

Оглавление

1 Направления развития ЭВМ. История механических и электромеханических приборов для вычислений.	2
2 Поколения ЭВМ.	2
3 Типы архитектур вычислительных систем.	2
4 Шинная организация ЭВМ.	3
5 Иерархия шин. Арбитраж шин.	4
6 Характеристики памяти. Классификация запоминающих устройств.	5
7 RAID-массивы.	5
8 Способы ввода-вывода информации	9
9 Машинная команда, структура.	9
10 Показатели производительности.	11
11 Процессор. Типы архитектур.	12
12 Способы адресации.	13
13 Устройство управления. Жесткое и микропрограммное управление.	14
14 Периферийные устройства ввода ЭВМ.	15
15 Периферийные устройства вывода ЭВМ.	15
16 Внутренняя структура процессора i8086.	17
17 Назначение входов процессора i8086.	18
18 Назначение выходов процессора i8086.	18
19 Схема демультипликации шин адреса и данных.	19
20 Организация памяти i8086. 21 Сегментация памяти i8086.	20
22 Организация прерываний на примере i8086.	22

1 Направления развития ЭВМ. История механических и электромеханических приборов для вычислений.

- 1) Эволюционная модификация ЭВМ Фон Неймана – А) последовательная обработка данных. Б) увеличение производительности за счет улучшения технических характеристик.
- 2) Параллельные вычислительные системы – А) параллельная обработка информации. Б) повышение производительности за счет увеличения количества вычислителей и улучшения технических характеристик отдельного вычислителя.

История вычислительной техники.

VI–V вв. до н.э. — Абак

- 1670 — Лейбниц — арифметическая машина
- 1808 — Жаккард — станки с перфокартами
- 1812 — Бэббедж — механический мозг
- 1878 — Чебышев — арифмометр
- 1938 — Цузе — машины Z1,Z2,Z3

2 Поколения ЭВМ.

- 1) 1945-1955 электронная лампа, перфокарта. (Военные цели)
- 2) 1955-1965 транзистор, магнитная лента.
- 3) 1965-1975 ИС (интегральная схема) малой интеграции, дискета.
- 4) 1975-н.в. БИС (большая интегральная схема) и СБИС, жесткий диск.

3 Типы архитектур вычислительных систем.

Классификация Флинна:

- **SISD — ОКОД**
(одиночный поток команд, одиночный поток данных)
- **MISD — МКОД**
(множественный поток команд, одиночный поток данных)
- **SIMD — ОКМД**
(одиночный поток команд, множественный поток данных)
- **MIMD — МКМД**
(множественный поток команд, множественный поток данных)

Классификация Хокни:



4 Шинная организация ЭВМ.

Шина – совокупность линий, идущих параллельно и имеющих одинаковое функциональное назначение.

Шина данных предназначена для пересылки кодов обрабатываемых данных, а также машинных кодов команд между устройствами ЭВМ.

Шина адреса несет адрес ячейки памяти или порта ввода-вывода, который взаимодействует с микропроцессором.

Шина управления несет сигналы управления, обеспечивающие правильное взаимодействие блоков ЭВМ друг с другом и с внешней средой.

Архитектуры:

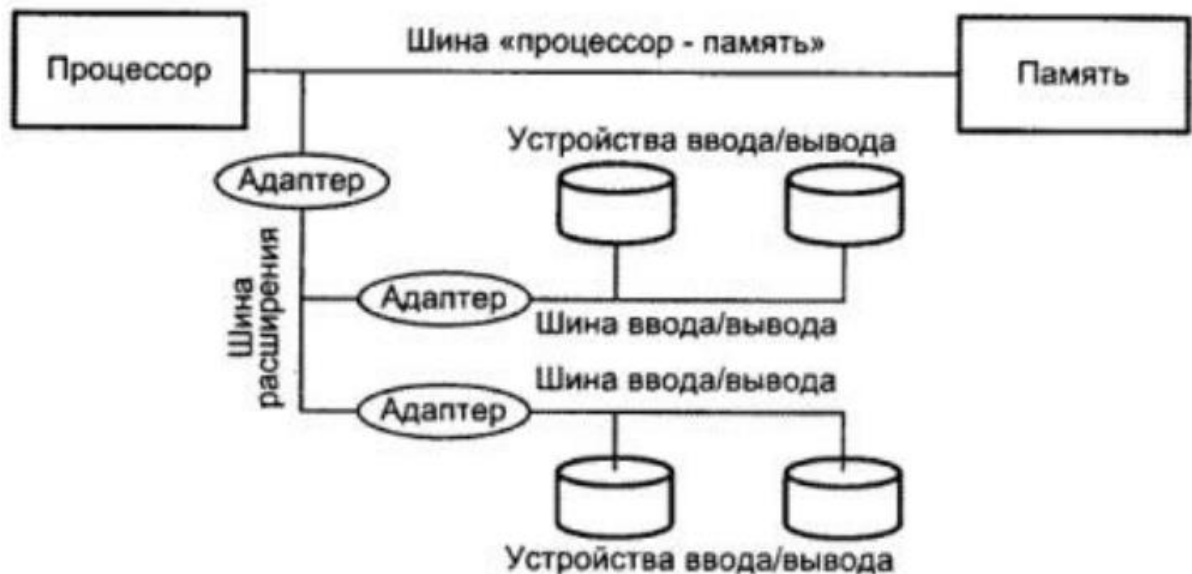
- трехшинные
- двухшинные (шина адреса + шина данных)
- одношинные

По способам передачи:

- параллельные (данные переносятся потактово словами: каждый бит — отдельным проводником)
- последовательные (биты данных переносятся поочередно по каналу, например, паре проводников).

5 Иерархия шин. Арбитраж шин.

Самая быстрая шина это шина «процессор – память». Основная функция адаптеров адаптация скоростей. Чтобы процессору передать данные устройству ввода/вывода, она передает их главному адаптеру, адаптер эти данные буферизирует и передает шине расширения. А шина расширения отдает данные адаптеру той линии на которой находится нужное устройство. Тот адаптер также буферизирует данные и передает их на той скорости устройству на которой он может эти данные принять.



Арбитраж шин

Арбитраж шин – предоставление шины только 1 устройству.

Арбитраж осуществляется по приоритету

Схемы смены приоритетов:

- 1) простая циклическая смена приоритетов;
- 2) циклическая смена приоритетов с учетом последнего запроса; (те устройства с которыми давно не было обмена, им даем максимальный приоритет)
- 3) смена приоритетов по случайному закону;
- 4) схема равных приоритетов (кто часто обменивается приоритет снижается, кто редко – увеличивается);
- 5) алгоритм наиболее давнего использования
- 6) алгоритм очереди (первым пришел — первым обслужен);
- 7) алгоритм фиксированного кванта времени

6 Характеристики памяти. Классификация запоминающих устройств.

Основные характеристики:

- 1) емкость (Бит, Байт)
- 2) разрядность
- 3) быстродействие
- 4) массо-габаритные показатели
- 5) энергозависимость
- 6) удельная стоимость (руб./байт)

Классификация механизмов доступа:

- 1) последовательный
- 2) прямой
- 3) произвольный
- 4) ассоциативный

Запоминающие устройства:

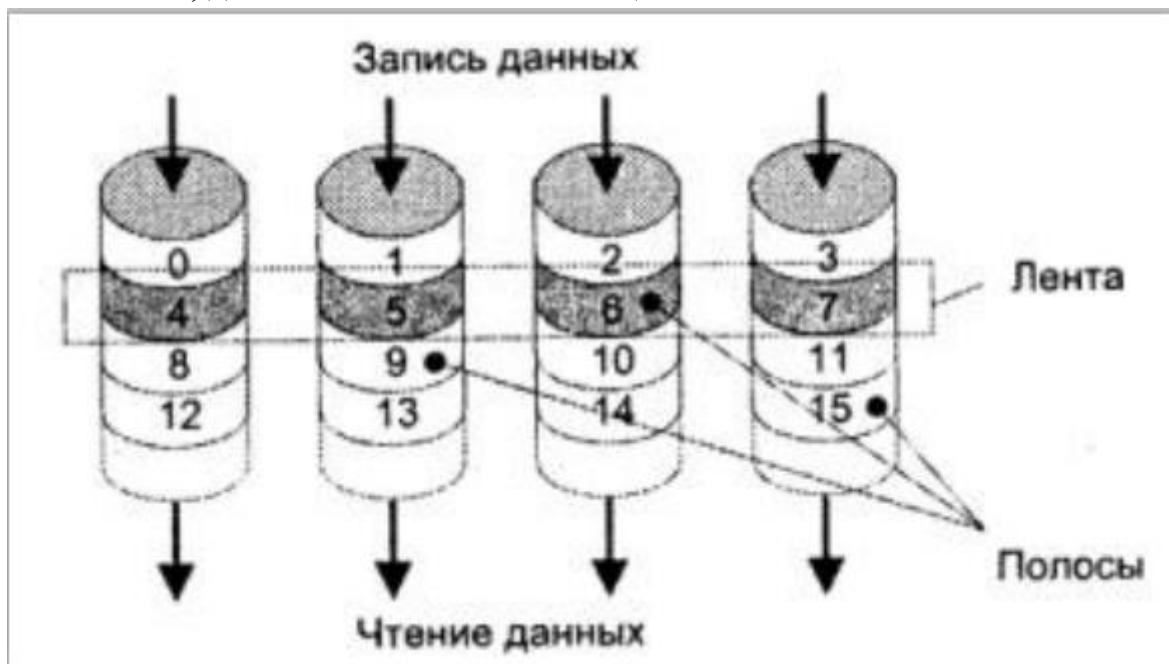
- 1) Оперативные:
 - DRAM (Dynamic Random Access Memory — динамическая память)
 - SRAM (Static Random Access Memory — статическая память)
- 2) Постоянные:
 - EPROM (Erasable Programmable ROM — стираемые программируемые ПЗУ)
 - EEPROM (Electrically Erasable Programmable ROM — электрически стираемые программируемые ПЗУ)
 - флэш-память

7 RAID-массивы.

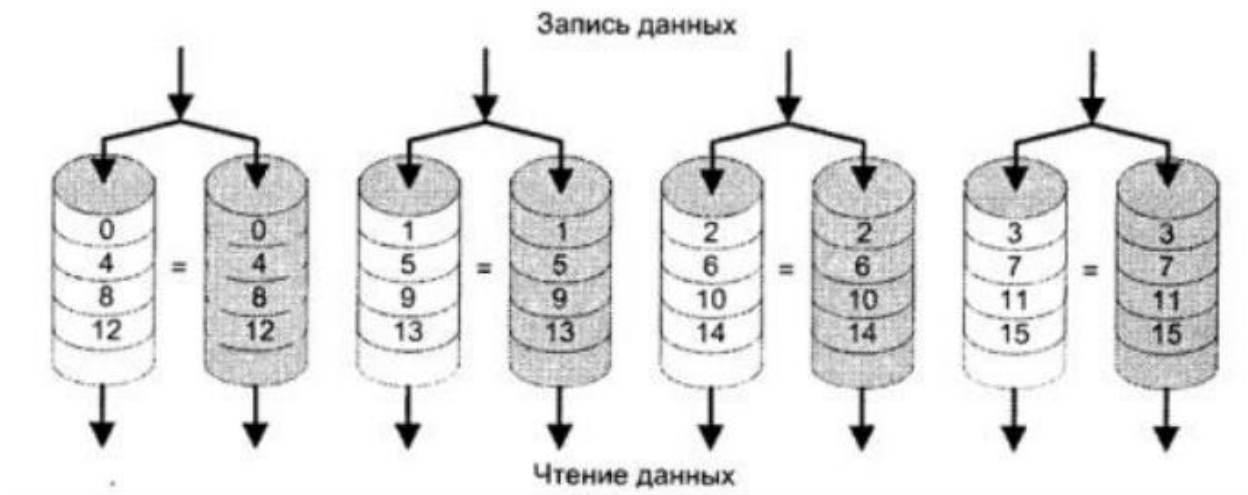
RAID представляет собой набор физических дисковых ЗУ, управляемых операционной системой и рассматриваемых как один логический диск.

Общепринятые типы: RAID 0, RAID 1, RAID 2, RAID 3, RAID 4, RAID 5, RAID 6.

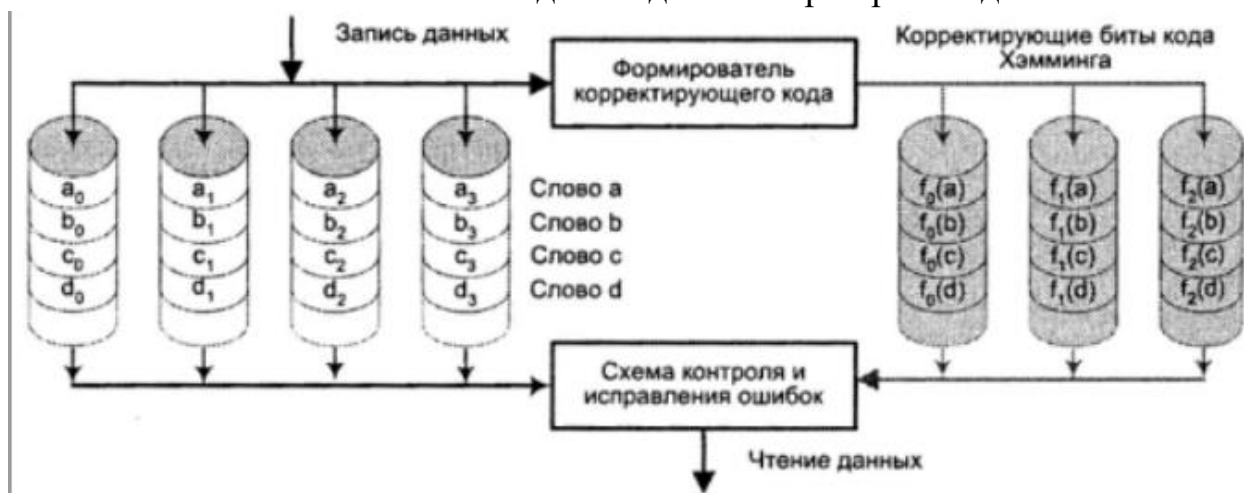
RAID 0: Расслоение данных, полосы распределены по всем дискам массива ЗУ, данные записываются по циклической схеме.



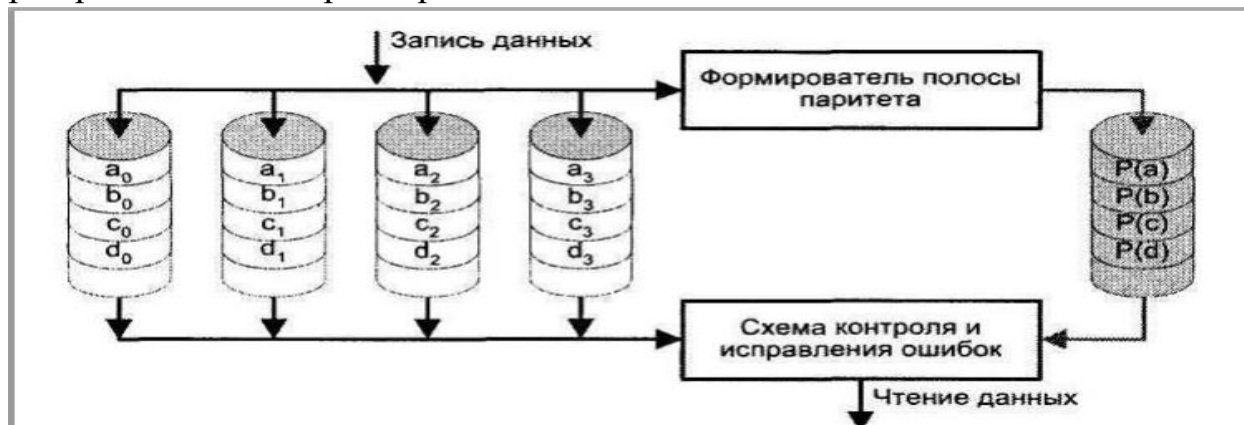
RAID 1: Запись одной информации на пару жестких дисков. Скорость считывания большая т.к можно считывать с 2 дисков одновременно. При потере данных на одном диске можно его восстановить при помощи второго диска.



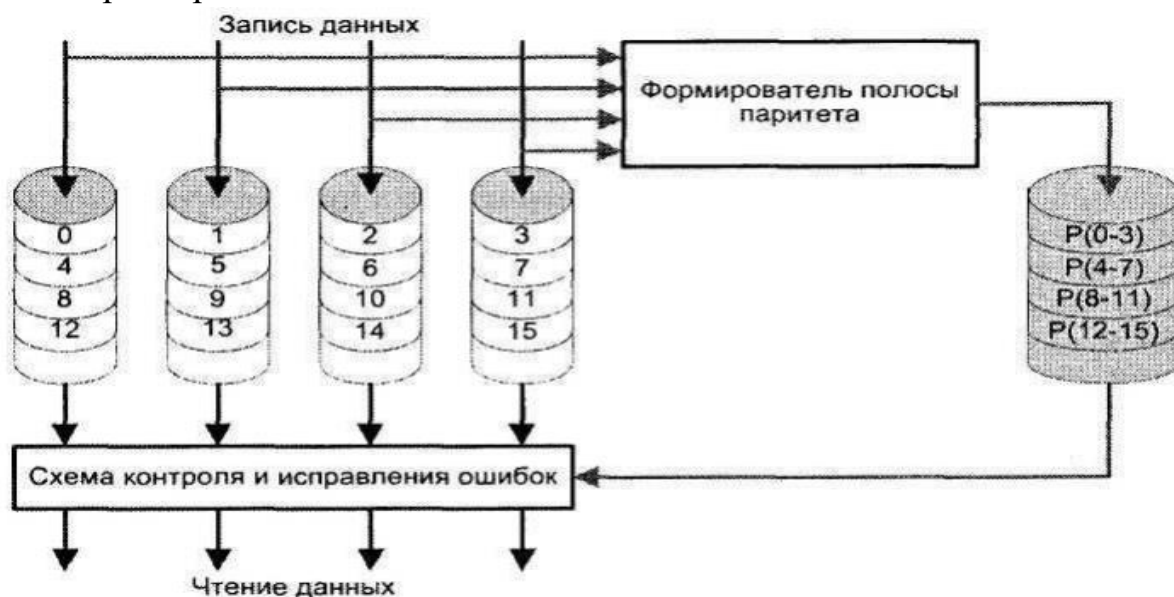
RAID 2: Высокая степень защиты информации с помощью кода Хэмминга можно восстановить один из дисков. 3 резервных диска.



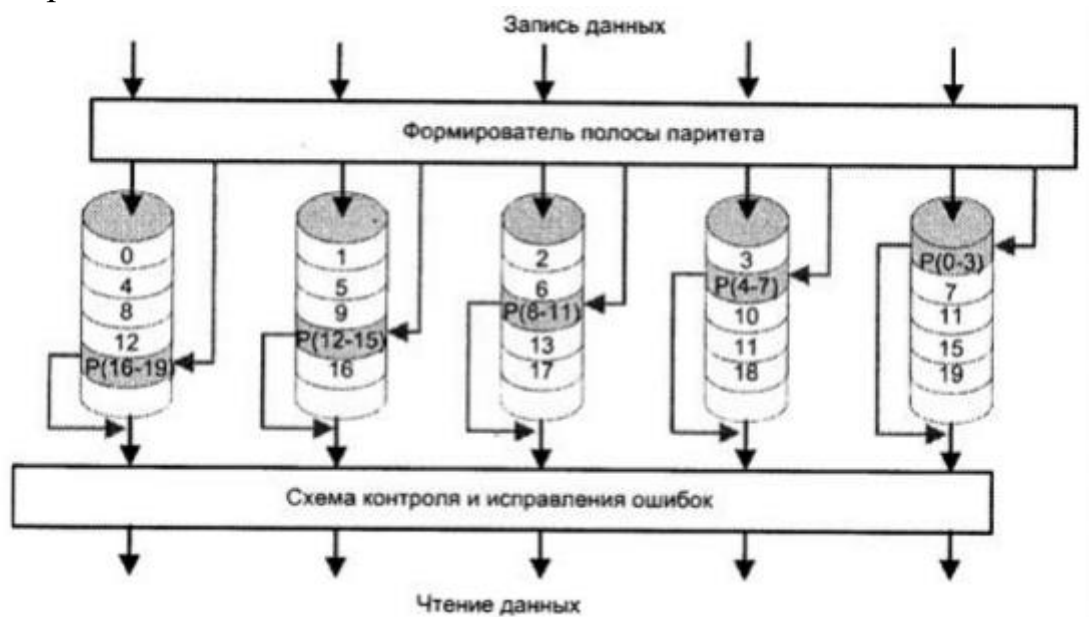
RAID 3: Запись на 4 диска. Формируется полосы паритета. Один резервный диск. Хорошо работает на большом объеме данных.



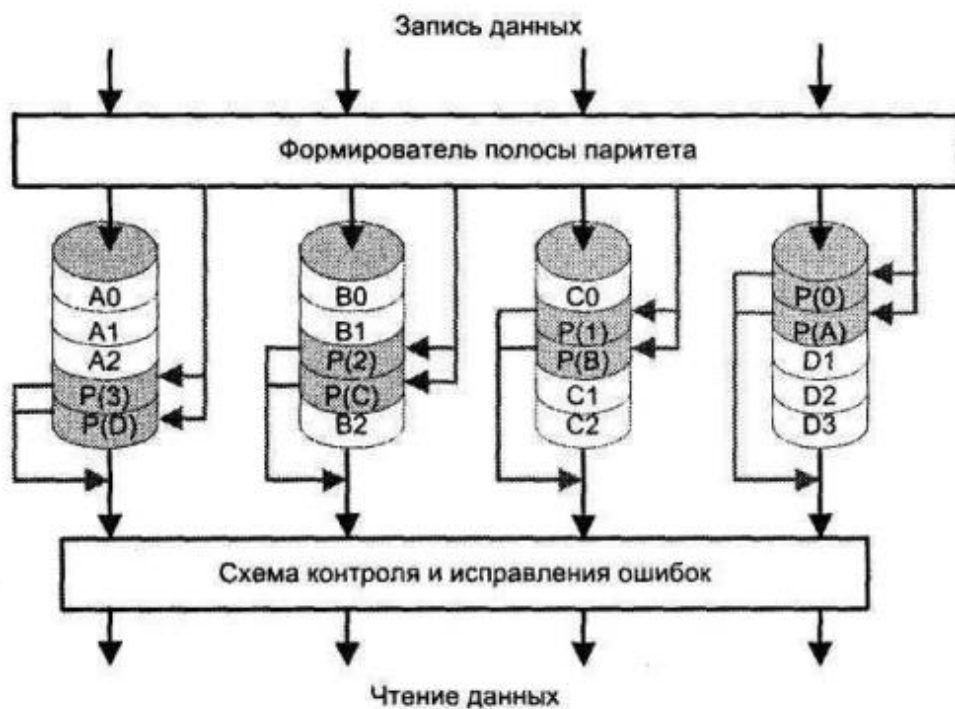
RAID 4: как RAID 3, только размер полос паритета больше. Используется техника независимого доступа. Хорошо работает на данных малого размера.



RAID 5: не содержит отдельного диска для хранения полос паритета, они разносятся по всем дискам.



RAID 6: как RAID 5, только на каждом диске хранится 2 полосы паритета (одна вертикальный срез, вторая горизонтальный). При отказе 2 дисков сможем восстановить оба.



8 Способы ввода-вывода информации

Четыре способа ввода/вывода:

- 1) Непосредственный – инициатором обмена является процессор, управление осуществляет процессор. Большой риск потери данных т.к. не проверяется состояние устройства.
- 2) По опросу готовности – инициатор обмена процессор, управление – процессор. Опрашивается состояние устройства. Возможны простои процессора при неготовности устройства.
- 3) Ввод/вывод по прерываниям – инициатором обмена внешнее устройство, управление – процессор. Обмен происходит только когда устройство готово (процессор ничего не ожидает – простоя нету). Сложная приоритизация. При подключении двух устройств следует использовать приоритеты.
- 4) Прямой доступ к памяти – самый быстрый. Обмен осуществляется без участия процессора. Инициатор – внешнее устройство, управление – контроллер прямого доступа к памяти. Процессор передает управление контроллеру. Сложность организации.

9 Машинная команда, структура.

Машинная команда - Код, определяющий операцию вычислительной машины, и данные, участвующие в операции.

Команда должна содержать в явной или неявной форме информацию об адресе результата операции, и об адресе следующей команды.

Машинная операция – это действия машины по преобразованию информации, выполняемые под воздействием одной команды.

Программа – последовательность команд, отображающих все действия, необходимые для решения задачи по некоторому алгоритму.

Машинный такт – период тактовой частоты работы процессора.

Машинный цикл – количество машинных тактов, требуемых для выполнения одной команды.

Основные группы команд:

- 1) Арифметические операции над числами с фиксированной или плавающей точкой.
- 2) Команды двоично-десятичной арифметики.
- 3) Логические (поразрядные) операции.
- 4) Пересылка операндов.
- 5) Операции ввода-вывода.
- 6) Передача управления.

7) Управление работой центрального процессора.

Структура машинной команды.



КОП – код операции.

В курсовой: в качестве приемника результата и одного из операндов используется аккумулятор (одноадресная).

Форматы машинных команд:

- 1) Фиксированный – все команды занимают одинаковое количество бит, как и каждое поле (код операции и адресная часть).
- 2) Плавающий – противоположность фиксированному.

10 Показатели производительности.

Под производительностью ЭВМ понимается потенциальная возможность по обработке информации (а не реальная).

В процессе обработки информации в ЭВМ реализуются те или иные операции из ее набора (или системы) операций.

Состав набора операций характеризует архитектуру ЭВМ и, следовательно, определяет ее производительность.

Показатели производительности ЭВМ:

- 1) Тактовая частота (возможно сравнение когда архитектура одинаковая)
- 2) Номинальное быстродействие $\vartheta_n = \frac{n}{\sum_{i=0}^n t_i}$, где $\sum_{i=0}^n t_i$ – среднее время выполнение одной команды.
- 3) Быстродействие по Гибсону $\vartheta_g = \frac{n}{\sum_{i=0}^n p_i t_i}$, где p_i – вероятность выполнения одной команды.
- 4) Оценка по независимым тестам.

Единицы измерения производительности:

Для оценки тактовой частоты: Гц, МГц, ГГц (Hz, MHz, GHz).

Для оценки номинального быстродействия и быстродействия по Гибсону, для чисел с фиксированной запятой: MIPS, MOPS.

По результатам тестов: 1 FLOPS (Floating-point Operations Per Second).

11 Процессор. Типы архитектур.

В общем случае процессор — это мозг компьютера. Он читает инструкции из памяти, которые указывают, что делать компьютеру.

Инструкции — это просто числа, которые интерпретируются специальным образом.

Существует фиксированное количество команд, которые понимает процессор. В мире представлено множество различных микропроцессоров, и они не используют одинаковый набор команд. Иными словами, они интерпретируют числа в инструкции по-разному.

Архитектура набора команд сильно влияет на архитектуру процессора.

Архитектура CISC

CISC – расширенный набор команд.

Отличительные особенности:

- большой набор команд
- использует переменные диапазоны форматов от 16-64 бит на инструкцию
- использует много способов адресации от 12 до 24
- архитектура использует 8-24 регистров общего назначения (мало)
- использует механизм памяти к памяти для выполнения операций
- использует унифицированный кэш для данных и инструкций
- имеет микрокодирование и использует управляющую память (ПЗУ)

Архитектура RISC

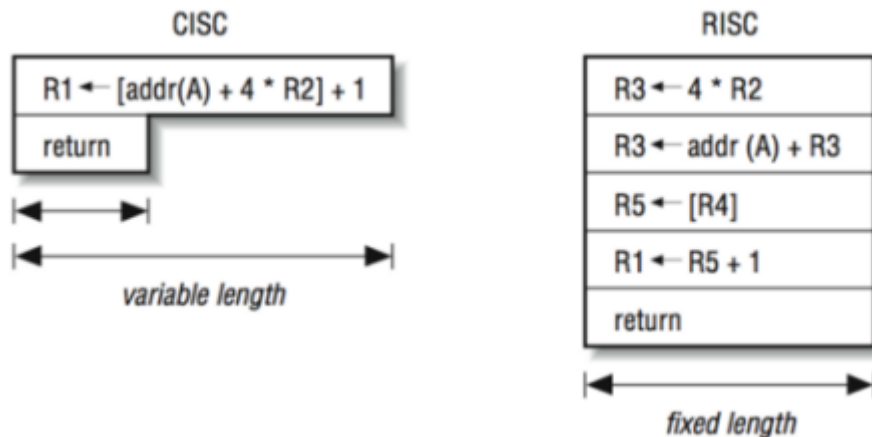
RISC – сокращенный набор команд.

Отличительные особенности:

- малый набор команд
- фиксированный формат машинной команды (32 бита)
- использует один такт на команду и режим ограниченной адресации (т.е. 3-5)
- число регистров общего назначения колеблется от 32 до 192 (большое)
- основные операции в процессоре выполняются над содержимым регистров
- имеет разделенный дизайн данных и кэш инструкций
- большая часть управления процессором является аппаратной

RISC – имеет большую производительность чем CISC. Но под него мало программного обеспечения.

Отличие форматов команд CISC и RISC:



У RISC фиксированная длина команд, насколько сложными они бы не были.

Гибридные архитектуры

Идея — обеспечить программную совместимость с CISC процессорами, но добиться эффективности RISC процессоров.

Микропроцессор имеет интерфейс CISC, а команды выполняет как RISC.

Поддерживается конвейеризация.

12 Способы адресации

Способ адресации — алгоритм получения физического адреса по командному.

В адресном поле команды содержится командный (исполнительный) адрес. Команда выбирается по физическому адресу.

Классификация способов адресации:

- 1) Явные — в адресном поле имеется командный адрес.
- 2) Неявные — отсутствует адрес.

Способы адресации. Явные.

- 1) Непосредственная адресация: операнд (байт или слово) размещается непосредственно в коде команды.
- 2) Прямая адресация: физический адрес совпадает с командным
- 3) Косвенная адресация: командный адрес содержит адрес ячейки памяти или регистра где содержится операнд.
- 4) Автоинкрементная: после выполнения операции с данными адрес операнда увеличивается на 1, если операнд имеет формат «байт», или на 2, если операнд имеет формат «слово».

- 5) Базовая адресация: физический адрес является суммой базового и командного адресов. (Для одномерного массива)
- 6) Базово-индексная адресация: физический адрес является суммой базового, индексного и командного адресов. (Для двумерного массива)
- 7) Укороченная адресация: адресный код содержит часть физического адреса, другая часть подразумевается.
- 8) Регистровая адресация: адресный код содержит номер регистра.
- 9) Стековая адресация: указывается вершина стека. (Популярна)

Способы адресации. Неявные.

- 1) Подразумевается операнд. ($i++$)
- 2) Подразумевается адрес операнда или приемника результата.

(В курсовой: адрес второго операнда подразумевается аккумулятор, также приемник результата является аккумулятором. Адрес следующей команды +1) (Также используется прямая адресация)

13 Устройство управления. Жесткое и микропрограммное управление.

В общем случае УУ формирует управляющие сигналы для выполнения следующих функций:

- 1) выборка из основной памяти (ОЗУ или ПЗУ) кода очередной команды;
- 2) расшифровка кода операции и признаков выбранной команды;
- 3) формирование исполнительного адреса операнда;
- 4) выборка операндов и выполнение машинной операции;
- 5) анализ запросов на прерывание исполняемой программы;
- 6) формирование адреса следующей команды

Принципы построения управляющих устройств.

- 1) Управляющие устройства с жесткой логикой (закон функционирования определяется способом соединения логических элементов).
Недостатки: невозможно внести правки в связи. Достоинства: работает очень быстро.
- 2) Микропрограммные устройства управления (закон функционирования задается программой, хранимой в ячейках памяти, называемых управляющей памятью). Достоинства: можно вносить правки в связи.

14 Периферийные устройства ввода ЭВМ.

• Клавиатуры (проводные и беспроводные) • Мыши (проводные и беспроводные) • Сканеры • Графические планшеты • Пенмаусы (аналог шариковой авторучки) • Световые перья • Цифровые видеокамеры и фотоаппараты, веб-камеры • Джойстики • Сенсорные панели (тачпады) • Средства речевого ввода

15 Периферийные устройства вывода ЭВМ.

Мониторы (ЭЛТ, ЖК, LED, OLED, qLED) • Сенсорные экраны (резистивные, емкостные) • Синтезаторы звука (спикер, бытовые, профессиональные) • Проекторы (LCD, DLP, LCoS, CRT) • Принтеры (ударные, струйные, электро и магнито-графические, электростатические, термические) • 3D принтеры (FDM, печать пластиком), SLA (фотополимерная печать), SLS (печать полиамидом), SLM (печать металлом) • Плоттеры (векторного, растрового типов, планшетные, барабанные, перьевые, струйные, электростатические, термические, карандашные)

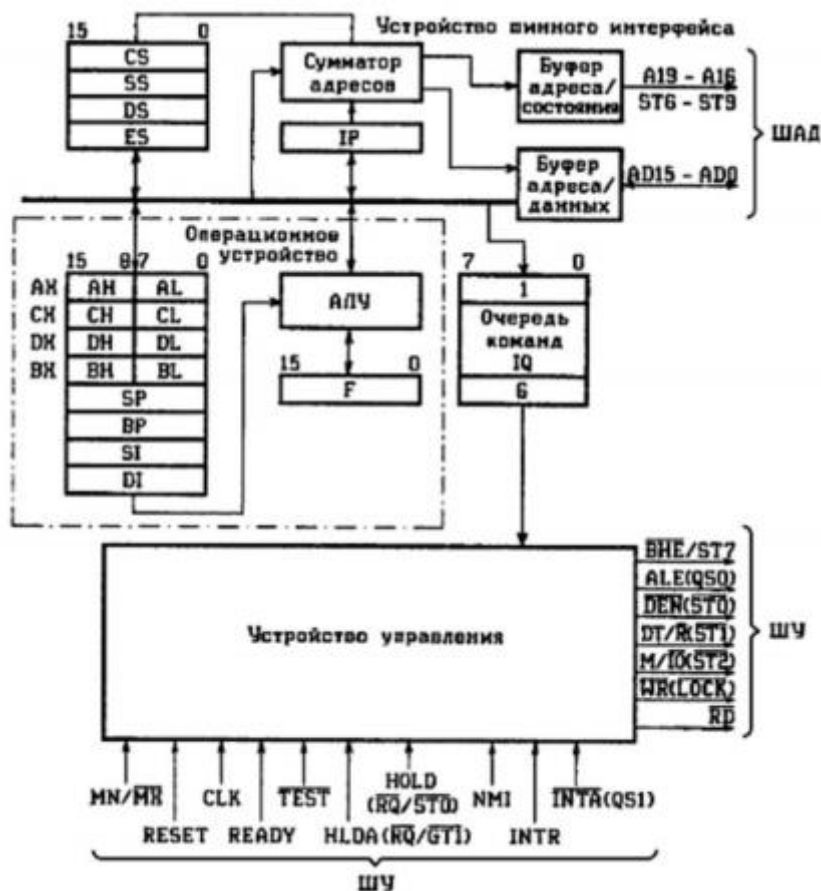
16 Внутренняя структура процессора i8086.

Особенности i8086:

- CISC архитектура
- 20 адресных линий
- может адресовать 2Мб оперативной памяти
- поддерживает прямой доступ к памяти
- обмен по запросу на прерывание
- внутренние регистры 16-ти разрядные
- поддерживает конвейер команд
- тактируется от внешнего генератора

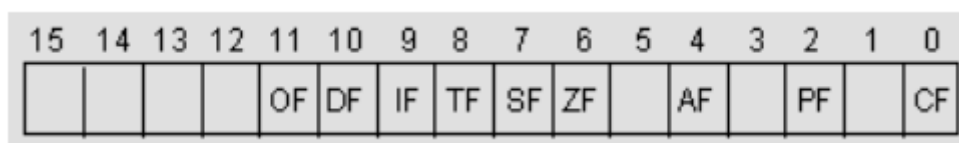
Внутренняя структура

Это был первый 16-битный процессор с 16-битным ALU, 16-битными регистрами, внутренней шиной данных и 16-битной внешней шиной данных, что привело к более быстрой обработке.



Регистр флагов.

Состоит из 16 бит, из них используются только 9.



CF — содержит перенос из старшего бита после арифметических операций и сдвигах

PF — показывает четность младших восьмибитовых данных (1 — четное, 0 — нечетное)

AF — признак межтетрадного переноса (из 3 бита)

ZF — флаг нулевого результата

SF — флаг знака (совпадает со старшим битом результата, 0 — плюс, 1 — минус)

TF — флаг пошагового режима (используется при отладке)

IF — флаг разрешения аппаратных прерываний

DF — флаг направления при строковых операциях. Обозначает левое или правое направление

OF — указывает на переполнение старшего бита

17 Назначение входов процессора i8086.

CLK — вход тактовых импульсов, задающих частоту процессора

NMI — немаскируемые запросы на прерывания

INTR — вход маскируемых запросов на прерывания

RESET — аппаратный сброс процессора

READY — готовность внешнего устройства или памяти к обмену данными

TEST — входной сигнал проверки. Используется в сочетании с сопроцессорами

MN/MX — задает режим работы процессора

HOLD — вход запроса на прямой доступ к памяти

+5 — питание (+5В)

GND — заземление (отрицательный контакт питания)

18 Назначение выходов процессора i8086.

AD0-AD15 — совмещенная шина адреса и данных

A16-A19 — старшие разряды шины адреса

BHE — для разрешения подключения старшего банка памяти (16 или 8)

RD — чтение из памяти или внешнего устройства

WR — запись в порт внешнего устройства или в память

DT/R — 1 - передача данных, 0 - прием

INTA — для подтверждения обработки запроса на прерывание

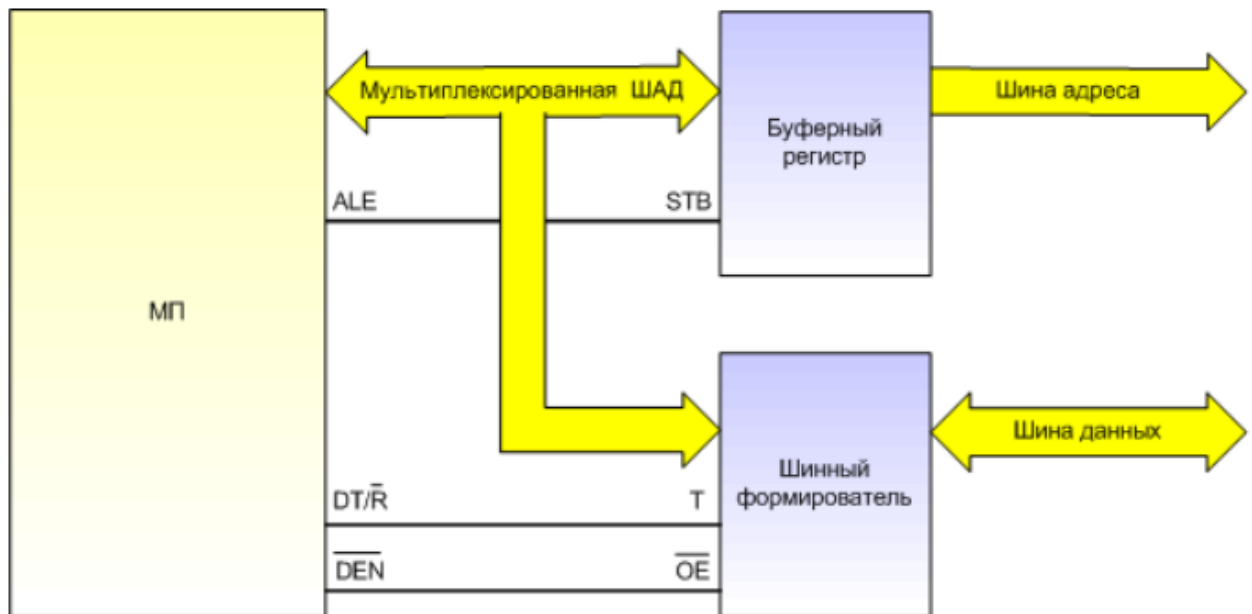
HLDA — выход подтверждения запроса на прямой доступ к памяти

DEN — по совмещенной шине передаются данные

ALE — при выставлении адреса на шину формируется синхронизирующий импульс

M/IO — разделение адресных пространств памяти и внешних устройств

19 Схема демultipликации шин адреса и данных



От процессора у нас подается мультиплексированная шина адреса и данных и передается на 2 устройства. Первое устройство это буферный регистр, второе – шинный формирователь. Буферный регистр это ячейка памяти, способная что-то хранить. При появлении единичного сигнала на входе запоминает что на входе, удерживает и выставляет на своем выходе, а именно на шину адреса. Шинный формирователь – его задача обеспечить определенное согласование. При наличии на входе den 1 шинный формирователь отключается и все что находится в ШАД дальше не выходит. Когда на входе den 0, то шинный формирователь подключается к внешним шинам и обеспечивает передачу в обоих направлениях.

20 Организация памяти i8086. 21 Сегментация памяти i8086.

ЭВМ выполняет вычисления в соответствии с программой, которая располагается в памяти ЭВМ.

Любая программа включает в себя команды (операторы) и данные (операнды). Программа выполняется с целью получения результирующих данных на основе преобразования исходных, с возможным формированием промежуточных данных.

Память представляет собой набор ячеек, каждая из которых имеет свой уникальный номер – адрес.

Поскольку команды и данные на уровне кодов неотличимы друг от друга, то для различия команд и данных используется их размещение в различных областях памяти – сегментах. (В курсовой: для отличия используется специальный бит)

Сегментация

Сегмент - это прямоугольная область памяти, характеризующаяся начальным адресом и длиной. Начальный адрес (адрес начала сегмента) – это номер (адрес) ячейки памяти, с которой начинается сегмент.

Длина сегмента – это количество входящих в него ячеек памяти. Сегменты могут иметь различную длину. Все ячейки, расположенные внутри сегмента, перенумеровываются, начиная с нуля.

Адресация ячеек внутри сегмента ведется относительно начала сегмента; адрес ячейки в сегменте называется смещением или эффективным адресом - ЕА (относительно начального адреса сегмента).

Переполнения при формировании адреса

Перенос из старшего бита, который может возникнуть при суммировании, игнорируется. Это приводит к так называемой кольцевой организации памяти, при которой за ячейкой с максимальным адресом FFFFF следует ячейка с нулевым адресом.

Аналогичную кольцевую организацию имеет и каждый сегмент.

Сегментные регистры

16-битная x86-архитектура, благодаря наличию четырёх сегментных регистров, позволяет одновременно иметь доступ к четырём сегментам памяти. Назначение сегментных регистров:

DS (data segment) — сегмент данных;

CS (code segment) — сегмент кода;

SS (stack segment) — сегмент стека;

ES (extra segment) — дополнительный сегмент.

Логический адрес на такой платформе принято записывать в виде сегмент:смещение, где сегмент и смещение задаются в шестнадцатеричной системе счисления.

В реальном режиме для вычисления физического адреса байта памяти происходит сдвиг влево на 4 разряда значения соответствующего сегментного регистра, а затем добавляется смещение.

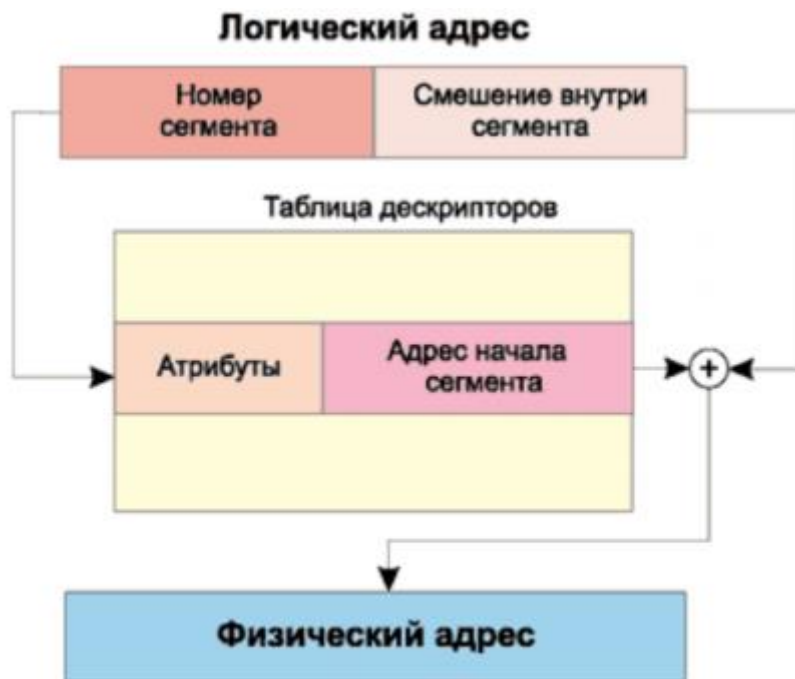
Особенности сегментации памяти в микропроцессоре i8086

- Сегменты памяти определяются только сегментными регистрами.
- Начальный адрес сегмента кратен 16
- Никаких средств проверки правильности использования сегментов нет.
- Размещение сегментов в памяти достаточно произвольно.
- Сегменты могут частично или полностью перекрываться, или не иметь общих частей.
- Программа может обращаться к любому сегменту как для считывания, так и для записи данных и команд.
- Для защиты памяти от несанкционированного доступа других программ требуются специальные "внешние" схемы.
- Система не делает различий между сегментами данных, кода и стека.
- Нет никаких препятствий для обращения к физически не существующей памяти.
- При обращении к несуществующей памяти результат непредсказуем (все зависит от разработчика аппаратного обеспечения компьютера.)

Защищенный режим.

Появился в процессоре 80286, сохранив понятия сегмента и смещения, и радикально модернизировал механизм сегментации, предложив концепцию логического (виртуального) адреса.

В защищенном режиме работы процессора содержимое сегментных регистров используется не как слагаемое при вычислении адреса, а как индекс или селектор, выбирающий дескриптор сегмента в специальной структуре, называемой дескрипторной таблицей и описывающей свойства каждого сегмента: базовый адрес, размер и атрибуты, связанные с управлением привилегиями и защитой памяти.



22 Организация прерываний на примере i8086.

Таблица векторов прерываний.

В реальном режиме работы в системе прерываний используется понятие вектора прерывания.

Каждый вектор прерывания состоит из четырех байт, или двух слов: первые два содержат новое значение для регистра IP, а следующие два – новое значение для регистра CS. Таблица векторов прерывания занимает 1024 байт. Таким образом, в ней может быть задано 256 векторов прерываний.

Виды прерываний.

- 1) Внутренние прерывания возникают в результате работы процессора в ситуациях, которые нуждаются в специальном обслуживании, или при выполнении специальных команд. Это следующие прерывания: а) прерывание при делении на 0 б) прерывание по флагу TF в) прерывание возникающие при выполнении команд INT называются программными. В качестве операнда команды INT указывается номер прерывания, которое нужно выполнить.
- 2) Внешние прерывания возникают по сигналу какого-нибудь внешнего устройства. Существует два специальных внешних сигнала среды входных сигналов процессор, при помощи которых можно прервать выполнение текущей программы и тем самым переключить работу ЦП.

Немаскируемые прерывания будут всегда обрабатываться процессором. Маскируемые прерывания будут обрабатываться только если у нас стоит флаг.

Этапы обработки прерываний.

- 1) Контроллер прерываний получает заявку от определенного периферийного устройства и, соблюдая схему приоритетов, генерирует запрос на прерывание, который является входным для микропроцессора.
- 2) Микропроцессор проверяет флаг IF в регистре флагов. Если он установлен в 1, то переходим к шагу 3. В противном случае работа процессора не прерывается.
- 3) Микропроцессор генерирует подтверждение прерывания. В ответ на этот сигнал контроллер прерываний посылает по шине данных номер прерывания.
- 4) В стек помещается регистр флагов.
- 5) Флаг включения-выключения прерываний IF и флаг трассировки TF, находящиеся в регистре флагов, обнуляются для блокировки других маскируемых прерываний и исключения пошагового режима исполнения команд.
- 6) Значение регистров CS и IP сохраняются в стеке.
- 7) Вычисляется адрес вектора прерывания и из вектора, соответствующего номеру прерывания, загружаются новые значения IP и CS.
- 8) Выполнение подпрограммы обработчика прерывания.
- 9) Извлечение из стека IP и CS.
- 10) Извлечение из стека регистра флагов.
- 11) Процессор продолжает работу с того момента, где он был прерван.

Использование прерываний.

- 1) Для обмена информацией между процессором и внешним устройством
- 2) В аварийных ситуациях, например, при понижении напряжения питания
- 3) При исключительных условиях, таких, как переполнение
- 4) При программных сбоях
- 5) Для профилактики, ремонта, тестирования и отладки системы
- 6) Для координации работы в многопроцессорных системах