

ASMZ80 assembler DDZT debugger



VIDEOTON

TV-Computer

209.195.02.01.

ASMZ80 ASSEMBLER ÉS DDZT DEBUGGER

TV-COMPUTERRE

FELHASZNALÓI KÉZIKÖNYV

TARTALOMJEGYZÉK

1. ASMZ80	5
1.1. Bevezetés	5
1.2. Az ASMZ80 használata	5
1.3. Az ASMZ80 file-jai	7
1.4. A forrásfile szintaktikája	8
1.5. Direktívák	11
1.5.1. ORG	12
1.5.2. EQU	12
1.5.3. SET	12
1.5.4. TITLE	12
1.5.5. PAGE	13
1.5.6. DEFB	13
1.5.7. DEFW	13
1.5.8. DEFM	13
1.5.9. DEFS	14
1.5.10. IF, ENDIF	14
1.5.11. END	14
1.6. Utasítások	15
1.6.1. Adatmozgató utasítások	15
1.6.2. Aritmetikai és logikai utasítások	16
1.6.3. Inkrementálási és dekrementálási utasítások	18
1.6.4. Összehasonlító utasítások	19
1.6.5. Ugró utasítások	20
1.6.6. Bit műveletek	20
1.6.7. Léptető utasítások	21
1.6.8. CPU vezérlő utasítások	24
1.6.9. Szubrutin hívási és visszatérési utasítások	24
1.6.10. Input, output utasítások	25
1.7. Hibaüzenetek	26

2. DDZT	28
2.1. Bevezetés	28
2.2. A DDZT használata	28
2.3. Parancsok	30
2.3.1. D parancs	30
2.3.2. F parancs	31
2.3.3. G parancs	31
2.3.4. I parancs	32
2.3.5. M parancs	32
2.3.6. R parancs	33
2.3.7. S parancs	33
2.3.8. T parancs	34
2.3.9. U parancs	34
2.3.10. X parancs	34
2.4. Hibaüzenetek	35
3. Függelék	36
3.1. A CPU regiszterkészlete	36
3.2. Utasítások műveleti kód szerint	38

1. ASMZ80

1.1. Bevezetés

Az ASMZ80 az UPM operációs rendszer alatt futó assembler program, mely alkalmas a Z80 processzorra írt assembly szintű programok fordítására. A program minden CP/M kompatibilis operációs rendszer alatt használható. Az ASM80 használatához szükséges az UPM operációs rendszer ismerete.

1.2. Az ASMZ80 használata

Az assembler az assembly nyelvű forrásfile-ból készít Z80 gépi kódú hexadecimális formátumú, és listázható formátumú file-okat. A program hívása:

```
ASMZ80 x:file-név $paraméterek
```

A file-név a fordítandó file neve és kiterjesztése. Ha a file kiterjesztése nincs megadva, a fordító ASM kiterjesztést tételez vel.

Az x az a lemezegység, melyen a forrásfile található és amelyre a generált file-ok kerülnek. Az itt kijelölt lemezegység a paraméterekkel módosítható.

A paraméterekkel a generált file-ok és a forrásfile helye definiálható. Minden file helye két karakterrel adható meg. Az első a file nevére, a második a helyére utal.

A lehetséges 1. karakterek:

A: forrásfile (ASM kiterjesztésű)

H: hexa file (HEX)

P: lista file (PRN)

E: hibalista file (ERR)

A 2. karakterek lehetséges értékei:

- A... P: az input vagy output lemezegység neve
- X: az output a konzolra megy
- Y: az output a lista eszköz (pl. sornyomtató)
- Z: a fordító nem generálja ezen file-t

A paraméterek a \$ jel után helyköz nélkül következnek egymás után a sor lezárásáig. A paraméterek részben vagy egészben elhagyhatók. Az A, H és P elhagyásakor a fordító az aktuálisan kiválasztott lemezegységet tételezi fel. Az E paraméter elhagyása esetén a hibalista a konzolra kerül.

A program hívása után assembler a következő üzenetet küldi a konzolra:

```
— ASMZ80 UPM ASSEMBLER V 1.2 FOR THE TV-COMPUTER —
```

A fordítás befejezésekor pedig:

```
NEXT PC = xxxx  
END OF ASSEMBLY
```

ahol xxxx a fordított program utolsó tárgykódja utáni memóriacím.

A "FOR THE TV-COMPUTER" (a TV-Computer számára) csak annyit jelent, hogy az assembler az ékezetes kis és nagybetűk kódjait a TV-Computer kódkészletének megfelelően kezeli. Amennyiben az ékezetes betűket nem használjuk, a program használható VT16-on is.

1.3. Az ASMZ80 file-jai

Az ASMZ80 forrásfile-ja a Z80 utasításait, operandusait és direktívákat tartalmazhat. Előállítására például az ED forrásszerkesztő használható.

Egy példa:

```
TITLE          PRÓBA
OR             100H
PRINT EQU      9
HIV EQU        5
BOOT EQU       0
LD             DE, TEXT
LD             C,PRINT
CALL          HIV
JP            BOOT
TEXT          DEFB 'PRÓBA ASMZ80–HEZ $'
END
```

A forrásfile-ból a fordító generálja a HEX, PRN és ERR kiterjesztésű file-okat. A HEX kiterjesztésű file a tárgykódú programot tartalmazza az alábbi formában:

```
: 10020000110B020E09CD0500C3000050524F4241B0
: 0D0210002041534D38305A2D48455A2024C6
: 0000000000
```

Egy sor felépítése a következő:

byte:	Tartalma:
1.	kettőspont
2–3.	a 10. helytől az n. helyig lévő byte-ok száma
4–7.	a sor hexadecimális értékeinek kezdőcíme a memóriában
8–9.	00
10-n.	az ábrázolt byte-ok
n+1, n+2	a 2. . .n. pozíciókban levő értékek kontroll összege. Ennek és a 2. . .n-en levő értékeknek az összege 0.

A PRN file a lefordított programot tartalmazza, listázható formátumban. A PRN file-ba bejelölésre kerülnek a hibák közvetlenül a generált gépi kódú érték: elé, egybetűs hibakóddal. Az előző példaprogram PRN file-ja:

```

0200                                TITLE      PRÓBA
0200                                ORG        200H
0009 =          PRINT              EQU      9
0005 =          HIV                 EQU      5
0000 =          BOOT                EQU      0
0200 110B02      LD                 DE, TEXT
0203 0E09        LD                 C,PRINT
0205 CD0500      CALL              HIV
020B C30000      JP                 BOOT
020B 50524F4241TEXT DEFB          'PRÓBA ASMZ80–HEZ $'
                                END

```

Az ERR file a program szintaktikai hibáinak listája.

1.4. A forrásfile szintaktikája

Az assembly nyelvű forrásprogramok sorok egymásutáni sorozatából állnak a felhasználó által megadott sorrendben. Minden sor maximum 4 mezőből állhat. A mezőket elsősorban sorrendjük azonosítja, valamint a határoló karakterek. Egy sor a következő mezőkből állhat:

címke műveleti kód operandus ; megjegyzés

A sorból egy vagy több elem elhagyható. A sor lezárására a CR (ctrl J) vagy a ! használható. A sor elemeit legalább egy elválasztó karakter választja el egymástól, ami lehet space, tabulátor, ill. címke után a : és megjegyzés előtt a pontosvessző (;).

A címke egy maximum 16 karakteres azonosító karaktersorozat, ami után megengedett a : használata. Pl.:

CIMKE: ugyanaz mint CIMKE

A forrásfile-ban egy azonosító egyszer fordulhat elő címkeként (SET direktívák azonosítónevei kivételével). A cím első karaktere lehet egy A–Z közötti ASCII karakter, a további karakterek pedig betűk (A–Z) (magyar ékezetes karaktereket is beleértve) vagy számok (0–9) lehetnek. A kisbetűk használata megengedett, de a fordító nagybetűkké konvertálja őket. A címke önállóan is alkothat egy sort és bármelyik sorból hiányozhat is.

Az utasítás a Z80 processzor gépi utasítása vagy direktíva lehet. A gépi utasításokat jellemző műveleti kódokkal reprezentáljuk. A direktívák részletes leírását ld. az 1.5. fejezetben. Az utasítások összefoglalása az 1.6. fejezetben található.

Az operandus mező az utasítás operandusát vagy operandusait tartalmazhatja. Az assembler a műveleti kód és az operandus mező együttes felhasználásával képezi a kijelölt utasítás gépi kódját. A műveleti kód mező az utasítások műveleti típusát, az operandus mező pedig a művelethez szükséges operandus értékét jelöli ki. Az operandus felépíthető konstansok és azonosítók kombinációjából, logikai kifejezésekből, karaktersorozatokból.

Az operandusok lehetséges fajtái:

Azonosító:

A címkék első 5 karakterét a fordító azonosító névnek tekinti, és egy 16 bites értéket rendel hozzájuk. Minden rá való hivatkozáskor a címke aktuális értéke bekerül a fordításba. Egy speciális azonosító a \$ karakter, aminek értéke mindig az aktuális program számláló értéke (PC). További fenntartott azonosítóknak a fordító ad értéket, ezek máshol azonosítóként nem használhatók:

A, B, C, BC, D, E, DE, H, L, HL, IX, IY, SP, PSW, AF, Z, NZ, NC, P, M, PE, PO

Természetesen nem használhatók azonosítónak az utasítás és direktíva nevek sem.

Numerikus konstans:

A konstansok lehetnek bináris, oktális, decimális vagy hexadecimális számrendszerbeli számok. A szám alapszámrendszerét a következő betűk jelölik:

B: bináris
O: oktális
D: decimális
H: hexadecimális

Ha a számrendszer típusát nem adjuk meg, a fordító tizes számrendszert tételez fel. A konstansoknak mindig számjegy karakterrel kell kezdődniük, — hexadecimális számoknál esetenként bevezető nullával — a számrendszer alapszámára utaló karakter a számjegy után következik.

Karaktersorozat:

Az ' (apostrof) jelek közé zárt karaktersorozatot ASCII kódú értékeként kezeli a fordító. A ' (27H) jel megadása egy karaktersorozatban két " jel megadásával lehetséges.

Aritmetikai és logikai operátorok:

Segítségükkel az azonosítók és konstansok kifejezésekké kapcsolhatók össze. A kifejezés értékét a fordító számolja ki a műveletek egy adott prioritási sorrend szerinti elvégzésével.

Az operátorok prioritási sorrendben:

()	zárójelezés
* / MOD SHL SHR	szorzás, osztás, modulo osztás, logikai léptetés balra, logikai léptetés jobbra
+ -	összeadás vagy pozitív előjel, kivonás vagy negatív előjel
NOT	logikai negálás
AND	logikai ÉS
OR XOR	logikai VAGY, logikai kizáró VAGY

Az egy sorban szereplő operátorok azonos prioritásúak, a leírás sorrendjében hajtódnak végre, egyébként az előbbi sorban lévő művelet eredményét számítja ki először a fordító. A betűkből álló operátorokat pont (.) vagy helyköz karakter választja el az operandustól. Ha egy kifejezés teljes egészében zárójelben van, akkor az memóriacímre való hivatkozást jelent.

Megjegyzés:

A megjegyzés mező a programhoz fűzött megjegyzéseket tartalmazza. A megjegyzés egy pontosvesszőt követő karakterek sorozata egy soron belül. A megjegyzések bármely oszlopban kezdődhetnek, így egy sor csak megjegyzésből is állhat.

1.5. Direktívák

A direktívák a fordítóprogramnak szóló utasítások. Tárgykódot az adatgeneráló utasítások kivételével nem képeznek, a gépi kód generálását vezérlik. A megengedett direktívák:

```
ORG
EQU
SET
TITLE
PAGE
DEFB
DEFW
DEFM
DEFS
IF
ENDIF
END
```

1.5.1. ORG

Formája:

név ORG kifejezés

Megadja a következő tárgykód helyét. Az ORG utáni gépi kódok a kifejezés értékétől fognak elhelyezkedni a memóriában. A név megadása opcionális.

1.5.2. EQU

Formája:

név EQU kifejezés

A címke mezőben megadott azonosító névhez az operandusban levő kifejezés értékét rendeli. Az azonosító nevet nem szabad újra definiálni.

1.5.3. SET

Formája:

név SET kifejezés

Az azonosító névhez az operandusban levő kifejezés értékét rendeli. Egy programon belül a SET direktívával az azonosító értéke tetszőlegesen sokszor megváltoztatható.

1.5.4. TITLE

Formája:

TITLE string

A megadott karaktersorozat a fordítási lista minden lapjának fejlécére kiíródik.

1.5.5. PAGE

Formája:

PAGE kifejezés

A kifejezés által megadott számú sor kerül egy lapra a fordítási listában.

1.5.6. DEFB

Formája:

név DEFB kifejezés, kifejezés, . . .

Adat byte-ok elhelyezése a memóriába. A kifejezés értéke az aktuális memória byte-ba töltődik. Több egymástól vesszővel elválasztott operandus szekvenciálisan egymás után töltődik a memóriába. A kifejezés értékének 1 byte-on ábrázolhatónak kell lennie. A név megadása opcionális.

1.5.7. DEFW

Formája:

név DEFW kifejezés, kifejezés, . . .

Adatszó elhelyezés a memóriába. A kifejezés alacsonyabb helyiértékű byte-ja az aktuális memória címre, a magasabb helyiértékű byte-ja az utána következő címre kerül. Több operandus megadása esetén az értékek egymás után szekvenciálisan töltődnek a memóriába. A név megadása opcionális.

1.5.8. DEFM

Formája:

név DEFM 'string', 'string', . . .

Az operandus aposztrófok közé zárt ASCII karaktersorozat. A karakterek 8 bites ASCII kódjai töltődnek a memóriába. Magát az aposztróft két aposztróffal jelölhetjük a stringben. A név megadása opcionális.

1.5.9. DEFS

Formája:

név	DEFS	kifejezés
-----	------	-----------

A kifejezésben megadott számú byte-ot lefoglalja az aktuális memória címtől kezdődően. A következő generált kódokat az assembler a DEFS által lefoglalt terület után fogja elhelyezni. A címkemező azonosító neve a lefoglalt memória mező kezdőcímére mutat. A név megadása opcionális.

1.5.10. IF, ENDIF

Formája:

név	IF	kifejezés
	.	
	.	; utasítások
	.	
	.	
	ENDIF	

Feltételes fordítást tesz lehetővé. Ha az IF operandusában levő kifejezés értéke nem nulla, a fordítás folytatódik. Ha a kifejezés értéke nulla, az IF és ENDIF közötti utasításokat figyelmen kívül hagyja a fordító, nem képez belőlük tárgykódot. A név megadása opcionális.

1.5.11. END

Formája:

END

A forrásprogram fizikai végét jelzi. A fordító END után fejezi be a fordítást.

1.6. Utasítások

Ez a fejezet az utasítások rövid leírását tartalmazza. Az operandusok csak jelezve vannak (x,y,f,b), az utasításokhoz a lehetséges operandusok más-más kombinációja rendelhető. Ezek konkrét értékeit lásd a 3.2. fejezetben.

A jelzők beállításánál a jelölések az alábbiak:

- a jelzőbit nem változik
- 1 a bit 1-be íródik
- 0 a bit 0-ba íródik
- I a művelet eredményétől függően változik
- x értéke ismeretlen
- P P/V a paritást jelzi
- * értéke azonos az IT IFF2 flip-flopjának értékével
- 2 P/V=0, ha BC-1=0, különben P/V=1
- 3 Z=1, ha A=(HL), különben Z=0
- 4 Z=1, ha B-1=0, különben Z=0

A flag byte felépítése a 3.1. fejezetben található.

1.6.1. Adatmozgató utasítások

	P C Z – S N H V	
LD x,y	– – – – –	Adatmozgatás y helyről x helyre. A jelzőbitek nem változnak, kivéve a következő két LD utasításnál:
LD A,I	– I * I 0 0	
LD A,R	– I * I 0 0	
PUSH x	– – – – –	Az x tartalma elmentődik a stack memóriába.

POP x	-----	x-be az SP által kijelölt memóriaszó értéke töltődik.
EX x,y	-----	x és y tartalma felcserélődik.
EXX	-----	A BC, DE és HL regiszterpárok tartalma felcserélődik BC', DE', és HL' regiszterpárok tartalmával.
LDI	--2-00	a HL regiszterpárral megcímzett memóriabyte tartalmát a DE regiszter párral megcímzett memóriabyte-ba tölti. Átmásolás után: HL=HL+1, DE=DE+1, BC=BC-1
LDIR	--0-00	LDI utasítás műveletsorozatát ismétli addig, amíg a BC regiszterpár tartalma 0-vá nem válik.
LDD	--2-00	A HL regiszterpárral megcímzett memóriabyte tartalmát a DE regiszterpárral megcímzett memóriabyte-ba tölti. Átmásolás után: HL=HL-1, DE=DE-1, BC=BC-1
LDDR	--0-00	LDD utasítás műveletsorozatát ismétli addig, amíg BC regiszterpár tartalma 0-vá nem válik.

1.6.2. Aritmetika és logikai utasítások

	P CZ - SNH V	
ADD A,x	----- IIVIOI	Az x hozzáadódik az A regiszter tartalmához, az eredmény az A-ban keletkezik.

ADD x,y	1 0 0 0 x	Az y hozzáadódik az x tartalmához, az eredmény az x-ben keletkezik (x lehet HL, IX vagy IY).
ADC A,x	1 1 V 1 0 1	Az x Carry-vel hozzáadódik az A tartalmához, az eredmény az A-ban keletkezik.
ADC HL,x	1 1 V 1 0 x	Az x a Carry-vel hozzáadódik a HL reg. pár tartalmához, az eredmény a HL-ben keletkezik.
SUB x	1 1 V 1 1 1	Az X kivonódik az A-ból, az eredmény az A-ban keletkezik.
SBC A,x	1 1 V 1 1 1	Az x operandus Carry-vel kivonódik az A regiszterből, eredmény az A-ban keletkezik.
SBC HL,x	1 1 V 1 1 x	Az x operandus Carry-vel kivonódik a HL regiszterpárból, eredmény a HL-ben keletkezik.
AND x	0 1 P 1 1 1	Logikai ÉS művelet A reg. és x között. Az eredmény az A-ban keletkezik.
OR x	0 1 P 1 0 0	Logikai VAGY művelet az A reg. és x között. Az eredmény az A-ban keletkezik.
XOR x	0 1 P 1 0 0	Logikai kizáró VAGY x és A reg. között. Az eredmény az A-ban keletkezik.
DAA	1 1 P 1 - 1	Az akkumulátor decimális rendező utasítása. Az A reg. tartalmát két BCD számjeggyé rendezi decimális számokkal való összeadás vagy kivonás után (ADD, ADC, INC, SUB, SBC, DEC, NEG).

CPL	---- 11	Akkumulátor tartalmának bitenkénti negálása (egyes komplement).
NEG	IVIII	Akkumulátor tartalmának 0-ból való kivonása (kettes komplement).
CCF	I--- 0x	Carry jelzőbit negálás.
SCF	1--- 00	Carry 1-be íródik.

1.6.3. Inkrementálási és dekrementálási utasítások

	P CZ-SNH V -----	
INC x	-IVIOI	x eggyel inkrementálódik ($x=x+1$) x lehet B, C, D, E, H, L, A, (HL), (IX+d) vagy (IY+d).
INC y	-----	y operandus BC, DE, HL, SP, IX vagy IY lehet
DEC x	-IVIII	x eggyel dekrementálódik ($x=x-1$).Az x operandus értékei ugyanazok lehetnek mint INC x utasításnál.
DEC y	-----	y operandus értékei ugyanazok lehetnek mint INC y utasításnál.

1.6.4. Összehasonlító utasítások

$$\begin{array}{c} P \\ CZ - SNH \\ V \\ \hline \end{array}$$

CP x

11V111

Az A reg. és x tartalmának összehasonlítása. Az összehasonlítás belső kivonás alapján történik, az A tartalmából kivonódik a kijelölt byte és a jelzők az eredménytől függően állnak be. Az A értéke nem változik. Ha a két byte megegyezik, a Z jelzőbit értéke 1.

CPI

- 3 2 1 1 1

A reg. tartalmát HL regiszterpárral megcímzett memóriabyte tartalmával összehasonlítja. Összehasonlítás után: HL=HL+1, BC=BC-1

CPIR

- 3 2 1 1 1

CPI műveletsorozatát ismétli addig, míg Z=1 vagy BC=0 nem lesz.

CPD

- 3 2 1 1 1

A reg. tartalmát HL regiszterpárral megcímzett memóriabyte tartalmával összehasonlítja. Összehasonlítás után: HL=HL-1, vagy BC=0 nem lesz.

CPDR

- 3 2 1 1 1

CPD műveletsorozatát ismétli addig míg Z=1, vagy BC=0 nem lesz.

1.6.5. Ugró utasítások

	P C Z – S N H V	
JP x	-----	Vezérlésátadás x címre (PC=x).
JP f,x	-----	Vezérlésátadás, ha f flag bit értéke 1.
JR x	-----	Relatív ugrási utasítás. Vezérlésátadás x címre.
JR f,x	-----	Relatív vezérlésátadás, ha flag bit értéke 1.
DJNZ x	-----	Mindaddig vezérlésátadás x-re, míg B reg. tartalma nem egyenlő nullával.

1.6.6. Bit műveletek

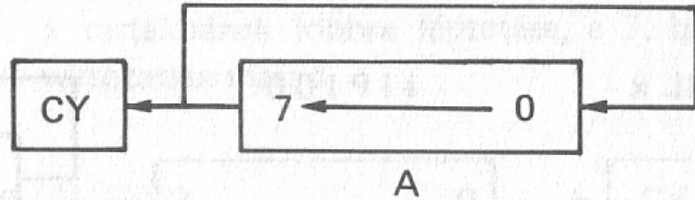
	P C Z – S N H V	
BIT b,x	----- - x x 0 1	x-nek a b-edik bitjét teszteli. A Z flag bit a vizsgált bit negáltját fogja tartalmazni.
SET b,x	-----	x b-edik bitjét 1-be állítja.
RES b,x	-----	x b-edik bitjét 0-ba állítja.

1.6.7. Léptető utasítások

P
C Z - S N H
V

RLCA

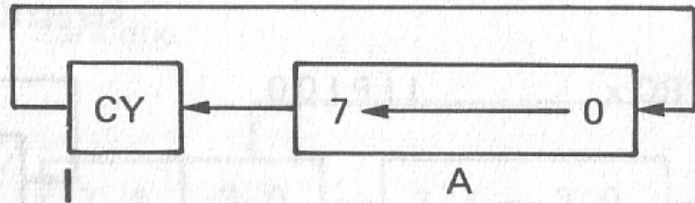
I --- 00



A reg. tartalmának cirkuláris balra léptetése.

RLA

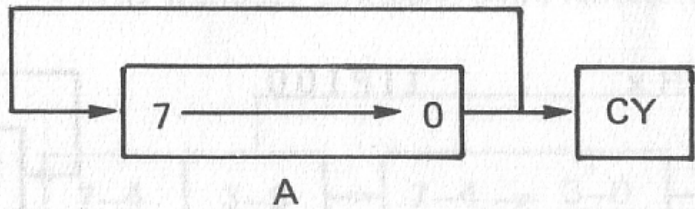
I --- 00



A reg. tartalmának balra forgatása a Carry-n keresztül

RRCA

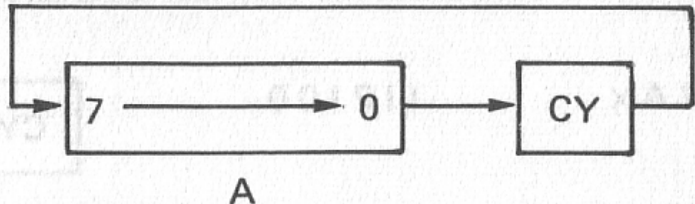
I --- 00



A reg. tartalmának cikruláris jobbra léptetése.

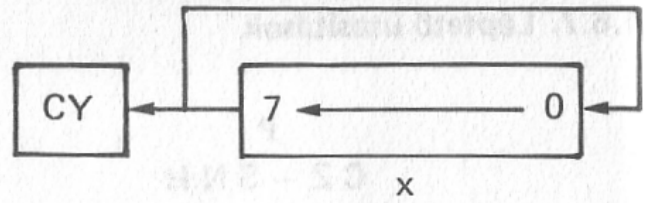
RRA

I --- 00



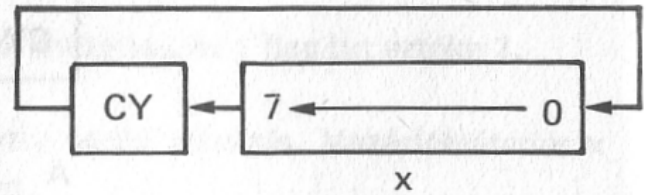
A reg. tartalmának jobbra forgatása a Carry-n keresztül.

RLC x IIP100



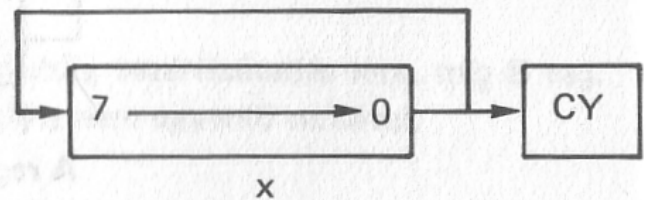
x tartalmának cikulásis balra léptetése.

RL x IIP100



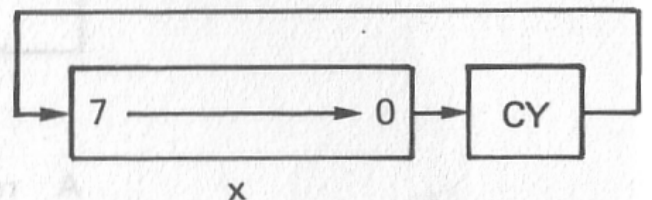
x tartalmának balra forgatása a Carry-n keresztül.

RRC x IIP100



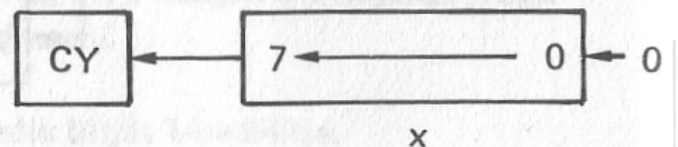
x tartalmának cikulásis jobbra léptetése.

RR x LIPI00



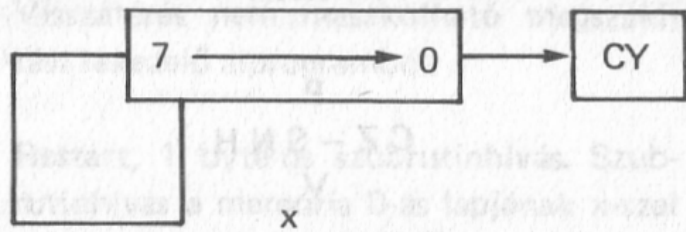
x tartalmának jobbra forgatása a Carry-n keresztül.

SLA x IIP100



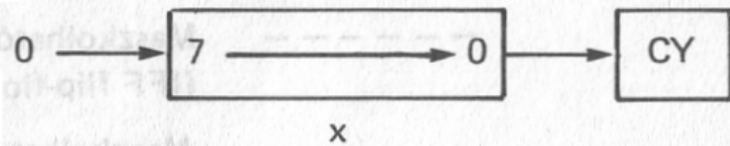
x tartalmának balra léptetése, a 0. bit törlődik.

SRA x IIP100



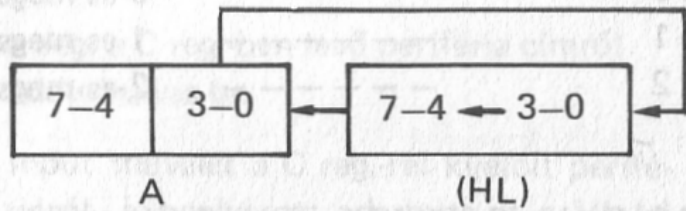
x tartalmának jobbra léptetése, a 7. bit változatlan marad.

SRL x IIP100



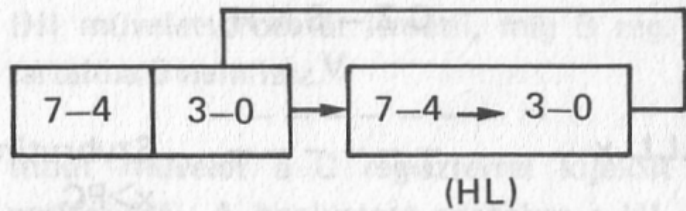
x tartalmának jobbra léptetése, a 7. bit törlődik.

RLD -IP100



3 BCD számjegy cirkuláris balra léptetése.

RRD -IP100



3 BCD számjegy cikruláris jobbra léptetése.

1.6.8. CPU vezérlő utasítások

	P	CZ	S	N	H	V	
NOP	---	---	---	---	---	---	A CPU nem hajt végre műveletet
HALT	---	---	---	---	---	---	A CPU működésének felfüggesztése
DI	---	---	---	---	---	---	Maszkolható megszakítások érvénytelenítése (IFF flip-flop-ok törlése)
EI	---	---	---	---	---	---	Maszkolható megszakítások érvényesítése (IFF flip-flop-ok 1-be írása).
IM 0	---	---	---	---	---	---	0-ás megszakítási üzemmód
IM 1	---	---	---	---	---	---	1-es megszakítási üzemmód
IM 2	---	---	---	---	---	---	2-es megszakítási üzemmód

1.6.9. Szubrutin hívási és visszatérési utasítások

	P	CZ	S	N	H	V	
CALL x	---	---	---	---	---	---	Szubrutin hívás x címről $PC \rightarrow (SP)$, $x \rightarrow PC$.
CALL f,x	---	---	---	---	---	---	Szubrutin hívás, ha f feltétel igaz.
RET	---	---	---	---	---	---	Visszatérés szubrutinból $(SP) \rightarrow PC$.
RET f	---	---	---	---	---	---	Visszatérés szubrutinból, ha f feltétel igaz.
RETI	---	---	---	---	---	---	Visszatérés megszakítást lekezelő alprogramból.

RETN -----

Visszatérés nem maszkolható megszakítást lekezelő alprogramból.

RST x -----

Restart, 1 byte-os szubrutinhívás. Szubrutinhívás a memória 0-ás lapjának x-szel kijelölt címéről.

1.6.10. Input, output utasítások

P
CZ-SNH
V

IN A,x -----

x címről adatbeolvasás az A reg.-be.

IN x, (C) - I P I O I

x reg. a C reg.-ben levő periféria címről adatot olvas be.

INI - 4 x x 1 x

Input művelet a C reg.-rel kijelölt perifériáról, a beolvasott adatbyte-ot a HL-lel megcímezett memória címre tölti, utána:
 $HL=HL+1, B=B-1$

INIR - 1 x x 1 x

INI műveletsorozatát ismétli, míg B reg. tartalma 0 nem lesz.

IND - 4 x x 1 x

Input művelet a C regiszterrel kijelölt perifériáról. A beolvasott adatokat a HL regiszterpárral megcímezett memóriabyte-ba tölti. Utána:
 $HL=HL-1, B=B-1$

INDR - 1 x x 1 x

Az IND műveletsorozatát ismétli mindaddig, amíg B regiszter tartalma nullává nem válik.

OUT x,y -----

Adatkiküldés y-ból x címre.

OUTI	- 4 x x 1 x	Adatkiküldés C regiszterrel kijelölt perifériára. A kiküldött adatot a HL regiszterpárral megcímzett memóriabyte-ból veszi. Utána: HL=HL+1, B=B-1
OTIR	- 1 x x 1 x	OUTI műveletsorozatát ismétli addig, míg B reg. tartalma nullává nem válik.
OUTD	- 4 x x 1 x	Adatkiküldés C regiszterrel kijelölt perifériára. A kiküldött adatot a HL regiszterpárral megcímzett memóriabyte-ból veszi. Utána: HL=HL-1, B=B-1
OTDR	- 1 x x 1 x	Az OUTD műveletsorozatát ismétli addig, míg B reg. nullává nem válik.

1.7. Hibaüzenetek

Ha az assembly nyelvű programban szintaktikai hiba van, azok egy karakterrel lesznek jelezve a PRN file-ban. A hibajelzések:

D	Hibás adat vagy operandus
E	Hibás aritmetikai vagy logikai kifejezés
L	Címkehiba
O	Túl nagy adat megadása, túlcsondulás
P	Fázishiba, pl. egy címke kétszer van definiálva
R	Regiszterhiba
U	Nem definiált címke

A hibajelzések másik fajtája a fordító hívásakor és a file-ok megadásakor elkövetett hiba esetén lép fel:

NO SOURCE FILE PRESENT

Nem létezik a megadott forrás file a kijelölt lemezegegyesén.

DISK FULL

A generált file-ok nem férnek fel a kijelölt lemezre.

SOURCE FILE READ ERROR

Nem tudja olvasni a forrás file-t a fordító

OUTPUT FILE ERROR

A fordító nem tudja felírni vagy megnyitni az output file-t.

PARAMETER ERROR

Paraméterezési hiba.

2. DDZT

2.1. Bevezetés

A DDZT lehetővé teszi a TV-Computer programjainak dinamikus interaktív tesztelését és nyomkövetését. A program az UPM operációs rendszer alatt fut. A DDZT használatához szükséges az UPM operációs rendszer felhasználói kézikönyv ismerete.

2.2. A DDZT használata

A program betöltése a következő parancsok valamelyikével lehetséges:

DDZT

DDZT x:file-név

ahol file-név a tesztelendő program neve, x pedig azt a lemezegységet jelenti, amelyen a file van, aktuális lemezegység esetén elhagyható. Mindegyik esetben a DDZT először betöltődik a 100H címre, majd önmagát a felhasználói terület (TPA) végére helyezi, közvetlenül az UPM elé, felülírva ezzel a CCP-t. A felhasználó rendelkezésére álló terület vége ilyenkor az 05H címen található JP utasítás operandusában található meg. A CALL 5 utasítások a DDZT-n keresztül hajtódnak végre, de ebből a felhasználó nem vesz észre semmit.

A memória felosztása:

U P M
DDZT
T P A
Rendszerváltozók
JMP DDZT

A program második formájú betöltésénél is ugyanaz történik mint az elsőnél, azzal a különbséggel, hogy a „file-név” programot betölti a felhasználói területre. Bármilyen file betölthető, de futtatni csak a futtatható formátumú file-okat lehet. HEX formátumú programból a DDZT állít elő futtatható formátumút.

Betöltés után a DDZT a következő üzenettel jelentkezik a konzolon:

```
--- DDZT DEBUGGER VER n.n ---
```

ahol n.n a verziószám. Utána *-gal jelzi, hogy kész a parancsok fogadására. A parancsok a következők lehetnek:

- D memória terület listázás
- F memória terület töltése konstanssal
- G indítás tetszőleges törésponttal
- I beírás a standard FCB-be
- M memória szegmens mozgatás
- R program beolvasás
- S memória érték módosítás
- T nyomkövetés
- U nem nyomkövetett program figyelés
- X CPU állapot vizsgálat

A parancs sorokat a következő standard UPM vezérlő karakterekkel lehet szerkeszteni:

- < (kurzor balra) utolsó karakter törlése
- ctrl U az egész sor törlése, után várja az elrontott parancssor helyesbítését.

A parancssor lezárása CR (carriage return) karakter gépelésével történik.

A parancsoknak lehetnek paramétereik, ezek helyköz kihagyása nélkül követik a parancsot és vesszővel vagy space-szel vannak elválasztva egymástól. A számértékek mindig hexadecimálisan értendők, mind paraméterként, mind memória vagy regiszter értéként.

A paramétereknél a debugger a parancstól függően mindig az utolsó kettő ill. négy számjegyet tekinti érvényesnek.

Pl.:

D100002000 ugyanaz mint D2000

Igy lehetőség van törlés nélküli javításra. A számértéket megelőző értéktelen nullákat nem szükséges kiírni. Ha valamelyik paramétert nem adják meg, az egyszerűen elhagyható, de a vesszőt ki kell tenni, ha van még paraméter.

Hibás parancsmegadás esetén kérdőjelet ír ki a program.

A DDZT futása alatt bármikor megállítható a ctrl C standard UPM vezérlő karakterrel (vagy GO paranccsal), ilyenkor visszatér az operációs rendszer szintjére. Visszatérés után a memóriatartalom elmenthető az UPM SAVE paranccsával.

A D, T, U parancsok végrehajtása egy karakter leütésével félbeszakítható.

A DDZT az RST28H-t használja, így ezt a rendszerben szabadon kell hagyni. A tesztelt program stackjében egy szintet igényel, ezt biztosítani kell számára.

2.3. Parancsok

2.3.1. D parancs

A D paranccsal kilistázható a memória hexadecimális és ASCII formában.

Formája: Dx,y

ahol x a listázandó memória terület kezdőcíme, y pedig a végcíme. A konzolon a következő formában jelenik meg a memóriakép:

cccc:

mem. cím 80H-nként

hh hh hh hh hh hh hh hh hh hh hh hh hh hh hh a a a a a a a a a a a a

memóriaérték hexadecimálisan

mem. érték
ASCII formában

A nem nyomtatható karakterek helyére pont (.) kerül az ASCII mezőben.

Ha az y paramétert elhagyjuk, az x címtől kezdve listáz 1 rekordot (80H).

A program az x értékét 10H-ra le, az Y-t pedig 10H-ra felkerekíti.

Az x elhagyása esetén az aktuális PC címtől listáz az y címig.

A paraméterek elhagyása esetén az aktuális címtől (betöltés után 100H)= listáz ki egy rekordot.

Túl hosszú listázáskor egy karakter leütésével megállítható a parancs végrehajtás.

Példák:

*D

*D,200

*D200,2FF

2.3.2. F parancs

Memória terület feltöltése konstanssal

Formája: Fk,v,h

ahol k a kezdőcím, v a végcím, ha pedig egy hexadecimális érték, mellyel az k-től v-ig tartó területet feltölti az F parancs, az v címet is beleértve.

2.3.3. G parancs

Programvégrehajtást a G paranccsal lehet indítani, 3 tetszőleges töréspont megadásával.

Formája: Gt,x,y,z

ahol t az indítási cím, x, y, és z pedig a töréspontok. A program futása a t címen indul, és az x, y, z címek elérésének bármelyikéig tart. Töréspontra futáskor a debugger kiírja a CPU állapotot az x parancs formátuma szerint.

A töréspontok közül egy, vagy több, akár az összes elhagyható, ilyenkor a DDZT nem állítja meg a program futást.

A t indítási cím elhagyása esetén az aktuális PC értéktől kezdődik a futás. Ha nem a program terület alatt van a töréspont, a vezérlés nem tud visszatérni a DDZT-be, és végrehajtás nélkül RST28H-ra fut.

2.3.4. I parancs

Lehetővé teszi file-név beírását a standard FCB-be.

Formája: Ix:file-név1 Y:file-név2

Az 5CH címre a file-név1-ből, a 6CH címre a file-név2-ből készíti FCB-t. 80H-ra teszi a parancs karakterszámot és 81H-tól a parancs sort (ld. UPM op. rendsz. fkk.).

Ha a file-ok HEX vagy COM típusúak, későbbi R parancssal be tudjuk olvasni a bináris vagy hexa formátumú programot.

2.3.5. M parancs

Lehetővé teszi memória szegmens mozgatását egyik helyről a másikra.

Formája: Mk,v,c

ahol k a másolandó terület kezdő címe, v a másolandó terület végcíme, c pedig a célállomás kezdőcíme.

2.3.6. R parancs

R paranccsal az I paranccsal kijelölt file-t beolvassa a memóriába.

Formája: Rx

ahol x tetszőleges eltérési cím ami a betöltési címhez hozzáadódik. Ha x-et elhagyjuk, x=0. Egy meghatározott címre fordított hexa program esetén cím+x-re, egyébként (100H+x) címre töltődik. HEX file esetén a file-t futtatható formátumúra konvertálja.

A DDZT betöltésénél a következő parancs:

DDZT x:file-név

ugyanolyan hatású mint a következő parancsorozat:

DZT

Ix:file-név

R

2.3.7. S parancs

Lehetővé teszi memória hely vizsgálatát és tetszőleges változtatását.

Formája: Sx

Az x címtől kezdve byte-onként írja ki a memória tartalmát. Minden byte után választ vár a felhasználótól.

Hexadecimális érték gépelésekor ez az érték lesz a memória byte új tartalma. CR-re a byte értéke nem változik. R karakter gépelésétől kezdve visszafelé fog haladni a memóriában a következő R gépeléséig.

Nem hexadecimális érték, vagy pont (.) gépelésekor kilép az S parancsból.

2.3.8. T parancs

T parancssal lépésenként követhetjük a program végrehajtását.

Formája: Tx

x utasítást végrehajt az aktuális PC címtől kezdve. Minden utasítás végrehajtása után kiírja a CPU állapotot a konzolra az X parancs formátuma szerint. Túl nagy x érték megadása esetén a nyomkövetés egy karakter leütésével megállítható.

2.3.9. U parancs

Lépésenkénti program végrehajtás

Formája: Ux

Hatása ugyanaz mint a T parancsnek, azzal a különbséggel, hogy csak az utolsó lépés végrehajtása után írja ki a konzolra a CPU állapotot.

2.3.10 X parancs

Regisztervizsgáló és módosító parancs

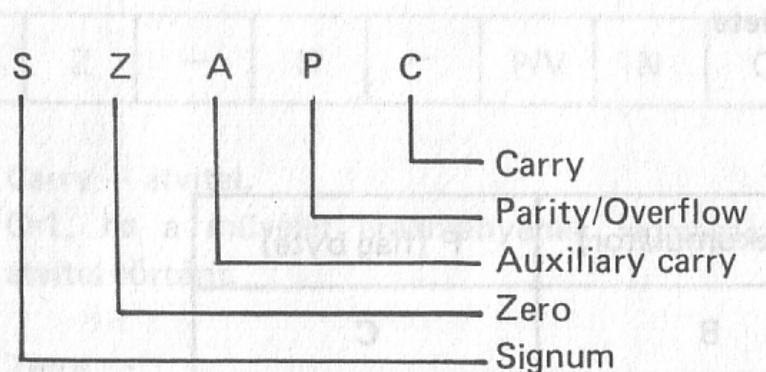
Formája: Xr

A parancs az r regiszter értékét kiírja a konzolra és választ vár a felhasználótól. Ha a válasz CR, a regiszter értéke nem változik. Hexadecimális érték gépelése esetén ez az érték lesz a regiszter új értéke.

Ha az r paramétert elhagyjuk, az összes regisztert és a következő 3 byte értékét (NI) jelzi ki az alábbi formában:

```
----- A=00 BC=0000 DE=0000 HL=0000 SP=0000 PC= 0000 NI=00 00 00  
----- A'00 BC'0000 DE'0000 HL'0000 IX'0000 IY'0000 I=00 R=00
```

A – jel a flagek 0 értékét jelzi, 1 érték esetén a flag nevének kezdőbetűje van a – jel helyén. A kijelzésnél a flagek sorrendje a következő:



A Flag byte felépítése a Függelékben található. Az IX, IY regiszterekre és a flag byte-ra való hivatkozás a következőképpen történik:

- XX
- XY
- XF

2.4. Hibaüzenetek

COMMAND SYNTAX ERROR

DDZT paramétereinek hibás megadása

HEX FILE ERROR

Egy hex file-ban nem hexadecimális karakter fordul elő, vagy check summa hiba van.

FILE NOT FOUND

A kijelölt file-t nem tudja megnyitni.

A parancsok paramétereinek hibája esetén a parancs végre nem hajtása jelzi a hibás parancs megadást.

3. FÜGGELÉK

3.1. A CPU regiszterkészlete

Műveleti regiszterek:

A (akkumulátor)	F (flag byte)
B	C
D	E
H	L

Segéd regiszterek:

A'	F'
B'	C'
D'	E'
H'	L'

Speciális műveleti regiszterek:

IX indexregiszter	
IY indexregiszter	
SP stack pointer	
PC utasítás számláló	
I (IT (lapcím))	R mem. refr.

A flag byte felépítése:

S	Z	—	H	—	P/V	N	C
---	---	---	---	---	-----	---	---

- C** Carry – átvitel.
C=1, ha a művelet eredményének legmagasabb helyiértékű bitjéről átvitel történt.
- Z** Zerus
Z=1, ha a művelet eredménye nulla.
- S** Signum – előjel.
S=1, ha az eredmény legmagasabb helyiértékű bitje=1.
- P/V** Parity/Overflow
Logikai műveleteknél paritást, aritmetikai műveleteknél túlcsordulást jelöl.
P=1, ha páros a paritás
V=1 túlcsorduláskor.
- H** Half carry
Átvétel fél byte-ról.
H=1, ha összeadási vagy kivonási műveleteknél átvitel vagy áthozat történt a 4. bitpozícióról.
- Nem használt bit.

3.2. Utasítások műveleti kód szerint

műv. kód	gépi kód	műv. kód	gépi kód
ADC A, (HL)	8E	ADD IY,BC	FD 09
ADC A, (IX+d)	DD 8E dd	ADD IY,DE	FD 19
ADC A, (IY+d)	FD 8E dd	ADD IY,IY	FD 29
ADC A,A	8F	ADD IY,SP	FD 39
ADC A,B	88	AND (HL)	A6
ADC A,C	89	AND (IX+d)	DD A6 dd
ADC A,D	8A	AND (IY+d)	FD A6 dd
ADC A,E	8B	AND A	A7
ADC A,H	8C	AND B	A0
ADC A,L	8D	AND C	A1
ADC A,n	CE nn	AND D	A2
ADC HL,BC	ED 4A	AND E	A3
ADC HL,DE	ED 5A	AND H	A4
ADC HL,HL	ED 6A	AND L	A5
ADC HL,SP	ED 7A	AND n	E6 nn
ADD A, (HL)	86	BIT 0, (HL)	CB 46
ADD A, (IX+d)	DD 86 dd	BIT 0, (IX+d)	DD CB dd 46
ADD A, (IY+d)	FD 86 dd	BIT 0, (IY+d)	FD CB dd 46
ADD A,A	87	BIT 0,A	CB 47
ADD A,B	80	BIT 0,B	CB 40
ADD A,C	81	BIT 0,C	CB 41
ADD A,D	82	BIT 0,D	CB 42
ADD A,E	83	BIT 0,E	CB 43
ADD A,H	84	BIT 0,H	CB 44
ADD A,L	85	BIT 0,L	CB 45
ADD A,n	C6 nn	BIT 1, (HL)	CB 4E
ADD HL,BC	09	BIT 1, (IX+d)	DD CB dd 4E
ADD HL,DE	19	BIT 1,A	CB 4F
ADD HL,HL	29	BIT 1,B	CB 48
ADD HL,SP	39	BIT 1,C	CB 49
ADD IX,BC	DD 09	BIT 1,D	CB 4A
ADD IX,DE	DD 19	BIT 1,E	CB 4B
ADD IX,IX	DD 29	BIT 1,H	CB 4C
ADD IX,SP	DD 39		

műveleti kód	gépi kód	műveleti kód	gépi kód
BIT 1,L	CB 4D	BIT 5,B	CB 68
BIT 2, (HL)	CB 56	BIT 5,C	CB 69
BIT 2, (IX+d)	DD CB dd 56	BIT 5,D	CB 6A
BIT 2, (IY+d)	FD CB dd 56	BIT 5,E	CB 6B
BIT 2,A	CB 57	BIT 5,H	CB 6C
BIT 2,B	CB 50	BIT 5,L	CB 6D
BIT 2,C	CB 51	BIT 6,(HL)	CB 76
BIT 2,D	CB 52	BIT 6,(IX+d)	DD CB dd 76
BIT 2,E	CB 53	BIT 6,(IY+d)	FD CB dd 76
BIT 2,H	CB 54	BIT 6,A	CB 77
BIT 2,L	CB 55	BIT 6,B	CB 70
BIT 3, (HL)	CB 5E	BIT 6,C	CB 71
BIT 3, (IX+d)	DD CB dd 5E	BIT 6,D	CB 72
BIT 3, (IY+d)	FD CB dd 5E	BIT 6,E	CB 73
BIT 3,A	CB 5F	BIT 6,H	CB 74
BIT 3,B	CB 58	BIT 6,L	CB 75
BIT 3,C	CB 59	BIT 7,(HL)	CB 7E
BIT 3,D	CB 5A	BIT 7,(IX+d)	DD CB dd 7E
BIT 3,E	CB 5B	BIT 7,(IY+d)	FD CB dd 7E
BIT 3,H	CB 5C	BIT 7,A	CB 7F
BIT 3,L	CB 5D	BIT 7,B	CB 78
BIT 4, (HL)	CB 66	BIT 7,C	CB 79
BIT 4, (IX+d)	DD CB dd 66	BIT 7D	CB 7A
BIT 4, (IY+d)	FD CB dd 66	BIT 7,E	CB 7B
BIT 4,A	CB 67	BIT 7,H	CB 7C
BIT 4,B	CB 60	BIT 7,L	CB 7D
BIT 4,C	CB 61	CALL C,ml	DC ll mm
BIT 4,D	CB 62	CALL M,ml	FC ll mm
BIT 4,E	CB 63	CALL NC,ml	D4 ll mm
BIT 4,H	CB 64	CALL ml	CD ll mm
BIT 4,L	CB 65	CALL NZ,ml	C4 ll mm
BIT 5, (HL)	CB 6E	CALL P,ml	F4 ll mm
BIT 5, (IX+d)	DD CB dd 6E	CALL PE,ml	EC ll mm
BIT 5, (IY+d)	FD CB dd 6E	CALL PO,ml	E4 ll mm
BIT 5,A	CB 6F	CALL Z,ml	CC ll mm

műveleti kód	gépi kód	műveleti kód	gépi kód
CCF	3F	DJNZ e	10 ee
CP (HL)	BE	EI	FB
CP (IX+d)	DD BE dd	EX (SP),HL	E3
CP (IY+d)	FD BE dd	EX (SP),IX	DD E3
CP A	BF	EX (SP),IY	FD E3
CP B	B8	EX AF,AF'	08
CP C	B9	EX DE,HL	EB
CP D	BA	EXX	D9
CP E	BB	HALT	76
CP H	BC	IM 0	ED 46
CP L	BD	IM 1	ED 56
CP n	FE nn	IM 2	ED 5E
CPD	ED A9	IN A,(C)	ED 78
CPDR	ED B9	IN A,(n)	DB nn
CPI	ED A1	IN B,(C)	ED 40
CPIR	ED B1	IN C,(C)	ED 48
CPL	2F	IN D,(C)	ED 50
DAA	27	IN E,(C)	ED 58
DEC (HL)	35	IN H,(C)	ED 60
DEC (IX+d)	DD 35 dd	IN L,(C)	ED 68
DEC (IY+d)	FD 35 dd	INC (HL)	34
DEC A	3D	INC (IX+d)	DD 34 dd
DEC B	05	INC (IY+d)	FD 34 dd
DEC BC	0B	INC A	3C
DEC C	0D	INC B	04
DEC D	15	INC BC	03
DEC DE	1B	INC C	0C
DEC E	1D	INC D	14
DEC H	25	INC DE	13
DEC HL	2B	INC E	1C
DEC IX	DD 2B	INC H	24
DEC IY	FD 2B	INC HL	23
DEC L	2D	INC IX	DD 23
DEC SP	3B	INC IY	FD 23
DI	F3	INC L	2C

műveleti kód	gépi kód	műveleti kód	gépi kód
INC SP	33	LD (IX+d),D	DD 72 dd
IND	ED AA	LD (IX+d),E	DD 73 dd
INDR	ED BA	LD (IX+d),H	DD 74 dd
INI	ED A2	LD (IX+d),L	DD 75 dd nn
INIR	ED B2	LD (IX+d),n	DD 36 dd nn
JP (HL)	E9	LD (IY+d),A	FD 77 dd
JP (IX)	DD E9	LD (IY+d),B	FD 70 dd
JP (IY)	FD E9	LD (IY+d),C	FD 71 dd
JP ml	C3 ll mm	LD (IY+d),D	FD 72 dd
JP C,ml	DA ll mm	LD (IY+d),E	FD 73 dd
JP M,ml	FA ll mm	LD (IY+d),H	FD 74 dd
JP NC,ml	D2 kk nn	LD (IY+d),L	FD 75 dd
JP NZ,ml	C2 ll mm	LD (IY+d),n	FD 36 dd nn
JP P,ml	F2 ll mm	LD (ml),A	32 ll mm
JP PE,ml	EA ll mm	LD (ml),BC	ED 43 ll mm
JP PO,ml	E2 ll mm	LD (ml),HL	22 ll mm
JP Z,ml	CA ll mm	LD (ml),IX	DD 22 ll mm
JR C,e	38 ee	LD (ml),IY	FD 22 ll mm
JR e	18 ee	LD (ml),SP	ED 73 ll mm
JR NC,e	30 ee	LD A,(BC)	0A
JR NZ,e	20 ee	LD A,(DE)	1A
JR Z,e	28 ee	LD A,(HL)	7E
LD (BC),A	02	LD A,(IX+d)	DD 7E dd
LD (DE),A	12	LD A,(IY+d)	FD 7E dd
LD (HL),A	77	LD A,(ml)	3A ll mm
LD (HL),B	70	LD A,A	7F
LD (HL),C	71	LD A,B	78
LD (HL),D	72	LD A,C	79
LD (HL),E	73	LD A,D	7A
LD (HL),H	74	LD A,E	7B
LD (HL),L	75	LD A,H	7C
LD (HL),n	36 nn	LD A,I	ED 57
LD (IX+d),A	DD 77 dd	LD A,L	7D
LD (IX+d),B	DD 70 dd	LD A,n	3E nn
LD (IX+d),C	DD 71 dd	LD A,R	ED 5F

műveleti kód	gépi kód	műveleti kód	gépi kód
LD B,(HL)	46	LD DE,ml	11 ll mm
LD B,(IX+d)	DD 46 dd	LD E,(HL)	5E
LD B,(IY+d)	FD 46 dd	LD E,(IX+d)	DD 5E dd
LD B,A	47	LD E,(IY+d)	FD 5E dd
LD B,B	40	LD E,A	5F
LD B,C	41	LD E,B	58
LD B,D	42	LD E,C	59
LD B,E	43	LD E,D	5A
LD B,H	44	LD E,E	5B
LD B,L	45	LD E,H	5C
LD B,n	06 nn	LD E,L	5D
LD BC,(ml)	ED 4B ll mm	LD E,n	1E nn
LD BC,ml	01 ll mm	LD H,(HL)	66
LD C,(HL)	4E	LD H,(IX+d)	DD 66 dd
LD C,(IX+d)	DD 4E dd	LD H,(IY+d)	FD 66 dd
LD C,(IY+d)	FD 4E dd	LD H,A	67
LD C,A	4F	LD H,B	60
LD C,B	48	LD H,C	61
LD C,C	49	LD H,D	62
LD C,D	4A	LD H,E	63
LD C,E	4B	LD H,H	64
LD C,H	4C	LD H,L	65
LD C,n	0E nn	LD H,n	26 nn
LD D,(HL)	56	LD HL,(ml)	2A ll mm
LD D,(IX+d)	DD 56 dd	LD HL,ml	21 ll mm
LD D,(IY+d)	FD 56 dd	LD I,A	ED 47
LD D,A	57	LD IX,(ml)	DD 2A ll mm
LD D,B	50	LD IX,ml	DD 21 ll mm
LD D,C	51	LD IY,(ml)	FD 2A ll mm
LD D,D	52	LD IY,ml	FD 21 ll mm
LD D,E	53	LD L,(HL)	6E
LD D,H	54	LD L,(IX+d)	DD 6E dd
LD D,L	55	LD L,(IY+d)	FD 6E dd
LD D,n	16 nn	LD L,A	6F
LD DE,(ml)	ED 5B ll mm	LD L,B	68

műveleti kód	gépi kód	műveleti kód	gépi kód
LD L,C	69	OUT (C),E	ED 59
LD L,D	6A	OUT (C),H	ED 61
LD L,E	6B	OUT (C),L	ED 69
LD L,H	6C	OUT (n),A	D3 nn
LD L,L	6D	OUTD	ED AB
LD L,n	2E nn	OUTI	ED A3
LD R,A	ED 4F	POP AF	F1
LD SP,(ml)	ED 7B ll mm	POP BC	C1
LD SP,HL	F9	POP DE	D1
LD SP,IX	DD F9	POP HL	E1
LD SP,IY	FD F9	POP IX	DD E1
LD SP,ml	31 ll mm	POP IY	FD E1
LDD	ED A8	PUSH AF	F5
LDDR	ED B8	PUSH BC	C5
LDI	ED A0	PUSH DE	D5
LDIR	ED B0	PUSH HL	E5
NEG	ED 44	PUSH IX	DD E5
NOP	00	PUSH IY	FD E5
OR (HL)	B6	RES 0,(HL)	CB 86
OR (IX+d)	DD B6 dd	RES 0,(IX+d)	DD CB dd 86
OR (IY+d)	FD B6 dd	RES 0,(IY+d)	FD CB dd 86
OR A	B7	RES 0,A	CB 87
OR B	B0	RES 0,B	CB 80
OR C	B1	RES 0,C	CB 81
OR D	B2	RES 0,D	CB 82
OR E	B3	RES 0,E	CB 83
OR H	B4	RES 0,H	CB 84
OR L	B5	RES 0,L	CB 85
OR n	F6 nn	RES 1,(HL)	CB 8E
OTDR	ED BB	RES 1,(IX+d)	DD CB dd 8E
OTIR	ED B3	RES 1,(IY+d)	FD CB dd 8E
OUT (C),A	ED 79	RES 1,A	CB 8F
OUT (C),B	ED 41	RES 1,B	CB 88
OUT (C),C	ED 49	RES 1,C	CB 89
OUT (C),D	ED 51	RES 1,D	CB 8A

műveleti kód	gépi kód	műveleti kód	gépi kód
RES 1,E	CB 8B	RES 5, (IY+d)	FD CB dd AE
RES 1,H	CB 8C	RES 5,A	CB AF
RES 1,L	CB 8D	RES 5,B	CB A8
RES 2,(HL)	CB 96	RES 5,C	CB A9
RES 2,(IX+d)	DD CB dd 96	RES 5,D	CB AA
RES 2,(IY+d)	FD CB dd 96	RES 5,E	CB AB
RES 2,A	CB 97	RES 5,H	CB AC
RES 2,B	CB 90	RES 5,L	CB AD
RES 2,C	CB 91	RES 6,(HL)	CB B6
RES 2,D	CB 92	RES 6'(IX+d)	DD CB dd B6
RES 2,E	CB 93	RES 6,(IY+d)	FD BC dd B6
RES 2,H	CB 94	RES 6,A	CB B7
RES 2,L	CB 95	RES 6,B	CB B0
RES 3,(HL)	CB 9E	RES 6,C	CB B1
RES 3,(IX+d)	DD CB dd 9E	RES 6,D	CB B2
RES 3,(IY+d)	FD CB dd 9E	RES 6,E	CB B3
RES 3,A	CB 9F	RES 6,H	CB B4
RES 3,B	CB 98	RES 6,L	CB B5
RES 3,C	CB 99	RES 7,(HL)	CB BE
RES 3,D	CB 9A	RES 7,(IX+d)	DD CB dd BE
RES 3,E	CB 9B	RES 7,(IY+d)	FD CB dd BE
RES 3,H	CB 9C	RES 7,A	CB BF
RES 3,L	CB 9D	RES 7,B	CB B8
RES 4,(HL)	CB A6	RES 7,C	CB B9
RES 4,(IX+d)	DD CB dd A6	RES 7,D	CB BA
RES 4,(IY+d)	FD CB dd A6	RES 7,E	CB BB
RES 4,A	CB A7	RES 7,H	CB BC
RES 4,B	CB A0	RES 7,L	CB BD
RES 4,C	CB A1	RET	C9
RES 4,D	CB A2	RET C	D8
RES 4,E	CB A3	RET M	F8
RES 4,H	CB A4	RET NC	D0
RES 4,L	CB A5	RET NZ	C0
RES 5,(HL)	CB Ae	RET P	F0
RES 5,(IX+d)	DD CB dd AE	RET PE	E8

műveleti kód	gépi kód	műveleti kód	gépi kód
RET PO	E0	RR H	CB 1C
RET Z	C8	RR L	CB 1D
RETI	ED 4D	RR A	1F
RETN	ED 45	RRC (HL)	CB 0E
RL (HL)	CB 16	RRC (IX+d)	DD CB dd 0E
RL (IX+d)	DD CB dd 16	RRC (IY+d)	FD CB dd 0E
RL (IY+d)	FD CB dd 16	RRC A	CB 0F
RL A	CB 17	RRC B	CB 08
RL B	CB 10	RRC C	CB 09
RL C	CB 11	RRC D	CB 0A
RL D	CB 12	RRC E	CB 0B
RL E	CB 13	RRC H	CB 0C
RL H	CB 14	RRC L	CB 0D
RL L	CB 15	RRCA	0F
RLA	17	RRD	ED 67
RLC (HL)	CB 06	RST 0	C7
RLC (IX+d)	DD CB dd 06	RST 08H	CF
RLC (IY+d)	FD CB dd 06	RST 10H	D7
RLC A	CB 07	RST 18H	DF
RLC B	CB 00	RST 20H	E7
RLC C	CB 01	RST 28H	EF
RLC D	CB 02	RST 30H	F7
RLC E	CB 03	RST 38H	FF
RLC H	CB 04	SBC A,(HL)	9E
RLC L	CB 05	SBC A,(IX+d)	DD 9E dd
RLCA	07	SBC A,(IY+d)	FD 9E dd
RLD	ED 6F	SBC A,A	9F
RR (HL)	CB 1E	SBC A,B	98
RR (IX+d)	DD CB dd 1E	SBC A,C	99
RR (IY+d)	FD CB dd 1E	SBC A,D	9A
RR A	CB 1F	SBC A,E	9B
RR B	CB 18	SBC A,H	9C
RR C	CB 19	SBC A,L	9D
RR D	CB 1A	SBC A,n	DE nn
RR E	CB 1B	DBC HL,BC	ED 42

műveleti kód	gépi kód	műveleti kód	gépi kód
SBC HL,DE	ED 52	SET 3,(IX+d)	DD CB dd DE
SBC HL,HL	ED 62	SET 3,(IY+d)	FD CB dd DE
SBC HL,SP	ED 72	SET 3,A	CB DF
SCF	37	SET 3,B	CB D8
SET 0,(HL)	CB C6	SET 3,C	CB D9
SET 0,(IX+d)	DD CB dd C6	SET 3,D	CB DA
SET 0,(IY+d)	FD CB dd C6	SET 3,E	CB DB
SET 0,A	CB C7	SET 3,H	CB DC
SET 0,B	CB C0	SET 3,L	CB DD
SET 0,C	CB C1	SET 4,(HL)	CB E6
SET 0,D	CB C2	SET 4,(IX+d)	DD CB dd E6
SET 0,E	CB C3	SET 4,(IY+d)	FD CB dd E6
SET 0,H	CB C4	SET 4,A	CB E7
SET 0,L	CB C5	SET 4,B	CB E0
SET 1,(HL)	CB CE	SET 4,C	CB E1
SET 1,(IX+d)	DD CB dd CE	SET 4,D	CB E2
SET 1,(IY+d)	FD CB dd CE	SET 4,E	CB E3
SET 1,A	CB CF	SET 4,H	CB E4
SET 1,B	CB C8	SET 4,L	CB E5
SET 1,C	CB C9	SET 5,(HL)	CB EE
SET 1'D	CB CA	SET 5,(IX+d)	DD CB dd EE
SET 1,E	CB CB	SET 5,(IY+d)	FD CB dd EE
SET 1,H	CB CC	SET 5,A	CB EF
SET 1,L	CB CD	SET 5,B	CB E8
SET 2,(HL)	CB D6	SET 5,C	CB E9
SET 2,(IX+d)	DD CB dd D6	SET 5,D	CB EA
SET 2,(IY+d)	FD CB dd D6	SET 5,E	CB EB
SET 2,A	CB D7	SET 5,H	CB EC
SET 2,B	CB D0	SET 5,L	CB ED
SET 2,C	CB D1	SET 6,(HL)	CB F6
SET 2,D	CB D2	SET 6,(IX+d)	DD CB dd F6
SET 2,E	CB D3	SET 6,(IY+d)	FD CB dd F6
SET 2,H	CB D4	SET 6,A	CB F7
SET 2,L	CB D5	SET 6,B	CB F0
SET 3,(HL)	CB DE	SET 6,C	CB F1

műveleti kód	gépi kód	műveleti kód	gépi kód
SET 6,D	CB F2	SRA L	CB 2D
SET 6,E	CB F3	SRL(HL)	CB 3E
SET 6,H	CB F4	SRL (IX+d)	DD CB dd 3E
SET 6,L	CB F5	SRL (IY+d)	FD CB dd 3E
SET 7,(HL)	CB FE	SRL A	CB 3F
SET 7,(IX+d)	DD CB dd FE	SRL B	CB 38
SET 7,(IY+d)	FD CB dd FE	SRL C	CB 39
SET 7,A	CB FF	SRL D	CB 3A
SET 7,B	CB F8	SRL E	CB 3B
SET 7,C	CB F9	SRL H	CB 3C
SET 7,D	CB FA	SRL L	CB 3D
SET 7'E	CB FB	SUB (HL)	96
SET 7'H	CB FC	SUB (IX+d)	DD 96 dd
SET 7,L	CB FD	SUB (IY+d)	FD 96 dd
SLA (HL)	CB 26	SUB A	97
SLA (IX+D)	DD CB dd 26	SUB B	90
SLA (IY+d)	FD CB dd 26	SUB C	91
SLA A	CB 27	SUB D	92
SLA B	CB 20	SUB E	93
SLA C	CB 21	SUB H	94
SLA D	CB 22	SUB L	95
SLA E	CB 23	SUB n	D6 nn
SLA H	CB 24	XOR (HL)	AE
SLA L	CB 25	XOR (IX+d)	DD AE dd
SRA (HL)	CB 2E	XOR (IY+d)	FD AE dd
SRA (IX+d)	DD CB dd 2E	XOR A	AF
SRA (IY+d)	FD CB dd 2E	XOR B	A8
SRA A	CB 2F	XOR C	A9
SRA B	CB 28	XOR D	AA
SRA C	CB 29	XOR E	AB
SRA D	CB 2A	XOR H	AC
SRA E	CB 2B	XOR L	AD
SRA H	CB 2C	XOR n	EE nn

Jelölések:

d	8 bites eltolási érték indexregiszteres címzésnél
dd	8 bites eltolási érték indexregiszteres gépi kódú ábrázolása
n	8 bites konstans vagy kifejezés
nn	8 bites konstans vagy kifejezés gépi kódú ábrázolása
ml	16 bites konstans vagy kifejezés, m a magasabb, l az alacsonyabb helyiértékű byte
mm	16 bites érték magasabb helyiértékű byte-jának gépi kódú ábrázolása
ll	16 bites érték alacsonyabb helyiértékű byte-jának gépi kódú ábrázolása
e	16 bites memóriacím relatív címzésnél
ee	e-ből képzett 8 bites érték gépi kódú ábrázolása
()	Zárójelben levő operandus az általa megcímzett memóriabyte-ra való hivatkozást jelent.

3.3. Utasítások gépi kód szerint

gépi kód	műv. kód	gépi kód	műv. kód
02	LD (BC),A	11 ll mm	LD DE,ml
03	INC BC	12	LD (DE),A
04	INC B	13	INC DE
05	DEC B	14	INC D
06 nn	LD B,n	15	DEC D
07	RLCA	16 nn	LD D,n
08	EX AF,AF'	17	RLA
09	ADD HL,BC	18 ee	JR e
0A	LD A,(BC)	19	ADD HL,DE
0B	DEC BC	1A	LD A,(DE)
0C	INC C	1B	DEC DE
0D	DEC C	1C	INC E
0E nn	LD C,n	1D	DEC E
0F	RRCA	1E nn	LD E,n
10 ee	DJNZ e	1F	RRA

műveleti kód	gépi kód	műveleti kód	gépi kód
20 ee	JR NZ,e	44	LD B,H
21 ll mm	LD HL,ml	45	LD B,L
22 ll mm	LD (ml),HL	46	LD B,(HL)
23	INC HL	47	LD B,A
24	INC H	48	LD C,B
25	DEC H	49	LD C,C
26 nn	LD H,n	4A	LD C,D
27	DAA	4B	LD C,E
28 ee	JR Z,e	4C	LD C,H
29	ADD HL,HL	4D	LD C,L
2A ll mm	LD HL, (ml)	4E	LD C,(HL)
2B	DEC HL	4F	LD C,A
2C	INC L	50	LD D,B
2D	DEC L	51	LD D,C
2E nn	LD L,n	52	LD D,D
2F	CPL	53	LD D,E
30 ee	JR NC,e	54	LD D,H
31 ll mm	LD SP,ml	55	LD D,L
32 ll mm	LD (ml),A	56	LD D,(HL)
33	INC SP	57	LD D,A
34	INC (HL)	58	LD E,B
35	DEC (HL)	59	LD E,C
36 nn	LD (HL),n	5A	LD E,D
37	SCF	5B	LD E,E
38 ee	JR C,e	5C	LD E,H
39	ADD HL,SP	5D	LD E,L
3A ll mm	LD A,(ml)	5E	LD E,(HL)
3B	DEC SP	5F	LD E,A
3C	INC A	60	LD H,B
3D	DEC A	61	LD H,C
3E nn	LD A,n	62	LD H,D
3F	CCF	63	LD H,E
40	LD B,B	64	LD H,H
41	LD B,C	65	LD H,L
42	LD B,D	66	LD H,(HL)
43	LD B,E	67	LD H,A

műveleti kód	gépi kód	műveleti kód	gépi kód
68	LD L,B	8B	ADC A,E
69	LD L,C	8C	ADC A,H
6A	LD L,D	8D	ADC A,L
6B	LD L,E	8E	ADC A,(HL)
6C	LD L,H	8F	ADC A,A
6D	LD L,L	90	SUB B
6E	LD L,(HL)	91	SUB C
6F	LD L,A	92	SUB D
70	LD (HL),B	93	SUB E
71	LD (HL),C	94	SUB H
72	LD (HL),D	95	SUB L
73	LD (HL),E	96	SUB (HL)
74	LD (HL),H	97	SUB A
75	LD (HL),L	98	SBC A,B
76	HALT	99	SBC A,C
77	LD (HL),A	9A	SBC A,D
78	LD A,B	9B	SBC A,E
79	LD A,C	9C	SBC A,H
7A	LD A,D	9D	SBC A,L
7B	LD A,E	9E	SBC A,(HL)
7C	LD A,H	9F	SBC A,A
7D	LD A,L	A0	AND B
7E	LD A,(HL)	A1	AND C
7F	LD A,A	A2	AND D
80	ADD A,B	A3	AND E
81	ADD A,C	A4	AND H
82	ADD A,D	A5	AND L
83	ADD A,E	A6	AND (HL)
84	ADD A,H	A7	AND A
85	ADD A,L	A8	XOR B
86	ADD A,(HL)	A9	XOR C
87	ADD A,A	AA	XOR D
88	ADC A,B	AB	XOR E
89	ADC A,C	AC	XOR H
8A	ADC A,D	AD	XOR L

műveleti kód

gépi kód

műveleti kód

gépi kód

AE	XOR (HL)
AF	XOR A
B0	OR B
B1	OR C
B2	OR D
B3	OR E
B4	OR H
B5	OR L
B6	OR (HL)
B7	OR A
B8	CP B
B9	CP C
BA	CP D
BB	CP E
BC	CP H
BD	CP L
BE	CP (HL)
BF	CP A
C0	RET NZ
C1	POP BC
C2 II mm	JP NZ,ml
C3 II mm	JP ml
C4 II mm	CALL NZ,ml
C5	PUSH BC
C6 nn	ADD A,n
C7	RST O
C8	RET Z
C9	RET
CA II mm	JP Z,ml
CC II mm	CALL Z,ml
CD II mm	CALL ml
CE nn	ADC A,n
CF	RST 8
D0	RET NC
D1	POP DE

D2 II mm
D3 nn
D4 II mm
D5
D6 nn
D7
D8
D9
DA II mm
DB nn
DC II mm
DE nn
DF
E0
E1
E2 II mm
E3
E4 II mm
E5
E6 nn
E7
E8
E9
EA II mm
EB
EC II mm
EE nn
EF
F0
F1
F2 II mm
F3
F4 II mm
F5
F6 nn

JP NC,ml
OUT (n),A
CALL NC,ml
PUSH DE
SUB n
RST 10H
RET C
EXX
JP C,ml
IN A,(n)
CALL C,ml
SBC A,n
RST 18H
RET PO
POP HL
JP PO,ml
EX (SP),HL
CALL PO,ml
PUSH HL
AND n
RST 20H
RET PE
JP (HL)
JP PE,ml
EX DE,HL
CALL PE,ml
XOR n
RST 28H
RET P
POP AF
JP P,ml
DI
CALL P,ml
PUSH AF
OR n

műveleti kód	gépi kód	műveleti kód	gépi kód
F7	RST 30H	CB 1B	RR E
F8	RET M	CB 1C	RR H
F9	LD SP,HL	CB 1D	RR L
FA ll mm	JP M,ml	CB 1E	RR (HL)
FB	EI	CB 1F	RR A
FC ll mm	CALL M,ml	CB 20	SLA B
FE nn	CP n	CB 21	SLA C
FF	RST 38H	CB 22	SLA D
CB 00	RLC B	CB 23	SLA E
CB 01	RLC C	CB 24	SLA H
CB 02	RLC D	CB 25	SLA L
CB 03	RLC E	CB 26	SLA (HL)
CB 04	RLC H	CB 27	SLA A
CB 05	RLC L	CB 28	SRA B
CB 06	RLC (HL)	CB 29	SRA C
CB 07	RLC A	CB 2A	SRA D
CB 08	RRC B	CB 2B	SRA E
CB 09	RRC C	CB 2C	SRA H
CB 0A	RRC D	CB 2D	SRA L
CB 0B	RRC E	CB 2E	SRA (HL)
CB 0C	RRC H	CB 2F	SRA A
CB 0D	RRC L	CB 38	SRL B
CB 0E	RRC (HL)	CB 39	SRL C
CB 0F	RRC A	CB 3A	SRL D
CB 10	RL B	CB 3B	SRL E
CB 11	RL C	CB 3C	SRL H
CB 12	RL D	CB 3D	SRL L
CB 13	RL E	CB 3E	SRL (HL)
CB 14	RL H	CB 3F	SRL A
CB 15	RL L	CB 40	BIT 0,B
CB 16	RL (HL)	CB 41	BIT 0,C
CB 17	RL A	CB 42	BIT 0,D
CB 18	RR B	CB 43	BIT 0,E
CB 19	RR C	CB 44	BIT 0,H
CB 1A	RR D	CB 45	BIT 0,L

műveleti kód	gépi kód	műveleti kód	gépi kód
CB 46	BIT 0,(HL)	CB 69	BIT 5,C
CB 47	BIT 0,A	CB 6A	BIT 5,D
CB 48	BIT 1,B	CB 6B	BIT 5,E
CB 49	BIT 1,C	CB 6C	BIT 5,H
CB 4A	BIT 1,D	CB 6D	BIT 5,L
CB 4B	BIT 1,E	CB 6E	BIT 5,(HL)
CB 4C	BIT 1,H	CB 6F	BIT 5,A
CB 4D	BIT 1,L	CB 70	BIT 6,B
CB 4E	BIT 1,(HL)	CB 71	BIT 6,C
CB 4F	BIT 1,A	CB 72	BIT 6,D
CB 50	BIT 2,B	CB 73	BIT 6,E
CB 51	BIT 2,C	CB 74	BIT 6,H
CB 52	BIT 2,D	CB 75	BIT 6,L
CB 53	BIT 2,E	CB 76	BIT 6,(HL)
CB 54	BIT 2,H	CB 77	BIT 6,A
CB 55	BIT 2,L	CB 78	BIT 7,B
CB 56	BIT 2,(HL)	CB 79	BIT 7,C
CB 57	BIT 2,A	CB 7A	BIT 7,D
CB 58	BIT 3,B	CB 7B	BIT 7,E
CB 59	BIT 3,C	CB 7C	BIT 7,H
CB 5A	BIT 3,D	CB 7D	BIT 7,L
CB 5B	BIT 3,E	CB 7E	BIT 7,(HL)
CB 5C	BIT 3,H	CB 7F	BIT 7,A
CB 5D	BIT 3,L	CB 80	RES 0,B
CB 5E	BIT 3,(HL)	CB 81	RES 0,C
CB 5F	BIT 3,A	CB 82	RES 0,D
CB 60	BIT 4,B	CB 83	RES 0,E
CB 61	BIT 4,C	CB 84	RES 0,H
CB 62	BIT 4,D	CB 85	RES 0,L
CB 63	BIT 4,E	CB 86	RES 0,(HL)
CB 64	BIT 4,H	CB 87	RES 0,A
CB 65	BIT 4,L	CB 88	RES 1,B
CB 66	BIT 4,(HL)	CB 89	RES 1,C
CB 67	BIT 4,A	CB 8A	RES 1,D
CB 68	BIT 5,B	CB 8B	RES 1,E

műveleti kód	gépi kód	műveleti kód	gépi kód
CB 8C	RES 1,H	CB AF	RES 5,A
CB 8D	RES 1,L	CB B0	RES 6,B
CB 8E	RES 1,(HL)	CB B1	RES 6,C
CB 8F	RES 1,A	CB B2	RES 6,D
CB 90	RES 2,B	CB B3	RES 6,E
CB 91	RES 2,C	CB B4	RES 6,H
CB 92	RES 2,D	CB B5	RES 6,L
CB 93	RES 2,E	CB B6	RES 6,(HL)
CB 94	RES 2,H	CB B7	RES 6,A
CB 95	RES 2,L	CB B8	RES 7,B
CB 96	RES 2,(HL)	CB B9	RES 7,C
CB 97	RES 2,A	CB BA	RES 7,D
CB 98	RES 3,B	CB BB	RES 7,E
CB 99	RES 3,C	CB BC	RES 7,H
CB 9A	RES 3,D	CB BD	RES 7,L
CB 9B	RES 3,E	CB BE	RES 7,(HL)
CB 9C	RES 3,H	CB BF	RES 7,A
CB 9D	RES 3,L	CB C0	SET 0,B
CB 9E	RES 3,(HL)	CB C1	SET 0,C
CB 9F	RES 3,A	CB C2	SET 0,D
CB A0	RES 4,B	CB C3	SET 0,E
CB A1	RES 4,C	CB C4	SET 0,H
CB A2	RES 4,D	CB C5	SET 0,L
CB A3	RES 4,E	CB C6	SET 0,(HL)
CB A4	RES 4,H	CB C7	SET 0,A
CB A5	RES 4,L	CB C8	SET 1,B
CB A6	RES 4,(HL)	CB C9	SET 1,C
CB A7	RES 4,A	CB CA	SET 1,D
CB A8	RES 5,B	CB CB	SET 1,E
CB A9	RES 5,C	CB CC	SET 1,H
CB AA	RES 5,D	CB CD	SET 1,L
CB AB	RES 5,E	CB CE	SET 1,(HL)
CB AC	RES 5,H	CB CF	SET 1,A
CB AD	RES 5,L	CB D0	SET 2,B
CB AE	RES 5,(HL)	CB D1	SET 2,C

műveleti kód

gépi kód

CB D2	SET 2,D
CB D3	SET 2,E
CB D4	SET 2,H
CB D5	SET 2,L
CB D6	SET 2,(HL)
CB D7	SET 2,A
CB D8	SET 3,B
CB D9	SET 3,C
CB DA	SET 3,D
CB DB	SET 3,E
CB DC	SET 3,H
CB DD	SET 3,L
CB DE	SET 3,(HL)
CB DF	SET 3,A
CB E0	SET 4,B
CB E1	SET 4,C
CB E2	SET 4,D
CB E3	SET 4,E
CB E4	SET 4,H
CB E5	SET 4,L
CB E6	SET 4,(HL)
CB E7	SET 4,A
CB E8	SET 5,B
CB E9	SET 5,C
CB EA	SET 5,D
CB EB	SET 5,E
CB EC	SET 5,H
CB ED	SET 5,L
CB EE	SET 5,(HL)
CB EF	SET 5,A
CB F0	SET 6,B
CB F1	SET 6,C
CB F2	SET 6,D
CB F3	SET 6,E
CB F4	SET 6,H

műveleti kód

gépi kód

CB F5	SET 6,L
CB F6	SET 6,(HL)
CB F7	SET 6,A
CB F8	SET 7,B
CB F9	SET 7,C
CB FA	SET 7,D
CB FB	SET 7,E
CB FC	SET 7,H
CB FD	SET 7,L
CB FE	SET 7,(HL)
CB FF	SET 7,A
DD 09	ADD IX,BC
DD 19	ADD IX,DE
DD 21 ll mm	LD IX,ml
DD 22 ll mm	LD (ml),IX
DD 23	INC IX
DD 29	ADD IX, IX
DD 2A ll mm	LD IX,(ml)
DD 2B	DEC IX
DD 34 dd	INC (IX+d)
DD 35 dd	DEC (IX+d)
DD 36 dd nn	LD (IX+d),n
DD 39	ADD IX,SP
DD 46 dd	LD B,(IX+d)
DD 4E dd	LD C,(IX+d)
DD 56 dd	LD D,(IX+d)
DD 5E dd	LD E,(IX+d)
DD 66 dd	LD H,(IX+d)
DD 6E dd	LD L,(IX+d)
DD 70 dd	LD (IX+d),B
DD 71 dd	LD (IX+d),C
DD 72 dd	LD (IX+d),D
DD 73 dd	LD (IX+d),E
DD 74 dd	LD (IX+d),H
DD 75 dd	LD (IX+d),L

műveleti kód	gépi kód	műveleti kód	gépi kód
DD 77 dd	LD (IX+d),A	DD CB dd AE	RES 5,(IX+d)
DD 7E dd	LD A,(IX+d)	DD CB dd B6	RES 6,(IX+d)
DD 86 dd	ADD A,(IX+d)	DD CB dd BE	RES 7,(IX+d)
DD 8E dd	ADC A,(IX+d)	DD CB dd C6	SET 0,(IX+d)
DD 96 dd	SUB (IX+d)	DD CB dd CE	SET 1,(IX+d)
DD 9E dd	SBC A,(IX+d)	DD CB dd D6	SET 2,(IX+d)
DD A6 dd	AND (IX+d)	DD CB dd DE	SET 3,(IX+d)
DD AE dd	XOR (IX+d)	DD CB dd E6	SET 4,(IX+d)
DD B6 dd	OR (IX+d)	DD CB dd EE	SET 5,(IX+d)
DD BE dd	CP (IX+d)	DD CB dd F6	SET 6,(IX+d)
DD E1	POP IX	DD CB dd FE	SET 7,(IX+d)
DD E3	EX (SP),IX	ED 40	IN B,(C)
DD E5	PUSH IX	ED 41	OUT (C),B
DD E9	JP (IX)	ED 42	SBC HL,BC
DD F9	LD SP,IX	ED 43 ll mm	LD (ml),BC
DD CB dd 06	RLC (IX+d)	ED 44	NEG
DD CB dd 0E	RRC (IX+d)	ED 45	RETN
DD CB dd 16	RL (IX+d)	ED 46	IM 0
DD CB dd 1E	RR (IX+d)	ED 47	LD I,A
DD CB dd 26	SLA (IX+d)	ED 48	IN C,(C)
DD CB dd 2E	SRA (IX+d)	ED 49	OUT (C),C
DD CB dd 3E	SRL (IX+d)	ED 4A	ADC HL,BC
DD CB dd 46	BIT 0,(IX+d)	ED 4B ll mm	LD BC,(ml)
DD CB dd 4E	BIT 1,(IX+d)	ED 4D	RETI
DD CB dd 56	BIT 2,(IX+d)	ED 4F	LD R,A
DD CB dd 5E	BIT 3,(IX+d)	ED 50	IN D,(C)
DD CB dd 66	BIT 4,(IX+d)	ED 51	OUT (C),D
DD CB dd 6E	BIT 5,(IX+d)	ED 52	SBC HL,DE
DD CB dd 76	BIT 6,(IX+d)	ED 53 ll mm	LD (ml),DE
DD CB dd 7E	BIT 7,(IX+d)	ED 56	IM 1
DD CB dd 86	RES 0,(IX+d)	ED 57	LD A,I
DD CB dd 8E	RES 1,(IX+d)	ED 58	IN E,(C)
DD CB dd 96	RES 2,(IX+d)	ED 59	OUT (C),E
DD CB dd 9E	RES 3,(IX+d)	ED 5A	ADC HL,DE
DD CB dd A6	RES 4,(IX+d)	ED 5B ll mm	LD DE,(ml)

műveleti kód	gépi kód	műveleti kód	gépi kód
ED 5E	IM 2	FD 22 II mm	LD (ml),IY
ED 5F	LD A,R	FD 23	INC IY
ED 60	IN H,(C)	FD 29	ADD IY,IY
ED 61	OUT (C),H	FD 2A II mm	LD IY,(ml)
ED 62	SBC HL,HL	FD 2B	DEC IY
ED 67	RRD	FD 34 dd	INC (IY+d)
ED 68	IN L,(C)	FD 35 dd	DEC (IY+d)
ED 69	OUT (C),L	FD 36 dd nn	LD (IY+d),n
ED 6A	ADC HL,HL	FD 39	ADD IY,SP
ED 6F	RLD	FD 46 dd	LD B,(IY+d)
ED 72	SBC HL,SP	FD 4E dd	LD C,(IY+d)
ED 73 II mm	LD (ml),SP	FD 56 dd	LD D,(IY+d)
ED 78	IN A,(C)	FD 5E dd	LD E,(IY+d)
ED 79	OUT (C),A	FD 66 dd	LD H,(IY+d)
ED 7A	ADC HL,SP	FD 6E dd	LD L,(IY+d)
ED 7B II mm	LD SP,(ml)	FD 70 dd	LD (IY+d),B
ED A0	LDI	FD 71 dd	LD (IY+d),C
ED A1	CPI	FD 72 dd	LD (IY+d),D
ED A2	INI	FD 73 dd	LD (IY+d),E
ED A3	OUTI	FD 74 dd	LD (IY+d),H
ED A8	LDD	FD 75 dd	LD (IY+d),L
ED A9	CPD	FD 77 dd	LD (IY+d),A
ED AA	IND	FD 7E dd	LD A,(IY+d)
ED AB	OUTD	FD 86 dd	ADD A,(IY+d)
ED B0	LDIR	FD 8E dd	ADC A,(IY+d)
ED B1	CPIR	FD 96 dd	SUB (IY+d)
ED B2	INIR	FD 9E dd	SBC A,(IY+d)
ED B3	OTIR	FD A6 dd	AND (IY+d)
ED B8	LDDR	FD AE dd	XOR (IY+d)
ED B9	CPDR	FD B6 dd	OR (IY+d)
ED BA	INDR	FD BE dd	CP (IY+d)
ED BB	OTDR	FD E1	POP IY
FD 09	ADD IY,BC	FD E3	EX (SP),IY
FD 19	ADD IY,DE	FD E5	PUSH IY
FD 21 II mm	LD IY,ml	FD E9	JP (IY)

műveleti kód	gépi kód	műveleti kód	gépi kód
FD F9	LD SP,IY	FD CB dd 86	RES 0,(IY+d)
FD CB dd 06	RLC (IY+d)	FD CB dd 8E	RES 1,(IY+d)
FD CB dd 0E	RRC (IY+d)	FD CB dd 96	RES 2,(IY+d)
FD CB dd 16	RL (IY+d)	FD CB dd 9E	RES 3,(IY+d)
FD CB dd 1E	RR (IY+d)	FD CB dd A6	RES 4,(IY+d)
FD CB dd 26	SLA (IY+d)	FD CB dd AE	RES 5,(IY+d)
FD CB dd 2E	SRA (IY+d)	FD CB dd B6	RES 6,(IY+d)
FD CB dd 3E	SRL (IY+d)	FD CB dd BE	RES 7,(IY+d)
FD CB dd 46	BIT 0,(IY+d)	FD CB dd C6	SET 0,(IY+d)
FD CB dd 4E	BIT 1,(IY+d)	FD CB dd CE	SET 1,(IY+d)
FD CB dd 56	BIT 2,(IY+d)	FD CB dd D6	SET 2,(IY+d)
FD CB dd 5E	BIT 3,(IY+d)	FD CB dd DE	SET 3,(IY+d)
FD CB dd 66	BIT 4,(IY+d)	FD CB dd E6	SET 4,(IY+d)
FD CB dd 6E	BIT 5,(IY+d)	FD CB dd EE	SET 5,(IY+d)
FD CB dd 76	BIT 6,(IY+d)	FD CB dd F6	SET 6,(IY+d)
FD CB dd 7E	BIT 7,(IY+d)	FD CB dd FE	SET 7,(IY+d)

VIDEOTON

**ELEKTRONIKAI VÁLLALAT
SZÁMÍTÁSTECHNIKAI GYÁRA**