>Сложение однобайтового числа, находящегося в AL, с содержимым 1-байтовой ячейки оперативной памяти, адрес которой определяется как сумма содержимого регистра BP и значения 20, т.е. с содержимым ячейки с вычисляемым адресом 0158 (138+20=0158).</p> <p>Результат размещается в ячейке 58.</p>
111	01 46 E0	ADD [BP-20],AX	<p><u>КОСВЕННАЯ АДРЕСАЦИЯ:</u></p> <p>Сложение 2-байтового числа, находящегося в AX с содержимым двух байт оперативной памяти, адресов которых вычисляются как BP-20:</p> <p>младший байт регистра AX, вычисляемый по адресу AL, складывается с содержимым ячейки, вычисляемый по адресу которой определяется как следующая сумма: 0118 (0138 - 20 + 0 = 0118)</p> <p>старший байт регистра AX, вычисляемый по адресу AH, складывается с содержимым ячейки с вычисляемым адресом 0119 (0138 - 20 + 1 = 0119)</p> <p>Результат размещается в ячейках 18 и 0119.</p>

Трассировка представленного выше кода :

Позволяет проследить, как выполняется операция сложения.

```

-u 100 113
0CC0:0100 053412      ADD     AX,1234
0CC0:0103 01062001     ADD     [0120],AX
0CC0:0107 031E2001     ADD     BX,[0120]
0CC0:010B BD3801      MOV     BP,0138
0CC0:010E 004620      ADD     [BP+20],AL
0CC0:0111 0146E0      ADD     [BP-20],AX
-d 118 150
0CC0:0110                00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0CC0:0120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0CC0:0130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0CC0:0140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0CC0:0150 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0CC0:0160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
-t=100 3
AX=1234  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0CC0  ES=0CC0  SS=0CC0  CS=0CC0  IP=0103  NU UP EI PL NZ NA PO NC
0CC0:0103 01062001     ADD     [0120],AX                DS:0120=0000
AX=1234  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0CC0  ES=0CC0  SS=0CC0  CS=0CC0  IP=0107  NU UP EI PL NZ NA PO NC
0CC0:0107 031E2001     ADD     BX,[0120]                DS:0120=1234
AX=1234  BX=1234  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0CC0  ES=0CC0  SS=0CC0  CS=0CC0  IP=010B  NU UP EI PL NZ NA PO NC
0CC0:010B BD3801      MOV     BP,0138
-D 120 110
0CC0:0120 34 12 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4.....

```

Первая команда **ADD AX,1234** приводит к тому, что **AX = 0 + 1234 = 1234**

Вторая команда **ADD [0120], AX** приводит к результату: **[0120] = [0120] + AX = 0 + 1234 = 1234** (**[0120] = 34** **[0121] = 12**)

Третья команда **ADD BX,[0120]** приводит к результату: **BX = BX + 1234 = 1234**

Трассировка второй группы команд даёт следующие результаты:

```

-t=10B 3
AX=1234 BX=1234 CX=0000 DX=0000 SP=FFEE BP=0138 SI=0000 DI=0000
DS=0CC0 ES=0CC0 SS=0CC0 CS=0CC0 IP=010E NU UP EI PL NZ NA PO NC
0CC0:010E 004620 ADD [BP+20],AL SS:0158=00

AX=1234 BX=1234 CX=0000 DX=0000 SP=FFEE BP=0138 SI=0000 DI=0000
DS=0CC0 ES=0CC0 SS=0CC0 CS=0CC0 IP=0111 NU UP EI PL NZ NA PO NC
0CC0:0111 0146E0 ADD [BP-20],AX SS:0118=0000

AX=1234 BX=1234 CX=0000 DX=0000 SP=FFEE BP=0138 SI=0000 DI=0000
DS=0CC0 ES=0CC0 SS=0CC0 CS=0CC0 IP=0114 NU UP EI PL NZ NA PO NC
0CC0:0114 06 PUSH ES
-d 118 15f
0CC0:0110 34 12 00 00 00 00 00 00 00 00 00 00 00 00 4.....
0CC0:0120 34 12 00 00 00 00 00 00 00 00 00 00 00 00 4.....
0CC0:0130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0CC0:0140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0CC0:0150 00 00 00 00 00 00 00 00 00 34 00 00 00 00 00 .....4.....

```

Первая команда **MOV BP,138** приводит к результату: **BP= 0138**

Вторая команда **ADD [BP+20],AL** приводит к результату: адрес= **0138 + 20 = 0158**. Далее, **[0158]= [0158] + AL = 0 + 34= 34**

Третья команда **ADD [BP-20],AX** приводит к результату: адрес= **0138 - 20 = 0118**. Далее, **[0118]= [0118] + AL = 0 + 34= 34** и **[0119]= [0119] + AH = 0 + 12= 12**

Оператор **sub** (команда вычитания).

Команда **sub** производит вычитание двух чисел.

Команда	Перевод	Назначение	Процессор
sub приемник, источник	Subtraction — вычитание	Вычитание	8086

Оператор **inc** (команда увеличения на 1).

Команда	Перевод	Назначение	Процессор
inc приемник	Increment — инкремент	Увеличение на единицу	8086

Команда **inc** увеличивает на единицу содержимое приемника (регистра или ячейки памяти). Она эквивалентна команде: **add источник, 1**, только выполняется быстрее и занимает меньше байт.

Оператор **dec** (команда уменьшения на 1).

Команда	Перевод	Назначение	Процессор
dec приемник	Decrement – декремент	Уменьшение на единицу	8086

Команда **dec** уменьшает на единицу содержимое приемника (регистра или ячейки памяти). Она эквивалентна команде: **sub источник, 1**, только выполняется быстрее и занимает меньше байт.

Задания к лабораторной работе:

Выполнить следующие задания:

3. Изучить содержание практической работы.
4. Рассмотреть выполнение разобранных в работе примеров.

1 Чем завершится выполнение следующей группы команд:

```

MOV     BX,80A0
MOV     AX,90E0
DD      BX,BX
DD      AX,BX
MOV [BX],AX
    
```

Результат выполнения группы команд записать в виде таблицы:

	H	L
AX	?	?
BX	?	?

2 Чем завершится выполнение следующей группы команд:

```

MOV     DX,     E0F0
MOV     CX,     A0C0
DD      CX,     DX
DD      DX,     CX
MOV [128], DX
    
```

Результат выполнения группы команд записать в виде таблицы:

	H	L
CX	?	?
DX	?	?
0128	?	
0129	?	

3 Чем завершится выполнение следующей группы команд:

```

MOV     DX,     E0B0
MOV     AX,     F0D0
DD      AX,     DX
DD      AH,     AL
DD      AL,     AH
MOV [132], AX
    
```

Результат выполнения группы команд записать в виде таблицы:

	H	L
AX	?	?
DX	?	?
0132	?	
0133	?	

4 Чем завершится выполнение следующей группы команд:

```

MOV     CX,     7090
MOV     BX,     9070
DD      BX,     BX
DD      CH,     BL
DD      CL,     BH
MOV [BX], CH
    
```

Результат выполнения группы команд записать в виде таблицы:

	H	L
AX	?	?
BX	?	?
CX	?	?

5	<p>Чем завершится выполнение следующей группы команд:</p> <pre> MOV BP, 190 MOV CX, 9070 DD CX, BP DD CH, CL DD CL, CH MOV [BP], CX </pre>	<p>Результат выполнения группы команд записать в виде таблицы:</p> <table border="1" data-bbox="868 248 1412 443"> <tr> <td></td> <td>H</td> <td>L</td> </tr> <tr> <td>BX</td> <td>?</td> <td>?</td> </tr> <tr> <td>CX</td> <td>?</td> <td>?</td> </tr> <tr> <td>BP</td> <td colspan="2">?</td> </tr> </table>		H	L	BX	?	?	CX	?	?	BP	?	
	H	L												
BX	?	?												
CX	?	?												
BP	?													
6	<p>Рассмотреть все предыдущие примеры (1-5), только вместо команды сложения (add) использовать команду вычитания (sub).</p>	<p>Результат выполнения группы команд записать в соответствующие для каждого примера таблицы:</p>												
7	<p>Пользуясь правилами формирования ассемблерных программ, напишите программу расчета значения A. (таблица №1, см. ниже).</p>	<p>Записать листинг программы и результат выполнения.</p>												

Таблица №1

№ варианта	Расчетная формула	B	C	D
1, 6, 11	$A=B+C-D$	1	35	23
2, 7, 12	$A=B+C+D$	65	1	1
3, 8, 13	$A=C-D+B$	1	33	1
4, 9, 14	$A=D+A-B$	18	1	88
5, 10, 15	$A= B-C+D$	45	10	1

Контрольные вопросы:

1. Команда add основное назначение?
2. Команда sub основное назначение?
3. Команда inc основное назначение?
4. Команда dec основное назначение?

Лабораторная работа №13.

Тема: «Программирование переходов»

Цель работы: изучение команд безусловных и условных переходов.

Условные обозначения:

При описании команд используются следующие обозначения:

- op1, op2 – операнды;
- imm или im – непосредственный операнд (операнд, указанный в самой команде);
- reg – регистр;
- mem – память.

Безусловный переход

Безусловный переход — это переход, который выполняется всегда. Безусловный переход осуществляется с помощью команды **JMP**. У этой команды один операнд, который может быть непосредственным адресом (меткой), регистром или ячейкой памяти, содержащей адрес.

Примеры безусловных переходов:

jmp metka ; *Переход на метку*

jmp bx ; *Переход по адресу в ВХ*

jmp word[bx] ; *Переход по адресу, содержащемуся в памяти по адресу в ВХ*

Условный переход

Условный переход осуществляется, если выполняется определённое условие, заданное флагами процессора (кроме одной команды, которая проверяет СХ на равенство нулю). Состояние флагов изменяется после выполнения арифметических, логических и некоторых других команд. Если условие не выполняется, то управление переходит к следующей команде.

Существует много команд для различных условных переходов. Также для некоторых команд есть синонимы (например, **JZ** и **JE** - это одно и то же). Для наглядности все команды условных переходов приведены в таблице:

Команда	Переход, если	Условие перехода
---------	---------------	------------------

JZ/JE	нуль или равно	ZF=1
JNZ/JNE	не нуль или не равно	ZF=0
JC/JNAE/JNB	есть переполнение/не выше и не ниже	CF=1
JNC/JAE/JNB	нет переполнения/выше или равно/ниже	CF=0
JP	число единичных бит чётное	PF=1
JNP	число единичных бит нечётное	PF=0
JS	знак равен 1	SF=1
JNS	знак равен 0	SF=0
JO	есть переполнение	OF=1
JNO	нет переполнения	OF=0
JA/JNBE	выше/не ниже и не равно	CF=0 и ZF=0
JNA/JBE	не выше/ниже или равно	CF=1 или ZF=1
JG/JNLE	больше/не меньше и не равно	ZF=0 и SF=OF
JGE/JNL	больше или равно/не меньше	SF=OF
JL/JNGE	меньше/не больше и не равно	SF≠OF
JLE/JNG	меньше или равно/не больше	ZF=1 или SF≠OF
JCXZ	содержимое CX равно нулю	CX=0

У всех этих команд один операнд — имя метки для перехода. Обратите внимание, что некоторые команды применяются для беззнаковых чисел, а другие — для чисел со знаком. Сравнения «выше» и «ниже» относятся к беззнаковым числам, а «больше» и «меньше» — к числам со знаком. Для беззнаковых чисел признаком переполнения будет флаг CF, а соответствующими командами перехода [JC](#) и [JNC](#). Для чисел со знаком о переполнении можно судить по состоянию флага OF, поэтому им соответствуют команды перехода [JO](#) и [JNO](#). Команды переходов не изменяют значения флагов.

В качестве примера рассмотрим небольшую программу для сложения двух чисел со знаком с проверкой переполнения. В случае переполнения будет выводиться сообщение об ошибке. Вы можете поменять значения объявленных переменных, чтобы переполнение возникало или не возникало при их сложении, и посмотреть, что будет выводить программа.

```

1  org 100h      ;Программа начинается с адреса 100h
2
3  mov al, [x]   ;AL = x

```



```

4      add al, [y]           ;AL = x + y
5      jo error             ;Переход, если переполнение
6      mov ah, 09h         ;\
7      mov dx, offset ok_msg ; > Вывод строки 'OK'
8      int 21h             ;/
9  exit:
1      mov ah, 09h         ;\
      mov dx, offset pak   ; > Вывод строки 'Press any key.'
1      int 21h             ;/

1      mov ah, 08h         ;\
      int 21h             ;/ Ввод символа

1
      mov ax, 4C00h        ;\
1      int 21h            ;/ Завершение программы
error:
1      mov ah, 09h         ;\
      mov dx, offset err_msg; > Вывод сообщения об ошибке
1      int 21h            ;/
      jmp exit             ;Переход на метку exit;
1

1  x          db -89
      y          db -55
1  err_msg    db 'Error: overflow detected.',13,10,'$'
      ok_msg     db 'OK',13,10,'$'
2  pak        db 'Press any key...$'

2

2

2

2

2

2

2

2

```

Команды CMP и TEST

Часто для формирования условий переходов используются команды [CMP](#) и [TEST](#). Команда [CMP](#) предназначена для сравнения чисел. Она выполняется аналогично команде [SUB](#): из первого операнда вычитается второй, но результат не записывается на место первого операнда, изменяются только значения флагов. Например:

```

cmp al,5      ;Сравнение AL и 5
jl  c1       ;Переход, если AL < 5 (числа со
;знаком)
cmp al,5      ;Сравнение AL и 5
jb  c1       ;Переход, если AL < 5 (числа без
;знака)

```

Команда [TEST](#) работает аналогично команде [AND](#), но также результат не сохраняется, изменяются только флаги. С помощью этой команды можно проверить состояние различных битов операнда. Например:

```

test bl,00000100b ;Проверить состояние 2-го бита
;
jz  c2           ;Переход, если 2-й бит равен 0

```

Пример программы

Простая программка, которая выводит меню и предлагает пользователю сделать выбор. Для ввода символа используется функция DOS 01h (при вводе символ отображается на экране). В зависимости от введённого символа осуществляется переход на нужную часть кода. Данные записаны в начале программы, а не в конце. Чтобы данные не выполнились как код, перед ними стоит команда безусловного перехода.

```

1  org 100h      ;Программа начинается с адреса
2  10h
3      jmp start ;Безусловный переход на метку
4  start
5  ;-----
6  ----
7  menu      db '1 - Print hello',13,10
8            db '2 - Print go away',13,10
9            db '0 - Exit',13,10,'$'
10 select   db 13,10,'Select>$'

```

```

11 hello    db 13,10,'Hello!',13,10,13,10,'$'
12 go_away db 13,10,'Go away!',13,10,13,10,'$'
13 ;-----
14 ----
15 start:
16     mov ah,09h           ;\
17     mov dx,menu         ; > Вывод меню
18     int 21h             ;/
19
20 select_loop:
21     mov ah,09h           ;\
22     mov dx,select       ; > Вывод строки 'Select>'
23     int 21h             ;/
24
25     mov ah,01h           ;Функция DOS 01h - ввод
26     ;символа
27     int 21h             ;Введённый символ помещается в
28     ;
29
30     cmp al,'1'          ;Сравнение введённого символа
31     '1'
32     je c1               ;Переход, если равно
33     cmp al,'2'          ;Сравнение введённого символа
34     '2'
35     je c2               ;Переход, если равно
36     cmp al,'0'          ;Сравнение введённого символа
37     '0'
38     je exit             ;Переход, если равно
39     jmp select_loop     ;Безусловный переход
40 c1:
41     mov ah,09h           ;\
42     mov dx,hello        ; > Вывод строки 'Hello'
43     int 21h             ;/
44     jmp start           ;Безусловный переход
45 c2:
46     mov ah,09h           ;\
47     mov dx,go_away      ; > Вывод строки 'Go away'
48     int 21h             ;/
49     jmp start           ;Безусловный переход
50 exit:
51     mov ax,4C00h        ;\
52     int 21h             ;/ Завершение программы

```

Скриншот работы программы:

```

emulator screen (39x19 chars)
1 - Print hello
2 - Print go away
0 - Exit

Select>1
Hello!

1 - Print hello
2 - Print go away
0 - Exit

Select>2
Go away!

1 - Print hello
2 - Print go away
0 - Exit

Select>0

```

Задания к работе:

Выполнить следующие задания:

5. Изучить содержание практической работы.
6. Рассмотреть выполнение разобранных в работе примеров.
7. Из таблицы взять задание и написать программу, используя операторы условного и безусловного переходов.

№ варианта	Задание
1.	$y = \begin{cases} a - b, & \text{если } a > 0 \\ a + b + 2, & \text{если } b > 0 \end{cases}$
2.	$y = \begin{cases} a + 1 - b, & \text{если } a - b > 0 \\ a - 1, & \text{если } b < 0 \end{cases}$
3.	$y = \begin{cases} a + b, & \text{если } a < 30 \\ a - b, & \text{иначе} \end{cases}$
4.	$y = \begin{cases} a - 2 + b, & \text{если } a < 10 \\ a + b - 1, & \text{еес } b > 10 \end{cases}$
5.	$y = \begin{cases} 2 + b - a, & \text{если } a < b \\ a + 1, & \text{иначе} \end{cases}$

6.	$y = \begin{cases} a+b+5, & \text{если } b > 0 \\ a+b-2, & \text{иначе} \end{cases}$
7.	$y = \begin{cases} a-2-b, & \text{если } b-3 > 0 \\ b-1, & \text{иначе} \end{cases}$
8.	$y = \begin{cases} a+b, & \text{если } b < a \\ a+b+18, & \text{иначе} \end{cases}$
9.	$y = \begin{cases} a-1+b, & \text{если } b < 20 \\ a+b+1, & \text{иначе} \end{cases}$
10.	$y = \begin{cases} a-b, & \text{если } b < 21 \\ a+b-1, & \text{иначе} \end{cases}$
11.	$y = \begin{cases} a-b, & \text{если } (a-b) > 0 \\ a-1, & \text{иначе} \end{cases}$
12.	$y = \begin{cases} a+b, & \text{если } (a-b) > 0 \\ a+1, & \text{иначе} \end{cases}$
13.	$y = \begin{cases} a+b, & \text{если } a < 0 \\ a+b-1, & \text{если } b > 0 \end{cases}$
14.	$y = \begin{cases} a-b, & \text{если } a > 0 \\ a+b+1, & \text{если } b > 0 \end{cases}$
15.	$y = \begin{cases} a+b, & \text{если } a > 0 \\ a+2, & \text{если } b > 0 \end{cases}$

Лабораторная работа № 14, 15

Тема: «Программирование ввода-вывода. Программирование и отладка программ».

Цель работы: ознакомиться с основными средствами вывода текстовых данных на экран посредством средств операционной системы DOS, средствами BIOS и средствами непосредственного (прямого) отображением в видеобуфер.

I. Средства DOS.

1. Функция DOS 02h - записать символ в STDOUT (существуют 3 файловых дескриптора (Дескриптор «описатель» - число, которое задает номер какого-либо ресурса (блок памяти, файл, курсор, шрифт, окно и тп.) при работе с ним.): *stdin* - стандартный ввод, *stdout* - стандартный вывод и *stderr* - стандартный поток ошибок) с проверкой на Ctrl-Break

Ввод:	AH = 02h DL = ASCII-код символа
Вывод:	Никакого, согласно документации, но на самом деле: AL = код последнего записанного символа (равен DL, кроме случая, когда L = 09h (табуляция), тогда в AL возвращается 20h).

Эта функция при выводе на экран обрабатывает некоторые управляющие символы - вывод символа BEL (07h) приводит к звуковому сигналу, символ BS (08h) приводит к движению курсора влево на одну позицию, символ HT (09h) заменяется на несколько пробелов, символ LF (0Ah) опускает курсор на одну позицию вниз, и CR (0Dh) приводит к переходу на начало текущей строки.

Если в ходе работы этой функции была нажата комбинация клавиш Ctrl+Break, вызывается прерывание 23h, которое по умолчанию осуществляет выход из программы.

Простой пример работы функции DOS 02h.

Пример № 1

org 100h	; начало COM-файла
begin:	; метка начала кода программы
mov dl,< ASCII-код символа >	; заносим в регистр dl - любой ASCII-код символа
mov ah,02h	; номер функции DOS "вывод символа"
int 21h	; вызов DOS
ret	; функция DOS "завершить программу"
end begin	; метка окончания кода программы

Эта программа, выводит на экран любой ASCII-символ, в установленную позицию курсора.

2. Функция DOS 06h - записать символ в STDOUT без проверки на Ctrl+Break

Ввод:	AH = 06h DL = ASCII-код символа (кроме FFh)
Вывод:	Никакого, согласно документации, но на самом деле: AL = код записанного символа (копия DL)

Эта функция не обрабатывает управляющие символы (CR, LF, HT и BS выполняют свои функции при выводе на экран, но сохраняются при перенаправлении вывода в файл) и не проверяет нажатие Ctrl+Break.

Замените в примере № 1 MOV AH, 02h на MOV AH, 06h и перекомпилируем этот пример. Работу откомпилированного примера смотрим в операционной системе MS-DOS.

3. Функция DOS 09h – записать строку в STDOUT с проверкой на Ctrl+Break

Ввод:	AH = 09h DS: DX = адрес строки, заканчивающейся
-------	---

	символом \$ (24h)
Вывод:	Никакого, согласно документации, но на самом деле: AL = 4h (код последнего символа)

Действие этой функции полностью аналогично действию функции 02h, но выводится не один символ, а целая строка.

4. Функция DOS 40h – записать в файл или устройство

Ввод:	AH = 40h BX = 1 для STDOUT или 2 для STDERR DS: DX = адрес начала строки CX = длина строки
Вывод:	CF = 0, AX = число записанных байтов

Эта функция предназначена для записи в файл, но, если в регистр BX поместить число 1, функция 40h будет выводить данные на STDOUT, а если BX = 2 - на устройство STDERR. STDERR всегда выводит данные на экран и не перенаправляется в файлы.

Простой пример работы функции DOS 40h.

Пример № 2

org 100h	; начало COM-файла
begin:	; метка начала кода программы
mov ah,40h	; номер функции DOS
mov bx,2	; указываем устройство STDERR
mov dx,offset message	; DS: DX - адрес строки
mov cx,25	; CX - длина строки
int 21h	; вызов DOS
ret	; функция DOS "завершить программу"
message db "This function can rint \$"	; строка содержащая выводимые данные.
end begin	; метка окончания кода программы

Прерывание INT 29H

INT 29h: Быстрый вывод символа на экран

Ввод:	AL = ASCII-код символа
-------	------------------------

Простой пример работы прерывания INT 29h.

Пример № 3

org 100h	; начало COM-файла
begin:	; метка начала кода программы
mov ah, < ASCII-код символа >	; заносим в регистр ah - любой ASCII-код символа
int 29h	; вызов прерывания DOS - вызов символа;
ret	; функция DOS "завершить программу"
end begin	; метка окончания кода программы

В большинстве случаев INT 29h просто немедленно вызывает функцию BIOS "вывод символа на экран в режиме телетайпа", так что никаких преимуществ, кроме экономии байт при написании как можно более коротких программ, она не имеет.

II. Средства BIOS

Функции DOS вывода на экран позволяют перенаправить вывод в файл, но не позволяют вывести текст в любую позицию экрана и не позволяют изменить цвет текста. DOS предполагает, что для более тонкой работы с экраном программы должны использоваться видеofункции BIOS. BIOS - обеспечивает доступ к некоторым устройствам, в частности к видеоадаптеру. Все функции видеосервиса BIOS вызываются через **прерывание 10h**.

1. Выбор видеорежима

BIOS предоставляет возможность переключения экрана в различные текстовые и графические режимы. Режимы отличаются друг от друга разрешением (для графических) и количеством строк и столбцов (для текстовых), а также количеством возможных цветов.

Стандартные видеорежимы:

INT 10h, AH = 00 - Установить видеорежим

Ввод:	AL = номер режима в младших 7 битах
Вывод:	Обычно никакого, но некоторые BIOS (Phoenix и AMI) помещают в AL 30h для текстовых режимов и 20h для графических

Пример работы.

Пример № 4

org 100h	; начало COM-файла
begin:	; метка начала кода программы
mov ah,00	; устанавливаем видеорежим

mov al,5	; устанавливаем номер режима
int 10h	; вызов прерывания DOS - вызов видеосервиса;
ret	; функция DOS "завершить программу"
end begin	; метка окончания кода программы

Вызов этой функции приводит к тому, что экран переводится в выбранный режим. Если старший бит AL не установлен в 1, экран очищается. Номера текстовых режимов - 0, 1, 2, 3 и 7.0 и 1 - 16-цветные режимы 40x25 (с 25 строками по 40 символов в строке), 2 и 3 - 16-цветные режимы 80x25, 7 - монохромный режим 80x25. Существует еще много текстовых режимов с более высоким разрешением (80x43, 80x60, 132x50 и т.д.), но их номера для вызова через эту функцию различны для разных видеоадаптеров (например, режим 61h - 132x50 для Cirrus 5320 и 132x29 для Genoa 6400). Однако, если видеоадаптер поддерживает стандарт VESA BIOS Extension, в режимы с высоким разрешением можно переключаться, используя функцию 4Fh.

2. SuperVGA-видеорежим

INT 10h, AH = 4Fh, AL = 02 - Установить SuperVGA-видеорежим

Ввод:	BX = номер режима в младших 13 битах
Вывод:	AL = 4Fh, если эта функция поддерживается AH = 0, если переключение произошло успешно AH = 1, если произошла ошибка

Если бит 15 регистра BX установлен в 1, видеопамять не очищается. Текстовые режимы, которые можно вызвать с использованием этой функции: 80x60 (режим 108h), 132x25 (109h), 132x43 (10Ah), 132x50 (10Bh), 132x60 (10Ch).

Видеорежим, используемый в DOS по умолчанию, - текстовый режим.

Управление положением курсора

1. Устанавливаем положение курсора

INT 10h, AH = 02 - Установить положение курсора

Ввод:	AH = 02 BH = номер страницы DH = строка DL = столбец
-------	--

Пример работы.

Пример № 5

org 100h	; начало COM-файла
begin:	; метка начала кода программы

mov ah,02	; устанавливаем положение курсора
mov bh,0	; устанавливаем номер страницы
mov dh,12	; строка 12
mov dl,29	; столбец 29
int 10h	; прерывания DOS - установить положение курсора в точку 12,29
mov ax, < ASCII-код символа >	; заносим в регистр ax - любой ASCII-код символа
int 29h	; вызов прерывания DOS - вызов символа;
ret	; функция DOS "завершить программу"
end begin	; метка окончания кода программы

С помощью этой функции можно установить курсор в любую позицию экрана, и дальнейший вывод текста будет происходить из этой позиции. Отсчет номера строки и столбца ведется от верхнего левого угла экрана (символ в левой верхней позиции имеет координаты 0,0). Номера страниц 0 - 3 (для режимов 2 и 3) и 0 - 7 (для режимов 1 и 2) соответствуют области памяти, содержимое которой в данный момент отображается на экране. Можно вывести текст в неактивную в настоящий момент страницу, а затем переключиться на нее, чтобы изображение изменилось мгновенно.

2. Считываем положение и размер курсора

INT 10h, AH = 03 - Считать положение и размер курсора

Ввод:	AH = 03 BH = номер страницы
Вывод:	DH, DL = строка и столбец текущей позиции курсора CH, L = первая и последняя строки курсора

Возвращает текущее состояние курсора на выбранной странице (каждая страница использует собственный независимый курсор).

3. Вывод символов на экран

Каждый символ на экране описывается двумя байтами - ASCII-кодом символа и байтом атрибута, указывающим цвет символа и фона, а также является ли символ мигающим.

Атрибут символа:

Бит 7: символ мигает (по умолчанию) или фон яркого цвета (если его действие было переопределено видеофункцией 10h).

Биты 6 - 4: цвет фона. Бит 3: символ яркого цвета (по умолчанию) или фон мигает

(если его действие было переопределено видеофункцией 11h).

Биты 2 - 0: цвет символа.

Цвета кодируются в битах, как показано в таблице №1.

Таблица №1. Атрибуты символов

	Обычный цвет	Яркий цвет
000b	черный	темно-серый
001b	синий	светло-синий
010b	зеленый	светло-зеленый
011b	голубой	светло-голубой
100b	красный	светло-красный
101b	пурпурный	светло-пурпурный
110b	коричневый	желтый
111b	светло-серый	белый

4. Считываем символ и атрибут символа в текущей позиции курсора.

INT 10h, AH = 08 - Считать символ и атрибут символа в текущей позиции курсора

Ввод:	AH = 08 BH = номер страницы
Вывод:	AH = атрибут символа AL = ASCII-код символа

5. Выводим символ с заданным атрибутом на экран

INT 10h, AH = 09 - Вывести символ с заданным атрибутом на экран

Ввод:	AH = 09 BH = номер страницы AL = ASCII-код символа BL = атрибут символа CX = число повторений символа
-------	---

С помощью этой функции можно вывести на экран любой символ, включая даже символы CR и LF, которые обычно интерпретируются как конец строки. В графических режимах CX не должен превышать число позиций, оставшееся до правого края экрана.

Пример работы.

Пример № 6

org 100h	; начало COM-файла
begin:	; метка начала кода программы
mov ah,09	; помещаем номер функции DOS "вывод"

	строки (9)" в регистр AH.
mov bh,0	; устанавливаем номер страницы
mov al, < ASCII-код символа >	; строка 12; заносим в регистр al - любой ASCII-код символа
mov bl, 00011111b	; атрибут символа (белый на голубом)
mov cx,555	; устанавливаем в счетчик кол-во выводимых символов
int 10h	; вызов прерывания DOS - вызов символа;
ret	; функция DOS "завершить программу"
end begin	; метка окончания кода программы

6. Выводим символ с текущим атрибутом на экран
INT 10h, AH = 0Ah - Вывести

Ввод:	AH = 0Ah BH = номер страницы AL = ASCII-код символа CX число повторений символа
-------	--

Эта функция также выводит любой символ на экран, но в качестве атрибута символа используется атрибут, который имел символ, находившийся ранее в этой позиции.

7. Выводим символ в режиме телетайпа
INT 10h, AH = 0Eh - Вывести символ в режиме телетайпа

Ввод:	AH = 0Eh BH = номер страницы AL = ASCII-код символа
-------	---

Символы CR (0Dh), LF (0Ah), BEL (7) интерпретируются как управляющие символы. Если текст при записи выходит за пределы нижней строки, экран прокручивается вверх. В качестве атрибута используется атрибут символа, находившегося в этой позиции.

8. Выводим строку символов с заданными атрибутами
INT 10h, AH = 13h - Вывести строку символов с заданными атрибутами

Ввод:	AH = 13h AL = режим вывода: Бит 0 - переместить курсор в конец строки после вывода Бит 1 - строка содержит не только символы, но также и атрибуты, так что каждый символ описывается двумя байтами: ASCII-код и
-------	--

	атрибут биты 2 - 7 зарезервированы CX = длина строки (только число символов) BL = атрибут, если строка содержит только символы DH, DL = строка и голбец, начиная с которых будет выводиться строки ES: BP = адрес начала строки в памяти
--	--

Функция 13h выводит на экран строку символов, интерпретируя управляющие символы CR (0Dh), LF (0Ah), BS (08) и BEL (07). Если строка подготовлена в формате символ, атрибут - гораздо быстрее просто скопировать ее в видеопамять.

Функции BIOS удобны для переключения и настройки видеорежимов, но часто оказывается, что вывод текста на экран гораздо быстрее и проще выполнять просто копированием изображения в видеопамять.

III. Прямая работа с видеопамятью

Все, что изображено на мониторе - и графика, и текст, одновременно присутствует в памяти, встроенной в видеоадаптер. Для того чтобы изображение появилось на мониторе, оно должно быть записано в память видеоадаптера. Для этого отводится специальная область памяти, начинающаяся с абсолютного адреса 0B800h: 0000h (для текстовых режимов) и заканчивающаяся на 0B800h: FFFFh. Все, что программы пишут в эту область памяти, немедленно пересылается в память видеоадаптера. В текстовых режимах для хранения каждого изображенного символа используются два байта: байт с ASCII-кодом символа и байт с его атрибутом, так что по адресу 0B800h: 0000h лежит байт с кодом символа, находящимся в верхнем левом углу экрана; по адресу 0B800h: 0001h лежит атрибут этого символа; по адресу 0B800h: 0002h лежит код второго символа в верхней строке экрана и т.д.

Таким образом, любая программа может вывести текст на экран, простой командой пересылки данных, не прибегая ни к каким специальным функциям DOS или BIOS.

Пример работ с видеопамятью.

Пример №7.

org 100h	; начало COM-файла
begin:	; метка начала кода программы
mov ax,0003h	; видеорежим 3 (очистка экрана)
int 10h	; прерывание DOS - очистка экрана;
mov ax,0B800h	; загружаем в сегментный регистр ES
mov es,ax	исло 0B800h
mov di,0	; загружаем в регистр DI нуль
mov ah,31	; заносим в регистр ah - атрибут

	Символа
mov al, < ASCII-код символа	заносим в регистр al - любой ASCII-код символа
mov es: [di],ax	; заносим по адресу 0B800: 0000h атрибут и ASCII-код символа
mov ah,10h	; вызываем функцию 10h - чтобы можно было остановить программу до нажатия любой клавиши
int 16h	; вызываем прерывание 16h - сервис работы с клавиатурой BIOS
ret	; функция DOS "завершить программу"
end begin	; метка окончания кода программы

При подготовке данных для копирования в видеопамять в этой программе в строках (7) и (8) загружаем в сегментный регистр ES число 0B800h, которое соответствует сегменту дисплея в текстовом режиме. В строке (9) загружаем в регистр DI нуль. Это будет смещение относительно сегмента 0B800h. В строках (10) и (11) в регистр AH заносится атрибут символа (31 - ярко-белый символ на синем фоне) и в AL - ASCII-код символа (01 - рожица) соответственно.

В строке (12) заносим по адресу 0B800: 0000h (т.е. первый символ в первой строке дисплея - верхний левый угол) атрибут и ASCII-код символа (31 и 01 соответственно).

Задание для выполнения практической работы:

1. С помощью редактора эмулятора EMU8086 напишите программы приведенных выше примеров (от 1 до 7).
2. Напишите программу вывода на экран строки 'Hello' согласно своему варианту (таблица №2).
3. Напишите программу работы переключения SuperVGA-видеорежимов согласно своему варианту (таблица №3)
4. Подготовьте отчет, который должен содержать тексты программ, а также укажите описание работы команд программ.
5. В отчете должны содержаться ответы на следующие вопросы.

Контрольные вопросы

1. Перечислите функции вывода на экран средствами операционной системы DOS?
2. Принцип работы функции DOS 02h?
3. Укажите основные управляющие символы вывода на экран?
4. Укажите отличие функции DOS 02h от 06h?
5. Прерывание int 29h. Преимущества использования?
6. С помощью каких функций можно установить нужный видеорежим (текстовый, цветной, монохромный)?
7. Укажите функции и прерывания управления положением курсора.
8. Перечислите функции считывания положения и размера курсора?
9. Вывод символов на экран средствами BIOS. Функции?
10. Укажите преимущества вывода на экран с помощью непосредственной работы с видеопамятью?
11. Если в примере № 2 длину строку указать большую, чем указанная что в данном случае будет выводиться на экрана

Таблица. №2

№ варианта	Функция вывода (DOS)	Функция вывода (BIOS)	Видеопамять
1,6,11,16,21,5	02h	Ah=02h	‘Hello’
2,7,12,17,22,7	06h	Ah=08h	
3,8,13,18,23,8	09h	Ah=09h	
4,9,14,19,24,9	40h	Ah=0Ah	
5,10,15,20,25,0	29h	Ah=13h	

Примечание: В примерах, в которых возможно задание различных параметров вывода (цвет символа, фона; номер строки, столбца, страницы и т.д.) выводите на экран слово "hello" с параметрами отличными от стандартных.

Таблица. №3

№ варианта	SuperVGA-видеорежим
1,6,11,16,21,26	108h
2,7,12,17,22,27	109h
3,8,13,18,23,28	10Ah
4,9,14,19,24,29	10Bh
5,10,15,20,25,3	10Ch

Лабораторная работа № 16

Тема: «Идентификация и установка процессора»

Цель работы: изучение характеристик процессора.

Методические указания:

Основа вычислительной системы – микропроцессор (МП). МП характеризуется следующими параметрами:

Тактовая частота

- Степень интеграции микросхемы (сколько транзисторов содержится в чипе).
- Внутренняя разрядность данных (количество бит, которые МП может обрабатывать одновременно);
- Внешняя разрядность данных (количество одновременно передаваемых бит в процессе обмена данными с памятью и другими устройствами);
- Адресуемая память (зависит от числа адресных бит).

Типы МП

Тип процессора	Поколение	Год выпуска	Разрядность шины данных	Разрядность шины адреса	Первичная ш-память, байт		Тактовая частота шины, МГц	Тактовая частота процессора, Гц	Количество транзисторов, млн	Размер минимальной структуры, мкм
					Команды	Данные				
8088	1	1979	8	20			4,77-8	4,77-8		
8086	1	1978	16	20			4,77-8	4,77-8	0,029	3,0
80286	2	1982	16	24			6-20	6-20	0,13	1,5
80386DX	3	1985	32	32			16-33	16-33	0,27	1,0
80386SX	3	1988	16	32	8		16-33	16-33	0,27	1,0
80486DX	4	1989	32	32	8		25-50	25-50	1,2	1,0-0,8
80486SX	4	1989	32	32	8		25-50	25-50	1,1	0,8
80486DX2	4	1992	32	32	8		25-40	25-40		
80486DX4	5	1994	32	32	8	8	25-40	75-120		
Pentium	5	1993	64	32	8	8	60-66	60-200	3.1-3.3	0.8-

										35
P-MMX	5	1997	64	32	16	16	66	166-33	4.5	0.6-35
Pentium Pro	6	1995	64	32	8	8	66	150-00	5.5	0.35
Pentium II	6	1997	64	32	16	16	66	233-00	7.5	0.35-25
Pentium II eleron	6	1998	64	32	16	16	66/10	266-33	7.5-19	0.25
Pentium eon	6	1998					100	400-700		0.18
Pentium III	6	1999	64	32	16	16	100	450-200	9.5-44	0.25-13
AMD thlon	7	1999	64	32	64	64	266	500-200	22	0.25
Pentium 4	7	2000	64	32	12	8	400	1.4-3.4 Гц	42-125	0,18-09
AMD thlon 64	8	2003	64	64	64	64	400	2 ГГц	54-106	0,13-09
Pentium 4 rescott	8	2004	64	32	16	16	800	2.8-3.4 Гц	125	0.09

В сущности МП – плоский квадратный слой кремния со схемами, выгравированными на его поверхности. Этот элемент укрепляется на основе – керамической или пластмассовой – образуя пакет с контактами, выполненными или по плоской нижней стороне или по одному из краев. Пакет ЦП связан с системной платой через разъем формы гнездо (Socket) и слот (Slot).

Характеристики интерфейсов процессоров

Наименование	Разъем (число контактов)	Описание
Socket 1	169	Устанавливался на системных платах 486, напряжение 5 В, поддерживает 486 чипсет
Socket 2	238	Минимальное обновление Socket 1, поддерживает те же кремные элементы
Socket 3	237	Рабочее напряжение 5 В, но может работать также и при 3 В, переключатели размещены на системной плате
Socket 4	273	Первый разъем, спроектированный для Pentium. рабочее напряжение 5 В
Socket 5	320	Работает при 3,3 В поддерживает Pentium от 75 до 133 ГГц

Soket 6	235	Разработан для процессора 486. напряжение 3,3 В
Soket 7	321	Разработан для Pentium-ММХ, используется для всех Pentium с частотой шины 66 МГц
Soket 8	387	Используется только для Pentium Pro, из-за дороговизны быстро вышел из употребления
Slot 1	242	Поддерживает кэш-память L1 до 512 Кбайт, состоящая из двух по 256 Кбайт; оперирует на половинной частоте МП. Использовался для Pentium II, Pentium III и Celeron
Slot 2	330	Аналогичен разъему Slot 1, но поддерживает до 2 Мбайт кэш-памяти, работающей на частоте МП. Использовался для Pentium II, Pentium III, Pentium Xeon
Slot A	242	Разработан для AMD Athlon, механически совместим с разъемом Slot 1, но поддерживает совершенно другие электрические цепи
Soket 370	370	Заменяет Slot 1 для МП Celeron. Также используется для Pentium III
Soket A	462	Разработан для процессора AMD Athlon, который держал на кристалле кэш-память L2
Soket 423	423	Введен для удовлетворения новых требований Pentium 4, который содержит совершенно новую системную шину (FSB). Включает теплорассеиватель
Soket 603	603	Предназначен для Pentium 4. дополнительные контакты ориентированы на МП, которые будут содержать на кристалле кэш-память, а также для подключения других процессоров в мультипроцессорных системах
Soket 478	478	Разработан для поддержки технологий 0,13-мкм для Pentium 4 Northwood.

Задания на лабораторную работу:

Задание 1.

Выберите процессор, подходящий для установки на целевой системной плате.

Установите процессор на целевую системную плату

Задание 2.

Идентифицируйте процессор целевого компьютера. Назовите его основные характеристики. Дайте рекомендации по модернизации целевого компьютера.

Контрольные вопросы:

1. Перечислите функции процессора
2. Прокомментируйте основные параметры процессора
3. Какие современные типы процессоров вы знаете?

Лабораторная работа № 17

Тема: «Проектирование геоинформационной системы»

Цель: рассмотреть основные понятия геоинформационной системы, изучить функциональные возможности геоинформационной системы, указать области применения геоинформационной системы.

Контрольные вопросы:

1. Дайте определение ГИС.
2. Перечислите функциональные возможности ГИС.
3. Перечислите области применения ГИС.
4. Перечислите источники данных для наполнения ГИС.
5. Что представляет собой пространственный объект, основные типы графических объектов.
6. Назовите типы данных, используемых для описания пространственных данных, средствами ГИС.
7. Сущность векторных моделей представления данных?
8. Чем отличаются топологические и нетопологические векторные модели.
9. Сущность растровых моделей представления данных?
10. Перечислите основные компоненты ГИС и дайте их краткую характеристику.
11. Техническое обеспечение ГИС (перечислите компоненты и их назначение).
12. Программное обеспечение ГИС.

Теоретический материал.

Понятие о геоинформационных системах

Географическая информационная система или геоинформационная система (ГИС) - это информационная система, обеспечивающая сбор, хранение, обработку, анализ и отображение пространственных данных и связанных с ними непространственных, а также получение на их основе информации и знаний о географическом пространстве.

Считается, что географические или пространственные данные составляют более половины объема всей циркулирующей информации, используемой организациями, занимающимися разными видами деятельности, в которых необходим учет пространственного размещения объектов. ГИС ориентирована на обеспечение возможности принятия оптимальных управленческих решений на основе анализа пространственных данных.

Ключевыми словами в определении ГИС являются - анализ пространственных данных или пространственный анализ. ГИС может ответить на следующие вопросы:

Что находится в заданной области?

Где находится область, удовлетворяющая заданному набору условий?

Современные ГИС расширили использование карт за счет хранения графических данных в виде отдельных тематических слоев, а качественных и количественных характеристик составляющих их объектов в виде баз данных.

«Данные», «информация», «знания» в геоинформационных системах

Конкретизируя термины "данные", "информация", "знания", применительно к оперированию ими в информационной системе, можно отметить, что, имея много общего, эти понятия различаются по своей сути.

Под данными понимается совокупность фактов, известных об объектах, либо результаты измерения этих объектов. Данные, используемые в ГИС, отличаются высокой степенью формализации. Данные - это как бы строительный элемент в процессе создания информации, поскольку она получается в процессе обработки данных.

Применительно к ГИС под информацией понимается совокупность сведений, определяющих меру наших знаний об объекте.

В таком контексте знания можно рассматривать как результат интерпретации информации. Наиболее общее определение: знание – результат познания действительности, получивший подтверждение в практике. Научное знание отличается своей систематичностью, обоснованностью и высокой степенью структуризации.

Информационные системы можно рассматривать как эффективный инструмент получения знаний.

Различия между терминами «данные», «информация» и «знания» прослеживаются в истории развития технических систем, так вначале появились банки данных, позднее информационные системы, затем появились системы, основанные на знаниях - интеллектуальные системы (экспертные системы).

В настоящее время на рынке программных продуктов представлено несколько видов систем, работающих с пространственно распределенной информацией, к ним в частности, относятся системы автоматизированного проектирования, автоматизированного картографирования и ГИС. ГИС по сравнению с другими автоматизированными системами обладают развитыми средствами анализа пространственных данных.

Принципиально новые аналитические возможности. Обобщенные функции ГИС-систем.

Большинство современных ГИС осуществляют комплексную обработку информации.

Обобщенные функции ГИС-систем:

1. Ввод и редактирование данных;
2. Поддержка моделей пространственных данных;
3. Хранение информации;
4. Преобразование систем координат и трансформация картографических

проекций;

5. Растрово-векторные операции;
6. Измерительные операции;
7. Полигональные операции;
8. Операции пространственного анализа;
9. Различные виды пространственного моделирования;
10. Цифровое моделирование рельефа и анализ поверхностей;
11. Вывод результатов в разных формах.

Классификация ГИС

ГИС системы разрабатываются с целью решения научных и прикладных задач по мониторингу экологических ситуаций, рациональному использованию природных ресурсов, а также для инфраструктурного проектирования, городского и регионального планирования, для принятия оперативных мер в условиях чрезвычайных ситуаций др.

Множество задач, возникающих в жизни, привело к созданию различных ГИС, которые могут классифицироваться по следующим признакам:

По функциональным возможностям:

- полнофункциональные ГИС общего назначения;
- специализированные ГИС ориентированы на решение конкретной задачи в какой либо предметной области;
- информационно-справочные системы для домашнего и информационно-справочного пользования.

Функциональные возможности ГИС определяются также архитектурным принципом их построения:

- закрытые системы - не имеют возможностей расширения, они способны выполнять только тот набор функций, который однозначно определен на момент покупки.
- открытые системы отличаются легкостью приспособления, возможностями расширения, так как могут быть достроены самим пользователем при помощи специального аппарата (встроенных языков программирования).

По пространственному (территориальному) охвату:

- глобальные (планетарные);
- общенациональные;
- региональные;
- локальные (в том числе муниципальные).

По проблемно-тематической ориентации:

- общегеографические;
- экологические и природопользовательские;
- отраслевые (водных ресурсов, лесопользования, геологические, туризма и т.д.);

По способу организации географических данных:

- векторные;
- растровые;
- векторно-растровые ГИС.

Источники данных и их типы

В качестве источников данных для формирования ГИС служат:

- картографические материалы (топографические и общегеографические карты, карты административно-территориального деления, кадастровые планы и др.). Сведения, получаемые с карт, имеют территориальную привязку, поэтому их удобно использовать в качестве базового слоя ГИС. Если нет цифровых карт на исследуемую территорию, тогда графические оригиналы карт преобразуются в цифровой вид.

- данные дистанционного зондирования (ДДЗ) все шире используются для формирования баз данных ГИС. К ДДЗ, прежде всего, относят материалы, получаемые с космических носителей. Для дистанционного зондирования применяют разнообразные технологии получения изображений и передачи их на Землю, носители съемочной аппаратуры (космические аппараты и спутники) размещают на разных орбитах, оснащают разной аппаратурой. Благодаря этому получают снимки, отличающиеся разным уровнем обзорности и детальности отображения объектов природной среды в разных диапазонах спектра (видимый и ближний инфракрасный, тепловой инфракрасный и радиодиапазон). Все это обуславливает широкий спектр экологических задач, решаемых с применением ДДЗ.

К методам дистанционного зондирования относятся и аэро- и наземные съемки, и другие неконтактные методы, например гидроакустические съемки рельефа морского дна. Материалы таких съемок обеспечивают получение как количественной, так и качественной информации о различных объектах природной среды.

- результаты полевых обследований территорий, включают геодезические измерения природных объектов, выполняемые нивелирами, теодолитами, электронными тахеометрами, GPS приемниками, а также результаты обследования территорий с применением геоботанических и других методов, например, исследования по перемещению животных, анализ почв и др.

- статистические данные содержат данные государственных статистических служб по самым разным отраслям народного хозяйства, а также данные стационарных измерительных постов наблюдений (гидрологические и метеорологические данные, сведения о загрязнении окружающей среды и т. д)).

- литературные данные (справочные издания, книги, монографии и статьи, содержащие разнообразные сведения по отдельным типам географических объектов).

В ГИС редко используется только один вид данных, чаще всего это сочетание разнообразных данных на какую-либо территорию.

ОСНОВНЫЕ КОМПОНЕНТЫ ГИС. Аппаратные (технические) средства

К основным компонентам ГИС относят: технические (аппаратные) и программные средства, информационное обеспечение.

Требования к компонентам ГИС определяются, в первую очередь, пользователем, перед которым стоит конкретная задача (учет природных ресурсов, либо управление инфраструктурой города), которая должна быть решена для определенной территории, отличающейся природными условиями и степенью ее освоения.

Технические средства – это комплекс аппаратных средств, применяемых при функционировании ГИС: рабочая станция или персональный компьютер (ПК), устройства ввода-вывода информации, устройства обработки и хранения данных, средства телекоммуникации.

Рабочая станция или ПК являются ядром любой информационной системы и предназначены для управления работой ГИС и выполнения процессов обработки данных, основанных на вычислительных или логических операциях. Современные ГИС способны оперативно обрабатывать огромные массивы информации и визуализировать результаты.

Ввод данных реализуется с помощью разных технических средств и методов: непосредственно с клавиатуры, с помощью дигитайзера или сканера, через внешние компьютерные системы. Устройства для обработки и хранения данных сконцентрированы в системном блоке, включающем в себя центральный процессор, оперативную память, внешние запоминающие устройства и пользовательский интерфейс.

Внешние запоминающие устройства подключаются к компьютеру, в качестве таких устройств используются: дискеты (1.44 Мбайт), ZIP- диски (100 Мбайт), Магнитные жесткие диски (свыше 30 Гбайт). Для архивации данных служат оптические и магнитные диски CD-ROM и DVD-ROM с емкостью от 650 Мбайт до 9.0 Гбайт.

Устройства вывода данных должны обеспечивать наглядное представление результатов, прежде всего на мониторе, а также в виде графических оригиналов, получаемых на принтере или плоттере (графопостроителе), кроме того, обязательна реализация экспорта данных во внешние системы.

Программное обеспечение

Программные средства – совокупность программных средств, реализующих функциональные возможности ГИС, и программных документов, необходимых при их эксплуатации.

Структурно программное обеспечение ГИС включает базовые и прикладные программные средства.

Базовые программные средства включают: операционные системы (ОС), программные среды, сетевое программное обеспечение и системы управления базами данных. Операционные системы предназначены для управления ресурсами ЭВМ и процессами, использующими эти ресурсы. На настоящее время основные ОС: Windows и Unix.

Любая ГИС работает с данными двух типов данных - пространственными и атрибутивными, следовательно, программное обеспечение должно включить систему управления базами тех и других данных (СУБД), а также модули

управления средствами ввода и вывода данных, систему визуализации данных и модули для выполнения пространственного анализа.

Прикладные программные средства предназначены для решения для специализированных задач в конкретной предметной области и реализуются в виде отдельных модулей (приложений) и утилит (вспомогательных средств).

Информационное обеспечение

Информационное обеспечение - совокупность массивов информации, систем кодирования и классификации информации. Информационное обеспечение составляют реализованные решения по видам, объемам, размещению и формам организации информации, включая поиск и оценку источников данных, набор методов ввода данных, проектирование баз данных, их ведение и метасопровождение. Особенность хранения пространственных данных в ГИС – их разделение на слои. Многослойная организация электронной карты, при наличии гибкого механизма управления слоями, позволяет объединить и отобразить гораздо большее количество информации, чем на обычной карте. Данные о пространственном положении (географические данные) и связанные с ними табличные могут подготавливаться самим пользователем либо приобретаться. Для такого обмена данными важна инфраструктура пространственных данных.

Инфраструктура пространственных данных определяется нормативно-правовыми документами, механизмами организации и интеграции пространственных данных, а также их доступность разным пользователям. Инфраструктура пространственных данных включает три необходимых компонента: базовую пространственную информацию, стандартизацию пространственных данных, базы метаданных и механизм обмена данными.

СТРУКТУРЫ И МОДЕЛИ ДАННЫХ. Отображение объектов реального мира в ГИС.

Объекты реального мира, рассматриваемые в геоинформатике, отличаются пространственными, временными и тематическими характеристиками.

Пространственные характеристики определяют положение объекта в заранее определенной системе координат, основное требование к таким данным – точность.

Временные характеристики фиксируют время исследования объекта и важны для оценки изменений свойств объекта с течением времени. Основное требование к таким данным – актуальность, что означает возможность их использования для обработки, неактуальные данные – это устаревшие данные.

Тематические характеристики описывают разные свойства объекта, включая экономические, статистические, технические и другие свойства, основное требование – полнота.

Для представления пространственных объектов в ГИС используют пространственные и атрибутивные типы данных.

Пространственные данные – сведения, которые характеризуют

местоположение объектов в пространстве относительно друг друга и их геометрию.

Пространственные объекты представляют с помощью следующих графических объектов: точки, линии, области и поверхности.

Описание объектов осуществляется путем указания координат объектов и составляющих их частей.

Точечные объекты – это такие объекты, каждый из которых расположен только в одной точке пространства, представленной парой координат X, Y . В зависимости от масштаба картографирования, в качестве таких объектов могут рассматриваться дерево, дом или город.

Линейные объекты, представлены как одномерные, имеющие одну размерность – длину, ширина объекта не выражается в данном масштабе или не существенна. Примеры таких объектов: реки, границы муниципальных округов, горизонтали рельефа.

Области (полигоны) – площадные объекты представляются набором пар координат (X, Y) или набором объектов типа линия, представляющих собой замкнутый контур. Такими объектами могут быть представлены территории, занимаемые определенным ландшафтом, городом или целым континентом.

Поверхность - при ее описании требуется добавление к площадным объектам значений высоты. Восстановление поверхностей осуществляется с помощью использования математических алгоритмов (интерполяции и аппроксимации) по исходному набору координат X, Y, Z .

Дополнительные непространственные данные об объектах образуют набор атрибутов.

Атрибутивные данные - это качественные или количественные характеристики пространственных объектов, выражающиеся, как правило, в текстово-цифровом виде.

Примеры таких данных: географическое название, видовой состав растительности, характеристики почв и т.п.

Природа пространственных и атрибутивных данных различна, соответственно различны и методы манипулирования (хранения, ввода, редактирования, поиска и анализа) для двух этих составляющих геоинформационной системы. Одна из основных идей, воплощенных в традиционных ГИС - это сохранение связи между пространственными и атрибутивными данными, при раздельном их хранении и, частично, раздельной обработке.

Общее цифровое описание пространственного объекта включает: наименование; указание местоположения; набор свойств; отношения с другими объектами. Наименованием объекта служит его географическое название (если оно есть), его условный код или идентификатор, присваиваемый пользователем или системой.

Однотипные объекты по пространственному и тематическому признакам объединяются в слои цифровой карты, которые рассматриваются как отдельные информационные единицы, при этом существует возможность совмещения всей имеющейся информации

Структуры данных

Для представления пространственных данных в ГИС применяют векторные и растровые структуры данных.

Векторная структура – это представление пространственных объектов в виде набора координатных пар (векторов), описывающих геометрию объектов (рис.1).

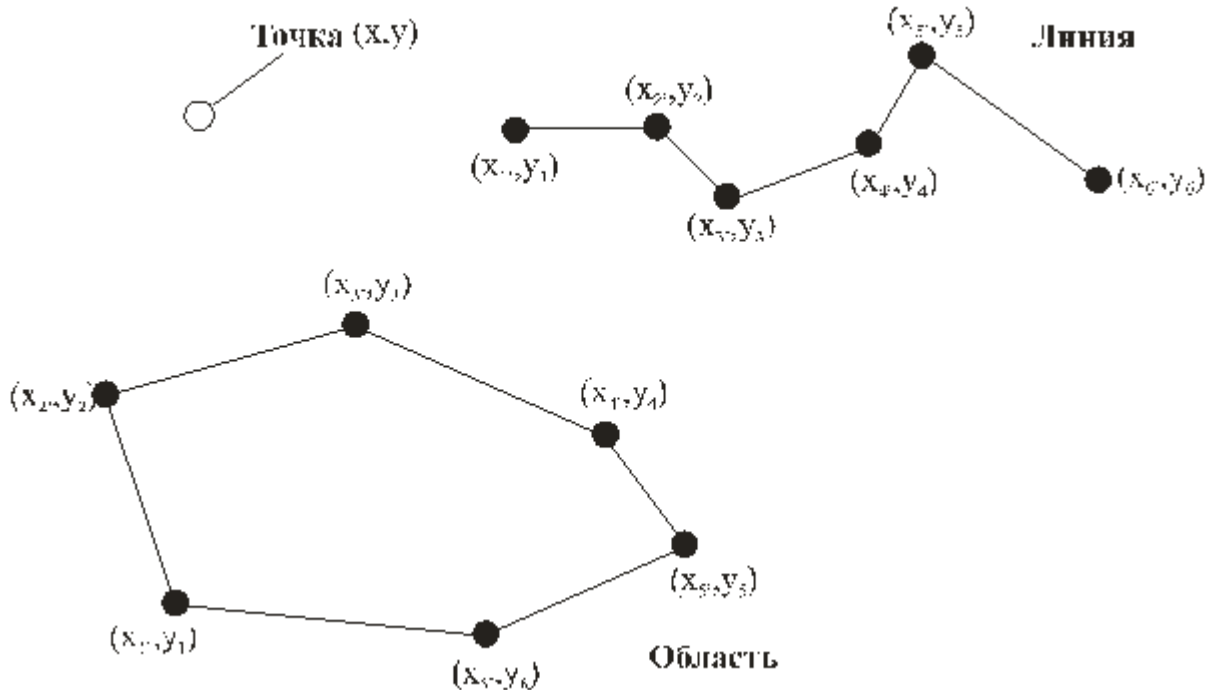


Рис. 1. Векторное представление пространственных данных

Растровая структура данных предполагает представления данных в виде двумерной сетки, каждая ячейка которой содержит только одно значение, характеризующее объект, соответствующий ячейке раstra на местности или на изображении. В качестве такой характеристики может быть код объекта (лес, луг и т.д.) высота или оптическая плотность.

Точность растровых данных ограничивается размером ячейки. Такие структуры являются удобным средством анализа и визуализации разного рода информации.

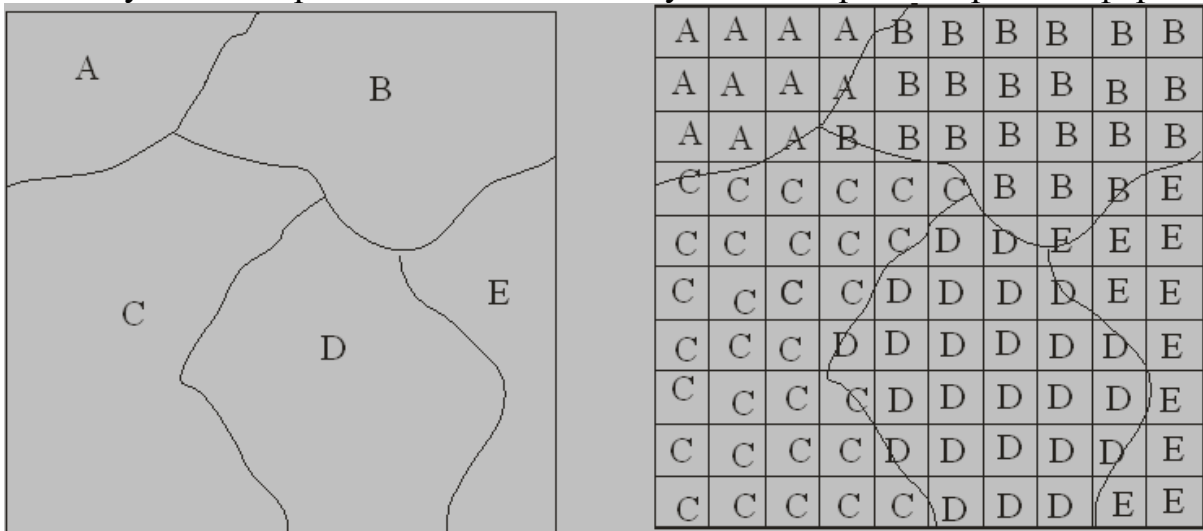


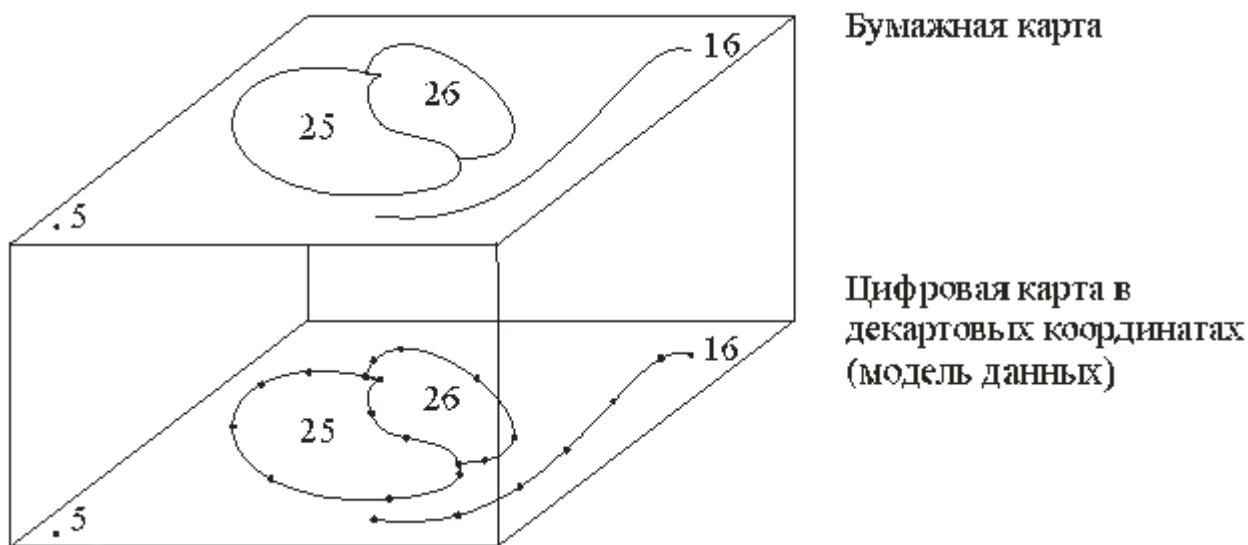
Рис. 2. Растровая структура данных

Для реализации растровых и векторных структур разработаны различные модели данных.

Модели данных

Модели пространственных данных – логические правила для формализованного цифрового описания пространственных объектов.

Векторные модели данных. Существует несколько способов объединения векторных структур данных в векторную модель данных, позволяющую исследовать взаимосвязи между объектами одного слоя или между объектами разных слоев. Простейшей векторной моделью данных является «спагетти»- модель (рис.3). В этом случае переводится «один в один» графическое изображение карты.



Бумажная карта

Цифровая карта в декартовых координатах (модель данных)

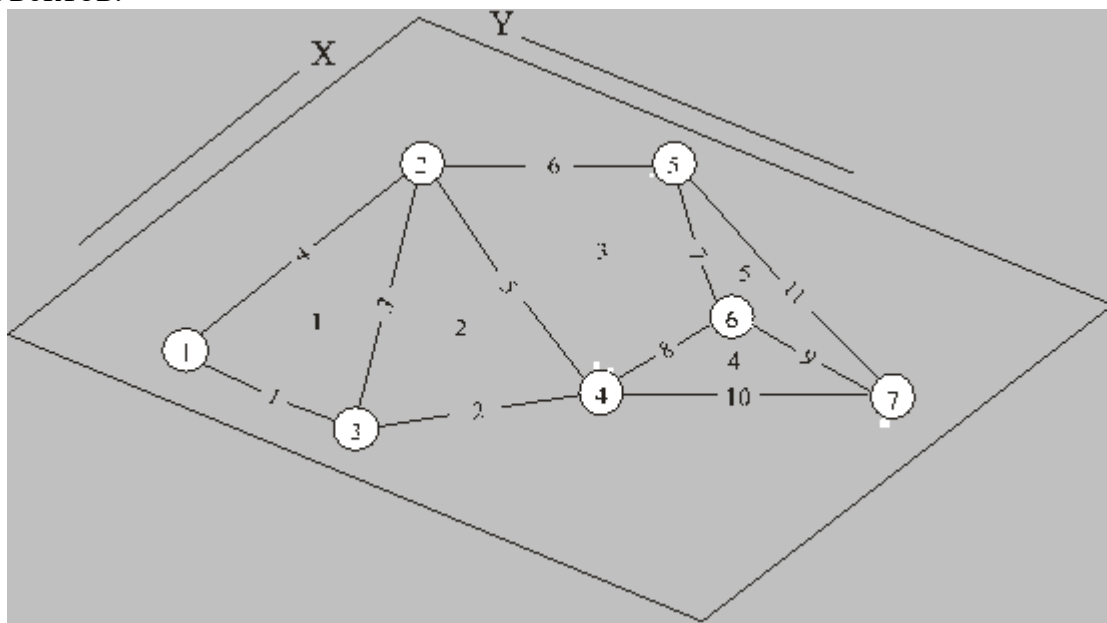
Объект	номер	Положение
Точка	5	Одна пара координат (x,y)
Линия	16	Набор пар координат (x,y)
Область	25	Набор пар координат (x,y), первая и последняя совпадают

Рис. 3. «Спагетти»-модель

В этой модели не содержится описания отношений между объектами, каждый геометрический объект хранится отдельно и не связан с другими, например общая граница объектов 25 и 26 записывается дважды, хотя с помощью одинакового набора координат. Все отношения между объектами должны вычисляться

независимо, что затрудняет анализ данных и увеличивает объем хранимой информации.

Векторные топологические модели (рис. 4) содержат сведения о соседстве, близости объектов и другие, характеристики взаимного расположения векторных объектов.



Файл узлов		
Номер узла	Координат X	Координат Y
1	19	6
2	15	15
3	27	13
4	24	19

Файл областей	
Номера областей	Список дуг
1	1, 4, 3
2	2, 3, 5
3	5, 6, 7, 8

Файл дуг				
Но мер дуг	Прав ый полиг	Левый полиг	Начальн ый узел	Конечн ый узел
1	1	2	1	2
2	2	3	2	3
3	2	1	2	3
4	1	2	3	4
5	2	3	5	6
6	3	4	6	7
7	3	4	7	8
8	3	4	8	7
9	3	4	6	7
10	4	3	4	7
11	3	4	5	6

	I	I		
1	1	0	3	1
2	2	0	4	3
3	2	1	3	2
4	1	0	1	2
5	3	2	4	2
6	3	0	2	5

Рис. 4. Векторная топологическая модель данных

Топологическая информация описывается набором узлов и дуг. Узел - это пересечение двух или более дуг, и его номер используется для ссылки на любую дугу, которой он принадлежит. Каждая дуга начинается и заканчивается либо в точке пересечения с другой дугой, либо в узле, не принадлежащем другим дугам. Дуги образуются последовательностью отрезков, соединённых промежуточными точками. В этом случае каждая линия имеет два набора чисел: пары координат промежуточных точек и номера узлов. Кроме того, каждая дуга имеет свой идентификационный номер, который используется для указания того, какие узлы представляют её начало и конец.

Разработаны и другие модификации векторных моделей, в частности, существуют специальные векторные модели для представления моделей поверхностей, которые будут рассмотрены далее.

Растровые модели используются в двух случаях. В первом случае – для хранения исходных изображений местности. Во втором случае, для хранения тематических слоев, когда пользователей интересуют не отдельные пространственные объекты, а набор точек пространства, имеющих различные характеристики (высотные отметки или глубины, влажность почв и т.д.), для оперативного анализа или визуализации.

Существует несколько способов хранения и адресации значений отдельных ячеек растра, и их атрибутов, названий слоев и легенд.

При использовании растровых моделей актуальным является вопрос сжатия растровых данных, для которого разработаны методы группового кодирования, блочного кодирования, цепочного кодирования и представления в виде квадродерева.

Форматы данных.

Форматы данных определяют способ хранения информации на жестком диске, а также механизм ее обработки. Модели данных и форматы данных определенным способом взаимосвязаны.

Существует большое количество форматов данных. Можно отметить, что во многих ГИС поддерживаются основные форматы хранения растровых данных (TIFF, JPEG, GIF, BMP, WMF, PCX), а также GeoSpot, GeoTIFF, позволяющие

передавать информацию о привязке растрового изображения к реальным географическим координатам, и MrSID - для сжатия информации. Наиболее распространенным среди векторных форматов является - DXF.

Все системы поддерживают обмен пространственной информацией (экспорт и импорт) со многими ГИС и САПР через основные обменные форматы: SHP, E00, GEN (ESRI), VEC (IDRISI), MIF (MapInfo Corp.), DWG, DXF (Autodesk), WMF (Microsoft), DGN (Bentley). Только некоторые, в основном отечественные системы, поддерживают российские обменные форматы – F1M (Роскартография), SXF (Военно-топографическая служба).

Довольно часто для эффективной реализации одних компьютерных операций предпочитают векторный формат, а для других растровый. Поэтому, в некоторых системах реализуются возможности манипулирования данными в том и в другом формате, и функции преобразования векторного в растровый, и наоборот, растрового в векторный форматы.

Базы данных и управление ими

Совокупность цифровых данных о пространственных объектах образует множество пространственных данных и составляет содержание баз данных.

База данных (БД) – совокупность данных организованных по определенным правилам, устанавливающим общие принципы описания, хранения и манипулирования данными.

Создание БД и обращение к ней (по запросам) осуществляется с помощью системы управления базами данных (СУБД).

Логическая структура элементов базы данных определяется выбранной моделью БД. Наиболее распространенными моделями БД являются иерархические, сетевые и реляционные и объектно-ориентированные.

Иерархические модели представляют древовидную структуру, в этом случае каждая запись связана только с одной записью, находящейся на более высоком уровне.

Такая система хорошо иллюстрируется системой классификации растений и животных. Примером может также служить структура хранения информации на дисках ПК. Главное понятие такой модели уровень. Количество уровней и их состав зависит от принятой при создании БД классификации. Доступ к любой из этих записей осуществляется путем прохода по строго определенной цепочке узлов. При такой структуре легко осуществлять поиск нужных данных, но если изначально описание неполное, или не предусмотрен какой либо критерий поиска, то он становится невозможным. Для достаточно простых задач такая система эффективна, но она практически непригодна для использования в сложных системах с оперативной обработкой запросов.

Сетевые модели были призваны устранить некоторые из недостатков иерархических моделей. В сетевой модели каждая запись в каждом узле сети может быть связана с несколькими другими узлами. Записи, входящие в состав сетевой структуры, содержат в себе указатели, определяющие местоположение других

записей, связанных с ними. Такая модель позволяет ускорить доступ к данным, но изменение структуры базы требует значительных усилий и времени.

Реляционные модели собирают данные в унифицированные таблицы. Таблице присваивается уникальное имя внутри БД. Каждый столбец - это поле, имеющее имя, соответствующее содержащемуся в нем атрибуту. Каждая строка в таблице соответствует записи в файле. Одно и тоже поле может присутствовать в нескольких таблицах. Так как строки в таблице не упорядочены, то определяется один или несколько столбцов, значения которых однозначно идентифицируют каждую строку. Такой столбец называется первичным ключом. Взаимосвязь таблиц поддерживается внешними ключами. Манипулирование данными осуществляется при помощи операций, порождающих таблицы. Пользователь может легко заносить в базу новые данные, комбинировать таблицы, выбирая отдельные поля и записи, и формировать новые таблицы для отображения на экране.

Объектно-ориентированные модели применяют, если геометрия определенного объекта способна охватывать несколько слоев, атрибуты таких объектов могут наследоваться, для их обработки применяют специфические методы.

Для обработки данных, размещенных в таблицах необходимы дополнительные сведения о данных, их называют метаданными.

Метаданные - данные о данных: каталоги, справочники, реестры и иные формы описания наборов цифровых данных.

ТЕХНОЛОГИИ ВВОДА ДАННЫХ. Способы ввода данных.

В соответствии с используемыми техническими средствами различают два способа ввода данных: дигитализацию и векторизацию. Для ручного ввода пространственных данных применяется дигитайзер. Он состоит из планшета (столика) с электронной сеткой, к которому присоединено устройство называемое курсором. Курсор представляет собой подобие графического манипулятора – мыши, имеет визир, нанесенный на прозрачную пластинку, с помощью которого оператор выполняет точное наведение на отдельные элементы карты. На курсоре помещены кнопки, которые позволяют фиксировать начало и конец линии или границы области, число кнопок зависит от уровня сложности дигитайзера. Дигитайзеры бывают разных форматов и обеспечивают разрешение 0,03 мм с общей точностью 0,08 мм на расстоянии 1,5 м. Существуют автоматизированные дигитайзеры, обеспечивающие автоматическое отслеживание линий.

Наибольшее распространение для ввода данных получили сканеры. Они позволяют вводить растровое изображение карты в компьютер. Существуют различные типы сканеров, которые различаются: по способу подачи исходного материала (планшетные и протяжные (барабанного типа)); по способу считывания информации (работающие на просвет или на отражение); по радиометрическому разрешению или глубине цвета; по оптическому (или геометрическому) разрешению. Последняя характеристика определяется минимальным размером элемента изображения, который различается сканером.

Процесс цифрования растрового изображения на экране компьютера называют векторизацией. Существует три способа векторизации: ручной, интерактивный и

автоматический. При ручной векторизации оператор обводит мышью на изображении каждый объект, при интерактивной - часть операций производится автоматически. Так, например, при векторизации горизонталей достаточно задать начальную точку и направление отслеживания линий, далее векторизатор сам отследит эту линию до тех пор, пока на его пути не встретятся неопределенные ситуации, типа разрыва линии. Возможности интерактивной векторизации прямо связаны с качеством исходного материала и сложностью карты. Автоматическая векторизация предполагает непосредственный перевод из растрового формата в векторный с помощью специальных программ, с последующим редактированием. Оно необходимо, поскольку даже самая изощренная программа может неверно распознать объект, принять например, символ за группу точек, и т.п.

Преобразование исходных данных

Отсканированные исходные карты создавались в определенной картографической проекции и системе координат. При оцифровке эта сложная проекция сводится в набор пространственных координат. Поэтому необходимо преобразовать карту к ее исходной проекции. Для этого в ГИС вводятся сведения об используемой проекции (обычно ГИС позволяет работать с большим числом проекций) и осуществляется ряд преобразований. Три основных из них, которые часто выполняются одновременно, это перенос, поворот и масштабирование.

Перенос – это просто перемещение всего графического объекта в другое место на координатной плоскости. Он выполняется добавлением определенных величин к координатам X и Y объекта:

$$X' = X + T_x, Y' = Y + T_y$$

Масштабирование тоже очень полезно, так как часто сканируются карты разных масштабов, для этого используют соотношение:

$$X' = X \cdot S_x, Y' = Y \cdot S_y$$

Поворот выполняется с использованием тригонометрических функций:

$$X' = X \cos \theta + Y \sin \theta, Y' = X \sin \theta + Y \cos \theta$$

Все необходимые преобразования могут быть выполнены и использованием этих трех основных графических операций по координатам опорных точек.

Ввод данных дистанционного зондирования

В ГИС используют не первичные материалы ДЗ, получаемые во время съемки, а производные, формируемые в результате их обработки. Данные со спутников подвергаются предварительной цифровой обработке для устранения радиометрических и геометрических искажений, влияния атмосферы и т.д. Для улучшения визуального качества исходных изображений могут применяться процедуры для изменения яркости и контрастности, фильтрации для устранения

шумов или подчеркивания контуров и мелких деталей. При использовании аэрофотоснимков следует обращать внимание на искажения, вызываемые углами наклонов снимков и рельефом местности, которые могут быть устранены в процессе трансформирования или ортофототрансформирования.

АНАЛИЗ ПРОСТРАНСТВЕННЫХ ДАННЫХ. Задачи пространственного анализа.

К средствам пространственного анализа относятся различные процедуры манипулирования пространственными и атрибутивными данными, выполняемые при обработке запросов пользователя. К ним относятся, например, операции наложения графических объектов, средства анализа сетевых структур или выделения объектов по заданным признакам.

Для каждого ГИС-пакета характерен свой набор средств пространственного анализа, обеспечивающий решение специфических задач пользователя, в тоже время можно выделить ряд основных функций, свойственных практически каждому ГИС-пакету. Это, прежде всего, организация выбора и объединения объектов в соответствии с заданными условиями, реализация операций вычислительной геометрии, анализ наложений, построение буферных зон, сетевой анализ.

Основные функции пространственного анализа данных

Выбор объектов по запросу: самой простой формой запроса является получение характеристик объекта указанного курсором на экране и обратная операция, когда изображаются объекты с заданными атрибутами. Более сложные запросы позволяют выбирать объекты по нескольким признакам, например по признаку удаленности одних объектов от других, совпадающие объекты, но расположенные в разных слоях и т. д.

Для выбора данных в соответствии с определенными условиями используются SQL-запросы. Для выполнения запросов разной сложности реализованы возможности использования при составлении запросов математических и статистических функций, а также географических операторов, позволяющих выбирать объекты на основании их взаимного расположения в пространстве (например, находится ли анализируемый объект внутри другого объекта или пересекается с ним).

Обобщение данных может проводиться по равенству значений определенного атрибута, в частности для зонирования территории. Еще один способ группировки – объединение объектов одного тематического слоя в соответствии с их размещением внутри полигональных объектов других тематических слоев.

Геометрические функции: к ним относят расчеты геометрических характеристик объектов или их взаимного положения в пространстве, при этом используются формулы аналитической геометрии на плоскости и в пространстве. Так для площадных объектов вычисляются занимаемые ими площади или периметры границ, для линейных – длины, а также расстояния между объектами и т.д.

Оверлейные операции (топологическое наложение слоев) являются одними из

самых распространенных и эффективных средств. В результате наложения двух тематических слоев образуется другой дополнительный слой в виде графической композиции исходных слоев. Учитывая, что анализируемые объекты могут относиться к разным типам (точка, линия, полигон), возможны разные формы анализа: точка на точку, точка на полигон и т.д. Наиболее часто анализируется совмещение полигонов.

Построение буферных зон. Одним из средств анализа близости объектов является построение буферных зон. Буферные зоны – это районы (полигоны), граница которых отстоит на заданном расстоянии от границы исходного объекта. Границы таких зон вычисляются на основе анализа соответствующих атрибутивных характеристик. При этом ширина буферной зоны может быть как постоянной, так и переменной. Например, буферная зона вокруг источника электромагнитного излучения, будет иметь форму круга, а зона загрязнения от дымовой трубы завода с учетом розы ветров будет иметь форму близкую к эллипсу.

Сетевой анализ позволяет пользователю проанализировать пространственные сети связных линейных объектов (дороги, линии электропередач и т. д.). Обычно сетевой анализ служит для задач определения ближайшего, наиболее выгодного пути, определения уровня нагрузки на сеть, определение адреса объекта или маршрута по заданному адресу и другие задачи.

Анализ пространственного распределения объектов

Анализ пространственного распределения объектов. Фактически во многих случаях необходимо знать не только объем пространства, занимаемый объектами, но и расположение объектов в пространстве, которое может характеризоваться количеством объектов в определенной области, например, распределение численности населения. Наиболее распространены методы анализа распределения точечных объектов. Мерой точечного распределения служит плотность. Она определяется как результат деления числа точек на значение площади территории, на которой они расположены. Кроме плотности распределения можно оценить форму распределения. Точечные распределения встречаются в одном из четырёх возможных вариантов: равномерном (если число точек в каждой малой подобласти такое же, как и в любой другой подобласти), регулярном (если точки, разделённые одинаковыми интервалами по всей области, расположены в узлах сетки), случайном, кластерном (если точки собраны в тесные группы).

Точечные распределения могут описываться не только количеством точек в пределах подобластей. Часто анализируются локальные отношения внутри пар точек. Вычисление этого статистического показателя включает определение среднего расстояния до ближайшей соседней точки среди всех возможных пар ближайших точек. Данный метод позволяет оценить меру разреженности точек в распределении.

Распределение линий также оценивается по плотности. Обычно вычисления выполняются для сравнения разных географических областей, например по густоте гидрографической сети. Линии могут также оцениваться по близости и возможным пересечениям. Другими важными характеристиками являются ориентация,

направленность и связанность.

Анализ распределения полигонов подобен анализу распределения точек, однако при оценке плотности определяют не количество полигонов на единицу площади, а относительную долю площади, занимаемой полигоном

МОДЕЛИРОВАНИЕ ПОВЕРХНОСТЕЙ. Поверхность и цифровая модель.

Основой для представления данных о земной поверхности являются цифровые модели рельефа.

Поверхности – это объекты, которые чаще всего представляются значениями высоты Z , распределенными по области, определенной координатами X и Y .

Цифровые модели рельефа (ЦМР) используют для компьютерного представления земных поверхностей.

ЦМР – средство цифрового представления рельефа земной поверхности.

Построение ЦМР требует определённой формы представления исходных данных (набора координат точек X, Y, Z) и способа их структурного описания, позволяющего восстанавливать поверхность путем интерполяции или аппроксимации исходных данных.

Источники данных для формирования ЦМР

Исходные данные для формирования ЦМР могут быть получены по картам – цифрованием горизонталей, по стереопарам снимков, а также в результате геодезических измерений или лазерного сканирования местности. Наиболее распространен первый способ, т.к. сбор по стереопарам снимков отличается трудоемкостью и требует специфического программного обеспечения, но в то же время позволяет обеспечить желаемую степень детальности представления земной поверхности. Лазерное сканирование перспективный современный метод, пока достаточно дорогой.

Интерполяции

Построение ЦМР требует определенной структуры данных, а исходные точки могут быть по разному распределены в пространстве. Сбор данных может осуществляться по точкам регулярной сетки, по структурным линиям рельефа или хаотично. Первичные данные с помощью тех или иных операций приводят к одному из наиболее распространенных в ГИС структур для представления поверхностей: GRID, TIN или TGRID.

TIN (Triangulated Irregular Network) – нерегулярная триангуляционная сеть, система неперекрывающихся треугольников. Вершинами треугольников являются исходные опорные точки. Рельеф в этом случае представляется многогранной поверхностью, каждая грань которой описывается либо линейной функцией (полиэдральная модель), либо полиномиальной поверхностью, коэффициенты которой определяются по значениям в вершинах граней треугольников. Для

получения модели поверхности нужно соединить пары точек ребрами определенным способом, называемым триангуляцией Делоне (рис. 5).

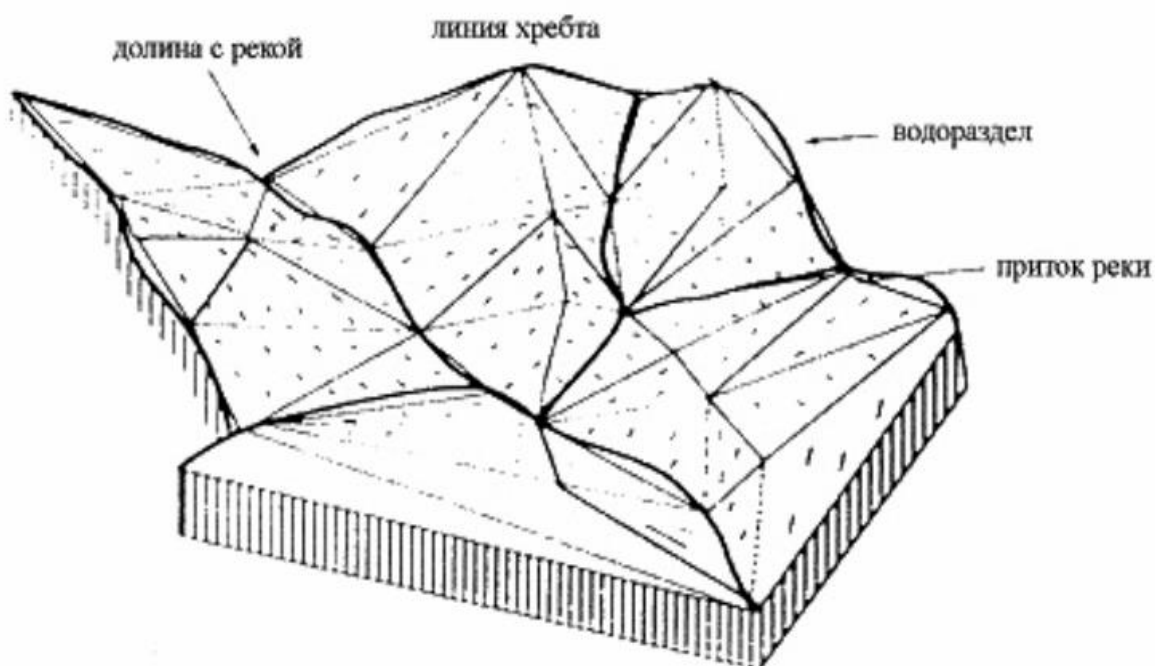


Рис. 5. TIN модель

Триангуляция Делоне в приложении к двумерному пространству формулируется следующим образом: система взаимосвязанных неперекрывающихся треугольников имеет наименьший периметр, если ни одна из вершин не попадает внутрь ни одной из окружностей, описанных вокруг образованных треугольников (рис.6).

Образовавшиеся треугольники при такой триангуляции максимально приближаются к равносторонним, а каждая из сторон образовавшихся треугольников из противоположной вершины видна под максимальным углом из всех возможных точек соответствующей полуплоскости. Интерполяция выполняется по образованным ребрам.

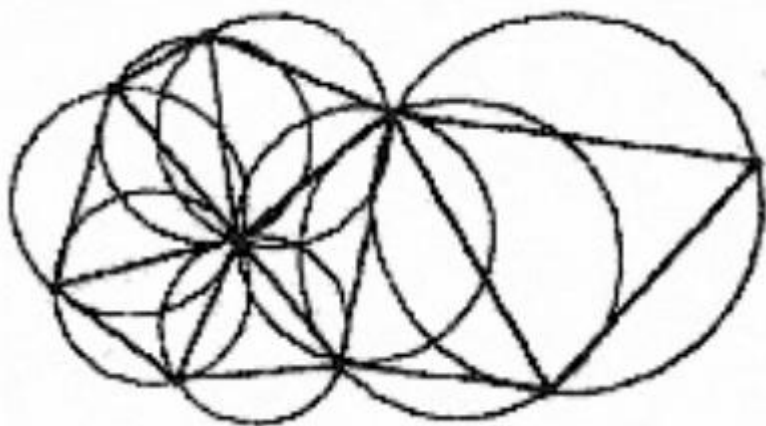


Рис. 6. Триангуляция Делоне

Отличительной особенностью и преимуществом триангуляционной модели является то, что в ней нет преобразований исходных данных. С одной стороны, это

не дает использовать такие модели для детального анализа, но с другой стороны, исследователь всегда знает, что в этой модели нет привнесенных ошибок, которыми грешат модели, полученные при использовании других методов интерполяции. Немаловажен и тот факт, что это самый быстрый метод интерполяции. Однако, если в ранних версиях большинства ГИС триангуляционный метод был основной, то сегодня большое распространение получили модели в виде регулярной матрицы значений высот.

GRID – модель, представляет собой регулярную матрицу значений высот, полученную при интерполяции исходных данных. Для каждой ячейки матрицы высота вычисляется на основе интерполяции. Фактически это сетка, размеры которой задаются в соответствии с требованиями точности конкретной решаемой задачи. Регулярная сетка соответствует земной поверхности, а не изображению.

При использовании GRID-модели существует некоторая сложность в выборе интервала между точками. Например, участки поверхности могут быть как сильно пересеченными, так и выположенными. В первом случае необходимо большее количество точек на единицу площади.

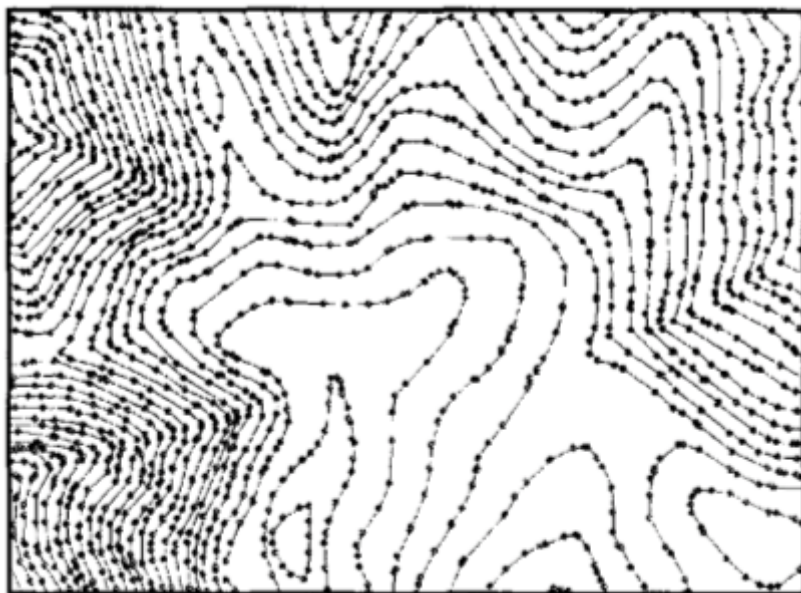


Рис. 7. Плотность точек в модели GRID

TGRID (triangulated grid) – модель, сочетающая в себе элементы моделей TIN и GRID. Такие модели имеют свои преимущества, например, позволяют использовать дополнительные данные для описания сложных форм рельефа (обрывы, скальные выступы).

Интерполяции

Восстановление поверхностей реализуется на основе интерполяции исходных данных.

Интерполяция – восстановление функции на заданном интервале по известным ее значениям конечного множества точек, принадлежащих этому интервалу.

В настоящее время известны десятки методов интерполяции

поверхностей, наиболее распространенные: линейная интерполяция; метод обратных взвешенных расстояний, кригинг; сплайн-интерполяция; тренд-интерполяция.

Кригинг. Метод интерполяции, который основан на использовании методов математической статистики. В его реализации применяется идея регионализованной переменной, т.е. переменной, которая изменяется от места к месту с некоторой видимой непрерывностью, поэтому не может моделироваться только одним математическим уравнением. Поверхность рассматривается в виде трех независимых величин. Первая - тренд, характеризует изменение поверхности в определенном направлении. Далее предполагается, что имеются небольшие отклонения от общей тенденции, вроде маленьких пиков и впадин, которые являются случайными, но все же связанными друг с другом пространственно.

Наконец, имеется случайный шум (например, валуны). С каждой из трех переменных надо оперировать в отдельности. Тренд оценивается с использованием математического уравнения, которое наиболее близко представляет общее изменение поверхности, во многом подобно поверхности тренда.

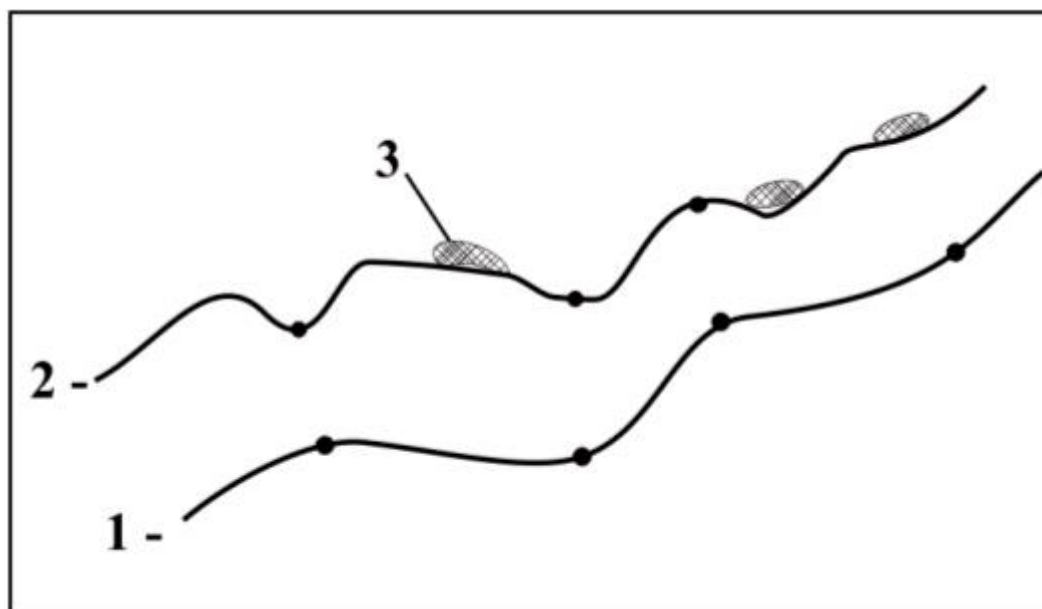


Рис. 8. Элементы кригинга: 1 - тренд, 2 - случайные, но пространственно связанные высотные колебания, 3 - случайный шум.

Ожидаемое изменение высоты измеряется по вариограмме, на которой по горизонтальной оси откладывается расстояние между отсчетами, а на вертикальной - полудисперсия. Полудисперсия определяется как половина дисперсии между значениями высоты исходных точек и высот соседних точек. Затем через точки данных проводится кривая наилучшего приближения. Дисперсия в какой-то момент достигает максимума и остается постоянной (выявляется предельный радиус корреляции).

Интерполяция методом кригинга в большинстве случаев дает хорошие результаты, даже когда плотность исходных точек не велика. Однако, при некотором расположении точек возможно появление резких пиков и впадин.

Метод обратных взвешенных расстояний. Этот метод основан на предположении,

что чем ближе друг к другу находятся исходные точки, тем ближе их значения. Для точного описания топографии набор точек, по которым будет осуществляться интерполяция, необходимо выбирать в некоторой окрестности определяемой точки, так как они оказывают наибольшее влияние на ее высоту. Это достигается следующим образом. Вводится максимальный радиус поиска или количество точек, ближайших по расстоянию от начальной (определяемой) точки. Затем значению высоты в каждой выбранной точке задается вес, вычисляемый в зависимости от квадрата расстояния до определяемой точки. Этим достигается, чтобы более близкие точки вносили больший вклад в определение интерполируемой высоты по сравнению с более удаленными точками.

Тренд интерполяция. В некоторых случаях исследователя интересуют общие тенденции поверхности, которые характеризуются поверхностью тренда.

Аналогично методу обратных взвешенных расстояний для поверхности тренда используется набор точек в пределах заданной окрестности. В пределах каждой окрестности строится поверхность наилучшего приближения на основе математических уравнений, таких как полиномы или сплайны.

Поверхности тренда могут быть плоскими, показывая общую тенденцию или более сложными. Тип используемого уравнения или степень полинома определяет величину волнистости поверхности. Например, поверхность тренда первого порядка будет выглядеть как плоскость, пересекающая под некоторым углом всю поверхность. Если поверхность имеет один изгиб, то такую поверхность называют поверхностью тренда второго порядка.

Сплайн интерполяция. Возможность описания сложных поверхностей с помощью полиномов невысоких степеней определяется тем, что при сплайн интерполяции вся территория разбивается на небольшие непересекающиеся участки. Аппроксимация полиномами осуществляется отдельно для каждого участка. Обычно используют полином третьей степени - кубический сплайн. Затем строится общая функция «склейки» на всю область, с заданием условия непрерывности на границах участков и непрерывности первых и вторых частных производных, т.е. обеспечивается гладкость склеивания полиномов.

Сглаживание сплайн-функциями особенно удобно при моделировании поверхностей, осложненных разрывными нарушениями, и позволяет избежать искажения типа «краевых эффектов».

КРАТКИЙ ОБЗОР ПРОГРАММНЫХ СРЕДСТВ, ИСПОЛЬЗУЕМЫХ В РОССИИ

В России используются ГИС, как профессионального уровня, так и специализированные. Программные продукты формируются на основе модульного принципа. Обычно выделяют базовый модуль и модули расширения (или приложения). В базовом модуле содержатся функции, реализующие основные операции ГИС, в том числе программная поддержка устройств ввода-вывода, экспорт и импорт данных и т.д. Следует отметить, что программные продукты разных фирм имеют много общего, так как производители вынуждены заимствовать друг у друга те или иные технологические разработки. В настоящее

время на рынке представлено около 20 хорошо известных ГИС-пакетов, которые можно отнести к полнофункциональным.

Характеризуя свойства полнофункциональных ГИС можно отметить их общие черты. Все системы работают на платформе Windows. Только некоторые из них имеют версии, работающие под управлением других операционных систем («Горизонт» - MS DOS, Unix, Linux, MC BC, Free BSD, Solaris, ИНТРОС; ПАРК – MS DOS; Arc GIS Arc Info-Solaris, Digital Unix, AIX и др.; ArcView GIS – Unix).

Все системы поддерживают обмен пространственной информацией (экспорт и импорт) со многими ГИС и САПР через основные обменные форматы.

Еще более однородными являются возможности по работе с атрибутивной информацией. Большинство систем обеспечивают работу со всеми основными СУБД через драйверы ODBC, BDE. Первой в ряду поддерживаемых или используемых СУБД стоит Oracle.

В преобладающем большинстве случаев современные полнофункциональные ГИС позволяют расширять свои возможности. Основным способом расширения возможностей является программирование на языках высокого уровня (MS Visual Basic, MS Visual C++, Borland Delphi, Borland C++ Builder) с подключением DLL и OCX-библиотек (ActiveX). Естественно имеются и исключения. Такие системы как MapInfo Professional используют Map Basic, а системы ArcView GIS - Avenue.

Наиболее распространенными зарубежными системами по разным причинам являются ArcView GIS, MapInfo Professional, MicroStation/J. Аналогичный перечень отечественных систем возглавляют ГеоГраф, Панорама (Карта 2000), ПАРК, GeoLink.

Коротко охарактеризуем наиболее распространенные программные продукты, отмечая особенности и области применения.

ArcGIS ArcInfo (разработчик фирма ESRI, США). Полнофункциональная ГИС, состоящая из двух независимо устанавливаемых программных пакетов – ArcInfo Workstation и ArcInfo Desktop. Первый состоит из трех базовых модулей: ArcMap – отображение, редактирование и анализ данных, ArcCatalog – доступ к данным и управление ими, ArcToolbox – инструмент расширенного пространственного анализа, управление проекциями и конвертацией данных. Дополнительные модули обеспечивают решение следующих задач:

- Arc COGO – набор средств и функций для работы с геодезическими данными;
- Arc GRID – имеет мощный набор средств анализа и управления непрерывно распределенными числовыми и качественными признаками, представляемыми в виде регулярных моделей, а также моделирования сложных процессов;
- ARC TIN – предназначен для моделирования топографических поверхностей;
- Arc NETWORK – для моделирования и анализа топологически связанных объектов в виде пространственных сетей, оценки и управление ресурсами, распределенными по сетям, и процессами в таких сетях.

ArcInfo обеспечивает создание геоинформационных систем, создание и ведение земельных, лесных, геологических и других кадастров, проектирование транспортных сетей, оценку природных ресурсов.

ArcGIS ArcView (разработчик фирма ESRI, США). Настольная ГИС, которая предоставляет пользователю средства выбора и просмотра разнообразных

геоданных, их редактирования, анализа и вывода (бизнес, наука, образование, управление, социология, демография, экология, транспорт, городское хозяйство).

Все продукты ArcGis могут использовать дополнительные модули для решения специализированных задач пространственного анализа:

- ArcGIS Spatial Analyst – программный модуль для работы с растровыми поверхностями. Позволяет анализировать характеристики поверхности, а также интерполировать пространственно распределенные данные для визуализации и анализа процессов;

- ArcGIS 3D Analyst – программа для создания, визуализации и анализа трехмерных объектов и поверхностей;

- ArcGIS Geostatistical Analyst – новый модуль для интерполяции поверхностей на основе статистического анализа пространственно распределенных данных;

- ArcView поддерживает реляционные СУБД, имеет развитую деловую графику (форма просмотра, табличная форма, форма диаграмм, создание макета), предусматривает создание профессионально оформленной картографической информации и разработку собственных приложений.

MapInfo Professional (разработка фирмы MapInfo Corp.США), одна из самых распространенных настольных ГИС в России. MapInfo специально спроектирован для обработки и анализа информации, имеющей адресную или пространственную привязку.

В MapInfo реализованы:

- поиск географических объектов;
- работа с базами данных;
- геометрические функции: расчеты площадей, длин, периметров, объемов, заключенных между поверхностями;
- построение буферных зон вокруг любого объекта или группы объектов;
- расширенный язык запросов SQL, запросы основываются на выражениях, осуществляют объединение, отображают доступные поля, позволяют делать подзапросы, объединения из нескольких таблиц и географические объединения.
- компьютерный дизайн и подготовку к изданию картографических документов.

ГеоГраф (разработка Центра информационных исследований Института географии РАН, Россия). Дает возможность создавать электронные тематические атласы и композиции карт на основе слоев цифровых карт и связанных с ними таблиц атрибутивных данных. Основные возможности ГеоГраф следующие:

- создание пространственных объектов в виде косметических слоев с привязкой к ним таблиц атрибутивных данных;
- подсистема управления атрибутивными данными, включая подсоединение таблиц, редактирование, выборку, сортировку, запросы по образцу и т.д.
- электронное тематическое картографирование и др.

Панорама (Россия) Построение и обработка цифровых и электронных карт, ведение картографической и атрибутивной баз данных.

Отдельно следует выделить профессиональные многофункциональные инструментальные ГИС, обеспечивающие возможность непосредственной обработки данных ДЗ. К ним относятся ERDAS IMAGINE, ERMapper и др.

ER Mapper (разработка ER Mapper) Обработка больших объёмов фотограмметрической информации, тематическое картографирование (геофизика, природные ресурсы, лесное хозяйство). Точность, печать карт, визуализация трёхмерного изображения, библиотека алгоритмов.

ERDAS IMAGINE (разработка Leica) – программный пакет, разработанный специально для обработки и анализа данных дистанционного зондирования, предоставляет полный набор инструментов для анализа данных из любого источника и представление результатов в различных формах – от печатных карт до трёхмерных моделей. ERDAS IMAGINE построен по модульному принципу в виде базовых комплектов – IMAGINE Essential, IMAGINE Advantage и IMAGINE Professional.

В ERDAS IMAGINE реализованы:

- широкие возможности по визуализации и импорту данных (поддерживает более 100 форматов);
- геометрическая коррекция;
- улучшающие преобразования и ГИС-анализ;
- дешифрирование снимков;
- инструменты обработки изображений и построение алгоритмов пространственных вычислений;
- создание карт.

Лабораторная работа № 18

Тема: «Принцип действия клавиатуры и мыши»

1. Цель работы

Изучение принципа действия клавиатуры и мыши.

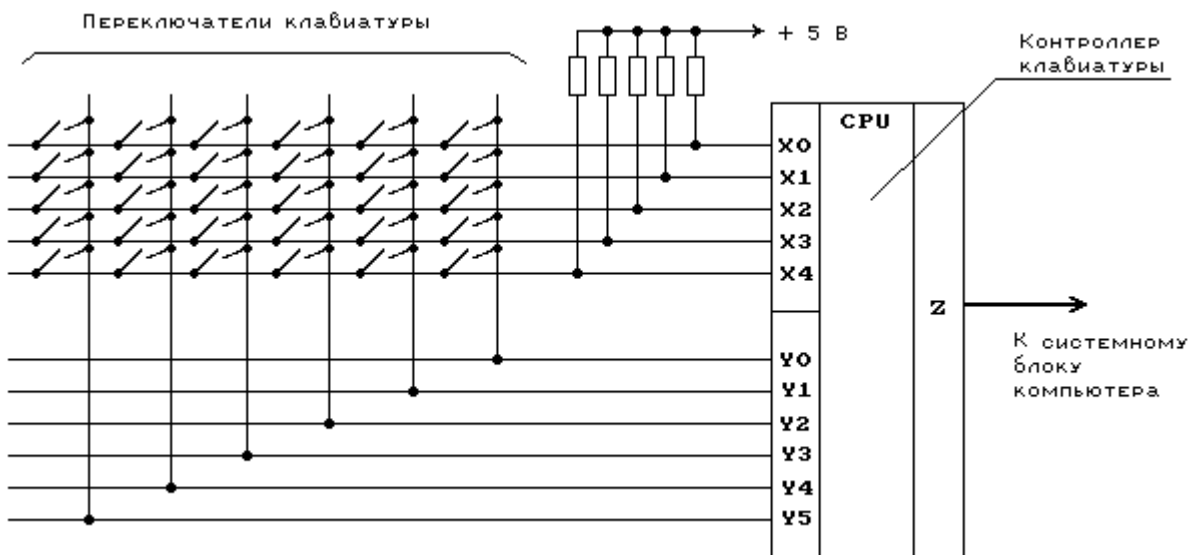
2. Теоретические сведения

Клавиатура

Клавиатура выполнена, как правило, в виде отдельного устройства, подключаемого к компьютеру тонким кабелем. Малогабаритные компьютеры Lap-Top используют встроенную клавиатуру.

Что же находится внутри клавиатуры? Оказывается, там есть компьютер! Только этот компьютер состоит из одной микросхемы и выполняет специализированные функции. Он отслеживает нажатия на клавиши и посылает номер нажатой клавиши в центральный компьютер.

Если рассмотреть сильно упрощенную принципиальную схему клавиатуры, представленную на рисунке, можно заметить, что все клавиши находятся в узлах матрицы:



Все горизонтальные линии матрицы подключены через резисторы к источнику питания +5 В. Клавиатурный компьютер имеет два порта - выходной и входной. Входной порт подключен к горизонтальным линиям матрицы (X0-X4), а выходной - к вертикальным (Y0-Y5).

Устанавливая по очереди на каждой из вертикальных линий уровень напряжения, соответствующий логическому 0, клавиатурный компьютер опрашивает состояние

горизонтальных линий. Если ни одна клавиша не нажата, уровень напряжения на всех горизонтальных линиях соответствует логической 1 (т.к. все эти линии подключены к источнику питания +5 В через резисторы).

Если оператор нажмет на какую-либо клавишу, то соответствующая вертикальная и горизонтальная линии окажутся замкнутыми. Когда на этой вертикальной линии процессор установит значение логического 0, то уровень напряжения на горизонтальной линии также будет соответствовать логическому 0.

Как только на одной из горизонтальных линий появится уровень логического 0, клавиатурный процессор фиксирует нажатие на клавишу. Он посылает в центральный компьютер запрос на прерывание и номер клавиши в матрице. Аналогичные действия выполняются и тогда, когда оператор отпускает нажатую ранее клавишу.

Номер клавиши, посылаемый клавиатурным процессором, однозначно связан с распайкой клавиатурной матрицы и не зависит напрямую от обозначений, нанесенных на поверхность клавиш. Этот номер называется скан-кодом (Scan Code).

Слово scan ("сканирование"), подчеркивает тот факт, что клавиатурный компьютер сканирует клавиатуру для поиска нажатой клавиши.

Но программе нужен не порядковый номер нажатой клавиши, а соответствующий обозначению на этой клавише ASCII-код. Этот код не зависит однозначно от скан-кода, т.к. одной и той же клавише могут соответствовать несколько значений ASCII-кода. Это зависит от состояния других клавиш. Например, клавиша с обозначением '1' используется еще и для ввода символа '!' (если она нажата вместе с клавишей SHIFT).

Поэтому все преобразования скан-кода в ASCII-код выполняются программным обеспечением. Как правило, эти преобразования выполняют модули BIOS. Для использования символов кириллицы эти модули расширяются клавиатурными драйверами.

Если нажать на клавишу и не отпускать ее, клавиатура перейдет в режим автоповтора. В этом режиме в центральный компьютер автоматически через некоторый период времени, называемый периодом автоповтора, посылается код нажатой клавиши. Режим автоповтора облегчает ввод с клавиатуры большого количества одинаковых символов.



Следует отметить, что клавиатура содержит внутренний 16-байтовый буфер, через который она осуществляет обмен данными с компьютером.

В настоящее время существует три различных типа клавиатуры. Это клавиатура для компьютеров IBM PC/XT, 84-клавишная клавиатура для IBM AT и 101-клавишная (расширенная) клавиатура для IBM AT. Некоторые клавиатуры имеют переключатель режима работы (XT/AT), расположенный на нижней крышке. Он должен быть установлен в правильное положение.

Принцип работы компьютерной мыши

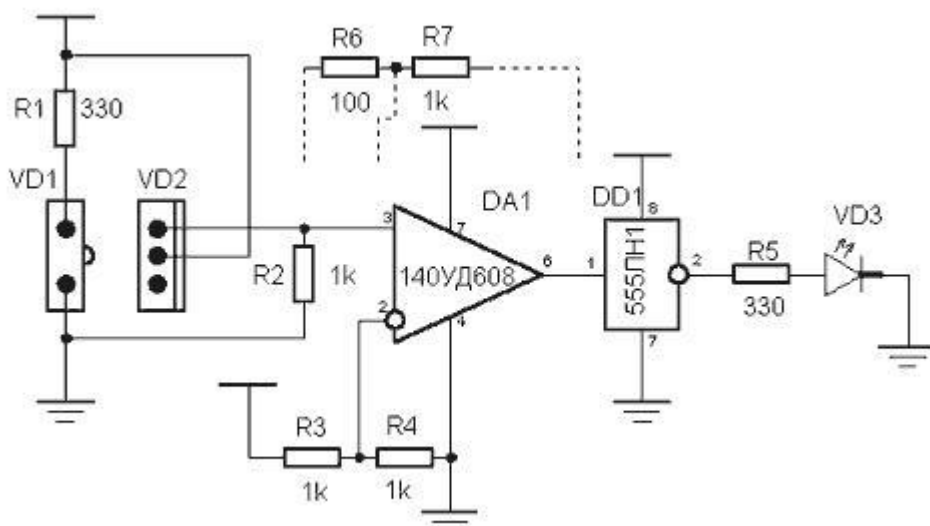
• Оптико-механические мыши



Это устройства на основе шарика и нескольких датчиков его перемещения. Они вполне подходят для большинства домашних и офисных задач. Основным недостатком можно назвать засорение и износ механических элементов (шарика, валиков). Впрочем, при должном уходе такая мышь может прослужить несколько лет, для этого лишь надо периодически ее чистить и использовать специальный коврик.

• Оптические мыши

В этих устройствах в качестве датчика перемещения используется система из источника света и фотозлемента, что позволяет точно регистрировать перемещения манипулятора. Оптические мыши необходимы большинству профессиональных пользователей (дизайнерам, конструкторам), а также всем пользователям, для которых важна повышенная точность при работе. За счет отсутствия подвижных механических элементов такие мыши весьма долговечны и практически не требуют специального ухода. Оптические модели надежны и неприхотливы: они устойчиво работают безо всякого коврика, на любой поверхности (за исключением лишь прозрачных и зеркальных, на которых возможны сбои позиционирования курсора).



• Датчик компьютерной мышки

Датчик разделен на две части: передающая (VD1) и принимающая (VD2). Передающая

это светодиод работающий в инфракрасном диапазоне, а принимающая это два фотодиода в одном корпусе. Понятно, что два - для того чтобы можно было определить в какую сторону крутится колесико мышки.

R1 - задает ток для светодиода. R2 служит для преобразования тока фотоприемника в напряжение (не использованная ножка VD2 это выход второго фотоприемника). Напряжение на этом сопротивлении меняется от 1.5В (нет света) до 3.4В (есть свет). Нижний предел в 1.5В слишком велик для переключения цифровой TTL схемы, в которой порог переключения составляет 0.8В. Поэтому используется операционный усилитель DA1 в режиме компаратора. Пороговое напряжение для него задает резистивный делитель R3, R4 которое составляет 2.5В. Можно считать, что с выхода DA1 идет уже цифровой сигнал.

Для уменьшения случайных наводок и ложных срабатывания, можно ввести гистерезис в уровни переключения DA1. Для этого нужно включить условно показанные резисторы R6, R7 (R6 в разрыв между R2 и DA1). Чем ближе номинал R6 к R7 тем шире будет петля гистерезиса или тем дальше друг от друга будут уровни переключения из "0" в "1" и из "1" в "0". Так при приведенных номиналах уровень переключения из "0" в "1" 2.8В, а из "1" в "0" 2.1В.

Далее для примера изображена простенькая схемка индикации из инвертора DD1 и светодиода. Когда между VD1 и VD2 будет находится предмет мешающий прохождению света, будет гореть светодиод VD3.

Необходимо учитывать возможность внешних, мешающих, воздействий. Датчик также хорошо принимает излучение, как на стороне передатчика так и с противоположной.



3 Подготовка к выполнению работы

- 3.1 Ознакомиться с инструкцией.
- 3.2 Проработать теоретический материал за темой работы.
- 3.3 Дать ответы на контрольные вопросы входного контроля.
- 3.4 Продумать методику выполнения работы.
- 3.5 Подготовить бланк отчета.

4 Оборудование:

Клавиатура , мышь.

5 Оформление отчета:

- 5.1 Цель работы.
- 5.2 Теоретические сведения.
- 5.3 Оборудование.
- 5.4 Задание.
- 5.5 Описание проделанной работы.

Лабораторная работа № 19

Тема: «Исследование работы дисплея ПК»

Цель работы: Определение типа адаптера, подключение и настройка, установка соответствующего драйвера видеоадаптера под различными операционными системами.

Используемое программное обеспечение: Тестирующие программы (WinCheckit, SysInfo, MSD и т.д.), графические операционные системы (Windows NT, Windows 3.1, OS/2, Windows 95), драйверы видеоадаптеров.

Содержание работы.

1. Дисплей:
 - определить тип монитора;
 - определить поддерживаемые режимы работы (все графические и текстовые);
 - определить разрешающую способность режимов.
2. Установка драйвера видеоадаптера под Windows 3.1.
3. Установка драйвера видеоадаптера под Windows 95.
4. Установка драйвера видеоадаптера под OS/2.
5. Смена видеорежимов.
6. Изменение частоты синхронизации.
7. Тестирование видеоподсистемы.
8. Русификация на уровне ОС и при помощи русификаторов текстового и графических режимов.
9. Определите тип видеоадаптера и доступ к периферии (в первую очередь – к дисплею), то есть определите уровень взаимодействия с ВС (с помощью BIOS, на уровне операционной системы MS DOS, прямое управление ВС через программно доступные регистры (порты I/O)).
10. Укажите характеристики наиболее распространенных видеоадаптеров: VGA (и SVGA) фирмы IBM (Video Graphic Array), MDA (Monochrome Display Adapter), CGA (Color Graphic Adapter), (Enhanced Graphic Adapter) – EGA, SVGA, Адаптер 8514/A, расширенная графическая матрица XGA (eXtended Graphic Array).
11. Определите тип локальной шины (VL-bus, PCI).
12. Какова основная функция ВА.
13. Определите размер видеобуфера (видеопамяти), являющегося частью адресного пространства ЦП.
14. Определите адресное пространство видеобуфера.
15. Что определяют группы бит в видеобуфере?
16. Опишите условно выделяемые логические блоки (в видеоадаптере VGA их шесть).
17. Рассчитайте требуемую емкость видеобуфера для поддерживаемых текстовых режимов (например, размер страницы в 16-ти-цветном текстовом режиме $(80 \cdot 25) \cdot 2 = 4000$ байт, а 16-ти-цветный графический режим с разрешением $640 \cdot 480$: $(640 \cdot 480 \cdot 4) / 8 = 153600$ байт (или 150 Кб)).

Лабораторная работа №20

Тема: «Исследование работы принтера»

Задание: Распечатать текст из файла таким образом: если строка начинается комбинации символов <F>, то она печатается в нижнем регистре.

Краткие теоретические сведения

Принтеры - это устройства вывода данных из ЭВМ, преобразующие информационные ASCII-коды в соответствующие им графические объекты (буквы, цифры, знаки и т.п.) и фиксирующие эти символы на бумаге. Принтеры являются наиболее широкой группой ВУ, насчитывающей до 100 различных модификаций.

Принтеры можно классифицировать по различным признакам:

- цветность (чerno-белые, цветные);
- способ формирования символов (знакопечатающие и знаковосинтезирующие);
- принцип действия (матричные, термические, струйные, лазерные);
- способы печати (ударные, безударные) и формирования строк (последовательные, параллельные);
- ширина каретки (с широкой (375-450 мм), с узкой (250мм) кареткой);
- длина печатной строки (80 и 132-136 символов);
- набор символов (вплоть до полного набора символов ASCII);
- скорость печати;
- разрешающая способность (наиболее употребительной единицей измерения является dpi (dots per inch - количество точек на дюйм)).

Многие принтеры обеспечивают эффективный вывод графической информации с помощью символов псевдографики, сервисные режимы печати, плотную печать, печать с двойной шириной, с подчеркиванием, с верхними и нижними индексами, выделенную печать (каждый символ печатается дважды), печать за два прохода (второй раз символ печатается с незначительным сдвигом) и многоцветную (до 100 различных цветов и оттенков) печать.

Передача управляющих кодов на принтер.

Большинство операций управляются программным обеспечением. Программа заставляет принтер выбрать определенный шрифт или подать бумагу на определенное расстояние после печати строки, начать печать на определенном месте страницы.

Некоторые программы позволяют Вам самим посылать команды на принтер. Это позволяет Вам улучшить формат печатаемых документов, используя способы, обычно недоступные из программного обеспечения. Например, если программа не поддерживает печать курсивом, Вы можете поместить управляющий код в соответствующем месте и использовать курсив там, где это необходимо. Команды, которые понимает Ваш принтер, приведены в справочнике команд.

Формат команд, который Вы используете зависит от программного обеспечения. Часть программ поддерживает только десятичный формат, часть - ASCII-формат. (Некоторые программы вообще не поддерживают использования управляющих

кодов). Кроме этого, программа, как правило, требует соблюдения соответствующей пунктуации при использовании управляющих кодов. Если используемая программа поддерживает управляющие коды, руководство по программному обеспечению должно точно описывать формат использования управляющих кодов.

Описание команд

Работа принтера

ASCII	Дес.	Шест.	Описание
ESC @	64	40	Инициализация принтера
DC1	17	11	Выбрать принтер
DC3	19	13	Отменить выбор принтера
ESC s 1/0	115	73	Включение/Выключение половинной скорости
ESC <	60	3C	Выбор однонаправленной печати (для одной строки)
ESC U	85	55	Включение/Выключение однонаправленной печати
ESC 8	56	38	Отключить определение отсутствия бумаги
ESC 9	57	39	Включить определение отсутствия бумаги

Управление данными

ASCII	Дес.	Шест.	Описание
CR	13	0D	Возврат каретки
CAN	24	18	Отменить строку
DEL	127	7F	Удалить символ

Стили печати

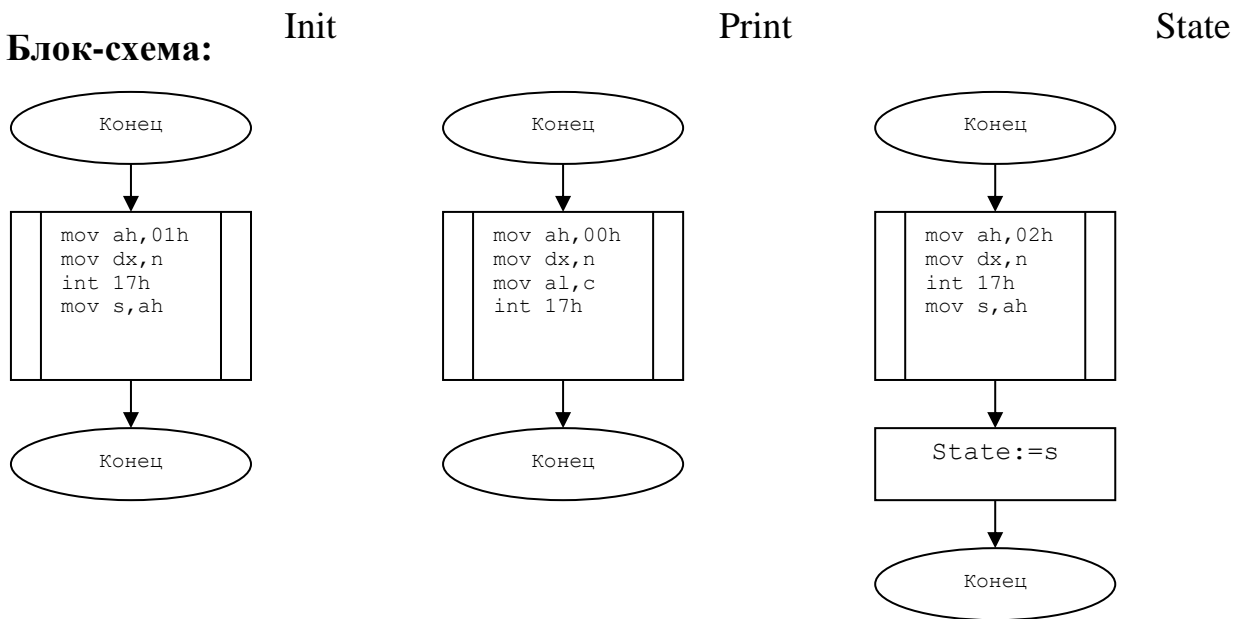
ASCII	Дес.	Шест.	Описание
ESC x n	120	78	Выбор NLQ или Draft n=0: Draft 1:NLQ
ESC k n	107	6B	Выбор шрифта высококачественной печати n=0: Roman 1: Sans Serif
ESC ! n	33	21	Выбор набора символов Для получения значения n, сложите вместе значения стилей списка приведенного ниже: n= 0:10 знд 1: 12 знд 2: пропорциональный 4: уплотненный 8: выделенный 16: двойной удар 32: двойная ширина 64: курсив 128: подчеркивание

Расширение возможностей печати

ASCII	Дес.	Шест.	Описание
ESC E	69	45	Включение режима выделенной печати
ESC F	70	46	Выключение режима выделенной печати
ESC G	71	47	Включение режима печати за два прохода
ESC H	72	48	Выключение режима печати за два прохода
ESC S0	83	53	Включение печати верхними индексами
ESC S1	83	53	Включение печати нижними индексами

ESC T	84	54	Выключение режима печати верхними и нижними
ESC 4	52	34	Включение курсива
ESC 5	53	35	Выключение курсива
ESC - 1/0	45	2D	Включить/выключить режим подчеркивания

Блок-схема:



Код программы:

```

program lab2;
  uses crt;
  var a:char;
      s:byte;
      st1,st2:string;
      n,i,j:integer;
      t:text;
      k:boolean;
procedure Init;
begin
  asm
    mov ah,01h
    mov dx,n
    int 17h
    mov s,ah
  end;
end;
procedure Print(c:char);
begin
  asm
    mov ah,00h
    mov dx,n
    mov al,c
    int 17h
  end;
end;
function State:byte;
begin

```

```

asm
    mov ah,02h
    mov dx,n
    int 17h
    mov s,ah
end;
State:=s;
end;
begin
    clrscr;
    assign(t,'lab2.txt');
    reset(t);
    writeln('Nomer printera (0,1,2): ');
    readln(n);
    Init;
    repeat
        if (s and 33 <>0) or (s and 152 <>152) then
            begin
                writeln('Error      Init.      Press      Esc      or      try
again(Enter) ');
                k:=false;
                Init;
            end
        else
            k:=true;
    until readkey=#27;
    if k=false then
        halt;
    s:=State;
    if (s and 33 <>0) or (s and 152 <>152) then
        begin
            writeln('Error printer:');
            if (s and 1=1) then
                writeln('Timeout error');
            if (s and 8<>8) then
                writeln('Printer error');
            if (s and 16<>16) then
                writeln('Printer is offline');
            if (s and 32=32) then
                writeln('No paper');
            if (s and 64<>64) then
                writeln('Printer is not connected');
            if (s and 128<>128) then
                writeln('Printer is busy');
            readln;
            halt;
        end;
    repeat
        read(t,st1);
        st1:=st1+#13#10;
        for i:=1 to length(st1) do
            begin
                if (copy(st1,i,3)='<F>') then

```

```
begin
    delete(st1,i,3);
    st2:=#27#83+st1+#27#84;
    for j:=1 to length(st2) do
        Print(st2[j]);
    end
end
else
    print(st1[i]);
end;
until eof(t);
writeln;
writeln('Pechat proshla uspeshno!');
close(t);
readln;
end.
```

Лабораторная работа №21

Тема: «Исследование устройства ввода и вывода звуковой и видеоинформации»

1. Назначение и состав систем мультимедиа

В технической литературе часто встречаются различные определения понятия «мультимедиа». Наиболее полно отражает существо вопроса следующее: мультимедиа — это компьютерная информационная технология, позволяющая объединять в компьютерной системе текст, звук, графические изображения и анимацию.

Согласно этому определению, основными отличительными признаками систем мультимедиа являются: интеграция в одном программном продукте многообразных видов информации: традиционных (текст, таблицы, иллюстрации) и оригинальных (речь, музыка, фрагменты видеофильмов и др.), а также наличие в составе компьютера специфических устройств, предназначенных для работы с различными видами информации (микрофона, аудиосистемы, проигрывателя оптических компакт-дисков, телевизора, видеомагнитофона, цифровой видеокамеры, электронных музыкальных инструментов и др.).

Появление возможности работы с новыми видами информации привело к тому, что компьютер в мультимедийном исполнении стал универсальным устройством, перерабатывающим различные виды информации. В ее число входит:

- **Числовая информация**: целые со знаком, целые без знака, числа с фиксированной и плавающей точкой, двоично-десятичные числа.

- **Логическая информация**, представляемая в виде логических переменных True и False (которые кодируются как 1 и 0).

- **Символьная информация**, представленная в виде скэн-кодов клавиатуры, кодов ASCII, Windows 1252, ANSI, кодов формы символов и полей переменной длины.

- **Аудиоинформация** (монотонный звуковой сигнал (меандр) — равномерный и затухающий; мелодия — гармоническая последовательность монотонных звуковых сигналов различной частоты и длительности; аккорд — совмещенная во времени смесь монотонных звуковых сигналов; звуковые эффекты; речь).

Вся аудиоинформация (в отличие от предыдущих типов) имеет динамический характер. Она существует только в режиме реального времени; ее нельзя остановить для более подробного изучения. Если изменить масштаб времени, аудиоинформация искажается. Это свойство иногда используется для получения звуковых эффектов.

- **Видеоинформация**: статическая и динамическая.

Статическая видеоинформация включает *штриховые рисунки* (к которым относятся графики, чертежи, таблицы) и *полутонные рисунки*, в которых форма предметов передается яркостью, освещенностью, цветом. Рисунки делятся на плоские (двухмерные) и объемные (трехмерные) и могут быть монохромными или цветными.

Для точности передачи изображения существенное значение имеет разрешающая способность считывающей или воспроизводящей аппаратуры, которая оценивается количеством наименьших элементов изображения на единицу длины.

Глаз человека способен различать около 200 оттенков серого. Поэтому требовать от аппаратуры большего разрешения не следует. Для оцифровки такого количества оттенков необходимо 8 бит на каждый пиксел, поэтому для хранения оцифрованного изображения размером 9х12 см при разрешающей способности 300 пиксел на дюйм необходимо около 1 Мбайта памяти.

Динамическая видеоинформация — это фильмы (видео-, мульт- и слайд-фильмы).

В их основе лежит последовательный показ на экране отдельных кадров вывода в соответствии со сценарием. Поэтому процесс обработки динамической видеoinформации часто называют «редактированием сценариев». Динамическая видеoinформация используется либо для передачи движущихся изображений (анимация), либо для последовательной демонстрации отдельных кадров вывода (слайд-фильмы).

При демонстрации более 16 кадров в секунду мелькание незаметно. Если кадры содержат различные фазы движения объекта, то такая демонстрация создает впечатление перемещения объекта.

Для обработки каждого вида информации существуют свои правила, свои операции, своя технология и свои программно-аппаратные средства. А поскольку разные виды информации могут использоваться совместно, то возникает проблема их сопряжения и преобразования из одного вида в другой.

При работе с видеoinформацией необходимы такие операции, как

- захват изображения с экрана с загрузкой его в файл;
- сжатие изображения и его восстановление;
- конструирование видеоэффектов;
- запись компьютерных изображений на видеомагнитофон и т.д.

Технология мультимедиа создает новый уровень интерактивного общения «человек — ЭВМ», когда в процессе диалога пользователь получает более обширную и разностороннюю информацию.

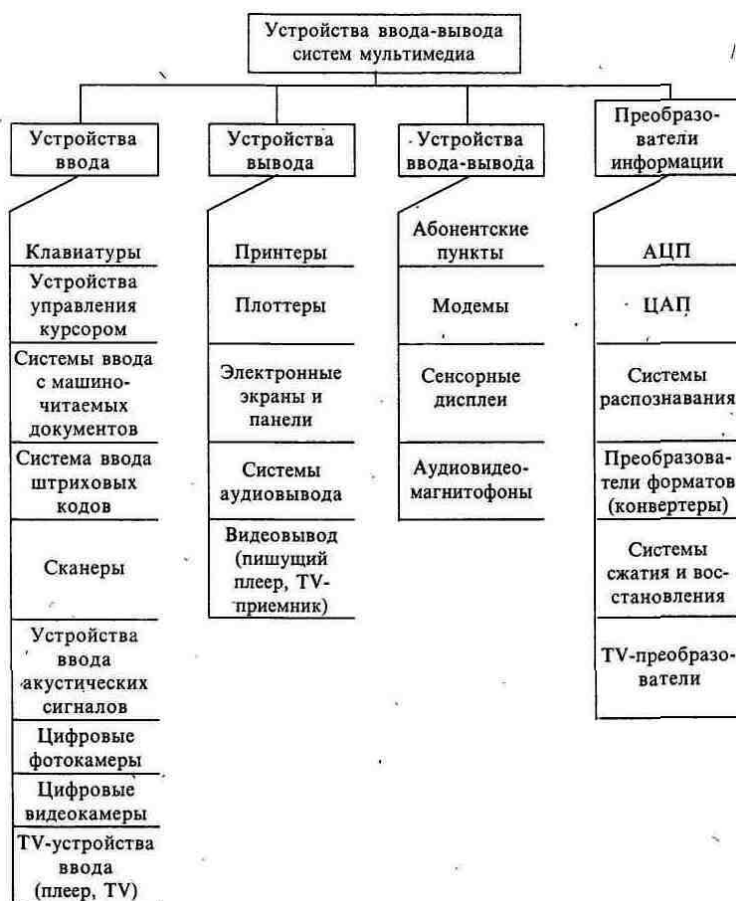
Применение мультимедиа привело к появлению новых источников информации — электронных книг, музейных и туристических путеводителей, обучающих, консультирующих, диагностических систем, рекламных роликов и др.

Специальным построением программного обеспечения в системах мультимедиа достигается эффект, который получил название *виртуальная реальность*. Этот эффект направлен на замену имеющейся у человека модели мира несуществующей, виртуальной моделью, в которой живут и действуют несуществующие в реальном мире персонажи (космические монстры, привидения и др.).

Системы мультимедиа работают в реальном режиме времени, что связано с использованием в них аудио- и видеoinформации. Поскольку аудио- и видеoinформация имеют аналоговую природу, ввод этих видов информации в ЭВМ и их вывод требуют применения специальных устройств, типы которых приведены на рис. 1. Как видно из рисунка, состав устройств весьма разнообразен.

Рис. 1. Устройства ввода-вывода систем мультимедиа

К *устройствам ввода* относятся клавиатура, различные устройства управления курсором (мышь,



джойстик/трекбол и др.), системы ввода с машиночитаемых документов, системы ввода штриховых кодов, сканеры, цифровые фото- и видеокамеры, TV-устройства ввода, устройства ввода акустических сигналов.

Устройства вывода - принтеры, плоттеры, электронные экраны и панели, системы аудиовывода, устройства видеовывода.

Устройства ввода-вывода – абонентские пункты, модемы, сенсорные дисплеи, CD-R и CD-RW.

Кроме того, в состав специальных устройств систем мультимедиа входят преобразователи информации: аналого-цифровые и цифро-аналоговые, системы распознавания, преобразователи форматов (конверторы), системы сжатия и восстановления аудио- и видеосигналов, преобразователи TV-сигналов в компьютерные и обратно.

2. Анимационные устройства ввода-вывода

Для создания и демонстрации компьютерной графики и анимации требуется:

- высокая производительность всего технического комплекса,
- специализированные преобразователи информации,
- технические средства для высококачественного отображения, ввода-вывода и хранения больших объемов информации.

Технический комплекс представляет собой последовательно соединенную систему с параллельными ветвями со стороны устройств ввода-вывода (УВВ).

Производительность такой системы зависит от

- быстродействия микропроцессорного комплекта,
- пропускной способности интерфейса ввода-вывода,
- производительности и способа подключения УВВ,
- наличия специальных «ускорителей» в различных устройствах,
- а также от принятой в системе технологии обмена информацией между отдельными частями технического комплекса (при этом нужно учитывать, что одним из элементов этого комплекса может являться человек-оператор).

Поскольку для воспроизведения фильмов необходимо выводить на экран не менее 24 кадров в секунду, а человеку для восприятия изображения надо не менее 30 сек, для их хранения нужны запоминающие устройства большого объема. Это удорожает системы и приводит к поиску способов сжатия информации.

Обилие разновидностей обрабатываемой в системах компьютерной графики и анимации информации приводит к необходимости использования различных устройств ввода: клавиатур, систем координатного ввода, оптических читающих устройств, устройств ввода акустической информации, анимационных устройств ввода и др. и соответствующих устройств вывода информации: дисплеев, графических экранных станций, синтезаторов речи, акустических систем, анимационных устройств вывода и др.

В состав анимационных устройств ввода-вывода входят: цифровая фотокамера, видеокамера, видеомаягнитофон и телевизор и преобразователи видеосигналов.

При кодировании сигнала изображения в ЭВМ применяются методы, отличные от методов кодирования телевизионного сигнала.

Преобразование TV сигналов в компьютерные заключается в выделении сигналов яркости Green, Blue, Red и синхроимпульсов, их очистке от сигналов звукового сопровождения и вычисления по ним сигналов Red, Green, Blue и адресов пиксел на экране.

Преобразование осуществляется видеоплатами (Video Blaster, Video Recorder, Video Converter), работающими с адаптером SVGA. Иногда видеоплаты (как, например, Tuner Blaster) имеют антенный вход, что позволяет обходиться без телевизионной аппаратуры.

Например, фирма Computer AG выпускает аппаратно-программные средства обработки ТВ изображений, позволяющих записывать изображение с ЭВМ на магнитофон (и считывать их обратно), воспроизводить изображение из ЭВМ на телеэкране. Видеоконвертер Micro Movie позволяет оцифровывать ТВ изображения, сохранять их в ЗУ ЭВМ для обработки графическим редактором, использования в публикациях, вставки их после редактирования и т.д. Аппаратный набор видеоконвертера состоит из платы адаптера дисплея, которая объединяет возможности адаптеров SVGA и адаптера Frame Grubber, оцифровывающего видеоизображение в реальном масштабе времени.

3. Устройства ввода-вывода звуковых сигналов

Системы мультимедиа начинались со звука, который воспринимается независимо от изображения, не наносит ущерба восприятию выводимой на экран информации, даже дополняет ее и повышает восприимчивость пользователя.

Однако звуковая информация имеет и самостоятельное значение. Можно выделить три направления в использовании звуковых возможностей систем мультимедиа:

- **бытовые системы мультимедиа** используют звуковые возможности ПЭВМ в обучающих, развивающих программах (обучение чтению, произношению, музыке); в энциклопедиях и справочниках. В бытовых системах использование музыкальных редакторов Skream Tracker, позволяет перейти на качественно новый уровень использования аудиосистем — от пассивного восприятия музыки к активной работе с музыкальными произведениями, например, к реализации цветомузыки на экране ПЭВМ;

- **мультимедиа бизнес-приложения** используют звук в целях презентации; проведения озвученных видеотелеконференций; голосовой почты; автоматического стенографирования; использования голоса пользователя в целях защиты (электронные замки, доступ к информации в ЭВМ, к банковским сейфам и др.);

- **профессиональные мультимедиа системы** — средства производства озвученных видеофильмов, домашние музыкальные студии. Музыкальные редакторы типа Skream Tracker, Whacker Tracker, Ezpno и др. позволяют наиграть мелодию, выполнить ее программную обработку (изменить высоту тона, длительность звучания, тип инструмента, скорость нажатия-отпускания клавиши, синтезировать звуковые эффекты), воспроизвести или записать на стандартную звукозаписывающую аппаратуру.

3.2. Ввод в ЭВМ и машинный синтез речи

Особое место в системах мультимедиа занимает использование аудиоаппаратуры для речевого общения.

Для распознавания и понимания речи необходимо ввести речевые сигналы в ЭВМ с помощью акустических устройств ввода и проанализировать вводимую речь.

Системы речевого ввода делятся на два типа по характеру распознаваемой речи:

- системы, ориентированные на восприятие отдельных слов;
- системы, воспринимающие связную речь.

Разница между ними весьма существенна, так как при слитном произношении слов изменяется их звучание.

При анализе отдельных слов (команд) осуществляется их оцифровка, идентификация и инициируется выполнение программы, отрабатывающей принятую команду. Этот же режим используется и для речевого ввода цифровой информации; в этом случае после идентификации введенное слово преобразуется в соответствующий код ASCII.

Фирма «Курцвейл» выпускала по этому принципу устройство Voice Writer, которое распознавало около 10000 отдельно произнесенных английских слов и печатало их на принтере.

Сегодня практически отсутствуют устройства для ввода динамически развивающихся звуковых сцен. Устройства ввода и программы-анализаторы не позволяют выделить эмоциональную составляющую речи, которая значительно корректирует смысл (и может даже изменить его до противоположного). Чаще всего эмоциональная составляющая рассматривается как помеха.

Системы речевого вывода называются *синтезаторами речи*.

Существуют 3 основных технологически различных подхода к проблеме синтеза речи:

- метод сжатия-восстановления формы сигналов;
- аналоговый метод синтеза формантных частот;
- цифровое моделирование голосового тракта.

Первый метод — самый простой. ЭВМ в этом случае служит как цифровой магнитофон. Фразы и слова записываются отдельно и выбираются для воспроизведения в нужный момент по командам, поступающим от соответствующей программы. В такой системе невозможно воспроизвести слово, которое не было заранее записано.

Для хранения оцифрованной речи требуется память большого объема, т.к. как хранить необходимо каждое слово с учетом различных падежных окончаний, рода, числа и т.д. Но зато качество воспроизведения речи очень высокое.

Ход работы:

1. Подключите видеоисточник к ПК
2. Произведите видеозахват фрагмента
3. Запишите получившийся файл
4. Заполните бланк отчета с описанием своих действий.

Контрольные вопросы:

1. Принцип действия платы видеозахвата
2. Принцип действия платы видеовывода
3. Принцип речевых синтезаторов
4. Принцип систем распознавания голоса

Литература

Основные источники

1. Колдаев В.Д. Архитектура ЭВМ: учеб. пособие для СПО. –М.: ИД ФОРУМ: НИЦ Инфра-М. 2016.
2. Сенкевич А.В. Архитектура ЭВМ и вычислительные системы –М.: ОИЦ «Акаде-мия», 2016
3. Гук М. Процессоры Pentium 4, Athlon и Duron. - Питер, 2015. – 512 с.
4. Крейгон Х. Архитектура компьютеров и ее реализация (пер. с англ.). Учеб. изд. – Мир, 2014. – 416 с.
5. Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем. Учебник: Профессиональное образование, ИНФРА-М, 2015. – 512 с.
6. Михеева Е. В. Информационные технологии в профессиональной деятельности: Учеб. пособие для сред. проф. образования. — М.: Издательский центр «Академия», 2014.
7. Петров В.А., Пискарев А.С., Шеин А.В. Информационная безопасность. Защита информации от несанкционированного доступа в автоматизированных системах: Учеб. пособие. М.: МИФИ, 2015. – 564с.
8. Пятибратов А.П., Гудыно Л.П., Кириченко А.А. Вычислительные системы, сети и телекоммуникации, М:Финансы и статистика, 2014. – 559 с.
9. Партыка Т. Л., Попов И.И. Периферийные устройства вычислительной техники: учеб. пособие. – М.: ФОРУМ: ИНФРА-М, 2014.

Дополнительные источники:

1. Воеводин В.В. Параллельные вычисления. Учебное пособие для вузов. – СПб.: БХВ-Петербург, 2008. – 600 с.
2. Гук М. Шины PCI, USB и FireWire: Энциклопедия. – СПб.: Питер, 2009. – 539 с.
3. Таненбаум Э. Архитектура компьютера. –5 изд-е. – СПб.: Питер, 2007. – 844 с.

Интернет-ресурсы :

1. Архитектура ЭВМ и систем: Учебно-методическое пособие. http://window.edu.ru/window/library?p_rid=59570.
2. Каталог образовательных ресурсов: [http://www.edu.ru/modules.php?op=modload&name=Web_Links&file=index&l_op=viewnk&cid=2767&fids\[\]=2666](http://www.edu.ru/modules.php?op=modload&name=Web_Links&file=index&l_op=viewnk&cid=2767&fids[]=2666).
3. Технические характеристики аппаратных платформ <http://www.hosting.ulstu.ru>.
4. Центр Информационных Технологий: <http://www.citforum.ru>.

