

Режимы работы МП

Режимы работы МП

- Начиная с процессора 80286+ МП способны работать в 2х режимах:
- -режим реальных адресов (реальный режим);
- -режим защищенной памяти (защищённый режим).

Режим реальных адресов

- При работе в реальном режиме любой рассматривается процессор 8086 с расширенным набором команд.
- Шина адреса любого процессора в реальном режиме 20 бит, следовательно объем доступной памяти 1 мб (2^{20}).
- В реальном режиме доступа только однозадачная работа, (те МП выполняет 1 программу. Размер сегмента фиксированный 64 КБ.
- После сигнала “сброс”, поступающего на вход Reset#, МП переключается в реальный режим работы.

Защищенный режим

- 1. Многозадачная обработка информации, т.е. можно запустить несколько программ одновременно.
- 2. защита памяти с помощью 4х уровневого механизма привелегий (0 уровень-операционная система, 3 уровень-пользовательские программы)
- 3. доступна вся шина адреса процессора (объем памяти $2^{40}=4\text{ТБ}$)
- 4. Размер сегмента- любой
- память делится на страницы; стандартный объём страницы-4кб

Сегментированная модель памяти

- Если в оперативной памяти есть 2^K ячеек то для ссылки на эти ячейки нужен k – разрядный адрес
- Уникальный адрес байта в памяти называется физическим адресом или абсолютным, с большим объемом памяти, большим будет и разрядность физического адреса, это приводит к увеличению длины команды и программы в целом
- Для того что бы длина команды не зависела от объёма ОП память делят на участки называемыми сегментами
- Размер(объем) сегмента фиксируется
- 2^M $M < K$

- объём сегмента накладывается ограничения, т.е. объём сегмента фиксируется.
- Объём сегмента 64КБ разрядность адреса в сегменте
- $m = \log_2 64К$
- $2^m = 64к$
- $2^m = 2^6 * 2^{10}$
- $m = 16$

- ФА первых ячеек сегмента может быть любым их называют базовыми адресами
- **Базовый адрес – уникальный адрес начала сегмента. Его разрядность совпадает с физическим адресом,**
- В этих условиях ФА ячейки памяти формируется
- $ФА = БА + OFS$
- БА – базовый адрес
- OFS – смещение то есть адрес ячейки в сегменте
- (относительный адрес эффективный адрес ,исполнительный адрес)
- $0 \leq OFS \leq 2^M - 1$
- **При разбиении памяти на сегменты уникальный адрес разделился на 2 части:**
 - **базовый адрес – уникальный адрес 1-й ячейки сегмента, т.е. адрес сегмента;**
- **относительный адрес (смещение) – адрес ячейки внутри сегмента относительно его начала**

- Процессор выставляет на шину адреса ФА а в команде указывают относительный адрес операнда
- ***Относительный адрес имеет меньшую разрядность и хранится в команде вместо физического адреса.***
- ***Базовый адрес получают с помощью сегментных регистров***

Диспетчер памяти

- У программы может существовать 3 типа сегментов:
- **1. сегмент кода (программный сегмент).** Служит для хранения команд программы. За него отвечает сегментный регистр CS;
- **2. сегмент стека служит для хранения точек возврата из подпрограмм и значений некоторых переменных.** Отвечает сегментный регистр SS.
- **3. сегмент данных для хранения значений переменных.** Для возможности обработки больших объёмов данных программа может использовать до 4-х сегментов данных. Отвечают регистры DS, ES, FS, GS.
- **Значение сегментного регистра в реальном режиме работы процессора называют БАЗА, в защищённом режиме работы – СЕЛЕКТОР.**

Диспетчер памяти

- Предназначен для формирования физ. адреса байта памяти. Сегментированная модель памяти представляет собой набор блоков (сегментов)
- Характеризуемых определенными атрибутами;
- размер (объём) сегмента- зависит от режима работы;
- расположение сегмента опр. Базовыми адресами
- тип сегмента;
- -программный сегмент (сегмент кода)
 - сегмент данных
 - сегмент стека
 - характеристика защиты (только защищенный режим)

- МП 80386+ содержит 6 16ти разрядных сегментных регистров. Сегментные регистры хранят селекторы
- по селектору диспетчер памяти определяет базовый адрес сегмента
- Селектор – содержимое сегментного регистра
- CS (code segment)-Сегментный регистр программного сегмента
- DS (data segment)-Сегментный регистр сегмента данных
ES (extended segment)-Сегментный регистр доп. сегмента данных
- SS (stack segment)-Сегментный регистр сегмента, стеки
GS и FS-Сегментные регистры доп сегментов данных

- Селектор в GS обеспечивает обращение к текущему программному сегменту
- Селектор в SS обеспечивает обращение к текущему сегменту стека
- Селектор в DS RS FS GS обеспечивают обращение к текущим сегментам данных

15	0
----	---

CS	
----	--

Сегментный регистр программного сегмента

DS	
----	--

Сегментный регистр сегмента данных

ES	
----	--

Сегментный регистр дополнения, сегмента данных

SS	
----	--

Сегментный регистр сегмента стека

FS	
----	--

GS	
----	--

Сегментные регистры

- При запуске программы управление получает специальная программа, ОС (функция)- загрузчик, который находит свободные сегменты в памяти, и через драйверы устройств обеспечивает копирование программы и ее данных с внешних носителей и загружает селекторы сегментов в соответствующие регистры.
- Селектор сегмента данных загрузчик присваивается глобальной константе-@data
- Первыми командами в программе, программист должен загрузить константу @ data в выбранный им сегментных регистров сегмента данных

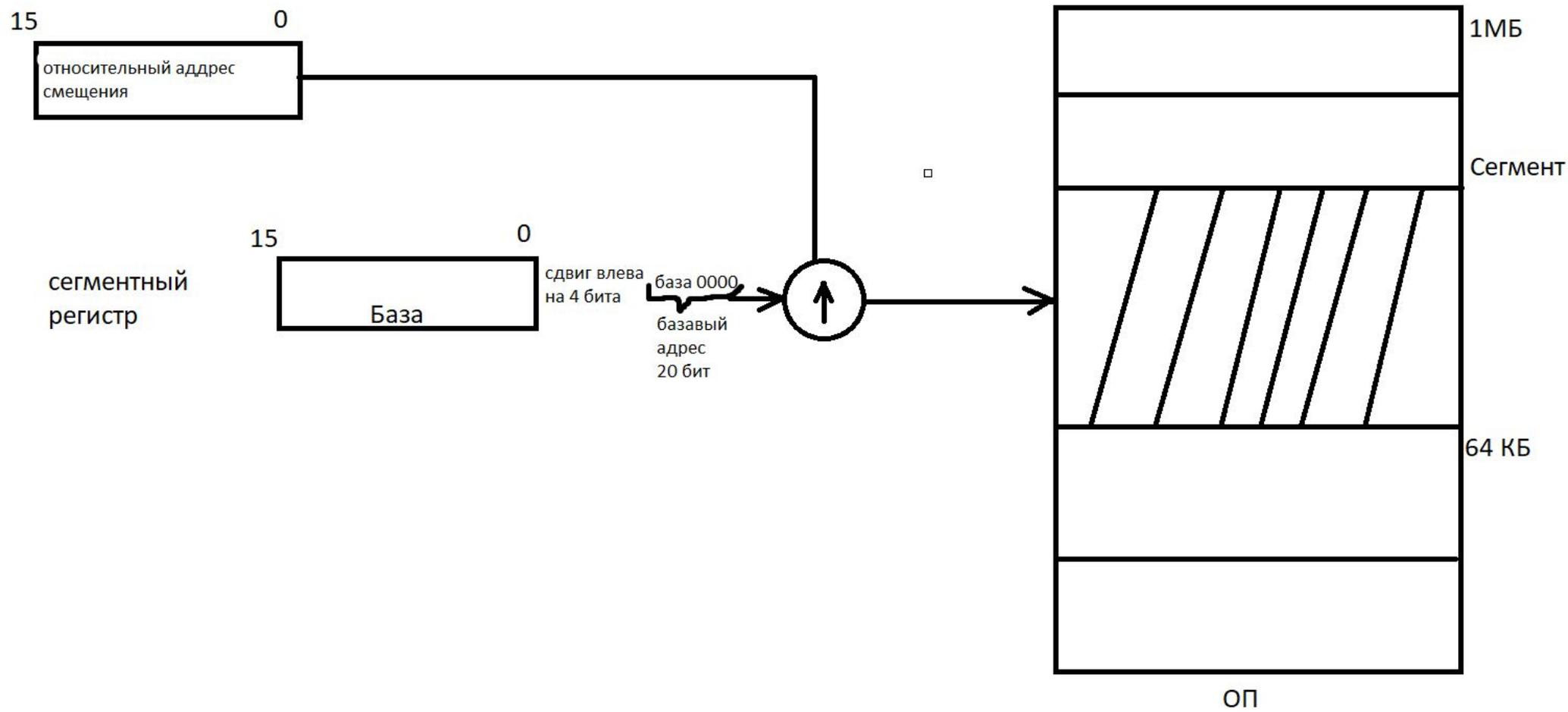
- MOV AX,AS
- MOV AX, ES:AS

- MOV AX, @ data загрузка селектора сегмента данных
- MOV DS,AX

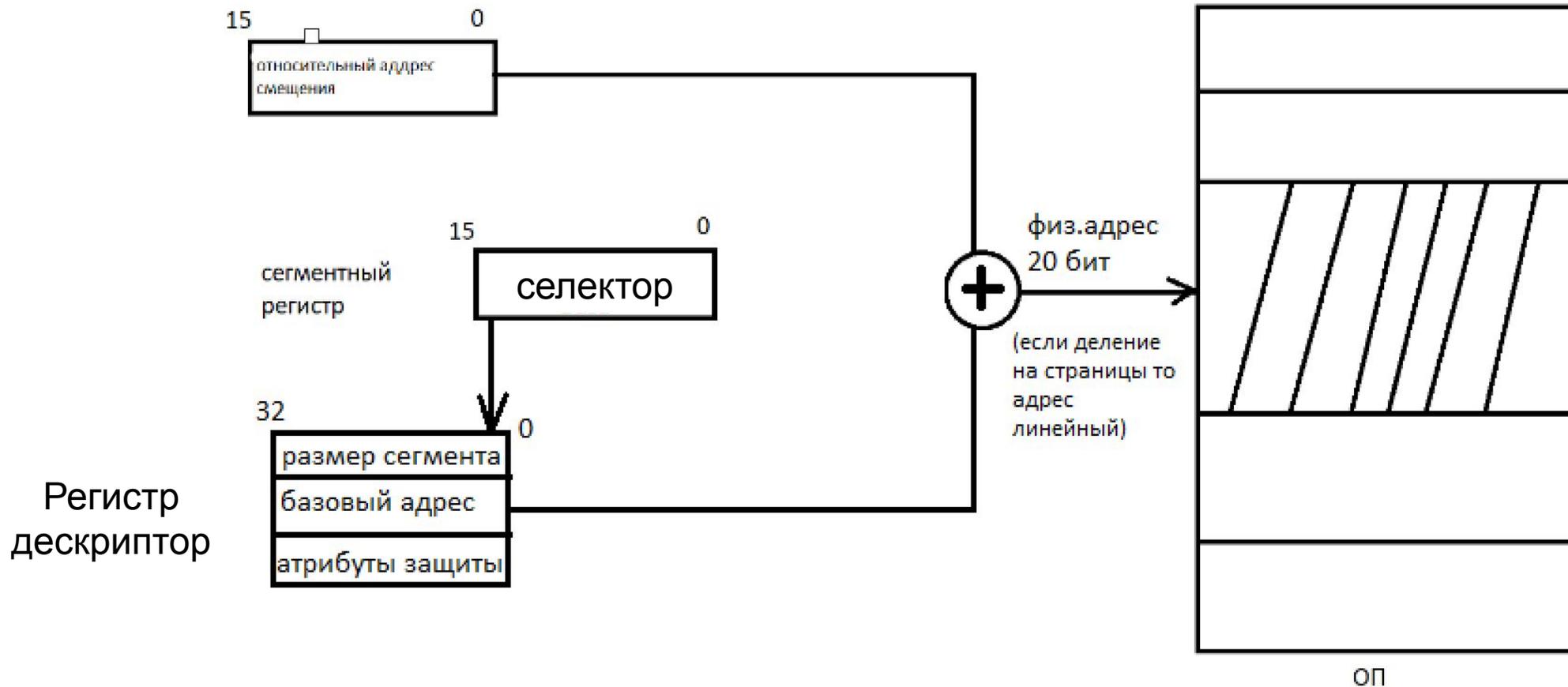
Формирование физического адреса в реальном режиме МП

- $ФА = БА + \text{offset}$

- в реальном режиме селектор база
База-старшие 16 разрядов базового адреса
- База (селектор) сдвигается в лево на 4 бита (умножается на 16). Таким образом формируется 20 Ти разрядный базовый адрес сегмента который складывается с относительным адресом (смещением). Таким образом формируется 20Ти разрядный физический адрес.
-



Формирование физического адреса в защищенном режиме работы



- В защищённом режиме в ОП создается таблица дескрипторов всех программ, которые загружены в ОП
- Дескриптор состоит из 3х частей. разрядность каждой части соответствует разрядности шины адреса МП.
- дескриптор хранит объем сегмента, базовый адрес, атрибут защиты.
- В защищенном режиме с каждым сегментным регистром. связан программнонедоступный регистр дескриптора.

- По селектору находящемуся в определенном дескрипторе в табл. дескрипторов в ОП и загружается в регистр дескриптора этого сегментного регистра. Базовый адрес из регистра дескриптора складывается с относительным адресом, таким образом формируется физический адрес.
- Если память делится на страницы, этот адрес называется линейным.

Выбор сегментного регистра и относительного адреса

Типы регистра	Сегментный регистр	относительный адрес (сменщик)
Программный сегмент (сегмент кода)	CS	счётчик команд IP (16 бит) EIP (32 бит)
Сегмент стека	SS	Указатель стека SP(16 бит) ESP(32 бит)
Сегмент данных	DS ES GS FS	зависит от способа адресации

- Логический адрес – селектор и относительный адрес

- Селектор : смещение

- (относительный адрес)

- 0300h(база) : 0006h

Реальный режим работы

$ФА = БА + ОА = 03000h : 0006h = 030006h$

DS: 0005h

- Логический адрес команды
- CS:IP

Способы адресации операнда

- Операнды могут находиться в
- В РОНЕ
- В команде (в поле непосредственный операнд)
- В памяти (ОП)

Директива размещения и инициализации данных

- Имя DB(data bite) (начальное значение 1, начальное значение 2, нах. значения)
- Эта директива размещает в сегменте данных, данные разрядностью байт
 - AS DB 40
- В ОП выделяется ячейка разрядностью байт, относительный адрес PS, в нее загружается число 40

	128	64	32	16	8	4	2	1	
	7	6	5	4	3	2	1	0	
0000h	0	0	1	0	1	0	0	0	AS

AS DB 30,2,8

128 64 32 16 8 4 2 1
7 6 5 4 3 2 1 0

0002h	0	0	0	0	1	0	0	0	AS + 2
0001h	0	0	0	0	0	0	1	0	AS + 1
0000h	0	0	0	1	1	1	1	0	AS

Команда пересылки

- MOV приемник, источник
- Содержимое источника копируется в приемник

MOV AL BL

MOV AL,BL; (BL) → (AL)

(Ph):=(Bh)

- До выполнения команды
- $(AL) = 20$
- $(BL) = 30$
- После выполнения команды
- $(AL) = 30$
- $(BL) = 30$

Загрузка относительного адреса

- LEA R, M
- MOV R, OFFSET M
- Относительный адрес ячейки памяти M загружается в регистр R
- LEA SI, AS+2; (SI): 0002h

Регистровая адресация

- Операнд находится в регистре (РОНы, сегментные регистры) в команде указывается имя регистра
- `MOV BL,AL; (AL)→(BL)`