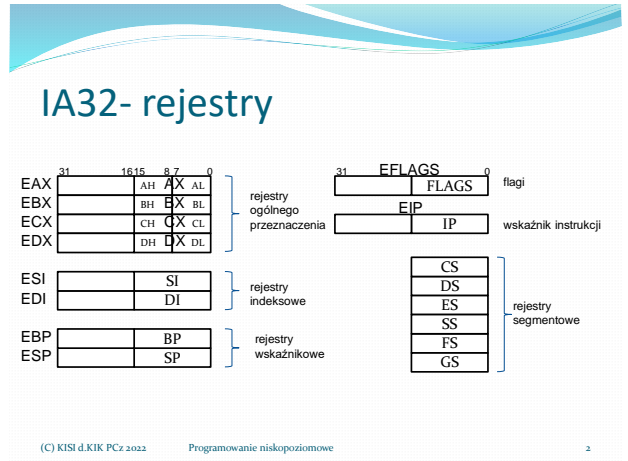


Instrukcje arytmetyczne



Rejestr flag

bit	Skróć/wartość	Opis	Typ
0	CF	flaga przeniesienia (carry)	S
2	PF	flaga parzystości (parity)	S
4	AF	flaga wyrównania (adjust)	S
6	ZF	flaga zera (zero)	S
7	SF	flaga znaku (sign)	S
10	DF	flaga kierunku (direction)	C
11	OF	flaga przepełnienia (overflow)	S

S: Znacznik stanu
C: Znacznik kontrolny
X: Znacznik systemowy

Instrukcje arytmetyczne

- ADD dodawanie całkowitoliczbowe
- ADC dodawanie z przeniesieniem
- ADCX dodawanie z przeniesieniem bez znaku
- ADOX dodawanie z przeniesieniem bez znaku
- SUB odejmowanie
- SBB odejmowanie z pożyczką
- MUL mnożenie bez znaku
- MULX mnożenie bez znaku
- IMUL mnożenie ze znakiem
- DIV dzielenie bez znaku
- IDIV dzielenie ze znakiem
- INC inkrementacja (zwiększenie)
- DEC dekrementacja (zmniejszenie)
- NEG zmiana znaku
- CMP porównanie

Wpływa na flagi: OSZAPC

Instrukcja ADD

```
add cel, źródło
```

Dodaje zawartość źródła i celu, sumę umieszcza w miejscu przeznaczenia (cel).

cel := cel + źródło

```
add eax, zmienna
add edx, [ebx+esi*4]
add rcx, rbx
```

Uwaga:
Jeśli źródło jest adresowane w trybie prostym może mieć do 32 bitów.

Przykład – oblicz sumę krawędzi prostopadłościanu

```
mov eax, a            ;wczytaj a
add eax, b            ;dodaj b
add eax, c            ;dodaj c
add eax, eax          ; *2
add eax, eax          ; *2
```

Wpływa na flagi: OSZAPC

Instrukcja ADC

```
adc cel, źródło
```

Dodaje zawartość źródła, celu i przeniesienia, sumę umieszcza w miejscu przeznaczenia (cel).

$$\text{cel} := \text{cel} + \text{źródło} + \text{CF}$$

```
adc eax, zmienna
```

```
adc edx, [ebx+esi*4]
```

```
add rcx, rbx
```

Uwaga:

Jeśli źródło jest adresowane w trybie prostym może mieć do 32 bitów.

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

7

Przykład – oblicz sumę liczb całkowitych 128-bitowych

```
mov esi, offset b ;adres zmiennej b
mov edi, offset a ;adres zmiennej a
mov eax, [esi] ;najmłodsze podwójne słowo b
add [edi], eax ;dodajemy do najmłodszego a
mov eax, [esi+4] ;drugie podwójne słowo b
adc [edi+4], eax ;do drugiego a plus przeniesienie
mov eax, [esi+8] ;trzecie podwójne słowo b
adc [edi+8], eax ;do trzeciego a plus przeniesienie
mov eax, [esi+12] ;czwarte podwójne słowo b
adc [edi+12], eax ;do czwartego a plus przeniesienie
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

8

Przykład – oblicz sumę liczb całkowitych 128-bitowych

```
mov rsi, offset b ;adres zmiennej b
mov rdi, offset a ;adres zmiennej a
mov rax, [rsi] ;najmłodsze poczwórne słowo b
add [rdi], rax ;dodajemy do najmłodszego a
mov rax, [rsi+8] ;drugie poczwórne słowo b
adc [rdi+8], rax ;do drugiego a plus przeniesienie
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

9

Wpływa na flagi: -----C
Wymagane: ADX

Instrukcja ADCX

```
adcx cel, źródło
```

Dodaje bez znaku zawartość źródła, celu i przeniesienia, sumę umieszcza w miejscu przeznaczenia (cel – rejestr 32|64 bitowy).

$$\text{cel} := \text{cel} + \text{źródło} + \text{CF}$$

```
adcx eax, zmienna
```

```
adcx edx, [ebx+esi*4]
```

```
adcx rcx, rbx
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

10

Wpływa na flagi: O-----
Wymagane: ADX

Instrukcja ADOX

```
adox cel, źródło
```

Dodaje bez znaku zawartość źródła, celu i flagi przepełnienia, sumę umieszcza w miejscu przeznaczenia (cel – rejestr 32|64 bitowy).

$$\text{cel} := \text{cel} + \text{źródło} + \text{OF}$$

```
adox eax, zmienna
```

```
adox edx, [ebx+esi*4]
```

```
adox rcx, rbx
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

11

Wpływa na flagi: OSZAPC

Instrukcja SUB

```
sub cel, źródło
```

Odejmuje zawartość źródła od celu, różnicę umieszcza w miejscu przeznaczenia (cel).

$$\text{cel} := \text{cel} - \text{źródło}$$

```
sub ecx, zmienna
```

```
sub ebx, [ebx+esi*4]
```

```
sub rax, rdx
```

Uwaga:

Jeśli źródło jest adresowane w trybie prostym może mieć do 32 bitów.

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

12

Wpływa na flagi: OSZAPC

Instrukcja SBB

sbb cel, źródło

Odejmuje zawartość źródła od celu z uwzględnieniem pożyczki, różnicę umieszcza w miejscu przeznaczenia (cel).

cel := cel - (źródło + CF)

sbb edx, zmienna

sbb eax, [ebx+esi*4]

sbb rax, rdx

Uwaga:

Jeśli źródło jest adresowane w trybie prostym może mieć do 32 bitów.

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

13

Wpływa na flagi: OxxxxC

Instrukcja MUL

mul źródło

Mnoży bez znaku zawartość akumulatora (al, ax, eax, rax) i źródła, wynik umieszcza w miejscu przeznaczenia (ax, dx:ax, edx:eax, rdx:rax). Flagi CF i OF są zerem, jeśli starsza połowa bitów wyniku jest równa zero.

wynik := acc * źródło

mul zmienna

mul word ptr[ebx + esi*4]

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

14

Przykład – oblicz pole prostokąta o bokach a, b

```
mov  eax, a           ;a
mul  b               ;a * b
je   poza_int        ;jeśli przekroczony zakres
mov  pole, eax       ;zapisz
...
...
poza_int: ...
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

15

Wpływa na flagi: OxxxxC
Wymaga BMI2

Instrukcja MULX

mulx cel1, cel2, źródło

Mnoży bez znaku zawartość rejestru edx | rdx i źródła, wynik umieszcza w rejestrach celu(cel1:cel2).

cel1:cel2 := e(r)dx * źródło

mulx ecx, ebx, [tab + esi*4]

mulx rcx, rbx, rax

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

16

Wpływa na flagi: OxxxxC

Instrukcja IMUL -1

imul źródło

Mnoży ze znakiem zawartość akumulatora (al, ax, eax, rax) i źródła, wynik umieszcza w miejscu przeznaczenia (ax, dx:ax, edx:eax, rdx:rax). Flagi CF i OF są zerem, jeśli iloczyn mieści się w młodszej połowie bitów wyniku.

wynik := acc * źródło

imul zmienna

imul ecx

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

17

Wpływa na flagi: OxxxxC

Instrukcja IMUL -2

imul cel, źródło

Mnoży ze znakiem zawartość rejestru celu (16|32|64 bity) przez źródło, wynik umieszcza w miejscu przeznaczenia (cel). Flagi CF i OF są zerem, jeśli iloczyn mieści się w rejestrze celu.

cel := cel * źródło

imul eax, zmienna

imul rdx, rcx

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

18

Wpływa na flagi: OxxxxC

Instrukcja IMUL -3

```
imul cel, źródło1, źródło2
```

Mnoży ze znakiem zawartość źródła1 (16|32|64 bity) przez źródło2 (stała), wynik umieszcza w miejscu przeznaczenia (cel). Flagi CF i OF są zerem, jeśli iloczyn mieści się w rejestrze celu.

cel := źródło1 * źródło2

```
imul eax, zmienna, 5
```

```
imul cx, dx, 77
```

Uwaga:

Jeśli źródło2 jest adresowane w trybie prostym może mieć do 32 bitów.

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

19

Przykład – oblicz sumę krawędzi prostopadłościanu

```
mov  eax, a           ;wczytaj a
add  eax, b           ;dodaj b
add  eax, c           ;dodaj c
imul eax, eax, 4      ;×4
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

20

Wpływa na flagi: xxxxxx

Instrukcja DIV

```
div źródło
```

Dzieli bez znaku zawartość AX, DX:AX, EDX:EAX, RDX:RAX przez źródło, iloraz umieszcza w AL, AX, EAX, RAX a resztę w AH, DX, EDX, RDX.

```
div byte ptr zmienna
```

```
div ebx
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

21

Wpływa na flagi: xxxxxx

Instrukcja IDIV

```
idiv źródło
```

Dzieli ze znakiem zawartość AX, DX:AX, EDX:EAX, RDX:RAX przez źródło, iloraz umieszcza w AL, AX, EAX, RAX a resztę w AH, DX, EDX, RDX.

```
idiv byte ptr zmienna
```

```
idiv ebx
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

22

Wpływa na flagi: OSZAP

Instrukcja INC

```
inc cel
```

Zwiększa zawartość celu o 1.

```
inc zmienna
```

```
inc edx
```

```
inc rcx
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

23

Wpływa na flagi: OSZAP

Instrukcja DEC

```
dec cel
```

Zmniejsza zawartość celu o 1.

```
dec zmienna
```

```
dec edx
```

```
dec r8
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

24

Wpływa na flagi: OSZAPC

Instrukcja NEG

```
neg cel
```

Zmienia znak celu w kodzie U₂.

```
cel := -cel
```

```
neg ax
```

```
neg byte ptr[ebx+esi*4]
```

```
neg r11
```

Flaga CF=0, tylko dla argumentu=0.

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

25

Wpływa na flagi: OSZAPC

Instrukcja CMP

```
cmp źródło1, źródło2
```

Porównuje zawartość źródła1 i źródła2, wynik nie jest zapamiętywany, tylko są ustawiane flagi.

```
źródło1-źródło2
```

```
cmp ax, zmienna
```

```
cmp edx, [ebx+esi*4]
```

```
cmp rcx, 123
```

Uwaga:

Jeśli źródło2 jest adresowane w trybie prostym może mieć do 32 bitów.

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

26

Przykład – oblicz sumę kwadratów liczb w tablicy

```
mov eax, 0 ;wartość początkowa sumy
```

```
mov esi, eax ;indeks
```

```
mov ebx, offset wektor ;tablica liczb całkowitych
```

```
mov ecx, 123 ;licznik
```

```
petla: mov edx, [ebx+esi*4]
```

```
imul edx, edx ;kwadrat liczby
```

```
add eax, edx ;suma
```

```
inc esi
```

```
dec ecx
```

```
jnz petla ;wynik w eax
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

27

Instrukcje arytmetyczne BCD

- DAA korekta upakowanego kodu BCD po dodawaniu
Decimal adjust after addition
- DAS korekta upakowanego kodu BCD po odejmowaniu
Decimal adjust after subtraction
- AAA ASCII korekta po dodawaniu
ASCII adjust after addition
- AAS ASCII korekta po odejmowaniu
ASCII adjust after subtraction
- AAM ASCII korekta po mnożeniu
ASCII adjust after multiplication
- AAD ASCII korekta przed dzieleniem
ASCII adjust before division

!!! Nie działają w trybie 64 bitowym !!!

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

28

Wpływa na flagi: OSZAPC

Instrukcja DAA

```
daa
```

Korekta upakowanego kodu BCD (w AL) po dodawaniu. Polega na dodaniu 6 najpierw do młodszego półbajta, a potem do starszego, jeśli ich zawartości były większe od 9 lub wystąpiło przeniesienie AF (CF)

```
daa
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

29

Wpływa na flagi: OSZAPC

Instrukcja DAS

```
das
```

Korekta upakowanego kodu BCD (w AL) po odejmowaniu. Polega na odjęciu 6 najpierw od młodszego półbajta, a potem od starszego, jeśli ich zawartości były większe od 9 lub wystąpiło przeniesienie AF (CF).

```
das
```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

30

Wpływa na flagi: OSZAPC

Instrukcja AAA

aaa

Korekta nieupakowanego kodu BCD (w AL) po dodawaniu. Polega na dodaniu 6 do młodszego półbajta i 1 do AH, jeśli zawartość AL była większa od 9 lub wystąpiło przeniesienie AF.

aaa

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

31

Wpływa na flagi: OSZAPC

Instrukcja AAS

aas

Korekta nieupakowanego kodu BCD (w AL) po odejmowaniu. Polega na odjęciu 6 od młodszego półbajta i 1 od AH, jeśli zawartość AL była większa od 9 lub wystąpiło przeniesienie AF.

aas

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

32

Wpływa na flagi: OSZAPC

Instrukcja AAM

aam

Korekta nieupakowanego kodu BCD (w AX) po mnożeniu. Polega na jednoczesnym wykonaniu:

$$AH = AL \text{ div } 10$$

$$AL := AL \text{ Mod } 10.$$

aam

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

33

Wpływa na flagi: OSZAPC

Instrukcja AAD

aad

Korekta nieupakowanego kodu BCD (w AX) przed dzieleniem. Polega na jednoczesnym wykonaniu:

$$AL := AH * 10 + AL.$$

$$AH = 0$$

aad

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

34

Przykład – dodawanie liczb BCD

```

mov  al, 0
add  al, al      ;CF=0
petla: mov  al, [esi] ;pobierz cyfrę źródła
      adc  al, ds:[edi] ;dodaj cyfrę celu z przeniesieniem
      aaa                ;korekta
      mov  ds:[edi], al ;zapamiętaj cyfrę
      inc  esi           ;następna cyfra
      inc  edi
      dec  ecx
      jnz  petla        ;CF nie zmieniło się od AAA!!!

```

(C) KISI d.KIK PCz 2022

Programowanie niskopoziomowe

35