1. Введение. Микропроцессорная система и ее программирование

1.1. Состав микропроцессорной системы

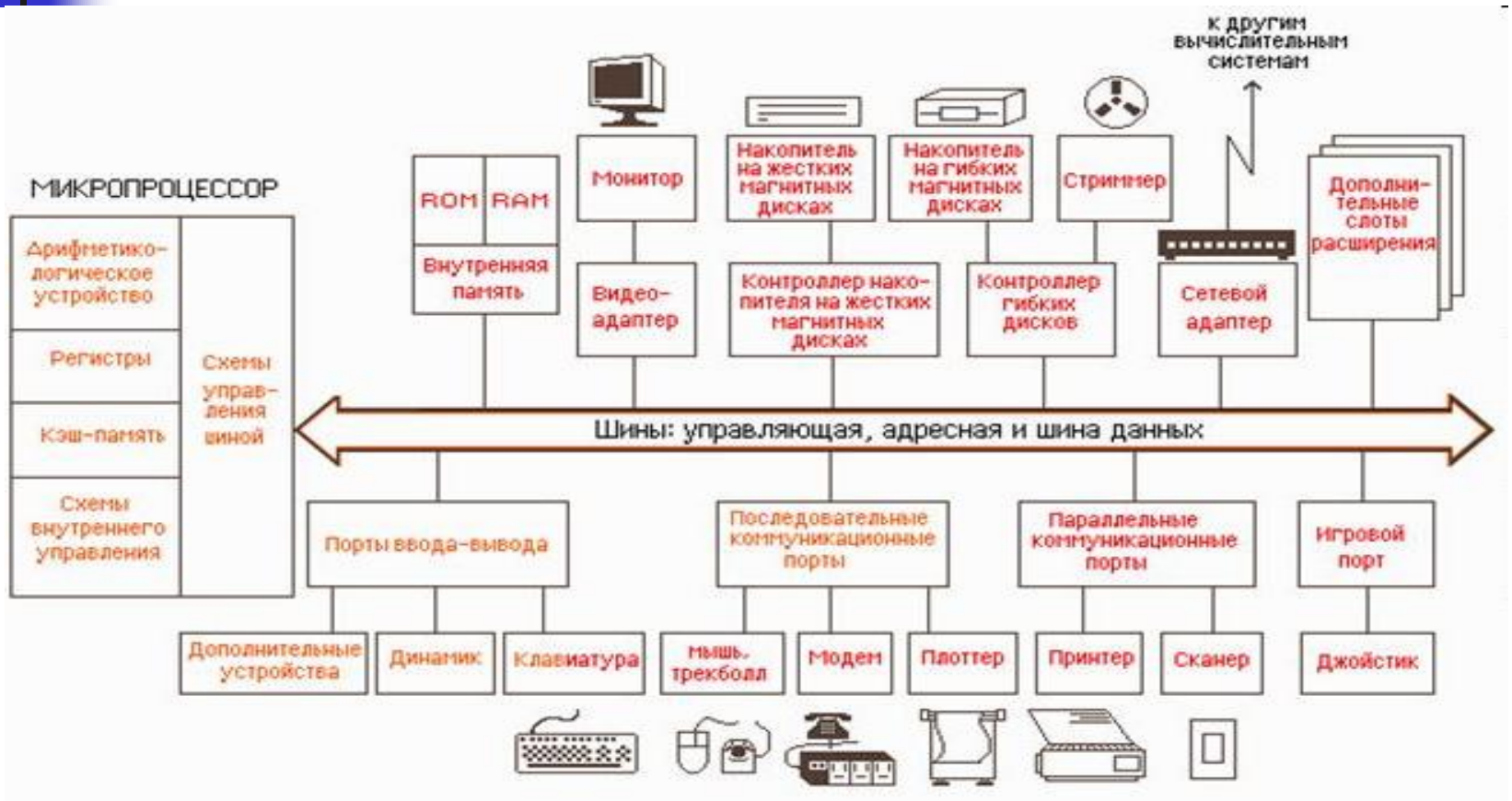
Состав микропроцессорной системы

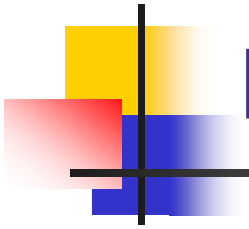
Микропроцессорная система включает:

- микропроцессор;
- память;
- устройства ввода-вывода.

Эти компоненты связаны посредством внутренней шины (семейства внутренних шин).

Состав системы (ПК)





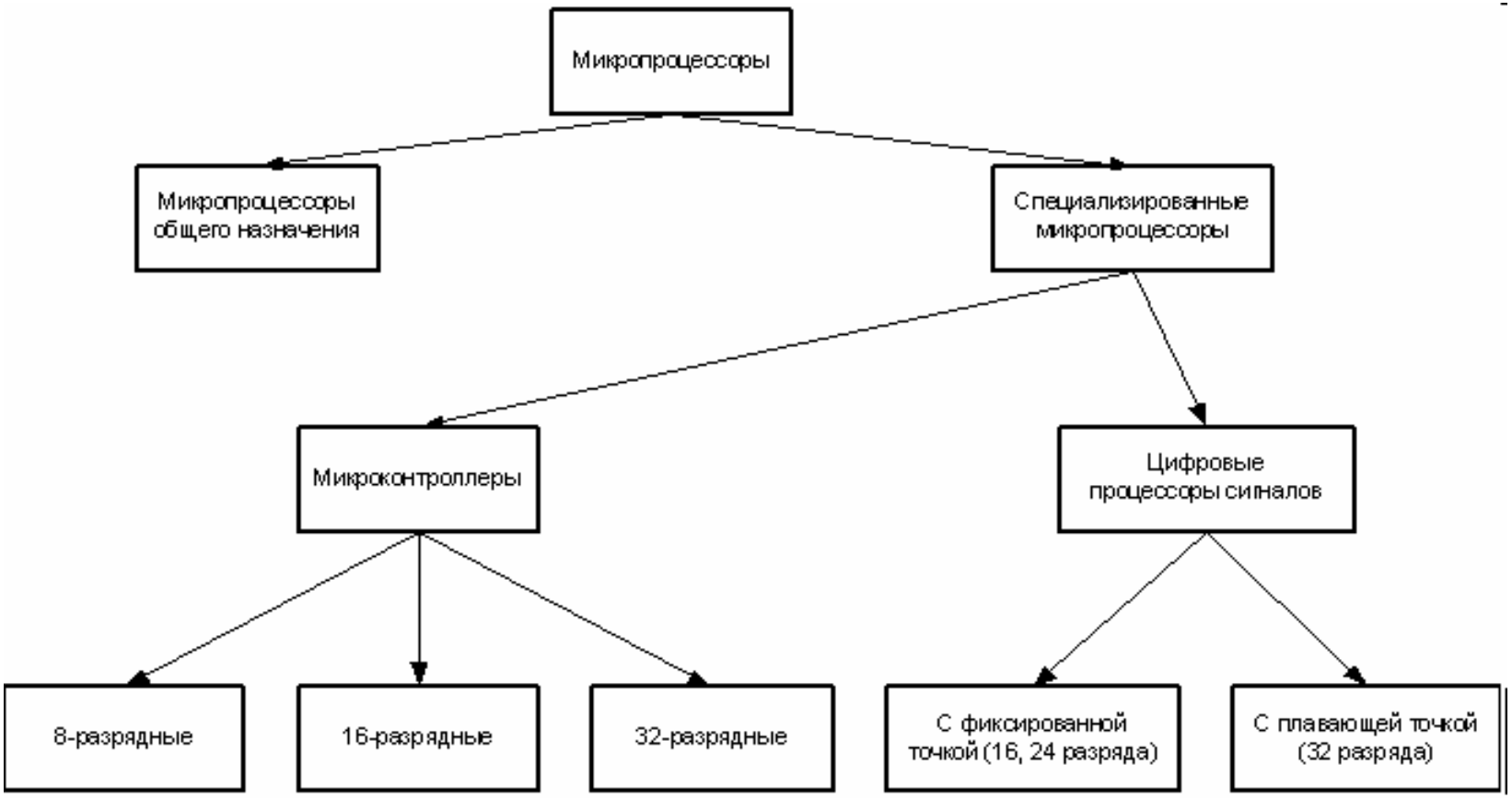
Микропроцессор

Микропроцессор – программно-управляемое электронное устройство для цифровой обработки информации, размещенное на одном или нескольких кристаллах.

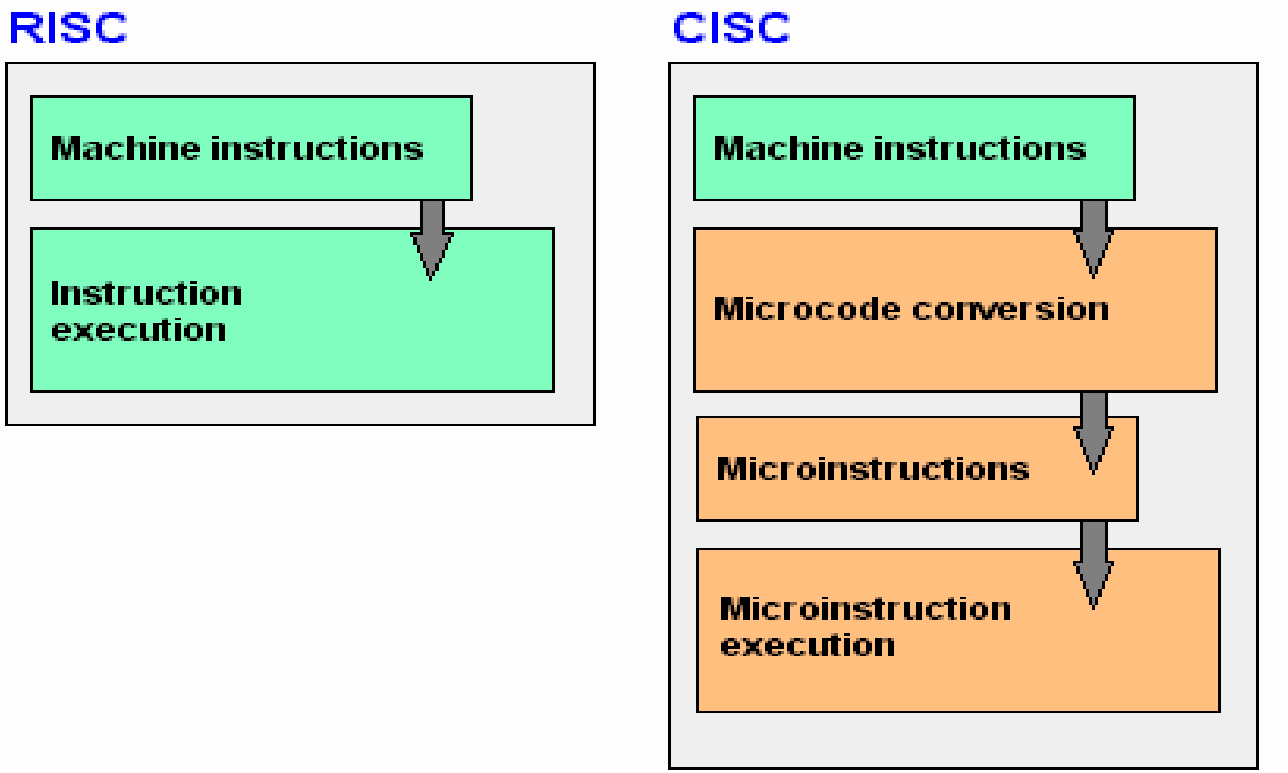
Если микропроцессор располагается на одном кристалле, он называется однокристальным или микропроцессором с фиксированной разрядностью (8, 16, 32, 64 разрядные и др.)

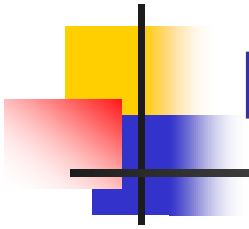
Существуют также микропроцессоры с переменной разрядностью (секционные) они состоят из нескольких секций небольшой разрядности, размещенных на разных кристаллах. Каждая секция функционально закончена, т.е. может выполнять все команды. Достоинство секционных процессоров – возможность наращивать разрядность исходя из требований применения.

Классификация микропроцессоров



RISC и SISC - процессоры

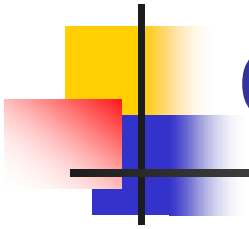




RISC - процессоры

RISC (англ. Restricted (reduced) instruction set computer - компьютер с сокращённым набором команд) архитектура процессора, в которой быстродействие увеличивается за счёт упрощения инструкций, чтобы их декодирование было более простым, а время выполнения - короче.

Цель - сделать инструкции настолько простыми, чтобы они легко конвейеризировались и тратили не более одного такта на каждом шаге конвейера на высоких частотах



CISC - процессоры

CISC (англ. complex instruction set computing, или англ. complex instruction set computer - компьютер с полным набором команд) - концепция проектирования процессоров, которая характеризуется следующим набором свойств:

- нефиксированное значение длины команды;
- арифметические действия кодируются в одной команде;
- небольшое число регистров, каждый из которых выполняет строго определённую функцию.

Микропроцессоры семейства Intel 86

1978 г. Intel® 8086:

- Частота 10 МГц.
- На его основе начали выпускать компьютеры IBM PC.
- Тех. характеристики: 29000 транзисторов; технология производства: 3 мкм; напряжение питания: 5 В; тактовая частота: 4,77-10 МГц; процессор 16-разрядный; шина данных 16-разрядная; адресная шина 20-разрядная; общая разрядность: 16.

Микропроцессоры семейства Intel 86

1982г. Intel® 80286:

- Впервые – аппаратная поддержка многозадачности
- Тех. характеристики: 134000 транзисторов; тактовая частота: 6-12 МГц; процессор 16-разрядный; шина данных 16-разрядная; адресная шина 24-разрядная; общая разрядность: 16.

Микропроцессоры семейства Intel 86

1985 г. Intel® 386™ DX:

- Первый 32-разрядный процессор
- Тех. характеристики: 275000 транзисторов; тактовая частота: 16-32 МГц; процессор 32-разрядный; шина данных 32-разрядная (16-32Мгц); адресная шина 32-разрядная; общая разрядность: 32.

Микропроцессоры семейства Intel 86

1989 г. Intel® 486™ DX:

- Первый процессор со встроенными кэшем первого уровня и математическим сопроцессором (FPU), который существенно ускорил обработку данных.
- Тех. характеристики: 1,25 млн. транзисторов; тактовая частота: 25-50 МГц; кэш первого уровня: 8 Кб; кэш второго уровня на материнской плате (до 512 Кб); процессор 32-разрядный; шина данных 32-разрядная (20-50Мгц);

  адресная шина 32-разрядная; общая разрядность: 32.

Микропроцессоры семейства Intel 86

1993г. Intel® Pentium® (P5):

- Первый процессор с двухконвейерной структурой.
- Кэш-память впервые была разделена – 8 Кб на данные и 8 Кб на инструкции.
- Тех. характеристики: 3,1 млн. транзисторов; технология производства: 0,8 мкм; тактовая частота: 60-66 МГц; кэш первого уровня: 16 Кб (8 Кб на данные и 8 Кб на инструкции); кэш второго уровня на материнской плате (до 1 Мб); процессор 64-разрядный; шина данных 64-разрядная (60-66 МГц); адресная шина 32-разрядная; общая разрядность: 32

Микропроцессоры семейства Intel 86

1997г. Intel® Pentium® MMX (P55C):

- Расширение MMX (Multi Media eXtention), содержащее 57 инструкций для вычислений с плавающей точкой, существенно увеличивающее производительность компьютера в мультимедиа-приложениях
- Тех. характеристики: 4,5 млн. транзисторов; технология производства: 0,28 мкм; тактовая частота: 166-233 МГц; кэш первого уровня: 32 Кб (16 Кб на данные и 16 Кб на инструкции); кэш второго уровня на материнской плате (до 1 Мб); процессор 64-разрядный; шина данных 64-разрядная (60-66 МГц); адресная шина 32-разрядная; общая разрядность: 32

Микропроцессоры семейства Intel 86

2000г. Intel® Pentium® 4

Принципиально новый процессор с гиперконвейеризацией (hyperpipelining) - с конвейером, состоящим из 20 ступеней. Применена 400 МГц системная шина (Quad-pumped), обеспечивающая пропускную способность в 3,2 ГБайта в секунду против 133 МГц шины с пропускной способностью 1,06 ГБайт у Pentium III.

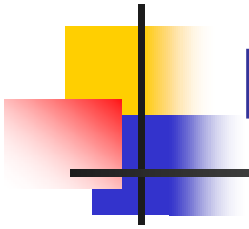
Тех. характеристики: технология производства: 0,18 мкм; тактовая частота: 1.3-2 ГГц; кэш первого уровня: 8 Кб; кэш второго уровня 256 Кб (полноскоростной); процессор 64-разрядный; шина данных 64-разрядная (400 МГц)

Микропроцессоры семейства Intel 86

2005 г. Intel ® Pentium Extreme Edition 840, Itanium 2 (Montecito)

Первые двухъядерные процессоры Оба ядра поддерживают технологию Hyper Threading, так что процессор распознается операционной системой как 4-процессорный.

Тех. характеристики (Itanium 2): 64-битный процессор содержит 1,7 миллиарда транзисторов, сделаны по технологии с разрешением 90 нм, имеет кэш-память емкостью 24 Мбайт и потребляет около 100 Вт от источника питания.



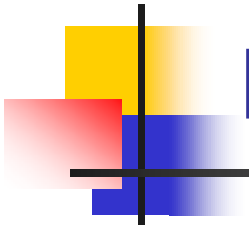
Память

Под памятью будем понимать совокупность ячеек для хранения информации, непосредственно адресуемых микропроцессором (т.е. входящую в *адресное пространство памяти*). Такая память в основном реализуется в виде электронных модулей (микросхем).

В памяти хранятся:

- команды;
- данные.

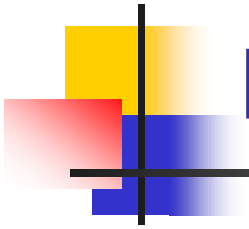
В некоторых системах адресные пространства команд и памяти разделены. В ПК команды и данные находятся в одном адресном пространстве.



Память

Виды памяти:

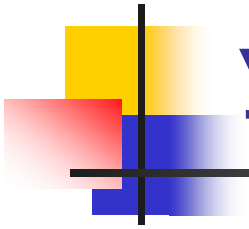
- энергонезависимая память (англ. nonvolatile storage) – память, реализованная ЗУ, записи в которых не стираются при снятии электропитания. К этому типу памяти относятся все виды памяти на ПЗУ и ППЗУ (BIOS);
- энергозависимая память (англ. volatile storage) – память, реализованная ЗУ, записи в которых стираются при снятии электропитания. К этому типу памяти относятся память, реализованная на ОЗУ, кэш-память.



Память

Виды энергозависимой памяти:

- статическая память (англ. static storage) — энергозависимая память, которой для хранения информации достаточно сохранения питающего напряжения (память параметров SMOS BIOS, питающаяся от аккамулятора);
- динамическая память (англ. dynamic storage) — энергозависимая память, в которой информация со временем разрушается (деградирует), и, кроме подачи электропитания, необходимо производить её периодическое восстановление (регенерацию) (основная память).

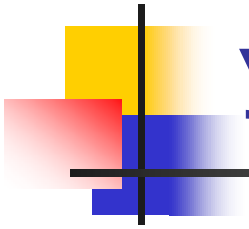


Устройства ввода-вывода

Под устройствами ввода-вывода в широком смысле понимают все периферийные устройства компьютера (микропроцессорной системы).

Для микропроцессора устройством ввода-вывода является любое устройство, входящее в *адресное пространство ввода-вывода* и адресуемое посредством номера порта.

Все периферийные устройства имеют в своем составе т.н. *контроллеры* – специализированные микросхемы или даже микропроцессорные системы, за которыми закреплены определенные ресурсы (номера портов, номера прерываний и т.д.). Через эти ресурсы они и доступны «главному» микропроцессору.

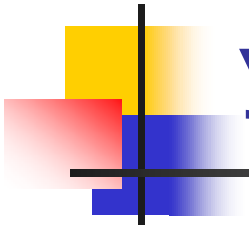


Устройства ввода-вывода

Способы управления вводом-выводом

Существуют два основных механизма обмена данными с внешними устройствами:

- программно-управляемый обмен (ввод-вывод).
- прямой доступ к памяти.



Устройства ввода-вывода

Программно-управляемый ввод-вывод означает обмен данными с внешними устройствами с использованием команд процессора. Передача данных происходит через регистры процессора и при этом в конечном счете может реализовываться обмен собственно с процессором, обмен внешнего устройства с памятью, обмен между внешними устройствами.

Такой обмен может быть:

синхронным (инициализируется процессором);

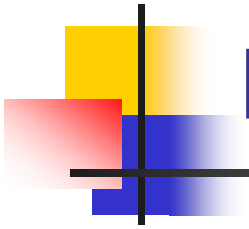
асинхронным (инициализируется устройством посредством прерывания).

Устройства ввода-вывода

Прямой доступ к памяти Direct Memory Access, DMA)— режим обмена данными между устройствами ввода/вывода или же между устройством и основной памятью (RAM), без участия Центрального Процессора (ЦП). В результате скорость передачи увеличивается, так как данные не пересылаются в ЦП и обратно.

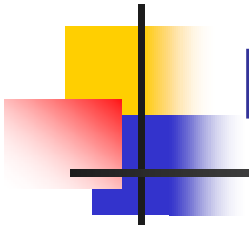
Кроме того, данные пересылаются сразу для многих слов, расположенных по подряд идущим адресам, что позволяет использование т. н. «взрывного» (burst) режима работы шины –1 цикл адреса и следующие за ним многочисленные циклы данных.

Перед началом обмена процессор инициирует контроллер DMA, записывая в его регистры информацию о обмене.



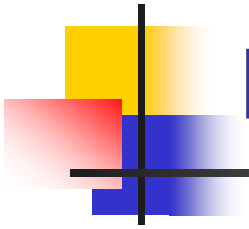
Шина

Компьютерная ши́на (от англ. computer bus, bidirectional universal switch – двунаправленный универсальный коммутатор) – в архитектуре компьютера подсистема, которая передаёт данные между функциональными блоками компьютера. Обычно шина управляется драйвером. В отличие от связи точка-точка, к шине можно подключить несколько устройств по одному набору проводников. Каждая шина определяет свой набор коннекторов (соединений) для физического подключения устройств, карт и кабелей.



Шина

Ранние компьютерные шины представляли собой параллельные электрические шины с несколькими подключениями, но сейчас данный термин используется для любых физических механизмов, предоставляющих такую же логическую функциональность, как параллельные компьютерные шины. Современные компьютерные шины используют как параллельные, так и последовательные соединения и могут иметь параллельные (multidrop) и цепные (daisy chain) топологии. В случае USB и некоторых других шин могут также использоваться хабы (концентраторы).

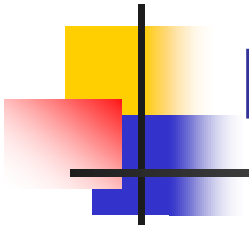


Шина

Функционально шину можно разделить на:

- шину адреса, по которой передаются адреса;
- шину данных для передачи данных;
- шину управления, содержащую специальные управляющие сигнальные линии.

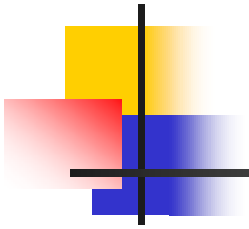
Однако в современных системах эти шины могут совместно использовать одни и те же физические линии.



Примеры шин

ISA (от англ. Industry Standard Architecture, ISA bus, произносится как ай-эс-эй) — 8- или 16-разрядная шина ввода/вывода IBM PC-совместимых компьютеров. Служит для подключения плат расширения стандарта ISA. Конструктивно выполняется в виде 62-х или 98-контактного разъёма на материнской плате.

Впервые шина ISA появилась на компьютерах IBM PC/XT в 1981 году. Это была 8-разрядная шина с частотой до 8 МГц и скоростью передачи данных до 4 МБайт/с (передача каждого байта требовала минимум двух тактов шины). Разъём состоял из 62 контактов, из которых 8 использовалось для данных, 20 – для адреса, остальные – для управляющих сигналов, а также подачи напряжений питания (GND, +5 В, -5 В, +12 В и -12 В).



ISA

В 1984 году шина была усовершенствована. Была удвоена разрядность данных (что повлекло удвоение пропускной способности) и добавлены четыре разряда адреса; кроме того, увеличилось число линий запросов прерываний (IRQ) и запросов прямого доступа к памяти (DMA). Кроме того, в 16-разрядной шине ISA любое подключенное к ней устройство могло выступать в роли задатчика, то есть инициировать операцию обмена данными (в 8-разрядной шине задатчиками были только процессор и контроллер DMA). Для подключения 16-разрядных устройств используются разъёмы, состоящие из двух частей: полностью совместимой с 8-разрядной шиной 62-контактной и новой 36-контактной.

ISA

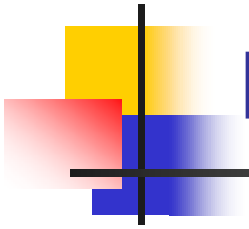
Примеры шин

PCI (англ. Peripheral component interconnect, дословно – взаимосвязь периферийных компонентов) – шина ввода/вывода для подключения периферийных устройств к материнской плате компьютера.

Разработана в 1991 года компанией Intel для процессоров 486, Pentium и Pentium Pro.

В 1997 году, в связи с развитием компьютерной графики и разработкой шины AGP, шина PCI перестала удовлетворять новым, повышенным требованием к видеокартам и перестала использоваться для установки видеокарт.

В настоящее время интерфейс PCI постепенно вытесняется интерфейсами PCI Express, HyperTransport и USB.

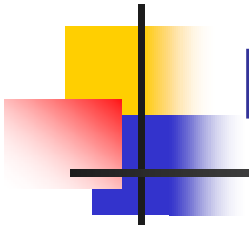


PCI

Спецификация:

- частота шины – 33,33 или 66,66 МГц, передача синхронная;
- разрядность шины – 32 или 64 бита, шина мультиплексированная (адрес и данные передаются по одним и тем же линиям);
- пиковая пропускная способность для 32-разрядного варианта, работающего на частоте 33,33 МГц — 133 Мбайт/с;
- адресное пространство памяти — 32 бита (4 байта);
- адресное пространство портов ввода-вывода — 32 бита (4 байта);
- конфигурационное адресное пространство (для одной функции) 256 байт;
- напряжение 3,3 или 5 В.

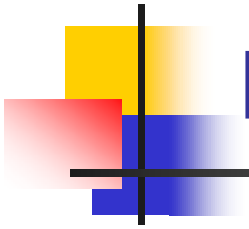
PCI



Примеры шин

PCI Express, или PCIe, или PCI-E – компьютерная шина, использующая программную модель шины PCI и высокопроизводительный физический протокол, основанный на последовательной передаче данных.

В отличие от шины PCI, использовавшей для передачи данных общую шину, PCI Express, в общем случае, является пакетной сетью с топологией типа звезда, устройства PCI Express взаимодействуют между собой через среду, образованную коммутаторами, при этом каждое устройство напрямую связано соединением типа точка-точка с коммутатором.



PCI Express

Для подключения устройства PCI Express используется двунаправленное последовательное соединение типа точка-точка, называемое линией.

Соединение (англ. link — связь, соединение) между двумя устройствами PCI Express и состоит из одной (x1) или нескольких (x2, x4, x8, x12, x16 и x32) двунаправленных последовательно соединённых линий. Каждое устройство должно поддерживать соединение по крайней мере с одной линией (x1).

На электрическом уровне каждое соединение использует низковольтную дифференциальную передачу сигнала (LVDS), приём и передача информации производится каждым устройством PCI Express по отдельным двум проводникам, таким образом, в простейшем случае, устройство подключается к коммутатору PCI Express всего лишь четырьмя проводниками.

PCI Express

Скорость передачи:

PCI Express 2.0 обладает скоростью передачи данных 5 GT/s (Гигатранзакций/с) и схемой кодирования 8b/10b.

PCI Express 3.0 обладает скоростью передачи данных 8 GT/s(Гигатранзакций/с). Но, несмотря на это, его реальная пропускная способность всё равно была увеличена вдвое по сравнению со стандартом PCI Express 2.0. Этого удалось достигнуть благодаря более агрессивной схеме кодирования 128b/130b, когда 128 бит данных пересылаемых по шине кодируются 130 битами.

PCI Express

Для расчёта пропускной способности шины необходимо учесть дуплексность и избыточность. Например, дуплексная пропускная способность соединения x1 PCI Express 2.0 составляет:

$5 \cdot 2 \cdot 0{,}8 = 8$ Гбит/с

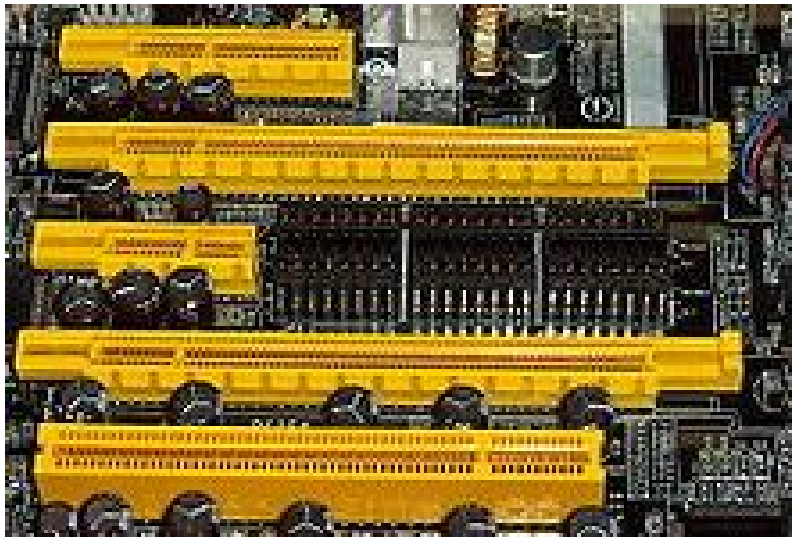
где 8 – битрейт, Гбит/с, 2 – учёт дуплексности (двунаправленности);

0,8 – учёт избыточности (8b/10b для 2.0; 0,985 – для 3.0).

В одну/обе стороны, Гбит/с

	Связей						
	x1	x2	x4	x8	x12	x16	x32
PCIe 1.0	2/4	4/8	8/16	16/32	24/48	32/64	64/128
PCIe 2.0	4/8	8/16	16/32	32/64	48/96	64/128	128/256
PCIe 3.0	8/16	16/32	32/64	64/128	96/192	128/256	256/512
PCIe 4.0 (предварительно) [3]	16/32	32/64	64/128	128/256	192/384	256/512	512/1024

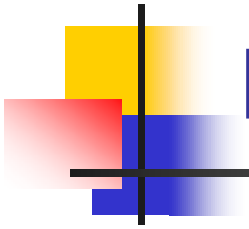
PCI Express



Слоты материнской платы DFI LanParty nForce4 SLI-DR (сверху вниз): x4 PCI Express, x16 PCI Express, x1 PCI Express, x16 PCI Express, стандартный 32-разрядный слот PC

2. Программирование микропроцессорной системы

Структура универсального 32 разрядного микропроцессора



Процессоры

Процессоры чисел с фиксированной* и плавающей точками непосредственно выполняют команды.

В своем составе они имеют:

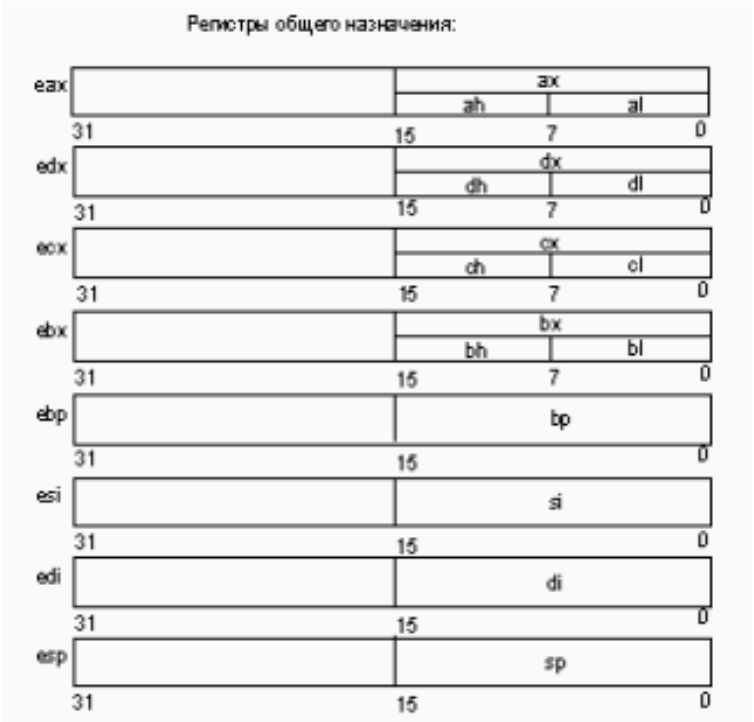
- АЛУ – арифметико-логические устройства, выполняющие операции;
- РОН – регистры общего назначения, необходимые для временного хранения операндов, управления адресацией команд и данных.

** под числами с фиксированной точкой в ПК нужно понимать целые числа*

Регистры

Регистры общего назначения:

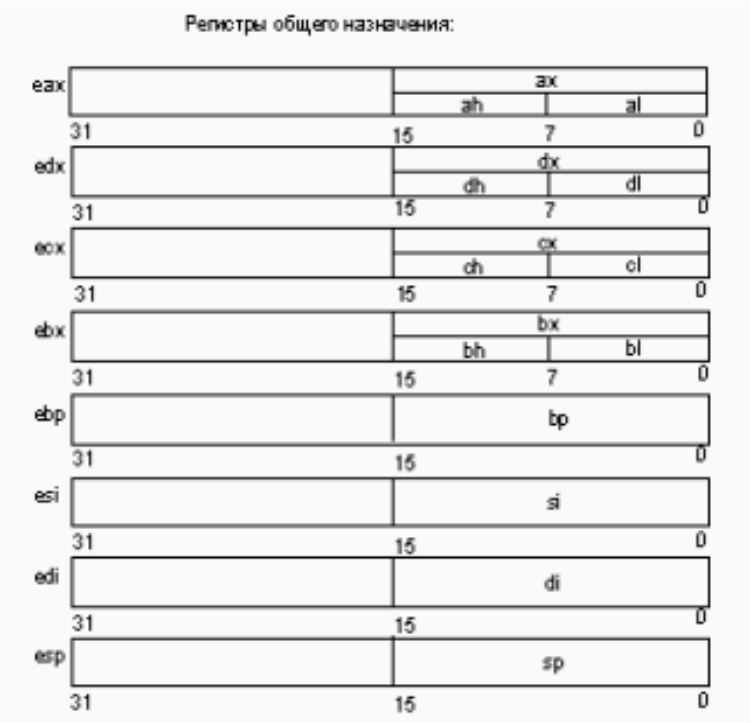
- **eax/ax/ah/al (Accumulator register)** - аккумулятор. Применяется для хранения промежуточных данных. В некоторых командах использование этого регистра обязательно;
- **ebx/bx/bh/bl (Base register)** - базовый регистр. Применяется для хранения базового адреса некоторого объекта в памяти;



Регистры

Регистры общего назначения:

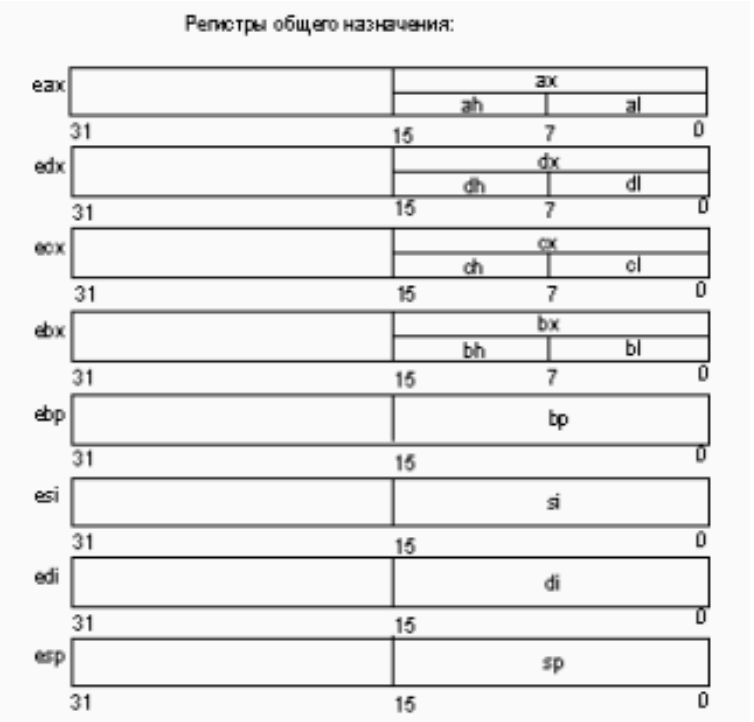
- **ecx/cx/ch/cl (Count register)** - регистр-счетчик. Применяется в командах, производящих некоторые повторяющиеся действия. Его использование зачастую неявно и скрыто в алгоритме работы соответствующей команды. К примеру, команда организации цикла loop кроме передачи управления команде, находящейся по некоторому адресу, анализирует и уменьшает на единицу значение регистра ecx/cx;



Регистры

Регистры общего назначения:

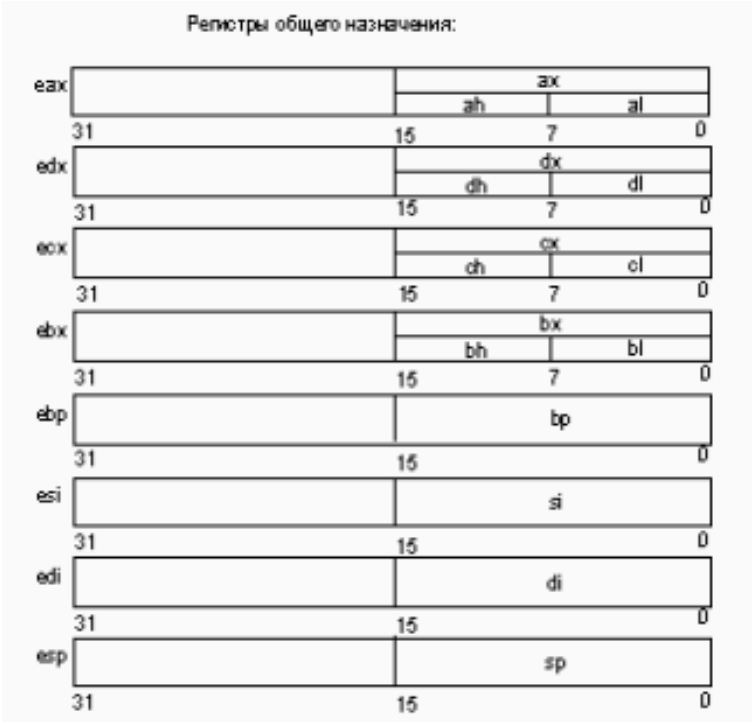
- **edx/dx/dh/dl (Data register)** - регистр данных. Так же, как и регистр eax/ax/ah/al, он хранит промежуточные данные. В некоторых командах его использование обязательно; для некоторых команд это происходит неявно.



Регистры

Регистры общего назначения:

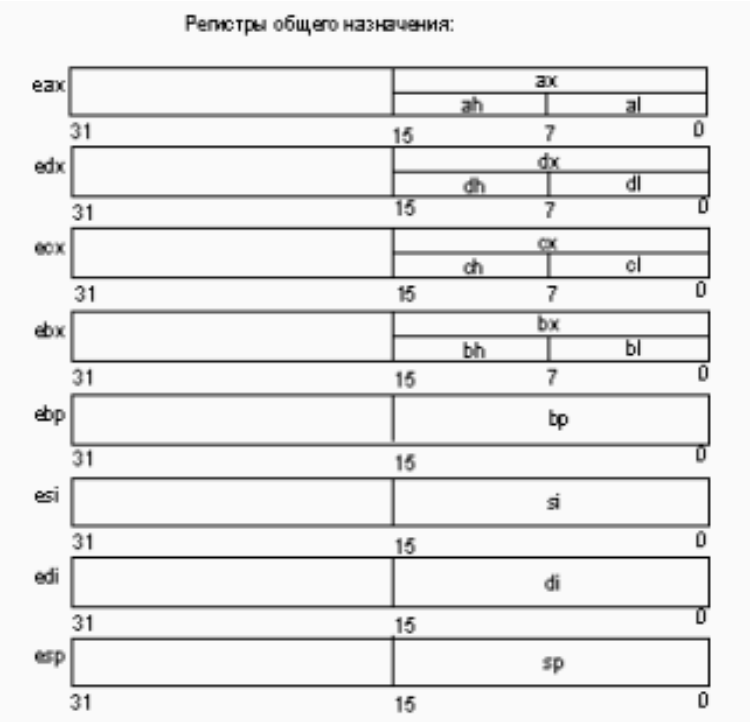
- **esi/si (Source Index register)** - индекс источника. Этот регистр в цепочечных операциях (операциях, производящих последовательную обработку цепочек элементов) содержит текущий адрес элемента в цепочке-источнике;
- **edi/di (Destination Index register)** - индекс приемника (получателя). Этот регистр в цепочечных операциях содержит текущий адрес в цепочке-приемнике.



Регистры

Регистры общего назначения:

- **esp/sp (Stack Pointer register)** - регистр указателя стека. Содержит указатель вершины стека в текущем сегменте стека.
- **ebp/bp (Base Pointer register)** - регистр указателя базы кадра стека. Предназначен для организации произвольного доступа к данным внутри стека.

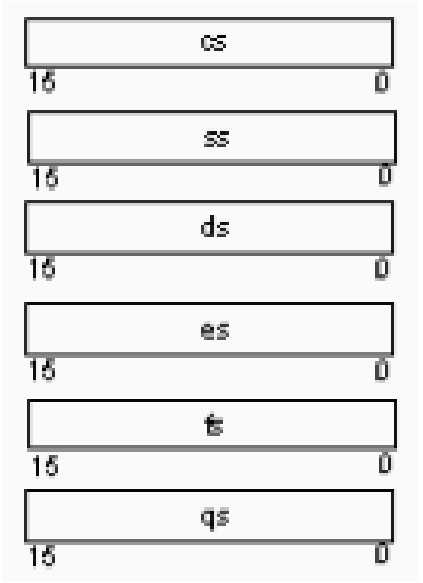


Регистры

Сегментные регистры

В программной модели микропроцессора имеется шесть сегментных регистров: **cs, ss, ds, es, gs, fs.**

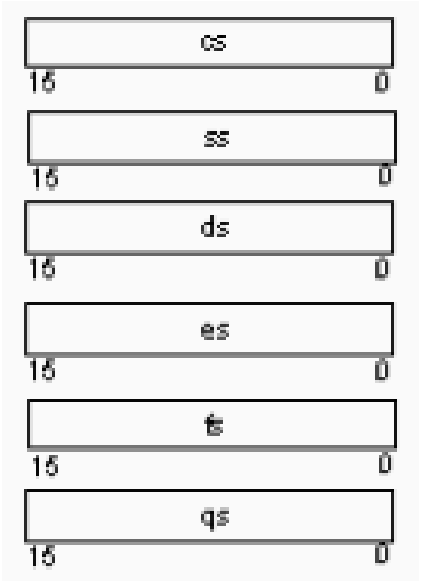
Их существование обусловлено спецификой организации и использования оперативной памяти микропроцессорами Intel. Она заключается в том, что микропроцессор аппаратно поддерживает структурную организацию программы в виде трех частей, называемых сегментами (*сегменты команд, данных и стека*). Соответственно, такая организация памяти называется сегментной.



Регистры

Сегментные регистры

Для того чтобы указать на сегменты, к которым программа имеет доступ в конкретный момент времени, и предназначены сегментные регистры. Фактически, с небольшой поправкой, как мы увидим далее, в этих регистрах содержатся адреса памяти с которых начинаются соответствующие сегменты. Логика обработки машинной команды построена так, что при выборке команды, доступе к данным программы или к стеку неявно используются адреса во вполне определенных сегментных регистрах.

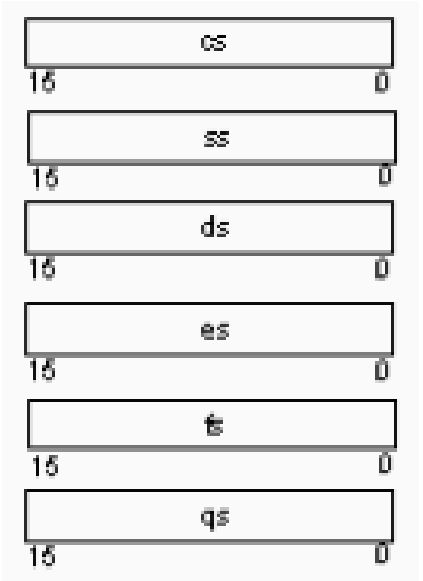


Регистры

Сегментные регистры

Микропроцессор поддерживает следующие типы сегментов:

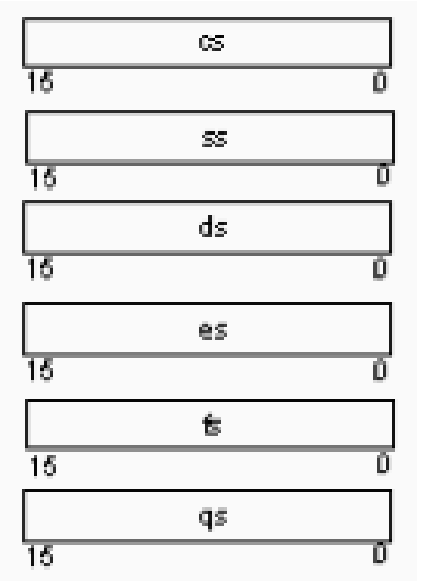
- **Сегмент кода**. Содержит команды программы. Для доступа к этому сегменту служит регистр **cs (code segment register)** - сегментный регистр кода. Он содержит адрес сегмента с машинными командами, к которому имеет доступ микропроцессор (то есть эти команды загружаются в конвейер микропроцессора).



Регистры

Сегментные регистры

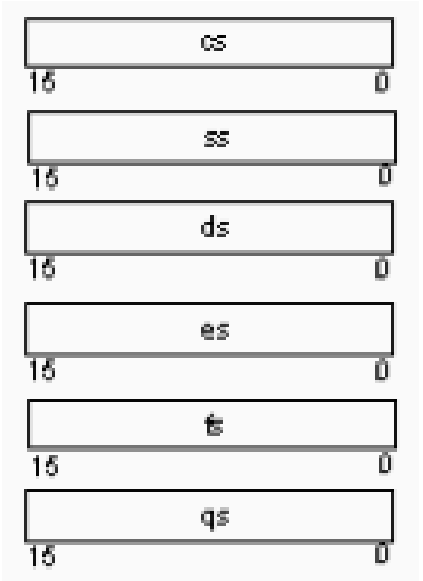
- **Сегмент данных.** Содержит обрабатываемые программой данные. Для доступа к этому сегменту служит регистр **ds (data segment register)** - сегментный регистр данных, который хранит адрес сегмента данных текущей программы.
- **Сегмент стека**. Этот сегмент представляет собой область памяти, называемую стеком. Работу со стеком микропроцессор организует по следующему принципу: последний записанный в эту область элемент выбирается первым. Для доступа к этому сегменту служит регистр **ss (stack segment register)** - сегментный регистр стека, содержащий адрес сегмента стека.



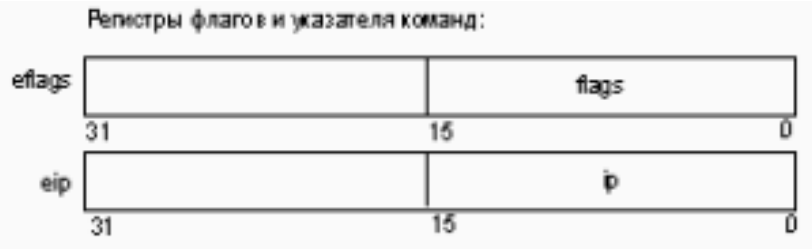
Регистры

Сегментные регистры

- **Дополнительный сегмент данных.** Неявно алгоритмы выполнения большинства команд предполагают, что обрабатываемые данные расположены в сегменте данных, связанном с регистром ds. Если программе недостаточно одного сегмента данных, то она имеет возможность использовать еще три дополнительных. В отличие от основного сегмента данных при использовании дополнительных сегментов их адреса требуется указывать явно с помощью специальных префиксов. Адреса дополнительных сегментов данных должны содержаться в регистрах **es, gs, fs (extension data segment registers).**



Регистры



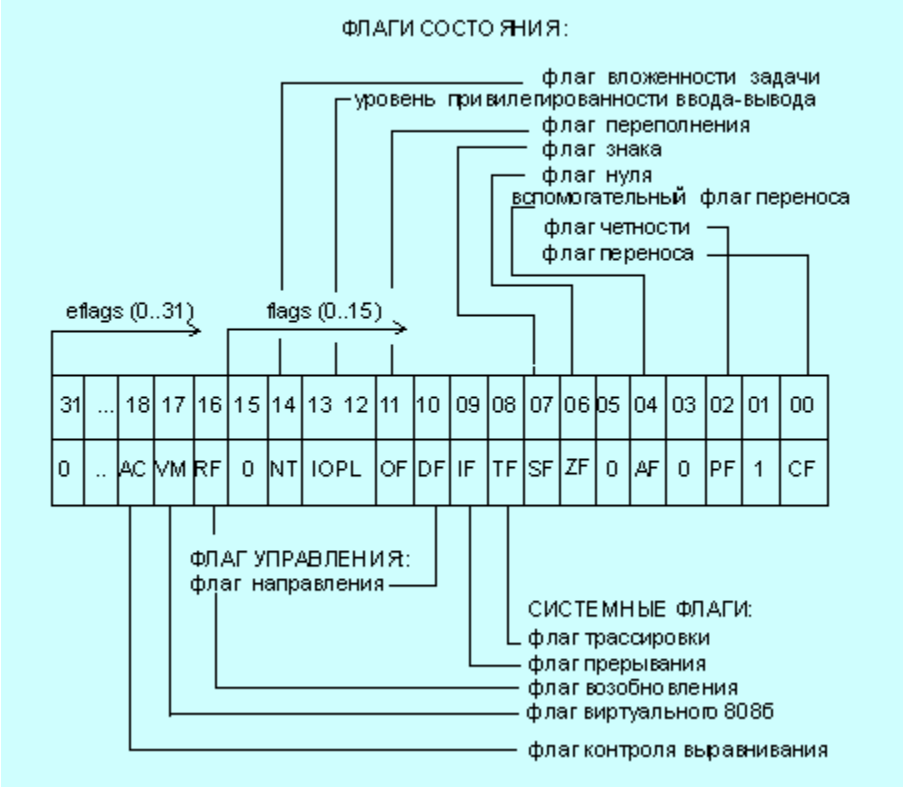
В микропроцессор включены регистры, которые постоянно содержат информацию о состоянии как самого микропроцессора, так и программы, команды которой в данный момент загружены на конвейер. К этим регистрам относятся:

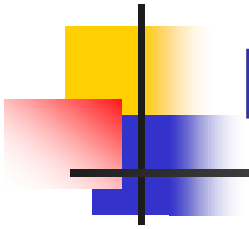
- **регистр флагов eflags/flags**;
- **регистр указателя команды eip/ip**.

Используя эти регистры, можно получать информацию о результатах выполнения команд и влиять на состояние самого микропроцессора.

Регистры

flags/flags (flag register) - регистр флагов. Отдельные биты данного регистра имеют определенное функциональное назначение и называются флагами.



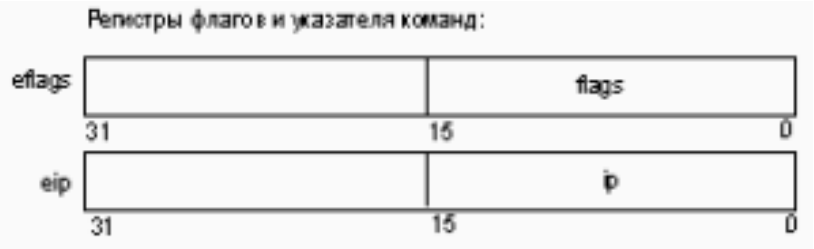


Регистры

Флаги регистра eflags/flags :

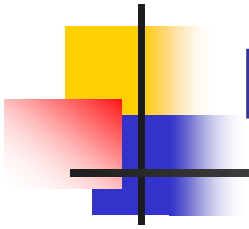
- **8 флагов состояния.** Эти флаги могут изменяться после выполнения машинных команд. Флаги состояния регистра eflags отражают особенности результата исполнения арифметических или логических операций. Это дает возможность анализировать состояние вычислительного процесса и реагировать на него с помощью команд условных переходов и вызовов подпрограмм.
- **1 флаг управления.** Обозначается df (Directory Flag). Он находится в 10-м бите регистра eflags и используется цепочечными командами. Значение флага df определяет направление поэлементной обработки в этих операциях: от начала строки к концу (df = 0) либо наоборот
- **5 системных флагов**, управляющих вводом/выводом, маскируемыми прерываниями, отладкой, переключением между задачами и виртуальным режимом 8086.

Регистры



eip/ip (Instraction Pointer register) - регистр-указатель команд.

Регистр eip/ip имеет разрядность 32/16 бит и содержит смещение следующей подлежащей выполнению команды относительно содержимого сегментного регистра cs в текущем сегменте команд. Этот регистр непосредственно недоступен программисту, но загрузка и изменение его значения производятся различными командами управления, к которым относятся команды условных и безусловных переходов, вызова процедур и возврата из процедур. Возникновение прерываний также приводит к модификации регистра eip/ip.



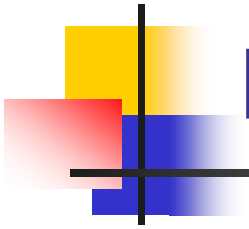
Регистры

Системные регистры микропроцессора

Выполняют специфические функции в системе. Использование системных регистров жестко регламентировано. Именно они обеспечивают работу защищенного режима. Их также можно рассматривать как часть архитектуры микропроцессора, которая намеренно оставлена видимой для того, чтобы квалифицированный системный программист мог выполнить самые низкоуровневые операции.

Системные регистры можно разделить на три группы:

- четыре регистра управления;
- четыре регистра системных адресов;
- восемь регистров отладки.



Регистры

Регистры управления

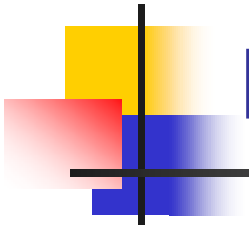
В группу регистров управления входят 4 регистра: cr0, cr1, cr2, cr3.

Эти регистры предназначены для общего управления системой. Регистры управления доступны только программам с уровнем привилегий 0 (ОС).

Регистр cr0 содержит системные флаги, управляющие режимами работы микропроцессора и отражающие его состояние глобально, независимо от конкретных выполняющихся задач.

Назначение системных флагов:

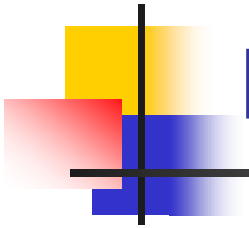
- pe (Protect Enable) - разрешение защищенного режима работы
- mp (Math Present) - наличие сопроцессора. Всегда 1.
- ts (Task Switched) - переключение задач.
- am (Aligment Mask) - маска выравнивания.
- cd (Cache Disable) - запрещение кэш-памяти первого уровня.
- pg (PaGing) - разрешение страничного преобразования.



Регистры

Регистр cr2 используется при страничной организации оперативной памяти для регистрации ситуации, когда текущая команда обратилась по адресу, содержащемуся в странице памяти, отсутствующей в данный момент времени в памяти. В такой ситуации в микропроцессоре возникает исключительная ситуация с номером 14, и линейный 32-битный адрес команды, вызвавшей это исключение, записывается в регистр cr2. Имея эту информацию, обработчик исключения 14 определяет нужную страницу, осуществляет ее подкачку в память и возобновляет нормальную работу программы.

Регистр cr3 также используется при страничной организации памяти. Это так называемый регистр каталога страниц первого уровня. Он содержит 20-битный физический базовый адрес каталога страниц текущей задачи. Этот каталог содержит 1024 32-битных дескриптора, каждый из которых содержит адрес таблицы страниц второго уровня. В свою очередь каждая из таблиц страниц второго уровня содержит 1024 32-битных дескриптора, адресующих страничные кадры в памяти. Размер страничного кадра - 4 Кбайт.



Регистры

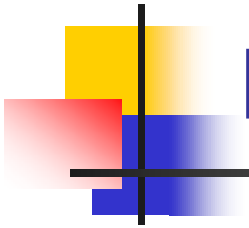
Регистры системных адресов

Эти регистры еще называют регистрами управления памятью.

Они предназначены для защиты программ и данных в мультизадачном (защищенном) режиме работы микропроцессора.

При работе в защищенном режиме микропроцессора адресное пространство делится на:

- глобальное - общее для всех задач;
- локальное - отдельное для каждой задачи.

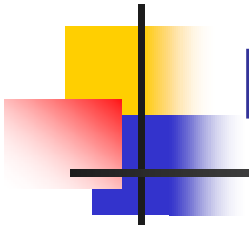


Регистры

Регистры системных адресов

- **регистр таблицы глобальных дескрипторов gdtr (Global Descriptor Table Register),** имеющий размер 48 бит и содержащий 32-битовый (биты 16-47) базовый адрес глобальной дескрипторной таблицы GDT и 16-битовое (биты 0-15) значение предела, представляющее собой размер в байтах таблицы GDT;
- **регистр таблицы локальных дескрипторов ldtr (Local Descriptor Table Register)** имеющий размер 16 бит и содержащий так называемый селектор дескриптора локальной дескрипторной таблицы LDT. Этот селектор является указателем в таблице GDT, который и описывает сегмент, содержащий локальную дескрипторную таблицу LDT.

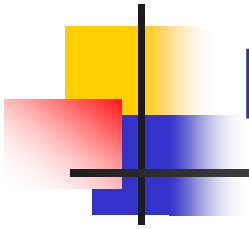
Регистры



Регистры

Регистры системных адресов

- **регистр таблиц дескрипторов прерываний idtr (Interrupt Descriptor Table Register)** имеющий размер 48 бит и содержащий 32-битовый (биты 16-47) базовый адрес дескрипторной таблицы прерываний IDT и 16-битовое (биты 0—15) значение предела, представляющее собой размер в байтах таблицы IDT;
- **16-битовый регистр задачи tr (Task Register)**, который подобно регистру ldtr, содержит селектор, то есть указатель на дескриптор в таблице GDT. Этот дескриптор описывает текущий сегмент состояния задачи (TSS - Task Segment Status). Этот сегмент создается для каждой задачи в системе, имеет жестко регламентированную структуру и содержит контекст (текущее состояние) задачи. Основное назначение сегментов TSS - сохранять текущее состояние задачи в момент переключения на другую задачу.

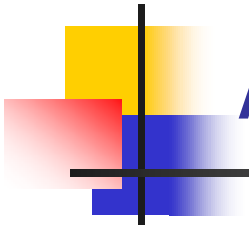


Регистры

Регистры отладки

Это группа регистров, предназначенных для аппаратной отладки. Средства аппаратной отладки впервые появились в микропроцессоре i486. Аппаратно микропроцессор содержит восемь регистров отладки, но реально из них используются только 6.

- **Регистры dr0, dr1, dr2, dr3** имеют разрядность 32 бит и предназначены для задания линейных адресов четырех точек прерывания.
- **Регистр dr6** называется регистром состояния отладки. Биты этого регистра устанавливаются в соответствии с причинами, которые вызвали возникновение последнего исключения с номером 1.
- **Регистр dr7** называется регистром управления отладкой. В нем для каждого из четырех регистров контрольных точек отладки имеются поля, с помощью которых можно уточнить условия, при которых следует сгенерировать прерывание.



Адресация и сегментация памяти

Адресация ячеек памяти в реальном режиме

От процессоров 8086/88 достался своеобразный образ задания адреса ячейки памяти в виде указателя «seg:offset».

Логический адрес состоит из 16-разрядных компонент: адреса сегмента памяти и смещения внутри сегмента.

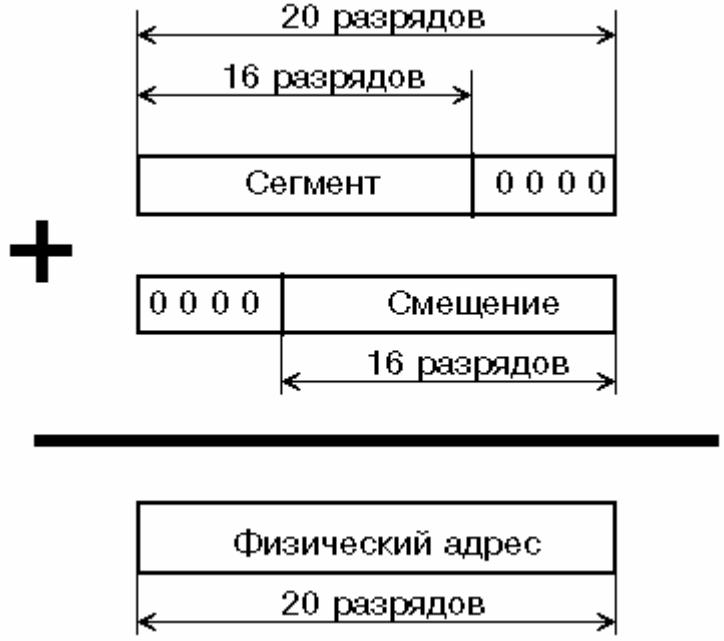
Адрес сегмента хранится в сегментном регистре ds, cs, ss.

Смещение хранится:

- для данных – в командах, которые обрабатывают эти данные;
- для команд – в регистре ip, который изменяется автоматически, после выполнения каждой команды или принудительно при переходах.
- для стека – в регистре sp, указывающем на вершину стека или в других регистрах, использующихся при работе со стеком.

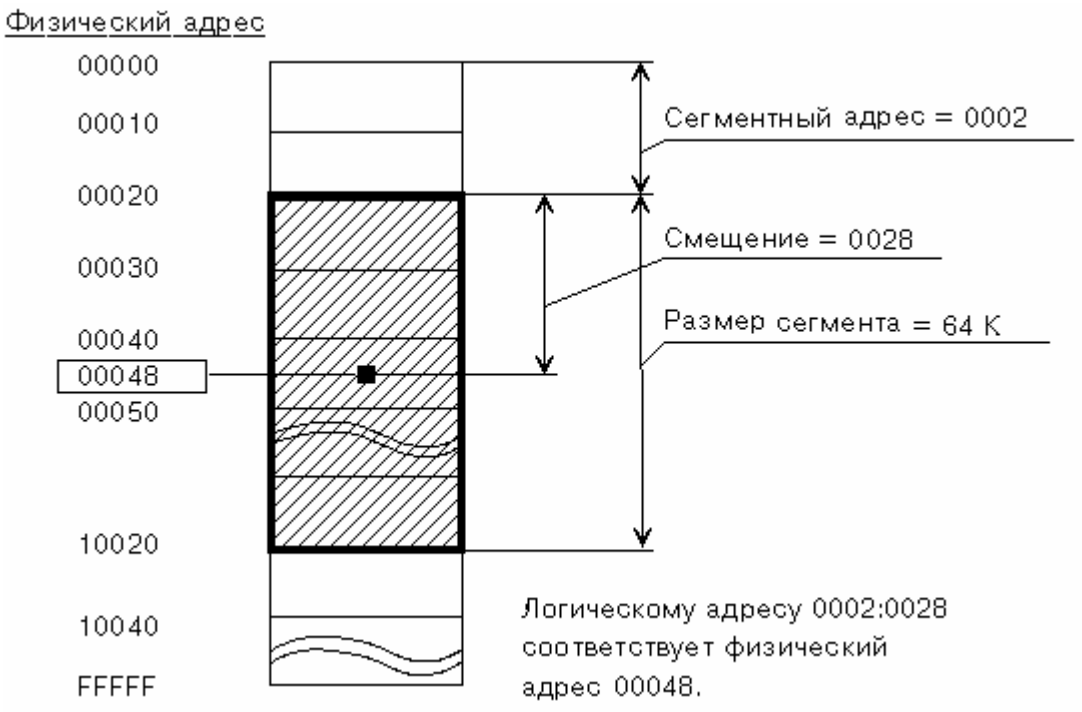
Адресация и сегментация памяти

Для получения 20-разрядного физического адреса к сегментной компоненте приписывается справа четыре нулевых бита (для расширения до 20 разрядов), затем полученное число складывается с компонентой смещения. Перед сложением к компоненте смещения слева дописывается четыре нулевых бита (также для расширения до 20 разрядов).



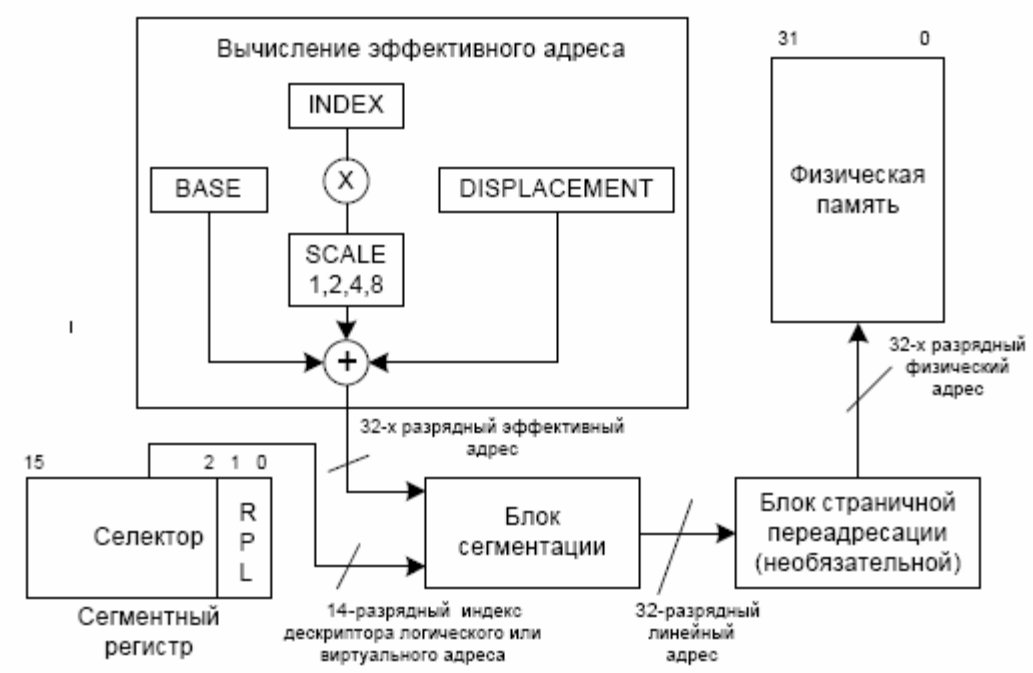
Адресация и сегментация памяти

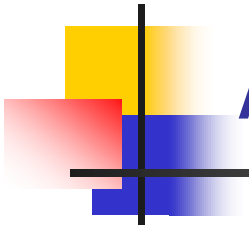
При запуске программы ей выдается несколько сегментов. Каждый сегмент должен иметь свой адрес, кратный 16 и размер не более 64 кбайт. Чтобы получить доступ к данным внутри сегмента используется смещение. Так как сегмент занимает не более 64 кбайт, то для хранения смещения хватит 16 разрядов.



Адресация и сегментация памяти

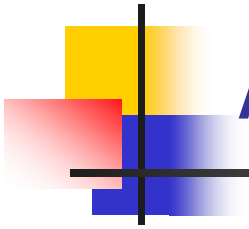
Организация и адресация памяти в защищенном режиме





Адресация и сегментация памяти

В защищенном режиме, также как и в реальном, существуют понятия логического и физического адреса. Кроме сегментации в защищенном режиме возможно разбиение (Paging) логической памяти на страницы размером 4 Кбайт, каждая из которых может отображаться на любую область физической памяти. Начиная с 5-го поколения, появилась возможность увеличения размера страницы до 4 Мбайт. Сегментация и разбиение на страницы могут применяться в любых сочетаниях. Сегментация является средством организации логической памяти на прикладном уровне. Разбиение на страницы применяется на системном уровне для управления физической памятью. Сегменты и страницы могут выгружаться из физической оперативной памяти на диск и по мере необходимости подкачиваться с него обратно в физическую память. Таким образом реализуется виртуальная память.



Адресация и сегментация памяти

Применительно к памяти различают три адресных пространства:

- логическое;
- линейное;
- физическое.

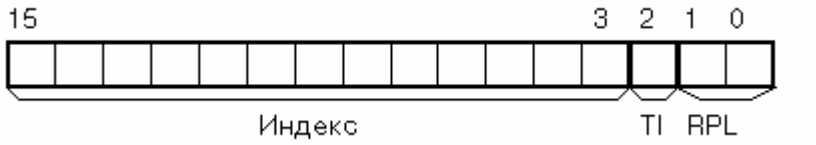
Основным режимом работы 32-разрядных процессоров считается защищенный режим, в котором работают все механизмы преобразования адресных пространств.

Логический адрес (или виртуальных адрес) состоит из селектора сегмента и эффективного адреса (смещения), обозначается seg:offset.

Адресация и сегментация памяти

Преобразование логического адреса в физический выполняется не простым сложением со сдвигом, а при помощи специальных таблиц преобразования адресов (LDT, GDT). В процессе преобразования логического адреса в линейный процессор прибавляет к базовому 32-разрядному адресу, содержащемуся в дескрипторе из таблицы, 32-разрядное смещение, т.е. линейный адрес образуется сложением базового адреса сегмента с эффективным адресом.

Индекс дескриптора в таблице берется из т.н. селектора, загруженного в сегментный регистр.

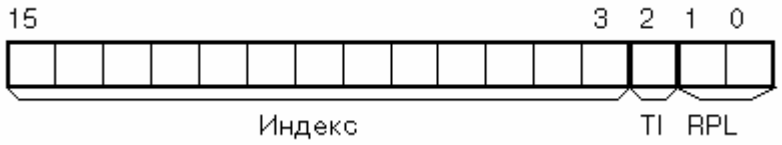


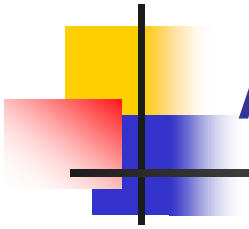
Адресация и сегментация памяти

В качестве индекса выступают старшие 13 бит.

Два младших бита (RPL) используются системой защиты памяти (они определяют уровень запроса, который далее сравнивается с уровнем сегмента).

Поле TI (Table Indicator) состоит из одного бита. Если этот бит равен нулю, для преобразования адреса используется так называемая глобальная таблица дескрипторов GDT (Global Descriptor Table), в противном случае - локальная таблица дескрипторов LDT (Local Descriptor Table).

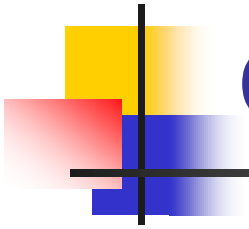




Адресация и сегментация памяти

Таблица дескрипторов - это просто таблица преобразования адресов, содержащая базовые адреса сегментов и некоторую другую информацию. То есть каждый элемент таблицы дескрипторов (дескриптор) содержит базовый адрес сегмента и другую информацию, описывающую сегмент.

Таблица GDT - единственная в системе. Обычно в ней находятся описания сегментов операционной системы. Таблиц LDT может быть много. Эти таблицы содержат описания сегментов программ, работающих под управлением операционной системы, т.е. отдельных задач. В каждый данный момент времени процессор может использовать только одну таблицу LDT.

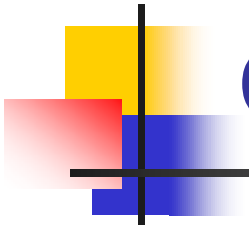


Страничная организация памяти

Страничная память - способ организации виртуальной памяти, при котором единицей отображения виртуальных адресов на физические является регион постоянного размера (т. н. страница). Типичный размер страницы 4096 байт.

Задачи:

- поддержка изоляции процессов и защиты памяти путём создания своего собственного виртуального адресного пространства для каждого процесса
- поддержка изоляции области ядра от кода пользовательского режима
- поддержка памяти «только для чтения» и неисполняемой памяти
- поддержка отгрузки давно не используемых страниц в область подкачки на диске (см. свопинг)
- поддержка отображённых в память файлов, в том числе загрузочных модулей
- поддержка разделяемой между процессами памяти, в том числе с для экономии физических страниц.



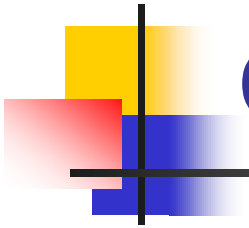
Страничная организация памяти

Адрес, используемый в машинном коде, то есть значение указателя, называется «виртуальный адрес» (или «математический адрес»).

Адрес, выставляемый процессором на шину, называется «физический адрес».

Процессор содержит в себе небольшой объём сверхбыстрой ассоциативной памяти, т. н. TLB (Translation Lookaside Buffer), в котором содержится преобразование нескольких (часто 64) виртуальных адресов страниц в физические. Все обращения процессора к памяти подлежат трансляции адресов через TLB.

Так как 64 строк таблицы явно недостаточно для реальных задач, в архитектуре используются таблицы страниц, размещённые в основной памяти. Каждая таблица страниц сама является страницей с теми же требованиями по выравниваю и тем же размером, и состоит из записей таблицы страниц (page table entries - PTE). Широко используется и отображение самой таблицы страниц как одной из страниц данных для внесения изменений в записи.

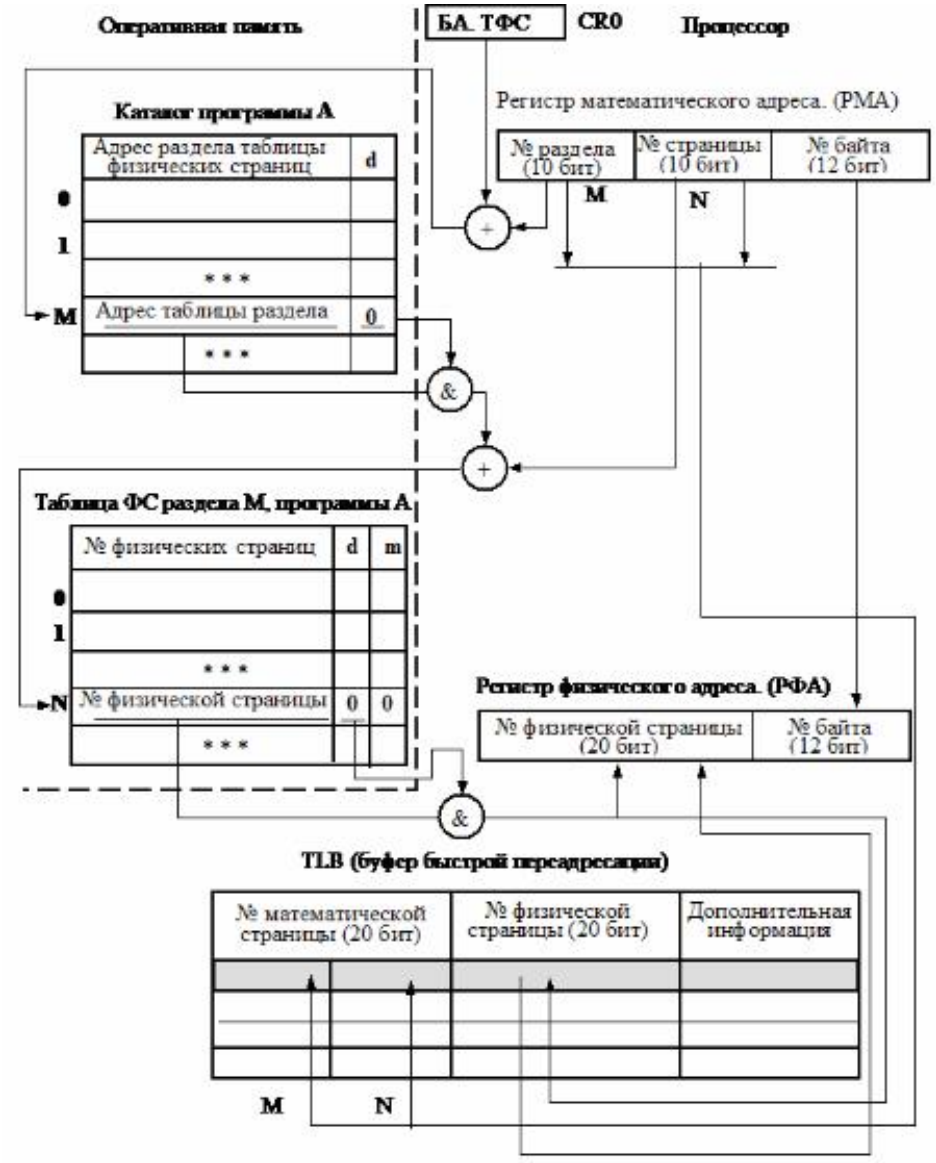


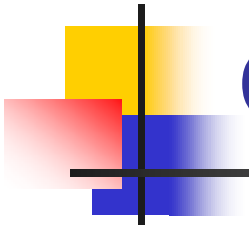
Страничная организация памяти

Запись таблицы страниц обычно содержит в себе следующую информацию:

- флаг «страница отображена»
- физический адрес
- флаг «страница доступна из режима пользователя». При неустановке данного флага страница доступна только из режима ядра.
- флаг «страница доступна только на чтение». В некоторых случаях используется только для режима пользователя, то есть в режиме ядра все страницы всегда доступны на запись.
- флаг «страница недоступна на исполнение».
- режим использования кэша для страницы. Влияет на тип шинных транзакций, инициируемых процессором при обращении через данную запись. Особенно часто используется для видеопамяти (комбинированная запись) и для отображенных в память регистров устройств (полное отсутствие кэширования).

Общая схема преобразования





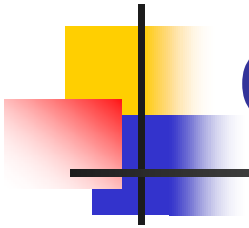
Страничная организация памяти

Если обращение к памяти не может быть оттранслировано через TLB, то микрокод процессора обращается к таблицам страниц и пытается загрузить PTE оттуда в TLB. Если и после такой попытки сохранились проблемы, то процессор исполняет специальное прерывание, называемое «отказ страницы» (page fault). Обработчик этого прерывания находится в подсистеме виртуальной памяти ядра ОС.

Причины отказа страницы (page fault):

- не существует таблицы, отображающей данный регион
- PTE не имеет взведённого флага «страница отображена».
- попытка обратиться из пользовательского режима к странице «только для ядра».
- попытка записи в страницу «только для чтения».
- попытка исполнения кода из страницы «исполнение запрещено».

Обработчик отказов в ядре может загрузить нужную страницу из файла или же из области подкачки (см. свопинг) или возбудить исключительную ситуацию

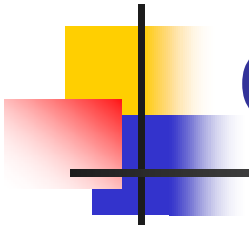


Страничная организация памяти

Каждый процесс имеет свой собственный набор таблиц страниц. Регистр «директория страниц» перегружается при каждом переключении контекста процесса. Также необходимо сбросить ту часть TLB, которая относится к данному процессу.

В большинстве случаев ядро ОС помещается в то же адресное пространство, что и процессы, для него резервируются верхние 1-2 гигабайта 32битного адресного пространства каждого процесса. Это делается с целью избежать переключения таблиц страниц при входе в ядро и выходе из него. Страницы ядра помечаются как недоступные для кода режима пользователя.

Память региона ядра часто совершенно одинакова для всех процессов, однако некоторые подрегионы региона ядра (например, регион Windows, где находится подсистема графики и драйвер видео) могут быть различными для разных групп процессов (сессий).



Страничная организация памяти

Обработчик отказа страницы в ядре способен прочитать данную страницу из файла.

Это приводит к возможности легкой реализации отображенных в память файлов. Концептуально это то же, что выделение памяти и чтение в неё отрезка файла, с той разницей, что чтение осуществляется неявно «по требованию», выраженному отказом страницы при попытке обращения к ней.

Вторым преимуществом такого подхода является - в случае отображения «только для чтения» - разделение одной и той же физической памяти между всеми процессами, отображающими данный файл (выделенная же память своя у каждого процесса).

Третьим преимуществом является возможность «забывания» (discard) некоторых отображенных страниц без выгрузки их в область подкачки, обязательной для выделенной памяти. В случае повторной потребности в странице она может быть быстро загружена из файла снова.

Система команд микропроцессора

Формат команды процессора допускает наличие полей

Код операции (КОП), 1,2 байта	Байты адресации, 0..2 байта	Смещение, 0..4 байта	Операнд(ы), 0..4 байта

Команды оперируют байтами или словами (16, 32 бит).

Команды бывают безадресные, одно-, двух- и более адресные.

Двухадресные команды:

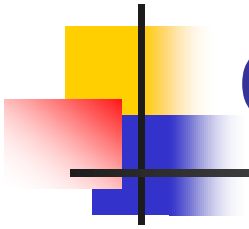
регистр - регистр

память - регистр

непосредственные - регистр

память - память

непосредственные - память

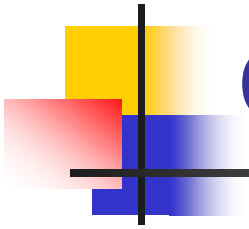


Система команд микропроцессора

Способы адресации:

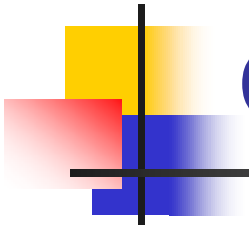
- регистровая
- косвенно-регистровая, например, DS:(E)BX
- прямая, например, DS:d32
- базовая, например, DS:EAX+d32
- индексная, например, DS:ESI+d32
- базово-индексная, например, DS:EAX+ESI

и др.



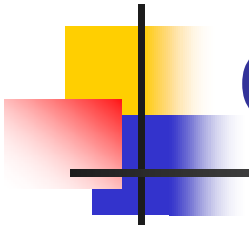
Система команд микропроцессора

- Команды пересылки данных
  - MOV – пересылка операндов
  - PUSH – запись операнда в стек
  - PUSHA – запись всех регистров в стек
  - POP – извлечение из стека
  - POPA – извлечение всех регистров из стека
  - IN – ввод данных из порта
  - OUT – запись данных в порт
  - и др.
- Арифметические операции
  - ADD – сложение
  - SUB – вычитание
  - MUL – умножение
  - DIV – деление
  - и д.р



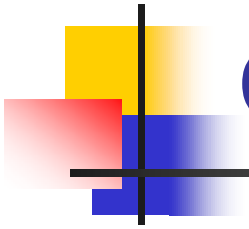
Система команд микропроцессора

- Логические команды и сдвиги
  - NOT – побитовое отрицание
  - OR – побитовое ИЛИ
  - AND – побитовое И
  - XOR – побитовое исключающее ИЛИ
  - SHL – сдвиг влево
  - SHR – сдвиг вправо
  - ROL – циклический сдвиг влево
  - ROR – циклический сдвиг вправо
  - и др.



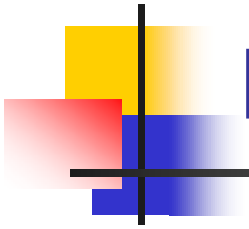
Система команд микропроцессора

- Команды управления
  - JMP – безусловный переход
  - CALL – вызов подпрограммы
  - RET – возврат из подпрограммы
  - Jусл – условный переход, например,
    - JS – переход, если результат последней операции – отрицательный (анализируется бит S регистра флагов)
    - JNZ - переход, если результат последней операции не равен рулю (анализируется бит Z регистра флагов)
  - LOOP – условный переход по состоянию регистра (E)CX, который уменьшается на единицу при каждой итерации
  - LOOP усл – цикл с дополнительной проверки (для досрочного выхода), например,
    - LOOPZ – досрочный выход из цикла при нулевом результате.



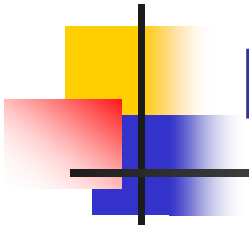
Система команд микропроцессора

- Команды управления
  - INT – прерывания
  - IRET – возврат из прерывания
  - CLI – запрет прерываний
  - STI – разрешение прерываний
- Команды процессора с плавающей точкой
- и др.



Прерывания

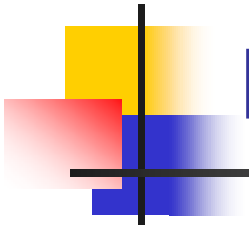
Прерывание (англ. interrupt) - сигнал, сообщающий процессору о наступлении какого-либо события. При этом выполнение текущей последовательности команд приостанавливается и управление передаётся обработчику прерывания, который реагирует на событие и обслуживает его, после чего возвращает управление в прерванный код.



Прерывания

В зависимости от источника возникновения сигнала прерывания делятся на:

- асинхронные или внешние (аппаратные) - события, которые исходят от внешних источников (например, периферийных устройств) и могут произойти в любой произвольный момент: сигнал от таймера, нажатие клавиш клавиатуры, движение мыши. Факт возникновения в системе такого прерывания трактуется как запрос на прерывание (англ. Interrupt request, IRQ);
- синхронные или внутренние - события в самом процессоре как результат нарушения каких-то условий при исполнении машинного кода: деление на ноль или переполнение, обращение к недопустимым адресам или недопустимый код операции;
- программные (частный случай внутреннего прерывания) - инициируются исполнением специальной инструкции в коде программы. Программные прерывания как правило используются для обращения к программного обеспечения драйверов и операционной системы.

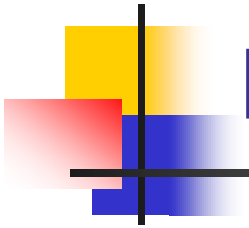


Прерывания

В зависимости от возможности запрета внешние прерывания делятся на:

- маскируемые - прерывания, которые можно запрещать установкой соответствующих битов в регистре маскирования прерываний (в x86-процессорах — сбросом флага IF в регистре флагов);
- немаскируемые (англ. Non maskable interrupt, NMI) - обрабатываются всегда, независимо от запретов на другие прерывания. К примеру, такое прерывание может быть вызвано сбоем в микросхеме памяти.

Обработчики прерываний обычно пишутся таким образом, чтобы время их обработки было как можно меньшим, поскольку во время их работы могут не обрабатываться другие прерывания, а если их будет много (особенно от одного источника), то они могут теряться.

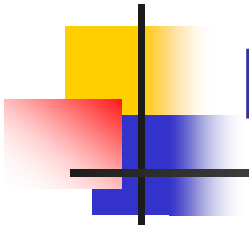


Прерывания

До окончания обработки прерывания обычно устанавливается запрет на обработку этого типа прерывания, чтобы процессор не входил в цикл обработки одного прерывания.

Приоритезация означает, что все источники прерываний делятся на классы и каждому классу назначается свой уровень приоритета запроса на прерывание. Приоритеты могут обслуживаться как относительные и абсолютные.

- Относительное обслуживание прерываний означает, что если во время обработки прерывания поступает более приоритетное прерывание, то это прерывание будет обработано только после завершения текущей процедуры обработки прерывания.
- Абсолютное обслуживание прерываний означает, что если во время обработки прерывания поступает более приоритетное прерывание, то текущая процедура обработки прерывания вытесняется, и процессор начинает выполнять обработку вновь поступившего более приоритетного прерывания. После завершения этой процедуры процессор возвращается к выполнению вытесненной процедуры обработки прерывания.



Прерывания

Вектор прерывания - закреплённый за прерыванием номер, который идентифицирует соответствующий обработчик прерываний. Векторы прерываний объединяются в таблицу векторов прерываний, содержащую адреса обработчиков прерываний. Местоположение таблицы зависит от типа и режима работы процессора.

В реальном режиме таблица прерываний находится в самом начале памяти (по адресу 0)

В защищенном режиме используется таблица дескрипторов прерываний IDTR, содержащая записи о адресах переходов и передаваемых параметрах.

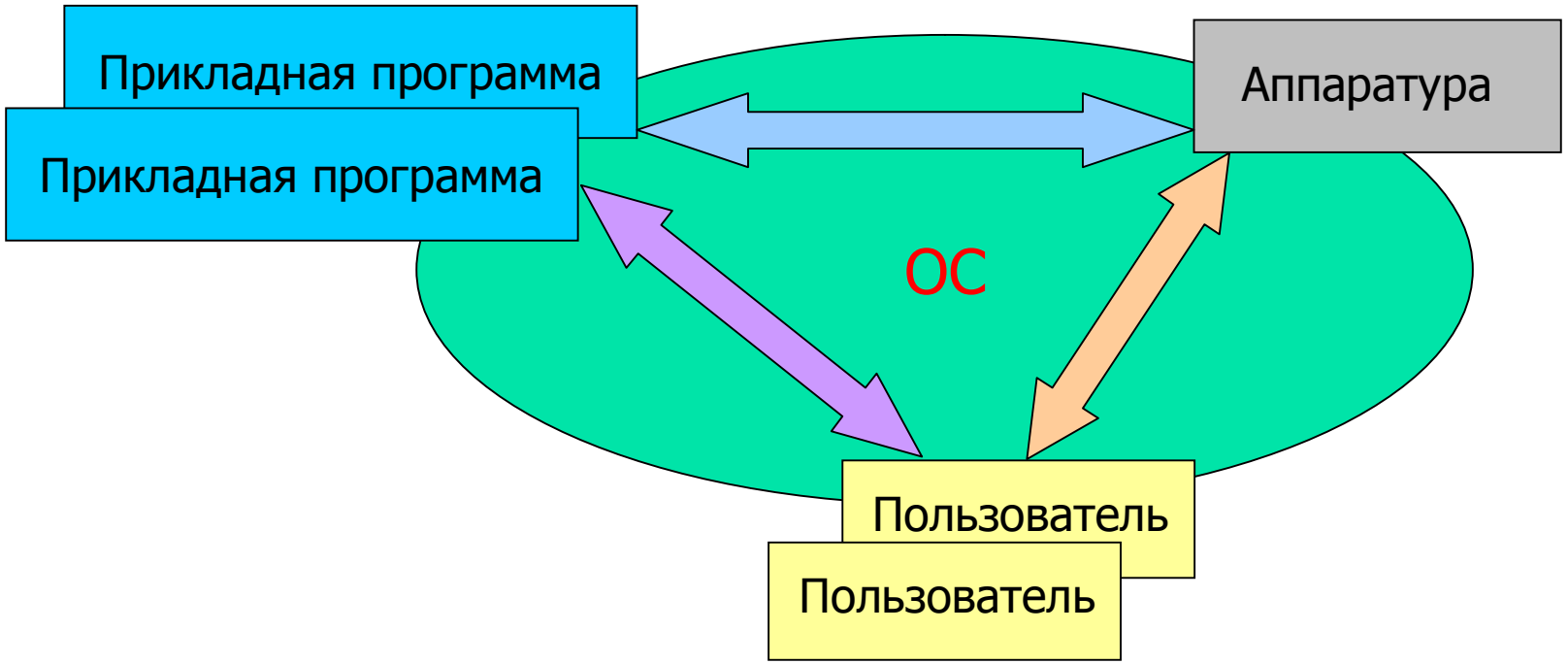
Программное обеспечение микропроцессорной системы

Программное обеспечение:

- системное ПО (ОС, драйверы)
- прикладное ПО

Программное обеспечение микропроцессорной системы

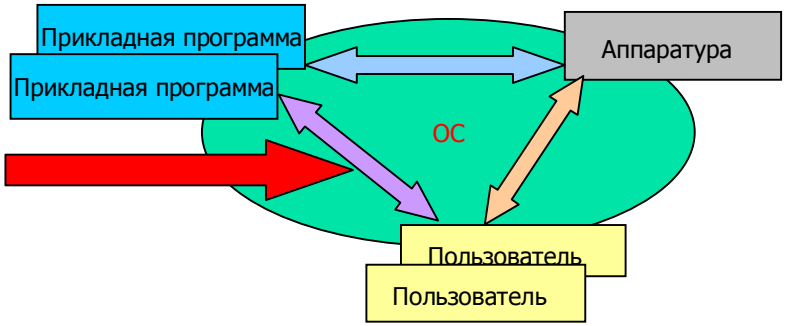
Операционная система



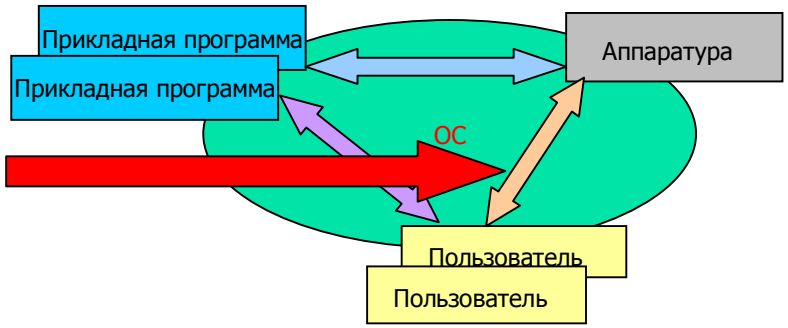
Программное обеспечение микропроцессорной системы

Операционная система

- файловая система
- установка и запуск прикладных программ
- интерфейс пользователя



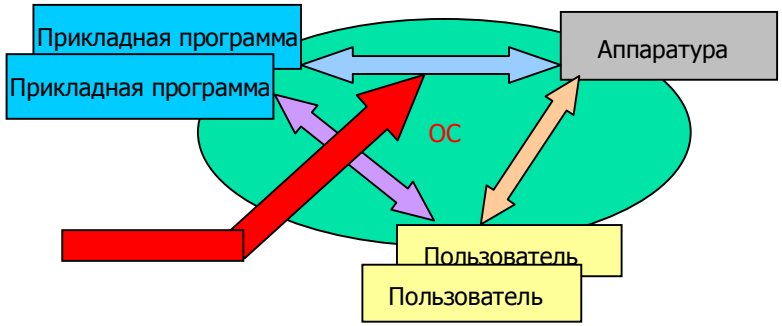
- установка
- настройка
- диагностика оборудования

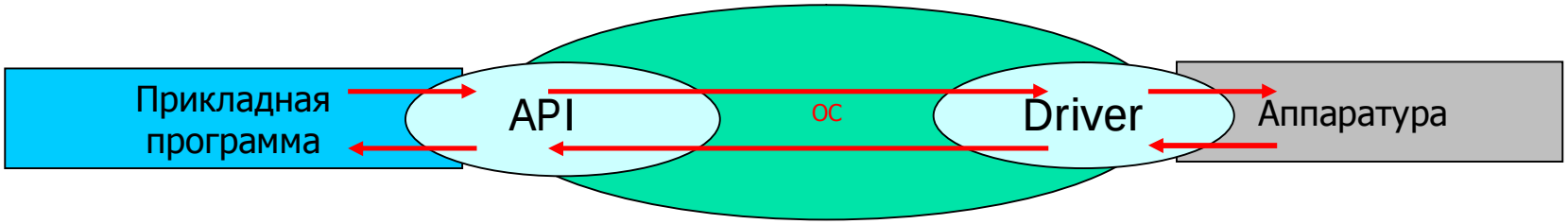


Программное обеспечение микропроцессорной системы

Операционная система

• предоставление и освобождение ресурсов (памяти, процессорного времени) при запуске, работе, остановке
• доступ и управление устройствами посредством функций API (applied program interface – интерфейс прикладных программ) и драйверов устройств





Программное обеспечение микропроцессорной системы

Взаимодействие прикладной программы с оборудованием (варианты и история)