

Arytmetyka

Dzisiaj zaczniemy od dodawania i odejmowania, potem przejdziemy do dzielenia i mnożenia. Zaczniemy od utworzenia sobie labeli ze zmiennymi a, b oraz c. Wynik przechowamy sobie w y. Dzisiejsze zadania będziemy realizować na tych trzech „zmiennych”.

```
a dw 0x05
b dw 0x3B
c dw 0x4C
y dw 0x0
```

Przypomnę, że przypisanie do rejestru wartości w ten sposób:

```
MOV AX, a
```

Ustawi nam wartość AX na adres labela a, a tego byśmy nie chcieli. Polecam sobie sprawdzić w insighcie. W tym wypadku interesuje nas ustawienie AX na wartość, więc

```
MOV AX, [a]
```

Zrobimy sobie proste zadanko z dodawaniem na początek.

1. Napisz program który obliczy wzór: $a+b+c$

Przypominam instrukcję ADD

```
ADD AX, 5h ; ax=ax+5h
```

Więc całe zadanie polegać będzie na tym, że do AX musimy przypisać wartość a, do np. CX wartość b, dodać, następnie co CX zapisać wartość c, dodać do AX, przenieść wynik do y i zaliczone.

```

org 100h

mov AX, [a]      ; | AX = 5h
mov CX, [b]      ; | CX = 3Bh
add AX, CX       ; | AX = AX + CX = 40h
mov CX, [c]      ; | CX = 4Ch
add AX, CX       ; | AX = AX + CX = 8Ch
mov [y], AX

mov AX, 4C00h
int 21h

a dw 0x05
b dw 0x3B
c dw 0x4C
y dw 0x0

```

Zadanko prościutkie i przyjemniutkie. Trochę gorzej dzieje się, gdy zajmiemy się mnożeniem i dzieleniem. Instrukcje MUL oraz DIV są trochę dziwne i nieintuicyjne, głównie przez to, że w przeciwieństwie do instrukcji ADD i SUB, gdzie oba operandy podajemy my, tutaj podajemy tylko jeden

MUL CX

Trochę nieintuicyjne. Jest tak dlatego, że w instrukcjach MUL i DIV pierwszy operand jest domyślny i jest nim rejestr AX. Gdyby składnia była jak w dodawaniu, polecenie wyglądałoby tak:

MUL AX, (jakaś liczba)

Tylko AX jest zahardkodowane i zawsze domyślne.

```

org 100h

mov AX, 5h      ; | AX = 5h
mov CX, 5h     ; | CX = 5h
mul CX         ; | AX = AX*CX => AX = 5h*5h = 19h

mov AX, 4C00h
int 21h

```

Dzielenie działa analogicznie, tylko mnemonik oczywiście inny, DIV.

Szybkie zadanie

Spróbujcie wykonać jakieś mnożenie, które da wynik większy niż 16 bitów i zobaczcie, co się stanie z wynikiem. Głównie interesować nas będą rejestry AX i DX.

IMUL i IDIV, czyli MUL i DIV na sterydach

IMUL służy generalnie do mnożenia, ale jego działanie jest zależne od liczby parametrów, oraz w przeciwieństwie do MUL pozwala na mnożenie liczb ze znakiem. Gdy dostanie jeden parametr, to działa jak zwykły MUL. Dla dwóch parametów, działa jak MUL ze składnią add

```
IMUL AX, 09h ; | AX=AX*9h
```

I tu śmieszna sprawa, bo idiv już na taką operację nie pozwala, ale nadal działa na liczbach ze znakiem.

Zadanka

1. Napisz program który obliczy wzór: $a + b + c$
2. Napisz program obliczający wzór: $a * b + c$
3. Napisz program obliczający wzór: $2a + 2b - 2c$
4. Napisz program obliczający wzór: $\frac{a}{b} + c$
5. Napisz program obliczający wzór: $\frac{a}{b+c}$
6. Napisz program obliczający wzór: $a\frac{b}{c}$
7. Napisz program obliczający wzór: $a^2 + 2b + c$
8. Napisz program obliczający wzór: $2a\frac{b}{2c}$

9. Napisz program obliczający wzór: $a \frac{b+a}{c}$
10. Napisz program obliczający wzór: $\frac{a}{2} + \frac{b}{3} + \frac{c}{4}$