

# Запоминающие устройства

Часть 3

Способом, или **режимом адресации** называют процедуру нахождения операнда для выполняемой команды. Если команда использует два операнда, то для каждого из них должен быть задан способ адресации, причем **режимы адресации первого и второго операнда могут как совпадать, так и различаться.** Операнды команды могут находиться в разных местах: непосредственно в составе кода команды, в каком-либо регистре, в ячейке памяти; в последнем случае существует несколько возможностей указания его адреса. Строго говоря, способы адресации являются элементом архитектуры процессора, отражая заложенные в нем возможности поиска операндов. С другой стороны, различные способы адресации определенным образом обозначаются в языке ассемблера и в этом смысле являются разделом языка.

**Регистровая адресация.** Операнд (байт или слово) находится в регистре. Этот способ адресации применим ко всем программно-адресуемым регистрам процессора.

**Непосредственная адресация.** Операнд (байт или слово) указывается в команде и после трансляции поступает в код команды; он может иметь любой смысл (число, адрес, код ASCII), а также быть представлен в виде символического обозначения.

Важным применением непосредственной адресации является пересылка относительных адресов (смещений). Чтобы указать, что речь идет об относительном адресе данной ячейки, а не об ее содержимом, используется описатель `onset` (смещение):

**Прямая адресация памяти.** Адресуется память; адрес ячейки памяти (слова или байта) указывается в команде (обычно в символической форме) и поступает в код команды:

## **По организации память м.б. Страничной**

В современных схемах управления памятью не принято размещать процесс в оперативной памяти одним непрерывным блоком.

В самом простом и наиболее распространенном случае страничной организации памяти как логическое адресное пространство, так и физическое представляются состоящими из наборов блоков или страниц одинакового размера. При этом образуются логические страницы, а соответствующие единицы в физической памяти называют физическими страницами или страничными кадрами. Страницы (и страничные кадры) имеют фиксированную длину, обычно являющуюся степенью числа 2, и не могут перекрываться.

## Логический адрес

Номер виртуальной страницы $p$	Смещение внутри виртуальной страницы $d$
--------------------------------	--

### *Таблица страниц*

Атрибуты	Номер физической страницы

Номер физической страницы $p'$	Смещение внутри физической страницы $d$
--------------------------------	---

## Физический адрес

## **Сегментная и сегментно-страничная организация памяти**

**Сегменты, в отличие от страниц, могут иметь переменный размер.** При сегментной организации виртуальный адрес является двумерным и состоит из двух полей – номера сегмента и смещения внутри сегмента.

Логическое адресное пространство – набор сегментов. Каждый сегмент имеет имя, размер и другие параметры (уровень привилегий, разрешенные виды обращений, флаги присутствия). В отличие от страничной схемы, где пользователь задает только один адрес, который разбивается на номер страницы и смещение прозрачным для программиста образом, в сегментной схеме пользователь специфицирует каждый адрес двумя величинами: именем сегмента

## Логический адрес

Номер сегмента  $s$

Смещение внутри сегмента  $d$

*Таблица дескрипторов*

Атрибуты	Адрес начала сегмента

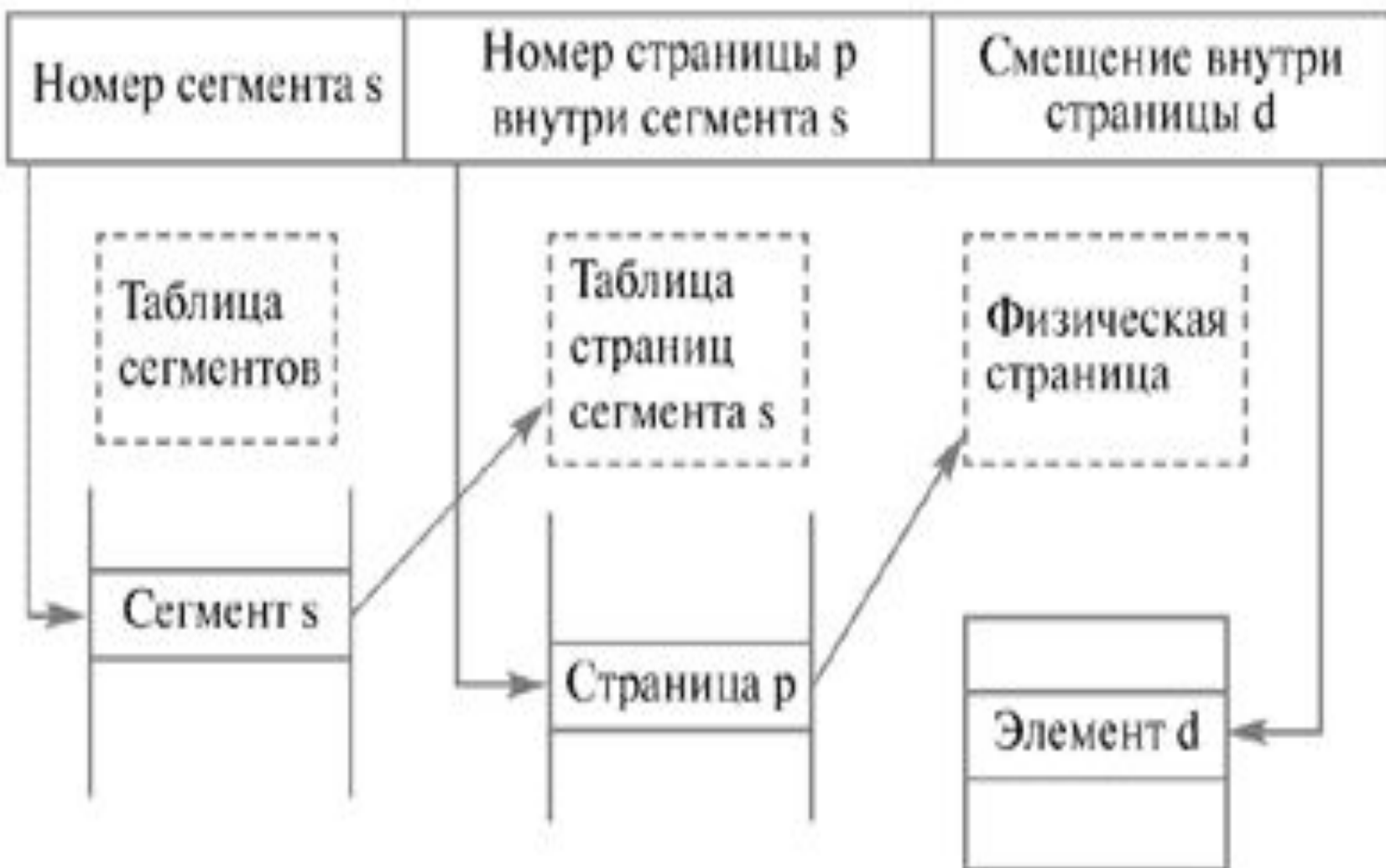
Физический адрес



При **сегментно-страничной организации памяти** происходит двухуровневая трансляция виртуального адреса в физический. В этом случае логический адрес состоит из трех полей: номера сегмента логической памяти, номера страницы внутри сегмента и смещения внутри страницы. Соответственно, используются две таблицы отображения – таблица сегментов, связывающая номер сегмента с таблицей страниц, и отдельная таблица страниц для каждого сегмента.



## Логический адрес



**Стековая память** получила широкое распространение. Но часто работа стековой памяти эмулируется в основной памяти ЭВМ: с помощью программ операционной системы выделяется часть памяти под стек (в IBM PC для этой цели выделяется 64 Кбайта). Специальный регистр микропроцессора (указатель стека) постоянно хранит адрес ячейки ОП, выполняющей функции вершины стека. Чтение числа всегда производится из вершины стека, после чего указатель стека изменяется и указывает на очередную ячейку стековой памяти (т.е. фактически стек остается неподвижным, а перемещается вершина стека). При записи числа в стек сначала номер ячейки в указателе стека модифицируется так, чтобы он указывал на очередную свободную ячейку, после чего производится запись числа по этому адресу. Такая работа указателя стека позволяет реализовать принцип «первым вошел — последним

## **Плоская и многосегментная модели памяти**

Простейшей организацией памяти в защищенном режиме **является плоская модель** памяти: вся память представляется единой линейной последовательностью байт. Это классическая реализация фон-неймановской архитектуры - здесь хранятся и данные, и коды.

Противоположностью плоской модели является **сегментированная защищенная модель памяти** для защищенного режима работы процессора. Распределением памяти ведаёт ОС. Память в данном случае состоит из независимых сегментов. Каждой программе в любой момент предоставляется сегмент кода, сегмент стека и до четырех сегментов данных. Сегменты выбираются селекторами из таблиц, подготовленных ОС.

**Кэш-память** – это сверхбыстрая память используемая процессором, для временного хранения данных, которые наиболее часто используются.

Кэш-память построена на триггерах, которые, в свою очередь, состоят из транзисторов.

Главное преимущество такой памяти – скорость.

Размещена она, на самом кристалле процессора, что значительно уменьшает время доступа к ней.

Главное назначение кэш-памяти – это хранение данных, которые часто используются процессором. Кэш является буфером, в который загружаются данные, и, несмотря на его небольшой объём, (около 4-16 Мбайт) в современных процессорах, он дает значительный прирост производительности в любых приложениях.

Современные процессоры, оснащены кэшем, который состоит, из 2 –ух или 3-ёх уровней

**Кэш первого уровня (L1)** – наиболее быстрый уровень кэш-памяти, который **работает напрямую с ядром процессора**, благодаря этому плотному взаимодействию, данный уровень обладает **наименьшим временем доступа** и работает на частотах близких процессору. **Является буфером между процессором и кэш-памятью второго уровня.**

Intel Core i7-3770К. Данный процессор оснащен 4x32 Кб кэш-памяти первого уровня  $4 \times 32 \text{ КБ} = 128 \text{ КБ}$ . (на каждое ядро по 32 КБ)

Кэш второго уровня (L2) – второй уровень более масштабный, нежели первый, но в результате, обладает меньшими «скоростными характеристиками».

Соответственно, служит буфером между уровнем L1 и L3.

Core i7-3770 K, то здесь объём кэш-памяти L2 составляет  $4 \times 256 \text{ Кб} = 1 \text{ Мб}$ .

Кэш третьего уровня (L3) – третий уровень, опять же, более медленный, нежели два предыдущих. Но всё равно он гораздо быстрее, нежели оперативная память. Объём кэша L3 в i7-3770K составляет 8 Мбайт. Если два предыдущих уровня разделяются на каждое ядро, то данный уровень является общим для всего процессора. Показатель довольно солидный, но не заоблачный. Так как, к примеру, у процессоров Extreme-серии по типу i7-3960X, он равен 15Мб, а у некоторых новых процессоров Xeon, более 20.

**Постоянная память** программируется при изготовлении ПК, во время работы используется только в режиме чтения и хранит программу тестирования ПК при включении питания, а также драйверы управления модулями ПК (BIOS), т.е. обработчики аппаратных и программных прерываний BIOS.

Микросхемы ПЗУ имеют байтовую структуру (емкостью от 16 Кбайт до 256 Кбайт) и разделяются на ПЗУ которые:

- программируются при их производстве (наиболее дешевые);
- программируются специальными устройствами (программаторами);
- перепрограммируемые ПЗУ (ППЗУ) с ультрафиолетовым стиранием, которые программируются программатором, но есть возможность стирать информацию путем облучения микросхем ультрафиолетовыми лучами для повторного программирования.



- Флэш-память - особый вид энергонезависимой перезаписываемой полупроводниковой памяти.
- Энергонезависимая - не требующая дополнительной энергии для хранения данных (энергия требуется только для записи).
- Перезаписываемая - допускающая изменение (перезапись) хранимых в ней данных.
- Полупроводниковая (твердотельная) - не содержащая механически движущихся частей (как обычные жёсткие диски или CD), построенная на основе интегральных микросхем (IC-Chip).

- Видеокарта является, по сути, «компьютером в компьютере», то у неё естественно имеется и своя **видеопамять**, которая является одной из её основных составляющих. Задача видеопамяти– запоминать видеоданные. Видеопамять играет роль некоего кадрового буфера, в который направляются видеоданные, для дальнейшего считывания и обработки их графическим процессором, также здесь хранятся текстуры.
- Если по своему назначению видеопамять напоминает оперативную память, то логично, что и параметры (характеристики) у них будут весьма схожи. **Основными характеристиками здесь будет пропускная способность шины памяти, тип видеопамяти, объём видеопамяти и латентность.**

- 1. Пропускная способность шины памяти:
- Пропускная способность шины памяти определяет количество передаваемых данных в единицу времени. Она определяется разрядностью шины и тактовой частотой работы памяти.
- 2. Тип видеопамяти: В современных видеокартах используется тип памяти GDDR5
- 3. Объём видеопамяти: (128Мб – 1 Гб)
- 4. Латентность:
- Латентность – это время выборки данных из памяти, чем меньше данный параметр, тем лучше, так как не будут наблюдаться значительные задержки при обращении к памяти. У современных видеокарт латентность схем памяти составляет менее 1-2 нс.