GÉPIPARI ÉS AUTOMATIZÁLÁSI MÜSZAKI FŐISKOLA
MATEMATIKA-FIZIKA TANSZÉK

S E G É D L E T

HT 1080Z számitógép programozásához
essembler nyelven
/Belső használatra/

ÖSSZEÁLLITOTTA: KÁRPÁTI BÉLA

T A R T A L O M

B E V E Z E T É S

A HT 1080Z gép alapvetően a BASIC programozási nyelv használatára készült személyi számitógép. A 12 KB terjedelmü BASIC interpreter ROM memóriában található, amely a gép bekapcsolása után rögtön a felhasználó rendelkezésére áll, vagyis a gép BASIC nyelven azonnal programozható.

A BASIC nyelvnek sok előnye van, ezek közül a legfontosabbak:

- a BASIC egy feladat orientált, nem gépfüggő nyelv, ezért a programozótól szinte semmi hardver ismeretet nem igényel,

- a BASIC interaktiv kapcsolatot tart a programozóval, igy a program irása, tesztelése és módositása könnyü.

A BASIC nyelvnek vannak hátrányai is:

- a BASIC nyelven irt programoknak a leghosszabb a futási idejük,
- a BASIC interpreternek a már tesztelt hibátlan programok futtatása idején is a memóriában kell lennie.

Ezek a hátrányok más magasszintü programozási nyelvek használatával /ha rendelkezésre állnak ezen nyelvek forditóprogramjai az adott gépre/ bizonyos mértékben kiküszöbölhetők. Vannak viszont olyan feladatok, melyek igazán

jól egyetlen magasszintü programozási nyelven sem oldhatók meg.

Ilyen feladatok:

- idő-kritikus feladatok /nagyon gyors futásigény/,
- minimális memória felhasználás,
- a gép perifériáinak közvetlen fizikai kezelése.

Ilyen feladatok elsősorban a számitógépes folyamatirányitás,perifériák illesztése és számitógépes célrendszerek megvalósitása közben adódnak.

A fenti feladatok megoldására azért nem alkalmasak a magasszintü nyelvek, mert mint gép független programozási nyelvek nem tudják a konkrét gép hardver adottságait optimálisan kihasználni, ugyanis a gép alap-utasitáskészletét, regisztereit a programozó közvetlenül nem tudja használni. Ezt a lehetőséget, már mint a gép mikroprocesszorának elemi utasitásait,regisztereit, cimzési módjait,megszakitási rendszerét leghatékonyabban kihasználni assembler szintü programozással lehet. Az assembler ugyanis egy gép /mikroprocesszor/ függő programozási nyelv, amelynek minden forrássora a gép egy elemi utasitását realizálja. A gép memóriájának a kezelése is /tárkiosztás/ az assembler nyelvben közvetlenül történik. Ezek miatt az assembler nyelven történő programozás részletesebb hardver /mikroprocesszor/ ismereteket követel.

Az assembler programozás folyamata:

1./ Az assembler nyelv szintaktikai szabályai szerint megirt forrásprogram bevitele a gépbe.

2,/ A forrsáprogram leforditása, listázása a gépi kódú tárgyprogram előállitása.

3./ Tárgyprogram futtatása, tesztelése.

A fenti lépések megvalósitásához a következő segédprogramokra van szükségünk:

1./ Szövegszerkesztő /editor/

2./ Assembler fordító

3./ Monitorprogram /debugger/

A HT 1080Z számitógép assembler szintü programozását ezen programok megléte teszi lehetővé. A HT 1080Z gép mikroprocesszora Z80 tipusú mikroprocesszor, amely felülről kompatibilis az INTEL 8080 és INTEL 8085 mikroprocesszorokkal, ezek a processzorok a legelterjedtebb 8 bites processzorok. A szocialista országokban szinte csak ezeket a mikroprocesszorokat gyártják, indokolttá teszik a HT 1080Z számitógépnek a BASIC programozás mellett ezen processzorok gépikódú programozásának elsajátitásához való használatát is.

A HT 1080Z számitógép felépitése assembler szintü programozáshoz

Az assembler szintü programozáshoz, mint mikroprocesszor függő programozási nyelvhez ismerni kell a gép mikroprocesszorának, jelen esetben a Z80 tipusú processzor

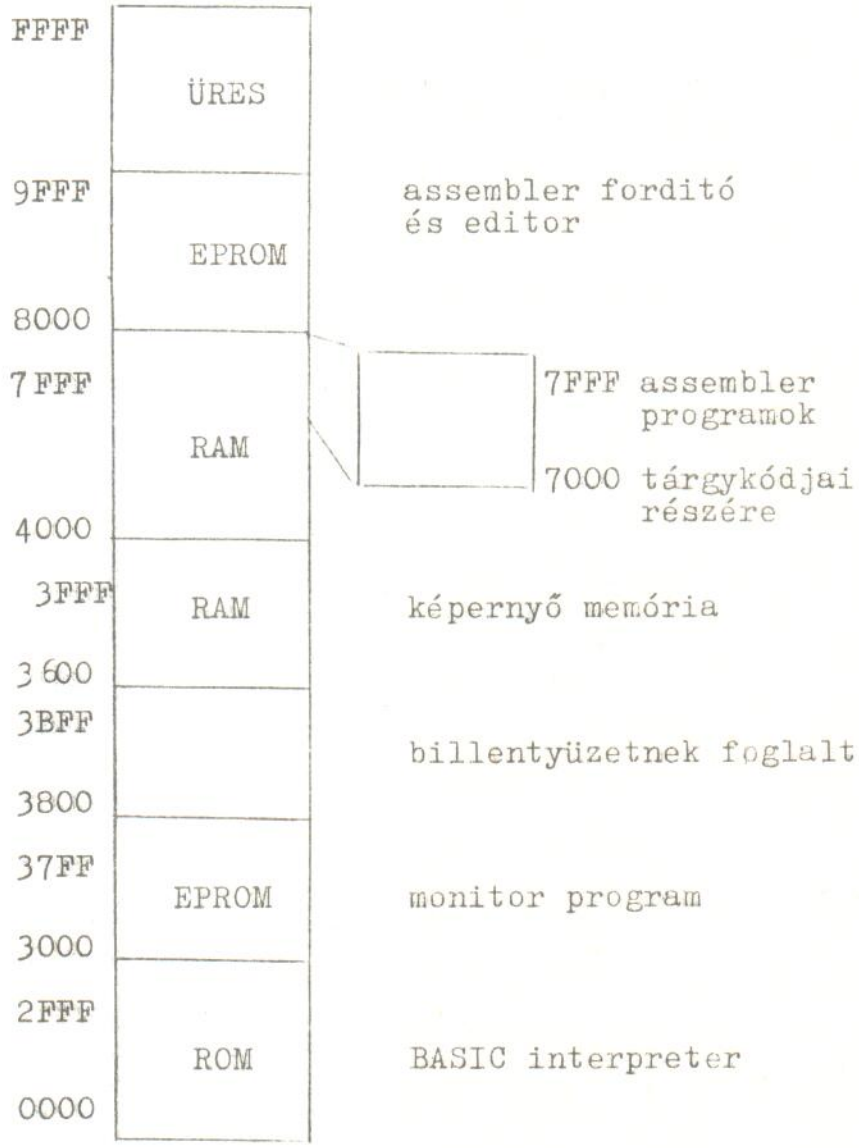
- regisztereit,
- cimzési módjait,
- utasitás készletét.

Ezeket a csak mikroprocesszortól függő ismereteket a "Programozási segédlet a Z80 tipusú mikroprocesszorhoz" cimü füzet tartalmazza.

Mivel essembler nyelven a memória kezelése is közvezlenül történik, ismerni kell a konkrét számitógép memória térképét.

Ez a HT 1080Z gépnél a következő:

Cim/memória/


FFFF
ÜRES
9FFF
EPROM
assembler fordító
és editor
8000
7FFF
RAM
7FFF assembler
programok
7000 tárgykódjai
részére
4000
3FFF
RAM
képernyő memória
3600
3BFF
billentyüzetnek foglalt
3800
37FF
EPROM
monitor program
3000
2FFF
ROM
BASIC interpreter
0000

A HT 1080Z számitógép monitor programja

A gép alapkiépitésben is tartalmaz egy monitor-programot, amely lehetővé teszi hexadecimális formában:

- a processzor regisztereinek kijelzését, módositását,
- a memória tartalom listázását,
- a memória tartalom módositását,
- gépi kódú program inditását.

A monitorprogram aktivizálása BASIC üzemmódból a SYSTEM paranccsal történik, amely után /12710 formában meg kell adni a monitorprogram belépési pontját.
Ezután a monitor a CPU regiszterek tartalmának kiirásával jelentkezik, ez a monitor alapállapota.
A monitor parancsai egybetüs kódokkal aktivizálhatók.

1. "B" visszatérés a BASIC üzemmódba.
2. "D" memória tartalom listázása.

   DHHHH

   ahol HHHH egy kezdőcim, amelytől kezdve 16 byte tartalmát irja ki egysorban.

   billentyü lenyomása csökkenti a cimeket,

   billentyü lenyomása növeli a cimeket.

   Kilépés: bármely más billentyü megnyomásával.

3. "R" regiszterek tartalmának módositása:
Kiirja a regiszter nevét és tartalmát ha a kérdéses regisztert nem akarjuk módositani, nyomjuk meg az X billentyüt,ellenkező esetben négy hexadecimális karakter megadásával állitsuk be a regiszter/pár/ új tartalmát.

4. "M" memória tartalom módositása:
MHHHH
ahol HHHH egy kezdőcim, amelynek kiirja a tartalmát és két hexadecimális karakter megadásával módosithatjuk a cim tartalmát,ezután automatikusan a következő cim tartalmát jelzi ki, ahol folytathatjuk a módositást.
Kilépés: X billentyü megnyomásával.

5. "G" gépikódú program inditása:
GHHHH,TTTT
ahol HHHH a program kezdőcime,
TTTT a program vége, vagy töréspont.
A töréspontként megadott cim elérésekor visszatér a monitor alapállapotába, vagyis a CPU regiszterek tartalmának kijelzéséhez.

A monitorprogram a fenti parancsok segitségével lehetővé teszi a gépikódú programok tesztelését,futtatását.

HT 1080Z számitógép szövegszerkesztő és assembler forditó programja

A gép alapkiépitésben nem, de bővitésként a 8000H cimtől kezdődően EPROM tárolóban tartalmaz egy programcsomagot az assembler szintü programozás különböző folyamatainak /forrásszöveg szerkesztés,forditás,tárgyprogram tesztelés/ lehetővé tételéhez.

Ezen programcsomag aktivizálása a gép bekapcsolása után a BASIC üzemmódból a következő lépésekkel történik:

SYSTEM

/32768

Ezek hatására a programcsomag az EPROM tárolóból bemásolódik a RAM tárolóba 4300H cimtől kezdődően és

? promttal bejelentkezik.

Nézzük először a forrásszöveg beirásának és javitásának lehetőségeit. Ezt alapvetően egy un. soreditor segiti, ami azt jelenti,hogy a szövegszerkesztőnek van egy pointere /sormutatója/ és a parancsok a pointer aktuális értékére vonatkoznak.

Az editor parancsai:

"E" forrássorok beirása vagy beszúrása a pointer aktuális értékétől kezdődően.

Ezen parancs kiadása után az aktuális sorszám jelenik meg, ahová az először begépelt forrássor kerül.

Egy forrássor begépelése után a sorszám automatikusan eggyel növekszik.

Ha egy új forrásprogram begépelését kezdjük, ne feledkezzünk meg arról, hogy a program első két sora:

ORG cim

LOAD cim

legyen. A cim értéke ne legyen kisebb 7000H értéknél, ugyanis ezek a direktivák határozzák meg az essembler fordító részére, hogy a leforditott tárgyprogram hová kerüljön. A forrássorok beirásánál még a következőkre ügyeljünk:

- ha van cimke, az a sor első pozicióján kezdődjön, a cimke első karaktere betü legyen és a cimkét ":" -tal fejezzük be,

- ha nincs cimke, legalább a sor első pozicióját üresen kell hagyni /space/,

- a müveleti mező és az operandus mező között legalább egy space legyen,

- az operandus mezőben használt számkonstansok alapértelmezésben decimálisak,

- hexadecimális számokat "H", oktális számokat "O" karakterrel kell lezárni.

- számkonstansnak mindig decimális számjegy karakterrel kell kezdődnie, még hexa számoknál is / pl.ØF6H/.

Ha a forrássorok folyamatos beirását meg akarjuk szüntetni, az "E" parancsból egy csak "."-ot tartalmazó sor begépelésével tudunk kilépni.
Begépelt forrássorok listázását,módositását,törlését, beszúrását a következő parancsok segitik:

"Pn" - a képernyőre listáz n /decimális/ sort. Listázás után a pointer az utolsó listázott sorra mutat.

"T" - pointer a file elejére megy.

"B" - pointer a file végére megy és az utolsó forrássor utánra mutat. A képernyőn az "EOF" felirat jelenik meg. Ha ezután alkalmazzuk az "E" parancsot, akkor hozzáirhatunk a file végéhez.

"Dn" - A pointert n sorral lefelé mozgatja. A parancs végrehajtása után kiirja azt a sort, ahova a pointer mutat.

"Un" - A pointert n sorral felfelé mozgatja. A parancs végrehajtása után kiirja azt a sort, ahová a pointer mutat.

"N" - Azt a sort,amelyre a pointer mutat törli, és helyére új sort lehet begépelni.

"Zn" - A pointer helyétől kezdve n sort töröl, Utána a pointer az első nem törölt sorra mutat, amit ki is ir.

"L forrássor" - A pointer állásától lefelé megkeresi az adott forrássort. Ha megtalálja, kiirja az adott forrássort és a pointer erre mutat. Ha nem találja "EOF" kiirás jelenik meg és a pointer a file végére mutat.

"K" - Törli a teljes forrásprogramot.

"H" - Kiirja a forrásprogram memóriában való elhelyezkedésének kezdő- és végcimét hexadecimálisan.

Assembler forditás

"A" parancs inditja a forditót amely a forrásszöveg első sorától az END direktivát tartalmazó sorig elvégzi a forditást.

Ezen parancs után még három lehetséges opciót lehet megadni:

V - képernyő
E - printer
C - kazetta

A különböző opcióknak megfelelően esetleg új kérdést tesz fel /kazettánál kezdőcim és név/.

Hosszabb programnál figyeli a képernyő illetve a lap méretét.

Képernyőre való listázáskor 15 sor után megáll és villogó kurzorral kb. 4 másodpercig vár.

Ilyenkor bármely billentyü lenyomásával a listázás megállitható, majd bármely billentyü lenyomásával újra inditható.
Nyomtatásnál a kurzor nem villog,csak ha megállitható a forditás. 60 sor után lehetőség van új lapot kezdeni.

"S" - forditás után a szimbólumtáblát irja ki ABC sorrendben. Opció - V és E lényegében ugyanazok a jellemzői mint az A parancsnak. Az "S" után ha betüt irunk, akkor csak ezzel a betüvel kezdődő szimbólumokat irja ki.

A kiirás formátuma:

szimbólum hexacim szimbólum hexacim ....

egy sorba négy szinbólumot és cimét irja ki.

Az assembler hibajelzései

A forditás során az esetleges hibánál a fordító leáll, hibajelzést ir ki, és a pointert a hibás sorra állitja, ami a képernyőn megjelenik.

Hibajelzések:

DOUBLE SYMBOL - többször definiált,deklarált szimbólum

RSVD - a szimbólum neve lefoglalt szó, /pl.utasitás/ használata tilos

FULL - a szimbólumtábla betelt

ORG	8-	nincs kezdőpont, az ORG direktiva hiányzik, vagy hibás
OPND	-	hibás az operandus
UNDEF	-	a szimbólum nincs definiálva, vagy hiányzik
EOP	-	nincs END utasitás
HUH?	-	értelmetlenséget talált az editor vagy a fordító

Kazetta parancsok

A parancsok kétféle programot kezelnek; forrás illetve object programokat. Object programok esetében a parancs után O karaktert kell irni /RO,WO,VO/.

"R" - kazettáról file-t olvas. Paraméterként a file nevét kell megadni. Forrás olvasásakor az egyébként minden olvasáskor megjelenő L karakter mellett, ami a helyes olvasást jelzi - az olvasás végén kiirja az aktuális file memória kezdő és végcimét A hibás olvasást karakter jelzi. Az olvasásnál az olvasott programot az esetlegesen memóriában lévő program után tölti be /Merge funkció/. Object módban az aktuális cimre.

"W" - kazettára file-t ir ki. Forrásnál csak file-nevet, objectnél cimet is kór.

"V" - ellenörzi a kivitel helyességét.

Programbelövő parancsok

"C" - memóriatartalom áthelyezése
Paraméterei START: honnan kezdőcim
STOP: -"- végcim
DEST: hova kezdőcim
Nem fogadja el a parancsot és HUH? hibajelzést ad, ha a STOP cim kisebb mint a START cime. A DEST hibás cime esetenként a teljes rendszer lefagyását okozhatja, akkor csak kikapcsolás után lehet újra életrekelteni. Az esetek többségében nem okoz problémát.

"F" - memória feltőltése adott konstanssal
Paraméterei: START: honnan kezdőcim
STOP: meddig végcim
DATA: 1 byte-os adat
/lehet decimális is/.
Amennyiben az adat nagyobb 1 byte-os számnál, nem jelez hibát.hanem a felső helyiértékekről nem vesz tudomást.
Hexa számokról csak az utolsó kettővel foglalkozik. Itt nem fogad el kisebb STOP cimet a START cimnél.
Ha nem adunk semmit START cimnek, automatikusan ØØØØ-nak tekinti, igy véletlenül felülirhatjuk az editor prigramját a RAM-ban. Ilyenkor RESET és SYSTEM /32768-cal újra kell tölteni az editort. A forrás nagy valószinüséggel elveszik.

"X" - Kiirja a képernyőre a regiszterek tartalmát. Felső sorba a fő regiszterek, alsó sorba a másodlagos /AF ;BC';.../ regiszterek tartalmát.

"M" - memória módositása:
Mnnnn cim begépelése után kiirja a memória tartalmát hexa formában. NL benyomásával változtatás nélkül egy cimmel továbblép. Memória módositásnál a > karakter után lehet beirni az új tartalmat decimális /pl.F6>122/, hexadecimális /pl. F6>C3H/ és oktális /pl. F6>750/ formában.
Az egy byte-nál magasabb helyiértékeket nem veszi figyelembe. A módositást NL billentyü lenyomásával zárjuk. A módositott értékkel felülirja a régi tartalmát,majd a következő memóriacimre lép. A funkcióból "." karakterrel lehet kilépni. Minden hibára HÜH? hibajelzéssel parancskérő állapotba tér vissza.

"Q" - memóriatartalom kiirása:
Qnnnn cimtől kezdve 64 byte tartalmát irja ki, egy sorba 8-at. Mellé 2x8-as sorba a tartalomnak megfelelő ASCII karaktert. Az első csoportban a 7. bit "Ø", a másodikban a 7. bit "1". A vezérlőkarakterek helyén "." látható. A következő Q parancsra folyamatosan irja ki a tartalmat.

"G" - felhasználói program inditása:

Paraméterei: ADDR inditócim
BKPT töréspont

Ügyelni kell,hogy az inditócim megadása helyesen történjen. /Hexa számnál H, oktálisnál O legyen a szám után /
Hibás inditócim megadása után a lehetséges javitás a töréspont helytelen megadása /pl. betüket/, akkor HUH? hibajelzéssel a rendszer parancsállapotba tér vissza.
Hibás inditásnál lemerevedik a rendszer, és sok esetben csak kikapcsolás után inditható újra!

Amennyiben nem adunk meg kezdőcimet, a rendszer inditócimnek az aktuális PC-tartalmat veszi annak.
Töréspontnál ez nincs. Célszerü megoldás, ha a program utolsó utasitása JP 43ØØH /12152/, vagy JP 31A6H /12710/.

JP 43ØØH-nál az editor üzemmódba tér vissza, JP 31A6H-nál az eredeti HT monitort inditja, ahol a regiszter-tartalmak láthatók és módosithatók.
Ilyenkor töréspont megadása nem szükséges.

HT 1080Z BASIC INTERPRETERJÉNEK A GÉPIKÓDU PROGRAMOZÁSHOZ HASZNÁLHATÓ RUTINJAI

Gyakran használt részfeladatok elvégzéséhez az alábbi szubrutinok használhatók:

NÉV	CIM	FELADATOK	PARAMÉTEREK
INCH1	002B	Billentyüzet olvasása azonnal visszatér	Be: - Ki: A=karakter kód A=Ø ha nem volt megnyomva
INCH2	0049	Billentyüzet olvasása, vár egy billentyü leütéséig	Be: - Ki: A=karakter kód
OUTCH	0033	Egy karakter kiirása a képernyőre	Be:A=karakter kód Ki: -
CLS	01C9	Képernyő törlése	Be: - Ki: -
INHX4	3392	Négy helyértékü hexa szám beolvasása	Be: - Ki:HL=bin.szám
INHX2	3393	Két helyértékü hexa szám beolvasása	Be: - Ki:A=B=bin.szám
EXP	3347	Visszatérés monitor-üzemmódba	Be: - Ki: -
OUTSTR	28A7	Sting kiirása, a string végét ØØ jelzi	Be:HL=string kezdőcime
OUTSEL	409C	Adatirány jelölő byte OUTSTR-hez	FF= kazetta 00= képernyő 01= nyomtató
DELAY	0060	/BC/*14,6 sec késleltetés	Be:BC=ciklus szám

DECOUT	OFAF	Egész tipusú BNDC konverzió és kiirás	Be:HL=bin.szám Ki: -
VAL	1E5A	String konvertálása bináris egészre String vége: ØØ	Be:HL=string kezdőcim Ki:DE=bin.szám
MLTINT	OBF2	egész tipusú szorzás túlcsordulás esetén az eredmény lebegőpontosan az SA-ba kerül	Be:HL=szorzandó DE=szorzó KI:HL=szorzat
DIVINT	2490	egész tipusú osztás	BE:DE=osztandó HL=osztó Ki:SA=hányados

LEBEGŐPONTOS SZÁMITÁSOKHOZ HASZNÁLHATÓ RUTINOK

Lebegőpontos szám ábrázolása:

1 byte exponens
3 byte mantissza

Az exponens ábrázolása

Az exponens byte a 2-es alapú kitevőt tartalmazza:

$$-128 \leq \text{kitevő} \leq +127$$

Kitevő értéke /dec/	exp.byte tartalma /hexa/	
0	80	
1	81	
-1	7F	
10	8A	
-10	76	
127	FF	
-127	01	
-128	00	/gépi nulla/

A mantissza ábrázolása:

A mantissza előjele külön nincs tárolva,illetve a pozitiv mantisszát úgy jelölik,hogy tároláskor a normált mantissza legmagasabb helyértékü bitje helyén 0 van, negativ mantissza esetén a legmagasabb helyértékü bit is tárolva van. A lebegőpontos aritmetikai müveletek elvégzése:

A müveletek egy 4 byte-os a RAM-ban elhelyezett un. szoftver akkumlátorban /SA/ és a BCDE regiszterek összevonásával keletkező un. regiszter akkumlátor /RA/ mint operandusok között mennek végve.

A lebegőpontos szám elhelyezése az SA-ban /memóriában/ és a regiszterekben:

SA	/4121/	/E regiszter/ mant.legkisebb helyérték
SA+1	/4122/	/D regiszter/ mant.középső "
SA+2	/4123/	/C regiszter/ mant.legmagasabb "
SA+3	/4124/	/B regiszter/ exponens

Pl. néhány szám lebegőpontos ábrázolása:

Szám	SA /E/	SA+1 /D/	SA+2 /C/	SA+3 /B/
1	00	00	00	81
-1	00	00	80	81
0,5	00	00	00	80
-0,5	ÖO	00	80	80
1027	00	60	00	8B
-1027	00	60	80	8B
0,1	CD	CC	4C	7D
-0,1	CD	CC	CC	7D
262159,25	E8	01	00	93

Lebegőpontos rutinok

Név	Cim	Feladat
ADDSP	0716	SA= SA+RA
SUBSP	0713	SA=RA-SA
MLTSP	0847	SA=SA RA
DIVSP	08A2	SA=RA/SA
SQR	13E7	SA=SQR/SA/
EXP	1439	SA=EXP/SA/
LOG	0809	SA=LOG/SA/
CPSP	0A0C	Összehasonlitás:SA-RA eredmény a Z és C inditó-torokban

LDDEHL	0902	/HL/ cimü számot /DE/ cimbe viszi /FLAGSP kell/
FLAGSP	0AEF	SA tipusfleg-ét egyszeres pontosságúra állitja
LDRAHL	09C2	/HL/ cimü számot RA-ba viszi /HL=HLJ4/
LDRASA	09BF	SA-t RA-ba viszi
LDHLSA	09CB	SA-t /HL/ cimre viszi /HL=HL+4/
LDSARA	09B4	RA-t SA-ba viszi
LDSAHL	09B1	/HL/ cimü számot SA-ba viszi
CSAASC	0FBD	SA-t decimálisra konvertálja a 4130 cimtől kezdődően,utána OUTSTR-el kiirható /előtte kell FLAGSP/
DCBNFL	ØE65	/HL/ kezdőcimü ASCII stringet /végén ØØ/ SA-ban konvertálja

C

ASCII TÁBLÁZAT

LSD \ MSD	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	0000		SP	0	@	P	`	p
1	0001	BREAK	!	1	A	Q	a	q
2	0010		"	2	B	R	b	r
3	0011		#	3	C	S	c	s
4	0100		$	4	D	T	d	t
5	0101		%	5	E	U	e	u
6	0110		&	6	F	V	f	v
7	0111		'	7	G	W	g	w
8	1000	BS	(	8	H	X	h	x
9	1001	→	)	9	I	Y	i	y
A	1010	↓	*	:	J	Z	j	z
B	1011	↑	+	;	K	[	k	{
C	1100	FF	HOME	,	<	L	\	l	|
D	1101	CR		-	=	M	]	m	}
E	1110		.	>	N	↑	n	~
F	1111		CLS	/	?	O	←	o	