Arhitectura sistemelor de calcul

Instrucțiuni aritmetice

ADC

```
adc EDX, EBX ; EDX ← EDX + EBX + CF
adc AX, [var]
adc [var], AX
adc EAX, 123456h
adc BYTE [var], 10
```

Restricții: Cei doi operanzi ai adunării trebuie să aibă același tip. Cel mult un operand poate fi o locație de memorie.

SBB

```
sbb EDX, EBX ; EDX ← EDX - EBX - CF
sbb AX, [var]
sbb [var], AX
sbb EAX, 123456h
sbb BYTE [var], 10
```

Restricții: Cei doi operanzi ai scăderii trebuie să aibă același tip. Cel mult un operand poate fi o locație de memorie.

IMUL

imul DH ; AX ← AL * DH

imul DX ; DX:AX ← AX * DX

imul EBX ; EDX:EAX ← EAX * EBX

Restricții:

Restricții: Lungimea operației de înmulțire se păstrează pe o lungime dublă față de lungimea operanzilor.

Instrucțiunea IMUL efectuează operația de înmulțire pentru întregi cu semn.

Se impune ca primul operand și rezultatul să se păstreze în registri.

Operandul explicit poate fi un registru sau o variabilă, dar nu poate fi o valoare imediată (constantă)

IDIV

idiv CL ; AL ← AX/CL , AH ← AX%CL

idiv SI ; AX ← DX:AX / SI , DX ← DX:AX % SI

idiv EBX ; EAX ← EDX:EAX / EBX , EDX ← EDX:EAX % EBX

idiv DWORD[var] ; EAX ← EDX:EAX / DWORD[var] , EDX ← EDX:EAX % DWORD[var]

Restricții Instrucțiunea IDIV efectuează operația de împărțire pentru întregi cu semn.
Se impune ca primul operand și rezultatul să se păstreze în registri.
Primul operand nu se specifică și are o lungime dublă față de al doilea operand
Operandul explicit poate fi un registru sau o variabilă dar nu poate fi o valoare imediată (constantă).
Prin împărțirea unui număr mare la un număr mic, există posibilitatea ca rezultatul să depășească capacitatea de reprezentare. În acest caz, se va declanșa aceeași eroare ca și la împărțirea cu 0.

Instrucțiuni de conversie cu semn

CBW (AL→AX)

cbw ; dacă AL = 01110111b atunci AX ← 0000000001110111b
; dacă AL = 11110111b atunci AX ← 1111111111110111b

Restricții: convertește cu semn BYTE-ul din AL în WORD-ul AX.

conversia se referă la extinderea reprezentării prin completarea cu bitul de semn.

CWD (AX → DX:AX)

cwd

Restricții: convertește cu semn WORD-ul din AX în DWORD-ul DX:AX.

conversia se referă la extinderea reprezentării prin completarea cu bitul de semn.

CWDE

cwde

Restricții: convertește cu semn WORD-ul din AL în DWORD-ul EAX.

conversia se referă la extinderea reprezentării prin completarea cu bitul de semn.

CDQ

cdq

Restricții: convertește cu semn DWORD-ul din EAX în QWORD-ul EDX:EAX.

conversia se referă la extinderea reprezentării prin completarea cu bitul de semn.

Conversia fără semn se face prin „zerorizare”:

```
mov AH, 0 ; AL -> AX

mov DX, 0 ; AX -> DX:AX

mov EDX, 0 ; EAX -> EAx:EDX
```

Little endian representation

Each byte has an address. X86 processors store data using little endian order (the byte representing the „end” of the number will be stored at the „little”-est address.

For example:

```
a db 12h

b dw 3456h

c dd 7890abcdh

d dq 1122334455667788h
```