Számítógép-architektúrák

Assembly nyelvű programozás alapjai
(vázlat)

Alapfogalmak

Assembly nyelv: gépközeli programozási nyelv, egy utasítása megfelel egy gépi utasításnak. Utasításait szimbolikus nevekkel illetjük, ami az adott utasítás angol nevének rövidítése (mnemonik). Utasításokon kívül tartalmaz még direktívákat (lásd a programok felépítésénél).

Assembler: fordítóprogram, ami assembly nyelvről tárgykódra (object code) fordít.

Linker (linkage editor): kapcsolatszerkesztő: az a program, ami a különböző magas szintű nyelvek (pl. Pascal, C, C++) és az assembly gépközeli nyelv programjainak fordítási egységeiből készült tárgykódú modulokat egy programmá szerkeszti össze.

Megjegyzések:

1. Egyes mikroszámítógépes assemblerek közvetlenül gépi kódú programot készítenek és azt a gép memóriájában helyezik el.

2. Cross compilernek hívjuk az olyan fordítóprogramot, amely az őt végrehajtó gép architektúrájától eltérő architektúrájú gépre fordít. (Ilyen fordítóprogramot használunk például, ha PC-n valamilyen mikrokontrollerre fejlesztünk programot.)

Assembly nyelvű programok felépítése

8080/8085 assembly esetén a program egy sora megfelelhet egy gépi utasításnak, állhat pusztán megjegyzésből, illetve tartalmazhat direktívát. Az első esetben a sor a következő részeket tartalmazhatja:

<címke mező> <műveleti mező> <operandus mező> <megjegyzés rész>

A **címke** betűvel kezdődik, betűket és számokat tartalmazhat, utána kettőspont áll.

A **műveleti mező**ben valamely gépi utasítás mnemonikja áll.

Az **operandus mező** tartalmát a gépi utasítás fajtája határozza meg, üres is lehet.

A **megjegyzés rész** opcionális, ha van, akkor pontosvesszővel kezdődik.

Ha egy sor pontosvesszővel kezdődik, akkor a fordító az egész sort megjegyzésnek tekinti.

A lehetséges direktívákat nézzük meg a Referenciakártyán!

Egy assembly nyelvű programnál mindig meg kell adnunk az ORG direktívával, hogy a gépi kódú program milyen memóriacímtől kezdődjön.

Változóink számára a megfelelő direktívákkal bájt vagy szó méretű helyeket tudunk lefoglalni...

Lehetőség van szimbolikus konstansok használatára is.

Assembly nyelvű programok írása

Nem teljesen triviális feladatok esetén a feladatot részekre bontjuk és megtervezzük a részek együttműködését (például: melyik szubrutin milyen funkcióért felelős, ezekből hogyan épül fel a teljes program; paraméterek átadásának módja, konkrétan mit és hogyan adunk át, stb.).

Egyszerű feladat (vagy összetett feladat kellően kicsi, így már egyszerű része) esetén célszerűen folyamatábrát készítünk, amelybe lehetőleg olyan tevékenységeket és feltételvizsgálatokat írunk, amit 1-1 gépi utasítással vagy legfeljebb néhánnyal (erre látunk példát) meg tudunk oldani.

Összetett feladatokra szubrutinhívást alkalmazunk.

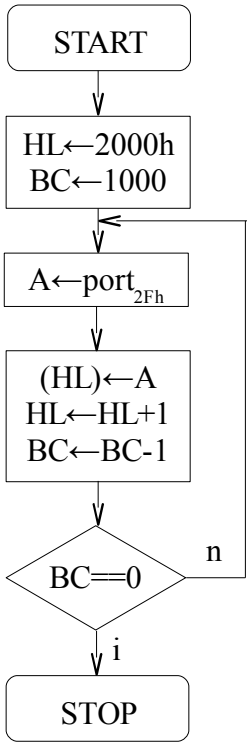
A 8085 regiszter- és utasításkészletének ismeretében kis gyakorlattal eldönthetjük, hogy mely változókat tároljunk memóriában, illetve regiszterben/regiszterpárban, és azok közül is melyikben.

Programok írásakor mindig használjuk a Referenciakártyát!

Mintafeladat

Olvassunk be 1000 bájt adatot a 2Fh I/O portról, és tároljuk el a 2000h címtől a memóriában!

A feladat *egy* lehetséges megoldása:




```
        ORG 0000H
;
; kezdőértékek beállítása
        LXI H,2000h
        LXI B,1000
;
; beolvasás a perifériáról
input:  IN 2Fh
;
; tárolás, mutató++, számláló--
        MOV M,A
        INX H
        DCX B
;
; BC regiszterpár értéke 0-e?
        MOV A,B
        ORA C
        JNZ input
;
        END
```

További feladatok

1. feladat: *Másoljuk át a memória 1000h címtől kezdődő 1kB méretű részét a 2000h címtől kezdően!*

2. feladat: *Mint az első feladat, de a másolandó blokk mérete legyen 5kB! Mit veszünk észre?*

3. feladat: *Specifikáljunk és írjunk szubrutint, amely 2 tetszőleges hosszúságú számot összead!*

Egy lehetséges specifikáció: A számokat binárisan ábrázoljuk (kettes komplemens kódban) az alvég bájtsorrendet követve (legkisebb helyiértékű bájt van elöl), hosszuk a BC, az első operandus címe a HL, a másodiké a DE regiszterpárban található. Az eredmény az első operandus helyén képződjön. Bemenő átvitel (carry) nincs, a kimenő átvitel értéke a legyen a C (carry) jelzőbitben!

Adjunk olyan specifikációt is, ahol az operandusok értéke NEM változik meg (az eredmény máshol képződik)!