Bit forgató (Rotate) és léptető (Shift) utasítások

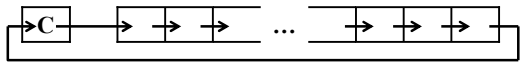
Forgatják (Rotate) illetve léptetik (Shift) **op** tartalmát. A forgatás/léptetés történhet

- 1 bittel,
- vagy byte illetve word esetén a **CL** regiszter alsó 3 illetve 4 bit-jén megadott bittel jobbra (Right) vagy balra (Left).

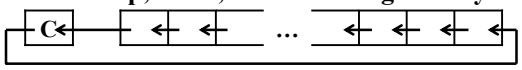
Az utoljára kilépő bit lesz a **Carry** új tartalma.

A rotálás történhet a **Carry**-n keresztül, ilyenkor a belépő bit a **Carry**-ből kapja az értékét:

RCR op,1/CL ; Rotate through Carry Right

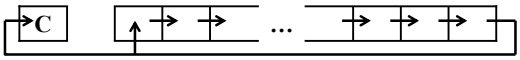


RCL op,1/CL ; Rotate through Carry Left

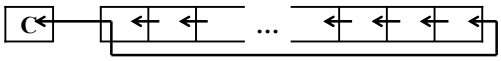


A rotálás történhet úgy, hogy **Carry** csak a kilépő bitet fogadja, a belépő bit értékét a kilépő bit szolgáltatja:

ROR op,1/CL ; ROtate Right

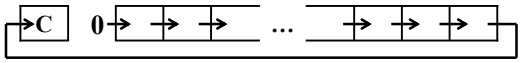


ROL op,1/CL ; ROtate Left



Logikai léptetés jobbra: A belépő bit **0**:

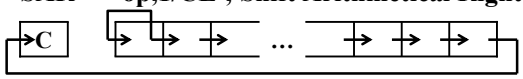
SHR op,1/CL ; SHift Right



Előjel nélküli egész számok **2 hatványával** történő osztására alkalmas.

Aritmetikai léptetés jobbra: A belépő bit **op** előjele:

SAR op,1/CL ; Shift Arithmetical Right



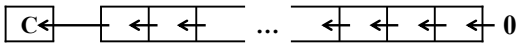
Előjeles egész számok **2 hatványával** történő osztására alkalmas. Negatív számok esetén csal!

Balra léptetéskor a belépő bit mindig **0**:

SHL op,1/CL ; SHift Left

SAL op,1/CL ; Shift Arithmetical Left

SAL ≡ SHL



Előjel nélküli vagy előjeles egész számok **2 hatványával** történő szorzására alkalmas.


```
hexa    proc                ; ax kiírása hexadecimálisan
; legyen a példa kedvéért:  ax = 1234H
        xchg    ah,al       ; ah és al felcserélése
; most:                     ax = 3412H, al = 12H
        call    hexa_b      ; al (az eredeti ah) kiírása
; kiírtuk, hogy 12
        xchg    ah,al       ; ah és al visszacserélése
; most újra:                ax = 1234H, al = 34H
        call    hexa_b      ; al kiírása
; most kiírtuk, hogy 34, tehát eddig kiírtuk, hogy 1234
        ret                 ; visszatérés
hexa    endp                ; a hexa eljárás vége
```

```
hexa_b  proc            ; al kiírása hexadecimálisan
; az első híváskor:     al = 12H
        push    cx      ; mentés a verembe
        mov     cl,4    ; 4 bit-es rotálás előkészítése
        ROR     al,CL   ; az első jegy az alsó 4 biten
; most:                 al = 21H
        call    h_jegy  ; az első jegy kiírása
; kiírtuk, hogy 1
        ROR     al,CL   ; a 2. jegy az alsó 4 biten
; most újra:            al = 12H
        call    h_jegy  ; a második jegy kiírása
; most kiírtuk, hogy 2, tehát eddig kiírtuk, hogy 12
        pop     cx      ; visszamentés a veremből
        ret             ; visszatérés
hexa_b  endp            ; a hexa_b eljárás vége
```



```
h_jegy  proc            ; hexadecimális jegy kiírása
        push    ax      ; mentés a verembe
        AND     al,0FH  ; a felső 4 bit 0 lesz,
                        ; a többi változatlan
        add     al,'0'  ; + 0 kódja
        cmp     al,'9'  ; ≤ 9 ?
        jle     h_jegy1 ; ugrás h_jegy1 -hez, ha igen
        add     al,'A'-0AH-'0' ; A…F hexadecimális
                        ; jegyek kialakítása
h_jegy1: mov    ah,14   ; BIOS hívás:
        int     10H     ; karakter kiírás
        pop     ax      ; visszamentés a veremből
        ret             ; visszatérés
h_jegy  endp            ; a h_jegy eljárás vége
```


Processzor vezérlő utasítások I.

A processzor állapotát módosít(hat)ják.

A **STATUS** bitjeinek módosítása			
Flag	**CL**ear	**S**e**T**	**C**o**M**plement
C	**CLC**	**STC**	**CMC**
D	**CLD**	**STD**	
I	**CLI**	**STI**	

Processzor vezérlő utasítások II.

NOP ; **NO** o**P**eration, üres utasítás,
; nem végez műveletet.

WAIT ; A processzor várakozik, amíg más
; processzortól (pl. lebegőpontos
; segédprocesszortól) kész jelzést nem kap.

HLT ; **HaLT**, leállítja a processzort.
; A processzor külső megszakításig
; várakozik.

Input, output (I/O) utasítások (I8086/88)

A külvilággal történő információ csere **port**-okon (kapukon) keresztül zajlik. A kapu egy memória cím, az információ csere erre a címre történő írással, vagy erről a címről való olvasással történik. Egy-egy cím vagy cím csoport egy-egy perifériához kötődik. A központi egység oldaláról a folyamat egységesen az **IN** (input) és az **OUT** (output) utasítással történik.

Input, output (I/O) utasítások (I8086/88)

A perifériától függ, hogy a hozzá tartozó port 8 vagy 16 bites. A központi egységnek az **AL** illetve **AX** regisztere vesz részt a kommunikációban. A port címzése 8 bites közvetlen adattal vagy a **DX** regiszterrel történik.

IN AL/AX,port
; AL/AX ⇐ egy byte/word a port-ról

OUT port,AL/AX
; port ⇐ egy byte/word AL/AX-ből

A periféria oldaláról a helyzet nem ilyen egyszerű. Az input információ „**csomag**okban” érkezik, az output információt „csomagolva” kell küldeni. A csomagolás (vezérlő információ) mondja meg, hogy hogyan kell kezelni a csomagba rejtett információt (adatot). Éppen ezért az operációs rendszerek olyan egyszerűen használható eljárásokat tartalmaznak (**BIOS** – Basic Input/Output System – rutinok, stb.), amelyek elvégzik a ki- és becsomagolás munkáját, és ezáltal lényegesen megkönnyítik a külvilággal való kommunikációt.

Megszakítás rendszer, interrupt utasítások

- Az **I/O** utasítás lassú ↔ a **CPU** gyors, a **CPU** várakozni kényszerül
- **I/O** regiszter (**port**): a **port** és a központi egység közötti információ átadás gyors, a periféria autonóm módon elvégzi a feladatát. Újabb perifériához fordulás esetén a **CPU** várakozni kényszerülhet.
- **Pollozásos technika** (~tevékeny várakozás)**:** a futó program időről időre megkérdezi a periféria állapotát, és csak akkor ad ki újabb **I/O** utasítást, amikor a periféria már fogadni tudja. A hatékonyság az éppen futó programtól függ.

Megszakítás

A (program) megszakítás azt jelenti, hogy az éppen futó program végrehajtása átmenetileg megszakad – a processzor állapota megőrződik, hogy a program egy későbbi időpontban folytatódhassék – és a processzor egy másik program, az úgynevezett **megszakítás kezelő** végrehajtását kezdi meg.

Miután a megszakítás kezelő elvégezte munkáját, gondoskodik a processzor megszakításkori állapotának visszaállításáról, és visszaadja a vezérlést a megszakított programnak: **Átlátszóság.**

Pl.: nyomtatás

- Nyomtatás pufferbe, később a tényleges nyomtatást vezérlő program indítása.
- Nyomtatás előkészítése (a nyomtató megnyitása), **HLT**.
- A továbbiak során a nyomtató megszakítást okoz, ha kész újabb adat nyomtatására. Ilyenkor a **HLT** utasítást követő címre adódik a vezérlés. A következő karakter előkészítése nyomtatásra, **HLT**. A bekövetkező megszakítás hatására a megszakító rutin mindig a következő adatot nyomtatja. Ha nincs további nyomtatandó anyag, akkor a nyomtatást vezérlő program lezárja a nyomtatót (nem következik be újabb megszakítás a nyomtató miatt), és befejezi a működését.

A **HLT** utasítás csak akkor szükséges, ha a nyomtatást kérő program befejezte a munkáját. Ellenkező esetben visszakaphatja a vezérlést. Ilyenkor az ő feladata az esetleg szükséges várakozásról gondoskodni a program végén.

Bevitel esetén olyankor is várakozni kell, ha még a beolvasás nem történt meg, és a további futáshoz szükség van ezekre az adatokra.

Jobb megoldás, ha a **HLT** utasítás helyett az operációs rendszer fölfüggeszti a program működését, és elindítja egy másik program futását.

Ez vezetett a **multiprogramozás** kialakulásához.

A megszakítás kezelő (megszakító rutin) megszakítható-e? Gyors periféria kiszolgálása közben megszakítás kérés, …

„Alap” állapot – „megszakítási” állapot, megszakítási állapotban nem lehet újabb megszakítás.

Hierarchia: megszakítási állapotban csak magasabb szintű ok eredményezhet megszakítást.

Bizonyos utasítások csak a központi egység bizonyos kitüntetett állapotában hajthatók végre, alap állapotban nem → csapda, szoftver megszakítás.

Megoldható az operációs rendszer védelme, a tár védelem stb.

A megoldás nem tökéletes: **vírus**.

I8086/88

Megszakítás kiszolgálásakor a **STATUS**, **CS** és **IP** a verembe kerül, az **I** és a **T** flag **0** értéket kap (az úgynevezett maszkolható megszakítások tiltása és folyamatos üzemmód beállítása), majd **(CS:IP)** felveszi a megszakítás kezelő kezdőcímét.

Átlátszóság: Amikor bekövetkezik egy megszakítás, akkor bizonyos utasítások végrehajtódnak, de amikor ennek vége, a **CPU** ugyanolyan állapotba kerül, mint amilyenben a megszakítás bekövetkezése előtt volt.

Megszakítás és csapda

Megszakítás (interrupt): Olyan automatikus eljárás hívás, amit általában nem a futó program, hanem valamilyen **B/K** eszköz idéz elő, pl. a program utasítja a lemezegységet, hogy kezdje el az adatátvitelt, és annak végeztével megszakítást küldjön. **Megszakítás kezelő**.

Csapda (trap): A program által előidézett feltétel (pl. túlcsordulás) hatására automatikus eljárás hívás. **Csapda kezelő**.

A csapda a **programmal szinkronizált**,
a megszakítás nem.

Interrupt (csapda) utasítások

Szoftver megszakítást (csapdát) eredményeznek.

INT i ; 0 ≤ i ≤ 255,
; megszakítás az **i.** ok szerint.

Az **INT 3** utasítás kódja csak egy byte (a többi 2 byte), így különösen alkalmas nyomkövető (**DEBUG**) programokban történő alkalmazásra.

A **DEBUG** program saját magához irányítja a **3**-as megszakítást. Az ellenőrzendő program megadott pontján (törés pont, **break point**) lévő utasítást (annak 1. bájtját) átmenetileg az **INT 3** utasításra cseréli, és átadhatja a vezérlést az ellenőrzendő programnak. Amikor a program az **INT 3** utasításhoz ér, a megszakítás hatására a **DEBUG** kapja meg a vezérlést. Kiírja a regiszterek tartalmát, és további információt kérhetünk a program állapotáról.
Később visszaírja azt a tartalmat, amit **INT 3** -ra cserélt, elhelyezi az újabb törés pontra az **INT 3** utasítást és visszaadja a vezérlést az ellenőrzendő programnak.

Ezek alapján érthetővé válik a **DEBUG** program néhány „furcsasága”:

- Miért „felejti el” a töréspontot? Ha ugyanis nem felejtené el – azaz nem cserélné vissza a töréspontra elhelyezett utasítást az eredeti utasításra – akkor a program nyomkövetésében nem tudna továbblépni.
- Miért nem lehet egy ciklus futásait egyetlen töréspont elhelyezésével figyeltetni, stb?

INTO ; megszakítás csak **O=1 (Overflow)**
; esetén a **4.** ok szerint

Visszatérés a megszakító rutinból

IRET ; IP, CS, STATUS feltöltése a
; veremből