

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 «Физические основы информационных технологий»
для студентов направления подготовки
09.03.02 «Информационные системы и технологии»
Профиль «Прикладное программирование в интеллектуальных
информационных системах»

Ставрополь, 2026

Данные методические указания предназначены для студентов факультета информационных систем и технологий обучающихся по направлению 09.03.02

В данные указания включены лабораторные работы, представляющие основы сжатия данных, основы построения систем ОЗУ ЭВМ, сетей и систем телеобработки, многопроцессорных и многоядерных ЭВМ.

ЛАБОРАТОРНАЯ РАБОТА №1. ИССЛЕДОВАНИЕ МЕТОДОВ НЕРАЗРУШАЮЩЕГО СЖАТИЯ ИНФОРМАЦИИ. ПРОСТЕЙШИЕ МЕТОДЫ НЕРАЗРУШАЮЩЕГО СЖАТИЯ

Цель работы

Цель работы – уточнить постановку задачи и классификацию методов неразрушающего сжатия информации, изучить простейшие методы неразрушающего сжатия.

Теоретическое обоснование

1. Введение и классификация

Сжатие информации - перевод информации в такую форму записи, при которой размер файла сокращается.

Сжатие бывает:

- а) неразрушающее - состоит в удалении из информации избыточной информации, полезная информация остается без изменений;
- б) разрушающее - кроме избыточной информации удаляется часть полезной, хотя может быть и не самая важная часть.

Применение: неразрушающее сжатие применяется к текстовым и программным файлам, разрушающее - к аудио и видео.

В дальнейшем, в ходе данной работы, будем рассматривать методы сжатия - применительно исключительно к неразрушающему сжатию. Для упрощения изложения - рассмотрим сжатие на примере текстовых файлов. Но, будем иметь в виду, рассматриваемые методы применимы к любым

файлам (сообщениям), содержимое которых можно представить как последовательность двоичных кодов.

2. Понятие кодовой таблицы.

При кодировании каждому символу исходного сообщения, ставится в однозначное соответствие некоторая битовая комбинация кода. Список, в котором указано соответствие кодов исходного сообщения кодам сжатого, называется "кодовой таблицей" или "таблицей соответствия", например:

Символ	Код
А	00
В	01
С	10

Основным приёмом кодирования информации для ее сжатия, является построение специальных кодовых таблиц (таблиц соответствия). Удачно построенная таблица обеспечивает наименьший размер файла (сообщения) и сжатие файла (сообщения).

3. Минимальное равномерное кодирование.

3.1 Принцип работы.

Для кодирования одного символа требуется как минимум $n = \lceil \log_2 M \rceil$ бит информации, где M - объем алфавита.

Такой код, с длиной $n = \lceil \log_2 M \rceil$ принято называть "минимальным", величину n - "минимальной кодовой длиной", а сообщения, составленные из кодов такой длины - "минимальными равномерными кодами".

В реальных файлах "стандартных форматов", используется кодирование символов с использованием "стандартных кодов". Кодовые таблицы для "стандартных кодов" составляются исходя из того, что код - должен обеспечить возможность представления всех возможных символов, которые хоть когда, но могут встретиться в файле данного формата. В реальном файле, с большой вероятностью, не все допускаемые форматом файла символы - будут использованы. В таком случае - мы будем иметь файл, коды в котором имеют длину большую чем "минимальная кодовая длина".

Пример: стандартные текстовые коды KOI8 всегда 8-битовые, независимо от того, каков реальный объем алфавита сообщения.

Такие (составленные из кодов длиной большей чем "минимальная кодовая длина") файлы (сообщения) – с целью сжатия, могут быть перекодированы с использованием кодовых таблиц использующих коды с длиной, соответствующей "минимальной кодовой длине".

3.2 Алгоритм кодирования.

1. Подсчитываем M - объем алфавита.
2. Вычисляем $n_{\min} = \lceil \log_2 M \rceil$.
3. Составляем кодовую таблицу, нестандартную, с кодами длиной $n_{\min}$ вместо кодов стандартной длины.

4. Получаем сжатый файл: используя составленную кодовую таблицу, перекодируем исходный файл из стандартных кодов - в "минимальные коды".
5. Для того, чтобы в дальнейшем было возможно прочитать содержимое сжатого файла, возможно, потребуется дописать (приложить) к файлу, полученную в п. 2 уникальную кодовую таблицу.

3.3 Алгоритм декодирования.

Декодер, используя приложенную к файлу уникальную кодовую таблицу, определяет символы, и далее заменяет "минимальные равномерные коды" на "стандартные коды" из "стандартной кодовой таблицы".

3.4 Адаптивные и вариантные кодовые таблицы

Не во всех случаях необходимо дописывать (прилагать) к сжатому файлу использованную при сжатии кодовую таблицу.

Возможен один из двух методов передачи:

1. С передачей уникальной кодовой таблицы составленной специально для данного сообщения. Такие кодовые таблицы принято называть "адаптивными".
2. С использованием заранее подготовленных таблиц (принято называть "вариантными"). Тогда, заранее, ещё при разработке программ сжатия файлов и данных, составляется множество вариантов кодовых таблиц для различных "типичных случаев" содержимого сжимаемого файла. В ходе кодирования определяется, какая из таблиц больше подходит под данный

файл, и именно эта таблица и используется при кодировании. В данном случае, вместо передачи "уникальной кодовой таблицы", потребуется передать только номер варианта таблицы.

Первый метод используется при кодировании файлов (сообщений) большого размера, второй – малого, когда выигрыш, полученный от использования уникальной кодовой таблицы, не может превысить проигрыш от необходимости дополнительной передачи таблицы.

4. Коэффициент сжатия

В различных источниках, в различных случаях для подсчёта коэффициента сжатия может использоваться как формула:

$$K = \frac{N_{\text{исходного}}}{N_{\text{сжатого}}}$$

так и:

$$K = \frac{N_{\text{сжатого}}}{N_{\text{исходного}}}$$

При минимальном равномерном кодировании без учета кодовой таблицы коэффициент сжатия составит $K = \frac{\lceil \log_2 M \rceil}{n}$, где M – число символов сообщения, n – длина стандартного кода.

5. Неразрушающее групповое кодирование

Дополнительное сжатие можно обеспечить, если в файле (сообщении) - встречаются часто повторяющиеся последовательности (строки) символов. Например – часто повторяются одни и те же слова.

Тогда, при составлении уникальной кодовой таблицы, можно предусмотреть передачу одним кодом не только отдельных символов, но и часто повторяющихся последовательностей символов. Кодеру будет просмотреть сжимаемый файл (сообщение), найти повторяющиеся последовательности, и включить в кодовую таблицу коды для каждой из найденных последовательностей.

Аппаратура и материалы

Для выполнения лабораторной работы необходим персональный компьютер со следующими характеристиками: процессор Intel с тактовой частотой 1800 МГц и выше, оперативная память – не менее 128 Мбайт, свободное дисковое пространство – не менее 500 Мбайт, устройство для чтения компакт – дисков, монитор типа Super VGA (число цветов – 256) с диагональю не менее 15 ". **Программное обеспечение** – операционная система WINDOWS 98 / NT / ME / 2000 / XP, пакет MathCAD, начиная с версии 2000 и выше , Visual C++ 6.0, Borland C 3.0 и выше.

Указания по технике безопасности

Техника безопасности при выполнении лабораторной работы совпадает с общепринятой для пользователей персональных компьютеров. В частности: самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения; в случае неисправности персонального компьютера сообщить об этом обслуживающему персоналу лаборатории (оператору, администратору); соблюдать правила техники безопасности при работе с электрооборудованием; не касаться электрических розеток металлическими предметами; рабочее место пользователя

персонального компьютера должно содержаться в чистоте; не разрешается возле персонального компьютера принимать пищу, напитки.

Содержание отчета и его форма

Отчет по лабораторной работе должен включать:

1. название лабораторной работы;
2. ответы на контрольные задания;
3. формулировки индивидуального задания и порядка его выполнения.

Отчет о выполнении лабораторной работы в письменном виде сдается преподавателю.

Задания

- 1) Определить минимальную кодовую длину для сообщения "мама мыла раму"
- 2) Закодировать в минимальный равномерный код сообщение "мама мыла раму"
- 3) Определить коэффициент сжатия при минимальном равномерном кодировании, для сообщения "мама мыла раму"
- 4) Декодировать из минимального равномерного кода сообщение "000001000001010000011100001010101001000110" если известна кодовая таблица:

СИМ ВОЛ	КОД
м	000
а	001

—	010
ы	011
л	100
р	101
у	110

ЛАБОРАТОРНАЯ РАБОТА №2. ИССЛЕДОВАНИЕ МЕТОДОВ НЕРАЗРУШАЮЩЕГО СЖАТИЯ ИНФОРМАЦИИ. МИНИМАЛЬНОЕ НЕРАВНОМЕРНОЕ КОДИРОВАНИЕ

Цель работы

Цель работы – изучить алгоритм "минимального неравномерного кодирования", одного из самых применяемых методов неразрушающего сжатия, освоить генерацию кодовых таблиц минимального неравномерного кодирования с использованием алгоритма Шеннона-Фано, а также изучить вопрос определения коэффициента сжатия для методов минимального неравномерного кодирования.

Теоретическое обоснование

1. Принцип работы.

Различие с методом минимального равномерного кодирования, проявляется при составлении кодовой таблицы. Для часто встречающихся символов используются очень короткие коды, длиной меньше $n_{\min} = \lceil \log_2 M \rceil$. Тогда для редких символов, придется использовать длинные коды, длиной даже больше чем $n_{\min}$.

Кроме этого, в силу того что задача дешифрации при использовании этого метода оказывается не так тривиальна, различия имеются также в алгоритме дешифрации.

2. Дешифрация минимальных неравномерных кодов.

Проблема: как определить границы между отдельными кодами?

Решение проблемы: возможно т. н. методом «последовательного разбора», но только если наложить на используемые коды дополнительное ограничение - никакие первые биты длинных кодов, не должны совпадать с короткими кодами.

Если, при составлении таблицы, означенное условие выполнено, возможна дешифрация с использованием "метода последовательного разбора".

3. Метод последовательного разбора.

1. Считываем из файла первый бит.
2. Интерпретируя этот бит - как возможный код, просматриваем кодовую таблицу и ищем - нет ли в таблице такого кода.
3. Если нет, считывается второй бит и т. д. пока не получим совпадение считанного кода с одним из кодов кодовой таблицы. Считаем - символ найден.

4. С того места, где закончен поиск предыдущего символа, аналогично начинаем поиск нового символа. Повторяем алгоритм поиска нового символа до тех пор, пока не будут определены все символы.

4. Алгоритмы генерации кодовой таблицы.

Задача: составить такую кодовую таблицу, чтобы часто встречающиеся символы в ней были закодированы короткими кодами и при этом короткие коды не совпадали с первыми битами длинных кодов.

Два наиболее популярных метода:

1. Алгоритм Шеннона-Фано - более простой и быстрый.
2. Алгоритм Лэмпэла-Зива - усовершенствованный алгоритм Шэннона-Фано, обеспечивает большее сжатие, но сложнее и требует больших вычислений при составлении кодовой таблицы.

5. Алгоритм Шэннона-Фано.

1. Будем считать список из разновидности символов встречаемых в тексте "группой символов".
2. Подсчитаем частоту появления F_i для каждого из символов, входящего в группу символов.
3. Ранжируем символы по частоте: от самых частых в начале списка к самым редким в конце списка.
4. Подсчитаем сумму частот F символов в группе.
5. Подсчитаем половину суммы частот $F/2$ символов в группе.
6. Разобьем группу на 2 подгруппы:
 - подгруппу нулей, куда будут включены более частые символы, их коды начинаются с 0,
 - подгруппу единиц, куда будем включать коды более редких символов.Разбивать будем так, чтобы сумма частот F_i символов в обеих подгруппах - оказалась по возможности одинаковой. Для этого:
 - а) начинаем включать символы в подгруппу единиц, начиная с самого редкого;
 - б) после включения каждого символа подсчитывается текущая сумма частот F_i в подгруппе единиц;
 - в) включаем символы в подгруппу нулей до тех пор, пока текущая сумма частот F_i в подгруппе единиц, не окажется равной или не превысит $F/2$;
 - г) оставшиеся символы включим в подгруппу нулей;

- д) проверим, остались ли символы в подгруппе единиц, если нет, то один из символов подгруппы нулей, последний из внесённых подгруппу единиц, перенесем в подгруппу нулей.
7. Результат выполнения пунктов 1-6 - получили первый бита кода для каждого из символов.
8. Определяем второй бит. Для этого:
- а) каждую из получившихся в пункте 6 подгрупп назовем группой;
 - б) если в группе оказывается всего один символ, других бит в коде этого символа уже не будет – код символа определён;
 - б) каждую группу, в которой более 1 символа, разобьем на подгруппы точно так же, как мы это делали в пунктах 1-7.
9. Процесс разбиения групп на подгруппы, и определения новых бит - будет продолжаться до тех пор, пока в каждой группе - не останется по одному символу.

Задача.

Составить кодовую таблицу для сжатия фразы «мама мыла раму».

Решение.

Оформим решение в виде таблицы:

С						
ИМВОЛ	i	ИТ	ИТ	ИТ	ИТ	ИТ
		1	2	3	4	5
М						
а						
—						
Ы						
л						
р						
у						

Ответ: получим следующие коды символов: М = 0; А = 10; _ = 1100;
Ы = 1101; Л = 1110; Р = 00001; У = 11111.

2. Коэффициент сжатия.

Как при использовании алгоритма Шеннона-Фано, так и при использовании любого другого алгоритма составления кодовых таблиц, для определения коэффициента сжатия необходимо знать (подсчитать) для каждого из символов кодовой таблицы:

- Частоту появления данного символа в сообщении F_i ;
- Длину кода для данного символа в кодовой таблице минимального неравномерного кодирования n_i ;

Тогда длина сжатого файла:

$$N_{сжатого} = \sum_{i=1}^M (n_i \times F_i)$$

Коэффициент сжатия:

Аналогично минимальному равномерному кодированию, в различных источниках используется как формула $K = \frac{N_{исходного}}{N_{сжатого}}$, так и $K = \frac{N_{сжатого}}{N_{исходного}}$

Аппаратура и материалы

Для выполнения лабораторной работы необходим персональный компьютер со следующими характеристиками: процессор Intel с тактовой частотой 1800 МГц и выше, оперативная память – не менее 128 Мбайт, свободное дисковое пространство – не менее 500 Мбайт, устройство для чтения компакт – дисков, монитор типа Super VGA (число цветов – 256) с диагональю не менее 15 ". **Программное обеспечение** – операционная система WINDOWS 98 / NT / ME / 2000 / XP, пакет MathCAD, начиная с версии 2000 и выше , Visual C++ 6.0, Borland C 3.0 и выше.

Указания по технике безопасности

Техника безопасности при выполнении лабораторной работы совпадает с общепринятой для пользователей персональных компьютеров. В частности: самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения; в случае неисправности персонального компьютера сообщить об этом обслуживающему персоналу лаборатории (оператору, администратору); соблюдать правила техники безопасности при работе с электрооборудованием; не касаться электрических

розеток металлическими предметами; рабочее место пользователя персонального компьютера должно содержаться в чистоте; не разрешается возле персонального компьютера принимать пищу, напитки.

Содержание отчета и его форма

Отчет по лабораторной работе должен включать:

1. название лабораторной работы;
2. ответы на контрольные задания;
3. формулировки индивидуального задания и порядка его выполнения.

Отчет о выполнении лабораторной работы в письменном виде сдается преподавателю.

Задания

1) Декодировать из минимального неравномерного кода сообщение "01001011000110111101011001111010011111", если известна кодовая таблица:

Симво л	Код
м	0
а	10
–	1100
ы	1101
л	1110
р	11110
у	11111

2) Объяснить, за счёт чего при минимальном неравномерном кодировании, может достигаться сжатие более сильное чем при минимальном равномерном кодировании. Как, по вашему: каковы условия, при которых сжатие с использованием минимального неравномерного кодирования, будет оказываться более сильным?

3) Закодировать в минимальный неравномерный код сообщение "мама мыла раму"

3) Определить коэффициент сжатия при минимальном неравномерном кодировании, для сообщения "мама мыла раму"

ЛАБОРАТОРНАЯ РАБОТА №3. ИССЛЕДОВАНИЕ МЕТОДОВ РАЗРУШАЮЩЕГО СЖАТИЯ ИНФОРМАЦИИ. ОСНОВНЫЕ ПРИЁМЫ И АЛГОРИТМЫ РАЗРУШАЮЩЕГО СЖАТИЯ

Цель работы

Цель работы – пока без изучения конкретики тех или иных многочисленных алгоритмов разрушающего сжатия, освоить основные приёмы и принципы, на которых, построена работа средств (программ или аппаратуры) разрушающего сжатия.

Теоретическое обоснование

1. Введение

Методы разрушающего сжатия предполагают удаление из данных не только избыточной, но и части полезной информации. Применяются, как правило, к сжатию графики, аудио и видео.

2. Приемы разрушающего сжатия.

2.1. Дифференциальное кодирование.

Применяют при сжатии аудио и видео.

Основано на том, что соседние кадры отличаются не так сильно, и описание различия между соседними кадрами по объему меньше, чем полное описание нового кадра. Вместо описания нового кадра передается т. н. «дифферент» - различие между текущим и предыдущим кадром.

Проблема: резкая смена кадров, при которой, изменение размывается на несколько соседних кадров, в результате нарушается динамика и появляются т. н. "хвосты" - следы предыдущих кадров.

D ("объем дифференциала") - объем информации, выделяемой для передачи различия между кадрами. Чем больше D , тем меньше сжатие и искажение. Изменением D регулируется степень сжатия.

2.2. Разрушающее групповое кодирование.

Анализируется кадр. Если в кадре присутствуют одинаковые фрагменты, то вместо индивидуальной передачи каждого фрагмента - передается образец и ссылки на него.

Проблема: совершенно одинаковых фрагментов практически не бывает, поэтому, как одинаковые передаются не только полностью идентичные, но и похожие фрагменты.

В результате исчезают мелкие детали, изображение принимает характер рисунка.

GD ("групповой дифференциал") - максимальное различие между фрагментами, при котором кодеру ещё разрешено применять групповую передачу. Чем больше GD , тем больше сжатие и меньше качество. При $GD = 0$ нет потерь информации, но сжатие, как правило, очень незначительно.

Сжатие палитры.

1. Анализируется кадр или фрагмент сюжета (для случая видеоданных).
Отбирается определенное количество цветов - наиболее важных для восприятия цвета в кадре. Данные цвета относят к так называемым "базовым", передаются без искажений.
2. Остальные цвета - заменяются на ближайшие из отобранных.

Результат: меньше количество используемых цветов, более компактный код цвета, искажение цвета – особенно, если число отобранных (т. н. базовых) цветов - мало.

2.4. Психофизиологические методы.

Это методы, основанные на учете особенностей зрения и слуха человека.

Например:

1. "Эффект временной усталости" - после громкого звука человек некоторое время не слышит тихий.
2. "Эффект маскирования" - мощный низкочастотный звук делает неслышимыми звуки, близкие по частоте, но более высокие.
3. Учет спектральной чувствительности слуха. Звуки, находящиеся на краях спектра, человек слышит хуже, чем в середине диапазона, в частности плохо отличает низкие и высокие звуки по частоте. Применяют прием сужения диапазона, когда очень низкие и очень высокие звуки заменяются на звуки, лежащие ближе к середине спектра. Кроме того, края диапазона в целом подвергаются более сильному сжатию с использованием различных методов, чем середина диапазона.
4. Учет спектральной чувствительности зрения. Искажение цветов на краях цветового диапазона (красный, синий) воспринимается не так заметно, как искажение цветов середины диапазона (оранжевый, желтый, зеленый).

3. Популярные форматы разрушающего сжатия

3.1 Сжатие статичной графики.

Распространение приобрел алгоритм сжатия фотографических изображений JPEG.

Коэффициент сжатия - от 1 до 100, искажение зависит от степени сжатия.

Другие стандарты (например - PNG, TIFF) – более ориентированы на сжатие схем, чертежей, рисунков и используют в зависимости от содержащейся в файле информации как методы разрушающего, так и неразрушающего сжатия.

3.2 Сжатие динамической графики

DCM - алгоритм, применяемый при передаче телевизионного сигнала, а также для хранения архивных видеоматериалов. В основном используется дифференциальное кодирование, передачу с использованием DCM иногда называют "телемодемной передачей".

3.3 Сжатие аудио

- 1) ATRAC (1.0 - 5.XX) - разработан Sony для системы звукозаписи «мини-диск». Коэффициент сжатия - около 3. Позволяет звуковую информацию, размещающуюся на CD (650 Mb) разместить на компактном мини-диске (240 Mb). Уровень искажений такой, что при передачи популярной музыки искажение как правило не заметны.
- 2) MP3 - коэффициент сжатия больше, чем у ATRAC и регулируемый.

Выделим три стандартных алгоритма с уровнем сжатия:

- 1 - 64 Кб/сек - пригоден для передачи речи;
- 2 - 128 Кб/сек - передача музыки без претензий на качество;

3 - 256 Кб/сек - качество звука приближается к ATRAC.

История MP3: MP3 - это вырезанный из стандарта MPEG3 фрагмент, занимающийся кодированием аудио. MPEG3 – разрабатывался как стандарт видеотелефонной связи. MP3 был выделен из MPEG3 хакерами, которых привлекла возможность сильного сжатия и отсутствие каких-либо средств защиты авторских прав. До сих пор распространение MP3 - предмет конфликтов между держателями авторских прав на аудиопroduкцию и некоторыми распространителями продукции в формате MP3.

3.4 Совместное сжатие аудио и видео

- 1) AVI - стандартный метод сжатия для ОС Windows, программы сжатия/декомпрессии входят в Windows. В последнее время из-за создания и распространения более совершенных форматов - применяется реже. Можно встретить в игровых программах.

Характеристики:

1. Степень сжатия - 1 час / 1 Гбайт.
2. разрешение - 200 * 320.
3. Качество картинки – посредственное. Заметно искажение цветов, динамики.
4. Требования к аппаратуре - низкие (применим на любой конфигурации, совместимой с Windows).

- 2) MPEG1 (MP1) - более поздняя разработка, чем AVI, выше качество изображения, лучше звук. Требования к компьютеру выше, чем у AVI, но по современным понятиям - очень скромные.

Характеристики:

1. Степень сжатия - 1 час / 1 Гбайт.
2. Разрешение - 240*320.

3. Качество картинки - удовлетворительное, иногда заметно нарушение динамики, умеренное искажение цвета.

3) MPEG4 (MP4) - последняя разработка для замены MPEG1.

Характеристики:

1. Степень сжатия - 2 часа / 1 Гбайт.
2. Разрешение - до 420 * 560.
3. Качество аудио соответствует качеству MP3, видео - качество хорошее, но при условии проигрывания на высокопроизводительном компьютере, иначе заметно нарушение динамики и "мультипликативность" изображения.

Требования к аппаратуре - по современным понятиям довольно высокие.

4) MP2 (Mpeg2) - формат с низким коэффициентом сжатия. Способен обеспечить качество сравнимое с качеством обычных телевизионный прием.

Характеристики:

1. Степень сжатия - 15 минут / 1 Гб.
2. Разрешение - до 578*720.
3. Аудио - шестиканальное аудио по системе 5.1 (5 колонок и 1 сабвуфер).

Применение: Конкретизация MPEG2 применяется как стандартный формат для DVD проигрывателей.

3.5 Универсальные методы сжатия

Сюда относятся методы, которые инвариантны к сжимаемым данным, т. е. сожмут любую информацию, поступившую на вход кодера.

Как правило, универсальность достигается за счет снижения качества сжатия и увеличения потерь полезной информации.

АО РСМ (АДИКМ) - импульсно-кодовая модуляция. Применяется на телефонных каналах для подключения удаленных пользователей. Один из главных «врагов» модема, поскольку скорость работы модема на каналах с использованием АДИКМ, как правило, не превышает 9600 бит/с.

Аппаратура и материалы

Для выполнения лабораторной работы необходим персональный компьютер со следующими характеристиками: процессор Intel с тактовой частотой 1800 МГц и выше, оперативная память – не менее 128 Мбайт, свободное дисковое пространство – не менее 500 Мбайт, устройство для чтения компакт – дисков, монитор типа Super VGA (число цветов – 256) с диагональю не менее 15 ". **Программное обеспечение** – операционная система WINDOWS 98 / NT / ME / 2000 / XP, пакет MathCAD, начиная с версии 2000 и выше , Visual C++ 6.0, Borland C 3.0 и выше.

Указания по технике безопасности

Техника безопасности при выполнении лабораторной работы совпадает с общепринятой для пользователей персональных компьютеров. В частности: самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения; в случае неисправности персонального компьютера сообщить об этом обслуживающему персоналу лаборатории (оператору, администратору); соблюдать правила техники безопасности при работе с электрооборудованием; не касаться электрических розеток металлическими предметами; рабочее место пользователя

персонального компьютера должно содержаться в чистоте; не разрешается возле персонального компьютера принимать пищу, напитки.

Содержание отчета и его форма

Отчет по лабораторной работе должен включать:

4. название лабораторной работы;
5. ответы на контрольные задания;
6. формулировки индивидуального задания и порядка его выполнения.

Отчет о выполнении лабораторной работы в письменном виде сдается преподавателю.

Задания

1) Проанализировать особенности основных разновидностей информации, из тех разновидностей информации, данные по которым - могут храниться и передаваться с применением разрушающего сжатия. Предложить методы, которые могли бы быть использованы для каждого из этих случаев.

ЛАБОРАТОРНАЯ РАБОТА №4. ИССЛЕДОВАНИЕ ОСНОВ ПОСТРОЕНИЯ ОЗУ. КЛАССИФИКАЦИИ И ЦИКЛ ДОСТУПА К ПАМЯТИ

Цель работы

Цель работы – освоить необходимые сведения о устройстве, характеристиках оперативных запоминающих устройств.

Теоретическое обоснование

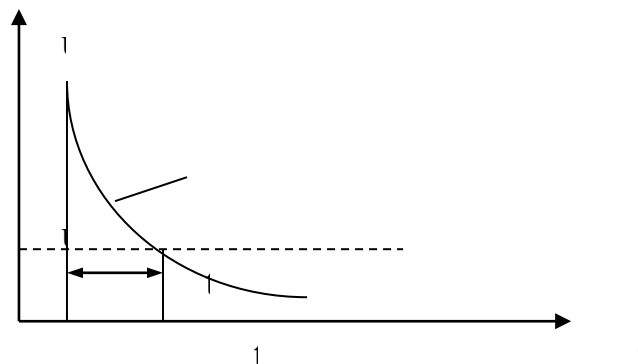
1. Классификация по типу запоминающих ячеек

Различают 2 типа микросхем:

- а) статические - 1 бит информации хранится на электронном устройстве, называемом триггером, и состоящим из 4 или 6 транзисторов. Информация о или 1, занесенная в триггер, хранится в нем до выключения питания;
- б) динамические - 1 бит информации хранится в виде заряда на 1 транзисторе (есть заряд - 1, нет заряда - 0). Записанная информация постепенно стирается, поскольку заряд с транзистора постепенно рассасывается.

2. Особенности ОЗУ на динамических ячейках памяти.

Рассмотрим процесс записи и хранения бита данных для динамического ОЗУ:



На рисунке:

t_3 - момент записи 1 в ячейку памяти ОЗУ;

t_c - момент стирания из-за падения напряжения; $u < u_1$ и 1 превратилась в 0;

u - напряжение на транзисторе;

u_1 - пороговое напряжение единицы. Если $u > u_1$, то ячейка памяти хранит 1, если $u < u_1$ - 0.

1 - с течением времени заряд уменьшается и напряжение падает;

t_x - время хранения (в течение которого 1 остается), для разных микросхем t_p от 100 мкс до 1 с.

Заметим: время хранения данных в ячейках динамического ОЗУ ограничено, и ограничено сильно. Кроме того, неприятная особенностью ячеек памяти построенных динамически состоит ещё и в том, что при считывании данных из ячейки динамической памяти заряд, хранимый на транзисторе рассасывается, и соответственно – информация разрушается, т.е. значение хранимое ячейкой (независимо от предыдущего состояния) при считывании сбрасывается в нулевое значение.

Возникает законный вопрос: каким же образом микросхемы динамической памяти – работоспособны, каким образом могут сохранять долго записанные в микросхему данные?

Выход – найден. Работу микросхем динамической памяти обеспечивает специальный способ построения памяти на микросхемах динамической памяти, а именно способ включающий поддержку как контроллером памяти, так и самими микросхемами динамической памяти двух специфических механизмов (приемов) работы, а именно:

- а) механизма восстановления
- б) механизма регенерации

3. Механизмы восстановления и регенерации динамической памяти.

Механизм восстановления:

Механизм регенерации обеспечивает сохранение правильности (точнее, на профессиональном языке говорят "целостности") данных в микросхеме ОЗУ, по завершению процесса считывания запрошенного из ОЗУ слова данных.

При считывании данных, считываемые данные не сразу поступают приёмнику информации (например – процессору), а сначала записываются в так называемый "статический буфер" – специальные ячейки статической памяти включённые в состав либо самих микросхем динамической памяти, либо в состав контроллера памяти (микросхем управляющих работой памяти). Далее из "статического буфера" данные не только передаются приёмнику, но и вновь записываются в те динамические ячейки, из которых они только что были выбраны. Процесс возврата данных в динамические ячейки после осуществления считывания этих ячеек (разумеется с целью восстановить только что разрушенную при считывании информацию) называется "восстановлением ячеек". Более того, заметим, механизм

восстановления реализуется таким образом, что при считывании слова данных из памяти в статический буфер считывается не только запрошенное слово данных, но и множество других соседних ячеек динамической памяти (считывается, как говорят, "строка данных" содержащая, в том числе и как часть считываемых данных - данные запрошенного слова), далее возвращается (восстанавливается) в ячейки динамической памяти не только запрошенное слово – все содержимое статического буфера, вся "строка данных" из статического буфера. Таким образом, при считывании слова данных, восстанавливаются не только данные запрошенного слова, но и данные всей строки, содержащей запрошенное слово. Такое "массовое", избыточное восстановление позволяет упростить реализацию следующего нами рассматриваемого механизма – механизма регенерации.

Механизм регенерации:

Механизм регенерации обеспечивает длительное хранение записанных в ОЗУ данных, хранения в течении времени большим чем t_x - время хранения данных, вплоть до "бесконечно" длительного хранения (настолько длительного насколько длительно работает ЭВМ)

Если ОЗУ построена на динамических микросхемах, то ЭВМ устроена так, что осуществляется периодическая т. н. «регенерация», состоящая в следующем:

1. через промежутки времени t_p ("время регенерации") такие что выполняется условие $t_p < t_x$ нормальная работа ОЗУ приостанавливается;
2. контроллер памяти осуществляет цикл чтения части ячеек памяти, считываются ячейки так, чтобы оказалось считанным слово данных из каждой из строк. Считываемые данные никуда далее не передаются, но в процессе их считывания происходит не только собственно считывание, но как мы установили раньше ещё и процесс восстановления данных, данные из считанных строк возвращаются в ячейки динамической памяти и...

соответственно как бы "обновляются" (уже частично утраченный в процессе хранения заряд – восстанавливается), начинается новый цикл хранения – до следующей регенерации.

3. по окончании цикла регенерации нормальная работа ОЗУ возобновляется.

4. Достоинства, недостатки, применение статических и динамических ОЗУ.

Достоинства статических микросхем:

1. Высокая скорость - время выборки может составлять менее 1 нс.
2. Нет необходимости регенерации.
3. После окончания выборки данных, микросхема практически сразу готова к новому циклу чтения или записи. В отличие от динамической памяти, нет периода "восстановления" - в течении которого микросхема не готова к работе.

Достоинства динамических микросхем:

1. Низкое энергопотребление (в 10-100 раз меньше, чем у статической памяти).
2. Низкая стоимость (раз в 4-6 ниже, чем у статической памяти).

Применение:

Статическая память - применяется для построения ОЗУ некоторых, наиболее быстрых ЭВМ старших классов, а также для построения КЭШ-памяти.

Динамическая память - применяется для построения ОЗУ большинства ЭВМ.

5. Цикл доступа к памяти

Оперативная память, особенно динамическая, - медленное устройство и цикл доступа к ней осуществляется в течении нескольких тактов:

	1	2	..	n		1	2		k
тд						тв			
тц									

A – такт адреса. Контроллер памяти в такте передает микросхемам памяти адрес запрашиваемых данных.

D – такт данных. В этом такте микросхемы памяти передают контроллеру памяти запрошенные данные.

W1-Wn – такты ожидания. В эти такты осуществляется поиск данных, и все устройства ждут (Wait), пока микросхемы выдадут данные. Чем микросхема медленнее, тем больше число тактов ожидания потребуется.

R1-Rk – такты восстановления.

В течение этих тактов микросхема памяти восстанавливает свое состояние и не готово к началу следующего цикла. Контроллеру следует подождать, прежде чем начинать следующий цикл.

t_d – время доступа.

t_v – время восстановления. (особенно велико время восстановления для микросхем динамической памяти, оно и понятно – в течении этого времени осуществляется описанный ранее процесс возврата считанных данных – процесс восстановления)

t_{Σ} – время цикла, при этом для динамических микросхем $t_d \approx t_v$, $t_{\Sigma} \approx 2t_d$.

Выводы:

1. На выборку одного элемента данных потребуется множество тактов.

2. Считав данные, прежде чем начинать - следующую выборку, придется ждать восстановления микросхемы.

Аппаратура и материалы

Для выполнения лабораторной работы необходим персональный компьютер со следующими характеристиками: процессор Intel с тактовой частотой 1800 МГц и выше, оперативная память – не менее 128 Мбайт, свободное дисковое пространство – не менее 500 Мбайт, устройство для чтения компакт – дисков, монитор типа Super VGA (число цветов – 256) с диагональю не менее 15 ". **Программное обеспечение** – операционная система WINDOWS 98 / NT / ME / 2000 / XP, пакет MathCAD, начиная с версии 2000 и выше , Visual C++ 6.0, Borland C 3.0 и выше.

Указания по технике безопасности

Техника безопасности при выполнении лабораторной работы совпадает с общепринятой для пользователей персональных компьютеров. В частности: самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения; в случае неисправности персонального компьютера сообщить об этом обслуживающему персоналу лаборатории (оператору, администратору); соблюдать правила техники безопасности при работе с электрооборудованием; не касаться электрических розеток металлическими предметами; рабочее место пользователя персонального компьютера должно содержаться в чистоте; не разрешается возле персонального компьютера принимать пищу, напитки.

Содержание отчета и его форма

Отчет по лабораторной работе должен включать:

7. название лабораторной работы;
8. ответы на контрольные задания;
9. формулировки индивидуального задания и порядка его выполнения.

Отчет о выполнении лабораторной работы в письменном виде сдается преподавателю.

Задания

- 1) Объяснить, почему микросхемы динамической памяти нуждаются в восстановлении.
- 2) Объяснить, как осуществляется регенерация динамической памяти.
- 3) Объяснить, почему в цикле доступа к памяти - требуются такты регенерации.

ЛАБОРАТОРНАЯ РАБОТА №5. ИССЛЕДОВАНИЕ ОСНОВ ПОСТРОЕНИЯ ОЗУ. СПЕЦИАЛЬНЫЕ СХЕМЫ ПАМЯТИ

Цель работы

Цель работы – изучить методы построения ОЗУ направленные на ускорение работы с памятью, а также некоторые важные для программирования особенности подсистемы памяти ЭВМ.

Теоретическое обоснование

1. Введение

В данном разделе будут рассмотрены два способа построения ОЗУ, позволяющие сконструировать более или менее быстрое ОЗУ, используя медленные микросхемы памяти.

Способы:

- ОЗУ с чередованием (устаревший).
- Конвейерное ОЗУ (современный)

Оба эти способы полностью основаны на одной особенности компьютерных программ, а именно: в большей части случаев вслед за выборкой элементов данных i программа будет запрашивать в следующем цикле элемент данных $i+1$, то есть следующий элемент данных. Такая выборка называется “последовательной” и встречается гораздо чаще, чем “случайная” выборка, когда следующим выбираемым элементом после элемента i , оказывается какой-либо другой элемент, а не $i+1$.

Рассмотренные ниже способы организации ОЗУ рассчитаны на то, чтобы обеспечить высокую скорость хотя бы только для последовательной выборки.

2. ОЗУ с чередованием

Пусть типичная длина слова данных, запрашиваемая программами, равна L , тогда сделаем следующее:

1. Организуем ОЗУ не одним куском, а в виде двух отдельных частей, называемых **банками**, причем, если выборка слова происходит из одного банка, то другой банк не участвует, а значит не должен будет потом восстанавливаться.

2. Адреса в ОЗУ между банками распределяются следующим образом:

	Банк 1	с
	Банк 2	
	Банк 1	
	Банк 2	

1. **Банк 1** – ячейки памяти, размещенные в банке 1.

2. **Банк 2** – ячейки памяти, размещенные в банке 2.

Группы ячеек банка 1 и банка 2 чередуются между собой.

L_c – длина слова, или число ячеек памяти в группе. Величина L_c при конструировании ОЗУ выбирается равной L .

Результат: пусть из памяти происходит последовательная выборка слов длины L , тогда каждое новое слово будет выбираться из нового банка и не потребуется ждать восстановления. А второй банк, из которого выбиралось предыдущее слово, будет восстанавливаться. Добились того, что при последовательной выборке словами длиной L восстановление микросхем не играет отрицательной роли.

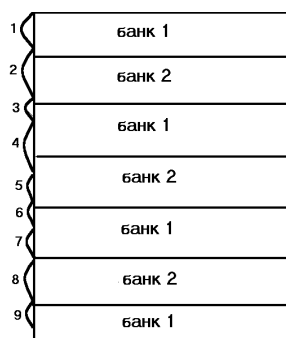
Заметим, что механизм будет срабатывать не всегда, а только при выполнении условий:

1 Выборка последовательная.

2 Все выбираемые слова расположены в пределах одного банка. Для того, чтобы при реальной работе машины эти условия почаще выполнялись, применяется специальный способ размещения данных в памяти “размещение данных с выравниванием на границу слова”.

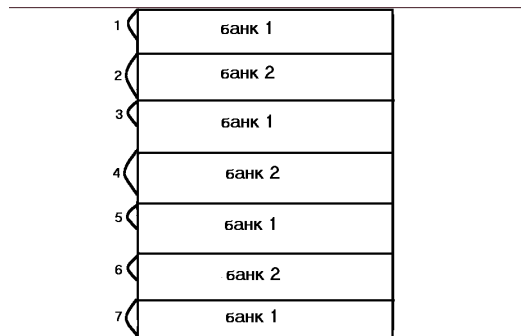
3. Выравнивание на границу слова

Размещение данных при способе размещения без выравнивания:



Результат: поскольку многие соседние слова данных оказались либо в одном банке, либо одно слово переходит через границу банка – чередование не работает и выборка данных замедляется в два раза.

Размещение данных при способе размещения с выравниваем на границу слова:



Результат: каждое очередное слово – в отдельном банке.

Недостаток размещения с выравниванием: неэкономное расходование памяти.

При программировании на языках высокого уровня транслятор по своему усмотрению размещает данные без выравнивания, либо с выравниванием. Программист может вмешаться в работу трансляторов и в настройках транслятора указать, какой способ размещения следует использовать.

4. Конвейерная память

Современная схема памяти состоит в следующем:

1. Память разбивается на множество банков, N – число банков или число ступеней конвейера.
2. Память распределяется между банками так, что слова памяти, относящиеся к различным банкам, чередуются.

3. Если поступает запрос на чтение данных из некоего банка i , контроллер кроме чтения из банка i также организует упреждающее чтение из следующих банков. Идея в том, что когда при последовательном чтении памяти дело дойдет до следующих банков, тем чтение заканчивается. Число банков N и длительность такта рассчитывается так, чтобы при последовательном чтении каждое новое слово успевало прочитаться ровно за один такт. Если очередное обращение окажется не последовательным (случайным), то есть последует не в тот банк, в котором контроллер ожидал обращение, произойдет “перезапуск” конвейера, и первое слово после перезапуска будет читаться в течение N -такта.

Аппаратура и материалы

Для выполнения лабораторной работы необходим персональный компьютер со следующими характеристиками: процессор Intel с тактовой частотой 1800 МГц и выше, оперативная память – не менее 128 Мбайт, свободное дисковое пространство – не менее 500 Мбайт, устройство для чтения компакт – дисков, монитор типа Super VGA (число цветов – 256) с диагональю не менее 15 ". **Программное обеспечение** – операционная система WINDOWS 98 / NT / ME / 2000 / XP, пакет MathCAD, начиная с версии 2000 и выше, Visual C++ 6.0, Borland C 3.0 и выше.

Указания по технике безопасности

Техника безопасности при выполнении лабораторной работы совпадает с общепринятой для пользователей персональных компьютеров. В частности: самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения; в случае неисправности

персонального компьютера сообщить об этом обслуживающему персоналу лаборатории (оператору, администратору); соблюдать правила техники безопасности при работе с электрооборудованием; не касаться электрических розеток металлическими предметами; рабочее место пользователя персонального компьютера должно содержаться в чистоте; не разрешается возле персонального компьютера принимать пищу, напитки.

Содержание отчета и его форма

Отчет по лабораторной работе должен включать:

- 10.название лабораторной работы;
- 11.ответы на контрольные задания;
- 12.формулировки индивидуального задания и порядка его выполнения.

Отчет о выполнении лабораторной работы в письменном виде сдается преподавателю.

Задания

- 1) Объяснить, для каких случаев оптимизации программы (по скорости или по размеру программы), требуется выравнивание на границу слова.
- 2) Объяснить, в чем состоят преимущества конвейерной памяти, по сравнению с памятью с чередованием.
- 3) Объяснить, почему при программировании современных вычислительных систем, часто наиболее выгодным оказывается алгоритм, обеспечивающий преимущественно последовательный доступ к памяти.

ЛАБОРАТОРНАЯ РАБОТА №6. ИССЛЕДОВАНИЕ ОСНОВ ПОСТРОЕНИЯ ОЗУ. ПОЛЬЗОВАТЕЛЬСКИЕ ХАРАКТЕРИСТИКИ ПОДСИСТЕМЫ ПАМЯТИ

Цель работы

Цель работы – получить необходимые сведения о технических характеристиках подсистемы памяти, и на основе этих знаний научиться обоснованно выбирать тип памяти для построения ЭВМ и выполнять приблизительный расчет пользовательских характеристик подсистемы памяти.

Теоретическое обоснование

1. Характеристики конвейерной памяти.

В современных ЭВМ применяется несколько систем конвейерной памяти. Рассмотрим те параметры, которыми, возможно характеризовать систему конвейерной памяти.

1.1 Производительность памяти

Производительность – это количество информации в битах или байтах, которые память может послать в центральный процессор в течение одной секунды. Данная характеристика, имеет особенно важное значение для программ, обрабатывающих большие массивы данных.

$$P = F_{\Pi} * N_{\Pi} * k * K$$

Где:

F_{Π} – тактовая частота памяти (сколько тактов выборки данных, производится за одну секунду)

k – коэффициент форсировки, некоторые виды современной конвейерной памяти позволяют в течение одного такта, по одному запросу данных передать не одну, а несколько слов данных: запрошенное слово а также последующие по адресам слова памяти.

N_{шп} – ширина шины одного модуля памяти (сколько бит или байт содержит слово данных, поступающее с модуля памяти).

K – число каналов памяти. Некоторые современные системы памяти для случаев, когда установлено несколько модулей памяти, позволяет организовать хранение данных так, что несколько модулей могут работать параллельно, и каждый из них будет выдавать часть запрошенного слова данных. Для такой памяти такие модули называются **каналами**, **K** – число установленных каналов.

1.2 Время последовательного доступа

Время последовательного доступа – это время, которое затрачивается на выборку из памяти слова из 64 бит или 8 байт. Это время можно вычислить: $t_d(n) = P/F_n$,

где **P** - число обращений к памяти, которые потребуются, чтобы выбрать все 64 бита. Если $N_{шп} = 64$, то потребуется одно обращение; если $N_{шп} < 64$ бит, то потребуется несколько обращений (целое число и, разумеется, более одного обращения).

Для памяти с шириной шины **N_{шп}** не менее 64 бит **P=1**.

Для многоканальной памяти с узкой шиной (менее 64 бит) памяти это число можно вычислить как ближайшее большее целое от:

$$P = \lceil 64 / (N_{шп} * k * K) \rceil$$

1.3 Время случайного доступа (латентность)

Время случайного доступа (иначе говоря – "латентность памяти" – это время, которое уходит на выборку первого слова данных после перезагрузки конвейера, когда не удалось угадать. Этот параметр очень важен для больших и сложных программ, обрабатывающих относительно небольшие массивы данных, поскольку при выполнении таких программ часто происходит перезагрузка конвейера.

Без учета задержек, вносимых конвейером памяти $t_d(c)$ можно вычислить:

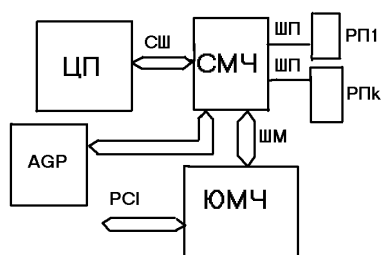
$t_d(c \min) = t_d(n) * N$, где N – число ступеней конвейера.

С учетом задержек контроллера:

$t_d(c) = t_d(c \min) + t_{зк}$, где $t_{зк}$ – дополнительные задержки контроллера – зависят не только от памяти, но и от конструкции контроллера.

Контроллер памяти в процессе работы вносит дополнительные задержки в передачу данных, в частности, дополнительную задержку при случайном доступе. Величины этих задержек, измеряемые в тактах, называются "тайминги" и записываются в виде трех или четырех цифр. Величины таймингов устанавливаются пользователем через Bios Setup. Меньшие тайминги – означают снижение задержек (латентности) увеличение скорости, большие – увеличение задержек (латентности) но одновременно и повышение надежности считывания. Для каждого модуля, специфицируются рекомендованные величины таймингов.

2. Общая схема организации материнской платы ПЭВМ



ШП – шина памяти,

ЦП – центральный процессор,

СШ – системная шина,

СМЧ – северный мост чипсета - микросхема, включающая в себя контроллер памяти и контроллер шины АГР.

РП1, РПк – разъемы для установки модулей памяти.

АГР – высокоскоростная шина и разъем для установки видеоадаптера.

ЮМЧ – южный мост чипсета – микросхема, которая содержит множество устройств, таких, как **PCI** – шина подключения плат расширения, кроме видеоадаптера.

ШМ – шина моста – высокоскоростная шина, соединяющая две микросхемы "северный мост чипсета" и "южный мост чипсета".

Вариации схемы:

1. Микросхемы СМЧ и ЮМЧ могут быть объединены в одну микросхему, тогда ШМ не используется – одночиповый вариант.

2. Компания AMD, в новой серии процессоров Athlon 64 / Opteron, для снижения задержек обращения к памяти, контроллер памяти перенесла с материнской платы в состав процессора.

Частоты:

FSB (Frequency System Bus) – частота системной шины.

Все остальные частоты в ЭВМ вычисляются как доли этой частоты.

n (FSB) – ширина шины FSB (64)

Fп – частота ядра процессора, определяет скорость вычислений процессора

Fп = FSB*Ку, где Ку – коэффициент умножения. FSB и Ку устанавливаются в BIOS.

n (шп) – ширина шины памяти.

Наиболее быстрый доступ к памяти тогда, когда $n(FSB)=n(шп)$, а $Fп = FSB$. В этом случае говорят о синхронном доступе к памяти. Если эти условия не выполняются, появляются дополнительные задержки контроллера.

3. Современные модули памяти

3.1 Сводная таблица характеристик систем памяти IBM PC

Характеристика	SDRAM	DDR DRAM	RAMBUS	DDR2 DRAM
Ширина шины модуля памяти (бит)	64	64	16	64
Коэффициент форсировки	1	2	1	4
Макс. возможное число каналов памяти	1	4	16	4
Число ступеней конвейера	4	4	32	4

Макс. возможная частота памяти (ориентировочно)	200	200	1600	200
Макс. возможная пропускная способность (ориентировочно)	1600	12800	52200	25600
Мин. возможное тд (нс) (ориентировочно)	20	20	20	20

SDRAM

Устаревающий вид динамической памяти широко применяется с 1999 по 2002 годы. Устанавливается в разъемы DIMM на материнской плате (максимум 4 модуля). Реально выпускаются модули с частотой Fп 66, 100, 133, 166. Максимальная частота Fп, на которую рассчитан модуль, присутствует в маркировке PC 66, PC 100 ...

DDR DRAM

DDR DRAM созданы на основе SDRAM и не сильно отличаются по стоимости производства и конструкции. По сравнению с SDRAM изменилось:

1. Появилась возможность выборки двух слов за один такт.
2. Устанавливается в разъемы DDR DIMM.
3. Модули памяти DDR DRAM маркируются в отличие от SDRAM не по частоте, а по удвоенной частоте.

3.4 DDR2 DRAM

Дальнейшие совершенствования DDR. По сравнению с DDR2 изменилось:

1. Появилась возможность выборки четырёх слов за один такт.
2. Устанавливается в разъемы DDR DIMM.
3. Модули памяти DDR2 DRAM маркируются в отличие от DDR не удвоенной, а учетверённой удвоенной тактовой частоте памяти.

3.5 RAMBUS

Разработано компанией RAMBUS. Стандартизована как "закрытый стандарт", что означает: производители модулей памяти этого стандарта должны перечислять компании RAMBUS "лицензионные отчисления" за предоставление права производства модулей.

По конструкции сильно отличается от SDRAM и DDR. Характерные особенности:

1. Каждый канал RAMBUS имеет узкую шину – 16 бит.
2. RAMBUS позволяет устанавливать параллельно 2 или 4 канала памяти. Современные контроллеры позволяют работать только с двумя каналами.
3. Модули памяти бывают трех типов:
 - 16-ти битные (1 канал)
 - 32-х битные (2 канала)
 - 64-х битные (4 канала)

Каждый из этих типов требует своего разъема: Rimm-16, Rimm-32, Rimm-64.

4. Если материнская плата рассчитана на 1-канальные модули, и на ней стоят разъемы Rimm-16, то для получения двух каналов необходимо установить либо 2, либо 4 модуля.

5. Применяют несколько систем маркировки, для 16 разрядных модулей наиболее распространена, маркировка по тактовой частоте (вида Rambus–800)

Аппаратура и материалы

Для выполнения лабораторной работы необходим персональный компьютер со следующими характеристиками: процессор Intel с тактовой частотой 1800 МГц и выше, оперативная память – не менее 128 Мбайт, свободное дисковое пространство – не менее 500 Мбайт, устройство для

чтения компакт – дисков, монитор типа Super VGA (число цветов – 256) с диагональю не менее 15 ". **Программное обеспечение** – операционная система WINDOWS 98 / NT / ME / 2000 / XP, пакет MathCAD, начиная с версии 2000 и выше , Visual C++ 6.0, Borland C 3.0 и выше.

Указания по технике безопасности

Техника безопасности при выполнении лабораторной работы совпадает с общепринятой для пользователей персональных компьютеров. В частности: самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения; в случае неисправности персонального компьютера сообщить об этом обслуживающему персоналу лаборатории (оператору, администратору); соблюдать правила техники безопасности при работе с электрооборудованием; не касаться электрических розеток металлическими предметами; рабочее место пользователя персонального компьютера должно содержаться в чистоте; не разрешается возле персонального компьютера принимать пищу, напитки.

Содержание отчета и его форма

Отчет по лабораторной работе должен включать:

- 13.название лабораторной работы;
- 14.ответы на контрольные задания;
- 15.формулировки индивидуального задания и порядка его выполнения.

Отчет о выполнении лабораторной работы в письменном виде сдается преподавателю.

Задания

1) Определить характеристики быстродействия для следующих видов памяти:

- a) SD DRAM, PC-133;

- б) DDR-333;
- в) Rambus-800 для двух каналов;
- г) DDR2 200МГц для одного канала.

2) Требуется память с параметрами быстродействия не хуже:

- Производительность не менее 3000МБ/с;
- t_d (с) не более 30нс.

Определить модули на какую тактовую частоту должны быть куплены для двух случаев:

- а) приобретается DDR;
- б) приобретается DDR2;
- в) приобретается Rambus.

ЛАБОРАТОРНАЯ РАБОТА №7. ИССЛЕДОВАНИЕ КОМПЬЮТЕРНЫХ СЕТЕЙ И СИСТЕМ ТЕЛЕОБРАБОТКИ

Цель работы

Цель работы – усвоить основные понятия и определения предметной области "Компьютерные сети и системы телеобработки", ознакомиться с техническими средствами построения компьютерной сети, представленными в лаборатории, и получить начальные сведения для дальнейшего изучения предмета.

Теоретическое обоснование

1. Назначение компьютерных сетей

Объединение компьютеров в сеть (network), предназначено для решения следующих пользовательских задач:

- 1) Коллективное использование данных.
- 2) Коллективное использование устройств.
- 3) Коллективное использование программ.
- 4) Поддержка электронная почта и коммуникации пользователей сети.
- 5) Коллективное решение задач.
- 6) Поддержка удалённого решения задач. "Удалённый ввод заданий" – решение задачи, с компьютера не достаточно приспособленного для решения данной задачи, переносится на другой компьютер (другой хост) сети.

2. Разновидности компьютерных сетей

В зависимости от территориальной локализации сети и производительности, используемых в сети каналов передачи данных, сети принято делить:

- 1) Локальные сети – относительно близнее расположение компьютеров (хостов) сети, при высоких скоростях передачи сетевых данных по сетевым каналам передачи данных. Позволяют решать любые пользовательские сетевые задачи, в том числе и требующие высоких скоростей передачи данных (высокой производительности использованных сетевых каналов.
- 2) Глобальные сети – дальнее расположение компьютеров сети, неограниченность территориального расположения хостов сети вплоть до объединения в общемировые сети, но при этом характерны умеренные скорости передачи сетевых данных по сетевым каналам передачи данных. Как правило, используются только для решения пользовательских задач, не требующих высокой производительности сетевых каналов, таких как электронная почта и другие задачи коммуникации пользователей работающих в сети.
- 3) Как промежуточные вариант между глобальными и локальными сетями, иногда в классификациях также выделяют так называемые "региональные сети" и "учрежденческие сети".

3. Понятия "сетевой станции" и "сервера"

Сети принято делить на:

- 1) Одноранговые сети;
- 2) Сети с выделенными серверами.

В "одноранговых сетях" все компьютеры, объединённые в сеть - потенциально равноправны, и любой из них при необходимости может быть использован как для работы в сети пользователей, так и для размещения ресурсов общего пользования, например:

- а) устройств общего пользования;
- б) программ и данных общего доступа.

В сетях с выделенными файл серверами отличают:

а) Сетевые станции (клиенты) - компьютеры в сети, используемый для работы пользователя сети, и не предназначенный для размещения ресурсов общего пользования.

б) Сервера – компьютеры в сети, предназначенные для размещения ресурсов общего пользования, и редко (в качестве исключения) используемые для организации работы пользователей сети.

4. Разновидности серверов

В крупных сетях, часто используют "специализированные сервера", например:

а) "Файл-сервер" – предназначен для размещения "файлов общего доступа", таким образом, реализует в сети простейший способ коллективного использования данных.

б) "Сервер приложений" – на серверах выполняются серверные части ПО, так называемых "клиент-серверных приложений". "Клиент серверное приложение" – программный комплекс, включающий "клиентские" и "серверные" части. При этом ПО "клиентской части" - не имеет прямого доступа к файлам, размещённым на сервере приложений, но зато может обращаться к ПО "серверной части" с запросами на поиск необходимых данных, и выполнение других требуемых клиенту операций.

в) "Коммуникационный сервер" – управляет потоками данных между данной сетью и другими сетями, обеспечивая, таким образом, "межсетевое взаимодействие".

г) "Принт-сервер" – используется для подключения "принтеров общего пользования" управляя "потокм заданий" на принтеры поступающим с других компьютеров сети, формирующий и поддерживающий "очереди печати", и выполняющий другие функции, необходимые для поддержки коллективного использования принтеров. Аналогично, для поддержки других устройств общего пользования могут быть использованы другие "выделенные серверы", например для организации коллективного использования модемов - "факс серверы".

д) "Почтовые серверы" – для поддержки использования в сети "электронной почты" и других технологий "электронной коммуникации пользователей".

В небольших сетях функции всех выше перечисленных (а также и других возможных, но неупомянутых специализированных серверов), могут реализоваться с использованием единственного сервера, который, в таком случае, часто называют "главный сервер сети".

5. Понятие системы телеобработки

"Система телеобработки" – может рассматриваться как частный случай сети, в которой степень централизации ресурсов особенно высока, и особая роль не только в организации коллективного использования сетевых ресурсов, но и в "диспетчеризации работы хостов сети", "организации коллективного доступа к среде передачи" - лежит на так называемом "сервере системы телеобработки". В классических сетях решение данных задач - разделяется между всеми хостами входящими в состав сети (реализуется с использованием "сетевых контроллеров" и

других специализированных аппаратных устройств, не обязательно привязанных к серверу сети).

6. Введение в "сетевую модель OSI"

OSI – Open System Interconnect (в русском переводе – Модель Открытых Систем).

OSI – международный стандарт ISO. Принят в 1990 г. Описывает:

- терминологию, рекомендованную к употреблению при описании сетей и сетевых технологий;
- стандартное разбиение всех средств сетевой работы по функциям на семь уровней, от нижних, тесно связанных с аппаратурой, к верхним, связанным с прикладным ПО и методами работы с прикладным ПО;
- интерфейсы между уровнями.

На момент принятия OSI существовало множество различных коммерческих стандартов на построение сетей, не совместимых между собой, и существовали большие проблемы с организацией взаимодействия различных сетей, построенных по разным стандартам.

OSI задал направление в котором развиваются коммерческие стандарты в плане унификации, обеспечения интерфейсов межсетевого взаимодействия, унификации терминологии.

7. Понятия о сетевом протоколе

Протокол – это детальное формализованное описание того, каким образом выполняется та или иная функция и каким образом сопрягаются те или иные компоненты.

Стопка протоколов – это набор протоколов, полностью описывающий какую-либо сложную систему.

Известны несколько популярных стопок протоколов, часто называемых просто протоколами, описывающих различные международные и фирменные стандарты на построение сети, например:

- 1) TCP/IP. Родился как часть ОС UNIX, но за 30 лет получил огромное распространение. Реализован в большинстве компьютерных систем, и является основой построения Internet.
- 2) Apple Talk. стандартный протокол для сети компьютеров Apple Macintosh. Интересен тем, что значительно лучше TCP/IP поддерживает передачу непрерывных данных.
- 3) NetBIOS. Стандарт времен DOS. В настоящее время, поддержка NetBIOS необходима для программ DOS.
- 4) NPX. Стандарт фирмы Novell. В начале 90-х г.г. конкурировал с TCP/IP. До сих пор на основе его использования построено множество локальных сетей.

8. Общая структура OSI

Все задачи, выполняемые при передачи сообщений по сети, модель OSI разбивает на семь уровней (layers).

Нумерация уровней: от нижних - к верхним:

7	Application (Прикладной)
6	Data Presentation (Представления данных)
5	Session (Сессионный, сеансовый)
4	Transport (Транспортный)
3	Network (Сетевой)
2	Channel (Канальный)
1	Physical (Физический)

Уровни 6 и 7 относят к "верхним".

Уровни 3, 4 и 5 – "средние". Уровни системного программного обеспечения, обеспечивающего работу сети. Системное ПО, в иных системах, также может быть поддержано специализированными аппаратными средствами.

Уровни 1-2 – "нижние". Уровни реализуемые сетевой аппаратурой.

Уровни 1-4 называются транспортной службой.

В процессе передачи сообщения переходят с одного уровня на другой (проходят по стеку). В процессе прохождения по стеку на каждом уровне выполняются специфические для данного уровня операции.

В процессе передачи сообщения то опускаются с верхних уровней к нижним, то поднимаются с нижних – к верхним. Процесс продвижения к нижним уровням называется "конвертацией сообщений", а процесс продвижения к верхним уровням – "деконвертацией сообщений".

Если компьютер А передает данные компьютеру В, на компьютере А происходит конвертация, на компьютере В – деконвертация.

9. Функции отдельных уровней

9.1 Прикладной уровень

Никаких преобразований сообщений уровень не предписывает.

Стандарты уровня описывают:

- а) логическую структуру сети, т. е. то, как пользователь видит сеть. Логическая структура – это виртуальный объект, реальное устройство сети может значительно отличаться от логической структуры;
- б) способы реализации тех или иных функций в прикладных программах;

в) способы реализации тех или иных функций в сетевых ОС и браузерах.

9.2 Уровень представления данных

На данном уровне описаны:

- а) рекомендованные для передачи по сети форматы данных;
- б) рекомендованные алгоритмы трансляции данных из форматов, не стандартизированных для передачи по сети - в стандартные;
- в) методы шифрования данных. Имеется в виду т. н. высокоуровневое шифрование, осуществляемое прикладными программами или средствами шифрования ОС. Следует отличать от т. н. низкоуровневого шифрования, которое осуществляет аппаратура передачи данных по сети;
- г) высокоуровневое помехозащищенное кодирование. Следует отличать от низкоуровневого помехозащищенного кодирования.

9.3 Сеансовый уровень

Стандарты описывают способ подключения прикладных программ и высокоуровневых средств операционной системы к транспортной службе сети, в том числе такие вопросы, как:

- а) инициация сеанса связи;
- б) разрыв сеанса связи;
- в) приостановка/возобновление сеанса связи;

9.4 Транспортный уровень

Программное обеспечение транспортного уровня решает две важнейшие задачи:

- а) коммутация сообщений;
- б) трансляция адресов.

Коммутация сообщений. Существуют различные технологии передачи сообщений. Например, в одних случаях сообщения передаются целиком, в других – разбиваются на части (*пакеты*). Кроме того, в одних случаях коммутация ведется по заранее не подготовленному каналу связи, в других – по заранее подготовленному (*скоммутированному*) каналу связи. То, как будет происходить коммутация сообщений, зависит от настройки транспортной службы, а конкретно от настройки программ коммутации.

Трансляция адресов. Пользователь, работающий в сети, видит логическую структуру сети и пользуется т. н. «логическими» сетевыми адресами. Логические сетевые адреса не отражают структуру сети. В дальнейшем для передачи данных необходимо преобразовать логические адреса в т. н. физические адреса. Физические адреса менее понятны и менее информативны для пользователя, но в них отражена физическая структура сети.

Сетевой уровень

На данном уровне решаются две главные задачи:

- а) маршрутизация;
- б) преобразование физических адресов в канальные.

Маршрутизация. Если сеть достаточно сложна, и в ней возможна передача сообщений по более чем одному альтернативному маршруту - требуется "маршрутизация", а именно - определение оптимального маршрута, из числа возможных.

Преобразование адресов. Аппаратура канала (сетевая плата, модем, телефонная линия и другие аппаратные средства, связывающие два компьютера) при передаче данных по каналу используется т. н. *канальные адреса*, которые никак не связаны с общей структурой сети и используются только в данном конкретном канале.

Таким образом: физические адреса, связанные с общей структурой сети, перед передачей по каналу преобразуются в канальные адреса (другое название канальных адресов – локальные физические адреса).

Канальный уровень

На канальном уровне описана аппаратура, осуществляющая передачу данных по каналу, и все специфические алгоритмы, которые используются для передачи данных, например:

- а) алгоритмы разделения линий связи;
- б) алгоритмы коллективного использования линий связи;
- в) алгоритмы замены неправильно переданных данных ;
- г) алгоритмы низкоуровневого шифрования;
- д) алгоритмы низкоуровневого помехозащищенного кодирования.

Часть этих алгоритмов специфична для компьютерных сетей, а часть общеупотребительна на любых компьютерных интерфейсах.

9.7 Физический уровень

На этом уровне описываются:

- а) стандарты на параметры используемых кабельных систем;
- б) стандарты на разъемы, врезки и другие средства подключения кабельных систем;
- в) электрические характеристики используемых сигналов;
- г) алгоритмы синхронизации передачи данных (поскольку - зависимы от типа используемого кабеля).

10. Модель OSI применительно к локальным сетям

Протокол IEEE802.0 уточняет модель OSI для случая локальных сетей.

Важнейшее уточнение – алгоритмы и протоколы канального уровня разбиты дополнительно на 2 подуровня: LLC и MAC:

	LLC
	MAC

где:

LLC (logical link connect) – протокол связи логического звена;

MAC (medium access control) – контроль доступа к среде.

К подуровню LLC относятся алгоритмы, реализация которых не зависит от типа сетевого контроллера, например: низкоуровневое шифрование, помехозащищенное кодирование, замена неправильно переданных данных.

К подуровню MAC относятся алгоритмы, реализация которых зависит или прямо связана с типом сетевого контроллера. Например, алгоритмы доступа к разделяемым каналам, коллективным линиям связи и другие.

Аппаратура и материалы

Для выполнения лабораторной работы необходим персональный компьютер со следующими характеристиками: процессор Intel с тактовой частотой 1800 МГц и выше, оперативная память – не менее 128 Мбайт, свободное дисковое пространство – не менее 500 Мбайт, устройство для чтения компакт – дисков, монитор типа Super VGA (число цветов – 256) с диагональю не менее 15 ". **Программное обеспечение** – операционная

система WINDOWS 98 / NT / ME / 2000 / XP, пакет MathCAD, начиная с версии 2000 и выше, Visual C++ 6.0, Borland C 3.0 и выше.

Указания по технике безопасности

Техника безопасности при выполнении лабораторной работы совпадает с общепринятой для пользователей персональных компьютеров. В частности: самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения; в случае неисправности персонального компьютера сообщить об этом обслуживающему персоналу лаборатории (оператору, администратору); соблюдать правила техники безопасности при работе с электрооборудованием; не касаться электрических розеток металлическими предметами; рабочее место пользователя персонального компьютера должно содержаться в чистоте; не разрешается возле персонального компьютера принимать пищу, напитки.

Содержание отчета и его форма

Отчет по лабораторной работе должен включать:

- 16.название лабораторной работы;
- 17.ответы на контрольные задания;
- 18.формулировки индивидуального задания и порядка его выполнения.

Отчет о выполнении лабораторной работы в письменном виде сдается преподавателю.

Задания

- 1) соображения о возможных сравнительных преимуществах и недостатках одноранговых сетей, и сетей с выделенными серверами.
- 2) Составить список соответствия между описанными в материалах лабораторной работы видами серверов, и типовыми задачами, решаемыми в компьютерных сетях.

3) Сформулировать причины, по которым потребовалась международная стандартизация и унификация сетевых систем и интерфейсов.

4) Задачи каких из уровней модели OSI решаются программными, какие - аппаратными средствами, в решении каких - может быть использовано как системное ПО, так и специализированная аппаратура.

5) Выделите в операционной системе части, задачи решаемые которыми относятся к задачам, описанным на верхних уровнях модели OSI и часть, задачи решаемые которыми относятся к задачам, описанным на верхних уровнях модели OSI;

6) Какие три типа адреса используются в сетевых системах, на каких уровнях модели OSI используется каждый из этих трёх типов адресов;

7) С использованием каких адресов производится маршрутизация данных, передаваемых по сети;

8) Свяжите материал работы с материалом первых лабораторных работ, тех в которых рассматривались алгоритмы помехозащищённого кодирования. Какие из алгоритмов помехозащищённого кодирования могли бы быть использованы в реализации высокоуровневого помехозащищённого кодирования, какие — в реализации низкоуровневого помехозащищённого кодирования;

ЛАБОРАТОРНАЯ РАБОТА №8. ИЗУЧЕНИЕ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ. ОСНОВНЫЕ ПОНЯТИЯ И КЛАССИФИКАЦИИ

Цель работы

Цель работы – пока без изучения особенностей конструкций тех или иных конкретных современных технологий построения многопроцессорных (мультимикропроцессорных) ЭВМ,, разобраться с основными особенностями применения мультимикропроцессорных вычислителей, с их классификациями.

Теоретическое обоснование

1. Введение

Высокую производительность и реактивность пытаются получить путем использования нескольких микропроцессоров. Развитие мультимикропроцессорных систем началось в 70-е годы XX века. Пионеры: IBM, CDC, Cray, Burroughs, DEC, HP, Interdata. В последние годы применяются при построении ЭВМ класса "сервер" и так называемых "супер-ЭВМ" (супер-ЭВМ – мультимикропроцессорная ЭВМ, ориентированная на решение сложных научных задач, преимущественно векторных). В последние годы применение мультимикропроцессорных ЭВМ расширяется, и мультимикропроцессорными становятся и массовые компьютеры, вплоть до "персональных ЭВМ".

2. Условия эффективности применения для мультимикропроцессорного вычислителя

Независимо от числа микропроцессоров в составе ЭВМ, один поток инструкций - всегда выполняется только на одном микропроцессоре. Поэтому, увеличение числа микропроцессоров имеет эффект только в следующих случаях:

- а) многозадачная обработка (параллельное выполнение большого числа задач);

б) выполнение так называемых "распараллеленных" задач (т.е. задач запрограммированных как множество отдельных и параллельно работающих потоков).

Многие современные задачи тяжело поддаются сколь эффективному распараллеливанию, многие из тех, что потенциально таки могут быть более-менее эффективно распараллелены, на практике, пока не распараллелены. Дальнейшие перспективы более широкого распространения многопроцессорных вычислителей сильно связаны с дальнейшими отраслевыми успехами в поисках новых путей более эффективного распараллеливания, и успехов в практическом распараллеливании наиболее типичных, массовых задач.

3. Геометрическое и алгоритмическое распараллеливание

Грубо и приблизительно, можно выделить два вида распараллеливания:

- а) геометрическое;
- б) алгоритмическое.

Типичными задачами, поддающимися так называемому **"геометрическому распараллеливанию"**, являются, к примеру, задачи анализа и синтеза изображений. Каждому процессору и работающему на этом процессоре программному процессу передаётся участок изображения, и соответственно - различные участки обрабатываются на разных процессорах, распределение нагрузки между процессорами естественным образом равномерное. Такое распараллеливание, достаточно часто, реализуется и относительно просто, и эффективно. По типу "геометрического распараллеливания", распараллеливаются и многие другие типы задач, такие, в которых равномерное распределение нагрузки между процессорами естественно и очевидно проистекает из особенностей самой задачи.

По типу **"алгоритмического распараллеливания"** распараллеливаются те задачи, в которых возможности и способы параллельного решения не присутствует в явном, очевидном виде. "Алгоритмическое распараллеливание" состоит в вычленении (и часто очень непростом) в алгоритме решаемой задачи, тех ветвей и операций, которые так могут быть выполнены параллельно, и в соответствующем программировании задач. Искусство алгоритмического распараллеливания является отдельной дисциплиной в рамках предмета "технологии программирования", и требует особых навыков от программиста. Попытки реализации в компиляторах технологий так называемого **"автоматического распараллеливания"** (то есть создания компиляторов способных автоматически, без особых усилий со стороны программиста, находить в исходном нераспараллеленном коде - параллельные ветви и операции, и автоматически преобразовать не распараллеленное решение - в распараллеленное) пока не увенчались особыми успехами. Данные разработки находятся на стадии исследований.

4. Степень связанности распараллеленной задачи

Достаточно условно, но можно выделить два класса задач, поддающихся распараллеливанию:

- а) сильносвязанные распараллеленные задачи
- б) слабосвязанные распараллеленные задачи.

Слабосвязанные задачи – те из распараллеленных задач, в которых выполняющиеся параллельно отдельные ветви (потoki), в значительной степени - независимы друг от друга (обмен данными между потоками редок, и постоянная синхронизация (согласование хода выполнения) потоков не требуется).

Сильносвязанные задачи – те из распараллеленных задач, в которых выполняющиеся параллельно отдельные ветви (потoki), в значительной

степени – связаны друг с другом (обмен данными между потоками требуется часто, или требуется постоянная синхронизация (согласование хода выполнения) потоков). К таким задачам, как правило, относятся, например, такие задачи в которых в потоках - обрабатываются одни и те же массивы данных.

Степень связанности распараллеленной задачи существенно влияет на требования, предъявляемые к многопроцессорному вычислителю:

Слабосвязанные задачи могут достаточно эффективно решаться на любом многопроцессорном вычислителе, даже и таком, в котором отдельные процессоры связаны - слабо (например реализованы как отдельные ЭВМ связанные между собой скоростной локальной сетью, такие многопроцессорные ЭВМ принято называть **"кластерными вычислителями"**). "Кластерные вычислители" очень часто рассматривают как особую разновидность многопроцессорной ЭВМ, но многие авторы, рассматривают этот класс вычислителей и отдельно, не относя к "многопроцессорным ЭВМ".

Сильносвязанные задачи могут эффективно решаться только на таких многопроцессорных ЭВМ, в которых обеспечены **"широкие"** (т.е. способные передать в течении интервала времени немалое кол-во данных) и **"короткие"** (или, другими словами, **"низколатентные"** т.е. способные доставлять данные от процессора к процессору без существенных задержек доставки) каналы межпроцессорной связи. Такую связь возможно реализовать лишь используя для связи процессоров - специальные высокоскоростные каналы передачи данных (**"высокоскоростные каналы прямой межпроцессорной связи"**), или используя **"общий кэш"** (или, другими словами, **"разделяемый кэш"** - кэш, прямой доступ к которому реализован для нескольких процессоров). Как вариант, для межпроцессорной связи может использоваться и **"общее ОЗУ"** (**"разделяемое ОЗУ"**), но в этом случае обмен данных между процессорами, таки, явно получается медленней чем через общий кэш, и во многих случаях, медленней чем при

использовании специальных высокоскоростных каналов прямой межпроцессорной связи.

5. Классификация многопроцессорных ЭВМ

Классификация по симметричности/асимметричности

Симметричные многопроцессорные ЭВМ - содержат в составе множество универсальных и одинаковых процессоров, которые, в принципе - могут быть взаимозаменяемы, любую ветвь задачи принципиально можно решать - на любом из процессоров.

Ассиметричные многопроцессорные ЭВМ - содержат множество специализированных процессоров, где отдельные процессоры - выделены для отдельных подзадач, и взаимонезаменяемы.

Преимущества симметричных:

— При решении любой распараллеленной (на количество ветвей, соответствующее или превышающее число процессоров) задачи возможно использовать все процессора многопроцессорной ЭВМ

— в симметричных обязательно реализована, но потенциально реализуема "**система резервирования**", позволяющая перераспределять потоки между процессорами при выходе из строя части процессоров или других связанных с отдельными процессорами подсистем. Разумеется система резервирования повышает так называемую "живучесть вычислителя" (защищённость от отказов аппаратуры);

— в симметричных вычислителях при наличии избыточного числа процессоров переключение с одной задачи на другую или запись новой не обязательно требует переключения контекста, вместо этого новая задача может быть запущена и на одном из простаивающих процессоров (если такие имеются), эта возможность потенциально способна повысить реактивность вычислительной системы.

Преимущества ассиметричных:

- реализация специализированных процессоров очень часто позволяет получить особенно высокую производительность именно на "критически трудоёмких задачах" (именно там, где она - особенно важна), и производительность - намного превышающую производительность универсального процессора, производительность - недостижимую для симметричных многопроцессорных ЭВМ;
- специализированные вычислители относительно недороги, поэтому часто возможно получить не только высокую производительность, но и лучшее соотношение производительность/цена.

Классификация по степени связанности:

- слабосвязанные многопроцессорные ЭВМ;
- сильносвязанные многопроцессорные ЭВМ ;
- кустовые многопроцессорные ЭВМ;

Слабосвязанные многопроцессорные ЭВМ – множество процессоров (до десятков тысяч) каким-то образом связанных между собой относительно низкоскоростными каналами передачи данных. В предельном случае каналами связи может выступать локальная сеть. В этом случае систему называют "кластерной".

Достоинства: относительно недорого можно построить многопроцессорную ЭВМ, поддерживающую очень большое число процессоров, и получить высокую производительность на слабосвязанных задачах и в условиях развитой многозадачности.

Недостатки: малополезны при решении сильносвязанных задач.

Сильносвязанные многопроцессорные ЭВМ - содержат относительно небольшое число процессоров, но зато - хорошо связаны между собой высокоскоростными и низколатентными каналами передачи

данных, или такие, в которых процессоры **работают с общим полем памяти** (ОЗУ или кэш).

Достоинства: достаточно эффективны для решения сильносвязанных задач.

Недостатки: организация быстрой связи между множеством процессоров - дорогостояща и технически осложнена, поэтому число процессоров в таких системах пока ограничена величиной не более 64-128 процессоров, а наиболее применимы сильносвязанные многопроцессорные ЭВМ с числом процессоров не более 16-ти.

Кустовые многопроцессорные ЭВМ – многопроцессорный вычислитель представляет из себя множество относительно слабо связанных между собой (по типу как процессора в слабосвязанных многопроцессорных ЭВМ) так называемых "**кустов**", но где каждый "куст" представляет из себя не один процессор, а сильносвязанную многопроцессорную ЭВМ.

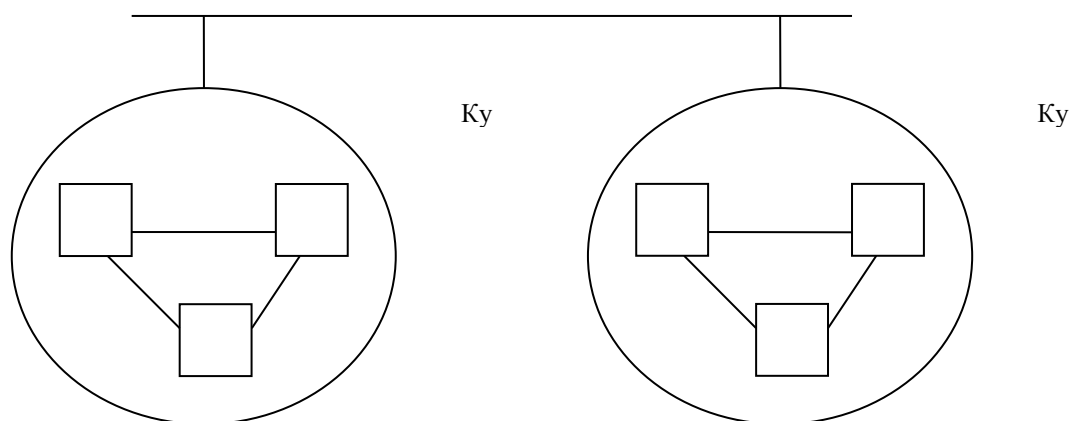


Рисунок 1. Схема построения кустовых многопроцессорных ЭВМ

П – процессор

В – высокоскоростная связь

Н – низкоскоростная связь

В современных системах реально применяются кусты размером до 128 процессоров, и используется число кустов в Кустовой многопроцессорной ЭВМ - до нескольких тысяч.

Канонические и неканонические многопроцессорные ЭВМ

Канонические многопроцессорные ЭВМ – или "традиционные многопроцессорные ЭВМ", такие, в которых связь между процессами и блоками ОЗУ – не изменяема, не зависит ни от характера решаемой задачи, ни от хода вычислительного процесса. Все современные широко используемые ЭВМ являются каноническими.

Неканонические многопроцессорные ЭВМ – так многими принято называть некоторые экспериментальные разработки, такие, в которых связь между процессорами и ОЗУ не является статичной, пересматривается, перенастраивается в зависимости и от характера решаемой задачи, или даже от хода вычислительного процесса.

Идея состоит в следующем: если практически невозможно организовать сильные связи между большим числом процессоров по принципу "каждый с каждым", то организовать такие высокоскоростные связи, которые данным вычислительным процессом - наиболее востребованы. Разумеется, такой вычислитель должен либо иметь возможности "аппаратной перенастройки под задачу", либо - окажется не универсальным.

6. Многоядерные процессоры

Если не акцентировать внимание на некоторых непринципиальных отличиях, **многоядерный процессор** аналогичен многопроцессорной вычислительной системе с несколькими процессорами, но только

размещенными либо на одном кристалле, либо на нескольких отдельных кристаллах (подложках), но – в одном корпусе.

Если попытаться предсказать возможные достоинства/недостатки системы построенной на многоядерном процессоре по сравнению с многопроцессорной системой на равном числе отдельных процессоров, можно отметить:

- 1) в многоядерной не сложно организовать быструю связь между ядрами, высокоскоростные каналы между ядрами размещёнными в одном корпусе, или даже общее поле кэш-памяти. Благодаря этому многоядерный процессор может получить преимущества во многих (но не всех) сильносвязанных задачах;
- 2) но с другой стороны, поскольку несколько ядер многоядерного процессора пользуются общим каналом доступа к ОЗУ, неизбежна конкуренция между ядрами за доступ к ОЗУ, и не каждое ядро всегда будет иметь необходимый доступ к ОЗУ (особенно при увеличении числа ядер многоядерного процессора). Из-за этого ЭВМ на многоядерном процессоре может оказаться в невыгодном положении по сравнению с многопроцессорной ЭВМ, при решении задач требующих быстрого доступа к локальным данным вычислительных процессов, или требующих больших объёмов перекачки данных в каналах доступа к ОЗУ.

Дальнейший прогресс в увеличении числа ядер, зависит как от того, насколько удачным получится распараллеливание популярных задач, так и от роста характеристик канала памяти (ОЗУ). Могут потребоваться и новые технологии построения ОЗУ.

Аппаратура и материалы

Для выполнения лабораторной работы необходим персональный компьютер со следующими характеристиками: процессор Intel с тактовой

частотой 1800 МГц и выше, оперативная память – не менее 128 Мбайт, свободное дисковое пространство – не менее 500 Мбайт, устройство для чтения компакт – дисков, монитор типа Super VGA (число цветов – 256) с диагональю не менее 15 ". **Программное обеспечение** – операционная система WINDOWS 98 / NT / ME / 2000 / XP, пакет MathCAD, начиная с версии 2000 и выше , Visual C++ 6.0, Borland C 3.0 и выше.

Указания по технике безопасности

Техника безопасности при выполнении лабораторной работы совпадает с общепринятой для пользователей персональных компьютеров. В частности: самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения; в случае неисправности персонального компьютера сообщить об этом обслуживающему персоналу лаборатории (оператору, администратору); соблюдать правила техники безопасности при работе с электрооборудованием; не касаться электрических розеток металлическими предметами; рабочее место пользователя персонального компьютера должно содержаться в чистоте; не разрешается возле персонального компьютера принимать пищу, напитки.

Содержание отчета и его форма

Отчет по лабораторной работе должен включать:

- 19.название лабораторной работы;
- 20.ответы на контрольные задания;
- 21.формулировки индивидуального задания и порядка его выполнения.

Отчет о выполнении лабораторной работы в письменном виде сдается преподавателю.

Задания

- 1) Является ли ваша домашняя ПЭВМ "мультипроцессорной" системой? Если да, то - какого типа? Какие процессоры в ней использованы?
- 2) Почему при построении "вычислительных систем повышенной надёжности" применяют симметричные многопроцессорные архитектуры?
- 3) В чем отличие ЭВМ построенной с использованием многоядерного процессора от многопроцессорной ЭВМ.
- 4) В решении каких задач многоядерные процессоры и многоядерные ЭВМ – эффективны, при решении каких задач многоядерная, многопроцессорная ЭВМ "пустая трата денег"?
- 5) Попробуйте проанализировать состав задач решаемых вами на вашей домашней ПЭВМ, попробуйте догадаться (или определить по данным предоставляемым пользователю операционной системой) какие из этих задач легко - поддаются распараллеливанию (и распараллелены), какие – нет. Составьте соответствующие списки задач.
- 6) Попробуйте проанализировать состав задач решаемых вами на вашей домашней ПЭВМ, попробуйте догадаться (или определить по данным предоставляемым пользователю операционной системой) какие из этих задач распараллеливаются геометрическим распараллеливанием.

ЛАБОРАТОРНАЯ РАБОТА №9. ИЗУЧЕНИЕ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ. ОСНОВНЫЕ СОВРЕМЕННЫЕ АРХИТЕКТУРНЫЕ РЕАЛИЗАЦИИ

Цель работы

Цель работы – изучение особенностей конструкций тех или иных конкретных современных технологий построения многопроцессорных (мультимикропроцессорных) ЭВМ.

Теоретическое обоснование

1. Многопроцессорная архитектура АМР

АМР (Asymmetric MultiProcessor) – протокол организации асимметричных вычислительных систем средней степени связанности, использует для межпроцессорной связи шину ввода/вывода.

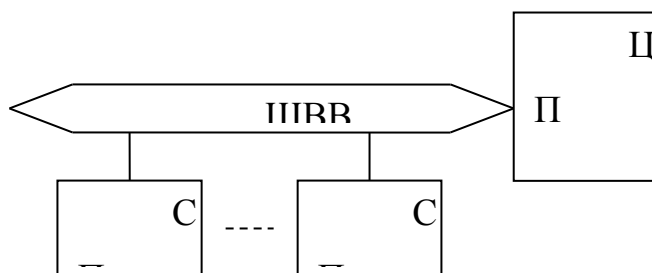


Рисунок 1. Многопроцессорная архитектура АМР

Разработана в 1987 году и используется во всех современных ПЭВМ.

Допускает максимально до восьми специализированных процессоров. На практике ПЭВМ могут иметь дополнительные ограничения, снижающие максимально доступное число процессоров.

2. Многопроцессорная архитектура SMP

SMP (Symmetric MultiProcessor) – стандарт на построение канонических сильносвязанных мультипроцессорных вычислителей, в которых взаимодействие процессоров происходит посредством общего поля памяти.

Широко применяется при построении относительно простых сильносвязанных систем с использованием Intel Xeon и Itanium, а также при построении кустов.

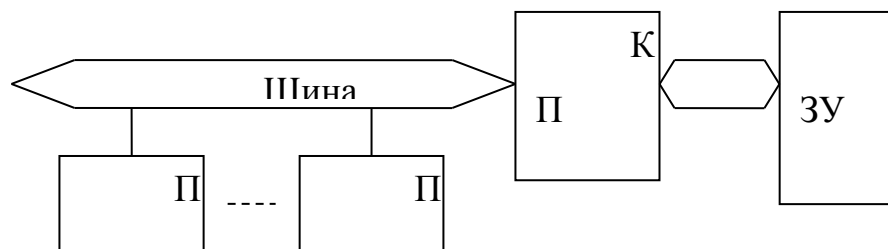


Рисунок 2. Многопроцессорная архитектура SMP

КП – контроллер памяти;

$П_1 \dots П_N$ – процессора, $N \leq 8$;

ОЗУ – общее ОЗУ, в котором хранятся как локальные данные, так и общие данные.

Поддерживает до восьми процессоров. Шина памяти общая для всех процессоров, ОЗУ общее, в котором хранятся как локальные данные каждого процессора, так и общие данные.

Достоинства:

- не велики затраты на реализацию;
- достаточно сильная связь между процессорами через общее поле памяти.

Недостатки:

- из-за конкуренции между процессорами за доступ к единственному ОЗУ, и по единственной шине – процессоры ограничены в доступе к памяти, что может приводить к значительному снижению эффективности многопроцессорной системы, что особенно заметно при большом числе процессоров или на задачах, требующих интенсивного доступа к памяти.

Зависимость производительности SMP от числа процессоров:

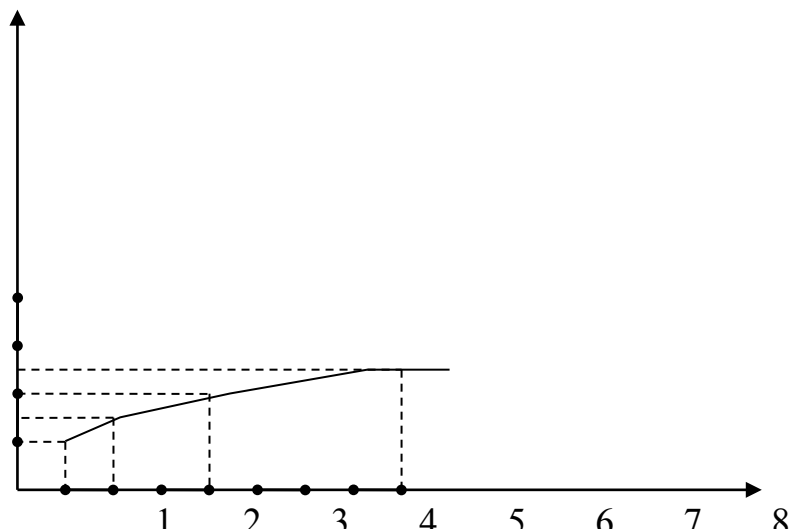


Рисунок 3. Зависимость производительности SMP от числа процессоров

n – количество процессоров, K – степень увеличения производительности по сравнению с однопроцессорной системой.

Видим, что рост производительности непропорционален числу процессоров. Из-за того, что в последние годы рост производительности процессов опережает темпы роста характеристик памяти, конфликт между процессорами в SMP системах все более усугубляется. На настоящий момент практически не применимы SMP-системы с числом процессоров большим четырех. Еще более, перспектива развития SMP ухудшается с распространением многоядерных процессоров. Многоядерные процессоры

предъявляют повышенные требования к подсистеме памяти, и организовать эффективное использование SMP не удастся.

3. Двухшинная SMP архитектура

Так как традиционная SMP архитектура все более перестает удовлетворять современным требованиям из-за обострения конкуренции за доступ процессора к памяти, в 2006 году Intel был предложен универсальный вариант SMP – DSMP.

Отличия:

- 1) процессоры разделены на две группы, каждая из которых осуществляет доступ к ОЗУ по собственной шине;
- 2) каждая группа процессоров получает в распоряжение собственное ОЗУ;
- 3) обмен данными между процессорами группы или чтение/запись данных ОЗУ осуществляется по специальной связи межгрупповой связи.

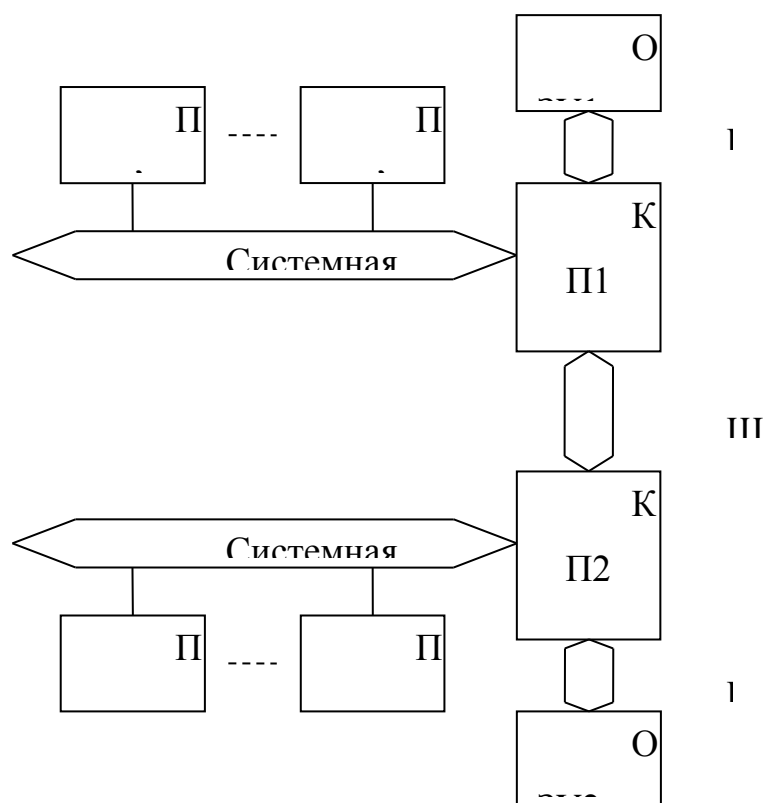


Рисунок 4. Двухшинная многопроцессорная архитектура SMP

N изменяется от одного до восьми. ШП1, ШП2 – шина памяти, ШМС – высокоскоростная низколатентная шина межгрупповой связи.

Результат: позволяет эффективно реализовать двухпроцессорную систему и достаточно эффективно 4-х и 8-ми процессорную системы.

4. S-Matrix многопроцессорная ЭВМ

Используется в последние годы для построения сильносвязанных многопроцессорных ЭВМ и кустов. В S-Matrix система поддерживает множество процессоров и ОЗУ, коммутируемых произвольно через так называемую коммутирующую матрицу.

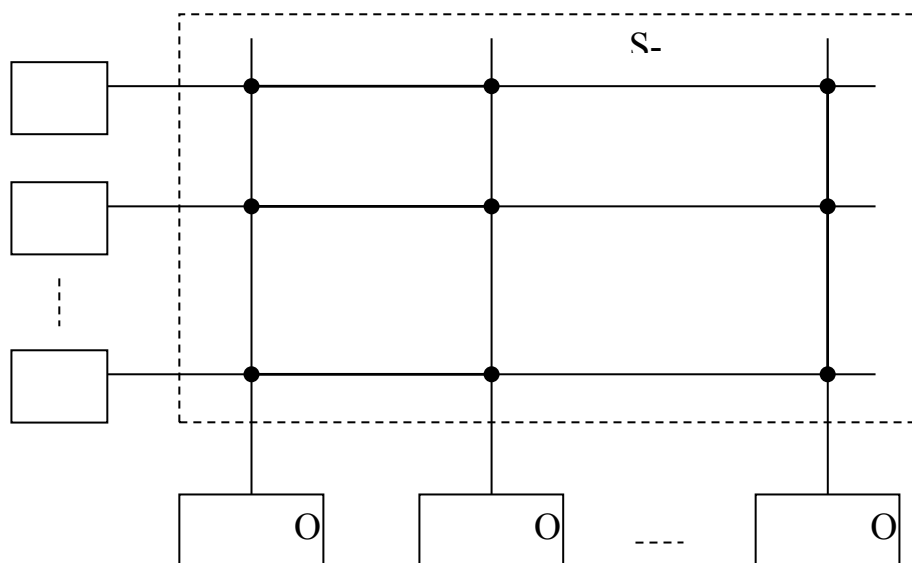


Рисунок 5. Система S-Matrix

S-Matrix – коммутирующая матрица, к – ключи коммутирующей матрицы.

Замыканием соответствующих ключей соединяются нужные шины и процессор динамически получает доступ к одному из ОЗУ. Распределение ОЗУ между процессорами происходит с участием операционной системы.

Достоинства: обеспечивается тесная связь между процессорами, доступ каждого процессора к собственному ОЗУ. С другой стороны, перекоммутированием шин возможно обеспечить запись/чтение любого из ОЗУ любым из процессоров.

Недостатки: сложность реализации коммутатора S-Matrix, увеличение латентности, S-Matrix предполагает наличие достаточно длинных шин, что затрудняет передачу данных с высокой скоростью и низкой латентностью.

Применение. Используется при построении некоторых сильносвязанных многопроцессорных ЭВМ старших классов. Более эффективны при решении определённых классов сильносвязанных задач. Во многих других задачах не так эффективны, т.к. сложно организовать быстрое коммутирование, и сложность организации высокоскоростной передачи данных через коммутатор все более ограничивает производительность ЭВМ. Поскольку от коммутаторов требуется работа на всё более высоких частотах, требования к коммутаторам постоянно растут, а организовать достаточно эффективный коммутатор становится все сложнее и сложнее.

5. Многопроцессорная архитектура HyperTransport (HT)

С начала 70-х годов – момента появления первых многопроцессорных ЭВМ – известна архитектура многопроцессорного вычислителя, обычно называемая «архитектура с неравномерным доступом к памяти».

Для этой архитектуры характерно:

- 1) каждому из процессоров передается в распоряжение собственное локальное ОЗУ процессора. В нем хранятся как локальные данные вычислительного процесса, так и общие, обрабатываемые разными процессами;

2) межпроцессорная связь организуется за счет использования так или иначе сконфигурированных межпроцессорных каналов связи, по которым происходит как прямой обмен данными между процессорами, так и доступ процессоров к ОЗУ другого процессора.

Например, может выглядеть так:

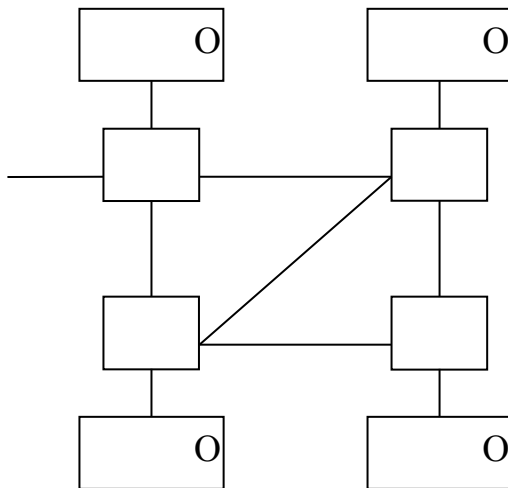


Рисунок 6. Архитектура HyperTransport

С момента появления таких архитектур их достоинством считается быстрый доступ к локальным данным при относительно медленном доступе к глобальным данным или данным других процессоров.

Эта архитектура разработана AMD в 2000 году и является одним из вариантов классической многопроцессорной архитектуры с неравномерным доступом к памяти.

Уникальной является исключительно разработка межпроцессорной шины, получившей название HyperTransport и характеризующейся рекордно низкой латентностью при передаче данных. Низкая латентность и высокая ПСП этой шины позволили AMD сократить время доступа к данным других процессоров и пропагандировать эту архитектуру, как архитектуру «с почти равномерным доступом к памяти», т.е. считать архитектурой нового класса.

Если быть точнее, шина HyperTransport может работать в двух режимах:

- а) асинхронном. Шина работает подобно традиционным шинам и обеспечивает универсальный доступ к различным устройствам, причем многоточечный доступ – доступ по одной шине к множеству устройств. Этот режим используется для подключения к процессору периферийных контроллеров;
- б) синхронный. Осуществляет доступ к единственному соседнему процессору в режиме сверхнизкой латентности. Используется для межпроцессорной связи.

Реализация архитектуры гипертранспорт первых поколений предусматривала, наличие до трех шин HyperTransport, каждая из которых может работать как в синхронном, так и асинхронном режиме.

В серверных процессорах доступны все три шины, в десктопных две из 3-х шин заблокированы и компьютеры лишены возможности использования многопроцессорной работы.

Другой особенностью архитектуры HyperTransport явилась глубокая интеграция памяти в состав процессора, что позволило избежать промежуточных операций по каналу ОЗУ – КП – процессор и получить сниженную примерно в полтора раза (и, в некоторых случаях, и более) латентность доступа к памяти.

Типичная двухпроцессорная конфигурация.

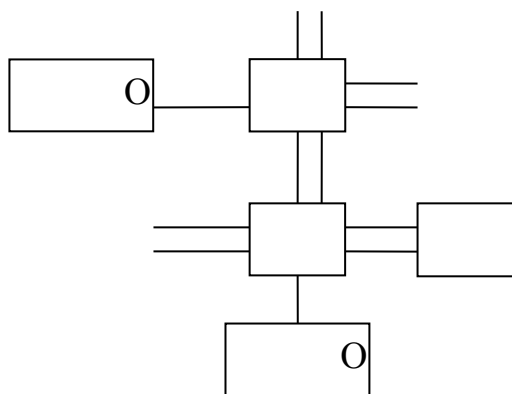


Рисунок 7. Двухпроцессорная конфигурация.

На рисунке:

П – процессора

ПК – пересерийные контроллеры

Ш – выделенные шины к ОЗУ

При необходимости можно использовать еще три шины НТ для организации многоканального доступа к периферийным контроллерам.

Четырехпроцессорная архитектура (сокращенная: без ОЗУ):

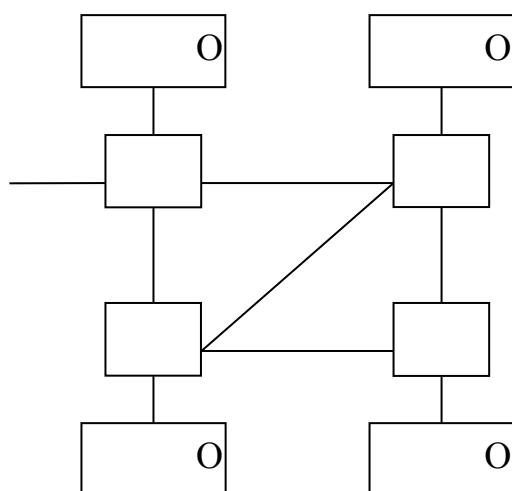


Рисунок 8. Четырехпроцессорная конфигурация.

Восьмипроцессорная конфигурация (один из возможных вариантов):

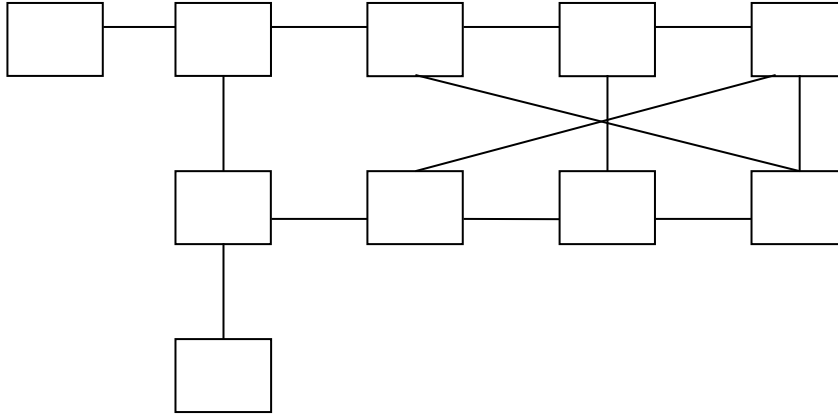


Рисунок 9. Восьмипроцессорная конфигурация.

Достоинства:

- 1) низкая латентность доступа к локальной памяти;
- 2) при увеличении числа процессоров каждый новый процессор получает собственное ОЗУ и не возникает конкуренции доступа процессоров к общему ОЗУ;
- 3) связь процессоров с ОЗУ без посредников приводит к повышению скорости доступа и снижению латентности.

Недостатки:

- 1) высока скорость доступа лишь к локальным данным, или данным соседнего по линку НТ процессора;
- 2) при увеличении числа процессоров в системе среднее число линков НТ через которые в связь между процессорами увеличивается и доступ к нелокальным данным замедляется. Таким образом, с увеличением числа процессоров свойства системы всё менее соответствуют сильносвязанной многопроцессорной системе.

Применение. Применяется все более широко с 2003 года при том, что есть задачи как более, так и менее оптимальные для HT. В целом показывает большие преимущества по сравнению с SMP, особенно при увеличении числа процессоров больше двух.

6. Развитие Hyper Transport

В новых реализациях обеспечиваются следующие улучшения:

- 1) увеличение тактовой частоты шин HT и соответственно возрастает ПСП и уменьшается латентность;
- 2) число линков HT, поддерживаемых одним процессором возрастает с трех до четырех;
- 3) появляется возможность так называемого расщепления линков, при которой один линк разделяется на два, каждый из которых с вдвое меньшей ПСП. Таким образом, число линков увеличивается до восьми, что позволяет организовать многопроцессорную систему со связью процессоров каждый к каждому с числом процессоров до восьми;
- 4) новые процессоры, поддерживающие HT2 будут совместимы сверху вниз с системным окружением (материнская плата) старого образца, поддерживающим HT, но при этом будут поддерживать лишь старый протокол HT.

7. Архитектура CSI

Разрабатывается Intel. Призвана заменить архитектуру SMP в многопроцессорных

ЭВМ на базе процессоров Intel. По своим архитектурным решениям совпадает с архитектурой HT, но прямо не совместима с HT. Появление

запланировано на 2008 год. Первоначально планируется использование на системах с процессором Itanium, а затем и перенос на x86.

8. Архитектура Деер

Деер – название научно-исследовательского проекта компании IBM. Одна из разработок центра, разработка новой архитектуры многопроцессорного вычислителя, реализующей свойства сильносвязанных вычислительных систем, но при числе процессоров, значительно превышающем число процессоров в современных сильносвязанных системах.

Основные черты:

- 1) каждый процессор имеет собственное ОЗУ для хранения локальных данных;
- 2) между процессорами устанавливаются связи с использованием относительно небольшого числа высокоскоростных низколатентных шин, подобных HT;
- 3) в отличие от HT, распределение линков между процессорами - гибкое, т.е. способ соединения процессоров зависит от решаемой задачи и может изменяться по ходу решения задачи;
- 4) системное окружение обеспечивает требуемую перекоммутацию процессоров.

Недостатки:

- 1) сложности организации системного окружения, обеспечивающего перекоммутацию линков;
- 2) принципиально не решены еще некоторые научные вопросы, связанные с оптимизацией графа межпроцессорной связи.

Применение: Построено и испытываются несколько поколений системы Deep. Deep Blue известна тем, что выиграла шахматный матч у Г. Каспарова. Некоторые разработки предполагают до 64000 процессоров в системе в целом и до 1000 процессоров в кусте.

Аппаратура и материалы

Для выполнения лабораторной работы необходим персональный компьютер со следующими характеристиками: процессор Intel с тактовой частотой 1800 МГц и выше, оперативная память – не менее 128 Мбайт, свободное дисковое пространство – не менее 500 Мбайт, устройство для чтения компакт – дисков, монитор типа Super VGA (число цветов – 256) с диагональю не менее 15 ". **Программное обеспечение** – операционная система WINDOWS 98 / NT / ME / 2000 / XP, пакет MathCAD, начиная с версии 2000 и выше , Visual C++ 6.0, Borland C 3.0 и выше.

Указания по технике безопасности

Техника безопасности при выполнении лабораторной работы совпадает с общепринятой для пользователей персональных компьютеров. В частности: самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения; в случае неисправности персонального компьютера сообщить об этом обслуживающему персоналу лаборатории (оператору, администратору); соблюдать правила техники безопасности при работе с электрооборудованием; не касаться электрических розеток металлическими предметами; рабочее место пользователя персонального компьютера должно содержаться в чистоте; не разрешается возле персонального компьютера принимать пищу, напитки.

Содержание отчета и его форма

Отчет по лабораторной работе должен включать:

- 22.название лабораторной работы;
- 23.ответы на контрольные задания;
- 24.формулировки индивидуального задания и порядка его выполнения.

Отчет о выполнении лабораторной работы в письменном виде сдается преподавателю.

Задания:

- 1) Как, согласно классификациям описанным в предыдущей работе, возможно классифицировать каждую из представленных архитектур многопроцессорных ЭВМ?
- 2) Какая из архитектур в наибольшей степени соответствует идеям сильносвязанной многопроцессорной системы?
- 3) Почему развитие многоядерных процессоров снижает шансы на применение в будущем архитектур использующих общую шину?
- 4) Подумайте, почему создание сильносвязанной многопроцессорной ЭВМ затруднено в рамках канонических архитектур, и почему в ЭВМ проекта Deep используется перекоммутация линков?

Рекомендованная литература

СПИСОК ОСНОВНОЙ ЛИТЕРАТУРЫ

1. Б.Я.Цилькер, С.А. Орлов Организация ЭВМ и систем: учебник для вузов. – СПб.: Питер, 2006. – 668с.: ил.
2. А.П.Пятибратов, Л.П.Гудыно, А.А.Кириченко. Вычислительные системы, сети и телекоммуникации. / Учебник под ред. А.П.Пятибратова / М.: Финансы и статистика, 2001
3. А.В.Гордеев Операционные системы. Учебник для вузов. 2-е изд. – СПб.: Питер, 2007. – 416с.: ил.
4. Информатика: Базовый курс: Учеб. пособие. Рек.МО/ Под ред. С. В. Симоновича. - СПб:Питер,2000,2001. - 640с. - (Учебник для вузов). - Библиогр.: с.620-622
5. Бройдо В. Л.Вычислительные системы, сети и телекоммуникации: Учеб. пособие. Доп.МО. - СПб.:Питер,2003. - 688с.

СПИСОК ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ

1. Э.Таненбаум Архитектура компьютера. – СПб. Питер, 2002.: 704с.: ил.
2. М.Гук Аппаратные средства локальных сетей: энциклопедия / СПб: Питер, 2001

3. Р. Карриган Микропроцессоры / М: Инфо-пресс, 1999
- 4 Э.Танненбаум Современные операционные системы. 2-е издание. – СПб.: Питер, 2006. -1038с.:ил.
5. П.В.Румянцев Азбука программирования в Win 32 API / М: Радио и связь,1999
6. Р.Линекер, Т.Арчер Программирование для Windows. Библия разработчика / М: Диалектика,1999
7. Финогенов, К.Г. Основы языка Ассемблера : учебный курс / К.Г. Финогенов. - М. : Радио и связь : Горячая линия - Телеком, 1999. - 288 с.
8. Дроздова В. И.Физические основы записи и хранения информации [Текст]: Учеб. пособие/ М-о РФ. СевКавГТУ. - Ставрополь:СевКавГТУ,2004. - 125 с. - Библиогр.: с. 125-126
9. Новиков Ю. В.Основы микропроцессорной техники [Текст]: Учеб. пособие/ Ю. В. Новиков, П. К. Скоробогатов. - Изд. 2-е, испр. - М.:Интернет-Университет Информационных Технологий,2004. - 440 с. - ("Основы информационных технологий"). - Библиогр.: с.431-432
10. Романец, Ю. В. Защита информации в компьютерных системах и сетях / Ю. В. Романец, П. А. Тимофеев, В. Ф. Шаньгин ; под ред. В. Ф. Шаньгина. - М. : Радио и связь, 1999. - 328 с. - Библиогр.: с. 319-325. - ISBN 5-256-01436-6

11. Могилев А. В. Информатика: Учеб. пособие. Рек.МО/ Под ред.: Е. К. Хеннера. - 2-е изд., стер. - М.: Академия, 2003. - 816с. - Библиогр.: в тексте

12. Мезенцева, О. С. (СевКавГТУ). Информатика и программирование : учеб. пособие : (курс лекций) / О. С. Мезенцева ; Мин-во образования и науки Рос. Федерации, ГОУ ВПО Сев. Кав. гос. техн. ун-т, Ч. 2. - Ставрополь : Издательство СевКавГТУ, 2006. - 105 с. - Библиогр.: с. 103

13. В.А.Велихов, К.С.Сафронов, Б.К.Леонтьев Компьютерные сети: учебное пособие по администрированию локальных и объединённых сетей. 3-е издание, дополненное и исправленное. – М.: ЗАО Новый Издательский Дом, 2005. – 304.с

14. В.Е.Карпов, К.А.Коньков Основы операционных систем. Курс лекций. Учебное пособие под редакцией В.П.Иванникова. – М.: ИНТУИРРУ, 2005. – 536с.

15. А.Танненбаум, А.Вудхолл Операционные системы: разработка и реализация (+CD). Классика CS. – СПб.: Питер, 2006 – 576с.:ил.

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

Методические указания
по организации и проведению самостоятельной работы
по дисциплине
«Физические основы информационных технологий»
для студентов направления подготовки
09.03.02 «Информационные системы и технологии»
направленность (профиль)
**«Прикладное программирование в интеллектуальных информационных
системах»**

Ставрополь, 2026

1. Общие положения

Самостоятельная работа – планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов (СРС) в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения.

К основным видам самостоятельной работы студентов относятся:

- формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);
- написание докладов;
- подготовка к семинарам, практическим и лабораторным работам, их оформление;
- составление аннотированного списка статей из соответствующих журналов по отраслям знаний (педагогических, психологических, методических и др.);
- выполнение учебно-исследовательских работ, проектная деятельность;
- подготовка практических разработок и рекомендаций по решению проблемной ситуации;
- выполнение домашних заданий в виде решения отдельных задач, проведения типовых расчетов, расчетно-компьютерных и индивидуальных работ по отдельным разделам содержания дисциплин и т.д.;
- компьютерный текущий самоконтроль и контроль успеваемости на базе электронных обучающих и аттестующих тестов;
- выполнение курсовых работ (проектов) в рамках дисциплин;
- выполнение выпускной квалификационной работы и др.

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей изучаемой дисциплины, объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);
- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

Самостоятельная работа по дисциплине «Физические основы информационных технологий» направлена на формирование следующих **компетенций**:

Код, формулировка компетенции	Код, формулировка индикатора
ОПК-3 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической	ОПК-3 _{ид-3.1} – Использует принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и

культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности
	ОПК-3 _{ид-3.2} – Решает стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной
	ОПК-3 _{ид-3.3} – Осуществляет подготовку обзоров, аннотаций, составления рефератов, научных докладов, публикаций, и библиографии по научно-исследовательской работе с учетом требований информационной безопасности

2. Цель и задачи самостоятельной работы

Ведущая цель организации и осуществления СРС совпадает с целью обучения студента – формирование набора общенаучных, профессиональных и специальных компетенций будущего бакалавра по соответствующему направлению подготовки

При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности. Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

3. Технологическая карта самостоятельной работы студента

Коды реализуемых компетенций	Вид деятельности студентов	Средства и технологии оценки	Объем часов, в том числе (акад.)		
			СРС	Контактная работа с преподавателям	Всего
3 семестр					
	Самостоятельное изучение литературы и источников	Собеседование	2	2	4
	Подготовка лабораторным занятиям	Защита ЛР	64	4	68
Итого за 3 семестр			66	6	72
Итого			66 / -		72

4. Порядок выполнения самостоятельной работы студентом

4.1. 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 на сколько осознанна 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.

4.2. Методические рекомендации по подготовке к практическим и лабораторным занятиям

Для того чтобы практические и лабораторные занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по вычитанному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

4.3. Методические рекомендации по самопроверке знаний

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется, провести самопроверку усвоенных знаний, ответив на контрольные вопросы по изученной теме.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала - умение решать задачи или пройти тестирование по пройденному материалу. Однако следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

4.4. Методические рекомендации по написанию научных текстов (докладов, докладов, эссе, научных статей и т.д.)

Перед тем, как приступить к написанию научного текста, важно разобраться, какова истинная цель вашего научного текста - это поможет вам разумно распределить свои силы и время.

Во-первых, сначала нужно определиться с идеей научного текста, а для этого необходимо научиться либо относиться к разным явлениям и фактам несколько

критически (своя идея – как иная точка зрения), либо научиться увлекаться какими-то известными идеями, которые нуждаются в доработке (идея – как оптимистическая позиция и направленность на дальнейшее совершенствование уже известного). Во-вторых, научиться организовывать свое время, ведь, как известно, свободное (от всяких глупостей) время – важнейшее условие настоящего творчества, для него наконец-то появляется время. Иногда именно на организацию такого времени уходит немалая часть сил и талантов.

Писать следует ясно и понятно, стараясь основные положения формулировать четко и недвусмысленно (чтобы и самому понятно было), а также стремясь структурировать свой текст. Каждый раз надо представлять, что ваш текст будет кто-то читать и ему захочется сориентироваться в нем, быстро находить ответы на интересующие вопросы (заодно представьте себя на месте такого человека). Понятно, что работа, написанная «сплошным текстом» (без заголовков, без выделения крупным шрифтом наиболее важным мест и т. п.), у культурного читателя должна вызывать брезгливость и даже жалость к автору (исключения составляют некоторые древние тексты, когда и жанр был иной и к текстам относились иначе, да и самих текстов было гораздо меньше – не то, что в эпоху «информационного взрыва» и соответствующего «информационного мусора»).

Объем текста и различные оформительские требования во многом зависят от принятых в конкретном учебном заведении порядков.

Доклад – это самостоятельное исследование студентом определенной проблемы, комплекса взаимосвязанных вопросов.

Доклад не должна составляться из фрагментов статей, монографий, пособий. Кроме простого изложения фактов и цитат, в доклад е должно проявляться авторское видение проблемы и ее решения.

Рассмотрим основные этапы подготовки
а студентом.

Выполнение доклада начинается с выбора темы.

Затем студент приходит на первую консультацию к руководителю, которая предусматривает:

- обсуждение цели и задач работы, основных моментов избранной темы;
- консультирование по вопросам подбора литературы;
- составление предварительного плана.

Следующим этапом является работа с литературой. Необходимая литература подбирается студентом самостоятельно.

После подбора литературы целесообразно сделать рабочий вариант плана работы. В нем нужно выделить основные вопросы темы и параграфы, раскрывающие их содержание.

Составленный список литературы и предварительный вариант плана уточняются, согласуются на очередной консультации с руководителем.

Затем начинается следующий этап работы – изучение литературы. Только внимательно читая и конспектируя литературу, можно разобраться в основных вопросах темы и подготовиться к самостоятельному (авторскому) изложению содержания доклада. Конспектируя первоисточники, необходимо отразить основную идею автора и его позицию по исследуемому вопросу, выявить проблемы и наметить задачи для дальнейшего изучения данных проблем.

Систематизация и анализ изученной литературы по проблеме исследования позволяют студенту написать работу.

Рабочий вариант текста доклада предоставляется руководителю на проверку. На основе рабочего варианта текста руководитель вместе со студентом обсуждает возможности доработки текста, его оформление. После доработки доклад сдается на кафедру для его оценивания руководителем.

Требования к написанию доклада

Написание 1 доклада является обязательным условием выполнения плана СРС по любой дисциплине профессионального цикла.

Тема доклада может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласовать с преподавателем.

Доклад должен быть написан научным языком.

Объем доклада должен составлять 20-25 стр.

Структура доклада:

- Введение (не более 3-4 страниц). Во введении необходимо обосновать выбор темы, ее актуальность, очертить область исследования, объект исследования, основные цели и задачи исследования.
- Основная часть состоит из 2-3 разделов. В них раскрывается суть исследуемой проблемы, проводится обзор мировой литературы и источников Интернет по предмету исследования, в котором дается характеристика степени разработанности проблемы и авторская аналитическая оценка основных теоретических подходов к ее решению. Изложение материала не должно ограничиваться лишь описательным подходом к раскрытию выбранной темы. Оно также должно содержать собственное видение рассматриваемой проблемы и изложение собственной точки зрения на возможные пути ее решения.
- Заключение (1-2 страницы). В заключении кратко излагаются достигнутые при изучении проблемы цели, перспективы развития исследуемого вопроса
- Список использованной литературы (не меньше 10 источников), в алфавитном порядке, оформленный в соответствии с принятыми правилами. В список использованной литературы рекомендуется включать работы отечественных и зарубежных авторов, в том числе статьи, опубликованные в научных журналах в течение последних 3-х лет и ссылки на ресурсы сети Интернет.
- Приложение (при необходимости).

Требования к оформлению:

- текст с одной стороны листа;
- шрифт Times New Roman;
- кегль шрифта 14;
- межстрочное расстояние 1,5;
- поля: сверху 2,5 см, снизу – 2,5 см, слева - 3 см, справа 1,5 см;
- доклад должен быть представлен в сброшюрованном виде.

Порядок защиты доклада:

Защита доклада проводится на практических занятиях, после окончания работы студента над ним и исправления всех недочетов, выявленных преподавателем в ходе консультаций. На защиту доклада отводится 5-7 минут времени, в ходе которого студент должен показать свободное владение материалом по заявленной теме. При защите доклада приветствуется использование мультимедиа-презентации.

Оценка доклада

Доклад оценивается по следующим критериям:

- соблюдение требований к его оформлению;
- необходимость и достаточность для раскрытия темы приведенной в тексте доклада информации;
- умение студента свободно излагать основные идеи, отраженные в докладе;
- способность студента понять суть задаваемых преподавателем и сокурсниками вопросов и сформулировать точные ответы на них.

Критерии оценки:

Оценка «отлично» выставляется студенту, если в докладе студент исчерпывающе, последовательно, четко и логически стройно излагает материал; свободно справляется с задачами, вопросами и другими видами применения знаний; использует для написания доклада современные научные материалы; анализирует полученную информацию; проявляет самостоятельность при написании доклада.

Оценка «хорошо» выставляется студенту, если качество выполнения доклада достаточно высокое. Студент твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопросы по теме доклада.

Оценка «удовлетворительно» выставляется студенту, если материал доклада излагается частично, но пробелы не носят существенного характера, студент допускает неточности и ошибки при защите доклада, дает недостаточно правильные формулировки, наблюдаются нарушения логической последовательности в изложении материала.

Оценка «неудовлетворительно» выставляется студенту, если он не подготовил доклад или допустил существенные ошибки. Студент неуверенно излагает материал доклада, не отвечает на вопросы преподавателя.

4.5. Методические рекомендации по выполнению исследовательских проектов

Исследовательская проектная работа – это групповая работа, для выполнения которой необходим выбор и приложение научной методики к поставленной задаче, получение собственного теоретического или экспериментального материала, на основании которого необходимо провести анализ и сделать выводы об исследуемом явлении. Выполнение проекта – это всегда коллективная, творческая практическая работа, предназначенная для получения определенного продукта или научно-технического результата. Такая работа подразумевает четкое, однозначное формирование поставленной задачи, определение сроков выполнения намеченного, определение требований к разрабатываемому объекту.

Выполнение 1 группового проекта является обязательным условием выполнения самостоятельной работы по любой дисциплине профессионального цикла. Тема проектного задания может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласовать с преподавателем.

Требования по выполнению и оформлению проекта

При выполнении проекта приветствуется работа в группе (2-3 человека). Проект – это исследовательская работа, в ходе которой студенты должны продемонстрировать владение навыками научного исследования, умения проводить анализ, обобщать информацию, делать выводы, предлагать свои решения проблемы, рассматриваемой в проекте.

При подготовке материалов проекта студенты должны продемонстрировать владение современными методами компьютерной обработки данных.

Критерии оценки работы участника проекта.

Для каждого из участников проекта оцениваются:

- профессиональные теоретические знания в соответствующей области;
- умение работать со справочной и научной литературой, осуществлять поиск необходимой информации в Интернет;
- умение работать с техническими средствами;
- умение пользоваться соответствующими выполняемому проекту информационными технологиями;
- умение готовить материалы проекта для презентации: составлять и редактировать тексты, формировать презентацию проекта;
- умение работать в команде;
- умение публично представлять результаты собственной деятельности;

- коммуникабельность, инициативность, творческие способности.

Критерии выставления оценки участникам проекта

Оценка	Профессиональные компетенции	Компетенции, связанные с использованием соответствующих выполняемому проекту технических средств и информационных технологий	Иные универсальные компетенции (коммуникабельность, инициативность, умение работать в «команде», управленческие навыки и т.д.)	Отчетность
«Отлично»	Работа выполнена на высоком профессиональном уровне. Представленный материал в основном фактически верен, допускаются негрубые фактические неточности. Студент свободно отвечает на вопросы, связанные с проектом.	Технические средства и информационные технологии освоены и использованы для реализации проекта полностью	Студент проявил инициативу, творческий подход, способность к выполнению сложных заданий, навыки работы в коллективе, организационные способности.	Проект представлен полностью и в срок.
«Хорошо»	Работа выполнена на достаточно высоком профессиональном уровне. Допущено до 4–5 фактических ошибок. Студент отвечает на вопросы, связанные с проектом, но недостаточно полно.	Обнаруживаются некоторые ошибки в использовании соответствующих технических средств и информационных технологий	Студент достаточно полно, но без инициативы и творческих находок выполнил возложенные на него задачи.	Проект представлен достаточно полно и в срок, но с некоторыми недоработками.
«Удовлетворительно»	Уровень недостаточно высок. Допущено до 8 фактических ошибок. Студент может ответить лишь на некоторые из заданных вопросов, связанных с проектом.	Обнаруживает недостаточное владение навыками работы с техническими средствами и соответствующим и информационным и технологиями	Студент выполнил большую часть возложенной на него работы.	Проект сдан со значительным опозданием (более недели) и не полностью
«Неудовлетворительно»	Работа не выполнена или выполнена на низком уровне. Допущено более 8 фактических	Навыков работы с техническими средствами нет, информационные технологии не	Студент практически не работал, не выполнил свои задачи или	Проект не сдан.

Оценка	Профессиональные компетенции	Компетенции, связанные с использованием соответствующих выполняемому проекту технических средств и информационных технологий	Иные универсальные компетенции (коммуникабельность, инициативность, умение работать в «команде», управленческие навыки и т.д.)	Отчетность
	ошибок. Ответы на связанные с проектом вопросы обнаруживают непонимание предмета и отсутствие ориентации в материале проекта.	освоены	выполнил лишь отдельные не существенные поручения в групповом проекте.	

Студенты должны: защитить проект в режиме презентации, предъявить файлы выполненного проекта, уметь рассказать о технологиях, использованных ими при выполнении проекта, дать оценку работы каждого члена группы (если проект групповой).

4.6. Методические рекомендации по подготовке к экзаменам и зачетам

Изучение многих общепрофессиональных и специальных дисциплин завершается экзаменом. Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене студент демонстрирует то, что он приобрел в процессе обучения по конкретной учебной дисциплине.

Экзаменационная сессия - это серия экзаменов, установленных учебным планом. Между экзаменами интервал 3-4 дня. Не следует думать, что 3-4 дня достаточно для успешной подготовки к экзаменам.

В эти 3-4 дня нужно систематизировать уже имеющиеся знания. На консультации перед экзаменом студентов познакомят с основными требованиями, ответят на возникшие у них вопросы. Поэтому посещение консультаций обязательно.

Требования к организации подготовки к экзаменам те же, что и при занятиях в течение семестра, но соблюдаться они должны более строго. Во-первых, очень важно соблюдение режима дня; сон не менее 8 часов в сутки, занятия заканчиваются не позднее, чем за 2-3 часа до сна. Оптимальное время занятий - утренние и дневные часы. В перерывах между занятиями рекомендуются прогулки на свежем воздухе, неустойчивые занятия спортом. Во-вторых, наличие хороших собственных конспектов лекций. Даже в том случае, если была пропущена какая-либо лекция, необходимо во время ее восстановить (переписать ее на кафедре), обдумать, снять возникшие вопросы для того, чтобы запоминание материала было осознанным. В-третьих, при подготовке к экзаменам у студента должен быть хороший учебник или конспект литературы, прочитанной по указанию преподавателя в течение семестра. Здесь можно эффективно использовать листы опорных сигналов.

Вначале следует просмотреть весь материал по сдаваемой дисциплине, отметить для себя трудные вопросы. Обязательно в них разобраться. В заключение еще раз целесообразно повторить основные положения, используя при этом листы опорных сигналов.

Систематическая подготовка к занятиям в течение семестра позволит использовать время экзаменационной сессии для систематизации знаний.

Контроль самостоятельной работы студентов

Контроль самостоятельной работы проводится преподавателем в аудитории.

Предусмотрены следующие виды контроля: собеседование, оценка доклада, оценка презентации, оценка участия в круглом столе, оценка выполнения проекта.

Подробные критерии оценивания компетенций приведены в Фонде оценочных средств для проведения текущей и промежуточной аттестации.

Список литературы для выполнения СРС

Перечень основной литературы:

1. Рыбальченко, М.В. Архитектура информационных систем : учебное пособие / М.В. Рыбальченко ; Министерство образования и науки Российской Федерации, Южный федеральный университет. - Таганрог : Издательство Южного федерального университета, 2020. - Ч. 1. - 92 с. - Библиогр. в кн. - ISBN 978-5-9275-1765-7 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=462011>, экземпляров неограничено

2. Цветкова, А. В. Информатика и информационные технологии Электронный ресурс : Учебное пособие для СПО / А. В. Цветкова. - Информатика и информационные технологии, 2020-08-30. - Саратов : Научная книга, 2019. - 190 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-9758-1891-1, экземпляров неограничено

Перечень дополнительной литературы:

1. Ипатова, Э.Р. Методологии и технологии системного проектирования информационных систем : учебник / Э.Р. Ипатова, Ю.В. Ипатов. - 2-е изд., стер. - Москва : Издательство «Флинта», 2017. - 257 с. : табл., схем. - (Информационные технологии). - Библиогр.: с. 95-96 - ISBN 978-5-89349-978-0 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=79551>

Методическая литература:

1. Методические рекомендации для самостоятельной работы студентов по дисциплине «Физические основы информационных технологий»
2. Методические указания к лабораторным работам по дисциплине «Физические основы информационных технологий»

Интернет-ресурсы:

1. <http://el.ncfu.ru/> – система управления обучением ФГАОУ ВО СКФУ. Дистанционная поддержка дисциплины «Физические основы информационных технологий».
2. <http://www.un.org> - Сайт ООН Информационно-коммуникационные технологии.
3. <http://www.intuit.ru> – Интернет-Университет Компьютерных технологий.