

# Instrukcije za pomeranje i rotiranje binarnog sadržaja

## SHL, SHR

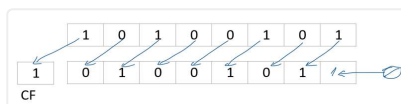
sintaksa: SHL dst, imd8 ili SHL dst, CL

Funkcionisanje: pomeranje binarnog sadržaja prvog operanda za broj mesta zadatih u drugom operandu ulevo (SHL - *SHift Left*) ili udesno (SHR - *SHift Right*).

Pomeranje za jedno mesto ulevo, na primeru:

```
MOV AL, 0A5h
```

```
SHL AL, 1
```



AL = 4Ah, CF=1

Svi bitovi se "pomeraju" za jedno mesto ulevo. Bit najveće težine koji pri pomeranju "ispadne" se prihvata u CF. Na "upražnjeno" mesto, u bit najmanje težine, se uvodi vrednost 0.

Šematski će se ova funkcionalnost uprošćeno prikazivati:



Pomeranje za jedno mesto udesno:

Svi bitovi se "pomeraju" za jedno mesto udesno. Bit najmanje težine koji pri pomeranju "ispadne" se prihvata u CF. Na "upražnjeno" mesto, u bit najveće težine, se uvodi vrednost 0.

Šematski:



Za prethodni primer:

```
MOV AL, 0A5h
```

```
SHR AL, 1
```

Rezultat:

AL=52h, CF=1

Pomeranje ulevo i udesno ima i aritmetičku interpretaciju: pomeranje ulevo za jedno mesto ima isti efekat kao množenje sa dva, pomeranje udesno kao celobrojno deljenje sa dva.

Iz prethodnih primera:  $A5h \cdot 2 = 14Ah$ ;  $A5:2 = 52h$  i ostatak 1

Kako je deljenje instrukcijom DIV skupa operacija, ukoliko treba podeliti poznatim deliocem vrednosti 2, efikasnije je koristiti SHR od DIV (i zbog brzine i manje potrebnih registara).

Ovo je jedna od retkih optimizacija na kojima se insistira u okviru ovog kursa.

Pomeranje za više mesta:

Drugi operand može da bude neposredni operand (konstanta), ili registar CL.

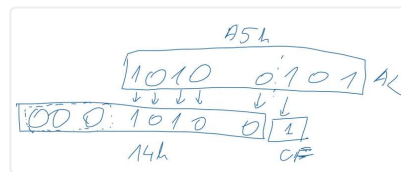
Ekvivalentno po efektima višestrukom pomeranju za po jedno mesto.

$SHL\ AL,\ 2 \Leftrightarrow SHL\ AL,\ 1;\ SHL\ AL,\ 1$

Primer:

`MOV AL, 0A5h`

`SHR AL, 3`



Samo poslednji od tri bita najmanje težine se čuva u CF, preostala dva su izgubljena; na tri mesta najveće težine se uvode nule.

Pomeranje ulevo i udesno za više mesta ima aritmetički ekvivalent u množenju i deljenju stepenom 2, gde je stepen jednak broju mesta za koliko se pomera.

Uz napomenu da se prekoračenje pri takvom množenju ne detektuje, a da se gube svi osim jednog bita ostatka pri deljenju.

## SAL, SAR

SHR se može koristiti kao efikasnija zamena za deljenje stepenom 2 samo za neoznačene podatke. Za označene bi "uvođenje" 0 sa najviše težine menjalo znak negativnom broju.

Stoga je uvedena i instrukcija SAR (*Shift Arithmetical Right*):



Bit najveće težine se zadržava nakon pomeranja udesno. Sada će negativni brojevi da zadržavaju znak.

SAL ima isti efekat kao SHL (tzv. sinonim sa SHL), uvedena je radi simetrije.

Primer:

```
mov ax, -128 ; EAX = ????FF80h
shl eax, 16 ; EAX = FF800000h
sar eax, 16 ; EAX = FFFFFFFF80h
```

## ROL, ROR, RCL, RCR

Rotacija binarnog sadržaja: *ROtate Left*, *ROtate Right*, *Rotate through Carry Left*, *Rotate through Carry Right*. Sintaksa je ista kao kod ostalih instrukcija iz ove grupe.

Funkcionisanje:



Binarni sadržaj se rotira ulevo/udesno, uz kopiranje bita najveće/najmanje težine i u CF.

Primer:

Ukoliko je pre operacije bilo AL=A5h,

- ROL AL,1 bi dalo: AL = 4Bh, CF=1
- ROR AL,1 bi dalo: AL = D2h, CF=1



Binarni sadržaj, uključujući i zatečenu vrednost CF (smatrajući prvi operand i CF jedinstvenim podatkom) se kopira ulevo ili udesno. Sada se CF i čita, ne samo upisuje u njega.

- Rotacije nemaju aritmetičku interpretaciju.

## Primene instrukcija za pomeranje i rotiranje binarnog sadržaja

- Pristup neimenovanim višim delovima registara E... i R...  
Na primer: da se pristupi višem delu registra EAX: ROR EAX, 16. Viši deo registra će se sada nalaziti u AX, prethodna vrednost AX nije izgubljena.
- DX:AX → EAX (i obrnuto)  
SHL EDX, 16  
OR EAX, EDX ; pod pretpostavkom da je viši deo EAX bio nula. Ako nije, mora se prvo obrisati, npr.: SHL AX, 16; SHR AX, 16
- Nalaženje obrazaca u binarnom sadržaju  
Npr naći broj jedinica u binarnom podatku.
  - korišćenjem CF  
MOV CL, 0 ; u CL će biti broj jedinica  
MOV AX, podatak

<petlja sa 16 ponavljanja:>

SHL podatak, 1

ADC CL,0

- bez korišćenja CF

MOV AX, podatak

MOV CL, 0

<petlja sa 16 ponavljanja:>

MOV DX,1

AND DX, AX

add CL, DL

SHR AX,1

*(dok se ne upoznaju instrukcije za petlje, navedeni primeri se mogu testirati višestrukim ponavljanjem koda tela petlje)*