Příznaky, aritmetika

ISU cv 4

http://www.fit.vutbr.cz/~isakin/isu

Úkol na zahřátí

- cv4_1.asm

```
; Vyčti dvě čísla (x, y) z paměti, proveď sečtení a odečtení, ulož výsledky do paměti (r1, r2). Můžeš je vypsat i na stdout

section .data
    ; unsigned char x = 98; y = 15;
section .bss
    ; unsigned char r1, r2;
section .text
main:
    ; unsigned char r1 = x + y;
    ; unsigned char r2 = x - y;
    ret
```

Příznaky

Příznaky [FLAGS register]

- Speciální bity, které nás informují o určitém stavu
- Obvykle výsledek aritmetických operací
- Ovlivňují některé výpočty
- Umožňují provádět podmíněné skoky

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	NT	IOPL		OF	DF	IF	TF	SF	ZF	0	AF	0	PF	1	CF

Příznaky [FLAGS register]

- OF - Overflow - přetečení (znaménková aritmetika)
- CF - Carry - přenos   (bezznaménková aritmetika)
- SF - Sign - záporný výsledek
- ZF - Zero - nulový výsledek

IntelCodeTable.pdf

Ukázky

- cv4_A.asm
- cv4_B.asm

ADD, ADC, INC, SUB, SBB, DEC, NEG

+ -

- **ADD** op1, op2 *op1 = op1 + op2*
- **SUB** op1, op2 *op1 = op1 - op2*

ADD a **SUB** nastavují a mažou **CF** a **OF**.

- **ADC** op1, op2 *op1 = op1 + op2 + CF* (with carry)
- **SBB** op1, op2 *op1 = op1 - op2 - CF* (with borrow)

Pokud předchozí instrukce **ne**nastaví **CF**, chovají se ADC a SBB stejně jako ADD a SUB

++ --

- **INC** op *op = op + 1*
- **DEC** op *op = op - 1*

INC a **DEC** **ne**ovlivňují nastavování **CF**.

- **NEG** op *op = 0 - op* (negace ve dvojkovém doplňku)

if op == 0 then CF=0 else CF=1

Úkol

- cv4_2.asm

Sečti čísla x a y s použitím jediného 8-bitového registru, výsledek do r.

```
section .data
    x   dd  0x4FE8AAFF
    y   dd  0x10203040
    r    dd  0
```

MUL, IMUL, DIV, IDIV

* Celočíselné násobení *

- **MUL** op (unsigned)
- **IMUL** op (signed)

Při násobení dochází ke zvětšení datového typu (rozsahu), registry jsou dané:

AL x [8-bit] = AH:AL ⇒ AX

AX x [16-bit] = DX:AX

EAX x [32-bit] = EDX:EAX

```
MUL  ebx      ; EDX:EAX = EAX * EBX
MUL  byte 10  ; nelze násobit konstantou
```

/ Celočíselné dělení /

- **DIV** op (unsigned)
- **IDIV** op (signed)

				kvocient		zbytek
AX	/	[8-bit]	=	AL	|	AH
DX:AX	/	[16-bit]	=	AX	|	DX
EDX:EAX	/	[32-bit]	=	EAX	|	EDX

```
DIV  ebx    ; EAX = EDX:EAX / EBX,    EDX = EDX:EAX % EBX
```

Ukázka

- cv4_C.asm
- cv4_D.asm

Úkol

- cv4_3.asm

; Vypočítej obsah lichoběžníku a vypiš výsledek.

```
section .data
    a   dw  10
    c   dw  20
    v   dw  5
```

MOVZX, MOVSX, CBW, CWD, CDQ

MOVy (přenosové instrukce)

op1 > op2

- **MOVZX** op1, op2 (move with zero-extend)

```
EDX = 10 DF 95 05        EDX = 10 DF 95 85
EAX = FA 35 11 FF        EAX = FA 35 11 FF
    MOVZX AX,DL              MOVZX EAX,DL
EDX = 10 DF 95 05        EDX = 10 DF 95 85
EAX = FA 35 00 05        EAX = 00 00 00 85
```

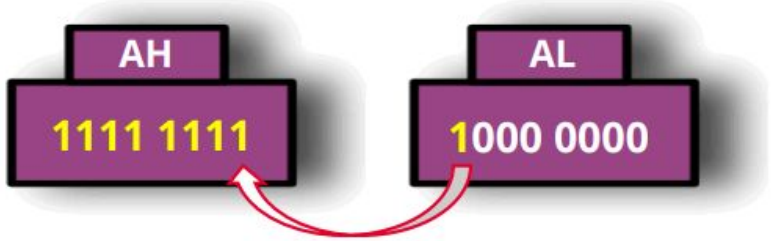
- **MOVSX** op1, op2 (move with sign-extension)

```
EDX = 10 DF 95 05        EDX = 10 DF 95 85
EAX = FA 35 11 FF        EAX = FA 35 11 FF
    MOVSX AX,DL              MOVSX EAX,DL
EDX = 10 DF 95 05        EDX = 10 DF 95 85
EAX = FA 35 00 05        EAX = FF FF FF 85
```

Converty (přenosové aritmetické instrukce)

Rozšíření (pouze) znaménkového čísla

- **CBW** ; **C**onvert **B**yte to **W**ord AL ⟶ AH:AL
- **CWD** ; **C**onvert **W**ord to **D**ouble AX ⟶ DX:AX
- **CDQ** ; **C**onvert **D**ouble to **Q**uadword EAX ⟶ EDX:EAX



Úkoly

- cv4_4.asm

Vyúčtování v restauraci:
Pivo 30,- 8x
Víno 50,- 3x
Kofola 25,- 4x
Ceny inicializuj do pole doublů, Počty inicializuj do pole wordů.
Vypočítej celkovou sumu a výsledek ulož do EAX.

- cv4_5.asm

*Vypočítej výraz y = ((a*2) - b + (c/4)) % 3*
kde a, b a c jsou ukazatele na hodnoty o velikosti 16 bitů.