

# Типы данных (по размеру в памяти)

Обозначение	Размер, байт	Числа без знака	Числа со знаком
Db	1	0..255	-128..+127
Dw	2	0..65535	-32768..+32767
Dd	4	0..4294967295	-2147483648.. +2147483647
Df	6		
Dq	8		
Dt	10		

# Объявление данных

Имя тип значение

**Mess db 'Primer','\$'**

**X1 db E2h**

**X2 dw A3E7h**

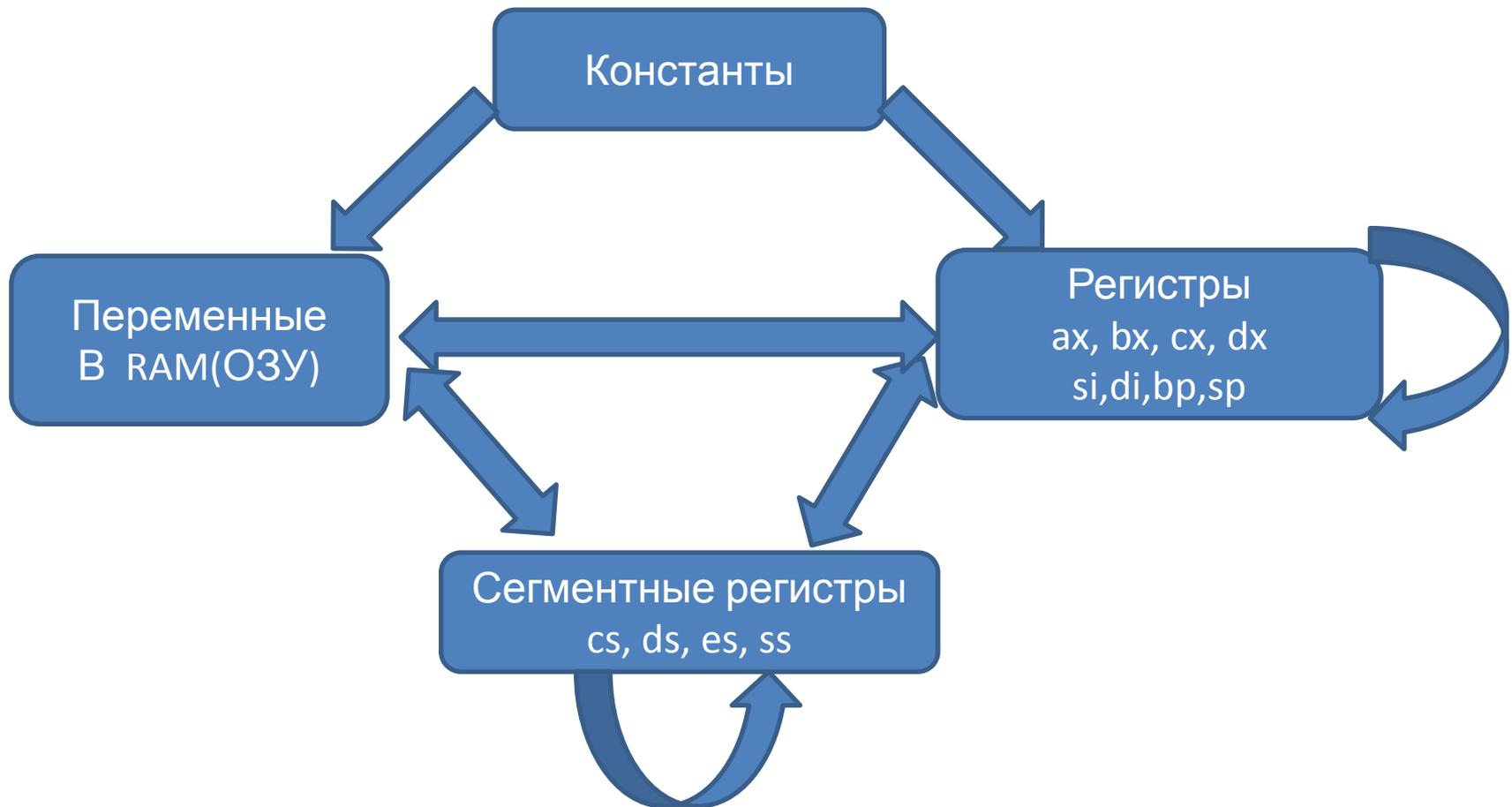
**Mass1 db 10 dup(' ')**

**Mass2 db 15 dup(?)**

**Mass3 db 31 dup(1,2,3,7,9)**

# Команды пересылки данных

## MOV приемник, источник



# Пример mov

```
mov dx,ax
```

```
mov bx,7
```

```
mov ax,09h
```

```
mov cx,0110b
```

```
mov x1,cx
```

# Обмен данными

XCHG оп1, оп2

xchg ax,bx

# Работа со стеком

Запись в стек:                    push операнд

Чтение из стека:                pop операнд

Примеры:

```
push ax  
push bx  
push cx
```

```
pop cx  
pop bx  
pop ax
```

# Исключающее ИЛИ XOR

`xor оп1,оп2`

`xor ax,ax` быстрее, чем `mov ax,0`

`xor ax,bx` быстрее, чем `xchg ax,bx`

# Виды адресации

Регистровая	<code>mov bx,cx</code>
Непосредственная	<code>mov bx,7</code>
Прямая	<code>mov bx,es:0002</code>
Косвенная	<code>mov bx,[cx]</code>
По базе со сдвигом	<code>mov bx,[cx+4]</code>

# Модели памяти

Tiny – код, данные и стек размещаются в одном сегменте. Для сом-программ.

Small – код в одном сегменте, а данные и стек – в другом сегменте(общем)

Flat – как и tiny, но использует 32-битные сегментные регистры (для 32-разрядных операционных систем) – адресуемое пространство до 4 Гб.

# Директивы набора допустимых команд

.8086 - только команды для процессора 8086

.286 – для 80286

.386 – для 80386

.486 – для 80486

.586 – для Pentium

.686 – для Pentium 2