

The following sequence is given:

```
Xor edx, edx    ; EDX = 0
Mov dl, 0fh
```

Write an instructions sequence that multiplies by 4 the quadword value from EDX:EAX.

„Shl edx:eax, 2”

Task : the first 2 bits from the initial EAX must become the last 2 bits from EDX

....?

Shl eax,1 ; the most significant bit from EAX is transferred in CF and a 0 is introduced at the right end of EAX

```
Rcl edx,1 ; ....
Shl eax,1
Rcl edx,1
```

```
Shl eax,2
Rcl edx, 2 ; NO WAY !!!!!!!
```

Instrucțiunile RCR și RCL sunt utile pentru realizarea de deplasări de biți implicând operanzi reprezentați pe mai multe cuvinte. De exemplu, secvența următoare multiplică cu 4 valoarea din EDX:EAX :

```
shl    eax, 1      ; bitul 31 din EAX este depus în CF
rcl    edx, 1      ; valoarea din CF se depune în bitul 0 din EDX
shl    eax, 1      ; bitul 31 din EAX se depune în CF
rcl    edx, 1      ; valoarea din CF se
```

depune în bitul 0 din EDX

### Ex.3 (discutat impreuna)

```
mov al, 200 ; AL = 11001000b = 0C8h = 200 (unsigned) = -56 (signed)
mov bl, -1   ; BL = 11111111b = 0FFh = 255 (unsigned) = -1 (signed)
cmp al, bl   ;
```

.....

```
jl et1 ; -56 < -1 ? YES, so the jump will be performed !!!
ja et2 ; 200 > 255 ? NO, so the jump will NOT be performed !!!
jb et3 ; 200 < 255 ? YES, so the jump will be made
jg et4 ; -56 > -1 ? NO so the jump will NOT be performed !!
```

### Ex.3 (din Exemple\_comentate – material in limba romana)

```
mov al, 200 ; al = 11001000b = 0C8h = 200 (fara semn) = -56 (cu semn)
mov bl, -1   ; bl = -1 = 255 = ffh = 11111111b
cmp al, bl   ;
```

```
jl et1 ; -56 < -1 , deci saltul SE VA EFECTUA !!!
ja et2 ; 200 < 255, deci saltul NU se va efectua !
jb et3 ; 200 < 255 , deci saltul SE VA EFECTUA !
jg et4 ; -56 < -1, deci saltul NU se va efectua !
```

```
mov al, 200 ; al = 11001000b = 0C8h = 200 (fără semn) = -56 (cu semn)
mov bl, -1   ; bl = 11111111b = 0FFh = 255 (fără semn) = -1 (cu semn)
cmp al, bl   ; al-bl = 11001001b = C9h = -55 (cu semn) = 201 (fără semn)
               (și se setează corespunzător OF=ZF=0 și CF=SF=1)
```

Deci pe cine comparăm de fapt aici? Pe 200 cu -1 așa cum precizează valorile de la inițializare?  
Sau poate pe 200 cu 255? Sau pe -56 cu -1 ? Sau pe -56 cu 255?

Răspuns: comparăm întotdeauna pe 0C8h cu 0FFh sau în exprimare binară pe 11001000 cu 11111111. Efectul va fi unul singur: afectarea corespunzătoare a flag-urilor în urma efectuării scăderii fictive AL-BL. Modul de exprimare corect al comparației efectuate în baza 10 nu este dedus din acțiunea instrucțiunii CMP (care nu distinge absolut de loc între cele 4 variante posibile de comparare de mai sus) ci pe baza unor eventuale instrucțiuni ulterioare care vor avea ele rolul de a interpreta în unul din cele 4 moduri de mai sus comparația efectuată. Să urmărim în acest sens variantele de comparare de mai jos identificate prin utilizarea instrucțiunilor corespunzătoare de salt condiționat:

jl et1 ; evident că  $200 \nless -1$  deci la prima vedere pare că nu este îndeplinită condiția necesară pentru efectuarea saltului... să nu uităm însă faptul că JL (Jump If Less) interpretează rezultatul comparației ca fiind cu semn (deci -55) aceasta însemnând implicit și faptul că scăderea este interpretată ca  $(-56 - (-1))$  deci și operanzii vor fi amândoi interpretați cu semn... cum  $-56 < -1$  iată că și intuitiv condiția se verifică (pe lângă justificarea tehnică a îndeplinirii condiției de salt  $SF \neq OF$ ) și deci **saltul se va efectua** ! Deci chiar dacă programatorul a furnizat la inițializare valorile 200 și -1, utilizarea instrucțiunii JL a provocat interpretarea comparației ca fiind între -55 și -1 și nu între 200 și -1! (explicația de aici și faptul că saltul se va efectua vă poate ajuta să “demonstrați” unor colegi cum 200 poate fi mai mic decât -1 !!!)

ja et2 ; deoarece  $200 > -1$  în acest caz ne-am aștepta ca saltul să se efectueze... însă utilizarea instrucțiunii JA (Jump if Above) impune interpretarea fără semn, deci varianta de comparație corectă aici este comparația lui 200 cu 255 și cum  $200 \nless 255$  condiția nu este îndeplinită și deci **saltul nu se va efectua** (iată deci cum se poate “demonstra” că 200 nu este superior valorii -1 !!!). Ca o confirmare, se poate vedea că nici condiția tehnică impusă de JA nu este îndeplinită: ar trebui să avem  $CF=ZF=0$ , însă în cazul nostru  $CF=1$  deci saltul nu se va efectua.

jb et3 ; intuitiv  $200 < 255$ , iar tehnic  $CF=1$  deci **saltul se efectuează**

jg et4 ; intuitiv  $-56 \nless -1$ , iar tehnic deși  $ZF=0$  nu este îndeplinită și condiția  $SF = OF$  deci  
; saltul **nu se va efectua**

Ca urmare din cele 4 situații teoretic posibile de mai sus, vom întâlni concret numai două:

- comparație fără semn (200 cu 255) - impusă de “above” sau “below”
- comparație cu semn (-56 cu -1) – impusă de “less than” sau “greater than”

Nu putem așadar compara de fapt pe 200 cu -1 așa cum au fost specificate valorile la inițializare și nici pe -56 cu 255 deoarece **interpretarea este ori cu semn ori fără semn pentru ambii operanzi!**