

Logika

ISU cv 5

www.fit.vut.cz/person/isakin/public/isu

Části převzaty od Husa (cv04) a Goldmann (cv05)



Logické operace

Logické operace

- Jsou rychlejší než aritmetické
- Pracují s jednotlivými bity
- Všechny nastavují nějaké příznaky!
- Operandy instrukcí musí mít stejnou velikost
- Operátory:

AND, OR, XOR, TEST, NOT, SHL, SHR, ROL, ROR, RCL, RCR

Ukázka/úkol

- cv5_1.asm - NEG vs NOT

AND, TEST, OR, XOR

A	B	AND	OR	XOR
0	0			
0	1			
1	0			
1	1			

AND == TEST, OR, XOR

A	B	AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

AND, TEST

- `AND op1, op2` *op1 = op1 AND op2*
 - Výsledek uložen v op1, nastavuje příznaky
- `TEST op1, op2` *op1 AND op2*
 - Neukládá výsledek, nastavuje příznaky

OR, XOR

- **OR** $op1, op2$ $op1 = op1 \text{ OR } op2$
 - Výsledek uložen v $op1$, nastavuje příznaky
- **XOR** $op1, op2$ $op1 = op1 \text{ XOR } op2$
 - Výsledek uložen v $op1$, nastavuje příznaky
 - Rychlé nulování registru (! příznaky)
 - Kryptografie - (de)šifrování dat

Ukázka

- cv5_A.asm - TEST, AND, OR, XOR

Úkol

- cv5_2.asm - šifrování



Posuvy

Posuvy

- Bitové posuvy (SHL, SHR, SAL, SAR) pohybují bity uvnitř registru
 - Instrukce co posunout, o kolik bitů
- Násobení / dělení dvěma
- Posuny ztrácím informaci

Posuvy

- Instrukce `co posunout`, o kolik bitů

- Při posunu **doleva** se prázdné bity v obou směrech doplňují **nulami**.

SHL – shift left.

```
mov al, 15 ;AL = 00001111 = 15
shl al, 1 ;AL = 00011110 = 30
mov al, -16 ;AL = 11110000 = -16
shl al, 1 ;AL = 11100000 = -32
```

SAL – shift arithmetic left.

```
mov al, 15 ;AL = 00001111 = 15
sal al, 1 ;AL = 00011110 = 30
mov al, -16 ;AL = 11110000 = -16
sal al, 1 ;AL = 11100000 = -32
```

- Při posunu **doprava** se prázdné bity doplňují **nulami** nebo **prvním bitem**.

SHR – shift right.

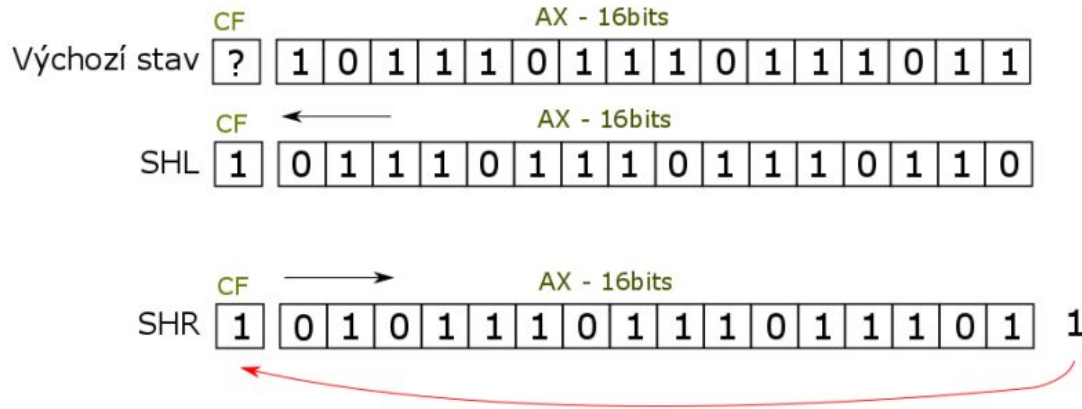
```
mov al, 15 ;AL = 00001111 = 15
shr al, 1 ;AL = 00000111 = 7
mov al, -16 ;AL = 11110000 = -16
shr al, 1 ;AL = 01111000 = 120
```

SAR – shift arithmetic right.

```
mov al, 15 ;AL = 00001111 = 15
sar al, 1 ;AL = 00000111 = 7
mov al, -16 ;AL = 11110000 = -16
sar al, 1 ;AL = 11111000 = -8
```

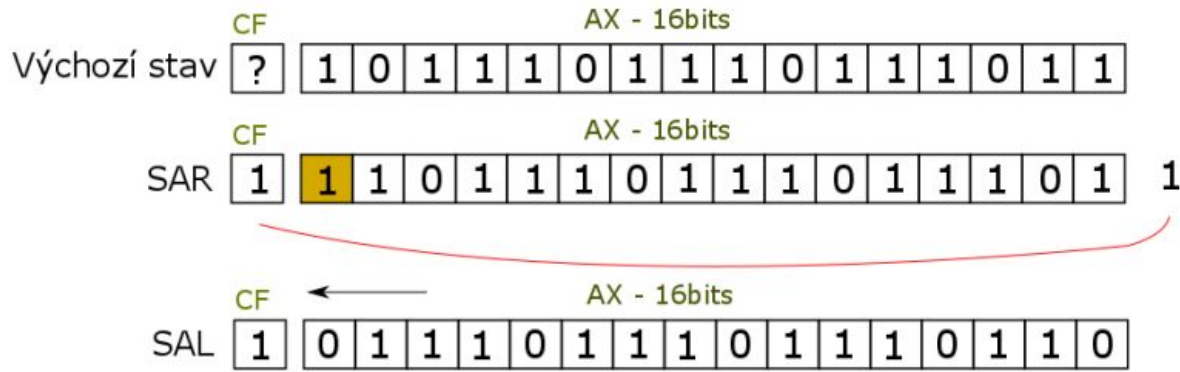
Posuvy SHL, SHR

- **SHL** - posun v rámci paměťového místa / registru **doleva**
- **SHR** - posun v rámci paměťového místa / registru **doprava**



Posuvy SAL, SAR

- SAL - aritmetický posun v rámci paměťového místa / registru **doleva**
- SAR - aritmetický posun v rámci paměťového místa / registru **doprava**
- Zachovávají znaménko



Ukázka

- cv5_B.asm

Úkol

- cv5_3.asm



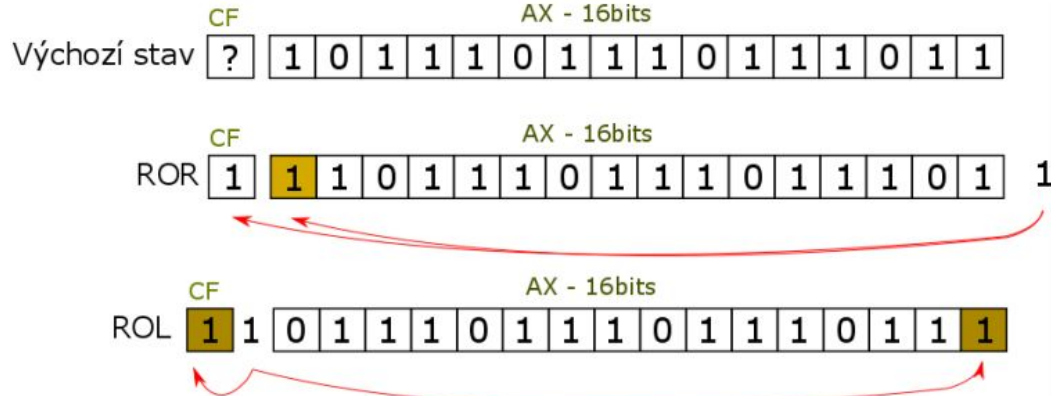
Rotace

Rotace

- Bitové rotace (**ROL**, **ROR**, **RCL**, **RCR**) pohybují bity uvnitř registru
 - `Instrukce` `co` `posunout`, `o` `kolik` `bitů`
- Bity, které opouští registr, se do něj vrací z druhé strany
(přímo nebo přes příznak **CF**)
- Nedestruktivní změnu pořadí bitů ⇒ Zachování informace
- Spojení výsledku násobení (DX:AX)

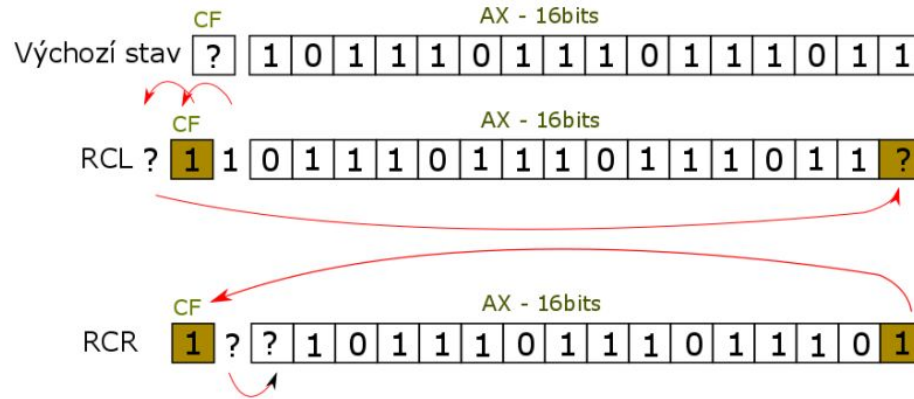
Rotace ROL, ROR

- **ROL** - rotace v rámci paměťového místa / registru **doleva**
- **ROR** - rotace v rámci paměťového místa / registru **doprava**



Rotace RCL, RCR

- **RCL** - rotace přes **CF** v rámci paměťového místa / registru **doleva**
- **RCR** - rotace přes **CF** v rámci paměťového místa / registru **doprava**



Rotace

- Instrukce co posunout, o kolik bitů

ROL – rotate left.

```
mov al, 102 ;AL= 01100110, CF= ?
rol al, 1 ;AL= 11001100, CF= 0
rol al, 1 ;AL= 10011001, CF= 1
rol al, 1 ;AL= 00110011, CF= 1
```

ROR – rotate right.

```
mov al, 102 ;AL= 01100110, CF= ?
ror al, 1 ;AL= 00110011, CF= 0
ror al, 1 ;AL= 10011001, CF= 1
ror al, 1 ;AL= 11001100, CF= 1
```

RCL – rotate through carry left.

```
mov al, 102 ;AL= 01100110, CF= ?
rcl al, 1 ;AL= 1100110?, CF= 0
rcl al, 1 ;AL= 100110?0, CF= 1
rcl al, 1 ;AL= 00110?01, CF= 1
```

RCR – rotate through carry right.

```
mov al, 102 ;AL= 01100110, CF= ?
rcr al, 1 ;AL= ?0110011, CF= 0
rcr al, 1 ;AL= 0?011001, CF= 1
rcr al, 1 ;AL= 10?01100, CF= 1
```

Ukázka

- cv5_C.asm

Úkol

- cv5_4.asm



!! Příští týden TEST 1 !!



Skoky

(jen základy)

Ne/podmíněné skoky

- Nepodmíněný skok
 - `jmp label`
 - Prove se vždy
- Podmíněný skok
 - `j_ label`
 - Skočí na návěští pouze při splnění podmínky
 - Splněné podmínky == dle nastavení příznaků

Ukázka

- cv5_D.asm

Úkoly

- cv5_5.asm - šifrování
- cv5_6.asm