SZTRÓKAY KÁLMÁN

A Z80 Assembler

HT-1080Z SZÁMÍTÓGÉPES PÉLDÁKKAL

MŰSZAKI KÖNYVKIADÓ

Sztrókay Kálmán

A Z80 Assembler
HT–1080Z számítógépes példákkal

SZTRÓKAY KÁLMÁN

A Z80 Assembler

HT-1080Z SZÁMÍTÓGÉPES PÉLDÁKKAL

MŰSZAKI KÖNYVKIADÓ, BUDAPEST, 1985

Lektorálta:

Theisz György
okl. matematika, fizika,
technika tanár

© *Sztrókay Kálmán,* Budapest, 1985
SCAN: GAFZ, 2003

ISBN: 963 10 6545 6
ETO: 800.92 assembler

Felelős szerkesztő: *Votisky Zsuzsa,* okl. matematika, fizika tanár

Kiadja a Műszaki Könyvkiadó
Felelős kiadó: Fischer Herbert igazgató

Révai Nyomda Egri Gyáregység
Felelős vezető: Horváth Józsefné dr.

Műszaki vezető: Kőrizs Károly
Műszaki szerkesztő: Bagi Miklós
A borítót és a kötést tervezte: Kováts Tibor
A könyv ábráit rajzolta: Hack Emil
A könyv formátuma: B/5
Ívterjedelme: 25,375 (A5)
Ábrák száma: 120
Papír minősége: 80 g ofszet
Betűcsalád és -méret: Press Roman, 10/11
Azonossági szám: 61325
MŰ: 3765-i-8588
A kézirat lezárva: 1984. november
Készült az MSZ 5601 és 5602 szerint

Tartalom

Bevezetés 7

1. fejezet. A Z80 mikroprocesszor 9
Bináris és hexadecimális számok 9
A mikroszámítógépek felépítése 10
A Z80 regiszterei 12
Jelzőbitek 14

2. fejezet. Adatmozgatás 17
Egybyte-os adatmozgató utasítások 17
Adatmozgatás indexregiszterekkel 19
Adatmozgatás közvetlen adattal 20
Adatmozgatás közvetlen címzéssel 22
Néhány további adatmozgató utasítás 23
Blokkmozgatás 24
Adatcserélő utasítások 28

3. fejezet. Aritmetika és logika 29
Növelés, csökkentés 29
8-bites aritmetika 31
16-bites aritmetika 33
Logikai műveletek 36

4. fejezet. Vezérlésátadás 41
Ugró utasítások 41
Szubrutinszervezés 43

5. fejezet. Eltolások és bitműveletek 47
Eltolások 47
Ciklikus eltolások 49
Bitműveletek 53

6. fejezet. Veremkezelés 57
Veremkezelő utasítások 58
Szubrutinok szervezése 60
Adattárolás, adatcsere 61
Adatátadás a szubrutinnak 63

7. fejezet. Gépi kód és assembler 65
Programozás gépi kódban 66
Programozás assemblerben 68
Pszeudo utasítások 70
Egy komolyabb program 72
Programbelövés, hibakeresés 73

8. fejezet. Adatmozgatási példák 75
Byte-ok mozgatása 75
Tárkitöltés 77
Blokkáthelyezés 79
Veremkezelés 82
9. fejezet. Aritmetikai példák 85
Egy kis aritmetika 85
Összeadás és kivonás 87
Decimális aritmetika 90
Összehasonlítás 91
Szorzás és osztás 92
10. fejezet. Egyéb példák 97
Számok és ASCII kódok 97
Karakterkeresés 98
Táblázatok 100
Rendezés 103
11. fejezet. HT-1080Z ROM-rutinok 105
Tárelrendezés 105
Még egy kis aritmetika 108
Változók mozgatása 112
Billentyűzetolvasás 114
Képernyőre írás 116
Adatkonverzió 117
Aritmetikai műveletek 119
Matematikai függvények 122
BASIC szubrutinok 125
12. fejezet. Be- és kimeneti műveletek 127
Be- és kimeneti portok 127
Kazettaműveletek 129
Adatszalag írása és olvasása 133
Nyomtatás 137
Megszakítások kezelése 138
Z80 mikroprocesszor 141
A Függelék Z80 utasításmnemonikok és jelentésük 142
Z80 utasítások hexadecimális kód szerint 146
Z80 utasítások mnemonik szerint 157
Jelzőbitek 174
B Függelék ROM-rutinok 175
RAM-táblázatok 191
C Függelék Kulcsszavak 197
LEVEL II. BASIC utasításkódok, belépési pontok 199
Aritmetikai ROM-rutinok 201
Hibaüzenetek 207
D Függelék ASCII karakterek 209
Decimális és hexadecimális számok 211
E Függelék EDI assembler/editor program 215

Bevezetés

A személyi számítógépek általánosan elterjedt nyelvének, a BASIC nyelvnek, mint a magasszintű nyelveknek általában, sok előnye van: könnyen, gyorsan megtanulható, egyszerűen alkalmazható. A különböző típusú számítógépek BASIC nyelvei között csak kisebb eltérések vannak, így a BASIC nyelv ismeretében sokféle számítógépre írhatunk programokat, és a programok – némi módosítással – átvihetők más típusú számítógépre. BASIC programok írásakor elég a legfontosabb BASIC utasításokat ismerni, a számítógép felépítéséről, működéséről szinte semmit sem kell tudni.

Miért foglalkozzunk az assembly nyelvvel, amikor a BASIC olyan egyszerű, és szinte minden feladatot meg tudunk BASIC-ban oldani? A válasz egyszerűen az, hogy az assembly nyelv a mikroprocesszor közvetlen közelsége révén olyan előnyöket nyújt, amilyeneket a magasszintű nyelvek sosem nyújthatnak.

Assemblyben programozva ugyanis a mikroprocesszor saját nyelvén írunk. Közvetlenül látjuk, hogy a mikroprocesszor milyen műveleteket végez, mit tartalmaznak a regiszterei, mi van a számítógép tárában. Az assemblyben írt program igen gyors, általában 100–300-szor gyorsabb, mint a megfelelő BASIC program. Ennek oka az, hogy a számítógépnek a BASIC program minden sorát, minden utasítását értelmeznie kell, majd a változók elhelyezésével, a megfelelő szubrutinok meghívásával meg kell szerveznie azok végrehajtását.

Gyorsasága miatt az adott számítógépen néha nincs is más választásunk, mint az assembly programozás. Pl. adatok egyidejű feldolgozása, hosszadalmas iterációk, nagyon sok adat rendezése általában csak assemblyben megoldható feladatok.

Az assembly program kevés tárat igényel, általában ötödannyit, mint a megfelelő BASIC program, tehát egy 4 Kbyte méretű assembly program 20–24 Kbyte méretű BASIC programnak felel meg. Így azt a feladatot, amelyet egy 16 Kbyte tárkapacitású számítógépen BASIC nyelven tárbővítés nélkül nem tudunk megoldani, assemblyben kényelmesen megoldhatjuk.

Az assembly program hossza nagymértékben függ a programozó ügyességétől. Minél jobban ismeri az adott mikroprocesszort, annak utasításkészletét, programozási trükkjeit, annál hatásosabb programot tud írni.

Az assembly nyelv közvetlen kapcsolatot tesz lehetővé a mikroprocesszorral. Pontosan tudjuk, hogy mit és mikor csinál a számítógép. A magasszintű nyelvek viszont elszigetelik a programozót ettől, ugyanis ott a programozó és a számítógép közötti kapcsolat létrehozása a nyelv fordítóprogramjának vagy értelmezőjének a feladata. A programozó a magasszintű nyelvvel

áll kapcsolatban, az mondja meg a mikroprocesszornak, hogy milyen utasításokat hajtson végre. Amikor egy magasszintű nyelven írt porgramot hajt végre a számítógép, akkor a nyelv fordítóprogramja vagy értelmezője határozza meg, hogy a feladat végrehajtása a mikroprocesszor utasításainak milyen sorozatával történjék.

Assembly nyelven azonban a programozónak kell megterveznie és irányítania, hogy a mikroprocesszor milyen utasításokat milyen sorrendben hajtson végre. Így assemblyben programozni lényegében annyit jelent, mint kiválasztani a mikroprocesszor utasításainak alkalmas sorozatát, amely végrehajtás során (remélhetőleg) a kívánt eredményt adja. Ehhez meg kell ismerkednünk a mikroprocesszor – könyvünkben a Z80 mikroprocesszor – működésével, utasításaival.

Az assembler programozást könnyebbé teszi, ha megismerkedünk számítógépünk – könyvünkben a HT1080Z számítógép – BASIC értelmezőjének szubrutinjaival és assembler programjainkban alkalmazzuk őket. Így ugyanis nem kell azzal foglalkoznunk, hogy olyan programokat és szubrutinokat írjunk, amelyeket mások már – valószínűleg jobban – megírtak, elég ha megtanuljuk, hogyan kell a BASIC értelmező szubrutinjait alkalmazni, és erőnket igazi feladatunk megoldására összpontosíthatjuk.

1. fejezet

A Z80 mikroprocesszor

Bináris és hexadecimális számok

A számítógép tárának tartalmát kétállású kapcsolókkal ábrázolhatjuk. A kapcsolók mindegyike bekapcsolt vagy kikapcsolt állapotban lehet. Egy tárrekesz tartalmát tehát pl.

be be be ki ki ki be ki

alakban, vagy a be helyett 1-et, a ki helyett 0-t írva

1 1 1 0 0 0 1 0

alakban, tehát kettes (bináris) számrendszerben 8-jegyű számmal ábrázolhatjuk. Példánkban a tárrekesz tartalma

$$1\cdot 2^7+1\cdot 2^6+1\cdot 2^5+0\cdot 2^4+0\cdot 2^3+0\cdot 2^2+1\cdot 2^1+0\cdot 2^0 =$$
$$= 128 + 64 + 32 + 0 + 0 + 0 + 2 + 0\,,$$

ami tízes számrendszerben 226.

A kettes számrendszerbeli számjegyeket *biteknek* nevezzük (bit az angol *bi*nary digi*t* rövidítése). Egy tárrekesz tartalmát 8-bites számmal ábrázolhatjuk – ezt *byte-nak* nevezzük.

A biteket a kitevőknek megfelelően jobbról balra sorszámozzuk:

sorszám: 7 6 5 4 3 2 1 0,
érték: 1 1 1 0 0 0 1 0.

A számokat kettes számrendszerből tízesbe – binárisból decimálissá – a fenti eljárással, decimálisból binárissá 2-vel történő ismételt osztással alakíthatjuk át, pl. így:

250	maradék: 0	ez lesz a jobb szélső bit
125	1	
62	0	
31	1	
15	1	
7	1	
3	1	
1		ez lesz a bal szélső bit

A kettes számrendszer ismerete a számítógép működésének megismeréséhez nélkülözhetetlen, viszont a kettes számrendszerben felírt számok nagyon sokjegyűek, nehezen megjegyezhetők. Ezért kettes számrendszer helyett *tizenhatos (hexadecimális, hexa)* számrendszerben szokás a tárrekeszek tar-

talmát, a tárrekeszek címét leírni. A 8-jegyű vagy 16 jegyű bináris szám számjegyeit négyesével csoportosítjuk:

$$01010001 = 0101\ 0001$$
$$0011100000001000 = 0011\ 1000\ 0000\ 1000$$

A 4-bites csoportokat bináris értéküknek megfelelően a hexadecimális számrendszer 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F számjegyeivel írhatjuk le, így a fenti két bináris szám hexadecimális alakban 51, ill. 3808. Hexadecimálisból decimálisba ugyanolyan eljárással alakíthatjuk át a számokat, mint binárisból decimálisba, így pl. a 2A45 hexadecimális szám decimális értéke

$$2 \cdot 16^3 + 10 \cdot 16^2 + 4 \cdot 16^1 + 5 \cdot 16^0$$

Íme néhány szám a három különböző számrendszerben felírva:

bináris	decimális	hexa
10	2	2
1000	8	8
1010	10	A
0001 0000	16	10
101	5	5
0010 1110	46	2E
0001 0011	19	13
1010 0000	160	A0

Egy byt-on a 0...255, két byt-on a 0...65535 decimális számokat lehet ábrázolni. Ugyanez hexadecimálisan egy byt-on 00-tól FF-ig, két byt-on 000-tól FFFF-ig. Nem könnyebb megjegyezni az FFFF-et, mint a 65535-öt? Azt, hogy a szám bináris, decimális vagy hexadecimális, a szám után írt B, D, H betűkkel szokás jelölni, így pl. 10B=2D és 10H=16D.

A számokat binárisból hexadecimálissá jegyeiket négyesével csoportosítva, hexadecimálisból binárissá a hexadecimális számjegyeket a megfelelő 4-bites bináris értékkel helyettesítve lehet átalakítani. Decimálisból hexadecimálissá ismételt osztással, hexadecimálisból decimálissá a számjegyeket 16 megfelelő hatványaival szorzva, majd a kapott szorzatokat összeadva lehet átalakítani.

A mikroszámítógépek felépítése

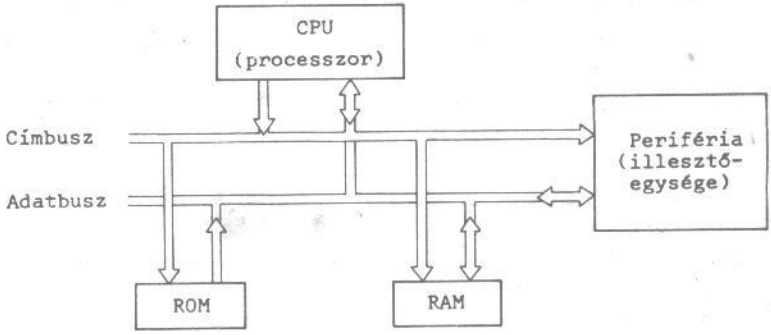
A mikroszámítógépek három alapvető részből állnak: a mikroprocesszorból, a tárból és a perifériákból.

A *mikroprocesszor (központi egység,* Central Processor Unit, CPU) a számítógép lelke, valójában ez maga a számítógép. Ez hajtja végre az utasításokat: az aritmetikai, logikai, adatmozgató, vezérlő és egyéb műveleteket, ez határozza meg, hogy az utasítások milyen sorrendben kerüljenek végrehajtásra.

A *tár* a programot vagy programokat és a különböző adatokat tárolja. A mikroprocesszor a tárból hozza be azokat az utasításokat, amelyeket

végrehajt. Az utasítások végrehajtásához szükséges adatokat a tárból vagy a perifériákról veszi, az eredményeket a tárba írja vagy perifériára küldi. A tár két alapvető típusa a csak olvasható ROM (Read Only Memory) és az írható–olvasható RAM (Random Access Memory: közvetlen elérésű tár). A ROM tartalmazza a mikroszámítógép működéséhez szükséges programokat, így elsősorban a BASIC értelmezőt. A RAM-ban helyezhetjük el programjainkat, adatainkat.

A *perifériák* a mikroszámítógép input–output eszközei, ezek segítségével kerülhetünk kapcsolatba a számítógéppel, így pl. írógép jellegű billentyű-



(Az ábrán a vezérlőbuszt nem tüntettük fel!)

1.1. ábra. Mikroszámítógépek felépítése

zeten vihetjük be programjainkat, adatainkat, parancsainkat, és olvasható formában, képernyőn vagy nyomtatón kaphatjuk vissza az eredményeket.

A mikroszámítógép részei közötti összeköttetést a Z80 mikroprocesszor adatátviteli vonalak *busznak* (sínnek) nevezett rendszerével valósítja meg. A mikroprocesszor működése egy 13 vonalból (vezetékből) álló vezérlőbuszra adott jelekkel időzíthető és vezérelhető.

A mikroprocesszor a mikroszámítógép többi részével egy 16 vonalból álló *címbusz* és egy 8 vonalból álló *adatbusz* segítségével tartja a kapcsolatot. Mindegyik vonal egy bitet szállít, így a címbusz egy tárrekesz címét, az adatbusz egy tárrekesz tartalmát szállíthatja. A címbusz egyirányú, a mikroprocesszorból kivisz egy 16 bites számot. Az adatbusz kétirányú, szállíthat egy byte-ot a mikroprocesszorból vagy a mikroprocesszorba, valójában ez biztosítja az adatáramlást a mikroprocesszor és a külvilág között.

A címbuszra helyezett 16 bites szám lényeges információkat hordoz. Ez a szám lehet egy tárrekesz címe a ROM-ban vagy a RAM-ban, vagy lehet olyan speciális cím, amely valamelyik perifériával kapcsolja össze a mikroprocesszort. Ha ilyen címet helyezünk a címbuszra, akkor a mikroprocesszor által küldött adat közvetlenül a megcímzett perifériára kerül, ha ilyen címről olvasunk, akkor a mikroprocesszor közvetlenül a perifériáról hoz be adatokat.

A Z80 mikroprocesszor másképpen is összeköthető a perifériákkal, az úgynevezett portok segítségével. Ezekről később, a megfelelő Z80 utasítások ismertetésekor lesz szó.

A Z80 regiszterei

A Z80 mikroprocesszor utasításai különböző aritmetikai, logikai és egyéb műveletek elvégzését teszik lehetővé. A műveletek adatai a tárban és a Z80 regisztereiben helyezkedhetnek el. A *regiszterek* a mikroprocesszor belső tárrekeszei. Sok művelet csak a regiszterekben levő adatokkal végezhető el, a tárban levőkkel nem. A regiszterekben levő adatok elérése gyorsabb, mint a tárban levő adatoké.

A Z80 mikroprocesszor aritmetikai és logikai műveletek céljaira szolgáló *általános regiszterei* 8-bitesek. Közülük kitüntetett szerepe van az A regiszternek, amelyet *akkumulátornak* nevezünk. A legtöbb aritmetikai és logikai műveletet csak úgy lehet végrehajtani, hogy az egyik opreandust az A regiszterben helyezzük el. A másik operandus elhelyezésére az A regiszteren kívül további hat általános célú regiszter szolgál, amelyeket B és C, D és E, H és L betűkkel jelölünk.

Az utóbbi hat általános célú regiszter párosával, 16 bites dupla regiszterekként is alkalmazható, mint BC, DE, ill. HL regiszterpár. Vannak regiszterpárokkal elvégezhető műveletek, ilyen pl. az egyszerű számlálás vagy a 16 bites egész számok összeadása.

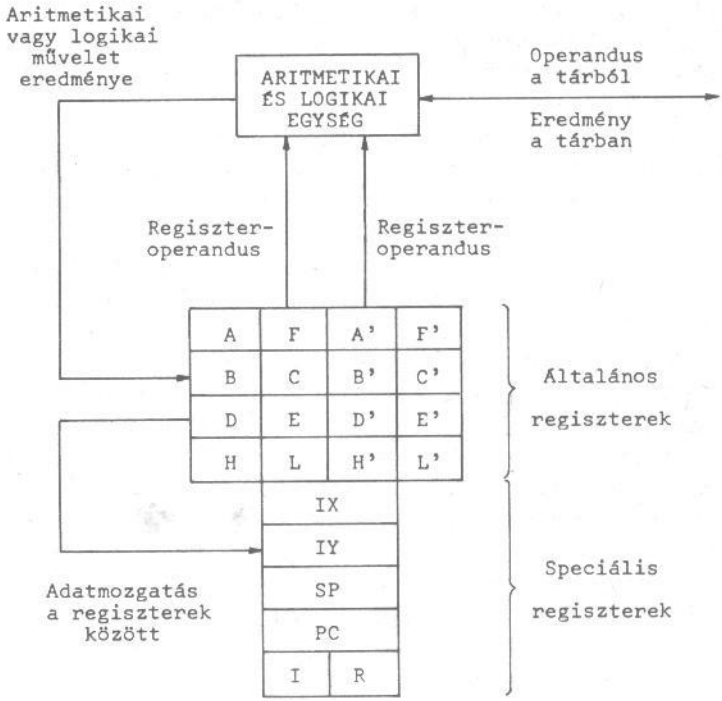
Az F regiszter jelzőbiteket tartalmaz (flag regiszter). A jelzőbitekről később részletesebben lesz szó, itt csak azért említjük, mert a Z80 mikroprocesszor a 8-bites F regisztert sokszor az ugyancsak 8-bites A regiszterrel együtt kezeli (pl. a verem műveletekben, amelyek egyszerrekét byte-ot kezelnek, a BC, DE és HL regiszterpárok mellett az AF is párban szerepel).

Az A, F, B, C, D, E, H, L nyolc regiszter kiegészül az A′, F′, B′, C′, D′, E′, H′, L′ nyolc regiszterrel, amelyek az előbbiekkel azonos funkciójúak, de a két regiszterkészlet közül egyszerre csak az egyik használható. Az aktív regiszterkészlet és a háttér-regiszterkészlet egyetlen utasítással felcserélhető, így a két regiszterkészlet lehetővé teszi a regiszterek tartalmának gyors ideiglenes tárolását, ami jelentősen csökkenti pl. a szubrutinhívásokkor szükséges regiszterkimentések végrehajtási idejét.

Ugyanazt a műveletet a Z80 általában gyorsabban tudja elvégezni a regiszterekben tárolt adatokkal, mint azokkal, amelyeket a tárból kell elővennie. Az éppen felhasználásra kerülő adatok ideiglenes tárolására tehát érdemes a regisztereket használni.

A Z80 mikroprocesszornak az eddig ismertetett általános célú regiszte-reken kívül néhány speciális regisztere is van. A 16 bites PC *programszámláló regiszter* mindig a következő végrehajtandó utasítás címét tartalmazza, ez vezérli tehát a program végrehajtásának folyamatát. Amikor egy programot elindítunk, akkor a PC regiszterbe be kell töltenünk azt a tárcímet, amelyen az első végrehajtandó utasítás található. Az utasítás végrehajtása során a PC regiszterben levő címtől beolvasásra kerül a végrehajtandó utasítás. A Z80 utasításai között vannak egy, két, három és négy byte hosszúságúak. A PC regiszter tartalma a következő végrehajtandó utasítás címére automatikusan mindig annyival növekszik, ahány byte-os volt az éppen végrehajtott utasítás.

A programszámláló közvetlenül nem érhető el, nem használható aritmetikai vagy logikai műveletek elvégzésére vagy adatok ideiglenes tárolására.



1.2. ábra. A Z80 mikroprocesszor regiszterei

A programszámláló feladata az, hogy mutassa a következő végrehajtandó utasítás címét. Ha a program végrehajtása során vezérlésátadó utasításhoz érkezünk, amelynek hatására nem a sorban következő utasítás lesz a következő végrehajtandó utasítás, akkor a programszámlálóba ez az új érték kerül, és a program végrehajtása az ezen az új címen található utasítással folytatódik.

A RAM-ban levő *veremterület* adatok, regisztertartalmak ideiglenes tárolására szolgál. Bizonyos típusú utasítások – szubrutinhívások, megszakítások – végrehajtása során a PC regiszter tartalma automatikusan kimentődik a veremterületre. A veremterület kezelése az SP regiszter (verem mutató, (stack pointer) segítségével történik. A 16 bites SP regiszter kezdeti értékének beállításával a veremterületet a RAM tetszőleges helyére helyezhetjük. A gyakorlatban a veremterületet a tár tetején szokás kijelölni, ugyanis ha a verembe adatot tárolunk, akkor az lefelé, a magasabb címektől az alacsonyabbak felé épül. A veremterületről mindig azt az adatot olvashatjuk ki először, amelyet utoljára helyeztünk oda (LIFO, Last-In-First-Out). Ha új adatot tárolunk a veremterületre, akkor SP értéke csökken, ha adatot olvasunk ki a veremből, akkor SP értéke növekszik.

Az IX és IY *indexregiszterek* segítségével indexelt címzést valósíthatunk meg. Az indexregiszterekben levő 16 bites címek ugyanis egybyte-os eltolásérték hozzáadásával módosíthatók, ami lehetővé teszi, pl. táblázatok gyors elérését. A két indexregiszter felhasználásával adatokat vihetünk át a tár egyik részéből egy másikba, ugyanis az egyik indexregiszterrel a forrásterületre,

a másikkal a célterületre mutatva egyszerűen megoldható az adatmozgatás. Ha az indexregisztereket a fenti célokra éppen nem használjuk, akkor a többi általános célú regiszterpárhoz hasonlóan adatok tárolására alkalmazhatjuk ezeket.

A Z80 mikroprocesszor a 8-bites R *tárfrissítő regisztert* (refresh register) a RAM állandó frissítésére használja. Erre azért van szükség, mert a dinamikus RAM az adatokat csak rövid ideig őrzi meg, így tartalmát annak kiolvasásával és újraírásával állandóan frissíteni kell. Ezt a Z80 a felhasználó számára észrevehetetlenül, teljesen automatikusan végzi. Az *I megszakításvektor-regisztert* a Z80 megszakítások kezelésére használja (erről a 12. fejezet végén lesz szó).

A Z80 mikroprocesszor legfontosabb része a 8-bites *aritmetikai és logikai egység.* Ebben történik a programozó által kiválasztott összeadás, kivonás, hasonlítás, logikai stb. művelet végrehajtása. Az aritmetikai és logikai egység a műveleteket két 8-bites adattal végzi, amelyek közül az egyik az A akkumulátorból érkezik, a másik jöhet az A, B, C, D, E, H, L általános regiszterek bármelyikéből vagy a tárból. A művelet eredménye az A regiszterbe kerül, felülírva annak előző tartalmát. Minthogy az aritmetikai és logikai egység 8-bites, speciális utasítások és programozási módszerek szükségesek ahhoz, hogy 16 bites, ill. hosszabb adatokkal végezhessünk műveleteket.

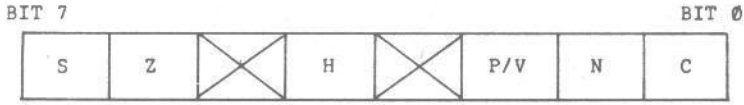
Jelzőbitek

Az *F (flag) regiszter* jelzőbiteket tartalmaz, amelyek az aritmetikai és logikai egység által végrehajtott művelet eredményének tulajdonságait mutatják. A jelzőbitek segítségével képes a Z80 mikroprocesszor döntések hozatalára, a feltételes vezérlésátadó utasítások a jelzőbitek állapotát vizsgálják, és azoknak megfelelően vagy végrehajtják a vezérlésátadást, vagy nem.

A *C (carry) jelzőbit* azt mutatja, hogy az aritmetikai művelet végrehajtása során keletkezett-e átvitel (túlcsordulás, alulcsordulás). A C jelzőbit értéke 1, ha az aritmetikai és logikai egység által végrehajtott aritmetikai művelet során volt átvitel, és 0, ha nem volt átvitel. Logikai műveletek végrehajtásakor a C jelzőbit értéke mindig 0 lesz. A C jelzőbit értéke feltételes vezérlésátadó utasításokkal vizsgálható.

Mivel a C jelzőbitet nagyon sok utasítás állítja, nehezen lehet megjósolni, hogy egy hosszabb utasítássorozat végrehajtása után mennyi lesz C értéke. Ha a program olyan pontjához érkezünk, ahol arra van szükségünk, hogy C értéke meghatározott legyen, akkor ott alkalmas utasítás segítségével beállíthatjuk a kívánt értéket.

A *Z (zero) jelzőbit* azt mutatja,hogy az aritmetikai vagy logikai művelet eredményeképpen az A akkumulátorba vagy más helyre éppen nulla került-e, ill. hogy az összehasonlított két regiszter tartalma azonos volt-e. Ha az aritmetikai vagy logikai művelet eredménye nulla, akkor a Z jelzőbit értéke 1, ha nem nulla, akkor Z értéke 0. (Figyeljük meg, hogy a kapcsolat fordított! Programíráskor gondoljunk erre!) Összehasonlító művelet után a Z jelzőbit értéke 1, ha az összehasonlított két érték azonos volt, és 0, ha különböző



1.3. ábra. Az F regiszter jelzőbitjei

volt. A bitvizsgáló utasításoknál a Z jelzőbit azt mutatja, hogy a vizsgált bit értéke nulla vagy egy – ennek megfelelően a Z jelzőbit értéke 1 vagy 0, a kapcsolat itt is fordított. A Z jelzőbit értéke feltételes vezérlésátadó utasításokkal vizsgálható. A programozó számára a jelzőbitek közül talán ez a legfontosabb.

Az *S (sign) jelzőbit* azt mutatja, hogy az aritmetikai vagy logikai művelet eredménye – kettes komplemens ábrázolásban – pozitív (S = 0) vagy negatív (S = 1), vagyis hogy az eredmény legértékesebb, bal szélső bitje 0 vagy 1. Az S jelzőbit feltételes vezérlésátadó utasításokkal vizsgálható. Programozástechnikai szempontból fontos, hogy az S jelzőbit a legértékesebb, bal szélső bit állapotát mutatja, így ennek értéke az S jelzőbit állapota alapján meghatározható.

A *P/V (parity/overflow) jelzőbitnek* kettős funkciója van. Aritmetikai műveletek végrehajtásakor túlcsordulást jelez. A kettes komplemens ábrázolású számok legértékesebb bitje a szám előjelét jelenti (pozitív számnál 0, negatív számnál 1), a többi bit tartalmazza a szám értékét. Ha két ilyen számot összeadunk vagy kivonunk, akkor előfordulhat, hogy hibás eredményt kapunk, mert az összeg vagy különbség egy byte-on, ill két byte-on nem ábrázolható. Ezt az eseményt jelzi a P/V jelzőbit túlcsordulás funkciója.

Más műveleteknél, pl. az eltolásoknál, logikai műveleteknél a túlcsordulásnak nincs jelentése. Ekkor ugyanez a jelzőbit az eredmény paritását mutatja. Ez jól használható adatok vizsgálatára, pl. perifériáról érkező adatok ellenőrzésére. Ha az eredmény páros paritású, vagyis páros számú egyes bitet tartalmaz, akkor P/V értéke 1, egyébként 0. A P/V jelzőbitet használják még a blokkmozgatás, blokkhasonlítás utasítások is. A P/V jelzőbit értéke feltételes vezérlésátadó utasításokkal vizsgálható.

A *H (half-carry) jelzőbitet* az összeadás, kivonás, növelés, csökkentés és hasonlítás utasítások állítják, ha a byte alsó négy bitjéről átvitel (túlcsordulás, alulcsordulás) van a felső négy bitre. A H jelzőbit más utasításokkal nem vizsgálható, ezt csak az aritmetikai és logikai egység használja egyes decimális aritmetikai utasítások elvégzésekor, pl. az A regiszterben levő bináris érték két decimális számjeggyé alakításához szükséges tízes igazítás elvégzéséhez.

Az *N (negativ) jelzőbitet* összeadás/kivonás vagy negatív jelzőbitnek szokták nevezni. Az N jelzőbitet a Z80 mikroprocesszor belső célra használja, ennek segítségével jegyzi meg, hogy az előző aritmetikai művelet összeadás vagy kivonás volt-e (N értéke összeadáskor 1, kivonáskor 0). Erre az összeadást vagy kivonást követő tízes igazítás utasítás végrehajtásakor van szükség. Programozói szempontból az előbb tárgyalt H jelzőbithez hasonlóan az N jelzőbitnek nincs jelentősége.

2. fejezet

Adatmozgatás

Az assembler programokban nagyon gyakran van szükség arra, hogy adatokat egy helyről – tárból, regiszterből – egy másik helyre – regiszterbe, tárba – mozgassunk, hiszen az aritmetikai, logikai műveletek elvégzéséhez az adatokat alkalmas módon el kell helyezni a regiszterekben, a műveletek elvégzése után pedig az eredményt újra a tárba kell vinni, hogy a regisztereket felszabadítsuk a következő műveletek elvégzéséhez szükséges adatok számára.

Az adatmozgatás az *LD (LoaD) utasítással** történik. Az LD utasításnak sok formája van, hiszen az adatmozgatás célja és forrása sokféle lehet. Pl. az LD A,B utasítás a B regiszter tartalmát átmozgatja az A regiszterbe, másképp szólva az A regiszterbe a B regiszter tartalmát tölti. A művelet során a forrásoperandus, a B regiszter változatlan marad, az LD utasítás valójában nem mozgatja, hanem másolja az adatokat. Még egy példa: az LD (4234H), A utasítás az A regiszter tartalmát átmozgatja a 4234H tárcímre, másképp szólva az A regiszter tartalmát a 4234H címre tárolja. Az LD utasításoknak mindig két operandusa van: az első a cél, a második a forrás. Fontos megjegyezni, hogy az LD utasítások a jelzőbiteket nem állítják.

Egybyte-os adatmozgató utasítások

Az adatmozgató utasítások közül először azokat tekintjük át, amelyek utasításkódja a tárban egy byte-ot foglal el. Ilyenek a regiszterek közötti adatmozgatást végző utasítások, valamint a regiszterek és a tár közötti adatforgalmat lebonyolító utasítások egyes csoportjai.

A regiszterből regiszterbe töltő adatmozgató utasítások a következők:

Mnemonik	Hexadecimális utasításkód
LD A,A	7F
LD A,B	78
.	.
.	.
.	.
LD A,L	7D
LD B,A	47
LD B,B	40

* A mnemonikus szótár megtalálható könyvünk függelékében.

.	.
.	.
.	.
LD L,L	6D

tehát a hét általános regiszter bármelyikének tartalmát bármelyikbe átmásolhatjuk – ez összesen 49 különböző LD utasítás. Figyeljük meg, hogy az LD A,B utasítást a B regiszter tartalmát tölti az A regiszterbe, az LD B,A utasítás pedig megfordítva, az A regiszter tartalmát tölti a B regiszterbe. A célregiszter megegyezhet a forrásregiszterrel, pl. tölthetjük az A regiszter tartalmát az A regiszterbe, de ekkor lényegében semmi sem történik – csak egy kicsit telik az idő.

A következő két utasításcsoport a regiszterek és a tár között végez adatmozgatást. Az

LD A,(HL)	7E
LD B,(HL)	46
LD C,(HL)	4E
LD D,(HL)	56
LD E,(HL)	5E
LD H,(HL)	66
LD L,(HL)	6E

utasítások a HL regiszterpár által meghatározott tárcím tartalmát töltik az adott regiszterbe. A 16 bites tárcím felső felét a H regiszter (H= high, magas), alsó felét az L regiszter (L = low, alacsony) tartalmazza. A felsorolt hét különböző LD utasítással az általános regiszterekbe bármely tárcím tartalmát betölthetjük. Mint az LD utasítások általában, ezek sem állítják a jelzőbiteket.

A fenti, regiszterbe töltő utasítások párjai a következő utasítások:

LD (HL),A	77
LD (HL),B	70
LD (HL),C	71
LD (HL),D	72
LD (HL),E	73
LD (HL),H	74
LD (HL),L	75

amelyek az adott regiszter tartalmát a tárnak arra a címére tárolják, amelyet az utasítás végrehajtásakor a HL regiszterpár tartalmaz.

Ha az utasítás egyik operandusa a tárban van, akkor annak címét általában a HL regiszterpár tartalmazza. Vannak azonban olyan utasítások is, amelyeknél a tárcím valamelyik másik regiszterpárban van. Az adatmozgató utasítások közül ilyenek az

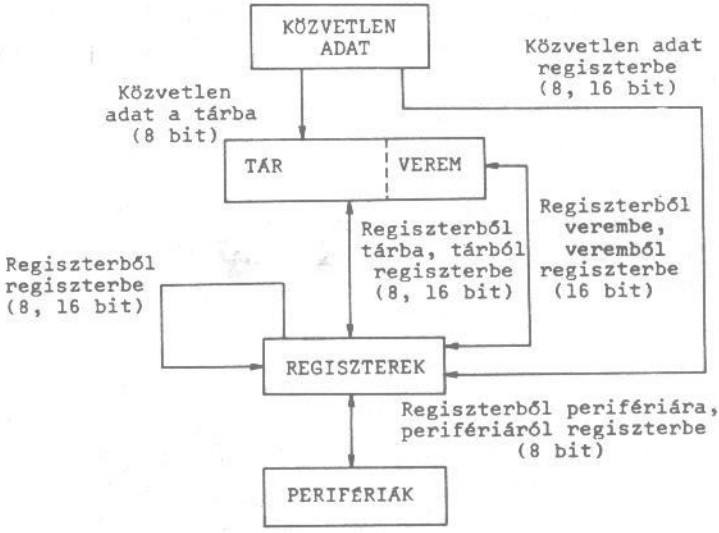
LD A,(BC)	0A
LD A,(DE)	1A

utasítások, amelyek a BC, ill. DE regiszterpár által mutatott tárcímről az A regisztert töltik, ill. az

LD (BC),A	02
LD (DE),A	12

utasítások, amelyek az A regiszter tartalmát a BC, ill. DE regiszterpárok által mutatott címekre tárolják.

Az utóbbi négy utasítás segítségével adatokat vihetünk át a tár egy részéről egy másikra oly módon, hogy az egyik regiszterpárral a célterületre, a másikkal a forrásterületre mutatunk.



2.1. ábra. Adatmozgatási lehetőségek

Mint látjuk, a HL regiszterpár által mutatott címről minden általános regiszter tölthető, ill. arra minden általános regiszter tárolható, a többi regiszterpár által mutatott címről viszont csak az A regiszter tölthető, ill. azokra csak az A regiszter tárolható.

Adatmozgatás indexregiszterekkel

A most következő LD utasításcsoport a tárcím meghatározására az IX, ill. IY indexregisztereket használja. Az operandus tárcímének kiszámítása úgy történik, hogy az indexregiszterben levő 16 bites címértékhez hozzáadódik az utasításban levő 8-bites *címzési eltolás (dis = displacement)*. Az egy byte-on, kettes komplemens ábrázolásban felírt címzési eltolás értéke –128 és +127 között lehet, így ezekkel az utasításokkal az indexregiszter adott tartalma mellett 256 egymást követő tárrekeszt érhetünk el. Az alábbi, indexregisztert használó LD utasítások három byte-osak, amiből két byte az utasításkód, a harmadik byte a címzési eltolás.

Az indexregiszterek és a címzési eltolás által meghatározott tárcím tartalmát az általános regiszterekbe töltő utasítások a következők:

LD A,(IX+dis)	DD 7E XX
LD B,(IX+dis)	DD 46 XX
LD C,(IX+dis)	DD 4E XX
LD D,(IX+dis)	DD 56 XX
LD E,(IX+dis)	DD 5E XX
LD H,(IX+dis)	DD 66 XX
LD L,(IX+dis)	DD 6E XX
LD A,(IY+dis)	FD 7E XX
LD B,(IY+dis)	FD 46 XX
LD C,(IY+dis)	FD 4E XX
LD D,(IY+dis)	FD 56 XX
LD E,(IY+dis)	FD 5E XX
LD H,(IY+dis)	FD 66 XX
LD L,(IY+dis)	FD 6E XX

ahol a hexadecimális kódban XX a címzési eltolást jelzi.

A fenti utasítások párjai a következő, az általános reigsztereket az indexregiszterek és a címzési eltolás által meghatározott tárcímre tároló utasítások:

LD (IX+dis),A	DD 77 XX
LD (IX+dis),B	DD 70 XX
LD (IX+dis),C	DD 71 XX
LD (IX+dis),D	DD 72 XX
LD (IX+dis),E	DD 73 XX
LD (IX+dis),H	DD 74 XX
LD (IX+dis),L	DD 75 XX
LD (IY+dis),A	FD 77 XX
LD (IY+dis),B	FD 70 XX
LD (IY+dis),C	FD 71 XX
LD (IY+dis),D	FD 72 XX
LD (IY+dis),E	FD 73 XX
LD (IY+dis),H	FD 74 XX
LD (IY+dis),L	FD 75 XX

Adatmozgatás közvetlen adattal

Vannak olyan adatmozgató utasítások, amelyekben a betöltendő, tárolandó adat közvetlenül az utasításkód után helyezkedik el. Ilyen pl. az LD A,1 utasítás, amely az A regiszterbe a 00000001 bináris értéket tölti. A közvetlen kifejezés arra utal, hogy az adat közvetlenül az utasításkód után helyezkedik el.

Az általános regisztereket közvetlenül töltő LD utasítások kétbyte-osak, az egybyte-os utasításkódot egybyte-os adat követi:

LD A, dd	3E dd
LD B, dd	06 dd
LD C, dd	0E dd
LD D, dd	16 dd
LD E, dd	1E dd
LD H, dd	26 dd
LD L, dd	2E dd

ahol dd az egybyte-os adatot jelenti.

Kétbyte-os közvetlen LD utasítás még a HL regiszter által mutatott címre tároló

LD (HL),dd	36 dd

utasítás, amely az utasításkódot közvetlenül követő egybyte-os adatot a HL által meghatározott tárrekeszbe tárolja.

A hexadecimális utasításkódokat figyelve láthatjuk, hogy a hét általános regisztert közvetlenül töltő utasításhoz nyolcadikként milyen szorosan illeszkedik ez a közvetlen tároló utasítás. A HL regiszterpár kitüntetett szerepét mutatja, hogy a többi regiszterpár által meghatározott tárcímre nem lehet közvetlen adatot tárolni, ezt a gyors tárba írási lehetőséget csak a HL regiszterpár biztosítja.

A BC, DE, HL regiszterpárokat és az SP regisztert tölthetjük közvetlen adattal. A 16 bites regiszterpárok töltéséhez szükséges adat két byte-on helyezhető el, így ezek az utasítások az egybyte-os utasításkóddal együtt hárombyte-osak:

LD BC, dddd	01 dd dd
LD DE, dddd	11 dd dd
LD HL, dddd	21 dd dd
LD SP, dddd	31 dd dd

Az utasításkódot közvetlenül követő kétbyte-os adat első byte-ja a regiszterpár alsó felébe (a C, E, ill. L regiszterbe), a második byte a regiszterpár felső felébe (a B, D, ill. H regiszterbe) kerül.

Az indexregisztereket közvetlen adattal töltő utasítások kódja kétbyte-os, amelyet kétbyte-os adat követ:

LD IX,dddd	DD 21 dd dd
LD IY,dddd	FD 21 dd dd

Az utasításkódot követő első byte az indexregiszter alsó felébe, a második byte az indexregiszter felső felébe kerül.

Egybyte-os adatot indexelt címzés segítségével is tárolhatunk a tárba, mégpedig az

LD (IX+dis), dd	DD 36 XX dd
LD (IY+dis), dd	FD 36 XX dd

utasításokkal. A négybyte-os utasítás első két byte-ja az utasításkód, a harmadik byte a címzési eltolás, a negyedik byte a közvetlen adat.

Adatmozgatás közvetlen címzéssel

Az adatmozgató utasítások következő csoportjában közvetlenül az utasításkód után helyezkedik el annak a tárrekesznek a címe, amelyről vagy amelyre az adatmozgatás történik – erre utal a közvetlen címzés elnevezés. Ilyen pl. a fejezet elején példaként szereplő LD (4234H),A utasítás, amely az A regiszter tartalmát a 4234H című tárrekeszbe tárolja, vagy az LD A,(4234H) utasítás, amely az A regisztert a 4234H tárcímről tölti. Az általános regiszterek közül ilyen módon csak az A akkumulátort lehet tölteni. Az

LD A,(addr) 3A XX XX

utasítás az addr tárcímről tölti az A regisztert. A hárombyte-os utasítás első byte-ja az utasításkód, a második byte a forrás tárrekeszcímének alsó fele, a harmadik byte a cím felső fele. Az

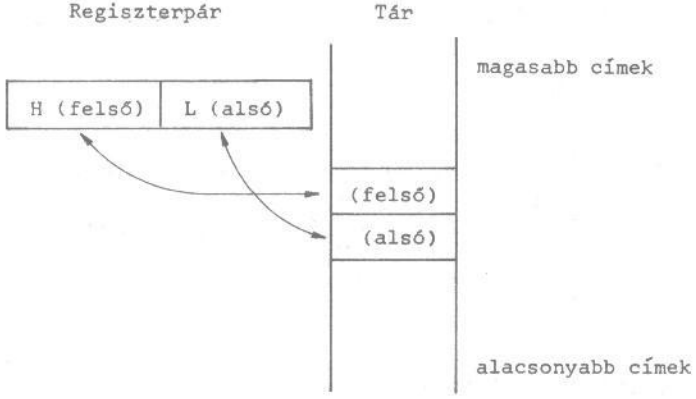
LD (addr),A 32 XX XX

utasítás az A regiszter tartalmát tárolja az addr tárcímre. Az előző utasításhoz hasonlóan az első byte az opreanduskód, a második byte a cél tárrekeszcímének alsó fele, a harmadik byte a cím felső fele.

A további közvetlen címzésű LD utasítások kétbyte-os adatokat mozgatnak. Az

LD HL,(addr) 2A XX XX

utasítás a HL regiszterpárba tölt adatot. Az utasításkód utáni két byte-on annak a tárrésznek a kezdőcíme található – előbb a cím alsó fele, majd a cím felső fele –, amelyről az utasítás a HL regiszterpárba adatot tölt. Az így meg-



2.2. ábra. 16 bites adatok mozgatása

határozott tárcím tartalma az L regiszterbe, a következő tárcím tartalma a H regiszterbe kerül.

Ne tévesszük össze ezt az utasítást az előbb tárgyalt LD HL,dddd utasítással, amely az utasításkód utáni két byte tartalmát tölti a HL regiszterpárba, míg az LD HL,(addr) utasítás az addr által meghatározott tárcímről tölt.

Az előbbi utasítás párja az

LD (addr),HL 22 XX XX

utasítás, amely a HL regiszter tartalmát a tár adott címétől tárolja. Az utasításkód utáni két byte által meghatározott címre az L regiszter tartalma, a következő címre a H regiszter tartalma kerül.

Következzék a négybyte-os LD utasításcsoport, ami tárcímről regiszterpárt tölt:

LD BC,(addr)	ED 4B XX XX
LD DE,(addr)	ED 5B XX XX
LD HL,(addr)	ED 6B XX XX
LD SP,(addr)	ED 7B XX XX

A kétbyte-os utasításkód után a tárcím alsó fele, majd felső fele következik. Az így meghatározott tárcímről a regiszterpár alsó felébe (a C, E, ill. L regiszterbe), a következő tárcímről a regiszterpár felső felébe (a B, D, ill. H regiszterbe) kerül az adat.

A Z80 mikroprocesszor érdekessége, hogy ebben az utasításcsoportban újra szerepel az LD HL,(addr) utasítás. Az előző, hárombyte-os utasítás kevesebb tárat foglal el, és végrehajtási ideje is rövidebb, mint a most szereplő négybyte-os utasításé. Ha választhatunk, nyilván az előbbi, hárombyte-os változatot választjuk – nehezen lehetne olyan assembler fordítóprogramot találni, amely a négybyte-os változatot állítja elő, csupán érdekes, hogy így tervezték a Z80 mikroprocesszort.

A tárcímről regiszterpárt töltő utasítások párjai a regiszterpárt tárcímre tároló

LD (addr),BC	ED 43 XX XX
LD (addr),DE	ED 53 XX XX
LD (addr),HL	ED 63 XX XX
LD (addr),SP	ED 73 XX XX

utasítások, amelyek a regiszterpár alsó regiszterének tartalmát az addr címre, felső regiszterének tartalmát az addr+1 címre tárolják. A HL regiszterpárt tároló utasítás helyett most is az előbb tárgyalt hárombyte-os változat alkalmazását ajánljuk.

Néhány további adatmozgató utasítás

A következő LD utasítások az SP veremmutató regisztert töltik. Az

LD SP,HL	F9

utasítás a HL regiszterpár tartalmát – általában a tár egy címét – tölti az SP regiszterbe, az

LD SP,IX	DD F9
LD SP,IY	FD F9

utasítások pedig az indexregiszterek tartalmát töltik az SP regiszterbe.

A további LD utasításokra felhasználói programok írásakor ritkán van szükség, inkább csak a teljesség kedvéért említjük meg őket. Az

LD I,A	ED 47
LD R,A	ED 4F

utasításokat, amelyek mint az eddigi adatmozgató utasítások mindegyike, változatlanul hagyják a jelzőbiteket. A következő két LD utasítás viszont állítja a jelzőbiteket. Az

LD A,I ED 57

utasítás az A regiszterbe tölti az I regiszter tartalmát. A Z és S jelzőbitek a szokásos módon mutatják, hogy milyen érték került az A regiszterbe, a H és N jelzőbitek törlődnek, a P/V jelzőbit a nem maszkolható megszakítást* engedélyező kapcsoló állapotát mutatja, a C jelzőbit változatlan marad. Az

LD A,R ED 5F

utasítás az A regiszterbe az R regiszter tartalmát tölti, a jelzőbiteket ugyanúgy állítja, mint az előző utasítás. Mivel az R regiszter tartalma állandóan változik, ezért ennek az utasításnak a hatására az A regiszterbe véletlenszerűen más-más érték kerülhet.

Blokkmozgatás

Az eddigi adatmozgató utasítások mindegyike egyetlen műveletet hajtott végre, egy egybyte-os vagy kétbyte-os adatot mozgatott át a forrásterületről a célterületre. A Z80 mikroprocesszor blokkmozgató utasításai ennél sokkal többet tudnak, ismételt adatmozgatást és több kiegészítő tevékenységet végeznek. Ezekkel az utasításokkal egyszerűen, gyorsan lehet a tár egy részét egy másik részre átmozgatni. A mozgatás közben más műveleteket is lehet végezni – lehet pl. az adatmozgatást egy adott karakter megjelenéséig folytatni.

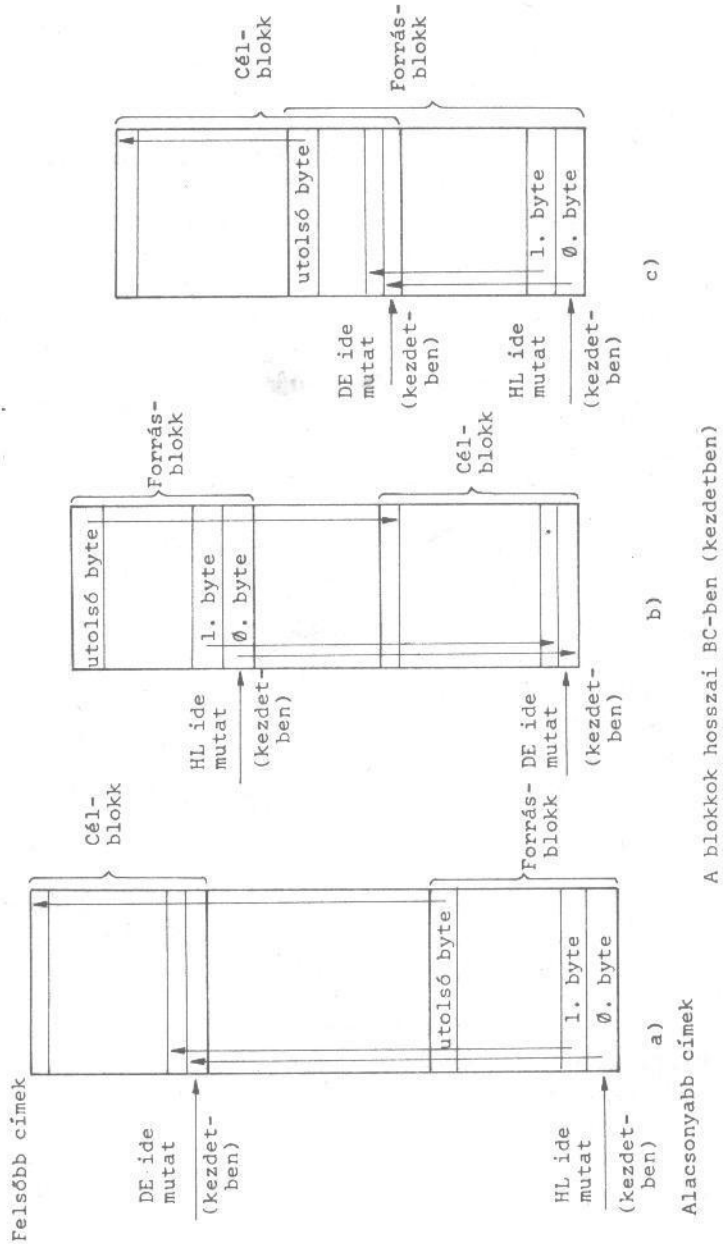
A blokkmozgató utasítások három regiszterpárban várják a végrehajtásukhoz szükséges adatokat: a HL regiszterpárban a forrástárcímet – az átmozgatandó első byte címét –, a DE regiszterpárban a célcímet – amelyre az átmozgatott első byte kerül – és a BC-ben a mozgatandó byte-ok számát (DE = destination, cél, BC = byte count, byte-szám).

Az

LDIR ED B0

utasítás (block LoaD Increment address auto-Repeat) a HL regiszterpár által mutatott tárrekesz tartalmát átmozgatja a DE regiszterpár által mutatott tárrekeszbe, majd a HL és DE regiszterek tartalmát eggyel növeli, a BC regiszter tartalmát pedig eggyel csökkenti. Ha BC tartalma nem nulla, akkor a PC programszámláló értékét 2-vel csökkenti, és így az LDIR utasítás újra végrehajtódik. Az ismétlés addig tart, amíg a BC regiszter tartalma a csökkentés után nulla lesz (ha BC-be kezdeti értéknek nullát töltünk, akkor az LDIR utasítás 65536-szor hajtódik végre). Az LDIR változatlanul hagyja a C jelzőbitet, a H, N és P/V jelzőbiteket pedig törli. Az LDIR utasítás végrehajtása közben a Z80 mikroprocesszor megszakításkérést az egyes byte-ok átmozgatása – a BC regiszter csökkentése és vizsgálata – után fogad el, így ez az uta-

* Ez a minden körülmény között létrejövő megszakításkiszolgálást jelenti.



A blokkok hosszai BC-ben (kezdetben)

2.3. ábra. Adatmozgatás LDIR utasítással

sítás többször is megszakítható, és alkalmas megszakítást kezelő rutin végrehajtása után folytatható.

Az

LDDR ED B8

utasítás (block LoaD Decrement address auto-Repeat) lényegében ugyanúgy működik, mint az LDIR utasítás, azzal a különbséggel, hogy a HL és DE regiszterpárok által tartalmazott tárcímeket eggyel csökkenti. Ezért míg az LDIR utasítás előtt a HL és DE regiszterpárokat a forrás- és céltárterületek kezdetére kell állítani, az LDDR utasításnál ugyanezek a tárterületek végére kell, hogy mutassanak. Az LDIR utasítással tárcímek szerinti növekvő sorrendben, az LDDR utasítással pedig csökkenő sorrendben történik az adatmozgatás. Az LDDR utasítás ugyanúgy állítja a jelzőbiteket, mint az LDIR utasítás.

Az LDI és LDD utasítások az előbbiekhez hasonlítanak azzal a lényeges különbséggel, hogy a PC programszámláló értékét nem csökkentik, így az LDI, ill. LDD után a következő utasítás végrehajtása következik. Ez lehetőséget nyújt az átmozgatott adatok egyenkénti feldolgozására, ami után az adatmozgatás folytatása céljából vissza lehet ugrani az LDI, ill. LDD utasításra, amihez persze meg kell őrizni a HL, DE és BC regiszterpárok tartalmát. Az

LDI ED A0

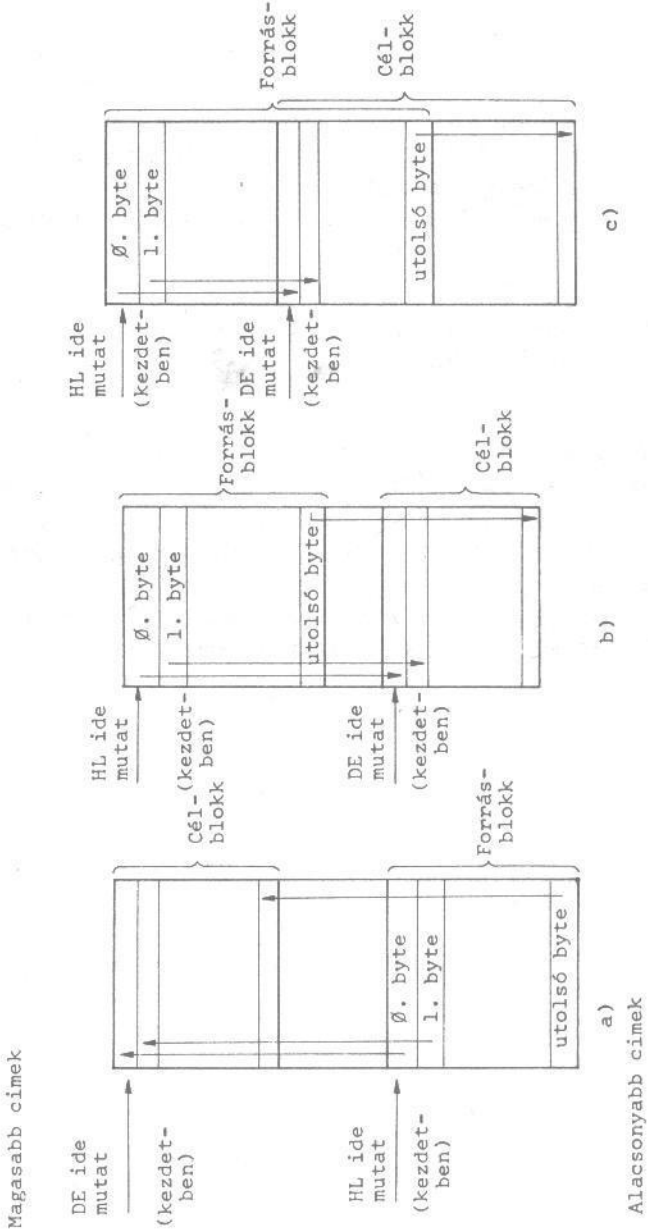
utasítás (block LoaD Increment address) ugyanúgy, mint az LDIR utasítás, a HL regiszterpár által mutatott tárrekesz tartalmát átmozgatja a DE regiszterpár által mutatott tárrekeszbe, majd a HL és DE regiszterpárok tartalmát eggyel növeli, a BC regiszterpár tartalmát eggyel csökkenti, viszont nem csökkenti a PC programszámláló értékét, és másképp állítja a jelzőbiteket. Az LDI változatlanul hagyja az S, Z és C jelzőbiteket, törli a H és N jelzőbiteket. A P/V jelzőbit a BC számláló tartalmát mutatja; ha a művelet befejezésekor a BC regiszterpár tartalma nulla, akkor a P/V jelzőbit törlődik, egyébként P/V értéke 1. Az LDI utasítás nem ismétlődik automatikusan, arra a P/V jelzőbitet vizsgáló feltételes vezérlésátadó utasítással vagy más módon vissza kell ugrani.

Az

LDD ED B8

utasítás (block LoaD Decrement address) lényegében ugyanúgy működik, mint az LDI utasítás, azzal a különbséggel, hogy a HL és DE regiszterpárok által tartalmazott tárcímeket eggyel csökkenti. Az LDD utasítás ugyanúgy állítja a jelzőbiteket, mint az LDI utasítás.

A blokkmozgató utasítások jelentősen egyszerűsítik az olyan programok írását, amelyek a tárban gyakran helyeznek át adatokat. A BC regiszterpárban elhelyezett byte-szám automatikusan vagy feltételes vezérlésátadó utasításokkal vizsgálható, az adatmozgatás a kívánt számú byte átmozgatásáig vagy más feltétel bekövetkezéséig folytatható.



A blokkok hosszai BC-ben kezdetben

2.4. ábra. Adatmozgatás LDDR utasitással

Adatcserélő utasítások

Az eddig tárgyalt adatmozgató utasítások az adatforrást változatlanul hagyva mozgatták a célterületre az adatot, az adatcserélő utasítások viszont felcserélik.

Az

EX DE,HL EB

utasítás (EXchange) felcseréli a DE és HL regiszterpárok tartalmát, a jelzőbiteket változatlanul hagyja. Jól alkalmazható pl. akkor, ha a HL regiszterpárral váltakozva két tárterületre kívánunk mutatni.

A következő utasítások a két általános regiszterkészlet cserélését biztosítják. Az

EX AF,AF′ Ø8

utasítás az A és F regisztereket felcseréli az A′ és F′ regiszterekkel. A jelzőbitek az F regiszter új tartalmát mutatják. Az

EXX D9

utasítás (EXchange) a BC, DE és HL regiszterpárokat sorra felcseréli a BC′, DE′, és HL′ regiszterpárokkal, a jelzőbiteket változatlanul hagyja. Megjegyezzük, hogy a két regiszterkészlet teljes cseréjéhez mindkét utasításra szükség van. Ez rugalmasabb programozást tesz lehetővé pl. megszakítás kezelésekor, amikor nem kell az összes általános regisztert kimenteni.

A további adatmozgató PUSH, POP és EX utasítások a veremkezeléssel kapcsolatosak, ezért azokat a Veremkezelés című fejezetben fogjuk tárgyalni.

3. fejezet

Aritmetika és logika

Ebben a fejezetben az aritmetikai és logikai egység által végrehajtott növelés, csökkentés, összeadás, kivonás, összehasonlítás utasításokat és a logikai utasításokat tárgyaljuk. Ezek az utasítások általában állítják a jelzőbiteket. A 8-bites összeadás, kivonás és a logikai műveletek két operandusa közül az első forrása mindig az A regiszter, és a művelet eredménye is mindig az A regiszterbe kerül.

Növelés, csökkentés

A növelő és csökkentő utasítások egyszerű számlálást végeznek, eggyel növelik, ill. csökkentik egy regiszter, regiszterpár vagy tárrekesz tartalmát. Jól használhatók pl. ciklusban számlálásra vagy táblázatok feldolgozásakor az egymás utáni byte-ok elérésére. Az

INC A	3C
INC B	04
INC C	0C
INC D	14
INC E	1C
INC H	24
INC L	2C

utasításokkal (INCrement) az általános regiszterek bármelyikéhez hozzáadhatunk egyet, így bármelyik regisztert használhatjuk pl. események számlálására. Az INC utasítások a Z, S és H jelzőbiteket állítják, a C jelzőbitet változatlanul hagyják, az N jelzőbitet törlik. Így a regiszter növelése után nulla vagy negatív regisztertartalom szerinti feltételes vezérlésátadó utasítás következhet. A P/V jelzőbit a kettes komplemens ábrázolással összhangban mindig törlődik, kivéve, ha túlcsordulás van, vagyis ha a 01111111 bináris értéket növeljük, amely utóbbi esetben P/V értéke 1 lesz. A H jelzőbit 1, ha a 3. bitről van átvitel, 0 egyébként.

A felsorolt hét, általános regisztert növelő utasításhoz nyolcadikként szorosan illeszkedik az

INC (HL)	34

utasítás, amely a HL regiszterpár által mutatott tárrekesz tartalmát növeli. Így lehet pl. a tárban elhelyezett számláló értékét növelni anélkül, hogy azt betöltenénk egy regiszterbe. Az INC (HL) utasítás ugyanúgy állítja a jelzőbiteket, mint az előző INC utasítások .

A

DEC A	3D
DEC B	05
DEC C	0D
DEC D	15
DEC E	1D
DEC H	25
DEC L	2D

utasítások (**DEC**rement) az INC utasításhoz hasonlóan működnek, de eggyel csökkentik az adott általános regiszter tartalmát. A jelzőbiteket ugyanúgy állítják, mint az INC utasítások. A hét, általános regisztert csökkentő utasításhoz ismét szorosan illeszkedik a

DEC (HL)	35

utasítás, amely a HL regiszterpár által mutatott tárrekesz tartalmát csökkenti, a jelzőbiteket ugyanúgy állítja, mint az előző INC és DEC utasítások.

Az INC (HL) és DEC (HL) utasításokhoz hasonlóak az

INC (IX+dis)	DD 34 XX
INC (IY+dis)	FD 34 XX
DEC (IX+dis)	DD 35 XX
DEC (IY+dis)	FD 35 XX

hárombyte-os utasítások, amelyek az IX, ill. IY indexregiszterek és az utasításban szereplő címzési eltolás által meghatározott tárrekesz tartalmát növelik, ill. csökkentik. A jelzőbiteket ugyanúgy állítják, mint az előző INC és DEC utasítások.

A további növelő és csökkentő utasítások 16 bites regiszterpárokat kezelnek, a jelzőbiteket nem állítják. Az

INC BC	03
INC DE	13
INC HL	23

utasítások a BC, DE, ill. HL regiszterpárok tartalmát mint 16 bites számot, eggyel növelik, a

DEC BC	0B
DEC DE	1B
DEC HL	2B

utasítások pedig eggyel csökkentik a BC, DE, ill. HL regiszterpárok tartalmát.

Az SP verem mutató tartalmát lehet növelni, ill. csökkenteni az

INC SP	33
DEC SP	3B

utasításokkal, végül az IX és IY regiszterek tartalma az

INC IX	DD 23
INC IY	FD 23
DEC IX	DD 2B
DEC IY	FD 2B

utasításokkal növelhető, ill. csökkenthető. Jegyezzük meg, hogy 16 bites INC és DEC utasítások a 8-bitesekkel ellentétben nem állítják a jelzőbiteket.

8-bites aritmetika

A mos következő aritmetikai utasítások első operandusa mindig az A regiszterből, második operandusa az általános regiszterek valamelyikéből vagy a tárból kerül az aritmetikai és logikai egységbe, amely a művelet eredményét mindig az A regiszterbe viszi, felülírva ezzel annak előző tartalmát. Az aritmetikai műveletek állítják a jelzőbiteket.

Az

ADD A,A	87
ADD A,B	80
ADD A,C	81
ADD A,D	82
ADD A,E	83
ADD A,H	84
ADD A,L	85

utasítások (**ADD**) az A regiszter tartalmához binárisan *hozzáadják* a második operandusként megadott általános regiszter tartalmát. A művelet eredménye az A regiszterbe kerül. A második operandus lenet maga az A regiszter, ami gyors lehetőség a 2-vel szorzásra. A második operandusként szereplő regiszter tartalma változatlan marad, kivéve természetesen, ha az maga az A regiszter. Az ADD utasítások az N jelzőbitet törlik, a többit aszerint állítják, hogy a művelet eredményeképpen mi lett az A regiszter tartalma. Az

ADC A,A	8F
ADC A,B	88
ADC A,C	89
ADC A,D	8A
ADC A,E	8B
ADC A,H	8C
ADC A,L	8D

utasítások (**AD**d with **C**arry) ugyanúgy bináris összeadást végeznek, mint az előző ADD utasítások, de az összeadási művelet során a második operandusként szereplő általános regiszter tartalmán kívül a C jelzőbit értékét – az átvitelt – is hozzáadják az A regiszter tartalmához. Ez felhasználható pl. többszörös pontosságú aritmetikai rutinok írására, ahogy ezt a 9. fejezetben bemutatjuk. A második operandusként szereplő általános regiszter az ADC A,A eset kivételével változatlan marad. Az ADC utasítások ugyanúgy állítják a jelzőbiteket, mint az ADD utasítások.

Az ADD összeadó utasítások párjai a

SUB A	97
SUB B	90
SUB C	91

SUB D	92
SUB E	93
SUB H	94
SUB L	95

utasítások (**SUB**tract), amelyek az A regisztert tartalmából binárisan kivonják az operandusként szereplő általános regiszter tartalmát. A kivonás eredménye az A regiszterbe kerül felülírva annak előző tartalmát. Az operandusként megadott általános regiszter tartalma változatlan marad, kivéve természetesen a SUB A esetet, amely egyébként egyszerű lehetőség arra, hogy nullát töltsünk az A regiszterbe. A SUB utasítások az N jelzőbitet 1-re állítják, a többi jelzőbit pedig azt mutatja, hogy a művelet eredményeképpen milyen érték került az A regiszterbe. Az

SBC A,A	9F
SBC A,B	98
SBC A,C	99
SBC A,D	9A
SBC A,E	9B
SBC A,H	9C
SCB A,L	9D

utasítások (**S**u**B**tract with **C**arry) ugyanúgy bináris kivonást végeznek, mint az előbb tárgyalt SUB utasítások, de a művelet során az A regiszter tartalmából a második operandusként szereplő regiszter tartalmán kívül a C jelzőbit értékét is kivonják. Ez felhasználható többszörös pontosságú aritmetikai rutinok írására, ugyanis C értéke az előző kivonásból származó alulcsordulást – beváltást – jelzi. A második operandusként szereplő általános regiszter az SBC A,A eset kivételével változatlan marad. Az SBC utasítások ugyanúgy állítják a jelzőbiteket, mint a SUB utasítások.

További aritmetikai utasítások az A regiszterrel és közvetlen adattal bináris összeadást, ill. kivonást végző

ADD A,dd	C6 dd
ADC A,dd	CE dd
SUB dd	D6 dd
SBC A,dd	DE dd

utasítások, az A regiszter és a HL regiszterpár által mutatott tárrekesz tartalmával bináris összeadást, ill. kivonást végző

ADD A,(HL)	86
ADC A,(HL)	8E
SUB (HL)	96
SBC A,(HL)	9E

utasítások, és végül az indexelt címzésű

ADD A,(IX+dis)	DD 86 XX
ADD A,(IX+dis)	FD 86 XX
ADC A,(IX+dis)	DD 8E XX
ADC A,(IY+dis)	FD 8E XX

SUB (IX+dis)	DD 96 XX
SUB (IY+dis)	FD 96 XX
SBC A,(IX+dis)	DD 9E XX
SBC A,(IY+dis)	FD 9E XX

utasítások, amelyek a jelzőbiteket mind ugyanúgy állítják, mint az előzőkben tárgyalt ADD, ADC, SUB és SBC utasítások.

Az összehasonlító utasítások lényegében ugyanúgy működnek, mint a SUB kivonás utasítások, azzal a különbséggel, hogy változatlanul hagyják az A regiszter tartalmát. Az összehasonlítás során tehát az aritmetikai és logikai egység az A regiszter tartalmából kivonja az operandusként megadott regiszter vagy tárrekesz tartalmát, az eredményt nem írja vissza az A regiszterbe, de állítja a jelzőbiteket. Így pl. ha az összehasonlított két adat egyenlő, akkor a kivonás eredménye nulla, így a Z jelzőbit értéke 1. Az összehasonlítást rendszerint a jelzőbitek állapota szerinti feltételes vezérlésátadás követi. A

CP A	BF
CP B	B8
CP C	B9
CP D	BA
CP E	BB
CP H	BC
CP L	BD

utasítások (ComPare) az A regiszter tartalmát sorra az általános regiszterek tartalmával hasonlítják össze. Megjegyezzük, hogy a CP A utasításnál a kivonás eredménye mindig nulla, a regiszterek tartalma nem változik, így ez az utasítás jól alkalmazható a Z jelzőbit 1-re állításához. A felsorolt hét CP utasításhoz nyolcadikként szorosan illeszkedik az A regiszter tartalmát a HL által mutatott tárrekesz tartalmával összehasonlító

CP (HL)	BE

utasítás. További összehasonlító utasítások a közvetlen adattal hasonlító

CP dd	FE dd

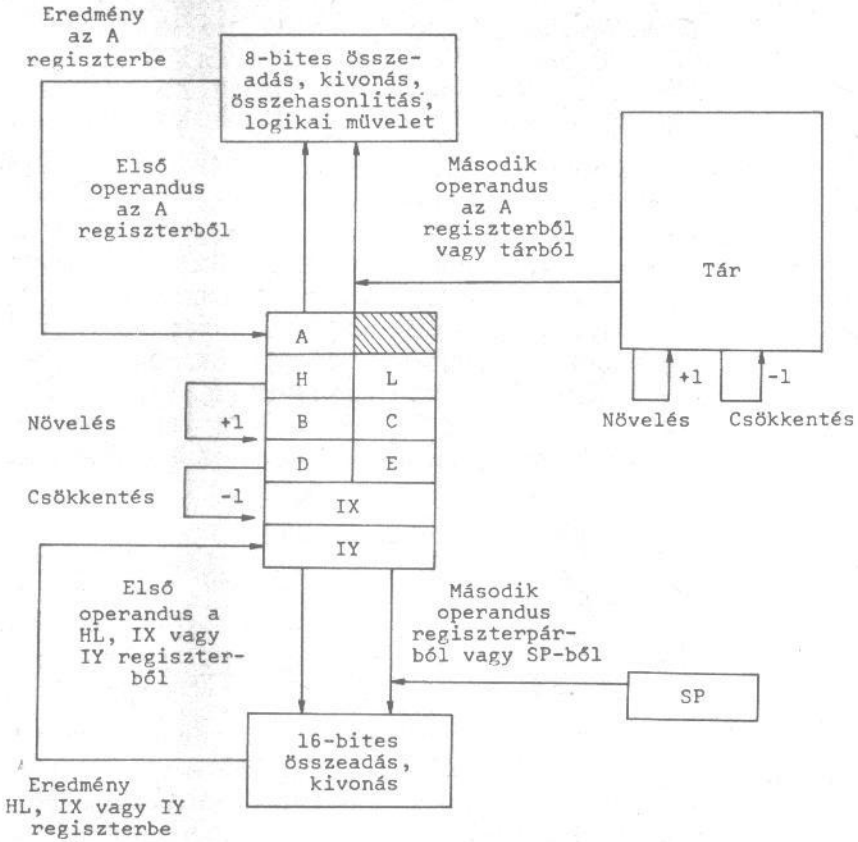
utasítás és az indexelt címzésű

CP (IX+dis)	DD BE XX
CP (IY+dis)	FD BE XX

utasítások, amelyek mind ugyanúgy működnek, mint az előző CP utasítások.

16 bites aritmetika

A következő utasítások regiszterpárok és 16 bites regiszterek körében végeznek összeadást és kivonást. Mint látni fogjuk, a 16 bites aritmetikai lehetőségek erősen korlátozottak, nem lehet a tárban levő adatokkal összeadást, kivonást végezni, nincs 16 bites hasonlítás, a meglevő néhány utasítás nem minden jelzőbitet állít.



3.1. ábra. Aritmetikai, összehasonlító és logikai műveletek

A 16 bites aritmetikai utasításokban kitüntetett szerepe van a HL regiszterpárnak, az utasítások nagy részében a HL regiszterpár tartalmazza az első operandust, és a művelet eredménye is a HL regiszterpárba kerül. Az

ADD HL,BC	09
ADD HL,DE	19
ADD HL,HL	29
ADD HL,SP	39

egybyte-os utasítások végrehajtása során az aritmetikai és logikai egység a HL regiszterpár tartalmához hozzáadja a második operandusként megadott regiszterpár, ill. 16 bites regiszter tartalmát, majd a művelet eredményét a HL regiszterpárba helyezi. Ezzel az utasítással lehet pl. a HL regiszterpárban tárolt tárcímhez címzési eltolást hozzáadni. A felsorolt utasítások a C jelzőbitet 1-re állítják, ha a 15. bitről (a regiszterpár felső regiszterének 7. bitjéről) van átvitel, a H jelzőbitet 1-re állítják, ha a 11. bitről (a felső regiszter 3. bitjéről) van átvitel, az N jelzőbitet pedig törlik. A többi jelzőbitet ezek az

utasítások nem állítják, így a 16 bites összeadás után a jelzőbitek vizsgálatával nem lehet eldönteni, hogy az eredmény nulla vagy negatív volt-e. Az

ADC HL,BC	ED 4A
ADC HL,DE	ED 5A
ADC HL,HL	ED 6A
ADC HL,SP	ED 7A

utasítások ugyanúgy összeadást végeznek, mint az előző ADD utasítások, de a művelet során a HL regiszterhez a második operandusként szereplő regiszterpár, ill. 16 bites regiszter tartalmán kívül a C jelzőbit értékét is hozzáadják, így ezek az utasítások a 8-bites ADC utasításokhoz hasonlóan felhasználhatók többszörös pontosságú aritmetikai rutinok írására. Ezek az utasítások az összes jelzőbitet állítják. Az N jelzőbitet törlik, a többi jelzőbitet a művelet eredménye szerint állítják. Megjegyezzük, hogy a C jelzőbit a 15. bitről való átvitelt, a H jelzőbit a 11. bitről való átvitelt mutatja. A felsorolt ADC utasítások párjai az

SBC HL,BC	ED 42
SBC HL,DE	ED 52
SBC HL,HL	ED 62
SBC HL,SP	ED 72

utasítások a HL regiszter tartalmából kivonják a C jelzőbit értékét, továbbá a második operandusként megadott regiszterpár, ill. 16 bites regiszter tartalmát. Ez felhasználható többszörös pontosságú aritmetikai rutinok írására, ugyanis a C értéke az előző kivonásból származó alulcsordulást – beváltást – jelzi. A kivonás eredménye a HL regiszterpárba kerül, a másik operandus tartalma nem változik. Az SBC utasítások az N jelzőbitet 1-re állítják, a többi jelzőbit pedig azt mutatja, hogy a művelet eredményeképpen milyen érték kerül a HL regiszterpárba. A C jelzőbit a 15. bitről (a H regiszter 7. bitjéről) származó alulcsordulást, a H jelzőbit a 12. bitről (a H regiszter 4. bitjéről) származó beváltást jelzi. Megjegyezzük, hogy a 16 bites aritmetikában nincsenek SUB utasítások, csak SBC utasítások.

A további 16 bites aritmetikai utasítások az indexregiszterek tartalmához ADD utasítással regiszterpár, ill. 16 bites regiszter tartalmát adják hozzá. Az IX indexregiszter tartalmát az

ADD IX,BC	DD 09
ADD IX,DE	DD 19
ADD IX,IX	DD 29
ADD IX,SP	DD 39

utasításokkal, az IY indexregiszter tartalmát pedig az

ADD IY,BC	FD 09
ADD IY,DE	FD 19
ADD IY,IY	FD 29
ADD IY,SP	FD 39

utasításokkal módosíthatjuk. Ezek az utasítások, amelyekkel egy lépésben nagyobb értékkel lehet növelni vagy – kettes komplemens értékekkel –

csökkenteni az indexregiszterek tartalmát, felhasználhatók pl. táblázatok adatainak gyors elérésére. Megjegyezzük, hogy a fenti utasításokban az indexregiszter játssza azt a kitüntetett szerepet, ami a korábbi 16 bites aritmetikai utasításokban a HL regiszteré volt, így nincs olyan utasítás, amelynek második operandusa a HL regiszterpár, továbbá nincs olyan utasítás sem, amelynek két operandusa a két indexregiszter. A felsorolt utasítások ugyanúgy, mint az előző 16 bites ADD utasítások, állítják a H és C jelzőbiteket, törlik az N jelzőbitet, nem állítják az S, Z és P/V jelzőbiteket.

Logikai műveletek

A logikai ÉS, VAGY, kizáró VAGY műveletek igazságtáblázata, ahol a logikailag hamis értéket 0, az igaz értéket 1 jelöli. Így pl. a 8-bites 11001010 és 00011110 számokkal végzett logikai műveletek eredménye a következő:

```
      11001010          11001010          11001010
AND   00011110     OR   00011110     XOR  00011110
---------------   ---------------   ---------------
      00001010          11011110          11010100
```

ahol AND a logikai ÉS, OR a logikai VAGY és XOR a logikai KIZÁRÓ VAGY.

3.1. táblázat.
Logikai műveletek igazságtáblázata

ÉS	h	i
h	h	h
i	h	i

VAGY	h	i
h	h	i
i	i	i

kizáró VAGY	h	i
h	h	i
i	i	h

AND	0	1
0	0	0
1	0	1

OR	0	1
0	0	1
1	1	1

XOR	0	1
0	0	1
1	1	0

Az aritmetikai és logikai egység AND, OR és XOR logikai műveletei az első operandust mindig az A regiszterből veszik, a második operandus a 8-bites összeadáshoz hasonlóan érkezhet az általános regiszterek bármelyikéből vagy a tárból. Az aritmetikai és logikai egység bitről bitre végrehajtja a kiválasztott logikai műveletet, majd az eredményt az A regiszterbe viszi. A logikai műveletek igen fontosak, nagyon sokféle célra alkalmazhatók.

Az AND, OR és XOR logikai műveletek a C jelzőbitet törlik, ami nagyon fontos akkor, ha C értékét meghatározottra kívánjuk beállítani, pl. egy

C szerinti feltételes vezérlésátadást tartalmazó programrészben. Az S, Z, P/V jelzőbitek azt mutatják, hogy a logikai művelet eredményeképpen milyen érték került az A regiszterbe. Ez felhasználható pl. az A-ba éppen betöltött érték vizsgálatára (mint tudjuk, az LD utasítások a jelzőbiteket nem állítják). A logikai műveletek során az N jelzőbit értéke mindig 0, a H jelzőbit értéke AND művelet esetén 1, különben 0 lesz.

A logikai AND műveletek az A regiszter tartalmát bitről bitre összehasonlítják a második operandusként megadott adattal, és ha mindkét helyen 1-et találnak, akkor az A regiszter megfelelő bitje 1 marad, egyébként 0 lesz. Az A regiszter és az általános regiszterek tartalmával logikai AND-et végző utasítások a következők:

AND A	A7
AND B	A0
AND C	A1
AND D	A2
AND E	A3
AND H	A4
AND L	A5

amelyekhez nyolcadikként szorosan illeszkedik az A regiszter tartalmával és a HL regiszterpár által mutatott tárrekesz tartalmával AND műveletet végző

AND (HL)	A6

utasítás. Az A regiszter tartalmával és az utasítás után álló közvetlen adattal végez logikai AND műveletet az

AND dd	E6 dd

utasítás, és végül az A regiszter tartalmával és az IX, ill. IY indexregiszterek és a címzési eltolás által meghatározott tárrekesz tartalmával végeznek logikai AND műveletet az

AND (IX+dis)	DD A6 XX
AND (IY+dis)	FD A6 XX

utasítások.

Megjegyezzük, hogy az AND A utasítás az A regiszter tartalmát változatlanul hagyja, csak a jelzőbiteket állítja.

A logikai OR műveletek az A regiszter tartalmát bitről bitre összehasonlítják a második operandusként megadott adattal, és ha bármelyik helyen vagy mindkét helyen 1-et találnak, akkor az A regiszter megfelelő bitjét 1-re állítják, ha nem – vagyis ha mindkét adat megfelelő bitje nulla – akkor az A regiszter megfelelő bitjét 0-ra állítják.

Az A regiszter tartalmával és az általános regiszterek tartalmával logikai OR műveletet végző utasítások a következők:

OR A	B7
OR B	BØ
OR C	B1
OR D	B2

OR E	B3
OR H	B4
OR L	B5

a tárban levő adattal OR műveletet végző utasítások – ugyanúgy, mint a logikai AND esetében – a következők:

OR (HL)	B6
OR dd	F6 dd
OR (IX+dis)	DD B6 XX
OR (IY+dis)	FD B6 XX

Megjegyezzük, hogy az OR A utasítás az A regiszter tartalmát változatlanul hagyja, csak a jelzőbiteket állítja.

A logikai XOR műveletek az A regiszter tartalmát bitről bitre összehasonlítják a második operandusként megadott adattal, és ha megfelelő bitek különbözők, akkor 1-et, ha azonosak, akkor 0-t írnak az A regiszter megfelelő bitjére. Az általános regiszterekben elhelyezett második operandussal az

XOR A	AF
XOR B	A8
XOR C	A9
XOR D	AA
XOR E	AB
XOR H	AC
XOR L	AD

utasítások, a tárból vett második operandussal – ugyanúgy, mint az előbb – az

XOR (HL)	AE
XOR dd	EE dd
XOR (IX+dis)	DD AE XX
XOR (IY+dis)	FD AE XX

utasítások végeznek XOR műveletet.

Megjegyezzük, hogy az XOR utasítás az A regisztert nullázza – valójában ez legjobb lehetőség arra, hogy az A regiszterbe 0-t töltsünk.

A logikai NEM művelet minden bit értékét az ellenkezőjére változtatja – igazból hamis, hamisból igaz lesz. Pl. a 8-bites 11001010 számmal végzett logikai NEM eredménye a következő:

```
NOT 11001010
----------------
    00110101
```

ahol NOT jelzi a logikai NEM-et.

Az aritmetikai és logikai egység a

CPL	2F

utasítással (ComPLement accumulator) végzi a logikai NOT műveletet. Ez az utasítás az A regiszter minden bitjét az ellenkezőjére változtatja – 0-ból 1-et, 1-ből 0-t csinál. A CPL utasítás – az AND, OR és XOR utasításokkal ellen-

tétben – nem állítja a jelzőbiteket, kivéve a H és N jelzőbiteket, amelyeket 1-re állít.

Az A regiszter tartalmának kettes komplemensét állítja elő a

NEG ED 44

utasítás, amely minden bitet az ellenkezőjére változtat – ugyanúgy, mint a CPL utasítás – majd a kapott értékhez 1-et hozzáad. Az eredmény tulajdonságait az S és Z jelzőbitek mutatják. A C jelzőbit értéke 0, ha az A regiszter tartalma a művelet előtt (és így a művelet után is) nulla, egyébként C értéke 1. A P/V jelzőbit 1, ha az A regiszter tartalma a művelet előtt (és így a művelet után is) hexadecimális 80, egyébként P/V értéke 0. A H jelzőbit nulla, ha a 4. bitről van alulcsordulás (beváltás), egyébként H értéke 1, végül az N jelzőbit értéke mindig 1.

4. fejezet

Vezérlésátadás

Az eddig tárgyalt Z80 utasítások a programszámlálót annyival állították előre, ahány byte-os volt az utasítás, így annak végrehajtása után a tárban következőként lerakott utasítás került beolvasásra és végrehajtásra. Gyakran van azonban arra szükseg, hogy ne a következő utasítással, hanem a tár egy másik részén elhelyezett utasítássorozat végrehajtásával folytatódjon a program. Pl. – a BASIC-ből ismert IF...THEN, FOR...NEXT és GOSUB...RETURN szerkezetekhez hasonlóan – lehet programunknak olyan része, amelyet csak valamilyen feltétel bekövetkezése esetén kell végrehajtani, lehet olyan programrész, amelyet egy feltétel bekövetkezéséig ismételni kívánunk, és lehetnek több helyről is hívott szubrutinjaink. Ezeket a funkciókat a vezérlésátadó utasításokkal lehet megvalósítani.

A vezérlésátadó utasítások között vannak feltétel nélküli és feltételes utasítások. A feltétel nélküli vezérlésátadó utasítások mindig új értéket írnak a programszámlálóba, tehát mindig végrehajtják a vezérlésátadást. A feltételes vezérlésátadó utasítások először a jelzőbitek segítségével feltételt vizsgálnak, és csak akkor írnak új értéket a programszámlálóba, ha a vizsgált feltétel teljesül, egyébként a sorban következő utasítás kerül végrehajtásra.

A vezérlésátadó utasítások általában több byte-osak, az első byte-on elhelyezett utasításkódot a vezérlésátadási cím követi.

Ugró utasítások

Feltétel nélküli vezérlésátadás következik be a közvetlen címzésű

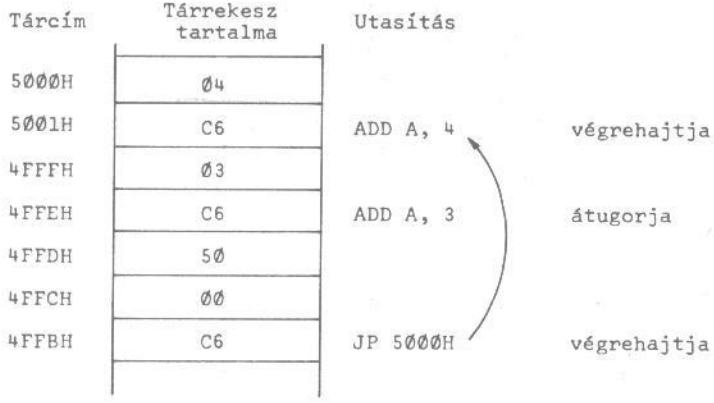
JP addr	C3 XX XX

utasítás (**JumP**) hatására. A programszámlálóba az utasításkód utáni két byte-on elhelyezett vezérlésátadási cím kerül (az első byte-on a cím alsó fele, a másodikon a felső fele).

A vezérlésátadási címet a HL regiszterpárban vagy az indexregiszterekben is elhelyezhetjük. A közvetett címzésű

JP (HL)	E9
JP (IX)	DD E9
JP (IY)	FD E9

utasítások a programszámlálóba a HL, IX, ill. IY tartalmát töltve hajtják végre a feltétel nélküli vezérlésátadást. A felsorolt JP utasítások nem állítják a jelzőbiteket.



4.1. ábra. Vezérlésátadás JP utasítással

A közvetlen címzésű

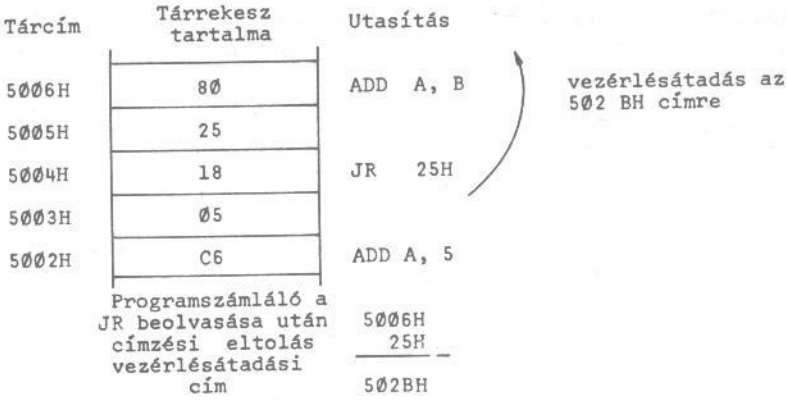
JP NZ,addr	C2 XX XX
JP Z,addr	CA XX XX
JP NC,addr	D2 XX XX
JP C,addr	DA XX XX
JP PO,addr	E2 XX XX
JP PE,addr	EA XX XX
JP P,addr	F2 XX XX
JP M,addr	FA XX XX

feltételes JP utasítások sorra csak akkor hajtják végre a vezérlésátadást, ha az utasítás végrehajtásakor a Z jelzőbit szerint az előző művelet eredménye nem nulla (NZ), ill. nulla (Z), a C jelzőbit állapota szerint nincs átvitel (NC), ill. van átvitel (C), a P/V jelzőbit páratlan paritást (P0), ill. páros paritást (PE) mutat, és végül ha az S jelzőbit szerint az előző művelet eredménye pozitív vagy nulla (P), ill. negatív (N). Így pl. a JP P,addr utasítás csak akkor helyezi az addr értéket a programszámlálóba, ha az S jelzőbit pozitív értéket mutat. Ha a vizsgált feltétel nem teljesül, akkor a programszámlálóba nem kerül új érték, és így a sorban következő utasítás végrehajtásával folytatódik a program. A feltételes JP utasítások nem állítják a jelzőbiteket. Megjegyezzük, hogy közvetett címzésű feltételes vezérlésátadó utasítások nincsenek.

A relatív címzésű

JR dis 18 XX

feltétel nélküli vezérlésátadó utasítás (Jump Relatív) első byte-ja az utasításkód, második byte-ja a címzési eltolás. A JR utasítás nem tölt új értéket a programszámlálóba, hanem az abban levő értékhez – a sorban következő utasítás címéhez – hozzáadja az egybyte-os kettes komplemens ábrázolású *dis* címzési eltolást. Így a JR utasítással az utasításkódtól számított –126... ... +129 byte-os tartományba lehet ugrani. A JR utasítás előnye, hogy egy byte-tal kevesebb tárat foglal el, mint a JP utasítás.



4.2. ábra. Vezérlésátadás JR utasítással

Megjegyezzük, hogy a –2 eltolásérték (kettes komplemens ábrázolásban hexadecimális FE) önmagára ugrást jelent. Kerülendő még a –1 eltolásérték (hexadecimális FF), mert az a címzési eltolást utasításként kezeli.

Relatív címzésű feltételes vezérlésátadást csak a

JR NZ,dis	20 XX
JR Z,dis	28 XX
JR NC,dis	30 XX
JR C,dis	38 XX

utasításokkal lehet végrehajtani, amelyek sorra a Z jelzőbit szerinti nem nulla (NZ) vagy nulla (Z), és a C jelzőbit szerinti nincs átvitel (NC) vagy van átvitel (C) feltételek teljesülése esetén adják hozzá a programszámláló értékéhez a címzési eltolást. A feltételes JR utasításokkal ugyanúgy az utasításkódtól számított –126 ... +129 byte-os tartományba lehet ugrani, mint a feltétel nélküli JR utasítással. A JR utasítások nem állítják a jelzőbiteket.

Összetett, tipikus ciklusszervezőfeladatot lát el a

DJNZ dis	10 XX

relatív címzésű vezérlésátadó utasítás (Decrement Jump if Non Zero), amely a B regiszter tartalmát eggyel csökkenti, majd ha az nem lett nulla, akkor – ugyanúgy, mint a JR utasítások – hozzáadja a címzési eltolást a programszámláló értékéhez. A DJNZ utasítással az utasításkódtól számított –126... ... +129 byte-os tartományba lehet ugrani. A DJNZ utasítás nem állítja a jelzőbiteket.

Szubrutinszervezés

Programok írásakor gyakran találunk olyan algoritmust, amelyet a program több pontján használhatunk. Ilyenkor nem érdemes ugyanazt az utasítássorozatot többször leírni – ezzel időnket is, a számítógép tárát is pazarolnánk, – sokkal okosabb, ha az utasítássorozatot csak egyszer helyezzük el a tárban, és ha a program valamelyik részén erre az utasítássorozatra van szükség,

akkor alkalmas módon ideugrunk – a főprogramból meghívjuk ezt a szubrutint, – majd visszaugrunk a főprogram következő utasítására – visszatérünk a szubrutinból.

A szubrutinhívás és a visszatérés feltétel nélküli vagy feltételes lehet. A szubrutinhívó utasítások a többi vezérlésátadó utasításhoz hasonlóan általában több byte-osak, az első byte az utasításkód, a következő két byte a vezérlésátadási cím.

Ha szubrutinhívás utasítást adunk ki, akkor a Z80 mikroprocesszor az SP regiszter által mutatott veremterületen megőrzi a programszámláló értékét, és PC-be a szubrutinhívás utasításban megadott címet tölti. Szubrutinból visszatéréskor a Z80 visszatölti a PC regiszterbe a veremterületre kimentett programszámlálóértéket, és így a vezérlés a szubrutinhívást követő utasításra kerül.

A feltétel nélküli szubrutinhívás a

CALL addr	CD XX XX

utasítással (**CALL**) történik. Az utasítás végrehajtása során az utasítást követő első byte címe kimentődik az SP regiszter által mutatott veremterületre, a programszámlálóba az utasításkód utáni két byte-on tárolt vezérlésátadási cím kerül (az első byte-on a cím alsó fele, a másodikon a cím felső fele), és SP értéke 2-vel csökken (SP így az alacsonyabb címek felé épülő veremterület utolsó adatának alsó byte-jára mutat).

A szubrutinból feltétel nélküli visszatérés a

RET	C9

utasítással (**RET**urn) történik, amely az SP regiszter által mutatott veremterületről kiolvassa az odamentett programszámláló-értéket, és azt visszatölti a programszámlálóba, továbbá 2-vel növeli az SP értékét (a veremterületről törli az oda kimentett PC értéket).

A feltételes vezérlésátadáshoz hasonlóan nyolc feltétel szerint lehet szubrutint hívni és szubrutinból visszatérni. A

CALL NZ,addr	C4 XX XX
CALL Z,addr	CC XX XX
CALL NC,addr	D4 XX XX
CALL C,addr	DC XX XX
CALL PO,addr	E4 XX XX
CALL PE,addr	EC XX XX
CALL P,addr	F4 XX XX
CALL M,addr	FC XX XX

szubrutinhívó és a

RET NZ	C0
RET Z	C8
RET NC	D0
RET C	D8
RET PO	E0

RET PE	E8
RET P	F0
RET M	F8

visszatérő utasítások paramétereként megadott feltételkódok megegyeznek a JP utasítások feltételkódjaival.

A CALL és RET utasítások a jelzőbiteket nem állítják.

A tár első nyolc 8-cal osztható címén kezdődő szubrutinok sorra az

RST 0	C7
RST 8	CF
RST 10	D7
RST 18	DF
RST 20	E7
RST 28	EF
RST 30	F7
RST 38	FF

egybyte-os utasításokkal (**ReST**art) is hívhatók. Az RST utasításokat elsősorban arra a célra tervezték, hogy a megszakításkor működő külső eszközről egybyte-os utasítással lehessen a megszakítást kiszolgáló rutinokhoz fordulni. Más programoknál tárat takaríthatunk meg azzal, hogy a gyakran használt szubrutinokat a tárban alkalmas módon elhelyezve az egybyte-os RST utasításokkal hívjuk. Az RST utasítások nem állítják a jelzőbiteket.

Megjegyezzük, hogy bekapcsoláskor a Z80 mikroprocesszor a 0000 címről indul, így az RST 0 utasítás hatására ugyanaz történik, mint a bekapcsoláskor.

Nem végez műveletet, csupán a programszámláló értékét növeli a

NOP	00

utasítás, amelyet tárkitöltő üres utasításként vagy várakozás céljából használhatunk. Megjegyezzük, hogy ugyanezt más Z80 utasításokkal – pl. LD A,A vagy OR A – is elérhetjük.

A Z80 mikroprocesszort a

HALT	76

utasítással megállíthatjuk. A HALT hatására a Z80 várakozik a következő megszakítás vagy törlés jelig (perifériáról érkező jelig vagy a RESET gomb megnyomásáig).

5. fejezet

Eltolások és bitműveletek

Assembler programokban gyakran kell az adatokat bitenként kezelni. A Z80 mikroprocesszor bit szintű utasításai a regiszterek, ill. tárrekeszek tartalmának balra és jobbra eltolását (egyszerű eltolás, léptetés) és ciklikus eltolását (forgatás) végző utasítások, valamint a bitállító, -törlő és -vizsgáló utasítások.

Az eltolások a regiszter vagy tárrekesz minden bitjét eggyel balra vagy jobbra léptetik. Az eltolás irányától függően a bal szélső vagy jobb szélső bit kiesik. Hogy ez ne okozzon információveszteséget, a C jelzőbit mutatja a kieső bit értékét. A ciklikus eltolásnál a kieső bit újra belép a másik irányból. Pl. ha jobbra végzünk ciklikus eltolást, akkor a kieső jobb szélső, legkevésbé értékes bitet körbevisszük, és balról legértékesebb bitként újra beléptetjük.

Az eltolások fontos alkalmazása a szorzás és az osztás. A bináris számok balra eltolása 2-vel szorzást, jobbra eltolása 2-vel osztást jelent, így a szorzás balra eltolásokra és összeadásokra, az osztás jobbra eltolásokra és kivonásokra vezethető vissza.

Eltolások

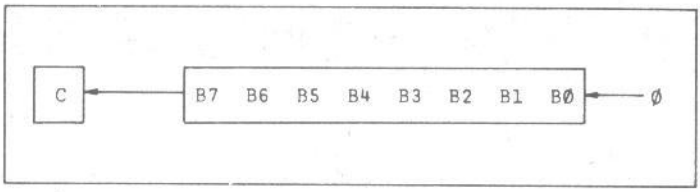
Az általános regiszterek tartalmát balra eltolják az

SLA A	CB 27
SLA B	CB 20
SLA C	CB 21
SLA D	CB 22
SLA E	CB 23
SLA H	CB 24
SLA L	CB 25

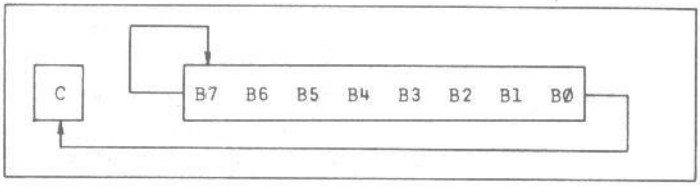
utasítások (Shift Left Arithmetic). Ezek az egyszerű eltolás utasítások a regiszter minden bitjét eggyel balra léptetik, a jobb szélső, megüresedő bit helyére 0-t írnak. A kieső bal szélső bit a C jelzőbitbe kerül. A H és N jelzőbitek törlődnek. Az S, Z és P/V jelzőbitek a művelet eredményének tulajdonságait mutatják (P/V itt paritást jelez).

Az

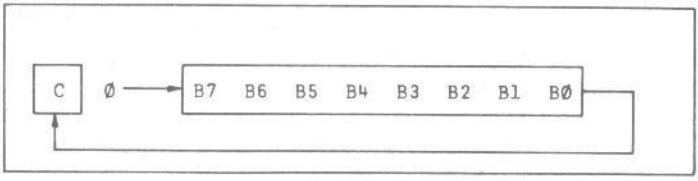
SLA (HL)	CB 26
SLA (IX+dis)	DD CB XX 26
SLA (IY+dis)	FD CB XX 26



SLA eltolás balra (Shift Left Arithnetic)



SRA eltolás jobbra (Shift Right Arithmetic)



SRL eltolás jobbra (Shift Right Logic)

5.1. ábra. Eltolások

utasítások a HL regiszterpár, ill. az indexregiszterek és a címzési eltolás által meghatározott tárrekesz tartalmával végeznek ugyanolyan balra eltolás műveletet, mint az előző SLA utasítások.

Jobbra eltolást két utasításcsoporttal lehet végezni. Az SRA utasítások az eltolás során a 7. bitet változatlanul hagyják és átmásolják a 6. bit helyére, míg az SRL utasítások a 7. bitre 0-t töltenek, tehát az SRA utasítások az előjeles számok előjelbitjét változatlanul hagyják (aritmetikai eltolás), az SRL utasítások viszont nem őrzik meg az előjelbitet (logikai eltolás).

Az általános regiszterek tartalmát az

SRA A	CB 2F
SRA B	CB 28
SRA C	CB 29
SRA D	CB 2A
SRA E	CB 2B
SRA H	CB 2C
SRA L	CB 2D

utasítások (Shift Right Aritmetic) oly módon léptetik jobbra, hogy a 7. bitet változatlanul hagyják és átmásolják a 6. bit helyére, a jobb szélső bitet pedig a C jelzőbitbe viszik. Az SRA utasítások ugyanúgy állítják a jelzőbiteket, mint az SLA utasítások.

A HL regiszterpár, ill. az indexregiszterek és a címzési eltolás által meghatározott tárrekesz tartalmával végeznek ugyanilyen eltolást az

SRA (HL)	CB 2E
SRA (IX+dis)	DD CB XX 2E
SRA (IY+dis)	FD CB XX 2E

utasítások.

Logikai eltolást végeznek az

SRL A	CB 3F
SRL B	CB 38
SRL C	CB 39
SRL D	CB 3A
SRL E	CB 3B
SRL H	CB 3C
SRL L	CB 3D

utasítások (Shift Right Logical), amelyek az általános regiszterek tartalmát oly módon léptetik jobbra, hogy a 7. bit helyére 0-t töltenek, a 0. bitet pedig a C jelzőbitbe viszik. Az SRL utasítások ugyanúgy állítják a jelzőbiteket, mint az előző eltolás utasítások.

A HL regiszterpár, ill. az indexregiszterek és a címzési eltolás által meghatározott tárrekesz tartalmával végeznek ugyanilyen eltolást az

SRL (HL)	CB 3E
SRL (IX+dis)	DD CB XX 3E
SRL (IY+dis)	FD CB XX 3E

utasítások.

Ciklikus eltolások

A balra és jobbra ciklikus eltolást végző utasítások egyik csoportja a kieső bitet az ellenkező oldalról közvetlenül belépteti, a másik csoportja a kieső bitet a C jelzőbitbe viszi, és annak előző tartalmát lépteti be az ellenkező oldalról.

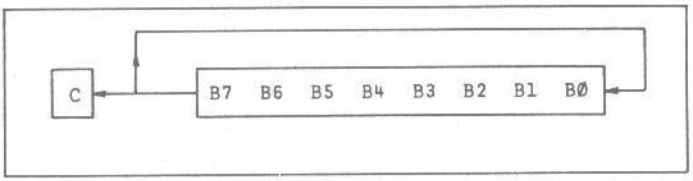
A C jelzőbit mindegyik ciklikus eltolásnál a kieső bit értékét mutatja. A H és N jelzőbiteket a ciklikus eltolások törlik. Az S, Z és P/V jelzőbiteket az A regiszter tartalmával ciklikus eltolást végző RLCA, RLA, RRCA és RRA utasítások változatlanul hagyják. A többi ciklikus eltolást végző utasítás a művelet eredménye szerint állítja ezeket a jelzőbiteket.

Az általános regiszterek tartalmával balra ciklikus eltolást végeznek az

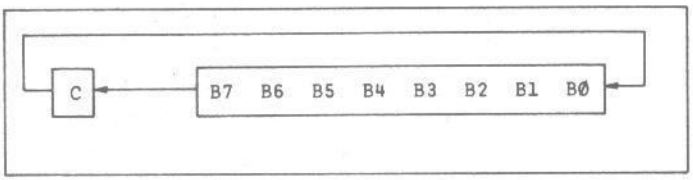
RLC A	CB 07
RLC B	CB 00
RLC C	CB 01
RLC D	CB 02
RLC E	CB 03

RLC H	CB 04
RLC L	CB 05

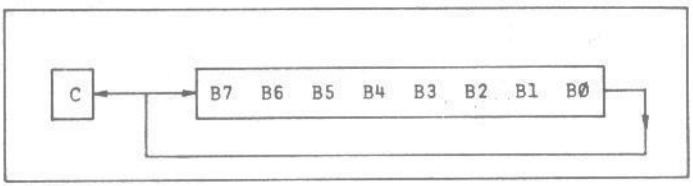
utasítások (Rotate Left Circular), amelyek a regiszter minden bitjét eggyel balra léptetik, a 7. bitet átviszik a 0. bit helyére, és a C jelzőbitet az átvitt 7. bit értéke szerint állítják.



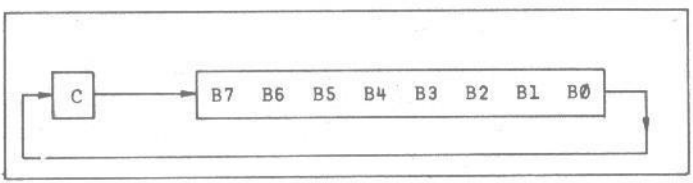
RLC ciklikus eltolás balra (Rotate Left Circular)



RL ciklikus eltolás C-n át balra (Rotate Left)



RRC ciklikus eltolás jobbra (Rotate Right Circular)



RR ciklikus eltolás C-n át jobbra (Rotate Right)

5.2. ábra. Ciklikus eltolások

A HL regiszterpár, ill. az indexregiszterek és a címzési eltolás által meghatározott tárrekesz tartalmával az

RLC (HL)	CB 06
RLC (IX+dis)	DD CB XX 06
RLC (IY+dis)	FD CB XX 06

utasítások végeznek balra ciklikus eltolást.

Az általános regiszterek tartalmával a C jelzőbiten át balra ciklikus eltolást végeznek az

RL A	CB 17
RL B	CB 10
RL C	CB 11
RL D	CB 12
RL E	CB 13
RL H	CB 14
RL L	CB 15

utasítások (Rotate Left), amelyek a 7. bitet a C jelzőbitbe viszik, a regiszter többi bitjét eggyel balra léptetik, a 0. bit helyére a C jelzőbit előző értékét töltik.

A HL regiszterpár, ill. az indexregiszterek és a címzési eltolás által meghatározott tárrekesz tartalmával az

RL (HL)	CB 16
RL (IX+dis)	DD CB XX 16
RL (IY+dis)	FD CB XX 16

utasítások végzik ugyanezt a műveletet.

Az általános regiszterek tartalmával jobbra ciklikus eltolást végeznek az

RRC A	CB 0F
RRC B	CB 08
RRC C	CB 09
RRC D	CB 0A
RRC E	CB 0B
RRC H	CB 0C
RRC L	CB 0D

utasítások, (Rotate Right Circular), amelyek a regiszter minden bitjét eggyel jobbra léptetik, a 0. bitet átviszik a 7. bit helyére, és a C jelzőbitet az átvitt 0. bit értéke szerint állítják.

A HL regiszterpár, ill. az indexregiszterek és a címzési eltolás által meghatározott tárrekesz tartalmával az

RRC (HL)	CB 0E
RRC (IX+dis)	DD CB XX 0E
RRC (IY+dis)	FD CB XX 0E

utasítások végzik ugyanezt a műveletet.

Az általános regiszterek tartalmával a C jelzőbiten át jobbra ciklikus eltolást végeznek az

RR A	CB 1F
RR B	CB 18
RR C	CB 19
RR D	CB 1A
RR E	CB 1B
RR H	CB 1C
RR L	CB 1D

utasítások (Rotate Right), amelyek a 0. bitet a C jelzőbitbe viszik, a regiszter többi bitjét eggyel jobbra léptetik, a 7. bit helyére a C jelzőbit előző értékét töltik.

A HL regiszterpár, ill. az indexregiszterek és a címzési eltolás által meghatározott tárrekesz tartalmával az

RR (HL)	CB 1E
RR (IX+dis)	DD CB XX 1E
RR (IY+dis)	FD CB XX 1E

utasítások végzik ugyanezt a műveletet.

A fenti RLC, RL, RRC és RR utasítások a H és N jelzőbiteket törlik, az S, Z és P/V jelzőbiteket a művelet eredménye szerint állítják (P/V az eredmény paritását jelzi).

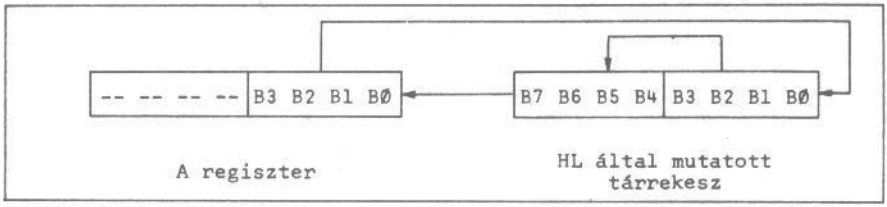
Az RLC A, RL A, RRC A és RR A utasításokhoz hasonlóak az

RLCA	07
RLA	17
RRCA	0F
RRA	1F

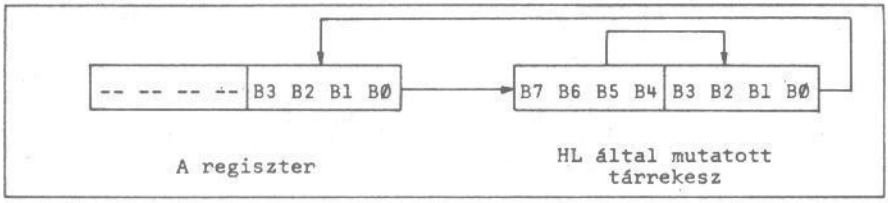
egybyte-os utasítások, amelyek az A regiszter tartalmát sorra a C jelzőbiten át balra, ill. jobbra ciklikusan eltolják, ugyanúgy, mint az előbb említett utasítások, törlik a H és N jelzőbiteket, de nem állítják az S, Z és P/V jelzőbiteket. Ezek az utasítások rövidebbek és gyorsabbak, mint az előbbi kétbyte-os hasonló utasítások, viszont csak akkor alkalmazhatók, ha az S, Z, P/V jelzőbitek vizsgálatára nincs szükség.

Az RLD és RRD utasítások három 4-bites adattal – a HL regiszterpár által mutatott tárrekesz alsó és felső 4 bitjével és az A regiszter első 4 bitjével – végeznek balra, ill. jobbra ciklikus eltolást. Az

RLD ED 6F



RLD félbyte-os ciklikus eltolás balra (Rotate Left Decimal)



RRD félbyte-os ciklikus eltolás jobbra (Rotate Right Decimal)

5.3. ábra. Félbyte-os ciklikus eltolások

utasítás (Rotate Left Digit) a HL által mutatott tárrekesz 0–3. bitjeit a 4–7. bitek helyére, 4–7. bitjeit az A regiszter 0–3. bitjei helyére, az A regiszter 0–3. bitjeit a HL által mutatott tárrekesz 0–3. bitjei helyére viszi, az A regiszter 4–7. bitjeit változatlanul hagyja.

Az RLD utasítás a C jelzőbitet változatlanul hagyja, a H és N jelzőbiteket törli, az S, Z és P/V jelzőbiteket az A regiszter új tartalma szerint állítja (P/V az eredmény paritását jelzi).

Az

RRD ED 67

utasítás (Rotate Right Digit) a HL által mutatott tárrekesz felső felét annak alsó felébe, alsó felét az A regiszter alsó felébe, az A regiszter alsó felét a HL által mutatott tárrekesz felső felébe viszi át, az A regiszter felső felét változatlanul hagyja. Az RRD utasítás ugyanúgy állítja a jelzőbiteket, mint az RLD utasítás.

Bitműveletek

A BIT, SET, RES utasításokkal bármely általános regiszter vagy tárrekesz bármely bitjét vizsgálhatjuk, 1-re állíthatjuk vagy törölhetjük. Ezeket a műveleteket a bitszintű utasítások nélkül, adatmozgató és logikai utasításokkal is el lehetne végezni, az itt következő utasítások azonban hatásosabb, gyorsabb lehetőséget adnak ugyanerre. Mivel a 8-bites adatok bármely bitjével lehet bitműveletet végezni, ezért olyan sok BIT, SET, RES utasítás van, hogy csak néhányat fogunk közülük felsorolni.

A

BIT 0,A	CB 47
BIT 0,B	CB 40
BIT 0,C	CB 41
.	.
.	.
.	.
BIT 0,L	CB 45
BIT 1,A	CB 4F
BIT 1,B	CB 48
.	.
.	.
.	.
BIT 7,L	CB 7D

bitvizsgáló utasítások (BIt Test) a második paraméterként megadott általános regiszter első paraméterként megadott sorszámú bitjét vizsgálják. Ha a vizsgált bit értéke 1, akkor a Z jelzőbitet törlik, ha a vizsgált bit értéke 0, akkor Z-t 1-re állítják.

A többi jelzőbit közül C változatlan marad, H értéke 1 lesz, N törlődik, az S és P/V jelzőbitek pedig előre nem látható módon kapnak értéket.

A HL regiszterpár, ill. az IX, IY indexregiszterek és a címzési eltolás által meghatározott tárrekesz tartalmának bitjeit vizsgálják a

```
BIT 0,(HL)          CB 46
BIT 1,(HL)          CB 4E
.                   .
.                   .
.                   .
BIT 7,(HL)          CB 7E
BIT 0,(IX+dis)      DD CB XX 46
BIT 1,(IX+dis)      DD CB XX 4E
.                   .
.                   .
.                   .
BIT 7,(IY+dis)      FD CB XX 7E
```

utasítások, amelyek a jelzőbiteket ugyanúgy állítják, mint az előző BIT utasítások.

A

```
SET 0,A             CB C7
SET 0,B             CB C0
.                   .
.                   .
.                   .
SET 0,L             CB C5
SET 1,A             CB CF
SET 1,B             CB C8
.                   .
.                   .
.                   .
SET 7,L             CB FD
```

bitállító utasítások (**SET**) a második paraméterként megadott általános regiszter első paraméterként megadott sorszámú bitjét 1-re állítják. A SET utasítások a jelzőbiteket változatlanul hagyják.

A HL regiszterpár, ill. az IX, IY indexregiszterek és a címzési eltolás által meghatározott tárrekesz tartalmának bitjeit állítják a

```
SET 0,(HL)          CB C6
SET 1,(HL)          CB CE
,                   .
.                   .
.                   .
SET 7,(HL)          CB FE
SET 0,(IX+dis)      DD CB XX C6
SET 1,(IX+dis)      DD CB XX CE
```

```
.                    .
.                    .
.                    .
SET 7,(IY+dis)       FD CB XX FE
```

utasítások, amelyek ugyanúgy, mint az előző SET utasítások, változatlanul hagyják a jelzőbiteket.

A

```
RES 0,A              CB 87
RES 0,B              CB 80
.                    .
.                    .
.                    .
RES 0,L              CB 85
RES 1,A              CB 8F
RES 1,B              CB 88
.                    .
.                    .
.                    .
RES 7,L              CB BD
```

bittörlő utasítások (**RES**et) a második paraméterként megadott általános regiszter első paraméterként megadott sorszámú bitjét 0-ra állítják. A RES utasítások a jelzőbiteket változatlanul hagyják.

A HL regiszterpár, ill. az IX, IY indexregiszterek és a címzési eltolás által meghatározott tárrekesz tartalmának bitjeit törlik a

```
RES 0,(HL)           CB 86
RES 1,(HL)           CB 8E
.                    .
.                    .
.                    .
RES 7,(HL)           CB BE
RES 0,(IX+dis)       DD CB XX 86
RES 1,(IX+dis)       DD CB XX 8E
.                    .
.                    .
.                    .
RES 7,(IY+dis)       FD CB XX BE
```

utasítások, amelyek ugyanúgy, mint az előző RES utasítások, változatlanul hagyják a jelzőbiteket.

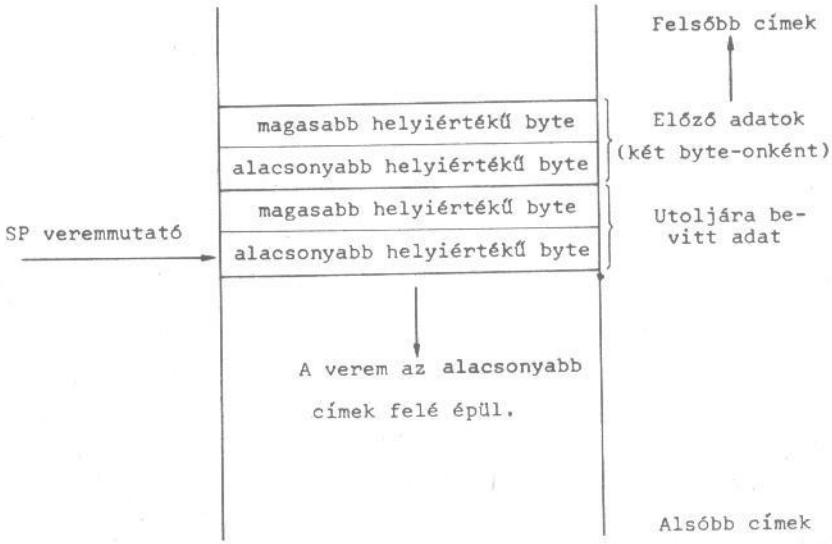
6. fejezet

Veremkezelés

Az adatok ideiglenes tárolására szolgáló veremterületről mindig az utoljára tárolt adat olvasható ki először (LIFO = **L**ast-**I**n-**F**irst-**O**ut = utolsó bemenő első kimenő). Ezzel a módszerrel kell tárolni pl. a szubrutinhívások visszatérési címeit, hiszen mindig az utoljára hívott szubrutinból kell először visszatérni. Vannak más olyan adatok is, amelyek ideiglenes tárolása így történhet (regisztertartalmak kimentése, regiszterpárok közötti adatmozgatást stb.).

A veremterületet a Z80 mikroprocesszor SP verem mutató regisztere kezeli. Az SP regiszternek valamilyen kezdeti értéket adva a veremterületet a tárban bárhol elhelyezhetjük. Mivel a verem lefelé, a magasabb tárcímektől az alacsonyabbak felé épül, ezért azt általában a tár felső részére szokás helyezni. Az SP regiszter a kezdeti érték beállítása után automatikusan követi az adatforgalmat. Ha a verembe adatot tárolunk, akkor SP értéke csökken, ha a veremből adatot olvasunk ki, akkor SP értéke növekszik. Mivel a veremterületet kétbyte-os adatok (tárcímek, regiszterpárok) tárolására tervezték, ezért az SP által mutatott tárcím a verembe íráskor automatikusan mindig 2-vel csökken, a veremből olvasáskor pedig 2-vel növekszik.

Pl. a CALL utasítás a vezérlésátadással egyidejűleg kiviszi a veremterületre a szubrutinból visszatérés után végrehajtandó első utasítás címét, a RET



6.1. ábra. Veremterület

utasítás pedig a veremterületről ezt a címet kiolvasva határozza meg, hogy hova kell visszatérnie. A CALL utasítás során SP tartalma 2-vel csökken, a RET utasítás hatására SP értéke 2-vel növekszik (így a kiolvasott adat a veremből mintegy „kiesik").

Veremkezelő utasítások

A tárban a veremterületet úgy jelölhetjük ki, hogy kezdeti értéket adunk az SP verem mutató regiszternek. Ezt az Adatmozgatás című fejezetben ismertetett

LD SP, dddd	31 dd dd
LD SP,HL	F9
LD SP,IX	DD F9
LD SP,IX	FD F9
LD SP,(addr)	ED 7B XX XX

utasításokkal tehetjük meg. Miután SP kezdeti értékének megadásával a veremterületet elhelyeztük a tárban, nem kell tovább foglalkoznunk SP értékének a változtatásával, mert a veremterületre adatot tároló, ill. onnan adatot kiolvasó utasítások ezt automatikusan megteszik. A fenti utasítások közül leggyakrabban talán az LD SP,dddd utasítást szokták alkalmazni, amely közvetlenül ad értéket az SP regiszternek.

Egy programban több veremterületet is kijelölhetünk. Ezt támogatják az LD SP,(addr) és az

LD (addr), SP	ED 73 XX XX

utasítások, amelyekkel a veremmutatónak a tárból adhatunk értéket, ill. a veremmutató értékét a tárba kimenthetjük.

Az SP regiszter értékét az Aritmetika és logika című fejezetben ismertetett

ADD HL,SP	39
ADD IX,SP	DD 39
ADD IY,SP	FD 39
ADC HL,SP	ED 7A
SBC HL,SP	ED 72

utasítások segítségével módosíthatjuk, ill. az

INC SP	33
DEC SP	3B

utasításokkal növelhetjük, ill. csökkenthetjük. Az ADD, ADC és SBC utasítások eredménye nem az SP regiszterbe kerül, az előző adatmozgató utasításokkal azonban átvihetjük azt az SP regiszterbe. Az INC és DEC utasítások az SP értékét csak 1-gyel növelik, ill. csökkentik, míg az alábbi, az SP értékét automatikusan állító utasítások 2-vel csökkentik, ill. növelik azt.

Regiszterpárok, ill. indexregiszterek értékét viszik a veremterületre a

PUSH AF	F5
PUSH BC	C5
PUSH DE	D5
PUSH HL	E5
PUSH IX	DD E5
PUSH IY	FD E5

utasítások (**PUSH**). Ezek az utasítások először 1-gyel csökkentik SP értékét, majd az SP által mutatott tárcímre tárolják a regiszterpár felső regiszterét, ill. az indexregiszter felső felét, ezután újból 1-gyel csökkentik az SP értékét, és a most mutatott tárcímre tárolják a regiszterpár alsó regiszterét, ill. az indexregiszter alsó felét. A művelet eredményeképpen tehát egy 16 bites adat tárolásra kerül a veremterületen, és az SP verem mutató értéke 2-vel csökken. A PUSH AF utasítás az A regisztert és a jelzőbiteket együtt tárolja, így az A regiszterrel együtt a jelzőbitek állapota is kimentésre kerül. A PUSH utasítások a jelzőbiteket nem állítják.

A veremterületre helyezett adatokat a

POP AF	F1
POP BC	C1
POP DE	D1
POP HL	E1
POP IX	DD E1
POP IY	FD E1

utasításokkal (**POP**) hozhatjuk vissza a regiszterpárokba, ill. az indexregiszterekbe. A POP utasítások végrehajtása során az SP regiszter által mutatott tárcímről betöltésre kerül a regiszterpár alsó regisztere, ill. az indexregiszter alsó fele, majd SP értéke 1-gyel növekszik, ezután az SP által mutatott címről betöltésre kerül a regiszterpár felső regisztere, ill. az indexregiszter felső fele, végül SP értéke újból 1-gyel növekszik. A művelet eredményeképpen tehát egy 16 bites adat a veremterületről kiolvasásra kerül, és az SP verem mutató értéke 2-vel növekszik, tehát a kiolvasott adat a veremterületen kívülre kerül. A POP AF utasítás az A regiszterrel együtt a jelzőbiteket is betölti, így a jelzőbitek állapota az előző PUSH AF utasítással kimentett korábbi állapotot tükrözheti. A többi POP utasítás a jelzőbiteket nem állítja.

Az SP regiszter által mutatott tárterület és a HL regiszterpár, ill. az IX, IY indexregiszterek között végeznek adatcserét az

EX (SP),HL	E3
EX (SP),IX	DD E3
EX (SP),IY	FD E3

utasítások, amelyek az SP által mutatott tárrekesz tartalmát a HL, IX, IY alsó felével, a következő tárcím tartalmát a HL, IX, IY felső felével cserélik fel. Az EX utasítások a jelzőbiteket nem állítják.

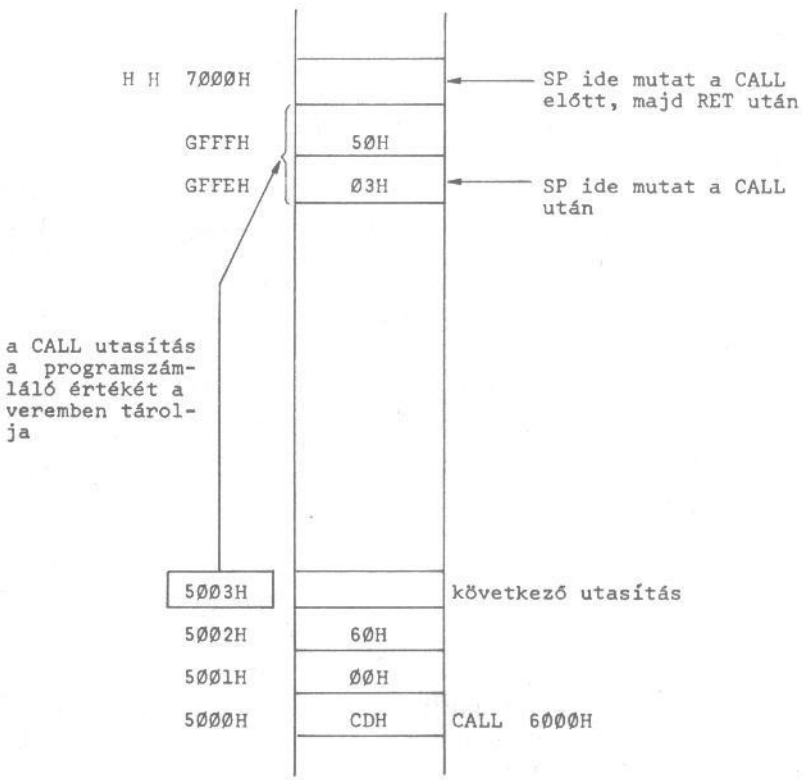
Szubrutinok szervezése

A CALL utasítással hívott szubrutinból visszatérve a program végrehajtása a CALL utáni utasítással kell, hogy folytatódjék. Ezért a szubrutin hívásakor a CALL utasítás utáni cím a veremterületen megőrződik, és a szubrutinból visszatéréskor a RET utasítás a veremterületről kiolvassa a CALL által odamentett visszatérési címet. Pontosabban, a hárombyte-os CALL utasítás a programszámláló értékét 3-mal növelve a következő utasítás kezdetére mutat, majd SP értékét 1-gyel csökkentve az így mutatott címre tárolja a következő utasítás címének felső felét, ezután SP értékét újra 1-gyel csökkentve tárolja a cím alsó felét, végül a programszámlálóba a CALL paramétereként megadott címet töltve a szubrutinnak adja át a vezérlést.

Pl. ha a veremterületet a 7000H tárcímtől lefelé helyezzük el, tehát SP a 7000H címre mutat, és a 6000H címen kezdődő szubrutint meghívó

CALL 6000H CD 00 60

utasítás első byte-ja az 5000H címen van, akkor a CALL utáni utasítás az 5003H címen kezdődik. A CALL utasítás SP értékét először 6FFFH-ra csökkenti, ide tárolja az 5003H visszatérési cím felső felét, az 50H értéket, majd SP értékét 6FFEH-ra csökkenti és ide tárolja a visszatérési cím alsó



6.2. ábra. CALL hatása a veremre

felét, a 03H értéket. A szubrutinból visszatérő RET utasítás az SP által mutatott 6FFEH címről kiolvassa a visszatérési cím alsó felét, majd SP értékét 1-gyel növelve kiolvassa a cím felső felét, ezután ismét növeli SP értékét, amely most újra a 7000H kiindulási értéket tartalmazza, végül a két címről kiolvasott 5003H visszatérési címet a programszámlálóba tölti, és így a program végrehajtása a CALL utáni utasítással folytatódik.

Adattárolás, adatcsere

Regisztertartalmak kimentésére, visszatöltésére és még sok más célra, sokféle programozási helyzetben alkalmazhatók a PUSH és POP utasítások.

Például gyakran előfordul, hogy az általános regiszterek tartalmát ideiglenesen tárolni kívánjuk. Ezt kényelmesebb úgy megtenni, hogy nem jelölünk ki erre a célra külön tárterületet, hanem a veremterületet használjuk. Ha meg akarjuk őrizni pl. a DE regiszterpár tartalmát, akkor azt a PUSH DE utasítással a veremterületre helyezzük, majd onnan a POP DE utasítással visszatölthetjük.

Ezzel a módszerrel több regiszter tartalmát is megőrizhetjük. Pl. ha egy szubrutin az összes általános regisztert használja, a meghívása előtt a regiszterek mindegyikét ki kell menteni, majd a visszatérés után az összes regiszterpárba vissza kell tölteni a kimentett előző tartalmat:

```
PUSH HL        E5
PUSH DE        D5
PUSH BC        C5
PUSH AF        F5
CALL 6000H     CD 00 60
POP AF         F1
POP BC         C1
POP DE         D1
POP HL         E1
```

A POP utasítások a PUSH utasításokhoz képest fordított sorrendben következnek, hiszen a veremterületről mindig az utoljára odahelyezett adatot olvashatjuk ki legelőször. Megjegyezzük, hogy a CALL utasítás után a veremterületen öt 16 bites adat van, felül a CALL által kimentett visszatérési cím (példánkban a POP AF utasítás címe), a tárban a magasabb címek felé haladva sorra a PUSH utasításokkal az AF, BC, DE, HL regiszterpárokból kimentett adatok következnek. A PUSH és POP utasításoknak mindig párban kell állniuk, különben elrontjuk a vermet, nem azt az adatot kapjuk vissza, amit kimentettünk, nem oda tér vissza a RET, ahonnan a szubrutint hívtuk stb.

A regiszterpárok közötti adatcsere a DE és HL regiszterpárok esetében az EX DE, HL utasítással egyszerűen megoldható, más regiszterpárok közötti adatcserére azonban nincs hasonló EX utasítás. Ezt a PUSH és POP utasításokkal valósíthatjuk meg oly módon, hogy az egyik regiszterpárból kimentett adatot a másik regiszterpárba töltjük vissza. Pl. a BC és DE regiszterpárok tartalmát a

```
PUSH DE        D5
PUSH BC        C5
POP DE         D1
POP BC         C1
```

utasításokkal cserélhetjük fel. A két PUSH utasítás után a veremterület tetején a BC regiszterpár, alatta a DE regiszterpár tartalma van, így az első POP a DE regiszterpárba a BC-ből kimentett adatot, a második POP a BC regiszterpárba a DE-ből kimentett adatot tölti.

A PUSH és POP utasításpár egy másik alkalmazása a jelzőbitek megőrzése. Pl. ha egy adat értéke szerint több helyen kívánunk feltételes vezérlésátadó utasításokat alkalmazni, akkor elég, ha az adatot a megfelelő LD utasítással egyszer betöltjük, az AND A utasítással beállítjuk a jelzőbiteket, majd a PUSH AF utasítással megőrizzük őket. A megfelelő helyen alkalmazott POP AF utasítással kimentett jelzőbitek újra az F regiszterbe kerülnek. Ez az eljárás gyorsabb, mint ha ugyanazt az adatot minden alkalommal betöltenénk és újra beállítanánk a jelzőbiteket.

Előfordulhat, hogy nem a verem tetején levő adatra van szükségünk, hanem valamelyik korábban kimentett adathoz szeretnénk először hozzáférni. Pl. ha a DE és HL regiszterpárok tartalmát a

```
PUSH DE        D5
PUSH HL        E5
```

utasításokkal a veremterületre tároljuk, akkor a verem tetején a HL regiszterpár, alatta a DE regiszterpár tartalma van. Ha először a DE regiszterpárból kimentett adatot szeretnénk kiolvasni, akkor megtehetjük, hogy egy éppen nem használt regiszterpárba vagy indexregiszterbe kivisszük a verem felső elemét, kiolvassuk a keresett, a veremben most már felül levő adatot, majd visszatöltjük a kimentett felső elemet. Ha az IY indexregisztert másra éppen nem használjuk, akkor ezt a

```
POP  IY        FD E1
POP  DE        D1
PUSH IY        FD E5
```

utasítássorozattal tehetjük meg.

Ha éppen nincs szabad regiszterpár, ill. indexregiszter, vagy ha a veremben mélyebben levő adathoz szeretnénk hozzáférni, akkor az

```
ADD HL,SP      39
ADD IX,SP      DD 39
ADD IY,SP      FD 39
```

utasításokkal kiszámolhatjuk a keresett adat címét.

Például ha a verem felülről számított tizedik elemét szeretnénk mondjuk a BC regiszterpárba tölteni, akkor, mivel a verem minden eleme kétbyte-os, 20-at (hexadecimális 14-et) töltünk az IX indexregiszterbe (ill. HL-be vagy IY-ba), és az

```
ADD IX,SP      DD 39
```

utasítással kiszámoljuk a keresett adat tárcímét, majd az

```
LD B,(IX+0)        DD 46 00
LD C,(IX+1)        DD 46 01
```

utasításokkal a BC regiszterpárba töltjük az adatot, miközben a veremmutató és a veremterület változatlan marad. Másik lehetőség, hogy az IX indexregiszterben levő tárcímet átmozgatjuk az SP regiszterbe és POP utasítással olvassuk ki a keresett adatot:

```
LD SP,IX           DD F9
POP BC             C1
```

de ezután gondoskodnunk kell SP értékének visszaállításáról.

Adatátadás a szubrutinnak

A szubrutinoknak a veremterületen keresztül adhatunk át adatokat úgy, hogy mielőtt meghívjuk a szubrutint, PUSH utasítással a verembe visszük az átadandó adatot. Pl., ha a DE regiszterpárban levő adatot kívánjuk a 6000H címen kezdődő szubrutinnak átadni, akkor a

```
PUSH DE            D5
CALL 6000H         CD 00 60
```

utasításokkal DE tartalmát a veremterületre tároljuk, majd meghívjuk a szubrutint. A CALL utasítás azonban a veremre helyezi a visszatérési címet, így a szubrutinba lépés után a továbbítandó adat már nem a verem tetején van. Ha a verem felső két elemét POP utasításokkal két regiszterpárba betöltjük, majd a visszatérési címet PUSH utasítással újból a verem tetejére helyezzük – mint ahogyan azt az előbb csináltuk –, akkor a feladatot megoldottuk. Van azonban ennél rövidebb, egyszerűbb és gyorsabb lehetőség is ugyanerre: POP utasítással a HL regiszterpárba töltjük a verem tetején levő visszatérési címet, majd EX (SP), HL utasítással az SP által mutatott veremelemet (a szubrutinnak továbbítandó adatot) felcseréljük a HL regiszterpár tartalmával (a visszatérési címmel):

```
POP HL             E1
EX (SP),HL         E3
```

Ezt az eljárást ismételve több adatot is adhatunk át a szubrutinnak.

A szubrutinból a főprogramnak lényegében ugyanezzel a módszerrel továbbíthatunk adatokat. A verem tetején a visszatérési cím van, ennek ott is kell maradnia. Ha az átadandó adatot a HL regiszterpárban helyeztük el, akkor az

```
EX (SP),HL         E3
PUSH HL            E5
```

utasításokkal először felcseréljük a HL regiszterpár tartalmát (az adatot) és a verem felső elemét (a visszatérési címet), majd a PUSH utasítással a verem te-

tejére helyezzük a visszatérési címet. Miután az ezt követő RET utasítás a verem tetején levő címet kiolvasva visszatér a szubrutinhívás utáni utasításhoz, a verem tetején éppen az átadandó adat lesz. Az eljárást ismételve több adatot is továbbíthatunk a főprogramnak.

Összefoglalva, a veremkezelő utasítások helyes alkalmazásával tömörebbé, szebbé, gyorsabbá tehetjük programjainkat.

7. fejezet

Gépi kód és assembler

Miután az eddigi fejezetekben megismerkedtünk a Z80 mikroprocesszor utasításainak nagy részével, nekiláthatunk annak, hogy programokat írjunk ezen a nyelven. Erre két alapvetően különböző lehetőség van: a gépi kódú és az assembly nyelvű programozás. A gépi kódú programozás azt jelenti, hogy a Z80 utasításokat – az előző fejezetekben jobb oldalon megadott hexadecimális formában – közvetlenül beírjuk a tárba, majd a vezérlést alkalmas módon az így bevitt utasításokra adjuk. Ha viszont assembly nyelven írunk, akkor a programot – az előző fejezetekben bal oldalon megadott – szimbolikus formában írjuk meg, ezt az assembler fordítóprogram segítségével gépi kódúvá alakítjuk, és az így átalakított programot helyezzük el a tárban. A két lehetőség végeredménye ugyanaz, de számunkra lényegesen könnyebb assemblerre írni, mint gépi kódban, már csak azért is, mert könnyebben meg lehet jegyezni a szimbolikus utasításokat, mint a hexadecimális számok sokaságát.

Az, hogy milyen módon lehet a gépi kódú programot a számítógép tárában elhelyezni, függ az adott számítógép felépítésétől, tárelrendezésétől. Az, hogy assembly nyelven hogyan kell programot írni, függ az adott assembler fordítóprogram működésétől. Továbbá már a legegyszerűbb assembly programokban sem lehet elkerülni, hogy kihasználjuk az adott számítógép lehetőségeit. Ezért – bár az eddigi fejezetekben elmondottak minden Z80 mikroprocesszorral működő számítógépre érvényesek – a továbbiakat arra fogjuk építeni, hogy HT-1080Z, HT-2080Z számítógépen, ill. az ugyanilyen tárelrendezésű, ugyanezen Microsoft Level II. BASIC értelmezővel működő TRS-80 vagy Video Genie vagy azzal kompatibilis számítógépen dolgozunk. Példáinkat a HT-1080Z számítógép EDI assembler/editor programja segítségével fordítottuk, futtattuk, de megjegyezzük, hogy ugyanezen számítógépre többféle assembler fordítóprogram van forgalomban, és a könyv további követésének egyáltalán nem feltétele, hogy az Olvasó ugyanezt az assembler/editort használja. Bármely assembler/editor programmal vagy gépi kód szinten a HT-1080Z számítógép 12710 címen kezdődő gépi kódú monitorjával, ill. a kazettáról betölthető TBUG programmal ki lehet próbálni példáinkat, és lehet saját programokat írni.

Noha a gépi kódú programozás az assemblerrel történő programozáshoz képest igen nehézkes, mégis érdemes egy-két rövidebb programot ezzel a körülményes módszerrel beírni a számítógépbe, mert ezzel közelebb juthatunk annak megértéséhez, hogy az assembler fordítóprogramok hogyan működnek.

Programozás gépi kódban

A gépi kódú programozást egy igen egyszerű feladaton mutatjuk be a HT-1080Z képernyőjének közepére kiírunk egy "A" betűt.

A HT-1080Z képernyője 16 soros, minden sorban 64 karakterrel, így a teljes képernyő 1024 karakter. A képernyő tartalmát tároló tárrekeszek a 15360 (3C00H) és a 16383 (3FFFH) tárcím között helyezkednek el. Ha a képernyő közepére, mondjuk a 9. sor 33. karakterére kívánunk írni, akkor az első sor első elemét tároló 15360 címhez hozzá kell adnunk soronként 64-et és a 9. sor elejétől közepéig még 32-t, így a 15360 +512+32 = 15904 (3E20H) címre kell az "A" karaktert tárolnunk. Ezt úgy oldhatjuk meg, hogy valamelyik általános regiszterbe betöltjük az "A" karakternek megfelelő ASCII kódot (a Függelék ASCII kódtáblázatából látjuk, hogy ez 41H, vagyis decimális 65), majd a regiszterből az adatot az előbb kiszámított tárcímre, a képernyő közepére mozgatjuk. Az

```
LD A,41H
LD (3E20H),A
```

utasítások közül az első közvetlen adattal, az "A" betű ASCII kódjával tölti az A regisztert, a második utasítás pedig közvetlenül a zárójelben megadott címre tárolja az A regiszter tartalmát.

Az Adatmozgatás című fejezetben, ill. a Függelék utasítástáblázataiban megtalálhatjuk, hogy az LD A,dd utasítás egybyte-os kódja 3EH, amit a betöltendő adat kódjának kell követnie, ami példánkban 41H. A második utasítás, az LD (addr),A egybyte-os kódja 32H, amit a cím alsó felének (20H), majd felső felének (3EH) kell követnie. Megvan tehát a gépi kód öt byte-ja, írjuk le azt az assembler utasítások elé:

```
3E 41          LD A,41H
32 20 3E       LD (3E20H),A
```

Most már csak az van hátra, hogy a gépi kódot elhelyezzük a tárban. A HT-1080Z tárjának a képernyőcímek alatti részét a BASIC értelmező foglalja el, a képernyő után a BASIC értelmező, majd a BASIC program számára fenntartott helyek következnek, így a gépi kódú programok elhelyezésére a magasabb tárcímek a legalkalmasabbak. Könyvünkben általában a 7000H tárcímtől, a 16 Kbyte tárkapaciátsú HT-1080Z számítógépek tárának felső 4 Kbyte-ján fogjuk a programokat elhelyezni. Példánkban az első utasítás két byte-ja sorra a 7000H, 7001H címekre, a második utasítás három byte-ja a 7002H, 7003H és 7004H címekre kerül:

```
7000H    3E 41       LD A,41H
7002H    32 20 3E    LD (3E20H),A
```

Következő teendőnk, hogy a számítógép tárába beírjuk ezeket a hexadecimális értékeket. A HT-1080Z számítógépen a SYSTEM parancs után a 12710 tárcímet megadva lépjünk be a gépi kódú monitorba, majd az M tármódosító paranccsal byte-onként írjuk be az utasításokat, végül az X válasszal lépjünk ki a tármódosítási funkcióból:

```
SYSTEM
*? /12710
M 7000
7000 00–3E
7001 00–41
7002 00–32
7003 00–20
7004 00–3E
7005 00–X
```

Miután így elhelyeztük a gépi kódú programot, a G 7000 paranccsal megkezdhetjük végrehajtását. Ha mindent hibátlanul csináltunk, akkor valóban meg fog jelenni az "A" betű a képernyő közepén, de csak alig észlelhetően rövid ideig lesz látható, ugyanis a program végét nem írtuk meg, a Z80 program számlálójának értéke a második utasítás végrehajtása során tovább növekszik, a gépi kód végrehajtása a következő címen tovább folytatódik. Mivel a következő címre nem írtunk utasításkódot, ezért ennek következményei kiszámíthatatlanok (ha szerencsénk van, megjelenik a READY? kérdés, ha nem, ki kell kapcsolnunk a gépet, és csak az újbóli bekapcsolás után folytathatjuk vagy inkább elölről kezdhetjük a munkát). Ha viszont a következő tárcímekre – a BASIC-ből ismert 100 GOTO 100 szerkezethez hasonló – önmagára ugró végtelen hurkot írunk, akkor a képernyő tartalma megmarad, és a RESET gomb megnyomásával újra életre lehet kelteni a gépet. Ehhez a következő címtől, a 7005H-tól a JP 7005H utasítást kell elhelyeznünk. A JP utasítás egybyte-os kódja C3, amit a cím alsó fele (05H), majd a cím felső fele (70H) követ. A program így már három utasításból áll:

7000H	3E 41	LD A,41H
7002H	32 20 3E	LD (3E20H),A
7005H	C3 05 70	JP 7005H

A gépi kódú monitorral folytathatjuk a program beírását:

```
7005 00–C3
7006 00–05
7007 00–07
7008 00–X
```

Ha most a 7000H címtől újra elindítjuk a programot, akkor a képernyőn megjelenő "A" betű ott is marad egészen addig, amíg a gépet ki nem szabadítjuk abból a végtelen hurokból, amibe a JP 7005H utasítással belerángattuk.

A TBUG program segítségével a fentiekhez hasonlóan, de kicsit más parancsokkal vihetünk be és futtathatunk gépi kódú programot.

Assemblerrel ugyanezt sokkal egyszerűbben, gyorsabban megcsinálhatjuk, nem szólva arról, hogy az assembler fordító a program fordítása során ellenőrzi, hogy hibátlanok-e az utasítások, és sok egyéb olyan funkciót is ellát, amire gépi kódban nekünk magunknak kell gondolnunk.

Programozás assemblerben

Gépi kódban úgy programoztunk, hogy eldöntöttük, milyen utasításokat kívánunk a számítógéppel végrehajtatni, kikerestük az utasításkódokat, kiszámoltuk a tárcímeket, operandusokat, byte-onként beírtuk az egészet a tárba, majd a vezérlést a tárban elhelyezett program elejére adtuk.

A továbbiakban assembly nyelven fogjuk írni programjainkat. Assemblerrel a programozás ugyanúgy történik, mint gépi kódban, a különbség csupán annyi, hogy a leírt kellemetlenül hosszadalmas tevékenységeket az assembler fordítóprogram automatikusan elvégzi. Lássuk tehát, hogyan lehet az előbbi, háromsoros

```
LD A,41H
LD (3E20H),A
JP 7005H
```

programot az EDI assembler/editor segítségével összeállítani, gépi kódra fordítani és futtatni. Megjegyezzük, hogy az EDI program kezelési utasítása megtalálható könyvünk Függelékében.

Az EDI assembler/editornak olyan formában kell beírnunk a programot, hogy abból elő lehessen állítani a gépi kódot. Ehhez mindenekelőtt azt kell megmondanunk, hogy a tár mely részén kívánjuk elhelyezni a programot, tehát meg kell adnunk a program kezdőcímét. Erre szolgál az ORG utasítás, amelyben előző példánknál maradva a 7000H kezdőcímet adjuk meg. A program végét az assembler/editor számára az END utasítás jelzi:

```
1                    ;1. LISTA
2                          ORG  7000H
3 7000 3E41                LD   A,41H
4 7002 32203E              LD   (3E20H),A
5 7005 C30570              JP   7005H
6                          END
```

Az ORG és END utasítások a program gépi kódra fordításához adnak információt, ezekből nem lesz a program futtatása során végrehajtandó utasítás. Az ilyen utasításokat pszeudo utasításoknak nevezzük.

Amikor gépi kódban írtuk a harmadik utasítást, ki kellett számolnunk azt a tárcímet, amelyre ez az utasítás ugrik, vagyis meg kellett határoznunk, hogy a tárban hova kerül ennek az utasításnak az első byte-ja. Assemblyben viszont nem kell a címek kiszámolásával bajlódnunk, a BASIC sorszámaihoz hasonlóan címkéket – szimbolikus címeket – adhatunk az utasításoknak. Így az önmagára ugró JP utasítást pl.:

```
HUROK :          JP HUROK
```

alakban írhatjuk fel. A HUROK címke számértékét, példánkban a 7005H értéket, az assembler/editor a fordítás során az ORG utasításban megadott kezdőcím alapján kiszámolja, és a gépi kódba a címke helyére beírja.

Így összeállítottuk az EDI assembler/editor számára programunkat:

```
1                    ;2. LISTA
2                            ORG  7000H
3 7000 3E41                  LD   A,41H
4 7002 32203E                LD   (3E20H),A
5 7005 C30570        HUROK:  JP   HUROK
6                            END
```

Megjegyezzük, hogy az önmagára ugró HUROK: JP HUROK utasítás helyett más helyekre is lehetnek ugrani, például a JP 1A19H utasítással visszatérhetünk a BASIC értelmezőhöz, vagy a JP 4300H utasítással újra beléphetünk az EDI assembler/editor programba.

Következő feladatunk az összeállított program begépelése. Ehhez először is be kell töltenünk az EDI assembler/editort. A sikeres betöltés után megjelenő ?> üzenetre az E parancsot adva sorszámokat kapunk, amelyekre egymás után beírjuk programsorainkat, majd az utolsó sor után csak ". " karaktert tartalmazó sor beírásával kilépünk az EDI szövegszerkesztő üzemmódjából:

```
?>  E
1   ORG 7000H
2   LD A,41H
3   LD (3E20H),A
4   HUROK: JP HUROK
5   END
6
```

A beírt sorokat T, majd P5 parancsokkal ellenőrizhetjük. Ha hibát követtünk el a program beírásakor, akkor a sormutatót állító parancsokkal kikeressük a javítandó sort, majd a szövegszerkesztő parancsok segítségével elvégezzük a szükséges cserét, törlést, beszúrást.

Ha több hibát nem találunk, akkor a program gépi kódra fordítása (assemblálása) következik. A fordítás a ?>A paranccsal történik, a fordítási lista az OPTION? kérdésre adott választól függően a képernyőre vagy a nyomtatóra kerül (adjunk most V választ). A fordítás után megjelenő lista tartalmazza az általunk írt assembler programot – a forrásprogramot – és előtte a gépi kódú programot – az object programot – ugyanúgy, mint ahogyan azt az előbb byte-onként mi magunk meghatároztuk. Mindazt, amit az előbb byte-onként kellett kikeresnünk, kiszámolnunk, a fordítóprogram egyetlen parancs megadásával automatikusan meghatározza. A program listáján kívül az ?>S paranccsal a címkéket és az azokhoz tartozó tárcímeket is kiírathatjuk.

Ha a fenti forrásprogramot kiegészítjük a LOAD pszeudo utasítással, akkor az EDI assembler/editor a fordítás során előállított gépi kódot a LOAD utasításban megadott címtől elhelyezi a tárban:

```
1                    ;3. LISTA
2                            ORG  7000H
3                            LOAD 7000H
4 7000 3E41                  LD   A,41H
5 7002 32203E                LD   (3E20H),A
6 7005 C30570        HUROK:  JP   HUROK
7                            END
```

Miután a lefordított programot a tárban elhelyeztük, a gépi kódú monitor G parancsához hasonló ?>G paranccsal megkezdhetjük a program futtatását. A ?>G parancs után megjelenő ADDR> kérdésre a kezdőcímet, példánkban a 7000H értéket kell megadni,az ezt követő BKPT> (breakpoint) kérdésre csak akkor kell NEWLINE-tól különböző választ adni, ha a program futását valamely ponton meg kívánjuk szakítani (adjunk most NEWLINE választ). Fáradságos munkánk eredményeképpen a képernyő közepén újra megjelenik az "A" betű (ha nem, akkor keressünk hibát!), és most is a RESET gomb megnyomásával szabadulhatunk ki a JP utasítás által okozott végtelen ciklusból. Ezután a SYSTEM parancsot, majd a *? kérdésre a 17152 kezdőcímet megadva újra beléphetünk az EDI programba. A P6, ill. a Q7000H parancsokkal láthatjuk, hogy a forrásprogram, ill. a gépi kód megőrződött (a forrásprogramot a K paranccsal, a gépi kódot az F paranccsal tüntethetjük el).

Pszeudo utasítások

Az EDI assembler/editornak az ORG, LOAD és END utasításokon kívül további néhány pszeudo utasítása is van.*

A DB pszeudo utasítással (Define Bytes) adatokat lehet a tárban elhelyezni. Pl. az előző programban szereplő 41H értéket ahelyett, hogy közvetlen adatként töltenénk az A regiszterbe, a DB 41H utasítással lerakhatjuk a tárba. A DB utasításnak címkét adva az A regiszterbe erre a címkére hivatkozva tölthetjük be a lerakott értéket:

```
1                   ;4. LISTA
2                           ORG  7000H
3                           LOAD 7000H
4 7000 3A0970               LD   A,(BETU)
5 7003 32203E               LD   (3E20H),A
6 7006 C30670       HUROK:  JP   HUROK
7 7009 41           BETU:   DB   41H
8                           END
```

A DB utasításnak több, egymástól vesszővel elválasztott operandusa is lehet. Az operandusok decimális és hexadecimális számok, egy vagy több byte hosszúságú karaktersorozatok lehetnek (a hexadecimális konstansokat 0-val kell kezdeni, ha nem a 0...9 számjegyek valamelyikével kezdődnek):

```
DB          7FH,25,'JO REGGELT'
```

Az assembler/editor a DB pszeudo utasítás által definiált értékeket sorban elhelyezi a tárban.

A DW utasítással (Define Word) 16 bites értékeket lehet – a szokásos fordított sorrendben – a tárban elhelyezni. A DW 3E20H utasítás hatására az alsó címen 20H, a felső címen 3EH fog tárolásra kerülni.

A DS pszeudo utasítással (Define Storage) pufferek, üzenetek, számítások stb. céljaira lehet tárterületet lefoglalni. Pl. a

* Az assembler/editor parancsait a függelék teljes egészében tartalmazza.

BUFFER: DS 30

utasítás hatására a BUFFER címkétől 30 byte üres terület következik.

Az EQU pszeudoutasítással (EQate) címkékhez számértéket lehet hozzárendelni. Pl. a BETU: EQU 41H utasítás hatására BETU értéke 41H lesz, így az

```
1                    ;5. LISTA
2                            ORG  7000H
3                            LOAD 7000H
4 7000 3E41                  LD   A,BETU
5 7002 32203E                LD   (3E20H),A
6 7005 C30570        HUROK:  JP   HUROK
7                    BETU:   EQU  41H
8                            END
```

program ugyanúgy a 41H értéket tölti az A regiszterbe, mint az előző program, de míg az előbb egy tárcímről történt a betöltés, most közvetlen adattal töltünk. Az EQU utasítással tárcímekhez is rendelhetünk címkét, pl. ha a VIDEO: EQU 3E20H utasítással definiáljuk a képernyő közepét, akkor az előző program képernyőre író utasítását LD VIDEO,A alakban írhatjuk fel.

Mindezek felhasználásával az eredetileg háromsoros, a képernyő közepére egy "A" betűt író programunkat nyolc sorban írhatjuk meg:

```
1                    ;6. LISTA
2                            ORG  7000H
3                            LOAD 7000H
4 7000 3E41                  LD   A,BETU
5 7002 32203E                LD   (VIDEO),A
6 7005 C30570        HUROK:  JP   HUROK
7                    BETU:   EQU  41H
8                    VIDEO:  EQU  3E20H
9                            END
```

amivel látszólag elbonyolítottuk a dolgot, viszont hosszabb programoknál komoly segítséget jelentenek a most megismert programozási segédeszközök.

A programsorokat – a BASIC-ből ismert REM utasításhoz hasonlóan – megjegyzésekkel, magyarázatokkal egészíthetjük ki. Ezeket az utasítás után ";" karakterrel kezdve lehet elhelyezni, de lehetnek csak megjegyzést tartalmazó programsorok is. Előző programunkat így egészíthetjük ki:

```
 1                   ;7. LISTA
 2                   ;EZ A PROGRGM EGY "A" BETUT IR
 3                   ;A KEPERNYO KOZEPERE
 4
 5                           ORG  7000H   ;KEZDOCIM
 6                           LOAD 7000H   ;BETOLTESI CIM
 7 7000 3E41                 LD   A,BETU  ;A-BA "A" KODJA
 8 7002 32203E               LD   (VIDEO),A;"A"-T KIIRJUK
 9 7005 C30570       HUROK:  JP   HUROK   ;ITT MEGALLUNK
10                   BETU:   EQU  41H     ;"A" ASCII KODJA
11                   VIDEO:  EQU  3E20H   ;KEPERNYO KOZEPE
12                           END          ;PROGRAM VEGE
```

Hosszabb programoknál nagyon fontosak a megjegyzések, magyarázatok. Ezt mindenki tudja, aki már keresett assembly programban hibát, vagy aki próbálta már saját régebbi programját újra megérteni.

Egy komolyabb program

A következő program egy hosszabb szöveget ír a képernyő utolsó sorába. Mint tudjuk, a képernyőcímek kezdete 3C00H, amihet 15*64-et kell hozzáadni, hogy a képernyő utolsó sorának első karaktercímét kapjuk. A kiírandó szöveg után hexadecimális 0-t teszünk, erről fogjuk tudni, hogy a szöveg végéhez érkeztünk. A program karakterenként viszi át a szöveget a tárból a képernyőre. A HL regiszterpár mindig a szöveg éppen átmozgatásra kerülő karakterére, az IX indexregiszter pedig mindig a képernyőre, a következő karakter helyére mutat.

```
 1                  ;8. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4                  VIDEO:  EQU  3C00H  ;KEPERNYO KEZDETE
 5 7000 211770              LD   HL,SZOVEG;MUTATO SZOVEGRE
 6 7003 DD21C03F            LD   IX,VIDEO+960
 7                  ;                   MUTATO KEPERNYORE
 8 7007 7E          CIKLUS: LD   A,(HL) ;KOVETKEZO KARAKTER
 9 7008 FE00                CP   0      ;VEGE?
10 700A 2809                JR   Z,VEGE ;UGRIK, HA KESZ
11 700C DD7700              LD   (IX+0),A;KARAKTER KIIRAS
12 700F 23                  INC  HL     ;KOVETKEZO KARAKTER
13 7010 DD23                INC  IX     ;KOVETKEZO POZICIO
14 7012 C30770              JP   CIKLUS ;FOLYTATAS
15 7015 18FE       VEGE:    JR   VEGE   ;VEGTELEN HUROK
16 7017 455A2045   SZOVEG:  DB   'EZ EGY KOMOLY PROGRAM'
16 701B 4759204B
16 701F 4F4D4F4C
16 7023 59205052
16 7027 4F475241
16 702B 4D
17 702C 00                  DB   00H
18                          END
```

Az ORG utasítás a programszámlálót a 7000H értékről indítja, a LOAD utasítás hatására a program e címtől kerül betöltésre. A következő két utasítás a HL regiszterpárba a szöveg első karakterének címét, az IX indexregiszterbe a képernyő kiválasztott része első karakterének címét tölti. A CIKLUS címkéjű utasítás az A regiszterbe betölti a szöveg következő karakterét. Ez az utasítás először az első karaktert tölti be, de később, ahogy növeljük a HL regiszterpár tartalmát, mindig a következő karakter fog az A regiszterbe kerülni. A karakter betöltése után megvizsgáljuk, hogy a szöveg végét jelző nulla került-e betöltésre. Ha igen, akkor az adatmozgatás végére értünk, és vezérlésátadás következik a VEGE címkére. Ha nem, akkor a Z jelzőbit nulla, így a program a JR Z,VEGE utasításról a következő utasításra lép, a szöveg következő karakterét tárolja az IX által mutatott helyre. Ezután eggyel növeli a szövegre mutató HL regiszterpár és a képernyő kiválasztott helyére mutató IX indexregiszter tartalmát, majd visszaugrik a CIKLUS címkére, és addig ismétli az előbbi utasításokat, amíg a szöveg végére nem jut.

Ugyanezt a programot sokféleképpen, ennél sokkal ügyesebben és sokkal ügyetlenebbül is meg lehet írni, de első próbálkozásnak ez a változat sem rossz. Olvasóinkat arra kérjük, hogy könyvünk további fejezeteit – ha van számítógép a közelükben – példáinkat kipróbálva kövessék.

Programbelövés, hibakeresés

Kivételes esetben előfordulhat ugyan, hogy assembly nyelven írt programunk első próbálkozásra hibátlanul lefut, az esetek többségében azonban ehelyett váratlan eredmények születnek.

A programok belövését segíti, hogy – akár az EDI assembler/editort, akár a gépi kódú monitort használjuk – a G parancs megadásakor kijelölhetünk a programban olyan pontot (töréspontot), amelynél a végrehajtás megszakad, és a regisztertartalmak megőrzése után az EDI, ill. a gépi kódú monitor visszakapja a vezérlést. Miután a program végrehajtását így felfüggesztettük, kiírathatjuk a regiszterek tartalmát, a tárrekeszek tartalmát, módosíthatjuk a program utasításait, adatait, majd ugyaninnen vagy a program bármely más pontjától folytathatjuk a végrehajtást.

Pl. ha az előző, szövegkiíró programot az EDI assembler/editor G parancsával indítjuk, az ADDR> kérdésre 7000H-t válaszolunk, a BKPT> kérdésre pedig a 7012H-t, a JP CIKLUS utasítás első byte-jának címét adjuk meg, akkor a program a szöveg első karakterét átmozgatja a képernyőre, növeli a HL regiszterpár és az IX indexregiszter tartalmát, de a következő karakter átmozgatása előtt megáll. Ekkor az X paranccsal a regiszterek tartalmát kiíratva azt látjuk, hogy az A regiszterben 45H van, ami a szöveg első karakterének, az "E" betűnek az ASCII kódja, a PC programszámláló a töréspont helyét mutatja, az F regiszterben a Z jelzőbit nulla, hiszen még nem értünk a szöveg végére. A HL regiszterpár és az IX indexregiszter tartalmát az INC utasításokkal egyszer már növeltük, így azok most a szöveg második karakterére, ill. a képernyő kiválasztott részének második pozíciójára mutatnak. A Q paranccsal a tár bármely részét kiírathatjuk a képernyőre (igaz, hogy az EDI assembler/editorba, ill. a gépi kódú monitorba visszatéréskor megváltozott a képernyő tartalma, így ezzel nem lehet ellenőrizni, hogy mit írtunk a képernyőre). Az M paranccsal gépi kód szinten átírhatjuk a program utasításait, módosíthatjuk az adatokat, végül újabb G paranccsal folytathatjuk a program futtatását.

8. fejezet

Adatmozgatási példák

Most példák következnek, megismerkedünk néhány adatmozgatási probléma Z80 assembly nyelvű megoldásával. A példák között lesznek olyanok, amelyekben a megoldandó feladat önmagában nem túlságosan érdekes, viszont a bennük bemutatott programozási módszerek alapvető fontosságúak.

Byte-ok mozgatása

A Z80 mikroprocesszor regisztereibe az LD utasítás számtalan formájával tölthetünk adatokat. Első példánkban közvetlen adattal az A regiszterbe 1-et, a B regiszterbe 2-t, az A′ regiszterbe 3-at és a B′ regiszterbe 4-et töltünk.

Az A...L és A′...L′ két regiszterkészlet közül mindig csak az egyiket lehet használni, az átkapcsolás az EX AF,AF′ és EXX utasításokkal történik. (Megjegyezzük, hogy az EX AF,AF′ utasítást az EDI assembler/editor EX AF,AF alakban kezeli, az ′ (aposztrof) ugyanis más célra van fenntartva.)

```
 1                    ;9. LISTA
 2                            ORG  7000H
 3                            LOAD 7000H
 4 7000 3E01                  LD   A,1
 5 7002 0602                  LD   B,2
 6 7004 08                    EX   AF,AF
 7 7005 D9                    EXX
 8 7006 3E03                  LD   A,3
 9 7008 0604                  LD   B,4
10 700A 08                    EX   AF,AF
11 700B D9                    EXX
12 700C 18FE          VEGE:   JR   VEGE
13                            END
```

Ha a programot a G paranccsal úgy indítjuk el, hogy a VEGE: JR VEGE utasításnál töréspontot helyezünk el, akkor a futtatás felfüggesztése után az X paranccsal meggyőződhetünk arról, hogy a regiszterekben valóban az általunk betöltött értékek találhatók.

Töltsünk most a tárból a B, C, D regiszterekbe adatokat! Közvetlen címzéssel a tárból csak az A regiszterbe lehet adatot tölteni, ezért mindegyik adatot először az A regiszterbe visszük, majd onnan átmozgatjuk a kívánt regiszterbe:

```
 1                    ;10. LISTA
 2                            ORG  7000H
 3                            LOAD 7000H
 4 7000 3A0E70                LD   A,(EGY)
 5 7003 47                    LD   B,A
 6 7004 3A0F70                LD   A,(KETTO)
```

```
 7 7007 4F                LD    C,A
 8 7008 3A1070            LD    A,(HAROM)
 9 700B 57                LD    D,A
10 700C 18FE      VEGE:   JR    VEGE
11 700E 01        EGY:    DB    1
12 700F 02        KETTO:  DB    2
13 7010 03        HAROM:  DB    3
14                        END
```

Ha az adatok a tárban egymás után vannak elhelyezve, akkor megtehetjük, hogy valamelyik indexregiszterbe betöltjük az első adat címét, és a regisztereket indexelt címzéssel töltjük:

```
 1                ;11. LISTA
 2                         ORG  7000H
 3                         LOAD 7000H
 4 7000 DD210F70           LD   IX,ADAT
 5 7004 DD4600             LD   B,(IX+0)
 6 7007 DD4E01             LD   C,(IX+1)
 7 700A DD5602             LD   D,(IX+2)
 8 700D 18FE      VEGE:    JR   VEGE
 9 700F 010203    ADAT:    DB   1,2,3
10                         END
```

Az LD IX,ADAT utasítás az ADAT címkéjű, adatokat tároló pszeudoutasítás címét tölti az IX indexregiszterbe. A DB utasítás a három adatot egymás utáni három byte-on tárolja.

Ugyanennek a feladatnak harmadik megoldása, hogy a HL regiszterpárba töltjük az első adat címét, a HL által mutatott címről töltjük a regisztereket, közben INC utasítással növeljük HL tartalmát:

```
 1                ;12. LISTA
 2                         ORG  7000H
 3                         LOAD 7000H
 4 7000 210A70             LD   HL,ADAT
 5 7003 46                 LD   B,(HL)
 6 7004 23                 INC  HL
 7 7005 4E                 LD   C,(HL)
 8 7006 23                 INC  HL
 9 7007 56                 LD   D,(HL)
10 7008 18FE      VEGE:    JR   VEGE
11 700A 010203    ADAT:    DB   1,2,3
12                         END
```

Hogy a sokféle lehetőség közül melyiket érdemes választani, arra nehéz receptet adni. Hiába minősítjük legjobbnak mondjuk a harmadik megoldást, ha az adott programozási helyzetben a HL regiszterpárt más célra használjuk, vagy ha az adatok a tárban nem egymás után vannak lerakva.

A regisztereket közvetlen címzéssel töltő program párja a következő, ugyanazokat a regisztereket tároló program. Ezt kivételesen nem a 7000H címtől kezdjük, hogy az előző, regisztereket töltő programot és ezt egyidejűleg elhelyezhessük a tárban, és hogy lássuk, megtörténik-e a regisztertartalmak tárolása:

```
 1                ;13. LISTA
 2                         ORG  7100H
 3                         LOAD 7100H
 4 7100 7A                 LD   A,D
 5 7101 321071             LD   (HAROM),A
 6 7104 79                 LD   A,C
 7 7105 320F71             LD   (KETTO),A
```

```
 8 7108 78                LD   A,B
 9 7109 320E71            LD   (EGY),A
10 710C 18FE      VEGE:   JR   VEGE
11                EGY:    DS   1
12                KETTO:  DS   1
13                HAROM:  DS   1
14                        END
```

Írjuk be az előző, regisztereket töltő programok valamelyikét és ezt a regisztereket tároló programot, fordítsuk le mindkettőt, majd futtassuk le először az első programot a VEGE címkéig, majd a második programot ugyancsak a VEGE címkéig, majd írassuk ki a tárnak azt a részét, amely a második programban az adatokat tárolja, vagyis – ha az EDI assembler/editorral dolgozunk – a két program lefordítása után adjuk meg sorra a

```
?>G
ADDR>7000H
BKPT >700CH
?>G
ADDR>7100H
BKPT >710CH
?>Q7100H
```

parancsokat.

A 16 bites adatmozgatási lehetőségek korlátozottabbak, pl. regiszterpárt lehet tölteni közvetlen adattal, de nem lehet közvetlen címzéssel. Ez utóbbit helyettesíthetjük két 8-bites adatmozgató utasítással. A regiszterpárok közötti adatmozgatást a PUSH és POP utasításokkal, a veremterületen keresztül valósíthatjuk meg, amint erre a későbbiekben fogunk példákat látni.

Tárkitöltés

Fontos feladat a tár meghatározott részének adott karakterrel kitöltése, ezzel a módszerrel lehet pl. pufferterületeknek, táblázatoknak kezdőértéket adni. Következő programunk, amely a képernyő első négy sorát csillagokkal tölti ki, az előző példáknál látványossabbnak igérkezik.

A programban az A regiszterbe töltjük a "*" karaktert, a HL regiszterbe visszük a képernyő kezdőcímét, és a B regiszterben számláljuk, hogy a képernyő hány pozícióját kell kitöltenünk. Azért választottuk, hogy négy sort, vagyis 4*64 = 256 pozíciót töltünk ki, mert egy egybyte-os regiszterben 0-tól 255-ig lehet számlálni, így az ebben a programban alkalmazott módszerrel éppen négy sornyi csillagot lehet kiírni. A program kicsit bonyolultabb az előzőnél, ezért most minden sorhoz írunk magyarázatot:

```
 1                ;14. LISTA
 2                        ORG  7000H   ;KEZDOCIM
 3                        LOAD 7000H   ;BETOLTESI CIM
 4 7000 3E2A              LD   A,'*'   ;TOLTO KARAKTER
 5 7002 21003C            LD   HL,3C00H;KEPERNYO KEZDETE
 6 7005 0E00              LD   C,0     ;256 CSILLAG
 7 7007 77        CIKL:   LD   (HL),A  ;BYTE KIIRASA
 8 7008 23                INC  HL      ;KOVETKEZO POZICIO
 9 7009 0D                DEC  C       ;SZAMLALO CSOKKEN
10 700A 20FB              JR   NZ,CIKL;NEM KESZ, VISSZA
11 700C 18FE      VEGE:   JR   VEGE    ;ITT MARAD
12                        END
```

A program az A regiszterbe helyezett "*" karaktert a HL regiszterpár által mutatott tárcímre tárolja, így a HL regiszterpár tartalmának növelésével ugyanaz a csillag sorra a képernyő egymás utáni pozícióira kerül kiírásra. A C regiszterbe nullát töltöttünk, ami az első DEC C utasítás hatására 255-re „csökken", így már először is teljesül az NZ feltétel, és a program csak akkor lép tovább a VEGE címkéjű utasításra, ha a C regiszterbe újra nulla kerül. Ez az apró trükk biztosítja, hogy pontosan 256 csillag kerüljön a képernyőre.

Ugyanezt a programot egy kicsit rövidebben is meg lehet írni, ha számlálóként a B regisztert használjuk, ugyanis a DJNZ utasítás csökkenti a B regiszter tartalmát, majd feltételes relatív ugrást hajt végre. Így az előző program DEC C és JR NZ,CIKLUS utasításait egyetlen utasítással, a DJNZ CIKLUS utasítással helyettesíthetjük.

A program képernyőkitöltő részét négyszer ismételve a teljes képernyőt teleírhatjuk csillagokkal. Egy újabb számlálóval, mondjuk a C regiszterrel biztosítjuk, hogy négyszer fusson a kitöltő rész:

```
 1                    ;15. LISTA
 2                          ORG  7000H   ;KEZDOCIM
 3                          LOAD 7000H   ;BETOLTESI CIM
 4 7000 3E2A                LD   A,'*'   ;TOLTO KARAKTER
 5 7002 21003C              LD   HL,3C00H;KEPERNYO KEZDETE
 6 7005 0E04                LD   C,4     ;4-SZER ISMETLUNK
 7 7007 0600          UJRA: LD   B,0     ;256 CSILLAG
 8 7009 77            CIKL: LD   (HL),A  ;BYTE KIIRASA
 9 700A 23                  INC  HL      ;KOVETKEZO POZICIO
10 700B 10FC                DJNZ CIKL    ;NEM KESZ, VISSZA
11 700D 0D                  DEC  C       ;C-T CSOKKENTI
12 700E 20F7                JR   NZ,UJRA;KOVETKEZO NEGYED
13 7010 18FE          VEGE: JR   VEGE    ;ITT MARAD
14                          END
```

Aki BASIC-ben vagy más magasszintű nyelven írt már programokat, csodálkozhat, hogy miért bonyolítottuk el a két ciklussal, két számlálóval ennyire a dolgot, amikor regiszterpárt is lehet számlálónak használni, és így egy ciklussal is meg lehet írni ugyanezt. Elsősorban azért érdemes az előző megoldást választani, mert a regiszterpárt növelő, csökkentő utasítások nem állítják a jelzőbiteket, így külön trükkökre van szükség ahhoz, hogy a regiszterpár csökkentése után a jelzőbitek mutassák, hogy nulla lett-e a regiszterpár tartalma:

```
 1                    ;16. LISTA
 2                          ORG  7000H   ;KEZDOCIM
 3                          LOAD 7000H   ;BETOLTESI CIM
 4 7000 162A                LD   D,'*'   ;TOLTO KARAKTER
 5 7002 21003C              LD   HL,3C00H;KEPERNYO KEZDETE
 6 7005 010004              LD   BC,1024;1024 CSILLAG
 7 7008 72            CIKL: LD   (HL),D  ;BYTE KIIRASA
 8 7009 23                  INC  HL      ;KOVETKEZO POZICIO
 9 700A 0B                  DEC  BC      ;SZAMLALO CSOKKEN
10 700B 78                  LD   A,B     ;NULLA-E?
11 700C B1                  OR   C       ;JELZOBIT ALLITAS
12 700D 20F9                JR   NZ,CIKL;NEM KESZ, VISSZA
13 700F 18FE          VEGE: JR   VEGE    ;ITT MARAD
14                          END
```

Azt, hogy a BC regiszterpárban nulla van-e, úgy vizsgáljuk, hogy a B és C regiszterekkel logikai OR műveletet végzünk. Logikai műveleteket viszont csak az A regiszterben lehet végezni, ezért a B regisztert átvisszük az A regisz-

terbe, és így OR-oljuk hozzá a C regisztert. Ha B-ben is, C-ben is minden bit nulla, akkor az A regiszterben végzett OR művelet eredménye nulla, így a Z jelzőbit értéke 1 lesz. (A töltő karaktert is azért helyeztük a D regiszterbe, mert az A regiszter a logikai műveletek miatt foglalt.)

Vegyük elő az utasítástáblázatot, és számoljuk ki az előző két program futási idejét! Az első változat mintegy kétszer olyan gyors, mint az egy byte-tal rövidebb második!

Következő példánkban függőleges egyenest rajzolunk a képernyőre. A program bemenő adatai az oszlop száma, a kezdősor száma és az egyenes hosszúsága. Egy sorban 64 karakter van, tehát a kezdő pozíciót úgy kapjuk meg, hogy a képernyő kezdőcíméhez hozzáadjuk az oszlop számát és a kezdősor számának 64-szeresét. A 64-gyel szorzás egyszerűen visszavezethető 2-vel szorzások egymásutánjára, tehát ismételt összeadásra.

```
 1                    ;17. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 DD212370            LD   IX,PARAM;ADATOK CIME
 5 7004 DD6E01              LD   L,(IX+1);VONAL ELSO SORA
 6 7007 2600                LD   H,0
 7 7009 114000              LD   DE,64  ;SOR HOSSZA
 8 700C 0606                LD   B,6
 9 700E 29          RAJZ1:  ADD  HL,HL  ;SZORZAS 64-GYEL
10 700F 10FD                DJNZ RAJZ1  ;TOVABB, HA B=0
11 7011 DD4E00              LD   C,(IX+0);OSZLOP SZAMA
12 7014 09                  ADD  HL,BC  ;KEZDO POZICIO
13 7015 01003C              LD   BC,3C00H;KEPERNYO KEZDETE
14 7018 09                  ADD  HL,BC  ;KEPERNYO POZICIO
15 7019 DD4602              LD   B,(IX+2);VONAL HOSSZA
16 701C 36BF       RAJZ2:   LD   (HL),0BFH;KARAKTER KIIRASA
17 701E 19                  ADD  HL,DE  ;KOVETKEZO SOR
18 701F 10FB                DJNZ RAJZ2
19 7021 18FE       VEGE:    JR   VEGE   ;ITT MARAD
20 7023 19030A     PARAM:   DB   25,3,10;OSZLOP,SOR,HOSSZ
21                          END
```

A PARAM címkéjű pszeudo utasításban megadott oszlopszámot 0-tól 63-ig, a sorszámot 0-tól 15-ig, a vonal hosszúságát 0-tól 15-ig változtathatjuk, de a sorszám és a vonal hosszúságának összege 15-nél több nem lehet.

Blokkáthelyezés

Gyakran van szükség arra, hogy a tár egyik részének tartalmát átmozgassuk a tár másik részére. Ilyen műveletet végzünk pl., ha egy karaktersorozatot kiírunk a képernyőre, vagy ha adatainkat táblázatba rendezzük.

A következő példában a tár első 1024 byte-ját fogjuk a képernyőre kiírni. A HL regiszterpárral a forrástárterületre, a DE regiszterpárral a célterületre mutatunk, a BC regiszterpárba pedig az átmozgatandó byte-ok számát töltjük. A HL regiszterpár által mutatott címről bármely regiszterpárt lehet tölteni, viszont a DE által mutatott címre csak az A regisztert lehet tárolni, ezért az adatmozgatást az A regiszteren keresztül fogjuk végezni (egyébként nincs is más szabad regiszter).

A HL regiszterpárba a nullát, vagyis a tár elejének címét töltjük, a DE regiszterpárba a képernyő kezdetének címe, 3C00H kerül, a BC regiszter-

párba pedig az átmozgatandó byte-ok számát, az 1024-et visszük. A számláló csökkentésekor a már ismert LD és OR trükkel fogjuk a jelzőbiteket állítani.

```
 1                    ;18. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 210000              LD   HL,0    ;FORRAS CIM
 5 7003 11003C              LD   DE,3C00H;CEL CIM
 6 7006 010004              LD   BC,1024;BYTE-OK SZAMA
 7 7009 7E            CIKL: LD   A,(HL) ;FORRAS BYTE
 8 700A 12                  LD   (DE),A ;BYTE KIIRASA
 9 700B 23                  INC  HL     ;KOVETKEZO BYTE
10 700C 13                  INC  DE     ;KOVETKEZO BYTE
11 700D 0B                  DEC  BC     ;SZAMLALO CSOKKEN
12 700E 78                  LD   A,B    ;NULLA-E?
13 700F B1                  OR   C      ;JELZOBIT ALLITAS
14 7010 20F7                JR   NZ,CIKL
15 7012 18FE         VEGE:  JR   VEGE
16                          END
```

Ha lefuttatjuk ezt a programot, a BASIC értelmező elejét fogjuk a képernyőn látni. A sokféle utasításnak, adatnak sokféle alfanumerikus és grafikus karakter felel meg, első pillantásra úgy tűnhet, hogy véletlenszerűen előállított karakterekkel töltöttük meg a képernyőt, de ha jobban megnézzük, ismerős dolgokat is találunk – pl. a gép bekapcsolásakor megjelenő "READY" szöveget.

Próbáljuk meg ugyanezt a programot különböző forráscímekkel lefuttatni! Felismerhetjük a BASIC kulcsszavakat és általában minden olyan szöveget, ami a képernyőn megjelenhet, amikor BASIC-ben programozunk.

Maga a program messze ügyesebben is megírható! Pontosan erre találták ki az LDIR blokkmozgató utasítást:

```
 1                    ;19. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 210000              LD   HL,0    ;FORRAS CIM
 5 7003 11003C              LD   DE,3C00H;CEL CIM
 6 7006 010004              LD   BC,1024;BYTE-OK SZAMA
 7 7009 EDB0          CIKL: LDIR         ;BLOKK MOZGATAS
 8 700B 18FE          VEGE: JR   VEGE
 9                          END
```

Amint látjuk, az LDIR utasítás a HL regiszterpárban a forrásterület kezdetének címét, a DE regiszterpárban a célterület kezdetének címét, a BC regiszterpárban az átmozgatandó byte-ok számát várja. A blokkmozgatás után (győződjünk meg róla!) a HL és DE regiszterpárok az utolsó átmozgatott byte utáni címre mutatnak, a BC regiszterpárban pedig nulla van.

Az LDDR utasítás ugyanúgy működik, mint az LDIR utasítás, de a HL regiszterpárral a forrásterület végére, a DE regiszterpárral a célterület végére kell mutatni. Az adatmozgatás az utolsó byte-nál kezdődik és ez elsőnél fejeződik be. A következő program az LDDR utasítás felhasználásával a képernyőt egy sorral lejjebb mozgatja:

```
 1                    ;20. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 21BF3F              LD   HL,3FBFH;UTOLSO ELOTTI SOR
 5 7003 11FF3F              LD   DE,3FFFH;UTOLSO SOR
 6 7006 01C003              LD   BC,960 ;BYTE-OK SZAMA
 7 7009 EDB8                LDDR        ;BLOKK MOZGATAS
 8 700B 18FE          VEGE: JR   VEGE
 9                          END
```

(Írjuk meg ugyanezt BASIC nyelven, és hasonlítsuk össze a két program sebességét! BASIC program beírása után az EDI-t újra be kell tölteni!)

Az LDI és LDD utasítások majdnem ugyanúgy működnek, mint az LDIR és LDDR utasítások, de minden átmozgatott byte után a következő utasításra adják a vezérlést. Ez lehetőséget ad arra, hogy az átmozgatott adatokkal egyenként különböző műveleteket végezzünk, pl. NEWLINE vagy null karakterig folytassuk az adatmozgatást, vagy csak bizonyos típusú karaktereket mozgassunk át. Az adatmozgatás folytatása céljából vissza kell ugrani az LDI, ill. LDD utasításra.

A tárkiíró program LDIR utasítását a következőképpen lehet LDI utasítással helyettesíteni:

```
 1                    ;21. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 210000              LD   HL,0    ;FORRAS CIM
 5 7003 11003C              LD   DE,3C00H;CEL CIM
 6 7006 010004              LD   BC,1024;BYTE-OK SZAMA
 7 7009 EDA0          CIKL: LDI          ;BLOKK MOZGATAS
 8 700B EA0970              JP   PE,CIKL;TOVABB, HA BC<>0
 9 700E 18FE          VEGE: JR   VEGE
10                          END
```

A JP utasításban azért a PE feltétel szerint ugrunk, mert az LDI és LDD utasítások után nem a Z jelzőbit, hanem a P/V jelzőbit mutatja, hogy a BC regiszterpárban a byte-ok száma nullára csökkent. Ha BC még nem érte el a nullát, akkor P/V értéke 1 (PE feltétel), ha BC=0, akkor P/V törlődik (PO feltétel).

A blokkmozgatás közben végezhető egyéb tevékenységeket mutatja a következő program, amely LDI utasítással a képernyő felső felét, majd LDD utasítással a képernyő alsó felét tölti ki, és minden byte átvitele után a LASSAN szubrutinban várakozik:

```
 1                    ;22. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 210000              LD   HL,0    ;FORRAS CIM
 5 7003 11003C              LD   DE,3C00H;CEL CIM
 6 7006 010002              LD   BC,512 ;BYTE-OK SZAMA
 7 7009 EDA0          C1:   LDI          ;EGY BYTE MOZGATASA
 8 700B E21470              JP   PO,C2  ;UGRIK, HA KESZ
 9 700E CD2A70              CALL LASSAN ;VARAKOZAS
10 7011 C30970              JP   C1     ;FOLYTATAS
11 7014 21FF03       C2:    LD   HL,3FFH;UJ FORRAS CIM
12 7017 11FF3F              LD   DE,3FFFH;UJ CEL CIM
13 701A 010002              LD   BC,512 ;UJRA 512 BYTE
14 701D EDA8         C3:    LDD         ;EGY BYTE HATULROL
15 701F E22870              JP   PO,C4  ;UGRIK, HA KESZ
16 7022 CD2A70              CALL LASSAN ;VARAKOZAS
17 7025 C31D70              JP   C3     ;FOLYTATAS
18 7028 18FE         C4:    JR   C4     ;VEGE
19 702A 3E80         LASSAN: LD  A,80H  ;SZAMLALO
20 702C 3D           C5:    DEC  A      ;CSOKKENTES
21 702D 20FD                JR   NZ,C5  ;TOVABB SZAMLAL
22 702F C9                  RET         ;VISSZATERES
23                          END
```

A programban minden egyes byte átmozgatása után, tehát az LDI és LDD utasítások minden egyes végrehajtása után meghívjuk a LASSAN szubrutint, amely egyszerű számlálással várakozik. Számlálónak azért az A regisztert

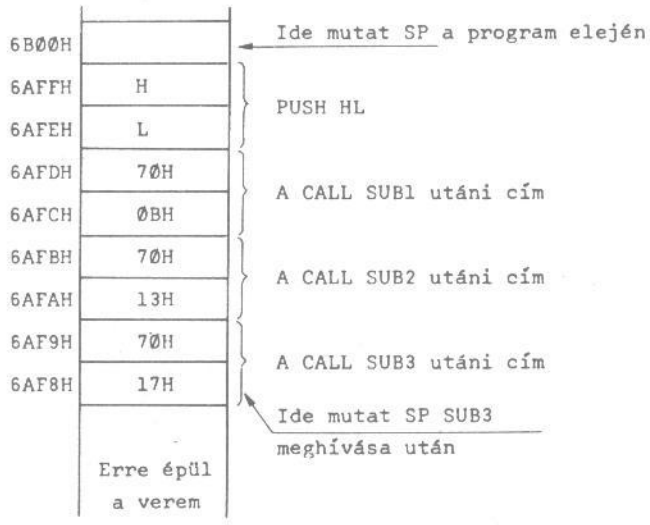
használjuk, mert a többi általános regisztert lefoglalják a blokkmozgató utasítások. Ha a feldolgozó szubrutinban több regiszterre van szükség, akkor a regisztereket kimenthetjük a veremterületre, vagy cserélhetünk regiszterkészletet (PUSH, POP, EXX).

Veremkezelés

Következő példánkban a veremterület kezelését mutatjuk be. Saját veremterületet definiálunk, hogy a tárat kiíratva jól követhessük a verem műveleteket. A BASIC, ill. az EDI által használt veremmutatót a program elején kimentjük a veremterületre, majd saját veremterületünk címét töltjük az SP regiszterbe. A program végén visszaállítjuk az SP eredeti tartalmát. A program további részében egymásba ágyazott szubrutinhívások következnek. A szubrutinok a lehető legrövidebbek, csak CALL és RET utasításokat tartalmaznak, de ez pontosan elegendő a veremkezelés bemutatására:

```
 1                    ;23. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 210000              LD   HL,0     ;HL NULLAZASA
 5 7003 39                  ADD  HL,SP    ;SP KIMENTESE HL-BE
 6 7004 31006B              LD   SP,6B00H;UJ STACK MUTATO
 7 7007 E5                  PUSH HL       ;REGI SP TAROLASA
 8 7008 CD0F70              CALL SUB1     ;SZUBRUTINHIVAS
 9 700B E1                  POP  HL       ;REGI SP TOLTESE
10 700C F9                  LD   SP,HL    ;HL-EN KERESZTUL
11 700D 18FE         VEGE:  JR   VEGE
12 700F CD1370       SUB1:  CALL SUB2     ;MASODIK SZINT
13 7012 C9                  RET
14 7013 CD1770       SUB2:  CALL SUB3     ;HARMADIK SZINT
15 7016 C9                  RET
16 7017 C9           SUB3:  RET           ;NEGYEDIK SZINT
17                          END
```

A veremterület tartalmát ábránk mutatja. (Futtassuk le a programot, és írassuk ki a képernyőre a tárnak ezt a részét!)



8.1. ábra. A veremterület kezelése

A veremterület a tár bármely, más célra nem használt részén elhelyezhetjük. Ha nem írunk nagyon bonyolult programot, akkor 30...40 byte tár általában elegendő erre a célra. Ügyeljünk arra, hogy minden CALL, ill. PUSH utasításhoz tartoznia kell pontosan egy RET, ill. POP utasításnak. BASIC-ben hibaüzenetet kapunk, ha nem párban állnak a GOSUB és RET utasítások. Assemblerben nem kapunk hibaüzenetet, folytatódik a program végrehajtása, de nem ott és nem úgy, ahogy elképzeltük.

9. fejezet

Aritmetikai példák

Ebben a fejezetben az aritmetikai és logikai egység által végzett műveletek néhány alkalmazását mutatjuk be. Először részletesebben foglalkozunk a Z80 aritmetikai lehetőségeivel, majd példák következnek az összeadásra, kivonásra, szorzásra, osztásra és más, aritmetikai jellegű műveletekre.

Egy kis aritmetika

A Z80 mikroprocesszor előjel nélküli egész számokkal, előjeles számokkal és binárisan kódolt decimális (BCD) számokkal tud összeadást, kivonást végezni. Az előző fejezetekben általában előjel nélküli számokkal – tárcímekkel, ciklusváltozókkal – számoltunk. Mint láttuk, egy byte-on 0-tól 255-ig, két byte-on 0-tól 65535-ig lehet előjel nélküli számokat ábrázolni.

Az összeadás és kivonás műveletekkel megfelelő formában ábrázolt előjeles számokat is lehet kezelni. Előjeles számok ábrázolásakor a legértékesebb bit a szám előjelét mutatja. Ha az előjelbit nulla, akkor a szám pozitív, és a maradék 7, ill. 15 biten binárisan ábrázoljuk a szám abszolút értékét. Így az előjel nélküli számok és az előjeles pozitív számok azonos alakúak, de az első bit most nulla, így a legnagyobb ábrázolható előjeles pozitív szám +127, ill. +32767. Ha a szám negatív, akkor az előjelbit 1, de a szám többi részével egészen más a helyzet, mint az előbb, ugyanis ha a maradék 7, ill. 15 biten ugyanúgy ábrázolnánk a szám abszolút értékét, mint a pozitív számoknál, akkor az összeadást és kivonást nem lehetne elvégezni. Ezért a negatív számokat úgy ábrázoljuk, hogy a megfelelő pozitív szám minden bitjét az ellenkezőjére változtatjuk, és az így kapott számhoz 1-et hozzáadunk. Így pl. a 38 komplemensét a következőképpen állíthatjuk elő:

eredeti szám	00100110	(+38)
1. lépés	11011001	
2. lépés	11011010	(–38)

Az előjeles számoknak ezt az ábrázolásmódját kettes komplemens ábrázolásnak nevezzük. Ne keressünk benne sok logikát, csupán arról van szó, hogy a Z80-at úgy tervezték, hogy az ilyen módon ábrázolt előjeles számokkal lehessen az összeadást és kivonást hibátlanul elvégezni. Kettes komplemens ábrázolásban egy byte-on –128-tól +127-ig, két byte-on –32768-tól +32767-ig lehet előjeles számokat ábrázolni. A –128 kettes komplemens alakja 10000000, a – 1 kettes komplemens alakja pedig 11111111.

A kettes komplemens ábrázolású számokkal hibátlanul el lehet végezni az összeadást, kivonást – mintha valamilyen 256 modulusú, ill. 65536 modulusú aritmetikában volnánk. Pl.

00101000	(+40)
+ 11011010	(–38)
00000010	(+ 2)

helyes eredményt ad.

Az aritmetikai műveletek eredményét az F regiszter jelzőbitjei mutatják. Ha a művelet eredménye nulla, pl. az

```
LD  A,23
SUB 23
```

utasítások után, akkor a Z jelzőbit értéke 1. Ha a legértékesebb bitről átvitel van (ugyanolyan átvitel, mint a tízes számrendszerben a közönséges összeadásnál), pl.

```
LD  A,255
ADD A,1
```

utasítások után, akkor a C jelzőbit értéke 1. Ha a kettes komplemens ábrázolású előjeles számokkal végzett művelet eredménye nem ábrázolható (egybyte-os számoknál az eredmény –128-nál kisebb, vagy +127-nél nagyobb), és ezért hibás eredmény adódik, akkor a P/V jelzőbit értéke 1. Ez történik, pl. az

10001101	–115	vagy	01000001	65
+ 11110001	– 15		+ 01000000	64
01111110	+126		10000001	–127

műveletek elvégzésekor.

A jelzőbitek állapotát egyszerűen követhetjük pl. a következő programmal:

```
 1                  ;24. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 210000              LD   HL,0     ;SP KIMENTESE
 5 7003 39                  ADD  HL,SP
 6 7004 31007A              LD   SP,7A00H;UJ STACK MUTATO
 7 7007 3A1970              LD   A,(BR) ;MASODIK OPERANDUS
 8 700A 47                  LD   B,A
 9 700B 3A1870              LD   A,(AR) ;ELSO OPERANDUS
10 700E 80                  ADD  A,B    ;OSSZEG
11 700F F5                  PUSH AF     ;OSSZEG ES JELZOBIT
12                  ;                    KIMENTES
13 7010 3A1870              LD   A,(AR)
14 7013 90                  SUB  B      ;KULONBSEG
15 7014 F5                  PUSH AF     ;A ES F KIMENTESE
16 7015 F9                  LD   SP,HL  ;REGI STACK MUTATO
17 7016 18FE        VEGE:   JR   VEGE
18 7018 21          AR:     DB   33     ;ELSO OPERANDUS
19 7019 40          BR:     DB   64     ;MASODIK OPERANDUS
20                          END
```

A programot lefuttatva az A és B címkénél lerakott két szám összegét, különbségét és a jelzőbiteket a saját veremterületről, a 7A00H alatti címekről kiolvashatjuk. Ugyanezt a programot más-más adatokkal lefuttatva tanulmányohatjuk, hogyan mutatják a jelzőbitek a műveletek eredményét.

Összeadás és kivonás

A következő példában lassan felfelé mozgatunk egy képelemet. A programban a HL regiszterpárba a képernyő utolsó sora közepének címét töltjük, amiből a DE regiszterpárba töltött 64-et kivonva a képernyőn mindig egy sorral feljebb jutunk. A sorokat az A regiszterben számláljuk.

```
 1                    ;25. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 21DF3F              LD   HL,3FDFH;KEZDOCIM
 5 7003 114000              LD   DE,64   ;CSOKKENTES
 6 7006 3E0F                LD   A,15    ;SOROK SZAMA
 7 7008 010000              LD   BC,0    ;KESLELTETES
 8 700B 3681          C1:   LD   (HL),81H;RAJZOLAS
 9 700D 05            C2:   DEC  B       ;1. KESLELTETES
10 700E 20FD                JR   NZ,C2
11 7010 0D                  DEC  C       ;2. KESLELTETES
12 7011 20FA                JR   NZ,C2
13 7013 3680                LD   (HL),80H;TORLES
14 7015 B7                  OR   A       ;C JELZOBIT TORLESE
15 7016 ED52                SBC  HL,DE   ;ELOZO SOR
16 7018 3D                  DEC  A       ;SORSZAM-1
17 7019 20F0                JR   NZ,C1
18 701B 18FE         VEGE:  JR   VEGE
19                          END
```

A program érdekessége, hogy a késleltetést kettős ciklusban végzett számlálással oldja meg. A BC regiszterpárba 0-t töltünk, a B regisztert 0-tól csökkentjük egészen addig, amíg újra 0-t kapunk, majd a külső ciklusban a C regisztert csökkentjük ugyanígy, tehát 256*256-ig, vagyis 65536-ig számlálunk. Ez körülbelül fél másodperc várakozást jelent. A 16 bites kivonás előtti OR A utasítás célja a C jelzőbit törlése. Erre azért van szükség, mert az SBC utasítás a C jelzőbitet is kivonja, ezért ahhoz, hogy valóban mindig 64-et vonjunk ki, törölni kell a C jelzőbitet, hiszen azt sok utasítás állítja, ezért nem lehet tudni, hogy éppen 0-t vagy 1-et tartalmaz-e.

A C jelzőbit törlése megoldható a 3. fejezetben nem ismertetett SCF és CCF utasításokkal is. Az

SCF 3F

utasítás (Set Carry Flag) a C jelzőbitet 1-re állítja, a H és N jelzőbiteket törli, a többit változatlanul hagyja. A

CCF 37

utasítás (Complement Carry Flag) a C jelzőbitet az ellenkezőjére változtatja (1-ről 0-ra, 0-ról 1-re). Az N jelzőbitet törli, a többi jelzőbitet változatlanul hagyja.

A fenti programban azért alkalmaztuk mégis az OR A utasítást, mert az gyorsabb és rövidebb, mint az SCF és CCF utasítások egymásutánja.

A Z80 mikroprocesszor aritmetikai és logikai egységének utasításai csak 8-bites és 16 bites számok összeadását, kivonását teszik lehetővé. Ha nagyobb számokat kell összeadnunk vagy kivonnunk, akkor 8-bites vagy 16 bites egységekben kell elvégeznünk a műveletet, és közben – ugyanúgy, mint a tízes számrendszerben a közönséges összeadásnál, kivonásnál – mindig gondoskodnunk kell az átvitel hozzáadásáról, ill. a beváltás kivonásáról. Az eljárást négybyte-os számokon, először papíron mutatjuk be. Pl. a +344050 és +500000 előjeles számokat így adhatjuk össze:

```
                 111   1111111 11          (átvitel)
00000000 00000101 00111111 11110010  (+344050)
00000000 00000111 10100001 00100000  (+500000)
-------------------------------------------
00000000 00001100 11100001 00010010  (+844050)
```

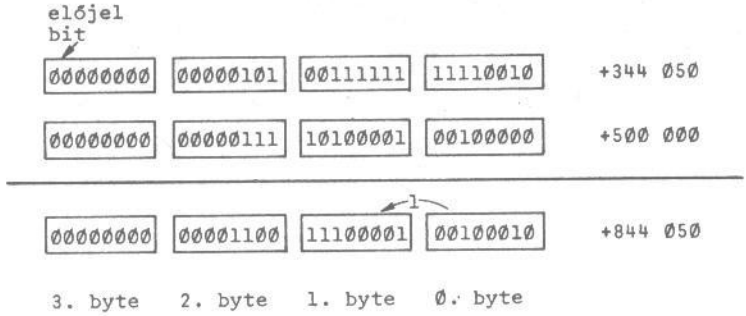
A legértékesebb bit a szám előjele, a többi biten az átvitel figyelembevételével egyszerű bináris összeadást végeztünk.

A következő program ugyanezeket a számokat byte-onként adja össze. A két számot a FORRAS és CEL címkéknél byte-onként tároljuk, az eredményt a CEL szám helyére írjuk:

```
 1                    ;26. LISTA
 2                            ORG  7000H
 3                            LOAD 7000H
 4 7000 DD212E70              LD   IX,CEL ;CEL CIM
 5 7004 FD213270              LD   IY,FORRAS;FORRAS CIM
 6 7008 DD7E03                LD   A,(IX+3);0. BYTE
 7 700B FD8603                ADD  A,(IY+3);OSSZEADAS
 8 700E DD7703                LD   (IX+3),A;TAROLAS
 9 7011 DD7E02                LD   A,(IX+2);1. BYTE
10 7014 FD8E02                ADC  A,(IY+2);OSSZEADAS
11 7017 DD7702                LD   (IX+2),A;TAROLAS
12 701A DD7E01                LD   A,(IX+1);2. BYTE
13 701D FD8E01                ADC  A,(IY+1);OSSZEADAS
14 7020 DD7701                LD   (IX+1),A;TAROLAS
15 7023 DD7E00                LD   A,(IX+0);3. BYTE
16 7026 FD8E00                ADC  A,(IY+0);OSSZEADAS
17 7029 DD7700                LD   (IX+0),A;TAROLAS
18 702C 18FE        VEGE:     JR   VEGE
19 702E 00053FF2    CEL:      DB   0,5,3FH,0F2H; +344050
20 7032 0007A120    FORRAS:   DB   0,7,0A1H,20H; +500000
21                            END
```

A programban az IX indexregiszter a céloperandusra, az IY indexregiszter a forrásoperandusra mutat. Az összeadandók legértékesebb byte-ja az IX+0, ill. IY+0, legkevésbé értékes byte-ja az IX+3, ill. IY+3 címen található. A négy összeadást megírhattuk volna egy ciklusban is, de így világosabban láthatók az összeadás lépései. Az első összeadás ADD utasítással a legkevésbé értékes byte-okat adja össze. Az összeadás után a C jelzőbit mutatja, hogy van-e átvitel a 7. bitről, amely most nem előjel, hanem értékes számjegy. A következő byte-okon ADC utasítással végezzük az összeadást, amely az összeadandókhoz az előző összeadásból származó átvitelt is hozzáadja (az LD utasítások nem állítják a jelzőbiteket).

A négybyte-os kivonást ugyanúgy lehet elvégezni, mint az összeadást, a legkevésbé értékes byte-on SUB, a további byte-okon SBC utasításokkal. (Megjegyezzük, hogy ez a négybyte-os összeadás, ill. kivonás nem azonos



9.1. ábra. Többszörös pontosságú összeadás

a LEVEL II. BASIC egyszeres pontosságú aritmetikájával, az ugyanis normálalakkal dolgozik, a számábrázolása tehát alapvetően eltérő.)

A következő két program általános eljárást ad tetszőleges pontosságú előjeles egész számok összeadására, kivonására. A szubrutinok CALL utasítással történő meghívása előtt az IX indexregiszterbe a céloperandus legértékesebb byte-jának címét, az IY indexregiszterbe a forrásoperandus legértékesebb byte-jainak számát kell betölteni. A többszörös pontosságú összeadást a következő szubrutin meghívásával lehet elvégezni:

```
 1                    ;27. LISTA
 2                            ORG  7E00H
 3                            LOAD 7E00H
 4 7E00 D5            PLUSZ:  PUSH DE      ;DE MEGORZESE
 5 7E01 1600                  LD   D,0
 6 7E03 58                    LD   E,B     ;BYTE-SZAM DE-BEN
 7 7E04 1B                    DEC  DE      ;BYTE-SZAM-1
 8 7E05 DD19                  ADD  IX,DE   ;ALSO BYTE CIME
 9 7E07 FD19                  ADD  IY,DE   ;ALSO BYTE CIME
10 7E09 D1                    POP  DE      ;DE VISSZATOLTESE
11 7E0A B7                    OR   A       ;C JELZOBIT TORLESE
12 7E0B DD7E00       L1:      LD   A,(IX+0);CEL BETOLTESE
13 7E0E FD8E00                ADC  A,(IY+0);OSSZEADAS
14 7E11 DD7700                LD   (IX+0),A;EREDMENY TAROLASA
15 7E14 1001                  DJNZ L2      ;UGRIK, HA NEM KESZ
16 7E16 C9                    RET          ;VISSZATERES
17 7E17 DD2B         L2:      DEC  IX      ;KOVETKEZO BYTE
18 7E19 FD2B                  DEC  IY      ;KOVETKEZO BYTE
19 7E1B C30B7E                JP   L1      ;FOLYTATAS
20                            END
```

A szubrutin az A és B regiszterek tartalmát megváltoztatja, az összes többi regisztert változatlanul hagyja. Többszörös pontosságú kivonást a következő szubrutinnal végezhetünk:

```
 1                    ;28. LISTA
 2                            ORG  7F00H
 3                            LOAD 7F00H
 4 7F00 D5            MINUSZ: PUSH DE      ;DE MEGORZESE
 5 7F01 1600                  LD   D,0
 6 7F03 58                    LD   E,B     ;BYTE-SZAM DE-BEN
 7 7F04 1B                    DEC  DE      ;BYTE-SZAM-1
 8 7F05 DD19                  ADD  IX,DE   ;ALSO BYTE CIME
 9 7F07 FD19                  ADD  IY,DE   ;ALSO BYTE CIME
10 7F09 D1                    POP  DE      ;DE VISSZATOLTESE
11 7F0A B7                    OR   A       ;C JELZOBIT TORLESE
```

```
12 7F0B DD7E00   L1:   LD   A,(IX+0);CEL BETOLTESE
13 7F0E FD9E00         SBC  A,(IY+0);KIVONAS
14 7F11 DD7700         LD   (IX+0),A;EREDMENY TAROLASA
15 7F14 1001           DJNZ L2      ;UGRIK, HA NEM KESZ
16 7F16 C9             RET          ;VISSZATERES
17 7F17 DD2B     L2:   DEC  IX      ;KOVETKEZO BYTE
18 7F19 FD2B           DEC  IY      ;KOVETKEZO BYTE
19 7F1B C30B7F         JP   L1      ;FOLYTATAS
20                     END
```

Decimális aritmetika

A decimális számokat binárisan kódolt decimális (BCD) számokká úgy alakíthatjuk át, hogy minden decimális számjegyet külön-külön bináris számmá alakítunk, és azokat 4 biten ábrázoljuk. Így pl. a négyjegyű decimális számok binárisan kódolt decimális ábrázolásához 16 bitre van szükség:

1234 = 0001 0010 0011 0100 (BCD)
3286 = 0011 0010 1000 0110 (BCD)
9991 = 1001 1001 1001 0001 (BCD)

A BCD számok kiíratása egyszerűbb feladat, mint a bináris számok megjelenítése, így az eredményeket számjegyekkel kiíró különböző berendezések programozásakor jól használható ez az ábrázolásmód. A BCD számokat binárissá, a bináris számokat BCD-re átalakítani ugyan nem nehéz feladat, mégis sokszor hasznos, hogy átalakítás nélkül, BCD formában is el lehet végezni az összeadást és a kivonást.

A BCD összeadás, ill. kivonás más, mint a bináris összeadás, ill. kivonás, ugyanis az 1010, 1011, 1100, 1101, 1110 és 1111 számjegycsoportok nem megengedett értékek, hiszen ezek nem decimális számok megfelelői. Ezért ha a művelet során valamelyik 4-bites egységben ilyen eredmény adódik, akkor azt ki kell igazítani úgy, hogy a 4-bites egységeken decimális számjegyek legyenek. Pl. ha az 1234 és 6777 binárisan kódolt decimális számokat adjuk össze, akkor a tízesátlépés miatt az első két 4-bites egységen nem decimális számjegyet kapunk. Az eredményt úgy javíthatjuk ki, hogy a hibás 4-bites egységek mindegyikéhez 6-ot hozzáadunk (9.2. ábra). Ha kivonást végzünk, akkor a hibás 4-bites egységekből 6-ot ki kell vonni.

```
        0001  0010  0011  0100      1234 (BCD)

        0110  0111  0111  0111      6777 (BCD)

ADD     0111  1001  1010  1011      79AB(közbülső eredmény)

DAA     0000  0110  0110  0110      decimális igazitás
        ------------------------
        1000  0000  0001  0001      8011 (végeredmény)
```

9.2. ábra. Decimális igazítás

Bár a leírt eljárást megvalósító program megírása nem nagy feladat, mégis felesleges, mert a Z80 mikroprocesszor az előzőkben még nem ismertetett

DAA 27

decimális igazítás utasítása (Decimal Adjust Accumulator) pontosan ezt a műveletet végzi az A regiszter tartalmával.

A DAA utasítás először megvizsgálja az A regiszter alsó felét. Ha ott 0 és 9 közötti számjegyet talál, és a H jelzőbit nulla, akkor az A regiszternek ezt a felét változatlanul hagyja. Ha az ott talált érték 9-nél nagyobb, vagy ha a H jelzőbit értéke 1, akkor az A regiszter tartalmát összeadás utasítás után 6-tal növeli, kivonás utasítás után 6-tal csökkenti (azt, hogy az előző művelet összeadás vagy kivonás volt-e, az N jelzőbit mutatja). Miután az A regiszter alsó fele már BCD számjegyet tartalmaz, a DAA utasítás megvizsgálja az A regiszter felső felét. Ha ott 0 és 9 közötti számjegyet talál, és a C jelzőbit nulla, akkor a művelet befejeződik. Ha valamelyik feltétel nem teljesül, akkor az A regiszter felső felét összeadás után 6-tal növeli, kivonás után 6-tal csökkenti. A DAA utasítás állítja a Z, S és P/V jelzőbiteket (P/V a paritást mutatja), változatlanul hagyja az N jelzőbitet.

A következő program 0-tól 99-ig kiszámolja és tárolja a bináris számokat és BCD megfelelőiket. A bináris értékeket 7A00H-tól, a BCD értékeket 7B00H-tól helyezi el a tárban. (A program lefuttatása után írassuk ki a képernyőre a tárnak ezeket a részeit!) Figyeljük meg, hogy az A regiszter tartalmát nem az INC A utasítással, hanem az ADD A,1 utasítással növeljük, ugyanis az INC utasítások a C jelzőbitet – amelyet a DAA utasítás használ – nem állítják.

```
 1                    ;29. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 110000              LD   DE,0     ;D=BCD, E=BINARIS
 5 7003 DD21007A            LD   IX,7A00H;BCD BUFFER
 6 7007 FD21007B            LD   IY,7B00H;BINARIS BUFFER
 7 700B 0664                LD   B,100    ;SZAMLALO
 8 700D DD7200      CIKL:   LD   (IX+0),D;BCD TAROLASA
 9 7010 FD7300              LD   (IY+0),E;BINARIS TAROLASA
10 7013 DD23                INC  IX       ;MUTATOK NOVELESE
11 7015 FD23                INC  IY
12 7017 7A                  LD   A,D      ;BCD ERTEK
13 7018 C601                ADD  A,1      ;NOVELES
14 701A 27                  DAA           ;DECIMALIS IGAZITAS
15 701B 57                  LD   D,A      ;BCD MEGORZESE
16 701C 7B                  LD   A,E      ;BINARIS ERTEK
17 701D 3C                  INC  A        ;NOVELES
18 701E 5F                  LD   E,A      ;BINARIS MEGORZESE
19 701F 10EC                DJNZ CIKL     ;TOVABB
20 7021 18FE       VEGE:    JR   VEGE
21                          END
```

Összehasonlítás

Az előjel nélküli számok összehasonlítása a Z és C jelzőbitek vizsgálatával egyszerűen megoldható. Pl. ha az A regiszter tartalma megegyezik a B regiszter tartalmával, akkor a CP B utasítás a két regiszter tartalmának különbségét

nullának találja, és így a Z jelzőbit értéke 1 lesz. Ha A tartalma kisebb, mint B tartalma, akkor a különbség negatív, és ekkor a C jelzőbit értéke 1 lesz. Így a

```
CP B
JP Z,EGYENLO
JP C,KISEBB
JP NAGYOBB
```

utasítások hatására sorra akkor adódik az EGYENLO, KISEBB, NAGYOBB címkékre a vezérlés, ha az A regiszter tartalma megegyezik a B regiszter tartalmával, kisebb, ill. nagyobb annál.

Az előjeles számok összehasonlítása az előjelbit sajátos szerepe miatt ennél sokkal bonyolultabb. A következő szubrutin összehasonlítja az A regiszterben és a B regiszterben levő előjeles számokat, és a vezérlést a szubrutinhívást követő három vezérlésátadó utasítás közül az elsőre adja, ha A tartalma kisebb, mint B tartalma, a másodikra, ha a két regiszter tartalma megegyezik, és a harmadikra, ha A tartalma nagyobb, mint B tartalma. A szubrutin hívása tehát az A és B regiszterek betöltése után a

```
CALL CMP
JP   KISEBB
JP   EGYENLO
JP   NAGYOBB
```

utasításokkal történik. A szubrutinból visszatérés most nem a RET utasítással, hanem a veremterületről a HL regiszterpárba kimentett és az összehasonlítás során módosított visszatérési cím felhasználásával történik.

```
 1                ;30. LISTA
 2                        ORG  7E00H
 3                        LOAD 7E00H
 4 7E00 E1        CMP:    POP  HL       ;VISSZATERESI CIM
 5 7E01 D5                PUSH DE       ;DE MEGORZESE
 6 7E02 110300            LD   DE,3     ;CIMZESI ELTOLAS
 7 7E05 B8                CP   B        ;OSSZEHASONLITAS
 8 7E06 280A              JR   Z,AZONOS;EGYENLOK
 9 7E08 F5                PUSH AF       ;JELZOBIT MEGORZES
10 7E09 A8                XOR  B        ;ELOJEL VIZSGALAT
11 7E0A 17                RLA           ;C JELZOBIT=ELOJEL
12 7E0B DA157E            JP   C,ELOJEL;KULONBOZO ELOJEL
13 7E0E F1                POP  AF       ;JELZOBIT BETOLTES
14 7E0F 3802              JR   C,KICSI;UGRIK, HA A<B
15 7E11 19       NAGY:    ADD  HL,DE    ;VISSZA A 3. JP-RE
16 7E12 19       AZONOS:  ADD  HL,DE    ;VISSZA A 2. JP-RE
17 7E13 D1       KICSI:   POP  DE       ;DE VISSZATOLTES
18 7E14 E9                JP   (HL)     ;VISSZATERES
19 7E15 F1       ELOJEL:  POP  AF       ;JELZOBIT BETOLTES
20 7E16 DA117E            JP   C,NAGY ;UGRIK, HA A>B
21 7E19 C3137E            JP   KICSI    ;A<B
22                        END
```

Szorzás és osztás

A Z80 mikroprocesszornak nincsenek szorzás és osztás utasításai, viszont a balra eltolás 2-vel szorzást, a jobbra eltolás 2-vel osztást eredményez, és az eltolások ismétlésével 2 hatványaival is szorozhatunk, oszthatunk, így a szor-

zást és osztást visszavezethetjük eltolásokra és összeadásokra, ill. eltolásokra és kivonásokra.

Első példánkban az A regiszterbe töltött előjel nélküli számot tízzel fogjuk megszorozni. A szám tízszerese a szám kétszeresének és nyolcszorosának összege, ezért először egy balra eltolással előállítjuk a szám kétszeresét, ezt megőrizzük, további két eltolással megkapjuk a szám nyolcszorosát, majd összeadjuk a részletszorzatokat. A programban az ADAT címkénél elhelyezett öt szám tízszeresét fogjuk kiszámolni:

```
 1                    ;31. LISTA
 2                            ORG  7000H
 3                            LOAD 7000H
 4 7000 211470                LD   HL,ADAT;ADAT TABLAZAT
 5 7003 0605                  LD   B,5     ;5 ELEM
 6 7005 7E            CIKL:   LD   A,(HL) ;BETOLTES
 7 7006 CB27                  SLA  A       ;SZAM*2
 8 7008 4F                    LD   C,A     ;SZAM*2 MEGORZESE
 9 7009 CB27                  SLA  A       ;SZAM*4
10 700B CB27                  SLA  A       ;SZAM*8
11 700D 81                    ADD  A,C     ;SZAM*10
12 700E 77                    LD   (HL),A ;TAROLAS
13 700F 23                    INC  HL      ;KOVETKEZO ERTEK
14 7010 10F3                  DJNZ CIKL    ;FOLYTATAS
15 7012 18FE         VEGE:    JR   VEGE
16 7014 01030A0F     ADAT:    DB   1,3,10,15,30
16 7018 1E
17                            END
```

Futtassuk le a programot, és nézzük meg a szorzások eredményeit! Látjuk, hogy az első négy esetben helyes eredményt kaptunk, az ötödik esetben azonban a szám tízszerese nem fér el a 8-bites regiszterben, így hibás eredmény adódik. (Az SLA és ADD utasítások után a C jelzőbit mutatja a túlcsordulást. Ennek felhasználásával kiegészíthetnénk a programot úgy, hogy jelezze a hibás eredményt.)

Ha nagyon kis számmal szorzunk, akkor az egyszerű ismételt összeadás is elég gyors eljárás. A következő program az ADAT1 címkénél lerakott 16 bites előjel nélküli szorzandót annyiszor adja hozzá saját magához, amennyi az ADAT2 címkénél lerakott szorzó értéke.

```
 1                    ;32. LISTA
 2                            ORG  7000H
 3                            LOAD 7000H
 4 7000 ED5B1070              LD   DE,(ADAT1);SZORZANDO
 5 7004 3A1270                LD   A,(ADAT2);SZORZO
 6 7007 47                    LD   B,A     ;SZORZO B REG-BEN
 7 7008 210000                LD   HL,0    ;KEZDOERTEK
 8 700B 19            CIKL:   ADD  HL,DE   ;ISMETELT OSSZEADAS
 9 700C 10FD                  DJNZ CIKL
10 700E 18FE          VEGE:   JR   VEGE
11 7010 E803          ADAT1:  DW   1000
12 7012 07            ADAT2:  DB   7
13                            END
```

Az ismételt eltolásokra és a részletszorzatok összeadására épülő általános szorzó eljárás lényegében ugyanaz, mint az általános iskolából ismert tízes számrendszerbeli írásbeli szorzás. Kettes számrendszerben azonban sokkal egyszerűbb a módszer, hiszen csak a 0 és az 1 számjegyek vannak, tehát a részletszorzatok kiszámításához nem is kell szorzást végezni. Ha a szorzan-

dóban egyes következik, akkor leírjuk a részletszorzatot, ha nulla, akkor nem írjuk le. Pl. a 8 biten előjel nélkül ábrázolt 23-at így szorozhatjuk meg 17-tel:

```
                0 0 0 1 0 1 1 1 * 0 0 0 1 0 0 0 1 = 23*17
-----------------------------------
                0 0 0 1 0 1 1 1
              0 0 0 0 0 0 0 0
            0 0 0 0 0 0 0 0
          0 0 0 0 0 0 0 0
        0 0 0 1 0 1 1 1
      0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0
-----------------------------------
0 0 0 0 0 0 1 1 0 0 0 0 1 1 1                    = 391
```

ahol az eljárás bemutatása céljából a csupa nulla részletszorzatokat is leírtuk.

A következő szubrutin a DE regiszterpárba töltött előjel nélküli egész szorzandót megszorozza a B regiszterbe töltött szorzóval. A szubrutinból visszatéréskor a szorzás eredménye a HL regiszterpárban található, a DE regiszterpár tartalma megváltozott, a B regiszterbe nulla került.

```
 1                    ;33. LISTA
 2                            ORG  7E00H
 3                            LOAD 7E00H
 4 7E00 210000        SZOROZ: LD   HL,0      ;KEZDOERTEK
 5 7E03 CB38          CIKL:   SRL  B         ;SZORZOBIT KITOLASA
 6 7E05 3001                  JR   NC,FOLYT;UGRIK, HA NULLA
 7 7E07 19                    ADD  HL,DE     ;RESZLETSZORZAT
 8                    ;                       OSSZEADASA
 9 7E08 C8            FOLYT:  RET  Z         ;VISSZATERES
10 7E09 EB                    EX   DE,HL     ;SZORZANDO HL-BEN
11 7E0A 29                    ADD  HL,HL     ;SZORZANDO ELTOLASA
12 7E0B EB                    EX   DE,HL     ;SZORZANDO DE-BEN
13 7E0C C3037E                JP   CIKL      ;FOLYTATAS
14                            END
```

A szubrutin akkor ad helyes eredményt, ha a szorzandó, a szorzó és a szorzat elfér a DE regiszterpárban, B regiszterben, ill. HL regiszterpárban, tehát ha sorra legfeljebb 16 bites, 8-bites, ill. 16 bites. Megjegyezzük, hogy az ADD HL,DE utasítás a Z jelzőbitet nem állítja, így a RET utasítás akkor tér vissza a főprogramba, ha a B regiszter eltolásának eredménye nulla volt.

Az osztás még körülményesebb, mint a szorzás. Lássuk először az ismételt kivonással történő osztást, amely ugyanúgy, mint a hasonló szorzási módszer, kis hányadosnál jól alkalmazható, nagyobb hányadosnál elviselhetetlenül lassú. A HL regiszterpárba töltött 16 bites előjel nélküli osztandóból ismételten kivonjuk a DE regiszterpárba töltött 16 bites előjel nélküli osztót, a kivonásokat a B regiszterben számláljuk:

```
 1                    ;34. LISTA
 2                            ORG  7000H
 3                            LOAD 7000H
 4 7000 2A1970                LD   HL,(ADAT2);OSZTO
 5 7003 E5                    PUSH HL
 6 7004 D1                    POP  DE        ;OSZTO DE-BEN
```

```
 7 7005 2A1770          LD   HL,(ADAT1);OSZTANDO HL-BEN
 8 7008 0600            LD   B,0      ;HANYADOS TORLESE
 9 700A B7       CIKL:  OR   A        ;C JELZOBIT TORLESE
10 700B ED52            SBC  HL,DE    ;OSZTANDO-OSZTO
11 700D FA1470          JP   M,KESZ   ;UGRIK, HA KESZ
12 7010 04              INC  B        ;HANYADOS SZAMLALAS
13 7011 C30A70          JP   CIKL     ;TOVABB
14 7014 19       KESZ:  ADD  HL,DE    ;MARADEK
15 7015 18FE     VEGE:  JR   VEGE
16 7017 204E     ADAT1: DW   20000
17 7019 F401     ADAT2: DW   500
18                      END
```

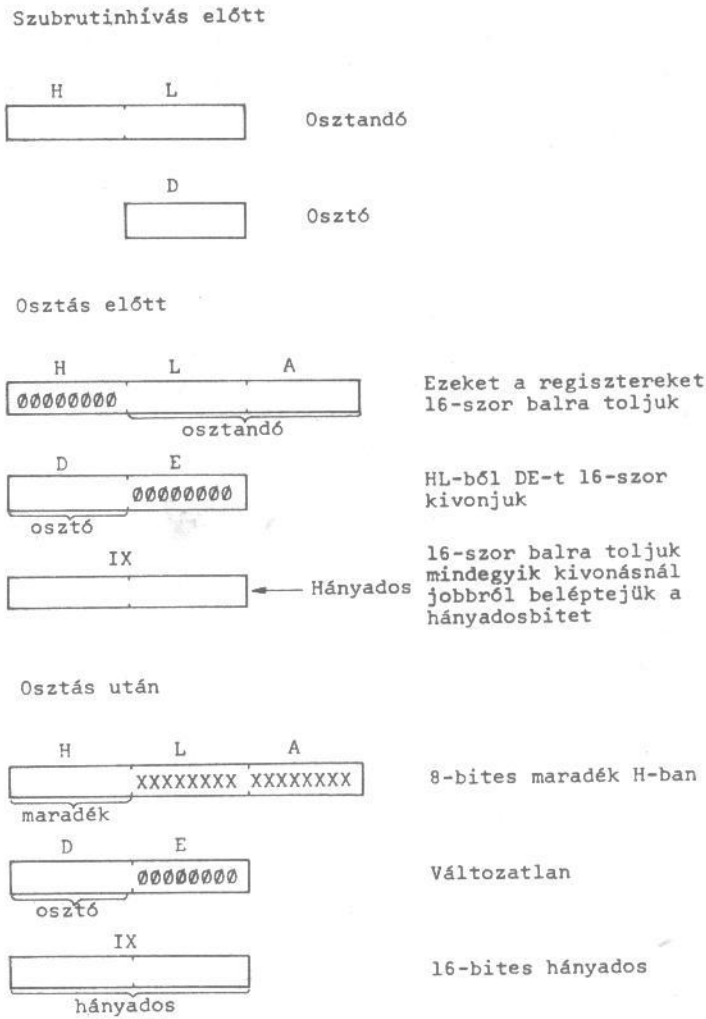
A kivonást addig ismételjük, amíg az eredmény negatív lesz. A maradékot ekkor az osztó hozzáadásával, a HL regiszterpárban kapjuk.

A következő szubrutin a HL regiszterpárban levő 16 bites, előjel nélküli osztandót elosztja a D regiszterben levő 8-bites, előjel nélküli osztóval. A szubrutinból visszatéréskor az IX indexregiszterben lesz a hányados, a H regiszterben a maradék. A szubrutin az L, E és A regiszterek tartalmát megváltoztatja, a D regisztert változatlanul hagyja.

```
 1                ;35. LISTA
 2                       ORG  7E00H
 3                       LOAD 7E00H
 4 7E00 7D        OSZT:  LD   A,L      ;OSZTANDO ALSO BYTE
 5 7E01 6C               LD   L,H      ;FELSO BYTE
 6 7E02 2600             LD   H,0      ;KIVONAS KEZDOERTEK
 7 7E04 1E00             LD   E,0
 8 7E06 0610             LD   B,16     ;16 ISMETLES
 9 7E08 DD210000         LD   IX,0     ;HANYADOS NULLAZAS
10 7E0C 29        CIKL:  ADD  HL,HL    ;OSZTANDO BALRA
11 7E0D 17               RLA           ;ALSO BYTE ELTOLAS
12 7E0E D2127E           JP   NC,CIKL1;UGRIK, HA BIT=0
13 7E11 23               INC  HL       ;HL ELTOLAS
14 7E12 DD29      CIKL1: ADD  IX,IX    ;HANYADOS BALRA
15 7E14 DD23             INC  IX       ;HANYADOSBAN BIT=1
16 7E16 B7               OR   A        ;C JELZOBIT TORLESE
17 7E17 ED52             SBC  HL,DE    ;KIVONASI KISERLET
18 7E19 D21F7E           JP   NC,FOLYT;SIKERES KIVONAS
19 7E1C 19               ADD  HL,DE    ;VISSZAALLITAS
20 7E1D DD2B             DEC  IX       ;HANYADOSBAN BIT=0
21 7E1F 10EB      FOLYT: DJNZ CIKL     ;FOLYTATAS
22 7E21 C9               RET
23                       END
```

A programban a HL regiszterpárban elhelyezett osztandó maradványából ismételten kivonjuk a D regiszterben levő osztót. Az írásbeli osztáshoz hasonló eljárással a maradványt minden ismétlés után eltoljuk egy bittel. Ha a kivonás eredménye nem negatív, akkor 1-et, ha negatív, akkor 0-t írunk a hányadosba. A hányadost minden ismétlés után egy jeggyel balra toljuk, hiszen sorra az egyre kevésbé értékes jegyek következnek. Végül 16 ismétlés után az IX indexregiszterben ott lesz a 16 bites hányados, a H regiszterben a 8-bites maradék, a D regiszterben megmarad az eredeti osztó és az E regiszterbe nulla kerül. A HL regiszterpárt és az IX indexregisztert önmagukhoz hozzáadással toljuk balra. A 9.3. ábrán látható, hogy mit tartalmaznak az egyes regiszterek az osztás előtt, az osztás közben és a szubrutinból való visszatéréskor.

Az eljárást könnyebben megértjük, ha úgy futtatjuk le a programot, hogy minden egyes ismétlés után megállunk és megvizsgáljuk a regiszterek tartalmát.



9.3. ábra. Regiszterhasználat az osztó szubrutinban

Az előjeles számok szorzása és osztása kicsit bonyolultabb, mint az előzők. Az előjel nélküli szorzás és osztás szubrutinokat átírhatjuk úgy, hogy az operandusok abszolút értékét vesszük, és a művelet elvégzése után helyreállítjuk az előjelet, de vigyázzunk, mert túlcsorduláskor – pl. ha –128-at megszorozzuk –128-cal – meglepetések érhetnek.

10. fejezet

Egyéb példák

Ebben a fejezetben adatok konvertálásával, szövegbeli karakterek keresésével, táblázatok kezelésével és adatok rendezésével kapcsolatos feladatokkal foglalkozunk.

Számok és ASCII kódok

A 0...9 számjegyeknek a 30H...39H ASCII kódok felelnek meg. A számok bináris értékéből 30H hozzáadásával vagy 30H-val és OR-ral kapjuk a megfelelő ASCII kódot:

```
LD A,bináris
OR 30H
```

Megfordítva, a számok ASCII kódjából a bináris értékeket a felső bitek kinullázásával kapjuk:

```
LD  A,ascii
AND 0FH
```

A következő program a képernyő első tíz sorának elejére sorra kiírja a 0...9 számjegyeket:

```
 1                  ;36. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 21003C              LD   HL,3C00H;ELSO SOR
 5 7003 0600                LD   B,0     ;KEZDOERTEK
 6 7005 0E39                LD   C,39H   ;UTOLSO ASCII KOD
 7 7007 114000              LD   DE,64   ;64 KARAKTERES SOR
 8 700A 78          CIKL:   LD   A,B     ;BINARIS ERTEK
 9 700B F630                OR   30H     ;ASCII KARAKTER
10 700D 77                  LD   (HL),A  ;KIIRATAS
11 700E 19                  ADD  HL,DE   ;KOVETKEZO SOR
12 700F 04                  INC  B       ;KOVETKEZO SZAM
13 7010 B9                  CP   C       ;UTOLSO KARAKTER?
14 7011 C20A70              JP   NZ, CIKL
15 7014 18FE        VEGE:   JR   VEGE
16                          END
```

A hexadecimális kiíratás valamivel nehezebb feladat. A következő szubrutin az A regiszter tartalmát 4-bitenként hexadecimális számjegyként kezeli és két ASCII karakterré alakítja. A 0...9 számjegyek ASCII kódja 30H...39H, az A...F betűk ASCII kódja 41H...46H, ezért az átalakításkor két esetet kell megkülönböztetni. Ha az átalakítandó érték numerikus számjegy, akkor 30H-t, ha betűszámjegy, akkor 37H-t kell hozzáadni. A szubrutinba belépéskor a 8-bites bináris számot az A regiszterbe kell helyezni, a visszatéréskor

a HL regiszterpár tartalmazza a két ASCII karaktert. A szubrutin az A és C regiszterek tartalmát megváltoztatja.

```
 1                    ;37. LISTA
 2                            ORG  7E00H
 3                            LOAD 7E00H
 4 7E00 4F            ASCHEX: LD   C,A     ;ALSO FELBYTE C-BEN
 5 7E01 CB3F                  SRL  A       ;FELSO FELBYTE
 6 7E03 CB3F                  SRL  A
 7 7E05 CB3F                  SRL  A
 8 7E07 CB3F                  SRL  A
 9 7E09 CD157E                CALL CONV    ;ATALAKITAS
10 7E0C 67                    LD   H,A     ;ASCII KOD MEGORZES
11 7E0D 79                    LD   A,C     ;EREDETI ERTEK
12 7E0E E60F                  AND  0FH     ;ALSO FELBYTE
13 7E10 CD157E                CALL CONV    ;ATALAKITAS
14 7E13 6F                    LD   L,A     ;ASCII KOD MEGORZES
15 7E14 C9                    RET
16 7E15 C630         CONV:    ADD  A,30H   ;SZAMJEGYEK KODJA
17 7E17 FE3A                  CP   3AH     ;SZAMJEGY VAGY BETU
18 7E19 FA1E7E                JP   M,CONV1 ;SZAMJEGY
19 7E1C C607                  ADD  A,7     ;BETU
20 7E1E C9           CONV1:   RET
21                            END
```

Az ASCII karakteres formában érkező bináris vagy hexadecimális számok átalakítása ugyanilyen egyszerű feladat. Ha az ASCII karakterek bináris számot adnak meg, akkor 30H kivonásával 0-t vagy 1-et kapunk, és így bitenként összegyűjthetjük a 8-bites vagy 16 bites bináris értéket. Hexadecimális számok esetén az ASCII karakterekből kivonunk 30H-t, majd ha az eredmény 9-nél nagyobb, kivonunk még 7-et, és a kapott értékeket 4-bitenként összegyűjtjük.

Bonyolultabb feladat a decimális számokat megadó ASCII karakterek bináris egész számmá alakítása. A 0...9 számjegyeknek megfelelő 30H...39H ASCII karakterekből 30H-t kivonva 4-bites, binárisan kódolt decimális számjegyet kapunk, amelyet 10 megfelelő hatványával meg kell szorozni. Pl. ha az 123 karaktersorozatot alakítjuk számmá, akkor az 1-et megszorozzuk 100-zal, a 2-t megszorozzuk 10-zel, 3-at változatlanul hagyjuk, és az eredményeket összeadjuk.

Decimális számok kiíratásakor a 8-bites vagy 16 bites bináris egész számot ismételt 10-zel osztás segítségével alakíthatjuk át ASCII karakteres alakra. A maradékból mindig 30H hozzáadásával ASCII karaktert képzünk, a kiíratás előtt egy munkaterületen összegyűjtjük a karaktereket. Ha megfordítjuk az eljárást, és a legmagasabb 10 hatvánnyal (16 bites értékeknél 1000-rel) kezdjük az osztást, akkor a kapott karaktereket egyenként rögtön kiíthatjuk.

Karakterkeresés

Szövegben egy adott karaktert úgy kereshetünk meg, hogy a CP utasítással a szöveg karaktereit az első karaktertől kezdve sorra összehasonlítjuk a keresett karakterrel, és közben számláljuk, hogy hányadik karakternél tartunk. A következő program "/" jelet keres a HL regiszterpár által mutatott szövegben. Ha talál "/" jelet, akkor a TALALT címkére ugrik, és a B regiszterből

meg lehet tudni, hogy a keresett karakter a szöveg hányadik karaktere. Ha nem talál "/" jelet, akkor a szokásos VEGE címkéhez kerül.

```
1                   ;38. LISTA
2                           ORG  7000H
3                           LOAD 7000H
4  7000 211170              LD   HL,SZOVEG;SZOVEGRE MUTAT
5  7003 0606                LD   B,6      ;SZAMLALO
6  7005 3E2F                LD   A,'/'    ;KERESETT KARAKTER
7  7007 BE          CIKL:   CP   (HL)     ;OSSZEHASONLITAS
8  7008 2805                JR   Z,TALALT;UGRIK, HA TALALT
9  700A 23                  INC  HL       ;KOVETKEZO
10 700B 10FA                DJNZ CIKL     ;KARAKTER
11 700D 18FE        VEGE:   JR   VEGE
12 700F 18FE        TALALT: JR   TALALT
13 7011 4142432F    SZOVEG: DB   'ABC/DE'
13 7015 4445
14                          END
```

Az eljárás nem túlságosan bonyolult, van azonban egyszerűbb lehetőség is. A blokkmozgatáshoz hasonló működésű

CPI	ED A1
CPIR	ED B1
CPD	ED A9
CPDR	ED B9

blokk-összehasonlító utasítások az A regiszter tartalmát összehasonlítják a HL regiszterpár által mutatott tárcím tartalmával. Az összehasonlítás – amely valójában kivonás – eredményét az S és Z jelzőbitek mutatják. Ha az összehasonlítás során a 4. bitről beváltás volt szükséges, akkor a H jelzőbit törlődik, egyébként H értéke 1. Az N jelzőbit értéke mindig 1 lesz, a C jelzőbit változatlan marad. Az összehasonlítás után a számlálóként működő BC regiszterpár tartalma eggyel csökken. Ha BC tartalma eléri a nullát, akkor P/V törlődik, egyébként P/V értéke 1.

Az összehasonlítás után a HL regiszterpár tartalma CPI és CPIR utasítások esetében eggyel növekszik, a CPD és CPDR utasítások esetében eggyel csökken. A CPIR és CPDR utasítások végrehajtása után a programszámláló értéke 2-vel csökken, így ezek az utasítások a keresett karakter megtalálásáig vagy P/V törlődéséig automatikusan ismétlődnek. A CPI és CPD utasítások minden egyes végrehajtása után a következő utasításra adódik a vezérlés, ami lehetőséget nyújt az adatok egyenkénti feldolgozására.

A következő program blokk-összehasonlítással keres "/" jelet. A program akkor lép a VEGE címkére, amikor az első "/" jelet megtalálja, vagy ha a BC regiszterpár tartalma nullára csökken.

```
1                   ;39. LISTA
2                           ORG  7000H
3                           LOAD 7000H
4  7000 210C70              LD   HL,SZOVEG;SZOVEGRE MUTAT
5  7003 010600              LD   BC,6     ;SZAMLALO
6  7006 3E2F                LD   A,'/'    ;KERESETT KARAKTER
7  7008 EDB1                CPIR          ;BLOKK KERESES
8  700A 18FE        VEGE:   JR   VEGE
9  700C 4142432F    SZOVEG: DB   'ABC/DE'
9  7010 4445
10                          END
```

A CPD utasítással (block ComPare Decrement) a szöveg végétől kezdhetjük a keresést. A következő programban a szöveg végétől kiindulva "/*" kétkarakteres jelsorozatot keresünk. A karakter megtalálását a Z jelzőbit, a BC regiszterpárba töltött byte-szám nullára csökkenését a P/V jelzőbit mutatja.

```
 1                    ;40. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 212670              LD   HL,SZOVEG+14;SZOVEG VEGE
 5 7003 010F00              LD   BC,15  ;SZAMLALO
 6 7006 3E2A        CIKL1:  LD   A,'*'  ;MASODIK KARAKTER
 7 7008 EDA9        CIKL2:  CPD         ;KERESES
 8 700A 2805                JR   Z,CIKL3;TALALT
 9 700C EA0870              JP   PE,CIKL2;KOVETKEZO
10 700F 18FE        VEGE1:  JR   VEGE1  ;NEM TALALT
11 7011 3E2F        CIKL3:  LD   A,'/'  ;ELSO KARAKTER
12 7013 BE                  CP   (HL)   ;OSSZEHASONLITAS
13 7014 20F0                JR   NZ,CIKL1;NEM TALALT
14 7016 18FE        VEGE2:  JR   VEGE2  ;MINDKETTO MEGVAN
15 7018 4142432F SZOVEG:    DB   'ABC/DE/*FGH/123'
15 701C 44452F2A
15 7020 4647482F
15 7024 313233
16                          END
```

Táblázatok

Assembler programokban gyakran használunk különféle táblázatokat. A legegyszerűbb táblázat rendezetlen formában egyenlő hosszúságú adatokat tartalmaz. Ha a táblázatban keresünk egy adatot, akkor annak sorszámát vagy indexét általában további információk elérésére vagy a meghívandó szubrutin kiválasztására használjuk fel.

Ha pl. olyan táblázatunk van, amely csupa egybetűs parancsot tartalmaz, és a billentyűzeten beadunk egy parancsot, akkor megvizsgálhatjuk, hogy a beütött parancs szerepel-e a táblázatban, és ha igen, hányadik elemként. Az elem sorszáma alapján azután kiválaszthatjuk a megfelelő feldolgozó szubrutint. A következő szubrutin megvizsgálja, hogy az A regiszterben levő karakter szerepel-e a TABLA címkénél lerakott táblázatban (valójában ez a táblázat nem más, mint egy ASCII karaktersorozat), és visszatéréskor a BC regiszterpárban adja a táblaelem indexét. Arról, hogy a beütött karakter hogyan jut az A regiszterbe, később lesz szó.

```
 1                    ;41. LISTA
 2                          ORG  7E00H
 3                          LOAD 7E00H
 4 7E00 21097E              LD   HL,TABLA;TABLAZAT CIME
 5 7E03 010900              LD   BC,9   ;TABLA HOSSZA
 6 7E06 EDB1                CPIR        ;KERESES
 7 7E08 C9                  RET
 8 7E09 5A454E47 TABLA:     DB   'ZENGCFXMQ'
 8 7E0D 4346584D
 8 7E11 51
 9                          END
```

Ha a keresés sikerrel járt, akkor a visszatéréskor a Z jelzőbit értéke 1, és a HL regiszterpár a keresett táblaelem címét tartalmazza. Ha a keresett elem a táblázatban nem szerepel, akkor a visszatéréskor a Z jelzőbit értéke 0, és a HL

regiszterpár a táblázat utáni első byte címét tartalmazza. A fenti szubrutint felhasználhatjuk a következő programban, amely a keresés eredménye alapján az adott parancsnak megfelelő címkére ugrik. A programban indexként a parancs sorszámának kétszeresét használva választjuk ki a SZUB.Z címkétől lerakott relatív ugró utasítások közül a megfelelőt.

```
 1                    ;42. LISTA
 2
 3                    ;FOPROGRAM
 4
 5                            ORG  7000H
 6                            LOAD 7000H
 7 7000 3E47                  LD   A,'G'   ;INPUT KARAKTER
 8 7002 CD007E                CALL KERES   ;TABLA KERESES
 9 7005 C21570                JP   NZ,NINCS;NEM TALALT
10 7008 01097E                LD   BC,TABLA;TABLA CIME
11 700B B7                    OR   A       ;C JELZOBIT TORLESE
12 700C ED42                  SBC  HL,BC   ;INDEX+1
13 700E 2B                    DEC  HL      ;INDEX
14 700F 29                    ADD  HL,HL   ;CIMZESI ELTOLAS
15 7010 011770                LD   BC,SZUBZ;UGROTABLA ELEJE
16 7013 09                    ADD  HL,BC   ;SZUB.X CIM
17 7014 E9                    JP   (HL)
18 7015 18FE          NINCS:  JR   NINCS   ;NEM TALALT
19 7017 18FE          SZUBZ:  JR   SZUBZ   ;PARANCSOK
20 7019 18FE          SZUBE:  JR   SZUBE   ;FELDOLGOZASA
21 701B 18FE          SZUBN:  JR   SZUBN
22 701D 18FE          SZUBG:  JR   SZUBG
23 701F 18FE          SZUBC:  JR   SZUBC
24 7021 18FE          SZUBF:  JR   SZUBF
25 7023 18FE          SZUBX:  JR   SZUBX
26 7025 18FE          SZUBM:  JR   SZUBM
27 7027 18FE          SZUBQ:  JR   SZUBQ
28
29                    ;TABLAKERESO SZUBRUTIN
30
31                            ORG  7E00H
32                            LOAD 7E00H
33 7E00 21097E        KERES:  LD   HL,TABLA;TABLAZAT CIME
34 7E03 010900                LD   BC,9    ;TABLA HOSSZA
35 7E06 EDB1                  CPIR         ;KERESES
36 7E08 C9                    RET
37 7E09 5A454E47      TABLA:  DB   'ZENGCFXMQ'
37 7E0D 4346584D
37 7E11 51
38                            END
```

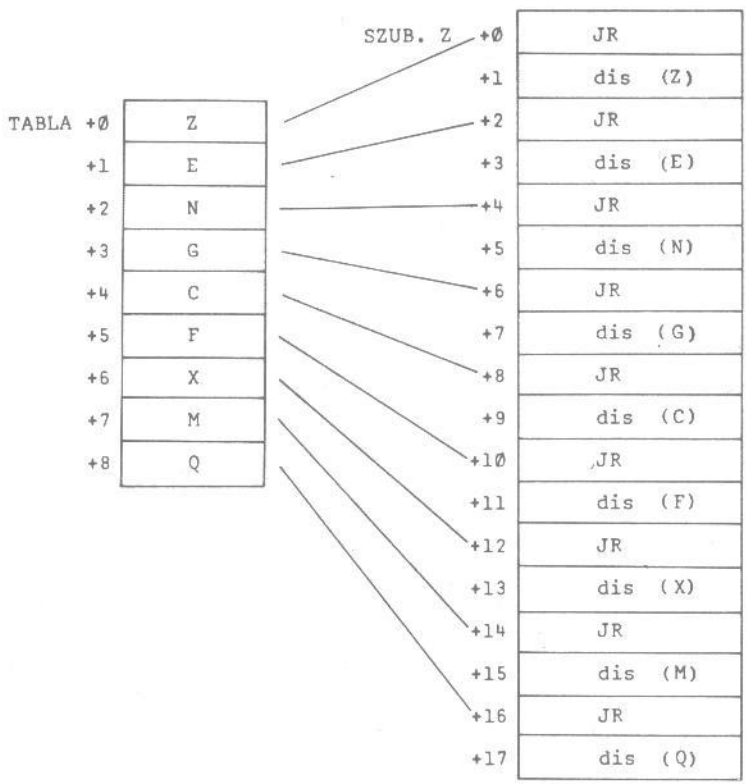
Az indexelési algoritmust a 10.1. ábra mutatja. Ha az adatok rendezetlenek, és a táblázat mindegyik elemét nagyjából ugyanannyiszor keressük, akkor a keresés során átlagosan az elemek feléig kell eljutni, hiszen van, amikor a táblázat elején, van, amikor a táblázat végén levő elemet keressük. Nagy táblázatoknál a leírt eljárás hosszadalmas, ezért ajánlatos az elemeket csoportosítani, rendezni, és a táblázat felépítésének megfelelő gyors keresési algoritmust alkalmazni.

Előző példánknál maradva a parancsokat és a hozzájuk tartozó vezérlésátadó utasításokat egy táblázatban is elrendezhetjük oly módon, hogy minden parancs után a megfelelő JR utasítás következzék:

```
TABLA:  DB 'Z'
        JR SZUB.Z
        DB 'E'
```

```
JR SZUB.E
DB 'N'
JR SZUB.N
.
.
.
```

Az ily módon elrendezett táblázatban blokk-összehasonlítással keresve a HL regiszterpár által mutatott tárcímet minden egyes összehasonlítás után 3-mal kell növelni. Az összehasonlítás után a HL regiszterpár éppen a megfelelő JR utasítás címét tartalmazza. A következő szubrutint az ilyen típusú táblázatokban keresés céljára állítottuk össze. A szubrutin hívása előtt az A regiszterbe a keresett karaktert, a HL regiszterpárba a táblázat kezdőcímét, a DE regiszterpárba a táblázat elemeinek hosszúságát, a C regiszterbe az elemek számát kell betölteni. A szubrutinból kilépéskor sikeres keresés után a Z jelzőbit értéke 1, és a HL regiszterpár a keresett táblaelemre mutat, sikertelen keresés után Z értéke 0 és HL a tábla végére mutat.



10.1. ábra. Táblázatok és indexelés

```
 1                    ;43. LISTA
 2                          ORG  7E00H
 3                          LOAD 7E00H
 4 7E00 0600      KERES:    LD   B,0      ;ELEMSZAM BC-BEN
 5 7E02 EDA1      CIKL:     CPI           ;OSSZEHASONLITAS
 6 7E04 2B                  DEC  HL       ;ELEM CIME+1-1
 7 7E05 CA0E7E              JP   Z,VEGE   ;TALALT
 8 7E08 E20E7E              JP   PO,VEGE;NINCS
 9 7E0B 19                  ADD  HL,DE    ;KOVETKEZO
10 7E0C 18F4                JR   CIKL
11 7E0E C9        VEGE:     RET
12                          END
```

Rendezés

Ha sok adatunk van, akkor a gyors elérés érdekében ajánlatos rendezetten tárolni azokat. Az adatokat a numerikus értékek, ill. az ASCII kódok szerint növekvő vagy csökkenő sorrendben tárolhatjuk. Ha a már rendezett adatok közé újabb adatot veszünk fel, akkor azt a megfelelő helyre kell beszúrni. Ha a meglevő adatokat módosítjuk, töröljük, akkor az adatokat esetleg újra kell rendezni. A rendezésre, beszúrásra, keresésre olyan sok módszer van, hogy külön könyvet írhatnánk róluk. Itt csupán az egyik legegyszerűbb rendezési eljárásra, a buborékos rendezésre mutatunk egy példát.

A rendezés során minden elemet összehasonlítunk a következő elemmel, és ha az alacsonyabb értékű, akkor a két elemet felcseréljük. Ezután a következő elempárral folytatjuk az összehasonlítást, majd ha az adatok végéhez értünk, újra előlről kezdjük a páronkénti összehasonlítást. Az eljárást addig folytatjuk, amíg már nincs szükség újabb cserékre. A következő program 16 egybyte-os elemet rendez ezzel az eljárással:

```
 1                    ;44. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 DD212A70 CIKL:      LD   IX,TABLA;TABLA KEZDOCIME
 5 7004 060F                LD   B,15     ;ADATOK SZAMA
 6 7006 0E00                LD   C,0      ;CSERE JELZO
 7 7008 DD7E00   CIKL1:     LD   A,(IX+0);ADAT BETOLTES
 8 700B DDBE01              CP   (IX+1) ;KOVETKEZO UGYANAZ?
 9 700E CA1F70              JP   Z,MARAD;EGYENLO
10 7011 DA1F70              JP   C,MARAD;KOVETKEZO NAGYOBB
11 7014 DD4E01              LD   C,(IX+1);KOVETKEZO BETOLTES
12 7017 DD7701              LD   (IX+1),A;ELOZO TAROLAS
13 701A DD7100              LD   (IX+0),C;KOVETKEZO TAROLAS
14 701D 0E01                LD   C,1      ;CSERE JELZOT ALLIT
15 701F DD23     MARAD:     INC  IX       ;KOVETKEZO ADATPAR
16 7021 10E5                DJNZ CIKL1    ;SZAMLALOT CSOKKENT
17 7023 CB41                BIT  0,C      ;VOLT-E CSERE?
18 7025 C20070              JP   NZ,CIKL;VOLT, UJRA KEZD
19 7028 18FE     VEGE:      JR   VEGE     ;RENDEZES VEGE
20               TABLA:     DS   16       ;16 ADAT HELYE
21                          END
```

Töltsük fel a táblázatot adatokkal, futtassuk le a programot, és írassuk ki a képernyőre a rendezett táblázatot!

A buborékos rendezés folyamatát jól követhetjük az alábbi programmal, amely a képernyő sorainak közepére (a 3C20H, 3C60H, ... , 3FE0H címekre) helyezett adatokat lassítva rendezi. Írjuk adatainkat a képernyő megfelelő helyeire, indítsuk el a programot, és figyeljük a képernyőt!

```
 1                  ;45. LISTA
 2                           ORG  7000H
 3                           LOAD 7000H
 4 7000 DD21203C CIKL:       LD   IX,3C20H;ELSO SOR KOZEPE
 5 7004 114000               LD   DE,64   ;64 KARAKTERES SOR
 6 7007 060F                 LD   B,15    ;SOROK SZAMA
 7 7009 0E00                 LD   C,0     ;CSERE JELZO
 8 700B DD7E00   CIKL1:      LD   A,(IX+0);ADAT BETOLTES
 9 700E DDBE40               CP   (IX+64);KOVETKEZO UGYANAZ?
10 7011 CA2B70               JP   Z,MARAD;EGYENLO
11 7014 DA2B70               JP   C,MARAD;KOVETKEZO NAGYOBB
12 7017 DD4E40               LD   C,(IX+64);KOVETKEZO ADAT
13 701A DD7740               LD   (IX+64),A;ELOZO ADAT TAROLAS
14 701D DD7100               LD   (IX+0),C;KOVETKEZO TAROLAS
15 7020 210000               LD   HL,0    ;SZAMLALO
16 7023 23      LASSAN:      INC  HL      ;LASSITO CIKLUS
17 7024 CB7C                 BIT  7,H
18 7026 CA2370               JP   Z,LASSAN
19 7029 0E01                 LD   C,1     ;CSERE JELZOT ALLIT
20 702B DD19    MARAD:       ADD  IX,DE   ;KOVETKEZO SOR
21 702D 10DC                 DJNZ CIKL1   ;SZAMLALO CSOKKEN
22 702F CB41                 BIT  0,C     ;VOLT-E CSERE?
23 7031 C20070               JP   NZ,CIKL;VOLT, UJRA KEZD
24 7034 18FE    VEGE:        JR   VEGE
25                           END
```

11. fejezet

HT–1080Z ROM-rutinok

A BASIC értelmező szubrutinjainak ismeretében lényegesen gyorsabbá, egyszerűbbé tehetjük az assembler programozást, ugyanis programjainkban sokszor olyan feladatokat kell megoldanunk, amelyekre a BASIC értelmezőben már kész szubrutinok vannak. Ebben a fejezetben a HT-1080Z számítógép ROM-rutinjainak ilyen alkalmazási lehetőségeivel fogunk megismerkedni.

Először a HT-1080Z tár elrendezését, a számoknak a BASIC értelmező által biztosított ábrázolási lehetőségeit ismertetjük, majd az adatmozgató, adatkonvertáló, billentyűzetről olvasó, képernyőre író, aritmetikai műveleteket végző és függvények helyettesítési értékeit kiszámoló ROM-rutinok alkalmazásával foglalkozunk.

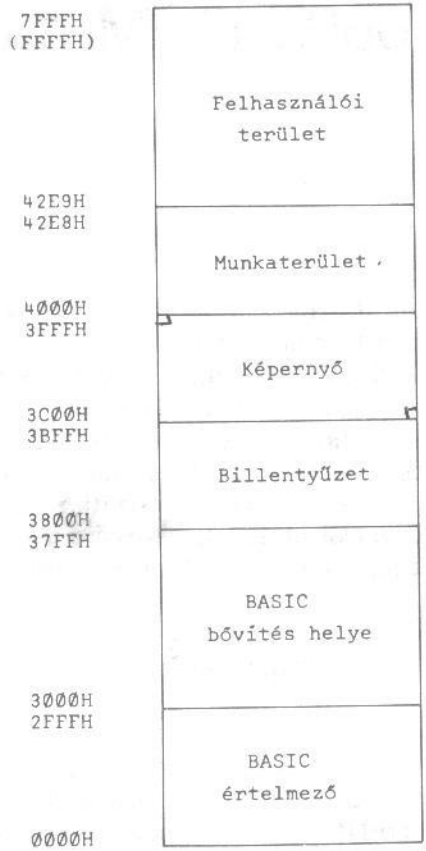
Tárelrendezés

A HT-1080Z számítógép tára két alapvető részből áll, a LEVEL II. BASIC értelmezőt tartalmazó, csak olvasható tárból (ROM), és a felhasználói tárból, amely írható és olvasható (RAM). Ezekhez különböző célokra lefoglalt, a programozó számára korlátozottan használható speciális területek (perifériacímek, billentyűzetmátrix, képernyő-memória) csatlakoznak.

A BASIC értelmező a tár 0000H-tól 2FFFH-ig terjedő alsó 12 Kbyte-ját foglalja el. Ezt a területet nem lehet felülírni, az ezekre a címekre kiadott tároló utasítások nem változtatják meg a tár tartalmát. A BASIC értelmező alapvető szubrutinjainak belépési pontjai megtalálhatók a Függelék ROM-táblázatában. Némelyik szubrutin CALL vagy JP utasítással egyszerűen elérhető, ilyen pl. a képernyő törlését végző szubrutin (CALL 01C9H), vagy a BASIC-be újra belépő, READY üzenetet kiadó szubrutin (JP 1A19H). Más szubrutinok meghatározott regiszterekben vagy tárcímeken elhelyezett bemenő adatokat várnak, ill. kimenő adatokat adnak vissza. Megjegyezzük, hogy a BASIC értelmezőnek több, nem lényegesen különböző változata van.

A következő tárcímeken, a 3000H-tól 37DDH-ig terjedő 2 Kbyte-on BASIC bővítés helyezhető el. Erre a területre a számítógép tervezett felhasználását figyelembe véve többféle, lényegesen különböző bővítés helyezhető el, amint azt a HT-1080Z számítógépeken is megfigyelhetjük. A következő néhány cím, 37DEH-tól 37FFH-ig üres.

A 3800H-tól a 3BFFH-ig a billentyűzet mátrixot találjuk. Itt valójában nincs tár, mégis ezekről a címekről olvasva olyan, mintha a tárból olvasnánk. A billentyűzet mátrix lényegében egy sereg, a cím- és adatvonalakat összekötő kapcsoló. Ha egy billentyűt megnyomunk, akkor a billentyűzet mátrix



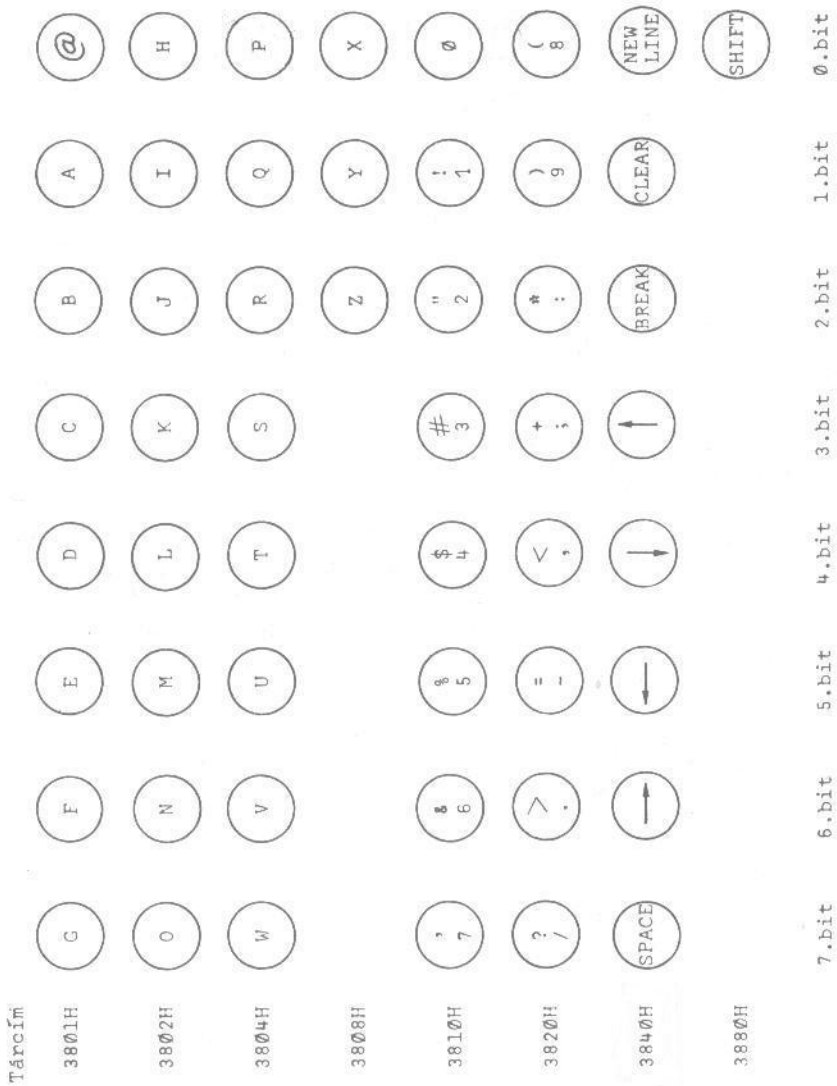
11.1. ábra HT-1080Z tárelrendezés

megfelelő bitjének 1 lesz az értéke. Az így kapott információt a BASIC értelmező szubrutinjai ASCII karakterekké alakítják át. A 11.2. ábra mutatja, hogy az egyes billentyűk megnyomásakor melyik címen melyik bitnek lesz 1 az értéke. Ha egyszerre több billentyűt nyomunk meg, akkor a megfelelő bitek mindegyikének 1 lesz az értéke.

A 3C00H-tól 3FFFH-ig terjedő közvetlenül írható és olvasható tár a képernyő 16*64=1024 karakterét tárolja.

A 4000H címen kezdődő RAM első 744 byte-ja, a 4000H-tól 42E8H címekig a BASIC és a lemezes rendszer munkaterülete. A BASIC értelmező itt tárolja a működéséhez szükséges változókról szóló információkat, itt vannak a ki- és bemenet lebonyolításához szükséges pufferterületek. A munkaterület felépítését a Függelék RAM-táblázata mutatja.

A 42E9H-tól a tár végéig (16 Kbyte-os gépen a 7FFFH címig, 48 Kbyte-os gépen az FFFFH címig) terjedő terület a felhasználói tár, ide kerülnek a BASIC programok és változók. A gépi kódú programok betöltésére, futtatására is ezt a területet használhatjuk.



11.2. ábra. Billentyűzetmátrix

Még egy kis aritmetika

A 9. fejezetben láttuk, hogy a Z80 mikroprocesszor aritmetikai utasításaival 8-bites és 16 bites előjel nélküli és előjeles számokat lehet összeadni és kivonni, a szorzás és osztás elvégzéséhez pedig rövidebb–hosszabb szubrutinokat kell írni.

A BASIC értelmező a 8-bites számábrázolásokat csak belső célra, a 16 bites előjel nélküli számábrázolást pedig csak speciális célra (sorszámok, sormutatók, tárcímek stb.) alkalmazza. A BASIC egész típusú változói 16 bites előjeles egész számok. A gyakorlatban nagyobb számokra és törtszámokra is szükség van, ezért a BASIC változók egyszeres vagy kétszeres pontosságú lebegőpontos számok is lehetnek. Az aritmetikai ROM-rutinok ez utóbbi három változótípussal végzik el az összeadást, kivonást, szorzást és az osztást.

A BASIC egész változókkal itt nem foglalkozunk, hiszen azok már korábban megismert, kettes komplemens ábrázolású, 16 bites előjeles egész számok.

Az egyszeres pontosságú lebegőpontos számok négy byte-ot foglalnak el, amelyből három byte a mantissza, egy byte a kitevő. A mantissza 24 bitje közül a legértékesebb a normalizálás miatt mindig 1 lenne, ezért az felhasználható az előjel ábrázolására. Ha ennek a bitnek a helyére 0 kerül, akkor a szám pozitív, ha 1, akkor a szám negatív. A mantissza három byte-ja a tárban fordított sorrendben kerül tárolásra, így a legalsó címen a legkevésbé értékes byte-ot, a középső címen a középső byte-ot és a legmagasabb címen a legértékesebb byte-ot találjuk. A negyedik byte a kettes alapú kitevő, amelyben a legértékesebb bit, az előjelbit pozitív számok esetén 1, negatív számok esetén 0, így a valódi kitevőből 128 hozzáadásával kapjuk a lebegőpontos kitevőt. A mantissza 24 bitje tízes számrendszerben 6 értékes jegy pontosságot jelent, a -128 és $+127$ közötti kettes alapú kitevővel 10^{-38} és 10^{38} nagyságrend közötti számok ábrázolhatók.

Példaként alakítsuk át az 123456 tízes számrendszerbeli számot egyszeres pontosságú lebegőpontos számmá:

$$123456_{10} = 11110001001000000_2,$$

amiből a mantisszát úgy kapjuk, hogy a tizedespontot 17 jeggyel balra, az első értékes jegy elé toljuk. Ezzel azonban a számot 2^{17}-nel osztottuk, így a kitevőbe 17-nek kell kerülnie:

$$0.11110001001000000_2 * 2^{17}.$$

A mantisszát 24 jegyre egészítjük és nyolcas csoportokra bontjuk:

11110001	00100000	00000000
felső byte	középső byte	alsó byte

A legértékesebb bit a normalizálás miatt mindig 1, ezért annak helyére az előjelbitet írjuk, amely pozitív számok esetében 0:

01110001	00100000	00000000
felső byte	középső byte	alsó byte

A kitevő $17_{10} = 00010001_2$, amiből az előjelbit hozzáírásával 10010001_2 adódik. A mantissza tárolása fordított sorrendben történik, így a fenti szám tárolása a következő:

00000000	00100000	01110001	10010001
alsó byte	középső byte	felső byte	kitevő

ami hexadecimálisan 00H, 20H, 71H, 91H. A decimális −123456 tárolása ettől csak a mantissza előjelbitjében különbözik:

00000000	00100000	11110001	10010001
alsó byte	középső byte	felső byte	kitevő

ami hexadecimálisan 00H, 20H, F1H, 91H.

A kétszeres pontosságú lebegőpontos számok ugyanilyen szerkezetűek, de nyolc byte-ot foglalnak el, amiből hét byte a mantissza és egy byte a kitevő. A mantissza 54 bitjével és az egybyte-os kitevővel 10^{-38} és 10^{38} nagyságrend közötti számokat 16 értékes jegy pontossággal ábrázolhatunk.

Az aritmetikai műveleteket végző és a matematikai függvények helyettesítési értékeit kiszámoló ROM-rutinok regiszterpárokban, ill. a munkaterület meghatározott részén várják az adatokat és az eredményeket ugyanezekben a regiszterpárokban, ill. ugyanezeken a munkaterületeken adják vissza. A BASIC változók tárolására az aritmetikai ROM-rutinok két munkaterületet használnak, mégpedig a 411DH...4124H (16669...16676) címeken elhelyezett szoftver akkumulátort (SA) és a 4127H...412EH (16679...16686) címeken elhelyezett másodlagos szoftverakkumulátort (SA1). Azt, hogy a szoftverakkumulátorokban elhelyezett számok milyen típusúak, a 40AFH (16559) címen elhelyezett típusjelző mutatja. Ha a szoftverakkumulátorban egész szám van, akkor a típusjelző értéke 2, ha egyszeres pontosságú lebegőpontos szám, akkor a típusjelző 4, ha pedig kétszeres pontosságú lebegőpontos szám, akkor a típusjelző 8 (a típusjelző a szám tárolásához szükséges byte-ok számát mutatja). A szoftverakkumulátorokban az egész számok az 5. és 6. byte-okon (a 4121H és 4122H, ill. 412BH és 412CH címekek), az egyszeres pontosságú számok az 5...8. byte-okon (a 4121H...4124H, ill. 412BH...412EH címeken) helyezkednek el, a kétszeres pontosságú számok pedig mind a nyolc byte-ot kitöltik.

A ROM-rutinok az általános regiszterek közül a DE és HL regiszterpárokat egész változók, a BCDE regiszternégyest pedig egyszeres pontosságú lebegőpontos számok tárolására használják. Ez utóbbi esetben a kitevő a B regiszterben, a mantissza felső byte-ja a C regiszterben, a középső byte-ja a D regiszterben, alsó byte-ja az E regiszterben kerül elhelyezésre.

Az egész számokkal műveleteket végző ROM-rutinok általában a DE és HL regiszterpárokat, valamint az SA elsődleges szoftverakkumulátort használják. Az egyszeres pontosságú lebegőpontos számokkal dolgozó ROM-rutinok a BCDE regiszternégyest és az SA szoftverakkumulátort használják, a műveletek eredményét az SA szoftverakkumulátorba írják. Az egyszeres pontosságú szorzás és a kétszeres pontosságú aritmetikai műveletek elvégzéséhez a ROM rutinok a 414AH...4151H munkaterületet is használják.

Megjegyezzük, hogy ASCII karakterváltozók esetén a típusjelző értéke 3, az SA szoftverakkumulátorban a szövegváltozó hossza a 4101H címen, a szövegváltozó kezdőcíme a 4102H...4103H címeken található.

Az aritmetikai ROM-rutinok alkalmazásához az adatokat megfelelő módon be kell tölteni a regiszterekbe, el kell helyezni a szoftverakkumulátorokban, továbbá gondoskodni kell a típusjelző értékének beállításáról. A típusjelző értékét közvetlenül is betölthetjük, de vannak olyan ROM-rutinok, amelyek elvégzik ezt a feladatot. A

```
CALL 0A9DH        SETINT
CALL 0AEFH        SETSGL
CALL 0AECH        SETDBL
```

ROM-rutinok a 40AFH címen levő típusjelzőt sorra a 2, 4, ill. 8 értékre állítják. A SETINT és SETSGL* rutinok csak az A regisztert, a SETDBL rutin az A regisztert és a BC regiszterpárt használja. A szubrutinokat a hexadecimális címek szerint lehet meghívni, a jobb oldalra csak a könnyebb megjegyezhetőség kedvéért írunk emlékeztető nevet (assemblerrel programozva a hexadecimális címhez EQU pszeudoutasítással az emlékeztető nevet hozzárendelve név szerint hívhatjuk meg a szubrutint).

Az elsődleges szoftverakkumulátor tartalmának vizsgálatára is van ROM-rutin. A

```
CALL 25D9H        TESTSA
```

rutin megvizsgálja az elsődleges szoftverakkumulátor tartalmát és a jelzőbiteket a következőképpen állítja:

Z=1	szövegváltozó
S=1	egész változó
P/V=1	egyszeres pontosságú változó
C=0	kétszeres pontosságú változó

A jelzőbitek vizsgálatával egyszerűen megoldható a szoftverakkumulátor tartalma szerinti elágazás:

```
TESTSA: EQU 25D9H
        CALL TESTSA
        JP Z,STRING          ; UGRIK, HA STRING
        JP M,EGESZ           ; UGRIK, HA EGESZ
```

* Megjegyezzük, hogy a TRS80 irodalomban az SGN műveleteket végrehajtó rutinok betűjele SGL.

	Tipus-jelző	Szoftverakkumulátor							
SA:	40AFH	411DH	411EH	411FH	4120H	4121H	4122H	4123H	4124H
SA 1		4127H	4128H	4129H	412AH	412BH	412CH	412DH	412EH
Egész	2					alsó byte	felső byte		
STRING	3					hossz	cim alsó	cim felső	
Egyszeres	4					alsó byte	2.byte	felső byte	kitevő
Kétszeres	8	alsó byte	2.byte	3.byte	4.byte	5.byte	6.byte	felső byte	kitevő

11.3. ábra. Szoftverakkumulátor

```
JP PO,EGYSZ        ; UGRIK, HA EGYSZERES
JP NC,KETSZ        ; UGRIK, HA KETSZERES
```

Megjegyezzük, hogy ugyanez a ROM-rutin az RST 20H utasítással is meghívható.

Változók mozgatása

Az aritmetikai ROM-rutinok alkalmazásához a változókat be kell tölteni a megfelelő regiszterekbe, ill. szoftverakkumulátorba. Ezt az adatmozgatást teszik egyszerűbbé a következő ROM-rutinok. A HL regiszterpár által mutatott egyszeres pontosságú változót a BCDE regiszternégyesbe viszi a

```
CALL 09C2H        LDHLRA
```

szubrutin (az emlékeztető névben LD az adatmozgatást, HL a regiszterpárt, RA a regiszterakkumulátornak is nevezett regiszternégyest jelöli). Az egyszeres pontosságú számot a BCDE regiszternégyesből az SA szoftverakkumulátorba a

```
CALL 09B4H        LDRASA
```

szubrutinnal vihetjük át. Az egyszeres pontosságú számot a HL regiszterpár által mutatott tárcímről az SA szoftverakkumulátorba az előző két szubrutint egymás után alkalmazó

```
CALL 09B1H        LDHLSA
```

szubrutinnal vihetjük át.

Az SA szoftverakkumulátorban levő egyszeres pontosságú számot a BCDE regiszternégyesbe tölthetjük a

```
CALL 09BFH        LDSARA
```

szubrutinnal, a HL regiszterpár által mutatott tárcímre, ill. az SP által mutatott veremterületre tárolhatjuk a

```
CALL 09CBH        LDSAHL
CALL 09A4H        LDSASP
```

szubrutinokkal.

A 40AFH típusjelző értékétől függő számú byte-ot lehet a

```
CALL 09FCH        MVSA
```

szubrutinnal az SA elsődleges szoftverakkumulátorból az SA1 másodlagos szoftverakkumulátorba, a

```
CALL 09F4H        MVSA1
```

szubrutinnal SA1-ből SA-ba átmozgatni.

Az előző szubrutin része a

```
CALL 09F7H        MVHLSA
```

szubrutin, amely a HL regiszterpár által mutatott címről mozgat a típusjelző értékétől függően egyszeres vagy kétszeres pontosságú számot az SA szoftverakkumulátorba.

A HL regiszterpárban levő egész számot az SA szoftverakkumulátorba tölti, és a típusjelző értékét 2-re állítja a

```
CALL 0A9AH          HLTOSA
```

szubrutin.

A DE regiszterpár által mutatott tárterületről a HL regiszterpár által mutatott tárterületre a

```
CALL 09D3H          MVDEHL
CALL 09D6H          MVA
CALL 09D7H          MVB
```

szubrutinokkal sorra a típusjelző értékétől függő számú byte-ot, az A regiszter, ill. a B regiszter tartalma által meghatározott számú byte-ot lehet átmozgatni. A fordított irányú, a HL regiszterpár által mutatott tárterületről a DE regiszterpár által mutatott tárterületre irányuló adatmozgatás úgy oldható meg, hogy a fenti szubrutinok meghívása előtt EX DE,HL utasítással felcseréljük a két regiszterpár tartalmát. Az MVDEHL szubrutin ilyen párja a

```
CALL 09D2H          MVHLDE
```

szubrutin, az MVA és MVB szubrutinoknak nincs megfordítása.

Az eddigi ROM-rutinok nem nélkülözhetetlen segédeszközök, hiszen néhány assembler utasításban bármelyiket megírhatjuk, de időt, fáradságot és sok-sok tárhelyet takarítunk meg, ha használjuk ezeket a szubrutinokat. A BASIC értelmezőt átgondoltan tervezték meg, így pl. az utolsó négy ROM-rutint összesen tíz utasításban írták meg. Némi előkészítés után az MVSAHL, MVSA és MVSA1 szubrutinok is ezekkel az utasításokkal végzik az adatmozgatást (ROM-lista 09D24-től 09DEH-ig).

```
09D2 EB          EX DE,HL
09D3 3AAF40      LD A,(40AFH)
09D6 47          LD B,A
09D7 1A          LD A, (DE)
09D8 77          LD (HL),A
09D9 13          INC DE
09DA 23          INC HL
09DB 05          DEC B
09DC 20F9        JR NZ,09D7H
09DE C9          RET
```

(A figyelmes Olvasó bizonyára észreveszi, hogy a DEC és JR utasítások helyett rövidebb lenne DJNZ utasítást írni. Ezt a Microsoft LEVEL II. BASIC értelmezőt készítő Paul Allen és Bill Gates is tudják, csakhogy ezt a programot nem Z80 mikroprocesszorra, hanem még régen, a DJNZ utasítást nem ismerő 8080 mikroprocesszorra írták. Egyébként ezzel magyarázható a BASIC értelmező nem éppen látványos sebessége is.)

Billentyűzetolvasás

A billentyűzetről úgy olvashatunk be egy karaktert, hogy megvizsgáljuk a 3800H...3BFFH címeken található billentyűzetmátrix bitjeit, és ha azok között nullától különbözőt találunk, akkor kikeressük az ahhoz tartozó ASCII karaktert és azt a megfelelő helyre betöltjük. Ezt a feladatot a

CALL 002BH KBDSCN

szubrutinnal végeztethetjük el, amely megvizsgálja a billentyűzetmátrixot, és a visszatéréskor az A regiszterbe a megnyomott billentyűhöz tartozó ASCII kódot tölti. Ha éppen egyetlen billentyű sincs megnyomva, akkor az A regiszter 00H-t tartalmaz. (A tulajdonképpeni billentyűzetvizsgálatot a 4015H...401CH billentyűzet-ellenőrző táblában lerakott 03E3H címen kezdődő szubrutin végzi. A fenti szubrutin a DE regiszterpárba a billentyűzet-ellenőrző tábla címét tölti, és később ennek felhasználásával történik az ebben a táblában lerakott címen kezdődő billentyűzetkezelő szubrutin meghívása. A BASIC bővítés a billentyűzet-ellenőrző táblába a 03E3H helyett más címet töltve éri el, hogy saját billentyűzetkezelő szubrutinja kerüljön meghívásra.)

A következő rövid program egészen addig, amíg megnyomunk egy billentyűt, újra és újra meghívja a KBDCSN szubrutint, majd visszatér az EDI programba:

```
 1                ;46. LISTA
 2                        ORG  7000H
 3                        LOAD 7000H
 4                KBDSCN: EQU  002BH
 5 7000 D5               PUSH DE
 6 7001 CD2B00    CIKL:  CALL KBDSCN ;ASCII KOD A REG-BE
 7 7004 B7               OR   A     ;VAN INPUT?
 8 7005 CA0170           JP   Z,CIKL ;NEM, NEZZUK UJRA
 9 7008 D1               POP  DE
10 7009 C30043           JP   4300H  ;VISSZETERES EDI-BE
11                       END
```

A program lefuttatása után a regisztertartalmak kiíratásával meggyőződhetünk arról, hogy az A regiszter valóban a megnyomott billentyű ASCII kódját tartalmazza.

A KBDSCN szubrutin két másik változatban is meghívható. A

CALL 0358H KBDSC1

azonos a KBDSCN szubrutinnal, de a fenti programhoz hasonlóan a DE regiszterpár tartalmát a veremterületen megőrzi. A

CALL 0049H GETCHR

szubrutin a DE regiszterpár tartalmát nem őrzi meg, viszont a KBDSCN ismételt meghívásával a fenti programhoz hasonlóan addig várakozik, amíg nem érkezik bemenő karakter. Megjegyezzük, hogy a fenti három, billentyűzetről olvasó szubrutin a beolvasott karaktert nem írja ki a képernyőre.

A billentyűzetről a

CALL 0361H INLINE

szubrutinnal legfeljebb 240 karakter hosszúságú adatokat olvashatunk be a

40A7H...40A8H által mutatott, alapértelmezésben a 41E8H tárcímen kezdődő pufferterületre. A karakterek beolvasása a NEW LINE megnyomásáig folytatódik, a szubrutin a pufferben 00H karakter elhelyezésével zárja a beolvasott karaktersorozatot. Ha a karakterek beolvasása közben megnyomjuk a BREAK billentyűt, akkor a visszatéréskor a C jelzőbit értéke 1 lesz. Ha a CLEAR billentyűt nyomjuk meg, akkor a szubrutin törli a képernyőt és újra kezdhetjük a karakterek beírását. A visszatéréskor a HL regiszterpár a puffer előtti címre, alapértelmezésben a 41E7H-ra mutat.

Az INLINE szubrutin felhasználásával működik a

CALL 1BB3H QINPUT

szubrutin, amely "? " kiírása után az INLINE szubrutinra ugorva várja a bemenő karaktereket.

Az INLINE, ill. QINPUT után a

CALL 1D78H RDLINE

szubrutinnal megvizsgálhatjuk, hogy numerikus karaktereket olvastunk-e be. Ez a szubrutin a HL regiszterpár által mutatott címtől indulva megkeresi az első értékes (blanktől, NEWLINE-től, vízszintes tabulálástól különböző) karaktert, a HL regiszterpárban visszaadja ennek címét, továbbá 1-re állítja a C jelzőbitet, ha numerikus karaktert talált, és 1-re állítja a Z jelzőbitet, ha 00H vagy ":" karaktert talál. Az RDLINE szubrutin az egybyte-os RST 10H utasítással is meghívható. (Megjegyezzük, hogy a BASIC értelmező ezzel a szubrutinnal elemzi a programsorokat, így a 4003H címen kezdődő RST 10H vezérlésátadási vektort felülírva megszakíthatjuk annak működését.)

Az INLINE vagy QINPUT által beolvasott numerikus karaktersorozatot a

CALL 0E6CH ASCBIN

szubrutinnal egész számmá, egyszeres vagy kétszeres pontosságú lebegőpontos számmá alakíthatjuk át. Ez a szubrutin megvizsgálja a HL regiszterpár által mutatott ASCII karaktersorozatot, és ha az egész változóként ábrázolható, akkor egész számként, ha nem, akkor az értékes jegyek számától függően egyszeres vagy kétszeres pontosságú számként az SA szoftverakkumulátorba viszi, és a típusjelzőt beállítja a megfelelő értékre.

A következő program a "? " üzenet kiírása után beolvassa a bemenő karaktereket, és ha azok numerikusak, elvégzi a fenti átalakítást:

```
 1                  ;47. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 CDB31B              CALL 1BB3H  ;BEOLVASAS
 5 7003 E5                  PUSH HL
 6 7004 D7          CIKL:   RST  10H    ;KARAKTER VIZSGALAT
 7 7005 38FD                JR   C,CIKL ;NUMERIKUS, TOVABB
 8 7007 2008                JR   NZ,HIBA;NEM NUMERIKUS
 9 7009 E1                  POP  HL
10 700A 23                  INC  HL     ;BUFFER ELEJE
11 700B CD6C0E              CALL 0E6CH  ;KONVERZIO
12 700E C3A631              JP   12710
13 7011 18FE       HIBA:    JR   HIBA
14                          END
```

A program nem numerikus adat esetén a HIBA címkére ugrik, numerikus adat esetben elvégzi a konverziót, és a BASIC bővítés monitorjába lép. Futtassuk le a programot különböző bemenő adatokkal (12345, 123.45, 12345.6789, 123E12 stb.), és írassuk ki a képernyőre a szoftverakkumulátor és a típusjelző tartalmát!

Az INLINE és QINPUT szubrutinok a beírt karaktereket a képernyőre is kiírják, tehát ezeket a szubrutinokat alkalmazva a bemenő karakterek a tár két helyére kerülnek, a 40A7H...40A8H által mutatott (alapértelmezésben a 41E8H címen kezdődő, pufferterületre és a képernyőre a 4020H...4021H által mutatott kurzorpozíciótól kezdve. A puffert a 40A7H...40A8H felülírásával a tár tetszőleges részére áthelyezhetjük. Keressünk a tárban alkalmas üres helyet, írjuk át a 40A7H...40A8H tartalmát erre a címre, futtassuk le újra a programot, és vizsgáljuk meg a futtatás után a tár tartalmát!

Képernyőre írás

A már megismert INLINE szubrutinon kívül az alábbi szubrutinokkal is írhatunk a képernyőre.

Az A regiszterben elhelyezett karaktert a 4020H...4021H kurzormutatóval meghatározott képernyő-pozícióra irhatjuk a

CALL 0033H VDCHR

szubrutinnal. Ugyanígy működik, de a veremterületre kimenti a DE regiszterpár tartalmát, a

CALL 033AH VDCHR1

szubrutin.

A következő két szubrutinnal a 409CH kimenetjelző értékétől függően bármely perifériára írhatunk. Ha erre a címre 00H-t töltünk, akkor a képernyőre, ha 01H-t töltünk, akkor a nyomtatóra, ha FFH-t töltünk, akkor a kazettára kerül az output. A

CALL 032AH OUTCHR

szubrutin az A regiszter tartalmát viszi a 409CH által meghatározott perifériára, a

CALL 28A7H OUTLIN

szubrutin pedig a HL regiszterpár által mutatott tárcímen kezdődő, 00H-val lezárt ASCII karaktersorozatot viszi a 409CH által meghatározott perifériára.

Az OUTLIN szubrutint számok kiírására is felhasználhatjuk. Az SA szoftverakkumulátorban levő, a 40AFH típusjelző által meghatározott típusú számot a

CALL 0FBDH BINASC

szubrutinnal karaktersorozattá alakíthatjuk. A karakteres formára alakított számot a szubrutin a 4130H címtől tárolja, a szám végét 00H jelzi. A szubrutinból visszatéréskor a HL regiszterpár a 4130H címre mutat.

Következő példánkban a QINPUT-tal beolvasott számot a képernyő törlése után az OUTLIN szubrutinnal újra kiírjuk:

```
 1                  ;48. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4                  CLS:    EQU  01C9H   ;KEPERNYO TORLES
 5                  QINPUT: EQU  1BB3H   ;BILLENTYUZET INPUT
 6                  OUTLIN: EQU  28A7H   ;KEPERNYORE IRAS
 7 7000 CDB31B              CALL QINPUT  ;BEOLVASAS
 8 7003 CDC901              CALL CLS     ;KEPERNYO TORLES
 9 7006 3E00                LD   A,0
10 7008 329C40              LD   (409CH),A;OUTPUT KEPERNYORE
11 700B 23                  INC  HL      ;BUFFER ELEJE
12 700C CDA728              CALL OUTLIN  ;KIIRAS
13 700F 18FE        VEGE:   JR   VEGE
14                          END
```

Megjegyezzük, hogy a CALL utasításokban csupán a programozás kényelmesebbé tétele céljából használunk címkéket, a CALL 01C9H ugyanaz, mint a CLS: EQU 01C9H utáni CALL CLS, de ha többször hívjuk ugyanazt a szubrutint, akkor jobb ez utóbbi módszert használni.

Adatkonverzió

Programjainkban gyakran van szükségünk arra, hogy adatainkat átalakítsuk, karakterekből számot, számból karaktereket, egész számból lebegőpontos számot csináljunk, a számokat formázottan írjuk ki a képernyőre stb. A ROM-rutinok között a már megismert ASCBIN és BINASC szubrutinokon kívül számos, ezt a tevékenységet támogató szubrutin található. Az alábbiakban a legfontosabb konverziós rutinokkal fogunk megismerkedni.

Az SA szoftverakkumulátorban elhelyezett tetszőleges típusú számot a

CALL 0AB1H	CSGL
CALL 0ADBH	CDBL

szubrutinokkal egyszeres, ill. kétszeres pontosságú számmá alakíthatjuk. Az SA szoftverakkumulátorban elhelyezett egyszeres vagy kétszeres pontosságú számot a

CALL 0A7FH	CINT
CALL 0B26H	FIX

szubrutinokkal egész számmá alakíthatjuk, ill. leválaszthatjuk a szám törtrészét. A FIX és INT szubrutinok túlcsorduláskor OV hibaüzenetet adnak.

Az ASCII karakteres formájú számokat a már megismert ASCBIN szubrutinon kívül két másik ROM-rutinnal is átalakíthatjuk numerikussá. A

CALL 0E65H	ASCDBL

szubrutin a HL regiszterpár által mutatott, 00H vagy ":" karakterre végződő ASCII karaktersorozatot kétszeres pontosságú számmá alakítva az SA szoftverakkumulátorba helyezi. A

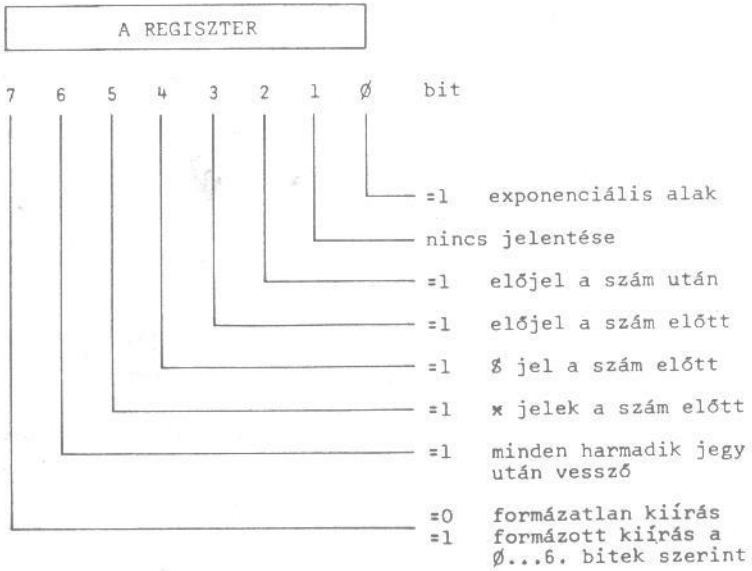
CALL 1E5AH	ASCINT

szubrutin a HL regiszterpár által mutatott, az első nem numerikus karakterig tartó karaktersorozatot 16 bites egész számmá alakítva a DE regiszterpárba tölti.

A számokat numerikusból ASCII karakteres formájúra a

CALL 0FBEH BINFOR

szubrutinnal alakíthatjuk, amely ha az A regiszterben nulla van, akkor ugyanúgy, mint a BINASC szubrutin, nem formattált kiíratást végez, ha pedig az A regiszter nem nullát tartalmaz, akkor az egyes bitek értékétől függően exponenciális, előjeles stb. kiíratást végez. Az A regiszter egyes bitjeinek jelentését a 11.4. ábra mutatja. A tizedesponttól balra álló számjegyek számát a B re-



11.4. ábra. BINFOR paraméterek

giszterben, a tizedesponttól jobbra álló számjegyek számát pedig a C regiszterben kell megadni. A szubrutin a karakterekké alakított számot a 4130H címen kezdődő pufferben helyezi el. A szubrutinból kilépéskor a HL regiszterpár erre a címre mutat.

Következő példánk a beolvasott numerikus karaktereket számmá alakítja, majd formázottan kiírja:

```
 1                    ;49. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 CDC901              CALL 01C9H   ;CLS
 5 7003 CDB31B              CALL 1BB3H   ;QIPNUT
 6 7006 23                  INC  HL
 7 7007 CD6C0E              CALL 0E6CH   ;ASCBIN
 8 700A 3EA0                LD   A,0A0H
 9 700C 0606                LD   B,06H
10 700E 0E03                LD   C,03H
11 7010 CDBE0F              CALL 0FBEH   ;BINFOR
12 7013 3E00                LD   A,0
```

```
13 7015 32C940          LD   (40C9H),A
14 7018 CDA728          CALL 28A7H  ;OUTLIN
15 701B CD4900          CALL 0049H  ;INPUTRA VAR
16 701E C30043          JP   4300H  ;EDI-BE VISSZATER
17                      END
```

Futtassuk le a programot, álljunk meg a szubrutinhívások után, és nézzük meg, mit tartalmaznak a regiszterek, a 4130H és 41E8H címeken kezdődő pufferek.

Aritmetikai műveletek

A következő ROM-rutinok egész számokkal, egyszeres és kétszeres pontosságú lebegőpontos számokkal összeadást, kivonást, szorzást, osztást és összehasonlítást végeznek.

A DE és HL regiszterpárokba helyezett egész számokkal a

CALL 0BD2H	ADDINT
CALL 0BC7H	SUBINT
CALL 0BF2H	MULINT
CALL 2490H	DIVINT

szubrutinok végzik az alapműveleteket. A szubrutinok meghívása előtt a 40AFH típusjelzőbe 2-t kell írni, visszatéréskor az összeadás, kivonás és szorzás eredményét a HL regiszterpárban és az SA szoftverakkumulátorban, túlcsordulás és osztás esetén az eredményt csak az SA szoftverakkumulátorban találjuk. A szubrutinok a típusjelzőt az eredmény típusa szerint állítják.

A DE és HL regiszterpárokba helyezett egész számokat a

CALL 0A39H	CMPINT

szubrutinnal hasonlíthatjuk össze. A szubrutin a két regiszterpár tartalmát változatlanul hagyja, az összehasonlítás eredményét visszatéréskor az A regiszterbe a következőképpen adja:

(DE) > (HL)	(A) = –1
(DE) = (HL)	(A) = 0
(DE) < (HL)	(A) = +1

A DE és HL regiszterpárokba helyezett 16 bites egész számokat a

CALL 1C90H	CPDEHL

szubrutinnal is összehasonlíthatjuk. A szubrutin a jelzőbiteket a következőképpen állítja:

Z=1	a két szám egyenlő
Z=0	a két szám különböző
C=1	(HL) < (DE)
C=0	(HL) > (DE)

A CPDEHL szubrutin az RST 18H egybyte-os szubrutinhívással is meghívható.

A következő programban a billentyűzetről beolvasunk két egész számot, kiszámoljuk az összegüket, és az eredményt kiírjuk a képernyőre:

```
 1                  ;50. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4                  QINPUT: EQU  1BB3H
 5                  ASCBIN: EQU  0E6CH
 6                  ADDINT: EQU  0BD2H
 7                  BINASC: EQU  0FBDH
 8                  CINT:   EQU  0A7FH
 9                  CLS:    EQU  01C9H
10                  OUTLIN: EQU  28A7H
11 7000 CD1E70             CALL INPUT  ;ELSO OPERANDUS
12 7003 ED5B2141           LD   DE,(4121H)
13 7007 D5                 PUSH DE
14 7008 CD1E70             CALL INPUT  ;MASODIK OPERANDUS
15 700B 2A2141             LD   HL,(4121H)
16 700E D1                 POP  DE
17 700F CDD20B             CALL ADDINT ;DE+HL
18 7012 CDBD0F             CALL BINASC ;KIIRATAS
19 7015 AF                 XOR  A
20 7016 32C940             LD   (40C9H),A
21 7019 CDA728             CALL OUTLIN
22 701C 18FE       VEGE:   JR   VEGE
23 701E CDC901     INPUT:  CALL CLS
24 7021 CDB31B             CALL QINPUT ;BEOLVASAS
25 7024 23                 INC  HL
26 7025 CD6C0E             CALL ASCBIN
27 7028 3AAF40             LD   A,(40AFH)
28 702B FE02               CP   2      ;EGESZ SZAM?
29 702D 2001               JR   NZ,HIBA;NEM, HIBA
30 702F C9                 RET
31 7030 C37F0A     HIBA:   JP   CINT   ;EGESSZE ALAKIT
32                         END
```

Az összeadást egyszerűbb és gyorsabb lenne az ADD HL,DE egybyte-os utasítással elvégezni, de az ADDINT szubrutin többet nyújt: túlcsordulás esetén automatikusan egyszeres pontosságú számokra vált. Ha biztosak vagyunk abban, hogy az összeg egész számként ábrázolható, akkor használjuk az ADD HL,DE utasítást, ha nem, akkor jobban járunk, ha meghívjuk az ADDINT szubrutint.

Az egyszeres pontosságú lebegőpontos számokkal aritmetikai műveleteket végző

CALL 0716H	ADDSGL
CALL 0713H	SUBSGL
CALL 0847H	MULSGL
CALL 08A2H	DIVSGL

szubrutinok az első operandust a BCDE regiszternégyesben, a második operandust az SA szoftverakkumulátorban várják, az összeadás, kivonás, szorzás és osztás eredményét az SA szoftverakkumulátorba írják.

A BCDE regiszternégyes és az SA szoftverakkumulátor által tartalmazott egyszeres pontosságú számokat a

CALL 0A0CH	CMPSGL

szubrutinnal hasonlíthatjuk össze. Az összehasonlítás eredményét az A regiszter mutatja:

(BCDE) > (SA)	(A) = –1
(BCDE) = (SA)	(A) = 0
(BCDE) < (SA)	(A) = +1

Az egyszeres pontosságú aritmetikai szubrutinokra példaként a billentyűzetről beolvasunk két egyszeres pontosságú számot, kivonást végzünk, és az eredményt kiírjuk a képernyőre:

```
 1                  ;51. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 CD1C70              CALL INPUT   ;KISEBBITENDO
 5 7003 CDBF09              CALL 09BFH   ;LDSARA
 6 7006 C5                  PUSH BC
 7 7007 D5                  PUSH DE
 8 7008 CD1C70              CALL INPUT   ;KIVONANDO
 9 700B D1                  POP  DE
10 700C C1                  POP  BC
11 700D CD1307              CALL 0713H   ;SUBSGL
12 7010 CDBD0F              CALL 0FBDH   ;BINASC
13 7013 AF                  XOR  A
14 7014 32C940              LD   (40C9H),A
15 7017 CDA728              CALL 28A7H   ;OUTLIN
16 701A 18FE       VEGE:    JR   VEGE
17 701C CDC901     INPUT:   CALL 01C9H   ;CLS
18 701F CDB31B              CALL 1BB3H   ;QINPUT
19 7022 23                  INC  HL
20 7023 CD6C0E              CALL 0E6CH   ;ASCBIN
21 7026 C3B10A              JP   0AB1H   ;CSGL
22                          END
```

A kétszeres pontosságú lebegőpontos számokkal aritmetikai műveleteket végző

CALL 0C77H	ADDDBL
CALL 0C70H	SUBDBL
CALL 0DA1H	MULDBL
CALL 0DE5H	DIVDBL

szubrutinok az első operandust az SA szoftverakkumulátorban, a második operandust az SA1 másodlagos szoftverakkumulátorban várják, az összeadás, kivonás, szorzás és osztás eredményét az SA szoftverakkumulátorba írják.

Az SA szoftverakkumulátorba és az SA1 másodlagos szoftverakkumulátorba helyezett kétszeres pontosságú számokat a

CALL 0A78H	CMPDBL

szubrutinnal hasonlíthatjuk össze. Az összehasonlítás eredményét az A regiszter mutatja:

(SA) > (SA1)	(A) = −1
(SA) = (SA1)	(A) = 0
(SA) < (AS1)	(A) = +1

A kétszeres pontosságú aritmetikai szubrutinokra példaként a billentyűzetről beolvasott kétszeres pontosságú számokat összehasonlítjuk, és az A regiszterből kiolvashatjuk az összehasonlítás eredményeképpen kapott −1, 0 vagy +1 értéket:

```
 1                  ;52. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 CD2770              CALL INPUT
 5 7003 CDEC0A              CALL 0AECH   ;SETDBL
 6 7006 211D41              LD   HL,411DH;SA CIME
 7 7009 113470              LD   DE,BUFF;BUFF CIME
```

```
 8 700C CDD209          CALL 09D2H  ;MVHLDE
 9 700F CD2770          CALL INPUT
10 7012 CDEC0A          CALL 0AECH  ;SETDBL
11 7015 CDFC09          CALL 09FCH  ;MVSA
12 7018 213470          LD   HL,BUFF
13 701B 111D41          LD   DE,411DH
14 701E CDD209          CALL 09D2H  ;ELSO ADAT SA-BAN
15 7021 CD780A          CALL 0A78H  ;CMPDBL
16 7024 C3A631          JP   12710  ;MONITORBA UGRAS
17 7027 CDC901   INPUT: CALL 01C9H  ;CLS
18 702A CDB31B          CALL 1BB3H  ;QINPUT
19 702D 23              INC  HL
20 702E CD6C0E          CALL 0E6CH  ;ASCBIN
21 7031 C3DB0A          JP   0ADBH  ;CDBL
22               BUFF:  DS   8
23                      END
```

Matematikai függvények

A ROM-rutinok függvényei között trigonometrikus, exponenciális, logaritmikus stb. függvényeket és véletlenszám-generátort találunk.

A trigonometrikus függvények közül a $\sin(x)$ helyettesítési értékeinek kiszámítása a

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!}$$

közelítés alapján legfeljebb 0.000003 hibával, a $\cos(x)$ és $\mathrm{tg}(x)$ kiszámítása a

$$\cos(x) = \sin(x) + \frac{\pi}{2} ,$$

$$\mathrm{tg}(x) = \frac{\sin(x)}{\cos(x)}$$

formulák alkalmazásával, az $\mathrm{arctg}(x)$ kiszámítása pedig az

$$\mathrm{arctg}(x) = -\frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \frac{x^{11}}{11} + \frac{x^{13}}{13} - \frac{x^{15}}{15} - \frac{x^{17}}{17}$$

közelítés alapján legfeljebb 0.026 hibával történik. A trigonometrikus szubrutinok radiánokban számolnak (1 radián = 57.29577951°). A

CALL 1547H	SIN
CALL 1541H	COS
CALL 15A8H	TAN
CALL 15BDH	ATN

trigonometrikus szubrutinok az argumentumot egyszeres pontosságú számként az SA szoftverakkumulátorban várják, a függvény értékét egyszeres pontossággal az argumentum helyén adják vissza.

A következő példában a fokokban megadott szöget átszámoljuk radiánokba, majd kiszámítjuk és kiírjuk a szög szinuszát:

```
 1                    ;53. LISTA
 2                           ORG  7000H
 3                           LOAD 7000H
 4 7000 CDC901               CALL 01C9H  ;CLS
 5 7003 CDB31B               CALL 1BB3H  ;QINPUT
 6 7006 23                   INC  HL
 7 7007 CD6C0E               CALL 0E6CH  ;ASCBIN
 8 700A CDDB0A               CALL 0ADBH  ;CDBL
 9 700D CDEC0A               CALL 0AECH  ;SETDBL
10 7010 2A2E70               LD   HL,(RADIAN)
11 7013 112741               LD   DE,4127H
12 7016 CDD209               CALL 09D2H  ;MVHLDE
13 7019 CDE50D               CALL 0DE5H  ;DIVDBL
14 701C CD1B0A               CALL 0A1BH  ;CSGL
15 701F CD4715               CALL 1547H  ;SIN
16 7022 CDBD0F               CALL 0FBDH  ;BINASC
17 7025 AF                   XOR  A
18 7026 32C940               LD   (40C9H),A
19 7029 CDA728               CALL 28A7H  ;OUTLIN
20 702C 18FE          VEGE:  JR   VEGE
21 702E 8D1BE9D2 RADIAN:     DB   8DH,1BH,0E9H,0D2H
22 7032 E02E6586             DB   0E0H,2EH,65H,86H
23                    ;                   57.29577951 KETSZE-
24                    ;                   RES PONTOSSAGGAL
25                           END
```

A természetes alapú logaritmus értékek kiszámítása az

$$\ln(x) = 2\left[\left(\frac{x-1}{x+1}\right) + \frac{1}{3}\left(\frac{x-1}{x+1}\right)^3 + \frac{1}{5}\left(\frac{x-1}{x+1}\right)^5 + \ldots\right]$$

közelítés alapján, az e^x függvény helyettesítési értékeinek kiszámítása az

$$e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \frac{x^5}{5!} + \frac{x^6}{6!} - \frac{x^7}{7!}$$

közelítés alapján történik. Az x^y függvény helyettesítési értékeinek kiszámítása az előző két függvény felhasználásával, a $\sqrt{x}$ értékek kiszámítása az x^y függvény alkalmazásával történik. A

CALL 0809H	LOG
CALL 1439H	EXP
CALL 13E7H	SQR

szubrutinok az argumentumot egyszeres pontosságú számként az SA szoftverakkumulátorban várják, a függvény értékét az argumentum helyén egyszeres pontossággal adják vissza. A

CALL 13F7H	X ↑ Y

szubrutin az X értéket egyszeres pontossággal a BCDE regiszternégyesben, az Y értéket egyszeres pontossággal az SA szoftverakkumulátorban várja, a hatványozás eredményét egyszeres pontossággal az SA szoftverakkumulátorban adja vissza.

A tízes alapú logaritmusértékek kiszámítására nincs külön szubrutin, de a természetes logaritmus és a

$$\log_{10}(x) = \frac{\log_e(x)}{\log_e(10)}$$

formula alapján kiszámíthatjuk azokat ($\log_e(10) = 2.302585093$), az $\sqrt[n]{x}$ függvények helyettesítési értékeit pedig az X ↑ Y függvény segítségével határozhatjuk meg.

Az SA szoftverakkumulátorben levő egész szám, egyszeres vagy kétszeres pontosságú lebegőpontos szám abszolút értékét a

CALL 0977H ABS

szubrutinnal számíthatjuk ki. A szubrutin a művelet eredményét az argumentumként megadott számával megegyező típusú számként az SA szoftverakkumulátorba helyezi.

Az SA szoftverakkumulátorban levő tetszőleges típusú szám előjelét a

CALL 098AH SGN

szubrutinnal határozhatjuk meg, amely az előjel értékétől függően a +1, 0 vagy −1 egész számot helyezi az SA szoftverakkumulátorba.

Az INT és FIX függvényekkel korábban, a konverziós rutinok ismertetésekor már foglalkoztunk.

Következő példánkban kiszámoljuk és kiírjuk a billentyűzetről beolvasott szám tízes alapú logaritmusát:

```
 1                  ;54. LISTA
 2                          ORG  7000H
 3                          LOAD 7000H
 4 7000 CDC901              CALL 01C9H  ;CLS
 5 7003 CDB31B              CALL 1BB3H  ;QINPUT
 6 7006 23                  INC  HL
 7 7007 CD6C0E              CALL 0E6CH  ;ASCBIN
 8 700A CD7709              CALL 0977H  ;ABS
 9 700D CDB10A              CALL 0AB1H  ;CSGL
10 7010 CD0908              CALL 0809H  ;LOG
11 7013 CDEF0A              CALL 0AEFH  ;SETSGL
12 7016 CDBF09              CALL 09BFH  ;LDSARA
13 7019 C5                  PUSH BC
14 701A D5                  PUSH DE
15 701B 213570              LD   HL,LOG10
16 701E 112141              LD   DE,4121H
17 7021 CDD209              CALL 09D2H  ;MVHLDE
18 7024 D1                  POP  DE
19 7025 C1                  POP  BC
20 7026 CDA208              CALL 08A2H  ;DIVSGL
21 7029 CDBD0F              CALL 0FBDH  ;BINASC
22 702C AF                  XOR  A
23 702D 32C940              LD   (40C9H),A
24 7030 CDA728              CALL 28A7H  ;OUTLIN
25 7033 18FE       VEGE:    JR   VEGE
26 7035 8E5D1382   LOG10:   DB   8EH,5DH,13H,82H
27                 ;                   LN 10 EGYSZERES
28                          END
```

A véletlenszámot generáló

CALL 14C9H RND

szubrutin az argumentum értékétől függően helyez véletlenszámot az SA szoftverakkumulátorba. Ha argumentumként az SA szoftverakkumulátorba 0-t írunk, akkor 0 és 1 közötti egyszeres pontosságú véletlenszámot kapunk.

Ha az argumentum 0-tól különböző X pozitív szám, akkor 1 és INT(X) közötti egész számot kapunk.

A véletlenszám-generátornak a

CALL 01D3H RANDOM

szubrutinnal adhatunk kezdőértéket. Ez a szubrutin kezdőértékként az R frissítőregiszter tartalmát a 40ABH címre, az RND munkaterületére másolva biztosítja, hogy más-más véletlenszám-sorozatok kerüljenek előállításra.

BASIC szubrutinok

A következő néhány ROM-rutin olyan tevékenységeket végez, amelyek a BASIC munkaterületével, táblázataival kapcsolatosak, ezért ezek csak a BASIC értelmező felügyelete alatt, tehát BASIC programból meghívott assembler szubrutinban alkalmazhatók.

A BASIC program szövegében a DE regiszterpárban megadott sorszámot a

CALL 1B2CH SLINE

szubrutinnal kereshetjük. Ha a szubrutin megtalálta a keresett számú sort, akkor a C és Z jelzőbiteket 1-re állítja, és a sor kezdőcímét a BC regiszterpárba, a következő sor címét a HL regiszterpárba töltve tér vissza. Ha nincs ilyen számú sor, mert ennél csak kisebb sorszámokat tartalmaz a program, akkor C=0, Z=1, HL és BC egyaránt a programzáró byte-okra mutatnak (a keresés a program végéig tart). Ha ilyen sorszám nem, de ennél nagyobb szerepel, akkor BC a keresett sorszámnál nagyobb sorszámú sorok közül a legkisebb sorszámú sorra mutat, HL pedig az ezt követő sor címét tartalmazza. A Z és C jelzőbitek mindegyike 0 lesz.

A BASIC változókat kereső

CALL 260DH VARPTR

szubrutin a HL regiszterpár által mutatott címen lerakott nevű változót keresi. Végignézi a megfelelő változótáblát, és ha abban megtalálja a keresett változót, akkor visszaadja a címét, ha nem találja meg, akkor a táblába nulla kezdőértékkel új változóként felveszi.

Assembly programból a BASIC főprogram szubrutinjait a

CALL 1EB1H GOSUB

ROM-rutinnal hívhatjuk meg, amely belépéskor a HL regiszterpár által mutatott tárcímen karakteres formában lerakott BASIC sorszámot vár.

A BASIC program szubrutinjából hívott gépi kódú szubrutinból közvetlenül a BASIC program GOSUB utáni utasításához térhetünk vissza a

JP 1EDFH RETURN

utasítással, amelynek kiadása előtt POP utasítással törölnünk kell a veremterületről a gépi kódú szubrutin visszatérési címét.

A BASIC programsorok nyomkövetését bekapcsolhatjuk, ill. kikapcsolhatjuk a

CALL 1DF7H	TRON
CALL 1DF8H	TROFF

szubrutinokkal. A BASIC program számára rendelkezésre álló szabad tár mennyiségét a

CALL 27C9H	MEM

szubrutin egyszeres pontosságú számként az SA szoftverakkumulátorba helyezi.

Végül megjegyezzük, hogy ebben a fejezetben csupán a legfontosabb ROM-rutinokat ismertettük. Néhány, ki- és bemenettel kapcsolatos ROM-rutinnal a következő fejezetben foglalkozunk, a Függelék ROM-táblázatában pedig több, mint 500 ROM-rutin belépési pontját és rövid leírását közöljük.

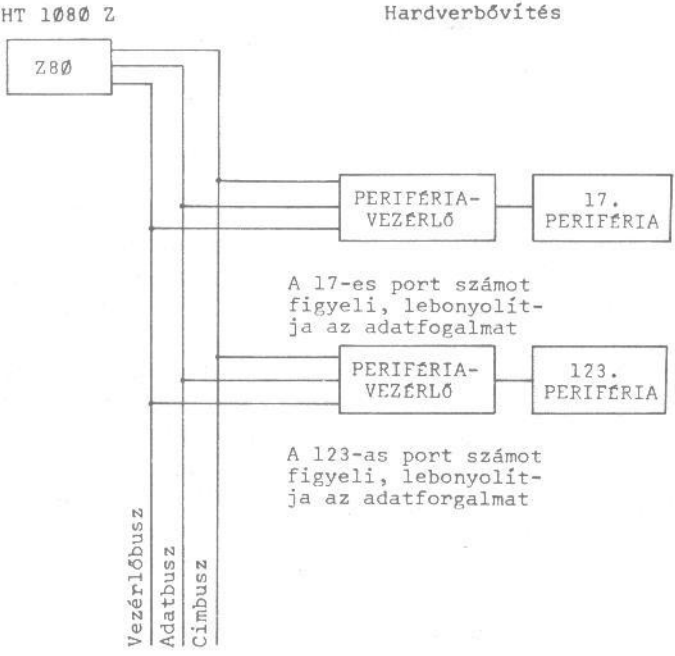
12. fejezet

Be- és kimeneti műveletek

Az előző fejezetekben több helyen foglalkoztunk a speciális tárcímeken keresztül történő adatátvitellel, a billentyűzetmátrixról olvasással és a képernyőre írással. Ebben a fejezetben a portcímeken keresztül történő be- és kimeneti műveletekkel és a HT-1080Z kazetta kezelésével fogunk megismerkedni.

Be- és kimeneti portok

A port-utasításokban egybyte-os portcímet kell megadnunk, amelyet a Z80 mikroprocesszor a címbusz alsó felére helyez, és a vezérlőbuszon jelzi, hogy a címbuszon most nem tárcím, hanem portcím van. A vezérlőbuszon küldött jelek alapján a számítógéphez hardverbővítésként csatlakozó perifériavezérlő egység vagy egységek érzékelik, hogy be- és kimeneti műveletet kell lebonyo-



12.1. ábra. Input/output portok

lítani, megkeresik a portcím által meghatározott perifériát és lebonyolítják az adatátvitelt.

A bemeneti port-utasítások közül legfontosabb a közvetlen címzésű

IN A,(pp)	DB pp

utasítás, amelynek első byte-ja az utasításkód, második byte-ja a 00H és FFH közötti portcím. Az IN A,(pp) utasítás a pp portcímű perifériáról az A regiszterbe visz egy byte-ot. Ez az utasítás a jelzőbiteket nem állítja, ezért a beolvasott adat vizsgálatához alkalmas utasítással (AND A vagy OR A) be kell állítani a jelzőbiteket.

Azt, hogy az egyes perifériák milyen portcímeken érhetők el, az adott számítógép hardverfelépítése határozza meg. Pl. a HT-1080Z számítógép beépített magnója az FFH portcímen található.

A bemeneti port címét a C regiszterbe helyezve az

IN A,(C)	ED 78
IN B,(C)	ED 40
IN C,(C)	ED 48
IN D,(C)	ED 50
IN E,(C)	ED 58
IN H,(C)	ED 60
IN L,(C)	ED 68

utasításokkal bármely általános regiszterbe beolvashatunk egy byte-ot. Ezek az utasítások az S, Z és P/V jelzőbiteket a beolvasott byte tartalma szerint állítják (P/V itt paritást jelez), a H és N jelzőbiteket törlik, a C jelzőbitet változatlanul hagyják.

A blokkmozgatás és blokk-összehasonlítás utasításokhoz hasonló

INI	ED A2
INIR	ED B2

blokkos bemeneti utasítások a C regiszterrel meghatározott portról érkező adatot a HL regiszterpár által mutatott tárcímre viszik, majd eggyel növelik a HL regiszterpár tartalmát, és eggyel csökkentik a számlálóként szolgáló B regiszter tartalmát. Az N jelzőbitet 1-re állítják, a C jelzőbitet változatlanul hagyják, a Z jelzőbitet 1-re állítják, ha a B regiszter tartalma nullára csökken, a többi jelzőbitet előre nem látható módon változtatják. Az INI utasítás után a vezérlés a következő utasításra kerül, ami lehetőséget ad a bemenet byte-onkénti feldolgozására. Az INIR utasítás a PC programszámláló értékét 2-vel csökkentve automatikusan addig ismétli önmagát, amíg a B regiszter tartalma el nem éri a nullát, így egyetlen INIR utasítással adott, legfeljebb 256 byte hosszúságú karaktersorozatot olvashatunk be a C regiszterrel meghatározott perifériáról a HL regiszterpár által mutatott tárterületre. Az

IND	ED AA
INDR	ED BA

utasítások a HL regiszterpár tartalmát mindig eggyel csökkentik, egyébként megegyeznek az előbb ismertetett INI és INIR utasításokkal.

Az A regiszter tartalmát a pp port-című perifériára az

OUT (pp),A	D3 pp

utasítással vihetjük át. Az utasítás első byte-ja az utasításkód, második byte-ja 00H és FFH közötti portcím.

A kimeneti port címét a C regiszterbe helyezve az

OUT (C),A	ED 79
OUT (C),B	ED 41
OUT (C),C	ED 49
OUT (C),D	ED 51
OUT (C),E	ED 59
OUT (C),H	ED 61
OUT (C),L	ED 69

utasításokkal bármely általános regiszter tartalmát kivihetjük a C regiszter tartalma által meghatározott perifériára. Az OUT utasítások a jelzőbiteket változatlanul hagyják.

A blokkos kimeneti utasítások közül az

OUTI	ED A3
OTIR	ED B3

utasítások a HL regiszterpár által mutatott tárcím tartalmát a C regiszterrel meghatározott portra viszik, majd eggyel növelik a HL regiszterpár tartalmát, és eggyel csökkentik a számlálóként szolgáló B regiszter tartalmát, a jelzőbiteket ugyanúgy állítják, mint az INI és INIR utasítások. Az OUTI utasítás után a vezérlés a következő utasításra kerül, ami lehetőséget ad a byte-ok egyenkénti kivitelére. Az OTIR utasítás a PC programszámláló értékét 2-vel csökkentve automatikusan ismétli önmagát, amíg a B regiszter tartalma el nem éri a nullát, így egyetlen OTIR utasítással adott, legfeljebb 256 byte hosszúságú karaktersorozatot vihetünk ki a HL regiszterpár által mutatott tárterületről a C regiszterrel meghatározott perifériára. Az

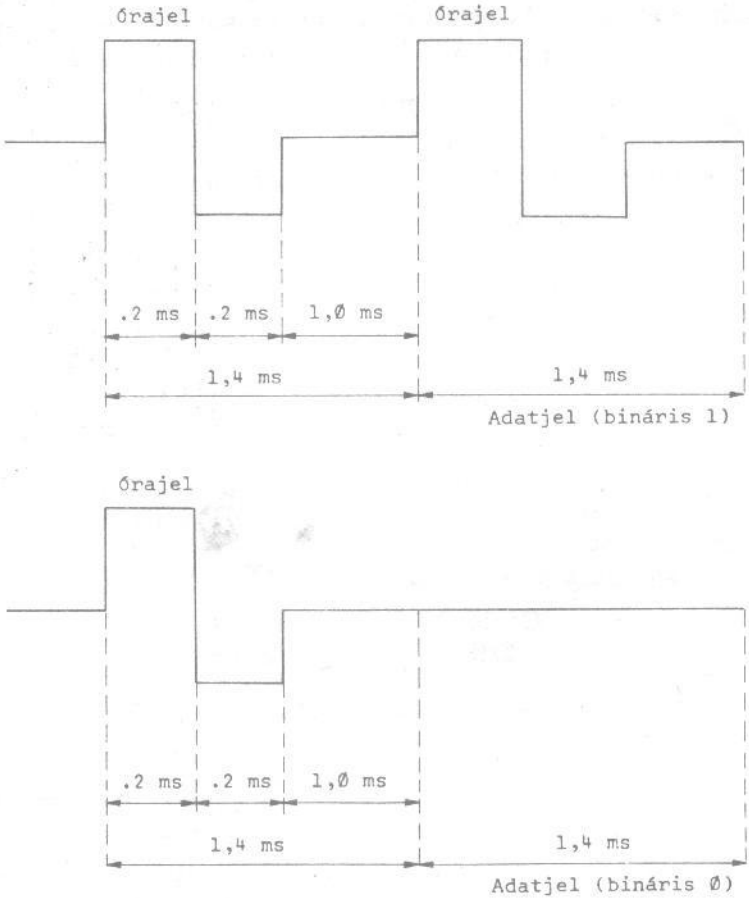
OUTD	ED AB
OTDR	ED BB

utasítások a HL regiszterpár tartalmát mindig eggyel csökkentik, egyébként megegyeznek az OUTI és OTIR utasításokkal.

Megjegyezzük, hogy a gépi kódú utasítások ugyanazon portokat azonosítják, mint a megfelelő BASIC utasítások. Az alapkiépítés portjairól a kézikönyvekből kaphatunk információkat.

Kazettaműveletek

A kazetta bemenet/kimenet több szempontból is eltér a többi periféria kezelésétől. A kazettára bitenként kerülnek az adatok, míg IN vagy OUT utasítással lebonyolított bemenet/kimenet művelet teljes byte-ot mozgat. Minden bit rög-

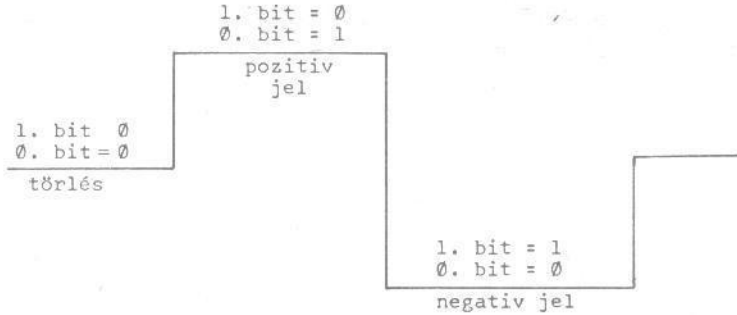


12.2. ábra. Adatformátum a kazettán

zítése órajellel* és rögzített hosszúságú törölt szalaggal kezdődik, amelyet bináris 1 kiírásakor második órajel, bináris 0 kiírásakor ugyanolyan időtartamú törölt szalag követ:
A jelek rögzített időtartama másodpercenként 500 bit átvitelét teszi lehetővé.

A kazettára OUT utasítással küldött minden egybyte-os adat egy bit felírásához ad információt. Az adat felső bitjeinek nincs jelentése, a 2. bit a magnó motorját kapcsolja, az értékes információt az 1. és 0. bitek szállítják. Ha 2. bitként 1-et küldünk ki, akkor egy kis relé segítségével bekapcsolódik a magnó motorja, ha 0-t küldünk ki, akkor a motor kikapcsolódik. Az 1. és 0. bit a négyszögjel három részét állítja elő:

* Az órajel közvetlenül a rendszer oszcillátorjele, ez a mikroprocesszor tovább már nem osztható időegysége.



12.3. ábra. Kazettajelek formátuma

A kazettaegység kiválasztása az FEH portra írt 00H vagy FFH értékkel történik. Ha a beépített magnóra kívánunk írni, vagy arról kívánunk olvasni, akkor 00H-t kell erre a portra küldeni. A magnót az FFH portcímen érhetjük el, minden OUT (0FFH), A utasítással erre írhatunk, minden IN A,(0FFH) utasítással erről olvashatunk egy bitet.

A kazettaegységre az A regiszterből egy byte-ot pl. a következő szubrutinnal írhatunk ki:

```
 1                  ;55. LISTA
 2                          ORG  7E00H
 3                          LOAD 7E00H
 4 7E00 F5                  PUSH AF
 5 7E01 C5                  PUSH BC
 6 7E02 D5                  PUSH DE
 7 7E03 E5                  PUSH HL
 8 7E04 2E08                LD   L,8      ;BITEK SZAMA
 9 7E06 67                  LD   H,A      ;ADATBYTE A-BAN
10 7E07 CD207E    CIKL:    CALL ORA      ;ORAJEL IRASA
11 7E0A 7C                 LD   A,H      ;ADATBYTE A-BAN
12 7E0B 07                 RLCA          ;FELSO BIT C-BEN
13 7E0C 67                 LD   H,A      ;MEGORZES
14 7E0D 300B               JR   NC,URES;NULLA, URES SZALAG
15 7E0F CD207E             CALL ORA      ;EGY, ORAJEL IRASA
16 7E12 2D        VEGE:    DEC  L        ;KESZ?
17 7E13 20F2               JR   NZ,CIKL;NEM, KOVETKEZO
18 7E15 E1                 POP  HL       ;IGEN, VISSZATERES
19 7E16 D1                 POP  DE
20 7E17 C1                 POP  BC
21 7E18 F1                 POP  AF
22 7E19 C9                 RET
23 7E1A 0687      URES:    LD   B,135    ;0.988 MSEC
24 7E1C 10FE      URES1:   DJNZ URES1    ;URES SZALAG IRASA
25 7E1E 18F2               JR   VEGE     ;KOVETKEZO BIT
26 7E20 3E05      ORA:     LD   A,5      ;POZITIV JEL, MOTOR
27 7E22 D3FF               OUT  (0FFH),A;ORAJEL KEZDETE
28 7E24 0639               LD   B,57     ;0.417 MSEC
29 7E26 10FE      ORA1:    DJNZ ORA1
30 7E28 3E06               LD   A,6      ;NEGATIV JEL, MOTOR
31 7E2A D3FF               OUT  (0FFH),A;ORAJEL FOLYTATASA
32 7E2C 0639               LD   B,57     ;0.417 MSEC
33 7E2E 10FE      ORA2:    DJNZ ORA2
34 7E30 3E04               LD   A,4      ;TORLES ES MOTOR
35 7E32 D3FF               OUT  (0FFH),A;URES SZALAG
36 7E34 0688               LD   B,136    ;0.995 MSEC
37 7E36 10FE      ORA3:    DJNZ ORA3
38 7E38 C9                 RET
39                         END
```

Az előző szubrutinnal készített szalagról az A regiszterbe egy byte-ot a következő szubrutinnal olvashatunk be:

```
 1                    ;56. LISTA
 2                            ORG  7E00H
 3                            LOAD 7E00H
 4 7E00 AF                    XOR  A       ;CELREG-BE NULLA
 5 7E01 C5                    PUSH BC
 6 7E02 D5                    PUSH DE
 7 7E03 E5                    PUSH HL
 8 7E04 0608          CIKL:   LD   B,8     ;BITEK SZAMA
 9 7E06 CD0F7E                CALL OLV     ;KOVETKEZO BIT
10 7E09 E1                    POP  HL
11 7E0A 10F8                  DJNZ CIKL    ;8-SZOR OLVASAS
12 7E0C D1                    POP  DE
13 7E0D C1                    POP  BC
14 7E0E C9                    RET          ;VISSZATERES
15 7E0F C5           OLV:     PUSH BC
16 7E10 F5                    PUSH AF
17 7E11 DBFF         OLV1     IN   A,(0FFH);ATMENETI TART
18 7E13 17                    RLA          ;OLVAS, URES?
19 7E14 30FB                  JR   NC,OLV1;IGEN, TOVABB KE-
20                   ;                      RES, HA NEM, ORA-
21                   ;                      JEL ELEJET FELTE-
22                   ;                      TELEZI
23 7E16 0639                  LD   B,57    ;VARAKOZAS
24 7E18 10FE         OLV2:    DJNZ OLV2    ;POZITIV JEL
25 7E1A 3E04                  LD   A,4     ;ATMENETI TART
26 7E1C D3FF                  OUT  (0FFH),A;TORLI
27 7E1E 06C1                  LD   B,193   ;VARAKOZAS AZ
28 7E20 10FE         OLV3:    DJNZ OLV3    ;ORAJEL VEGEIG
29 7E22 DBFF                  IN   A,(0FFH);ADATJEL ELEJE
30                   ;                    ORAJEL OLVASAS
31 7E24 47                    LD   B,A     ;JEL MEGORZESE
32 7E25 F1                    POP  AF      ;EDDIGI BITEK
33 7E26 CB10                  RL   B       ;C=0 URES SZALAG
34                   ;                      C=1 NEM URES
35 7E28 17                    RLA          ;UJ BIT A-BA
36 7E29 F5                    PUSH AF      ;UJ BIT MEGORZES
37 7E2A 3E04                  LD   A,4     ;ATMENETI TART
38 7E2C D3FF                  OUT  (0FFH),A;TORLI
39 7E2E 06F0                  LD   B,240   ;ADATJEL VEGE?
40 7E30 10FE         OLV4:    DJNZ OLV4
41 7E32 C1                    POP  BC
42 7E33 F1                    POP  AF      ;ADATBYTE A-BAN
43 7E34 C9                    RET
44                            END
```

A kazettával kapcsolatos tevékenységet ROM-rutinokkal is elvégezhetjük. A kazettaegység kiválasztására és a motor elindítására a

CALL 0212H DEFCAS

szubrutint hívhatjuk meg, amely ha az A regiszterben 0 van, akkor a beépített magnót, ha az A regiszterben FFH van, akkor a külső magnót választja ki és kapcsolja be. Az A regiszterben levő byte-ot a

CALL 0264H WRBYTE

szubrutin meghívásával írhatjuk ki, a kazettáról egy byte-ot az A regiszterbe a

CALL 0235H RDBYTE

szubrutinnal, két byte-ot (címet) a HL regiszterpárba a

CALL 0314H RDADDR

szubrutinnal olvashatunk be. A magnó motorját a

CALL 01F8H CASOFF

szubrutinnal kapcsolhatjuk ki.

Adatszalag írása és olvasása

A BASIC értelmező a kazettára minden adathalmaz, BASIC program és SYSTEM program (gépi kódú program) elejére fejlécet ír, amely 256 byte nullát és egy A5H szinkron byte-ot tartalmaz.

A fejlécet és az A5H szinkron byte-ot a

CALL 0287H WRHDR

szubrutinnal írhatjuk ki. Szalag olvasásakor használjuk a

CALL 0296H RDHDR

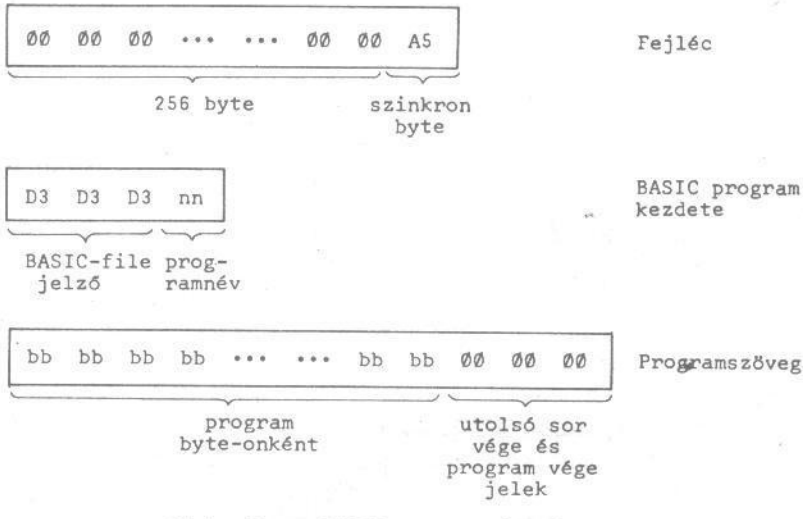
szubrutint, amely a kiválasztott kazettaegységről addig olvas, amíg fejléc végéhez (A5H szinkron byte-hoz) nem érkezik. Ha ezt megtalálja, akkor a képernyő jobb felső sarkába kiír két csillagot. A jobb oldali csillag a

CALL 022CH BLSTAR

szubrutin ismételt meghívásával villogtatható.

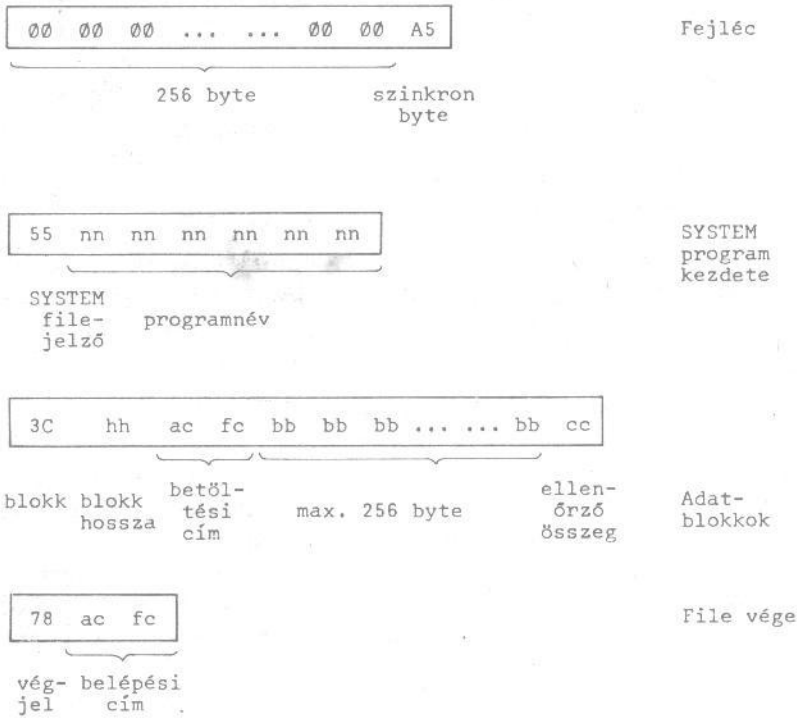
Adatok kiírásakor a fejléc után kötetlen formában írhatunk a szalagra, magunk határozhatjuk meg, hogy milyen alakú és milyen hosszúságú jeleket írunk, de ügyelnünk kell arra, hogy a kiírás és a beolvasás összhangban legyen.

BASIC program kiírásakor a BASIC értelmező a fejléc után három byte D3H-t ír, amit a CSAVE parancsban megadott egybyte-os programnév követ. Ezután a BASIC program első byte-jától az utolsóig byte-onként kiírásra kerül a program. A file végét három nulla (utolsó sor vége és program vége) jelzi.



12.4. ábra. BASIC program formátuma

SYSTEM program esetén a fejlécet egy 55H byte, majd hat byte-on a programnév követi. Ezután legfeljebb 256 byte hosszúságú blokkokban a gépi kódú program byte-jai következnek. Minden blokk elejét egy 3CH byte jelzi, amelyet egy byte-on a blokk hossza 256=00H, majd két byte-on a betöltési cím követ, majd a megadott számú adatbyte után egybyte-os ellenőrző összeg zárja a blokkot. Ezután további hasonló blokkok következhetnek, végül az utolsó adatblokk után egy 78H byte és két byte-on a program belépési címe zárja a SYSTEM file-t.



12.5. ábra. SYSTEM program formátuma

A jobb oldali csillagot BASIC program beolvasásakor (CLOAD parancs) minden sor, gépi kódú program beolvasásakor (SYSTEM parancs) minden blokk után villogtatja a BLSTAR szubrutin.

A kazettakezelő ROM-rutinok alkalmazására első példaként egy olyan programot mutatunk, amely először fejlécet keres a szalagon, majd a szalag további részének tartalmát byte-onként olvasva a tárba tölti. A fejléc beolvasásának végét a képernyő jobb felső sarkában megjelenő két csillag jelzi, a további byte-okat a program a 6000H tárcímtől helyezi el. Minthogy előre nem lehet tudni, mi van a szalagon, ezért a program a végtelenségig olvas. Ezt persze nem kell kivárnunk, egy idő után, ha már üresnek látjuk a szalagot, a RESET gomb megnyomásával leállíthatjuk az olvasást. Ezután a tárat kiíratva megvizsgálhajtuk, hogy mi van a szalagon. Pl. ha gépi kódú program szalagját olvassuk ezzel a módszerrel, akkor a 6000H címen a SYSTEM prog-

ramjelzőt (55H), a következő hat címen a program nevét találjuk, majd a 12.5. ábrán látható formában következnek a program blokkjai.

```
 1                    ;57. LISTA
 2                            ORG  7E00H
 3                            LOAD 7E00H
 4 7E00 AF                    XOR  A       ;BELSO MAGNO
 5 7E01 CD1202                CALL 0212H
 6 7E04 CD9602                CALL 0296H   ;FEJLEC OLVASAS
 7 7E07 210060                LD   HL,6000H;BETOLTESI CIM
 8 7E0A CD3502        OLVAS:  CALL 0235H   ;BYTE OLVASASA
 9 7E0D 77                    LD   (HL),A  ;ES TAROLASA
10 7E0E 23                    INC  HL      ;KOVETKEZO CIM
11 7E0F 18F9                  JR   OLVAS   ;KOVETKEZO BYTE
12                            END
```

Második példánkban a tár adott részének tartalmát a következő formában írjuk ki kazettára:

fejléc	(256 byte nulla és A5H szinkronbyte)
file hossza	(alsó és felső fele 2 byte-on)
kezdőcím	(alsó és felső fele 2 byte-on)
adatbyte-ok	

A képernyő törlése után megjelenő első "?" üzenetre a kimentendő tárrész első byte-jának címét, a második "?" üzenetre az utolsó byte címét kell megadni. A program ezután elindítja a magnót, kiviszi az adatokat, kikapcsolja a magnót, majd a VEGE címkénél várakozik.

```
 1                    ;58. LISTA
 2                            ORG  7000H
 3                            LOAD 7000H
 4 7000 CDC901                CALL 01C9H   ;CLS
 5 7003 CD3870                CALL CIM     ;KEZDOCIM BEOLVASAS
 6 7006 E5                    PUSH HL      ;ES MEGORZESE
 7 7007 CD3870                CALL CIM     ;VEGCIM BEOLVASAS
 8 700A D1                    POP  DE      ;KEZDOCIM DE-BEN
 9 700B AF                    XOR  A       ;C JELZOBIT TORLESE
10 700C ED52                  SBC  HL,DE   ;HOSSZ-1=HL-DE
11 700E 23                    INC  HL      ;HOSSZ
12 700F D5                    PUSH DE      ;KEZDOCIM STACKBEN
13 7010 E5                    PUSH HL      ;HOSSZ STACKBEN
14 7011 CD1202                CALL 0212H   ;DEFCAS
15 7014 CD8702                CALL 0287H   ;WRHDR
16 7017 D1                    POP  DE      ;HOSSZ DE-BEN
17 7018 7B                    LD   A,E     ;HOSSZ ALSO BYTE
18 7019 CD6402                CALL 0264H   ;WRBYTE
19 701C 7A                    LD   A,D     ;HOSSZ FELSO BYTE
20 701D CD6402                CALL 0264H   ;WRBYTE
21 7020 E1                    POP  HL      ;KEZDOCIM HL-BEN
22 7021 7D                    LD   A,L     ;KEZDOCIM ALSO BYTE
23 7022 CD6402                CALL 0264H   ;WRBYTE
24 7025 7C                    LD   A,H     ;FELSO BYTE
25 7026 CD6402                CALL 0264H   ;WRBYTE
26 7029 7E           ADAT:    LD   A,(HL)  ;ADAT BETOLTESE
27 702A CD6402                CALL 0264H   ;WRBYTE
28 702D 23                    INC  HL      ;KOVETKEZO CIM
29 702E 1B                    DEC  DE      ;HOSSZ-1
30 702F 7B                    LD   A,E     ;HOSSZ=0?
31 7030 B2                    OR   D
32 7031 20F6                  JR   NZ,ADAT;KOVETKEZO BYTE
33 7033 CDF801                CALL 01F8H   ;CASOFF
34 7036 18FE         VEGE:    JR   VEGE
```

```
35 7038 CDB31B   CIM:   CALL 1BB3H  ;QINPUT
36 703B 23              INC  HL
37 703C CD6C0E          CALL 0E6CH  ;ASCINT
38 703F 2A2141          LD   HL,(4121H);CIM HL-BE
39 7042 C9              RET
40                      END
```

A programot kiegészíthetnénk azzal, hogy a billentyűzetről érkező karaktereket – ahogyan az előző fejezetben láttuk – az RST 10H szubrutin felhasználásával megvizsgáljuk.

Az előző programmal készített adatszalagot a következő programmal olvashatjuk be:

```
 1                ;59. LISTA
 2                        ORG  7000H
 3                        LOAD 7000H
 4 7000 AF                XOR  A       ;BELSO MAGNO
 5 7001 CD1202            CALL 0212H   ;DEFCAS
 6 7004 CD9602            CALL 0296H   ;RDHDR
 7 7007 CD1403            CALL 0314H   ;RDADDR
 8 700A E5                PUSH HL      ;HOSSZ MEGORZES
 9 700B CD1403            CALL 0314H   ;RDADDR
10 700E D1                POP  DE      ;HL=KEZD, DE=HOSSZ
11 700F CD3502   ADAT:    CALL 0235H   ;RDBYTE
12 7012 77                LD   (HL),A  ;ADAT TAROLASA
13 7013 1B                DEC  DE      ;HOSSZ-1
14 7014 23                INC  HL      ;KOVETKEZO CIM
15 7015 7B                LD   A,E     ;VAN MEG ADAT?
16 7016 B2                OR   D
17 7017 20F6              JR   NZ,ADAT;KOVETKEZO ADAT
18 7019 CDF801            CALL 01F8H   ;CASOFF
19 701C 18FE     VEGE:    JR   VEGE
20                        END
```

Megjegyezzük, hogy az előző fejezetben tárgyalt OUTLIN szubrutinnal teljes sorokat írhatunk a kazettára, ha a 409CH tárcímre FFH-t írunk.

SYSTEM formájú szalag betöltését a

CALL 02B2H SYSTEM

szubrutin meghívásával kezdeményezhetjük. A szubrutin hívásának hatására megjelenik a "*?" üzenet, amelyre a SYSTEM program nevét válaszolva a magnó elindul, és a BASIC értelmező betölti a gépi kódú programot. SYSTEM szalag írására nincs ROM-rutin, utolsó példánk egy ilyen szalagot készítő program, amelynek NAME, START, ENDADR, EXEC címkéinél a program elindítása előtt el kell helyezni a program nevét, kezdőcímét, végcímét és belépési címét.

```
 1                ;60. LISTA
 2                        ORG  7E00H
 3                        LOAD 7E00H
 4 7E00 FD21787E          LD   IY,NAME;IY=NEV MUTATO
 5 7E04 AF                XOR  A       ;BELSO MAGNO
 6 7E05 CD1202            CALL 0212H   ;DEFCAS
 7 7E08 2A807E            LD   HL,(ENDADR);HL=VEGCIM
 8 7E0B ED5B7E7E          LD   DE,(START);KEZDOCIM
 9 7E0F AF                XOR  A       ;C JELZOBIT TORLESE
10 7E10 ED52              SBC  HL,DE   ;HOSSZ-1=HL-DE
11 7E12 23                INC  HL      ;HOSSZ
12 7E13 CD8702            CALL 0287H   ;WRHDR
13 7E16 3E55              LD   A,55H   ;SYSTEM FILE JELZO
14 7E18 CD6402            CALL 0264H   ;WRBYTE
15 7E1B 0606              LD   B,6     ;6 BYTE PROGRAMNEV
```

```
16 7E1D FD7E00    NEV:    LD   A,(IY+0)
17 7E20 CD6402            CALL 0264H  ;WRBYTE
18 7E23 FD23              INC  IY     ;KOVETKEZO KARAKTER
19 7E25 10F6              DJNZ NEV
20 7E27 25        CIKL:   DEC  H      ;ADATBLOKK IRASA
21 7E28 FA397E            JP   M,CIKL1;256-NAL KISEBB?
22 7E2B 3E3C              LD   A,3CH  ;BLOKK ELEJE
23 7E2D CD6402            CALL 0264H  ;WRBYTE
24 7E30 AF                XOR  A      ;BLOKKHOSSZ=256=00H
25 7E31 CD6402            CALL 0264H  ;WRBYTE
26 7E34 CD5F7E            CALL BLOKK  ;ADATBLOKK IRASA
27 7E37 18EE              JR   CIKL
28 7E39 AF        CIKL1:  XOR  A      ;UTOLSO BLOKK
29 7E3A BD                CP   L      ;HOSSZ=0?
30 7E3B 280C              JR   Z,VEGJEL;0=FILE VEGE
31 7E3D 3E3C              LD   A,3CH  ;BLOKK ELEJE
32 7E3F CD6402            CALL 0264H  ;WRBYTE
33 7E42 7D                LD   A,L    ;BLOKK HOSSZA
34 7E43 CD6402            CALL 0264H  ;WRBYTE
35 7E46 CD5F7E            CALL BLOKK  ;ADATBLOKK IRASA
36 7E49 3E78      VEGJEL: LD   A,78H  ;VEGJEL
37 7E4B CD6402            CALL 0264H  ;WRBYTE
38 7E4E 3A827E            LD   A,(EXEC);BELEPESI CIM ALSO
39 7E51 CD6402            CALL 0264H  ;WRBYTE
40 7E54 3A837E            LD   A,(EXEC+1);CIM FELSO
41 7E57 CD6402            CALL 0264H  ;WRBYTE
42 7E5A CDF801            CALL 01F8H  ;CASOFF
43 7E5D 18FE      VEGE:   JR   VEGE
44 7E5F 47        BLOKK:  LD   B,A    ;BYTE-OK SZAMA
45 7E60 7B                LD   A,E    ;BETOLTESI CIM ALSO
46 7E61 CD6402            CALL 0264H  ;WRBYTE
47 7E64 7A                LD   A,D    ;CIM FELSO
48 7E65 CD6402            CALL 0264H  ;WRBYTE
49 7E68 83                ADD  A,E    ;ELLENORZO OSSZEG
50 7E69 4F                LD   C,A    ;SZAMOLAS
51 7E6A 1A        BLOKK1: LD   A,(DE) ;KOVETKEZO ADATBYTE
52 7E6B CD6402            CALL 0264H  ;WRBYTE
53 7E6E 81                ADD  A,C    ;ELLENORZO OSSZEG
54 7E6F 4F                LD   C,A    ;SZAMOLAS
55 7E70 13                INC  DE     ;KOVETKEZO BYTE
56 7E71 10F7              DJNZ BLOKK1
57 7E73 79                LD   A,C    ;ELLENORZO OSSZEG
58 7E74 CD6402            CALL 0264H  ;WRBYTE
59 7E77 C9                RET
60 7E78 20202020  NAME:   DB           ;PROGRAMNEV
60 7E7C 2020
61 7E7E 0070      START:  DW   7000H  ;KEZDOCIM
62 7E80 0071      ENDADR: DW   7100H  ;VEGCIM
63 7E82 0070      EXEC:   DW   7000H  ;BELEPESI CIM
64                        END
```

Nyomtatás

A nyomtatóra a

CALL 039CH PRTCHR

szubrutin meghívásával írhatunk. Ez a szubrutin az A regiszter tartalmát a 4025H...4032H nyomtatót-ellenőrző blokkal meghatározott módon viszi a nyomtatóra, felismeri a nyomtatót vezérlő karaktereket, és szükség szerint átírja a 4029H sorszámot és a 409BH karakterszámot. A nyomtatót-vezérlő karakterek a következők:

00 H nyomtató állapot lekérdezés
0A H soremelés (ugyanaz, mint 0DH)

0BH lapemelés
0CH feltételes lapemelés (csak akkor emel, ha éppen nem a lap tetején van)
0DH soremelés (ha a lap betelt, akkor lapemelés)

Az előbbitől annyiban különbözik a PRTCHR szubrutin része, a

CALL 003BH PRBYTE

szubrutin, hogy nem foglalkozik a vezérlőkarakterek észlelésével és a 4029H sorszám, 409BH karakterszám átírásával, így pl. ennek meghívásával két karaktert írhatunk egymásra. A nyomtatóra teljes sort a már több helyen említett OUTLIN szubrutinnal lehet kivinni.

Megszakítások kezelése

Az előzőkben leírt be- és kimeneti szubrutinok közül a kazettakezelő szubrutinok várakozással, a nyomtató szubrutinok pedig az állapotbitek folyamatos figyelésével várták, hogy a periféria mikor kész arra, hogy újabb adatot fogadjon. A várakozás, ill. a folyamatos figyelés időt köt le, amely alatt a számítógép más tevékenységet nem végezhet. Az adatátvitel egy másik, hatásosabb módja az, hogy a számítógép nem figyeli a periféria állapotát, hanem csak a főprogramot hajtja végre, és a periféria egy megszakításkérő vonalon jelet küld, ha kész az adatátvitelre. A megszakításkérés hatására a számítógép megállítja az éppen végrehajtás alatt álló programot, és a vezérlést a megszakítást kiszolgáló rutinnak adja át, amely végrehajtja az adatátvitelt, majd visszaadja a vezérlést az eredeti programnak.

Ha megszakításkérés érkezik, akkor a számítógép befejezi az éppen végrehajtás alatt álló utasítást, megőrzi a következő végrehajtandó utasítás címét, és átadja a vezérlést a megszakítást kezelő rutinnak. A megszakított program futtatásának folytatásához meg kell őrizni a regiszterek és a jelzőbitek tartalmát, ezért a megszakítást kezelő rutinnak azzal kell kezdődnie, hogy ezeket az információkat kimenti. A megszakítást kezelő rutin ezután végrehajtja a számítógép és a periféria közötti adatátvitel lebonyolításához szükséges utasításokat. Az adatátvitel befejeződése után visszatölti a regiszterek és a jelzőbitek előző tartalmát, és ezután a megszakított program végrehajtása a következő utasításnál folytatódhat.

A Z80 mikroprocesszornál a megszakítások két alapvető típusa a nem maszkolható és a maszkolható megszakítás.

A vezérlőbuszon érkező nem maszkolható megszakításkérés hatására a Z80 a programszámláló értékét kimenti a veremterületre, majd a 0066H címre ugrik. A nem maszkolható megszakítást kezelő rutinnak mindig ezen a címen kell kezdődnie. A nem maszkolható megszakításkérés hatására feltétlen ugrás következik a megszakítást kiszolgáló rutinra, tehát a megszakításkérés elfogadását nem lehet letiltani.

A vezérlőbuszon érkező maszkolható megszakításkérés elfogadását a programozó a

DI F3

utasítással (Disable Interrupt) tilthatja, az

EI FB

utasítással (Enable Interrupt) engedélyezheti. A megszakítást akkor szokás tilani, ha a program olyan részénél tartunk (pl. inicializálás, másik megszakítás kiszolgálása stb.), amelynek megszakítása hibákat eredményezhetne. A maszkolható megszakításkérés elfogadása tiltja a további ugyanilyen kérések elfogadását, ezért a megszakítást kezelő rutin végén EI utasítással újra engedélyezni kell a megszakításokat.

A maszkolható megszakítás háromféle lehet. Az

IM 0 ED 46

utasítással (Interrupt Mode) beállítható "0" módú megszakításkezelési üzemmódban a Z80 az adatbuszról byte-onként beolvasott utasítást hajtja végre. Az

IM 1 ED 56

utasítással beállítható "1" módú megszakítás RST 38H szubrutinhívást eredményez. A HT-1080Z ezt a megszakítástípust a lemezes be- és kimeneti műveletek kiszolgálására használja. Az

IM 2 ED 5E

utasítással beállítható "2" módú megszakítás az I regiszter 8 bitjéből, és az adatbuszra helyezett 8 bitből képzett címen levő szubrutint hívja meg.

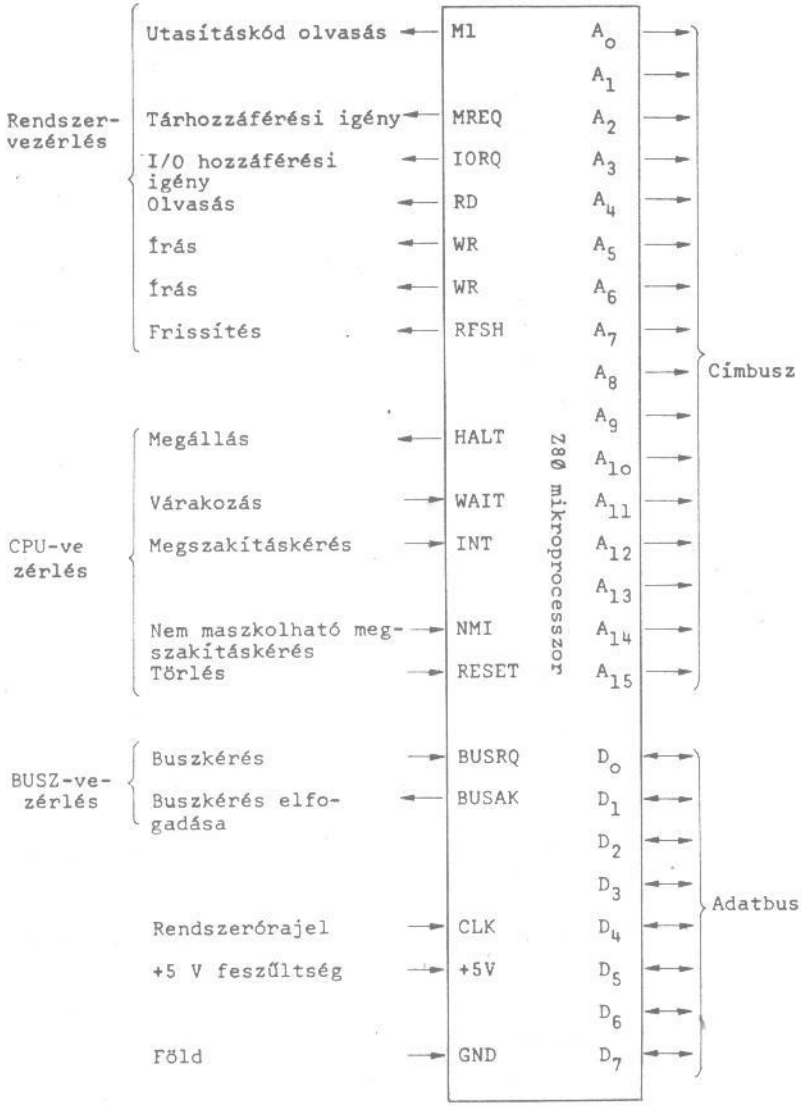
A nem maszkolható, ill. a maszkolható megszakítást kezelő rutinból a

RETN ED 45
RETI ED 4D

utasításokkal (**RET**urn from **N**on-maskable, **RET**urn from Interrupt) lehet visszatérni. A megszakításkezeléssel kapcsolatos utasítások a jelzőbiteket nem állítják.

A Függelék

Z80 mikroprocesszor



Z80 utasítás mnemonikok és jelentésük

mnemonik	angol	magyar
ADC	**AD**d with **C**arry	összeadás átvitellel
ADD	**ADD**	összeadás
AND	**AND**	és
BIT	**BI**t Test	bitvizsgálat
CALL	**CALL**	hívás
CCF	**C**omplement **C**arry **F**lag	C jelzőbit átállítása
CP	**C**om**P**are	összehasonlítás
CPD	block **C**om**P**are **D**ecrement address	blokk-összehasonlítás csökkentéssel
CPDR	block **C**om**P**are **D**ecrement address, auto-**R**epeat	blokk-összehasonlítás cím csökkentéssel, automatikus ismétléssel
CPI	block **C**om**P**are **I**ncrement address	blokk-összehasonlítás cím növeléssel
CPIR	block **C**om**P**are **I**ncrement address, auto-**R**epeat	blokk-összehasonlítás cím növeléssel, automatikus növeléssel
CPL	**C**om**PL**ement accumulator	A regiszter komplemense
DAA	**D**ecimal **A**djust **A**ccumulator	A regiszter decimális igazítása
DEC	**de**crement	csökkentés
DI	**D**isable **I**nterrupt	megszakítástiltás
DJNZ	**D**ecrement B and **J**ump if **N**on-**Z**ero	B csökkentése és ugrás, ha nem nulla
EI	**E**nable **I**nterrupt	megszakítás-engedélyezés
EX,EXX	**EX**change	felcserélés
HALT	**HALT**	megállás
IM	**I**nterrupt **M**ode	megszakítási mód
IN	**IN**put	bevitel
INC	**INC**rement	növelés
IND	block **IN**put **D**ecrement address	blokkbevitel cím csökkentéssel
INDR	block **IN**put **D**ecrement address auto-**R**epeat	blokkbevitel cím csökkentéssel, automatikus ismétléssel
INI	block **IN**put **I**ncrement address	blokkbevitel cím növeléssel
INIR	block **IN**put **I**ncrement address auto-**R**epeat	blokkbevitel cím növeléssel, automatikus ismétléssel
JP	**J**um**P**	ugrás
JR	**J**ump **R**elative	relatív ugrás

mnemonik	angol	magyar
LD	**LoaD**	átmozgatás
LDD	block **LoaD** Decrement address	blokkmozgatás cím csökkentéssel
LDDR	block **LoaD** Decrement address auto-Repeat	blokkmozgatás cím csökkentéssel, automatikus ismétléssel
LDI	block **LoaD** Increment address	blokkmozgatás cím növelésével
LDIR	bolck **LoaD** Increment address auto-Repeat	blokkmozgatás cím növeléssel, automatikus ismétléssel
NEG	**NEG**ate A	A ellentettje
NOP	No **OP**eration	üres utasítás
OR	**OR**	vagy
OTDR	block **OuTput** Decrement address auto-Repeat	blokk-kivitel cím csökkentéssel, automatikus ismétléssel
OTIR	block **OuTput** Increment address auto-Repeat	blokk-kivitel cím növeléssel, automatikus ismétléssel
OUT	**OUT**put	kivitel
OUTD	block **OUT**put Decrement address	blokk-kivitel cím csökkentéssel
OUTI	block **OUT**put Increment address	blokk-kivitel cím növeléssel
POP	**POP**	(veremről)elővesz
PUSH	**PUSH**	(veremre) tesz
RES	**RES**et	kikapcsolás
RET	**RET**urn	visszatérés
RETI	**RET**urn from **I**nterrupt	visszatérés megszakításból
RETN	**RET**urn from Non-maskable interrupt	nem maszkolható megszakításból
RL	Rotate Left	ciklikus eltolás balra C-n át
RLA	Rotate Left Accumulator	A regiszter ciklikus eltolása balra C-n át
RLC	Rotate Left Circular	ciklikus eltolás balra
RLCA	Rotate Left Circular Accumulator	A regiszter ciklikus eltolása balra
RLD	Rotate Left Digit	félbyte-os ciklikus eltolás balra
RR	Rotate Right	ciklikus eltolás jobbra C-n át

név	angol	magyar
RRA	**R**otate **R**ight **A**ccumulator	A regiszter ciklikus eltolása jobbra C-n át
RRC	**R**otate **R**ight **C**ircular	ciklikus eltolás jobbra
RRCA	**R**otate **R**ight **C**ircular **A**ccumulator	A regiszter ciklikus eltolása jobbra
RRD	**R**otate **R**ight **D**igit	félbyte-os ciklikus eltolás jobbra
RST	**R**e**ST**art	újraindítás
SBC	**S**u**B**tract with **C**arry	kivonás átvitellel
SCF	**S**et **C**arry **F**lag	C jelzőbit állítása
SET	**SET**	bekapcsolás
SLA	**S**hift **L**eft **A**rithmetic	aritmetikai eltolás balra
SRA	**S**hift **R**ight **A**rithmetic	aritmetikai eltolás jobbra
SRL	**S**hift **R**ight **L**ogical	logikai eltolás jobbra
SUB	**SUB**tract	kivonás
XOR	e**X**clusive **OR**	kizáró vagy

Regiszternevek, jelzőbitek és jelentésük

név	angol	magyar
A	Accumulator	akkumulátor, tároló
BC	Byte Count	byte-szám
DE	DEstination	cél
H	High	felső
L	Low	alsó
F	Flag	jelzőbit
IX	IndeX	index
SP	Stack Pointer	veremmutató
PC	Program Counter	programszámláló
R	Refresh	frissítő
I	Interrupt vector	megszakításvektor
S	Sign	előjel
Z	Zero	nulla
H	Half-carry	félbyte-os átvitel
P/V	Parity/oVerflow	paritás/túlcsordulás
N	Negative	negatív
C	Carry	átvitel

Z80 utasítások hexadecimális kód szerint
1. Egybyte-os utasítások

Kód	Mnemonik	Kód	Mnemonik	Kód	Mnemonik
00	NOP	10 e	DJNZ e	20 e	RJ NZ,e
01 n	LD BC,n	11 n	LD DE,n	21 n	LD HL,n
02	LD (BC),A	12	LD (DE),A	22 nn	LD (nn),HL
03	INC BC	13	INC DE	23	INC HL
04	INC B	14	INC D	24	INC H
05	DEC B	15	DEC D	25	DEC H
06 n	LD B n	16 n	LD D,n	26 n	LD H,n
07	RLCA	17	RLA	27	DAA
08	EX AF,AF′	18 e	JR e	28 e	JR Z,e
09	ADD HL,BC	19	ADD HL,DE	29	ADD HL,HL
0A	LD A,(BC)	1A	LD A,(DE)	2A nn	LD HL,(nn)
0B	DEC BC	1B	DEC DE	2B	DEC HL
0C	INC C	1C	INC E	2C	INC L
0D	DEC C	1D	DEC E	2D	DEC L
0E n	LD C,n	1E n	LD E,n	2E n	LD L,n
0F	RRCA	1F	RRA	2F	CPL

Kód	Mnemonik	Kód	Mnemonik	Kód	Mnemonik
30 e	JR NC,e	40	LD B,B	50	LD D,B
31 n	LD SP,n	41	LD B,C	51	LD D,C
32 n	LD (nn),A	42	LD B,D	52	LD D,D
33	INC SP	43	LD B,E	53	LD D,E
34	INC (HL)	44	LD B,H	54	LD D,H
35	DEC (HL)	45	LD B,L	55	LD D,L
36 n	LD (HL),n	46	LD B,(HL)	56	LD D,(HL)
37	SCF	47	LD B,A	57	LD D,A
38 e	JR C,e	48	LD C,B	58	LD E,B
39	ADD HL,SP	49	LD C,C	59	LD E,C
3A nn	LD A,(nn)	4A	LD C,D	5A	LD E,D
3B	DEC SP	4B	LD C,E	5B	LD E,E
3C	INC A	4C	LD C,H	5C	LD E,H
3D	DEC A	4D	LD C,L	5D	LD E,L
3E n	LD A,n	4E	LD C,(HL)	5E	LD E,(HL)
3F	CCF	4F	LD C,A	5F	LD E,A

Kód	Mnemonik	Kód	Mnemonik	Kód	Mnemonik
60	LD H,B	70	LD (HL),B	80	ADD A,B
61	LD H,C	71	LD (HL),C	81	ADD A,C
62	LD H,D	72	LD (HL),D	82	ADD A,D
63	LD H,E	73	LD (HL),E	83	ADD A,E
64	LD H,H	74	LD (HL),H	84	ADD A,H
65	LD H,L	75	LD (HL),L	85	ADD A,L
66	LD H,(HL)	76	HALT	86	ADD A,(HL)
67	LD H,A	77	LD (HL),A	87	ADD A,A
68	LD L,B	78	LD A,B	88	ADC A,B
69	LD L,C	79	LD A,C	89	ADC A,C
6A	LD L,D	7A	LD A,D	8A	ADC A,D
6B	LD L,E	7B	LD A,E	8B	ADC A,E
6C	LD L,H	7C	LD A,H	8C	ADC A,H
6D	LD L,L	7D	LD A,L	8D	ADC A,L
6E	LD L,(HL)	7E	LD A,(HL)	8E	ADC A,(HL)
6F	LD L,A	7F	LD A,A	8F	ADC A,A

Kód	Mnemonik	Kód	Mnemonik	Kód	Mnemonik
90	SUB B	A0	AND B	B0	OR B
91	SUB C	A1	AND C	B1	OR C
92	SUB D	A2	AND D	B2	OR D
93	SUB E	A3	AND E	B3	OR E
94	SUB H	A4	AND H	B4	OR H
95	SUB L	A5	AND L	B5	OR L
96	SUB (HL)	A6	AND (HL)	b6	OR (HL)
97	SUB A	A7	AND A	B7	OR A
98	SBC A,B	A8	XOR B	B8	CP B
99	SBC A,C	A9	XOR C	B9	CP C
9A	SBC A,D	AA	XOR C	BA	CP D
9B	SBC A,E	AB	XOR E	BB	CP E
9C	SBC A,H	AC	XOR H	BC	CP H
9D	SBC A,L	AD	XOR L	BD	CP L
9E	SBC A,(HL)	AE	XOR (HL)	BE	CP(HL)
9F	SBC A,A	AF	XOR A	BF	CD A

Kód	Mnemonik
C0	RET NZ
C1	POP BC
C2 nn	JP NZ,nn
C3 nn	JP nn
C4 nn	CALL NZ,nn
C5	PUSH BC
C6 n	ADD A,n
C7	RST 0H
C8	RET Z
C9	RET
CA nn	JP Z,nn
CB (l. a 2.1. táblázatot)	
CC nn	CALL Z,nn
CD nn	CALL nn
CE n	ADC A,n
CF	RST 8H

Kód	Mnemonik
D0	RET NC
D1	POP DE
D2 nn	JP NC,nn
D3 n	OUT n,A
D4 nn	CALL NC,nn
D5	PUSH DE
D6 n	SUB n
D7	RST 10H
D8	RET C
D9	EXX
DA nn	JP C,nn
DB n	IN A,n
DC nn	CALL C,nn
DD (l. a 2.3. táblázatot)	
DE n	SBC A,n
DF	RST 18H

Kód	Mnemonik
E0	RET PO
E1	POP HL
E2 nn	JP PO,nn
E3	EX (SP),HL
E4 nn	CALL PO,nn
E5	PUSH HL
E6 n	AND n
E7	RST 20H
E8	RET PE
E9	JP (HL)
EA nn	JP PE,nn
EB	EX DE,HL
EC nn	CALL PE,nn
ED (l. a 2.2. táblázat)	
EE n	XOR n
EF	RST 28H

Kód	Mnemonik
F0	RET P
F1	POP AF
F2 nn	JP P,nn
F3	DI
F4 nn	CALL P,nn
F5	PUSH AF
F6 n	OR n
F7	RST 30H
F8	RET M
F9	LD SP,HL
FA nn	JP M,nn
FB	EI
FC nn	CALL M,nn
FD (l. a 2.3. táblázatot)	
FE n	CP n
FF	RST 38H

2.1. CB XX kétbyte-os utasítások

XX	Mnemonik	XX	Mnemonik	XX	Mnemonik
00	RLC B	10	RL B	20	SLA B
01	RLC C	11	RL C	21	SLA C
02	RLC D	12	RL D	22	SLA D
03	RLC E	13	RL E	23	SLA E
04	RLC H	14	RL H	24	SLA H
05	RLC L	15	RL L	25	SLA L
06	RLC (HL)	16	RL (HL)	26	SLA (HL)
07	RLC A	17	RL A	27	SLA A
08	RRC B	18	RR B	28	SRA B
09	RRC C	19	RR C	29	SRA C
0A	RRC D	1A	RR D	2A	SRA D
0B	RRC E	1B	RR E	2B	SRA E
0C	RRC H	1C	RR H	2C	SRA H
0D	RRC L	1D	RR L	2D	SRA L
0E	RRC (HL)	1E	RR (HL)	2E	SRA (HL)
0F	RRC A	1F	RR A	2F	SRA A

XX	Mnemonik	XX	Mnemonik	XX	Mnemonik
30	–	40	BIT 0,B	50	BIT 2,B
31	–	41	BIT 0,C	51	BIT 2,C
32	–	42	BIT 0,D	52	BIT 2,D
33	–	43	BIT 0,E	53	BIT 2,E
34	–	44	BIT 0,H	54	BIT 2,H
35	–	45	BIT 0,L	55	BIT 2,L
36	–	46	BIT 0,(HL)	56	BIT 2,(HL)
37	–	47	BIT 0,A	57	BIT 2,A
38	SRL B	48	BIT I,B	58	BIT 3,B
39	SRL C	49	BIT 1,C	59	BIT 3,C
3A	SRL D	4A	BIT 1,D	5A	BIT 3,D
3B	SRL E	4B	BIT 1,E	5B	BIT 3,E
3C	SRL H	4C	BIT 1,H	5C	BIT 3,H
3D	SRL L	4D	BIT 1,L	5D	BIT 3,L
3E	SRL (HL)	4E	BIT 1,(HL)	5E	BIT 3,(HL)
3F	SRL A	4F	BIT 1,A	5F	BIT 3,A

XX	Mnemonik	XX	Mnemonik	XX	Mnemonik
60	BIT 4,B	70	BIT 6,B	80	RES 0,B
61	BIT 4,C	71	BIT 6,C	81	RES 0,C
62	BIT 4,D	72	BIT 6,D	82	RES 0,D
63	BIT 4,E	73	BIT 6,E	83	RES 0,E
64	BIT 4,H	74	BIT 6,H	84	RES 0,H
65	BIT 4,L	75	BIT 6,L	85	RES 0,L
66	BIT 4,(HL)	76	BIT 6,(HL)	86	RES 0,(HL)
67	BIT 4,A	77	BIT 6,A	87	RES 0,A
68	BIT 5,B	78	BIT 7,B	88	RES 1,B
69	BIT 5,C	79	BIT 7,C	89	RES 1,C
6A	BIT 5,D	7A	BIT 7,D	8A	RES 1,D
6B	BIT 5,E	7B	BIT 7,E	8B	RES 1,E
6C	BIT 5,H	7C	BIT 7,H	8C	RES 1,H
6D	BIT 5,L	7D	BIT 7,L	8D	RES 1,L
6E	BIT 5,(HL)	7E	BIT 7,(HL)	8E	RES 1,(HL)
6F	BIT 5,A	7F	BIT 7,A	8F	RES 1,A

XX	Mnemonik	XX	Mnemonik	XX	Mnemonik
90	RES 2,B	A0	RES 4,B	B0	RES 6,B
91	RES 2,C	A1	RES 4,C	B1	RES 6,C
92	RES 2,D	A2	RES 4,D	B2	RES 6,D
93	RES 2,E	A3	RES 4,E	B3	RES 6,E
94	RES 2,H	A4	RES 4,H	B4	RES 6,H
95	RES 2,L	A5	RES 4,L	B5	RES 6,L
96	RES 2,(HL)	A6	RES 4,(HL)	B6	RES 6,(HL)
97	RES 2,A	A7	RES 4,A	B7	RES 6,A
98	RES 3,B	A8	RES 5,B	B8	RES 7,B
99	RES 3,C	A9	RES 5,C	B9	RES 7,C
9A	RES 3,D	AA	RES 5,D	BA	RES 7,D
9B	RES 3,E	AB	RES 5,E	BB	RES 7,E
9C	RES 3,H	AC	RES 5,H	BC	RES 7,H
9D	RES 3,L	AD	RES 5,L	BD	RES 7,L
9E	RES 3,(HL)	AE	RES 5,(HL)	BE	RES 7,(HL)
9F	RES 3,A	AF	RES 5,A	BF	RES 7,A

XX	Mnemonik
C0	SET 0,B
C1	SET 0,C
C2	SET 0,D
C3	SET 0,E
C4	SET 0,H
C5	SET 0,L
C6	SET 0,(HL)
C7	SET 0,A
C8	SET 1,B
C9	SET 1,C
CA	SET 1,D
CB	SET 1,E
CC	SET 1,H
CD	SET 1,L
CE	SET 1,(HL)
CF	SET 1,A

XX	Mnemonik
D0	SET 2,B
D1	SET 2,C
D2	SET 2,D
D3	SET 2,E
D4	SET 2,H
D5	SET 2,L
D6	SET 2,(HL)
D7	SET 2,A
D8	SET 3,B
D9	SET 3,C
DA	SET 3,D
DB	SET 3,E
DC	SET 3,H
DD	SET 3,L
DE	SET 3,(HL)
DF	SET 3,A

XX	Mnemonik
E0	SET 4,B
E1	SET 4,C
E2	SET 4,D
E3	SET 4,E
E4	SET 4,H
E5	SET 4,L
E6	SET 4,(HL)
E7	SET 4,A
E8	SET 5,B
E9	SET 5,C
EA	SET 5,D
EB	SET 5,E
EC	EET 5,H
ED	SET 5,L
EE	SET 5,(HL)
EF	SET 5,A

XX	Mnemonik
F0	SET 6,B
F1	SET 6,C
F2	SET 6,D
F3	SET 6,E
F4	SET 6,H
F5	SET 6,L
F6	SET 6,(HL)
F7	SET 6,A
F8	SET 7,B
F9	SET 7,C
FA	SET 7,D
FB	SET 7,E
FC	SET 7,H
FD	SET 7,L
FE	SET 7,(HL)
FF	SET 7,A

2.2. ED XX kétbyte-os utasítások

XX	Mnemonik	XX	Mnemonik	XX	Mnemonik
40	IN B,(C)	50	IN D,(C)	60	IN H,(c)
41	OUT (C),B	51	OUT (C),D	61	OUT (C),H
42	SBC HL,BC	52	SBC HL,DE	62	SBC HL,HL
43 nn	LD (nn),BC	53 nn	LD (nn),DE	63	–
44	NEG	54	–	64	–
45	RETN	55	–	65	–
46	IM 0	56	IM 1	66	–
47	LD I,A	57	LD A,I	67	RRD
48	IN C,(C)	58	IN E,(C)	68	IN L,(C)
49	OUT (C),C	59	OUT (C), E	69	OUT (C),L
4A	ADC HL,BC	5A	ADC HL,DE	6A	ADC HL,HL
4B nn	LD BC,(nn)	5B nn	LD DE,(nn)	6B	–
4C	–	5C	–	6C	–
4D	RETI	5D	–	6D	–
4E	–	5E	IM 2	6E	–
4F	–	5F	–	6F	RLD

XX	Mnemonik	XX	Mnemonik	XX	Mnemonik
70	–	A0	LDI	B0	LDIR
71	–	A1	CPI	B1	CPIR
72	SBC HL,SP	A2	INI	B2	INIR
73 nn	LD (nn),SP	A3	OUTI	B3	OTIR
74	–	A4	–	B4	–
75	–	A5	–	B5	–
76	–	A6	–	B6	–
77	–	A7	–	B7	–
78	IN A,(C)	A8	LDD	B8	LDDR
79	OUT (C),A	A9	CPD	B9	CPDR
7A	ADC HL,SP	AA	IND	BA	INDR
7B nn	LD SP,(nn)	AB	OUTD	BB	OTDR
7C	–	AC	–	BC	–
7D	–	AD	–	BD	–
7E	–	AE	–	BE	–
7F	–	AF	–	BF	–

2.3. DD XX és FD XX kétbyte-os utasítások (IX) = DD XX (IY) = FD XX

XX	Mnemonik	XX	Mnemonik	XX	Mnemonik
00	ADD IX,BC	10	–	20	–
01	–	11	–	21 nn	LD IX,nn
02	–	12	–	22 nn	LD (nn),IX
03	–	13	–	23	INC IX
04	–	14	–	24	–
05	–	15	–	25	–
06	–	16	–	26	–
07	–	17	–	27	–
08	–	18	–	28	–
09	ADD IX,BC	19	ADD IX,DE	29	ADD IX,IX
0A	–	1A	–	2A nn	LD IX,(nn)
0B	–	1B	–	2B	DEC IX
0C	–	1C	–	2C	–
0D	–	1D	–	2D	–
0E	–	1E	–	2E	–
0F	–	1F	–	2F	–
	–				

XX	Mnemonik	XX	Mnemonik	XX	Mnemonik
30	–	40	–	50	–
31	–	41	–	51	–
32	–	42	–	52	–
33	–	43	–	53	–
34 d	INC (IX+d)	44	–	54	–
35 d	DEC (IX+d)	45	–	55	–
36 d n	LD (IX+d),n	46 d	LD B,(IX+d)	56 d	LD D,(IX+d)
37	–	47	–	57	–
38	–	48	–	58	–
39	ADD IX,SP	49	–	59	–
3A	–	4A	–	5A	–
3B	–	4B	–	5B	–
3C	–	4C	–	5C	–
3D	–	4D	–	5D	–
3E	–	4E d	LD C,(IX+d)	5E d	LD E,(IX+d)
3F	–	4F	–	5F	–

XX	Mnemonik
60	–
61	–
62	–
63	–
64	–
65	–
66 d	LD H,(IX+d)
67	–
68	–
69	–
6A	–
6A	–
6C	–
6D	–
6E d	LD L,(IX+d)
6F	–

XX	Mnemonik
70 d	LD (IX+d),B
71 d	LD (IX+d),C
72 d	LD (IX+d),D
73 d	LD (IX+d),É
74 d	LD (IX+d),H
75 d	LD (IX+d),L
76	–
77 d	LD (IX+d),A
78	–
79	–
7A	–
7B	–
7C	–
7D	–
7E d	LD A,(IX+d)
7F	–

XX	Mnemonik
80	–
81	–
82	–
83	–
84	–
85	–
86 d	ADD A,(IX+d)
87	–
88	–
89	–
8A	–
8B	–
8C	–
8D	–
8E d	ADC A,(IX+d)
8F	–

XX	Mnemonik
90	–
91	–
92	–
93	–
94	–
95	–
96 d	SUB (IX+d)
97	–
98	–
99	–
9A	–
9B	–
9C	–
9D	–
9E d	SBC A,(IX+d)
9F	–

XX	Mnemonik
A0	–
A1	–
A2	–
A3	–
A4	–
A5	–
A6 d	AND (IX+d)
A7	–
A8	–
A9	–
AA	–
AB	–
AC	–
AD	–
AE d	XOR (IX+d)
AF	–

XX	Mnemonik
B0	–
B1	–
B2	–
B3	–
B4	–
B5	–
B6 d	OR (IX+d)
B7	–
B8	–
B9	–
BA	–
BB	–
BC	–
BD	–
BE d	CP (IX+d)
BF	–

XX	Mnemonik
C0	–
C1	–
C2	–
C3	–
C4	–
C5	–
C6	–
C7	–
C8	–
C9	–
CA	–
CB	(l. a 3. tábl.)
CC	–
CD	–
CE	–
CF	–

XX	Mnemonik
D0	–
D1	–
D2	–
D3	–
D4	–
D5	–
D6	–
D7	–
D8	–
D9	–
DA	–
DB	–
DC	–
DD	–
DE	–
DF	–

XX	Mnemonik
E0	–
E1	POP IX
E2	–
E3	EX (SP),IX
E4	–
E5	PUSH IX
E6	–
E7	–
E8	–
E9	JP (IX)
EA	–
EB	–
EC	–
ED	–
EE	–
EF	–

XX	Mnemonik
F0	–
F1	–
F2	–
F3	–
F4	–
F5	–
F6	–
F7	–
F8	–
F9	LD SP,IX
FA	–
FB	–
FC	–
FD	–
FE	–
FF	–

3. Hárombyte-os utasítások
(IX) = DD CB d XX
(IY) = FD CB d XX

XX	Mnemonik
CB d 06	RLC (IX+d)
CB d 0E	RRC (IX+d)
CB d 16	RL (IX+d)
CB d 1E	RR (IX+d)
CB d 26	SLA (IX+d)
CB d 2E	SRA (IX+d)
CB d 3E	SRL (IX+d)
CB d 46	BIT 0,(IX+d)
CB d 4E	BIT 1,(IX+d)
CB d 56	BIT 2,(IX+d)
CB d 5E	BIT 3,(IX+d)
CB d 66	BIT 4,(IX+d)
CB d 6E	BIT 5,(IX+d)
CB d 76	BIT 6,(IX+d)
CB d 7E	BIT 7,(IX+d)

XX	Mnemonik
CB d 86	RES 0,(IX+d)
CB d 8E	RES 1,(IX+d)
CB d 96	RES 2,(IX+d)
CB d 9E	RES 3,(IX+d)
CB d A6	RES 4,(IX+d)
CB d AE	RES 5,(IX+d)
CB d B6	RES 6,(IX+d)
CB d BE	RES 7,(IX+d)
CB d C6	SET 0,(IX+d)
CB d CE	SET 1,(IX+d)
CB d D6	SET 2,(IX+d)
CB d DE	SET 3,(IX+d)
CB d E6	SET 4,(IX+d)
CB d EE	SET 5,(IX+d)
CB d F6	SET 6,(IX+d)
CB d FE	SET 7,(IX+d)

Z80 utasítások mnemonik szerint

Ebben a táblázatban a Z80 utasításokat mnemonik szerint, oktális és hexadecimális kódokkal együtt soroljuk fel. Az utolsó oszlopban az utasítások órajelciklusban mért végrehajtási ideje található. A feltételes utasításoknál a két végrehajtási idő közül az első a feltétel teljesülése, a második a feltétel nem teljesülése esetén érvényes.

Mnemonik	Oktális	Hexa	Végrehajtási idő
ADC A,(HL)	216	8E	7
ADC A,(IX+dis)	335 216 XXX	DD 8E XX	19
ADC A,(IY+dis)	375 216 XXX	FD 8E XX	19
ADC A,A	217	8F	4
ADC A,B	210	88	4
ADC A,C	211	89	4
ADC A,D	212	8A	4
ADC A,dd	316 dd	CE dd	7
ADC A,E	213	8B	4
ADC A,H	214	8C	4
ADC A,L	215	8D	4
ADC HL,BC	355 112	ED 4A	15
ADC HL,DE	355 132	ED 5A	15
ADC HL,HL	355 152	ED 6A	15
ADC HL,SP	355 172	ED 7A	15
ADD A,(HL)	206	86	7
ADD A,(IX+dis)	335 206 XXX	DD 86 XX	19
ADD A,(IY+dis)	375 206 XXX	FD 86 XX	19
ADD A,A	207	87	4
ADD A,B	200	80	4
ADD A,C	201	81	4
ADD A,D	202	82	4
ADD A,dd	306 ddd	C6 dd	7
ADD A,E	203	83	4
ADD A,H	204	84	4
ADD A,L	205	85	4
ADD HL,BC	011	09	11
ADD HL,DE	031	19	11
ADD HL,HL	051	29	11
ADD HL,SP	071	39	11
ADD IX,BC	335 011	DD 09	15
ADD IX,DE	335 031	DD 19	15
ADD IX,IX	335 051	DD 29	15

Mnemonik	Oktális	Hexa	Végrehajtási idő
ADD IX,SP	335 071	DD 93	15
ADD IY,BC	375 011	FD 09	15
ADD IY,DE	375 031	FD 19	15
ADD IY,IY	375 051	FD 29	15
ADD IY,SP	375 071	FD 39	15
AND (HL)	246	A6	7
AND (IX+dis)	335 246 XXX	DD A6 XX	19
AND (IY+dis)	375 246 XXX	FD A6 XX	19
AND A	247	A7	4
AND B	240	A0	4
AND C	241	A1	4
AND D	242	A2	4
AND dd	346 ddd	E6 dd	7
AND E	243	A3	4
AND H	244	A4	4
AND L	245	A5	4
BIT 0,(HL)	313 106	CB 46	12
BIT 0,(IX+dis)	335 313 XXX 106	DD CB XX 46	20
BIT 0,(IY+dis)	375 313 XXX 106	FD CB XX 46	20
BIT 0,A	313 170	CB 47	8
BIT 0,B	313 100	CB 40	8
BIT 0,C	313 101	CB 41	8
BIT 0,D	313 102	CB 42	8
BIT 0,E	313 103	CB 43	8
BIT 0,H	313 104	CB 44	8
BIT 0,L	313 105	CB 45	8
BIT 1,(HL)	313 116	CB 4E	12
BIT 1,(IX+dis)	335 313 XXX 116	DD CB XX 4E	20
BIT 1,(IY+dis)	375 313 XXX 116	FD CB XX 4E	20
BIT 1,A	313 117	CB 4F	8
BIT 1,B	313 110	CB 48	8
BIT 1,C	313 111	CB 49	8
BIT 1,D	313 112	CB 4A	8
BIT 1,E	313 113	CB 4B	8
BIT 1,H	313 114	CB 4C	8
BIT 1,L	313 115	CB 4D	8
BIT 2,(HL)	313 126	CB 56	12
BIT 2,(IX+dis)	335 313 XXX 126	DD CB XX 56	20
BIT 2,(IY+dis)	375 313 XXX 126	FD CB XX 56	20
BIT 2,A	313 127	CB 57	8
BIT 2,B	313 120	CB 50	8
BIT 2,C	313 121	CB 51	8
BIT 2,D	313 122	CB 52	8
BIT 2,E	313 123	CB 53	8

Mnemonik	Oktális	Hexa	Végrehajtási idő
BIT 2,H	313 124	CB 54	8
BIT 2,L	313 125	CB 55	8
BIT 3,(HL)	313 136	CB 5E	12
BIT 3,(IX+dis)	335 313 XXX 136	DD CB XX 5E	20
BIT 3,(IY+dis)	375 313 XXX 136	FD CB XX 5E	20
BIT 3,A	313 137	CB 5F	8
BIT 3,B	313 130	CB 58	8
BIT 3,C	313 131	CB 59	8
BIT 3,D	313 132	CB 5A	8
BIT 3,E	313 133	CB 5B	8
BIT 3,H	313 134	CB 5C	8
BIT 3,L	313 135	CB 5D	8
BIT 4,(HL)	313 146	CB 66	12
BIT 4,(IX+dis)	335 313 XXX 146	DD CB XX 66	20
BIT 4,(IY+dis)	375 313 XXX 146	FD CB XX 66	20
BIT 4,A	313 147	CB 67	8
BIT 4,B	313 140	CB 60	8
BIT 4,C	313 141	CB 61	8
BIT 4,D	313 142	CB 62	8
BIT 4,E	313 143	CB 63	8
BIT 4,H	313 144	CB 64	8
BIT 4,L	313 145	CB 65	8
BIT 5,(HL)	313 156	CB 6E	12
BIT 5,(IX+dis)	335 313 XXX 156	DD CB XX 6E	20
BIT 5,(IY+dis)	375 313 XXX 156	FD CB XX 6E	20
BIT 5,A	313 157	CB 6F	8
BIT 5,B	313 150	CB 68	8
BIT 5,C	313 151	CB 69	8
BIT 5,D	313 152	CB 6A	8
BIT 5,E	313 153	CB 6B	8
BIT 5,H	313 154	CB 6C	8
BIT 5,L	313 155	CB 6D	8
BIT 6,(HL)	313 166	CB 76	12
BIT 6,(IX+dis)	335 313 XXX 166	DD CB XX 76	20
BIT 6,(IY+dis)	375 313 XXX 166	FD CB XX 76	20
BIT 6,A	313 167	CB 77	8
BIT 6,B	313 160	CB 70	8
BIT 6,C	313 161	CB 71	8
BIT 6,D	313 162	CB 72	8
BIT 6,E	313 163	CB 73	8
BIT 6,H	313 164	CB 74	8
BIT 6,L	313 165	CB 75	8
BIT 7,(HL)	313 176	CB 7E	12
BIT 7,(IX+dis)	335 313 XXX 176	DD CB XX 7E	20

Mnemonik	Oktális	Hexa	Végrehajtási idő
BIT 7,(IY+dis)	375 313 XXX 176	FD CB XX 7E	20
BIT 7,A	313 177	CB 7F	8
BIT 7,B	313 170	CB 78	8
BIT 7,C	313 171	CB 79	8
BIT 7,D	313 172	CB 7A	8
BIT 7,E	313 173	CB 7B	8
BIT 7,H	313 174	CB 7C	8
BIT 7,L	313 175	CB 7D	8
CALL ADDR	315 XXX XXX	CD XX XX	17
CALL C,ADDR	334 XXX XXX	DC XX XX	17/10
CALL M,ADDR	374 XXX XXX	FC XX XX	17/10
CALL NC,ADDR	324 XXX XXX	D4 XX XX	17/10
CALL NZ,ADDR	304 XXX XXX	C4 XX XX	17/10
CALL P,ADDR	364 XXX XXX	F4 XX XX	17/10
CALL PE,ADDR	354 XXX XXX	EC XX XX	17/10
CALL PO,ADDR	344 XXX XXX	E4 XX XX	17/10
CALL Z,ADDR	314 XXX XXX	CC XX XX	17/10
CCF	077	3F	4
CP (HL)	276	BE	7
CP (IX+dis)	335 276 XXX	DD BE XX	19
CP (IY+dis)	375 276 XXX	FD BE XX	19
CP A	277	BF	4
CP B	270	B8	4
CP C	271	B9	4
CP D	272	BA	4
CP dd	376 ddd	FE dd	7
CP E	273	BB	4
CP H	274	BC	4
CP L	275	BD	4
CPD	355 251	ED A9	16
CPDR	355 271	ED B9	21/16
CPI	355 241	ED A1	16
CPIR	355 261	ED B1	21/16
CPL	057	2F	4
DAA	047	27	4
DEC (HL)	065	35	11
DEC (IX+dis)	335 065 XXX	DD 35 XX	23
DEC (IY+dis)	375 065 XXX	FD 35 XX	23
DEC A	075	3D	4
DEC B	005	05	4
DEC BC	013	0B	6
DEC C	015	0D	4
DEC D	025	15	4

Mnemonik	Oktális	Hexa	Végrehajtási idő
DEC DE	033	1B	6
DEC E	035	1D	4
DEC H	045	25	4
DEC HL	053	2B	6
DEC IX	335 053	DD 2B	10
DEC IY	375 053	FD 2B	10
DEC L	055	2D	4
DEC SP	073	3B	6
DI	363	F3	4
DJNZ,dis	020 XXX	10 XX	13/8
EI	373	FB	4
EX (SP),HL	343	E3	19
EX (SP),IX	335 343	DD E3	23
EX (SP),IY	375 343	FD E3	23
EX AF,AF'	010	08	4
EX DE,HL	353	EB	4
EXX	331	D9	4
HLT	166	76	4
IM 0	355 106	ED 46	8
IM 1	355 126	ED 56	8
IM 2	355 136	ED 5E	8
IN A,(C)	355 170	ED 78	12
IN A,port	333 XXX	DB XX	11
IN B,(C)	355 100	ED 40	12
IN C,(C)	355 110	ED 84	12
IN D,(C)	355 120	ED 50	12
IN E,(C)	355 130	ED 58	12
IN H,(C)	355 140	ED 60	12
IN L,(C)	355 150	ED 68	12
INC (HL)	064	34	11
INC (IX+dis)	335 064 XXX	DD 34 XX	23
INC (IY+dis)	375 064 XXX	FD 34 XX	23
INC A	074	3C	4
INC B	004	04	4
INC BC	003	03	6
INC C	014	0C	4
INC D	024	14	4
INC DE	023	13	6
INC E	034	1C	4
INC H	044	24	4
INC HL	043	23	6
INC IX	335 043	DD 23	10
INC IY	375 043	FD 23	10

Mnemonik	Oktális	Hexa	Végrehajtási idő
INC L	054	2C	4
INC SP	063	33	6
IND	355 252	ED AA	16
INDR	355 272	ED BA	21/16
INI	355 242	ED A2	16
INIR	355 262	ED B2	21/16
JP (HL)	351	E9	4
JP (IX)	335 351	DD E9	8
JP (IY)	375 351	FD E9	8
JP ADDR	303 XXX XXX	C3 XX XX	10
JP C,ADDR	332 XXX XXX	DA XX XX	10
JP M,ADDR	372 XXX XXX	FA XX XX	10
JP NC,ADDR	322 XXX XXX	D2 XX XX	10
JP NZ,ADDR	302 XXX XXX	C2 XX XX	10
JP P,ADDR	362 XXX XXX	F2 XX XX	10
JP PE,ADDR	352 XXX XXX	EA XX XX	10
JP PO,ADDR	342 XXX XXX	E2 XX XX	10
JP Z,ADDR	312 XXX XXX	CA XX XX	10
JR C,dis	070 XXX	38 XX	12
JR dis	030 XXX	18 XX	12/7
JR NC,dis	060 XXX	30 XX	12/7
JR NZ,dis	040 XXX	20 XX	12/7
JR Z,dis	050 XXX	28 XX	12/7
LD (ADDR),A	062 XXX XXX	32 XX XX	13
LD (ADDR),BC	355 103 XXX XXX	ED 43 XX XX	20
LD (ADDR),DE	355 123 XXX XXX	ED 53 XX XX	20
LD (ADDR),HL	355 143 XXX XXX	ED 63 XX XX	20
LD (ADDR),HL	042 XXX XXX	22 XX XX	16
LD (ADDR), IX	335 042 XXX XXX	DD 22 XX XX	20
LD (ADDR), IX	375 042 XXX XXX	FD 22 XX XX	20
LD (ADDR),SP	355 163 XXX XXX	ED 73 XX XX	20
LD (BC),A	002	02	7
LD (DE),A	022	12	7
LD (HL),A	167	77	7
LD (HL),B	160	70	7
LD (HL),C	161	71	7
LD (HL),D	162	72	7
LD (HL),dd	066 ddd	36 dd	10
LD (HL),E	163	73	7
LD (HL),H	164	74	7
LD (HL), L	165	75	7
LD (IX+dis),A	335 167 XXX	DD 77 XX	19
LD (IX+dis),B	335 160 XXX	DD 70 XX	19
LD (IX+dis),C	335 161 XXX	DD 71 XX	19

Mnemonik	Oktális	Hexa	Végrehajtási idő
LD (IX+dis),D	335 162 XXX	DD 72 XX	19
LD (IX+dis),dd	335 066 XXX ddd	DD 36 XX dd	19
LD (IX+dis),E	335 163 XXX	DD 73 XX	19
LD (IX+dis),H	335 164 XXX	DD 74 XX	19
LD (IX+dis),L	335 165 XXX	DD 75 XX	19
LD (IY+dis),A	375 167 XXX	FD 77 XX	19
LD (IY+dis),B	375 160 XXX	FD 70 XX	19
LD (IY+dis),C	375 161 XXX	FD 71 XX	19
LD (IY+dis),D	375 162 XXX	FD 72 XX	19
LD (IY+dis),dd	375 066 XXX ddd	FD 36 XX dd	19
LD (IY+dis),E	375 163 XXX	FD 73 XX	19
LD (IY+dis),H	375 164 XXX	FD 74 XX	19
LD (IY+dis),L	375 165 XXX	FD 75 XX	19
LDA,(ADDR)	072 XXX XXX	3A XX XX	13
LD A,(BC)	012	0A	7
LD A,(DE)	032	1A	7
LD A,(HL)	176	7E	7
LD A,(IX+dis)	335 176 XXX	DD 7E XX	19
LD A,(IY+dis)	375 176 XXX	FD 7E XX	19
LD A,A	177	7F	4
LD A,B	170	78	4
LD A,C	171	79	4
LD A,D	172	7A	4
LD A,dd	076 ddd	3E dd	7
LD A,E	173	7B	4
LD A,H	174	7C	4
LD A,I	355 127	ED 57	9
LD A,L	175	7D	4
LD A,R	355 137	ED 5F	9
LD B,(HL)	106	46	7
LD B,(IX+dis)	335 106 XXX	DD 46 XX	19
LD B,(IY+dis)	375 106 XXX	FD 46 XX	19
LD B,A	107	47	4
LD B,B	100	40	4
LD B,C	101	41	4
LD B,D	102	42	4
LD B,dd	006 ddd	06 dd	7
LD B,E	103	43	4
LD B,H	104	44	4
LD B,L	105	45	4
LD BC,(ADDR)	355 113 XXX XXX	ED 4B XX XX	20
LD BC,dddd	001 ddd ddd	01 dd dd	10
LD C,(HL)	116	4E	7
LD C,(IX+dis)	335 116 XXX	DD 4E XX	19

Mnemonik	Oktális	Hexa	Végrehajtási idő
LD C,(IY+dis)	375 116 XXX	FD 4E XX	19
LD C,A	117	4F	4
LD C,B	110	48	4
LD C,C	111	49	4
LD C,D	112	4A	4
LD C,dd	016 ddd	0E dd	7
LD C,E	113	4B	4
LD C,H	114	4C	4
LD C,L	115	4D	4
LD D,(HL)	126	56	7
LD D,(IX+dis)	335 126 XXX	DD 56 XX	19
LD D,(IY+dis)	375 126 XXX	FD 56 XX	19
LD D,A	127	57	4
LD D,B	120	50	4
LD D,C	121	51	4
LD D,D	122	52	4
LD D,dd	026 ddd	16 dd	7
LD D,E	123	53	4
LD D,H	124	54	4
LD D,L	125	55	4
LD DE,(ADDR)	355 133 XXX XXX	ED 5B XX XX	20
LD DE,dddd	021 ddd ddd	11 dd dd	10
LD E,(HL)	136	5E	7
LD E,(IX+dis)	335 136 XXX	DD 5E XX	19
LD E,(IY+dis)	375 136 XXX	FD 5E XX	19
LD E,A	137	5F	4
LD E,B	130	58	4
LD E,C	131	59	4
LD E,D	132	5A	4
LD E,dd	036 ddd	1E dd	7
LD E,E	133	5B	4
LD E,H	134	5C	4
LD E,L	135	5D	4
LD H,(HL)	146	66	7
LD H,(IX+dis)	335 146 XXX	DD 66 XX	19
LD H,(IY+dis)	375 146 XXX	FD 66 XX	19
LD H,A	147	67	4
LD H,B	140	60	4
LD H,C	141	61	4
LD H,D	142	62	4
LD H,dd	046 ddd	26 dd	7
LD H,E	143	63	4
LD H,H	144	64	4
LD H,L	145	65	4

Mnemonik	Oktális	Hexa	Végrehajtási idő
LD HL,(ADDR)	355 153 XXX XXX	ED 6B XX XX	20
LD HL,(ADDR)	052 XXX XXX	2A XX XX	16
LD HL,dddd	041 ddd ddd	21 dd dd	10
LD I,A	355 170	ED 47	9
LD IX,(ADDR)	335 052 XXX XXX	DD 2A XX XX	20
LD IX,dddd	335 041 ddd ddd	DD 21 dd dd	14
LD IY,(ADDR)	375 052 XXX XXX	FD 2A XX XX	20
LD IY,dddd	375 041 ddd ddd	FD 21 dd dd	14
LD L,(HL)	156	6E	7
LD L,(IX+dis)	335 156 XXX	DD 6E XX	19
LD L,(IY+dis)	375 156 XXX	FD 6E XX	19
LD L,A	157	6F	4
LD L,B	150	68	4
LD L,C	151	69	4
LD L,D	152	6A	4
LD L,dd	056 ddd	2E dd	7
LD L,E	153	6B	4
LD L,H	154	6C	4
LD L,L	155	6D	4
LD R,A	355 117	ED 4F	9
LD SP,(ADDR)	355 173 XXX XXX	ED 7B XX XX	20
LD SP,dddd	061 ddd ddd	31 dd dd	10
LD SP,HL	371	F9	6
LD SP,IX	335 371	DD F9	10
LD SP,IY	375 371	FD F9	10
LDD	355 250	ED A8	16
LDDR	355 270	ED B8	21/16
LDI	355 240	ED A0	16
LDIR	355 260	ED B0	21/16
NEG	355 104	ED 44	8
NOP	000	00	4
OR (HL)	266	B6	7
OR (IX+dis)	355 266 XXX	DD B6 XX	19
OR (IY+dis)	375 266 XXX	FD B6 XX	19
OR A	267	B7	4
OR B	260	B0	4
OR C	261	B1	4
OR D	262	B2	4
OR dd	366 ddd	F6 dd	7
OR E	263	B3	4
OR H	264	B4	4
OR L	265	B5	4
OTDR	355 273	ED BB	21/16
OTIR	355 263	ED B3	21/16

Mnemonik	Oktális	Hexa	Végrehajtási idő
OUT (C),A	355 171	ED 79	12
OUT (C),B	355 101	ED 41	12
OUT (C),C	355 111	ED 49	12
OUT (C),D	355 121	ED 51	12
OUT (C),E	355 131	ED 59	12
OUT (C),H	355 141	ED 61	12
OUT(C),L	355 151	ED 69	12
OUT port,A	323 port	D3 port	11
OUTD	355 253	ED AB	16
OUTI	355 243	ED A3	16
POP AF	361	F1	10
POP BC	301	C1	10
POP DE	321	D1	10
POP HL	341	E1	10
POP IX	335 341	DD E1	14
POP IY	375 341	FD E1	14
PUSH AF	365	F5	11
PUSH BC	305	C5	11
PUSH DE	325	D5	11
PUSH HL	345	E5	11
PUSH IX	335 345	DD E5	15
PUSH IY	375 345	FD E5	15
RES 0,(HL)	313 206	CB 86	15
RES 0,(IX+dis)	335 313 XXX 206	DD CB XX 86	23
RES 0,(IY+dis)	375 313 XXX 206	FD CB XX 86	23
RES 0,A	313 207	CB 87	8
RES 0,B	313 200	CB 80	8
RES 0,C	313 201	CB 81	8
RES 0,D	313 202	CB 82	8
RES 0,E	313 203	CB 83	8
RES 0,H	313 204	CB 84	8
RES 0,L	313 205	CB 85	8
RES 1,(HL)	313 216	CB 8E	15
RES 1,(IX+dis)	335 313 XXX 216	DD CB XX 8E	23
RES 1,(IY+dis)	375 313 XXX 216	FD CB XX 8E	23
RES 1,A	313 217	CB 8F	8
RES 1,B	313 210	CB 88	8
RES 1,C	313 211	CB 89	8
RES 1,D	313 212	CB 8A	8
RES 1,E	313 213	CB 8B	8
RES 1,H	313 214	CB 8C	8
RES 1,L	313 215	CB 8D	8
RES 2,(HL)	313 226	CB 96	15
RES 2,(IX+dis)	335 313 XXX 226	DD CB XX 96	23

Mnemonik	Oktális	Hexa	Végrehajtási idő
RES 2,(IY+dis)	375 313 XXX 226	FD CB XX 96	23
RES 2,A	313 227	CB 97	8
RES 2,B	313 220	CB 90	8
RES 2,C	313 221	CB 91	8
RES 2,D	313 222	CB 92	8
RSE 2,E	313 223	CB 93	8
RES 2,H	313 224	CB 94	8
RES 2,L	313,225	CB 95	8
RES 3,(HL)	313 236	CB 9E	15
RES 3,(IX+dis)	335 313 XXX 236	DD CB XX 9E	23
RES 3,(IY+dis)	375 313 XXX 236	FD CB XX 9E	23
RES 3,A	313 237	CB 9F	8
RES 3,B	313 230	CB 98	8
RES 3,C	313 231	CB 99	8
RES 3,D	313 232	CB 9A	8
RES 3,E	313 233	CB 9B	8
RES 3,H	313 234	CB 9C	8
RES 3,L	313 235	CB 9D	8
RES 4,(HL)	313 246	CB A6	15
RES 4,(IX+dis)	335 313 XXX 246	DD CB XX A6	23
RES 4,(IY+dis)	375 313 XXX 246	FD CB XX A6	23
RES 4,A	313 247	CB A7	8
RES 4,B	313 240	CB A0	8
RES 4,C	313 241	CB A1	8
RES 4,D	313 242	CB A2	8
RES 4,E	313 243	CB A3	8
RES 4,H	313 244	CB A4	8
RES 4,L	313 245	CB A5	8
RES 5,(HL)	313 256	CB AE	15
RES 5,(IX+dis)	335 313 XXX 256	DD CB XX AE	23
RES 5,(IY+dis)	375 313 XXX 256	FD CB XX AE	23
RES 5,A	313 257	CB AF	8
RES 5,B	313 250	CB A8	8
RES 5,C	313 251	CB A9	8
RES 5,D	313 252	CB AA	8
RES 5,E	313 253	CB AB	8
RES 5,H	313 254	CB AC	8
RES 5,L	313 255	CB AD	8
RES 6,(HL)	313 266	CB B6	15
RES 6,(IX+dis)	335 313 XXX 266	DD CB XX B6	23
RES 6,(IY+dis)	375 313 XXX 266	FD CB XX B6	23
RES 6,A	313 267	CB B7	8
RES 6,B	313 260	CB B0	8
RES 6,C	313 261	CB B1	8

Mnemonik	Oktális	Hexa	Végrehajtási idő
RES 6,D	313 262	CB B2	8
RES 6,E	313 263	CB B3	8
RES 6,H	313 264	CB B4	8
RES 6,L	313 265	CB B5	8
RES 7,(HL)	313 276	CB BE	15
RES 7,(IX+dis)	335 313 XXX 276	DD CB XX BE	23
RES 7,(IY+dis)	375 313 XXX 276	FD CB XX BE	23
RES 7,A	313 277	CB BF	8
RES 7,B	313 270	CB B8	8
RES 7,C	313 271	CB B9	8
RES 7,D	313 272	CB BA	8
RES 7,E	313 273	CB BB	8
RES 7,H	313 274	CB BC	8
RES 7,L	313 275	CB BD	8
RET	311	C9	10
RET C	330	D8	11/5
RET M	370	F8	11/5
RET NC	320	D0	11/5
RET NZ	300	C0	11/5
RET P	360	F0	11/5
RET PE	350	E8	11/5
RET PO	340	E0	11/5
RET Z	310	C8	11/5
RETI	355 215	ED 4D	14
RETN	355 105	ED 45	14
RL (HL)	313 026	CB 16	15
RL (IX+dis)	335 313 XXX 026	DD CB XX 16	23
RL (IY+dis)	375 313 XXX 026	FD CB XX 16	23
RL A	313 027	CB 17	8
RL B	313 020	CB 10	8
RL C	313 021	CB 11	8
RL D	313 022	CB 12	8
RL E	313 023	CB 13	8
RL H	313 024	CB 14	8
RL L	313 025	DB 15	8
RLA	027	17	4
RLC (HL)	313 006	CB 06	15
RLC (IX+dis)	335 313 XXX 006	DD CB XX 06	23
RLC (IY+dis)	375 313 XXX 006	FD CB XX 06	23
RLC A	313 007	CB 07	8
RLC B	313 000	CB 00	8
RLC C	313 001	CB 01	8
RLC D	313 002	CB 02	8
RLC E	313 003	CB 03	8

Mnemonik	Oktális	Hexa	Végrehajtási idő
RLC H	313 004	CB 04	8
RLC L	313 005	CB 05	8
RLCA	007	07	4
RLD	355 157	ED 6F	18
RR (HL)	313 036	CB 1E	15
RR (IX+dis)	335 313 XXX 036	DD CB XX 1E	23
RR (IY+dis)	375 313 XXX 036	FD CB XX 1E	23
RR A	313 037	CB 1F	8
RR B	313 030	CB 18	8
RR C	313 031	CB 19	8
RR D	313 032	CB 1A	8
RR E	313 033	CB 1B	8
RR H	313 034	CB 1C	8
RR L	313 035	CB 1D	8
RRA	037	1F	4
RRC (HL)	313 016	CB 0E	15
RRC (IX+dis)	335 313 XXX 016	DD CB XX 0E	23
RRC (IY+dis)	375 313 XXX 016	FD CB XX 0E	23
RRC A	313 017	CB 0F	8
RRC B	313 010	CB 08	8
RRC C	313 011	CB 09	8
RRC D	313 012	CB 0A	8
RRC E	313 013	CB 0B	8
RRC H	313 014	CB 0C	8
RRC L	313 015	CB 0D	8
RRCA	017	0F	4
RRD	355 147	ED 67	18
RST 00	307	C7	11
RST 08	317	CF	11
RST 10	327	D7	11
RST 18	337	DF	11
RST 20	347	E7	11
RST 28	357	EF	11
RST 30	367	F7	11
RST 38	377	FF	11
SBC A,(HL)	236	9E	7
SBC A,(IX+dis)	335 236 XXX	DD 9E XX	19
SBC A,(IY+dis)	375 236 XXX	FD 9E XX	19
SBC A,A	237	9F	4
SBC A,B	230	98	4
SBC A,C	231	99	4
SBC A,D	232	9A	4
SBC A,ddd	336 ddd	DE dd	7
SBC A,E	233	9B	4

Mnemonik	Oktális	Hexa	Végrehajtási idő
SBC A,H	234	9C	4
SBC A,L	235	9D	4
SBC HL,BC	355 102	ED 42	15
SBC HL,DE	355 122	ED 52	15
SBC HL,HL	355 142	ED 62	15
SBC HL,SP	355 162	ED 72	15
SCF	067	37	4
SET 0,(HL)	313 306	CB C6	15
SET 0,(IX+dis)	335 313 XXX 306	DD CB XX C6	23
SET 0,(IY+dis)	375 313 XXX 306	FD CB XX C6	23
SET 0,A	313 307	CB C7	8
SET 0,B	313 300	CB C0	8
SET 0,C	313 301	CB C1	8
SET 0,D	313 302	CB C2	8
SET 0,E	313 303	CB C3	8
SET 0,H	313 304	CB C4	8
SET 0,L	313 305	CB C5	8
SET 1,(HL)	313 316	CB CE	15
SET 1,(IX+dis)	335 313 XXX 316	DD CB XX DE	23
SET 1,(IY+dis)	375 313 XXX 316	FD CB XX CE	23
SET 1,A	313 317	CB CF	8
SET 1,B	313 310	CB C8	8
SET 1,C	313 311	CB C9	8
SET 1,D	313 312	CB CA	8
SET 1,E	313 313	CB CB	8
SET 1,H	313 314	CB CC	8
SET 1,L	313 315	CB CD	8
SET 2,(HL)	313 326	CB D6	15
SET 2,(IX+dis)	335 313 XXX 326	DD CB XX D6	23
SET 2,(IY+dis)	375 313 XXX 326	FD CB XX D6	23
SET 2,A	313 327	CB D7	8
SET 2,B	313 320	CB D0	8
SET 2,C	313 321	CB D1	8
SET 2,D	313 322	CB D2	8
SET 2,E	313 323	CB D3	8
SET 2,H	313 324	CB D4	8
SET 2,L	313 325	CB D5	8
SET 3,(HL)	313 336	CB DE	15
SET 3,(IX+dis)	335 313 XXX 336	DD CB XX DE	23
SET 3,(IY+dis)	375 313 XXX 336	FD CB XX DE	23
SET 3,A	313 337	CB DF	8
SET 3,B	313 330	CB D8	8
SET 3,C	313 331	CB D9	8
SET 3,D	313 332	CB DA	8

Mnemonik	Oktális	Hexa	Végrehajtási idő
SET 3,E	313 333	CB DB	8
SET 3,H	313 334	CB DC	8
SET 3,L	313 335	CB DD	8
SET 4,(HL)	313 346	CB E6	15
SET 4,(IX+dis)	335 313 XXX 346	DD CB XX E6	23
SET 4,(IY+dis)	375 313 XXX 346	FD CB XX E6	23
SET 4,A	313 347	CB E7	8
SET 4,B	313 340	CB E0	8
SET 4,C	313 341	CB E1	8
SET 4,D	313 342	CB E2	8
SET 4,E	313 343	CB E3	8
SET 4,H	313 344	CB E4	8
SET 4,L	313 345	CB E5	8
SET 5,(HL)	313 356	CB EE	15
SET 5,(IX+dis)	335 313 XXX 356	DD CB XX EE	23
SET 5,(IY+dis)	375 313 XXX 356	FD CB XX EE	23
SET 5,A	313 357	CB EF	8
SET 5,B	313 350	CB E8	8
SET 5,C	313 351	CB E9	8
SET 5,D	313 352	CB EA	8
SET 5,E	313 353	CB EB	8
SET 5,H	313 354	CB EC	8
SET 5,L	313 355	CB ED	8
SET 6,(HL)	313 366	CB F6	15
SET 6,(IX+dis)	335 313 XXX 366	DD CB XX F6	23
SET 6,(IY+dis)	375 313 XXX 366	FD CB XX F6	23
SET 6,A	313 367	CB F7	8
SET 6,B	313 360	CB F0	8
SET 6,C	313 361	CB F1	8
SET 6,D	313 362	CB F2	8
SET 6,E	313 363	CB F3	8
SET 6,H	313 364	CB F4	8
SET 6,L	313 365	CB F5	8
SET 7,(HL)	313 376	CB FE	15
SET 7,(IX+dis)	335 313 XXX 376	DD CB XX FE	23
SET 7,(IY+dis)	375 313 XXX 376	FD BC XX FE	23
SET 7,A	313 377	CB FF	8
SET 7,B	313 370	CB F8	8
SET 7,C	313 371	CB F9	8
SET 7,D	313 372	CB FA	8
SET 7,E	313 373	CB FB	8
SET 7,H	313 374	CB FC	8
SET 7,L	313 375	CB FD	8
SLA (HL)	313 046	CB 26	15

Mnemonik	Oktális	Hexa	Végrehajtási idő
SLA (IX+dis)	335 313 XXX 046	DD CB XX 26	23
SLA (IY+dis)	375 313 XXX 046	FD CB XX 26	23
SLA A	313 047	CB 27	8
SLA B	313 040	CB 20	8
SLA C	313 041	CB 21	8
SLA D	313 042	CB 22	8
SLA E	313 043	CB 23	8
SLA H	313 044	CB 24	8
SLA L	313 045	CB 25	8
SRA (HL)	313 056	CB 2E	15
SRA (IX+dis)	335 313 XXX 056	DD CB XX 2E	23
SRA (IY+dis)	375 313 XXX 056	FD CB XX 2E	23
SRA A	313 057	CB 2F	8
SRA B	313 050	CB 28	8
SRA C	313 051	CB 29	8
SRA D	313 052	CB 2A	8
SRA E	313 053	CB 2B	8
SRA H	313 054	CB 2C	8
SRA L	313 055	CB 2D	8
SRL (HL)	313 076	CB 3E	15
SRL (IX+dis)	335 313 XXX 076	DD CB XX 3E	23
SRL (IY+dis)	375 313 XXX 076	FD CB XX 3E	23
SRL A	313 077	CB 3F	8
SRL B	313 070	CB 38	8
SRL C	313 071	CB 39	8
SRL D	313 072	CB 3A	8
SRL E	313 073	CB 3B	8
SRL H	313 074	CB 3C	8
SRL L	313 075	CB 3D	8
SUB (HL)	226	96	7
SUB (IX+dis)	335 226 XXX	DD 96 XX	19
SUB (IY+dis)	375 226 XXX	FD 96 XX	19
SUB A	227	97	4
SUB B	220	90	4
SUB C	221	91	4
SUB D	222	92	4
SUB ddd	326 ddd	D6 dd	7
SUB E	223	93	4
SUB H	224	94	4
SUB L	225	95	4
XOR (HL)	256	AE	7
XOR (IX+dis)	335 256 XXX	DD AE XX	19
XOR (IY+dis)	375 256 XXX	FD AE XX	19
XOR A	257	AF	4

Mnemonik	Oktális	Hexa	Végrehajtási idő
XOR B	250	A8	4
XOR C	251	A9	4
XOR D	252	AA	4
XOR dd	356 ddd	EE dd	7
XOR E	253	AB	4
XOR H	254	AC	4
XOR L	255	AD	4

Jelzőbitek

Utasítás	bit 7 S	6 Z	5	4 H	3	2 P/V	1 N	0 C	Megjegyzés
ADD, ADC (8-bites)	↕	↕		↕		V	0	↕	
SUB, SBC, CP, NEG (8-bites)	↕	↕		↕		V	1	↕	
AND	↕	↕		1		P	0	0	
OR, XOR	↕	↕		0		P	0	0	
INC	↕	↕		↕		V	0	=	
DEC	↕	↕		↕		V	1	=	
ADD (16 bites)	=	=		X		=	0	↕	
ADC (16 bites)	↕	↕		X		V	0	↕	
SBC (16 bites)	↕	↕		X		V	1	↕	
RLA,RLCA,RRA, RRCA	=	=		0		=	0	↕	
RL,RLC,RR,RRC, SLA,SRA,SRL	↕	↕		0		P	0	↕	
RLD, RRD	↕	↕		0		P	0	=	
DAA	↕	↕		↕		P	=	↕	
CPL	=	=		1		=	1	=	
SCF	=	=		0		=	0	1	
CCF	=	=		X		=	0	↕	
IN	↕	↕		0		P	0	=	
INI,IND,OUTI,OUTD, INIR.INDR,OTIR, OTDR	X	↕		X		X	1	X	Z=0, ha B≠ 0, egyébként Z=1
LDI,LDD	X	X		0		↕	0	=	P/V=1, ha BC≠0, egyébként P/V=0
LDIR,LDDR	X	X		0		0	0	=	
CPI,CPIR,CPD,CPDR	↕	↕		↕		↕	1	=	Z=1, ha A=(HL), egyébként Z=0 P/V=1, ha BC≠0,egyébként P/V=0
LD A,I; LD A,R	↕	↕				0 IFF		=	P/V-be a megszakítás flip-flop tartalmát írja
BIT	X	↕		1		X	0	=	Z-be a vizsgált bitet másolja

Jelmagyarázat

=	a jelzőbit változatlan marad
0	a művelet törli a jelzőbitet
1	a művelet 1-re állítja a jelzőbitet
X	a jelzőbitnek nincs szerepe
P	P/V paritást jelez
V	P/V túlcsordulást jelez
↕	a művelet állítja a jelzőbitet

B Függelék

ROM-rutinok

Ebben a táblázatban a LEVEL II. Basic értelmező fontosabb ROM-rutinjainak belépési címe (hexadecimális és decimális) és rövid leírása található. A jobb oldali oszlopokban az általános regiszterek tartalmának változását jelezzük (xx=változik, hexadecimális szám=rögzített érték, pntr=mutató). A nagybetűs ROM-rutin nevek megegyeznek a 11., 12. fejezetekben szereplő nevekkel.

hexa	dec	leírás	A F	B C	D E	H L
0000	00000	RST 00H BASIC hidegindítás	xxxx	xxxx	xxxx	xxxx
0008	00008	RST 08H (JP 1C96H) a következő byte és a HL által mutatott byte összehasonlítása	xxxx			pntr
0010	00016	RSH 10H (JP 1D78H) programszöveg-elemzés	xxxx			pntr
0018	00024	RST 18H (JP 1C90H) HL és DE összehasonlítása	xxxx			
0020	00032	RST 20H (JP 25D9H) szoftver-akkumulátor tartalma, típusának meghatározása	xxxx			
0028	00040	RST 28H kazettás rendszerben BREAK vizsgálat (RET), lemezes rendszerben belépési pont (JP 4BA2H)				
002B	00043	KBDSCN belépési pont, billentyűzetvizsgálat, a megnyomott billentyű ASCII-kódja vagy 00H az A regiszterben, DE=billentyűzet-ellenőrző blokk címe	xxxx		4015	
0030	00048	RST 30H kazettás rendszerben nem használatos (RET), lemezes rendszerben belépési pont (JP 44B4H)				
0033	00051	VDCHR belépési pont, A regiszter tartalmának kiírása, DE=video ellenőrzőblokk címe	xx		401D	

hexa	dec	leírás	A F	B C	D E	H L
0038	00056	RST 38H kazettás rendszerben nem használatos; lemezes rendszerben megszakítások belépési pontja (JP 4518H)				
003B	00059	PRBYTE belépési pont, A regiszter tartalmának kinyomtatása, DE=nyomtató ellenőrző blokk címe	xxxx		4025	
0049	00073	GETCHR billentyűzetvizsgálat, mint KBDSCN de várakozik, amíg adat érkezik	xxxx		4015	
0050	00080	Billentyűzetdekódoló rutin (03E3H) táblázata, NEWLINE, CLEAR, BREAK, felfelé nyíl, lefelé nyíl, balra nyíl, jobbra nyíl és space ASCII kódjai				
0060	00096	Várakozás BC tartalmától függően((BC−1)*26+17)*2.255 időciklusig, kb. BC*14.6 μs-ig	0044	0000		
0066	00102	Nem maszkolható megszakítást kezelő rutin (RESET belépési pont), kazettás rendszerben melegindítás, lemezes rendszerben hidegindítás	xxxx	xxxx	xxxx	xxxx
0075	00117	Rendszer-inicializálás (RST-vektorok betöltése, periféria-ellenőrző táblák betöltése, be- és kimeneti puffer kijelölése, lemezes BASIC parancsok, READY? kérdés, tárvizsgálat, tárméret-beállítás, változótáblák és mutatók inicializálása, READY üzenet)	xxxx	xxxx	xxxx	xxxx
0105	00261	"READY" szöveg				
010B	00267	Üres terület (0DH byte-ok)				
012D	00301	SN-Error hibaüzenet (L3-Error)				
0132	00306	POINT, SET, RESET grafikus parancsok	xx			
0132	00306	POINT belépési pont	00			
0135	00309	SET belépési pont	80			
0138	00312	RESET belépési pont	01			
0150	00340	POINT, SET, RESET végrehajtása				
019D	00413	INKEY$ utasítás				
01BC	00444	READY szöveg SA-ba				xxxx
01C9	00457	CLS utasítás, képernyőtörlés	1Fxx			

hexa	dec	leírás	A F	B C	D E	H L
01D3	00467	RANDOM rutin, R regiszterből vett véletlen érték	xx			
01D9	00473	Egy bit írása, a kazettára, OUT(0FFH),01; OUT(0FFH),02; OUT(0FFH),00; teljes impulzus ideje: .836 ms	xxxx	00		FC00
01F8	00504	CASOFF a kazettaegység motorjának kikapcsolása	xxxx			
01FE	00510	Kazettaegység számának meghatározása	xxxx	xxxx	xxxx	pntr
0212	00530	DEFCAS kazettaegység kiválasztása, motor bekapcsolása	xxxx			
022C	00556	BLSTAR jobb oldali csillagot villogtató rutin				
0235	00565	RDBYTE kazettáról egy byte olvasása az A regiszterbe	xxxx			
0264	00612	WRBYTE kazettára egy byte kiírása				
0287	00647	WRHDR fejléc (256 nulla) és A5H szinkron byte kiírása	xxxx	xxxx	xxxx	pntr
0296	00662	RDHDR fejléc és A5H szinkron byte olvasása	2Axx	xxxx	xxxx	pntr
02A9	00681	SYSTEM program betöltésének befejezése (betöltési cím olvasása, motor ki)				
02B2	00690	SYSTEM parancs belépési pont				
02CE	00718	SYSTEM program beolvasása				
0314	00788	RDADDR két byte (cím) olvasása, betöltése HL-be	xxxx			xxxx
031D	00797	SYSTEM program indítása a file végén, ill. a *? kérdésre megadott címről	xxxx			xxxx
032A	00810	OUTCHR az A regiszter tartalmát a 409CH értékétől függően képernyőre (00H), kazettára (FFH) vagy nyomtatóra (01H) írja	xx			
033A	00826	VDCHR1 mint VDCHR, de DE-t a verembe tárolja, kurzorpozíciót (40A6H) állítja	xx			
0358	00856	KBDSC1 mint KBDSCN, de DE-t a verembe tárolja	xxxx			
0361	00865	INLINE billentyűzetről legfeljebb 240 karaktert olvas	xxxx		401D	xxxx

hexa	dec	leírás	A F	B C	D E	H L
0384	00900	KBDSC1 billentyűzet olvasó rutin meghívása	xxxx			
038B	00907	NEWLINE karakter kiírása (sor elején RET)	xxxx			
039C	00924	PRTCHR az A regiszter tartalmának kinyomtatása, vezérlőkarakterek kezelése, karakterszám növelése				
03C2	00962	Perifériakezelő rutinok belépési pontja, B=periféria típusa, DE= periféria-ellenőrző tábla címe				
03E3	00995	Billentyűzetvezérlő program, belépés csak a KBDSCN, GETCHR és KBDSC1 rutinokból				
0458	01112	Képernyővezérlő program, belépés a periféria-ellenőrző táblákból				
04B8	01208	Képernyővezérlő karakterek kiírása 04B8H kurzor be (0E) 04BDH kurzor ki (0F) 04C0H kurzor HOME (1C) 04CEH kurzor vissza (08) 04DAH balra nyíl (18) 04ECH jobbra nyíl (19) 04F6H 32 karakter/sor (17)				
0506	01286	Vezérlőkarakterek feldolgozása				
0541	01345	Karakter kiírása, szükség szerint SCROLL				
0573	01395	Törlés a sor végéig (1E)				
057C	01404	Törlés a képernyő végéig (1F)				
058D	01421	Nyomtató vezérlőprogram				
05D1	01489	PRSTAT nyomtató állapotjelző rutin	xxxx			
05D9	01497	Billentyűzetről olvasó rutin, HL=puffercím, B=pufferméret (INLINE is hívja)	xxxx	xxxx	401D	
0674	01652	Kazettás rendszer inicializálása (RST 00H)	xxxx	xxxx	xxxx	xxxx
06CC	01740	Inicializálás vége, BASIC melegindítás				
06D2	01746	RST-vektorok és perifériaellenőrző táblák, amelyek inicializáláskor a 4000H...4035H címekre kerülnek				

hexa	dec	leírás	A F	B C	D E	H L
		Aritmetikai rutinok				
0708	01800	Egyszeres pontosságú összeadás, belépési pontok: 0708H .5→ BCDE, BCDE+SA → SA 070BH (HL)+SA → SA 0710H (HL) → BCDE, –SA→ SA, BCDE+SA → SA 0713H BCDE–SA→ SA 0716H BCDE+SA → SA	xxxx	xxxx	xxxx	xxxx
0765	01893	Egész szám egyszeres pontosságúra alakítása				
07B7	01975	Mantissza-összeadás (HL)+CDE → CDE	xxxx	xxxx	xxxx	+2
07C3	01987	4-byte-os negatív egész szám kettes komplemens számmá alakítása	xxxx	xxxx	xxxx	
07D7	02007	CDE eltolása jobbra A bittel	xx	xxxx	xxxx	xx
07F8	02040	1.0 egyszeres pontosságú konstans				
07FC	02044	Hatványsorok és logaritmusszámítás egyszeres pontosságú konstansai: 07FDH 0.5988 0801H 0.96145 0805H 2.88539				
0809	02057	Logaritmusszámolás LOG(SA) → SA	xxxx	xxxx	xxxx	xxxx
0841	02113	SA * LOG(2) → SA	xxxx	xxxx	xxxx	xxxx
0847	02119	Egyszeres pontosságú szorzás SA * BCDE → SA	xxxx	xxxx	xxxx	xxxx
0897	02199	SA / 10 →SA	xxxx	xxxx	xxxx	xxxx
08A2	02211	Egyszeres pontosságú osztás BCDE / SA → SA	xxxx	xxxx	xxxx	xxxx
0907	02311	Egyszeres pontosságú szorzás kitevőjének és előjelének meghatározása				
093E	02366	SA * 10 → SA	xxxx	xxxx	xxxx	xxxx
0955	02389	SA előjelének vizsgálata	xxxx			
0964	02404	Egész szám egyszeres pontosságúra alakítása	xxxx	xxxx	xxxx	xxxx
0977	02423	ABS(SA) → SA	xxxx	xxxx	xxxx	xxxx
0983	02434	–SA → SA	xxxx			4123
098A	02442	SGN(SA) → SA	xxxx			xxxx

hexa	dec	leírás	A F	B C	D E	H L
0994	02452	SA adattípus és előjel meghatározás	xxxx			xxxx
09A4	02468	LDSASP, az SA tárolása a veremterületre, SA → (SP)			xxxx	
09B1	02481	LDHLSA, (HL) → BCDE → SA		xxxx	xxxx	+4
09B4	02484	LDRASA, BCDE → SA			xxxx	
09BF	02495	LDSARA, SA → BCDE		xxxx	xxxx	4125
09C2	02498	LDHLRA, (HL) → BCDE		xxxx	xxxx	+4
09CB	02507	LDSAHL, (SA) → (HL)	xxxx	00	4125	+4
09D2	02514	MVHLDE, (HL) → (DE)	xxxx	00	+4	+4
09D3	02515	MVDEHL, (DE) → (HL) (+4, ha egyszeres pontosságú)	xxxx	00	+4	+4
09D6	02518	MVA, (DE) → (HL), A byte mozgatása	xxxx	xxxx	xxxx	xxxx
09D7	02519	MVB, (DE) → (HL), B byte mozgatása	xxxx	xxxx	xxxx	xxxx
09DF	02527	Összeadás, kivonás előkészítése	xxxx	xx		4125
09F4	02548	MVSA1, SA1 → SA	xxxx	00	xxxx	xxxx
09F7	02551	MVHLSA, (HL) → SA	xxxx	00	xxxx	xxxx
09FC	02556	MVSA, SA → SA1	xxxx	00	xxxx	xxxx
0A03	02563	SA érték kezdőcímének meghatározása a típusjelző alapján	xxxx		xxxx	
0A0C	02572	SA és BCDE egyszeres pontosságú számok összehasonlítása	xxxx			xxxx
0A26	02598	0A0CH szubrutinja	xxxx	xxxx		xxxx
0A39	02617	HL és DE egész számok összehasonlítása	xxxx			
0A49	02633	SA és SA1 kétszeres pontosságú számok összehasonlítáas	xxxx	xxxx	xxxx	xxxx
04AF	02639	SA és (DE) kétszeres pontosságú számok összehasonlítása	xxxx	xxxx	xxxx	xxxx
0A7F	02687	CINT, INT(SA) → HL → SA	xxxx	xxxx	xxxx	xxxx
0A9A	02714	HLTOSA, HL → SA	02			
0A9D	02717	SETINT típusjelző-állítás 02 → (40AFH)	02			
0AB1	02737	CSGL, CSNG(SA) → SA	xxxx	xxxx	xxxx	xxxx
0ADB	02778	CDBL, CDBL(SA) → SA	xxxx	xxxx	xxxx	xxxx
0AEC	02796	SETDBL típusjelző-állítás 08 → (40AFH)	08			
0AEF	02799	SETSGL típusjelző-állítás 04 → (40AFH)	04			
0AF4	02804	Adattípus-vizsgálat, TM hiba, ha nem szövegváltozó	xxxx			
0AF6	02806	TM hiba				

hexa	dec	leírás	A F	B C	D E	H L
0AFB	02811	Egyszeres pontosságú szám egésszé alakítása	xxxx	xxxx	xxxx	xxxx
0B26	02853	FIX, FIX(SA) → SA	xxxx	xxxx	xxxx	xxxx
0B37	02871	INT, INT(SA) → SA	xxxx	xxxx	xxxx	xxxx
0B59	02905	Kétszeres pontosságú szám egésszé alakítása	xxxx	xxxx	xxxx	xxxx
0BA0	02976	INT szubrutinja				
0BAA	02986	DIM szubrutinja, tömbváltozó tárcímének kiszámítása				
0BC7	03015	SUBINT, HL–DE → HL	xxxx	xxxx	xxxx	xxxx
0BD2	03026	ADDINT, HL+DE → HL	xxxx	xxxx	xxxx	xxxx
0BF2	03058	MULINT, HL*DE → HL, túlcsorduláskor egyszeres pontosságra vált	xxxx	xxxx	xxxx	xxxx
0C5B	03163	Egész szám túlcsordulás-vizsgálata túlcsorduláskor egyszeres pontosságra vált	xx			
0C70	03184	SUBDBL, SA–SA1 → SA	xxxx	xxxx	xxxx	xxxx
0C77	03191	ADDDBL, SA+SA1 → SA	xxxx	xxxx	xxxx	xxxx
0D20	03360	SA+1 → SA, kétszeres pontosságú		xx		xxxx
0D33	03379	Kétszeres pontosságú mantisszák összeadása	xxxx	00	4121	412E
0D45	03397	Kétszeres pontosságú mantisszák kivonása	xxxx	00	4124	412E
0D57	03415	SA kétszeres pontosságú szám egyes komplemense	xxxx	xxxx		xxxx
0D69	03433	HL által mutatott kétszeres pontosságú szám A bittel jobbra eltolása	xxxx	xx	0000	
0D90	03476	HL által mutatott kétszeres pontosságú szám 1 bittel jobbra eltolása	xxxx		0000	4123
0DA1	03489	MULDBL, SA*SA1 → SA	xxxx	xxxx	xxxx	xxxx
0DD4	03540	10 kétszeres pontosságú konstans				
0DD8	03544	10 egyszeres pontosságú konstans				
0DDC	03548	SA/10 → SA, kétszeres pontosságú	xxxx	xxxx	xxxx	xxxx
0DE5	03557	DIVDBL, SA/SA1 → SA	xxxx	xxxx	xxxx	xxxx
0E4D	03661	SA*2 → SA kétszeres pontosságú	xxxx	xxxx	xxxx	xxxx

hexa	dec	leírás	A F	B C	D E	H L
		Adatkonverzió				
0E65	03685	ASCDBL, HL által mutatott szövegváltozó kétszeres pontosságú számmá alakítása, SA-ba helyezése	xxxx	xxxx	xxxx	pntr
0E6C	03692	ASCBIN, HL által mutatott szövegváltozó egész, egyszeres, kétszeres pontosságú számmá alakítása, SA-ba helyezése	xxxx	xxxx	xxxx	pntr
0F0A	03850	SA*10 → SA, egyszeres vagy kétszeres pontosságú	xxxx	xxxx	xxxx	xxxx
0F18	03864	SA/10 → SA, egyszeres vagy kétszeres pontosságú	xxxx	xxxx	xxxx	xxxx
0F29	03881					
0F89	03977	SA+A → SA egyszeres pontosságú	xxxx	xxxx	00xx	xxxx
0FA7	04007	"IN" és sorszám kiírása	xxxx	xxxx	00xx	xxxx
0FAF	04015	HL karakteres kiírása	xxxx	xxxx	00xx	xxxx
0FBD	04029	BINASC, SA szövegváltozóvá alakítva a 4130H pufferbe kerül, végén 00H	xxxx	xxxx	xxxx	xxxx
0FBE	04930	BINFOR, SA formázott szövegváltozóvá alakítva kerül a 4130H pufferbe, végén 00H	xxxx	xxxx	xxxx	xxxx
1364	04964	Konverziós konstansok 1364H 10^{10} kétszeres 136CH 10^{15} kétszeres 1376H 10^{16} kétszeres 137CH 0.5 kétszeres 1380H 0.5 egyszeres 1384H 10^{16} kétszeres 138AH 0.502778 egyszeres 138CH 100^{15} kétszeres mantisszája 1393H 10^{14} kétszeres mantisszája 139AH 10^{10} kétszeres mantisszája 13A1H 10^{12} kétszeres mantisszája 13A8H 10^{11} kétszeres mantisszája 13AFH 10^{10} kétszeres mantisszája				

hexa	dec	leírás	A F	B C	D E	H L
		13B6H 10^9 kétszeres mantisszája				
		13BDH 10^8 kétszeres mantisszája				
		13C4H 10^7 kétszeres mantisszája				
		13CBH 10^6 kétszeres mantisszája				
		13D2H 100000 egyszeres mantisszája				
		13D5H 100000 egyszeres mantisszája				
		13D8H 10000 egész				
		13DAH 1000 egész				
		13DCH 100 egész				
		13DEH 10 egész				
		Aritmetikai függvények				
13E2	05090	SQR és TAN szubrutinja, –SA → SA				xxxx
13E7	05095	SQR(SA) → SA, a 13F2H általános hatványozó szubrutin felhasználásával	xxxx	xxxx	xxxx	xxxx
13F2	05106	Hatványozás, BCDE ↑ SA → SA	xxxx	xxxx	xxxx	xxxx
1439	05177	e^x számolás, EXP(SA) → SA	xxxx	xxxx	xxxx	xxxx
1479	05241	e^x hatványsor konstansai				
149A	05274	Általános polinom helyettesítési értékét számoló szubrutinok,	xxxx	xxxx	xxxx	xxxx
		149AH páratlan kitevők				
		14A9H páros és páratlan kitevők				
		HL az együtthatók táblázatára mutat, eredmény SA-ban				
14C9	05321	RND, véletlenszám SA-ban	xxxx	xxxx	xxxx	xxxx
14F0	05360	RND(0), eredmény SA-ban	xxxx	xxxx	xxxx	xxxx
1541	05441	COS, COS(SA) → SA	xxxx	xxxx	xxxx	xxxx
1547	05447	SIN, SIN(SA) → SA	xxxx	xxxx	xxxx	xxxx
158B	05515	$\pi/2$ egyszeres konstans				
158F	05519	0.25 egyszeres konstans				
1593	05523	SIN hatványsor konstansai				
		1593H 05 tagok száma (egy byte)				
		1594H 39.7107				

hexa	dec	leírás	A F	B C	D E	H L
		1598H –76.575				
		159CH 81.6022				
		15A0H –41.3417				
		154AH 6.28319				
		mind egyszeres pontosságú				
		számok				
15A8	05544	TAN, SIN(SA)/COS(SA)→ SA	xxxx	xxxx	xxxx	xxxx
15BD	05565	ATN, ATN(SA) → SA	xxxx	xxxx	xxxx	xxxx
15E3	05603	ATN hatványsor konstansai				
		15E3H 09 tagok száma				
		(egy byte)				
		15E4H $2.86623*10^{-3}$				
		15E8H –0.0161657				
		15ECH 0.0429096				
		15F0H –0.0752896				
		15F4H 0.105586				
		15F8H –0.142089				
		15FCH 0.199936				
		1600H –0.333331				
		1604H 1.0000				

Táblázatok

1608	05640	BASIC függvények ugrótáblája
1650	05712	BASIC kulcsszavak listája kódok szerint rendezve, az első karakterek 7. bitje 1-re állítva
1822	06178	BASIC utasítások ugrótáblája
189A	06298	Műveletek prioritástáblázata
18A1	06305	Numerikus konvertáló rutinok ugrótáblája
18AB	06315	Aritmetikai műveletek ugrótáblája
18C9	06345	Hibakódok táblázata
18F7	06428	Osztás szubrutin, amely inicializáláskor a 4080H...408DH címekre kerül
1905	06442	Kezdeti értékek, tárcímek, amelyek inicializáláskor a 408EH... ...40A5H címekre kerülnek
191D	06429	Szövegek (sor vége jelekkel)
		191DH "ERROR"
		1924H "IN"
		1929H "READY"
		1930H "BREAK"

hexa	dec	leírás	A F	B C	D E	H L
		BASIC értelmező				
1936	06454	BASIC veremben index szerint kereső szubrutin				
1955	06485	Stringterületen helyet kereső, változót (általában szövegest) áthelyező szubrutin				
1963	06499	Stringterületen 2*C helyet kereső szubrutin, ha nincs hely, OM hiba	xxxx	00		
197A	06522	OM hiba				
197E	06526	NR hiba				
1991	06545	Hibakezelés (hibaüzenetek, ON ERROR, RESUME) 1991H SN hiba DATA sorban 1997H SN hiba 199AH /0 hiba 199DH NF hiba 19A0H RW hiba				
1A19	06681	BASIC főprogram belépési pont (kazetta ki, READY üzenet, szintaktikus hibánál EDIT hívása)				
1A3F	06719	Programbevitel AUTO módban				
1A76	06774	Programbevitel (">"üzenet, utasítások kódolása, új sor beillesztése, sormutatók állítása stb.)	xxxx	xxxx	xxxx	xxxx
1B10	06928	LIST/DELETE subrutin (argumentumok elemzése)	xxxx	xxxx	xxxx	pntr
1B2C	06956	SLINE adott számú sort kereső szubrutin (DE=sorszám)	xxxx	xxxx		xxxx
1B49	06985	NEW parancs (CLS, TROFF, AUTO törlése, 40A4H...40A5H mutató törlése, változók törlése)				
1B5D	07006	RUN parancs				
1B64	07012	RUN sorszám parancs				
1BB3	07091	QINPUT "?" kiírása, majd INLINE (JP 0361H)	xx			
1BC0	07104	Programszöveg-elemzés, utasítások kódolása	xxxx	xxxx	xxxx	xxxx
1C90	07312	RST 18H szubrutin (CPDEHL) HL és DE összehasonlítása	xxxx			
1C96	07318	RST 08H szubrutin, (HL) és az RST utasítás utáni byte összehasonlítása, egyenlőség esetén RST 10H hívása, egyébként SN hiba	xxxx			pntr

hexa	dec	leírás	A F	B C	D E	H L
1CA1	07329	FOR szubrutin (FOR...STEP...TO...NEXT keresés, elemzés)				
1D1E	07454	Programsor végrehajtásának folytatása				
1D5A	07514	BASIC program végrehajtás belépési pontja (utasításkód keresése, utasítást végrehajtó rutin meghívása, visszatérés 1D1EH-ra, minden végrehajtott utasítás szubrutin után BREAK, sor vége, program vége, TRON vizsgálat)				
1D78	07544	RST 10H szubrutin (RDLINE), programszöveg-elemzés, karaktervizsgálat	xxxx			pntr
1D91	07569	RESTORE szubrutin				
1D9B	07579	Billentyűzetvizsgálat (programvégrehajtás, LIST közben)				
1DA9	07593	STOP szubrutin				
1DAE	07598	END szubrutin (BREAK esetén belépés 1DB4H címen)				
1DE4	07652	CONT szubrutin				
1DF7	07671	TRON/TROFF szubrutin 1DF7H TRON belépési pont 1DF8H TROFF belépési pont				
1E00	07680	DEFSTR/DEFINT/DEFSNG/DEFDBL szubrutinok, a 4101H...411AH változótípus-táblát módosítják 1E00H DEFSTR belépési pont 1E03H DEFINT belépési pont 1E06H DEFSNG belépési pont 1E09H DEFDBL belépési pont				
1E3D	07741	Stringkezelő szubrutin (következő karakter alfabetikus-e)	xxxx			pntr
1E45	07749	Indexvizsgálat	xxxx	xxxx	xxxx	pntr
1E4A	07754	FC hiba				
1E4F	07759	Programszöveg numerikus karakterienek bináris számmá alakítása	xxxx	xxxx	xxxx	pntr
1E5A	07770	ASCINT szövegváltozó egész számmá alakítása	xxxx	xxxx	xxxx	pntr
1E7A	07802	CLEAR szubrutin				
1EA3	07843	RUN szubrutin (változók inicializálása és GOTO)				

hexa	dec	leírás	A F	B C	D E	H L
1EB1	07857	GOSUB szubrutin (BASIC verem és GOTO)				
1EC2	07874	GOTO szubrutin				
1ED9	07897	UL hiba				
1EDE	07909	RETURN szubrutin				
1F05	07941	DATA szubrutin				
1F07	07943	REM subrutin				
1F21	07969	LET szubrutin				
1F6C	08044	ON szubrutinok (ON, ON ERROR, ON...GOTO, ON...GOSUB)				
		1F70H ON ERROR belépési cím				
1FAF	08111	RESUME szubrutin				
1FF4	08180	ERROR szubrutin				
2008	08200	AUTO szubrutin				
2039	08249	IF...THEN...ELSE szubrutin				
2067	08295	PRINT szubrutinok				
		2067H LPRINT belépési pont				
		206FH PRINT belépési pont				
		(PRINT@,PRINT#, kazettakezelés, PRINT TAB,NEWLINE)				
2178	08568	"?REDO " szöveg				
217F	08575	INPUT és READ hibaüzenetek				
219A	08602	INPUT szubrutin (INPUT és INPUT#)				
21EF	08687	READ szubrutin				
2286	08838	"EXTRA IGNORED" szöveg				
2296	08854	Következő DATA keresése				
22B6	08886	NEXT szubrutin				
2335	09013	Aritmetikai kifejezések kiszámítása (zárójel, műveleti jel keresés, műveletek sorrendjének meghatározása, konverzió, műveletek végrehajtása, logikai műveletek, függvények kiszámítása, szövegváltozók összehasonlítása, következő műveleti jel keresése stb.)	xxxx	xxxx	xxxx	pntr
		2490H DIVINT HL/DE → SA	xxxx	xxxx	xxxx	xxxx
		24CFH ERR szubrutin				
		24DDH ERL szubrutin				
		24EBH VARPTR szubrutin				
		24FFH USR szubrutin				

hexa	dec	leírás	A	F	B	C	D	E	H	L
25D9	09689	RST 20H szubrutin (TESTSA) SA vizsgálata, eredmény:								

adat-típus	típus-jelző	Z	C	N	P/V	A-reg.
INT	02	NZ	C	N	E	–1
STR	03	Z	C	P	E	0
SNG	04	NZ	C	P	0	1
DBL	08	NZ	NC	P	E	5

hexa	dec	leírás	A	F	B	C	D	E	H	L
25E9	09705	Logikai műveletek kiszámítása								
2603	09731	Változótáblában keresés 2603H egyszerű változó 2608H tömbváltozó 260DH VARPTR szubrutin 269EH új változó 26D5H tömbváltozó keresése 273DH BS hiba	xxxx		xxxx		xxxx		pntr	
27C9	10185	MEM szubrutin								
27D4	10196	FRE szubrutin								
27F5	10229	POS szubrutin								
27FE	10238	USR szubrutin								
2819	10265	Aritmetikai műveletek eredményének konverziója	xxxx		xxxx		xxxx		xxxx	
2828	10280	INPUT parancs módban vizsgálata (ID hiba)	xxxx							
2836	10294	Stringterület-kezelés és stringfüggvények								
2857	10327	Stringterületen hely keresése	xx		xxxx		xxxx		40D3	
2865	10341	Szövegkonstans átvitele a stringterületre 28A1H ST hiba	xxxx		xxxx		40D6		pntr	
28A6	10406	Üzenetkiíró szubrutin 28A7H OUTLIN szubrutin	xxxx		xxxx		00xx		xxxx	
28BF	10639	Stringterületen rendelkezésre álló üres hely meghatározása, stringterület újraszervezése, új szövegváltozó bevitele	xxxx		xxxx		xxxx		xxxx	
29D7	10711	VAL, FRE és PRINT szubrutinok folytatása								
2A03	10755	LEN szubrutin LEN(SA) → SA → HL	xxxx		27F8		xxxx		xxxx	
2A0F	10767	ASC szubrutin ASC(SA) → SA → HL	xxxx		27F8		xxxx		xxxx	
2A1F	10783	CHR$ szubrutin								
2A2F	10799	STRING$ szubrutin								

hexa	dec	leírás	A F	B C	D E	H L
2A61	10849	LEFT$ szubrutin				
2A91	10897	RIGHT$ szubrutin				
2A9A	10906	MID$ szubrutin				
2AC5	10949	VAL szubrutin				
2ADF	10975	LEFT$, MID$, RIGHT$ zárójel kereső szubrutinja				
2AE7	10983	MID$ csak lemezes rendszerben lehet bal oldalon, kazettás rendszerben SN hiba				
2AEF	10991	INP szubrutin				
2AF8	11003	OUT szubrutin				
2B01	11009	Kifejezés értékének kiszámítása, egész eredmény DE-ben	xxxx	xxxx	xxxx	pntr
2B0E	11022	INP és OUT folytatása, portszám vizsgálata (FC hiba)	xxxx	xxxx	xxxx	pntr
2B29	11049	LIST szubrutin 2B29H LLIST belépési pont 2B2EH LIST belépési pont				
2BC6	11206	DELETE szubrutin				
2BF5	11253	CSAVE szubrutin				
2C1F	11295	CLOAD és CLOAD? szubrutin, betöltés (D=0), ill. összehasonlítása (D=FFH), file-név E-ben				
2CA5	11429	"BAD" szöveg				
2CAA	11434	PEEK szubrutin				
2CB1	11441	POKE szubrutin				
2CBD	11453	PRINT USING szubrutin				
2E53	11859	EDIT szubrutin 2E53H belépés SN hibából 2E60H belépés EDIT paranccsal				

Szövegszerkesztő parancsok

2F0AH	space	másolás
2F16H	K	törlés
2F1CH	S	keresés
2F40H	L	listázás
2F65H	C	csere
2F75H	H	törlés és hozzáírás
2F78H	X	hozzáírás
2F7DH	I	beszúrás
2FD2H	←	visszalépés

hexa	dec	leírás	A	F	B	C	D	E	H	L
		2FE0H NEWLINE változtatások megőrzése								
		2FE3H E eddigi változtatások megőrzése								
		2FF6H Q változtatások törlése és kilépés								
2FFF	12287	ROM vége								

RAM-táblázatok

Tárcím hexa	Tárcím dec	Tartalom
		RST-vektorok
4000	16384	RST 08H-vektor (JP 1C96H)
4003	16387	RST 10H-vektor (JP 1D78H)
4006	16390	RST 18H-vektor (JP 1C90H)
4009	16393	RST 20H-vektor (JP 25D9H)
400C	16396	RST 28H-vektor (RET), BREAK megnyomásakor INCH hívja
400F	16399	RST 30H-vektor (RET), lemezes rendszer használja
4012	16402	RST 38H-vektor (EI, RET), megszakításkezelő rutin hívására használható
		Periféria-ellenőrző blokkok
4015	16405	Billentyűzet-ellenőrző blokk (8 byte)
4015	16405	Periféria típusa (1)
4016	16406	Vezérlőprogram címe (03E3H)
4018	16408	Üres terület (3 byte)
401B	16411	RAM-puffercím (494BH)
401D	16413	Video-ellenőrző blokk (8 byte)
401D	16413	Periféria típusa (7)
401E	16414	Vezérlőprogram címe (0458H)
4020	16416	Kurzor címe (3C00H...3FFFH)
4022	16418	Kurzor elnyomás (00=nincs kurzor, egyébként a kurzor pozícióban álló karakterkódja)
4023	16419	DOS-puffercím (4F44H)
4025	16421	Nyomtató-ellenőrző blokk (8 byte)
4025	16421	Periféria típusa (6)
4026	16422	Vezérlőprogram címe (058DH)
4028	16424	Sorok száma egy lapon (43H=67)
4029	16425	Eddig kinyomtatott sorok száma
402A	16426	Üres terület (1 byte)
402B	16427	DOS-puffercím (5250H)
402D	16429	Lemezes rendszer SYS1 moduljának hívása (JP 4400H; LD A,A3H; RST 28H, kazettás rendszerben JP 5000H; RST 00H)
4033	16435	Perifériavezérlő riasztás (ellenőrző blokkban hibás típus, kazettás rendszerben LD A,0; RET, lemezes rendszerben JP 44BBH)
4036	16438	Billentyűzetvezérlő program munkaterülete

Tárcím hexa	Tárcím dec	Tartalom
403D	16445	Video-ellenőrző byte
403E	16446	Lemezes rendszer munkaterülete(403EH...407FH), kazettás rendszerben felhasználói terület, lemezes rendszerben idő (4041H...4043H), dátum(4044H...4046H), rendszer utility-programok betöltési címe (4047H...4048H),megszakítási állapotjelzők stb.
4080	16512	Osztás által használt kivonás-szubrutin, amely a rendszer indításakor az 18F7H...1904H címekről kerül ide
408E	16526	USR-szubrutin kezdőcíme (alapértelmezésben 1E4A=FC-hiba)
4090	16528	Véletlenszám-generátor munkaterülete
4093	16531	INP szubrutinja (IN A,(00); RET)
4096	16534	OUT szubrutinja (OUT (00),A; RET)
4099	16537	INKEY$-tárolómező, a BREAK után utoljára begépelt karaktert tárolja
409A	16538	Utolsó hibakód (RESUME jelző)
409B	16539	Karakterek száma (kurzor helyzete) az aktuális PRINT sorban
409C	16540	Output periféria kódja (00=video, 01=nyomtató, FF=kazetta)
409D	16541	Képernyő sormérete (40H)
409E	16542	Utolsó tabulátorpozíció
409F	16543	Üres terület (1 byte)
40A0	16544	Mutató a stringterület kezdetére (434CH)
40A2	16546	Aktuális sorszám (FFFEH)
40A4	16548	Mutató a programszöveg kezdetére (42E9H)
40A6	16550	Kurzor helyzete a sorban (a képernyőn)
40A7	16551	Mutató az be- és kimeneti puffer kezdetére (41E8H)
40A9	16553	Bemeneti periféria kódja (00=billentyűzet, egyébként kazetta)
40AA	16554	Véletlenszám-generátor kezdőérték (seed)
40AB	16555	R regiszterből vett érték (véletlenszám)
40AC	16556	Utolsó véletlenszám
40AE	16558	Változó keresési jelző (00=keresés, FF=új változó beírása)
40AF	16559	SA szoftverakkumulátor típusjelzője (02=egész, 03=szövegváltozó, 04=egyszeres, 08=kétszeres)
40B0	16560	Kifejezések kiszámításához közbülső érték tárolása
40B1	16561	Tárméret (BASIC számára fenntartott tár vége)
40B3	16563	Stringműveletek munkaterülete (literál stringmező, 2+30 byte)
40B3	16563	Mutató a stringműveletek munkaterületének következő szabad elemére
40B5	16565	Stringműveletek munkaterületének elemei (10 elem, mindegyike stringhosszúságot és stringcímet tartalmaz, 10*3 byte)

Tárcím hexa	Tárcím dec	Tartalom
40D3	16595	A stringterületre mozgatandó string hossza és címe (1+2 byte)
40D6	16598	Mutató a stringterület következő szabad byte-jára
40D8	16600	Az aktuális utasítás utolsó végrehajtott byte-jának indexe, PRINT USING utasításban formátumjelző
40DA	16602	Az utoljára olvasott DATA utasítás sorszáma
40DC	16604	Ciklusjelző (01=FOR ciklus kezdődött, 00= egyébként)
40DD	16605	INPUT jelző (01=INPUT kezdődött, 00=egyébként)
40DE	16606	READ jelző (00=READ utasítás kezdődött, 01=INPUT utasítás kezdődött), ugyanezt a byte-ot a PRINT USING szöveges és numerikus változók elválasztására használja
40DF	16607	SYSTEM program kezdőcíme
40E1	16609	AUTO jelző (00=nincs AUTO üzemmód)
40E2	16610	AUTO aktuális sorszám
40E4	16612	AUTO sorszámnövekmény
40E6	16614	Bemeneti fázisban az aktuális utasításkód stringcíme, végrehajtási fázisban az éppen végrehajtás alatt álló utasítás sorszáma
40E8	16616	Mutató a BASIC veremterületre (FOR, GOSUB utasítások)
40EA	16618	Utolsó hiba sorszáma
40EC	16620	". " opció sorszáma (2 byte)
40EE	16622	Hiba bekövetkezésének címe
40F0	16624	ON ERROR hibakezelő rutin címe (0000=nincs hibakezelő rutin)
40F2	16626	Hibakezelő rutin jelző (hibakezelő rutinba belépéskor FF-re, RESUME utasítással kilépéskor 00-ra állítódik)
40F3	16627	Tizedespont címe (kimeneti pufferben)
40F5	16629	Sorszám az utolsó programmegszakítás (BREAK, STOP) pillanatában
40F7	16631	Hiba esetén az utolsó végrehajtott byte címe
40F9	16633	Mutató az egyszerű változók táblázatának kezdetére (programszöveg vége+2)
40FB	16635	Mutató a tömbváltozók táblázatának kezdetére
40FD	16637	Mutató a tömbváltozók táblázatának végére (szabad tárterület kezdete)
40FF	16639	Mutató a READ utasítással olvasott utolsó karakter utáni byte-ra (elválasztó karakterre)
4101	16641	Változótípus-tábla (26 byte), az ábécé 26 betűjével kezdődő változók alapértelmezése szerinti típusa, inicializáláskor 04 (egyszeres pontosságú), DEFINT, DEFSNG, DEFBDL, DEFSTR utasításokkal állítható
411B	16667	Nyomkövetés jelző (00=TROFF, AF=TRON)
411C	16668	Aritmetikai rutinok munkaterülete (lebegőpontos számok átalakításakor az utolsó byte-ot tárolja)

Tárcím hexa	Tárcím dec	Tartalom
411D	16669	SA szoftverakkumulátor (8 byte), egyszeres pontosságú számok a 4121H...4124H, egész számok s 4121H...4122H címeken
4125	16676	Aritmetikai rutinok munkaterülete (eredmény előjele, kétszeres pontosságú összeadás átvitele)
4126	16677	SA1 másodlagos szoftverakkumulátor (8 byte)
412F	16687	Aritmetikai rutinok munkaterülete
4130	16688	PRINT puffer (26 byte) számok karakterré alakításához
414A	16714	Aritmetikai rutinok munkaterülete (kétszeres pontosságú osztásnál az osztót tárolja)
4151	16721	RAM-munkaterület vége

Lemezes rendszer parancsainak táblázata

Kazettás rendszer inicializálásakor mindenhova JP 012DH=SN-Error kerül

4152	16722	JP 5E46H (CVI)
4155	16725	JP 558EH (FN)
4158	16728	JP 5E49H (CVS)
415B	16731	JP 5655H (DEF)
415E	16734	JP 5E4CH (CVD)
4161	16737	JP 61EBH (EOF)
4164	16740	JP 6231H (LOC)
4167	16743	JP 6242H (LOF)
416A	16746	JP 5E20H (MKI$)
416D	16749	JP 5E30H (MKS$)
4170	16752	JP 5E33H (MKD$)
4173	16755	JP 56C4H (CMD)
4176	16758	JP 5714H (TIME$)
4179	16761	JP 6349H (OPEN)
417C	16764	JP 60ABH (FIELD)
417F	16767	JP 627CH (GET)
4182	16770	JP 627BH (PUT)
4185	16773	JP 606FH (CLOSE)
4188	16776	JP 5F7BH (LOAD)
418B	16779	JP 600BH (MERGE)
418E	16782	JP 6346H (NAME)
4191	16785	JP 63C0H (KILL)
4194	16788	JP 58B7H (&)
4197	16791	JP 60E6H (LSET)
419A	16794	JP 60E5H (RSET)
419D	16797	JP 582FH (INSTR)
41A0	16800	JP 6044H (SAVE)
41A3	16803	JP 5756H (LINE)

Tárcím hexa	Tárcím dec	Tartalom		

Lemezes rendszerrutinok hívása

A BASIC értelmező a lemezes rendszer szubrutinjait az alábbi vezérlésátadási vektorokkal hívja. Kazettás rendszerben inicializáláskor RET (C9 00 00) kerül ezekre a címekre.

		Szubrutin-hívás	Funkció	ROM-cím (hexa)
41A6	16806	LD A,8FH RST 28H	Hibaüzenet	19EC
41A9	16809	JP 5679H	USR függvény	27FE
41AC	16812	JP 5FFCH	Főprogram	1A1C
41AF	16815	JP 598EH	INLINE	0368
41B2	16818	JP 6033H	Főprogram	1AA1
41B5	16821	JP 5BD7H	Sorok rendezése	1AEC
41B8	16824	JP 5B8CH	Sorok rendezése	1AF2
41BB	16827	JP 60A1H	NEW parancs	1B8C
41BE	16830	JP 577CH	kazettakikapcsolás	2174
41C1	16833	JP 59CDH	A-regiszter kiírása	032C
41C4	16836	JP XXXXH	INCH3	0358
41C7	16839	JP 5 F78H	RUN parancs	1EA6
41CA	16842	JP 5A15H	PRINT parancs	206F
41CD	16845	JP 5B9AH	PRINT parancs	20C6
41D0	16848	JP 5B99H	NEWLINE kiírása	2103
41D3	16851	JP 5B65H	PRINT/ parancs	2108 2141
41D6	16854	JP 5784H	INPUT parancs	219E
41D9	16857	JP XXXXH	MID$ parancs	2AEC
41DC	16860	JP 5E63H	READ parancs	222D
41DF	16863	JP 579CH	READ/LIST parancs	2278 2B44
41E2	16866	JP 5B51H	SYSTEM parancs	02B2

41E8	16872	Be- és kimeneti puffer
42E8	17128	Munkaterület vége

C Függelék

Kulcsszavak (1650H–1821H)

Szó	Kód (hexa)	Szó	Kód(hexa)	Szó	Kód (hexa)
END	80	LSET	AB	<	D6
FOR	81	RSET	AC	SGN	D7
RESET	82	SAVE	AD	INT	D8
SET	83	SYSTEM	AE	ABS	D9
CLS	84	LPRINT	AF	FRE	DA
CMD	85	DEF	B0	INP	DB
RANDOM	86	POKE	B1	POS	DC
NEXT	87	PRINT	B2	SQR	DD
DATA	88	CONT	B3	RND	DE
INPUT	89	LIST	B4	LOG	DF
DIM	8A	LLIST	B5	EXP	E0
READ	8B	DELETE	B6	COS	E1
LET	8C	AUTO	B7	SIN	E2
GOTO	8D	CLEAR	B8	TAN	E3
RUN	8E	CLOAD	B9	ATN	E4
IF	8F	CSAVE	BA	PEEK	E5
RESTORE	90	NEW	BB	CVI	E6
GOSUB	91	TAB(	BC	CVS	E7
RETURN	92	TO	BD	CVD	E8
REM	93	FN	BE	EOF	E9
STOP	94	USING	BF	LOC	EA
ELSE	95	VARPTR	C0	LOF	EB
TRON	96	USR	C1	MKI$	EC
TROFF	97	ERL	C2	MKS$	ED
DEFSTR	98	ERR	C3	MKD$	EE
DEFINT	99	STRING$	C4	CINT	EF
DEFSNG	9A	INSTR	C5	CSNG	F0
DEFDBL	9B	POINT	C6	CDBL	F1
LINE	9C	TIME$	C7	FIX	F2
EDIT	9D	MEM	C8	LEN	F3
ERROR	9E	INKEY$	C9	STR$	F4
RESUME	9F	THEN	CA	VAL	F5
OUT	A0	NOT	CB	ASC	F6

Szó	Kód(hexa)	Szó	Kód(hexa)	Szó	Kód (hexa)
ON	A1	STEP	CC	CHR$	F7
OPEN	A2	+	CD	LEFT$	F8
FIELD	A3	–	CE	RIGHT$	F9
GET	A4	*	CF	MID$	FA
PUT	A5	/	D0		FB
CLOSE	A6	↑	D1		
LOAD	A7	AND	D2		
MERGE	A8	OR	D3		
NAME	A9	>	D4		
KILL	AA	=	D5		

LEVEL II. BASIC utasításkódok, belépési pontok (1822H–1899H)

Utasítás	Kód (decimális)	Kód (hexa)	ROM-rutin címe (hexa)
ABS	217	D9	0977
ASC	246	F6	2A0F
ATN	228	E4	15BD
AUTO	183	B7	2008
CDBL	241	F1	0ADB
CHR$	247	F7	2A1F
CINT	239	EF	0A7F
CLEAR	184	B8	1E7A
CLOAD	185	B9	2C1F
CLS	132	84	01C9
CONT	179	B3	1DE4
COS	225	E1	1541
CSAVE	186	BA	2BF5
CSNG	240	F0	0AB1
DATA	136	88	1F05
DEFDBL	155	9B	1E09
DEFINT	153	99	1E03
DEFSNG	154	9A	1E06
DEFSTR	152	98	1E00
DELETE	182	B6	2BC6
DIM	138	8A	2608
EDIT	157	9D	2E60
ELSE	58 149	3A 95	1F07
END	128	80	1DAE
ERL	194	C2	24DD
ERR	195	C3	24CF
ERROR	158	9E	1FF4
EXP	224	E0	1439
FIX	242	F2	0B26
FOR	129	81	1CA1
FRE	218	DA	27D4
GOSUB	145	91	1EB1
GOTO	141	8D	1EC2
IF	143	8F	2039
INKEY$	201	C9	019D
INP	219	DB	2AEF
INPUT	137	89	219A
INT	216	D8	0B37
LEFT$	248	F8	2A61
LEN	243	F3	2A03

Utasítás	Kód (decimális)	Kód (hexa)	ROM-rutin címe (hexa)
LET	140	8C	1F21
LIST	180	B4	2B2E
LLIST	181	B5	2B29
LOG	223	DF	0809
LPRINT	175	AF	2067
MID$	250	FA	2A9A
MEM	200	C8	27C9
NEW	187	BB	1B49
NEXT	135	87	22B6
NOT	203	CB	25C4
ON	161	A1	1F6C
OUT	160	A0	2AFB
PEEK	229	E5	2CAA
POINT	198	C6	0132
POKE	177	B1	2CB1
POS	220	DC	27F5
PRINT	178	B2	206F
RANDOM	134	86	01D3
READ	139	8B	21EF
REM	147	93	1F07
RESET	130	82	0138
RESTORE	144	90	1D91
RESUME	159	9F	1FAF
RETURN	146	92	1EDE
RIGHT$	249	F9	2A91
RND	222	DE	14C9
RUN	142	8E	1EA3
SET	131	83	0135
SGN	215	D7	098A
SIN	226	E2	1547
SQR	221	DD	13E7
STOP	148	94	1DA9
STR$	244	F4	2836
STRING$	196	C4	2A2F
SYSTEM	174	AE	02B2
TAN	227	E3	15A8
TROFF	151	97	1DF8
TRON	150	96	1DF7
USR	193	01	27FE
VAL	245	F5	2AC5
VARPTR	192	C0	24EB

Aritmetikai ROM-rutinok

Egész számok

Célregiszter	Művelet	Forrásregiszterek
HL	Összeadás	HL + DE
HL	Kivonás	HL – DE
HL	Szorzás	HL * DE
SA	Osztás	DE / HL

Egyszeres pontosságú számok

Célregiszter	Művelet	Forrásregiszterek
SA	Összeadás	SA + (BCDE)
SA	Kivonás	SA – (BCDE)
SA	Szorzás	SA * (BCDE)
SA	Osztás	SA / (BCDE)

Kétszeres pontosságú számok

Célregiszter	Művelet	Forrásregiszterek
SA	Összeadás	SA + SA1
SA	Kivonás	SA – SA1
SA	Szorzás	SA * SA1
SA	Osztás	SA / SA1

Aritmetikai ROM-rutinok belépési címei (18ABH–18AAH)

	Egész szám	Egyszeres pontosságú szám	Kétszeres pontosságú szám	String
Összeadás	0BD2	0716	0C77	298F
Kivonás	0BC7	0713	0C70	–
Szorzás	0BF2	0847	0DA1	–
Osztás	2490	08A2	0DE5	–
Összehasonlítás	0A39	0A0C	0A78	–

Aritmetikai műveletek végrehajtási idői

	Egész	Egyszeres	Kétszeres
Összeadás	130 μs	630 μs	1300 μs
Kivonás	210 μs	1300 μs	1300 μs
Szorzás	900 μs	2200 μs	2200 μs
Osztás	5100 μs	4800 μs	42000 μs

Matematikai függvények végrehajtási idői

SIN	25 ms
COS	25 ms
ATN	27 ms
TAN	54 ms
LOG	19 ms
EXP	28 ms
SQR	48 ms
X ↑ Y	50 ms

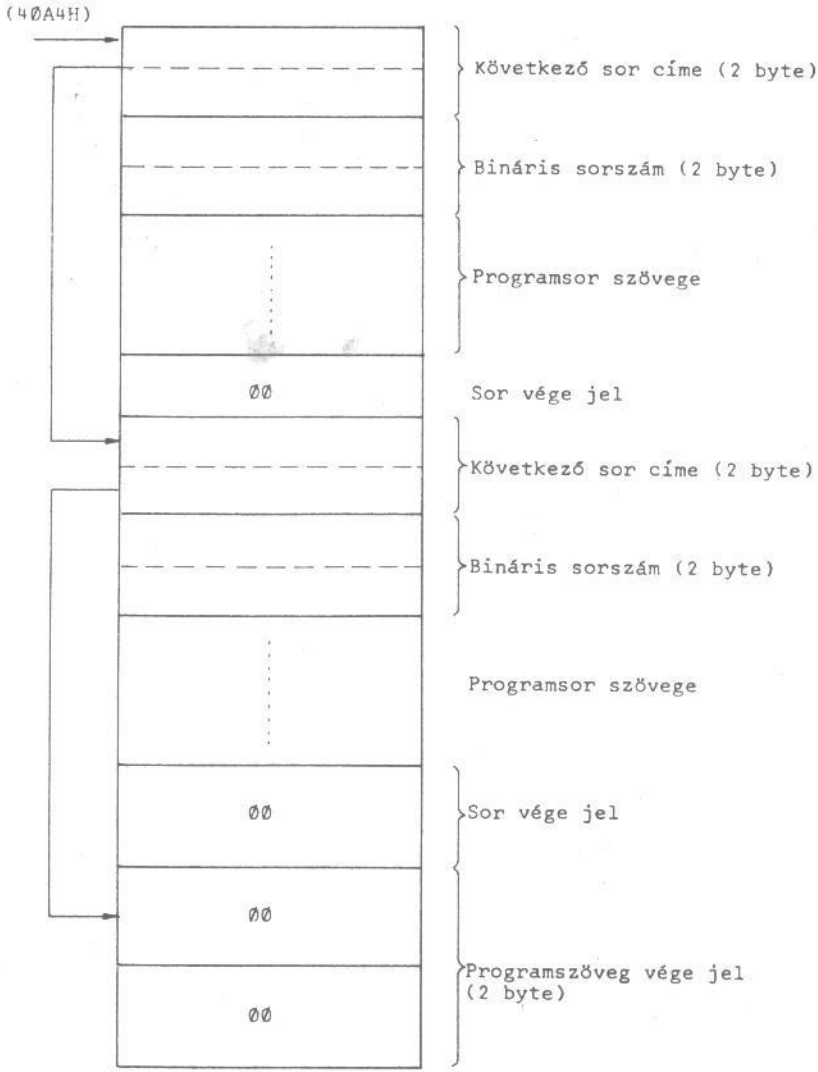
Néhány matematikai konstans

	Egyszeres pontosságú			
	alsó byte	2. byte	felső byte	kitevő
π	DB	0F	49	82
2π	DB	0F	49	83
e	54	F8	2D	82
$180°/\pi$	E1	2E	65	86
$\pi/180°$	35	FA	0E	7B
10	00	00	20	84
ln 10	8E	5D	13	82

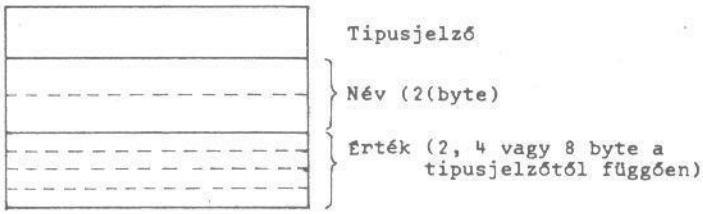
	Kétszeres pontosságú							
	alsó byte	2. byte	3. byte	4. byte	5. byte	6. byte	felső byte	kitevő
π	7E	2A	92	A2	DA	0F	49	82
2π	7E	2A	92	A2	DA	0F	49	83
e	DA	BC	24	58	54	F8	2D	82
$180°/\pi$	8D	1B	E9	D2	E0	2E	65	86
$\pi/180°$	1A	CC	94	12	35	FA	0E	7B
10	00	00	00	00	00	00	20	84
ln 10	AB	4B	AC	DD	8D	5D	13	82

BASIC TÁBLÁZATOK

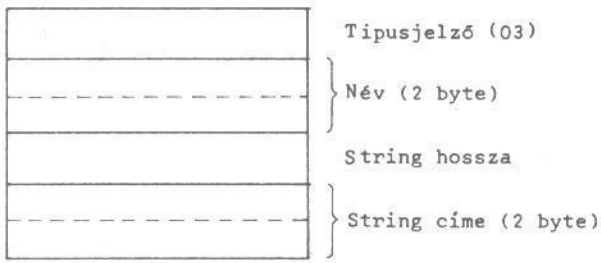
1. Programszöveg



2. Számok

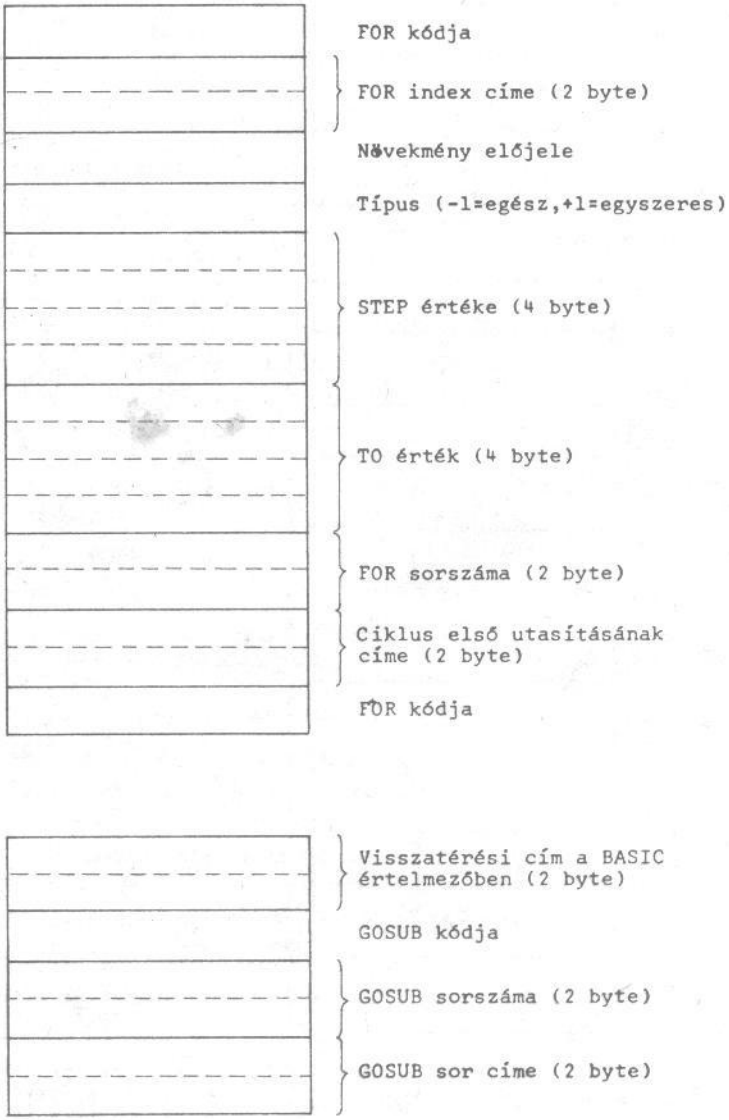


3. Stringek



4. Tömbök

5. BASIC verem



Hibaüzenetek (18C9H–18F6H)

Hiba száma	Hiba kódja	Hiba oka	Hibaüzenet forrása
00	NF	FOR nélküli NEXT	22C2
02	SN	Szintaktikus hiba	00DA,02C7,0EEF, 1C9E,1D32,1E0E, 1E66,2022,235B, 2615,2AE9,2DE2
04	RG	GOSUB nélküli RETURN	1EEC
06	OD	Adatlista vége (READ)	2214,22A2
08	FC	Hibás függvényargumentum	1E4C
0A	OV	Túlcsordulás	07B2
0C	OM	Tár betelt	197C
0E	UL	Nem létező sorszám	1EDB
10	BS	Hibás index	273F
12	DD	Kétszer definiált szimbólum	2735
14	/0	Nullával osztás	08A5,0DE9,1401
16	ID	INPUT helytelen használata	2833
18	TM	Változó-típuskeveredés	0AF8
1A	OS	Stringterület betelt	28DD
1C	LS	Túl hosszú szövegváltozó	29A5
1E	ST	Literál stringmező betelt	28A3
20	CN	CONT helytelen használata	1DEB
22	NR	RESUME helytelen használata	198C
24	UE	Érvénytelen hibakód	2005
26	UE	Érvénytelen hibakód	2005
28	MO	Hiányzó operandus	24A2
2A	FD	Adathiba a kazettán	218C
2C	SN	DISK-BASIC parancs kazettás rendszerben (L3-Error)	12DF

D Függelék

ASCII karakterek

Vezérlőkarakterek			Számok Jelek	Nagybetűk		Kisbetűk
0 NUL 00H	16 DLE 10H	32 SP 20H	48 0 30H	64 @ 40H	80 P 50H	96 \ 60H	111 p 70H
1 BREAK S0H 01H	17 DC1 11H	33 ! 21H	49 1 31H	65 A 41H	81 Q 51H	97 a 61H	113 q 71H
2 STX 02H	18 DC2 12H	34 " 22H	50 2 32H	66 B 42H	82 R 52H	98 b 62H	114 r 72H
3 ETX 03H	19 DC3 13H	35 # 23H	51 3 33H	67 C 43H	83 S 53H	99 c 63H	115 s 73H
4 E0T 04H	20 DC4 14H	36 $ 24H	52 4 34H	68 D 44H	84 T 54H	100 d 64H	116 t 74H
5 ENQ 05H	21 NAK 15H	37 % 25H	53 5 35H	69 E 45H	85 U 55H	101 e 65H	117 u 75H
6 ACK 06H	22 SYN 16H	38 & 26H	54 6 36H	70 F 46H	86 V 56H	102 f 66H	118 v 76H
7 BEL 07H	23 32. KAR.MOD. ETB 17H	39 27H	55 7 37H	71 G 47H	97 W 57H	103 g 67H	119 w 77H
8 BS 08H	24 BALRA CAN ← 18H	40 (28H	56 8 38H	72 H 48H	88 X 58H	104 h 68H	120 x 78H
9 HT 09H	25 JOBBRA EM → 19H	41) 29H	57 9 39H	73 I 49H	89 Y 59H	105 i 69H	121 y 79H

Vezérlőkarakterek		Számok Jelek		Nagybetűk		Kisbetűk	
10 LF 0AH	26 LE ↓ SUB 1AH	42 * 2AH	58 : 3AH	74 J 4AH	90 Z 5AH	106 j 6AH	122 z 7AH
11 VT 0BH	27 FEL ↑ ESC 1BH	43 + 2BH	59 ; 3BH	75 K 4BH	91 [5BH	107 k 6BH	123 { ↑ 7BH
12 FF 0CH	28 HOME 64 KAR. FS Mód 1CH	44 , 2CH	60 < 3CH	76 L 4CH	92 \ 5CH	108 l 6CH	124 ↓ 7CH
13 NEWLINE CR 0DH	29 KURZOR SOR GS ELEJÉRE 1DH	45 – 3DH	61 = 3DH	77 M 4DH	93] 5DH	109 m 6DH	125 } ← 7DH
14 KURZOR BE SO 0EH	30 KURZOR SOR RS VÉGÉRE 1EH	46 . 2EH	62 > 3EH	78 N 4EH	94 ^ 5EH	110 n 6EH	126 ~ → 7EH
15 KURZOR KI SI 0FH	31 CLS US 1FH	47 / 2FH	63 ? 3FH	79 O 4FH	95 KURZOR – 5FH	111 o 6FH	127 delete 7FH

FORMÁTUM:

DECIMÁLIS	HT1080Z
ASCII KARAKTER	
HEXA	

Decimális és hexadecimális számok

Hexa	Decimális felső	Decimális alsó
00	0	0
01	256	1
02	512	2
03	768	3
04	1024	4
05	1280	5
06	1536	6
07	1792	7
08	2048	8
09	2304	9
0A	2560	10
0B	2816	11
0C	3072	12
0D	3328	13
0E	3584	14
0F	3840	15
10	4096	16
11	4352	17
12	4608	18
13	4864	19
14	5120	20
15	5376	21
16	5632	22
17	5888	23
18	6144	24
19	6400	25
1A	6656	26
1B	6912	27
1C	7168	28
1D	7424	29
1E	7680	30
1F	7936	31
20	8192	32
21	8448	33
22	8704	34
23	8960	35
24	9216	36
25	9472	37
26	9728	38
27	9948	39
28	10240	40
29	10496	41
2A	10752	42
2B	11008	43
2C	11264	44
2D	11520	45
2E	11776	46
2F	12032	47
30	12288	48
31	12544	49
32	12800	50
33	13056	51
34	13312	52
35	13568	53
36	13824	54
37	14080	55
38	14080	55
39	14336	56
3A	14848	58
3B	15104	59
3C	15360	60
3D	15616	61
3E	15872	62
3F	16128	63
40	16384	64
41	16640	65
42	16896	66
43	17152	67
44	17408	68
45	17664	69
46	17920	70
47	18176	71
48	18432	72
49	18688	73
4A	18944	74
4B	19200	75
4C	19456	76
4D	19712	77
4E	19968	78
4F	20224	79
50	24080	80
51	20736	81
52	20992	82
53	21248	83
54	21504	84
55	21760	85
56	22016	86
57	22272	87
58	22528	88
59	22784	89
5A	23040	90
5B	23296	91
5C	23552	92
5D	23808	93
5E	24064	94
5F	24320	95
60	24576	96
61	24832	97
62	25088	98
63	25344	99
64	25600	100
65	25856	101
66	26112	102
67	26368	103
68	26624	104
69	26880	105
6A	27136	106
6B	27392	107
6C	27648	108
6D	27904	109
6E	28160	110
6F	28416	111
70	28672	112
71	28928	113
72	29184	114
73	29440	115
74	29696	116
75	29952	117
76	30208	118
77	30464	119
78	30720	120
79	30976	121
7A	31232	122
7B	31488	123
7C	31744	124
7D	32000	125
7E	32256	126
7F	32512	127
80	32768	128
81	33024	129
82	33280	130
83	33536	131
84	33792	132
85	34048	133
86	34304	134
87	34560	135
88	34816	136
89	35072	137
8A	35328	138
8B	35584	139
8C	35840	140
8D	36096	141
8E	36352	142
8F	36608	143
90	36864	144
91	37120	145
92	37376	146
93	37632	147
94	37888	148
95	38144	149
96	38400	150
97	38656	151
98	38912	152

Hexa	Decimális felső	Decimális alsó
99	39168	153
9A	39424	154
9B	39680	155
9C	39936	156
9D	40192	157
9E	40448	158
9F	40704	159
A0	40960	160
A1	41216	161
A2	41472	162
A3	41728	163
A4	41984	164
A5	42240	165
A6	42496	166
A7	42752	167
A8	43008	168
A9	43264	169
AA	43520	170
AB	43776	171
AC	44032	172
AD	44288	173
AE	44544	174
AF	44800	175
B0	45056	176
B1	45312	177
B2	45568	178
B3	45824	179
B4	46080	180
B5	46336	181
B6	46592	182
B7	46848	183
B8	47104	184
B9	47360	185
BA	47616	186

Hexa	Decimális felső	Decimális alsó
BB	47872	187
BC	48128	188
BD	48384	189
BE	48640	190
FB	48896	191
C0	49152	192
C1	49408	193
C2	49664	194
C3	49920	195
C4	50176	196
C5	50432	197
C6	50688	198
C6	50944	199
C8	51200	200
C9	51456	201
CA	51712	202
CB	51968	203
CC	52224	204
CD	52480	205
CE	52736	206
CF	52992	207
D0	53248	208
D1	53504	209
D2	53760	210
D3	54016	211
D4	54272	212
D5	54528	213
D6	54784	214
D7	55040	215
D8	55296	216
D9	55552	217
DA	55808	218
DB	56064	219
DC	56320	220
DD	56576	221

Hexa	Decimális felső	Decimális alsó
DE	56832	222
DF	57088	223
E0	57344	224
E1	57600	225
E2	57856	226
E3	58112	227
E4	58368	228
E5	58624	229
E6	58880	230
E7	59136	231
E8	59392	232
E9	59648	233
EA	59904	234
EB	60160	235
EC	60416	236
ED	60672	237
EE	60928	238
EF	61184	239
F0	61440	240
F1	61696	241
F2	61952	242
F3	62208	243
F4	62464	244
F4	62720	245
F6	62976	246
F7	63232	247
F8	63488	248
F9	63744	249
FA	64000	250
FB	64256	251
FC	64512	252
FD	64768	253
FE	65024	254
FF	65280	255

Címzési eltolás decimális és hexadecimális értékei

Előre (00H–7FH)

	0	1	2	3	4	5	6	7	8	9	A	C	B	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127

Hátra (80H–FFH)

8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	128	127	126	125	124	122	122	121	120	119	118	117	116	115	114	113
9	112	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97
A	96	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81
B	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65
C	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49
D	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
E	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
F	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

E Függelék

EDI ASSEMBLER/EDITOR program

Az EDI assembler/editor gépi kódú program, betöltése a SYSTEM paranccsal történik, file-neve EDI, kezdőcíme 17152 (4300H), így betöltéséhez a *? kérdésre EDI választ kell adni. Betöltés után automatikusan elindul, *?/17152-vel újraindítható. Az EDI parancsait a ?> inputot váró üzenetre válaszként kell megadni, először a parancs betűjét, azután a címet kell beírni. A címek decimálisak, hexadecimálisak (a végén H) vagy oktálisak (a végén O) lehetnek. Több paraméter esetén mindegyikre külön kérdés jelenik meg. A gépelési hibák visszaléptetéssel (balra nyíl) javíthatók. A ?> üzenetre NEWLINE választ adva az EDI törli a képernyőt és újra ?> szintre kerül.

Sormutató parancsok

Az EDI assembler/editor sororientált, ugyanúgy, mint maga a Z80 assembler is. A sormutató mindig a forrásfile aktuális sorára mutat.

- T – top – a sormutató a forrásfile elejére megy
- B – bottom – a sormutató a forrásfile végére megy
- Dn – down – a sormutató n sorral lejjebb megy (alapértelmezésben n=1)
- Un – up – a sormutató n sorral feljebb megy (alapértelmezésben n=1)
- Pn – print – a képernyőre a sormutató helyétől n sort listáz (alapértelmezésben n=1), a listázás után a sormutató az utolsó kiírt sorra mutat
- L – locate – a forrásfile-ban megkeres egy karakterváltozót, a sormutatót a megtalált szövegváltozót tartalmazó sorra állítja, ha nincs ilyen, akkor a sormutató a file végére áll.

Szövegszerkesztő parancsok

- E – enter – beírás a sormutató helye elé, folyamatosan sorszámokat ad, kilépés csak ". " karaktert tartalmazó sorral (a sorszámok csupán a szerkesztést segítik, a forrásfile-nak nem részei)

N – new – a sormutató által mutatott sort új sorra cseréli
Zn – zap – a sormutató helyétől kezdve n sort töröl (alapértelmezésben n=1)

Általános parancsok

H – how big – kiírja a forrásfile kezdőcímét és végcímét
K – kill – törli a forrásfile-t
A – assemble – a forrásfile kezdetétől az END pszeudoutasításig terjedő forrásprogramot gépi kódra lefordítja és a LOAD pszeudoutasításban megadott tárcímre betölti, az OPTION? kérdésre lehetséges válaszok:
V – video – képernyő
E – external – nyomtató
C – casette – kazettára SYSTEM szalagot ír, tárcímeket és file-nevet kérdez
(alapértelmezés a leggyorsabb mód, amíg a forrásprogram hibás)

Kazettaparancsok

A forrásprogramok és a gépi kódú programok kimenthetők és betölthetők, gépi kódú program esetén a parancs után O betűt kell írni.

W – write – file-t ír a kazettára, forrás módban a tárban levő forrásfile-t írja ki, gépi kódú programoknál címeket kérdez, SYSTEM formátumú szalagot készít, file-nevet mindkét módban kérdez
R – read – kazettáról file-t olvas, a forrásprogramot a tárban levő forrásprogram utánra tölti (egyszerű merge), a gépi kódú programot címek szerint tölti be, file-nevet kérdez, betöltés közben "L" látható, hibás betöltés után "*", hibátlan betöltés után a file-méret jelenik meg
V – valid – ellenőrzi, hogy a kazettára kiírt file hibátlan-e (ugyanaz, mint az R, de nem tölt a tárba)

Az assembler

A forrásprogram utasításai Z80 assembler utasítások. Minden programsor legfeljebb négy mezőt tartalmazhat:

CIMKE:UTASITÁSKÓD OPERANDUS;MEGJEGYZÉS

Az utasítások formája kötetlen, de az utasításkód után space-t kell hagyni. A programsorok jobbra nyíllal tabulálhatók.

Az utasításokat címkékkel lehet megjelölni. A címkék (szimbolikus nevek) kétféleképpen adhatók meg: az utasításkód elé írva (az assembler a programszámláló értékét rendeli hozzá) vagy EQU pszeudoutasítással (ezzel tetszés szerinti értéket lehet a címkékhez rendelni). A címke első karaktere csak betű lehet, a további karakterek között számok is lehetnek, hossza legfeljebb 7 karakter. Utasításkódok, regiszternevek és feltételkódnevek nem lehetnek címkék.

Az utasításkódok a Z80 utasításkódok, továbbá a következő pszeudoutasítás-kódok lehetnek:

END – end – forrásprogram lezárása, operandusa nincs

DS – define storage – helyfoglalás, az operandusban megadott számú byte-ot üresen hagy (puffer, verem stb. céljára)

DW – define word – kétbyte-os szót generál, a byte-okat fordított sorrendben (16 bites egész számok, tárcímek stb.)

DB – define bytes – az operandus értékét a gépi kódú file-ba generálja, több operandus is lehet, vesszőkkel elválasztva, pl.: DB 6,93H,'I' 0EFH, a programszámláló minden operandus után növekszik, mintha azok külön sorban lennének, az operandusok ASCII karakterstringek is lehetnek (pl.: DB ' JO REGGELT ')

EQU – equate – szimbólumhoz értéket rendel (pl.: NEWLINE: EQU 12)

ORG – origin – a gépi kódú program kezdőcímét határozza meg, egy programfile-ban tetszés szerint többször is alkalmazható a programrészek/programok különböző kezdőcímekre helyezéséhez

LOAD – load – a gépi kódú program betöltése az operandusként megadott címre, ez a LOAD független az A parancsban az OPTION? kérdésre adott választól, a betöltés a következő ORG utasításig tart

Az operandusok száma nulla (pl. NOP), egy (pl. CP), kettő (pl. BIT), és egy vagy kettő (pl. JR) lehet. Az operandusok lehetnek regiszternevek (pl. A, B, DE, IX, ...), feltételnevek (pl. Z, NZ, C, M, ...), számok (számjegygyel kezdődő decimális, hexadecimális vagy oktális értékek), ASCII karaktersorozatok (' vagy " közé zárt karakterek), végül lehet a programszámláló futás közbeni értéke ($). Számok helyett az előző adattípusokból matematikai műveletekkel alkotott kifejezések is használhatók.

Megjegyzések a programban bárhol elhelyezhetők, lehetnek csak megjegyzést tartalmazó programsorok is. A megjegyzés vége a programsor vége.

Az assembler fordítási lista az A parancs OPTION? kérdésére adott választól függően képernyőre vagy nyomtatóra kerülhet. A lista laponként készül, rövid szünetekkel, a szünet ideje alatt bármely billentyűt megnyomva a listázás megszakad (a lap a képernyőn marad), a listázás bármely billentyű

megnyomásával folytatható. A lapméret képernyőn 15 sor, nyomtatón 60 sor.

Ha az assembler fordítóprogram hibát talál a forrásprogramban, akkor megáll, hibaüzenetet ad, a hibás sort kiírja, a sormutatót erre a sorra állítja. A hibaüzenetek a következők:

DOUBLE SYMBOL	– többször definiált címke
RSVD	– a címke neve lefoglalt szó
FULL	– a szimbólumtábla betelt
ORG	– nincs ORG utasítás
OPND	– operandushiba
UNDEF	– hiányzó címke
EOF	– hiányzó END
HUH?	– egyéb hiba

Futtatás, hibakeresés

Az A paranccsal lefordított, a LOAD pszeudoutasítás hatására a tárba betöltött gépi kódú program futtatása, nyomkövetése, a hibák keresése a következő parancsokkal történhet:

G	– goto	– vezérlésátadás a gépi kódú programnak, belépési címet és töréspontcímet kérdez (belépési cím alapértelmezésben a felhasználói programszámláló aktuális értéke, töréspont alapértelmezésben nincs)
C	– copy	– a kezdőcímtől végcímig terjedő tárblokk áthelyezése a célcímre
F	– fill	– a kezdőcímtől a végcímig feltölti a tárat egy adatkonstanssal
X	– examine	– a felhasználói gépi kódhoz tartozó regisztertartalmakat írja ki
M	– modify	– a megadott tárcímtől byte-onként kiírja a tár tartalmát, a byte-ok átírhatók; kilépés "."-tal
Q	– query	– a tár megadott címétől 8 sorban soronként 8 byte-ot kiír a következő formában: tárcím, 8 byte hexadecimálisan, ugyanaz ASCII karakteres formában a 7. bit nélkül, majd a 7. bittel együtt (a harmadik mező megjeleníti azokat a karaktereket, amelyekben a 7. bit 1, a negyedik mező ezeket grafikus karakterekként mutatja, a vezérlőkarakterek "."-tá alakítva jelennek meg), alapértelmezés a 0000H, ill. az előző kiírást követő cím,

HT-1080Z ROM LISTA

```
0000 F3          DI
0001 AF          XOR A
0002 C37406      JP 0674H
0005 C30040      JP 4000H
0008 C30040      JP 4000H
000B E1          POP HL
000C E9          JP (HL)
000D C39F06      JP 069FH
0010 C30340      JP 4003H
0013 C5          PUSH BC
0014 0601        LD B,01H
0016 182E        JR 0046H
0018 C30640      JP 4006H
001B C5          PUSH BC
001C 0602        LD B,02H
001E 1826        JR 0046H
0020 C30940      JP 4009H
0023 C5          PUSH BC
0024 0604        LD B,04H
0026 181E        JR 0046H
0028 C30C40      JP 400CH
002B 111540      LD DE,4015H
002E 18E3        JR 0013H
0030 C30F40      JP 400FH
0033 111D40      LD DE,401DH
0036 18E3        JR 001BH
0038 C31240      JP 4012H
003B 112540      LD DE,4025H
003E 18DB        JR 001BH
0040 C3D905      JP 05D9H
0043 C9          RET
0044 00          NOP
0045 00          NOP
0046 C3C203      JP 03C2H
0049 CD2B00      CALL 002BH
004C B7          OR A
004D C0          RET NZ
004E 18F9        JR 0049H

0000-004E   ROM  1

0050  0D         .   .
0051  0D         .   .
0052  1F         .   .
0053  1F         .   .
0054  01         .   .
0055  01         .   .
0056  5B         [   [
0057  1B         .   .
0058  0A         .   .
0059  1A         .   .
005A  08         .   .
005B  18         .   .
005C  09         .   .
005D  19         .   .
005E  20
005F  20

0000-005F   ROM  1
```

```
0060 0B          DEC BC
0061 78          LD A,B
0062 B1          OR C
0063 20FB        JR NZ,0060H
0065 C9          RET
0066 310006      LD SP,0600H
0069 3AEC37      LD A,(37ECH)
006C 3C          INC A
006D FE02        CP 02H
006F D20000      JP NC,0000H
0072 C3CC06      JP 06CCH
0075 118040      LD DE,4080H
0078 21F718      LD HL,18F7H
007B 012700      LD BC,0027H
007E EDB0        LDIR
0080 21E541      LD HL,41E5H
0083 363A        LD (HL),3AH
0085 23          INC HL
0086 70          LD (HL),B
0087 23          INC HL
0088 362C        LD (HL),2CH
008A 23          INC HL
008B 22A740      LD (40A7H),HL
008E 112D01      LD DE,012DH
0091 061C        LD B,1CH
0093 215241      LD HL,4152H
0096 36C3        LD (HL),C3H
0098 23          INC HL
0099 73          LD (HL),E
009A 23          INC HL
009B 72          LD (HL),D
009C 23          INC HL
009D 10F7        DJNZ 0096H
009F 0615        LD B,15H
00A1 36C9        LD (HL),C9H
00A3 23          INC HL
00A4 23          INC HL
00A5 23          INC HL
00A6 10F9        DJNZ 00A1H
00A8 21E842      LD HL,42E8H
00AB 70          LD (HL),B
00AC 31F841      LD SP,41F8H
00AF CD8F1B      CALL 1B8FH
00B2 CDC901      CALL 01C9H
00B5 210501      LD HL,0105H
00B8 CDA728      CALL 28A7H
00BB CDB31B      CALL 1BB3H
00BE 38F5        JR C,00B5H
00C0 D7          RST 10
00C1 B7          OR A
00C2 2012        JR NZ,00D6H
00C4 214C43      LD HL,434CH
00C7 23          INC HL
00C8 7C          LD A,H
00C9 B5          OR L
00CA 281B        JR Z,00E7H
00CC 7E          LD A,(HL)
00CD 47          LD B,A
00CE 2F          CPL
00CF 77          LD (HL),A

0060-00CF   ROM  2
```

```
00D0 BE          CP (HL)
00D1 70          LD (HL),B
00D2 28F3        JR Z,00C7H
00D4 1811        JR 00E7H
00D6 CD5A1E      CALL 1E5AH
00D9 B7          OR A
00DA C29719      JP NZ,1997H
00DD EB          EX DE,HL
00DE 2B          DEC HL
00DF 3E8F        LD A,8FH
00E1 46          LD B,(HL)
00E2 77          LD (HL),A
00E3 BE          CP (HL)
00E4 70          LD (HL),B
00E5 20CE        JR NZ,00B5H
00E7 2B          DEC HL
00E8 111444      LD DE,4414H
00EB DF          RST 18
00EC DA7A19      JP C,197AH
00EF 11CEFF      LD DE,FFCEH
00F2 22B140      LD (40B1H),HL
00F5 19          ADD HL,DE
00F6 22A040      LD (40A0H),HL
00F9 CD4D1B      CALL 1B4DH
00FC 211101      LD HL,0111H
00FF CDA728      CALL 28A7H
0102 C3191A      JP 1A19H
```

00D0-0102 ROM 3

```
0105  52         R    R
0106  45         E    E
0107  41         A    A
0108  44         D    D
0109  59         Y    Y
010A  20
010B  00         .    .
010C  00         .    .
010D  00         .    .
010E  00         .    .
010F  00         .    .
0110  00         .    .
0111  0D         .    .
0112  0D         .    .
0113  0D         .    .
0114  0D         .    .
0115  0D         .    .
0116  0D         .    .
0117  0D         .    .
0118  0D         .    .
0119  0D         .    .
011A  0D         .    .
011B  0D         .    .
011C  0D         .    .
011D  0D         .    .
011E  0D         .    .
011F  0D         .    .
0120  0D         .    .
0121  0D         .    .
0122  0D         .    .
```

00D0-0122 ROM 3

```
0123  0D         .    .
0124  0D         .    .
0125  0D         .    .
0126  0D         .    .
0127  0D         .    .
0128  0D         .    .
0129  0D         .    .
012A  0D         .    .
012B  0D         .    .
012C  00         .    .
```

0123-012C ROM 4

```
012D 1E2C        LD E,2CH
012F C3A219      JP 19A2H
0132 D7          RST 10
0133 AF          XOR A
0134 013E80      LD BC,803EH
0137 013E01      LD BC,013EH
013A F5          PUSH AF
013B CF          RST 8
013C 28CD        JR Z,010BH
013E 1C          INC E
013F 2B          DEC HL
0140 FE80        CP 80H
0142 D24A1E      JP NC,1E4AH
0145 F5          PUSH AF
0146 CF          RST 8
0147 2C          INC L
0148 CD1C2B      CALL 2B1CH
014B FE30        CP 30H
014D D24A1E      JP NC,1E4AH
0150 16FF        LD D,FFH
0152 14          INC D
0153 D603        SUB 03H
0155 30FB        JR NC,0152H
0157 C603        ADD A,03H
0159 4F          LD C,A
015A F1          POP AF
015B 87          ADD A,A
015C 5F          LD E,A
015D 0602        LD B,02H
015F 7A          LD A,D
0160 1F          RRA
0161 57          LD D,A
0162 7B          LD A,E
0163 1F          RRA
0164 5F          LD E,A
0165 10F8        DJNZ 015FH
0167 79          LD A,C
0168 8F          ADC A,A
0169 3C          INC A
016A 47          LD B,A
016B AF          XOR A
016C 37          SCF
016D 8F          ADC A,A
016E 10FD        DJNZ 016DH
0170 4F          LD C,A
0171 7A          LD A,D
0172 F63C        OR 3CH
```

0123-0172 ROM 4

```
0174 57          LD D,A
0175 1A          LD A,(DE)
0176 B7          OR A
0177 FA7C01      JP N,017CH
017A 3E80        LD A,80H
017C 47          LD B,A
017D F1          POP AF
017E B7          OR A
017F 78          LD A,B
0180 2810        JR Z,0192H
0182 12          LD (DE),A
0183 FA8F01      JP N,018FH
0186 79          LD A,C
0187 2F          CPL
0188 4F          LD C,A
0189 1A          LD A,(DE)
018A A1          AND C
018B 12          LD (DE),A
018C CF          RST 8
018D 29          ADD HL,HL
018E C9          RET
018F B1          OR C
0190 18F9        JR 018BH
0192 A1          AND C
0193 C6FF        ADD A,FFH
0195 9F          SBC A,A
0196 E5          PUSH HL
0197 CD8D09      CALL 098DH
019A E1          POP HL
019B 18EF        JR 018CH
019D D7          RST 10
019E E5          PUSH HL
019F 3A9940      LD A,(4099H)
01A2 B7          OR A
01A3 2006        JR NZ,01ABH
01A5 CD5803      CALL 0358H
01A8 B7          OR A
01A9 2811        JR Z,01BCH
01AB F5          PUSH AF
01AC AF          XOR A
01AD 329940      LD (4099H),A
01B0 3C          INC A
01B1 CD5728      CALL 2857H
01B4 F1          POP AF
01B5 2AD440      LD HL,(40D4H)
01B8 77          LD (HL),A
01B9 C38428      JP 2884H
01BC 212819      LD HL,1928H
01BF 222141      LD (4121H),HL
01C2 3E03        LD A,03H
01C4 32AF40      LD (40AFH),A
01C7 E1          POP HL
01C8 C9          RET
01C9 3E1C        LD A,1CH
01CB CD3A03      CALL 033AH
01CE 3E1F        LD A,1FH
01D0 C33A03      JP 033AH
01D3 ED5F        LD   A,R
01D5 32AB40      LD (40ABH),A
01D8 C9          RET
```



```
01D9 2101FC      LD HL,FC01H
01DC CD2102      CALL 0221H
01DF 060B        LD B,0BH
01E1 10FE        DJNZ 01E1H
01E3 2102FC      LD HL,FC02H
01E6 CD2102      CALL 0221H
01E9 060B        LD B,0BH
01EB 10FE        DJNZ 01EBH
01ED 2100FC      LD HL,FC00H
01F0 CD2102      CALL 0221H
01F3 065C        LD B,5CH
01F5 10FE        DJNZ 01F5H
01F7 C9          RET
01F8 E5          PUSH HL
01F9 2100FB      LD HL,FB00H
01FC 181B        JR 0219H
01FE 7E          LD A,(HL)
01FF D623        SUB 23H
0201 3E00        LD A,00H
0203 200D        JR NZ,0212H
0205 CD012B      CALL 2B01H
0208 CF          RST 8
0209 2C          INC L
020A 7B          LD A,E
020B A2          AND D
020C C602        ADD A,02H
020E D24A1E      JP NC,1E4AH
0211 3D          DEC A
0212 D3FE        OUT (FEH),A
0214 00          NOP
0215 E5          PUSH HL
0216 2104FF      LD HL,FF04H
0219 CD2102      CALL 0221H
021C E1          POP HL
021D C9          RET
021E 2100FF      LD HL,FF00H
0221 3A3D40      LD A,(403DH)
0224 A4          AND H
0225 B5          OR L
0226 D3FF        OUT (FFH),A
0228 323D40      LD (403DH),A
022B C9          RET
022C 3A3F3C      LD A,(3C3FH)
022F EE0A        XOR 0AH
0231 323F3C      LD (3C3FH),A
0234 C9          RET
0235 C5          PUSH BC
0236 E5          PUSH HL
0237 0608        LD B,08H
0239 CD4102      CALL 0241H
023C 10FB        DJNZ 0239H
023E E1          POP HL
023F C1          POP BC
0240 C9          RET
0241 C5          PUSH BC
0242 F5          PUSH AF
0243 DBFF        IN A,(FFH)
0245 17          RLA
0246 30FB        JR NC,0243H
0248 0641        LD B,41H
```

```
024A 10FE        DJNZ 024AH
024C CD1E02      CALL 021EH LABEL2
024F 0676        LD B,76H
0251 10FE        DJNZ 0251H
0253 DBFF        IN A,(FFH)
0255 47          LD B,A
0256 F1          POP AF
0257 CB10        RL B
0259 17          RLA
025A F5          PUSH AF
025B CD1E02      CALL 021EH LABEL2
025E F1          POP AF
025F C1          POP BC
0260 C9          RET
0261 CD6402      CALL 0264H
0264 E5          PUSH HL
0265 C5          PUSH BC
0266 D5          PUSH DE
0267 F5          PUSH AF
0268 0E08        LD C,08H
026A 57          LD D,A
026B CDD901      CALL 01D9H OKAJEL
026E 7A          LD A,D
026F 07          RLCA
0270 57          LD D,A
0271 300B        JR NC,027EH ciro
0273 CDD901      CALL 01D9H okajel
0276 0D          DEC C
0277 20F2        JR NZ,026BH
0279 F1          POP AF
027A D1          POP DE
027B C1          POP BC
027C E1          POP HL
027D C9          RET
027E 0687        LD B,87H
0280 10FE        DJNZ 0280H
0282 18F2        JR 0276H
0284 CDFE01      CALL 01FEH
0287 06FF        LD B,FFH
0289 AF          XOR A
028A CD6402      CALL 0264H
028D 10FB        DJNZ 028AH
028F 3EA5        LD A,A5H
0291 18D1        JR 0264H
0293 CDFE01      CALL 01FEH
0296 E5          PUSH HL
0297 AF          XOR A
0298 CD4102      CALL 0241H
029B FEA5        CP A5H
029D 20F9        JR NZ,0298H
029F 3E2A        LD A,2AH
02A1 323E3C      LD (3C3EH),A
02A4 323F3C      LD (3C3FH),A
02A7 E1          POP HL
02A8 C9          RET
02A9 CD1403      CALL 0314H
02AC 22DF40      LD (40DFH),HL
02AF CDF801      CALL 01F8H
02B2 CDE241      CALL 41E2H
02B5 318842      LD SP,4288H
```

024A-02B5 ROM 7

```
02B8 CDFE20      CALL 20FEH
02BB 3E2A        LD A,2AH
02BD CD2A03      CALL 032AH
02C0 CDB31B      CALL 1BB3H
02C3 DACC06      JP C,06CCH
02C6 D7          RST 10
02C7 CA9719      JP Z,1997H
02CA FE2F        CP 2FH
02CC 284F        JR Z,031DH
02CE CD9302      CALL 0293H
02D1 CD3502      CALL 0235H
02D4 FE55        CP 55H
02D6 20F9        JR NZ,02D1H
02D8 0606        LD B,06H
02DA 7E          LD A,(HL)
02DB B7          OR A
02DC 2809        JR Z,02E7H
02DE CD3502      CALL 0235H
02E1 BE          CP (HL)
02E2 20ED        JR NZ,02D1H
02E4 23          INC HL
02E5 10F3        DJNZ 02DAH
02E7 CD2C02      CALL 022CH
02EA CD3502      CALL 0235H
02ED FE78        CP 78H
02EF 28B8        JR Z,02A9H
02F1 FE3C        CP 3CH
02F3 20F5        JR NZ,02EAH
02F5 CD3502      CALL 0235H
02F8 47          LD B,A
02F9 CD1403      CALL 0314H
02FC 85          ADD A,L
02FD 4F          LD C,A
02FE CD3502      CALL 0235H
0301 77          LD (HL),A
0302 23          INC HL
0303 81          ADD A,C
0304 4F          LD C,A
0305 10F7        DJNZ 02FEH
0307 CD3502      CALL 0235H
030A B9          CP C
030B 28DA        JR Z,02E7H
030D 3E43        LD A,43H
030F 323E3C      LD (3C3EH),A
0312 18D6        JR 02EAH
0314 CD3502      CALL 0235H
0317 6F          LD L,A
0318 CD3502      CALL 0235H
031B 67          LD H,A
031C C9          RET
031D EB          EX DE,HL
031E 2ADF40      LD HL,(40DFH)
0321 EB          EX DE,HL
0322 D7          RST 10
0323 C45A1E      CALL NZ,1E5AH
0326 208A        JR NZ,02B2H
0328 EB          EX DE,HL
0329 E9          JP (HL)
032A C5          PUSH BC
032B 4F          LD C,A
```

02B8-032B ROM 8

```
032C CDC141     CALL 41C1H
032F 3A9C40     LD A,(409CH)
0332 B7         OR A
0333 79         LD A,C
0334 C1         POP BC
0335 FA6402     JP N,0264H
0338 2062       JR NZ,039CH
033A D5         PUSH DE
033B CD3300     CALL 0033H
033E F5         PUSH AF
033F CD4803     CALL 0348H
0342 32A640     LD (40A6H),A
0345 F1         POP AF
0346 D1         POP DE
0347 C9         RET
0348 3A3D40     LD A,(403DH)
034B E608       AND 08H
034D 3A2040     LD A,(4020H)
0350 2803       JR Z,0355H
0352 0F         RRCA
0353 E61F       AND 1FH
0355 E63F       AND 3FH
0357 C9         RET
0358 CDC441     CALL 41C4H
035B D5         PUSH DE
035C CD2B00     CALL 002BH
035F D1         POP DE
0360 C9         RET
0361 AF         XOR A
0362 329940     LD (4099H),A
0365 32A640     LD (40A6H),A
0368 CDAF41     CALL 41AFH
036B C5         PUSH BC
036C 2AA740     LD HL,(40A7H)
036F 06F0       LD B,F0H
0371 CDD905     CALL 05D9H
0374 F5         PUSH AF
0375 48         LD C,B
0376 0600       LD B,00H
0378 09         ADD HL,BC
0379 3600       LD (HL),00H
037B 2AA740     LD HL,(40A7H)
037E F1         POP AF
037F C1         POP BC
0380 2B         DEC HL
0381 D8         RET C
0382 AF         XOR A
0383 C9         RET
0384 CD5803     CALL 0358H
0387 B7         OR A
0388 C0         RET NZ
0389 18F9       JR 0384H
038B AF         XOR A
038C 329C40     LD (409CH),A
038F 3A9B40     LD A,(409BH)
0392 B7         OR A
0393 C8         RET Z
0394 3E0D       LD A,0DH
0396 D5         PUSH DE
0397 CD9C03     CALL 039CH
```



```
039A D1         POP DE
039B C9         RET
039C F5         PUSH AF
039D D5         PUSH DE
039E C5         PUSH BC
039F 4F         LD C,A
03A0 1E00       LD E,00H
03A2 FE0C       CP 0CH
03A4 2810       JR Z,03B6H
03A6 FE0A       CP 0AH
03A8 2003       JR NZ,03ADH
03AA 3E0D       LD A,0DH
03AC 4F         LD C,A
03AD FE0D       CP 0DH
03AF 2805       JR Z,03B6H
03B1 3A9B40     LD A,(409BH)
03B4 3C         INC A
03B5 5F         LD E,A
03B6 7B         LD A,E
03B7 329B40     LD (409BH),A
03BA 79         LD A,C
03BB CD3B00     CALL 003BH
03BE C1         POP BC
03BF D1         POP DE
03C0 F1         POP AF
03C1 C9         RET
03C2 E5         PUSH HL
03C3 DDE5       PUSH IX
03C5 D5         PUSH DE
03C6 DDE1       POP IX
03C8 D5         PUSH DE
03C9 21DD03     LD HL,03DDH
03CC E5         PUSH HL
03CD 4F         LD C,A
03CE 1A         LD A,(DE)
03CF A0         AND B
03D0 B8         CP B
03D1 C23340     JP NZ,4033H
03D4 FE02       CP 02H
03D6 DD6E01     LD L,(IX+01H)
03D9 DD6602     LD H,(IX+02H)
03DC E9         JP (HL)
03DD D1         POP DE
03DE DDE1       POP IX
03E0 E1         POP HL
03E1 C1         POP BC
03E2 C9         RET
03E3 213640     LD HL,4036H
03E6 010138     LD BC,3801H
03E9 1600       LD D,00H
03EB 0A         LD A,(BC)
03EC 5F         LD E,A
03ED AE         XOR (HL)
03EE 73         LD (HL),E
03EF A3         AND E
03F0 2008       JR NZ,03FAH
03F2 14         INC D
03F3 2C         INC L
03F4 CB01       RLC C
03F6 F2EB03     JP P,03EBH
```

```
03F9 C9          RET
03FA 5F          LD E,A
03FB 7A          LD A,D
03FC 07          RLCA
03FD 07          RLCA
03FE 07          RLCA
03FF 57          LD D,A
0400 0E01        LD C,01H
0402 79          LD A,C
0403 A3          AND E
0404 2005        JR NZ,040BH
0406 14          INC D
0407 CB01        RLC C
0409 18F7        JR 0402H
040B 3A8038      LD A,(3880H)
040E 47          LD B,A
040F 7A          LD A,D
0410 C640        ADD A,40H
0412 FE60        CP 60H
0414 3013        JR NC,0429H
0416 CB08        RRC B
0418 3031        JR NC,044BH
041A C620        ADD A,20H
041C 57          LD D,A
041D 3A4038      LD A,(3840H)
0420 E610        AND 10H
0422 2828        JR Z,044CH
0424 7A          LD A,D
0425 D660        SUB 60H
0427 1822        JR 044BH
0429 D670        SUB 70H
042B 3010        JR NC,043DH
042D C640        ADD A,40H
042F FE3C        CP 3CH
0431 3802        JR C,0435H
0433 EE10        XOR 10H
0435 CB08        RRC B
0437 3012        JR NC,044BH
0439 EE10        XOR 10H
043B 180E        JR 044BH
043D 07          RLCA
043E CB08        RRC B
0440 3001        JR NC,0443H
0442 3C          INC A
0443 215000      LD HL,0050H
0446 4F          LD C,A
0447 0600        LD B,00H
0449 09          ADD HL,BC
044A 7E          LD A,(HL)
044B 57          LD D,A
044C 01AC0D      LD BC,0DACH
044F CD6000      CALL 0060H
0452 7A          LD A,D
0453 FE01        CP 01H
0455 C0          RET NZ
0456 EF          RST 28
0457 C9          RET
0458 DD6E03      LD L,(IX+03H)
045B DD6604      LD H,(IX+04H)
045E 383A        JR C,049AH
```



```
0460 DD7E05      LD A,(IX+05H)
0463 B7          OR A
0464 2801        JR Z,0467H
0466 77          LD (HL),A
0467 79          LD A,C
0468 FE20        CP 20H
046A DA0605      JP C,0506H
046D FE80        CP 80H
046F 3035        JR NC,04A6H
0471 FE40        CP 40H
0473 3808        JR C,047DH
0475 D640        SUB 40H
0477 FE20        CP 20H
0479 3802        JR C,047DH
047B D620        SUB 20H
047D CD4105      CALL 0541H
0480 7C          LD A,H
0481 E603        AND 03H
0483 F63C        OR 3CH
0485 67          LD H,A
0486 56          LD D,(HL)
0487 DD7E05      LD A,(IX+05H)
048A B7          OR A
048B 2805        JR Z,0492H
048D DD7205      LD (IX+05H),D
0490 365F        LD (HL),5FH
0492 DD7503      LD (IX+03H),L
0495 DD7404      LD (IX+04H),H
0498 79          LD A,C
0499 C9          RET
049A DD7E05      LD A,(IX+05H)
049D B7          OR A
049E C0          RET NZ
049F 7E          LD A,(HL)
04A0 C9          RET
04A1 7D          LD A,L
04A2 E6C0        AND C0H
04A4 6F          LD L,A
04A5 C9          RET
04A6 FEC0        CP C0H
04A8 38D3        JR C,047DH
04AA D6C0        SUB C0H
04AC 28D2        JR Z,0480H
04AE 47          LD B,A
04AF 3E20        LD A,20H
04B1 CD4105      CALL 0541H
04B4 10F9        DJNZ 04AFH
04B6 18C8        JR 0480H
04B8 7E          LD A,(HL)
04B9 DD7705      LD (IX+05H),A
04BC C9          RET
04BD AF          XOR A
04BE 18F9        JR 04B9H
04C0 21003C      LD HL,3C00H
04C3 3A3D40      LD A,(403DH)
04C6 E6F7        AND F7H
04C8 323D40      LD (403DH),A
04CB D3FF        OUT (FFH),A
04CD C9          RET
04CE 2B          DEC HL
```

```
04CF 3A3D40      LD A,(403DH)
04D2 E608        AND 08H
04D4 2801        JR Z,04D7H
04D6 2B          DEC HL
04D7 3620        LD (HL),20H
04D9 C9          RET
04DA 3A3D40      LD A,(403DH)
04DD E608        AND 08H
04DF C4E204      CALL NZ,04E2H
04E2 7D          LD A,L
04E3 E63F        AND 3FH
04E5 2B          DEC HL
04E6 C0          RET NZ
04E7 114000      LD DE,0040H
04EA 19          ADD HL,DE
04EB C9          RET
04EC 23          INC HL
04ED 7D          LD A,L
04EE E63F        AND 3FH
04F0 C0          RET NZ
04F1 11C0FF      LD DE,FFC0H
04F4 19          ADD HL,DE
04F5 C9          RET
04F6 3A3D40      LD A,(403DH)
04F9 F608        OR 08H
04FB 323D40      LD (403DH),A
04FE D3FF        OUT (FFH),A
0500 23          INC HL
0501 7D          LD A,L
0502 E6FE        AND FEH
0504 6F          LD L,A
0505 C9          RET
0506 118004      LD DE,0480H
0509 D5          PUSH DE
050A FE08        CP 08H
050C 28C0        JR Z,04CEH
050E FE0A        CP 0AH
0510 D8          RET C
0511 FE0E        CP 0EH
0513 384F        JR C,0564H
0515 28A1        JR Z,04B8H
0517 FE0F        CP 0FH
0519 28A2        JR Z,04BDH
051B FE17        CP 17H
051D 28D7        JR Z,04F6H
051F FE18        CP 18H
0521 28B7        JR Z,04DAH
0523 FE19        CP 19H
0525 28C5        JR Z,04ECH
0527 FE1A        CP 1AH
0529 28BC        JR Z,04E7H
052B FE1B        CP 1BH
052D 28C2        JR Z,04F1H
052F FE1C        CP 1CH
0531 288D        JR Z,04C0H
0533 FE1D        CP 1DH
0535 CAA104      JP Z,04A1H
0538 FE1E        CP 1EH
053A 2837        JR Z,0573H
053C FE1F        CP 1FH
```



```
053E 283C        JR Z,057CH
0540 C9          RET
0541 77          LD (HL),A
0542 23          INC HL
0543 3A3D40      LD A,(403DH)
0546 E608        AND 08H
0548 2801        JR Z,054BH
054A 23          INC HL
054B 7C          LD A,H
054C FE40        CP 40H
054E C0          RET NZ
054F 11C0FF      LD DE,FFC0H
0552 19          ADD HL,DE
0553 E5          PUSH HL
0554 11003C      LD DE,3C00H
0557 21403C      LD HL,3C40H
055A C5          PUSH BC
055B 01C003      LD BC,03C0H
055E EDB0        LDIR
0560 C1          POP BC
0561 EB          EX DE,HL
0562 1819        JR 057DH
0564 7D          LD A,L
0565 E6C0        AND C0H
0567 6F          LD L,A
0568 E5          PUSH HL
0569 114000      LD DE,0040H
056C 19          ADD HL,DE
056D 7C          LD A,H
056E FE40        CP 40H
0570 28E2        JR Z,0554H
0572 D1          POP DE
0573 E5          PUSH HL
0574 54          LD D,H
0575 7D          LD A,L
0576 F63F        OR 3FH
0578 5F          LD E,A
0579 13          INC DE
057A 1804        JR 0580H
057C E5          PUSH HL
057D 110040      LD DE,4000H
0580 3620        LD (HL),20H
0582 23          INC HL
0583 7C          LD A,H
0584 BA          CP D
0585 20F9        JR NZ,0580H
0587 7D          LD A,L
0588 BB          CP E
0589 20F5        JR NZ,0580H
058B E1          POP HL
058C C9          RET
058D 79          LD A,C
058E B7          OR A
058F 2840        JR Z,05D1H
0591 FE0B        CP 0BH
0593 280A        JR Z,059FH
0595 FE0C        CP 0CH
0597 201B        JR NZ,05B4H
0599 AF          XOR A
059A DDB603      OR (IX+03H)
```

```
059D 2815        JR Z,05B4H
059F DD7E03      LD A,(IX+03H)
05A2 DD9604      SUB (IX+04H)
05A5 47          LD B,A
05A6 CDD105      CALL 05D1H
05A9 20FB        JR NZ,05A6H
05AB 3E0A        LD A,0AH
05AD 00          NOP
05AE D3FD        OUT (FDH),A
05B0 10F4        DJNZ 05A6H
05B2 1818        JR 05CCH
05B4 F5          PUSH AF
05B5 CDD105      CALL 05D1H
05B8 20FB        JR NZ,05B5H
05BA F1          POP AF
05BB 00          NOP
05BC D3FD        OUT (FDH),A
05BE FE0D        CP 0DH
05C0 C0          RET NZ
05C1 DD3404      INC (IX+04H)
05C4 DD7E04      LD A,(IX+04H)
05C7 DDBE03      CP (IX+03H)
05CA 79          LD A,C
05CB C0          RET NZ
05CC DD360400    LD (IX+04H),00H
05D0 C9          RET
05D1 00          NOP
05D2 DBFD        IN A,(FDH)
05D4 E6F0        AND F0H
05D6 FE30        CP 30H
05D8 C9          RET
05D9 E5          PUSH HL
05DA 3E0E        LD A,0EH
05DC CD3300      CALL 0033H
05DF 48          LD C,B
05E0 CD4900      CALL 0049H
05E3 FE20        CP 20H
05E5 3025        JR NC,060CH
05E7 FE0D        CP 0DH
05E9 CA6206      JP Z,0662H
05EC FE1F        CP 1FH
05EE 2829        JR Z,0619H
05F0 FE01        CP 01H
05F2 286D        JR Z,0661H
05F4 11E005      LD DE,05E0H
05F7 D5          PUSH DE
05F8 FE08        CP 08H
05FA 2834        JR Z,0630H
05FC FE18        CP 18H
05FE 282B        JR Z,062BH
0600 FE09        CP 09H
0602 2842        JR Z,0646H
0604 FE19        CP 19H
0606 2839        JR Z,0641H
0608 FE0A        CP 0AH
060A C0          RET NZ
060B D1          POP DE
060C 77          LD (HL),A
060D 78          LD A,B
060E B7          OR A
```

059D-060E ROM 15

```
060F 28CF        JR Z,05E0H
0611 7E          LD A,(HL)
0612 23          INC HL
0613 CD3300      CALL 0033H
0616 05          DEC B
0617 18C7        JR 05E0H
0619 CDC901      CALL 01C9H
061C 41          LD B,C
061D E1          POP HL
061E E5          PUSH HL
061F C3E005      JP 05E0H
0622 CD3006      CALL 0630H
0625 2B          DEC HL
0626 7E          LD A,(HL)
0627 23          INC HL
0628 FE0A        CP 0AH
062A C8          RET Z
062B 78          LD A,B
062C B9          CP C
062D 20F3        JR NZ,0622H
062F C9          RET
0630 78          LD A,B
0631 B9          CP C
0632 C8          RET Z
0633 2B          DEC HL
0634 7E          LD A,(HL)
0635 FE0A        CP 0AH
0637 23          INC HL
0638 C8          RET Z
0639 2B          DEC HL
063A 3E08        LD A,08H
063C CD3300      CALL 0033H
063F 04          INC B
0640 C9          RET
0641 3E17        LD A,17H
0643 C33300      JP 0033H
0646 CD4803      CALL 0348H
0649 E607        AND 07H
064B 2F          CPL
064C 3C          INC A
064D C608        ADD A,08H
064F 5F          LD E,A
0650 78          LD A,B
0651 B7          OR A
0652 C8          RET Z
0653 3E20        LD A,20H
0655 77          LD (HL),A
0656 23          INC HL
0657 D5          PUSH DE
0658 CD3300      CALL 0033H
065B D1          POP DE
065C 05          DEC B
065D 1D          DEC E
065E C8          RET Z
065F 18EF        JR 0650H
0661 37          SCF
0662 F5          PUSH AF
0663 3E0D        LD A,0DH
0665 77          LD (HL),A
0666 CD3300      CALL 0033H
```

060F-0666 ROM 16

```
0669 3E0F        LD A,0FH
066B CD3300      CALL 0033H
066E 79          LD A,C
066F 90          SUB B
0670 47          LD B,A
0671 F1          POP AF
0672 E1          POP HL
0673 C9          RET
0674 D3FF        OUT (FFH),A
0676 21D206      LD HL,06D2H
0679 110040      LD DE,4000H
067C 013600      LD BC,0036H
067F EDB0        LDIR
0681 3D          DEC A
0682 3D          DEC A
0683 20F1        JR NZ,0676H
0685 0627        LD B,27H
0687 12          LD (DE),A
0688 13          INC DE
0689 10FC        DJNZ 0687H
068B 3A4038      LD A,(3840H)
068E E604        AND 04H
0690 C27500      JP NZ,0075H
0693 317D40      LD SP,407DH
0696 3AEC37      LD A,(37ECH)
0699 3C          INC A
069A FE02        CP 02H
069C DA7500      JP C,0075H
069F 3E01        LD A,01H
06A1 32E137      LD (37E1H),A
06A4 21EC37      LD HL,37ECH
06A7 11EF37      LD DE,37EFH
06AA 3603        LD (HL),03H
06AC 010000      LD BC,0000H
06AF CD6000      CALL 0060H
06B2 CB46        BIT 0,(HL)
06B4 20FC        JR NZ,06B2H
06B6 AF          XOR A
06B7 32EE37      LD (37EEH),A
06BA 010042      LD BC,4200H
06BD 3E8C        LD A,8CH
06BF 77          LD (HL),A
06C0 CB4E        BIT 1,(HL)
06C2 28FC        JR Z,06C0H
06C4 1A          LD A,(DE)
06C5 02          LD (BC),A
06C6 0C          INC C
06C7 20F7        JR NZ,06C0H
06C9 C30042      JP 4200H
```

0669-06C9 ROM 17

```
06CC 01          .   .
06CD 18          .   .
06CE 1A          .   .
06CF C3          .   C
06D0 AE          .   .
06D1 19          .   .
06D2 C3          .   C
06D3 96          .   .
```

0669-06D3 ROM 17

```
06D4 1C          .   .
06D5 C3          .   C
06D6 78          x   x
06D7 1D          .   .
06D8 C3          .   C
06D9 90          .   .
06DA 1C          .   .
06DB C3          .   C
06DC D9          .   Y
06DD 25          %   %
06DE C9          .   I
06DF 00          .   .
06E0 00          .   .
06E1 C9          .   I
06E2 00          .   .
06E3 00          .   .
06E4 FB          .   {
06E5 C9          .   I
06E6 00          .   .
06E7 01          .   .
06E8 E3          .   c
06E9 03          .   .
06EA 00          .   .
06EB 00          .   .
06EC 00          .   .
06ED 4B          K   K
06EE 49          I   I
06EF 07          .   .
06F0 58          X   X
06F1 04          .   .
06F2 00          .   .
06F3 3C          <   <
06F4 00          .   .
06F5 44          D   D
06F6 4F          O   O
06F7 06          .   .
06F8 8D          .   .
06F9 05          .   .
06FA 43          C   C
06FB 00          .   .
06FC 00          .   .
06FD 50          P   P
06FE 52          R   R
06FF C3          .   C
0700 00          .   .
0701 50          P   P
0702 C7          .   G
0703 00          .   .
0704 00          .   .
0705 3E          >   >
0706 00          .   .
0707 C9          .   I
```

06D4-0707 ROM 18

```
0708 218013      LD HL,1380H
070B CDC209      CALL 09C2H
070E 1806        JR 0716H
0710 CDC209      CALL 09C2H
0713 CD8209      CALL 0982H
```

06D4-0713 ROM 18

```
0716 78          LD A,B
0717 B7          OR A
0718 C8          RET Z
0719 3A2441      LD A,(4124H)
071C B7          OR A
071D CAB409      JP Z,09B4H
0720 90          SUB B
0721 300C        JR NC,072FH
0723 2F          CPL
0724 3C          INC A
0725 EB          EX DE,HL
0726 CDA409      CALL 09A4H
0729 EB          EX DE,HL
072A CDB409      CALL 09B4H
072D C1          POP BC
072E D1          POP DE
072F FE19        CP 19H
0731 D0          RET NC
0732 F5          PUSH AF
0733 CDDF09      CALL 09DFH
0736 67          LD H,A
0737 F1          POP AF
0738 CDD707      CALL 07D7H
073B B4          OR H
073C 212141      LD HL,4121H
073F F25407      JP P,0754H
0742 CDB707      CALL 07B7H
0745 D29607      JP NC,0796H
0748 23          INC HL
0749 34          INC (HL)
074A CAB207      JP Z,07B2H
074D 2E01        LD L,01H
074F CDEB07      CALL 07EBH
0752 1842        JR 0796H
0754 AF          XOR A
0755 90          SUB B
0756 47          LD B,A
0757 7E          LD A,(HL)
0758 9B          SBC A,E
0759 5F          LD E,A
075A 23          INC HL
075B 7E          LD A,(HL)
075C 9A          SBC A,D
075D 57          LD D,A
075E 23          INC HL
075F 7E          LD A,(HL)
0760 99          SBC A,C
0761 4F          LD C,A
0762 DCC307      CALL C,07C3H
0765 68          LD L,B
0766 63          LD H,E
0767 AF          XOR A
0768 47          LD B,A
0769 79          LD A,C
076A B7          OR A
076B 2018        JR NZ,0785H
076D 4A          LD C,D
076E 54          LD D,H
076F 65          LD H,L
0770 6F          LD L,A
```

0716-0770 ROM 19

```
0771 78          LD A,B
0772 D608        SUB 08H
0774 FEE0        CP E0H
0776 20F0        JR NZ,0768H
0778 AF          XOR A
0779 322441      LD (4124H),A
077C C9          RET
077D 05          DEC B
077E 29          ADD HL,HL
077F 7A          LD A,D
0780 17          RLA
0781 57          LD D,A
0782 79          LD A,C
0783 8F          ADC A,A
0784 4F          LD C,A
0785 F27D07      JP P,077DH
0788 78          LD A,B
0789 5C          LD E,H
078A 45          LD B,L
078B B7          OR A
078C 2808        JR Z,0796H
078E 212441      LD HL,4124H
0791 86          ADD A,(HL)
0792 77          LD (HL),A
0793 30E3.       JR NC,0778H
0795 C8          RET Z
0796 78          LD A,B
0797 212441      LD HL,4124H
079A B7          OR A
079B FCA807      CALL N,07A8H
079E 46          LD B,(HL)
079F 23          INC HL
07A0 7E          LD A,(HL)
07A1 E680        AND 80H
07A3 A9          XOR C
07A4 4F          LD C,A
07A5 C3B409      JP 09B4H
07A8 1C          INC E
07A9 C0          RET NZ
07AA 14          INC D
07AB C0          RET NZ
07AC 0C          INC C
07AD C0          RET NZ
07AE 0E80        LD C,80H
07B0 34          INC (HL)
07B1 C0          RET NZ
07B2 1E0A        LD E,0AH
07B4 C3A219      JP 19A2H
07B7 7E          LD A,(HL)
07B8 83          ADD A,E
07B9 5F          LD E,A
07BA 23          INC HL
07BB 7E          LD A,(HL)
07BC 8A          ADC A,D
07BD 57          LD D,A
07BE 23          INC HL
07BF 7E          LD A,(HL)
07C0 89          ADC A,C
07C1 4F          LD C,A
07C2 C9          RET
```

0771-07C2 ROM 20

```
07C3 212541      LD HL,4125H
07C6 7E          LD A,(HL)
07C7 2F          CPL
07C8 77          LD (HL),A
07C9 AF          XOR A
07CA 6F          LD L,A
07CB 90          SUB B
07CC 47          LD B,A
07CD 7D          LD A,L
07CE 9B          SBC A,E
07CF 5F          LD E,A
07D0 7D          LD A,L
07D1 9A          SBC A,D
07D2 57          LD D,A
07D3 7D          LD A,L
07D4 99          SBC A,C
07D5 4F          LD C,A
07D6 C9          RET
07D7 0600        LD B,00H
07D9 D608        SUB 08H
07DB 3807        JR C,07E4H
07DD 43          LD B,E
07DE 5A          LD E,D
07DF 51          LD D,C
07E0 0E00        LD C,00H
07E2 18F5        JR 07D9H
07E4 C609        ADD A,09H
07E6 6F          LD L,A
07E7 AF          XOR A
07E8 2D          DEC L
07E9 C8          RET Z
07EA 79          LD A,C
07EB 1F          RRA
07EC 4F          LD C,A
07ED 7A          LD A,D
07EE 1F          RRA
07EF 57          LD D,A
07F0 7B          LD A,E
07F1 1F          RRA
07F2 5F          LD E,A
07F3 78          LD A,B
07F4 1F          RRA
07F5 47          LD B,A
07F6 18EF        JR 07E7H
```

07C3-07F6 ROM 21

```
07F8  00         .    .
07F9  00         .    .
07FA  00         .    .
07FB  81         .    .
07FC  03         .    .
07FD  AA         .    *
07FE  56         V    V
07FF  19         .    .
0800  80         .    .
0801  F1         .    q
0802  22         "    "
0803  76         v    v
0804  80         .    .
```

07C3-0804 ROM 21

```
0805  45         E    E
0806  AA         .    *
0807  38         8    8
0808  82         .    .
```

0805-0808 ROM 22

```
0809 CD5509      CALL 0955H
080C B7          OR A
080D EA4A1E      JP PE,1E4AH
0810 212441      LD HL,4124H
0813 7E          LD A,(HL)
0814 013580      LD BC,8035H
0817 11F304      LD DE,04F3H
081A 90          SUB B
081B F5          PUSH AF
081C 70          LD (HL),B
081D D5          PUSH DE
081E C5          PUSH BC
081F CD1607      CALL 0716H
0822 C1          POP BC
0823 D1          POP DE
0824 04          INC B
0825 CDA208      CALL 08A2H
0828 21F807      LD HL,07F8H
082B CD1007      CALL 0710H
082E 21FC07      LD HL,07FCH
0831 CD9A14      CALL 149AH
0834 018080      LD BC,8080H
0837 110000      LD DE,0000H
083A CD1607      CALL 0716H
083D F1          POP AF
083E CD890F      CALL 0F89H
0841 013180      LD BC,8031H
0844 111872      LD DE,7218H
0847 CD5509      CALL 0955H
084A C8          RET Z
084B 2E00        LD L,00H
084D CD1409      CALL 0914H
0850 79          LD A,C
0851 324F41      LD (414FH),A
0854 EB          EX DE,HL
0855 225041      LD (4150H),HL
0858 010000      LD BC,0000H
085B 50          LD D,B
085C 58          LD E,B
085D 216507      LD HL,0765H
0860 E5          PUSH HL
0861 216908      LD HL,0869H
0864 E5          PUSH HL
0865 E5          PUSH HL
0866 212141      LD HL,4121H
0869 7E          LD A,(HL)
086A 23          INC HL
086B B7          OR A
086C 2824        JR Z,0892H
086E E5          PUSH HL
086F 2E08        LD L,08H
0871 1F          RRA
0872 67          LD H,A
```

0805-0872 ROM 22

```
0873 79          LD A,C
0874 300B        JR NC,0881H
0876 E5          PUSH HL
0877 2A5041      LD HL,(4150H)
087A 19          ADD HL,DE
087B EB          EX DE,HL
087C E1          POP HL
087D 3A4F41      LD A,(414FH)
0880 89          ADC A,C
0881 1F          RRA
0882 4F          LD C,A
0883 7A          LD A,D
0884 1F          RRA
0885 57          LD D,A
0886 7B          LD A,E
0887 1F          RRA
0888 5F          LD E,A
0889 78          LD A,B
088A 1F          RRA
088B 47          LD B,A
088C 2D          DEC L
088D 7C          LD A,H
088E 20E1        JR NZ,0871H
0890 E1          POP HL
0891 C9          RET
0892 43          LD B,E
0893 5A          LD E,D
0894 51          LD D,C
0895 4F          LD C,A
0896 C9          RET
0897 CDA409      CALL 09A4H
089A 21D80D      LD HL,0DD8H
089D CDB109      CALL 09B1H
08A0 C1          POP BC
08A1 D1          POP DE
08A2 CD5509      CALL 0955H
08A5 CA9A19      JP Z,199AH
08A8 2EFF        LD L,FFH
08AA CD1409      CALL 0914H
08AD 34          INC (HL)
08AE 34          INC (HL)
08AF 2B          DEC HL
08B0 7E          LD A,(HL)
08B1 328940      LD (4089H),A
08B4 2B          DEC HL
08B5 7E          LD A,(HL)
08B6 328540      LD (4085H),A
08B9 2B          DEC HL
08BA 7E          LD A,(HL)
08BB 328140      LD (4081H),A
08BE 41          LD B,C
08BF EB          EX DE,HL
08C0 AF          XOR A
08C1 4F          LD C,A
08C2 57          LD D,A
08C3 5F          LD E,A
08C4 328C40      LD (408CH),A
08C7 E5          PUSH HL
08C8 C5          PUSH BC
08C9 7D          LD A,L
```



```
08CA CD8040      CALL 4080H
08CD DE00        SBC A,00H
08CF 3F          CCF
08D0 3007        JR NC,08D9H
08D2 328C40      LD (408CH),A
08D5 F1          POP AF
08D6 F1          POP AF
08D7 37          SCF
08D8 D2C1E1      JP NC,E1C1H
08DB 79          LD A,C
08DC 3C          INC A
08DD 3D          DEC A
08DE 1F          RRA
08DF FA9707      JP N,0797H
08E2 17          RLA
08E3 7B          LD A,E
08E4 17          RLA
08E5 5F          LD E,A
08E6 7A          LD A,D
08E7 17          RLA
08E8 57          LD D,A
08E9 79          LD A,C
08EA 17          RLA
08EB 4F          LD C,A
08EC 29          ADD HL,HL
08ED 78          LD A,B
08EE 17          RLA
08EF 47          LD B,A
08F0 3A8C40      LD A,(408CH)
08F3 17          RLA
08F4 328C40      LD (408CH),A
08F7 79          LD A,C
08F8 B2          OR D
08F9 B3          OR E
08FA 20CB        JR NZ,08C7H
08FC E5          PUSH HL
08FD 212441      LD HL,4124H
0900 35          DEC (HL)
0901 E1          POP HL
0902 20C3        JR NZ,08C7H
0904 C3B207      JP 07B2H
0907 3EFF        LD A,FFH
0909 2EAF        LD L,AFH
090B 212D41      LD HL,412DH
090E 4E          LD C,(HL)
090F 23          INC HL
0910 AE          XOR (HL)
0911 47          LD B,A
0912 2E00        LD L,00H
0914 78          LD A,B
0915 B7          OR A
0916 281F        JR Z,0937H
0918 7D          LD A,L
0919 212441      LD HL,4124H
091C AE          XOR (HL)
091D 80          ADD A,B
091E 47          LD B,A
091F 1F          RRA
0920 A8          XOR B
0921 78          LD A,B
```

```
0922 F23609      JP P,0936H
0925 C680        ADD A,80H
0927 77          LD (HL),A
0928 CA9008      JP Z,0890H
092B CDDF09      CALL 09DFH
092E 77          LD (HL),A
092F 2B          DEC HL
0930 C9          RET
0931 CD5509      CALL 0955H
0934 2F          CPL
0935 E1          POP HL
0936 B7          OR A
0937 E1          POP HL
0938 F27807      JP P,0778H
093B C3B207      JP 07B2H
093E CDBF09      CALL 09BFH
0941 78          LD A,B
0942 B7          OR A
0943 C8          RET Z
0944 C602        ADD A,02H
0946 DAB207      JP C,07B2H
0949 47          LD B,A
094A CD1607      CALL 0716H
094D 212441      LD HL,4124H
0950 34          INC (HL)
0951 C0          RET NZ
0952 C3B207      JP 07B2H
0955 3A2441      LD A,(4124H)
0958 B7          OR A
0959 C8          RET Z
095A 3A2341      LD A,(4123H)
095D FE2F        CP 2FH
095F 17          RLA
0960 9F          SBC A,A
0961 C0          RET NZ
0962 3C          INC A
0963 C9          RET
0964 0688        LD B,88H
0966 110000      LD DE,0000H
0969 212441      LD HL,4124H
096C 4F          LD C,A
096D 70          LD (HL),B
096E 0600        LD B,00H
0970 23          INC HL
0971 3680        LD (HL),80H
0973 17          RLA
0974 C36207      JP 0762H
0977 CD9409      CALL 0994H
097A F0          RET P
097B E7          RST 20
097C FA5B0C      JP N,0C5BH
097F CAF60A      JP Z,0AF6H
0982 212341      LD HL,4123H
0985 7E          LD A,(HL)
0986 EE80        XOR 80H
0988 77          LD (HL),A
0989 C9          RET
098A CD9409      CALL 0994H
098D 6F          LD L,A
098E 17          RLA
```

0922-098E ROM 25

```
098F 9F          SBC A,A
0990 67          LD H,A
0991 C39A0A      JP 0A9AH
0994 E7          RST 20
0995 CAF60A      JP Z,0AF6H
0998 F25509      JP P,0955H
099B 2A2141      LD HL,(4121H)
099E 7C          LD A,H
099F B5          OR L
09A0 C8          RET Z
09A1 7C          LD A,H
09A2 18BB        JR 095FH
09A4 EB          EX DE,HL
09A5 2A2141      LD HL,(4121H)
09A8 E3          EX (SP),HL
09A9 E5          PUSH HL
09AA 2A2341      LD HL,(4123H)
09AD E3          EX (SP),HL
09AE E5          PUSH HL
09AF EB          EX DE,HL
09B0 C9          RET
09B1 CDC209      CALL 09C2H
09B4 EB          EX DE,HL
09B5 222141      LD (4121H),HL
09B8 60          LD H,B
09B9 69          LD L,C
09BA 222341      LD (4123H),HL
09BD EB          EX DE,HL
09BE C9          RET
09BF 212141      LD HL,4121H
09C2 5E          LD E,(HL)
09C3 23          INC HL
09C4 56          LD D,(HL)
09C5 23          INC HL
09C6 4E          LD C,(HL)
09C7 23          INC HL
09C8 46          LD B,(HL)
09C9 23          INC HL
09CA C9          RET
09CB 112141      LD DE,4121H
09CE 0604        LD B,04H
09D0 1805        JR 09D7H
09D2 EB          EX DE,HL
09D3 3AAF40      LD A,(40AFH)
09D6 47          LD B,A
09D7 1A          LD A,(DE)
09D8 77          LD (HL),A
09D9 13          INC DE
09DA 23          INC HL
09DB 05          DEC B
09DC 20F9        JR NZ,09D7H
09DE C9          RET
09DF 212341      LD HL,4123H
09E2 7E          LD A,(HL)
09E3 07          RLCA
09E4 37          SCF
09E5 1F          RRA
09E6 77          LD (HL),A
09E7 3F          CCF
09E8 1F          RRA
```

098F-09E8 ROM 26

```
09E9 23          INC HL
09EA 23          INC HL
09EB 77          LD (HL),A
09EC 79          LD A,C
09ED 07          RLCA
09EE 37          SCF
09EF 1F          RRA
09F0 4F          LD C,A
09F1 1F          RRA
09F2 AE          XOR (HL)
09F3 C9          RET
09F4 212741      LD HL,4127H
09F7 11D209      LD DE,09D2H
09FA 1806        JR 0A02H
09FC 212741      LD HL,4127H
09FF 11D309      LD DE,09D3H
0A02 D5          PUSH DE
0A03 112141      LD DE,4121H
0A06 E7          RST 20
0A07 D8          RET C
0A08 111D41      LD DE,411DH
0A0B C9          RET
0A0C 78          LD A,B
0A0D B7          OR A
0A0E CA5509      JP Z,0955H
0A11 215E09      LD HL,095EH
0A14 E5          PUSH HL
0A15 CD5509      CALL 0955H
0A18 79          LD A,C
0A19 C8          RET Z
0A1A 212341      LD HL,4123H
0A1D AE          XOR (HL)
0A1E 79          LD A,C
0A1F F8          RET N
0A20 CD260A      CALL 0A26H
0A23 1F          RRA
0A24 A9          XOR C
0A25 C9          RET
0A26 23          INC HL
0A27 78          LD A,B
0A28 BE          CP (HL)
0A29 C0          RET NZ
0A2A 2B          DEC HL
0A2B 79          LD A,C
0A2C BE          CP (HL)
0A2D C0          RET NZ
0A2E 2B          DEC HL
0A2F 7A          LD A,D
0A30 BE          CP (HL)
0A31 C0          RET NZ
0A32 2B          DEC HL
0A33 7B          LD A,E
0A34 96          SUB (HL)
0A35 C0          RET NZ
0A36 E1          POP HL
0A37 E1          POP HL
0A38 C9          RET
0A39 7A          LD A,D
0A3A AC          XOR H
0A3B 7C          LD A,H
```

09E9-0A3B ROM 27

```
0A3C FA5F09      JP N,095FH
0A3F BA          CP D
0A40 C26009      JP NZ,0960H
0A43 7D          LD A,L
0A44 93          SUB E
0A45 C26009      JP NZ,0960H
0A48 C9          RET
0A49 212741      LD HL,4127H
0A4C CDD309      CALL 09D3H
0A4F 112E41      LD DE,412EH
0A52 1A          LD A,(DE)
0A53 B7          OR A
0A54 CA5509      JP Z,0955H
0A57 215E09      LD HL,095EH
0A5A E5          PUSH HL
0A5B CD5509      CALL 0955H
0A5E 1B          DEC DE
0A5F 1A          LD A,(DE)
0A60 4F          LD C,A
0A61 C8          RET Z
0A62 212341      LD HL,4123H
0A65 AE          XOR (HL)
0A66 79          LD A,C
0A67 F8          RET N
0A68 13          INC DE
0A69 23          INC HL
0A6A 0608        LD B,08H
0A6C 1A          LD A,(DE)
0A6D 96          SUB (HL)
0A6E C2230A      JP NZ,0A23H
0A71 1B          DEC DE
0A72 2B          DEC HL
0A73 05          DEC B
0A74 20F6        JR NZ,0A6CH
0A76 C1          POP BC
0A77 C9          RET
0A78 CD4F0A      CALL 0A4FH
0A7B C25E09      JP NZ,095EH
0A7E C9          RET
0A7F E7          RST 20
0A80 2A2141      LD HL,(4121H)
0A83 F8          RET N
0A84 CAF60A      JP Z,0AF6H
0A87 D4B90A      CALL NC,0AB9H
0A8A 21B207      LD HL,07B2H
0A8D E5          PUSH HL
0A8E 3A2441      LD A,(4124H)
0A91 FE90        CP 90H
0A93 300E        JR NC,0AA3H
0A95 CDFB0A      CALL 0AFBH
0A98 EB          EX DE,HL
0A99 D1          POP DE
0A9A 222141      LD (4121H),HL
0A9D 3E02        LD A,02H
0A9F 32AF40      LD (40AFH),A
0AA2 C9          RET
0AA3 018090      LD BC,9080H
0AA6 110000      LD DE,0000H
0AA9 CD0C0A      CALL 0A0CH
0AAC C0          RET NZ
```

0A3C-0AAC ROM 28

```
0AAD 61          LD H,C
0AAE 6A          LD L,D
0AAF 18E8        JR 0A99H
0AB1 E7          RST 20
0AB2 E0          RET PO
0AB3 FACC0A      JP N,0ACCH
0AB6 CAF60A      JP Z,0AF6H
0AB9 CDBF09      CALL 09BFH
0ABC CDEF0A      CALL 0AEFH
0ABF 78          LD A,B
0AC0 B7          OR A
0AC1 C8          RET Z
0AC2 CDDF09      CALL 09DFH
0AC5 212041      LD HL,4120H
0AC8 46          LD B,(HL)
0AC9 C39607      JP 0796H
0ACC 2A2141      LD HL,(4121H)
0ACF CDEF0A      CALL 0AEFH
0AD2 7C          LD A,H
0AD3 55          LD D,L
0AD4 1E00        LD E,00H
0AD6 0690        LD B,90H
0AD8 C36909      JP 0969H
0ADB E7          RST 20
0ADC D0          RET NC
0ADD CAF60A      JP Z,0AF6H
0AE0 FCCC0A      CALL N,0ACCH
0AE3 210000      LD HL,0000H
0AE6 221D41      LD (411DH),HL
0AE9 221F41      LD (411FH),HL
0AEC 3E08        LD A,08H
0AEE 013E04      LD BC,043EH
0AF1 C39F0A      JP 0A9FH
0AF4 E7          RST 20
0AF5 C8          RET Z
0AF6 1E18        LD E,18H
0AF8 C3A219      JP 19A2H
0AFB 47          LD B,A
0AFC 4F          LD C,A
0AFD 57          LD D,A
0AFE 5F          LD E,A
0AFF B7          OR A
0B00 C8          RET Z
0B01 E5          PUSH HL
0B02 CDBF09      CALL 09BFH
0B05 CDDF09      CALL 09DFH
0B08 AE          XOR (HL)
0B09 67          LD H,A
0B0A FC1F0B      CALL N,0B1FH
0B0D 3E98        LD A,98H
0B0F 90          SUB B
0B10 CDD707      CALL 07D7H
0B13 7C          LD A,H
0B14 17          RLA
0B15 DCA807      CALL C,07A8H
0B18 0600        LD B,00H
0B1A DCC307      CALL C,07C3H
0B1D E1          POP HL
0B1E C9          RET
0B1F 1B          DEC DE
```

0AAD-0B1F ROM 29

```
0B20 7A          LD A,D
0B21 A3          AND E
0B22 3C          INC A
0B23 C0          RET NZ
0B24 0B          DEC BC
0B25 C9          RET
0B26 E7          RST 20
0B27 F8          RET N
0B28 CD5509      CALL 0955H
0B2B F2370B      JP P,0B37H
0B2E CD8209      CALL 0982H
0B31 CD370B      CALL 0B37H
0B34 C37B09      JP 097BH
0B37 E7          RST 20
0B38 F8          RET N
0B39 301E        JR NC,0B59H
0B3B 28B9        JR Z,0AF6H
0B3D CD8E0A      CALL 0A8EH
0B40 212441      LD HL,4124H
0B43 7E          LD A,(HL)
0B44 FE98        CP 98H
0B46 3A2141      LD A,(4121H)
0B49 D0          RET NC
0B4A 7E          LD A,(HL)
0B4B CDFB0A      CALL 0AFBH
0B4E 3698        LD (HL),98H
0B50 7B          LD A,E
0B51 F5          PUSH AF
0B52 79          LD A,C
0B53 17          RLA
0B54 CD6207      CALL 0762H
0B57 F1          POP AF
0B58 C9          RET
0B59 212441      LD HL,4124H
0B5C 7E          LD A,(HL)
0B5D FE90        CP 90H
0B5F DA7F0A      JP C,0A7FH
0B62 2014        JR NZ,0B78H
0B64 4F          LD C,A
0B65 2B          DEC HL
0B66 7E          LD A,(HL)
0B67 EE80        XOR 80H
0B69 0606        LD B,06H
0B6B 2B          DEC HL
0B6C B6          OR (HL)
0B6D 05          DEC B
0B6E 20FB        JR NZ,0B6BH
0B70 B7          OR A
0B71 210080      LD HL,8000H
0B74 CA9A0A      JP Z,0A9AH
0B77 79          LD A,C
0B78 FEB8        CP B8H
0B7A D0          RET NC
0B7B F5          PUSH AF
0B7C CDBF09      CALL 09BFH
0B7F CDDF09      CALL 09DFH
0B82 AE          XOR (HL)
0B83 2B          DEC HL
0B84 36B8        LD (HL),B8H
0B86 F5          PUSH AF
```

0B20-0B86 ROM 30

```
0B87 FCA00B      CALL N,0BA0H
0B8A 212341      LD HL,4123H
0B8D 3EB8        LD A,B8H
0B8F 90          SUB B
0B90 CD690D      CALL 0D69H
0B93 F1          POP AF
0B94 FC200D      CALL N,0D20H
0B97 AF          XOR A
0B98 321C41      LD (411CH),A
0B9B F1          POP AF
0B9C D0          RET NC
0B9D C3D80C      JP 0CD8H
0BA0 211D41      LD HL,411DH
0BA3 7E          LD A,(HL)
0BA4 35          DEC (HL)
0BA5 B7          OR A
0BA6 23          INC HL
0BA7 28FA        JR Z,0BA3H
0BA9 C9          RET
0BAA E5          PUSH HL
0BAB 210000      LD HL,0000H
0BAE 78          LD A,B
0BAF B1          OR C
0BB0 2812        JR Z,0BC4H
0BB2 3E10        LD A,10H
0BB4 29          ADD HL,HL
0BB5 DA3D27      JP C,273DH
0BB8 EB          EX DE,HL
0BB9 29          ADD HL,HL
0BBA EB          EX DE,HL
0BBB 3004        JR NC,0BC1H
0BBD 09          ADD HL,BC
0BBE DA3D27      JP C,273DH
0BC1 3D          DEC A
0BC2 20F0        JR NZ,0BB4H
0BC4 EB          EX DE,HL
0BC5 E1          POP HL
0BC6 C9          RET
0BC7 7C          LD A,H
0BC8 17          RLA
0BC9 9F          SBC A,A
0BCA 47          LD B,A
0BCB CD510C      CALL 0C51H
0BCE 79          LD A,C
0BCF 98          SBC A,B
0BD0 1803        JR 0BD5H
0BD2 7C          LD A,H
0BD3 17          RLA
0BD4 9F          SBC A,A
0BD5 47          LD B,A
0BD6 E5          PUSH HL
0BD7 7A          LD A,D
0BD8 17          RLA
0BD9 9F          SBC A,A
0BDA 19          ADD HL,DE
0BDB 88          ADC A,B
0BDC 0F          RRCA
0BDD AC          XOR H
0BDE F2990A      JP P,0A99H
0BE1 C5          PUSH BC
```



```
0BE2 EB          EX DE,HL
0BE3 CDCF0A      CALL 0ACFH
0BE6 F1          POP AF
0BE7 E1          POP HL
0BE8 CDA409      CALL 09A4H
0BEB EB          EX DE,HL
0BEC CD6B0C      CALL 0C6BH
0BEF C38F0F      JP 0F8FH
0BF2 7C          LD A,H
0BF3 B5          OR L
0BF4 CA9A0A      JP Z,0A9AH
0BF7 E5          PUSH HL
0BF8 D5          PUSH DE
0BF9 CD450C      CALL 0C45H
0BFC C5          PUSH BC
0BFD 44          LD B,H
0BFE 4D          LD C,L
0BFF 210000      LD HL,0000H
0C02 3E10        LD A,10H
0C04 29          ADD HL,HL
0C05 381F        JR C,0C26H
0C07 EB          EX DE,HL
0C08 29          ADD HL,HL
0C09 EB          EX DE,HL
0C0A 3004        JR NC,0C10H
0C0C 09          ADD HL,BC
0C0D DA260C      JP C,0C26H
0C10 3D          DEC A
0C11 20F1        JR NZ,0C04H
0C13 C1          POP BC
0C14 D1          POP DE
0C15 7C          LD A,H
0C16 B7          OR A
0C17 FA1F0C      JP N,0C1FH
0C1A D1          POP DE
0C1B 78          LD A,B
0C1C C34D0C      JP 0C4DH
0C1F EE80        XOR 80H
0C21 B5          OR L
0C22 2813        JR Z,0C37H
0C24 EB          EX DE,HL
0C25 01C1E1      LD BC,E1C1H
0C28 CDCF0A      CALL 0ACFH
0C2B E1          POP HL
0C2C CDA409      CALL 09A4H
0C2F CDCF0A      CALL 0ACFH
0C32 C1          POP BC
0C33 D1          POP DE
0C34 C34708      JP 0847H
0C37 78          LD A,B
0C38 B7          OR A
0C39 C1          POP BC
0C3A FA9A0A      JP N,0A9AH
0C3D D5          PUSH DE
0C3E CDCF0A      CALL 0ACFH
0C41 D1          POP DE
0C42 C38209      JP 0982H
0C45 7C          LD A,H
0C46 AA          XOR D
0C47 47          LD B,A
```

```
0C48 CD4C0C     CALL 0C4CH
0C4B EB         EX DE,HL
0C4C 7C         LD A,H
0C4D B7         OR A
0C4E F29A0A     JP P,0A9AH
0C51 AF         XOR A
0C52 4F         LD C,A
0C53 95         SUB L
0C54 6F         LD L,A
0C55 79         LD A,C
0C56 9C         SBC A,H
0C57 67         LD H,A
0C58 C39A0A     JP 0A9AH
0C5B 2A2141     LD HL,(4121H)
0C5E CD510C     CALL 0C51H
0C61 7C         LD A,H
0C62 EE80       XOR 80H
0C64 B5         OR L
0C65 C0         RET NZ
0C66 EB         EX DE,HL
0C67 CDEF0A     CALL 0AEFH
0C6A AF         XOR A
0C6B 0698       LD B,98H
0C6D C36909     JP 0969H
0C70 212D41     LD HL,412DH
0C73 7E         LD A,(HL)
0C74 EE80       XOR 80H
0C76 77         LD (HL),A
0C77 212E41     LD HL,412EH
0C7A 7E         LD A,(HL)
0C7B B7         OR A
0C7C C8         RET Z
0C7D 47         LD B,A
0C7E 2B         DEC HL
0C7F 4E         LD C,(HL)
0C80 112441     LD DE,4124H
0C83 1A         LD A,(DE)
0C84 B7         OR A
0C85 CAF409     JP Z,09F4H
0C88 90         SUB B
0C89 3016       JR NC,0CA1H
0C8B 2F         CPL
0C8C 3C         INC A
0C8D F5         PUSH AF
0C8E 0E08       LD C,08H
0C90 23         INC HL
0C91 E5         PUSH HL
0C92 1A         LD A,(DE)
0C93 46         LD B,(HL)
0C94 77         LD (HL),A
0C95 78         LD A,B
0C96 12         LD (DE),A
0C97 1B         DEC DE
0C98 2B         DEC HL
0C99 0D         DEC C
0C9A 20F6       JR NZ,0C92H
0C9C E1         POP HL
0C9D 46         LD B,(HL)
0C9E 2B         DEC HL
0C9F 4E         LD C,(HL)
```

0C48-0C9F ROM 33

```
0CA0 F1         POP AF
0CA1 FE39       CP 39H
0CA3 D0         RET NC
0CA4 F5         PUSH AF
0CA5 CDDF09     CALL 09DFH
0CA8 23         INC HL
0CA9 3600       LD (HL),00H
0CAB 47         LD B,A
0CAC F1         POP AF
0CAD 212D41     LD HL,412DH
0CB0 CD690D     CALL 0D69H
0CB3 3A2641     LD A,(4126H)
0CB6 321C41     LD (411CH),A
0CB9 78         LD A,B
0CBA B7         OR A
0CBB F2CF0C     JP P,0CCFH
0CBE CD330D     CALL 0D33H
0CC1 D20E0D     JP NC,0D0EH
0CC4 EB         EX DE,HL
0CC5 34         INC (HL)
0CC6 CAB207     JP Z,07B2H
0CC9 CD900D     CALL 0D90H
0CCC C30E0D     JP 0D0EH
0CCF CD450D     CALL 0D45H
0CD2 212541     LD HL,4125H
0CD5 DC570D     CALL C,0D57H
0CD8 AF         XOR A
0CD9 47         LD B,A
0CDA 3A2341     LD A,(4123H)
0CDD B7         OR A
0CDE 201E       JR NZ,0CFEH
0CE0 211C41     LD HL,411CH
0CE3 0E08       LD C,08H
0CE5 56         LD D,(HL)
0CE6 77         LD (HL),A
0CE7 7A         LD A,D
0CE8 23         INC HL
0CE9 0D         DEC C
0CEA 20F9       JR NZ,0CE5H
0CEC 78         LD A,B
0CED D608       SUB 08H
0CEF FEC0       CP C0H
0CF1 20E6       JR NZ,0CD9H
0CF3 C37807     JP 0778H
0CF6 05         DEC B
0CF7 211C41     LD HL,411CH
0CFA CD970D     CALL 0D97H
0CFD B7         OR A
0CFE F2F60C     JP P,0CF6H
0D01 78         LD A,B
0D02 B7         OR A
0D03 2809       JR Z,0D0EH
0D05 212441     LD HL,4124H
0D08 86         ADD A,(HL)
0D09 77         LD (HL),A
0D0A D27807     JP NC,0778H
0D0D C8         RET Z
0D0E 3A1C41     LD A,(411CH)
0D11 B7         OR A
0D12 FC200D     CALL N,0D20H
```

0CA0-0D12 ROM 34

```
0D15 212541     LD HL,4125H
0D18 7E         LD A,(HL)
0D19 E680       AND 80H
0D1B 2B         DEC HL
0D1C 2B         DEC HL
0D1D AE         XOR (HL)
0D1E 77         LD (HL),A
0D1F C9         RET
0D20 211D41     LD HL,411DH
0D23 0607       LD B,07H
0D25 34         INC (HL)
0D26 C0         RET NZ
0D27 23         INC HL
0D28 05         DEC B
0D29 20FA       JR NZ,0D25H
0D2B 34         INC (HL)
0D2C CAB207     JP Z,07B2H
0D2F 2B         DEC HL
0D30 3680       LD (HL),80H
0D32 C9         RET
0D33 212741     LD HL,4127H
0D36 111D41     LD DE,411DH
0D39 0E07       LD C,07H
0D3B AF         XOR A
0D3C 1A         LD A,(DE)
0D3D 8E         ADC A,(HL)
0D3E 12         LD (DE),A
0D3F 13         INC DE
0D40 23         INC HL
0D41 0D         DEC C
0D42 20F8       JR NZ,0D3CH
0D44 C9         RET
0D45 212741     LD HL,4127H
0D48 111D41     LD DE,411DH
0D4B 0E07       LD C,07H
0D4D AF         XOR A
0D4E 1A         LD A,(DE)
0D4F 9E         SBC A,(HL)
0D50 12         LD (DE),A
0D51 13         INC DE
0D52 23         INC HL
0D53 0D         DEC C
0D54 20F8       JR NZ,0D4EH
0D56 C9         RET
0D57 7E         LD A,(HL)
0D58 2F         CPL
0D59 77         LD (HL),A
0D5A 211C41     LD HL,411CH
0D5D 0608       LD B,08H
0D5F AF         XOR A
0D60 4F         LD C,A
0D61 79         LD A,C
0D62 9E         SBC A,(HL)
0D63 77         LD (HL),A
0D64 23         INC HL
0D65 05         DEC B
0D66 20F9       JR NZ,0D61H
0D68 C9         RET
0D69 71         LD (HL),C
0D6A E5         PUSH HL
```



```
0D6B D608       SUB 08H
0D6D 380E       JR C,0D7DH
0D6F E1         POP HL
0D70 E5         PUSH HL
0D71 110008     LD DE,0800H
0D74 4E         LD C,(HL)
0D75 73         LD (HL),E
0D76 59         LD E,C
0D77 2B         DEC HL
0D78 15         DEC D
0D79 20F9       JR NZ,0D74H
0D7B 18EE       JR 0D6BH
0D7D C609       ADD A,09H
0D7F 57         LD D,A
0D80 AF         XOR A
0D81 E1         POP HL
0D82 15         DEC D
0D83 C8         RET Z
0D84 E5         PUSH HL
0D85 1E08       LD E,08H
0D87 7E         LD A,(HL)
0D88 1F         RRA
0D89 77         LD (HL),A
0D8A 2B         DEC HL
0D8B 1D         DEC E
0D8C 20F9       JR NZ,0D87H
0D8E 18F0       JR 0D80H
0D90 212341     LD HL,4123H
0D93 1601       LD D,01H
0D95 18ED       JR 0D84H
0D97 0E08       LD C,08H
0D99 7E         LD A,(HL)
0D9A 17         RLA
0D9B 77         LD (HL),A
0D9C 23         INC HL
0D9D 0D         DEC C
0D9E 20F9       JR NZ,0D99H
0DA0 C9         RET
0DA1 CD5509     CALL 0955H
0DA4 C8         RET Z
0DA5 CD0A09     CALL 090AH
0DA8 CD390E     CALL 0E39H
0DAB 71         LD (HL),C
0DAC 13         INC DE
0DAD 0607       LD B,07H
0DAF 1A         LD A,(DE)
0DB0 13         INC DE
0DB1 B7         OR A
0DB2 D5         PUSH DE
0DB3 2817       JR Z,0DCCH
0DB5 0E08       LD C,08H
0DB7 C5         PUSH BC
0DB8 1F         RRA
0DB9 47         LD B,A
0DBA DC330D     CALL C,0D33H
0DBD CD900D     CALL 0D90H
0DC0 78         LD A,B
0DC1 C1         POP BC
0DC2 0D         DEC C
0DC3 20F2       JR NZ,0DB7H
```

```
0DC5 D1          POP DE
0DC6 05          DEC B
0DC7 20E6        JR NZ,0DAFH
0DC9 C3D80C      JP 0CD8H
0DCC 212341      LD HL,4123H
0DCF CD700D      CALL 0D70H
0DD2 18F1        JR 0DC5H
```

0DC5-0DD2 ROM 37

```
0DD4 00          .  .
0DD5 00          .  .
0DD6 00          .  .
0DD7 00          .  .
0DD8 00          .  .
0DD9 00          .  .
0DDA 20
0DDB 84          .  .
```

0DC5-0DDB ROM 37

```
0DDC 11D40D      LD DE,0DD4H
0DDF 212741      LD HL,4127H
0DE2 CDD309      CALL 09D3H
0DE5 3A2E41      LD A,(412EH)
0DE8 B7          OR A
0DE9 CA9A19      JP Z,199AH
0DEC CD0709      CALL 0907H
0DEF 34          INC (HL)
0DF0 34          INC (HL)
0DF1 CD390E      CALL 0E39H
0DF4 215141      LD HL,4151H
0DF7 71          LD (HL),C
0DF8 41          LD B,C
0DF9 114A41      LD DE,414AH
0DFC 212741      LD HL,4127H
0DFF CD4B0D      CALL 0D4BH
0E02 1A          LD A,(DE)
0E03 99          SBC A,C
0E04 3F          CCF
0E05 380B        JR C,0E12H
0E07 114A41      LD DE,414AH
0E0A 212741      LD HL,4127H
0E0D CD390D      CALL 0D39H
0E10 AF          XOR A
0E11 DA1204      JP C,0412H
0E14 3A2341      LD A,(4123H)
0E17 3C          INC A
0E18 3D          DEC A
0E19 1F          RRA
0E1A FA110D      JP N,0D11H
0E1D 17          RLA
0E1E 211D41      LD HL,411DH
0E21 0E07        LD C,07H
0E23 CD990D      CALL 0D99H
0E26 214A41      LD HL,414AH
0E29 CD970D      CALL 0D97H
0E2C 78          LD A,B
0E2D B7          OR A
0E2E 20C9        JR NZ,0DF9H
```

0DC5-0E2E ROM 37

```
0E30 212441      LD HL,4124H
0E33 35          DEC (HL)
0E34 20C3        JR NZ,0DF9H
0E36 C3B207      JP 07B2H
0E39 79          LD A,C
0E3A 322D41      LD (412DH),A
0E3D 2B          DEC HL
0E3E 115041      LD DE,4150H
0E41 010007      LD BC,0700H
0E44 7E          LD A,(HL)
0E45 12          LD (DE),A
0E46 71          LD (HL),C
0E47 1B          DEC DE
0E48 2B          DEC HL
0E49 05          DEC B
0E4A 20F8        JR NZ,0E44H
0E4C C9          RET
0E4D CDFC09      CALL 09FCH
0E50 EB          EX DE,HL
0E51 2B          DEC HL
0E52 7E          LD A,(HL)
0E53 B7          OR A
0E54 C8          RET Z
0E55 C602        ADD A,02H
0E57 DAB207      JP C,07B2H
0E5A 77          LD (HL),A
0E5B E5          PUSH HL
0E5C CD770C      CALL 0C77H
0E5F E1          POP HL
0E60 34          INC (HL)
0E61 C0          RET NZ
0E62 C3B207      JP 07B2H
0E65 CD7807      CALL 0778H
0E68 CDEC0A      CALL 0AECH
0E6B F6AF        OR AFH
0E6D EB          EX DE,HL
0E6E 01FF00      LD BC,00FFH
0E71 60          LD H,B
0E72 68          LD L,B
0E73 CC9A0A      CALL Z,0A9AH
0E76 EB          EX DE,HL
0E77 7E          LD A,(HL)
0E78 FE2D        CP 2DH
0E7A F5          PUSH AF
0E7B CA830E      JP Z,0E83H
0E7E FE2B        CP 2BH
0E80 2801        JR Z,0E83H
0E82 2B          DEC HL
0E83 D7          RST 10
0E84 DA290F      JP C,0F29H
0E87 FE2E        CP 2EH
0E89 CAE40E      JP Z,0EE4H
0E8C FE45        CP 45H
0E8E 2814        JR Z,0EA4H
0E90 FE25        CP 25H
0E92 CAEE0E      JP Z,0EEEH
0E95 FE23        CP 23H
0E97 CAF50E      JP Z,0EF5H
0E9A FE21        CP 21H
0E9C CAF60E      JP Z,0EF6H
```

0E30-0E9C ROM 38

```
0E9F FE44       CP 44H
0EA1 2024       JR NZ,0EC7H
0EA3 B7         OR A
0EA4 CDFB0E     CALL 0EFBH
0EA7 E5         PUSH HL
0EA8 21BD0E     LD HL,0EBDH
0EAB E3         EX (SP),HL
0EAC D7         RST 10
0EAD 15         DEC D
0EAE FECE       CP CEH
0EB0 C8         RET Z
0EB1 FE2D       CP 2DH
0EB3 C8         RET Z
0EB4 14         INC D
0EB5 FECD       CP CDH
0EB7 C8         RET Z
0EB8 FE2B       CP 2BH
0EBA C8         RET Z
0EBB 2B         DEC HL
0EBC F1         POP AF
0EBD D7         RST 10
0EBE DA940F     JP C,0F94H
0EC1 14         INC D
0EC2 2003       JR NZ,0EC7H
0EC4 AF         XOR A
0EC5 93         SUB E
0EC6 5F         LD E,A
0EC7 E5         PUSH HL
0EC8 7B         LD A,E
0EC9 90         SUB B
0ECA F40A0F     CALL P,0F0AH
0ECD FC180F     CALL N,0F18H
0ED0 20F8       JR NZ,0ECAH
0ED2 E1         POP HL
0ED3 F1         POP AF
0ED4 E5         PUSH HL
0ED5 CC7B09     CALL Z,097BH
0ED8 E1         POP HL
0ED9 E7         RST 20
0EDA E8         RET PE
0EDB E5         PUSH HL
0EDC 219008     LD HL,0890H
0EDF E5         PUSH HL
0EE0 CDA30A     CALL 0AA3H
0EE3 C9         RET
0EE4 E7         RST 20
0EE5 0C         INC C
0EE6 20DF       JR NZ,0EC7H
0EE8 DCFB0E     CALL C,0EFBH
0EEB C3830E     JP 0E83H
0EEE E7         RST 20
0EEF F29719     JP P,1997H
0EF2 23         INC HL
0EF3 18D2       JR 0EC7H
0EF5 B7         OR A
0EF6 CDFB0E     CALL 0EFBH
0EF9 18F7       JR 0EF2H
0EFB E5         PUSH HL
0EFC D5         PUSH DE
0EFD C5         PUSH BC
```



```
0EFE F5         PUSH AF
0EFF CCB10A     CALL Z,0AB1H
0F02 F1         POP AF
0F03 C4DB0A     CALL NZ,0ADBH
0F06 C1         POP BC
0F07 D1         POP DE
0F08 E1         POP HL
0F09 C9         RET
0F0A C8         RET Z
0F0B F5         PUSH AF
0F0C E7         RST 20
0F0D F5         PUSH AF
0F0E E43E09     CALL PO,093EH
0F11 F1         POP AF
0F12 EC4D0E     CALL PE,0E4DH
0F15 F1         POP AF
0F16 3D         DEC A
0F17 C9         RET
0F18 D5         PUSH DE
0F19 E5         PUSH HL
0F1A F5         PUSH AF
0F1B E7         RST 20
0F1C F5         PUSH AF
0F1D E49708     CALL PO,0897H
0F20 F1         POP AF
0F21 ECDC0D     CALL PE,0DDCH
0F24 F1         POP AF
0F25 E1         POP HL
0F26 D1         POP DE
0F27 3C         INC A
0F28 C9         RET
0F29 D5         PUSH DE
0F2A 78         LD A,B
0F2B 89         ADC A,C
0F2C 47         LD B,A
0F2D C5         PUSH BC
0F2E E5         PUSH HL
0F2F 7E         LD A,(HL)
0F30 D630       SUB 30H
0F32 F5         PUSH AF
0F33 E7         RST 20
0F34 F25D0F     JP P,0F5DH
0F37 2A2141     LD HL,(4121H)
0F3A 11CD0C     LD DE,0CCDH
0F3D DF         RST 18
0F3E 3019       JR NC,0F59H
0F40 54         LD D,H
0F41 5D         LD E,L
0F42 29         ADD HL,HL
0F43 29         ADD HL,HL
0F44 19         ADD HL,DE
0F45 29         ADD HL,HL
0F46 F1         POP AF
0F47 4F         LD C,A
0F48 09         ADD HL,BC
0F49 7C         LD A,H
0F4A B7         OR A
0F4B FA570F     JP N,0F57H
0F4E 222141     LD (4121H),HL
0F51 E1         POP HL
```

```
0F52 C1          POP BC
0F53 D1          POP DE
0F54 C3830E      JP 0E83H
0F57 79          LD A,C
0F58 F5          PUSH AF
0F59 CDCC0A      CALL 0ACCH
0F5C 37          SCF
0F5D 3018        JR NC,0F77H
0F5F 017494      LD BC,9474H
0F62 110024      LD DE,2400H
0F65 CD0C0A      CALL 0A0CH
0F68 F2740F      JP P,0F74H
0F6B CD3E09      CALL 093EH
0F6E F1          POP AF
0F6F CD890F      CALL 0F89H
0F72 18DD        JR 0F51H
0F74 CDE30A      CALL 0AE3H
0F77 CD4D0E      CALL 0E4DH
0F7A CDFC09      CALL 09FCH
0F7D F1          POP AF
0F7E CD6409      CALL 0964H
0F81 CDE30A      CALL 0AE3H
0F84 CD770C      CALL 0C77H
0F87 18C8        JR 0F51H
0F89 CDA409      CALL 09A4H
0F8C CD6409      CALL 0964H
0F8F C1          POP BC
0F90 D1          POP DE
0F91 C31607      JP 0716H
0F94 7B          LD A,E
0F95 FE0A        CP 0AH
0F97 3009        JR NC,0FA2H
0F99 07          RLCA
0F9A 07          RLCA
0F9B 83          ADD A,E
0F9C 07          RLCA
0F9D 86          ADD A,(HL)
0F9E D630        SUB 30H
0FA0 5F          LD E,A
0FA1 FA1E32      JP N,321EH
0FA4 C3BD0E      JP 0EBDH
0FA7 E5          PUSH HL
0FA8 212419      LD HL,1924H
0FAB CDA728      CALL 28A7H
0FAE E1          POP HL
0FAF CD9A0A      CALL 0A9AH
0FB2 AF          XOR A
0FB3 CD3410      CALL 1034H
0FB6 B6          OR (HL)
0FB7 CDD90F      CALL 0FD9H
0FBA C3A628      JP 28A6H
0FBD AF          XOR A
0FBE CD3410      CALL 1034H
0FC1 E608        AND 08H
0FC3 2802        JR Z,0FC7H
0FC5 362B        LD (HL),2BH
0FC7 EB          EX DE,HL
0FC8 CD9409      CALL 0994H
0FCB EB          EX DE,HL
0FCC F2D90F      JP P,0FD9H
```



```
0FCF 362D        LD (HL),2DH
0FD1 C5          PUSH BC
0FD2 E5          PUSH HL
0FD3 CD7B09      CALL 097BH
0FD6 E1          POP HL
0FD7 C1          POP BC
0FD8 B4          OR H
0FD9 23          INC HL
0FDA 3630        LD (HL),30H
0FDC 3AD840      LD A,(40D8H)
0FDF 57          LD D,A
0FE0 17          RLA
0FE1 3AAF40      LD A,(40AFH)
0FE4 DA9A10      JP C,109AH
0FE7 CA9210      JP Z,1092H
0FEA FE04        CP 04H
0FEC D23D10      JP NC,103DH
0FEF 010000      LD BC,0000H
0FF2 CD2F13      CALL 132FH
0FF5 213041      LD HL,4130H
0FF8 46          LD B,(HL)
0FF9 0E20        LD C,20H
0FFB 3AD840      LD A,(40D8H)
0FFE 5F          LD E,A
0FFF E620        AND 20H
1001 2807        JR Z,100AH
1003 78          LD A,B
1004 B9          CP C
1005 0E2A        LD C,2AH
1007 2001        JR NZ,100AH
1009 41          LD B,C
100A 71          LD (HL),C
100B D7          RST 10
100C 2814        JR Z,1022H
100E FE45        CP 45H
1010 2810        JR Z,1022H
1012 FE44        CP 44H
1014 280C        JR Z,1022H
1016 FE30        CP 30H
1018 28F0        JR Z,100AH
101A FE2C        CP 2CH
101C 28EC        JR Z,100AH
101E FE2E        CP 2EH
1020 2003        JR NZ,1025H
1022 2B          DEC HL
1023 3630        LD (HL),30H
1025 7B          LD A,E
1026 E610        AND 10H
1028 2803        JR Z,102DH
102A 2B          DEC HL
102B 3624        LD (HL),24H
102D 7B          LD A,E
102E E604        AND 04H
1030 C0          RET NZ
1031 2B          DEC HL
1032 70          LD (HL),B
1033 C9          RET
1034 32D840      LD (40D8H),A
1037 213041      LD HL,4130H
103A 3620        LD (HL),20H
```

```
103C C9          RET
103D FE05        CP 05H
103F E5          PUSH HL
1040 DE00        SBC A,00H
1042 17          RLA
1043 57          LD D,A
1044 14          INC D
1045 CD0112      CALL 1201H
1048 010003      LD BC,0300H
104B 82          ADD A,D
104C FA5710      JP N,1057H
104F 14          INC D
1050 BA          CP D
1051 3004        JR NC,1057H
1053 3C          INC A
1054 47          LD B,A
1055 3E02        LD A,02H
1057 D602        SUB 02H
1059 E1          POP HL
105A F5          PUSH AF
105B CD9112      CALL 1291H
105E 3630        LD (HL),30H
1060 CCC909      CALL Z,09C9H
1063 CDA412      CALL 12A4H
1066 2B          DEC HL
1067 7E          LD A,(HL)
1068 FE30        CP 30H
106A 28FA        JR Z,1066H
106C FE2E        CP 2EH
106E C4C909      CALL NZ,09C9H
1071 F1          POP AF
1072 281F        JR Z,1093H
1074 F5          PUSH AF
1075 E7          RST 20
1076 3E22        LD A,22H
1078 8F          ADC A,A
1079 77          LD (HL),A
107A 23          INC HL
107B F1          POP AF
107C 362B.       LD (HL),2BH
107E F28510      JP P,1085H
1081 362D        LD (HL),2DH
1083 2F          CPL
1084 3C          INC A
1085 062F        LD B,2FH
1087 04          INC B
1088 D60A        SUB 0AH
108A 30FB        JR NC,1087H
108C C63A        ADD A,3AH
108E 23          INC HL
108F 70          LD (HL),B
1090 23          INC HL
1091 77          LD (HL),A
1092 23          INC HL
1093 3600        LD (HL),00H
1095 EB          EX DE,HL
1096 213041      LD HL,4130H
1099 C9          RET
109A 23          INC HL
109B C5          PUSH BC
```



```
109C FE04        CP 04H
109E 7A          LD A,D
109F D20911      JP NC,1109H
10A2 1F          RRA
10A3 DAA311      JP C,11A3H
10A6 010306      LD BC,0603H
10A9 CD8912      CALL 1289H
10AC D1          POP DE
10AD 7A          LD A,D
10AE D605        SUB 05H
10B0 F46912      CALL P,1269H
10B3 CD2F13      CALL 132FH
10B6 7B          LD A,E
10B7 B7          OR A
10B8 CC2F09      CALL Z,092FH
10BB 3D          DEC A
10BC F46912      CALL P,1269H
10BF E5          PUSH HL
10C0 CDF50F      CALL 0FF5H
10C3 E1          POP HL
10C4 2802        JR Z,10C8H
10C6 70          LD (HL),B
10C7 23          INC HL
10C8 3600        LD (HL),00H
10CA 212F41      LD HL,412FH
10CD 23          INC HL
10CE 3AF340      LD A,(40F3H)
10D1 95          SUB L
10D2 92          SUB D
10D3 C8          RET Z
10D4 7E          LD A,(HL)
10D5 FE20        CP 20H
10D7 28F4        JR Z,10CDH
10D9 FE2A        CP 2AH
10DB 28F0        JR Z,10CDH
10DD 2B          DEC HL
10DE E5          PUSH HL
10DF F5          PUSH AF
10E0 01DF10      LD BC,10DFH
10E3 C5          PUSH BC
10E4 D7          RST 10
10E5 FE2D        CP 2DH
10E7 C8          RET Z
10E8 FE2B        CP 2BH
10EA C8          RET Z
10EB FE24        CP 24H
10ED C8          RET Z
10EE C1          POP BC
10EF FE30        CP 30H
10F1 200F        JR NZ,1102H
10F3 23          INC HL
10F4 D7          RST 10
10F5 300B        JR NC,1102H
10F7 2B          DEC HL
10F8 012B77      LD BC,772BH
10FB F1          POP AF
10FC 28FB        JR Z,10F9H
10FE C1          POP BC
10FF C3CE10      JP 10CEH
1102 F1          POP AF
```

```
1103 28FD        JR Z,1102H
1105 E1          POP HL
1106 3625        LD (HL),25H
1108 C9          RET
1109 E5          PUSH HL
110A 1F          RRA
110B DAAA11      JP C,11AAH
110E 2814        JR Z,1124H
1110 118413      LD DE,1384H
1113 CD490A      CALL 0A49H
1116 1610        LD D,10H
1118 FA3211      JP N,1132H
111B E1          POP HL
111C C1          POP BC
111D CDBD0F      CALL 0FBDH
1120 2B          DEC HL
1121 3625        LD (HL),25H
1123 C9          RET
1124 010EB6      LD BC,B60EH
1127 11CA1B      LD DE,1BCAH
112A CD0C0A      CALL 0A0CH
112D F21B11      JP P,111BH
1130 1606        LD D,06H
1132 CD5509      CALL 0955H
1135 C40112      CALL NZ,1201H
1138 E1          POP HL
1139 C1          POP BC
113A FA5711      JP N,1157H
113D C5          PUSH BC
113E 5F          LD E,A
113F 78          LD A,B
1140 92          SUB D
1141 93          SUB E
1142 F46912      CALL P,1269H
1145 CD7D12      CALL 127DH
1148 CDA412      CALL 12A4H
114B B3          OR E
114C C47712      CALL NZ,1277H
114F B3          OR E
1150 C49112      CALL NZ,1291H
1153 D1          POP DE
1154 C3B610      JP 1086H
1157 5F          LD E,A
1158 79          LD A,C
1159 B7          OR A
115A C4160F      CALL NZ,0F16H
115D 83          ADD A,E
115E FA6211      JP N,1162H
1161 AF          XOR A
1162 C5          PUSH BC
1163 F5          PUSH AF
1164 FC180F      CALL N,0F18H
1167 FA6411      JP N,1164H
116A C1          POP BC
116B 7B          LD A,E
116C 90          SUB B
116D C1          POP BC
116E 5F          LD E,A
116F 82          ADD A,D
1170 78          LD A,B
```

1103-1170 ROM 45

```
1171 FA7F11      JP N,117FH
1174 92          SUB D
1175 93          SUB E
1176 F46912      CALL P,1269H
1179 C5          PUSH BC
117A CD7D12      CALL 127DH
117D 1811        JR 1190H
117F CD6912      CALL 1269H
1182 79          LD A,C
1183 CD9412      CALL 1294H
1186 4F          LD C,A
1187 AF          XOR A
1188 92          SUB D
1189 93          SUB E
118A CD6912      CALL 1269H
118D C5          PUSH BC
118E 47          LD B,A
118F 4F          LD C,A
1190 CDA412      CALL 12A4H
1193 C1          POP BC
1194 B1          OR C
1195 2003        JR NZ,119AH
1197 2AF340      LD HL,(40F3H)
119A 83          ADD A,E
119B 3D          DEC A
119C F46912      CALL P,1269H
119F 50          LD D,B
11A0 C3BF10      JP 10BFH
11A3 E5          PUSH HL
11A4 D5          PUSH DE
11A5 CDCC0A      CALL 0ACCH
11A8 D1          POP DE
11A9 AF          XOR A
11AA CAB011      JP Z,11B0H
11AD 1E10        LD E,10H
11AF 011E06      LD BC,061EH
11B2 CD5509      CALL 0955H
11B5 37          SCF
11B6 C40112      CALL NZ,1201H
11B9 E1          POP HL
11BA C1          POP BC
11BB F5          PUSH AF
11BC 79          LD A,C
11BD B7          OR A
11BE F5          PUSH AF
11BF C4160F      CALL NZ,0F16H
11C2 80          ADD A,B
11C3 4F          LD C,A
11C4 7A          LD A,D
11C5 E604        AND 04H
11C7 FE01        CP 01H
11C9 9F          SBC A,A
11CA 57          LD D,A
11CB 81          ADD A,C
11CC 4F          LD C,A
11CD 93          SUB E
11CE F5          PUSH AF
11CF C5          PUSH BC
11D0 FC180F      CALL N,0F18H
11D3 FAD011      JP N,11D0H
```

1171-11D3 ROM 46

```
11D6 C1          POP BC
11D7 F1          POP AF
11D8 C5          PUSH BC
11D9 F5          PUSH AF
11DA FADE11      JP N,11DEH
11DD AF          XOR A
11DE 2F          CPL
11DF 3C          INC A
11E0 80          ADD A,B
11E1 3C          INC A
11E2 82          ADD A,D
11E3 47          LD B,A
11E4 0E00        LD C,00H
11E6 CDA412      CALL 12A4H
11E9 F1          POP AF
11EA F47112      CALL P,1271H
11ED C1          POP BC
11EE F1          POP AF
11EF CC2F09      CALL Z,092FH
11F2 F1          POP AF
11F3 3803        JR C,11F8H
11F5 83          ADD A,E
11F6 90          SUB B
11F7 92          SUB D
11F8 C5          PUSH BC
11F9 CD7410      CALL 1074H
11FC EB          EX DE,HL
11FD D1          POP DE
11FE C3BF10      JP 10BFH
1201 D5          PUSH DE
1202 AF          XOR A
1203 F5          PUSH AF
1204 E7          RST 20
1205 E22212      JP PO,1222H
1208 3A2441      LD A,(4124H)
120B FE91        CP 91H
120D D22212      JP NC,1222H
1210 116413      LD DE,1364H
1213 212741      LD HL,4127H
1216 CDD309      CALL 09D3H
1219 CDA10D      CALL 0DA1H
121C F1          POP AF
121D D60A        SUB 0AH
121F F5          PUSH AF
1220 18E6        JR 1208H
1222 CD4F12      CALL 124FH
1225 E7          RST 20
1226 300B        JR NC,1233H
1228 014391      LD BC,9143H
122B 11F94F      LD DE,4FF9H
122E CD0C0A      CALL 0A0CH
1231 1806        JR 1239H
1233 116C13      LD DE,136CH
1236 CD490A      CALL 0A49H
1239 F24B12      JP P,124BH
123C F1          POP AF
123D CD0B0F      CALL 0F0BH
1240 F5          PUSH AF
1241 18E2        JR 1225H
1243 F1          POP AF
```



```
1244 CD180F      CALL 0F18H
1247 F5          PUSH AF
1248 CD4F12      CALL 124FH
124B F1          POP AF
124C B7          OR A
124D D1          POP DE
124E C9          RET
124F E7          RST 20
1250 EA5E12      JP PE,125EH
1253 017494      LD BC,9474H
1256 11F823      LD DE,23F8H
1259 CD0C0A      CALL 0A0CH
125C 1806        JR 1264H
125E 117413      LD DE,1374H
1261 CD490A      CALL 0A49H
1264 E1          POP HL
1265 F24312      JP P,1243H
1268 E9          JP (HL)
1269 B7          OR A
126A C8          RET Z
126B 3D          DEC A
126C 3630        LD (HL),30H
126E 23          INC HL
126F 18F9        JR 126AH
1271 2004        JR NZ,1277H
1273 C8          RET Z
1274 CD9112      CALL 1291H
1277 3630        LD (HL),30H
1279 23          INC HL
127A 3D          DEC A
127B 18F6        JR 1273H
127D 7B          LD A,E
127E 82          ADD A,D
127F 3C          INC A
1280 47          LD B,A
1281 3C          INC A
1282 D603        SUB 03H
1284 30FC        JR NC,1282H
1286 C605        ADD A,05H
1288 4F          LD C,A
1289 3AD840      LD A,(40D8H)
128C E640        AND 40H
128E C0          RET NZ
128F 4F          LD C,A
1290 C9          RET
1291 05          DEC B
1292 2008        JR NZ,129CH
1294 362E        LD (HL),2EH
1296 22F340      LD (40F3H),HL
1299 23          INC HL
129A 48          LD C,B
129B C9          RET
129C 0D          DEC C
129D C0          RET NZ
129E 362C        LD (HL),2CH
12A0 23          INC HL
12A1 0E03        LD C,03H
12A3 C9          RET
12A4 D5          PUSH DE
12A5 E7          RST 20
```

```
12A6 E2EA12      JP PO,12EAH
12A9 C5          PUSH BC
12AA E5          PUSH HL
12AB CDFC09      CALL 09FCH
12AE 217C13      LD HL,137CH
12B1 CDF709      CALL 09F7H
12B4 CD770C      CALL 0C77H
12B7 AF          XOR A
12B8 CD7B0B      CALL 0B7BH
12BB E1          POP HL
12BC C1          POP BC
12BD 118C13      LD DE,138CH
12C0 3E0A        LD A,0AH
12C2 CD9112      CALL 1291H
12C5 C5          PUSH BC
12C6 F5          PUSH AF
12C7 E5          PUSH HL
12C8 D5          PUSH DE
12C9 062F        LD B,2FH
12CB 04          INC B
12CC E1          POP HL
12CD E5          PUSH HL
12CE CD480D      CALL 0D48H
12D1 30F8        JR NC,12CBH
12D3 E1          POP HL
12D4 CD360D      CALL 0D36H
12D7 EB          EX DE,HL
12D8 E1          POP HL
12D9 70          LD (HL),B
12DA 23          INC HL
12DB F1          POP AF
12DC C1          POP BC
12DD 3D          DEC A
12DE 20E2        JR NZ,12C2H
12E0 C5          PUSH BC
12E1 E5          PUSH HL
12E2 211D41      LD HL,411DH
12E5 CDB109      CALL 09B1H
12E8 180C        JR 12F6H
12EA C5          PUSH BC
12EB E5          PUSH HL
12EC CD0807      CALL 0708H
12EF 3C          INC A
12F0 CDFB0A      CALL 0AFBH
12F3 CDB409      CALL 09B4H
12F6 E1          POP HL
12F7 C1          POP BC
12F8 AF          XOR A
12F9 11D213      LD DE,13D2H
12FC 3F          CCF
12FD CD9112      CALL 1291H
1300 C5          PUSH BC
1301 F5          PUSH AF
1302 E5          PUSH HL
1303 D5          PUSH DE
1304 CDBF09      CALL 09BFH
1307 E1          POP HL
1308 062F        LD B,2FH
130A 04          INC B
130B 7B          LD A,E
```



```
130C 96          SUB (HL)
130D 5F          LD E,A
130E 23          INC HL
130F 7A          LD A,D
1310 9E          SBC A,(HL)
1311 57          LD D,A
1312 23          INC HL
1313 79          LD A,C
1314 9E          SBC A,(HL)
1315 4F          LD C,A
1316 2B          DEC HL
1317 2B          DEC HL
1318 30F0        JR NC,130AH
131A CDB707      CALL 07B7H
131D 23          INC HL
131E CDB409      CALL 09B4H
1321 EB          EX DE,HL
1322 E1          POP HL
1323 70          LD (HL),B
1324 23          INC HL
1325 F1          POP AF
1326 C1          POP BC
1327 38D3        JR C,12FCH
1329 13          INC DE
132A 13          INC DE
132B 3E04        LD A,04H
132D 1806        JR 1335H
132F D5          PUSH DE
1330 11D813      LD DE,13D8H
1333 3E05        LD A,05H
1335 CD9112      CALL 1291H
1338 C5          PUSH BC
1339 F5          PUSH AF
133A E5          PUSH HL
133B EB          EX DE,HL
133C 4E          LD C,(HL)
133D 23          INC HL
133E 46          LD B,(HL)
133F C5          PUSH BC
1340 23          INC HL
1341 E3          EX (SP),HL
1342 EB          EX DE,HL
1343 2A2141      LD HL,(4121H)
1346 062F        LD B,2FH
1348 04          INC B
1349 7D          LD A,L
134A 93          SUB E
134B 6F          LD L,A
134C 7C          LD A,H
134D 9A          SBC A,D
134E 67          LD H,A
134F 30F7        JR NC,1348H
1351 19          ADD HL,DE
1352 222141      LD (4121H),HL
1355 D1          POP DE
1356 E1          POP HL
1357 70          LD (HL),B
1358 23          INC HL
1359 F1          POP AF
135A C1          POP BC
```

```
135B 3D          DEC A
135C 20D7        JR NZ,1335H
135E CD9112      CALL 1291H
1361 77          LD (HL),A
1362 D1          POP DE
1363 C9          RET
```

135B-1363 ROM 51

```
1364  00        .   .
1365  00        .   .
1366  00        .   .
1367  00        .   .
1368  F9        .   y
1369  02        .   .
136A  15        .   .
136B  A2        .   "
136C  FD        .   }
136D  FF        .
136E  9F        .   .
136F  31        1   1
1370  A9        .   )
1371  5F        _   _
1372  63        c   c
1373  B2        .   2
1374  FE        .   ~
1375  FF        .
1376  03        .   .
1377  BF        .   ?
1378  C9        .   I
1379  1B        .   .
137A  0E        .   .
137B  B6        .   6
137C  00        .   .
137D  00        .   .
137E  00        .   .
137F  00        .   .
1380  00        .   .
1381  00        .   .
1382  00        .   .
1383  80        .   .
1384  00        .   .
1385  00        .   .
1386  04        .   .
1387  BF        .   ?
1388  C9        .   I
1389  1B        .   .
138A  0E        .   .
138B  B6        .   6
138C  00        .   .
138D  80        .   .
138E  C6        .   F
138F  A4        .   $
1390  7E        ~   ~
1391  8D        .   .
1392  03        .   .
1393  00        .   .
1394  40        @   @
1395  7A        z   z
1396  10        .   .
```

135B-1396 ROM 51

```
1397  F3        .   s
1398  5A        Z   Z
1399  00        .   .
139A  00        .   .
139B  A0        .
139C  72        r   r
139D  4E        N   N
139E  18        .   .
139F  09        .   .
13A0  00        .   .
13A1  00        .   .
13A2  10        .   .
13A3  A5        .   %
13A4  D4        .   T
13A5  E8        .   h
13A6  00        .   .
13A7  00        .   .
13A8  00        .   .
13A9  E8        .   h
13AA  76        v   v
13AB  48        H   H
13AC  17        .   .
13AD  00        .   .
13AE  00        .   .
13AF  00        .   .
13B0  E4        .   d
13B1  0B        .   .
13B2  54        T   T
13B3  02        .   .
13B4  00        .   .
13B5  00        .   .
13B6  00        .   .
13B7  CA        .   J
13B8  9A        .   .
13B9  3B        ;   ;
13BA  00        .   .
13BB  00        .   .
13BC  00        .   .
13BD  00        .   .
13BE  E1        .   a
13BF  F5        .   u
13C0  05        .   .
13C1  00        .   .
13C2  00        .   .
13C3  00        .   .
13C4  80        .   .
13C5  96        .   .
13C6  98        .   .
13C7  00        .   .
13C8  00        .   .
13C9  00        .   .
13CA  00        .   .
13CB  40        @   @
13CC  42        B   B
13CD  0F        .   .
13CE  00        .   .
13CF  00        .   .
13D0  00        .   .
13D1  00        .   .
13D2  A0        .
```

1397-13D2 ROM 52

```
13D3  86        .   .
13D4  01        .   .
13D5  10        .   .
13D6  27        '   '
13D7  00        .   .
13D8  10        .   .
13D9  27        '   '
13DA  E8        .   h
13DB  03        .   .
13DC  64        d   d
13DD  00        .   .
13DE  0A        .   .
13DF  00        .   .
13E0  01        .   .
13E1  00        .   .
```

13D3-13E1 ROM 53

```
13E2 218209     LD HL,0982H
13E5 E3         EX (SP),HL
13E6 E9         JP (HL)
13E7 CDA409     CALL 09A4H
13EA 218013     LD HL,1380H
13ED CDB109     CALL 09B1H
13F0 1803       JR 13F5H
13F2 CDB10A     CALL 0AB1H
13F5 C1         POP BC
13F6 D1         POP DE
13F7 CD5509     CALL 0955H
13FA 78         LD A,B
13FB 283C       JR Z,1439H
13FD F20414     JP P,1404H
1400 B7         OR A
1401 CA9A19     JP Z,199AH
1404 B7         OR A
1405 CA7907     JP Z,0779H
1408 D5         PUSH DE
1409 C5         PUSH BC
140A 79         LD A,C
140B F67F       OR 7FH
140D CDBF09     CALL 09BFH
1410 F22114     JP P,1421H
1413 D5         PUSH DE
1414 C5         PUSH BC
1415 CD400B     CALL 0B40H
1418 C1         POP BC
1419 D1         POP DE
141A F5         PUSH AF
141B CD0C0A     CALL 0A0CH
141E E1         POP HL
141F 7C         LD A,H
1420 1F         RRA
1421 E1         POP HL
1422 222341     LD (4123H),HL
1425 E1         POP HL
1426 222141     LD (4121H),HL
1429 DCE213     CALL C,13E2H
142C CC8209     CALL Z,0982H
142F D5         PUSH DE
1430 C5         PUSH BC
```

13D3-1430 ROM 53

```
1431 CD0908     CALL 0809H
1434 C1         POP BC
1435 D1         POP DE
1436 CD4708     CALL 0847H
1439 CDA409     CALL 09A4H
143C 013881     LD BC,8138H
143F 113BAA     LD DE,AA3BH
1442 CD4708     CALL 0847H
1445 3A2441     LD A,(4124H)
1448 FE88       CP 88H
144A D23109     JP NC,0931H
144D CD400B     CALL 0B40H
1450 C680       ADD A,80H
1452 C602       ADD A,02H
1454 DA3109     JP C,0931H
1457 F5         PUSH AF
1458 21F807     LD HL,07F8H
145B CD0B07     CALL 070BH
145E CD4108     CALL 0841H
1461 F1         POP AF
1462 C1         POP BC
1463 D1         POP DE
1464 F5         PUSH AF
1465 CD1307     CALL 0713H
1468 CD8209     CALL 0982H
146B 217914     LD HL,1479H
146E CDA914     CALL 14A9H
1471 110000     LD DE,0000H
1474 C1         POP BC
1475 4A         LD C,D
1476 C34708     JP 0847H
```

1431-1476 ROM 54

```
1479  08        .   .
147A  40        @   @
147B  2E        .   .
147C  94        .   .
147D  74        t   t
147E  70        p   p
147F  4F        O   O
1480  2E        .   .
1481  77        w   w
1482  6E        n   n
1483  02        .   .
1484  88        .   .
1485  7A        z   z
1486  E6        .   f
1487  A0        .
1488  2A        *   *
1489  7C        |   |
148A  50        P   P
148B  AA        .   *
148C  AA        .   *
148D  7E        ~   ~
148E  FF        .
148F  FF        .
1490  7F
1491  7F
1492  00        .   .
```

1431-1492 ROM 54

```
1493  00          .    .
1494  80          .    .
1495  81          .    .
1496  00          .    .
1497  00          .    .
1498  00          .    .
1499  81          .    .
```

1493-1499 ROM 55

```
149A CDA409      CALL 09A4H
149D 11320C      LD DE,0C32H
14A0 D5          PUSH DE
14A1 E5          PUSH HL
14A2 CDBF09      CALL 09BFH
14A5 CD4708      CALL 0847H
14A8 E1          POP HL
14A9 CDA409      CALL 09A4H
14AC 7E          LD A,(HL)
14AD 23          INC HL
14AE CDB109      CALL 09B1H
14B1 06F1        LD B,F1H
14B3 C1          POP BC
14B4 D1          POP DE
14B5 3D          DEC A
14B6 C8          RET Z
14B7 D5          PUSH DE
14B8 C5          PUSH BC
14B9 F5          PUSH AF
14BA E5          PUSH HL
14BB CD4708      CALL 0847H
14BE E1          POP HL
14BF CDC209      CALL 09C2H
14C2 E5          PUSH HL
14C3 CD1607      CALL 0716H
14C6 E1          POP HL
14C7 18E9        JR 14B2H
14C9 CD7F0A      CALL 0A7FH
14CC 7C          LD A,H
14CD B7          OR A
14CE FA4A1E      JP N,1E4AH
14D1 B5          OR L
14D2 CAF014      JP Z,14F0H
14D5 E5          PUSH HL
14D6 CDF014      CALL 14F0H
14D9 CDBF09      CALL 09BFH
14DC EB          EX DE,HL
14DD E3          EX (SP),HL
14DE C5          PUSH BC
14DF CDCF0A      CALL 0ACFH
14E2 C1          POP BC
14E3 D1          POP DE
14E4 CD4708      CALL 0847H
14E7 21F807      LD HL,07F8H
14EA CD0B07      CALL 070BH
14ED C3400B      JP 0B40H
14F0 219040      LD HL,4090H
14F3 E5          PUSH HL
14F4 110000      LD DE,0000H
14F7 4B          LD C,E
```

1493-14F7 ROM 55

```
14F8 2603        LD H,03H
14FA 2E08        LD L,08H
14FC EB          EX DE,HL
14FD 29          ADD HL,HL
14FE EB          EX DE,HL
14FF 79          LD A,C
1500 17          RLA
1501 4F          LD C,A
1502 E3          EX (SP),HL
1503 7E          LD A,(HL)
1504 07          RLCA
1505 77          LD (HL),A
1506 E3          EX (SP),HL
1507 D21615      JP NC,1516H
150A E5          PUSH HL
150B 2AAA40      LD HL,(40AAH)
150E 19          ADD HL,DE
150F EB          EX DE,HL
1510 3AAC40      LD A,(40ACH)
1513 89          ADC A,C
1514 4F          LD C,A
1515 E1          POP HL
1516 2D          DEC L
1517 C2FC14      JP NZ,14FCH
151A E3          EX (SP),HL
151B 23          INC HL
151C E3          EX (SP),HL
151D 25          DEC H
151E C2FA14      JP NZ,14FAH
1521 E1          POP HL
1522 2165B0      LD HL,B065H
1525 19          ADD HL,DE
1526 22AA40      LD (40AAH),HL
1529 CDEF0A      CALL 0AEFH
152C 3E05        LD A,05H
152E 89          ADC A,C
152F 32AC40      LD (40ACH),A
1532 EB          EX DE,HL
1533 0680        LD B,80H
1535 212541      LD HL,4125H
1538 70          LD (HL),B
1539 2B          DEC HL
153A 70          LD (HL),B
153B 4F          LD C,A
153C 0600        LD B,00H
153E C36507      JP 0765H
1541 218B15      LD HL,158BH
1544 CD0B07      CALL 070BH
1547 CDA409      CALL 09A4H
154A 014983      LD BC,8349H
154D 11DB0F      LD DE,0FDBH
1550 CDB409      CALL 09B4H
1553 C1          POP BC
1554 D1          POP DE
1555 CDA208      CALL 08A2H
1558 CDA409      CALL 09A4H
155B CD400B      CALL 0B40H
155E C1          POP BC
155F D1          POP DE
1560 CD1307      CALL 0713H
```

14F8-1560 ROM 56

```
1563 218F15      LD HL,158FH
1566 CD1007      CALL 0710H
1569 CD5509      CALL 0955H
156C 37          SCF
156D F27715      JP P,1577H
1570 CD0807      CALL 0708H
1573 CD5509      CALL 0955H
1576 B7          OR A
1577 F5          PUSH AF
1578 F48209      CALL P,0982H
157B 218F15      LD HL,158FH
157E CD0B07      CALL 070BH
1581 F1          POP AF
1582 D48209      CALL NC,0982H
1585 219315      LD HL,1593H
1588 C39A14      JP 149AH
```

1563-1588 ROM 57

```
158B  DB    .   [
158C  0F    .   .
158D  49    I   I
158E  81    .   .
158F  00    .   .
1590  00    .   .
1591  00    .   .
1592  7F
1593  05    .   .
1594  BA    .   :
1595  D7    .   W
1596  1E    .   .
1597  86    .   .
1598  64    d   d
1599  26    &   &
159A  99    .   .
159B  87    .   .
159C  58    X   X
159D  34    4   4
159E  23    #   #
159F  87    .   .
15A0  E0    .   `
15A1  5D    ]   ]
15A2  A5    .   %
15A3  86    .   .
15A4  DA    .   Z
15A5  0F    .   .
15A6  49    I   I
15A7  83    .   .
```

1563-15A7 ROM 57

```
15A8 CDA409      CALL 09A4H
15AB CD4715      CALL 1547H
15AE C1          POP BC
15AF E1          POP HL
15B0 CDA409      CALL 09A4H
15B3 EB          EX DE,HL
15B4 CDB409      CALL 09B4H
15B7 CD4115      CALL 1541H
15BA C3A008      JP 08A0H
```

1563-158A ROM 57

```
15BD CD5509      CALL 0955H
15C0 FCE213      CALL N,13E2H
15C3 FC8209      CALL N,0982H
15C6 3A2441      LD A,(4124H)
15C9 FE81        CP 81H
15CB 380C        JR C,15D9H
15CD 010081      LD BC,8100H
15D0 51          LD D,C
15D1 59          LD E,C
15D2 CDA208      CALL 08A2H
15D5 211007      LD HL,0710H
15D8 E5          PUSH HL
15D9 21E315      LD HL,15E3H
15DC CD9A14      CALL 149AH
15DF 218B15      LD HL,158BH
15E2 C9          RET
```

15BD-15E2 ROM 58

```
15E3  09    .   .
15E4  4A    J   J
15E5  D7    .   W
15E6  3B    ;   ;
15E7  78    x   x
15E8  02    .   .
15E9  6E    n   n
15EA  84    .   .
15EB  7B    {   {
15EC  FE    .   ~
15ED  C1    .   A
15EE  2F    /   /
15EF  7C    |   |
15F0  74    t   t
15F1  31    1   1
15F2  9A    .   .
15F3  7D    }   }
15F4  84    .   .
15F5  3D    =   =
15F6  5A    Z   Z
15F7  7D    }   }
15F8  C8    .   H
15F9  7F
15FA  91    .   .
15FB  7E    ~   ~
15FC  E4    .   d
15FD  BB    .   ;
15FE  4C    L   L
15FF  7E    ~   ~
1600  6C    l   l
1601  AA    .   *
1602  AA    .   *
1603  7F
1604  00    .   .
1605  00    .   .
1606  00    .   .
1607  81    .   .
```

15BD-1607 ROM 58

```
1608          098A
```

15BD-1608 ROM 58

160A	0B37
160C	0977
160E	27D4
1610	2AEF
1612	27F5
1614	13E7
1616	14C9
1618	0809
161A	1439
161C	1541
161E	1547
1620	15A8
1622	15BD
1624	2CAA
1626	4152
1628	4158
162A	415E
162C	4161
162E	4164
1630	4167
1632	416A
1634	416D
1636	4170
1638	0A7F
163A	0AB1
163C	0ADB
163E	0B26
1640	2A03
1642	2836
1644	2AC5
1646	2A0F
1648	2A1F
164A	2A61
164C	2A91
164E	2A9A

160A-164E ROM 59

1650	C5	. E
1651	4E	N N
1652	44	D D
1653	C6	. F
1654	4F	O O
1655	52	R R
1656	D2	. R
1657	45	E E
1658	53	S S
1659	45	E E
165A	54	T T
165B	D3	. S
165C	45	E E
165D	54	T T
165E	C3	. C
165F	4C	L L
1660	53	S S
1661	C3	. C
1662	4D	M M
1663	44	D D
1664	D2	. R
1665	41	A A

160A-1665 ROM 59

1666	4E	N N
1667	44	D D
1668	4F	O O
1669	4D	M M
166A	CE	. N
166B	45	E E
166C	58	X X
166D	54	T T
166E	C4	. D
166F	41	A A
1670	54	T T
1671	41	A A
1672	C9	. I
1673	4E	N N
1674	50	P P
1675	55	U U
1676	54	T T
1677	C4	. D
1678	49	I I
1679	4D	M M
167A	D2	. R
167B	45	E E
167C	41	A A
167D	44	D D
167E	CC	. L
167F	45	E E
1680	54	T T
1681	C7	. G
1682	4F	O O
1683	54	T T
1684	4F	O O
1685	D2	. R
1686	55	U U
1687	4E	N N
1688	C9	. I
1689	46	F F
168A	D2	. R
168B	45	E E
168C	53	S S
168D	54	T T
168E	4F	O O
168F	52	R R
1690	45	E E
1691	C7	. G
1692	4F	O O
1693	53	S S
1694	55	U U
1695	42	B B
1696	D2	. R
1697	45	E E
1698	54	T T
1699	55	U U
169A	52	R R
169B	4E	N N
169C	D2	. R
169D	45	E E
169E	4D	M M
169F	D3	. S
16A0	54	T T
16A1	4F	O O

1666-16A1 ROM 60

16A2	50	P	P
16A3	C5	.	E
16A4	4C	L	L
16A5	53	S	S
16A6	45	E	E
16A7	D4	.	T
16A8	52	R	R
16A9	4F	O	O
16AA	4E	N	N
16AB	D4	.	T
16AC	52	R	R
16AD	4F	O	O
16AE	46	F	F
16AF	46	F	F
16B0	C4	.	D
16B1	45	E	E
16B2	46	F	F
16B3	53	S	S
16B4	54	T	T
16B5	52	R	R
16B6	C4	.	D
16B7	45	E	E
16B8	46	F	F
16B9	49	I	I
16BA	4E	N	N
16BB	54	T	T
16BC	C4	.	D
16BD	45	E	E
16BE	46	F	F
16BF	53	S	S
16C0	4E	N	N
16C1	47	G	G
16C2	C4	.	D
16C3	45	E	E
16C4	46	F	F
16C5	44	D	D
16C6	42	B	B
16C7	4C	L	L
16C8	CC	.	L
16C9	49	I	I
16CA	4E	N	N
16CB	45	E	E
16CC	C5	.	E
16CD	44	D	D
16CE	49	I	I
16CF	54	T	T
16D0	C5	.	E
16D1	52	R	R
16D2	52	R	R
16D3	4F	O	O
16D4	52	R	R
16D5	D2	.	R
16D6	45	E	E
16D7	53	S	S
16D8	55	U	U
16D9	4D	M	M
16DA	45	E	E
16DB	CF	.	O
16DC	55	U	U
16DD	54	T	T

16DE	CF	.	O
16DF	4E	N	N
16E0	CF	.	O
16E1	50	P	P
16E2	45	E	E
16E3	4E	N	N
16E4	C6	.	F
16E5	49	I	I
16E6	45	E	E
16E7	4C	L	L
16E8	44	D	D
16E9	C7	.	G
16EA	45	E	E
16EB	54	T	T
16EC	D0	.	P
16ED	55	U	U
16EE	54	T	T
16EF	C3	.	C
16F0	4C	L	L
16F1	4F	O	O
16F2	53	S	S
16F3	45	E	E
16F4	CC	.	L
16F5	4F	O	O
16F6	41	A	A
16F7	44	D	D
16F8	CD	.	M
16F9	45	E	E
16FA	52	R	R
16FB	47	G	G
16FC	45	E	E
16FD	CE	.	N
16FE	41	A	A
16FF	4D	M	M
1700	45	E	E
1701	CB	.	K
1702	49	I	I
1703	4C	L	L
1704	4C	L	L
1705	CC	.	L
1706	53	S	S
1707	45	E	E
1708	54	T	T
1709	D2	.	R
170A	53	S	S
170B	45	E	E
170C	54	T	T
170D	D3	.	S
170E	41	A	A
170F	56	V	V
1710	45	E	E
1711	D3	.	S
1712	59	Y	Y
1713	53	S	S
1714	54	T	T
1715	45	E	E
1716	4D	M	M
1717	CC	.	L
1718	50	P	P
1719	52	R	R

171A	49	I	I
171B	4E	N	N
171C	54	T	T
171D	C4	.	D
171E	45	E	E
171F	46	F	F
1720	D0	.	P
1721	4F	O	O
1722	4B	K	K
1723	45	E	E
1724	D0	.	P
1725	52	R	R
1726	49	I	I
1727	4E	N	N
1728	54	T	T
1729	C3	.	C
172A	4F	O	O
172B	4E	N	N
172C	54	T	T
172D	CC	.	L
172E	49	I	I
172F	53	S	S
1730	54	T	T
1731	CC	.	L
1732	4C	L	L
1733	49	I	I
1734	53	S	S
1735	54	T	T
1736	C4	.	D
1737	45	E	E
1738	4C	L	L
1739	45	E	E
173A	54	T	T
173B	45	E	E
173C	C1	.	A
173D	55	U	U
173E	54	T	T
173F	4F	O	O
1740	C3	.	C
1741	4C	L	L
1742	45	E	E
1743	41	A	A
1744	52	R	R
1745	C3	.	C
1746	4C	L	L
1747	4F	O	O
1748	41	A	A
1749	44	D	D
174A	C3	.	C
174B	53	S	S
174C	41	A	A
174D	56	V	V
174E	45	E	E
174F	CE	.	N
1750	45	E	E
1751	57	W	W
1752	D4	.	T
1753	41	A	A
1754	42	B	B
1755	28	(	(

171A-1755 ROM 63

1756	D4	.	T
1757	4F	O	O
1758	C6	.	F
1759	4E	N	N
175A	D5	.	U
175B	53	S	S
175C	49	I	I
175D	4E	N	N
175E	47	G	G
175F	D6	.	V
1760	41	A	A
1761	52	R	R
1762	50	P	P
1763	54	T	T
1764	52	R	R
1765	D5	.	U
1766	53	S	S
1767	52	R	R
1768	C5	.	E
1769	52	R	R
176A	4C	L	L
176B	C5	.	E
176C	52	R	R
176D	52	R	R
176E	D3	.	S
176F	54	T	T
1770	52	R	R
1771	49	I	I
1772	4E	N	N
1773	47	G	G
1774	24	$	$
1775	C9	.	I
1776	4E	N	N
1777	53	S	S
1778	54	T	T
1779	52	R	R
177A	D0	.	P
177B	4F	O	O
177C	49	I	I
177D	4E	N	N
177E	54	T	T
177F	D4	.	T
1780	49	I	I
1781	4D	M	M
1782	45	E	E
1783	24	$	$
1784	CD	.	M
1785	45	E	E
1786	4D	M	M
1787	C9	.	I
1788	4E	N	N
1789	4B	K	K
178A	45	E	E
178B	59	Y	Y
178C	24	$	$
178D	D4	.	T
178E	48	H	H
178F	45	E	E
1790	4E	N	N
1791	CE	.	N

1756-1791 ROM 64

1792	4F	O	O
1793	54	T	T
1794	D3	.	S
1795	54	T	T
1796	45	E	E
1797	50	P	P
1798	AB	.	+
1799	AD	.	-
179A	AA	.	*
179B	AF	.	/
179C	DB	.	[
179D	C1	.	A
179E	4E	N	N
179F	44	D	D
17A0	CF	.	O
17A1	52	R	R
17A2	BE	.	>
17A3	BD	.	=
17A4	BC	.	<
17A5	D3	.	S
17A6	47	G	G
17A7	4E	N	N
17A8	C9	.	I
17A9	4E	N	N
17AA	54	T	T
17AB	C1	.	A
17AC	42	B	B
17AD	53	S	S
17AE	C6	.	F
17AF	52	R	R
17B0	45	E	E
17B1	C9	.	I
17B2	4E	N	N
17B3	50	P	P
17B4	D0	.	P
17B5	4F	O	O
17B6	53	S	S
17B7	D3	.	S
17B8	51	Q	Q
17B9	52	R	R
17BA	D2	.	R
17BB	4E	N	N
17BC	44	D	D
17BD	CC	.	L
17BE	4F	O	O
17BF	47	G	G
17C0	C5	.	E
17C1	58	X	X
17C2	50	P	P
17C3	C3	.	C
17C4	4F	O	O
17C5	53	S	S
17C6	D3	.	S
17C7	49	I	I
17C8	4E	N	N
17C9	D4	.	T
17CA	41	A	A
17CB	4E	N	N
17CC	C1	.	A
17CD	54	T	T

17CE	4E	N	N
17CF	D0	.	P
17D0	45	E	E
17D1	45	E	E
17D2	4B	K	K
17D3	C3	.	C
17D4	56	V	V
17D5	49	I	I
17D6	C3	.	C
17D7	56	V	V
17D8	53	S	S
17D9	C3	.	C
17DA	56	V	V
17DB	44	D	D
17DC	C5	.	E
17DD	4F	O	O
17DE	46	F	F
17DF	CC	.	L
17E0	4F	O	O
17E1	43	C	C
17E2	CC	.	L
17E3	4F	O	O
17E4	46	F	F
17E5	CD	.	M
17E6	4B	K	K
17E7	49	I	I
17E8	24	$	$
17E9	CD	.	M
17EA	4B	K	K
17EB	53	S	S
17EC	24	$	$
17ED	CD	.	M
17EE	4B	K	K
17EF	44	D	D
17F0	24	$	$
17F1	C3	.	C
17F2	49	I	I
17F3	4E	N	N
17F4	54	T	T
17F5	C3	.	C
17F6	53	S	S
17F7	4E	N	N
17F8	47	G	G
17F9	C3	.	C
17FA	44	D	D
17FB	42	B	B
17FC	4C	L	L
17FD	C6	.	F
17FE	49	I	I
17FF	58	X	X
1800	CC	.	L
1801	45	E	E
1802	4E	N	N
1803	D3	.	S
1804	54	T	T
1805	52	R	R
1806	24	$	$
1807	D6	.	V
1808	41	A	A
1809	4C	L	L

180A	C1	.	A
180B	53	S	S
180C	43	C	C
180D	C3	.	C
180E	48	H	H
180F	52	R	R
1810	24	$	$
1811	CC	.	L
1812	45	E	E
1813	46	F	F
1814	54	T	T
1815	24	$	$
1816	D2	.	R
1817	49	I	I
1818	47	G	G
1819	48	H	H
181A	54	T	T
181B	24	$	$
181C	CD	.	M
181D	49	I	I
181E	44	D	D
181F	24	$	$
1820	A7	.	'
1821	80	.	.

180A-1821 ROM 67

1822	1DAE
1824	1CA1
1826	0138
1828	0135
182A	01C9
182C	4173
182E	01D3
1830	22B6
1832	1F05
1834	219A
1836	2608
1838	21EF
183A	1F21
183C	1EC2
183E	1EA3
1840	2039
1842	1D91
1844	1EB1
1846	1EDE
1848	1F07
184A	1DA9
184C	1F07
184E	1DF7
1850	1DF8
1852	1E00
1854	1E03
1856	1E06
1858	1E09
185A	41A3
185C	2E60
185E	1FF4
1860	1FAF
1862	2AFB

180A-1862 ROM 67

1864	1F6C
1866	4179
1868	417C
186A	417F
186C	4182
186E	4185
1870	4188
1872	418B
1874	418E
1876	4191
1878	4197
187A	419A
187C	41A0
187E	02B2
1880	2067
1882	415B
1884	2CB1
1886	206F
1888	1DE4
188A	2B2E
188C	2B29
188E	2BC6
1890	2008
1892	1E7A
1894	2C1F
1896	2BF5
1898	1B49

1864-1898 ROM 68

189A	79	y	y
189B	79	y	y
189C	7C	|	|
189D	7C	|	|
189E	7F		
189F	50	P	P
18A0	46	F	F

1864-18A0 ROM 68

18A1	0ADB
18A3	0000
18A5	0A7F
18A7	0AF4
18A9	0AB1
18AB	0C77
18AD	0C70
18AF	0DA1
18B1	0DE5
18B3	0A78
18B5	0716
18B7	0713
18B9	0847
18BB	08A2
18BD	0A0C
18BF	0BD2
18C1	0BC7
18C3	0BF2
18C5	2490
18C7	0A39

1864-18C7 ROM 68

```
18C9  4E        N    N
18CA  46        F    F
18CB  53        S    S
18CC  4E        N    N
18CD  52        R    R
18CE  47        G    G
18CF  4F        O    O
18D0  44        D    D
18D1  46        F    F
18D2  43        C    C
18D3  4F        O    O
18D4  56        V    V
18D5  4F        O    O
18D6  4D        M    M
18D7  55        U    U
18D8  4C        L    L
18D9  42        B    B
18DA  53        S    S
18DB  44        D    D
18DC  44        D    D
18DD  2F        /    /
18DE  30        0    0
18DF  49        I    I
18E0  44        D    D
18E1  54        T    T
18E2  4D        M    M
18E3  4F        O    O
18E4  53        S    S
18E5  4C        L    L
18E6  53        S    S
18E7  53        S    S
18E8  54        T    T
18E9  43        C    C
18EA  4E        N    N
18EB  4E        N    N
18EC  52        R    R
18ED  52        R    R
18EE  57        W    W
18EF  55        U    U
18F0  45        E    E
18F1  4D        M    M
18F2  4F        O    O
18F3  46        F    F
18F4  44        D    D
18F5  53        S    S
18F6  4E        N    N
18F7  D6        .    V
18F8  00        .    .
18F9  6F        o    o
18FA  7C        |    |
18FB  DE        .    ^
18FC  00        .    .
18FD  67        g    g
18FE  78        x    x
18FF  DE        .    ^
1900  00        .    .
1901  47        G    G
1902  3E        >    >
1903  00        .    .
1904  C9        .    I
```

18C9-1904 ROM 69

```
1905  4A        J    J
1906  1E        .    .
1907  40        @    @
1908  E6        .    f
1909  4D        M    M
190A  DB        .    [
190B  00        .    .
190C  C9        .    I
190D  D3        .    S
190E  00        .    .
190F  C9        .    I
1910  00        .    .
1911  00        .    .
1912  00        .    .
1913  00        .    .
1914  40        @    @
1915  30        0    0
1916  00        .    .
1917  4C        L    L
1918  43        C    C
1919  FE        .    ~
191A  FF        .
191B  E9        .    i
191C  42        B    B
191D  20
191E  45        E    E
191F  72        r    r
1920  72        r    r
1921  6F        o    o
1922  72        r    r
1923  00        .    .
1924  20
1925  69        i    i
1926  6E        n    n
1927  20
1928  00        .    .
1929  52        R    R
192A  45        E    E
192B  41        A    A
192C  44        D    D
192D  59        Y    Y
192E  0D        .    .
192F  00        .    .
1930  42        B    B
1931  72        r    r
1932  65        e    e
1933  61        a    a
1934  6B        k    k
1935  00        .    .
```

1905-1935 ROM 70

```
1936 210400     LD HL,0004H
1939 39         ADD HL,SP
193A 7E         LD A,(HL)
193B 23         INC HL
193C FE81       CP 81H
193E C0         RET NZ
193F 4E         LD C,(HL)
1940 23         INC HL
```

1905-1940 ROM 70

```
1941 46          LD B,(HL)
1942 23          INC HL
1943 E5          PUSH HL
1944 69          LD L,C
1945 60          LD H,B
1946 7A          LD A,D
1947 B3          OR E
1948 EB          EX DE,HL
1949 2802        JR Z,194DH
194B EB          EX DE,HL
194C DF          RST 18
194D 010E00      LD BC,000EH
1950 E1          POP HL
1951 C8          RET Z
1952 09          ADD HL,BC
1953 18E5        JR 193AH
1955 CD6C19      CALL 196CH
1958 C5          PUSH BC
1959 E3          EX (SP),HL
195A C1          POP BC
195B DF          RST 18
195C 7E          LD A,(HL)
195D 02          LD (BC),A
195E C8          RET Z
195F 0B          DEC BC
1960 2B          DEC HL
1961 18F8        JR 195BH
1963 E5          PUSH HL
1964 2AFD40      LD HL,(40FDH)
1967 0600        LD B,00H
1969 09          ADD HL,BC
196A 09          ADD HL,BC
196B 3EE5        LD A,E5H
196D 3EC6        LD A,C6H
196F 95          SUB L
1970 6F          LD L,A
1971 3EFF        LD A,FFH
1973 9C          SBC A,H
1974 3804        JR C,197AH
1976 67          LD H,A
1977 39          ADD HL,SP
1978 E1          POP HL
1979 D8          RET C
197A 1E0C        LD E,0CH
197C 1824        JR 19A2H
197E 2AA240      LD HL,(40A2H)
1981 7C          LD A,H
1982 A5          AND L
1983 3C          INC A
1984 2808        JR Z,198EH
1986 3AF240      LD A,(40F2H)
1989 B7          OR A
198A 1E22        LD E,22H
198C 2014        JR NZ,19A2H
198E C3C11D      JP 1DC1H
1991 2ADA40      LD HL,(40DAH)
1994 22A240      LD (40A2H),HL
1997 1E02        LD E,02H
1999 011E14      LD BC,141EH
199C 011E00      LD BC,001EH
```

1941-199C ROM 71

```
199F 011E24      LD BC,241EH
19A2 2AA240      LD HL,(40A2H)
19A5 22EA40      LD (40EAH),HL
19A8 22EC40      LD (40ECH),HL
19AB 01B419      LD BC,19B4H
19AE 2AE840      LD HL,(40E8H)
19B1 C39A1B      JP 1B9AH
19B4 C1          POP BC
19B5 7B          LD A,E
19B6 4B          LD C,E
19B7 329A40      LD (409AH),A
19BA 2AE640      LD HL,(40E6H)
19BD 22EE40      LD (40EEH),HL
19C0 EB          EX DE,HL
19C1 2AEA40      LD HL,(40EAH)
19C4 7C          LD A,H
19C5 A5          AND L
19C6 3C          INC A
19C7 2807        JR Z,19D0H
19C9 22F540      LD (40F5H),HL
19CC EB          EX DE,HL
19CD 22F740      LD (40F7H),HL
19D0 2AF040      LD HL,(40F0H)
19D3 7C          LD A,H
19D4 B5          OR L
19D5 EB          EX DE,HL
19D6 21F240      LD HL,40F2H
19D9 2808        JR Z,19E3H
19DB A6          AND (HL)
19DC 2005        JR NZ,19E3H
19DE 35          DEC (HL)
19DF EB          EX DE,HL
19E0 C3361D      JP 1D36H
19E3 AF          XOR A
19E4 77          LD (HL),A
19E5 59          LD E,C
19E6 CDF920      CALL 20F9H
19E9 21C918      LD HL,18C9H
19EC CDA641      CALL 41A6H
19EF 57          LD D,A
19F0 3E3F        LD A,3FH
19F2 CD2A03      CALL 032AH
19F5 19          ADD HL,DE
19F6 7E          LD A,(HL)
19F7 CD2A03      CALL 032AH
19FA D7          RST 10
19FB CD2A03      CALL 032AH
19FE 211D19      LD HL,191DH
1A01 E5          PUSH HL
1A02 2AEA40      LD HL,(40EAH)
1A05 E3          EX (SP),HL
1A06 CDA728      CALL 28A7H
1A09 E1          POP HL
1A0A 11FEFF      LD DE,FFFEH
1A0D DF          RST 18
1A0E CA7406      JP Z,0674H
1A11 7C          LD A,H
1A12 A5          AND L
1A13 3C          INC A
1A14 C4A70F      CALL NZ,0FA7H
```

199F-1A14 ROM 72

```
1A17 3EC1        LD A,C1H
1A19 CD8B03      CALL 038BH
1A1C CDAC41      CALL 41ACH
1A1F CDF801      CALL 01F8H
1A22 CDF920      CALL 20F9H
1A25 212919      LD HL,1929H
1A28 CDA728      CALL 28A7H
1A2B 3A9A40      LD A,(409AH)
1A2E D602        SUB 02H
1A30 CC532E      CALL Z,2E53H
1A33 21FFFF      LD HL,FFFFH
1A36 22A240      LD (40A2H),HL
1A39 3AE140      LD A,(40E1H)
1A3C B7          OR A
1A3D 2837        JR Z,1A76H
1A3F 2AE240      LD HL,(40E2H)
1A42 E5          PUSH HL
1A43 CDAF0F      CALL 0FAFH
1A46 D1          POP DE
1A47 D5          PUSH DE
1A48 CD2C1B      CALL 1B2CH
1A4B 3E2A        LD A,2AH
1A4D 3802        JR C,1A51H
1A4F 3E20        LD A,20H
1A51 CD2A03      CALL 032AH
1A54 CD6103      CALL 0361H
1A57 D1          POP DE
1A58 3006        JR NC,1A60H
1A5A AF          XOR A
1A5B 32E140      LD (40E1H),A
1A5E 18B9        JR 1A19H
1A60 2AE440      LD HL,(40E4H)
1A63 19          ADD HL,DE
1A64 38F4        JR C,1A5AH
1A66 D5          PUSH DE
1A67 11F9FF      LD DE,FFF9H
1A6A DF          RST 18
1A6B D1          POP DE
1A6C 30EC        JR NC,1A5AH
1A6E 22E240      LD (40E2H),HL
1A71 F6FF        OR FFH
1A73 C3EB2F      JP 2FEBH
1A76 3E3E        LD A,3EH
1A78 CD2A03      CALL 032AH
1A7B CD6103      CALL 0361H
1A7E DA331A      JP C,1A33H
1A81 D7          RST 10
1A82 3C          INC A
1A83 3D          DEC A
1A84 CA331A      JP Z,1A33H
1A87 F5          PUSH AF
1A88 CD5A1E      CALL 1E5AH
1A8B 2B          DEC HL
1A8C 7E          LD A,(HL)
1A8D FE20        CP 20H
1A8F 28FA        JR Z,1A8BH
1A91 23          INC HL
1A92 7E          LD A,(HL)
1A93 FE20        CP 20H
1A95 CCC909      CALL Z,09C9H
```



```
1A98 D5          PUSH DE
1A99 CDC01B      CALL 1BC0H
1A9C D1          POP DE
1A9D F1          POP AF
1A9E 22E640      LD (40E6H),HL
1AA1 CDB241      CALL 41B2H
1AA4 D25A1D      JP NC,1D5AH
1AA7 D5          PUSH DE
1AA8 C5          PUSH BC
1AA9 AF          XOR A
1AAA 32DD40      LD (40DDH),A
1AAD D7          RST 10
1AAE B7          OR A
1AAF F5          PUSH AF
1AB0 EB          EX DE,HL
1AB1 22EC40      LD (40ECH),HL
1AB4 EB          EX DE,HL
1AB5 CD2C1B      CALL 1B2CH
1AB8 C5          PUSH BC
1AB9 DCE42B      CALL C,2BE4H
1ABC D1          POP DE
1ABD F1          POP AF
1ABE D5          PUSH DE
1ABF 2827        JR Z,1AE8H
1AC1 D1          POP DE
1AC2 2AF940      LD HL,(40F9H)
1AC5 E3          EX (SP),HL
1AC6 C1          POP BC
1AC7 09          ADD HL,BC
1AC8 E5          PUSH HL
1AC9 CD5519      CALL 1955H
1ACC E1          POP HL
1ACD 22F940      LD (40F9H),HL
1AD0 EB          EX DE,HL
1AD1 74          LD (HL),H
1AD2 D1          POP DE
1AD3 E5          PUSH HL
1AD4 23          INC HL
1AD5 23          INC HL
1AD6 73          LD (HL),E
1AD7 23          INC HL
1AD8 72          LD (HL),D
1AD9 23          INC HL
1ADA EB          EX DE,HL
1ADB 2AA740      LD HL,(40A7H)
1ADE EB          EX DE,HL
1ADF 1B          DEC DE
1AE0 1B          DEC DE
1AE1 1A          LD A,(DE)
1AE2 77          LD (HL),A
1AE3 23          INC HL
1AE4 13          INC DE
1AE5 B7          OR A
1AE6 20F9        JR NZ,1AE1H
1AE8 D1          POP DE
1AE9 CDFC1A      CALL 1AFCH
1AEC CDB541      CALL 41B5H
1AEF CD5D1B      CALL 1B5DH
1AF2 CDB841      CALL 41B8H
1AF5 C3331A      JP 1A33H
```

```
1AF8 2AA440     LD HL,(40A4H)
1AFB EB         EX DE,HL
1AFC 62         LD H,D
1AFD 6B         LD L,E
1AFE 7E         LD A,(HL)
1AFF 23         INC HL
1B00 B6         OR (HL)
1B01 C8         RET Z
1B02 23         INC HL
1B03 23         INC HL
1B04 23         INC HL
1B05 AF         XOR A
1B06 BE         CP (HL)
1B07 23         INC HL
1B08 20FC       JR NZ,1B06H
1B0A EB         EX DE,HL
1B0B 73         LD (HL),E
1B0C 23         INC HL
1B0D 72         LD (HL),D
1B0E 18EC       JR 1AFCH
1B10 110000     LD DE,0000H
1B13 D5         PUSH DE
1B14 2809       JR Z,1B1FH
1B16 D1         POP DE
1B17 CD4F1E     CALL 1E4FH
1B1A D5         PUSH DE
1B1B 280B       JR Z,1B28H
1B1D CF         RST 8
1B1E CE11       ADC A,11H
1B20 FAFFC4     JP N,C4FFH
1B23 4F         LD C,A
1B24 1EC2       LD E,C2H
1B26 97         SUB A
1B27 19         ADD HL,DE
1B28 EB         EX DE,HL
1B29 D1         POP DE
1B2A E3         EX (SP),HL
1B2B E5         PUSH HL
1B2C 2AA440     LD HL,(40A4H)
1B2F 44         LD B,H
1B30 4D         LD C,L
1B31 7E         LD A,(HL)
1B32 23         INC HL
1B33 B6         OR (HL)
1B34 2B         DEC HL
1B35 C8         RET Z
1B36 23         INC HL
1B37 23         INC HL
1B38 7E         LD A,(HL)
1B39 23         INC HL
1B3A 66         LD H,(HL)
1B3B 6F         LD L,A
1B3C DF         RST 18
1B3D 60         LD H,B
1B3E 69         LD L,C
1B3F 7E         LD A,(HL)
1B40 23         INC HL
1B41 66         LD H,(HL)
1B42 6F         LD L,A
1B43 3F         CCF
```



```
1B44 C8         RET Z
1B45 3F         CCF
1B46 D0         RET NC
1B47 18E6       JR 1B2FH
1B49 C0         RET NZ
1B4A CDC901     CALL 01C9H
1B4D 2AA440     LD HL,(40A4H)
1B50 CDF81D     CALL 1DF8H
1B53 32E140     LD (40E1H),A
1B56 77         LD (HL),A
1B57 23         INC HL
1B58 77         LD (HL),A
1B59 23         INC HL
1B5A 22F940     LD (40F9H),HL
1B5D 2AA440     LD HL,(40A4H)
1B60 2B         DEC HL
1B61 22DF40     LD (40DFH),HL
1B64 061A       LD B,1AH
1B66 210141     LD HL,4101H
1B69 3604       LD (HL),04H
1B6B 23         INC HL
1B6C 10FB       DJNZ 1B69H
1B6E AF         XOR A
1B6F 32F240     LD (40F2H),A
1B72 6F         LD L,A
1B73 67         LD H,A
1B74 22F040     LD (40F0H),HL
1B77 22F740     LD (40F7H),HL
1B7A 2AB140     LD HL,(40B1H)
1B7D 22D640     LD (40D6H),HL
1B80 CD911D     CALL 1D91H
1B83 2AF940     LD HL,(40F9H)
1B86 22FB40     LD (40FBH),HL
1B89 22FD40     LD (40FDH),HL
1B8C CDBB41     CALL 41BBH
1B8F C1         POP BC
1B90 2AA040     LD HL,(40A0H)
1B93 2B         DEC HL
1B94 2B         DEC HL
1B95 22E840     LD (40E8H),HL
1B98 23         INC HL
1B99 23         INC HL
1B9A F9         LD SP,HL
1B9B 21B540     LD HL,40B5H
1B9E 22B340     LD (40B3H),HL
1BA1 CD8B03     CALL 038BH
1BA4 CD6921     CALL 2169H
1BA7 AF         XOR A
1BA8 67         LD H,A
1BA9 6F         LD L,A
1BAA 32DC40     LD (40DCH),A
1BAD E5         PUSH HL
1BAE C5         PUSH BC
1BAF 2ADF40     LD HL,(40DFH)
1BB2 C9         RET
1BB3 3E3F       LD A,3FH
1BB5 CD2A03     CALL 032AH
1BB8 3E20       LD A,20H
1BBA CD2A03     CALL 032AH
1BBD C36103     JP 0361H
```

```
1BC0 AF          XOR A
1BC1 32B040      LD (40B0H),A
1BC4 4F          LD C,A
1BC5 EB          EX DE,HL
1BC6 2AA740      LD HL,(40A7H)
1BC9 2B          DEC HL
1BCA 2B          DEC HL
1BCB EB          EX DE,HL
1BCC 7E          LD A,(HL)
1BCD FE20        CP 20H
1BCF CA5B1C      JP Z,1C5BH
1BD2 47          LD B,A
1BD3 FE22        CP 22H
1BD5 CA771C      JP Z,1C77H
1BD8 B7          OR A
1BD9 CA7D1C      JP Z,1C7DH
1BDC 3AB040      LD A,(40B0H)
1BDF B7          OR A
1BE0 7E          LD A,(HL)
1BE1 C25B1C      JP NZ,1C5BH
1BE4 FE3F        CP 3FH
1BE6 3EB2        LD A,B2H
1BE8 CA5B1C      JP Z,1C5BH
1BEB 7E          LD A,(HL)
1BEC FE30        CP 30H
1BEE 3805        JR C,1BF5H
1BF0 FE3C        CP 3CH
1BF2 DA5B1C      JP C,1C5BH
1BF5 D5          PUSH DE
1BF6 114F16      LD DE,164FH
1BF9 C5          PUSH BC
1BFA 013D1C      LD BC,1C3DH
1BFD C5          PUSH BC
1BFE 067F        LD B,7FH
1C00 7E          LD A,(HL)
1C01 FE61        CP 61H
1C03 3807        JR C,1C0CH
1C05 FE7B        CP 7BH
1C07 3003        JR NC,1C0CH
1C09 E65F        AND 5FH
1C0B 77          LD (HL),A
1C0C 4E          LD C,(HL)
1C0D EB          EX DE,HL
1C0E 23          INC HL
1C0F B6          OR (HL)
1C10 F20E1C      JP P,1C0EH
1C13 04          INC B
1C14 7E          LD A,(HL)
1C15 E67F        AND 7FH
1C17 C8          RET Z
1C18 B9          CP C
1C19 20F3        JR NZ,1C0EH
1C1B EB          EX DE,HL
1C1C E5          PUSH HL
1C1D 13          INC DE
1C1E 1A          LD A,(DE)
1C1F B7          OR A
1C20 FA391C      JP M,1C39H
1C23 4F          LD C,A
1C24 78          LD A,B
```

1BC0-1C24 ROM 77

```
1C25 FE8D        CP 8DH
1C27 2002        JR NZ,1C2BH
1C29 D7          RST 10
1C2A 2B          DEC HL
1C2B 23          INC HL
1C2C 7E          LD A,(HL)
1C2D FE61        CP 61H
1C2F 3802        JR C,1C33H
1C31 E65F        AND 5FH
1C33 B9          CP C
1C34 28E7        JR Z,1C1DH
1C36 E1          POP HL
1C37 18D3        JR 1C0CH
1C39 48          LD C,B
1C3A F1          POP AF
1C3B EB          EX DE,HL
1C3C C9          RET
1C3D EB          EX DE,HL
1C3E 79          LD A,C
1C3F C1          POP BC
1C40 D1          POP DE
1C41 EB          EX DE,HL
1C42 FE95        CP 95H
1C44 363A        LD (HL),3AH
1C46 2002        JR NZ,1C4AH
1C48 0C          INC C
1C49 23          INC HL
1C4A FEFB        CP FBH
1C4C 200C        JR NZ,1C5AH
1C4E 363A        LD (HL),3AH
1C50 23          INC HL
1C51 0693        LD B,93H
1C53 70          LD (HL),B
1C54 23          INC HL
1C55 EB          EX DE,HL
1C56 0C          INC C
1C57 0C          INC C
1C58 181D        JR 1C77H
1C5A EB          EX DE,HL
1C5B 23          INC HL
1C5C 12          LD (DE),A
1C5D 13          INC DE
1C5E 0C          INC C
1C5F D63A        SUB 3AH
1C61 2804        JR Z,1C67H
1C63 FE4E        CP 4EH
1C65 2003        JR NZ,1C6AH
1C67 32B040      LD (40B0H),A
1C6A D659        SUB 59H
1C6C C2CC1B      JP NZ,1BCCH
1C6F 47          LD B,A
1C70 7E          LD A,(HL)
1C71 B7          OR A
1C72 2809        JR Z,1C7DH
1C74 B8          CP B
1C75 28E4        JR Z,1C5BH
1C77 23          INC HL
1C78 12          LD (DE),A
1C79 0C          INC C
1C7A 13          INC DE
```

1C25-1C7A ROM 78

```
1C7B 18F3        JR 1C70H
1C7D 210500      LD HL,0005H
1C80 44          LD B,H
1C81 09          ADD HL,BC
1C82 44          LD B,H
1C83 4D          LD C,L
1C84 2AA740      LD HL,(40A7H)
1C87 2B          DEC HL
1C88 2B          DEC HL
1C89 2B          DEC HL
1C8A 12          LD (DE),A
1C8B 13          INC DE
1C8C 12          LD (DE),A
1C8D 13          INC DE
1C8E 12          LD (DE),A
1C8F C9          RET
1C90 7C          LD A,H
1C91 92          SUB D
1C92 C0          RET NZ
1C93 7D          LD A,L
1C94 93          SUB E
1C95 C9          RET
1C96 7E          LD A,(HL)
1C97 E3          EX (SP),HL
1C98 BE          CP (HL)
1C99 23          INC HL
1C9A E3          EX (SP),HL
1C9B CA781D      JP Z,1D78H
1C9E C39719      JP 1997H
1CA1 3E64        LD A,64H
1CA3 32DC40      LD (40DCH),A
1CA6 CD211F      CALL 1F21H
1CA9 E3          EX (SP),HL
1CAA CD3619      CALL 1936H
1CAD D1          POP DE
1CAE 2005        JR NZ,1CB5H
1CB0 09          ADD HL,BC
1CB1 F9          LD SP,HL
1CB2 22E840      LD (40E8H),HL
1CB5 EB          EX DE,HL
1CB6 0E08        LD C,08H
1CB8 CD6319      CALL 1963H
1CBB E5          PUSH HL
1CBC CD051F      CALL 1F05H
1CBF E3          EX (SP),HL
1CC0 E5          PUSH HL
1CC1 2AA240      LD HL,(40A2H)
1CC4 E3          EX (SP),HL
1CC5 CF          RST 8
1CC6 BD          CP L
1CC7 E7          RST 20
1CC8 CAF60A      JP Z,0AF6H
1CCB D2F60A      JP NC,0AF6H
1CCE F5          PUSH AF
1CCF CD3723      CALL 2337H
1CD2 F1          POP AF
1CD3 E5          PUSH HL
1CD4 F2EC1C      JP P,1CECH
1CD7 CD7F0A      CALL 0A7FH
1CDA E3          EX (SP),HL
```



```
1CDB 110100      LD DE,0001H
1CDE 7E          LD A,(HL)
1CDF FECC        CP CCH
1CE1 CC012B      CALL Z,2B01H
1CE4 D5          PUSH DE
1CE5 E5          PUSH HL
1CE6 EB          EX DE,HL
1CE7 CD9E09      CALL 099EH
1CEA 1822        JR 1D0EH
1CEC CDB10A      CALL 0AB1H
1CEF CDBF09      CALL 09BFH
1CF2 E1          POP HL
1CF3 C5          PUSH BC
1CF4 D5          PUSH DE
1CF5 010081      LD BC,8100H
1CF8 51          LD D,C
1CF9 5A          LD E,D
1CFA 7E          LD A,(HL)
1CFB FECC        CP CCH
1CFD 3E01        LD A,01H
1CFF 200E        JR NZ,1D0FH
1D01 CD3823      CALL 2338H
1D04 E5          PUSH HL
1D05 CDB10A      CALL 0AB1H
1D08 CDBF09      CALL 09BFH
1D0B CD5509      CALL 0955H
1D0E E1          POP HL
1D0F C5          PUSH BC
1D10 D5          PUSH DE
1D11 4F          LD C,A
1D12 E7          RST 20
1D13 47          LD B,A
1D14 C5          PUSH BC
1D15 E5          PUSH HL
1D16 2ADF40      LD HL,(40DFH)
1D19 E3          EX (SP),HL
1D1A 0681        LD B,81H
1D1C C5          PUSH BC
1D1D 33          INC SP
1D1E CD5803      CALL 0358H
1D21 B7          OR A
1D22 C4A01D      CALL NZ,1DA0H
1D25 22E640      LD (40E6H),HL
1D28 ED73E840    LD (40E8H),SP
1D2C 7E          LD A,(HL)
1D2D FE3A        CP 3AH
1D2F 2829        JR Z,1D5AH
1D31 B7          OR A
1D32 C29719      JP NZ,1997H
1D35 23          INC HL
1D36 7E          LD A,(HL)
1D37 23          INC HL
1D38 B6          OR (HL)
1D39 CA7E19      JP Z,197EH
1D3C 23          INC HL
1D3D 5E          LD E,(HL)
1D3E 23          INC HL
1D3F 56          LD D,(HL)
1D40 EB          EX DE,HL
1D41 22A240      LD (40A2H),HL
```

```
1D44 3A1B41      LD A,(411BH)
1D47 B7          OR A
1D48 280F        JR Z,1D59H
1D4A D5          PUSH DE
1D4B 3E3C        LD A,3CH
1D4D CD2A03      CALL 032AH
1D50 CDAF0F      CALL 0FAFH
1D53 3E3E        LD A,3EH
1D55 CD2A03      CALL 032AH
1D58 D1          POP DE
1D59 EB          EX DE,HL
1D5A D7          RST 10
1D5B 111E1D      LD DE,1D1EH
1D5E D5          PUSH DE
1D5F C8          RET Z
1D60 D680        SUB 80H
1D62 DA211F      JP C,1F21H
1D65 FE3C        CP 3CH
1D67 D2E72A      JP NC,2AE7H
1D6A 07          RLCA
1D6B 4F          LD C,A
1D6C 0600        LD B,00H
1D6E EB          EX DE,HL
1D6F 212218      LD HL,1822H
1D72 09          ADD HL,BC
1D73 4E          LD C,(HL)
1D74 23          INC HL
1D75 46          LD B,(HL)
1D76 C5          PUSH BC
1D77 EB          EX DE,HL
1D78 23          INC HL
1D79 7E          LD A,(HL)
1D7A FE3A        CP 3AH
1D7C D0          RET NC
1D7D FE20        CP 20H
1D7F CA781D      JP Z,1D78H
1D82 FE0B        CP 0BH
1D84 3005        JR NC,1D8BH
1D86 FE09        CP 09H
1D88 D2781D      JP NC,1D78H
1D8B FE30        CP 30H
1D8D 3F          CCF
1D8E 3C          INC A
1D8F 3D          DEC A
1D90 C9          RET
1D91 EB          EX DE,HL
1D92 2AA440      LD HL,(40A4H)
1D95 2B          DEC HL
1D96 22FF40      LD (40FFH),HL
1D99 EB          EX DE,HL
1D9A C9          RET
1D9B CD5803      CALL 0358H
1D9E B7          OR A
1D9F C8          RET Z
1DA0 FE60        CP 60H
1DA2 CC8403      CALL Z,0384H
1DA5 329940      LD (4099H),A
1DA8 3D          DEC A
1DA9 C0          RET NZ
1DAA 3C          INC A
```



```
1DAB C3B41D      JP 1DB4H
1DAE C0          RET NZ
1DAF F5          PUSH AF
1DB0 CCBB41      CALL Z,41BBH
1DB3 F1          POP AF
1DB4 22E640      LD (40E6H),HL
1DB7 21B540      LD HL,40B5H
1DBA 22B340      LD (40B3H),HL
1DBD 21F6FF      LD HL,FFF6H
1DC0 C1          POP BC
1DC1 2AA240      LD HL,(40A2H)
1DC4 E5          PUSH HL
1DC5 F5          PUSH AF
1DC6 7D          LD A,L
1DC7 A4          AND H
1DC8 3C          INC A
1DC9 2809        JR Z,1DD4H
1DCB 22F540      LD (40F5H),HL
1DCE 2AE640      LD HL,(40E6H)
1DD1 22F740      LD (40F7H),HL
1DD4 CD8B03      CALL 038BH
1DD7 CDF920      CALL 20F9H
1DDA F1          POP AF
1DDB 213019      LD HL,1930H
1DDE C2061A      JP NZ,1A06H
1DE1 C3181A      JP 1A18H
1DE4 2AF740      LD HL,(40F7H)
1DE7 7C          LD A,H
1DE8 B5          OR L
1DE9 1E20        LD E,20H
1DEB CAA219      JP Z,19A2H
1DEE EB          EX DE,HL
1DEF 2AF540      LD HL,(40F5H)
1DF2 22A240      LD (40A2H),HL
1DF5 EB          EX DE,HL
1DF6 C9          RET
1DF7 3EAF        LD A,AFH
1DF9 321B41      LD (411BH),A
1DFC C9          RET
1DFD F1          POP AF
1DFE E1          POP HL
1DFF C9          RET
1E00 1E03        LD E,03H
1E02 011E02      LD BC,021EH
1E05 011E04      LD BC,041EH
1E08 011E08      LD BC,081EH
1E0B CD3D1E      CALL 1E3DH
1E0E 019719      LD BC,1997H
1E11 C5          PUSH BC
1E12 D8          RET C
1E13 D641        SUB 41H
1E15 4F          LD C,A
1E16 47          LD B,A
1E17 D7          RST 10
1E18 FECE        CP CEH
1E1A 2009        JR NZ,1E25H
1E1C D7          RST 10
1E1D CD3D1E      CALL 1E3DH
1E20 D8          RET C
1E21 D641        SUB 41H
```

```
1E23 47          LD B,A
1E24 D7          RST 10
1E25 78          LD A,B
1E26 91          SUB C
1E27 D8          RET C
1E28 3C          INC A
1E29 E3          EX (SP),HL
1E2A 210141      LD HL,4101H
1E2D 0600        LD B,00H
1E2F 09          ADD HL,BC
1E30 73          LD (HL),E
1E31 23          INC HL
1E32 3D          DEC A
1E33 20FB        JR NZ,1E30H
1E35 E1          POP HL
1E36 7E          LD A,(HL)
1E37 FE2C        CP 2CH
1E39 C0          RET NZ
1E3A D7          RST 10
1E3B 18CE        JR 1E0BH
1E3D 7E          LD A,(HL)
1E3E FE41        CP 41H
1E40 D8          RET C
1E41 FE5B        CP 5BH
1E43 3F          CCF
1E44 C9          RET
1E45 D7          RST 10
1E46 CD022B      CALL 2B02H
1E49 F0          RET P
1E4A 1E08        LD E,08H
1E4C C3A219      JP 19A2H
1E4F 7E          LD A,(HL)
1E50 FE2E        CP 2EH
1E52 EB          EX DE,HL
1E53 2AEC40      LD HL,(40ECH)
1E56 EB          EX DE,HL
1E57 CA781D      JP Z,1D78H
1E5A 2B          DEC HL
1E5B 110000      LD DE,0000H
1E5E D7          RST 10
1E5F D0          RET NC
1E60 E5          PUSH HL
1E61 F5          PUSH AF
1E62 219819      LD HL,1998H
1E65 DF          RST 18
1E66 DA9719      JP C,1997H
1E69 62          LD H,D
1E6A 6B          LD L,E
1E6B 19          ADD HL,DE
1E6C 29          ADD HL,HL
1E6D 19          ADD HL,DE
1E6E 29          ADD HL,HL
1E6F F1          POP AF
1E70 D630        SUB 30H
1E72 5F          LD E,A
1E73 1600        LD D,00H
1E75 19          ADD HL,DE
1E76 EB          EX DE,HL
1E77 E1          POP HL
1E78 18E4        JR 1E5EH
```



```
1E7A CA611B      JP Z,1B61H
1E7D CD461E      CALL 1E46H
1E80 2B          DEC HL
1E81 D7          RST 10
1E82 C0          RET NZ
1E83 E5          PUSH HL
1E84 2AB140      LD HL,(40B1H)
1E87 7D          LD A,L
1E88 93          SUB E
1E89 5F          LD E,A
1E8A 7C          LD A,H
1E8B 9A          SBC A,D
1E8C 57          LD D,A
1E8D DA7A19      JP C,197AH
1E90 2AF940      LD HL,(40F9H)
1E93 012800      LD BC,0028H
1E96 09          ADD HL,BC
1E97 DF          RST 18
1E98 D27A19      JP NC,197AH
1E9B EB          EX DE,HL
1E9C 22A040      LD (40A0H),HL
1E9F E1          POP HL
1EA0 C3611B      JP 1B61H
1EA3 CA5D1B      JP Z,1B5DH
1EA6 CDC741      CALL 41C7H
1EA9 CD611B      CALL 1B61H
1EAC 011E1D      LD BC,1D1EH
1EAF 1810        JR 1EC1H
1EB1 0E03        LD C,03H
1EB3 CD6319      CALL 1963H
1EB6 C1          POP BC
1EB7 E5          PUSH HL
1EB8 E5          PUSH HL
1EB9 2AA240      LD HL,(40A2H)
1EBC E3          EX (SP),HL
1EBD 3E91        LD A,91H
1EBF F5          PUSH AF
1EC0 33          INC SP
1EC1 C5          PUSH BC
1EC2 CD5A1E      CALL 1E5AH
1EC5 CD071F      CALL 1F07H
1EC8 E5          PUSH HL
1EC9 2AA240      LD HL,(40A2H)
1ECC DF          RST 18
1ECD E1          POP HL
1ECE 23          INC HL
1ECF DC2F1B      CALL C,1B2FH
1ED2 D42C1B      CALL NC,1B2CH
1ED5 60          LD H,B
1ED6 69          LD L,C
1ED7 2B          DEC HL
1ED8 D8          RET C
1ED9 1E0E        LD E,0EH
1EDB C3A219      JP 19A2H
1EDE C0          RET NZ
1EDF 16FF        LD D,FFH
1EE1 CD3619      CALL 1936H
1EE4 F9          LD SP,HL
1EE5 22E840      LD (40E8H),HL
1EE8 FE91        CP 91H
```

```
1EEA 1E04      LD E,04H
1EEC C2A219    JP NZ,19A2H
1EEF E1        POP HL
1EF0 22A240    LD (40A2H),HL
1EF3 23        INC HL
1EF4 7C        LD A,H
1EF5 B5        OR L
1EF6 2007      JR NZ,1EFFH
1EF8 3ADD40    LD A,(40DDH)
1EFB B7        OR A
1EFC C2181A    JP NZ,1A18H
1EFF 211E1D    LD HL,1D1EH
1F02 E3        EX (SP),HL
1F03 3EE1      LD A,E1H
1F05 013A0E    LD BC,0E3AH
1F08 00        NOP
1F09 0600      LD B,00H
1F0B 79        LD A,C
1F0C 48        LD C,B
1F0D 47        LD B,A
1F0E 7E        LD A,(HL)
1F0F B7        OR A
1F10 C8        RET Z
1F11 B8        CP B
1F12 C8        RET Z
1F13 23        INC HL
1F14 FE22      CP 22H
1F16 28F3      JR Z,1F0BH
1F18 D68F      SUB 8FH
1F1A 20F2      JR NZ,1F0EH
1F1C B8        CP B
1F1D 8A        ADC A,D
1F1E 57        LD D,A
1F1F 18ED      JR 1F0EH
1F21 CD0D26    CALL 260DH
1F24 CF        RST 8
1F25 D5        PUSH DE
1F26 EB        EX DE,HL
1F27 22DF40    LD (40DFH),HL
1F2A EB        EX DE,HL
1F2B D5        PUSH DE
1F2C E7        RST 20
1F2D F5        PUSH AF
1F2E CD3723    CALL 2337H
1F31 F1        POP AF
1F32 E3        EX (SP),HL
1F33 C603      ADD A,03H
1F35 CD1928    CALL 2819H
1F38 CD030A    CALL 0A03H
1F3B E5        PUSH HL
1F3C 2028      JR NZ,1F66H
1F3E 2A2141    LD HL,(4121H)
1F41 E5        PUSH HL
1F42 23        INC HL
1F43 5E        LD E,(HL)
1F44 23        INC HL
1F45 56        LD D,(HL)
1F46 2AA440    LD HL,(40A4H)
1F49 DF        RST 18
1F4A 300E      JR NC,1F5AH
```



```
1F4C 2AA040    LD HL,(40A0H)
1F4F DF        RST 18
1F50 D1        POP DE
1F51 300F      JR NC,1F62H
1F53 2AF940    LD HL,(40F9H)
1F56 DF        RST 18
1F57 3009      JR NC,1F62H
1F59 3ED1      LD A,D1H
1F5B CDF529    CALL 29F5H
1F5E EB        EX DE,HL
1F5F CD4328    CALL 2843H
1F62 CDF529    CALL 29F5H
1F65 E3        EX (SP),HL
1F66 CDD309    CALL 09D3H
1F69 D1        POP DE
1F6A E1        POP HL
1F6B C9        RET
1F6C FE9E      CP 9EH
1F6E 2025      JR NZ,1F95H
1F70 D7        RST 10
1F71 CF        RST 8
1F72 8D        ADC A,L
1F73 CD5A1E    CALL 1E5AH
1F76 7A        LD A,D
1F77 B3        OR E
1F78 2809      JR Z,1F83H
1F7A CD2A1B    CALL 1B2AH
1F7D 50        LD D,B
1F7E 59        LD E,C
1F7F E1        POP HL
1F80 D2D91E    JP NC,1ED9H
1F83 EB        EX DE,HL
1F84 22F040    LD (40F0H),HL
1F87 EB        EX DE,HL
1F88 D8        RET C
1F89 3AF240    LD A,(40F2H)
1F8C B7        OR A
1F8D C8        RET Z
1F8E 3A9A40    LD A,(409AH)
1F91 5F        LD E,A
1F92 C3AB19    JP 19ABH
1F95 CD1C2B    CALL 2B1CH
1F98 7E        LD A,(HL)
1F99 47        LD B,A
1F9A FE91      CP 91H
1F9C 2803      JR Z,1FA1H
1F9E CF        RST 8
1F9F 8D        ADC A,L
1FA0 2B        DEC HL
1FA1 4B        LD C,E
1FA2 0D        DEC C
1FA3 78        LD A,B
1FA4 CA601D    JP Z,1D60H
1FA7 CD5B1E    CALL 1E5BH
1FAA FE2C      CP 2CH
1FAC C0        RET NZ
1FAD 18F3      JR 1FA2H
1FAF 11F240    LD DE,40F2H
1FB2 1A        LD A,(DE)
1FB3 B7        OR A
```

```
1FB4 CAA019      JP Z,19A0H
1FB7 3C          INC A
1FB8 329A40      LD (409AH),A
1FBB 12          LD (DE),A
1FBC 7E          LD A,(HL)
1FBD FE87        CP 87H
1FBF 280C        JR Z,1FCDH
1FC1 CD5A1E      CALL 1E5AH
1FC4 C0          RET NZ
1FC5 7A          LD A,D
1FC6 B3          OR E
1FC7 C2C51E      JP NZ,1EC5H
1FCA 3C          INC A
1FCB 1802        JR 1FCFH
1FCD D7          RST 10
1FCE C0          RET NZ
1FCF 2AEE40      LD HL,(40EEH)
1FD2 EB          EX DE,HL
1FD3 2AEA40      LD HL,(40EAH)
1FD6 22A240      LD (40A2H),HL
1FD9 EB          EX DE,HL
1FDA C0          RET NZ
1FDB 7E          LD A,(HL)
1FDC B7          OR A
1FDD 2004        JR NZ,1FE3H
1FDF 23          INC HL
1FE0 23          INC HL
1FE1 23          INC HL
1FE2 23          INC HL
1FE3 23          INC HL
1FE4 7A          LD A,D
1FE5 A3          AND E
1FE6 3C          INC A
1FE7 C2051F      JP NZ,1F05H
1FEA 3ADD40      LD A,(40DDH)
1FED 3D          DEC A
1FEE CABE1D      JP Z,1DBEH
1FF1 C3051F      JP 1F05H
1FF4 CD1C2B      CALL 2B1CH
1FF7 C0          RET NZ
1FF8 B7          OR A
1FF9 CA4A1E      JP Z,1E4AH
1FFC 3D          DEC A
1FFD 87          ADD A,A
1FFE 5F          LD E,A
1FFF FE2D        CP 2DH
2001 3802        JR C,2005H
2003 1E26        LD E,26H
2005 C3A219      JP 19A2H
2008 110A00      LD DE,000AH
200B D5          PUSH DE
200C 2817        JR Z,2025H
200E CD4F1E      CALL 1E4FH
2011 EB          EX DE,HL
2012 E3          EX (SP),HL
2013 2811        JR Z,2026H
2015 EB          EX DE,HL
2016 CF          RST 8
2017 2C          INC L
2018 EB          EX DE,HL
```

1FB4-2018 ROM 87

```
2019 2AE440      LD HL,(40E4H)
201C EB          EX DE,HL
201D 2806        JR Z,2025H
201F CD5A1E      CALL 1E5AH
2022 C29719      JP NZ,1997H
2025 EB          EX DE,HL
2026 7C          LD A,H
2027 B5          OR L
2028 CA4A1E      JP Z,1E4AH
202B 22E440      LD (40E4H),HL
202E 32E140      LD (40E1H),A
2031 E1          POP HL
2032 22E240      LD (40E2H),HL
2035 C1          POP BC
2036 C3331A      JP 1A33H
2039 CD3723      CALL 2337H
203C 7E          LD A,(HL)
203D FE2C        CP 2CH
203F CC781D      CALL Z,1D78H
2042 FECA        CP CAH
2044 CC781D      CALL Z,1D78H
2047 2B          DEC HL
2048 E5          PUSH HL
2049 CD9409      CALL 0994H
204C E1          POP HL
204D 2807        JR Z,2056H
204F D7          RST 10
2050 DAC21E      JP C,1EC2H
2053 C35F1D      JP 1D5FH
2056 1601        LD D,01H
2058 CD051F      CALL 1F05H
205B B7          OR A
205C C8          RET Z
205D D7          RST 10
205E FE95        CP 95H
2060 20F6        JR NZ,2058H
2062 15          DEC D
2063 20F3        JR NZ,2058H
2065 18E8        JR 204FH
2067 3E01        LD A,01H
2069 329C40      LD (409CH),A
206C C39B20      JP 209BH
206F CDCA41      CALL 41CAH
2072 FE40        CP 40H
2074 2019        JR NZ,208FH
2076 CD012B      CALL 2B01H
2079 FE04        CP 04H
207B D24A1E      JP NC,1E4AH
207E E5          PUSH HL
207F 21003C      LD HL,3C00H
2082 19          ADD HL,DE
2083 222040      LD (4020H),HL
2086 7B          LD A,E
2087 E63F        AND 3FH
2089 32A640      LD (40A6H),A
208C E1          POP HL
208D CF          RST 8
208E 2C          INC L
208F FE23        CP 23H
2091 2008        JR NZ,209BH
```

2019-2091 ROM 88

```
2093 CD8402     CALL 0284H
2096 3E80       LD A,80H
2098 329C40     LD (409CH),A
209B 2B         DEC HL
209C D7         RST 10
209D CCFE20     CALL Z,20FEH
20A0 CA6921     JP Z,2169H
20A3 FEBF       CP BFH
20A5 CABD2C     JP Z,2CBDH
20A8 FEBC       CP BCH
20AA CA3721     JP Z,2137H
20AD E5         PUSH HL
20AE FE2C       CP 2CH
20B0 CA0821     JP Z,2108H
20B3 FE3B       CP 3BH
20B5 CA6421     JP Z,2164H
20B8 C1         POP BC
20B9 CD3723     CALL 2337H
20BC E5         PUSH HL
20BD E7         RST 20
20BE 2832       JR Z,20F2H
20C0 CDBD0F     CALL 0FBDH
20C3 CD6528     CALL 2865H
20C6 CDCD41     CALL 41CDH
20C9 2A2141     LD HL,(4121H)
20CC 3A9C40     LD A,(409CH)
20CF B7         OR A
20D0 FAE920     JP N,20E9H
20D3 2808       JR Z,20DDH
20D5 3A9B40     LD A,(409BH)
20D8 86         ADD A,(HL)
20D9 FE84       CP 84H
20DB 1809       JR 20E6H
20DD 3A9D40     LD A,(409DH)
20E0 47         LD B,A
20E1 3AA640     LD A,(40A6H)
20E4 86         ADD A,(HL)
20E5 B8         CP B
20E6 D4FE20     CALL NC,20FEH
20E9 CDAA28     CALL 28AAH
20EC 3E20       LD A,20H
20EE CD2A03     CALL 032AH
20F1 B7         OR A
20F2 CCAA28     CALL Z,28AAH
20F5 E1         POP HL
20F6 C39B20     JP 209BH
20F9 3AA640     LD A,(40A6H)
20FC B7         OR A
20FD C8         RET Z
20FE 3E0D       LD A,0DH
2100 CD2A03     CALL 032AH
2103 CDD041     CALL 41D0H
2106 AF         XOR A
2107 C9         RET
2108 CDD341     CALL 41D3H
210B 3A9C40     LD A,(409CH)
210E B7         OR A
210F F21921     JP P,2119H
2112 3E2C       LD A,2CH
2114 CD2A03     CALL 032AH
```

2093-2114 ROM 89

```
2117 184B       JR 2164H
2119 2808       JR Z,2123H
211B 3A9B40     LD A,(409BH)
211E FE70       CP 70H
2120 C32B21     JP 212BH
2123 3A9E40     LD A,(409EH)
2126 47         LD B,A
2127 3AA640     LD A,(40A6H)
212A B8         CP B
212B D4FE20     CALL NC,20FEH
212E 3034       JR NC,2164H
2130 D610       SUB 10H
2132 30FC       JR NC,2130H
2134 2F         CPL
2135 1823       JR 215AH
2137 CD1B2B     CALL 2B1BH
213A E63F       AND 3FH
213C 5F         LD E,A
213D CF         RST 8
213E 29         ADD HL,HL
213F 2B         DEC HL
2140 E5         PUSH HL
2141 CDD341     CALL 41D3H
2144 3A9C40     LD A,(409CH)
2147 B7         OR A
2148 FA4A1E     JP N,1E4AH
214B CA5321     JP Z,2153H
214E 3A9B40     LD A,(409BH)
2151 1803       JR 2156H
2153 3AA640     LD A,(40A6H)
2156 2F         CPL
2157 83         ADD A,E
2158 300A       JR NC,2164H
215A 3C         INC A
215B 47         LD B,A
215C 3E20       LD A,20H
215E CD2A03     CALL 032AH
2161 05         DEC B
2162 20FA       JR NZ,215EH
2164 E1         POP HL
2165 D7         RST 10
2166 C3A020     JP 20A0H
2169 3A9C40     LD A,(409CH)
216C B7         OR A
216D FCF801     CALL N,01F8H
2170 AF         XOR A
2171 329C40     LD (409CH),A
2174 CDBE41     CALL 41BEH
2177 C9         RET
```

2117-2177 ROM 90

```
2178  3F        ?   ?
2179  52        R   R
217A  45        E   E
217B  44        D   D
217C  4F        O   O
217D  0D        .   .
217E  00        .   .
```

2117-217E ROM 90

```
217F 3ADE40      LD A,(40DEH)
2182 B7          OR A
2183 C29119      JP NZ,1991H
2186 3AA940      LD A,(40A9H)
2189 B7          OR A
218A 1E2A        LD E,2AH
218C CAA219      JP Z,19A2H
218F C1          POP BC
2190 217821      LD HL,2178H
2193 CDA728      CALL 28A7H
2196 2AE640      LD HL,(40E6H)
2199 C9          RET
219A CD2828      CALL 2828H
219D 7E          LD A,(HL)
219E CDD641      CALL 41D6H
21A1 D623        SUB 23H
21A3 32A940      LD (40A9H),A
21A6 7E          LD A,(HL)
21A7 2020        JR NZ,21C9H
21A9 CD9302      CALL 0293H
21AC E5          PUSH HL
21AD 06FA        LD B,FAH
21AF 2AA740      LD HL,(40A7H)
21B2 CD3502      CALL 0235H
21B5 77          LD (HL),A
21B6 23          INC HL
21B7 FE0D        CP 0DH
21B9 2802        JR Z,21BDH
21BB 10F5        DJNZ 21B2H
21BD 2B          DEC HL
21BE 3600        LD (HL),00H
21C0 CDF801      CALL 01F8H
21C3 2AA740      LD HL,(40A7H)
21C6 2B          DEC HL
21C7 1822        JR 21EBH
21C9 01DB21      LD BC,21DBH
21CC C5          PUSH BC
21CD FE22        CP 22H
21CF C0          RET NZ
21D0 CD6628      CALL 2866H
21D3 CF          RST 8
21D4 3B          DEC SP
21D5 E5          PUSH HL
21D6 CDAA28      CALL 28AAH
21D9 E1          POP HL
21DA C9          RET
21DB E5          PUSH HL
21DC CDB31B      CALL 1BB3H
21DF C1          POP BC
21E0 DABE1D      JP C,1DBEH
21E3 23          INC HL
21E4 7E          LD A,(HL)
21E5 B7          OR A
21E6 2B          DEC HL
21E7 C5          PUSH BC
21E8 CA041F      JP Z,1F04H
21EB 362C        LD (HL),2CH
21ED 1805        JR 21F4H
21EF E5          PUSH HL
21F0 2AFF40      LD HL,(40FFH)
```



```
21F3 F6AF        OR AFH
21F5 32DE40      LD (40DEH),A
21F8 E3          EX (SP),HL
21F9 1802        JR 21FDH
21FB CF          RST 8
21FC 2C          INC L
21FD CD0D26      CALL 260DH
2200 E3          EX (SP),HL
2201 D5          PUSH DE
2202 7E          LD A,(HL)
2203 FE2C        CP 2CH
2205 2826        JR Z,222DH
2207 3ADE40      LD A,(40DEH)
220A B7          OR A
220B C29622      JP NZ,2296H
220E 3AA940      LD A,(40A9H)
2211 B7          OR A
2212 1E06        LD E,06H
2214 CAA219      JP Z,19A2H
2217 3E3F        LD A,3FH
2219 CD2A03      CALL 032AH
221C CDB31B      CALL 1BB3H
221F D1          POP DE
2220 C1          POP BC
2221 DABE1D      JP C,1DBEH
2224 23          INC HL
2225 7E          LD A,(HL)
2226 B7          OR A
2227 2B          DEC HL
2228 C5          PUSH BC
2229 CA041F      JP Z,1F04H
222C D5          PUSH DE
222D CDDC41      CALL 41DCH
2230 E7          RST 20
2231 F5          PUSH AF
2232 2019        JR NZ,224DH
2234 D7          RST 10
2235 57          LD D,A
2236 47          LD B,A
2237 FE22        CP 22H
2239 2805        JR Z,2240H
223B 163A        LD D,3AH
223D 062C        LD B,2CH
223F 2B          DEC HL
2240 CD6928      CALL 2869H
2243 F1          POP AF
2244 EB          EX DE,HL
2245 215A22      LD HL,225AH
2248 E3          EX (SP),HL
2249 D5          PUSH DE
224A C3331F      JP 1F33H
224D D7          RST 10
224E F1          POP AF
224F F5          PUSH AF
2250 014322      LD BC,2243H
2253 C5          PUSH BC
2254 DA6C0E      JP C,0E6CH
2257 D2650E      JP NC,0E65H
225A 2B          DEC HL
225B D7          RST 10
```

```
225C 2805        JR Z,2263H
225E FE2C        CP 2CH
2260 C27F21      JP NZ,217FH
2263 E3          EX (SP),HL
2264 2B          DEC HL
2265 D7          RST 10
2266 C2FB21      JP NZ,21FBH
2269 D1          POP DE
226A 00          NOP
226B 00          NOP
226C 00          NOP
226D 00          NOP
226E 00          NOP
226F 3ADE40      LD A,(40DEH)
2272 B7          OR A
2273 EB          EX DE,HL
2274 C2961D      JP NZ,1D96H
2277 D5          PUSH DE
2278 CDDF41      CALL 41DFH
227B B6          OR (HL)
227C 218622      LD HL,2286H
227F C4A728      CALL NZ,28A7H
2282 E1          POP HL
2283 C36921      JP 2169H
```

225C-2283 ROM 93

```
2286  3F        ?   ?
2287  45        E   E
2288  78        x   x
2289  74        t   t
228A  72        r   r
228B  61        a   a
228C  20
228D  69        i   i
228E  67        g   g
228F  6E        n   n
2290  6F        o   o
2291  72        r   r
2292  65        e   e
2293  64        d   d
2294  0D        .   .
2295  00        .   .
```

225C-2295 ROM 93

```
2296 CD051F      CALL 1F05H
2299 B7          OR A
229A 2012        JR NZ,22AEH
229C 23          INC HL
229D 7E          LD A,(HL)
229E 23          INC HL
229F B6          OR (HL)
22A0 1E06        LD E,06H
22A2 CAA219      JP Z,19A2H
22A5 23          INC HL
22A6 5E          LD E,(HL)
22A7 23          INC HL
22A8 56          LD D,(HL)
22A9 EB          EX DE,HL
```

225C-22A9 ROM 93

```
22AA 22DA40      LD (40DAH),HL
22AD EB          EX DE,HL
22AE D7          RST 10
22AF FE88        CP 88H
22B1 20E3        JR NZ,2296H
22B3 C32D22      JP 222DH
22B6 110000      LD DE,0000H
22B9 C40D26      CALL NZ,260DH
22BC 22DF40      LD (40DFH),HL
22BF CD3619      CALL 1936H
22C2 C29D19      JP NZ,199DH
22C5 F9          LD SP,HL
22C6 22E840      LD (40E8H),HL
22C9 D5          PUSH DE
22CA 7E          LD A,(HL)
22CB 23          INC HL
22CC F5          PUSH AF
22CD D5          PUSH DE
22CE 7E          LD A,(HL)
22CF 23          INC HL
22D0 B7          OR A
22D1 FAEA22      JP M,22EAH
22D4 CDB109      CALL 09B1H
22D7 E3          EX (SP),HL
22D8 E5          PUSH HL
22D9 CD0B07      CALL 070BH
22DC E1          POP HL
22DD CDCB09      CALL 09CBH
22E0 E1          POP HL
22E1 CDC209      CALL 09C2H
22E4 E5          PUSH HL
22E5 CD0C0A      CALL 0A0CH
22E8 1829        JR 2313H
22EA 23          INC HL
22EB 23          INC HL
22EC 23          INC HL
22ED 23          INC HL
22EE 4E          LD C,(HL)
22EF 23          INC HL
22F0 46          LD B,(HL)
22F1 23          INC HL
22F2 E3          EX (SP),HL
22F3 5E          LD E,(HL)
22F4 23          INC HL
22F5 56          LD D,(HL)
22F6 E5          PUSH HL
22F7 69          LD L,C
22F8 60          LD H,B
22F9 CDD20B      CALL 0BD2H
22FC 3AAF40      LD A,(40AFH)
22FF FE04        CP 04H
2301 CAB207      JP Z,07B2H
2304 EB          EX DE,HL
2305 E1          POP HL
2306 72          LD (HL),D
2307 2B          DEC HL
2308 73          LD (HL),E
2309 E1          POP HL
230A D5          PUSH DE
230B 5E          LD E,(HL)
```

22AA-230B ROM 94

```
230C 23          INC HL
230D 56          LD D,(HL)
230E 23          INC HL
230F E3          EX (SP),HL
2310 CD390A      CALL 0A39H
2313 E1          POP HL
2314 C1          POP BC
2315 90          SUB B
2316 CDC209      CALL 09C2H
2319 2809        JR Z,2324H
231B EB          EX DE,HL
231C 22A240      LD (40A2H),HL
231F 69          LD L,C
2320 60          LD H,B
2321 C31A1D      JP 1D1AH
2324 F9          LD SP,HL
2325 22E840      LD (40E8H),HL
2328 2ADF40      LD HL,(40DFH)
232B 7E          LD A,(HL)
232C FE2C        CP 2CH
232E C21E1D      JP NZ,1D1EH
2331 D7          RST 10
2332 CDB922      CALL 22B9H
2335 CF          RST 8
2336 282B        JR Z,2363H
2338 1600        LD D,00H
233A D5          PUSH DE
233B 0E01        LD C,01H
233D CD6319      CALL 1963H
2340 CD9F24      CALL 249FH
2343 22F340      LD (40F3H),HL
2346 2AF340      LD HL,(40F3H)
2349 C1          POP BC
234A 7E          LD A,(HL)
234B 1600        LD D,00H
234D D6D4        SUB D4H
234F 3813        JR C,2364H
2351 FE03        CP 03H
2353 300F        JR NC,2364H
2355 FE01        CP 01H
2357 17          RLA
2358 AA          XOR D
2359 BA          CP D
235A 57          LD D,A
235B DA9719      JP C,1997H
235E 22D840      LD (40D8H),HL
2361 D7          RST 10
2362 18E9        JR 234DH
2364 7A          LD A,D
2365 B7          OR A
2366 C2EC23      JP NZ,23ECH
2369 7E          LD A,(HL)
236A 22D840      LD (40D8H),HL
236D D6CD        SUB CDH
236F D8          RET C
2370 FE07        CP 07H
2372 D0          RET NC
2373 5F          LD E,A
2374 3AAF40      LD A,(40AFH)
2377 D603        SUB 03H
```



```
2379 B3          OR E
237A CA8F29      JP Z,298FH
237D 219A18      LD HL,189AH
2380 19          ADD HL,DE
2381 78          LD A,B
2382 56          LD D,(HL)
2383 BA          CP D
2384 D0          RET NC
2385 C5          PUSH BC
2386 014623      LD BC,2346H
2389 C5          PUSH BC
238A 7A          LD A,D
238B FE7F        CP 7FH
238D CAD423      JP Z,23D4H
2390 FE51        CP 51H
2392 DAE123      JP C,23E1H
2395 212141      LD HL,4121H
2398 B7          OR A
2399 3AAF40      LD A,(40AFH)
239C 3D          DEC A
239D 3D          DEC A
239E 3D          DEC A
239F CAF60A      JP Z,0AF6H
23A2 4E          LD C,(HL)
23A3 23          INC HL
23A4 46          LD B,(HL)
23A5 C5          PUSH BC
23A6 FAC523      JP N,23C5H
23A9 23          INC HL
23AA 4E          LD C,(HL)
23AB 23          INC HL
23AC 46          LD B,(HL)
23AD C5          PUSH BC
23AE F5          PUSH AF
23AF B7          OR A
23B0 E2C423      JP PO,23C4H
23B3 F1          POP AF
23B4 23          INC HL
23B5 3803        JR C,23BAH
23B7 211D41      LD HL,411DH
23BA 4E          LD C,(HL)
23BB 23          INC HL
23BC 46          LD B,(HL)
23BD 23          INC HL
23BE C5          PUSH BC
23BF 4E          LD C,(HL)
23C0 23          INC HL
23C1 46          LD B,(HL)
23C2 C5          PUSH BC
23C3 06F1        LD B,F1H
23C5 C603        ADD A,03H
23C7 4B          LD C,E
23C8 47          LD B,A
23C9 C5          PUSH BC
23CA 010624      LD BC,2406H
23CD C5          PUSH BC
23CE 2AD840      LD HL,(40D8H)
23D1 C33A23      JP 233AH
23D4 CDB10A      CALL 0AB1H
23D7 CDA409      CALL 09A4H
```

```
23DA 01F213      LD BC,13F2H
23DD 167F        LD D,7FH
23DF 18EC        JR 23CDH
23E1 D5          PUSH DE
23E2 CD7F0A      CALL 0A7FH
23E5 D1          POP DE
23E6 E5          PUSH HL
23E7 01E925      LD BC,25E9H
23EA 18E1        JR 23CDH
23EC 78          LD A,B
23ED FE64        CP 64H
23EF D0          RET NC
23F0 C5          PUSH BC
23F1 D5          PUSH DE
23F2 110464      LD DE,6404H
23F5 21B825      LD HL,25B8H
23F8 E5          PUSH HL
23F9 E7          RST 20
23FA C29523      JP NZ,2395H
23FD 2A2141      LD HL,(4121H)
2400 E5          PUSH HL
2401 018C25      LD BC,258CH
2404 18C7        JR 23CDH
2406 C1          POP BC
2407 79          LD A,C
2408 32B040      LD (40B0H),A
240B 78          LD A,B
240C FE08        CP 08H
240E 2828        JR Z,2438H
2410 3AAF40      LD A,(40AFH)
2413 FE08        CP 08H
2415 CA6024      JP Z,2460H
2418 57          LD D,A
2419 78          LD A,B
241A FE04        CP 04H
241C CA7224      JP Z,2472H
241F 7A          LD A,D
2420 FE03        CP 03H
2422 CAF60A      JP Z,0AF6H
2425 D27C24      JP NC,247CH
2428 21BF18      LD HL,18BFH
242B 0600        LD B,00H
242D 09          ADD HL,BC
242E 09          ADD HL,BC
242F 4E          LD C,(HL)
2430 23          INC HL
2431 46          LD B,(HL)
2432 D1          POP DE
2433 2A2141      LD HL,(4121H)
2436 C5          PUSH BC
2437 C9          RET
2438 CDDB0A      CALL 0ADBH
243B CDFC09      CALL 09FCH
243E E1          POP HL
243F 221F41      LD (411FH),HL
2442 E1          POP HL
2443 221D41      LD (411DH),HL
2446 C1          POP BC
2447 D1          POP DE
2448 CDB409      CALL 09B4H
```

23DA-2448 ROM 97

```
244B CDDB0A      CALL 0ADBH
244E 21AB18      LD HL,18ABH
2451 3AB040      LD A,(40B0H)
2454 07          RLCA
2455 C5          PUSH BC
2456 4F          LD C,A
2457 0600        LD B,00H
2459 09          ADD HL,BC
245A C1          POP BC
245B 7E          LD A,(HL)
245C 23          INC HL
245D 66          LD H,(HL)
245E 6F          LD L,A
245F E9          JP (HL)
2460 C5          PUSH BC
2461 CDFC09      CALL 09FCH
2464 F1          POP AF
2465 32AF40      LD (40AFH),A
2468 FE04        CP 04H
246A 28DA        JR Z,2446H
246C E1          POP HL
246D 222141      LD (4121H),HL
2470 18D9        JR 244BH
2472 CDB10A      CALL 0AB1H
2475 C1          POP BC
2476 D1          POP DE
2477 21B518      LD HL,18B5H
247A 18D5        JR 2451H
247C E1          POP HL
247D CDA409      CALL 09A4H
2480 CDCF0A      CALL 0ACFH
2483 CDBF09      CALL 09BFH
2486 E1          POP HL
2487 222341      LD (4123H),HL
248A E1          POP HL
248B 222141      LD (4121H),HL
248E 18E7        JR 2477H
2490 E5          PUSH HL
2491 EB          EX DE,HL
2492 CDCF0A      CALL 0ACFH
2495 E1          POP HL
2496 CDA409      CALL 09A4H
2499 CDCF0A      CALL 0ACFH
249C C3A008      JP 08A0H
249F D7          RST 10
24A0 1E28        LD E,28H
24A2 CAA219      JP Z,19A2H
24A5 DA6C0E      JP C,0E6CH
24A8 CD3D1E      CALL 1E3DH
24AB D24025      JP NC,2540H
24AE FECD        CP CDH
24B0 28ED        JR Z,249FH
24B2 FE2E        CP 2EH
24B4 CA6C0E      JP Z,0E6CH
24B7 FECE        CP CEH
24B9 CA3225      JP Z,2532H
24BC FE22        CP 22H
24BE CA6628      JP Z,2866H
24C1 FECB        CP CBH
24C3 CAC425      JP Z,25C4H
```

244B-24C3 ROM 98

```
24C6 FE26      CP 26H
24C8 CA9441    JP Z,4194H
24CB FEC3      CP C3H
24CD 200A      JR NZ,24D9H
24CF D7        RST 10
24D0 3A9A40    LD A,(409AH)
24D3 E5        PUSH HL
24D4 CDF827    CALL 27F8H
24D7 E1        POP HL
24D8 C9        RET
24D9 FEC2      CP C2H
24DB 200A      JR NZ,24E7H
24DD D7        RST 10
24DE E5        PUSH HL
24DF 2AEA40    LD HL,(40EAH)
24E2 CD660C    CALL 0C66H
24E5 E1        POP HL
24E6 C9        RET
24E7 FEC0      CP C0H
24E9 2014      JR NZ,24FFH
24EB D7        RST 10
24EC CF        RST 8
24ED 28CD      JR Z,24BCH
24EF 0D        DEC C
24F0 26CF      LD H,CFH
24F2 29        ADD HL,HL
24F3 E5        PUSH HL
24F4 EB        EX DE,HL
24F5 7C        LD A,H
24F6 B5        OR L
24F7 CA4A1E    JP Z,1E4AH
24FA CD9A0A    CALL 0A9AH
24FD E1        POP HL
24FE C9        RET
24FF FEC1      CP C1H
2501 CAFE27    JP Z,27FEH
2504 FEC5      CP C5H
2506 CA9D41    JP Z,419DH
2509 FEC8      CP C8H
250B CAC927    JP Z,27C9H
250E FEC7      CP C7H
2510 CA7641    JP Z,4176H
2513 FEC6      CP C6H
2515 CA3201    JP Z,0132H
2518 FEC9      CP C9H
251A CA9D01    JP Z,019DH
251D FEC4      CP C4H
251F CA2F2A    JP Z,2A2FH
2522 FEBE      CP BEH
2524 CA5541    JP Z,4155H
2527 D6D7      SUB D7H
2529 D24E25    JP NC,254EH
252C CD3523    CALL 2335H
252F CF        RST 8
2530 29        ADD HL,HL
2531 C9        RET
2532 167D      LD D,7DH
2534 CD3A23    CALL 233AH
2537 2AF340    LD HL,(40F3H)
253A E5        PUSH HL
```



```
253B CD7B09    CALL 097BH
253E E1        POP HL
253F C9        RET
2540 CD0D26    CALL 260DH
2543 E5        PUSH HL
2544 EB        EX DE,HL
2545 222141    LD (4121H),HL
2548 E7        RST 20
2549 C4F709    CALL NZ,09F7H
254C E1        POP HL
254D C9        RET
254E 0600      LD B,00H
2550 07        RLCA
2551 4F        LD C,A
2552 C5        PUSH BC
2553 D7        RST 10
2554 79        LD A,C
2555 FE41      CP 41H
2557 3816      JR C,256FH
2559 CD3523    CALL 2335H
255C CF        RST 8
255D 2C        INC L
255E CDF40A    CALL 0AF4H
2561 EB        EX DE,HL
2562 2A2141    LD HL,(4121H)
2565 E3        EX (SP),HL
2566 E5        PUSH HL
2567 EB        EX DE,HL
2568 CD1C2B    CALL 2B1CH
256B EB        EX DE,HL
256C E3        EX (SP),HL
256D 1814      JR 2583H
256F CD2C25    CALL 252CH
2572 E3        EX (SP),HL
2573 7D        LD A,L
2574 FE0C      CP 0CH
2576 3807      JR C,257FH
2578 FE1B      CP 1BH
257A E5        PUSH HL
257B DCB10A    CALL C,0AB1H
257E E1        POP HL
257F 113E25    LD DE,253EH
2582 D5        PUSH DE
2583 010816    LD BC,1608H
2586 09        ADD HL,BC
2587 4E        LD C,(HL)
2588 23        INC HL
2589 66        LD H,(HL)
258A 69        LD L,C
258B E9        JP (HL)
258C CDD729    CALL 29D7H
258F 7E        LD A,(HL)
2590 23        INC HL
2591 4E        LD C,(HL)
2592 23        INC HL
2593 46        LD B,(HL)
2594 D1        POP DE
2595 C5        PUSH BC
2596 F5        PUSH AF
2597 CDDE29    CALL 29DEH
```

```
259A D1          POP DE
259B 5E          LD E,(HL)
259C 23          INC HL
259D 4E          LD C,(HL)
259E 23          INC HL
259F 46          LD B,(HL)
25A0 E1          POP HL
25A1 7B          LD A,E
25A2 B2          OR D
25A3 C8          RET Z
25A4 7A          LD A,D
25A5 D601        SUB 01H
25A7 D8          RET C
25A8 AF          XOR A
25A9 BB          CP E
25AA 3C          INC A
25AB D0          RET NC
25AC 15          DEC D
25AD 1D          DEC E
25AE 0A          LD A,(BC)
25AF BE          CP (HL)
25B0 23          INC HL
25B1 03          INC BC
25B2 28ED        JR Z,25A1H
25B4 3F          CCF
25B5 C36009      JP 0960H
25B8 3C          INC A
25B9 8F          ADC A,A
25BA C1          POP BC
25BB A0          AND B
25BC C6FF        ADD A,FFH
25BE 9F          SBC A,A
25BF CD8D09      CALL 098DH
25C2 1812        JR 25D6H
25C4 165A        LD D,5AH
25C6 CD3A23      CALL 233AH
25C9 CD7F0A      CALL 0A7FH
25CC 7D          LD A,L
25CD 2F          CPL
25CE 6F          LD L,A
25CF 7C          LD A,H
25D0 2F          CPL
25D1 67          LD H,A
25D2 222141      LD (4121H),HL
25D5 C1          POP BC
25D6 C34623      JP 2346H
25D9 3AAF40      LD A,(40AFH)
25DC FE08        CP 08H
25DE 3005        JR NC,25E5H
25E0 D603        SUB 03H
25E2 B7          OR A
25E3 37          SCF
25E4 C9          RET
25E5 D603        SUB 03H
25E7 B7          OR A
25E8 C9          RET
25E9 C5          PUSH BC
25EA CD7F0A      CALL 0A7FH
25ED F1          POP AF
25EE D1          POP DE
```



```
25EF 01FA27      LD BC,27FAH
25F2 C5          PUSH BC
25F3 FE46        CP 46H
25F5 2006        JR NZ,25FDH
25F7 7B          LD A,E
25F8 B5          OR L
25F9 6F          LD L,A
25FA 7C          LD A,H
25FB B2          OR D
25FC C9          RET
25FD 7B          LD A,E
25FE A5          AND L
25FF 6F          LD L,A
2600 7C          LD A,H
2601 A2          AND D
2602 C9          RET
2603 2B          DEC HL
2604 D7          RST 10
2605 C8          RET Z
2606 CF          RST 8
2607 2C          INC L
2608 010326      LD BC,2603H
260B C5          PUSH BC
260C F6AF        OR AFH
260E 32AE40      LD (40AEH),A
2611 46          LD B,(HL)
2612 CD3D1E      CALL 1E3DH
2615 DA9719      JP C,1997H
2618 AF          XOR A
2619 4F          LD C,A
261A D7          RST 10
261B 3805        JR C,2622H
261D CD3D1E      CALL 1E3DH
2620 3809        JR C,262BH
2622 4F          LD C,A
2623 D7          RST 10
2624 38FD        JR C,2623H
2626 CD3D1E      CALL 1E3DH
2629 30F8        JR NC,2623H
262B 115226      LD DE,2652H
262E D5          PUSH DE
262F 1602        LD D,02H
2631 FE25        CP 25H
2633 C8          RET Z
2634 14          INC D
2635 FE24        CP 24H
2637 C8          RET Z
2638 14          INC D
2639 FE21        CP 21H
263B C8          RET Z
263C 1608        LD D,08H
263E FE23        CP 23H
2640 C8          RET Z
2641 78          LD A,B
2642 D641        SUB 41H
2644 E67F        AND 7FH
2646 5F          LD E,A
2647 1600        LD D,00H
2649 E5          PUSH HL
264A 210141      LD HL,4101H
```

```
264D 19          ADD HL,DE
264E 56          LD D,(HL)
264F E1          POP HL
2650 2B          DEC HL
2651 C9          RET
2652 7A          LD A,D
2653 32AF40      LD (40AFH),A
2656 D7          RST 10
2657 3ADC40      LD A,(40DCH)
265A B7          OR A
265B C26426      JP NZ,2664H
265E 7E          LD A,(HL)
265F D628        SUB 28H
2661 CAE926      JP Z,26E9H
2664 AF          XOR A
2665 32DC40      LD (40DCH),A
2668 E5          PUSH HL
2669 D5          PUSH DE
266A 2AF940      LD HL,(40F9H)
266D EB          EX DE,HL
266E 2AFB40      LD HL,(40FBH)
2671 DF          RST 18
2672 E1          POP HL
2673 2819        JR Z,268EH
2675 1A          LD A,(DE)
2676 6F          LD L,A
2677 BC          CP H
2678 13          INC DE
2679 200B        JR NZ,2686H
267B 1A          LD A,(DE)
267C B9          CP C
267D 2007        JR NZ,2686H
267F 13          INC DE
2680 1A          LD A,(DE)
2681 B8          CP B
2682 CACC26      JP Z,26CCH
2685 3E13        LD A,13H
2687 13          INC DE
2688 E5          PUSH HL
2689 2600        LD H,00H
268B 19          ADD HL,DE
268C 18DF        JR 266DH
268E 7C          LD A,H
268F E1          POP HL
2690 E3          EX (SP),HL
2691 F5          PUSH AF
2692 D5          PUSH DE
2693 11F124      LD DE,24F1H
2696 DF          RST 18
2697 2836        JR Z,26CFH
2699 114325      LD DE,2543H
269C DF          RST 18
269D D1          POP DE
269E 2835        JR Z,26D5H
26A0 F1          POP AF
26A1 E3          EX (SP),HL
26A2 E5          PUSH HL
26A3 C5          PUSH BC
26A4 4F          LD C,A
26A5 0600        LD B,00H
```



```
26A7 C5          PUSH BC
26A8 03          INC BC
26A9 03          INC BC
26AA 03          INC BC
26AB 2AFD40      LD HL,(40FDH)
26AE E5          PUSH HL
26AF 09          ADD HL,BC
26B0 C1          POP BC
26B1 E5          PUSH HL
26B2 CD5519      CALL 1955H
26B5 E1          POP HL
26B6 22FD40      LD (40FDH),HL
26B9 60          LD H,B
26BA 69          LD L,C
26BB 22FB40      LD (40FBH),HL
26BE 2B          DEC HL
26BF 3600        LD (HL),00H
26C1 DF          RST 18
26C2 20FA        JR NZ,26BEH
26C4 D1          POP DE
26C5 73          LD (HL),E
26C6 23          INC HL
26C7 D1          POP DE
26C8 73          LD (HL),E
26C9 23          INC HL
26CA 72          LD (HL),D
26CB EB          EX DE,HL
26CC 13          INC DE
26CD E1          POP HL
26CE C9          RET
26CF 57          LD D,A
26D0 5F          LD E,A
26D1 F1          POP AF
26D2 F1          POP AF
26D3 E3          EX (SP),HL
26D4 C9          RET
26D5 322441      LD (4124H),A
26D8 C1          POP BC
26D9 67          LD H,A
26DA 6F          LD L,A
26DB 222141      LD (4121H),HL
26DE E7          RST 20
26DF 2006        JR NZ,26E7H
26E1 212819      LD HL,1928H
26E4 222141      LD (4121H),HL
26E7 E1          POP HL
26E8 C9          RET
26E9 E5          PUSH HL
26EA 2AAE40      LD HL,(40AEH)
26ED E3          EX (SP),HL
26EE 57          LD D,A
26EF D5          PUSH DE
26F0 C5          PUSH BC
26F1 CD451E      CALL 1E45H
26F4 C1          POP BC
26F5 F1          POP AF
26F6 EB          EX DE,HL
26F7 E3          EX (SP),HL
26F8 E5          PUSH HL
26F9 EB          EX DE,HL
```

```
26FA 3C          INC A
26FB 57          LD D,A
26FC 7E          LD A,(HL)
26FD FE2C        CP 2CH
26FF 28EE        JR Z,26EFH
2701 CF          RST 8
2702 29          ADD HL,HL
2703 22F340      LD (40F3H),HL
2706 E1          POP HL
2707 22AE40      LD (40AEH),HL
270A D5          PUSH DE
270B 2AFB40      LD HL,(40FBH)
270E 3E19        LD A,19H
2710 EB          EX DE,HL
2711 2AFD40      LD HL,(40FDH)
2714 EB          EX DE,HL
2715 DF          RST 18
2716 3AAF40      LD A,(40AFH)
2719 2827        JR Z,2742H
271B BE          CP (HL)
271C 23          INC HL
271D 2008        JR NZ,2727H
271F 7E          LD A,(HL)
2720 B9          CP C
2721 23          INC HL
2722 2004        JR NZ,2728H
2724 7E          LD A,(HL)
2725 B8          CP B
2726 3E23        LD A,23H
2728 23          INC HL
2729 5E          LD E,(HL)
272A 23          INC HL
272B 56          LD D,(HL)
272C 23          INC HL
272D 20E0        JR NZ,270FH
272F 3AAE40      LD A,(40AEH)
2732 B7          OR A
2733 1E12        LD E,12H
2735 C2A219      JP NZ,19A2H
2738 F1          POP AF
2739 96          SUB (HL)
273A CA9527      JP Z,2795H
273D 1E10        LD E,10H
273F C3A219      JP 19A2H
2742 77          LD (HL),A
2743 23          INC HL
2744 5F          LD E,A
2745 1600        LD D,00H
2747 F1          POP AF
2748 71          LD (HL),C
2749 23          INC HL
274A 70          LD (HL),B
274B 23          INC HL
274C 4F          LD C,A
274D CD6319      CALL 1963H
2750 23          INC HL
2751 23          INC HL
2752 22D840      LD (40D8H),HL
2755 71          LD (HL),C
2756 23          INC HL
```



```
2757 3AAE40      LD A,(40AEH)
275A 17          RLA
275B 79          LD A,C
275C 010B00      LD BC,000BH
275F 3002        JR NC,2763H
2761 C1          POP BC
2762 03          INC BC
2763 71          LD (HL),C
2764 23          INC HL
2765 70          LD (HL),B
2766 23          INC HL
2767 F5          PUSH AF
2768 CDAA0B      CALL 0BAAH
276B F1          POP AF
276C 3D          DEC A
276D 20ED        JR NZ,275CH
276F F5          PUSH AF
2770 42          LD B,D
2771 4B          LD C,E
2772 EB          EX DE,HL
2773 19          ADD HL,DE
2774 38C7        JR C,273DH
2776 CD6C19      CALL 196CH
2779 22FD40      LD (40FDH),HL
277C 2B          DEC HL
277D 3600        LD (HL),00H
277F DF          RST 18
2780 20FA        JR NZ,277CH
2782 03          INC BC
2783 57          LD D,A
2784 2AD840      LD HL,(40D8H)
2787 5E          LD E,(HL)
2788 EB          EX DE,HL
2789 29          ADD HL,HL
278A 09          ADD HL,BC
278B EB          EX DE,HL
278C 2B          DEC HL
278D 2B          DEC HL
278E 73          LD (HL),E
278F 23          INC HL
2790 72          LD (HL),D
2791 23          INC HL
2792 F1          POP AF
2793 3830        JR C,27C5H
2795 47          LD B,A
2796 4F          LD C,A
2797 7E          LD A,(HL)
2798 23          INC HL
2799 16E1        LD D,E1H
279B 5E          LD E,(HL)
279C 23          INC HL
279D 56          LD D,(HL)
279E 23          INC HL
279F E3          EX (SP),HL
27A0 F5          PUSH AF
27A1 DF          RST 18
27A2 D23D27      JP NC,273DH
27A5 CDAA0B      CALL 0BAAH
27A8 19          ADD HL,DE
27A9 F1          POP AF
```

```
27AA 3D      DEC A
27AB 44      LD B,H
27AC 4D      LD C,L
27AD 20EB    JR NZ,279AH
27AF 3AAF40  LD A,(40AFH)
27B2 44      LD B,H
27B3 4D      LD C,L
27B4 29      ADD HL,HL
27B5 D604    SUB 04H
27B7 3804    JR C,27BDH
27B9 29      ADD HL,HL
27BA 2806    JR Z,27C2H
27BC 29      ADD HL,HL
27BD B7      OR A
27BE E2C227  JP PO,27C2H
27C1 09      ADD HL,BC
27C2 C1      POP BC
27C3 09      ADD HL,BC
27C4 EB      EX DE,HL
27C5 2AF340  LD HL,(40F3H)
27C8 C9      RET
27C9 AF      XOR A
27CA E5      PUSH HL
27CB 32AF40  LD (40AFH),A
27CE CDD427  CALL 27D4H
27D1 E1      POP HL
27D2 D7      RST 10
27D3 C9      RET
27D4 2AFD40  LD HL,(40FDH)
27D7 EB      EX DE,HL
27D8 210000  LD HL,0000H
27DB 39      ADD HL,SP
27DC E7      RST 20
27DD 200D    JR NZ,27ECH
27DF CDDA29  CALL 29DAH
27E2 CDE628  CALL 28E6H
27E5 2AA040  LD HL,(40A0H)
27E8 EB      EX DE,HL
27E9 2AD640  LD HL,(40D6H)
27EC 7D      LD A,L
27ED 93      SUB E
27EE 6F      LD L,A
27EF 7C      LD A,H
27F0 9A      SBC A,D
27F1 67      LD H,A
27F2 C3660C  JP 0C66H
27F5 3AA640  LD A,(40A6H)
27F8 6F      LD L,A
27F9 AF      XOR A
27FA 67      LD H,A
27FB C39A0A  JP 0A9AH
27FE CDA941  CALL 41A9H
2801 D7      RST 10
2802 CD2C25  CALL 252CH
2805 E5      PUSH HL
2806 219008  LD HL,0890H
2809 E5      PUSH HL
280A 3AAF40  LD A,(40AFH)
280D F5      PUSH AF
280E FE03    CP 03H
```



```
2810 CCDA29  CALL Z,29DAH
2813 F1      POP AF
2814 EB      EX DE,HL
2815 2A8E40  LD HL,(408EH)
2818 E9      JP (HL)
2819 E5      PUSH HL
281A E607    AND 07H
281C 21A118  LD HL,18A1H
281F 4F      LD C,A
2820 0600    LD B,00H
2822 09      ADD HL,BC
2823 CD8625  CALL 2586H
2826 E1      POP HL
2827 C9      RET
2828 E5      PUSH HL
2829 2AA240  LD HL,(40A2H)
282C 23      INC HL
282D 7C      LD A,H
282E B5      OR L
282F E1      POP HL
2830 C0      RET NZ
2831 1E16    LD E,16H
2833 C3A219  JP 19A2H
2836 CDBD0F  CALL 0FBDH
2839 CD6528  CALL 2865H
283C CDDA29  CALL 29DAH
283F 012B2A  LD BC,2A2BH
2842 C5      PUSH BC
2843 7E      LD A,(HL)
2844 23      INC HL
2845 E5      PUSH HL
2846 CDBF28  CALL 28BFH
2849 E1      POP HL
284A 4E      LD C,(HL)
284B 23      INC HL
284C 46      LD B,(HL)
284D CD5A28  CALL 285AH
2850 E5      PUSH HL
2851 6F      LD L,A
2852 CDCE29  CALL 29CEH
2855 D1      POP DE
2856 C9      RET
2857 CDBF28  CALL 28BFH
285A 21D340  LD HL,40D3H
285D E5      PUSH HL
285E 77      LD (HL),A
285F 23      INC HL
2860 73      LD (HL),E
2861 23      INC HL
2862 72      LD (HL),D
2863 E1      POP HL
2864 C9      RET
2865 2B      DEC HL
2866 0622    LD B,22H
2868 50      LD D,B
2869 E5      PUSH HL
286A 0EFF    LD C,FFH
286C 23      INC HL
286D 7E      LD A,(HL)
286E 0C      INC C
```

```
286F B7          OR A
2870 2806        JR Z,2878H
2872 BA          CP D
2873 2803        JR Z,2878H
2875 B8          CP B
2876 20F4        JR NZ,286CH
2878 FE22        CP 22H
287A CC781D      CALL Z,1D78H
287D E3          EX (SP),HL
287E 23          INC HL
287F EB          EX DE,HL
2880 79          LD A,C
2881 CD5A28      CALL 285AH
2884 11D340      LD DE,40D3H
2887 3ED5        LD A,D5H
2889 2AB340      LD HL,(40B3H)
288C 222141      LD (4121H),HL
288F 3E03        LD A,03H
2891 32AF40      LD (40AFH),A
2894 CDD309      CALL 09D3H
2897 11D640      LD DE,40D6H
289A DF          RST 18
289B 22B340      LD (40B3H),HL
289E E1          POP HL
289F 7E          LD A,(HL)
28A0 C0          RET NZ
28A1 1E1E        LD E,1EH
28A3 C3A219      JP 19A2H
28A6 23          INC HL
28A7 CD6528      CALL 2865H
28AA CDDA29      CALL 29DAH
28AD CDC409      CALL 09C4H
28B0 14          INC D
28B1 15          DEC D
28B2 C8          RET Z
28B3 0A          LD A,(BC)
28B4 CD2A03      CALL 032AH
28B7 FE0D        CP 0DH
28B9 CC0321      CALL Z,2103H
28BC 03          INC BC
28BD 18F2        JR 28B1H
28BF B7          OR A
28C0 0EF1        LD C,F1H
28C2 F5          PUSH AF
28C3 2AA040      LD HL,(40A0H)
28C6 EB          EX DE,HL
28C7 2AD640      LD HL,(40D6H)
28CA 2F          CPL
28CB 4F          LD C,A
28CC 06FF        LD B,FFH
28CE 09          ADD HL,BC
28CF 23          INC HL
28D0 DF          RST 18
28D1 3807        JR C,28DAH
28D3 22D640      LD (40D6H),HL
28D6 23          INC HL
28D7 EB          EX DE,HL
28D8 F1          POP AF
28D9 C9          RET
28DA F1          POP AF
```



```
28DB 1E1A        LD E,1AH
28DD CAA219      JP Z,19A2H
28E0 BF          CP A
28E1 F5          PUSH AF
28E2 01C128      LD BC,28C1H
28E5 C5          PUSH BC
28E6 2AB140      LD HL,(40B1H)
28E9 22D640      LD (40D6H),HL
28EC 210000      LD HL,0000H
28EF E5          PUSH HL
28F0 2AA040      LD HL,(40A0H)
28F3 E5          PUSH HL
28F4 21B540      LD HL,40B5H
28F7 EB          EX DE,HL
28F8 2AB340      LD HL,(40B3H)
28FB EB          EX DE,HL
28FC DF          RST 18
28FD 01F728      LD BC,28F7H
2900 C24A29      JP NZ,294AH
2903 2AF940      LD HL,(40F9H)
2906 EB          EX DE,HL
2907 2AFB40      LD HL,(40FBH)
290A EB          EX DE,HL
290B DF          RST 18
290C 2813        JR Z,2921H
290E 7E          LD A,(HL)
290F 23          INC HL
2910 23          INC HL
2911 23          INC HL
2912 FE03        CP 03H
2914 2004        JR NZ,291AH
2916 CD4B29      CALL 294BH
2919 AF          XOR A
291A 5F          LD E,A
291B 1600        LD D,00H
291D 19          ADD HL,DE
291E 18E6        JR 2906H
2920 C1          POP BC
2921 EB          EX DE,HL
2922 2AFD40      LD HL,(40FDH)
2925 EB          EX DE,HL
2926 DF          RST 18
2927 CA6B29      JP Z,296BH
292A 7E          LD A,(HL)
292B 23          INC HL
292C CDC209      CALL 09C2H
292F E5          PUSH HL
2930 09          ADD HL,BC
2931 FE03        CP 03H
2933 20EB        JR NZ,2920H
2935 22D840      LD (40D8H),HL
2938 E1          POP HL
2939 4E          LD C,(HL)
293A 0600        LD B,00H
293C 09          ADD HL,BC
293D 09          ADD HL,BC
293E 23          INC HL
293F EB          EX DE,HL
2940 2AD840      LD HL,(40D8H)
2943 EB          EX DE,HL
```

```
2944 DF          RST 18
2945 28DA        JR Z,2921H
2947 013F29      LD BC,293FH
294A C5          PUSH BC
294B AF          XOR A
294C B6          OR (HL)
294D 23          INC HL
294E 5E          LD E,(HL)
294F 23          INC HL
2950 56          LD D,(HL)
2951 23          INC HL
2952 C8          RET Z
2953 44          LD B,H
2954 4D          LD C,L
2955 2AD640      LD HL,(40D6H)
2958 DF          RST 18
2959 60          LD H,B
295A 69          LD L,C
295B D8          RET C
295C E1          POP HL
295D E3          EX (SP),HL
295E DF          RST 18
295F E3          EX (SP),HL
2960 E5          PUSH HL
2961 60          LD H,B
2962 69          LD L,C
2963 D0          RET NC
2964 C1          POP BC
2965 F1          POP AF
2966 F1          POP AF
2967 E5          PUSH HL
2968 D5          PUSH DE
2969 C5          PUSH BC
296A C9          RET
296B D1          POP DE
296C E1          POP HL
296D 7D          LD A,L
296E B4          OR H
296F C8          RET Z
2970 2B          DEC HL
2971 46          LD B,(HL)
2972 2B          DEC HL
2973 4E          LD C,(HL)
2974 E5          PUSH HL
2975 2B          DEC HL
2976 6E          LD L,(HL)
2977 2600        LD H,00H
2979 09          ADD HL,BC
297A 50          LD D,B
297B 59          LD E,C
297C 2B          DEC HL
297D 44          LD B,H
297E 4D          LD C,L
297F 2AD640      LD HL,(40D6H)
2982 CD5819      CALL 1958H
2985 E1          POP HL
2986 71          LD (HL),C
2987 23          INC HL
2988 70          LD (HL),B
2989 69          LD L,C
```



```
298A 60          LD H,B
298B 2B          DEC HL
298C C3E928      JP 28E9H
298F C5          PUSH BC
2990 E5          PUSH HL
2991 2A2141      LD HL,(4121H)
2994 E3          EX (SP),HL
2995 CD9F24      CALL 249FH
2998 E3          EX (SP),HL
2999 CDF40A      CALL 0AF4H
299C 7E          LD A,(HL)
299D E5          PUSH HL
299E 2A2141      LD HL,(4121H)
29A1 E5          PUSH HL
29A2 86          ADD A,(HL)
29A3 1E1C        LD E,1CH
29A5 DAA219      JP C,19A2H
29A8 CD5728      CALL 2857H
29AB D1          POP DE
29AC CDDE29      CALL 29DEH
29AF E3          EX (SP),HL
29B0 CDDD29      CALL 29DDH
29B3 E5          PUSH HL
29B4 2AD440      LD HL,(40D4H)
29B7 EB          EX DE,HL
29B8 CDC629      CALL 29C6H
29BB CDC629      CALL 29C6H
29BE 214923      LD HL,2349H
29C1 E3          EX (SP),HL
29C2 E5          PUSH HL
29C3 C38428      JP 2884H
29C6 E1          POP HL
29C7 E3          EX (SP),HL
29C8 7E          LD A,(HL)
29C9 23          INC HL
29CA 4E          LD C,(HL)
29CB 23          INC HL
29CC 46          LD B,(HL)
29CD 6F          LD L,A
29CE 2C          INC L
29CF 2D          DEC L
29D0 C8          RET Z
29D1 0A          LD A,(BC)
29D2 12          LD (DE),A
29D3 03          INC BC
29D4 13          INC DE
29D5 18F8        JR 29CFH
29D7 CDF40A      CALL 0AF4H
29DA 2A2141      LD HL,(4121H)
29DD EB          EX DE,HL
29DE CDF529      CALL 29F5H
29E1 EB          EX DE,HL
29E2 C0          RET NZ
29E3 D5          PUSH DE
29E4 50          LD D,B
29E5 59          LD E,C
29E6 1B          DEC DE
29E7 4E          LD C,(HL)
29E8 2AD640      LD HL,(40D6H)
29EB DF          RST 18
```

```
29EC 2005      JR NZ,29F3H
29EE 47        LD B,A
29EF 09        ADD HL,BC
29F0 22D640    LD (40D6H),HL
29F3 E1        POP HL
29F4 C9        RET
29F5 2AB340    LD HL,(40B3H)
29F8 2B        DEC HL
29F9 46        LD B,(HL)
29FA 2B        DEC HL
29FB 4E        LD C,(HL)
29FC 2B        DEC HL
29FD DF        RST 18
29FE C0        RET NZ
29FF 22B340    LD (40B3H),HL
2A02 C9        RET
2A03 01F827    LD BC,27F8H
2A06 C5        PUSH BC
2A07 CDD729    CALL 29D7H
2A0A AF        XOR A
2A0B 57        LD D,A
2A0C 7E        LD A,(HL)
2A0D B7        OR A
2A0E C9        RET
2A0F 01F827    LD BC,27F8H
2A12 C5        PUSH BC
2A13 CD072A    CALL 2A07H
2A16 CA4A1E    JP Z,1E4AH
2A19 23        INC HL
2A1A 5E        LD E,(HL)
2A1B 23        INC HL
2A1C 56        LD D,(HL)
2A1D 1A        LD A,(DE)
2A1E C9        RET
2A1F 3E01      LD A,01H
2A21 CD5728    CALL 2857H
2A24 CD1F2B    CALL 2B1FH
2A27 2AD440    LD HL,(40D4H)
2A2A 73        LD (HL),E
2A2B C1        POP BC
2A2C C38428    JP 2884H
2A2F D7        RST 10
2A30 CF        RST 8
2A31 28CD      JR Z,2A00H
2A33 1C        INC E
2A34 2B        DEC HL
2A35 D5        PUSH DE
2A36 CF        RST 8
2A37 2C        INC L
2A38 CD3723    CALL 2337H
2A3B CF        RST 8
2A3C 29        ADD HL,HL
2A3D E3        EX (SP),HL
2A3E E5        PUSH HL
2A3F E7        RST 20
2A40 2805      JR Z,2A47H
2A42 CD1F2B    CALL 2B1FH
2A45 1803      JR 2A4AH
2A47 CD132A    CALL 2A13H
2A4A D1        POP DE
```



```
2A4B F5        PUSH AF
2A4C F5        PUSH AF
2A4D 7B        LD A,E
2A4E CD5728    CALL 2857H
2A51 5F        LD E,A
2A52 F1        POP AF
2A53 1C        INC E
2A54 1D        DEC E
2A55 28D4      JR Z,2A2BH
2A57 2AD440    LD HL,(40D4H)
2A5A 77        LD (HL),A
2A5B 23        INC HL
2A5C 1D        DEC E
2A5D 20FB      JR NZ,2A5AH
2A5F 18CA      JR 2A2BH
2A61 CDDF2A    CALL 2ADFH
2A64 AF        XOR A
2A65 E3        EX (SP),HL
2A66 4F        LD C,A
2A67 3EE5      LD A,E5H
2A69 E5        PUSH HL
2A6A 7E        LD A,(HL)
2A6B B8        CP B
2A6C 3802      JR C,2A70H
2A6E 78        LD A,B
2A6F 110E00    LD DE,000EH
2A72 C5        PUSH BC
2A73 CDBF28    CALL 28BFH
2A76 C1        POP BC
2A77 E1        POP HL
2A78 E5        PUSH HL
2A79 23        INC HL
2A7A 46        LD B,(HL)
2A7B 23        INC HL
2A7C 66        LD H,(HL)
2A7D 68        LD L,B
2A7E 0600      LD B,00H
2A80 09        ADD HL,BC
2A81 44        LD B,H
2A82 4D        LD C,L
2A83 CD5A28    CALL 285AH
2A86 6F        LD L,A
2A87 CDCE29    CALL 29CEH
2A8A D1        POP DE
2A8B CDDE29    CALL 29DEH
2A8E C38428    JP 2884H
2A91 CDDF2A    CALL 2ADFH
2A94 D1        POP DE
2A95 D5        PUSH DE
2A96 1A        LD A,(DE)
2A97 90        SUB B
2A98 18CB      JR 2A65H
2A9A EB        EX DE,HL
2A9B 7E        LD A,(HL)
2A9C CDE22A    CALL 2AE2H
2A9F 04        INC B
2AA0 05        DEC B
2AA1 CA4A1E    JP Z,1E4AH
2AA4 C5        PUSH BC
2AA5 1EFF      LD E,FFH
```

```
2AA7 FE29      CP 29H
2AA9 2805      JR Z,2AB0H
2AAB CF        RST 8
2AAC 2C        INC L
2AAD CD1C2B    CALL 2B1CH
2AB0 CF        RST 8
2AB1 29        ADD HL,HL
2AB2 F1        POP AF
2AB3 E3        EX (SP),HL
2AB4 01692A    LD BC,2A69H
2AB7 C5        PUSH BC
2AB8 3D        DEC A
2AB9 BE        CP (HL)
2ABA 0600      LD B,00H
2ABC D0        RET NC
2ABD 4F        LD C,A
2ABE 7E        LD A,(HL)
2ABF 91        SUB C
2AC0 BB        CP E
2AC1 47        LD B,A
2AC2 D8        RET C
2AC3 43        LD B,E
2AC4 C9        RET
2AC5 CD072A    CALL 2A07H
2AC8 CAF827    JP Z,27F8H
2ACB 5F        LD E,A
2ACC 23        INC HL
2ACD 7E        LD A,(HL)
2ACE 23        INC HL
2ACF 66        LD H,(HL)
2AD0 6F        LD L,A
2AD1 E5        PUSH HL
2AD2 19        ADD HL,DE
2AD3 46        LD B,(HL)
2AD4 72        LD (HL),D
2AD5 E3        EX (SP),HL
2AD6 C5        PUSH BC
2AD7 7E        LD A,(HL)
2AD8 CD650E    CALL 0E65H
2ADB C1        POP BC
2ADC E1        POP HL
2ADD 70        LD (HL),B
2ADE C9        RET
2ADF EB        EX DE,HL
2AE0 CF        RST 8
2AE1 29        ADD HL,HL
2AE2 C1        POP BC
2AE3 D1        POP DE
2AE4 C5        PUSH BC
2AE5 43        LD B,E
2AE6 C9        RET
2AE7 FE7A      CP 7AH
2AE9 C29719    JP NZ,1997H
2AEC C3D941    JP 41D9H
2AEF CD1F2B    CALL 2B1FH
2AF2 329440    LD (4094H),A
2AF5 CD9340    CALL 4093H
2AF8 C3F827    JP 27F8H
2AFB CD0E2B    CALL 2B0EH
2AFE C39640    JP 4096H
```



```
2B01 D7        RST 10
2B02 CD3723    CALL 2337H
2B05 E5        PUSH HL
2B06 CD7F0A    CALL 0A7FH
2B09 EB        EX DE,HL
2B0A E1        POP HL
2B0B 7A        LD A,D
2B0C B7        OR A
2B0D C9        RET
2B0E CD1C2B    CALL 2B1CH
2B11 329440    LD (4094H),A
2B14 329740    LD (4097H),A
2B17 CF        RST 8
2B18 2C        INC L
2B19 1801      JR 2B1CH
2B1B D7        RST 10
2B1C CD3723    CALL 2337H
2B1F CD052B    CALL 2B05H
2B22 C24A1E    JP NZ,1E4AH
2B25 2B        DEC HL
2B26 D7        RST 10
2B27 7B        LD A,E
2B28 C9        RET
2B29 3E01      LD A,01H
2B2B 329C40    LD (409CH),A
2B2E C1        POP BC
2B2F CD101B    CALL 1B10H
2B32 C5        PUSH BC
2B33 21FFFF    LD HL,FFFFH
2B36 22A240    LD (40A2H),HL
2B39 E1        POP HL
2B3A D1        POP DE
2B3B 4E        LD C,(HL)
2B3C 23        INC HL
2B3D 46        LD B,(HL)
2B3E 23        INC HL
2B3F 78        LD A,B
2B40 B1        OR C
2B41 CA191A    JP Z,1A19H
2B44 CDDF41    CALL 41DFH
2B47 CD9B1D    CALL 1D9BH
2B4A C5        PUSH BC
2B4B 4E        LD C,(HL)
2B4C 23        INC HL
2B4D 46        LD B,(HL)
2B4E 23        INC HL
2B4F C5        PUSH BC
2B50 E3        EX (SP),HL
2B51 EB        EX DE,HL
2B52 DF        RST 18
2B53 C1        POP BC
2B54 DA181A    JP C,1A18H
2B57 E3        EX (SP),HL
2B58 E5        PUSH HL
2B59 C5        PUSH BC
2B5A EB        EX DE,HL
2B5B 22EC40    LD (40ECH),HL
2B5E CDAF0F    CALL 0FAFH
2B61 3E20      LD A,20H
2B63 E1        POP HL
```

```
2B64 CD2A03      CALL 032AH
2B67 CD7E2B      CALL 2B7EH
2B6A 2AA740      LD HL,(40A7H)
2B6D CD752B      CALL 2B75H
2B70 CDFE20      CALL 20FEH
2B73 18BE        JR 2B33H
2B75 7E          LD A,(HL)
2B76 B7          OR A
2B77 C8          RET Z
2B78 CD2A03      CALL 032AH
2B7B 23          INC HL
2B7C 18F7        JR 2B75H
2B7E E5          PUSH HL
2B7F 2AA740      LD HL,(40A7H)
2B82 44          LD B,H
2B83 4D          LD C,L
2B84 E1          POP HL
2B85 16FF        LD D,FFH
2B87 1803        JR 2B8CH
2B89 03          INC BC
2B8A 15          DEC D
2B8B C8          RET Z
2B8C 7E          LD A,(HL)
2B8D B7          OR A
2B8E 23          INC HL
2B8F 02          LD (BC),A
2B90 C8          RET Z
2B91 F2892B      JP P,2B89H
2B94 FEFB        CP FBH
2B96 2008        JR NZ,2BA0H
2B98 0B          DEC BC
2B99 0B          DEC BC
2B9A 0B          DEC BC
2B9B 0B          DEC BC
2B9C 14          INC D
2B9D 14          INC D
2B9E 14          INC D
2B9F 14          INC D
2BA0 FE95        CP 95H
2BA2 CC240B      CALL Z,0B24H
2BA5 D67F        SUB 7FH
2BA7 E5          PUSH HL
2BA8 5F          LD E,A
2BA9 215016      LD HL,1650H
2BAC 7E          LD A,(HL)
2BAD B7          OR A
2BAE 23          INC HL
2BAF F2AC2B      JP P,2BACH
2BB2 1D          DEC E
2BB3 20F7        JR NZ,2BACH
2BB5 E67F        AND 7FH
2BB7 02          LD (BC),A
2BB8 03          INC BC
2BB9 15          DEC D
2BBA CAD828      JP Z,28D8H
2BBD 7E          LD A,(HL)
2BBE 23          INC HL
2BBF B7          OR A
2BC0 F2B72B      JP P,2BB7H
2BC3 E1          POP HL
```



```
2BC4 18C6        JR 2B8CH
2BC6 CD101B      CALL 1B10H
2BC9 D1          POP DE
2BCA C5          PUSH BC
2BCB C5          PUSH BC
2BCC CD2C1B      CALL 1B2CH
2BCF 3005        JR NC,2BD6H
2BD1 54          LD D,H
2BD2 5D          LD E,L
2BD3 E3          EX (SP),HL
2BD4 E5          PUSH HL
2BD5 DF          RST 18
2BD6 D24A1E      JP NC,1E4AH
2BD9 212919      LD HL,1929H
2BDC CDA728      CALL 28A7H
2BDF C1          POP BC
2BE0 21E81A      LD HL,1AE8H
2BE3 E3          EX (SP),HL
2BE4 EB          EX DE,HL
2BE5 2AF940      LD HL,(40F9H)
2BE8 1A          LD A,(DE)
2BE9 02          LD (BC),A
2BEA 03          INC BC
2BEB 13          INC DE
2BEC DF          RST 18
2BED 20F9        JR NZ,2BE8H
2BEF 60          LD H,B
2BF0 69          LD L,C
2BF1 22F940      LD (40F9H),HL
2BF4 C9          RET
2BF5 CD8402      CALL 0284H
2BF8 CD3723      CALL 2337H
2BFB E5          PUSH HL
2BFC CD132A      CALL 2A13H
2BFF 3ED3        LD A,D3H
2C01 CD6402      CALL 0264H
2C04 CD6102      CALL 0261H
2C07 1A          LD A,(DE)
2C08 CD6402      CALL 0264H
2C0B 2AA440      LD HL,(40A4H)
2C0E EB          EX DE,HL
2C0F 2AF940      LD HL,(40F9H)
2C12 1A          LD A,(DE)
2C13 13          INC DE
2C14 CD6402      CALL 0264H
2C17 DF          RST 18
2C18 20F8        JR NZ,2C12H
2C1A CDF801      CALL 01F8H
2C1D E1          POP HL
2C1E C9          RET
2C1F CD9302      CALL 0293H
2C22 7E          LD A,(HL)
2C23 D6B2        SUB B2H
2C25 2802        JR Z,2C29H
2C27 AF          XOR A
2C28 012F23      LD BC,232FH
2C2B F5          PUSH AF
2C2C 2B          DEC HL
2C2D D7          RST 10
2C2E 3E00        LD A,00H
```

```
2C30 2807        JR Z,2C39H
2C32 CD3723      CALL 2337H
2C35 CD132A      CALL 2A13H
2C38 1A          LD A,(DE)
2C39 6F          LD L,A
2C3A F1          POP AF
2C3B B7          OR A
2C3C 67          LD H,A
2C3D 222141      LD (4121H),HL
2C40 CC4D1B      CALL Z,1B4DH
2C43 2A2141      LD HL,(4121H)
2C46 EB          EX DE,HL
2C47 0603        LD B,03H
2C49 CD3502      CALL 0235H
2C4C D6D3        SUB D3H
2C4E 20F7        JR NZ,2C47H
2C50 10F7        DJNZ 2C49H
2C52 CD3502      CALL 0235H
2C55 1C          INC E
2C56 1D          DEC E
2C57 2803        JR Z,2C5CH
2C59 BB          CP E
2C5A 2037        JR NZ,2C93H
2C5C 2AA440      LD HL,(40A4H)
2C5F 0603        LD B,03H
2C61 CD3502      CALL 0235H
2C64 5F          LD E,A
2C65 96          SUB (HL)
2C66 A2          AND D
2C67 2021        JR NZ,2C8AH
2C69 73          LD (HL),E
2C6A CD6C19      CALL 196CH
2C6D 7E          LD A,(HL)
2C6E B7          OR A
2C6F 23          INC HL
2C70 20ED        JR NZ,2C5FH
2C72 CD2C02      CALL 022CH
2C75 10EA        DJNZ 2C61H
2C77 22F940      LD (40F9H),HL
2C7A 212919      LD HL,1929H
2C7D CDA728      CALL 28A7H
2C80 CDF801      CALL 01F8H
2C83 2AA440      LD HL,(40A4H)
2C86 E5          PUSH HL
2C87 C3E81A      JP 1AE8H
2C8A 21A52C      LD HL,2CA5H
2C8D CDA728      CALL 28A7H
2C90 C3181A      JP 1A18H
2C93 323E3C      LD (3C3EH),A
2C96 0603        LD B,03H
2C98 CD3502      CALL 0235H
2C9B B7          OR A
2C9C 20F8        JR NZ,2C96H
2C9E 10F8        DJNZ 2C98H
2CA0 CD9602      CALL 0296H
2CA3 18A2        JR 2C47H
```

2C30-2CA3 ROM **119**

```
2CA5  42         B    B
```

2C30-2CA5 ROM **119**

```
2CA6  41         A    A
2CA7  44         D    D
2CA8  0D         .    .
2CA9  00         .    .
```

2CA6-2CA9 ROM **120**

```
2CAA CD7F0A      CALL 0A7FH
2CAD 7E          LD A,(HL)
2CAE C3F827      JP 27F8H
2CB1 CD022B      CALL 2B02H
2CB4 D5          PUSH DE
2CB5 CF          RST 8
2CB6 2C          INC L
2CB7 CD1C2B      CALL 2B1CH
2CBA D1          POP DE
2CBB 12          LD (DE),A
2CBC C9          RET
2CBD CD3823      CALL 2338H
2CC0 CDF40A      CALL 0AF4H
2CC3 CF          RST 8
2CC4 3B          DEC SP
2CC5 EB          EX DE,HL
2CC6 2A2141      LD HL,(4121H)
2CC9 1808        JR 2CD3H
2CCB 3ADE40      LD A,(40DEH)
2CCE B7          OR A
2CCF 280C        JR Z,2CDDH
2CD1 D1          POP DE
2CD2 EB          EX DE,HL
2CD3 E5          PUSH HL
2CD4 AF          XOR A
2CD5 32DE40      LD (40DEH),A
2CD8 BA          CP D
2CD9 F5          PUSH AF
2CDA D5          PUSH DE
2CDB 46          LD B,(HL)
2CDC B0          OR B
2CDD CA4A1E      JP Z,1E4AH
2CE0 23          INC HL
2CE1 4E          LD C,(HL)
2CE2 23          INC HL
2CE3 66          LD H,(HL)
2CE4 69          LD L,C
2CE5 181C        JR 2D03H
2CE7 58          LD E,B
2CE8 E5          PUSH HL
2CE9 0E02        LD C,02H
2CEB 7E          LD A,(HL)
2CEC 23          INC HL
2CED FE25        CP 25H
2CEF CA172E      JP Z,2E17H
2CF2 FE20        CP 20H
2CF4 2003        JR NZ,2CF9H
2CF6 0C          INC C
2CF7 10F2        DJNZ 2CEBH
2CF9 E1          POP HL
2CFA 43          LD B,E
2CFB 3E25        LD A,25H
2CFD CD492E      CALL 2E49H
```

2CA6-2CFD ROM **120**

```
2D00 CD2A03      CALL 032AH
2D03 AF          XOR A
2D04 5F          LD E,A
2D05 57          LD D,A
2D06 CD492E      CALL 2E49H
2D09 57          LD D,A
2D0A 7E          LD A,(HL)
2D0B 23          INC HL
2D0C FE21        CP 21H
2D0E CA142E      JP Z,2E14H
2D11 FE23        CP 23H
2D13 2837        JR Z,2D4CH
2D15 05          DEC B
2D16 CAFE2D      JP Z,2DFEH
2D19 FE2B        CP 2BH
2D1B 3E08        LD A,08H
2D1D 28E7        JR Z,2D06H
2D1F 2B          DEC HL
2D20 7E          LD A,(HL)
2D21 23          INC HL
2D22 FE2E        CP 2EH
2D24 2840        JR Z,2D66H
2D26 FE25        CP 25H
2D28 28BD        JR Z,2CE7H
2D2A BE          CP (HL)
2D2B 20D0        JR NZ,2CFDH
2D2D FE24        CP 24H
2D2F 2814        JR Z,2D45H
2D31 FE2A        CP 2AH
2D33 20C8        JR NZ,2CFDH
2D35 78          LD A,B
2D36 FE02        CP 02H
2D38 23          INC HL
2D39 3803        JR C,2D3EH
2D3B 7E          LD A,(HL)
2D3C FE24        CP 24H
2D3E 3E20        LD A,20H
2D40 2007        JR NZ,2D49H
2D42 05          DEC B
2D43 1C          INC E
2D44 FEAF        CP AFH
2D46 C610        ADD A,10H
2D48 23          INC HL
2D49 1C          INC E
2D4A 82          ADD A,D
2D4B 57          LD D,A
2D4C 1C          INC E
2D4D 0E00        LD C,00H
2D4F 05          DEC B
2D50 2847        JR Z,2D99H
2D52 7E          LD A,(HL)
2D53 23          INC HL
2D54 FE2E        CP 2EH
2D56 2818        JR Z,2D70H
2D58 FE23        CP 23H
2D5A 28F0        JR Z,2D4CH
2D5C FE2C        CP 2CH
2D5E 201A        JR NZ,2D7AH
2D60 7A          LD A,D
2D61 F640        OR 40H
```

2D00-2D61 ROM 121

```
2D63 57          LD D,A
2D64 18E6        JR 2D4CH
2D66 7E          LD A,(HL)
2D67 FE23        CP 23H
2D69 3E2E        LD A,2EH
2D6B 2090        JR NZ,2CFDH
2D6D 0E01        LD C,01H
2D6F 23          INC HL
2D70 0C          INC C
2D71 05          DEC B
2D72 2825        JR Z,2D99H
2D74 7E          LD A,(HL)
2D75 23          INC HL
2D76 FE23        CP 23H
2D78 28F6        JR Z,2D70H
2D7A D5          PUSH DE
2D7B 11972D      LD DE,2D97H
2D7E D5          PUSH DE
2D7F 54          LD D,H
2D80 5D          LD E,L
2D81 FE5B        CP 5BH
2D83 C0          RET NZ
2D84 BE          CP (HL)
2D85 C0          RET NZ
2D86 23          INC HL
2D87 BE          CP (HL)
2D88 C0          RET NZ
2D89 23          INC HL
2D8A BE          CP (HL)
2D8B C0          RET NZ
2D8C 23          INC HL
2D8D 78          LD A,B
2D8E D604        SUB 04H
2D90 D8          RET C
2D91 D1          POP DE
2D92 D1          POP DE
2D93 47          LD B,A
2D94 14          INC D
2D95 23          INC HL
2D96 CAEBD1      JP Z,D1EBH
2D99 7A          LD A,D
2D9A 2B          DEC HL
2D9B 1C          INC E
2D9C E608        AND 08H
2D9E 2015        JR NZ,2DB5H
2DA0 1D          DEC E
2DA1 78          LD A,B
2DA2 B7          OR A
2DA3 2810        JR Z,2DB5H
2DA5 7E          LD A,(HL)
2DA6 D62D        SUB 2DH
2DA8 2806        JR Z,2DB0H
2DAA FEFE        CP FEH
2DAC 2007        JR NZ,2DB5H
2DAE 3E08        LD A,08H
2DB0 C604        ADD A,04H
2DB2 82          ADD A,D
2DB3 57          LD D,A
2DB4 05          DEC B
2DB5 E1          POP HL
```

2D63-2DB5 ROM 122

```
2DB6 F1          POP AF
2DB7 2850        JR Z,2E09H
2DB9 C5          PUSH BC
2DBA D5          PUSH DE
2DBB CD3723      CALL 2337H
2DBE D1          POP DE
2DBF C1          POP BC
2DC0 C5          PUSH BC
2DC1 E5          PUSH HL
2DC2 43          LD B,E
2DC3 78          LD A,B
2DC4 81          ADD A,C
2DC5 FE19        CP 19H
2DC7 D24A1E      JP NC,1E4AH
2DCA 7A          LD A,D
2DCB F680        OR 80H
2DCD CDBE0F      CALL 0FBEH
2DD0 CDA728      CALL 28A7H
2DD3 E1          POP HL
2DD4 2B          DEC HL
2DD5 D7          RST 10
2DD6 37          SCF
2DD7 280D        JR Z,2DE6H
2DD9 32DE40      LD (40DEH),A
2DDC FE3B        CP 3BH
2DDE 2805        JR Z,2DE5H
2DE0 FE2C        CP 2CH
2DE2 C29719      JP NZ,1997H
2DE5 D7          RST 10
2DE6 C1          POP BC
2DE7 EB          EX DE,HL
2DE8 E1          POP HL
2DE9 E5          PUSH HL
2DEA F5          PUSH AF
2DEB D5          PUSH DE
2DEC 7E          LD A,(HL)
2DED 90          SUB B
2DEE 23          INC HL
2DEF 4E          LD C,(HL)
2DF0 23          INC HL
2DF1 66          LD H,(HL)
2DF2 69          LD L,C
2DF3 1600        LD D,00H
2DF5 5F          LD E,A
2DF6 19          ADD HL,DE
2DF7 78          LD A,B
2DF8 B7          OR A
2DF9 C2032D      JP NZ,2D03H
2DFC 1806        JR 2E04H
2DFE CD492E      CALL 2E49H
2E01 CD2A03      CALL 032AH
2E04 E1          POP HL
2E05 F1          POP AF
2E06 C2CB2C      JP NZ,2CCBH
2E09 DCFE20      CALL C,20FEH
2E0C E3          EX (SP),HL
2E0D CDDD29      CALL 29DDH
2E10 E1          POP HL
2E11 C36921      JP 2169H
2E14 0E01        LD C,01H
```



```
2E16 3EF1        LD A,F1H
2E18 05          DEC B
2E19 CD492E      CALL 2E49H
2E1C E1          POP HL
2E1D F1          POP AF
2E1E 28E9        JR Z,2E09H
2E20 C5          PUSH BC
2E21 CD3723      CALL 2337H
2E24 CDF40A      CALL 0AF4H
2E27 C1          POP BC
2E28 C5          PUSH BC
2E29 E5          PUSH HL
2E2A 2A2141      LD HL,(4121H)
2E2D 41          LD B,C
2E2E 0E00        LD C,00H
2E30 C5          PUSH BC
2E31 CD682A      CALL 2A68H
2E34 CDAA28      CALL 28AAH
2E37 2A2141      LD HL,(4121H)
2E3A F1          POP AF
2E3B 96          SUB (HL)
2E3C 47          LD B,A
2E3D 3E20        LD A,20H
2E3F 04          INC B
2E40 05          DEC B
2E41 CAD32D      JP Z,2DD3H
2E44 CD2A03      CALL 032AH
2E47 18F7        JR 2E40H
2E49 F5          PUSH AF
2E4A 7A          LD A,D
2E4B B7          OR A
2E4C 3E2B        LD A,2BH
2E4E C42A03      CALL NZ,032AH
2E51 F1          POP AF
2E52 C9          RET
2E53 329A40      LD (409AH),A
2E56 2AEA40      LD HL,(40EAH)
2E59 B4          OR H
2E5A A5          AND L
2E5B 3C          INC A
2E5C EB          EX DE,HL
2E5D C8          RET Z
2E5E 1804        JR 2E64H
2E60 CD4F1E      CALL 1E4FH
2E63 C0          RET NZ
2E64 E1          POP HL
2E65 EB          EX DE,HL
2E66 22EC40      LD (40ECH),HL
2E69 EB          EX DE,HL
2E6A CD2C1B      CALL 1B2CH
2E6D D2D91E      JP NC,1ED9H
2E70 60          LD H,B
2E71 69          LD L,C
2E72 23          INC HL
2E73 23          INC HL
2E74 4E          LD C,(HL)
2E75 23          INC HL
2E76 46          LD B,(HL)
2E77 23          INC HL
2E78 C5          PUSH BC
```

```
2E79 CD7E2B      CALL 2B7EH
2E7C E1          POP HL
2E7D E5          PUSH HL
2E7E CDAF0F      CALL 0FAFH
2E81 3E20        LD A,20H
2E83 CD2A03      CALL 032AH
2E86 2AA740      LD HL,(40A7H)
2E89 3E0E        LD A,0EH
2E8B CD2A03      CALL 032AH
2E8E E5          PUSH HL
2E8F 0EFF        LD C,FFH
2E91 0C          INC C
2E92 7E          LD A,(HL)
2E93 B7          OR A
2E94 23          INC HL
2E95 20FA        JR NZ,2E91H
2E97 E1          POP HL
2E98 47          LD B,A
2E99 1600        LD D,00H
2E9B CD8403      CALL 0384H
2E9E D630        SUB 30H
2EA0 380E        JR C,2EB0H
2EA2 FE0A        CP 0AH
2EA4 300A        JR NC,2EB0H
2EA6 5F          LD E,A
2EA7 7A          LD A,D
2EA8 07          RLCA
2EA9 07          RLCA
2EAA 82          ADD A,D
2EAB 07          RLCA
2EAC 83          ADD A,E
2EAD 57          LD D,A
2EAE 18EB        JR 2E9BH
2EB0 E5          PUSH HL
2EB1 21992E      LD HL,2E99H
2EB4 E3          EX (SP),HL
2EB5 15          DEC D
2EB6 14          INC D
2EB7 C2BB2E      JP NZ,2EBBH
2EBA 14          INC D
2EBB FED8        CP D8H
2EBD CAD22F      JP Z,2FD2H
2EC0 FEDD        CP DDH
2EC2 CAF02F      JP Z,2FE0H
2EC5 FEF0        CP F0H
2EC7 2841        JR Z,2F0AH
2EC9 FE31        CP 31H
2ECB 3802        JR C,2ECFH
2ECD D620        SUB 20H
2ECF FE21        CP 21H
2ED1 CAF62F      JP Z,2FF6H
2ED4 FE1C        CP 1CH
2ED6 CA402F      JP Z,2F40H
2ED9 FE23        CP 23H
2EDB 283F        JR Z,2F1CH
2EDD FE19        CP 19H
2EDF CA7D2F      JP Z,2F7DH
2EE2 FE14        CP 14H
2EE4 CA4A2F      JP Z,2F4AH
2EE7 FE13        CP 13H
```



```
2EE9 CA652F      JP Z,2F65H
2EEC FE15        CP 15H
2EEE CAE32F      JP Z,2FE3H
2EF1 FE28        CP 28H
2EF3 CA782F      JP Z,2F78H
2EF6 FE1B        CP 1BH
2EF8 281C        JR Z,2F16H
2EFA FE18        CP 18H
2EFC CA752F      JP Z,2F75H
2EFF FE11        CP 11H
2F01 C0          RET NZ
2F02 C1          POP BC
2F03 D1          POP DE
2F04 CDFE20      CALL 20FEH
2F07 C3652E      JP 2E65H
2F0A 7E          LD A,(HL)
2F0B B7          OR A
2F0C C8          RET Z
2F0D 04          INC B
2F0E CD2A03      CALL 032AH
2F11 23          INC HL
2F12 15          DEC D
2F13 20F5        JR NZ,2F0AH
2F15 C9          RET
2F16 E5          PUSH HL
2F17 215F2F      LD HL,2F5FH
2F1A E3          EX (SP),HL
2F1B 37          SCF
2F1C F5          PUSH AF
2F1D CD8403      CALL 0384H
2F20 5F          LD E,A
2F21 F1          POP AF
2F22 F5          PUSH AF
2F23 DC5F2F      CALL C,2F5FH
2F26 7E          LD A,(HL)
2F27 B7          OR A
2F28 CA3E2F      JP Z,2F3EH
2F2B CD2A03      CALL 032AH
2F2E F1          POP AF
2F2F F5          PUSH AF
2F30 DCA12F      CALL C,2FA1H
2F33 3802        JR C,2F37H
2F35 23          INC HL
2F36 04          INC B
2F37 7E          LD A,(HL)
2F38 BB          CP E
2F39 20EB        JR NZ,2F26H
2F3B 15          DEC D
2F3C 20E8        JR NZ,2F26H
2F3E F1          POP AF
2F3F C9          RET
2F40 CD752B      CALL 2B75H
2F43 CDFE20      CALL 20FEH
2F46 C1          POP BC
2F47 C37C2E      JP 2E7CH
2F4A 7E          LD A,(HL)
2F4B B7          OR A
2F4C C8          RET Z
2F4D 3E21        LD A,21H
2F4F CD2A03      CALL 032AH
```

```
2F52 7E        LD A,(HL)
2F53 B7        OR A
2F54 2809      JR Z,2F5FH
2F56 CD2A03    CALL 032AH
2F59 CDA12F    CALL 2FA1H
2F5C 15        DEC D
2F5D 20F3      JR NZ,2F52H
2F5F 3E21      LD A,21H
2F61 CD2A03    CALL 032AH
2F64 C9        RET
2F65 7E        LD A,(HL)
2F66 B7        OR A
2F67 C8        RET Z
2F68 CD8403    CALL 0384H
2F6B 77        LD (HL),A
2F6C CD2A03    CALL 032AH
2F6F 23        INC HL
2F70 04        INC B
2F71 15        DEC D
2F72 20F1      JR NZ,2F65H
2F74 C9        RET
2F75 3600      LD (HL),00H
2F77 48        LD C,B
2F78 16FF      LD D,FFH
2F7A CD0A2F    CALL 2F0AH
2F7D CD8403    CALL 0384H
2F80 B7        OR A
2F81 CA7D2F    JP Z,2F7DH
2F84 FE08      CP 08H
2F86 280A      JR Z,2F92H
2F88 FE0D      CP 0DH
2F8A CAE02F    JP Z,2FE0H
2F8D FE1B      CP 1BH
2F8F C8        RET Z
2F90 201E      JR NZ,2FB0H
2F92 3E08      LD A,08H
2F94 05        DEC B
2F95 04        INC B
2F96 281F      JR Z,2FB7H
2F98 CD2A03    CALL 032AH
2F9B 2B        DEC HL
2F9C 05        DEC B
2F9D 117D2F    LD DE,2F7DH
2FA0 D5        PUSH DE
2FA1 E5        PUSH HL
2FA2 0D        DEC C
2FA3 7E        LD A,(HL)
2FA4 B7        OR A
2FA5 37        SCF
2FA6 CA9008    JP Z,0890H
2FA9 23        INC HL
2FAA 7E        LD A,(HL)
2FAB 2B        DEC HL
2FAC 77        LD (HL),A
2FAD 23        INC HL
2FAE 18F3      JR 2FA3H
2FB0 F5        PUSH AF
2FB1 79        LD A,C
2FB2 FEFF      CP FFH
2FB4 3803      JR C,2FB9H
```

2F52-2FB4 ROM 127

```
2FB6 F1        POP AF
2FB7 18C4      JR 2F7DH
2FB9 90        SUB B
2FBA 0C        INC C
2FBB 04        INC B
2FBC C5        PUSH BC
2FBD EB        EX DE,HL
2FBE 6F        LD L,A
2FBF 2600      LD H,00H
2FC1 19        ADD HL,DE
2FC2 44        LD B,H
2FC3 4D        LD C,L
2FC4 23        INC HL
2FC5 CD5819    CALL 1958H
2FC8 C1        POP BC
2FC9 F1        POP AF
2FCA 77        LD (HL),A
2FCB CD2A03    CALL 032AH
2FCE 23        INC HL
2FCF C37D2F    JP 2F7DH
2FD2 78        LD A,B
2FD3 B7        OR A
2FD4 C8        RET Z
2FD5 05        DEC B
2FD6 2B        DEC HL
2FD7 3E08      LD A,08H
2FD9 CD2A03    CALL 032AH
2FDC 15        DEC D
2FDD 20F3      JR NZ,2FD2H
2FDF C9        RET
2FE0 CD752B    CALL 2B75H
2FE3 CDFE20    CALL 20FEH
2FE6 C1        POP BC
2FE7 D1        POP DE
2FE8 7A        LD A,D
2FE9 A3        AND E
2FEA 3C        INC A
2FEB 2AA740    LD HL,(40A7H)
2FEE 2B        DEC HL
2FEF C8        RET Z
2FF0 37        SCF
2FF1 23        INC HL
2FF2 F5        PUSH AF
2FF3 C3981A    JP 1A98H
2FF6 C1        POP BC
2FF7 D1        POP DE
2FF8 C3191A    JP 1A19H
```

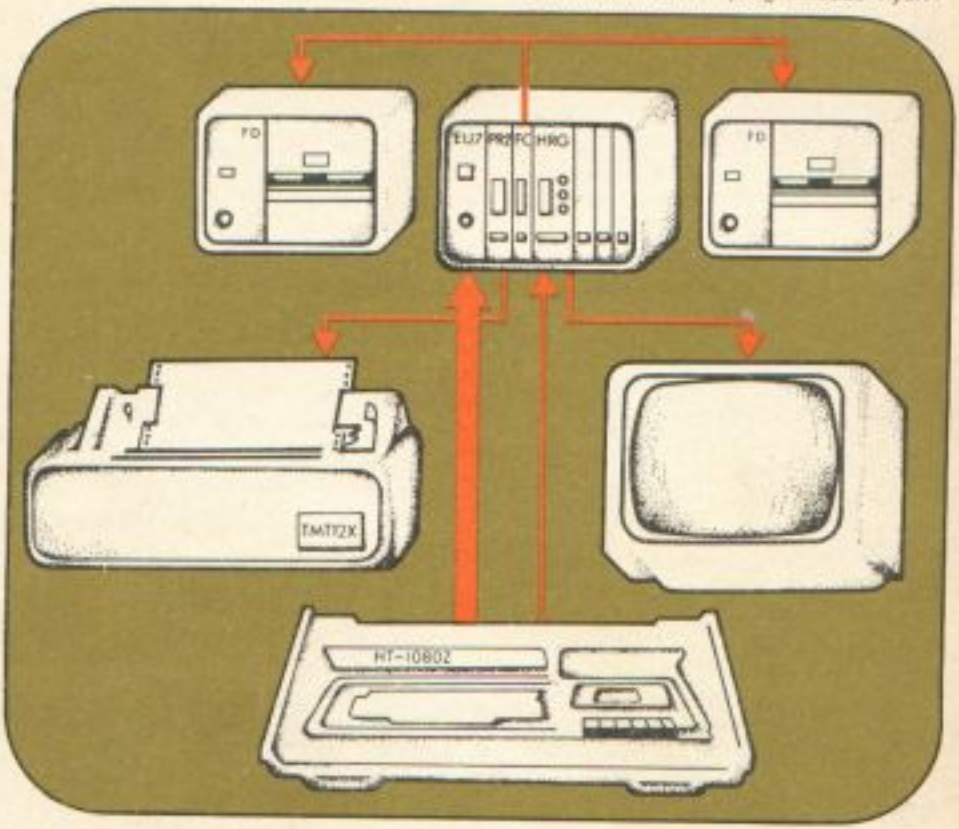
2FB6-2FF8 ROM 128

```
2FFB  DE   .   ^
2FFC  C3   .   C
2FFD  C3   .   C
2FFE  44   D   D
2FFF  B2   .   2
```

2FB6-2FFF ROM 128

HT—1080Z/64 School Computer

Az alapgép már 64 kbyte memóriakapacitású! BASIC és FORTH programozási nyelv!



Nyomtató illesztő

Minden ún. Centronics típusú nyomtatóhoz alkalmazható (pl. Epson RX-80, Seikosha GP80 GP100, TERTA TMT 12X, MOM K 6311, stb.)

HT–1080Z/PR1 kábelre szerelt változat
HT–1080Z/PR2 bővítő egységbe tolható kártya

Bővítő egység HT–1080Z/EU7

Különféle illesztő és bővítő kártyák (max. 7 db) fogadására alkalmas doboz. Busz illesztő és tápegységet tartalmaz.

Floppy illesztő HT–1080Z/F1

Illesztő egység hajlékony mágneslemezes tárolóhoz (4 db MOM 1800/900 típusú drive meghajtásra alkalmas.) Bővítő egységbe tolható kártya.

Nagybontású grafika HT–1080Z/HRG

512×512 felbontású grafikus megjelenítést tesz lehetővé. Minden képpont külön címezhető. Képpontonként négyféle fényesség állítható be. Nagyítási lehetőség! BASIC-ből programozható.

Igényesebb alkalmazásokhoz kifejlesztettük a HT 2080Z/64 személyi számítógépet. Moduláris konstrukciója lehetővé teszi, hogy a bővítő egység üres kártyahelyein alhelyezhetők speciális igényeket kielégítő illesztőegységek (pl. ügyviteli célú rendszerek, mérés és adatgyűjtés — ügyviteli adatfeldolgozás).

Szives érdeklődésére készséggel áll rendelkezésre a Híradástechnika Szövetkezet Számítástechnikai Vevőszolgálata 1116 Budapest XI., Temesvár u. 20. Telefon: 869-522/130

– Ft

HÍRADÁSTECHNIKA SZÖVETKEZET
1519 Budapest, Pf. 268