Kaszanyiczki László

A VIDEOTON TV-COMPUTER ROM LISTÁJA

TUDOMÁNYSZERVEZÉSI ÉS INFORMATIKAI INTÉZET
Budapest, 1988

Kaszanyiczki László
A VIDEOTON TV-COMPUTER ROM LISTÁJA

Kaszanyiczki László

A VIDEOTON TV-COMPUTER ROM LISTÁJA

TUDOMÁNYSZERVEZÉSI ÉS INFORMATIKAI INTÉZET
Budapest, 1988

Készült a Tudományszervezési és Informatikai Intézet felkérésére

Lektorálták: Gombás Tibor
Dr. Varga András

Engedélyszám: 59330
Felelős kiadó:
Páris György, a Tudományszervezési
és Informatikai Intézet igazgatója
Terjedelem: 30,03 A/5 ív
Példányszám: 3000
880901—Kossuth Sokszorosító

TARTALOMJEGYZEK

RAM rutinok és táblázatok..... 7
ROM lista..... 19
I. SYSTEM ROM..... 19
II. EXTENSION ROM..... 245
ROM rutinok..... 303
A rendszer ROM fö részének rutinjai..... 303
A rendszer ROM bövítés..... 315
ROM táblázatok..... 319
A ROM fö részének táblázatai..... 319
A ROM bövítés táblázatai..... 323
Kiegészítés,magyarázat a ROM rutinokhoz..... 325
RST 30H rutin..... 325
Hibakezelö rutinok..... 327
Megszakításkezelés..... 329
"Hibák" a ROM listában..... 331
BASIC utasítások és függvények közvetlen hívása... 333
Ami a számítógéphez adott programozási segédletböl kimaradt..... 335

Tisztelt Olvasó!

Ha ezt a könyvet a kezébe vette, akkor önnek valószínüleg rendelkezésre áll otthon (az iskolában, a munkahelyen) egy Videoton TV-Computer és a róla szóló nem túl nagyszámú kiadvány. Ezekből elsajátíthatta a gép BASIC-jét, mely némiképp különbözik más elterjedt géptipus (C-64, C+4, ZX-Spectrum) BASIC tájszólásától. Megismerhette az interpreter jó és rossz oldalát, használatának korlátait. Előbb, vagy utóbb szükek lesznek ezek a korlátok, a megoldandó feladatok, a kíváncsiság átlépni kényszerít rajtuk. Ilyenkor szokás az Assembler-hez fordulni segítségül.

Szerencsére jobbnál jobb Z-80 Assembler tankönyvek közül választhatunk. Az is jó dolog, hogy ma már többféle Assembler, Monitor, Debugger program is beszerezhető erre a géptipusra is. Néhány hónapja megjelent az operációs rendszer tömör leírása. Ez a kiadvány nélkülözhetetlen ahhoz, hogy teljesebben kihasználhatók legyenek a TV-Computer adta lehetöségek.

Nem tudom ki hogy van ezzel, de ha egy új tipusú számítógép kerül a kezembe, elöször azt nézem meg, hogy mit tud. Ez rendszerint a gép BASIC-jének kivesézését jelenti. Ehhez a feladathoz elegendö a géppel adott leírás és néhány nap nehéz fizikai munkának számító billentyüzet gyötrés.

Másodikként azt igyekszem kideríteni, hogy a gép mit, hol, hogyan tárol a RAM-ban. Ez már bonyolultabb feladat, ha nem áll a rendelkezésünkre a gépen és a fejünkön kívül semmi segédeszköz. Szerencsére a fontosabb rendszerváltozók és funkciójuk több létezö kiadványban is fellelhetök (BASIC programozási segédlet, Operációs rendszer cimü könyvecskék).

Végül rendszerint mindenki eljut a kíváncsiságnak arra a fokára, amikor már azt is szeretné tudni, hogy amit tud a számítógépe azt hogyan csinálja, mi van a gép tudását meghatározó ROM-ban, hogy müködik a BASIC interpreter, az operációs rendszer rutinjai.

Többnyire nem csak a meddö tudásvágy hajt ilyenkor bennünket. Valószínübb, hogy olyan feladatot szeretnénk a géppel megoldani, melyet sem a BASIC, sem az operációs rendszer rutinjainak közvetlen hívása nem támogat eléggé. Legrosszabb esetben magunk írjuk meg a szükséges rutint assemblerben. Hogy ne kelljen mindig a "legrosszabb" megoldást választani, többek között ezért született ez a könyv.

A könyv tartalmazza az BASIC 1.2 verzióval rendelkezö gépek teljes alap ROM-jának az assembler nyelvü listáját. A ROM disas-

semblálása az A-D-Teszter nevü programmal történt. A lista a ROM táblázatait is tartalmazza. Minden fontosabb, elkülöníthetö rutin, szubrutin kezdetét, belépését feltüntettem a listában. A listában szereplö megjegyzések alapján könnyen nyomon követhetjük a rutinok futását.

Külön táblázatban összefoglaltam a fontosabb ROM-rutinok belépési címeit és rövid leírását. Néhány, az interpreter müködése szempontjából fontos rutin müködését részletesebben is ismertetem.

A ROM-listában és a táblázatokban a számok rendszerint hexadecimális számok. Ahol decimális számokat használok, ott az a környezetböl vagy egyértelmüen kiderül, hogy annak decimálisnak kell lenni, vagy ezt külön kihangsúlyozom. Az Assembler mnemonikok megegyeznek a Sztrókay: A Z-80 Assembler ... címü könyvben alkalmazott mnemonikokkal.

RAM RUTINOK ÉS TÁBLÁZATOK

A 0. lapon található RAM

A könyv a TVC ROM-járól akar szólni, mégis a RAM ismertetésével kezdem. Ennek legfőbb oka az, hogy az itt használt változó neveket a ROM rutinok magyarázatához felhasználom, ennyiből ez a fejezet jelölés magyarázatnak is tekinthető. Részletesebben ismertetem a RAM ezen byte-jainak mibenlétét, mint az eddig megjelent kiadványok.

Az itt közölt lista jórészt a ROM-ból kerül ide hideg-RESET után. Ezek talán a legfontosabb rutinjai a rendszernek: hibakezelő rutinok, az operációs rendszer funkcióhívási rutinjai, a megszakításkezelő rutin. Azért is fontos és hasznos, hogy ezek a RAM-ban találhatók, mert így "el lehet téríteni" ezeket a rutinokat, azaz a saját rutinjainkat beilleszthetjük a rendszerbe, úgy módosíthatjuk ezeket a rutinokat, ahogy az nekünk megfelel.

```
0000  DB   00 00 00
0003  DB   F0              ; a memórialapozást meghatározó byte
0004  DB   00 00 00 00

0008  POP  HL              ; A Basic interpreter hibakezelő ru-
0009  LD   A,(HL)          ; tinja. A RST 08H utasítást követő
000A  OR   A               ; byte tartalmazza a hiba kódját.
000B  PUSH HL
000C  NOP
000D  NOP
000E  NOP
000F  NOP
0010  RET  Z               ; Az RST 10H hibakezelő rutin. Az A
0011  POP  HL              ; regiszter tartalmazza a hiba kód-
0012  JP   FD5C            ; ját.
0015  NOP
0016  NOP
0017  NOP

0018  JP   FB82            ; Az RST 18H aritmetikai rutin.

001B  LD   (001F),A        ; Az operációs rendszer rutinjainak
001E  RST  30H             ; hívása. Az A regiszter tartalmazza
001F  DB   00              ; a rutin számát.
```

```
0020  RET

0021  DW   0000          : Az EXT 0 rutin címe.
0023  DW   0000          : Az EXT 1 rutin címe.
0025  DW   0000          : Az EXT 2 rutin címe.
0027  DW   0000          : Az EXT 3 rutin címe.
0029  DW   0000          : Az EXT 4 rutin címe.
002B  DW   0000          : Az EXT 5 rutin címe.
002D  DW   0000          : Az EXT 6 rutin címe.

002F  DB   00

0030  JP   0B23          : Az operációs rendszer rutinjai
0033  NOP                ; hívásának a belépési pontja.
0034  NOP
0035  NOP
0036  NOP
0037  NOP

0038  PUSH AF            : A megszakításkezelő rutin belépési
0039  LD   70            ; pontja.
003B  OUT  02,A
003D  JP   C412

A 0. csatoló azonosító:
0040  DB   00 00 00 00 00 00 00 00

A 0. csatoló RAM terület:
0048  DB   00 00 00 00 00 00 00 00
0050  DB   00 00 00 00 00 00 00 00
0058  DB   00 00 00 00 00 00 00 00
0060  DB   00 00 00 00 00 00 00 00
0068  DB   00 00 00 00 00 00 00 00

Az 1. csatoló azonosító:
0070  DB   00 00 00 00 00 00 00 00

Az 1. csatoló RAM terület:
0078  DB   00 00 00 00 00 00 00 00
0080  DB   00 00 00 00 00 00 00 00
0088  DB   00 00 00 00 00 00 00 00
0090  DB   00 00 00 00 00 00 00 00
0098  DB   00 00 00 00 00 00 00 00
```

A 2. csatoló azonosító:

```
00A0  DB   00 00 00 00 00 00 00 00
```

A 2. csatoló RAM terület:

```
00A8  DB   00 00 00 00 00 00 00 00
00B0  DB   00 00 00 00 00 00 00 00
00B8  DB   00 00 00 00 00 00 00 00
00C0  DB   00 00 00 00 00 00 00 00
00C8  DB   00 00 00 00 00 00 00 00
```

A 3. csatoló azonosító:

```
00D0  DB   00 00 00 00 00 00 00 00
```

A 3. csatoló RAM terület:

```
00D8  DB   00 00 00 00 00 00 00 00
00E0  DB   00 00 00 00 00 00 00 00
00E8  DB   00 00 00 00 00 00 00 00
00F0  DB   00 00 00 00 00 00 00 00
00F8  DB   00 00 00 00 00 00 00 00
```

A 0100H-06FFH RAM terület az ASCII képernyö.

A 0700H-073FH RAM terület a 25. képernyösor helye az ASCII képernyön.

A 0740H-0AFFH RAM terület a definiálható és újra definiálható karakterek mátrixa (96*10 byte).

Rendszerváltozók:

```
0B00  DB   FF 01 02 FF FF 05 06 FF  : az IN_TABLE
0B08  DB   00 FF 02 FF 04 05 06 FF  : az OUT_TABLE
0B10  DB   INT_DES      : a kurzor által kiszolgált eszközök
0B11  DB   PORT03       : a 3-as port tükörképe
0B12  DB   PORT05       : az 5-ös port tükörképe
0B13  DB   PORT06       : a 6-os port tükörképe
0B14  DB   SND_ACTIVE   ; ha FFH: hang van folyamatban
0B15  DB   SND_IRQ      : ha FFH: az új hang megszakítja a régit
0B16  DB   STOP_FLAG    : ha FFH: CTRL+ESC-et nyomtak
0B17  DW   ST_LIMIT     : a verem alsó határa képkitöltésnél
0B19  DW   HI_MEM       : a legmagasabb használható RAM cím a
                          BASIC számára
0B1B  DB   P3RAM        : 0: a P3 RAM jó
0B1C  DB   EXT_DEF      ; bövítökártya alaphozzárendelés
0B1D  DW   INTINC       : IT számláló
```

```
  0B1F  DB    IRQ_STAT     ; csatolókártya IT engedélyezés
  0B20  DB    INT_FLAG     ; IT kiszolgálás alatt
  0B21  DB    WARM_FLAG    ; ha FFH: meleg reset van folyamatban
  0B22  DB    COLD_FLAG    ; ha FFH: a meleg reset tiltott

A funkcióhívások kezdö programrésze:
  0B23  EX    SP,HL
  0B24  LD    A,(HL)       ; A-ba a hívott funkció kódja
  0B25  INC   HL
  0B26  EX    SP,HL
  0B27  EX    AF,AF'
  0B28  PUSH  AF
  0B29  LD    A,(0003)
  0B2C  PUSH  AF
  0B2D  LD    A,70
  0B2F  LD    (0003),A
  0B32  OUT   02,A         ; memórialapozás: U U U S lapkiosztás
  0B34  JP    C363         ; ugrás a ROM-ba.

A funkcióhívások befejezö programrésze:
  0B37  EX    AF,AF'
  0B38  POP   AF
  0B39  LD    (0003),A
  0B3C  OUT   02,A         ; az hívás elötti lapkiosztás visszaál-
                             lítása
  0B3E  POP   AF
  0B3F  EX    AF,AF'
  0B40  RET

A megszakításkezelö rutin befejezö része:
  0B41  LD    (0003),A
  0B44  OUT   02,A         ; az hívás elötti lapkiosztás visszaál-
                             lítása
  0B46  POP   AF
  0B47  EI                 ; az IT kérés elfogadás engedélyezése
  0B48  RET

  0B49  DW    STACK        ; az SP regiszter ideiglenes tárolása


A video változók:

  0B4B  DB    L_MODE       ; vonalkeresztezési mód
  0B4C  DB    L_STYLE      ; vonaltipus
  0B4D  DB    INK          ; tintaszín
```

```
  0B4E  DB   PAPER       ; papírszín
  0B4F  DB   BORDER      ; keretszín
  0B50  DB   V_FLAG      ; karakter felülírási flag
```

A billentyüzet változók:

```
PICTURE:
  0B51  DB   00 00 00 00 00 00 00 00 00 00
OLDPICT:
  0B5B  DB   00 00 00 00 00 00 00 00 00 00

  0B65  DB   DELAY_KEY   ; auto-repeat késleltetés
  0B66  DB   LOCK_KEY    ; CAPS/SHIFT/ALT lock állapot
  0B67  DB   RATE_KEY    ; auto-repeat ismétlési ütem
  0B68  DB   HOLD_DIS    ; FFH: CTRL+P hatástalan
```

Soros vonal változók:

```
  0B69  DB   BAUD        ; a soros vonal sebessége
  0B6A  DB   FORMAT      ; az USART üzemmód
```

Kazetta változók:

```
  0B6B  DB   BUFFER      ; 0/FFH: nem pufferelt/pufferelt file
  0B6C  DB   REMRED      ; a motorvezérlés kijelölése
  0B6D  DB   PROTECT     ; 0/FFH: nem védett/védett file
  0B6E  DB   EOF         ; FFH: file vége
  0B6F  DW   MUDDLE      ; CRC alapérték
```

Hang változó:

```
  0B71  DB   SER_OK      ; 0/FFH az órafrekvencia jó/nem jó
```

Video munkaterület:

```
  0B73  DB   GR_MODE     ; a grafikus mód
  0B74  DB   P_FLAG      ; 0/FFH: a toll felemelve/letéve
  0B75  DB   00          ; az aktuális képpont helyzete a pontot
                           megvalósító byte által megvalósított
                           pontok között
```

```
  0B76  DW   0000          ; az aktuális pontot megvalósító byte
                             címe a video RAM-ban
  0B78  DW   0000          ; a tollpozícióhoz tartozó X fizikai ko-
                             ordináta
  0B7A  DW   0000          ; a bal alsó sarokhoz viszonyított Y fi-
                             zikai koordináta
  0B7C  DW   0000          ; X logikai koordináta
  0B7E  DW   0000          ; a bal felsö sarokhoz viszonyított Y
                             logikai koordináta

  0B83  DB   00            ; az L_STYLE által meghatározott vonal-
                             tipus képét meghatározó byte

  0B86  DW   0000          ; a vonalhúzás irányától függö rutin cím

  0B8A  DB   00            ; a FILL rutin festö byte-ja

  0B93  DB   paper byte    ; egy paper színü pontot megvalósító
                             byte
  0B94  DB   ink byte      ; egy ink színü pontot megvalósító byte

A billentyüzet munkaterület:

  0BE5  DB   KB_STATUS     ; ha 0/FFH: nincs/van beolvasható karak-
                             ter
  0BE9  DB   00            ; a beolvasható billentyü
  0BEA  DB   00            ; DELAY_KEY számláló
  0BEB  DB   00            ; RATE_KEY számláló

A hang munkaterület:

  0BEF  DB   00            ; hang idötartam

A kazetta munkaterület:

  0BF0  DB   BORDER        ; keret csíkozáshoz a border kód
Input munkaterület (0BF0H-0D13H):
  0BF1  DB   VERIFY_FLAG; ha 0/FFH: beolvasás/összehasonlítás
  0BF2  DB   IT_BUFFER     ; az IT rutin elsö byte-jának átmeneti
                             tárolója
  0BF3  DB   00            ; az input file tipusbyte-ja
  0BF4  DB   00            ; a file név hossza
```

```
A file név:
  0BF5  DB   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Az Input puffer (256 byte): 0C05H-0D04H tartomány

  0D05  DW   0000         ; az input puffer be nem olvasott byte-
                            jainak a száma
  0D07  DW   0000         ; a puffer következő beolvasható byte-
                            jának a címe
  0D09  DW   0000         ; a beolvasandó blokk mérete
  0D0B  DB   ERROR_FLAG   ; beállított hibakód
  0D0C  DB   00           ; beolvasott írásvédelmi byte
  0D0D  DB   00           ; szektorszám mutató
  0D0E  DB   EOF_FLAG     ; file vége jelző
  0D0F  DB   00           ; beolvasott tömbtipus azonosító
  0D10  DW   0000         ; a beolvasott blokk kezdöcíme a memóri-
                            ában
  0D13  DB   00           ; megadott tömbtipus azonosító

Output munkaterület (0D14H-0E34H):
  0D14  DB   00           ; az output file tipusbyte-ja
  0D15  DB   00           ; a file név hossza
A file név:
  0D16  DB   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Az output puffer (256 byte): 0D26H-0E25H tartomány

  0E26  DW   0000         ; a pufferben lévö byte-ok száma
  0E28  DW   0000         ; az aktuális byte címe a pufferben
  0E2A  DB   ERROR_FLAG   ; beállított hibakód
  0E2B  DB   EOF_FLAG     ; a file vége jelző
  0E2C  DW   0000         ; a kiírandó terület kezdö címe
  0E2E  DB   00           ; a kiírandó byte
  0E2F  DB   00           ; ha nem 0, a nem pufferelt file-ból
                            CH_OUT-tal kiírható byte-ok száma
  0E30  DB   00           ; az írásvédelmi byte
  0E31  DB   00           ; szektor mutató
  0E32  DB   00           ; blokktipus mutató


  0E33  DW   STACK        ; az SP regiszter ideiglenes tárolója

Az editor munkaterület:

  0E48  DB   00           ; a kurzor IT számláló
```

```
  0E49  DW   0000          ; a kurzor sor-oszlop pozíció
  0E4B  DW   0000          ; a karakter pozíció címe az ASCII kép-
                             ernyön
  0E4D  DB   CFIX_FLAG     ; 0: nincs CFIX
  0E4E  DW   0000          ; a megjegyzett kurzor pozíció

A LINE_TAB tábla: 0E50H-0E67H tartomány

  0E68  JP   nnnn          ; a kurzor kirajzoló ugró utasítás
  0E6B  DB   00            ; maximális karakter egy sorban
  0E6C  DB   GR_MODE       ; aktuális grafikus mód
A kurzor alatti kép elmentésére szolgáló puffer: 0E6DH-0E94H
  0E95  DB   ink byte      ; egy ink színü pontot elöállító byte
  0E96  DB   paper byte    ; egy paper színü pontot elöállító byte


BASIC változók:

  1700  DB   IX            ; a BASIC alatt IX=1700H. Ennek a byte-
                             nak a bitjei érdekesek:
                             B0=1 : TRACE ON
                               =0 : TRACE OFF
                             B1=0 : "OK"-t kiírni
                               =1 : nincs "OK"
                             B2=0 : parancs mód
                               =1 : program fut
                             B3=0 : nincs nyitott file
                               =1 : van nyitott file

  1702  DB   IF_FLAG       ; ha a 0. bit 1, IF-en belül van a ve-
                             zérlés
  1703  DB   NEW           ; ha FFH, parancs módban törlödik a
                             BASIC program

  1705  DB   00            ; az in/out eszköz számának 10-szerese
  1706  DB   00            ; az az eszköz, melyre a nyomkövetés ír
  1707  DB   AUTO          ; FFH: auto-RUN
  1708  DB   TYPE          ; a szimbólumtábla aktuális elemének ti-
                             pusa
  1709  DB   00            ; egy byte-os elö-véletlenszám
  170A  DW   0000          ; két byte-os elö-véletlenszám
  170C  DW   START         ; az aktuális BASIC sor kezdöcíme
  170E  DW   STOP          ; a STOP esetén az aktuális START cím
  1710  DW   RESTART       ; a STOP-ot követö utasítás címe
  1712  DW   0000          ; az aktuális DATA sor címe
```

```
  1714  DW   0000          ; az aktuális DATA-ban a következö adat
                             címe
  1718  DW   RST 18H       ; az RST 18H utasítás következö byte-já-
                             ra mutat
  171A  DW   IY            ; IY értéke egy utasítás végrehajtása e-
                             lött
  1720  DW   VLOMEM        ; a BASIC terület kezdete
  1722  DW   TEXT          ; a BASIC program kezdöcíme
  1724  DW   CHAIN         ; a szimbólumtábla utolsó eleme címe
  1726  DW   TOP           ; a szimbólumtábla végének a címe
```

Az aktuális BASIC utasítássort tartalmazó puffer (255 byte):
1732H-1830H tartomány

Az INPUT utasítás puffere (255 byte): 1831H-192FH tartomány

Puffer a STR$ függvénynek:

```
  1930  DB   00 00 00 00 00 00 00 00 00
```

Az X aritmetikai regiszter:

```
  19C0  DB   00 00 00 00 00 00 00
```

Az Y aritmetikai regiszter:

```
  19C7  DB   00 00 00 00 00 00 00
```

A nem pufferelt file feje:

```
  19DF  DB   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

A BASIC terület kezdete: 19EFH

ROM LISTA

I. SYSTEM ROM

A rendszer ROM fő része

```
C000  JP   0229        ; bekapcsoláskor ez a 0000 cím
```

Szorzótábla:

```
C003  DB   00 00 00 00 00 00 00 00 00 00  ; a számok a táblá-
C00D  DB   00 01 02 03 04 05 06 07 08 09  ; zatban BCD számok
C017  DB   00 02 04 06 08 10 12 14 16 18
C021  DB   00 03 06 09 12 15 18 21 24 27
C02B  DB   00 04 08 12 16 20 24 28 32 36
C035  DB   00 05 10 15 20 25 30 35 40 45
C03F  DB   00 06 12 18 24 30 36 42 48 54
C049  DB   00 07 14 21 28 35 42 49 56 63
C053  DB   00 08 16 24 32 40 48 56 64 72
C05D  DB   00 09 18 27 36 45 54 63 72 81
```

A BASIC utasítások ugrótáblája:

```
C067  DW   DBBB DBBB DB80 DBBB   ; sor vége jel,!,:,REM
C06F  DW   DFF2 E89B DFFC DD65   ; DATA, CLOSE, CLS, CONTINUE
C077  DW   E002 DD93 E053 E104   ; DEF, DELETE, DIM, ELSE
C07F  DW   E10E E15C E910 E382   ; END, FOR, GET, GOSUB
C087  DW   E3B2 E733 E2EE E1CB   ; GOTO, GRAPHICS, IF, INPUT
C08F  DW   E3C1 DD85 DD80 E951   ; LET, LIST, LLIST, LOAD
C097  DW   E452 DE08 E4B6 DB06   ; LOMEM, NEW, NEXT, OK
C09F  DW   E332 E8C9 E573 E542   ; ON, OPEN, OUTPUT, OUT
C0A7  DW   E754 E553 E573 E6C7   ; PLOT, POKE, PRINT, RANDOMIZE
C0AF  DW   E21B E6F4 E70F DE1B   ; READ, RESTORE, RETURN, RUN
C0B7  DW   E982 E790 E833 FFA3   ; SAVE, SET, SOUND, STOP
C0BF  DW   DE31 E9D3 E117 E570   ; TRACE, VERIFY, EXT, LPRINT
```

AZ "RST 18" (aritmetikai) rutinok ugrótáblája:

```
C0C7  DW   F493   ;  0. rutin: (IY+9)=(IY+9)+(IY)   IY=IY+9
C0C9  DW   F5FB   ;  1. rutin: (IY+9)=(IY+9)/(IY)   IY=IY+9
C0CB  DW   F512   ;  2. rutin: (IY+9)=(IY+9)*(IY)   IY=IY+9
```

```
C0CD  DW   F48E   ;  3. rutin: (IY+9)=(IY+9)-(IY)    IY=IY+9
C0CF  DW   F726   ;  4. rutin:   (IY)=-(IY)          IY=IY
C0D1  DW   EA82   ;  5. rutin: (IY)=n-dik konstans   IY=IY-9
C0D3  DW   EA9F   ;  6. rutin: (IY)= X aritm. reg.   IY=IY-9
C0D5  DW   EA9A   ;  7. rutin: (IY)= Y aritm. reg.   IY=IY-9
C0D7  DW   EAD2   ;  8. rutin: X aritm. reg.=(IY)    IY=IY
C0D9  DW   EACD   ;  9. rutin: Y aritm. reg.=(IY)    IY=IY
C0DB  DW   EAC3   ; 10. rutin: X aritm. reg.=(IY)    IY=IY+9
C0DD  DW   EABE   ; 11. rutin: Y aritm. reg.=(IY)    IY=IY+9
C0DF  DW   FA92   ; 12. rutin: (IY-9)=(IY)           IY=IY-9
C0E1  DW   FB28   ; 13. rutin: (HL)=(IY)             IY=IY+9
C0E3  DW   EA68   ; 14. rutin: (IY)=függv. argum.    IY=IY-9

C0E5  DB   "Copyright (c) 1984  Intelligent Software Ltd"
```

A ROM-ban tárolt számozott konstansok:

```
C111  DB   00 00 00 00 00 50 3F   ;  0. konstans: 0.5
C118  DB   00 00 00 00 00 10 40   ;  1. konstans: 1.0
C11F  DB   31 24 19 49 79 26 3F   ;  2. konstans
C126  DB   57 07 08 05 32 17 40   ;  3. konstans
C12D  DB   69 75 80 50 20 73 3F   ;  4. konstans
C134  DB   74 48 34 08 40 14 C0   ;  5. konstans
C13B  DB   98 88 84 26 00 72 BF   ;  6. konstans
C142  DB   19 89 03 25 20 43 40   ;  7. konstans
C149  DB   99 45 58 22 52 47 40   ;  8. konstans
C150  DB   07 38 96 88 85 86 3F   ;  9. konstans
C157  DB   00 00 00 00 51 11 40   ; 10. konstans
C15E  DB   23 70 49 46 25 29 3C   ; 11. konstans
C165  DB   06 95 88 64 44 50 42   ; 12. konstans
C16C  DB   63 75 99 82 00 14 41   ; 13. konstans
C173  DB   16 65 64 73 28 33 3E   ; 14. konstans
C17A  DB   01 79 97 92 08 10 43   ; 15. konstans
C181  DB   97 10 08 94 20 11 42   ; 16. konstans
C188  DB   99 92 50 58 02 23 40   ; 17. konstans
C18F  DB   90 37 14 68 15 29 C0   ; 18. konstans
C196  DB   57 15 49 03 63 31 40   ; 19. konstans
C19D  DB   78 14 60 81 35 67 BF   ; 20. konstans
C1A4  DB   42 95 06 04 07 10 C1   ; 21. konstans
C1AB  DB   21 40 81 69 96 16 41   ; 22. konstans
C1B3  DB   67 54 04 80 90 81 C0   ; 23. konstans
C1BA  DB   07 38 96 88 85 86 3F   ; 24. konstans
C1C0  DB   88 60 66 66 66 16 BF   ; 25. konstans
```

```
C1C7  DB   56 20 07 33 33 83 3D   ; 26. konstans
C1CE  DB   31 82 32 08 84 19 BC   ; 27. konstans
C1D5  DB   78 06 71 39 52 27 3A   ; 28. konstans
C1DC  DB   60 40 46 83 86 23 B8   ; 29. konstans
C1E3  DB   00 00 00 07 36 22 3F   ; 30. konstans
C1EA  DB   00 00 00 27 44 89 3F   ; 31. konstans
C1F1  DB   17 60 76 27 62 31 3F   ; 32. konstans
C1F8  DB   31 51 79 57 29 57 41   ; 33. konstans
C1FF  DB   59 53 26 59 41 31 40   ; 34. konstans: PI
C206  DB   79 26 63 79 70 15 40   ; 35. konstans: PI/2
C20D  DB   20 51 75 19 47 10 40   ; 36. konstans
C214  DB   98 55 77 98 35 52 3F   ; 37. konstans
C21B  DB   00 00 00 80 76 32 44   ; 38. konstans
C222  DB   00 99 99 99 99 99 7E   ; 39. konstans


C229  DI                  ; a maszkolható megszakítás kiszolgálás
                            letiltása
C22A  IM 1                ; az 1-es megszakítás kezelési mód beál-
                            lítása
C22C  LD   A,40
C22E  OUT  02,A           ; memórialapozás: S U V S lapkiosztás
C230  JP   0233           ; ugrás a P0 memórialapra
C233  LD   A,C0
C235  OUT  02,A           ; memórialapozás: S U V E lapkiosztás
C237  JP   F13D           ; ugrás az Extension ROM-ba
C23A  LD   A,40
C23C  OUT  02,A           ; memórialapozás: S U V S lapkiosztás
C23E  JP   C241           ; ugrás a P3 memórialapra
C241  LD   A,50
C243  OUT  02,A           ; memórialapozás: U U V S lapkiosztás

A paletta regiszterek törlése:
C245  XOR  A
C246  LD   BC,0460
C249  OUT  (C),A
C24B  INC  C
C24C  DJNZ FB

A videovezérlő (6845) regisztereinek inicializálása:
C24E  LD   HL,C545        ; HL-be az inicializáló táblázat címe
C251  LD   B,10           ; B-be a vezérlő regisztereinek száma
C253  LD   A,B
C254  DEC  A
```

```
C255  OUT  70,A        ; a vezérlö egy regiszterének kiválasz-
                         tása
C257  LD   A,(HL)      ; A-ba a kiválasztott regiszterbe töl-
                         tendö adat
C258  OUT  71,A        ; a vezérlö adott regiszterének feltöl-
                         tése
C25A  INC  HL          ; a következö regiszter kiválasztása
C25B  DJNZ F6

C25D  LD   A,80
C25F  OUT  06,A        ; egy STROBE jel kiküldése
C261  LD   A,(0B22)    ; a COLD_FLAG az A regiszterbe
C264  INC  A
C265  JR   NZ,04       ; ugrás a C26AH címre, ha a "meleg"
                         RESET nem tiltott
C267  LD   (0B21),A    ; a WARM_FLAG beállítása: "hideg" RESET
C26A  LD   A,08
C26C  INC  A
C26D  JR   Z,4C        ; ugrás a C2BBH címre
C26F  LD   SP,BFFF
C272  LD   HL,0000     ; HL-be a P0 memórialap kezdöcíme
C275  CALL C33E        ; P0 lap tesztelése
C278  JR   Z,05        ; ugrás a C27FH címre
```

Ha a P0 RAM lap hibás, a BORDER-t villogtató végtelen ciklusba kerül a gép:

```
C27A  DEC  A
C27B  OUT  00,A        ; a BORDER beállítása
C27D  JR   FB          ; ugrás a C27AH címre

C27F  LD   SP,16AC
C282  LD   HL,8000     ; a video-RAM kezdö címe a HL-be
C285  CALL C33E        ; a video-RAM tesztelése
C288  JR   Z,04        ; ha a video-RAM jó, ugrás a C28EH címre
C28A  LD   A,88
C28C  OUT  (C),A
C28E  LD   A,70
C290  OUT  02,A        ; memórialapozás: U U U S lapkiosztás
C292  LD   HL,4000     ; HL-be a P1 lap kezdete
C295  CALL C33E        : a P1 lap tesztelése
C298  CALL Z,C33E      ; ha P1 hibátlan, a P2 RAM tesztelése
C29B  DEC  HL
C29C  LD   (0B19),HL   : a HI_MEM változó feltöltése
C29F  LD   A,40
C2A1  OUT  02,A        ; memórialapozás: S U V S lapkiosztás
```

```
C2A3  LD   SP,BFFF
C2A6  JP   02A9          ; ugrás a P0 lapra
C2A9  LD   A,80
C2AB  OUT  02,A          ; memórialapozás: S U V U+ lapkiosztás
C2AD  LD   HL,C000
C2B0  CALL 0338          ; az U+ felhasználói RAM tesztelése
C2B3  LD   A,00
C2B5  JR   Z,01          ; ugrás a C2B8H címre, ha az U+ RAM rossz
C2B7  DEC  A             ; A-ba FFH kerül
C2B8  EX   AF,AF'
C2B9  JR   07            ; ugrás a C2C2H címre
C2BB  LD   H,A
C2BC  LD   L,A
C2BD  DEC  HL
C2BE  LD   A,H
C2BF  OR   L
C2C0  JR   NZ,FB         ; ha HL nem nulla, ugrás a C2BDH címre
C2C2  LD   A,40
C2C4  OUT  02,A          ; memórialapozás: S U V S lapkiosztás
C2C6  JP   02C9          ; ugrás a P0 lapra
C2C9  LD   A,C0
C2CB  OUT  02,A          ; memórialapozás: S U V E lapkiosztás
C2CD  LD   SP,16AC       ; a Stack Pointer beállítása
C2D0  JP   F000          ; ugrás az EXTENSION ROM első byte-jára
C2D3  XOR  A             ; A-ba 0-t tölt
C2D4  LD   (0B11),A      ; a PORT03 rendszerváltozó törlése
C2D7  OUT  03,A          ; PORT03 írása
C2D9  OUT  07,A          ; cursor/hang IT törlés

Csatolókártyák IT tiltása:
C2DB  OUT  58,A
C2DD  OUT  59,A
C2DF  OUT  5A,A
C2E1  OUT  5B,A

C2E3  OUT  00,A          ; BORDER beállítása feketére
C2E5  LD   (0B4F),A      ; a BORDER video változó beállítása
C2E8  LD   A,80
C2EA  OUT  06,A          ; PORT06 írása (STROBE jel)
C2EC  LD   (0B13),A      ; a PORT06 rendszerváltozó beállítása

A rendszer inicializálása egy táblázatból:
C2EF  LD   HL,C555       ; a táblázat címe a HL-be
C2F2  LD   BC,0470       ; B-be a táblázat elemeinek száma (4),
                         ; C-be a memórialapozást meghatározó byte
```

(70H) kerül

```
  C2F5  PUSH BC
```

DE-be a meghívandó rutin címe, HL-be a táblázat következő elemének címe kerül:

```
  C2F6  LD   E,(HL)
  C2F7  INC  HL
  C2F8  LD   D,(HL)
  C2F9  INC  HL

  C2FA  PUSH HL
  C2FB  EX   DE,HL       ; HL-be a meghívandó rutin címe kerül
  C2FC  CALL C321        ; "CALL (HL)" rutin
  C2FF  POP  HL
  C300  POP  BC
  C301  LD   A,C         ; C-be 70H kerül
  C302  OUT  02,A        ; memórialapozás: U U U S lapkiosztás
  C304  DJNZ EF          ; ha B nem 0, ugrás a C2F5H címre
  C306  LD   A,60
  C308  OUT  02,A        ; memórialapozás: S U U S lapkiosztás
  C30A  JP   030D        ; ugrás a P0 lapra
  C30D  LD   A,20
  C30F  OUT  02,A        ; memórialapozás: S U U C lapkiosztás
```

Az oldalt bedugható ROM-modul vizsgálata:

```
  C311  LD   HL,C000     ; HL-be a ROM-modul kezdete
  C314  LD   DE,0334     ; DE-be a P0 lapra lapozott SYSTEM ROM-
                         ; ban található "MOPS" azonosító címe
  C317  LD   B,04
  C319  LD   A,(DE)
  C31A  CP   (HL)
  C31B  JR   NZ,05       ; ha a ROM-modul nem "MOPS" karaktersor-
                         ; rozattal kezdődik, ugrás a C332H címre
  C31D  INC  DE
  C31E  INC  HL
  C31F  DJNZ F8          ; ha B nulla, a következő utasítás a ve-
                         ; zérlést a ROM-modulnak adja át

  C321  JP   (HL)
  C322  LD   A,60
  C324  OUT  02,A        ; memórialapozás: S U U S lapkiosztás
  C326  JP   C329        ; ugrás a P3 lapra
  C329  LD   A,70
  C32B  OUT  02,A        ; memórialapozás: U U U S lapkiosztás
  C32D  LD   (0003),A    ; a P_SAVE rendszerváltozó elmentése
```

```
  C330  EI                ; maszkolható megszakítás engedélyezése
  C331  JP   D9EF         ; ugrás a BASIC interpreter inicializá-
                          ; lására

  C334  DB   "MOPS"

A RAM lapok tesztelése (HL-ben a memórialap kezdöcime található):
  C338  PUSH HL           ; a tesztelendö memórialap a P3 lapon
  C339  CALL 0348         ; található
  C33C  JR   04           ; ugrás a C342H címre

  C33E  PUSH HL
  C33F  CALL C348         ; ugrás a tényleges tesztelésre
  C342  POP  DE
  C343  RET  NZ           ; RET, ha hibás a memórialap
  C344  EX   DE,HL
  C345  LD   A,AA
  C347  LD   BC,553E      ; ha ide adódik a vezérlés, "üres" uta-
                            sítás
  ***
  C348  LD   A,55         ; ez csak akkor létezö utasítás, ha a
  ***                       vezérlés ide adódik

  C34A  PUSH HL

Az adott memórialap feltöltése az A registerben tárolt értékkel:
  C34B  LD   E,L
  C34C  LD   D,H
  C34D  INC  DE
  C34E  LD   (HL),A
  C34F  LD   BC,3FFF
  C352  LDIR

  C354  POP  HL           ; HL-be a memórialap kezdöcíme
  C355  LD   B,40         ; BC-be 4000H kerül

A feltöltött memórialap ellenörzése, törlése:
  C357  DEC  A
  C358  DEC  (HL)
  C359  CPI               ; ellenörzés
  C35B  DEC  HL
  C35C  LD   (HL),00      ; törlés
  C35E  RET  NZ           ; RET, ha hibás a RAM
  C35F  INC  HL
  C360  RET  PO           ; RET, ha vége az ellenörzésnek
```

```
  C361  JR   F5          ; ugrás a C358H címre
A RAM lap tesztelés rutin vége.

Az RST 30H rutin kezdete, regiszterek elmentése a STACK-be:
  C363  PUSH HL
  C364  PUSH IX
  C366  PUSH IY
  C368  EXX
  C369  PUSH BC
  C36A  PUSH DE
  C36B  PUSH HL
  C36C  EXX

  C36D  EX   AF,AF'
  C36E  CALL C37E        ;az RST 30H rutin magjának meghívása

Az RST 30H rutin vége, a regiszterek visszaállítása:
  C371  EXX
  C372  POP  HL
  C373  POP  DE
  C374  POP  BC
  C375  EXX
  C376  POP  IY
  C378  POP  IX
  C37A  POP  HL

  C37B  JP   0B37        ; ugrás a RAM-ba

Az RST 30H rutin magja:
  C37E  PUSH AF          ; A-ban a rutin kódja található
  C37F  PUSH BC          ; BC-ben és DE-ben a rutin INPUT para-
  C380  PUSH DE          ; méterei lehetnek
  C381  LD   C,A
  C382  OR   A
  C383  EX   AF,AF'
  C384  LD   HL,0B01     ; HL-be a billentyüzet funkcióosztály
                         IN_TABLE- beli címe kerül
  C387  LD   A,(HL)      ; az A-ba a billentyüzet funkcióosztály-
                         hoz rendelt input eszköz száma kerül
  C388  CP   02
  C38A  JR   NZ,01       ; ha nem az editor van a billentyüzet
                         osztályhoz rendelve, ugrás a C38DH-ra
  C38C  DEC  (HL)        ; különben a billentyüzet eszköz vissza-
                         rendelése a billentyüzet osztályhoz
  C38D  LD   DE,0B00     ; DE- be az IN_TABLE tábla címe
```

```
C390  LD   A,C          ; A-ba az OS rutinjának kódja kerül
C391  RLCA              ; a CARRY flag-be a funkcióhívás I/O
                          bitje kerül
C392  JR   C,03         ; INPUT hívás esetén ugrás a C399H címre
C394  LD   DE,0B08      ; OUTPUT hívás esetén DE-be az OUT_TABLE
                          tábla címe kerül
C397  AND  E0
C399  RLCA
C39A  RLCA
C39B  RLCA              ; A-ba a meghívott funkcióosztály száma
C39C  LD   B,A          ; B-be a funkcióosztály száma
C39D  LD   A,C          ; A-ba a funkciókód
C39E  AND  0F           ; A-ba a funkciószám
C3A0  LD   C,A          ; C-be a funkciószám
C3A1  LD   A,B          ; A-ba a funkcióosztály száma
C3A2  CP   07
C3A4  JR   Z,32         ; ha csatolókártya I/O kijelölésről van
                          szó, ugrás a C3D8H címre
C3A6  LD   L,B
C3A7  XOR  A            ; A-ba 00H
C3A8  LD   H,A          ; HL-be a funkcióosztály száma
C3A9  ADD  HL,DE        ; HL-be a funkcióosztályhoz rendelt esz-
                          köz byte-jának címe
C3AA  DEC  A            ; A-ba FFH
C3AB  CP   (HL)         ; értelmezett-e a funkció?
C3AC  JR   NZ,05        ; ha igen,ugrás a C3B3H címre
C3AE  LD   A,C          ; A-ba a funkciószám
C3AF  CP   03
C3B1  JR   NC,1D        ; ha a funkciószám nagyobb 2-nél, ugrás
                          a C3D0H címre
C3B3  LD   A,(HL)       ; A-ba az eszközszám
C3B4  RES  7,A          ; a legfelsö bit törlése
C3B6  CP   07
C3B8  JR   C,06         ; ha az eszközszám kisebb 7-nél, a C3C0H
                          címen folytatódik a rutin
C3BA  LD   A,FE         ; különben pedig A-ba a hibakód kerül
C3BC  POP  DE
C3BD  POP  BC
C3BE  JR   3C           ; ugrás a C3FCH címre
C3C0  BIT  7,(HL)       ; az eszközszám legfelsö bitjének vizs-
                          gálata
C3C2  JR   Z,06         ; ha B7=0, ugrás a C3CA címre
C3C4  LD   HL,F166      ; különben HL-be az EXTENSION ROM egy
                          rutinjának címe (csatolókártya tipusá-
                          nak meghatározása)
```

```
C3C7  JP   FFF0      ; ugrás az EXT. ROM-ba átvivö rutinra
C3CA  LD   A,C       ; A-ba a funkciószám
C3CB  CP   03
C3CD  JR   NC,01     ; ha a funkciószám >2, ugrás a C3D0H-ra
C3CF  LD   B,(HL)    ; különben B-be az osztályhoz rendelt
                       eszközszám
C3D0  LD   A,B       ; A-ba az eszközszám
C3D1  CP   06        ; csatolókártya az eszköz?
C3D3  LD   HL,F16C   ; az EXT. ROM-beli csatolókártya rutin
                       címe a HL-be
C3D6  JR   Z,EF      ; ha csatolókártya rutint kell végrehaj-
                       tani, ugrás a C3C7H címre
C3D8  LD   DE,C55D   ; különben DE-be az eszköz rutinok
                       ugrótábláinak címét tartalmazó tábla
                       címe kerül
C3DB  LD   L,B
C3DC  LD   H,00      ; HL-be az eszközszám kerül
C3DE  ADD  HL,HL
C3DF  ADD  HL,DE     ; HL-be ez eszköz rutinok ugrótáblája
                       címének a címe kerül
C3E0  LD   E,(HL)
C3E1  INC  HL
C3E2  LD   D,(HL)
C3E3  EX   DE,HL     ; HL-be az eszköz rutinok ugrótáblájának
                       címe kerül
C3E4  LD   A,(HL)    ; A-ba az eszköz rutinok száma
C3E5  DEC  A
C3E6  CP   C
C3E7  LD   A,FF
C3E9  JR   C,D1      ; ha a meghívott rutin száma nagyobb a
                       létezö rutinok számánál, ugrás a C3BCH
                       címre (nem létezö eszköz rutin)
C3EB  INC  HL        ; különben a 0. eszköz rutin címének a
                       címe a HL-be
C3EC  EX   DE,HL
C3ED  LD   L,C
C3EE  LD   H,00      ; HL-be a funkciószám
C3F0  ADD  HL,HL
C3F1  ADD  HL,DE     ; HL-be a keresett eszköz rutin címének
                       a címe
C3F2  LD   E,(HL)
C3F3  INC  HL
C3F4  LD   D,(HL)
C3F5  EX   DE,HL     ; HL-be a keresett eszköz rutin címe
C3F6  EX   AF,AF'
```

```
  C3F7  POP  DE
  C3F8  POP  BC          ; DE-be és BC-be a rutin IN paraméterei
  C3F9  CALL C321        ; "CALL (HL)"
  C3FC  OR   A
  C3FD  POP  HL
  C3FE  RET  Z           ; RET, ha hiba nélkül futott a rutin
  C3FF  PUSH AF
  C400  LD   A,(0B20)    ; A-ba az INT_FLAG
  C403  INC  A
  C404  JR   Z,0A        ; ugrás a C410H címre, ha nincs IT ki-
                           szolgálás alatt
  C406  PUSH BC
  C407  PUSH DE
  C408  LD   A,H         ; A-ba a funkcióhívás kódja
  C409  LD   HL,F1E6     ; HL-be az alap hozzárendelést vissza-
                           állító rutin EXT. ROM-beli címe
  C40C  JR   B9          ; ugrás a C3C4H címre
  C40E  POP  DE
  C40F  POP  BC
  C410  POP  AF
  C411  RET              ; az RST 30H rutin vége a ROM-ban

Az RST 38H (IT kiszolgáló) rutin kezdete a ROM-ban
  C412  LD   A,(0003)    ; A-ba a P_SAVE rendszerváltozó

A regiszterek elmentése a STACK-be:
  C415  PUSH AF
  C416  PUSH HL
  C417  PUSH DE
  C418  PUSH BC
  C419  PUSH IX
  C41B  PUSH IY
  C41D  EX   AF,AF'
  C41E  PUSH AF
  C41F  EXX
  C420  PUSH HL
  C421  PUSH DE
  C422  PUSH BC

  C423  CALL C437        ; az RST 38H rutin törzsének meghívása

A regiszterek visszaállítása az IT kiszolgálása után:
  C426  POP  BC
  C427  POP  DE
  C428  POP  HL
```

```
  C429  EXX
  C42A  POP  AF
  C42B  EX   AF,AF'
  C42C  POP  IY
  C42E  POP  IX
  C430  POP  BC
  C431  POP  DE
  C432  POP  HL
  C433  POP  AF

  C434  JP   0B41          ; ugrás a RAM-ba

Az RST 38H rutin törzse:
  C437  LD   A,FF
  C439  LD   (0B20),A      ; az INT_FLAG beállítása: IT kiszolgálás
                             alatt
  C43C  LD   HL,(0B1D)     ; HL-be az IT számláló értéke
  C43F  INC  HL            ; HL növelése
  C440  LD   (0B1D),HL     ; az IT számláló elmentése
  C443  LD   A,(0B4F)      ; a BORDER rendszerváltozó az A-ba
  C446  OUT  00,A          ; a border beállítása a BORDER értéke
                             szerint
  C448  IN   A,59          ; A-ba a függöben levö IT kéréseket meg-
                             határozó érték
  C44A  LD   C,A
  C44B  BIT  4,A           ; curzor/hang IT?
  C44D  SET  4,C           ; A 4. bitje legyen 1
  C44F  OUT  07,A          ; curzor/hang IT törlés
  C451  CALL Z,C47D        ; ha curzor/hang IT kérés jött, szubru-
                             tin hívás
  C454  LD   A,C
  C455  OR   F0
  C457  INC  A
  C458  JR   Z,1E          ; ha nincs más IT kérés, ugrás a C478H
                             címre
  C45A  LD   A,(0B1F)      ; A-ba az IRQ_STAT rendszerváltozó
  C45D  RRCA
  C45E  RRCA
  C45F  LD   H,C
  C460  LD   L,C
  C461  LD   BC,0458       ; B-be 04H, C-be 58H
  C464  LD   D,A

Csatolókártya IT tiltás/törlés (58H-5BH portok írása):
  C465  XOR  A
```

```
C466  RRC  D
C468  RRA
C469  RRC  H
C46B  JR   C,02
C46D  OUT  (C),A
C46F  INC  C
C470  DJNZ F3

C472  LD   C,L          ; C visszaállítása
C473  LD   HL,C478      ; visszatérési cím a csatolókártya IT
                          kiszolgálás után
C476  JR   2B           ; ugrás a C4A3H címre
C478  XOR  A            ; A-ba 00H
C479  LD   (0B20),A     ; az INT_FLAG törlése
C47C  RET               ; az RST 38H rutin magjának vége

C47D  PUSH BC
C47E  LD   A,(0B10)     ; A-ba az INT_DES rendszerváltozó
C481  LD   C,A
C482  LD   B,04
C484  RR   C
C486  PUSH BC
C487  JR   C,10
C489  LD   A,04
C48B  SUB  B
C48C  LD   B,A          ; B-be az IT kiszolgálást kérö eszköz
                          száma
C48D  LD   C,00         ; C-be az eszköz IT funkciószáma
C48F  LD   HL,C499      ; HL-be a visszatérési cím az IT kiszol-
                          gálás után
C492  PUSH HL
C493  PUSH HL
C494  PUSH HL
C495  PUSH HL
C496  JP   C3D8         ; ugrás az eszköz IT kiszolgáló rutinra
C499  LD   A,70
C49B  OUT  02,A         ; memórialapozás: U U U S lapkiosztás
C49D  POP  BC
C49E  DJNZ E4           ; ha B nem nulla, ugrás a C484H címre
C4A0  LD   HL,C4AA
C4A3  PUSH HL           ; a visszatérési cím letétele a STACK-be
C4A4  LD   HL,F227      ; a csatolókártya IT kiszolgáló rutin
                          címe az EXT. ROM-ban
C4A7  JP   FFF0         ; ugrás az EXT. ROM-ba vivö rutinra
C4AA  POP  BC
```

```
C4AB  RET                ; az RST 38H rutin magjának vége
```

A kernel rutinok ugrótáblája:

```
C4AC  DB   05      ; a rutinok száma
C4AD  DW   C509    ; KER_INT (kernel interapt) rutin címe
C4AF  DW   C4B7    ; HI_MEM_SET rutin címe
C4B1  DW   C4D0    ; EXP_DIRECT rutin címe
C4B3  DW   C4E2    ; I/O_DIRECT rutin címe
C4B5  DW   C50E    ; SLOT_NUM rutin címe
```

A HI_MEM_SET rutin:

```
C4B7  LD   HL,(0B19)   ; HL-be a HI_MEM rendszerváltozó
C4BA  LD   A,FB        ; "nincs elég memória" hibakód A-ba
C4BC  OR   A           ; a CARRY flag törlése
C4BD  SBC  HL,DE
C4BF  RET  C           ; a lefoglalni kívánt memória nagyobb.
                         mint a teljes memória
C4C0  PUSH HL
C4C1  LD   DE,1EAC
C4C4  SBC  HL,DE
C4C6  POP  HL
C4C7  RET  C           ; a helyfoglalás után 1EACH-nál kevesebb
                         hely marad
C4C8  LD   D,H
C4C9  LD   E,L
C4CA  INC  DE          ; DE-be az új HI_MEM+1
C4CB  LD   (0B19),HL   ; az új HI_MEM elmentése
C4CE  XOR  A           ; A-ba 00H: nincs hiba
C4CF  RET              ; a HI_MEM_SET rutin vége
```

Az EXP_DIRECT rutin:

```
C4D0  LD   HL,0B07     ; HL-be az IN_TABLE tábla csatolókártya
                         input kijelölés elemének a címe
C4D3  INC  B
C4D4  JR   Z,03        ; B-ben FFH volt: input kijelölés. ugrás
                         a C4D9H címre
C4D6  LD   HL,0B0F     ; különben output kijelölés: HL-be az
                         OUT_TABLE tábla csatolókártya output
                         kijelölés elemének a címe
C4D9  PUSH HL
C4DA  CALL C50E        ; a SLOT_NUM rutin meghívása
C4DD  POP  HL
C4DE  OR   A           ; hiba nélkül futott a SLOT_NUM rutin?
```

```
  C4DF  RET  NZ          ; ha nem, hibával visszatérés az
                           EXP_DIRECT rutinból
  C4E0  LD   (HL),C      ; az egységszám betöltése az IN/OUT_TABLE
                           tábla csatolókártya kijelölés elemének
                           a címére
  C4E1  RET              ; az EXP_DIRECT rutin vége

Az I/O_DIRECT rutin:
  C4E2  LD   HL,0B00     ; HL-be az IN_TABLE tábla címe
  C4E5  INC  D
  C4E6  JR   Z,03        ; D-ben FFH volt: input hozzárendelés
  C4E8  LD   HL,0B08     ; különben HL-be az OUT_TABLE tábla címe
  C4EB  PUSH HL
  C4EC  LD   A,06        ; A-ba a csatolókártya funkcióosztály
                           száma
  C4EE  CP   B
  C4EF  JR   C,19        ; hozzárendelési hiba, ugrás a C50AH-ra
  C4F1  LD   E,B
  C4F2  LD   D,00        ; DE-be a funkcióosztály száma
  C4F4  ADD  HL,DE       ; HL-be a funkcióosztály IN/OUT_TABLE
                           táblabeli címe
  C4F5  LD   E,(HL)      ; E-be az adott funkcióosztályhoz ren-
                           delt eszköz száma
  C4F6  INC  E
  C4F7  JR   Z,11        ; E-be FFH került: értelmetlen hozzáren-
                           delés, ugrás a C50AH címre
  C4F9  POP  DE
  C4FA  PUSH HL
  C4FB  EX   DE,HL
  C4FC  CP   C
  C4FD  JR   C,0B        ; az eszköz szám nagyobb 6-nál, ugrás a
                           C50AH címre
  C4FF  LD   E,C         ; E-be az eszköz száma
  C500  XOR  A
  C501  LD   D,A         ; D-be 00H
  C502  ADD  HL,DE
  C503  LD   E,(HL)      ; E-be az adott eszközhöz tartozó funk-
                           cióosztály IN/OUT_TABLE-beli eleme
  C504  INC  E
  C505  JR   Z,03        ; az adott byte FFH volt: értelmetlen
                           hozzárendelés, ugrás a C50AH címre
  C507  POP  HL
  C508  LD   (HL),C      ; a funkcióosztály-eszköz hozzárendelés
  C509  RET              ; az I/O_DIRECT rutin vége hibátlan fu-
                           táskor
```

```
  C50A  POP  HL
  C50B  LD   A,FE          ; A-ba a hibakód
  C50D  RET                ; az I/O_DIRECT rutin vége hiba esetén

A SLOT_NUM rutin:
  C50E  PUSH DE
  C50F  LD   IX,0040       ; IX-be a csatolókártya azonosítók táb-
                             lázatának címe
  C513  LD   B,04          ; B-be a kártyahelyek száma
  C515  PUSH IX
  C517  POP  HL
  C518  PUSH BC
  C519  LD   A,(DE)        ; A-ba a kártyaazonosító név hossza
  C51A  INC  A
  C51B  LD   B,A
  C51C  LD   A,(DE)
  C51D  CP   (HL)
  C51E  JR   NZ,17         ; ha a megadott kártyaazonosító nem a
                             vízsgált csatlakozó helyen van, ugrás
                             a C537H címre
  C520  INC  HL
  C521  INC  DE
  C522  DJNZ F8            ; ha nincs vége az összehasonlításnak,
                             ugrás a C51CH címre
  C524  POP  BC
  C525  PUSH IX
  C527  POP  HL
  C528  LD   DE,0007
  C52B  ADD  HL,DE
  C52C  LD   A,(HL)        ; A-ba az adott azonosítójú bővítökártya
                             egységszáma
  C52D  CP   C             ; ezt kerestük?
  C52E  JR   NZ,08         ; ha nem, ugrás a C538H címre
  C530  LD   A,04
  C532  SUB  B             ; A-ba a csatlakozószám
  C533  LD   C,A           ; C-be a keresett csatlakozószám
  C534  XOR  A             ; A-ba 00H: hibátlan futás
  C535  POP  DE
  C536  RET                ; a SLOT_NUM rutin vége

A következő csatlakozóhoz tartozó bejegyzés címének meghatározá-
sa a csatolókártyák azonosítóinak táblázatában:
  C537  POP  BC
  C538  LD   DE,0030
  C53B  ADD  IX,DE
```

```
C53D  POP  DE
C53E  PUSH DE

C53F  DJNZ D4          ; ugrás a C515H címre, ha van még ellen-
                         örizetlen csatlakozó hely
C541  LD   A,FD        ; különben A-ba a hibakód kerül
C543  JR   F0          ; ugrás a C536H címre

C545  DB   FF 0E 00 00 03 03 03 00   ; a képernyövezérlö re-
C54D  DB   42 3C 02 4D 32 4B 40 63   ; gisztereinek tartalma
                                     ; reset után


C555  DW   C9F2    ; a rendszer inicializálása során
C557  DW   D5EC    ; használt rutinok táblázata
C559  DW   D960
C55B  DW   D9E2

C55D  DW   C974    ; a video rutinok ugrótáblájának címe
C55F  DW   D5E3    ; a billentyüzet rutinok ugrótáblájának címe
C561  DW   CF98    ; az editor rutinok ugrótáblájának címe
C563  DW   D92A    ; a hangkezelö rutinok ugrótáblájának címe
C565  DW   D8FF    ; a nyomtató rutinok ugrótáblájának címe
C567  DW   D9BB    ; a kazetta rutinok ugrótáblájának címe
C569  DW   0000
C56B  DW   C4AC    ; a kernel rutinok ugrótáblájának címe
```

A HI_MEM átlépést ellenörzö rutin:

```
C56D  EX DE,HL
C56E  PUSH HL
C56F  PUSH DE
C570  PUSH BC
C571  PUSH HL
C572  PUSH DE
C573  EX   DE,HL
C574  LD   HL,(0B19)   ; HL-be a HI_MEM értéke
C577  OR   A           ; a CARRY flag törlése
C578  SBC  HL,DE
C57A  LD   A,FA        ; A-ba a hibakód
C57C  POP  DE
C57D  POP  HL
C57E  RET  C           ; HI_MEM átlépés történne: hibás blokk
                         I/O paraméterezés
C57F  LD   C,(HL)      ; C-be a következö cím tartalma
C580  EX   DE,HL
```

```
C581  CALL C58E        ; "CALL (HL)"
C584  POP  BC
C585  POP  DE
C586  POP  HL
C587  OR   A
C588  RET  NZ
C589  CPI              ; HL növelése 1-gyel és BC csökkenté-
                         se 1-gyel
C58B  RET  PO          ; BC tartalma 00H lett: nincs több
                         HI_MEM átlépés ellenörzés
C58C  JR   E0          ; ugrás a C56EH címre
C58E  JP   (HL)
```

A HI_MEM átlépést ellenörzö rutin:

```
C58F  PUSH HL
C590  PUSH DE
C591  PUSH BC
C592  PUSH HL
C593  PUSH DE
C594  LD   HL,(0B19)   ; HL-be HI_MEM értéke
C597  OR   A
C598  SBC  HL,DE
C59A  LD   A,FA        ; A-ba a hibakód
C59C  POP  DE
C59D  POP  HL
C59E  RET  C           ; HI_MEM átlépés történne: hibás blokk
                         I/O paraméterezés
C59F  CALL C58E        ; "CALL (HL)"
C5A2  EX   AF,AF'
C5A3  LD   A,C
C5A4  POP  BC
C5A5  POP  DE
C5A6  POP  HL
C5A7  EX   AF,AF'
C5A8  OR   A
C5A9  RET  NZ
C5AA  EX   AF,AF'
C5AB  LD   (DE),A
C5AC  XOR  A
C5AD  EX   DE,HL
C5AE  CPI              ; HL növelése 1-gyel és a BC csökkenté-
                         se 1-gyel
C5B0  EX   DE,HL
C5B1  RET  PO          ; BC tartalma 00H lett: nincs több
                         HI_MEM átlépés ellenörzés
C5B2  JR   DB          ; ugrás a C58FH címre
```

32 - 127 kódú karaktereket definiáló táblázat:

```
C5B4  DB   00 00 00 00 00 00 00 00 00 00   ; space
C5BE  DB   00 18 18 18 18 18 00 18 00 00   ; !
C5C8  DB   00 36 36 36 00 00 00 00 00 00   ; "
C5D2  DB   00 36 36 7F 36 7F 36 36 00 00   ; #
C5DC  DB   00 18 3E 58 3C 1A 7C 18 00 00   ; $
C5E6  DB   00 60 66 0C 18 30 66 06 00 00   ; %
C5F0  DB   00 10 28 28 30 54 48 34 00 00   ; &
C5FA  DB   00 18 18 30 00 00 00 00 00 00   ; '
C604  DB   00 0C 18 30 30 30 18 0C 00 00   ; (
C60E  DB   00 30 18 0C 0C 0C 18 30 00 00   ; )
C618  DB   00 00 10 54 38 38 54 10 00 00   ; *
C622  DB   00 00 18 18 7E 18 18 00 00 00   ; +
C62C  DB   00 00 00 00 00 00 18 18 30 00   ; ,
C636  DB   00 00 00 00 7C 7C 00 00 00 00   ; -
C640  DB   00 00 00 00 00 00 18 18 00 00   ; .
C64A  DB   00 00 06 0C 18 30 60 00 00 00   ; /
C654  DB   00 3C 66 6E 7E 76 66 3C 00 00   ; 0
C65E  DB   00 18 38 18 18 18 18 18 00 00   ; 1
C668  DB   00 3C 66 06 1C 30 60 7E 00 00   ; 2
C672  DB   00 7E 06 0C 1C 06 46 3C 00 00   ; 3
C67C  DB   00 0C 1C 2C 4C 7E 0C 0C 00 00   ; 4
C686  DB   00 7E 60 7C 06 06 46 3C 00 00   ; 5
C690  DB   00 3C 60 60 7C 66 66 3C 00 00   ; 6
C69A  DB   00 7E 06 0C 18 30 60 60 00 00   ; 7
C6A4  DB   00 3C 66 66 3C 66 66 3C 00 00   ; 8
C6AE  DB   00 3C 66 66 3E 06 0C 38 00 00   ; 9
C6B8  DB   00 00 00 18 18 00 18 18 00 00   ; :
C6C2  DB   00 00 00 18 18 00 18 18 30 00   ; ;
C6CC  DB   00 06 0C 18 30 18 0C 06 00 00   ; <
C6D6  DB   00 00 00 7C 00 7C 00 00 00 00   ; =
C6E0  DB   00 30 18 0C 06 0C 18 30 00 00   ; >
C6EA  DB   00 3C 66 06 0C 18 00 18 00 00   ; ?
C6F4  DB   00 3E 63 67 6B 6F 60 3C 00 00   ; @
C6FE  DB   00 1C 3E 63 63 7F 63 63 00 00   ; A
C708  DB   00 7E 63 63 7E 63 63 7E 00 00   ; B
C712  DB   00 3E 63 60 60 60 63 3E 00 00   ; C
C71C  DB   00 7E 33 33 33 33 33 7E 00 00   ; D
C726  DB   00 7E 60 60 7C 60 60 7E 00 00   ; E
C730  DB   00 7E 60 60 7C 60 60 60 00 00   ; F
C73A  DB   00 3E 63 60 60 67 63 3E 00 00   ; G
C744  DB   00 63 63 63 7F 63 63 63 00 00   ; H
C74E  DB   00 3C 18 18 18 18 18 3C 00 00   ; I
C758  DB   00 06 06 06 06 66 66 3C 00 00   ; J
```

```
C762  DB   00 63 66 6C 78 6C 66 63 00 00   ; K
C76C  DB   00 60 60 60 60 60 60 7E 00 00   ; L
C776  DB   00 63 77 6B 63 63 63 63 00 00   ; M
C780  DB   00 66 66 76 6E 66 66 66 00 00   ; N
C78A  DB   00 3E 63 63 63 63 63 3E 00 00   ; O
C794  DB   00 7E 63 63 7E 60 60 60 00 00   ; P
C79E  DB   00 3E 63 63 63 6B 67 3E 01 00   ; Q
C924  DB   00 00 00 66 3C 18 3C 66 00 00   ; x
C92E  DB   00 00 00 66 66 66 66 3E 06 3C   ; y
C938  DB   00 00 00 7E 0C 18 30 7E 00 00   ; z
C942  DB   00 0E 18 18 70 18 18 0E 00 00   ; {
C94C  DB   00 18 18 18 00 18 18 18 00 00   ; |
C956  DB   00 70 18 18 0E 18 18 70 00 00   ; }
C960  DB   00 00 00 33 6B 66 00 00 00 00   ; ~
C96A  DB   00 7E 7E 7E 7E 7E 7E 7E 00 00   ; alap kurzor
```

A video rutinok ugrótáblája:

```
C974  DB   0D      ; a rutinok száma
C975  DW   C9A0    ; a VID_IRQ rutin címe
C977  DW   CC94    ; a VID_CHOUT rutin címe
C979  DW   CC86    ; a VID_BKOUT rutin címe
C97B  DW   CF4B    ; a BTEXT rutin címe
C97D  DW   C9F4    ; a VMODE rutin címe
C97F  DW   CA49    ; a CLS rutin címe
C981  DW   CADA    ; a BABS rutin címe
C983  DW   CAD7    ; a BREL rutin címe
C985  DW   CBF3    ; a BON rutin címe
C987  DW   CBFF    ; a BOFF rutin címe
C989  DW   CD48    ; a FILL rutin címe
C98B  DW   CF2C    ; a DEFC rutin címe
C98D  DW   CA38    ; a PAL rutin címe
```

A video RAM belapozása a video rutinok számára, az eredeti lapkiosztás visszaállítása a video RAM használata után:

```
C98F  POP  HL          ; visszatérési cím HL-be
C990  LD   A,(0003)
C993  PUSH AF          ; az eredeti lapkiosztás elmentése
C994  LD   A,50
C996  LD   (0003),A
C999  OUT  02,A        ; memórialapozás: U U V S lapkiosztás
C99B  PUSH HL          ; visszatérési cím a STACK-be
C99C  LD   HL,C9A1     ; visszalapozórutin címe HL-be
C99F  EX   SP,HL       ; visszatérési cím HL-be, a visszalapo-
```

```
                                pozó rutin címe a STACK-be
  C9A0  JP    (HL)          ; RET
  C9A1  EX    AF,AF'
  C9A2  POP   AF
  C9A3  LD    (0003),A
  C9A6  OUT   02,A          ; az eredeti memórialap kiosztás vissza-
                              állítása
  C9A8  EX    AF,AF'
  C9A9  RET                 ; a rutin vége

Ugrásicím táblázatból keresö rutin (HL-ben a táblázat címe):
  C9AA  ADD   A,A           ; A-ba a keresett elem sorszámának két-
                              szerese
  C9AB  LD    E,A
  C9AC  LD    D,00          ; DE-be A értéke töltödik
  C9AE  ADD   HL,DE         ; HL-be a táblázat keresett elemének címe
  C9AF  LD    E,(HL)
  C9B0  INC   HL
  C9B1  LD    D,(HL)        ; DE-be kerül az ugrótábla keresett e-
                              leme
  C9B2  RET                 ; a rutin vége

Az X logikai koordináta vizsgálata (BC-ben X értéke):
  C9B3  LD    HL,03FF       ; HL-be X maximális értéke (1023)
  C9B6  OR    A             ; a CARRY flag törlése
  C9B7  SBC   HL,BC         ; ha X nagyobb a megengedettnél, a
                              CARRY flag átvitelt jelez
  C9B9  RET                 ; a rutin vége

Az Y logikai koordináta vizsgálata (DE-ben az Y értéke):
  C9BA  LD    HL,03BF       ; HL-be Y maximális értéke (959)
  C9BD  OR    A
  C9BE  SBC   HL,DE         ; ha Y nagyobb a megengedettnél, a
                              CARRY flag átvitelt jelez
  C9C0  RET                 ; a rutin vége

A fizikai koordináta meghatározása a logikai koordinátából (HL)
a grafikus felbontás szerin:
  C9C1  LD    A,(0B73)      ; A-ba a GR_MODE
  C9C4  LD    B,A
  C9C5  INC   B
HL osztása 2-vel:
  C9C6  SRL   H
  C9C8  RR    L
```

```
  C9CA  DJNZ FA          ; ha B>0, ugrás a C9C6H címre
  C9CC  RET              ; a rutin vége

Változók elökészítése a vonalat húzó rutinok számára:
  C9CD  CALL CB6D        ; a vonaltipus byte elökészítése
  C9D0  CALL CC05        ; a paper byte elökészítése
  C9D3  LD   (0B93),A    ; a paper byte eltárolása
  C9D6  LD   DE,0040
  C9D9  CALL CB8F        ; a vonalkeresztezési módtól függö rutin
                           cím meghatározása (IY-ban)
  C9DC  CALL CC0D        ; az ink byte meghatározása
  C9DF  LD   (0B94),A    ; az ink byte eltárolása
  C9E2  LD   A,(0B75)    ; A-ba a sugárpozíció video RAM-beli cí-
                           mének "tört byte"-ja
  C9E5  LD   C,A
  C9E6  LD   HL,(0B76)   ; HL-be az aktuális sugárpozíció címe
                           a video RAM-ban
  C9E9  RET              ; a rutin vége


  C9EA  DB   00 50 44 41   ; a paletta kódok reset után a szín-
                             kikapcsoló bekapcsolt állapotánál
  C9EE  DB   00 55 50 44   ; a paletta kódok reset után a szín-
                             kikapcsoló kikapcsolt állapotánál

  C9F2  LD   C,01

A V_MODE rutin (C-ben a GR_MODE értéke):
  C9F4  LD   A,C         ; A-ba a GR_MODE
  C9F5  CP   03
  C9F7  LD   A,F7        ; A-ba a hibakód
  C9F9  RET  NC          ; a rutin vége hiba esetén: nem létezö
                           grafikus mód
  C9FA  XOR  A
  C9FB  LD   (0B4B),A    ; L_MODE törlése
  C9FE  LD   (0B4E),A    ; PAPER törlése
  CA01  LD   (0B50),A    ; V_FLAG törlése
  CA04  INC  A
  CA05  LD   (0B4C),A    ; L_STILE=1
  CA08  CP   C
  CA09  JR   NC,0A       ; 2 és 4 színü grafikus mód esetén ugrás
                           a CA15H címre
A színkikapcsoló állásának beolvasása a CARRY flag-be:
  CA0B  IN   A,59
  CA0D  RLCA
```

```
  CA0E  RLCA

  CA0F  LD    A,0F        ; INK fekete-fehér színkikapcsoló állás-
                            nál, 16 színü módban
  CA11  JR    NC,02       ; színkikapcsoló kikapcsolva, ugrás
  CA13  LD    A,0C        ; INK szines 16 színü módban
  CA15  LD    (0B4D),A    ; INK feltöltése
  CA18  LD    A,C
  CA19  LD    (0B73),A    ; GR_MODE elmentése
  CA1C  PUSH  BC
  CA1D  CALL  CA49        ; CLS rutin
A PORT06 változó törlése,írása, és a 6-os port írása:
  CA20  POP   BC
  CA21  LD    A,C
  CA22  LD    HL,0B13
  CA25  XOR   (HL)
  CA26  AND   FC
  CA28  XOR   C
  CA29  LD    (HL),A
  CA2A  OUT   06,A
A színkikapcsoló vizsgálata
  CA2C  IN    A,59
  CA2E  RLCA
  CA2F  RLCA
A színkikapcsoló állásától függöen a paletta regiszterek feltöl-
tése:
  CA30  LD    DE,C9EE     ; palettakódok címe fekete-fehér módban
  CA33  JR    NC,03
  CA35  LD    DE,C9EA     ; DE-ben a palettakódok címe szines mód-
                            ban

A PAL rutin:
  CA38  LD    A,(DE)
  CA39  OUT   60,A
  CA3B  INC   DE
  CA3C  LD    A,(DE)
  CA3D  OUT   61,A
  CA3F  INC   DE
  CA40  LD    A,(DE)
  CA41  OUT   62,A
  CA43  INC   DE
  CA44  LD    A,(DE)
  CA45  OUT   63,A
  CA47  XOR   A           ; A-ba 00H: hibátlan futás
  CA48  RET               ; a V_MODE és a PAL rutin vége
```

A CLS rutin:

```
  CA49  CALL C98F        ; videomemória belapozása a P2 lapra
  CA4C  CALL CFD4        ; az editor munkaterület alaphelyzetbe
                           állítása
  CA4F  CALL CC05        ; a paper byte elöállítása az A regisz-
                           terbe
```

A grafikus képernyö törlése (a video RAM alsó 15 KB-jának feltöltése a paper byte-tal):

```
  CA52  LD   HL,8000     ; HL-be a video RAM kezdete
  CA55  LD   DE,8001
  CA58  LD   (HL),A
  CA59  LD   BC,3BFF     ; BC-be a video RAM mérete-1
  CA5C  LDIR

  CA5E  CALL CBFF        ; a BOFF rutin
  CA61  LD   B,A
  CA62  LD   C,A         ; BC-be 00H
  CA63  LD   D,A
  CA64  LD   E,A         ; DE-be 00H
```

A toll pozíció beállítása:

```
  CA65  LD   (0B7E),DE   ; Y logikai koordináta beállítása
  CA69  LD   (0B7C),BC   ; X logikai koordináta beállítása
  CA6D  LD   HL,(0B7C)   ; HL-be az X logikai koordináta
  CA70  PUSH HL
  CA71  LD   B,04
  CA73  CALL C9C6        ; HL=HL/16
  CA76  EX   SP,HL       ; HL-be az X logikai koordináta
  CA77  CALL C9C1        ; HL-be az X fizikai koordináta
  CA7A  LD   (0B78),HL   ; az X fizikai koordináta elmentése
```

Annak a byte-nak a meghatározása, mely megmutatja, hogy adott X fizikai koordinátájú (HL-ben) pont a video RAM-nak a pontot tartalmazó byte-ja által tárolt pontok közül az adott pont éppen melyik:

```
  CA7D  LD   C,A         ; C-be a GR_MODE
  CA7E  LD   A,L
  CA7F  LD   HL,CAB4     ; "tört byte" meghatározásához szükséges
                           táblázat címe a HL-be
  CA82  ADD  HL,BC       ; HL-be az adott grafikus felbontáshoz
                           tartozó elemének címe
  CA83  AND  07          ; A-ba az (X fizikai koordináta)/8-nak
                           a maradéka kerül
  CA85  INC  A
  CA86  LD   B,A
  CA87  LD   A,(HL)
  CA88  LD   E,0F
```

```
***
  CA89  RRCA
***
  CA8A  DJNZ FD          ; ha B>0, ugrás a CA89H címre
  CA8C  LD   (0B75),A    ; a keresett byte elmentése. Az adott
                           byte 1 értékü bitjei jelzik, hogy a
                           grafikus felbontástól függöen az egy
                           byte által ábrázolható pontok közül
                           a "mi" pontunk melyik
  CA8F  LD   DE,(0B7E)   ; DE-be az Y logikai koordináta
  CA93  LD   HL,03BF     ; HL-be Y maximális értéke
  CA96  XOR  A
  CA97  SBC  HL,DE       ; HL-be a bal alsó sarokhoz viszonyított
                           Y logikai koordináta
  CA99  LD   B,02
  CA9B  CALL C9C6        ; HL-be a bal alsó sarokhoz viszonyított
                           Y fizikai koordináta
  CA9E  LD   (0B7A),HL   ; a bal alsó sarokhoz viszonyított
                           Y fizikai koordináta elmentése
```

Az adott logikai koordinátájú ponthoz tartozó video RAM cím meghatározása:

```
  CAA1  LD   H,L
  CAA2  SRL  H
  CAA4  RRA
  CAA5  SRL  H
  CAA7  RRA
  CAA8  LD   L,A         ; HL=HL*64
  CAA9  POP  BC          ; BC=(X logikai koordináta)/16
  CAAA  ADD  HL,BC       ; HL-be a pontot magába foglaló byte
                           címe -1
  CAAB  LD   DE,8000
  CAAE  ADD  HL,DE
  CAAF  LD   (0B76),HL   ; a keresett video RAM cím elmentése
  CAB2  XOR  A           ; a rutin hiba nélkül futott
  CAB3  RET              ; a CLS rutin vége,a BABS,BREL ruti-
                           nok vége kikapcsolt rajzoló sugárnál,
                           VID_CHOUT rutin vége

  CAB4  DB   80          ; 2 színü grafikus módhoz tartozó byte
  CAB5  DB   88          ; 4 színü grafikus módhoz tartozó byte
  CAB6  DB   AA          ; 16 színü grafikus módhoz tartozó byte
```

A vonalhúzás során egy pont kirajzolása a beállított rendszerváltozók szerint (HL a pont címe a video RAM-ban, C a "tört byte"):

```
  CAB7  JR   NC,0C      ; PAPER színű pont következik, ugrás a
                          CAC5H címre
  CAB9  AND  01
  CABB  JR   Z,03
  CABD  RET
  CABE  JR   NC,08      ; PAPER színű pont következik, ugrás a
                          CAC8H címre
  CAC0  LD   A,(0B94)   ; INK színű pont következik
  CAC3  JP   (IY)       ; ugrás a vonalkeresztezési mód szabta
                          címre
  CAC5  AND  02
  CAC7  RET  NZ
  CAC8  LD   A,(0B93)
  CACB  JP   (IY)       ; ugrás a vonalkeresztezési mód szabta
                          címre
```

AND vonalkeresztezési mód esetén egy pont kirajzolása:

```
  CACD  AND  (HL)
```

"Felülírás" vonalkeresztezési mód esetén egy pont kirajzolása:

```
  CACE  XOR  (HL)
```

XOR vonalkeresztezési mód esetén egy pont kirajzolása:

```
  CACF  AND  C
  CAD0  XOR  (HL)
  CAD1  LD   (HL),A
  CAD2  RET             ; a pont rajzoló rutin vége
```

OR vonalkeresztezési mód esetén egy pont kirajzolása:

```
  CAD3  AND  C
  CAD4  OR   (HL)
  CAD5  LD   (HL),A
  CAD6  RET             ; a pont rajzoló rutin vége
```

A BREL rutin:

```
  CAD7  CALL CC7A       ; a relatív koordináták alapján az ab-
                          szolut koordináták meghatározása: BC-
                          be X, DE-be Y
```

A BABS rutin:

```
  CADA  CALL C98F       ; a video RAM belapozása a P2 lapra
  CADD  LD   A,F9       ; A-ba a hibakód
  CADF  CALL C9B3       ; az X koordináta létezésének vizsgálata
  CAE2  CALL NC,C9BA    ; ha X jó, az Y jóságának vizsgálata
  CAE5  RET  C          ; valamelyik koordináta nem megengedett,
                          visszatérés a rutinból hibával
```

```
CAE6  LD   A,(0B74)    ; a B_STAT az A-ba
CAE9  OR   A
CAEA  JP   Z,CA65      ; a rajzoló sugár kikapcsolva, ugrás
CAED  EXX
CAEE  LD   BC,(0B78)   ; BC'-be az elözö X fizikai koordináta
CAF2  LD   DE,(0B7A)   ; DE'-be az elözö Y fizikai koordináta
                         (a bal alsó sarokhoz viszonyítva)
CAF6  PUSH BC
CAF7  PUSH DE
CAF8  CALL C9CD        ; a vonalhúzáshoz szükséges változók e-
                         lökészítése
CAFB  EXX
CAFC  CALL CA65        ; az új tollpozíciót meghatározó válto-
                         zók átírása
CAFF  LD   BC,(0B78)   ; BC-be az új X fizikai koordináta
CB03  LD   DE,(0B7A)   ; DE-be az új Y fizikai koordináta (a
                         bal alsó sarokhoz viszonyítva)
CB07  XOR  A
CB08  EX   DE,HL
CB09  POP  DE          ; DE-be a régi Y fizikai koordináta
CB0A  CALL CB61        ; HL=abs(HL-DE), ha a különbség negatív,
                         A=1, különben 0: az új és a régi Y fi-
                         zikai koordináta különbsége HL-be
CB0D  POP  DE          ; DE-be a régi X fizikai koordináta
CB0E  PUSH HL
CB0F  LD   H,B
CB10  LD   L,C         ; HL-be az új X fizikai koordináta
CB11  CALL CB61        ; HL=abs(HL-DE), ha a különbség negatív,
                         A=2*A+1, különben 2*A: az új és a régi
                          X fizikai koordináta különbsége HL-be
CB14  PUSH HL
CB15  LD   HL,CBDB     ; HL-be a "vonalhúzás iránya" táblázat
                         címe
CB18  CALL C9AA        ; a táblázat megfelelö eleme DE-be
CB1B  LD   (0B86),DE
CB1F  POP  HL          ; HL-be a régi és az új X fizikai koor-
                         dináta különbsége
CB20  POP  DE          ; DE-be a régi és az új Y fizikai koor-
                         dináta különbsége
CB21  PUSH HL
CB22  SBC  HL,DE       ; ha Y változik nagyobbat, CARRY=1
CB24  POP  HL          ; HL-be X változása
CB25  JR   C,01
CB27  EX   DE,HL       ; ha X változik nagyobbat, DE-be X vál-
                         tozása, HL-be Y változása
```

```
  CB28  LD   B,H
  CB29  LD   C,L          ; különben BC-be X változása
  CB2A  PUSH DE
  CB2B  ADC  A,00         ; A=A+CARRY flag
HL=HL-DE/2:
  CB2D  SBC  HL,DE
  CB2F  PUSH HL
  CB30  SRL  D
  CB32  RR   E
  CB34  ADD  HL,DE
  CB35  PUSH HL

  CB36  LD   HL,CBE3      ; HL-be a "vonalhúzás iránya II." táblá-
                            zat címe
  CB39  CALL C9AA         ; a táblázat megfelelö eleme DE-be
  CB3C  LD   (0B84),DE
  CB40  POP  HL
  CB41  POP  DE
  CB42  EXX
  CB43  EX   SP,HL        ; HL-be a nagyobbat változó fizikai ko-
                            ordináta változása, a STACK-be a pont
                            címe a video RAM-ban
  CB44  LD   A,H          ; A=0, ha a változás < 256, különben A=1
  CB45  LD   B,L
  CB46  INC  B
  CB47  INC  A
  CB48  EX   AF,AF'
  CB49  POP  HL           ; HL-be a pont címe a video RAM-ban
  CB4A  JR   04           ; ugrás a CB50H címre
  CB4C  EX   AF,AF'
  CB4D  CALL CBA8         ; a következö kirajzolandó pont video
                            RAM-beli címének és a cím "tört byte"-
                            jának meghatározása
  CB50  LD   A,(0B83)     ; A-ba a vonal képét meghatározó byte
  CB53  RLCA              ; a byte módosítása
  CB54  LD   (0B83),A     ; a módosított byte elmentése
  CB57  CALL CABE         ; a vonal soronkövetkezö pontjának a ki-
                            rajzolása
  CB5A  DJNZ F1           ; ha van még kirajzolandó pont, ugrás a
                            CB4DH címre
  CB5C  EX   AF,AF'
  CB5D  DEC  A            ; 256-nál kevesebb pontot kellett kiraj-
                            zolni?
  CB5E  JR   NZ,EC        ; ha nem, ugrás a CB4CH címre: még 255
                            pont kirajzolása
```

```
  CB60  RET             ; a BABS és BREL rutin vége, ha a rajzo-
                          ló sugár be van kapcsolva

HL=abs(HL-DE), CARRY=1, ha HL<DE, különben 0:
  CB61  SBC  HL,DE
  CB63  JR   NC,06
  CB65  EX   DE,HL
  CB66  LD   HL,0001
  CB69  SBC  HL,DE
  CB6B  RLA             ; A=CARRY
  CB6C  RET             ; a rutin vége

A vonaltipus minta-byte meghatározása és elmentése:
  CB6D  LD   A,(0B4C)   ; A-ba L_STYLE
  CB70  DEC  A
  CB71  AND  0F
  CB73  LD   C,A
  CB74  LD   HL,CB7F
  CB77  LD   B,00
  CB79  ADD  HL,BC
  CB7A  LD   A,(HL)
  CB7B  LD   (0B83),A
  CB7E  RET             ; a rutin vége

  CB7F  DB   FF AA CC EE 88 DA E4 F6   ; a vonaltipusok képét
  CB87  DB   FA FE FC F8 F0 EA FF FF   ; definiáló byte-ok

A vonalkeresztezési módtól függő pontrajzoló rutin címének meg-
határozása:
  CB8F  PUSH DE
  CB90  LD   A,(0B4B)   ; A-ba L_MODE
  CB93  AND  03
  CB95  LD   HL,CBA0
  CB98  CALL C9AA
  CB9B  PUSH DE
  CB9C  POP  IY         ; a keresett cím IY-ban
  CB9E  POP  DE
  CB9F  RET             ; a rutin vége

A vonalkeresztezési mód ugrótáblája

  CBA0  DW   CACE       ; felülírás
  CBA2  DW   CAD3       ; OR
  CBA4  DW   CACD       ; AND
  CBA6  DW   CACF       ; XOR
```

```
A következő kirajzolandó pont címének meghatározása vonalhúzás-
kor:
  CBA8  EXX
  CBA9  BIT  7,H
  CBAB  JR   Z,08          ; ha H 7. bitje 0, ugrás a CBB5H címre
A 2.1 rutin:
***
  CBAC  EX   AF,AF'
***
  CBAD  ADD  HL,BC
  CBAE  EXX
  CBAF  LD   IX,(0B84)
  CBB3  JP   (IX)
  CBB5  ADD  HL,DE
  CBB6  EXX
  CBB7  LD   IX,(0B86)
  CBBB  JP   (IX)
A 3. rutin:
  CBBD  OR   A
  CBBE  SBC  HL,DE
Az 1.0 és a 3.0 rutin:
  CBC0  RLC  C
  CBC2  RET  NC
  CBC3  DEC  L
  CBC4  RET
A 2. rutin:
  CBC5  RRC  C
  CBC7  JR   NC,01
  CBC9  INC  L
A 3.1 rutin:
  CBCA  OR   A
  CBCB  SBC  HL,DE
  CBCD  RET
Az 1. rutin:
  CBCE  RLC  C
  CBD0  JR   NC,01
  CBD2  DEC  L
A 0.1 és a 1.1 rutin:
  CBD3  ADD  HL,DE
  CBD4  RET
A 0. rutin:
  CBD5  ADD  HL,DE
A 0.0 és a 2.0 rutin:
  CBD6  RRC  C
  CBD8  RET  NC
```

```
  CBD9  INC  L
  CBDA  RET

A "vonalhúzás iránya" ugrótábla:
  CBDB  DW   CBD5        ; 0. rutin címe: X és Y is nö
  CBDD  DW   CBCE        ; 1. rutin címe: X csökken és Y nö
  CBDF  DW   CBC5        ; 2. rutin címe: X nö és Y csökken
  CBE1  DW   CBBD        ; 3. rutin címe: X és Y is csökken

A "vonalhúzás iránya II:" ugrótábla:
  CBE3  DW   CBD6        ; 0.0 rutin címe: X és Y is nö, X vál-
                           tozása nagyobb
  CBE5  DW   CBD3        ; 0.1 rutin címe: Y változása nagyobb

  CBE7  DW   CBC0        ; 1.0 rutin címe: X csökken és Y nö, X
                           változása nagyobb
  CBE9  DW   CBD3        ; 1.1 rutin címe: Y változása nagyobb

  CBEB  DW   CBD6        ; 2.0 rutin címe: X nö és Y csökken, X
                           változása nagyobb
  CBED  DW   CBAC        ; 2.1 rutin címe: Y változása nagyobb

  CBEF  DW   CBC0        ; 3.0 rutin címe: X és Y csökken, X vál-
                           tozása nagyobb
  CBF1  DW   CBCA        ; 3.1 rutin címe: Y változása nagyobb


A BON rutin:
  CBF3  CALL C98F        ; a video RAM belapozása a P2 lapra
  CBF6  CALL C9D9        ; a rendszerváltozók elökészítése a
                           toll pozíciójában lévö pont kirajzo-
                           lásához
  CBF9  CALL CAC0        ; a tollpozícióban levö pont kirajzolása
  CBFC  LD   A,FF
  CBFE  LD   H,AF

A BOFF rutin:
***
  CBFF  XOR  A
***
  CC00  LD   (0B74),A    ; a B_STAT rendszerváltozó átírása
  CC03  XOR  A
  CC04  RET              ; a BON és a BOFF rutin vége
```

```
A paper és az ink byte elökészítése rajzoláshoz:
  CC05  LD   A,(0B4E)    ; A-ba INK
  CC08  CALL CC10        ; a byte elökészítése a GR_MODE-tól
                           függöen
  CC0B  LD   B,A
  CC0C  RET              ; az ink byte rutin vége
  CC0D  LD   A,(0B4D)    ; A-ba PAPER
  CC10  LD   HL,(0B73)   ; L-be GR_MODE
  CC13  SRL  L
  CC15  JR   C,09        ; 4 színü mód, ugrás a CC20H címre
  CC17  SRL  L
  CC19  JR   C,16        ; 16 színü mód, ugrás a CC31H címre
  CC1B  AND  01
  CC1D  RRA
  CC1E  SBC  A,A
  CC1F  RET              ; rutin vége 2 színü módban
  CC20  AND  03
  CC22  RRA
  CC23  RL   L
  CC25  RRA
  CC26  SBC  A,A
  CC27  RR   L
  CC29  RRA
  CC2A  SRA  A
  CC2C  SRA  A
  CC2E  SRA  A
  CC30  RET              ; a rutin vége 4 színü módban
  CC31  AND  0F
  CC33  LD   H,04
  CC35  RRA
  CC36  RR   L
  CC38  SRA  L
  CC3A  DEC  H
  CC3B  JR   NZ,F8       ; ugrás a CC35H címre, ha H>0
  CC3D  LD   A,L
  CC3E  RET              ; a rutin vége 16 színü módban

Adott képpont byte elökészítése a FILL rutinhoz:
  CC3F  CALL C9E2        ; HL-be és C-be az aktuális tollpozíció
                           video RAM-beli címe
  CC42  LD   A,(0B73)    ; A-ba GR_MODE
  CC45  LD   B,A
  CC46  LD   A,C
  CC47  AND  (HL)
  CC48  SRL  B
```

```
CC4A  JR   C,0A        ; 4 színü mód, ugrás a CC56H címre
CC4C  SRL  B
CC4E  JR   C,15        ; 16 színü mód, ugrás a CC65H címre
CC50  ADD  A,FF
CC52  RLA
CC53  AND  01
CC55  RET              ; a rutin vége 2 színü módban
CC56  ADD  A,F0
CC58  RL   C
CC5A  AND  0F
CC5C  ADD  A,FF
CC5E  RLA
CC5F  RR   C
CC61  RLA
CC62  AND  03
CC64  RET              ; a rutin vége 4 színü módban
CC65  ADD  A,C0
CC67  RLA
CC68  AND  7F
CC6A  ADD  A,E0
CC6C  RLA
CC6D  AND  3F
CC6F  ADD  A,F0
CC71  RLA
CC72  AND  1F
CC74  ADD  A,F8
CC76  RLA
CC77  AND  0F
CC79  RET              ; a rutin vége 16 színü módban
```

A relatív koordinátákból az abszolut koordináták meghatározása (BC-ben a relatív X, DE-ben a relatív Y logikai koordináta):

```
CC7A  LD   HL,(0B7C)   ; HL-be a régi X logikai koordináta
CC7D  ADD  HL,BC
CC7E  LD   B,H
CC7F  LD   C,L         ; BC-be az új abszolut X logikai koor-
                         dináta
CC80  LD   HL,(0B7E)   ; HL-be a régi Y logikai koordináta
CC83  ADD  HL,DE
CC84  EX   DE,HL       ; DE-be az új abszolut Y logikai koor-
                         dináta
CC85  RET              ; a rutin vége
```

A VID_BKOUT rutin:

```
CC86  EX   DE,HL
```

```
  CC87  PUSH HL
  CC88  PUSH BC          ; az input paraméterek elmentése
  CC89  LD   C,(HL)      ; C-be a kiírandó karakter
  CC8A  CALL CC94        ; a karakter kiíratása a VID_CHOUT ru-
                           tinnal
  CC8D  POP  BC
  CC8E  POP  HL
  CC8F  CPI              ; HL=HL+1, BC=BC-1, ha BC=0, a P/V flag
                           1 lesz
  CC91  RET  PO          ; a VID_BKOUT rutin vége
  CC92  JR   F3          ; ugrás a CC87H címre

A VID_CHOUT rutin:
  CC94  CALL C98F        ; a video RAM belapozása a P2 lapra
  CC97  LD   A,C         ; A-ba a kiírandó karakter
  CC98  CP   20
  CC9A  JR   NC,0C       ; ugrás a CCA8H címre, ha vezérlö karak-
                           ter
  CC9C  CP   0A
  CC9E  JP   Z,CD31      ; ugrás, ha "soremelés" karakter
  CCA1  CP   0D
  CCA3  JP   Z,CD2B      ; ugrás, ha "kocsi vissza" karakter
  CCA6  XOR  A
  CCA7  RET              ; a VID_CHOUT vége, ha vezérlö karakter
                           a kiíratandó
  CCA8  CP   E0
  CCAA  LD   A,00
  CCAC  RET  NC          ; a VID_CHOUT rutin vége, ha a kiírandó
                           karakter kódja >223
  CCAD  EXX
  CCAE  LD   DE,FFDC     ; -36 relatív Y koordináta (9 kép sor)
  CCB1  LD   A,(0B73)    ; A-ba a GR_MODE
  CCB4  LD   B,A
  CCB5  LD   A,01
  CCB7  INC  B
  CCB8  ADD  A,A
  CCB9  DJNZ FD
  CCBB  LD   (0B82),A
  CCBE  LD   L,A
  CCBF  ADD  A,A
  CCC0  ADD  A,L
  CCC1  ADD  A,A
  CCC2  ADD  A,L
  CCC3  LD   C,A         ; BC-be a 8 fizikai koordinátának megfe-
                           lelö relatív logikai koordináta
  CCC4  XOR  A
```

Annak ellenörzése, hogy van-e elegendö hely a karakter kiírásához:

```
  CCC5  CALL CC7A       ; a kiírandó karakter jobb alsó sarkának logikai koordinátái
  CCC8  CALL C9BA       ; Y ellenörzése
  CCCB  RET  C          ; a VID_CHOUT rutin vége, ha a képernyö alján nincs elegendö hely
  CCCC  CALL C9B3       ; X ellenörzése
  CCCF  JR   NC,06      ; az adott sorban elfér a karakter, ugrás a CCD7H címre
  CCD1  CALL CD2B       ; "kocsi vissza"
  CCD4  CALL CD31       ; "soremelés"
  CCD7  CALL C9D0       ; a megfelelö rendszerváltozók elökészítése
  CCDA  EXX
  CCDB  LD   DE,C474    ; DE-be 32-127 kódú karakterek karaktergenerátorának címe
  CCDE  BIT  7,C
  CCE0  JR   Z,03       ; ugrás, ha a karakter kódja > 127
  CCE2  LD   DE,0740    ; DE-be a definiálható karakterek karaktergenerátora
  CCE5  RES  7,C        ; a karakter legfelsö bitjének törlése
  CCE7  LD   B,00
  CCE9  LD   H,B
  CCEA  LD   L,C
  CCEB  ADD  HL,HL
  CCEC  ADD  HL,HL
  CCED  ADD  HL,BC
  CCEE  ADD  HL,HL
  CCEF  ADD  HL,DE      ; a karakter karaktermátrix címe HL-be
  CCF0  LD   B,0A       ; 10 sorból áll egy  karakter
  CCF2  LD   A,(0B50)   ; A-ba V_FLAG
  CCF5  LD   E,A
  CCF6  DEC  HL
Következö sor kiíratása:
  CCF7  INC  HL
  CCF8  LD   C,(HL)
  CCF9  EXX
  CCFA  LD   B,08
  CCFC  JR   03
Egy sor kiíratása:
  CCFE  CALL CBD6
  CD01  EXX
  CD02  RL   C
  CD04  LD   A,E
```

```
  CD05  EXX
  CD06  CALL CAB7        ; a karakter egy pontjának kirajzolása
  CD09  DJNZ F3          ; ugrás a CCFEH címre, ha a karaktermát-
                           rix adott sora még nincs kiírva
  CD0B  CALL CBD3
  CD0E  LD   B,07
  CD10  CALL CBC0
  CD13  DJNZ FB          ; ugrás a CD10H címre
  CD15  EXX
  CD16  DJNZ DF          ; ha van még kiíratlan sora a karakter-
                           mátrixnak, ugrás a CCF7H címre
  CD18  LD   A,(0B82)
  CD1B  ADD  A,A
  CD1C  ADD  A,A
  CD1D  ADD  A,A
  CD1E  LD   C,A         ; BC-be X=8 relatív fizikai koordinátá-
                           nak megfelelő logikai koordináta
  CD1F  CALL CC7A        ; abszolut logikai koordináta meghatáro-
                           zása
  CD22  CALL C9B3        ; X jóságának ellenörzése
  CD25  JR   NC,07       ; még egy karakter kifér a sorba, ugrás
                           a CD2EH címre
A tollpozíció beállítása: X=0, Y(logikai)=Y(logikai)-40:
  CD27  LD   BC,CD31     ; BC-be a "soremelés" rutin címe
  CD2A  PUSH BC
A "kocsi vissza" rutin:
  CD2B  LD   BC,0000     ; X=0
  CD2E  JP   CA69        ; az új tollpozíció elmentése
A "soremelés" rutin:
  CD31  LD   DE,FFB4     ; DE-be Y relatív logikai koordináta
  CD34  CALL CC7A        ; abszolut Y meghatározása
  CD37  XOR  A
  CD38  CALL C9BA        ; Y ellenörzése
  CD3B  RET  C           ; nem fér el új sor a képernyön, RET
  CD3C  LD   DE,FFD8     ; új sornak megfelelő relatív Y koord.
  CD3F  LD   BC,0000     ; X nem változik
  CD42  CALL CC7A        ; abszolut logikai koordináták
  CD45  JP   CA65        ; az új tollpozíció elmentése

A FILL rutin:
  CD48  CALL C98F        ; a video RAM belapozása a P2 lapra
  CD4B  LD   (0B88),SP
  CD4F  LD   IY,CACF     ; XOR vonalkeresztezési mód rutin címe
Az aktuális pont byte meghatározása:
  CD53  CALL CC3F
```

```
CD56  CALL CC10
CD59  LD   (0B8A),A    ; az aktuális pont byte elmentése
CD5C  LD   E,A
CD5D  CALL CC0D        ; az ink byte meghatározása
CD60  XOR  E
CD61  LD   HL,0000
CD64  RET  Z           ; az adott pont tinta színü, nincs mit
                         festeni, ekkor a FILL rutin vége
CD65  LD   (0B91),HL
CD68  PUSH HL
CD69  LD   HL,(0B7C)   ; HL-be az X logikai koordináta
CD6C  LD   B,04
CD6E  CALL C9C6        ; HL=HL/16
CD71  PUSH HL
CD72  CALL C9DF        ; az aktuális pont video RAM-beli címe
CD75  LD   A,40        ; A=64, a képsoronkénti byte szám
CD77  POP  DE          ; E-ben az adott pont sorelejéhez viszo-
                         nyított címe
CD78  SUB  E
CD79  LD   B,A         ; B-be az adott pont és a képsor vége
                         közti távolság byte-ban
CD7A  LD   (0B8D),HL
CD7D  LD   (0B8B),BC
CD81  LD   B,01
CD83  JR   1E          ; ugrás a CDA3 címre
CD85  POP  HL
CD86  LD   A,H
CD87  OR   L
CD88  RET  Z           ; RET. ha HL=0
CD89  POP  BC
CD8A  LD   (0B91),HL
CD8D  LD   (0B8B),BC
CD91  POP  HL
CD92  LD   DE,0040
CD95  ADD  HL,DE       ; HL-be az egy sorral lentebbi pont címe
CD96  LD   (0B8D),HL
CD99  LD   B,02
CD9B  LD   DE,BC00     ; DE-be a képernyö utolsó byte-jának a
                         címe
CD9E  SBC  HL,DE
CDA0  JP   NC,CE63     ; még nem érte el a képernyö jobb alsó
                         sarkát, ugrás
CDA3  EXX
CDA4  CALL CE92        ; ST_LIMIT átlépés ellenörzése
CDA7  JR   C,DC        ; még van hely a STACK-ban, ugrás a
                         CD85H címre
```

```
CDA9  LD   HL,(0B91)
CDAC  LD   A,H
CDAD  RES  7,H
CDAF  RES  6,H
CDB1  LD   (0B8F),HL
CDB4  EX   DE,HL
CDB5  BIT  6,A
CDB7  JR   Z,09          : ugrás a CDC2H címre, ha A 6. bitje 0
CDB9  RLCA
CDBA  EXX
CDBB  XOR  B
CDBC  EXX
CDBD  AND  01
CDBF  JP   NZ,CE62
CDC2  LD   HL,(0B8D)     : HL-be a pont címe
CDC5  LD   BC,(0B8B)
CDC9  LD   A,(0B8A)      : A-ba a pont byte
CDCC  XOR  (HL)
CDCD  AND  C
CDCE  JR   NZ,5F         : ugrás a CE2FH címre
CDD0  LD   A,40
CDD2  SUB  B
CDD3  INC  A
CDD4  LD   B,A           : B-be az adott pont távolsága byte-okban a sor elejétől
CDD5  LD   DE,0000
CDD8  JR   0B            : ugrás a CDE9H címre
Festés a sor elejétől egy sorban:
CDDA  LD   A,(0B8A)
CDDD  XOR  (HL)
CDDE  AND  C
CDDF  JR   NZ,0B         : nem a kiválasztott pont színével egyező színű ponthoz ért az adott sorban, ugrás a CDECH címre
CDE1  CALL CAC0          : egy pont kirajzolása
CDE4  INC  DE
CDE5  RLC  C
CDE7  JR   NC,F1         : következő byte-hoz tartozó pontok kirajzolása, ugrás a CDDAH címre
CDE9  DEC  L
CDEA  DJNZ EE            : még nem ért a pixel sor adott pontjáig, ugrás a CDDAH címre
CDEC  LD   A,40
CDEE  SUB  B
CDEF  INC  A
```

```
CDF0  LD   B,A          ; B-be egy adott pont távolsága byte-ban
                          a pixelsor végétől
CDF1  RRC  C
CDF3  JR   NC,02        ; az adott pont nem az adott byte-hoz
                          tartozó utolsó pont, ugrás a CDF7H
                          címre
CDF5  INC  L
CDF6  DEC  B            ; HL és B a következő byte-ra mutat
CDF7  PUSH HL
CDF8  PUSH BC
CDF9  PUSH DE
CDFA  LD   HL,(0B8D)
CDFD  LD   BC,(0B8B)
CE01  CALL CE7A         ; a következő festendő pontok vizsgálata
CE04  EX   SP,HL
CE05  LD   A,H
CE06  OR   L
CE07  JR   Z,02         ; ha HL=0. ugrás a CE0BH címre
CE09  SET  7,H
CE0B  ADD  HL,DE
CE0C  EX   SP,HL
CE0D  PUSH HL
CE0E  LD   HL,(0B8F)
CE11  OR   A            ; CARRY törlése
CE12  SBC  HL,DE
CE14  POP  DE
CE15  EX   SP,HL
CE16  PUSH AF
CE17  LD   A,80
CE19  BIT  7,H
CE1B  JR   NZ,09        ; ha HL 7. bitje 0, ugrás a CE26H címre
CE1D  JR   C,09         ; ugrás a CE28H címre, ha CARRY=1
CE1F  EXX
CE20  RRC  B
CE22  RLC  B
CE24  EXX
CE25  RRA
CE26  XOR  H
CE27  LD   H,A
CE28  POP  AF
CE29  EX   SP,HL
CE2A  JR   C,36         ; ugrás a CE62H címre
CE2C  JR   Z,34         ; ugrás a CE62H címre
CE2E  EX   DE,HL
CE2F  CALL CE92         ; ST_LIMIT átlépés
```

```
CE32  JP   C,CD85      ; még van hely a STACK-ban. ugrás
CE35  LD   A,E
CE36  OR   A
CE37  JR   NZ,01
CE39  DEC  D
CE3A  LD   A,(0B8A)    ; az aktuális pont byte az A-ba
CE3D  XOR  (HL)
CE3E  AND  C
CE3F  JR   Z,0B        ; ugrás a CE4CH címre
CE41  DEC  E
CE42  JR   Z,1A        ; ugrás a CE5EH címre
CE44  RRC  C
CE46  JR   NC,F2       ; még az adott byte-hoz tartozó pixele-
                         ket vizsgálja, ugrás a CE3AH címre
CE48  INC  L
CE49  DEC  B
CE4A  JR   EE          ; ugrás a CE3AH címre
CE4C  LD   A,E
CE4D  OR   A
CE4E  JR   NZ,01
CE50  INC  D
CE51  LD   (0B8F),DE
CE55  PUSH HL
CE56  EX   SP,HL
CE57  PUSH BC
CE58  CALL CE7A        ; adott pont byte vizsgálat és festés a
                         sor vége felé
CE5B  PUSH DE
CE5C  JR   AF          ; ugrás a CE0DH címre
CE5E  DEC  D
CE5F  JP   P,CE44
CE62  EXX
CE63  DEC  B
CE64  JP   Z,CD85
CE67  EXX
CE68  LD   HL,(0B8D)
CE6B  LD   DE,FF80
CE6E  ADD  HL,DE       ; HL=HL-80H
CE6F  LD   (0B8D),HL
CE72  BIT  7,H
CE74  JP   Z,CD85
CE77  JP   CDA4
Festés az adott sorban egy pont pozícióból a sor vége felé:
CE7A  LD   DE,0000
CE7D  LD   A,(0B8A)    ; A-ba az aktuális pont byte
```

```
CE80  XOR  (HL)
CE81  AND  C
CE82  RET  NZ          : más színü pont, az adott sorban vége
                         a festésnek
CE83  INC  DE
CE84  LD   A,(0B94)    : A-ba az ink byte
CE87  AND  C
CE88  XOR  (HL)
CE89  LD   (HL),A      : az adott pont INK színüre festése
CE8A  RRC  C
CE8C  JR   NC,EF       : még az adott memória byte-hoz tartozó
                         pontok nem fogytak el, ugrás a CE7DH
                         címre
CE8E  INC  L
CE8F  DJNZ EC          : még nem érte el a sor elejét, a követ-
                         kezö címhez tartozó pontok festhetök,
                         ugrás a CE7DH címre
CE91  RET              : sorfestés vége
Az ST_LIMIT átlépés ellenörzése:
CE92  PUSH HL
CE93  PUSH DE
CE94  LD   HL,FFEA
CE97  ADD  HL,SP       ; HL=SP-16H
CE98  LD   DE,(0B17)   ; DE-be ST_LIMIT
CE9C  SBC  HL,DE
CE9E  JP   NC,CEC9     : nincs ST_LIMIT átlépés, ugrás
CEA1  PUSH BC
CEA2  EXX
CEA3  PUSH BC
CEA4  LD   HL,(0B88)   : HL-be az elmentett FILL rutin eleji SP
CEA7  DEC  HL
CEA8  DEC  HL
CEA9  DEC  HL          : HL=HL-3
CEAA  LD   D,H
CEAB  LD   E,L         : DE-be HL
CEAC  LD   C,L
CEAD  LD   B,H         ; BC-be HL
CEAE  LD   HL,0009
CEB1  ADD  HL,SP       ; HL=SP+9
CEB2  OR   A
CEB3  SBC  HL,BC       ; HL=SP+9-(FILL kezdeti SP)-3
CEB5  LD   H,B
CEB6  LD   L,C         ; HL-be BC
CEB7  JR   NZ,13       ; ugrás a CECCH címre
CEB9  LD   BC,000A
```

```
CEBC  LDDR                ; a STACK alsó elemeinek áthelyezése az
                            STACK új aljára
CEBE  EX   DE,HL
CEBF  LD   SP,HL          ; a STACK új alja-1-nek a betöltése
CEC0  INC  SP             ; SP új értéke
CEC1  OR   A
CEC2  EX   DE,HL
CEC3  SBC  HL,DE
CEC5  CCF
CEC6  POP  BC
CEC7  EXX
CEC8  POP  BC
CEC9  POP  DE
CECA  POP  HL
CECB  RET                 ; az ST_LIMIT rutin vége
CECC  CALL CEFE           ; regiszterek töltése a STACK-ból
CECF  ADD  HL,BC
CED0  LD   BC,BC00        ; BC-be a képernyö végének a címe+1
CED3  SBC  HL,BC          ; HL a képernyöre mutat?
CED5  JR   NC,06          ; nem, ugrás a CEDDH címre
CED7  ADD  HL,BC          ; HL visszaállítása
CED8  CALL CF14           ; képpont byte vizsgálat
CEDB  JR   Z,0F           ; ugrás a CEECH címre
CEDD  CALL CEFD           ; regiszterek töltése a STACK-ból
CEE0  OR   A
CEE1  SBC  HL,BC
CEE3  LD   BC,8000        ; BC-be a képmemória kezdete
CEE6  SBC  HL,BC
CEE8  ADD  HL,BC
CEE9  CALL NC,CF14        ; ugrás, ha HL a képhatár fölé mutat
CEEC  EXX
CEED  LD   BC,0006
CEF0  JR   NZ,05
CEF2  LDDR                ; a STACK aljának átmozgatása
CEF4  JP   CEAC
CEF7  OR   A
CEF8  SBC  HL,BC
CEFA  JP   CEAC
CEFD  EXX
CEFE  PUSH HL
A regiszterek töltése a STACK-ból (HL a STACK területre mutat):
CEFF  EXX
CF00  POP  HL
CF01  LD   D,(HL)
CF02  DEC  HL
```

```
  CF03  LD    E,(HL)
  CF04  DEC   HL
  CF05  DEC   HL
  CF06  LD    B,(HL)
  CF07  DEC   HL
  CF08  LD    A,(HL)
  CF09  AND   3F
  CF0B  DEC   HL
  CF0C  LD    L,(HL)
  CF0D  LD    H,A
  CF0E  LD    A,B
  CF0F  LD    BC,0040
  CF12  EX    DE,HL
  CF13  RET
  CF14  LD    C,A
  CF15  LD    A,(0B8A)      ; A-ba a pont byte
  CF18  LD    B,A
  CF19  LD    A,B
  CF1A  XOR   (HL)
  CF1B  AND   C
  CF1C  RET   Z
  CF1D  DEC   E
  CF1E  JR    Z,07          ; ha E=0. DEC D
  CF20  RRC   C
  CF22  JR    NC,F5         ; ugrás a CF19H címre
  CF24  INC   HL
  CF25  JR    F2            ; ugrás a CF19H címre
  CF27  DEC   D
  CF28  JP    P,CF20
  CF2B  RET

A DEFC rutin:
  CF2C  LD    A,C           ; A-ba a definiálandó karakter kódja
  CF2D  CP    E0
  CF2F  JR    NC,01         ; a kód > 223. nem definiálható karak-
                              ter, ugrás a CF32H címre
  CF31  OR    A
  CF32  LD    A,F8          ; A-ba a hibakód
  CF34  RET   P             ; a DEFC rutin vége, ha nem definiálha-
                              ható karakter kódot adtunk meg
  CF35  RES   7,C           ; C=C-128
  CF37  LD    B,00
  CF39  LD    L,C
  CF3A  LD    H,B           ; HL-be BC
  CF3B  ADD   HL,HL
```

```
   CF3C  ADD  HL,HL
   CF3D  ADD  HL,BC
   CF3E  ADD  HL,HL        ; HL=10*BC
   CF3F  LD   BC,0740      ; BC-be a definiálható karakterek mát-
                             rixát tartalmazó táblázat címe
   CF42  ADD  HL,BC        ; HL-be az adott kódú karakter mátrixá-
                             nak címe
   CF43  EX   DE,HL        ; DE-ben a karakter új mátrixa volt
   CF44  LD   BC,000A      ; BC-be 10, mert 10 byte ír le egy ka-
                             raktert
   CF47  LDIR              ; a karaktermátrix átírása
   CF49  XOR  A            ; A=0, hibátlanul futott a rutin
   CF4A  RET               ; A DEFC rutin vége

A BTEXT rutin:
   CF4B  PUSH BC
   CF4C  LD   HL,0000
   CF4F  DEC  B
   CF50  JP   M,CF54       ; a megadott oszlop pozíció 0 volt
   CF53  LD   L,B
   CF54  LD   A,(0B73)     ; A-ba GR_MODE
   CF57  ADD  A,04
   CF59  LD   B,A
   CF5A  ADD  HL,HL        ; HL-be (GR_MODE+4)^(oszlop pozíció-1)
   CF5B  DJNZ FD           ; ugrás a CF5AH címre
   CF5D  PUSH HL
   CF5E  LD   DE,FFD8      ; DE-be a karakter sorok távolsága lo-
                             gikai koordinátában
   CF61  LD   HL,03BF      ; HL-be az Y logikai koordináta maximá-
                             lis értéke
   CF64  LD   B,C          ; B-be a karaktersor pozíció
   CF65  DEC  B
   CF66  JP   M,CF6E       ; a megadott sorpozíció 0 volt
   CF69  JR   Z,03         ; a megadott sorpozíció 1 volt
   CF6B  ADD  HL,DE        ; HL-be a sorpozíciónak megfelelő Y lo-
                             gikai koordináta kerül
   CF6C  DJNZ FD           ; ugrás a CF6BH címre
   CF6E  EX   DE,HL
   CF6F  POP  BC
   CF70  EXX
   CF71  LD   BC,(0B7C)    ; BC'-be a volt X logikai koordináta
   CF75  LD   DE,(0B7E)    ; DE'-be a volt Y logikai koordináta
   CF79  EXX
   CF7A  POP  HL
   CF7B  LD   A,L          ; A-ba a sor pozíció
```

```
CF7C  OR   A
CF7D  JR   NZ,04        ; változik a sor pozíció, ugrás a
                          CF83H címre
CF7F  EXX
CF80  PUSH DE
CF81  EXX
CF82  POP  DE           ; a régi Y logikai koordináta DE-be
CF83  LD   A,H          ; A-ba az oszlop pozíció
CF84  OR   A
CF85  JR   NZ,04        ; változik az oszlop pozíció, ugrás a
                          CF8BH címre
CF87  EXX
CF88  PUSH BC
CF89  EXX
CF8A  POP  BC           ; a BC-be a régi X logikai koordináta
CF8B  LD   A,F9         ; A-ba a hibakód
CF8D  CALL C9B3         ; X koordináta ellenörzése
CF90  RET  C            ; a BTEXT rutin vége, ha az oszlop po-
                          zíció nem megfelelő
CF91  CALL C9BA         ; Y koordináta ellenörzése
CF94  RET  C            ; a BTEXT rutin vége, ha a sor pozíció
                          nem megengedett
CF95  JP   CA65         ; az új toll pozíció elmentése, a BTEXT
                          rutin vége
```

Az editor rutinok ugrótáblája:

```
CF98  DB   05     ; a rutinok száma
CF99  DW   CFA3   ; az ED_INT rutin címe
CF9B  DW   D052   ; az ED_CHIN/OUT rutin címe
CF9D  DW   D041   ; az ED_BKIN/OUT rutin címe
CF9F  DW   D01D   ; a CPOS rutin címe
CFA1  DW   D013   ; a CFIX rutin címe
```

Az ED_INT rutin:

```
CFA3  LD   A,50
CFA5  OUT  02,A         ; memórialapozás: U U V S lapkiosztás
CFA7  LD   HL,0E48      ; HL-be az "IT számláló a kurzor villo-
                          gás időzítéséhez" címe
CFAA  LD   BC,(0E49)
CFAE  INC  (HL)         ; kurzor IT számláló növelése
CFAF  LD   A,94
CFB1  SUB  (HL)         ; kurzor IT számláló vizsgálata
CFB2  JR   Z,1C         ; ha Z, ugrás a CFD0H címre
```

```
CFB4  CP   80
CFB6  RET  NZ          ; az ED_INT rutin vége, ha a kurzor po-
                         zícióban semminek nem kell változni
CFB7  LD   (HL),A      ; a kurzor IT számláló nullázása
CFB8  LD   A,(0B66)    ; A-ba LOCK_KEY
CFBB  LD   D,7F        ; D-be az alap kurzor kódja
CFBD  OR   A
CFBE  JR   Z,0C        ; ugrás a CFCCH címre
CFC0  LD   D,9E        ; D-be a CAPS kurzor kódja
CFC2  RRCA
CFC3  JR   C,07        ; ugrás a CFCCH címre
CFC5  LD   D,9F        ; D-de a SHIFT kurzor kódja
CFC7  RRCA
CFC8  JR   C,02        ; ugrás a CFCCH címre
CFCA  LD   D,8F        ; D-be az ALT kurzor kódja
CFCC  LD   A,D         ; A-ba az aktuális kurzor kódja
CFCD  JP   D420        ; a kurzor kirajzolása
CFD0  LD   (HL),A      ; a kurzor IT számláló törlése
CFD1  JP   D491        ; a kép visszaállítása a kurzor pozíci-
                         óban
```

Az EDITOR munkaterület alaphelyzetbe állítása:
A 0E48H-0E68H terület törlése:

```
CFD4  LD   HL,0E48
CFD7  LD   B,20
CFD9  XOR  A
CFDA  LD   (HL),A
CFDB  INC  HL
CFDC  DJNZ FC
```

Az ASCII képernyö törlése (feltöltése szóköz karakterrel):

```
CFDE  LD   HL,0101
CFE1  LD   (0E49),HL
CFE4  LD   HL,0100     ; HL-be az ASCII képernyö kezdö címe
CFE7  LD   DE,0101
CFEA  LD   BC,05FF     ; BC-be az ASCII képernyö mérete
CFED  LD   (HL),20     ; a szóköz karakter betöltése minden
                         címre
CFEF  LDIR
```

A grafikus módtól függö ugrási utasítás elhelyezése az EDITOR részére:

```
CFF1  LD   A,(0B73)    ; A-ba a GR_MODE
CFF4  ADD  A,A
CFF5  ADD  A,A
```

```
CFF6  LD   HL,D007     ; HL-be az ugrási cím táblázat kezdete
CFF9  LD   C,A
CFFA  ADD  HL,BC       ; HL-be a grafikus felbontástól függö
                         elem címe
CFFB  LD   DE,0E68     ; az elhelyezendö ugró utasítás címe a
                         RAM-ban
CFFE  LD   A,C3        ; A-ba a JP nnnn utasítás elsö kódja
D000  LD   (DE),A
D001  INC  DE
D002  LD   C,04
D004  LDIR             ; a táblázat 4 byte-jának elhelyezése a
                         RAM-ban
D006  RET              ; az editor munkaterületet alaphelyzetbe
                         állító rutin vége


D007  DB   AD D3 40 01
D00B  DB   DC D3 20 02
D00F  DB   D9 D3 10 04

A CFIX rutin:
D013  LD   HL,(0E49)   ; az aktuális kurzor pozíció a HL-be
D016  LD   (0E4E),HL   ; az aktuális kurzor pozíció elmentése
D019  LD   A,80        ; A legfelsö bitje 1 (van megjegyzett
                         kurzor pozíció)
D01B  JR   1C          ; ugrás a D039H címre

A CPOS rutin:
D01D  LD   HL,(0E49)   ; HL-be az aktuális kurzor pozíció
D020  LD   A,B         ; A-ba az új oszlop pozíció
D021  OR   A
D022  JR   Z,07        ; az oszlop pozíció változatlan marad,
                         ugrás a D02BH címre
D024  LD   A,(0E6B)    ; A-ba a maximális karakter pozíció egy
                         képernyö sorban
D027  CP   B
D028  JR   C,14        ; nem megengedett oszlop pozíció, ugrás
                         a D03EH címre
D02A  LD   H,B
D02B  LD   A,C         ; A-ba az új sor pozíció
D02C  OR   A
D02D  JR   Z,06        ; a sor pozíció változatlan marad, ugrás
                         a D035H címre
D02F  LD   A,18        ; a képernyö sorok maximális száma A-ba
D031  CP   C
```

```
  D032  JR   C,0A        ; nem megengedett sor pozíció, ugrás a
                           D03EH címre
  D034  LD   L,C
  D035  LD   (0E49),HL   ; az új kurzor pozíció elmentése
  D038  XOR  A           ; A=0, nincs megjegyzett kurzor pozíció
  D039  LD   (0E4D),A
  D03C  XOR  A           ; a rutin hibátlanul futott
  D03D  RET              ; az ED_CHOUT, CFIX és a CPOS rutinok vége
  D03E  LD   A,F6        ; A-ba a hibakód
  D040  RET              ; a CPOS rutin vége hiba esetén

Az ED_BKIN és ED_BKOUT rutinok:
  D041  EXX
  D042  CALL D449        ; a paper és az ink byte elökészítése
  D045  EXX
  D046  LD   HL,D0B7     ; az ED_CHOUT rutin címe HL-be
  D049  JP   P,C56D      ; a megadott paraméterek szerint az
                           ED_CHOUT rutin ismételt meghívása,
                           közben HIMEM átlépés ellenörzés
  D04C  LD   HL,D058     ; az ED_CHIN rutin címe HL-be
  D04F  JP   C58F        ; a megadott paraméterek szerint az
                           ED_CHIN rutin ismételt meghívása,
                           közben HIMEM átlépés ellenörzés

Az ED_CHIN és az ED_CHOUT rutinok:
  D052  CALL D449        ; a paper és az ink byte elökészítése
  D055  JP   P,D0B7      ; ugrás az ED_CHOUT rutinra
Az ED_CHIN rutin
  D058  LD   A,50
  D05A  LD   (0003),A
  D05D  OUT  02,A        ; memórialapozás: U U V S lapkiosztás
  D05F  LD   A,(0E4D)    ; A-ba a CFIX_FLAG
  D062  RRCA
  D063  JP   C,D0FA      ; a bekezdés következö karakterének át-
                           adása
  D066  CALL D39C        ; olvasás a LINE_TAB-ból
  D069  LD   BC,(0E49)   ; BC-be a kurzor pozíció
  D06D  PUSH BC
  D06E  LD   DE,(0E4E)   ; DE-be a megjegyzett kurzor pozíció
  D072  LD   A,C
  D073  CP   E
  D074  JR   C,06        ; az aktuális sor pozíció kisebb, mint
                           a megjegyzett
  D076  JR   NZ,09       ; az aktuális és a megjegyzett sor po-
                           zíció egyenlö, ugrás a D081H címre
```

```
D078  LD    A,B
D079  CP    D
D07A  JR    NC,0F       ; az aktuális oszlop pozíció nagyobb,
                          mint a megjegyzett
D07C  CALL  D038        ; CFIX_FLAG törlése
D07F  JR    0A          ; ugrás a D08BH címre
D081  DEC   C
D082  JR    Z,F8        ; az elsö sorban van a kurzor, ugrás a
                          D07CH címre
D084  DEC   HL
D085  BIT   7,(HL)      ; a LINE_TAB megfelelö elemének vízsgá-
                          lata
D087  LD    D,01
D089  JR    NZ,E7       ; nem ebben a sorban fejezödik be az a-
                          dott bekezdés (logikai sor), ugrás a
                          D072H címre
D08B  POP   BC          ; a kurzor pozíció BC-be
D08C  CALL  D477        ; a kurzor "alatti" képernyö rész elmen-
                          tése
D08F  LD    A,13
D091  LD    (0E48),A    ; a kurzor IT számláló töltése
D094  LD    HL,0B10     ; HL-be INT_DES címe
D097  RES   2,(HL)      ; a kurzor IT az editor-hoz rendelve
D099  RST   30H
D09A  DB    91          ; a KB_CHIN rutin meghívása
D09B  SET   2,(HL)      ; a kurzor IT nincs az editor-hoz ren-
                          delve
D09D  PUSH  AF
D09E  PUSH  BC          ; C-ben a beolvasott karakter
D09F  LD    BC,(0E49)   ; BC-be a kurzor pozíció
D0A3  CALL  D491        ; a kurzor "alatti" kép visszaállítása
D0A6  POP   BC
D0A7  POP   AF
D0A8  RET   NZ          ; a billentyüzeten CTRL-ESC-t nyomtak,
                          kilépés az editor rutinból
D0A9  LD    A,C         ; A-ba a megnyomott billentyü kódja
D0AA  CP    1B
D0AC  JR    Z,8A        ; ESC-t nyomtak, ugrás a D022H címre
D0AE  CP    0D
D0B0  JR    Z,39        ; RETURN-t nyomtak, ugrás a D0EBH címre
D0B2  CALL  D124        ; a bevitt karakter kiíratása, végrehaj-
                          tása
D0B5  JR    AF          ; ugrás a D066H címre, a program a beol-
                          vasó-szerkesztö hurokban marad
```

Az ED_CHOUT rutin:

```
  D0B7  LD   A,50
  D0B9  LD   (0003),A
  D0BC  OUT  02,A          ; memórialapozás: U U V S lapkiosztás
  D0BE  LD   HL,D038
  D0C1  PUSH HL            ; a CFIX_FLAG törlö rutin címe a STACK-
                             be
  D0C2  LD   A,C           ; A-ba a kiírandó karakter kódja
  D0C3  CP   0D
  D0C5  JR   NZ,06         ; nem RETURN karakter, ugrás a D0CDH
                             címre
  D0C7  LD   A,01
  D0C9  LD   (0E4A),A      ; az oszlop pozíció felülírása
  D0CC  RET                ; ugrás a D038H címre
  D0CD  CP   0A
  D0CF  JR   NZ,15         ; nem "új sor" karakter, ugrás a D0E6H
                             címre
  D0D1  LD   BC,(0E49)     ; BC-be a kurzor pozíció
  D0D5  LD   A,C
  D0D6  CALL D39F          ; olvasás a LINE_TAB-ból
  D0D9  INC  C
  D0DA  JR   C,F9          ; ugrás a D0D5H címre
  D0DC  DEC  C
  D0DD  PUSH BC
  D0DE  CALL D363          ; kurzor a következö sor elejére
  D0E1  POP  AF
  D0E2  LD   (0E4A),A      ; az oszlop pozíció visszaállítása
  D0E5  RET                ; ugrás a D038H címre
  D0E6  CALL D124          ; a kiíró szerkesztö rutin hívása
  D0E9  XOR  A
  D0EA  RET                ; ugrás a D038H címre

  D0EB  LD   A,(0E4D)      ; A-ba a CFIX_FLAG
  D0EE  RLCA
  D0EF  LD   BC,(0E4E)     ; BC-be a megjegyzett kurzor pozíció
  D0F3  CALL NC,D370       ; adott bekezdés elejének meghatározása
  D0F6  LD   (0E49),BC     ; a bekezdés elejének eltárolása
  D0FA  LD   BC,(0E49)     ; a kurzor pozíció BC-be
  D0FE  LD   A,C
  D0FF  CALL D39F          ; olvasás a LINE_TAB-ból
  D102  CP   B
  D103  JR   C,17          ; ha a bekezdés vége után áll a kurzor,
                             ugrás a D11CH címre
  D105  CALL D384          ; a kurzor pozícióban lévö karakter cí-
                             me a memóriában
```

```
D108  INC  B
D109  LD   A,(0E6B)    ; A-ba egy képernyö sor hossza karakter-
                         ben
D10C  CP   B
D10D  JR   NC,03       ; még nem ért a fizikai sor végére, ug-
                         rás a D112H címre
D10F  LD   B,01        ; különben új sor következik
D111  INC  C
D112  LD   (0E49),BC   ; a kurzor pozíció elmentése
D116  LD   C,(HL)      ; C-be az adott kurzor pozícióban talál-
                         ható karakter kódja
D117  LD   A,01        ; "még van át nem adott karakter" CFIX_
                         FLAG kód A-ba
D119  JP   D039        ; CFIX_FLAG eltárolása, az ED_CHOUT ru-
                         tin vége

D11C  CALL D363        ; kurzor a következö sor elejére
D11F  LD   C,0D        ; C-be RETURN kódja
D121  JP   D038        ; CFIX_FLAG törlése, az ED_CHOUT rutin
                         vége
A kiíró-szerkesztö rutin:
D124  CP   20
D126  JR   C,2F        ; kontrol karakter, ugrás a D157H címre
D128  CP   E0
D12A  JR   NC,2B       ; nem létezö karakter, ugrás a D157H
                         címre
D12C  EX   AF,AF'
D12D  CALL D384        ; a karakter címe az ASCII képernyön
D130  EX   AF,AF'
D131  LD   (HL),A      ; a karakter eltárolása az ASCII képer-
                         nyön
D132  LD   BC,(0E49)
D136  PUSH BC
D137  CALL D420        ; a megfelelö kurzor kirajzolása a kur-
                         zor pozícióba
D13A  POP  BC
D13B  LD   A,C
D13C  CALL D39F        ; olvasás a LINE_TAB-ból
D13F  CP   B
D140  JR   NC,01       ; még nem a bekezdés vége, ugrás a D143H
                         címre
D142  LD   (HL),B      ; sor hosszának tárolása a LINE_TAB-ban
D143  INC  B
D144  LD   A,(0E6B)    ; A-ba a sor maximális hossza
D147  CP   B
```

```
  D148  JR    NC,08        ; még nem a sor vége, ugrás a B152H-ra
  D14A  LD    B,01
  D14C  INC   C            ; kurzor a következő sor elején
  D14D  BIT   7,(HL)
  D14F  CALL  Z,D2CB       ; új sor füzése a bekezdéshez
  D152  LD    (0E49),BC    ; a karakter pozíció eltárolása
  D156  RET                ; a kiíró rutin vége
A szerkesztő rutin:
  D157  LD    HL,D170      ; a szövegszerkesztő funkciók ugrótáb-
                             lájának címe HL-be
A szerkesztő funkció címének meghatározása:
  D15A  LD    B,0B
  D15C  CP    (HL)
  D15D  INC   HL
  D15E  LD    E,(HL)
  D15F  INC   HL
  D160  LD    D,(HL)
  D161  INC   HL
  D162  JR    Z,03         ; létező szerkesztő funkció, ugrás a
                             D167H címre
  D164  DJNZ  F6
  D166  RET                ; nem létező fukció, a szerkesztő rutin
                             vége
  D167  LD    BC,(0E49)    ; BC-be a kurzor pozíció
  D16B  LD    A,(0E6B)     ; A-ba a sorok maximális hossza
  D16E  EX    DE,HL
  D16F  JP    (HL)         ; ugrás a konkrét szerkesztő rutinba


A Basic beépített szövegszerkesztöje funkcióinak ugrótáblája:

  D170  DB    13           ; RJL (kurzor balra) kódja
  D171  DW    D191         ; RJL rutin címe
  D173  DB    04           ; RJR (kurzor jobbra) kódja
  D174  DW    D198         ; RJR rutin címe
  D176  DB    05           ; RJU (kurzor fel) kódja
  D177  DW    D195         ; RJU rutin címe
  D179  DB    18           ; RJD (kurzor le) kódja
  D17A  DW    D19E         ; RJD rutin címe
  D17C  DB    16           ; INS kódja
  D17D  DW    D205         ; INS rutin címe
  D17F  DB    07           ; DC (shift+del) kódja
  D180  DW    D287         ; DC rutin címe
```

```
D182  DB   08          ; DEL kódja
D183  DW   D278        ; DEL rutin címe
D185  DB   19          ; DL (ctrl+Y) kódja
D186  DW   D1FD        ; DL rutin címe
D188  DB   0E          ; IL (ctrl+N) kódja
D189  DW   D1F7        ; IL rutin címe
D18B  DB   0B          ; CEL (ctrl+K) kódja
D18C  DW   D1A8        ; CEL rutin címe
D18E  DB   09          ; TAB (ctrl+I) kódja
D18F  DW   D1CD        ; TAB rutin címe
```

Az RJL (kurzor balra) rutin:

```
D191  DEC  B
D192  JR   NZ,0F       ; még nem a sor eleje, ugrás a D1A3H-ra
D194  LD   B,A
```

Az RJU (kurzor fel) rutin:

```
D195  DEC  C           ; kurzor az elözö sor végére
D196  JR   0A          ; ugrás a D1A2H címre
```

A RJR (kurzor jobbra) rutin:

```
D198  INC  B
D199  CP   B
D19A  JR   NC,07       ; még nem a sor vége, ugrás a D1A3H-ra
D19C  LD   B,01
```

Az RJD (kurzor le) rutin:

```
D19E  INC  C           ; kurzor pozíció a következö sor elejére
D19F  LD   A,C
D1A0  CP   19
D1A2  RET  Z           ; a kurzor a képernyö elsö vagy utolsó
                         pozíciójában volt
D1A3  LD   (0E49),BC   ; az új kurzor pozíció elmentése
D1A7  RET
```

A CEL (törlés a logikai sor végéig) rutin:

```
D1A8  PUSH BC
D1A9  CALL D4AD        ; a sor törlése a video RAM-ból
D1AC  POP  BC
D1AD  CALL D384        ; kurzor pozíció címe az ASCII képer-
                         nyön
```

A sor törlése az ASCII képernyön:

```
D1B0  DEC  B
D1B1  LD   A,(0E6B)
D1B4  SUB  B
D1B5  LD   (HL),20     ; a SPACE karakter töltése az ASCII kép-
                         ernyö megfelelö helyére
```

```
D1B7  INC  HL
D1B8  DEC  A
D1B9  JR   NZ,FA
D1BB  CALL D39C        ; olvasás a LINE_TAB-ból
D1BE  LD   (HL),B      ; a LINE_TAB megfelelö elemének módosí-
                         tása
D1BF  INC  C
D1C0  INC  HL
D1C1  RET  NC          ; a logikai sor nem folytadódik a követ-
                         kezö sorban, RET
D1C2  LD   B,(HL)
D1C3  LD   A,C
D1C4  EXX
D1C5  CALL D31E

D1C8  EXX
D1C9  RLC  B
D1CB  JR   F4          ; ugrás a D1C1H címre

A TAB rutin:
D1CD  CALL D39C        ; olvasás a LINE_TAB-ból
D1D0  EX   AF,AF'
D1D1  LD   A,B
D1D2  DEC  A
D1D3  AND  F8
D1D5  ADD  A,09        ; A-be a következö TAB pozíciónak meg-
                         felelö oszlop pozíció
D1D7  LD   B,A
D1D8  LD   A,(0E6B)
D1DB  CP   B
D1DC  JR   NC,0A       ; volt még TAB-nyi hely a sorban, ugrás
                         a D1E8H címre
D1DE  OR   80
D1E0  LD   (HL),A      ; a bekezdés a következö sorban folyta-
                         tódik
D1E1  LD   B,01
D1E3  INC  C           ; a kurzor a következö sor elején
D1E4  EX   AF,AF'
D1E5  CALL NC,D2CB     ; új sor füzése a bekezdéshez
D1E8  LD   A,C
D1E9  CALL D39F        ; olvasás a LINE_TAB-ból
D1EC  DEC  B
D1ED  CP   B
D1EE  JR   NC,01       ; túl vagyunk a bekezdés végén, ugrás
D1F0  LD   (HL),B      ; a LIN_TAB megfelelö elemének átírása
```

```
  D1F1  INC  B
  D1F2  LD   (0E49),BC  ; új kurzor pozíció elmentése
  D1F6  RET             ; a TAB rutin vége

Az IL (sor beszúrás) rutin:
  D1F7  CALL D370       ; a kurzor a bekezdés elejére
  D1FA  JP   D2D6       ; sor beszúrás

A DL (logikai sor törlés) rutin:
  D1FD  CALL D370       ; a kurzor a bekezdés elejére
  D200  CALL D39C       ; olvasásá a LINE_TAB-ból
  D203  JR   BD         ; ugrás a D1C2H címre

Az INS (SPACE beszúrás) rutin:
  D205  CALL D39C       ; olvasás a LINE_TAB-ból
  D208  CP   B
  D209  RET  C          ; túl vagyunk a bekezdés végén
  D20A  LD   E,01
  D20C  JR   02         ; ugrás a D210H címre
  D20E  INC  HL
  D20F  INC  E
  D210  LD   A,(HL)     ; a LINE_TAB megfelelö eleme A-ba
  D211  RLCA
  D212  INC  C
  D213  JR   C,F9       ; a bekezdés a következö sorban folyta-
                          tódik, ugrás a D20EH cimre
  D215  INC  (HL)       ; a sor hossz byte növelése
  D216  LD   A,(0E6B)   ;
  D219  CP   (HL)
  D21A  JR   NZ,17      ; a bekezdés vége nem a sor vége, ugrás
                          a D233H címre
  D21C  LD   A,E
  D21D  ADD  A,C
  D21E  CP   31
  D220  JR   NZ,04      ; ugrás a D226H címre
  D222  DEC  (HL)       ; a sor hossz byte csökkentése
  D223  SCF
  D224  JR   0E         ; ugrás a D234H címre
  D226  PUSH DE
  D227  PUSH BC
  D228  CALL D2CB       ; új sor füzése a bekezdéshez
  D22B  POP  AF
  D22C  POP  DE
  D22D  LD   B,A
  D22E  LD   A,C
```

```
D22F  DEC  A
D230  CALL D39F      ; olvasás a LINE_TAB-ból
D233  OR   A
D234  EX   AF,AF'
D235  OR   A
D236  DEC  C
D237  DEC  E
D238  PUSH DE
D239  PUSH BC
D23A  PUSH HL
D23B  PUSH AF
D23C  JR   Z,02      ; elértük a beszúrás sor pozícióját, ug-
                       rás a D240H címre
D23E  LD   B,01      ; pozícionálás a sor elejére
D240  PUSH BC
D241  CALL D592      ; az adott sor módosítása a video RAM-
                       ban
D244  POP  BC
D245  LD   DE,0E6B
D248  LD   A,(DE)    ; A-ba a sorok maximális hossza
D249  LD   H,A
D24A  LD   L,C       ; HL-be az adott sor vége pozíció
D24B  CALL D387      ; a HL-ben tárolt karakter pozíció címe
                       az ASCII képernyön
D24E  PUSH HL
D24F  EX   AF,AF'
D250  JR   NC,02     ; a logikai sor nem folytatódik a követ-
                       kező sorban, ugrás a D254H címre
D252  DEC  HL
D253  INC  B
D254  LD   A,(DE)
D255  SUB  B
D256  LD   C,A
D257  LD   B,00      ; BC-be a kurzor távolsága a sor végétől
D259  LD   D,H
D25A  LD   E,L       ; DE-be HL
D25B  DEC  HL
D25C  JR   Z,02      ; a sor vége, ugrás a D260H címre
D25E  LDDR           ; a sor tartalmának eltolása jobbra egy
                       karakterrel az ASCII képernyön
D260  POP  HL
D261  LD   BC,FFC0
D264  ADD  HL,BC     ; HL-be az előző sor vége címe
D265  LD   B,(HL)    ; B-be az utolsó karakter
D266  POP  AF
```

```
  D267  JR   NZ,02
  D269  LD   B,20          ; B-be a SPACE kódja
  D26B  LD   A,B
  D26C  LD   (DE),A        ; a sor elejére az elözö sor utolsó
                             karaktere kerül
  D26D  POP  HL
  D26E  POP  BC
  D26F  POP  DE
  D270  DEC  HL
  D271  JR   NZ,C1         ; még nincs kész a beszúrás utáni szö-
                             veg eltolása, ugrás a D234H címre
  D273  LD   (0E49),BC     ; a kurzor pozíció elmentése
  D277  RET                ; az INS rutin vége

A DEL (karakter törlés balra) rutin:
  D278  DEC  B             ; az oszlop pozíció csökkentése
  D279  JR   NZ,08         ; még nem a sor elje, ugrás a D283H cím-
                             re
  D27B  DEC  C             ; az elözö sorra mutat a sor pozíció
  D27C  RET  Z             ; a képernyö elejére értünk, a DEL rutin
                             vége
  D27D  LD   A,C
  D27E  CALL D39F          ; olvasás a LINE_TAB-ból
  D281  RET  NC            ; a bekezdés elejére értünk, a DEL rutin
                             vége
  D282  LD   B,A           ; B-be a sor vége pozíció
  D283  LD   (0E49),BC     ; a kurzor pozíció elmentése
A DL ( törlés a kurzor pozícióban) rutin:
  D287  CALL D39C          ; olvasás a LINE_TAB-ból
  D28A  CP   B
  D28B  RET  C
  D28C  EX   AF,AF'
  D28D  LD   A,(HL)
  D28E  AND  7F
  D290  SUB  B
  D291  JR   C,30          ; a logikai sor vége után vagyunk, ug-
                             rás a D2C3H címre
  D293  PUSH HL
  D294  PUSH AF
  D295  PUSH DE
  D296  PUSH BC
  D297  CALL D542          ; az adott sor képe átírása a video RAM-
                             ban
  D29A  POP  BC
  D29B  POP  DE
```

```
D29C  LD   H,B
D29D  LD   L,C
D29E  CALL D387      ; az adott kurzor pozíció címe az ASCII
                       képernyön
D2A1  EX   AF,AF'
D2A2  LD   A,(HL)
D2A3  JR   NC,01     ; a logikai sor az adott sorban befeje-
                       zödik, ugrás a D2A6H címre
D2A5  LD   (DE),A
D2A6  LD   D,H
D2A7  LD   E,L       ; DE-be HL
D2A8  INC  HL
D2A9  POP  AF
D2AA  PUSH BC
D2AB  JR   Z,05      ; nem értünk a sor végére, ugrás a
                       D2B2H címre
D2AD  LD   C,A
D2AE  LD   B,00      ; BC-be a következö sor eleje
D2B0  LDIR           ; az adott sor mozgatása az ASCII kép-
                       ernyön
D2B2  LD   A,20      ; A-ba a SPACE kódja
D2B4  LD   (DE),A
D2B5  POP  BC
D2B6  INC  C
D2B7  LD   B,01      ; BC-be a következö sor eljének pozíció-
                       ja
D2B9  POP  HL
D2BA  BIT  7,(HL)
D2BC  INC  HL
D2BD  SCF
D2BE  JR   NZ,CC     ; ugrás a D28CH címre, ha nincs kész a
                       törlés utáni karakterek átmozgatása
D2C0  DEC  HL
D2C1  DEC  (HL)
D2C2  RET            ; a DEL és a DL rutin vége

D2C3  DEC  HL        ; HL-be az elözö sor címe a LINE_TAB-ban
D2C4  DEC  (HL)      ; sor hosszának csökkentése
D2C5  RES  7,(HL)    ; ez a bekezdés utolsó sora
D2C7  LD   A,C
D2C8  JP   D31E

Egy új sor füzése egy bekezdéshez:
D2CB  SET  7,(HL)    ; a LINE_TAB megfelelö elemének módosí-
                       tása: a bekezdés a következö sorban
```

```
                        folytatódik
D2CD  LD   A,C          ; A-ba a sor pozíció
D2CE  CP   19
D2D0  JR   Z,3F         ; már nem fér ki a képernyöre, scroll,
                          ugrás a D311H címre
D2D2  INC  HL
D2D3  LD   A,(HL)       ; A-ba a következö sor hossza
D2D4  OR   A
D2D5  RET  Z            ; a következö sor üres, rutin vége
D2D6  PUSH BC
D2D7  CALL D509         ; a sor beszúrása a video RAM-ban
D2DA  POP  BC
D2DB  PUSH BC
D2DC  LD   A,19
D2DE  SUB  C
D2DF  LD   HL,0E66
D2E2  LD   DE,0E67      ; DE-be a LINE_TAB vége
D2E5  PUSH DE
D2E6  LD   B,00
D2E8  LD   C,A          ; BC-be az átmozgatandó sorok száma
D2E9  LDDR              ; a LINE_TAB megfelelö elemeinek átmoz-
                          gatása
D2EB  INC  DE
D2EC  EX   ,HL
D2ED  LD   (HL),C
D2EE  POP  HL
D2EF  RES  7,(HL)
D2F1  RRA
D2F2  RR   C
D2F4  RRA
D2F5  RR   C
D2F7  LD   B,A
D2F8  LD   HL,06BF      ; HL-be az utolsó sor elsö karakterének
                          címe az ASCII képernyön
D2FB  LD   DE,06FF      ; DE-be a képernyö utolsó karakterének
                          címe az ASCII képernyön
D2FE  LDDR
D300  LD   B,40         ; egy sor hossza az ASCII képernyön
D302  LD   A,20         ; A-ba a SPACE kódja
D304  INC  DE
D305  LD   (DE),A       ; a beszurt sor karakterhelyeinek fel-
                          töltése
D306  DJNZ FC           ; ha nincs kész, ugrás a D304H címre
D308  POP  BC
D309  LD   HL,0E4E
```

```
D30C  LD   A,(HL)       ; A-ba a megjegyzett sor pozíció
D30D  SUB  C
D30E  RET  C            ; a megjegyzett pozíció után vagyunk,
                          a rutin vége
D30F  INC  (HL)         ; a sorhossz byte növelése
D310  RET               ; a beszúró rutinok vége

D311  LD   A,01
D313  CALL D31E
D316  LD   BC,0118      ; BC-be az utolsó sor elejének pozíciója
D319  LD   (0E49),BC    ; a kurzor pozíció elmentése
D31D  RET               ; a rutin vége

D31E  PUSH AF
D31F  LD   C,A          ; C-be a sor pozíció
D320  LD   HL,0E4E      ; HL-be a megjegyzett sor pozíció
D323  SUB  (HL)
D324  CALL Z,D039       ; a megjegyzett sorban vagyunk, ugrás,
                          CFIX_FLAG törlés
D327  JR   NC,01        ; a megjegyzett sor alatt vagyunk, ug-
                          rás a D32AH címre
D329  DEC  (HL)         ; a megjegyzett sor pozíció csökkentése
D32A  CALL D4E0         ; a video RAM-ban törlés
D32D  POP  BC
D32E  LD   A,B          ; A-ba a sor pozíció
D32F  CALL D39F         ; olvasás a LINE_TAB-ból
D332  PUSH HL
D333  LD   L,B
D334  LD   H,01         ; HL-be a sor kezdetének pozíciója
D336  CALL D387         ; az adott kurzor pozíció címe az ASCII
                          képernyön
D339  EX   DE,HL
D33A  LD   HL,0040
D33D  ADD  HL,DE        ; az egy sorral lentebbi pozíció címe az
                          ASCII képernyön
D33E  LD   A,18
D340  SUB  B
D341  PUSH AF
D342  JR   Z,0B         ; az utolsó sor volt, ugrás a D34FH-ra
D344  LD   C,00
D346  RRA
D347  RR   C
D349  RRA
D34A  RR   C
D34C  LD   B,A          ; az átmozgatandó byte-ok száma
```

```
D34D  LDIR              ; egy sor eltüntetése az ASCII képernyön
D34F  LD   B,40
D351  DEC  HL
D352  LD   (HL),20      ; a bekezdés vége utáni sor feltöltése
                          SPACE karakterrel az ASCII képernyön
D354  DJNZ FB           ; nincs kész, ugrás a D351H címre
D356  POP  AF
D357  LD   C,A
D358  POP  HL
D359  LD   D,H
D35A  LD   E,L          ; DE-be HL: a LINE_TAB megfelelö eleme
D35B  INC  HL
D35C  JR   Z,02         ; utolsó sor, ugrás a D360H címre
D35E  LDIR              ; a sorhossz byte-ok átmozgatása a LINE_
                          TAB táblán
D360  XOR  A
D361  LD   (DE),A       ; a bekezdés vége utáni sor hossza 0
D362  RET               ; a törlö rutinok vége

D363  LD   A,C          ; A-ba a sor pozíció
D364  CP   18
D366  JR   Z,A9         ; utolsó sor, ugrás a D311H címre
D368  INC  C
D369  LD   B,01         ; BC-be a következö sor elsö karakteré-
                          nek a pozíciója
D36B  LD   (0E49),BC    ; a kurzor pozíció elmentése
D36F  RET               ; a rutin vége

Egy bekezdés elsö sorának meghatározása:
D370  LD   BC,(0E49)    ; BC-be a kurzor pozíció
D374  OR   A
D375  DEC  C
D376  LD   A,C          ; A-ba az elözö sor pozíció
D377  CALL NZ,D39F      ; nem a képernyö teteje volt, ugrás
D37A  JR   C,F8         ; a sor nem egy bekezdés utolsó sora,
                          nézzük a megelözöt
D37C  INC  C            ; C-be a bekezdés elsö sora
D37D  LD   B,01
D37F  LD   (0E49),BC    ; a kurzor pozíció a bekezdés elejére
D383  RET               ; a rutin vége

A kurzor pozíció címe az ASCII képernyön:
D384  LD   HL,(0E49)    ; HL-be a kurzor pozíció
D387  DEC  H
D388  DEC  L
```

```
D389  LD   A,H
D38A  ADD  A,A
D38B  ADD  A,A
D38C  LD   H,L
D38D  RR   H
D38F  RRA
D390  RR   H
D392  RRA
D393  LD   L,A
D394  LD   A,H
D395  ADD  A,01
D397  LD   H,A          ; HL-be 256+64*L+H
D398  LD   (0E4B),HL    ; a keresett cím elmentése
D39B  RET               ; a rutin vége

A sorhossz byte meghatározása:
D39C  LD   A,(0E49)     ; A-ba a sor pozíció
D39F  ADD  A,4F
D3A1  LD   L,A
D3A2  LD   A,0E
D3A4  ADC  A,00
D3A6  LD   H,A          ; HL-be 0E4FH+A, a LINE_TAB megfelelö e-
                          lemének a címe
D3A7  LD   A,(HL)       ; A-ba a sorhossz byte
D3A8  RLCA              ; A 7. bitje a CARRY-be
D3A9  OR   A
D3AA  RR   A            ; A visszaállítása
D3AC  RET               ; a rutin vége

A megfelelö kurzor kirajzolása a képernyöre 2 színü módban:
D3AD  LD   DE,0040      ; DE-be a soronkénti karakter szám
D3B0  EXX
D3B1  LD   A,(HL)       ; A-ba a kurzor karaktermátrixának egy
                          sora
D3B2  EXX
D3B3  AND  C
D3B4  XOR  B            ; A módosítása a paper és az ink byte-
                          ok segítségével
D3B5  LD   (HL),A       ; a kurzor megfelelö pontjainak kirajzo-
                          lása a képernyöre
D3B6  ADD  HL,DE        ; HL-be a kurzor következö sorának
                          video RAM címe
D3B7  EXX
D3B8  INC  HL
D3B9  DJNZ F6           ; még nincs kész a kurzor, ugrás a
```

```
                                D3B1H címre
  D3BB  RET                   ; a rutin vége

A kurzor kirajzolása 4 színü módban:
  D3BC  EXX
  D3BD  LD    A,(HL)          ; A-ba a kurzor karaktermátrixának egy
                                sora
  D3BE  EXX
  D3BF  LD    D,A
  D3C0  RRCA
  D3C1  RRCA
  D3C2  RRCA
  D3C3  RRCA
  D3C4  XOR   D
  D3C5  AND   C
  D3C6  LD    E,A
  D3C7  AND   0F
  D3C9  XOR   D
  D3CA  AND   C
  D3CB  XOR   B
  D3CC  LD    (HL),A          ; a kurzor fél sorának kirajzolása
  D3CD  INC   L
  D3CE  XOR   E
  D3CF  LD    (HL),A          ; a kurzor másik fél sorának kirajzolása
  D3D0  LD    DE,003F
  D3D3  ADD   HL,DE           ; a kurzor következö sorának címe HL-be
  D3D4  EXX
  D3D5  INC   HL
  D3D6  DJNZ  E5              ; nincs kész a kurzor, ugrás a D3BDH-ra
  D3D8  RET                   ; a rutin vége

A kurzor kirajzolása 16 színü módban:
  D3D9  EXX
  D3DA  LD    A,(HL)          ; A-ba a kurzor karaktermátrixának a
                                megfelelö sora
  D3DB  EXX
  D3DC  RLA
  D3DD  LD    E,A
  D3DE  SBC   A,A
  D3DF  LD    D,A
  D3E0  RL    E
  D3E2  SBC   A,A
  D3E3  XOR   D
  D3E4  AND   55
  D3E6  XOR   D
```

```
D3E7  AND  C
D3E8  XOR  B
D3E9  LD   (HL),A      ; a kurzor megfelelö sora elsö negyedé-
                         nek a kirajzolása
D3EA  INC  L
D3EB  RL   E
D3ED  SBC  A,A
D3EE  LD   D,A
D3EF  RL   E
D3F1  SBC  A,A
D3F2  XOR  D
D3F3  AND  55
D3F5  XOR  D
D3F6  AND  C
D3F7  XOR  B
D3F8  LD   (HL),A      ; a kurzor megfelelö sora 2. negyedének
                         a kirajzolása
D3F9  INC  L
D3FA  RL   E
D3FC  SBC  A,A
D3FD  LD   D,A
D3FE  RL   E
D400  SBC  A,A
D401  XOR  D
D402  AND  55
D404  XOR  D
D405  AND  C
D406  XOR  B
D407  LD   (HL),A      ; a kurzor megfelelö sora 3. negyedének
                         a kirajzolása
D408  INC  L
D409  RL   E
D40B  SBC  A,A
D40C  LD   D,A
D40D  RL   E
D40F  SBC  A,A
D410  XOR  D
D411  AND  55
D413  XOR  D
D414  AND  C
D415  XOR  B
D416  LD   (HL),A      ; a kurzor megfelelö sora 4. negyedének
                         a kirajzolása
D417  LD   DE,003D
D41A  ADD  HL,DE       ; HL-be a kurzor következö sorának a cí-
```

```
                                me a video RAM-ban
  D41B  EXX
  D41C  INC  HL
  D41D  DJNZ BB             ; ha nincs kész a kurzor, ugrás a D3DAH
                              címre
  D41F  RET                 ; a rutin vége

A kurzor kirajzoló rutin:
  D420  EX   AF,AF'
  D421  CALL D459           ; a kurzor bal felsö sarka címének meg-
                              határozása
  D424  PUSH HL
  D425  EX   AF,AF'
  D426  LD   C,A            ; C-be a kurzor karakter kódja
  D427  LD   B,00
  D429  LD   L,C
  D42A  LD   H,B            ; HL-be BC
  D42B  ADD  HL,HL
  D42C  ADD  HL,HL
  D42D  ADD  HL,BC
  D42E  ADD  HL,HL
  D42F  BIT  7,C
  D431  LD   BC,C474
  D434  JR   Z,03           ; alap kurzor az aktuális kurzor, ugrás
                              a D439H címre
  D436  LD   BC,0240
  D439  ADD  HL,BC          ; HL-be a kurzor karaktermátrixának a
                              címe
  D43A  LD   B,0A           ; egy karakter 10 sorból áll
  D43C  EXX
  D43D  LD   A,(0E96)
  D440  LD   B,A            ; B-be a paper byte
  D441  LD   A,(0E95)
  D444  LD   C,A            ; C-be az ink byte
  D445  POP  HL
  D446  JP   0E68           ; ugrás a grafikus felbontástól függö
                              rutinra

A paper és az ink byte meghatározása:
  D449  EX   AF,AF'
  D44A  CALL CC05
  D44D  LD   (0E96),A       ; a paper byte eltárolása
  D450  CALL CC0D
  D453  XOR  B
  D454  LD   (0E95),A       ; az ink byte eltárolása
```

```
  D457  EX   AF,AF'
  D458  RET                ; a rutin vége

A kurzor bal felsö sarka címének meghatározása a video RAM-ban:
  D459  DEC  B
  D45A  LD   A,(0B73)      ; A-ba a GR_MODE
  D45D  OR   A
  D45E  JR   Z,05          ; két szinü mód, ugrás a D464H címre
  D460  SLA  B
  D462  DEC  A
  D463  JR   NZ,FB         ; ugrás a D460H címre
  D465  DEC  C
  D466  LD   A,C
  D467  ADD  A,A
  D468  ADD  A,A
  D469  ADD  A,C
  D46A  ADD  A,A
  D46B  LD   H,A
  D46C  XOR  A
  D46D  SRL  H
  D46F  RRA
  D470  SCF
  D471  RR   H
  D473  RRA
  D474  OR   B
  D475  LD   L,A           ; HL-be a kurzor bal felsö sarkának a
                             video RAM címe
  D476  RET                ; a rutin vége

A kurzor alatti kép elmentése:
  D477  CALL D459          ; A kurzor video RAM címének meghatáro-
                             zása
  D47A  LD   DE,0E6D       ; DE-be a puffer címe
  D47D  LD   A,(0E6C)      ; A-ba a GR_MODE byte
  D480  LD   C,A
  D481  LD   B,0A
  D483  PUSH BC
  D484  PUSH HL
  D485  LD   B,00
  D487  LDIR               ; a kurzor alatti kép egy sorának az el-
                             mentése
  D489  POP  HL
  D48A  LD   C,40
  D48C  ADD  HL,BC         ; HL-be a következö képsor címe
  D48D  POP  BC
```

```
D48E  DJNZ F3          ; még nincs kész, ugrás a D483H címre
D490  RET              ; a rutin vége

A kurzor alatti kép visszaállítása:
D491  CALL D459        ; a kurzor címe a video RAM-ban
D494  LD   DE,0E6D     ; DE-be a kép puffer címe
D497  LD   A,(0E6C)    ; A-ba GR_MODE
D49A  LD   C,A
D49B  LD   B,0A
D49D  EX   DE,HL
D49E  PUSH BC
D49F  PUSH DE
D4A0  LD   B,00
D4A2  LDIR             ; egy képsor visszaállítása
D4A4  EX   DE,HL
D4A5  POP  HL
D4A6  LD   C,40
D4A8  ADD  HL,BC       ; HL-be a következő képsor címe
D4A9  POP  BC
D4AA  DJNZ F1          ; még nincs kész, ugrás a D49DH címre
D4AC  RET              ; a rutin vége

Sor képének törlése a kurzor pozíciótól a sor végéig:
D4AD  CALL D459        ; a kurzor címe a video RAM-ban
D4B0  LD   A,(0E96)    ; A-ba a paper byte
D4B3  EX   AF,AF'
D4B4  LD   A,3F
D4B6  SUB  L
D4B7  AND  3F
D4B9  JR   Z,18        ; a kurzor a sor végén van, ugrás a
                         D4D3H címre
D4BB  LD   C,A         ; C-be a karakter helyek száma a sor
                         végéig
D4BC  LD   A,L
D4BD  LD   B,0A
D4BF  AND  3F
D4C1  OR   L
D4C2  LD   L,A         ; HL-be a sor vége cím
D4C3  EX   AF,AF'
D4C4  LD   (HL),A      ; az adott karakter pozíció egy sorának
                         paper színűre festése
D4C5  LD   D,H
D4C6  LD   E,L
D4C7  INC  E
D4C8  PUSH BC
```

```
  D4C9  LD   B,00
  D4CB  LDIR              ; egy pixel sor paper színüre festése a
                            kurzortól a sor végéig
  D4CD  POP  BC
  D4CE  EX   DE,HL
  D4CF  EX   AF,AF'
  D4D0  DJNZ ED           ; nincs vége, ugrás a D4BFH címre
  D4D2  RET               ; a rutin vége

Egy karakter törlése a video RAM-ban:
  D4D3  LD   DE,0040
  D4D6  LD   B,0A
  D4D8  LD   A,(0E96)     ; A-ba a paper byte
  D4DB  LD   (HL),A       ; a karakter hely egy sorának paper szí-
                            nüre festése
  D4DC  ADD  HL,DE        ; HL-be a következö sor címe
  D4DD  DJNZ FC           ; még nincs vége, ugrás a D4DBH címre
  D4DF  RET               ; a rutin vége

Egy karakter sor törlése a video RAM-ban:
  D4E0  LD   A,C
  D4E1  CP   18
  D4E3  JR   Z,14         ; utolsó sor, ugrás a D4F9H címre
  D4E5  LD   B,01
  D4E7  CALL D459         ; az adott sor kezdetének címe
  D4EA  EX   DE,HL
  D4EB  LD   HL,B980      ; az utolsó sor címe
  D4EE  OR   A
  D4EF  SBC  HL,DE        ; HL-be az adott sor eleje és az utolsó
                            sor eleje közötti távolság a video
                            RAM-ban
  D4F1  LD   B,H
  D4F2  LD   C,L
  D4F3  LD   HL,0280
  D4F6  ADD  HL,DE        ; HL-be a következö sor elejének a címe
  D4F7  LDIR              ; az adott sor törlése, az alatta levö
                            sor felfelé mozgatása
A képernyö utolsó sorának törlése:
  D4F9  LD   HL,B980
  D4FC  LD   DE,B981
  D4FF  LD   BC,027F
  D502  LD   A,(0E96)
  D505  LD   (HL),A
  D506  LDIR
  D508  RET               ; a rutin vége
```

Egy sor beszúrása a képernyön:

```
D509  LD   A,18
D50B  SUB  C
D50C  JR   Z,EB        ; az utolsó sor, ugrás a D4F9H címre
D50E  LD   C,A
D50F  ADD  A,A
D510  ADD  A,A
D511  ADD  A,C
D512  ADD  A,A
D513  LD   B,A
D514  XOR  A
D515  SRL  B
D517  RRA
D518  SRL  B
D51A  RRA
D51B  LD   C,A         ; BC-be a kurzor sorának eleje és a kép-
                         ernyö vége közötti távolság a video
                         RAM-ban
D51C  LD   HL,B97F     ; az utolsó elötti sor végének a címe
D51F  LD   DE,BBFF     ; az utolsó sor végének a címe
D522  LDDR             ; egy karakter sornyival lefelé mozognak
                         a sorok a kurzor sorától
D524  INC  HL
D525  LD   D,H
D526  LD   E,L
D527  INC  DE
D528  JR   D5          ; ugrás a D4FFH címre: az üres sor paper
                         színüre festése
```

Egy karakter áthelyezése a képernyön:

```
D52A  LD   A,(0E6C)    ; A-ba GR_MODE
D52D  LD   C,A
D52E  LD   B,0A
D530  PUSH BC
D531  PUSH HL
D532  PUSH DE
D533  LD   B,00
D535  LDIR             ; a karakter egy sorának átmozgatása
D537  LD   C,40
D539  POP  HL
D53A  ADD  HL,BC
D53B  EX   DE,HL
D53C  POP  HL
D53D  ADD  HL,BC
D53E  POP  BC
```

```
  D53F  DJNZ EF          ; még nincs vége, ugrás a D530H címre
  D541  RET              ; a rutin vége

A DEL és DC rutin video RAM rutinja:
  D542  JR   Z,1E        ; sor vége, ugrás a D552H címre
  D544  PUSH HL
  D545  PUSH BC
  D546  LD   E,A
  D547  LD   A,(0B73)    ; A-ba GR_MODE
  D54A  OR   A
  D54B  JR   Z,05        ; két színü mód, ugrás a D552H címre
  D54D  SLA  E
  D54F  DEC  A
  D550  JR   NZ,FB
  D552  CALL D459        ; a kurzor címe
  D555  LD   C,E
  D556  LD   D,H
  D557  LD   A,(0E6C)    ; A-ba GR_MODE
  D55A  ADD  A,L
  D55B  LD   E,A
  D55C  EX   DE,HL
  D55D  CALL D52E        ; egy karakter átmozgatása
  D560  POP  BC
  D561  POP  HL
  D562  LD   B,(HL)
  D563  BIT  7,B
  D565  JR   Z,11        ; ugrás a D578H címre, ha a bekezdés nem
                           folytatódik a következö sorban
  D567  PUSH BC
  D568  RES  7,B
  D56A  CALL D459        ; a kurzor címe
  D56D  EX   DE,HL
  D56E  POP  BC
  D56F  INC  C
  D570  LD   B,01        ; BC-be a következö sor eleje
  D572  CALL D459        ; a kurzor címe
  D575  JP   D52A        ; egy karakter átmozgatása
SPACE rajzolása adott kurzor pozícióba:
  D578  CALL D459        ; a kurzor címe
  D57B  LD   C,0A
  D57D  LD   DE,0040
  D580  LD   A,(0E6C)    ; A-ba GR_MODE
  D583  LD   B,A
  D584  LD   A,(0E96)    ; A-ba paper byte
  D587  PUSH HL
```

```
  D588  LD   (HL),A
  D589  INC  HL
  D58A  DJNZ FC          ; nem kész a karakter sora, ugrás a
                           D588H címre
  D58C  POP  HL
  D58D  ADD  HL,DE
  D58E  DEC  C
  D58F  JR   NZ,EF       ; nem kész a karakter, ugrás a D580H-ra
  D591  RET              ; a rutin vége

Az INS rutin RAM video rutinja:
  D592  PUSH AF
  D593  LD   A,(HL)
  D594  AND  7F
  D596  SUB  B
  D597  JR   Z,35        ; sor vége, ugrás a D5CEH címre
  D599  LD   E,A
  D59A  LD   A,(0B73)    ; A-ba GR_MODE
  D59D  OR   A
  D59E  JR   Z,05        ; 2 színű mód, ugrás a D5A5H címre
  D5A0  SLA  E
  D5A2  DEC  A
  D5A3  JR   NZ,FB
  D5A5  PUSH HL
  D5A6  PUSH BC
  D5A7  LD   B,(HL)      ; B-be a sorhossz
  D5A8  RES  7,B
  D5AA  CALL D459        ; a kurzor címe
  D5AD  DEC  HL
  D5AE  LD   C,E
  D5AF  LD   B,00
  D5B1  EX   DE,HL
  D5B2  LD   A,(0E6C)    ; A-ba GR_MODE
  D5B5  LD   L,A
  D5B6  LD   H,B
  D5B7  ADD  HL,DE
  D5B8  EX   DE,HL
  D5B9  LD   B,0A
  D5BB  PUSH BC
  D5BC  PUSH HL
  D5BD  PUSH DE
  D5BE  LD   B,00
  D5C0  LDDR             ; a karakter egy sorának átmozgatása
  D5C2  LD   C,40
  D5C4  POP  HL
```

```
D5C5  ADD  HL,BC
D5C6  EX   DE,HL
D5C7  POP  HL
D5C8  ADD  HL,BC
D5C9  POP  BC
D5CA  DJNZ EF          ; nem kész, ugrás a D5BBH címre
D5CC  POP  BC
D5CD  POP  HL
D5CE  POP  AF
D5CF  JR   Z,A7        ; ugrás a D588H címre
D5D1  EX   DE,HL
D5D2  PUSH BC
D5D3  CALL D459        ; a kurzor címe
D5D6  POP  BC
D5D7  EX   DE,HL
D5D8  DEC  C
D5D9  DEC  HL
D5DA  LD   B,(HL)      ; B-be a sorhossz
D5DB  RES  7,B
D5DD  CALL D459        ; a kurzor címe
D5E0  JP   D52A
```

A billentyüzet rutinok ugrótáblája:

```
D5E3  DB   04      ; a rutinok száma
D5E4  DW   D62D    ; a KB_INT rutin címe
D5E6  DW   D618    ; a KB_CHIN rutin címe
D5E8  DW   D62C    ; a KB_BKIN rutin címe (a rutin egyetlen
                     RET utasítás
D5EA  DW   D612    ; a KB_STATUS rutin címe
```

A billentyüzet inicializáló rutinja:

```
D5EC  LD   A,1E
D5EE  LD   (0B65),A    ; a DELAY_KEY írása
D5F1  LD   A,03
D5F3  LD   (0B67),A    ; a RATE_KEY írása
D5F6  XOR  A
D5F7  LD   (0B66),A    ; LOCK_KEY alapállapotba
```

A PICTURE tábla törlése:

```
D5FA  LD   HL,0B51
D5FD  LD   DE,0B52
D600  LD   BC,0013
D603  LD   (HL),A
```

```
  D604  LDIR
A billentyüzet munkaterület törlése:
  D606  LD    HL,0BE5
  D609  LD    DE,0BE6
  D60C  LD    C,09
  D60E  LD    (HL),A
  D60F  LDIR
  D611  RET                 ; az inicializáló rutin vége

A KB_STATUS rutin:
  D612  LD    A,(0BE5)      ; A-ba a KB_STATUS_FLAG
  D615  LD    C,A
  D616  XOR   A
  D617  RET                 ; a KB_STATUS rutin vége

A KB_CHIN rutin:
  D618  LD    HL,0BE5
  D61B  LD    A,(HL)        ; A-ba a KB_STATUS_FLAG
  D61C  OR    A
  D61D  JR    Z,FC          ; nincs beolvasható billentyü, ugrás a
                              D61BH címre
  D61F  LD    A,(0BE9)      ; A-ba a beolvasható karakter
  D622  LD    C,A
  D623  LD    (HL),00       ; a KB_STATUS_FLAG törlése
  D625  LD    A,(0B16)      ; A-ba a STOP_FLAG
  D628  OR    A
  D629  RET   Z             ; a KB_CHIN rutin vége, ha nem volt
                              CTRL+ESC
  D62A  LD    A,F5          ; STOP hiba kód A-ba
  D62C  RET                 ; a KB_CHIN rutin vége CTRL+ESC -nél

A KB_INT rutin:
  D62D  LD    A,(0B66)      ; A-ba a LOCK_KEY
  D630  AND   0B
  D632  LD    B,A
  D633  NEG
  D635  AND   B
  D636  LD    (0B66),A      ; a LOCK_KEY írása
  D639  CALL  D7A5          ; a billentyüzet mátrix olvasása
  D63C  LD    HL,0BE7
  D63F  BIT   5,(IY+06H)
  D643  JR    Z,0D          ; nem LOCK lett nyomva, ugrás a D652H
                              címre
  D645  LD    A,(HL)
  D646  OR    A
```

```
D647  RET  NZ
D648  DEC  (HL)          ; (HL)-be FFH
D649  LD   A,01
D64B  CALL D790          ; a módosító billentyük vizsgálata
D64E  LD   (0B66),A      ; LOCK_KEY írása
D651  RET  NZ
D652  LD   (HL),B
D653  LD   A,04
D655  CALL D790          ; a módosító billentyük vizsgálata
D658  LD   (0BE8),A      ; a lock állapot elmentése
D65B  XOR  A
D65C  CALL D6C7          ; a beolvasható karakter meghatározása
D65F  INC  E
D660  JR   Z,10          ; ugrás a D672H címre
D662  LD   A,C
D663  LD   (0BE9),A      ; a beolvasott karakter eltárolása
D666  LD   A,FF
D668  LD   (0BE5),A      ; a KB_STATUS_FLAG beállítása
D66B  LD   A,(0B65)      ; A-ba DELAY_KEY
D66E  LD   (0BEA),A
D671  RET                ; a KB_INT rutin vége

D672  LD   A,(HL)
D673  OR   A
D674  JR   NZ,E6         ; ugrás a D65CH címre
D676  LD   HL,0BEA       ; HL-be a DELAY_KEY számláló címe
D679  CALL D6C0          ; DELAY vizsgálat
D67C  INC  HL
D67D  CALL NC,D6C0       ; RATE_KEY vizsgálat
D680  RET  NC
D681  LD   A,(0BEE)
D684  LD   C,A
D685  LD   DE,(0BEC)
D689  LD   HL,0B64       ; HL-be OLDPICT végének a címe
D68C  SBC  HL,DE
D68E  EX   DE,HL
D68F  JR   Z,0A          ; ugrás a D69BH címre
D691  OR   A
D692  EX   DE,HL
D693  LD   HL,0B64       ; HL-be OLDPICT végének a címe
D696  SBC  HL,DE
D698  EX   DE,HL
D699  JR   NZ,09         ; ugrás a D6A4H címre
D69B  LD   D,66
D69D  AND  D
```

```
  D69E  JR   Z,04          ; ugrás a D6A4H címre
  D6A0  XOR  D
  D6A1  AND  (HL)
  D6A2  JR   NZ,11         ; ugrás a D6B5H címre
  D6A4  LD   A,C
  D6A5  AND  (HL)
  D6A6  JR   NZ,0D         ; ugrás a D5B5H címre
OLDPICT törlése:
  D6A8  LD   HL,0B5B       ; HL-be OLDPICT elejének a címe
  D6AB  LD   B,0A
  D6AD  LD   (HL),A
  D6AE  INC  HL
  D6AF  DJNZ FC            ; ugrás a D6ADH címre
  D6B1  LD   (0BEB),A
  D6B4  RET
  D6B5  CPL
  D6B6  AND  (HL)
  D6B7  LD   (HL),A
  D6B8  LD   A,(0B67)      ; A-ba RATE_KEY
  D6BB  LD   (0BEB),A      ; a RATE_KEY számláló töltése
  D6BE  JR   9B            ; ugrás a D65BH címre
A RATE_KEY és a DELAY_KEY számláló csökkentése:
  D6C0  LD   A,(HL)
  D6C1  OR   A
  D6C2  RET  Z             ; a számláló 0 volt
  D6C3  DEC  (HL)
  D6C4  RET  NZ
  D6C5  SCF                ; a számláló 0 lett
  D6C6  RET
A megnyomott billentyűhöz tartozó karakter kódjának meghatáro-
zása:
  D6C7  CALL NZ,D7A5       ; a billentyüzet mátrix leolvasása
A SHIFT, CTRL, ALT és a LOCK törlése a mátrixban:
  D6CA  RES  3,(IY+06H)
  D6CE  RES  4,(IY+07H)
  D6D2  RES  0,(IY+07H)
  D6D6  RES  5,(IY+06H)
  D6DA  LD   HL,0BE6
  D6DD  LD   DE,000A
  D6E0  EXX
  D6E1  LD   HL,0B65
  D6E4  LD   DE,0B5B
  D6E7  DEC  DE            ; DE-be a PICTURE kezdö címe
  D6E8  DEC  HL            ; HL-be az OLDPICT kezdö címe
  D6E9  EXX
```

```
D6EA  DEC  E
D6EB  RET  M
D6EC  EXX
D6ED  LD   A,(DE)
D6EE  LD   B,A
D6EF  XOR  (HL)
D6F0  CPL
D6F1  AND  (HL)
D6F2  LD   (HL),A
D6F3  XOR  B
D6F4  JR   Z,F1          ; ugrás a D6E7H címre
D6F6  LD   B,A
D6F7  NEG
D6F9  AND  B
D6FA  LD   B,A
D6FB  XOR  (HL)
D6FC  LD   (HL),A
D6FD  LD   A,B
D6FE  LD   (0BEC),HL
D701  LD   (0BEE),A
D704  LD   A,50
D706  SUB  0A
D708  SRL  B
D70A  JR   NC,FA         ; ugrás a D706H címre
D70C  EXX
D70D  ADD  A,E
D70E  LD   E,A
D70F  LD   A,(0BE8)
D712  LD   B,A
D713  LD   A,(0B66)      ; A-ba LOCK_KEY
D716  XOR  B
D717  LD   B,A
D718  PUSH HL
D719  CALL D747          ; billentyüzet dekódolás
D71C  POP  HL
D71D  LD   A,(0B68)      ; A-ba HOLD_DIS
D720  INC  A
D721  RET  Z             ; CTRL+P hatástalan
D722  LD   A,(HL)
D723  OR   A
D724  LD   A,C           ; A-ba a beolvasott karakter kódja
D725  JR   Z,15          ; ugrás a D73CH címre
D727  LD   A,(0B11)      ; A-ba PORT03
D72A  AND  F0
D72C  OR   07
```

```
D72E  OUT  03,A        ; a billentyüzet mátrix 7. sorának ki-
                         választása
D730  IN   A,58
D732  AND  18
D734  JR   Z,0A        ; CTRL+ESC-t nyomtak, ugrás a D740H-ra
D736  XOR  A           ; A-ba 0
D737  LD   (0B16),A    ; STOP_FLAG írása
D73A  JR   04          ; ugrás a D740H címre
D73C  CP   10
D73E  JR   Z,03        ; CTRL+P, ugrás a D743H címre
D740  LD   (HL),00
D742  RET
D743  DEC  (HL)
D744  LD   E,FF
D746  RET

A megnyomott billentyü dekódolása:
D747  LD   HL,D784
D74A  PUSH HL          ; visszatérési cím a STACK-be
D74B  LD   HL,D85F     ; HL-be a CTRL mátrix címe
D74E  BIT  2,B
D750  RET  NZ          ; ugrás a D784H címre
D751  BIT  0,B
D753  JR   Z,1F        ; ugrás a D774H címre
D755  LD   HL,D7BF     ; HL-be az alap mátrix címe
D758  ADD  HL,DE
D759  LD   A,(HL)
D75A  CP   61
D75C  JR   C,16        ; a kód kisebb 97-nél, ugrás a D774H-ra
D75E  CP   7B
D760  JR   C,08        ; a kód kisebb 123-nál, ugrás a D76AH-ra
D762  CP   90
D764  JR   C,0E        ; a kód kisebb 144-nél, ugrás a D774H-ra
D766  CP   99
D768  JR   NC,0A       ; a kód nagyobb 152-nél, ugrás a D774H
                         címre
D76A  LD   HL,D80F     ; HL-be a SHIFT mátrix címe
D76D  BIT  1,B
D76F  RET  Z           ; ugrás a D784H címre
D770  LD   HL,D7BF     ; HL-be az alap mátrix címe
D773  RET              ; ugrás a D784H címre
D774  LD   HL,D80F     ; HL-be a SHIFT mátrix címe
D777  BIT  1,B
D779  RET  NZ          ; ugrás a D784H címre
D77A  LD   HL,D8AF     ; HL-be az ALT mátrix címe
```

```
  D77D  BIT  3,B
  D77F  RET  NZ           ; ugrás a D784H címre
  D780  POP  HL
  D781  LD   HL,D7BF      ; HL-be az alap mátrix címe
  D784  ADD  HL,DE
  D785  LD   C,(HL)       ; C-be a karakter kódja
  D786  LD   A,C
  D787  CP   FF
  D789  RET  NZ           ; jó karakter kód, a dekódoló rutin vége
  D78A  LD   (0B16),A     ; a STOP_FLAG beállítása
  D78D  LD   C,1B         ; C-be ESC kódja
  D78F  RET               ; STOP-ot nyomtak, a dekódoló rutin vége

A módosító billentyük vizsgálata:
  D790  BIT  4,(IY+07H)
  D794  RET  NZ           ; CTRL-t nyomtak
  D795  LD   A,02
  D797  BIT  3,(IY+06H)
  D79B  RET  NZ           ; SHIFT-t nyomtak
  D79C  BIT  0,(IY+07H)
  D7A0  LD   A,08
  D7A2  RET  NZ           ; ALT-t nyomtak
  D7A3  XOR  A
  D7A4  RET               ; nem nyomtak módosító billentyüt

A billentyüzet mátrix letapogatása:
  D7A5  LD   A,(0B11)     ; A-ba PORT03
  D7A8  AND  F0
  D7AA  LD   C,A
  D7AB  LD   HL,0B51      ; HL-be a PICTURE címe
  D7AE  PUSH HL
  D7AF  POP  IY
  D7B1  LD   B,0A
  D7B3  LD   A,C
  D7B4  OUT  03,A         ; a mátrix sorának kiválasztása
  D7B6  IN   A,58
  D7B8  CPL
  D7B9  LD   (HL),A       ; a PICTURE töltése
  D7BA  INC  C
  D7BB  INC  HL
  D7BC  DJNZ F5           ; ugrás a D7B3H címre
  D7BE  RET               ; a rutin vége
```

A billentyüzet mátrix dekódoló táblázatai:

Az alapállapotú billentyüzet mátrix:

```
D7BF  DB   34 37 72 75 66 6A 76 6D 2A 2A
D7C9  DB   31 94 71 70 61 91 79 2D 13 F3
D7D3  DB   92 93 40 96 3C 98 2A 20 04 E4
D7DD  DB   36 2A 7A 5B 68 0D 6E 2A 01 E1
D7E7  DB   30 97 3B 95 5C 90 2A 1B 06 E6
D7F1  DB   32 39 77 6F 73 6C 78 2E 18 F8
D7FB  DB   33 38 65 69 64 6B 63 2C 05 E5
D805  DB   35 5E 74 5D 67 08 62 2A 16 43
```

A SHIFT billentyüzet mátrix:

```
D80F  DB   21 3D 52 55 46 4A 56 4D 2A 2A
D819  DB   27 84 51 50 41 81 59 5F 13 F3
D823  DB   82 83 60 86 3E 88 2A 20 04 E4
D82D  DB   2F 23 5A 7B 48 0D 4E 2A 01 E1
D837  DB   26 87 24 85 7C 80 2A 1B 06 E6
D841  DB   22 29 57 4F 53 4C 58 3A 18 F8
D84B  DB   2B 28 45 49 44 4B 43 3F 05 E5
D855  DB   25 7E 54 7D 47 07 42 2A 16 49
```

A CTRL billentyüzet mátrix:

```
D85F  DB   8B 9C 12 15 06 0A 16 0D 2A 2A
D869  DB   99 DC 11 10 01 D9 19 1F 13 F3
D873  DB   DA DB 00 DE 3C 98 2A 20 04 E4
D87D  DB   8C 8E 1A 1B 08 0D 0E 2A 01 E1
D887  DB   89 DF 3B DD 1C CF 2A FF 06 E6
D891  DB   8A 9D 17 0F 13 0C 18 2E 18 F8
D89B  DB   9A 8D 05 09 04 0B 03 2C 05 E5
D8A5  DB   9B 1E 14 1D 07 08 02 2A 16 53
```

Az ALT billentyüzet mátrix:

```
D8AF  DB   A4 A7 C2 C5 B6 BA C6 BD 2A 2A
D8B9  DB   A1 D4 C1 C0 B1 D1 C9 AD 13 F3
D8C3  DB   D2 D3 B0 D6 AC D8 2A 20 04 E4
D8CD  DB   A6 AA CA CB B8 0D BE 2A 01 E1
D8D7  DB   A0 D7 AB D5 CC D0 2A 1B 06 E6
```

```
  D8E1  DB   A2 A9 C7 BF C3 BC C8 AE 18 F8
  D8EB  DB   A3 A8 B5 B9 B4 BB B3 AF 05 E5
  D8F5  DB   A5 CE C4 CD B7 08 B2 2A 16 4C
```

A printer rutinok ugrótáblája:

```
  D8FF  DB   03      ; a rutinok száma
  D900  DW   D929    ; a PAR_INT rutin címe
  D902  DW   D90C    ; a PAR_CHOUT rutin címe
  D904  DW   D906    ; a PAR_BKOUT rutin címe
```

```
A PAR_BKOUT rutin:
  D906  LD   HL,D90C     ; HL-be a PAR_CHOUT rutin címe
  D909  JP   C56D        ; HIMEM átlépés ellenörzés
A PAR_CHOUT rutin:
  D90C  LD   A,(0B16)    ; A-ba STOP_FLAG
  D90F  INC  A
  D910  LD   A,F5        ; A-ba a hiba kód
  D912  RET  Z           ; STOP-ot nyomtak, a rutin vége
  D913  IN   A,59
  D915  RLCA
  D916  JR   NC,F4       ; nem érkezett ACK jel, ugrás a D90CH
                           címre
  D918  DI               ; megszakítás tíltás
  D919  LD   A,C
  D91A  OUT  01,A        ; karakter kiküldése a nyomtatóra
STROBE jel kiküldése:
  D91C  LD   A,(0B13)    ; PORT06 A-ba
  D91F  AND  7F
  D921  OUT  06,A
  D923  OR   80
  D925  OUT  06,A

  D927  EI               ; megszakítás engedélyezése
  D928  XOR  A
  D929  RET              ; a printer rutinok vége
```

A hangkezelő rutinok ugrótáblája:

```
  D92A  DB   04      ; a rutinok száma
  D92B  DB   D933    ; TONE_INT rutin címe
  D92D  DB   D960    ; TONE_CH rutin címe
```

```
D92F  DB   D960    ; TONE_BK rutin címe
D931  DB   D961    ; TONE_SET rutin címe

A TON_INT rutin:
D933  LD   A,(0B14)    ; A-ba SND_ACTIVE
D936  INC  A
D937  RET  NZ          ; nincs hang folyamatban
D938  LD   A,(0B16)    ; A-ba STOP_FLAG
D93B  INC  A
D93C  JR   Z,08        ; STOP, ugrás a D946H címre
D93E  LD   A,(0BEF)
D941  DEC  A
D942  LD   (0BEF),A
D945  RET  NZ
D946  LD   (0B14),A    ; SND_ACTIVE törlése
D949  LD   (0BEF),A    ; a hang munkaterület törlése
D94C  LD   A,(0B10)    ; A-ba INT_DES
D94F  OR   08
D951  LD   (0B10),A    ; INT_DES írása
D954  LD   A,(0B12)    ; A-ba PORT05
D957  AND  CF
D959  LD   (0B12),A    ; PORT05 írása
D95C  OUT  05,A
D95E  LD   A,F5        ; A-ba a hibakód
D960  RET              ; a TONE_INT rutin vége

A TONE_SET rutin:
D961  LD   A,(0B15)    ; A-ba SND_IRQ
D964  INC  A
D965  JR   Z,07        ; a folyamatban lévö hang megszakad,
                         ugrás a D96EH címre
D967  LD   A,(0BEF)
D96A  SUB  02
D96C  JR   NC,F3       ; a folyamatban lévö hang nem ért vé-
                         get, ugrás a D961H címre
D96E  XOR  A
D96F  LD   (0B14),A    ; az elözö hang befejezödött
D972  LD   A,(0B16)    ; A-ba STOP_FLAG
D975  INC  A
D976  JR   Z,CE        ; STOP, ugrás a D946H címre
D978  LD   A,B         ; A-ba az idötartam
D979  OR   A
D97A  JR   NZ,05       ; ugrás a D981H címre, van hang
D97C  CALL D946
D97F  XOR  A
```

```
D980  RET
D981  LD   (0BEF),A    ; a hang időtartam számláló töltése
D984  LD   A,C         ; A-ba az amplitudó
D985  AND  0F
D987  RLCA
D988  RLCA
D989  LD   C,A
D98A  LD   A,(0B13)    ; A-ba PORT06
D98D  AND  C3
D98F  OR   C
D990  LD   (0B13),A    ; a PORT06 új értékének elmentése
D993  OUT  06,A
D995  LD   A,D         ; A-ba PITCH felső 4 bitje
D996  AND  0F
D998  OR   10
D99A  LD   D,A
D99B  LD   A,(0B12)    ; A-ba PORT05
D99E  AND  C0
D9A0  OR   D
D9A1  LD   (0B12),A    ; PORT05 új értékének elmentése
D9A4  OUT  05,A
D9A6  LD   A,E         ; A-ba PITCH alsó 8 bitje
D9A7  OUT  04,A
D9A9  LD   A,FF
D9AB  LD   (0B14),A    ; SND_ACTIVE beállítása
D9AE  LD   (0B71),A    ; SER_OK beállítása
D9B1  LD   A,(0B10)    ; A-ba INT_DES
D9B4  AND  F7
D9B6  LD   (0B10),A    ; INT_DES írása
D9B9  XOR  A
D9BA  RET              ; a TONE_SET rutin vége
```

A kazetta rutinok ugrótáblája:

```
D9BB  DB   06      ; a rutinok száma
D9BC  DB   D9E7    ; CAS_INT rutin címe
D9BE  DB   D9D2    ; CAS_CH rutin címe
D9C0  DB   D9D7    ; CAS_BK rutin címe
D9C2  DB   D9C8    ; CAS_OPEN rutin címe
D9C4  DB   D9CD    ; CAS_CLOSE rutin címe
D9C6  DB   D9DC    ; CAS_VERIFY rutin címe
```

```
A CAS_OPEN rutinok:
  D9C8  LD   HL,F3E2
  D9CB  JR   12          ; ugrás a D9DFH címre
A CAS_CLOSE rutinok:
  D9CD  LD   HL,F3E7
  D9D0  JR   0D          ; ugrás a D9DFH címre
A CAS_CH rutinok:
  D9D2  LD   HL,F3D8
  D9D5  JR   08          ; ugrás a D9DFH címre
A CAS_BK rutinok:
  D9D7  LD   HL,F3DD
  D9DA  JR   03          ; ugrás a D9DFH címre
A CAS_VERIFY rutin:
  D9DC  LD   HL,F3EC
  D9DF  JP   FFF0
A kazetta munkaterület törlése:
  D9E2  LD   HL,F3F1
  D9E5  JR   F8          ; ugrás a D9DFH címre
  D9E7  RET              ; a kazetta rutinok vége


A rendszer inicializálás folytatása:
  D9E8  DB   '(c) ISL'
  D9EF  LD   HL,1700
  D9F2  PUSH HL
  D9F3  POP  IX
  D9F5  LD   A,(0B21)    ; A-ba WARM_FLAG
  D9F8  OR   A
  D9F9  JP   NZ,DAD3     ; meleg reset van folyamatban, ugrás

Csak hideg reset esetén  hajtódik végre:

A 1700H-19EFH RAM terület törlése:
  D9FC  LD   BC,02EF
  D9FF  LD   (HL),A
  DA00  CPI
  DA02  JP   PE,D9FF

  DA05  LD   (1720),HL   ; VLOMEM-be 19EFH
  DA08  LD   (1722),HL   ; TEXT-be 19EFH

RAM terület töltése a ROM-ból:
  DA0B  LD   HL,FB5B
  DA0E  LD   DE,0008
  DA11  LD   BC,0027
  DA14  LDIR
```

```
  DA16  CALL DE10        ; NEW parancs
  DA19  LD   DE,DC15
  DA1C  RST  30H
  DA1D  DB   0C          ; a PAL rutin
  DA1E  LD   A,02
  DA20  LD   (0B4F),A    ; a BORDER változó írása
  DA23  LD   (IX+05H),00

A bejelentkező villogó grafika megvalósítása:
  DA27  LD   BC,0303     ; sor-oszlop pozíció a BTEXT rutinhoz
  DA2A  LD   A,0C
  DA2C  PUSH BC
  DA2D  PUSH AF
  DA2E  RST  30H
  DA2F  DB   03          ; BTEXT rutin
  DA30  CALL DBF2        ; kiíratás a képernyöre
  DA33  POP  AF
  DA34  PUSH AF
  DA35  CALL DC0C
  DA38  CALL DBF2        ; kiíratás a képernyöre
  DA3B  CALL FE93        ; "soremelés-kocsivissza"
  DA3E  POP  AF
  DA3F  POP  BC
  DA40  INC  B
  DA41  INC  C
  DA42  INC  C
  DA43  SUB  02
  DA45  JR   NC.E5       ; még nem érte el a grafika alját. ugrás
                           a DA2CH címre
  DA47  LD   BC.0D15     ; sor-oszlop pozíció
  DA4A  RST  30H
  DA4B  DB   03          ; BTEXT rutin
  DA4C  CALL DBF2        ; kiíratás  a képernyöre
Az alsó villogó szöveg:
  DA4F  LD   BC,0B12     ; sor-oszlop pozíció
  DA52  RST  30H
  DA53  DB   03          ; BTEXT rutin
  DA54  LD   HL,DBFF     ; HL-be a string címe
  DA57  LD   B,(HL)      ; B-be a string hossza
  DA58  INC  HL
  DA59  LD   A,(HL)      ; A-ba a következö karakter
  DA5A  CALL FE9A        ; A kiíratása a képernyöre
  DA5D  PUSH HL
  DA5E  PUSH BC
  DA5F  CALL E6D8        ; véletlen szám generálás
```

```
DA62  AND  03
DA64  JR   Z,F9          ; új tintaszínt, ugrás a DA5FH címre
DA66  POP  BC
DA67  POP  HL
DA68  LD   (0B4D),A      ; INK elmentése
DA6B  DEC  C
DA6C  JR   NZ,FD         ; késleltetés
DA6E  DJNZ E8            ; nem kész a string, ugrás a DA58H címre
DA70  RST  30H
DA71  DB   93            ; KBD_STATUS rutin
DA72  INC  C
DA73  JR   NZ,DA         ; nincs billentyü nyomás, ugrás a DA4FH
                           címre
DA75  RST  30H
DA76  DB   91            ; KBD_CHIN rutin
DA77  LD   C,01          ; 4 színü üzemmód
DA79  RST  30H
DA7A  DB   04            ; VMODE rutin
DA7B  LD   (0B4F),A      ; BORDER elmentése
DA7E  CALL FC18          ; az INK és PAPER összehasonlítása
DA81  CALL FE79          ; az interpreter bejelenkezö szövegének
                           kiíratása

DA84  DB   32
DA85  DB   'TV COMPUTER'
DA90  DB   ' BASIC 1.2'
DA9A  DB   0D 0A
DA9C  DB   'Copyright '
DAA6  DB   '1985 VIDEOTON'
DAB3  DB   0D 0A 0D 0A

DAB7  CALL EC4C          ; FREE függvény
DABA  CALL FEC1          ; a BASIC STACK-ben lévö szám kiíratása
DABD  CALL FA1B          ; 9 byte felszabadítása a BASIC STACK-
                           böl
DAC0  CALL FE79          ; az utasítást követö szöveg kiíratása

DAC3  DB   0F
DAC4  DB   'bytes free'
DACF  DB   0D 0A 0D 0A
```

A BASIC interpreter kezdete

```
DAD3  XOR  A
DAD4  LD   (0B21),A     ; a WARM_FLAG törlése
DAD7  CALL DCFC         ; BASIC változók alapba állítása
DADA  LD   SP,16AC
DADD  LD   HL,1703      ; HL-be NEW címe
DAE0  LD   A,(HL)       ; A-ba NEW
DAE1  OR   A
DAE2  LD   (HL),00      ; NEW törlése
DAE4  CALL NZ,DE10      ; NEW rutin
DAE7  LD   (IX+05H),20
DAEB  RES  2,(IX+00H)
DAEF  BIT  1,(IX+00H)
DAF3  JR   NZ,0A        ; ugrás a DAFFH címre
DAF5  CALL FC18
DAF8  CALL FE79         ; az "ok" kiírása


DAFB  DB   03
DAFC  DB   'ok'
DAFE  DB   0B

DAFF  RES  1,(IX+00H)
DB03  CALL FE93         ; "soremelés-kocsivissza"

Az OK utasítás belépési pontja:
  DB06  CALL FF4F       ; az interpreter input rutinja
  DB09  CP   F5
  DB0B  JP   Z,FFA3     ; STOP lett nyomva, ugrás
  DB0E  CP   EC
  DB10  JR   Z,E3       ; file vége jel, ugrás a DAF5H címre

A begépelt szöveg értelmezése, tokenizálása:
  DB12  CALL DC19       ; az adott sor tömörítése
  DB15  LD   E,L
  DB16  LD   D,H        ; DE-be HL
  DB17  INC  HL
  DB18  CALL F914
  DB1B  JR   C,0D       ; ugrás a DB2AH címre
  DB1D  LD   A,(HL)
  DB1E  CP   A8
  DB20  JR   Z,03       ; ugrás a DB25H címre
  DB22  INC  A
  DB23  JR   NZ,3C      ; ugrás a DB61H címre
```

```
DB25  CALL FA1B       ; 9 byte felszabadítása a BASIC STACK-ből
DB28  JR   DC         ; ugrás a DB06H címre
DB2A  RLA
DB2B  JR   C,34       ; A legfelső bitje 1 volt, ugrás a DB61H
                        címre
DB2D  LD   B,(HL)
DB2E  LD   A,(IY+08H)
DB31  CP   40
DB33  JR   C,2C       ; A<40H, ugrás a DB61H címre
DB35  CP   44
DB37  JR   NC,28      ; A>43H, ugrás a DB61H címre
DB39  LD   A,(IY+06H)
DB3C  OR   A
DB3D  JR   Z,22       ; A=0, ugrás a DB61H címre
DB3F  LD   A,(HL)
DB40  INC  HL
DB41  CP   20
DB43  JR   Z,FA       ; ha SPACE, a következő byte
DB45  DEC  HL
DB46  EX   DE,HL
DB47  LD   A,(HL)
DB48  INC  HL
DB49  ADD  A,02
DB4B  SBC  HL,DE
DB4D  ADD  A,L
DB4E  LD   C,A
DB4F  CALL FAC3
DB52  EX   DE,HL
DB53  DEC  HL
DB54  LD   (HL),D
DB55  DEC  HL
DB56  LD   (HL),E     ; a programsor sorszámának beírása
DB57  DEC  HL
DB58  LD   (HL),C     ; a programsor hossza C-ből
DB59  LD   (170C),HL  ; START írása
DB5C  CALL DCAF
DB5F  JR   A5         ; ugrás a DB06H címre

DB61  EX   DE,HL
DB62  CALL FA1B       ; 9 byte felszabadítása a BASIC STACK-ből
DB65  LD   C,(HL)
DB66  XOR  A
DB67  LD   (HL),A
DB68  DEC  HL
DB69  LD   (HL),A
```

```
  DB6A  DEC  HL
  DB6B  INC  C
  DB6C  INC  C
  DB6D  LD   (HL),C       ; az adott utasítás sor hossza C-böl
  DB6E  LD   (170C),HL    ; START elmentése
  DB71  INC  HL           ; HL a sor sorszámára mutat
  DB72  BIT  0,(IX+00H)
  DB76  CALL NZ,DE4D      ; a nyomkövetés bekapcsolva (TRACE ON),
                            ugrás
  DB79  RES  0,(IX+02H)
  DB7D  INC  HL
  DB7E  INC  HL           ; HL az elsö utasítás byte-ra mutat
  DB7F  EXX

A soronkövetkezö utasítás értelmezése:
  DB80  EXX
  DB81  CALL FF9D         ; STOP vízsgálat
  DB84  LD   (171A),IY
  DB88  LD   A,(HL)       ; A-ba az utasítás byte
  DB89  CP   FE
  DB8B  JR   NC,2E        ; az utasítás "!" vagy sor vége jel, ug-
                            rás a DBBBH címre
  DB8D  INC  HL           ; HL-be a következö utasítás byte címe
  DB8E  CP   20
  DB90  JR   Z,F6         ; a SPACE-k figyelmen kívül hagyása
  DB92  CP   CA
  DB94  JP   C,E3BC       ; "másod" utasítás, ugrás
  DB97  PUSH AF
  DB98  EXX
Az utasítás rutin címének meghatározása a tokenböl:
  DB99  CPL
  DB9A  ADD  A,A
  DB9B  ADD  A,67
  DB9D  LD   L,A
  DB9E  LD   H,C0
  DBA0  LD   A,(HL)
  DBA1  INC  L
  DBA2  LD   H,(HL)
  DBA3  LD   L,A          ; HL-be az utasítás rutinjának a címe
  DBA4  POP  AF           ; A-ba a token
  DBA5  CP   FB
  DBA7  CALL C,FC43       ; nem "üres" utasítás, ugrás
  DBAA  LD   SP,16AC
  DBAD  JP   (HL)         ; ugrás az utasítást megvalósító rutinra
```

```
  DBAE  CALL FC43        ; a következö program byte B-be
  DBB1  EXX
  DBB2  LD   A,B
  DBB3  EXX
  DBB4  CP   FD
  DBB6  JR   Z,C8        ; ":", ugrás a DB80H címre
  DBB8  JP   C,FD5A      ; szintaktikai hiba, ugrás

REM, !, sor vége utasítások belépési pontja:
  DBBB  LD   HL,(170C)   ; HL-be a START cím
  DBBE  LD   C,(HL)      ; C-be az utasítás sor hossza
  DBBF  XOR  A
  DBC0  LD   B,A
  DBC1  ADD  HL,BC       ; HL-be a sor vége utáni byte címe
  DBC2  OR   (HL)
  DBC3  JR   NZ,A9       ; van következö BASIC sor, ugrás a DB6EH
                           címre
  DBC5  LD   A,(IY+00H)
  DBC8  CP   2B
  DBCA  JR   Z,0E        ; ugrás a DBDAH címre, FOR-hoz tartozó
                           cím a BASIC STACK-ben
  DBCC  CP   06
  DBCE  JR   Z,0A        ; ugrás a DBDAH címre, GOSUB-hoz tartozó
                           cím a BASIC STACK-ben
  DBD0  BIT  2,(IX+00H)
  DBD4  JP   Z,DADA      ; parancs mód, ugrás
  DBD7  JP   E10E        ; END rutin
  DBDA  LD   E,(IY+03H)
  DBDD  LD   D,(IY+04H)  ; DE-be a GOSUB-t tartalmazó sor címe
  DBE0  LD   HL,1831     ; HL-be a BUFFER címe
  DBE3  SBC  HL,DE
  DBE5  JR   C,E9        ; ugrás a DBD0H címre, ha DE>HL
  DBE7  LD   C,A
  DBE8  LD   B,00
  DBEA  ADD  IY,BC       ; az elözö bejegyzés a BASIC STAC-ben
  DBEC  LD   (171A),IY
  DBF0  JR   D3          ; ugrás a DBC5H címre

Rutin a bejelentkezö villogó grafikához:
  DBF2  CALL FE79        ; az utasítást követö string kiíratása
  DBF5  DB   08
  DBF6  DB   'VIDEOTON'
  DBFE  RET

  DBFF  DB   0C
  DC00  DB   'TV  COMPUTER'
```

```
DC0C  DEC  A
DC0D  RET  M
DC0E  PUSH AF
DC0F  CALL FEC7        ; SPACE kiíratás
DC12  POP  AF
DC13  JR   F7          ; ugrás a DC0CH címre

DC15  DB   01 44 54 51; paletta byte-ok
```

A begépelt utasítás sorozat tömörítése, tokenizálása:

```
DC19  PUSH IY
DC1B  EX   DE,HL
DC1C  LD   IY,1735     ; IY-ba COMMAND címe +3
DC20  LD   BC,0000
DC23  INC  DE          ; DE a BUFFER-be mutat
DC24  LD   HL,DE6E     ; HL-be a BASIC kulcsszó tábla címe
DC27  LD   A,FE
DC29  EX   AF,AF'
DC2A  INC  IY
DC2C  LD   (171C),DE
DC30  CALL FBBF        ; A-ba a BUFFER akturális byte-ja
DC33  LD   (IY+00H),A
DC36  INC  A
DC37  JR   Z,66        ; ugrás a DC9FH címre, ha a BUFFER u-
                         tolsó beírt byte-ja
DC39  LD   A,B
DC3A  OR   A
DC3B  JR   Z,04        ; ugrás a DC41H címre, ha 0
DC3D  CP   3A
DC3F  JR   NZ,0A       ; ugrás a DC4BH címre
DC41  LD   A,(DE)      ; A-ba egy byte a BUFFER-ből
DC42  CP   22
DC44  JR   NZ,05       ; nem " karakter, ugrás a DC4BH címre
DC46  LD   C,B
DC47  LD   B,A
DC48  INC  DE
DC49  JR   DF          ; ugrás a DC2AH címre
DC4B  CALL FBBF        ; A-ba a BUFFER aktuális byte-ja
DC4E  INC  DE
DC4F  OR   A
DC50  JR   Z,0E        ; ugrás a DC60H címre
DC52  CP   B
DC53  JR   NZ,0B       ; ugrás a DC60H címre
DC55  LD   B,C
DC56  LD   C,00
```

```
DC58  CP   22
DC5A  JR   Z,C8          ; ugrás a DC24H címre, ha " karakter
DC5C  LD   A,FD
DC5E  JR   28            ; ugrás a DC88H címre

DC60  INC  B
DC61  DEC  B
DC62  JR   NZ,C6         ; B nem 0, ugrás a DC2AH címre
DC64  XOR  (HL)
DC65  ADD  A,A
DC66  JR   NZ,25         ; A és (HL) a legfelsö bittöl eltekintve
                           nem egyenlö, ugrás a DC8DH címre
DC68  INC  HL
DC69  JR   NC,E0         ; ugrás a DC4BH címre, ha A és (HL) meg-
                           egyezett
DC6B  EX   AF,AF'
DC6C  LD   C,B
DC6D  CP   A6
DC6F  JR   Z,08          ; ugrás a DC79H címre, ha a token A6
DC71  CP   A5
DC73  JR   Z,04          ; ugrás a DC79H címre, ha a token A5
DC75  CP   A3
DC77  JR   NZ,02         ; ugrás a DC7BH címre, ha a token nem A3
DC79  SUB  08
DC7B  CP   FD
DC7D  JR   Z,09          ; ugrás a DC88H címre, ha a token FD
DC7F  CP   FB
DC81  JR   C,05          ; ugrás a DC88H címre, ha a token< FB
DC83  LD   B,A
DC84  JR   NZ,02         ; ugrás a DC88H címre
DC86  LD   B,3A
DC88  LD   (IY+00H),A    ; a tömörített sor egy byte-jának eltá-
                           rolása a COMMAND területen
DC8B  JR   97            ; ugrás a DC24H címre

DC8D  LD   DE,(171C)
DC91  BIT  7,(HL)
DC93  INC  HL
DC94  JR   Z,FB
DC96  EX   AF,AF'
DC97  DEC  A             ; A-ban a HL által címzett kulcsszó to-
                           kenje
DC98  EX   AF,AF'
DC99  LD   A,(HL)        ; A-ba a kulcsszó tábla aktuális eleme
DC9A  INC  A
```

```
  DC9B  JR   Z,86          ; a kulcsszó tábla vége, ugrás a DC23H
                             címre
  DC9D  JR   AC            ; ugrás a DC4B címre
A sorhossz byte meghatározása:
  DC9F  PUSH IY
  DCA1  POP  HL
  DCA2  INC  HL
  DCA3  LD   (HL),A        ; sor vége jel elhelyezése
  DCA4  LD   DE,1735
  DCA7  OR   A
  DCA8  SBC  HL,DE
  DCAA  EX   DE,HL
  DCAB  LD   (HL),E        ; a sorhossz byte eltárolása
  DCAC  POP  IY
  DCAE  RET                ; a tömörítő-tokenizáló rutin vége

A begépelt programsor beillesztése a programba:
  DCAF  LD   (171C),HL
  DCB2  PUSH BC
  DCB3  CALL DD45          ; az adott sorszámú BASIC sor helyének
                             megkeresése
  DCB6  CALL NC,DCE6       ; már van ilyen számú sor, ugrás
  DCB9  POP  BC
  DCBA  INC  B
  DCBB  RET  Z
  DCBC  LD   B,00
  DCBE  CALL DCC9          ; szabad hely ellenörzés, szabad hely
                             "csinálás" a sornak
  DCC1  EX   DE,HL
  DCC2  LD   HL,(171C)
  DCC5  LDIR               ; az adott programsor átmozgatása á
                             COMMAND pufferből a BASIC területre
  DCC7  JR   33            ; ugrás a DCFCH címre

  DCC9  CALL DCFC          ; BASIC rendszer változók írása
  DCCC  CALL FC8E          ; szabad hely ellenörzés
  DCCF  PUSH BC
  DCD0  PUSH HL
  DCD1  CALL DD41          ; a program végének meghatározása
  DCD4  POP  DE
  DCD5  PUSH HL
  DCD6  OR   A
  DCD7  SBC  HL,DE
  DCD9  LD   C,L
  DCDA  LD   B,H
```

```
DCDB  POP  DE
DCDC  POP  HL
DCDD  PUSH HL
DCDE  ADD  HL,DE
DCDF  EX   DE,HL
DCE0  INC  BC
DCE1  LDDR              ; hely felszabadítás az adott sor szá-
                          mára
DCE3  INC  HL
DCE4  POP  BC
DCE5  RET

DCE6  LD   C,(HL)       ; C-be az adott sor hossza
DCE7  LD   B,00
DCE9  PUSH HL
DCEA  ADD  HL,BC
DCEB  PUSH HL
DCEC  CALL DD41         ; a program végének meghatározása
DCEF  POP  DE
DCF0  OR   A
DCF1  SBC  HL,DE
DCF3  LD   C,L
DCF4  LD   B,H          ; BC-be HL
DCF5  EX   DE,HL
DCF6  POP  DE
DCF7  PUSH DE
DCF8  INC  BC
DCF9  LDIR              ; az adott sorszámú régi sor törlése,
                          kizárása
DCFB  POP  HL
DCFC  PUSH HL
DCFD  PUSH DE
DCFE  PUSH BC
DCFF  LD   IY,(0B19)    ; IY-ba HIMEM
DD03  LD   HL,0000
DD06  LD   (IY+00H),L
DD09  LD   (170E),HL
DD0C  LD   (1714),HL
DD0F  LD   HL,F094
DD12  LD   (1724),HL    ; CHAIN írása
DD15  LD   HL,1709
DD18  LD   (HL),L
DD19  LD   (170A),HL
DD1C  LD   HL,(1722)    ; HL-be TEXT
DD1F  LD   (1712),HL
```

```
DD22  CALL DD41        ; a program végének meghatározása
DD25  INC  HL
DD26  LD   (1726),HL   ; TOP írása
DD29  POP  BC
DD2A  POP  DE
DD2B  POP  HL
DD2C  RET
```

A BASIC program egy sorának kiíratása:

```
DD2D  PUSH AF
DD2E  INC  HL
DD2F  LD   E,(HL)
DD30  INC  HL
DD31  LD   D,(HL)      ; DE-be a sor sorszáma
DD32  INC  HL
DD33  PUSH HL
DD34  EX   DE,HL
DD35  CALL FF19        ; a sorszám kiíratása
DD38  POP  HL
DD39  CALL FEDD        ; a programsor kiíratása
DD3C  POP  AF
DD3D  RET  NC
DD3E  JP   FE8E        ; a CEL "soremelés-kocsivissza" karakte-
                         rek kiíratása
```

A program végének meghatározása:

```
DD41  LD   HL,FFFF
DD44  EX   DE,HL
```

A program adott sorszámú sora címének meghatározása:

```
DD45  LD   HL,(1722)   ; HL-be TEXT
DD48  LD   BC,0000
DD4B  ADD  HL,BC       ; HL-be a sor címe
DD4C  LD   C,(HL)      ; C-be a sor hossza
DD4D  INC  C
DD4E  DEC  C
DD4F  SCF
DD50  RET  Z           ; a program vége, RET
DD51  PUSH DE
DD52  INC  HL
DD53  LD   E,(HL)
DD54  INC  HL
DD55  LD   D,(HL)      ; DE-be az adott sor sorszáma
DD56  DEC  HL
DD57  DEC  HL
DD58  EX   SP,HL       ; HL-be a keresett sorszám
```

```
DD59  OR   A
DD5A  SBC  HL,DE
DD5C  ADD  HL,DE
DD5D  EX   DE,HL
DD5E  POP  HL
DD5F  RET  Z          ; HL-ben a keresett sorszámú sor címe
DD60  JR   NC,E9      ; a sor sorszáma kisebb a keresettnél,
                        ugrás a DC4BH címre
DD62  RET             ; HL a keresett sorszámnál nagyobb sor-
                        számú sorok legkisebb számújára mutat

A "Cannot CONTINUE" hiba:
DD63  RST  08H
DD64  DB   0A

A CONTINUE parancs:
DD65  BIT  2,(IX+00H)
DD69  JP   NZ,DBB1    ; ugrás, ha fut a program
DD6C  SET  2,(IX+00H)
DD70  LD   HL,(170E)
DD73  LD   A,H
DD74  OR   L
DD75  JR   Z,EC       ; ugrás a hibajelzésre, ha HL=0
DD77  LD   (170C),HL
DD7A  LD   HL,(1710)
DD7D  JP   DB81       ; a program folytatódik a következő uta-
                        sítással

Az LLIST utasítás:
DD80  LD   C,40
DD82  OR   C
DD83  JR   03         ; ugrás a DD88H címre

A LIST utasítás:
DD85  LD   C,20
DD87  XOR  A
DD88  PUSH AF
DD89  CALL FBEE       ; van kijelölt periféria?
DD8C  POP  AF
DD8D  JR   NZ,05      ; ugrás a DD94H címre, ha LLIST
DD8F  LD   A,C
DD90  CP   20
DD92  LD   D,F6
DD94  SCF
```

A DELETE rutin:

```
***
  DD93  OR   37
***
  DD95  EX   AF,AF'
  DD96  EXX
  DD97  LD   A,B
  DD98  EXX
  DD99  LD   HL,0000
  DD9C  LD   DE,FFFF
  DD9F  CP   FD
  DDA1  JR   NC,2E        ; az utasítás a teljes programra vonat-
                            kozik (a szegmensek listája hiányzik),
                            ugrás a DDC1H címre
  DDA3  JR   11           ; ugrás a DDB6H címre
  DDA5  EXX
  DDA6  LD   A,B
  DDA7  EXX
  DDA8  CP   A4           ; (,)
  DDAA  JR   Z,07         ; több szegmens van megadva, ugrás a
                            DDB3H címre
  DDAC  EX   AF,AF'
  DDAD  JP   C,DBB1       ; a parancs programból lett kiadva, a
                            program folytatódik
  DDB0  JP   DADA         ; a parancs parancsként lett kiadva

  DDB3  CALL FC43         ; a következő utasítás byte beolvasása
  DDB6  CALL DDFB         ; a megadott sorszám átalakítása
  DDB9  INC  D
  DDBA  JR   NZ,08        ; ugrás a DDC4H címre
  DDBC  CP   A2           ; (-)
  DDBE  JP   NZ,FD5A      ; nem "-" karakter következett, szintak-
                            tikai hiba
  DDC1  LD   DE,0100
  DDC4  DEC  D
  DDC5  PUSH DE
  DDC6  CP   A2           ; (-)
  DDC8  JR   NZ,06        ; nem "-" karakter következett, ugrás a
                            DDD0H címre
  DDCA  CALL FC43         ; a következő utasítás byte
  DDCD  CALL DDFB         ; a szorszám konvertálása
  DDD0  POP  HL
  DDD1  PUSH DE
  DDD2  CALL DD44         ; programsor helye a memóriában
  DDD5  POP  DE
```

```
  DDD6  LD    A,(HL)
  DDD7  OR    A
  DDD8  JR    Z,CB          ; az adott sorszám nagyobb a legnagyobb
                              sorszámnál, ugrás a DDA5H címre
  DDDA  INC   HL
  DDDB  LD    C,(HL)
  DDDC  INC   HL
  DDDD  LD    B,(HL)        ; BC-be a HL című sor sorszáma
  DDDE  DEC   HL
  DDDF  DEC   HL
  DDE0  EX    DE,HL
  DDE1  SBC   HL,BC
  DDE3  ADD   HL,BC
  DDE4  EX    DE,HL
  DDE5  JR    C,BE          ; ugrás a DDA5H címre, ha elérte a meg-
                              adott szegmens végét
  DDE7  PUSH  DE
  DDE8  EX    AF,AF'
  DDE9  JR    NC,05         ; DELETE rutin, ugrás a DDF0H címre
  DDEB  CALL  DD2D          ; programsor kiíratása
  DDEE  JR    04            ; ugrás a DDF4H címre
  DDF0  CALL  DCE6          ; egy programsor kiejtése, törlése
  DDF3  OR    A
  DDF4  EX    AF,AF'
  DDF5  CALL  FF9D          ; CTRL+ESC vizsgálat
  DDF8  POP   DE
  DDF9  JR    DB            ; ugrás a DDD6H címre

Sorszám konvertálása:
  DDFB  LD    DE,FFFF
  DDFE  CP    02
  DE00  RET   NZ
  DE01  CALL  FAC3          ; a sorszám konvertálása
  DE04  EX    DE,HL
  DE05  JP    FC43          ; a következő program byte

A NEW rutin:
  DE08  LD    HL,DADA       ; HL-be a parancs mód címe
  DE0B  PUSH  HL
  DE0C  RES   0,(IX+00H)    ; nyomkövetés kikapcsolva
  DE10  LD    HL,(1720)     ; HL-be VLOMEM
  DE13  LD    (1722),HL     ; TEXT írása
  DE16  LD    (HL),00       ; a program "törlése"
  DE18  JP    DCFC          ; BASIC változók írása
```

A RUN parancs:

```
DE1B  LD   HL,(1722)   ; HL-be TEXT
DE1E  CP   02
DE20  CALL Z,FBDE      ; sorszámos RUN, ugrás
DE23  CALL DCFC        ; a következö utasítás byte
DE26  CALL FC3E        ; a kazetta file-ok zárása
DE29  SET  2,(IX+00H)  ; a program mód beállítása
DE2D  XOR  A
DE2E  JP   DBC2        ; a program indítása
```

A TRACE rutin:

```
DE31  CALL FBEC        ; a kimeneti periféria beállítása
DE34  LD   (IX+06H),C
DE37  CP   E3          ; (ON)
DE39  JR   Z,0B        ; a nyomkövetés bekapcsolása, ugrás a
                         DE46H címre
DE3B  CP   C1          ; (OFF)
DE3D  JP   NZ,FD5A     ; szintaktikai hiba, ugrás
DE40  RES  0,(IX+00H)  ; nyomkövetés kikapcsolva
DE44  JR   04          ; ugrás a DE4AH címre
DE46  SET  0,(IX+00H)  ; a nyomkövetés bekapcsolva
DE4A  JP   DBAE        ; a program folytatása
```

A végrehajtott utasítássor sorszámának kiíratása nyomkövetéskor:

```
DE4D  BIT  2,(IX+00H)
DE51  RET  Z           ; parancs mód, nincs nyomkövetés
DE52  LD   A,(1706)
DE55  LD   (1705),A
DE58  PUSH HL
DE59  LD   E,(HL)
DE5A  INC  HL
DE5B  LD   D,(HL)      ; DE-be a sor sorszáma
DE5C  EX   DE,HL
DE5D  LD   A,3C
DE5F  CALL FE9A        ; "<" karakter kiíratása
DE62  LD   B,00
DE64  CALL FF1B        ; a sorszám kiíratása
DE67  POP  HL
DE68  LD   A,3E
DE6A  JP   FE9A        ; ">" karakter kiíratása és a program
                         folytatása
```

A BASIC kulcsszavak táblázata (a név utolsó byte-jának legfelsö bitje 1-re van állítva):

```
                                                        token
DE6D  DB   FF                          ; sor vége jel   FF
DE6E  DB   A1                          ; !              FE
DE6F  DB   BA                          ; :              FD
DE70  DB   52 45 CD                    ; REM            FC
DE73  DB   44 41 54 C1                 ; DATA           FB
DE77  DB   43 4C 4F 53 C5              ; CLOSE          FA
DE7C  DB   43 4C D3                    ; CLS            F9
DE7F  DB   43 4F 4E 54 49 4E 55 C5     ; CONTINUE       F8
DE87  DB   44 45 C6                    ; DEF            F7
DE8A  DB   44 45 4C 45 54 C5           ; DELETE         F6
DE90  DB   44 49 CD                    ; DIM            F5
DE93  DB   45 4C 53 C5                 ; ELSE           F4
DE97  DB   45 4E C4                    ; END            F3
DE9A  DB   46 4F D2                    ; FOR            F2
DE9D  DB   47 45 D4                    ; GET            F1
DEA0  DB   47 4F 53 55 C2              ; GOSUB          F0
DEA5  DB   47 4F 54 CF                 ; GOTO           EF
DEA9  DB   47 52 41 50 48 49 43 D3     ; GRAPHICS       EE
DEB1  DB   49 C6                       ; IN             ED
DEB3  DB   49 4E 50 55 D4              ; INPUT          EC
DEB8  DB   4C 45 D4                    ; LET            EB
DEBB  DB   4C 49 53 D4                 ; LIST           EA
DEBF  DB   4C 4C 49 53 D4              ; LLIST          E9
DEC4  DB   4C 4F 41 C4                 ; LOAD           E8
DEC8  DB   4C 4F 4D 45 CD              ; LOMEM          E7
DECD  DB   4E 45 D7                    ; NEW            E6
DED0  DB   4E 45 58 D4                 ; NEXT           E5
DED4  DB   4F CB                       ; OK             E4
DED6  DB   4F CE                       ; ON             E3
DED8  DB   4F 50 45 CE                 ; OPEN           E2
DEDC  DB   4F 55 54 50 55 D4           ; OUTPUT         E1
DEE2  DB   4F 55 D4                    ; OUT            E0
DEE5  DB   50 4C 4F D4                 ; PLOT           DF
DEE9  DB   50 4F 4B C5                 ; POKE           DE
DEED  DB   50 52 49 4E D4              ; PRINT          DD
DEF2  DB   52 41 4E 44 4F 4D 49 5A C5  ; RANDOMIZE      DC
DEFB  DB   52 45 41 C4                 ; READ           DB
DEFF  DB   52 45 53 54 4F 52 C5        ; RESTORE        DA
DF06  DB   52 45 54 55 52 CE           ; RETURN         D9
DF0C  DB   52 55 CE                    ; RUN            D8
DF0F  DB   53 41 56 C5                 ; SAVE           D7
DF13  DB   53 45 D4                    ; SET            D6
```

```
DF16  DB   53 4F 55 4E C4                  ; SOUND          D5
DF1B  DB   53 54 4F D0                     ; STOP           D4
DF1F  DB   54 52 41 43 C5                  ; TRACE          D3
DF24  DB   56 45 52 49 46 D9               ; VERIFY         D2
DF2A  DB   45 58 D4                        ; EXT            D1
DF2D  DB   4C 50 52 49 4E D4               ; LPRINT         D0
DF33  DB   A1 A1 A1 A1 A1 A1               ;                CF--CA
DF39  DB   41 4E C4                        ; AND            C9
DF3C  DB   43 48 41 52 41 43 54 45 D2      ; CHARACTER      C8
DF45  DB   44 45 4C 41 D9                  ; DELAY          C7
DF4A  DB   44 55 52 41 54 49 4F CE         ; DURATION       C6
DF52  DB   49 4E 4B 45 59 A4               ; INKEY$         C5
DF58  DB   49 4E CB                        ; INK            C4
DF5B  DB   4D 4F 44 C5                     ; MODE           C3
DF5F  DB   4E 4F D4                        ; NOT            C2
DF62  DB   4F 46 C6                        ; OFF            C1
DF65  DB   4F 52 C4                        ; ORD            C0
DF68  DB   4F D2                           ; OR             BF
DF6A  DB   50 41 49 4E D4                  ; PAINT          BE
DF6F  DB   50 41 4C 45 54 54 C5            ; PALETTE        BD
DF76  DB   50 41 50 45 D2                  ; PAPER          BC
DF7B  DB   50 49 54 43 C8                  ; PITCH          BB
DF80  DB   50 52 4F 4D 50 D4               ; PROMPT         BA
DF86  DB   52 41 54 C5                     ; RATE           B9
DF8A  DB   53 54 45 D0                     ; STEP           B8
DF8E  DB   53 54 59 4C C5                  ; STYLE          B7
DF93  DB   54 41 C2                        ; TAB            B6
DF96  DB   54 48 45 CE                     ; THEN           B5
DF9A  DB   54 CF                           ; TO             B4
DF9C  DB   56 4F 4C 55 4D C5               ; VOLUME         B3
DFA2  DB   58 4F D2                        ; XOR            B2
DFA5  DB   41 54 CE                        ; ATN            B1
DFA8  DB   41 D4                           ; AT             B0
DFAA  DB   55 53 49 4E C7                  ; USING          AF
DFAF  DB   42 4F 52 44 45 D2               ; BORDER         AE
DFB5  DB   A1 A1 A1 A1 A1                  ;                AD--A9
DFBA  DB   AA                              ; *              A8
DFBB  DB   A3                              ; #              A7
DFBC  DB   3D BE                           ; =>             A6
DFBE  DB   3E BC                           ; ><             A5
DFC0  DB   AC                              ; ,              A4
DFC1  DB   3D BC                           ; =<             A3
DFC3  DB   AD                              ; -              A2
DFC4  DB   AF                              ; /              A1
DFC5  DB   BB                              ; ;              A0
```

```
DFC6  DB   DE                              ; ^          9F
DFC7  DB   3E DB                           ; >=         9E
DFC9  DB   3C BE                           ; <>         9D
DFCB  DB   BE                              ; >          9C
DFCC  DB   3C BD                           ; <=         9B
DFCE  DB   BD                              ; =          9A
DFCF  DB   BC                              ; <          99
DFD0  DB   AB                              ; +          98
DFD1  DB   A6                              ; &          97
DFD2  DB   A8                              ; (          96
DFD3  DB   A9                              ; )          95
DFD4  DB   FF                              ;            94
DFD5  DB   20 6D 69 73 73 69 6E E7         ; missing    93
DFDD  DB   72 67 75 6D 65 6E 74            ; rgument    92
DFE4  DB   42 61 64 A0                     ; Bad        91
DFE8  DB   4E 6F A0                        ; No         90
DFEB  DB   43 61 6E 6E 6F 74 A0            ; Cannot     8F
```

A DATA rutin:

```
DFF2  EXX
DFF3  LD   A,(HL)      ; A-ba az utasítás sor aktuális byte-ja
DFF4  CP   FD
DFF6  JP   NC,DB81     ; vége az adatoknak, a program folyta-
                         tódik
DFF9  INC  HL          ; HL a következő program byte-ra mutat
DFFA  JR   F7          ; urgás a DFF3H címre
```

A CLS utasítás rutinja:

```
DFFC  RST  30H
DFFD  DB   05          ; a video eszköz CLS rutinja
DFFE  RST  10H         ; volt hiba?
DFFF  JP   DBB1        ; a program folytatása
```

A DEF rutin:

```
E002  AND  81
E004  CP   01
E006  JP   NZ.FD5A     ; szintaktikai hiba
E009  BIT  2,(IX+00H)
E00D  JR   Z,1F        ; parancs mód, ugrás a E02EH címre
E00F  EXX
E010  PUSH HL
E011  PUSH DE
E012  EXX
E013  POP  HL
```

```
  E014  DEC  HL
  E015  BIT  7,(HL)
  E017  JR   Z,35          ; már létező változó név, ugrás a hiba-
                             jelzésre, az E04EH címre
  E019  RES  7,(HL)
  E01B  SET  2,(HL)        ; a változó tipus byte-jának beállítása
  E01D  INC  HL
  E01E  LD   DE,(170C)     ; DE-be START
  E022  LD   (HL),E
  E023  INC  HL
  E024  LD   (HL),D        ; a DEF utasítást tartalmazó sor címé-
                             nek eltárolása a változóban
  E025  INC  HL
  E026  POP  DE
  E027  LD   (HL),E
  E028  INC  HL
  E029  LD   (HL),D        ; az aktuális programsorban a függvény
                             neve után mutató cím eltárolása a vál-
                             tozóban
  E02A  INC  HL
  E02B  LD   (1726),HL     ; TOP új értékének elmentése
  E02E  EXX
  E02F  LD   C,01
  E031  DEC  C
  E032  LD   A,0C
***
  E033  INC  C
***
  E034  LD   A,(HL)
  E035  CP   FE
  E037  JP   NC,E04B       ; ugrás, ha "!", vagy sor vége jel kö-
                             vetkezik
  E03A  INC  HL
  E03B  CP   96
  E03D  JR   Z,F4          ; ugrás az E033H címre, ha "(" jel
  E03F  CP   95
  E041  JR   Z,EE          ; ugrás az E031H címre, ha ")" jel
  E043  CP   FD
  E045  JR   NZ,ED         ; ugrás az E034H címre, ha nem ":" jel
  E047  INC  C
  E048  DEC  C
  E049  JR   NZ,E9         ; ugrás az E034H címre, ha még nem befe-
                             jezett a DEF utasítás
  E04B  JP   DB81          ; ugrás a következő utasítás értelmezé-
                             sére
```

```
A "Variable declared twice" hibajelzés
  E04E  RST  08H
  E04F  DB   0F

  E050  CALL FC43

A DIM utasítás:
  E053  OR   A
  E054  JP   M,FD5A      ; szintaktikai hiba, ugrás
  E057  BIT  0,A
  E059  JP   Z,FD5A      ; szintaktikai hiba, ugrás
  E05C  EXX
  E05D  LD   A,C
  E05E  AND  88
  E060  JR   Z,EC        ; már létező változó, ugrás az E04EH
                           hibajelzésre
  E062  AND  08
  E064  CALL NZ,F40B     ; a változó újra definiálása
  E067  DEC  DE
  E068  LD   A,(DE)      ; A-ba a változó tipusbyte-ja
  E069  XOR  81          ; a tipusbyte beállítása
  E06B  LD   (1701),A
  E06E  LD   (DE),A      ; a tipusbyte visszatöltése
  E06F  INC  DE          ; DE az első adatbyte-ra mutat
  E070  LD   A,FF
  E072  LD   (DE),A      ; dimenzió byte írása
  E073  PUSH DE
  E074  INC  DE
  E075  LD   (1726),DE   ; TOP új értéke
  E079  EXX
  E07A  CALL FC43        ; a következő utasítás byte
  E07D  BIT  1,(IX+01H)
  E081  JR   NZ,20       ; ugrás az E0A3H címre, ha numerikus
                           tömb
  E083  CP   A8          ; (*)
  E085  JR   NZ,1C       ; ugrás az E0A3H címre, ha a string-
                           tömb elemének mérete nem megadott
  E087  CALL FB16        ; a megadott string hossz ellenörzése
  E08A  INC  A
  E08B  JR   Z,57        ; ugrás az E0E4H címre, ha
  E08D  DEC  A
  E08E  LD   C,A
  E08F  POP  HL
  E090  LD   (1726),HL   ; TOP új értéke
  E093  DEC  HL
```

```
E094  RES  0,(HL)
E096  CALL F3C1        ; a string maximális hosszának beállí-
                         tása
E099  EXX
E09A  LD   A,B
E09B  EXX
E09C  CP   A4          ; (,)
E09E  JR   Z,B0        ; ugrás az E050H címre, ha még van
                         újabb tömb
E0A0  JP   DBB4        ; a DIM utasítás vége

E0A3  LD   A,96        ; A-ba "(" jel
E0A5  CALL FD54        ; ha a következő karakter nem "(", szin-
                         taktikai hiba
E0A8  SCF
E0A9  CALL NC,FC43     ; a következő utasítás byte
E0AC  CALL FAC4        ; a megadott dimenzió ellenörzése
E0AF  JP   M,FB14      ; rossz argumentum, hiba jelzés
E0B2  INC  HL
E0B3  EX   DE,HL
E0B4  LD   HL,(1726)   ; HL-be TOP
E0B7  LD   (HL),E
E0B8  INC  HL
E0B9  LD   (HL),D      ; a dimenzió méret eltárolása
E0BA  INC  HL
E0BB  LD   (1726),HL   ; TOP új értéke
E0BE  EX   DE,HL
E0BF  CALL FA2B
E0C2  POP  HL
E0C3  INC  (HL)
E0C4  PUSH HL
E0C5  CALL NZ,F512     ; a 2. aritmetikai rutin
E0C8  EXX
E0C9  LD   A,B
E0CA  EXX
E0CB  CP   A4          ; (,)
E0CD  JR   Z,DA        ; több dimenziós a tömb, ugrás az E0A9H
                         címre
E0CF  LD   A,95        ; A-ba a ")" jel
E0D1  CALL FD54        ; ha a programsorban nem ")" jel követ-
                         kezik, szintaktikai hiba
E0D4  POP  HL
E0D5  INC  (HL)
E0D6  BIT  1,(IX+01H)
```

```
  E0DA  JR   NZ,0D       ; ugrás az E0E9H címre, ha numerikus
                           tömb
  E0DC  CP   A8          ; (*)
  E0DE  LD   A,12        ; az alap string hossz (18) A-ba
  E0E0  CALL Z,FB16      ; a string-tömb elem hossza definiált,
                           ugrás
  E0E3  INC  A
  E0E4  JP   Z,FB14      ; rossz argument, ugrás
  E0E7  DEC  A
  E0E8  LD   C,A
  E0E9  PUSH BC
  E0EA  RST  18H
  E0EB  DB   0C 85 26    ; aritmetikai rutinok hívása
  E0EE  CALL F693        ; van elegendö hely a tömbnek?
  E0F1  JP   P,FCB1      ; nincs, hibajelzés
  E0F4  CALL FAC3
  E0F7  POP  BC
  E0F8  PUSH HL
  E0F9  CALL F3BB        ; kezdö érték a string elemeinek
  E0FC  POP  HL
  E0FD  DEC  HL
  E0FE  LD   A,L
  E0FF  OR   H
  E100  JR   NZ,F6       ; HL nem 0, ugrás az E0F8H címre
  E102  JR   95          ; ugrás az E099H címre

Az ELSE rutin:
  E104  BIT  0,(IX+02H)
  E108  JP   Z,FD5A      ; szintaktikai hiba, ugrás
  E10B  JP   DBBB        ; a következö utasításra ugrás

Az END rutin:
  E10E  LD   HL,0000
  E111  LD   (170E),HL   ; a program futás vége
  E114  JP   DADA        ; ugrás az interpreter indulására

Az EXT rutin:
  E117  CALL FB1B        ; az argumentum ellenörzése
  E11A  CP   07
  E11C  JP   NC,FB14     ; rossz argumentum
  E11F  ADD  A,A         ; a szubrutin sorszám kétszerese
  E120  PUSH AF
  E121  EXX
  E122  LD   A,B
  E123  EXX
```

```
  E124  CP    A4          ; (,)
  E126  JR    NZ,16       ; nincs több paraméter, ugrás az E13EH
                            címre
  E128  CALL  FABA        ; a paraméter átalakítása 2 byte-ossá
  E12B  EX    DE,HL
  E12C  CP    A4          ; (,)
  E12E  JR    NZ,0E       ; nincs több paraméter, ugrás az E13EH
                            címre
  E130  CALL  FABA        ; a paraméter átalakítása 2 byte-ossá
  E133  CP    A4
  E135  JR    NZ,07       ; nincs több paraméter, ugrás az E13EH
                            címre
  E137  PUSH  HL
  E138  CALL  FABA        ; a paraméter átalakítása 2 byte-ossá
  E13B  LD    C,L
  E13C  LD    B,H         ; 3. paraméter a BC-be
  E13D  POP   HL
  E13E  EX    SP,HL
  E13F  PUSH  DE
A rutin címének meghatározása:
  E140  LD    E,H
  E141  LD    D,00        ; DE-de a rutin sorszámának kétszerese
  E143  LD    HL,0021     ; HL-be USRTAB címe
  E146  ADD   HL,DE       ; HL-be a rutin címének a címe
  E147  LD    E,(HL)
  E148  INC   HL
  E149  LD    D,(HL)      ; DE-be a rutin címe
  E14A  LD    A,D
  E14B  OR    E
  E14C  JP    Z,DBB1      ; ha DE=00H, ugrás a program következö
                            utasítására
  E14F  LD    (0016),DE   ; a rutin cím elhelyezése
  E153  POP   DE          ; DE-be az elsö paraméter
  E154  LD    HL,DBB1     ; visszatérési cím HL-be
  E157  EX    SP,HL       ; HL-be a második paraméter
  E158  EX    DE,HL
  E159  JP    0015        ; ugrás a user rutinra, az EXT rutin
                            vége

A FOR rutin:
  E15C  CP    03
  E15E  JR    Z,07        ; megfelelö változó FOR után, ugrás az
                            E167H címre
  E160  AND   82
  E162  JP    NZ,FD5A     ; ugrás, ha szintaktikai hiba
```

```
E165  RST  08H
E166  DB   0E          ; "Type mismatch" hiba jelzés
E167  CALL F42E
E16A  LD   A,9A        ; (=)
E16C  CALL FD54        ; ha nem "=" jel következik, szintakti-
                         kai hiba
E16F  CALL FC8E        ; van elég hely?
E172  PUSH HL
E173  CALL F0A7        ; a ciklusváltozó eltárolása
E176  POP  HL
E177  PUSH HL
E178  CALL FB28        ; 13. aritmetikai rutin: a BASIC STACK-
                         be a HL által címzett 9 byte
E17B  LD   A,B4        ; (TO)
E17D  CALL FD54        ; ha a következö byte nem a TO tokenje,
                         szintaktikai hiba
E180  CALL F0A7        ; TO értékének eltárolása
E183  LD   DE,FFEE
E186  ADD  IY,DE       ; IY csökkentése 18-cal
E188  CP   B8          ; (STEP)
E18A  JR   Z,09        ; ha van STEP, ugrás az E195H címre
E18C  EX   AF,AF'
E18D  LD   HL,0001     ; STEP értéke 1
E190  CALL FA2B        ; két byte-os számból 9 byte-os
E193  EX   AF,AF'
E194  SCF
E195  CALL NC,F0A4     ; STEP értékének eltárolása
E198  CP   FD
E19A  JP   C,FD5A      ; ha nincs vége a FOR utasításnak, szin-
                         taktikai hiba
E19D  PUSH IY
E19F  POP  HL
E1A0  POP  DE
E1A1  DEC  HL
E1A2  LD   (HL),D
E1A3  DEC  HL
E1A4  LD   (HL),E      ; a ciklusváltozó címének elrakása
E1A5  LD   DE,(170C)   ; DE-be START
E1A9  DEC  HL
E1AA  LD   (HL),D
E1AB  DEC  HL
E1AC  LD   (HL),E      ; a FOR-t tartalmazó sor címének eltá-
                         rolása
E1AD  EXX
E1AE  PUSH HL
```

```
E1AF  EXX
E1B0  POP  DE
E1B1  DEC  HL
E1B2  LD   (HL),D
E1B3  DEC  HL
E1B4  LD   (HL),E      ; a FOR-t követö utasítás címének eltá-
                         rolása
E1B5  DEC  HL
E1B6  LD   (HL),2B     ; a FOR-hoz tartozó bejegyzés hossza a
                         bejegyzés elsö byte-jába
E1B8  PUSH HL
E1B9  POP  IY
E1BB  JP   DBB1        ; a FOR rutin vége, ugrás a következö u-
                         tasításra


E1BE  CP   F5
E1C0  RST  10H         ; CTRL+ESC ellenörzés
E1C1  EXX
E1C2  DEC  HL
E1C3  LD   A,(HL)
E1C4  CP   EC          ; (INPUT)
E1C6  JR   NZ,FA       ; ugrás az E1C2H címre, amíg HL nem az
                         INPUT utasításra mutat
E1C8  JP   FFA4        ; STOP rutin
```

Az INPUT utasítás rutinja:

```
E1CB  EX   AF,AF'
E1CC  SCF
E1CD  EX   AF,AF'
E1CE  LD   HL,E2EB
E1D1  LD   (IX+05H),20; az EDITOR az alap kimeneti periféria
E1D5  SCF
E1D6  CALL NC,FC43     ; a következö utasítás byte
E1D9  CP   BA          ; (PROMPT)
E1DB  JR   NZ,1F       ; nincs PROMPT, ugrás az E1FCH címre
E1DD  CALL FC43        ; programszöveg vizsgálat
E1E0  CALL F294
E1E3  PUSH IY
E1E5  CALL FA21        ; a string helyének felszabadítása a
                         BASIC STACK-ben
E1E8  POP  HL
E1E9  INC  HL
E1EA  EXX
E1EB  LD   A,B
```

```
  E1EC  EXX
  E1ED  EX   AF,AF'
  E1EE  OR   A
  E1EF  EX   AF,AF'
  E1F0  CP   A4          ; (,)
  E1F2  JR   Z,E2        ; ugrás az E1D6H címre, ha "," következik
  E1F4  CP   FD          ; (:)
  E1F6  JR   Z,0C        ; ugrás az E204H címre, ha ":" következik
  E1F8  JR   NC,0D       ; ugrás az E207H címre, ha kiíratás
  E1FA  RST  08H
  E1FB  DB   01          ; szintaktikai hiba
  E1FC  CALL FBFB        ; más megadott print periféria?
  E1FF  JR   Z,EC        ; igen, ugrás az E1EDH címre
  E201  EX   AF,AF'
  E202  JR   NC,F6       ; szintaktikai hiba, ugrás az E1FAH-ra
  E204  CALL NC,FC43     ; következő program byte
  E207  LD   A,(1705)
  E20A  CP   20
  E20C  CALL Z,FE7F      ; kiíratás, ha A-ba 20H került
  E20F  RST  30H
  E210  DB   24          ; CFIX rutin
  E211  CALL FF4F        ; input
  E214  JR   NZ,A8       ; nem RET-tel ért véget az input, ugrás az E1BEH címre
  E216  INC  HL
  E217  LD   (1716),HL
  E21A  OR   37

A READ rutin:
***
  E21B  SCF
***
  E21C  EX   AF,AF'
  E21D  EXX
  E21E  LD   A,B         ; A-ba a program byte
  E21F  EXX
  E220  SCF
  E221  CALL NC,FC43     ; a következő program byte, ha nincs CARRY
  E224  CP   FD
  E226  JP   NC,DBB4     ; ugrás, ha az utasítás vége
  E229  AND  81
  E22B  CP   01
```

```
E22D  JP   NZ,FD5A     ; ugrás, ha szintaktikai hiba
E230  EX   AF,AF'
E231  JR   NC,49       ; ugrás az E27CH címre, ha INPUT
E233  LD   HL,(1714)
E236  LD   A,L
E237  OR   H
E238  JR   Z,2E        ; ugrás az E268H címre, ha HL=00H
E23A  LD   A,(HL)
E23B  INC  HL
E23C  CP   20
E23E  JR   Z,FA        ; ugrás az E23AH címre, ha SPACE
E240  DEC  HL
E241  CP   21
E243  JR   Z,19        ; ugrás az E25EH címre, ha "!"
E245  CP   FD
E247  JR   C,33        ; ugrás az E27CH címre, ha utasítás vége
E249  EXX
E24A  PUSH HL
E24B  PUSH DE
E24C  PUSH BC
E24D  CALL FCC5        ; a rákövetkező utasítás elejének meg-
                         keresése
E250  EXX
E251  POP  BC
E252  POP  DE
E253  POP  HL
E254  EXX
E255  INC  HL
E256  CP   FB          ; (DATA)
E258  JR   Z,E0        ; HL DATA utasításra mutat, ugrás az
                         E23AH címre
E25A  CP   FE
E25C  JR   C,EB        ; nem "!", vagy sor vége jel, ugrás az
                         E249H címre
E25E  LD   HL,(1712)   ; HL-be a DATA sor címe
E261  LD   E,(HL)
E262  LD   D,00        ; DE-be a sor hossza
E264  ADD  HL,DE       ; HL-be a következő programsor címe
E265  LD   (1712),HL
E268  LD   HL,(1712)   ; HL-be a programsor hossza
E26B  LD   A,(HL)
E26C  OR   A
E26D  JR   Z,0B        ; a program vége, ugrás az E27AH címre:
                         No DATA
E26F  INC  HL
```

```
E270  INC  HL
E271  INC  HL          ; HL-be a sor elsö utasítás byte-ja
E272  LD   A,(HL)      ; A-ba az utasítás byte
E273  INC  HL
E274  CP   20
E276  JR   Z,FA        ; ha SPACE, ugrás az E272H címre
E278  JR   DC          ; ugrás az E256H címre

E27A  RST  08H
E27B  DB   07          ; "No DATA" hibajelzés

E27C  EX   AF,AF'
E27D  PUSH HL
E27E  EXX
E27F  PUSH BC
E280  CALL F42F        ; tömb elem input
E283  POP  BC
E284  BIT  1,C
E286  EX   SP,HL
E287  JR   NZ,0B       ; ha numerikus változó input, ugrás az
                         E294H címre
E289  CALL F8A1        ; string input
E28C  EX   SP,HL
E28D  CALL FB41        ; string átmozgatása a BASIC STACK-böl
                         a változó területre
E290  JR   1E          ; ugrás az E2B0H címre

E292  RST  08H
E293  DB   0C          ; "Cannot READ" hibajelzés

E294  CALL FC8E        ; van elég hely?
E297  CALL F914        ; numerikus változó
E29A  EX   AF,AF'
E29B  JR   NC,09       ; INPUT, ugrás az E2A6H címre
E29D  EX   AF,AF'
E29E  JR   NC,F2       ; string változó READ következik, ug-
                         rás az E292H címre
E2A0  RLA
E2A1  JP   C,F912      ; "Cannot divide by 0" hiba
E2A4  JR   06          ; ugrás az E2ACH címre
E2A6  EX   AF,AF'
E2A7  PUSH HL
E2A8  CALL EC7F
E2AB  POP  HL
E2AC  EX   SP,HL
```

```
E2AD  CALL FB28        ; a 13. aritmetikai rutin
E2B0  POP  HL
E2B1  EX   AF,AF'
E2B2  JR   NC,1A       ; INPUT, ugrás az E2CEH címre
E2B4  EX   AF,AF'
E2B5  LD   A,(HL)
E2B6  INC  HL
E2B7  CP   20
E2B9  JR   Z,FA        ; a szóközök kiejtése
E2BB  CP   2C
E2BD  JR   Z,0A        ; van még beolvasható változó, ugrás az
                         E2C9H címre
E2BF  DEC  HL
E2C0  CP   21
E2C2  JR   Z,05        ; ugrás az E2C9H címre, ha "!"
E2C4  BIT  7,(HL)
E2C6  JP   Z,E292      ; ugrás a Cannot READ hibára, ha
E2C9  LD   (1714),HL   ; HL eltárolása a következö READ-hez
E2CC  JR   0E          ; ugrás az E2DCH címre
E2CE  EX   AF,AF'
E2CF  LD   A,(HL)
E2D0  INC  HL
E2D1  CP   2C
E2D3  JR   Z,04        ; van következö beolvasható változó, ug-
                         rás az E2D9H címre
E2D5  INC  A
E2D6  JR   NZ,F7       ; ugrás az E2CFH-ra, ha A=FFH volt
E2D8  DEC  HL
E2D9  LD   (1716),HL   ; HL eltárolása a következö változó be-
                         olvasásához
E2DC  EXX
E2DD  LD   A,B
E2DE  EXX
E2DF  CP   FD
E2E1  JP   NC,DBB4     ; vége az inputnak, ugrás
E2E4  CP   A4
E2E6  JP   Z,E221      ; még van beolvasandó változó, ugrás

E2E9  RST  08H
E2EA  DB   01          ; szintaktikai hiba

E2EB  DB   02
E2EC  DB   '? '        ; az INPUT rutin használja
```

```
Az IF rutin:
  E2EE  SET  0,(IX+02)
  E2E2  CALL F0A7          ; a logikai kifejezés kiértékelése
  E2F5  LD   A,(IY+06H)    ; A-ba a logikai kifejezés értéke
  E2F8  OR   A
  E2F9  EX   AF,AF'
  E2FA  CALL FA1B          ; 9 byte felszabadítása a BASIC STACK-
                             ből
  E2FD  EXX
  E2FE  PUSH HL
  E2FF  EXX
  E300  POP  HL
  E301  LD   A,B5          ; (THEN)
  E303  CALL FD54          ; szintaktikai hiba, ha nem THEN követ-
                             kezik
  E306  EX   AF,AF'
  E307  JR   Z,13          ; ugrás az E31CH címre, ha a logikai ki-
                             fejezés hamis
  E309  EX   AF,AF'
  E30A  JR   08            ; ugrás az E314H címre
  E30C  INC  HL
  E30D  PUSH HL
  E30E  EXX
  E30F  POP  HL
  E310  EXX
  E311  CALL FC43          ; következő program byte
  E314  CP   02
  E316  JP   Z,E3B2        ; szám következik, ugrás a GOTO rutinra
  E319  JP   DB81          ; a következő utasítással folytatódik
ELSE ág:
  E31C  LD   IY,(171A)
  E320  CALL FCC5          ; a következő utasítás token megkeresése
  E323  CP   F4            ; (ELSE)
  E325  JR   Z,E5          ; ugrás az E30CH-ra, ha ELSE
  E327  CP   FE
  E329  JP   NC,DBBB       ; ugrás a következő sorra, ha sor vége
  E32C  PUSH HL
  E32D  EXX
  E32E  POP  HL
  E32F  EXX
  E330  JR   EE            ; ugrás az E320H címre

Az ON rutin:
  E332  SET  0,(IX+02H)    ; az IF_FLAG beállítása
  E336  CALL FB1B          ; az ON utáni kifejezés kiértékelése
```

```
E339  LD   B,A
E33A  EXX
E33B  LD   A,B
E33C  EXX
E33D  CP   EF          ; (GOTO)
E33F  JR   Z,6E        ; ugrás az E3AFH címre, ha GOTO
E341  CP   F0          ; (GOSUB)
E343  JR   NZ,24       ; ugrás az E36AH címre, ha nem GOSUB
E345  CALL E357        ; az ON utasítás kiértékelése
E348  CALL FBDE        ; adott sorszámú programsor megkeresése
E34B  LD   IY,(171A)
E34F  PUSH HL
E350  CALL FCD1        ; adott sorban az első utasítás megke-
                         resése
E353  EX   DE,HL
E354  POP  HL
E355  JR   37          ; ugrás az E38EH címre
Az ON utasítás kiértékelése:
E357  CALL FC43        ; a következő program byte
E35A  DEC  B
E35B  RET  Z           ; meg van az ugrási cím
E35C  EXX
E35D  PUSH HL
E35E  EXX
E35F  POP  HL
E360  CALL FC43        ; a következő program byte
E363  JR   NC,06       ; ugrás az E36BH címre, ha az ON utáni
                         kifejezés 0, vagy nagyobb a megadott
                         címek számánál
E365  CP   A4          ; (,)
E367  JR   Z,EE        ; ugrás az E357H címre, ha még van mega-
                         dott ugrási cím

E369  RST  08H
E36A  DB   01          ; szintaktikai hiba

E36B  POP  DE
E36C  LD   IY,(171A)
E370  EXX
E371  CALL FCC5        ; az adott sorban a következő utasítás
                         token megkeresése
E374  CP   F4          ; (ELSE)
E376  JR   Z,94        ; ugrás az E30CH címre ha ELSE
E378  CP   FE
E37A  JP   NC,FB14     ; "Bad argument" hiba, ha nem sor vége
```

```
  E37D  PUSH HL
  E37E  EXX
  E37F  POP  HL
  E380  JR   EE          ; ugrás az E370H címre

A GOSUB rutin:
  E382  CALL FBDE        ; az adott sorszámú programsor megkere-
                           sése
  E385  CALL FC43        ; a következő program byte
  E388  JR   C,DF        ; szintaktikai hiba, ugrás az E369H-ra
  E38A  EXX
  E38B  PUSH HL
  E38C  EXX
  E38D  POP  DE
  E38E  PUSH HL
  E38F  CALL FC8E        ; van elég szabad hely?
Bejegyzés elkészítése a BASIC STACK-ben:
  E392  PUSH IY
  E394  POP  HL
  E395  DEC  HL
  E396  LD   A,(1702)    ; ON GOSUB-nál ELSE ág adott?
  E399  LD   (HL),A
  E39A  LD   BC,(170C)   ; BC-be START
  E39E  DEC  HL
  E39F  LD   (HL),B
  E3A0  DEC  HL
  E3A1  LD   (HL),C      ; a GOSUB-t tartalmazó sor címének eltá-
                           rolása
  E3A2  DEC  HL
  E3A3  LD   (HL),D
  E3A4  DEC  HL
  E3A5  LD   (HL),E      ; a GOSUB-t követő utasítás címének eltá-
                           rolása
  E3A6  DEC  HL
  E3A7  LD   (HL),06     ; GOSUB-hoz tartozó bejegyzés jele
  E3A9  PUSH HL
  E3AA  POP  IY
  E3AC  POP  HL
  E3AD  JR   0A          ; ugrás az E3B9H címre

  E3AF  CALL E357        ; az ON utasítás kiértékelése

A GOTO rutin:
  E3B2  CALL FBDE        ; adott sorszámú sor megkeresése
  E3B5  LD   IY,(171A)
```

```
  E3B9  JP   DE29        ; ugrás az adott sorra

LET nélküli értékadó rutin:
  E3BC  DEC  HL
  E3BD  EXX
  E3BE  CALL FC43        ; a következő program byte

A LET rutin:
  E3C1  AND  FD
  E3C3  CP   01
  E3C5  JP   NZ,FD5A     ; szintaktikai hiba, ugrás
  E3C8  EXX
  E3C9  BIT  2,C
  E3CB  JP   NZ,FD5A     ; szintaktikai hiba, ugrás
  E3CE  BIT  3,C
  E3D0  CALL NZ,F40B     ; az új változó bejegyzése
  E3D3  LD   (IX+01H),C
  E3D6  EXX
  E3D7  CALL F42E        ; tömb elem?
  E3DA  EXX
  E3DB  LD   A,B
  E3DC  EXX
  E3DD  PUSH HL
  E3DE  CALL E3E8        ; az értékadó kifejezés értékének megha-
                           tározása
  E3E1  POP  HL
  E3E2  CALL FB3B        ; a változó értékének átmozgatása a
                           BASIC STACK-ből a változó területre
  E3E5  JP   DBB1        ; az értékadás vége, ugrás a következő
                           utasításra

  E3E8  SUB  96
  E3EA  JR   NZ,5E       ; ugrás az E44AH címre, ha nem "(" kö-
                           vetkezik
  E3EC  BIT  1,C
  E3EE  JP   NZ,FD5A     ; szintaktikai hiba, ugrás
  E3F1  INC  HL
  E3F2  PUSH HL
  E3F3  LD   C,(HL)
  E3F4  LD   B,A
  E3F5  CALL FD27
  E3F8  LD   A,D
  E3F9  CP   E
  E3FA  JR   NC,02       ; ugrás az E3FDH címre, ha E<=D
  E3FC  LD   A,E
```

```
E3FD  DEC  A
E3FE  LD   D,A
E3FF  LD   A,C
E400  SUB  D
E401  POP  HL
E402  PUSH IY
E404  PUSH HL
E405  PUSH DE
E406  PUSH BC
E407  JR   C,0E        ; ugrás az E417H címre. ha D>C
E409  JR   Z,0C        ; ugrás az E417H címre. ha D=C
E40B  ADD  HL.BC
E40C  LD   C,A
E40D  PUSH IY
E40F  POP  DE
E410  DEC  DE
E411  LDDR             ; a BASIC STACK-be tölt
E413  INC  DE
E414  PUSH DE
E415  POP  IY
E417  CALL E44A        ; az értékadó kifejezés kiértékelése
E41A  POP  BC
E41B  POP  DE
E41C  POP  HL
E41D  LD   A,E
E41E  PUSH IY
E420  POP  DE          ; DE-be IY
E421  INC  DE
E422  CP   02
E424  JR   C,0C        ; ugrás az E432H címre. ha CARRY
E426  DEC  A
E427  CP   C
E428  JR   NC,01       ; ugrás az E42BH címre. ha C<=A
E42A  LD   C,A
E42B  ADD  HL.BC
E42C  INC  C
E42D  DEC  C
E42E  JR   Z.02        ; ugrás az E432H címre. ha C=00H
E430  LDDR             ; adat mozgatás a BASIC STACK-ben
E432  POP  HL
E433  DEC  HL
E434  OR   A
E435  SBC  HL,DE
E437  INC  H
E438  DEC  H
```

```
E439  JP   NZ,F912     ; "Cannot divide by 0" hiba, ha H-1 volt
E43C  INC  L
E43D  JP   Z,F912      ; "Cannot divide by 0" hiba, ha L=FFH volt
E440  DEC  L
E441  EX   DE,HL
E442  LD   (HL),E
E443  DEC  HL
E444  LD   (HL),01     ; sting jelzése a BASIC STACK-ben
E446  PUSH HL
E447  POP  IY          ; a BASIC STACK új értéke
E449  RET

E44A  LD   A,9A        ; (=)
E44C  CALL FD54        ; szintaktikai hiba, ha nem "=" követke-
                         zik
E44F  JP   F28D        ; az értékadó kifejezés kiértékelése

A LOMEM rutin:
E452  CALL FAC4        ; az aritmetikai kifejezés kiértékelése
E455  LD   DE,(1720)   ; DE-be VLOMEM
E459  OR   A
E45A  SBC  HL,DE
E45C  ADD  HL,DE
E45D  JP   C,FB14      ; "Bad argument" hiba, ugrás
E460  LD   DE,(1722)   ; DE-be TEXT
E464  SBC  HL,DE
E466  JP   Z,DBB4      ; a program címe nem változik, ugrás a
                         következö BASIC utasításra
E469  PUSH HL
E46A  CALL C,FC86      ; HL=-HL, ha a program új kezdete kisebb
E46D  LD   C,L
E46E  LD   B,H         ; BC-be HL
E46F  EX   DE,HL
E470  JR   C,13        ; ugrás az E485H címre, ha CARRY
E472  POP  DE
E473  PUSH DE
E474  PUSH HL
E475  INC  D
E476  JP   Z,FB14      ; "Bad argument" hiba, ugrás
E479  CALL FC99        ; van elég memória?
E47C  POP  HL
E47D  PUSH HL
E47E  CALL DCC9        ; a program mozgatása a megadott címre
E481  POP  HL
E482  ADD  HL,BC
```

```
  E483  JR   08          ; ugrás az E48DH címre
  E485  OR   A
  E486  SBC  HL,BC
  E488  PUSH HL
  E489  CALL DCE9        ; a program mozgatása a megadott címre
  E48C  POP  HL
  E48D  LD   (1722),HL   ; az új TEXT eltárolása
  E490  POP  BC
  E491  BIT  2,(IX+00H)
  E495  JR   Z,0C        ; parancsként lett a LOMEM utasítás ki-
                           adva, ugrás az E4A3H címre
  E497  LD   HL,(170C)   ; HL-be START
  E49A  ADD  HL,BC
  E49B  LD   (170C),HL   ; az új START elmentése
  E49E  PUSH BC
  E49F  EXX
  E4A0  POP  DE
  E4A1  ADD  HL,DE       ; a következő program byte címének módo-
                           sítása
  E4A2  EXX
  E4A3  CALL DCFC        ; a BASIC változóinak írása
  E4A6  JP   DBB1        ; a következő utasítással folytatódik a
                           program

  E4A9  RST  08H
  E4AA  DB   08          ; "No FOR" hibajelzés

  E4AB  EXX
  E4AC  LD   A,B
  E4AD  EXX
  E4AE  CP   A4          ; (,)
  E4B0  JP   NZ,DBB4     ; nincs több ciklusváltozó, a NEXT ru-
                           tin vége
  E4B3  CALL FC43        ; a következő program byte

A NEXT rutin:
  E4B6  SET  2,(IX+00H)  ; program mód
  E4BA  JR   NC,27       ; ugrás az E4E3H címre, ha nincs megadva
                           ciklusváltozó
  E4BC  CP   03
  E4BE  JR   Z,07        ; ugrás az E4C7H címre, ha A=3
  E4C0  AND  82
  E4C2  JP   NZ,FD5A     ; szintaktikai hiba, ugrás

  E4C5  RST  08H
```

```
  E4C6  DB   0E          ; "Type mismatch" hiba

  E4C7  CALL F42E        ; a ciklusváltozó tömb elem?
  E4CA  EX   DE,HL
  E4CB  LD   HL,09FD
***
  E4CC  ADD  IY,BC
***
  E4CE  LD   A,2B        ; A-ba 43
  E4D0  CALL FCE7        ; az utóbbi FOR-hoz tartozó bejegyzés
                           megkeresése
  E4D3  JR   C,D4        ; nem volt FOR, ugrás a No FOR hibára
  E4D5  LD   C,05
  E4D7  ADD  HL,BC       ; HL-be a FOR ciklusváltozója címének a
                           címe
  E4D8  LD   C,2B        ; C-be 43
  E4DA  LD   A,(HL)
  E4DB  CP   E
  E4DC  JR   NZ,EE       ; a NEXT ciklusváltozója nem ehhez a
                           FOR-hoz tartozik, ugrás az E4CCH címre
  E4DE  INC  HL
  E4DF  LD   A,(HL)
  E4E0  CP   D
  E4E1  JR   NZ,E9       ; a NEXT ciklusváltozója nem ehhez a
                           FOR-hoz tartozik, ugrás az E4CCH címre
  E4E3  LD   A,2B        ; A-ba 43
  E4E5  CALL FCE7        ; az utóbbi FOR-hoz tartozó bejegyzés
                           megkeresése
  E4E8  JR   C,BF        ; nincs FOR, hibajelzés
  E4EA  INC  HL
  E4EB  LD   E,(HL)
  E4EC  INC  HL
  E4ED  LD   D,(HL)      ; DE-be a ciklusváltozó címe
  E4EE  PUSH DE
  E4EF  INC  HL
  E4F0  LD   E,(HL)
  E4F1  INC  HL
  E4F2  LD   D,(HL)      ; a FOR-t tartalmazó sor kezdöcíme DE-be
  E4F3  PUSH DE
  E4F4  INC  HL
  E4F5  LD   E,(HL)
  E4F6  INC  HL
  E4F7  LD   D,(HL)      ; a FOR-t követö utasítás címe DE-be
  E4F8  PUSH DE
  E4F9  INC  HL
```

```
E4FA  EX   DE,HL
E4FB  LD   DE,0022      ; DE-be 34
E4FE  ADD  IY,DE
E500  CALL FA63         ; a ciklusváltozó értékét tölti a BASIC
                          STACK-be
E503  DEC  DE
E504  LD   HL,FFF7
E507  ADD  HL,DE
E508  LD   BC,0009
E50B  LDDR              ; a ciklusváltozó értékének duplikálása
                          a FOR bejegyzésben
E50D  INC  DE
E50E  PUSH DE
E50F  POP  IY
E511  BIT  7,(IY+08H)
E515  PUSH AF
E516  CALL F493         ; a 0. aritmetikai rutin: a ciklusválto-
                          zóhoz hozzáadja a STEP értékét
E519  POP  AF
E51A  POP  HL
E51B  PUSH AF
E51C  PUSH IY
E51E  CALL FB28         ; a 13. aritmetikai rutin: elmenti a
                          ciklusváltozó új értékét
E521  POP  IY
E523  CALL F693         ; a TO értékének és a ciklusváltozó ér-
                          tékének az összehasonlítása
E526  POP  BC
E527  POP  DE
E528  POP  HL
E529  LD   A,C
E52A  RLA
E52B  JR   NC,06        ; ugrás az E533H címre, ha
E52D  JR   Z,07         ; ugrás az E536H címre, ha a ciklusvál-
                          tozó értéke egyenlö TO értékével
E52F  JP   P,E4AB       ; a következö ciklusváltozó vizsgálata
E532  XOR  A
E533  JP   M,E4AB       ; a következö ciklusváltozó vizsgálata
E536  LD   (170C),DE    ; START új értéke a FOR sora
E53A  LD   DE,FFD5
E53D  ADD  IY,DE        ; IY az aktuális FOR bejegyzés alá mutat
E53F  JP   DB81         ; ugrás a FOR-t követö utasításra

Az OUT rutin:
  E542  CALL FB1B       ; A-ba a periféria címe
```

```
  E545  LD   C,A
  E546  LD   A,A4          ; (,)
  E548  CALL FD54          ; ha nem "," következik, szintaktikai
                             hiba
  E54B  CALL FB1B          ; A-ba a kiküldendö byte
  E54E  OUT  (C),A
  E550  JP   DBB1          ; az OUT rutin vége

A POKE rutin:
  E553  CALL FCFF          ; a töltendö memóriacím HL-be
  E556  LD   A,A4          ; (,)
  E558  CALL FD54          ; szintaktikai hiba, ha nem "," követ-
                             kezik
  E55B  CALL FB1B          ; a töltendö byte A-ba
  E55E  LD   C,A
  E55F  CALL FFE7          ; ROM cím?
  E562  JR   C,03          ; nem, ugrás az E567H címre
  E564  DI                 ; maszkolható megszakításkérés elfogadá-
                             sának tiltása
  E565  OUT  02,A          ; memórialapozás
  E567  LD   (HL),C
  E568  LD   A,70
  E56A  OUT  02,A          ; memórialapozás: U U U S lapkiosztás
  E56C  EI                 ; megszakítás engedélyezés
  E56D  JP   DBB1          ; a POKE rutin vége

Az LPRINT rutin:
  E570  LD   C,40
  E572  LD   DE,200E

A PRINT és az OUTPUT rutin:
***
  E573  LD   C,20
***
  E575  LD   (IX+05H),C
  E578  EX   AF,AF'
  E579  XOR  A
  E57A  EX   AF,AF'
  E57B  SCF
  E57C  CALL NC,FC43       ; a következö program byte
  E57F  CP   AF            ; (USING)
  E581  JR   NZ,23         ; ugrás az E5A6H címre, ha nem USING
A USING rutin:
  E583  CALL FC43          ; a következö program byte
  E586  CALL F294          ; a formátum string a BASIC STACK-be
```

```
E589  LD   HL,1831     ; HL-be a BUFFER címe
E58C  LD   (HL),7F
E58E  PUSH HL
E58F  CALL FB41        ; a formátum string a BUFFER-be
E592  POP  HL
E593  INC  HL
E594  LD   E,(HL)      ; E-be a formátum string hossza
E595  LD   D,00
E597  INC  HL
E598  LD   (172E),HL   ; a formátum string kezdetének címe
E59B  ADD  HL,DE
E59C  LD   (HL),D
E59D  LD   (1730),HL   ; a formátum string végének címe
E5A0  EX   AF,AF'
E5A1  SET  0,A
E5A3  EX   AF,AF'
E5A4  JR   14          ; ugrás az E5BAH címre

E5A6  CP   B0          ; (AT)
E5A8  JR   NZ,22       ; nem AT, ugrás az E5DCH címre
E5AA  CALL FB16        ; a sor pozíció A-ba
E5AD  LD   C,A
E5AE  LD   A,A4        ; (,)
E5B0  CALL FD54        ; ha nem "," következik, szintaktikai
                         hiba
E5B3  CALL FB1B        ; az oszlop pozíció A-ba
E5B6  LD   B,A
E5B7  CALL E68B        ; a BTEXT rutin meghívása
E5BA  EXX
E5BB  LD   A,B
E5BC  EXX
E5BD  EX   AF,AF'
E5BE  OR   A
E5BF  EX   AF,AF'
E5C0  CP   A4          ; (,)
E5C2  JR   Z,B8        ; ugrás az E57CH címre, ha ","
E5C4  CP   FD
E5C6  JR   Z,0E        ; ugrás az E5D6H címre, ha ":"
E5C8  JR   NC,0F       ; ugrás az E5D9H cimre, ha utasítás vége

E5CA  RST  08H
E5CB  DB   01          ; szintaktikai hiba

E5CC  CALL FC00        ; adva van periféria?
E5CF  JR   Z,EC        ; igen, ugrás az E5BDH címre
```

```
  E5D1  EX   AF,AF'
  E5D2  JR   C,F6          ; szintaktikai hiba, ugrás az E5CAH-ra
  E5D4  EX   AF,AF'
  E5D5  SCF
  E5D6  CALL NC,FC43       ; a következő program byte
  E5D9  EX   AF,AF'
  E5DA  SCF
  E5DB  EX   AF,AF'
  E5DC  EX   AF,AF'
  E5DD  BIT  0,A
  E5DF  PUSH AF
  E5E0  CALL NZ,E69C       ; ugrás, ha USING
  E5E3  POP  AF
  E5E4  EX   AF,AF'
  E5E5  EXX
  E5E6  LD   A,B
  E5E7  EXX
  E5E8  CP   A4
  E5EA  JR   NZ,06         ; ugrás az E5F2H címre, ha nem ","
  E5EC  LD   A,09          ; A-ba a TAB karakter kódja
  E5EE  CALL FE9A          ; a TAB karakter kiíratása
  E5F1  JP   NZ,A0FE       ; üres utasítás, mert ha ide adódik a
                             vezérlés, a ZERO flag mindig 0
***
  E5F2  CP   A0
***
  E5F4  JR   NZ,08         ; ugrás az E5FEH címre, ha nem ";)
  E5F6  CALL FC43          ; a következő program byte
  E5F9  EX   AF,AF'
  E5FA  OR   A
  E5FB  JP   E67F

  E5FE  CP   B6            ; (TAB)
  E600  JR   NZ,15         ; ugrás az E617H címre, ha nem TAB
  E602  LD   A,96
  E604  CALL FD51          ; szintaktikai hiba, ha nem "("
  E607  CALL FB1B          ; A-ba a TAB pozíció
  E60A  LD   B,A
  E60B  LD   A,95
  E60D  CALL FD54          ; szintaktikai hiba, ha nem ")"
  E610  LD   C,00
  E612  CALL E68B          ; a BTEXT rutin hívása
  E615  JR   66            ; ugrás az E67DH címre
  E617  CP   FD
  E619  JR   NC,68         ; ugrás az E683H címre, ha utasítás vége
```

```
E61B  EX   AF,AF'
E61C  BIT  0,A
E61E  PUSH AF
E61F  JR   Z,17          ; ugrás az E638H címre, ha nem USING
E621  EX   AF,AF'
E622  LD   HL,(172E)     ; HL-be az aktuális formátum byte címe
E625  LD   DE,(1730)     ; DE-be a formátum string vége
E629  OR   A
E62A  SBC  HL,DE
E62C  JR   C,0B          ; ugrás az E639H címre, ha HL<DE
E62E  LD   HL,1833       ; HL-be a BUFFER címe +2
E631  LD   (172E),HL
E634  CALL E69C          ; a formátumvezérlö karakter azonosítása
E637  EX   AF,AF'
E638  EX   AF,AF'
E639  EXX
E63A  LD   A,B
E63B  EXX
E63C  CP   02
E63E  JR   NC,25         ; ugrás az E665H címre, ha A>1
E640  CALL F294
E643  RLA
E644  JR   NC,23         ; ugrás az E669H címre, ha CARRY
E646  POP  AF
E647  LD   HL,FF9C       ; EXT ROM rutin cím HL-be
E64A  JP   NZ,FFF0       ; ugrás az EXT. ROM-ba, ha USING
E64D  OR   FF
E64F  PUSH IY
E651  POP  HL
E652  INC  HL
E653  PUSH AF
E654  CALL FE7F          ; a BASIC STACK-böl string kiíratása
E657  CALL FA21          ; a string által elfoglalt terület fel-
                           szabadítása
E65A  POP  AF
E65B  DEC  A
E65C  JP   M,E67D
E65F  PUSH AF
E660  CALL FEC7          ; egy szóköz kiíratása
E663  JR   F5            ; urgás az E65AH címre
E665  CALL F0A7          ; kifejezés kiértékelése, az eredmény a
                           BASIC STACK-be
E668  RLA
E669  JP   NC,FD5A       ; szintaktikai hiba, ugrás
E66C  POP  AF
```

```
E66D  LD   HL,FCB5     ; EXT. ROM rutin címe HL-be
E670  JP   NZ,FFF0     ; ugrás az EXT. ROM-ba, ha
E673  PUSH IY
E675  POP  HL
E676  INC  HL
E677  CALL FEBA        ; szám kiíratása
E67A  CALL FA1B        ; 9 byte felszabadítása a BASIC STACK-
                         ből
E67D  EX   AF,AF'
E67E  SCF
E67F  EX   AF,AF'
E680  JP   E5DC        ; következö kiíratni való
E683  EX   AF,AF'
E684  CALL C,FE93      ; új sor
E687  EX   AF,AF'
E688  JP   DBB4        ; vége az utasításnak
E68B  LD   A,(1705)    ; A-ba az output eszközt meghatározó
                         byte
E68E  CP   20
E690  JR   Z,03        ; editor, ugrás
E692  CP   00
E694  RET  NZ          ; video, ret
E695  OR   03          ; az eszköz 3. rutinja
E697  CALL 001B        ; az eszköz rutin meghívása
E69A  RST  10H         ; hibakezelés
E69B  RET
E69C  LD   DE,(1730)   ; DE-be a formátum string vége
E6A0  LD   HL,(172E)   ; HL-be a formátum string aktuális byte-
                         jának címe
E6A3  LD   A,(HL)      ; A-ba az aktuális byte
E6A4  OR   A
E6A5  SBC  HL,DE
E6A7  RET  NC
E6A8  ADD  HL,DE
E6A9  PUSH HL
E6AA  LD   HL,E6BD     ; HL-be a formátum vezérlö karakterek
                         táblájának a címe
E6AD  LD   BC,000A
E6B0  CPIR
E6B2  POP  HL
E6B3  RET  Z           ; az adott karakter formátum vezérlö
E6B4  CALL FE9A        ; a karakter kiíratása
E6B7  INC  HL
E6B8  LD   (172E),HL   ; HL eltárolása
E6BB  JR   E6          ; ugrás az E6A3H címre
```

```
A PRINT USING formátum vezérlő karakterek :
  E6BD  DB   3C    ; <
  E6BE  DB   3E    ; >
  E6BF  DB   23    ; #
  E6C0  DB   2A    ; *
  E6C1  DB   25    ; %
  E6C2  DB   2B    ; +
  E6C3  DB   2D    ; -
  E6C4  DB   24    ; $
  E6C5  DB   5E    ; ^
  E6C6  DB   2E    ; .


A RANDOMIZE rutin
  E6C7  LD   A,R
  E6C8  LD   E,A
  E6C9  LD   (1709),A    ; a véletlenszerű R regisztertartalom
                           eltárolása
  E6CC  LD   HL,(0B1D)   ; HL-be az INTINC IT számláló tartalma
  E6CF  LD   (170A),HL
  E6D2  CALL E6D8        ; egy művelet sorozat végzése
  E6D5  JP   DBB1        ; vége az utasításnak
  E6D8  LD   B,10
  E6DA  LD   A,(1709)    ; az R által meghatározott érték A-ba
  E6DD  LD   HL,(170A)   ; az INTINC által meghatározott érték
                           HL-be
  E6E0  LD   C,A
  E6E1  RRCA
  E6E2  RRCA
  E6E3  RRCA
  E6E4  XOR  C
  E6E5  RLA
  E6E6  RLA
  E6E7  ADC  HL,HL       ; A-tól függően 2*HL vagy 2*HL+1
  E6E9  LD   A,C
  E6EA  ADC  A,A         ; vagy 2*A vagy 2*A+1
  E6EB  DJNZ F3          ; ugrás az E6E0H címre
  E6ED  LD   (170A),HL
  E6F0  LD   (1709),A    ; a véletlen értékek eltárolása
  E6F3  RET

A RESTORE rutin:
  E6F4  LD   HL,(1722)   ; HL-be TEXT
  E6F7  CP   02
  E6F9  JR   NZ,06       : nincs sorszám megadva, ugrás az E701H
```

```
                                címre
  E6FB  CALL FBDE       ; adott sorszámú programsor megkeresése
  E6FE  CALL FC43       ; a következö program byte
  E701  LD   (1712),HL  ; a megadott sor címének eltárolása
  E704  LD   HL,0000
  E707  LD   (1714),HL  ; a két byte törlése
  E70A  JP   DBB4       ; vége az utasításnak

  E70D  RST  08H
  E70E  DB   09         ; "No GOSUB" hibajelzés

A RETURN rutin:
  E70F  LD   A,06       ; A-ba a GOSUB bejegyzés azonosítója
  E711  CALL FCE7       ; GOSUB bejegyzés keresése a BASIC STACK-
                          ben
  E714  JR   C,F7       ; nincs GOSUB, ugrás az E70DH címre
  E716  SET  2,(IX+00H) ; program mód beállítása
  E71A  INC  HL
  E71B  LD   E,(HL)
  E71C  INC  HL
  E71D  LD   D,(HL)     ; DE-be a GOSUB utáni utasítás címe
  E71E  INC  HL
  E71F  LD   C,(HL)
  E720  INC  HL
  E721  LD   B,(HL)     ; BC-be a GOSUB-ot tartalmazó sor címe
  E722  LD   (170C),BC  ; START átírása
  E726  INC  HL
  E727  LD   A,(HL)
  E728  LD   (1702),A   ; az ELSE_FLAG ON GOSUB-nál
  E72B  INC  HL
  E72C  PUSH HL
  E72D  POP  IY
  E72F  EX   DE,HL
  E730  JP   DB81       ; az utasítás vége

A GRAPHICS rutin:
  E733  CALL FB1B       ; a grafikus felbontást meghatározó ki-
                          fejezés értéke A-ba
  E736  LD   C,00
  E738  CP   02
  E73A  JR   Z,0B       ; 2 színü mód, ugrás az E747H címre
  E73C  INC  C
  E73D  CP   04
  E73F  JR   Z,06       ; 4 színü mód, ugrás az E747H címre
  E741  INC  C
```

```
E742  CP   10
E744  JP   NZ,FB14     ; nem 16 színü mód, hibajelzés
E747  RST  30H
E748  DB   04          ; a VMODE rutin meghívása
E749  RST  10H         ; hibakezelés
E74A  JP   DBB1        ; vége az utasításnak

E74D  RST  30H
E74E  DB   08          ; a BON rutin meghívása
E74F  CALL FC43        ; a következö program byte
E752  JR   NC,39       ; nincs több paraméter, ugrás az E78DH
                         címre

A PLOT rutin:
E754  JR   NC,2C       ; ugrás az E782H címre
E756  CP   BE          ; (PAINT)
E758  JR   NZ,05       ; nem PAINT, ugrás az E75FH címre
E75A  RST  30H
E75B  DB   0A          ; a FILL rutin meghívása
E75C  RST  10H         ; hibakezelés
E75D  JR   1B          ; ugrás az E77EH címre
E75F  CP   A0          ; (;)
E761  JR   Z,EA        ; ";", ugrás az E74DH címre
E763  CP   A4          ; (,)
E765  JR   Z,1D        ; ",", ugrás az E784H címre
E767  CALL FAC4        ; HL-be az X koordináta
E76A  LD   C,L
E76B  LD   B,H         ; BC-be HL
E76C  LD   A,A4
E76E  CALL FD54        ; ha nem "," következik, szintaktikai
                         hiba
E771  CALL FAC4        ; HL-be az Y koordináta
E774  EX   DE,HL
E775  PUSH AF
E776  RST  30H
E777  DB   06          ; BABS rutin meghívása
E778  RST  10H         ; hibakezelés
E779  POP  AF
E77A  CP   A0          ; (;)
E77C  JR   Z,CF        ; ha ";", ugrás az E74DH címre
E77E  CP   A4          ; (,)
E780  JR   Z,02        ; ha ",", ugrás az E784H címre
E782  RST  30H
E783  DB   08          ; a BON rutin
E784  RST  30H
```

```
E785  DB   09          ; a BOFF rutin
E786  EXX
E787  LD   A,B
E788  EXX
E789  CP   FD
E78B  JR   C,C2        ; ha nincs vége az utasításnak, ugrás az
                         E74FH címre
E78D  JP   DBB4        ; vége a PLOT utasításnak

A SET rutinok:
E790  LD   HL,E7E4     ; HL-be a SET utasítások ugrótáblája
E793  CALL FD12        ; a soron következő SET rutin címének
                         meghatározása, a rutin végrehajtása
E796  JR   C,42        ; ugrás az E7DAH címre, ha egy SET rutin
                         után van
E798  CALL FB1B        ; a paraméter kifejezés kiértékelése
E79B  LD   E,00
E79D  LD   (001F),A    ; A eltárolása operációs rendszer rutin
                         sorszámként
E7A0  LD   HL,1941
E7A3  PUSH HL
E7A4  LD   B,7F
E7A6  DEC  E
E7A7  JP   P,E7B4      ; ugrás, ha >=0
E7AA  EXX
E7AB  LD   A,B
E7AC  EXX
E7AD  CP   A4          ; (,)
E7AF  JR   NZ,1C       ; nem "," következik, ugrás az E7CDH-ra
E7B1  CALL FC43        ; következő program byte
E7B4  CP   02
E7B6  JR   NC,0D       ; szám következik, ugrás az E7C5H címre
E7B8  CALL F294        ; string a BASIC STACK-be
E7BB  LD   HL,193F
E7BE  LD   (HL),7F
E7C0  CALL FB41        ; string az 1940H címre
E7C3  JR   0D          ; ugrás az E7D2H címre
E7C5  CALL FB1B        ; aritmetikai kifejezés kiértékelése
E7C8  LD   (HL),A
E7C9  INC  HL
E7CA  DJNZ DA          ; ugrás az E7A6H címre
E7CC  INC  B
E7CD  LD   (HL),00
E7CF  INC  HL
E7D0  DJNZ FB          ; ugrás az E7CDH címre
```

```
  E7D2  POP  DE
  E7D3  CALL 001E        ; operációs rendszer rutinjának hívása
  E7D6  RST  10H         ; hibakezelés
  E7D7  EXX
  E7D8  LD   A,B
  E7D9  EXX
  E7DA  CP   A0          ; (;)
  E7DC  JP   NZ,DBB4     ; nem ";", a rutin vége
  E7DF  CALL FC43        ; a következö program byte
  E7E2  JR   AC          ; ugrás az E790H címre

A 'SET' utasítások ugrótáblázata:

  E7E4  DB   C3 11    ; SET MODE        : a rutin címe E7F7
  E7E6  DB   B7 12    ; SET STYLE       : a rutin címe E7FA
  E7E8  DB   C4 13    ; SET INK         : a rutin címe E7FD
  E7EA  DB   BC 14    ; SET PAPER       : a rutin címe E800
  E7EC  DB   C7 15    ; SET DELAY       : a rutin címe E803
  E7EE  DB   B9 16    ; SET RATE        : a rutin címe E806
  E7F0  DB   C8 24    ; SET CHARACTER   : a rutin címe E816
  E7F2  DB   BD 2D    ; SET PALETTE     : a rutin címe E821
  E7F4  DB   AE 34    ; SET BORDER      : a rutin címe E82A

  E7F6  NOP

A SET MODE rutin:
  E7F7  LD   E,00
  E7F9  LD   HL,011E

A SET STYLE rutin:
***
  E7FA  LD   E,01
***
  E7FC  LD   HL,021E

A SET INK rutin:
***
  E7FD  LD   E,02
***
  E7FF  LD   HL,031E

A SET PAPER rutin:
***
  E800  LD   E,03
***
```

```
  E802  LD    HL,1A1E

A SET DELAY rutin:
***
  E803  LD    E,1A
***
  E805  LD    HL,1C1E

A SET RATE rutin:
***
  E806  LD    E,1C
***
  E808  LD    D,00
  E80A  LD    HL,0B4B     ; HL-be a video rendszerváltozó tábla
                            kezdö címe
  E80D  ADD   HL,DE       ; HL-be a megfelelö rendszerváltozó címe
  E80E  CALL  FB1B        ; A-ba a rendszerváltozó értéke
  E811  LD    (HL),A      ; a rendszerváltozó átírása
  E812  EXX
  E813  LD    A,B
  E814  EXX
  E815  RET

A SET CHARACTER rutin:
  E816  POP   HL
  E817  POP   HL
  E818  CALL  FB1B        ; A-ba a karakter kódja
  E81B  LD    C,A
  E81C  LD    A,0B        ; a DEFC rutin kódja
  E81E  JP    E79B        ; a karakter definiáló byte-ok átmozga-
                            tása, a DEFC rutin meghívása

A SET PALETTE rutin:
  E821  POP   HL
  E822  POP   HL
  E823  LD    A,0C        ; a PAL rutin kódja
  E825  LD    E,01
  E827  JP    E79D        ; a paletta kódok átmozgatása, a PAL ru-
                            tin meghívása

A SET BORDER rutin:
  E82A  CALL  FB1B        ; A-ba a border kódja
  E82D  ADD   A,A
  E82E  LD    (0B4F),A    ; a BORDER felülírása
  E831  JR    DF          ; ugrás az E812H címre
```

A SOUND utasítások:

```
E833  LD   HL,0D15     ; HL-be az alap PITCH érték
E836  LD   (172A),HL   ; PITCH eltárolása
E839  LD   HL,3207     ; HL-be VOLUME és DURATION alapértéke
E83C  LD   (002C),HL
E83F  LD   HL,0B15
E842  LD   (HL),FF     ; a SND_IRQ beállítása
E844  SCF
E845  CALL NC,FC43
E848  CP   A0          ; (;)
E84A  JR   Z,23        ; ugrás az E86FH címre, ha ";"
E84C  LD   HL,E87B     ; HL-be a SOUND utasítások ugrótáblájá-
                         nak címe
E84F  CALL FD12        ; a megfelelö rutin címének meghatározá-
                         sa, ugrás a rutinra
E852  CP   A4          ; (,)
E854  JR   Z,EF        ; ugrás az E845H címre, ha ","
E856  PUSH AF
E857  LD   DE,(172A)   ; DE-be PITCH
E85B  LD   BC,(002C)   ; B-be DURATION, C-be VOLUME
E85F  RST  30H
E860  DB   33          ; a TONE_SET rutin meghívása
E861  XOR  A
E862  LD   (0B15),A    ; a SND_IRQ törlése
E865  POP  AF
E866  CP   FD
E868  JR   NC,0E       ; utasítás vége, ugrás az E878H címre
E86A  CP   A0          ; (;)
E86C  JP   NZ,FD5A     ; ha nem ";", szintaktikai hiba
E86F  XOR  A
E870  LD   (0B15),A    ; a SND_IRQ törlése
E873  CALL FC43        ; következö program byte
E876  JR   C,CD        ; nincs vége a SOUND utasításnak, ugrás
                         az E845H címre
E878  JP   DBB4        ; SOUND utasítás vége
```

A 'SOUND' utasítások ugrótáblája:

```
E87B  DB   BB 05          ; PITCH     : a rutin címe E882
E87D  DB   C6 0C          ; DURATION  : a rutin címe E88B
E87F  DB   B3 0B          ; VOLUME    : a rutin címe E88C

E901  NOP
```

```
A PITCH rutin:
  E882  CALL FAC4        ; HL-be PITCH értéke
  E885  LD   (172A),HL
  E888  RET  P
  E889  RST  08H
  E88A  DB   04          ; "Bad argument" hibajelzés

A DURATION rutin:
  E88B  OR   37

A VOLUME rutin:
***
  E88C  SCF
  E88D  LD   DE,002C
  E890  JR   C,01        ; ugrás az E893H címre, ha VOLUME
  E892  INC  DE
  E893  CALL FB1B        ; a paraméter A-ba
  E896  LD   (DE),A      ; a paraméter eltárolása
  E897  EXX
  E898  LD   A,B
  E899  EXX
  E89A  RET

A CLOSE utasítás rutin:
  E89B  LD   C,50        ; alapértelmezés szerinti eszköz a ka-
                           zetta
  E89D  CALL FBEE        ; más eszköz van megadva?
  E8A0  CP   E1          ; (OUTPUT)
  E8A2  JR   Z,0B        ; ha OUTPUT, ugrás az E8AFH címre
  E8A4  SET  7,C
  E8A6  CP   FD
  E8A8  JR   NC,08       ; utasítás vége, ugrás az E8B2H címre
  E8AA  CP   EC          ; (INPUT)
  E8AC  JP   NZ,FD5A     ; nem INPUT, szintaktikai hiba
  E8AF  CALL FC43        ; a következő program byte
  E8B2  LD   A,C
  E8B3  AND  7F
  E8B5  CP   50
  E8B7  JR   Z,04        ; kazetta CLOSE, ugrás az E8BDH címre
  E8B9  CP   60
  E8BB  JR   NZ,0A       ; nem soros vonal CLOSE, ugrás az E8C7H
                           címre, hibajelzés
  E8BD  LD   A,C
  E8BE  OR   04          ; a CLOSE a 4. eszköz rutin
  E8C0  CALL 001B        ; az eszközrutin meghívása
```

```
E8C3  RST  10H        ; hibakezelés
E8C4  JP   DBB1       ; vége az utasításnak

E8C7  RST  08H
E8C8  DB   FF         ; nem létező funkciószám hívása hiba

Az OPEN utasítás rutinja:
E8C9  LD   C,50       ; alap eszköz hozzárendelés: kazetta
E8CB  CALL FBEE       ; van más eszköz megadva?
E8CE  EX   AF,AF'
E8CF  LD   A,C
E8D0  CP   50
E8D2  JR   Z,04       ; kazetta OPEN, ugrás az E8D8H címre
E8D4  CP   60
E8D6  JR   NZ,EF      ; nem soros vonali OPEN, ugrás az E8C7H
                        címre, hiba
E8D8  OR   03         ; az OPEN a harmadik eszköz rutin
E8DA  LD   C,A
E8DB  EX   AF,AF'
E8DC  LD   DE,19CE    ; DE-be a FILENAME címe
E8DF  CP   E1         ; (OUTPUT)
E8E1  JR   Z,0F       ; OPEN OUTPUT ,ugrás az E8F2H címre
E8E3  SET  7,C
E8E5  CP   02
E8E7  JR   C,10       ; van file név megadva, ugrás az E8F9H
                        címre
E8E9  CP   FD
E8EB  JR   NC,18      ; az utasítás vége, ugrás az E905H címre
E8ED  CP   EC         ; (INPUT)
E8EF  JP   NZ,FD5A    ; nem INPUT, szintaktikai hiba
E8F2  CALL FC43       ; a következő program byte
E8F5  CP   02
E8F7  JR   NC,0C      ; nincs file név megadva, ugrás az E905H
                        címre
E8F9  PUSH BC
E8FA  CALL F294       ; file név a BASIC STACK-ből a FILENAME
                        pufferbe
E8FD  PUSH IY
E8FF  CALL FA21       ; a file név felszabadítása a BASIC
                        STACK-ből
E902  POP  DE
E903  INC  DE
E904  POP  BC
E905  LD   A,C
E906  LD   (0B6B),A
```

```
  E909  CALL 001B       ; az eszköz rutin meghívása
  E90C  RST  10H        ; hibakezelés
  E90D  JP   DBB1       ; rutin vége

A GET rutin:
  E910  LD   C,10       ; alap eszköz a billentyüzet
  E912  CALL FBEE       ; van más eszköz hozzárendelve?
  E915  EX   AF,AF'
  E916  LD   A,C
  E917  OR   81         ; 1. eszköz rutin: karakter beolvasás
  E919  CALL 001B       ; az eszköz rutin meghívása
  E91C  PUSH IY
  E91E  POP  HL
  E91F  CP   EC         ; file vége hiba
  E921  JR   Z,06       ; file vége hiba esetén ugrás az E929H
                          címre
  E923  OR   A
  E924  RST  10H        ; hibakezelés
  E925  DEC  HL
  E926  LD   (HL),C     ; a beolvasott karakter eltárolása
  E927  INC  A
  E928  LD   C,AF
***
  E929  XOR  A
***
  E92A  DEC  HL
  E92B  LD   (HL),A     ; a string hossza
  E92C  DEC  HL
  E92D  LD   (HL),01    ; stringhez tartozó bejegyzés
  E92F  EX   AF,AF'
  E930  CP   FD
  E932  JP   NC,DBB4    ; változó nélküli GET, az utasítás vége
  E935  PUSH HL
  E936  POP  IY
  E938  CP   01
  E93A  JR   NZ,03      ; ugrás az E93FH címre, ha A><1
  E93C  EXX
  E93D  BIT  2,C
  E93F  JP   NZ,FD5A    ; szintaktikai hiba, ugrás
  E942  BIT  3,C
  E944  CALL NZ,F40B    ; új változó bejegyzése
  E947  EXX
  E948  CALL F42E       ; tömb elem?
  E94B  CALL FB41       ; változó átmozgatása a BASIC STACK-ből
                          a változó területre
```

```
  E94E  JP    DBB1          ; az utasítás vége

A LOAD rutin:
  E951  CALL  EA35          ; a file megnyitása, fej beolvasása
  E954  CALL  DE10          ; a BASIC változók alaphelyzetbe
  E957  PUSH  HL
  E958  LD    DE,(19E1)     ; a file méret DE-be
  E95C  PUSH  DE
  E95D  CALL  FC99          ; elég a memória?
  E960  POP   BC
  E961  POP   DE
  E962  PUSH  DE
  E963  LD    A,(1705)      ; A-ba az input-eszköz byte
  E966  OR    82            ; blokk input kódja
  E968  CALL  001B          ; a blokk input rutin hívása
  E96B  PUSH  AF
  E96C  CALL  NZ,DE10       ; hiba esetén változók alaphelyzetbe
  E96F  POP   AF
  E970  RST   10H           ; hibakezelö rutin
  E971  CALL  DCFC          ; bizonyos változók alaphelyzetbe
  E974  CALL  EA5A          ; a file zárása
  E977  POP   HL
  E978  LD    A,(19E3)      ; A-ba az auto-run byte
  E97B  OR    A
  E97C  JP    Z,DADA        ; nincs auto-run
  E97F  JP    DE23          ; RUN

A SAVE rutin:
  E982  XOR   A
  E983  CALL  E9FB          ; file megnyitása írásra
  E986  LD    DE,19EF       ; DE-be a HEAD_BUFFER kezdete
  E989  LD    B,10
  E98B  XOR   A
  E98C  DEC   DE
  E98D  LD    (DE),A        ; a HEAD_BUFFER törlése
  E98E  DJNZ  FC            ; ugrás az E98CH címre
  E990  PUSH  DE
  E991  INC   A
  E992  LD    (19E0),A      ; a tipusbyte 1, program file
  E995  LD    A,(1707)      ; A-ba az AUTO byte
  E998  LD    (19E3),A
  E99B  CALL  DD41          ; HL-be a program vége
  E99E  INC   HL
  E99F  LD    DE,(1722)     ; DE-be TEXT
  E9A3  XOR   A
```

```
E9A4  LD   (1707),A   ; az AUTO byte törlése
E9A7  SBC  HL,DE
E9A9  LD   (19E1),HL  ; a program méret eltárolása
E9AC  EX   SP,HL
E9AD  PUSH DE
E9AE  LD   B,10
E9B0  LD   A,(1705)   ; A-ba az output-eszköz byte
E9B3  OR   01         ; karakter output
E9B5  LD   (001F),A
E9B8  LD   C,(HL)     ; C-be a HEAD_BUFFER egy byte-ja
E9B9  INC  HL
E9BA  PUSH BC
E9BB  CALL 001E       ; a karakter output rutin hívása
E9BE  RST  10H        ; hibakezelö rutin
E9BF  POP  BC
E9C0  DJNZ F6         ; ugrás az E9B8H címre
E9C2  POP  DE
E9C3  POP  BC
E9C4  LD   A,(1705)   ; A-ba az output-eszköz byte
E9C7  OR   02         ; blokk output
E9C9  CALL 001B       ; a blokk output rutin hívása
E9CC  RST  10H        ; a hibakezelö rutin
E9CD  CALL EA5A       ; a file zárása
E9D0  JP   DBB1       ; az utasítás vége

A VERIFY rutin:
E9D3  CALL EA35       ; file megnyitása olvasásra és a fej be-
                        olvasása
E9D6  CALL DD41       ; a program végének meghatározása
E9D9  INC  HL
E9DA  LD   DE,(1722)  ; DE-be TEXT
E9DE  OR   A
E9DF  SBC  HL,DE      ; HL-be a program mérete
E9E1  LD   BC,(19E1)  ; BC-be a file mérete
E9E5  SBC  HL,BC
E9E7  JP   NZ,EA58    ; a program és a file különbözö méretü,
                        ugrás
E9EA  LD   A,(1705)   ; A-ba az eszköz byte
E9ED  OR   85         ; verify
E9EF  CALL 001B       ; a verify rutin hívása
E9F2  RST  10H        ; hibakezelö rutin
E9F3  CALL EA5A       ; a file zárása
E9F6  JP   DBB1       ; az utasítás vége
```

```
File megnyitása olvasásra:
  E9F9  LD   A,80          ; input rutin lesz
File megnyitása írásra:
  E9FB  SET  3,(IX+00H)
  E9FF  PUSH AF
  EA00  LD   C,50          ; az eszköz a kazetta
  EA02  CALL FBEE          ; van más eszköz megadva?
  EA05  EX   AF,AF'
  EA06  LD   A,C
  EA07  CP   50
  EA09  JR   Z,05          ; ugrás az EA10H címre, ha kazetta
  EA0B  CP   60
  EA0D  JP   NZ,E8C7       ; ugrás, ha nem soros vonal
  EA10  POP  BC
  EA11  OR   B
  EA12  LD   (1705),A      ; az eszköz byte eltárolása
  EA15  OR   03            ; open rutin
  EA17  LD   (001F),A
  EA1A  EX   AF,AF'
  EA1B  LD   DE,19CE       ; DE-be a FILENAME címe
  EA1E  CP   02
  EA20  JR   NC,0A         ; nincs file név megadva, ugrás az EA2CH
                             címre
  EA22  CALL F294          ; file név a BASIC STACK-ből a FILENAME
                             pufferbe
  EA25  PUSH IY
  EA27  CALL FA21          ; a file név felszabadítása a BASIC
                             STACK-ből
  EA2A  POP  DE
  EA2B  INC  DE
  EA2C  XOR  A
  EA2D  LD   (0B6B),A      ; a BUFFER_FLAG törlése
  EA30  CALL 001E          ; az open rutin meghívása
  EA33  RST  10H           ; hibakezelö rutin
  EA34  RET

  EA35  CALL E9F9          ; open rutin

A file fejrészének beolvasása:
  EA38  LD   A,(1705)      ; A-ba az eszköz byte
  EA3B  OR   81            ; karakter input
  EA3D  LD   (001F),A
  EA40  LD   HL,19DF       ; HL-be a HEAD_BUFFER címe
  EA43  PUSH HL
  EA44  LD   B,10
```

```
  EA46  PUSH BC
  EA47  CALL 001E        ; karakter input rutin hívása
  EA4A  RST  10H         ; hibakezelö rutin
  EA4B  LD   (HL),C      ; a beolvasott karakter eltárolása
  EA4C  INC  HL
  EA4D  POP  BC
  EA4E  DJNZ F6          ; ugrás az EA46H címre
  EA50  POP  HL
  EA51  OR   (HL)
  EA52  JR   NZ,04       ; ugrás az EA58H címre, ha a fej elsö
                           byte-ja nem 0
  EA54  INC  HL
  EA55  INC  A
  EA56  SUB  (HL)
  EA57  RET  Z           ; a rutin vége, ha minden rendben

  EA58  RST  08H
  EA59  DB   10          ; Bad file hibajelzés

Close rutin:
  EA5A  LD   A,(1705)    ; A-ba az eszköz byte
  EA5D  OR   04          ; close
  EA5F  CALL 001B        ; a close eszköz rutin hívása
  EA62  RST  10H         ; hibakezelö rutin
  EA63  RES  3,(IX+00H)
  EA67  RET              ; a rutin vége

A 14. aritmetikai rutin:
  EA68  LD   A,96
  EA6A  CALL FD54        ; ha nem "(" következik, szintaktikai
                           hiba
  EA6D  CALL F0A7        ; a zárójelben levö aritmetikai kifeje-
                           zés kiértékelése
  EA70  LD   A,95
  EA72  JP   FD54        ; ha nem ")" következik, szintaktikai
                           hiba, különben a rutin vége

A 14. string rutin (string kifejezés értéke a BASIC STACK-be):
  EA75  LD   A,96
  EA77  CALL FD54        ; ha nem "(" következik, szintaktikai
                           hiba
  EA7A  CALL F294        ; a zárójelben levö string kifeje-
                           zés kiértékelése
  EA7D  LD   A,95
  EA7F  JP   FD54        ; ha nem ")" következik, szintaktikai
                           hiba, külön a rutin vége
```

Az 5. aritmetikai rutin:

```
EA82  LD   HL,(1718)
EA85  LD   A,(HL)       ; A-ba a konstans sorszáma
EA86  INC  HL
EA87  LD   (1718),HL
EA8A  LD   L,A
EA8B  XOR  A
EA8C  LD   H,A
EA8D  PUSH HL
EA8E  ADD  HL,HL
EA8F  ADD  HL,HL
EA90  ADD  HL,HL
EA91  POP  DE
EA92  SBC  HL,DE        ; HL-be a konstans sorszámának 7-szerese
EA94  LD   DE,C111      ; DE-be a konstans-tábla címe
EA97  ADD  HL,DE        ; HL-be a kiválasztott konstans címe
EA98  JR   08           ; ugrás az EAA2H címre
```

A 7. aritmetikai rutin:

```
EA9A  LD   HL,19C7      ; HL-be az Y aritmetikai regiszter címe
EA9D  JR   03           ; ugrás az EAA2H címre
```

A 6. aritmetikai rutin:

```
EA9F  LD   HL,19C0      ; HL-be az X aritmetikai regiszter címe
```

A HL által címzett 7 byte-os szám mozgatása a BASIC STACK-be:

```
EAA2  CALL FC8E         ; van elég hely?
EAA5  LD   DE,0006
EAA8  ADD  HL,DE        ; HL a szám végére mutat
EAA9  PUSH IY
EAAB  POP  DE
EAAC  DEC  DE
EAAD  LDD               ; a kitevö átmozgatása
EAAF  XOR  A            ; A-ba 0
EAB0  LD   (DE),A       ; két további számjegy
EAB1  DEC  DE
EAB2  LD   BC,0006
EAB5  LDDR              ; 6 byte átmozgatása
EAB7  LD   A,09
EAB9  LD   (DE),A       ; a szám jelzö byte-ja
EABA  PUSH DE
EABB  POP  IY
EABD  RET               ; a rutin vége
```

```
A 11. aritmetikai rutin:
  EABE  LD   DE,19C7    ; DE-be az Y aritmetikai regiszter címe
  EAC1  JR   03         ; ugrás az EAD2H címre

A 10. aritmetikai rutin:
  EAC3  LD   DE,19C0    ; DE-be az X aritmetikai regiszter címe
  EAC6  CALL EAD5       ; 7 byte mozgatása a BASIC STACK-ből
  EAC9  PUSH HL
  EACA  POP  IY         ; az átmozgatott terület felszabadítása
  EACC  RET

A 9. aritmetikai rutin:
  EACD  LD   DE,19C7    ; DE-be az Y aritmetikai regiszter címe
  EAD0  JR   03         ; ugrás az EAD5H címre

A 8. aritmetikai rutin:
  EAD2  LD   DE,19C0    ; DE-be az X aritmetikai regiszter címe

7 byte-os szám mozgatása a BASIC STACK-ből a DE címre:
  EAD5  PUSH IY
  EAD7  POP  HL
  EAD8  INC  HL         ; a jelző byte átugrása
  EAD9  LD   BC,0006
  EADC  LDIR            ; 6 byte átmozgatása
  EADE  INC  HL         ; egy byte átugrása
  EADF  LDI             ; a kitevő átmozgatása
  EAE1  RET

Az ABS függvény rutinja:
  EAE2  DW   0000       ; ez a szimbólumtábla első eleme
  EAE4  DB   03
  EAE5  DB   'ABS'
  EAE8  DB   0A         ; beépített numerikus függvény
  EAE9  CALL EA68       ; 14. aritmetikai rutin
  EAEC  RES  7,(IY+08H) ; az előjel bit törlése
  EAF0  RET             ; az ABS rutin vége

Az ATN függvény rutinja:
  EAF1  CALL FC43       ; a következő program byte
  EAF4  CALL EA68       ; a függvény argumentum a BASIC STACK-be
  EAF7  LD   BC,0000
  EAFA  LD   A,(IY+08H) ; A-ba a kitevő
  EAFD  AND  80         ; A-ba az előjel bit
  EAFF  LD   C,A
  EB00  PUSH BC
```

```
EB01  CALL NZ,F726     ; negatív argumentum, a 4. aritmetikai
                         rutin meghívása
EB02  RST  18H
EB05  DB   08 85 01    ; aritmetikai rutinok hívása
EB08  CALL F693        ; az argumentum és az 1. konstans össze-
                         hasonlítása
EB0B  JP   M,EB1A      ; ha az argumentum > 1, ugrás
EB0E  JR   Z,0A        ; ha az argumentum = 1, ugrás az EB1AH
                         címre
EB10  POP  BC
EB11  LD   B,02
EB13  PUSH BC
EB14  RST  18H         ; aritmetikai rutinok hívása
EB15  DB   05 01 06
EB18  DB   01 8A
EB1A  RST  18H         ; aritmetikai rutinok hívása
EB1B  DB   06 85 02
EB1E  CALL F693        ; az Y aritmetika regiszter tartalma és
                         a 2. konstans összehasonlítása
EB21  JP   M,EB3C      ; a 2. konstans kisebb, ugrás
EB24  JR   Z,16        ; a kettő egyenlő, ugrás az EB3CH címre
EB26  POP  BC
EB27  INC  B
EB28  PUSH BC
EB29  RST  18H         ; aritmetikai rutinok hívása
EB2A  DB   05 04 06
EB2D  DB   02 05 00
EB30  DB   03 05 00
EB33  DB   03 06 00
EB36  DB   05 03 06
EB39  DB   00 01 8A
EB3C  RST  18H         ; aritmetikai rutinok hívása
EB3D  DB   06 0C 0C
EB40  DB   0C 02 08
EB43  DB   05 06 02
EB46  DB   05 05 00
EB49  DB   06 02 06
EB4C  DB   05 08 00
EB4F  DB   06 02 05
EB52  DB   07 00 01
EB55  DB   02 80
EB57  POP  BC
EB58  LD   A,B
EB59  CP   02
EB5B  CALL NC,F726     ; ha B>1, a 4. aritmetikai rutin hívása
```

```
  EB5E  PUSH BC
  EB5F  LD   HL,C214     ; HL-be a 37. konstans címe
  EB62  DEC  B
  EB63  JR   Z,0C        ; ugrás az EB71H címre, ha B 0 lett
  EB65  LD   HL,C206     ; HL-be a 35. konstans címe
  EB68  DEC  B
  EB69  JR   Z,06        ; ugrás az EB71H címre, ha B 0 lett
  EB6B  LD   HL,C20D     ; HL-be a 36. konstans címe
  EB6E  DEC  B
  EB6F  JR   NZ,06       ; ugrás az EB77H címre, ha B nem 0
  EB71  CALL EAA2        ; a HL által címzett szám a BASIC STACK-
                           be kerül
  EB74  CALL F493        ; a 0. aritmetikai rutin hívása
  EB77  POP  BC
  EB78  LD   A,C
  EB79  XOR  (IY+08H)    ; A-ba a kitevö, módosított elöjel bittel
  EB7C  LD   (IY+08H),A
  EB7F  RET              ; az ATN rutin vége

A CHR$ függvény rutinja:
  EB80  DW   EAE2
  EB82  DB   04
  EB83  DB   'CHR$'
  EB87  DB   08
  EB88  CALL EA68        ; az argumentum a BASIC STACK-be
  EB8B  CALL FB1A        ; A-ba az argumentum
  EB8E  PUSH IY
  EB90  POP  HL
  EB91  DEC  HL
  EB92  LD   (HL),A      ; a karakter kód a BASIC STACK-be
  EB93  DEC  HL
  EB94  LD   (HL),01     ; a string hossza 1 karakter
  EB96  DEC  HL
  EB97  LD   (HL),01     ; a string jelzö byte-ja
  EB99  PUSH HL
  EB9A  POP  IY
  EB9C  RET              ; a CHR$ rutin vége

A COS függvény rutinja:
  EB9D  DW   EB80
  EB9F  DB   03
  EBA0  DB   'COS'
  EBA3  DB   0A
  EBA4  CALL EA68        ; az argumentum a BASIC STACK-be
  EBA7  RST  18H
```

```
  EBA8  DB    05 23 80    ; a 35. konstans az X aritmetikai re-
                            giszterbe
  EBAB  JP    EE5E        ; ugrás a SIN rutinra

Az EXP függvény rutinja:
  EBAE  DW    EB9D
  EBB0  DB    03
  EBB1  DB    'EXP'
  EBB4  DB    0A
  EBB5  CALL  EA68        ; az argumentum a BASIC STACK-be
  EBB8  RST   18H         ; aritmetikai rutinok hívása
  EBB9  DB    0C 05 09
  EBBC  DB    82
  EBBD  CALL  ECAF        ; az INT rutin hívása
  EBC0  LD    A,(IY+08H)  ; A-ba a kitevö byte
  EBC3  LD    C,A
  EBC4  AND   7F          ; az elöjel bit törlése
  EBC6  CP    43
  EBC8  JR    C,0A        ; a kitevö kisebb 3-nál, ugrás az EBD4H
                            címre
  EBCA  OR    C
  EBCB  JP    P,F912      ; 0-val osztás hiba, ha P
  EBCE  CALL  FA1B        ; 9 byte felszabadítása a BASIC STACK-
                            böl
  EBD1  JP    F9F9        ; 9 byte-os 0 szám a BASIC STACK-be
  EBD4  CALL  FA92        ; 12. aritmetika rutin
  EBD7  CALL  FAC3        ; a BASIC STACK-ben lévö szám HL-be, ha
                            megfelelö
  EBDA  LD    E,L
  EBDB  LD    D,H         ; DE-be HL
  EBDC  LD    BC,007E
  EBDF  OR    A
  EBE0  ADC   HL,BC
  EBE2  JP    M,EBCE      ; ugrás, ha negatív
  EBE5  SLA   C
  EBE7  SBC   HL,BC
  EBE9  JP    NC,F912     ; 0-val osztás hiba, ha NC
  EBEC  PUSH  DE
  EBED  RST   18H         ; aritmetikai rutinok hívása
  EBEE  DB    09 05 0A
  EBF1  DB    02 03 07
  EBF4  DB    05 0B 02
  EBF7  DB    03 0C 0C
  EBFA  DB    02 08 05
  EBFD  DB    10 00 06
```

```
EC00  DB   02 05 0F
EC03  DB   00 0B 05
EC06  DB   0E 06 02
EC09  DB   05 0D 00
EC0C  DB   06 02 05
EC0F  DB   0C 00 02
EC12  DB   08 07 06
EC15  DB   03 01 05
EC18  DB   00 00 0C
EC1B  DB   00 8A
EC1D  POP  DE
EC1E  BIT  0,E
EC20  JR   Z,15        ; ugrás az EC37H címre, ha Z
EC22  PUSH DE
EC23  RST  18H         ; aritmetikai rutinok hívása
EC24  DB   06 05 20
EC27  DB   02 8A
EC29  POP  DE
EC2A  INC  D
EC2B  DEC  D
EC2C  INC  DE
EC2D  JR   Z,08        ; ugrás az EC37H címre, ha Z
EC2F  JP   M,EC37      ; ugrás, ha negatív
EC32  LD   HL,19C6     ; HL-be az X aritmetikai regiszter ki-
                         tevőjének a címe
EC35  INC  (HL)
EC36  DEC  DE
EC37  LD   HL,19C6     ; HL-be az X aritmetikai regiszter ki-
                         tevőjének a címe
EC3A  XOR  A
EC3B  OR   E
EC3C  RRA
EC3D  ADD  A,(HL)
EC3E  AND  7F
EC40  LD   (HL),A
EC41  JP   EA9F        ; ugrás a 6. aritmetikai rutinra, az EXP
                         rutin vége

A FREE rutin:
EC44  DW   EBAE
EC46  DB   04
EC47  DB   'FREE'
EC4B  DB   0A
EC4C  PUSH IY
EC4E  POP  HL
```

```
  EC4F  LD   DE,(1726)   ; DE-be TOP
  EC53  OR   A
  EC54  SBC  HL,DE       ; a BASIC STACK mutató - TOP
  EC56  LD   DE,0100
  EC59  SBC  HL,DE
  EC5B  LD   DE,7FFF
  EC5E  SBC  HL,DE
  EC60  PUSH DE
  EC61  CALL FA2B        ; HL-ben tárolt szám a BASIC STACK-be
  EC64  POP  HL
  EC65  CALL FA2B        ; HL-ben tárolt szám a BASIC STACK-be
  EC68  JP   F493        ; ugrás a 0. aritmetikai rutinra, a FREE
                           rutin vége

Az IN függvény rutinja:
  EC6B  DW   EC44
  EC6D  DB   02
  EC6E  DB   'IN'
  EC70  DB   0A
  EC71  CALL EA68        ; az argumentum a BASIC STACK-be
  EC74  CALL FB1A        ; A-ba az argumentum byte
  EC77  LD   C,A
  EC78  IN   L,(C)       ; olvasás a megadott perifériáról
  EC7A  LD   H,00
  EC7C  JP   FA2B        ; a HL-ben lévő szám a BASIC STACK-be,
                           az IN rutin vége

  EC7F  RET  NC
  EC80  RLA
  EC81  RET  NC
  EC82  CALL FA1B        ; 9 byte felszabadítása a BASIC STACK-
                           ből
  EC85  RST  18H         ; a 39. konstans a BASIC STACK-be
  EC86  DB   85 27
  EC88  RET

Az INKEY$ rutin:
  EC89  CALL FC43        ; a következő program byte
  EC8C  RST  30H
  EC8D  DB   93          ; a KBD_STATUS rutin hívása
  EC8E  RST  10H         ; hibakezelés
  EC8F  PUSH IY
  EC91  POP  HL
  EC92  DEC  HL
  EC93  LD   A,C         ; A-ba a beolvasott STATUS_FLAG
```

```
  EC94  OR   A
  EC95  JR   Z,06          : nincs beolvasandó karakter, ugrás az
                             EC9DH címre
  EC97  RST  30H
  EC98  DB   91            : a KBD_CHIN rutin hívása
  EC99  RST  10H           : hibakezelés
  EC9A  LD   (HL),C        : karakter a BASIC STACK-be
  EC9B  DEC  HL
  EC9C  INC  A
  EC9D  LD   (HL),A        : string hossza
  EC9E  DEC  HL
  EC9F  LD   (HL),01       : string bejegyzés jelzö byte-ja
  ECA1  PUSH HL
  ECA2  POP  IY
  ECA4  RET                : az INKEY$ rutin vége

Az INT rutin:
  ECA5  DW   EC6B
  ECA7  DB   03
  ECA8  DB   'INT'
  ECAB  DB   0A
  ECAC  CALL EA68          : argumentum a BASIC STACK-be
  ECAF  LD   A,(IY+08H)    : A-ba a kitevö
  ECB2  AND  7F
  ECB4  SUB  40
  ECB6  JR   C,20          : 1-nél kisebb argumentum, ugrás az
                             ECD8H címre
  ECB8  INC  A
  ECB9  PUSH AF
  ECBA  CALL F707          : a negatív argumentum konvertálása
  ECBD  CALL F9DF          : az elsö nem 0 számjegy megkeresése
  ECC0  POP  AF
  ECC1  ADD  A,03
  ECC3  LD   B,A
  ECC4  LD   A,00
  ECC6  CALL C,F7E3
  ECC9  INC  B
  ECCA  LD   A,B
  ECCB  CP   10
  ECCD  JR   C,F5          : ugrás az ECC4H címre, ha A<16
  ECCF  CALL F707          ; negatív szám konvertálása
  ECD2  CALL F9DF          ; az elsö nem 0 számjegy megkeresése
  ECD5  JP   F734          ; a rutin vége, ha az -1<argumentum<1
  ECD8  LD   A,(IY+08H)
  ECDB  OR   A
```

```
  ECDC  JP   P,F9F9      ; ha 0<argumentum<1, ugrás
  ECDF  CALL FA08        ; 1E+00 szám a BASIC STACK-be
  ECE2  JP   F726        ; 4. aritmetikai rutin

Az IO függvény rutinja:
  ECE5  DW   ECA5
  ECE7  DB   02
  ECE8  DB   'IO'
  ECEA  DB   0A
  ECEB  CALL EA68        ; argumentum a BASIC STACK-be
  ECEE  CALL FB1A        ; az argumentum byte A-ba
  ECF1  LD   L,A
  ECF2  LD   H,00        ; HL-be A
  ECF4  LD   DE,1941
  ECF7  ADD  HL,HL
  ECF8  ADD  HL,DE       ; HL-be 1941H+2*argumentum
  ECF9  LD   E,(HL)
  ECFA  INC  HL
  ECFB  LD   D,(HL)
  ECFC  EX   DE,HL
  ECFD  JP   FA2B        ; HL tartalma a BASIC STACK-be számként

A LEN rutin:
  ED00  DW   ECE5
  ED02  DB   03
  ED03  DB   'LEN'
  ED06  DB   0A
  ED07  CALL EA75        ; az argumentum string a BASIC STACK-be
  ED0A  PUSH IY
  ED0C  CALL FA21        ; a string felszabadítása a BASIC STACK-
                           ből
  ED0F  POP  HL
  ED10  INC  HL          ; HL-be a string hosszának címe
  ED11  LD   L,(HL)      ; L-be a string hossza
  ED12  LD   H,00
  ED14  JP   FA2B        ; a string hossza a BASIC STACK-be

A LOG rutin:
  ED17  DW   ED00
  ED19  DB   03
  ED1A  DB   'LOG'
  ED1D  DB   0A
  ED1E  CALL EA68        ; az argumentum a BASIC STACK-be
  ED21  CALL EAC3        ; a 10. aritmetikai rutin
  ED24  LD   HL,19C5     ; HL-be az X aritmetikai regiszter leg-
```

```
                                nagyobb helyiértékének a címe
ED27  XOR  A
ED28  LD   D,A
ED29  OR   (HL)
ED2A  JP   Z,FB14       ; az argumentum 0. hibajelzés
ED2D  INC  HL           ; HL az argumentum kitevöjére mutat
ED2E  LD   E,(HL)
ED2F  INC  E
ED30  DEC  E
ED31  JP   M,FB14       ; negatív argumentum, hibajelzés
ED34  LD   (HL),3F      ; a kitevö byte-ba -1
ED36  PUSH DE
ED37  PUSH HL
ED38  RST  18H          ; aritmetikai rutinok hívása
ED39  DB   06 85 20
ED3C  CALL F693         ; összehasonlítás
ED3F  POP  HL
ED40  JR   Z,03         ; egyenlök, ugrás az ED45H címre
ED42  JP   P,ED49       ; az argumentum <= 32. konstans, ugrás
ED45  POP  DE
ED46  DEC  DE
ED47  INC  (HL)
ED48  PUSH DE
ED49  RST  18H          ; aritmetikai rutinok meghívása
ED4A  DB   06 05 00
EDAD  DB   03 05 00
ED50  DB   03 06 05
ED53  DB   01 00 01
ED36  DB   08 0C 0C
ED39  DB   02 08 05
ED3C  DB   14 02 05
ED3F  DB   13 00 06
ED42  DB   02 05 12
ED45  DB   00 06 02
ED68  DB   06 05 17
ED6B  DB   00 06 02
ED6E  DB   05 16 00
ED71  DB   06 02 05
ED74  DB   15 00 01
ED77  DB   05 18 00
ED7A  DB   82
ED7B  POP  HL
ED7C  LD   DE,003F
ED7F  OR   A
ED80  SBC  HL,DE
```

```
  ED82  CALL FA2B        ; HL értéke a BASIC STACK-be
  ED85  RST  18H         ; aritmetikai rutinok hívása
  ED86  DB   00 05 11
  ED89  DB   82
  ED8A  RET              ; a LOG rutin vége


  ED8B  CALL FAA6        ; a BASIC STACK alsó 18 byte-ját a BASIC
                           STACK aljára másolja
  ED8E  CALL EAD2        ; a 8. aritmetikai rutin
  ED91  CALL F5FB        ; az 1. aritmetikai rutin
  ED94  CALL ECAF        ; az INT rutin
  ED97  RST  18H         ; aritmetikai rutinok hívása
  ED98  DB   02 03 8C
  ED9B  RES  7,(IY+08H)  ; az elöjel bit törlése
  ED9F  CALL EA9F        ; a 6. aritmetikai rutin
  EDA2  CALL F693        ; a BASIC STACK-ben egymás felett lévö
                           számok összehasonlítása
  EDA5  RET  M           ; RET, ha negatív
  EDA6  JP   F9F9        ; 9 byte-os 0 a BASIC STACK-be

Az ORD függvény rutinja:
  EDA9  CALL FC43        ; a következö program byte
  EDAC  CALL EA75        ; az argumentum string a BASIC STACK-be
  EDAF  PUSH IY
  EDB1  CALL FA21        ; a string felszabadítása a BASIC STACK-
                           böl
  EDB4  POP  HL
  EDB5  INC  HL
  EDB6  LD   A,(HL)      ; A-ba a string hossza
  EDB7  OR   A
  EDB8  JP   Z,FB14      ; üres string, hibajelzés
  EDBB  JP   ED10        ; a string elsö byte-jának kódja a BASIC
                           STACK-be

A PEEK rutin:
  EDBE  DW   ED17
  EDC0  DB   04
  EDC1  DB   'PEEK'
  EDC5  DB   0A
  EDC6  CALL EA68        ; az argumentum a BASIC STACK-be
  EDC9  CALL FD02        ; HL-be az argumentum
  EDCC  CALL FFE7        ; a ROM-ba mutat?
  EDCF  JR   C,03        ; nem, ugrás az EDD4H címre
  EDD1  DI               ; a maszkolható megszakítás tiltása
```

```
  EDD2  OUT   02,A        ; memórialapozás: U U V S lapkiosztás
  EDD4  LD    L,(HL)      ; L-be a keresett byte
  EDD5  LD    A,70
  EDD7  OUT   02,A        ; memórialapozás: U U U S lapkiosztás
  EDD9  EI                ; megszakítás engedélyezése
  EDDA  JP    ED12        ; a keresett byte a BASIC STACK-be

A PI-t megvalósító rutin:
  EDDD  DW    EDBE
  EDDF  DB    02
  EDE0  DB    'PI'
  EDE2  DB    0A
  EDE3  RST   18H         ; a 34. konstans a BASIC STACK-be
  EDE4  DB    85 22
  EDE6  RET

Az RND függvény rutinja:
  EDE7  DW    EDDD
  EDE9  DB    03
  EDEA  DB    'RND'
  EDED  DB    0A
  EDEE  CP    96
  EDF0  JR    Z,10        ; van változó megadva, ugrás az EE02H-ra
  EDF2  CALL  E6D8        ; az 1 és két byte-os véletlenszám puf-
                            ferbe a következő értékek
  EDF5  CALL  FA2B        ; HL a BASIC STACK-be
  EDF8  RST   18H         ; aritmetikai rutinok hívása
  EDF9  DB    05 26 00
  EDFC  DB    05 26 0C
  EDFF  DB    00 81
  EE01  RET

Az argumentumos RND rutin:
  EE02  CALL  EA68        ; az argumentum a BASIC STACK-be
  EE05  CALL  FAC3        ; HL-be az argumentum
  EE08  BIT   7,H
  EE0A  JP    NZ,FB14     ; hibajelzés, ha H felső bitje 1
  EE0D  LD    A,H
  EE0E  OR    L
  EE0F  JP    Z,FB14      ; hibajelzés, ha HL=00H
  EE12  PUSH  HL
  EE13  XOR   A
  EE14  INC   A
  EE15  ADD   HL,HL
  EE16  JR    NC,FC       ; ugrás az EE14H címre, ha NC
```

```
  EE18  DEC  A
  EE19  POP  HL
  EE1A  DEC  HL
  EE1B  PUSH HL
  EE1C  PUSH AF
  EE1D  CALL E6D8        ; az 1 és két byte-os véletlenszám puf-
                           ferbe a következö értékek
  EE20  POP  AF
  EE21  LD   B,A
  EE22  SRL  H
  EE24  RR   L
  EE26  DJNZ FA          ; ugrás az EE22H címre
  EE28  EX   DE,HL
  EE29  POP  HL
  EE2A  OR   A
  EE2B  SBC  HL,DE
  EE2D  ADD  HL,DE
  EE2E  JR   C,EB        ; ugrás az EE1BH címre, ha CARRY
  EE30  EX   DE,HL
  EE31  JP   FA2B        ; a véletlen egész HL-böl a BASIC STACK-
                           be

A SGN rutin:
  EE34  DW   EDE7
  EE36  DB   03
  EE37  DB   'SGN'
  EE3A  DB   0A
  EE3B  CALL EA68        ; az argumentum a BASIC STACK-be
  EE3E  LD   A,(IY+06H)  ; a legnagyobb helyiértékü számjegy A-ba
  EE41  OR   A
  EE42  RET  Z           ; 0 volt az argumentum
  EE43  LD   A,(IY+08H)  ; a kitevö A-ba
  EE46  PUSH AF
  EE47  CALL FA08        ; 1E+00 a BASIC STACK-be
  EE4A  POP  AF
  EE4B  AND  80
  EE4D  OR   (IY+08H)    ; a kitevö bit A-ba
  EE50  LD   (IY+08H),A  ; az elöjel bit beírása
  EE53  RET

A SIN rutin:
  EE54  DW   EE34
  EE56  DB   03
  EE57  DB   'SIN'
  EE5A  DB   0A
```

```
EE5B  CALL EA68        ; az argumentum a BASIC STACK-be
EE5E  RST  18H         ; aritmetikai rutinok hívása
EE5F  DB   05 22 05
EE62  DB   22 80       ; 2*PI
EE64  CALL ED8B        ; az argumentum az elsö periódusban
EE67  LD   A,(IY+08H)  ; A-ba a kitevö
EE6A  OR   A
EE6B  RRA
EE6C  PUSH AF
EE6D  RST  18H         ; 10. aritmetikai rutin
EE6E  DB   8A
EE6F  RST  18H         ; aritmetikai rutinok hívása
EE70  DB   06 85 23
EE73  POP  AF
EE74  PUSH AF
EE75  CALL M,F726      ; ha a redukált argumentum negatív, 4.
                         aritmetikai rutin hívása
EE78  CALL F693        ; összehasonlítás
EE7B  POP  BC
EE7C  PUSH BC
EE7D  JR   Z,1B        ; az összehasonlított számok egyenlök,
                         ugrás az EE9AH címre
EE7F  XOR  B
EE80  JP   M,EE93
EE83  RST  18H         ; aritmetikai rutinok hívása
EE84  DB   06 85 22
EE87  POP  AF
EE88  XOR  40
EE8A  PUSH AF
EE8B  CALL P,F726      ; ha P. a 4. aritmetikai rutin hívása
EE8E  RST  18H         ; a 0. és a 10. aritmetikai rutin hívása
EE8F  DB   00 8A
EE91  JR   DC          ; ugrás az EE6FH címre
EE93  POP  AF
EE94  XOR  80
EE96  PUSH AF
EE97  JP   M,EE6F
EE9A  RST  18H         ; aritmetikai rutinok hívása
EE9B  DB   06 0C 0C
EE9E  DB   0C 02 08
EEA1  DB   05 1D 02
EEA4  DB   05 1C 00
EEA7  DB   06 02 05
EEAA  DB   1B 00 06
EEAD  DB   02 05 1A
```

```
  EEB0  DB    00 06 02
  EEB3  DB    05 19 00
  EEB6  DB    06 02 02
  EEB9  DB    80
  EEBA  POP   AF
  EEBB  ADD   A,A
  EEBC  RET   P
  EEBD  JP    F726          ; ugrás a 4. aritmetikai rutinra, ha M

Az SQR rutin:
  EEC0  DW    EE54
  EEC2  DB    03
  EEC3  DB    'SQR'
  EEC6  DB    0A
  EEC7  CALL  EA68          ; az argumentum a BASIC STACK-be
  EECA  BIT   7,(IY+08H)
  EECE  JP    NZ,FB14       ; hibajelzés, ha negatív
  EED1  LD    A,(IY+06H)
  EED4  OR    A
  EED5  RET   Z             ; a rutin vége, ha 0
  EED6  CALL  EAC3          ; a 10. aritmetikai rutin
  EED9  LD    HL,19C6       ; HL-be az X aritmetikai regiszter kite-
                              vő byte-jának címe
  EEDC  LD    A,(HL)
  EEDD  SUB   3F
  EEDF  PUSH  AF
  EEE0  LD    (HL),3F
  EEE2  RST   18H           ; aritmetikai rutinok hívása
  EEE3  DB    06 05 1F
  EEE6  DB    02 05 1E
  EEE9  DB    00 8B
  EEEB  LD    B,04
  EEED  PUSH  BC
  EEEE  RST   18H           ; aritmetikai rutinok hívása
  EEEF  DB    07 06 07
  EEF2  DB    01 07 03
  EEF5  DB    05 00 02
  EEF8  DB    00 8B
  EEFA  POP   BC
  EEFB  DJNZ  F0            ; ugrás az EEEDH címre
  EEFD  POP   AF
  EEFE  BIT   0,A
  EF00  JR    Z,09          ; ugrás az EF0BH címre, ha Z
  EF02  PUSH  AF
  EF03  RST   18H           ; aritmetikai rutinok hívása
```

```
  EF04  DB    07 05 20
  EF07  DB    02 8B
  EF09  POP   AF
  EF0A  INC   A
  EF0B  SRL   A
  EF0D  LD    HL,19CD     ; HL-be az Y aritmetikai regiszter ki-
                            tevö byte-jának címe
  EF10  ADD   A,(HL)
  EF11  AND   7F          ; a kitevö bit törlése
  EF13  LD    (HL),A
  EF14  JP    EA9A        ; 7. aritmetikai rutin

A hatványozás (^) rutin:
  EF17  LD    A,(IY+06H)
  EF1A  OR    A
  EF1B  JR    NZ,06       ; a kitevö nem 0, ugrás az EF23H címre
  EF1D  CALL  FA1B        ; a BASIC STACK-böl a kitevö felszabadí-
                            tása
  EF20  JP    FA08        ; a hatványalap helyére 1E+00, mivel
                            minden szám 0. hatványa 1
  EF23  LD    A,(IY+0FH)  ; A-ba a hatványalap legnagyobb helyi-
                            értékü számjegye
  EF26  OR    A
  EF27  JP    Z,FA1B      ; a hatványalap 0, a 0 minden nem 0.
                            hatványa 0, ugrás
  EF2A  LD    A,(IY+08H)  ; A-ba a kitevö kitevöje
  EF2D  AND   7F          ; az elöjel bit törlése
  EF2F  CP    43
  EF31  JR    NC,4A       ; ha a karakterisztika >2, ugrás az
                            EF7DH címre
  EF33  CALL  FA92        ; 12. aritmetikai rutin
  EF36  CALL  FAC3        ; HL-be a kitevö egészrésze
  EF39  PUSH  HL
  EF3A  CALL  FA92        ; a 12. aritmetikai rutin
  EF3D  CALL  F767        ; a túlcsordulás byte törlése kerekítés-
                            sel
  EF40  POP   HL
  EF41  PUSH  HL
  EF42  CALL  FA2B        ; HL a BASIC STACK-be
  EF45  CALL  F693        ; összehasonlítás
  EF48  POP   HL
  EF49  JR    NZ,32       ; a két összehasonlított szám nem egyen-
                            lö, ugrás az EF7DH címre
  EF4B  BIT   7,H
  EF4D  PUSH  AF
```

```
  EF4E  CALL NZ,FC86     ; ha H felsö bitje 1, ugrás
  EF51  PUSH HL
  EF52  RST  18H         ; aritmetikai rutinok  hívása
  EF53  DB   0A 0A 85
  EF56  DB   01
  EF57  POP  HL
  EF58  PUSH HL
  EF59  BIT  0,L
  EF5B  JR   Z,06        ; ugrás az EF63H címre, ha Z
  EF5D  CALL EA9F        ; a 6. aritmetikai rutin
  EF60  CALL F512        ; a 2. aritmetikai rutin
  EF63  POP  HL
  EF64  SRL  H
  EF66  RR   L
  EF68  LD   A,H
  EF69  OR   L
  EF6A  JR   Z,08        ; ugrás az EF74H címre, ha HL/2=0
  EF6C  PUSH HL
  EF6D  RST  18H         ; aritmetikai rutinok hívása
  EF6E  DB   06 06 02
  EF71  DB   8A
  EF72  JR   E3          ; ugrás az EF57H címre
  EF74  POP  AF
  EF75  RET  Z
  EF76  RST  18H         ; aritmetikai rutinok hívása
  EF77  DB   0B 05 01
  EF7A  DB   07 81
  EF7C  RET
  EF7D  CALL EABE        ; 11. aritmetikai rutin
  EF80  CALL ED21        ; LOG rutin
  EF83  CALL EA9A        ; 7. aritmetikai rutin
  EF86  CALL F512        ; 2. aritmetikai rutin
  EF89  JP   EBB8        ; EXP rutin

A STR$ rutin:
  EF8C  DW   EEC0
  EF8E  DB   04
  EF8F  DB   'STR$'
  EF93  DB   08
  EF94  CALL EA68        ; az argumentum a BASIC STACK-be
  EF97  CALL F80E        ; az argumentumból képzett string az
                           1931H címtöl
  EF9A  CALL FA1B        ; az argumentum felszabadítás a BASIC
                           STACK-ból
  EF9D  JP   FA7D        ; az argumentum string képe a BASIC
                           STACK-be
```

```
A STRING$ függvény rutinja:
  EFA0  DW   EF8C
  EFA2  DB   07
  EFA3  DB   'STRING$'
  EFAA  DB   08
  EFAB  LD   A,96
  EFAD  CALL FD54          ; ha nem "(" következik, szintaktikai
                             hiba
  EFB0  CALL FB1B          ; A-ba az elsö változó
  EFB3  PUSH AF
  EFB4  LD   A,A4
  EFB6  CALL FD54          ; ha nem "," következik, szintaktikai
                             hiba
  EFB9  CP   02
  EFBB  JR   NC,13         ; a második változó szám, ugrás az EFD0H
                             címre
  EFBD  CALL F294          ; a string változó a BASIC STACK-be
  EFC0  LD   A,(IY+01H)    ; A-ba a string hossza
  EFC3  OR   A
  EFC4  JR   Z,07          ; ha üres string, ugrás az EF7DH címre
  EFC6  LD   A,(IY+02H)    ; A-ba a string elsö karaktere
  EFC9  CALL FA21          ; a string felszabadítása a BASIC STACK-
                             böl
  EFCC  LD   BC,F5C1
***
  EFCD  POP  BC
  EFCE  PUSH AF
***
  EFCF  SCF
  EFD0  CALL NC,FB1B       ; ha a 2. változó szám, a 2. változó az
                             A-ba
  EFD3  PUSH AF
  EFD4  CALL EA70          ; ha a záró zárójel hiányzik, hibajelzés
  EFD7  POP  DE            ; D-be a láncolandó karakter kódja
  EFD8  PUSH IY
  EFDA  POP  HL
  EFDB  DEC  HL
  EFDC  POP  AF            ; A-ba a láncszemek száma
  EFDD  INC  A
  EFDE  JP   Z,F912        ; túl hosszú, ugrás a hibajelzésre
  EFE1  DEC  A
  EFE2  JR   Z,05          ; üres string, ugrás az EFE9H címre
A karakterlánc a BASIC STACK-be:
  EFE4  LD   B,A
  EFE5  LD   (HL),D        ; láncszem a BASIC STACK-be
```

```
  EFE6  DEC  HL
  EFE7  DJNZ FC          ; még nem a vége, ugrás az EFE5H címre
  EFE9  LD   (HL),A      ; a string hossza
  EFEA  DEC  HL
  EFEB  LD   (HL),01     ; a string jel byte
  EFED  PUSH HL
  EFEE  POP  IY
  EFF0  RET

A TAN rutin:
  EFF1  DW   EFA0
  EFF3  DB   03
  EFF4  DB   'TAN'
  EFF7  DB   0A
  EFF8  CALL EA68        ; az argumentum a BASIC STACK-be
  EFFB  CALL EACD        ; 9. aritmetikai rutin
  EFFE  CALL EE5E        ; a SIN rutin
  F001  RST  18H         ; a BASIC STACK-be az argumentum + PI/2
  F002  DB   07 05 23
  F005  DB   80
  F006  CALL EE5E        ; a SIN rutin
  F009  JP   F5FB        ; az 1. aritmetikai rutin

A USR függvény rutinja:
  F00C  DW   EFF1
  F00E  DB   03
  F00F  DB   'USR'
  F012  DB   0A
  F013  LD   A,96
  F015  CALL FD54        ; ha nem "(" következik, hibajelzés
  F018  CALL F0A7        ; az argumenrum a BASIC STACK-be
  F01B  CP   95
  F01D  JR   Z,0A        ; ha nincs több argumentum, ugrás az
                           F029H címre
  F01F  LD   A,A4
  F021  CALL FD54        ; ha nem "," következik, hibajelzés
  F024  CALL FAC4        ; a második paraméter a HL-be
  F027  LD   A,95
  F029  CALL FD54        ; ha nem ")" következik, hibajelzés
  F02C  LD   DE,FA2B     ; a user rutin végrehajtása utáni cím
  F02F  PUSH DE
  F030  PUSH HL
  F031  CALL FD02        ; HL-be a user rutin címe
  F034  EX   SP,HL       ; a STACK-be a user rutin címe
  F035  RET              ; ugrás a user rutinra
```

A VAL függvény rutinja:

```
F036  DW   F00C
F038  DB   03
F039  DB   'VAL'
F03C  DB   0A
F03D  CALL EA75        ; az argumentum string a BASIC STACK-be
F040  PUSH IY
F042  POP  HL
F043  INC  HL
F044  PUSH HL
F045  LD   C,(HL)      ; C-be a string hossza
F046  LD   B,00
F048  ADD  HL,BC
F049  INC  HL
F04A  LD   A,(HL)      ; az elözö bejegyzés elsö byte-jának el-
                         mentése
F04B  LD   (HL),B      ; a string végét 00H jelzi
F04C  EX   SP,HL       ; HL-be a string hosszának címe
F04D  PUSH AF
F04E  INC  HL          ; HL-be a string elsö byte-ának a címe
F04F  CALL F914        ; a stringböl szám a BASIC STACK-be
F052  CALL EC7F
F055  POP  AF
F056  POP  HL
F057  LD   (HL),A      ; a string elötti bejegyzés elsö byte-
                         jának visszaállítása
F058  DEC  HL
F059  EX   DE,HL
F05A  PUSH IY
F05C  POP  HL
F05D  LD   BC,0009
F060  ADD  HL,BC
F061  DEC  HL
F062  LDDR             ; a strigböl lett szám mozgatása a string
                         helyére
F064  INC  DE
F065  PUSH DE
F066  POP  IY
F068  RET              ; a VAL rutin vége
```

A VARPTR rutin:

```
F069  DW   F036
F06B  DB   06
F06C  DB   'VARPTR'
F072  DB   0A
```

```
  F073  LD    A,96
  F075  CALL  FD54          ; ha nem "(" következik, hibajelzés, kü-
                              lönben HL-be a változó címe
  F078  AND   FD
  F07A  CP    01
  F07C  JP    NZ,FB14       ; a változó tipusa nem jó, hibajelzés
  F07F  EXX
  F080  LD    A,C
  F081  AND   7F
  F083  LD    (1708),A      ; változó tipusbyte-jának eltárolása
  F086  AND   0C
  F088  JR    NZ,F2         ; nem megfelelö változó, ugrás az F07CH
                              címre
  F08A  EXX
  F08B  CALL  F42E          ; tömb elem?
  F08E  CALL  FA2B          ; HL értéke a BASIC STACK-be számként
  F091  JP    EA70          ; ha a ")" hiányzik, hibajelzés

A VERNUM rutin:
  F094  DW    F069
  F096  DB    06
  F097  DB    'VERNUM'
  F09D  DB    0A
  F09E  LD    HL,000C       ; HL-be 12
  F0A1  JP    FA2B          ; HL a BASIC STACK-be, a rutin vége

Egy tetszöleges kifejezés kiértékelése:
  F0A4  CALL  FC43          ; a következö program byte
  F0A7  EXX
  F0A8  LD    A,B           ; A-ba a végrehajtandó müvelet tokenje
  F0A9  EXX
  F0AA  CALL  F0D7          ; a kifejezés rész kiértékelése
  F0AD  EXX
  F0AE  LD    A,B           ; A-ba a müvelet tokenje
  F0AF  EXX
  F0B0  CP    BF            ; (OR)
  F0B2  JR    Z,04          ; ha OR, ugrás az F0B8H címre
  F0B4  CP    B2            ; (XOR)
  F0B6  RET   NZ            ; a kiértékelö rutin vége, ha NZ

Az XOR müvelet rutinja:
  F0B7  SCF

Az OR müvelet rutinja:
  F0B8  PUSH  AF
```

```
F0B9  CALL FC43        ; a következő utasítás byte
F0BC  CALL F0D7        ; az OR/XOR utáni kifejezés kiértékelése
F0BF  CALL FABF        ; a művelet két tényezője HL-be és DE-be
F0C2  POP  AF
F0C3  JR   C,07        ; ha XOR, ugrás az F0CCH címre
F0C5  LD   A,L
F0C6  OR   E
F0C7  LD   L,A         ; L=L OR E
F0C8  LD   A,H
F0C9  OR   D           ; A=H OR D
F0CA  JR   05          ; ugrás az F0D1H címre
F0CC  LD   A,L
F0CD  XOR  E
F0CE  LD   L,A         ; L=L XOR E
F0CF  LD   A,H
F0D0  OR   D           ; A=H OR D (!!!)
F0D1  LD   H,A
F0D2  CALL FA2B        ; HL a BASIC STACK-be számként
F0D5  JR   D6          ; ugrás az F0ADH címre

F0D7  CALL F0F4        ; kiértékelés
F0DA  EXX
F0DB  LD   A,B         ; A-ba a művelet tokenje
F0DC  EXX
F0DD  CP   C9          ; (AND)
F0DF  RET  NZ          ; nem AND, RET

Az AND művelet rutinja:
F0E0  CALL FC43        ; a következő program byte
F0E3  CALL F0F4        ; a magasabb prioritású műveletek kiér-
                         tékelése
F0E6  CALL FABF        ; az AND művelet két változója HL-be és
                         DE-be
F0E9  LD   A,L
F0EA  AND  E
F0EB  LD   L,A         ; L=L AND E
F0EC  LD   A,H
F0ED  AND  D
F0EE  LD   H,A         ; H=H AND D
F0EF  CALL FA2B        ; HL a BASIC STACK-be számként
F0F2  JR   E6          ; ugrás az F0DAH címre


F0F4  CP   C2          ; (NOT)
F0F6  JR   NZ,12       ; nem NOT, ugrás az F10AH címre
```

```
A NOT müvelet rutinja:
  F0F8  CALL FC43        ; a következö program byte
  F0FB  CALL F0F4        ; a magasabb prioritású müveletek elvég-
                           zése
  F0FE  CALL FAC3        ; HL-be a müvelet tényezöje
  F101  LD   A,L
  F102  CPL
  F103  LD   L,A         ; L= NOT L
  F104  LD   A,H
  F105  CPL
  F106  LD   H,A         ; H= NOT H
  F107  JP   FA2B        ; HL a BASIC STACK-be számként

  F10A  CP   02
  F10C  JR   C,15        ; ugrás az F123H címre, ha string kife-
                           jezések
  F10E  CALL F155        ; kiértékelés
  F111  CP   99
  F113  RET  C
  F114  CP   9F
  F116  RET  NC

Az =,<>,>,<,<=,>= müveletek rutinja számokra:
  F117  PUSH AF
  F118  CALL FC43        ; a következö program byte
  F11B  CALL F155        ; a magasabb prioritású müveletek elvég-
                           zése
  F11E  CALL F693        ; az összehasonlítások kiértékelése
  F121  JR   17          ; ugrás az F13AH címre

A string kifejezések kiértékelése:
  F123  CALL F294        ; kiértékelés
  F126  CP   99
  F128  JP   C,F35D      ; rossz müveleti jel, hibajelzés
  F12B  CP   9F
  F12D  JP   NC,F35D     ; rossz müveleti jel, hibajelzés

Az =,<>,>,<,<=,>= müveletek rutinja stringekre:
  F130  PUSH AF
  F131  CALL FC43        ; a következö program byte
  F134  CALL F294        ; a magasabb prioritású string müveletek
                           elvégzése
  F137  CALL F6D7        ; két string összehasonlítása
  F13A  POP  HL
  F13B  LD   A,H
```

```
F13C  CALL F142       ; az összehasonlítás eredményének érté-
                        kelése
F13F  JP   FA2B       ;. HL értéke a BASIC STACK-be

F142  LD   HL,FFFF
F145  RRCA
F146  JR   NC,01      ; a müveleti jel nem "<" jellel kezdödik,
                        ugrás az F149H címre
F148  RET  M          ; az összehasonlítás eredménye: kisebb
F149  RRCA
F14A  JR   NC,01      ; a müveleti jel "<" jellel kezdödik
F14C  RET  Z          ; az összehasonlítás eredménye: e-
                        gyenlö
F14D  RRCA
F14E  JR   NC,03      ; a müveleti jel nem a "<",ugrás az
                        F153H címre
F150  JR   Z,01       ; ha az összehasonlitás eredménye: egyen-
                        lö, ugrás az F153H címre
F152  RET  P          ; az összehasonlítás eredménye: nem
                        kisebb
F153  INC  HL         ; az müvelet eredmény: hamis
F154  RET

Aritmetikai kifejezés kiértékelése:
Az összeadás és a kivonás:
F155  CALL F16C       ; "+" és "-" kiértékelése
F158  CALL NZ,F181    ; nem "+", "-" jel volt, kiértékelés
F15B  INC  E
F15C  CALL NZ,F726    ; negatív elöjel, a 4. aritmetikai rutin
F15F  CALL F16C       ; "+" és "-" kiértékelése
F162  RET  NZ
F163  INC  E
F164  CALL Z,F493     ; összeadás, a 0. aritmetikai rutin
F167  CALL NZ,F48E    ; kivonás, a 3. aritmetikai rutin
F16A  JR   F3         ; ugrás az F15FH címre

F16C  EXX
F16D  LD   A,B
F16E  EXX
F16F  LD   E,FF       ; E-be az elöjel flag: FFH=pozitív
F171  CP   98         ; (+)
F173  JR   Z,04       ; ha "+", ugrás az F179H címre
F175  CP   A2         ; (-)
F177  RET  NZ
F178  INC  E          ; E=00H, negatív
```

```
  F179  CALL FC43        ; a következö program byte
  F17C  CALL F181        ; a magasabb prioritású müveletek kiér-
                           tékelése
  F17F  XOR  A           ; A-ba 0
  F180  RET

A szorzás és az osztás:
  F181  CALL F1A2        ; kiértékelés
  F184  EXX
  F185  LD   A,B
  F186  EXX
  F187  CP   A8          ; (*)
  F189  JR   Z,03        ; szorzás, ugrás az F18EH címre
  F18B  CP   A1          ; (/)
  F18D  RET  NZ
  F18E  LD   D,A         ; a müveleti jel D-be
  F18F  CALL FC43        ; a következö program byte
  F192  CALL F1A2        ; a magasabb prioritású müveletek elvég-
                           zése
  F195  PUSH DE
  F196  LD   A,D         ; A-ba a müveleti jel
  F197  CP   A8
  F199  CALL Z,F512      ; szorzás, a 2. aritmetikai rutin
  F19C  CALL NZ,F5FB     ; osztás, az 1. aritmetikai rutin
  F19F  POP  DE
  F1A0  JR   E2          ; ugrás az F184H címre

A hatványozás:
  F1A2  PUSH DE
  F1A3  CALL F1BB        ; a magasabb prioritású müveletek kiér-
                           tékelése
  F1A6  POP  DE
  F1A7  EXX
  F1A8  LD   A,B
  F1A9  EXX
  F1AA  CP   9F          ; (^)
  F1AC  RET  NZ
  F1AD  PUSH DE
  F1AE  CALL FC43        ; a következö program byte
  F1B1  CALL F1BB        ; a magasabb prioritású müveletek kiér-
                           tékelése
  F1B4  CALL EF17        ; a hatványozó rutin
  F1B7  JR   ED          ; ugrás az F1A6H címre

  F1B9  RST  08H
```

```
F1BA  DB   03          ; "Argument missing" hibajelzés

F1BB  PUSH AF
F1BC  CALL FC8E        ; van elég szabad memória?
F1BF  POP  AF
F1C0  CP   C0          ; (ORD)
F1C2  JP   Z,EDA9      ; az ORD függvény, ugrás
F1C5  CP   B1          ; (ATN)
F1C7  JP   Z,EAF1      ; az ATN függvény, ugrás
F1CA  CP   02
F1CC  JP   C,F35D      ; string változó, hibajelzés
F1CF  JP   Z,FC43      ; numerikus, következő program byte
F1D2  CP   FD
F1D4  JR   NC,E3       ; utasítás vége, ugrás a hibajelzésre
F1D6  CP   96          ; "("
F1D8  JR   NZ,08       ; nem "(", ugrás az F1E2H címre
F1DA  CALL F0A4        ; a zárójelen belüli kifejezés kiértéke-
                         lése
F1DD  CP   95          ; ")"
F1DF  JP   Z,FC43      ; ha ")", következő program byte


F1E2  CP   03
F1E4  JP   NZ,FD5A     ; nem numerikus tömb elem, hibajelzés
F1E7  CALL F42E        ; a tömb elemének címe HL-be
F1EA  BIT  3,C
F1EC  JP   NZ,DBAD     ; beépített függvény, JP(HL)
F1EF  BIT  2,C
F1F1  JP   Z,FA63      ; ha numerikus, a változó értéke a BASIC
                         STACK-be

Felhasználó által definiált függvény kiértékelése:
F1F4  LD   D,(IX+01H) ; aktuális változó tipusbyte-ja D-be
F1F7  PUSH DE
F1F8  PUSH HL          ; HL a változó tipusbyte-ja után mutat
F1F9  SUB  96
F1FB  SUB  01
F1FD  SBC  A,A
F1FE  PUSH AF
F1FF  LD   A,C
F200  JR   Z,14        ; ugrás az F216H címre, ha A>96H (nincs
                         argumentum)
F202  EX   AF,AF'
F203  CALL FC43        ; a következő program byte
F206  OR   A
```

```
F207  JP   M,FD5A      ; hibajelzés, ha A felsö bitje 1
F20A  LD   (IX+01H),A
F20D  CALL F28D        ; változó eltárolása a BASIC STACK-be
F210  LD   A,95
F212  CALL FD54        ; hibajelzés, ha nem ")" következik
F215  EX   AF,AF'
F216  EX   AF,AF'
F217  POP  AF
F218  LD   HL,(1726)   ; HL-be TOP
F21B  EX   SP,HL
F21C  PUSH HL          ; TOP a visszatérési cím fölé
F21D  LD   HL,(1724)   ; HL-be CHAIN
F220  EX   SP,HL
F221  EXX
F222  PUSH HL
F223  LD   HL,(170C)   ; HL-be a START
F226  PUSH HL
F227  PUSH BC
F228  EXX
F229  LD   E,(HL)
F22A  INC  HL
F22B  LD   D,(HL)      ; DE-be a DEF utasítást tartalmazó prog-
                         ramsor címe
F22C  INC  HL
F22D  LD   (170C),DE   ; az új START cím eltárolása
F231  LD   E,(HL)
F232  INC  HL
F233  LD   D,(HL)      ; DE-be a felhasználói függvény neve u-
                         táni elsö karakter címe
F234  EX   DE,HL
F235  EXX
A DEF függvény definíció kiértékelése:
F236  LD   C,A
F237  CALL FC43        ; a következö program byte
F23A  SUB  96
F23C  SUB  01
F23E  SBC  A,A
F23F  XOR  C
F240  JP   NZ,FD5A     ; ha NZ, szintaktikai hiba
F243  XOR  C
F244  JR   Z,27        ; ugrás az F26DH címre, ha nincs argu-
                         mentum
F246  CALL FC43        ; a következö program byte
F249  AND  81
F24B  CP   01
```

```
F24D  JP   NZ,FD5A     ; ha tömb változó, hibajelzés
F250  EXX
F251  BIT  7,C
F253  CALL Z,F40B      ; új változó bejegyzése a változó terü-
                         letre
F256  LD   A,C
F257  XOR  (IX+01H)
F25A  AND  02
F25C  JP   NZ,F35D     ; nem megfelelö változó tipus, hiba
F25F  PUSH DE
F260  EXX
F261  POP  HL
F262  CALL FB3B        ; a meghíváskori változó mozgatása a
                         BASIC.STACK-böl a definíció beli vál-
                         tozó bejegyzésébe
F265  CALL FC43        ; a következö program byte
F268  LD   A,95
F26A  CALL FD54        ; ha nem ")" következik, hibajelzés
F26D  LD   A,9A
F26F  CALL FD54        ; ha nem "=" következik, hibajelzés
F272  EX   AF,AF'
F273  LD   (IX+01H),A  ; az argumentum tipusbyte-jának eltáro-
                         lása
F276  CALL F28D        ; a DEF függvény értéke a BASIC STACK-be
F279  POP  BC
F27A  POP  HL
F27B  LD   (170C),HL   ; a START régi értékének visszatöltése
F27E  POP  HL
F27F  EXX
F280  POP  HL
F281  LD   (1724),HL   ; CHAIN régi értéke
F284  POP  HL
F285  LD   (1726),HL   ; a TOP régi értéke
F288  POP  AF
F289  LD   (1701),A
F28C  RET

Egy értékadó kifejezés kiértékelése:
F28D  BIT  1,(IX+01H)
F291  JP   NZ,F0A7     ; numerikus változó értékadása, ugrás

String kifejezés kiértékelése:
F294  EXX
F295  LD   A,B
F296  EXX
```

```
  F297  CALL F2E0       ; a magasabb prioritású müveletek kiér-
                          tékelése
  F29A  EXX
  F29B  LD   A,B
  F29C  EXX
  F29D  CP   97         ; (&)
  F29F  RET  NZ

Az & (string összegzés) művelet rutinja:
  F2A0  CALL FC43       ; a következö program byte
  F2A3  CALL F2E0       ; a magasabb prioritású müveletek kiér-
                          tékelése
  F2A6  PUSH IY
  F2A8  POP  HL
  F2A9  INC  HL
  F2AA  LD   E,L
  F2AB  LD   D,H        ; DE-be a második tényezö címe
  F2AC  LD   C,(HL)
  F2AD  LD   B,00       ; BC-be a string hossza
  F2AF  ADD  HL,BC
  F2B0  INC  HL         ; HL-be az elsö tényezö címe
  F2B1  PUSH HL
  F2B2  POP  IY
  F2B4  INC  HL
  F2B5  LD   C,(HL)     ; C-be a string hossza
  F2B6  CALL FC8E       ; van elég szabad memória
  F2B9  PUSH HL
  F2BA  ADD  HL,BC      ; HL-be az elsö tényezö végének címe
  F2BB  LD   A,C
  F2BC  OR   A
  F2BD  LD   A,(DE)     ; A-ba a 2. tényezö hossza
  F2BE  JR   Z,02       ; ugrás az F2C2H címre, ha az elsö té-
                          nyezö üres string
  F2C0  LDDR            ; az elsö tényezö mozgatása
  F2C2  POP  HL
  F2C3  ADD  A,(HL)     ; A-ba a két string együttes hossza
  F2C4  LD   (DE),A     ; az eredmény string hossza
  F2C5  JP   C,F912     ; túl hosszú, hibajelzés
  F2C8  CP   FF
  F2CA  JP   Z,F912     ; túl hosszú, hibajelzés
  F2CD  LD   C,(HL)     ; C-be az elsö string hossza
  F2CE  LD   E,L
  F2CF  LD   D,H
  F2D0  DEC  DE
  F2D1  DEC  DE         ; DE-be a második tényezö végének címe
```

```
  F2D2  ADD  HL,BC       ; HL-be az elsö tényezö végének címe
  F2D3  EX   DE,HL
  F2D4  LD   C,A         ; C-be az eredmény string hossza
  F2D5  INC  BC
  F2D6  LDDR             ; az eredmény string feltolása a BASIC
                           STACK-ben
  F2D8  EX   DE,HL
  F2D9  LD   (HL),01     ; a string bejegyzést jelzö byte
  F2DB  PUSH HL
  F2DC  POP  IY
  F2DE  JR   BA          ; ugrás az F29AH címre

String szeletelés rutin:
  F2E0  CALL F32D        ; a magasabb prioritású müveletek elvég-
                           zése
  F2E3  EXX
  F2E4  LD   A,B
  F2E5  EXX
  F2E6  CP   96          ; "("
  F2E8  RET  NZ
  F2E9  CALL FD27        ; E-be a szelet kezdete, D-be a szelet
                           vége
  F2EC  PUSH IY
  F2EE  POP  HL
  F2EF  INC  HL
  F2F0  LD   A,(HL)      ; A-ba a string hossza
  F2F1  LD   C,A
  F2F2  LD   B,00
  F2F4  CP   D
  F2F5  JR   C,01        ; a string rövidebb a szelet végénél,
                           ugrás az F2F8H címre
  F2F7  LD   A,D
  F2F8  LD   D,A
  F2F9  OR   A
  F2FA  JR   Z,0D        ; a string üres, ugrás az F308H címre
  F2FC  LD   A,C         ; A-ba a string hossza
  F2FD  CP   E
  F2FE  JR   C,09        ; a string rövidebb a szelet kezdeténél,
                           ugrás az F309H címre
  F300  INC  E
  F301  DEC  E
  F302  JR   NZ,01       ; E nem 0, ugrás az F305H címre
  F304  INC  E
  F305  LD   A,D
  F306  SUB  E
```

```
F307  JR   NC,04      ; a szelet vége nagyobb, mint a kezdete,
                        ugrás az F30DH címre
F309  ADD  HL,BC      ; HL-be a string végének címe
F30A  XOR  A          ; A-ba 0
F30B  JR   0B         ; ugrás az F318H címre
F30D  INC  A
F30E  PUSH HL
F30F  ADD  HL,BC      ; HL-be a string végének címe
F310  EX   DE,HL
F311  LD   C,H        ; C-be a szelet vége
F312  POP  HL
F313  ADD  HL,BC      ; HL-be a szelet végének címe
F314  LD   C,A        ; C-be a szelet "vastagsága"
F315  LDDR            ; a szelet feltolása a BASIC STACK-ben
F317  EX   DE,HL
F318  LD   (HL),A     ; az eredmény string hossza
F319  DEC  HL
F31A  LD   (HL),01    ; stringhez tartozó jelző byte
F31C  PUSH HL
F31D  POP  IY
F31F  JR   C2         ; ugrás az F2E3H címre

F321  CALL FAC4       ; kifejezés értéke HL-be
F324  LD   D,00
F326  INC  H
F327  RET  Z
F328  DEC  D
F329  DEC  H
F32A  RET  NZ
F32B  LD   D,L
F32C  RET

String függvények kiértékelése:
F32D  CALL FC8E       ; van elég szabad memória?
F330  EXX
F331  PUSH BC
F332  EXX
F333  POP  BC
F334  LD   A,B
F335  CP   01
F337  JP   C,FC43     ; sting, a következő program byte
F33A  JR   Z,0B       ; tömb elem, ugrás az F347H címre
F33C  CP   04
F33E  JR   C,1D       ; numerikus változó, ugrás az F35DH cím-
                        re, hiba
```

```
F340  CP   FD
F342  JP   C,FD5A      ; utasítás vége, szintaktikai hiba

F345  RST  08H
F346  DB   03          ; "Argument missing" hibajelzés

F347  LD   A,C
F348  CP   C5          ; (INKEY$)
F34A  JP   Z,EC89      ; INKEY$ függvény, ugrás
F34D  CALL F42E        ; HL-be a tömb elem címe
F350  BIT  3,C
F352  JP   NZ,DBAD     ; beépített függvény, JP(HL)
F355  BIT  2,C
F357  JP   NZ,F1F4     ; felhasználói (DEF) függvény, ugrás
F35A  JP   FA7C        ; string változó értéke a BASIC STACK-be

F35D  RST  08H
F35E  DB   0E          ; "Type mismatch" hibajelzés

Egy új változónév bejegyzése a változó területre:
F35F  CALL FC8E        ; van elég szabad hely?
F362  LD   DE,(1728)
F366  LD   HL,(1726)   ; HL-be TOP
F369  LD   (1724),HL   ; CHAIN új értéke
F36C  LD   (HL),E
F36D  INC  HL
F36E  LD   (HL),D      ; az új bejegyzés első 2 byte-ja
F36F  INC  HL
F370  LD   C,00
F372  PUSH HL
F373  LD   DE,(1728)
F377  LD   A,(DE)
F378  CP   FD
F37A  JR   NC,27       ; vége az utasításnak, ugrás az F3A3H
                         címre
F37C  CP   20
F37E  JR   C,1C        ; vezérlö karakter, ugrás az F39CH címre
F380  CP   24
F382  JR   Z,19        ; ha "$" karakter, ugrás az F39DH címre
F384  CP   2E
F386  JR   Z,14        ; ha "." karakter, ugrás az F39CH címre
F388  CP   30
F38A  JR   C,17        ; ha a karakter kód < 48, ugrás az F3A3H
                         címre
F38C  CP   3A
```

```
  F38E  JR   C,0C        ; számjegy karakter, ugrás az F39CH cím-
                           re
  F390  CP   3F
  F392  JR   C,0F        ; ha karakter kód < 63, ugrás az F3A3H
                           címre
  F394  CP   60
  F396  JR   C,04        ; ha karakter kód < 96, ugrás az F39CH
                           címre
  F398  CP   A9
  F39A  JR   C,07        ; ha a karakter kód < 169, ugrás az
                           F3A3H címre
  F39C  SCF
  F39D  INC  DE
  F39E  INC  C
  F39F  INC  HL
  F3A0  LD   (HL),A      ; a változónév karakter a szimbólumtábla
                           bejegyzésbe
  F3A1  JR   C,D4        ; még nincs vége a változó névnek, ugr-
                           rás az F377H címre
  F3A3  LD   A,(HL)      ; A-ba a változónév utolsó karaktere
  F3A4  INC  HL
  F3A5  LD   (1726),HL   ; az új TOP
  F3A8  EX   SP,HL
  F3A9  LD   (HL),C      ; a változónév hossza
  F3AA  POP  HL
  F3AB  RET

Numerikus változó kezdő értéke:
  F3AC  CALL FC8E        ; van elég szabad memória?
  F3AF  LD   HL,(1726)   ; HL-be TOP
  F3B2  LD   B,06        ; a szám 6 byte-ot foglal el
  F3B4  XOR  A           ; A-ba 0
  F3B5  LD   (HL),A      ; a számjegy 0
  F3B6  INC  HL
  F3B7  DJNZ FC          ; ugrás az F3B5H címre
  F3B9  JR   13          ; ugrás az F3CEH címre

Változó kezdő értéke:
  F3BB  BIT  1,(IX+01H)
  F3BF  JR   NZ,EB       ; numerikus változó, ugrás az F3ACH cím-
                           re

String változó kezdő értéke:
  F3C1  CALL FC8E        ; van elég szabad memória?
  F3C4  LD   HL,(1726)   ; HL-be TOP
```

```
  F3C7  LD   (HL),C      ; a string változó maximális hossza
  F3C8  INC  HL
  F3C9  LD   B,00
  F3CB  LD   (HL),B      ; a string aktuális hossza: 0
  F3CC  ADD  HL,BC       ; HL-be a string vége
  F3CD  INC  HL
  F3CE  LD   (1726),HL   ; TOP új értékének elmentése
  F3D1  RET

Adott változónév megkeresése a szimbólumtáblában:
  F3D2  LD   (1728),HL   ; HL eltárolása
  F3D5  LD   HL,1725
  F3D8  LD   A,E1
***
  F3D9  POP  HL
***
  F3DA  LD   D,(HL)
  F3DB  DEC  HL
  F3DC  LD   E,(HL)      ; DE-be a szimbólumtábla következő ele-
                           mének címe
  F3DD  LD   A,E
  F3DE  OR   D
  F3DF  LD   HL,(1728)   ; HL visszaállítása
  F3E2  RET  Z           ; nincs több szimbólumtábla elem, RET
  F3E3  EX   DE,HL
  F3E4  INC  HL
  F3E5  PUSH HL
  F3E6  INC  HL
  F3E7  LD   B,(HL)      ; B-be az adott szimbólumtábla elem ne-
                           vének hossza
  F3E8  LD   A,(DE)      ; A-ba az adott változónév aktuális
                           byte
  F3E9  INC  DE
  F3EA  INC  HL
  F3EB  CP   (HL)
  F3EC  JR   NZ,EB       ; ha a két összehasonlított byte külön-
                           bözik, ugrás az F3D9H címre
  F3EE  DJNZ F8          ; ugrás az F3E8H címre
  F3F0  CP   24
  F3F2  JR   Z,10        ; ha "$" a változónév utolsó karaktere,
                           ugrás az F404H címre
  F3F4  LD   A,(DE)      ; A-ba a változónevet követő byte
  F3F5  CP   20
  F3F7  JR   Z,0B        ; SPACE, ugrás az F404H címre
  F3F9  CP   FD
```

```
  F3FB  JR   NC,07        ; utasítás vége, ugrás az F404H címre
  F3FD  CP   A9
  F3FF  JR   NC,D8        ; utasítás token, ugrás az F3D9H címre
  F401  RLA
  F402  JR   NC,D5        ; nem token, ugrás az az F3D9H címre
  F404  POP  AF
  F405  INC  HL
  F406  RES  7,(HL)       ; a tipusbyte felsö bitjének törlése
  F408  SCF
  F409  JR   1D           ; ugrás az F428H címre

Egy új változó bejegyzése a szimbólumtáblán:
  F40B  CALL F35F         ; a változónév bejegyzése
  F40E  SUB  24           ; ($)
  F410  JR   Z,02         ; string változó, ugrás az F414H címre
  F412  LD   A,02         ; numerikus változó tipusbyte-ja
  F414  LD   (HL),A       ; változó tipusbyte-jának beírása
  F415  PUSH HL
  F416  INC  HL
  F417  LD   (1726),HL    ; TOP új értéke
  F41A  JR   NZ,06        ; numerikus változó, ugrás az F422H cím-
                            re
  F41C  LD   C,12         ; C-be 18
  F41E  CALL F3C1         ; kezdö érték a stringnek: üres string
  F421  XOR  A
  F422  CALL NZ,F3AC      ; numerikus változó kezdö értéke: 0
  F425  POP  HL
  F426  SET  7,(HL)       ; a tipusbyte felsö bitje 1
  F428  LD   C,(HL)       ; C-be a tipusbyte
  F429  INC  HL
  F42A  EX   DE,HL
  F42B  RET

  F42C  RST  08H
  F42D  DB   05           ; "Bad subscript" hibajelzés

HL-be a változó adatbyte-jai kezdetének a címe:
  F42E  EXX
  F42F  PUSH DE
  F430  BIT  0,C
  F432  LD   A,C
  F433  EXX
  F434  POP  HL
  F435  LD   C,A
  F436  JP   Z,FC43       ; a változó nem tömb, a következö prog-
```

```
                                ram byte
F439  LD    A,96
F43B  CALL  FD51          ; ha nem "(" következik, hibajelzés
F43E  XOR   A             ; A-ba 0
F43F  LD    E,A
F440  LD    D,A           ; DE törlése
F441  LD    A,(HL)        ; A-ba a tömb dimenziószáma
F442  INC   HL
F443  EX    DE,HL
F444  PUSH  BC
F445  PUSH  HL
F446  PUSH  AF
F447  LD    A,A4
F449  CALL  NZ,FD54       ; ha NZ, és a következő byte nem ",",
                            hibajelzés
F44C  CALL  FAC4          ; HL-be az index értéke
F44F  EX    DE,HL
F450  LD    C,(HL)
F451  INC   HL
F452  LD    B,(HL)        ; BC-be az aktuális dimenzióhoz tartozó
                            maximális index
F453  INC   HL
F454  EX    DE,HL
F455  OR    A
F456  SBC   HL,BC
F458  JR    NC,D2         ; a megadott index nagyobb a maximális
                            indexnél, ugrás az F42CH címre
F45A  ADD   HL,BC         ; HL visszaállítása
F45B  POP   AF
F45C  PUSH  AF
F45D  PUSH  DE
F45E  DEC   A
F45F  JR    Z,0E          ; nincs több dimenzió, ugrás az F46FH
                            címre
F461  EX    DE,HL
F462  LD    C,(HL)
F463  INC   HL
F464  LD    B,(HL)        ; BC-be a következő dimenzió maximális
                            indexe
F465  INC   HL
F466  PUSH  HL
F467  PUSH  AF
F468  CALL  FCB3          ; HL-be adott dimenzióhoz tartozó első
                            elem sorszáma -1
F46B  POP   AF
```

```
  F46C  POP  DE
  F46D  JR   EF          ; ugrás az F45EH címre
  F46F  POP  DE
  F470  POP  AF
  F471  POP  BC
  F472  ADD  HL,BC       ; HL-be az aktuális dimenzióhoz tartozó
                           BC-dik elem sorszáma
  F473  DEC  A
  F474  JR   NZ,CF       ; van még dimenzió, ugrás az F445H címre
  F476  LD   A,95
  F478  CALL FD54        ; ha nem ")" következik, hibajelzés
  F47B  EX   DE,HL
  F47C  POP  BC
  F47D  BIT  1,C         ; tipusbyte vizsgálat
  F47F  LD   BC,0006
  F482  JR   NZ,03       ; numerikus, ugrás az F487H címre
  F484  LD   C,(HL)      ; C-be a maximális string hossz
  F485  INC  BC
  F486  INC  BC
  F487  PUSH HL
  F488  CALL FCB3        ; HL-be a tömb elemének címe a tömb el-
                           sö eleméhez viszonyítva
  F48B  POP  DE
  F48C  ADD  HL,DE       ; HL-be a tömb elem címe
  F48D  RET

A 3. aritmetikai rutin (kivonás):
  F48E  PUSH AF
  F48F  CALL F726        ; a 4. aritmetikai rutin
  F492  POP  AF

A 0. aritmetikai rutin (összeadás):
  F493  PUSH AF
  F494  PUSH IY
  F496  LD   DE,0009
  F499  ADD  IY,DE       ; az elsö tényezö címe IY-ba
  F49B  EX   SP,IY       ; a második tényezö címe IY-ba
  F49D  LD   A,(IY+08H)  ; A-ba a karakterisztika byte
  F4A0  AND  7F          ; az elöjelbit törlése
  F4A2  LD   C,A
  F4A3  LD   A,(IY+11H)  ; A-ba az elsö tényezö kitevöje
  F4A6  AND  7F          ; az elöjelbit törlése
  F4A8  SUB  C           ; a két karakterisztika különbsége
  F4A9  JP   P,F4BF      ; az elsö tényezö karakterisztikája >=,
                           ugrás
```

```
F4AC  EX   SP,IY
F4AE  PUSH AF
F4AF  NEG              ; A=-A
F4B1  CALL F790        ; közös karakterisztikára hozás
F4B4  POP  BC
F4B5  EX   SP,IY
F4B7  LD   A,(IY+11H)
F4BA  SUB  B           ; A-ba a közös karakterisztika
F4BB  LD   (IY+11H),A  ; az eredmény karakterisztikája
F4BE  XOR  A           ; A-ba 0
F4BF  CALL NZ,F790     ; közös karakterisztikára hozás
F4C2  LD   A,(IY+08H)  ; A-ba a második tényező kitevője
F4C5  XOR  (IY+11H)
F4C8  RLCA
F4C9  CALL NC,F4F4     ; azonos elöjelü a két tényező, ugrás az
                         összeadásra
F4CC  EX   SP,IY
F4CE  JR   NC,1E       ; azonos a két tényező elöjele, ugrás az
                         F4EEH címre
F4D0  CALL F71C        ; ha negatív az elsö tényező, kivonás
F4D3  EX   SP,IY
F4D5  CALL F71C        ; ha negatív a második tényező, kivonás
F4D8  CALL F4F4        ; összeadás

F4DB  EX   SP,IY
F4DD  LD   A,(IY+07H)  ; A-ba az eredmény legnagyobb helyiérté-
                         kü két számjegye
F4E0  OR   A
F4E1  JR   Z,0B        ; ha 0, ugrás az F4EEH címre
F4E3  CALL F70C        ; kivonás
F4E6  LD   A,80
F4E8  XOR  (IY+08H)
F4EB  LD   (IY+08H),A  ; az új elöjel a régi ellentéte
F4EE  POP  AF
F4EF  CALL F734        ; kerekítés 6 byte-os mantisszára
F4F2  POP  AF
F4F3  RET              ; a 0. és a 2. aritmetikai rutin vége
```

Két szám mantisszájának összeadása:

```
F4F4  PUSH AF
F4F5  PUSH IY
F4F7  POP  DE          ; DE az egyik összeadandóra mutat
F4F8  LD   HL,0009
F4FB  ADD  HL,DE       ; HL a másik összeadandóra és az ered-
                         ményre mutat
```

```
  F4FC  LD   B,07
  F4FE  INC  HL
  F4FF  INC  DE
  F500  LD   A,(DE)
  F501  ADC  A,(HL)       ; a megfelelö számjegy párok összege
  F502  DAA               ; decimális igazítás a BCD alakhoz
  F503  LD   (HL),A       ; az eredmény eltárolása
  F504  DJNZ F8           ; a számjegyek még nem fogytak el, ugrás
                            az F4FEH címre
  F506  POP  AF
  F507  RET

0 szám egy másik szám helyére:
  F508  LD   DE,0009
  F50B  ADD  IY,DE        ; a régi szám felszabadítása
  F50D  CALL F9F9         ; minden számjegy helyére 00H
  F510  POP  AF
  F511  RET

A 2. aritmetikai rutin (szorzás):
  F512  PUSH AF
  F513  LD   A,(IY+0FH)   ; A-ba az elsö tényezö legmagasabb he-
                            lyiértékü byte-ja
  F516  OR   A
  F517  JR   Z,EF         ; ha 0, ugrás az F508H címre
  F519  LD   A,(IY+06H)   ; A-ba a második tényezö legmagasabb
                            helyiértékü byte-ja
  F51C  OR   A
  F51D  JR   Z,E9         ; ha 0, ugrás az F508H címre
  F51F  LD   A,(IY+11H)   ; A-ba az elsö tényezö kitevöje
  F522  LD   L,(IY+08H)   ; L-be a második tényezö kitevöje
  F525  XOR  L
  F526  LD   H,A
  F527  XOR  L            ; A eredeti értékének visszaállítása
  F528  AND  7F           ; az elöjelbit törlése
  F52A  LD   L,A
  F52B  LD   A,(IY+08H)   ; A-ba a második tényezö kitevöje
  F52E  AND  7F           ; az elöjelbit törlése
  F530  ADD  A,L          ; a két karakterisztika összege A-ba
  F531  SUB  40
  F533  JR   C,D3         ; az eredmény túl kicsi, ugrás az
                            F508H címre
  F535  JP   M,F912       ; hibajelzés
  F538  INC  A
  F539  JP   M,F912       ; hibajelzés
```

```
F53C  DEC  A
F53D  LD   L,A
F53E  PUSH HL
F53F  EXX
F540  EX   SP,HL
F541  PUSH DE
F542  PUSH BC
F543  PUSH HL
F544  PUSH IY
F546  POP  HL
F547  LD   E,L
F548  LD   D,H          ; DE-be HL, ill. IY, a második tényező
                          címe
F549  LD   BC,0010
F54C  ADD  HL,BC        ; HL az első tényező "elöcsordulás"
                          byte-jára mutat
F54D  CALL F5E8         ; az első tényező széthúzása egy karak-
                          ter-egy byte-ra
F550  PUSH DE
F551  LD   (HL),A
F552  DEC  HL
F553  LD   (HL),A
F554  DEC  HL           ; HL a második tényező "elöcsordulás"
                          byte-jára mutat
F555  CALL F5E8         ; a második tényező széthúzása egy karak-
                          ter-egy byte-ra
F558  LD   HL,000A
F55B  ADD  HL,DE
F55C  LD   C,L
F55D  LD   B,H
F55E  LD   DE,0013
F561  ADD  HL,DE
F562  POP  DE
F563  INC  DE           ; DE-be a második tényező elejének címe
F564  PUSH HL
F565  OR   02
F567  EX   AF,AF'
F568  EXX
F569  LD   C,0C
F56B  EXX
F56C  POP  HL
F56D  PUSH HL
F56E  PUSH BC
F56F  LD   A,(DE)       ; A-ba a második tényező aktuális szám-
                          jegye
```

```
F570  EXX
F571  EX    AF,AF'
F572  PUSH  AF
F573  EX    AF,AF'
F574  ADD   A,A
F575  JR    Z,39          ; ha 0, ugrás az F5B0H címre
F577  LD    D,A
F578  ADD   A,A
F579  ADD   A,A
F57A  ADD   A,D           ; A=10*A
F57B  LD    D,A
F57C  LD    H,C0          ; a szorzótábla a C003H címen kezdödik
F57E  LD    E,00
F580  LD    B,10
F582  EX    AF,AF'
F583  JP    P,F58E
F586  INC   E
F587  DEC   E
F588  JR    Z,26          ; ha E=0, ugrás az F5B0H címre
F58A  EX    AF,AF'
F58B  XOR   A
F58C  JR    0D            ; ugrás az F59BH címre
F58E  CP    0C
F590  JR    NC,F8         ; ha A>11, ugrás az F58AH címre
F592  EX    AF,AF'
F593  EXX
F594  LD    A,(BC)        ; A-ba az egyik tényezö számjegye
F595  EXX
F596  ADD   A,D           ; A-ba az egyik tényezö + a másik tényezö 10-szerese
F597  ADD   A,03
F599  LD    L,A           ; HL-be a két tényezö megfelelö számjegyeinek szorzatát meghatározó elem címe a Szorzótáblában
F59A  LD    A,(HL)        ; A-ba a két számjegy szorzata BCD-ben
F59B  ADD   A,E           ; átvitel hozzáadása
F59C  DAA                 ; decimális igazítás
F59D  EXX
F59E  ADD   A,(HL)
F59F  DAA                 ; decimális igazítás
F5A0  LD    (HL),00
F5A2  RLD                 ; az eredmény eltétele
F5A4  INC   HL
F5A5  INC   BC
F5A6  EXX
```

```
  F5A7  RLCA
  F5A8  RLCA
  F5A9  RLCA
  F5AA  RLCA              ; A felsö 4 bitje az alsó 4 helyére és
                            viszont
  F5AB  LD    E,A
  F5AC  EX    AF,AF'
  F5AD  DEC   A
  F5AE  DJNZ  D3          ; ugrás az F583H címre
  F5B0  POP   AF
  F5B1  INC   A
  F5B2  EX    AF,AF'
  F5B3  DEC   C
  F5B4  EXX
  F5B5  POP   BC
  F5B6  DEC   BC
  F5B7  INC   DE
  F5B8  JR    NZ,B2       ; még nincs vége a szorzásnak, ugrás az
                            F57CH címre
  F5BA  POP   HL          ; HL-be az eredmény címe
  F5BB  LD    BC,000F
  F5BE  ADD   HL,BC       ; HL az eredmény legnagyobb helyiértékü
                            byte-jára mutat
  F5BF  LD    E,L
  F5C0  LD    D,H         ; DE-be HL
  F5C1  CALL  F5DA        ; az eredmény egy byte-egy számjegy a-
                            lakból egy byte-két számjegy alakba
  F5C4  PUSH  HL
  F5C5  POP   IY
  F5C7  POP   HL          ; L-be az eredmény karakterisztikája, H-
                            ba az elöjelek szorzata
  F5C8  LD    A,H
  F5C9  AND   80
  F5CB  OR    L           ; A-ba az eredmény kitevö byte-ja kerül
  F5CC  LD    (IY+08H),A  ; a kitevö byte beírása
  F5CF  LD    (IY+00H),09; a szám jelzö byte beírása
  F5D3  POP   BC
  F5D4  POP   DE
  F5D5  POP   HL
  F5D6  EXX
  F5D7  JP    F4EF        ; kerekítés 6 byte-ra

Egy byte-egy digit -- egy byte-két digit konverzió:
  F5DA  LD    B,07
  F5DC  LD    A,(DE)      ; A-ba egy digit
```

```
F5DD  DEC  DE
F5DE  RLD              ; a digit a HL címezte byte-ba
F5E0  LD   A,(DE)      ; A-ba a következő digit
F5E1  DEC  DE
F5E2  RLD              ; a digit a HL címezte byte alsó felére,
                         az előző digit a byte felső felére
F5E4  DEC  HL
F5E5  DJNZ F5          ; ugrás az F5DCH címre, ha még nem kész
F5E7  RET

Egy byte-két digit -- egy byte-egy digit konverzió:
F5E8  LD   B,07
F5EA  XOR  A           ; A-ba 0
F5EB  RLD              ; A-ba a HL címezte byte alsó digitje
F5ED  LD   (DE),A      ; a digit letétele
F5EE  DEC  DE
F5EF  XOR  A           ; A-ba 0
F5F0  RLD              ; A-ba a HL címezte byte felső bitje
F5F2  LD   (DE),A      ; a digit letétele a következő byte-ba
F5F3  DEC  DE
F5F4  DEC  HL
F5F5  DJNZ F3          ; ugrás az F5EAH címre
F5F7  XOR  A
F5F8  RET

F5F9  RST  08H
F5FA  DB   0B          ; "Cannot divide by 0" hibajelzés

Az 1. aritmetikai rutin (osztás):
F5FB  PUSH AF
F5FC  LD   A,(IY+06H) ; A-ba az osztó felső byte-ja
F5FF  OR   A
F600  JR   Z,F7        ; ha 0, ugrás az F5F9H címre
F602  LD   A,(IY+0FH) ; A-ba az osztandó felső byte-ja
F605  OR   A
F606  JP   Z,FA1A      ; ha 0, ugrás
F609  EXX
F60A  PUSH HL
F60B  PUSH DE
F60C  PUSH BC
F60D  PUSH IY
F60F  LD   DE,0009
F612  ADD  IY,DE       ; IY az osztandóra mutat
F614  LD   L,(IY+FFH) ; L-be az osztó kitevő byte-ja
F617  LD   H,(IY+08H) ; H-ba az osztandó kitevő byte-ja
```

```
F61A  EX   SP,HL
F61B  LD   (171C),HL   ; az osztó címének eltárolása
F61E  PUSH HL
F61F  EX   SP,IY       ; IY-ba az osztó címe
F621  CALL F9F1        ; 0 szám a BASIC STACK-be
F624  LD   B,04
F626  LD   HL,(171C)   ; HL-be az osztó címe
F629  EXX
F62A  LD   BC,FFF7
F62D  LD   HL,(171C)   ; HL-be az osztó címe
F630  LD   E,L
F631  LD   D,H         ; DE-be HL
F632  OR   A
F633  SBC  HL,BC       ; HL-be az osztandó címe
F635  EX   DE,HL
F636  ADD  HL,BC
F637  ADD  HL,BC       ; HL a BASIC STACK alá 9 byte-tal mutat
```

Az osztandó és az osztó mantisszájának különbsége:

```
F638  LD   B,07
F63A  OR   A
F63B  INC  DE
F63C  LD   A,(DE)      ; az osztandó egy byte-ja
F63D  EXX
F63E  INC  HL
F63F  SBC  A,(HL)      ; A-ból az osztó megfelelö byte-ja
F640  EXX
F641  DAA              ; decimális igazítás
F642  INC  HL
F643  LD   (HL),A      ; a különbség eltárolása
F644  DJNZ F5          ; még nincs kész, ugrás az F63BH címre
F646  AND  F0
F648  JR   NZ,13       ; ha a különbség legfelsö byte-ja nem 0,
                       ugrás az F65DH címre
F64A  EXX
F64B  CALL F7D2        ; A-ba a leendö eredmény megfelelö byte-
                       ja a megfelelö módon
F64E  INC  A           ; a hányados megfelelö byte-ja 1-gyel nö
F64F  CALL F7E3        ; A visszatétele a hányadosba
F652  LD   HL,(171C)   ; HL-be az osztó címe
F655  EXX
F656  LD   BC,0007
F659  LDDR             ; a különbség az osztandó helyére
F65B  JR   DB          ; ugrás az F638H címre
F65D  EX   SP,IY
```

```
  F65F  CALL F784        ; az osztandóban lévö szám jegyeinek
                           jobbra shift-telése 1 digittel
  F662  EX   SP,IY
  F664  EXX
  F665  INC  B           ; a hányados következö digitjének megha-
                           tározása következik
  F666  LD   A,B
  F667  CP   10
  F669  JR   C,BB        ; ugrás az F626H címre, ha még nincs
                           kész az osztás
  F66B  EXX
  F66C  LD   HL,(171C)   ; HL-be az osztó címe
  F66F  DEC  HL
  F670  DEC  HL          ; HL az eredmény mantissza végére mutat
  F671  LD   BC,0008
  F674  LDDR             ; a hányados mántissza az osztandó he-
                           lyére
  F676  POP  IY
  F678  POP  HL
  F679  LD   A,H         ; A-ba az osztó kitevö byte-ja
  F67A  XOR  L
  F67B  AND  80
  F67D  LD   C,A         ; C-be az osztó és osztandó elöjelének
                           szorzata
  F67E  RES  7,H
  F680  RES  7,L
  F682  LD   A,H
  F683  SUB  L
  F684  ADD  A,40        ; A-ba a hányados karakterisztikája
  F686  XOR  C           ; A-ba a hányados kitevö byte-ja
  F687  LD   (IY+08H),A  ; a hányados kész
  F68A  XOR  C
  F68B  JP   P,F5D3      ; ugrás, ha a karakterisztika >=0
  F68E  CALL F9F9        ; az eredmény túl kicsi, ezért 0
  F691  JR   F8          ; ugrás az F68BH címre

A BASIC STACK két alsó számának összehasonlítása:
  F693  PUSH IY
  F695  POP  HL
  F696  LD   BC,0009
  F699  ADD  HL,BC       ; HL-be a felsö szám kezdete
  F69A  LD   E,L
  F69B  LD   D,H
  F69C  DEC  DE          ; DE-be az alsó szám kitevö byte-jának
                           címe
```

```
  F69D  ADD  HL,BC      ; HL a felsö szám fölé mutat
  F69E  PUSH HL
  F69F  POP  IY
  F6A1  DEC  HL         ; HL-be a felsö szám kitevö byte-jának
                          címe
  F6A2  LD   A,(HL)     ; A-ba a felsö szám kitevö byte-ja
  F6A3  AND  80         ; A-ba a felsö szám elöjele
  F6A5  RRCA
  F6A6  LD   C,A
  F6A7  LD   A,(DE)     ; A-ba az alsó szám kitevö byte-ja
  F6A8  AND  80         ; A-ba az alsó szám elöjele
  F6AA  RRCA
  F6AB  SUB  C
  F6AC  RET  NZ         ; a két szám elöjele különböző, RET
  F6AD  ADD  A,C
  F6AE  JR   Z,01       ; ha pozitívak, ugrás az F6B1H címre
  F6B0  EX   DE,HL
  F6B1  LD   A,(DE)     ; A-ba az egyik szám kitevö byte-ja
  F6B2  DEC  DE
  F6B3  AND  7F         ; az elöjelbit törlése
  F6B5  LD   C,A
  F6B6  LD   A,(HL)     ; a másik szám kitevö byte-ja A-ba
  F6B7  DEC  HL
  F6B8  AND  7F         ; az elöjelbit törlése
  F6BA  SUB  C
  F6BB  RET  NZ         ; a két szám karakterisztikája különböző
A mantisszák összehasonlítása:
  F6BC  LD   B,07
A byte-ok felsö digitjének összehasonlítása:
  F6BE  LD   A,(DE)     ; A-ba a mantissza byte
  F6BF  AND  F0         ; az alsó digit törlése
  F6C1  RRCA
  F6C2  LD   C,A
  F6C3  LD   A,(HL)     ; A-ba a másik mantissza megfelelö byte-
                          ja
  F6C4  AND  F0         ; az alsó digit törlése
  F6C6  RRCA
  F6C7  SUB  C
  F6C8  RET  NZ         ; a megfelelö számjegyek eltérök, RET
A byte-ok alsó digitjének összehasonlítása:
  F6C9  LD   A,(DE)     ; A-ba a mantissza byte
  F6CA  AND  0F         ; a felsö digit törlése
  F6CC  LD   C,A
  F6CD  LD   A,(HL)     ; A-ba a mésik szám megfelelö byte-ja
  F6CE  AND  0F         ; a felsö digit törlése
```

```
F6D0  SUB  C
F6D1  RET  NZ          ; a két digit különböző, RET
F6D2  DEC  DE
F6D3  DEC  HL
F6D4  DJNZ E8          ; ugrás az F6BEH címre
F6D6  RET              ; az összehasonlítás vége, ha egyenlőek

Két, a BASIC STACK alján lévő string összehasonlítása:
F6D7  PUSH IY
F6D9  POP  HL
F6DA  INC  HL
F6DB  PUSH HL
F6DC  LD   C,(HL)      ; C-be az alsó string hossza
F6DD  LD   B,00
F6DF  ADD  HL,BC
F6E0  INC  HL          ; HL-be a felsö string kezdete
F6E1  INC  HL
F6E2  PUSH HL
F6E3  LD   A,(HL)      ; A-ba a felsö string hossza
F6E4  CP   C
F6E5  JR   C,01        ; a felsö string rövidebb, ugrás az
                         F6E8H címre
F6E7  LD   A,C
F6E8  LD   C,(HL)      ; C-be a felsö string hossza
F6E9  ADD  HL,BC
F6EA  INC  HL
F6EB  PUSH HL
F6EC  POP  IY          ; a két string felszabadítása a BASIC
                         STACK-ből
F6EE  LD   C,A         ; C-be a rövidebb string hossza
F6EF  INC  BC
F6F0  POP  DE
F6F1  POP  HL
F6F2  LD   A,(DE)      ; A-ba az egyik string megfelelö byte-ja
F6F3  INC  DE
F6F4  CPI              ; összehasonlítás a másik string megfel-
                         elö byte-jával
F6F6  JP   PO,F6FB     ; túlcsordulás, ugrás
F6F9  JR   Z,F7        ; egyenlök, ugrás az F6F2H címre
F6FB  DEC  HL
F6FC  LD   C,(HL)
F6FD  LD   L,A
F6FE  LD   H,B
F6FF  SBC  HL,BC
F701  LD   A,H
```

```
  F702  RET  Z          ; a két string egyenlö, A-ban 0
  F703  RET  M          ; a felsö string nagyobb
  F704  LD   A,01
  F706  RET             ; a string összehasonlítás vége

Negatív szám mantisszájának komplemense:
  F707  BIT  7,(IY+08H)
  F70B  RET  Z          ; ha a szám pozitív, RET
  F70C  PUSH IY
  F70E  POP  HL
  F70F  LD   DE,0007    ; a mantissza byte-ok száma E-be, D-be 0
  F712  OR   A          ; a CARRY törlése
  F713  INC  HL
  F714  LD   A,D        ; A-ba 0
  F715  SBC  A,(HL)     ; 00H-CARRY-(az mantissza byte)
  F716  DAA             ; decimális igazítás
  F717  LD   (HL),A     ; az eredmény vissza a megfelelö helyre
  F718  DEC  E
  F719  JR   NZ,F8      ; még nincs vége, ugrás az F713H címre
  F71B  RET

Negatív szám komplemense:
  F71C  BIT  7,(IY+08H)
  F720  RET  Z          ; pozitív, RET
  F721  CALL F70C       ; a mantissza komplemense
  F724  JR   05         ; ugrás az F72BH címre

A 4. aritmerikai rutin (szorzás -1-gyel):
  F726  LD   A,(IY+06H)
  F729  OR   A
  F72A  RET  Z          ; a szám 0, RET
  F72B  LD   A,80
  F72D  XOR  (IY+08H)   ; A-ba az elöjel ellentetje
  F730  LD   (IY+08H),A ; az új elöjel beírása
  F733  RET

A BASIC STACK-ben levö szám átalakítása 6 byte-os mantisszájú
számmá:
  F734  CALL F9DF       ; csupa 0 a mantissza?
  F737  RET  Z          ; igen, RET
  F738  LD   A,(IY+07H) ; A-ba az "elöcsordulás" byte
  F73B  OR   A
  F73C  JR   Z,13       ; az elöcsordulás 0, ugrás az F751H cím-
                          re
  F73E  LD   A,01
```

```
F740  CALL F790        ; a digitek eltolása balra 1 digittel
F743  LD   A,(IY+08H)  ; A-ba a kitevö byte
F746  AND  7F          ; az elöjel bit törlése
F748  CP   7E
F74A  JP   Z,F912      ; túl nagy kitevö, hibajelzés
F74D  INC  (IY+08H)    ; a kitevö növelése 1-gyel
F750  RET
F751  LD   A,(IY+06H)  ; A-ba a következö legmagasabb helyiér-
                         tékü byte
F754  AND  F0          ; az alsó digit törlése
F756  RET  NZ          ; a felsö digit nem nulla, RET
F757  CALL F784        ; a mantissza digitek eltolása jobbra 1
                         digittel
F75A  LD   A,(IY+08H)  ; A-ba a kitevö byte
F75D  DEC  (IY+08H)    ; a kitevö csökkentése
F760  AND  7F          ; az elöjelbit törlése
F762  JR   NZ,ED       ; nem 0 a karakterisztika, ugrás az
                         F751H címre
F764  JP   F9F9        ; a szám helyére 0

A túlcsordulás byte törlése kerekítéssel:
F767  PUSH HL
F768  PUSH IY
F76A  POP  HL
F76B  INC  HL          ; HL-be a szám legalsó, tulcsordulás
                         byte-jának címe
F76C  LD   A,(HL)      ; A-ba a túlcsordulás byte
F76D  CP   50
F76F  LD   (HL),00     ; a túlcsordulás byte törlése
F771  JR   C,0F        ; ha a túlcsorduló kétjegyü szám < 50,
                         ugrás az F782H címre
A mantisszához 0.000000001 hozzáadása:
F773  LD   B,06
F775  INC  HL
F776  LD   A,(HL)      ; A-ba a második legkisebb helyiértékü
                         mantissza byte
F777  ADD  A,01        ; 1 hozzáadása
F779  DAA              ; decimális igazítás
F77A  LD   (HL),A      ; az új érték visszatöltése
F77B  JR   NC,02       ; ha az összeadáskor nincs átvitel, ug-
                         rás az F77FH címre
F77D  DJNZ F6          ; ugrás az F775H címre, ha még nem kész
F77F  CALL F734        ; kerekítés 6 byte-ra
F782  POP  HL
F783  RET
```

A BASIC STACK-ben levö szám mantisszájának szorzása 10-zel:

```
F784  PUSH IY
F786  POP  HL
F787  LD   B,07
F789  XOR  A          ; A-ba 0
F78A  INC  HL         ; HL a mantissza aljáról indul
F78B  RLD             ; a mantissza digitek jobbra tolása egy
                        digittel
F78D  DJNZ FB         ; ugrás az F78AH címre, ha még van byte
F78F  RET
```

A BASIC STACK-ben lévö szám mantisszájának eltolása balra az A-ban található számú digittel:

```
F790  CP   0E
F792  JR   NC,30      ; ha A>13, ugrás az F7C4H címre
F794  LD   BC,00FF
F797  PUSH IY
F799  POP  HL
F79A  PUSH HL
F79B  LD   E,L
F79C  LD   D,H        ; DE-be HL
F79D  INC  HL
F79E  SUB  02
F7A0  INC  C
F7A1  JR   NC,FA      ; ugrás az F79DH címre, ha A>0
F7A3  JR   Z,12       ; ugrás az F7B7H címre, ha C=0
F7A5  INC  DE         ; DE a szám alsó byte-jára mutat
F7A6  PUSH AF
F7A7  LD   A,07
F7A9  SUB  C          ; A-ba az eltolás nagysága byte-ban
F7AA  LD   C,A
F7AB  LDIR            ; az egész byte-os eltolás
F7AD  EX   DE,HL
F7AE  SUB  08
F7B0  CPL
F7B1  LD   (HL),B     ; a felszabadult helyre 0 töltés
F7B2  INC  HL
F7B3  DEC  A
F7B4  JR   NZ,FB      ; ha nem kész, ugrás az F7B1H címre
F7B6  POP  AF
F7B7  POP  HL
F7B8  INC  A
F7B9  RET  NZ         ; ha páros digitü eltolás volt, RET
```

Egy digites eltolás balra

```
F7BA  LD   C,07
```

```
  F7BC  ADD  HL,BC       ; HL-be a mantissza felsö byte-jának cí-
                           me
  F7BD  LD   B,C
  F7BE  RRD              ; eltolás balra egy digittel
  F7C0  DEC  HL
  F7C1  DJNZ FB          ; ugrás az F7BEH címre, ha nincs vége
  F7C3  RET
A mantissza törlése:
  F7C4  LD   C,(IY+08H) ; C-be a kitevö byte
  F7C7  CALL F9F9       ; a szám törlése
  F7CA  LD   (IY+08H),C ; a kitevö byte visszaállítása
  F7CD  RET

  F7CE  LD   HL,FEB6    ; HL-be visszatérési cím
  F7D1  PUSH HL         ; visszatérési cím a STACK-be
  F7D2  PUSH HL
  F7D3  PUSH BC
  F7D4  CALL F7FD       ; A-ba a szám megfelelö byte-ja
  F7D7  JR   C,05       ; ugrás az F7DEH címre, ha CARRY=1
  F7D9  LD   A,(HL)     ; A-ba a szám megfelelö byte-ja
  F7DA  RLCA
  F7DB  RLCA
  F7DC  RLCA
  F7DD  RLCA            ; A alsó és felsö felének felcserélése
  F7DE  AND  0F         ; A felsö digitjének törlése
  F7E0  POP  BC
  F7E1  POP  HL
  F7E2  RET

  F7E3  PUSH HL
  F7E4  PUSH BC
  F7E5  CALL F7FD       ; A-ba a szám megfelelö byte-ja
  F7E8  LD   B,F0
  F7EA  JR   C,0B       ; ugrás az F7F7H címre, ha CARRY
  F7EC  SLA  C
  F7EE  SLA  C
  F7F0  SLA  C
  F7F2  SLA  C          ; C alsó fele a felsö fél byte-ba, az
                          alsó felének törlése
  F7F4  LD   A,(HL)     ; A-ba a szám megfelelö byte-ja
  F7F5  LD   B,0F
  F7F7  AND  B          ; az alsó digit törlése
  F7F8  OR   C
  F7F9  LD   (HL),A     ; a byte visszatöltése
  F7FA  POP  BC
```

```
  F7FB  POP  HL
  F7FC  RET

  F7FD  PUSH IY
  F7FF  POP  HL
  F800  SRL  B          ; B=B/2
  F802  PUSH AF
  F803  LD   A,08
  F805  SUB  B          ; A=A-B, a megfelelö mantissza byte re-
                          latív helye
  F806  LD   C,A
  F807  LD   B,00
  F809  ADD  HL,BC      ; HL a megfelelö byte-ra mutat
  F80A  POP  AF         ; A visszaállítása
  F80B  LD   C,A
  F80C  LD   A,(HL)     ; A-ba a megfelelö mantissza byte
  F80D  RET

A STR$ függvény magja:
  F80E  LD   A,(IY+06H) ; A-ba a szám legfelsö byte-ja
  F811  OR   A
  F812  LD   HL,1931    ; HL-be a STR$_BUFFER címe
  F815  LD   (HL),30    ; a "0" karakter a STR$_BUFFER-be
  F817  JR   Z,7E       ; a STR$ argumentuma 0, ugrás az F887H
                          címre
  F819  BIT  7,(IY+08H)
  F81D  JR   Z,06       ; a szám pozitív, ugrás az F825H címre
  F81F  LD   (HL),2D    ; negatív elöjel ("-") a STR$_BUFFER-be
  F821  INC  HL
  F822  CALL F726       ; a 4. aritmetikai rutin
  F825  CALL F767       ; a túlcsordulás byte törlése kerekítés-
                          sel
A szám elsö nem 0 digitjének megkeresése:
  F828  LD   B,0E
  F82A  DEC  B
  F82B  LD   A,B
  F82C  CP   04
  F82E  CALL NC,F7D2    ; a megfelelö digit A-ba
  F831  JR   Z,F7       ; a digit 0, ugrás az F82AH címre

  F833  LD   C,B        ; C-be a digit sorszáma
  F834  LD   B,04
  F836  LD   E,(IY+08H) ; E-be a kitevö byte
  F839  LD   A,E
  F83A  SUB  3B         ; A-ba a karakterisztika 10^(-5)-szerese
```

```
                              binárisan
F83C  LD    D,A
F83D  LD    A,E
F83E  CP    4A
F840  JR    NC,16       ; ha a karakterisztika > 10, ugrás az
                          F858H címre
F842  CP    40
F844  JR    NC,0A       ; a karakterisztika pozitív, ugrás az
                          F850H címre
F846  SUB   C
F847  JR    C,10        ; ha A<C, ugrás az F859H címre
F849  CP    32
F84B  JR    C,0C        ; a számnak van elvesztett mantissza
                          digitje, ugrás az F859H címre
F84D  LD    B,D
F84E  JR    0B          ; ugrás az F85BH címre
F850  LD    A,C
F851  CP    D
F852  JR    NC,07       ; ugrás az F85BH címre, ha D<=C
F854  LD    C,D
F855  OR    A
F856  JR    04          ; ugrás az F85CH címre
F858  SCF
F859  LD    D,05
F85B  INC   C
F85C  PUSH  AF
F85D  LD    A,B
F85E  CP    D
F85F  JR    NZ,03       ; még egész jegy, ugrás az F864H címre
F861  LD    (HL),2E     ; a tizedes pont kitétele
F863  INC   HL
F864  LD    A,B
F865  SUB   04
F867  LD    A,00
F869  CALL  P,F7D2      ; A-ba a következő digit, ha B>3
F86C  ADD   A,30        ; A-ba a számjegy kódja
F86E  LD    (HL),A      ; a számjegy a STR$_BUFFER-be
F86F  INC   HL
F870  INC   B
F871  LD    A,B
F872  CP    C
F873  JR    NZ,E9       ; még van kiírandó mantissza digit, ug-
                          rás az F85EH címre
F875  POP   AF
F876  JR    NC,20       ; ha nincs kiírandó karakterisztika, ug-
```

```
                                rás az F898H címre
  F878  LD    (HL),45     ; az "E" karakter a CHR$_BUFFER-be
  F87A  INC   HL
  F87B  LD    (HL),2B     ; "+" előjel eltárolása
  F87D  LD    A,E
  F87E  SUB   40
  F880  JR    NC,04       ; a karakterisztika pozitív, ugrás az
                            F886H címre
  F882  LD    (HL),2D     ; "_" előjel eltárolása
A karakterisztika számból karakter:
  F884  NEG
  F886  INC   HL
  F887  LD    B,FF
  F889  INC   B
  F88A  LD    C,A
  F88B  SUB   0A          ; A-ból 10
  F88D  JR    NC,FA       ; ugrás az F889H címre, ha A-ban még
                            megvan a 10
  F88F  LD    A,B         ; A-ba a karakterisztika digit
  F890  ADD   A,30        ; A-ba a digit ASCII kódja
  F892  LD    (HL),A      ; a kód eltárolása
  F893  INC   HL
  F894  SUB   B           ; A-ba 30H
  F895  ADD   A,C         ; A-ba a másik karakterisztika digit
                            kódja
  F896  LD    (HL),A      ; a karakter eltárolása
  F897  INC   HL
  F898  LD    DE,1930
  F89B  SCF
  F89C  SBC   HL,DE       ; HL-be a szám string hossza
  F89E  EX    DE,HL
  F89F  LD    (HL),E      ; a string hosszának eltárolása
  F8A0  RET

A programsor értelmezésekor string feldolgozása:
  F8A1  PUSH DE
  F8A2  PUSH BC
  F8A3  LD    E,FF
  F8A5  LD    C,00
A szóköz karakterek kiejtése a string előtt:
  F8A7  LD    A,(HL)      ; A-ba a programsor következő byte-ja
  F8A8  INC   HL
  F8A9  CP    20
  F8AB  JR    Z,FA        ; ha SPACE, ugrás az F8A7H címre
  F8AD  CP    22
```

```
F8AF  JR   NZ,03      ; nem idézöjellel (") kezdödik a string,
                        ugrás az H8B4H címre
F8B1  INC  C
F8B2  LD   A,(HL)     ; A-ba a string karaktere
F8B3  INC  HL
F8B4  INC  E
F8B5  INC  C
F8B6  DEC  C
F8B7  JR   Z,0E       ; a string túl hosszú, ugrás az F8C7H
                        címre
F8B9  CP   FF
F8BB  JR   Z,16       ; programsor vége jel, ugrás az F8D3H-ra
F8BD  CP   22
F8BF  JR   NZ,F1      ; nem idézöjel ("), ugrás az F8B2H címre
F8C1  CP   (HL)
F8C2  INC  HL
F8C3  JR   Z,ED       ; a következö karakter is idézöjel, ug-
                        rás az F8B2H címre
F8C5  JR   0C         ; ugrás az F8D3H címre
F8C7  CP   2C
F8C9  JR   Z,08       ; (,) követi a stringet, ugrás az F8D3H
                        címre
F8CB  CP   21
F8CD  JR   Z,04       ; "!! karakter követi a stringet, ugrás
                        az F8D3H címre
F8CF  CP   FD
F8D1  JR   C,DF       ; nem utasítást befejezö jel következik,
                        ugrés az F8B2H címre
F8D3  DEC  HL
F8D4  PUSH HL
F8D5  PUSH DE
F8D6  LD   B,E        ; B-be a string hossza
F8D7  PUSH IY
F8D9  POP  DE
F8DA  CALL FC8E       ; elfér a string a BASIC STACK-ben?
F8DD  INC  B
F8DE  DEC  B
F8DF  DEC  DE
F8E0  JR   Z,16       ; kész a string átmozgatása a BASIC
                        STACK-be, ugrás az F8F8H címre
F8E2  DEC  HL
F8E3  LD   A,(HL)     ; A-ba a karakter a programsorból
F8E4  CP   22
F8E6  JR   NZ,06      ; nem idézöjel, ugrás az F8EEH címre
F8E8  INC  C
```

```
F8E9  DEC  C
F8EA  JR   Z,02          ; ugrás az F8EEH címre, ha nem string-
                           záró idézöjel
F8EC  DEC  HL
F8ED  LD   A,(HL)        ; a következö karakter A-ba
F8EE  LD   (DE),A        ; a karakter a BASIC STACK-be
F8EF  CP   20
F8F1  JR   NC,EB         ; nem ékezetes karakter, ugrás az F8DEH
                           címre
F8F3  OR   80            ; a legfelsö bit visszaállítása
F8F5  LD   (DE),A        ; a karakter a BASIC STACK-be
F8F6  JR   E6            ; ugrás az F8DEH címre
F8F8  EX   DE,HL
F8F9  POP  DE
F8FA  LD   (HL),E        ; a string hossza
F8FB  DEC  HL
F8FC  LD   (HL),01       ; string bejegyzés jelzö byte
F8FE  PUSH HL
F8FF  POP  IY
F901  POP  HL
F902  POP  BC
F903  POP  DE
F904  RET

Karakterlánc feldolgozása programsor értelmezésekor:
F905  LD   A,(HL)        ; A-ba a programsor következö byte-ja
F906  CP   22
F908  JR   Z,97          ; ha (") karakter, ugrás az F8A1H címre
F90A  CALL F914          ; a szám feldolgozása
F90D  JP   NC,FD5A
F910  RLA
F911  RET  NC

F912  RST  08H
F913  DB   0B            ; "Cannot divide by 0" hibajelzés

F914  PUSH BC
F915  PUSH DE
F916  CALL F9F4          ; 0 szám a BASIC STACK-be
F919  PUSH HL
F91A  PUSH IY
F91C  POP  HL
F91D  LD   DE,0007
F920  ADD  HL,DE         ; HL a legfelsö mantissza byte-ra mutat
F921  EX   SP,HL         ; HL a programsorba mutat
```

```
F922  LD    E,3F
F924  LD    B,0B
F926  LD    A,(HL)      ; A-ba a karakterlánc byte-ja
F927  INC   HL
F928  CP    20
F92A  JR    Z,FA        ; ha szóköz, ugrás az F926H címre
F92C  CP    2B
F92E  JR    Z,06        ; ha "+", ugrás az F936H címre
F930  CP    2D
F932  JR    NZ,04       ; ha nem "-", ugrás az F938H címre
F934  SET   1,D
F936  LD    A,(HL)      ; A-ba a karakterlánc következő byte-ja
F937  INC   HL
F938  CP    2E
F93A  JR    Z,2D        ; ha "." (tizedes pont), ugrás az F969H
                          címre
F93C  SUB   3A
F93E  JR    NC,31       ; ha A>39H, ugrás az F971H címre
F940  ADD   A,0A
F942  JR    NC.2D       ; ha a karakter nem számjegy, ugrás az
                          F971H címre
F944  LD    C,A         ; C-be a számjegy binárisan
F945  SET   7,D
F947  LD    A,D
F948  RRCA
F949  JR    NZ,09       ; ha A nem 0, ugrás az F954H címre
F94B  BIT   2,D
F94D  JR    NZ,07       ; ha D 2. bitje nem 0, ugrás az F956H
                          címre
F94F  JR    NC,E5       ; ha az előjel "+" volt, ugrás az F936H
                          címre
F951  DEC   E
F952  JR    E2          ; ugrás az F936H címre
F954  SET   2,D
F956  JR    C,01        ; ha pozitív, ugrás az F959H címre
F958  INC   E
F959  LD    A,B
F95A  DEC   A
F95B  JR    Z,D9        ; ugrás az F936H címre, ha B 1 volt
F95D  LD    B,A         ; B 1-gyel csökkent
F95E  EX    SP,HL       ; HL a BASIC STACK-be mutat
F95F  RRCA
F960  JR    C,01        ; ugrás az F963H címre, ha nem a követ-
                          kező mantissza byte-ba kerül a digit
F962  DEC   HL
```

```
F963  LD   A,C          ; A-ba a szám egy digitje
F964  RLD               ; a digit a BASIC STACK-be
F966  EX   SP,HL        ; HL a karakterláncra mutat
F967  JR   CD           ; ugrás az F936H címre
F969  SET  7,D
F96B  BIT  0,D
F96D  SET  0,D
F96F  JR   Z,C5         ; ugrás az F936H címre, ha Z
F971  EX   SP,HL        ; HL a BASIC STACK-be mutat
F972  DEC  B
F973  BIT  0,B
F975  JR   Z,03         ; B páros, ugrás az F97AH címre
F977  XOR  A            ; A-ba 0
F978  RLD               ; a soronkövetkező digit helyére 0
F97A  POP  HL           ; HL a karakterláncra mutat
F97B  DEC  HL
F97C  LD   A,(HL)       ; A-ba a következő byte
F97D  CALL FBC8         ; a byte vizsgálata
F980  CP   45
F982  LD   A,E
F983  JR   NZ,40        ; ha nem "E", ugrás az F9C5H címre
F985  SET  7,D
F987  INC  HL
F988  LD   A,(HL)       ; A-ba a következő byte
F989  CP   98
F98B  JR   Z,0E         ; ha "+", ugrás az F99BH címre
F98D  CP   2B
F98F  JR   Z,0A         ; ha "+", ugrás az F99BH címre
F991  CP   A2
F993  JR   Z,04         ; ha "-", ugrás az F999H címre
F995  CP   2D
F997  JR   NZ,03        ; ha nem "-", ugrás az F99CH címre
F999  SET  5,D
F99B  INC  HL
F99C  LD   B,00
F99E  DEC  HL
F99F  INC  HL
F9A0  LD   A,(HL)       ; A-ba a következő byte
F9A1  SUB  3A
F9A3  JR   NC,12        ; A<39H, ugrás az F9B7H címre
F9A5  ADD  A,0A
F9A7  JR   NC,0E        ; nem számjegy volt, ugrás az F9B7H cím-
                          re
F9A9  LD   C,A
F9AA  LD   A,B
```

```
F9AB  ADD  A,A
F9AC  LD   B,A          ; B=2*B
F9AD  ADD  A,A
F9AE  ADD  A,A
F9AF  ADD  A,B          ; A=10*B
F9B0  ADD  A,C
F9B1  LD   B,A
F9B2  JP   P,F99F       ; ha P, a karakterisztika következő di-
                          gitje A alsó felébe
F9B5  SET  6,D
F9B7  LD   A,B
F9B8  BIT  5,D
F9BA  JR   Z,02         ; ugrás az F9BEH címre, ha Z
F9BC  NEG
F9BE  ADD  A,E
F9BF  CP   7F
F9C1  JR   C,02         ; ugrás az F9C5H címre, ha A<7FH
F9C3  SET  6,D
F9C5  PUSH DE
F9C6  LD   (IY+08H),A   ; a karakterisztika beírása
F9C9  BIT  1,D
F9CB  CALL NZ,F726      ; ha a szám negatív, a 4. aritmetikai
                          rutin
F9CE  CALL F9DF         ; a szám 0?
F9D1  POP  DE
F9D2  LD   A,D
F9D3  RRA
F9D4  AND  D
F9D5  AND  20
F9D7  LD   A,D
F9D8  RLA
F9D9  CALL NZ,F9F9      ; 0 szám a BASIC STACK-be, ha NZ
F9DC  POP  DE
F9DD  POP  BC
F9DE  RET

A BASIC STACK-ben található szám mantisszája 0?:
F9DF  PUSH HL
F9E0  PUSH IY
F9E2  POP  HL
F9E3  LD   B,07
F9E5  XOR  A            ; A-ba 00H
F9E6  INC  HL
F9E7  OR   (HL)         ; a mantissza byte 0?
F9E8  JR   NZ,05        ; nem, ugrás a rutin végére
```

```
  F9EA  DJNZ FA          ; ha van még vizsgálatlan byte, ugrás az
                           F9E6H címre
  F9EC  LD   (IY+08H),A  ; a mantissza 0, a kitevö is legyen az
  F9EF  POP  HL
  F9F0  RET

0 szám a BASIC STACK-be:
  F9F1  CALL FC8E        ; van elég szabad memória?
  F9F4  LD   DE,FFF7
  F9F7  ADD  IY,DE       ; a BASIC STACK mutató csökkentése
  F9F9  PUSH IY
  F9FB  EX   SP,HL
  F9FC  LD   (HL),09     ; számhoz tartozó jelzö byte
  F9FE  LD   B,08
  FA00  XOR  A           ; A-ba 0
  FA01  INC  HL
  FA02  LD   (HL),A      ; a szám jegyei 0-k
  FA03  DJNZ FC
  FA05  POP  HL
  FA06  SCF
  FA07  RET

1 szám a BASIC STACK-be:
  FA08  CALL F9F9        ; 0 szám a BASIC STACK-be
  FA0B  LD   (IY+08H),40; a kitevö +E00
  FA0F  LD   (IY+06H),10; a mantissza 1.0
  FA13  RET

Az F9F9H rutin hívása az EXT. ROM-ból:
  FA14  LD   HL,FEB6     ; az EXT. ROM-ba vivö rutin címe HL-be
  FA17  PUSH HL
  FA18  JR   DF          ; ugrás az F9F9H címre

9 byte felszabadítása a BASIC STACK-ből:
  FA1A  POP  AF
  FA1B  LD   DE,0009
  FA1E  ADD  IY,DE
  FA20  RET

String felszabadítása a BASIC STACK-ből:
  FA21  LD   E,(IY+01H) ; E-be a string hossza
  FA24  LD   D,00
  FA26  INC  DE
  FA27  INC  DE
  FA28  ADD  IY,DE
  FA2A  RET
```

```
A HL-ben lévö 2 byte-os számból 9 byte-os szám a BASIC STACK-be:
  FA2B  CALL F9F1        ; 0 szám a BASIC STACK-be
  FA2E  INC  IY
  FA30  LD   (171C),IY   ; IY eltárolása
  FA34  DEC  IY
  FA36  PUSH HL
  FA37  CALL FC83        ; HL complementálása, ha a legfelsö bit 1
  FA3A  SCF
  FA3B  ADC  HL,HL       ; HL=HL+HL+CARRY
  FA3D  JR   NC,FC       ; ugrás az FA3BH címre, ha NC
  FA3F  LD   C,01
  FA41  JR   0F          ; ugrás az FA52H címre
  FA43  INC  B
  FA44  INC  C
  FA45  LD   A,41        ; üres utasítás
***
  FA46  LD   B,C
***
  FA47  INC  DE
  FA48  LD   A,(DE)      ; A-ba a szám egy byte-ja
  FA49  ADC  A,A
  FA4A  DAA              ; decimális igazítás
  FA4B  LD   (DE),A      ; a byte visszatöltése
  FA4C  DJNZ F9          ; ugrás az FA47H címre, ha B nem 0
  FA4E  JR   C,F3        ; ugrás az FA43H címre, ha C
  FA50  ADC  HL,HL
  FA52  LD   DE,(171C)   ; DE a bejegyzendö szám alsó byte-jára
                           mutat
  FA56  JR   NZ,EE       ; ugrás az FA46H címre, ha NZ
  FA58  POP  AF          ; AF-be az "erdeti" HL
  FA59  AND  80
  FA5B  OR   49
  FA5D  LD   (IY+08H),A  ; a kitevö byte beírása
  FA60  JP   F734        ; a beírt szám 7 byte-osra kerekítése

Egy numerikus változó értékét tölti a BASIC STACK-be:
  FA63  CALL FC8E        ; van elég szabad memória?
  FA66  LD   BC,0005
  FA69  ADD  HL,BC       ; HL a változó értékére mutat a szimbó-
                           lumtáblán
  FA6A  PUSH IY
  FA6C  POP  DE
  FA6D  DEC  DE          ; DE a BASIC STACK aljára mutat
  FA6E  LDD              ; a kitevö byte átmozgatása
  FA70  XOR  A           ; A-ba 0
```

```
  FA71  LD   (DE),A      ; az "elöcsordulás" byte 0
  FA72  DEC  DE
  FA73  INC  BC
  FA74  LDDR             ; a változó mantisszájának átmozgatása
  FA76  LD   (DE),A      ; a túlcsordulás byte 0
  FA77  LD   A,09        ; szám azonosító byte A-ba
  FA79  DEC  DE
  FA7A  JR   11          ; ugrás az FA8DH címre

String változó értéke a BASIC STACK-be:
  FA7C  INC  HL          ; HL a szimbólumtábla bejegyzésre mutat
  FA7D  LD   C,(HL)      ; C-be a string hossza
  FA7E  LD   B,00
  FA80  CALL FC8E        ; van elég szabad hely?
  FA83  ADD  HL,BC
  FA84  INC  BC
  FA85  PUSH IY
  FA87  POP  DE
  FA88  DEC  DE          ; DE a BASIC STACK aljára mutat
  FA89  LDDR             ; a string átmozgatása
  FA8B  LD   A,01        ; A-ba a string jelzö byte
  FA8D  LD   (DE),A      ; a bejegyzést jelzö byte beírása
  FA8E  PUSH DE
  FA8F  POP  IY
  FA91  RET

A 12. aritmetikai rutin:
  FA92  CALL FC8E        ; van elég szabad hely?
  FA95  PUSH IY
  FA97  POP  DE
  FA98  DEC  DE          ; DE a BASIC STACK aljára mutat
  FA99  LD   HL,0009
  FA9C  LD   C,L
  FA9D  LD   B,H         ; BC-be 9
  FA9E  ADD  HL,DE       ; HL a másolandó szám elejére mutat
  FA9F  LDDR             ; a szám duplikálása
  FAA1  INC  DE
  FAA2  PUSH DE
  FAA3  POP  IY
  FAA5  RET

A BASIC STACK alján lévö string duplikálása:
  FAA6  CALL FC8E        ; van elég szabad hely?
  FAA9  PUSH IY
  FAAB  POP  HL
```

```
  FAAC  DEC  HL          ; HL a BASIC STACK aljára mutat
  FAAD  LD   E,L
  FAAE  LD   D,H         ; DE-be HL
  FAAF  LD   BC,0012     ; BC-be 18
  FAB2  ADD  HL,BC       ; HL a string tetejére mutat
  FAB3  LDDR             ; a string duplikálása
  FAB5  INC  DE
  FAB6  PUSH DE
  FAB7  POP  IY
  FAB9  RET

A programsorban lévö aritmetikai kifejezés eredménye HL-be:
  FABA  CALL FC43        ; a következö program byte
  FABD  JR   05          ; ugrás az FAC4H címre
  FABF  CALL FAC3        ; HL-be az aritmetikai kifejezés ered-
                           nye
  FAC2  EX   DE,HL

A BASIC STACK-ben található szám HL-be:
  FAC3  OR   37          ; itt üres utasítás

Aritmetikai kifejezés értéke HL-be:
***
  FAC4  SCF
***
  FAC5  PUSH BC
  FAC6  PUSH DE
  FAC7  CALL C,F0A7      ; az aritmetikai kifejezés értéke a BA-
                           SIC STACK-be
  FACA  LD   A,(IY+08H)  ; A-ba a szám kitevö byte-ja
  FACD  PUSH AF
  FACE  AND  7F          ; az elöjelbit törlése
  FAD0  SBC  HL,HL
  FAD2  CP   45
  FAD4  JR   NC,3E       ; a szám >= 1E+5, ugrás az FB14H címre
  FAD6  SUB  40
  FAD8  JR   C,2C        ; a szám < 1, ugrás az FB06H címre
  FADA  ADD  A,05
  FADC  LD   C,A
  FADD  LD   B,04
  FADF  LD   A,H
  FAE0  AND  E0
  FAE2  JR   NZ,30       ; ugrás az FB14H címre, ha NZ
  FAE4  ADD  HL,HL
  FAE5  LD   E,L
```

```
  FAE6  LD   D,H
  FAE7  ADD  HL,HL
  FAE8  ADD  HL,HL
  FAE9  ADD  HL,DE        ; HL=10*HL
  FAEA  JR   C,28         ; ugrás az FB14H címre, ha CARRY
  FAEC  CALL F7D2         ; A-ba a megfelelö digit
  FAEF  LD   E,A
  FAF0  LD   D,00         ; DE-be A
  FAF2  ADD  HL,DE
  FAF3  JR   C,1F         ; ugrás az FB14H címre, ha CARRY
  FAF5  INC  B
  FAF6  LD   A,C
  FAF7  CP   B
  FAF8  JR   NZ,E5        ; ugrás az FADFH címre, ha B<>C
  FAFA  POP  AF
  FAFB  PUSH AF
  FAFC  OR   A
  FAFD  CALL M,FC86       ; HL komplementálása, ha M
  FB00  POP  AF
  FB01  XOR  H
  FB02  JP   M,FB14
  FB05  LD   A,F1         ; itt üres utasítás
***
  FB06  POP  AF
***
  FB07  LD   DE,0009
  FB0A  ADD  IY,DE        ; szám felszabadítása a BASIC STACK-böl
  FB0C  LD   A,H
  FB0D  OR   A
  FB0E  EXX
  FB0F  LD   A,B
  FB10  EXX
  FB11  POP  DE
  FB12  POP  BC
  FB13  RET

  FB14  RST  08H
  FB15  DB   04           ; "Bad argument" hibajelzés

Aritmetikai kifejezés értéke A-ba:
  FB16  CALL FC43         ; a következö program byte
  FB19  LD   A,F6         ; itt üres utasítás
  FB1B  SCF
```

```
A BASIC STACK-ben található szám A-ba:
***
  FB1A  OR   37          ; CARRY törlése
***
  FB1C  PUSH HL
  FB1D  CALL FAC5        ; a szám HL-be
  FB20  INC  H
  FB21  DEC  H
  FB22  JP   NZ,FB14     ; rossz argumentum, ha H nem 0
  FB25  LD   A,L         ; A-ba a keresett szám
  FB26  POP  HL
  FB27  RET

A 13. aritmetikai rutin:
  FB28  CALL F767        ; kerekítés 5 byte-os mantisszára
  FB2B  EX   DE,HL
  FB2C  PUSH IY
  FB2E  POP  HL
  FB2F  INC  HL
  FB30  INC  HL          ; HL a tulcsordulás byte fölé mutat
  FB31  LD   BC,0005
  FB34  LDIR             ; a mantissza átmozgatása a szimbólum-
                           táblába
  FB36  INC  HL
  FB37  LDI              ; a kitevö byte átmozgatása
  FB39  JR   1C          ; ugrás az FB47H címre

A BASIC STACK alsó elemének átmozgatása a szimbólumtáblába:
  FB3B  BIT  1,(IX+01H)
  FB3F  JR   NZ,E7       ; a szimbólumtábla elem szám, ugrás az
                           FB28H címre

String mozgatása a BASIC STACK-böl a szimbólumtáblába:
  FB41  LD   C,(HL)      ; C-be a stringváltozó maximális hossza
  FB42  INC  HL
  FB43  PUSH IY
  FB45  POP  DE
  FB46  INC  DE
  FB47  LD   A,(DE)      ; A-ba a string hossza
  FB48  INC  C
  FB49  JR   NZ,01       ; ugrás az FB4CH címre, ha C FFH volt
  FB4B  DEC  C
  FB4C  CP   C
  FB4D  JP   NC,F912     ; hibajelzés, ha a string hosszabb a
                           maximális hossznál
```

```
FB50  LD   C,A
FB51  LD   B,00
FB53  INC  BC
FB54  EX   DE,HL
FB55  LDIR              ; a string átmozgatása
FB57  PUSH HL
FB58  POP  IY
FB5A  RET
```

A 0008-002E RAM területre kerülö blokk:

```
FB5B  DB   E1 7E B7 E5 00
FB60  DB   00 00 00 C8
FB64  DB   E1 C3 5C FD
FB68  DB   C3 00 00 C3
FB6C  DB   82 FB 32 1F
FB70  DB   00 F7 00 C9
FB74  DB   00 00 00 00
FB78  DB   00 00 00 00
FB7C  DB   00 00 00 00
FB80  DB   00 00
```

Az RST 18H rutin :

```
FB82  LD   (171E),HL  ; HL eltárolása
FB85  LD   (1704),A   ; A eltárolása
FB88  LD   HL,(1718)  ; HL-be az RST_18_BUFFER
FB8B  EX   SP,HL
FB8C  LD   A,(HL)     ; A-ba a következö aritmetikai rutin
                        sorszáma
FB8D  INC  HL         ; HL a rákövetkezö rutinsorszámra mutat
FB8E  LD   (1718),HL  ; HL eltárolása az RST_18_BUFFER-ben
FB91  ADD  A,A        ; A-ba a rutinsorszám kétszerese
FB92  PUSH AF         ; ha CARRY=1, az utolsó meghívandó rutin
FB93  ADD  A,C7
FB95  LD   L,A
FB96  LD   H,C0       ; HL-be C0C7H, az RST 18H rutinok ugró-
                        táblájának címe + a rutinsorszám két-
                        szerese
FB98  LD   A,(HL)
FB99  INC  HL
FB9A  LD   H,(HL)
FB9B  LD   L,A        ; HL-be a megfelelö rutin címe
FB9C  LD   (0016),HL  ; az 15H címen található JP utasítás ug-
                        rási címének átírása
FB9F  LD   HL,(171E)  ; HL visszaállítása
```

```
FBA2  LD   A,(1704)    ; A visszaállítása
FBA5  CALL 0015        ; a megfelelö aritmetikai rutin meghívá-
                         sa
FBA8  LD   (1704),A    ; A eltárolása
FBAB  LD   (171E),HL   ; HL eltárolása
FBAE  LD   HL,(1718)   ; HL-be a következö rutinsorszám címe
FBB1  POP  AF
FBB2  JR   NC,D8       ; van még végrehajtandó aritmetikai ru-
                         tin, ugrás az FB8CH címre
FBB4  EX   SP,HL       ; a visszatérési cím az utolsó aritme-
                         tikai rutinsorszám utáni byte-ra mutat
FBB5  LD   (1718),HL
FBB8  LD   HL,(171E)   ; HL visszaállítása
FBBB  LD   A,(1704)    ; A visszaállítása
FBBE  RET

Programbyte vizsgálat:
FBBF  LD   A,(DE)      ; A-ba a vizsgálandó byte
FBC0  INC  B
FBC1  DEC  B
FBC2  RET  NZ          ; ez a byte nem a vizsgált egységhez
                         tartozik, RET
FBC3  CP   FF
FBC5  RET  Z           ; sor vége jel, RET
FBC6  AND  7F
FBC8  CP   20
FBCA  JR   C,09        ; újra definiálható karakter, ugrás az
                         FBD5H címre
FBCC  CP   61
FBCE  RET  C           ; nem kis betü, RET
FBCF  CP   7B
FBD1  RET  NC          ; nem kisbetü, RET
FBD2  AND  DF          ; a kisbetü átalakítása nagybetüvé
FBD4  RET
FBD5  CP   10
FBD7  RET  C           ; 144-nél kisebb kódú karakter, RET
FBD8  CP   19
FBDA  RET  NC          ; nem ékezetes kisbetü, RET
FBDB  AND  EF          ; az ékezetes kisbetüböl nagybetü
FBDD  RET

A BASIC STACK-ben megadott sorszámú programsor megkeresése:
FBDE  CP   02
FBE0  JP   NZ,FD5A     ; nem numerikus változó, hiba
FBE3  CALL FAC3        ; HL-be a programsor sorszáma
```

```
  FBE6  CALL DD44        ; az adott sorszámú sor helyének megha-
                           tározása
  FBE9  RET  NC          ; van ilyen számú sor, RET

  FBEA  RST  08H
  FBEB  DB   02          ; "Line missing" hibajelzés

Az adott BASIC utasításnál van megadott I/O eszköz?
  FBEC  LD   C,20        ; az alap eszköz az EDITOR
  FBEE  CALL FBFD        ; van megadott eszköz?
  FBF1  RET  NZ          ; nincs, RET
  FBF2  CP   FD
  FBF4  JR   Z,4D        ; az eszközkód után ":" van, ugrás az
                           FC43H címre
  FBF6  CP   FE
  FBF8  RET  NC          ; utasítás vége jel következik, RET

  FBF9  RST  08H
  FBFA  DB   01          ; szintaktikai hiba

  FBFB  LD   C,20        ; az alap eszköz az EDITOR
  FBFD  LD   (IX+05H),C  ; az alap eszköz kódjának eltárolása
  FC00  EXX
  FC01  LD   A,B
  FC02  EXX
  FC03  CP   A7          ; (#)
  FC05  RET  NZ          ; a következő byte nem "#", RET
  FC06  CALL FB16        ; A-ba az eszköz száma
  FC09  AND  07          ; az eszköz szám moduló 7
  FC0B  ADD  A,A
  FC0C  ADD  A,A
  FC0D  ADD  A,A
  FC0E  ADD  A,A         ; A-ba az eszköz szám 16-szorosa
  FC0F  LD   (1705),A    ; az eszköz kód eltárolása
  FC12  LD   C,A
  FC13  XOR  A
  FC14  EXX
  FC15  LD   A,B
  FC16  EXX
  FC17  RET

A hibakezelő használta rutin:
  FC18  LD   (IX+05H),20; az I/O eszköz az EDITOR
  FC1C  LD   A,(0B4E)    ; A-ba a PAPER
  FC1F  LD   C,A
```

```
FC20  LD   A,(0B4D)    ; A-ba az INK
FC23  CP   C
FC24  RET  NZ          ; az INK és a PAPER különböző, RET
FC25  INC  A           ; az INK megváltoztatása
FC26  LD   (0B4D),A    ; az új INK beírása
FC29  LD   A,(0B13)    ; A-ba PORT06
FC2C  BIT  1,A
FC2E  RET  Z           ; nem 16 színű mód, RET
FC2F  LD   A,C         ; A-ba a PAPER
FC30  CPL
FC31  LD   (0B4D),A    ; az új INK beírása
FC34  RET
```

A hibakezelés használta rutin:

```
FC35  BIT  3,(IX+00H)
FC39  RET  Z           ; nincs nyitott file, RET
FC3A  RES  3,(IX+00H)  ; az OPEN_FILE_FLAG törlése
FC3E  RST  30H
FC3F  DB   54          ; a CAS_CLOSE_OUT rutin
FC40  RST  30H
FC41  DB   D4          ; a CAS_CLOSE_IN rutin
FC42  RET
```

A soronkövetkező programbyte-ot feldolgozó rutin:

```
FC43  EXX
FC44  LD   A,(HL)      ; A-ba a soronkövetkező programbyte
FC45  LD   B,A
FC46  CP   FE
FC48  JR   NC,30       ; nincs több utasítás az adott sorban,
                         ugrás az FC7AH címre
FC4A  INC  HL
FC4B  CP   C5          ; (INKEY$)
FC4D  JR   Z,2F        ; INKEY$ függvény következik, ugrás az
                         FC7EH címre
FC4F  CP   80
FC51  JR   NC,27       ; utasítás token, ugrás az FC7AH címre
FC53  CP   20
FC55  JR   Z,ED        ; szóköz, ugrás az FC44H címre
FC57  DEC  HL
FC58  LD   B,00
FC5A  CP   22
FC5C  SCF
FC5D  JR   Z,17        ; idézőjel ("), ugrás az FC76H címre
FC5F  SET  1,B
FC61  CP   41
```

```
  FC63  JR    C,11        ; nem betü, ugrás az FC76H címre
  FC65  CP    5B
  FC67  CCF
  FC68  JR    C,0C        ; ugrás az FC76H címre, ha CARRY
  FC6A  CALL  F3D2        ; változónév keresés a szimbólumtáblában
  FC6D  CALL  NC,F40B     ; nincs ilyen nevü változó, új változó
                            bejegyzése a szimbólumtáblába
  FC70  LD    A,C         ; a változó tipusbyte-ja A-ba
  FC71  AND   02
  FC73  OR    01
  FC75  LD    B,A
  FC76  CALL  C,F905      ; ha string vagy szám a programsorban,
                            ugrás
  FC79  LD    A,B
  FC7A  EXX
  FC7B  CP    FD
  FC7D  RET
  FC7E  LD    C,A
  FC7F  LD    B,01
  FC81  JR    F6          ; ugrás az FC79H címre

HL complementálása, ha a legfelsö bitje 1:
  FC83  BIT   7,H
  FC85  RET   Z           ; a legfelsö bit 0, RET
  FC86  DEC   HL          ; HL=HL-1
  FC87  LD    A,L
  FC88  CPL               ; L complementálása
  FC89  LD    L,A
  FC8A  LD    A,H
  FC8B  CPL               ; H complementálása
  FC8C  LD    H,A
  FC8D  RET

Van elég szabad memória?
  FC8E  PUSH  HL
  FC8F  PUSH  DE
  FC90  LD    DE,0100     ; megkívánt szabad memória mérete DE-be
  FC93  CALL  FC99        ; van ennyi szabad hely?
  FC96  POP   DE
  FC97  POP   HL
  FC98  RET
  FC99  LD    HL,(1726)   ; HL-be TOP
  FC9C  ADD   HL,DE
  FC9D  JR    C,12        ; TOP túl magas, hibajelzés
```

```
FC9F  EX   DE,HL        ; DE-be TOP+100H
FCA0  PUSH IY
FCA2  POP  HL           ; HL-be a BASIC STACK alja
FCA3  SBC  HL,DE
FCA5  JR   C,0A         ; nincs elég hely, ugrás az FCB1H címre
FCA7  SBC  HL,HL        ; HL-be 0
FCA9  ADD  HL,SP        ; HL-be SP
FCAA  LD   DE,(0B17)    ; DE-be ST_LIMIT
FCAE  SBC  HL,DE
FCB0  RET  NC           ; a STACK-nek még van hely, RET

FCB1  RST  08H
FCB2  DB   06           ; "No memory" hibajelzés


FCB3  XOR  A            ; A-ba 0
FCB4  LD   L,A
FCB5  LD   H,A          ; HL-be 0
FCB6  LD   A,10
FCB8  DEC  A
FCB9  RET  M            ; A FFH lett, RET
FCBA  ADC  HL,HL
FCBC  SLA  C
FCBE  RL   B            ; BC=2*BC
FCC0  JR   NC,F6        ; ugrás az FCB8H címre, ha NC
FCC2  ADD  HL,DE
FCC3  JR   F3           ; ugrás az FCB8H címre
```

Egy BASIC utasítás végének megkeresése:

```
FCC5  CALL FCD1         ; a következő token megkeresése
FCC8  RET  NC           ; a programsorban nincs több utasítás
FCC9  CP   CA
FCCB  RET  NC           ; a token "elsö rendü" utasítás tokenje
FCCC  CALL FCD5         ; a következő token megkeresése
FCCF  JR   F7           ; ugrás az FCC8H címre
```

A soronkövetkező token megkeresése az adott programsorban:

```
FCD1  EXX
FCD2  PUSH HL
FCD3  EXX
FCD4  POP  HL
FCD5  LD   A,(HL)       ; A-ba a következő programbyte
FCD6  CP   FE
FCD8  RET  NC           ; nincs több utasítás az adott sorban
FCD9  INC  HL
FCDA  CP   FD
```

```
  FCDC  JR   NZ,F7        ; nem a (:) tokenje, ugrás az FCD5H cím-
                            re
  FCDE  LD   A,(HL)       ; A-ba a következö programbyte
  FCDF  INC  HL
  FCE0  CP   20
  FCE2  JR   Z,FA         ; ha szóköz, ugrás az FCDEH címre
  FCE4  DEC  HL           ; HL a következö utasítás elsö byte-já-
                            ra mutat
  FCE5  SCF
  FCE6  RET
```

Megfelelö tipusú bejegyzés keresése a BASIC STACK-ben:

```
  FCE7  LD   B,00
  FCE9  PUSH IY
  FCEB  POP  HL
  FCEC  LD   C,(HL)       ; C-be a következö bejegyzés jelzö byte-
                            ja
  FCED  CP   C            ; A-ban a keresendö jelzö byte
  FCEE  RET  Z            ; megvan, RET
  FCEF  INC  C
  FCF0  DEC  C
  FCF1  SCF
  FCF2  RET  Z            ; kifutott a BASIC STACK-böl, RET
  FCF3  DEC  C
  FCF4  JR   NZ,02        ; a most elért bejegyzés nem string, ug-
                            rás az FCF8H címre
  FCF6  INC  HL
  FCF7  LD   C,(HL)       ; C-be a bejegyzés hossza
  FCF8  INC  C
  FCF9  ADD  HL,BC        ; HL-be a következö bejegyzés címe
  FCFA  PUSH HL
  FCFB  POP  IY
  FCFD  JR   ED           ; ugrás az FCECH címre
```

Egy aritmetikai kifejezéssel megadott memóriacím HL-be:

```
  FCFF  CALL F0A7         ; az aritmetikai kifejezés eredménye a
                            BASIC STACK-be
  FD02  LD   HL,8000
  FD05  PUSH HL
  FD06  CALL FA2B         ; 8000H a BASIC STACK-be számként
  FD09  CALL F493         ; a 0. aritmetikai rutin
  FD0C  CALL FAC3         ; az eredmény HL-be
  FD0F  POP  DE           ; DE-be 8000H
  FD10  ADD  HL,DE        ; HL-be a memóriacím
  FD11  RET
```

```
Az adott SET utasítás rutinja címének meghatározása:
  FD12  LD   C,A          ; a másod utasítás tokenje C-be
  FD13  LD   A,(HL)       ; a SET utasítások ugrótáblája következő
                            byte-ja A-ba
  FD14  OR   A
  FD15  LD   A,C
  FD16  RET  Z            ; a tábla vége, RET
  FD17  SUB  (HL)
  FD18  INC  HL
  FD19  LD   E,(HL)       ; E-be a keresett cím meghatározásához
                            szükséges eltolási cím
  FD1A  INC  HL
  FD1B  JR   NZ,F6        ; nem ez a keresett utasítás, ugrás az
                            FD13H címre
  FD1D  LD   D,A          ; D-be 0
  FD1E  ADD  HL,DE        ; HL-be a keresett rutin címe
  FD1F  CALL FC43         ; a következő programbyte
  FD22  CALL DBAD         ; JP (HL) utasítás
  FD25  SCF
  FD26  RET

String szelet szelethatárainak meghatározása:
  FD27  CALL FC43         ; a következő programbyte
  FD2A  LD   E,01         ; E-ben az alsó határ lesz
  FD2C  JR   Z,08         ; ":" a következő, ugrás az FD36H címre
  FD2E  CALL FD45         ; az alsó határ meghatározása
  FD31  LD   E,D          ; E-be az alsó határ
  FD32  CP   FD
  FD34  JR   NZ,0A        ; ha nem ":" következik (felsö határ
                            nincs megadva), ugrás az FD40H címre
  FD36  CALL FC43         ; a következő programbyte
  FD39  LD   D,FF         ; D-ben lesz a felsö határ
  FD3B  CP   95
  FD3D  CALL NZ,FD45      ; nem ")" következik, a felsö határ meg-
                            határozása
  FD40  LD   A,95
  FD42  JP   FD54         ; ha nem ")" következik, hibajelzés

A szelethatár D-be:
  FD45  CALL FAC4         ; a határt megszabó aritmetikai kifeje-
                            zés értéke HL-be
  FD48  LD   D,00
  FD4A  INC  H
  FD4B  RET  Z            ; ha HL=FFH volt, D=0
  FD4C  DEC  D
```

```
  FD4D  DEC  H
  FD4E  RET  NZ          ; ha H>0, D=FFH
  FD4F  LD   D,L         ; ha H=0, D=L
  FD50  RET

Hibakezelö rutin:
  FD51  CALL FC43        ; a következö programbyte
  FD54  EXX
  FD55  CP   B
  FD56  EXX
  FD57  JP   Z,FC43      ; ha az A-ban lévövel azonos, ugrás
  FD5A  LD   A,01        ; A-ba a szintaktikai hiba kódja
  FD5C  LD   IY,(171A)   ; az elözö utasítás végrehajtása utáni
                           IY visszaállítása
  FD60  CP   F5
  FD62  JP   Z,FFA3      ; STOP hiba, ugrás a STOP rutinra
  FD65  PUSH AF
  FD66  CP   06
  FD68  CALL Z,DCFC      ; ha "No memory" hiba, a BASIC rendszer-
                           változóinak inicializálása
  FD6B  CALL FC35        ; a nyitott file-ok zárása
  FD6E  CALL FC18        ; a hibajelzés kiírás az EDITOR eszközre
                           fog történni
  FD71  CALL FE79        ; az utasítást követö 6 byte kiíratása
  FD74  DB   06 0D 0A
  FD77  DB   '*** '
  FD7B  POP  AF
  FD7C  BIT  7,A
  FD7E  JR   NZ,18       ; operációs rendszer hiba, ugrás az
                           FD98H címre
  FD80  LD   HL,FDC6     ; HL-be a BASIC hibaüzenet tábla címe
  FD83  LD   BC,FFFF
  FD86  INC  BC
  FD87  ADD  HL,BC       ; HL-be a következö hibaüzenet címe
  FD88  LD   C,(HL)      ; C-be a hibaüzenet száma
  FD89  INC  HL
  FD8A  INC  C
  FD8B  DEC  C
  FD8C  JR   Z,04        ; az utolsó hibaüzenet, ugrás az FD92H
                           címre
  FD8E  CP   C
  FD8F  LD   C,(HL)      ; C-be az adott üzenet hossza
  FD90  JR   NZ,F4       ; nem ez a keresett hiba, ugrás az FD86H
                           címre
  FD92  INC  HL          ; HL a hibaüzenet elsö karakterére mutat
```

```
FD93  CALL FEDD        ; a hibaüzenet kiíratása
FD96  JR   18          ; ugrás az FDB0H címre

FD98  LD   L,A         ; L-be a hibakód
FD99  CALL FE79        ; az utasítást követö szöveg kiíratása
FD9C  DB   0D
FD9D  DB   'System error '
FDAA  XOR  A           ; A-ba 0
FDAB  LD   H,A         ; H-ba 0
FDAC  LD   B,A         ; B-be 0
FDAD  CALL FF1B        ; HL tartalmának kiíratása
FDB0  CALL FE79        ; "." , "CEL", "soremelés-kocsivissza"
                         karakterek kiíratása
FDB3  DB   04 2E 0B 0D 0A
FDB8  SET  1,(IX+00H)  ; az "ok" kiíratás engedélyezés
FDBC  LD   HL,(170C)   ; HL-be START
FDBF  XOR  A
FDC0  CALL DD2D        ; a hibás programsor kiíratása
FDC3  JP   E10E        ; az END rutin
```

A BASIC hibaüzenetei:

```
FDC6  DB   01 0F                        ; 1. Not understood
FDC8  DB   4E 6F 74 20 75 6E 64 65
FDD0  DB   72 73 74 6F 6F 64
FDD6  DB   FF

FDD7  DB   02 06                        ; 2. Line missing
FDD8  DB   4C 69 6E 65 93
FDDE  DB   FF

FDDF  DB   03 04                        ; 3. Argument missing
FDE1  DB   41 92 93
FDE4  DB   FF

FDE5  DB   04 04                        ; 4. Bad argument
FDE7  DB   91 61 92
FDEA  DB   FF

FDEB  DB   05 0B                        ; 5. Bad subscript
FDED  DB   91 73 75 62 73 63 72 69
FDF5  DB   70 74
FDF7  DB   FF
```

```
FDF8  DB   06 08                          ; 6. No memory
FDFA  DB   90 6D 65 6D 6F 72 79
FE01  DB   FF

FE02  DB   07 03                          ; 7. No DATA
FE04  DB   90 FB
FE06  DB   FF

FE07  DB   08 03                          ; 8. No FOR
FE09  DB   90 F2
FE0B  DB   FF

FE0C  DB   09 03                          ; 9. No GOSUB
FE0E  DB   90 F0
FE10  DB   FF

FE11  DB   0A 03                          ; 10. Cannot CONTINUE
FE13  DB   8F F8
FE15  DB   FF

FE16  DB   0B 0D                          ; 11. Cannot divide by 0
FE18  DB   8F 64 69 76 69 64 65 20
FE20  DB   62 79 20 30
FE24  DB   FF

FE25  DB   0C 03                          ; 12. Cannot READ
FE27  DB   8F DB
FE29  DB   FF

FE2A  DB   0D 09                          ; 13. Overflow
FE2C  DB   4F 76 65 72 66 6C 6F 77
FE34  DB   FF

FE35  DB   0E 0E                          ; 14. Type mismatch
FE37  DB   54 79 70 65 20 6D 69 73
FE3F  DB   6D 61 74 63 68
FE44  DB   FF

FE45  DB   0F 18                          ; 15. Variable declared
FE47  DB   56 61 72 69 61 62 6C 65        ;     twice
FE4F  DB   20 64 65 63 6C 61 72 65
FE57  DB   64 20 74 77 69 63 65
FE5E  DB   FF

FE5F  DB   10 06                          ; 16. Bad file
```

```
  FE61  DB    91 66 69 6C 65
  FE66  DB    FF

  FE67  DB    00 0F                       ; 0. BASIC corrupted
  FE69  DB    42 41 53 49 43 20 63 6F
  FE71  DB    72 72 75 70 74 65 64
  FE78  DB    FF
```

A STACK-ben tárolt címen található string kiíratása, és a string utáni utasításra ugrás:

```
  FE79  EX    SP,HL         ; HL-be a string címe
  FE7A  CALL  FE7F          ; a string kiíratása
  FE7D  EX    SP,HL         ; a string utáni cím a visszatérési cím
  FE7E  RET
```

A HL által mutatott string kiíratása:

```
  FE7F  LD    A,(HL)        ; A-ba a string hossza
  FE80  INC   HL
  FE81  OR    A
  FE82  RET   Z             ; üres string, RET
  FE83  PUSH  BC
  FE84  LD    B,A           ; B-be a karakter hossza
  FE85  LD    A,(HL)        ; A-ba a kiírandó karakter
  FE86  INC   HL
  FE87  CALL  FE9A          ; a karakter kiíratása
  FE8A  DJNZ  F9            ; ugrás az FE85H címre, ha B nem 0
  FE8C  POP   BC
  FE8D  RET
```

A 0BH, 0DH, 0AH karakterek kiíratása:

```
  FE8E  LD    A,0B
  FE90  CALL  Z,FE9A        ; a CEL karakter kiíratása, ha Z
  FE93  LD    A,0D
  FE95  CALL  FE9A          ; 0DH kódú karakter kiíratása
  FE98  LD    A,0A
```

Az A-ban lévő karakter kiíratása:

```
  FE9A  PUSH  AF
  FE9B  PUSH  BC
  FE9C  PUSH  DE
  FE9D  LD    C,A           ; a kiírandó karakter C-be
  FE9E  CALL  FEA6          ; a kiíratő operációs rendszer rutin
  FEA1  RST   10H           ; hibakezelő rutin
  FEA2  POP   DE
```

```
  FEA3  POP  BC
  FEA4  POP  AF
  FEA5  RET

A megfelelö CHOUT rutin meghívása:
  FEA6  LD   DE,007F
  FEA9  LD   A,(1705)    ; A-ba az I/O eszköz kódja
  FEAC  OR   01          ; 1. eszköz rutin
  FEAE  OR   D
  FEAF  AND  E           ; output rutin
  FEB0  JP   001B        ; az eszköz rutin meghívása

A karakter kiíratö rutin az EXT. ROM-ból híva:
  FEB3  CALL FE9A        ; a karakter kiíratása
  FEB6  POP  HL          ; HL-be az EXT. ROM-beli visszatérési
                           cím
  FEB7  JP   FFF0        ; vissza az EXT. ROM-ba

Szám kiíratása a BASIC STACK-böl:
  FEBA  BIT  7,(IY+08H)
  FEBE  CALL Z,FEC7      ; ha pozitív, egy szóköz kiíratása
  FEC1  CALL F80E        ; a számból string, melyre HL mutat
  FEC4  CALL FE7F        ; a string kiíratása
  FEC7  LD   A,20        ; A-ba a szóköz kódja
  FEC9  JR   CF          ; ugrás az FE9AH címre

Tokeneket is tartalmazó karakterlánc kiíratása:
  FECB  ADD  A,A         ; A=2*A
  FECC  CP   40
  FECE  RRA              ; ha A-ban újradefiniálható karakter
                           byte-ja (a kód-128) volt, most a ka-
                           rakter kódja lesz A-ban
  FECF  CALL FE9A        ; karakter kiíratása
  FED2  CP   22
  FED4  JR   NZ,09       ; ha a karakter idézöjel (") volt, ugrás
                           az FEDFH címre
  FED6  CP   C
  FED7  JR   Z,02        ; ha C=A, ugrás az FEDCH címre
  FED9  LD   B,C
  FEDA  LD   C,78        ; itt üres utasítás
***
  FEDB  LD   A,B
***
  FEDC  LD   C,AF        ; itt üres utasítás
```

```
***
  FEDD  XOR  A
***
  FEDE  LD   C,A
  FEDF  LD   A,(HL)      ; A-ba a következő karakter
  FEE0  INC  HL
  FEE1  INC  A
  FEE2  RET  Z           ; ha A=FFH (a karakterlánc végét jelzi),
                           a kiíratás vége
  FEE3  DEC  A
  FEE4  JP   P,FECB      ; karakter kiíratása, ha P
  FEE7  PUSH HL
  FEE8  INC  C
  FEE9  DEC  C
  FEEA  PUSH AF
  FEEB  CP   FB          ; (DATA)
  FEED  JR   Z,04        ; ha DATA token, ugrás az FEF3H címre
  FEEF  CP   FC
  FEF1  JR   C,02        ; ha A<FCH, ugrás az FEF5H címre
  FEF3  LD   C,FF
  FEF5  CP   FD          ; (:)
  FEF7  JR   NZ,03       ; nem ":" karakter, ugrás az FEFCH címre
  FEF9  POP  AF
  FEFA  INC  C
  FEFB  PUSH AF
  FEFC  POP  AF
  FEFD  JR   NZ,12       ; az A-ban lévő kód nem token, ugrás az
                           FF11H címre
  FEFF  CPL              ; A-ba a tokenhez tartozó string sorszá-
                           ma BASIC kulcsszó táblázatban
  FF00  LD   HL,DE6D     ; HL-be a BASIC kulcsszó tábla címe
  FF03  BIT  7,(HL)
  FF05  INC  HL
  FF06  JR   Z,FB        ; HL nem a kulcsszó elejére mutat, ugrás
                           az FF03H címre
  FF08  DEC  A
  FF09  JR   NZ,F8       ; ez még nem a keresett kulcsszó, ugrás
                           FF03H címre

A kulcsszó kiíratása:
  FF0B  LD   A,(HL)      ; A-ba a kulcsszó következő karaktere
  FF0C  INC  HL
  FF0D  BIT  7,A
  FF0F  RES  7,A         ; a felső bit 0-ra állítása
  FF11  CALL FE9A        ; a karakter kiíratása
```

```
  FF14  JR   Z,F5        ; még nem az utolsó karakter, ugrás az
                           FF0BH címre
  FF16  POP  HL
  FF17  JR   C6          ; ugrás az FEDFH címre
```

A HL-ben lévő bináris szám kiíratása (nem lehet nagyobb 9999-nél):

```
  FF19  LD   B,FF
  FF1B  SBC  A,A
  FF1C  AND  20
  FF1E  LD   C,A
  FF1F  PUSH HL
  FF20  LD   HL,FF47     ; HL-be a Decimális tábla címe
  FF23  LD   E,(HL)
  FF24  INC  HL
  FF25  LD   D,(HL)      ; DE-be a Decimális tábla soronkövetkező
                           eleme
  FF26  INC  HL
  FF27  EX   SP,HL       ; HL-be a kiírandó szám
  FF28  XOR  A           ; A-ba 0
  FF29  SBC  HL,DE
  FF2B  INC  A
  FF2C  JR   NC,FB       ; ugrás az FF29H címre, ha a HL nagyobb
                           volt DE-nél
  FF2E  ADD  HL,DE       ; a maradék visszaállítása
  FF2F  DEC  A           ; A-ban az adott nagyságrendhez tartozó
                           digit binárisan
  FF30  JR   Z,07        ; ha a digit 0, a szám elején nem íródik
                           ki
  FF32  LD   C,30
  FF34  ADD  A,C         ; A-ba a digit kódja
  FF35  CALL FE9A        ; a számjegy kiíratása
  FF38  LD   A,C
  FF39  XOR  C
  FF3A  CALL NZ,FE9A     ; a szám elején 0 helyett szóköz, külön-
                           ben 0 kiíratása
  FF3D  EX   SP,HL       ; HL-be a Decimális tábla következő ele-
                           mének címe
  FF3E  DEC  E
  FF3F  JR   NZ,E2       ; még nem az 1-es helyiérték volt, ugrás
                           az FF23H címre
  FF41  POP  HL
  FF42  LD   A,C
  FF43  AND  B
  FF44  JR   NZ,81       ; ugrás az FEC7H címre, ha NZ
  FF46  RET
```

Decimális helyiértékek táblázata:

```
  FF47  DW   03E8    ; dec 1000
  FF49  DW   0064    ; dec 100
  FF4B  DW   000A    ; dec 10
  FF4D  DW   0001    ; dec 1
```

A BASIC interpreter input rutinja:

```
  FF4F  LD   HL,1831     ; HL-be az input BUFFER címe
  FF52  PUSH HL
  FF53  LD   B,00
  FF55  PUSH BC
  FF56  LD   DE,80FF
  FF59  CALL FEA9        ; egy sor beolvasása az input eszközről
  FF5C  POP  DE
  FF5D  LD   B,D
  FF5E  JR   Z,12        ; ugrás az FF72H címre, ha Z
  FF60  POP  HL          ; HL-be a BUFFER címe
  FF61  LD   B,00
  FF63  LD   (HL),B      ; a BUFFER első byte-ja 0
  FF64  INC  HL
  FF65  LD   (HL),FF
  FF67  DEC  HL
  FF68  CP   F5
  FF6A  JR   Z,2F        ; CTRL+ESC-et nyomtak, ugrás az FF9BH
                           címre
  FF6C  CP   EC
  FF6E  JR   Z,2B        ; file vége jelzés, ugrás az FF9BH címre
  FF70  OR   A
  FF71  RST  10H         ; hibakezelés
  FF72  LD   A,(0B16)    ; A-ba STOP_FLAG
  FF75  OR   A
  FF76  LD   A,F5        ; A-ba a STOP hiba kódja
  FF78  JR   NZ,E6       ; ugrás az FF60H címre, ha nem STOP
  FF7A  LD   A,C         ; A-ba a meghívott operációs rendszer
                           rutin által visszaadott karakter
  FF7B  CP   0D          ; (RET)
  FF7D  JR   Z,13        ; a bekezdés utolsó karaktere, ugrás az
                           FF92H címre
  FF7F  CP   20
  FF81  JR   C,D2        ; nem kiírandó karakter, ugrás az FF55H
                           címre
  FF83  SUB  80
  FF85  JR   C,05        ; nem definiálható karakter, ugrás az
                           FF8CH címre
```

```
  FF87  CP   20
  FF89  JR   NC,01        ; definiálható karakter, ugrás az FF8CH
                            címre
  FF8B  LD   C,A
  FF8C  LD   A,B
  FF8D  CP   FB
  FF8F  JR   NC,C4        ; ugrás az FF55H címre, ha túl hosszú a
                            bekezdés
  FF91  INC  B
  FF92  INC  HL
  FF93  LD   (HL),C       ; a bekezdés következö karaktere a BUF-
                            FER-be
  FF94  JR   NZ,BF        ; ugrás az FF55H címre, ha nem a bekez-
                            dés utolsó karaktere
  FF96  LD   (HL),FF      ; a BUFFER-be tett szöveg vége
  FF98  XOR  A
  FF99  POP  HL           ; HL-be a BUFFER címe
  FF9A  LD   (HL),B       ; a BUFFER-be tett szöveg hossza
  FF9B  OR   A
  FF9C  RET

A STOP_FLAG vizsgálata:
  FF9D  LD   A,(0B16)     ; A-ba a STOP_FLAG
  FFA0  OR   A
  FFA1  RET  Z            ; nincs STOP, RET
  FFA2  EXX

A STOP  utasítás rutinja:
  FFA3  EXX
  FFA4  XOR  A            ; A-ba 0
  FFA5  LD   (0B16),A     ; a STOP_FLAG törlése
  FFA8  CALL FC35         ; a nyitott file-ok zárása
  FFAB  CALL FC18         ; az output eszköz, INK és PAPER beállí-
                            tása
  FFAE  CALL FE79         ; az utasítást követö string kiíratása
  FFB1  DB   06 0D 0A
  FFB4  DB   'STOP'
  FFB8  BIT  2,(IX+00H)
  FFBC  JR   Z,22         ; parancs mód, ugrás az FFE0H címre
  FFBE  LD   (1710),HL    ; a STOP-t követö utasítás címének eltá-
                            rolása
  FFC1  LD   HL,(170C)    ; HL-be START
  FFC4  LD   (170E),HL    ; a STOP-t tartalmazó programsor címe
  FFC7  CALL FE79         ; az utasítást követö string kiíratása
  FFCA  DB   09
```

```
  FFCB  DB    ' at line '
  FFD4  LD    HL,(170C)   ; HL-be START
  FFD7  INC   HL
  FFD8  LD    E,(HL)
  FFD9  INC   HL
  FFDA  LD    D,(HL)      ; DE-be az adott programsor sorszáma
  FFDB  EX    DE,HL
  FFDC  OR    A
  FFDD  CALL  FF19        ; a sorszám kiíratása
  FFE0  XOR   A
  FFE1  CALL  FE8E        ; a CEL, soremelés, kocsivissza karakte-
                            rek kiíratása
  FFE4  JP    DADA        ; a STOP rutin vége

A ROM-ba mutató címek átalakítása a video-RAM-ba mutatóra:
  FFE7  LD    A,H
  FFE8  CP    C0
  FFEA  RET   C           ; HL nem a P3 lapra mutató címet tartal-
                            maz, RET
  FFEB  RES   6,H         ; HL=HL-4000H
  FFED  LD    A,50        ; A-ba az U U V S lapkiosztást meghatá-
                            rozó byte
  FFEF  RET

Az EXTENSION ROM-ba átvivö rutin:
  FFF0  PUSH  AF
  FFF1  LD    A,F0
  FFF3  LD    (0003),A
  FFF6  OUT   02,A        ; memórialapozás: U U U E lapkiosztás

Az EXTENSION ROM-ból SYSTEM ROM rutin hívás:
  FFF8  POP   AF
  FFF9  JP    (HL)        ; ugrás a HL-ben található cím ü rutinra

  FFFA  DB    79 FE 0D
  FFFD  DB    28 13 FE
```

II. EXTENSION ROM

A rendszer ROM bövítés

```
  F000  LD   A,D0
  F002  OUT  02,A        ; memórialapozás: U U V E lapkiosztás
  F004  LD   A,(0B21)    ; A-ba a WARM_FLAG
  F007  OR   A
  F008  JR   NZ,04       ; meleg RESET van folyamatban, ugrás az
                           F00EH címre
  F00A  EX   AF,AF'
  F00B  LD   (0B1B),A    ; P3RAM rendszerváltozó értékadása
  F00E  LD   HL,FB6B     ; az újradefiniálható karakterek karak-
                           tergenerátorának címe
  F011  LD   DE,0740     ; a karaktergenerátor RAM-beli címe
  F014  LD   BC,0140     ; a karaktergenerátor mérete
  F017  LDIR             ; az újradefiniálható karakterek karak-
                           terggenerátorának átmozgatása a ROM-
                           ból a RAM-ba

A definiálható karakterek karaktergenerátorának törlése:
  F019  LD   A,40        ; 64 karakter definiálható
  F01B  PUSH HL          ; HL-ben FCABH van
  F01C  LD   C,0A        ; egy karaktert 10 byte ír le
  F01E  LDIR             ; 10 byte 00H átmozgatása
  F020  POP  HL          ; HL-be FCABH
  F021  DEC  A
  F022  JR   NZ,F7       ; van még törlendö terület, ugrás az
                           F01BH címre

Az IN_TABLE és az OUT_TABLE feltöltése a ROM-ból:
  F024  LD   HL,FB0E     ; az áttöltendö byte-ok kezdete
  F027  LD   DE,0B00     ; DE-be az IN_TABLE címe
  F02A  LD   C,10
  F02C  LDIR             ; 16 byte átmozgatása a RAM-ba

Az OS_ENTRY és az IT_ENTRY területek feltöltése a ROM-ból:
  F02E  LD   DE,0030
  F031  LD   C,10
  F033  LDIR

Az INFUNC, az OUTFUNC és az INT_EXIT területek feltöltése a
ROM-ból:
  F035  LD   DE,0B23
```

```
  F038  LD    C,26
  F03A  LDIR

Az INT_DES rendszerváltozó inicializálása:
  F03C  LD    A,FC
  F03E  LD    (0B10),A

A PORT06 és az IRQ_STAT rendszerváltozók inicializálása:
  F041  LD    A,80
  F043  LD    (0B13),A    ; PORT06 írása
  F046  RLCA              ; A-ba 01H
  F047  LD    (0B1F),A    ; IRQ_STAT írása

Az ST_LIMIT beállítása:
  F04A  LD    HL,0F10     ; HL-be ST_LIMIT értéke
  F04D  LD    (0B17),HL   ; ST_LIMIT eltárolása

A csatolókártya azonosítók táblázatának kitöltése:
  F050  IN    A,5A        ; a csatolókártyákon bekötött 2 bites a-
                            zonosítók beolvasása A-ba
  F052  LD    BC,0403     ; B-be 04, C-be 03
  F055  LD    HL,0040     ; HL-be az azonosítók táblájának kezdete
  F058  LD    D,A
  F059  LD    A,04
  F05B  SUB   B           ; A-ba a csatlakozó száma
  F05C  RRCA
  F05D  RRCA
  F05E  OUT   03,A        ; a csatolókártya memória belapozása a
                            P3 lapra
  F060  LD    A,D
  F061  AND   C
  F062  CP    03
  F064  PUSH  HL
  F065  EXX
  F066  JR    Z,11        ; azonosítatlan kártya, ugrás az F079H
                            címre
  F068  LD    HL,FB5C     ; az "RS232" azonosítónév címe
  F06B  OR    A
  F06C  JR    Z,1B        ; soros vonal az adott csatlakozóban,
                            ugrás az F089H címre
  F06E  LD    HL,FB58     ; a "VGB" azonosítónév címe
  F071  DEC   A
  F072  JR    Z,15        ; 01 a bekötött azonosító, ugrás az
                            F089H címre
  F074  LD    HL,FB62     ; a "DISK" azonosítónév címe
```

```
  F077  JR   10          ; ugrás az F089H címre
  F079  LD   HL,C000     ; HL-be a csatolókártya memória kezdete
  F07C  LD   DE,FB54     ; DE-be a "MOPS" azonosító címe

A csatolókártya memória "MOPS" azonosítóval kezdödik?
  F07F  LD   B,04
  F081  LD   A,(DE)      ; A-ba az azonosító következö byte-ja
  F082  CP   (HL)
  F083  JR   NZ,0D       ; a csatolókártya memória nem "MOPS" a-
                           zonosítóval kezdödik, ugrás az F092H
                           címre
  F085  INC  DE
  F086  INC  HL
  F087  DJNZ F8          ; nincs vége az összehasonlításnak, ug-
                           rás az F081H címre
  F089  LD   C,(HL)      ; C-be az azonosítónév hossza
  F08A  INC  C
  F08B  LD   B,00
  F08D  POP  DE
  F08E  LDIR             ; az azonosítónév betöltése az azonosí-
                           ták táblázatába
  F090  JR   03          ; ugrás az F095H címre
  F092  POP  HL
  F093  LD   (HL),FF     ; azonosítatlan kártya az adott csatla-
                           kozóban
  F095  EXX
  F096  LD   A,D
  F097  RRCA
  F098  RRCA             ; A alsó két bitjébe a következö kártya-
                           helyre bedugott kártya bekötött azono-
                           sítója
  F099  LD   DE,0030
  F09C  ADD  HL,DE       ; HL-be a kártyaazonosítók táblája kö-
                           vetkezö elemének címe
  F09D  DJNZ B9          ; ugrás az F058H címre, ha még van vizs-
                           gálatlan csatlakozóhely

A csatlakozókhoz tartozó egységszám meghatározása:
  F09F  LD   B,04
  F0A1  LD   D,A
  F0A2  AND  C
  F0A3  JR   NZ,0B       ; ugrás az F0B0H címre, ha nem RS232
  F0A5  LD   A,04
  F0A7  SUB  B
  F0A8  LD   (0B07),A    ; csatolókártya input kijelölés
```

```
F0AB  LD   (0B0F),A   ; csatolókártya output kijelölés
F0AE  JR   07         ; ugrás az F0B7H címre
F0B0  LD   A,D
F0B1  RRCA
F0B2  RRCA
F0B3  DJNZ EC         ; ugrás az F0A1H címre, ha B nem 0
F0B5  LD   A,FF
F0B7  LD   (0B1C),A   ; EX_DEF írása

Az egységszám byte-ok törlése:
F0BA  LD   HL,0047    ; HL-be az elsö egységszám byte címe
F0BD  LD   DE,0030
F0C0  LD   B,04
F0C2  XOR  A          ; A-ba 0
F0C3  LD   (HL),A     ; az egységszám byte törlése
F0C4  ADD  HL,DE      ; HL-be a következö egységszám byte címe
F0C5  DJNZ FC         ; ugrás az F0C3H címre, ha B nem 0

F0C7  LD   E,40
F0C9  LD   HL,0070
F0CC  PUSH DE
F0CD  POP  IX         ; IX egy csatoló azonosító elejére mutat
F0CF  PUSH HL
F0D0  POP  IY         ; IY az IX által címzett azonosítónál
                        magasabb sorszámú csatlakozó helyhez
                        tartozó azonosítóra mutat

A két csatlakozóban azonos kártyák vannak?
F0D2  LD   B,(HL)
F0D3  INC  B
F0D4  LD   A,(DE)
F0D5  CP   (HL)
F0D6  JR   NZ,07      ; a két kártya különbözö, ugrás az F0DFH
                        címre
F0D8  INC  HL
F0D9  INC  DE
F0DA  DJNZ F8

F0DC  INC  (IY+07H)   ; az egységszám növelése, ha azonosak a
                        csatolókártyák
F0DF  PUSH IY
F0E1  POP  HL
F0E2  LD   DE,0030
F0E5  ADD  HL,DE      ; HL-be a következö azonosítóhely címe
F0E6  BIT  0,H
```

```
F0E8  JR   Z,0C        ; HL-ben jó a cím, ugrás az F0F6H címre
F0EA  PUSH IX
F0EC  POP  HL
F0ED  ADD  HL,DE       ; HL-be a következö azonosítóhely címe
F0EE  BIT  0,H
F0F0  JR   NZ,09       ; jó cím HL-ben, ugrás az F0FBH címre
F0F2  PUSH HL
F0F3  POP  IX
F0F5  ADD  HL,DE       ; HL-be a következö azonosítóhely címe
F0F6  PUSH IX
F0F8  POP  DE
F0F9  JR   D4          ; ugrás az F0CFH címre

A megfelelö csatolókártyák inicializáló rutinjainak meghívása:
F0FB  LD   A,F0
F0FD  OUT  02,A        ; memórialapozás: U U U E lapkiosztás
F0FF  CALL FFE8        ; a soros vonal inicializálása
F102  LD   IX,0040     ; IX-be az elsö azonosító címe
F106  LD   B,04
F108  IN   A,5A        ; A-ba a csatolókártyák bekötött 2 bites
                         azonosítói
F10A  LD   C,A
F10B  LD   A,04
F10D  SUB  B
F10E  RRCA
F10F  RRCA
F110  OUT  03,A        ; a csatolókártya memória belapozása
F112  LD   A,(IX+00H)  ; az azonosító elsö byte-ja A-ba
F115  INC  A
F116  JR   Z,15        ; azonosítatlan kártya az adott csatla-
                         kozóban, ugrás az F12DH címre
F118  LD   A,C
F119  AND  03
F11B  JR   Z,10        ; soros vonal az adott csatlakozóban,
                         ugrás az F12DH címre
F11D  LD   HL,(C00B)   ; HL-be a kártyán lévö inicializáló ru-
                         tin címe
F120  PUSH BC
F121  PUSH IX
F123  CALL FFF9        ; JP (HL)
F126  POP  IX
F128  POP  BC
F129  LD   A,F0
F12B  OUT  02,A        ; memórialapozás: U U U E lapkiosztás
F12D  LD   DE,0030
```

```
  F130  ADD  IX,DE       ; IX-be a következő azonosító címe
  F132  LD   A,C
  F133  RRCA
  F134  RRCA
  F135  DJNZ D3          ; ugrás az F10AH címre, ha B nem 0
  F137  LD   HL,C2D3     ; SYS. ROM cím HL-be
  F13A  JP   FFF0        ; ugrás a SYS. ROM-ba

A rendszer inicializálás része:
  F13D  EX   AF,AF'
  F13E  LD   A,D0
  F140  OUT  02,A        ; memórialapozás: U U V E lapkiosztás
  F142  LD   DE,FB2E     ; DE-be a RAM terület ROM-beli címe
  F145  LD   HL,0B23     ; HL-be a RAM terület címe
  F148  LD   BC,001E     ; az összehasonlítandó terület mérete
  F14B  LD   A,(DE)
  F14C  INC  DE
  F14D  CPI
  F14F  JR   NZ,0B       ; az adott RAM terület hibás, ugrás az
                           F15CH címre
  F151  JP   PE,F14B     ; nincs vége az összehasonlításnak, ug-
                           rás
  F154  LD   A,(0B21)    ; A-ba WARM_FLAG
  F157  OR   A
  F158  LD   A,FF
  F15A  JR   Z,01        ; ha nem meleg RESET volt, ugrás az
                           F15DH címre
  F15C  XOR  A           ; A-ba 0
  F15D  LD   (0B21),A    ; a WARM_FLAG írása
  F160  EX   AF,AF'
  F161  OUT  02,A        ; memórialapozás
  F163  JP   023A        ; ugrás vissza a SYS. ROM-ba

Az RST 30H rutin része, ha bővítökártya rutin hívás:
  F166  LD   L,B
  F167  LD   H,00
  F169  ADD  HL,DE
  F16A  JR   05          ; ugrás az F171H címre
  F16C  LD   HL,0007
  F16F  ADD  HL,DE
  F170  LD   A,(HL)      ; A-ba a csatolókártya IO kijelölés byte
  F171  CP   04
  F173  JR   NC,64       ; ha A>3, ugrás az F1D9H címre
  F175  EXX
  F176  LD   HL,0040     ; HL-be az első csatolókártya azonosító
```

```
                                címe
  F179  LD    DE,0030
  F17C  LD    B,A
  F17D  RRCA
  F17E  RRCA
  F17F  LD    (0B11),A    ; a PORT03 rendszerváltozó írása
  F182  OUT   03,A        ; a csatolókártya memória belapozása
  F184  LD    A,B
  F185  OR    A
  F186  JR    Z,03        ; az elsö (0.) csatlakozóhely, ugrás az
                            F18BH címre
  F188  ADD   HL,DE       ; HL-be a következö azonosító címe
  F189  DJNZ  FD          ; ugrás az F188H címre, ha B nem 0
  F18B  LD    A,(HL)      ; A-ba az azonosító elsö byte-ja
  F18C  INC   A
  F18D  JR    Z,49        ; azonosítatlan bövítökártya, ugrás az
                            F1D8H címre

RS232 rutin a meghívott?
  F18F  LD    C,(HL)
  F190  INC   HL
  F191  LD    DE,FB5D     ; DE-be az "RS232" azonosítónév címe
  F194  LD    A,(DE)
  F195  INC   DE
  F196  CPI
  F198  JR    NZ,09       ; nem soros vonal rutin, ugrás az F1A3H
                            címre
  F19A  JP    PE,F194
  F19D  EXX
  F19E  LD    HL,F291     ; HL-be a soros vonal rutinok ugrótábla-
                            jának a címe
  F1A1  JR    04          ; ugrás az F1A7H címre
  F1A3  EXX
  F1A4  LD    HL,C00D     ; HL-be a csatolókártyán található rutin
                            ugrótábla címe
  F1A7  LD    A,(HL)      ; A-ba a rutinok száma
  F1A8  DEC   A
  F1A9  CP    C
  F1AA  JR    C,34        ; a meghívandó rutin nem létezik, ugrás
                            az F1E0H címre
  F1AC  INC   HL
  F1AD  EX    DE,HL
  F1AE  LD    L,C
  F1AF  LD    H,00
  F1B1  ADD   HL,HL
```

```
  F1B2  ADD  HL,DE       ; HL-be a meghívandó rutin címének címe
  F1B3  LD   E,(HL)
  F1B4  INC  HL
  F1B5  LD   D,(HL)      ; DE-be a meghívandó rutin címe
  F1B6  EX   DE,HL
  F1B7  LD   A,(0B11)    ; A-ba PORT03
  F1BA  RLCA
  F1BB  RLCA
  F1BC  AND  03
  F1BE  LD   IX,0048     ; IX-be az első (0.) csatlakozóhelyhez
                           tartozó azonosító terület RAM-jának a
                           címe
  F1C2  LD   DE,0030
  F1C5  JR   Z,05        ; a 0. csatlakozóban van, ugrás az F1CCH
                           címre
  F1C7  LD   B,A
  F1C8  ADD  IX,DE       ; IX-be a következő csatlakozóhoz tarto-
                           zó RAM terület címe
  F1CA  DJNZ FC          ; B nem 0, ugrás az F1C8H címre
  F1CC  EX   AF,AF'
  F1CD  POP  DE
  F1CE  POP  BC
  F1CF  CALL FFF9        ; JP (HL)
  F1D2  LD   HL,C3FC     ; SYS. ROM cím HL-be
  F1D5  JP   FFF0        ; ugrás a SYS. ROM-ba

Hibás rutinhívás kezelése:
  F1D8  EXX
  F1D9  LD   A,(0B1C)    ; A-ba EX_DEF
  F1DC  LD   (HL),A      ; eredeti hozzárendelés visszaállítása
  F1DD  LD   A,FE        ; hibakód A-ba
  F1DF  LD   DE,FF3E     ; itt üres utasítás
***
  F1E0  LD   A,FF        ; A-ba a hibakód
***
  F1E2  POP  DE
  F1E3  POP  BC
  F1E4  JR   EC          ; ugrás az F1D2H címre

A funkcióosztályokhoz a saját eszközök visszarendelése hibás
bővítőkártya rutinhívás esetén :
  F1E6  RLCA
  F1E7  LD   HL,0B00     ; HL-be az IN_TABLE címe
  F1EA  LD   DE,FB0E     ; az IN_TABLE elemeinek címe a ROM-ban
  F1ED  LD   IX,0B07     ; az IN_TABLE vége
```

```
  F1F1  JR   C,0A        ; ugrás az F1FDH címre, ha input rutin
  F1F3  LD   HL,0B08     ; HL-be az OUT_TABLE címe
  F1F6  LD   DE,FB16     ; az OUT_TABLE elemeinek címe a ROM-ban
  F1F9  LD   IX,0B0F     ; az OUT_TABLE vége
  F1FD  RLCA
  F1FE  RLCA
  F1FF  RLCA
  F200  AND  07          ; A-ban az eszköz száma
  F202  CP   07
  F204  JR   Z,12        ; ugrás az F218H címre, ha Z
  F206  LD   C,A
  F207  LD   B,00
  F209  ADD  HL,BC       ; HL-be a funkcióosztály címe
  F20A  EX   DE,HL
  F20B  ADD  HL,BC       ; HL-be a saját eszköz kódjának címe
  F20C  EX   DE,HL
  F20D  LD   A,(HL)      ; A-ba az adott funkcióosztályhoz ren-
                           delt eszköz kódja
  F20E  CP   06
  F210  CALL Z,F21E      ; ha az eszköz a csatolókártya, ugrás
  F213  RLCA
  F214  JR   C,02        ; ugrás az F218H címre, ha CARRY
Eszköz visszarendelés:
  F216  LD   A,(DE)
  F217  LD   (HL),A
  F218  LD   HL,C40E     ; SYS. ROM-beli rutin cím
  F21B  JP   FFF0        ; ugrás a SYS. ROM-ba
Csatolókártya IO kijelölés visszaállítása:
  F21E  EX   AF,AF'
  F21F  LD   A,(0B1C)    ; A-ba EX_DEF
  F222  LD   (IX+00H),A
  F225  EX   AF,AF'
  F226  RET

A megszakítás kezelö rutin része, ha a megszakítást csatolókár-
tya kérte:
  F227  LD   B,04
  F229  RR   C
  F22B  JR   C,13        ; ugrás az F240H címre, ha nem ez a csa-
                           toló kért megszakítást
  F22D  PUSH BC
  F22E  LD   A,04
  F230  SUB  B           ; A-ba a csatlakozó hely száma
  F231  RRCA
  F232  RRCA
```

```
F233  OUT  03,A        ; a bövítökártya memória belapozása
F235  LD   HL,(C00E)   ; HL-be a kártya IT rutinjának címe
F238  CALL FFF9        ; ugrás az IT rutinra

F23B  LD   A,F0
F23D  OUT  02,A        ; memórialapozás: U U U E lakiosztás
F23F  POP  BC
F240  DJNZ E7          ; ugrás az F229H címre, ha B nem 00
F242  LD   A,(0B11)    ; A-ba PORT03
F245  OUT  03,A        ; a bövítökártya memória belapozása
F247  POP  HL
F248  JP   FFF0        ; ugrás a SYS. ROM-ba

A HI_MEM átlépés ellenörzése, blokk out rutin:
F24B  EX   DE,HL
F24C  PUSH HL
F24D  PUSH DE
F24E  PUSH BC
F24F  PUSH HL
F250  PUSH DE
F251  EX   DE,HL
F252  LD   HL,(0B19)   ; HL-be HI_MEM
F255  OR   A
F256  SBC  HL,DE
F258  LD   A,FA        ; A-ba a hibakód
F25A  POP  DE
F25B  POP  HL
F25C  RET  C           ; nem megfelelö blokk paraméter, RET
F25D  LD   C,(HL)      ; C-be a következö byte
F25E  EX   DE,HL
F25F  CALL FFF9        ; JP (HL)
F262  POP  BC
F263  POP  DE
F264  POP  HL
F265  OR   A
F266  RET  NZ          ; A hibát jelez, RET
F267  CPI
F269  RET  PO          ; vége a blokk müveletnek, RET
F26A  JR   E0          ; ugrás az F24CH címre

A HI_MEM átlépés ellenörzés, blokk in rutin:
F26C  PUSH HL
F26D  PUSH DE
F26E  PUSH BC
F26F  PUSH HL
```

```
  F270  PUSH DE
  F271  LD   HL,(0B19)   ; HL-be HI_MEM
  F274  OR   A
  F275  SBC  HL,DE
  F277  LD   A,FA        ; A-ba a hibakód
  F27A  POP  HL
  F27B  RET  C           ; nem megfelelö blokk paraméter, RET
  F27C  CALL FFF9        ; JP (HL)
  F27F  EX   AF,AF'
  F280  LD   A,C         ; A-ba az IN karakter
  F281  POP  BC
  F282  POP  DE
  F283  POP  HL
  F284  EX   AF,AF'
  F285  OR   A
  F286  RET  NZ          ; A-ban hibakód van, RET
  F287  EX   AF,AF'
  F288  LD   (DE),A      ; az IN karakter eltárolása
  F289  XOR  A
  F28A  EX   DE,HL
  F28B  CPI
  F28D  EX   DE,HL
  F28E  RET  PO          ; vége a blokk inputnak, RET
  F28F  JR   DB          ; ugrás az F26CH címre
```

A soros vonal rutinok ugrótáblája:

```
  F291  DB   05      ; a rutinok száma
  F292  DB   F305    ; a SER_INT rutin címe
  F294  DB   F339    ; a SER_CH rutin címe
  F296  DB   F3B7    ; a SER_BK rutin címe
  F298  DB   F2A4    ; a SER_SET rutin címe
  F29A  DB   F357    ; a SER_CLOSE rutin címe
```

A soros vonal inicializálása:

```
  F29C  LD   (0B69),A    ; a BAUD rendszerváltozó írása
  F29F  LD   A,EE        ; USART üzemmód byte A-ba
  F2A1  LD   (0B6A),A    ; a FORMAT rendszerváltozó írása
```

A SER_SET rutin:

```
  F2A4  PUSH BC
  F2A5  PUSH HL
  F2A6  LD   A,(0B13)    ; A-ba PORT06
```

```
F2A9  AND  C3          ; A módosítása
F2AB  LD   (0B13),A    ; PORT06 írása
F2AE  OUT  06,A
F2B0  LD   A,(0B69)    ; A-ba BAUD
F2B3  CP   09
F2B5  JR   C,02        ; ugrás az F2B9H címre, ha A-ban jó ér-
                         ték van
F2B7  LD   A,08        ; 19200 baud a sebesség
F2B9  LD   L,A
F2BA  LD   H,00
F2BC  ADD  HL,HL       ; HL-be a BAUD kétszerese
F2BD  LD   DE,F3C6     ; DE-be a BAUD RATE tábla címe
F2C0  ADD  HL,DE       ; HL-be a megfelelö BAUD RATE címe
F2C1  LD   E,(HL)
F2C2  INC  HL
F2C3  LD   D,(HL)      ; DE-be a megfelelö átviteli sebességet
                         meghatározó két byte-os érték
F2C4  LD   A,D
F2C5  AND  0F
F2C7  OR   10
F2C9  LD   D,A
F2CA  LD   A,(0B12)    ; A-ba PORT05
F2CD  AND  C0
F2CF  OR   D
F2D0  LD   (0B12),A    ; PORT05 új értéke
F2D3  OUT  05,A        ; az átviteli sebesség felsö fél byte-ja
F2D5  LD   A,E
F2D6  OUT  04,A        ; az átviteli sebesség alsó byte-ja
F2D8  LD   A,(0B6A)    ; A-ba FORMAT
F2DB  AND  B4
F2DD  OR   4A
F2DF  LD   H,A
F2E0  LD   BC,0411
F2E3  LD   DE,4003
F2E6  IN   A,5A        ; a bövítökártyán bekötött 2 bites azo-
                         nosító beolvasása
F2E8  LD   L,A
F2E9  AND  E
F2EA  JR   NZ,0A       ; ugrás az F2F6H címre, ha nem soros vo-
                         nal az adott kártya
```

Az USART írása:

```
F2EC  OUT  (C),A
F2EE  OUT  (C),A
F2F0  OUT  (C),A
F2F2  OUT  (C),D
```

```
  F2F4  OUT   (C),H
  F2F6  LD    A,10
  F2F8  ADD   A,C          ; A-ba a következö csatolóhoz tartozó
                             USART cím
  F2F9  LD    C,A          ; C-be az USART cím
  F2FA  LD    A,L
  F2FB  RRCA
  F2FC  RRCA
  F2FD  DJNZ  E9           ; ugrás az F2E8H címre, ha B nem 0
  F2FF  XOR   A            ; A-ba 0
  F300  LD    (0B71),A     ; SER_OK írása
  F303  POP   HL
  F304  POP   BC
  F305  RET                ; a SER_SET rutin vége

Az USART parancsregiszter címének meghatározása:
  F306  CALL  F331         ; volt CTRL+ESC ?
  F309  RET   NZ           ; ha igen, RET
  F30A  LD    A,(0B14)     ; A-ba SND_ACTIVE
  F30D  OR    A
  F30E  JR    NZ,F6        ; hang van folyamatban, ugrás az F306H
                             címre
  F310  LD    H,C
  F311  LD    A,(0B11)     ; A-ba PORT03
  F314  RRCA
  F315  RRCA
  F316  AND   30
  F318  ADD   A,11         ; A-ba az USART parancsregiszter címe
  F31A  LD    C,A
  F31B  XOR   A
  F31C  CCF
  F31D  RET

  F31E  NOP

Az USART állapotregiszter vizsgálata:
  F31F  CALL  F30A         ; van hang folyamatban?
  F322  RET   NZ           ; STOP-t nyomtak, RET
  F323  IN    A,(C)        ; az USART állapotregiszterének beolva-
                             sása
  F325  AND   81           ; csak az alsó és a felsö bit érdekes
  F327  CP    81
  F329  CCF
  F32A  RET   C            ; a DSR bit 1, RET
  F32B  CALL  F331         ; volt STOP ?
```

```
  F32E  JR   Z,F3          ; ugrás az F323H címre, ha nem volt STOP
  F330  RET

A STOP_FLAG vizsgálata
  F331  LD   A,(0B16)      ; A-ba STOP_FLAG
  F334  OR   A
  F335  RET  Z             ; nem volt CTRL+ESC
  F336  LD   A,F5          ; A-ba a STOP hiba kódja
  F338  RET

A SER_CHIN/OUT rutinok:
  F339  JP   M,F359        ; ha M, SER_CHIN rutin

A SER_CHOUT rutin:
  F33C  CALL F31F          ; a DSR bit vizsgálata
  F33F  RET  NC            ; STOP-t nyomtak, a rutin vége
  F340  LD   A,(0B71)      ; A-ba SER_OK
  F343  OR   A
  F344  CALL NZ,F2A4       ; nem jó órafrekvencia, ugrás a SER_SET
                             rutinra
  F347  LD   A,05          ; A-ba az USART parancs byte
  F349  OUT  (C),A         ; a parancs kiküldése
  F34B  CALL F331          ; volt STOP ?
  F34E  RET  NZ            ; ha igen, vége a rutinnak
  F34F  IN   A,(C)         ; az USART állapotregiszter olvasása
  F351  RRCA
  F352  JR   NC,F7         ; adásra nem kész az USART, ugrás az
                             F34BH címre
  F354  DEC  C             ; C-be az adatregiszter címe
  F355  OUT  (C),H         ; adat az USART-ba

A SER_CLOSE rutin:
  F357  XOR  A
  F358  RET                ; a SET_CHOUT és a SER_CLOSE rutin vége

A SER_CHIN rutin:
  F359  CALL F30A          ; az USART címének meghatározása
  F35C  RET  NC            ; STOP-t nyomtak, RET
  F35D  LD   A,(0B71)      ; A-ba SER_OK
  F360  OR   A
  F361  CALL NZ,F2A4       ; nem jó órafrekvencia, ugrás a SER_SET
                             rutinra
  F364  IN   A,(C)         ; az USART állapotregiszterének olvasása
  F366  RRCA
  F367  RRCA
```

```
F368  JR   NC,06       ; ha nincs vett karakter, ugrás az F370H
                         címre
F36A  DEC  C           ; C-be az adatregiszter címe
F36B  IN   A,(C)       ; adat beolvasása az USART-ból
F36D  LD   C,A
F36E  XOR  A           ; A-ba 0, nincs hiba
F36F  RET
F370  DI               ; a megszakítás elfogadás tíltása
F371  LD   A,(0B11)    ; A-ba PORT03
F374  AND  F0
F376  OR   07
F378  OUT  03,A        ; a 7. billentyüzetsor kiválasztása
F37A  LD   A,25
F37C  OUT  (C),A       ; az USART parancsregiszter írása
F37E  IN   A,58        ; billentyüzetadat beolvasás
F380  AND  18
F382  JR   NZ,09       ; nem nyomtak CTRL+ESC-t, ugrás az
                         F38DH címre
F384  LD   HL,0B62     ; az OLDPICT megfelelö byte-jának címe
                         HL-be
F387  SET  3,(HL)      ; az ESC-hez tartozó bit beállítása
F389  LD   A,F5        ; A-ba a STOP hiba kódja
F38B  EI               ; a megszakítás elfogadás engedélyezése
F38C  RET
F38D  IN   A,(C)       ; az USART állapotregiszterének olvasása
F38F  RRCA
F390  RRCA
F391  JR   NC,EB       ; ugrás az F37EH címre, ha nincs beolva-
                         sandó adat
F393  LD   A,05
F395  OUT  (C),A       ; az USART parancsregiszterének írása
F397  IN   A,(C)       ; az állapot byte beolvasása
F399  DEC  C
F39A  IN   H,(C)       ; H-ba a beolvasott karakter
F39C  EI               ; a megszakítás elfogadás engedélyezése
F39D  AND  38
F39F  JR   Z,14        ; a beolvasás hibátlan, ugrás az F3B5H
                         címre
F3A1  LD   B,A
F3A2  INC  C           ; C-be a parancsregiszter címe
F3A3  LD   A,11
F3A5  OUT  (C),A       ; hibatörlés, adás engedélyezés
F3A7  LD   A,F4        ; paritáshiba kódja A-ba
F3A9  BIT  3,B
F3AB  JR   NZ,08       ; ha paritáshiba, ugrás az F3B5H címre
```

```
  F3AD  LD   A,F2        ; a túlfutás hiba kódja A-ba
  F3AF  BIT  4,B
  F3B1  JR   NZ,02       ; túlfutás, ugrás az F3B5H-ra
  F3B3  LD   A,F3        ; a kerethiba kódja A-ba
  F3B5  LD   C,H         ; C-be a vett byte
  F3B6  RET              ; a SER_CHIN rutin vége

A SER_BK rutinok:
  F3B7  JP   M,F3C0      ; ha M, blokk input

A SER_BKOUT rutin:
  F3BA  LD   HL,F33C     ; HL-be a SER_CHOUT rutin címe
  F3BD  JP   F24B        ; ugrás a blokk outputra

A SER_BKIN rutin:
  F3C0  LD   HL,F359     ; HL-be a SER_CHIN rutin címe
  F3C3  JP   F26C        ; ugrás a blokk inputra


A soros vonal "baud rate"-jét meghatározó értékek:

  F3C6  DW   0C88
  F3C8  DW   0D75
  F3CA  DW   0EBA
  F3CC  DW   0F5D
  F3CE  DW   0FAF
  F3D0  DW   0FD7
  F3D2  DW   0FEC
  F3D4  DW   0FF6
  F3D6  DW   0FFB


A CAS_CH rutinok :
  F3D8  CALL F593
  F3DB  JR   17          ; ugrás az F3F4H címre

A CAS_BK rutinok:
  F3DD  CALL F630
  F3E0  JR   12          ; ugrás az F3F4H címre

A CAS_OPEN, CAS_CRTE rutinok:
  F3E2  CALL F3FA
  F3E5  JR   0D          ; ugrás az F3F4H címre
```

```
A CAS_CLOSE rutinok:
  F3E7  CALL F558
  F3EA  JR   08          ; ugrás az F3F4H címre

A CAS_VERIFY rutin:
  F3EC  CALL F605
  F3EF  JR   03          ; ugrás az F3F4H címre
  F3F1  CALL F707        ; a kazetta munkaterület törlése
  F3F4  LD   HL,D9E7     ; SYS. ROM rutin címe HL-be
  F3F7  JP   FFF0        ; ugrás a SYS. ROM-ba

A CAS_OPEN rutinok:
  F3FA  JP   P,F4CF      ; ha P, ugrás a CAS_CRTE rutinr

A CAS_OPEN rutin:
  F3FD  CALL F762        ; van már olvasásra megnyitott file?
  F400  LD   A,EB        ; A-ba a hibakód
  F402  RET  NZ          ; már van nyitott file, RET
  F403  LD   HL,0BF3
  F406  LD   BC,0121
  F409  CALL F54F        ; a kazetta input munkaterület törlése
  F40C  LD   HL,0BF4
  F40F  CALL F526        ; a FILENAME átmozgatása a HL címre
  F412  CALL F73C        ; a kurzort a következö sor elejére
  F415  LD   HL,F74E
  F418  CALL F730        ; a "Searching" szöveg kiíratása
  F41B  CALL F73C        ; a kurzort a következö sor eljére

Kazetta input változók beállítása:
  F41E  LD   A,FF
  F420  LD   (0D13),A
  F423  LD   HL,0C05
  F426  LD   (0D10),HL
  F429  LD   HL,0100
  F42C  LD   (0D09),HL
  F42F  XOR  A
  F430  LD   (0D0B),A
  F433  LD   (0D0C),A
  F436  LD   (0D0D),A
  F439  CALL F77B        ; a file fejtömbjének beolvasása
  F43C  JR   C,13        ; hibátlan a beolvasás, ugrás az F451H
                           címre
  F43E  LD   A,(0D0B)    ; A-ba a hibakód
  F441  CP   F5
  F443  JR   NZ,D9       ; nem STOP hiba, ugrás az F41EH címre
```

```
F445  LD   HL,0000
F448  LD   (0B6F),HL   ; a MUDDLE törlése
F44B  LD   HL,0BF3
F44E  LD   (HL),00     ; a file tipusbyte törlése
F450  RET              ; a rutin vége STOP-nál
F451  LD   HL,F758     ; HL-be a "Found:" szöveg címe
F454  EXX
F455  LD   HL,0C05     ; HL-be az input puffer kezdete
F458  PUSH HL
F459  LD   DE,0BF4     ; DE-be a megadott file név címe
F45C  LD   A,(DE)      ; A-ba a megadott file név hossza
F45D  OR   A
F45E  JR   Z,0A        ; nincs file név megadva, ugrás az F46AH
                         címre
F460  LD   B,A
F461  INC  B
F462  LD   A,(DE)      ; A-ba a megadott file név karaktere
F463  SUB  (HL)
F464  JR   NZ,08       ; a beolvasott file név kölönbözik, ug-
                         rás az F46EH címre
F466  INC  DE
F467  INC  HL
F468  DJNZ F8          ; nincs vége az összehasonlításnak, ug-
                         rás az F462H címre
F46A  LD   HL,F744     ; HL-be a "Reading:" szöveg címe
F46D  EXX
F46E  EXX
F46F  EX   AF,AF'
F470  CALL F730        ; a HL mutatta szöveg kiíratása
F473  POP  HL
F474  PUSH HL
F475  LD   A,(HL)      ; A-ba a beolvasott file név hossza
F476  OR   A
F477  CALL NZ,F730     ; a beolvasott file név üres string, a
                         kurzort a következö sor elejére
F47A  CALL F73C        ; a beolvasott file név kiíratása
F47D  EX   AF,AF'
F47E  POP  HL
F47F  JR   NZ,9D       ; nem a keresett file, ugrás az F41EH
                         címre
F481  LD   A,(0D14)    ; A-ba a file tipusbyte-ja
F484  OR   A
F485  JR   Z,0E        ; pufferelt file, ugrás az F495H címre
F487  LD   HL,0D0C
F48A  LD   A,(HL)      ; A-ba a file írásvédelmi byte-ja
```

```
  F48B  OR   A
  F48C  JR   Z,07          ; a file nem írásvédett, ugrás az F495H
                             címre
  F48E  XOR  A
  F48F  LD   (HL),A        ; az írásvédelmi byte törlése
  F490  LD   A,E6          ; A-ba a hibakód
  F492  JP   F445

  F495  LD   A,(0BF3)      ; A-ba a file tipusbyte-ja
  F498  SUB  11
  F49A  LD   (0B6B),A      ; BUFFER írása
  F49D  LD   A,00
  F49F  LD   (0D13),A      ; a tömbtipus azonosító átírása

A beolvasott file név átmozgatása a megadott helyére:
  F4A2  LD   HL,0C05
  F4A5  LD   DE,0BF4
  F4A8  LD   A,(HL)
  F4A9  CP   11
  F4AB  JR   C,02
  F4AD  LD   A,10
  F4AF  INC  A
  F4B0  LD   C,A
  F4B1  LD   B,00
  F4B3  LDIR

  F4B5  LD   HL,0C05
  F4B8  LD   E,(HL)        ; E-be a beolvasott file név hossza
  F4B9  INC  E
  F4BA  LD   D,00
  F4BC  ADD  HL,DE
  F4BD  LD   (0D07),HL     ; az input puffer kezdetének címe
  F4C0  LD   HL,(0D05)
  F4C3  XOR  A
  F4C4  SBC  HL,DE
  F4C6  LD   (0D05),HL     ; a puffer mutató eltárolása
  F4C9  LD   DE,0BF4       ; DE-be a file név címe
  F4CC  JP   F5B5          ; az EOF beállítása, a CAS_OPEN rutin
                             vége

A CAS_CRTE rutin:
  F4CF  CALL F76B          ; van írásra megnyitott file?
  F4D2  LD   A,EB          ; A-ba a hibakód
  F4D4  RET  NZ            ; már van nyitott file, RET
  F4D5  LD   A,(0D0C)      ; A-ba az olvasásra nyitott file írásvé-
```

```
                                  delmi byte-ja
F4D8  OR   A
F4D9  LD   A,E6          ; A-ba a hibakód
F4DB  RET  NZ            ; védett file van beolvasva, RET
F4DC  LD   HL,0D14
F4DF  LD   BC,0120
F4E2  CALL F54F          ; az output munkaterület törlése
F4E5  LD   HL,0D15
F4E8  CALL F526          ; a FILENAME átmozgatása a HL mutatta
                           címre
F4EB  LD   A,(0B6B)      ; A-ba BUFFER
F4EE  OR   A
F4EF  LD   A,01
F4F1  JR   NZ,02         ; pufferelt file, ugrás az F4F5H címre
F4F3  LD   A,11
F4F5  LD   (0D14),A      ; a megnyitandó file tipusbyte-jának el-
                           tárolása
F4F8  LD   (0E2F),A
F4FB  LD   HL,0B6D
F4FE  LD   A,(HL)        ; A-ba PROTECT
F4FF  LD   (HL),00
F501  LD   (0E30),A      ; az írásvédelmi byte eltárolása
F504  LD   A,FF
F506  LD   (0E32),A      ; a tömbtipus mutató beállítása
F509  LD   HL,0D15
F50C  LD   DE,0D26
F50F  LD   (0E2C),DE     ; a kiírandó terület kezdetének beállí-
                           tása
F513  LD   C,(HL)        ; C-be a file név hossza
F514  INC  C
F515  LD   B,00
F517  LD   (0E26),BC     ; az output puffer megtöltött mérete
F51B  LDIR               ; a file név átmozgatása az output puf-
                           ferba
F51D  LD   (0E28),DE     ; a kiírandó terület végének beállítása
F521  LD   DE,0D15
F524  XOR  A
F525  RET                ; a CAS_CRTE rutin vége

A FILENAME átmozgatása a HL mutatta címre:
F526  EX   DE,HL
F527  LD   A,(HL)        ; A-ba a FILENAME hossza
F528  CP   11
F52A  JR   C,02          ; nem hosszabb 17 karakternél, ugrás az
                           F5CEH címre
```

```
F52C  LD   A,10
F52E  LD   (DE),A      ; a file név hosszának átmozgatása
F52F  OR   A
F530  RET  Z           ; a file név üres string, RET
F531  LD   B,A         ; B-be a file név hossza
F532  INC  HL
F533  INC  DE
F534  LD   A,(HL)      ; A-ba a file név következő karaktere
F535  CP   61
F537  JR   C,12        ; nem kisbetű, ugrás az F54BH címre
F539  CP   7B
F53B  JR   NC,04       ; nem ASCII kisbetű, ugrás az F541H
                         címre
F53D  AND  DF          ; kisbetűből nagy betű
F53F  JR   0A          ; ugrás az F54BH címre
F541  CP   90
F543  JR   C,06        ; nem ékezetes kisbetű, ugrás az F54BH
                         címre
F545  CP   99
F547  JR   NC,02       ; nem ékezetes kisbetű, ugrás az F54BH
                         címre
F549  SUB  10          ; ékezetes kisbetűből nagy betű
F54B  LD   (DE),A      ; a karakter átmozgatása
F54C  DJNZ E4          ; nincs kész az átmozgatás, ugrás az
                         F532H címre
F54E  RET

A HL mutatta címtől kezdve BC számú byte törlése:
F54F  LD   (HL),00
F551  INC  HL
F552  DEC  BC
F553  LD   A,B
F554  OR   C
F555  JR   NZ,F8       ; BC nem 0, ugrás az F54FH címre
F557  RET

A CAS_CLOSE rutinok:
F558  JP   M,F585      ; ha M, ugrás a CAS_CLOSE_IN rutinra

A CAS_CLOSE_OUT rutin:
F55B  CALL F76B        ; van írásra nyitott file?
F55E  JR   Z,1C        ; nincs, ugrás az F57CH címre
F560  LD   HL,0D26
F563  LD   (0E2C),HL   ; az output puffer címének eltárolása
F566  LD   A,FF
```

```
  F568  LD    (0E2B),A    ; az EOF_FLAG beállítása EOF-ra
  F56B  LD    A,(0E2A)    ; A-ba az esetleges hibakód
  F56E  OR    A
  F56F  JR    NZ,0B       ; ugrás az F57CH címre, ha nem volt hiba
  F571  CALL  F972        ; az utolsó szektor felírása
  F574  JR    C,06        ; ugrás az F57CH címre, ha CARRY
  F576  EX    AF,AF'
  F577  CALL  F57C
  F57A  EX    AF,AF'
  F57B  RET
  F57C  XOR   A
  F57D  LD    (0D14),A    ; a file tipusbyte törlése
  F580  LD    (0E2A),A    ; a hibamutató törlése
  F583  JR    07          ; ugrás az F58CH címre

A CAS_CLOSE_IN rutin:
  F585  XOR   A
  F586  LD    (0BF3),A    ; a file tipusbyte törlése
  F589  LD    (0D0C),A    ; az írásvédelmi byte törlése
  F58C  LD    HL,0000
  F58F  LD    (0B6F),HL   ; MUDDLE törlése
  F592  RET

A CAS_CH rutinok:
  F593  JP    P,F653      ; ha P, ugrás a CAS_CHOUT rutinra

A CAS_CHIN rutin:
  F596  CALL  F762        ; van olvasásra megnyitott file?
  F599  RET   Z           ; igen, RET
  F59A  CALL  F772        ; van hibajelzés?
  F59D  RET   NZ          ; van, RET
  F59E  LD    BC,(0D05)   ; BC-be az input puffer be nem olvasott
                            byte-jainak száma
  F5A2  LD    A,B
  F5A3  OR    C
  F5A4  JR    Z,1B        ; új szektort kell beolvasni, ugrás az
                            F5C1H címre
  F5A6  LD    HL,(0D07)   ; A-ba a buffer következö beolvasható
                            byte-jának a címe
  F5A9  LD    C,(HL)      ; a byte beolvasása
  F5AA  INC   HL
  F5AB  LD    (0D07),HL   ; HL eltárolása
  F5AE  LD    HL,(0D05)   ; HL-be az input puffer aktív méret
  F5B1  DEC   HL
  F5B2  LD    (0D05),HL   ; a módosított input puffer aktív méret
```

```
                              eltárolása
  F5B5  LD   A,H
  F5B6  OR   L
  F5B7  JR   NZ,06          ; van  még be nem olvasott karakter, ug-
                              rás az F5BFH címre
  F5B9  LD   A,(0D0E)       ; A-ba az input EOF_FLAG
  F5BC  LD   (0B6E),A       ; EOF írása
  F5BF  XOR  A
  F5C0  RET
  F5C1  LD   A,(0BF3)       ; A-ba a file tipusbyte-ja
  F5C4  DEC  A
  F5C5  LD   A,E7
  F5C7  RET  NZ             ; nem pufferelt file, RET
  F5C8  LD   HL,0100
  F5CB  LD   (0D09),HL      ; a beolvasandó blokk mérete
  F5CE  LD   HL,0C05
  F5D1  LD   (0D10),HL      ; az input puffer címe
  F5D4  LD   (0D07),HL      ; az input puffer mutató beállítás
  F5D7  LD   A,(0D0E)       ; A-ba az EOF_FLAG
  F5DA  OR   A
  F5DB  LD   A,EC
  F5DD  CALL Z,F77B         ; van még beolvasható szektor, ugrás
  F5E0  JR   NC,0C          ; ugrás az F5EBH címre, ha NC
  F5E2  LD   A,(0BF3)       ; A-ba a file tipusbyte
  F5E5  DEC  A
  F5E6  JR   Z,BE           ; pufferelt file, ugrás az F5A6H címre
  F5E8  XOR  A
  F5E9  SCF
  F5EA  RET
  F5EB  LD   (0D0B),A       ; a hibakód eltárolása
  F5EE  LD   HL,0D0B
  F5F1  LD   A,(HL)         ; A-ba a hibakód
  F5F2  LD   (HL),EC        ; hibakód betöltése
  F5F4  LD   HL,0D0E
  F5F7  LD   (HL),FF        ; EOF_FLAG beállítása
  F5F9  LD   HL,0000
  F5FC  LD   (0D05),HL      ; az input puffer aktív méret törlése
  F5FF  LD   HL,0B6E
  F602  LD   (HL),FF        ; EOF beállítása
  F604  RET

A CAS_VERIFY rutin:
  F605  CALL F762           ; van olvasásra megnyitott file?
  F608  RET  Z              ; nincs, RET
  F609  CALL F772           ; van beállított hiba jelzés?
```

```
F60C  RET  NZ          ; igen, RET
F60D  LD   A,FF
F60F  LD   (0BF1),A    ; a VERIFY_FLAG beállítása
F612  LD   A,(0BF3)    ; A-ba a file tipusbyte-ja
F615  DEC  A
F616  JR   NZ,27       ; nem pufferelt file, ugrás az F63FH
                         címre
F618  PUSH BC
F619  PUSH DE
F61A  CALL F59E        ; C-be a file következö karaktere
F61D  LD   L,C
F61E  POP  DE
F61F  POP  BC
F620  OR   A
F621  RET  NZ          ; hiba történt, RET
F622  LD   A,(DE)      ; A-ba az összehasonlítandó byte
F623  CP   L
F624  LD   A,E8
F626  JP   NZ,F5EB     ; a két byte különböző, ugrás
F629  INC  DE          ; DE a következö összehasonlítandó byte-
                         ra mutat
F62A  DEC  BC          ; a byte számláló csökken
F62B  LD   A,B
F62C  OR   C
F62D  JR   NZ,E9       ; van még összehasonlítandó byte, ugrás
                         az F618H címre
F62F  RET

A CAS_BK rutinok:
F630  JP   P,F6AD      ; ha P, ugrás a CAS_BKOUT rutinra

A CAS_BKIN rutin:
F633  CALL F762        ; van olvasásra nyitott file?
F636  RET  Z           ; nincs, RET
F637  CALL F772        ; van beállított hiba jelzés?
F63A  RET  NZ          ; igen, RET
F63B  LD   A,(0BF3)    ; A-ba a file tipusbyte-ja
F63E  DEC  A
F63F  LD   HL,F59E     ; HL-be a CAS_CHIN rutin címe
F642  JP   Z,F26C      ; pufferelt file, ugrás
F645  LD   (0D10),DE   ; a beolvasandó file kezdö memória címe
F649  LD   (0D09),BC   ; a beolvasandó file mérete
F64D  CALL F5D7        ; a file beolvasása
F650  JP   F5F4
```

A CAS_CHOUT rutin:

```
F653  CALL F76B       ; van írásra megnyitott file?
F656  RET  Z          ; nincs, RET
F657  LD   A,(0E2A)   ; A-ba az esetleges hibakód
F65A  OR   A
F65B  LD   A,EC
F65D  RET  NZ         ; van beállított hiba jelzés, RET
F65E  LD   A,C        ; a felírandó karakter kódja A-ba
F65F  LD   (0E2E),A   ; a kiírandó karakter eltárolása
F662  LD   HL,0E2F
F665  LD   A,(HL)     ; A-ba a file tipusbyte-ja
F666  OR   A
F667  JR   Z,04       ; pufferelt file, ugrás az F66DH címre
F669  DEC  (HL)
F66A  JP   Z,F689     ; nem pufferelt file, ugrás
F66D  LD   DE,(0E26)  ; DE-be a puffer megtelt mérete
F671  LD   HL,0100    ; HL-be a kiírandó blokk mérete
F674  OR   A
F675  SBC  HL,DE
F677  JP   Z,F689     ; az output puffer megtelt, ugrás
F67A  INC  DE
F67B  LD   (0E26),DE  ; a növelt DE eltárolása
F67F  LD   HL,(0E28)  ; HL-be a puffer következö üres byte-já-
                        nak címe
F682  LD   (HL),C     ; a felírandó karakter az output puffer-
                        be
F683  INC  HL
F684  LD   (0E28),HL  ; az új output puffer mutató eltárolása
F687  XOR  A
F688  RET
F689  LD   HL,0D26    ; HL-be az output puffer címe
F68C  LD   (0E2C),HL  ; HL eltárolása
F68F  CALL F972       ; a szektor felírása
F692  JR   C,05       ; ugrás az F699H címre, ha CARRY
F694  LD   HL,0E2A
F697  DEC  (HL)
F698  RET
F699  LD   HL,0001
F69C  LD   (0E26),HL  ; az output puffer megtelt méret mutató
                        alaphelyzetbe állítása
F69F  LD   A,(0E2E)   ; A-ba a felírandó karakter kódja
F6A2  LD   HL,0D26    ; az output buffer kezdö címe HL-be
F6A5  LD   (HL),A     ; a kiírandó karakter a pufferbe
F6A6  INC  HL
F6A7  LD   (0E28),HL  ; a puffer mutató eltárolása
```

```
  F6AA  XOR  A
  F6AB  SCF
  F6AC  RET

A CAS_BKOUT rutin:
  F6AD  CALL F76B        ; van írásra megnyitott file?
  F6B0  RET  Z           ; nincs, RET
  F6B1  LD   A,(0D14)    ; A-ba a file tipusbyte
  F6B4  DEC  A
  F6B5  LD   HL,F657     ; HL-be a CAS_CHOUT rutin címe
  F6B8  JP   Z,F24B      ; pufferelt file, ugrás
  F6BB  LD   A,(0E2A)    ; A-ba az esetleges hibakód
  F6BE  OR   A
  F6BF  LD   A,EC
  F6C1  RET  NZ          ; van beállított hiba jelzés, RET
  F6C2  PUSH BC
  F6C3  PUSH DE
  F6C4  LD   HL,0E2F
  F6C7  LD   A,(HL)      ; A-ba a file tipusbyte
  F6C8  LD   (HL),00
  F6CA  OR   A
  F6CB  SCF
  F6CC  CALL NZ,F689     ; még van byte-onkénti kiírási lehetöség
                           nem pufferelt file-nál
  F6CF  POP  DE
  F6D0  POP  BC
  F6D1  JR   NC,C1       ; ugrás az F684H címre, ha NC
  F6D3  LD   (0E26),BC   ; a kiírandó blokk mérete
  F6D7  LD   (0E2C),DE   ; a kiírandó blokk kezdö címe
  F6DB  JP   F566        ; a blokk felírása

A motor vezérlése:
  F6DE  LD   L,40
  F6E0  OR   A
  F6E1  JR   Z,01        ; ugrás az F6E4H címre, ha olvasás
  F6E3  ADD  HL,HL       ; L=80H
  F6E4  LD   A,(0B6C)    ; A-ba REMRED
  F6E7  AND  L
  F6E8  JR   Z,04        ; ugrás az F6EEH címre, ha Z
  F6EA  LD   A,C0
  F6EC  XOR  L
  F6ED  LD   L,A
  F6EE  DI               ; a megszakitáskérés elfogadásának til-
                           tása
  F6EF  LD   A,(0B12)    ; A-ba PORT05
```

```
F6F2  OR   L
F6F3  INC  D
F6F4  DEC  D
F6F5  JR   Z,01          ; ugrás az F6F8H címre, ha D=0
F6F7  XOR  L
F6F8  OUT  05,A          ; a megfelelö motor indítása
F6FA  LD   (0B12),A      ; PORT05 eltárolása
F6FD  EI                 ; a megszakításkérés elfogadás engedé-
                           lyezése
F6FE  RET
F6FF  DI                 ; a megszakításkérés elfogadás tiltása
F700  LD   A,(0B12)      ; A-ba a PORT05
F703  AND  3F            ; a motorindító bitek törlése
F705  JR   F1            ; ugrás az F6F8H címre
```

A kazetta munkaterület törlése:

```
F707  XOR  A
F708  LD   HL,0BF0       ; HL-be a kazetta munkaterület kezdete
F70B  LD   DE,0BF1
F70E  LD   BC,0244       ; BC-be a munkaterület mérete
F711  LD   (HL),A
F712  LDIR               ; a munkaterület törlése
F714  LD   (0B6D),A      ; PROTECT törlése
F717  LD   (0B6E),A      ; EOF törlése
F71A  LD   HL,0000
F71D  LD   (0B6F),HL     ; MUDDLE törlése
F720  LD   A,80
F722  LD   (0B6C),A      ; REMRED alaphelyzetbe
F725  LD   A,(0B12)      ; A-ba PORT05
F728  AND  3F            ; a motor vezérlö bitek törlése
F72A  LD   (0B12),A      ; PORT05 visszatöltése
F72D  OUT  05,A          ; motorok leállítása
F72F  RET
```

A HL mutatta string kiíratása:

```
F730  LD   B,(HL)        ; B-be a string hossza
F731  INC  HL
F732  PUSH BC
F733  PUSH HL
F734  LD   C,(HL)        ; C-be a string következö karaktere
F735  RST  30H
F736  DB   21            ; ED_CHOUT rutin
F737  POP  HL
F738  POP  BC
F739  DJNZ F6            ; nincs vége a stringnek, ugrás az F731H
                           címre
F73B  RET
```

```
Kurzor a következö sor elejére:
  F73C  LD   HL,F741
  F73F  JR   EF          ; ugrás az F730H címre

  F741  DB   02 0D 0A 09
  F745  DB   'Reading: '
  F74E  DB   09
  F74F  DB   'Searching'
  F758  DB   09
  F759  DB   'Found:   '

Van olvasásra megnyitott file?
  F762  XOR  A
  F763  LD   (0BF1),A    ; a VERIFY_FLAG törlése
  F766  LD   A,(0BF3)    ; A-ba a file tipusbyte-ja
  F769  JR   03          ; ugrás az F76EH címre

Van írásra megnyitott file?
  F76B  LD   A,(0D14)    ; A-ba a file tipusbyte-ja
  F76E  OR   A
  F76F  LD   A,E9        ; A-ba a hibakód
  F771  RET

Hibajelzés vizsgálat olvasásnál:
  F772  LD   HL,0D0B
  F775  LD   A,(HL)      ; A-ba az esetleges hibakód
  F776  OR   A
  F777  RET  Z           ; nincs beállított hibajelzés, RET
  F778  LD   (HL),EC     ; hibajelzés beállítása
  F77A  RET

Egy tömb beolvasása a kazettáról:
  F77B  LD   (0E33),SP   ; SP eltárolása
  F77F  XOR  A
  F780  LD   D,A
  F781  CALL F6DE        ; motorvezérlés
  F784  CALL FA7C        ; az IT kiszolgáló rutin módosítása

A visszaolvasás szinkronizálása, szalagsebesség megállapítása:
  F787  XOR  A
  F788  LD   (0BF0),A    ; a border-puffer törlése
  F78B  EXX
  F78C  LD   HL,0000
  F78F  LD   B,00
  F791  EXX
```

```
  F792  LD   A,DC
  F794  OUT  04,A        ; a PITCH alsó 8 bitjének beállítása (a
                           felsö négy bit: 0FH)
  F796  LD   C,00
  F798  CALL F93C        ; szinkronizálás
A szalagsebesség vizsgálata:
  F79B  LD   HL,0000
  F79E  LD   B,20
  F7A0  LD   D,H         ; D-be 0
  F7A1  CALL F938        ; E-be a jel frekvenciájára jellemzö ér-
                           ték
  F7A4  ADD  HL,DE
  F7A5  DJNZ FA          ; ugrás az F7A1H címre, ha B nem 0
Az átlagos jel frekvencia meghatározása:
  F7A7  ADD  HL,HL
  F7A8  ADD  HL,HL
  F7A9  ADD  HL,HL       ; HL=HL*8
  F7AA  LD   A,H
  F7AB  RL   L
  F7AD  ADC  A,B         ; A-ba HL/256 egészre kerekítve, az át-
                           lagos jel frekvenciára jellemzö érték
  F7AE  LD   D,A
  F7AF  LD   HL,0400
  F7B2  CALL F938        ; E-be a jel frekvenciájára jellemzö ér-
                           ték
  F7B5  EXX
  F7B6  ADD  A,B
  F7B7  LD   B,A
  F7B8  JR   NC,01       ; ugrás az F7BBH címre, ha nincs átvitel
  F7BA  INC  HL
  F7BB  EXX
  F7BC  LD   A,E
  F7BD  SUB  D
  F7BE  JR   NC,02       ; ugrás az F7C2H címre, ha E-D>0
  F7C0  NEG
  F7C2  CP   03
  F7C4  JR   NC,C1       ; ugrás az F787H címre, ha a szalagsebes-
                           ség ingadozás túlzott
  F7C6  DEC  HL
  F7C7  LD   A,H
  F7C8  OR   L
  F7C9  JR   NZ,E7       ; ugrás az F7B2H címre, ha HL nem 0
  F7CB  EXX
Az átlagos jelfrekvencia meghatározása:
  F7CC  ADD  HL,HL
```

```
F7CD  ADD  HL,HL
F7CE  ADD  HL,HL
F7CF  ADD  HL,HL
F7D0  ADD  HL,HL
F7D1  ADD  HL,HL        ; HL=HL*64
F7D2  LD   A,H
F7D3  RL   L
F7D5  ADC  A,00         ; A-ba HL/256 egészre kerekítve
F7D7  EXX
F7D8  PUSH AF
F7D9  LD   A,88
F7DB  LD   (0BF0),A     ; a border-puffer módosítása
F7DE  LD   A,E8
F7E0  OUT  04,A         ; PITCH alsó 8 bitjének írása
F7E2  CALL F93C         ; szinkronizálás
F7E5  LD   HL,0000
F7E8  PUSH HL
F7E9  LD   B,H
F7EA  LD   D,H          ; D=0 és B=0
F7EB  EX   SP,HL
F7EC  CALL F92B         ; E-be a jel frekvenciájára jellemzö ér-
                          ték
F7EF  ADD  HL,DE
F7F0  ADD  HL,DE        ; HL=HL+2*DE
F7F1  DJNZ F8           ; ugrás az F7EBH címre, ha B nem 0
F7F3  LD   A,H
F7F4  RL   L
F7F6  ADC  A,B          ; A-ba HL/256 egészre kerekítve
F7F7  LD   D,A
F7F8  POP  HL
F7F9  LD   A,H
F7FA  RL   L
F7FC  ADC  A,B          ; A-ba HL/256 egészre kerekítve
F7FD  LD   H,A
F7FE  LD   L,D
F7FF  POP  DE
F800  CALL F92B         ; E-be a jel frekvenciájára jellemzö ér-
                          ték
F803  LD   A,L
F804  LD   L,H
F805  LD   H,A
F806  CALL F92B         ; E-be a jel frekvenciájára jellemzö ér-
                          ték
F809  SUB  H
F80A  JR   NC,07        ; ugrás az F813H címre, ha NC
```

```
F80C  ADD  A,02
F80E  JR   C,F3        ; ugrás az F803H címre, ha CARRY
F810  JP   F787
F813  CP   04
F815  JR   C,EC        ; ugrás az F803H címre, ha CARRY
F817  LD   A,DC
F819  OUT  04,A        ; PITCH alsó 8 bitjének módosítása
F81B  CALL F92B        ; E-be a jel frekvenciájára jellemző ér-
                         ték, a jel fázisának megállapítása
F81E  LD   IY,0000
F822  LD   IX,(0D10)   ; a beolvasott blokk helye
F826  CALL F919        ; az üres byte beolvasása
F829  EXX
F82A  LD   HL,(0B6F)   ; HL-be MUDDLE
F82D  LD   DE,(0D09)   ; DE-be a beolvasandó byte-ok száma
F831  EXX
F832  CALL F919        ; a szinkronbyte beolvasása
F835  CP   6A
F837  JP   NZ,F787     ; rossz a szinkronizálás, ugrás
F83A  CALL F919        ; a tömbazonosító beolvasása
F83D  LD   HL,0D13     ; HL-be a tömbazonosító címe
F840  CP   (HL)
F841  JR   Z,09        ; a beolvasott azonosító jó, ugrás az
                         F84CH címre
F843  LD   A,(HL)      ; A-ba a megadott tömbazonosító
F844  CP   FF
F846  JP   Z,F787      ; fejtömböt vár, ugrás
F849  JP   F8FD        ; hibajelzés
F84C  LD   (0D0F),A    ; a beolvasott tömbtipus azonosító eltá-
                         rolása
F84F  CALL F919        ; a file tipusbyte beolvasása
F852  LD   (0BF3),A    ; a beolvasott file tipusbyte eltárolása
F855  CALL F919        ; az írásvédelmi byte beovasása
F858  LD   (0D0C),A    ; az írásvédelmi byte eltárolása
F85B  CALL F919        ; a szektorszám beolvasása
F85E  LD   B,A
F85F  PUSH BC          ; a szektorszám a STACK-be
F860  JR   06          ; ugrás az F868H címre
F862  PUSH BC
F863  EXX
F864  LD   HL,(0B6F)   ; HL-be MUDDLE
F867  EXX
F868  CALL F919        ; a szektorsorszám beolvasása
F86B  LD   HL,0D0D     ; HL-be a várható szektorsorszám címe
F86E  CP   (HL)
```

```
F86F  JP   NZ,F8FD     ; hibajelzés, nem megfelelö szektorsor-
                         szám
F872  CALL F919        ; az adott szektorban lévö adatbyte-ok
                         számának beolvasása
F875  LD   B,A
F876  CALL F919        ; adatbyte beolvasása
F879  EX   AF,AF'
F87A  EXX
F87B  LD   A,D
F87C  OR   E
F87D  DEC  DE
F87E  EXX
F87F  JP   Z,F901      ; nem kell több byte-ot beolvasni, ugrás
F882  LD   A,(0BF3)    ; A-ba a file tipusbyte-ja
F885  DEC  A
F886  JR   Z,0F        ; pufferelt file, ugrás az F897H címre
F888  LD   A,(0BF1)    ; A-ba a VERIFY_FLAG
F88B  OR   A
F88C  JR   Z,09        ; beolvasás, nem összehasonlítás, ugrás
                         az F897H címre
F88E  EX   AF,AF'
F88F  CP   (IX+00H)    ; a beolvasott byte és a megfelelö memó-
                         riabyte összehasonlítása
F892  JR   Z,07        ; ugrás az F89BH címre, ha a két byte e-
                         gyenlö
F894  JP   F909        ; összehasonlítási hiba
F897  EX   AF,AF'
F898  LD   (IX+00H),A  ; a beolvasott byte a megfelelö helyre
F89B  INC  IX
F89D  INC  IY
F89F  DJNZ D5          ; még van beolvasandó byte, ugrás az
                         F876H címre
F8A1  CALL F919        ; a szektorvége byte beolvasása
F8A4  LD   (0D0E),A    ; az EOF_FLAG írása
F8A7  EXX
F8A8  PUSH HL
F8A9  EXX
F8AA  CALL F919        ; a CRC magas helyiértékü byte-jának be-
                         olvasása
F8AD  CP   A
F8AE  EX   AF,AF'
F8AF  POP  HL          ; HL-be a számított CRC
F8B0  LD   B,H
F8B1  CALL F919        ; a CRC alacsony helyiértékü byte-jának
                         a beolvasása
```

```
  F8B4  LD   C,L          ; BC-be a számított CRC
  F8B5  LD   H,A
  F8B6  EX   AF,AF'
  F8B7  LD   L,A          ; HL-be a beolvasott CRC
  F8B8  SBC  HL,BC
  F8BA  POP  BC
  F8BB  JR   NZ,40        ; CRC hiba, ugrás az F8FDH címre
  F8BD  LD   (0D05),IY    ; a beolvasott adatbyte-ok száma
  F8C1  LD   (0D10),IX    ; a beolvasott terület vége
  F8C5  LD   HL,0D0D
  F8C8  INC  (HL)         ; a szektorsorszám mutató állítása
  F8C9  EXX
  F8CA  LD   (0D09),DE    ; a még beolvasandó byte-ok száma
  F8CE  EXX
  F8CF  DJNZ 91           ; van még beolvasandó szektor, ugrás az
                            F862H címre
  F8D1  LD   A,(0BF3)     ; A-ba a file tipusbyte-ja
  F8D4  DEC  A
  F8D5  JR   Z,0C         ; pufferelt file, ugrás az F8E3H címre
  F8D7  LD   A,(0D13)     ; A-ba a tömbtipus azonosító
  F8DA  CP   FF
  F8DC  JR   Z,05         ; fejtömb, ugrás az F8E3H címre
  F8DE  EXX
  F8DF  LD   A,D
  F8E0  OR   E
  F8E1  JR   NZ,1E        ; még kellene beolvasni, ugrás az F901H
                            címre
  F8E3  JP   FABD         ; az IT kiszolgálás visszaállítása

Hibajelzések:
  F8E6  LD   A,F5         ; A-ba a STOP hiba kódja
  F8E8  LD   (0E2A),A     ; hibajelzés beállítása kazetta íráskor
  F8EB  JR   05           ; ugrás az F8F2H címre
  F8ED  LD   A,F5         ; A-ba a STOP hiba kódja
  F8EF  LD   (0D0B),A     ; hibajelzés beállítása kazetta olvasás-
                            kor
  F8F2  LD   HL,0B62      ; az OLDPICT 7. byte-ja címe HL-be
  F8F5  SET  3,(HL)       ; ESC nyomás beállítása
Késleltető ciklus:
  F8F7  LD   B,00
  F8F9  DJNZ FE           ; ugrás az F8F9H-ra ha B nem 0

  F8FB  JR   11           ; ugrás az F90EH címre
  F8FD  LD   A,EA         ; A-ba a CRC hiba kódja
  F8FF  JR   0A           ; ugrás az F90EH címre
```

```
F901  LD   A,(0BF1)    ; A-ba a VERIFY_FLAG
F904  OR   A
F905  LD   A,E7        ; A-ba a belsö hiba kódja
F907  JR   Z,02        ; nincs összehasonlítás, ugrás az F90BH
                         címre
F909  LD   A,E8        ; A-ba a komparálási hiba kódja
F90B  LD   (0D0B),A    ; az olvasási hiba kód eltárolása
F90E  PUSH AF
F90F  CALL FABD        ; az IT kezelés visszaállítása
F912  POP  AF
F913  OR   A
F914  LD   SP,(0E33)   ; a veremmutató visszaállítása
F918  RET
```

Egy byte beolvasása kazettáról:

```
F919  LD   H,80
F91B  CALL F938        ; a kazetta jel vizsgálata
F91E  CP   D
F91F  PUSH AF          ; a CARRY flag jelzi a beolvasott bitet
F920  SBC  A,A
F921  CALL FAF7        ; az ellenörzö összeg számítása
F924  POP  AF
F925  RR   H           ; H-ba a beolvasott bit
F927  JR   NC,F2       ; még nincs kész a byte, ugrás az F91BH
                         címre
F929  LD   A,H         ; A-ba a beolvasott byte
F92A  RET
```

A jel fázisának, frekvenciájának meghatározása:

```
F92B  IN   A,59        ; olvasás a kazettáról
F92D  AND  20          ; A-ba a jelszint
F92F  XOR  20          ; A-ba az ellentétes jelszint
F931  LD   C,A
F932  LD   E,00
F934  EI               ; a megszakítás elfogadás engedélyezése
F935  HALT             ; hang megszakításra vár a rendszer
F936  JR   0F          ; ugrás az F947H címre
F938  LD   E,00
F93A  EI               ; a megszakítás elfogadás engedélyezése
F93B  HALT             ; hang IT-re vár a rendszer
F93C  INC  E
F93D  CALL Z,F969      ; ha E=0, CTRL+ESC ellenörzés
F940  IN   A,59        ; olvasás a kazettáról
F942  XOR  C
F943  AND  20
```

```
  F945  JR   Z,F5          ; ugrás az F93CH címre, ha Z
  F947  INC  E
  F948  CALL Z,F969        ; CTRL+ESC vizsgálat, ha E=0
  F94B  IN   A,59          ; olvasás a kazettáról
  F94D  XOR  C
  F94E  AND  20
  F950  JR   NZ,F5         ; ugrás az F947H címre, ha NZ
  F952  IN   A,5B          ; a frekvenciaosztó törlése
  F954  OUT  07,A          ; a hang IT törlése

A border csíkozása:
  F956  LD   A,(0BF0)      ; A-ba a border byte
  F959  OR   A
  F95A  JR   Z,0D          ; ha Z, nincs border csíkozás, ugrás az
                             F969H címre
  F95C  CP   88
  F95E  LD   A,A0          ; A-ba a border byte
  F960  JR   Z,02          ; 88H volt a border, ugrás az F964H-ra
  F962  LD   A,88
  F964  LD   (0BF0),A      ; A-ba az aktuális border byte eltárolása
  F967  OUT  00,A          ; az új border beállítása
  F969  IN   A,58          ; a billentyüzet byte beolvasása
  F96B  AND  18
  F96D  JP   Z,F8ED        ; CTRL+ESC-et nyomtak, ugrás
  F970  LD   A,E
  F971  RET
Blokk kiírása a kazettára:
  F972  LD   (0E33),SP     ; SP elmentése
  F976  XOR  A
  F977  LD   D,A           ; D-be 0
  F978  DEC  A             ; A-ba FFH
  F979  CALL F6DE          ; motorvezérlés

Késleltetö ciklus:
  F97C  LD   BC,0000
  F97F  EX   SP,HL
  F980  DEC  BC
  F981  LD   A,B
  F982  OR   C
  F983  JR   NZ,FA         ; BC nem 0, ugrás az F97FH címre

  F985  CALL FA7C          ; az IT kiszolgálás módosítása
  F988  LD   A,88
  F98A  LD   (0BF0),A      ; a kezdö border byte eltárolása
  F98D  LD   BC,0014       ; C-be 20
```

```
  F990  LD   A,(0E32)    ; A-ba a kiírandó tömb tipusa
  F993  OR   A
  F994  JR   Z,03        ; adattömb, ugrás az F999H címre
  F996  LD   BC,0028     ; C-be 40
  F999  CALL FA31        ; a bevezetö hang felírása
  F99C  CALL FA62
  F99F  CALL FA62        ; a szikronperiódus felírása

A tömböt bevezetö 6 byte felírása:
  F9A2  CALL FA3D        ; az üres byte felírása
  F9A5  LD   HL,(0B6F)   ; HL-be MUDDLE
  F9A8  EXX
  F9A9  LD   C,6A        ; C-be a szinkronbyte
  F9AB  CALL FA3D        ; a szinkronbyte felírása
  F9AE  LD   IY,(0E2C)   ; a kitöltendö memória kezdete
  F9B2  LD   A,(0E32)    ; A-ba a tömbtipus azonosító
  F9B5  LD   C,A
  F9B6  CALL FA3D        ; a tömbtipus azonosító felírása
  F9B9  LD   A,(0D14)    ; A-ba a file tipusbyte
  F9BC  LD   C,A
  F9BD  CALL FA3D        ; a file tipusbyte felírása
  F9C0  LD   A,(0E30)    ; A-ba az írásvédelmi byte
  F9C3  LD   C,A
  F9C4  CALL FA3D        ; az írásvédelmi byte felírása
  F9C7  LD   HL,(0E26)   ; HL-be a tömb mérete
  F9CA  LD   C,H
  F9CB  LD   A,L
  F9CC  OR   A
  F9CD  JR   Z,01        ; ha L=0, ugrás az F9D0H címre
  F9CF  INC  C
  F9D0  CALL FA3D        ; a szektorszám felírása
  F9D3  JR   05          ; ugrás az F9DAH címre
  F9D5  EXX
  F9D6  LD   HL,(0B6F)   ; HL-be MUDDLE
  F9D9  EXX

Egy szektor felírása:
  F9DA  LD   A,(0E31)    ; A-ba a szektormutató
  F9DD  LD   C,A
  F9DE  CALL FA3D        ; a szektorsorszám felírása
  F9E1  LD   A,H
  F9E2  OR   A
  F9E3  LD   D,L
  F9E4  JR   Z,02        ; ugrás az F9E8H címre, ha H=0
  F9E6  XOR  A
  F9E7  LD   D,A
```

```
  F9E8  LD   C,D          ; C-be a szektorban lévö adatbyte-ok
                            száma
  F9E9  CALL FA3D         ; C felírása
  F9EC  LD   C,(IY+00H)   ; C-be a felírandó adatbyte
  F9EF  INC  IY
  F9F1  DEC  HL
  F9F2  CALL FA3D         ; az adatbyte felírása
  F9F5  DEC  D
  F9F6  JR   NZ,F4        ; a szektor még nincs kész, ugrás az
                            F9ECH címre
  F9F8  LD   A,H
  F9F9  OR   L
  F9FA  LD   A,(0E2B)     ; A-ba az EOF_FLAG
  F9FD  JR   Z,01         ; a tömb kész, ugrás az FA00H címre
  F9FF  XOR  A            ; az EOF_FLAG=0
  FA00  LD   C,A
  FA01  CALL FA3D         ; az EOF_FLAG kiírása
  FA04  EX   DE,HL
  FA05  EXX

Az ellenörzö összeg felírása:
  FA06  LD   C,L
  FA07  CALL FA3D
  FA0A  LD   C,H
  FA0B  CALL FA3D

  FA0E  EXX
  FA0F  EX   DE,HL
  FA10  LD   (0E26),HL    ; a fel nem írt byte-ok száma
  FA13  LD   (0E2C),IY    ; a következö felírandó byte címe
  FA17  LD   A,(0E31)     ; A-ba a szegmens mutató
  FA1A  INC  A
  FA1B  LD   (0E31),A     ; az új szektorsorszám eltárolása
  FA1E  LD   A,H
  FA1F  OR   L
  FA20  JR   NZ,B3        ; a blokk még nincs felírva, ugrás az
                            F9D5H címre
A tömböt záró 5 periódus felírása:
  FA22  LD   BC,0501
  FA25  CALL FA31         ; az 5 periódus felírása
  FA28  EI                ; a megszakítás engedélyezése
  FA29  HALT              ; a hang megszakításra vár a rendszer
  FA2A  XOR  A
  FA2B  LD   (0E32),A     ; az EOF_FLAG törlése
  FA2E  JP   FABD         ; az IT kiszolgálás visszaállítása
```

```
Adott számú jelperiódus felírása:
  FA31  CALL FA5D
  FA34  CALL FA5D          ; egy periódus felírása
  FA37  DJNZ F8            ; ha B nem 0, ugrás az FA31H címre
  FA39  DEC  C
  FA3A  JR   NZ,F5         ; ha C nem 0, ugrás az FA31H címre
  FA3C  RET

Egy byte felírása:
  FA3D  LD   B,08
  FA3F  RRC  C             ; a felírandó byte C-ben volt
  FA41  CALL FA4F          ; egy fél periódus felírása
  FA44  BIT  7,C
  FA46  CALL FAF7          ; ellenörzö összeg számítás
  FA49  CALL FA4F          ; a másik fél periódus felírása
  FA4C  DJNZ F1            ; nem kész a byte, ugrás az FA3FH címre
  FA4E  RET

Egy fél periódus felírása:
  FA4F  BIT  7,C
  FA51  JR   Z,05          ; a felírandó periódus 0 bit, ugrás az
                             FA58H címre
  FA53  LD   A,(FAF4)      ; A-ba a PITCH alsó 8 bitje
  FA56  JR   0D            ; ugrás az FA65H címre
  FA58  LD   A,(FAF5)      ; A-ba a PITCH alsó 8 bitje
  FA5B  JR   08            ; ugrás az FA65H címre
  FA5D  LD   A,(FAF3)      ; A-ba a PITCH alsó 8 bitje
  FA60  JR   03            ; ugrás az FA65H címre
  FA62  LD   A,(FAF6)      ; A-ba a PITCH alsó 8 bitje

CTRL+ESC vizsgálat:
  FA65  EX   AF,AF'
  FA66  IN   A,58          ; a billentyüzetadat beolvasása
  FA68  AND  18
  FA6A  JP   Z,F8E6        ; CTRL+ESC volt, ugrás
  FA6D  EI                 ; megszakítás engedélyezése
  FA6E  HALT               ; hang IT-re várakozás
  FA6F  OUT  50,A          ; magnó adat ki
  FA71  EX   AF,AF'
  FA72  OUT  04,A          ; a PITCH alsó 8 bitjének beállítása
  FA74  IN   A,5B          ; frekvenciaosztó törlése
  FA76  OUT  07,A          ; hang IT törlése
  FA78  CALL F956          ; border csíkozás
  FA7B  RET
```

```
A megszakítás kiszolgálás módosítása:
  FA7C  DI              ; a megszakításkérés elfogadás letiltása
  FA7D  XOR  A
A bövítökártyák megszakításkérése elfogadásának tiltása:
  FA7E  OUT  58,A
  FA80  OUT  59,A
  FA82  OUT  5A,A
  FA84  OUT  5B,A

  FA86  OUT  04,A       ; PITCH alsó 8 bitjének törlése
  FA88  DEC  A          ; A-ba FFH
  FA89  LD   (0B71),A   ; SER_OK írása
  FA8C  LD   A,(0B11)   ; A-ba PORT03
  FA8F  AND  F0
  FA91  OR   07
  FA93  OUT  03,A       ; a 7. billentyüzetsor kiválasztása
  FA95  LD   A,(0B12)   ; A-ba PORT05
  FA98  AND  EF         ; a hangjel tiltás beállítása
  FA9A  LD   (0B12),A   ; PORT05 írása
  FA9D  AND  C0
  FA9F  OR   2F
  FAA1  OUT  05,A       ; PITCH felsö 4 bitjének beállítása, a
                          hang IT engedélyezése
  FAA3  XOR  A
  FAA4  LD   (0B14),A   ; SND_ACTIVE törlése
A videovezérlö regiszterének módosítása:
  FAA7  LD   A,0A
  FAA9  OUT  70,A       ; a vezérlö 10. regiszterének kiválasz-
                          tása
  FAAB  LD   A,23
  FAAD  OUT  71,A
  FAAF  IN   A,5B       ; a frekvenciaosztó törlése
  FAB1  OUT  07,A       ; kurzor/hang IT törlés
  FAB3  LD   HL,0038
  FAB6  LD   A,(HL)     ; A-ba az IT kiszolgáló rutin elsö byte-
                          ja
  FAB7  LD   (HL),C9    ; az IT kiszolgáló rutin egyetlen RET
  FAB9  LD   (0BF2),A   ; A elmentése
  FABC  RET

Az IT kiszolgálás visszaállítása:
  FABD  DI              ; megszakításkérés elfogadás tiltása
  FABE  CALL F6FF       ; motor kikapcsolás
  FAC1  LD   A,(0B11)   ; A-ba PORT03
  FAC4  OUT  03,A
```

```
  FAC6  LD   A,(0B12)    ; A-ba PORT05
  FAC9  OUT  05,A
  FACB  LD   A,(0B1F)    ; A-ba IRQ_STAT
  FACE  LD   B,A
  FACF  BIT  0,B
  FAD1  JR   Z,08        ; ugrás az FADBH címre, ha Z
A videovezérlö visszaprogramozása:
  FAD3  LD   A,0A
  FAD5  OUT  70,A        ; a 10. regiszter kiválasztása
  FAD7  LD   A,03
  FAD9  OUT  71,A
  FADB  LD   A,B
  FADC  RLCA
  FADD  RLCA
A bövítökártya IT visszaállítása:
  FADE  OUT  5B,A
  FAE0  RLCA
  FAE1  OUT  5A,A
  FAE3  RLCA
  FAE4  OUT  59,A
  FAE6  RLCA
  FAE7  OUT  58,A

  FAE9  LD   A,(0BF2)
  FAEC  LD   (0038),A    ; az IT kiszolgáló rutin elsö byte-jának
                           visszaállítása
  FAEF  XOR  A
  FAF0  SCF
  FAF1  EI               ; a megszakítás engedélyezése
  FAF2  RET

A kazettára írt jelek frekvenciáját meghatározó byte-ok:
  FAF3  DB   D6 DE CE BC

Az ellenörzö összeg számítása:
  FAF7  EXX
  FAF8  LD   A,80
  FAFA  JR   NZ,01       ; a kiírandó bit 1, ugrás az FAFDH címre
  FAFC  XOR  A           ; A-ba 0
  FAFD  XOR  H
  FAFE  RLA
  FAFF  JR   NC,09       ; ugrás az FB0AH címre, ha NC
  FB01  LD   A,H
  FB02  XOR  08
  FB04  LD   H,A
```

```
FB05  LD   A,L
FB06  XOR  10
FB08  LD   L,A
FB09  SCF
FB0A  ADC  HL,HL        ; HL-ben az új ellenörző összeg
FB0C  EXX
FB0D  RET
```

A 0B00-0B48 RAM terület

```
FB0E  DB   FF 01
FB10  DB   02 FF FF 05 06 FF 00 FF
FB18  DB   02 FF 04 05 06 FF C3 23
FB20  DB   0B 00 00 00 00 00 F5 3E
FB28  DB   70 D3 02 C3 12 C4 E3 7E
FB30  DB   23 E3 08 F5 3A 03 00 F5
FB38  DB   3E 70 32 03 00 D3 02 C3
FB40  DB   63 C3 08 F1 32 03 00 D3
FB48  DB   02 F1 08 C9 32 03 00 D3
FB50  DB   02 F1 FB C9

FB54  DB   'MOPS'
FB58  DB   03
FB59  DB   'VGB'
FB5C  DB   05
FB5D  DB   'RS232'
FB62  DB   04
FB63  DB   'DISK'
FB67  DB   'CISL'
```

A 128-160 kódú karaktereket definiáló táblázat

```
FB6B  DB   08 08 3E 6B 63 7F 63 63 00 00
FB75  DB   08 08 7E 68 60 7C 60 7E 00 00
FB7F  DB   08 08 3C 18 18 18 18 3C 00 00
FB89  DB   08 08 3E 6B 63 63 63 3E 00 00
FB93  DB   14 00 3E 63 63 63 63 3E 00 00
FB9D  DB   14 14 3E 63 63 63 63 3E 00 00
FBA7  DB   08 08 6B 63 63 63 63 3E 00 00
FBB1  DB   14 00 63 63 63 63 63 3E 00 00
FBBB  DB   14 14 77 63 63 63 63 3E 00 00
FBC5  DB   00 00 00 00 1F 1F 18 18 18 18
FBCF  DB   18 18 18 18 1F 1F 00 00 00 00
```

```
FBD9  DB   18 18 18 18 18 18 18 18 18 18
FBE3  DB   18 18 18 18 1F 1F 18 18 18 18
FBED  DB   00 00 00 00 FF FF 18 18 18 18
FBF7  DB   18 18 18 18 FF FF 18 18 18 18
FC01  DB   FF E7 C3 99 99 81 99 99 FF FF
FC0B  DB   08 08 08 3C 06 3E 66 3E 00 00
FC15  DB   08 08 08 3C 66 7E 60 3C 00 00
FC1F  DB   08 08 00 38 18 18 18 3C 00 00
FC29  DB   08 08 08 3C 66 66 66 3C 00 00
FC33  DB   00 24 00 3C 66 66 66 3C 00 00
FC3D  DB   24 24 00 3C 66 66 66 3C 00 00
FC47  DB   08 08 08 66 66 66 66 3E 00 00
FC51  DB   00 24 00 66 66 66 66 3E 00 00
FC5B  DB   24 24 00 66 66 66 66 3E 00 00
FC65  DB   00 00 00 00 F8 F8 18 18 18 18
FC6F  DB   18 18 18 18 F8 F8 00 00 00 00
FC79  DB   00 00 00 00 FF FF 00 00 00 00
FC83  DB   18 18 18 18 F8 F8 18 18 18 18
FC8D  DB   18 18 18 18 FF FF 00 00 00 00
FC97  DB   FF C3 99 9F 9F 9F 99 C3 FF FF
FCA1  DB   FF C3 99 9F C3 F9 99 C3 FF FF
FCAB  DB   00 00 00 00 00 00 00 00 00 00
```

A PRINT rutin része:

```
FCB5  EXX
FCB6  PUSH BC
FCB7  PUSH DE
FCB8  PUSH HL
FCB9  LD   HL,(172E)   ; HL-be az aktuális byte címe
FCBC  PUSH HL
FCBD  EX   SP,IY       ; IY-ba HL
FCBF  XOR  A           ; A-ba 0
FCC0  LD   B,A
FCC1  LD   C,A         ; BC-be 0
FCC2  EXX
FCC3  LD   H,A
FCC4  LD   L,A         ; HL-be 0
FCC5  LD   D,A
FCC6  LD   E,A         ; DE-be 0
FCC7  LD   B,A
FCC8  LD   C,A         ; BC-be 0
FCC9  LD   A,(IY+00H) ; A-ba a következő byte
FCCC  INC  IY
```

```
  FCCE  PUSH HL
  FCCF  PUSH BC
  FCD0  LD   HL,FF3F     ; HL-be a vezérlö karakterek táblázatá-
                           nak a címe
  FCD3  LD   BC,000B     ; a tábla mérete BC-be
  FCD6  CPIR
  FCD8  JR   NZ,6B       ; nem vezérlö karakter következik, ugrás
                           az FD45H címre
  FCDA  SLA  C           ; C=2*C
  FCDC  LD   HL,FF4A     ; HL-be a vezérlö karakterek ugrótábláza-
                           tának a címe
  FCDF  ADD  HL,BC       ; HL-be az adott vezérlökarakterhez tar-
                           tozó ugrótáblaelem cím
  FCE0  LD   A,(HL)
  FCE1  INC  HL
  FCE2  LD   H,(HL)
  FCE3  LD   L,A         ; HL-be a vezérlökarakter rutinjának cí-
                           me
  FCE4  POP  BC
  FCE5  EX   SP,HL       ; a STACK-be a rutin címe
  FCE6  LD   A,L
  FCE7  OR   A
  FCE8  RET              ; ugrás a vezérlökarakter rutinjára

A (.)-hoz tartozó rutin:
  FCE9  LD   A,40
  FCEB  JP   C,803E      ; itt üres utasítás

A (^)-hoz tartozó rutin:
***
  FCEC  LD   A,80
***
  FCEE  JR   NZ,57       ; ugrás az FD47H címre, ha NZ
  FCF0  PUSH AF
  FCF1  OR   C
  FCF2  LD   C,A
  FCF3  POP  AF          ; AF visszaállítása
  FCF4  XOR  03          ; A alsó két bitjének módosítása
  FCF6  AND  C
  FCF7  LD   C,A
  FCF8  JR   CF          ; ugrás az FCC9H címre

A (,)-höz tartozó rutin:
  FCFA  JR   NZ,4B       ; ugrás az FD47H címre, ha NZ
```

```
  FCFC  SET  1,C
  FCFE  JR   C9          ; ugrás az FCC9H címre

A (%) és (#) karakterekhez tartozó rutin:
  FD00  JR   NZ,45       ; ugrás az FD47H címre, ha NZ
  FD02  INC  L
  FD03  JR   C4          ; ugrás az FCC9H címre

A (*) és (>) karakterekhez tartozó rutin:
  FD05  INC  L
  FD06  INC  H
  FD07  EXX
  FD08  LD   A,B
  FD09  OR   C
  FD0A  EXX
  FD0B  JR   NZ,BC       ; ha BC' nem 0, ugrás az FCC9H címre
  FD0D  DEC  H
  FD0E  LD   A,(IY+FFH)  ; A-ba a vezérlökarakter
  FD11  CP   2A          ; (*)
  FD13  LD   E,A         ; E-be a karakter kódja
  FD14  JR   Z,B3        ; ugrás az FCC9H címre, ha (*) karakter
  FD16  LD   E,30
  FD18  JR   AF          ; ugrás az FCC9H címre

"Bad argument" hibajelzés:
  FD1A  LD   A,04        ; A-ba a hiba kódja
  FD1C  LD   HL,FD5C     ; a hibakezelö rutin címe a SYS. ROM-ban
  FD1F  JP   FFF0        ; ugrás a SYS. ROM-ba

A (<)-hoz tartozó rutin:
  FD22  INC  L
  FD23  EXX
  FD24  LD   A,B
  FD25  OR   C
  FD26  EXX
  FD27  JR   Z,A0        ; ugrás az FCC9H címre, ha BC'=0
  FD29  INC  H
  FD2A  JR   9D          ; ugrás az FCC9H címre

A (-)-hez tartozó rutin:
  FD2C  EXX
  FD2D  LD   A,B
  FD2E  OR   C
  FD2F  JR   NZ,04       ; ugrás az FD35H címre, ha BC' nem 0
```

```
  FD31  PUSH IY
  FD33  POP  BC
  FD34  DEC  BC          ; BC'-be az adott karakter címe
  FD35  EXX
  FD36  INC  D
  FD37  JR   90          ; ugrás az FCC9H címre

A (+)-hoz tartozó rutin:
  FD39  EXX
  FD3A  LD   A,B
  FD3B  OR   C
  FD3C  JR   NZ,DC       ; ugrás az FD1AH címre
  FD3E  PUSH IY
  FD40  POP  BC
  FD41  DEC  BC          ; BC' a (+) karakterre mutat
  FD42  EXX
  FD43  JR   84          ; ugrás az FCC9H címre
  FD45  POP  BC
  FD46  POP  HL
  FD47  EXX
  FD48  LD   A,B
  FD49  OR   C
  FD4A  JR   NZ,04       ; BC' nem 0, ugrás az FD50H címre
  FD4C  PUSH IY
  FD4E  POP  BC
  FD4F  DEC  BC          ; BC' az adott karakterre mutat
  FD50  POP  IY          ; IY visszaállítása
  FD52  EXX
  FD53  LD   (171C),HL   ; HL eltárolása
  FD56  LD   A,D
  FD57  CP   04
  FD59  JR   NC,0D       ; ugrás az FD68H címre, ha D>3
  FD5B  OR   A
  FD5C  JP   NZ,FD1A
  FD5F  LD   A,(IY+08H)  ; A-ba a BASIC STACK alján lévö szám ki-
                           tevö byte-ja
  FD62  AND  7F          ; az elöjelbit törlése
  FD64  SUB  3F
  FD66  ADD  A,H
  FD67  LD   L,A
  FD68  PUSH BC
  FD69  LD   A,(IY+06H)  ; A-ba a szám legfelsö byte-ja
  FD6C  OR   A
  FD6D  JR   Z,11        ; a szám 0, ugrás az FD80H címre
  FD6F  LD   A,(IY+08H)  ; A-ba a szám kitevö byte-ja
```

```
  FD72  AND  80          ; A-ba az elöjelbit
  FD74  JR   Z,0A        ; pozitív, ugrás az FD80H címre
  FD76  XOR  (IY+08H)
  FD79  LD   (IY+08H),A  ; az elöjelbit törlése
  FD7C  POP  BC
  FD7D  SET  0,C
  FD7F  LD   A,C1        ; itt üres utasítás
***
  FD80  POP  BC
***
  FD81  LD   A,(IY+08H)  ; A-ba a szám kitevö byte-ja
  FD84  SUB  3F
  FD86  LD   B,A
  FD87  EXX
  FD88  LD   E,C
  FD89  LD   D,B         ; DE'=BC'
  FD8A  EXX
  FD8B  XOR  A
  FD8C  LD   H,A
  FD8D  LD   L,A         ; HL=00H
  FD8E  OR   D
  FD8F  JR   NZ,05       ; ugrás az FD96H címre, ha NZ
  FD91  OR   B
  FD92  JP   M,FD96
  FD95  LD   L,B
  FD96  EXX
  FD97  DEC  DE
  FD98  OR   A
  FD99  SBC  HL,DE
  FD9B  ADD  HL,DE
  FD9C  LD   A,(DE)
  FD9D  EXX
  FD9E  JR   Z,02        ; ugrás az FDA2H címre, ha Z
  FDA0  JR   NC,19       ; ugrás az FDBBH címre, ha HL'>DE'
  FDA2  CP   2C          ; (,)
  FDA4  JR   Z,01        ; ugrás az FDA7H címre, ha A-ban (,) van
  FDA6  SCF
  FDA7  INC  D
  FDA8  DEC  D
  FDA9  JR   NZ,0B       ; ugrás az FDB6H címre, ha D nem 0
  FDAB  INC  L
  FDAC  DEC  L
  FDAD  JP   M,FDB8
  FDB0  JR   Z,06        ; ugrás az FDB8H címre, ha L=0
  FDB2  JR   NC,E2       ; ugrás az FD96H címre, ha NC
```

```
FDB4  DEC  L
FDB5  OR   A
FDB6  JR   NC,DE        ; ugrás az FD96H címre, ha NC
FDB8  INC  H
FDB9  JR   DB           ; ugrás az FD96H címre
FDBB  BIT  1,C
FDBD  JR   Z,01         ; ugrás az FDC0H címre, ha C 1. bitje 0
FDBF  DEC  H
FDC0  BIT  0,C
FDC2  JR   NZ,05        ; ugrás az FDC9H címre, ha C 0. bitje 1
FDC4  LD   A,C
FDC5  AND  FC
FDC7  JR   Z,01         ; ugrás az FDCAH címre, ha C<4
FDC9  DEC  H
FDCA  BIT  7,H
FDCC  JP   NZ,FD1A      ; ugrás, ha H>127
FDCF  LD   A,L
FDD0  DEC  A
FDD1  JP   P,FD1A
FDD4  INC  D
FDD5  DEC  D
FDD6  JR   NZ,14        ; ugrás az FDECH címre, ha D nem 0
FDD8  XOR  A
FDD9  BIT  7,B
FDDB  JR   NZ,01        ; ugrás az FDDEH címre, ha B>127
FDDD  LD   A,B
FDDE  LD   L,A
FDDF  OR   A
FDE0  JR   NZ,0E        ; ugrás az FDF0H címre, ha A nem 0
FDE2  INC  H
FDE3  DEC  H
FDE4  JR   Z,0A         ; ugrás az FDF0H címre, ha H=0
6  SET  2,C
FDE8  LD   L,01
FDEA  JR   04           ; ugrás az FD80H címre
FDEC  LD   L,H
FDED  LD   A,B
FDEE  SUB  H
FDEF  LD   B,A          ; B=B-H
FDF0  INC  D
FDF1  DEC  D
FDF2  JR   NZ,06        ; ugrás az FDFAH címre, ha D nem 0
FDF4  INC  E
FDF5  DEC  E
FDF6  JR   NZ,02        ; ugrás az FDFAH címre, ha E nem 0
```

```
FDF8  LD   E,20
FDFA  CALL NZ,FF62     ; a ($) karakter kiírása az aktuális
                         kurzor pozícióba, ha C 1. bitje 1
FDFD  INC  H
FDFE  DEC  H
FDFF  JR   Z,13        ; ugrás az FE14H címre, ha H=0
FE01  INC  D
FE02  DEC  D
FE03  JR   NZ,0F       ; ugrás az FE14H címre, ha D nem 0
FE05  BIT  2,C
FE07  JR   Z,05        ; ha C 2. bitje 0, ugrás az FE0EH címre
FE09  LD   A,H
FE0A  DEC  A
FE0B  CALL Z,FF60      ; ha H=1, egy karakter kiíratása
FE0E  LD   A,E
FE0F  CALL FF8D        ; az A-ban lévö karakter kiíratása
FE12  JR   EA          ; ugrás az FDFEH címre
FE14  LD   A,E         ; A-ba a karakter kódja
FE15  CP   20
FE17  CALL Z,FF62      ; ha szóköz, ugrás
FE1A  EXX
FE1B  LD   E,C
FE1C  LD   D,B         ; DE'=BC'
FE1D  EXX
FE1E  INC  B
FE1F  DEC  B
FE20  JR   Z,03        ; ugrás az FE26H címre, ha B=0
FE22  JP   P,FE29
FE25  LD   A,L
FE26  DEC  A
FE27  JR   Z,16        ; ugrás az FE3FH címre, ha Z
FE29  INC  L
FE2A  DEC  L
FE2B  JR   Z,12        ; ugrás az FE3FH címre, ha L=0
FE2D  EXX
FE2E  DEC  DE
FE2F  LD   A,(DE)      ; A-ba a következö karakter
FE30  CP   2C          ; (,)
FE32  JR   Z,FA        ; ugrás az FE2EH címre, ha (,) karakter
FE34  EXX
FE35  JR   F3          ; ugrás az FE2AH címre
FE37  CALL FF96        ; 0 szám a BASIC STACK-be
FE3A  LD   B,04
FE3C  LD   C,B
FE3D  JR   33          ; ugrás az FE72H címre
```

```
  FE3F  PUSH DE
  FE40  LD   C,B
  FE41  PUSH BC
  FE42  LD   E,B
  FE43  DEC  E
  FE44  LD   A,(IY+08H) ; A-ba a szám kitevö byte-ja
  FE47  SUB  3B
  FE49  LD   C,A
  FE4A  SUB  04
  FE4C  LD   HL,(171C)  ; HL visszaállítása
  FE4F  DEC  H
  FE50  ADD  A,H
  FE51  JP   M,FE37
  FE54  CP   H
  FE55  JR   NC,03      ; ugrás az FE5AH címre, ha NC
  FE57  LD   A,H
  FE58  LD   C,04
  FE5A  ADD  A,04
  FE5C  LD   B,A
  FE5D  LD   L,B
  FE5E  DEC  B
  FE5F  LD   A,B
  FE60  CP   10
  FE62  LD   A,00
  FE64  CALL C,FF87     ; ha B<16, ugrás a SYS. ROM-ba az F7CEH
                          címre
  FE67  OR   A
  FE68  JR   NZ,08      ; ha A nem 0, ugrás az FE72H címre
  FE6A  LD   A,B
  FE6B  DEC  A
  FE6C  CP   C
  FE6D  JR   NC,EF      ; ugrás az FE5EH címre, ha C<=A
  FE6F  LD   L,00
  FE71  JP   NC,0C48    ; itt üres utasítás
***
  FE72  LD   C,B
  FE73  INC  C
***
  FE74  LD   B,04
  FE76  EXX
  FE77  EX   DE,HL
  FE78  OR   A          ; a CARRY flag törlése
  FE79  SBC  HL,BC
  FE7B  ADD  HL,BC
  FE7C  EX   DE,HL
  FE7D  JR   Z,17       ; ugrás az FE96H címre, ha HL'=BC'
```

```
FE7F  LD   A,(DE)      ; A-ba a következő karakter
FE80  INC  DE
FE81  EXX
FE82  CP   2C          ; (,)
FE84  JR   Z,0B        ; ugrás az FE91H címre, ha (,)
FE86  LD   A,B
FE87  CP   10
FE89  LD   A,00
FE8B  CALL C,FF87      ; ha B<16, ugrás a SYS. ROM-ba az F7CEH
                         címre
FE8E  ADD  A,30        ; A-ba a digit karakterének kódja
FE90  INC  B
FE91  CALL FF8D        ; a karakter kiíratása
FE94  JR   E0          ; ugrás az FE76H címre
FE96  LD   A,(BC)      ; A-ba a következő karakter
FE97  CP   2E          ; (.)
FE99  JR   NZ,45       ; nem pont karakter, ugrás az FEE0H cím-
                         re
FE9B  CALL FF8D        ; a karakter kiíratása
FE9E  INC  BC
FE9F  LD   A,(BC)      ; A-ba a következő karakter
FEA0  CP   2C          ; (,)
FEA2  JR   Z,F7        ; (,) karakter következett, ugrás az
                         FE9BH címre
FEA4  EXX
FEA5  LD   A,D
FEA6  OR   A
FEA7  JR   NZ,0C       ; ugrás az FEB5H címre, ha D nem 0
FEA9  LD   A,(171D)    ; A-ba az eltárolt H
FEAC  CPL
FEAD  CP   E
FEAE  JR   NC,05       ; ugrás az FEB5H címre, ha NC
FEB0  XOR  A           ; A-ba 0
FEB1  INC  E
FEB2  JP   M,FEDB
FEB5  EXX
FEB6  LD   A,(BC)      ; A-ba a következő byte
FEB7  CP   2A          ; (*)
FEB9  JR   Z,0B        ; ugrás az FEC6H címre, ha (*) karakter
FEBB  CP   25          ; (%)
FEBD  JR   Z,07        ; ugrás az FEC6H címre, ha (%) karakter
FEBF  CP   23          ; (#)
FEC1  JR   NZ,1D       ; ugrás az FEDFH címre, ha nem (#) ka-
                         rakter
FEC3  EXX
```

```
FEC4  JR   0C          ; ugrás az FED2H címre
FEC6  EXX
FEC7  LD   A,B
FEC8  SUB  C
FEC9  JR   C,07        ; ugrás az FED2H címre, ha C>B
FECB  OR   L
FECC  JR   Z,0C        ; ugrás az FEDAH címre, ha Z
FECE  LD   A,F0
FED0  JR   08          ; ugrás az FEDAH címre
FED2  LD   A,B
FED3  CP   10
FED5  LD   A,00
FED7  CALL C,FF87      ; ha B<16, ugrás a SYS. ROM-ba az F7CEH
                         címre
FEDA  INC  B
FEDB  EXX
FEDC  ADD  A,30        ; A-ba a digit karakterkódja
FEDE  JR   BB          ; ugrás az FE9BH címre
FEE0  LD   A,(BC)      ; A-ba a karakterkód
FEE1  CP   5E          ; (^)
FEE3  EXX
FEE4  POP  BC
FEE5  POP  DE
FEE6  LD   B,C
FEE7  JR   NZ,47       ; ugrás az FF30H címre, ha nem (^)
FEE9  LD   A,(IY+06H)  ; A-ba a szám legfelsö mantissza byte-ja
FEEC  OR   A
FEED  JR   NZ,01       ; ugrás az FEF0H címre, ha nem 0 a szám
FEEF  LD   B,A
FEF0  LD   A,45        ; A-ba az "E" kódja
FEF2  CALL FF8D        ; a karakter kiíratása
FEF5  BIT  7,B
FEF7  LD   A,2B        ; A-ba "+" kódja
FEF9  JR   Z,06        ; ugrás az FF01H címre, ha Z
FEFB  LD   A,B
FEFC  NEG
FEFE  LD   B,A
FEFF  LD   A,2D        ; A-ba "-" kódja
FF01  CALL FF8D        ; a karakter kiíratása
FF04  DEC  D
FF05  LD   A,D
FF06  CP   04
FF08  JR   C,0A        ; ha D<4, ugrás az FF14H címre
FF0A  LD   A,30        ; A-ba "0" kódja
FF0C  CALL FF8D        ; karakter kiíratása
```

```
FF0F  EXX
FF10  INC  BC
FF11  EXX
FF12  JR   F0          ; ugrás az FF04H címre
FF14  LD   A,B
FF15  LD   BC,0AFF     ; B-be 10
FF18  SUB  B
FF19  INC  C
FF1A  JR   NC,FC       ; ugrás az FF18H címre, ha NC
FF1C  ADD  A,B
FF1D  LD   B,A
FF1E  LD   A,C
FF1F  ADD  A,30        ; A-ba a számjegy kódja
FF21  CALL FF8D        ; a karakter kiíratása
FF24  LD   A,B
FF25  ADD  A,30        ; A-ba a számjegy kódja
FF27  CALL FF8D        ; karakter kiíratása
FF2A  EXX
FF2B  INC  BC
FF2C  INC  BC
FF2D  INC  BC
FF2E  INC  BC
FF2F  EXX
FF30  EXX
FF31  LD   (172E),BC   ; BC eltárolása
FF35  POP  HL
FF36  POP  DE
FF37  POP  BC
FF38  EXX
FF39  LD   HL,E67A
FF3C  JP   FFF0        ; ugrás a SYS. ROM-ba a PRINT rutinba
```

Vezérlökarakterek táblázata:

```
FF3F  DB   2B   ; +
FF40  DB   2D   ; -
FF41  DB   24   ; $
FF42  DB   3C   ; <
FF43  DB   3E   ; >
FF44  DB   2A   ; *
FF45  DB   25   ; %
FF46  DB   23   ; #
FF47  DB   2C   ; ,
FF48  DB   5E   ; ^
FF49  DB   2E   ; .
```

A vezérlőkarakterekhez tartozó rutinok címei:

```
FF4A  DW   FD39
FF4C  DW   FD2C
FF4E  DW   FCC9
FF50  DW   FD22
FF52  DW   FD05
FF54  DW   FD05
FF56  DW   FD00
FF58  DW   FD00
FF5A  DW   FCFA
FF5C  DW   FCEC
FF5E  DW   FCE9
```

```
FF60  LD   E,30
FF62  BIT  1,C
FF64  LD   A,24          ; A-ba "$" kódja
FF66  CALL NZ,FF8D       ; karakter kiíratása, ha NZ
FF69  BIT  0,C
FF6B  LD   A,2D          ; A-ba "-" kódja
FF6D  JR   NZ,1E         ; ugrás az FF8DH címre, ha C 0. bitje 1
FF6F  LD   A,C
FF70  AND  F8
FF72  ADD  A,A
FF73  LD   A,20          ; A-ba a szóköz kódja
FF75  JR   C,16          ; ugrás az FF8DH címre, ha CARRY
FF77  RET  P
FF78  LD   A,2B          ; A-ba "+" kódja
FF7A  JR   11            ; ugrás az FF8DH címre
FF7C  DEC  A
FF7D  RET  M
FF7E  PUSH AF
FF7F  LD   A,20          ; A-ba a szóköz kódja
FF81  CALL FF8D          ; karakter kiíratása
FF84  POP  AF
FF85  JR   F5            ; ugrás az FF7CH címre
FF87  PUSH HL
FF88  LD   HL,F7CE       ; HL-be egy SYS. ROM-beli cím
FF8B  JR   04            ; ugrás az FF91H címre
FF8D  PUSH HL
FF8E  LD   HL,FEB3       ; a karakter kiíró rutin címe a SYS.
                           ROM-ban
FF91  CALL FFF0          ; ugrás a SYS. ROM-ba
```

```
FF94  POP   HL
FF95  RET
FF96  LD    HL,FA14      ; HL-ba SYS. ROM-beli cím
FF99  JP    FFF0         ; ugrás a SYS. ROM-ba
FF9C  LD    HL,(172E)
FF9F  LD    A,(HL)       ; A-ba a következö karakter
FFA0  LD    DE,0000
FFA3  CP    3C           ; (<)
FFA5  JR    Z,06         ; A-ban "<" van, ugrás az FFADH címre
FFA7  INC   E
FFA8  CP    3E
FFAA  JR    NZ,06        ; A-ban nem ">" van, ugrás az FFB2H cím-
                           re
FFAC  INC   E
FFAD  INC   D
FFAE  INC   HL
FFAF  LD    (172E),HL    ; HL eltárolása
FFB2  LD    A,(HL)       ; A-ba a következö karakter
FFB3  CP    23
FFB5  JR    Z,F6         ; A-ban "#" van, ugrás az FFADH címre
FFB7  CP    25
FFB9  JR    Z,F2         ; A-ban "%" van, ugrás az FFADH címre
FFBB  CP    2A
FFBD  JR    Z,EE         ; A-ban "*" van, ugrás az FFADH címre
FFBF  LD    A,(IY+01H)   ; A-ba a BASIC STACK alján lévö karakter
                           hossza
FFC2  SUB   D
FFC3  NEG
FFC5  JP    M,FD1A
FFC8  DEC   E
FFC9  JP    M,FFD2
FFCC  LD    C,A
FFCD  JR    NZ,02        ; ugrás az FFD1H címre, ha NZ
FFCF  SRL   C
FFD1  SUB   C
FFD2  PUSH  AF
FFD3  LD    A,C
FFD4  CALL  P,FF7C
FFD7  POP   AF
FFD8  LD    HL,E64F      ; SYS. ROM-beli cím HL-be
FFDB  JP    FFF0         ; ugrás a SYS. ROM-ba

FFDE  DB    '(C)1985ISL'

FFE8  LD    A,04         ; A-ba a BAUD alapértéke
```

```
  FFEA  JP   F29C        ; a soros vonal inicializálása
  FFED  NOP
  FFEE  NOP
  FFEF  NOP

Vissza a SYSTEM ROM-ba:
  FFF0  PUSH AF
  FFF1  LD   A,70
  FFF3  LD   (0003),A
  FFF6  OUT  02,A        ; memórialapozás: U U U S lapkiosztás

Ugrás a SYSTEM ROM-ból meghívott rutinra:
  FFF8  POP  AF
  FFF9  JP   (HL)

  FFFA  DB   7B E6 1F
  FFFD  DB   1B FE 04
```

ROM-rutinok

Ebben a fejezetben a fontosabb, az érdeklödésre számot tartó ROM-rutinok belépési címe, rövid leírása, jellegzetessége található. Mint az ismeretes, a BASIC 1.2 20 KByte-nyi memóriát foglal el. Az is tudott, hogy a Videoton TV-Computer a memóriát 16 KByte-os lapokban kezeli. A teljes ROM így több, mint egy lapot foglal el: egy teljes lapot és egy másik lap felsö negyedét. A ROM szervezése olyan, hogy a BASIC alatt a 3. lap mindig ROM, ezért a ROM-rutinok címét is ehhez igazítottam annak ellenére, hogy bizonyos esetekben a ROM a 0. lapra is belapozódik.

A rendszer ROM fö részének rutinjai

C000 A számítógép bekapcsolásakor ez a 0000h cím. Kerüljük az ide ugrást.

C229 BASIC hidegindítás

C306 Programmodul vizsgálat. Ha az oldalt bedugható programmodul elsö négy byte-ja a "MOPS" karakterek, a vezérlés átadódik a modul ötödik byte-jára.

C321 JP (HL) utasítás

C363 A RST 30H belépési címe

C371 Az RST 30H rutin "vége"

C412 A RST 38H rutin belépési címe. Mivel a processzor 1-es megszakítási módban dolgozik, ez a rutin végzi a megszakítás kiszolgálást.

C437 A megszakítás kiszolgáló rutin magja.

C4B7 A HI_MEM_SET rutin. A rutin a DE regiszterben várja a HI_MEM felett lefoglalni kívánt byte-ok számát. Visszatéréskor a DE regiszter az új HI_MEM+1 értéket tartal-

maz, ha van elegendö memória, ellenkezö esetben az A regiszterben FBh-t ad vissza a rutin.

C4D0 Az EXP_DIRECT rutin. A rutin a DE regiszterben várja a kártyaazonosító címét, a C regiszterben az egységszámot és a B regiszterben a kijelölés irányát (FFH esetén input kártya kijelölés történik, különben output). Ha a csatolókártya nem létezik, az A regiszterben FDh értékkel tér vissza a rutin. A rutin meghívja a SLOT_NUM rutint.

C4E2 Az I/O_DIRECT rutin. A rutin a B regiszterben várja a funkcióosztály számát, a C regiszterben az eszköz számot, a D regiszterben pedig a hozzárendelés irányát (FFH jelenti az input hozzárendelést). Hozzárendelési hiba esetén az A regiszterben FEH értékkel tér vissza a rutin.

C50E A SLOT_NUM rutin. A DE regiszterben a kártyaazonosító címét, a C regiszterben pedig az egységszámot várja a rutin. Nemlétezö kártya esetén az A regiszterben FDH-t kapunk vissza, különben pedig a C regiszterben kapjuk meg a keresett kártya csatlakozószámát.

C98F Ez a rutin elöször egy (U0 U1 VID SYS) lapozást hajt végre, a STACK-ben kicseréli a visszatérési címet egy önmagába mutató címre, és egy JP (HL) utasítással adja vissza a vezérlést annak a rutinnak, amely öt hívta. Az öt hívó rutinnak RET utasítással kell végzödni, hogy a RET utasítás hatására visszaadódhasson ide a vezérlés. Ekkor a rutin visszaállítja a hívást megelözö memória kiosztást, és visszatér az öt hívó rutint hívó rutinba.

C9B3 A rutin megvizsgálja, hogy a BC regiszterben megadott X koordináta létezö-e. A Carry flag jelzi az eredményt : túl nagy érték esetén átvitelt jelez.

C9BA A rutin megvizsgálja, hogy a DE regiszterben megadott Y koordináta létezö-e. A Carry flag jelzi az eredményt :

C9F2 Négyszínü grafikus üzemmód beállítása.

C9F4 A VMODE rutin.

CA38 A PAL rutin.

CA49 A CLS rutin.

CA4F Csak a grafikus képernyöt és változókat törli, az editor munkaterületét nem változtatja meg.

CA65 Az abszolut tollpozíció beállítása. Ez a rutin nem ellenörzi, hogy a pozíció létezik-e, és nem húz vonalat.

CAD7 A BREL rutin.

CADA A BABS rutin.

CB61 HL és DE különbségének abszolutértékét veszi, de elötte a Carry flag-t törölni kell. A különbség a HL regiszben lesz. Az A regiszter legfelsö bitje jelzi, hogy melyik regiszter tartalma volt a nagyobb: 0 esetén HL>DE, 1 esetén DE>HL.

CB6D A beállított vonaltipusnak megfelelö " minta byte-ot " kikeresi, és a B83h címre teszi, elökészítve azt a vonal húzáshoz.

CBF3 A BON rutin.

CBFC Leteszi a tollat, de nem rajzol pontot.

CBFF A BOFF rutin.

CC86 A VID_BKOUT rutin.

CC94 A VID_CHOUT rutin.

CCA8 VID_CHOUT rutin, de nem lehet a karakter kódja kisebb, mint 20h (decimális 32).

CCAD VID_CHOUT rutin a 20h-DFh (32-223) kódú karakterekre.

CCD7 Az elözö rutin, de a sugár (toll) pozíciót meg kell adni a BC és DE regiszterekben: az X pozíciót BC-be az Y pozíciót a DE-be kell tenni.

CD2B A 0Dh (RETURN) karakter kiíratása a VIDEO rutinnal.

CD31 Soremelés a VIDEO rutinnal.

CD48 A FILL rutin.

CF2C A DEFC rutin.

CF3F Karakter definiálás. Ez a rutin nem ellenöriz, és a HL regiszterben a (karakter kód-128) tízszeresét várja.

CF4B A BTEXT rutin.

CFA3 A ED_INT rutin. A kurzort villogtatja.

CFD4 Törli, alaphelyzetbe állítja az editor munkaterületet.

D013 A CFIX rutin.

D01D A CPOS rutin.

D035 A kurzor pozícionálását végzi, de semmit nem ellenöriz. Az oszlop pozíciót a H regiszterben, a sor pozíciót az L regiszterben várja a rutin.

D041 Az ED_BKIN/OUT rutinok.

D046 Az ED_BKOUT rutin elágazási pontja. Ha a PAPER és az INK rendszerváltozókat nem módosítottuk, ez a rutin az ED_BKOUT rutinnal azonos hatású.

D04C Az ED_BKIN rutin elágazási pontja. Ha a PAPER és az INK rendszerváltozókat nem módosítottuk, ez a rutin az ED_BKIN rutinnal azonos hatású.

D052 Az ED_CHIN/OUT rutinok.

D055 Az ED_CHOUT rutin elágazási pontja.

D058 Ha a PAPER és az INK rendszerváltozókat nem módosítottuk, ez a rutin az ED_CHIN rutinnal azonos hatású.

D0B7 Ha a PAPER és az INK rendszerváltozókat nem módosítottuk, ez a rutin az ED_CHOUT rutinnal azonos hatású.

A BASIC beépített szövegszerkesztőjének funkciói

D157 A szövegszerkesztő rutin címének meghatározása. A vezérlő karakter kódját az A regiszterben kéri a rutin. Az aktuális kurzor pozíció a BC regiszterpárba kerül, az A regiszterben pedig a grafikus felbontástól függően a soronkénti maximális karakterszám.

D191 A "kurzor balra" rutin.

D195 A "kurzor fel" rutin.

D198 A "kurzor jobbra" rutin.

D19E A "kurzor le" rutin.

D1A8 A "törlés a sor végéig" rutin.

D1CD A "tab" rutin.

D1F7 A "sor beszúrás" rutin.

D1FD A "sor törlés" rutin.

D205 Az "egy karakter beszúrás" rutin.

D278 A "karakter törlés balra" rutin.

D287 A "karakter törlés a kurzor pozícióban" rutin.

D5EC A "Billentyüzet" eszköz iniciáló rutinja. Beállítja a DELAY_KEY, LOCK_KEY, RATE_KEY változókat, törli a PICTURE és OLDPIC területeket.

D612 A KB_STATUS rutin.

D618 A KB_CHIN rutin.

D61F Ez a rutin nem vár billentyü megnyomásra, a legutoljára megnyomott billentyü kódját tartalmazó pufferből olvas. A HL által mutatott címre 0-t tesz.

D62D A KB_INT rutin. Ez a rutin végzi a billentyüzet mátrix leolvasását.

D906 A PAR_BKOUT rutin.

D90C A PAR_CHOUT rutin.

D913 PAR_CHOUT rutin, de Ctrl-Esc megnyomásra érzéketlen.

D933 A TONE_INT rutin.

D961 A TON_SET rutin.

D96E TON_SET rutin, de az új hang mindenképp megszakítja a régit.

D978 TON_SET rutin, de az új hang mindenképp megszakítja a régit, és a Ctrl-Esc hatástalan.

D9C8 A CAS_OPEN és CAS_CRTE rutinok.

D9CD A CAS_CLOSE rutin.

D9D2 A CAS_CHIN/OUT rutinok.

D9D7 A CAS_BKIN/OUT rutinok.

D9DC A CAS_VERIFY rutin.

DADA A Basic interpreter újraindulási pontja parancs, program végrehajtása után.

DB06 Az OK rutin. Ez a rutin Basic parancs vagy programsor begépelésére vár.

DB80 Az adott utasítás sor következö utasításának értelmezése.

DBBB A REM rutin.

DD65 A CONTINUE rutin.

DD80 A LLIST rutin.

DD85 A LIST rutin.

DD93 A DELETE rutin.

DE08 A NEW rutin.

DE1B A RUN rutin.

DE31 A TRACE rutinok.

DE40 A TRACE OFF rutin.

DE46 A TRACE ON rutin.

DFF2 A DATA rutin.

DFFC A CLS Basic rutin.

E002 A DEF rutin.

E053 A DIM rutin.

E104 Az ELSE rutin.

E10E Az END rutin.

E117 Az EXT rutin.

E15C A FOR rutin.

E1CB Az INPUT rutin.

E1DD Az INPUT PROMPT rutin.

E21B A READ rutin.

E2EE Az IF rutin.

E332 Az ON rutin.

E382 A GOSUB rutin.

E3B2 A GOTO rutin.

E3C1 A LET rutin.

E452 A LOMEM rutin.

E4B6 A NEXT rutin.

E542 Az OUT rutin.

E553 A POKE rutin.

E570 Az LPRINT rutin.

E573 Az OUTPUT és PRINT rutin.

E6C7 A RANDOMIZE rutin.

E6F4 A RESTORE rutin.

E70F A RETURN rutin.

E754 A PLOT rutin.

E75A A PLOT PAINT rutin.

E733 A GRAPHICS rutin.

E790 A SET rutinok.

E7F7 A SET MODE rutin.

E7FA A SET STYLE rutin.

E7FD A SET INK rutin.

E800 A SET PAPER rutin.

E803 A SET DELAY rutin.

E806 A SET RATE rutin.

E816 A SET CHARACTER rutin.

E821 A SET PALETTE rutin.

E82A A SET BORDER rutin.

E833 A SOUND rutin.

E882 PITCH beállítás a SOUND rutinhoz.

E88B DURATION beállítása a SOUND rutinhoz.

E88C VOLUME beállítása a SOUND rutinhoz.

E89B A Basic CLOSE rutin.

E8C9 Az OPEN Basic rutin.

E910 A GET rutin.

E951 A LOAD rutin.

E982 A SAVE rutin.

E99F Ez a rutin egy írásra megnyitott file-t ment ki lemezre vagy kazettára programfile-ként. A DE regiszterben várja a rutin a kimentendö memória kezdetét, a HL regiszter pedig a végét. A rutin zárja is a file-t. A file 16 byte-os fejlécére az SP regiszter mutat.

E9D3 A VERIFY rutin.

E9EB Ez a rutin egy filenevet vár a FILENAME_PUFFER -ben, és azt a file-t írásra megnyitja.

EA68 14. aritmetikai rutin: a Basic beépített függvényének az argumentumát a Basic Stack-be helyezi.

EA82 5. aritmetikai rutin: a ROM-ban tárolt n. konstanst a Basic Stack-be teszi. "n" értékét (0-39) a rutin sorszámát meghatározó byte-ot követö byte értéke határozza meg.

EA9A 7. aritmetikai rutin: az Y aritmetikai regisztert (7 byte) a Basic Stack-be helyezi.

EA9F 6. aritmetikai rutin: az X aritmetikai regisztert (7 byte) a Basic Stack-be helyezi.

EAA2 A HL regiszter által mutatott számot (7 byte) a Basic Stack-be helyezi (9 byte).

EABE 11. aritmetikai rutin: a Basic Stack-böl az Y aritmetikai regiszterbe mozgat. A Basic Stack-böl 9 byte felszabadul.

EAC3 10. aritmetikai rutin: a Basic Stack-ből az X aritmetikai regiszterbe mozgat. A Basic Stack-ből 9 byte felszabadul.

EAC6 A Basic Stack-ből a DE regiszter által mutatott helyre mozgat. A Basic Stack-ből 9 byte felszabadul.

EACD 9. aritmetikai rutin: a Basic Stack-ből az Y aritmetikai regiszterbe másol. A Stack mutató (IY regiszter) értéke nem változik.

EAD2 8. aritmetikai rutin: a Basic Stack-ből az X aritmetikai regiszterbe másol. A Stack mutató (IY regiszter) értéke nem változik.

EAD5 A Basic Stack-ből a DE regiszter által mutatott helyre mozgat. A Basic Stack nem változik.

EAE9 Az ABS beépített függvény.

EAEC A Basic Stack-ben lévő 9 byte-on tárolt szám abszolutértékét veszi.

EB88 A CHR$ beépített függvény.

EBA4 A COS beépített függvény.

EBB5 Az EXP beépített függvény.

EC4C A FREE beépített függvény.

EC71 Az IN beépített függvény.

ECAC AZ INT beépített függvény.

ECEB Az IO beépített függvény.

ED07 A LEN beépített függvény.

ED1E A LOG beépített függvény.

EDC6 A PEEK beépített függvény.

EDEE Az RND beépített függvény.

EE3B A SGN beépített függvény.

EE5B A SIN beépített függvény.

EEC7 A SQR beépített függvény.

EF94 A STR$ beépített függvény.

EFAB A STRING$ beépített függvény.

EFF8 A TAN beépített függvény.

F013 A USR beépített függvény.

F03D A VAL beépített függvény.

F073 A VARPTR beépített függvény.

F09E A VERNUM beépített függvény.

F48E 3. aritmetikai rutin: a Basic Stack-ban egymás alatt tárolt két 9 byte-os szám különbségét veszi (elsö operandus-második operandus). Az eredmény az elsö operandus helyén keletkezik.

F493 0. aritmetikai rutin: a Basic Stack-ben egymás alatt levö két 9 byte-on tárolt számot összeadja. Az eredmény az elsönek letett operandus helyén keletkezik.

F512 2. aritmetikai rutin: a Basic Stack-ben egymás alatt tárolt két 9 byte-os szám szorzatát veszi. Az eredmény az elsö operandus helyén keletkezik.

F5FB 1. aritmetikai rutin: a Basic Stack-ben egymás alatt lévö két 9 byte-on tárolt szám hányadosát veszi (elsö operandus/második operandus). Az eredmény az elsö operandus helyén keletkezik.

F726 4. aritmetikai rutin: a Basic Stack-ben levö 9 byte-os szám elöjelét az ellenkezöjére változtatja.

FA63 Basic numerikus változó értékét tölti a Basic Stack-be.

FA92 12. aritmetikai rutin: a Basic Stack alsó 9 byte-ját a Stack-re helyezi mégegyszer.

FB28 13. aritmetikai rutin: a Basic Stack-böl a HL regiszter által mutatott helyre mozgat. A Stackböl felszabadul 9 byte.

FC43 Az interpreter parancs értelmezö rutinja, az A regiszterbe tölti a forrásprogram következö elemét. A forrásprogram aktuális elemére a HL' regiszter mutat.

FE79 A rutin hívás utáni byte-on kezdödö szöveg kiíratása a képernyöre. A szöveg elsö byte-ja adja meg a szöveg hosszát. A kiíratás után a vezérlés a szöveg utáni elsö byte-ra adódik vissza.

FE7F A HL regiszter által mutatott szöveg kiíratása a képernyöre. A szöveg elsö byte-ja adja meg a szöveg hosszát.

FF4F A Basic interpreter input rutinja (parancsra, utasításra vár az interpreter).

FF9D A STOP_FLAG vizsgálata. Ha nem 0, a rutin BASIC STOP utasítást hajt végre.

FFA3 A STOP rutin.

FFF0 Belapozza a ROM bövítést a harmadik lapra, a ROM fö részét a nulladik lapra, és egy JP (HL) utasítást hajt végre.

Rendszer ROM bövítés

Ez a 4 Kbyte ROM szerves része a VTC ROM-jának, nem ROM bövítés. Azért nevezem mégis bövítésnek, mert ezen a memórialapon alapkiépítésben 12 Kbyte-tal bövíthetö a rendszer memóriája. Ez a bövítés lehet RAM is és ROM is akár vegyesen is. Az alsó 8 K a bövitö kártyákon helyezhetö el (pl. itt van a floppy csatoló ROM-ja és RAM-ja), 4 K pedig az alaplapon bövíthetö.

A P3 lapra lapozva ezt a memórialapot, az F000h címtöl kezdödik a rendszer ROM e része. A két ROM rész közötti vezérlés átadás rendszerint úgy történik, hogy a cél ROM a P3 lapra, a start ROM a P0 lapra lapozódik, és egyszerü JP utasítással adódik át a vezérlés.

Ez a ROM rész néhány inicializáló rutint, a soros vonal kezelö rutinokat, valamint a kazetta rutinokat tartalmazza néhány ROM táblázat mellett.

F000 Inicializáló rutin, mely nem önálló rutin (nem RET utasítással végzödik). Néhány táblázatot tölt át a RAM-ba, inicializálja a bövitö kártyákat, és visszatér a ROM fö részébe.

F13D Az inicializáló rutin része. RESET esetén ellenörzi a RAM bizonyos rutinjainak sértetlenségét. A vizsgálat eredményétöl föggöen hajtódik végre meleg-RESET vagy hideg-RESET.

F166 Az RST 30h rutin része, a bövítökártya rutinok hívása esetén a hívott rutin címének meghatározása, a kiválasztott rutin meghívása INPUT rutinok esetén.

F16C u.a., mint az F166h, de OUTPUT rutinok esetén.

F1E6 Bövítökártya rutin hívás hibakezelö rutinjának része: a funkcióosztályokhoz visszarendeli a saját eszközöket.

F227 A megszakításkiszolgáló rutin része, ha a megszakításkiszolgáló rutin a bövítökártyán van.

F29C A soros vonal inicializáló rutinja.

F2A4 A SER_SET rutin.

F339 A SER_CHIN/OUT rutinok.

F33C A SER_CHOUT rutin.

F357 A SER_CLOSE rutin.

F359 A SER_CHIN rutin.

F3B7 A SER_BKIN/OUT rutinok.

F3BA A SER_BKOUT rutin.

F3C0 A SER_BKIN rutin.

F3D8 A CAS_CHIN/OUT rutinok belépési pontja a ROM bövítésben.

F3DD A CAS_BKIN/OUT rutinok belépési pontja a ROM bövítésben.

F3E2 A CAS_OPEN, CAS_CRTE rutinok belépési pontja a ROM bövítésben.

F3E7 A CAS_CLOSE rutinok belépési pontja a ROM bövítésben.

F3EC A CAS_VERIFY rutin belépési pontja a ROM bövítésben.

F3FA A CAS_OPEN, CAS_CRTE szubrutinok.

F3FD A CAS_OPEN szubrutin.

F4CF A CAS_CRTE szubrutin.

F526 A DE által mutatott file név átmozgatása a HL által mutatott címre. A rutin a kisbetüket nagybetükké alakítja. A file név csak a megengedett karaktereket tartalmazhatja.

F527 A HL által mutatott file név átmozgatása a DE által mutatott címre. A file név csak a megengedett karaktereket tartalmazhatja. A rutin a kisbetüket nagybetükké konvertálja.

F54F A HL által mutatott címtöl kezdve a BC-ben tárolt számnak megfelelö számú byte feltöltése 0-val.

F558 A CAS_CLOSE szubrutinok.

F55B A CAS_CLOSE szubrutin olvasásnál.

F585 A CAS_CLOSE szubrutin írásnál.

F593 A CAS_CHIN/OUT szubrutinok.

F596 A CAS_CHIN szubrutin.

F605 A CAS_VERIFY szubrutin.

F630 A CAS_BKIN/OUT szubrutinok.

F633 A CAS_BKIN szubrutinok.

F653 A CAS_CHOUT szubrutin.

F6AD A CAS_BKOUT szubrutin.

F730 A HL által mutatott címen kezdödö szöveg kiíratása az aktuális kurzor pozíciótól kezdödöen. A szöveg elsö byte-ja a szöveg hosszát tartalmazza.

ROM táblázatok

A ROM nemcsak különböző szubrutinok összessége, hanem különböző táblázatokat is tartalmaz, melyek nélkülözhetetlenek az egész ROM müködéséhez. Itt megadom az általam felfedezett ROM táblázatok címeit, röviden ismertetem azok funkcióit. Az RST 8H, RST 18H és RST 30H szubrutinhívások jellegzetessége, hogy a hívás után következö egy vagy több byte nem utasításként értelmezendö, hanem adatként. Ezeket a "táblázatokat" itt nem tüntetem fel, hisz ezek nem valódi táblázatok, adatterületek a mi szempontunkból nézve.

A ROM fö részének táblázatai

C003 Szorzótábla a 0*0 -tól a 9*9 -ig. Minden egyes byte BCD számként értendö. A következö módon lehet két egyjegyü szám szorzatát meghatározni a táblázat segítségével:
Pl.: 6*7 eredménye a táblázat 67-dik eleme (a 0. elemmel kezdödik a táblázat), azaz a C003H+67=C003H+0043H=C046 címen található byte.

C067 Basic utasítások ugrótáblázata. Ez a táblázat az FF-D0 tokenü utasításokat megvalósító rutinok címeit tartalmazza sorban az FF tokenüvel kezdödve.

C0C7 Az "RST 18H" aritmetikai rutinok ugrótáblája. A 0-14 sorszámú rutinok címeit tartalmazza a táblázat sorban.

C111 A ROM-ban tárolt számozott konstansok táblázata. A ROM 40 konstanst tárol, melyekre a sorszámukkal lehet hivatkozni (0-39). Minden konstans 7 byte-ot foglal el: 6 mantissza (BCD számjegyekkel) és egy karakterisztika byte. A karakterisztika legfelsö bitje határozza meg a mantissza elöjelét: 0-pozitív, 1-negatív. A 40H karakterisztika felel meg a 0 kitevönek.

C334 "MOSP" szöveg. Ennek a négy karakternek kell az oldalt bedugható ROM modul elsö négy byte-jának lenni, ha azt akarjuk, hogy RESET után automatikusan a modulban lévö program induljon.

C4AC A KERNEL rutinok ugrótáblázata. A táblázat elsö byte-ja a rutinok számát adja meg, azt követik a rutinok címei.

C545 A képernyövezérlö (6845-ös videovezérlö) regisztereinek inicializáló értékeit tartalmazó táblázat.

C555 A rendszer inicializálása során hívott rutinok táblázata. Az elsö rutin beálítja a négyszínü üzemmódot. A második rutin a billentyüzet eszközt inicializálja. A harmadik rutin egyetlen RET utasítás. A negyedik rutin pedig alapállapotba helyezi a kazetta munkaterületet.

C55D Az eszközrutinok ugrótábláinak címét tartalmazó táblázat. Rendre a video, billentyüzet, editor, hang, nyomtató, kazetta, kernel eszközök ugrótábláinak címe található itt meg.

C5B4 A 32-127 kódú karaktereket (a nem újradefiniálható karaktereket) definiáló byte-ok táblázata. Minden karaktert 10 byte definiál úgy, ahogy azt a SET CHARACTER utasításnál megismertük.

C974 A VIDEO rutinok ugrótáblázata. Az elsö byte a rutinok számát adja meg, azt követik rendre a rutinok címei.

C9EA Az induló paletta kódok bekapcsolt színkikapcsoló esetén (4 byte).

C9EE Az induló paletta kódok kikapcsolt színkikapcsoló esetén (4 byte).

CB7F A vonaltipusok képét definiáló byte-ok. Minden byte egy vonaltipust határoz meg: 1-es bit= van pont, 0-ás bit = nincs pont. A 16 byte-ból 14 különböző csak.

CBA0 A vonalkeresztezési mód ugrótáblázata. Az L_MODE rendszerváltozó értékétöl függöen adódik a vezérlés a táblázatban megadott címekre.

CBDB A BABS, BREL rutinok által vonalhúzáskor használt rutinok táblázata.

CF98 Az EDITOR rutinok ugrótáblázata. Az elsö byte a rutinok számát adja, azt követik a rutinok címei.

D007 Az editor által használt 3-szor 4 byte. A 4 byte-os csoportokon belül az elsö két byte egy rutin címe, a harmadik byte az adott grafikus felbontásban a soronkénti maximális karakterszám, a negyedik byte pedig az egy képpont színét meghatározó bitek számát adja meg. Az aktuális négy byte a RAM-ban a 0E69 címtöl kezdve is megtalálható.

D170 A Basic beépített szövegszerkeszto funkcióinak ugrótáblázata. A táblázat egy bejegyzése 3 byte. Az elsö byte a vezérlö karakter kódja, az azt követö két byte a funkciót megvalósító rutin.

D5E3 A billentyüzet rutinok ugrótáblázata. Az elsö byte a rutinok számát adja, azt követik a rutinok címei.

D7BF A billentyözet mátrix dekódoló táblázatai. Négy külön egységböl áll a táblázat a négyféle billentyüzetnek megfelelöen (alap, SHIFT, CTRL, ALT). Az altáblázatok sorfolytonosan tartalmazzák a billentyüzet mátrixot.

D8FF A printer rutinok ugrótáblázata. Az elsö byte a rutinok számát adja, azt követik a rutinok címei.

D92A A hangkezelö rutinok ugrótáblázata. Az elsö byte a rutinok számát adja, azt követik a rutinok címei.

D9BB A kazetta rutinok ugrótáblázata. Az elsö byte a rutinok számát adja, azt követik a rutinok címeit.

DA84 A "TV COMPUTER BASIC 1.2
Copyright 1985 VIDEOTON"
szöveg.

DAC3 A " bytes free" szöveg.

DAFB Az "ok" szöveg, és a "törlés a sor végéig" vezérlö karakter.

DBF6 A "VIDEOTON" szöveg.

DC00 A "TV COMPUTER" szöveg.

DC15 A hideg indításkor bejelentkező ábra paletta kódjai.

DE6D A Basic kulcsszavak táblázata csökkenö tokenek szerint rendezve. Minden kulcsszó utolsó byte-jának legmagasabb bitje 1-re van állítva.

E6BD Speciális formátumvezérlö karakterek táblázata. Ezek a karakterek a PRINT USING utasításban használatosak.

E7E4 A "SET" utasítások táblázata. A táblázat két byte-os bejegyzéseket tartalmaz. A bejegyzések elsö byte-ja a a "SET" utáni Basic kulcsszó tokenje. A második byte az utasítást megvalósító rutin címét határozza meg: a következö byte címéhez kell hozzáadni az adott byte-ot, az eredmény a rutin címe lesz.

E87B A "SOUND" utasítások ugrótáblázata. A táblázat felépítése és használata megegyezik a "SET" táblázatnál írttal.

FB5B Inicializáláskor a RAM-ba kerülö terület. A 0008h-002Eh címekre töltödö terület.

FD74 A "*** " szöveg. Ez vezeti be a Basic hibajelzéseket.

FD9C A "System error " szöveg.

FDC6 A Basic hibaüzenetei. A szöveg egy részét tokenek helyettesítik. A bejegyzések elsö byte-ja a hibaüzenet sorszáma, azt az üzenet hossza, majd az üzenet követi. A bejegyzés végét egy FFh byte jelzi.

FF47 Decimális 1000 binárisan.

FF49 Decimális 100 binárisan.

FF4B Decimális 10 binárisan.

FF4D Decimális 1 binárisan.

FFB1 A "STOP" szöveg.

FFCA Az " at line " szöveg.

A ROM bövítés táblázatai

F291 A soros vonal rutinok ugrótáblázata. Az elsö byte a rutinok számát adja, azt követik a rutinok címei.

F3C6 A soros vonal BAUD RATE-jét meghatározó két byte-os értékek.

F741 A "Reading:" szöveg.

F74E A "Searching" szöveg.

F758 A "Found:" szöveg.

FAF3 A kazetta rutinok használta byte-ok.

FB0E A rendszer inicializálásakor a 0B00-0B48 RAM területre kerülö memória tartomány.

FB54 A "MOPS" szöveg.

FB58 A "VGB" eszköz azonosító.

FB5C Az "RS232" eszköz azonosító.

FB62 A "DISK" eszköz azonosító.

FB67 A "CISL" szöveg.

FB6B A 128-160 kódú karakterek képét leíró byte-ok táblázata. Minden karakterhez 10 byte tartozik, melyek a SET GRAPHICS utasításhoz hasonlóan határozzák meg a karakter grafikus képét.

FF3F Speciális karakterek táblázata. A karaktereket követö címek a karakterek sorrendjében a megfelelö karakterhez tartozó rutin cím. Ezt a táblázatot a PRINT rutin használja.

FFDE A "(c)1985ISL" szöveg.

Kiegészítés, magyarázat a ROM rutinokhoz

Ebben a fejezetben néhány, érdeklödésre különösen számottartó rutincsoportról szólok bövebben. Ezek azok a rutinok, melyeket érdemes meghívni, beépíteni bizonyos programokba.

Az operációs rendszer rutinjainak közvetlen hívása (RST 30H rutin)

Az ismert, hogy az operációs rendszer rutinjait az RST 30H utasítással hívhatjuk meg. A rutin számát az RST 30H utasítás utáni byte-ra kell tenni. Ennél a módszernél a rutin gondoskodik a nem használt (a rutin számára bemenö paramétereket nem tartalmazó) regiszterek elmentéséröl, visszaállításáról, a rutin száma alapján a tényleges eszköz kezelö rutin címének meghatározásáról. Az operációs rendszer rutinjainak közvetlen hívásával több tucat (esetleg a 100-at is meghaladó számú) utasítás végrehajtatását is megspórolhatjuk.

Milyen módon kell az operációs rendszer rutinjait közvetlenül hívni? Majdnem ugyanúgy, mint a hagyományos módon. A rutin számára szükséges paramétereket ugyanúgy, ugyanabban a regiszterben kell megadni, mint ha az RST 30H rutin hívást használnánk. Azonban sokmindenre nekünk kell vigyáznunk, amit különben az operációs rendszer automatikusan elintézett. Lássuk ezeknek a kényes pontoknak a felsorolását:

- A nem használt regiszterek tartalmának megörzéséröl nekünk kell gondoskodni. Csak azoknak a regisztereknek a tartalma használható fel a meghívott rutinból történt visszatérés után, melyekbe a rutin ad vissza adatokat (ezek ugyanazok a regiszterek, mint az RST 30H szerinti rutinhívásnál). Az A regiszterben minden esetben a hibakódot kapjuk vissza.

- Ha a meghívandó rutin a ROM bövítésben található, nekünk kell gondoskodnunk a memória lapozásról a rutin meghívásakor, és a rutin visszatérése után. A meghívandó rutinnak mindig a P3 lapon kell kezdödni!

Ha ezeket betartjuk, nagy baj nem történhet.

Mindezek után lássunk néhány egyszerű példát a leírtak szemléltetésére:

1. példa: VID_BREL rutin közvetlen hívása

```
LD   BC,X_elmozdulás
LD   DE,Y_elmozdulás
CALL CAD7
OR   A
JP   NZ,error
```

2. példa: ED_CPOS (kurzorpozíció megjegyzése) rutin hívása

```
CALL D013
```

3. példa: CAS_CRTE (file nyitás írásra) rutin hívása

```
CALL PAGE XXXE          ; A ROM bövítés belapozása a P3
                          lapra.
LD   DE,FNAME_CéM
CALL F4CF
OR   A
JP   NZ,ERROR
...                     ; DE- ben a file név új címe
                          lesz.
```

Amennyiben nincs szükségünk egy operációs rendszer rutin minden szolgáltatására, megtehetjük, hogy nem az "elején" lépünk be egy rutinba. Minden ilyen esetben legyünk nagyon körültekintöek.

Hibakezelö rutinok
(RST 8H, RST 10H rutinok)

Már biztos feltünt, hogy kétféle hibajelzést használ a számítógép. Az egyik esetben szöveggel (angolul) jelzi a hiba miben létét. 17-féle ilyen jellegü hibajelzéssel találkozhatunk. Mivel ezek a hibajelzések csak a Basic interpreter müködése alatt jöhetnek létre, nevezhetjük ezeket a Basic hibajelzéseinek.

A ROM-ban az interpreter rutinjai az RST 8H szubrutin híváson keresztül végzik a hibakezelést. A hiba kódja az RST 8H utasítást követö byte-on található. Az RST 8H rutin csak elökészíti az A regisztert: a hiba kódja kerül az A regiszterbe. A tényleges hibakezelés az RST 10H rutinon keresztül történik.

A hibajelzések másik csoportja csak egy számmal jelzi a hiba kódját. Ezek az operációs rendszer hibajelzései.

A ROM az RST 10H utasítással végzi az operációs rendszer hibakezelését úgy, hogy a hiba kódja az A regiszterben található. Innen már közös a kétféle hibatipus kezelése. Ha az A regiszterben lévö hibakód nagyobb, mint 127, a hiba az operációs rendszer hibája, különben a Basic interpreteré.

Mi is írhatunk saját hibakezelö rutint, illetve módosíthatjuk a meglévö hibakezelést. Hogy a saját hibakezelö eljárásunkat beillesszük a meglévö hibakezelö rendszerbe, egyik módja az RST 8H vagy az RST 10H rutin "eltérítése". Tekintsünk erre egy példát:

1. példa: Azt szeretnénk, hogy szintaktikai hiba esetén ne a "Not understood." jelzés, hanem a "Nem értem." jelzés jelenjen meg. Irjuk meg a saját hibakezelö rutint:

```
Saját_Rutin: RET Z
             CP   01                    ; a minket érdeklö
                                        ; hiba kódja
             RET  NZ
             CALL FE79
             DB   0E 0D 0A
             DB   '*** Nem értem'
             JP   FDB0                  ; ugrás a ROM-ba
```

Ezután módosítsuk az RST 8H rutint:

```
000C  CALL Saját_Rutin
```

Hasonló módon az egész hibakezekést módosíthatjuk, a ROM hibakezelő rutinjai helyett a mi rutinunkat használja az interpreter, az operációs rendszer.

Természetesen a ROM-tól teljesen függetlenül is használhatunk saját hibakezelést. Pl. szeretnénk, ha lenne a BASIC-ünknek ON ERROR GOTO utasítása. Ezt viszonylag egyszerüen megvalósíthatjuk a hibakezelő rutin "eltérítésével". Ajánlatosabb talán az RST 10H rutint eltéríteni, mert így az OS hibákat is lekezelhetjük.

Az operációs rendszer hibakódjainak jelentése a gépkönyvben és a bövítö eszközök kezelési útmutatóiban megtalálhatók.

Megszakításkezelés
(RST 38H rutin)

Itt most nem kívánok a megszakítások okairól, a ROM megszakításkezelö rutinjairól szólni. Amiböl kiindulok: a processzor 1-es maszkolható megszakításkezelö módban dolgozik, ezért a megszakítást kiszolgáló rutin az RST 38H rutin. Mivel ez RAM-ban található, így módosítható. Erre mutatok egy példát itt.

1. példa: A Videoton TV Computeren nincs billentyüzet hang. Bizonyos esetekben ez hasznos dolog lehetne. Elöször írjuk meg a csipogó rutint:

```
KB_Hang:  PUSH DE
          PUSH BC
          PUSH AF
          LD   A,(0BE5)          ; KB_STATUS byte
          OR   A
          JR   Z,END             ; nincs megnyomott bil-
                                   lentyü
          LD   DE,PITCH
          LD   B,VOLUME
          LD   C,DURATION
          LD   A,FF
          LD   (0B15),A          ; a billentyüzet-hang
                                 ; megszakítja a folyó
                                 ; hangot
          RST  30H               ; TONE_SET rutin meghí-
          DB   33                ; vása
END:      POP  AF
          POP  BC
          POP  DE
          LD   (0003),A
          RET
```

Ezután térítsük el az "RST 38H" rutint:

```
          0B41  CALL KB_Hang
```

A TONE_SET rutint itt csak az RST 30H rutin segítségével hívható meg, különben nincs biztosítva a regiszterek tartalmának meg-

örzése a megszakításkiszolgálás alatt.

Hasonlóan készíthetünk egy órát, mely mindig látható a képernyön. Ehhez azt a tényt használhatjuk fel, hogy a RAM-ban található egy két byte-os interapt számláló a 0B1DH címen. Az IT számláló 20 ms-onként növekszik, ha a megszakítás éppen nincs tíltva. A megszakítás kiszolgálás tíltása idejére ez az óra természetesen megáll. A hangkezelö, kazetta, a sorosvonali rutinok szintén befolyásolják az óra müdését.

"Hibák" a ROM listában

Itt nem azokról a hibákról kívánok szólni, melyek minden igyekezetem ellenére a listában maradtak. A címben szereplö " hibák " valójában nem azok, csak bizonyos esetekben annak látszanak. Egy példával szemléltetem, hogy mire gondolok:

Nézzük figyelmesen a rendszer ROM következö részletét:

```
FD93  CALL FEDD
      .
      .
      .
FED6  CP   C
FED7  JR   Z,2
FED9  LD   B,C
FEDA  LD   C,78
FEDC  LD   C,AF
FEDE  LD   C,A
```

Elsö látásra az mondhatjuk rá, hogy rossz ez a program részlet. Rossz, mert egyrészt kétszer is olyan helyre adjuk a vezérlést, ahol nem kezdödik utasítás, egy több byte-os utasítás belsejébe ugratunk, másrész feleslegesnek tünik a C regiszterbe különbözö értékeket tölteni, ha a végén az A regisztert töltjük a C-be.

Azonban szó sincs semmiféle hibáról, csak helytakarékos megoldásról. A következö rutinok sürítménye az elöbbi ROM részlet:

Ha a CP C összehasonlítás eredményeként a két regiszter tartalma nem egyenlö:

```
FED9  LD   B,C     ; a B regiszterbe C tartalma,
FEDA  LD   C,78    ; a C regiszterbe pedig az A
FEDC  LD   C,AF    ; regiszter tartalma töltödik
FEDE  LD   C,A
```

Ha az összehasonlítás eredményeként a két regiszter tartalma egyenlö:

```
FEDB  LD   A,B     ; mind az A, mind a C regisz-
FEDC  LD   C,AF    ; terbe a B regiszter tartalma
FEDE  LD   C,A     ; kerül
```

Az FEDD címen kezdödö rutin:

```
FEDD  XOR  A
FEDE  LD   C,A      ; C-be és A-ba nulla kerül.
```

Nem az egyetlen ilyen "hiba" a bemutatott. Böven található hasonló megoldás a ROM-ban. Ilyen megoldású az LPRINT és a PRINT rutinok közös kezdete, a "SET" és "SOUND" utasításokat megvalósító rutinok. A ROM ezen "hibás" részeinél legyünk körültekintöek, ha egyes rutinok müködését próbáljuk nyomon követni. Segítségül a listában (***) karaktersorral jelöltem meg az ilyen "hibás reszek kezdetét és végét.

A Basic utasítások és függvények közvetlen hívása

Elöljáróban megjegyzem, hogy a Basic utasításait általában nem nagyon érdemes közvetlenül hívni. Ennek legfőbb oka, hogy az utasításokat megvalósító rutin nem szubrutin, nem valamiféle RET utasítással adja vissza a vezérlést, a Basic értelmező nem is CALL utasítással hívja meg. Amennyiben mégis úgy döntünk, hogy közetlenül akarjuk valamelyik Basic utasítást megvalósító rutint meghívni, a következőket vegyük figyelembe:

- az IX regiszter a Basic változó területére (munka területére) mutat.
- az IY regiszter a Basic Stack aljára mutat, a Basic Stack Pointer. (Megkülönböztetendő az SP regisztertől, mely a processzor Stack Pointer-e most is, és mindig, minden körülmények között!)
- a HL' regiszter a forrásprogram (Basic program, Basic parancs) következő elemére mutat.
- a B' regiszter a forrásprogram aktuális elemével kapcsolatos értéket tartalmaz (sokszor ez éppen a forrásprogram aktuális eleme).

A többi regiszter szabadon használható.

A Basic függvények közvetlen hívásával is van némi probléma. Ha az "elejéről" hívjuk meg valamelyik Basic függvényt, a függvény változóját az interpreter megfelelő rutinjai készítik elő a függvény számára. (Numerikus függvények esetén ez a rutin az RST 18H rutinok közül a 0EH sorszámú, a EA68H címen kezdődő, string függvények esetén pedig az EA75H címen kezdődő.)

Ajánlatosabb "később" belépni a függvényeket megvalósító rutinokba, ott, ahol a függvény már a Basic Stack-be várja a paramétereket. Az eredmények szintén a Basic Stack-ben keletkeznek.

Példaként vegyük a következőket:

1. példa: egy a memóriában található Basic programot akarunk elindítani.

```
LD   HL,PROGR_KEZDET  ; a Basic program címe a
```

```
LD    (1722),HL      ; TEXT változóba kerül
XOR   A
JP    DE1B           ; ugrás a RUN rutinra
```

E program hatására a PROGR_KEZDET címen kezdödö Basic program elindul. Ezzel a módszerrel a memóriában egyszerre több programot tárolva, bármelyiket elindíthatjuk.

2. példa: a Basic Stack-be helyezett számjegyekből álló stringnek vegyük a cosinusa abszolutértékének logaritmusát:

```
CALL F040      ; a Basic Stack-be helyezett
               ; stringből szám lesz a
               ; Basic Stack-ben.

CALL EBA7      ; a Basic Stack-be az ott talál-
               ; ható szám cosinusa kerül

CALL EAEC      ; a Basic Stack-be az ott talál-
               ; ható szám abszolutértéke kerül

CALL ED21      ; a Basic Stack-be a kívánt ered-
               ; mény kerül
```

Ami a számítógéphez adott programozási segédletből kimaradt

A számítógéphez két vékony könyvecskét is mellékelnek minden vevö számára. Ezek nagy segítséget nyújtanak a gép Basicjének elsajátításához. Ahogy az már lenni szokott, néhány dolog kimaradt belölük.

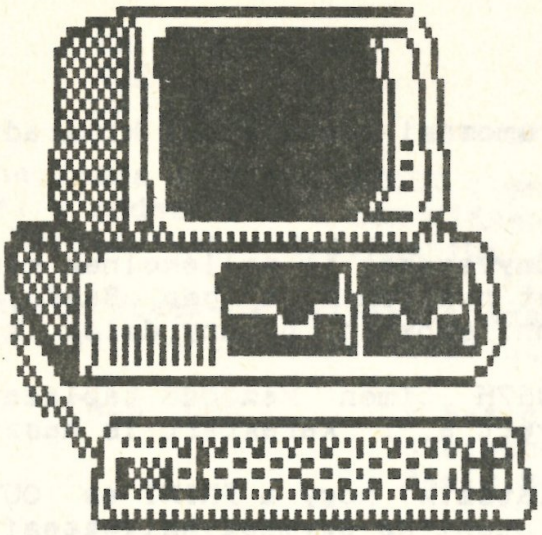
Ha figyelmesen megnézzük a C067H címen kezdödö táblázatot, kitünik, hogy a REM utasítás helyett a '!' karaktert is használhatjuk.

Szintén ebböl a táblázatból látszik, hogy a PRINT és OUTPUT rutinok is azonosak, tehát a két utasítás egymást helyettesítheti.

A DED4H - DED5H byte-okon egy ismeretlen Basic kulcsszót találhatunk. Ez az OK. Ha az utasítást megvalósító rutint megvizsgáljuk, azt találhatjuk, hogy ez egy nagyon érdekes utasítás. Helyezzük el egy programba ezt az utasítást. Azt tapasztaljuk, hogy amikor a program futása során az OK utasításhoz ér, a Basic visszatér parancs módba. Ekkor tetszöleges Basic parancsot vagy kettöspontokkal elválasztott parancssorozatot begépelhetünk. Ezt végrehajtva, a megszakított programot folytatja az interpreter. Tehát az OK utasítás egy különleges input lehetöséget teremt meg számunkra: nem számokat vagy stringeket vár a gép az OK utasítás végrehajtásakor, hanem parancsot, parancssorozatot vagy akár programsort (!).

A Basic beépített függvényei között is akad ismeretlen, az az IO függvény. A függvény változója egy byte-os szám (0-255) lehet. Ha x a függvény változója, az eredmény az 1941H+2*x címen található két byte-os egész lesz.

Végül egy ROM hibára szeretném felfívni a figyelmet. Rossz a XOR-t megvalósító rutin. Pontosabban szólva csak félig jó. Amíg egy byte-os számok között végzünk XOR müveletet, semmi baj nincs. Azonban két byte-os számok között már nem ad jó eredmény: a számok felsö byte-jaival csak OR müvelet fog végrehajtódni. Az XOR müvelet alkalmazásakor vegyük ezt figyelembe.



A SZÁMITÁSTECHNIKÁÉRT,
AZ OKTATÁSÉRT,
ÉS
A SZÁMITÁSTECHNIKA
OKTATÁSÁÉRT

MA már több, mint 12.000 VIDEOTON TV-Computer működik hazánkban, amelyek jelentős része - mintegy 8000 db - Intézetünk segítségével került az oktatási intézményekbe.

Az alapgépen kívül perifériákkal és egyéb kiegészítőkkel állunk az érdeklődők rendelkezésére.

A VIDEOTON TV-Computeren futó **OKTATÓPROGRAMJAINK** több tárgy egyéni és csoportos elsajátítását segítik, választékuk illetve mennyiségük folyamatosan bővül.

Érdeklődésüket
levélben és telefonon egyaránt
várják munkatársaink!

TUDOMÁNYSZERVEZÉSI
ÉS
INFORMATIKAI INTÉZET

Budapest, Pf.454. 1372
Telefon: 665-011/2663;2664
813-197

260,– Ft

A VIDEOTON TV-COMPUTER ROM LISTÁJA

Kaszanyiczki László