

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Пономарева Светлана Викторовна
Должность: Проректор по УР и НО
Дата подписания: 03.08.2022 23:09:38
Уникальный программный ключ:
bb52f959411e646175b6ef2977697e87139b1a2d



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ДГТУ)

Колледж экономики, управления и права

Методические указания по организации
практической работы студентов
по учебной дисциплине
Основы архитектуры, устройство и функционирование вычислительных систем

09.02.04 Информационные системы (по отраслям)

Ростов-на-Дону

2018


Методические рекомендации по учебной дисциплине «Основы архитектуры, устройство и функционирование вычислительных систем» разработаны с учетом по специальности среднего профессионального образования (далее – СПО) технического профиля: Информационные системы (по отраслям) предназначены для обучающихся и преподавателей колледжа.

Методические рекомендации разработаны на основе требований ФГОС среднего общего образования, предъявляемых к структуре, содержанию и результатам освоения учебной дисциплины «Основы архитектуры, устройство и функционирование вычислительных систем».

Составитель (автор): А.С Пегливанова преподаватель колледжа ЭУП

Рассмотрены на заседании предметной (цикловой) комиссии специальности общеобразовательных дисциплин


Протокол № 1 от «31» 08 2018 г

Председатель предметной (цикловой) комиссии  С.В.Шинакова

личная подпись

и одобрены решением учебно-методического совета колледжа.

Протокол № 1 от «31» 08 2018 г

Председатель учебно-методического совета колледжа комиссии  С.В.Шинакова

личная подпись

Рекомендованы к практическому применению в образовательном процессе

Рецензенты:

Содержание:	
Практическая работа № 1-2.....	5
Тема: «Изучение принципов представления информации в десятичной, двоичной, восьмеричной, шестнадцатеричной системе счисления	5
Практическая работа № 3.....	11
Тема: Использование обратного и дополнительного двоичных кодов для реализации всех арифметических операций.	11
Практическое занятие № 4	13
Тема: «Изучение основных логических функций и принципов работы логических элементов»	13
Практическая работа №5.....	15
Тема: работа и особенности работы логических элементов ЭВМ.....	15
Практическое занятие № 6	20
Тема: «Изучение работы сумматоров различных типов»	20
Практическое занятие №7	22
Тема: «Изучение работы триггеров и принципы их работы».....	22
Практическая работа № 8-9.....	26
Тема: «Знакомство с моделью учебной ЭВМ».....	26
Практическая работа № 10	34
Тема: «Регистры процессора. Память.»	34
Практическое занятие № 11	38
Тема: «Подключение сетевых устройств, установка и настройка оборудования, работа с вычислительной сетью»	38
Практическая занятие № 12	46
Тема: «Принципы работы кэш-памяти»	46
Практическое занятие № 13	49
Тема: «Алгоритмы замещения строк кэш-памяти»	49
Практическое занятие №14	51
Тема: «Организация памяти в компьютерах типа ibm pc».....	51
Практическое занятие № 15-16.....	54
Тема: « Команды дисковой операционной системы, каталогов и файлов»	54
Практическая работа № 17-18.....	62
Тема: «Настройка BIOS»	62
Практическая работа № 19-20.....	75
Тема: «Установка windows xp на виртуальную машину virtualbox».	75

Практическая работа № 1-2

Тема: «Изучение принципов представления информации в десятичной, двоичной, восьмеричной, шестнадцатеричной системе счисления (ССЧ)»

Цель работы: изучить принципы представления информации в десятичной системе счисления, изучить принципы представления информации в двоичной системе счисления.

Теоритические сведения

Система счисления — способ записи чисел с помощью заданного набора специальных символов (цифр) и сопоставления этим записям реальных значений. Все системы счисления можно разделить на непозиционные и позиционные. В непозиционных системах счисления, которые появились значительно раньше позиционных, смысл каждого символа не зависит от того места, на котором он стоит. Примером такой системы счисления является римская, в которой для записи чисел используются буквы латинского алфавита. При этом буква I всегда означает единицу, буква — V пять, X — десять, L — пятьдесят, C — сто, D — пятьсот, M — тысячу и т.д. Например, число 264 записывается в виде CCLXIV. Недостатком непозиционных систем является отсутствие формальных правил записи чисел и, соответственно, арифметических действий с многозначными числами. Правила выполнения вычислений с многозначными числами в позиционной системе счисления были разработаны средневековым математиком Мухамедом аль-Хорезми и в Европе были названы алгоритмами (от латинского написания имени аль-Хорезми – Algorithmi).

В вычислительной технике применяются **позиционные системы счисления**. Позиционных систем счисления существует множество и отличаются они друг от друга алфавитом — множеством используемых цифр. Размер алфавита (число цифр в нем) называется основанием системы счисления. Последовательная запись символов алфавита (цифр) изображает число. Позиция символа в изображении числа называется разрядом. Разряду с номером 0 соответствует младший разряд целой части числа. Каждому символу соответствует определенное число, которое меньше основания системы счисления. В зависимости от позиции (разряда) числа значение символа умножается на степень основания, показатель которой равен номеру разряда.

Таким образом, целое положительное число A в позиционной системе счисления можно представить выражением:

$$A = a_n a_{n-1} \dots a_1 a_0 = a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p^1 + a_0 p^0 \quad 1)$$

$$\text{или } A = \sum_{k=0}^n a_k p^k, \text{ где } p \text{ — основание системы счисления, целое положительное число;}$$

a — символ (цифра); n — номер старшего разряда числа.

Обозначения цифр берутся из алфавита, который содержит p символов. Каждой цифре соответствует определенный количественный эквивалент. Обозначение a_k следует понимать как цифру в k -м разряде. Всегда выполняется неравенство: $a_k < p$.

Запись $A(p)$ указывает, что число A представлено в системе счисления с основанием p :

$$A_{(p)} = a_n a_{n-1} \dots a_1 a_0. \quad (2)$$

Примером системы счисления является всем нам хорошо известная десятичная система счисления. Любое число в ней записывается с помощью цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Важно, что значение каждой цифры зависит от того места, на котором она стоит в этой записи. Например, 1575: цифра 5 в записи числа встречается дважды: цифра 5 в последнем разряде — число единиц, а цифра 5, находящаяся в записи числа левее, — число сотен. Т.к. значение каждой цифры (ее "вес") определяется той позицией, которую цифра занимает в записи числа, то система счисления называется позиционной. В десятичной системе счисления значение единицы каждого разряда в 10 раз больше единицы соседнего с ним правого разряда.

Само число 10 называется основанием системы счисления, а цифры, используемые в десятичной системе — базисными числами этой системы.

Но в качестве основания системы счисления можно выбрать любое целое число. Чтобы отличить, в какой системе счисления записано число, будем указывать основание системы счисления в виде индекса в десятичной системе счисления, заключенного в круглые скобки. Если основание системы счисления равно 10 или очевидно из контекста, то индекс будет опущен.

Двоичная система счисления имеет набор цифр $\{0, 1\}$, $p=2$. В общем виде, используя формулу (1), двоичное число можно представить выражением:

$$A_{(2)} = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \dots + a_1 \cdot 2^1 + a_0 \cdot 2^0 \quad 3)$$

Например, число 101101(2) можно записать так:

$$101101(2) = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

Двоичная система счисления имеет особую значимость в информатике: внутреннее представление любой информации в компьютере является двоичным, т.е. описывается набором символов только из двух знаков 0 и 1.

Шестнадцатеричная система счисления имеет набор цифр $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$, $p = 16$. Для изображения чисел в шестнадцатеричной системе счисления требуются 16 цифр. Для обозначения первых десяти цифр используются цифры десятичной системы счисления, шесть остальных — первых шесть прописных букв латинского алфавита. По формуле (1) шестнадцатеричное число может быть представлено так:

$$A_{(16)} = a_n \cdot 16^n + a_{n-1} \cdot 16^{n-1} + \dots + a_1 \cdot 16^1 + a_0 \cdot 16^0 \quad (4)$$

Пример

1. Число E7F8140 по формуле (4) запишется так:

$$E \cdot 16^6 + 7 \cdot 16^5 + F \cdot 16^4 + 8 \cdot 16^3 + 1 \cdot 16^2 + 4 \cdot 16^1 + 0 \cdot 16^0$$

Представление информации, хранящейся в памяти компьютера, в ее истинном двоичном виде весьма громоздко из-за большого количества цифр. Поэтому при записи такой информации на бумаге или выводе ее на экран принято использовать восьмеричную или шестнадцатеричную системы счисления. В современных компьютерах чаще используется шестнадцатеричная система счисления.

Полезно помнить некоторые степени двойки и шестнадцати.

k	1	2	3	4	5	6	7	8	9	10	11	12
2 ^k	2	4	8	16	32	64	128	256	512	1024	2048	4096
16 ^k	16	256	4096	65536	1048576							

Соответствие чисел в различных системах счисления

Десятичная	Шестнадцатеричная	Двоичная
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

В восьмеричной системе счисления числа записываются с помощью восьми цифр: 0, 1, 2, 3, 4, 5, 6, 7. Сама восьмерка (как и основание любой системы счисления) записывается двумя цифрами: 10 (почему?). Для записи чисел в шестнадцатеричной системе счисления необходимо располагать уже шестнадцать различными символами, используемыми как цифры. В качестве первых десяти шестнадцатеричных цифр используются цифры десятичной системы счисления. Для обозначения остальных шести цифр (в десятичной системе они соответствуют числам 10, 11, 12, 13, 14, 15) используют буквы латинского алфавита — А, В, С, D, E, F. Особая привязанность программистов к восьмеричной и шестнадцатеричной системам счисления объясняется тем, что переход к записи числа в любой из этих систем от его двоичной записи очень прост, так как 8 и 16 являются степенями числа 2.

Полезно привести таблицу:

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

При переводе в восьмеричную систему счисления двоичное число из трех и менее цифр записывается одной восьмеричной цифрой.

При переводе в шестнадцатеричную систему счисления двоичное число из четырех и менее цифр записывается одной шестнадцатеричной цифрой.

Например:

1. Перевести числа из двоичной системы счисления в восьмеричную и шестнадцатеричную: 11100011; 101110,1; 111111,01001.
2. Перевести числа из восьмеричной системы счисления в двоичную: 567; 204,72.
Перевести числа из шестнадцатеричной системы счисления в двоичную: 4D2A; 31,05; 40A,7.

Вопросы для самоконтроля

1. Какими символами обозначаются системы счисления: десятичная, двоичная, шестнадцатеричная, восьмеричная?
2. К какому типу счисления относится двоичная система счисления?
3. Перечислите правила сложения двоичных чисел?
4. Перечислите правила умножения двоичных чисел?
5. Какой основной недостаток двоичных чисел?
6. Каким образом производится переход из восьмеричной системы счисления в двоичную?
7. Каким образом производится переход из десятичной системы счисления в восьмеричную?
8. Каким образом производится переход из шестнадцатеричной системы счисления в двоичную?
9. Каким образом производится переход из шестнадцатеричной системы счисления в восьмеричную?

Выполните следующие задания:

1. Представьте следующие десятичные числа в виде позиционной записи:
а) 576; б) 842,3; в) 1924,803; г) 1000; д) 0100,0001; е) 0,002; ж) 25,75;
з) 89; и) 13,5; к) 0,25; л) 834,25; м) 34226; н) 236,14
2. Имеются позиционные записи десятичных чисел:
а) $8 \cdot 10^2 + 5 \cdot 10^1 + 3 \cdot 10^0 + 7 \cdot 10^{-1} + 6 \cdot 10^{-2}$;
б) $0 \cdot 10^4 + 1 \cdot 10^3 + 8 \cdot 10^2 + 4 \cdot 10^1 + 0 \cdot 10^0 + 0 \cdot 10^{-1} + 9 \cdot 10^{-2}$;
в) $9 \cdot 10^5 + 4 \cdot 10^3 + 3 \cdot 10^0 + 4 \cdot 10^{-2} + 4 \cdot 10^{-3}$;
г) $6 \cdot 10^2 + 1 \cdot 10^1 + 4 \cdot 10^0 + 8 \cdot 10^{-1} + 6 \cdot 10^{-2}$;
д) $1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2}$;
3. Переведите римскую запись в арабскую:
а) LX; б) XL; в) CXI; г) IXC; д) MDCCCXII; е) MCMLXI
4. Переведите арабскую запись чисел в римскую:
а) 45; б) 55; в) 900; г) 1500; д) 1554; е) 1917
5. Переведите в двоичную запись десятичные числа:
7; 17; 31; 48; 98; 102; 193; 254; 513; 999; 25,75; 1505,25

6. Переведите в десятичную запись десятичные числа:

101; 1001; 1100; 10111; 11011; 1011000; 10111011; 100010011; 10000000011;
010101010101; 11001,11; 1011001,01; 100101,01

Контрольные вопросы

1. Что называется системой счисления?
2. На какие два типа можно разделить все системы счисления?
3. Какие системы счисления называются непозиционными? Почему? Приведите пример такой системы счисления и записи чисел в ней?
4. Какие системы счисления применяются в вычислительной технике: позиционные или непозиционные? Почему?
5. Какие системы счисления называются позиционными?
6. Как изображается число в позиционной системе счисления?
7. Что называется основанием системы счисления?
8. Что называется разрядом в изображении числа?
9. Как можно представить целое положительное число в позиционной системе счисления?
10. Приведите пример позиционной системы счисления.
11. Опишите правила записи чисел в десятичной системе счисления: какие символы образуют алфавит десятичной системы счисления?
12. Что является основанием десятичной системы счисления?
13. Как изменяется вес символа в записи числа в зависимости от занимаемой позиции?
14. Какие числа можно использовать в качестве основания системы счисления?
15. Какие системы счисления применяются в компьютере для представления информации?
16. Охарактеризуйте двоичную систему счисления: алфавит, основание системы счисления, запись числа.
17. Почему двоичная система счисления используется в информатике?

Так как сумма является кодом отрицательного числа (знак 1), то необходимо перевести результаты в прямой код:

- из обратного кода $(X+Y)_{\text{обр}}=1,1110100$ $(X+Y)_{\text{пр}}=1,0001011$;

- из дополнительного кода $(X+Y)_{\text{доп}}=1,1110101$ $(X+Y)_{\text{пр}}=1,0001010+0,0000001=1,0001011$.

Таким образом, $X+Y = -1011$ и полученный результат совпадает с обычной записью

2. Задания для самостоятельного выполнения.

Задание 1. Вычислить выражение $-3_{(10)} - 2_{(10)}$ в прямом и обратном коде.

Задание 2. Вычислить выражение $7_{(10)} - 3_{(10)}$ в прямом и обратном коде.

Задание 3. Число $-5_{(10)}$ перевести в дополнительный код и обратно.

Задание 4. Число $-44_{(10)}$ ($10101100_{(2)}$) перевести в дополнительный код и

обратно Задание 5. Перевести в дополнительный код модуль числа -44 .

Задание 6. Вычислить алгебраическую сумму $26 - 34$. (с использованием обратного кода).

Задание 7. Перевести число 1110 из дополнительного кода в десятичную систему.

Практическое занятие № 4

Тема: «Изучение основных логических функций и принципов работы логических элементов»

Цель работы: изучить основные логические функции и принципы работы логических элементов.

Теоретические сведения

Основным понятием математической логики является понятие высказывания. Высказывание — предложение, про которое всегда можно сказать, истинно оно или ложно. Высказывания бывают простые и сложные. Сложное высказывание состоит из простых, соединенных знаками логических операций. Простые высказывания обычно обозначают большими латинскими буквами: A, B, C , — и т.д.

Рассмотрим следующие логические операции:

не(отрицание),

или(конъюнкция),

и(дизъюнкция),

→(следование).

Логические операции определяются через таблицы истинности.

Пусть имеются два высказывания: A — юноша в школе; B — юноша на уроке. Составим таблицы, которые показывают смысл операций и, или, не.

Операция “и”		
A	B	A и B
И	И	И
И	Л	Л
Л	И	Л
Л	Л	Л

Операция “не”	
A	не A
И	Л
Л	И

Операция “или”		
A	B	A и B
И	И	И
И	Л	И
Л	И	И
Л	Л	Л

Пусть имеются два высказывания: A — у человека высокая температура, B — человек болен. Составим таблицу, которая показывает смысл операции следования.

A	B	Из A следует B
И	И	И
И	Л	Л
Л	И	И
Л	Л	И

Решение задач.

Дана таблица:

Формула	Высказывание	Тигр	Волк	Бурундук	Заяц
A	Зверь полосатый				
B	Зверь хищный				
не A					
не B					
A и B					
A или B					

Ответ:

Формула	Высказывание	Тигр	Волк	Бурундук	Заяц
A	Зверь полосатый	И	Л	И	Л
B	Зверь хищный	И	И	Л	Л
не A	Зверь не полосатый	Л	И	Л	И
не B	Зверь не хищный	Л	Л	И	И
A и B	Зверь полосатый и хищный	И	Л	Л	Л
A или B	Зверь полосатый или хищный	И	И	И	И

Вопросы для самоконтроля

1. Операция конъюнкции. Таблица истинности.
2. Операция дизъюнкции. Таблица истинности.
3. Операция отрицания. Таблица истинности.

Выполните следующие задания

1. Пользуясь формулой, построить схему комбинационного устройства.

$$f = a \wedge b \vee a \wedge \bar{b}$$

$$f = a \vee b \wedge (\bar{b} \vee a) \wedge a$$

$$f = a \wedge (a \vee b) \vee (b \wedge a)$$

Практическая работа №5.

Тема: работа и особенности работы логических элементов ЭВМ.

Цель работы: теоретическое изучение логических элементов, реализующих элементарные функции алгебры логики (ФАЛ).

Задачи:

1. Изучить логические элементы, реализующие элементарные функции алгебры логики (ФАЛ).
2. Выполнить задания по теме (решение задач).
3. Оформить отчет по лабораторной работе и представить преподавателю.

Теоретические сведения

Логическое выражение состоит из логических операндов, соединенных с помощью логических операций. В качестве логических операндов могут выступать логические константы, переменные, а также отношения (сравнения) между двумя величинами. Логические выражения могут принимать одно из двух значений: ИСТИНА (TRUEили 1), ЛОЖЬ (FALSEили 0).

Существует несколько логических операций, все возможные значения которых описывают обычно с помощью таблиц истинности (это возможно по той причине, что все сочетания значений логических операндов очень легко перечислить) (табл. 4.1).

Приоритет операций при вычислении значения логического выражения следующий (в порядке понижения):

- 1) отрицание (NOT, НЕ);
- 2) конъюнкция (AND, И);
- 3) дизъюнкция и исключающее ИЛИ (OR, ИЛИ;XOR, ИСКЛЮЧАЮЩЕЕ ИЛИ);
- 4) операции отношения (равно, не равно, больше, меньше, больше или равно).

Если существует необходимость изменения порядка вычисления значения выражения, надо использовать круглые скобки. Чаще всего это применяется к операциям отношения, поскольку они имеют самый низкий приоритет, а их чаще всего необходимо вычислить в первую очередь.

Например, вычислим значение выражения $(a \leq b) \text{ OR } (c \neq b)$ при $a=2, b=3, c=3$:

- 1) $2 \leq 3 \rightarrow \text{TRUE}$;
- 2) $3 \neq 3 \rightarrow \text{FALSE}$;
- 3) $\text{TRUE OR FALSE} \rightarrow \text{TRUE}$.

Логические элементы

При всей сложности устройства электронных блоков современных ЭВМ выполняемые ими действия осуществляются с помощью комбинаций относительно не большого числа типовых логических узлов.

Основные из них:

- регистры;
- комбинационные преобразователи кодов (шифратор, мультиплексор и др.);
- счетчики (кольцевой, синхронный, асинхронный и др.);
- арифметико-логические узлы (сумматор, узел сравнения и др.).

Из этих узлов строятся интегральные микросхемы очень высокого уровня интеграции: микропроцессоры, модули ОЗУ, контроллеры внешних устройств и т.д.

Сами указанные узлы собираются из основных базовых логических элементов как простейших, реализующих логические функции И, ИЛИ, НЕ, И—НЕ, ИЛИ—НЕ и им подобных (элементы комбинационной логики, для которых значение функции на выходе однозначно определяется комбинацией входных переменных в данный момент времени), так и более сложных, таких как триггеры (элементы последовательностной логики, для которых значение функции зависит не только от текущих значений переменных на входе, но и от их предшествующих значений).

Условные обозначения основных элементов комбинационной логики приведены на рис. 8.1, соответствующие значения переменных («таблицы истинности») в табл. 8.1. Отметим, что кружочек на схеме на выходе из логического элемента означает, что элемент производит логическое отрицание результата операции, указанной внутри прямоугольника.

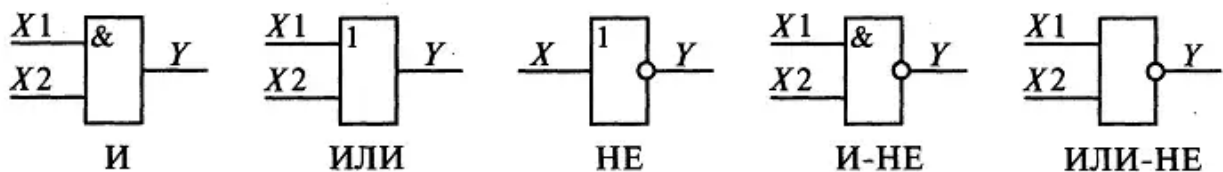


Рис.8.1.Основные элементы комбинационной логики.

Таблица 8.1. истинности логических операций

Задание 1. Найти значение приведенных ниже выражений;

- 1) $x > y$ при а) $x = 2, y = 2$;
б) $x = 2, y = -8$;
- 2) $A \text{ OR } B \text{ AND NOT } C$ при $A = \text{False}, B = \text{True}, C = \text{False}$;
- 3) $\text{NOT}(A < B)$ при а) $A = 7, B = 9$;
б) $A = 0, B = 2$;
- 4) $(x < y) \text{ OR } (x = z)$ при а) $x = 0, y = 0, z = 0$;
б) $x = 0, y = -8, z = 0$;
- 5) $(a \leq z) \text{ AND } (z > 2) \text{ AND } (a \neq 5)$ при а) $a = 2, z = 4$;
б) $a = -5, z = 0$;
- 6) $A \leq B$ при а) $A = 2, B = 2$;
б) $A = 2, B = -8$;

7) $A \text{ AND } B \text{ OR } \text{NOT } C$ при $A = \text{False}$, $B = \text{True}$, $C = \text{False}$;

8) $\text{NOT}(x \geq y)$ при а) $x = 7$, $y = 9$;

б) $x = 0$, $y = 2$;

9) $(x < y) \text{ AND } (x = z)$ при а) $x = 0$, $y = 0$, $z = 0$;

б) $x = 0$, $y = -8$, $z = 0$;

10) $(a \leq z) \text{ OR } (z > 2) \text{ OR } (a \neq 5)$ при а) $a = 5$, $z = -4$;

б) $a = -5$, $z = 0$;

Задание 2. По заданной логической схеме (рис.4.2) составить логическое выражение и выполнить для него таблицу истинности.

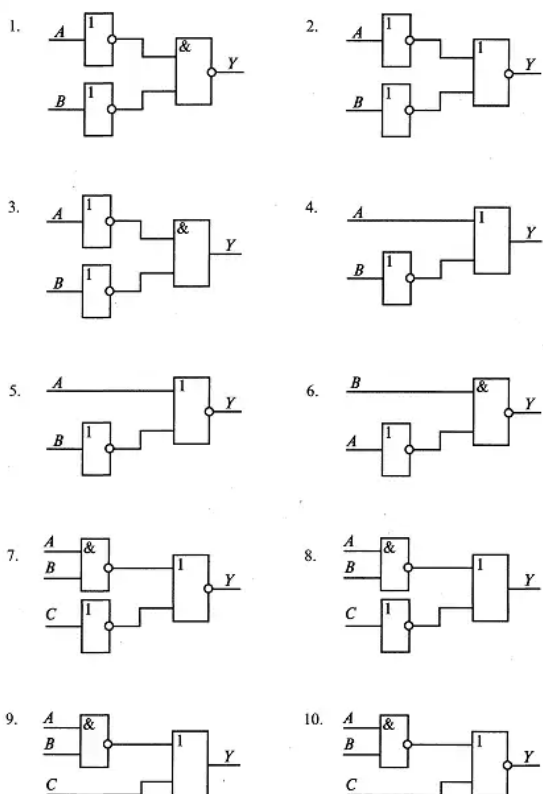


Рис. 8.2. Логические схемы

Задание 3. По заданному логическому выражению составить логическую схему и построить таблицу истинности:

1. $A \text{ AND } B \text{ OR } \text{NOT } C$.

2. $A \text{ AND } \text{NOT } B \text{ OR } C$;

3. $\text{NOT}(A \text{ AND } \text{NOT } B) \text{ OR } C$

4. $A \text{ OR } \text{NOT } B \text{ AND } C$

5. $A \text{ OR } \text{NOT}(\text{NOT } B \text{ AND } C)$;

6. $\text{NOT}(A \text{ OR } B) \text{ AND } \text{NOT } C$,

7. $\text{NOT}(A \text{ AND } B) \text{ OR } \text{NOT } C$,

8. $\text{NOT } A \text{ OR } B \text{ AND } C$,

9. $\text{NOT}(\text{NOT } A \text{ OR } B \text{ OR } C)$;

10. NOT (NOT A OR B AND NOT C).

Задание 4. Логические элементы И—НЕи ИЛИ—НЕ называют базовыми, поскольку любой из перечисленных на рис. 4.1 логических элементов можно выразить только через И—НЕ (или ИЛИ—НЕ). Соответствующие схемы для одного из этих случаев приведены на рис. 8.3.

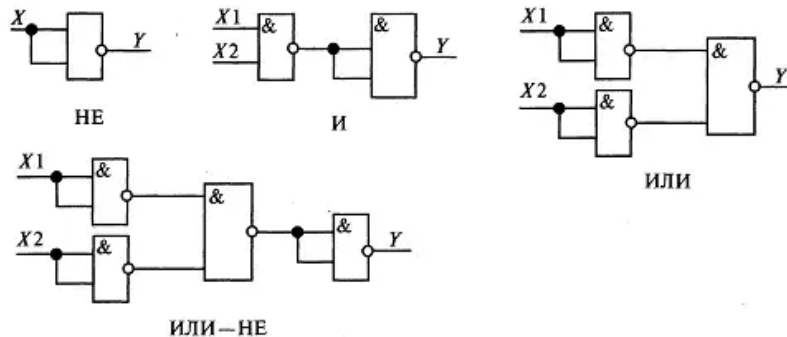


Рис.8.3 Реализация логических элементов через базовый И-НЕ

Для того чтобы убедиться в справедливости сформулированного выше утверждения, достаточно перебрать все возможные комбинации входных сигналов и найти результат. Покажем это на примере схемы для «И»; промежуточный результат обозначим через Z(табл. 8.2).

Таблица 8.2.Реализация схемы «И»

Таким образом, сравнивая с табл. 8.1, убеждаемся в справедливости высказанного выше утверждения.

Выполнить указанную проверку для всех схем на рис. 8.3.

Разработать схемы реализации элементов НЕ, И, ИЛИ, И—НЕ через базовый логический элемент ИЛИ-НЕ.

Задание 5. Кроме указанных выше одно- и двухвходовых элементов комбинационной логики, используют и более сложные — трех-, четырехвходовые и др., реализующие определенные логические функции более чем двух аргументов. Один из таких элементов изображен на рис. 8.4. (а); он реализует действие-

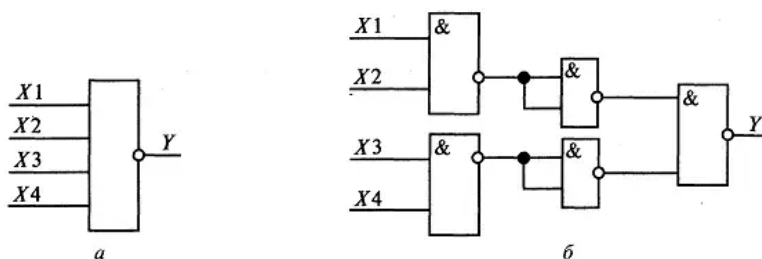


Рис. 8.4. Один из четырехвходовых элементов комбинационной логки (а) и его реализация через двухвходовые элементы (б)

Проверить, что четырехвходовый элемент, изображенный на рис. 8.4 (а), эквивалентен комбинации двухвходовых элементов, изображенной на рис. 8.4 (б).

Содержание отчета

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание и его решение.
4. Вывод по работе.

Вопросы для самоконтроля

1. Назовите элементарные логические элементы и приведите их обозначения на схемах.
2. Изобразите электрические схемы, реализующие элементарные логические элементы.
3. Приведите примеры построения схем на логических элементах на основе логического выражения.
4. Приведите примеры построения логических выражений по заданным логическим схемам.
5. Что такое триггер? Перечислите виды триггеров и коротко их охарактеризуйте.
6. Чем отличается синхронный триггер от несинхронного?
7. Какое состояние триггера называют недопустимым?
8. Расскажите об элементе «Исключающее ИЛИ», приведите таблицу истинности для соответствующей логической операции.
9. Расскажите о полусумматоре.
10. Расскажите о сумматоре и организации переноса «запасного» разряда.

Практическое занятие № 6

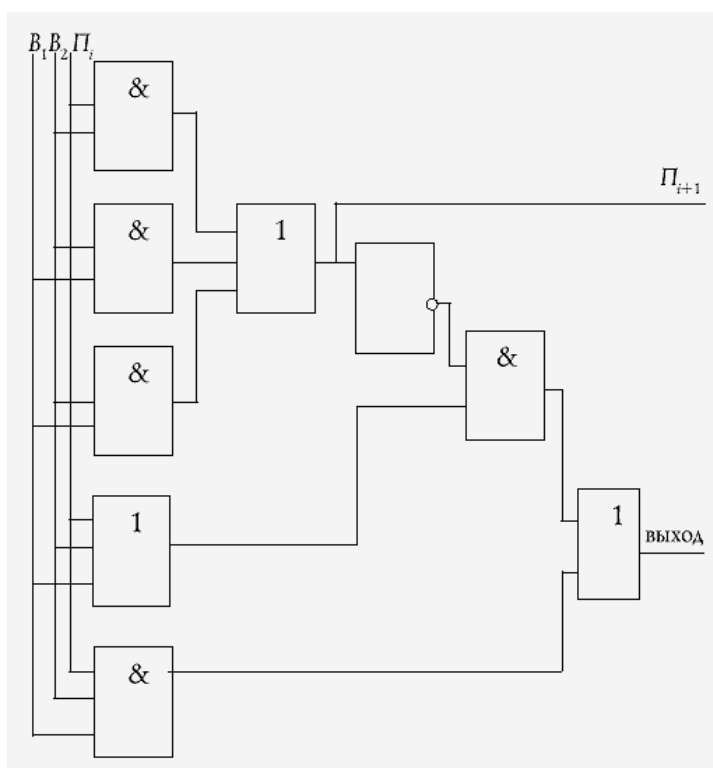
Тема: «Изучение работы сумматоров различных типов»

Цель работы: изучить работу сумматоров различного типа.

Теоретические сведения

Одноразрядный сумматор. Рассмотрим схему работы сумматора, результаты занесем в таблицу.

На схеме использованы следующие обозначения: V_1 (вход 1) — первый операнд (слагаемое); V_2 (вход 2) — второй операнд операции сложения; Π_i — признак переноса 1 из предыдущего разряда; Π_{i+1} — признак, указывающий, будет ли осуществлен перенос в следующий разряд после выполнения операции сложения.



Вход 1	Вход 2	Перенос 1	Перенос 2	Выход
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
1	0	1	1	0
0	1	1	1	0

1	1	1	1	1
---	---	---	---	---

Вопросы для самоконтроля

1. Одноразрядный сумматор на 2 входа. Таблица истинности.
2. Одноразрядный сумматор на 3 входа. Таблица истинности.
3. Сумматор последовательного действия. Принцип работы.

Выполните следующие задания

1. Синтезировать узел, осуществляющий суммирование двух 1-разрядных двоичных чисел на элементах И, ИЛИ, НЕ.
2. По формуле построить таблицу истинности и схему:

$$Y = (x_2 \vee x_4) \wedge (x_1 \vee x_3) \wedge (x_1 \vee x_2 \vee x_4)$$

Практическое занятие №7

Тема: «Изучение работы триггеров и принципы их работы»

Цель работы: изучить разновидности триггеров и принципы их работы.

Теоритические сведения

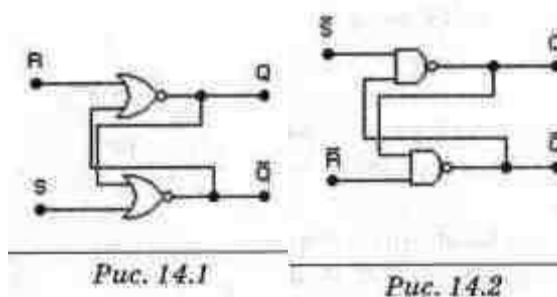
Триггер - простейшая цифровая схема последовательностного типа. У рассмотренных в предыдущих разделах комбинационных схем состояние выхода Y в любой момент времени определяется только текущим состоянием входа X : $Y = \Gamma(X)$. В отличие от них, состояние выхода последовательностной схемы (цифрового автомата) зависит еще и от внутреннего состояния схемы Q : $Y = F(X, Q)$. Другими словами, цифровой автомат является не только преобразователем, но и хранителем предшествующей и источником текущей информации (состояния). Это свойство обеспечивается наличием в схемах обратных связей. Основой последовательностных схем являются триггеры. Триггер имеет два устойчивых состояния: $Q=1$ и $Q=0$, поэтому его иногда называют бистабильной схемой. В каком из этих состояний окажется триггер, зависит от сигналов на входах триггера и от его предыдущего состояния, т. е. он имеет память. Можно сказать, что триггер является элементарной ячейкой памяти. Тип триггера определяется алгоритмом его работы. В зависимости от алгоритма работы, триггер может иметь установочные, информационные и управляющие входы. Установочные входы устанавливают состояние триггера независимо от состояния других входов. Входы управления разрешают запись данных, подающихся на информационные входы. Наиболее распространенными являются триггеры RS, JK, D и T-типов.

Триггер типа RS RS-триггер - простейший автомат с памятью, который может находиться в двух состояниях. Триггер имеет два установочных входа: установки S (set - установка) и сброса R (reset - сброс), на которые подаются входные сигналы от внешних источников. При подаче на вход установки активного логического уровня триггер устанавливается в 1 ($Q = 1$, $\bar{Q} = 0$), при подаче активного уровня на вход сброса триггер устанавливается в 0 ($Q = 0$, $\bar{Q} = 1$). Если подать на оба входа установки (возбуждения) пассивный уровень, то триггер будет сохранять предыдущее состояние выходов: $Q=0$

$$(\bar{Q} = 1) \text{ либо } Q=1 (\bar{Q} = 0).$$

Каждое состояние устойчиво и поддерживается за счет действия обратных связей. Для триггеров этого типа является недопустимой одновременная подача активного уровня на оба входа установки, т.к. триггер по определению не может одновременно быть установлен в ноль и единицу. На практике подача активного уровня на установочные входы приводит к тому, что это состояние не может быть сохранено и невозможно определить, в каком состоянии будет находиться триггер при последующей подаче на установочные входы сигналов пассивного уровня. На рис. 14.1 и 14.2 показаны два вида RS-триггеров,

выполненных на элементах ИЛИ-НЕ и И-НЕ. Для схемы на рис. 14.1 активным уровнем является уровень логической единицы, для схемы на рис. 14.2 - уровень логического нуля. Схема на рис. 14.2 получила название RS-триггера с инверсными входами - RS-триггер. RS-триггер является основным узлом для построения последовательных схем. Название схем такого типа «последовательные» означает, что состояние выхода зависит от того, в какой последовательности на входы подаются входные наборы и каково было предшествующее внутреннее состояние. Так, если в RS-триггере (рис. 14.1) вначале установить комбинацию R=0, S=1 (сокращенная запись - 01), а потом перейти к R=0, S=0 (00), то состояние выхода Q=1. Если же вначале установить комбинацию 10, а потом перейти к 00, то состояние выхода будет другим - Q=0, несмотря на одинаковые комбинации сигналов на входах. Таким образом, при одном и том же входном наборе 00 выход триггера может находиться в разных состояниях. Условия переходов триггеров из одного состояния в другое (алгоритм работы) можно описать табличным, аналитическим или графическим способами. Табличное описание работы RS-триггера (рис. 14.1) представлено в таблице 14.1 (таблица переходов) и таблице 14.2 (таблица функций возбуждения).



R	S	Qt+1
0	0	Qt
0	1	1
1	0	0
1	1	-

Таблица 14.1

Qt	Qt+1	R	S
0	0	X	0
0	1	0	1
1	0	1	0
1	1	0	X

Таблица 14.2

В таблицах использованы следующие обозначения: Qt - предшествующее состояние выхода; Qt+1 - новое состояние, устанавливающееся после перехода (возможно Qt+1= Qt); x - безразличное значение сигнала: 0 или 1; — - неопределенное состояние. Аналитическое описание (характеристическое уравнение) можно получить из таблиц 14.1, 14.2 по правилам алгебры логики:

$$Q_{t+1} = \bar{R}S \vee \bar{R}Q_t = \bar{R}(S \vee Q_t)$$

Работа JK-триггера описывается характеристическим уравнением:

$$Q_{t+1} = J_t \bar{Q}_t \vee Q_t K_t$$

Один из вариантов функциональной схемы JK-триггера со входами установки логическим нулем и его условное графическое обозначение (УГО) приведены на рис. 14.4а,б. Временные диаграммы его работы при R=S=1 приведены на рис. 14.5.

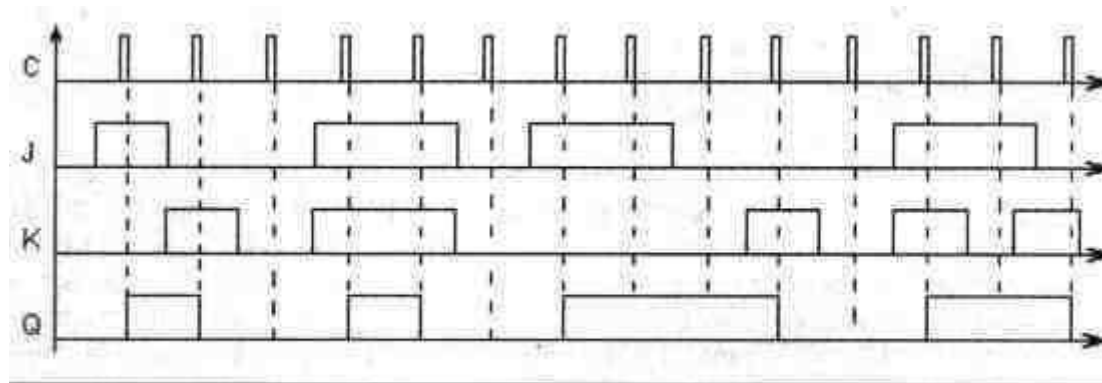


Рис. 14.5

Подобно RS-триггеру, изменение состояния JK-триггера можно изобразить графом переходов (рис. 14.6). Входные сигналы, которые могут принимать любые значения (как 0, так и 1), обозначены как X, а позиция обозначения соответствует последовательности J, K. Этот рисунок не должен вводить в заблуждение: если X=1, то при JK=11 схема будет переходить из состояния Q=0 в состояние Q=1. Но из этого состояния схема должна возвратиться в Q=0 и т. д. Этот граф описывает работу автогенератора. В данном случае все изменения выхода происходят только в момент отрицательного перепада тактового сигнала C. Действительно, если J=K=1, то с каждым новым тактовым импульсом выход будет изменять свое значение на противоположное и триггер будет выполнять функцию делителя частоты на 2, а не автогенератора. 3. D-триггер. D-триггер имеет один информационный вход D (data - данные). Информация со входа D заносится в триггер по положительному перепаду импульса на счетном входе C и сохраняется до следующего положительного перепада на счетном входе триггера.

Функциональная схема D-триггера может быть получена из схемы JK-триггера (рис.14.5d) путем подключения входа K ко входу J через инвертор: D=J=K. 4. T-триггер (счетный триггер) На основе JK-триггеров и D-триггеров можно построить схемы, осуществляющие так называемый счетный режим. Такие схемы называют T-триггерами или счетными триггерами, связывая с этим способ их функционирования. На рис. 14.9 представлены схемы

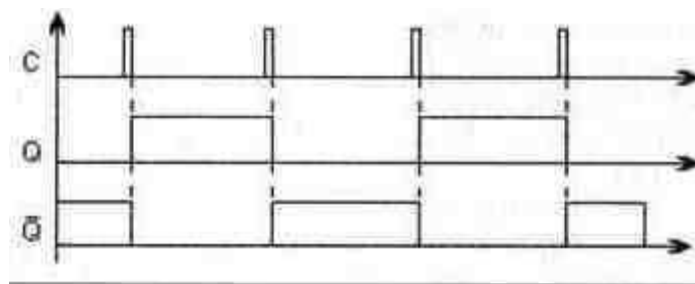


Рис. 14.10

организации Т-триггера на основе JK и D-триггеров. Счетный режим иллюстрируется временными диаграммами рис. 14.10. В JK-триггере со входами установки логическим нулем счетный режим реализуется путем подачи констант $J=K=1$ и $R=S=1$ и входного сигнала Т на вход С. В соответствии с таблицей функционирования при каждом отрицательном перепаде входного сигнала Т состояние триггера изменяет свое значение на противоположное. В D-триггере счетный режим реализуется при помощи обратной связи (на вход D подается сигнал с инверсного выхода). Таким образом, всегда существует неравенство сигнала на входе D и сигнала на выходе Q: если $Q=1$, $D=0$. Следовательно, при каждом положительном перепаде сигнала на счетном входе С, в соответствии с принципом действия D-триггера состояние выхода будет изменяться на противоположное. Таким образом, на каждые два входных тактовых импульса Т-триггер формирует один период выходного сигнала Q. Следовательно, триггер осуществляет деление частоты f_T на его входе на 2:

$$f_Q = f_T / 2$$

Контрольные вопросы

1. Чем отличается работа RS-триггера с прямыми входами от работы RS-триггера с инверсными входами?
2. Почему комбинация сигналов 11 на входах RS-триггера называется «запрещенной»?
3. В чём отличие таблицы переходов триггера от таблицы функций возбуждения?
4. Как свойство запоминания отражается в характеристических уравнениях триггеров?
5. В чём принципиальное отличие работы синхронных триггеров от асинхронных?
6. Какова приоритетность информационных и установочных входов в синхронных триггерах?

Практическая работа № 8-9.

Тема: «Знакомство с моделью учебной ЭВМ»

Цель работы:

- Знакомство с архитектурой модели учебной ЭВМ.
- Изучение системы команд модельной ЭВМ, действиями основных классов команд и способов адресации.

Теоретические сведения:

Программная модель учебной ЭВМ представлена одним exe-файлом, не требует инсталляции, может размещаться в любом каталоге и запускается обычным образом.

При работе программы формируется текстовый log-файл, в который помещается информация о процессе функционирования модели. Текст этого файла выводится в основное окно модели «**Модель учебной ЭВМ**», а при закрытии программы сохраняется по умолчанию в текущем каталоге или записывается по пути, указанному в диалоговом окне, вызываемом через последовательность: **Работа | Настройки | Параметры системы**.

В папке размещены два исполняемых файла:

- CompModel.exe, объемом 496 Кбайт;
- CompModel-2К.exe, объемом 868 Кбайт.

Основным файлом является CompModel.exe, который может работать на любой ПЭВМ с операционной системой WindowsXP.

Для работы с предыдущими версиями ОС Windows(Windows2000 иWindows 98) можно использовать файлCompModel-2К.exe, обладающий несколько меньшими функциональными возможностями. Эта программа реализует ту же модель ЭВМ, что иCompModel, кроме возможности подключения модели кэш-памяти.

При желании можно запустить программу CompModel.exe под Windows 2000 или Windows 98 — функционировать модель будет правильно (в том числе и кэш-память), но возникнут проблемы с сохранением созданного текста из окна **Текстпрограммы**.

Текстовые файлы, созданные в других редакторах, загружаются в это окно без проблем. Для сохранения написанного (отредактированного) в окне **Текст программы** файла можно выделить весь текст (нажать комбинацию клавиш <Ctrl>+<A>), скопировать его в буфер обмена (комбинация клавиш <Ctrl>+<C>) и вставить в окно любого предварительно открытого текстового редактора (комбинация клавиш <Ctrl>+<V>), после чего сохранить файл средствами этого редактора.

Можно обойтись и без внешнего редактора, если *после успешного ассемблирования* сохранить дизассемблированный текст из окна **Программа**. Однако при таком способе сохранения текста программы имена меток (если они существуют в программе) будут заменены на их числовые значения.

Современные процессоры и операционные системы — не слишком благоприятная среда для начального этапа изучения архитектуры ЭВМ.

Одним из решений этой проблемы может быть создание программных моделей учебных ЭВМ, которые, с одной стороны, достаточно просты, чтобы обучаемый мог освоить базовые понятия архитектуры (система команд, командный цикл, способы адресации, уровни памяти, способы взаимодействия процессора с памятью и внешними устройствами), с другой стороны — архитектурные особенности модели должны соответствовать тенденциям развития современных ЭВМ.

Программная модель позволяет реализовать доступ к различным элементам ЭВМ, обеспечивая удобство и наглядность. С другой стороны, модель позволяет игнорировать те особенности работы реальной ЭВМ, которые на данном уровне рассмотрения не являются существенными.

Моделируемая ЭВМ включает:

- процессор;
- блок регистров общего назначения;
- оперативную память (ОЗУ);
- сверхоперативную память (СОЗУ) (кэш-память);
- устройства ввода (УВв);
- устройства вывода (УВыв).

Процессор, в свою очередь, состоит из:

- центрального устройства управления (УУ);
- арифметического устройства (АУ);
- блока системных регистров (CR, PC, SP и др.)

Структурная схема ЭВМ показана на рисунке 1.

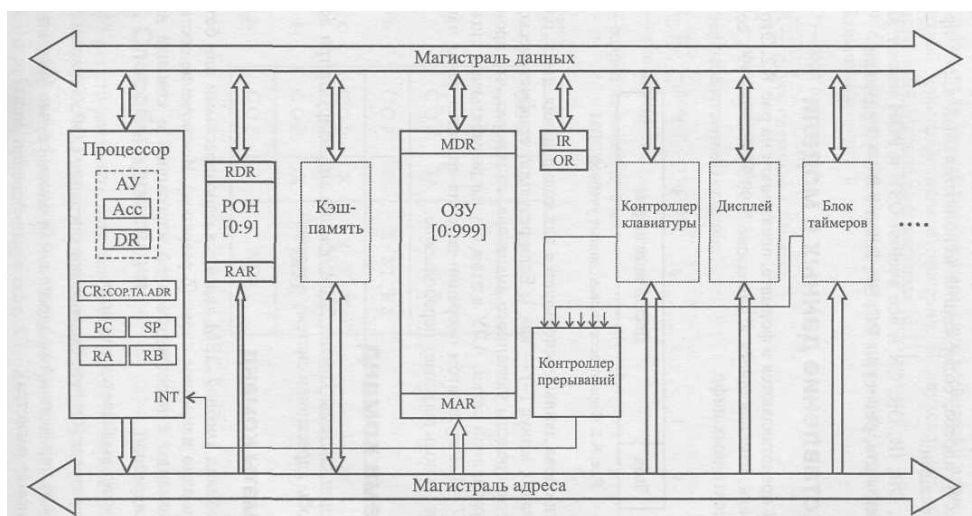


Рис. 1 -Общая структура учебной ЭВМ

В ячейках ОЗУ хранятся команды и данные. Емкость ОЗУ составляет 1000 ячеек. По сигналу **MWr** выполняется запись содержимого регистра данных (MDR) в ячейку памяти с адресом, указанным в регистре адреса (MAR). По сигналу **MRd** происходит считывание — содержимое ячейки памяти с адресом, содержащимся в MAR, передается в MDR.

Блок регистров общего назначения (РОН) с прямой адресацией содержит десять регистров общего назначения R0—R9. Доступ к ним осуществляется (аналогично доступу к ОЗУ) через регистры RAR и RDR.

АУ осуществляет выполнение одной из арифметических операций, определяемой кодом операции (COP), над содержимым аккумулятора (Acc) и регистра операнда (DR). Результат операции всегда помещается в Acc. При завершении выполнения операции АУ вырабатывает сигналы признаков результата:

- Z (равен 1, если результат равен нулю);
- S (равен 1, если результат отрицателен);
- OV (равен 1, если при выполнении операции произошло переполнение разрядной сетки).

В случаях, когда эти условия не выполняются, соответствующие сигналы имеют нулевое значение.

В модели ЭВМ предусмотрены внешние устройства двух типов.

Во-первых, это регистры IRиOR, которые могут обмениваться с аккумулятором с помощью безадресных команд in (Acc :=IR) и out (OR:= Acc).

Во-вторых, это набор моделей внешних устройств, которые могут подключаться к системе и взаимодействовать с ней в соответствии с заложенными в моделях алгоритмами. Каждое внешнее устройство имеет ряд программно-доступных регистров, может иметь собственный *обозреватель* (окно видимых элементов).

Подробнее эти внешние устройства описаны в разделе описания программы моделирующей работу ЭВМ.

УУ осуществляет выборку команд из ОЗУ в последовательности, определяемой естественным порядком выполнения команд (т. е. в порядке возрастания адресов команд в ОЗУ) или командами передачи управления; выборку из ОЗУ операндов, задаваемых адресами команды; инициирование выполнения операции, предписанной командой; останов или переход к выполнению следующей команды.

В качестве сверхоперативной памяти может подключаться модель кэш-памяти.

В состав УУ ЭВМ входят:

- PC— счетчик адреса команды, содержащий адрес текущей команды;
- CR— регистр команды, содержащий код команды;

- RB— регистр базового адреса, содержащий базовый адрес;
- SP— указатель стека, содержащий адрес верхушки стека;
- RA— регистр адреса, содержащий исполнительный адрес при косвенной адресации.

Регистры данных, такие как : - Acc,DR,IR,OR,CRи все ячейки ОЗУ и РОИ имеют длину 6 десятичных разрядов, а регистры команд: -PC,SP,RAиRB— 3 разряда.

Представление данных в модели

Данные в ЭВМ представляются в формате, показанном на рис. 2. Это целые десятичные числа, изменяющиеся в диапазоне "-99 999... +99 999", содержащие знак и 5 десятичных цифр.

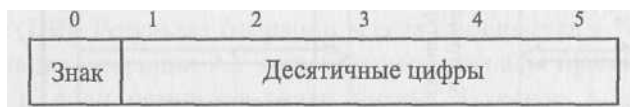


Рис. 2. Формат десятичных данных учебной ЭВМ

Старший разряд слова данных используется для кодирования знака: плюс (+) изображается как 0, минус (-) — как 1. Если результат арифметической операции выходит за пределы указанного диапазона, то говорят, что произошло переполнение разрядной сетки. АЛУ в этом случае вырабатывает сигнал переполнения $OV= 1$. Результатом операции деления является целая часть частного. Деление на ноль вызывает переполнение.

Система команд

При рассмотрении системы команд ЭВМ обычно анализируют три аспекта: форматы, способы адресации и систему операций.

Форматы команд

Большинство команд учебной ЭВМ являются одноадресными или безадресными, длиной в одно машинное слово (6 разрядов). Исключение составляют двухсловные команды с непосредственной адресацией и команда mov, являющаяся двухадресной.

В форматах команд выделяется три поля:

- два старших разряда [0:1] определяют код операции COP;
- разряд 2 может определять тип адресации (в одном случае (формат 5a) он определяет номер регистра);
- разряды [3:5] могут определять прямой или косвенный адрес памяти, номер регистра (в команде mov номера двух регистров), адрес перехода или короткий непосредственный операнд. В двухсловных командах непосредственный операнд занимает поле [6:11].

Полный список форматов команд показан на рис. 3, где приняты следующие обозначения:

- COP — код операции;
- ADR— адрес операнда в памяти;
- ADC— адрес перехода;
- I— непосредственный операнд;
- R,R1,R2 — номер регистра; П ТА — тип адресации;
- X— разряд не используется.

Номер формата	0	1	2	3	4	5
1	COP	X	X	X	X	
2	COP	TA	ADR			
3	COP	TA	X	X	R	
3a	COP	TA	X	R1	R2	
4	COP	X	X	X	X	I
5	COP	X	ADC			
5a	COP	R	ADC			

Рис. 3. Форматы команд учебной ЭВМ

Способы адресации

В ЭВМ принято различать пять основных способов адресации:

- прямая,
- косвенная,
- непосредственная,
- относительная,
- безадресная.

Каждый способ адресации имеет разновидности. В модели учебной ЭВМ реализованы семь способов адресации, приведенные в табл. 1.

Таблица 1. Адресация в командах учебной ЭВМ

Код ТА	Тип адресации	Исполнительный адрес
0	Прямая (регистровая)	ADR (R)
1	Непосредственная	—
2	Косвенная	ОЗУ(ADR)[3:5]
3	Относительная	ADR + RB
4	Косвенно-регистровая	РОН(R)[3:5]
5	Индексная с постинкрементом	РОН(R)[3:5], R:=R + 1
6	Индексная с преддекрементом	R:=R-1,РОН(R)[3:5]

Система операций

Система команд учебной ЭВМ включает команды следующих классов:

- *арифметико-логические и специальные:*

сложение,

вычитание,

умножение,

деление;

- *пересылки и загрузки:*

чтение,

запись,

пересылка (из регистра в регистр),

помещение в стек,

извлечение из стека,

загрузка указателя стека,

загрузка базового регистра;

- *ввода/вывода:*

ВВОД,

ВЫВОД;

- *передачи управления:*

безусловный переход,

шесть команд условного перехода,

вызов подпрограммы,

возврат из подпрограммы,

цикл,

программное прерывание,

возврат из прерывания;

системные:

пустая операция,

разрешить прерывание,

запретить прерывание,

стоп.

Список команд учебной ЭВМ приведен в табл.

Таблица 3. Система команд учебной ЭВМ

КОП	Мнемо	Название	Действие
00	NOP	Пустая операция	Нет
01	IN	Ввод	Acc ← IR
02	OUT	Вывод	OR ← Acc
03	IRET	Возврат из прерывания	FLAGS.PC ← M(SP); INC(SP)
04	WRRB	Загрузка RB	RB ← CR[ADR]
05	WRSP	Загрузка SP	SP ← CR[ADR]
06	PUSH	Поместить в стек	DEC(SP); M(SP) ← R
07	POP	Извлечь из стека	R → M(SP); INC(SP)
08	RET	Возврат	PC → M(SP); INC(SP)
09	HLT	Стоп	Конец командных циклов
10	JMP	Безусловный переход	PC ← CR[ADR]
11	JZ	Переход, если 0	if Acc = 0 then PC ← CR[ADR]
12	JNZ	Переход, если не 0	if Acc ≠ 0 then PC ← CR[ADR]
13	JS	Переход, если отрицательно	if Acc < 0 then PC ← CR[ADR]
14	JNS	Переход, если положительно	if Acc ≥ 0 then PC ← CR[ADR]
15	JO	Переход, если переполнение	if Acc > 99999 then PC ← CR[ADR]
16	JNO	Переход, если нет переполнения	if Acc ≤ 99999 then PC ← CR[ADR]
17	JRNZ	Цикл	DEC(R); if R > 0 then PC ← CR[ADR]
18	INT	Программное прерывание	DEC(SP); M(SP) ← FLAGS.PC; PC ← M(V)
19	CALL	Вызов подпрограммы	DEC(SP); M(SP) ← PC; PC ← CR(ADR)
20	Нет		
21	RD	Чтение	Acc ← DD
22	WR	Запись	M(*) ← Acc
23	ADD	Сложение	Acc ← Acc + DD
24	SUB	Вычитание	Acc ← Acc - DD
25	MUL	Умножение	Acc ← Acc x DD
26	DIV	Деление	Acc ← Acc/DD
27	Нет		
28	EI	Разрешить прерывание	IF ← 1
29	DI	Запретить прерывание	IF ← 0
30	MOV	Пересылка	R1 ← R2
31	RD	Чтение	Acc ← R*
32	WR	Запись	R* ← Acc
33	ADD	Сложение	Acc ← Acc + R*
34	SUB	Вычитание	Acc ← Acc - R*
35	MUL	Умножение	Acc ← Acc x R*
36	DIV	Деление	Acc ← Acc/R*

37	IN	Ввод	Acc ← ВУ(CR[ADR*])
38	OUT	Вывод	ВУ(CR[ADR*]) ← Acc
39	Нет		
40	Нет		
41	RDI	Чтение	Acc←1
42	Нет		
43	ADI	Сложение	Acc ← Acc + I
44	SBI	Вычитание	Acc ← Acc - I
45	MULI	Умножение	Acc ← Acc x I
46	DIVI	Деление	Acc←Acc/I

Состояния и режимы работы ЭВМ

Ядром УУ ЭВМ является управляющий автомат (УА), вырабатывающий сигналы управления, которые инициируют работу АЛУ, РОН, ОЗУ и УВВ, передачу информации между регистрами устройств ЭВМ и действия над содержимым регистров УУ.

ЭВМ может находиться в одном из двух состояний: **Останов** и **Работа**.

В состояние **Работа** ЭВМ переходит по действию команд **Пуск** или **Шаг**. Команда **Пуск** запускает выполнение программы, представляющую собой последовательность команд, записанных в ОЗУ, в автоматическом режиме до команды **hlt** или точки останова. Программа выполняется по командам, начиная с ячейки ОЗУ, на которую указывает РС, причем изменение состояний объектов модели отображается в окнах обозревателей.

В состоянии **Останов** ЭВМ переходит по действию команды **Стоп** или автоматически в зависимости от установленного режима работы.

Команда **Шаг**, в зависимости от установленного режима работы, запускает выполнение одной команды или одной микрокоманды (если установлен **Режим микрокоманд**), после чего переходит в состояние **Останов**.

В состоянии **Останов** допускается просмотр и модификация объектов модели: регистров процессора и РОН, ячеек ОЗУ, устройств ввода/вывода. В процессе модификации ячеек ОЗУ и РОН можно вводить данные для программы, в ячейки ОЗУ — программу в кодах. Кроме того, в режиме **Останов** можно менять параметры модели и режимы ее работы, вводить и/или редактировать программу в мнемосокодах, ассемблировать мнемосокоды, выполнять стандартные операции с файлами.

Практическая работа № 10

Тема: «Регистры процессора. Память.»

Цель работы: Приобретение навыков работы с регистрами процессора и памятью.

Порядок выполнения работы

1. Прочитать задания к работе
2. Оформите отчет, который должен содержать:
 - титульный лист (см. приложение);
 - постановку задачи;
 - описание пошагового исполнения;
 - отчет о полученном результате

Задания к работе.

Для решения с помощью ЭВМ некоторой задачи должна быть разработана программа. Программа на языке ЭВМ представляет собой последовательность команд. Код каждой команды определяет выполняемую операцию, тип

адресации и адрес. Выполнение программы, записанной в памяти ЭВМ, осуществляется последовательно по командам в порядке возрастания адресов команд или в порядке, определяемом командами передачи управления.

Для того чтобы получить результат выполнения программы, пользователь должен:

- ввести программу в память ЭВМ;
- определить, если это необходимо, содержимое ячеек ОЗУ и РОИ, содержащих исходные данные, а также регистров IR и BR;
- установить в РС стартовый адрес программы;
- перевести модель в режим **Работа**.

Каждое из этих действий выполняется посредством интерфейса модели, описанного в *главе 8*. Ввод программы может осуществляться как в машинных кодах непосредственно в память модели, так и в мнемокодах в окне **Текст программы** с последующим ассемблированием.

Цель настоящей лабораторной работы — знакомство с интерфейсом модели ЭВМ, методами ввода и отладки программы, действиями основных классов команд и способов адресации. Для этого необходимо ввести в память ЭВМ и выполнить в режиме **Шаг** некоторую последовательность команд (определенную вариантом задания) и зафиксировать все изменения на уровне программно-доступных объектов ЭВМ, происходящие при выполнении этих команд.

Команды в память учебной ЭВМ вводятся в виде шестизначных десятичных чисел (см. форматы команд на рис. 8.3, коды команд и способов адресации в табл. 8.2—8.4).

В настоящей лабораторной работе будем программировать ЭВМ в машинных кодах.

9.1.2. Пример 1

Дана последовательность мнемочкодов, которую необходимо преобразовать в машинные коды, занести в ОЗУ ЭВМ, выполнить в режиме **Шаг** и зафиксировать изменение состояний программно-доступных объектов ЭВМ (табл. 9.1).

Таблица 9.1. Команды и коды

Последовательность	Значения					
	Команды	RD #20	WR 30	DD 5	WR @30	NZ 002
Коды	21 1 020	22 0 030	3 1 05	22 2 030	12 0002	

Введем полученные коды последовательно в ячейки ОЗУ, начиная с адреса 000. Выполняя команды в режиме **Шаг**, будем фиксировать изменения программно-доступных объектов (в данном случае это Асс, РС и ячейки ОЗУ 020 и 030) в табл. 9.2.

Таблица 9.2. Содержимое регистров

РС	Асс	М(30)	М(20)	РС	Асс	М(30)	М(20)
000	000000	000000	000000	004			000025
001	000020			002			
002		000020		003	000030		
003	000025			004			000030

9.1.3. Задание 1

1. Ознакомиться с архитектурой ЭВМ
2. Записать в ОЗУ "программу", состоящую из пяти команд— варианты задания выбрать из табл.. Команды разместить в последовательных ячейках памяти.
3. При необходимости установить начальное значение в устройство ввода IR.
4. Определить те программно-доступные объекты ЭВМ, которые будут изменяться при выполнении этих команд.
5. Выполнить в режиме **Шаг** введенную последовательность команд, фиксируя изменения значений объектов, определенных в п. 4, в таблице (см. форму табл. 9.2).
6. Если в программе образуется цикл, необходимо просмотреть не более двух

повторений каждой команды, входящей в тело цикла.

Таблица 9.3. Варианты задания 1

№	IR	Команда 1	Команда 2	Команда 3	Команда 4	Команда S
1	000007	IN	MUL #2	WR10	WR @10	JNS 001
2	X	RD #17	SUB #9	WR16	WR @16	JNS 001
3	100029	IN	ADD #16	WR8	WR @8	JS 001
4	X	RD #2	MUL #6	WR 11	WR @11	JNZ 00
5	000016	IN	WR8	DIV #14	WR @8	JMP 002
6	X	RD #4	WR 11	RD @11	ADD #330	JS 000
7	000000	IN	WR9	RD @9	SUB#1	JS 001
8	X	RD 4	SUB #8	WR8	WR @8	JNZ 001
9	100005	IN	ADD #12	WR 10	WR @10	JS 004
10	X	RD 4	ADD #15	WR 13	WR @13	JMP 001
11	000315	IN	SUB #308	WR11	WR @11	JMP 001
12	X	RD #988	ADD #19	WR9	WR @9	JNZ 001
13	000017	IN	WR11	ADD 11	WR @11	JMP 002
14	X	RD #5	MUL #9	WR10	WR @10	JNZ 001

Содержание отчета

1. Формулировка варианта задания.
2. Машинные коды команд, соответствующих варианту задания.
3. Результаты выполнения последовательности команд в форме табл. 9.2.

Контрольные вопросы

1. Из каких основных частей состоит ЭВМ и какие из них представлены в модели?
2. Что такое система команд ЭВМ?
3. Какие классы команд представлены в модели?
4. Какие действия выполняют команды передачи управления?
5. Какие способы адресации использованы в модели ЭВМ? В чем отличие между ними?
6. Какие ограничения накладываются на способ представления данных модели ЭВМ?
7. Какие режимы работы предусмотрены в модели и в чем отличие между ними?
8. Как записать программу в машинных кодах в память модели ЭВМ.
9. Как просмотреть содержимое регистров процессора и изменить содержимое

некоторых регистров?

10. Как просмотреть и, при необходимости, отредактировать ячейки памяти?

11. Как запустить выполнение программы в режиме приостановки работы после выполнения каждой команды?

12. Какие способы адресации операндов применяются в командах ЭВМ?

13. Какие команды относятся к классу передачи управления?

Практическое занятие № 11

Тема: «Подключение сетевых устройств, установка и настройка оборудования, работа с вычислительной сетью»

Цель работы: изучить основные блоки и периферийные устройства персонального компьютера, способы их соединения, разъемы, основные характеристики (название, тип разъема, количество контактов, скорость передачи данных, дополнительные свойства); научиться определять по внешнему виду типы разъемов, подключаемое к ним оборудование, знать основные устройства персонального компьютера, их назначение и основные характеристики; научиться определять компоненты системного блока по внешнему виду, уяснить порядок и способы их соединения.

Оборудование: системный блок, кабели в комплекте, монитор, клавиатура, периферийные устройства для различных разъемов (мышь, принтер, модем и др.), системный блок в сборе, макеты видеоадаптера, материнской платы, корпуса, жесткого диска, накопителя на флоппи-дисках, интерфейсные кабели.

Теоретические сведения

В основу устройства компьютера положен **принцип открытой архитектуры**, т.е. возможность подключения к системе дополнительных независимо разработанных устройств для различных прикладных применений. Все устройства подключаются к системе и взаимодействуют друг с другом через **общую шину**.

В базовую конфигурацию ПК входят: системный блок, монитор, клавиатура и ручной манипулятор - мышь.

Системный блок. Системный блок является центральной частью ПК. В корпусе системного блока размещены внутренние устройства ПК.

Основой корпуса (рис. 1.4) системного блока является каркас (1), к которому крепятся: блок питания (2), панель крепления материнской платы (3), передняя панель (4), а также секции для дисководов размером 5,25- (5) и 3,5- (6). Оба типа секций можно использовать для накопителей на жестких магнитных дисках.

В состав системного блока входят следующие аппаратные средства ПК:

1. Системная (материнская) плата с микропроцессором.
2. Оперативная память.
3. Накопитель на жестком магнитном диске.
4. Контроллеры или адаптеры для подключения и управления внешними устройствами ПК (монитор, звуковые колонки и др.).
5. Порты для подключения внешних устройств (принтер, мышь и др.).

6. Внешние запоминающие устройства для гибких магнитных дисков и лазерных дисков CD и DVD.

На системной плате помимо процессора расположены (1.1):

1. **Чипсет** (микропроцессорный комплект) - набор микросхем, которые управляют работой внутренних устройств ПК и определяют основные функциональные возможности материнской платы.

2. **Шины** - набор проводников, по которым происходит обмен сигналами между внутренними устройствами компьютера.

3. **Оперативная память** - набор микросхем, предназначенных для временного сохранения данных, пока включен компьютер.

4. **Постоянное запоминающее устройство** - микросхема, предназначенная для долговременного хранения данных, даже при отключенном компьютере.

5. **Разъемы (слоты)** для подсоединения дополнительных устройств.

Основные элементы системной платы показаны на рис 1.1, где цифрами обозначены:

1. Разъем для микропроцессора.
2. Слоты для модулей оперативной памяти.
3. Интерфейсы шины PCI.
4. Микросхема системной логики (чипсет, 4.1 - северный мост, а 4.2 - южный мост).
5. Интерфейсы для подключения жестких дисков.
6. Блок портов ввода/вывода.
7. Интерфейс шины AGP для подключения видеоадаптера.

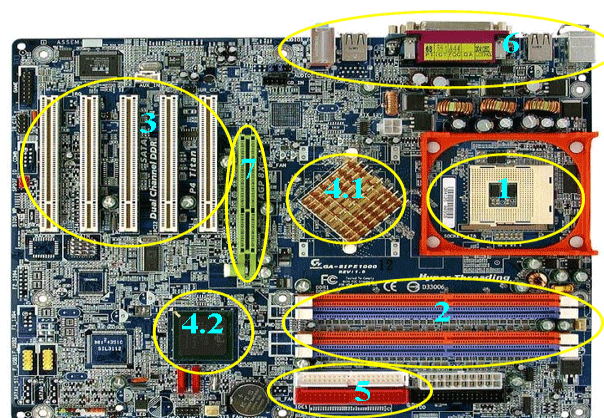


Рисунок 1.1.

Интерфейсы ПК.

Рассмотрим набор и внешний вид интерфейсов, размещенных на задней стенке системного блока (рис 1.2). Все эти интерфейсы предназначены для подключения периферийных устройств к персональному компьютеру.

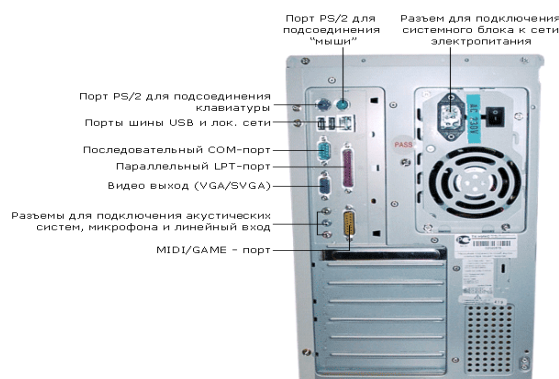
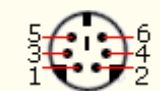



Рисунок 1.2

Порт PS/2 - шестиконтактный разъем, используемый для подключения клавиатуры и ручного манипулятора. Эти разъемы подключены к единому контроллеру.

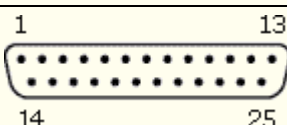
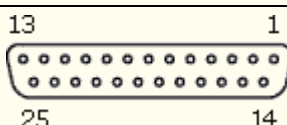
Вилка (устанавливается на кабеле)	Розетка (устанавливается на корпусе системного блока)
	

Последовательный COM-порт (RS-232) - данный порт используется для подключения модема. Ранее использовался и для подключения ручного манипулятора ("мыши"). Порт стандартизирован в двух вариантах 9 (DB9) и 25-контактный (DB25). Последний вариант практически не реализуется в современных системных блоках. Для асинхронного режима принято несколько стандартных скоростей обмена: 9600, 19200, 38400, 57600 и 115200 бит/с.

Вилка (устанавливается на корпусе системного блока)	Розетка (устанавливается на кабеле)
	
	

Параллельный порт (LPT) - этот порт изначально разрабатывался как интерфейс для подключения принтера (сканера, плоттера). Скорость обмена не выше 150 Кбайт/с при значительной нагрузке процессора. В 1994 г. был принят стандарт IEEE1284, определивший спецификацию портов SPP, EPP и ECP. Дополнительные режимы EPP (Enhanced Parallel Port - улучшенный параллельный порт) и ECP (Extended Capability Port - порт с расширенными возможностями) позволили ввести поддержку

двухстороннего обмена с аппаратным сжатием данных (устанавливается программой Setup BIOS). В качестве разъемов спецификацией определены Тип А (DB-25), Тип В (Centronics) и тип С (компактный 36-контактный).

Вилка (устанавливается на кабеле)	Розетка (устанавливается на корпусе системного блока)
	

Видеовыход (15-контактный разъем) - используется для подключения VGA/SVGA монитора к системному блоку, а именно, к видеоадаптеру. В случае интегрированного в

системную плату видеоадаптера видеовыход размещается на стандартной панели, как показано на рис 1.2.

Разъем для подключения к локальной сети (RJ-45) - восьмиконтактный интерфейс для подключения компьютера к локальной сети. В случае интегрированного в системную плату сетевого адаптера интерфейс RJ-45 размещается на стандартной панели интерфейсов (как на рис 1.2). Другой вариант - размещается на установленном сетевом адаптере.

MIDI/GAME порт - используется для подключения мультимедийных игровых устройств, например, синтезатора и игрового манипулятора "джойстика".

В современных ПК к таким внешним шинам и интерфейсам относятся: USB, FireWire, IrDA, Bluetooth. Последние два интерфейса относятся к классу беспроводных интерфейсов.

Шина и интерфейс USB. периферийные устройства с интерфейсом Архитектура шины USB представляет собой USB. Периферийные концентраторы могут классическую топологию "звезда" сподключаться друг к другу, образуя каскады. последовательной передачей данных, в соответствии с которой в системе должен быть корневой (ведущий) концентратор USB, к которому подключаются периферийные концентраторы USB (рис 1.3, внешний концентратор на 4 порта USB), а непосредственно к ним подключаются



Рисунок 1.3.

Корневой концентратор расположен в одной из микросхем системной логики (как правило, это южный мост чипсета).).

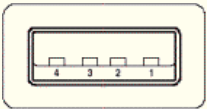
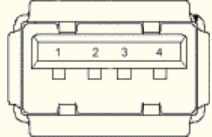
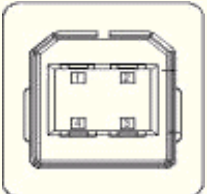
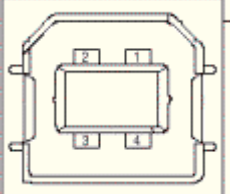
Проблема низкой пропускной способности частично решена версией интерфейса USB 2.0, в соответствии с которой пиковая пропускная способность увеличена до 480 Мбит/с (60 Мбайт/с). Этого вполне достаточно для работы типичных современных USB-устройств. Все устройства USB соединяются между собой четырехжильным кабелем (рис 1.4).



Рисунок 1.4.

По одной паре передаются данные, по другой - электропитание, которое автоматически подключается устройством при необходимости. На концах кабеля

монтируются разъемы типов "А" и "В". С помощью разъема "А" устройство подключают к концентратору. Разъем типа "В" устанавливают на концентраторы для связи с другим концентратором и на устройства, от которых кабель должен отключаться (например, сканеры).

<p>Вилка типа "А" (устанавливается на кабеле)</p>	<p>Розетка типа "А" (устанавливается на корпусе системного блока)</p>
	
<p>Вилка типа "В" (устанавливается на кабеле)</p>	<p>Розетка типа "В" (устанавливается на корпусе периферийного устройства)</p>
	

Интерфейс IEEE1394 (FireWire). Конкурентом интерфейса USB 2.0 на сегодняшний день является последовательный цифровой интерфейс FireWire, называемый также IEEE1394 (iLink - торговая марка Sony).

Подключение осуществляется с помощью стандартной пары "вилка - розетка" (рис 1.5, рис 1.6). Корневое устройство интерфейса выполняет функции управления шиной. Первоначально такие устройства разрабатывались в виде плат расширения (рис 1.5), в дальнейшем поддержка IEEE1394 стала реализовываться в наборе системной логики (чипсете) системной платы.



Рисунок 1.10.

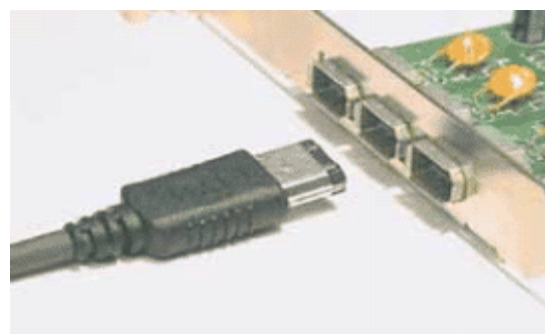


Рисунок 1.11.

Автоматическая конфигурация интерфейса IEEE1394 происходит после включения питания, отсоединения или подключения устройства. При изменении конфигурации подается сигнал сброса и производится новая идентификация дерева.

Как и USB, шина IEEE 1394 обеспечивает возможность переконфигурации аппаратных средств компьютера без его выключения. В соответствии с принятым стандартом IEEE1394 существует два варианта разъемов и кабелей (рис 1.7).

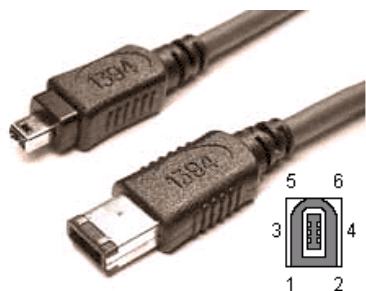


Рисунок 1.7.

Интерфейс Bluetooth относится к перспективным беспроводным интерфейсам передачи данных. Этот интерфейс активно разрабатывается и продвигается консорциумом Bluetooth Special Interest Group (Bluetooth SIG).

Технология Bluetooth разрабатывалась для построения беспроводных персональных сетей (WPAN, Wireless Personal Area Network). В 2001 году был принят стандарт IEEE 802.15.1, описывающий технологию построения таких сетей, а в 2002 году технология получила развитие в стандарте IEEE 802.15.3 (протокол связи для беспроводных частных сетей).

Задания:

1. Убедитесь в том, что компьютерная система обесточена (при необходимости, отключите систему от сети).
2. Разверните системный блок задней стенкой к себе.
3. По наличию или отсутствию разъемов USB установите форм-фактор материнской платы (при наличии разъемов USB - форм-фактор ATX, при их отсутствии - AT).
4. Установите местоположение и снимите характеристики следующих разъемов:
 - питания системного блока;
 - питания монитора;
 - сигнального кабеля монитора;
 - клавиатуры;
 - последовательных портов (два разъема);
 - параллельного порта;
 - других разъемов.

5. Убедитесь в том, что все разъемы, выведенные на заднюю стенку системного блока, не взаимозаменяемы, то есть каждое базовое устройство подключается одним единственным способом.

6. Изучите способ подключения мыши.

Мышь может подключаться к разьему последовательного порта или к специальному порту PS/2, имеющему разъем круглой формы. Последний способ является более современным и удобным. В этом случае мышь имеет собственный выделенный порт, что исключает возможность ее конфликта с другими устройствами, подключаемыми к последовательным портам. Последние модели могут подключаться к клавиатуре через разъем интерфейса USB.

7. Заполните таблицу:

Разъем	Тип разъема	Количество контактов	Примечания

8. Определить наличие основных устройств персонального компьютера.

9. Установите местоположение блока питания, выясните мощность блока питания (указана на ярлыке).

10. Установите местоположение материнской платы.

11. Установите характер подключения материнской платы к блоку питания.

Для материнских плат в форм-факторе AT подключение питания выполняется двумя разъемами. Обратите внимание на расположение проводников черного цвета - оно важно для правильной стыковки разъемов.

12. Установите местоположение жесткого диска.

Установите местоположение его разъема питания. Проследите направление шлейфа проводников, связывающего жесткий диск с материнской платой. Обратите внимание на местоположение проводника, окрашенного в красный цвет (на жестком диске он должен быть расположен рядом с разъемом питания).

13. Установите местоположения дисководов гибких дисков и дисковода CD-ROM.

Проследите направление их шлейфов проводников и обратите внимание на положение проводника, окрашенного в красный цвет, относительно разъема питания.

14. Установите местоположение платы видеоадаптера.

Определите тип интерфейса платы видеоадаптера.

15. При наличии прочих дополнительных устройств выявите их назначение, опишите характерные особенности данных устройств (типы разъемов, тип интерфейса и др.).

16. Заполните таблицу:

Устройство	Характерные особенности	Куда и при помощи чего подключается

Подготовьте отчет по лабораторной работе.

Отчет должен содержать:

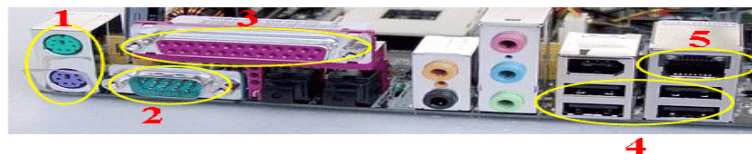
- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- письменные ответы на контрольные вопросы;
- вывод о проделанной работе.

Контрольные вопросы

1. Какие устройства входят в базовую конфигурацию ПК?
2. Назначение, основные характеристики, интерфейс устройств персонального компьютера (по каждому устройству), входящих в состав системного блока.
3. Назовите основные устройства жесткого диска SSD.
4. Перечислите состав базовой аппаратной конфигурации.
5. Укажите основные характеристики монитора.
6. Характеристики (тип разъема, количество контактов, скорость передачи данных) разъемов: видеоадаптера; последовательных портов; параллельного порта; шины USB; сетевой карты; питания системного блока; питания монитора.
7. Назовите типы периферийных устройств.
8. Что понимается под интерфейсом передачи данных?
9. К каким интерфейсам ПК относятся разъемы, представленные на этих рисунках?



10. По представленному рисунку составьте список с названиями интерфейсов.



11. Что больше 400 Мбит/с или 50 Мбайт/с?

Практическая занятие № 12

Тема: «Принципы работы кэш-памяти»

Цель работы:

- 1 Приобретение навыков работы с кэш-памятью.
- 2 Проверить работу различных алгоритмов замещения при различных режимах записи в кэш-память.

Порядок выполнения работы

- Прочитать методические указания
- Разобрать приведенные примеры
- Выполнить задания к работе
- Ответить на контрольные вопросы

Ввести в модель учебной ЭВМ текст своего варианта программы (см. табл. 2), ассемблировать его и сохранить на диске в виде txt-файла.

Установить параметры кэш-памяти размером 4 ячейки, выбрать режим записи и алгоритм замещения в соответствии с первой строкой своего варианта из табл. 1.

В шаговом режиме выполнить программу, фиксируя после каждого шага состояние кэш-памяти.

Для одной из команд записи (WR) перейти в режим **Такт** и отметить, в каких микрокомандах происходит изменение кэш-памяти.

Для кэш-памяти размером 8 ячеек установить параметры в соответствии со второй строкой своего варианта из табл. 1 и выполнить программу в шаговом режиме еще раз, фиксируя последовательность номеров замещаемых ячеек кэш-памяти.

Оформите отчет, который должен содержать:

- титульный лист (см. приложение);
- постановку задачи;
- Вариант задания — текст программы и режимы кэш-памяти.
- Последовательность состояний кэш-памяти размером 4 ячейки при однократном выполнении программы (команды 1—7).
- Последовательность микрокоманд при выполнении команды WR с отметкой тех микрокоманд, в которых возможна модификация кэш-памяти.
- Для варианта кэш-памяти размером 8 ячеек — последовательность номеров замещаемых ячеек кэш-памяти для второго варианта параметров кэш памяти при двукратном выполнении программы (команды 1—7).
- Вывод

В качестве задания предлагается некоторая короткая "программа" (табл. 9.14),

которую необходимо выполнить с подключенной кэш-памятью (размером 4 и 8 ячеек) в шаговом режиме для следующих двух вариантов алгоритмов замещения (табл.).

Таблица 1. Пояснения к вариантам задания

Номера вариантов	Режим записи	Алгоритм замещения
1,7, 11	Сквозная	СЗ, без учета бита записи
	Обратная	О, с учетом бита записи
2,5,9	Сквозная	БИ, без учета бита записи
	Обратная	О, с учетом бита записи
3,6, 12	Сквозная	О, без учета бита записи
	Обратная	СЗ, с учетом бита записи
4, 8, 10	Сквозная	БИ, без учета бита записи
	Обратная	БИ, с учетом бита записи

Таблица 2. Варианты задания 7

№ вари-	Номера команд программы						
	1	2	3	4	5	6	7
1	RD #12	WR 10	WR §10	ADD 12	WR R0	SUB 10	PUSH R0
2	RD #65	WRR2	MOV R4,R2	WR 14	PUSH R2	POP R3	CALL 002
3	RD #16	SUB #5	WR 9	WR @9	WR R3	PUSH R3	POP R4
4	RD #99	WR R6	MOV R7,R6	ADD R7	PUSH R7	CALL 006	POP R8
5	RD #11	WR R2	WR -@R2	PUSH R2	CALL 005	POP R3	RET
6	RD #19	SUB #10	WR9	ADD #3	WR ©9	CALL 006	POPR4
7	RD #6	CALL 006	WR11	WRR2	PUSH R2	RET	JMP 002
8	RD#8	WRR2	WR @R2+	PUSH R2	POP R3	WR -@R3	CALL 003
9	RD #13	WR14	WR@14	WR@13	ADD 13	CALL 006	RET
10	RD #42	SUB #54	WR16	WR@16	WRR1	ADD	PUSH R1
11	RD #10	WRR5	ADD R5	WRR6	CALL 005	PUSH R6	RET
12	JMP 006	RD #76	WR 14	WRR2	PUSH R2	RET	CALL 001

Не следует рассматривать заданную последовательность команд как фрагмент программы¹. Некоторые конструкции, например, последовательность команд push R6, ret в общем случае не возвращает программу в точку вызова подпрограммы. Такие группы команд введены в задание для того, чтобы обратить внимание студентов на особенности функционирования стека.

Напомним, что программа определяется как последовательность команд, выполнение которых позволит получить некий результат.

Контрольные вопросы

- 1 В чем смысл включения кэш-памяти в состав ЭВМ?
- 2 Как работает кэш-память в режиме обратной записи? Сквозной записи?
- 3 Как зависит эффективность работы ЭВМ от размера кэш-памяти?
- 4 В какую ячейку кэш-памяти будет помещаться очередное слово, если свободные ячейки отсутствуют?
- 5 Какие алгоритмы замещения ячеек кэш-памяти вам известны?

Практическое занятие № 13

Тема: «Алгоритмы замещения строк кэш-памяти»

Цель: Изучение влияния параметров кэш-памяти и выбранного алгоритма замещения на эффективность работы системы.

Теоретические сведения

Эффективность в данном случае оценивается числом кэш-попаданий по отношению к общему числу обращений к памяти. Учитывая разницу в алгоритмах в режимах *сквозной* и *обратной записи*, эффективность использования кэш-памяти вычисляется выражениям (8.2) и (8.3) соответственно для сквозной и обратной записи.

Очевидно, эффективность работы системы с кэш-памятью будет зависеть не только от параметров кэш-памяти и выбранного алгоритма замещения, но и от класса решаемой задачи. Так, линейные программы должны хорошо работать с алгоритмами замещения типа *очередь*, а программы с большим числом условных переходов, зависящих от случайных входных данных, могут давать неплохие результаты с алгоритмами *случайного замещения*. Можно предположить, что программы, имеющие большое число повторяющихся участков (часто вызываемых подпрограмм и/или циклов) при прочих равных условиях обеспечат более высокую эффективность применения кэш-памяти чем линейные программы. И, разумеется, на эффективность напрямую должен влиять размер кэш-памяти.

Задание

В данной лабораторной работе все варианты задания одинаковы: исследовать эффективность работы кэш-памяти при выполнении двух разнотипных программ, написанных и отлаженных вами при выполнении лабораторных работ № 2 и 4.

Порядок выполнения работы

Загрузить в модель учебной ЭВМ отлаженную программу из лабораторной работы № 2.

В меню **Работа** установить режим **Кэш-память**.

В меню **Вид** выбрать команду **Кэш-память**, открыв тем самым окно **Кэш-память**, в нем нажать первую слева кнопку на панели инструментов, открыв диалоговое окно **Параметры кэш-памяти**, и установить следующие параметры кэш-памяти: размер - 4, режим записи — сквозная, алгоритм замещения — случайное, без учета бита записи (W).

Запустить программу в автоматическом режиме; по окончании работы просмотреть результаты работы кэш-памяти в окне **Кэш-память**, вычислить значение коэффициента эффективности K и записать в ячейку табл. 9.15, помеченную

звездочкой.

Выключить кэш-память модели (**Работа | Кэш-память**) и изменить один из ее параметров — установить флаг с учетом бита записи (в окне **Параметры кэш-памяти**).

Повторить п. 4, поместив значение полученного коэффициента эффективности в следующую справа ячейку табл. 9.15.

Последовательно меняя параметры кэш-памяти, повторить пп. 3—5, заполняя все ячейки табл. 9.15.

Совет _____

При очередном запуске программы не забывайте устанавливать процессор модели в начальное состояние, нажимая кнопку R в окне **Процессор!**

Повторить все действия, описанные в пп. 1—7 для программы из лабораторной работы, заполняя вторую таблицу по форме табл. 1

Содержание отчета

Две таблицы по форме табл. 9.15 с результатами моделирования программ из лабораторных работ № 2 и 4 при разных режимах работы кэш-памяти.

Выводы, объясняющие полученные результаты.

Контрольные вопросы

1 Как работает алгоритм замещения *очередь* при установленном флажке С учетом бита записи в диалоговом окне Параметры кэш-памяти?

2 Какой алгоритм замещения будет наиболее эффективным в случае применения кэш-памяти большого объема (в кэш-память целиком помещается программа)?

3 Как скажется на эффективности алгоритмов замещения учет значения бита записи W при работе кэш-памяти в режиме обратной записи? Сквозной записи?

4 Для каких целей в структуру ячейки кэш-памяти включен бит использования. Как устанавливается и сбрасывается этот бит?

Таблица 3. Результаты эксперимента

Способ	Сквозная запись					
	Случайное замещение		Очередь		Бит U	
Алгоритм	безW	cW	безW	c W	безW	cW
Размер						
4	*					
8						
16						
32						
4						
8						
16						

Практическое занятие №14

Тема: «Организация памяти в компьютерах типа IBM PC»

Цель работы: закрепить знания о логической организации памяти, получить навыки использования специализированных программ для получения сведений о распределении памяти, исследовать влияние менеджеров памяти на ее распределение

Краткие теоретические сведения

Оперативная память компьютера, иначе называемая оперативным запоминающим устройством (ОЗУ), используется для оперативного обмена информацией (командами и данными) между процессором, внешней памятью (например, дисковой) и различными подсистемами (видеоподсистема, подсистема ввода/вывода, коммуникации и т.д.). ОЗУ представляет собой память с произвольным доступом, поэтому для ее обозначения часто применяется аналогичное английское понятие RAM (Random Access Memory). Произвольность доступа подразумевает возможность операций чтения/записи с любой ячейкой ОЗУ в произвольном порядке. Основными требованиями, предъявляемыми к оперативной памяти являются следующие:

- Большой (по меркам электронной памяти) объем, измеряемый десятками и сотнями мегабайт.
- Быстродействие и производительность, позволяющие наиболее эффективно задействовать мощь современных процессоров.
- Высокая надежность хранения данных.

Физически ОЗУ представляет собой просто набор последовательно расположенных ячеек, каждая из которых определяется собственным уникальным адресом. В сумме все ячейки оперативной памяти образуют **адресное пространство** (*address space*). В простейшем случае адресом некоторой ячейки памяти является просто ее номер относительно начала адресного пространства - нулевого адреса. В реальности же дело обстоит совсем иначе, поскольку по многим причинам программы не всегда могут использовать простейшую схему адресации оперативной памяти. Поэтому с точки зрения программиста наиболее важна не физическая, а логическая организация памяти. **Логическая организация адресного пространства - это совокупность способов использования программами и операционными системами оперативной памяти компьютера.**

Эффективное использование имеющегося ОЗУ является одной из главных задач, стоящих как перед программистами, так и перед пользователями, стремящимися получить максимальный выигрыш от использования персонального компьютера в своей

деятельности. Поэтому для эффективного использования памяти необходимо четкое понимание принципов ее логической организации.

Практическая часть

Для получения информации о текущем распределении памяти используются программы для просмотра состояния памяти. Некоторые из этих программ достаточно просты и позволяют получить информацию лишь самого общего характера. Для более полного обзора необходимо применять специализированные утилиты, которые часто входят в пакеты различных диагностических программ и менеджеров памяти (например, программу Quarterdeck Manifest из пакета Quarterdeck QEMM).

Просмотр состояния памяти посредством утилиты MEM

Утилита MEM, несмотря на свои достаточно ограниченные возможности, позволяет получить подробные сведения о распределении памяти. Ее важнейшим достоинством является доступность, так как она входит в комплект поставки Windows '95/98 и практически всегда находится в каталоге COMMAND системного каталога Windows. Так, например, если Windows установлена в каталог C:\WINDOWS, то MEM находится в каталоге C:\WINDOWS\COMMAND. Таким образом, использование MEM - это простейший способ получить информацию о состоянии памяти системы.

В простейшем случае утилита MEM запускается без параметров и в этом режиме выдает всю основную информацию, сведенную в несколько таблиц. Самая первая таблица показывает информацию о программных модулях, расположенных в стандартной памяти с указанием их размера. Вторая таблица представляет собой карту использования адресного пространства, в которой указаны объемы используемой стандартной и расширенной памяти. В последних строчках вывода MEM показан ряд дополнительных сведений, в частности, указано местонахождение ядра DOS (в нижней памяти или в НМА). Для получения некоторых дополнительных сведений можно использовать различные ключи при запуске MEM. Чаще всего используются ключи /C, /D, /F и /P. Ключ /P включает режим постраничного вывода и может быть использован в сочетании с любыми другими ключами. Прочие ключи являются взаимоисключающими. Использование ключа /C приводит к выводу той же информации, что и в режиме без параметров. Ключ /F показывает свободные участки памяти, а ключ /D используется для получения наиболее полной информации о распределении памяти, которая может быть использована в целях отладки. Примеры запуска:

MEM /P - постраничный вывод информации о распределении памяти

MEM /D /P - вывод детальных сведений о распределении памяти

Задание к практической работе

1. Работая в Windows '9x, выполнить запуск утилиты MEM, рассмотреть и проанализировать выводимые сведения.
2. Перезагрузиться в режиме MS-DOS, выполнить запуск MEM, проанализировать текущее распределение памяти.
3. Выполнить полную перезагрузку компьютера, по нажатию клавиши F5 в момент загрузки отключить драйвера. Исследовать распределение памяти

Вопросы для самоконтроля

- 1 Понятие о регионах. Вычисление адресов регионов.
- 2 Охарактеризовать применение расширенной и дополнительной памяти.
- 3 Структура верхней памяти.
- 4 Область НМА и особенности адресации процессоров 8086 и 80286.
- 5 Особенности адресации в защищенном режиме.

Практическое занятие № 15-16

Тема: « Команды дисковой операционной системы, каталогов и файлов»

Цель работы:

1. изучение команд операционной системы MS-DOS;
2. приобретение навыков работы с носителями информации.

Содержание работы:

- включение ПК, начальная загрузка;
- знакомство с основными устройствами ПК;
- изучение расположения клавиш на клавиатуре;
- выполнение команд операционной системы MS-DOS;

3. Порядок выполнения работы.

1. Знакомство с основными устройствами ПК.

- определите все основные устройства ПК: системный блок (процессор); дисплей (монитор); клавиатура и печатающее устройство (принтер).

2. Изучение расположения клавиш на клавиатуре.

Найдите на клавиатуре основные группы клавиш:

- функциональные или программируемые клавиши (F1-F12), в каждой программе имеют свое назначение;
- основная алфавитно-цифровая группа клавиш служит для ввода информации. Расположение клавиш на ней аналогично расположению клавиш на обычной пишущей машинке;
- малая цифровая группа служит для ввода цифровой информации или для управления курсором. Для переключения режимов работы этой клавиатуры служит клавиша Num Lock. В режиме “включено” (лампочка Num Lock горит) работает режим ввода цифр, в режиме “выключено” (лампочка Num Lock не горит) клавиатура действует как панель управления курсором;
- группа клавиш для управления курсором: Home — перевести курсор на начало текущей строки; End — перевести курсор в конец текущей строки; Page Up — перевести курсор на страницу вверх; Page Down -перевести курсор на страницу вниз; Insert — переход между режимами вставки или замены; Delete — удаление символа в позиции которого стоит курсор; Backspace
 - удаление символа слева от курсора;
 - служебные клавиши: Esc — отменить текущее действие (набранную команду, предложенное действие и т.п.); Shift — смена регистра (верхний, нижний); CapsLock

- фиксация верхнего регистра; Tab — табулятор, действие его определяется конкретной программой; Ctrl -используется в сочетании с другими клавишами, изменяя их назначение; Alt- действует аналогично клавише Ctrl; Enter — используется для выполнения указанной команды, в текстовых редакторах- переход на новую строку; Print Screen — печать содержимого экрана; Pause — временная приостановка выполнения каких-либо действий.

3. Включение ПК, начальная загрузка.

- Включите ПК тумблером, находящимся на передней панели или правой стенке системного блока;
- Затем включите дисплей тумблером на передней панели или на правой стенке; ПК начинает самотестирование и на экране дисплея появляются сменяющие друг друга сообщения. После успешного завершения самотестирования ПК выполняет попытку загрузки операционной системы с дискового устройства. Если попытка завершится неудачно (отсутствует дисковое устройство или дискета в устройстве), загрузка операционной системы происходит с жесткого диска (винчестера);
- По окончании загрузки на экране дисплея появляется приглашение операционной системы к вводу команд:

C:>_

Операционная система — это программа, которая автоматически загружается при включении компьютера. Персональный компьютер типа IBM PC работает под управлением операционной системы MS-DOS.

4. Выполнение команд операционной системы MS—DOS.

Операционная система имеет множество команд, которые делятся на внутренние и внешние. Внутренние команды входят в состав командного процессора Command.com. Внешние команды — это самостоятельные программы, представляющие собой отдельные файлы.

4.1. Команда установки даты DATE и времени TIME.

(внутренние команды)

- введите команду DATE без параметров и получите текущую дату на вашем ПК;
- если дата неправильная, с помощью этой же команды установите новую дату;
- правильность ввода проверьте введением команды DATE без параметров.
- аналогичные действия проведите с командой TIME — проверка и установка времени.

4.2. Команда включения (отключения) верификации (проверки) обмена с дисковыми устройствами VERIFY.(внутренняя команда)

- включите верификацию командой VERIFY ;
- проверьте командой VERIFY;
- отключите верификацию командой VERIFY OFF;
- проверьте командой VERIFY.

4.3. Команда проверки версии MS—DOS VER. (внутренняя команда)

- проверьте версию MS-DOS командой VER.

4.4. Команда перехода на другой диск.

• задайте команду D: и проконтролируйте выполнение появлением приглашения системы

D:>_

- перейдите назад на диск C:

5. Команды работы с каталогами, (внутренние команды)

5.1. Команда создания каталога MD.

Создайте на рабочем винчестерском диске D: каталог, в котором в качестве имени используйте 8 символов своей фамилии:

а. перейти на рабочий винчестерский диск D:

б. введите команду

MD имя каталога

5.2. Команда смены каталога CD.

Перейдите в свой каталог, используя команду CD.

CD имя каталога

С помощью команды CD можно перейти в каталог на уровень выше:

CD..

и в корневой каталог:

CD\

В своем каталоге создайте подкаталог:

MD имя подкаталога

Примечание: Не забудьте имя подкаталога, также как и каталога образуется по правилам образования имени файла, т.е. до 8 символов, не должно включать стандартных зарезервированных имен устройств (CON:,PRN:, NUL, AUX, COM1, COM2, LPT1, LPT2, LPT3), может состоять из цифр (0-9), и знаков (“минус”, “подчеркивание”, расположенных в произвольном порядке, а также некоторых других символов: \$, #, @, %, (,), {, }, ~, !)

— перейдите в свой подкаталог:

CD имя подкаталога

5.3. Команда удаления каталога *RD*.

Удаляется только пустой каталог.

6. Команды работы с файлами. (внутренние команды)

6.1. Команда копирования *COPY*.

Скопируйте с диска D из каталога WORK все файлы в свой подкаталог.

COPY D:\WORK\BABYTYPE.** Имя диска:\имя каталога\ имя подкаталога

Скопируйте с диска D: из каталога BABYTYPE все файлы, имеющие расширение pdd на винчестерский рабочий диск в подкаталог со своим именем:

*COPY D:\WORK\BABYTYPE *.pdd* Имя диска:\имя каталога\имя подкаталога

ВНИМАНИЕ: В предыдущем пункте Вы скопировали в рабочий каталог все файлы, в том числе и файлы с именами, отвечающими шаблону *.pdd, поэтому при повторном копировании файлов с уже существующими именами, на экране дисплея появится сообщение, что данный файл уже существует (already exists) и запрос: снять копирование (cancel) или переписать поверх существующего файла (overwrite).

Снимите копирование, оно не имеет смысла.

Создайте свой текстовый файл *COPY CON* имя файла

После ввода произвольного текста необходимо нажать комбинацию клавиш CTRL+Z, файл будет сохранен в вашем каталоге.

Прочитайте с экрана дисплея содержимое текстовых файлов 2 способами:

а. *COPY* имя вашего файла *CON* б. *TYPE* имя вашего файла

Используя команду *COPY* можно распечатать содержимое текстового файла на принтере:

COPY имя файла *PRN*

Создайте резервную копию своего файла, переименовав его в файл с именем имя вашего файла.01 *COPY* имя вашего файла имя вашего файла.01

Используя команду *COPY* можно объединять несколько файлов в один:

COPY имя файла1+имя файла2+... имя файла. Если не указывать имя файла, полученного в результате объединения, объединенный файл получит имя первого файла.

6.2. Команда переименования файлов *REN*

Переименуйте свой файл в вашем каталоге в файл имя файла.98:

REN имя файла имя файла.98

6.3. Команда чтения оглавления каталога *DIR*.

Команда *DIR* позволяет просмотреть содержимое любого каталога (корневого, текущего, указанного конкретно). При использовании команды *DIR* можно указывать шаблон имен файлов. Ключи команды *DIR*:

ключ /P — позволяет выводить информацию постранично, с остановками после заполнения экрана монитора. Информация выводится в полном виде (имена файлов, их объем в байтах, дата и время создания файлов);

ключ /W — информация выводится в сжатом виде (только имена файлов и каталогов), в строчку.

Выполните описанные ниже действия и объясните смысл:

- а. C>DIR D:\WORK\KURSDOS
- б. C>DIR D:\WORK\KURSDOS\VOPROSI.*
- в. C> DIR D:\WORK\KURSDOS*.d
- г. C>DIR
- д. C>DIR /P
- е. C>DIR /W

6.4. Команды удаления файлов ERASE, DEL

Удалите 2 способами файл и группу файлов в своем каталоге:

- а. ERASE имя файла
- б. DEL *.pdd

Во втором случае на экране дисплея появится запрос:

Are you sure? (y/n) Вы уверены? (да/нет) Ответьте Y (yes) — да.

После удаления всех файлов удалите свой подкаталог и свой каталог, для этого:

- выйдите из своего подкаталога: CD..
- удалите свой подкаталог:

RD имя подкаталога

- выйдите из своего каталога: CD\
- удалите свой каталог:

RD имя каталога

7. Команды работы с дисками

(внешние команды)

7.1. Команда форматирования диска —FORMAT.

• введите команду форматирования дискеты, находящейся в дисковом устройстве A: и подождите проверки правильности введенной команды преподавателем:

C:> FORMAT A: /S /T:40/N:9

• Ключ /S — на формируемый диск записываются Системные файлы MS-DOS (IO.SYS, MSDOS.SYS, COMMAND.COM);

- /T:40 — дискета размечается на 40 дорожек,
- /N:9 — девять секторов на каждой дорожке.

При таком задании ключей объем дискеты равен 360 Кб.

При форматировании дискеты команда `FORMAT` выводит сообщение

`Insert diskette for drive A: and strike ENTER when ready`

(Вставьте дискету в дисковод A: и нажмите клавишу `ENTER`)

По окончании форматирования выводится сообщение о полном объеме дискеты; объеме памяти, занятой `DOS`; объеме памяти, доступной для записи; объеме памяти, занятой сбойными участками.

7.2. Копирование диска на диск — *DISKCOPY*.

`DISKCOPY` позволяет получать точную копию диска. Для этого нужно иметь диск источник (`SOURCE`) и диск приемник (`TARGET`).

- если дисковое устройство одно, то копирование проводится с диска A: на диск A:
- если дисковых устройств два, то копирование производится с диска A: на диск B:

Программа `DISKCOPY` копирует дискеты по дорожкам.

Если форматы дискет разные, то программа выдаст сообщение:

`Drive types or diskette types not compatible`

(Типы дисководов или дискет несовместимы)

или `Disks must of the same size`

(Дискеты должны быть одного размера)

Если дискета не форматирована, то программа ее форматирует. Формат тот же что и у исходной дискеты.

- введите команду

`C:> DISKCOPY A: A:`

- после считывания исходной дискеты следует указание вставить дискету приемник.

7.3. Проверка точности копирования дискет — *DISKCOMP*.

- перейдите на диск C:
- сравните две дискеты командой `DISKCOMP A: A:`. Сравнение заканчивается вопросом: Сравнить еще? (да/нет). Ответьте нет.

7.4. Проверка дисков и файлов — *CHKDSK*.

- введите команду

`C:> CHKDSK A: /F`

- проверяется диск A:. Ключ `/F` задает режим, при котором системой будут предприниматься попытки автоматического устранения обнаруженных ошибок (указывается только при проверке диска).

- введите команду

C:> CHKDSK A:*. *

- команда проверяет все файлы на диске A: .

В сообщении указывается емкость диска, количество скрытых файлов (если они есть), объем памяти, который они занимают, число пользовательских файлов и их объем, свободный объем памяти на диске, дефектные дорожки или сектора и их объем, общий объем памяти ЭВМ и свободная память после размещения DOS.

- выполнить команду для диска D:, но без ключа.

Итог

Итак, Вы познакомились с составными частями ПЭВМ IBM PC/AT и научились работать с командами операционной системы MS-DOS: форматировать и проверять дискеты, копировать дискету на дискету, читать и выводить на печать оглавление активного или любого другого диска, каталога, подкаталога, создавать, по мере необходимости, собственные каталоги (подкаталоги), копировать, переименовывать в них файлы или группы файлов по маске, приобрели навыки удаления ненужной (устаревшей) информации; закрепили приемы работы с активным дисководом, переходом от одного накопителя к другому.

Задание повышенного уровня сложности (дополнительное).

1. Проанализируйте истины или ложны приведенные ниже высказывания:

а. команда DIR не меняет содержимого дисков, каталогов, подкаталогов, а лишь показывает их оглавление.

б. команда DIR в своих разновидностях изменяет содержимое каталогов, в зависимости от шаблона (умолчаний — “*” или комбинация символов “?”).

2. Возможно использование сокращенных обозначений в маршрутах для указания текущего каталога, расположенного на один уровень выше.

Например:

а. DIR.. — выдаются файлы в каталоге на уровень выше текущего.

б. DIR../.. — на экран дисплея выводится оглавление всех файлов в каталоге на 1 уровень выше.

Приведите примеры содержимого винчестерских дисков C:, D:, E:.

3. Прочитать с экрана дисплея содержимое текстовых файлов inf.txt и fiz.txt с разбиением на порции: командой MORE.

Контрольные вопросы к защите лабораторной работы:

1. Понятие и назначение ДОС.

2. Что такое файл? Как задается имя файла?
3. Что такое каталог (директория), подкаталог?
4. Полный путь к файлу?
5. Что такое шаблон (маска) имен файлов?
6. Какими символами обозначаются дисководы?
7. Понятие внешние и внутренние команды ОС.
8. Что означает операция форматирования и как ее выполнить?
9. Как загрузить ОС с дискеты, а не с жесткого диска? Опция для

указания записи системных файлов при форматировании.

10.Как переписать информацию с одной дискеты на другую? Требования, предъявляемые к дискетам при этом?

11.Как проверить правильность копирования дискет?

12.Как проверить дискету на наличие сбойных участков?

13.Команды работы с каталогами.

14.Команды работы с файлами.

15.Команды общесистемного назначения

Практическая работа № 17-18

Тема: «Настройка BIOS»

Цель работы: «Изучение специфики и возможностей базовой системы ввода/вывода материнской платы»

Теоритические сведения

В данном руководстве вы сможете найти описание разных настроек BIOS в картинках, ведь именно сложность или просто страх перед этой микропрограммой не даёт многим установить Windows самостоятельно.

Примером послужит настройка BIOS ASUS материнской платы P5QL SE.



Первое загрузочное окно BIOS материнской платы ASUS P5QL SE

Что же такое BIOS

BIOS расшифровывается как Basic Input Output System что в переводе с английского Базовая система ввода-вывода - которая представляет из себя микропрограммы входящие в состав системного ПО для доступа к аппаратной части компьютера и подключенным к нему устройствам. BIOS – это специальная программа, записанная на чип материнской платы. Именно здесь хранятся все настройки системы. Данную микросхему можно отнести к разряду энергозависимых запоминающих устройств, но при этом она остаётся постоянным запоминающим устройством. Можно подумать, что при отключении питания ПК, все настройки в таком случае, которые хранятся в BIOS, собьются, но это не так, потому что у BIOS есть свой собственный источник питания – батарейка, которая также устанавливается в материнскую плату. Основной задачей BIOS является опознание всех подключенных устройств к компьютеру (видеокарты, процессора, оперативной памяти и т.д.), проверить их на работоспособность, и запустить с определёнными параметрами, после чего BIOS передаёт управление загрузчику операционной системы. Таким образом, BIOS является базовой подсистемой ввода/вывода, и посредником между аппаратной частью компьютера и операционной

системой, т.е. BIOS обеспечивает взаимодействие между устройствами друг с другом и с операционной системой.

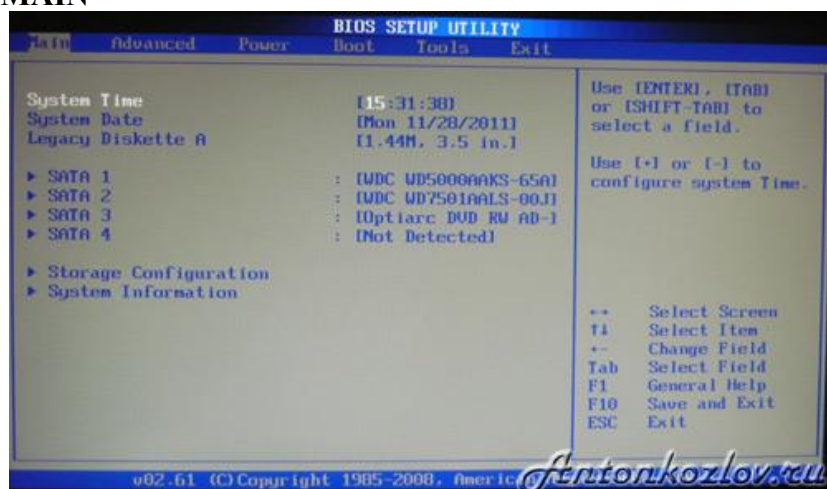
На данный момент существует несколько версий BIOS, наиболее популярными из которых считаются BIOS от Phoenix Award и BIOS American Megatrends (AMI). Чтобы настраивать свой BIOS, если вам ни разу не доводилось этого делать раньше, то лучше узнать версию и модель вашего BIOS и посмотреть подробности по настройке в интернете. В данной статье описывается BIOS AMI настройка.

Что надо сделать, чтобы открыть меню настроек BIOS?

Сразу после включения ПК, BIOS начинает грузиться самым первым. Быстро загрузившись, он начинает выполнять процедуру POST, с помощью которой опрашивает все подключенные устройства на работоспособность и готовность к работе. Если все подключённые устройства готовы к работе, и работают нормально, то прозвучит короткий звуковой сигнал из динамика, который находится в BIOS. После этого короткого сигнала, следует быстро нажать клавишу «del» или «F10», на вашей клавиатуре. Если же вы не слышите сигнал, по каким-то причинам, то нажимать на эти клавиши можно в то время как на экране, на чёрном фоне будут выводиться различные данные, которые являются результатом проводимого POST на готовность устройств. Для уверенности, что вы попадёте в BIOS, на клавиши можно нажимать много раз, сразу после того, как начнет включаться компьютер. Если при включении компьютера, вы услышите вместо привычного короткого сигнала BIOS, сигнал, который более длительный, или повторяющийся, то во время процедуры POST, выяснилось, что одно или несколько подключенных устройств, работают некорректно или вышли из строя.

После того, как вы удачно попали в окно настроек BIOS, в версии BIOS AMI 02.61 (и во многих других версиях AMI), вы увидите:

Раздел MAIN



Раздел AMI BIOS — MAIN

В этом разделе вам будет представлена возможность самостоятельно настроить время и дату системы, а также настроить подключенные жесткие диски и другие накопители. Как правило, BIOS автоматически определяет все подключенные устройства, поэтому их не надо будет здесь включать вручную, но можно изменить некоторые параметры при их автоматическом включении. Для этого следует выбрать нужный вам жесткий диск или другой накопитель, и нажать «Enter» на клавиатуре. После этого вы попадете в меню настроек выбранного накопителя. В нашем случае мы видим всю информацию о подключенном к первому каналу жестком диске. Если к каналу не подключено ни одно устройство, то мы видим надпись «Not Detected».

Попав в настройки жесткого диска, чтобы внести свои настройки, следует изменить параметр «Type» с Auto на User.

LBA Large Mode – данный параметр отвечает за поддержку накопителей, объемом которых более 504 Мбайт. Таким образом здесь желательно выбрать значение AUTO.

Block (Multi-Sector Transfer) – с помощью этого параметра можно отключить передачу данных нескольких секторов одновременно по 512 байт, т.е., отключая данный параметр, резко снизится скорость работы жесткого диска, ведь за раз будет передаваться только 1 сектор равный 512 байт. Для быстрой работы лучше поставить режим AUTO.

PIO Mode – с помощью данного параметра, можно заставить жесткий диск работать в устаревшем режиме обмена данными. Автоматически современные жесткие диски работают в самом быстром режиме, поэтому, здесь также лучше выставить значение AUTO.

DMA Mode – прямой доступ к памяти. Для достижения более высокой скорости чтения/записи, следует оставить значение AUTO.

SMART Monitoring – с помощью этой технологии можно отслеживать состояние жесткого диска. Другими словами – это технология самонаблюдения, отчетности и анализа. Также лучше выставить значение AUTO.

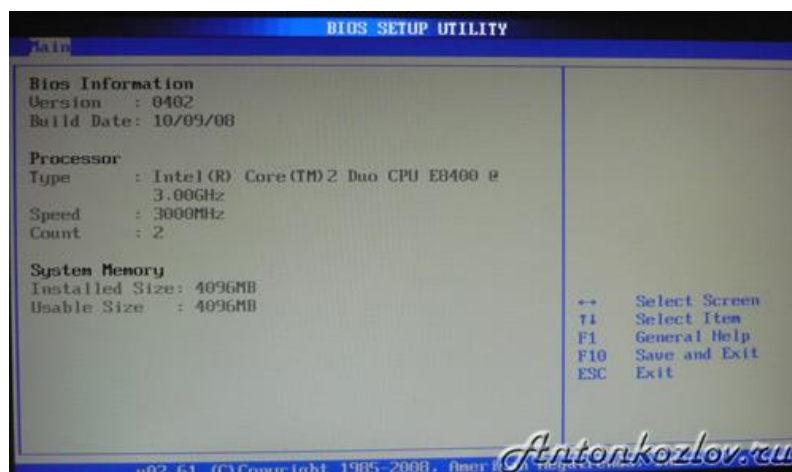
32 Bit Transfer – если стоит значение Enabled, то передаваемые по шине PCI данные, будут передаваться в 32-битном режиме. Если этот параметр отключен, то в 16-битном.



Данные жесткого диска на SATA 1 в разделе MAIN

System Information

В разделе MAIN, также можно узнать некоторую информацию о системе. Для этого надо в данном разделе выбрать пункт System Information. В открывшемся окне вы увидите Версию BIOS и дату его производства, также тут есть информация о процессоре и системной памяти.



System Information (системная информация) в разделе MAIN

Storage Configuration

Выбрав в разделе MAIN пункт Storage Configuration, мы перейдем в настройки дисковой подсистемы. Здесь, изменяя параметры SATA Configuration, мы можем отключить SATA-контролер, который впаян в материнскую плату, выбрав параметр Disabled. Выбрав режим Enhanced, система будет работать в обычном, привычном для всех современных операционных систем режиме. И наконец, выбрав режим Compatible, дисковая подсистема будет работать в режиме совместимости с устаревшими операционными системами. (Windows 98,95,Me)

Configure SATA as. Изменяя данный параметр, вы можете выбрать режим IDE, используя который, вы будете видеть все подключенные диски в виде IDE-устройств, или же, можно выбрать режим AHCI, который поддерживают большинство современных ОС. Выбрав AHCI, позволит вам использовать современные технологии, такие как Plug-in-Play.

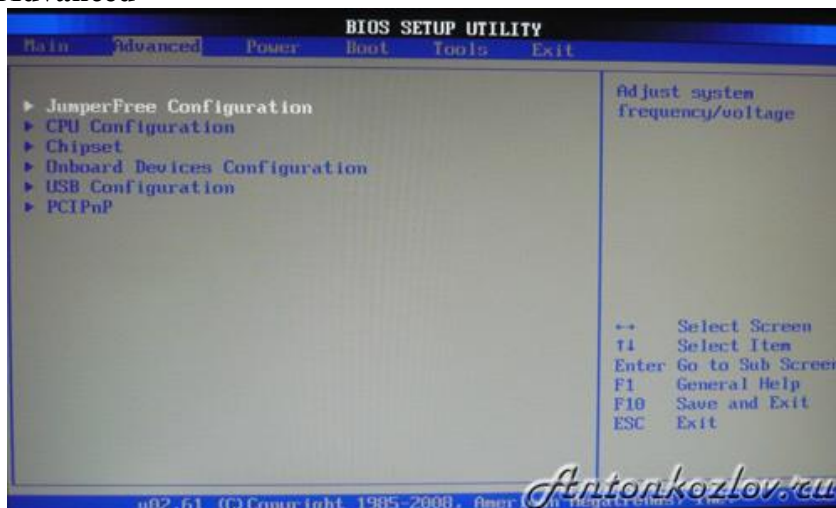
Hard Disk Write Protect и SATA Detect Time out. Основной задачей данных параметров является защита дисков от записи, т.е. лучше оставить параметр Hard Disk Write Protect в режиме Disabled. Изменяя параметр SATA Detect Time out, можно изменить время, которое будет потрачено компьютером на поиск дисковой подсистемы при включении. Соответственно задав меньшее время, загрузка будет проходить быстрее, но сильно уменьшать его не надо, а лучше наоборот немного увеличить или оставить по

умолчанию, ведь при малом времени, диски просто могут не успеть определиться системой во время прохождения процедуры POST.



Конфигурация шины SATA

Раздел Advanced



Раздел AMI BIOS — Advanced

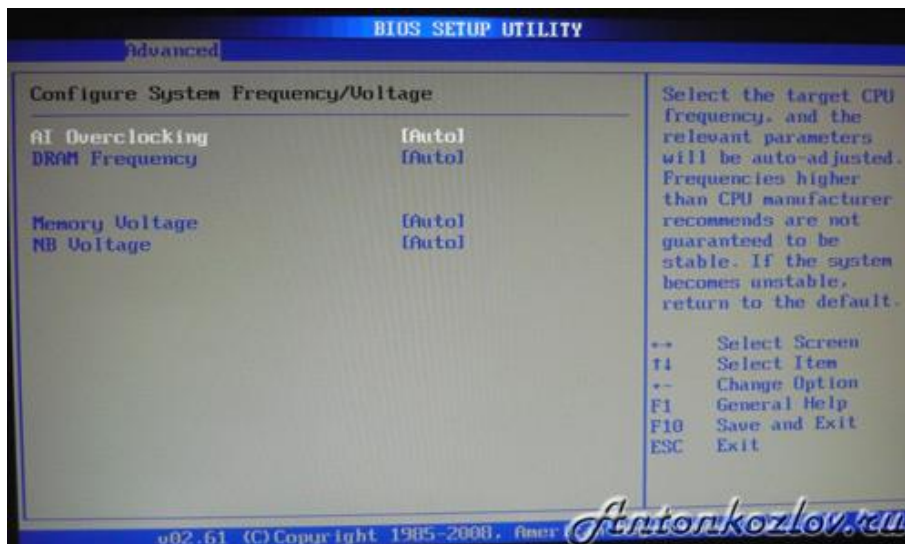
Выбрав в данном разделе пункт JumperFree Configuration, мы попадаем в раздел Configure System Frequency/Voltage.

Configure System Frequency/Voltage

AI Overclocking – Выбрав режим AUTO в данном параметре, можно произвести автоматический разгон процессора, если же здесь выбрать режим Manual, то настройки для разгона процессора можно произвести вручную.

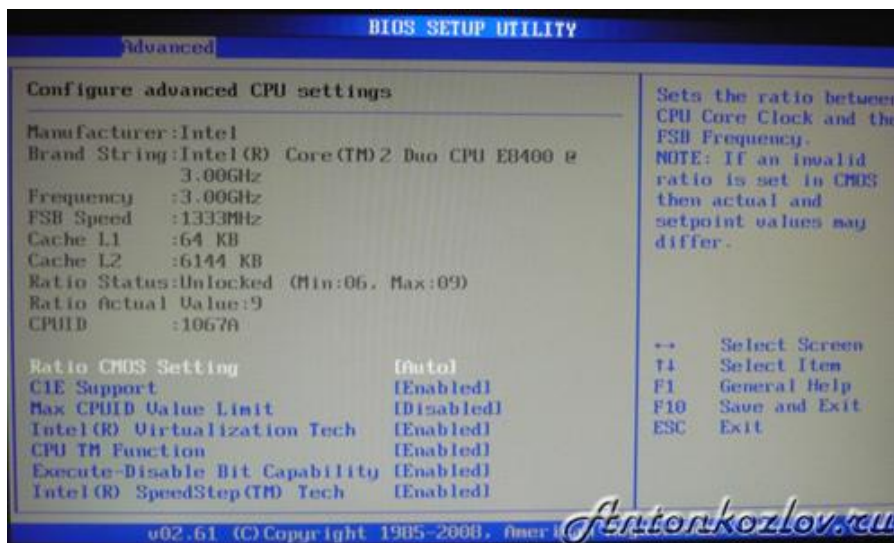
DRAM Frequency – этот параметр позволяет задать частоту шины памяти независимо от частоты шины процессора.

Memory Voltage и **NB Voltage** – данные параметры позволяют вручную настроить напряжение питания модулей памяти и задать напряжение системного чипсета.



Конфигурация оперативной памяти

Перейдя из раздела Advanced в CPU Configuration, у нас откроется окно **Configure Advanced CPU Settings**. В этом разделе можно изменять все настройки связанные с центральным процессором, также можно наблюдать за его работой и узнавать все сведения о центральном процессоре.



Конфигурация процессора

Onboard Devices Configuration

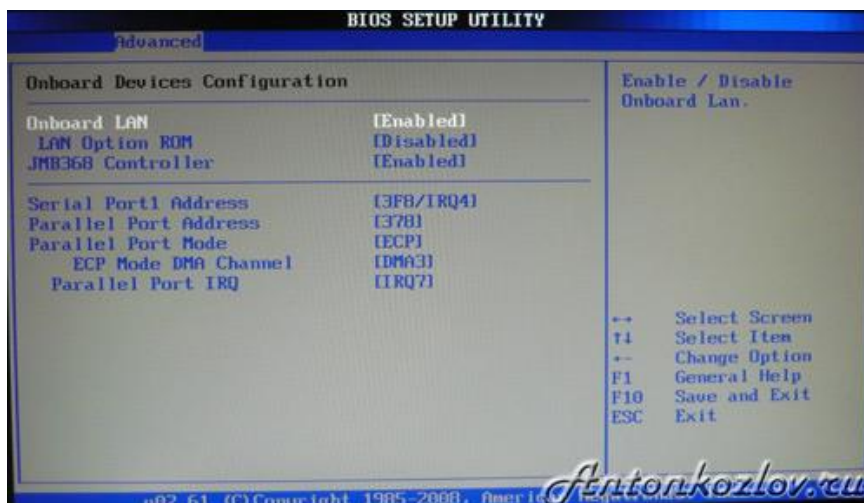
Здесь находятся параметры, которые влияют на работу встроенных контролеров и портов. Например, изменяя значения Onboard LAN, мы можем отключить или наоборот включить встроенный сетевой адаптер.

Serial Port1 Address – позволяет присвоить COM-порту один из закреплённых адресов, и номер прерывания, который будет в дальнейшем использоваться операционной системой.

Parallel Port Address — присваивает LPT-порту один из закреплённых адресов, который в дальнейшем будет использоваться операционной системой.

Parallel Port Mode — указывает BIOS, в каком режиме должен работать параллельный порт компьютера.

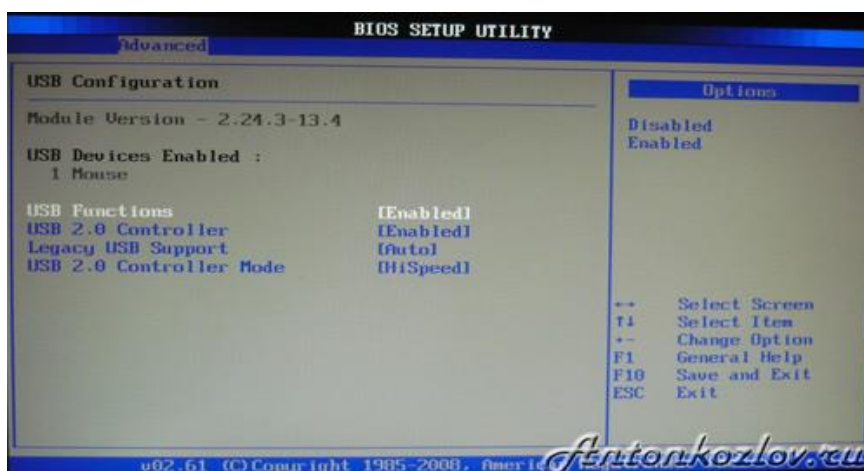
Также в этом разделе можно присваивать адреса и другим портам.



Настройка устройств встроенных в материнскую плату

USB Configuration

В этом разделе можно производить изменения работы последовательного интерфейса USB. Здесь можно также полностью отключить все USB разъёмы на вашем компьютере.



Конфигурация в BIOS портов USB

Раздел POWER (параметры питания)

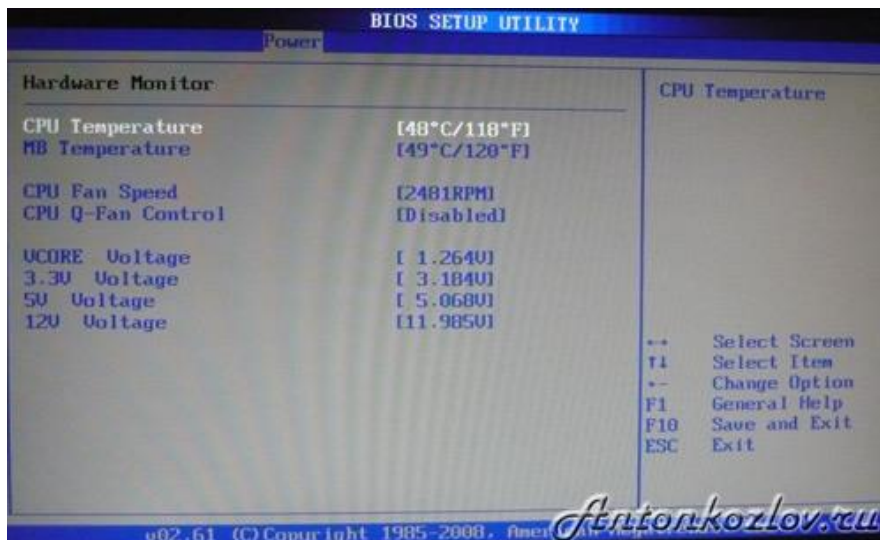
В разделе Power можно настроить функции энергосбережения и модель включения и отключения вашего компьютера.

ACPI – Advanced Configuration and Power Interface – этот интерфейс расширенного управления питанием.



Раздел AMI BIOS — Power

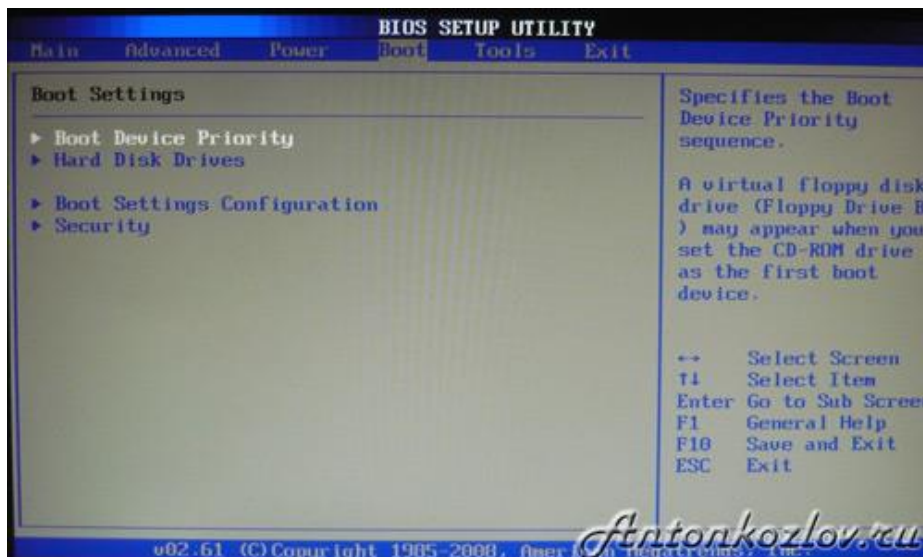
Перейдя в пункт Hardware Monitor, мы найдём всю информацию о температуре нашего процессора или о скорости вращения вентиляторов. Также тут можно узнать информацию со всех остальных датчиков компьютера, и внести изменения в некоторые параметры блока питания.



Данные температуры и вольтажа в разделе BIOS Hardware Monitor

Раздел Boot (параметры загрузки)

В данном разделе можно произвести изменения в параметрах загрузки.



Раздел AMI BIOS — Boot (загрузка)

Перейдём в раздел **Boot Device Priority**.

Тут у нас есть возможность настроить, какой накопитель у нас будет загружаться в первую очередь. Таким образом, на рисунке видно, что в первую очередь у нас загружается информация с жесткого диска, т.е. при включении компьютера, операционная система начнёт грузиться с жесткого диска.



Раздел BIOS Boot Device Priority — выбор порядка приоритета устройств. На скриншоте первый идёт жесткий диск

Эти параметры не трудно поменять, и уже из следующего рисунка мы видим, что в первую очередь начнёт загружаться информация с подключенного дисковод. Данные параметры следует менять, если вы используете переносную версию операционной системы или при переустановке ОС.

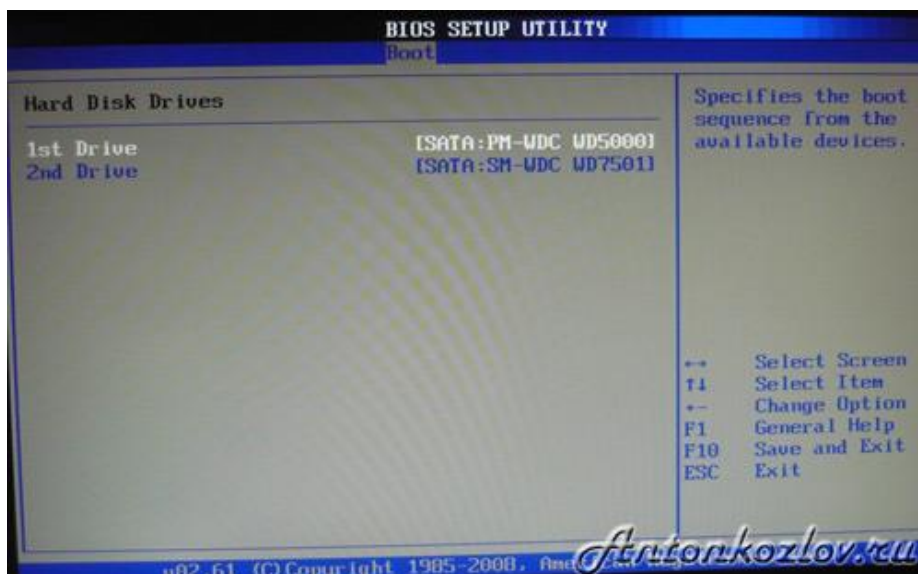


Для того, чтобы установить операционную систему нужно первым поставить привод лазерных дисков

Hard Disk Drivers

Данный параметр позволит вам изменить жёсткий диск, с которого будет первоначально загружаться операционная система.

Данную опцию следует использовать, когда в вашем компьютере установлено несколько жестких дисков.



Конфигурация жестких дисков на ведущего и ведомого при помощи BIOS и раздела Hard Disk Drives

Boot Setting Configuration (Boot)

Это раздел, в котором содержатся настройки, влияющие на процесс загрузки операционной системы, инициализацию клавиатура и мышки, обработку ошибок и т.д.

Quick Boot – опция, активировав которую, каждый раз при включении компьютера, BIOS будет проводить тест оперативной памяти, что приведёт к более быстрой загрузке операционной системы.

Full Screen Logo – активировав данный параметр, каждый раз при перезагрузке компьютера, на монитор будет выводиться графическое изображение, вместо текстовой информации, на котором можно будет увидеть информацию о процессоре, видеокарте, оперативной памяти и других компонентах ПК.

Add On ROM Display Mode – опция, которая определяет порядок появления на экране информации об устройствах, которые подключены через платы расширения и имеют свой собственный BIOS.

Bootup Num-Lock – опция, определяющая, в каком состоянии при включении ПК должна быть клавиша «Num Lock»

Wait For 'F1' If Error – включение этой опции, заставит пользователя нажать клавишу «F1», если на начальной стадии загрузки ПК обнаружится ошибка.

Hit 'DEL' Message Display – опция, которая управляет появлением на экране (или скрыванием) надписи, которая говорит о том, какую клавишу следует нажать, чтобы открыть окно настроек BIOS. Для многих пользователей не секрет, что для того, чтобы попасть в BIOS, следует нажимать клавишу «del», поэтому данную функцию можно отключить.



Раздел Bios — Boot Setting Configuration

Security Setting – настройки защиты.

Supervisor Password – данная функция позволяет изменить, удалить, или задать новый пароль администратора для доступа в BIOS.

User Password – данная функция позволяет изменить пароль, придумать новый или удалить, только для обычных пользователей.

Отличие эти функций в том, войдя в BIOS с помощью пароля пользователя, у вас будет лишь возможность просматривать все установленные там настройки, не проводя

никаких изменений, а если же вы вошли в BIOS как администратор, то у вас будет доступ к изменению всех параметров BIOS.

При нажатии клавиши «del», после запуска компьютера, чтобы попасть в BIOS, каждый раз будет вылазить окно с просьбой ввести пароль.



В разделе BIOS Security Setting можно задать пароль для доступа к BIOS и загрузке

Раздел Tools

ASUS EZ Flash – при помощи данной опции, у вас есть возможность обновлять BIOS с таких накопителей как: дискета, Flash-диск или компакт-диск.

AI NET – воспользовавшись этой опцией, можно получить информацию о подключенном к сетевому контролеру кабеле.



Раздел в BIOS Asus с набором фирменных утилит для обновления прошивки BIOS

Раздел Exit

Exit & Save Changes (F10) – используется, чтобы выйти из BIOS сохранив при этом все внесённые изменения.

Exit & Discard Changes — выход с отменой всех внесенных изменений.

Discard Changes – отмена всех внесённых изменений.

Load Setup Defaults – установка значений по умолчанию.



Последний раздел в BIOS — Exit

Вот в общих чертах и всё. После того, как вы прочитаете это руководство по BIOS, вы можете настроить свой компьютер, сможете без посторонней помощи настроить BIOS для установки Windows, включить или отключить встроенные устройства, например, аудио или сетевую плату. Кстати, не забудьте прочитать о новой версии BIOS UEFI, именно она скоро будет установлена во все современные материнские платы

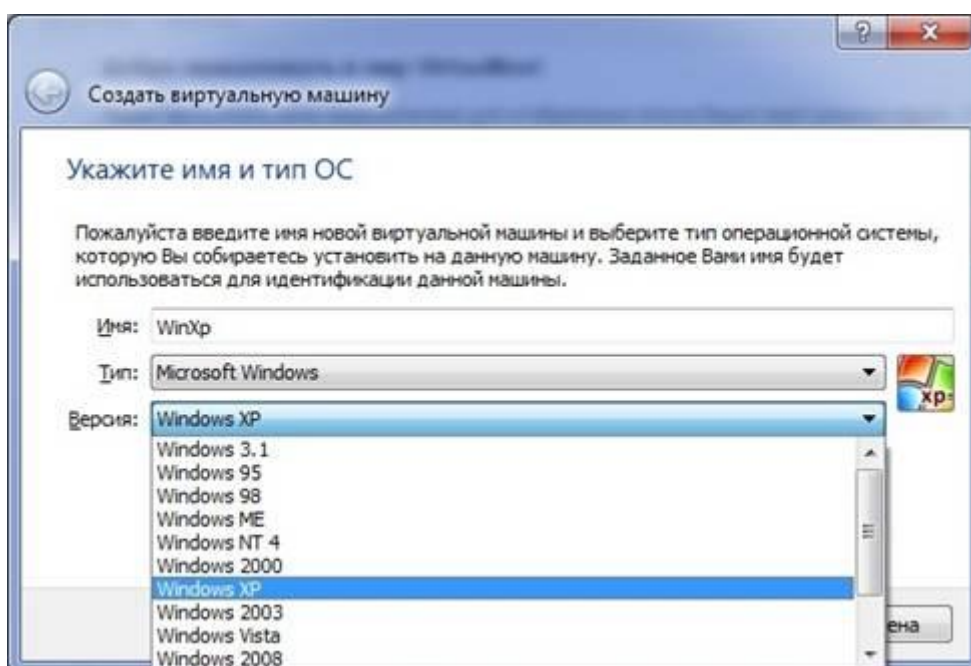
Практическая работа № 19-20

Тема: «Установка windows xp на виртуальную машину virtualbox».

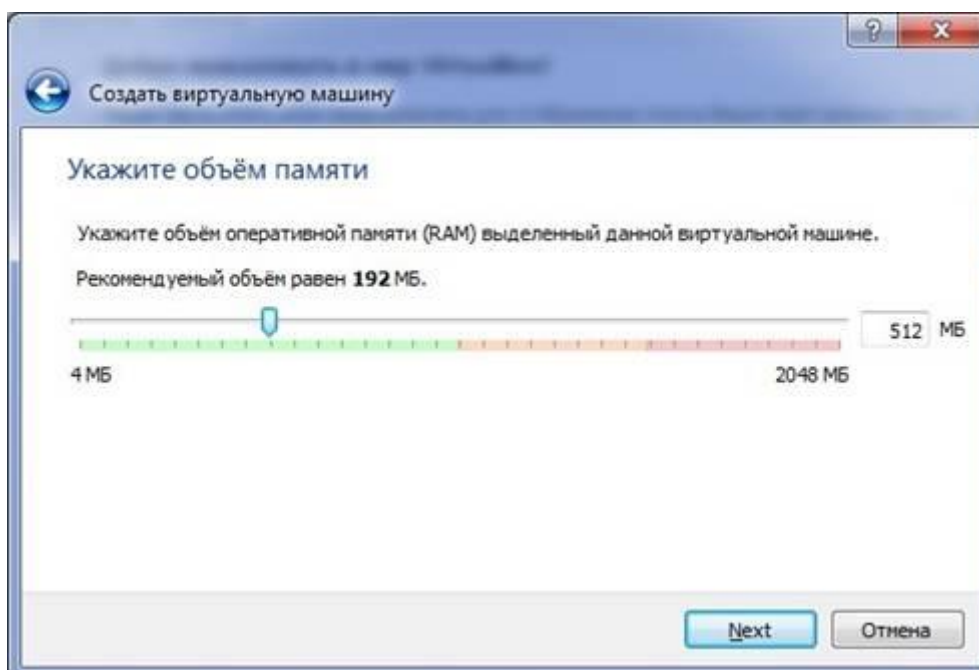
Цель работы: Научится устанавливать операционную систему *Windows XP* в *VirtualBox*, которая пригодится для запуска программного обеспечения, работы с реестром и использованием служебными возможностями ОС .

Задание:

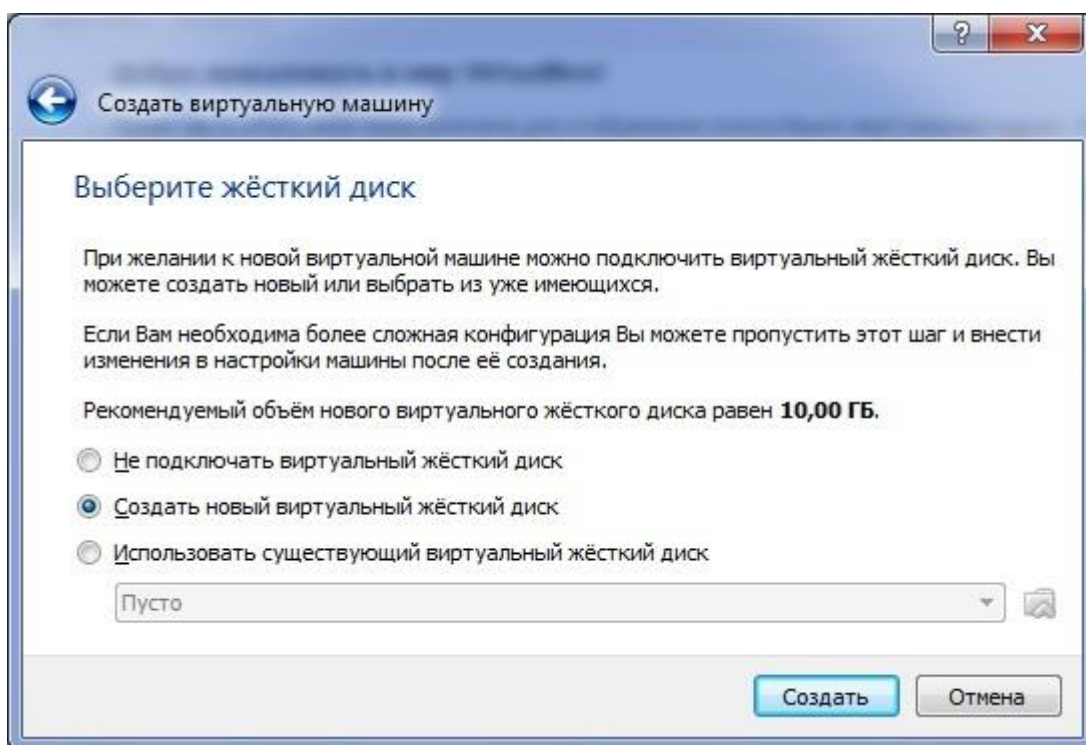
1) Запускаем *VirtualBox*. Нажимаем на кнопку «Создать» в верхнем левом углу. В появившемся окне вписываем имя нашей операционной системы. Выбираем версию «*Windows XP*». Кликаем «*Next*».



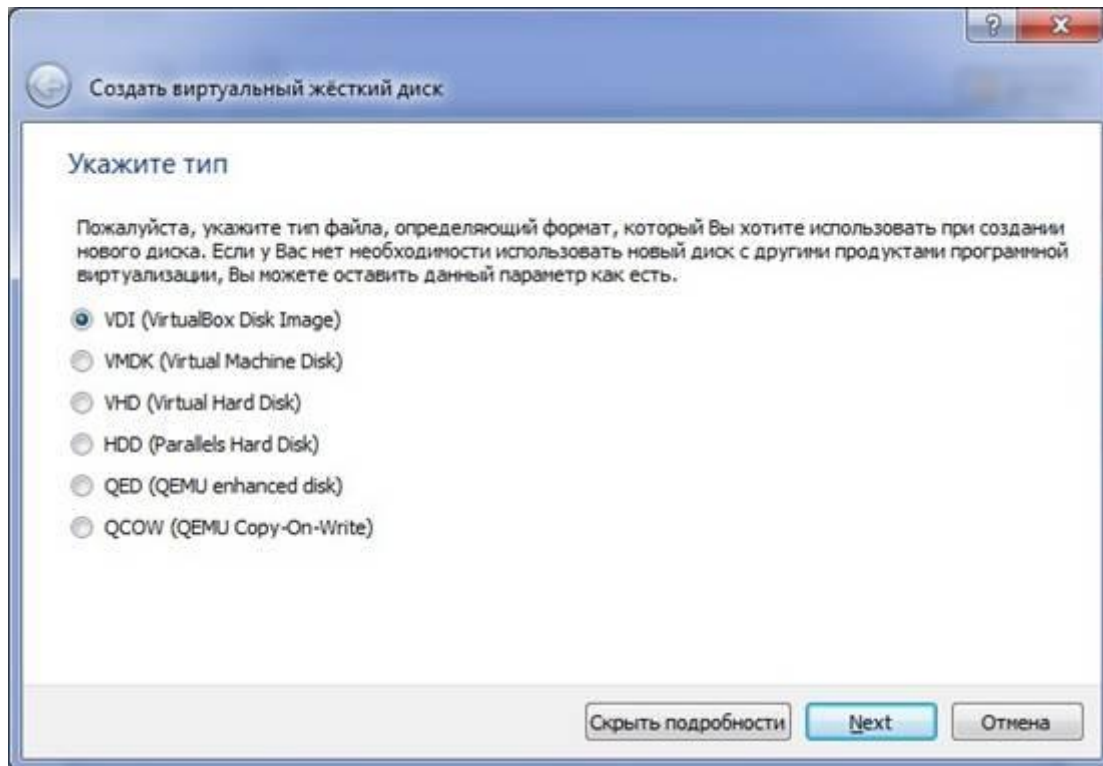
2) Выбираем размер оперативной памяти, который может использовать виртуальная машина.



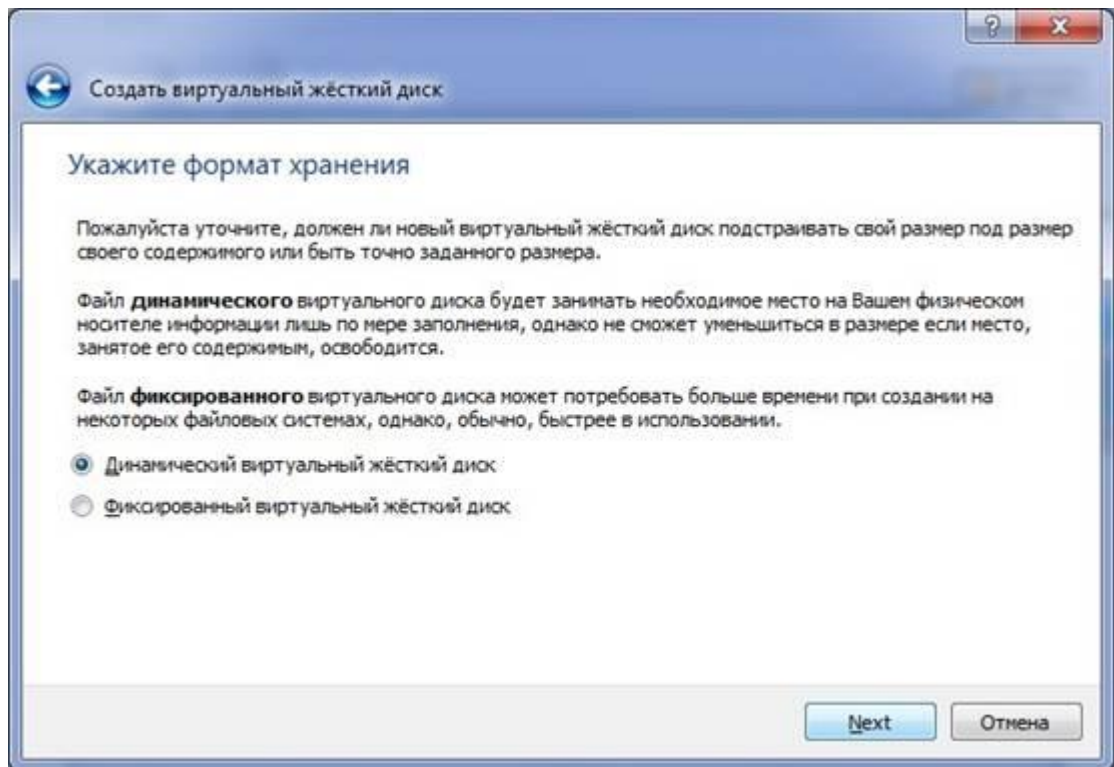
3) Создадим новый жёсткий диск. Выбираем пункт «Создать новый виртуальный жёсткий диск» и кликаем «Создать».



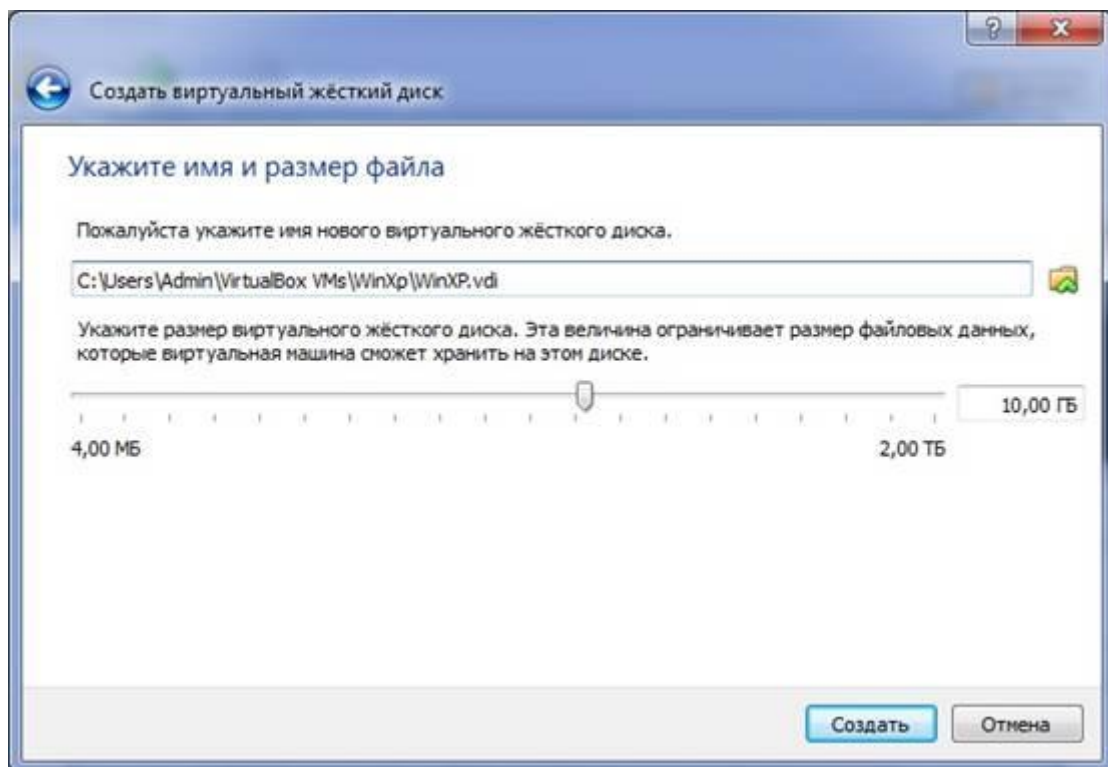
4) В новом окне указываем тип файла, который будет использован при создании диска «VDI (VirtualBox Disk Image)». Нажимаем «Next».



5) Указываем формат хранения «Динамический виртуальный жёсткий диск». Кликаем «Next».



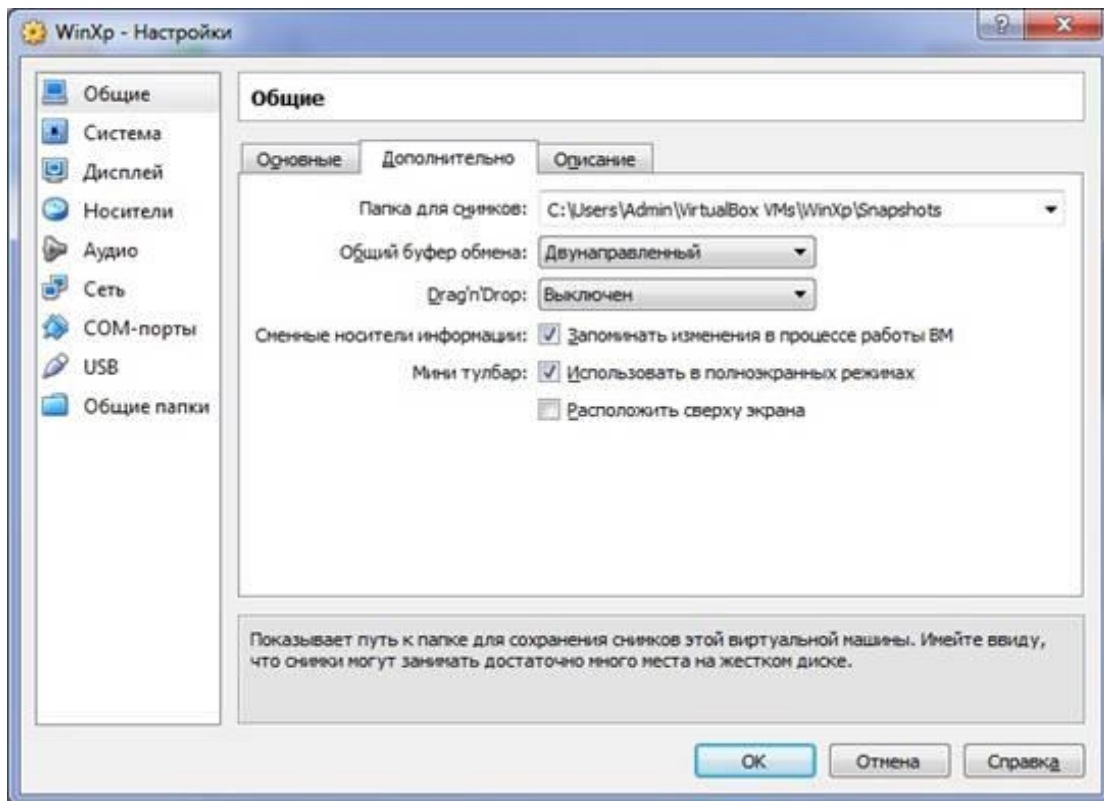
б) Вводим имя нового виртуального диска и указываем его размещение. Далее задаем его размер 10,00 ГБ. (рекомендовано для *Windows XP*), щелкаем «Создать».



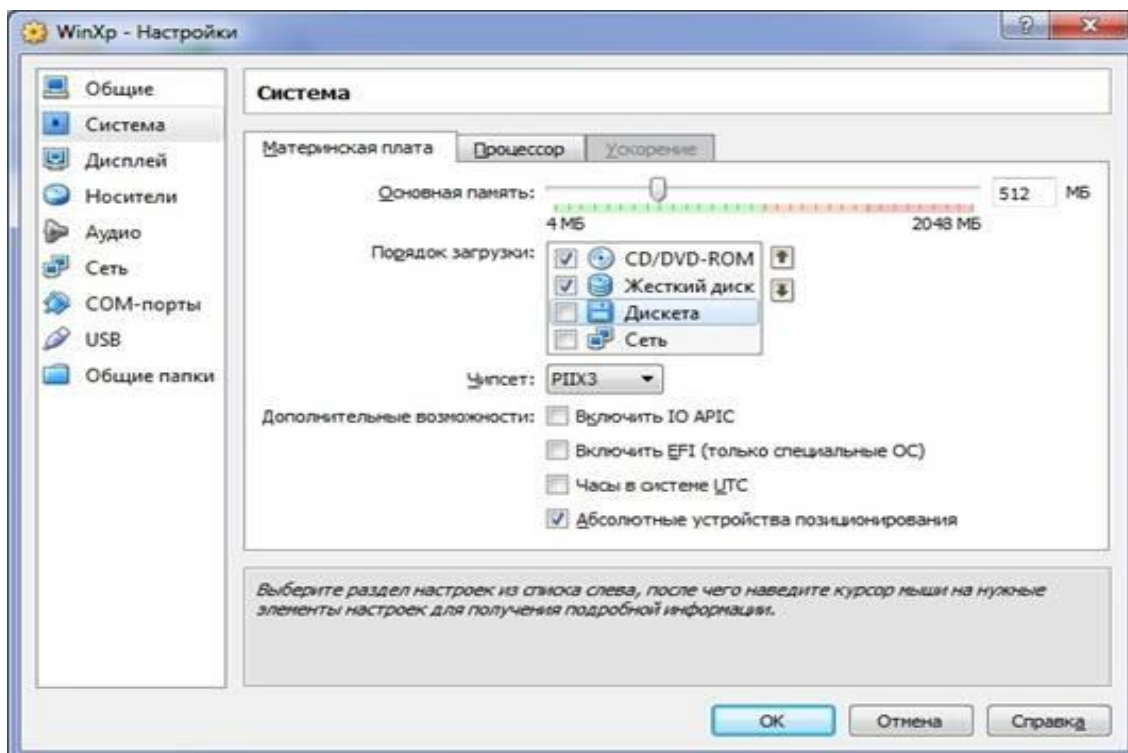
Виртуальная машина создана.

7) Заходим в «Настройки» в верхнем левом углу и нажимаем на вкладку «Общие».

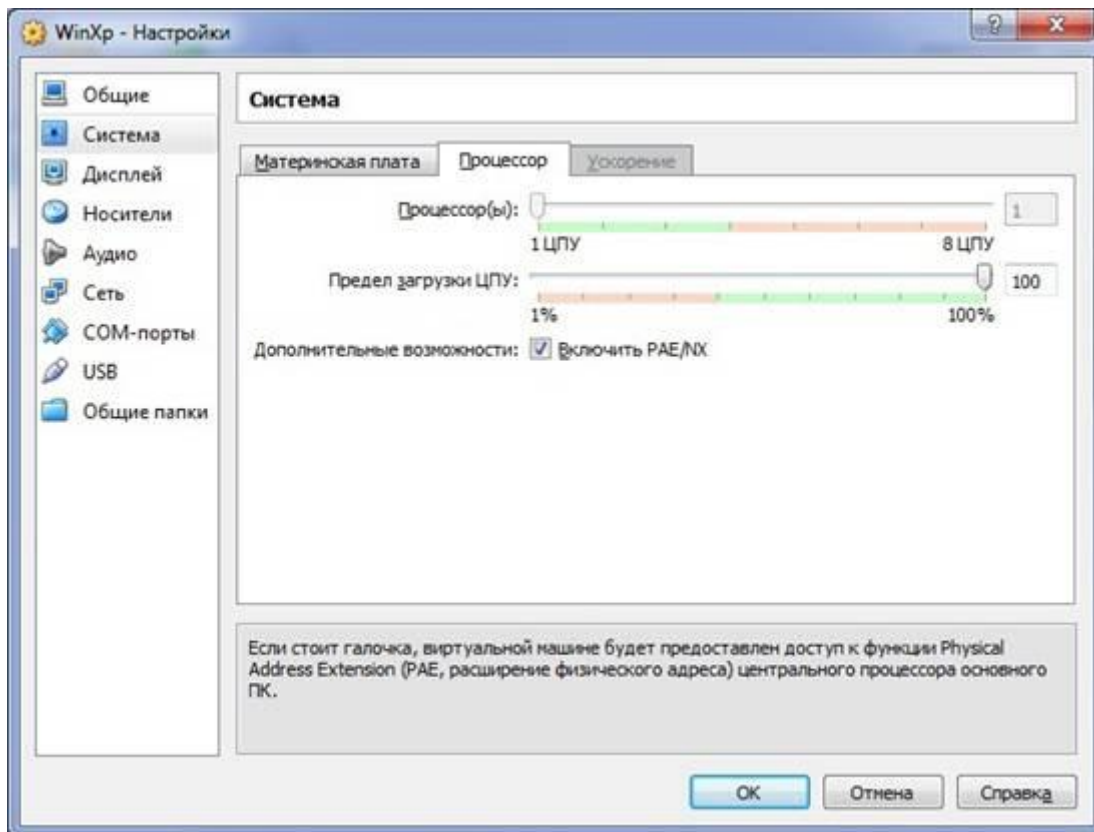
Кликаем на «Дополнительно». Настроим режим работы буфера обмена, между вашим компьютером и виртуальной машиной. В выпадающем списке «Общий буфер обмена» выбираем «Двунаправленный». Ставим галочку напротив «Использовать в полноэкранных режимах» и «Запоминать изменения в процессе работы ВМ».



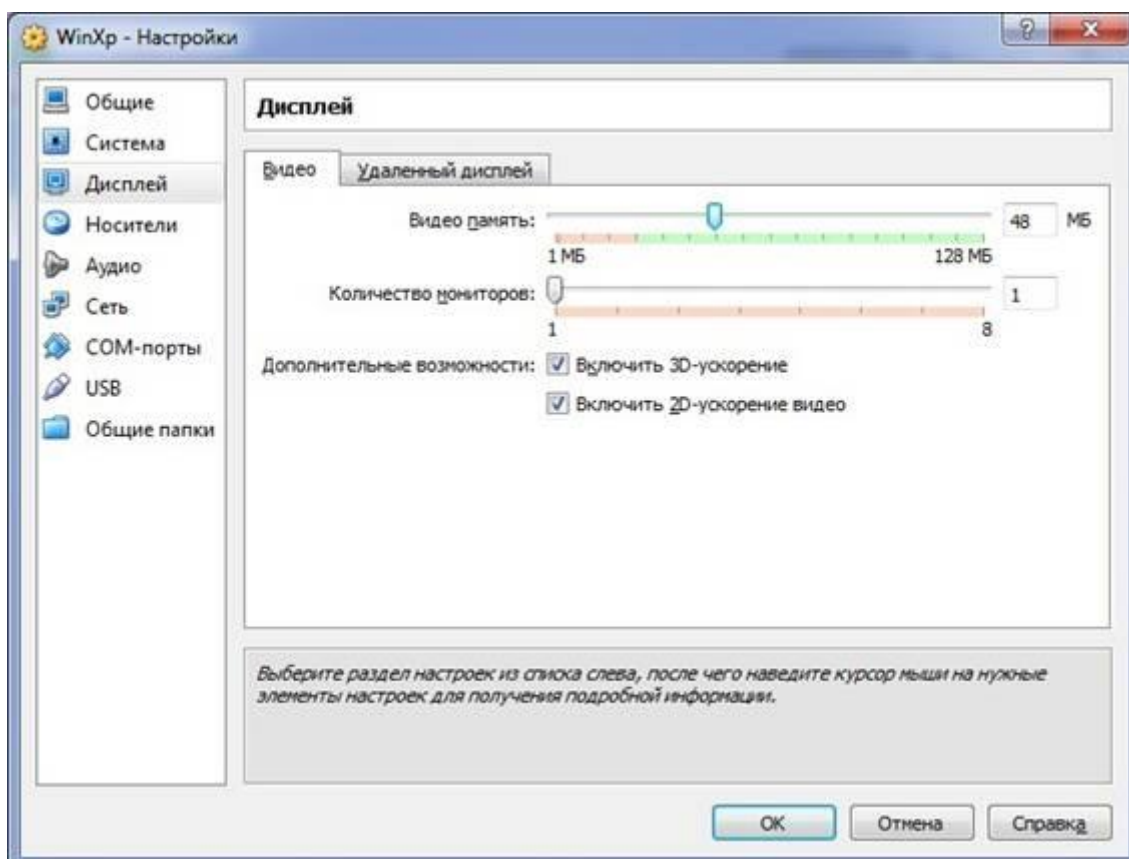
8) Заходим на вкладку «Система». Выставляем загрузку с CD/DVD-ROM.



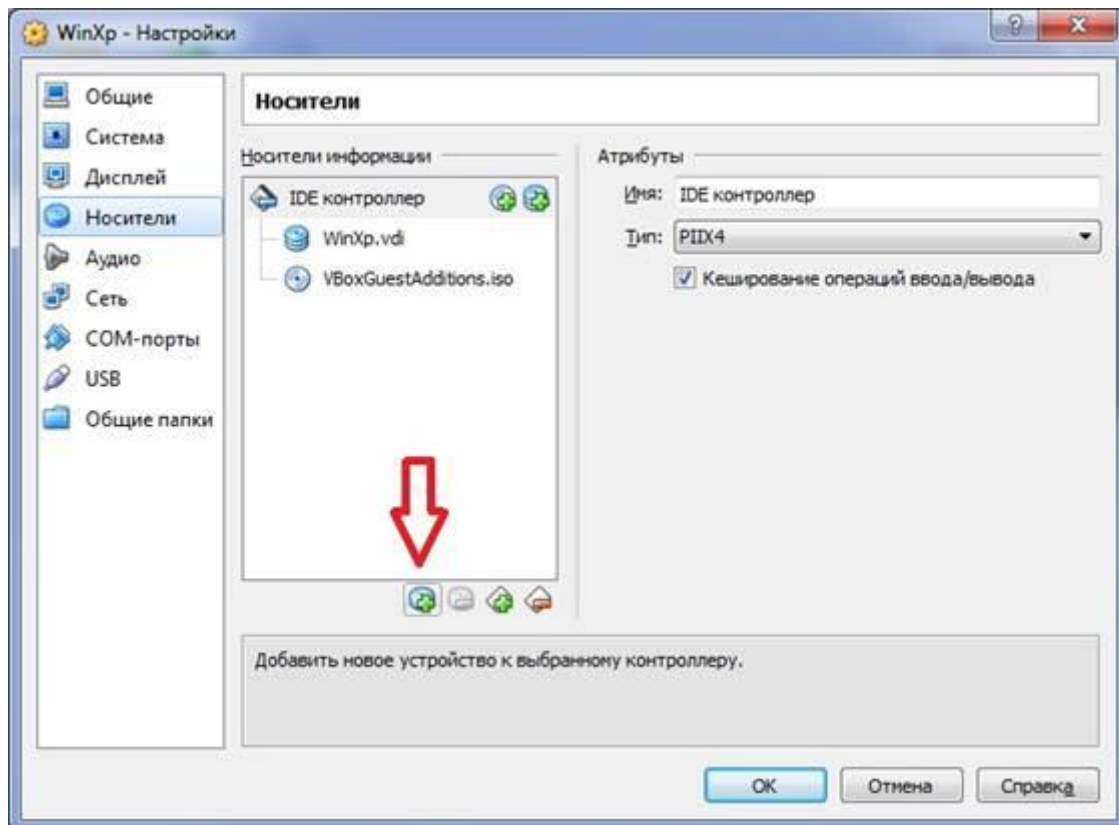
Переходим на «Процессор» и ставим галочку «Включить PAE/NX».



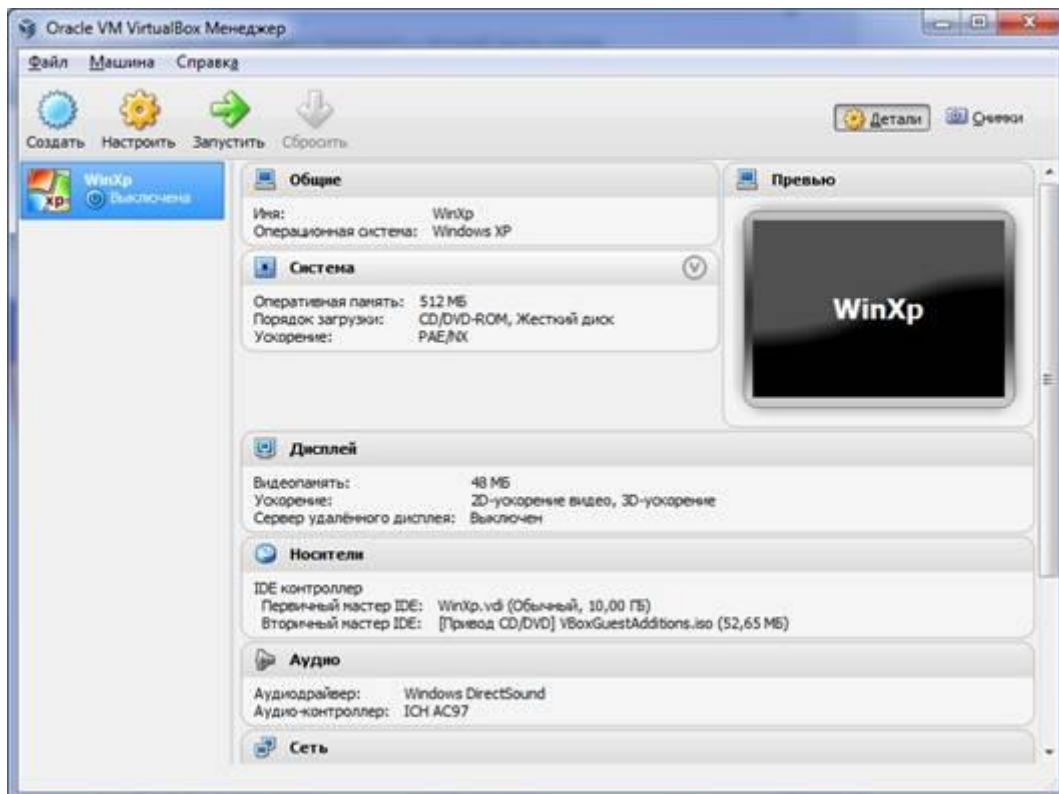
9) Нажимаем на вкладку «Дисплей», ставим галочки напротив: «Включить 3D-ускорение» и «Включить 2D-ускорение видео».



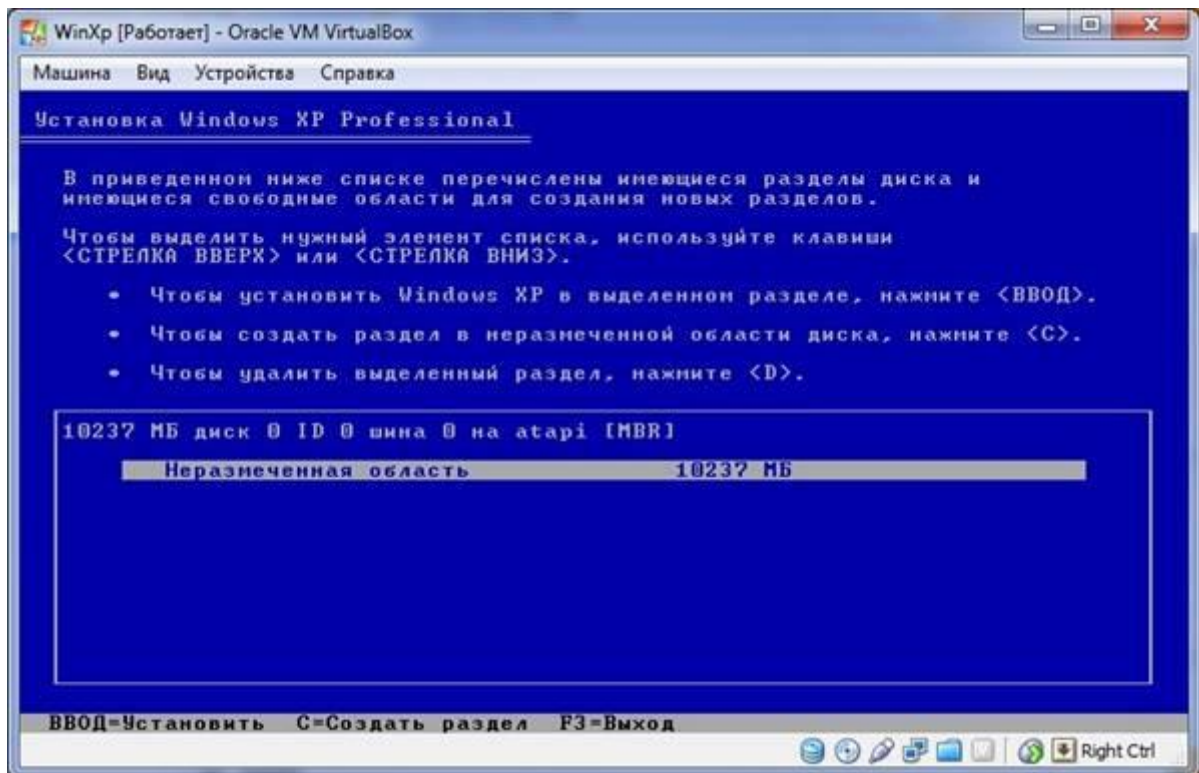
10) Заходим на вкладку «Носители». Добавляем новое устройство «Добавить привод оптических дисков» и выбираем образ системы которую устанавливаем.



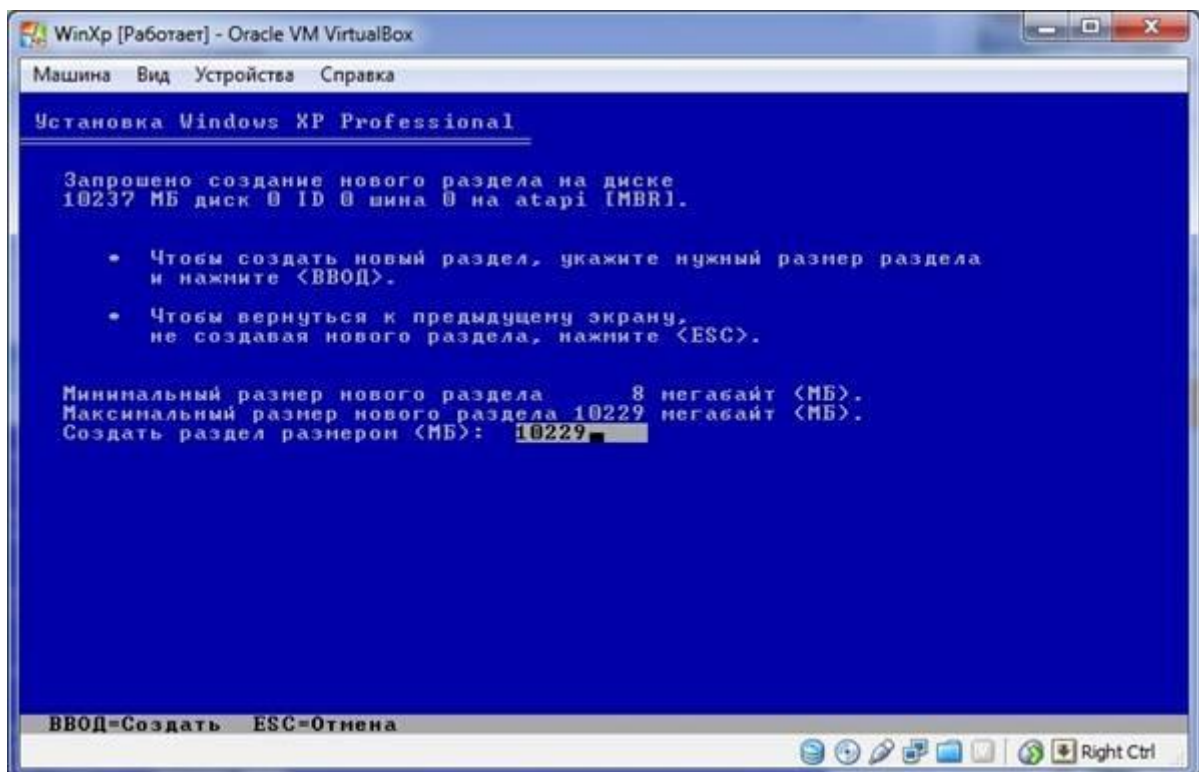
Машина готова к запуску нажимаем «Запустить».



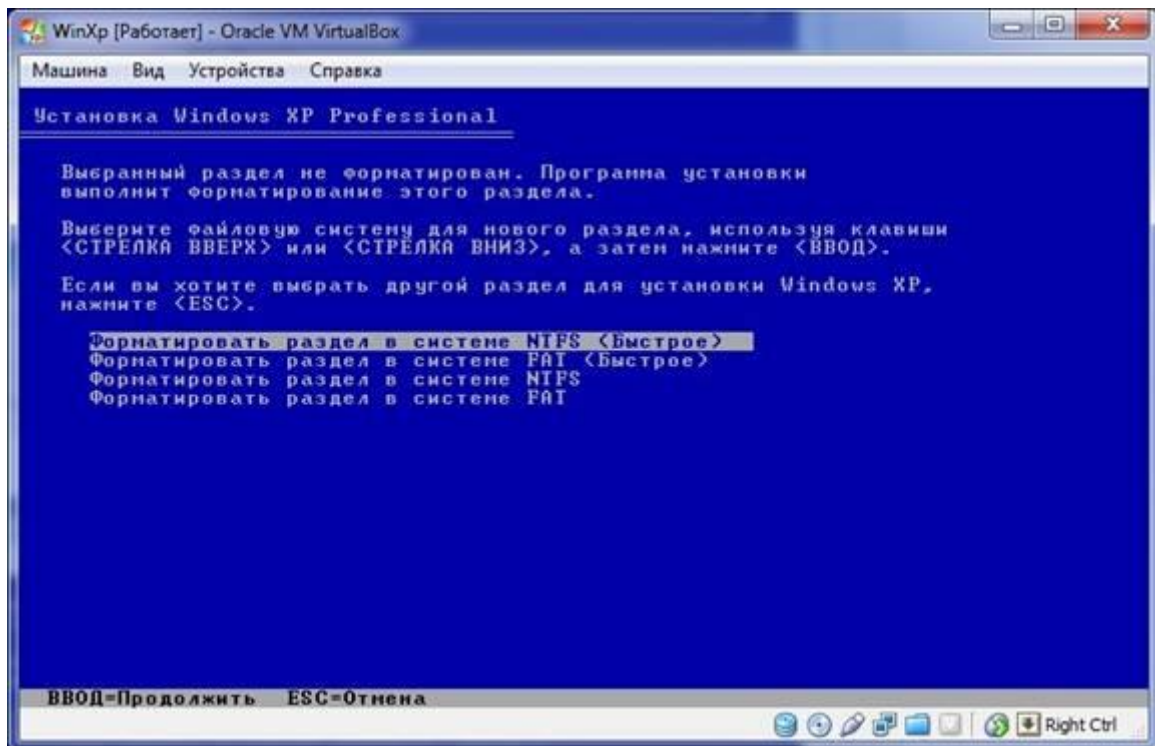
11) Запускается установщик *Windows XP* нажимаем «Enter». Попадаем в мастер раздела диска, но у нас диск всего лишь 10 ГБ., поэтому жмем «C».



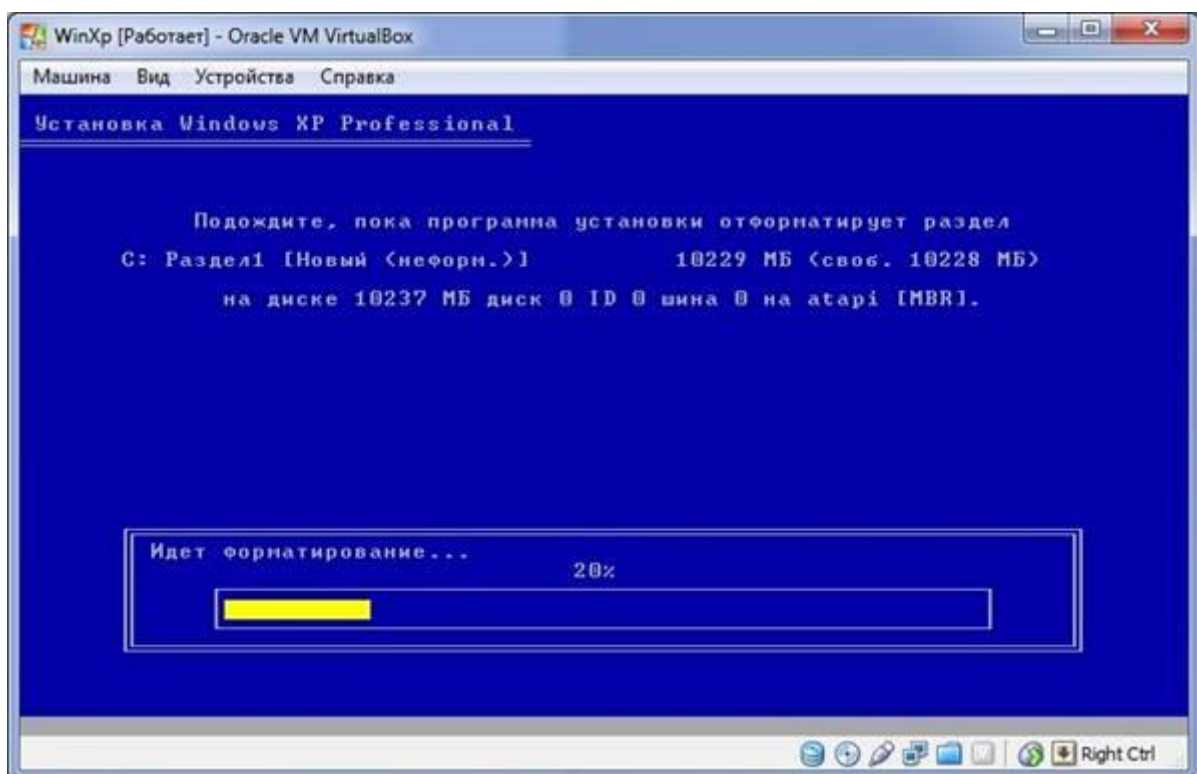
и нажимаем «Enter».



12) Выбираем курсором «Форматировать раздел в системе NTFS <Быстрое>» и жмем «Enter».

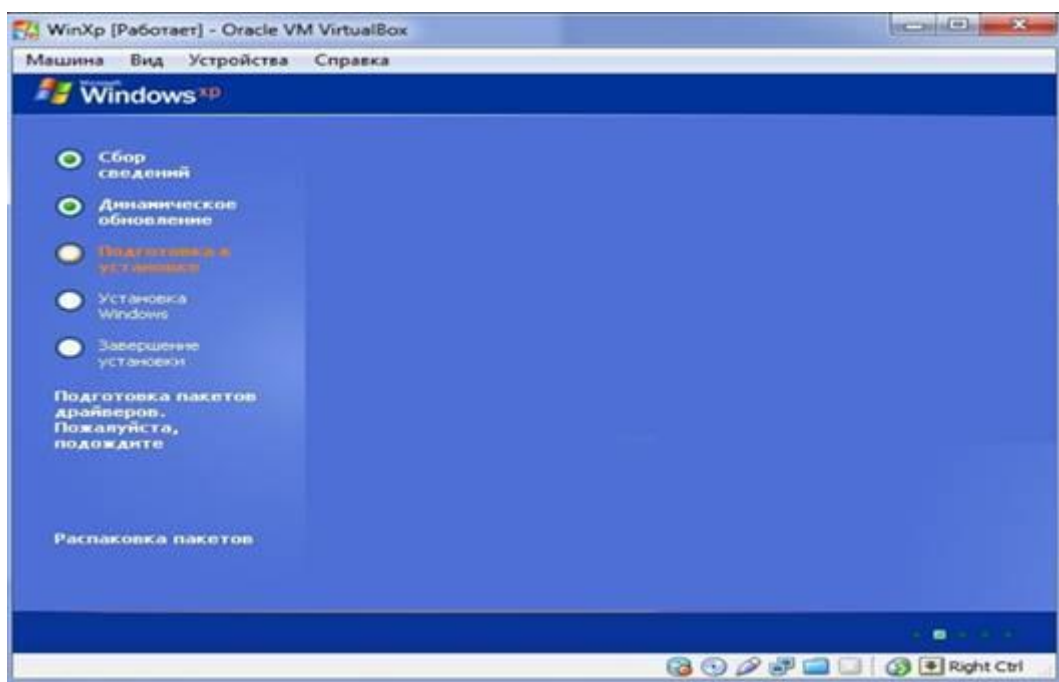


Ждем, пока идет форматирование диска.

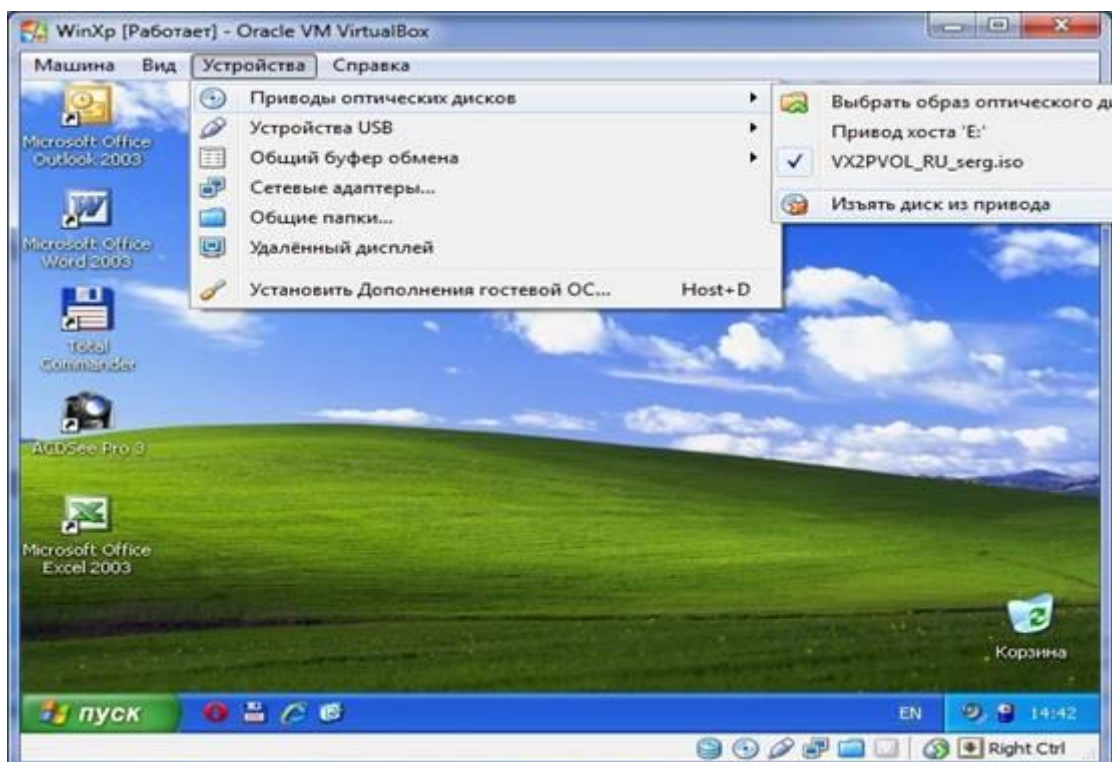


Далее установщик будет копировать ваши файлы на виртуальный жёсткий диск и после копирования перезагрузит виртуальную машину.

12) Начало установки самой *Windows XP*.



13) После завершения установки, извлечем образ диска с которого мы делали установку. Нажимаем на вкладку «Устройства» переходим на «Приводы оптических дисков» и выбираем «Изъять диск из привода».



Предоставьте преподавателю результаты выполненной работы. В отчете должны присутствовать:

1. Описание выбранной ОС, для установки на виртуальную машину.
2. Принципы установки, настройки виртуальной машины со скришотами.

3. Задать имя виртуальной машины- ФИО студента и группы.
4. Принципы установки, настройки выбранной ОС со скришотами .



