

Лекция №10**Тема "Арифметические команды. Умножение и деление"****Команды умножения MUL, IMUL**

умножение целых без знака

`mul операнд`

умножение целых со знаком

`imul операнд`**Умножение байтов:**`AX := AL * op` (op: r8, m8)**Умножение слов:**`(DX, AX) := AX * op` (op: r16, m16)**Примечание:**

1. Место нахождения первого операнда и результата *фиксировано* и поэтому явно не указывается.
2. При умножении под результат отводится в два раза больше места, чем под сомножители. Это связано с тем, что умножение n -значных чисел в общем виде дает произведение из $2n$ цифр.
3. Вопрос о том, помещается ли результат умножения в формате сомножителей или нет, решается с помощью анализа флага переноса CF и переполнения OF, которые в обеих командах умножения меняются синхронно и по следующему правилу:

CF = OF = 1 – произведение занимает двойной формат

CF = OF = 0 – произведению достаточен формат сомножителей

Примеры

```

N      DB      10
. . .
mov    al, 2
mul    n          ; ax = 2 * 10 = 20 = 0014h
                    ah = 00h  al = 14h

mov    al, 26
mul    n          ; ax = 26 * 10 = 260 = 0104h
                    ah = 01h  al = 04h

mov    ax, 8
mov    bx, -1
imul  bx          ; (dx, ax) = -8 = 0FFFFFFF8h
                    dx = 0FFFFh  ax = 0FFF8h

```

Команды деления DIV, IDIV

деление целых без знака

```
div операнд
```

деление целых со знаком

```
idiv операнд
```

Деление слова на байт:

$AH := AX \bmod op$; $AL := AX \operatorname{div} op$ (op: r8, m8)

Деление двойного слова на слово:

$DX := (DX, AX) \bmod op$; $AX := (DX, AX) \operatorname{div} op$ (op: r16, m16)

Примечание:

1. Место нахождения первого операнда (делимого) и результата *фиксировано* и поэтому явно не указывается. Указывается только второй операнд (делитель).

2. При делении возникает ошибка “деление на 0 или переполнение”:

- делитель равен 0 (op = 0);
- неполное частное не вмещается в отведенное ему место (регистр AL или AX); это произойдет при делении 600 на 2.

При такой ошибке происходит прекращение выполнения программы.

Изменение размера числа

Расширение байта до слова

cbw

Местонахождение операнда и результата фиксировано:

AL - операнд;

AX - результат

Команда записывает в регистр AH число 00h или 0FFh в зависимости от знака числа из регистра AL:

$$AH := \begin{cases} 00h, & AL \geq 0 \\ 0FFh, & AL < 0 \end{cases}$$

Флаги эта команда не меняет.

Пример

```
mov  al, 32          ; al = 20h
cbw                          ; ax = 0020h

mov  al, -32         ; al = 0E0h
cbw                          ; ax = 0FFE0h
```

Расширение слова до двойного слова

cwd

Местонахождение операнда и результата фиксировано:

AX - операнд;

DX, AX - результат

Задачи

1. К числу из регистра ВХ надо прибавить число из регистра АL:
 $BX := BX + AL$

Числа без знака:

```
mov  ah, 0           ; al → ax (расширение без знака)
add  bx, ax
```

Числа со знаком:

```
cbw  ax             ; al → ax (расширение со знаком)
add  bx, ax
```

2. Можно делить слово на байт, но не байт на байт. Если необходимо разделить байт на байт, то прежде всего надо первый байт (делимое) расширить до слова:

 $AL := AL \text{ div } CH$
Числа без знака:

```
mov  ah, 0           ; al → ax (расширение без знака)
div  ch              ; al := ax div ch    ah := ax mod ch
```

Числа со знаком:

```
cbw  ax             ; al → ax (расширение без знака)
idiv ch             ; al := ax div ch    ah := ax mod ch
```