



FAKULTA ústav inteligentních
INFORMAČNÍCH systémů
TECHNOLOGIÍ

Programování na strojové úrovni

Cvičení 5: Logické instrukce, posuvy, rotace

Tomáš Goldmann, igoldmann@fit.vutbr.cz
www.fit.vutbr.cz/~igoldmann/courses

Rev.: 1A/2017

Rozcvička 😊



Spočítejte kolik **0** je v registru EAX.



```
MOV ECX, ____
```

```
CYCLE: _____
```

```
INC EBX
```

```
NOT_INC:
```

```
_____ CYCLE
```

Rozcvička ☺ - řešení



Spočítejte kolik je **0** v registru EAX.



```
MOV ECX, 32
                                MOV EBX, 0
CYCLE:                          SHL EAX, 1
                                JC NOT_INC
                                INC EBX
NOT_INC:
                                LOOP CYCLE
```

Logické instrukce – součet, součin, bitová inverze



OR operand1 (cíl), operand2 (zdroj)

- Provede logický bitový součet (neprovádějí se přenosy do vyšších řádů)

AND operand1 (cíl), operand2 (zdroj)

$$\begin{array}{r} * \begin{array}{|c|c|c|c|} \hline 1 & 1 & 0 & 1 \\ \hline 1 & 1 & 0 & 1 \\ \hline \end{array} \\ \hline \begin{array}{|c|c|c|c|} \hline 1 & 1 & 0 & 1 \\ \hline \end{array} \end{array}$$

- Provede logický součin

NOT operand1

- Provede inverzi jednotlivých bitů (jedničkový komplement)

$$\begin{array}{r} \begin{array}{|c|c|c|c|} \hline 1 & 1 & 0 & 1 \\ \hline \end{array} \\ \text{NOT} \hline \begin{array}{|c|c|c|c|} \hline 0 & 0 & 1 & 0 \\ \hline \end{array} \end{array}$$

Logické instrukce – exkluzivní OR



XOR operand1 (cíl), operand2 (zdroj)

- Operace exkluzivní OR

$$\begin{array}{r} \text{XOR} \begin{array}{|c|c|c|c|} \hline 1 & 1 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \\ \hline \end{array} \\ \hline \begin{array}{|c|c|c|c|} \hline 0 & 1 & 0 & 1 \\ \hline \end{array} \end{array}$$

Instrukce pro posuvy– SHL, SHR



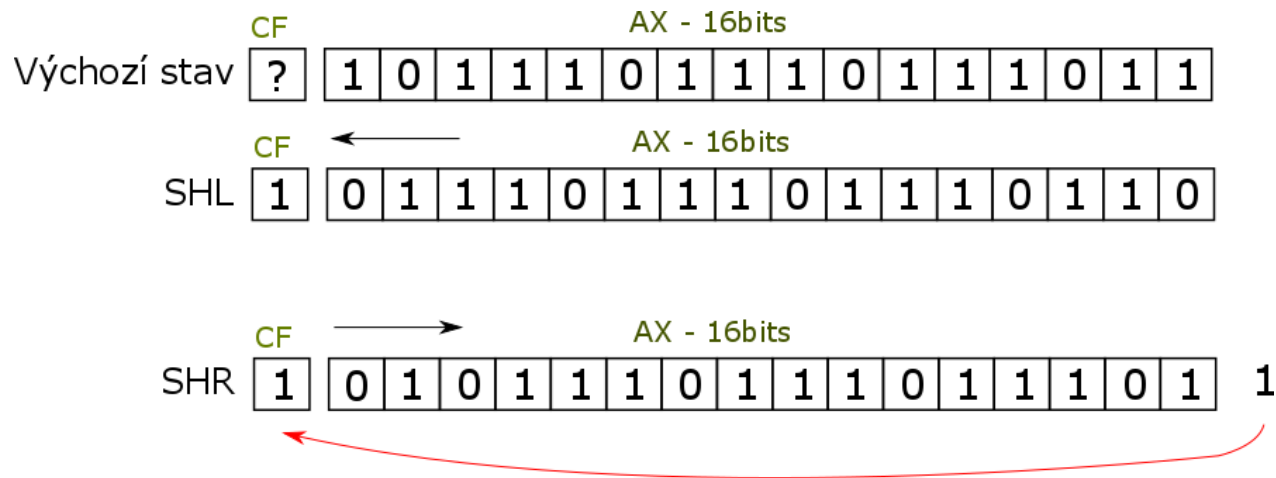
Posuvy slouží ke změně pozic bitů se zachováním bitové posloupnosti. Bity určené prvním operandem jsou pomocí instrukcí SHL, SHR, SAL, SAR posunuty o n bitů (určených druhým operandem doleva nebo doprava).

SHL operand1 (cíl), operand2 (zdroj)

Provádí posun v rámci paměťového místa/registru doleva

SHR operand1 (cíl), operand2 (zdroj)

Provádí posun v rámci paměťového místa/registru doprava



Instrukce pro posuvy– SAL, SAR



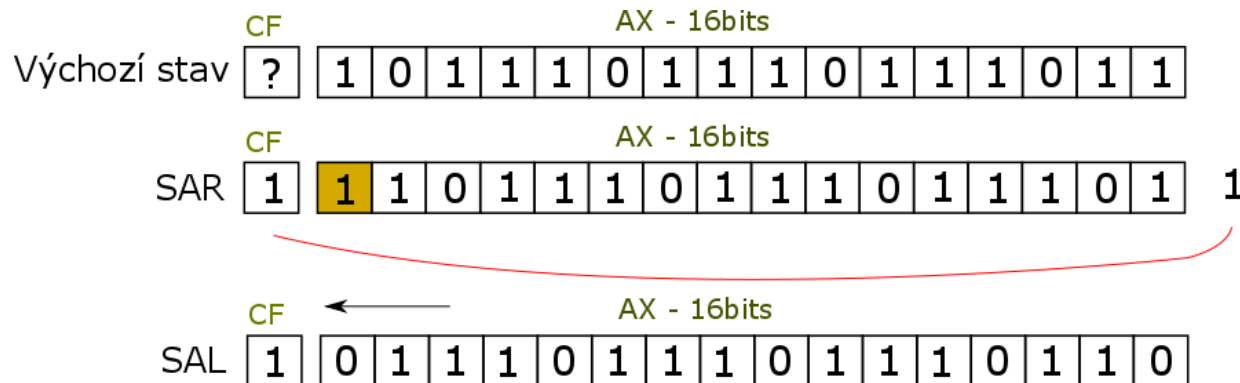
Tyto instrukce slouží pro aritmetické posuny (**posuny se zachováním znaménka**)

SAL operand1 (cíl), operand2 (zdroj)

Provádí **aritmetický** posun v rámci paměťového místa/registru doleva

SAR operand1 (cíl), operand2 (zdroj)

Provádí **aritmetický** posun v rámci paměťového místa/registru doprava



Instrukce pro rotace– ROL, ROR



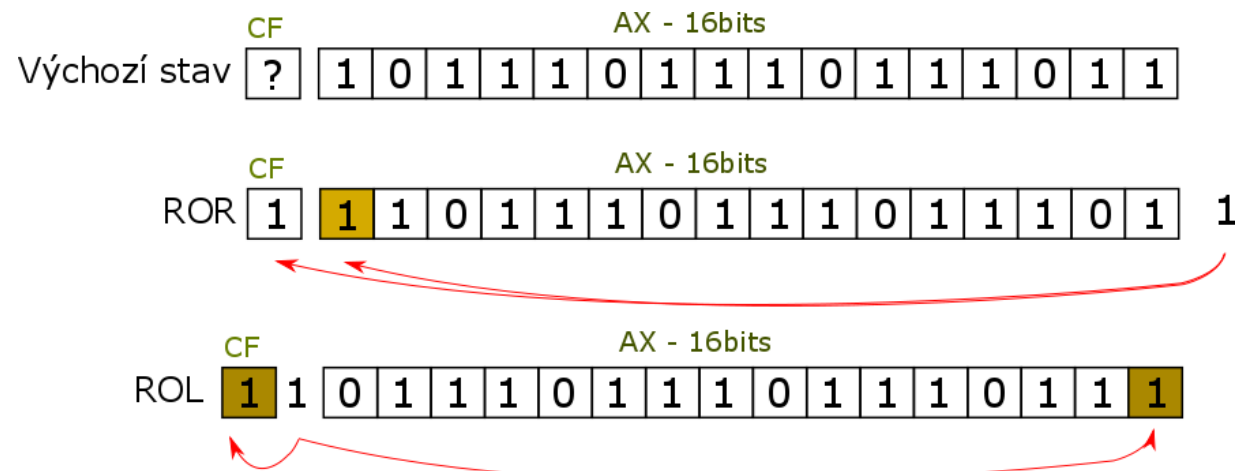
Rotace na rozdíl od posuvů provádějí nedestruktivní změnu pořadí bitů. Při rotacích nedochází ke ztrátě informace, jelikož se hodnoty pixelů pohybují cyklicky v rámci místa určeného operandem 1.

ROL operand1 (cíl), operand2 (zdroj)

Provádí **rotaci** v rámci paměťového místa/registru doleva

ROR operand1 (cíl), operand2 (zdroj)

Provádí **rotaci** v rámci paměťového místa/registru doprava



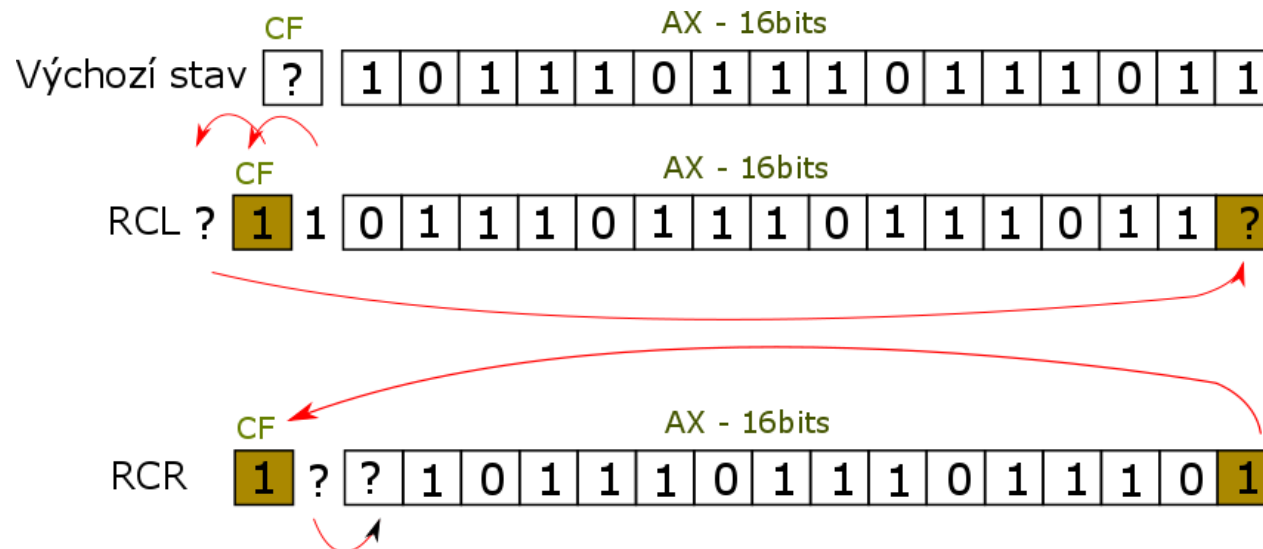
Instrukce pro rotace– RCL, RCR

RCL operand1 (cíl), operand2 (zdroj)

Provádí **rotaci přes CF** v rámci paměťového místa/registru doleva

RCR operand1 (cíl), operand2 (zdroj)

Provádí **rotaci přes CF** posun v rámci paměťového místa/registru doprava



Podmíněné a nepodmíněné skoky



Nepodmíněný skok

Jedná se o typ skoku, který se vždy provede. Nezávisí na výsledku předchozí instrukce. Instrukce JMP. Tato instrukce má jeden operand, který určuje na které návěští se bude skákat.

JMP <návěští>

Podmíněný skok

Jedná se o typ skoku, který se provádí na základě příznaků v EFLAGS. Instrukce podmíněného skoku obvykle začínají na J__

J__ <návěští>

1. Do registru EBX uložte hodnotu 0xA1FF. Aplikujte masku 0x00FF na registr EBX. Uložte horní polovinu registru EBX do registru AX a pomocí instrukce `call WriteUInt16` vypište hodnotu.
2. **Zaměňte** horní polovinu registru EBX s dolní polovinou registru EAX.
3. Vyzkoušejte násobení a dělení celočíselných datových typů pomocí posuvu.
4. Vypočítejte příklad $(-16/2 + -5*3)$, výsledek převedte na kladné číslo pomocí logických instrukcí (nápodvěda: vzpomeňte si, jak se převádí číslo do doplňkového kódu).
5. Vypočítejte průměrnou hodnotu pole (pole db 10,20,30,15,16,17,18,19,20,21,18)