5. Команды умножения и деления целых чисел

Умножение знаковых и беззнаковых чисел выполняется разными командами: MUL для беззнаковых и IMUL для знаковых чисел. Обе команды являются однооперандными, что приводит к необходимости использования для второго сомножителя строго фиксированных регистров.

Правила использования команд умножения:

- при умножении двух однобайтовых чисел одно задается как операнд команды (регистр или именованная область памяти), а второй обязательно находится в регистре AL; при этом результат всегда помещается в регистр AX (интересно, что такой подход никогда не приводит к выходу результата за разрядную сетку!)
- при умножении двух чисел длиной в слово одно задается как операнд команды, второе всегда в регистре AX, а результат рассматривается как двойное слово, размещаемое в двух регистрах: младшая часть в регистре AX, а старшая часть – в регистре DX
- нельзя использовать непосредственный операнд, его предварительно приходится размещать в памяти или в регистре

Пример:

```
N10      DB   10          ; небольшое однобайтовое число
N1000    DW  1000         ; двухбайтовое число
--------------------
MOV   AL, 2     ; (AL) = 2
MUL   N10       ; (AX) = 20 = 0014₁₆ , причем (AH) = 00₁₆, (AL) = 14₁₆
MOV   AL, 26    ; (AL) = 26
MUL   N10       ; (AX) = 260 = 0104₁₆ , причем (AH) = 01₁₆, (AL) = 04₁₆
MOV   AX, 8     ; (AX) = 8
MOV BX, -1      ; (BX) = -1
IMUL   BX       ; (AX)*(BX) = -8 = FFFF FFF8₁₆ , причем (DX) = FFFF₁₆
                  ;  (AX) = FFF8₁₆
```

Аналогично используются команды деления: для беззнаковых чисел – команда DIV, для знаковых - команда IDIV. Делить можно слово на байт или двойное слово на слово. В первом случае делимое всегда берется из регистра AX, делитель (байтовое число!) задается как операнд команды (регистр или область памяти), а результат размещается в двух регистрах: AH содержит остаток от деления, а AL - целую часть. Условно это можно записать так:

(AX) / операнд = $(AH)_{\text{остаток}}$ и $(AL)_{\text{целая часть}}$

Во втором случае делимое размещается в паре регистров DX и AX, делитель (слово) – в операнде команды, а результат – в регистре AX (целая часть) и в DX (остаток):

(DX, AX) / операнд = $(DX)_{\text{остаток}}$ и $(AX)_{\text{целая часть}}$

Если при умножении никакие ошибки возникнуть не могут, то при делении возможно появление ошибок двух типов:

- деление на ноль (операнд = 0)
- переполнение частного, когда целая часть результата не помещается в формат делителя, например:

```
MOV     AX, 600     ; (AX) = 600
MOV     BH, 2       ; (BH) = 2
DIV     BH          ; 600/2 = 300, что не размещается в регистре AL
```

В обоих случаях процессор просто аварийно прекращает выполнение программы.

Вся арифметика основана на использовании согласованных операндов, но иногда надо работать с операндами **разных** размеров. Для этого более короткий операнд надо расширить до более длинного: байт – до слова, слово – до двойного слова. Для беззнаковых чисел это выполняется просто - в старший байт добавляются нули:

```
MOV     AL, 2       ; (AL) = 02₁₆
MOV     AH, 0       ; (AH) = 00, (AL) = 02, а вместе (AX) = 0002₁₆
```

Сложнее – для знаковых чисел, т.к. приходится анализировать знак: если число больше 0, то добавляются нули, если число меньше 0, то добавляются единицы. Другими словами, на старший байт дублируется знаковый бит числа. Для упрощения этих операций можно использовать специальную команду с мнемоникой CBW (Convert Byte to Word). Ее операнды фиксированы: исходный байт всегда в AL, а результат – всегда в AX. Аналогично, для преобразования слова в двойное слово есть команда CWD (Convert Word to Double). Ее операнды тоже фиксированы: исходное слово всегда в AX, результат всегда в (DX, AX).

Пример: поделить байт на байт, что сразу сделать невозможно. Сначала делимое надо расширить до слова:

```
MOV AL, 10      ; (AL) = 10 и делить нельзя
CBW             ; AL→AX, (AX) = 10 и делить можно
IDIV  BL        ; (AX) / (BL)
```

Практические задания к теме №5.

Задание 1. Программа выполняет умножение двух операндов, один из которых отрицателен. Операнд, указываемый в команде - это один из сомножителей; он может находиться в регистре или в памяти, но не может быть непосредственным операндом. Местонахождение другого сомножителя фиксировано и поэтому в команде не указывается явно: при умножении байтов он берется из регистра AL, а при умножении слова из регистра AX. Фрагмент программы:

```
N  db   4
   mov   AL, 2
   mul    N          ; (AX) = 2*4 = 8 = 0008h,  (AH) = 00h, (AL) = 08h
   mov    BX, -1
   imul    BX ; (DX, AX) =-8 = 0FFFFFFF8h, (DX)=0FFFFh, AX=00FFF8h
```

Задания 2-11. Написать и выполнить в пошаговом режиме программы для вычисления следующих выражений:

1. (A+B)*C, результат записать на место операнда B.
2. A-B+C, результат записать в операнд A
3. (A+B)-(C+D)результат на место D
4. A*B*C, результат на место B
5. (A+B)*(C+D), результат на место A
6. A+B*C-D,результат сохранить в регистре H
7. (A+B)/C, результат на место A
8. (A-B)/D*(-C), результат на место B
9. (A+B-D)*C/(-A+B), результат в регистре H
10. (A*B-C)/(-D)*(-A)*B, результат в регистре H