

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания
по выполнению лабораторных работ
по дисциплине «АППАРАТНЫЕ СРЕДСТВА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ»
для студентов специальности 10.05.01 Компьютерная безопасность
специализация «Информационно-аналитическая и техническая экспертиза
компьютерных систем»

Ставрополь
2026

ВВЕДЕНИЕ

Целью изучения дисциплины «Аппаратные средства вычислительной техники» является формирование общепрофессиональной компетенции студента по специальности 10.05.01 - Компьютерная безопасность.

Задачами освоения дисциплины «Аппаратные средства вычислительной техники» являются:

- изучение аппаратных средств ЭВМ различных типов, устройств ввода - вывода данных, периферийных устройств и технологических аспектов их создания;
- отслеживание тенденции дальнейшего совершенствования аппаратных средств вычислительных и телекоммуникационных систем, а также их интеграции;
- анализ технологических аспектов создания и внедрения микропроцессоров, чипсетов и микропроцессорной техники;
- исследование системных, периферийных и локальных аппаратных интерфейсов;

Учебно-методическое пособие содержит 18 практических занятий, подлежащих выполнению студентами.

Каждое практическое занятие состоит из следующих разделов:

- наименование занятия;
- цель занятия;
- общие сведения/сценарий;
- необходимые ресурсы;
- теоретический материал;
- методические пояснения и рекомендации по выполнению практического занятия;
- содержание отчета.

Учебно-методическое пособие содержит список литературы, общий для всех занятий. Методические указания выступают в качестве информационного и практического источника и могут быть использованы для очного и дистанционного обучения.

Указания по технике безопасности при выполнении практических занятий

Требования безопасности перед началом работ

- **ЗАПРЕЩАЕТСЯ:** переодеваться, пользоваться огнем, курить, принимать пищу в лаборатории;
- убедиться в отсутствии видимых повреждений на рабочем месте;

- разместить на столе тетради, учебные пособия так, чтобы они не мешали работе на компьютере;

- включение компьютера производить только после получения допуска по выполняемой работе и разрешения преподавателя или сотрудника лаборатории.

Требования безопасности во время работы

- ЗАПРЕЩАЕТСЯ: присоединять или отсоединять кабели, трогать разъемы, провода и розетки, открывать системный блок, пытаться самостоятельно устранять неисправности в работе оборудования, приносить и запускать компьютерные игры;

- выполняя практическое занятие, студенты обязаны использовать только вычислительную технику, периферийное оборудование, соединительные кабели, измерительное оборудование и носители информации, непосредственно относящиеся к данной работе;

- подключение и отключение составляющих вычислительного комплекса производить только при полном снятии напряжения со всех составляющих вычислительного комплекса;

- при обнаружении неисправностей в оборудовании немедленно отключить источники питания и доложить об этом руководителю занятий или сотруднику лаборатории;

Требования безопасности по окончании работы

- доложить руководителю занятий или сотруднику лаборатории о завершении работ;

- привести в порядок и сдать рабочее место сотруднику лаборатории, и доложить руководителю занятий о сдаче.

СОДЕРЖАНИЕ

Методические указания к лабораторным работам	5
Раздел № 1. ОСНОВЫ ПОСТРОЕНИЯ КОМПЬЮТЕРА	5
Лабораторная работа №1 Исследование внутреннего представления чисел в ЭВМ.....	5
Лабораторная работа №2 Исследование арифметических основ ЭВМ	16
Раздел № 2. ЦИФРОВОЙ ЛОГИЧЕСКИЙ УРОВЕНЬ ЭВМ	22
Лабораторная работа №3 Моделирование и исследование основ булевой алгебры и базовых логических элементов	22
Лабораторная работа №4 Моделирование и исследование комбинационных цифровых устройств	41
Раздел № 3. БАЗОВАЯ ОРГАНИЗАЦИЯ ЭВМ	54
Лабораторная работа №5 Моделирование и исследование арифметико-логического устройства	54
Лабораторная работа №6 Моделирование и исследование оперативной и постоянной памяти	58
Лабораторная работа №7 Шинная структура связи компьютера	74
Лабораторная работа №8 Организация памяти ЭВМ	81
Лабораторная работа №9 Системная плата компьютера.....	87
Раздел №4 УРОВЕНЬ МИКРОАРХИТЕКТУРЫ ЭВМ	96
Лабораторная работа №10 Принципы построения и работы микропроцессоров	96
Лабораторная работа №11 Принципы работы микропроцессоров.....	98
Лабораторная работа №12 Исследование системы машинных команд микропроцессора КР580109	
Раздел №5 ВНЕШНИЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА ЭВМ.....	125
Лабораторная работа №13 Принципы построения и работы накопителей на жестких магнитных дисках.....	125
Лабораторная работа №14 Исследование программ и утилит для тестирования и диагностирования внешних запоминающих устройств	134
Раздел №6 ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА ЭВМ.....	141
Лабораторная работа №15 Исследование видеоадаптера и жидкокристаллического монитора	141
Лабораторная работа №16 Организация межкомпьютерной связи.....	145
Лабораторная работа №17 Многопроцессорные вычислительные системы.....	157
Лабораторная работа №18 Исследование структуры персонального компьютера	167

Методические указания к лабораторным работам
Раздел № 1. ОСНОВЫ ПОСТРОЕНИЯ КОМПЬЮТЕРА
Лабораторная работа №1 Исследование внутреннего представления чисел в ЭВМ
(2 часа)

Содержание:

1. Внутреннее представление чисел в ЭВМ.

Цель работы:

Изучение арифметических основ ЭВМ, получение навыков машинной арифметики в двоичной системе счисления. Формирование умений применять знания о внутреннем представлении чисел в ЭВМ для решения практических задач получения, хранения, обработки и передачи информации, а также выработать навык выполнения машинных арифметических операций для обучения и обеспечения учебных задач обучающихся.

Теоретические сведения:

С возникновением ЭВМ возникла потребность в двоичной системе счисления, как наиболее простой и удобной для физической и технической реализации вычислений. Для более удобного переноса информации в системе человек – ЭВМ человек возникла потребность в восьмеричной и шестнадцатеричной системах счисления, т.к. перевод чисел из двоичной в восьмеричную или шестнадцатеричную системы и обратно намного проще, чем из двоичной в десятичную и обратно.

Двоичная система счисления – это способ записи чисел с помощью алфавита из 2 цифр – $\{0,1\}$. Основание двоичной системы счисления равно 2.

Восьмеричная система счисления – это способ записи чисел с помощью алфавита из 8 цифр – $\{0,1,2,3,4,5,6,7\}$. Основание восьмеричной системы счисления равно 8.

Шестнадцатеричная система счисления – это способ записи чисел с помощью алфавита из 16 цифр – $\{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$. Основание шестнадцатеричной системы счисления равно 16.

Вычисления во всех позиционных системах счисления выполняются по одинаковым правилам:

Арифметические действия над числами должны выполняться поразрядно (обычно начинают с младших разрядов).

В случае, когда результат выполнения арифметического действия над некоторым разрядом чисел превышает максимально возможное в разряде количество единиц (определяется основанием системы счисления), то разность между полученным в данном разряде результатом и основанием системы счисления записывают в этот разряд, а в старший разряд добавляют единицу переноса из данного – младшего – разряда.

В случае двоичной системы арифметические действия над числами упрощаются до минимума, что позволяет минимизировать и их физическую реализацию в ЭВМ. Например, при сложении двух двоичных разрядов возможно только 4 варианта:

$$0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1, 1 + 1 = 10$$

В последнем случае сумма дает 0 единиц и перенос в разряд десятков, т.к. имеет место переполнение разряда. Поскольку в двоичной системе нет цифры 2, и все числа записываются в алфавите из двух цифр - $\{0,1\}$, то $1+1$ – это такое количество, которое в двоичной системе обозначается как единица следующего разряда, т.е. $10_{(2)}$.

В общем виде, запись произвольного числа x в P -ичной позиционной системе счисления основывается на представлении этого числа в виде многочлена

$$x = a_n P^n + \dots + a_i P^i + \dots + a_1 P^1 + a_0 P^0 + a_{-1} P^{-1} + \dots + a_{-j} P^{-j} + \dots + a_{-m} P^{-m} \quad (1)$$

где P – основание позиционной системы счисления, a – цифры алфавита в диапазоне $\{0, 1, \dots, P-1\}$, $i = 0$, n – разряды целой части числа, $j = -1$, $-m$ – разряды дробной

Арифметические действия над числами в любой позиционной системе счисления производятся по тем же правилам, что и десятичной системе, так как все они основываются на правилах выполнения действий над соответствующими многочленами. При этом нужно только пользоваться теми таблицами сложения и умножения, которые соответствуют данному основанию P системы счисления.

При переводе чисел из десятичной системы счисления в систему с основанием $P > 1$ обычно используют следующий алгоритм:

2) Если переводится дробная часть числа, то она умножается на P , после чего целая часть запоминается и отбрасывается. Вновь полученная дробная часть умножается на P и т.д. Процедура продолжается до тех пор, пока дробная часть не станет равной нулю. Целые части выписываются после двоичной запятой в порядке их получения. Результатом может быть либо конечная, либо периодическая двоичная дробь. Поэтому, когда дробь является периодической, приходится обрывать умножение на каком-либо шаге и довольствоваться приближенной записью исходного числа в системе с основанием P .

Решение:

The diagram illustrates the conversion of two decimal numbers to their binary equivalents using the division-by-2 method.

Conversion of 115:

$$\begin{array}{r} \underline{115} \quad | \underline{2} \\ \underline{114} \quad \textcircled{1} \\ -57 \quad | \underline{2} \\ \underline{56} \quad \textcircled{1} \\ -28 \quad | \underline{2} \\ \underline{28} \quad \textcircled{0} \\ -14 \quad | \underline{2} \\ \underline{14} \quad \textcircled{0} \\ -7 \quad | \underline{2} \\ \underline{6} \quad \textcircled{1} \\ -3 \quad | \underline{2} \\ \underline{2} \quad \textcircled{1} \\ -0 \end{array}$$

The remainders are collected from bottom to top to form the binary number: $\Rightarrow 115_{(2)} = 1110011$.

Conversion of 94:

$$\begin{array}{r} 0, 94 \\ \times \quad \underline{2} \\ \underline{\textcircled{1}, 88} \\ \times \quad \underline{2} \\ \underline{\textcircled{1}, 76} \\ \times \quad \underline{2} \\ \underline{\textcircled{1}, 52} \\ \times \quad \underline{2} \\ \underline{\textcircled{1}, 04} \\ \times \quad \underline{2} \\ \underline{\textcircled{0}, 08} \\ \times \quad \underline{2} \\ \underline{\textcircled{0}, 16} \end{array}$$

The remainders are collected from bottom to top to form the binary number: $\Rightarrow 0,94_{(2)} = 111100 = 11110$.

Алгоритм перевода двоичного числа в другую систему счисления, основанием которой является степень двойки. Для этого вначале надо сгруппировать цифры двоичного

числа в группы по столько цифр, каков показатель степени. При этом в целой части группировка цифр производится справа налево, в дробной – слева направо. Если в последней группе недостает цифр, дописываются нули: в целой части – слева, в дробной – справа. Затем каждая группа заменяется соответствующей цифрой новой системы.

Пример 2. Перевести полученное число $1110011,11110_{(2)}$ из двоичной системы в шестнадцатеричную систему:

$$1110011,11110_{(2)} = \underbrace{0111}_{7_{(16)}} \underbrace{0011}_{3_{(16)}} \underbrace{1111}_{F_{(16)}} \underbrace{0000}_{0_{(16)}} = 73,F_{(16)}.$$

Алгоритм перевода чисел из системы счисления с основанием P в десятичную систему счисления

При переводе чисел из системы счисления с основанием P в десятичную систему счисления необходимо пронумеровать разряды целой части справа налево, начиная с нулевого, и в дробной части, начиная с разряда сразу после запятой слева направо (начальный номер -1). Затем вычислить сумму произведений соответствующих значений разрядов на основание системы счисления в степени, равной номеру разряда, т.е. воспользоваться выражением (1). Это и есть представление исходного числа в десятичной системе счисления.

Пример 3. Перевести полученное число $1110011,11110_{(2)}$ из двоичной системы в десятичную систему:

$$1110011,11110_{(2)} = 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} = 64 + 32 + 16 + 2 + 1 + 0,25 + 0,5 + 0,125 + 0,0625 = 115,9375_{(10)}.$$

Примечание. Очевидно, что если в каком-либо разряде стоит нуль, то соответствующее слагаемое можно опускать.

Внутреннее представление целых чисел в памяти ЭВМ.

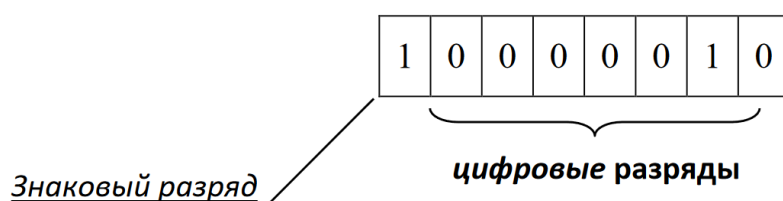
Целые числа хранятся в ЭВМ в ячейках размером 8, 16, или 32 разряда в двоичных кодах в двух формах – знаковой и беззнаковой.

В беззнаковой форме все содержимое ячейки памяти рассматривается как значащие цифры, а число в целом – как положительное натуральное. При этом младшая цифра числа хранится в младшем разряде. **Диапазон представления чисел** в беззнаковой форме равен $\angle = 2^n$, где n – количество разрядов в ячейке памяти. Беззнаковая форма позволяет хранить числа в диапазоне $[0, 2^n - 1]$, например для 8-ми разрядной ячейки от 0 до 255.

В знаковой форме старший разряд в ячейке памяти хранит знак (0 означает «+», а 1 – «-»), а все остальные разряды – цифровые. При этом диапазон чисел $[-2^{n-1}, 2^{n-1}]$, т.е. для ранее рассмотренного примера от -127 до 127.

На рис. 1 показано внутреннее представление чисел – знакового отрицательного (-2), знакового положительного 2.

а) внутреннее представление знакового числа (-2) в восьмиразрядной сетке



б) внутреннее представление знакового числа 2 в восьмиразрядной сетке

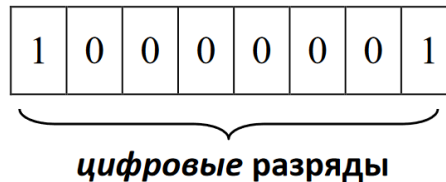


Рис. 1 – Внутреннее представление целых чисел в памяти ЭВМ

Использование дополнительных кодов.

Использование дополнительных двоичных кодов чисел позволяет свести операцию вычитания к сложению дополнительных кодов слагаемых. Это позволяет упростить конструкцию ЭВМ, т.к. вместо двух устройств для вычислений потребуется только одно.

Двоичное представление числа в отведенном для него пространстве памяти ЭВМ называется **прямым кодом числа**.

Дополнительный код – это число, дополняющее отрицательное число до переполнения ячейки памяти, где оно хранится (т.е. разрядной сетки). Чтобы получить дополнительный код отрицательного числа надо инвертировать все цифровые разряды (знаковый разряд не изменять), т.е. получить обратный код, и прибавить к полученному **обратному коду** единицу.

Обратным кодом называется инверсия прямого кода (не затрагивающая знаковый разряд).

Если число представлено в десятичном виде, то алгоритм получения дополнительного двоичного кода можно представить следующим образом:

Вычислить дополнительный код числа в десятичном виде, равный

$$X_{\text{доп}[10]} = 2^n - |X_{\text{пр}[10]}|.$$

Перевести дополнительный код числа в двоичный вид.

Пример 4. Определить дополнительный код числа -115 в 8-разрядной сетке вторым способом, и убедиться в верности полученного результата.

1) $[-115]_{\text{доп}} = 256_{(10)} - 115_{(10)} = 141_{(10)}$, тогда $[-115]_{\text{доп}} = 10001101_{(2)}$.

2) $[-115]_{\text{пр}} = 01110011_{(2)}$, $[-115]_{\text{обр}} = 10001100_{(2)}$, $[-115]_{\text{доп}} = 10001101_{(2)}$.

Для положительных чисел обратный и дополнительный коды не используются. Считается, что обратный и дополнительный коды положительного числа совпадают с его прямым кодом.

Внутреннее представление в памяти ЭВМ вещественных чисел.

Вещественные числа в памяти ЭВМ хранятся в экспоненциальной форме, а именно: виде произведения мантиссы и характеристики числа, т.е.

$$F = M \oplus E^n, \quad (2)$$

где **M** – **мантисса** (дробная часть логарифма числа), **E** – основание показательной функции, **n** – целое число или порядок.

Например, $2400000000000000 = 2,4 \cdot 10^{15}$ или $0,0000000024 = 2,4 \cdot 10^{-9}$.

Такая форма записи позволяет компактно записывать «длинные» числа, как очень большие, так и очень малые. Благодаря этому в ограниченном пространстве памяти, отводимом в ЭВМ для хранения некоторого числа, можно разместить гораздо больший диапазон чисел, чем это можно было бы сделать, если хранить числа в естественной форме.

Однако это потребует дополнительных устройств для вычислений над числами, представленными в экспоненциальной форме, т.к. алгоритмы сложения, умножения и т. д. для чисел в экспоненциальной форме иные, чем для чисел в естественной форме.

Кроме того, в экспоненциальной форме записи возможны различные варианты записи одного и того же числа, например

$$2400000000000000 = 2,4 \cdot 10^{15} = 24 \cdot 10^{14} = 0,24 \cdot 10^{16} \text{ и др.}$$

Поэтому числа в экспоненциальной форме по другому называются **числа с плавающей точкой** (можно «запятой»). Естественная форма записи вещественных чисел называется **формой представления с фиксированной точкой**.

Форма записи, в которой мантисса лежит в интервале $1 \leq \text{мантисса} < 10$ называется **нормализованной**. Именно в нормализованной форме вещественные числа хранятся в памяти ЭВМ.

Внутреннее представление вещественных чисел в памяти ЭВМ может быть различным. **В архитектуре Intel придерживаются стандарта IEEE-754** («Стандарт института электротехники и электроники для двоичной арифметики с плавающей точкой»). По этому стандарту **вещественные числа в ЭВМ могут быть 3 типов**:

- 1) **Single precision** – вещественные одинарной точности.
- 2) **Double precision** – вещественные двойной точности.
- 3) **Extended precision** – вещественные расширенной точности.

Внутреннее представление вещественных чисел по стандарту IEEE-754 приведено на рис. 2.

Характеристика, в данном случае, это величина, указывающая на порядок мантиссы, но не сам порядок. Точнее – это порядок, записанный со сдвигом таким образом, чтобы диапазон значений порядка, который можно вписать в отведенное для него пространство памяти был симметричен относительно 0. Например, для типа single для записи порядка (характеристики) отведено 8 бит что ограничивает диапазон значений порядка числом 255 (в двоичном виде 1111 1111). Чтобы сделать этот диапазон симметричным и получить возможность работать как с положительными порядками (большие числа, >1) так и с отрицательными порядками (малые числа, <1) надо сдвинуть 0 на середину диапазона. Середина диапазона – это будет число 127 (в двоичном формате 0111 1111). Это и есть характеристика для нулевого порядка. Тогда положительные порядки будут иметь характеристики $> 0111\ 1111$, т.е. будут иметь 1 в старшем разряде, а отрицательные значения порядка будут иметь в старшем разряде характеристики 0.

Таким образом, характеристика и порядок числа связаны следующим соотношением:

Характеристика = порядок + фиксированное смещение.

Величина фиксированного смещения равна:

для типа single – 127,

для типа double – 1023,

для типа extended – 16383.

По стандарту IEEE-754 нормализованной считается такая форма записи вещественного числа, в которой мантисса лежит в интервале $1 \leq \text{мантисса} < 2$, однако в памяти хранится только дробная часть мантиссы, а целая (она равна 1) добавляется при вычислениях.



где:

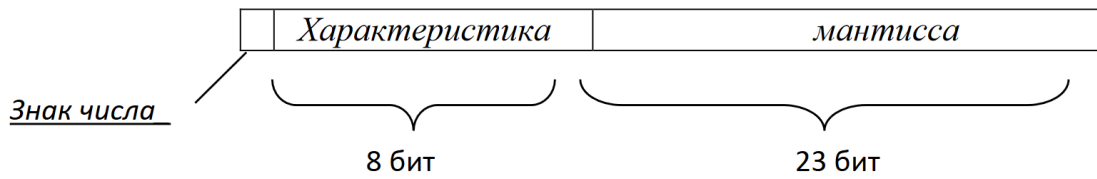
S – бит знака, если $S=0$ - положительное число; $S=1$ - отрицательное число;

E – характеристика или смещенная экспонента двоичного числа;

M – остаток мантиссы двоичного нормализованного числа с плавающей точкой.

Внутреннее представление вещественного числа типа single precision (одинарной точности).

Внутреннее представление вещественного числа типа single precision (одинарной точности)



Внутреннее представление вещественного числа типа double precision (двойной точности)



Внутреннее представление вещественного числа типа extended precision (расширенной точности)

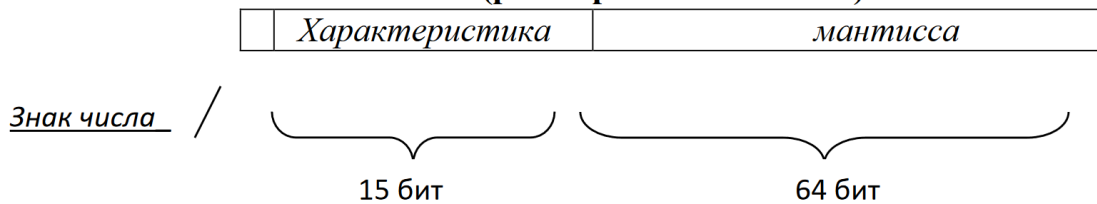


Рис. 2 – Внутреннее представление вещественных чисел в памяти ЭВМ

Пример 5. Записать внутреннее представление числа 243,625 в форме с плавающей точкой типа single. Записать это внутреннее представление этого числа в 16-ричном формате в памяти (ОП) и в процессоре.

Решение.

Запишем двоичное представление заданного числа с фиксированной точкой:

$$243,625 = 128 + 64 + 32 + 16 + 2 + 1 + 0,5 + 0,125 = 1111\ 0011,101_2$$

Теперь надо записать это число в нормализованной форме. Для этого запятую надо перенести влево на 7 разрядов и умножить полученное в результате такого переноса число на $(10_{(2)})^7$:

$$11110011,101_2 = 1,1110011101_2 \cdot (10_2)^7 = 1,1110011101 \cdot 2^{11}.$$

Теперь вычислим характеристику для порядка числа, равного 7 (111_2). Для этого надо прибавить к порядку величину смещения 127 ($0111\ 1111_2$):

$$E = 7 + 127 = 111_2 + 0111\ 1111_2 = 1000\ 0110_2 = 134_{(10)}.$$

Запишем полученное нормализованное число в разрядной сетке числа с плавающей точкой одинарной точности (типа single):



При заполнении мантиссы мы заполнили незанятые младшие разряды нулями, т.к.

более мелких долей в этом числе не содержится. А если бы были, то это бы означало, что мы ограничили точность представления этого числа десятью знаками после запятой. Максимально возможная точность для этого типа 23 двоичных знака после запятой – такое количество разрядов имеет мантисса у типа single.

Это 32-разрядное внутреннее представление вещественного числа 243,625 займет 4 байта. Запишем это 4-байтное представление в 16-ричной системе счисления:

0100 0011 0111 0011 1010 0000 0000 0000₂ = 43 73 A0 00₁₆

В оперативной памяти (ОП) это число, как и все числа, занимающие более 1 байта, хранится в перевернутом виде, т.е. сначала младшие байты затем старшие. Это сделано для более рациональной выборки данных из памяти для обработки. Сначала должны быть выбраны младшие байты, ведь вычисления начинаются с младших разрядов. Поэтому в памяти это вещественное число будет храниться в виде. Итак, внутреннее представление числа 243,625, записанное в 16-ричном формате:

в памяти ЭВМ: 00 A0 73 43,

в процессоре: 43 73 A0 00.

Преобразования числа формата 32 бит IEEE 754 в десятичное число.

Чтобы записать число в стандарте IEEE 754 или восстановить его, необходимо знать три параметра:

S – бит знака (31-й бит),

E – смещенная экспонента (30 – 23 биты),

M – остаток от мантиссы (22 – 0 биты) в десятичной системе счисления.

Это целые числа, которые записанные в числе IEEE 754 в двоичном виде.

Приведём формулу для получения десятичного числа F из чисел стандарта IEEE754:

$$F = (-1)^S \cdot 2^{(E-2^{(b-1)}+1)} \cdot (1 + M/2^n), \quad (3)$$

Проверяем наш пример 5. Поскольку исследуемое число в двоичном виде было представлено в формате одинарной точности, выражение (3) преобразуется в вид (4), с учетом того, что $b = 8$, $n = 23$:

$$F = (-1)^S \cdot 2^{(E-127)} (1 + M/2^{23}) \quad (4)$$

Тогда получаем $F = (-1)^0 \cdot 2^{(134-127)} \cdot (1 + 7577600/2^{23}) = 2^7 \cdot (1 + 0,9033203125) = 128 \cdot 1,9033203125 = 243,625$.

Вывод этой формулы приводить не буду, всё видно и так. Поясню только $(1 + M/2^{23})$ – это мантисса, единица в этой формуле – это та единица, которую мы выбросили из 23 бит, а остаток мантиссы в десятичном виде находим отношением двух целых чисел – остатка мантиссы к целому.

Особые случаи во внутреннем представлении чисел с плавающей точкой.

1) Если характеристика будет иметь значение 0000 0000 и мантисса также будет равна 0, это будет означать **ноль (со знаком)**. Если характеристика будет иметь значение 1111 1111, а мантисса будет равна 0, это будет означать **бесконечность (со знаком)**.

2) Если же характеристика будет иметь значение 1111 1111, а мантисса будет отлична от 0, это будет означать что в данной ячейке памяти находится **не число или NAN(No a Numbers)**, во всяком случае, это не число с плавающей точкой. К ним относятся символы, или результаты недопустимых операций.

3) Если характеристика будет иметь значение 0000 0000 при значении мантиссы, отличном от 0, это будет означать, что записанное в памяти **число денормализованное**, т.е. выходит за рамки заданной точности. Это числа, мантиссы которых лежат в диапазоне $0.1 \leq M < 1$. Денормализованные числа находятся ближе к нулю, чем нормализованные. Денормализованные числа как бы разбивают минимальный разряд нормализованного числа на некоторое подмножество. Сделано так потому, что в технической практике чаще

встречаются величины близкие к нулю.

Формула расчета денормализованных чисел:

$$F = (-1)^s \cdot 2^{(E-2^{(b-1)+2})} \cdot (1 + M/2^n), \quad (5)$$

Полный диапазон чисел одинарной точности (32 бит) по стандарту IEEE754 приведен на рис. 3.

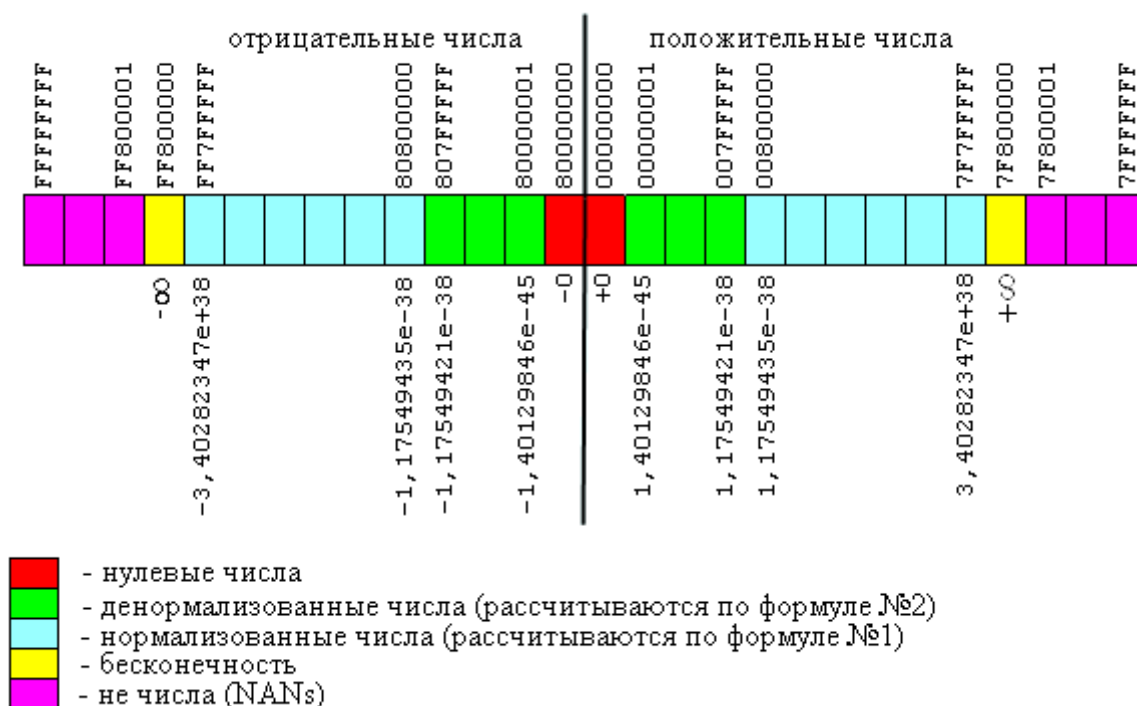


Рис. 3 – Диапазон чисел формата одинарной точности (32 бита), представленных по стандарту IEEE 754

Задание для выполнения работы:

3.1. Практикум по внутреннему представлению целых чисел в памяти ЭВМ

Задание 3.1.1. Представить в 8-ми и 16-и разрядных ячейках памяти ЭВМ целые числа в двух формах – знаковой и беззнаковой. Для этого выполните следующие операции:

1) Определить свой номер варианта N_{var} по формуле

$$N_{var} = \text{mod}(N_{spis}; 10) + 1, \quad (6)$$

где N_{spis} – номер студента по списку в журнале, mod – функция, вычисляющая остаток числа по основанию, в данном случае, 10.

2) Выписать из табл. 1 по номеру своего варианта заданные целые числа.

3) Преобразовать заданные числа в двоичный код, привести их в знаковой и беззнаковой формах внутреннего представления в памяти ЭВМ.

4) Сделайте выводы по результатам проделанной работы.

Таблица 1 – Исходные данные для задания 3.1.1.

N_{var}	Знаковая форма представления чисел		Беззнаковая форма представления чисел	
	полож. числа	отриц. числа	полож. числа	отриц. числа
1	75	-37	101	-103
	22491	-20850	23832	-18641
2	78	-56	98	-112
	18509	-28180	28882	-19070

3	67	-99	59	-121
	19835	-22248	18156	-28844
4	88	-60	74	-108
	29407	-25342	23641	-23070
5	35	-89	119	-13
	30539	-26147	22583	-28122
6	92	-44	66	-82
	17863	-25893	24255	-26686
7	77	-113	95	-27
	28658	-29614	31014	-24013
8	68	-96	100	-50
	27898	-24268	19518	-16334
9	39	-77	119	-83
	18346	-25688	31397	-21029
10	110	-120	80	-70
	30147	-29765	27439	-30000

3.2. Практикум по переводу вещественных чисел из десятичной системы счисления в двоичную и шестнадцатеричную систему.

Задание 3.2.1. Перевести заданные вещественные числа из десятичной системы счисления в двоичную систему в нормализованной форме, затем в шестнадцатеричную систему. Для этого выполните следующие операции:

- 1) Выписать из таблицы 2 по номеру своего варианта заданные вещественные числа.

Таблица 2 – Исходные данные для задания 3.2.1.

N _{var}	Десятичные числа	
1	340,125	0,055
2	277,055	0,075
3	318,195	0,026
4	299,36	0,082
5	404,67	0,095
6	283,725	0,078
7	326,82	0,079
8	500,065	0,047
9	379,725	0,068
10	257,47	0,075

- 2) Представить заданное вещественное число в двоичном коде с фиксированной точкой.

- 3) Преобразовать его в нормализованный вид с плавающей точкой.

- 4) Записать полученное нормализованное число в разрядной сетке числа с плавающей точкой одинарной точности (типа single) в форме, приведенной на рис. 2:

записать бит знака числа;

записать двоичный код характеристики числа;

записать двоичный код мантииссы.

- 5) Записать полученное двоичное представление в 16-ричной системе счисления.

- 6) Записать представление вещественного числа в 16-ричном формате в форме, предназначенной для хранения чисел в оперативной памяти ЭВМ.

- 7) Выполнить пункты 2 – 6 для других заданных вещественных чисел.

- 8) Сделайте выводы по результатам проделанной работы.

3.3. Практикум по переводу вещественных чисел из шестнадцатеричной системы счисления в двоичную и десятичную систему.

Задание 3.3.1. Перевести заданные вещественные числа из шестнадцатеричной системы счисления в двоичную систему и затем в десятичную систему. Для этого выполните

следующие операции:

1) Выписать из таблицы 3 по номеру своего варианта заданные вещественные числа в шестнадцатеричном коде.

Таблица 3 – Исходные данные для задания 3.3.1.

N _{var}	Шестнадцатеричные числа	
1	43 73 F0 00	C2 85 10 00
2	42 A7 B0 00	C3 19 D0 00
3	43 61 C0 00	C2 B4 30 00
4	42 95 70 00	C3 57 F0 00
5	43 69 A0 00	C2 88 10 00
6	42 E1 70 00	C3 D9 E0 00
7	43 CB 90 00	C2 F1 20 00
8	42 D2 50 00	C3 77 20 00
9	43 6D 60 00	C2 90 F0 00
10	42 C4 80 00	C3 6A B0 00

- 2) Перевести число из шестнадцатеричного кода в двоичный код.
- 3) Записать полученное двоичное число в стандарте IEEE 754.
- 4) Определить параметры: S, E и M.
- 5) Перевести двоичное число в десятичный вид, воспользовавшись формулой (3).
- 6) Повторить пункты 2 – 5 для другого числа.
- 7) Преобразовать числа двоичного нормализованного формата, полученные в предыдущем задании, в десятичный вид.
- 8) Сравнить полученные десятичные вещественные числа с исходными числами, приведенными в таблице 2, и оценить погрешность.
- 9) Сделайте выводы по результатам проделанной работы.

3.4. Анализ особых случаев представления чисел с плавающей точкой в стандарте IEEE 754.

Задание 3.4.1. Провести анализ особых случаев представления чисел с плавающей точкой в стандарте IEEE 754. Для этого выполните следующие операции:

- 1) Определить десятичные значения чисел, представленных в шестнадцатеричном коде 00 00 00 00₍₁₆₎ и 80 00 00 00₍₁₆₎.
- 2) Определить десятичные значения чисел, представленных в шестнадцатеричном коде 7F 80 00 00₍₁₆₎ и FF 80 00 00₍₁₆₎.
- 3) Определить десятичные значения чисел, представленных в шестнадцатеричном коде 7F 80 00 01₍₁₆₎, 7F FF FF FF₍₁₆₎, FF 80 00 01, FF FF FF FF₍₁₆₎.
- 4) Сделайте выводы по результатам проделанной работы.

3.5. Определение границ диапазона для чисел одинарной точности стандарта IEEE 754.

Задание 3.5.1. Определить минимальные и максимальные значения нормализованных чисел одинарной точности в стандарте IEEE 754. Для этого выполните следующие операции:

- 1) Записать в двоичном коде минимальное нормализованное число одинарной точности в формате стандарта IEEE 754.
- 2) Преобразовать данное нормализованное минимальное число в шестнадцатеричный код.
- 3) По формуле (4) вычислить минимальное нормализованное число в десятичном виде.
- 4) Записать в двоичном коде максимальное нормализованное число одинарной точности в формате стандарта IEEE 754.
- 5) Преобразовать данное нормализованное максимальное число в шестнадцатеричный код.
- 6) По формуле (4) вычислить максимальное нормализованное число в десятичном

виде.

7) Повторить выполнения пунктов 1 – 6 для оценки границ диапазона отрицательных нормализованных чисел.

8) Сделайте выводы по результатам проделанной работы.

Задание 3.5.2. Определить минимальные и максимальные значения денормализованных чисел одинарной точности в стандарте IEEE 754. Для этого выполните следующие операции:

1) Записать в двоичном коде минимальное денормализованное число одинарной точности в формате стандарта IEEE 754.

2) Преобразовать данное денормализованное минимальное число в шестнадцатеричный код.

3) По формуле (5) вычислить минимальное денормализованное число в десятичном виде.

4) Записать в двоичном коде максимальное денормализованное число одинарной точности в формате стандарта IEEE 754.

5) Преобразовать данное денормализованное максимальное число в шестнадцатеричный код.

6) По формуле (5) вычислить максимальное денормализованное число в десятичном виде.

7) Повторить выполнения пунктов 1 – 6 для оценки границ диапазона отрицательных нормализованных чисел.

8) Сделайте выводы по результатам проделанной работы.

Контрольные вопросы:

- 1) Определение системы счисления.
- 2) Общие правила вычисления во всех позиционных системах счисления.
- 3) Правило записи произвольного числа в Р-ичной позиционной системе счисления.
- 4) Порядок перевода чисел из одной позиционной системы счисления в другую.
- 5) Алгоритм перевода двоичного числа в другую систему счисления, основанием которой является степень двойки.
- 6) Алгоритм перевода чисел из системы счисления с основанием Р в десятичную систему счисления.
- 7) Принципы внутреннего представления целых чисел в памяти ЭВМ.
- 8) Порядок использования дополнительных и обратных кодов.
- 9) Алгоритм получения дополнительного двоичного кода.
- 10) Принципы внутреннего представления в памяти ЭВМ вещественных чисел.
- 11) Типы представления вещественных чисел в ЭВМ.
- 12) Принципы внутреннего представления вещественных чисел по стандарту IEEE-754.
- 13) Порядок преобразования чисел формата IEEE 754 в десятичное число.
- 14) Характеристика особых случаев во внутреннем представлении чисел с плавающей точкой.
- 15) Формулы расчета нормализованных и денормализованных чисел.
- 16) Характеристика диапазона чисел формата одинарной точности (32 бита), представленных по стандарту IEEE 754.
- 17) Порядок определения границ диапазона для чисел одинарной точности стандарта IEEE 754.

Лабораторная работа №2 Исследование арифметических основ ЭВМ (2 часа)

Содержание:

1. Арифметика машинных кодов в ЭВМ.

Цель работы:

Изучение арифметических основ ЭВМ, получение навыков машинной арифметики в двоичной системе счисления. Формирование умений применять знания о внутреннем представлении чисел в ЭВМ для решения практических задач получения, хранения, обработки и передачи информации, а также выработать навык выполнения машинных арифметических операций для обучения и обеспечения учебных задач обучающихся.

Теоретические сведения:

Для выполнения арифметических операций в системе счисления с основанием P необходимо иметь соответствующие таблицы сложения и умножения. Для $P = 2, 8$ и 16 таблицы представлены ниже.

Таблица 4 – Таблицы сложения и умножения для двоичной системы счисления

+	0	1	×	0	1
0	0	1	0	0	0
1	1	10	1	0	1

Таблица 5 – Таблицы сложения и умножения для восьмеричной системы счисления

+	0	1	2	3	4	5	6	7	×	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7	0	0	0	0	0	0	0	0	0
1	1	2	3	4	5	6	7	10	1	0	1	2	3	4	5	6	7
2	2	3	4	5	6	7	10	11	2	0	2	4	6	10	12	14	16
3	3	4	5	6	7	10	11	12	3	0	3	6	11	14	17	22	25
4	4	5	6	7	10	11	12	13	4	0	4	10	14	20	24	30	34
5	5	6	7	10	11	12	13	14	5	0	5	12	17	24	31	36	43
6	6	7	10	11	12	13	14	15	6	0	6	14	22	30	36	44	52
7	7	10	11	12	13	14	15	16	7	0	7	16	25	34	43	52	61

Таблица 5 – Таблицы сложения и умножения для шестнадцатеричной системы счисления

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	×	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	2	0	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	3	0	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	4	0	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	5	0	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	6	0	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	7	0	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	8	0	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	9	0	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	A	0	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	B	0	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	C	0	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	D	0	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	E	0	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	F	0	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

Все операции в ЭВМ выполняются над числами, представленными специальными машинными кодами. Их использование позволяет обрабатывать знаковые разряды чисел так же, как и значащие разряды, а также заменять операцию вычитания операцией сложения.

Различают прямой код (П), обратный код (ОК) и дополнительный код (ДК) двоичных чисел. Они подробно рассмотрены в 1-м учебном вопросе.

Арифметические действия в машинных кодах

Сложение (вычитание). Операция вычитания приводится к операции сложения путем преобразования чисел в обратный или дополнительный код согласно таблице 6.

Таблица 6 – Схема приведения операции вычитания к операции сложения

Требуемая операция	Необходимое преобразование
$A+B$	$A+B$
$A-B$	$A+(-B)$
$-A+B$	$(-A)+B$
$-A-B$	$(-A)+(-B)$

Здесь A и B неотрицательные числа. Скобки в представленных выражениях указывают на замену операции вычитания операцией сложения с обратным или дополнительным кодом соответствующего числа. Сложение двоичных чисел осуществляется последовательно, поразрядно в соответствии с таблицей.

При выполнении сложения цифр необходимо соблюдать следующие правила:

Слагаемые должны иметь одинаковое число разрядов. Для выравнивания разрядной сетки слагаемых можно дописывать незначащие нули слева к целой части числа и незначащие нули справа к дробной части числа.

Знаковые разряды участвуют в сложении так же, как и значащие.

Необходимые преобразования кодов производятся с изменением знаков чисел. Приписанные незначащие нули изменяют свое значение при преобразованиях по общему правилу.

При преобразовании единицы переноса из старшего знакового разряда, в случае использования ОК, эта единица складывается с младшим числовым разрядом. При использовании ДК единица переноса теряется. Знак результата формируется автоматически, результат представляется в том коде, в котором представлены исходные слагаемые.

Пример 1. Сложить два числа: $A_{10}=7$, $B_{10}=16$.

$A_2 = +111 = 0|111$; $B_2 = +10000 = 0|10000$.

Исходные числа имеют различную разрядность, необходимо провести выравнивание разрядной сетки:

$[A_2]_П = [A_2]_{ОК} = [A_2]_{ДК} = 0|00111$; $[B_2]_П = [B_2]_{ОК} = [B_2]_{ДК} = 0|10000$.

Сложение в обратном или дополнительном коде дает один и тот же результат:

$0|00111$

$+0|10000$

$C_2 = 0|10111$

$C_{10} = +23$

Пример 2. Сложить два числа: $A_{10} = +16$, $B_{10} = -7$ в ОК и ДК.

По таблице необходимо преобразование $A+(-B)$, в которой второй член преобразуется с учетом знака.

$[A_2]_П = [A_2]_{ОК} = [A_2]_{ДК} = 0|10000$;

$[B_2]_П = 1|111 = 1|00111$; $[B_2]_{ОК} = 1|11000$; $[B_2]_{ДК} = 1|11001$

$$\begin{array}{r}
 \text{Сложение в ОК} \\
 [A_2]_{\text{ок}} = 0|10000 \\
 + [B_2]_{\text{ок}} = 1|11000 \\
 \hline
 \textcolor{red}{1}0|01000 \\
 + \backslash \text{-----} 1 \\
 \hline
 0|01001 \\
 C_2=0|01001_2 \text{ ® } C_{10}=+9
 \end{array}$$

$$\begin{array}{r}
 \text{Сложение в ДК} \\
 [A_2]_{\text{ок}} = 0|10000 \\
 + [B_2]_{\text{ок}} = 1|11001 \\
 \hline
 \textcolor{red}{1}0|01001 \\
 C_2=0|01001_2 \text{ ® } C_{10}=+9
 \end{array}$$

При сложении чисел в ОК и ДК были получены переносы в знаковый разряд и из знакового разряда. В случае ОК перенос из знакового разряда требует дополнительного прибавления единицы младшего разряда (п.4 правил). В случае ДК этот перенос игнорируется.

Умножение и деление двоичных чисел производится в прямом коде методом сдвига и сложения или вычитания.

Пример 3. Умножить два числа, представленных в двоичном коде:

$$11_{10} = 1011_2 \text{ и } 6_{10} = 110_2$$

Выполним операцию их умножения:

$$\begin{array}{r}
 1011 \\
 \times \quad 110 \\
 \hline
 0000 \\
 1011 \\
 1011 \\
 \hline
 100010
 \end{array}$$

Из данной процедуры видно, что при умножении на разряд множимое либо повторяется, либо записываются нули. В данном случае была произведена операция умножения $11 \times 6 = 66$.

Итак, операция умножения в двоичных кодах очень проста: при наличии «1» в младшем разряде множителя множимое записывается в накапливающий сумматор, в противном случае, если разряд множителя содержит «0», то в сумматор заносятся нули. Затем производится сдвиг множимого влево на один Разряд. Далее процедура повторяется. Через количество тактов, равное количеству разрядов множителя в сумматоре накапливается результат произведения.

Для повышения быстродействия операции умножения применяют табличные методы, в которых таблицы умножения записываются в ПЗУ. При подачи на адресные входы ПЗУ кодов операндов A_1 и A_2 из соответствующей ячейки ПЗУ считываются коды произведения.

При умножении, как впрочем, и делении чисел со знаками для получения знака результата произведения или деления знаковые разряды суммируются. Если в знаковом разряде находится единица, то результат представляет собой отрицательное число, а если — ноль, то положительное число.

Процедура операции деления сложнее операции умножения.

Пример 4. Разделить число 56 (двоичный код 111000) на 8 (двоичный код 1000).

$$\begin{array}{r}
 \underline{111000} | 1000 \\
 1000 \quad 111 \\
 \hline
 1100 \\
 - 1000 \\
 \hline
 1000 \\
 - 1000 \\
 \hline
 0000
 \end{array}$$

Как видим, в результате получилось число 111 (т.е. 7). Алгоритм деления выглядит так. Из старших разрядов делимого вычитается делитель, если разность больше нуля, в результат заносится «1» делитель сдвигается на один разряд вправо (если разность меньше нуля, в результат заносится «0»), В принципе (в отличие от умножения), этот процесс может продолжаться бесконечно (вспомним бесконечные дроби). Но в реальной аппаратуре процесс ограничивается, в ущерб точности.

Совершенно по-другому выглядит процедура умножения (деления) чисел плавающей точкой.

Напомню из первого вопроса, что нормализованное двоичное число в стандарте IEEE 754 представлено в следующем виде:

$$(-1)^S \cdot 1, M \cdot \exp_2^{(p+127)},$$

где M – бинарное, p – порядок числа.

Таким образом, процедура умножения (деления) операндов нормализованных двоичных чисел включает следующие операции:

суммирование знаковых разрядов для определения знака произведения (частного);

суммирование порядков сомножителей (вычитание порядка делителя из порядка делимого);

вычисление произведения мантисс сомножителей (частного мантисс делимого и делителя) с последующей нормализацией произведения (частного) путем переноса запятой (плавающей точки);

суммирование к величине порядка произведения (частного) числа, равного числу разрядов, на которые производился перенос занятой, при этом если запятая сдвигалась влево, то число будет положительное, а если – вправо, то число будет отрицательное.

Пример 5. Умножить число $A = 1,5 \cdot 10^0$ на число $B = -5 \cdot 10^{-1}$.

$A_{(10)} = 1,5 \text{ ® } A_{(2)} = 1,1 \cdot e_2^0$, т.е. $P_A = 0$, тогда $E_A = 127 + 0 = 127$,

$B_{(10)} = -0,5 \text{ ® } A_{(2)} = 0,1 = 1,0 \cdot e_2^{-1}$, т.е. $P_B = -1$, тогда $E_B = 127 - 1 = 126$,

Итак, числа A и B в двоичном нормализованном виде одинарной точности будут иметь вид:

S	E	M
$A_{(2)} = 0$	$ 0111 \ 1111 $	$1,1000.$
$B_{(2)} = 1$	$ 0111 \ 1110 $	$1,0000.$

Выполняем процедуру умножения. Для этого суммируя знаковые разряды, получаем $S = 1$. Далее суммируем порядки $P_A + P_B = 0 + (-1) = -1$; и определяем характеристику произведения $E_C = 127 - 1 = 126$.

Умножаем мантиссы $1,1 \cdot 1,0 = 1,1$. В конечном итоге получаем:

$C_{(2)} = 1 | 0111 \ 1110 | 1,1000.$

Преобразуем произведение в десятичный вид, воспользовавшись выражением (4). С учетом того, что $S_C = 1$, $E_C = 126$, $b = 8$, $n = 23$, $M = 2^{22}$, получаем

$$C_{(10)} = (-1)^1 \cdot 2^{(126-127)} \cdot (1 + 2^{22}/2^{23}) = -2^{-1} \cdot (1+0,5) = -0,5 \cdot 1,5 = -0,75.$$

Пример 6. Разделить число $B = -5 \cdot 10^{-1}$ на число $A = 1,5 \cdot 10^0$.

Из предыдущего примера установлено, что числа B и A в двоичном нормализованном виде одинарной точности имеют вид:

S	E	M
$B_{(2)} = 1$	$ 0111 \ 1110 $	$1,0000.$
$A_{(2)} = 0$	$ 0111 \ 1111 $	$1,1000.$

Выполняем процедуру деления. Для этого суммируя знаковые разряды, получаем $S = 1$. Далее вычитаем из порядка делимого порядок делителя, т.е.

$P_B - P_A = -1 - 0 = -1$; и определяем характеристику частного $E_C = 127 - 1 = 126$.

Делим мантиссы $1,0 / 1,1 = 0,101010101\dots$, т.е. получаем бесконечную дробь. Путем округления и нормализации дроби мантисса частного примет вид:

$$M_C = 1,010\ 1010\ 1010\ 1010\ 1011e^{-1}.$$

Таким образом, порядок частного уменьшился на 1, т.к. запятая в мантиссе сдвинулась на один разряд вправо, следовательно, необходимо уменьшить характеристику на 1, тогда $E_C = 126 - 1 = 125$.

В конечном итоге получаем:

$$C_{(2)} = 1\ 0111\ 1101\ 1,010101010101010101011.$$

Преобразуем частное в десятичный вид, воспользовавшись выражением (4). С учетом того, что $S_C = 1$, $E_C = 125$, $b = 8$, $n = 23$, $M = 2796203$, получаем

$$C_{(10)} = (-1)^1 \cdot 2^{(125-127)} \cdot (1 + 2796203 / 2^{23}) = -2^{-2} \cdot (1 + 0,33333\dots) = -0,25 \cdot 1,33333\dots = -0,3333333\dots$$

Задание для выполнения работы:

Практикум по выполнению арифметических операций сложения и вычитания двоичных операндов.

Задание 3.6.1. Выполнить заданные арифметические операции. Для этого выполните следующие действия:

1) Выписать из таблицы 7 по номеру своего варианта исходные данные.

Таблица 7 – Исходные данные для задания 3.6.1.

N _{var}	Целые числа		Вещественные числа	
	Операнд А	Операнд В	Операнд А	Операнд В
1	123	±656	40,25	±34,12
2	435	±279	77,55	±27,05
3	652	±156	18,95	±31,19
4	396	±901	99,6	±29,3
5	223	±487	14,7	±40,67
6	543	±511	83,25	±28,72
7	683	±637	26,2	±32,8
8	339	±732	10,65	±50,06
9	263	±832	79,25	±37,72
10	780	±884	57,7	±25,4

2) Вычислить сумму (разность) двоичных операндов, имеющих знаковую форму представления в прямом коде.

3) Вычислить сумму (разность) двоичных операндов, имеющих беззнаковую форму представления в дополнительном коде.

4) Вычислить сумму (разность) двоичных операндов, представленных в виде нормализованных вещественных чисел одинарной точности в формате стандарта IEEE 754.

5) Сделайте выводы по результатам проделанной работы.

3.7. Практикум по выполнению арифметической операции умножения двоичных операндов.

Задание 3.7.1. Выполнить заданные арифметические операции. Для этого выполните следующие действия:

1) Выписать из таблицы 8 по номеру своего варианта исходные данные.

2) Вычислить произведение двоичных операндов, имеющих знаковую форму представления в прямом коде.

3) Вычислить произведение двоичных операндов, представленных в виде нормализованных вещественных чисел одинарной точности в формате стандарта IEEE 754.

4) Сделайте выводы по результатам проделанной работы.

Таблица 8 – Исходные данные для задания 3.7.1. и 3.8.1.

N _{var}	Целые числа		Вещественные числа	
	Операнд А	Операнд В	Операнд А	Операнд В
1	±12	56	4,5	-0,05
2	±43	79	-7,4	-0,75
3	±65	56	8,9	-0,02
4	±39	90	-9,6	-0,82
5	±22	48	4,7	-0,95
6	±54	11	-8,2	-0,08
7	±68	37	2,2	-0,07
8	±33	73	-1,6	-0,04
9	±26	83	-7,2	-0,06
10	±78	84	5,3	-0,01

3.8. Практикум по выполнению арифметической операции деления двоичных операндов.

Задание 3.8.1. Выполнить заданные арифметические операции. Для этого выполните следующие действия:

- 1) Выписать из таблицы 8 по номеру своего варианта исходные данные.
- 2) Вычислить частное двоичных операндов, имеющих знаковую форму представления в прямом коде.
- 3) Вычислить частное двоичных операндов, представленных в виде нормализованных вещественных чисел одинарной точности в формате стандарта IEEE 754.
- 4) Сделайте выводы по результатам проделанной работы.

Контрольные вопросы:

- 1) Правила применения таблиц сложения и умножения для двоичной, восьмеричной и шестнадцатеричной системы счисления.
- 2) Порядок сложения (вычитания) целых чисел в знаковой форме.
- 3) Порядок сложения (вычитания) целых чисел в дополнительном коде.
- 4) Порядок процедуры сложения (вычитания) операндов нормализованных двоичных чисел.
- 5) Порядок умножения целых двоичных чисел.
- 6) Порядок деления целых двоичных чисел.
- 7) Порядок процедуры умножения (деления) операндов нормализованных двоичных чисел.

Раздел № 2. ЦИФРОВОЙ ЛОГИЧЕСКИЙ УРОВЕНЬ ЭВМ

Лабораторная работа №3 Моделирование и исследование основ булевой алгебры и базовых логических элементов (2 часа)

Содержание:

1. Моделирование и исследование основ булевой алгебры и базовых логических элементов

Цель работы:

В результате выполнения крактического занятия студенты должны сформировать умение использовать знания принципов построения и работы базовых элементов компьютера для анализа и синтеза информационных систем и процес-сов, а также привить навык анализа и синтеза цифровых логических устройств ЭВМ с помощью программы схемотехнического моделирования Electronics Workbench.

Теоретические сведения:

Известно, что математической основой цифровых вычислительных устройств является двоичная арифметика, в которой используются всего два числа – 0 и 1. Выбор двоичной системы счисления диктовался требованиями простоты технической реализации самых сложных задач с использованием всего одного базового элемента – электронного ключа, который имеет два состояния: включен (замкнут) или выключен (разомкнут). В ЭВМ первое замкнутое состояние ключа принято за условный (логический) ноль, то второе разомкнутое состояние будет отражать условную (логическую) единицу (см. рис. 1).

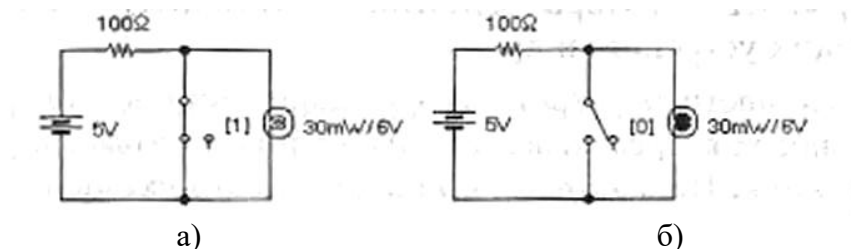


Рис. 1 – Электромеханические имитаторы: а) логической единицы, б) нуля в инверсном режиме

При нажатии на клавишу 1 индикатор гаснет, фиксируя состояние 0, и, наоборот, при отжатии клавиши индикатор загорается, фиксируя – 1. Схемы, приведенные на рис. 1, по выходному сигналу (состоянию индикаторных лампочек) обратны или инверсны состоянию ключа, поэтому такие ключи называют инверторами.

Поскольку в цифровых системах содержится огромное количество ключей (только в одном микропроцессоре их несколько миллионов) и они не могут сообщать друг другу о своем состоянии миганием лампочек, то для взаимного обмена информацией используются электрические сигналы напряжения. При этом ключи, как правило, применяются в инверсном режиме.

Теоретической основой применения логических элементов при создании разнообразных цифровых устройств комбинационного типа (сумматоры, мультиплексоры и демультиплексоры, шифраторы и дешифраторы, цифровые компараторы и др.) является булева алгебра, которая оперирует двоичными переменными, действия над которыми производятся по правилам логических операций.

Простейших логических операций три:

Отрицание (инверсия, операция НЕ). Операция отрицания выполняется над одной переменной и характеризуется следующими свойствами: функция $Y = 1$ при аргументе $X = 0$ и $Y = 0$, если $X = 1$. Обозначается отрицание чертой над переменной, с которой производится

операция: $Y = \bar{X}$ (игрек равен не икс). В программе схемотехнического моделирования Electronics Workbench (EWB) отрицание (инверсия) обозначается как $Y = X'$ (далее инверсия будет обозначаться именно таким образом).

Логическое умножение (конъюнкция, операция И). Операция логического умножения (конъюнкция) $Y = X_1 \oplus X_2$ для двух переменных выражается следующим образом: $0 \cdot 0 = 0$; $0 \oplus 1 = 0$; $1 \oplus 0 = 0$; $1 \oplus 1 = 1$, т. е. нулевое значение хотя бы одного из аргументов обеспечивает нулевой результат операции. Операция может быть распространена и на большее число переменных.

Логическое сложение (дизъюнкция, операция ИЛИ). Операция логического сложения (дизъюнкция) $Y = X_1 \vee X_2$ или $Y = X_1 + X_2$ двух переменных характеризуется следующими свойствами: $0+0 = 0$; $0+1 = 1$; $1+0 = 1$; $1+1 = 1$, т.е. единичное значение хотя бы одного из аргументов обеспечивает единичный результат операции. Дизъюнкция, как и конъюнкция, может осуществляться со многими переменными.

Более сложные логические преобразования можно свести к указанным операциям.

Совокупность различных значений переменных называют набором. Булева функция n аргументов может иметь до $N = 2^n$ наборов. Поскольку функция принимает только два значения, общее число булевых функций n аргументов равно $2^N = 2^k$, где $k = 2^n$. Таким образом, функция одного аргумента может иметь четыре значения: $Y = X$; $Y = X'$; $Y = 1$ (константа 1); $Y = 0$ (константа 0).

Два аргумента дают уже 16 значений функции, из которых, кроме названных выше трех, на практике используются еще пять:

ИСКЛЮЧАЮЩЕЕ ИЛИ, т.е. $Y = X_1 \boxplus X_2$ ($0+0 = 0$; $0+1 = 1$; $1+0 = 1$; $1+1 = 0$), по другому можно назвать функцией неравнозначности.

ИЛИ-НЕ $Y = (X_1 + X_2)'$, эта операция иначе называется стрелка Пирса ($0+0 = 1$; $0+1 = 0$; $1+0 = 0$; $1+1 = 0$), т.е. единичное значение хотя бы одного из аргументов обеспечивает нулевой результат операции.

И-НЕ $Y = (X_1 \oplus X_2)'$, эта операция иначе называется штрих Шеффера ($0 \oplus 0 = 1$; $0 \oplus 1 = 1$; $1 \oplus 0 = 1$; $1 \oplus 1 = 0$), т.е. единичное значение хотя бы одного из аргументов обеспечивает нулевой результат операции.

Запрет по X_2 , т.е. $Y = X_1 \oplus X_2'$ ($0 \oplus 0 = 0$; $0 \oplus 1 = 0$; $1 \oplus 0 = 1$; $1 \oplus 1 = 0$), т.е. только при нулевом значении аргумента X_2 и единичном значении аргумента X_1 обеспечивается единичный результат операции.

Запрет по X_1 , т.е. $Y = X_1' \oplus X_2$ ($0 \oplus 0 = 0$; $0 \oplus 1 = 1$; $1 \oplus 0 = 0$; $1 \oplus 1 = 0$), здесь наоборот только при нулевом значении аргумента X_1 и единичном значении аргумента X_2 обеспечивается единичный результат операции.

Булева алгебра базируется на нескольких аксиомах, из которых выводят основные законы для преобразований с двоичными переменными. Обоснованность выбора этих аксиом подтверждается таблицами истинности для рассмотренных операций. Каждая аксиома представлена в двух видах, что вытекает из принципа дуальности (двойственности) логических операций, согласно которому операции конъюнкции и дизъюнкции допускают взаимную замену, если одновременно поменять логическую 1 на логический 0, а 0 на 1, знак «V» (или «+») на знак « $\oplus$ », а знак « $\oplus$ » на "V" (или «+»).

Аксиомы операции отрицания: $0 = 1$; $1 = 0$.

Аксиомы операций конъюнкции и дизъюнкции:

- | | |
|--------------------------|-----|
| 1) $0 + 1 = 1$; | (a) |
| $1 + 1 = 1$; | (б) |
| 2) $1 + 0 = 0 + 1 = 0$; | (a) |
| $0 + 1 = 1 + 0 = 1$; | (б) |
| 3) $1 + 1 = 1$; | (a) |
| $0 + 0 = 0$. | (б) |

Аксиома 1б не имеет аналога в двоичной арифметике, в которой $1+1=10$ (здесь цифры и знаки имеют обычный арифметический смысл).

Законы булевой алгебры вытекают из аксиом и также имеют две формы выражения: для конъюнкции и дизъюнкции. Здесь они приводятся без доказательств. Их правильность легко проверить по таблицам истинности либо путем подстановки 0 и 1 вместо соответствующих значений переменных.

Переместительный закон:

$$X_1 \oplus X_2 = X_2 \oplus X_1; \quad (a)$$

$$X_1 + X_2 = X_2 + X_1. \quad (б)$$

Сочетательный закон:

$$X_1(X_2X_3) = (X_1X_2)X_3 = X_1X_2X_3; \quad (a)$$

$$X_1+(X_2+X_3) = (X_1+X_2)+X_3 = X_1+X_2+X_3. \quad (б)$$

Закон повторения (тавтологии):

$$X X = X; \quad (a)$$

$$X + X = X. \quad (б)$$

Закон обращения:

если $X_1 = X_2$, то $X_1' = X_2'$.

Закон двойной инверсии:

$$(X')' = X.$$

Закон нулевого множества:

$$X 0 = 0; \quad (a)$$

$$X + 0 = X. \quad (б)$$

Закон универсального множества:

$$X 1 = X; \quad (a)$$

$$X + 1 = 1. \quad (б)$$

Закон дополнительности:

$$X X' = 0; \quad (a)$$

$$X + X' = 1. \quad (б)$$

Распределительный закон:

$$X_1(X_2+X_3) = X_1X_2+X_1X_3 \quad (a)$$

$$X_1+X_2X_3 = (X_1+X_2)(X_1+X_3). \quad (б)$$

Закон поглощения:

$$X_1+X_1X_2 = X_1; \quad (a)$$

$$X_1X_2+X_1X_2' = X_1. \quad (б)$$

Закон склеивания:

$$(X_1+X_2)(X_1+X_2') = X_1; \quad (a)$$

$$X_1X_2+X_1X_2' = X_1. \quad (б)$$

Закон инверсии (закон Де Моргана):

$$(X_1X_2)' = X_1' + X_2'; \quad (a)$$

$$(X_1 + X_2)' = X_1'X_2' \quad (б)$$

В цифровой технике практические аналоги рассмотренных схем принято называть логическими элементами. Они различаются характером реализуемых функций, числом входов (по числу одновременно действующих переменных), числом выходов и другими признаками. Работа их оценивается только с точки зрения логики, без учета практического воплощения (технической базы, способа питания и т.п.).

Входы и выходы логических элементов в зависимости от уровня сигнала, при котором воспринимается или вырабатывается определенное значение двоичной переменной, подразделяются на прямые и инверсные. На прямом входе (выходе) двоичная переменная имеет значение логической 1, когда сигнал на этом входе (выходе) имеет значение, принятое за 1. На инверсном входе (выходе) двоичная переменная имеет значение 1, когда уровень сигнала на этом входе (выходе) соответствует состоянию, принятому за 0.

На логические входы можно подавать постоянные логические уровни 1 и 0 (константа 1 и константа 0) согласно законам универсального и нулевого множества. Входы, равноценные в логическом отношении (которые можно менять местами без ущерба для выполняемой функции), допускают объединение по закону повторения: при этом они действуют как один вход.

В отечественной литературе и документации логические элементы согласно ГОСТ 2.743-91 «Обозначения условные графические в схемах. Элементы цифровой техники» изображают прямоугольником (так называемое основное поле), в верхней части которого указывают символ функции: & для И, 1 для ИЛИ. Входы показывают с левой стороны прямоугольника, выходы – с правой. Допускается другая ориентация прямоугольника, при которой входы располагают сверху, а выходы снизу. Инверсные входы и выходы выделяются индикатором логического отношения небольшим кружком у вывода. Выводы питания и общий (земля) обычно не показываются. В случае необходимости шины, не несущие логической информации (в том числе питания и общие), подводят к левой или правой стороне прямоугольника и помечают звездочкой.

В цифровых устройствах логические состояния представляются двумя уровнями напряжения (потенциалов): высоким, близким к напряжению источника питания, и низким, близким к нулю. Это так называемая потенциальная система представления информации, для которой характерны непосредственная связь между отдельными элементами схемы. Длительность потенциальных сигналов определяется частотой смены информации, а переключающими импульсами служат перепады напряжения от одного уровня к другому.

Два уровня напряжения, характеризующие логические состояния, определяются просто как более высокий Н (High – высокий) и низкий L (Low – низкий). Эти два значения называют логическими уровнями. Существуют два рода так называемых логических соглашений в зависимости от того, каким уровнем напряжения кодировать логическую 1 (и соответственно логический 0). В соглашении положительной логики более высокий уровень напряжения (Н) соответствует логической 1, а низкий – логическому 0. В соглашении отрицательной логики — наоборот.

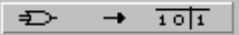
Элемент, выполняющий логические функции, можно оценивать с позиций как положительной, так и отрицательной логики. Его функциональная роль в обоих случаях будет различной. Это важное положение, которым часто пользуются на практике, вытекает из законов Де Моргана. Например, по правилам положительной логики ($H = 1$) элемент выполняет операцию И, а в отрицательной логике ($H = 0$) он действует как элемент ИЛИ, что и следует из законов Де Моргана.

С учетом сказанного элементы, выполняющие логические операции, допускается изображать на схемах в двух логически эквивалентных формах. Имея изображение логического элемента, его эквивалентную форму можно получить, проделав следующие преобразования:

- а) в основном поле изображения элемента символ операции & заменить на символ 1 либо наоборот;
- б) все прямые входы заменить инверсными, а инверсные прямыми;
- в) все прямые выходы заменить инверсными, а инверсные – прямыми.

На рис. 2 показаны обозначения базовых логических элементов, принятые в программе EWB, согласно зарубежным (американским) стандартам.

Моделирование логических схем в программе EWB целесообразно проводить с помощью логического преобразователя (Logic Converter). В качестве примера на рис. 3, а приведена схема для исследования элемента ИСКЛЮЧАЮЩЕЕ ИЛИ. Порядок подключения исследуемого элемента к логическому преобразователю очевиден из рис. 3. Очевидно также и то, что при наличии двух входов возможны только четыре комбинации входных сигналов, что отображается на экране преобразователя в виде таблицы истинности, которая

генерируется после нажатия кнопки 

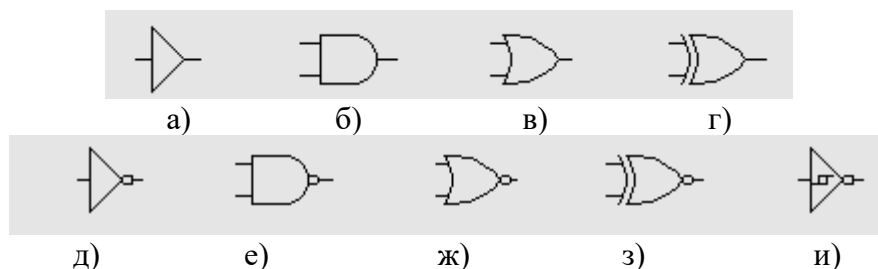


Рис. 2 – Графические обозначения логических элементов в EWB

На рис. 2 представлены:

- а) буферный логический элемент;
- б) элемент И (AND);
- в) элемент ИЛИ (OR);
- г) элемент ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR);
- д) инвертор (NOT);
- е) элемент И-НЕ (NAND);
- ж) элемент ИЛИ-НЕ (NOR);
- з) элемент ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ (XNOR), иначе равнозначность;
- и) триггер Шмита.

Для получения булева выражения исследуемого элемента необходимо нажать кнопку



. Это выражение приводится на дополнительном дисплее, расположенном в нижней части лицевой панели, в виде двух слагаемых, соответствующих выходному сигналу ИСТИНА (сигнал логической единицы на выходе OUT). Сопоставление полученного выражения с таблицей истинности убеждает нас в том, что таких комбинаций действительно две, $A'B$ и AB' , следовательно булева функция имеет вид: $A'B + AB'$, где знак + соответствует логической операции ИЛИ.

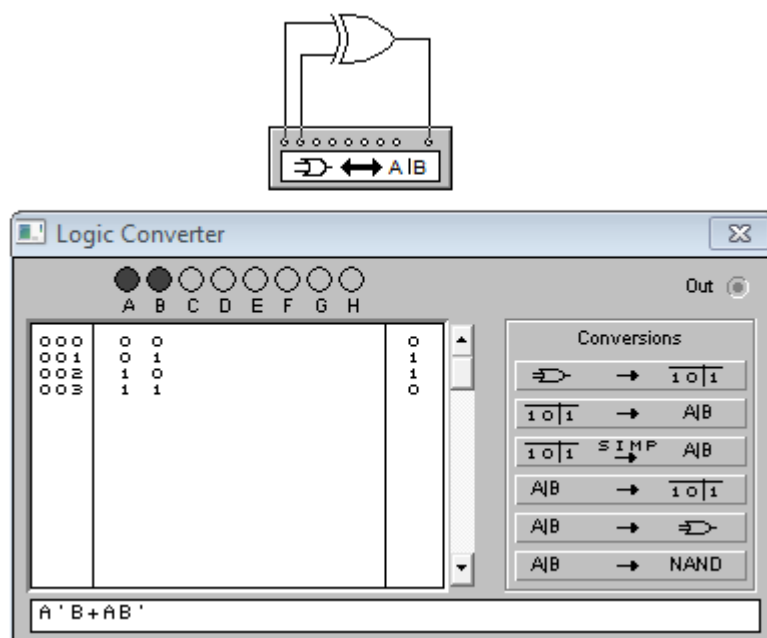


Рис. 3 – Исследование логических элементов с помощью логического преобразователя

В качестве примера использования генератора слова (Word Generator) для исследования логических элементов на рис. 4 приведена схема, в которой на входы элементов И (AND), И-НЕ (NAND), ИЛИ (OR), ИЛИ-НЕ (NOR) подается соответствующая двоичная

комбинация, а реакция элементов индицируется включенными на их выходах логическими пробниками.

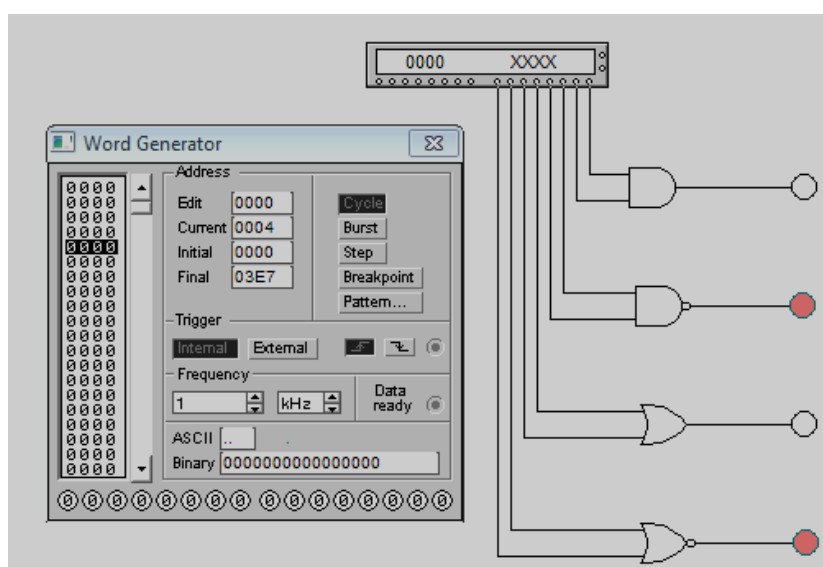


Рис. 4 – Исследование логических элементов с помощью генератора слова

С помощью логического преобразователя можно проводить не только анализ логических устройств, но и их синтез. Допустим, что нам требуется составить схему и булево выражение для логического элемента, у которого выходная комбинация в таблице истинности не 0110, как на рис. 3, а 1101. Для внесения необходимых изменений отмечаем курсором в столбце OUT подлежащий изменению символ, изменяем его с помощью клавиатуры и затем, перемещаясь по столбцу клавишами управления курсором, изменяем по необходимости символы в других строках. После внесения всех изменений последовательно нажимаем на кнопки, приведенные на рис. 5 (в правом нижнем углу), и получаем результат, представленный на рис. 5.

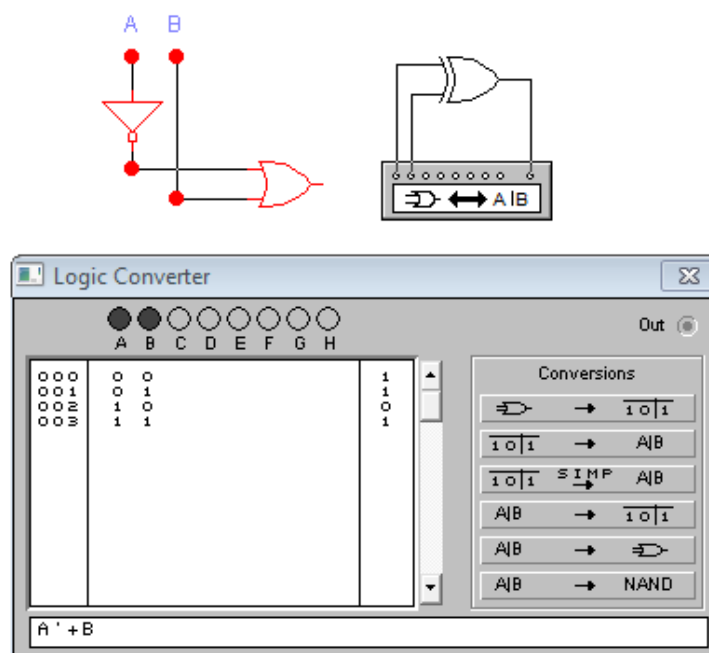


Рис. 5 – Результат синтеза логического устройства по заданной таблице истинности

Синтезированное логическое устройство показано в верхнем левом углу рис. 5, а его булево выражение $A+B$ представлено на дополнительном дисплее в нижней части Logic Converter.

В общем случае для выполнения синтеза целесообразно действовать следующим образом. Щелчком курсора по иконке логического преобразователя непосредственно на линейке приборов раскрываем его лицевую панель. Активизируем курсором клеммы-кнопки А, В, ..., Н (начиная с А), количество которых равно количеству входов синтезируемого устройства. Вносим необходимые изменения в столбец OUT и после нажатия на панели преобразователя указанных выше кнопок управления получаем результат в виде схемы на рабочем поле программы и булево выражение в дополнительном дисплее.

В заключение заметим, что для двухвходовых элементов, представленных на рис. 2, можно увеличить количество входов до восьми, открывая двойным щелчком по значку компонента диалоговое окно (см. рис. 6). По умолчанию в этом окне указано минимально возможное число входов, равное двум. Заметим также, что при оптимизации логических устройств с целью получения наиболее простой схемы используются различные методы минимизации булевых выражений (карты Карно, диаграммы Вейча и др.), однако при числе входов более пяти этот процесс становится крайне трудоемким.

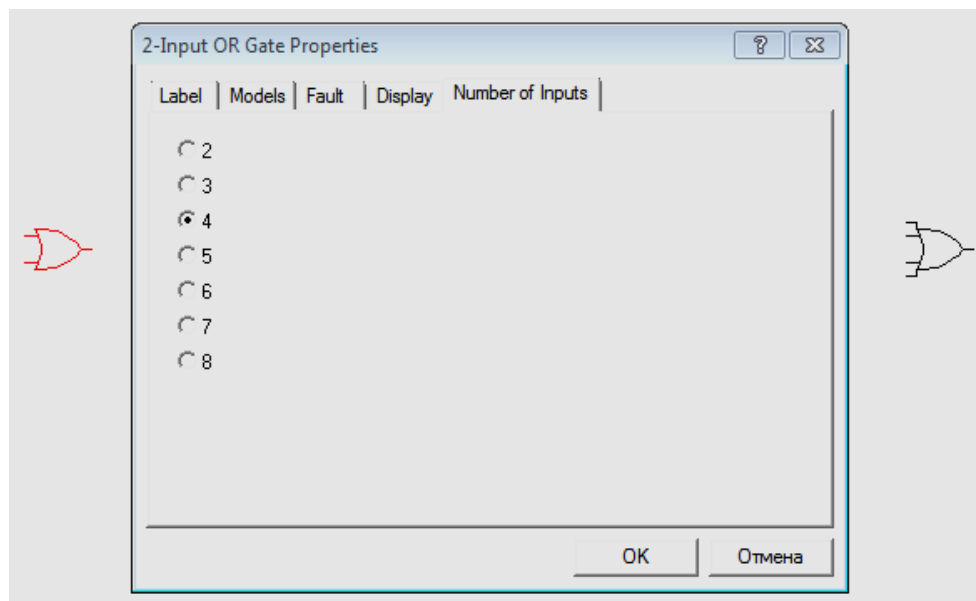


Рис. 6 – Окно установки количества входов логического элемента

Задание для выполнения работы:

3.1. Ознакомление с основами работы и интерфейсом программы Electronics Workbench

Для ознакомления с интерфейсом программы схемотехнического моделирования Electronics Workbench EWB 5.12 выполните следующие операции:

1) Скопировать папку **EWB512**, размещенную в папке **ЛР Арх ЭВМ №2**, на несистемный логический диск компьютера.

2) Открыть программу Electronics Workbench EWB 5.12, запустив из установленной папки файл **WEWB32.exe**.

Описание Electronics Workbench

Среда моделирования

После загрузки Electronics Workbench EWB 5.12 на экране появляется окно (см. рис. 7), содержащее рабочее поле, выше которого расположены:

- строка меню;
- панели: стандартная, компонентов и инструментов.

Исследуемая схема собирается на рабочем поле из имеющихся компонентов и приборов при одновременном использовании мыши и клавиатуры.

Назначение пунктов меню:

File – используется для операций с файлами;

Edit – используется для редактирования изображения на экране, позволяет скопировать выделенный фрагмент схемы в буфер обмена;

Circuit – используется для операций с выделенными частями схемы;

Analysis – используется для управления моделированием исследуемой схемы;

Window – используется для операций с окном Electronics Workbench.

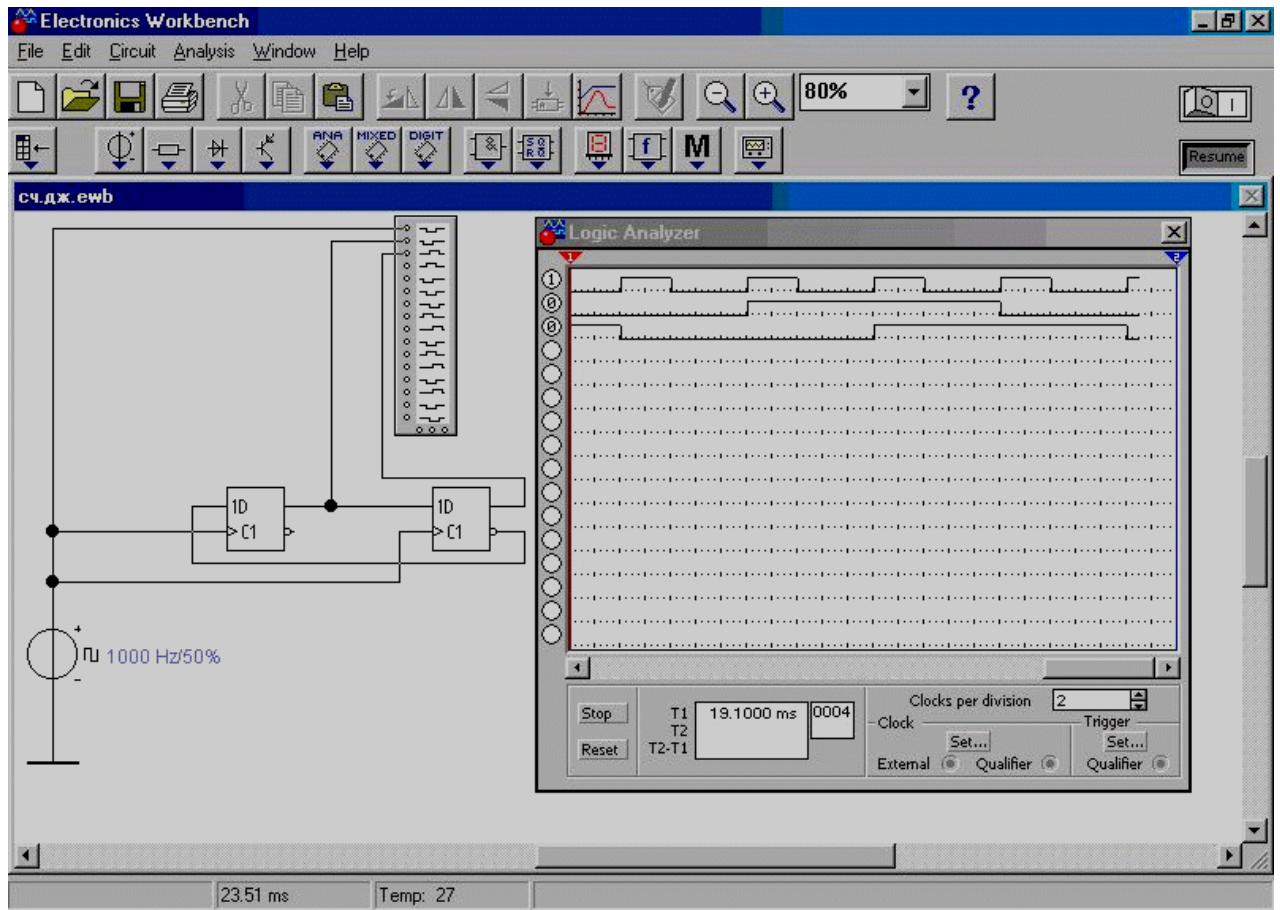


Рис. 7 – Окно программы Electronics Workbench EWB 5.12

Кнопки стандартной панели инструментов дублируют часть команд меню.

Кнопки панели компонентов объединены в группы, имеющие следующее назначение (по порядку, как показано на рисунке):

- набор наиболее часто используемых и формируемых пользователем элементов;
- основные дискретные аналоговые компоненты;
- интегральные микросхемы;
- логические и цифровые элементы;
- индикаторы, специальные и дополнительные элементы;
- измерительные приборы.

Компоненты

В данной методической разработке рассматриваются только те компоненты, которые необходимы для проведения практического занятия.

Источники (Sources) (см. рис. 8):

- кнопка выбора группы ;
- земля (Ground) – нулевая потенциальная точка (*первая кнопка слева в верхнем ряду*) может приниматься за нулевой (высокий) логический уровень;
- аккумуляторная батарея (вторая кнопка слева в верхнем ряду);
- генератор прямоугольных импульсов Clock (*первая кнопка справа в верхнем ряду*). Параметрами генератора являются частота, амплитуда, скважность (Duty cycle) – отношение длительности импульса к частоте следования;
- источник напряжения (*третья кнопка справа в верхнем ряду*) может приниматься за единичный (высокий) логический уровень.

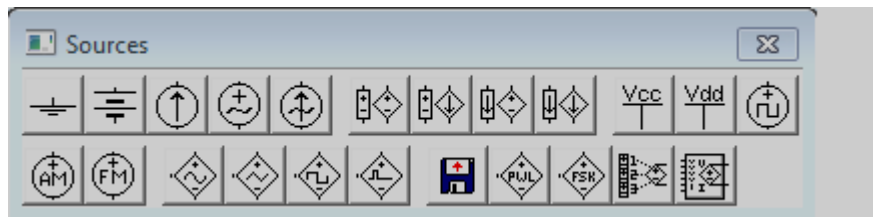


Рис. 8 – Диалоговое окно компонентов источника в EWB 5.12

Пассивные компоненты (Basic) (см. рис. 9):

- кнопка выбора группы ;
- соединительный узел (Connector), обозначаемый точкой (*первая кнопка слева в верхнем ряду*). Используется для создания электрического соединения в схеме. От коннектора может отходить до четырёх соединений под углом 90 градусов относительно друг друга.
- переключатель (Switch) (*пятая кнопка справа в верхнем ряду*)– связывается с клавишей на клавиатуре, нажатием которой в процессе моделирования осуществляется переключение ключа;
- резистор, подключенный одним выводом к источнику постоянного напряжения (Pull-Up Resistor) (*первая кнопка справа в верхнем ряду*). Задаётся сопротивлением (R) и напряжением (V). Используется в схеме для задания сигнала логической единицы;
- резистор (*вторая кнопка слева в верхнем ряду*) – задает сопротивление, используется для задания силы тока в цепи.

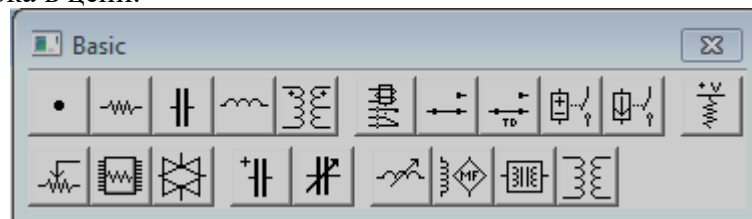


Рис. 9 – Диалоговое окно пассивных компонентов в EWB 5.12

Логические элементы (см. рис. 10):

- кнопка выбора группы ;
- кнопки элементов верхнего ряда слева на право: 2И (2 –Input AND Gate), 2ИЛИ (2-Input OR Gate), НЕ (NOT Gate), 2ИЛИ-НЕ (2-Input NOR Gate), 2И-НЕ (2-Input NAND Gate), 2 ИСКЛЮЧАЮЩЕЕ ИЛИ (2-Input XOR Gate), 2 ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ (2-Input NXOR Gate), иначе элемент равнозначности, буферный элемент с тремя состояниями на выходе, для создания шинных формирователей, буферный элемент.

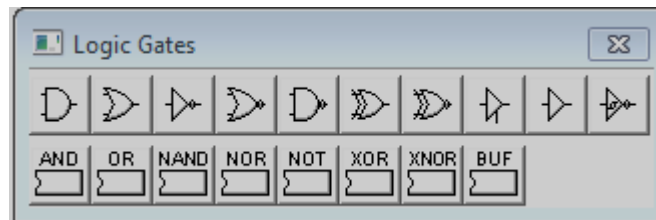


Рис. 10 – Диалоговое окно логических компонентов в EWB 5.12

Цифровые узлы (Digital) (см. рис. 11):

- кнопка выбора группы ;
- кнопки выбора элементов верхнего ряда (слева на право):
- полусумматор (Half-Adder);
- полный сумматор (Full-Adder),
- асинхронный RS-триггер (RS Flip-Flop);
- синхронный JK-триггер с прямыми асинхронными R и S входами (JK Flip-Flop With Active High Asynch Inputs);
- синхронный JK-триггер с инверсными асинхронными R и S входами (JK Flip-Flop With Active Low Asynch Inputs);
- синхронный D-триггер (D Flip-Flop);
- синхронный D-триггер с инверсными асинхронными R и S входами (D Flip-Flop With Active Low Asynch Inputs);
- кнопки выбора типов микросхем нижнего ряда (слева на право):
- кнопка выбора мультиплексоров;
- кнопка выбора дешифраторов-демультиплексоров;
- кнопка выбора кодеров;
- кнопка выбора элементов арифметико-логических устройств;
- кнопка выбора счетчиков;
- кнопка выбора регистров;
- кнопка выбора триггеров.

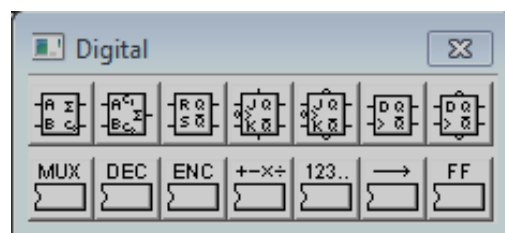


Рис. 11 – Диалоговое окно цифровых узлов в EWB 5.12

Индикаторы (Indicators) (см. рис. 12):

- кнопка выбора группы ;
- логический пробник Red Probe (четвертая кнопка слева). Светится красным светом, если на него подаётся сигнал с уровнем логической единицы. В окне Свойств цвет может быть изменён на зелёный (Green) или синий (Blue).
- семисегментный индикатор Decoded Seven-Segment Display (пятая кнопка слева);
- семисегментный индикатор с дешифратором (Decoded Seven-Segment Display), отображает шестнадцатеричные эквиваленты двоичного кода (шестая кнопка слева).

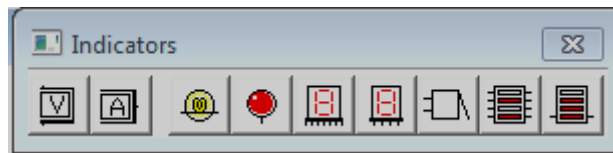


Рис. 12 – Диалоговое окно индикаторов в EWB 5.12

Измерительные приборы (Instruments) (см. рис. 13):

- Кнопка выбора группы .
- Генератор слов **Word Generator** или кодовый генератор (третья кнопка справа), используется для формирования последовательностей двоичных 16-ти разрядных слов. На схему выводится уменьшенное изображение генератора слов. Двойным щелчком по уменьшенному изображению открывается расширенное изображение генератора слов. Генератор слов подключается к схеме с помощью выводов на его уменьшенном изображении. Значения слов могут задаваться в левом окне в шестнадцатеричном виде расширенного изображения генератора, в двоичном виде в окне Binary, в виде ASCII кодов в окне ASCII. В секции Address выводятся: номер редактируемого слова, текущего слова, адреса начальной и конечной ячейки. Генератор слов может формировать последовательности в пошаговом, пакетном и циклическом режимах, которые задаются соответственно кнопками Step (шаг), Burst (пакет) и Cycle (цикл). Возможно задание точки остановки кнопкой Breakpoint. Кнопка Pattern открывает окно, позволяющее очистить буфер выводимых данных, загрузить данные из файла, сохранить их в файле с расширением .db, заполнить его комбинациями, возрастающими на 1 от 0, убывающими на 1 от FFFF, комбинациями со сдвигом 1 влево и вправо.
- Логический анализатор **Logic Analyzer** (вторая кнопка справа). Предназначен для отображения на экране монитора 16-разрядных кодовых последовательностей одновременно в 16 точках схемы. Анализатор имеет две визирные линейки 1 и 2, точные значения времени T1 и T2, а также интервала T2-T1 отображаются в окне анализатора. Синхронизация анализатора может быть внутренней (Internal) или внешней (External). Клеммы для внешней синхронизации находятся внизу изображения анализатора. Параметры синхронизации задаются в меню, вызываемом кнопками Set.
- Логический преобразователь (первая кнопка справа). На лицевой панели преобразователя представлены 8 клемм-индикаторов и одного выхода. Он содержит экран для отображения таблицы истинности исследуемой схемы, экран строка для отображения ее булева выражения в нижней части. В правой части панели расположены кнопки управления (*сверху вниз*): для отображения таблицы истинности исследуемого устройства; для определения булева выражения по таблице истинности; для минимизации булева выражения; для синтеза схемы устройства на логических элементах без ограничения; для синтеза схемы устройства только на логических элементах И-НЕ.

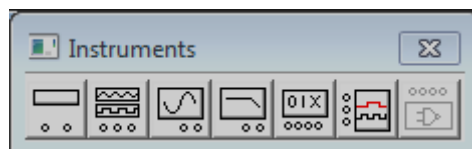


Рис. 13 – Диалоговое окно измерительных приборов в EWB 5.12

Моделирование схем

Перед анализом работы схемы выполняется её построение и подключение индикаторов или измерительных приборов. Анализ работы начинается после нажатия выключателя в правом верхнем углу окна программы (см. рис. 7).

Построение и редактирование схем

Элементы схем, после их выбора из соответствующей группы элементов, переносятся в нужное место на рабочее поле мышкой с нажатой левой кнопкой. Связи между выводами элементов также устанавливаются мышкой с нажатой левой кнопкой. После установления связи её положение может быть отредактировано после выделения мышкой. Цвет проводника можно изменить в окне его свойств, открываемом двойным щелчком на изображении проводника. Связь может быть установлена также между выводом элемента и проводником (соединением), иногда для установления соединения может потребоваться дополнительная промежуточная точка (соединитель), которая находится в группе компонентов Basic.

Элемент схем может быть выделен щелчком левой кнопки мыши, его цвет меняется на красный. Выделенный элемент можно копировать, удалить. Можно изменять его положение через меню *Circuit*, кнопки стандартной панели инструментов, а также через контекстное меню, открываемое нажатием правой клавиши мыши. Элемент можно вращать и разворачивать по горизонтальной и вертикальной осям. Через меню *Circuit* и контекстное меню можно открыть окно свойств *Component Properties* выделенного элемента. Это можно сделать также двойным щелчком левой кнопки мыши по элементу. Окно свойств позволяет редактировать значения свойств выделенного элемента. Выделенный элемент можно перемещать мышкой с нажатой левой кнопкой, связи между элементами при этом сохраняются.

Можно выделять группу элементов, находящихся в прямоугольной области, которая определяется мышкой с нажатой левой кнопкой. Выделенную группу можно копировать, удалять, перемещать мышкой, вращать элементы группы.

Можно копировать фрагмент схемы как точечный рисунок командой *Copy as Bitmap* в меню *Edit*, копирование выполняется после выбора команды выделением мышью области копирования.

После того, как схема построена, можно вставить в нее дополнительные компоненты. Для этого нужно мышью переместить компонент в требуемую точку схемы и, поместив его над проводником, отпустить кнопку мыши. Компонент автоматически вставится в цепь.

Для подключения прибора к схеме нужно мышью перетащить прибор с панели инструментов на рабочее поле и подключить выводы прибора к исследуемым точкам.

Эксперименты над цифровыми схемами

При исследовании схем используются рассмотренные ранее элементы и приборы.

В зависимости от задания схема либо собирается из перечисленных элементов и приборов, либо используется готовая из файла.

Эксперименты проводятся при пошаговом формировании сигналов и фиксации результатов с помощью логических пробников или семисегментного индикатора, или при формировании входных сигналов генератором и фиксацией результатов с помощью логического анализатора.

3.2. Анализ основных логических элементов с помощью программы схемотехнического моделирования Electronics Workbench

Задание 3.2.1. Провести анализ логического элемента НЕ (инвертора). Для этого выполните следующие операции:

1) В рабочем поле программы Electronics Workbench собрать схемы исследования, приведенные на рис. 14. В левой части рисунка приведена логическая схема исследования с помощью логического преобразователя, в правой верхней части – схема электрическая принципиальная, в которой исследование проводится с помощью осциллографа, в правой нижней части – эквивалентная электрическая схема замещения, в которой исследование проводится более упрощенно с помощью логического пробника.

2) Двойным щелчком открыть логический преобразователь.

3) Отобразить таблицу истинности инвертора, нажав на кнопку .

4) Записать в отчет таблицу истинности инвертора.

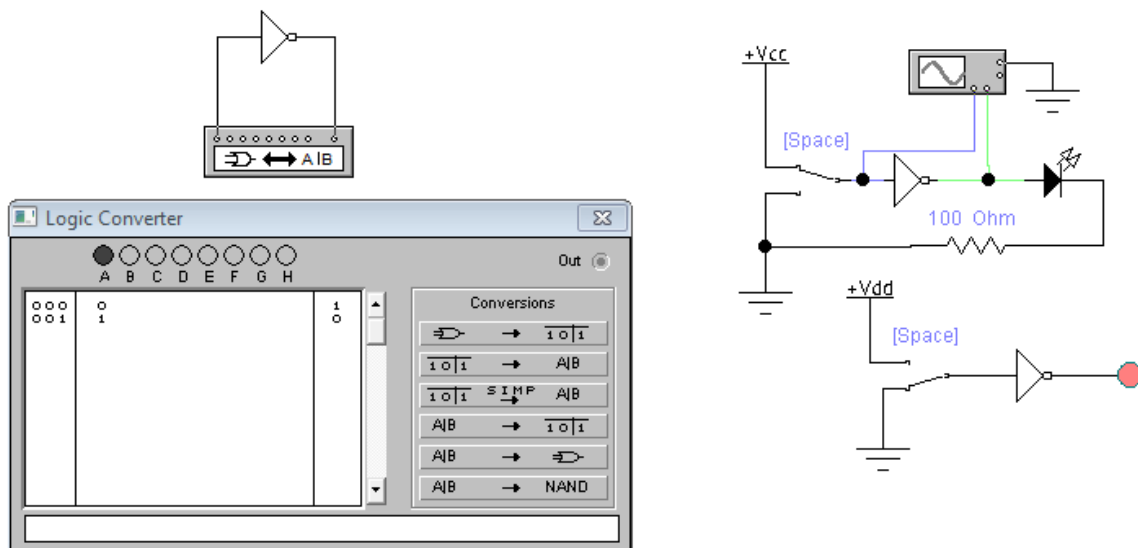


Рис. 14 – Схемы исследования инвертора

5) Двойным щелчком открыть осциллограф, затем развернуть его полностью, нажав кнопку Expand.

6) Настроить осциллограф, установив временную развертку (Time base) на 0,10 мс/дел, а вертикальные (амплитудные) развертки по обоим каналам (Channel A, Channel B) на 2V/дел.

7) Включить схему в работу, нажав выключатель в правом верхнем углу окна программы.

8) Переключая переключатель, нажатием клавиши «Пробел», с помощью осциллографа измерить и отобразить в отчете логические уровни напряжения на входе и выходе инвертора. Сравнить эти логические уровни с таблицей истинности, полученной с помощью логического преобразователя.

9) Исследовать работу инвертора с помощью логического пробника на упрощенной схеме замещения, подключая по очереди к его входу логическую единицу (источник питания) и логический нуль (нулевая шина или шина заземления). Проанализировать состояние пробника от логического уровня напряжения на выходе инвертора.

10) Сделайте выводы по результатам проделанной работы.

Задание 3.2.2. Провести анализ буферного элемента. Для этого выполните следующие операции:

1) В схемах, приведенных на рис. 14, заменить инвертор буферным элементом, для чего удалить инвертор командой Cut в контекстно-зависимом меню, вызываемом нажатием правой кнопкой мыши на УГО элемента. Открыть панель логических элементов, выбрать буферный элемент и подсоединить его к схемам.

2) Для собранных схем исследования буферного элемента повторить процедуру исследования, приведенную в пунктах 2 – 9 предыдущего задания.

3) Сделайте выводы по результатам проделанной работы.

Задание 3.2.3. Провести анализ особенностей работы буферного элемента на три состояния. Для этого выполните следующие операции:

1) В рабочем поле программы Electronics Workbench собрать схему исследования, приведенную на рис. 15.

2) Настроить генератор тактовых импульсов, установив напряжение (Voltage) – 5 В, частоту (Frequency) – 1 Гц, а скважность оставив 50%.

3) Настроить мультиметр (тестер), нажав кнопку сопротивления (Ω) и постоянного тока (--), как указано на рис. 15.

4) Настроить осциллограф, открыв его двойным щелчком и установив временную развертку (Time base) на 0,50 с/дел, а вертикальные (амплитудные) развертки по обоим

каналам (Channel A, Channel B) на 5V/дел.

5) Нажав выключатель запустить схему в работу и измерить с помощью мультиметра сопротивление цепи на выходе буферного элемента.

6) Подключить к входу буфера генератор тактовых импульсов, нажав клавишу «Пробел». В отчете отобразить эпюры напряжения с экрана осциллографа.

7) Отключить работу схемы, вернуть переключатели в исходное состояние.

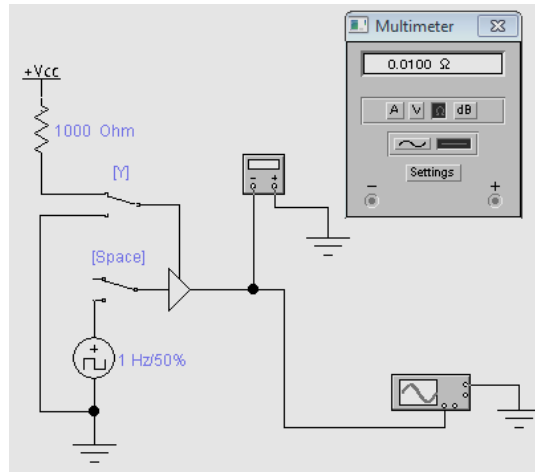


Рис. 15 – Схема исследования буферного элемента на три состояния

8) Изменить уровень логического напряжения на управляющем входе буферного элемента, для этого переключения нажать клавишу «Y». Измерить с помощью мультиметра сопротивление цепи на выходе буферного элемента.

9) Подключить к входу буфера генератор тактовых импульсов, нажав клавишу «Пробел». В отчете отобразить эпюры напряжения с экрана осциллографа.

10) Сделайте выводы по результатам проделанной работы.

Задание 3.2.4. Провести анализ логических элементов И, И-НЕ. Для этого выполните следующие операции:

1) В рабочем поле программы Electronics Workbench собрать схемы исследования, приведенные на рис. 16.

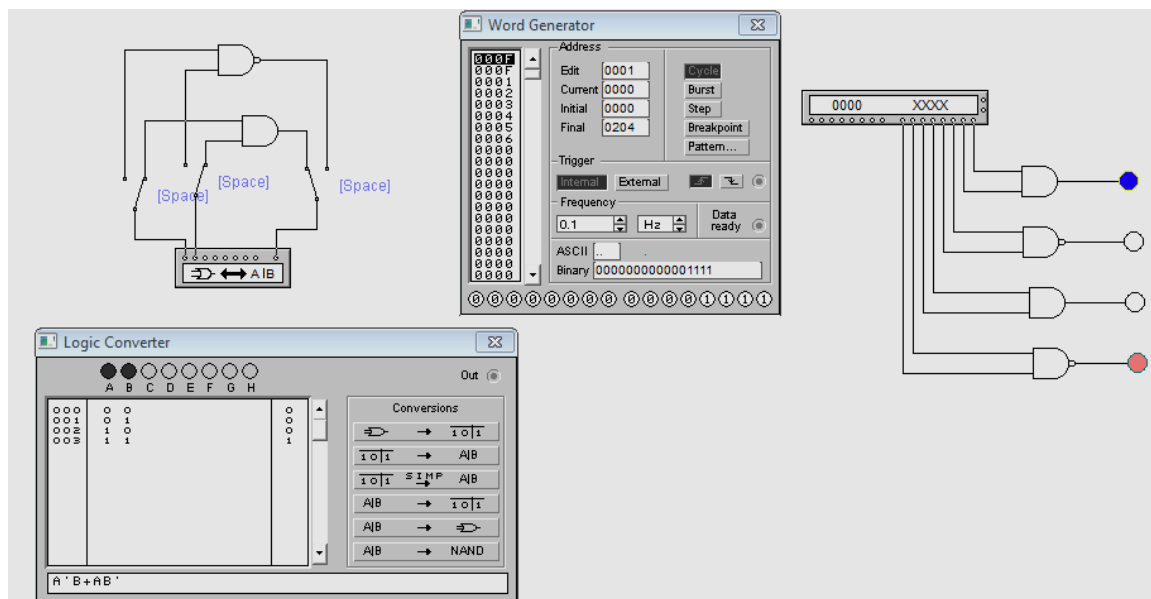


Рис. 16 – Схемы исследования логических элементов И, И-НЕ

- 2) В логическом преобразователе нажать кнопку отображения таблицы истинности.
- 3) Занести в отчет таблицу истинности элемента И.
- 4) Нажатием клавиши «Пробел» подсоединить к логическому преобразователю другой элемент И-НЕ, затем нажать кнопку отображения таблицы истинности и занести ее в отчет.
- 5) В генератор слова в первые (верхние) ячейки записать кодовые слова 0000 (в ячейку с адресом 0000), 0055 (в ячейку с адресом 0001), 00AA (в ячейку с адресом 0002) и 00FF (в ячейку с адресом 0003), в результате обращения к этим ячейкам на выходе генератора сформируются логические потенциалы соответственно 0000 0000 0000 0000, 0000 0000 0101 0101, 0000 0000 1010 1010 и 0000 0000 1111 1111.
- 6) В окне Instial (начальный адрес, запуска генератора) записать код нулевой ячейки – 0000. Запустить генератор, нажав кнопку Step.
- 7) Проанализировать зависимость уровней логического напряжения на выходе элементов И, И-НЕ от уровней на их входе.
- 8) Изменяя начальный адрес последовательно на 0001, 0002 и 0003, запуская генератор нажатием кнопки Step, провести анализ зависимости уровней логического напряжения на выходе элементов И, И-НЕ от уровней на их входе.
- 9) Сделайте выводы по результатам проделанной работы.

Задание 3.2.5. Провести анализ логических элементов ИЛИ и ИЛИ-НЕ. Для этого выполните следующие операции:

- 1) В схемах исследования, приведенных на рис. 16, заменить присутствующие элементы логическими элементами ИЛИ и ИЛИ-НЕ, меняя элемент И элементом ИЛИ, а элемент И-НЕ – ИЛИ-НЕ.
- 2) Выполнить пункты 2 – 8 предыдущего задания для элементов ИЛИ и ИЛИ-НЕ.
- 9) Сделайте выводы по результатам проделанной работы.

Задание 3.2.6. Провести анализ логических элементов ИСКЛЮЧАЮЩЕЕ ИЛИ и равнозначности. Для этого выполните следующие операции:

- 1) В схемах исследования, приведенных на рис. 16, заменить присутствующие элементы логическими элементами Иключающее ИЛИ и Иключающее ИЛИ-НЕ (элемент равнозначности), меняя элемент ИЛИ элементом Иключающее ИЛИ, а элемент ИЛИ-НЕ – Иключающее ИЛИ-НЕ.
- 2) Выполнить пункты 2 – 8 предыдущего задания для элементов Иключающее ИЛИ и Иключающее ИЛИ-НЕ (элемент равнозначности).
- 9) Сделайте выводы по результатам проделанной работы.

3.3. Подтверждение законов булевой алгебры с помощью программы схемотехнического моделирования Electronics Workbench

Задание 3.3.1. Подтвердить переместительный закон. Для этого выполните следующие операции:

- 1) С помощью логического преобразователя проанализируйте таблицы истинности элементов И, ИЛИ.
- 2) Сделайте выводы по результатам анализа.

Задание 3.3.2. Подтвердить сочетательный закон. Для этого выполните следующие операции, приведенные на рис. 17:

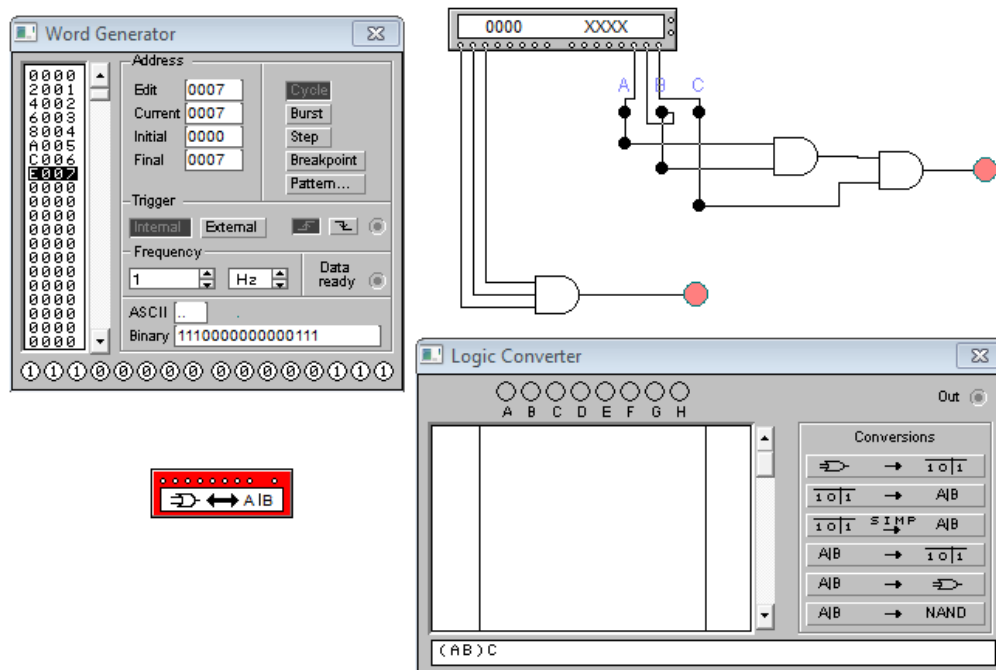
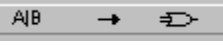


Рис. 17 – Схема исследования для подтверждения сочетательного закона

1) В экране-строке логического преобразователя запишите булеву функцию для сочетательного закона (см. рис. 17):

$A(BC)$ или $(AB)C$ либо $A+(B+C)$ или $(A+B)+C$.

2) Нажмите кнопку синтеза логического устройства .

3) Синтезированное устройство подключите к трем левым выходам генератора слова, как на рис. 17, а к выходу устройства подсоедините логический пробник.

4) Переместите на рабочее поле из компоненты Логические элементы элемент И либо ИЛИ в зависимости от того, какая булева функция подвергается анализу.

5) Установите для выбранного элемента количество входов, равное трем. Для этого правой кнопкой мыши щелкните по логическому элементу и в открывшемся контекстно-зависимом меню выберите команду Component Properties ..., в диалоговом окне откройте вкладку Number of inputs и установите опцию напротив цифры 3.

6) Подключите 3-х входовой логический элемент слева к первым 3-м выходам генератора слова, а к выходу элемента подключите логический пробник как показано на рис. 17.

7) В генераторе слова запишите в нижеприведенные ячейки следующие коды:

Ячейка	Кодовое слово
0000	0000
0001	2001
0002	4002
0003	6003
0004	8004
0005	A005
0006	C006
0007	E007

8) В генераторе слова установите начальный адрес цикла – 0000, финальный адрес – 0007. Тактовую частоту установите 1 Гц.

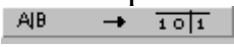
9) Запустите генератор, нажав кнопку Cycle.

10) Проанализируйте состояние выходов генератора и работу подключенных к нему элементов.

11) Сделайте выводы по результатам анализа.

Задание 3.3.3. Подтвердить закон повторения. Для этого выполните следующие операции:

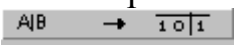
1) В экране-строке логического преобразователя запишите булеву функцию этого закона AA либо $A+A$.

2) Нажмите на кнопку отображения таблицы истинности, соответствующей этому выражению .

3) Проанализируйте полученную таблицу истинности, и сделать выводы.

Задание 3.3.4. Подтвердить закон двойной инверсии. Для этого выполните следующие операции:

1) В экране-строке логического преобразователя запишите булеву функцию этого закона A'' .

2) Нажмите на кнопку отображения таблицы истинности, соответствующей этому выражению .

3) Проанализируйте полученную таблицу истинности, и сделать выводы.

Задание 3.3.5. Подтвердить законы универсального и нулевого множеств. Для этого выполните следующие операции:

1) Собрать схему исследования, приведенную на рис.18. В исходном состоянии к входам элементов подключен потенциал логической единицы.

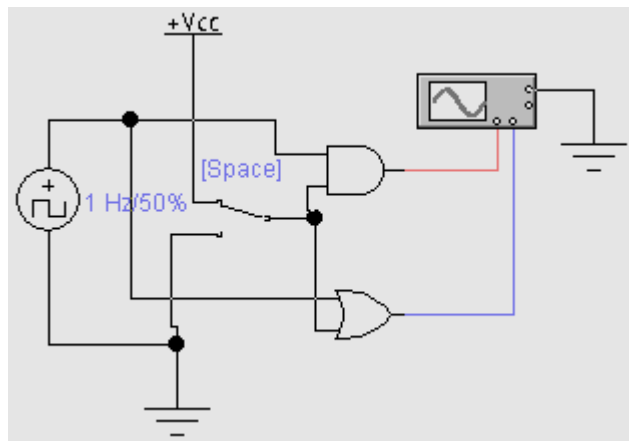


Рис. 18 – Схема исследования для подтверждения законов нулевого и универсального множеств

2) На генераторе таковых импульсов установить частоту 1 Гц и напряжение 5 В.

3) На осциллографе установить горизонтальную (временную) развертку – 0,50 с/дел, а вертикальную (по амплитуде) – 5 В/дел по обоим каналам.

4) Запустить работу схемы, нажав на выключатель.

5) Отобразить эпюры напряжения с экрана осциллографа в отчете.

6) Нажав клавишу Пробел, подключить к входам элементов И, ИЛИ потенциал логического нуля.

7) Повторить пункты 5 и 6.

8) Провести анализ и сделать выводы.

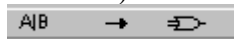
Задание 3.3.6. Подтвердить распределительный закон булевой алгебры. Для этого выполните следующие операции:

1) В экране-строке логического преобразователя запишите булеву функцию этого закона $A+BC$.

2) Нажмите на кнопку отображения таблицы истинности, соответствующей этому выражению . Запишите таблицу истинности в

отчет.

3) Нажмите кнопку синтеза логического устройства по данной булевой функции



4) В экране-строке логического преобразователя запишите тождественную булеву функцию $(A+B)(A+C)$.

5) Определите и запишите в отчет таблицу истинности этой функции. Сравните с ранее полученной в пункте 2 таблицей истинности.

6) С помощью логического преобразователя синтезируйте логическое устройство по данной функции.

7) Синтезированные устройства подключите к трем левым и к трем правым выходам генератора слова, как на рис. 19, а к выходам устройств подсоедините логические пробники.

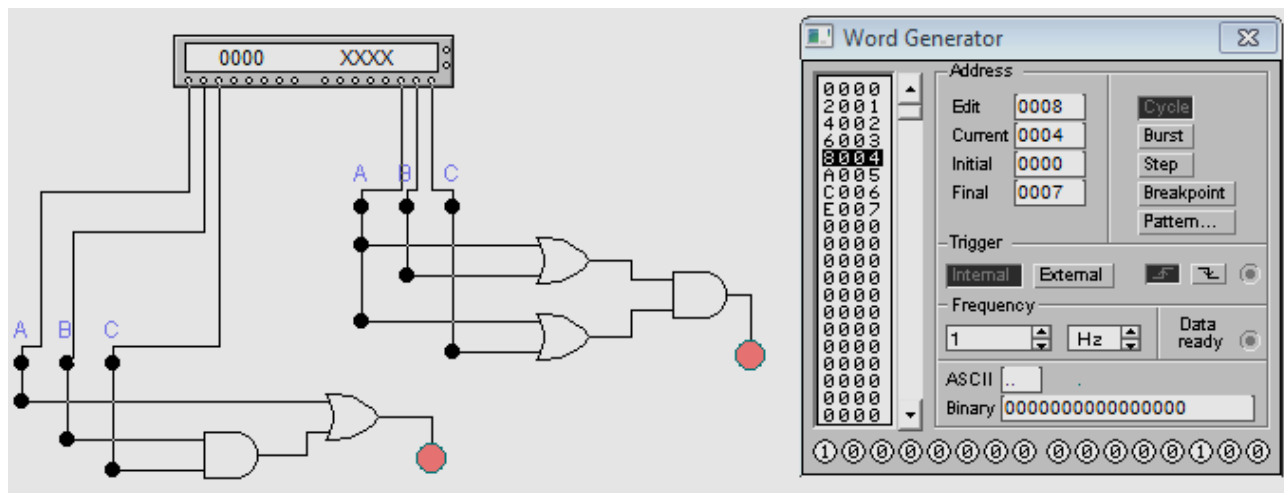


Рис. 19 – Схема исследования для подтверждения распределительного закона

8) В генераторе слова запишите в нижеприведенные ячейки следующие коды:

Ячейка	Кодовое слово
0000	0000
0001	2001
0002	4002
0003	6003
0004	8004
0005	A005
0006	C006
0007	E007

9) В генераторе слова установите начальный адрес цикла – 0000, финальный адрес – 0007. Тактовую частоту установите 1 Гц.

10) Запустите генератор, нажав кнопку Cycle.

11) Проанализируйте состояние выходов генератора и работу подключенных к нему элементов.

12) Сделайте выводы по результатам проведенного анализа.

Задание 3.3.7. В часы самостоятельной работы подтвердить закон Де Моргана. Для этого выполните следующие операции:

1) Провести процедуру исследования аналогичную предыдущему пункту.

2) В отчете отразить таблицы истинности и схемы исследования.

3) Сделайте выводы по результатам проведенного анализа.

Задание 3.3.8. В часы самостоятельной работы подтвердить закон дополнительности. Для этого выполните следующие операции:

1) В экране-строке логического преобразователя запишите булеву функцию этого

закона AA' или $A+A'$.

2) Нажмите на кнопку  отображения таблицы истинности, соответствующей этим выражениям. Запишите таблицы истинности в отчет и проанализируйте.

3) Сделайте выводы по результатам проведенного анализа.

Задание 3.3.9. В часы самостоятельной работы подтвердить законы поглощения и склеивания. Для этого выполните следующие операции:

1) Провести алгебраические преобразования булевых выражений и воспользовавшись аксиомами булевой алгебры подтвердить данные законы.

2) Отобразить в отчете данные преобразования и сделать выводы по результатам проведенного анализа.

3.4. Синтез логических устройств с помощью программы схемотехнического моделирования Electronics Workbench

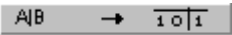
Задание 3.4.1. Синтезировать по заданной булевой функции логические устройства. Для этого выполните следующие операции:

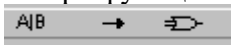
1) Записать в отчет заданную по номеру варианта булеву функцию из таблицы 1.

Таблица 1 – Таблица вариантов заданий

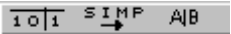
Номер варианта	Функция
1.	$F1=A'B'C+A'BC'+A'BC+AB'C'$; $F2=A'BC'+AB'C'+AB'C+ABC'+ABC$
2.	$F1=A'B'C+A'BC'+AB'C'+AB'C$; $F2=A'BC+AB'C'+AB'C+ABC'+ABC$
3.	$F1=A'B'C+A'BC'+AB'C+ABC'$; $F2=A'B'C+A'BC+AB'C+ABC'+ABC$
4.	$F1=A'B'C+A'BC'+ABC'+ABC$; $F2=A'B'C+A'BC+AB'C'+ABC'+ABC$
5.	$F1=A'B'C'+A'BC'+AB'C'+ABC'$; $F2=A'B'C'+A'BC'+AB'C'+AB'C+ABC$
6.	$F1=A'B'C'+A'BC+AB'C+ABC'$; $F2=A'B'C'+A'B'C+A'BC'+ABC'+ABC$
7.	$F1=A'B'C+A'BC'+AB'C+ABC$; $F2=A'B'C'+A'BC+AB'C'+AB'C+ABC$
8.	$F1=A'B'C'+AB'C'+AB'C+ABC$; $F2=A'B'C+A'BC'+A'BC+ABC'+ABC$
9.	$F1=A'B'C'+A'B'C+A'BC'+AB'C'$; $F2=A'BC+AB'C'+AB'C+ABC'+ABC$
10.	$F1=A'BC'+A'BC+AB'C'+AB'C$; $F2=A'B'C'+A'B'C+A'BC+ABC'+ABC$

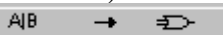
2) Записать функцию F1 в экран-строку логического преобразователя.

3) Отобразить таблицу истинности устройства, нажав кнопку , и проверить правильность набора функции.

4) Нажать кнопку  синтеза логического устройства по данной булевой функции.

5) Отобразить синтезированное устройство в отчете.

6) Минимизировать заданную функцию, нажав кнопку .

7) Синтезировать минимизированное логическое устройство, нажав кнопку .

8) Отобразить минимизированное синтезированное устройство в отчете, сравнить его с ранее синтезированным устройством.

9) Повторить пункты 2–8 для синтеза устройства по булевой функции F2.

10) С помощью генератора слова проверить работу синтезированного устройства.

11) Сделайте выводы по результатам проведенного синтеза.

Контрольные вопросы:

- 1) Сравнительная характеристика комбинационных и последовательностных (накапливающих) цифровых устройств.
- 2) Принцип работы электронного ключа (вентилля).

- 3) Характеристика простейших логических операций.
- 4) Аксиомы и законы булевой алгебры.
- 5) Условно-графическое обозначение и таблицы истинности базовых логических элементов.
- 6) Краткая характеристика и порядок моделирования сем с помощью программы схемотехнического моделирования Electronics Workbench.
- 7) Характеристика и порядок работы с логическим преобразователем программа схемотехнического моделирования Electronics Workbench.
- 8) Характеристика и порядок работы с генератором слова программа схемотехнического моделирования Electronics Workbench.
- 9) Характеристика и порядок работы с логическим анализатором программа схемотехнического моделирования Electronics Workbench.
- 10) Порядок анализа основных логических элементов с помощью программы схемотехнического моделирования Electronics Workbench.
- 11) Методика подтверждение законов булевой алгебры с помощью программы схемотехнического моделирования Electronics Workbench.
- 12) Порядок синтеза логических устройств с помощью программы схемотехнического моделирования Electronics Workbench.

Лабораторная работа №4 Моделирование и исследование комбинационных цифровых устройств (2 часа)

Содержание:

1. Моделирование и исследование комбинационных цифровых устройств.

Цель работы:

В результате выполнения практического занятия студенты должны сформировать умение использовать знания принципов построения и работы базовых элементов компьютера для анализа и синтеза информационных систем и процессов, а также привить навык анализа и синтеза цифровых логических устройств ЭВМ с помощью программы схемотехнического моделирования Electronics Workbench.

Теоретические сведения:

Под *комбинационным цифровым устройством (КЦУ)* понимается цифровое устройство, обеспечивающее преобразование совокупности N входных цифровых сигналов в M выходных, при этом состояние выходных сигналов в данный момент времени определяется состоянием входных сигналов в этот же момент времени. Иными словами, КЦУ «не помнит» предыстории поступления сигналов на его входы. Правила функционирования КЦУ определяются реализуемыми ими функциями алгебры логики.

Реализация КЦУ предполагает выбор определенных логических элементов из заданного набора и их соединение таким образом, чтобы обеспечивалась зависимость цифровых выходных сигналов от входных с заданными правилами функционирования. При реализации КЦУ широко используются интегральные комбинационные логические микросхемы малой степени интеграции, образующие основу элементной базы цифровой электроники.

Арифметические сумматоры

Арифметические сумматоры являются составной частью так называемых арифметико-логических устройств (АЛУ) микропроцессоров (МП). Они используются также для формирования физического адреса ячеек памяти в МП с сегментной организацией памяти. В программе EWB арифметические сумматоры представлены в библиотеке компонентов Digital

двумя базовыми устройствами, показанными на рис. 20: полусумматором и полным сумматором.

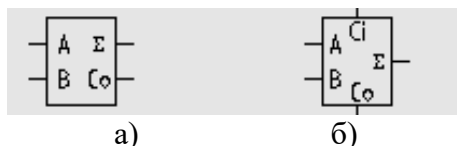


Рис. 20 – Схемы полусумматора (а) и полного сумматора (б)

Они имеют следующие назначения выводов: А, В – входы суммируемых операндов, Σ – результат суммирования, Co – выход переноса, Ci – вход переноса. Многоразрядный сумматор создается на базе одного полусумматора и n полных сумматоров. В качестве примера на рис. 21 приведена структура трехразрядного сумматора. На входы А1, А2, А3 и В1, В2, В3 подаются первое и второе слагаемые соответственно, а с выходов S1, S2, S3 снимается результат суммирования.

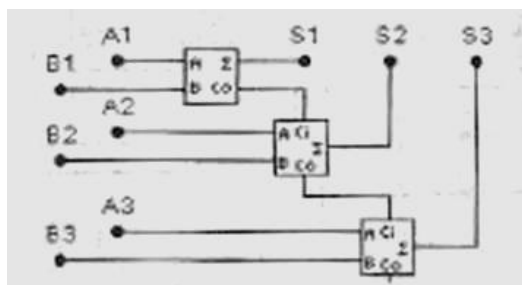


Рис. 21 – Трехразрядный сумматор

Для исследования внутренней структуры и логики функционирования сумматоров используют логический преобразователь.

Дешифраторы и шифраторы

Дешифратор (декодер) – комбинационное устройство с несколькими входами и выходами, у которого определенным комбинациям входных сигналов соответствует активнее состояние одного из выходов. Дешифратор является обращенным по входам демультиплексором, у которого адресные входы стали информационными, а бывший информационный вход стал входом разрешения. Поэтому часто дешифраторы называют дешифраторами-демультиплексорами и наоборот.

Дешифраторы в виде серийных ИМС средней степени интеграции широко используются в информационно-измерительной технике и микропроцессорных системах управления, в частности, в качестве коммутаторов-распределителей информационных сигналов и синхроимпульсов, для демультиплексирования данных и адресной логики в запоминающих устройствах, а также для преобразования двоично-десятичного кода в десятичный с целью управления индикаторными и печатающими устройствами.

Дешифраторы, как самостоятельные изделия электронной техники имеют 4, 8 или 16 выходов. Если требуется большее число выходов, дешифраторы наращиваются в систему.

Схема дешифратора 2х4 (рис. 22, а) содержит три инвертора U1 ... U3 и четыре элемента ЗИ-НЕ U4 ... U7; входы 2^0 , 2^1 – адресные, G' – вход разрешения (активный уровень – сигнал логического нуля); выходы 0 ... 3 – инверсные.

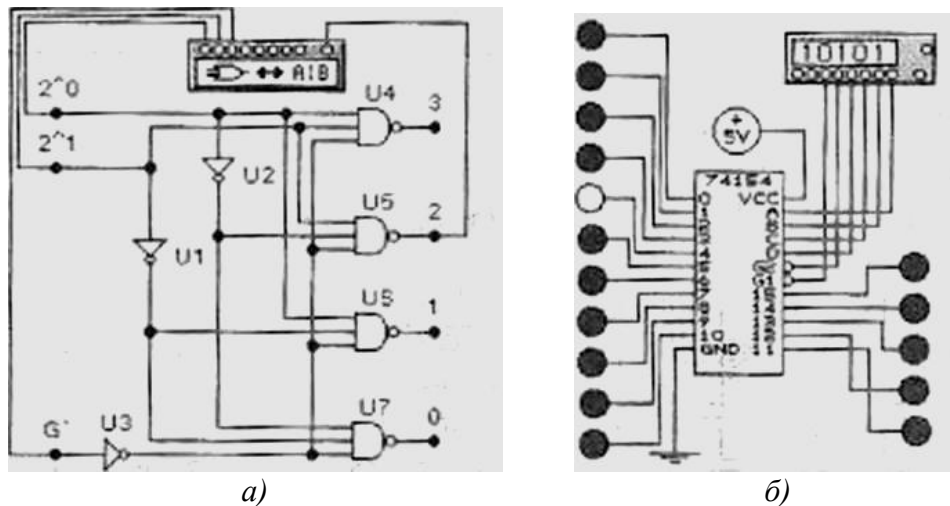


Рис. 22 – Дешифратор 2х4 (а) и схема включения ИМС 74154 (б)

Заметим, что в серийных ИМС дешифраторов универсального типа выходы всегда инверсные, т.е. в исходном состоянии на его выходах сигнал логической единицы. Это вызвано тем, что они чаще всего используются в адресных селекторах для выбора того или иного устройства (например, микросхемы ОЗУ), в которых сигнал выбора, как правило, должен иметь низкий уровень. В качестве примера на рис. 22, б) приведена схема включения дешифратора 74154 (отечественный аналог К155ИДЗ). Эта ИМС имеет четыре адресных входа А, В, С, D, два входа разрешения G1, G2 и шестнадцать выходов 0...15 (выходы не прямые, как обозначено в EWB, а инверсные). В режиме дешифратора с генератора слова на входы G1, G2 подается 0, а на адресные входы – код в диапазоне 0000 ... 1111. **В режиме демультиплексора один из разрешающих входов, например G1, используется в качестве информационного входа.** Информационный сигнал в виде логического 0 с этого выхода распределяется по выходам 0 ... 15 в соответствии с состоянием адресных входов, т.е. режимы дешифратора и демультиплексора трудно различимы.

Шифраторы (кодеры) используются для кодирования унитарного кода в двоичный код, а также очень часто для преобразования десятичных чисел в двоичный или двоично-десятичный код, например, в микрокалькуляторах, в которых нажатие десятичной клавиши соответствует генерации соответствующего двоичного кода.

Электромеханический аналог шифратора для кодирования десятичных чисел 1...7 в двоичный код (рис. 23) содержит «кнопки» указанных цифр в виде одинарных, сдвоенных или строенных переключателей, управляемых цифровыми клавишами клавиатуры. Состояние «кнопок» - включена/ выключена – устанавливается тактовыми импульсами опроса от источника однополярных импульсов U_t и на выходе шифратора индицируется логическими пробниками для двоичного кода и алфавитно-цифровым индикатором для десятичного.

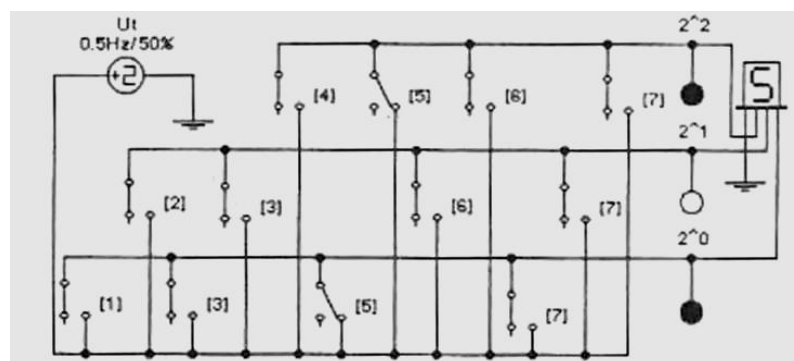


Рис. 23 – Схема электромеханического шифратора

Поскольку в реальных клавиатурах возможно нажатие сразу нескольких клавиш, в шифраторах используется принцип приоритета старшего разряда, т.е. при нажатии, например, клавиш 9, 5 и 2 на выходе шифратора будет генерироваться код 1001, соответствующий цифре 9. Следует отметить, что шифраторы как отдельный класс функциональных устройств представлены в наиболее богатой ТТЛ-серии всего двумя ИМС – 74147 и 74148, причем последняя ИМС имеется и в библиотеке программы EWB. Схема ее включения показана на рис. 24, а), а режимы работы используемого генератора слова – на рис. 24, б).

Назначение выводов ИМС 74148: 0 ... 7 – входы; A0, A1, A2 – выходы; E1 – вход разрешения; E0, GS – выходы для каскадирования шифраторов. При моделировании необходимо обратить внимание на реализацию принципа приоритета, при этом следует учесть, что все входы и выходы – инверсные (на функциональной схеме ИМС в программе EWB они показаны прямыми).

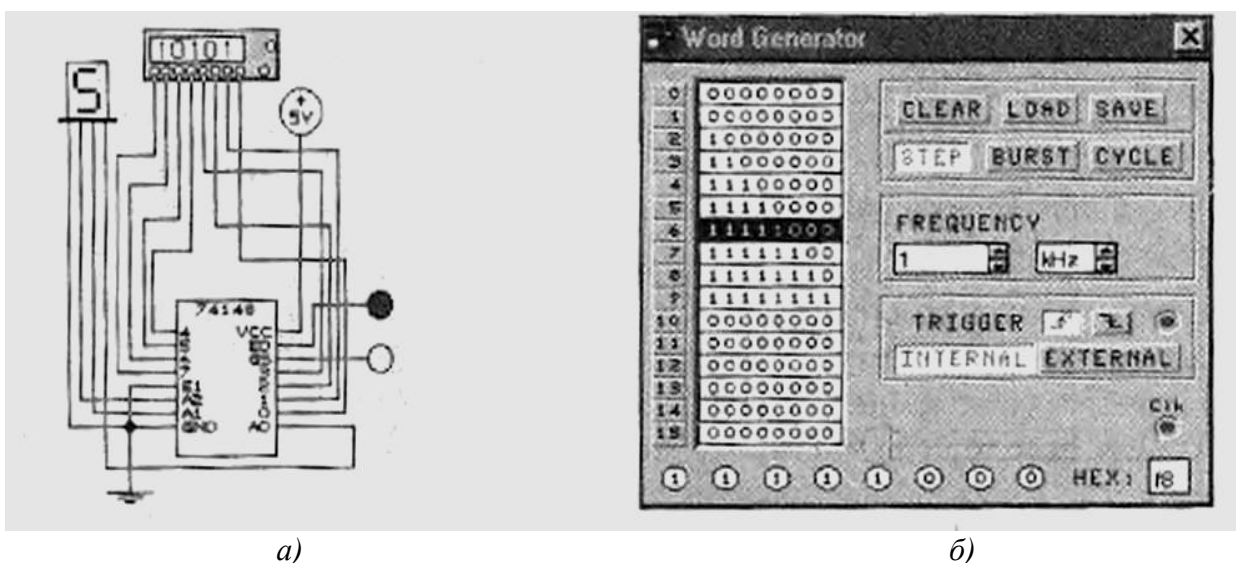


Рис. 24 – Схема включения ИМС 74148 (а)

и панель генератора слова с установками для испытания (б)

Мультиплексоры и демультиплексоры

Мультиплексоры (от англ. *multiplex* – многократный) используются для коммутации в заданном порядке сигналов, поступающих с нескольких входных шин на одну выходную. У мультиплексора может быть, например, 16 входов и 1 выход. Это означает, что если к этим входам присоединены 16 источников цифровых сигналов – генераторов последовательных цифровых слов, то байты от любого из генераторов можно передавать на единственный выход. Для выбора любого из 16 каналов необходимо иметь 4 входа селекции ($2^4 = 16$), на которые подается двоичный адрес канала. Так, для передачи данных от канала номер 9 на входах селекции необходимо установить код 1001. В силу этого мультиплексоры часто называют селекторами или селекторами-мультиплексорами.

Мультиплексоры применяются, например, в микропроцессорах (МП) для выдачи на одни и те же выходы МП адреса и данных, что позволяет существенно сократить общее количество выводов микросхемы; в микропроцессорных системах управления мультиплексоры устанавливают на удаленных объектах для возможности передачи информации по одной линии связи от нескольких установленных на них датчиков.

На рис. 25, а) приведена схема двухканального мультиплексора, состоящего из элементов ИЛИ, НЕ и двух элементов И, а на рис. 25, б) – функциональная схема ИМС 74151

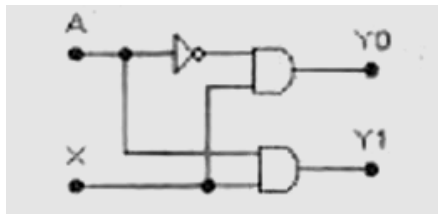


Рис. 27 – Схема двухканального демультиплексора

Задание для выполнения работы:

3.5. Исследование арифметических сумматоров

Задание 3.5.1. Исследовать внутреннюю структуру и логику функционирования полусумматора, используя логический преобразователь и генератор слова. Для этого выполните следующие операции:

- 1) Собрать схему исследования полусумматора, приведенную на рис 28.

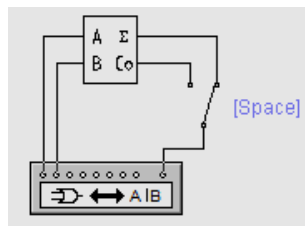
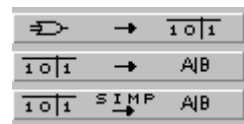


Рис. 28 – Схема исследования полусумматора



- 2) Последовательно нажать кнопки: . В результате получить и записать в отчет таблицу истинности и булево выражение.

- 3) Сравнить полученные данные с результатами исследования базовых логических элементов в предыдущем разделе, сделать вывод о том, какой элемент реализует полученную булеву функцию при подключении вывода © полусумматора к зажиму OUT преобразователя (как показано на рис. 28).

- 4) Синтезировать данный элемент в полнофункциональном базисе И, ИЛИ, НЕ, нажав кнопку .

- 5) Подключить клемму OUT преобразователя к выходу Co полусумматора с помощью переключателя, нажав клавишу «Пробел», и проделать аналогичные действия, сделав вывод о том, какой логический элемент реализует функции полусумматора при таком включении, произведя его синтез.

- 6) В отчете отобразить схему синтезированного полусумматора.

- 7) Проверить работу синтезированного полусумматора, подключив к его выходам логические пробники, и задав комбинации на входе с помощью генератора слова. Для этого впервые 4 ячейки генератора записать соответственно коды 0000, 0001, 0002 и 0003.

- 8) Сделайте выводы по результатам проведенного анализа и синтеза.

Задание 3.5.2. Исследовать внутреннюю структуру и логику функционирования полного сумматора, используя логический преобразователь и генератор слова. Для этого выполните следующие операции:

- 1) Собрать схему исследования полного сумматора как на рис 29.

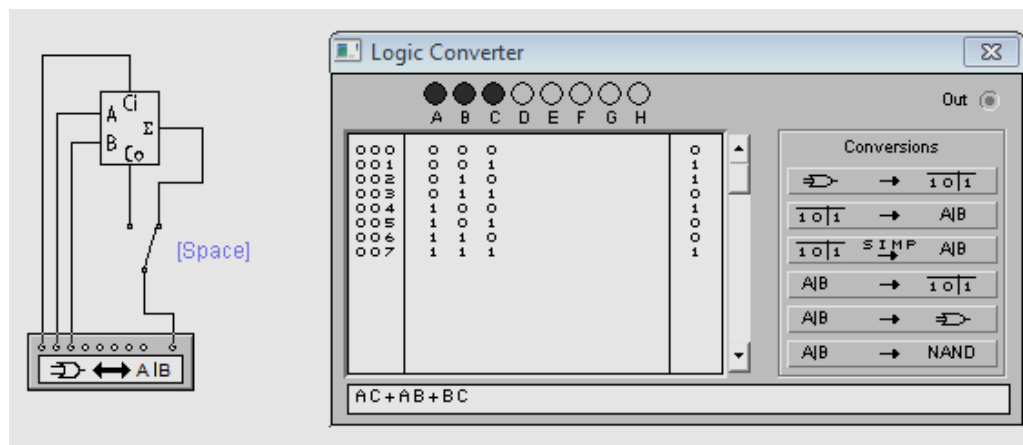


Рис. 29 – Схема исследования полного сумматора

2) Повторить пункты 2 – 7 предыдущего задания.

3) Сделайте выводы по результатам проведенного анализа и синтеза.

Задание 3.5.3. Исследовать логику функционирования 4-х разрядного сумматора с помощью логического преобразователя и генератора слова. Для этого выполните следующие операции:

1) Собрать схему исследования 4-х разрядного сумматора как на рис 30.

2) Запрограммировать ячейки генератор слова таким образом, чтобы на входы **B** сумматора подавалась комбинация кодов от 0000 до 1111, а на входах **A** постоянно присутствовал код, указанный преподавателем, например в приведенной схеме исследования он равен – 0101.

3) В пошаговом режиме генератора слова проверить функционирование сумматора, записав все его результаты вычислений в отчетную таблицу 2.

Таблица 2.

Код операнда A	Код операнда B	Код суммы

4) Сделайте выводы по результатам проведенного исследования.

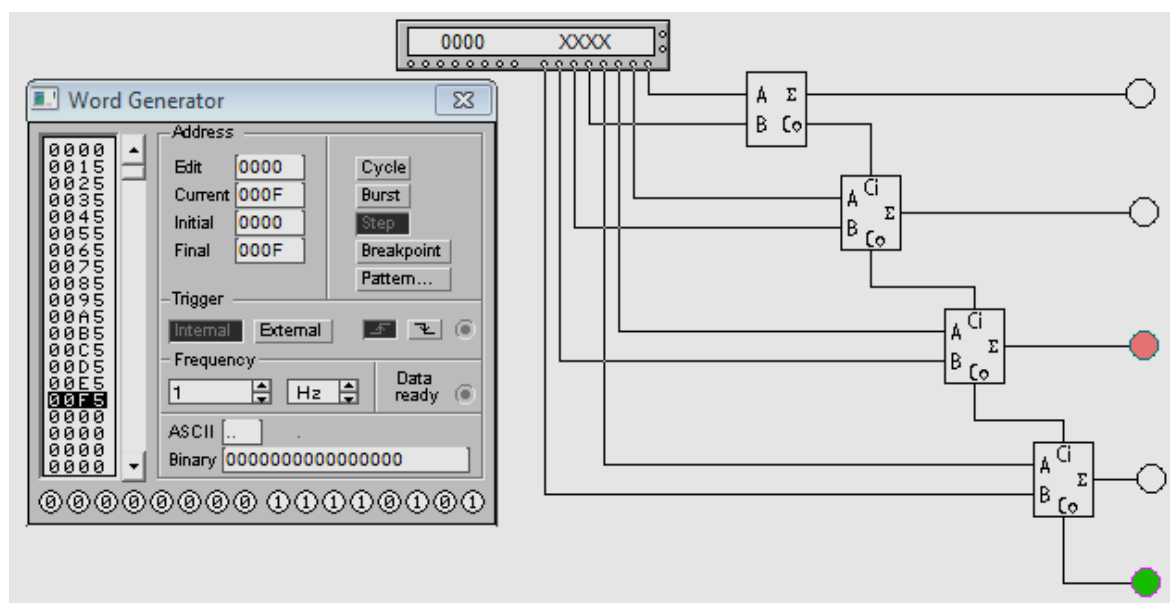


Рис. 30 – Схема исследования 4-х разрядного сумматора

3.6. Исследование шифраторов и дешифраторов

Задание 3.6.1. Синтезировать и исследовать логику функционирования дешифратора 3х8 с помощью логического преобразователя и генератора слова. Для этого выполните следующие операции:

1) Открыть логический преобразователь, активизировать клеммы А, В и С, нажав на них курсором.

2) В таблице истинности, появившейся на экране, внести изменения таким образом, чтобы в выходном столбце была записана «1» в строке с адресом 000, а в остальных строках столбца должны находиться «0».

3) Нажать кнопку отображения булевой функции, по которой синтезировать часть схемы дешифратора, которая будет отвечать за активацию нулевого выхода, т.е. за появления на этом выходе логической единицы, при условии, что на всех трех входах будут присутствовать логические нули. Занести часть схемы в отчет.

4) Последовательно повторить действия для строк с адресами 001, 002 и т.д. до 007, определяя булевы функции, по которым аналогичным образом синтезировать другие части дешифратора, отвечающие за активацию первого, второго и так далее седьмого выходов.

5) В отчете отобразить схему дешифратора, которую необходимо минимизировать, удалив лишние инверторы.

6) Проверить работу синтезированной схемы дешифратора с помощью генератора слова и логических пробников, подключив к их выходам дешифратора, например как показано на рис. 31. Для этого задать в генераторе слова цикл с 0000 по 0007, в котором записать соответствующие коды.

7) Сделайте выводы по результатам проведенного исследования.

Задание 3.6.2. Исследовать логику функционирования дешифратора 4х16, реализованного в виде ИМС 74154 (К155ИД3) с помощью генератора слова. Для этого выполните следующие операции:

1) В панели Digital нажать кнопку DEC и выбрать микросхему 74154. Собрать схему исследования, как показано на рис. 32.

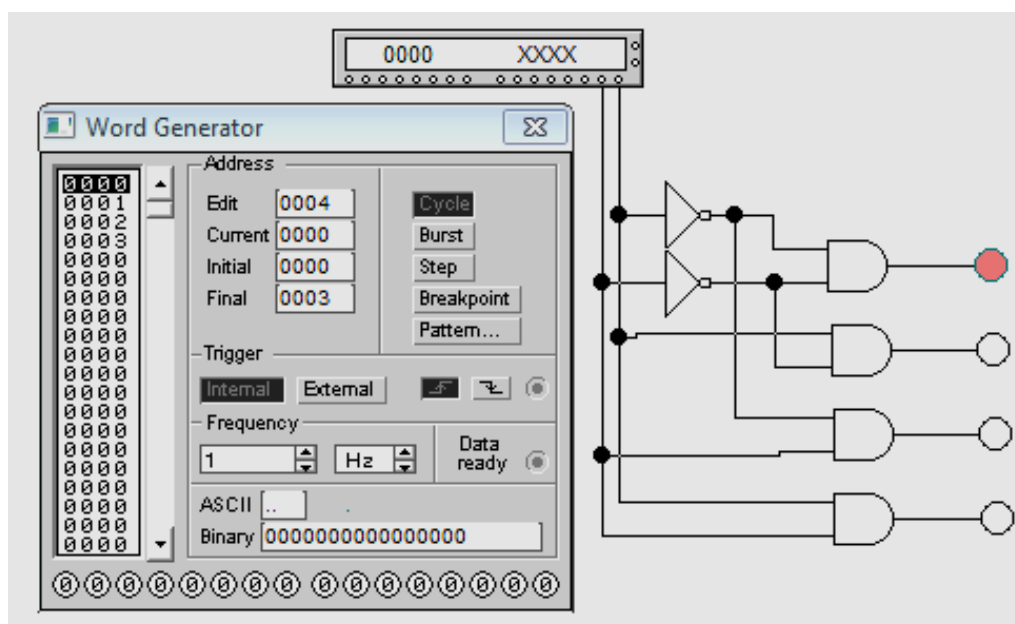


Рис. 31 – Схема исследования дешифратора 2х4

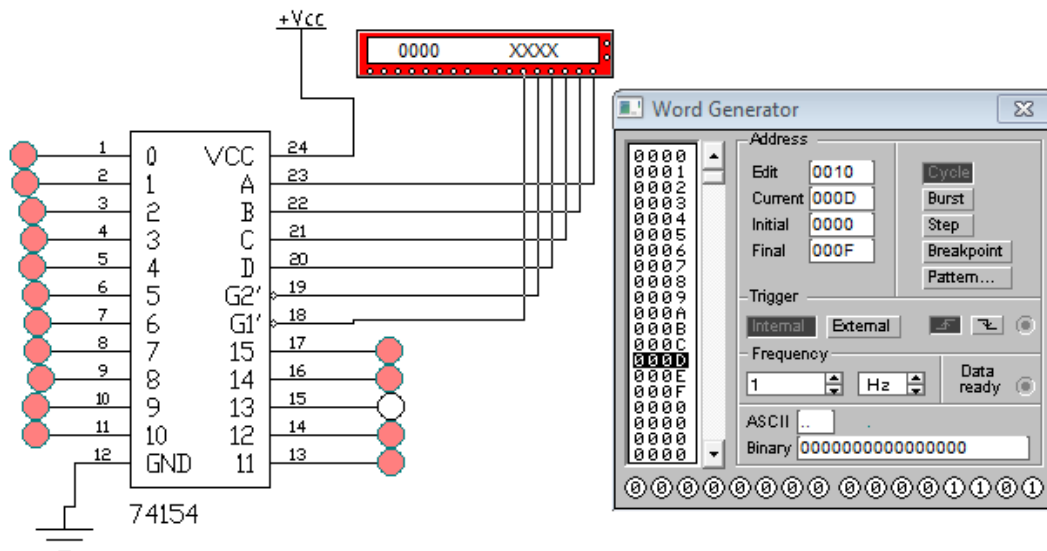


Рис. 32 – Схема включения ИМС 74154 (К155ИД3)

2) Задать в генераторе слова цикл с 0000 по 000F, в котором записать соответствующие коды.

3) Запустить генератор слова с тактовой частотой 1 Гц, провести анализ логики функционирования дешифратора 4x16.

4) Сделайте выводы по результатам проведенного исследования.

Задание 3.6.3. Синтезировать и исследовать логику функционирования шифратора 4x2 с помощью генератора слова. Для этого выполните следующие операции:

1) Записать таблицу истинности шифратора 4x2, имеющую 4 входа и 2 выхода.

2) Записать 4 булевы функции по входным операндам унитарного кода.

3) Синтезировать логические устройства, реализующие данные функции, с помощью полнофункционального базиса И, ИЛИ, НЕ, как на рис. 33.

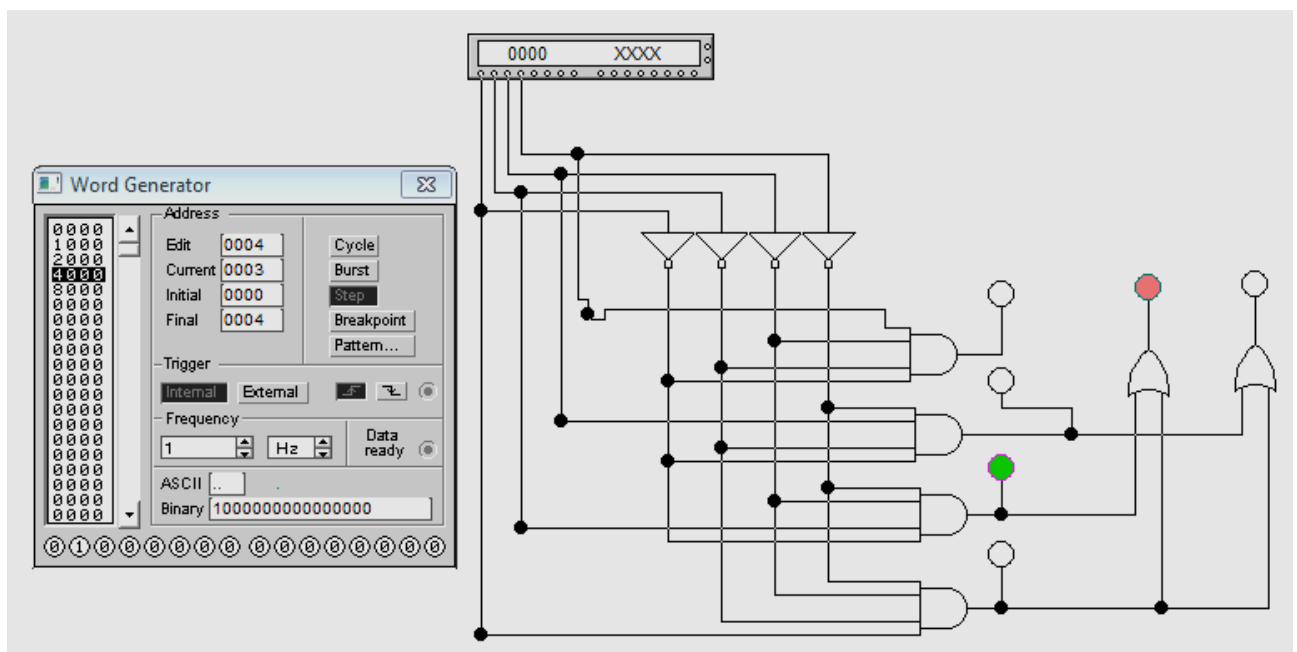


Рис. 33 – Схема исследования шифратора 4x2

4) Синтезировать 2 выходные логические функции шифраторов путем логического суммирования по единицам на выходах таблицы истинности.

5) Собрать схему шифратора 4х2, как показано на рис. 33.

6) Провести исследование логики функционирования шифратора с помощью генератора слова в пошаговом режиме. Для чего в ячейки генератора с 0000 по 0004 записать последовательно коды 0000, 1000, 2000, 4000 и 8000, которые формируют на выходных клеммах генератора унитарный код.

7) Определить в схеме шифратора избыточные (лишние) логические элементы, после чего минимизировать схему, удалив избыточные элементы.

8) Повторить исследование логики функционирования минимизированной схемы шифратора, убедиться в правильности ее работы.

9) Сделайте выводы по результатам проведенного исследования.

Задание 3.6.4. Исследовать логику функционирования шифратора 8х3, реализованного в виде ИМС 74148 (KM555IB1) с помощью генератора слова. Для этого выполните следующие операции:

1) В панели Digital нажать кнопку ENC и выбрать микросхему 74148. Собрать схему исследования, как показано на рис. 34, используя при этом отображения двоичного кода типовой семисегментный индикатор с дешифратором, находящийся в панели Indicators.

2) Записать таблицу истинности шифратора 8х3 в прямом и инверсном виде.

3) При моделировании унитарного кода в генераторе слова следует учесть, что все входы и выходы – инверсные, хотя на функциональной схеме ИМС в программе EWB они показаны прямыми. В связи с этим задать в генераторе слова цикл с 0000 по 0007, в котором записать последовательно следующие коды: 007F, 00BF, 00DF, 00EF, 00F7, 00FB, 00FD, 00FE.

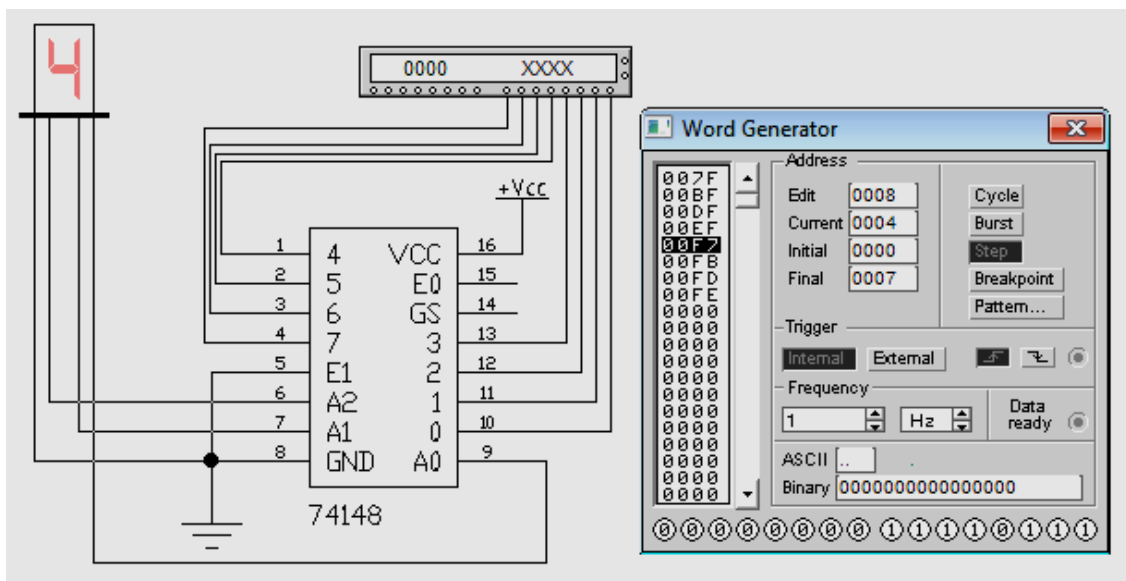


Рис. 34– Схема включения ИМС 74148 (KM555IB1)

4) Запустить генератор слова с тактовой частотой 1 Гц, провести анализ логики функционирования шифратора 8х3.

5) Сделайте выводы по результатам проведенного исследования.

3.7. Исследование мультиплексоров и демультиплексоров

Задание 3.7.1. Синтезировать и исследовать логику функционирования 4-х канального мультиплексора с помощью логического анализатора. Для этого выполните следующие операции:

1) Провести анализ структуры 4-х канального мультиплексора, из которого следует, что мультиплексор должен состоять из 4-х ключей (логических элементов И), 4-х входного элемента ИЛИ и дешифратора 2х4, вырабатывающего унитарный код, который обеспечивает включение одного из 4-х ключей.

2) Собрать схему исследования 4-х канального мультиплексора, приведенную на рис. 35. Генераторы тактовых импульсов с частотами 0,5 КГц, 1 КГц, 2 КГц и 4 КГц имитируют каналы передачи данных. Генераторы тактовых импульсов с частотами 10 Гц и 20 Гц обеспечивают автоматическое переключение адресов для коммутации каналов. Логические пробники предназначены для определения номера коммутируемого канала.

3) Двойным щелчком открыть логический анализатор и запустить работу исследуемой схемы.

4) С помощью логического анализатора и логических пробников проанализировать логику работы мультиплексора. После останова логического анализатора просмотреть на его экране кодовые последовательности входных и выходного сигналов возможно с помощью горизонтальной прокрутки. Отобразить логические последовательности, полученные с помощью анализатора, в отчет.

5) Сделайте выводы по результатам проведенного исследования.

Задание 3.7.2. Исследовать логику функционирования восьмиканального селектора-мультиплексора, реализованного в виде ИМС 74151 (КМ155КП7) с помощью логического анализатора. Для этого выполните следующие операции:

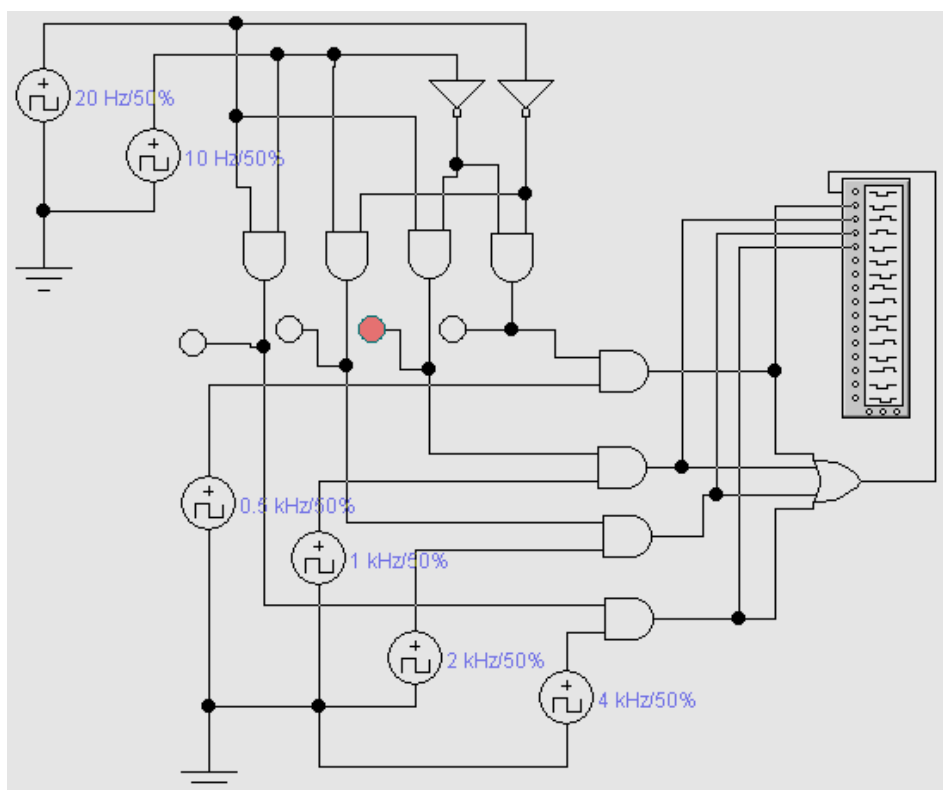


Рис. 35 – Схема исследования 4-х канального мультиплексора

1) В панели Digital нажать кнопку MUX и выбрать микросхему 74151. Собрать схему исследования, как показано на рис. 36, используя для отображения кодовых последовательностей логический анализатор. В схеме 36 генераторы тактовых импульсов с частотами 0,5 КГц, 1 КГц, 2 КГц и 4 КГц имитируют каналы передачи данных. Генераторы тактовых импульсов с частотами 10 Гц, 20 Гц и 40 Гц обеспечивают автоматическое переключение адресов для коммутации каналов.

2) Двойным щелчком открыть логический анализатор и запустить работу исследуемой схемы.

3) С помощью логического анализатора проанализировать логику работы мультиплексора. После останова логического анализатора просмотреть на его экране кодовые

последовательности входных и выходного сигналов возможно с помощью горизонтальной прокрутки. Отобразить логические последовательности, полученные с помощью анализатора, в отчет.

4) Сделайте выводы по результатам проведенного исследования.

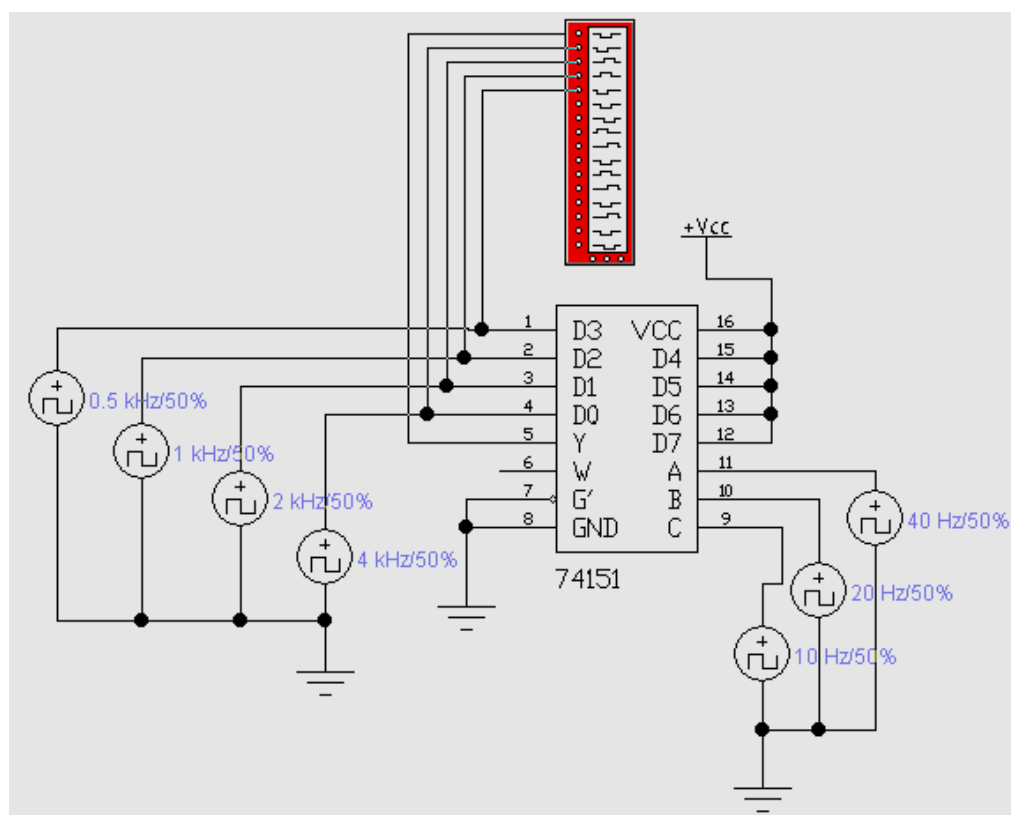


Рис. 36– Схема включения ИМС 74151 (KM155KP7)

Задание 3.7.3. Синтезировать и исследовать логику функционирования 4-х канального демультиплексора с помощью логического анализатора. Для этого выполните следующие операции:

1) Провести анализ структуры 4-х канального демультиплексора, из которого следует, что демультиплексор должен состоять из 4-х ключей (логических элементов И) и дешифратора 2х4, вырабатывающего унитарный код, который обеспечивает включение одного из 4-х ключей.

2) Собрать схему исследования 4-х канального демультиплексора, приведенную на рис. 37. Генератор тактовых импульсов с частотой 1 КГц имитирует канал передачи данных. Генераторы тактовых импульсов с частотами 10 Гц и 20 Гц обеспечивают автоматическое переключение адресов для коммутации каналов. Логические пробники предназначены для определения номера коммутируемого канала.

3) Двойным щелчком открыть логический анализатор и запустить работу исследуемой схемы.

4) С помощью логического анализатора и логических пробников проанализировать логику работы демультиплексора. После останова логического анализатора просмотреть на его экране кодовые последовательности входных и выходного сигналов возможно с помощью горизонтальной прокрутки. Отобразить логические последовательности, полученные с помощью анализатора, в отчет.

5) Сделайте выводы по результатам проведенного исследования.

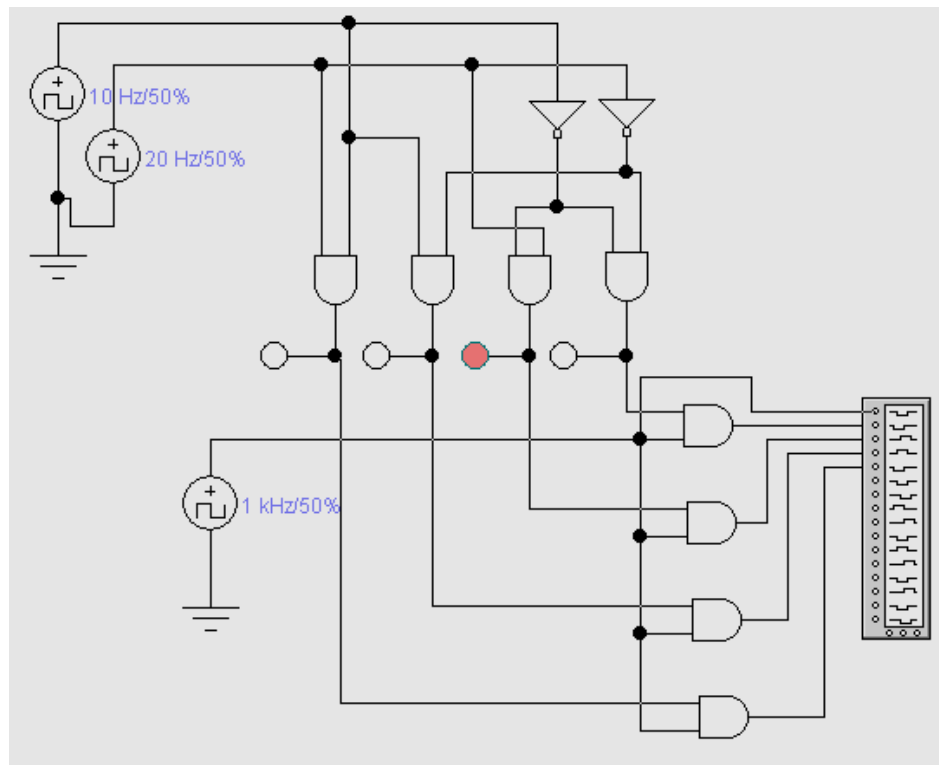


Рис. 37 – Схема исследования 4-х канального демультиплексора

Контрольные вопросы:

- 1) Перечень и характеристика комбинационных цифровых устройств.
- 2) Характеристика принципов построения и работы арифметических сумматоров.
- 3) Характеристика принципов построения и работы дешифраторов и шифраторов.
- 4) Характеристика принципов построения и работы мультиплексоров и демультиплексоров.
- 5) Порядок исследования внутренней структуры и логики функционирования полусумматора и полного сумматора с помощью логического преобразователя и генератора слова.
- 6) Порядок исследования внутренней структуры и логики функционирования 4-х разрядного сумматора с помощью логического преобразователя и генератора слова.
- 7) Порядок синтеза и исследования логики функционирования дешифратора 3x8 с помощью логического преобразователя и генератора слова.
- 8) Порядок исследования логики функционирования дешифратора 4x16, реализованного в виде ИМС 74154 (К155ИД3) с помощью генератора слова.
- 9) Порядок синтеза и исследования логики функционирования шифратора 4x2 с помощью генератора слова.
- 10) Порядок исследования логики функционирования шифратора 8x3, реализованного в виде ИМС 74148 (КМ555ИВ1) с помощью генератора слова.
- 11) Порядок синтеза и исследования логики функционирования 4-х канального мультиплексора с помощью логического анализатора.
- 12) Порядок исследования логики функционирования восьмиканального селектора-мультиплексора, реализованного в виде ИМС 74151 (КМ155КП7) с помощью логического анализатора.
- 13) Порядок синтеза и исследования логики функционирования 4-х канального демультиплексора с помощью логического анализатора.

Раздел № 3. БАЗОВАЯ ОРГАНИЗАЦИЯ ЭВМ

**Лабораторная работа №5 Моделирование и исследование арифметико-логического устройства
(2 часа)**

Содержание:

1. Моделирование арифметико-логического устройства.
2. Исследование арифметико-логического устройства.

Цель работы:

В результате выполнения крактического занятия студенты должны сформировать умение применять основные центральные устройства ПЭВМ для решения задач получения, хранения, обработки и передачи информации, а также привить навык анализа и синтеза основных центральных устройств ЭВМ с помощью программы схемотехнического моделирования Electronics Workbench.

Теоретические сведения:

Арифметико-логическое устройство (АЛУ) – центральная часть процессора, выполняющая арифметические и логические операции. АЛУ реализует важную часть процесса обработки данных. Она заключается в выполнении набора простых операций. Более сложные операции выполняются программно.

Для проведения исследований в библиотеке программы EWB имеется четырехразрядное АЛУ на базе серийной микросхемы 74181 (отечественный аналог К155ИПЗ). ИМС 74181 обеспечивает 32 режима работы АЛУ в зависимости от состояния управляющих сигналов на входах М, S0 ... S3, а также допускает наращивание разрядности с помощью входа CN и выхода CN+4, предназначенных для организации переносов. Вход М предназначен для включения логических (М=1) или арифметико-логических (М=0) операций. Входы S0 ... S3 предназначены для подачи в АЛУ кодов операций, которые в реальных АЛУ поступают из микропрограммного устройства управления процессора.

На рис. 1 представлена типовая схема включения ИМС 74181. Она позволяет оперативно реализовать все упоминавшиеся режимы.

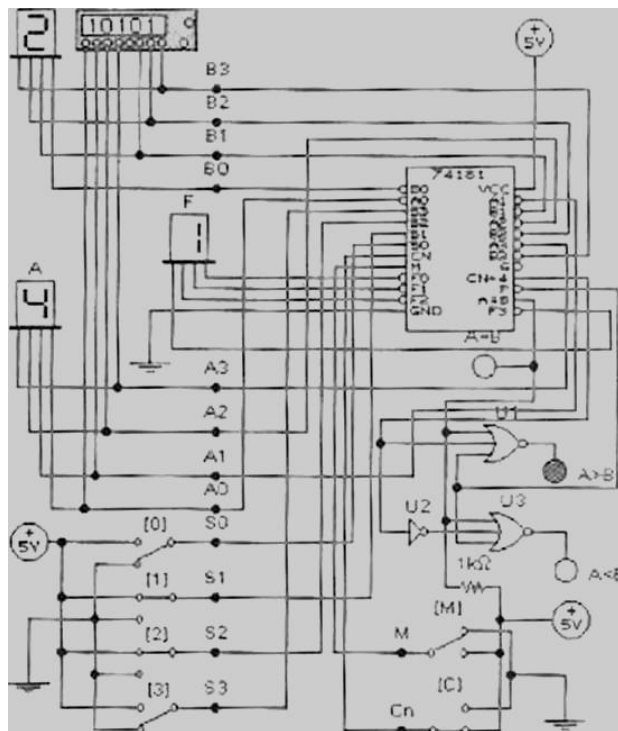


Рис. 1 – Схема включения АЛУ на ИМС 74181 (К155ИПЗ)

Возможные режимы управления АЛУ на рис.1 задаются с помощью переключателей 0, 1, 2, 3 для подачи сигналов логического нуля («земля») или логической единицы (+5 В) на входы управления S0, S1, S2, S3. В положении переключателя М, показанном на рис. 1 (сигнал 0 на входе М), выполняются 16 арифметических или арифметико-логических операций, поскольку 16 комбинаций сигналов S0 ... S3, с учетом переноса по входу CN, когда переключатель С в показанном на рис. 1 положении, т.е. CN=1, или без учета переноса, когда переключатель С подает на вход CN сигнал 0. При переводе ключа М в другое положение на входе М появится сигнал логической единицы, тогда выполняются 16 логических операций, задаваемых теми же переключателями 0 ... 3.

Значения четырехразрядных операндов А и В задаются с помощью генератора слова и в шестнадцатеричном коде отображаются на алфавитно-цифровых индикаторах. На выходах F0...F3 результат операции отображается индикатором F. При коде 1111 на этих выходах и при равенстве операндов выход А=В переводится в единичное состояние. Поскольку этот выход представляет собой каскад с открытым коллектором, то на него подается питание +5В через резистор 1 кОм. Выход А=В совместно с выходом переноса CN+4 и выходом Р подтверждения переноса используются для формирования признаков А>В и А<В с помощью дополнительных логических элементов U1, U2, U3.

Изменяя состояния сигналов на управляющих входах, можно промоделировать большинство функций АЛУ, используемых в микропроцессорах. Перечень этих функций приведен в таблице 1.

Таблица 1 – Перечень операций, реализуемых АЛУ на ИМС 74181.

№ состояния	набор управляющих сигналов				M=1	M=0
	S3	S2	S1	S0	сигнал на выходе F_i , соответствующей логической функции	результат арифметической или арифметико-логической операции
0	0	0	0	0	$\overline{A_i}$	$A + C_0$
1	0	0	0	1	$\overline{A_i} \vee \overline{B_i}$	$(A \vee B) + C_0$
2	0	0	1	0	$\overline{A_i} \cdot B_i$	$(A \vee \neg B) + C_0$
3	0	0	1	1	0	$2^4 - 1 + C_0$ (0 при $C_0=1$)
4	0	1	0	0	$\overline{A_i} \cdot B_i$	$A + (A \wedge \neg B) + C_0$
5	0	1	0	1	$\overline{B_i}$	$(A \vee B) + (A \wedge \neg B) + C_0$
6	0	1	1	0	$A_i + B_i$	$A - B - 1 + C_0$
7	0	1	1	1	$A_i \cdot \overline{B_i}$	$(A \wedge \neg B) - 1 + C_0$
8	1	0	0	0	$\overline{A_i} \vee B_i$	$A + (A \wedge B) + C_0$
9	1	0	0	1	$\overline{A_i} \vee \overline{B_i}$	$A + B + C_0$
10	1	0	1	0	B_i	$(A \vee \neg B) + (A \wedge B) + C_0$
11	1	0	1	1	$A_i \cdot B_i$	$(A \wedge B) - 1 + C_0$
12	1	1	0	0	1	$A + A + C_0$ (СДЛП при $C_0=0$)
13	1	1	0	1	$A_i \vee \overline{B_i}$	$(A \vee B) + A + C_0$
14	1	1	1	0	$A_i \vee B_i$	$(A \vee \neg B) + A + C_0$
15	1	1	1	1	A_i	$A - 1 + C_0$

При моделировании следует иметь ввиду, что логические функции выполняются поразрядно, при этом переносы не учитываются.

Рассмотрим логические операции:

Код 0000 на входах S3, S2, S1, S0; при этом выполняется логическая функция A', т.е. данные с входов A передаются на выходы F с инверсией. Этот код может быть использован в команде CMA (здесь и далее используется мнемоника команд микропроцессоров семейства 80xx фирмы Intel).

0001 – $(A+B)'$ – поразрядная операция ИЛИ с инверсией результата над операндами A и B.

0010 – $A'B$ – операция И инвертированного операнда A и операнда B.

0011 – 0 – нет операции, во всех разрядах записан 0.

0100 – $(AB)'$ – операция И над операндами A и B с инверсией.

0101 – B' – инверсия операнда B.

0110 – $A \oplus B$ – операция ИСКЛЮЧАЮЩЕЕ ИЛИ над операндами A и B, команда XRA.

0111 – AB' – операция И над операндами A и инверсией операнда B.

1000 – $A'+B$ – операция ИЛИ над инверсией операнда A и операндом B.

1001 – $(A \oplus B)'$ – операция ИСКЛЮЧАЮЩЕЕ ИЛИ с инверсией.

1010 – B – передача на выход операнда B.

1011 – AB – операция И над операндами A и B, команда ANA.

1100 – 1 – нет операций, во всех разрядах записана 1.

1101 – $A+B'$ – операция ИЛИ над инверсией операнда B и операндом A.

1110 – $A+B$ – операция ИЛИ над операндами A и B, команда ORA.

1111 – A – передача на выход операнда A.

Особенность арифметических операций состоит в том, что они могут выполняться без переноса, когда вход CN=1, и с переносом, когда вход CN=0. В первом случае перенос $Co = 0$, а во втором случае – $Co = 1$.

Анализ таблицы 1 показывает, что к непосредственно арифметическим операциям относятся:

- команда инкремента, т.е. $A+1$ – суммирование операнда A с 1 переноса при коде операции (КОП) равном 0 и CN=0;
- команда декремента, т.е. $A-1$ – вычитание из операнда A единицы переноса при КОП=15 и CN=1;
- команда ADD суммирования без учета переноса, т.е. $A+B$ – суммирования операндов A и B без учета переносов при КОП=9 и CN=1;
- команда ADC суммирования с учетом переноса, т.е. $A+B+Co$ – суммирования операндов A и B с учетом переносов при КОП=9 и CN=0;
- команда SUB вычитания, т.е. $A-B$ – вычитание из операнда A операнда B при КОП=6 и CN=0;
- команда SBB вычитания, т.е. $A-B-1$ – вычитание из разности операндов A и B единицы при КОП=6 и CN=1;
- особая команда вычитание 1 из числа 2^4 без учетов переносов при КОП=3 и CN=0;
- особая команда вычитание 1 из числа 2^4 с учетом переносов при КОП=3 и CN=1.

Остальные операции при M=0 относятся к арифметико-логическим операциям. Например, при КОП=8 производится операция суммирования операнда A с результатом поразрядного логического умножения операндов A и B. Далее от комментариев воздержусь, предоставив возможность обучаемым самостоятельно разобраться с оставшимися арифметико-логическими операциями, в надежде на то, что из вышеизложенного примера все и так очевидно.

Задание для выполнения работы:

3.1. Ознакомление с принципами функционирования ИМС 74181 (К155ИПЗ)

Задание 3.1.1. Изучить и записать в отчет перечень операций, реализуемых АЛУ на ИМС 74181.

3.2. Исследование АЛУ на базе ИМС 74181 (К155ИПЗ) с помощью программы схемотехнического моделирования Electronics Workbench

Задание 3.2.1. Исследовать логику функционирования АЛУ на основе ИМС 74181 (К155ИПЗ) из библиотеки EWB с помощью генератора слова и семисегментных индикаторов. Для этого выполните следующие операции:

1) Запустить программу Electronics Workbench и в его рабочем поле собрать схему исследования, приведенную на рис. 2.

Генератор слова в данной схеме имитирует микропрограммное устройство управления, вырабатывая код операции, а также регистры процессора, из которых в АЛУ поступают операнды. Семисегментные индикаторы включены в схему для удобства анализа кодов операндов А и В, кодов операций S и кодов результатов операций F.

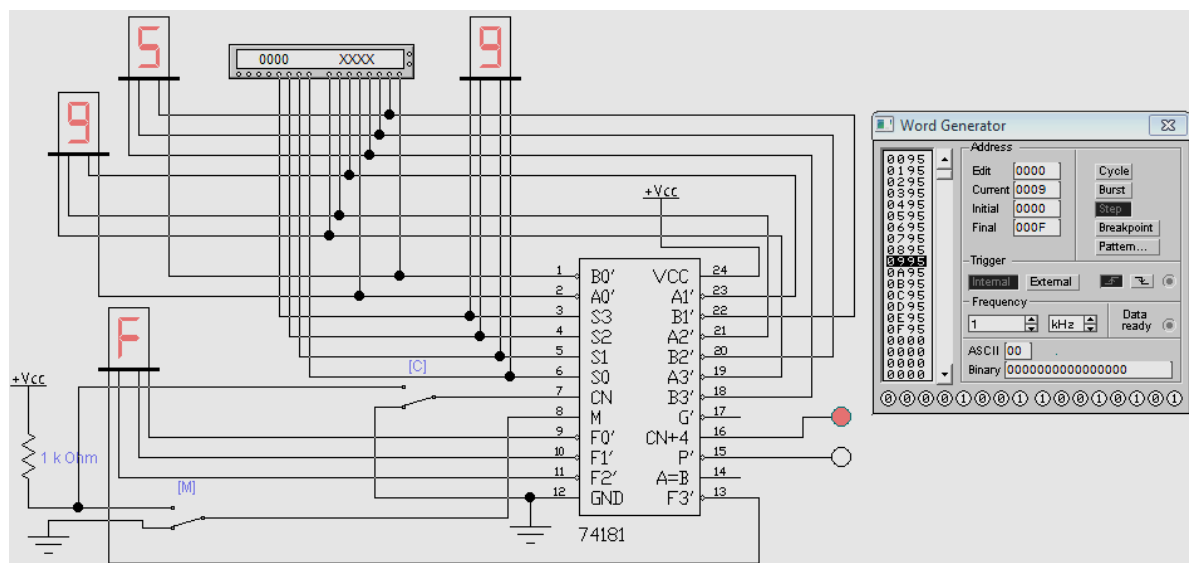


Рис. 2 – Схема исследования АЛУ

2) Запрограммировать генератор слова, записав во второй полубайт слева коды операций 0 ... 15, во второй полубайт справа код операнда А и в крайний справа полубайт код операнда В. Коды операндов взять из таблицы 2, предварительно получив от преподавателя номер варианта индивидуального задания.

Таблица 2 – Исходные данные для индивидуального задания

№ варианта	Код операнда А	Код операнда В
1	F	A
2	D	B
3	E	8
4	B	9
5	C	7
6	A	6
7	9	7
8	7	3
9	8	5
10	6	4

3) Установить выключатели М и С так, чтобы на входы М и CN подавались логические единицы.

4) Запустить генератор слова в пошаговом режиме и выполнить все логические операции. Проверить результаты вычислений и занести их таблицу выходных данных.

Таблица 3 – Выходных данных

КОП	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
формула																
код А																
код В																
код F																

5) Установить выключатели М и С так, чтобы вход М=0, а CN=1.

6) Запустить генератор слова в пошаговом режиме и выполнить все арифметико-логические операции без учета переносов. Проверить результаты вычислений и занести их таблицу выходных данных.

7) Установить выключатели М и С так, чтобы вход М=0 и CN=0.

8) Запустить генератор слова в пошаговом режиме и выполнить все арифметико-логические операции с учетом переносов. Проверить результаты вычислений и занести их таблицу выходных данных.

9) Сделайте выводы по результатам проделанной работы.

Контрольные вопросы:

- 1) . Предназначение и принципы работы арифметико-логического устройства.
- 2) Логические операции, реализуемые АЛУ на ИМС 74181 (К155ИПЗ).
- 3) Арифметические операции, реализуемые АЛУ на ИМС 74181 (К155ИПЗ).
- 4) Порядок исследования логики функционирования АЛУ на основе ИМС 74181 (К155ИПЗ) из библиотеки EWB с помощью генератора слова и семисегментных индикаторов.

Лабораторная работа №6 Моделирование и исследование оперативной и постоянной памяти (2 часа)

Содержание:

1. Моделирование оперативной и постоянной памяти
2. Исследование оперативной и постоянной памяти

Цель работы:

В результате выполнения крактического занятия студенты должны сформировать умение применять основные центральные устройства ПЭВМ для решения задач получения, хранения, обработки и передачи информации, а также привить навык анализа и синтеза основных центральных устройств ЭВМ с помощью программы схемотехнического моделирования Electronics Workbench.

Теоретические сведения:

Оперативные запоминающие устройства (ОЗУ) являются неотъемлемой частью ПЭВМ различного назначения; они делятся на два класса: статические и динамические. В статических ОЗУ запоминание информации производится на триггерах, а в динамических – на конденсаторах емкостью порядка 0,5 пФ. Длительность хранения информации в статических ОЗУ не ограничена, тогда как в динамических она ограничена временем саморазряда конденсатора, что требует специальных средств регенерации и дополнительных затрат времени на этот процесс (около 5... 12% времени работы системы).

Эволюция микросхем динамического ОЗУ вплотную связана с эволюцией ПЭВМ. В

первых IBM PC ОЗУ было выполнено на микросхемах емкостью 16 Кбит каждая. Спустя год после представления IBM PC/XT (первого ПК) появились микросхемы емкостью 64 Кбит, причем одна такая микросхема была дешевле чем 4 по 16 Кбит. В 1984 г. был сделан еще один шаг по увеличению объема памяти в одном корпусе – появились 256-килобитные микросхемы, которые были установлены на первых IBM PC/AT. Затем были разработаны микросхемы емкостью 1 Мбит, в дальнейшем начали появляться компоненты динамических ОЗУ в других конструктивных исполнениях и с непрерывной тенденцией увеличения емкости.

Конструктивно любое ОЗУ состоит из двух блоков – матрицы запоминающих элементов и дешифратора адреса. По технологическим соображениям матрица чаще всего имеет двухкоординатную дешифрацию адреса – по строкам и столбцам. На рис. 3 показана матрица 16-битного статического ОЗУ: она состоит из 16 ячеек памяти mem_i , схема которой приведена на рис. 4.

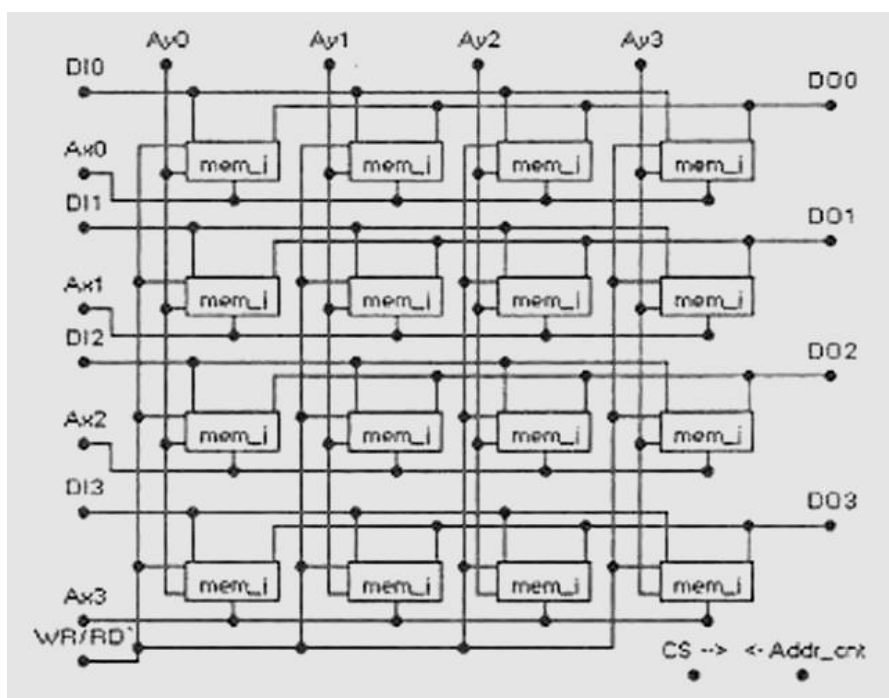


Рис. 3 – Матрица 16-битного ОЗУ

Каждая ячейка памяти адресуется по входам X, Y путем выбора дешифраторами адресных линий по строкам $Ax0...Ax3$ и по столбцам $Ay0...Ay3$ (см. рис. 4) и подачи по выбранным линиям сигнала логической единицы. При этом в выбранной ячейке памяти срабатывает двухвходовой элемент И ($U1$), подготавливая цепи чтения-записи информации на входных $DI0...DI3$ или выходных $DO0...DO3$ разрядных шинах. Разрешающим сигналом для выдачи адреса является CS (chip select – выбор кристалла), который подается на вход разрешения счетчика адреса ($Addr_cnt$) или такой же вход дешифраторов, подключенных к выходам счетчика. При записи в ячейку памяти (см. рис. 4) на соответствующей разрядной шине данных устанавливается 1 или 0, на входе WR/RD' – сигнал 1 и после стробирования счетчика или дешифраторов адреса сигналом CS срабатывают элементы 2И $U1, U2$. Положительный перепад сигнала с элемента $U2$ поступает на тактовый вход D-триггера $U4$, в результате чего в нем записывается 1 или 0 в зависимости от уровня сигнала на его D-входе.

При чтении из ячейки памяти на входе WR/RD' устанавливается 0, при этом срабатывают элементы $U1, U3, U5$ и на вход разрешения выхода буферного элемента $U6$ поступает разрешающий сигнал, в результате чего сигнал с Q-выхода D-триггера передается на разрядную шину данных $DO0...DO3$. Для проверки функционирования ячейки памяти используется генератор слова (рис. 4).

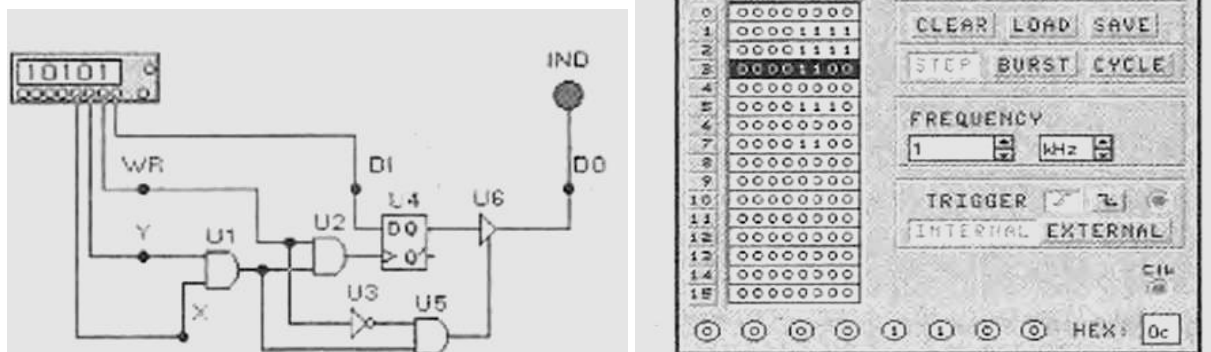


Рис. 4 – Схема ячейки памяти mem_i с лицевой панелью генератора слова с установками для ячейки памяти

Современные запоминающие устройства статического типа с произвольным доступом (Static Random Access Memory – SRAM) отличаются высоким быстродействием и в микропроцессорных системах используются ограниченно из-за сравнительно высокой стоимости. В таких системах они используются только в качестве так называемой кэш-памяти. Cache (запас) обозначает быстродействующую буферную память между процессором и основной памятью, служащую для частичной компенсации разницы в скорости процессора и основной памяти – в нее заносятся наиболее часто используемые данные. Когда процессор первый раз обращается к ячейке памяти, ее содержимое параллельно копируется в кэш, и в случае повторного обращения может быть с гораздо большей скоростью из нее извлечено. При записи в память информация попадает в кэш и одновременно копируется в память (схема Write Through – прямая или сквозная запись) или копируется через некоторое время (схема Write Back – обратная запись). При обратной записи, называемой также буферизованной сквозной записью, информация копируется в память в первом же свободном такте, а при отложенной (Delayed Write) – когда для помещения в кэш нового значения не оказывается свободной области; при этом в основное ОЗУ вытесняются сравнительно редко используемые данные. Вторая схема более эффективна, но и более сложна за счет необходимости поддержания соответствия содержимого кэша и основной памяти.

Кэш-память состоит из области данных, разбитой на блоки (строки), которые являются элементарными единицами информации при работе кэша, и области признаков (tag), описывающей состояние строк (свободна, занята, помечена для дозаписи и т.п.). В основном используются две схемы организации кэша: с прямым отображением (direct mapped), когда каждый адрес памяти может кэшироваться только одной строкой (в этом случае номер строки определяется младшими разрядами адреса), и n-связный ассоциативный (n-way associative), когда каждый адрес может кэшироваться несколькими строками. Ассоциативный кэш более сложен, однако позволяет более гибко кэшировать данные; наиболее распространены четырехсвязные системы кэширования.

Микропроцессоры 486 и выше имеют также внутренний (Internal) кэш объемом 8... 16 Кбайт. Он также обозначается как Primary (первичный) или L1 (Level 1 – первый уровень) в отличие от внешнего (External), расположенного на плате и обозначаемого Secondary (вторичный) или L2. В большинстве процессоров внутренний кэш работает по схеме с прямой записью, а в 486 (процессор Intel P24D и последние DX4-100, AMD DX4-120, 5x86) и Pentium он может работать и с отложенной записью. Последнее требует специальной поддержки со стороны системной платы, чтобы при обмене по DMA (прямое обращение к памяти устройств ввода-вывода) можно было поддерживать согласованность данных в памяти и внутреннем кэше. Процессоры Pentium Pro и выше имеют также встроенный кэш второго уровня объемом 256, 512 или 1024 Кбайт с собственным контроллером и локальной

64-разрядной шиной данных, работающей на внутренней частоте МП; использована дополнительная внутренняя оптимизация, ускорена работа конвейера и степень параллелизма, улучшена система предсказания переходов (Dynamic & Speculative Execution).

В микропроцессорных системах в качестве ОЗУ чаще всего используются динамические ОЗУ с запоминающим конденсатором, которые отличаются большим многообразием. Приведем данные по наиболее распространенным типам таких ОЗУ.

В динамической памяти (Dynamic Random Access Memory – DRAM) ячейки выполнены на основе областей с накоплением зарядов, занимающих гораздо меньшую площадь, нежели триггеры, и практически не потребляющих энергии при хранении информации. При записи бита в такую ячейку в ней формируется электрический заряд, который сохраняется в течение нескольких миллисекунд; для постоянного сохранения заряда ячейки необходимо регенерировать (перезаписывать) ее содержимое. Ячейки микросхем динамической памяти также организованы в виде прямоугольной матрицы: при обращении к микросхеме на ее входы вначале подается адрес строки матрицы, сопровождаемый сигналом RAS (Row Address Strobe – строб адреса строки), затем, через некоторое время – адрес столбца, сопровождаемый сигналом CAS (Column Address Strobe – строб адреса столбца). При каждом обращении к отдельной ячейке регенерируются все ячейки выбранной строки, поэтому для полной регенерации матрицы достаточно перебрать адреса строк. Ячейки динамической памяти имеют сравнительно малое быстродействие (десятки – сотни наносекунд), но большую удельную плотность (порядка нескольких мегабайт на корпус) и меньшее энергопотребление.

Обычные ОЗУ называют часто асинхронными, так как установка адреса и подача управляющих сигналов могут выполняться в произвольные моменты времени, необходимо только соблюдение временных соотношений между этими сигналами. В них включены так называемые охранные интервалы, необходимые для установления сигналов. Существуют также синхронные виды памяти, получающие внешний синхросигнал, к импульсам которого жестко привязаны моменты подачи адресов и обмена данными: они позволяют более полно использовать внутреннюю конвейеризацию и блочный доступ.

FPM DRAM (Fast Page Mode DRAM – динамическая память с быстрым страничным доступом) отличается от обычной динамической памяти тем, что после выбора строки матрицы и удержания сигнала RAS допускает многократную установку адреса столбца, стробируемого сигналом CAS, а также быструю регенерацию по схеме «CAS прежде RAS». Первое позволяет ускорить блочные передачи, когда весь блок данных или его часть находятся внутри одной строки матрицы, называемой в этой системе страницей, а второе – снизить затраты времени на регенерацию памяти.

EDO (Extended Data Out – расширенное время удержания данных на выходе) фактически представляют собой обычные микросхемы FPM, на выходе которых установлены регистры-защелки данных. При страничном обмене такие микросхемы работают в режиме простого конвейера: удерживают на выходах данных содержимое последней выбранной ячейки, в то время как на их входы уже подается адрес следующей выбираемой ячейки. Это позволяет примерно на 15% по сравнению с FPM ускорить процесс считывания последовательных массивов данных. При случайной адресации такая память ничем не отличается от обычной.

BEDO (Burst EDO – EDO с блочным доступом) – память на основе EDO, работающая не одиночными, а пакетными циклами чтения/записи. Современные процессоры благодаря внутреннему и внешнему кэшированию команд и данных обмениваются с основной памятью преимущественно блоками слов максимальной ширины. При наличии памяти BEDO отпадает необходимость постоянной подачи последовательных адресов на входы микросхем с соблюдением необходимых временных задержек, достаточно стробировать переход к очередному слову отдельным сигналом.

SDRAM (Synchronous DRAM – синхронная динамическая память) – память с

синхронным доступом, работающая быстрее обычной асинхронной (FPM/EDO/BEDO). Кроме синхронного доступа, SDRAM использует внутреннее разделение массива памяти на два независимых банка, что позволяет совмещать выборку из одного банка с установкой адреса в другом. SDRAM также поддерживает блочный обмен. Основное преимущество SDRAM состоит в поддержке последовательного доступа в синхронном режиме, где не требуется дополнительных тактов ожидания. При случайном доступе SDRAM работает практически с той же скоростью, что и FPM/EDO.

PB SRAM (Pipelined Burst SRAM – статическая память с блочным конвейерным доступом) – разновидность синхронных SRAM с внутренней конвейеризацией, за счет которой примерно вдвое повышается скорость обмена блоками данных.

DDR SDRAM (Double Data Rate SDRAM – память SDRAM с удвоенной скоростью передачи данных) – первоначально память такого типа применялась в видеоплатах, но позднее появилась поддержка DDR SDRAM со стороны чипсетов. Память DDR SDRAM работает на частотах в 100, 133, 166 и 200 МГц, её время полного доступа – 30 и 22,5 нс, а время рабочего цикла – 5; 3,75; 3 и 2,5 нс. Так как частота синхронизации лежит в пределах от 100 до 200 МГц, а данные передаются по 2 бита на один синхроимпульс, как по фронту, так и по срезу тактового импульса, то эффективная частота передачи данных лежит в пределах от 200 до 400 МГц. Такие модули памяти обозначаются DDR200, DDR266, DDR333, DDR400.

DR DRAM (Direct Rambus DRAM) – разновидность синхронной памяти; снабжена специальным интерфейсом фирмы Rambus, для связи с контроллером памяти используется специальная быстродействующая шина Rambus Channel, допускающая использование тактовой частоты 400 МГц и обмен по обоим фронтам импульсов, т.е. обмен с частотой 800 МГц, что соответствует пропускной способности 1,6 Гбит/с; при максимально возможном количестве каналов 4 пропускная способность возрастает до 6,4 Гбит/с.

DDR2 SDRAM – конструктивно новый тип оперативной памяти DDR2 SDRAM был выпущен в 2004 году. Основываясь на технологии DDR SDRAM, этот тип памяти за счёт технических изменений показывает более высокое быстродействие и предназначен для использования на современных компьютерах. Память может работать с тактовой частотой шины 200, 266, 333, 337, 400, 533, 575 и 600 МГц. При этом эффективная частота передачи данных соответственно будет 400, 533, 667, 675, 800, 1066, 1150 и 1200 МГц. Некоторые производители модулей памяти помимо стандартных частот выпускают и образцы, работающие на нестандартных (промежуточных) частотах. Они предназначены для использования в разогнанных системах, где требуется запас по частоте. Время полного доступа – 25; 11,25; 9; 7,5 нс и менее. Время рабочего цикла – от 5 до 1,67 нс.

DDR3 SDRAM – этот тип памяти основан на технологиях DDR2 SDRAM со вдвое увеличенной частотой передачи данных по шине памяти. Отличается пониженным энергопотреблением по сравнению с предшественниками. Частота полосы пропускания лежит в пределах от 800 до 2400 МГц (рекорд частоты – более 3000 МГц), что обеспечивает большую пропускную способность по сравнению со всеми предшественниками.

Кроме основного ОЗУ, устройством памяти снабжается и устройство отображения информации – видеографическая система. Такая память называется видеопамью и располагается на плате видеоадаптера.

Видеопамью служит для хранения изображения. От ее объема зависит максимально возможное разрешение видеокарты – $A \times B \times C$, где A – количество точек по горизонтали, B – по вертикали, C – количество возможных цветов каждой точки. Например, для разрешения $640 \times 480 \times 16$ достаточно иметь видеопамью 256 Кбайт, для $800 \times 600 \times 256$ – 512 КБ, для $1024 \times 768 \times 65536$ (другое обозначение – $1024 \times 768 \times 64k$) – 2 Мбайт и т.д. Поскольку для хранения цветов отводится целое число разрядов, количество цветов всегда является целой степенью 2 (16 цветов – 4 разряда, 256 – 8 разрядов, 64k – 16 и т.д.).

В видеоадаптерах используются следующие типы видеопамью.

FPM DRAM (Fast Page Mode Dynamic RAM – динамическое ОЗУ с быстрым страничным доступом) — основной тип видеопамяти, идентичный используемой в системных платах. Активно применялась до 1996 г. Наиболее распространенные микросхемы FPM DRAM – четырехразрядные DIP и SOJ, а также шестнадцатиразрядные SOJ.

VRAM (Video RAM – видео-ОЗУ) – так называемая двухпортовая DRAM с поддержкой одновременного доступа со стороны видеопроцессора и центрального процессора компьютера. Позволяет совмещать во времени вывод изображения на экран и его обработку в видеопамяти, что сокращает задержки и увеличивает скорость работы.

EDO DRAM (Extended Data Out DRAM – динамическое ОЗУ с расширенным временем удержания данных на выходе) – память с элементами конвейеризации, позволяющей несколько ускорить обмен блоками данных с видеопамятью.

SGRAM (Synchronous Graphics RAM – синхронное графическое ОЗУ) – вариант DRAM с синхронным доступом, когда все управляющие сигналы изменяются одновременно с системным тактовым синхросигналом, что позволяет уменьшить временные задержки.

WRAM (Window RAM – оконное ОЗУ) – EDO VRAM, в котором окно, через который обращается видеоконтроллер, сделано меньшим, чем окно для центрального процессора.

MDRAM (Multibank DRAM – многобанковое ОЗУ) – вариант DRAM, организованный в виде множества независимых банков объемом по 32 Кбайт каждый, работающих в конвейерном режиме.

Увеличение скорости обращения видеопроцессора к видеопамяти, кроме повышения пропускной способности адаптера, позволяет повысить максимальную частоту регенерации изображения, что снижает утомляемость глаз оператора.

Микросхемы памяти имеют четыре основные характеристики – тип, объем, структуру и время доступа. Тип обозначает статическую или динамическую память, объем показывает общую емкость памяти, а структура – количество ячеек памяти и разрядность каждой ячейки. Например, 28/32-выводные DIP-микросхемы SRAM имеют 8-разрядную структуру (8К x 8, 16К x 8, 32К x 8, 64К x 8, 128К x 8), кэш объемом 256 Кбайт состоит из восьми микросхем 32К x 8 или четырех микросхем 64К x 8 (речь идет об области данных, дополнительные микросхемы для хранения признаков могут иметь другую структуру). Две микросхемы по 128К x 8 поставить уже нельзя, так как нужна 32-разрядная шина данных, что могут обеспечить только четыре микросхемы. Распространенные PB SRAM в 100-выводных корпусах PQFP имеют 32-разрядную структуру 32К x 32 или 64К x 32 и используются по две или по четыре в платах для Pentium.

30-контактные SIMM имеют 8-разрядную структуру и используются с процессорами 286, 386SX и 486SLC по две, а с 386DX, 486DX и обычными 486DX – по четыре. 72-контактные SIMM имеют 32-разрядную структуру и могут использоваться с 486DX по одной, а с Pentium и Pentium Pro – по две. 168-контактные DIMM имеют 64-разрядную структуру и используются в Pentium и Pentium Pro по одной. Установка модулей памяти или микросхем кэша в количестве больше минимального для данной системной (материнской) платы позволяет ускорить работу с ними, используя принцип чередования (Interleave).

Время доступа характеризует скорость работы микросхемы и обычно указывается в наносекундах после тире в конце наименования. На более медленных микросхемах могут указываться только первые цифры (7 вместо 70, 15 вместо 150), на более быстрых статических "15" или "20" обозначает реальное время доступа к ячейке. Часто на микросхемах указывается минимальное из всех возможных времен доступа, например, распространена маркировка 50 EDO DRAM вместо 70, или 45 – вместо 60, хотя такой цикл достижим только в блочном режиме, а в одиночном режиме микросхема по-прежнему имеет время доступа 70 или 60 нс. Аналогичная ситуация имеет место и в маркировке PB SRAM: 6 вместо 12 и 7 вместо 15. Микросхемы SDRAM обычно маркируются временем доступа в блочном режиме (10 или 12 нс).

ИМС памяти реализуются в корпусах следующих типов:

DIP (Dual In line Package – корпус с двумя рядами выводов) – классические микросхемы, применявшиеся в блоках основной памяти IBM PC/XT и ранних PC/AT, сейчас применяются в блоках кэш-памяти.

SIP (Single In line Package – корпус с одним рядом выводов) – микросхема с одним рядом выводов, устанавливаемая вертикально.

SIPP (Single In line Pinned Package – модуль с одним рядом игольчатых выводов) – модуль памяти, вставляемый в панель наподобие микросхем DIP/SIP; применялся в ранних IBM PC/AT.

SIMM (Single In line Memory Module – модуль памяти с одним рядом контактов) – модуль памяти, вставляемый в зажимный разъем; применяется во всех современных платах, а также во многих адаптерах, принтерах и прочих устройствах. SIMM имеет контакты с двух сторон модуля, но все они соединены между собой, образуя как бы один ряд контактов. На SIMM в настоящее время устанавливаются преимущественно микросхемы FPM/EDO/BEDO.

DIMM (Dual In line Memory Module – модуль памяти с двумя рядами контактов) – модуль памяти, похожий на SIMM, но с отдельными контактами (обычно 2 x 84), за счет чего увеличивается разрядность или число банков памяти в модуле. На DIMM устанавливаются микросхемы EDO/BEDO/SDRAM/DDR. Модули памяти типа SDRAM наиболее распространены в виде 168-контактных DIMM-модулей, памяти типа DDR SDRAM – в виде 184-контактных, а модули типа DDR2, DDR3 – 240-контактных модулей.

CELP (Card Edge Low Profile – невысокая карта с ножевым разъемом на краю) – модуль внешней кэш-памяти, собранный на микросхемах SRAM (асинхронный) или PB SRAM (синхронный). По внешнему виду похож на 72-контактный SIMM, имеет емкость 256 или 512 Кбайт. Другое название – COAST (Cache On A STick – буквально «кэш на палочке»).

SO-DIMM (Small outline DIMM) – аналоги модулей DIMM в компактном исполнении для экономии места, предназначены для портативных и компактных устройств (материнских плат форм-фактора Mini-ITX, ноутбуков, планшетов и т.п.), а также принтеров, сетевой и телекоммуникационной техники.

RIMM (Rambus In-line Memory Module) – модуль памяти для DR DRAM. Они представлены 168- и 184-контактными разновидностями, причём на материнской плате такие модули обязательно должны устанавливаться только в парах, в противном случае в пустые разъёмы устанавливаются специальные модули-заглушки (это связано с особенностями конструкции таких модулей).

Модули динамической памяти кроме основных ячеек памяти, могут иметь дополнительные ячейки для хранения битов четности (Parity) для байтов данных; такие SIMM иногда называют 9- и 36-разрядными модулями (по одному биту четности на байт данных). Биты четности служат для контроля правильности считывания данных из модуля, позволяя обнаружить часть ошибок. Модули с битами четности имеет смысл применять лишь там, где требуется повышенная надежность. Для обычных применений подходят и тщательно проверенные модули без битов четности, однако при условии, что системная плата поддерживает такие типы модулей.

Проще всего определить тип модуля по маркировке и количеству микросхем памяти на нем: например, если на 30-контактном SIMM две микросхемы одного типа и еще одна – другого, то две первых являются основными (каждая – по четыре разряда), а третья предназначена для хранения битов четности (она одноразрядная). В 72-контактном SIMM с двенадцатью микросхемами восемь из них хранят данные, а четыре – биты четности. Модули с количеством микросхем 2, 4 или 8 не имеют памяти для хранения битов четности.

Иногда на модули ставится так называемый имитатор четности – микросхема-сумматор, выдающая при считывании ячейки всегда правильный бит четности; предназначена для установки в платы, где проверка четности не отключается.

72-контактные SIMM имеют четыре специальные линии PD (Presence Detect –

обнаружение наличия), на которых при помощи перемычек может быть установлено до 16 комбинаций сигналов. Линии PD используются в некоторых материнских платах для определения наличия модулей памяти в разъемах и их параметров (объема и быстродействия). В большинстве универсальных плат производства «третьих фирм», как и выпускаемые ими SIMM, линии PD не используются.

В модулях DIMM в соответствии со спецификацией JEDEC технология PD реализуется при помощи перезаписываемого ПЗУ с последовательным доступом (Serial EEPROM) и носит название Serial Presence Detect (SPD). ПЗУ представляет собой 8-выводную микросхему, размещенную в углу платы DIMM, а его содержимое описывает конфигурацию и параметры модуля.

2.2. Постоянные запоминающие устройства

Постоянные запоминающие устройства (ПЗУ) делятся на четыре типа:

- 1) масочные, программируемые на заводе-изготовителе с применением специальных масок;
- 2) однократно программируемые потребителем путем пережигания нихромовых или поликремневых перемычек;
- 3) многократно программируемые потребителем со стиранием записанной информации ультрафиолетовым излучением;
- 4) многократно программируемые потребителем с электрическим стиранием информации.

Рассмотрим ПЗУ второго типа, которое состоит из дешифратора $n \times 2^n$ и подключенных к его выходам схем ИЛИ с плавкими перемычками (рис. 5). ПЗУ содержит дешифратор 2×4 в виде подсхемы *pzu_dcd* (A, B – кодовые входы, E – входы разрешения, активный сигнал высокого уровня), к выходам которых можно подключить четыре элемента 4ИЛИ с дополнительными устройствами. На рис. 5 показаны два таких элемента, выполненных в виде отдельных подсхем *pzu_ur1* и *pzu_un2*. Хотя эти элементы одинаковы, наращивание их на схеме путем копирования исключено из-за наличия пережигаемых перемычек: при наличии одноименных подсхем пережигание перемычки в одной подсхеме автоматически приведет к пережиганию такой же перемычки в другой. Поскольку программа не позволяет копировать подсхемы с их переименованием, все их приходится выполнять отдельно. На схеме D0, D1 выходы младшего и первого разрядов.

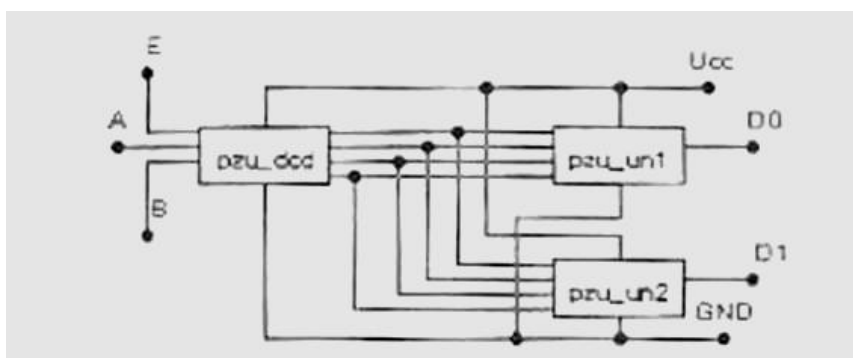


Рис. 5 – Схема двухразрядного ПЗУ

Дешифратор pzu_dcd (рис. 6) выполнен на трех элементах НЕ и четырех элементах ЗИЛИ-НЕ на транзисторах (рис. 7).

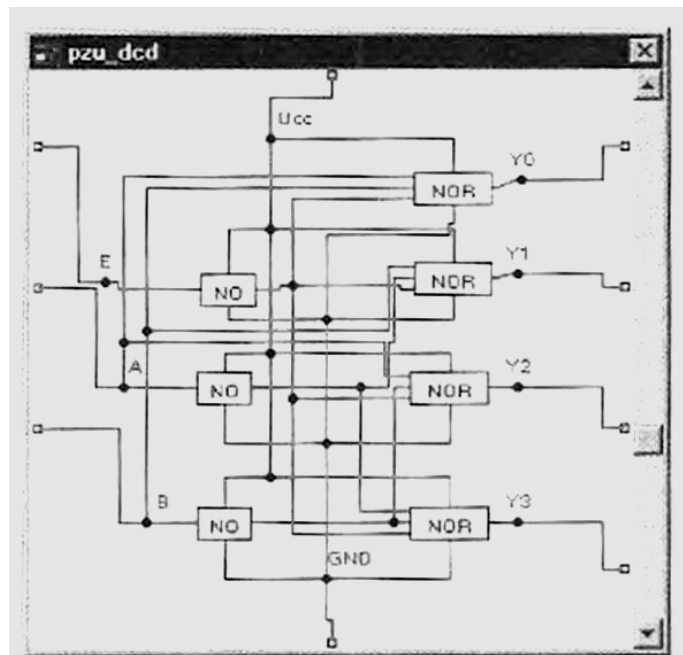


Рис. 6 – Внутренняя структура подсхемы дешифратора

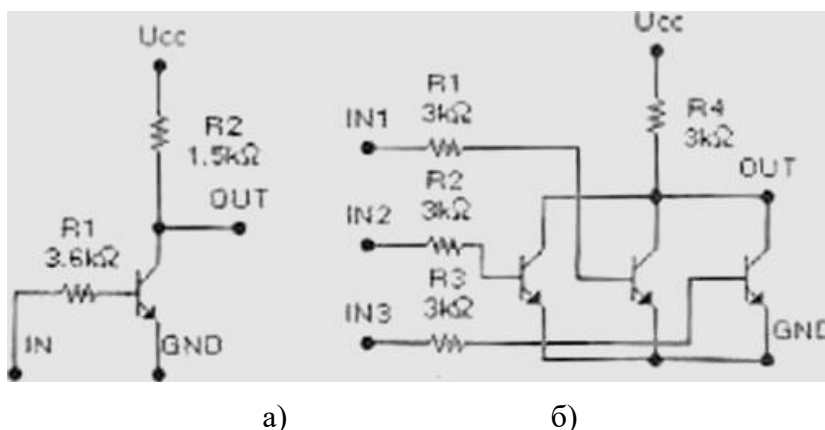


Рис. 7 – Схемы элементов НЕ (а) и ИЛИ-НЕ (б)

Необходимость выполнения элементов дешифратора на транзисторах объясняется тем, что используемые в программе EWB модели цифровых ИМС не всегда позволяют подключать к ним обычные транзисторные схемы и, в частности, применяемые в рассматриваемом ПЗУ ячейки памяти в виде подсхемы pzu_uni. Ее внутренняя структура аналогична структуре ячейки памяти, используемой в ПЗУ K155PE3 (рис. 8).

В отличие от ИМС K155PE3, в которой в качестве элементе ИЛИ применяется многоэмиттерный транзистор, на рис. 8 используются отдельные транзисторы T1...T4, эмиттеры которых через пережигаемые перемычки S1...S4 (имитируются предохранителями на 10 мА) соединены с формирователем на транзисторах T5, T6 и стабилитроне D. Транзистор T5 и стабилитрон D используются только в режиме программирования и в рабочем режиме не оказывают влияния на работу выходного каскада на транзисторе T6 (каскад с открытым коллектором), поскольку транзистор T5 закрыт низким потенциалом на его базе (напряжение пробоя стабилитрона D выбирается несколько больше напряжения питания транзистора T6, подаваемого во второй подсхеме в точку D0 или D1 через резистор нагрузки).

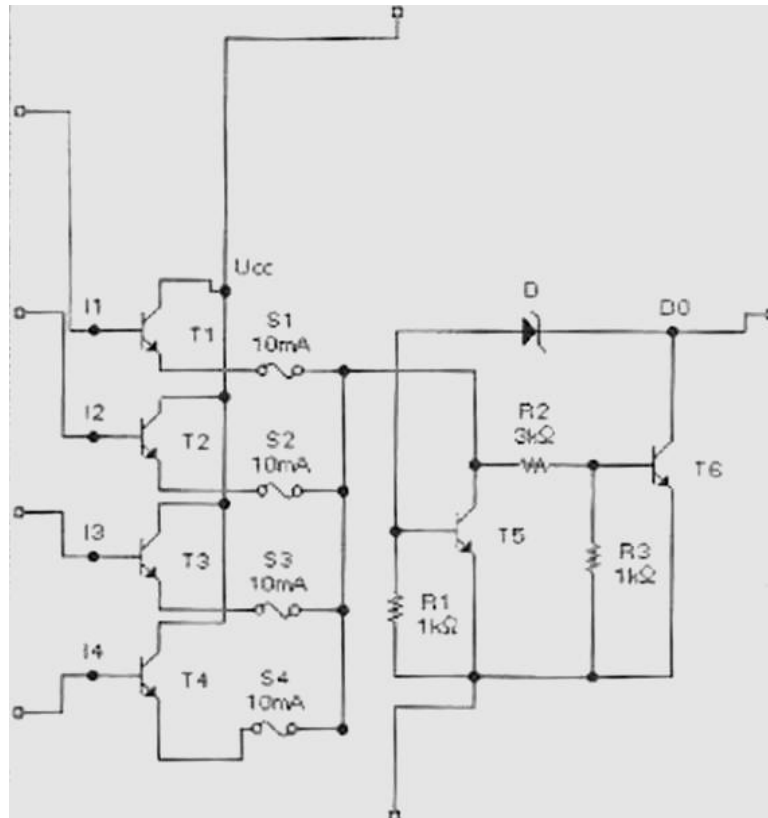


Рис. 8 – Внутренняя структура ячейки памяти двухразрядного ПЗУ

Ячейка ПЗУ работает следующим образом. В исходном состоянии транзисторы Т1 ... Т4 и Т6 закрыты, и при подключенной к Т6 нагрузке на его выходе D0 формируется сигнал логической единицы (около + 5 В). При подаче на входы А, В дешифратора заданной кодовой комбинации, а на вход разрешения Е – сигнала логической единицы, один из транзисторов Т1 ... Т4 откроется и на выходе D0 сформируется сигнал логического нуля. Так, например, при А=В=1 откроется транзистор Т4 и сигнал логической единицы с его эмиттера через переключку S4 поступит на делитель на резисторах R2, R3, транзистор Т6 откроется, и на его выходе сформируется сигнал логического нуля. Очевидно, что и при любой другой двоичной комбинации будет происходить то же самое до тех пор, пока не будет разрушена соответствующая переключка.

Пережигание переключек составляет суть программирования и осуществляется отдельно для каждого разряда (каждой ячейки) следующим образом:

на входы А, В (см. рис. 5) подается двоичная комбинация, соответствующе адресу пережигаемой переключки в программируемом разряде (в ячейке **pzu_unx**, где x – номер ячейки);

к выходу ячейки Dх через резистор нагрузки (его сопротивление для конкретных ИМС указывается в документации, для K155PE3 составляет около 300 Ом подключается источник напряжения 12,5 В, в результате чего стабилитрон D пробивается и транзистор Т5 открывается;

на вход разрешения Е на короткое время подается сигнал логической единицы при этом через один из открытых транзисторов Т1 ... Т2 и Т5 протекает ток, достаточный для пережигания соответствующей переключки (длительность разрешающего сигнала на входе Е в промышленных программаторах может автоматически увеличиваться после нескольких неудачных попыток программирования одной и той же ячейки);

источник 12,5 В отключается; после раскрытия соответствующей подсхемы можно

убедиться, что перемычка действительно разрушена (в промышленных программаторах этот процесс сводится к проверке записи программируемой ячейки и при отрицательном результате производится повторное программирование при большей длительности разрешающего сигнала).

Заключительным этапом программирования серийных микросхем ПЗУ в промышленных условиях является электротермотренировка, которая проводится чаще всего в течение 168 часов при повышенной температуре, после чего производится дополнительный контроль записанной информации. Если при этом обнаруживается ошибка, допускается повторное программирование. Если ошибка снова повторяется, микросхема бракуется.

Для моделирования процесса программирования к программируемой схеме необходимо подключить дополнительные элементы. Моделирование целесообразно начинать с одноразрядного ПЗУ (рис. 9).

Следует отметить, что рассматриваемая модель ПЗУ (как на рис. 9, так и на рис. 9) достаточно капризна и при некоторых комбинациях входных сигналов моделирование не выполняется. Признаком невозможности моделирования является отсутствие слева от включателя питания (в верхнем правом углу экрана) таблицы с индикацией временных интервалов отсчета. По истечении некоторого времени может быть выдана рекомендация изменить установку погрешности моделирования (по умолчанию она равна 1%). Целесообразно установить ее максимально возможной (10%) в меню Circuit (команда Analysis Options, параметр Tolerance). Целесообразно также варьировать сопротивлениями входных резисторов и резисторов нагрузки элементов НЕ и ИЛИ-НЕ (рис. 7), а также попробовать изменить параметры транзисторов. В крайнем случае, можно ограничиться простейшим случаем – обойтись без дешифратора и использовать только одну ячейку памяти (см. рис. 8), подключив к выходу и к одному из ее входов дополнительный элементу как показано на рис. 9.

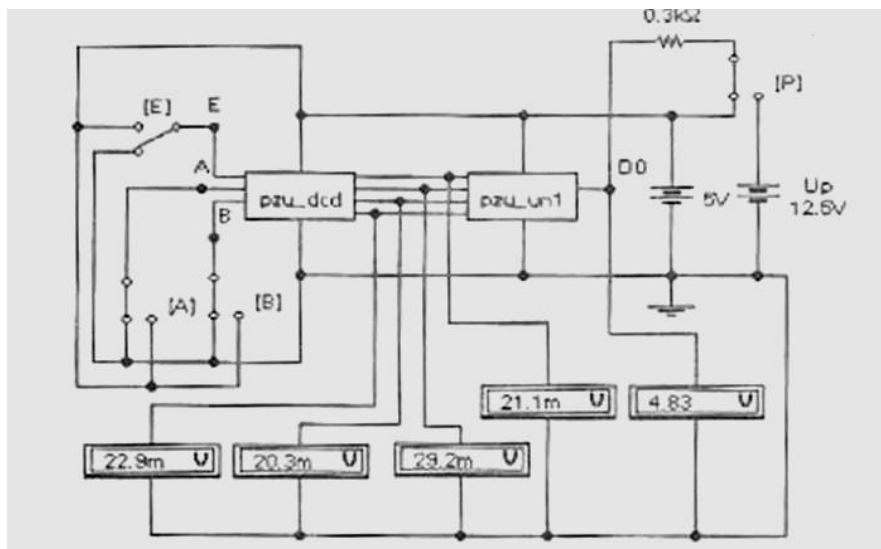


Рис. 9 – Модель ПЗУ с дополнительными элементами

ПЗУ с пережигаемыми перемычками используются чаще всего в качестве специализированных дешифраторов, например, для селекции УВВ.

ПЗУ с ультрафиолетовым стиранием используются в микропроцессорных системах для хранения управляющих программ, в частности, для размещения BIOS (Basic Input/Output System – основной системы ввода/вывода, записанной в ПЗУ, отсюда ее полное название – ROM BIOS). BIOS представляет собой набор программ проверки и обслуживания аппаратуры компьютера и выполняет роль посредника между операционной системой (ОС) и аппаратной частью ПК. BIOS получает управление при включении системной платы, тестирует саму плату и основные блоки компьютера – видеоадаптер, клавиатуру, контроллеры дисков и

портов ввода/вывода, настраивает чипсет платы и загружает внешнюю ОС. При работе под управлением DOS/Windows 3.x/95/98 BIOS управляет основными устройствами, при работе под OS/2, Unix, Windows NT BIOS практически не используется, выполняя лишь начальную проверку и настройку.

Обычно на системной плате установлено только ПЗУ с системным (Main System) BIOS, отвечающим за саму плату и контроллеры FDD (флоппи-дисков), HDD (жестких дисков), портов и клавиатуры; в системный BIOS практически всегда входит System Setup – программа настройки системы. Видеоадаптеры и контроллеры HDD с интерфейсом ST-506 (MFM) и SCSI имеют собственные BIOS в отдельных ПЗУ; их также могут иметь и другие платы – интеллектуальные контроллеры дисков и портов, сетевые карты и т.п.

Обычно BIOS для современных системных плат разрабатывается одной из специализированных фирм: Award Software, American Megatrends (AMI), реже: Phoenix Technology, Microicl Research. Некоторые производители плат (например, IBM, Intel и Acer) сами разрабатывают BIOS для них. Иногда для одной и той же платы имеются версии BIOS разных производителей, в этом случае допускается копировать прошивки или заменять микросхемы ПЗУ; в общем же случае каждая версия BIOS привязана к конкретной модели платы.

Раньше BIOS помещался в однократно программируемые ПЗУ либо ПЗУ с ультрафиолетовым стиранием; сейчас в основном выпускаются платы с электрически перепрограммируемыми ПЗУ (Flash ROM), которые допускают перепрограммирование BIOS средствами самой платы. Это позволяет исправлять заводские ошибки в BIOS, изменять заводские установки по умолчанию, программировать собственные экранные заставки и т.п.

Тип микросхемы ПЗУ обычно можно определить по маркировке: 27xxxx – обычное ПЗУ, 28xxxx или 29xxxx – перепрограммируемые. Если на корпусе микросхемы 27xxxx есть прозрачное окно – это ПЗУ с ультрафиолетовым стиранием, если его нет – это однократно программируемое ПЗУ, которое можно лишь заменить на другое.

Видео-ПЗУ (Video ROM) – постоянное запоминающее устройство, в котором записаны видео-BIOS, экранные шрифты, служебные таблицы и т.п. ПЗУ не используется видеоконтроллером напрямую, к нему обращается только центральный процессор в результате выполнения программ, записанных в том же ПЗУ. На многих современных видеокартах устанавливаются электрически перепрограммируемые ПЗУ (EEPROM, Flash ROM), допускающие перезапись пользователем под управлением специальной программы из комплекта карты.

ПЗУ необходимо только для первоначального запуска видеоадаптера и работы; в режиме DOS, Novell Netware и других ОС, функционирующих преимущественно в текстовом режиме; ОС Windows, OS/2 и им подобные, работающие через собственные видеодрайверы, не используют ПЗУ для управления адаптером либо используют его только при выполнении программ для DOS.

При создании видео-BIOS все разработчики придерживаются рекомендации VESA и VBE. VESA (Video Electronics Standards Association – ассоциация стандартизации видеоэлектроники) – организация, выпускающая различные стандарты в области электронных видеосистем и их программного обеспечения. VBE (**V**ESA BIOS **E**xtension – расширение BIOS в стандарте VESA) – дополнительные функции видео-BIOS по отношению к стандартному видео-BIOS для VGA, позволяют запрашивать у адаптера список поддерживаемых видеорежимов и их параметры (разрешение, цветность, способы адресации, развертка и т.п.) и изменять эти параметры для согласования адаптера с конкретным монитором. По сути, VBE является унифицированным стандартом программного интерфейса с VESA-совместимыми картами, при работе через видео-BIOS он позволяет обойтись без специализированного драйвера видеокарты.

Задание для выполнения работы:

3.3. Исследование ячейки оперативной памяти

Задание 3.3.1. Исследовать внутреннюю структуру и логику функционирования ячейки статической оперативной памяти с помощью генератора слова. Для этого выполните следующие операции:

1) Запустить программу Electronics Workbench и в его рабочем поле собрать схему исследования, приведенную на рис. 10.

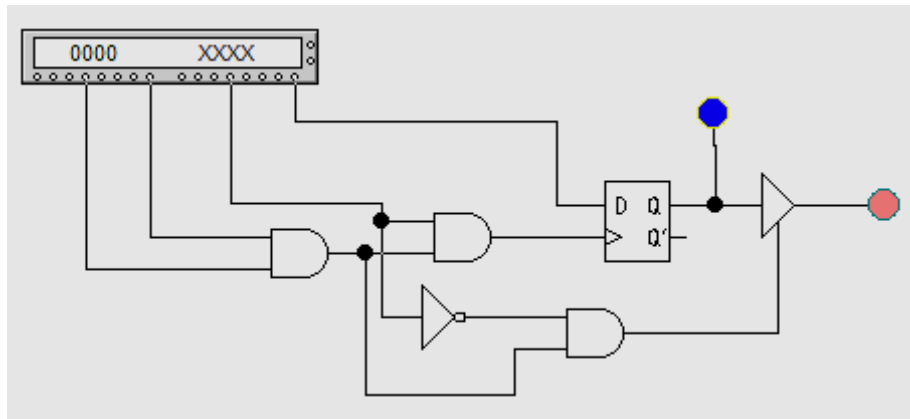


Рис. 10 – Схема исследования ячейки статической оперативной памяти

Генератор слова в данной схеме имитирует общую шину.

2) Запрограммировать генератор слова, таким образом, чтобы в младшем полубайте находился бит данных, равный 1, следующем полубайте – команды управления записью/чтением WR/RD', а в двух старших полубайтах адреса строки и столбца ячейки памяти для данных логических схем, имитирующих дешифратор адреса.

3) Запустить генератор слова в пошаговом режиме и выполнить операции записи и чтения. В отчет занести адрес данной ячейки памяти, на схеме указать сигнал выбора микросхемы CS, определить с помощью какого элемента производится считывание данных из ячейки памяти.

4) Изменить логическую схему дешифратора, настроив ее на другой адрес по своему усмотрению.

5) Повторить пункты 2 и 3.

6) Сделайте выводы по результатам проделанной работы.

Задание 3.3.2. Исследовать внутреннюю структуру и логику функционирования ячейки динамической оперативной памяти с помощью осциллографа. Для этого выполните следующие операции:

1) Запустить программу Electronics Workbench и в его рабочем поле собрать схему исследования, приведенную на рис. 11.

Данная схема состоит из схемы замещения ячейки динамической оперативной памяти, выполненной на транзисторе и конденсаторе, и схемы, моделирующей процесс регенерации записанных данных. Автоматические переключатели предназначены для моделирования процесса записи логической единицы. Автоматический выключатель, расположенный над транзистором, имитирует процедуру выбора ячейки памяти дешифратором адреса или, проще говоря, задает адрес ячейки. Автоматический выключатель, расположенный слева от транзистора имитирует подачу данных в виде логической единицы. Переключатель, управляемый клавишей Q, предназначен для подключения к ячейке схемы регенерации данных, выполненной на двух генераторах тактовых импульсов (ГТИ) и аналоговом коммутаторе. Параметры ГТИ указаны на схеме (см. рис. 11).

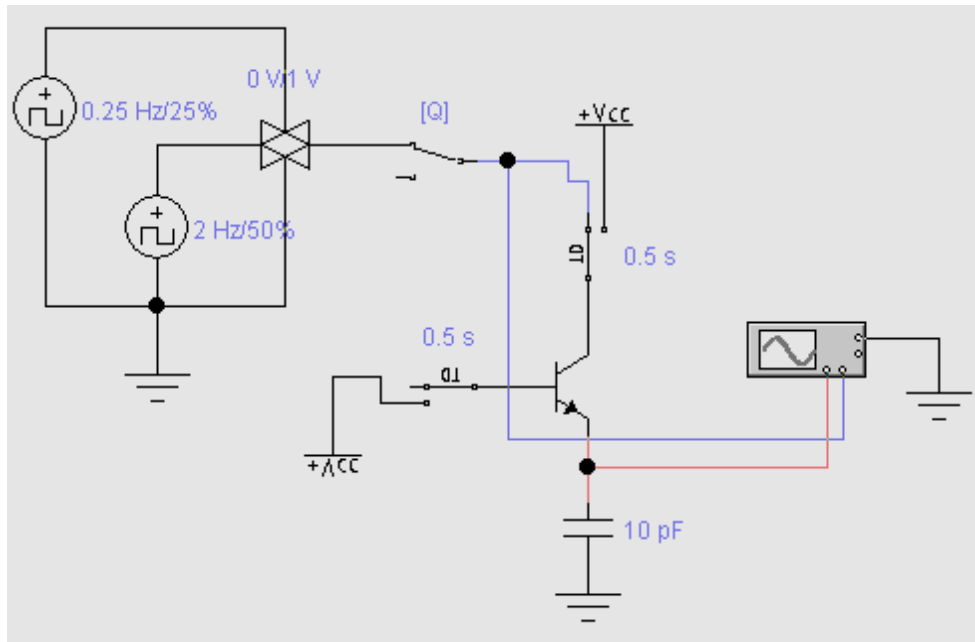


Рис. 11 – Схема исследования ячейки динамической оперативной памяти

- 2) Установить выходные напряжения ГТИ с частотой 0,25 Гц равное 5 В, а другого ГТИ равное 15 В.
- 3) Двойным щелчком запустить осциллограф и нажать в нем кнопку «Expand». Установить временную (горизонтальную) развертку осциллографа, равную 1 с/дел, вертикальную развертку по каналу А – 2 В/дел, по каналу В – 5 В/дел.
- 4) Отключить схему регенерации, нажав клавишу Q, и запустить схему в работу.
- 5) Убедившись в том, что в ячейку памяти записалась логическая единица, а затем наступил процесс разряда конденсатора, подключите схему регенерации данных.
- 6) Схему исследования и временные диаграммы, изображенные в осциллографе занести в отчет по практическому занятию.
- 7) Сделайте выводы по результатам проделанной работы.

3.4. Исследование 4-х битного ОЗУ

Задание 3.4.1. Исследовать внутреннюю структуру и логику функционирования 4-х битного статического ОЗУ с помощью генератора слова. Для этого выполните следующие операции:

- 1) Запустить программу Electronics Workbench и в его рабочем поле собрать схему исследования, приведенную на рис. 12.

Данное ОЗУ построено на базе ИМС 74155 дешифратора адреса из библиотеки EWB, типовых библиотечных D-триггеров и логических элементов. Ячейка статической оперативной памяти построена и работает также как в задании 3.3.1. Генератор слова в данной схеме имитирует общую шину, по которой поступают адреса ячеек, данные и сигналы управления запись/чтение. Логические пробники предназначены для мониторинга процедур записи и чтения.

- 2) Запрограммировать генератор слова, таким образом, чтобы вначале в каждую ячейку ОЗУ была последовательно произведена запись логической единицы, затем каждая ячейка была считана, и в заключение в каждую ячейку был последовательно записан логический ноль. При этом адреса всех четырех ячеек ОЗУ должны быть записаны в младшем полубайте слова, команды управления записью/чтением WR/RD' – в младшем разряде следующего полубайта, а данные для записи в ячейки – в третьем полубайте слова.

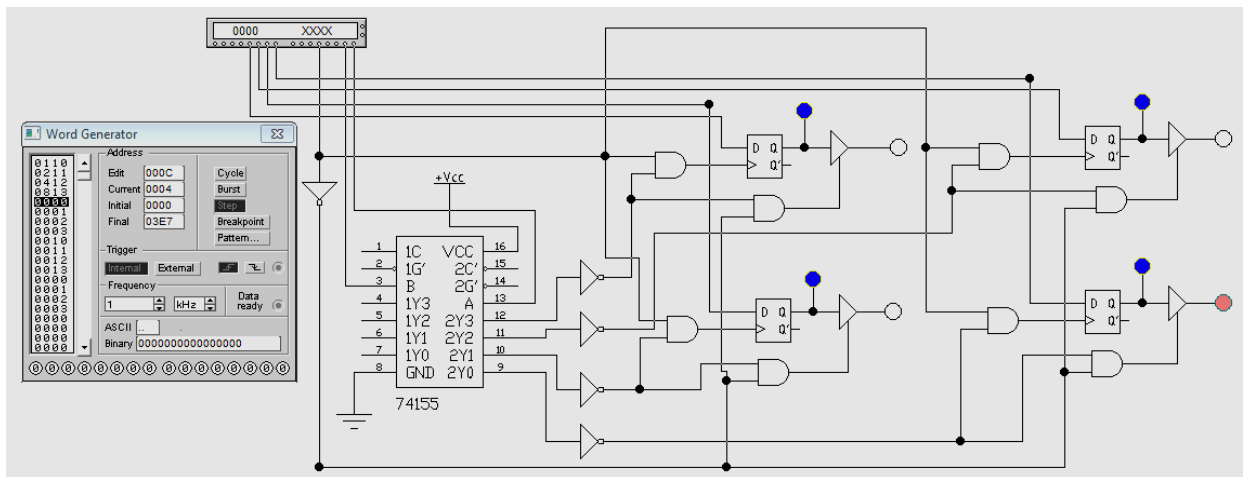


Рис. 12 – Схема исследования 4-х битного статического ОЗУ

- 3) Запустить генератор слова в пошаговом режиме и выполнить последовательно операции записи и чтения сначала логической единицы, затем логического нуля.
- 4) Определить адреса каждой ячейки и обозначить их на схеме занесенной в отчет.
- 5) Определить, для чего выходы дешифратора подключены к инверторам.
- 6) Сделайте выводы по результатам проделанной работы.

3.5. Исследование ячейки памяти ПЗУ

Задание 3.5.1. Исследовать внутреннюю структуру и логику функционирования ячейки постоянной памяти, программируемой путем пережигания перемычек. Для этого выполните следующие операции:

- 1) Запустить программу Electronics Workbench и в его рабочем поле собрать схему исследования, приведенную на рис. 13.

Ячейка памяти ПЗУ построена на транзисторе в цепи эмиттера, которого находится прожигаемая перемычка (в схеме моделирования – предохранитель, величиной 5 мА). При выборе данной ячейки памяти с дешифратора адреса на базу транзистора подается логическая единица, в результате транзистор открывается. Электрический ток небольшой величины, проходя через транзистор и перемычку с помощью делителя напряжения на резисторах, создает сигнал логической единицы на базе второго транзистора. Транзистор открывается, формируя на выходе сигнал логического нуля. Для записи в ячейку ПЗУ логической единицы прожигается перемычка путем подключения программатора в виде высоковольтного (15 в) источника напряжения. Это приводит к увеличению силы тока протекающей через первый транзистор и прожиганию перемычки. В результате оба транзистора всегда будут закрыты и, следовательно, в ячейке памяти будет находиться логическая единица.

- 5) Привести схему в исходное состояние, как на рис. 13 – выключатель Р должен подключать к схеме батарею 5 В, выключатель А замкнут на землю (общий провод).
- 6) Включить схему в работу и убедиться в том, что при замыкании и размыкании ключа А, имитирующего выбор данной ячейки памяти управляющим сигналом дешифратора адреса, транзисторы открываются и закрываются, включая и выключая логические пробники.
- 7) Установить, какой код может храниться в ячейке с непрожженной перемычкой.
- 8) Привести схему в исходное состояние и перевести выключатель Р в другое положения, имитируя подключение программатора для прожига перемычки.
- 9) Подключить данную ячейку к программатору, нажав выключатель А. Определить, к чему приводит данная процедура. Установить, какой код может храниться в ячейке с прожженной перемычкой.
- 10) Сделайте выводы по результатам проделанной работы.

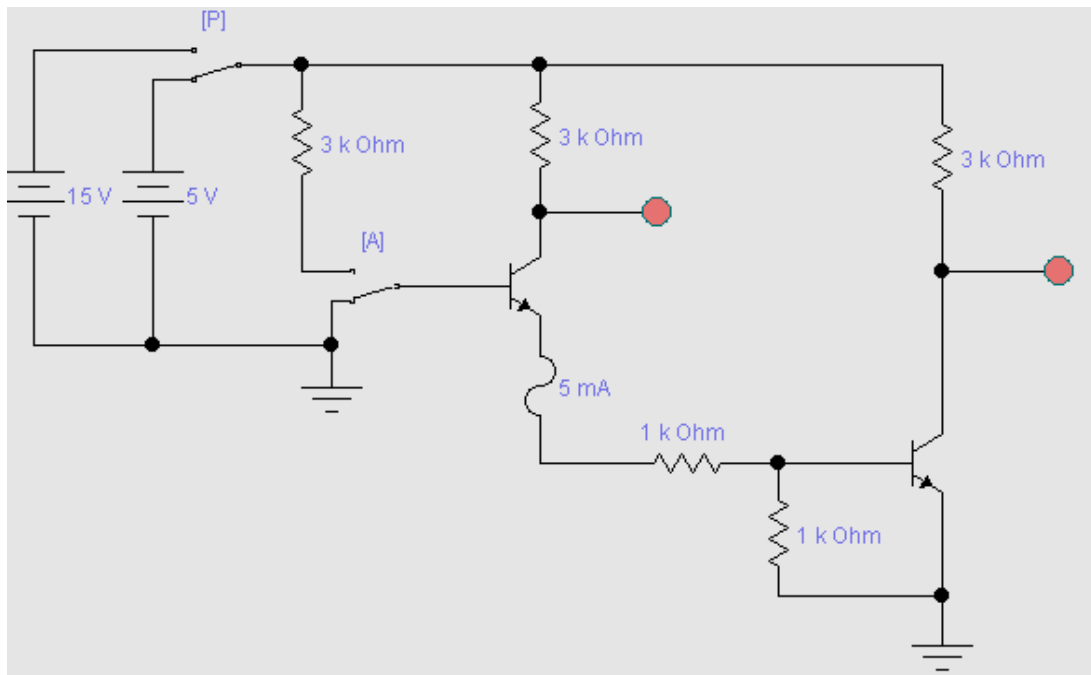


Рис. 13 – Схема ячейки памяти ПЗУ, программируемой путем пережигания перемычек

Контрольные вопросы:

- 1) Предназначение и классификация оперативных запоминающих устройств.
- 2) Характеристика внутренней структуры и принципов функционирования типового оперативного запоминающего устройства.
- 3) Характеристика логики функционирования оперативных запоминающих устройств статического типа.
- 4) Характеристика логики функционирования оперативных запоминающих устройств динамического типа.
- 5) Характеристика современных типов оперативной памяти.
- 6) Характеристика видеопамати.
- 7) Характеристика современных типов ИМС оперативной памяти.
- 8) Предназначение и классификация постоянных запоминающих устройств.
- 9) Характеристика и принципы функционирования ячейки постоянной памяти, однократно программируемой путем пережигания перемычек.
- 10) Практическое применение постоянных запоминающих устройств различных типов.
- 11) Порядок исследования внутренней структуры и логики функционирования ячейки статической оперативной памяти с помощью генератор слова.
- 12) Порядок исследования внутренней структуры и логики функционирования ячейки динамической оперативной памяти с помощью осциллографа.
- 13) Порядок исследования внутренней структуры и логики функционирования 4-х битного статического ОЗУ с помощью генератора слова.
- 14) Порядок исследования внутренней структуры и логики функционирования ячейки постоянной памяти, программируемой путем пережигания перемычек.

Лабораторная работа №7 Шинная структура связи компьютера (2 часа)

Содержание:

1.Шинная структура связей

Теоретические сведения:

Для достижения максимальной универсальности и упрощения протоколов обмена информацией в микропроцессорных системах применяется так называемая **шинная структура связей** между отдельными устройствами, входящими в систему. Суть шинной структуры связей сводится к следующему.

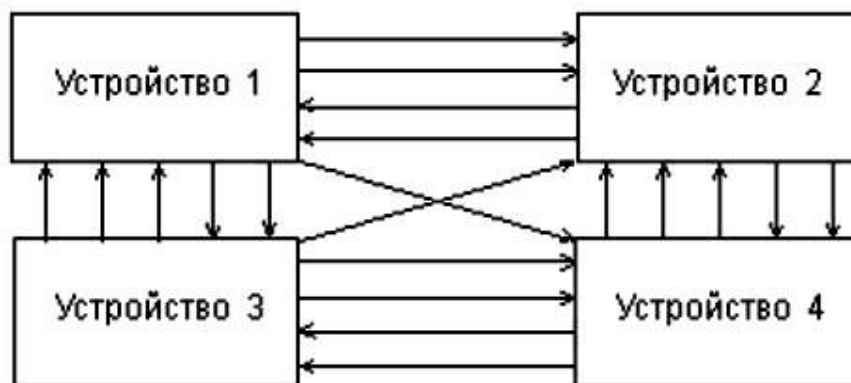


Рисунок 1 - Классическая структура связей

При классической структуре связей все сигналы и коды между устройствами передаются по отдельным линиям связи. Каждое устройство, входящее в систему, передает свои сигналы и коды независимо от других устройств. При этом в системе получается очень много линий связи и разных протоколов обмена информацией.

При шинной структуре связей все сигналы между устройствами передаются по одним и тем же линиям связи, но в разное время (это называется мультиплексированной передачей).

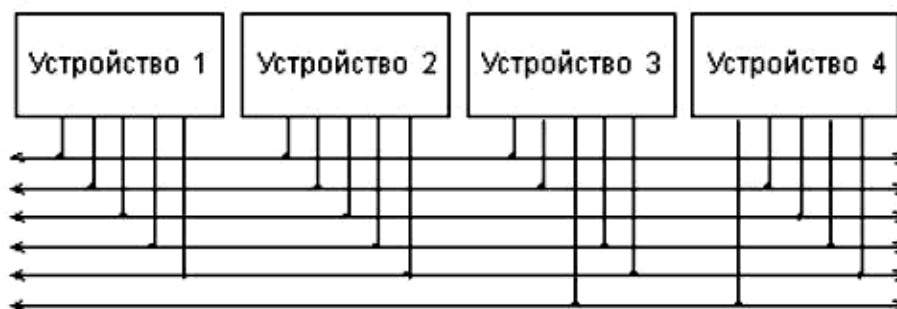


Рисунок 2 - Шинная структура связей

Причем передача по всем линиям связи может осуществляться в обоих направлениях (так называемая двунаправленная передача). В результате количество линий связи существенно сокращается, а правила обмена (протоколы) упрощаются. **Группа линий связи, по которым передаются сигналы или коды как раз и называется шиной** (англ. **bus**).

При шинной структуре связей вся информация передается по линиям связи *последовательно* во времени, что снижает быстродействие системы по сравнению с классической структурой связей.

Большое достоинство шинной структуры связей состоит в том, что все устройства,

подключенные к шине, должны принимать и передавать информацию по одним и тем же протоколам обмена информацией по шине. Соответственно, все узлы, отвечающие за обмен с шиной в этих устройствах, должны быть единообразны, унифицированы.

В системах с шинной структурой связей применяют три существующие разновидности выходных каскадов цифровых микросхем:

- стандартный выход (или выход с двумя состояниями - обозначается 2С, 2S, реже ТТЛ, TTL);
- выход с открытым коллектором - обозначается ОК, ОС;
- выход с тремя состояниями (с возможностью отключения - обозначается 3С, 3S).

Упрощенно эти три типа выходных каскадов могут быть представлены в виде следующих схем:

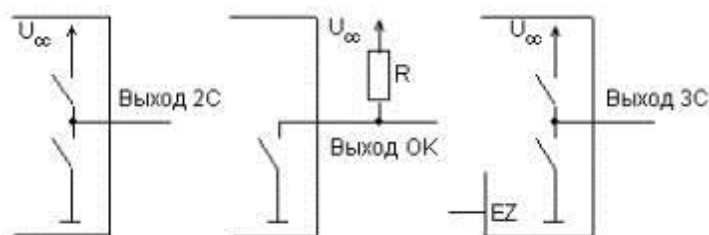
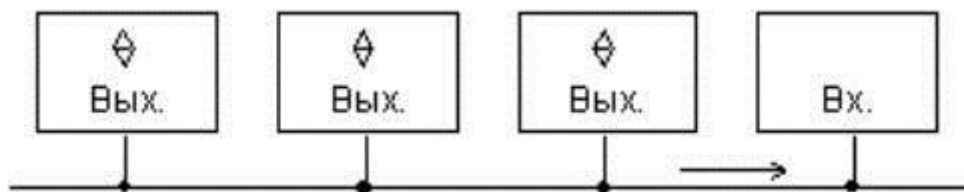


Рисунок 3 - Три типа выходов цифровых микросхем

У выхода 2С два ключа замыкаются по очереди, что соответствует уровням логической единицы (верхний ключ замкнут) и логического нуля (нижний ключ замкнут). У выхода ОК замкнутый ключ формирует уровень логического нуля, разомкнутый — логической единицы. У выхода 3С ключи могут замыкаться по очереди (как в случае 2С), а могут размыкаться одновременно, образуя третье, высокоимпедансное, состояние. Переход в третье состояние (Z-состояние) управляется сигналом на специальном входе EZ.

Выходные каскады типов 3С и ОК позволяют объединять несколько выходов микросхем для получения **мультиплексированных**



или двунаправленных линий.

Рисунок 4 - Мультиплексированная линия

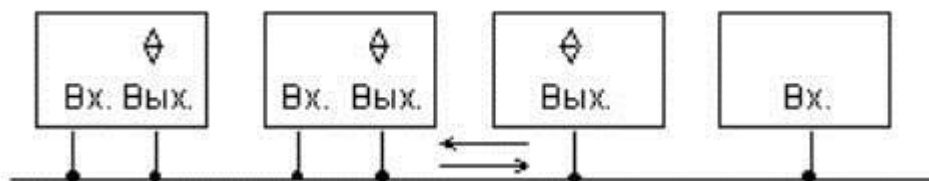


Рисунок 5 - Двунаправленная линия

При этом в случае выходов 3С необходимо обеспечить, чтобы на линии всегда работал только один активный выход, а все остальные выходы находились бы в это время в третьем состоянии, иначе возможны конфликты. Объединенные выходы ОК могут работать все одновременно, без всяких конфликтов.

Типичная структура микропроцессорной системы выглядит так:

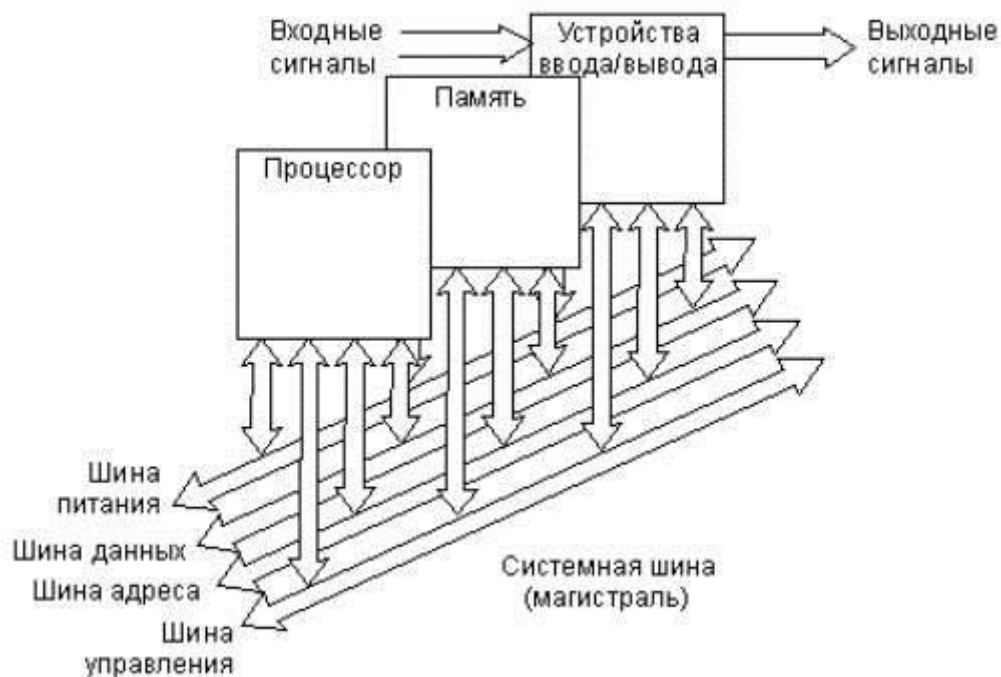


Рисунок 6 - Структура микропроцессорной системы. Она включает в себя три основных типа устройств:

- процессор;
- память, включающую оперативную память (ОЗУ, RAM — Random Access Memory) и постоянную память (ПЗУ, ROM — Read Only Memory), которая служит для хранения данных и программ;
- устройства ввода/вывода (УВВ, I/O — Input/Output Devices), служащие для связи микропроцессорной системы с внешними устройствами, для приема (ввода, чтения, Read) входных сигналов и выдачи (вывода, записи, Write) выходных сигналов.

Все устройства микропроцессорной системы объединяются *общей системной шиной* (она же называется еще *системной магистралью* или *каналом*). Системная магистраль включает в себя четыре основные шины нижнего уровня:

1. шина адреса (Address Bus);
2. шина данных (Data Bus);
3. шина управления (Control Bus);
4. шина питания (Power Bus).

Шина адреса служит для определения адреса (номера) устройства, с которым процессор обменивается информацией в данный момент. Каждому устройству (кроме процессора), каждой ячейке памяти в микропроцессорной системе присваивается собственный адрес. Когда код какого-то адреса выставляется процессором на шине адреса, устройство с таким адресом понимает, что ему предстоит обмен информацией. Шина адреса определяет максимально возможную сложность микропроцессорной системы, то есть допустимый объем памяти и, следовательно, максимально возможный размер программы и максимально возможный объем запоминаемых данных.

Количество адресов, обеспечиваемых шиной адреса, определяется как 2^N , где N — количество разрядов. Например, 16-разрядная шина адреса обеспечивает 65 536 адресов. Разрядность шины адреса обычно кратна 4 и может достигать 32 и даже 64. Шина адреса может быть однонаправленной (когда магистралью всегда управляет только процессор) или двунаправленной (когда процессор может временно передавать управление магистралью другому устройству, например контроллеру ПДП). Наиболее часто используются типы выходных каскадов с тремя состояниями или обычные ТТЛ (с двумя

состояниями).

Шина данных — это основная шина, которая используется для передачи информационных кодов между всеми устройствами микропроцессорной системы. Обычно в пересылке информации участвует процессор, который передает код данных в какое-то устройство или в ячейку памяти или же принимает код данных из какого-то устройства или из ячейки памяти. Но возможна также и передача информации между устройствами без участия процессора. Шина данных всегда *двунаправленная*. Наиболее часто встречающийся тип выходного каскада для линий этой шины — выход с тремя состояниями.

Количество ее разрядов (линий связи) определяет скорость и эффективность информационного обмена, а также максимально возможное количество команд. Обычно шина данных имеет 8, 16, 32 или 64 разряда. Понятно, что за один цикл обмена по 64-разрядной шине может передаваться 8 байт информации, а по 8-разрядной — только один байт. Разрядность шины данных определяет и разрядность всей магистрали. Например, когда говорят о 32-разрядной системной магистрали, подразумевается, что она имеет 32-разрядную шину данных.

Шина управления в отличие от шины адреса и шины данных состоит из отдельных управляющих сигналов. Каждый из этих сигналов во время обмена информацией имеет свою функцию. Некоторые сигналы служат для стробирования передаваемых или принимаемых данных (то есть определяют моменты времени, когда информационный код выставлен на шину данных).

Другие управляющие сигналы могут использоваться для подтверждения приема данных, для сброса всех устройств в исходное состояние, для тактирования всех устройств и т.д. Кроме того, управляющие сигналы обеспечивают согласование работы процессора (или другого хозяина магистрали, задатчика, master) с работой памяти или устройства ввода/вывода (устройства-исполнителя, slave). Управляющие сигналы также обслуживают запрос и предоставление прерываний, запрос и предоставление прямого доступа.

Линии шины управления могут быть как однонаправленными, так и двунаправленными. Типы выходных каскадов могут быть самыми разными: с двумя состояниями (для однонаправленных линий), с тремя состояниями (для двунаправленных линий), с открытым коллектором (для двунаправленных и мультиплексированных линий).

Наконец, **шина питания** предназначена не для пересылки информационных сигналов, а для питания системы. Она состоит из линий питания и общего провода. В микропроцессорной системе может быть один источник питания (чаще +5 В) или несколько источников питания (обычно еще –5 В, +12 В и –12 В). Каждому напряжению питания соответствует своя линия связи. Все устройства подключены к этим линиям параллельно.

Если в микропроцессорную систему надо ввести входной код (или входной сигнал), то процессор по шине адреса обращается к нужному устройству ввода/вывода и принимает по шине данных входную информацию. Если из микропроцессорной системы надо вывести выходной код (или выходной сигнал), то процессор обращается по шине адреса к нужному устройству ввода/вывода и передает ему по шине данных выходную информацию.

Если информация должна пройти сложную многоступенчатую обработку, то процессор может хранить промежуточные результаты в системной оперативной памяти. Для обращения к любой ячейке памяти процессор выставляет ее адрес на шину адреса и передает в нее информационный код по шине данных или же принимает из нее информационный код по шине данных. В памяти (оперативной и постоянной) находятся также и управляющие коды (команды выполняемой процессором программы), которые процессор также читает по шине данных с адресацией по шине адреса. Постоянная память используется в основном для хранения программы начального пуска микропроцессорной системы, которая выполняется каждый раз после включения питания. Информация в нее заносится изготовителем раз и навсегда.

Важно учитывать, что устройства ввода/вывода чаще всего представляют собой

устройства на «жесткой логике». На них может быть возложена часть функций, выполняемых микропроцессорной системой. Поэтому у разработчика всегда имеется возможность перераспределять функции системы между аппаратной и программной реализациями оптимальным образом. Чаще всего применяется комбинирование аппаратных и программных функций.

Иногда устройства ввода/вывода имеют в своем составе процессор, то есть представляют собой небольшую специализированную микропроцессорную систему. Это позволяет переложить часть программных функций на устройства ввода/вывода, разгрузив центральный процессор системы.

2. Типы обмена информацией по шинам

Как уже упоминалось, в системную магистраль (системную шину) микропроцессорной системы входит три основные информационные шины: адреса, данных и управления.

Как в шине данных, так и в шине адреса может использоваться *положительная логика* или *отрицательная логика*. При положительной логике высокий уровень напряжения соответствует логической единице на соответствующей линии связи, низкий — логическому нулю. При отрицательной логике — наоборот. В большинстве случаев уровни сигналов на шинах — ТТЛ.

Сигналы шины управления также могут передаваться как в положительной логике (реже), так и в отрицательной логике (чаще).

Для снижения общего количества линий связи магистрали часто применяется **мультиплексирование** шин адреса и данных. То есть одни и те же линии связи используются в разные моменты времени для передачи как адреса, так и данных (в начале цикла — адрес, в конце цикла — данные). Для фиксации этих моментов (стробирования) служат специальные сигналы на шине управления.

Понятно, что мультиплексированная шина адреса/данных обеспечивает меньшую скорость обмена, требует более длительного цикла обмена. По типу шины адреса и шины данных все магистрали также делятся на мультиплексированные и немультимплексированные.



Рисунок 7 – Типы магистралей

В некоторых мультиплексированных магистралях после одного кода адреса передается несколько кодов данных (массив данных). Это позволяет существенно повысить быстродействие магистрали. Иногда в магистралях применяется частичное мультиплексирование, то есть часть разрядов данных передается по немультимплексированным линиям, а другая часть — по мультиплексированным с адресом линиям.

Самые главные управляющие сигналы — это **стробы обмена**, то есть сигналы, формируемые процессором и определяющие моменты времени, в которые производится пересылка данных по шине данных, обмен данными. Чаще всего в магистрали используются два различных строба обмена:

- **Строб записи** (вывода), который определяет момент времени, когда устройство-исполнитель может принимать данные, выставленные процессором на шину данных;
- **Строб чтения** (ввода), который определяет момент времени, когда устройство-исполнитель должно выдать на шину данных код данных, который будет прочитан

процессором.

При этом большое значение имеет то, как процессор заканчивает обмен в пределах цикла, в какой момент он снимает свой строб обмена. Возможны два пути решения (рис. 8):

1. При *синхронном* обмене процессор заканчивает обмен данными самостоятельно, через раз и навсегда установленный временной интервал выдержки ($t_{\text{выд}}$), то есть без учета интересов устройства-исполнителя;

2. При *асинхронном* обмене процессор заканчивает обмен только тогда, когда устройство-исполнитель подтверждает выполнение операции специальным сигналом (так называемый режим handshake — рукопожатие).

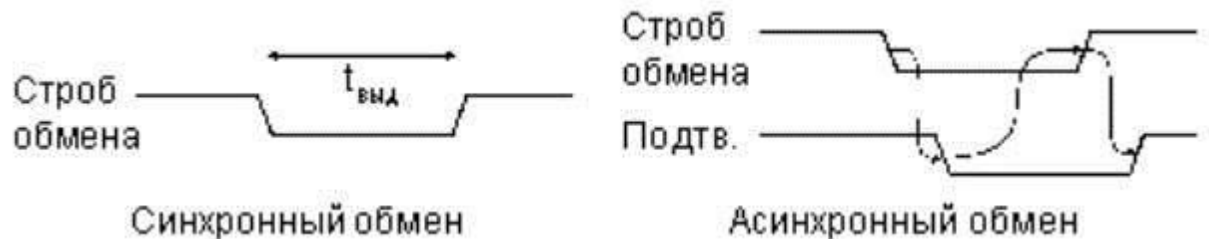


Рисунок 8 – Способы организации информационного обмена

Достоинства синхронного обмена — более простой протокол обмена, меньшее количество управляющих сигналов. Недостатки — отсутствие гарантии, что исполнитель выполнил требуемую операцию, а также высокие требования к быстродействию исполнителя.

Достоинства асинхронного обмена — более надежная пересылка данных, возможность работы с самыми разными по быстродействию исполнителями. Недостаток — необходимость формирования сигнала подтверждения всеми исполнителями, то есть дополнительные аппаратные затраты.

3. Шины компьютера. Основные интерфейсы компьютера



Рисунок 8 – Современные интерфейсы ПК

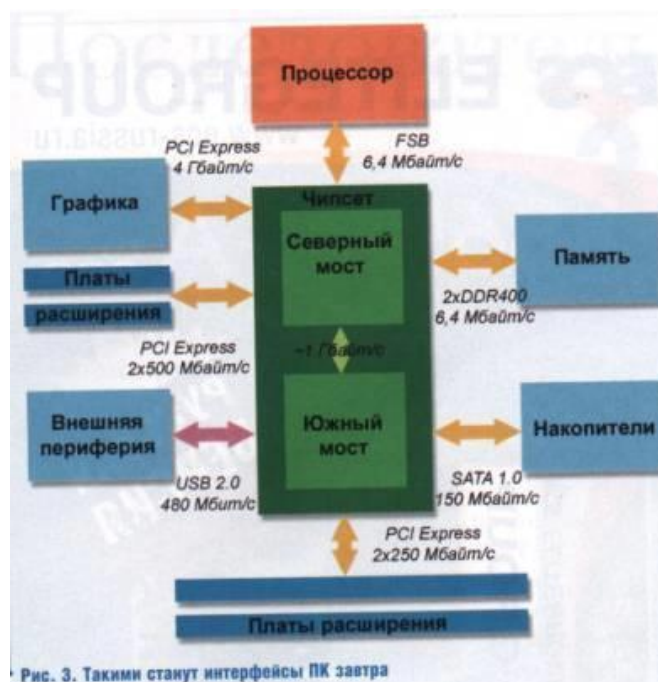


Рисунок 9 – Интерфейсы ПК в будущем

Задание:

- 1) Составить принципиальную схему замещения стандартного выходного каскада цифровой микросхемы, пояснить принцип работы такого выходного каскада.
- 2) Составить принципиальную схему замещения выходного каскада с открытым коллектором для цифровой микросхемы, пояснить принцип работы такого выходного каскада.
- 3) Составить принципиальную схему замещения выходного каскада с тремя состояниями для цифровой микросхемы, пояснить принцип работы такого выходного каскада.
- 4) Составить структурные схемы мультиплексированной и двунаправленной линий шины, пояснить их принципы работы.
- 5) Составить структурную схему физического уровня шины PCI Express.
- 6) Разработать алгоритм преобразования данных на физическом уровне шины PCI Express при передаче и приеме.
- 7) Изобразить и пояснить временные диаграммы цикла передачи данных в интерфейсе Centronics.
- 8) Изобразить и пояснить формат передаваемых данных интерфейса RS-232C, уровни сигналов RS-232C на передающем и принимающем концах линии связи.
- 9) Составить принципиальную схему мультиплексированной линии, к которой подключены три выходных каскада цифровых микросхем с открытыми коллекторами, пояснить принцип работы такой мультиплексированной линии.
- 10) Составить принципиальную схему мультиплексированной линии, к которой подключены три выходных каскада цифровых микросхем с тремя состояниями, пояснить принцип работы такой мультиплексированной линии.
- 11) Изобразить и пояснить временные диаграммы мультиплексированной шины адреса/данные при синхронном и асинхронном обмене данными.
- 12) Составить таблицу со сравнительными характеристиками шин ISA, PCI, PCI Express.
- 13) Составить таблицу со сравнительными характеристиками для интерфейсов SCSI, IDE, EIDE.

Лабораторная работа №8 Организация памяти ЭВМ (2 часа)

Содержание:

1. Организация памяти ЭВМ.

Теоретические сведения:

Во время появления первых действующих компьютеров сложилось деление памяти на *внутреннюю* и *внешнюю*. Под внутренней подразумевалась память, расположенная внутри процессорного "шкафа" (или примыкающая к нему). Сюда входила и электронная и магнитная память (на магнитных сердечниках). Для внутренней памяти характерен одномерный (линейный) адрес, который представляет собой одно двоичное число. Доступ к памяти осуществляется по адресу, заданному программой и для процессора она является непосредственно доступной.

Сегодня внутренняя (и оперативная, и постоянная) электронная (полупроводниковая) память является хранилищем программного кода, который непосредственно может быть исполнен процессором и устанавливается на системной плате или на платах расширения.

Обычно под компьютерной памятью понимают только "внутреннюю память" компьютера, которая в свою очередь **классифицируется** на:

ОЗУ, оперативное запоминающее устройство с оперативной памятью (RAM, Random Access Memory) - памятью с произвольным доступом (английский термин), определяет допустимый объем и скорость одновременно выполняемых процедур. Произвольность доступа подразумевает возможность операций записи или чтения с любой ячейкой ОЗУ в произвольном порядке.

Самые первые ЭВМ имели память на ртутных линиях задержки или электронно-лучевых трубках. Эти экзотические устройства были быстро вытеснены запоминающими устройствами на магнитных сердечниках, которые получили широкое распространение и применялись довольно долго. Намагниченное состояние небольшого ферритового кольца, которое служило элементом для запоминания одного бита информации, соответствовало единице, ненамагниченное – нулю.

Наконец, развитие технологий интегральной микроэлектроники позволило изготовить компактную полупроводниковую память, которая в данный момент и применяется в компьютерах. До последних лет из микросхем памяти использовали два основных типа: статическую (SRAM - Static RAM) и динамическую (DRAM - Dynamic RAM).

При выключении компьютера вся информация, которая находилась в оперативной памяти компьютера, безвозвратно теряется.

ПЗУ (постоянное запоминающее устройство, ROM, Read Only Memory, память только для чтения) - постоянная память, информацию из которой можно только считывать.

Позднее появились конструкции ПЗУ, которые можно было запрограммировать несколько раз. Они получили название перепрограммируемые ПЗУ (ППЗУ). Стирание старой информации, т. е. установка всех битов памяти в единичное состояние, производилось ультрафиолетовым излучением (интенсивности бытового прибора для загара вполне хватало!), для которого в корпусе микросхемы делалось специальное прозрачное окошечко.

Наконец, самые последние разработки ППЗУ позволяют производить обновление информации чисто электрическим путем, причем уже без специального программатора и не вынимая из платы.

ПЗУ - это одна или несколько микросхем, расположенных на системной плате и содержащих набор подпрограмм. ПЗУ играет в современных компьютерах очень важную роль. Прежде всего, каждый компьютер содержит ПЗУ с программой начальной загрузки, о котором мы уже говорили.

В этой же самой микросхеме обычно хранятся минимальные программы работы с

клавиатурой и другими устройствами, поэтому ее часто называют *BIOS*— Basic Input/Output System (данные программы можно сравнить с врожденными безусловными рефлексам у живого существа). Кроме того, в ПЗУ может помещаться информация, необходимая для работы внешних устройств, например о начертании символов: так называемые знакогенераторы, которые были в каждом матричном принтере. Наконец, многие современные платы управления внешними устройствами, например, жестким диском, фактически являются настоящими специализированными ЭВМ и содержат собственное ПЗУ со встроенным программным обеспечением.

Cache-память (Кэш-память, Cache Memory) - сверхоперативная память (СОЗУ), является буфером между ОЗУ и процессором (несколькими процессорами) и другими абонентами системной шины.

Кэш представляет собой "быструю" статическую память небольшого объема, которая служит для ускорения доступа к полному объему "медленной" динамической памяти. Основная идея работы кэш-памяти заключается в том, что извлеченные из ОЗУ данные или команды программы копируются в кэш; одновременно в специальный каталог адресов, который находится в той же самой памяти, запоминается, откуда информация была извлечена. Если данные потребуются повторно, то уже не надо будет терять время на обращение к ОЗУ — их можно получить из кэш-памяти значительно быстрее.

Поскольку объем кэша существенно меньше объема оперативной памяти, его контроллер (управляющая схема) тщательно следит за тем, какие данные следует сохранять в кэш, а какие заменять: удаляется та информация, которая используется реже или совсем не используется. Контроллер также обеспечивает своевременную запись измененных данных из кэша обратно в основное ОЗУ.

Кэш в современных компьютерах строится по двух-уровневой схеме:

- **L1 Cache** (Level 1 Cashe, Первичный кэш) - кэш 1 уровня, внутренний (Internal, Integrated) кэш процессоров класса 486 и старше.
- **L2 Cache** (Level 2 Cashe, Вторичный кэш) - кэш 2 уровня. Работает на частоте шины. Для кэша 2-го уровня всегда используется SRAM. Характерные емкости - от 256kB до 1MB на процессор. Объем и быстродействие *L2 Cashe* оказывают огромное влияние на быстродействие ПК в целом.

Но иногда установка в ПК систему дополнительной памяти может и заметно замедлить ее работу, если контроллер не поддерживает кэширование этой памяти. Обычно это **внешний** (*External*) кэш, установленный на системной плате. Для Pentium Pro и Pentium II вторичный кэш расположен в одном корпусе с самим процессором.

Flash-память (флэш-память). Предназначенного для хранения информации и/или переноса между мобильными устройствами и ПК. Для исполнения программы на ПК в моменты программирования и стирания флэш-памяти требуется ОЗУ.

Подобно ОЗУ, флэш-память модифицируется электрически внутрисистемно, но подобно ПЗУ, flash-память энергонезависима и хранит данные даже после отключения питания.

В отличие от ОЗУ, flash-память нельзя переписывать побайтно. Она читается и записывается байт за байтом и ее нужно полностью стереть перед тем, как записывать новые данные.

Наибольшее распространение получили карты памяти и тип Boot Block. Flash-память типа Boot Block служит для хранения обновляемых программ и данных в сотовых телефонах (SIM-карта), модемах, BIOS-ах, системы управления автомобильными двигателями и т.п.

Организация памяти.Адресация операндов

Большая часть команд процессора работает с кодами данных (операндами). Одни команды требуют входных операндов (одного или двух), другие выдают выходные операнды (чаще один операнд). Входные операнды называются еще операндами- источниками, а выходные называются операндами-приемниками. Все эти коды операндов (входные и

выходные) должны где-то располагаться. Они могут находиться во внутренних регистрах процессора (наиболее удобный и быстрый вариант). Они могут располагаться в системной памяти (самый распространенный вариант). Наконец, они могут находиться в устройствах ввода/вывода (наиболее редкий случай). Определение места положения операндов производится кодом команды. Причем существуют разные методы, с помощью которых код команды может определить, откуда брать входной операнд и куда помещать выходной операнд. Эти методы называются методами адресации. Эффективность выбранных методов адресации во многом определяет эффективность работы всего процессора в целом.

Методы адресации

Количество методов адресации в различных процессорах может быть от 4 до 16. Рассмотрим несколько типичных методов адресации операндов, используемых сейчас в большинстве микропроцессоров.

Непосредственная адресация предполагает, что операнд (входной) находится в памяти непосредственно за кодом команды.



Рисунок 1 - Непосредственная адресация

Операнд обычно представляет собой константу, которую надо куда-то переслать, к чему-то прибавить и т.д. Например, команда может состоять в том, чтобы прибавить число 6 к содержимому какого-то внутреннего регистра процессора. Это число 6 будет располагаться в памяти, внутри программы в адресе, следующем за кодом данной команды сложения.

Прямая (она же абсолютная) **адресация** предполагает, что операнд (входной или выходной) находится в памяти по адресу, код которого находится внутри программы сразу же за кодом команды. Например, команда может состоять в том, чтобы очистить (сделать нулевым) содержимое ячейки памяти с адресом 1000000. Код этого адреса 1000000 будет располагаться в памяти, внутри программы в следующем адресе за кодом данной команды очистки.



Рисунок 2 - Прямая адресация

Регистровая адресация предполагает, что операнд (входной или выходной) находится во внутреннем регистре процессора.



Рисунок 3 - Регистровая адресация

Например, команда может состоять в том, чтобы переслать число из нулевого регистра в первый. Номера обоих регистров (0 и 1) будут определяться кодом команды пересылки.

Косвенно-регистровая (она же **косвенная**) **адресация** предполагает, что во внутреннем регистре процессора находится не сам операнд, а его адрес в памяти.

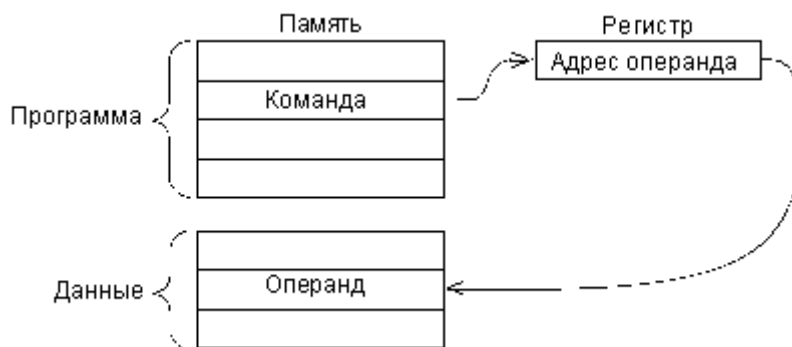


Рисунок 4 - Косвенная адресация

Например, команда может состоять в том, чтобы очистить ячейку памяти с адресом, находящимся в нулевом регистре. Номер этого регистра (0) будет определяться кодом команды очистки.

Реже встречаются еще два метода адресации.

Автоинкрементная адресация очень близка к косвенной адресации, но отличается от нее тем, что *после* выполнения команды содержимое используемого регистра увеличивается на единицу или на два. Этот метод адресации очень удобен, например, при последовательной обработке кодов из массива данных, находящегося в памяти. После обработки какого-то кода адрес в регистре будет указывать уже на следующий код из массива. При использовании косвенной адресации в данном случае пришлось бы увеличивать содержимое этого регистра отдельной командой.

Автодекрементная адресация работает похоже на автоинкрементную, но только содержимое выбранного регистра уменьшается на единицу или на два *перед* выполнением команды. Эта адресация также удобна при обработке массивов данных. Совместное использование автоинкрементной и автодекрементной адресаций позволяет организовать память стекового типа.

Из других распространенных методов адресации можно упомянуть об **индексных методах**, которые предполагают для вычисления адреса операнда прибавление к содержимому регистра заданной константы (индекса). Код этой константы располагается в памяти непосредственно за кодом команды.

Выбор того или иного метода адресации в значительной степени определяет время выполнения команды. Самая быстрая адресация — это регистровая, так как она не требует дополнительных циклов обмена по магистрали. Если же адресация требует обращения к памяти, то время выполнения команды будет увеличиваться за счет длительности необходимых циклов обращения к памяти. Чем больше внутренних регистров у процессора, тем чаще и свободнее можно применять регистровую адресацию, и тем быстрее будет работать система в целом.

Говоря об адресации, нельзя обойти вопрос о сегментировании памяти, применяемой в некоторых процессорах, например в процессорах IBM PC-совместимых персональных компьютеров. В процессоре Intel 8086 сегментирование памяти организовано следующим образом.

Вся память системы представляется не в виде непрерывного пространства, а в виде нескольких кусков — сегментов заданного размера (по 64 Кбайта), положение которых в пространстве памяти можно изменять программным путем.

Для хранения кодов адресов памяти используются не отдельные регистры, а пары регистров:

- сегментный регистр определяет адрес начала сегмента (то есть положение сегмента в памяти);
- регистр указателя (регистр смещения) определяет положение рабочего адреса внутри сегмента.

При этом физический 20-разрядный адрес памяти, выставляемый на внешнюю шину адреса, образуется так, как показано на рис. 20, то есть путем сложения смещения и адреса сегмента со сдвигом на 4 бита. Положение этого адреса в памяти показано на рисунке.



Рисунок 5 - Формирование физического адреса памяти из адреса сегмента и смещения

Например, сегмент в шестнадцатеричном виде равен 1200, а смещение — 3ABE.

Сегмент (1200)	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Смещение (3ABE)					0	0	1	1	1	1	0	0	1	1	0	1	1
Адрес (15ABE)	0	0	0	1	0	1	0	1	1	1	0	0	1	1	0	1	1



Рисунок 6 - Физический адрес в сегменте (все коды — шестнадцатеричные).

Сегмент может начинаться только на 16-байтной границе памяти (так как адрес начала сегмента, по сути, имеет четыре младших нулевых разряда, как видно из рис. 19), то есть с адреса, кратного 16. Эти допустимые границы сегментов называются границами параграфов.

Отметим, что введение сегментирования, прежде всего, связано с тем, что внутренние регистры процессора 16-разрядные, а физический адрес памяти 20-разрядный (16-разрядный адрес позволяет использовать память только в 64 Кбайт, что явно недостаточно). В появившемся в то же время процессоре MC68000 фирмы Motorola внутренние регистры 32-разрядные, поэтому там проблемы сегментирования памяти не возникает.

Применяются и более сложные методы сегментирования памяти. Например, в процессоре Intel 80286 в так называемом **защищенном режиме** адрес памяти вычисляется в соответствии с рисунком.

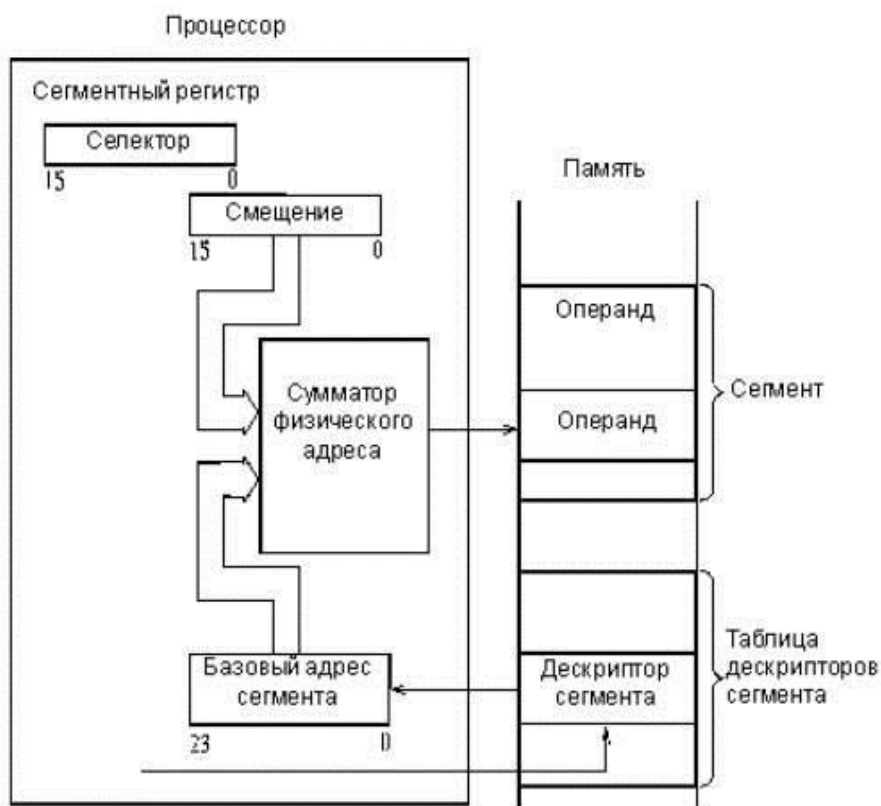


Рисунок 7 - Адресация памяти в защищенном режиме процессора Intel 80286

В сегментном регистре в данном случае хранится не **базовый** (начальный) адрес сегментов, а коды селекторов, определяющие адреса в памяти, по которым хранятся дескрипторы (то есть описатели) сегментов. Область памяти с дескрипторами называется таблицей дескрипторов. Каждый дескриптор сегмента содержит базовый адрес сегмента, размер сегмента (от 1 до 64 Кбайт) и его атрибуты. Базовый адрес сегмента имеет разрядность 24 бит, что обеспечивает адресацию 16 Мбайт физической памяти.

Еще более сложный метод адресации памяти с сегментированием использован в процессорах Intel 80386 и далее.

Задание:

- 1) Составить иерархическую структуру памяти ЭВМ.
- 2) Рассчитать физический адрес оперативной памяти ЭВМ по заданным адресам сегмента и смещения.
- 3) Составить структурную схему адресации оперативной памяти ЭВМ в защищенном режиме процессора Intel 80286.
- 4) Составить структурную схему адресации оперативной памяти ЭВМ в защищенном режиме процессора Intel 80386.
- 5) Составить спецификацию дополнительной памяти ЭВМ по стандарту XMS 3.0.
- 6) Определить тип корпуса и форм-факторы указанной на учебном стенде микросхемы памяти ЭВМ.
- 7) Произвести расшифровку модуля памяти типа DDR I по указанной маркировке.
- 8) Произвести расшифровку модуля памяти типа DDR II по указанной маркировке.
- 9) Произвести расшифровку модуля памяти типа DDR III по указанной маркировке.
- 10) Изобразить принципы записи и считывания в ячейку памяти на ферроэлектрическом конденсаторе.
- 11) Составить принципиальную схему одностранзисторной ячейки памяти 1T FeRAM, пояснить ее принцип работы.
- 12) Составить принципиальную схему одноконденсаторной ячейки памяти 1C FeRAM, пояснить ее принцип работы.
- 13) Составить принципиальную схему транзисторно-конденсаторной ячейки памяти 1T-1C FeRAM, пояснить ее принцип работы.
- 14) Составить принципиальную схему двойной транзисторно-конденсаторной ячейки памяти 2T-2C FeRAM, пояснить ее принцип работы.
- 15) Выполнить работу по извлечению и (или) установке всех видов памяти используемых на стенде и проверка сохранения работоспособности после совершенных манипуляций.

Лабораторная работа №9 Системная плата компьютера (2 часа)

Содержание:

1. Системная (материнская) плата компьютера.

Теоретические сведения:

Системной (System Board) или **материнской платой** (Mother Board, m/b, MB, "мама") называют основную печатную плату ПК, на которой устанавливаются другие системные компоненты. Она представляет собой плоский лист фольгированного стеклотекстолита (в последние годы - пластика), на котором размещены основные компоненты.

Компоненты подразделяются на:

Внутренние. Соединение этих элементов выполняется предварительным травлением

медной фольги нанесенной на подложку стеклотекстолита. В технологическом цикле несколько листов соединяются в многослойную структуру, покрытую защищенным лаком.

Внешние. В системную плату устанавливается процессор и память, а также платы таких устройств, как внутренний модем, сетевая плата, видео-плата и т.п.

У некоторых компьютеров на одной системной плате могут быть все элементы, необходимые для его работы. Это контроллеры дисковых накопителей, видеоадаптер, контроллеры портов и др. Называются такие платы All-In-One или "все на борту". Но чаще всего к стандарту добавляют только два компонента - звук и видео. У большинства же плат - только основные узлы.

Отсутствующие элементы располагают на отдельных платах, которые вставляются в специальные разъемы расширения на системной плате. Разъемы расширения этих плат связаны друг с другом на м/б рядом параллельных проводников, по которым осуществляется передача данных, адресов и управляющих сигналов. Разъемы бывают разборные и неразборные.

На платах All-In-One иногда все же имеется только один разъем расширения для так называемой riser-карты. На ней размещаются несколько разъемов расширения и вставляемые в них платы располагаются параллельно материнской.

Системная плата содержит множество **внутренних компонентов**, определяющих архитектуру компьютера. Это:

- Разъем под центральный процессор (1-2 шт., Socket-Slot);
- Cache (кэш, L1-L2, напр. объемом 256 Кбайт) и разъем под него;
- Контроллер винчестера (как минимум требуется поддержка и "UltraDMA" и 7200 об/мин) и разъемы шлейфов к жестким дискам;
- Слоты (разъемы) под оперативную память (два, три, четыре);
- Контроллер (коннектор) клавиатуры и мыши;
- Микросхема, обслуживающая floppy-дисковод, последовательные порты и параллельный порт;
- Перезаписываемая Bios (старое название Flash-память);
- Системные преобразователи напряжения;
- Слот (разъемы) AGP-шины (для видео-карты, карты Internet и др.);
- Гнезда расширения (слоты, Slot);
- Контроллеры внешних устройств;
- Разъемы прочих шин (интерфейсные схемы);
- CNR-слот (либо для сетевой карты, либо для модема);
- AMR-слот (либо для звуковой карты [Audio Riser, либо для модема [ModemRiser,);
- Прочие компоненты;

В рамках одного семейства производители выпускают большое число плат- близнецов, отличающихся только наличием или отсутствием дополнительных компонентов (SCSI-адаптер, звуковая или сетевая карта и т.п.). И производительность, и другие характеристики таких "близняшек" одинаковы.

Слот (Slot) представляет собой щелевой разъем, в который устанавливается либо процессор (напр. сокет-Socket) или какая-либо печатная плата (это слот расширения). Другое определение слота расширения (Expansion Slot) - разъем системной шины в совокупности с прорезью в задней стенке компьютера. Обычно это ISA Slot, PCI Slot, AGP Slot. Т.е. это посадочное место для установки карты расширения. Платой расширения, или картой расширения (Expansion Card), называют печатную плату с краевым разъемом, устанавливаемую в слот расширения. Карты расширения, привносящие в PC какой-либо дополнительный интерфейс, называются интерфейсными картами (Interface Card). Поскольку интерфейсная плата представляет собой "приспособление" для подключения какого-либо устройства, к ней применимо и название адаптер (Adapter). Например, дисплейный адаптер (Display adapter) служит для подключения дисплея-монитора. Адаптер и интерфейсная карта

практически синонимы, и например, NIC (Network Interface Card) часто переводится как адаптер ЛВС (локальной вычислительной сети). Каждая материнская плата, формально декларируя работу на одной частоте:

33 МГц, 66 МГц, 100 МГц, 133 МГц, 200 МГц, 266 МГц, 333 МГц,
533 МГц,

имеет несколько частот работы: частота шины между мостами чипсета (напр. 266 MBps), частота FSB (напр. 400 MHz), частота памяти (напр. 100/133/200/266). При этом выбор системной частоты подразумевает как выбор PCI-частоты напрямую 66/33 или через делитель частоты 66(1/2):

66/33, 68/34, 75/37, 83/41, 100/33, 103/34, 105/35, 110/36, 112/37, 115/38, 117/29,
117/30, 117/39, 120/40, 124/31, 124/41, 129/32, 133/33, 133/44, 138/34, 140/35, 143/36,
148/37,
150/37, 153/38,

так и выбор-соотношение отношение частоты шины AGP к системной частоте:
/1, 2/3

Важно, что при частоте шине PCI, значительно отличающейся от стандартного значения 33 МГц, возможна нестабильная работа устройств, установленных в ее слотах, а также некоторых жестких дисков.

Джампер (Jumper) представляет собой маленькую съемную перемычку, устанавливаемую на торчащие из m/b контакты. Джамперы используются для конфигурирования различных компонентов, для которых не требуется оперативного управления (как выключатели или переключатели). Установку на m/b требуемую частоту процессора делают jumper-ом. Напр. процессор PIII 1000Mhz - это частота шины 133Mhz (1-й jumper) и коэффициент умножения 7,5 (2-й jumper). Коэффициент умножения имеет значения от 2,0 до 10,0, обычный диапазон меньше - напр. 2,0-8,0 с шагом 0,5.

Разгон процессора до недавних пор делался именно jumper-ами. Например, чтобы заставить PIII 666Mhz работать как PIII 750Mhz, достаточно было переключить или jumper коэффициента с 5 на 5,5, или jumper частоты с 133 до 150. Правда выигрыш будет только в том случае, если устройства (минимум m/b и процессор) могут работать на итоговой новой повышенной частоте. Работать стабильно! Однако в последние годы фирмы-производители стали блокировать jumper частоты шины и ограничивать кол-во вариантов jumper-а коэффициентов. Т.о. возможности разгона резко сократились. Счастливое исключение - платы ABIT, где вместе с "Руководством" по самой плате дается еще и Overclocking Gid.

Есть тенденция постепенного перехода к безджамперным технологиям. В этом случае появляется возможность производить установки для процессора непосредственно из Bios-a.

DIP-переключатели (DIP Switches) представляют собой малогабаритные выключатели в m/b, применяемые для тех же целей, что и джамперы. Их преимущество - в более легком переключении. Недостатком является большее (по сравнению с джамперами), занимаемое на плате место и более высокая цена. Кроме того, они обычно являются только выключателями, что делает их применение менее гибким, чем применение джамперов.

Системные преобразователи регулируют уровень напряжения. Например, можно установить либо стандартное напряжение 3,5В (для DRAM, AGP и чипсета) или повысить его до 3,65В. Также можно выбрать напряжение на ядре процессора (напр. из значений 2,00, 2,05, 2,10, 2,20, 2,30, 2,40В). Также регулируются частоты на AGP - три положения: автоопределение "по умолчанию", выбор делителя от тактовой (1/1, 2/3).

В современных компонентах стремятся сокращать количество переключателей или джамперов, стараясь переложить все конфигурационные функции на программно-управляемые электронные компоненты. Платы-карты, в которых удастся изжить jumper-ы полностью (но которые требуют конфигурирования), называются Jumperless Cards - карты, свободные от джамперов. Компоненты, которые после установки конфигурируются автоматически, относятся к классу PnP - Plug and Play – «вставляй и играй».

Системные платы выпускает большое количество фирм-производителей. Самые известные торговые марки: Abit, Acorp, Albatron, AOpen, Asus, BioStar, Chaintech, DFI, ECS/Elpina/PCChips, Epox, FIC, Gigabyte, Gainward, Intel, Iwill, Jetway, Leadtek, LuckyStar, MSI, NMC, Octek, PCPartner, Soltek, Soyo, Suma, SuperMicro, Shuttle, Tekram, Tyan, VIA/VPSP, USI, Zida.

Таблица 1 - Систематизация материнских плат

Систематизация по Chipset-у:	Систематизация по форм-фактору:	Систематизация по типу памяти:	Систематизация по разъему ЦП:
от Intel; от VIA; от SiS; прочие;	AT; ATX; NLX, ITX; WTX;	SIMM (SRAM); DIMM (SDRAM); QDR (SDRAM); DDR (DDR-II) DRAM; RIMM (Rambus DRAM);	Socket; Slot;

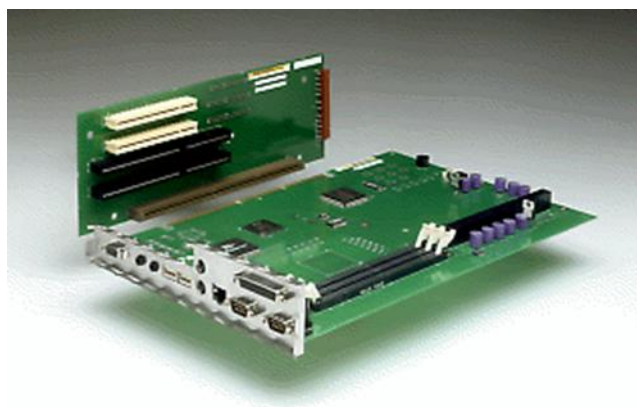


Рисунок 1 – Материнская плата в форм-факторе NLX

Спецификация выпущена Intel в середине 90-х годов. Так обозначается физическая конструкция и типоразмер системных плат и корпусов. Могут встретиться платы типоразмеров ATX, Baby AT, NLX. Еще бывают Form factor-ы ITX и WTX.

ATX раскрывается как AT Extension, что означает развитие AT. Периодически выходят новые версии спецификации ATX, напр. версия 2.03 от 1998 года.

Особенности семейства:

- **ATX** - большой размер (305x244 мм), позволяющий разместить целых 7 слотов карт расширения (PCI, AGP, AMR, CNR, ACR, ISA). Плата совместима с корпусами десктоп и башнями. Является аналогом Baby AT.
- **Mini-ATX** - имеет меньшие размеры (284x208 мм) и позволяет разместить 6 слотов карт. Совместима с теми же корпусами, что и ATX.
- **MicroATX** - уменьшенная ATX (244x244 мм): большая сторона прямоугольника уменьшилась до квадрата. Позволяет разместить 4 слота. Основное применение - офисные компьютеры.
- **FlexATX** - еще уменьшенный вариант MicroATX (229x191 мм). Одно из его применений - Интернет-приставки. MicroATX и FlexATX иногда называют SFX.

На платы AT подается готовое напряжение 3.3В; от блока питания идет один разъем на

плату; компоновка платы, уменьшающая длину интерфейсных кабелей; выключение компьютера из OS. Цвет проводов стандарта ATX следующий:

Цепь	Цвет провода	Пояснение
+5V	Красный	Основное напряжение
+12V	Желтый	Питание двигателей устройств и интерфейсных цепей
-5V	Белый	Не используется. Присутствует для соблюдения стандарта ISA Bus
-12V	Синий	Питание интерфейсных цепей
+3,3V	Оранжевый	Питание процессора
+3,3V Sence	Коричневый	Сигнал обратной связи стабилизатора +3,3V
+5VSB	Малиновый	Дежурный маломощный источник +5V
PS-ON	Зеленый	Сигнал включения источников питания
PW-OK	Серый	Сигнал питания в норме
GND	Черный	Общий, относительно питающих напряжений

В спецификации **LPX** была введена стойка. Т.е. карты расширения вставляются не в материнскую плату, а в подключаемую к плате вертикальную стойку, параллельно материнской плате. Это позволило заметно уменьшить высоту корпуса, максимальное количество подключаемых карт - 2-3 штуки. Еще одно нововведение, родившееся в LPX - это интегрированный на материнскую плату видеочип. Размер корпуса для LPX оставляет 9x13", для Mini-LPX - 8 x 10". После появления NLX, LPX начал вытесняться.



Сверхкомпактный стандарт вновь родился в ноябре 1997г., когда Intel представила спецификацию **NLX**. Сегодня NLX - открытая спецификация системных плат low-profile(v.1.2), т.е. разработкой форм-фактора занимались совместно многие поставщики ПК. В NLX слоты расширения расположены на отдельной карте, что позволяет легко вытаскивать системную плату - все платы расширения остаются в этой отдельной карте. Кроме того, форм-фактор был разработан как решения для построения дешевых NetPC-систем.

Спецификация также требует специальный корпус, так как дизайн NLX позволяет платам расширения и системной плате располагаться на специальных рельсах, для удобного изъятия их из корпуса. Форм-фактор корпуса иногда называется NLX-slim. На системной плате устанавливается специальная карта для плат расширения (riser card). Предназначенна для использования в системах различных типов.

Стандарт Mini-ITX был разработан в компании VIA Technologies и представлен ею в ноябре 2001 года (т.н. инициатива Total Connectivity, в т.ч. разработка "систем-в-материнской-плате"). Системы в корпусе Mini-ITX не требуют вентиляции. Компактность Mini-ITX позволяет изготавливать бесшумные полнофункциональные системы, занимающие очень мало места. Размеры семейства:

- **ITX** - большой размер (215x191 мм).
- **Mini-ITX** - маленький размер (170x170 мм).

В 1998 году родилась спецификация **WTH**, ориентированная на поддержку двухпроцессорных материнских плат любых конфигураций, поддержку сегодняшних и завтрашних технологий видеокарт и памяти. Первые серийные образцы WTH появились осенью 1999 года.

В этой спецификации разработчики попытались отойти от привычной модели, когда материнская плата крепится к корпусу посредством расположенных в определенных местах крепежных отверстий. Здесь она крепится к BAP (Board Adapter Plate), причем способ крепления оставлен на совести производителя платы, а стандартный BAP крепится к корпусу.

WTH описывает архитектуру Flex Slot. На подобных картах могут размещаться любые PCI, SCSI или IEEE 1394 контроллеры, звук, сетевой интерфейс, параллельные и последовательные порты, USB, средства для контроля за состоянием системы.

Тип оперативной памяти.

Плата и процессор поддерживают один (очень редко - два) типа оперативной памяти.

Это **SIMM**-ы, **DIMM**-ы, **DDR DRAM**, **RIMM**-ы. (см. «Внутренняя память»).

В зависимости от типа памяти количество слотов для ее установки может быть различным. Так, если речь идет о SDRAM PC133-памяти, то обычно на плате будет три слота. Для памяти DDR SDRAM может быть два или три слота, а для двухканальной DRDRAM-памяти разводится 4 слота (по два слота на канал). Слоты для установки различных типов памяти не совместимы друг с другом, поэтому ошибиться невозможно. Количество слотов уже не является столь критичным, как ранее, т.к. объем модулей памяти постоянно растет. Часто и двух слотов вполне достаточно, чтобы установить максимальный объем памяти, поддерживаемый современными чипсетами.

Разъемы центрального процессора (ЦП)

Бывает всего два типа разъема (гнезда), в которое устанавливается центральный процессор (рисунок 2):

- **Socket** - плоский разъем для установки микросхемы с выводами, перпендикулярными корпусу.
- **Slot** - щелевой разъем для установки платы с контактами по краю.

Для каждого типа разъема, помимо физического расположения и количества контактов, имеется своя схема соответствия контактов электрическим сигналам. Поэтому, несмотря на то, что процессоры Pentium физически могут быть установлены в любой из разъемов Socket 5/7, правильная (безопасная и полноценная) работа процессора в "чужом" разъеме возможна лишь в случае совместимости существующей разводки сигналов с типом установленного процессора. Это касается выводов коэффициента умножения, управления внутренним кэшем, автоматического определения напряжения питания и т.п. функций, имеющих различное расположение и назначение в процессорах разных типов и серий.

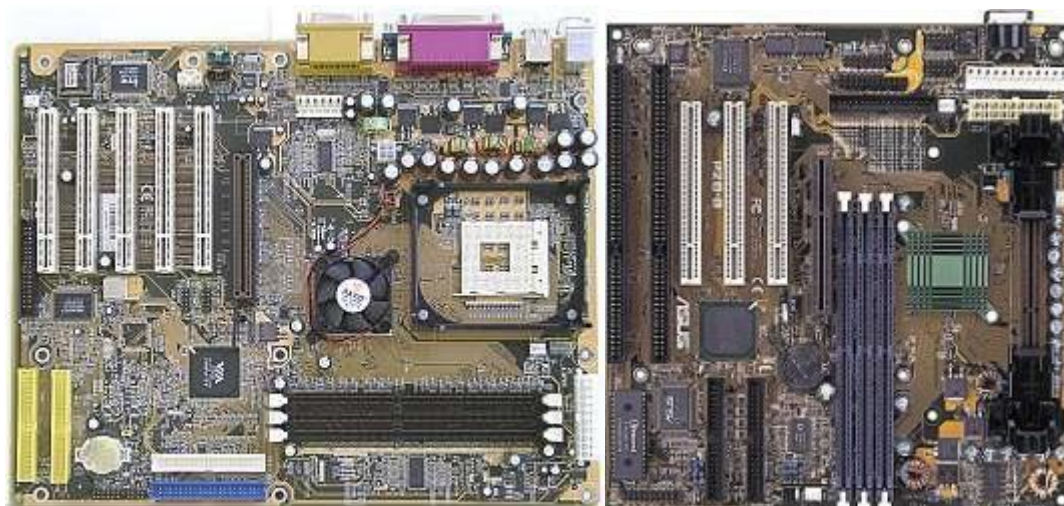


Рисунок 2 – Системные платы типа Socket (слева) и Slot (справа)

Самые распространенные разъемы:

Socket 1 (PGA-169)	под процессоры 486SX/DX/DX2 (5B);
Socket 2 (PGA-238)	под процессоры 486SX/DX/DX2/DX4 (5B);
Socket 3 (PGA-237)	под процессоры 486SX/DX/DX2/DX4 (5B/3,3B);
Socket 462 (Socket A)	под процессоры AMD Thunderbird/Duron до 1.5 ГГц;
Socket 478	под процессоры Intel Pentium 4 для настольных ПК;
Socket 479	под процессоры Intel для мобильных Pentium 4;
Socket 603	под процессоры Intel Intel Xeon (Prestonia);
Socket 754	под процессоры AMD ClawHammer (линейка Athlon XP 64);
Socket 924	под процессор AMD 1 ГГц Dual-Duron (плата Solo);
Socket 940	под процессоры AMD SledgeHammer;
Slot 1 (SEC-242)	под процессоры Pentium II/Celeron;
Slot 2 (SEC-330)	под процессоры Pentium II Xeon;
Slot A	под процессоры AMD Athlon (K7-K75) 500-1000 МГц;
Slot A	под процессоры AMD Thunderbird (K7-K75) 650-1000 МГц;

Socket-переходники используют для установки нового процессора в старое гнездо m/b. Первым появился адаптер S370 -> Slot 1 вследствие политики Intel-а по разделению двух веток процессоров: Pentium II для Slot I для серьезных задач и Celeron под Socket 370 "для дома". Это переходники от фирмы ASUSTeK (S370), от фирмы MSI (MS-6905), PL- iP3/T. Затем появились Socket 423 -> Socket 478, Socket A -> Slot A.

Примечание. AMD неоднократно заявляла о невозможности создания переходников Socket A -> Slot A ... но вот "невозможный" адаптер вышел. Чудо-разработчик - немецкая компания NMC. Название - NMC Slotket A. Согласно заявлениям разработчиков, Slotket A будет работать в любых системных платах с разъемом Slot A.

Типы чипсетов

Чипсет (ChipSets, МСЛ, микросхема системной логики). ПК состоит из устройств, которые подключены к материнской плате и занимаются приемом, обработкой и передачей какой-либо информации. Логической организацией этой работы (с помощью

интегрированных в его состав контроллеров) и занимаются чипсеты, т.е. именно чипсет обеспечивает работу основных подсистем компьютера (центральный процессор, видеоадаптер, оперативная память).

Ранее материнские платы несли на себе до ста микросхем. Первая большая микросхема была выпущена в 1986 году фирмой "Chips & Technologies" и называлась 82C206. Она выполняла функции: 1. контроллера шин, 2. генератора тактовой частоты 3. системного таймера 4. контроллера прерываний 5. контроллера прямого доступа к памяти 6. CMOS-а. Вместе с еще 4-мя микросхемами в качестве буферов и контроллеров получился первый чипсет, названный CS8220. Следующий набор 82C836 Single Chip (SCAT) уже состоял из одной микросхемы. Внешне микросхемы Chipset'a выглядят, как самые большие чипы (после процессора), с количеством выводов от нескольких десятков до сотен. Сегодня на плате размещают 1–2 микросхемы.

Современные ChipSet-ы содержат в себе периферийный контроллер (содержащий два контроллера прерываний IRQ0–7 и IRQ8–15), два контроллера DMA (каналы 0–3 и 4–7), контроллер для Enhanced IDE устройств, часы реального времени с CMOS-памятью (около

100 байт), прямой доступ к памяти, таймер, систему управления памятью и шиной; поддерживают спецификацию шины PCI, синхронную и асинхронную кэш-память, EDO и FPM DRAM, AGP. Обычно в одну из микросхем набора входят также и клавиатурный контроллер (может присутствовать и в виде отдельных чипов). В последние годы в состав МСЛ стали включать и контроллеры внешних устройств.

Чипсет состоит из двух микросхем, называемых северным и южным мостами. Эти мосты выполняют функции связи различных шин и интерфейсов.

Северный мост всегда закрыт радиатором (иногда с вентилятором) и располагается рядом с процессором (северный мост гораздо "быстрее" южного и больше нагревается). Северный мост отвечает за взаимодействие с оперативной памятью (шина памяти), взаимодействие с центральным процессором (системная шина), взаимодействие с графической подсистемой (шина AGP) и взаимодействие с южным мостом. Т.е. северный мост контролирует потоки из четырех шин.

Южный мост соединен с достаточно медленными компонентами, а также с медленными периферийными устройствами (жесткий диск, интерфейс USB и т.д.). Управляя работой высокоскоростных шин, чипсет использует развитые механизмы арбитража, буферизации, очередей и т.о. оптимизирует основные информационные потоки в системе компьютера.

Название набора обычно происходит от маркировки основной микросхемы (напр. SiS471, i440BX). При этом используется только код микросхемы внутри серии: например, полное наименование SiS471 - SiS85C471. Последние разработки используют и собственные имена.

Раньше наборы системной логики (они же ChipSets) выпускали компании "Acer", "Chips & Technologies", "OPTi", "Erso", "Headland Technology", "PC Chips", "Suntac", "Symphony", "UMC", "VLSI".

Но в 1989 году "Compaq" разработала шину EISA, а предоставить контроллер к ней отказалась. И в Intel-е за короткое время сами разработали контроллер. В 1994 году, выпустив процессор i486, фирме Intel пришлось ждать, когда производители чипсетов подготовят подходящий набор системной логики (МСЛ). Чтобы больше не терять время, при выпуске процессора Pentium (в 1995 году), Intel сразу же разработала и выпустила собственный набор.

Так на рынок ChipSets-ов вышел монстр Intel и ... почти все другие фирмы-разработчики скоро закрылись. Даже прародитель чипсетов ("Chip&T") был выжит с этого рынка, а затем и вовсе куплен Intel'ом. С тех пор у Intel'a: вышел новый процессор – вышел к нему чипсет.

Хотя формально и появились новые фирмы-производители чипсетов (всего 12

компаний), но в основном МСЛ выпускают три "монстра" (выделены): Ali (Acer), **AMD**, ATI, IBM, **Intel**, Micron, NVidia, OPTi, ServerWorks, SiS, **VIA Technologies**, VLSI.

Именно ChipSet определяет функции, реализуемые на плате - тип оперативной памяти, поддержку частот, процессорного слота, типов шин и т.п.

Задания:

- 1) Определить тип и форм-фактор материнской платы в учебном системном блоке ПЭВМ.
- 2) Определить название микросхем, размещенных на материнской плате в учебном системном блоке ПЭВМ.
- 3) Определить номиналы питания, подводимые к материнской плате в учебном системном блоке ПЭВМ.
- 4) Определить тип сокета или слота для процессора на материнской плате в учебном системном блоке ПЭВМ.
- 5) Определить тип чипсета материнской платы в учебном системном блоке ПЭВМ.
- 6) Найти и определить коммутационные элементы, имеющиеся процессора на материнской плате в учебном системном блоке ПЭВМ.
- 7) Определить тип северного моста и южного моста на материнской плате в учебном системном блоке ПЭВМ.
- 8) Выполнить работу по демонтажу всех навесных элементов на учебной материнской плате.
- 9) Выполнить работу по демонтажу и извлечению материнской платы из системного блока ПЭВМ.
- 10) Выполнить работу по установке и подключению материнской платы в системный блок ПЭВМ.
- 11) Выполнить работы по подключению всех навесных элементов на учебной материнской плате.

Раздел №4 УРОВЕНЬ МИКРОАРХИТЕКТУРЫ ЭВМ

Лабораторная работа №10 Принципы построения и работы микропроцессоров (2 часа)

Содержание:

1. Принципы построения микропроцессоров.
2. Принципы работы микропроцессоров.

Теоретические сведения:

1. Принципы построения микропроцессоров

Процессор — это блок ЭВМ, предназначенный для автоматического считывания команд программы, их расшифровки и выполнения. Термин, по-видимому, наиболее близко приближается к английскому глаголу "process", имеющему один из вариантов перевода "обрабатывать". И действительно, основное назначение процессора состоит в автоматической обработке информации по заданной программе.

Будучи центральным устройством ЭВМ, процессор во многом определяет ее возможности и производительность. Не случайно при характеристике компьютера тип процессора всегда указывается в первую очередь. Тем не менее не следует забывать, что общая эффективность вычислительной системы зависит не только от процессора, но и от остальных ее компонентов, даже от их согласования между собой. Иными словами, "медленная" память или устаревшая графическая часть всегда могут свести на нет все преимущества новейшего процессора.

Типы процессоров:

Модель	Тактовая частота, МГц	Разрядность	Год
8086	4-8	16	1978
80286	8-20	16	1982
80386	20-40	32	1985
80486	20-100	32	1989
Intel Pentium	60-150	64	1993
Intel Pentium Pro	100-200	64	1995
Intel Pentium II	233-300	64	1997
Intel Pentium III	450-500	64	1999
Intel Pentium IV	до 2800	64	2001
Pentium 4 3,2 ГГц	3200	64	2003

Традиционными главными частями процессора являются арифметико-логическое устройство (АЛУ) и устройство управления (УУ).

Первоначально процессор изготовлялся из отдельных деталей: транзисторов,

микросхем невысокого уровня интеграции (с небольшим числом элементов) и т. д.; его размеры могли быть весьма значительными. Пожалуй, наиболее впечатляюще выглядел процессор ЭВМ серии ЕС (Единая Система ЭВМ относится к третьему поколению). Он представлял собой отдельный весьма внушительный шкаф.

Прогресс в области микроэлектроники привел к тому, что весь процессор удалось разместить внутри одного кристалла. Таким образом, он стал отдельной самостоятельной микросхемой и получил новое название — *микропроцессор*.

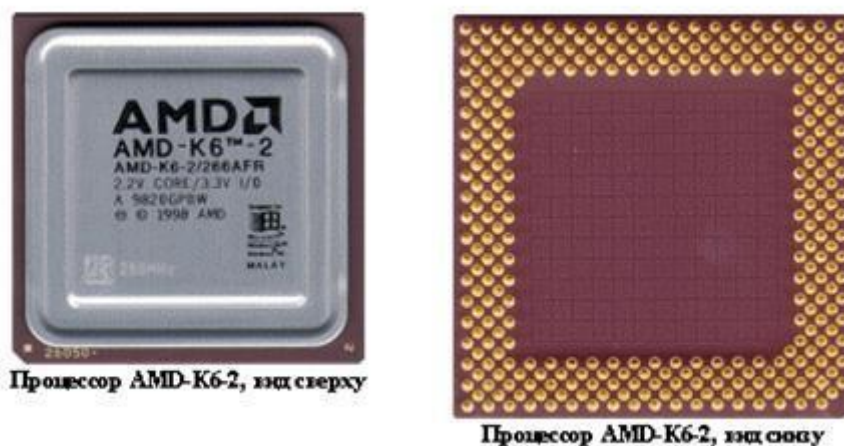


Рисунок 1 – Процессор AMD-K6-2

Следует иметь в виду, что размер корпуса в основном определяется необходимостью разместить несколько сот металлических выводов, с помощью которых процессор присоединяется к плате; собственно кристалл значительно меньше и имеет размер порядка 1 см.

То, что процессор удалось поместить внутри одной микросхемы, не просто уменьшило размеры этого узла, но и создало предпосылки для существенного увеличения скорости работы процессора (благодаря сокращению длины соединений) и повышения его надежности (за счет отсутствия внешних проводников). К сожалению, есть и отрицательные последствия: уменьшение габаритов процессора приводит к ухудшению условий теплоотдачи от микроскопических его элементов, что существенно повышает требования к теплоотводящим свойствам конструкции. Не случайно для охлаждения современных процессоров, особенно работающих на очень высоких частотах, обязательно используются металлические радиаторы достаточно большой площади и всевозможные вентиляторы (часто их жаргонно называют "кулерами" — т. е. по-русски "охладителями", "холодильниками"). Последние модели микропроцессоров даже содержат внутри кристалла специальные датчики температуры и поэтому способны автоматически выключаться, предохраняя себя от перегрева.

Сегодня развитие микропроцессоров требуется не только для вычислительной техники, где они используются в качестве центрального процессора и для управления сложными периферийными устройствами. Все большее число таких микросхем ставится в изделия, напрямую не связанные с ЭВМ, в том числе и бытовые: лазерные аудио- и видеопроекторы, телетекст и пейджинговая связь, программируемые микроволновые печи и стиральные машины, и многое-многое другое. Очевидно, что количество таких управляемых микропроцессорами устройств будет все время возрастать.

Задания:

- 1) Определить тип процессора, установленный в учебном системном блоке ПЭВМ, и его основные характеристики.
- 2) Составить таблицу со сравнительными характеристиками современных процессоров.
- 3) Составить блок-схему внутренней структуры типового микропроцессора.
- 4) Составить структурную схему регистров типового микропроцессора.
- 5) Разработать алгоритм работы процессора.
- 6) Составить блок-схему модулей процессора Pentium II.

**Лабораторная работа №11 Принципы работы микропроцессоров
(2 часа)**

Содержание:

1. Принципы работы микропроцессоров.

Теоретические сведения:

За 17 лет семейство процессоров x86 проделало длинный путь: процессор P6 обеспечивает почти в тысячу раз более высокое быстродействие, чем 8086. Вот несколько важных дат из истории эволюции индустрии ПК.

1978	Intel 8086. 16-разрядные регистры, 16-разрядная шина данных. Выпущен в июне 1978 г. 29 тыс. транзисторов, 4,77-10 МГц, менее 1 млн. операция/с. Начальная цена: 360 долл.
1979	Intel 8088. Процессор 8086 с 8-разрядной шиной данных. Выпущен в июне 1979 г. 4,77-8 МГц.
1981	выпущен IBM PC с процессором 8088.
1982	Intel 80286. 16-разрядные регистры, 16-разрядная шина данных, работа в защищенном режиме. Выпущен в феврале 1982 г. 134 тыс. транзисторов, 6-12 МГц, 1-2 млн. операция/с. Начальная цена: 360 долл.
1984	выпущен IBM PC AT с процессором 80286.
1985	Intel 80386. 32-разрядные регистры, 32-разрядная шина данных, работа в защищенном режиме. Выпущен в октябре 1985 г. 275 тыс. транзисторов, 16-33 МГц, 6-12 млн. операция/с. Начальная цена: 299 долл.
1985	выпущена Microsoft Windows.
1986	выпущена модель Compaq DeskPro 386/16.
1987	выпущено семейство IBM PS/2 с архитектурой Micro Channel и операционной системой OS/2.
1988	Intel 80386SX. Процессор 80386 с 16-разрядной шиной данных. Выпущен в июне 1988 г., 16-33 МГц.
1989	Intel 486DX. 32-разрядные регистры, 32-разрядная шина данных. Встроенный математический сопроцессор. Выпущен в апреле 1989 г., 1,2 млн. транзисторов, 25-50 МГц, 20-40 млн. операция/с. Начальная цена: 950 долл.
1990	Выпущены ПК с процессором 486DX/25.
1991	Intel 486SX. Процессор 486DX без математического сопроцессора. Выпущен в апреле 1991 г., 16-33 МГц.
1992	Intel 486DX2. Процессор 486DX с удвоением тактовой частоты. Выпущен в марте 1989 г., 50-66 МГц.

1993	Intel Pentium. 32-разрядные регистры, 64-разрядная шина данных. Суперскалярна архитектура. Выпущен в марте 1993 г., 3,2 млн. транзисторов, 60- 133 МГц, 100 - 200 и более млн. операция/с. Начальная цена: 878 долл.
1994	Intel 90-МГц Pentium "P54C" . Pentium с напряжением питания 3,3 В.
1994	Intel DX4. Процессор 486DX с утроением тактовой частоты и кэш-памятью увеличенной емкости. Выпущен в марте 1994 г., 75-100 МГц.
1994-1996	Конкуренты Pentium: NexGen Nx586, выпущен в сентябре 1994 г. Cyrix M1, должен появиться в конце 1995 г. AMD K5, должен появиться в начале 1996 г.
1995	Intel P6. 32-разрядные регистры, 64-разрядная шина данных. Суперскалярная архитектура; исполнение с изменением последовательности команд; встроенный кэш второго уровня. Должен появиться во второй половине 1995 г. 5,5 млн. транзисторов (исключая кэш второго уровня). Начальные тактовые частоты: 133 и 150 МГц.

Оперативная информация о производительности P6

Для получения самых последних данных о процессорах вы можете получить из архива (<http://www.ziff.com/~pcmag>) с помощью анонимного FTP (<ftp.pcmag.ziff.com>) или в библиотеке аппаратных средств журнала PC Magazine сети ZiffNet/CompuServe.

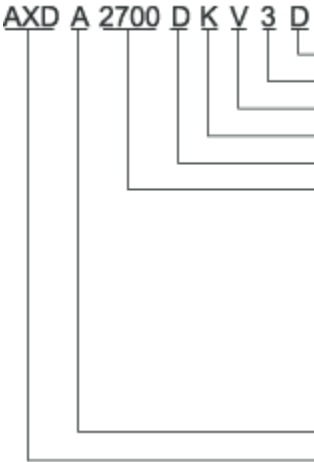
Классификация процессоров

Марка процессора	Комментарий
Intel	
Pentium	Самые первые процессоры семейства P5 появились в далеком марте 1993-го. Тогда Intel, чтобы не повторить ошибки с i486 (суд отклонил иск к AMD по поводу названия) решила дать своему детищу имя, которое в последствие стало нарицательным. Первое поколение Pentium носило кодовое имя P5 он же 80501, напряжение питания 5 вольт, расположение выводов - "матрица" и работало на тактовых частотах 60 и 66 МГц, выпускаясь по 0.80-микронной технологии (правда стоит отметить, что частота шины у этих процессоров была равна частоте ядра). Выпускались они исключительно под Socket 4. Следующим шагом в развитии этого семейства стал P54 он же 80502, напряжение питания 3.3 вольта, расположение выводов - "шахматная матрица". Появился ровно через год после P5. При его изготовлении использовался уже 0.50, а затем и 0.35-микронный технологический процесс. Тактовая частота находилась в пределах 75-200 МГц, а шина - 50-66 МГц. Объем-кэш памяти первого уровня - 16Кбайт, причем впервые был применен отдельный кэш - 8 Кбайт на данные и 8 Кбайт на инструкции. Форм-фактор - Socket 5. Архитектура IA32, набор команд не менялся со времен i386.
Pentium w/MMX technology	Следующим большим шагом стал выпуск P55: процессора в котором впервые был реализован новый набор из 57 команд MMX. Произошло это 8 января 1997 года. С развитием технологии процессоры стали выпускаться по 0.35 мкм технологии. Изменилось напряжение питания (уменьшилось до 2.8 вольта), соответственно потребовались изменения в конструктивах системных плат - требовалась установка дополнительного стабилизатора напряжения. Объем кэш-памяти первого уровня был увеличен в два раза - 32 Кбайта. Внутренняя тактовая частота составляла 166-233 МГц, а

	частота шины - исключительно 66 МГц. Рассчитан на Socket 7.
Tillamook	Процессор, изначально создавшийся для применения в ноутбуках. Благодаря усовершенствованному 0.25-микронному процессу удалось одновременно поднять тактовую частоту вплоть до 266 МГц, а также снизить напряжение ядра и рассеиваемую мощность. Таким образом, мобильные компьютеры встали в один ряд с настольными. Он является продолжением линейки Pentium и включает 32 Кбайта L1 кэша и набор команд MMX. Выпускался для работы на тактовых частотах от 133 до 266+ МГц с частотой шины 60-66 МГц. Тип упаковки: TSP и MMC (хотя существуют переходники для установки Tillamook в гнездо Super7). Появился 8 января 1997 года.
Pentium Pro	Первый процессор шестого поколения. Довольно революционный для своего времени. В нем впервые Intel решилась применить кэш-память второго уровня, объединенную в одном корпусе с ядром и оперирующую на частоте процессора. Имел очень высокую себестоимость изготовления, которая так практически и не снизилась за все время его существования с 1 ноября 1995 года. Выпускался как по 0.50, так и по 0.35-микронной технологии. 0.35 мкм позволили увеличить объем кэша. Кэш второго уровня имел объем 256, 512, 1024 или 2048 Кбайт. Тактовая частота - от 150 до 200 МГц. Частота системной шины - 60-66 МГц. Кэш первого уровня объемом 16 Кбайт. Выпускался исключительно для Socket 8. Pentium Pro поддерживал все инструкции процессора Pentium (естественно, не MMX), а также ряд новых по сравнению с Pentium инструкций (cmov, fcomi, и т.д.). Введена двойная независимая шина (DIB). В дальнейшем все новшества унаследовал Pentium II (в свое время, задолго до своего выпуска, Klamath представлялся просто как Pentium Pro с добавлением MMX и улучшенной работой с 16-битными приложениями).
Pentium II/III	семейство P6/6x86, первые представители появились в мае 1997 года. Объединяет общим именем процессоры, предназначенные для разных сегментов рынка. Pentium II (Klamath, Deschutes, Katmai и др.) для массового рынка ПК среднего уровня, усовершенствованная технология изготовления 0.25 микрон, питание - 2.0 В. Соответственно, удалось поднять тактовую частоту 266-450+ МГц, частота системной шины 66-100 МГц, кэш-память второго уровня 512 Кбайт размещена на процессорной плате, вышел 26 января 1998, Slot 1. Кэш первого уровня 32 Кбайта. Последнее ядро официально применявшееся в процессорах Pentium II, хотя последние модели Pentium II 350-450 шли с ядром, уже больше напоминавшим Katmai - только, естественно, с обрезанным SSE. Да и картридж тогда уже стал SECC2 (кэш с одной стороны от ядра (а не с двух, как в стандартном

		Deschutes, измененное крепление кулера.). Celeron (Covington, Mendocino, Dixon и др.) - для недорогих low-end компьютеров - революционный в некотором смысле процессор: Intel наконец-то обратила внимание на массовый рынок недорогих компьютеров. В общем, это целое семейство недорогих процессоров как с кэшем второго уровня, так и без него. В данный момент выпускались или выпускаются следующие его представители Covington, Mendocino, Dixon. Впервые появился в апреле 1998 года. Выпускается в вариантах для Socket 370, Slot 1. Xeon (Xeon, Tanner, Cascades и др.) - для высокопроизводительных серверов и рабочих станций - спустя несколько лет Intel решила на выпуск замены Pentium Pro. Как и в его предшественнике, кэш память второго уровня здесь оперирует на частоте процессора. Правда, если в PPro кэш и ядро были объединены одним корпусом, то в Xeon - одним картриджем. Это первый процессор для Slot 2, и предназначен в первую очередь для мощных серверов и рабочих станций. Способен работать в мультипроцессорных конфигурациях. Построен на ядре Deschutes и выпускается, как и собственный кэш, по 0.25 микронной технологии. Кстати, сам кэш имеет объем 512, 1024, 2048 Кбайт, что во многом определяет высокую стоимость и тепловыделение. Имеет модификации для Slot 1, Slot 2, Socket 370, а также варианты в мобильном исполнении.
Pentium Willamette	4	принципиально новый IA-32 процессор Intel для обычных PC. Использует новую системную шину вместо старой GTL+ - Quad Pumped 100 МГц, с результирующей частотой 400 МГц.. Кэш L1 - 8 Кбайт, L2 - 256 Кбайт, предпринят ряд шагов, направленных на увеличение производительности: добавленные исполнительные модули, декодеры, увеличенный объем буферов, и т.д. Введен новый набор инструкций - SSE2, на который Intel делает большую ставку в плане увеличения производительности нового процессора. Вышел 20 октября 2000 года с тактовой частотой 1.4-1.5 ГГц. В течение 2001 года должен дорасти до 2 ГГц, после чего уступить свое место Northwood. Форм-фактор - Socket-423.
AMD		
K5		Первый процессор AMD, который всерьез предназначался для конкуренции с Pentium. Платформа - Socket 5. Подобно Cyrix 6x86, использовал PR-рейтинг, от 75 до 166 МГц. При этом используемая частота системной шины составляла от 50 до 66 МГц. Кэш L1 - 24 Кбайт (16 Кбайт для инструкций и 8 для данных), кэш L2 на материнской плате, работает на частоте системной шины. Существовало четыре версии процессора: K5-75, 90, 100 (PR-rating соответствовал частоте процессора, технологический процесс 0.6 мкм); процессор K5-100 (0.35 мкм), процессоры K5-PR120, PR133 (PR-рейтинг соответствует частотам 90 и 100 MHz, технологический процесс 0.35 мкм, переработанное ядро) и процессор K5-PR166 (реальная частота 66MHz x 1.75) с нестандартным коэффициентом умножения.

K6	Начал поставляться с апреля 1997 года (это Model 6), на месяц раньше выхода Pentium II, производился на базе 0.35 мкм (позднее 233 MHz K6 производились с использованием 0.25 мкм процесса) технологического процесса. Процессор работал на частоте от 166 до 233 МГц (причем последний - официально разогнанный вариант - 3.2/3.3В вместо стандартных 2.9/3.3В). Был создан на базе дизайна процессора 686, созданного приобретенной AMD компанией NexGen. По сравнению со своим предшественником, получил модуль MMX, увеличился объем кэша L1 - до 64 Кбайт (по 32 Кбайт для инструкций и данных). Возросшая тактовая частота позволила AMD отказаться от PR-рейтинга. В K6 предлагались решения, которые позволяли добиться большей производительности на той же частоте, а именно исполнение по предположению и внутренняя риск-подобная организация. Все последующие процессоры от AMD унаследовали это свойство. Позднее начались поставки K6 Model 7 (мобильный вариант) с частотами 266 и 300 МГц, технологический процесс 0.25 мкм, FSB 66 MHz.
K6-2	Следующее поколение K6. Вышел в мае 1998 года, основными усовершенствованиям относительно его предшественника стали поддержка дополнительного набора инструкций 3DNow! и частоты системной шины 100 МГц. Кэш L1 64 Кбайт (по 32 Кбайт для инструкций и данных), кэш L2 находится на материнской плате, и может иметь объем от 512 Кбайт до 2 Мбайт, работая на частоте системной шины. Первоначально был выпущен на частоте 266 МГц, сегодня максимальная тактовая частота составляет 475 МГц, в скором времени должен выйти 500 МГц вариант. Было две модели процессора K6-2: первая, работающая на частотах 266 (66*4), 300 (100*3), 333 (95*3.5), 350 (100*3.5) и 366 (66.*5.5) МГц. И вторая модель AMD, работающая с частотами 200, 233, 350, 380, 400, 450 и 475 МГц. В ней используется новое ядро, такое же, как в K6-III. Главное отличие в новом ядре (CXT) это модифицированный метод работы с кэшем.
Sharptooth (K6-III)	Первый процессор от AMD, имеющий кэш второго уровня объединенный с ядром. Последний их процессор, сделанный под платформу Socket 7. Фактически, представляет из себя просто K6-2 с 256 Кбайт кэша L2 на чипе, работающих на той же частоте, что и процессор. Кэш L1 имеет объем 64 Кбайт (по 32 Кбайт для инструкций и данных), кэш L3 находится на материнской плате, и может иметь объем от 512 Кбайт до 2 Мбайт, работая на частоте системной шины. Был выпущен в феврале 99 года в вариантах 400 и 450 МГц Sharptooth-SC50 - переход на 0.18 мкм.
K6-2+	Один из последних Socket7 процессоров AMD. И первый Socket7 процессор, сделанный с использованием 0.18 мкм техпроцесса. Должен содержать на чипе 128 Кбайт кэша L2, работающих на частоте процессора. Предполагаемая скорость - от 533 МГц. Естественно, поддержка 3DNow!

K7 (Athlon)	Первый проект AMD, в котором она была вынуждена отойти от даже частичного заимствования архитектур Intel, и предложить рынку абсолютно свой вариант платформы для PC. Процессор имеет непревзойденный для сегодняшних x86 процессоров объем кэша первого уровня - 128 Кбайт (по 64 Кбайт для инструкций и данных). Кэш L2 - 512 Кбайт, работающий на 1/2, 2/5 или 1/3 частоты процессора. Системная шина - EV-6, та же, что и в процессорах Alpha, что потенциально дает возможность создания материнских плат, поддерживающих оба процессора. Частота системной шины - 200 МГц, но имеет потенциал до 400 МГц и выше. Поддерживаемые наборы инструкций - MMX, расширенный по сравнению с K6-III 3DNow!. Форм-фактор - Slot A. Первый процессор AMD, получивший при выходе собственное имя - Athlon. На сегодня доступны модели 500-1000 МГц. Подклассы: K75 - алюминиевые соединения, K76 - медные. OPN  Частота системной шины (FSB): C = 266, D = 333 Размер кэша L2 : 3 = 256 Кбайт Максимальная температура: T = 90°C, V = 85°C Vcore: L = 1.50 V, U = 1.60 V, K = 1.65 V Тип упаковки: D = OPGA Модель : 1700 = реальная тактовая частота 1467 МГц 1800 = реальная тактовая частота 1533 МГц 1900 = реальная тактовая частота 1600 МГц 2000 = реальная тактовая частота 1667 МГц 2100 = реальная тактовая частота 1733 МГц 2200 = реальная тактовая частота 1800 МГц 2400 = реальная тактовая частота 2000 МГц 2600 = реальные тактовые частоты 2083 МГц или 2133 МГц 2700 = реальная тактовая частота 2167 МГц Класс энергопотребления: A = Desktopный процессор Архитектура: AXD = AMDAthlon™ XP Model 8 с архитектурой Quar Маркировка процессоров AMD Athlon XP на ядре Thoroughbred
ClawHammer	первый 64-бит процессор AMD. Или, по крайней мере, частично 64- бит. В отличие от Itanium, этот процессор будет ориентирован главным образом на 32-бит инструкции, нежели наоборот. Одновременно с его выходом ожидается появление новой шины - HyperTransport (Lightning Data Transport - LDT), используемой для связи с процессорами и устройствами ввода/вывода. LDT должна стать не заменой, а дополнением к системной шине EV6 или EV7. Предполагаемая скорость - 2 ГГц и выше, предполагаемый срок выхода - первая половина 2002 года.
Cyrix 6x86	Или M1. Для оценки производительности использовался P-Rating, когда производительность процессора сравнивается со скоростью процессора Pentium, на которой ему пришлось бы работать для достижения той же производительности. PR 6x86 составлял от 120 до 200 МГц. Есть устойчивое мнение, что первоначальные варианты процессора были знамениты наличием ошибок, ведущих к частым зависаниям PC, и необходимости для производителей программ выпускать патчи специально под этот процессор. На самом деле все проблемы были связаны именно с ошибками в программах. Особенно известны проблемы с программами, написанными на Clipper. Слухи

	про проблемы под Windows NT не подтвердились. Кэш первого уровня - 16 Кбайт (единый). Частота системной шины - от 50 до 75 МГц. Платформа - Socket 5, затем (когда появилась версия с двойным питанием) Socket 7.
MediaGX	Ответвление в семействе процессоров Cyrix, первый процессор, сделанный по идеологии PC-on-a-chip. К ядру 5x86 были добавлены контроллеры памяти и PCI, в чип интегрирован видеоускоритель, с кадровым буфером в основной памяти PC. И лишь в самых последних моделях используется ядро 6x86. В чипе-компаньоне реализован мост PCI-ISA и интегрирован звук. PR-рейтинг от 180 до 233 МГц, кэш L1 16 кбайт (единый). Производился по техпроцессу 0.5 мкм. Сегодня National делает на базе этого процессора два продукта - Geode GXLV (0.35 мкм, 166-266 МГц) и Geode GX1400 (с добавленной аппаратной поддержкой MPEG-2, Dolby AC3 и т.д).
6x86MX	(позднее переименован в М-П) - несколько переработанный для большей производительности 6x86. Вчетверо увеличился кэш L1 - до 64 Кбайт (единый), увеличилась общая производительность процессора, добавился блок MMX, появилась поддержка раздельного питания. Использовал частоту системной шины от 60 до 75 МГц, имел P-рейтинг от 166 до 266 МГц. Процессоры 6x86MX делала и компания IBM. Их 6x86MX имели рейтинг от 166 до 333 и были рассчитаны на шину 66, 75 или 83 МГц. Позднее, по маркетинговым соображениям, Cyrix переименовал свои процессоры в МП, а IBM до конца сотрудничества продавало их под маркой 6x86MX.
МП	Последний процессор Cyrix, начал производиться в марте 98 года. Кэш первого уровня - 64 Кбайт (единый), L2 - как обычно, для Socket 7, находится на материнской плате, и имеет объем от 512 Кбайт до 2 Мбайт, работая на частоте системной шины. Поддерживаемые наборы инструкций - MMX. Использует P-рейтинг. Реальная скорость в МГц, как правило, значительно ниже - так, Cyrix МП PR366 имеет реальную частоту 250 МГц. При производстве применяется техпроцесс 0.25 мкм. Продаются модели, имеющие PR-рейтинг 300-433 МГц.
Gobi (МП+)	Процессор имеет сразу два кодовых имени, что несколько необычно. Вначале он назывался Jedi, но в дальнейшем, по требованию владельца авторских прав на это название, Lucas Film, был переименован в Gobi. Первый из процессоров Cyrix, рассчитанный на платформу Socket 370. Поддерживаемые наборы инструкций - MMX, 3D Now!. Значительно переработан блок операций с числами с плавающей запятой. Кэш L1 - 64 Кбайт (единый), кэш L2 - 256 Кбайт на чипе, работающие на полной частоте процессора.
Mojave (М3)	Процессор содержит 32 Кбайт (по 16 для данных и инструкций) кэша L1 и 256 Кбайт интегрированного на чипе кэша L2. 0.18 мкм техпроцесс, значительно улучшенная по сравнению с предшественниками архитектура. Поддерживаемые наборы инструкций - MMX, 3D Now!. Скорость в момент выпуска - 600-800 МГц (не P-рейтинг), частота системной шины - 100-133 МГц. Чип будет иметь интегрированные контроллер памяти и 3D ускоритель. В связи с покупкой Cyrix VIA вряд ли когда-нибудь появится.
Rise	

mP6	Первый процессор компании Rise, преимущественно предназначен для ноутбуков, использующих Socket 7 процессоры. Соответственно, отличается очень малым тепловыделением. Кэш L1 - 16 Кбайт (по 8 кбайт для данных и инструкций), L2 - на материнской плате, от 512 Кбайт до 2 Мбайт, работает на
	частоте системной шины. Поддерживается дополнительный набор инструкций MMX. При оценке производительности своих процессоров, Rise, как и Cyrix, использует PR-рейтинг. PR-рейтинг от 166 до 366 МГц.
mP6 II	То же самое, по сравнению с mP6, что и K6-III по сравнению с K6-2. То же ядро, к которому добавлено 256 Кбайт кэша L2 на чипе. Была обещана поддержка SSE, производительность от PR-200 и выше. В августе 99 было объявлено об отмене планов по выходу процессора, в связи со значительным его удорожанием после добавления кэша L2 на чип.
Centaur	
Winchip C6	Процессор изначально создавался, будучи ориентированным на дешевые PC, как следствие, по скорости уступал по производительности своим конкурентам. Системная шина - 60, 66, 75 МГц, платформа - Socket 5. При производстве использовался техпроцесс 0.35 мкм. Поддерживаемый набор инструкций - MMX. Вышел в октябре 1997 года, работал на скоростях от 180 до 240 МГц.
Winchip-2	Производится по техпроцессу 0.25 мкм. Кэш L1 64 Кбайт (по 32 Кбайт для инструкций и данных), кэш L2 находится на материнской плате, 512 Кбайт - 2 Мбайт. Поддерживаются наборы инструкций MMX и 3DNow!. Платформа - Socket 7. От Winchip отличается значительно ускорившейся работой с числами с плавающей запятой. Появилась поддержка частоты системной шины 100 МГц. Появился в ноябре 98 года, скорость от 200 до 300 МГц.
Winchip-3	Кэш L1 64 Кбайт (по 32 Кбайт для инструкций и данных), кэш L2 - 128 Кбайт на чипе, работающих на частоте процессора, кэш L3 находится на материнской плате, 512 Кбайт-2 Мбайт. Планировался к выходу в первой половине 99 года, со скоростью 300 МГц и выше. В связи с покупкой Centaur VIA, по всей видимости, выход процессора был отменен.
Winchip-4	Планировался к выходу в конце 99 года, на скоростях порядка 400-500 МГц, а при переходе на 0.18 мкм техпроцесс - и 500-700 МГц. Предполагалась смена форм-фактора.
VIA	
Joshua	Первым процессором VIA, намеченным к выпуску, стал приобретенный вместе с Cyrix их дизайн Gobi. Детали см. в разделе Cyrix.
Samuel (C5A)	Ядро Winchip4, доставшееся VIA в наследство от Centaur, чип работает на частоте 500-700 МГц, производится National и TSMC с использованием 0.18 мкм техпроцесса. Процессор использует SIMD-набор 3D Now!, форм-фактор - Socket-370. Кэш L1 128 Кбайт. Быстро сменил собой Joshua под тем же именем Cyrix III. Тактовая частота - 500-667 МГц.
Samuel2 (C5B)	Следующее ядро группы Centaur. Добавился кэш L2 объемом 64 Кбайт, увеличилась тактовая частота - 667-800+ МГц. Частота системной шины 100/133 МГц, форм-фактор - Socket-370.

Ezra (C5C)	Совместная разработка групп Cyrix и Centaur. Первое действительно новое ядро VIA. Первый их процессор с поддержкой SSE, кэш L1 128 Кбайт, кэш L2 64 Кбайта. Предполагается значительно возросшая производительность и тактовая частота. Процесс 0,15 с переходом на 0,13 мкм. Выход во втором квартале 2001 с тактовой частотой от 750 МГц с потенциалом роста выше 1 ГГц. TSMC подтвердила информацию о том, что она сделала 1 ГГц Ezra.
C5X	Процессор от VIA, который работает на частотах порядка 1.2-1.5 ГГц Если у Ezra - 128 Кбайт кэша L1 и 64 Кбайт - L2, то у этого кэш L2 - 256
	Кбайт. Плюс, впервые для VIA, использование медных соединений. Площадь чипа, по сравнению с Ezra, увеличится весьма незначительно - с 52 до 65 кв.мм.
SiS	
5xx	Ядро mP6 от Rise (у SiS есть лицензия на него), интегрированное видео и коммуникации. Ответ на VIA Matthew
Transmeta	
Crusoe	19 января Transmeta, наконец, подвела итог своей 5 летней деятельности, объявив аппаратно-программный комплекс - процессор Crusoe. Компания решила не конкурировать с Intel и AMD в той области, где те традиционно сильны, позиционируя свой процессор, как лучшее решение для мобильных систем, НРС, и т.д. Для начала процессор выходит в двух вариантах - 333-400 МГц TM3120 и 500-700 МГц TM5600. Для первого объем кэша составляет 96 Кбайт L1, для второго - 125 Кбайт L1 + 256 Кбайт L2. Плавающее энергопотребление составляет от 10-20 мВт до 1-3 Вт, в зависимости от выполняемой работы. TM5800 - 0,13 мкм версия процессора, которая дойдет до 1 ГГц.

Архитектура процессора

Существуют процессоры различной архитектуры.

CISC (Complex Instruction Set Computing). Эта конфигурация для повышения гибкости и расширения возможностей предусматривает:

- увеличение числа различных по формату и длине команд;
- введение большого числа различных режимов адресации;
- сложную кодировку инструкции.

Процессору с архитектурой CISC приходится иметь дело с более сложными инструкциями неодинаковой длины. Выполнение одиночной CISC-инструкции может происходить быстрее, однако обрабатывать несколько таких инструкций параллельно сложнее.

Облегчение отладки программ на ассемблере влечет за собой загромождение узлами микропроцессорного блока. Для повышения быстродействия следует увеличить тактовую частоту и степень интеграции, что вызывает необходимость совершенствования технологии и, как следствие, удорожание производства.

RISC (Reduced Instruction Set Computing). Процессор с сокращенным набором команд. Система команд имеет упрощенный вид. Все команды одинакового формата с простой кодировкой. Обращение к памяти происходит посредством команд загрузки и записи, остальные команды типа регистр-регистр. Команда, поступающая в CPU, уже разделена по полям и не требует дополнительной дешифрации.

Часть кристалла освобождается для включения дополнительных компонентов. Степень интеграции ниже, чем в предыдущем архитектурном варианте, поэтому при высоком

быстродействию допускается более низкая тактовая частота. Команда меньше загромождает ОЗУ, CPU дешевле. Программной совместимостью указанные архитектуры не обладают. Отладка программ на RISC более сложна. Данная технология может быть реализована программно-совместимым с технологией CISC (например, суперскалярная технология).

Поскольку RISC-инструкции просты, для их выполнения нужно меньше логических элементов, что в конечном итоге снижает стоимость процессора. Но большая часть программного обеспечения сегодня написана и откомпилирована специально для CISC-процессоров фирмы Intel. Для использования архитектуры RISC нынешние программы должны быть перекомпилированы, а иногда и переписаны заново. **MISC** (Multipurpose Instruction Set Computer) сочетает преимущества вышерассмотренных архитектур. Элементная база состоит из двух частей, которые либо выполнены в отдельных корпусах, либо объединены. Основная часть – RISC CPU, расширяемый подключением второй части – ПЗУ микропрограммного управления. Система приобретает свойства CISC. Основные команды работают на RISC CPU, а команды расширения преобразуются в адрес микропрограммы. RISC CPU выполняет все команды за один такт, а вторая часть эквивалентна CPU со сложным набором команд. Наличие ПЗУ устраняет недостаток RISC, выраженный в том, что при компиляции с языка высокого уровня микрокод генерируется из библиотеки стандартных функций, занимающей много места в ОЗУ. Поскольку микропрограмма уже дешифрована и открыта для программиста, то времени выборки из ОЗУ на дешифрацию не требуется.

Режимы работы процессора

CPU I386 полностью совместим по объектному коду с семейством Ix86 и является его дальнейшим развитием. CPU может быть использован в следующих режимах работы:

Real Mode (RM) – режим реальной адресации, соответствующий работе системы с CPU 8086, используется только в MS DOS. Область адресов, доступных системе (1 Мбайт), не защищена. Реализованы двадцать адресных линий, режим однопользовательский. Однако при этом работают 32-битовые регистры CPU. По умолчанию используются все имеющиеся команды, длина операндов 16 бит. Для работы с 32-разрядными операндами и использования дополнительных режимов адресации применяют специальный префикс переадресации. 32-разрядный адрес не превышает границы сегмента 64 Кбайт (0000H- FFFFH), в противном случае констатируется особая ситуация - 13H. Для обеспечения доступа к адресному пространству шириной до 1 Мбайт в работе участвуют линии A [19/02], ВЕО#-ВЕЗ#. Исполнительный адрес всегда соответствует физическому, страничный механизм отключен. Все сегменты могут находиться в состоянии записи, считывания или выполнения.

Protected Mode (PM) – режим виртуальной адресации (защищенный режим).

Виртуальная адресация – это способ организации доступа к информации, при котором большая ее часть располагается не в физическом ОЗУ, а в устройствах внешней памяти (УВП), откуда она периодически подкачивается в ОЗУ (swapping), что создает иллюзию расширения его размеров.

При включении ПЭВМ первоначально принимает RM. Для перевода в PM используют системные команды LMSW, SMSW. При установке бита PF=1 в MSW CPU переходит в PM, в котором реализуется доступ к 4 Гбайт ОЗУ в 32-битовом адресном физическом пространстве исполнительных адресов. Доступ к 64 Тбайт памяти реализуется в виртуальном (логическом) адресном пространстве.

Предусмотрена защита по многоуровневому принципу операционной системы и прикладных задач. Реализуется мультипрограммность.

Для обратного перевода из PM в RM команды LMSW, SMSW не используются. Система должна быть перезагружена либо аппаратно («холодный» старт), либо программно-аппаратно – через порты 60H, 64H контроллера 8042 командой вывода out и далее через внутренний сигнал RC и узел Shut down ко входу Reset I386 («теплый» старт осуществляется нажатием клавиш).

Для осуществления программного перехода из РМ в RM может быть использована команда MOV CR0 (LCR0).

Protected Mode 86 (VM 86) – режим виртуальной адресации 86 - устанавливает исполнительную среду I86 внутри защищенной многозадачной среды РМ CPU I386. При этом поддерживается исполнение всех программ, написанных для предыдущих поколений. Первоначально в рамках VM 86 оборудование формирует 20-разрядный линейный адрес по системе, аналогичной RM. Однако, благодаря включению механизма страничной адресации и системе двухуровневой защиты, линейный адрес шириной в 1 Мбайт может быть разбит на 256 страниц по 4 Кбайт каждая, и размещен в физическом адресном пространстве до 4 Гбайт. В этом объеме физической памяти можно в окнах по 1 Мбайт расположить множество копий MS DOS или других операционных систем и их программ, представляющих отдельные виртуальные машины, работающие в мультизадачном режиме. Вход и выход в режим VM86 возможен посредством перезагрузки регистра флагов (бит 17), либо при переключении с задачи на задачу с привлечением сегмента состояния задачи (TSS), либо в процедуре прерывания (команда IRET), либо при использовании команды POPF.

Page Protected Mode – режим страничной адресации.

Это альтернативный режим управления памятью, позволяющий разделить крупные массивы информации на более компактные блоки размером по 4 Кбайт. Аппаратно для реализации режима в CPU I386 дополнительно включается страничный механизм, транслирующий линейный адрес в физический. Переход из RM в Page M программный: командами LMSW, SMSW, MOV CR0, MOV CR3. Обратный переход осуществляется либо перезагрузкой, либо программным путем (манипуляция битом 31 регистра CR0).

Структура микропроцессора

Упрощенно структуру микропроцессора можно представить в следующем виде:

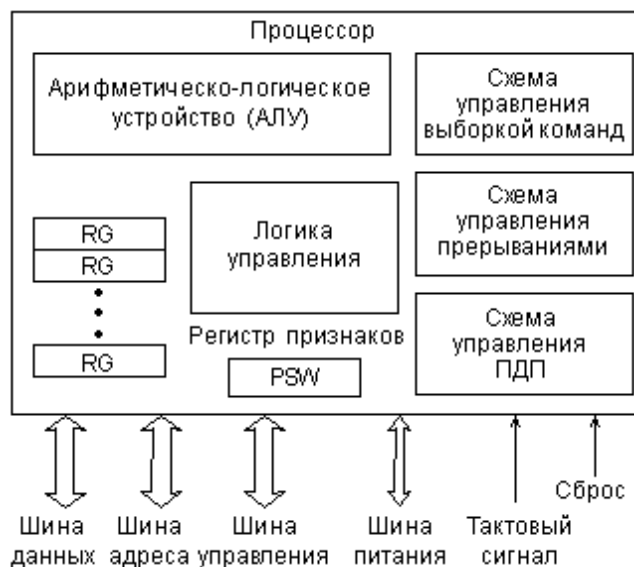


Рисунок 2 -Внутренняя структура микропроцессора.

Задания:

- 1) Составить блок-схему модуля загрузки/декодирования инструкций процессора Pentium II.
- 2) Составить блок-схему модуля диспетчеризации/исполнения инструкций процессора Pentium II.
- 3) Составить блок-схему процессора AMD-K6.
- 4) Составить блок-схему модуля декодирования процессора AMD-K6-2.
- 5) Составить блок-схему вычислительных модулей групп Register X и Register Y

процессора AMD-K6-2.

6) Выполнить работу по извлечению процессора из системной платы учебной ПЭВМ.

7) Выполнить работу по установке процессора и проверке его работоспособности.

Лабораторная работа №12 Исследование системы машинных команд микропроцессора КР580 (2 часа)

Содержание:

1. Исследование системы машинных команд микропроцессора КР580

Цель практического занятия:

изучение особенностей МП КР580 и работы эмулятора emКР580, изучение машинных команд МП КР580, составление и отладка управляющих программ по достижению целей практических занятий.

Теоретические сведения:

Процессор содержит ряд узлов, выполняющих различные функции, такие, как **арифметико-логическое устройство (АЛУ)**, используемое для осуществления операций над данными, **дешифратор команд (ДК)** и **устройство управления (УУ)**, которые анализируют команды, поступающие из программной памяти, и генерируют необходимые импульсы для выполнения этих операций, а также ряд регистров. Процессор КР580 имеет **семь регистров общего назначения**, обозначаемых А, В, С, D, E, H, L. Они предназначены для хранения как постоянных, так и переменных данных и адресов.

В структуре процессора предусмотрена реализация команд пересылки данных из одного регистра в другой, из регистра в память и наоборот, или же команд для выполнения арифметических или логических операций над содержимым 2-х регистров, при этом один из регистров обязательно должен быть регистром А (нельзя, например, суммировать содержимое регистров В и С).

Регистр А особый, называется аккумулятором, т.к. в нем аккумулируются результаты ряда арифметических операций.

Регистр команд (РК) хранит команду, которая выполняется. Он соединен с дешифратором команд, который анализирует команду и определяет вид обработки. Команда хранится и используется в течение всего времени ее выполнения.

Буферные регистры данных и адреса предназначены для временного хранения данных и адреса с целью обеспечения нормальной работы других узлов микро-ЭВМ.

Регистры счетчик команд (СК, РС), указатель стека (УС, SP) и регистр флагов (F), играют очень важную роль в программировании микропроцессора.

Все пять бит регистра признаков (F) устанавливаются по результату выполнения операции в АЛУ:

- Признак переноса (C) устанавливается в 1, если при выполнении команд появляется единица переноса из старшего разряда.

- Дополнительный признак переноса (AC) устанавливается в 1, если при выполнении команд возникает единица переноса из третьего разряда числа. Состояние разряда может быть проанализировано лишь командой десятичной коррекции.

- Признак знака (S) устанавливается в 1, если седьмой (старший) разряд числа равен 1. Машинное слово представляется числом от –128 до +127. Седьмой разряд числа указывает на знак числа. Если он равен 0, то число положительное, если 1, то отрицательное.

- Разряд признака нулевого содержимого аккумулятора (Z) устанавливается в 1, если при выполнении команды результат равен нулю.

- Разряд признака паритета (P) устанавливается в 1, если число единичных битов аккумулятора четно, в противном случае этот разряд будет установлен в нулевое состояние.

Счетчик команд (СК) содержит текущий адрес памяти, к которому обращается программа. Его содержимое автоматически изменяется в течение каждого цикла команды.

Указатель стека (УС) содержит адрес стековой памяти, начиная с которого ее можно применять для хранения и восстановления содержимого программно доступных регистров МП.

Выполнение каждой команды производится МП в строго определенной последовательности, определяемой кодом команды, и синхронизируется во времени сигналами C_1 и C_2 тактового генератора. **Период синхросигналов** C_1 или C_2 называют **машинным тактом** (МТ). Длительность машинного такта от 0,5 до 2 мкс.

Машинный цикл (МЦ) – время, требуемое для извлечения 1 байта информации из памяти или выполнения команды, определяемой одним машинным словом. Машинный цикл **может состоять из 3 – 5 машинных тактов**. **Время выполнения команды** – время получения, дешифрации и отработки команды процессором. В зависимости от вида команды, это время **может состоять из 1 – 5 машинных циклов**.

Существуют десять различных типов машинных циклов:

- 1) Извлечение кода команды.
- 2) Чтение данных из памяти.
- 3) Запись данных в память.
- 4) Извлечение из стека.
- 5) Запись данных в стек.
- 6) Ввод данных из внешнего устройства.
- 7) Запись данных во внешнее устройство.
- 8) Цикл обслуживания прерывания.
- 9) Останов.
- 10) Обслуживание прерывания при работе в режиме останова.

Первым машинным циклом при выполнении любой команды является извлечение кода команды. **На первом такте каждого машинного цикла МП указывает тип выполняемого цикла с помощью 8-разрядного слова состояния**, выдаваемого на ШД. Отдельные разряды слова состояния используются для формирования шины управления.

Способы адресации памяти

Данные (элементы информации), **участвующие в операции, определяются с помощью адресов**, указанных в командах. Существует ряд способов представления адресов в командах и их определения (вычисления) с целью доступа к операндам (данным) на основе информации, указанной в команде.

Способ адресации – правило определения (вычисления) адреса операнда на основе информации, указанной в команде, т.е. способ адресации определяет порядок выполнения действий над адресной частью команды и содержимым одного или нескольких регистров МП для вычисления исполнительного адреса $A_{исп}$, по которому хранится операнд в памяти МП (в регистре МП или ячейке ОЗУ).

Исполнительный адрес $A_{исп}$ – адрес операнда в памяти (М) или в блоке РОНов (А, В, С, D и др.). $A_{исп}$ – это целое двоичное число без знака, число разрядов n которого определяет возможное число ячеек памяти, к которым можно обращаться.

Основные способы адресации в МП КР580:

- 1) **Непосредственная адресация**. Команда с непосредственной адресацией содержит в одном из своих полей операнд. Команды с непосредственной адресацией не обращаются к памяти для извлечения операнда – они сами содержат операнд.

Примеры:

Загрузить в аккумулятор операнд 2А				
Мнемоника команды	КОП	Запись команды в памяти		
MVI A, 2A	3E	3E	2A	

- 2) **Прямая адресация**. При прямой адресации команда содержит адрес памяти, в которой

находятся данные. Такая команда занимает три байта памяти, причем второй и третий байты содержат адрес. Пример:

Загрузить содержимое ячейки 0013 в аккумулятор				
Мнемоника команды	КОП	Запись команды в памяти		
LDA 0013	3A	3A	13	00

- 3) **Косвенная (регистровая) адресация.** При косвенной адресации байт адресуется через пару регистров, т.е. адрес ячейки памяти может быть определен с помощью содержимого пары регистров. Для большинства команд используется регистры H и L. Регистр H содержит старший байт адреса, регистр L – младший. Пример:

Загрузить в регистр B содержимое ячейки памяти с адресом 1F2A, указанным в регистрах HL				
Мнемоника команды	КОП	Запись в памяти	Регистры	
			H	L
MOV B, M	46	46	1F	2A
Загрузить операнд FF в ячейку памяти с адресом, указанным в регистрах HL				
Мнемоника команды	КОП	Запись команды в памяти		
MVI M, FF	36	36	FF	
Запись данных в памяти по адресу, указанному в регистре HL				FF

- 4) Адресация через указатель стека.

Здесь адресация может осуществляться через 16-разрядный регистр – UC. Существуют две операции со стеком: запись в стек (PUSH) и выборка из стека (POP). Запись в стек используется для пересылки 16 бит данных из пары регистров или из программного счетчика (СК) в область памяти, отведенную под стек. Извлечение из стека используется для пересылки 16 бит из стековой области памяти, в любую пару регистров или в программный счетчик.

Система команд микропроцессора

Команды МП можно разделить на 4 категории:

1) **Команды пересылки данных** (см. таблицы П.3 – П.6, приведенные в **Приложении А** к практическому занятию). Эти команды **приводят к перемещению элементов информации между регистрами или между регистром и ячейкой памяти** (портом ввода/вывода). Общее число таких команд велико, т.к. имеется много видов адресации, а также большое число ограничений на форматы команд, обусловленное небольшой длиной слова процессора. Это приводит к усложнению архитектуры.

2) **Арифметические и логические команды** (см. таблицы П.7 – П.9, приведенные в **Приложении А** к практическому занятию). К ним относятся команды, объединяющие два элемента данных арифметически или логически, или выполняют операции над одним числом. Для этого используется регистр A.

3) **Команды передачи управления** (см. таблицы П.10 – П.12, приведенные в **Приложении А** к практическому занятию). Они включают в себя команды разветвления, перехода к подпрограммам и возврата из подпрограмм.

4) **Команды различного назначения или специальные команды** (см. таблицу П.13, приведенную в **Приложении А** к практическому занятию). К данному классу относятся команды, которые нельзя точно классифицировать.

Команды пересылки данных.

Команды пересылки «регистр-регистр» – MOV R1, R2. Формат этих команд имеет вид 01 DDD SSS, где DDD – регистр приемник, SSS – регистр источник. Регистры определяются номерами B=0, C=1, D=2, E=3, H=4, L=5, A=7. Таким образом, команда 4C (01001100) означает – переслать содержимое регистра H в регистр C. Если значение DDD или

SSS равно $6_{(10)}$ или $110_{(2)}$, то приемником или источником является ячейка памяти, адрес которой хранится в паре регистров HL (косвенная адресация).

Команда MVI служит для пересылки одного байта в любой из 7 РОН. Формат команды 00 DDD 110. Если DDD = 0, 1, ..., 5, 7, то пересылается константа, следующая в программе за этой командой в регистры B, C, ..., L, A. Если поле DDD = 6, то константа пересылается в ячейку памяти, адрес которой хранится в паре регистров H, L.

Для быстрой загрузки регистров H, L существует команда LXI H, пересылающая следующие за командой LXI H два байта (в программе) в указанную пару регистров. Таким же образом команда LXI D загружает два байта в пару регистров D и E, а команда (LXI B) – два байта в регистры B и C. **Эти команды называются командами непосредственной загрузки.**

Существуют **команды, выделяющие регистр A для специальных целей**. Команды STAX и LDAX служат для передачи данных между регистром A и ячейкой памяти, адрес которой находится либо в паре регистров BC, либо в паре регистров DE, причем STAX B – пересылает содержимое A в ячейку памяти, адрес которой находится в BC, а LDAX D – осуществляет обратную передачу содержимого ячейки памяти с адресом, записанным в DE в регистр A. В этих командах **доступ к ячейке памяти становится возможным** только после помещения адреса в пару регистров. Это эффективно, если неоднократно используется один и тот же адрес. При одноразовом использовании адреса используют метод абсолютной адресации.

К этой же группе команд могут быть отнесены команды ввода/вывода. **Команды ввода/вывода IN и OUT осуществляют передачу данных** между процессором и каналом ввода/вывода (портом ввода/вывода), номер которого приведен вслед за командой.

Арифметические и логические команды.

К арифметическим командам относятся: команды сложения и вычитания с учетом флага переноса (ADC, ACI, SBB и SBI) и без учета флага переноса (ADD, ADI, SUB и SUI), команды инкрементации и декрементации (INR, DCR, INX и DCX), команда десятичной коррекции аккумулятора (DAA) и команда DAD, выполняющая увеличение регистровой пары HL на величину, содержащуюся в другой регистровой паре (BC или DE).

К логическим командам относятся команды логического И, ИЛИ, исключающего ИЛИ, НЕ; команды сдвига влево и вправо с учетом и без учета переноса; команда инверсии аккумулятора (CMA); команды сравнения (CMP, CPI). Сюда же можно отнести несколько команд изменения содержимого триггера переноса в регистре флагов: STC – установить в единицу триггер переноса; CMC – инвертировать содержимое триггера переноса.

Команды передачи управления.

Команды передачи управления **делятся на три типа:** ветвления, обращения к подпрограммам и возврата из подпрограмм. Адреса команд ветвления хранятся за 1-м байтом команды.

Для команд перехода к подпрограммам характерно наличие адресов возврата, хранимых в стеках, а команды возврата из подпрограмм не имеют ассоциированных с ними адресов, они получают адреса возврата из стека.

Каждый из трех видов перехода может быть безусловным и условным в соответствии с одним из двоичных разрядов регистра признаков. Команды JMP, CALL, RET предназначены **для осуществления безусловных переходов.**

Команда RCHL приводит к тому, что содержимое регистров HL передается в программный счетчик (PC). Это переход на ячейку памяти, указанную парой HL (безусловный переход с косвенной адресацией). Команда RST – команда начального запуска прерывания программы, которая является специально командой перехода, используемой совместно с процедурами прерывания.

Есть и другие команды, относящиеся к данной группе (см. таблицы П.10 – П.12, приведенные в **Приложении А** к кратическому занятию).

Специальные команды.

К ним относятся команды EI и DI – разрешающие и, соответственно, запрещающие прерывания; NOP – пустая команда, не выполняющая никакой операции, но включаемая в программу для коррекции времени выполнения или последовательности команд и HLT – команда останова.

Все команды, необходимые для выполнения данной кратического занятия, приведены в таблице 2.

Таблица 2 – Основные команды микропроцессора KP580.

	Мнемоническое обозначение	Описание команды	2-й или 16-й код	Число ТИ
Передача, загрузка и хранение				
	MOV R1, R2	Передать содержимое одного регистра в другой.	01DDDSSS	5
	MOV M, R	Передать содержимое регистра в память.	01110SSS	7
	MOV R, M	Передать содержимое памяти в регистр.	01DDD110	7
	MVI R, D8	Загрузить регистр вторым байтом команды.	00DDD110	7
	MVI M, D8	Загрузить память вторым байтом команды.	00110110	10
	LXI B	Загрузить пару регистров BC вторым и третьими байтами команды.	00000001	10
	LXI D	Загрузить пару регистров DE вторым и третьими байтами команды.	00010001	10
	LXI H	Загрузить пару регистров HL вторым и третьим байтами команды.	00100001	10
	STAX B	Записать содержимое аккумулятора в память по адресу, указанному в паре регистров BC.	00000010	7
0	STAX D	Записать содержимое аккумулятора в память по адресу, указанному в паре регистров DE.	00010010	7
1	LDAX B	Загрузить аккумулятор содержимым ячейки, адрес которой указан в паре регистров BC.	00001010	7
2	LDAX D	Загрузить аккумулятор содержимым ячейки, адрес которой указан в паре регистров DE.	00011010	7
3	STA	Загрузить содержимое аккумулятора в память по адресу, указанному во втором и третьем байтах команды.	00110010	13
4	LDA	Загрузить аккумулятор содержимым ячейки, адрес которой указан во втором и третьем байтах команды.	00111010	13
5	SHLD	Записать в память содержимое пары регистров HL по адресу, указанному во втором и третьем байтах команды.	00100010	16
6	LHLD	Загрузить пару регистров HL содержимым ячейки, адрес которой указан во втором и третьем байтах команды.	00101010	16
7	XCHG	Поменять местами содержимое пар регистров DE и HL.	11101011	4
Операции со стеком				
8	PUSH B	Записать содержимое пары регистров BC в стек.	11000101	11

	Мнемоническое обозначение	Описание команды	2-й или 16-й код	Число ТИ
9	PUSH D	Записать содержимое пары регистров DE в стек.	11010101	11
0	PUSH H	Записать содержимое пары регистров HL в стек.	11100101	11
1	PUSH PSW	Записать содержимое аккумулятора и регистра признаков в стек.	11110101	11
2	POP B	Загрузить пару регистров BC из стека.	11000001	10
3	POP D	Загрузить пару регистров DE из стека.	11010001	10
4	POP H	Загрузить пару регистров HL из стека.	11100001	10
5	POP PSW	Загрузить аккумулятор и регистр признаков из стека.	11110001	10
6	XTHL	Поменять местами содержимое верхней ячейки стека и пары регистров HL.	11100011	18
7	SPHL	Передать содержимое пары регистров HL в указатель стека.	11111001	5
8	LXI SP	Загрузить указатель стека вторым и третьими байтами команды.	00110001	10
9	INX SP	Увеличить на 1 содержимое указателя стека.	00110011	5
0	DCX SP	Уменьшить на 1 содержимое указателя стека.	00111011	5
Переходы				
1	JMP	Безусловный переход.	11000011	10
2	JC	Условный переход по единице триггера переноса.	11011010	10
3	JNC	Условный переход по нулевому значению триггера переноса.	11010010	10
4	JZ	Условный переход по нулевому значению результата.	11001010	10
5	JNZ	Условный переход по ненулевому значению результата.	11000010	10
6	JP	Условный переход по положительному значению результата.	11110010	10
7	JM	Условный переход по отрицательному значению результата.	11111010	10
8	JPE	Условный переход по четности кода результата.	11101010	10
9	JPO	Условный переход по нечетности кода результата.	11100011	10
0	PCHL	Передать содержимое пары регистров HL в программный счетчик.	11101001	5
Вызовы				
1	CALL	Безусловный переход к подпрограмме.	11001101	17

	Мнемоническое обозначение	Описание команды	2-й или 16-й код	Число ТИ
2	CC	Переход к подпрограмме по единичному значению триггера переноса.	11011100	11/17
3	CNC	Переход к подпрограмме по нулевому значению результата.	11010100	11/17
4	CZ	Переход к подпрограмме по нулевому значению результата.	11001100	11/17
5	CNZ	Переход к подпрограмме по ненулевому значению результата.	11000100	11/17
6	CP	Переход к подпрограмме по положительному значению результата.	11110100	11/17
7	CM	Переход к подпрограмме по отрицательному значению результата.	11111100	11/17
8	CPE	Переход к подпрограмме по четности кода результата.	11101100	11/17
9	CPO	Переход к подпрограмме по нечетности кода результата.	11100100	11/17
Возвраты				
0	RET	Возврат из подпрограммы.	11001001	10
1	RC	Условный возврат из подпрограммы по единичному значению триггера переноса.	11011000	5/11
2	RNC	Условный возврат из подпрограммы по нулевому значению триггера переноса.	11010000	5/11
3	RZ	Условный возврат из подпрограммы по нулевому значению результата.	11001000	5/11
4	RNZ	Условный возврат из подпрограммы по ненулевому значению результата.	11000000	5/11
5	RP	Условный возврат из подпрограммы по положительному значению результата.	11110000	5/11
6	RM	Условный возврат из подпрограммы по отрицательному значению результата.	11111000	5/11
7	RPE	Условный возврат из подпрограммы по четности кода результата.	11101000	5/11
8	RPO	Условный возврат из подпрограммы по нечетности кода результата.	11100000	5/11
Рестарт				
9	RST	Начальный запуск прерывающей программы.	11AAA111	11
Увеличение и уменьшение				
0	INR	Увеличить содержимое регистра на единицу.	00DDD100	5
1	DCR	Уменьшить содержимое регистра на единицу.	00DDD101	5
2	INR M	Увеличить содержимое памяти на единицу.	00110100	10
3	DCR M	Уменьшить содержимое памяти на единицу.	00110101	10

	Мнемоническое обозначение	Описание команды	2-й или 16-й код	Число ТИ
4	INX B	Увеличить на единицу содержимое пары регистров BC.	00000011	5
5	INX D	Увеличить на единицу содержимое пары регистров DE.	00010011	5
6	INX H	Увеличить на единицу содержимое пары регистров HL.	00100011	5
7	DCX B	Уменьшить на единицу содержимое пары регистров BC.	00001011	5
8	DCX D	Уменьшить на единицу содержимое пары регистров DE.	00011011	5
9	DCX H	Уменьшить на единицу содержимое пары регистров HL.	00101011	5
Сложение				
0	ADD	К содержимому аккумулятора прибавить содержимое регистра.	10000SSS	4
1	ADC	К содержимому аккумулятора прибавить содержимое регистра и триггера переноса.	10001SSS	4
2	ADD M	К содержимому аккумулятора прибавить содержимое памяти.	10000110	7
3	ADC M	К содержимому аккумулятора прибавить содержимое памяти и триггера переноса.	11000110	7
4	ADI	К содержимому аккумулятора прибавить второй байт команды.	11000110	7
5	ACI	К содержимому аккумулятора прибавить второй байт команды и содержимое триггера переноса.	11001110	7
6	DAD B	К содержимому пары регистров HL прибавить содержимое пары регистров BC.	00001001	10
7	DAD D	К содержимому пары регистров HL прибавить содержимое пары регистров DE.	00011001	10
8	DAD H	К содержимому пары регистров HL прибавить содержимое пары регистров DE.	00101001	10
9	DAD SP	К содержимому пары регистров HL прибавить содержимое указателя стека.	00111001	10

Задание:

3.3. Исследование способов адресации данных в МП КР580

Задание 3.3.1. Исследовать способ непосредственной адресации данных в МП КР580 путем загрузки в регистр (РОН) операнда с помощью команды, содержащей в одном из своих полей код операнда. Для этого выполните следующие операции:

1) Составить две машинные команды MVI R, D8; которые в своем коде содержат адрес приемника информации, а во втором байте – код операнда. Первая команда должна осуществлять загрузку операнда в аккумулятор, а вторая – в один из регистров общего назначения. Варианты кодов операндов и регистров-приемников приведены в табл. 3.

Таблица 3 – Исходные данные для команды MVI R, D8

№ варианта	Код операнда D8 _h	Регистры - приемники
1	2A _h	А и В
2	7F _h	А и С

№ варианта	Код операнда D8 _h	Регистры - приемники
3	36 _h	A и D
4	D0 _h	A и E
5	B9 _h	A и H
6	79 _h	A и L
7	48 _h	A и B
8	6E _h	A и C
9	1C _h	A и D
10	5D _h	A и E
11	9A _h	A и H
12	F3 _h	A и L

2) Записать составленные команды в текстовом редакторе эмулятора.

3) Провести ассемблирование текста программы (F9) и запустить эмулятор (Ctrl+F9).

4) Записать код программы, записанный в памяти МП-системы.

5) В пошаговом режиме произвести выполнение программы (F7), убедиться в правильности выполнения команд. Записать в отчет мнемоники команд и их шестнадцатеричные и двоичные коды, коды, а также конечное состояние регистров и памяти МП.

6) Сделать выводы по результатам проделанной работы.

Задание 3.3.2. Исследовать способ прямой адресации данных в МП КР580 путем загрузки содержимое ячейки с указанным адресом в аккумулятор с помощью команды, в которой задается адрес операнда. Для этого выполните следующие операции:

1) Составить машинную команду LDA ADR16, которая загружает аккумулятор содержимым ячейки памяти, адрес которой указан во втором и третьем байтах команды. Варианты адресов ADR16 ячеек памяти приведены в табл. 4.

Таблица 4 – Исходные данные для команд LDA ADR16; MVI M, D8; LXI H, D16

№ варианта	Адрес (ADR16 _h) ячейки памяти (M)	Код операнда D8 _h
1	0010 _h	FF _h
2	0011 _h	8E _h
3	0012 _h	75 _h
4	0013 _h	37 _h
5	0014 _h	BC _h
6	0015 _h	AD _h
7	0016 _h	9C _h
8	0017 _h	C4 _h
9	0018 _h	5A _h
10	0019 _h	64 _h
11	001A _h	F3 _h
12	001B _h	0B _h

2) Для проверки выполнения команды LDA ADR16 в эмуляторе МП КР580 составить две дополнительные машинные команды MVI M, D8 и LXI H, D16. Первая команда обеспечивает запись данных в ячейку памяти с адресом, указанным регистровой парой H, L. Вторая команда осуществляет загрузку пары регистров HL вторым и третьим байтами команды, являющимися адресами ячейки памяти M. Коды операндов D8 (данных, находящихся в ячейки памяти) по вариантам представлены в табл. 4.

3) Записать составленные команды в текстовом редакторе эмулятора в следующей последовательности:

LXI H, D16h

MVI M, D8h
LDA ADR16h

3) Провести ассемблирование текста программы (F9), и запустить эмулятор (Ctrl+F9).

4) Записать код программы, записанный в памяти МП-системы.

5) В пошаговом режиме произвести выполнение программы (F7), убедиться в правильности выполнения команд. Записать в отчет мнемоники команд и их шестнадцатеричные и двоичные коды, а также конечное состояние регистров и памяти МП.

6) Сделать выводы по результатам проделанной работы.

Задание 3.3.3. Исследовать способ косвенной (регистровой) адресации данных в МП КР580 путем загрузки в регистр (РОН) содержимого ячейки памяти с адресом, указанным в паре других регистров (РОН) с помощью команды, в которой адрес ячейки памяти задается содержимым пары регистров H и L. Для этого выполните следующие операции:

1) Составить машинную команду MOV R, M; которая передает содержимое ячейки памяти по адресу, хранящемуся в регистровой паре H и L, в один из регистров (РОН). Варианты адресов ячеек памяти и регистров-приемников приведены в табл. 5.

Таблица 5 – Исходные данные для команд MOV R, M; MVI M, D8; LXI HL, D16

№ варианта	Регистр – приемник (R)	Адрес (D16 _h) ячейки памяти (M)	Код операнда D8 _h
1	B	0018 _h	9C _h
2	C	0019 _h	C4 _h
3	D	001A _h	5A _h
4	E	001B _h	64 _h
5	B	001C _h	F3 _h
6	C	001D _h	0B _h
7	D	001E _h	2A _h
8	E	001F _h	7F _h
9	B	0020 _h	36 _h
10	C	0021 _h	D0 _h
11	D	0022 _h	B9 _h
12	E	0023 _h	79 _h

2) Для проверки выполнения команды MOV R, M в эмуляторе МП КР580 составить две дополнительные машинные команды MVI M, D8 и LXI H, D16, как в задании 3.3.2. Коды операндов D8 (данных, находящихся в ячейки памяти) по вариантам представлены в табл. 5.

3) Записать составленные команды в текстовом редакторе эмулятора в следующей последовательности:

LXI H, D16h

MVI M, D8h

MOV R, M

3) Провести ассемблирование текста программы (F9), и запустить эмулятор (Ctrl+F9).

4) Записать код программы, записанный в памяти МП-системы.

5) В пошаговом режиме произвести выполнение программы (F7), убедиться в правильности выполнения команд. Записать в отчет мнемоники команд и их шестнадцатеричные и двоичные коды, а также конечное состояние регистров и памяти МП.

6) Сделать выводы по результатам проделанной работы.

Задание 3.3.4. Исследовать способ стековой адресации данных в МП КР580 путем загрузки в регистровую пару содержимого ячеек из стековой области памяти с помощью команд операций со стеком. Для этого выполните следующие операции:

1) Составить машинную команду POP RP, которая загружает операнды в регистровую

пару РОН из ячеек стека. Варианты регистровых пар-приемников данных приведены в табл. 6.

Таблица 6 – Исходные данные для команд POP RP; PUSH RP2; LXI RP2, D16

№ варианта	Регистры–приемники (PR)	Коды операндов (D16 _h)	Регистры–источники (PR2)
1	BC	C518 _h	DE
2	DE	4B19 _h	HL
3	HL	F71A _h	BC
4	BC	831B _h	HL
5	DE	361C _h	BC
6	HL	281D _h	DE
7	BC	A01E _h	DE
8	DE	641F _h	HL
9	HL	4120 _h	BC
10	BC	1721 _h	HL
11	DE	2922 _h	BC
12	HL	DC23 _h	DE

2) Для проверки выполнения команды POP RP в эмуляторе МП КР580 составить две дополнительные машинные команды LXI RP2, D16 и PUSH RP2. Первая команда осуществляет предварительную загрузку пары регистров RP2 вторым и третьим байтами команды D16. Вторая команда записывает содержимое пары регистров RP2 (регистров-источников) в стек. Варианты кодов операндов D16 (16-разрядных данных) и регистров-источников представлены в табл. 6.

3) Записать составленные команды в текстовом редакторе эмулятора в следующей последовательности:

LXI RP2, D16_h

PUSH RP2

POP RP

3) Провести ассемблирование текста программы (F9), и запустить эмулятор (Ctrl+F9).

4) Записать код программы, записанный в памяти МП-системы.

5) В пошаговом режиме произвести выполнение программы (F7), с помощью указателя стека (SP) определить и записать в отчет адрес ячеек стековой памяти, в которую записаны данные. Записать в отчет мнемоники команд и их шестнадцатеричные и двоичные коды, а также конечное состояние регистров и памяти МП.

6) Сделать выводы по результатам проделанной работы.

3.4. Исследование машинных команд пересылки данных и ввода-вывода

Задание 3.4.1. Исследовать машинные команды однобайтных пересылок данных. Для этого выполните следующие операции:

1) Записать в текстовом редакторе эмулятора МП КР580 программу на ассемблере, содержащую все команды однобайтных пересылок данных:

MVI A, FFh

MVI C, 11h

MVI E, 13h

MVI L, 15h

STAX D

MOV A, B

LDAX D

STA 0015h

LDA 0011h

2) Провести ассемблирование текста программы (F9) и, исправив ошибки, запустить эмулятор (Ctrl+F9).

3) Записать в отчет программу в машинных кодах, представленную в эмуляторе, и код программы, записанный в памяти МП-системы.

4) В пошаговом режиме произвести выполнение программы (F7), составить таблицу состояния регистров общего назначения (РОН) и ячеек памяти, указанных в программе:

Команда	A	B	C	D	E	H	L	0011	0013	0015
MVI A, FFh										
...										
LDA 0011h										

5) Определить, какие операции осуществляет каждая команда, и записать в тексте исходной программы комментарии, характеризующие каждую команду.

6) Сделать выводы по результатам проделанной работы.

Задание 3.4.2. Исследовать машинные команды двухбайтных пересылок и обмена данных. Для этого выполните следующие операции:

1) Записать в текстовом редакторе эмулятора МП КР580 программу на ассемблере, содержащую основные команды двухбайтных пересылок и обмена данных:

LXI H, 1963h

LXI D, 2012h

XCHG

LHLD 1994h

PUSH D

POP B

SPHL

2) Провести ассемблирование текста программы (F9) и, исправив ошибки, запустить эмулятор (Ctrl+F9).

3) Записать в отчет программу в машинных кодах, представленную в эмуляторе, и код программы, записанный в памяти МП-системы.

4) В пошаговом режиме произвести выполнение программы (F7), составить таблицу состояния регистров общего назначения (РОН), счетчика команд, указателя стека и ячеек памяти стека, указанных в программе:

Команда	PC	B	C	D	E	H	L	SP	FFFD	FFFE
LXI H, 1963h										
...										
SPHL										

5) Определить, какие операции осуществляет каждая команда, и записать в тексте исходной программы комментарии, характеризующие каждую команду.

6) Сделать выводы по результатам проделанной работы.

Задание 3.4.3. Исследовать машинные команды ввода-вывода данных. Для этого выполните следующие операции:

1) Записать в текстовом редакторе эмулятора МП КР580 программу на ассемблере, содержащую команды ввода-вывода данных:

MVI A, XXX ; где XXX – трехзначное десятичное число не более 255

OUT 01

IN 02

2) Провести ассемблирование текста программы (F9) и, исправив ошибки, запустить эмулятор (Ctrl+F9).

3) Записать в отчет программу в машинных кодах, представленную в эмуляторе, и код

программы, записанный в памяти МП-системы.

4) В пошаговом режиме произвести выполнение программы (F7). При этом после второго шага – команды OUT – ввести в порт с адресом 02 данные в виде трехзначного десятичного числа со значением, не превышающим 255 и отличным от ранее выводимого значения. Продолжить пошаговое выполнение программы.

5) Составить таблицу состояния аккумулятора и портов ввода-вывода, указанных в программе:

Команда	A	Порт 01	Порт 02
MVI A, XXX			
OUT 01			
IN 02			

6) Сделать выводы по результатам проделанной работы.

3.5. Исследование арифметических и логических машинных команд

Задание 3.5.1. Исследовать программу по выполнению арифметических и логических операций с одним операндом (над содержимым аккумулятора). Для этого выполните следующие операции:

1) Записать в текстовом редакторе эмулятора МП КР580 программу на ассемблере, содержащую команды арифметических и логических операций с одним операндом:

```
LHLD 0100h
MVI R, D8
MOV A, R
CMA
CALL PROC_1
INR A
CALL PROC_1
DCR A
CALL PROC_1
ADD A
CALL PROC_1
ANA A
CALL PROC_1
ORA A
CALL PROC_1
CMP A
CALL PROC_1
DAA
CALL PROC_1
HLT
PROC_1:  MOV M, A
         INX H
         MOV A, R
         RET
```

Варианты кодов данных D8 и регистра R приведены в табл. 7.

Таблица 7 – Исходные данные для команд MVI R, D8 и MOV A, R

№ варианта	Регистр R	Код операнда D8 _h
1	B	1E _h
2	C	8E _h

№ варианта	Регистр R	Код операнда D8 _h
3	D	75 _h
4	E	37 _h
5	B	BC _h
6	C	AD _h
7	D	9C _h
8	E	C4 _h
9	B	5A _h
10	C	64 _h
11	D	D3 _h
12	E	2B _h

2) Провести ассемблирование текста программы (F9) и, исправив ошибки, запустить эмулятор (Ctrl+F9).

3) Записать в отчет программу в машинных кодах, представленную в эмуляторе, с комментариями, характеризующими каждую команду

4) В пошаговом режиме произвести выполнение программы (F7), обращая особое внимание на состояние РОН и изменение регистра флагов. Составить таблицу состояния аккумулятора, регистров общего назначения (РОН), используемых в программе, регистра флагов и счетчика команд.

Команда	PC	A	R (РОН)	H	L	Z	P
LHLD 0100h							
...							
HLT							

5) Определить адреса ячеек памяти, в которых хранятся данные (результаты выполнения) программы. Занести результаты выполнения программы в таблицу 8 в 16-ричной системе счисления

Команда	Акк	CMA	INR	DCR	ADD	ANA	ORA	CMP	DAA
Результат									

6) Сделать выводы по результатам проделанной работы.

Задание 3.5.2. Исследовать программу по выполнению арифметических и логических операций с двумя операндами, один из которых содержится в аккумуляторе. Для этого выполните следующие операции:

1) Записать в текстовом редакторе эмулятора МП КР580 программу на ассемблере, содержащую команды арифметических и логических операций с двумя операндами:

```
LHLD 0100h
MVI R1, D1
MVI R2, D2
MOV A, R1
ADD R2
CALL PROC_1
SUB R2
CALL PROC_1
ANA R2
CALL PROC_1
ORA R2
CALL PROC_1
XRA R2
CALL PROC_1
CMP R2
```

```

CALL PROC_1
HLT
PROC_1:  MOV M, A
        INX H
        MOV A, R1
        RET

```

Варианты исходных кодов операндов D1 и D2, а также регистров R1 и R2 приведены в табл. 8.

Таблица 8 – Исходные данные для команд

№ варианта	Регистры R1 и R2	Код операнда D1 _h	Код операнда D2 _h
1	В и С	7C _h	9C _h
2	В и D	33 _h	C4 _h
3	В и E	2E _h	5A _h
4	С и В	D8 _h	64 _h
5	С и D	25 _h	D3 _h
6	С и E	98 _h	2B _h
7	В и С	A5 _h	1E _h
8	В и D	6F _h	8E _h
9	В и E	4C _h	75 _h
10	С и В	72 _h	37 _h
11	С и D	92 _h	BC _h
12	С и E	C3 _h	AD _h

2) Провести ассемблирование текста программы (F9) и, исправив ошибки, запустить эмулятор (Ctrl+F9).

3) Записать в отчет программу в машинных кодах, представленную в эмуляторе, с комментариями, характеризующими каждую команду

4) В пошаговом режиме произвести выполнение программы (F7), обращая особое внимание на состояние РОН и изменение регистра флагов. Составить таблицу состояния аккумулятора, регистров общего назначения (РОН), используемых в программе, регистра флагов и счетчика команд.

Команда	PC	A	R1	R2	H	L	Z	S	P
LHLD 0100h									
...									
HLT									

5) Определить адреса ячеек памяти, в которых хранятся данные (результаты выполнения) программы. Занести результаты выполнения программы в таблицу 8 в 16-ричной системе счисления

Команда	Акк	R1	R2	ADD	SUB	ANA	ORA	XRA	CMP
Результат									

6) Определить по состоянию флагов Z и S, какой операнд больше.

7) Сделать выводы по результатам проделанной работы.

Контрольные вопросы:

- 1) Порядок выполнения машинной команды микропроцессора.
- 2) Характеристика машинного цикла микропроцессора и основных типов машинных циклов.
- 3) Характеристика способов адресации памяти в МП.
- 4) Характеристика команд пересылки данных и команд ввода-вывода.
- 5) Характеристика арифметических и логических команд.
- 6) Характеристика работы команды CMP, CPI.

- 7) Характеристика команд передачи управления и специальных команд.
- 8) Порядок исследования способа непосредственной адресации данных в МП КР580 путем загрузки в регистр (РОН) операнда с помощью команды, содержащей в одном из своих полей код операнда.
- 9) Порядок исследования способа прямой адресации данных в МП КР580 путем загрузки содержимое ячейки с указанным адресом в аккумулятор с помощью команды, в которой задается адрес операнда.
- 10) Порядок исследования способа косвенной (регистровой) адресации данных в МП КР580 путем загрузки в регистр (РОН) содержимого ячейки памяти с адресом, указанным в паре других регистров (РОН) с помощью команды, в которой адрес ячейки памяти задается содержимым пары регистров Н и L.
- 11) Порядок исследования способа стековой адресации данных в МП КР580 путем загрузки в регистровую пару содержимого ячеек из стековой области памяти с помощью команд операций со стеком.
- 12) Порядок исследования машинных команд однобайтных пересылок данных.
- 13) Порядок исследования машинных команд двухбайтных пересылок и обмена данных.
- 14) Порядок исследования машинных команд ввода-вывода данных.
- 15) Порядок исследования программы по выполнению арифметических и логических операций с одним операндом (над содержимым аккумулятора).
- 16) Порядок исследования программы по выполнению арифметических и логических операций с двумя операндами, один из которых содержится в аккумуляторе.

Раздел №5 ВНЕШНИЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА ЭВМ.
Лабораторная работа №13 Принципы построения и работы накопителей на жестких магнитных дисках
(2 часа)

Содержание:

1. Принципы построения и работы накопителей на жестких магнитных дисках.

Теоретические сведения:

Накопитель на жестких магнитных дисках - (Hard Disk Drive, **HDD**, винчестер, НЖМД, жесткий диск, "винт", "хард") - это устройство, предназначенное для долговременного хранения операционных систем, программ и данных. По способу записи и чтения информации винчестеры относятся к магнитным накопителям.

Все файлы, размещенные на HDD, будут сохраняться без каких-либо потерь независимо от того, включен ПК или нет. Любые файлы могут быть скопированы, а программы проинсталлированы на HDD.

В начале 70-х годов фирмой IBM был разработан первый накопитель на жестких магнитных дисках (14-дюймовый). Диск позволял записать 30 дорожек по 30 секторов в каждой из них (30/30) и мог хранить до 16 Кбайт информации. Вначале ему присвоили название 30/30. Но по аналогии с американскими автоматическими винтовками "Winchester", имеющими калибр 30/30, дисковые устройства с несъемными дисками (жесткие диски) стали называться винчестерами.

В 1973 году фирма IBM создала первый HDD с несколькими дисками емкостью 140Мб, который продавался по цене \$8600.

Развитие HDD-технологий можно разбить на пять этапов:

- 1) Первый (до 1979 года) - использование "классических" индуктивных головок записи/воспроизведения.
- 2) Второй этап (1979-1991гг.) - применение тонкопленочных головок.
- 3) Третий (1991-1995гг.) - применение магниторезисторных (MR, Magneto-Resistive) головок.
- 4) Четвертый (1995-2000гг.) - применение супермагниторезистивных головок (GMR, Giant Magneto-Resistive): уменьшение магнитного зазора в записывающей головке и повышение чувствительности головки чтения за счет использования материалов с аномально высоким коэффициентом магниточувствительности.
- 5) Пятый (с 2000 года) - появление моделей с новым типом магнитного покрытия — с антиферромагнитной связью (AFC) при сохранении параметров магнитных головок.

Относительно корпуса ПК различают внутренние и внешние винчестеры.

Внутренние HDD - дешевле, но их максимальное количество ограничивается числом свободных отсеков корпуса, мощностью и количеством соответствующих разъемов блока питания. Установка и замена внутренних HDD требует выключения ПК. Внутренние HDD с возможностью "горячей" замены (Hot Swap) представляют собой те же винчестеры, но установленные в специальные кассеты с разъемами. Кассеты вставляются в специальные отсеки со стороны лицевой панели корпуса, конструкция позволяет вынимать и вставлять накопители при включенном питании. Для стандартных корпусов существуют недорогие приспособления (Mobile Rack), обеспечивающие оперативную съемность стандартных винчестеров.

Внешние HDD имеют собственные корпуса и блоки питания, их максимальное количество определяется возможностями интерфейса. Обслуживание внешних накопителей может производиться и при работающем ПК (критично для серверов), хотя и может требовать прекращения доступа к части дисков.

Для больших объемов хранимых данных применяются блоки внешних HDD - дисковые массивы и стойки - сложные устройства с собственными интеллектуальными

контроллерами, обеспечивающими (кроме обычных режимов работы), диагностику и тестирование своих накопителей.

Самые сложные и надежные устройства хранения состоят из множества HDD и называются RAID-массивами.

Жесткий диск имеет восемь основных параметров:

Протокол передачи данных. Есть винчестеры со следующими интерфейсами: IDE/SCSI/FC-AL/IEEE/USB. Первые винчестеры в PC XT имели интерфейс ST412/ST506; так как он ориентирован на метод записи MFM, его часто называют MFM-интерфейсом.

Среднее время доступа (Average Seek Time) - процесс позиционирования головки записи/чтения на нужное место HDD. Бывает время при чтении и время при записи. Состоит из:

времени перемещения головки с текущего трека на трек с нужным сектором (Track-to-Track Seek Time);

времени ожидания, пока диск повернется так, что нужный сектор окажется под головкой записи/чтения;

Время измеряется в миллисекундах (мс) и сегодня составляет 3,6-11,5 мс.

Скорость вращения (Spindle Speed) шпинделя - это скорость, с которой вращаются диски. Измеряется в оборотах в минуту (rpm). Она влияет:

на скорость чтения с поверхности диска. Чем быстрее диск крутится, тем больше информации считывается за единицу времени;

на время доступа к нужной информации. Информация в HDD записывается по кольцевым дорожкам, а каждая дорожка разбита на сектора. Время поиска информации определяется временем выбора нужной дорожки (не зависит от скорости вращения диска) и временем, необходимым для того, чтобы диск провернулся так, чтобы под головкой оказался нужный сектор. Чем скорость вращения выше, тем меньше это время. Скорость вращения 3600 - 15000 об/мин.

Для увеличения плотности записи зазор между поверхностью диска и головкой необходимо уменьшить до минимума. В современных винчестерах эта задача решается с использованием аэродинамической подъемной силы, создаваемой потоком воздуха, который увлекает за собой вращающаяся рабочая поверхность диска. Для возникновения подъемной силы рабочим поверхностям головок придают специальную форму в виде крыла. Для того чтобы головка не «улетала» далеко от поверхности диска, она закрепляется на пружинящем поводке.

Поскольку величина подъемной силы определяется плотностью воздуха, которая зависит от атмосферного давления, то винчестеры общего применения имеют ограничения по максимальной высоте подъема над уровнем моря (приблизительно до 2000...3000 м).

В современных накопителях скорость вращения пакета дисков может достигать 15 000 об/мин. Однако высокие скорости вращения порождают проблемы, связанные с его балансировкой, гироскопическим эффектом и аэродинамикой головок. Во время работы головки ни в коем случае не должны механически соприкасаться с рабочими поверхностями

случайное касание поверхности практически всегда приводит к полному или частичному повреждению соответствующей дорожки рабочей поверхности и очень часто к обрыву самой головки.

Объем. Измеряется в гигабайтах (Gb). На самих HDD раньше писали емкость в миллионах байт и указывалась неформатированная емкость (реальная - на 10-15% меньше). Бывает, что Bios'ы выдают емкость не в Gb, а Mb или даже в млн.байт. Сегодня в продаже - HDD емкостью 10-80Gb, максимальный объем дисков постоянно растет и пока равен 320Gb (MaXLine от Maxtor).

Плотность записи. Измеряется в гигабайтах на пластину. Внутри HDD находится один или несколько дисков. Она влияет:

на скорость: чем больше плотность записи, тем больше информации помещается на

одну дорожку, и, соответственно, больше скорость считывания этой информации при одинаковой скорости вращения диска;

на охлаждение: меньшее число пластин уменьшает тепловыделение (диск меньше греется).

До 5Gb на пластину - старые винчестеры; HDD в продаже - с плотностью записи 10Gb/пл.(устар.), 15Gb/пл., 20Gb/пл., 30Gb/пл.; максимальная плотность пока равна 125Gb/пл. (винчестеры от Seagate Technology).

В большой степени максимальная плотность записи зависит от конструкции и характеристик головок записи/чтения. Раньше в винчестерах использовались магнитные головки, представляющие собой миниатюрные катушки индуктивности, намотанные на магнитный сердечник.

Позднее стали использовать тонкопленочные магнитные головки, а в современных винчестерах используются высокочувствительные магниторезистивные головки (MRH - Magneto-Resistive Heads) чтения (представляет собой резистор, сопротивление которого изменяется в зависимости от напряженности магнитного поля, причем амплитуда уже практически не зависит от скорости изменения поля. Это позволяет намного более надежно считывать информацию и диска и, как следствие, значительно повысить предельную плотность записи. MR-головки используются только для считывания; запись по-прежнему выполняется индуктивными головками), конструктивно объединенные с тонкопленочными головками записи. Головки собираются в блок.

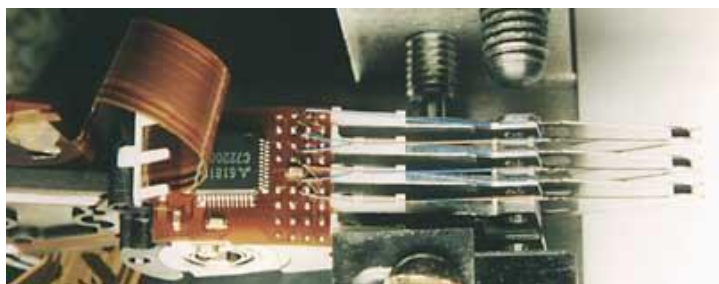


Рисунок 1 - Внешний вид блока головок

В современных винчестерах используется система позиционирования блока головок с поворотной подвижной катушкой, помещенной в зазоре мощного постоянного магнита, которая и является исполнительным элементом системы позиционирования.

В основе этой системы лежит предварительная (произведенная при изготовлении винчестера) запись специальных цифровых последовательностей, которые называются сервометками, в специально отведенные для этого на каждой дорожке сектора. Во время работы контроллер винчестера ориентируется на эти сервометки, вырабатывая управляющие сигналы, подаваемые в подвижную катушку, и поворачивает головку таким образом, чтобы она установилась точно над дорожкой, а затем удерживает ее на этой дорожке до поступления команды о переводе головки в новое положение.

Объем кэша (мульти сегментного - Multisegmented Cache, Cache memory). Учитывается, что следующей командой винчестеру потребуется считать данные из секторов, следующих за текущим. Поэтому идет чтение данных из оставшихся секторов на треке, которые записываются во внутреннюю память дисководов ("кэш"). Измеряется в килобайтах (мегабайтах). Ранее Cache составлял 128-1.024 Кбайт, сегодня 2-8 Мбайт. Некоторые производители (напр. Quantum), используют часть кэша под свое программное обеспечение. У других (напр. Western Digital) для хранения firmware используются специально отведенные сектора на диске, невидимые для любых операционных систем. По включению питания эта программа загружается в обычную память.

Потоковая скорость передачи данных (Sustained Transfer Rate). При размере

считываемой информации во много раз превышающих размер выделенного для него кэша идет непрерывное Cache-считывание секторов. Измеряется в мегабайтах в секунду и сегодня составляет до 80Мбайт/с.

MTBF (Mean Time Between Failures, среднее время наработки на отказ). Это надежность винчестера и измеряется в часах работы. MTBF - величина усредненная (по партии), но MBFT=900т.час лучше MBFT=300т.час. Современные HDD имеют от 500,000 до 1,000,000 часов. Т.е. это 20-40 лет (при 8-часовой работе). Лидер пока SCSI-винчестер Cheetah с 1,200,000 часов.

Кроме основных параметров, важны:

Перегрузка от удара в рабочем/нерабочем состоянии (Operating/Nonoperating Shock), G - параметр, характеризующий устойчивость винчестера к механическим воздействиям.

Рабочая температура (Operating temperature), °C - параметр, по которому можно судить о "жаростойкости" винчестера.

Потребляемая мощность (Power Management), Вт - параметр, о том, насколько винчестер будет нагреваться.

Срок гарантии - от 6 месяцев до 5 лет.

Фирма-производитель.

Разделы HDD и подключение дисков. Винчестер "физический" можно разделить на несколько "логических" (т.е. содержит logical область). Вы можете создать любую конфигурацию разделов.

Разделы могут быть четырех типов:

Master Boot Record (главная загрузочная запись, MBR). Здесь (в первом блоке HDD) хранится информация о разбиении диска и там можно разместить Boot Manager;

Primary (основной). Это раздел, в котором всегда устанавливается операционная система. Многие "простые" OS (напр. DOS, Windows) инсталлируются только в primary;

Extended (расширенный). Это раздел для программ пользователя, к которому OS имеет доступ. Extended может использоваться целиком (как единый логический диск) или же разбит на несколько логических дисков (см. рис);

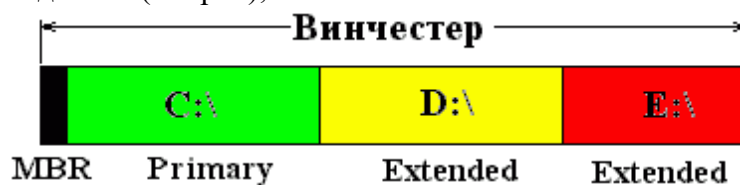


Рисунок 2 – Разделы HDD

Other (иной) раздел. Это Extended-раздел, выделяемое для установки другой OS, отличной от установленной в primary;

При разделении HDD всегда получается только один primary-раздел и один или несколько extended-разделов. Размер любого HDD-раздела имеет верхний предел. Максимальное число логических дисков для файловых систем FATxx составляет 26.

Из-за более низкой стоимости IDE-дисков (по сравнению с SCSI-дисками и учитывая неразвитость USB-дисков) фактически они и преобладают. На одном IDE-кабеле могут работать до двух IDE-устройств: Master (MA) - основной (первый) и Slave (SL) - дополнительный (второй).

Установка Master/Slave производится HDD-Jumper-ами. Если устройство на кабеле одно, его устанавливают в режим Master, однако у некоторых HDD есть отдельный режим Single. Напрямую не допускается работа устройства в режиме Slave при отсутствии Master-устройства, но некоторые новые модели HDD могут так работать в этом режиме при условии соответствующего Bios-а или драйвера. Это необходимо, т.к. многие драйверы, обнаружив отсутствие Master-устройства, прекращают дальнейшую работу с данным контроллером.

Существует режим, в котором HDD само устанавливается в режим Master/Slave в

зависимости от типа разъема на интерфейсном кабеле - Cable Select (CS, CSel, выбор по разъему кабеля).

Есть вариант и увеличения максимального количества подключенных устройств IDE-устройств (стандарт - не более 4-х штук). Для этого нужен свободный PCI-слот m/b. По внешнему виду это карта с двумя (или четырьмя) IDE-контроллерами, установленная в PCI-разъем материнской платы. Для активации контроллера необходима настройка Bios-а карты.

Анатомия винчестера

HDD состоит из гермоблока и платы электроники. В гермоблоке размещены ("запечатаны") все механические части и предусилитель, на плате - почти вся управляющая электроника.



Рисунок 3 - Гермоблок

Внутренний объем гермоблока не изолирован от внешней среды – обязательно предусматривается возможность перетока воздуха снаружи в камеру и наоборот. Это необходимо для выравнивания давления внутри камеры с внешним давлением для предотвращения деформаций корпуса.



Рисунок 4 - Плата электроники

Достигается это с помощью так называемого барометрического фильтра, размещаемого внутри камеры. Фильтр способен задерживать частицы, размер которых превышает величину рабочего зазора между головкой и поверхностью диска. Так что под герметичностью камеры подразумевается только невозможность проникновения внутрь посторонних частиц, способных при попадании в рабочий зазор повредить рабочую поверхность и головку.

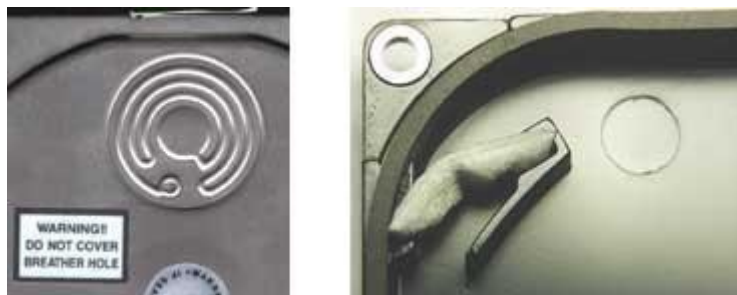


Рисунок 5 - Барометрический фильтр и фильтр рециркуляции

Кроме того, в камере винчестера обязательно размещается фильтр рециркуляции, предназначенный для улавливания частиц, которые могут возникать внутри самой камеры, например за счет осыпания поверхности дисков при «взлетах» и «посадках» головок в зоне парковки, или проникать внутрь камеры через барометрический фильтр. Место расположения фильтра рециркуляции выбирается с учетом движения воздушного потока и возможных траекторий движения частиц таким образом, чтобы обеспечить максимальную степень очистки воздушного потока внутри камеры.

Так как попадание посторонних частиц внутрь камеры абсолютно недопустимо, сборка винчестеров производится только в специальных «чистых помещениях», оборудованных дорогостоящими фильтровентиляционными установками тонкой очистки воздуха – в кубическом футе воздуха может быть не более 100 частиц (пылинок) размером более 0,5 мкм.

Для вращения дисков применяется специальный электродвигатель (шпиндель), чем-то похожий на двигатель флоппи-дисковода: неподвижный якорь с обмотками и вращающийся постоянный магнит. Основное отличие его состоит в более высокой точности изготовления и наличии специальных подшипников, которые могут быть как обычными шариковыми, так и более совершенными - жидкостными (вместо шариков в них используется специальное масло, поглощающее ударные нагрузки, что увеличивает долговечность двигателя). Жидкостные подшипники имеют более низкий уровень шума и почти не выделяют тепло во время работы. Кроме того, некоторые современные винчестеры имеют двигатель, целиком погруженный в герметичный сосуд с маслом, что способствует эффективному отводу тепла от обмоток.

Магнитные диски представляет собой пластины из алюминия, керамики или стекла, поверхность которых обработана по высочайшему классу точности. На диски нанесен тонкий слой высококачественного ферромагнетика - на основе окиси хрома (ранее - окиси железа и бариевых ферритов). Количество дисков - одного до трех (как правило), но в некоторых моделях достигает 10. Под дисками находится двигатель, вращающий их.

Ближе к разъемам, с левой или правой стороны от шпинделя, находится поворотный позиционер (head positioner) - с одной стороны - несущие магнитных головок, а с другой - короткий и более массивный хвостовик с обмоткой электромагнитного привода. Магнитная головка также представляет собой сложную конструкцию, состоящую из десятков деталей. Эти детали настолько малы, что изготавливаются методом фотолитографии так же, как и современные микросхемы. Рабочая поверхность керамического корпуса головки

отполирована с такой же высокой точностью, как и диск. Привод головок (позиционер) представляет собой плоскую катушку-соленоид из медной проволоки, помещенную между полюсами постоянного магнита и закрепленную на конце рычага, вращающегося на подшипнике. Встречаются поворотные и линейные позиционеры.

При поворотном приводе головки перемещаются по дуге, как в обычном электропроигрывателе. Такие приводы обеспечивают меньшую инертность и более быстрое позиционирование головок.

Линейный привод обеспечивает перемещение головок по радиусу диска. Преимущество линейного привода заключается в том, что зазор магнитной головки всегда перпендикулярен дорожке и расстояние между дорожками сохраняется постоянным.

Данные с поверхности диска считываются непосредственно магнитной головкой. При записи головка создает магнитное поле, намагничивая тем самым участок диска - при считывании же, наоборот, поле диска возбуждает сигнал в головке. Современные приводы содержат несколько магнитных головок - как правило, по одной на каждую сторону каждого диска.

При вращении дисков внутри корпуса возникает воздушный поток, который и приподнимает головки над поверхностью - головки "посадили" на воздушную подушку. Зазор между головкой и поверхностью диска в современных винчестерах составляет 0,1 мкм, что в 500 раз меньше толщины человеческого волоса.

Но эта конструкция требует парковки головок - перемещения их за пределы рабочей области диска (landing zone) во время выключения компьютера. В современных винчестерах операция вывода головок в зону парковки (находится вблизи центра дисков, головки прижаты к сторонам пластин легкими пружинами) выполняется автоматически при снижении скорости вращения двигателя ниже номинальной или при пропадании напряжения питания, а вывод головок в рабочую зону разрешается только после достижения номинальной скорости вращения дисков. Управляет всем этим позиционер.

В нерабочем состоянии накопителя головки находятся вблизи центра дисков, в "зоне парковки" и прижаты к сторонам пластин легкими пружинами. Но стоит дискам начать вращение - и поток воздуха приподнимает головки над поверхностью блинов, преодолевая усилие пружин. Головки "всплывают" и с этого момента находятся над диском, совершенно не касаясь его.

Плата электроники - съемная, подключается к гермоблоку через один-два разъема различной конструкции. В состав контроллера входят следующие функциональные узлы: схема управления двигателем, схема управления позиционированием головок, канал чтения-записи, цифровой сигнальный процессор, микропроцессор управления, буфер памяти накопителя и интерфейсная логика.

Микропроцессор управления представляет собой очень быструю специализированную микро-ЭВМ, имеющую свою оперативную память, постоянную энергонезависимую память и программное обеспечение, состоящее из нескольких модулей. Оно образует специализированную операционную систему. Некоторые из ее компонентов могут располагаться в микросхеме ПЗУ на плате электроники, а другие записаны непосредственно на диск в служебной области.

При включении питания первым "просыпается" микропроцессор управления и тестирует электронику винчестера. Если все в порядке, подается команда на включение электродвигателя. Это самый сложный момент в работе накопителя, так как при этом головки соприкасаются с диском в зоне парковки и изнашиваются. Кроме того, двигатель в момент разгона работает в форсированном режиме, что сопровождается большим потреблением тока и повышенной нагрузкой на электронику.

После "всплытия" головок осуществляется процесс их распарковки: в обмотку соленоида подается импульс тока, перемещающий головки в информационную зону дисков. Начинается поиск сервометок, которые, в данном случае, используются для определения

скорости вращения. Убедившись в том, что блины крутятся с нужной скоростью, микропроцессор перемещает головки в зону, где записана служебная информация, и считывает с диска в свою память микропрограмму, которая анализирует конфигурационные параметры и таблицу дефектных секторов.

Затем выполняется еще несколько внутренних тестов, термокалибровка, чтение таблицы S.M.A.R.T.-параметров (о которой мы поговорим чуть ниже), тестирование правильности позиционирования головок путем чтения нескольких дорожек, расположенных в разных местах диска.

В процессе работы винчестера через обмотки двигателя и катушку позиционирования текут очень большие импульсные токи, поэтому процессор управляет ими не напрямую, а через микросхемы усилителей тока. В современных накопителях данные читаются и записываются на диск не в цифровой, а в аналоговой форме методом частотной модуляции, поэтому в канале чтения-записи применяется цифровой сигнальный процессор, включающий в себя АЦП и ЦАП, преобразующие прочитанные головками ультразвуковые сигналы в "цифру" и наоборот.

Микропроцессор винчестера функционирует все время, пока на него подано питание. Под его контролем все системы накопителя работают дружно, образуя несколько замкнутых систем авторегулирования, поддерживающих постоянную скорость вращения дисков и обеспечивающих точное попадание головок на дорожки и доступ к любому сектору независимо от физического износа механики и внешних ударных или тепловых воздействий. Для питания накопителей настольных компьютеров обычно используется два напряжения: +5 В (для схем усиления и обработки сигналов) и +12 В (для силовых цепей). Это позволяет уменьшить помехи от двигателей и упростить схему. Обычно требования к стабильности источника +5 В гораздо выше, чем к +12 В. Это справедливо и для многих

других устройств.

При внезапном выключении питания, электроника винчестера продолжает некоторое время работать, получая энергию от двигателя, который, вращаясь по инерции, вырабатывает электрический ток, достаточный для успешной парковки головок. Некоторые накопители умеют сами скрывать вновь образующиеся дефекты поверхности.

В связи с тем, что длина внешних и внутренних дорожек на дисках отличается, то и секторов на них помещается разное количество. Когда винчестер работает, его микропроцессор производит пересчет физических параметров о числе головок, числе секторов в дорожке и количестве цилиндров в вид, воспринимаемый внешними устройствами. Этот параметр называется геометрией накопителя, а процесс пересчета - трансляцией, и служит для удобства работы внешних устройств.

Именно эти "стандартные" параметры и указываются на корпусе винчестера (например, 16 головок и 63 сектора в дорожке, хотя на самом деле головок может быть всего 2). Емкость винчестера в байтах можно подсчитать, перемножив число головок, число секторов в дорожке, число цилиндров и размер сектора, который равен 512 байт. Например, $16 \cdot 63 \cdot 39714 \cdot 512 = 20,4$ Гб (конечно, имеются в виду "неправильные" гигабайты производителей - по миллиарду байт).

После успешного завершения всех внутренних тестов микропроцессор производит разблокировку интерфейса, сообщая об этом материнской плате компьютера.

На одних винчестерах программа процессора полностью хранится в ПЗУ, на других определенная ее часть записана в служебной области диска. На диске также записаны параметры HDD (производитель, модель, серийный номер и т.п.). Некоторые винчестеры хранят эту информацию в электрически репрограммируемом ПЗУ (EEPROM).

Многие винчестеры имеют на плате электроники специальный технологический интерфейс с разъемом, через который при помощи стендового оборудования можно выполнять различные сервисные операции с накопителем - тестирование, форматирование, переназначение дефектных участков и т.п.

Размещение информации на диске

Всю информацию, хранящуюся на диске, условно делят на служебную и пользовательскую. Первая обеспечивает нормальную работу и изначально присутствует в любом HDD - ее записывает завод-изготовитель.

Поверхность диска никогда не используется для записи произвольным образом. Данные всегда записываются в виде концентрических окружностей, называемых дорожками, состоящих из нескольких меньших отрезков - секторов. Каждой дорожке и каждому сектору на каждой из сторон диска присваивается свой порядковый номер. Расположенные одна над другой несколько дорожек с одинаковыми номерами называются цилиндрами.

Такое деление дискового пространства на участки называется форматом нижнего уровня и выполняется на заводе - изготовителе винчестера.

В процессе низкоуровневого форматирования дисков может выясниться, что на поверхности пластин имеется один или несколько маленьких участков, чтение или запись в которые сопровождается ошибками (так называемые сбойные секторы, или бэд-блоки).

Однако из-за этого диск не выбрасывают и не считают его испорченным, а всего лишь помечают эти секторы особым образом, и они в дальнейшем игнорируются. Чтобы пользователь не видел этого безобразия, винчестер содержит некоторое количество запасных дорожек, которыми электроника накопителя "на лету" подменяет дефектные участки поверхности, делая их абсолютно прозрачными для операционной системы и таких программ, как, например, дисковые редакторы, ScanDisk и Norton Disk Doctor.

Но не вся область диска отведена для записи данных. Часть информационной поверхности используется накопителем для собственных нужд. Это область служебной, как ее еще иногда называют, инженерной информации. Она скрыта от пользователей и становится доступной при переводе винчестера в специальный технологический режим, осуществляемый при помощи стендового оборудования и особых утилит

Служебная информация.

Служебную информацию можно разделить на несколько типов:

- сервометки, предназначенные для стабилизации скорости вращения дисков, поиска секторов и точной установки головок на дорожки;

- информация, служащая для адресации секторов с данными пользователя и контроля целостности этих данных;

- рабочие программы (микрокод), предназначенные для управления работой всех систем накопителя;

- паспорт винчестера, в котором записана информация о количестве дисков, головок, название фирмы-производителя и модели накопителя, дата его изготовления, страна изготовитель, номер конвейера, номер рабочей смены и многое другое; здесь же хранится и уникальный серийный номер винчестера;

- таблица дефектных секторов, служащая для аппаратной подмены сбойных участков поверхности из резерва. Эта информация используется электроникой винчестера в процессе работы и является важнейшей его частью, без которой физически полностью исправный накопитель был бы бесполезным куском железа.

Каждый HDD разделен на зоны (notches), в каждую из которых входит обычно от 20 до 30 цилиндров с одинаковым количеством секторов. Секторов может уместиться от 17 до 150 (как правило) на одной дорожке. Их нумерация начинается с 1, тогда как нумерация головок и цилиндров начинается с 0. Количество секторов на дорожке не равное. Чем дальше дорожка от центра, тем больше число секторов на диске.

Задания:

- 1) Составить структурную схему накопителя на жестких магнитных дисках.
- 2) Составить блок-схему блока электроники накопителя на жестких магнитных дисках.

- 3) Произвести расшифровку типа и характеристик накопителя на жестких магнитных дисках по указанной маркировке.
- 4) Определите тип интерфейса подключенного к накопителю на жестких магнитных дисках в учебном системном блоке ПЭВМ.
- 5) Выполнить работу по извлечению накопителя на жестких магнитных дисках из учебного системного блока ПЭВМ.
- 6) Выполнить работу по установке накопителя на жестких магнитных дисках на прежнее место в учебном системном блоке ПЭВМ и проверке его работоспособности.
- 7) Выполнить работу по замене шлейфов подключения накопителя на жестких магнитных дисках к системной плате и блоку питания, проверке работоспособности накопителя.
- 8) Выполнить работу по подключению двух накопителей на жестких магнитных дисках одновременно, произвести коммутацию перемычек на системной плате для установки первого накопителя в режим «Master», а второго – в режим «Slave».
- 9) Выполнить работу по установке и подключению второго накопителя на жестких магнитных дисках вместо дисководов, и произвести проверку его работоспособности.
- 10) Выполнить работу по установке и подключению второго накопителя на жестких магнитных дисках вместо CD/DVD привода, и произвести проверку его работоспособности.

Лабораторная работа №14 Исследование программ и утилит для тестирования и диагностирования внешних запоминающих устройств (2 часа)

Содержание:

1. Исследование программ и утилит для тестирования и диагностирования внешних запоминающих устройств.

Цель практического занятия:

целью данного практикума является выработка умения оценивать характеристики внешних запоминающих устройств ПЭВМ с помощью тестовой программы Victoria и утилиты HD Tune для тестирования накопителей. В результате выполнения практического занятия студенты также выработают навыки применения внешних запоминающих устройств ПЭВМ при решении задач получения, хранения, обработки и передачи информации.

Теоретические сведения:

Программа Victoria для профессионального тестирования HDD

Программа ориентирована на широкий круг пользователей ПК, и **предназначена** для тестирования, сервисного обслуживания и помощи при восстановлении информации с любых жестких дисков с интерфейсами IDE и Serial ATA. **Программа представляет собой** полностью готовое решение для всесторонней, глубокой, и в тоже время максимально быстрой оценки реального технического состояния HDD. В ней собраны возможности большинства фирменных диагностических утилит для HDD, и другие полезные функции, имеющиеся в ATA стандарте на современные жесткие диски, однако нет ограничений на поддерживаемые модели.

Область применения программы – обнаружение неисправности жесткого диска. Нередко пользователь остается в неведении о состоянии жесткого диска, вплоть до серьезной его аварии, когда уже ничего нельзя сделать (информация сильно испорчена, или для ее восстановления требуются серьезные материальные затраты). Данная программа позволяет протестировать винчестер пользователям без специальных знаний. Другая возможная область применения – измерение скоростных характеристик накопителей. Большинство бенчмарк-программ написаны под многозадачные ОС, и работают с диском через цепочку API+дисковый драйвер, что требует инициализации накопителя в системе, и наличия включенного режима UDMA. Виктория работает с винчестером напрямую, и кроме всего,

позволяет интерактивно управлять тестированием.

Следует отметить, что Виктория показывает только физическое состояние диска. Дефекты файловых систем останутся незамеченными - для их обнаружения существуют системные утилиты (например, SCANDISK, NDD). В тоже время, если на ПК периодически пропадают или портятся данные, а на его винчестере обнаруживаются явные физические проблемы - разумно предположить, что именно они и являются причиной сбоев.

Все операции с диском реализованы через порты контроллера, поэтому программе не требуется поддержка испытываемых винчестеров материнской платой и определение их в BIOS. Также, программе не важно, есть ли на диске какая либо файловая система, какой размер кластера, и сколько на HDD логических разделов – она показывает диск в таком виде, как это задумали разработчики HDD (в соответствии со спецификацией ATA/ATAPI7), на весь физический объем, независимо от Вашего компьютера, BIOS'a и ОС.

Основные возможности программы:

- 1) **Чтение паспорта диска и вывод на экран** полной технической информации о накопителе;
- 2) Определение установленных в системе ATA/SATA контроллеров (включая дополнительные);
- 3) Управление уровнем акустического шума (AAM);
- 4) **Просмотр S.M.A.R.T. параметров накопителя**, быстрая оценка его состояния по псевдографическим шкалам и по регистру статуса;
- 5) **Работа с Host Protected Area**: изменение и восстановление физического объема диска;
- 6) **5 режимов тестирования поверхности**: верификацией, чтением и записью, с подсчетом и отображением адресов дефектных блоков;
- 7) **2 режима построения графика поверхности**: полный и оценочный (аналогично тому, как сделано в программе HD Tach);
- 8) **Дефектоскоп**: анализ состояния поверхности 3-мя видами тестов, с подсчетом и отображением нестабильных участков, с указанием точных адресов каждого нестабильного сектора и автоматическим занесением их в текстовый файл;
- 9) **Тестирование буферной памяти и интерфейса** на наличие «глюков» и искажения информации при приеме и передаче;
- 10) **Измерение частоты вращения вала HDD**, в том числе на новых дисках без поля INDEX;
- 11) **Скрытие дефектов поверхности** методом переназначения секторов из резерва (гемар) на любом из 3-х тестов;
- 12) Измерение производительности жесткого диска (бенчмарк функции):
 - **Измерение скорости** линейного, нелинейного и случайного **чтения с HDD**;
 - **Измерение скорости позиционирования** головок HDD и времени доступа к секторам;
 - Измерение скорости чтения графическими методами;
- 13) Очистка диска (или его части) от информации – **низкоуровневое форматирование**;
- 14) **Управление опциями безопасности**: установка пароля на HDD, снятие пароля, быстрое стирание информации без возможности её восстановления и т.п.
- 15) Возможность остановки и запуска шпиндельного двигателя HDD;
- 16) **Тест позиционирования головок HDD** (аналогично тому, как это делает ОС при интенсивной работе), с целью выявления надежности и термоустойчивости дисковой подсистемы ПК (приводит к разогреву HDD);
- 17) **Посекторное копирование произвольной области HDD в файл**, с пропуском дефектных участков (может быть полезно для спасения информации с поврежденного диска);
- 18) Посекторное копирование файлов на HDD;
- 19) **Просмотр информации о логических разделах HDD** с указанием границ разделов (без определения HDD в BIOS);

- 20) **Индикация режимов работы HDD**, содержимого регистров, и визуализация кодов ошибок по индикаторным лампочкам;
- 21) Встроенная контекстно-зависимая система помощи.

1.3. Утилита HD Tune для тестирования накопителей

HD Tune Pro – **утилита для планомерного детального тестирования** текущей работоспособности и производительности жестких дисков с различными интерфейсами (SCSI, SATA или IDE), а так же карт памяти и flash-дисков. Располагаясь в трее (области уведомления рабочего стола), запущенная **утилита показывает** текущую температуру, проверяет скорость передачи данных, время доступа, уровень загрузки процессора, выдает информацию о версии firmware, серийном номере и т.д.

HD Tune Pro производит замер и наглядно отображает такие параметры накопителей, как скорость передачи данных, время чтения, записи и время доступа, уровень загрузки процессора, температуру винчестера, показывает в виде сводной таблицы данные поставляемые S.M.A.R.T, позволяет сканировать жесткие диски и другие накопители на предмет наличия ошибок, выдает информацию о версии firmware, серийном номере, объеме диска и подробности по потребителям полезной площади, его КЭШа, возможном и активном режимах передачи данных и т.д. и т.п.

Функциональные возможности утилиты:

- измерение быстродействия и производительности;
- отображение информации о разделах;
- отображение поддерживаемых функций;
- представление информации о версии прошивки, серийного номера, объем накопителя, размер буфера, режим передачи данных;
- отображение в виде сводной таблицы S.M.A.R.T данных;
- сканирование на наличие ошибок;
- вывод информации о температуре накопителей;
- копирование информации в виде текста в буфер обмена;
- копирование в буфер обмена скриншотов с информацией.

Задание:

3.3. Провести тестирование и диагностирование внешних запоминающих устройств ПЭВМ с помощью программы Victoria

Задание 3.3.1. Произвести чтение паспортов внешних запоминающих устройств ПЭВМ с помощью программы Victoria. Для этого выполните следующие операции:

- 1) Подключить флэш-накопитель к USB порту компьютера.
- 2) Запустить программу с помощью файла vcr446f.exe.
- 3) Открыть окно «Standart» программы и установить режим API (интерфейса прикладного программирования).
- 4) В правом верхнем окне программы курсором выбрать жесткий диск и нажать кнопку «Passport».
- 5) Занести в отчет паспорт жесткого диска, т.е. информацию, приведенную в окне «Drive passport» (Паспорт устройства).
- 6) Расшифровать с помощью информации, представленной в Приложении А, данные приведенные в паспорте жесткого диска.
- 7) В правом верхнем окне программы курсором выбрать флэш-накопитель и нажать кнопку «Passport».
- 8) Занести в отчет паспорт флэш-накопителя, т.е. информацию, приведенную в окне «Drive passport» (Паспорт устройства).
- 9) Расшифровать с помощью информации, представленной в Приложении А, данные приведенные в паспорте флэш-накопителя.

10) В правом верхнем окне программы курсором выбрать CD(DVD)-привод и нажать кнопку «Passport».

11) Занести в отчет паспорт CD(DVD)-привода, т.е. информацию, приведенную в окне «Drive passport» (Паспорт устройства).

12) Расшифровать с помощью информации, представленной в Приложении А, данные приведенные в паспорте CD(DVD)-привода.

13) Сделать выводы по результатам проделанной работы.

Задание 3.3.2. Провести тест на дефектоскопию поверхности жесткого диска с помощью программы Victoria. Для этого выполните следующие операции:

1) Открыть окно «Standart» программы и убедиться в том, что установлен режим API (интерфейса прикладного программирования).

2) В правом верхнем окне программы курсором выбрать жесткий диск и нажать кнопку «Test».

3) В открытом тестовом окне программы установить опции: read и Ignore, после чего нажать кнопку «Start».

4) Занести результаты тестирования жесткого диска в отчет.

5) Сделать выводы по результатам проделанной работы.

Задание 3.3.3. Провести тест на дефектоскопию для флеш-накопителя и CD(DVD)-диска с помощью программы Victoria. Для этого выполните следующие операции:

1) Подключить флэш-накопитель к USB порту компьютера и вставить CD(DVD)-диск в привод.

2) Запустить программу, открыть окно «Standart» программы и убедиться в том, что установлен режим API.

3) В правом верхнем окне программы курсором выбрать флеш-накопитель и нажать кнопку «Test».

4) В открытом тестовом окне программы установить опции: read и Ignore, после чего нажать кнопку «Start».

5) Занести результаты тестирования флеш-памяти в отчет.

6) В правом верхнем окне «Standart» программы курсором выбрать CD(DVD)-диск и нажать кнопку «Test».

7) В открытом тестовом окне программы установить опции: read и Ignore, после чего нажать кнопку «Start».

8) Занести результаты тестирования CD(DVD)-диска в отчет.

9) Сделать выводы по результатам проделанной работы.

Задание 3.3.4. Продиагностировать состояние жесткого диска ПЭВМ с помощью технологии S.M.A.R.T. Для этого выполните следующие операции:

1) Открыть окно «Standart» программы и убедиться в том, что установлен режим API.

2) В правом верхнем окне программы курсором выбрать жесткий диск и нажать кнопку «SMART».

3) В открытом технологическом окне программы нажать кнопку «Get SMART».

4) Занести показания SMART жесткого диска в отчет, расшифровав их с помощью информации, представленной в Приложении Б.

5) Сделать выводы по результатам проделанной работы.

Задание 3.3.5. Определить среднее время позиционирования головок (доступа к данным на диске). Для этого выполните следующие операции:

1) Открыть окно «Standart» программы и убедиться в том, что установлен режим API.

2) В правом верхнем окне программы курсором выбрать жесткий диск.

3) В открытом окне программы нажать кнопку «Seek».

4) Занести в отчет результаты тестирования в виде среднего времени позиционирования головок (доступа к данным на диске).

5) Сделать выводы по результатам проделанной работы.

3.4. Провести тестирование текущей работоспособности и производительности жесткого диска и флэш-накопителя с помощью утилиты HD Tune

Задание 3.4.1. Получить подробную информацию о жестком диске ПЭВМ и флэш-памяти с помощью утилиты HD Tune. Для этого выполните следующие операции:

- 1) Подключить флэш-накопитель к USB порту компьютера.
- 2) Запустить утилиту HD Tune с помощью файла HDTunePro_Rus.exe.
- 3) В окне, расположенном под строкой основного меню в верхнем левом углу, выбрать жесткий диск ПЭВМ и нажать панель «Информация».
- 4) Занести в отчет паспорт жесткого диска, т.е. информацию, приведенную в раскрывшемся окне «Информация».
- 5) В окне, расположенном под строкой основного меню в верхнем левом углу, выбрать флэш-память и нажать панель «Информация».
- 6) Занести в отчет паспорт флэш-памяти, т.е. информацию, приведенную в раскрывшемся окне «Информация».
- 7) Сделать выводы по результатам проделанной работы.

Задание 3.4.2. Исследовать зависимость производительности жесткого диска от уровня акустического шума с помощью утилиты HD Tune. Для этого выполните следующие операции:

- 1) В окне, расположенном под строкой основного меню в верхнем левом углу, выбрать жесткий диск ПЭВМ и нажать панель «Уровень шума».
- 2) В открывшемся окне определить текущее логическое значение уровня шума.
- 3) Изменяя с помощью курсора шкалы-линейки и устанавливая с помощью кнопки «Установить» уровень шума от минимального значения (128) до максимального значения (254) с шагом 64, измерить среднее время доступа к данным на диске, нажимая каждый раз кнопку «Тест».
- 4) Построить график зависимости производительности жесткого диска от величины уровня акустического шума.
- 5) Сделать выводы по результатам проделанной работы.

Задание 3.4.3. Провести тестирование жесткого диска с помощью утилиты HD Tune с целью определения скорости передачи данных и времени доступа к данным. Для этого выполните следующие операции:

- 1) В окне, расположенном под строкой основного меню в верхнем левом углу, выбрать жесткий диск ПЭВМ и нажать панель «Тестирование».
- 2) В открывшемся окне установить опцию «Чтение» и нажать кнопку «Запустить».
- 3) Занести в отчет значения максимальной, минимальной и средней скорости передачи данных на жесткий диск. Построить график зависимости скорости передачи данных от объема накопителя. По результатам тестирования определить тип организации данных на жестком диске.
- 4) Занести в отчет значение среднего времени доступа к данным на жестком диске. Построить график зависимости времени доступа к данным от объема накопителя.
- 5) Сделать выводы по результатам проделанной работы.

Задание 3.4.4. Провести сканирование жесткого диска и флеш-памяти на предмет наличия ошибок с помощью утилиты HD Tune. Для этого выполните следующие операции:

- 1) В окне, расположенном под строкой основного меню в верхнем левом углу, выбрать флэш-память и нажать панель «Поиск ошибок».
- 2) В открывшемся окне установить единицы измерения – МБ, нажать кнопку «Запустить».
- 3) Занести в отчет результаты тестирования.
- 4) В окне, расположенном под строкой основного меню в верхнем левом углу, выбрать жесткий диск ПЭВМ и нажать панель «Поиск ошибок».

5) В открывшемся окне установить единицы измерения – сектор, нажать кнопку «Запустить».

6) Занести в отчет результаты тестирования.

7) Сделать выводы по результатам проделанной работы.

Задание 3.4.5. Провести мониторинг жесткого диска и флеш-памяти с помощью утилиты HD Tune. Для этого выполните следующие операции:

1) В окне, расположенном под строкой основного меню в верхнем левом углу, выбрать жесткий диск ПЭВМ и нажать панель «Мониторинг диска».

2) В открывшемся окне установить опцию все диски и нажать кнопку «Запустить».

3) По истечению 8 минут мониторинга остановить тестирование и занести в отчет результаты тестирования, в том числе скриншот диаграммы.

4) В окне, расположенном под строкой основного меню в верхнем левом углу, выбрать флеш-память и нажать панель «Мониторинг диска».

5) В открывшемся окне установить опцию все диски и нажать кнопку «Запустить».

6) По истечению 8 минут мониторинга остановить тестирование и занести в отчет результаты тестирования, в том числе скриншот диаграммы.

7) Сделать выводы по результатам проделанной работы.

Задание 3.4.6. Качественно оценить состояние жесткого диска ПЭВМ по показателям SMART-данных с помощью утилиты HD Tune. Для этого выполните следующие операции:

1) В окне, расположенном под строкой основного меню в верхнем левом углу, выбрать жесткий диск ПЭВМ и нажать панель «Здоровье».

2) В открывшемся окне проанализировать текущее значение и качественное состояние параметров S.M.A.R.T.

3) Сделать выводы о состоянии работоспособности жесткого диска.

3.5. Самостоятельная диагностика и тестирование внешних запоминающих устройств домашней ПЭВМ

На домашней ПЭВМ выполните все задания, приведенные пунктах 3.2 и 3.4, результаты занести в отчет по крактическому занятию.

Контрольные вопросы:

- 1) Предназначение и область применения программы Victoria.
- 2) Основные возможности программы Victoria.
- 3) Предназначение и основные функциональные возможности утилиты HD Tune.
- 4) Порядок установки программы Victoria и характеристика ее интерфейса.
- 5) Порядок чтения паспортов внешних запоминающих устройств ПЭВМ с помощью программы Victoria.
- 6) Порядок тестирования поверхности жесткого диска на дефектоскопию с помощью программы Victoria.
- 7) Порядок тестирования флеш-накопителя и CD(DVD)-диска с помощью программы Victoria.
- 8) Порядок диагностики состояния жесткого диска ПЭВМ с помощью технологии SMART в программе Victoria.
- 9) Порядок определения среднего времени позиционирования головок (доступа к данным на диске) с помощью программы Victoria.
- 10) Порядок установки утилиты HD Tune и характеристика ее интерфейса.
- 11) Порядок получения подробной информации о жестком диске ПЭВМ и флэш-памяти с помощью утилиты HD Tune.
- 12) Порядок исследования зависимости производительности жесткого диска от уровня акустического шума с помощью утилиты HD Tune.
- 13) Порядок тестирования жесткого диска с помощью утилиты HD Tune с целью

определения скорости передачи данных и времени доступа к данным.

- 14) Порядок сканирования жесткого диска и флеш-памяти на предмет наличия ошибок с помощью утилиты HD Tune.
- 15) Порядок мониторинга жесткого диска и флеш-памяти с помощью утилиты HD Tune.
- 16) Порядок качественной оценки состояние жесткого диска ПЭВМ по показателям SMART-данных с помощью утилиты HD Tune.

Раздел №6 ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА ЭВМ

Лабораторная работа №15 Исследование видеоадаптера и жидкокристаллического монитора (2 часа)

Содержание:

1. 1. Исследование видеоадаптера и жидкокристаллического монитора.

Цель практического занятия: выработка умения тестировать и оценивать характеристики видеоадаптеров и мониторов ПЭВМ, прежде всего наиболее распространенных жидкокристаллических мониторов, с помощью специализированных тестовых программ. В качестве тестовых программ используются следующие: GPU-Z.0.5.5, TFTtest версия 1.52, IsMyLsdOK 1.71, Atrise Lutcurve 1.5.3. В результате выполнения практического занятия студенты также выработают навыки настройки жидкокристаллических мониторов.

Теоретические сведения:

Задание:

1.2. План выполнения практической работы по учебному вопросу:

- 1) Ознакомление и исследование прикладной программы GPU-Z.0.5.5 для отображения технической информации о видеоадаптере.
- 2) Провести тестирование тестирования видеоадаптера с помощью программы GPU-Z.0.5.5.
- 3) Ознакомление и исследование прикладной программы TFT_test_1.52 для выявления всех дефектов монитора.
- 4) Провести проверку монитора на битые или сгоревшие пиксели с помощью программы TFT_test_1.52.
- 5) Ознакомление и исследование прикладной программы IsMyLcdOK для выявления наличие битых или поврежденных пикселей на LCD мониторах.
- 6) Провести проверку монитора на наличие битых или поврежденных пикселей с помощью программы IsMyLcdOK.
- 7) Ознакомление и исследование прикладной программы Atrise Lutcurve для корректировки любых нелинейности LCD мониторов.
- 8) Произвести исследование всех возможных функций программы Atrise Lutcurve.

3 Порядок выполнения работы:

3.1. Ознакомление и исследование прикладной программы **GPU-Z.0.5.5** для отображения технической информации о видеоадаптере.

Задание 3.1.1. Запустить программу GPU-Z.0.5.5 и ознакомиться с ее интерфейсом. Для этого выполните следующие операции:

- 1) Разархивировать самораспаковывающийся файл GPU-Z.0.5.5_CrashBox.Ru.rar, находящийся в папке ЛР Арх ЭВМ №7, разместив его на рабочем столе в папке GPU-Z.0.5.5_CrashBox.Ru
- 2) Запустить программу с помощью файла GPU-Z.0.5.5_CrashBox.Ru.exe

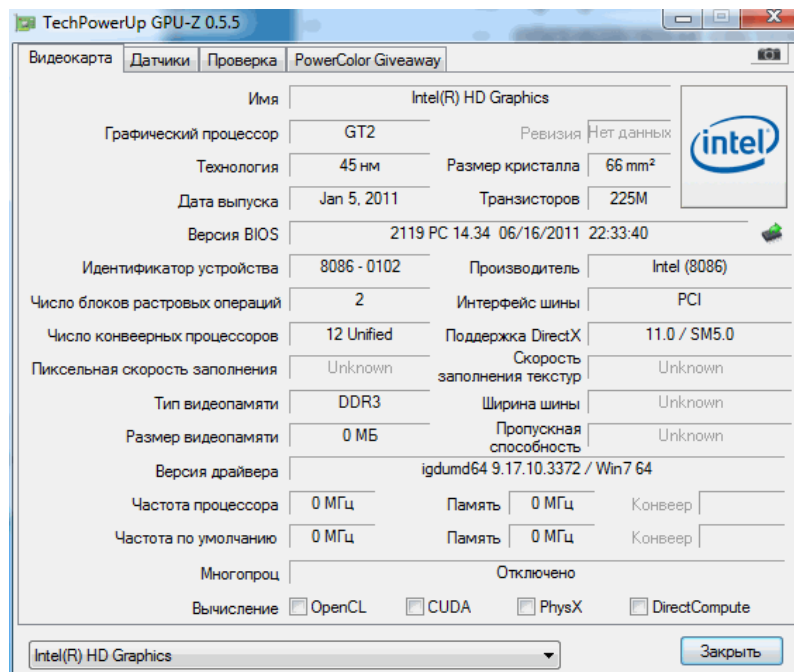


Рисунок 1 – Окно программы GPU-Z

3) Ознакомиться с интерфейсом программы и занести его в отчет по ЛР.

4) Сделать выводы.

3.2. Провести тестирование тестирования видеоадаптера с помощью программы GPU-Z.0.5.5.

1) С помощью программы определить тип видеокарты и её температуру.

2) Сделать выводы по результатам проделанной работы.

3.3. Ознакомление и исследование прикладной программы TFT_test_1.52 для выявления всех дефектов монитора.

1) Разархивировать самораспаковывающийся файл TFT_test_1.52_rus_Portable.rar, находящийся в папке ЛР Арх ЭВМ №7, разместив его на рабочем столе в папке TFT test 1.52

2) Запустить программу с помощью файла TFTTEST.EXE

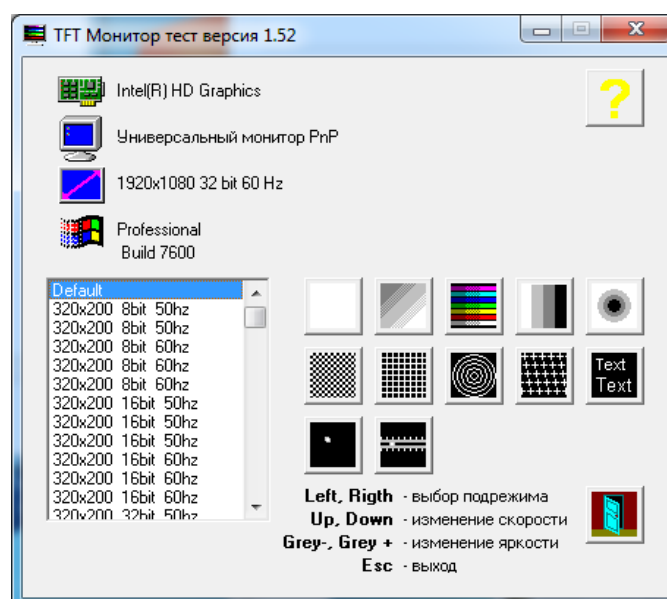


Рисунок 2 – Окно программы TFT test

- 3) Ознакомиться с интерфейсом программы и занести его в отчет по ЛР.
- 4) Сделать выводы.

3.4 Провести проверку монитора на битые или сгоревшие пиксели с помощью программы TFT_test_1.52.

- 1) Запустить поочередно все режимы с помощью клавиатуры и мышки в программе TFT test 1.52 и попытаться выявить битые или сгоревшие пиксели на учебном мониторе.
- 2) Сделать выводы по результатам проделанной работы.

3.5. Ознакомление и исследование прикладной программы IsMyLcdOK для выявления наличие битых или поврежденных пикселей на LCD мониторах.

- 1) Разархивировать самораспаковывающийся файл IsMyLcdOK_1.71_Portable.rar, находящийся в папке ЛР Арх ЭВМ №7, разместив его на рабочем столе в папке IsMyLcdOK_1.71_Portable
- 2) Запустить программу с помощью файла IsMyLcdOK.exe

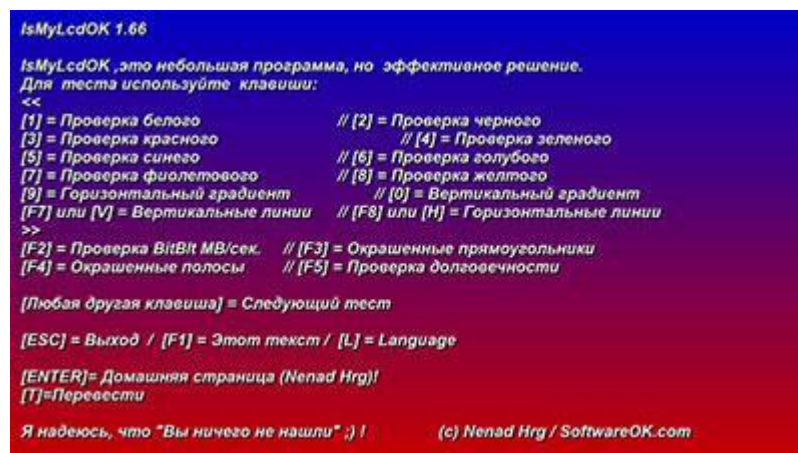


Рисунок 2 - IsMyLcdOK

- 3) Ознакомиться с интерфейсом программы и занести его в отчет по ЛР.
- 4) Сделать выводы.

3.6. Провести проверку монитора на наличие битых или поврежденных пикселей с помощью программы IsMyLcdOK.

- 1) Запустить программу TFT test 1.52 и выполнять рекомендации программы для обнаружения битых или повреждённых пикселей на учебном мониторе.
- 2) Сделать выводы по результатам проделанной работы.

3.7. Ознакомление и исследование прикладной программы Atrise Lutcurve для коррективки любых нелинейности LCD мониторов.

- 1) Зайти в папку ЛР Арх ЭВМ №7 => Portable Atrise Lutcurve 1.5.3 Rus.
- 2) Запустить программу с помощью файла Lutcurve.exe
- 3) Ознакомиться с интерфейсом программы и занести его в отчет по ЛР.
- 4) Сделать выводы.

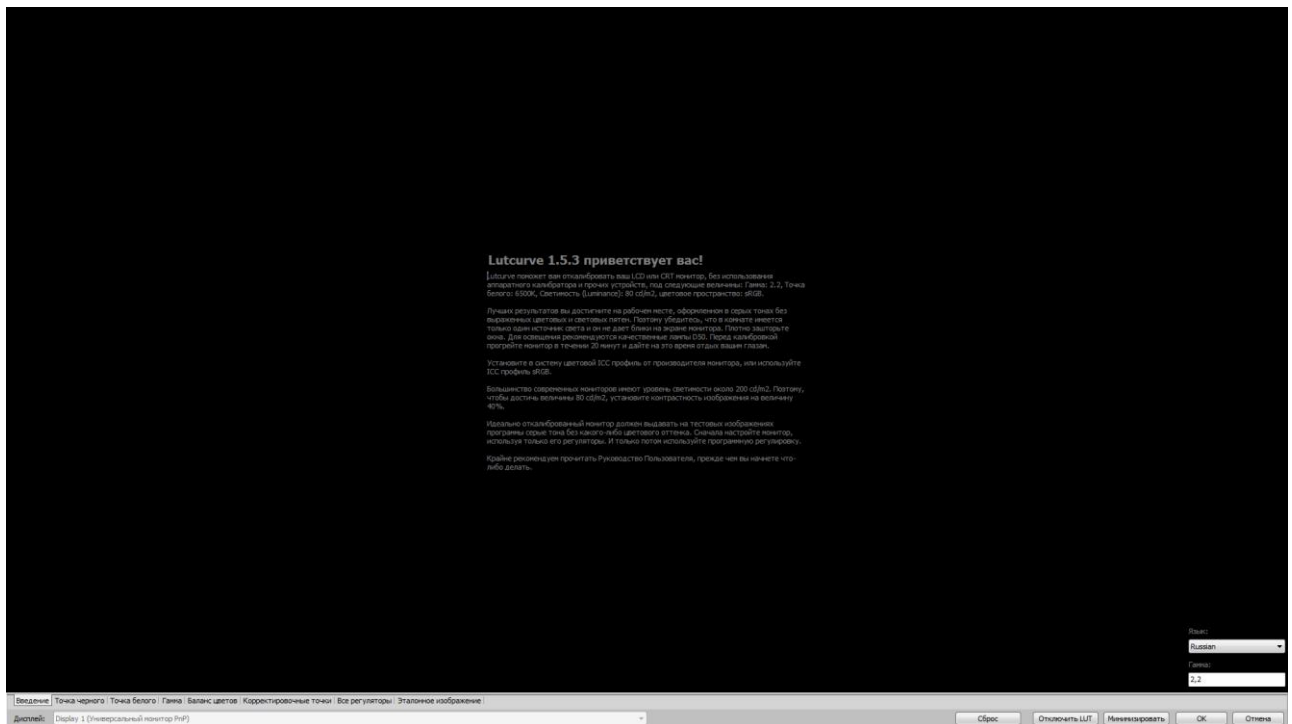


Рисунок 3 - Atrise Lutcurve

3.8 Произвести исследование всех возможных функций программы Atrise Lutcurve.

1) При первом запуске откроется подробное меню и вам представится выбор корректировочных функций.

Функции:

1. Точка черного.
2. Точка белого.
3. Гамма.
4. Баланс цветов.
5. Корректировочные точки.
6. Выбор всех регуляторов.
7. Эталон изображения.

2) Произведите корректировку по каждому из пунктов программы и опишите их в ЛР.

Качество калибровки будет зависеть от вашей аккуратности. Оно может быть очень близко к качеству аппаратной калибровки и даже превосходить ее.

3) Сделать выводы по результатам проделанной работы

Контрольные вопросы:

- 1) Устройство современной видеокарты.
- 2) Типы и виды видеоадаптеров.
- 3) Технологии построения трехмерных изображений.
- 4) Аппаратное ускорение графических функций.
- 5) Производители видеокарт и графических процессоров.
- 6) Технические характеристики современных видеокарт.

Лабораторная работа №16 Организация межкомпьютерной связи (2 часа)

Содержание:

1. Организация межкомпьютерной связи.

Теоретические сведения:

Часто при организации связи между двумя компьютерами за одним компьютером закрепляется роль поставщика ресурсов (программ, данных и т.д.), а за другим — роль пользователя этих ресурсов. В этом случае первый компьютер называется **сервером**, а второй — **клиентом** или рабочей станцией. Работать можно только на компьютере-клиенте под управлением специального программного обеспечения.



Рисунок 1 - Сетевой сервер HP LD PRO

Сервер (англ. serve — обслуживать) — это высокопроизводительный компьютер с большим объёмом внешней памяти, который обеспечивает обслуживание других компьютеров путем управления распределением дорогостоящих ресурсов совместного пользования (программ, данных и периферийного оборудования).

Примерами серверов могут служить:

- сервер телекоммуникаций, обеспечивающий услуги по связи данной локальной сети с внешним миром;
- вычислительный сервер, дающий возможность производить вычисления, которые невозможно выполнить на рабочих станциях;
- дисковый сервер, обладающий расширенными ресурсами внешней памяти и предоставляющий их в использование рабочим станциям и, возможно, другим серверам;
- файловый сервер, поддерживающий общее хранилище файлов для всех рабочих станций;
- сервер баз данных фактически обычная СУБД, принимающая запросы по локальной сети и возвращающая результаты.

Сервер локальной сети предоставляет ресурсы (услуги) рабочим станциям и/или другим серверам.

Клиент (иначе, рабочая станция) — любой компьютер, имеющий доступ к услугам сервера.

Клиентом также называют прикладную программу, которая от имени пользователя получает услуги сервера. Соответственно, программное обеспечение, которое позволяет компьютеру предоставлять услуги другому компьютеру, называют сервером — так же, как и сам компьютер. Для преодоления несовместимости интерфейсов отдельных компьютеров вырабатывают специальные стандарты, называемые протоколами коммуникации. Итак, клиентом принято называть компонент локальной сети, запрашивающий услуги у некоторого сервера и сервером - компонент локальной сети, оказывающий услуги некоторым клиентам.

Для работы с сетью необходимо наличие специального сетевого программного обеспечения, которое обеспечивает передачу данных в соответствии с заданным протоколом.

Протокол коммуникации — это согласованный набор конкретных правил обмена информацией между разными устройствами передачи данных. Имеются протоколы для скорости передачи, форматов данных, контроля ошибок и др.

Протоколы коммуникации предписывают разбить весь объём передаваемых данных на пакеты — отдельные блоки фиксированного размера. Пакеты нумеруются, чтобы их затем можно было собрать в правильной последовательности.

К данным, содержащимся в пакете, добавляется дополнительная информация примерно такого формата:

Адрес получателя	Адрес отправителя	Длина	Данные	Поле контрольной суммы
---------------------	----------------------	-------	--------	------------------------------

Контрольная сумма данных пакета содержит информацию, необходимую для контроля ошибок. Первый раз она вычисляется передающим компьютером. После того, как пакет будет передан, контрольная сумма повторно вычисляется принимающим компьютером. Если значения не совпадают, это означает, что данные пакета были повреждены при передаче. Такой пакет отбрасывается, и автоматически направляется запрос повторно передать пакет.

При установлении связи устройства обмениваются сигналами для согласования коммуникационных каналов и протоколов. Этот процесс называется подтверждением установления связи (англ. HandShake — рукопожатие).

Что такое компьютерная сеть

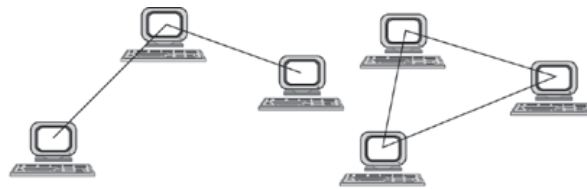
Компьютерная сеть (англ. Computer NetWork, от net — сеть и work — работа) — совокупность компьютеров, соединенных с помощью каналов связи и средств коммутации в единую систему для обмена сообщениями и доступа пользователей к программным, техническим, информационным и организационным ресурсам сети.

Компьютерную сеть представляют как совокупность узлов (компьютеров и сетевого оборудования) и соединяющих их ветвей (каналов связи). Ветвь сети — это путь, соединяющий два смежных узла. Различают узлы оконечные, расположенные в конце только одной ветви, промежуточные, расположенные на концах более чем одной ветви, и смежные — такие узлы соединены по крайней мере одним путём, не содержащим никаких других узлов. Компьютеры могут объединяться в сеть разными способами.

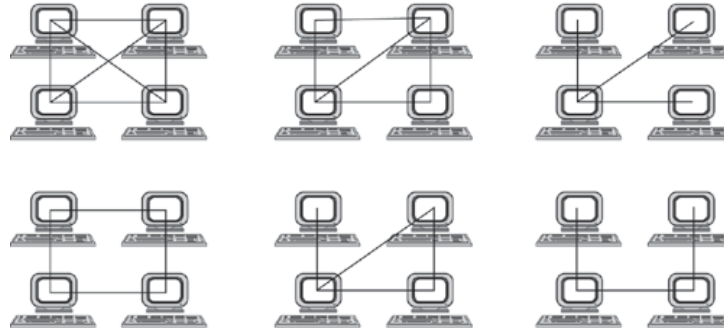
Логический и **физический** способы соединения компьютеров, кабелей и других компонентов, в целом составляющих сеть, называется ее **топологией**. Топология характеризует свойства сетей, не зависящие от их размеров. При этом не учитывается производительность и принцип работы этих объектов, их типы, длины каналов, хотя при проектировании эти факторы очень важны.

Как только компьютеров становится больше двух, возникает проблема выбора **конфигурации физических связей** или **топологии**. Под топологией сети понимается конфигурация графа, вершинам которого соответствуют конечные узлы сети (например, компьютеры) и коммуникационное оборудование (например, маршрутизаторы), а ребрам — электрические и информационные связи между ними.

Число возможных конфигураций резко возрастает при увеличении числа связываемых устройств. Так, если три компьютера мы можем связать двумя способами, то для четырех компьютеров можно предложить уже шесть топологически различных конфигураций (при условии неразличимости компьютеров).



а) вариант связи трех компьютеров



а) вариант связи четырех компьютеров

Рисунок 2 - Варианты связи компьютеров

Можно соединять каждый компьютер с каждым или же связывать их последовательно, предполагая, что они будут общаться, передавая друг другу сообщения "транзитом". При этом транзитные узлы должны быть оснащены специальными средствами, позволяющими выполнять эту специфическую посредническую операцию. В роли транзитного узла может выступать как универсальный компьютер, так и специализированное устройство.

От выбора топологии связей зависят многие характеристики сети. Например, наличие между узлами нескольких путей повышает надежность сети и делает возможной балансировку загрузки отдельных каналов. Простота присоединения новых узлов, свойственная некоторым топологиям, делает сеть легко расширяемой. Экономические соображения часто приводят к выбору топологий, для которых характерна минимальная суммарная длина линий связи.

Среди множества возможных конфигураций различают полносвязные и неполносвязные:



Рисунок 3 - Типы возможных конфигураций

Полносвязная топология соответствует сети, в которой каждый компьютер непосредственно связан со всеми остальными.

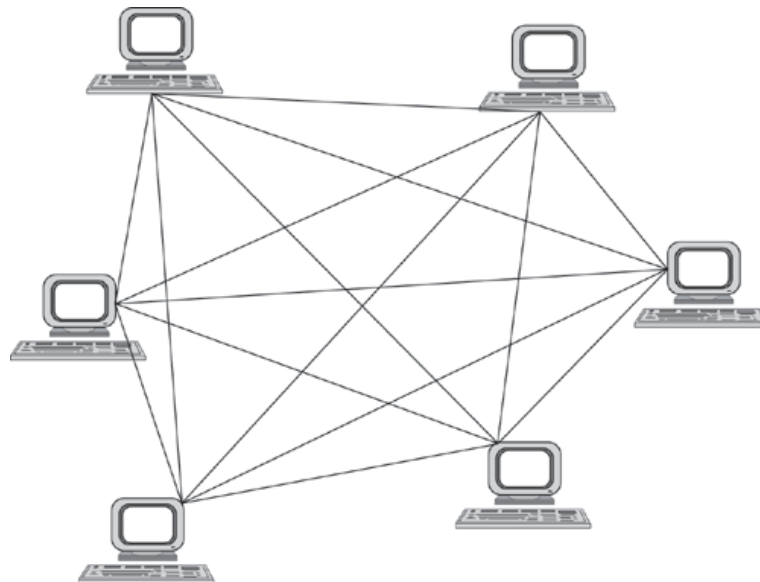


Рисунок 4 - Полносвязная конфигурация

Несмотря на логическую простоту, это вариант громоздкий и неэффективный. Действительно, каждый компьютер в сети должен иметь большое количество коммуникационных портов, достаточное для связи с каждым из остальных компьютеров. Для каждой пары компьютеров должна быть выделена отдельная физическая линия связи. (В некоторых случаях даже две, если невозможно использование этой линии для двусторонней передачи.) Полносвязные топологии в крупных сетях применяются редко, так как для связи N узлов требуется $N(N-1)/2$ физических дуплексных линий связи, т.е. имеет место квадратическая зависимость. Чаще этот вид топологии используется в многомашинных комплексах или в сетях, объединяющих небольшое количество компьютеров.

Все другие варианты основаны на неполносвязных топологиях, когда для обмена данными между двумя компьютерами может потребоваться промежуточная передача данных через другие узлы сети.

Ячеистая топология (mesh) получается из полносвязной путем удаления некоторых возможных связей. Ячеистая топология допускает соединение большого количества компьютеров и характерна для крупных сетей.

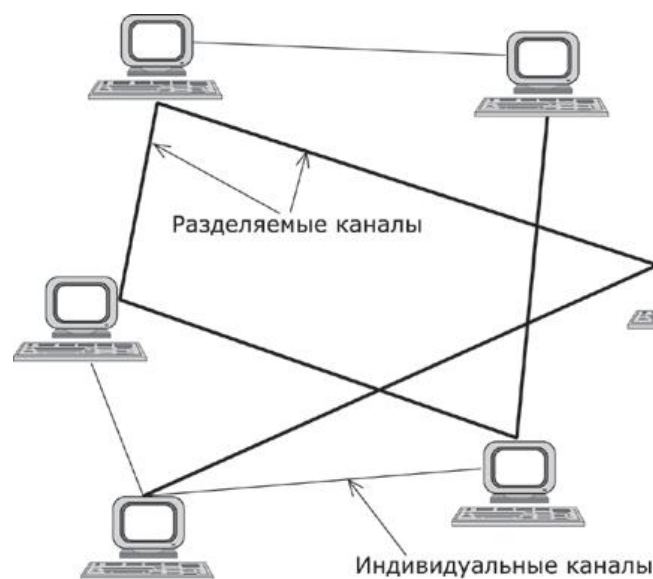


Рисунок 5 - Ячеистая топология

В сетях с **кольцевой конфигурацией** данные передаются по кольцу от одного компьютера к другому.

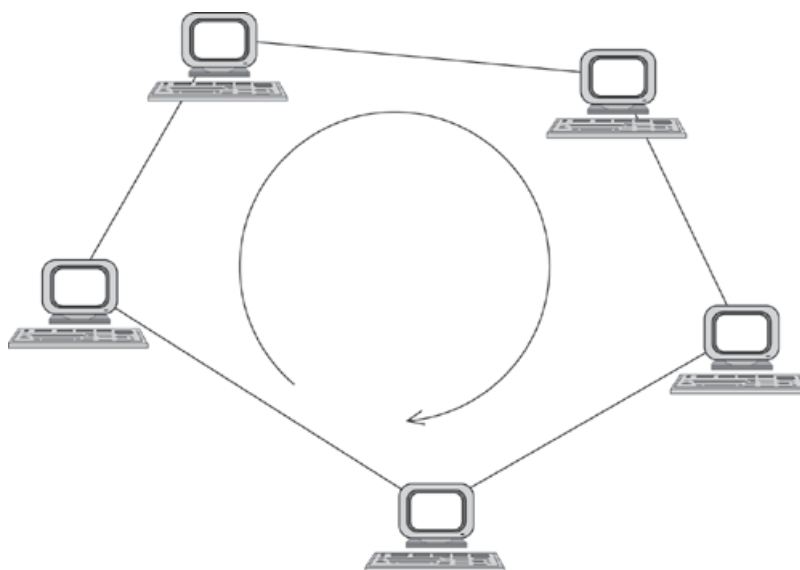


Рисунок 6 - Топология "кольцо"

Главное достоинство "кольца" в том, что оно по своей природе обладает свойством резервирования связей. Действительно, любая пара узлов соединена здесь двумя путями — по часовой стрелке и против. "Кольцо" представляет собой очень удобную конфигурацию и для организации обратной связи — данные, сделав полный оборот, возвращаются к узлу-источнику. Поэтому отправитель в данном случае может контролировать процесс доставки данных адресату. Часто это свойство "кольца" используется для тестирования связности сети и поиска узла, работающего некорректно. В то же время в сетях с кольцевой топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какой-либо станции не прерывался канал связи между остальными станциями "кольца".

Топология **"звезда"** (рис.4.5) образуется в том случае, когда каждый компьютер с помощью отдельного кабеля подключается к общему центральному устройству, называемому концентратором.

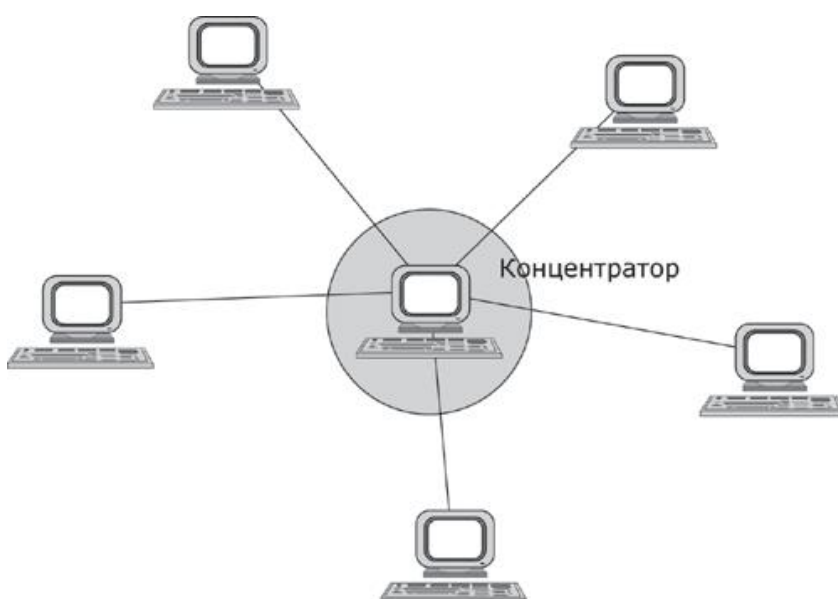


Рисунок 7 - Топология "звезда"

В функции концентратора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. В роли концентратора может выступать как компьютер, так и специализированное устройство, такое как многоходовый повторитель, коммутатор или маршрутизатор. К недостаткам топологии типа "звезда" относится более высокая стоимость сетевого оборудования, связанная с необходимостью приобретения специализированного центрального устройства. Кроме того, возможности наращивания количества узлов в сети ограничиваются количеством портов концентратора.

Иногда имеет смысл строить сеть с использованием нескольких концентраторов, иерархически соединенных между собой связями типа "звезда". Получаемую в результате структуру называют также **деревом**. В настоящее время дерево является самым распространенным типом топологии связей, как в локальных, так и в глобальных сетях.

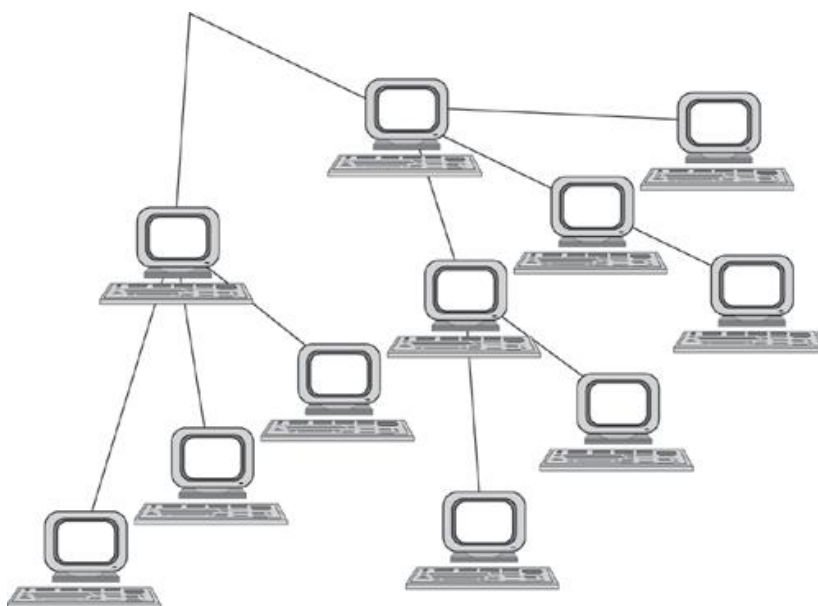


Рисунок 8 - Топология "иерархическая звезда" или "дерево"

Особым частным случаем конфигурации звезда является конфигурация **"общая шина"** (рис. 4.7). Здесь в роли центрального элемента выступает пассивный кабель, к которому по схеме "монтажного ИЛИ" подключается несколько компьютеров (такую же топологию имеют многие сети, использующие беспроводную связь — роль общей шины здесь играет общая радиосреда). Передаваемая информация распространяется по кабелю и доступна одновременно всем присоединенным к нему компьютерам.



Рисунок 9 - Топология "общая шина"

Основными преимуществами такой схемы являются низкая стоимость и простота наращивания, т.е. присоединения новых узлов к сети.

Самым серьезным недостатком "общей шины" является ее недостаточная надежность: любой дефект кабеля или какого-нибудь из многочисленных разъемов полностью парализует всю сеть. Другой недостаток "общей шины" — невысокая производительность, так как при таком способе подключения в каждый момент времени только один компьютер может передавать данные по сети, поэтому пропускная способность канала связи всегда делится между всеми узлами сети. До недавнего времени "общая шина" являлась одной из самых популярных топологий для локальных сетей.

Архитектура сети

Важнейшая характеристика компьютерной сети — её **архитектура**.

Архитектура сети — это реализованная структура сети передачи данных, определяющая её топологию, состав устройств и правила их взаимодействия в сети. В рамках архитектуры сети рассматриваются вопросы кодирования информации, её адресации и передачи, управления потоком сообщений, контроля ошибок и анализа работы сети в аварийных ситуациях и при ухудшении характеристик.

Наиболее распространённые архитектуры:

- **Ethernet** (англ. ether — эфир) — широковещательная сеть. Это значит, что все станции сети могут принимать все сообщения. Топология — линейная или звездообразная. Скорость передачи данных 10 или 100 Мбит/сек.
- **Arcnet** (Attached Resource Computer Network — компьютерная сеть соединённых ресурсов) — широковещательная сеть. Физическая топология — дерево. Скорость передачи данных 2,5 Мбит/сек.
- **Token Ring** (эстафетная кольцевая сеть, сеть с передачей маркера) — кольцевая сеть, в которой принцип передачи данных основан на том, что каждый узел кольца ожидает прибытия некоторой короткой уникальной последовательности битов — маркера — из смежного предыдущего узла. Поступление маркера указывает на то, что можно передавать сообщение из данного узла дальше по ходу потока. Скорость передачи данных 4 или 16 Мбит/сек.
- **FDDI** (Fiber Distributed Data Interface) — сетевая архитектура высокоскоростной передачи данных по оптоволоконным линиям. Скорость передачи — 100 Мбит/сек. Топология — двойное кольцо или смешанная (с включением звездообразных или древовидных подсетей). Максимальное количество станций в сети — 1000. Очень высокая стоимость оборудования.
- **ATM** (Asynchronous Transfer Mode) — перспективная, пока ещё очень дорогая архитектура, обеспечивает передачу цифровых данных, видеоинформации и голоса по одним и тем же линиям. Скорость передачи до 2,5 Гбит/сек. Линии связи оптические.

Соединение устройств сети между собой

Для этого используется специальное оборудование:



Рисунок 11 - Сетевой интерфейсный адаптер

Сетевые интерфейсные адаптеры (Network Interface Card, NIC) – это периферийное устройство компьютера, непосредственно взаимодействующее со средой передачи данных, которая прямо или через другое коммуникационное оборудование связывает его с другими компьютерами. Это устройство решает задачи надежного обмена двоичными данными, представленными соответствующими электромагнитными сигналами, по внешним линиям связи. Как и любой контроллер компьютера, сетевой адаптер работает под управлением драйвера операционной системы и распределение функций между сетевым адаптером и драйвером может изменяться от реализации к реализации.. В соответствии с определённым протоколом управляют доступом к среде передачи данных. Размещаются в системных блоках компьютеров, подключенных к сети.

Сетевые адаптеры различаются по типу и разрядности используемой в компьютере внутренней шины данных - ISA, EISA, PCI, MCA.

Сетевые адаптеры различаются также по типу принятой в сети сетевой технологии – Ethernet, Token Ring, FDDI и т.п. Как правило, конкретная модель сетевого адаптера работает по определенной сетевой технологии (например, Ethernet). В связи с тем, что для каждой технологии сейчас имеется возможность использования различных сред передачи данных (тот же Ethernet поддерживает коаксиальный кабель, неэкранированную витую пару и оптоволоконный кабель), сетевой адаптер может поддерживать как одну, так и одновременно несколько сред.

Сетевые кабели (коаксиальные, состоящие из двух изолированных между собой концентрических проводников, из которых внешний имеет вид трубки; оптоволоконные; кабели на витых парах, образованные двумя переплетёнными друг с другом проводами, и др.).

Для построения локальных связей в вычислительных сетях в настоящее время используются различные виды кабелей - коаксиальный кабель, кабель на основе экранированной и неэкранированной витой пары и оптоволоконный кабель. Наиболее популярным видом среды передачи данных на небольшие расстояния (до 100 м) становится неэкранированная витая пара, которая включена практически во все современные стандарты и технологии локальных сетей и обеспечивает пропускную способность до 100 Мб/с.

Оптоволоконный кабель широко применяется как для построения локальных связей, так и для образования магистралей глобальных сетей. Оптоволоконный кабель может обеспечить очень высокую пропускную способность канала (до нескольких Гб/с) и передачу на значительные расстояния (до нескольких десятков километров без промежуточного усиления сигнала).

В качестве среды передачи данных в вычислительных сетях используются также электромагнитные волны различных частот - КВ, УКВ, СВЧ. Однако, пока в локальных сетях радиосвязь используется только в тех случаях, когда оказывается невозможной прокладка кабеля, например, в зданиях, являющихся памятниками архитектуры. Это объясняется прежде всего недостаточной надежностью сетевых технологий, построенных на использовании электромагнитного излучения.

Кабель на основе скрученных пар (витая пара, TP)

Витая пара (Twisted Pair — TP) состоит из четырех пар цельных или витых изолированных проводников. Из них используются только две пары — одна для приема, другая для передачи данных. При использовании защитной экранизации кабеля он маркируется как STP (Shielded TP), без экранизации — как UTP (Unshielded TP). Есть несколько различных категорий этого вида кабелей, для прокладки 100-мегабитных сетей рекомендуется использовать кабели категорий 5 и 5е.

Скорость передачи данных при использовании сетевых интерфейсов (сетевых карт) типа 10Base-T составляет 10 Мбит/с, а для 100Base-TX — 100 Мбит/с.

Максимальная длина сегмента сети на витой паре без повторителей 100 м. Используя витую пару, можно соединить два компьютера напрямую, без дополнительных сетевых

устройств. Для этого используется кабель, в котором перекрестно соединены пары передающих и принимающих контактов сетевых интерфейсов.

Коаксиальный кабель

Состоит из центрального проводника (сплошного или многожильного), покрытого слоем полимерного изолятора, поверх которого расположен другой проводник (экран). Экран представляет собой оплетку из медного провода вокруг изолятора или обернутую вокруг изолятора фольгу. В высококачественных кабелях присутствуют и оплетка и фольга. Коаксиальный кабель обеспечивает более высокую помехоустойчивость по сравнению с витой парой, но он дороже. Существуют различные виды коаксиальных кабелей. При установке сети следует выбрать кабель в точном соответствии со спецификацией.

Оптический кабель

Состоит из одного или нескольких "кварцевых волокон (иногда полимерных), покрытых защитной оболочкой. Оболочка, как правило, состоит из нескольких слоев для обеспечения лучшей защиты волокон.

Коннекторы (соединители) для подключения кабелей к компьютеру; разъёмы для соединения отрезков кабеля. К разъёмам адаптеров подключается сетевой кабель.

Трансиверы повышают уровень качества передачи данных по кабелю, отвечают за приём сигналов из сети и обнаружение конфликтов.

Хаб (концентраторы) и коммутирующие хабы (коммутаторы) расширяют топологические, функциональные и скоростные возможности компьютерных сетей. Хаб с набором разнотипных портов позволяет объединять сегменты сетей с различными кабельными системами. К порту хаба можно подключать как отдельный узел сети, так и другой хаб или сегмент кабеля.

Повторители (репитеры) усиливают сигналы, передаваемые по кабелю при его большой длине.



Рисунок 12 - Повторитель

Классификация компьютерных сетей по степени географического распространения

По степени географического распространения сети делятся на локальные, городские, корпоративные, глобальные и др.

Локальная сеть (ЛВС или LAN — Local Area NetWork) — сеть, связывающая ряд компьютеров в зоне, ограниченной пределами одной комнаты, здания или предприятия.

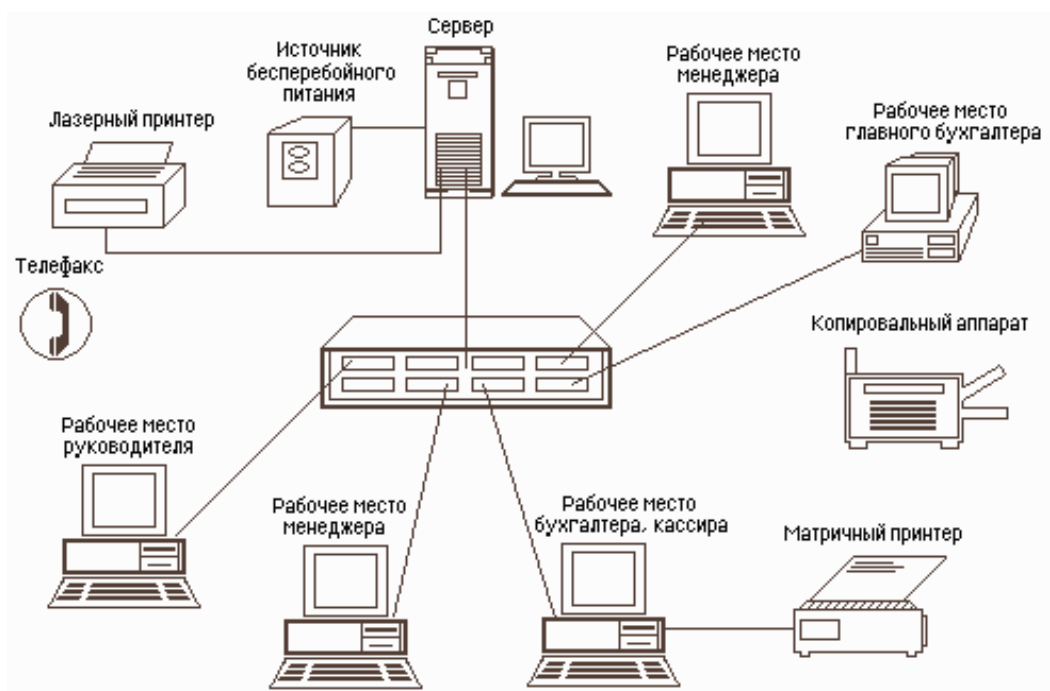


Рисунок 12 - Небольшая офисная локальная сеть

Глобальная сеть (ГВС или WAN — World Area NetWork) — сеть, соединяющая компьютеры, удалённые географически на большие расстояния друг от друга. Отличается от локальной сети более протяжёнными коммуникациями (спутниковыми, кабельными и др.). Глобальная сеть объединяет локальные сети.

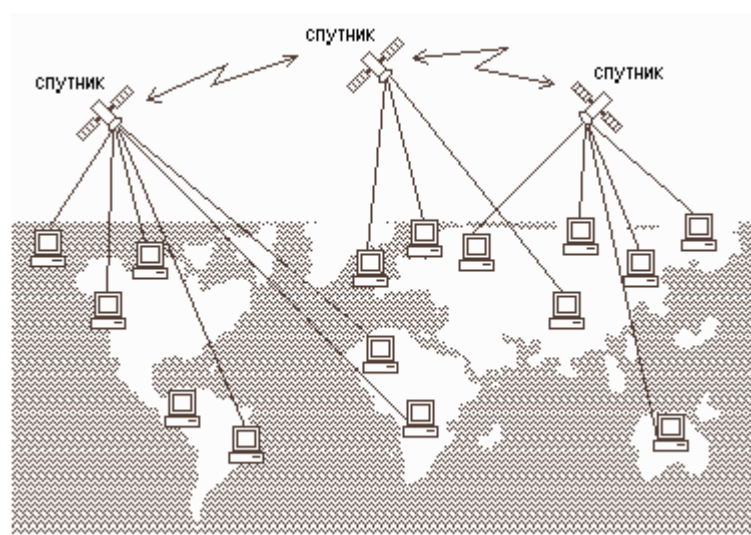


Рисунок 13 - Глобальная сеть

Городская сеть (MAN — Metropolitan Area NetWork) — сеть, которая обслуживает информационные потребности большого города.

Соединение между собой локальных сетей

Для соединения локальных сетей используются следующие устройства, которые различаются между собой по назначению и возможностям:

Мост (англ. Bridge) — связывает две локальные сети. Передаёт данные между сетями в пакетном виде, не производя в них никаких изменений. Ниже на рисунке показаны три локальные сети, соединённые двумя мостами.

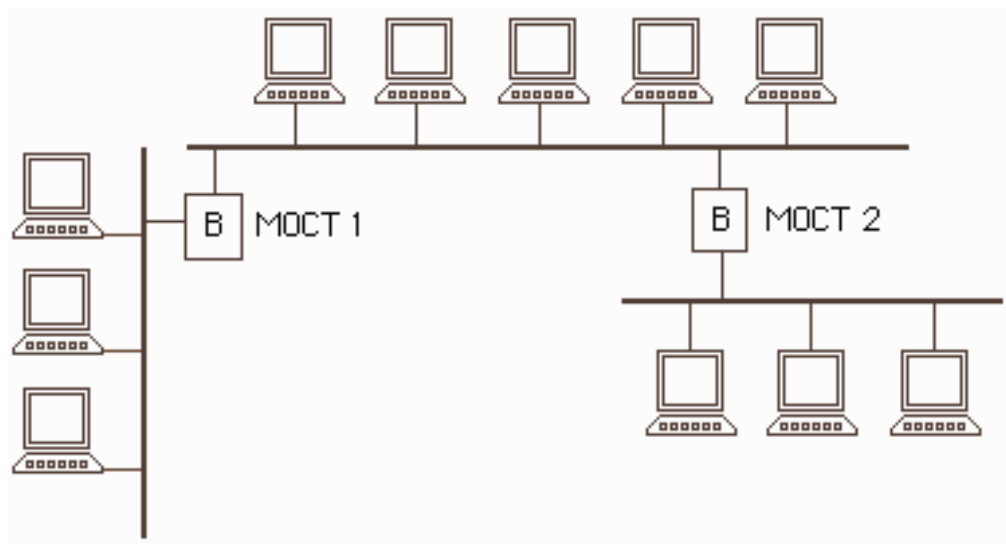


Рисунок 14 - Соединение локальных сетей посредством мостов

Здесь мосты создали расширенную сеть, которая обеспечивает своим пользователям доступ к прежде недоступным ресурсам. Кроме этого, мосты могут фильтровать пакеты, охраняя всю сеть от локальных потоков данных и пропуская наружу только те данные, которые предназначены для других сегментов сети.

Маршрутизатор (англ. Router) объединяет сети с общим протоколом более эффективно, чем мост. Он позволяет, например, расщеплять большие сообщения на более мелкие куски, обеспечивая тем самым взаимодействие локальных сетей с разным размером пакета.

Маршрутизатор может пересылать пакеты на конкретный адрес (мосты только отфильтровывают ненужные пакеты), выбирать лучший путь для прохождения пакета и многое другое. Чем сложнее и больше сеть, тем больше выгода от использования маршрутизаторов.

Мостовой маршрутизатор (англ. Brouter) — это гибрид моста и маршрутизатора, который сначала пытается выполнить маршрутизацию, где это только возможно, а затем, в случае неудачи, переходит в режим моста.

Шлюз (англ. GateWay), в отличие от моста, применяется в случаях, когда соединяемые сети имеют различные сетевые протоколы. Поступившее в шлюз сообщение от одной сети преобразуется в другое сообщение, соответствующее требованиям следующей сети. Таким образом, шлюзы не просто соединяют сети, а позволяют им работать как единая сеть. С помощью шлюзов также локальные сети подсоединяются к мейнфреймам — универсальным мощным компьютерам.

Работа беспроводных сетей

Беспроводные сети используются там, где прокладка кабелей затруднена, нецелесообразна или просто невозможна. Например, в исторических зданиях, промышленных помещениях с металлическим или железобетонным полом, в офисах, полученных в краткосрочную аренду, на складах, выставках, конференциях и т.п.

В этих случаях сеть реализуется при помощи сетевых радио-адаптеров, снабжённых всенаправленными антеннами и использующих в качестве среды передачи информации радиоволны. Такая сеть реализуется топологией “Все-Со-Всеми” и работоспособна при дальности 50–200 м.

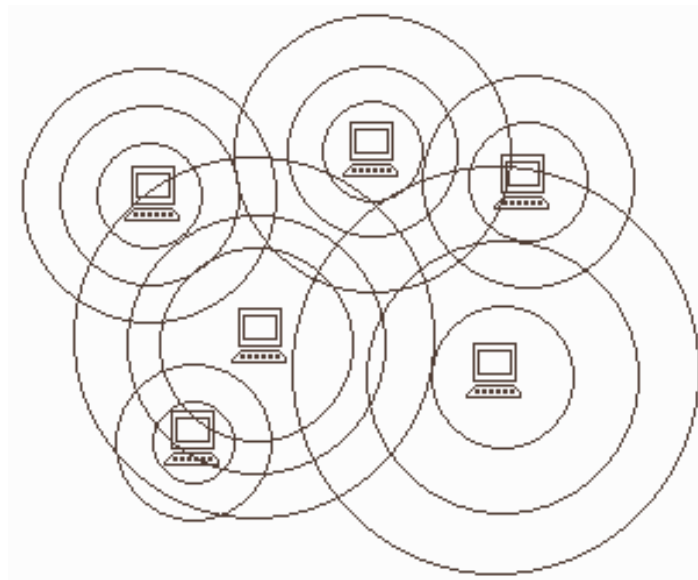


Рисунок 17 - Топология "Все-Со-Всеми"

Для связи между беспроводной и кабельной частями сети используется специальное устройство, называемое **точкой входа** (или **радиомостом**). Можно использовать и обычный компьютер, в котором установлены два сетевых адаптера — беспроводной и кабельный.

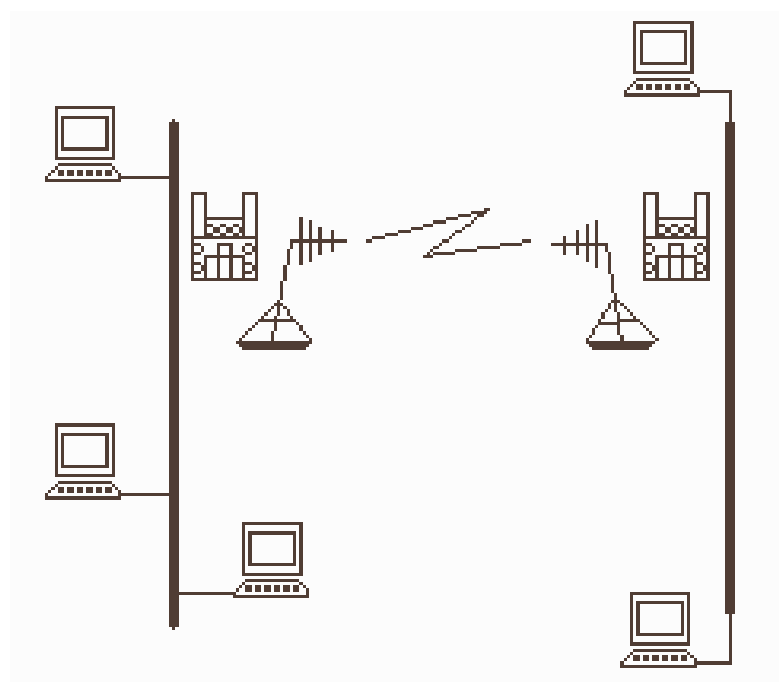


Рисунок 18 - Топология "точка-точка"

Другой важной областью применения беспроводных сетей является организация связи между удалёнными сегментами локальных сетей при отсутствии инфраструктуры передачи данных (кабельных сетей общего доступа, высококачественных телефонных линий и др.), что типично для нашей страны. В этом случае для наведения беспроводных мостов между двумя удалёнными сегментами используются радиомосты с антенной направленного типа.

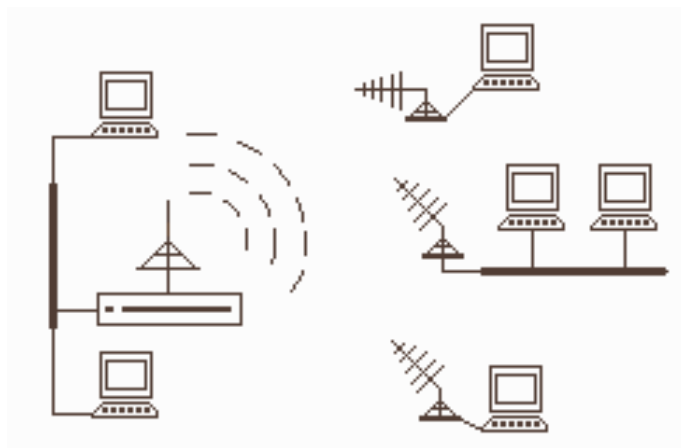


Рисунок 19 -Топология типа "звезда"

Если в сеть нужно объединить несколько сегментов, то используется топология типа «звезда». При этом в центральном узле устанавливается всенаправленная антенна, а удалённых узлах — направленные. Сети звездообразной топологии могут образовывать сети разнообразной конфигурации.

Сетевая магистраль с беспроводным доступом позволяет отказаться от использования медленных модемов

Задания

- 1) Составить монтажную схему полносвязного соединения 4-х ПЭВМ.
- 2) Составить монтажную схему соединения 4-х ПЭВМ по топологии «кольцо».
- 3) Составить монтажную схему соединения 4-х ПЭВМ по топологии «звезда».
- 4) Составить монтажную схему соединения 4-х ПЭВМ по топологии «общая шина».
- 5) Составить блок-схему сетевого интерфейсного адаптера.
- 6) Составить алгоритм работы сетевого интерфейсного адаптера.
- 7) Составить монтажную схему соединения 2-х ПЭВМ через повторитель.
- 8) Составить монтажную схему соединения 2-х ПЭВМ через концентратор и коммутатор.
- 9) Составить блок-схему локальной сети компьютерной аудитории.
- 10) Составить блок-схему локальной сети компьютерной аудитории с выходом в глобальную сеть через мостовой маршрутизатор.
- 11) Составить блок-схему беспроводной локальной сети компьютерной аудитории с помощью Wi-Fi маршрутизатора.
- 12) Выполнить работы по монтажу витой пары и подключению ее к компьютеру и маршрутизатору.
- 13) Выполнить работы по организации связи между 2-мя ПЭВМ с помощью витой
- 14) пары.
- 15) Выполнить работы по установке сетевой карты в учебный системный блок
- 16) ПЭВМ.
- 17) Выполнить работы по замене сетевой карты в учебном системном блоке ПЭВМ с установкой нового драйвера.

Лабораторная работа №17 Многопроцессорные вычислительные системы (2 часа)

Содержание:

1. Многопроцессорные вычислительные системы.

Теоретические сведения:

В настоящее время тенденция в развитии микропроцессоров и систем, построенных на их основе, направлена на все большее повышение их производительности. Вычислительные

возможности любой системы достигают своей наивысшей производительности благодаря двум факторам:

Использованию высокоскоростных элементов и параллельному выполнению большого числа операций. Специальности, связанные с повышением производительности отдельных микропроцессоров, мы рассматривали в предыдущих лекциях, а в этой лекции остановимся на вопросах распараллеливания обработки информации.

Существует несколько вариантов классификации систем параллельной обработки данных. По-видимому, самой ранней и наиболее известной является классификация архитектур вычислительных систем, предложенная в 1966 году М. Флинном. Элементов, команд или данных, обрабатываемая процессором. На основе числа потоков команд и потоков данных выделяются четыре класса архитектур: SISD, MISD, SIMD, MIMD. SISD (sINgleINsTRuctionsTReam / sINgledatasTReam) - одиночный поток команд и одиночный поток данных. К этому классу относятся прежде всего

классические последовательные машины, или, иначе, машины фон-неймановского типа. В таких машинах есть только один поток команд, все команды обрабатываются последовательно друг за другом и каждая команда инициирует одну операцию с одним потоком данных. Не имеет значения тот факт, что для увеличения скорости обработки команд и скорости выполнения арифметических операций процессор может использовать конвейерную обработку. В таком

понимании машины данного класса фактически не относятся к параллельным системам.

SIMD (sINgleINsTRuctionsTReam / multIPledatasTReam) - одиночный поток команд и множественный поток данных. Применительно к одному микропроцессору этот подход реализован в MMX- и SSE- расширениях современных микропроцессоров. Микропроцессорные системы типа SIMD состоят из большого числа идентичных процессорных элементов, имеющих собственную память. Все процессорные элементы в такой машине выполняют одну и ту же программу. Это позволяет выполнять одну арифметическую операцию сразу над многими данными - элементами вектора. Очевидно, что такая система, составленная из большого числа процессоров, может обеспечить существенное повышение производительности только на тех задачах, при решении которых все процессоры могут делать одну и ту же работу.

MISD (multIPeINsTRuctionsTReam / sINgledatasTReam) - множественный поток команд и одиночный поток данных. Определение подразумевает наличие в архитектуре многих процессоров, обрабатывающих один и тот же поток данных. Ряд исследователей к данному классу относят конвейерные машины.

MIMD (multIPeINsTRuctionsTReam / multIPledatasTReam) - множественный поток команд и множественный поток данных. Базовой моделью вычислений в этом случае является совокупность независимых процессов, эпизодически обращающихся к разделяемым данным. В такой системе каждый процессорный элемент выполняет свою программу достаточно независимо от других процессорных элементов. Архитектура MIMD дает большую гибкость: при наличии адекватной поддержки со стороны аппаратных средств и программного обеспечения MIMD может работать как однопользовательская система, обеспечивая высокопроизводительную обработку данных для одной прикладной задачи, как многопрограммная машина, выполняющая множество задач параллельно, и как некоторая комбинация этих возможностей. К тому же архитектура MIMD может использовать все преимущества современной микропроцессорной технологии на основе строгого учета соотношения стоимость/производительность. В действительности практически все современные многопроцессорные системы строятся на тех же микропроцессорах, которые можно найти в персональных компьютерах, рабочих станциях и небольших однопроцессорных серверах.

Как и любая другая, приведенная выше классификация несовершенна: существуют

машины, прямо в нее не попадающие, имеются также важные признаки, которые в этой классификации не учтены. Рассмотрим классификацию многопроцессорных и многомашинных систем на основе другого признака - степени разделения вычислительных ресурсов системы.

В этом случае выделяют следующие 4 класса систем:

- системы с симметричной мультипроцессорной обработкой (symmeTRicmultIProcessINg), или SMP-системы;
- системы, построенные по технологии неоднородного доступа к памяти (non-unIFORMmemoryaccess), или NUMA-системы;
- кластеры;
- системы вычислений с массовым параллелизмом (massivelyparallelprocessor), или MPP-системы.

Самым высоким уровнем интеграции ресурсов обладает система с симметричной мультипроцессорной обработкой, или **SMP-система**.

В этой архитектуре все процессоры имеют равноправный доступ ко всему пространству оперативной памяти и ввода/вывода. Поэтому SMP-архитектура называется симметричной. Ее интерфейсы доступа к пространству ввода/вывода и ОП, система управления кэш-памятью, системное ПО и т. п. построены таким образом, чтобы обеспечить согласованный доступ к разделяемым ресурсам. Соответствующие механизмы блокировки заложены и в шинном интерфейсе, и в компонентах операционной системы, и при построении кэша.

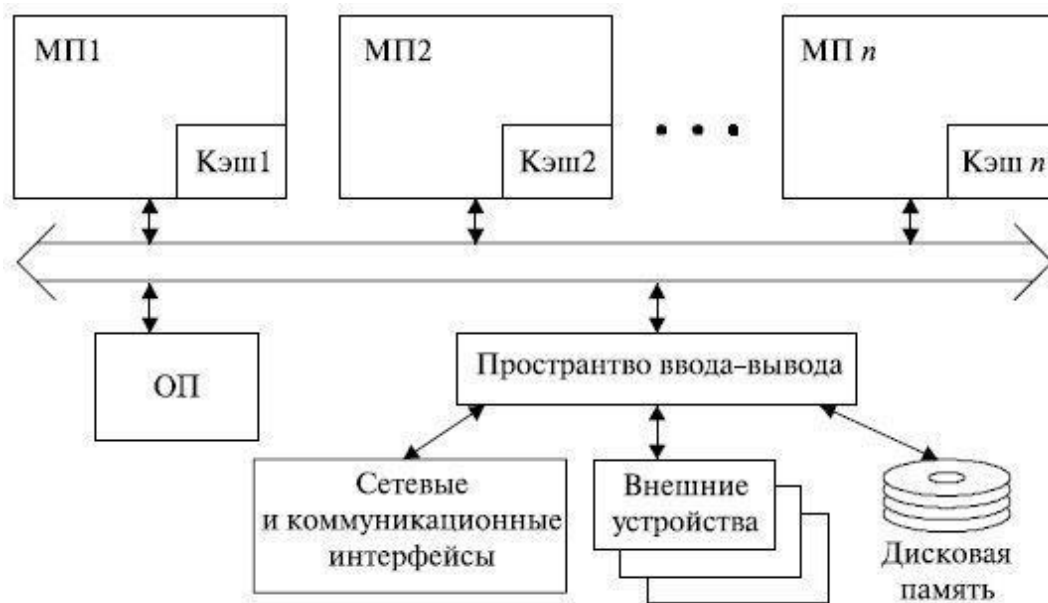


Рис. 1. Система с симметричной мультипроцессорной обработкой

С точки зрения прикладной задачи, **SMP-система** представляет собой единый вычислительный комплекс с вычислительными ресурсами, пропорциональными количеству процессоров. Распараллеливание вычислений обеспечивается операционной системой, установленной на одном из процессоров. Вся система работает под управлением единой ОС (обычно UNIX-подобной, но для Intel-платформ поддерживается WINdows NT).

ОС автоматически в процессе работы распределяет процессы по процессорным ядрам, оптимизируя использование ресурсов. Ядра задействуются равномерно, и прикладные программы могут выполняться параллельно на всем множестве ядер. При этом достигается максимальное быстродействие системы. Важно, что для синхронизации приложений вместо сложных механизмов и протоколов межпроцессорной коммуникации применяются

стандартные функции ОС. Таким образом, проще реализовать проекты с распараллеливанием программных потоков. Общая для совокупности ядер ОС позволяет с помощью служебных инструментов собирать статистику, единую для всей архитектуры. Соответственно, можно облегчить отладку и оптимизацию приложений на этапе разработки или масштабирования для других форм многопроцессорной обработки.

В общем случае приложение, написанное для однопроцессорной системы, не требует модификации при его переносе в мультипроцессорную среду. Однако для оптимальной работы программы или частей ОС они переписываются специально для работы в мультипроцессорной среде.

Сравнительно небольшое количество процессоров в таких машинах позволяет иметь одну централизованную общую память и объединить процессоры и память с помощью одной шины.

Сдерживающим фактором в подобных системах является пропускная способность магистрали, что приводит к их плохой масштабируемости. Причиной этого является то, что в каждый момент времени шина способна обрабатывать только одну транзакцию, вследствие чего возникают проблемы разрешения конфликтов при одновременном обращении нескольких процессоров к одним и тем же областям общей физической памяти. Вычислительные элементы начинают мешать друг другу. Когда произойдет такой конфликт, зависит от скорости связи и от количества вычислительных элементов. Кроме того, системная шина имеет ограниченное число слотов. Все это очевидно препятствует увеличению производительности при увеличении числа процессоров. В реальных системах можно задействовать не более 32 процессоров.

В современных микропроцессорах поддержка построения мультипроцессорной системы закладывается на уровне аппаратной реализации МП, что делает многопроцессорные системы сравнительно недорогими.

Так, для обеспечения возможности работы на общую магистраль каждый микропроцессор фирмы Intel начиная с PentiumPro имеет встроенную поддержку двухразрядного идентификатора процессора – APIC (Advanced Programmable Interrupt Controller). По умолчанию CPU с самым высоким номером идентификатора становится процессором начальной загрузки. Такая идентификация облегчает арбитраж шины данных в **SMP-системе**. Подобные средства мы видели и в МП Power4, где на аппаратном уровне поддерживается создание микросхемного модуля MCM из 4 микропроцессоров, включающего в совокупности 8 процессорных ядер.

Сегодня SMP широко применяют в многопроцессорных суперкомпьютерах и серверных приложениях. Однако если необходимо детерминированное исполнение программ в реальном масштабе времени, например, при визуализации мультимедийных данных, возможности сугубосимметричной обработки весьма ограничены. Может возникнуть ситуация, когда приложения, выполняемые на различных ядрах, обращаются к одному ресурсу ОС. В этом случае доступ получит только одно из ядер.

Остальные будут простаивать до высвобождения критической области.

Естественно, при этом резко снижается производительность приложений реального времени.

Исчерпание производительности системной шины в **SMP-системах** при доступе большого числа процессоров к общему пространству оперативной памяти и принципиальные ограничения шинной технологии стали причиной сдерживания роста производительности **SMP-систем**. На данный момент эта проблема получила два решения. Первое - замена системной шины на высокопроизводительный коммутатор, обеспечивающий одновременный неблокирующий доступ к различным участкам памяти. Второе решение предлагает технология NUMA.

Система, построенная по технологии NUMA, представляет собой набор узлов, каждый из которых, по сути, является функционально законченным однопроцессорным или

SMP-компьютером. Каждый имеет свое локальное пространство оперативной памяти и ввода/вывода. Но с помощью специальной логики каждый имеет доступ к пространству оперативной памяти и ввода/вывода любого другого узла (рис. 2). Физически отдельные устройства памяти могут адресоваться как логически единое адресное пространство - это означает, что любой процессор может выполнять обращения к любым ячейкам памяти, в предположении, что он имеет соответствующие права доступа. Поэтому иногда такие системы называются системами с распределенной разделяемой памятью (DSM - disTRibutedsharedmemory).

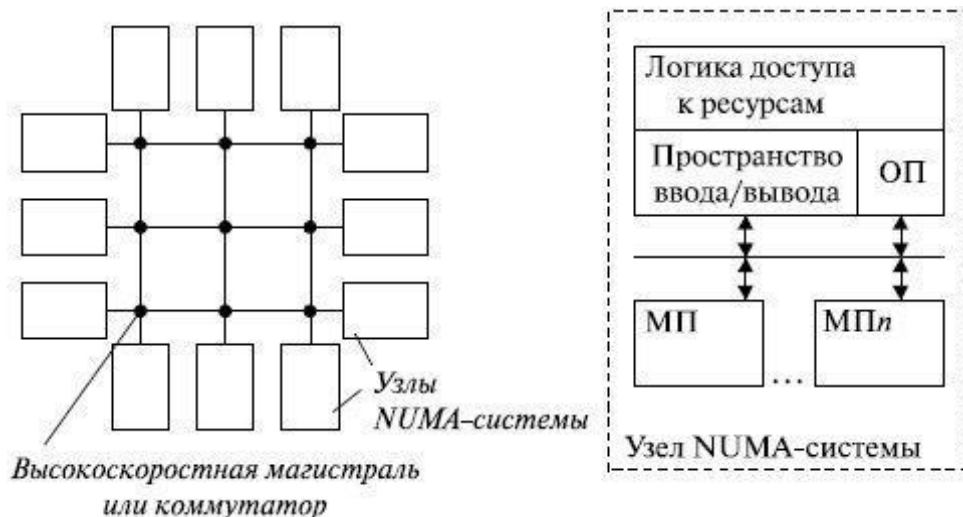


Рис. 2. Система, построенная по технологии неоднородного доступа к памяти

При такой организации память каждого узла системы имеет свою адресацию в адресном пространстве всей системы. Логика доступа к ресурсам определяет, к памяти какого узла относится выработанный процессором адрес. Если он не принадлежит памяти данного узла, организуется обращение к другому узлу согласно заложенной в логике доступа карте адресов. При этом доступ к локальной памяти осуществляется в несколько раз быстрее, чем к удаленной.

При использовании наиболее распространенного сейчас варианта сс-NUMA (cache-coherentNUMA - неоднородный доступ к памяти с согласованием содержимого кэш-памяти) обеспечивается кэширование данных оперативной памяти других узлов.

Обычно вся система работает под управлением единой ОС, как в SMP. Возможны также варианты динамического разделения системы, когда отдельные разделы системы работают под управлением разных ОС.

Довольно большое время доступа к оперативной памяти соседних узлов по сравнению с доступом к ОП своего узла в **NUMA-системах** на настоящий момент делает такое использование не вполне оптимальным.

Так что полной функциональностью **SMP-систем** **NUMA-компьютеры** на сегодняшний день не обладают. Однако среди систем общего назначения **NUMA-системы** имеют один из наиболее высоких показателей по масштабируемости и, соответственно, по производительности. На сегодня максимальное число процессоров в сс-NUMA-системах может превышать 1000 (серия OrigIN3000). Один из наиболее производительных суперкомпьютеров - Тера 10 - имеет производительностью 60 Тфлопс и состоит из 544 SMP-узлов, в каждом из которых находится от 8 до 16 процессоров Itanium 2.

Следующим уровнем в иерархии параллельных систем являются комплексы, также состоящие из отдельных машин, но лишь частично разделяющие некоторые ресурсы. Речь

идет о **кластерах**.

Кластер представляет собой систему из нескольких компьютеров (в большинстве случаев серийно выпускаемых), имеющих общий разделяемый ресурс для хранения совместно обрабатываемых данных (обычно набор дисков или дисковых массивов) и объединенных высокоскоростной магистралью (рис.3).

В кластерной системе некоторое распределенное приложение параллельно на нескольких узлах обрабатывает общий набор данных, как правило, таким образом, чтобы у пользователя возникла иллюзия работы на одной машине.

Обычно в кластерных системах не обеспечивается единая операционная среда для работы общего набора приложений на всех узлах кластера. То есть каждый компьютер кластера - это автономная система с отдельным экземпляром ОС и своими, принадлежащими только ей системными ресурсами: набором заведенных пользователей, системными буферами, областью свопинга и т. п. Приложение, запущенное на нем, может видеть только общие диски или отдельные участки памяти. На узлах кластера работают специально написанные для такой конфигурации приложения, параллельно обрабатывающие общий набор данных. На каждой из машин они представлены рядом процессов, программ, взаимодействующих с помощью кластерного программного обеспечения. Таким образом, кластерное ПО - это лишь средство для взаимодействия узлов и синхронизации доступа к общим данным. Кластер как параллельная система формируется на прикладном уровне, а не на уровне операционной системы.

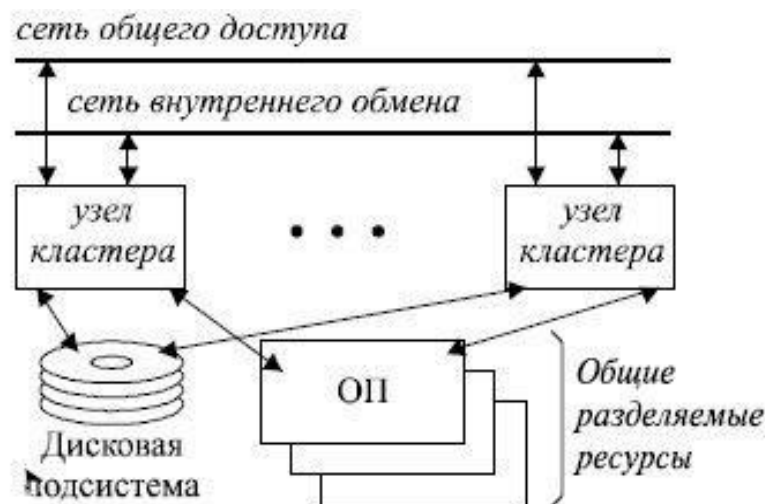


Рис. 3. Структура кластера

В настоящее время такие системы имеют две основные области применения: параллельные серверы баз данных и высоконадежные вычислительные комплексы. Рынок параллельных СУБД и есть фактически рынок кластеров приложений. Высоконадежные комплексы представляют собой группу узлов, на которых независимо друг от друга выполняются некоторые важные приложения, требующие постоянной, непрерывной работы. То есть в такой системе на аппаратном уровне фактически поддерживается основной механизм повышения надежности - резервирование. Причем узлы находятся в так называемом "горячем" резерве, и каждый из них в любой момент готов продолжить вычисления при выходе из строя какого-либо узла. При этом все приложения с отказавшего узла автоматически переносятся на другие машины комплекса. Такая система также формально является кластером, хотя в ней отсутствует параллельная обработка общих данных. Эти данные обычно монополично используются выполняемыми в рамках кластера приложениями и должны быть доступны для всех узлов.

Если в кластере его узлы разделяют некоторые ресурсы, то параллельные системы

другого класса - **системы вычислений с массовым параллелизмом (МРР)** - строятся из отдельных полностью независимых компьютеров, соединенных только высокоскоростной магистралью или коммуникационными каналами (рис. 4). Это могут быть либо просто несколько серийно выпускаемых UNIX-машин, соединенных с помощью высокопроизводительной сетевой среды, либо специально сконструированная система из отдельных функциональных блоков, объединенных коммутатором.

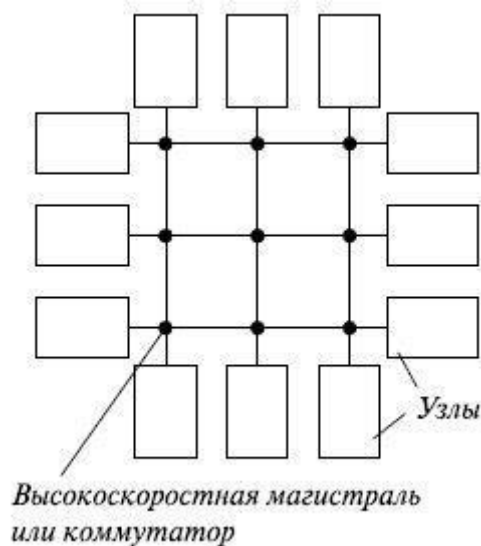


Рис. 4. Структура МРР-системы

В такой системе адресное пространство состоит из отдельных адресных пространств, которые логически не связаны между собой и доступ к которым не может быть осуществлен аппаратно другим процессором.

При этом для обмена данными используется механизм передачи сообщений между процессорами. Поэтому эти машины часто называют машинами с передачей сообщений. Пользователь может определить логический номер процессора, к которому он подключен, и организовать обмен сообщениями с другими процессорами.

На машинах МРР-архитектуры используются два варианта работы операционной системы. В одном из них полноценная операционная система работает только на управляющей машине (front-end); на каждом отдельном модуле функционирует сильно урезанный вариант ОС, обеспечивающий работу только расположенной в нем ветви параллельного приложения. Во втором варианте на каждом модуле работает полноценная, как правило, UNIX-подобная ОС, устанавливаемая отдельно.

Программирование в такой системе - достаточно сложная задача.

Она требует специального инструментария и особого системного программного обеспечения для работы параллельных приложений, которые ориентированы на функционирование параллельных процессов, распределенных по узлам МРР-системы, с обменом сообщениями между ними.

Повышение производительности машин с массовым параллелизмом путем увеличения в них числа процессоров имеет определенные ограничения. Чем большее число процессоров входит в состав МРР-системы, тем длиннее каналы передачи управления и данных, а значит, и тем меньше тактовая частота. Происшедшее возрастание нормы массивности для больших машин до 512 и даже 64К процессоров обусловлено не ростом размеров машины, а увеличением степени интеграции схем, позволившей за последние годы резко повысить плотность размещения элементов в устройствах. Топология сети межпроцессорного обмена

в такого рода системах может быть различной.

Главным преимуществом MPP-систем является их хорошая масштабируемость: в отличие от SMP-систем, здесь каждый процессор имеет доступ только к своей локальной памяти, в связи с чем не возникает необходимости в потактовой синхронизации процессоров. Практически все рекорды по производительности на сегодня устанавливаются на машинах именно такой архитектуры, состоящих из нескольких тысяч процессоров.

Основными недостатками систем данного типа являются следующие:

- отсутствие общей памяти заметно снижает скорость межпроцессорного обмена, поскольку нет общей среды для хранения данных, предназначенных для обмена между процессорами;
- требуется специальная техника программирования для реализации обмена сообщениями между процессорами;
- каждый процессор может использовать только ограниченный объем локального банка памяти;
- вследствие указанных архитектурных недостатков требуются значительные усилия для того, чтобы максимально задействовать системные ресурсы, следствием чего является высокая цена программного обеспечения для MPP-систем с отдельной памятью.

Подведем некоторые итоги, касающиеся областей применимости систем параллельной обработки данных различных типов.

SMP-системы потенциально обладают достаточными возможностями для обеспечения необходимой для большинства применений производительности: вполне естественно увеличивать число процессоров, а не ставить рядом еще один компьютер. Добавление одного процессора гарантированно увеличивает производительность, а добавление, например, узла в кластер адекватного ускорения не даст. Более того, в некоторых случаях общая производительность системы может даже упасть, когда узлы кластера начинают активно конкурировать за доступ к общим ресурсам, и взаимные блокировки сводят на нет преимущества параллельной обработки.

NUMA-системы создаются для вполне определенных целей - обеспечения масштабных расчетов. Системы, использующие эту архитектуру, прежде всего применяются для уникальных высококачественных и высокопроизводительных прикладных программ, требующих более восьми процессоров. Однако они имеют высокую стоимость и требуют уникального ПО (прикладные программы и ОС).

Для современных систем помимо вполне традиционных требований по производительности, масштабируемости, цене дополнительные высокие требования предъявляются к надежности их работы. Именно по этим соображениям вычислительные комплексы на основе кластеров или MPP-машин завоевывают все большую популярность.

MPP-системы обладают рядом преимуществ, главным из которых является лучшая среди всех рассмотренных архитектур масштабируемость. Именно поэтому MPP-компьютеры обычно используются при больших ресурсоемких вычислениях. Конечно, они применяются и при построении больших баз данных, и в отказоустойчивых вычислительных комплексах. Но здесь их использование довольно ограничено. Это отчасти связано с тем, что они все-таки дороже кластеров и имеют достаточно большую начальную цену. Кластер же можно построить из относительно дешевых машин произвольной конфигурации.

Приведенная классификация систем параллельной обработки данных достаточно условна. Разработчики вычислительных систем не проектируют машину какого-то специального класса, а стараются создать более производительную архитектуру. Кроме этого, сам пользователь может с использованием стандартных компонентов спроектировать комплекс, архитектурно и функционально наиболее подходящий для решения конкретной задачи.

Транспьютеры

Для построения многопроцессорных систем могут быть использованы специально

разработанные процессоры, называемые транспьютерами. Они были созданы в середине 1980-х годов фирмой INMOS Ltd (ныне - подразделение STMicroelectronics).

Транспьютер - это микропроцессор со встроенными средствами межпроцессорной коммуникации, предназначенной для построения многопроцессорных систем. Его название происходит от слов TRansfer (передатчик) и computer (вычислитель).

Транспьютер включает в себя средства для выполнения вычислений (ЦП, АЛУ с плавающей точкой, внутрикристальную память) и 4 канала для связи (линки) с другими транспьютерами и/или другими устройствами. Каждый линк представляет собой 2 однонаправленных последовательных канала передачи информации. Встроенный интерфейс позволяет подключать внешнюю память емкостью до 4 Гбайт (рис. 5).

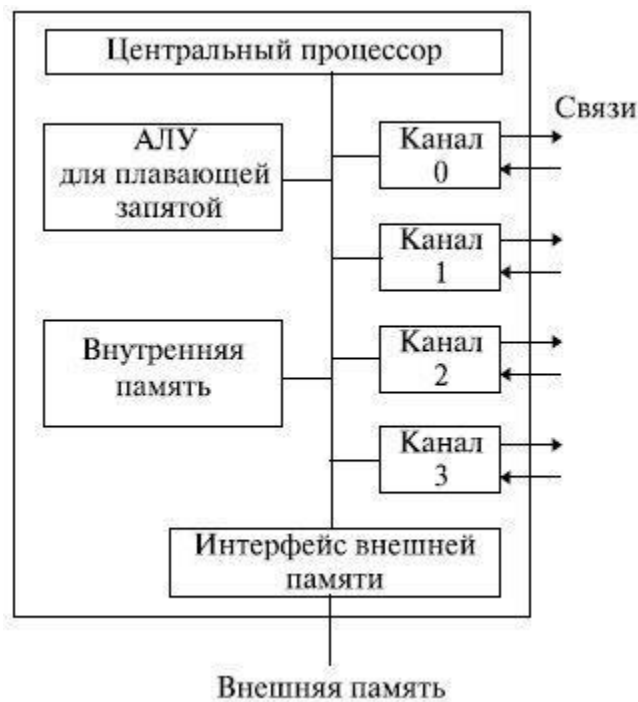


Рис. 5. Структура транспьютера

Многопроцессорная система может создаваться из набора транспьютеров, которые функционируют независимо и взаимодействуют через последовательные каналы связи (рис. 6).

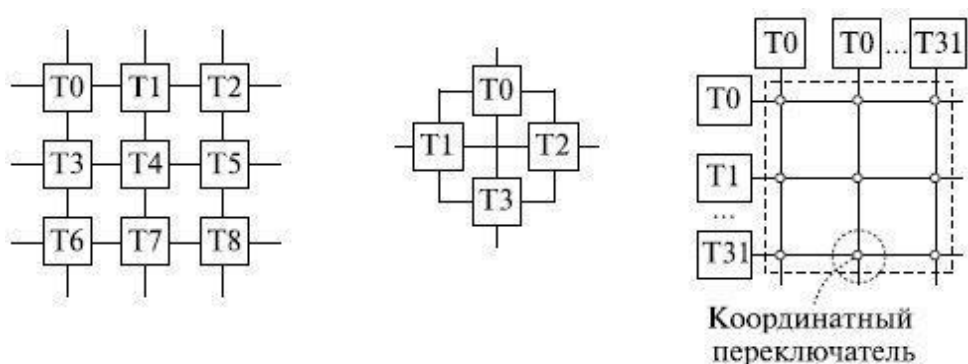


Рис. 6. Структуры многопроцессорных систем на базе транспьютеров

Организация транспьютеров и многопроцессорных систем на их основе базируется на языке Оссат, специально разработанном в начале 1980-х годов группой ученых из Оксфорда

под руководством Дэвида Мэяпо заданию компании INMOS в рамках работ по созданию транспьютеров. Отметим основные характеристики этого языка с точки зрения его применения для программирования транспьютерных систем:

- выполнение задачи разбивается на процессы, которые могут выполняться параллельно;
- размещение процессов не привязано к конкретному оборудованию: необязательно каждому процессу должен ставиться в соответствие свой транспьютер - несколько процессов могут выполняться на одном транспьютере;
- язык позволяет описать размещение процессов по оборудованию;
- язык позволяет описать, как эти процессы обмениваются между собой данными (какой процесс принимает информацию, от какого процесса и в каком объеме).

При передаче данных в линк процесс должен исполнить команду вывода. Процесс, исполнивший такую команду, задерживается до тех пор, пока все данные не будут переданы. Аналогично, при приеме данных из линка процесс должен исполнить команду ввода. При исполнении такой команды процесс блокируется до тех пор, пока буфер не будет заполнен данными. Взаимодействие с внешним устройством через линк позволяет транспьютеру синхронизовать свою деятельность с этими устройствами без использования механизма прерываний.

Использование такого подхода позволило организовать виртуальные каналы связи между процессами, которые могли размещаться как на единственном транспьютере, так и на нескольких транспьютерах, и виртуальные линки между процессами. Любой транспьютер может одновременно образовывать любое число параллельных процессов. Он имеет специальный планировщик, который производит распределение процессорного времени между этими процессами. Тем самым появляется возможность, имея всего лишь один транспьютер, написать параллельную программу, которая полностью выполняется на нем. Задача разбивается на ряд процессов, и все эти процессы параллельно протекают внутри одного транспьютера, периодически останавливаясь для получения данных друг от друга. Систему можно расширить другими транспьютерами и перенести на них ряд процессов. При этом нужно просто переопределить таблицу связей процессов, указав, на каком транспьютере теперь выполняется тот или иной процесс. Сама же программа изменений не претерпевает, а вычислительная мощность системы, естественно, увеличивается.

Помимо интересных возможностей, связанных с построением мультипроцессорных систем без привлечения дополнительного оборудования, в транспьютерах были реализованы идеи, направленные на повышение их вычислительной мощности. Среди них хотелось бы отметить то, что блок регистров транспьютера организован в виде стека. Это привело к использованию преимущественно безадресной системы команд, что обеспечило даже более высокую производительность, чем RISC-архитектура. Второй момент, который следует выделить, - это одновременное исполнение группы, в которую входило до 8 команд, что обеспечивало полную загрузку устройств процессора. И все это было реализовано в конце 1980-х годов, задолго до появления EPIC и Itanium.

На момент своего появления транспьютеры были самыми быстродействующими 32-разрядными микропроцессорами. В процессе своего недолгого развития их характеристики достигли следующих значений:

- производительность: 200 MIPS, 25 MFLOPS (на 64-разрядном процессоре с плавающей точкой);
- емкость внутрикристальной памяти: 16 Кбайт;
- скорость обмена по линку: 100 Мбит/с.

В середине 1990-х, в эпоху расцвета микропроцессоров этого семейства, фирма INMOS помимо собственно транспьютеров поставляла широкий набор трэмов (TRem - TRansputerExtensionModule) - устройств ввода-вывода с линком в качестве интерфейса. В частности, поставлялись трэмы, позволявшие подключить к транспьютеру через линк

адаптеры Ethernet или SCSI. В целях увеличения числа физических связей был разработан программируемый коммутатор, осуществляющий передачу сообщения с любого из 32 входов на любой из 32 выходов.

Транспьютеры успешно использовались в различных областях от встроенных систем до суперЭВМ. Однако технология транспьютеров серьезного развития не получила, так как начиная с PentiumPro в универсальные микропроцессоры введена возможность соединения процессоров в микропроцессорную систему, что обесценило главное преимущество транспьютеров - возможность построения многопроцессорных систем без дополнительных аппаратных затрат. В настоящее время транспьютеры не производятся, они вытеснены похожими разработками конкурентов, особенно Texas Instruments (TMS320) и Intel (80860).

Задание:

- 1) Составить структурную схему многопроцессорной системы архитектуры класса MIMD.
- 2) Составить структурную схему многопроцессорной системы с симметричной мультипроцессорной обработкой (SMP-система).
- 3) Составить структурную схему многопроцессорной системы, построенной по технологии неоднородного доступа к памяти (NUMA).
- 4) Составить структурную схему кластерной многопроцессорной системы.
- 5) Составить структурную схему системы вычислений с массовым параллелизмом (MPP). Составить структурную схему транспьютера.
- 6) Составить структурную схему многопроцессорных систем на базе транспьютеров.

**Лабораторная работа №18 Исследование структуры персонального компьютера
(2 часа)**

Содержание:

1. Исследование структуры персонального компьютера

Цель практического занятия:

В результате выполнения практического занятия студенты должны получить навык владения методами определения конфигурации и параметров устройств ПЭВМ, оценки производительности компьютера, а также научиться применять знания структуры ПЭВМ для решения задач педагогической деятельности

Теоретические сведения:

Персональные ЭВМ типа IBM PC совместимых имеют явно выраженную модульную структуру, что позволяет собирать из базовых модулей конфигурацию, соответствующую потребностям и возможностям пользователя.

Состав ПЭВМ, различаясь для конкретных экземпляров, имеет базовые компоненты, обязательные для любой модификации. *Обычно в любой ПЭВМ имеются следующие узлы:*

- процессор;
- материнская (системная) плата;
- оперативная память;
- видеоадаптер;
- жесткий диск;
- гибкий диск;
- корпус системного блока с блоком питания;
- монитор;
- клавиатура;
- мышь.

Определить конфигурацию ПЭВМ можно по данным, которые высвечиваются на экране монитора при его запуске в процессе тестирования компьютера базовой системой ввода-вывода (BIOS). Состав данных может меняться в зависимости от типа и версии BIOS.

Обычно выводится следующая информация:

- тип и версия BIOS;
- типа материнской платы;
- тип процессора (CPU Type).
- наличие сопроцессора (Co-Processor);
- частоту (ядра) процессора (CPU Clock);
- объем основной части оперативной памяти (Base Memoir);
- объем расширенной оперативной памяти (Extended Memory);
- объем кэш-памяти (Cache Memoir);
- параметры основного диска 1-го канала IDE (Primair Master);
- параметры вспомогательного диска 1-го канала IDE (Primair Slave);
- параметры основного диска 2-го канала IDE (Secondary Master);
- параметры вспомогательного диска 2-го канала IDE (Secondary Slave);
- а также вид дисплея, установки портов ввода-вывода, тип оперативной памяти и кэш-памяти.

Однако *пользоваться этими данными крайне неудобно*, поскольку информация появляется на экране монитора очень кратковременно (доли секунды), а в современных ПЭВМ эти сообщения могут вообще не выводиться. Для того чтобы ее посмотреть необходимо на начальном этапе процесса загрузки ОС зайти в BIOS, это означает запустить подпрограмму BIOS Setup.

Обычно *система предлагает зайти в BIOS* (иными словами, запустить BIOS Setup) после окончания самотестирования (POST), в момент начальной инициализации. В большинстве случаев на экране компьютера появляется надпись, которая предлагает войти в BIOS Setup (например, **Press DEL to enter SETUP**), как показано на рис. 1. Пользователю остается лишь выполнить инструкцию и нажать на клавишу Del (Delete) на клавиатуре компьютера, пока надпись не исчезла.

Иногда это сообщение не выводится. *Чтобы зайти в BIOS в этом случае нужно* включить компьютер, и, как только произойдет инициализация клавиатуры (это можно определить по миганию индикаторов Num Lock, Caps Lock и Scroll Lock), нажать на клавишу DEL несколько раз подряд. Если не получилось, то возможно, в вашей BIOS вызов BIOS Setup осуществляется при помощи других комбинаций клавиш. *Это могут быть такие клавиши и сочетания:*

- F1 – для отдельных версий Phoenix BIOS;
- F2 – для отдельных версий Phoenix BIOS и материнских плат Intel;
- F10 – для отдельных версий Phoenix BIOS;
- Esc;
- Ctrl + Alt + Esc;
- Ctrl + Alt + Ins;
- Ctrl + Alt.

Если эти комбинации клавиш не указываются при загрузке, их можно найти в руководстве к компьютеру или материнской плате. При этом клавиша Del остается наиболее распространенным вариантов. Иногда однократного нажатия клавиши или комбинации клавиш не хватает, чтобы зайти в BIOS. Лучше всего нажать клавишу или сочетание около 10-15 раз в течение примерно 10 секунд.

Напомню, что изменять параметры BIOS без крайней надобности не рекомендуется!

Итак, для определения состава и характеристик ПЭВМ и отдельных их устройств, проверки работоспособности и производительности ПЭВМ и ее компонент более удобно пользоваться диагностическими и тестовыми программами.

Эти программы, основанные на непосредственном обращении к аппаратным ресурсам ПЭВМ, обычно предназначены для использования под управлением операционной системы DOS. Запуск многих из таких программ ОС Windows может привести к получению

искаженных результатов. Наряду с ними существуют версии тестовых программ для Windows, такие как:

- утилита Windows XP/7 «Сведения о системе» (MSinfo32.exe);
- программа «System Information Viewer»;
- программа «Everest Ultimate», современная версия программы «Aida64 Extreme Edition»;
- программа «SiSoftware Sandra».

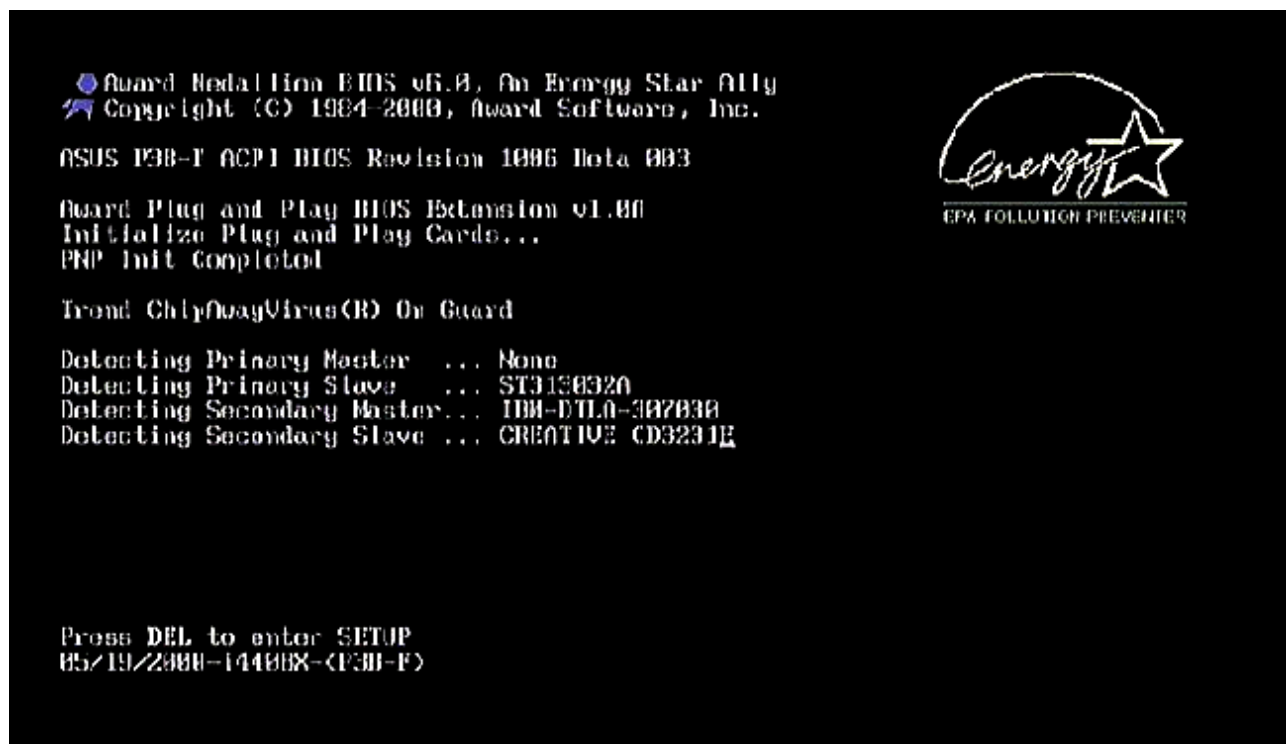


Рис. 1 – Загрузка ОС: момент начальной инициализации

Оценка производительности, как правило, производится в диагностических программах либо в относительном виде (проценты, индекс), либо в виде некоторого численного показателя, определяемого для конкретной совокупности проверочных заданий. Результаты таких измерений часто имеют вероятностный характер и для получения усредненного значения должны производиться многократно.

Рассмотрим возможности указанных программ.

Данные программы для анализа, диагностики и тестирования ПК проведут его самую полную "инвентаризацию", т.е. соберут и представят вам всю необходимую, подробную информацию об операционной системе ОС, программном обеспечении ПО, о всех установленных программных продуктах, аппаратных компонентах, комплектующих, производительности компьютера.

Для чего это нужно?

Во-первых, если Вы захотите модернизировать свой ПК, заняться ремонтом, купить комплектующие или, например, продать, то без наличия необходимой информации о компьютере ничего не получится.

Во-вторых, анализ состояния компьютера, его диагностика позволяют выявить неисправности, сбои в работе и главное – саму причину. Определить некоторые причины поломки иногда очень сложно и выходом может быть только использование диагностической программы. А если у вас эта программа будет записана на "реанимационный" компакт-диск, тогда даже при неработающей ОС можно узнать, какой именно компонент вышел из строя.

Некоторые программы диагностики могут не только информировать об установленных в ПК комплектующих, но и проверять отдельные компоненты на наличие "слабых мест", давать советы, подробные справочные сведения. Таким образом, если ваш ПК "виснет", отключается, дает сбой, ухудшается его производительность и стабильность работы, случилась поломка – значит, вам нужна всесторонняя, полноценная и качественная диагностика компьютера.

В-третьих, результат тестирования производительности вашего ПК можно сравнить с эталонными системами или другими аналогами, чтобы принять правильное решение о том, стоит ли проводить модернизацию компьютера, есть ли необходимость срочно покупать новое оборудование, устанавливать новое ПО.

Далее, программы анализа, диагностики и тестирования ПК позволяют получить множество необходимой и полезной информации для компьютера. Например, об актуальных версиях драйверов и установлены ли последние из них на вашем ПК, о появлении обновлений для различных устройств с прямыми ссылками на сайты разработчиков, советы по повышению производительности работы компьютера и многое, многое другое.

System Information Viewer (SIV) – бесплатная программа, показывающая самую разнообразную и очень подробную информацию о системе, локальной сети и аппаратном обеспечении компьютера. Ее вид представлен на рис. 2.

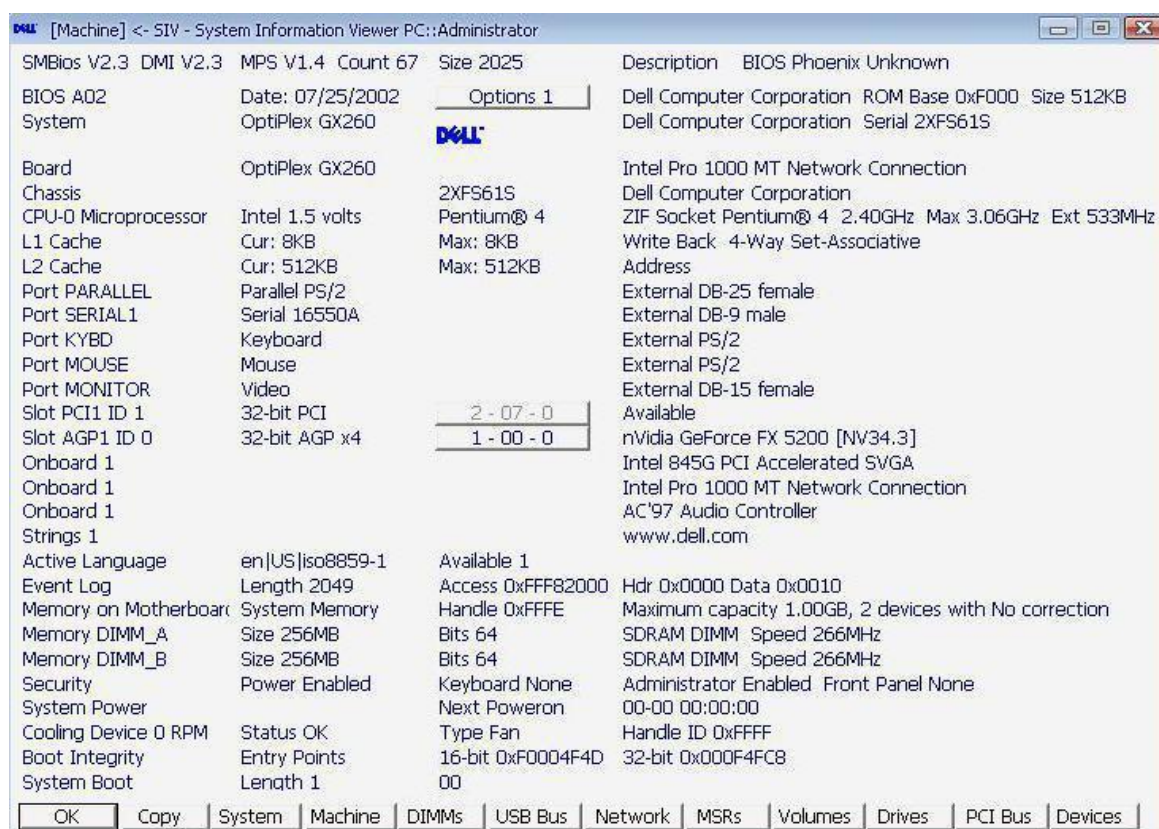


Рис. 2 – Вид программы System Information Viewer

Программа SIV вошла в список "Лучшие бесплатные программы". Она выдает все, что нужно знать о компьютере.

Использование System Information Viewer позволяет иметь практически всю информацию об имеющейся системе, о характеристиках локального компьютера и рабочих станций, об установленном оборудовании, операционной системе, программном обеспечении.

SIV (System Information Viewer) осуществляет отображение всех аппаратных подробностей о вашем компьютере, наблюдение за тем, как расходуются системные ресурсы

при работе с разными приложениями. Утилита в реальном времени показывает, сколько оперативной памяти свободно на данный момент и как используется файл подкачки. Программа обеспечивает поддержку контроля в режиме реального времени. Интерфейс программы представлен на русском языке и английском языке.

Платформой для программы являются все версии ОС Windows. Статус программы – бесплатная программа.

EVEREST Ultimate Edition или AIDA64 Extreme Edition

EVEREST Ultimate Edition или AIDA64 Extreme Edition – одна из самых лучших программ для детальной диагностики и тестирования аппаратных средств компьютера.

AIDA64 Extreme Edition выдает наиболее полную информации об установленном железе, обо всем аппаратном и программном обеспечении, осуществляет тесты различных модулей ПК и т.д. Отчеты сохраняются в HTML и TXT форматах и включают более 100 страниц сведений. Кроме того, программа производит сетевой аудит и настройку компьютера на оптимальную работу.

AIDA64 Extreme Edition позволяет найти наиболее эффективное решение для работы операционной системы; сравнить текущее состояние ОС с предыдущими вариантами или с другими системами; предоставляет самую полную информацию относительно установленного ПО, ОС, степени безопасности; осуществляет полную диагностику.

Программа AIDA64 Extreme Edition отображает физическую информацию о вашем CPU, материнской плате, жестком диске, оптическом драйве, системной шине, температурных сенсорах, установленных программах и системных событиях. Встроенная база данных программы имеет подробную информацию о более чем 22.000 компьютерных компонентах, каждое обновление версии программы добавляет поддержку новых аппаратных технологий (последние чипсеты, новые процессоры, платформы, расширенная поддержка памяти, поддержка видео от ATi, nVIDIA и т.п.). Таким образом, нет ничего такого, чего бы вы не смогли узнать о компьютере с помощью программы EVEREST.

В процессе оптимизации и тонкой настройки программа обеспечивает получение необходимой системной и оверклокерской информации, предоставляет продвинутые возможности мониторинга и диагностики аппаратного обеспечения для оценки эффекта, достигнутого применением тех или иных настроек. Тесты стабильности системы и тесты монитора, производительности центрального процессора, модуля вычислений с плавающей точкой, а также памяти помогают измерить реальную производительность системы и сравнить ее с ранее полученными результатами или с другими компьютерами. Более того, полная информация о программном обеспечении, операционной системе и настройках безопасности переводит продукт AIDA64 Extreme Edition в ранг всеобъемлющего средства диагностики, предоставляющего более 100 страниц информации о вашем компьютере.

Программа предоставляет информацию об аппаратном обеспечении:

- материнская плата и процессор – точная низкоуровневая информация о материнской плате, центральном процессоре и системе ввода-вывода BIOS, включая данные о чипсете, DMI, информацию о конфигурации AGP, список модулей памяти SPD, тайминг DRAM и поддерживаемый набор инструкций процессора;
- видеоадаптер и монитор – подробная информация о видеоадаптере, драйверах и мониторе, включая данные DDC, серийный номер монитора и определение поддерживаемых видеорежимов, низкоуровневые данные GPU, список поддерживаемых возможностей OpenGL и Direct3D;
- устройства хранения – информация обо всех жестких и оптических дисках, включая автоопределение IDE, мониторинг диска S.M.A.R.T., список устройств и данные разделов ASPI SCSI;
- сетевые адаптеры, мультимедиа, устройства ввода – исчерпывающая информация о сетевых адаптерах, звуковых картах, клавиатуре, мыши, игровых контролерах, включая определение MAC-адреса сетевой карты, список IP и DNS, мониторинг сетевого трафика,

информация о DirectSound, DirectMusic и DirectInput;

- другое железо – информация о PCI, PnP, PCMCIA и USB устройствах, портах связи, данные об управлении питанием, список ресурсов, используемых устройством, информация о принтерах.

Программа предоставляет информацию о программном обеспечении:

- операционная система – подробная информация о Windows, включая дату установки и лицензионный ключ, системные службы и список системных устройств, информация о процессах, список переменных окружения, список системных папок, системные файлы и содержимое журналов событий, список файлов AX и DLL, данные о безостановочной работе;

- сервер и дисплей – информация об общих сетевых ресурсах, список пользователей и групп, список залогиненных пользователей, список открытых файлов, список шрифтов и подробные данные о конфигурации рабочего стола;

- сетевая информация – большой объем информации о сетевом статусе, учетных записях почты и удаленного доступа, сетевых ресурсах и настройках интернет. Список сетевых маршрутов, Internet Explorer cookie и история посещенных веб-страниц;

- установленное программное обеспечение – подробная информация об установленных программах, запланированных задачах и запускающихся при запуске компьютера программах;

- безопасность – безопасность Windows. Информация о безопасности операционной системы, включая статус DEP (Data Execution Prevention), список установленных патчей и статус системы восстановления;

- приложения защиты – список межсетевых экранов, антишпионов и анти-троянов. Информация об антивирусном программном обеспечении, включая версию антивирусных баз.

Программа осуществляет диагностику:

- тест стабильности системы – возможность одновременного стрессового тестирования системной памяти, процессора и локальных дисков в реальном времени посредством полноценного интерфейса. Состояние нагрева системы контролируется путем постоянного мониторинга зависимости температуры процессора от его загруженности;

- Панель CPUID – панель EVEREST CPUID обеспечивает компактное отображение процессора, материнской платы, памяти и чипсета. Бесценная информация для разогнанных систем, динамическое обновление для поддержки технологий Enhanced Intel SpeedStep и AMD Cool'n'Quiet;

- мониторинг аппаратного обеспечения – данные сенсоров, включая температуру системы, центрального и графического процессоров, статус кулеров, мониторинг напряжения CPU, GPU, AGP и DRAM, статус дисков S.M.A.R.T. Поддержка модулей памяти Corsair Xpert;

- иконки сенсоров – функция позволяет отображать актуальную системную температуру и напряжение в системном трее (области уведомления);

- тесты производительности CPU и FPU – современные методы мультипоточного тестирования производительности для измерения быстродействия как старых, так и новейших процессоров. Список аналогичных систем для сравнения производительности;

- тесты производительности памяти – скорость чтения и записи, измерение задержек для стрессового тестирования памяти и КЭШа, включая возможность сравнения с аналогичными системами;

- модуль тестирования производительности дисков – измерение быстродействия жестких дисков, оптических приводов и USB-дисков. Графическое отображение быстродействия дисковой системы, измеренного на различных областях поверхности диска.

Программа выдает подсказки и рекомендации:

- обнаружение возможных проблем настройки и совместимости программного и

аппаратного обеспечения.

Программа предоставляет отчеты:

- мастер отчетов – простой в использовании метод создания отчета, применяя заранее настроенные шаблоны или вручную отобранные информационные блоки;
- форматы отчетов – три различных формата файла отчета: текст, HTML и уникальный MHTML. Отчеты в формате MHTML включают иконки и идеальны для распечатки;
- отправка по почте и распечатка отчетов – встроенный почтовый клиент для отправки отчетов поддерживает SMTP, MAPI и Outlook. Отображение отчета и распечатка одним кликом путем поддержки технологии IE4+.

Платформой для программы являются все версии ОС Windows. Статус программы: условно бесплатная, ограничение в работе программы 30 дней. Интерфейс: многоязычный, включая русский язык.

SiSoftware Sandra

SiSoftware Sandra – сокращение от System ANalyser, Diagnostic and Re-orting Assistant. Это информационная и диагностическая программа, которая предоставляет практически всю информацию о вашем аппаратном и программном обеспечении (включая недокументированную), которая вам может понадобиться.

SiSoftware Sandra одна из лучших, широко известная и конкурентно способная программа для системной диагностики, эталонного тестирования компьютера, получения самой обширной, максимально объективной информации относительно имеющихся аппаратных средств и установленного ПО. Программа объединяет возможности для сравнения производительности вашего процессора или оперативной памяти с более или менее производительными аналогами (как на высоком, так и на низком уровне), имеет богатую информационную и постоянно обновляющуюся базу железа, может работать в режиме клиент/сервер для анализа не только локального ПК, но и всех компьютеры локальной сети.

В программе SiSoftware Sandra удобный, продуманный многоязычный, в т.ч. русский интерфейс, огромное количество полезных модулей. Используйте программу для удаленного анализа, тестирования и диагностики компьютеров, серверов, КПК, смартфонов, небольших домашних и офисных сетей, а также для сетей предприятий.

Программа SiSoftware Sandra имеет несколько доступных версий:

- Sandra Lite – бесплатна для личного пользования, а также для использования в образовательных целях, не имеет ограничений по времени работы, напоминающих окон, и т.п.;
- Sandra Advanced – для производителей;
- Sandra Professional – коммерческая;
- Sandra Engineer – эксплуатация в коммерческих целях;
- Sandra Enterprise – коммерческая;
- Sandra Legacy – для энтузиастов.

Программа Sandra выдает информацию:

- о материнской плате/чипсете/системных мониторах;
- о процессоре и BIOS;
- об APM & ACPI (Advanced Power Management);
- об устройствах и шинах PCI(e), AGP, CardBus, PCMCIA;
- о видеосистеме (монитор, видеокарта, видео BIOS, и т.д.);
- о OpenGL;
- о DirectX (DirectDraw, Direct3D, DirectSound (3D), DirectMusic, DirectPlay, DirectInput);
- о клавиатуре и мыши, о джойстике;
- о звуковой карте (wave, midi, aux, mix);

- об устройствах MCI (mpeg, avi, seq, vcr, video-disc, wave);
- о принтерах;
- о Windows и ее памяти;
- о шрифтах (Raster, Vector, TrueType, OpenType);
- о модеме/ISDN TA;
- о сети и об IP-сети;
- о WinSock & Интернет-безопасности;
- о дисках (съемные жесткие диски, CD-ROM/DVD, RamDrives, и т.д.);
- о портах (последовательный, параллельный);
- об удаленных соединениях (Dial-Up, Internet);
- об объектах OLE;
- информация о процессах и потоках;
- о модулях (DLL, DRV);
- о службах и драйверах;
- об интерфейсах SCSI, ATA/ATAPI;
- об источниках информации;
- о смарт- и СИМ-картах.

Программа Sandra осуществляет:

- арифметический тест процессора;
- мультимедийный тест процессора (включая MMX, MMX Enh, 3DNow!, 3DNow!

Enh, SSE(2));

- тест файловой системы (съемные диски, жесткие диски, сетевые диски, RamDrives);

- тест съемных/флэш накопителей;
- тест CD-ROM/DVD;
- тест пропускной способности памяти;
- тест кэш и памяти;
- тест пропускной способности сети;
- тест соединения с Интернет и тест скорости Интернет;
- использование аппаратных прерываний;
- использование каналов DMA;
- использование портов ввода/вывода;
- использование диапазона памяти;
- использование перечислителя Plug & Play;
- настройки аппаратного реестра;
- настройки окружения;

Программа Sandra проверяет:

- зарегистрированные типы файлов;
- ключевые приложения (веб-браузер, e-mail, новости, антивирус, брандмау-эр, и

т.д.);

- установленные приложения;
- установленные программы;
- приложения меню Пуск;
- программы и библиотеки на диске;
- установленные веб-пакеты (ActiveX, классы Java);
- журнал событий;

Программа Sandra имеет:

- мастер стресс-тестирования (тестирование стабильности компьютера);
- мастер соединений (соединение с удаленными компьютерами, КПК, смарт-фонами и другими устройствами);

- мастер обобщенного индекса производительности (оценка общей производительности компьютера);
 - мастер создания отчетов (сохранение, печать, отправка по факсу или e-mail в форматах CIM (SMS/DMI), HTML, XML, RPT или TEXT);
 - мастер увеличения производительности (настройка компьютера);
 - мастер мониторинга окружения (температуры, напряжения, скорости вентиляторов, термостойкость системы охлаждения, и т.д.);
 - мастер обновлений для автоматического обновления версий;
 - обширный файл помощи, содержащий более 500 подсказок;
 - подробную он-лайн документацию (HTML) с базой вопросов и ответов.
- Платформой для программы являются все версии ОС Windows.

Задание:

3.1. Практическое ознакомление со структурой системной платы ПЭВМ и ее устройствами, размещенными в системном корпусе.

Задание 3.1.1. Практически изучить состав и устройство системной платы. Для этого выполнить следующие операции:

- 1) Снимите крышку учебного системного блока ПК.
- 2) Приведите в отчете перечень устройств, находящихся в данном системном блоке.
- 3) Воспользовавшись приложениями А и Б к крактическому занятию №4, в которых приведены описание материнской платы и ее компонентов, их внешний вид, самостоятельно ознакомьтесь со структурой платы и ее компонентами, записав в отчет следующие данные:
 - тип платы (форм-фактор);
 - тип процессора и вид его подключения;
 - количество разъемов (слотов) для оперативной памяти и ее тип и количество модулей памяти;
 - тип питания материнской платы, количество pin;
 - количество и тип слотов PCI;
 - количество и тип слотов для подключения внешней памяти;
 - чипсеты платы;
 - наличие контролеров внешних устройств и их предназначение;
 - название системных устройств компьютера, имеющих на материнской плате.
- 4) Сделайте выводы о проделанной работе.

3.2. Определение конфигурации и производительности персонального компьютера с помощью встроенной в ОС Windows утилиты «Сведения о системе» (MSinfo32.exe).

Задание 3.2.1. Определить конфигурацию исследуемой рабочей ПЭВМ. Для этого выполнить следующие операции:

- 1) Загрузить исследуемую рабочую ПЭВМ в режиме Windows.
- 2) Запустить утилиту «Сведения о системе». Для этого надо выбрать команду Сведения о системе в меню Пуск / Все программы / Стандартные / Служебные. Либо набрать и запустить утилиту MSinfo32.exe в командной строке или строке Выполнить ...
- 3) Ознакомиться со структурой окна программы. Системная информация на панели обзора разделена на пять категорий: корневой узел Сведения о системе и четыре основных дополнительных узла: ресурсы аппаратуры; компоненты; программная среда; параметры обозревателя.
- 4) С помощью информации, представляемой утилитой MSinfo32.exe, заполнить данными соответствующие ячейки таблицы конфигурации ПЭВМ (см. табл. 1).
- 5) Провести тестирование производительности исследуемой ПЭВМ.
- 6) Сделайте выводы о проделанной работе.

Таблица 1 – Конфигурация ПЭВМ

Характеристики процессора	Значение характеристики		
	Утилита MSinfo32.exe	Программа AIDA64	Программа Sandra
Процессор			
Изготовитель			
Тип			
Семейство			
Модель			
Частота ядра			
Частота шины			
Индексы производительности			
Системная плата			
Чипсет системной платы: - северный мост; - южный мост.			
Другие системные устройства: - контролеры; -			
Последовательный порт (тип/кол-во)			
Параллельный порт (тип/кол-во)			
BIOS			
Видеоадаптер/объем памяти			
Аудиоадаптер			
Сетевой адаптер			
Оперативная память			
Модель или тип оперативной памяти			
Количество логических банков памяти			
Объем оперативной памяти			
Скорость передачи оперативной памяти			
Объем кэша L1			
Скорость передачи кэша L1			
Объем кэша L2			
Скорость передачи кэша L2			
Жесткий диск			
Изготовитель			

Характеристики процессора	Значение характеристики		
	Утилита MSinfo32.exe	Программа AIDA64	Программа Sandra
Тип			
Объем			
Физическая организация			
Логическая геометрия (CHS, LBA)			
Режим обмена			
Скорость вращения шпинделя			
Время поиска			
Скорость передачи			
Логические диски			
Периферийные устройства			
Клавиатура			
Манипулятор мышь			
Монитор			
Принтер			

3.3. Определение конфигурации персонального компьютера и характеристик устройств аппаратного обеспечения с помощью тестовой программы System Information Viewer (просмотрщика системной информации).

Задание 3.3.1. Определить конфигурацию исследуемой рабочей ПЭВМ с помощью программы SIV. Для этого выполнить следующие операции:

1) Разархивировать файл SIV_System_Information_Viewer_4.27_Portable.rar, находящейся в папке с практическим занятием, с помощью контекстно-зависимого меню, выбрав команду Извлечь в папку SIV_System_Information_Viewer_4.27_Portable\.

2) Запустить программу с помощью файла SIV32.exe или SIV64.exe в зависимости от разрядности операционной системы рабочей ПЭВМ.

3) Установить в программе русскоязычный интерфейс, набрав команду Copy/ Language/ Russian.

4) Ознакомиться со структурой окна программы и ее меню.

5) С помощью информации, представляемой программой SIV, найти и заполнить данными соответствующие ячейки таблицы конфигурации ПЭВМ (см. табл. 1).

6) Сделайте выводы о проделанной работе.

3.4. Определение конфигурации и производительности персонального компьютера и характеристик устройств аппаратного обеспечения с помощью программы тестирования AIDA64 Extreme Edition.

Задание 3.4.1. Определить конфигурацию исследуемой рабочей ПЭВМ с помощью программы AIDA64. Для этого выполнить следующие операции:

1) Разархивировать файл Aida64extreme230.zip, находящейся в папке с практическим занятием, с помощью контекстно-зависимого меню, выбрав команду Извлечь в папку Aida64extreme230\.

2) Запустить программу с помощью файла Aida64.exe.

3) Воспользовавшись приложением В ознакомиться со структурой окна программы и ее меню.

4) С помощью информации, представляемой программой Aida64, найти и заполнить данными соответствующие ячейки таблицы конфигурации ПЭВМ (см. табл. 1).

5) Провести тестирование производительности исследуемой ПЭВМ и ее основных устройств с помощью программы Aida64.

6) Сделайте выводы о проделанной работе.

3.5. Определение конфигурации и производительности персонального компьютера и характеристик устройств аппаратного обеспечения с помощью диагностической программы SiSoftware Sandra.

Задание 3.5.1. Определить конфигурацию исследуемой рабочей ПЭВМ с помощью программы Sandra. Для этого выполнить следующие операции:

1) Разархивировать файл SiSoftware.rar, находящейся в папке с практическим занятием, с помощью контекстно-зависимого меню, выбрав команду Извлечь в папку SiSoftware\.

2) Запустить программу с помощью файла Sandra.exe.

3) Воспользовавшись приложением В ознакомиться со структурой окна программы и ее меню.

4) С помощью информации, представляемой программой SiSoftware Sandra, найти и заполнить данными соответствующие ячейки таблицы конфигурации ПЭВМ (см. табл. 1).

5) Провести тестирование производительности исследуемой ПЭВМ и ее основных устройств с помощью программы Sandra.

6) Сделайте выводы о проделанной работе.

3.6. Разработка структурной схемы исследуемой ПЭВМ.

Задание 3.6.1. Составить структурную схему ПЭВМ. Для этого выполнить следующие операции:

1) Изобразить в отчете типовую структурную схему системной платы, приведенную в лекции №4.

2) Нанести на эту схему обозначение типов (моделей) устройств, имеющихся на системной плате, исследуемого компьютера (из таблицы 1).

3) Внести изменения в реальной структурной схеме, связанные с интерфейсами периферийных устройств.

4) Добавить к структурной схеме системной платы периферийные устройства исследуемого компьютера.

5) Представить в отчете отредактированную структурную схему исследуемой ПЭВМ.

6) Сделайте выводы о проделанной работе.

3.7. Исследование конфигурации, производительности и разработка структурной схемы домашней ПЭВМ.

Задание 3.7.1. С помощью тестовых и диагностических программ в часы самостоятельной работы провести исследование конфигурации домашнего компьютера, составить его структурную схему, оценить его производительность.

Контрольные вопросы:

1) Состав типовой ПЭВМ.

2) Структурная схема типовой ПЭВМ.

3) Структурная схема типовой системной платы ПЭВМ.

4) Порядок определения конфигурацию ПЭВМ при его запуске в процессе тестирования компьютера базовой системой ввода-вывода BIOS.

5) Перечень тестовых программ под ОС Windows для диагностирования аппаратного обеспечения компьютера.

6) Предназначение и краткая характеристика утилиты Windows XP/7 «Сведения о системе» (MSinfo32.exe).

7) Предназначение и краткая характеристика программы «System Information Viewer».

8) Предназначение и краткая характеристика программы «Aida64 Extreme Edition».

- 9) Предназначение и краткая характеристика программы «SiSoftware Sandra».
- 10) Порядок определения конфигурации персонального компьютера с помощью встроенной в ОС Windows утилиты «Сведения о системе» (MSInfo32.exe).
- 11) Порядок определения производительности персонального компьютера с помощью встроенной в ОС Windows утилиты «Сведения о системе» (MSInfo32.exe).
- 12) Порядок определения конфигурации персонального компьютера и характеристик устройств аппаратного обеспечения с помощью тестовой программы System Information Viewer.
- 13) Порядок определения конфигурации персонального компьютера и характеристик устройств аппаратного обеспечения с помощью программы тестирования AIDA64 Extreme Edition.
- 14) Порядок определения производительности персонального компьютера и устройств аппаратного обеспечения с помощью программы тестирования AIDA64 Extreme Edition.
- 15) Порядок определения конфигурации персонального компьютера и характеристик устройств аппаратного обеспечения с помощью диагностической программы SiSoftware Sandra.
- 16) Порядок определения производительности персонального компьютера и устройств аппаратного обеспечения с помощью диагностической программы SiSoftware Sandra.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания
по организации и проведению самостоятельной работы
по дисциплине **«АППАРАТНЫЕ СРЕДСТВА ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКИ»**
для студентов специальности 10.05.01 Компьютерная безопасность
специализация «Информационно-аналитическая и техническая экспертиза
компьютерных систем»

**Ставрополь
2025**

ВВЕДЕНИЕ

Целью изучения дисциплины «Аппаратные средства вычислительной техники» является формирование общепрофессиональной компетенции студента по специальности 10.05.01 Компьютерная безопасность. Задачами освоения дисциплины «Аппаратные средства вычислительной техники» являются:

- изучение аппаратных средств ЭВМ различных типов, устройств ввода - вывода данных, периферийных устройств и технологических аспектов их создания;
- изучение арифметических основ вычислительной техники;
- отслеживание тенденции дальнейшего совершенствования аппаратных средств вычислительных и телекоммуникационных систем, а также их интеграции;
- анализ технологических аспектов создания и внедрения микропроцессоров, чипсетов и микропроцессорной техники;
- исследование системных, периферийных и локальных аппаратных интерфейсов
- изучение методов и средства проектирования цифровых узлов ЭВМ.

Общая характеристика самостоятельной работы студента

Основными видами учебной работы по достижению результатов освоения дисциплины являются лекции, практические задания и самостоятельная работа студентов (СРС).

На лекциях раскрываются основные положения и понятия курса, формируются знания в области документооборота.

На лабораторных работах формируются умения, необходимые для решения задач профессиональной деятельности.

Приступая к изучению учебной дисциплины, необходимо ознакомиться с учебной программой, получить в библиотеке рекомендованные учебные пособия, а также получить у преподавателя в электронном виде конспекты лекций, методические рекомендации к практическим заданиям, завести новую тетрадь для конспектирования лекций и выполнения практических и работ.

Для изучения дисциплины предлагается список основной и дополнительной литературы. Основная литература предназначена для обязательного изучения, дополнительная – поможет более глубоко освоить отдельные вопросы, подготовить исследовательские задания и выполнить задания для самостоятельной работы.

В ходе лекционных занятий студент обязан осуществлять конспектирование учебного материала, особое внимание, обращая на категории, формулировки, раскрывающие содержание тех или иных понятий, физических явлений, процессов, эффектов. В рабочих конспектах желательно оставлять поля, на которых следует делать пометки из рекомендованной литературы, дополнять материал прослушанной лекции, формулировать научные выводы и практические рекомендации. Студент имеет право задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

Самостоятельная работа студентов над материалом учебной дисциплины является неотъемлемой частью учебного процесса и должна предполагать углубление знания учебного материала, излагаемого на аудиторных заданиях, и приобретение дополнительных знаний по отдельным вопросам самостоятельно.

Основными видами самостоятельной работы студентов по учебной дисциплине являются:

- самостоятельное выполнение заданий по заданным темам;
- самостоятельное изучение учебного материала по заданным темам;
- подготовка к практическим заданиям, оформление отчета по выполненному практическому занятию и подготовка к собеседованию по результатам работы.

Основными методами самостоятельной работы студентов являются:

- 1) для овладения знаниями:
 - чтение текста (конспекта лекций, учебника, дополнительной литературы) и конспектирование текста;
 - работа с нормативными документами;
 - поиск информации в Интернет;
- 2) для закрепления и систематизации знаний:
 - работа с конспектом лекции (обработка текста);
 - повторная работа над учебным материалом (учебника, дополнительной литературы, слайдами);
 - составление плана и тезисов ответа или графическое изображение структуры ответа;
 - составление таблиц для систематизации учебного материала;
 - изучение нормативных документов;
 - ответы на контрольные вопросы;
- 3) для формирования умений:
 - подготовка к практическим заданиям;
 - решение задач и практических заданий;
 - подготовка отчетов по практическим заданиям;
 - рефлексивный анализ профессиональных умений.

В случае пропуска учебного задания студент может воспользоваться содержанием различных блоков учебно-методического комплекса (лекции, практические задания, контрольные вопросы и т.д.) для самоподготовки и освоения темы. Для самоконтроля необходимо использовать вопросы и задания, предлагаемые к практическим заданиям, а также варианты тестовых заданий.

Технологическая карта самостоятельной работы студента

Коды реализуемых компетенций, индикатора(ов)	Вид деятельности студентов	Средства и технологии оценки	Объем часов, в том числе		
			СРС	Контактная работа с преподавателем	Всего
3 семестр					
ОПК-4 ИД-1 ОПК-4 ИД-2 ОПК-4 ИД-3 ОПК-4	Подготовка к коллоквиуму	коллоквиум	9.50	0.50	10.00
ОПК-4 ИД-1 ОПК-4 ИД-2 ОПК-4 ИД-3 ОПК-4	Подготовка к лекции	собеседование	1.90	0.10	2.00
ОПК-4 ИД-1 ОПК-4 ИД-2 ОПК-4 ИД-3 ОПК-4	Подготовка реферата, доклада	доклад	14.25	0.75	15.00
Итого			25.65	1.35	27.00

Порядок выполнения самостоятельной работы

Методические рекомендации по работе с учебной литературой

При работе с книгой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой – это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того, насколько осознанно читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют четыре основные установки в чтении научного текста:

- информационно-поисковый (задача – найти, выделить искомую информацию)
- усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить, как сами сведения, излагаемые автором, так и всю логику его рассуждений)
- аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)
- творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

Основные виды систематизированной записи прочитанного:

Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;

Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;

Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;

Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;

Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

Методические рекомендации по составлению конспекта:

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта;
2. Выделите главное, составьте план;
3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора;
4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.
5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

Вопросы для коллоквиумов

Базовый уровень:

- 1) Определение системы счисления.
- 2) Общие правила вычисления во всех позиционных системах счисления.
- 3) Правило записи произвольного числа в P -ичной позиционной системе счисления.
- 4) Порядок перевода чисел из одной позиционной системы счисления в другую.
- 5) Алгоритм перевода двоичного числа в другую систему счисления, основанием которой является степень двойки.
- 6) Алгоритм перевода чисел из системы счисления с основанием P в десятичную систему счисления.
- 7) Принципы внутреннего представления целых чисел в памяти ЭВМ.
- 8) Порядок использования дополнительных и обратных кодов.
- 9) Алгоритм получения дополнительного двоичного кода.
- 10) Принципы внутреннего представления в памяти ЭВМ вещественных чисел.
- 11) Типы представления вещественных чисел в ЭВМ.
- 12) Устройство накопителя на жестких магнитных дисках.
- 13) Характеристика структурной схемы накопителя на жестких магнитных дисках.
- 14) Характеристика гермоблока.
- 15) Характеристика магнитных дисков.
- 16) Характеристика блока магнитных головок.
- 17) Характеристика контроллера накопителя на жестких магнитных дисках.
- 18) Геометрия жесткого диска и его логическая структура.
- 19) Адресация данных в накопителях на жестких магнитных дисках.
- 20) Принцип зонно-секционной записи.
- 21) Характеристика видов организации данных на жестком диске.
- 22) Порядок обмена данными в накопителе на жестких магнитных дисках.

23) Принцип позиционирования магнитных головок.

24) Характеристика технологии SMART.

Повышенный уровень:

25) Порядок исследования внутренней структуры и логики функционирования 4-х битного статического ОЗУ с помощью генератора слова.

26) Порядок исследования внутренней структуры и логики функционирования ячейки постоянной памяти, программируемой путем пережигания перемычек.

27) Принципы построения и характеристика логики функционирования устройств ввода-вывода.

28) Порядок исследования внутренней структуры и логики функционирования цепи формирования сигнала разрешения выбора адреса порта УВВ.

29) Порядок выявления (локализации) ошибок в логических схемах с помощью логического преобразователя.

30) Порядок исследования внутренней структуры и логики функционирования цепи формирования адресов УВВ на базе ИМС 74154 с помощью генератора слова и логических пробников.

31) Порядок исследования внутренней структуры и логики функционирования одnorазрядного шинного формирователя.

32) Порядок исследования внутренней структуры и логики функционирования четырехразрядного шинного формирователя на базе ИМС 74244 из библиотеки EWB с помощью генератора слова и семисегментных индикаторов.

33) Порядок синтеза внутренней структуры и исследования логики функционирования типового устройства ввода-вывода.

34) Предназначение цифровых компараторов и устройств контроля четности (нечетности).

35) Принципы построения и характеристика логики функционирования цифровых компараторов.

36) Принципы построения и характеристика логики функционирования устройств контроля четности (нечетности).

37) Предназначение и классификация устройств управления ЭВМ.

38) Структура и принципы функционирования устройства управления схемного типа.

39) Структура и принципы функционирования микропрограммного устройства управления.

40) Порядок синтеза внутренней структуры и исследования логики функционирования одnorазрядного цифрового компаратора с помощью логического преобразователя и генератора слова.

41) Порядок исследования внутренней структуры и логики функционирования устройства формирования бита четности (нечетности) четырехразрядного кода с помощью генератора слова.

42) Порядок исследования логики функционирования устройства формирования бита четности (нечетности) на основе ИМС 742800 из библиотеки EWB с помощью генератора слова.

43) Порядок реализации устройства микропрограммного управления с помощью генератора слова.

1. Критерии оценивания компетенций

Оценка «отлично» выставляется студенту, если он на высоком уровне: понимает арифметические основы вычислительной техники; владеет навыками обработки экспериментальной и теоретической физической информации о состоянии ЭВМ; применяет методы и средства проектирования цифровых узлов ЭВМ.

Оценка «хорошо» выставляется студенту, если на среднем уровне: понимает арифметические основы вычислительной техники; владеет навыками обработки

экспериментальной и теоретической физической информации о состоянии ЭВМ; применяет методы и средства проектирования цифровых узлов ЭВМ.

Оценка «удовлетворительно» выставляется студенту, если на минимальном уровне: понимает арифметические основы вычислительной техники; владеет навыками обработки экспериментальной и теоретической физической информации о состоянии ЭВМ; применяет методы и средства проектирования цифровых узлов ЭВМ.

Оценка «неудовлетворительно» выставляется студенту, если он с грубыми ошибками: понимает арифметические основы вычислительной техники; владеет навыками обработки экспериментальной и теоретической физической информации о состоянии ЭВМ. анализируемой информации; применяет методы и средства проектирования цифровых узлов ЭВМ.

2. Описание шкалы оценивания

Максимально возможный балл за весь текущий контроль устанавливается равным 55. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

Уровень выполнения контрольного задания	Рейтинговый балл (в % от максимального балла за контрольное задание)
Отличный	100
Хороший	80
Удовлетворительный	60
Неудовлетворительный	0

3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедура проведения данного оценочного мероприятия включает в себя:

- устный ответ по вопросам из указанных тем;

Предлагаемые студенту вопросы позволяют проверить ОПК-4 (ИД-1 ОПК-4, ИД-2 ОПК-4, ИД-3 ОПК-4) компетенции.

Для подготовки к данному оценочному мероприятию необходимо изучить рекомендованные источники и научную литературу и на их основании подготовиться к указанным в методических материалах вопросам для обсуждения. Для подготовки к данному оценочному мероприятию необходимо 10 часов, в течение которых обучающийся изучает научную и учебную литературу, готовится к сдаче коллоквиума.

При подготовке к ответу студенту предоставляется право пользования конспектом.

При проверке задания, оцениваются умение последовательно, четко и логически стройно излагать избранную проблему, обобщать, сопоставлять различные точки зрения по рассматриваемой проблеме, аргументировать основные положения и выводы, использовать научную литературу.

Оценочный лист студента (ки) _____ Ф.И.О.		
№ гр. _____		
Оценка складывается как среднее арифметическое из трех оценок: правильность ответа; умение приводить примеры; умение отвечать на дополнительные вопросы		
Оценка правильности ответа	Оценка умения приводить примеры	Оценка умения отвечать на дополнительные вопросы
Итоговая оценка		

Вопросы для подготовки к собеседованию по материалам лекций

Тема 1.

1. Порядок перевода чисел из одной позиционной системы счисления в другую.
2. Алгоритм перевода двоичного числа в другую систему счисления, основанием которой является степень двойки.

Тема 2.

3. Характеристика принципов построения и работы арифметических сумматоров.
4. Характеристика принципов построения и работы дешифраторов и шифраторов.

Тема 3.

5. Предназначение и классификация оперативных запоминающих устройств.
6. Характеристика внутренней структуры и принципов функционирования типового оперативного запоминающего устройства.

Тема 4.

7. Порядок исследования программы нахождения суммы чисел, имеющих одинаковую длину.
8. Порядок исследования программы нахождения разности чисел, имеющих одинаковую длину.

Тема 5.

9. Порядок тестирования жесткого диска с помощью утилиты hd tune с целью определения скорости передачи данных и времени доступа к данным.
10. Порядок сканирования жесткого диска и флеш-памяти на предмет наличия ошибок с помощью утилиты hd tune.

Тема 6.

11. Составить алгоритм работы струйного принтера.
12. Составить алгоритм работы лазерного принтера.
13. Порядок определения конфигурацию пэвм при его запуске в процессе тестирования компьютера базовой системой ввода-вывода bios.
14. Перечень тестовых программ под ос windows для диагностирования аппаратного обеспечения компьютера.

Повышенный уровень

Тема 1.

1. порядок процедуры умножения (деления) операндов нормализованных двоичных чисел.

Тема 2.

2. Порядок исследования логики функционирования шифратора 8х3, реализованного в виде имс 74148 (км555ив1) с помощью генератора слова.

Тема 3.

3. Порядок исследования внутренней структуры и логики функционирования одноразрядного шинного формиратора.

Тема 4.

4. Порядок исследования программы вычисления квадрата числа с помощью таблицы.

Тема 5.

5. Порядок чтения паспортов внешних запоминающих устройств пэвм с помощью программы victoria.

Тема 6.

6. Подключить лазерный принтер к пэвм в сетевом режиме, проверить его работоспособность.
7. Предназначение и краткая характеристика программы «aida64 extreme edition»

1. Критерии оценивания компетенций

Оценка «отлично» выставляется студенту, если он на высоком уровне: понимает арифметические основы вычислительной техники; владеет навыками обработки экспериментальной и теоретической физической информации о состоянии ЭВМ; применяет методы и средства проектирования цифровых узлов ЭВМ.

Оценка «хорошо» выставляется студенту, если на среднем уровне: понимает арифметические основы вычислительной техники; владеет навыками обработки экспериментальной и теоретической физической информации о состоянии ЭВМ; применяет методы и средства проектирования цифровых узлов ЭВМ.

Оценка «удовлетворительно» выставляется студенту, если на минимальном уровне: понимает арифметические основы вычислительной техники; владеет навыками обработки экспериментальной и теоретической физической информации о состоянии ЭВМ; применяет методы и средства проектирования цифровых узлов ЭВМ.

Оценка «неудовлетворительно» выставляется студенту, если он с грубыми ошибками: понимает арифметические основы вычислительной техники; владеет навыками обработки экспериментальной и теоретической физической информации о состоянии ЭВМ. анализируемой информации; применяет методы и средства проектирования цифровых узлов ЭВМ.

2. Описание шкалы оценивания

Максимально возможный балл за весь текущий контроль устанавливается равным 55. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

Уровень выполнения контрольного задания	Рейтинговый балл (в % от максимального балла за контрольное задание)
Отличный	100
Хороший	80
Удовлетворительный	60
Неудовлетворительный	0

3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедура проведения данного оценочного мероприятия включает в себя:

- отдельное собеседование по каждой из указанных тем;

Предлагаемые студенту вопросы позволяют проверить ОПК-4 (ИД-1 ОПК-4, ИД-2 ОПК-4, ИД-3 ОПК-4) компетенции.

Для подготовки к данному оценочному мероприятию необходимо изучить рекомендованные источники и научную литературу и на их основании подготовиться к указанным в методических материалах вопросам для обсуждения. Для подготовки к данному оценочному мероприятию необходимо 2 часа, в течение которых обучающийся изучает научную и учебную литературу, составляет тезисы.

При подготовке к ответу студенту предоставляется право пользования конспектом.

При проверке задания, оцениваются умение последовательно, четко и логически стройно излагать избранную проблему, обобщать, сопоставлять различные точки зрения по

рассматриваемой проблеме, аргументировать основные положения и выводы, использовать научную литературу.

Оценочный лист студента (ки) _____ Ф.И.О.			
№ гр. _____			
Оценка складывается как среднее арифметическое из трех оценок: правильность ответа; умение приводить примеры; умение отвечать на дополнительные вопросы			
Оценка правильности ответа	Оценка умения приводить примеры	Оценка умения отвечать на дополнительные вопросы	Итоговая оценка

Темы докладов Базовый уровень

1. История развития вычислительной техники.
2. Блоки питания АТХ.
3. Материнские платы.
4. Процессоры.
5. Видеоадаптеры.
6. Оперативная память
7. Жесткие диски.
8. FDD, CD, DVD приводы.
9. Флэш - память типы и принципы работы.
10. Порты ввода – вывода (COM, LPT, USB)
11. Сетевые карты.
12. Сетевое оборудование (коммутаторы)
13. Сетевое оборудование (маршрутизаторы)
14. Оборудование беспроводного доступа (Wi Fi)

Повышенный уровень

1. Сравнение различных алгоритмов сжатия видеоизображения в системах охранного телевидения.
2. Обоснование программно-аппаратной конфигурации для различных систем охранного телевидения, базирующихся на компьютере.
3. Расчет необходимых объемов хранилищ данных в системах охранного телевидения.
4. Обоснование характеристик вычислительной сети для использования в системе охранного телевидения на IP-видеокамерах.
5. Обоснование структуры и состава программно-аппаратного комплекта системы охранного телевидения с управлением через Интернет.
6. Обоснование программно-аппаратного комплекта для распределенной системы охранного телевидения, базирующейся на компьютерах.
7. Активные элементы радиоэлектронной аппаратуры.

1. Критерии оценивания компетенций

Оценка «отлично» выставляется студенту, если он на высоком уровне: понимает арифметические основы вычислительной техники; владеет навыками обработки экспериментальной и теоретической физической информации о состоянии ЭВМ; применяет методы и средства проектирования цифровых узлов ЭВМ.

Оценка «хорошо» выставляется студенту, если на среднем уровне: понимает арифметические основы вычислительной техники; владеет навыками обработки экспериментальной и теоретической физической информации о состоянии ЭВМ; применяет методы и средства проектирования цифровых узлов ЭВМ.

Оценка «удовлетворительно» выставляется студенту, если на минимальном уровне: понимает арифметические основы вычислительной техники; владеет навыками обработки экспериментальной и теоретической физической информации о состоянии ЭВМ; применяет методы и средства проектирования цифровых узлов ЭВМ.

Оценка «неудовлетворительно» выставляется студенту, если он с грубыми ошибками: понимает арифметические основы вычислительной техники; владеет навыками обработки экспериментальной и теоретической физической информации о состоянии ЭВМ. анализируемой информации; применяет методы и средства проектирования цифровых узлов ЭВМ.

2. Описание шкалы оценивания

Максимально возможный балл за весь текущий контроль устанавливается равным 55. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

Уровень выполнения контрольного задания	Рейтинговый балл (в % от максимального балла за контрольное задание)
Отличный	100
Хороший	80
Удовлетворительный	60
Неудовлетворительный	0

3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедура проведения данного оценочного мероприятия включает в себя:

- устный доклад по одной из указанных тем;

Предлагаемые студенту вопросы позволяют проверить ОПК-4 (ИД-1 ОПК-4, ИД-2 ОПК-4, ИД-3 ОПК-4) компетенции.

Для подготовки к данному оценочному мероприятию необходимо изучить рекомендованные источники и научную литературу и на их основании подготовиться к указанным в методических материалах вопросам для обсуждения. Для подготовки к данному оценочному мероприятию необходимо 15 часов, в течение которых обучающийся изучает научную и учебную литературу, составляет текст доклада и готовит презентацию.

При подготовке к ответу студенту предоставляется право пользования текстом доклада.

При проверке задания, оцениваются умение последовательно, четко и логически стройно излагать избранную проблему, обобщать, сопоставлять различные точки зрения по рассматриваемой проблеме, аргументировать основные положения и выводы, использовать научную литературу.

Оценочный лист студента (ки) _____ Ф.И.О.		
№ гр. _____		
Оценка складывается как среднее арифметическое из трех оценок: правильность ответа; умение приводить примеры; умение отвечать на дополнительные вопросы		
Оценка правильности ответа	Оценка умения приводить примеры	Оценка умения отвечать на дополнительные вопросы
		Итоговая оценка

Список рекомендуемой литературы

Перечень основной литературы:

1. Айдинян, А. Р. Аппаратные средства вычислительной техники : учебник / А.Р. Айдинян. - Москва|Берлин : Директ-Медиа, 2016. - 125 с. : ил., схем., табл. -<http://biblioclub.ru/>. - Библиогр. в кн. - ISBN 978-5-4475-8443-6
2. Царев, Р. Ю. Программные и аппаратные средства информатики Электронный ресурс : Учебник / Р. Ю. Царев, А. В. Прокопенко, А. Н. Князьков. - Красноярск : Сибирский федеральный университет, 2015. - 160 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-7638-3187-0 8.1.2.

Перечень дополнительной литературы:

1. Информатика. Базовый курс: учебник для вузов / под. ред. С. В. Симоновича. - СПб. : Питер, 2014. - 564 с.
2. Олифер В. Г. Сетевые операционные системы / В. Г. Олифер, Н. А. Олифер - СПб.: Питер, 2014. - 544 с.
3. Душкин А. В. Информационные технологии и системы : учебник / С. В. Белокуров, А. В. Душкин, В. И. Сумин и др. - Воронеж : Научная книга, 2012. - 568 с.
4. Аппаратные интерфейсы ПК: Энциклопедия / М. Гук - СПб. : Питер, 2002. - 528 с.
5. Калачев А. В. Многоядерные процессоры / А. В. Калачев. - Интернет- университет информационных технологий, Бином. Лаборатория знаний, 2014. - 248 с.

Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине (модулю)

1. Вычислительная техника и информационные технологии. Практикум Электронный ресурс / сост. З. С. Онуприенко. - Вычислительная техника и информационные технологии. Практикум, 2022-04-04. - Москва : Московский технический университет связи и информатики, 2016. - 32 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. – ISBN 2227-8397 2
2. Орлов, С. А. Организация ЭВМ и систем : [фундаментальный курс по архитектуре и структуре современных компьютерных средств] : учебник для студентов вузов / С. А.
3. Орлов, Б. Я. Цилькер. - 2-е изд. - Санкт-Петербург [и др.] : ПИТЕР, 2011. - 687 с. : прил. - (Учебник для вузов). - Библиогр.: с. 665-672. - Алфавит. указ.: с. 673-686. - ISBN 978-5-49807-862-5
4. Функциональные блоки аппаратных средств вычислительной техники [Электронный ресурс] : Практикумы NoNo 5, 6, 7, 8 по дисциплине Аппаратные средства вычислительной техники / сост. Г. В. Жуков. - Функциональные блоки аппаратных средств вычислительной техники, 2022-04-04. - Москва: Московский технический университет связи и информатики, 2015. - 26 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 2227-839

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)

1. <http://elibrary.ru/defaultx.asp> - Научная электронная библиотека
2. <http://www.intuit.ru> – Интернет-Университет Информационных Технологий.
3. <http://www.iprbookshop.ru>
4. www.paralab.ru