BOÉR LÁSZLÓ · DÓRA GYULA · FENYŐ LÁSZLÓ · SERES ATTILA

AZ IBM PC-K BELSŐ FELÉPÍTÉSE

LSI ALKALMAZÁSTECHNIKAI
TANÁCSADÓ SZOLGÁLAT

AZ LSI ATSZ KÖNYVAJÁNLATA

Bartha Attila: **A NORTON...**

„PC-s körökben" – és főleg rendszerprogramozók között – a Norton név fogalom. E könyvben a már hazánkban is elterjedt és közkedvelt Norton rendszer használatát ismertetjük. Részletesen kitérünk a Norton **Commander**, a Norton **Editor**, a Norton **Guide**, a Norton **Utilities** kezelésére, valamint a Norton **Integrator** alprogramjainak önálló illetve batch-file-ban történő alkalmazására.
A Norton Commander olyan rezidens program, amely kényelmesebbé teszi a DOS használatát. Ezen kívül lehetőséget nyújt saját menürendszerek kialakítására, lehetővé teszi a file-ok paraméter szerinti keresését, a tartalomjegyzékek struktúrájának megjelenítését...
A Norton Editor rendkívül intelligens szövegszerkesztő program: alkalmas több file egyidejű feldolgozására, a szöveg formázott kinyomtatására.
A Norton Guide – ha nem is hagyományos – adatbáziskezelő program. Segítségével saját „HELP" menüt hozhatunk létre, melyben a definiált címszavakkal az általunk meghatározott információk érhetők el. Mivel ez is rezidens program, a funkciói bármikor – más program futása közben is – elérhetők.
A Norton Utilities segítségével a lemezek „lelki világában" mélyedhetünk el. Nagyon hasznos funkciója a törölt file-ok visszaállítása.
A Norton Integrator alprogramjai gyakorlatilag a DOS hiányosságait pótolják.
Ajánljuk ezt a könyvet – és a Norton rendszer használatát – minden kezdő IBM PC alkalmazónak, továbbá minden olyan haladó felhasználónak, akinek figyelmét eddig elkerülte.
Ára: 150.-Ft 96 oldal

LEVÉLCÍM: LSI ATSZ
1300 BUDAPEST, Pf. 114

Orosz Judit: **IBM PC-HÁLÓZATOK (LAN rendszerek: Nowell NetWare, PC Network, 3Com3+)**

A kiadvány célja, hogy az **IBM** személyi számítógépeken alkalmazott lokális hálózatok alapjaival ismertessen meg. Az első három fejezet a kezdőknek szól, betekintést nyújt a hálózatok alapvető tulajdonságaiba. Kitér felépítési szabványaikra, piaci viszonyaikra, valamint a hálózatok alapját képező operációs rendszerrel, az **MS-DOS**-szal való kapcsolataikra. A könyv második fele a haladók ismereteit foglalja kiválóan felépített rendszerbe, erős elméleti alapot szolgáltatva ezáltal. Foglalkozik hálózatok egymáshoz kapcsolásával; **NETBIOS**-szal. Az alapok ismertetését a különböző hálózati implementációk leírásával teszi teljessé ezt a precízen felépített munkát a Szerző.
Mindezek figyelembevételével ajánljuk ezt a kiadványt a számítógépes hálózatok iránt érdeklődő kezdő és haladó szakemberek figyelmébe.
Ára: 123.-Ft 52 oldal

Dr.Úry László: **SYMPHONY**

A sorozat célja a SYMPHONY programrendszer használatának ismertetése.
Az első rész a SYMPHONY installálásával, az elektronikus lappal, valamint a SYMPHONY makroival és programozásával foglalkozik.
A második kötet a SYMPHONY kezelésével kapcsolatos alaptechnikák legfontosabb eseteit tárgyalja, különös tekintettel az ablaktechnikára. Számos, sokszor egyedi, de nagyon találó mintapéldával szemlélteti a könyv a legcélszerűbbnek tűnő megoldásokat. A könyv oda-vissza hivatkozásokat tartalmaz az első kötettel, így használatuk célszerűen csak együtt lehetséges.
I.kötet ára: 160.-Ft 132 oldal
II.kötet ára: 133.-Ft 90 oldal
A további kötetek előkészületben

BOÉR LÁSZLÓ · DÓRA GYULA · FENYŐ LÁSZLÓ · SERES ATTILA

AZ IBM PC-K BELSŐ FELÉPÍTÉSE

Lektorálta: Serf Egyed

LSI

ALKALMAZÁSTECHNIKAI TANÁCSADÓ SZOLGÁLAT

BUDAPEST, 1989

Szerkesztette: Dóra Gyula

Készítették: Bóér László: 1. rész
Dóra Gyula: 2. rész, 4. rész (4.8 – 4.11 fejezetek), 5. rész (5.4 fejezet), 8. rész
Fenyő László: 5. rész (5.1 – 5.3 fejezetek), 7. rész
Pajor Gábor: Függelék
Seres Attila: 3. rész, 4. rész (4.1 – 4.7 fejezetek), 6. rész

Lektorálta: Serf Egyed

Technikai szerkesztő: Rucz Lajos

Munkatársak: Bajczár Istvánné
Koltai Ferencné

Kiadó: LSI Alkalmazástechnikai Tanácsadó Szolgálat
Felelős kiadó: dr. Kovács Magda
Témafelelős: Székely László
Szedés: LSI ATSz Info Osztály
Engedélyszám: 16276
ISBN 963 592 903 X
Készült: ERFATERV Ny. • Budapest VII., Csengery u. 11.
Tsz.: 89.115 • Példányszám: 4000 • Terjedelem: 23,5 (A/5 ív)
Felelős vezető: Bedő Tibor

TARTALOMJEGYZÉK

Táblázatok jegyzéke 6
Ábrák jegyzéke 8

Előszó 11

1. Rész -- Processzorok 13

1.1 CPU-k 15
1.2 Aritmetikai társprocesszorok 28

2. Rész -- Az IBM/XT és az IBM/AT alapkiépítése 63

2.1 Bevezetés 64
2.2 Az XT alapkártya (System board) 69
2.2.1 Az alapkártya leírása 69
2.2.2 Blokkdiagram 71
2.2.3 Az alapkártyára épített memóriák 87
2.2.4 I/O eszközök csatolása az alapkártyán 92
2.2.5 A 8259A interrupt vezérlő, az XT interrupt rendszere, hardware interrupt lista; software interruptok és az interrupt vektorok elhelyezése 96
2.2.6 A 8255 programozható periféria interface és a rendszer konfigurációt beállító DIP kapcsolók 101
2.2.7 A 8253 típusú számláló/időzítő és a hangszóró interface 107
2.2.8 A 8237A DMA (Direct Memory Access) vezérlő, és az IBM XT DMA csatornái 110
2.2.9 Az IBM XT I/O csatornája 114
2.2.10 Az alapkártyán található egyéb csatlakozók és beállítási lehetőségek, az XT TURBO alaplemez 119
2.3 Az AT alapkártya ismertetése 120
2.3.1 Az AT alaplemez leírása 122
2.3.2 Az AT alapkártya blokkdiagramja 125
2.3.3 RAM és ROM memória 134
2.3.3.1 Memória kiterjesztés és virtuális memória 135
2.3.3.2 A bővített memória (Expanded Memory) 138
2.3.3.3 A ROM memóriák 141
2.3.3.4 A RAM memóriák 142
2.3.4 Az AT interrupt rendszere, hardware interruptok, interrupt vektorok listája 145
2.3.5 A 8254-2 típusú számláló/időzítő és a hangszóró meghajtó áramkörök. Programozható parallel interface (PPI) 150
2.3.6 Az IBM AT DMA csatornái 152
2.3.7 Interface a billentyűzet részére 158
2.3.8 A teleppel ellátott MC146818 típusú óraáramkörök és CMOS RAM memóriák 165

2.3.9 Az IBM AT I/O címtérképe 173
2.3.10 Az IBM AT I/O csatornája 175
2.3.11 Az IBM AT alapkártya csatlakozói 185

3. Rész -- A tápegység 189

3.1 Bemeneti specifikáció 190
3.2 Kimenetek 190
3.2.1 Kimeneti védelem 191
3.2.2 Be- és kikapcsolási sorrend 191
3.3 A 'Power-Good' jel 191
3.4 A kimeneti csatlakozók jelkiosztása, terhelhetőségek 192
3.5 A tápegység elektromos kapcsolási vázlata 193

4. Rész -- A billentyűzet és az egér (mouse) 197

4.1 Bekapcsolás utáni funkciók 198
4.2 AT rendszerparancsok (84 billentyűs kivitel) 199
4.3 AT billentyűzet parancsok (84 billentyűs kivitel) 201
4.4 Adatátvitel 202
4.4.1 Az AT billentyűzet adatátvitele 202
4.4.2 Az AT billentyűzet adat küldése (output) 202
4.4.3 Az AT billentyűzet adat vétele (input) 203
4.4.4 Az XT billentyűzet adatátvitele (output) 203
4.5 A billentyűzet csatlakozója 203
4.6 IBM billentyűzet kiosztás (84 billentyűs kivitel) 205
4.7 Scan kódok (84 billentyűs kivitel) 205
4.8 IBM 101/102 nyomógombos billentyűzet 207
4.8.1 AT rendszerparancsok (101/102-es kivitel) 207
4.8.2 AT billentyűzet parancsok (101/102-es kivitel) 211
4.9 IBM 101/102 nyomógombos billentyűzet scan kódjai 212
4.10 A billentyűzet elektromos kapcsolási vázlata 224
4.11 Az egér (mouse) 227

5. Rész -- Monitorok és monitor adapterek 233

5.1 Az MC6845 CRT vezérlő 234
5.1.1 CRT vezérlő regiszterkészlet 235
5.1.2 Processzor - CRTC interface 243
5.2 IBM monochrome display és printer adapter 245
5.3 IBM színes grafikus monitor adapter (CGA) 247
5.3.1 Alfanumerikus mód 249
5.3.2 Grafikus módok 250
5.3.2.1 Közepes felbontású színes grafikus mód 250
5.3.2.2 Nagyfelbontású fekete-fehér grafikus mód 252
5.3.3 Az adapter programozása 252
5.3.4 Az adapter csatlakozói 257
5.4 Nagyobb felbontású grafikus monitor adapterek és monitorok 258

6. Rész -- MULTI I/O kártya 265
6.1 A MULTI I/O kártya rövid leírása 268
6.2 A MULTI I/O kártya alkalmazása 269
6.2.1 Mágneslemez meghajtó 269
6.2.2 Párhuzamos nyomtató illesztő 270
6.3 RS-232 soros illesztő csatlakozás 271
6.4 Teleppel ellátott óra/naptár 273
6.5 Játék vezérlő adapter 274
6.6 PC TIMER 1.3 ÓRA/NAPTÁR alkalmazása 274
6.6.1 A TIMER.COM program funkciói 274
6.6.2 Indítás 274
6.6.3 TIMER utasítások 276

7. Rész -- Floppy-k és Winchester-ek 279
7.1 Merevlemezes tárolók 280
7.2 IBM merevlemezes meghajtó adapter 285
7.2.1 A lemezvezérlő programozása 285
7.2.1.1 Státuszregiszter 285
7.2.1.2 Hibablokk 286
7.2.1.3 Adatregiszter 288
7.2.1.4 Parancstáblázat 288
7.3 IBM PC/AT Winchester és floppy-disk adapter 292

8. Rész -- Tervezési és kivitelezési szempontok saját fejlesztésű adapterkártyák esetén 297
8.1 Az áramköri tervezés 298
8.2 Az adapterkártya elkészítése, szerelése 304
8.3 A kártya bemérése, a hibák kijavítása 306
8.4 Az IBM prototípus kártya 309
8.5 Az I/O csatorna kiterjesztése, az IBM XT I/O buszkiterjesztő egység 313
8.5.1 Az I/O kiterjesztő kártya (extender card) 315
8.5.2 A vevő kártya (receiver card) 318

Irodalomjegyzék 321

Függelék 323
A1.) 8086-8088-80286 utasításkészlet ABC sorrendben 324
A2.) 8087-80287 utasításkészlet ABC sorrendben (A 8086-8088-80286 utasításkészlet kiegészítése) 349
B1.) IBM PC karakterkészlet 358
B2.) IBM PC karakterkészlet decimális és hexadecimális kódokkal 360
B3.) ASCII vezérlő karakterek 362
B4.) Táblázat rajzoló karakterek 363

Tárgymutató 365

TÁBLÁZATOK JEGYZÉKE

1. Rész -- Processzorok

1-1 A regiszterek kódolása 21
1-2 Operandus-címzési módok 21
1-3 A bázis- és indexregiszterek kijelölése az *r/m* mező segítségével 22
1-4 Eltolás, amelyet a *mod* mező definiál 22
1-5 A lebegőpontos számok decimális ábrázolási tartománya és pontossága 29
1-6 A gyorsabb *integer* aritmetikájú processzorok 36
1-7 80287 adatmozgató utasítások 42
1-8 Alap aritmetikai utasítások 47
1-9 Egyéb aritmetikai utasítások 47
1-10 A 80287-es közelítő sebesség jellemzői 5,3 MHz órajel frekvencia esetén 48
1-11 Komparáló utasítások 49
1-12 Állapotkódok komparálás után 49
1-13 Állapotkódok FTST utasítás után 50
1-14 Transzcendentális utasítások 52
1-15 A trigonometrikus függvényeket megvalósító egyenletek 52
1-16 Az inverz trigonometrikus függvényeket megvalósító egyenletek 53
1-17 A logaritmikus függvények egyenletei 53
1-18 Az inverz hiperbolikus függvények egyenletei 54
1-19 A hiperbolikus függvények egyenletei 55
1-20 Az exponenciális függvények egyenletei 55
1-21 Adminisztratív utasítások 58
1-22 A 80287-es az inicializálást követően 59

2. Rész -- Az IBM/XT és az IBM/AT alapkiépítése

2.1 Bevezetés

2-1 IBM PC rendszerkomponensek 65

2.2 Az XT alapkártya (System board)

2-2 Tápfeszültség csatlakozások 69
2-3 Az IBM XT 1 MByte-os memória címtérképe 87
2-4 Az XT I/O címtérképe 94
2-5 Hardware interrupt lista 98
2-6 Az IBM XT interrupt-vektor táblája 100
2-7 A 8255 PPI bit-térképe 105

2.3 Az AT alapkártya ismertetése

2-8 Az IBM AT 16 Mbyte-os memória térképe 135
2-9 Az IBM AT RAM memória térképe (0-512 Kbyte) 144
2-10 AT hardware interrupt lista 145
2-11 Az IBM AT interrupt-vektor táblája 146

2-12 Az IBM AT DMA vezérlőinek I/O címei 157
2-13 Az IBM AT DMA csatornáihoz rendelt I/O eszközök 158
2-14 Keyboard vezérlő I/O bitek 163
2-15 Az AT CMOS RAM címtérképe 165
2-16 'Real-time' óra információk (címek: 00-0Dh) 166
2-17 Merevlemezes diszk típusok 1-14 170
2-18 Merevlemezes diszk típusok 16-23 172
2-19 Az IBM AT I/O címtérképe 173
2-20 Az I/O csatorna jeleinek a felsorolása 178
2-21 Az IBM AT alapkártya csatlakozói 185

3. Rész -- A tápegység

3-1 A PC tápegységek kimeneti csatlakozói 192

4. Rész -- A billentyűzet és az egér (*mouse*)

4-1 IBM XT és AT scan kódok (84 nyomógombos kivitel) 206
4-2 101/102 gombos billentyűzet 1.típusú scan kódkészlete 212
4-3 101/102 gombos billentyűzet 2.típusú scan kódkészlete 217
4-4 101/102 gombos billentyűzet 3.típusú scan kódkészlete 221
4-5 Az RS-232 interface-szel rendelkező egér csatlakozó kiosztása 227
4-6 Egér BIOS szintű függvény hívások (Genius Mouse) 230

5. Rész -- Monitorok és monitor adapterek

5-1 Az MC6845 belső regiszterei 240
5-2 Színkombinációk egyszínű monitor esetén (attribútum byte) 246
5-3 Az egyszínű CRT adapter kártya I/O címtérképe 247
5-4 Színkombinációk CGA monitor adapter esetén (attribútum byte IRGB bitek) 250
5-5 A képmemória szervezése CGA 320x200 felbontás esetén 251
5-6 A CGA adapter I/O címtérképe 252
5-7 Az MC6845 CRTC regiszter értékek a CGA adapter alfanumerikus és grafikus módjai esetén 253
5-8 A CGA adapter csatlakozói 257
5-9 Képernyő megjelenítési módok 259
5-10 PC monitorok jellemzői 261
5-11 PC monitorok típusai 261
5-12 *VGA* megjelenítési módok 262
5-13 *Advanced* megjelenítési módok 263

8. Rész -- Tervezési és kivitelezési szempontok saját fejlesztésű adapterkártyák esetén

8-1 A buszkiterjesztés jeleinek késleltetései 315
8-2 Az I/O buszkiterjesztő adó-kártya címei és funkciói 317
8-3 Az I/O buszkiterjesztő vevő-kártya címei és funkciói 319

ÁBRÁK JEGYZÉKE

1. Rész -- Processzorok

1-1 A memória felépítése 15
1-2 A szavak elhelyezkedése a memóriában 16
1-3 A szavak tárolása a memóriában 16
1-4 Példa. Memória szegmentáció 17
1-5 20 bites byte- vagy szócím előállítása 17
1-6 Példa egy byte címének a kiszámítására 18
1-7 A 80286 regisztereinek csoportosítása 19
1-8 Egyoperandusos utasítás, ahol az operandus a regiszterben található 20
1-9 Utasítás, amely 1-gyel növeli a BP regiszter tartalmát 20
1-10 Egyoperandusos utasítás, ahol az operandus egyaránt lehet regiszterben, vagy memóriában 23
1-11 Utasítás, amely 1-gyel növeli a CL regiszter tartalmát 23
1-12 Példa egy memória operandusra 23
1-13 Szegmenskijelölést módosító prefix 24
1-14 Példa a szegmensmódosító prefix használatára 24
1-15 Utasítás, amely az aktuális adatszegmens 0101 1010 1111 0000 offszetű byte-ját 1-gyel megnöveli 24
1-16 Tipikus kétoperandusos utasítás 25
1-17 Példa egy kétoperandusos utasításra 25
1-18 Legegyszerűbb *immediate* operandusú utasítás 25
1-19 Példa *immediate* operandusú utasításra 26
1-20 *Immediate* operandusú utasítás, amely használja a *mod* és *r/m* mezőket 26
1-21 Példa az 1-20 ábrán látható típusú utasításra 26
1-22 *Immediate* operandusú utasítás, amely *s* mezőt is tartalmaz 26
1-23 Példa az 1-22 ábrán látható típusú utasításra 27
1-24 A 80287-es lebegőpontos számábrázolása 28
1-25 A lebegőpontos számok ábrázolási tartományai 30
1-26 A 80287-es kerekítési módjai 31
1-27 A végtelen értelmezése 34
1-28 A 80287-es egész (*integer*) formátuma 35
1-29 A 80287-es regiszterkészlete 37
1-30 A numerikus regiszterek *stack*-es címzése 38
1-31 A 80287-es *control* regisztere 38
1-32 A 80287-es *status* regisztere 39
1-33 A 80287-es *tag* (toldalék) regiszter 40
1-34 A 80287-es *exception* pointerei 41
1-35 80287-es utasítás formátumok 41
1-36 Az adatmozgató utasítások hatása 43
1-37 Különböző kivonó utasítások hatása 46
1-38 Állapotkódok FXAM utasítás után 50
1-39 A 80286-os és a 80287-es állapotkódjai közötti összefüggés 51
1-40 A 80287-es architektúrája 57
1-41 Az FSTENV és FLDENV utasításokhoz tartozó memória struktúra 60
1-42 Az FSAVE és FRSTOR utasításokhoz tartozó memória struktúra 60

2. Rész -- Az IBM/XT és az IBM/AT alapkiépítése

2.1 Bevezetés

2-1 Az IBM PC alapkártya 67
2-2 Az XT rendszer vázlatrajza 68

2.2 Az XT alapkártya (System board)

2-3 Az XT alapkártya alkatrészelrendezése 70
2-4 Az XT alapkártya blokkdiagramja 72
2-5 Az XT alapkártya kapcsolási rajzai 74
2-6 A 8088 CPU és a működését segítő áramkörök 80
2-7 A 8088 memória olvasási ciklus idődiagramja 83
2-8 A 4164 és 41256 típusú memória chip-ek kivezetései 89
2-9 Intelligens periféria IC interface 92
2-10 I/O címdekóder 93
2-11 Az Intel 8259A programozható interrupt vezérlő blokkdiagramja 96
2-12 A 8255 PPI blokkdiagramja 102
2-13 SW1 DIP kapcsoló beállítási lehetőségei 106
2-14 A 8253 számláló/időzítő blokkdiagramja 107
2-15 Hangszóró interface 109
2-16 A 8237A DMA vezérlő blokkdiagramja 111
2-17 Az IBM XT I/O csatorna csatlakozó jelei az alaplemez alkatrész oldala felől nézve 115
2-18 A keyboard és a hangszóró csatlakozók bekötése 119

2.3 Az AT alapkártya ismertetése

2-19 Az alaprendszer blokkdiagramja 123
2-20 Az AT alapkártya alkatrész elrendezési rajza 124
2-21 Az IBM AT alapkártya blokkdiagramja 126
2-22 Az AT alapkártya adatbusz kialakítása 130
2-23 Adatátviteli utak 132
2-24 A virtuális memória 137
2-25 A bővített memória 140
2-26 Osztott-interrupt kezelő áramkör. Logikai diagram 149
2-27 A számláló/időzítő és a hangszóró interface blokkvázlata 151
2-28 Programozható parallel interface (PPI) 152
2-29 (I.) DMA vezérlő áramkörök 153
2-29 (II.) DMA lapregiszter 154
2-30 A 8042 keyboard-vezérlő interface blokkdiagramja 159
2-31 Az I/O csatorna elhelyezkedése az alaplemezen (AT) 176
2-32 Az I/O csatorna csatlakozó pontjainak számozása (J1-J8 csatlakozók) 177
2-33 Az I/O csatorna csatlakozó pontjainak számozása (J10-J14 és J16 csatlakozók) 177

3. Rész -- A tápegység

3-1 A PC-k tápegységének kapcsolási vázlata 194

4. Rész -- A billentyűzet és az egér (*mouse*)

4-1 U.S.English billentyűzet kiosztás és billentyűzet sorszámok (84 nyomógombos kivitel) 204
4-2 101 nyomógombos U.S.English billentyűzet kiosztás és billentyűzet sorszámok 208

4-3 102 nyomógombos U.K.English billentyűzet kiosztás és billentyűzet sorszámok 209
4-4 Az IBM 101/102 nyomógombos billentyűzet kapcsolási rajza 225

5. Rész -- Monitorok és monitor adapterek

5-1 Letapogatási módok 234
5-2 Időzítések 235
5-3 Karakter megjelenítés 236
5-4 Vízszintes időzítések 237
5-5 Letapogatási módok 239
5-6 Kurzorszimbólum elhelyezése 239
5-7 Az MC6845 lábkiosztása 243
5-8 Egyszínű adapter kártya blokkdiagramja 245
5-9 Karakterkód és attribútuma 246
5-10 Színes grafikus adapter blokkdiagramja 248
5-11 Attribútum byte (CGA adapter) 249
5-12 Képmemória-bitek és képpontok megfeleltetése CGA 640x200 felbontás esetén 252
5-13 A CGA adapter színkiválasztó regisztere 255
5-14 A CGA adapter üzemmód kiválasztó regisztere 255
5-15 A CGA adapter státusz regisztere 256

6. Rész -- MULTI I/O kártya

6-1 A MULTI I/O kártya elrendezési rajza 266
6-2 A kártya behelyezési vázlata 267
6-3 A kártya rögzítése 267
6-4 A mágneslemez meghajtók csatlakoztatása 269
6-5 A nyomtató megszakítási szintjének kiválasztása 270
6-6 A nyomtató csatlakozó jelkivezetései 271
6-7 A kommunikációs csatorna megszakítási szintjének kiválasztása 272
6-8 Az RS-232 csatlakozó kivezetései 272
6-9 Az ÓRA megszakítási szintjei 273
6-10 A botkormány csatlakozó érintkezőinek kiosztása 274

7. Rész -- Floppy-k és Winchester-ek

7-1 A merevlemezes tárolók felépítése 281
7-2 Az IBM merevlemezes meghajtó blokkdiagramja 285

8. Rész -- Tervezési és kivitelezési szempontok saját fejlesztésű adapter-kártyák esetén

8-1 A prototípus kártya blokkdiagramja 309
8-2 A prototípus kártya kapcsolási rajza 310
8-3 A prototípus kártya alkatrész beültetési rajza 310
8-4 Hátoldali csatlakozó felszerelése a prototípus kártyára 311
8-5 Prototípus kártya (alkatrész oldal) 312
8-6 Protípus kártya (forrasztási oldal) 312
8-7 A buszkiterjesztő egység vázlata 314
8-8 Az I/O extender kártya mikrokapcsoló beállításai 316
8-9 Az I/O buszkiterjesztő adó-kártya blokkdiagramja 318
8-10 Az I/O buszkiterjesztő vevő-kártya blokkdiagramja 319
8-11 Az I/O buszkiterjesztő adó- és vevő-kártyák hátoldali csatlakozóinak jelkiosztása 320

ELŐSZÓ

Ez a könyv azoknak a szakembereknek készült, akik szeretnének részletesebben megismerkedni az IBM PC-k belső felépítésével, működésével. A PC elnevezés valamennyi típus átfogó jelölésére szolgál, tehát ide tartozik pl. az XT, AT, PCjr, az eredeti PC, az XT286 és valamennyi, ezekkel kompatibilis gép. A könyv a két legelterjedtebb típussal, az érdeklődésre leginkább számot tartó XT-vel és AT-vel foglalkozik elsősorban, de utalások találhatók más típusokra is. Igyekeztünk mindvégig összehasonlításokat tenni az XT és az AT között, megismertetni a különbségeket, és azokat a plusz lehetőségeket, amelyeket az AT gépek nyújtanak.

Természetes, hogy bárki gond nélkül használhatja a számítógépét anélkül, hogy bármit tudna a belső felépítéséről, a *hardware*-ről, az egységek egymás közötti kommunikációjáról. Azonban a gép részletesebb ismerete sokat segíthet abban, hogy a felhasználó jobban kihasználja a PC-k belső lehetőségeit (pl. DMA vezérlés, számlálók/időzítők használata, interrupt kezelés, stb.), biztosan tudja kiválasztani azokat a *hardware* elemeket és *software*-eket, melyek a gépéhez a legoptimálisabbak, és ami a legfontosabb, váratlan helyzetekben, meghibásodásoknál (amelyek még a legüzembiztosabb berendezéseknél is előfordulhatnak) ezek az ismeretek sokat segíthetnek a problémák leküzdésénél.

A PC alkalmazók között egyre nő azoknak a száma, akik a gépeket valamilyen nagyobb rendszerhez illesztik, vagy a gép bővítésével alakítanak ki mérésadatgyűjtő rendszereket, ipari szabályozó berendezéseket. Az ilyen jellegű feladatok rendszerint az XT vagy AT I/O csatornájára csatlakoztatható új modulok, adapter kártyák kifejlesztését is igénylik. A könyv részletesen ismerteti az I/O csatorna felépítését, jelforgalmát és azokat az általános tervezési szempontokat, amelyeket egy I/O bővítő modul tervezésekor figyelembe kell venni.

Úgy véljük, hogy nem kevés azoknak a felhasználóknak a száma sem, akik egyszerűen kíváncsiságból, vagy azért gyűjtenek hardware ismereteket, hogy jobban megismerjék azt az eszközt, amivel dolgoznak.

A PC-k iránt érdeklődők között sok olyan van, aki nem számítástechnikai szakember, de rendelkezik elektromos ismeretekkel. Ezért talán nem haszontalan néhány gyakori fogalom aprólékosabb leírása, mint pl. a processzor címzési módok, adattípusok, hardware/software interruptok, intelligens periféria vezérlők, közvetlen memória hozzáférés (DMA), bővített és kiterjesztett memóriák (*Expanded, Extended Memory*), stb.

A működés áttekintésekor felsorolásra kerülnek *Technical Reference* szintű táblázatok, diagramok, amelyeknek a tapasztaltabb szakemberek vehetik hasznát, olyanok akik a PC-k BIOS (*Basic Input Output System*) szintű lehetőségeit szeretnék kihasználni, vagy valamely egységet nem a szokásos, hanem saját igényeik szerint akarnak programozni. A gyorsabb keresést és áttekintést segíti a táblázatok és ábrák jegyzéke, valamint a tárgymutató.

Az első IBM PC-k 1981-ben jelentek meg, az Intel 8086/8088 processzorcsalád elemeire alapítva. Az azóta eltelt idő alatt számtalan XT, AT, PS, vagy I80386-os alapú gép került kibocsátásra, és az eltelt időszakot úgy lehet jellemezni, hogy egy személyi számítógép

szabvány került megvalósításra. Valamennyi gép magán viseli azokat a meghatározó jegyeket, amelyek elsősorban az Intel 8086/8088, 80286 és 80386 processzorcsalád elemeitől származnak. A fejlődés során egyre gyorsabb, nagyobb memória és háttértár kapacitású gépek jelentek meg, de mindegyikben megtalálhatók olyan közös tulajdonságok, amelyek végigkísérték az egész fejlődési folyamatot. Az AT-k megjelenése, gyorsaságukkal, 16 bites adatátviteli útjaikkal, a „virtuális védett" mód lehetőségével jelentős előrelépést jelentett.
A legjobban elterjedt PC-DOS operációs rendszer szintén jellemzője valamennyi típusnak.

A könyv első része a processzorokkal és az aritmetikai társprocesszorokkal foglalkozik. Elsősorban a felépítésüket, az utasításképzést mutatja be röviden, tekintettel arra, hogy az utasításkészletről, az assembler szintű programozásról több kiadvány is megjelent. Valamivel részletesebb az aritmetikai processzorokról szóló ismertetés, feltételezve, hogy az aritmetikai processzorokkal kapcsolatos alapszintű ismeretek kevésbé elterjedtek, mint a CPU-król szóló.
A második rész az XT és AT alapkiépítését részletezi. Áramköri szinten ismerteti az XT és AT alaplemezek felépítését, bemutatja a funkcionális egységeket, az azokkal kapcsolatos programozási tudnivalókat. Kitér az XT-k és AT-k felépítése közötti eltérésekre. Itt kerül leírásra az XT és az AT I/O csatornája, a csatlakozók jeleinek a felsorolása és definíciója. A második rész elsősorban az alaplemezt tárgyalja, az I/O csatlakozókba dugaszolható adapter modulok az ezt követő részekben kerülnek ismertetésre.
A harmadik rész a szabványos, kapcsolóüzemű tápegységeket ismerteti és bemutat egy tipikusnak tekinthető tápegység kapcsolási rajzot.
A negyedik rész a 84 és 101/102 nyomógombos billentyűzettel és az egérrel (*mouse*) foglalkozik. Megismerhetők a rendszer és keyboard parancsok, a billentyűzet kiosztása és a *scan* kódok, továbbá a billentyűzetet kiszolgáló elektromos kapcsolás vázlata.
Az ötödik, hatodik és hetedik részek az I/O csatornára csatlakozó adapter kártyák közül mutatnak be néhány fontosabbat. Az ötödik részben a monitorok és a monitorokat vezérlő különféle típusú adapter kártyák kerülnek ismertetésre (CGA, MDA, HERC, EGA). A felhasználó útmutatót kap az MC6845 típusú monitor vezérlő IC regisztereinek a programozásához, az alfanumerikus és grafikus képernyő-megjelenítések megértéséhez.
A hatodik rész az RS-232 interface-t, floppy vezérlőt, nyomtató vezérlőt, telepes óra/naptárt és a *joy-stick* adaptert tartalmazó MULTI I/O kártyát mutatja be. Ismerteti az óra/naptár beállítását és lekérdezését, a kártyán választható beállítási lehetőségeket, és a csatlakozó kiosztásokat.
A hetedik rész a merevlemezes tárolókkal és a floppy vezérlőkkel foglalkozik. Felsorol néhány, a PC-kben használatos *Winchester* típust, ismerteti az XT merevlemezes tároló vezérlő kártyáját és az AT *floppy* diszk - *Winchester* diszk vezérlő kártyáját.
Végül a nyolcadik, utolsó rész a saját fejlesztésű adapterkártyák tervezési és kivitelezési szempontjait tárgyalja. Bemutat néhány, dokumentáció készítést és NYÁK-tervezést segítő PC programot, továbbá ismerteti az XT-hez csatolható I/O buszkiterjesztő egységet.

A Kiadó

1. RÉSZ

Processzorok.

1.1 CPU-k

1.2 Aritmetikai társprocesszorok

A mikroprocesszor a PC-k kulcseleme. Ha valaki szeretné megérteni a gép működését, alapvetően fontos, hogy ismerje a beépített processzor jellemzőit, a processzor által nyújtott lehetőségeket.

Az IBM PC-k a 8086-8088-80286 processzor család elemeire épülnek. Bár több kiadványban és katalógusban is megtalálhatjuk a processzorra vonatkozó adatokat, akinek kevés az ismerete ebben a témakörben, annak segítségére lehet ez a rész. Itt elsősorban a 8086-8088-80286 processzorok és az őket kiegészítő 8087-80287 aritmetikai társ-processzorok vagy másnéven NPU-k (*Numeric Processing Unit*) belső architektúrája, címzési és utasításképzési megoldásai kerülnek ismertetésre, és nem maguk az utasítások (ez alól kivételek az aritmetikai processzorok, ahol az utasítástípusok is részletezve vannak). A 8086-8088-80286 utasításkészletet ABC sorrendben az A.) függelék tartalmazza.

A 80286 processzor felülről kompatibilis a 8086-8088 típussal. Ezért az ismertetés a 80286-os processzort veszi alapul, de az itt leírtak „valós" módban a 8086-8088 elemekre is vonatkoznak.
A CPU utasításképzési típusaira minden esetben egy-egy példa kerül bemutatásra.
A CPU-k belső struktúrájának az ismerete segíthet az XT/AT alaplemez felépítésének a megértésében is.

Az aritmetikai processzorok a CPU-knál valamivel részletesebben kerülnek leírásra. Megismerhetők a lebegőpontos számábrázolással kapcsolatos fogalmak, az adattípusok, és a lebegőpontos műveletek során előálló problémák, azok kezelése. Ismertetésre kerülnek a végtelent kezelő eljárások, a kerekítési módok, az aritmetikai társ-processzorok belső regiszterei, az utasításformátumok és maguk az utasítások.

Az aritmetikai társ-processzorok utasításkészletében nem találhatók trigonometrikus, inverz trigonometrikus, logaritmikus, exponenciális, hiperbolikus és inverz hiperbolikus függvények. Ezeket az alap-transzcendentális utasításokból és az aritmetikai utasításkészlet más elemeiből kell megvalósítani. A fent említett hiányzó függvényeket meghatározó egyenletek ebben a részben szintén megtalálhatók. Ezeknek az egyenleteknek a figyelembevételével a hiányzó függvényeket szubrutinokkal lehet realizálni.

A CPU és aritmetikai processzorok együttműködésére, szinkronizálására vonatkozó ismeretek is megtalálhatók ebben a részben.

A 8087 és 80287 utasításkészlete minimális mértékben tér el, a leírás mind a 8087-es, mind a 80287-es aritmetikai társ-processzorokra vonatkozik.

1.1 CPU-k

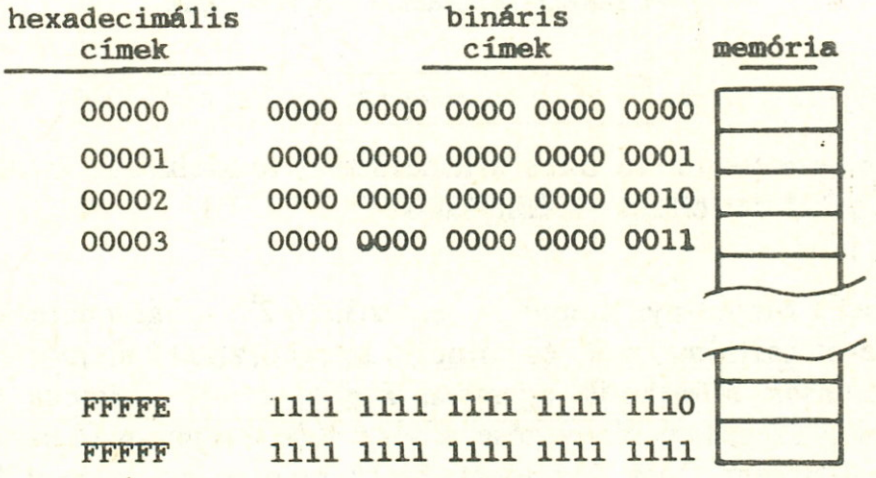
A 80286-os integrált áramkör a legtöbb olyan komponenst tartalmazza, amely elengedhetetlenül szükséges egy számítógép elkészítésekor. A számítógép összes funkcióját vezérlő áramkör, a regiszterek és a flag-ek mind erre a chip-re kerültek. Memória és I/O ugyan nincs a chip-en, de csatlakoztatásuk egyszerűen megvalósítható.

A processzornak két működési módja van: a Valós Címzési mód (Real Address Mode), és a Védett Virtuális mód (Protected Virtual Mode). Valós módban a 80286-os úgy viselkedik, mint egy gyors 8086-os, ezért a továbbiakban a 8086-osnak csak a 80286-ostól eltérő tulajdonságait fogjuk ismertetni.

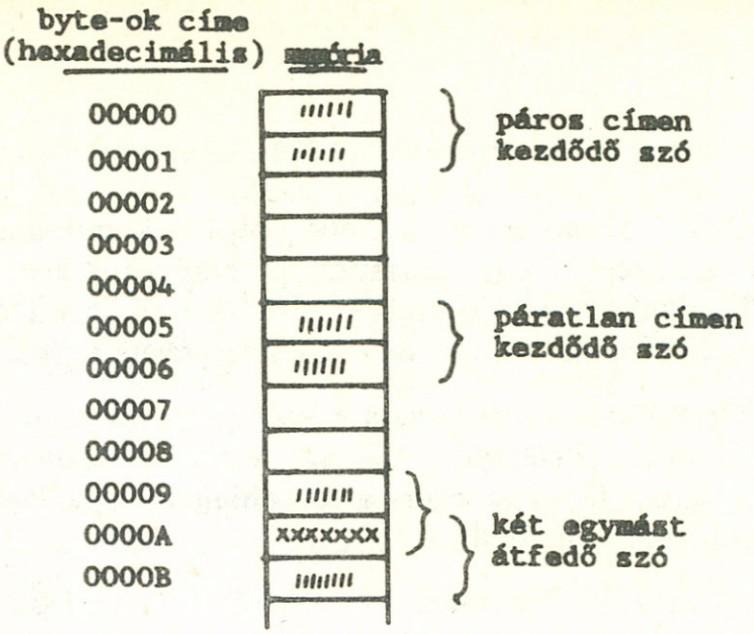
A 80286-os regiszter készlete 4 csoportra bontható. Az első ilyen csoport az általános célú regiszterek, amelyek az átmeneti adatok tárolására használatosak. A második csoportot a pointer és index regiszterek alkotják, amelyek a memória egy kijelölt részén belül adatok elérésére szolgálnak. A harmadik csoport a szegmens regisztereket tartalmazza, segítségükkel jelölhetők ki a memórián belül az egyes tartományok. A negyedik csoportot az utasításszámláló (Instruction Pointer = IP) és a 9 flag-ből álló állapotszó alkotják. A flag-ek tárolják a processzor állapotát, ill. vezérlik a működését.

Memória-szerkezet

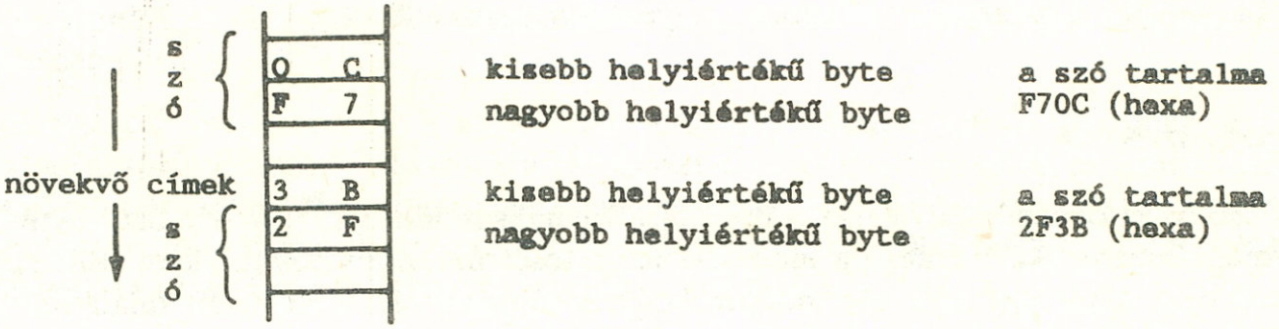
A 80286-os valós módban 2^{20} (kb 1.000.000), virtuális módban 2^{24} (16.000.000) byte-ot képes megcímezni. Az 1-1 ábra a memória felépítését mutatja valós módban. Két egymást követő byte alkot egy szót oly módon, hogy a nagyobb memóriacímen található byte a nagyobb helyiértékű. (Lásd az 1-2., és az 1-3. ábrát)



1-1 ábra.
A memória felépítése.



1-2 ábra.
A szavak elhelyezkedése a memóriában.



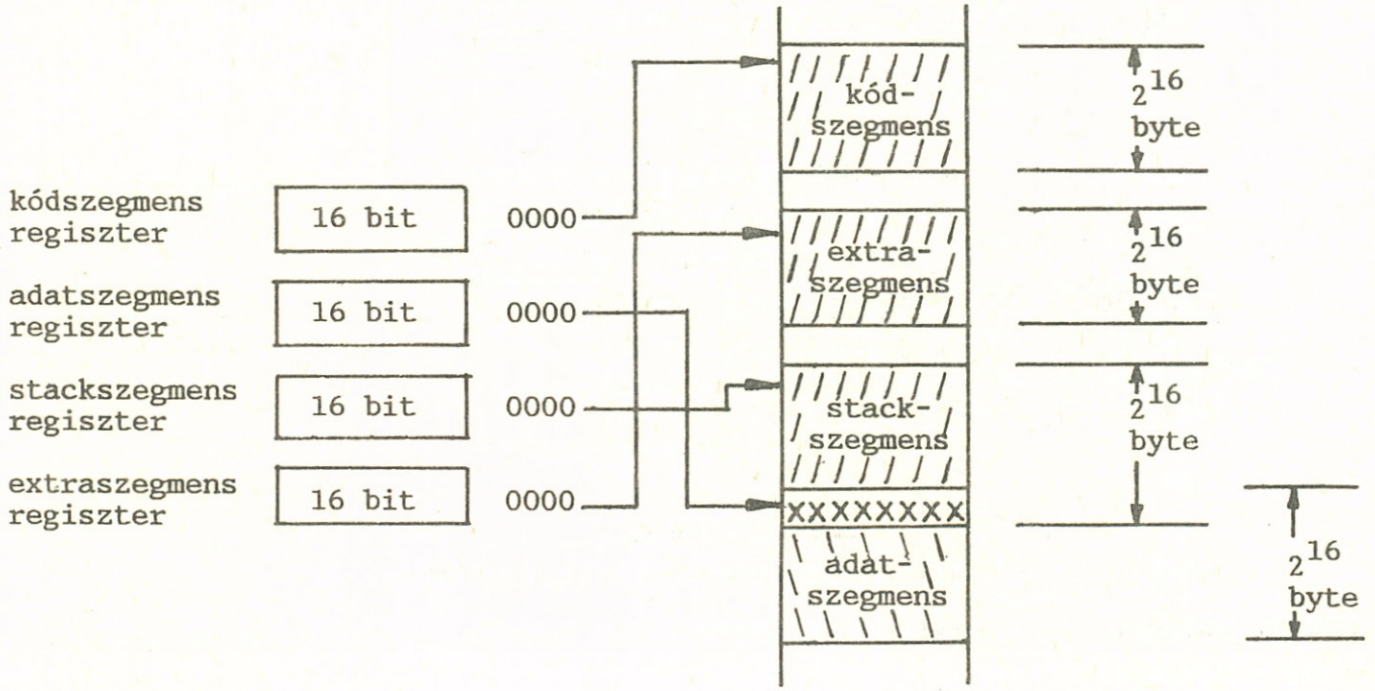
1-3 ábra.
A szavak tárolása a memóriában.

Szegmentáció

Mivel a 8086-os processzor 16 bites aritmetikával rendelkezik, szükség volt egy külön mechanizmusra a 20 bites címek előállításához.

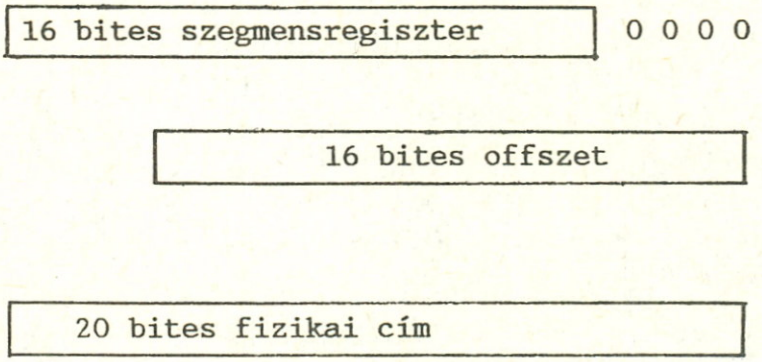
Valós módban az 1 Mbyte-nyi memória felosztható 2^{16} szegmensre, ahol az egyes szegmensek 2^{16} byte-ot tartalmaznak, és mindig 16-tal osztható memóriacímen kezdődnek. Az egyes szegmensek átfedhetik egymást. Egyszerre 4 szegmens tartalma érhető el közvetlenül. A négy szegmens elnevezése rendre: Kód szegmens (Current Code Segment), Adat szegmens (Current Data Segment), Stack szegmens (Current Stack Segment), és Extra szegmens (Current Extra Segment). (Az extra szegmens egy általános célú terület, amelyet leginkább adat-tárolóként kezelnek.)

Minden szegmenshez tartozik egy szegmensregiszter (lásd az 1-4. ábrán).

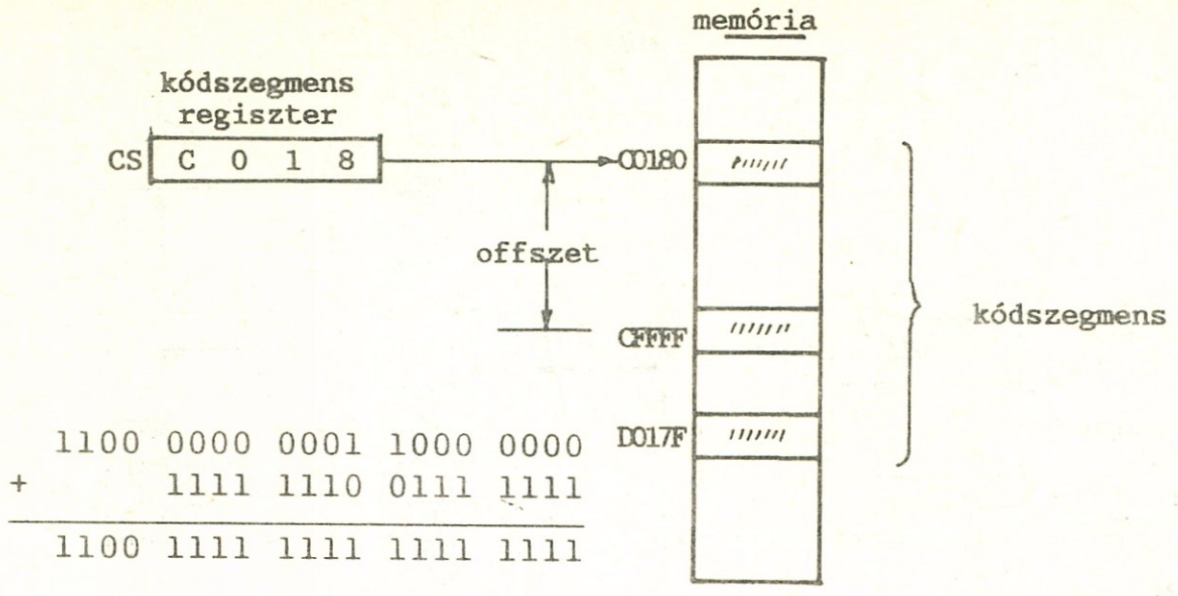


1-4 ábra.
Példa. Memória szegmentáció.

A szegmensregiszterekben található 16 bites érték adja négy 0-val kiegészítve az egyes szegmensregiszterek 20 bites kezdőcímét. A szegmensen belüli byte-okra vagy szavakra való hivatkozás egy 16 bites offszet segítségével történik. A processzor a szegmensregisztert az alsó helyiértéken 4 0-val kiegészítve hozzáadja az offszethez és így állítja elő a 20 bites címet (lásd az 1-5., és 1-6. ábrán).



1-5 ábra.
20 bites byte- vagy szócím előállítása.



1-6 ábra.
Példa egy byte címének a kiszámítására.

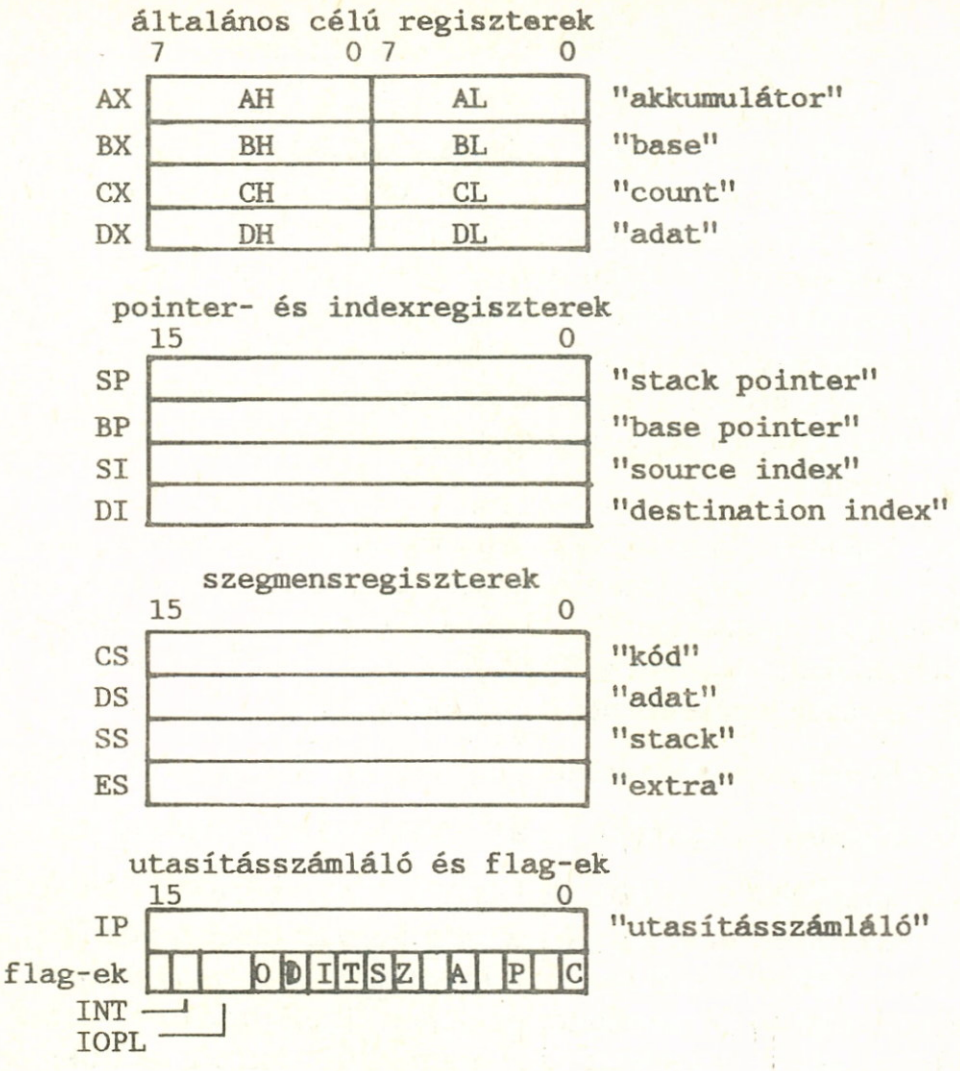
Virtuális módban, bár továbbra is vannak szegmensek és offszetek, a címszámítás másképpen történik.

I/O

A 80286 2^{16} 8 bites porttal rendelkezhet. (Ezeken a portokon keresztül képes a processzor a külvilággal érintkezni, a külső eseményeket figyelni, ill. a rájuk kiadott jelekkel ezeket az eseményeket befolyásolni.) A portok és a memória használata megegyezik, a 8 bites portok ugyanúgy címezhetők, mint a memóriában az egyes byte-ok, ill. két egymást követő címen levő 8 bites port felfogható úgy, mint egy 16 bites, hasonlóan a két byte összekapcsolásából keletkező szóhoz.

Regiszterek

A 4 általános célú regiszter az AX, BX, CX, és DX 16 bites, és mindegyike felhasználható mint két 8 bites regiszter, amelyekre külön névvel hivatkozhatunk. Az elnevezések jelzik, hogy melyik 16 bites regiszter melyik fél byte-járól van szó: (low = alsó: AL, BL, CL és DL); (high = felső: AH, BH, CH és DH). A regisztereknek ez a kettős természete teszi lehetővé mind a byte-ok, mind a szavak egyformán könnyű kezelését. Bár a regiszterek a legtöbb esetben egyenrangúak, mégis vannak bizonyos kivételek, amikor egy adott célra csak egy bizonyos regiszter használható. A regiszterek ilyen speciális felhasználására emlékeztet elnevezésük: ACCUMULATOR (AX), BASE (BX), COUNT (CX) és DATA (DX). A *pointer* és *index* regiszterek általában a szegmenseken belüli címek kiszámításához szükséges offszeteket tartalmazzák. Ezen regiszterek használatával csökken az utasítások hossza (az utasításkódba nem kerül bele a konkrét fizikai cím, csak az adott regiszter kódja), és megvalósítható a memóriacímek programfutás közbeni kiszámítása. Az ilyen számításokra leginkább a magasszintű nyelveknél, a változók offszetjének meghatározásánál van szükség.



1-7 ábra.
A 80286 regisztereinek csoportosítása.

A pointer regiszterek (SP, BP) stack szegmensen belüli adatok kényelmes elérését teszik lehetővé, míg az index regiszterek (SI, DI) általában az adatszegmensen belüli memóriarekeszekre vonatkoznak. A regiszterek speciális felhasználására utal az elnevezésük: STACK POINTER, BASE POINTER, SOURCE INDEX, DESTINATION INDEX.

A 4 *szegmens* regiszter a CS, DS, SS és ES (Code Segment, Data Segment, Stack Segment és Extra Segment) mindegyike azonosít egy külön szegmenst. Az egyes szegmensek, ill. regisztereik egymás között nem cserélhetők fel. Mint a memória szegmentálásnál említettük, a szegmensregiszterek adják (valós módban) a 20 bites cím felső 16 bitjét, amihez egy 16 bites offsetet hozzáadva kapjuk a végleges címet. Hogy az offszet hol található, az az utasítástól függ. Minden utasítás elővétel a kódszegmensből történik. Ebben az esetben az offszet, amelynek segitségével megkapjuk a következő utasítás címét, az IP (INSTRUCTION POINTER) regiszterben van. Hogy az utasítás melyik szegmensből vegye elő az operandust, azt az általában előtte álló 1 byte-os prefix határozza meg. Ennek hiányában az operandus az adatszegmensben található, kivéve, ha

a.) az offszet bármelyik pointer regiszterben van, ilyenkor a stack szegmens;
b.) az operandus egy string művelet eredménye, ekkor az extra szegmens kerül felhasználásra.

Flag-ek

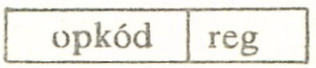
A 80286 kilenc flag-je funkciójuk szerint két csoportra osztható. Az egyik csoport az *állapot* flag-ek (status flag), amelyek általában aritmetikai vagy logikai utasítások végrehajtása után kapnak a művelet eredményének bizonyos tulajdonságaitól függő értéket. Ezt a csoportot a következő flag-ek alkotják: a Carry flag (CF), értéke 1 lesz, ha a legnagyobb helyiértékű biten átvitel keletkezett; az Auxiliary (segéd) Carry flag (AF), értéke 1 lesz, ha az alsó 4 biten átvitel keletkezett; az Owerflow flag (OF), értéke 1, ha az előjeles mennyiségnek tekintett eredmény értéke kívül esik az ábrázolási tartományon; a Zero flag (ZF), értéke 1, ha az eredmény nulla; a Sign (előjel) flag (SF), értéke 1, ha az eredmény negatív; a Parity (paritás) flag (PF), értéke 1, ha az eredmény páros számú 1-es bitet tartalmaz.

A másik csoport a *vezérlő*, vagy Control flag-ek csoportja. Ide tartozik a Direction flag (DF) amely a string manipulációs utasítások irányát vezérli; az Interrupt-enable (megszakítást engedélyező) flag (IF), amely engedélyezi, ill. letiltja a külső megszakításokat, valamint a Trap flag (TF), amelynek értékét 1-be állítva a processzor single-step, azaz lépésenkénti üzemmódba kerül.

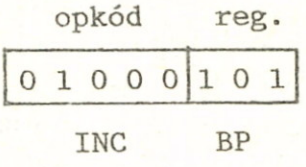
A 80286-os virtuális üzemmódban további két flag-gel (NT, IOPL) rendelkezik.

Az utasítások felépítése és az operandus címzési módok

A 80286 utasításai általában egy vagy két operanduson hajtanak végre műveleteket. Az operandus címzési mód határozza meg, hogy az utasítás végrehajtásakor honnan és hogyan veszi elő az operandusát vagy operandusait. Az egyoperandusú utasítások legegyszerűbb alakja egy byte-on ábrázolható, és akkor áll elő, mikor az utasítás egy 16 bites regiszter tartalmával végez műveletet. Az operandus címzési módoknak ezt a formáját *regiszter címzési mód*nak nevezik.



1-8 ábra.
Egyoperandusos utasítás, ahol az operandus a regiszterben található.



1-9 ábra.
Utasítás, amely 1-gyel növeli a BP regiszter tartalmát.

Az utasításbyte felső 5 bitjén az utasítás kódja, más néven az opkód helyezkedik el, a maradék 3 bit pedig a 8 lehetséges 16 bites regiszter közül jelöli ki az egyiket. A regiszterek kódolását az 1-1 táblázat mutatja be.
Ha w = 0, a 3 bit a nyolc 8 bites regiszter közül jelöli ki az egyiket.
(A w értelmezését ld. a 22. oldalon)

Kód	16 bites regiszterek	8bites regiszterek
000	AX	AL
001	CX	CL
010	DX	DL
011	BX	BL
100	SP	AH
101	BP	CH
110	SI	DH
111	DI	BH

1-1 táblázat.
A regiszterek kódolása.

KÖZVETLEN
REGISZTER
DIREKT MEMÓRIACÍMZÉS
INDIREKT MEMÓRIACÍMZÉS
- bázisregiszter
- indexregiszter
- bázisregiszter + indexregiszter
- bázisregiszter + eltolás
- indexregiszter + eltolás
- bázisregiszter + indexregiszter + eltolás

1-2 táblázat.
Operandus-címzési módok.

r/m mező	bázisregiszter	indexregiszter
000	BX	SI
001	BX	DI
010	BP	SI
011	BP	DI
100	--	SI
101	--	DI
110	BP	--
111	BX	--

mod = 00 és *r/m* = 110 esetén az 1-4 táblázat megjegyzése érvényes.

1-3 táblázat.
A bázis- és indexregiszterek kijelölése az *r/m* mező segítségével (*mod* ≠ 11)

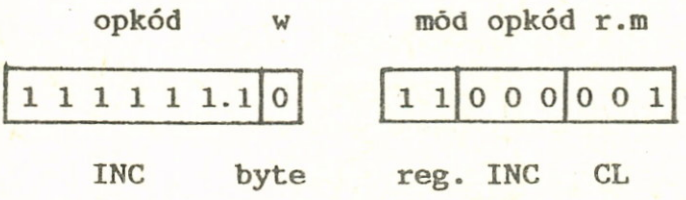
mod	eltolás	megjegyzés
00	nincs	
01	8 bites eltolás	az utasítás 1 byte-tal bővül
10	16 bites eltolás	az utasítás 2 byte-tal bővül

Megjegyzés: mod = 00 és r/m = 110 esetén az 1-3, 1-4 táblázatok nem érvényesek, az utasítás 2 byte-tal bővül, és az offszet értéket ez a két byte tartalmazza.

1-4 táblázat.
Eltolás, amelyet a *mod* mező definiál (*mod* ≠ 11).

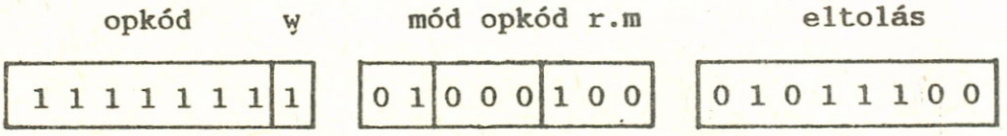
Az egyoperandusú utasítások általánosabb formája 2 byte hosszú (1-10 ábra), és operandusa lehet bármely általános célú pointer- vagy indexregiszter akár 8, akár 16 bites, ill. bármely szó vagy byte a memóriában. Az opkód mező két részre oszlik, 7 bit az utasítás első, 3 a második byte-jában helyezkedik el. Az utasítás első byte-jának legalsó bitje a W mező (word). Ha a W értéke 0, akkor az operandus 8, egyébként pedig 16 bit hosszú. Az utasítás második byte-jában található a *mod* és az *r/m* mező. A *mod* és az *r/m* együttesen határozzák meg az operandus címzési módját, amelytől függően az utasitás maximum további két byte-tal bővülhet. Ha a *mod* értéke 11, az operandus valamelyik regiszter tartalma (*regiszter cimzési mód*). A regisztert az *r/m* mező jelöli ki a 1-1 táblázatban látható módon. Ha a *mod* értéke 11-től eltérő, az operandus a memóriában van.

1-10 ábra.
Egyoperandusos utasítás, ahol az operandus egyaránt lehet regiszterben, vagy memóriában.



1-11 ábra.
Utasítás, amely 1-gyel növeli a CL regiszter tartalmát.

A címszámításnak ezt a módját *indirekt memóriacímzés*nek nevezzük. Az offszet maximum 3 szám összege. Az összetevők a következők lehetnek: a *mod* értékétől függően egy 8 vagy 16 bites eltolás (displacement), amely az utasítást követő 1 ill. 2 byte-on helyezkedik el, és az *r/m* értékétől függően valamelyik indexregiszter és/vagy valamelyik bázisregiszter tartalma. Az egyes összetevőket a *mod* az 1-4, az *r/m* az 1-3 táblázatban látható módon jelöli ki. Ha az offszet kiszámításában a BP regiszter nem vesz részt, az operandus az adat szegmensben, ellenkező esetben a stack szegmensben található.



1-12 ábra.
Példa egy memória operandusra.

Vizsgáljuk meg a 1-12 ábrán látható példát.
Legyen az SI és DI regiszterek értéke a következő:

```
SI 1010 0000 1000 0110
DS 1111 0000 1111 0000
```

Az opkód értéke 1111 1110 00, ez az INC utasítás kódja. A W értéke 1, amely jelzi, hogy az operandus 16 bit széles. A *mod* értéke 01, amely mutatja, hogy az operandus a memóriában van, és hogy az utasítást egy 1 byte széles eltolás követi. Az *r/m* értéke 100, amely jelzi, hogy az SI indexregiszter tartalmát kell hozzáadni az eltoláshoz, hogy megkapjuk az offszetet. Így az offszet értéke:

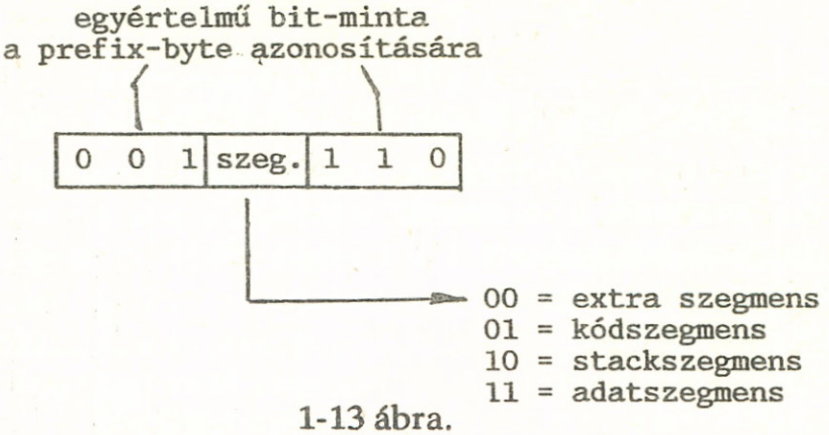
```
  1010 0000 1000 0110      az SI tartalma
+ 0000 0000 0101 1100      eltolás (előjel helyesen
  ------------------                kiterjesztve 16 bitesre)
  1010 0000 1110 0010      offszet
```

Mivel a BP regiszter nem szerepel a címszámításban, az offszet az aktuális adatszegmensre vonatkozik. Az operandus címe így a következő:

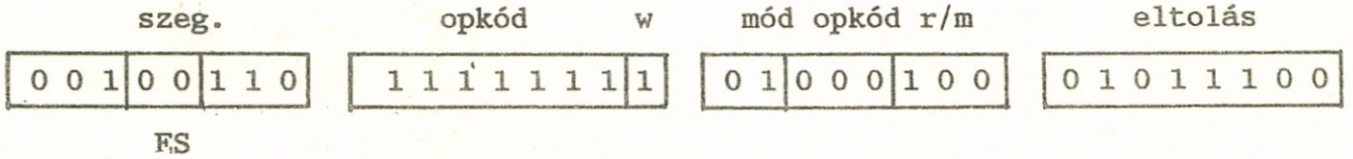
```
   1111 0000 1111 0000          a DS tartalma
+       1010 0000 1110 0010     offszet
   ------------------------
   1111 1010 1111 1110 0010     memóriacím
```

Mivel az operandus 16 bit széles (W = 1), ezért az operandus alsó (kisebb helyiértékű byte-ja a 1111 1010 1111 1110 0010 címen, a felső byte-ja az ezt követő 1111 1010 1111 1110 0011 címen található.

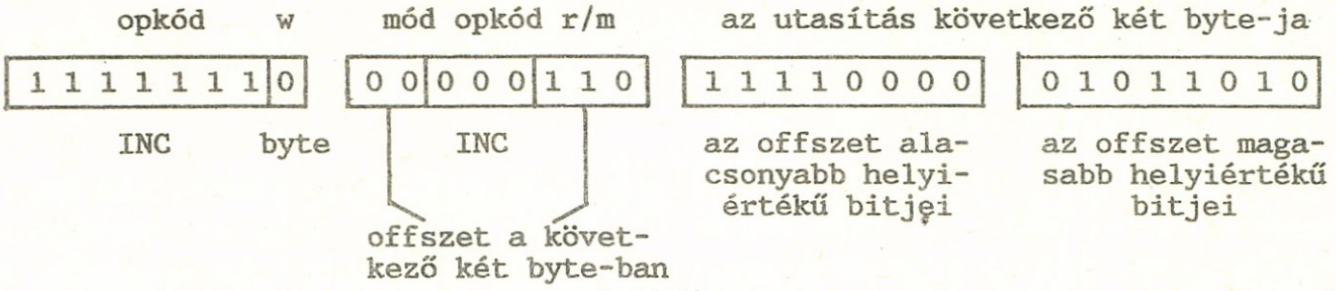
Amennyiben szükséges, az utasítást megelőző 1 byte-os prefix segítségével elérhető, hogy az adat vagy stack szegmensen kívül más szegmenst is kijelöljünk, ahonnan az operandus elővétele történik. A prefix byte értelmezése az 1-13 ábrán látható.



1-13 ábra.
Szegmenskijelölést módosító prefix.



1-14 ábra.
Példa a szegmensmódosító prefix használatára.



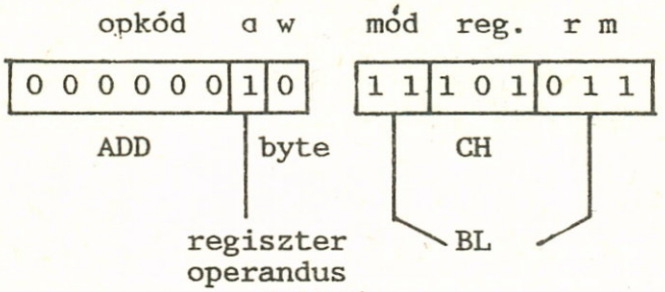
1-15 ábra.
Utasítás, amely az aktuális adatszegmens
0101 1010 1111 0000 offszetű byte-ját 1-gyel megnöveli.

Amikor az operandus ismert, fix címen helyezkedik el, az offszet közvetlenül megadható az utasítást követő két byte-on. Ezt az operandus címzési módot *közvetlen* (direkt) *memória címzés*nek nevezzük. Mivel a tervezők a *mod* és az *r/m* összes lehetséges kombinációját kihasználták az indirekt memória címzésnél, ezért egy ritkán használatos címzési mód kódját elvették a direkt memória címzés számára. Ez a kód a *mod* 00, és az

r/m 110 eredetileg a BP indirekt címzést jelentené (indexregiszter és eltolás nélkül), helyette *mod* 01, *r/m* 110-et és egy 1 byte-os 0 értékű eltolást kell használni. Az 1-15 ábrán látható utasítás az adatszegmensben a 0101 1010 1111 0000 offszeten található byte-ot inkrementálja.

A kétoperandusú utasításokat helykímélés céljából úgy oldották meg, hogy az egyik operandus mindig egy regiszter tartalma. Így az utasításban csak egy *mod* és egy *r/m* mező van az egyik, és egy *reg* mező a másik operandus számára. Ez látható az 1-16-os ábrán.

1-16 ábra.
Tipikus kétoperandusos utasítás.

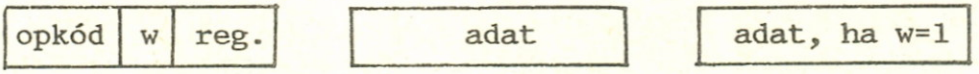


1-17 ábra.
Példa egy kétoperandusos utasításra.

A kétoperandusú utasítások (ugyanúgy mint az egyoperandusúak) a W mezőt annak eldöntésére használják, hogy az operandusok 8 (W = 0), vagy 16 bitesek-e (W = 1). Az itt bevezetett *d* (destination = cél) mező meghatározza, hogy az eredmény a *mod* és *r/m* (d = 0) vagy a *reg* (d = 1) mezőhöz tartozó operandusba kerüljön.

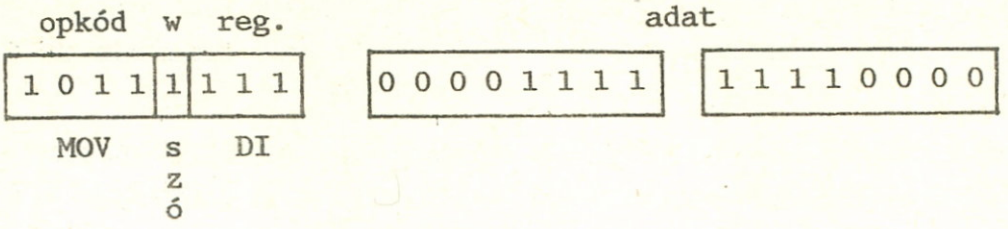
Az 1-17 ábra az ADD utasítást mutatja be. Az ADD opkódja 000000. A W mező 0, tehát mindkét operandus 8 bit széles. A *reg* mező által kijelölt operandus a CH regiszter. A *mod* értéke 11, tehát a másik operandus szintén regiszter, az *r/m* értéke alapján a BL. A *d* mező megadja, hogy az eredményt a *reg*-hez tartozó operandusba, azaz a CH regiszterbe kell tölteni.

A kétoperandusú utasítások egyik operandusa lehet konstans érték is, amit az utasítás maga tartalmaz. Az ilyen ún. *immediate címzési mód* előnye az adat gyors elérése. Az egyik ilyen utasítás a MOV (adatmozgató utasítás), amelynek egyik leggyakoribb felhasználása konstans érték betöltése egy regiszterbe (általános célú pointer, vagy indexregiszter).

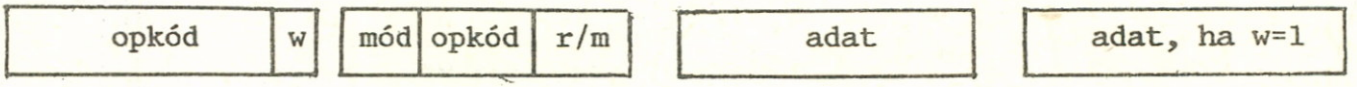


1-18 ábra.
Legegyszerűbb *immediate* operandusú utasítás.

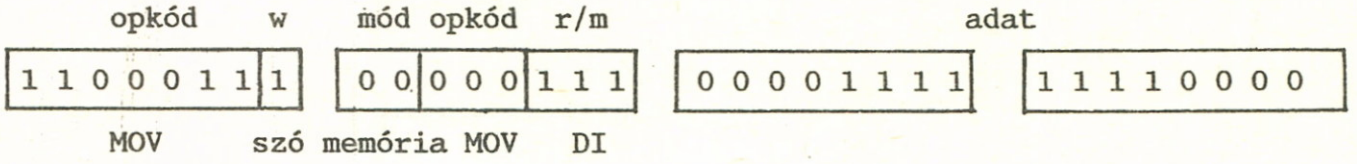
Az immediate operandus címzésnek ez a legegyszerűbb esete látható az 1-18 ábrán. A *reg* mező a céloperandusként szolgáló regisztert jelöli ki. A W mező értékétől függően az operandusok 8 vagy 16 bitesek.



1-19 ábra.
Példa *immediate* operandusú utasításra.



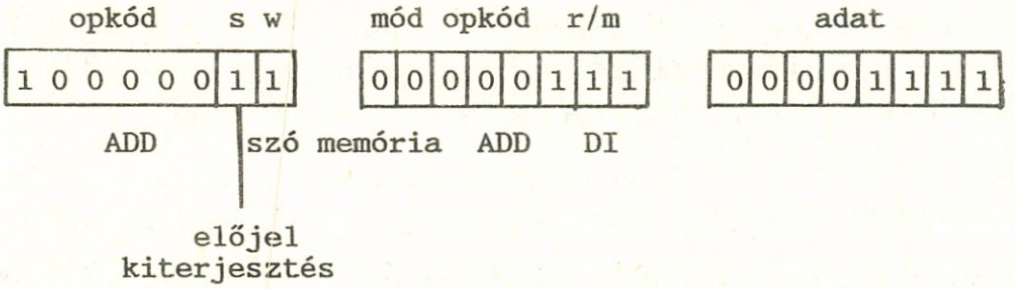
1-20 ábra.
Immediate operandusú utasítás,
amely használja a *mod* és *r/m* mezőket.



1-21 ábra.
Példa az 1-20 ábrán látható típusú utasításra.

1-22 ábra.
Immediate operandusú utasítás,
amely *s* mezőt is tartalmaz.

A közvetlen címzés kicsit bonyolultabb, általánosabb formája, amikor a céloperandust a *reg* mező helyett a *mod* és az *r/m* mező jelöli ki. Ebben az esetben a céloperandus lehet a memóriában is, de az utasítás ábrázolásához szükség volt egy további byte-ra is. Ez látható az 1-20 ábrán. Mivel az utasításoknak ebben a formájában csak egy W mező van, mindkét operandus vagy 8 vagy 16 bit széles. Azonban a konstansok általában kis számok, amelyek nem igényelnek két byte-ot. Ez különösen igaz az összeadó, kivonó, összehasonlító utasításoknál, ezért ezek az utasítások tartalmaznak egy *S* mezőt (sign extended = előjelkiterjesztett). Az S csak W = 1 esetén értelmezett, és jelzi, hogy a közvetlen operandus 16 (S = 0) vagy 8 (S = 1) bit széles. Az utasításoknak ezt a formáját mutatja be az 1-22 ábra. Az 1-23 ábra egy példát mutat az utasításoknak erre a formájára.



1-23 ábra.
Példa az 1-22 ábrán látható típusú utasításra.

Ezzel a 8088/86 és 80286 utasításképzésének az áttekintését befejeztük, és rátérünk a CPU-t támogató aritmetikai társprocesszor a 8087 és a 80287 tárgyalására.

1.2 Aritmetikai társprocesszorok

80286, hasonlóan az elődjéhez, a 8086-hoz önmagában nem képes lebegőpontos utasítások végrehajtására. Ez a hiányosság pótolható lebegőpontos programcsomag alkalmazásával, azonban a számítások sebessége bizonyos esetekben alacsony lehet.

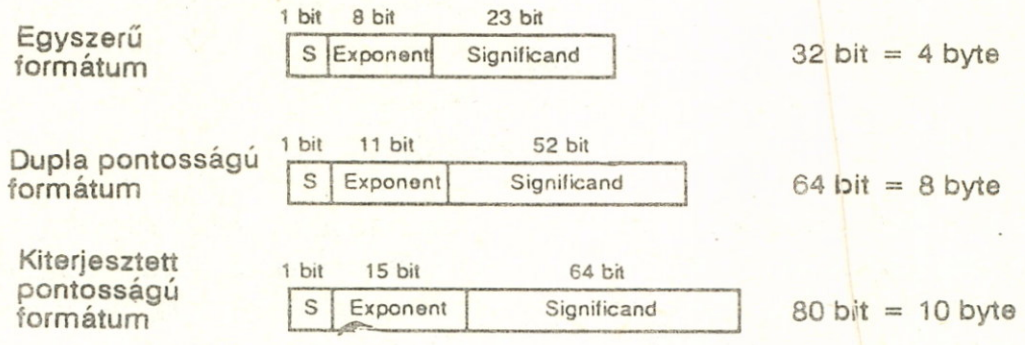
Az Intel által gyártott aritmetikai co-processzorok használata az ilyen műveletek végrehajtási sebességét 10-100-szorosára növeli a szoftver megvalósításokhoz képest.

A 8086-oshoz és 80286-oshoz külön aritmetikai processzor tartozik, a 8087 ill. a 80287. A 8087 és a 80287 között programozói szempontból csak kevés különbség van, ezért az egyszerűség kedvéért részletesen a 80287-t tárgyaljuk, a 8087-estől eltérő eseteket külön kiemelve.

Amikor a 80286-os elér egy 80287-es utasításhoz, két eset lehetséges. Ha van 80287-es a rendszerben, a 80286-os automatikusan átadja az utasítást neki, amit az azonnal végre is hajt. Ellenkező esetben a vezérlés egy exception-handler-re kerül, amely szoftveresen emulálja a 80287-est. Emiatt egy közönséges program nem képes észlelni a 80287-es jelenlétét, eltekintve attól, hogy az emuláció 10-100-szor lassabb.

Lebegőpontos számábrázolás

A 80287 háromféle módon képes a lebegőpontos számokat ábrázolni, *egyszerű* (single precision), *dupla pontosságú* (doubled precision) és *kibővített* (extended precision) formátumban. (Egyes Intel irodalmakban ennek megfelelői a 'short real', 'long real' és a 'temporary real' elnevezések.)



1-24 ábra.
A 80287-es lebegőpontos számábrázolása.

Az egyes számok - formátumuktól függetlenül - három részre, előjel, kitevő és szignifikáns mezőre bonthatóak, és - formátumuktól függően - négy, nyolc ill. tíz byte-on helyezkednek el. A memóriában lévő szám legalacsonyabb címe a szignifikáns legkisebb helyiértékű byte-jára, a legmagasabb az exponens 7 legnagyobb helyiértékű bit-jét és az 1 bit-es előjel bitet magába foglaló byte-ra mutat. Az 1-24 ábra a lebegőpontos számok három lehetséges

formátumát mutatja be.

Az *előjel mező* értéke 0, ha a szám pozitív, 1, ha a szám negatív. Mivel a számok nem kettes komplemensben vannak ábrázolva, így értelmezett mind a + mind a - nulla. A 80287-es komparaló utasításai a két nullát egyenlőként kezelik, de azoknak, akik a viszonylag lassú 80287 komparáló utasításokat 80286-os utasításokkal akarják szimulálni, ügyelniük kell erre.
A szignifikáns mező a nagyobb ábrázolási pontosság kedvéért normalizált alakban, a vezető bit elhagyásával (amely mindig 1 lenne) tartalmazza a szignifikánst, kivéve, ha a szám kibővített formátumú, mert ekkor a bevezető bit is a szignifikáns mező részét képezi.

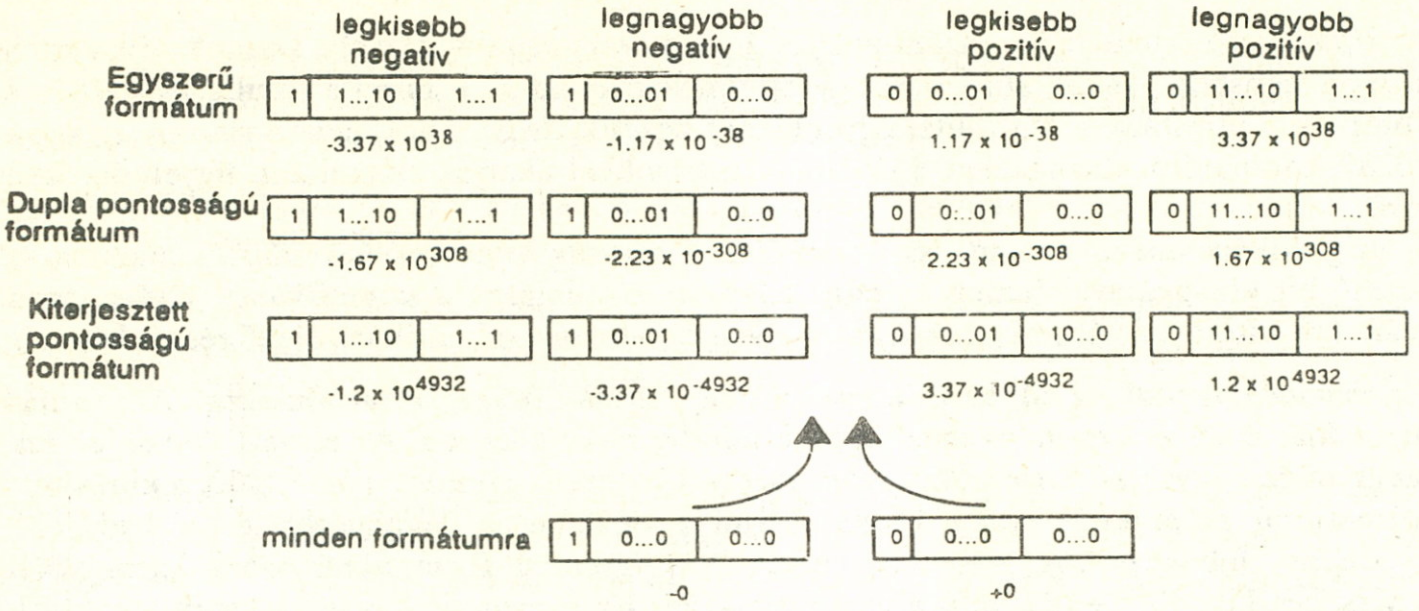
Az *exponens mező* az aktuális kitevő és egy eltolás összegét tartalmazza. Az eltolás biztosítja, hogy az exponens mező értéke mindig pozitív legyen. Az eltolás értéke, ennek megfelelően egyszerű formátumú szám esetén 127, dupla pontosságúnál 1023, a kibővített formátumú számoknál pedig 16383. Mivel a 80287-es a legnagyobb és a legkisebb exponenst hibakezelésre használja, ezért egy egyszerű formátumú szám legnagyobb lehetséges kitevője +127, míg a legkisebb -126. Ez a megszorítás vonatkozik a számok másik két formátumára is.

Formátum	**értékes számjegyek**	**a legkisebb lehetséges 10-es hatvány**	**a legnagyobb lehetséges 10-es hatvány**
Egyszerű	6	-37	38
Dupla pontosságú	15	-307	308
Kiterjesztett	19	-4931	4932

1-5 táblázat.
A lebegőpontos számok decimális ábrázolási tartománya és pontossága.

A lebegőpontos számok *ábrázolási tartományait* mutatja be az 1-25 ábra. Érdemes megfigyelni hogy a negatív számok teljesen szimmetrikusak a pozitív számokra, így az abszolút érték képzés sohasem okozhat alul- vagy túlcsordulást. (Túlcsordulásnak nevezik, ha egy művelet eredménye nagyobb, mint a legnagyobb ábrázolható pozitív szám, vagy kisebb mint a legkisebb negatív. Alulcsordulásról akkor beszélünk, ha az eredmény nulla, és a legkisebb pozitív, vagy a legnagyobb negatív szám közé esik).

Most vizsgáljuk meg, mi történik, ha a legkisebb pozitív számot összeszorozzuk a legnagyobb pozitív számmal. Az eredmény mindhárom formátum esetén kb. 4. Más szavakkal, a legnagyobb pozitív szám kb. 4-szerese a legkisebb negatív szám reciprokának, ami azt jelenti, hogy az olyan gyakran előforduló műveletek, mint 1/x 2/x 3/x ¶/x sohasem okozhatnak túlcsordulást, csak alulcsordulást.



1-25 ábra.
A lebegőpontos számok ábrázolási tartományai.

A 80287-es rendelkezik bizonyos eszközökkel az ilyen és ehhez hasonló problémák okozta hatások csillapítására.

A következő szakasz ennek a megvalósítását tárgyalja. Az itt látható felsorolás a numerikus hibák súlyosság szerint, növekvő sorrendbe szedett listája:

Pontatlan eredmény; feltétele: az eredmény nem ábrázolható pontosan az adott formátumon belül. Pl.: 1 osztva 3-mal, vagy a 9.87654321 duplapontos szám konvertálása egyszerű formátumba, ahol csak 8 értékes számjegy ábrázolható.

Alulcsordulás; feltétele: az eredmény nulla és a legkisebb pozitív, vagy a legnagyobb negatív szám közé esik. Pl.: 1 osztva a legnagyobb kibővített formátumú szammal, vagy a legkisebb, nullától különböző pozitív, kibővített formátumú szám konvertálása dupla pontosságúvá.

Denormált operandus exception keletkezik, valahányszor egy utasítás operandusa denormált szám (magyarázatát lásd később).

Nullával való osztás

Túlcsordulás; feltétele: az eredmény nagyobb, mint a legnagyobb ábrázolható pozitív szám, vagy kisebb, mint a legkisebb negatív.

Érvénytelen művelet; ez a kategória magába foglalja az összes eddig nem említett hibát. Ilyen lehet a 0 osztása 0-val, negatív szám négyzetgyöke, vagy egy nemlétező 80287-es regiszterre való hivatkozás.
Egy népszerű módszer a hibák lekezelésére a művelet(ek) azonnali felfüggesztése és átadása egy exception-lekezelő eljárásnak. Ezt a módszert *trapping*-nek hívják (trap-csapda).

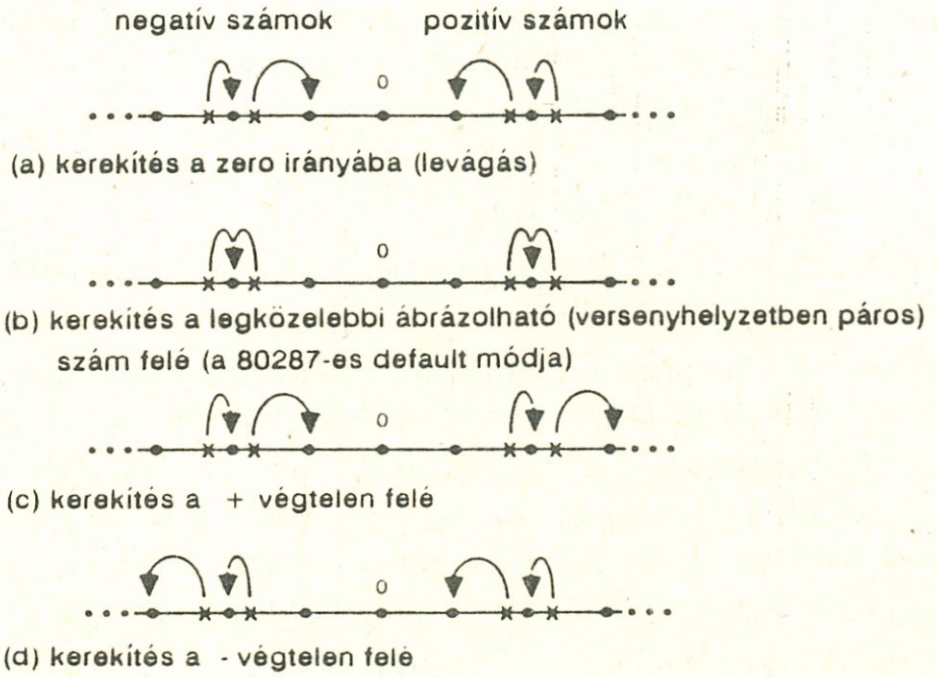
Egy másik módszer, amikor az akadályozott művelet egy speciális hibakódot vagy exception értéket ad vissza, és biztosítja a számítás végrehajtását.

A 80287-es a tiltott exponens értékeket, azaz a 000...00-t és az 111...11-t tartja fenn a hibakódok számára. A hibakódok ilyen formája lehetővé tette, hogy a komparáló műveletek akkor is helyes értéket adjanak, ha a programozó figyelmen kívül hagyja őket. Mivel a lebegőpontos műveletek általában közelítő számítások sorozatából állnak, ezért előfordulhat, hogy bár a program hibát jelez egy művelet közbenső eredményéhez, a végeredmény mégis korrekt értéket ad. Ebből a szempontból a hibakódok nem feltétlenül jelentenek hibát, ezért a hibakód helyett szerencsésebb elnevezés az *exception*-érték (exception-kivétel).

A 80287-es biztosítja az exception értékeket és az opcionális trapping-et. A trapping külön-külön minden egyes hibánál engedélyezhető vagy maszkolható.

A következő szakaszban leírjuk a különböző típusú hibákhoz tartozó exception értékeket ill. azt, hogy mi történik, ha maszkoltak.

Pontatlan eredmény; Ha az exception maszkolt, a 80287 ahelyett, hogy egy speciális exception-értékkel térne vissza, az eredményt közönséges lebegőpontos számmá kerekíti. A 80287 4 féle kerekítési módot kínál (ld. 1-26 ábra). A számegyenesen látható X-ek a valódi számot jelzik. A pontok a számhoz legközelebb álló ábrázolható számot jelképezik.



1-26 ábra.
A 80287-es kerekítési módjai.

Alulcsordulás; az exponens 000...00 értéke jelzi az exception-t.

Ha maszkolt, a 80287-es válasza az alulcsordulásra egy *denormált* szám előállítása a következőképpen:

- az egyszerű formátumú és dupla pontosságú számoknál a rejtett, vezetőbit értéke 1 helyett 0-ként
- az exponens 000...00 helyett 000...01-ként (azaz a legkisebb lehetséges exponensként) értelmezett. Az itt látható egyszerű formátumú számok erre mutatnak példát.

Bár a kiterjesztett formátumú számoknál a vezető bit is ábrázolva van, a következetesség kedvéért az exponens mező 000000000000000 és 000000000000001 értéke egyaránt -16382-nek felel meg.

Ha az alulcsordult szám olyan kicsi, hogy denormáltként sem lehet ábrázolni, 0 lesz belőle. Az alulcsordulás ilyen közelítését fokozatos alulcsordulásnak (gradual underflow) nevezik.
A másik lehetséges közelítés a *hirtelen alulcsordulás* (abruptly underflow). Ekkor, ha az alulcsordulás maszkolva van, minden alulcsordult számból 0 lesz.

Denormált operandus lekezelésének leginkább akkor van haszna, amikor a 80287 más típusú aritmetikát szimulál. Általában az alulcsordulás maszkolt, ezzel engedélyezve a denormált számok előállítását. Így valószínűleg a denormált operandus exception is maszkolt, amely az ilyen számok használatát engedélyezi.

Vizsgáljuk meg mi történik, ha a denormált operandus exception maszkolva van.

A pontosságot feláldozva a nagyobb ábrázolási tartomány kedvéért, denormált eredményt kapunk. A 80287-es jelzés nélkül próbálja meg megakadályozni a számítások maradékainak egyre romló pontosságából származó alacsony pontosságú denormáltakat. Ez a jelzés létrejöhet *exception-érték*, vagy *trap* formájában. Így a denormált operandusokat tartalmazó műveletek eredménye vagy

1. *zero*, ha az eredmény olyan kicsi, hogy denormálként sem ábrázolható (pl. két denormált szorzata), vagy
2. *denormált*, ha az eredmény olyan kicsi, hogy csak denormálként ábrázolható (pl. két denormált összege), vagy
3. *normált*, ha a denormált operandus pontossága olyan kicsi, hogy az eredményt nem befolyásolja (pl. egy nagy szám és egy denormált összege), vagy
4. *unnormált* (definícióját lsd később), ha az eredmény szabályszerűen nem ábrázolható, de túl nagy ahhoz, hogy denormált legyen, és a céloperandus kibővített formátumú (pl. egy nagy szám és egy denormált szorzata), vagy
5. *érvénytelen művelet exception*, ha az eredmény teljesíti a 4. pont feltételeit, de a céloperandus nem kibővített formátumú.

Ezenkívül, minden denormálttal való osztási kísérlet, vagy denormáltan végrehajtott négyzetgyökvonás érvénytelen művelet exception-t okoz.
Az ***unnormáltak*** olyanok, mint a közönséges, kibővített formátumú számok, azzal a

különbséggel, hogy szignifikáns mezejük vezető bitjének értéke 0 (és nem 1). Unnormáltak csak kibővített formátumban fordulhatnak elő, tekintve, hogy ezenkívül nincs olyan ábrázolási formátum, amely tartalmazná a szignifikáns mező vezető bitjét. Az unnormáltak, úgyanúgy mint a denormáltak, a fenti (1-5) szabályoknak engedelmeskednek. Az unnormáltak egy furcsa típusa, ún. pszeudo-zero keletkezik, ha két 0 szignifikáns mezejű unnormáltat összeszorzunk. A pszeudo-zero exponense nullától különböző, szignifikánsa nullát tartalmaz. Viselkedése megegyezik az unnormált számokéval, kivéve az összehasonlító műveleteket, ahol nullaként értelmezett.

A fenti 5 szabály biztosítja, hogy a denormáltakkal végzett műveletek eredményeit vagy nem befolyásolja a denormáltak pontosságának hiánya, vagy a 80287-es jelzi, hogy az eredmény nem közönséges formátumú.

Nullával való osztás: Azok a lebegőpontos számok, amelyeknek exponense 11...111 és szignifikánsa 00...000, a végtelent reprezentálják. Mivel az előjel bit, ettől függetlenül lehet akár 0, akár 1, külön értelmezett a $+\infty$ és $-\infty$. ∞ keletkezhet nullával való osztáskor, ha ez az exception maszkolt ill. túlcsorduláskor ha a túlcsordulás exception maszkolt.

Mivel a 80287-es megkülönbözteti a +0-t a -0-tól az előjeles végtelen az alábbi módon származtatható

$$x/+0 = +\infty$$
$$-x/+0 = -\infty$$
$$x/-0 = -\infty$$
$$-x/-0 = +\infty$$

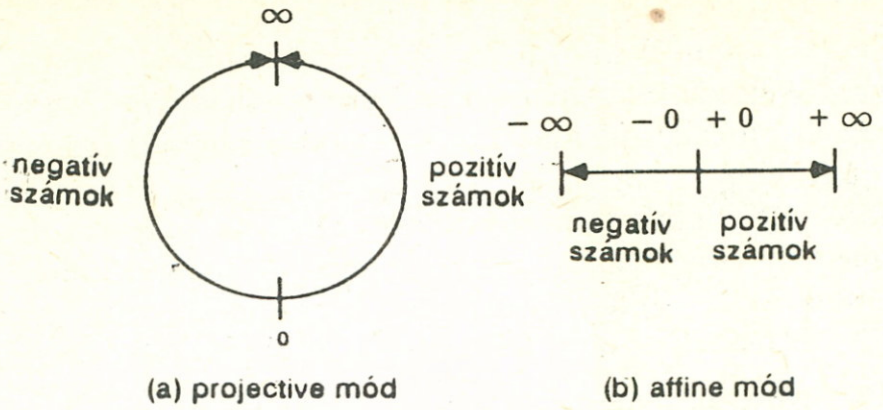
ahol x nagyobb mint 0 és nem végtelen.

A szorzások és osztások, ha egyik vagy mindkét operandusuk végtelen, a következőképpen viselkednek

$$x/+\infty = +0, \qquad -x/+\infty = -0, \text{ stb.}$$
$$x*+\infty = +\infty, \qquad -x*+\infty = -\infty, \text{ stb.}$$
$$+\infty/x = +\infty, \qquad -\infty/x = -\infty, \text{ stb.}$$
$$+\infty*+\infty = +\infty, \qquad -\infty*+\infty = -\infty, \text{ stb.}$$

ahol x ismét nagyobb mint 0 és nem végtelen. Minden olyan szorzás és osztás, amelynek egyik operandusa 0, a másik ∞ , érvénytelen művelet exception-t okoz, éppúgy mint végtelen osztása végtelennel.

Az összeadás, kivonás és komparálás esete némileg bonyolultabb. A 80287-es a végtelen kezelésére kétféle megközelítést kínál, névszerint a *projective* és az *affine* módot. A konzervatívabb projectíve mód a 80287-es default módja. A projectíve mód nem tesz különbséget végtelen és végtelen között, éppúgy, ahogy a 80287-es közönséges esetben elrejti azt a tényt, hogy van +0 ill. -0 is (ld. 1-27/a ábra). Az affine mód felismeri a + és a $-\infty$ -t ill. a + és -0-t (ld. 1-27/b ábra).



1-27 ábra.
A végtelen értelmezése.

Projectív módban egy véges szám komparálása egy végtelennel érvénytelen művelet exception-t okoz, míg két végtelen komparálásának eredménye mindig "egyenlő". Másrészről, két végtelen összeadása vagy kivonása érvénytelen művelet exceptiont állít elő, míg az összeadás vagy kivonás eredménye, ha az egyik operandusa véges és a másik végtelen, végtelen.

Az *affine* mód a végtelennel végzett műveleteknek több fajtáját teszi lehetővé. A projectíve módhoz képest, itt lehetséges két azonos előjelű végtelen összeadása, ill. két különböző előjelű végtelenen végzett kivonás. A komparálások affine módban, a projectíve módtól eltérően működnek:

$-\infty < x < +\infty$ minden véges x-re.

Túlcsordulás. Ha a túlcsordulás exception maszkolt, a túlcsordulás eredménye egy olyan végtelen lesz, amelynek előjele azonos a valódi előjellel. Pl.: ha egy pozitív számot elosztunk egy negatív számmal és túlcsordulás jön létre, az eredmény $-\infty$.

A túlcsordulás exception maszkolása veszélyes lehet, mivel (1) az előjel megbízhatatlan lehet, (2) az eredmény nem valódi végtelen, mint a nullával való osztáskor, hanem egyszerűen csak túl nagy ahhoz, hogy ábrázolni lehessen. Pl.: elosztva egy nagy számot egy nagyon kicsivel, az eredmény előjele függ a kis szám előjelétől is, ami lehetséges, hogy egy korábbi kerekítési hiba miatt rossz.

Érvénytelen művelet. Korábban már láthattuk, hogy valahányszor nincs más kiút, a legkomolyabb exception, az érvénytelen művelet exception keletkezik. Tételezzük fel, hogy az exception maszkolva van és nézzük meg mi történik.

Elsőként vizsgáljuk meg az érvénytelen művelet exception hatását a komparálásokra. A 80287-es komparáló utasításai eredményüket a 80286-os 'flag'-jeihez hasonló 'condition' kódok beállításával jelzik. Közönséges esetben a 'condition' kódok kisebb, egyenlő, vagy nagyobb értéket kapnak két szám összehasonlítása után. Ha komparálás közben érvénytelen művelet exception keletkezik (pl. *projective* módban egy véges szám komparálása végtelennel) a 'condition' kód speciális 'not comparable' értéket kap (nem komparálható).
Ha numerikus művelet közben keletkezik érvénytelen művelet exception, akkor az

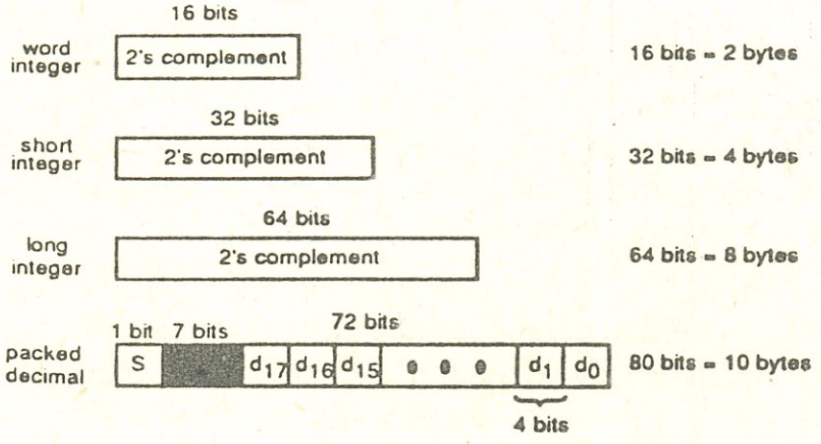
eredménye 'NaN'. (Not a Number - nem szám) Az összes olyan lebegőpontos szám, amelynek szignifikánsa különbözik 00...000-tól és exponense 11...111 egy 'NaN'-t reprezentál. Ebből adódóan, számos, különböző 'NaN' létezik.

A 'NaN'-ok előállítását és használatát szabályozó törvények meglehetősen egyszerűek. Ha egy művelet valamelyik operandusa 'NaN', érvénytelen művelet exception-t okoz. Ebben az esetben az eredmény a legnagyobb abszolút értékű 'NaN' operandus. Ha érvénytelen művelet exception jön létre, anélkül hogy bármelyik operandus 'NaN' lenne, az eredmény egy speciális 'NaN', amelyet *'indefinite'*-nek neveznek.

Az *'indefinite'* exponense 11...111, szignifikánsa 1. Mivel az egyszerű és duplapontosságú formátumok rejtett bevezető bittel rendelkeznek, az indefinitek ezekben a formátumokban 1000..00 szignifikáns mezővel rendelkeznek, míg kibővített formátumban 11000...0. A 80287-es, az 'indefinite'-en kívül, nem definiál több speciális 'NaN'-t. A fennmaradó 'NaN'-ok kódolása a programozók tetszése szerint történhet.

INTEGER formátumok

A 80287-es rendelkezik olyan utasításokkal, amelyek egész számot alakítanak át lebegőpontos formátumúvá és lebegőpontosokat egésszé. Az 1-28 ábra bemutatja azokat az integer formátumokat, amelyekkel a 80287-es műveleteket képes végezni. Az egy szó terjedelmű 'word' integer megegyezik a 80286-os integer formátumával. A 'short' és 'long' integer hasonlít az előbbihez, csak hosszabb. A 'Packed decimal', azaz a decimális formátum 18 BCD számjegyet tartalmaz, minden byte-on kettőt. A decimális számok legmagasabb címén, egy külön byte-ban helyezkedik el az előjel bit (0 = pozitív, 1 = negatív). A byte alsó hét bitjének 0-nak kell lennie.



1-28 ábra.
A 80287-es egész (*integer*) formátuma.

Az alábbi számok integer-ré konvertálási kísérlete érvénytelen művelet exception-t okoz: 'NaN', 'denormált', 'unnormált', végtelen, olyan szám, amely meghaladja az integer ábrázolási tartományt. Ha az érvénytelen művelet exception maszkolva van, a 80287-es megpróbál egy 'indefinite'-tet tárolni. Mivel az integer-eknél nem maradt szabad bit a hibakódok reprezentálására, egy 'indefinite' konvertálása az integer formátumok bármelyikébe a legnegatívabb egész számot, azaz az 100...00-t eredményezi. Ezt

visszakonvertálva lebegőpontos formátumba nem 'indefinite'-tet kapunk, hanem egy közönséges negatív számot. 'Indefinite' konvertálása decimálissá az alábbi számot adja: 1111 1111 1111 1111 XXXX ..... XXXX, X = definiálatlan azaz 8 byte szemét és két byte 1-es. Ezt visszakonvertálva értelmetlen eredményt kapunk.

A konvertálások legfőbb felhasználási területe a vegyes, integer-lebegőpontos számításokban van. Lehetőség van arra is, hogy a 80287-es szolgáltatásait felhasználva írjunk integer aritmetikát, a számok oda-vissza konvertálásával. Az 1-6 táblázat segít annak eldöntésében, hogy ez a megoldás mikor kifizetődő és mikor nem. (A táblázat adatai változhatnak a memória sebesség, az órajel frekvencia és a program függvényében.)

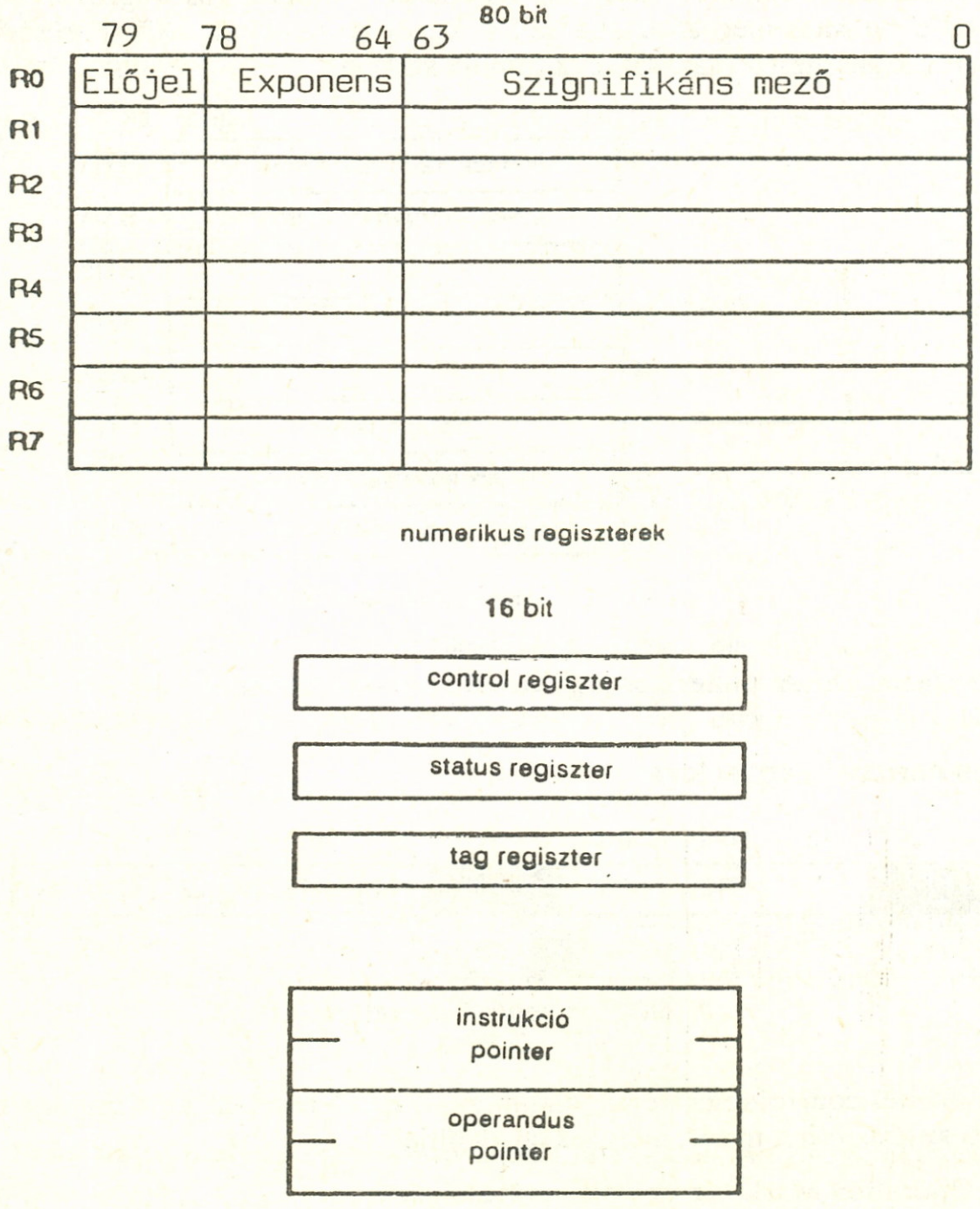
Művelet	16 bit	32 bit	64 bit
Összeadás/kivonás	286	286	286
Szorzás/osztás	286	286	287

1-6 táblázat.
A gyorsabb *integer* aritmetikájú processzorok.

A 80287-es regiszterkészlete

A 80287-es további, a lebegőpontos számításoknál hasznos regiszterek biztosításával támogatja a 80286-ost. A regiszterek az 1-29 ábrán láthatóak. A 80287-es utasítások operandusai egyaránt lehetnek a memóriában, vagy a nyolc numerikus regiszter valamelyikében. A numerikus regiszterek a számokat csak kibővített formátumban tárolják. A 80287-es gyorsabban képes elérni a numerikus regiszterekben levő operandusokat mint a memóriában levőket.

A *control regiszter* tartalmazza az exception maszk biteket, amelyek a 80287-es viselkedését szabályozzák. A regiszter tartalma csak akkor változik meg, ha közvetlenül töltik programból. A *status regiszter* bit-jei a 80287-es mindenkori állapotáról adnak tájékoztatást és általában nincsenek hatással a viselkedésére, ezek azok a bitek azaz 'condition' kódok, amelyek a komparáló műveletek eredményeit jelzik. A *'tag' regisztert* a 80287-es használja a numerikus regiszterek tartalmának összegzésére azért, hogy a hatékonyságot növelje és felfedezze az exception-okat. A programozók számára nem sok jelentőséggel bír. Az *instruction pointer* és az *operandus pointer* a leggyakrabban használt 80287-es utasítások és memória operandusaik (ha vannak) memória címeit biztosítják. Ez az információ az exception handler-ek számára hasznos. Vizsgáljuk meg részletesebben az egyes regiszter fajtákat.

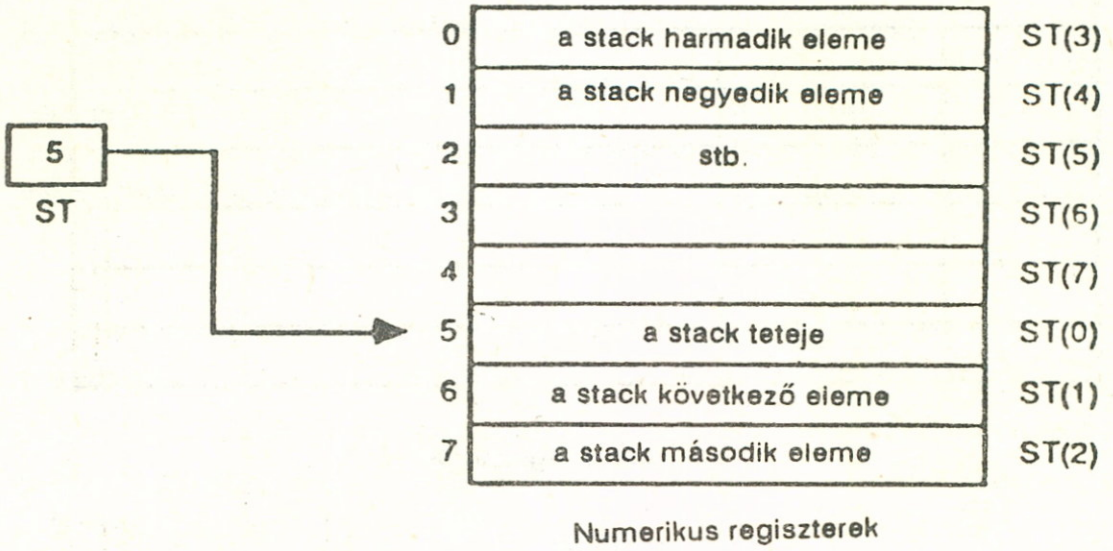


1-29 ábra.
A 80287-es regiszterkészlete.

A nyolc numerikus regiszter

A numerikus regiszterek címzése némileg szokatlan. Egy numerikus regiszter - a 80287-es utasításban meghatározott - sorszámát a 80287-es mindig hozzáadja a status regiszter ST (stack top) mezejéhez. A kifejezés értéke határozza meg az adott numerikus regisztert. Ennek a megoldásnak az az oka, hogy a numerikus regisztereket úgy lehessen használni, mint egy stack-et. Az 1-30 ábra illusztrálja a numerikus regiszterek stack-es értelmezését, azt feltételezve, hogy az ST tartalma 5. Mivel a 80287-es tervezői feltételezték, hogy a

legtöbb programozó stack-esen szeretné használni a numerikus regisztereket, ezért számos 80287-es utasítás megváltoztatja az ST mezőt is. Azoknak, akik jobb szeretnék a regisztereket "közönségesen" használni, ügyelniük kell erre.



1-30 ábra.
A numerikus regiszterek *stack*-es címzése.

A 80287-es control regisztere

1-31 ábra.
A 80287-es *control* regisztere.

A 80287-es 16 bites control regiszterének különböző mezőit az 1-31 ábra mutatja be. Ezek közül az első az 'exception mask', melynek egyes bitjei:

IM Invalid Operation Mask - érvénytelen művelet maszk
DM Denormalized Operand Mask - denormált operandus maszk
ZM Zero-Divide Mask - nullával való osztás maszk
OM Overflow Mask - túlcsordulás maszk
UM Underflow Mask - alúlcsordulás maszk
PM Precision (Inexact Result) Mask - pontatlan eredmény maszk

Ha a maszk bit 0, a hozzátartozó exception előfordulása a program felfüggesztését és egy 80286-os megszakítás érvényre jutását fogja okozni. Ha maszk bit 1, a hozzátartozó exception maszkolt és exception-értékek jönnek létre, ahogy azt korábban már leírtuk ebben a fejezetben.
A következő mező a 2 bites 'Precision Control' (PC), azaz pontosságvezérlő mező. Értékei a következők lehetnek:

11 kibővített pontosságú (default)
10 dupla pontosságú
00 egyszeres pontosságú.

A PC mező célja, hogy a 80287-es minden számot, mielőtt bekerülne valamelyik numerikus regiszterbe, a PC mező értékének megfelelő pontosságúra kerekítsen.

A következő 2 bites 'Rounding Control' (RC), kerekítés vezérlő mező a korábban már említett kerekítési módokat vezérli a következőképpen:

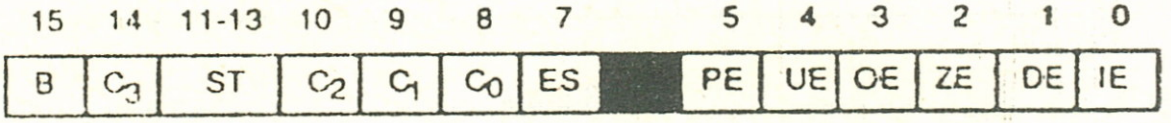
00 kerekítés a legközelebb eső ábrázolható szám felé (default)
01 kerekítés a negatív ∞ felé
10 kerekítés a pozitív ∞ felé
11 kerekítés a 0 felé.

A control regiszter utolsó mezője az 'Infinity Control' (IC) bit, amely a korábban már tárgyalt végtelen értelmezési módok valamelyikét jelöli ki, a következőképpen:

0 projective mód (default)
1 affine mód.

A 80287-es elődje, a 8087-es rendelkezik egy további mezővel is (IEM) a control regiszterben. Ez az interrupt-okat vezérlő mező a 80286-osban már meglevő interrupt kezelő mechanizmus kidolgozásának köszönhetően a 80287-esben már nem szükséges.

A 287-es status regisztere



1-32 ábra.
A 80287-es *status* regisztere.

Az 1-32 ábra a status regiszter különböző mezőit mutatja be. Az alsó biteken az exception flag-ek találhatók:

IE Invalid Operation Exception (érvénytelen művelet exception)
DE Denormalized Operand Exception (denormált operandus exception)
ZE Zero-divide Exception (0-val való osztás exception)
OE Overflow Exception (túlcsordulás exception)
UE Underflow Exception (alúlcsordulás exception)
PE Precision (Inexact Result) Exception (pontatlan eredmény exception)

Valahányszor egy numerikus exception lép fel (maszkolt vagy maszkolatlan) a 80287-es a megfelelő exception flag-et 1-be állítja. Az exception flag-ek "ragadósak", azaz a 80287-es

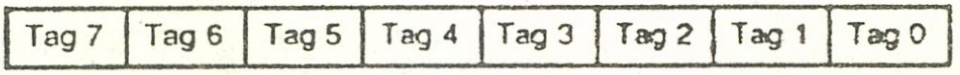
automatikusan állítja be őket, de mindaddig beállítva maradnak, amíg a programozó közvetlenül fel nem tölti a status regisztert egy új értékkel. Így pl. egy program kitörölheti az exception flag-eket, maszkolhatja az összes exception-t, és aztán végrehajthat néhány numerikus számítást. Ekkor a számítások végén a program ellenőrízheti az exception flag-eket, hogy lássa, tönkrement-e valami a számítások közben. Az exception-oknak ez a jól kézbentartható megközelítése gyakran sokkal egyszerűbb, mint írni egy exception handlert, ami a számítások közben bármikor meghívható. A következő mező az Error Summary (ES), azaz a hibaösszegző bit. A bit értéke 1 lesz, valahányszor egy 80287-es utasítás egy maszkolatlan exception-t állít elő, minden egyéb esetben 0.

A C0, C1, C2, C3 bitek a komparálásokat vagy a maradékképző utasításokat követően beálló állapotkódok. Az állapotkód bitek értelmezése utasításonként változhat, ezért azt majd maguknál az utasításoknál tárgyaljuk.

A 3 bites Stack-Top (ST), a stack tetejére mutató mezőt már tárgyaltuk; tartalma modulo 8, minden numerikus regiszter sorszámához hozzáadódik. Így, ha a numerikus regisztereket stack-esen használjuk, az ST annak a regiszternek a sorszámát tartalmazza, amely a legfelső eleme a stack-nek (ld. 1-30 ábra).

A status regiszter utolsó mezője a Busy (B) foglaltság jelző bit. A bit értéke 1 miközben a 80287-es utasítást hajt végre, vagy megszakítást kér, 0, amikor a 80287-es nem hajt végre semmit.

A tag (toldalék) regiszter



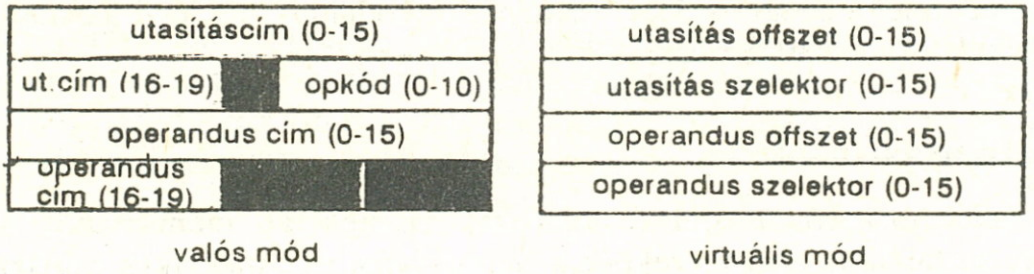
1-33 ábra
A 80287-es *tag* (toldalék) regiszter.

A 16 bites tag regiszter az 1-33 ábrán látható, 8, egyenként 2 bites mezőt tartalmaz, amit *tag*-nek neveznek. A mezők mindegyike egy numerikus regiszterhez van hozzárendelve, és azok tartalmát összegzi, a következőképpen:

00 érvényes (azaz minden véges 0-tól különböző szám)
01 zéro
10 érvénytelen (azaz NaN vagy végtelen)
11 üres

A tagregiszter értéke akkor 11 (üres), ha a hozzátartozó numerikus regiszter inicializálása még nem történt meg, vagy le van véve (pop) a numerikus regiszter stack-ről. Így a 80287-es az ilyen tag-okat a stack alúlcsordulás (túl sok pop) és a stack túlcsordulás (túl sok push) felderítésére használja. A stack alúl- és túlcsordulás érvénytelen művelet exception-t okoz. Ha ez az exception maszkolt, és egy művelet közben stack alúl- vagy túlcsordulás jön létre, a 80287-es úgy fogja beállítani az ST stack pointer-t, mintha semmi nem történt volna, de a művelet eredménye egy indefinite lesz.

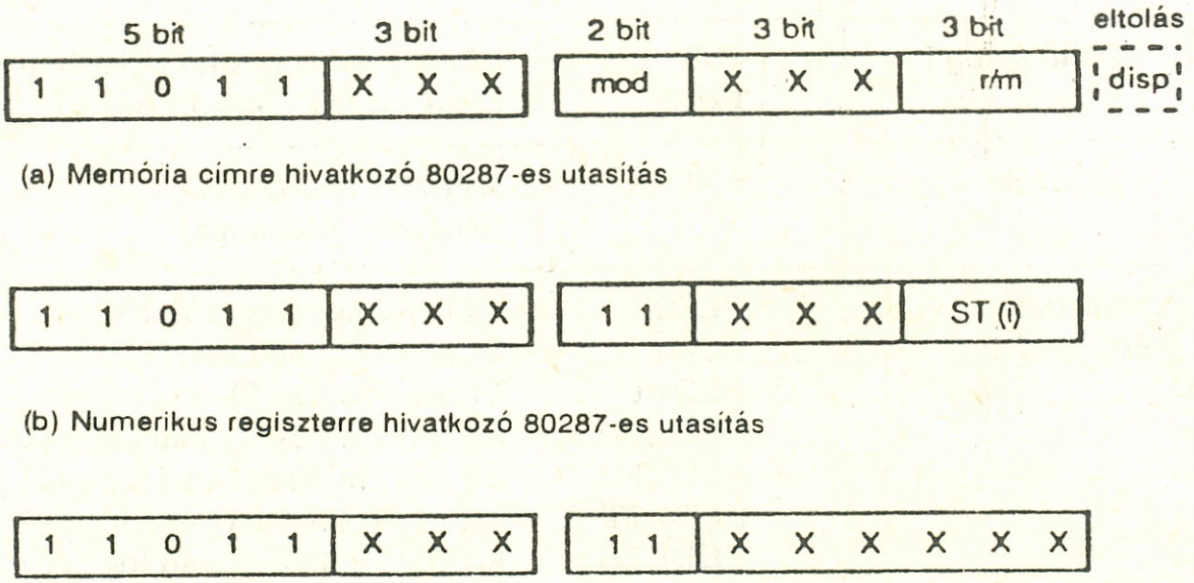
Az exception pointer-ek



1-34 ábra.
A 80287-es *exception* pointerei.

Az exception pointer-eket (utasítás és operandus) a 80287-es az exception handler-ek kiszolgálására használja. Két lehetséges formátumuk van, attól függően, hogy a 80287-es valós vagy virtuális üzemmódban van. A két formátum az 1-34 ábrán látható. Virtuális üzemmódban a pointer-ek az exception-t létrehozó utasítás és operandusának (ha van) szelektorát és offszet-jét tartalmazzák. Valós üzemmódban a 80287-es az exception handler-t az exception-t okozó utasítás: két részre bontott, 20 bites címével és a pointer alsó 11 bitjén annak opkódjával támogatja. Megjegyzendő, hogy ez az extra információ az utasítás pointer-rel könnyen elérhető. Bár a 80287-es valós üzemmódjának exception poïnter-ei ugyanolyan formátumúak, mint a 8087-éi, a 80287-es abban különbözik a 8087-estől, hogy utasítás pointer-e az opkódot megelőző első prefix-re (ha van) mutat, míg a 8087-es utasítás pointer-e minden prefix-et átugrik.

A 80287-es utasítás formátumai



1-35 ábra.
80287-es utasítás formátumok.

A 80287-es és a 80286-os utasítás formátumai megegyeznek. Az 1-35 ábra a 80287-es utasítás formátumokat foglalja össze. Minden 80287-es utasítás opkódja 11011-el kezdődik. Ez megfelel a 80286-os ESC (escape) opkódjának. Ez jelzi a 80286-osnak, hogy az utasítást át kell adnia a 80287-esnek.

Adatmozgató utasítások

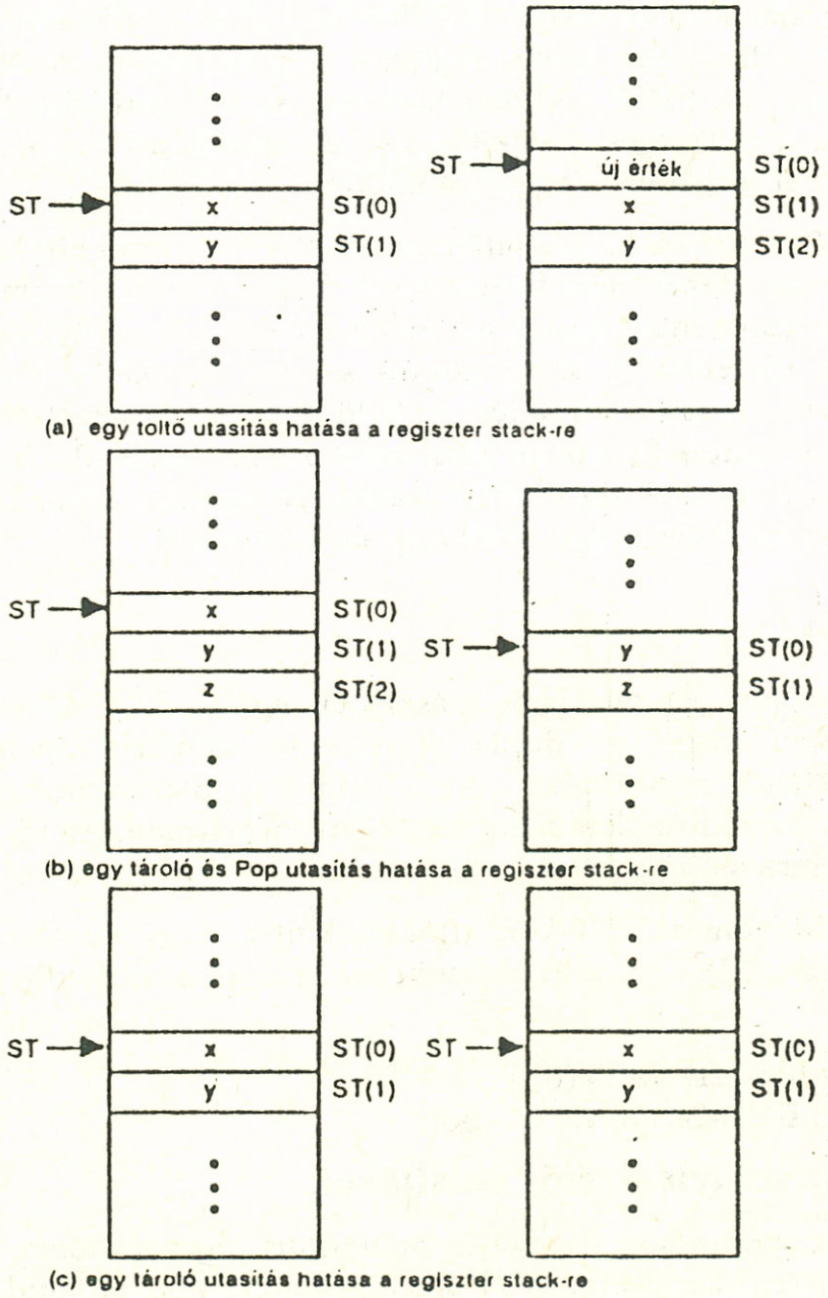
A 80287-es adatmozgató utasításai adatokat visznek át memóriából a numerikus regiszterekbe, numerikus regiszterekből a memóriába, és numerikus regiszterek között. További adatmozgató utasítások a numerikus regisztereket inicializálják. Az adatmozgató utasítások az 1-7 táblázatban találhatók.

Utasítások	Mnemonikus kód	Megnevezés
Push jellegű	FLD	lebegőpontos töltés (Floating Load)
	FILD	egész töltés (Integer Load)
	FBLD	BCD töltés (BCD Load)
Pop jellegű	FSTP	lebegőpontos tárolás és Pop (Floating Store and Pop)
	FISTP	egész tárolás és Pop (Integer Store and Pop)
	FBSTP	BCD tárolás és Pop (BCD Store and Pop)
Tároló jellegű	FST	lebegőpontos tárolás (Floating Store)
	FIST	egész tárolás (Integer Store)
Felcserélő	FXCH	regiszter felcserélés (Register Exchange)
Konstans értékadó	FLDZ	nulla töltése (Load Zero)
	FLD1	1 töltése (Load One)
	FLDPI	¶ töltése (Load ¶)
	FLDLG2	$\log_{10}(2)$ töltése (Load $\log_{10}(2)$)
	FLDLN2	$\log_e(2)$ töltése (Load $\log_e(2)$)
	FLDL2T	$\log_2(10)$ töltése (Load $\log_2(10)$)
	FLDL2E	$\log_2(e)$ töltése (Load $\log_2(e)$)

1-7 táblázat.
80287 adatmozgató utasítások.

Load (töltő) utasítások

Az FLD utasítás egy egyszerű operandust - ami lehet egy numerikus regiszter vagy bármilyen formátumú lebegőpontos szám a memóriában - helyez a regiszter stack-re (push). Az FLD hatása az 1-36/a ábrán látható. Részletesen: az FLD először előveszi az operandusát, és ha szükséges kibővített formátumúvá konvertálja, aztán az ST tartalmát csökkenti 1-gyel (modulo 8), végül az operandust elhelyezi a stack (új) legfelső regiszterében.



1-36 ábra.
Az adatmozgató utasítások hatása.

A FILD és FBLD utasítások megegyeznek az FLD-vel, attól eltekintve, hogy a memóriában lévő operandusai integer ill. decimális formátumúak. A FILD az integer-ek összes formáját (16, 32 és 64 bites) képes használni. Mielőtt az operandus a regiszter stack-re kerülne, kibővített formátumú lebegőpontos számmá konvertálódik. Így a FILD egy álcázott integer-lebegőpontos konvertáló utasítás, s ugyanígy az FBLD egy decimális-lebegőpontos átalakító.

Store és pop (tároló) utasítások

Az FLD ellentéte az FSTP, amely a regiszter stack tetején lévő adatot leveszi, és egy meghatározott argumentumba teszi. Az FSTP hatása az 1-36/b ábrán látható. Az FSTP operandusa egyaránt lehet numerikus regiszter vagy bármilyen formátumú lebegőpontos szám a memóriában. Az FSTP elsőként kiolvassa az ST által jelzett regiszter tartalmát, ezután végrehajtja a szükséges átalakításokat, az eredményt tárolja az operandusában, végül az ST tartalmát megnöveli eggyel (modulo 8).

Az FISTP és FBSTP utasítások hasonlítanak az FSTP-re, attól eltekintve, hogy memória operandusuk (minden lehetséges formátumú) integer-ek ill. decimális számok. Így az FISTP egy lebegőpontos-integer, míg a FBSTP egy lebegőpontos-decimális konverter. A konvertáló eljárás részeként a lebegőpontos számot kerekíteni kell. Az FISTP ezt a kerekítést, a control regiszter kerekítést vezérlő mezőjének megfelelően hajtja végre. Az FBSTP a control regisztert figyelmen kívül hagyva a kerekítést 0.5 hozzáadásával, majd a tizedesek levágásával végzi el. Az FBSTP utasítás az egyetlen olyan 287-es utasítás, amely a kerekítést vezérlő mező figyelmen kívül hagyásával kerekít.

Store utasítások

Az FST utasítás a ST (0) tartalmát másolja át egy numerikus regiszterbe, vagy egy egyszerű formátumú, vagy egy dupla pontosságú memória operandusba, miközben végrehajtja a szükséges konverziókat. Az ST stack pointer változatlan marad. Az FST utasítás memória operandusa nem lehet kerekített formátumú. Az FST hatását a regiszter stack-ra az 1-36/c ábra mutatja be.

Az FIST utasítás hasonlít az FST-hez, attól eltekintve, hogy memória operandusa 16, 32 vagy 64 bites integer. FBST utasítás nincsen, helyettesítése megoldható egy push-t követő pop-pal, pl.

```
FLD ST(0)  ; megduplázza az ST(0)-t
FBSTP DECVAR; BCD-be konvertál  pop
```

Register Exchange (felcserélő) utasítások

Az FXCH utasítás operandusa bármelyik numerikus regiszter lehet, amelynek tartalmát felcseréli az ST(0) regiszterével. Az ST stack pointer-t változatlanul hagyja. Kb. kétszer olyan gyors, mint az FLD, FSTP és FST.

Constant Load (konstans-betöltő utasítások)

A 80287-es rendelkezik az utasításoknak egy olyan csoportjával, amelynek minden tagja a regiszter stack konstansokkal való feltöltésére szolgál. Pl. az FLD1 utasítás 1.0-t tölt a regiszter stack-re. A konstans-betöltő utasítások az 1-7 táblázatban láthatók. Ezek az utasítások kb. kétszer olyan gyorsak, mint a közönséges load.

Aritmetikai utasítások

A 80287-esen az alapvető aritmetikai utasítások - összeadás, kivonás, szorzás, osztás - mindegyike két operandusú (source és destination) forrás és céloperandus. Hatásuk a következő:

cél ← cél op forrás, ahol op = +, *, -, /

A 287-es az olyan esetekben, ahol az operandusok nem felcserélhetőek (kivonás és osztás) a számítások fordított változatát is biztosítja.

cél ← forrás op cél, ahol op = -, /,.

Az operandusok egyikének minden esetben az ST(0)-nak kell lennie. A lehetséges hat változat a következő:

a. Fxxx
b. Fxxx mem
c. FIxxx mem
d. Fxxx ST, ST(i)
e. Fxxx ST(i), ST
f. FxxxP ST(i), ST

ahol xxx = ADD, SUB, MUL, DIV, SUBR (fordított kivonás), vagy DIVR (fordított osztás).

Az egyes változatok hatásai az 1-37 ábrán láthatóak (kivonás esetén).

Az a. változatban a 80287-es leveszi a két operandust a stack tetejéről, és az eredményt visszateszi a stack-re.

A b. változatban a forrás operandus a memóriában van, a cél operandus az ST(0). Az ST stack pointer változatlan marad.
Az egyszerű formátumú és a dupla pontosságú memória operandusok megengedettek, a kibővített formátumúak nem.

A c. változat majdnem olyan, mint a b. attól eltekintve, hogy a memória operandus egy 16 vagy 32 bites integer. Lebegőpontos formátumba történő átalakítása automatikus.

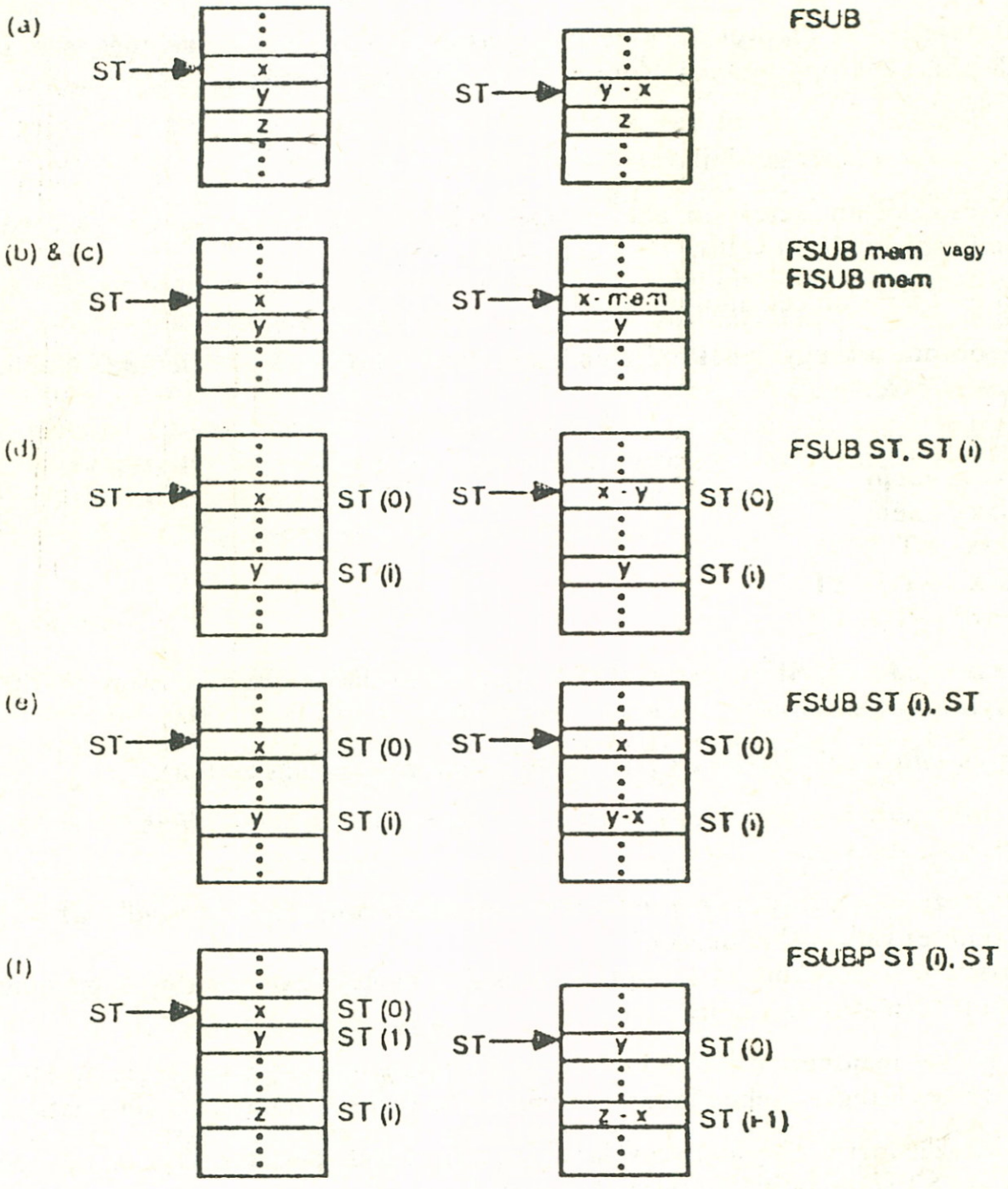
A d. változat forrás operandusa valamelyik numerikus regiszter ST(i), a cél operandus az ST(0). Az ST stack pointer változatlan marad.

Az e. változat az ST(0)-t használja forrás és valamelyik ST(i)-t cél operandusaként.
A stack pointer itt sem változik.

Végül az f. változat cél operandusa az ST(i) és forrás operandusa az ST(0). A művelet

hatásaként a forrás ST(0) lekerül a stack-ről.

Az 1-8 táblázat a tizennyolc lehetséges aritmetikai utasítás mnemonik-ját sorolja fel.



1-37 ábra.
Különböző kivonó utasítások hatása.

Művelet	Stack	Integer	Pop
Összeadás	FADD	FIADD	FADDP
Kivonás	FSUB	FISUB	FSUBP
Felcserélt operandusú kivonás	FSUBR	FISUBR	FSUBRP
Szorzás	FMUL	FIMUL	FMULP
Osztás	FDIV	FIDIV	FDIVP
Felcserélt operandusú osztás	FDIVR	FIDIVER	FDIVERP

1-8 táblázat
Alap aritmetikai utasítások.

További vegyes aritmetikai utasítások

A 80287-es a fent említett aritmetikai alapműveleteken kívül rendelkezik még néhány utasítással. Ezek a különféle aritmetikai utasítások az 1-9 táblázatban láthatók.

Mnemonik	Művelet
FSQRT	gyökvonás (Square Root)
FSCALE	hatványozás (Scale by Power of Two)
FPREM	részleges maradékképzés (Partial Remainder)
FRNDINT	kerekítés (Round to Integer)
FXTRACT	szétválasztás exponensre és szignifikánsra (Extract Exponent and Significand)
FABS	abszolút érték képzés (Absolute Value)
FCHS	előjel váltás (Change Sign)

1-9 táblázat.
Egyéb aritmetikai utasítások.

Az FABS utasítás (nincs explicit operandusa) az ST(0) tartalmát megváltoztatja az abszolut értékére. Hasonlóan az FCHS utasítás megváltoztatja (azaz komplementálja) az ST(0) előjelét, az FRNDINT az ST(0) integer-ré kerekíti, és az FSQRT az ST(0)-ban annak négyzetgyökét helyezi el. Az FSQRT rendkívül gyors, gyorsabb mint az osztás.
A hatványozó utasítás, az FSCALE az ST(1)-ben található lebegőpontos integer-nek feltételezett értéket hozzáadja az ST(0) exponenséhez, megváltoztatva ezzel a ST(0)-t. Ez

A hatványozó utasítás, az FSCALE az ST(1)-ben található lebegőpontos integer-nek feltételezett értéket hozzáadja az ST(0) exponenséhez, megváltoztatva ezzel a ST(0)-t. Ez a lebegőpontos számok 2 hatványaira való emelésének gyors módszere.

Az FXTRACT az ST(0)-t két lebegőpontos számra bontja szét, az eltolás nélküli exponensére és az előjeles szignifikánsára. Az exponens a régi ST(0) helyére kerül, a szignifikáns pedig a stack tetejére. Megjegyzendő, hogy az FXTRACT utasítást követő FSCALE visszaállítja az eredeti számot.

Az FPREM utasítás az ST(0)/ST(1) maradékát adja. A maradék előjele megegyezik a ST(0)-éval. Az eredmény a régi ST(0) helyére kerül. Az FPREM rövidítés a 'partial remainder' azaz közbülső maradék rövidítésből származik, amely arra utal, hogy az utasítás végetérhet anélkül is, hogy a munkáját befejezné. Az FPREM ismételt kivonásokat hajt végre, ha 64 vagy ennél kevesebb kivonás elegendő ahhoz, hogy az ST(0) kisebb legyen, mint az ST(1), akkor az FPREM végrehajtása után, az ST(0) a korrekt maradékot fogja tartalmazni, és a C2 állapotkód bit a status regiszterben nulla lesz. Ha 64 kivonás nem elegendő, a C2 értéke 1, és az ST(0) a 64 kivonással csökkentett értéket fogja tartalmazni. Az FPREM utasítás ismételt végrehajtása addig, amíg a C2 0 nem lesz, pontosan a kívánt maradékot fogja előállítani. Az FPREM azért hajt végre egyszerre csak 64 kivonást, hogy a 80287-es megszakítható legyen az FPREM egymást követő végrehajtásai között. A 64-es szám garantálja, hogy az FREM ne tartson tovább egy osztás utasításnál, és így a 287-est használó rendszer interrupt érvényre jutási idejét ne növelje.

Az FPREM ezenkívül feltehetőleg beállítja a C0, C3, C1 állapotkód biteket a quociens alsó három bitjének megfelelően. Sajnálatos módon ez a szolgáltatás nincs helyesen megvalósítva a 80287-en, s ezért az állapotkód bitek az FPREM egymást követő végrehajtása közben megbízhatatlanok.

Mielőtt befejeznénk a 80287-es aritmetikai utasításainak tárgyalását, érdemes egy pillantást vetnünk az 1-10 táblázatra, amely a különböző aritmetikai utasítások végrehajtási idejének megközelítő értékeit tartalmazza. A táblázat feltételezi, hogy a 80287-es 5.3 MHz-es órajel frekvenciával működik. 8 MHz-es órajel frekvencia esetén a 80287-es kb. 50%-al gyorsabb.

Művelet	Végrehajtási idő (us)	Sebesség (1000 művelet/s)
* r1 = r1 + r2	16	63
r1 = r1*r2	26	38
r1 = r1/r2	37	27
r1 = r1↑0.5	35	29
dupla pontosságú töltés	9	111
dupla pontosságú tárolás	19	53
integer (32 bit) töltés	11	91
integer (32 bit) tárolás	17	59

* - Megjegyzés: r1, r2 numerikus regiszterek

1-10 táblázat.
A 80287-es közelítő sebesség jellemzői 5,3 MHz órajel frekvencia esetén.

Komparáló utasítások

A 80287-es komparáló utasításai az 1-11 táblázatban láthatók. Az FCOM x utasítás összehasonlítja az ST(0)-t az operandusával (x), és az állapotkódokat ennek megfelelően - az 1-12 táblázatban látható módon - állítja be. Operandusa lehet egy numerikus regiszter, vagy egy egyszerű formátumú, vagy egy dupla pontosságú szám a memóriában. Az ST stack pointer nem változik.

Mnemonik	Művelet
FCOM	komparálás (Compare)
FICOM	integer komparálás (Integer Compare)
FCOMP	komparálás és pop (Compare & pop)
FICOMP	integer komparálás és pop (Integer Compare & pop)
FCOMPP	komparálás és kétszeres pop (Compare & pop twice)
FTST	komparálás nullával (Compare to zero)
FXAM	flag-ek vizsgálata (Examine)

1-11 táblázat.
Komparáló utasítások.

C3	C0	Feltétel
0	0	ST(0) > x
0	1	ST(0) < x
1	0	ST(0) = x
1	1	ST(0) & x nem összehasonlítható (típusa nem egyezik, vagy egyéb hiba)

1-12 táblázat.
Állapotkódok komparálás után.

A FICOM utasítás ugyanolyan, mint az FCOM, eltekintve attól, hogy operandusa 16, vagy 32 bites integer.

Az FCOMP és a FICOMP ugyanúgy működik, mint az FCOM, azzal a különbséggel, hogy a stack-en a komparálás végrehajtása után egy pop-ot hajt végre.

Végül az FCOMPP (amelynek nincs explicit operandusa) működése megegyezik az

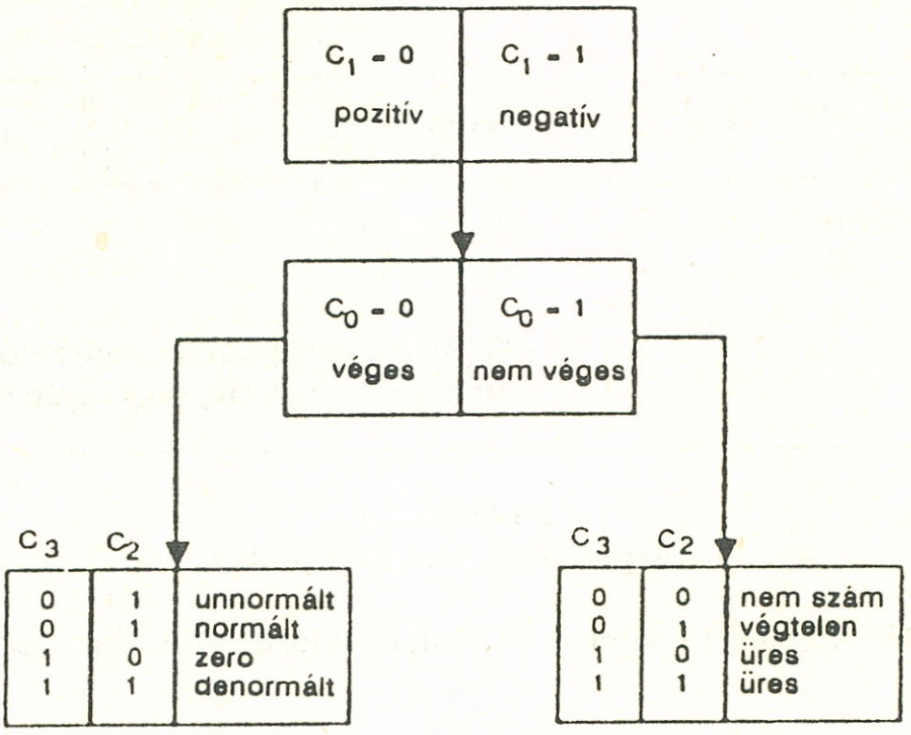
FCOM-éval, de mindkét operandusát leveszi a stack-ről a komparálás végrehajtása után.

C3	C0	Feltétel
0	0	ST(0) > 0
0	1	ST(0) < 0
1	0	ST(0) = 0
1	1	ST(0) & 0 az operandus nem komparálható 0-val

1-13 táblázat.
Állapotkódok FTST utasítás után.

Az FTST (test) utasítás az ST(0)-t hasonlítja össze zero-val, s ennek alapján állítja be - az 1-13 táblázatban látható módon - az állapotkódokat.

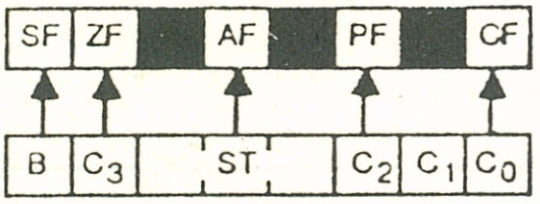
Az FXAM (examine) utasítás ugyanolyan mint az FTST, azzal a különbséggel, hogy több információt szolgáltat az ST(0)-ról, mint az FTST. Az állapotkódok az FXAM végrehajtását követően - az 1-38 ábrán látható módon - állnak be. Paradox módon, a FXAM kb. kétszer olyan gyors, mint az FTST. Az FXAM utasítás végrehajtása a leggyorsabb mód az állapotkódok inicializálására.



1-38 ábra.
Állapotkódok FXAM utasítás után.

Most eljutottunk oda, hogy hogyan használjuk a 80287-es állapotkódjait a feltételes

ugrásoknál. A feltételes ugrások használata egy kicsit kényelmetlen, mivel a 80286-os (amely végrehajtja az ugrást) és a 80287-es (amely tartalmazza az állapotkódokat) egymástól független chip-ek. Még rosszabb eset a 8086-é, mivel a 8086-os már a piacon volt, mikor 8087-t tervezték, s ezért a 8086-ost nem tudták úgy megváltoztatni, hogy jól együttműködjön a 8087-essel.



Az AH leképzése a SAHF utasítással

A 80287-es státusz regiszterének leképzése az AX regiszterbe az FSTSW AX utasítással

1-39 ábra.
A 80286-os és a 80287-es állapotkódjai közötti összefüggés.

A 80287-es FSTSW AX utasítása a 80287-es status regiszterének tartalmát másolja át a 80286-os AX regiszterébe. Így az állapotkódokat 80286-os utasításokkal meg lehet vizsgálni. Erre van egy gyors, trükkös megoldás. A 80286-os rendelkezik egy SAHF (Store AH into FLAGS) utasítással, amelyet eredetileg azért készítettek a 8086-osba, hogy biztosítsa a kompatibilitást az Intel 8080-al. Ha az FCOM utasítást használjuk, két lebegőpontos szám összehasonlítására, majd az FSTSW utasítással a 80287-es status regiszterét az AX-be tesszük, majd ezt követően a 80286-os SAHF utasítással beállítjuk a 80286-os flag-jeit az AH regiszter (az AX regiszter felső byte-ja) értékének megfelelően - az 1-39 ábrán látható eredményt kapjuk. Sajátságos módon a 80286-os ZF zero flag-je a 80287-es C3 bitjének, míg a CF carry flag-je a C0-nak felel meg. Az olvasó most ellenőrizheti, hogy a 80287-es FCOM utasítása hasonló 80286-os flag beállításhoz vezet, mint a 80286-os CMP integer komparáló utasítás, azzal a különbséggel, hogy az 'incomparable' állapot nem érhető el a 80286-oson.

Transzcendentális utasítások

Az elemi transzcendentális függvények magukban foglalják a trigonometrikai függvényeket (sinus, cosinus, tangens stb.), az inverz trigonometrikus függvényeket (arcsin, arccos, arctg, stb.), a logaritmus függvényeket (log2, log10, loge), az exponenciális függvényeket (X↑z, 2↑x, 10↑x, e↑x), a hiperbolikus függvényeket (sinh, cosh, tgh, stb.), és az inverz hiperbolikus függvényeket (arcsinh, arccosh, arctgh, stb.). A 287-es az elemi transzcendentális függvényeket dupla pontosságú argumentumokon végzi el, s eredményül 1 bit pontossággal dupla pontosságú számot ad. Azok a különálló blokkok, amelyek a kibővített formátumot lehetővé teszik, a gyors négyzetgyök vonó utasítás, a nagy pontosságú maradékképző utasítás, és - az 1-14 táblázatban látható - transzcendentális utasítások. A 287-es ezeknél az utasításoknál annak a tervezési elvnek a jegyében született, mely szerint csak a leginkább időpusztító számításokat kell közvetlenül hardware-ben megoldani. Így az egyes elemi függvényeket kis software szubrutinokkal kell megvalósítani.

Mnemonik	Művelet
FPTAN	parciális tg (Partial tangent)
FPATAN	parciális arctg (Partial arctangent)
FYL2X	y↑log2 (x)
FYL2XP1	y↑log2 (x+1)
F2XM1	2↑x - 1

1-14 táblázat.
Transzcendentális utasítások.

A FPTAN parciális tg utasítás eredményét két számra bontva, x-ben és y-ban helyezi el, úgy, hogy y/x = tg (ST(0)). Az y az ST(0) tartalmát írja felül, míg az x a stack tetejére kerül. Az utasítás argumentumainak normalizáltnak kell lenniük; a denormáltak, unnormáltak, végtelenek és NaN-ok érvénytelenek. Ez a megszorítás minden transzcendentális utasításra igaz. Ezenkívül az FPTAN argumentumának 0 és ¶/4 közé kell esnie (0<=ST(0)<=¶/4). Ha az argumentum érvénytelen, vagy a fenti feltételnek nem tesz eleget, az FPTAN hibás eredményt fog adni, anélkül, hogy exception-t okozna. Ez a tulajdonság jellemző minden olyan transzcendentális utasításra, amelynek argumentumának egy adott tartományon belül kell lennie. Ezért az érvényesség vizsgálatot és az argumentum átalakítását úgy, hogy az a megengedett tartományon belülre essen, a programozónak kell elvégeznie.

A további trigonometrikus függvények az FPTAN felhasználásával valósíthatók meg az 1-15 táblázat egyenletei szerint. Az 1-15 táblázat egyenletei tg(z) helyett tg(z/2) alakban vannak kifejezve. A 2-vel való osztás a FSCALE utasítás segítségével gyorsan végrehajtható, és kimutatható, hogy ez a közelítés nagymértékben csökkenti a kerekítés hibáit.

$$\sin(z) = \frac{2\,(y/x)}{[1 + (y/x)^2]} \qquad \csc(z) = \frac{[1 + (y/x)^2]}{2\,(y/x)}$$

$$\cos(z) = \frac{[1 - (y/x)^2]}{[1 + (y/x)^2]} \qquad \sec(z) = \frac{[1 + (y/x)^2]}{[1 - (y/x)^2]}$$

$$\tan(z/2) = \frac{y}{x} \qquad \cot(z/2) = \frac{x}{y}$$

z = a stack tetején lévő érték az FPTAN végrehajtása előtt
x, y = a stack felső és ezt követő értéke a z 2-vel való osztása után;
az FPTAN-t ez után kell végrehajtani

1-15 táblázat.
A trigonometrikus függvényeket megvalósító egyenletek.

Az FPATAN az arctg(ST(1)/ST(0))-t számítja ki. Az operandusoknak ki kell elégíteniük a $0<ST(1)<ST(0)$. A többi inverz trigonometrikus függvény hasonló módon - az 1-16 táblázatban látható egyenletek alapján - a FPATAN és FSQRT függvények segítségével számíthatók ki.

$$\arcsin(z) = \arctan\left(\frac{z}{\sqrt{(1-z)(1+z)}}\right) = \arctan\left(\frac{y}{x}\right)$$

$$\arccos(z) = 2\arctan\left(\sqrt{\frac{1-z}{1+z}}\right) = 2\arctan\left(\frac{y}{x}\right)$$

$$\arctan(z) = \arctan\left(\frac{z}{1}\right) = \arctan\left(\frac{y}{x}\right)$$

$$\operatorname{arccot}(z) = \arctan\left(\frac{1}{z}\right) = \arctan\left(\frac{y}{x}\right)$$

$$\operatorname{arccsc}(z) = \arctan\left(\frac{\operatorname{sign}(z)}{\sqrt{(z-1)(z+1)}}\right) = \arctan\left(\frac{y}{x}\right)$$

$$\operatorname{arcsec}(z) = 2\arctan\left(\sqrt{\frac{z-1}{z+1}}\right) = 2\arctan\left(\frac{y}{x}\right)$$

z = az inverz trigonometrikus függvény argumentuma
x, y = a stack-en lévő két érték az FPATAN végrehajtása előtt;
az FPATAN-t követően a stack-en a kívánt f(z) érték található.

1-16 táblázat.
Az inverz trigonometrikus függvényeket megvalósító egyenletek.

$$\log_2(x) = \text{FYL2X}(x)$$

$$\log_e(x) = \log_e(2)\,\log_2(x) = \text{FYL2X}(\log_e(2),x) = \text{FYL2X}(\text{FLDLN2},x)$$

$$\log_{10}(x) = \log_{10}(2)\,\log_2(x) = \text{FYL2X}(\log_{10}(2),x) = \text{FYL2X}(\text{FLDLG2},x)$$

1-17 táblázat.
A logaritmikus függvények egyenletei.

Az FYL2X az ST(1) log2ST(0)-t számítja ki. A két operandus lekerül a stack-ről, az eredmény pedig felkerül rá. A FYL2C megkívánja, hogy $ST(0)>0$. A többi logaritmus

függvény az 1-17 táblázatban látható egyenletek alapján az FYL2X, az FLDLN2 és az FLDLG2 utasítások segítségével valósítható meg. Egy további logaritmus utasítás a FYL2XP1 az ST(1) log2[ST(0) + 1]-t számítja ki. Az ST(0)-nak ki kell elégítenie az alábbi feltételt:

$|ST(0)| < 1 - 1/2\uparrow 0.5$

$$\text{arcsinh}(x) = [\text{sign}(x)]\ [\log_e(2)]\ [\log_2(1+z)]$$
$$= \text{FYL2XP1}(\text{sign}(x)\ \text{FLDLN2}, z)$$

ahol $$z = |x| + \frac{|x|}{\frac{1}{|x|} + \sqrt{1 + \left(\frac{1}{|x|}\right)^2}}$$

$$\text{arccosh}(x) = [(\log_e(2)]\ [\log_2(1+z)] + \text{FYL2XP1}(\text{FLDLN2}, z)$$

ahol $$z = x - 1 + \sqrt{(x-1)\ (x+1)}$$

és $$x \geqslant 1$$

$$\text{arctanh}(x) = [\text{sign}(x)]\ [\log_e(2)]\ [\log_2(1+z)] =$$
$$= \text{FYL2XP1}(\text{FLDLN2 sign}(x), z)$$

ahol $$z = \frac{2\ |x|}{1 - |x|}$$ és $$-1 < x < 1$$

$$\text{arccoth}(x) = \text{arctanh}(1/x) \qquad \text{arccsch}(x) = \text{arcsinh}(1/x)$$
$$\text{arcsech}(x) = \text{arccosh}(1/x)$$

1-18 táblázat.
Az inverz hiperbolikus függvények egyenletei.

Az utasítás létrejöttének oka, hogy a log(1+x) függvényt nagy pontossággal lehessen kiszámítani. Ez a függvény igen hasznos az inverz hiperbolikus függvények számításánál (ld. 1-18 táblázat egyenletei).

Az F2XM1 hatványozó utasítás a $2\uparrow ST(0)-1$ értéket számítja ki. Az eredmény az ST(0)-ba kerül. Az ST(0)-nak 0 és 1/2 közé kell esnie.

A 80287-es a 2x helyett a $2\uparrow x-1$-et számítja ki, azért, hogy elkerülje a pontosság romlását, amikor az X értéke közel esik a 0-hoz. A hiperbolikus függvények kiszámításánál ez az eset fordul elő, ahogy az az 1-19 táblázatban látható. A többi hatványozó függvény a F2MX1 használatával - az 1-20 táblázatban foglaltak szerint - valósítható meg.

$$\sinh(x) = \frac{\mathrm{sign}(x)}{2}\left[(e^{|x|}-1) + \frac{e^{|x|}-1}{e^{|x|}}\right]$$

$$\cosh(x) = \frac{1}{2}\left[e^{|x|} + \frac{1}{e^{|x|}}\right]$$

$$\tanh(x) = \mathrm{sign}(x) - \left[\frac{e^{2|x|}-1}{e^{2|x|}+1}\right]$$

$$\coth(x) = 1/\tanh(x)$$

$$\mathrm{csch}(x) = 1/\sinh(x)$$

$$\mathrm{sech}(x) = 1/\cosh(x)$$

1-19 táblázat.
A hiperbolikus függvények egyenletei.

$$2^x=(2^x-1)+1=F2XM1(x)+1$$

$$e^x=1+(2^{x\ \log_2(e)}-1)=1+F2XM1(x\ \log_2(e))=1+F2XM1(xFLDL2E)$$

$$10^x=1+(2^{x\ \log_2(10)}-1)=1+F2XM1(x\ \log_2(10))=1+F2XM1(xFLDL2T)$$

$$x^y=1+(2^{y\ \log_2(x)}-1)=1+F2XM1(y\ \log_2(x))=1+F2XM1(FYL2X(y,x))$$

1-20 táblázat.
Az exponenciális függvények egyenletei.

Elérkeztünk a 80287-es utasítások utolsó csoportjához, az adminisztratív utasításokhoz, de mielőtt ezeket áttekintenénk, érdemes megvizsgálnunk azt a problémát, ami abból adódik, hogy a 80286-os és a 80287-es két, különálló processzor.

A 80286-os és a 80287-es konkurrens működése

Lehetőség van arra, hogy a 80287-es és a 80286-os egyidejűleg működjön. Amikor a 80286-os elér egy 80287-es utasítást:

1. vár, amíg a 80287-es szabaddá válik

2. biztosítja, hogy a 80287-es elkezdje az utasítás végrehajtását
3. azonnal továbblép a következő utasításra.

A 80286-os nem várja meg, hogy a 80287-es befejezze munkáját. Ha a 80286-os elér egy következő 80287-es utasítást, és a 80287-es még nem fejezte be az előző utasítás végrehajtását, akkor a 80286-os az 1. pont szerint várakozni fog. Ez a stratégia biztosítja a 80286-os folytonos futását, miközben a 80287-es egy lassú utasítást hajt végre. Mialatt a 80286-os az 1. pont szerint várakozik, garantálja, hogy majdnem minden program ugyanazt az eredményt fogja előállítani, mintha kettő helyett csak egyetlen processzor lenne. Azonban van két olyan ritka eset, ami ezt az illúziót megtöri.

Tételezzük fel, hogy a 80287-es utasítást kapott, hogy tároljon valamit a memóriában, amit ezután a 286-os azonnal felhasznál. Pl.

```
FIST I
MOV  AX,I.
```

Ebben az esetben a 80286-os előbb fogja végrehajtani a MOV utasítást, mint ahogy a 80287-es a FIST-t befejezi. Így az AX hibás értéket fog kapni. Megjegyzendő, hogy ez a probléma nem jelentkezik, ha a FIST után van egy közbülső 80287-es utasítás, mint pl.

```
FIST I
FADD ST(3)
MOV  AX,I
```

mivel ebben az esetben a 80286-os megvárja a FIST végét, mert a FADD 80287-os utasítás.

Azonban 80286-os utasítás beszúrása

```
FIST I
MOV  J,BX
MOV  AX,I
```

nem fog segíteni, hacsak az utasítás késleltetési ideje nem elegendő, hogy a 80287-es befejezze a FIST-t.
A 80286-os FWAIT utasítás hatására a 80286-os addig várakozik, amíg a 80287-es szabaddá nem válik. Így a probléma megoldható egy FWAIT beszúrásával:

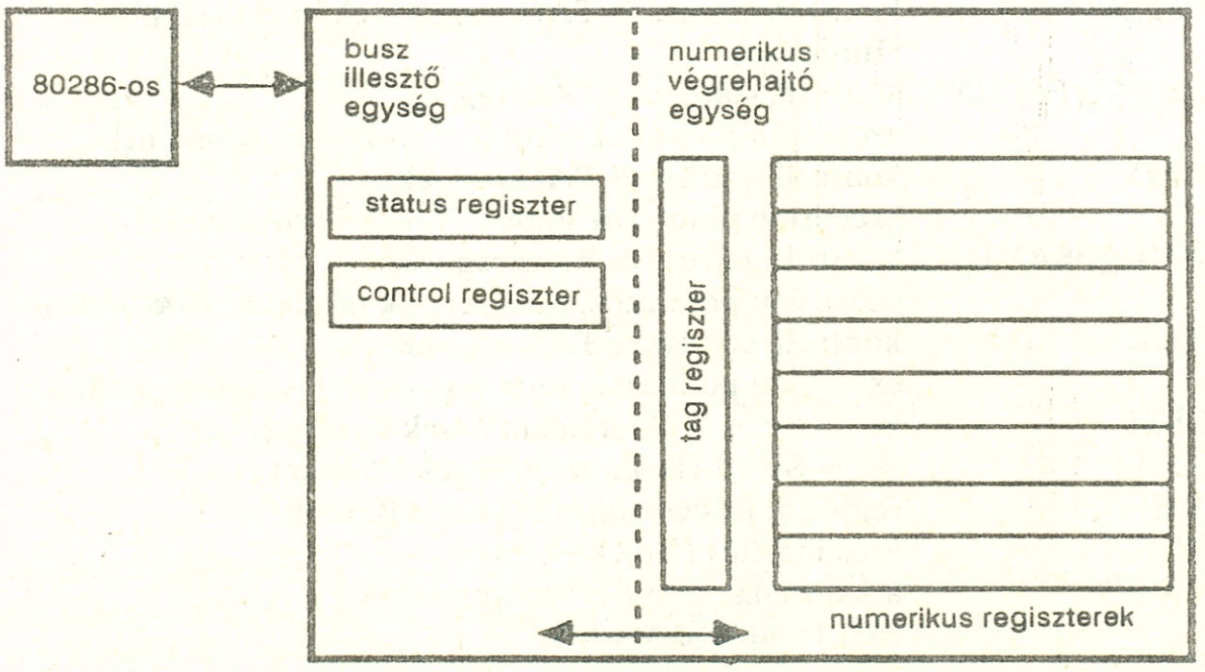
```
FIST I      ;
FWAIT       ; várakozás a 80287-esre
MOV  AX,I   ;
```

Általánosságban ez a processzor szinkronizálási probléma mindannyiszor jelentkezik, valahányszor a 80287-es olyan memória helyre hivatkozik, amelyet közvetlenül ezután a 80286-os használ. Minden ilyen esetben a probláma megoldható FWAIT beszúrásával.

Az eredmény katasztrófális lehet, ha a 80286-os nem követi az 1. szabályt, és nem vár a 80287-esre. A 80286-os elődje, a 8086-os figyelmen kívül hagyja ezt a szabályt. Ezért a 8086-os assembler csöndben és automatikusan minden 8087-es utasítás elé beszúr egy FWAIT-t. Ezek az extra FWAIT utasítások szükségtelenek a 80286-osban, ezért ezeket a 80286-os assemblere nem hozza létre.
A második processzor szinkronizációs probléma az exception-oknál jelentkezik. Ha egy

80287-es utasítás közben egy exception lép fel, a 80286-os azalatt az idő alatt, ami a hibás 80287-es utasítás megkezdése és az exception detektálása között eltelt, feltehetőleg tovább futott és már számos utasítást hajtott végre. Ezen túlmenően, a 80286-os figyelmen kívül hagyja a 80287-es exception-okat, hacsak el nem ér egy FWAIT vagy egy más 80287-es utasításhoz, ebben az esetben a 80286-os megszakítja futását. Így a megszakítás ideje alatt a 80286-os utasításszámlálója (IP) nem arra a 80287-es utasításra fog mutatni, amelyik az exception-t okozta. Azonban a 80287-es exception pointerei helyesen fogják azonosítani a hibás utasítást. Valójában ez az oka annak, hogy a 80287-es rendelkezik exception pointerekkel.

A 80286-80287 rendszerek konkurrenciájának másik forrása a 80287-esben magában található. Ahogy az 1-40 ábrán látható, a 80287-es két részből áll, a BUS Interface Unit (BIU) és a Numeric Execution Unit (NEU) (numerikus végrehajtó egység). A BIU kommunikál a külvilággal (jelen esetben a 80286-ossal), és tartalmazza a status és control regisztereket. A NEU tartalmazza a numerikus regisztereket, és végzi el az összes "piszkos" munkát. A BIU saját maga is képes bizonyos utasításokra válaszolni, annak ellenére, hogy jelen van a NEU is. Pl. a status regiszter tartalmazza azt a Busy bitet, amely jelzi, hogy a NEU foglalt vagy szabad. A 80286-os ezt a bitet figyeli az FWAIT végrehajtása közben, ill. az 1. pontnak megfelelően, ha elér egy 80287-es utasításhoz. A 80287-es FNSTSW utasítással elérhető a status regiszter. Az FNSTSW-t a BIU hajtja végre, a NEU segítsége nélkül. A 80286-os elhagyja az 1. pont szerinti várakozást, ha FNSTSW-t lát. Így az FNSTSW felhasználható a 80287-es foglaltságának vizsgálatára anélkül, hogy a 80286-os megállna, és várakozna a 80287-esre. Így egy program elindíthat egy hosszú utasítást a 80287-esen, és periodikusan figyelheti annak fejlődését.



1-40 ábra.
A 80287-es architektúrája.

A 80286-os az FNSTSW kezelésénél tanúsított 'no wait' viselkedése tipikusan érvényes a

80287 utasításoknak azon kis csoportjára, amelyeket közvetlenül a BIU hajt végre. Minden ilyen utasítás beletartozik utolsó csoportunkba, nevezetesen az adminisztratív utasítások csoportjába.

Adminisztratív utasítások

A 80287-es adminisztratív utasításai (ld. 1-21 táblázat) a 80287-es nem numerikus regisztereinek elérését szolgálják. Azok az utasítások, melyeknek mnemonic-ja FN-nel kezdődik, 'no wait' utasítások, azaz a 80286-os nem fogja ellenőrizni, hogy a 80287-es NEU-ja szabad-e, mielőtt átadja őket. Sőt, a 80286-os, ha 'no wait' utasítást talál, a numerikus exception-okat is figyelmen kívül hagyja. Az assembler biztosítja a 'no wait' utasítások 'Wait' formáját; ha a programozó elhagyja a "N"-t a 'no wait' utasítás, FN mnemonic-jából (ld. 1-21 táblázat). Az assembler az utasítás, 'wait' formáját egy FWAIT beszúrásával oldja meg. Ennek hatására a 80286-os figyelembe veszi a 80287-es exception-okat, és megvárja, amíg a NEU befejezi a 'no wait' utasítást.

Mnemonik	Művelet
FNSTCW (FSTCW)	kontroll szó tárolása (Store Control Word)
FLDCW	kontroll szó töltése (Load Control Word)
FNSTSW (FSTSW)	státusz szó tárolása (Store Status Word)
FNSTSW AX (FSTSW AX)	státusz szó az AX-ben (Store Status Word in to AX)
FNCLEX (FCLEX)	exception flag-ek törlése (Clear Exceptions)
FNINIT (FINIT)	kontroll, státusz és TAG regiszterek inicializálása (Initialize)
FNSTENV (FSTENV)	kontroll, státusz és TAG reg.-ek, exception pointerek tárolása (Store Environment)
FLDENV	kontroll, státusz és TAG reg.-ek, exception pointerek töltése (Load Environment)
FNSAVE (FSAVE)	kontroll, státusz és TAG reg.-ek, exception pointerek, num reg.-ek tárolása (Save State)
FRSTOR	kontroll, státusz és TAG reg.-ek, exception pointerek, num reg.-ek töltése (Restore State)
FINCSTP	SP = SP + 1 (Increment Stack Pointer)
FDECSTP	SP = SP - 1 (Decrement Stack Pointer)
FFREE	regiszter felszabadítása (Free a Register)
FNOP	üres utasítás (No Operation)
FSETPM	átkapcsolás virtuális védett módba (Set Protected Mode)

1-21 táblázat.
Adminisztratív utasítások.

Az FNSTCW (FSTCW) utasítás a 80287-es control regiszterét egy memória helyre tölti.

Az FLDCW betölti a control regiszterbe egy memóriahely tartalmát. Ez a két utasítás nagyon hasznos a 80287-es működési módjának megváltoztatásakor. A 80287-es garantálja, hogy ezek az utasítások végetérnek, mielőtt a 80286-os egy másik utasítás végrehajtását kezdhetné el, így ebben az esetben a korábban már említett szinkronizációs problémák nem jelentkezhetnek.

Az FNSTSW (FSTSW) utasítás a 80287-es status regiszterét tölti be egy memóriahelyre. Az FNSTSW AX (FSTSW AX) a status regisztert a 80286-os AX regiszterébe tölti. A FSTSW AX-es változatának hiánya a 8087-esben a feltételes ugrások kezelését nehézkessé teszi. Az FNSTSW és az FNSTSW AX garantáltan befejeződik, mielőtt a 80286-os új utasításba kezdhetne, így szinkronizációs problémák itt sem léphetnek fel.

Az FNCLEX (FCLEX) kitörli az összes exception flag-et, a hibaösszegző (Error Summary) flag-et és a foglaltság jelző (Busy) flag-et a status regiszterben.

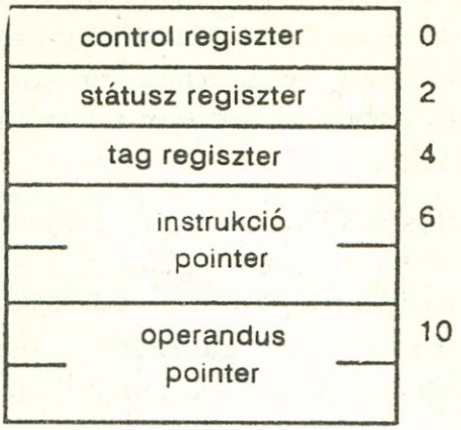
Az FNINIT (FINIT) a control, status, és tagregisztereket inicializálja az 1-22 táblázatban látható értékekkel. A hardware reset jel ugyanezeket az értékeket állítja elő.

Control regiszter	
végtelen értelmezés	projective
kerekítés	a legközelebbi ábrázolható szám felé
pontosság	kiterjesztett
valamennyi exception	maszkolt
Status regiszter	
'Busy' foglaltság-jelző bit zero	
az állapotkódok határozatlanok	
'Stack Top' mező zero	
hibaösszegző bit zero	
minden exception flag zero	
Tag regiszter	
minden 'tag' üres (empty) állapotban	

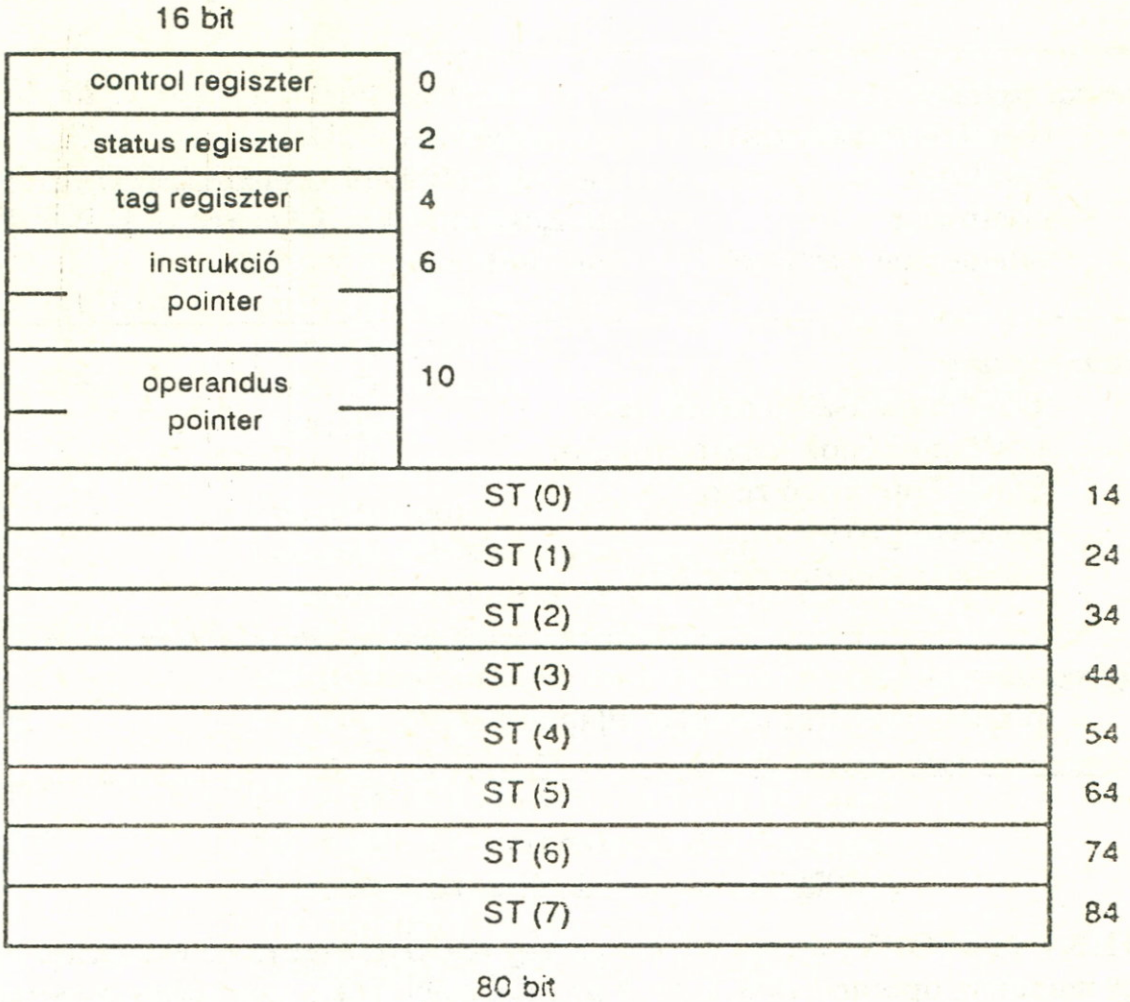
1-22 táblázat.
A 80287-es az inicializálást követően.

Az FNSTENV (FSTENV) a control, status, tag regisztereket és az exception pointereket írja be egy memória operandusba. Ennek az információnak a szerkezete a memóriában az 1-41 ábrán látható. Az FNSTENV ugyanúgy inicializálja a regisztereket, mint az FNINIT. Az FLDENV utasítás a korábban az FNSTENV-vel elmentett adatokat tölti vissza a regiszterekbe.
A FNSAVE (FSAVE) utasítás hasonlít a FNSTENV-re, azzal a különbséggel, hogy elmenti a numerikus regisztereket is. Ez 94 adatbyte betöltését eredményezi a memóriába

az 1-42 ábrán látható módon.



1-41 ábra.
Az FSTENV és FLDENV utasításokhoz tartozó memória struktúra.



1-42 ábra.
Az FSAVE és FRSTOR utasításokhoz tartozó memória struktúra.

Az FRSTOR utasítás visszaállítja az összes 80287-es regisztert egy olyan memória területről, amelyet korábban az FNSAVE-val töltöttek fel.

Az FINCSTP és az FDECSTP utasítások inkrementálják ill. dekrementálják az ST stack pointert. Nem változtatják meg a tagregisztert, és egyik numerikus regisztert sem.

Az FFREE ST(i) az ST(i) tag-jét Empty-be teszi.

Az FNOP utasításnak ugyanaz a hatása, mint a FST ST,ST(0)-nak, azaz üres utasítás (nem csinál semmit).

Az FSETPM utasítás a 80287-est virtuális üzemmódba kapcsolja.

Hardware reset után a 80287-es (éppúgy, mint a 80286-os) valós módba kerül. Az FSETPM utasítás végrehajtása után, ahhoz, hogy a 80287-es ismét valós módba kerüljön, újból hardware reset-et kell kiadni.

Ezzel a 80287-es utasításkészletének végére értünk.

2. RÉSZ

Az IBM/XT és az IBM/AT alapkiépítése.

2.1 Bevezetés

2.2 Az XT alapkártya ismertetése

2.3 Az AT alapkártya ismertetése

2.1 Bevezetés

Az Intel 8086/88 és 80286 mikroprocesszor család processzorainak és aritmetikai processzorainak áttekintése után részletesebben ismertetjük az IBM/XT és IBM/AT hardware felépítését, elsősorban az eredeti IBM dokumentáció alapján. Ez a rész az XT és AT gépek alapkiépítését tárgyalja és nem foglalkozik az I/O buszra csatlakoztatható, nagyszámú, különféle igényeket kielégítő adapter kártyákkal. Ezek közül a legalapvetőbbek az 5., 6. és 7. részekben kerülnek majd bővebb ismertetésre.

Bevezetésként "a történelmi visszatekintés" kedvéért röviden bemutatjuk az IBM által 1981-ben kibocsátott és általában IBM PC elnevezéssel jelölt személyi számítógépet, amely a későbbiekben elterjedt XT, AT, PCjr, XT 286, és az újabb PS/2-30, 50, 60 mikrogép sorozatok elődjének tekinthető. A gép első tagja volt az IBM által kibocsátott és az elmúlt években széleskörűen elterjedt személyi számítógépeknek.

Az IBM PC szerényebb lehetőségei mellett jónéhány olyan jellemzővel rendelkezik, amely megfigyelhető a többi sorozaton is.

Az IBM a személyi számítógépek tervezésekor szakított a régebbi hagyományokkal, és sem az alkatrészek, sem az operációs rendszer tekintetében nem a belső lehetőségeit használta fel. Processzor vonatkozásban a már bevált Intel 8086/88 típust választották, a PC DOS operációs rendszert pedig a Microsoft cégtől adaptálták, felhasználva ezen jólismert intézmény programozási nyelvek és az operációs rendszerek fejlesztésében már meglévő tapasztalatait.
Az igazság kedvéért érdemes megemlíteni, hogy az IBM PC tervezésekor a 8 bites mikroprocesszoros világban a *Digital Research* CP/M operációs rendszere volt az uralkodó. Az IBM eredetileg ennek 16 bites változatát, a CP/M-86 -ot szerette volna a gépein alkalmazni. Az üzleti tárgyalások azonban nem voltak eredményesek a két cég között, és ezek után fordult az IBM a *Microsoft* céghez, amely rövid idő alatt elkészítette a PC DOS operációs rendszert.

Mivel mind az építőelemek, mind a software a tervezés kezdeti fázisában hozzáférhetőek voltak, az IBM PC kifejlesztése mindössze 13 hónapot vett igénybe. Az IBM így rendkívül rövid idő alatt, viszonylag kis ráfordítással egy nagyteljesítményű, software-rel és dokumentációval jól ellátott, olcsó személyi számítógépet bocsátott ki a piacra. Az alapgondolat egyszerű volt: egy nagy számítógép jellemzőivel rendelkező mikrogépet kellett elhelyezni egy kis dobozban. A fejlesztésnél figyelembe vették az akkoriban már széleskörű személyi számítógép-gyártás eredményeit, és felhasználva az IBM-nek a világ számítógép-piacán elfoglalt helyét, tulajdonképpen egy személyi számítógép szabványt hoztak létre, melynek egyre több követője akadt.

Mivel processzorként a 8086/88 típust választották, rövidesen ez a típus vált az egyik legnépszerűbbé a mikroprocesszoros világban.
Bár a fejlesztés kezdetén már rendelkezésre álltak elegánsabb, nagyobb teljesítményű, 32 bites belső regiszterekkel ellátott processzorok, mint pl. a Motorola 68000 vagy a National 16032, a választás mégis az Intel 8088/86 típusra esett. Ennek oka az volt, hogy abban az időben ez a processzor volt a legjobban tesztelt, legmegbízhatóbb típus, és több software-rel rendelkezett, mint a többi 16 bites processzor. Jelentős előrelépésnek tekinthető a 1048576 byte-os (1 megabyte) memóriacímzési lehetőség, amely megoldotta az akkori 8

bites személyi számítógépeknél (Tandy, Apple) jelentkező max. 65536 byte címzési problémákat, és 16-szorosára növelte a címzési tartományt. Bár az 1Mbyte tartomány még mindig kevésnek bizonyult sok alkalmazásnál, az esetek többségében elegendő arra, hogy feloldja a 64 Kbyte-os címezhetőségből adódó nehézségeket.

A PC, a PCjr és az XT gépekbe a 8088-as processzor került beépítésre, amely a 16 bites 8086-os 8 bites adatbusszal rendelkező változata.
A 8086 processzor 8 MHz-es változatát találjuk a PS/2-30 jelű gépekben, míg az AT, az XT286, PS/2-50, 60 sorozatoknál a 80286 típust alkalmazták.
Az egyre nagyobb teljesítményű személyi számítógépek fejlesztését elősegítette, hogy a 8086/88 mikroprocesszor család több felülről kompatibilis taggal bővült. Ezek sorra a 80186, 80286 és 80386 típusszámokkal jelölt processzorok. A software kompatibilitás következménye, hogy az alapvető assembler rutinok hosszabb életűvé váltak, mivel az újabb típusú gépeknél is felhasználásra kerültek.

Az első IBM PC-k rendszer-komponenseit a 2-1 táblázat foglalja össze.

Az alapkártya elrendezését a funkcionális egységek megjelelölésével a 2-1 ábrán láthatjuk.
A hat darab ROM foglalatba összesen 48Kbyte-nyi ROM memória helyezhető, melyből 40 K-t használunk. (5 db 8 K-s ROM-ot)
A ROM-ok tartalmazzák az operációs rendszer BIOS-át (Basic Input-Output System) és töltőprogramját, továbbá a BASIC fordítóprogramot. A dinamikus RAM chip-ek 4 sorban helyezhetők el, minden sorban 9 darab.

Megnevezés		Ismertetés
Rendszer-egység:	Méretek:	20 inch széles (508 mm), 5,5 inch magas (140 mm), 16 inch mély (406 mm), súlya diszk meghajtók nélkül 28 pond (12,7 kg).
	Teljesítmény felvétel:	120 VA
	Processzor:	Intel 8088
	Jellemzők:	IBM billentyűzet, kazettás magnetofon csatlakozás, 5 db csatlakozási lehetőség az I/O csatornán bővítési célokból, BASIC fordító program PROM-ban, 16K RAM.
Diszk meghajtó:		Max 2 db 5 1/4" diszk meghajtó alkalmazható. A lemezek formattált kapacitása 160 vagy 320 Kbyte (egy- vagy kétoldalas lemezek)

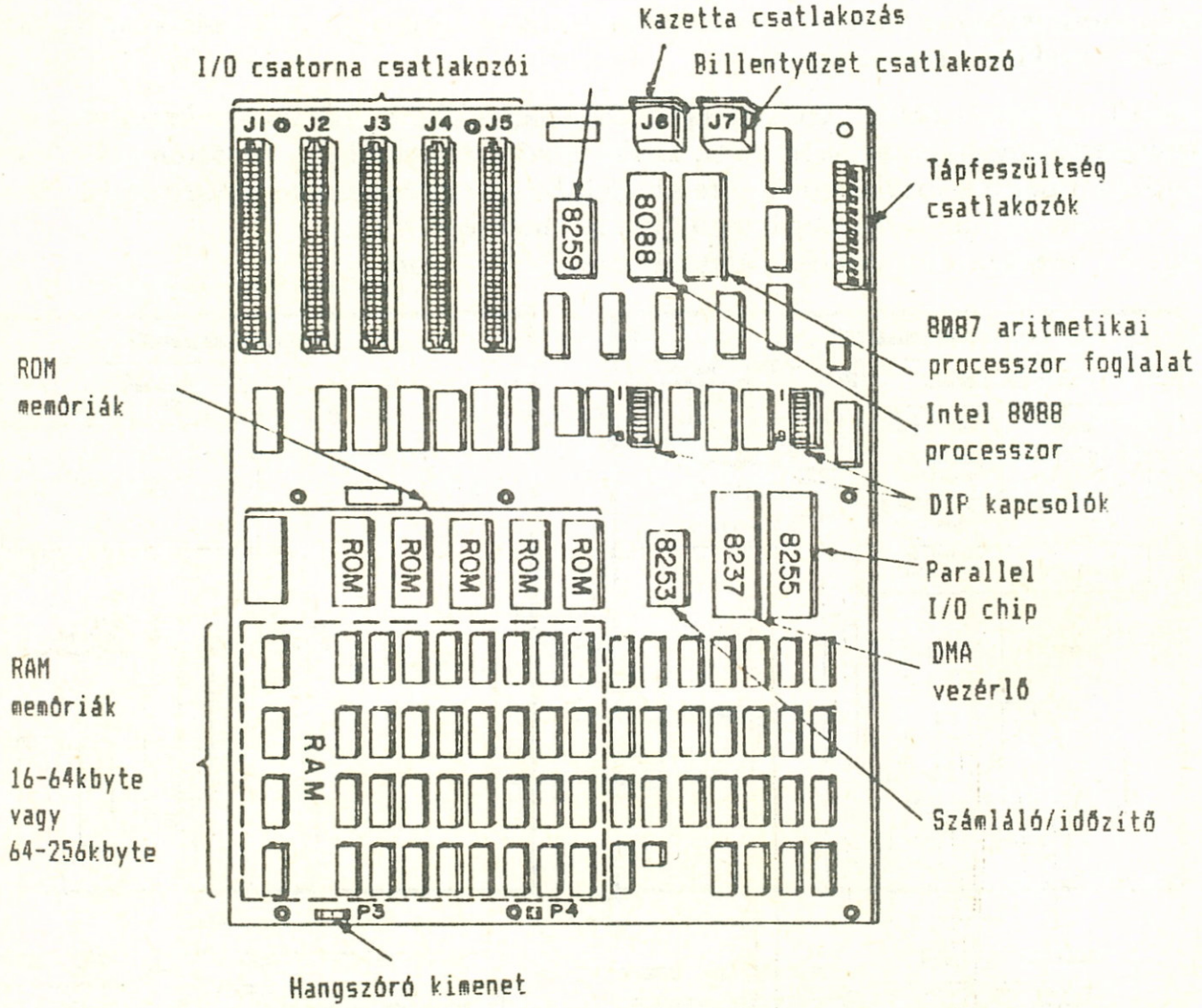
Megnevezés	Ismertetés
Operációs rendszer:	PC DOS (Microsoft)
Opcionális hardware:	Egyszínű video monitor Egyszínű monitor adaptert és nyomtató adaptert tartalmazó kártya Színes grafikus monitor adapter kártya 16K RAM bővítés 32K RAM bővítés 64K RAM bővítés Floppy diszk meghajtó adapter kártya 5 1/4 " floppy diszk Aszinkron kommunikációs adapter "Joy-stick" adapter Nyomtató adapter kártya Nyomtató kábel

2-1 táblázat.
IBM PC rendszerkomponensek.

A táblázatot áttekintve rögtön szembetűnnek a szerényebb kiindulási lehetőségek, egy ma használatos XT-vel, vagy AT-vel összehasonlítva.

A RAM memória írás műveletnél byte-onként paritás-bitet generál, amelyet a 9. memória chip-ben tárol, olvasásnál pedig ellenőrzi a tárolt és a kiolvasott byte-ból képzett paritásbit azonosságát. Ha eltérést észlel, megszakítás-kérés keletkezik, a program leáll, és hibaüzenet jelenik meg a képernyőn.
Az első PC-k 16 Kbit-es 4116 típusú dinamikus memória chipekkel kerültek kibocsátásra, majd ezeket felváltották a 64 Kbyte-os típusok.
A 8087 típusú aritmetikai processzor helyét már az első PC-k alapkártyái is tartalmazták. Az aritmetikai processzort opcióként árulták.

A bevezetőül röviden bemutatott IBM PC megjelenése egy új korszak kezdetének tekinthető a személyi számítógépek világában, amely korszak a legjobban az XT és AT gépek nagymértékű elterjedésével jellemezhető.

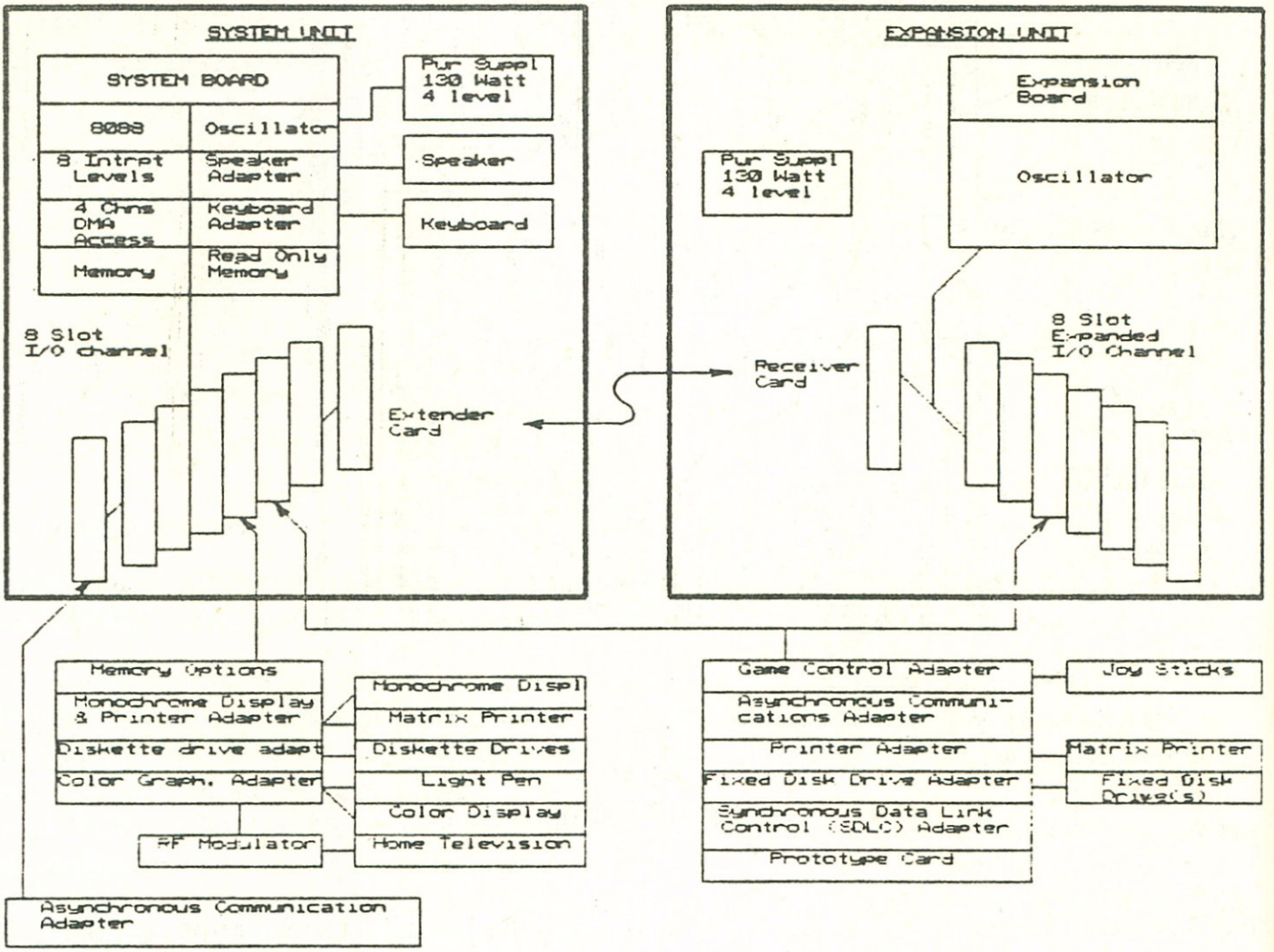


2-1 ábra.
Az IBM PC alapkártya

1983 tavaszán bocsátották ki az IBM XT személyi számítógépet, mint az IBM PC kibővített változatát. Az XT fő jellemzője, hogy beépítésre került egy 10 Mbyte kapacitású (a későbbiekben 20 Mbyte) Winchester típusú diszk meghajtó egység, egy 360 Kbyte kapacitású floppy diszk meghajtó, az alapkártyán elhelyezhető RAM memória területet 640 Kbyte-ra növelték, és a bővítésre szánt I/O csatlakozók számát 5-ről 8-ra emelték. Az operációs rendszernek egy új, bővített változatát vezették be: a DOS 2.0-t, amely egy új diszk formátumot használva (9 szektor/track, 512 byte/szektor) 360 Kbyte információ tárolását tette lehetővé egy lemezen, szemben a régebbi 320 Kbyte-tal. A DOS 2.0 felülről kompatibilis az előző DOS 1.0 és 1.1 verziókkal, és több új utasítást tartalmaz a Winchester és floppy diszkek kezelésére. A PC DOS operációs rendszer folyamatosan

bővült, több újabb verzió keletkezett, mindegyik újabb utasításokkal egészült ki, a 3.0-ás változatnál magasabb számú verziók pedig a hálózatban való működést is lehetővé teszik. Jelenleg a legutolsó verzió a PC DOS 3.3. Ezt az operációs rendszert használják a PS/2 gépeknél is. Ez a kiadvány elsősorban hardware jellegű ismertető, így a különféle PC DOS operációs rendszer változatokat részleteiben nem tárgyalja, ezek más kiadványokból és kézikönyvekből megismerhetők.

Az XT személyi számítógépek megjelenésével egyidőben kibocsátásra került egy I/O busz kiterjesztő egység (expansion unit), amely mind a PC, mind az XT esetén az I/O csatorna további 8 hellyel történő bővítésére szolgál. Ez az egység önálló tápegységgel ellátott, és Winchester és floppy diszkek beépítésére is lehetőséget ad.
Egy XT rendszer vázlatos összeállítását mutatja a 2-2 ábra.



2-2 ábra.
Az XT rendszer vázlatrajza.

Ezen általános bevezetés után részletesen ismertetjük a legjobban elterjedt XT és AT gépek alapkiépítését.

2.2 Az XT alapkártya (System board)

Az XT személyi számítógép központi része a "rendszer egység" (*system unit*), amely magába foglalja az alapkártyát, egy 150 Watt teljesítményű tápegységet, floppy és Winchester diszk meghajtó egységeket, és hátoldali csatlakozókat a monitor, billentyűzet, nyomtató, stb. részére.

Az alapkártya tartalmazza a 8088-as mikroprocesszort, 40K ROM és alapkiépítésben 128K RAM memóriát, 8 csatlakozási lehetőséget az I/O buszhoz való hozzáféréshez, valamint a billentyűzet és hangszóró illesztőket.

2.2.1 Az alapkártya leírása

A többrétegű nyomtatott áramköri lapra kivitelezett alapkártya vízszintesen helyezkedik el a rendszer egységben, közelítő méretei: 8 1/2" x 12" (216x305 mm). Az alkatrészek elhelyezkedését a kártyán a 2-3 ábra szemlélteti. A tápfeszültségek két hatpontos csatlakozón keresztül jutnak az alapkártyára. Ugyancsak erre a csatlakozóra kerül a tápfeszültségek előírt értékeit jelző, a helyes működésről informáló "Power good" elnevezésű jel.
A csatlakozó bekötése:

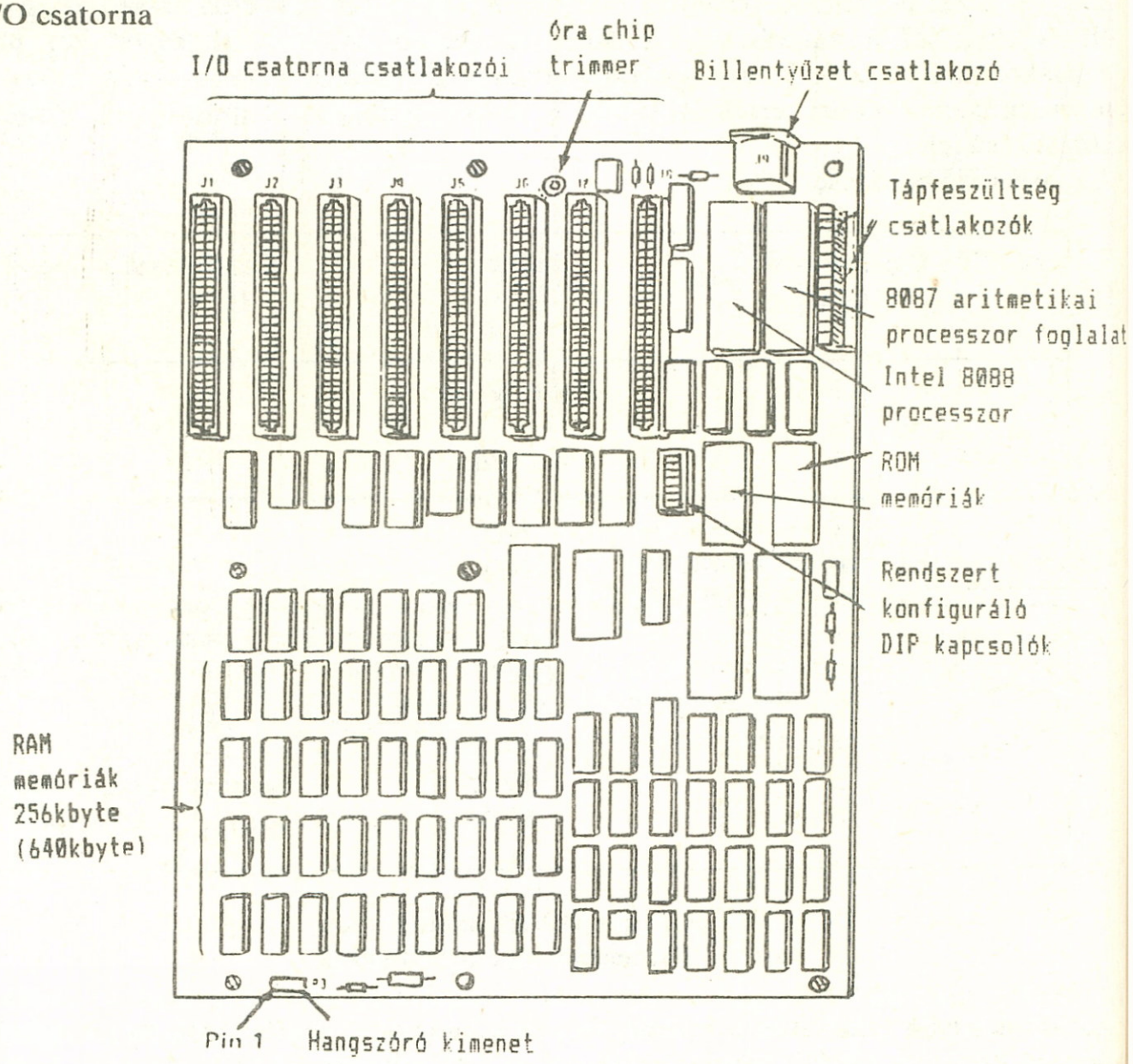
Csatlakozási pont	Jelölés	Csatlakozó
1	Power good	
2	-	
3	+12Vdc	P1
4	-12Vdc	
5	GND	
6	GND	
1	GND	
2	GND	
3	-5Vdc	P2
4	+5Vdc	
5	+5Vdc	
6	+5Vdc	

2-2 táblázat.
Tápfeszültség csatlakozások.

Az alapkártyán találhatók a billentyűzet és a hangszóró csatlakozói, továbbá 8 darab 62 pontos kétsoros (2x31) kártyacsatlakozó az I/O adapter kártyák részére. A J8 jelzésű csatlakozó némileg különbözik a többi héttől, mivel az itt elhelyezett egység "kártya

kiválasztva" (*card selected*) válaszjelet köteles kiadni az alapkártya részére, valahányszor megszólításra kerül. (I/O csatlakozó B8 pontja: CARD SLCTD)
Egy nyolc kapcsolóból álló DIP (*dual-in-line package*) kapcsolósort is találunk az alaplemezen, melynek állapotát a program beolvashatja. Ezáltal ezzel a kapcsolóval a rendszer-software-t informálhatjuk bizonyos hardware jellemzőkről, például, hogy mennyi RAM terület áll rendelkezésre az alapkártyán, milyen monitor adapter került felhasználásra, hány darab diszk meghajtót alkalmazunk, vagy milyen megjelenítési módot szeretnénk kapni a monitoron a bekapcsolás után.
Az alapkártya öt funkcionális területre osztható:

1. A 8088 processzor, és az annak működését segítő alkatrészek
2. A ROM típusú memória (csak olvasható)
3. Az R/W típusú memória (írható/olvasható)
4. Az alapkártyán elhelyezkedő I/O eszközök
5. Az I/O csatorna



2-3 ábra.
Az XT alapkártya alkatrészelrendezése.

2.2.2 Blokkdiagram

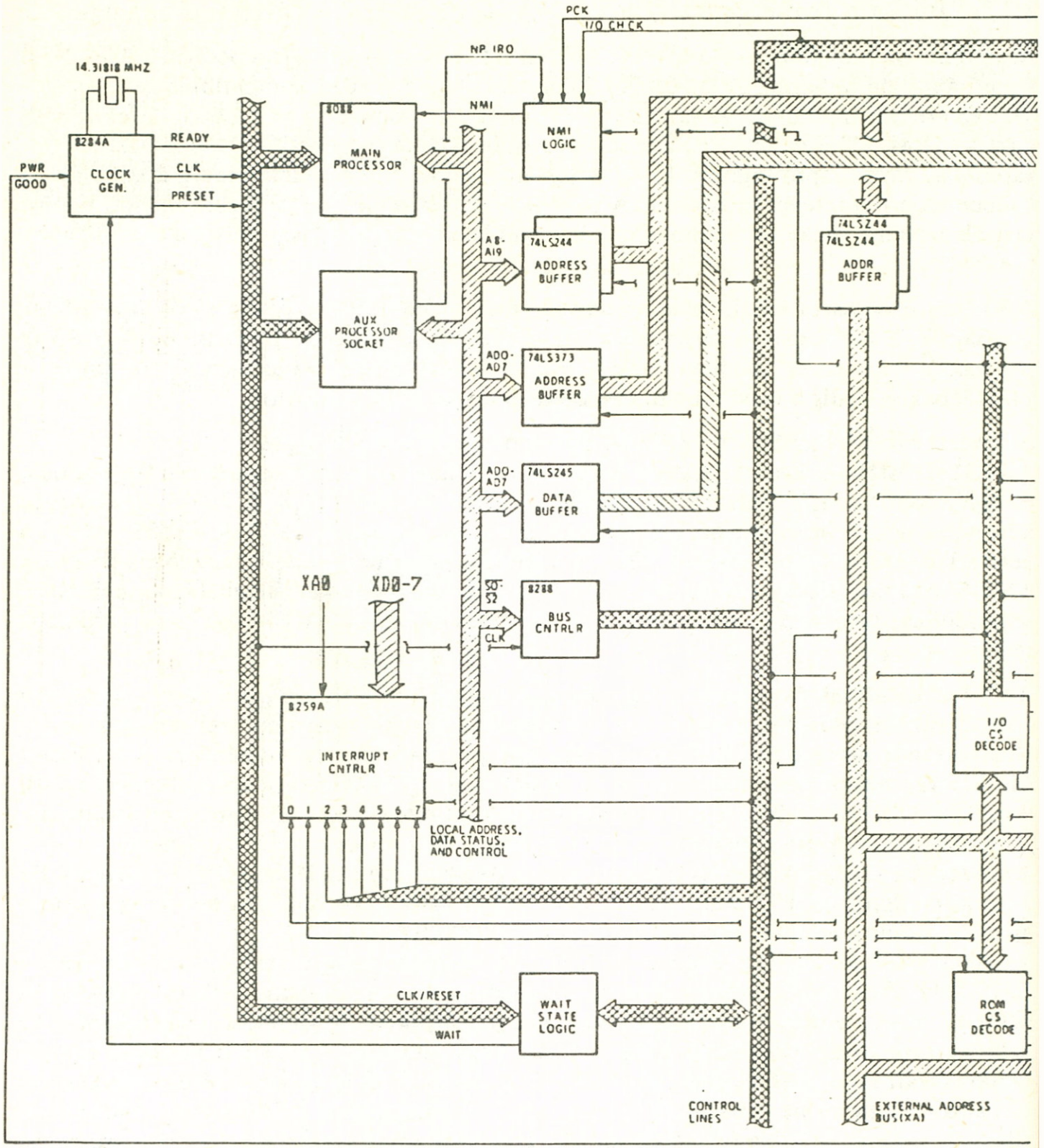
Az alapkártya blokkdiagramja a 2-4.1, 2-4.2 ábrákon, kapcsolási rajzai pedig a 2-5.1 - 2-5.6 számú ábrákon láthatók. A közölt XT kapcsolási rajz egy IBM kompatibilis gép rajza és nem az eredeti IBM gépé. A választás azért esett erre, mert ez a rajz jobban áttekinthető mint az IBM XT Technical Reference könyvben található eredeti XT kapcsolási rajz. A kapcsolási rajz 6 lapra tagozódik. Az egyes lapokon a bemenő jelek mindig a lap alján, a kimenetek a lap tetején találhatók. A bemenő- és kimenő jel megnevezése mellett jelölve vannak azok a lapszámok is, ahonnan a bemenő jelek érkeznek, vagy ahová a kimenő jelek tartanak.

A 8088-as processzor lokális buszára csatlakoznak a cím-puffer regiszterek és az adatbusz meghajtók, a 8288 típusú buszvezérlő és a 8087-es aritmetikai processzor. A 8088-as processzor "maximum" módban üzemel, azaz a MN/-MX jelű, 33-as lába földre kötött, a 24-31 lábakon pedig a maximum mód esetére érvényes jelek találhatók.

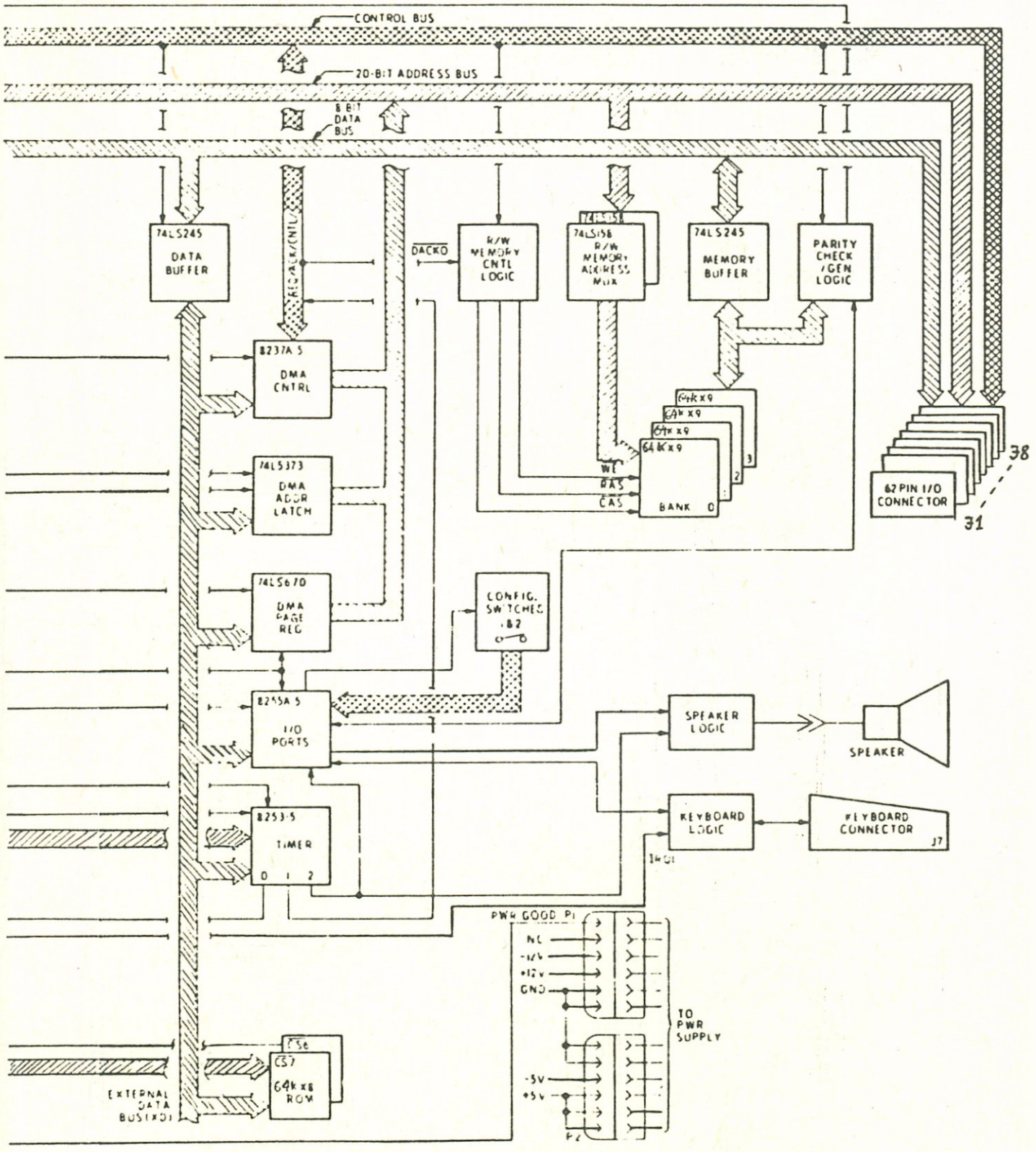
A rendszer részére az órajelet a 8284A típusú óragenerátor állítja elő.
A 14,31818 MHz-es kvarckristállyal felépített oszcillátor jelét az óragenerátor harmadolja, és így kapjuk a 4,77 MHz frekvenciájú rendszer-órajelet (CLK 8. láb). Egy további kimeneten ezen frekvenciájú jel kettővel leosztott változatát találjuk (2,385 MHz) a perifériikus I/O eszközök részére. (PCLK 2. láb). A programozható I/O eszközök javarészt NMOS technológiával készültek, és az 5 MHz-es órajel már meghaladná működési sebességüket, ezért volt szükséges egy 2,5 MHz-es órajel létrehozása ezen eszközök részére.

Az XT alapkártyák egy nagyon elterjedt változata a TURBO-640 típusú kártya, amelyben 8088-2 típusú gyorsabb processzort alkalmaznak, ezáltal nagyobb frekvenciájú órajel alkalmazható, ami a működési sebesség kb. 70%-os növekedését eredményezi. A TURBO-640 kártyák kettős óra-rendszerében az órajel vagy 4,77 MHz (normál működési mód), vagy 8 MHz (turbo mód). A két működési mód kapcsolóval vagy software úton átváltható. A TURBO-640 típusú gépekről a 2.2.10 fejezetben még röviden említést teszünk.
Az oszcillátor megerősített leválasztott jele az óragenerátor 12 lábán (OSC) nyerhető. Ez a jel közvetlenül az I/O csatorna B30 OSC nevű csatlakozó pontjára kerül, és a színes grafikus monitor adapter kártya (CGA) az elsődleges felhasználója.
A színes grafikus monitor adapter az OSC frekvenciáját néggyel osztja, és ebből nyeri a 3,579 MHz frekvenciájú szín-segédvivő burst jelet a színes televíziók részére. A 14,31818 MHz-es jelet pedig közvetlenül használja a képpont felvillantás órajeleként. Az óragenerátor további feladatai közé tartozik, hogy általános érvényű, a rendszer órához szinkronizált törlő jelet hozzon létre (RESET), és a 8088-as processzor részére várakozási ciklusokat iktasson be a READY jel segítségével, amennyiben ez szükséges. Várakozási ciklusok válhatnak szükségessé lassú I/O eszközök alkalmazása esetén, vagy memória hozzáférésű műveleteknél, amennyiben a hozzáférés az órajel által determinált normál memória vagy I/O ciklus alatt nem valósul meg.

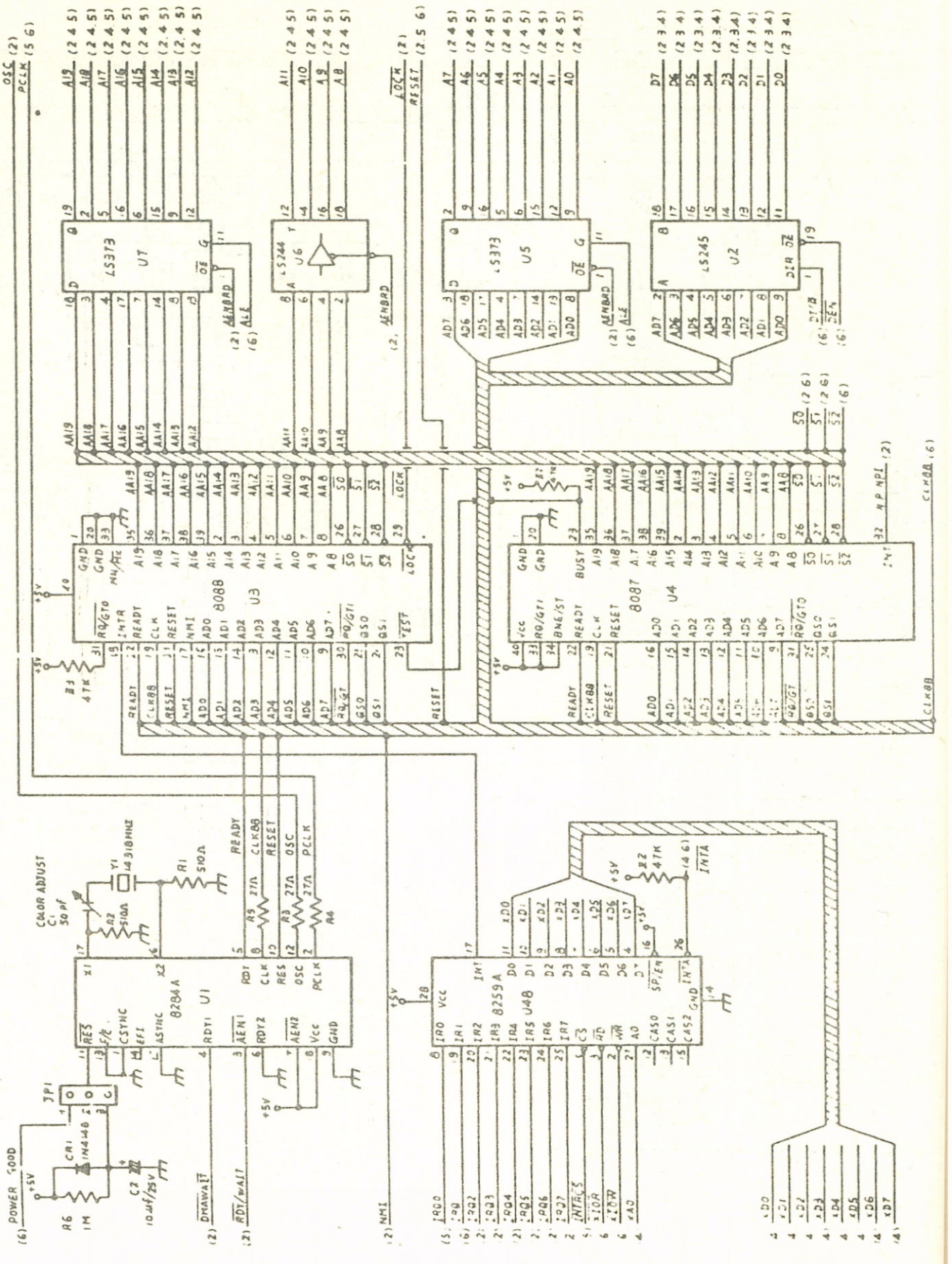
A 2-6. ábra a 8086/88 processzor lábkiosztását mutatja, és a processzor működését segítő óragenerátor , rendszervezérlő és busz-szétválasztó áramkörök kapcsolódását "maximum" módban. Ez az alapkártya magja, ami köré a többi egység épül.



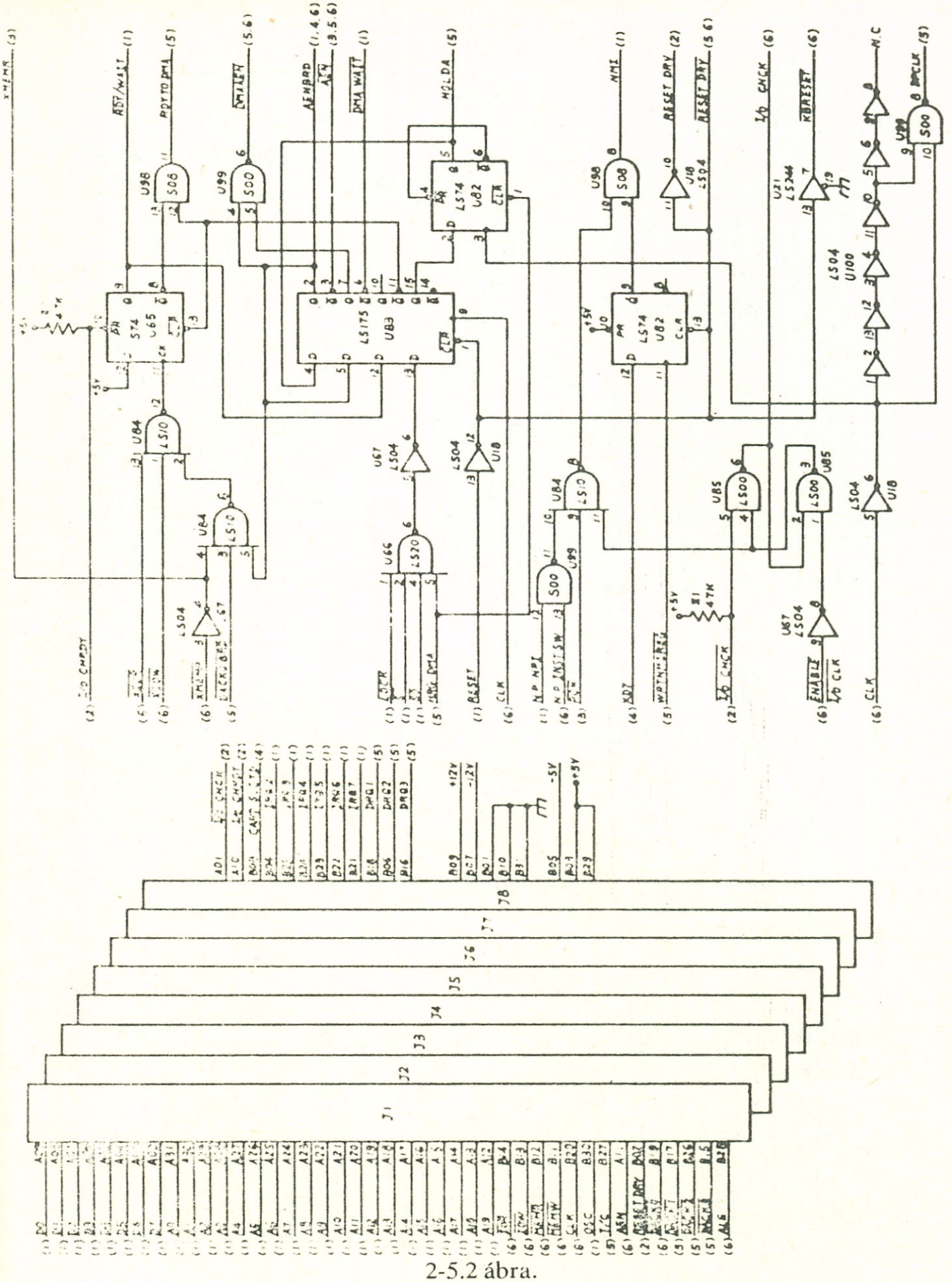
2-4.1 ábra.
Az XT alapkártya blokkdiagramja (I.).



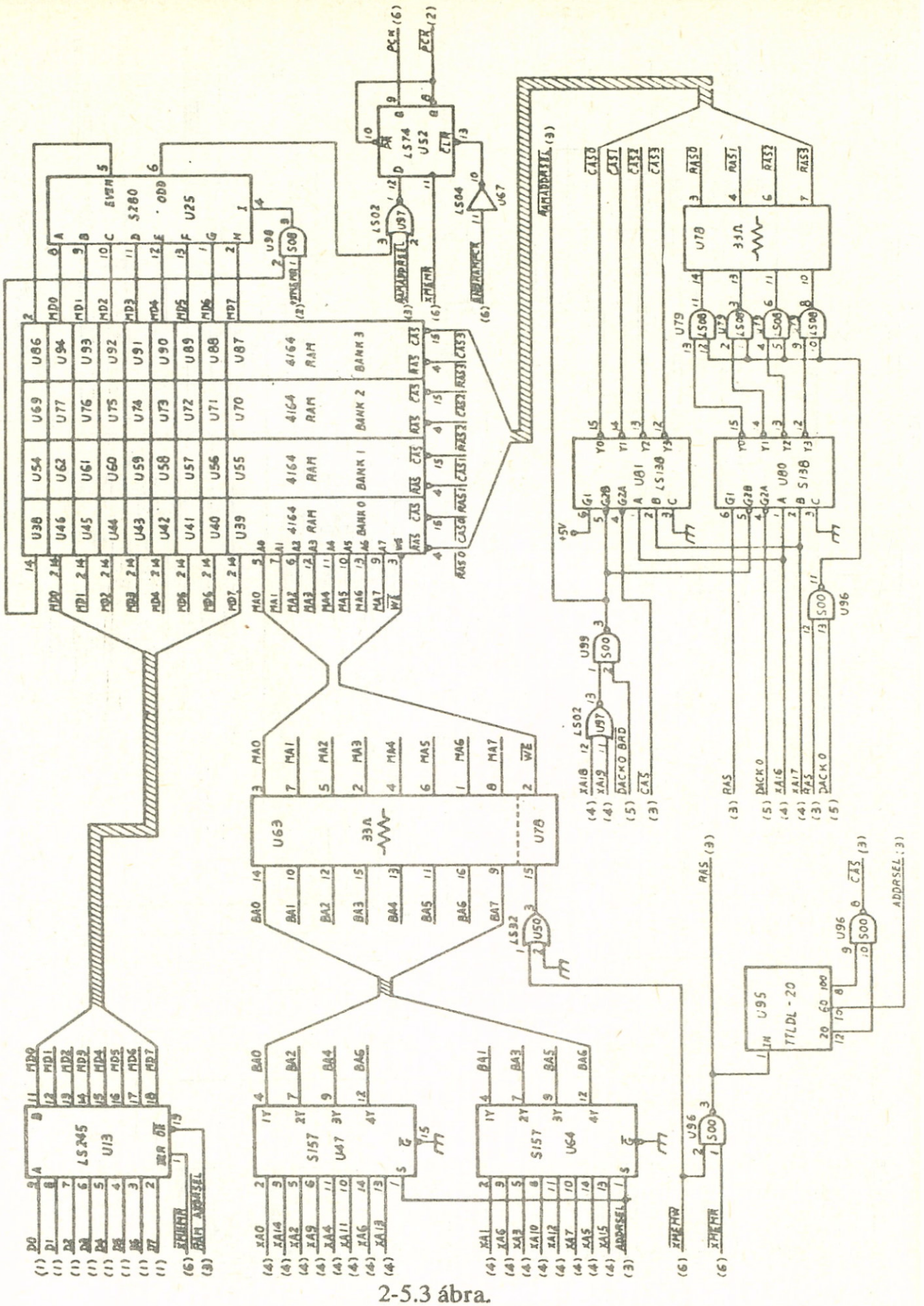
2-4.2 ábra.
Az XT alapkártya blokkdiagramja (II.).



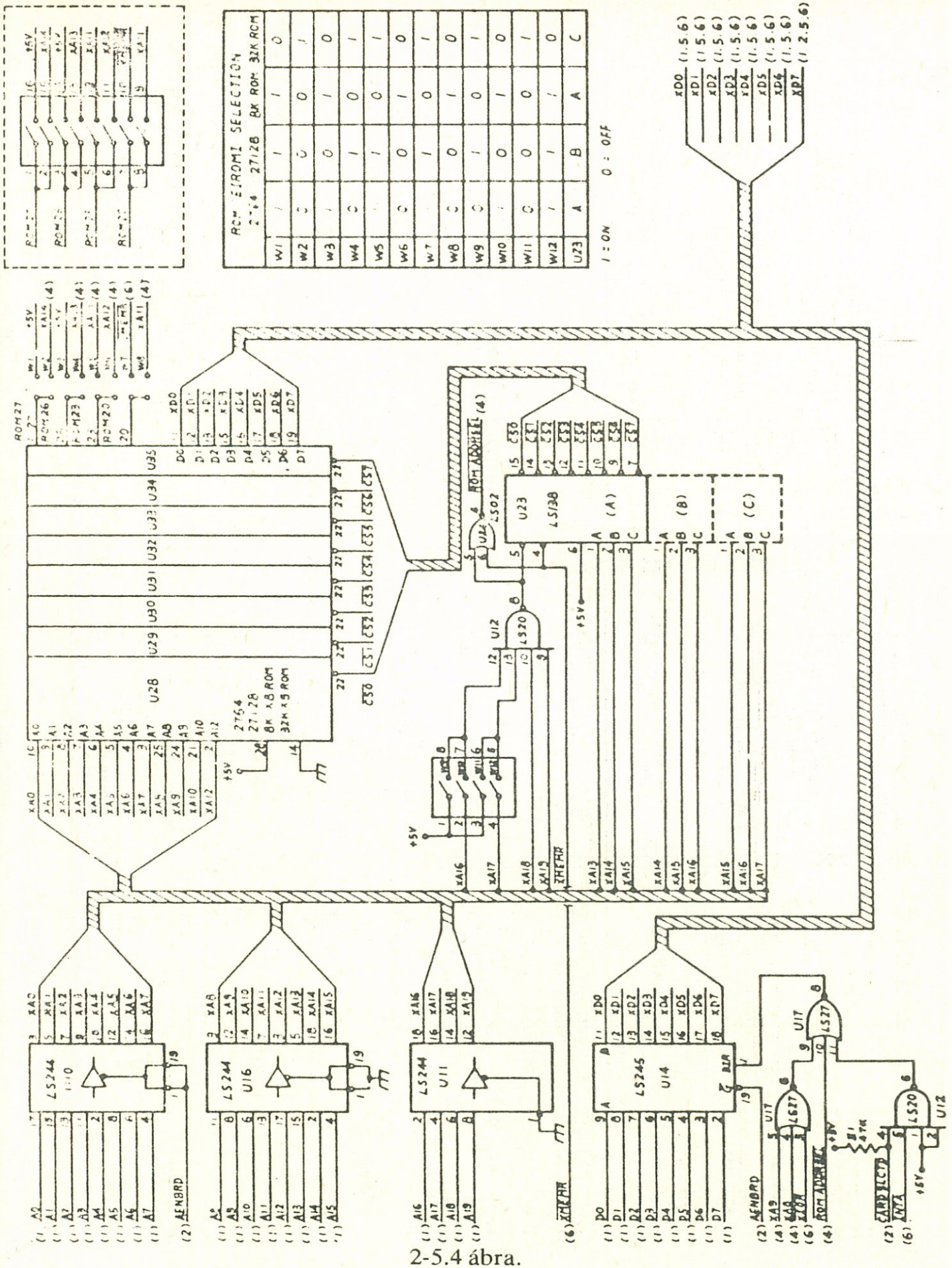
2-5.1 ábra.
Az XT alapkártya kapcsolási rajzai (1).



2-5.2 ábra.
Az XT alapkártya kapcsolási rajzai (2).

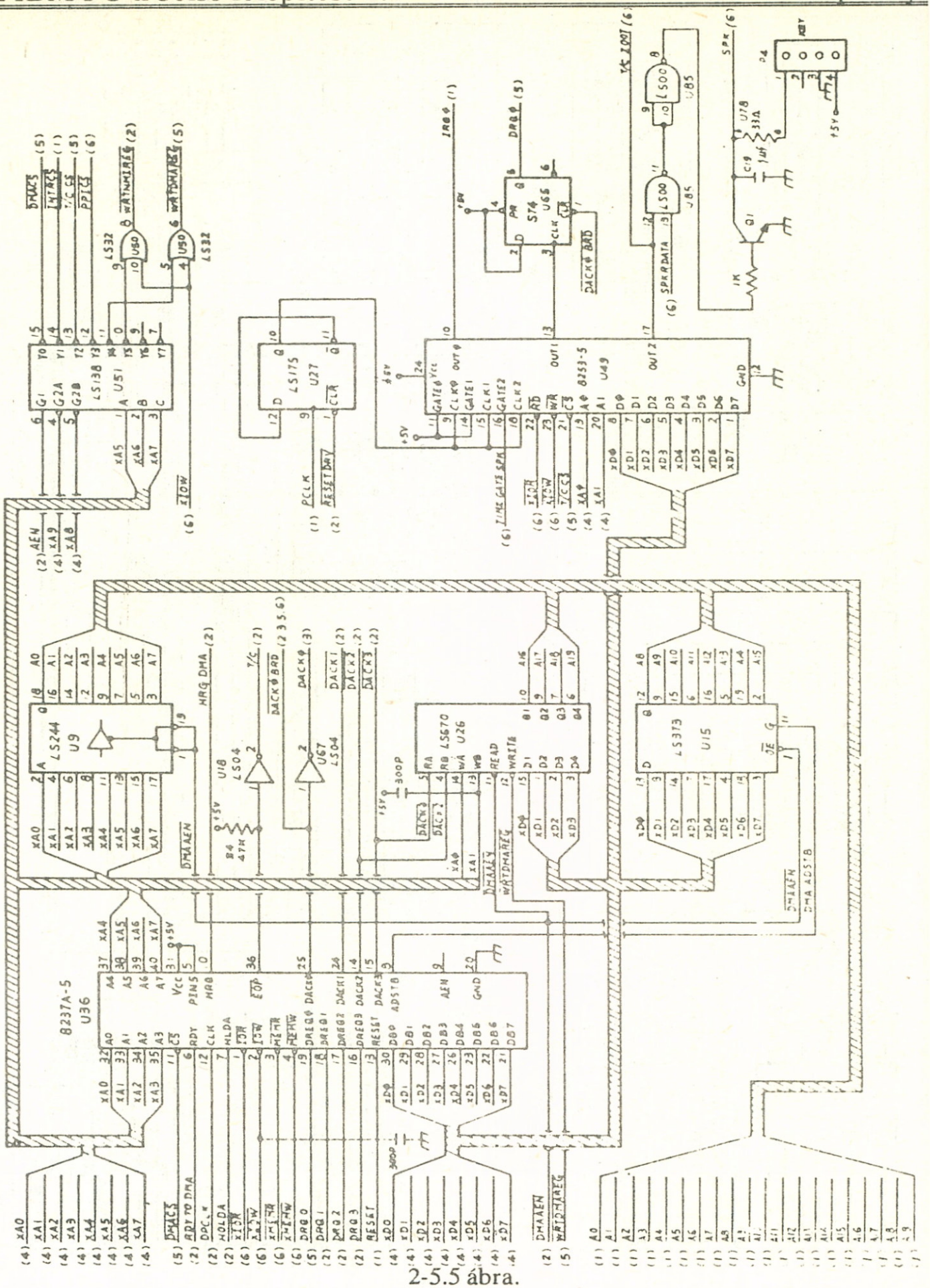


2-5.3 ábra.
Az XT alapkártya kapcsolási rajzai (3).



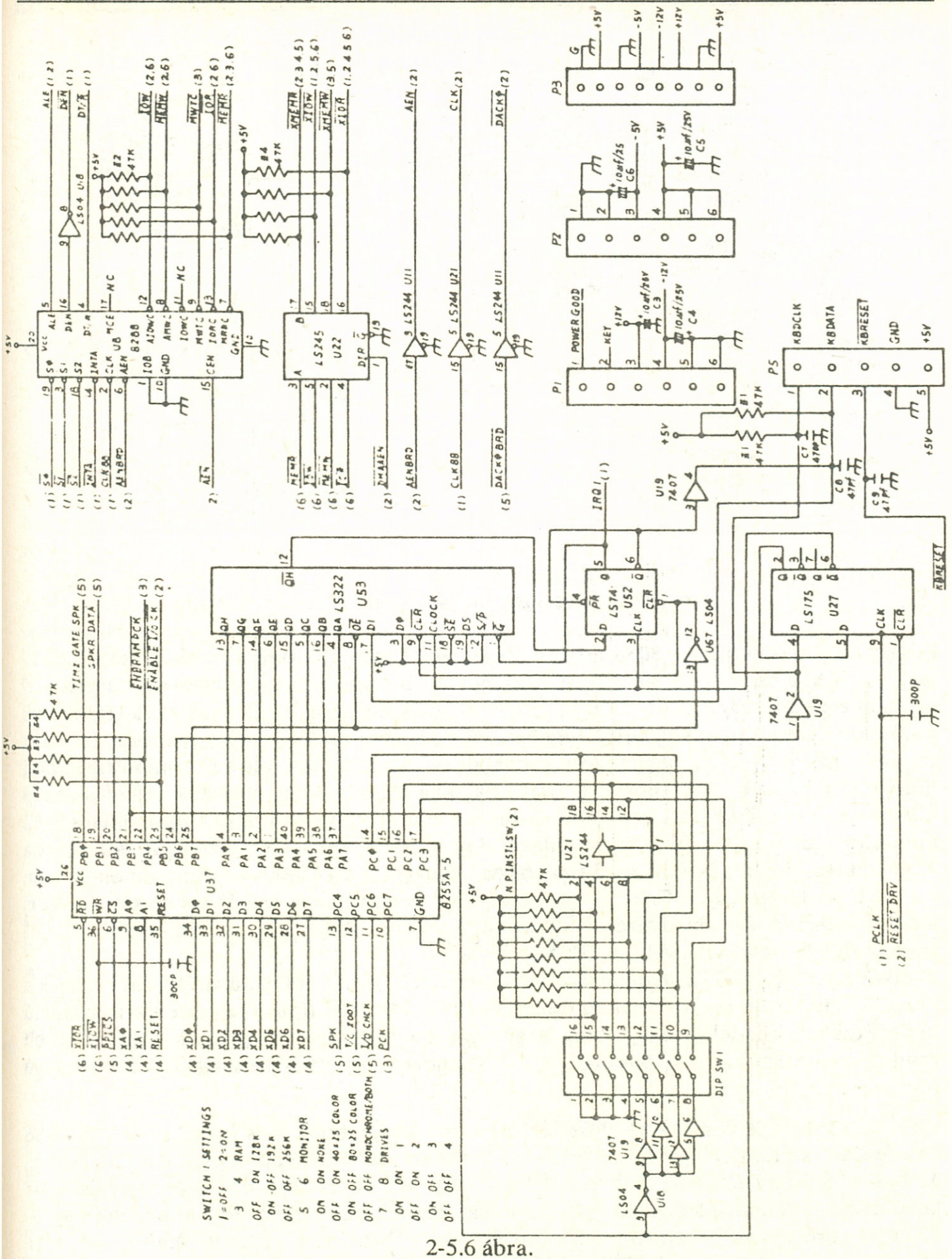
2-5.4 ábra.

Az XT alapkártya kapcsolási rajzai (4).

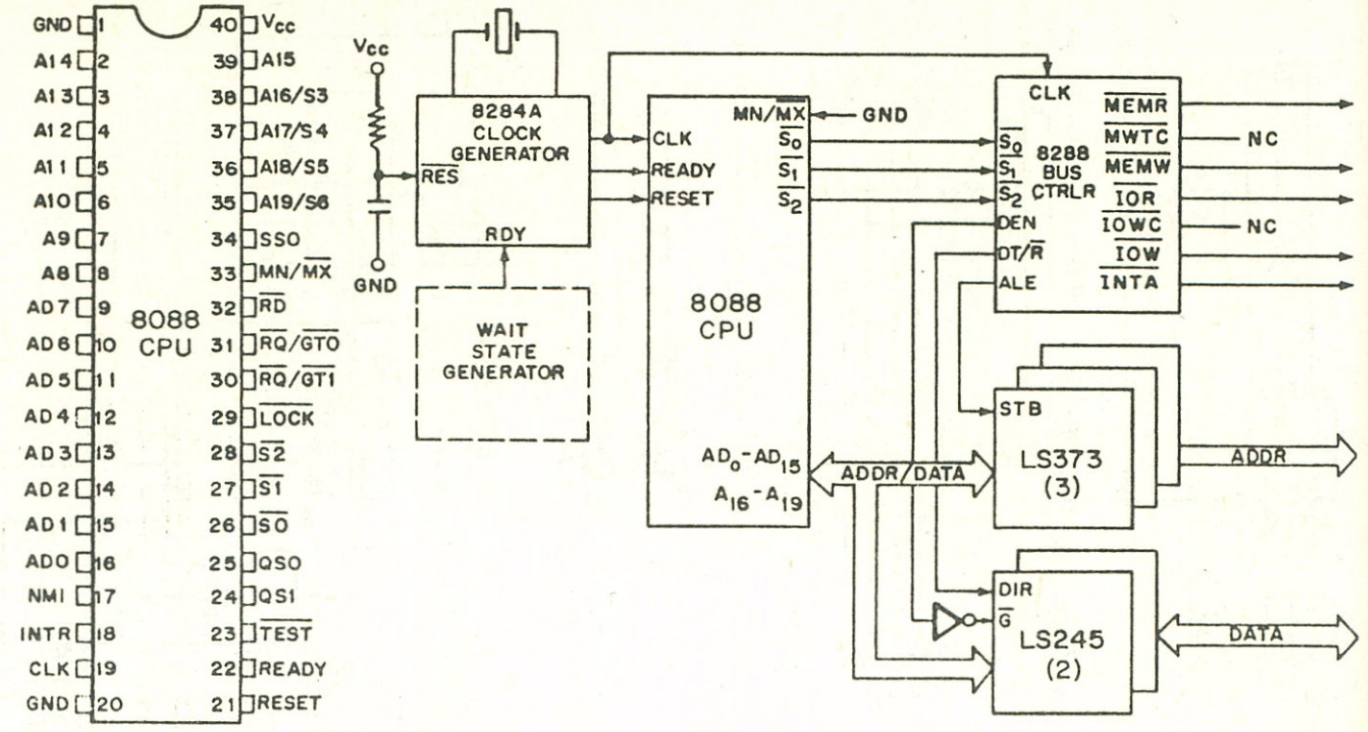


2-5.5 ábra.

Az XT alapkártya kapcsolási rajzai (5).



2-5.6 ábra.
Az XT alapkártya kapcsolási rajzai (6).



2-6 ábra.
A 8088 CPU és a működését segítő áramkörök.

Ahogy az ábrán látható, a 8088 processzor 40 lába közül 20 a címkivezetés. Ezek az AD0-AD7, A8-A15 és A16/S3-A19/S6. Ahhoz, hogy a processzor által megkívánt "háztartást" ezen a kevésnek számító 40 lábon keresztül megvalósíthassák, a 20 címvonal közül 12-nek különböző jelentést adtak, attól függően, hogy a gépi ciklus melyik szakasza az aktív. Például az AD0-AD7 elnevezésű vonalak a gépi ciklus elején címinformációkat hordoznak, majd később kétirányú adatvonalakká alakulnak át. A kétféle jelentés az idő függvénye, a vázolt eljárást pedig idő-osztásnak nevezhetjük. Az LS373 típusú oktális latch áramkörök mintavételezik a címvonalakat a gépi ciklus kezdetén, biztosítják a címeket a teljes ciklus idejére, és felszabadítják az AD vonalakat a későbbi adatforgalom számára. Az ALE (Address Latch Enable) nevű vezérlőjel mintavételezi az AD vonalakat. Az ALE jelet a 8288 típusú buszvezérlő integrált áramkör állítja elő, a 8088 által küldött státuszjelek felhasználásával (S0, S1, S2).
Amikor az ALE jel TTL logikai 1 szintről 0-ra vált, a váltás pillanatában érvényes címek a 74LS373 címtároló regiszterekben rögzítődnek. A 74LS373 típusú regiszterek nagyáramú kimenetekkel rendelkeznek, több áramkör meghajtására alkalmasak, így nagyobb rendszerek felépítésére is használhatók. (Szemben a gyakran használt 74LS244 típusú regiszterekkel.)

Az A16/S3-A19/S6 vonalak az előzőekhez hasonlóan szintén kettős jelentésűek. Az S3-S6 elnevezések státuszjelekre utalnak, ezeket azonban nem használja a gép. Az A16-A19 és az A8-A15 címértékek az AD vonalaknál leírt módon szintén egy-egy címregiszterbe kerülnek. A demultiplexelt A0-A19 vonalakat, melyek a címregiszterek kimenetei, együttesen a processzor címbuszának nevezik, és 1 byte kiválasztását segítik elő az 1 Mbyte-os címtartományból (2^{20} = 1048576). A TTL magas szint logikai 1-et, az alacsony

szint logikai 0-t jelent ezeken a vonalakon. A címvonalak közül az A0-A15 jelűek I/O műveleteknél az I/O portok címzését végzik, az A16-A19 vonalaknak ekkor nincs jelentősége.

A gépi ciklus egy későbbi szakaszában az AD vonalak kétirányú adatvonalakká válnak, azaz a 8088 processzor meghajtja ezeket a vonalakat amikor kiad egy byte-ot, és olvassa őket amikor bevesz egy byte-ot. Ugyanúgy mint a címvonalaknál, a TTL magas szint logikai 1-et, az alacsony szint pedig logikai 0-t jelent (Olyan jeleknél, ahol a TTL magas szint logikai 0-t és az alacsony szint pedig logikai 1-et jelent, ezt a jel neve elé helyezett - jellel jelöljük.).
Az adatvonalak leválasztását a kétirányú 74LS245 típusú meghajtó/vevő áramkörök végzik.

Itt jegyezzük meg, hogy a 8086 típusú processzor esetében az A8-A15 jelű kivezetések AD8-AD15 elnevezésűekké válnak, és innen választhatók le a 16 bites adatbusz D8-D15 vonalai. Így nyerhető a 8086 16 bites adatbusza, kibővítve a 8088 8 bites adatbuszát.

Vizsgáljuk ezek után a vezérlő buszt. Ez a busz mindazon vezérlőjelek összessége , melyeken keresztül a 8088 vezérli a memóriát és az I/O portokat. Ezeket a jeleket a 8288 buszvezérlő állítja elő a 8088 státuszvonalaiból (-S0, -S1, -S2). A buszvezérlő nem rendelkezik saját intelligenciával, csak egyszerűen dekódolja az -S0, -S1, -S2 vonalakon kapott jeleket. Például, ha a -S0 és a -S2 logikai 1 és a -S1 logikai 0, a 8288 logikai 0-t állít elő a -MEMR vonalon. A vezérlő busz vonalainak aktív állapota logikai 0, és valamennyi vonal csak a címbusz mintavételezése után (ALE 1-- > 0) veheti fel aktív állapotát.
A -MEMR vonalon megjelenő logikai 0 azt jelenti, hogy a processzor egy byte-ot kíván beolvasni a címbusz által meghatározott helyről, azaz a memóriát arra készteti, hogy egy byte-ot helyezzen az adatbuszra. Ha most a rendelkezésre álló memória címtartományába nem esik bele a címbusz által meghatározott érték, a processzor egy véletlenszerű, információt nem hordozó byte-ot fog kapni, mivel az adatbuszt semmi nem fogja meghajtani, és felhúzó ellenállások sincsenek rajta, amelyek az összes vonalat azonos logikai értékre állítanák.
A -MEMW (Memory Write) vonalon megjelenő logikai 0 azt jelenti, hogy a processzor az adatbuszra helyezett byte-ot a címbusz által specifikált helyre kívánja írni a memóriában.

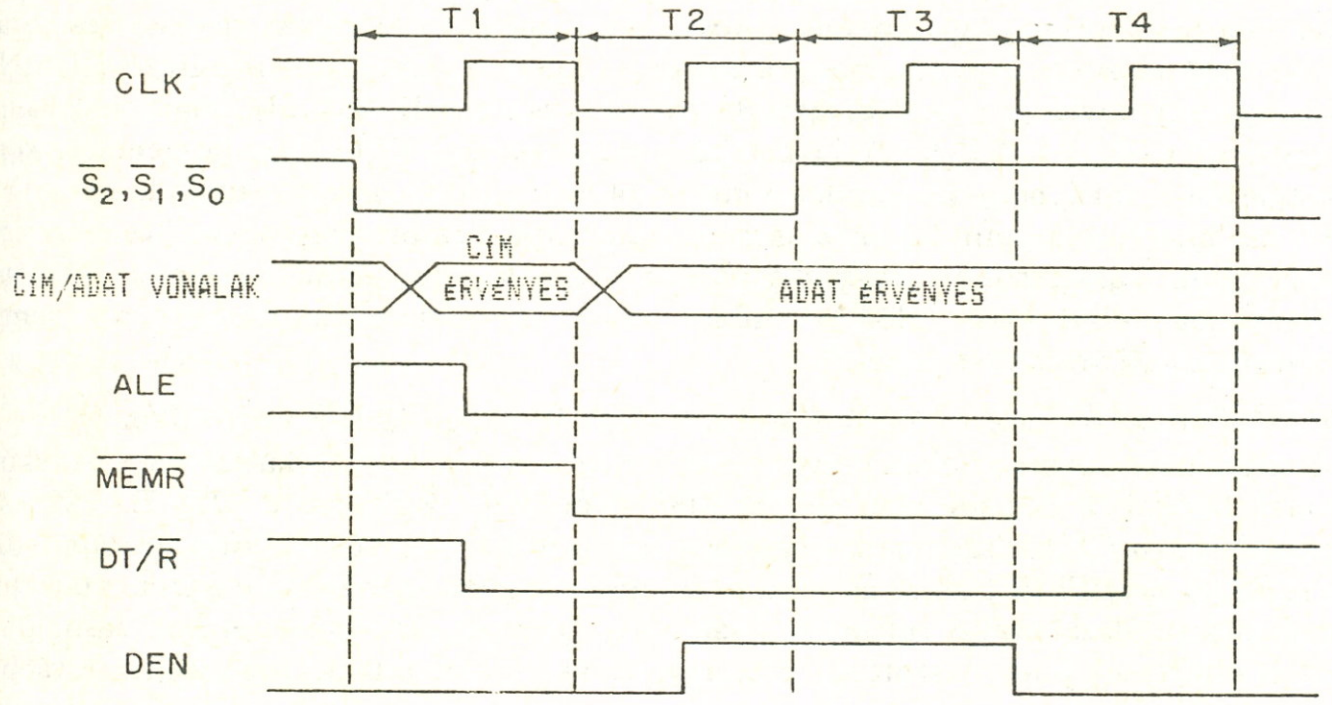
I/O műveleteknél, az I/O portok olvasásakor és írásakor a -IOR és -IOW vezérlőjeleket használjuk, amelyek a -MEMR és -MEMW jelek I/O megfelelői. I/O műveleteknél csak az A0-A15 vonalakat használjuk I/O címzésekre, ami azt jelenti, hogy maximum 65536 port címezhető.

Egy I/O eszköz kétféleképpen tervezhető: vagy úgy működjék, mint egy memória, felhasználva a -MEMR és -MEMW jeleket, vagy úgy, mint egy I/O port, az -IOR és -IOW vezérlőjeleket alkalmazva. A kétféle kivitel könnyen átváltható egyikből a másikba, ha áthuzalozzuk a -MEMR és -MEMW, valamint a -IOR és -IOW társjeleket.
Egy memóriaként használt I/O eszköz előnyökkel rendelkezhet magas szintű nyelvek, pl. PL/M vagy C használata esetén, mivel ezek a nyelvek alapvetően nem fordulhatnak I/O eszközökhöz, mivel nem rendelkeznek a BASIC INP és OUT utasításainak megfelelő parancsokkal. (Egy régebbi példaként említhető a VICTOR 9000 típusú gép, amely a PL/M nyelvet a legtöbb I/O eszközzel 'memory-mapped' módban használja.)

Ha a megszólított memória, vagy az I/O eszközök valamelyike nem tud olyan gyorsan válaszolni ahogy a CPU működik, az I/O CH READY (I/O Channel Ready) vonalat logikai 0 (not ready) állapotba kell állítani mindaddig, amíg a válasz megérkezik. Az I/O CH READY jel kerül némi módosítás után (wait state generator) a 8284 típusú óragenerátor AEN1 (3) bemenetére -RDY/WAIT elnevezéssel, melynek hatására az óragenerátor READY kimenőjele logikai 0 állapotba kerül az órajel megfelelő átváltásához szinkronizálva. A READY vonal 0 állapota a 8088-as processzort várakozásra készteti, és ez mindaddig tart, amíg a READY vonal újból logikai 1 állapotba nem kerül. A várakozás ideje alatt a processzor extra órajeleket iktat a gépi ciklusába (wait states) és semmit sem tesz ebben az állapotban, kivéve, hogy figyeli a READY vonalat a várakozásból való kilépés céljából. A leírt módszer igen hatásosan alkalmazható a CPU és az őt körülvevő eszközök közötti adatforgalom szinkronizálására, amikor ezek az eszközök nem olyan gyorsak, mint a CPU. Az egyetlen megkötés, hogy a READY vonalat 2.1 μsec-nál hosszabb ideig nem szabad logikai 0 állapotban tartani.
A -TEST vonal egy másfajta szinkronizálási technikát biztosít: miután a CPU végrehajt egy WAIT utasítást, mindaddig vár, amíg a -TEST vonal 0 állapotba nem kerül. Az IBM XT-n ez a vonal egy felhúzó ellenálláson keresztül állandóan logikai 1 állapotban van, és ha a 8087-es aritmetikai processzort nem használják a gépben (a 8087 foglalata üres), az XT végtelen hosszú várakozási állapotba kerül egy WAIT utasítás végrehajtása után, azaz a rendszer "befagy".

Amikor a CPU végrehajt egy programot, utasításokat olvas ki a memóriából, sorozatos memória-olvasási gépi ciklusokat végrehajtva. Ez a gépi ciklus a következőképpen folyik le az időben: először a címvonalakon jelenik meg az aktuális információ az AD vonalak mintavételezése után, majd a -MEMR jel vált 1-ről 0 állapotba, ahogy a gépi ciklus idődiagramja mutatja a 2-7. ábrán. Ezen az ábrán a legfelső sor mutatja az órajel változásait és a T1, T2, T3, T4 -gyel jelzett ütemeket, melyek együttesen a gépi ciklust adják, ha nincsenek beiktatott várakozási állapotok. Minden további, az ábrán látható jel az órához szinkronizált, és némi késleltetés megengedett, míg a jelek felveszik végleges értéküket. 4,77 MHz-es órafrekvencia esetén egy óraperiódus 210 nsec. TURBO alapkártya esetén ez az érték 125 nsec (8MHz). A T1 óraperiódus közepén az ALE vonal 0 értékre vált, és rögzíti a váltás pillanatában érvényes címeket a 74LS373 típusú regiszterekben. Ugyanakkor a DT/-R (Data Transmit/-Receive) vonal szintén 0 állapotot vesz fel és az adatvonalakat leválasztó kétirányú 74LS245 meghajtó/vevő áramköröket vevő irányban aktivizálja. A DT/-R vonal tehát az adatforgalom irányának kijelölésére szolgál.

A T1 óraütem végén a -MEMR jel vesz fel logikai 0 állapotot, jelezve a memóriának, hogy a CPU olvasni szeretne egy byte-ot. A T2 óraütem közepén az adatengedélyező DEN (Data ENable) vonal logikai 1 értéket vesz fel, lehetővé téve, hogy az eddig 'tri-state' állapotban lévő 74LS245 leválasztó vevőkimenetei a CPU AD vonalaira kerüljenek. A T3 óraütem végefelé a READY vonalat a 8088 mintavételezi, és ha a READY logikai 0, extra várakozási óraütemek kerülnek a gépi ciklusba beiktatásra, mindaddig, amíg ez a jel 0. A T4-es ütem kezdetén (T4 mindig a T3-at, vagy a közbeiktatott T-ket követi) a CPU beolvassa az adatokat az AD vonalakról.



2-7 ábra.
A 8088 memória olvasási ciklus idődiagramja.

A memóriába történő beírás nagyon hasonlít az olvasási ciklushoz, de itt a -MEMW vonal vezérli a műveletet az előbbi -MEMR helyett. Ezenkívül a DT/-R jel végig logikai 1 marad, továbbá, a DEN vonal a T2 ütem elején logikai 1 értéket vesz fel, és ezt a T4 ütem közepéig megtartja. Az I/O olvasás és írás ciklusok idődiagramjai megegyeznek a memória olvasás és írás diagramjaival, azzal a különbséggel, hogy a -MEMR és -MEMW vonalak szerepét a -IOR és -IOW vonalak veszik át.

A 8088 processzor egy speciális kivezetése az MN/-MX (33), amelyet az XT gépek esetén földre kötnek a "maximum mód" kiválasztása céljából. Ha erre a kivezetésre +5V-t kapcsolnak, a 8088 "minimum mód"-ba kerül, és a processzor néhány kivezetése más értelmezést kap. Az átdefiniált pontok a 8288 busz-vezérlő helyettesítésére szolgálnak kis rendszerek esetében. Ebben az üzemmódban működött a kevésbé elterjedt IBM PCjr olcsó árkategóriájú gép. A minimum mód elsődleges hátránya, hogy ebben a kiépítésben társprocesszorok nem csatlakoztathatók a rendszerhez, így a 8087-es aritmetikai processzor sem.

Maximum módban a -RD (32) kivezetés felesleges, mivel a -S0, -S1, -S2 státusz vonalak szolgáltatják a -RD-re vonatkozó információt is. Ezért a PC nem használja a -RD vonalat. Továbbá nem használja a -SS0 kivezetést sem, amely maximum módban állandóan logikai 1 állapotban található.

A még nem említett CPU kivezetések egyike az INTR (18) vonal. Az I/O eszközök a 8259A interrupt vezérlőn keresztül tudják ezt a vonalat logikai 1 állapotba hozni a megszakítás-kérés alkalmával. Ha a CPU engedélyezi az interrupt kezelést, egy indirekt

hívás aktiválja az interrupt kezelő rutint, amely bárhol elhelyezkedhet a rendelkezésre álló memóriában. Az interrupt kérésekkel, kiszolgálásokkal, az interruptok lefolyásával a 2.2.5 fejezetben részletesen foglalkozunk. Most csak a megszakításokkal kapcsolatos további két vonalról teszünk említést, melyek közül az egyik az interrupt nyugtázó jel: -INTA (interrupt acknowledge), melyet a 8288 típusú buszvezérlő állít elő, amikor valamelyi státuszvonal (-S0, -S1, -S2) logikai 0 értékű, a másik pedig a CPU 17-es bemenetére kerülő legmagasabb szintű, nem maszkolható interrupt kérés: NMI (nonmaskable interrupt). Az NMI automatikusan mindig a 2. számú interrupt vektort használja. Rendszerint a legsúlyosabb, katasztrófális meghibásodások hoznak létre ilyen kérelmet, például a tápfeszültség ellátásban előálló hiba, vagy memóriahiba, melyet a memória paritásellenőrző áramköre generál.

A RESET vonal a rendszer újraindítására szolgál. Bekapcsoláskor, vagy valamely program hibás működése következtében előállt rendszer-hiba esetén kerül alkalmazásra. Amikor ez a vonal logikai 1 értéket vesz fel, a 8088 processzor nullázza a DS, ES, SS és IP regisztereket, a CS kódszegmens regisztert pedig 0FFFFh értékre állítja. Az újraindítási cím (CS:IP), így 0FFFFF0h-ra adódik. A számítógép memóriában ezen a címen találjuk az inicializáló rendszerindító rutint. A lejátszódó folyamat egy interrupt-hoz hasonlítható, azzal az eltéréssel, hogy visszatérési cím nem kerül a verem (stack) regiszterbe, továbbá az interrupt-tól eltérően az előbbiekben említett regiszterek mindig kezdőértékeket vesznek fel.

A legtöbb számítógépen találunk egy nyomógombot, amivel újraindíthatjuk a rendszert a RESET vonalon keresztül. Az IBM XT-ben azonban nyomógomb helyett a tápegység POWER GOOD jele végzi az említett műveletet. Ha a tápegység helyesen működik, ez a jel logikai 1. Bekapcsoláskor, vagy a tápellátásban előálló hiba esetén ez a jel logikai 0 állapotból indul, illetve 0 állapotra vált, ami a rendszer inicializálását vonja maga után. Ez a megoldás azt is jelenti, hogy rendszer-lefagyáskor (például egy új program kipróbálásakor) a gépet néhány másodpercre ki kell kapcsolnunk, majd újraindítanunk. Ha a meghibásodás nem sértette meg az interrupt vektor táblát, és az interruptot engedélyező állapot fennáll, a Ctrl-Alt-Del billentyűk egyidejű lenyomása is újraindítást eredményez. Néhány XT típusú személyi számítógépet gyártó cég eltért ettől a nem túl kényelmes megoldástól, és egy nyomógombot épített a gépbe, amely a régebbeni megszokott hardware újraindítás-ra szolgál.

Korábban már említettük, hogy a CPU várakozási állapotba kerül, ha READY vonalát logikai 0 állapotban tartjuk. A közvetlen memóriahozzáférés néven ismert (DMA = Direct Memory Access) nagysebességű adatátviteli módnál a READY vonal szintén felhasználásra kerül. DMA módban egy processzor-szerű integrált áramkör, a DMA vezérlő felfüggeszti a 8088 CPU működését és átveszi a cím- és adatbuszok vezérlését, felhasználva a saját maga által előállított címeket és vezérlőjeleket. Az adatátvitel sebessége elérheti az 1,5 Mbyte/sec értéket. A DMA vezérlő hardware kialakítása és a rendszer arhitektúrája teszi lehetővé ennek a jelentős átviteli sebességnek az elérését. Az XT gépbe épített meglehetősen bonyolult szinkronizáló áramkör biztosítja, hogy a 8237 típusú DMA vezérlő csak a gépi ciklus végén veheti át a buszok vezérlését, de csak abban az esetben, ha a -LOCK vonal nincs aktív 0 állapotban. A -LOCK vonalat egy LOCK prefix-szel ellátott utasítás aktivizálhatja. A -LOCK vonal logikai 0 értékű marad a

utasítás teljes időtartama alatt. Ha a hardware kialakítása hibátlan, más processzor által kért memória hozzáférés a -LOCK vonal 0 állapota alatt nem kerül kiszolgálásra. A -LOCK vonal használatára egy egyszerű példa az XCHG assembler utasítás alkalmazása, amikor az egyik operandus a memóriában található, a másik pedig valamelyik regiszterben. Az operandusok cseréje mindenféleképpen két memória ciklust igénylő művelet, függetlenül attól, hogy az utasítás egyetlen szó. Kritikus esetekben előfordulhat, hogy az utasítás végrehajtásának közepén, az első memóriaciklus lezajlása után DMA igény érkezik, és az XCHG utasítás hibásan fejeződik be. Ezért célszerű az XCHG utasítást LOCK prefix-szel ellátni, amely az utasítás teljes idejére tiltja a másik processzor hozzáférését a buszhoz.
A DMA vezérlő aktív állapotában várakozási ciklusokat kényszerít a processzorra a 8284 READY vonalán keresztül, a 8088 processzor cím, adat és vezérlő kimenetei pedig tri-state állapotba kerülnek.
A DMA működéséről részletesebb információ található a 2.2.8 fejezetben.

A processzor lábkivezetések egy utolsó csoportja még nem került ismertetésre. Ezek a -RQ/-GT0, -RQ/-GT1, QS0 és QS1 elnevezésű pontok. Ezek a vonalak koordinálják azon "társprocesszorok" működését, amelyek a multiplexelt AD buszokra kapcsolódnak. Kétféle társprocesszor típust különböztethetünk meg:

- Az első csoportba azok tartoznak, amelyek saját utasításkészlettel rendelkeznek, és a busz-hozzáférés során önállóan hajtják végre műveleteiket. Erre a típusra példa egy további 8086/8088 "társprocesszor" vagy egy 8089 I/O processzor, melyek az AD buszt használják.
- A második csoportba azok a társprocesszorok tartoznak, amelyek csak kibővítik a 8086/8088 utasításkészletét az ESC (escape) utasítás felhasználásával. Erre tipikus példa a 8087 típusú aritmetikai processzor alkalmazása az IBM XT gépnél. A 8087 aritmetikai processzor alkalmazása jelentősen lecsökkenti a matematikai számítások idejét.

A 8087-es társprocesszor nem rendelkezik a 8086/8088 címaritmetikájával, így a 8087 műveletek során a 8086/8088 CPU biztosítja a kezdőcímeket. A folyamat a következőképpen zajlik le:

A program végrehajtása közben elérünk egy olyan utasításhoz, amely az aritmetikai processzor részvételét igényli. (Itt jegyezzük meg, hogy valamennyi 8087 utasítás assembler mnemonikus kódja F betűvel kezdődik.)
A 8087 társprocesszor folyamatosan figyeli a CPU által végrehajtandó utasításokat, de semmit sem tesz mindaddig, amíg nem azonosít egy ESC utasítást. Az ESC utasítást úgy tekinthetjük, mint a CPU figyelemfelhívását, amivel közli, hogy a soron következő művelet(ek)hez segítségre van szüksége. Az ESC utasítás a kétbyte-os utasítások osztályába tartozik. Az utasítás gépi kódja 11011XXX modXXXr/m alakú, ahol az X-ek 0 vagy 1 értékűek lehetnek, kombinációik a megvalósítandó műveletekre utalnak. Az aritmetikai processzor utasításainak egy része memória hozzáférést igényel. Ha a program végrehajtása során egy ilyen utasítás kerül sorra, a 8086/8088 CPU előállítja az utasításban megadott effektív címet, ezt az időmultiplexelt AD buszra helyezi, és egy memória olvasási ciklust kezdeményez. A 8087 processzor ekkor aktív állapotba kerül, beolvassa a

memóriából jövő adatokat, és befejezi a CPU által indított utasítást. A CPU ezalatt elvégezheti a következő utasítás(oka)t, vagy várakozhat, míg a 8087 befejezi a részére kiadott feladatot. Ha ez utóbbi eset áll fenn, az ESCape utasítást egy WAIT utasítás fogja követni. A WAIT várakozási utasítás mindaddig felfüggeszti a CPU tevékenységét, míg az aritmetikai processzor választ nem küld a CPU -TEST bemenetére, amivel jelzi, hogy befejezte az utasítást. A várakozási állapot alatt a CPU figyeli a -TEST vonalat, és amikor az logikai 0 állapotot vesz fel, a program folytatódik. A CPU -TEST bemenete az aritmetikai processzor BUSY foglaltsági kimenetéhez csatlakozik.

Az aritmetikai processzor utasításkészletének definiálására makrókat készítettek. Ezek a makrók tartalmazzák az ESCape és WAIT utasításokat. Az ESC utasítás elé is kerül egy WAIT utasítás, ami megakadályozza, hogy az aritmetikai processzor egy új műveletbe kezdjen azalatt, míg az előző feladatát be nem fejezte. Megállapíthatjuk tehát, hogy az ESC és WAIT utasítások együttesen biztosítják a szinkronizmust a 8088 CPU és a 8087 aritmetikai processzor között.
A kétbyte-os ESC utasítás "mod" mezője 2 bites, az "r/m" mezője pedig 3 bites. Ismeretes, hogy ez a két mező határozza meg a címzési módot. Ha a "mod" mező 11(bin) értékű, az utasítás nem memória vonatkozású. Ebben az esetben a CPU az ESC utasítást egyszerűen figyelmen kívül hagyja, tehát nem képez címet és memória olvasási ciklust, az aritmetikai processzor viszont végrehajtja a számára kijelölt, memória hozzáférést nem igénylő utasítást.

Az aritmetikai processzorok alkalmazása a PC-kben nagy jelentőségű. A magasszintű nyelvek fordítóprogramjainak nagy része megengedi az aritmetikai processzorok használatát. Általában a compiler indításához használt utasításszavak után elhelyezett kulcsokkal adhatunk parancsot arra, hogy a fordítóprogram vegye figyelembe a 8087 aritmetikai processzor jelenlétét és a lefordított program használja is azt, vagy a megfelelő könyvtári rutinokat hívja az aritmetikai utasítások emulálására.
A FORTRAN-77, TURBOPASCAL, C, PLM/86, MASM (macro assembler) fordítóprogramokat említhetjük mint gyakoribb példákat a 8087 használatára.

A 8086/88 processzorokkal kapcsolatban végül meg kell említeni a QS0 és QS1 elnevezésű kivezetéseket, amelyek arról informálnak, hogy a processzor a soron következő utasítást a belső, négy byte mélységű FIFO-jából (Firts-in first-out buffer), vagy a külső memóriából veszi. Információt szolgáltatnak arról is, hogy a FIFO-ban levő utasítás hányadik byte-ja kerül feldolgozásra a processzorban. A QS0 és QS1 kivezetések jeleit az aritmetikai processzor használja a -S0, -S1 és -S2 jelekkel egyetemben az ESC utasítás helyes fázisban való azonosítására.

Célszerű itt ismertetni az előbb említett FIFO szerepét: a 8086/88 processzorokba épített ezen 4 byte-os hardware egység a program végrehajtását gyorsítja. Egy busz interface állandóan tölti a FIFO-t a program számláló (IP) által mutatott új értékeknek megfelelően, a CPU műveleteitől függetlenül. A CPU végrehajtó egysége egy-egy művelet befejezése után azonnal elkezdheti a soron következő új utasítást és nem kell várnia a memória hozzáférési időre. Az egyetlen kivételt az ugró utasítások adják, amelyek megszakítják az egymás utáni utasítások sorrendjét. Abban az esetben, ha az éppen feldolgozott utasítás ugró típusú, a FIFO-t ki kell üríteni és fel kell tölteni az új utasítással. Ez egy memória hozzáférésnél hosszabb időt jelent, továbbá minél nagyobb

mélységű a FIFO, ez az idő annál nagyobb. Viszont egy programban az ugró utasítások általában ritkábban fordulnak elő, így a FIFO alkalmazása jelentősen megnöveli a végrehajtási sebességet.

A QS0 és QS1 jelek állapotai és a FIFO státusza közötti összefüggések:

QS1	QS0	FIFO státusz
0	0	nincs működés
0	1	a FIFO kimenete az utasítás első byte-ja
1	0	a FIFO üres
1	1	a FIFO kimenete az utasítás későbbi byte-ja

2.2.3 Az alapkártyára épített memóriák.

Ebben a fejezetben ismertetésre kerülnek az XT gép alaplemezén elhelyezkedő ROM és RAM típusú memóriák, a memória címtérképek, és a memóriák működése. Mint ismeretes az XT típusú gépek 8088-as processzora 1 Mbyte memóriaterületen belül tud előállítani valós címeket egy-egy memória rekesz tartalmának az elérésére. A memória címtartományok elhelyezkedését az XT gépeknél a 2-3 számú táblázat mutatja. Egy-egy címtartomány szigorúan előírt célokra használható, amit valamennyi gyártó pontosan betart. Egy XT gép maximálisan 640 kbyte RAM memóriával rendelkezhet, a 0-9FFFFh címtartományban, az alaprendszer ROM tartománya pedig a 64 kbyte nagyságú 0F0000h-0FFFFFh címtartományba esik.

CÍM [hex]	FUNKCIÓ
0-3FF	Interrupt vektorok (lásd 2-6 táblázat)
400-47F	BIOS adatterület (keyboard, disk, nyomtató, video vezérlő adatai, üzemmódjai, paraméterei)
480-5FF	BASIC és DOS hívások munkaterülete
600-9FFFF	Programok számára kijelölt memóriaterület
0A0000-0AFFFF	Video bővítési célokra fenntartott terület (EGA)
0B0000-0B7FFF	Video adatterület egyszínű grafikus adapter esetén
0B8000-0BFFFF	Video adatterület színes grafikus adapter esetén (CGA)
0C0000-0EFFFF	I/O adaptereken lévő ROM-ok számára fenntartott terület
0F0000-0F3FFF	Rendszer ROM terület
0F4000-0F5FFF	Rendszer ROM terület
0F6000-0FDFFF	Rendszer ROM terület (BASIC)
0FE000-0FFFFF	Rendszer ROM terület BIOS

2-3 táblázat.
Az IBM XT 1 Mbyte-os memória címtérképe.

A rendelkezésre álló RAM terület alsó egy kbyte-ját a négy byte terjedelmű megszakítás-kérés vektorok foglalják el, összesen 256 darab. Ezt követi egy 128 byte hosszúságú rész,

melyet a ROM BIOS, majd egy 384 byte hosszúságú rész, melyet a DOS és a BASIC használ munkaterületként. Ezeken a helyeken olyan adatokat tárolnak, amelyek a különféle I/O eszközök állapotainak azonosítására szolgálnak. Például itt kerül rögzítésre, hogy a display vezérlő grafikus- vagy szöveg-módban üzemel-e, vagy a billentyűzet puffer regisztere üres-e vagy billentyű kódokat tartalmaz. A 600h-9FFFFh címtartomány a programok részére kijelölt hely. Ennek alsó részét rendszerint a DOS operációs rendszer foglalja el, amely a felhasználói programokat közvetlenül maga fölé tölti, amennyiben külön előírások nincsenek a program betöltésére vonatkozóan. A DOS operációs rendszer feletti rész egészen a 9FFFFh címig bezáróan (640 Kbyte) gyakorlatilag probléma mentesen használható mint felhasználói programterület. A 0A0000h-0AFFFFh című memória szegmenst video RAM kiterjesztési célokra használják, amennyiben a display vezérlő adapter ezt lehetővé teszi. Az állandó video RAM terület a 0B0000h címnél kezdődik egyszínű monitor vezérlőkártya esetén, és a 0B8000h címnél kezdődik színes grafikus adapter (CGA) kártya esetén. A standard, grafikus lehetőséget nem tartalmazó IBM egyszínű monitor adapter video RAM tartománya a 0B0000h és 0B0FFFh címek közé esik. Ennek az adapter kártyának nagy hátránya a grafikus mód hiánya. Manapság egyre ritkábban használják, mivel megjelentek ezen kártya grafikus lehetőséggel bővített változatai. A video RAM puffer akár két teljes képernyő tartalmat is tárolhat a 0B0000h-0B7FFFh és a 0B8000h-0BFFFFh memória tartományokban.
A standard színes grafikus adapter video puffer tárolója a 0B8000h-0BFFFFh címterületekre esik és egy képernyő tartalom tárolását biztosítja.

A 0C0000h-0EFFFFh memória címeket az I/O csatornára csatlakoztatható adapterek és külső I/O eszközök ROM területeinek biztosítására tartják fenn. Például a Winchester diszk vezérlő kártyán lévő ROM-ok címtartománya a 0C8000h-0CBFFFh címtartományba esik, egy korábbi példát felhozva pedig: az IBM PCjr cartridge ROM-jai a 0D0000h-0EFFFh címterületre kerülnek.

A 0F0000h-0FFFFh szegmenst a rendszer ROM-ok részére biztosítják. Az XT gép ROM-ba égetett BIOS (Basic Input Output System) programja a 0FE000h-0FFFFFh címtartományba esik, a ROM-ba égetett BASIC program pedig a 0F6000h-0FDFFFh memória területre kerülhet.

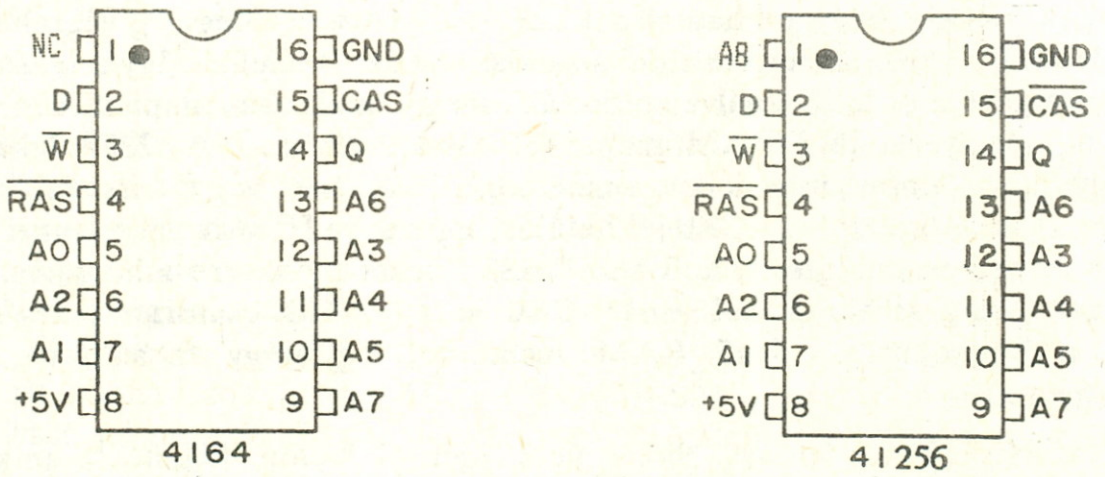
Az 1 Mbyte-ot átfogó címtartományt a leírtak szerint használják valamennyi IBM típusú személyi számítógépben, beleértve az AT gépeket is, figyelembe véve, hogy AT gépeknél az alapkártya címezhető RAM tartománya 512 Kbyte lehet (0-7FFFFh) az első 1 Mbyte-os sávon belül.

Az eredeti IBM XT gépek alaplemezén 2 darab 32Kx8 bit kapacitású ROM memória chip részére van hely, amely a teljes 0F0000h-0FFFFFh rendszer-ROM területet kitölti. A legtöbb gyártó ettől a megoldástól eltért, általában 6-8 28-lábú IC foglalatot helyezett el az alaplemezen és különféle ROM-ok és EPROM-ok választási lehetőségét adja, amivel a 64 Kbyte-os ROM szegmens teljesen, vagy csak a szükségletnek megfelelően lefedhető. Nagyon gyakori eset, hogy a gépben csak 1 db 8Kx8 bites ROM-ot vagy EPROM-ot találunk, amely a BIOS-t tartalmazza. A közölt XT kapcsolási rajzon (2-5.4 ábra) látható, hogy választhatók 2764, 27128 típusú EPROM-ok, 8Kx8 bit kapacitású, vagy 32Kx8 bit kapacitású ROM-ok.

A ROM címdekóder némileg bonyolultabbá vált, de egy 74LS138 3 -- > 8 dekódolón, két kapuáramkörön és egy 4-es DIP mikrokapcsolón kívül más elemet nem igényel. Egy 8-as DIP mikrokapcsolóval beállíthatjuk, hogy a foglalatokba milyen típusú ROM-ot vagy EPROM-ot akarunk tenni. A címdekóder kimenetek a memória chipek -CS (ChipSelect) bemeneteihez kapcsolódnak. Lényeges, hogy a címdekóder a felső címbiteken kivül a -MEMR (Memory Read) jelet is figyelembe veszi a dekódolásnál. Mivel a -MEMR megjelenésekor már valamennyi címvonal stabil állapotában található, a dekódolt kimeneteken nem jelenhetnek meg az egyidejű címvonal átváltásokból adódó zavar-impulzusok. A címdekóder által nem használt alsóbb címvonalak közvetlenül címzik a ROM/EPROM chip-eket a címbemeneteiken keresztül. Az egyes chip-ek -OE (Output Enable) bemeneteire általában a -MEMR jelet kapcsolják.

Az alapkártyán elhelyezett memóriák másik fajtája az írható/olvasható maximum 640 Kbyte terjedelmű DRAM típusú (Dynamic Random Access Memory) memória. Az alkalmazott memória chip-ek 4164, vagy 41256 típusjelűek lehetnek. Az előbbiek 64Kx1 bit, az utóbbiak 256Kx1 bit kapacitásúak. Ahhoz, hogy ezekből az integrált áramkörökből egy byte szervezésű memória tömböt kapjunk, nyolc darabot kell parallel kötni. Így egy ilyen tömb kapacitása 64Kx8 vagy 256Kx8 bit. Ezekből a tömbökből 4 helyezhető el az XT alaplemezen úgy, hogy címtartományaik egymás fölé kerülnek. A valóságban egy tömbben 9 darab memória chip helyezkedik el, mivel minden byte-hoz paritás bitet kell rendelni paritásellenőrzési célokból. Erről később esik szó.

Attól függően, hogy 64 Kbyte-os vagy 256 Kbyte-os tömbökből mennyit helyezünk el az alaplemezen, különféle nagyságú RAM területeket nyerhetünk. Alapkiépítésben a minimális RAM terület 128 Kbyte (0-1FFFFh). Ez két 64 Kbyte-os tömb beépítését igényli a 0-ás és az 1-es sorba. Maximális, 640 Kbyte-os konfiguráció létrehozásához két 256 Kbyte-os és két 64 Kbyte-os tömb szükséges. A 4164 és a 41256 típusú memóriák kivezetés-elrendezését mutatja a 2.8 ábra. Látható, hogy csak az A8 címvonalnál van eltérés.



2-8.ábra.
A 4164 és 41256 típusú memória chip-ek kivezetései.

Az első IBM XT gépeknél csak 64 Kbyte-os memóriákat lehetett használni, és az alaplemezen maximum 256 Kbyte-nyi RAM területet lehetett elhelyezni, az ezen felüli részt az I/O csatornára csatlakozó memória bővítő kártyákra építették. Mivel a 4164 és 41256 típusú memóriákból képzett tömbök mérete eltérő (64 vagy 256 Kbyte), az alkalmazott memóriatípusnak megfelelő 'jumper' átkapcsolásokat kell beállítani az alaplemezen (RAS0-3, CAS0-3 dekóderek).

Mint ismeretes, a dinamikus RAM chip-eknek bitenként csak egy tranzisztorra van szükségük és emiatt kicsi a teljesítmény- és helyigényük. Alapvetően minden egyes bit-cella egy mintavevő-és-tartó áramkör, amelyben a kapacitív elemet vagy feltöltik (logikai 1), vagy kisütik (logikai 0). Az egyetlen probléma ezekkel a memóriákkal, hogy a kisértékű kapacitív elemeken a töltések hamar elszivárognak, és a bit-cellák elfelejtik a bennük tárolt értékeket. Hogy ezt megakadályozzák, a kapacitív elemeket periódikusan, a bennük tárolt értéknek megfelelően kell utántölteni. Ezt a folyamatot "frissítés"-nek nevezik (refresh). A dinamikus RAM chip-ek belső felépítéséről kell egy kicsit többet tudni ahhoz, hogy a frissítés folyamata világosabb legyen. A dinamikus bit cellák egy négyzet alakú, sorokból és oszlopokból álló tömb elemeinek tekinthetők. A memória író/olvasó áramkörök először a sor-címet teszik a chip címbemeneteire, majd egy kicsit később az oszlop-címet. A sor-oszlop címkombináció egyetlen bitet jelöl ki a tömbben. A nyolc párhuzamosan kapcsolt chip esetében ez azt jelenti, hogy a sor-oszlop címkombináció egy teljes byte kijelölését elvégzi. Valahányszor a dinamikus memóriát kiolvassuk, a sor-oszlop címek által meghatározott sor valamennyi bitje felfrissítésre kerül. Ha a bit-cellákat 0,002 másodpercenként frissítjük, a cella az információ tartalmát biztonságosan megtartja.
Vannak olyan alkalmazások, amelyeknél a frissítés automatikusan megtörténik. Ehhez az szükséges, hogy minden sor legalább egy bitje 2 ezred másodpercenként kiolvasásra kerüljön. Ilyen helyzet áll elő például a képernyő frissítő memóriák esetében. Általában azonban nem teljesül a fenti feltétel, ezért a memória frissítést külső eszközökre bízzák. Az IBM XT gépeknél a problémát elegánsan oldották meg: a rendelkezésre álló négy DMA csatorna közül a 0-val jelöltet használják fel frissítési célokra. Az alapkártyán található 8253 programozható számláló/időzítő egyik kimenete 15 µsec-ként DMA ciklust kezdeményez (DRQ0), melynek hatására a 8237A DMA vezérlőegység végrehajt egy-egy memória olvasási ciklust, az egymásutáni sorokhoz tartozó címeken. Így a frissítést a már meglévő áramköröket felhasználva oldották meg, mind az alaplemezen, mind a bővítőkártyán elhelyezkedő DRAM memóriák vonatkozásában. A 8237A DMA vezérlő -DACK0 nyugtázó impulzusai jelzik a memóriák részére, hogy frissítési ciklus van folyamatban. A memóriák a -DACK0 jel hatására egyszerre frissítik valamennyi 64 Kbyte-os vagy 256 Kbyte-os tömbjüket, a RAS0-RAS3 jeleket egyszerre kibocsátva, a CAS0-CAS3 jeleket pedig tiltva. Mivel a -DACK0 jelet az I/O csatorna csatlakozóira is kivezették, a bővítőkártyán levő RAM memóriák ugyanúgy frissíthetők, mint az alaplemezen lévők.

A megoldás kétségkívüli előnye, hogy nem igényel külön frissítő áramköröket a memóriakártyákon, de hátránynak tekinthető, hogy az alkalmazások szempontjából így csak három DMA csatorna jöhet számításba, továbbá hogy a DMA vezérlő minden 15 µsec-os periódusban busz-hozzáférési kérelemmel jelentkezik.

A dinamikus memóriák megbízhatóságát növeli a paritás ellenőrzés. A 64Kbyte-os vagy 256 Kbyte-os tömbökhöz hozzákapcsolnak egy kilencedik memória chip-et, amelyben a byte-onkénti paritásbiteket tárolják. A paritásbit értéke 1, ha a byte-ban található egyesek száma páros, és 0, ha az egyesek száma páratlan. A paritás-képzést és ellenőrzést az e célra konstruált SN74S280 típusú integrált áramkör végzi (lásd az XT kapcsolási rajzát!). Memória írás ciklus alatt ez az áramkör előállítja és a kilencedik memóriába beírja a megfelelő paritásbitet, memória olvasás ciklus alatt pedig újból képezi, majd összehasonlítja a memóriából kiolvasott paritásbit értékkel. Ha a két érték nem egyezik meg, "nem maszkolható megszakítás kérés" jelenik meg (NMI = Non-Maskable Interrupt), amely a gépet azonnal megállítja, és a képernyőn a "Paritás ellenőrzési hiba" (Parity Check Error) üzenet kerül kiírásra. Ebben az esetben ajánlatos a gépet néhány másodpercre kikapcsolni, majd újraindítani. Ha a hiba oka egy meghibásodott memória chip, a bekapcsoláskor induló memória tesztprogram a hibát, és a hiba helyét kimutatja. (A legtöbb ROM BIOS ilyen). Meg kell jegyezni, hogy paritáshibák rendkívül ritkán fordulnak elő. Esetenként előfordulhat, hogy egy rossz program hoz létre ilyen hibaüzenetet, azáltal hogy "teljes összeomlást" idéz elő a DRAM memóriákban. A hibaüzenet megjelenhet, annak ellenére, hogy a memóriák között nincs hibás működésű.
A dinamikus memóriákat vezérlő áramkörök három funkciót valósítanak meg:

a.) címdekódolást végeznek
b.) előállítják a dinamikus memória chip-eket közvetlenül vezérlő jeleket (RAS0-3, CAS0-3, -WE)
c.) a cím-multiplexer megfelelő ütemben a chip-ek címbemeneteire kapcsolja a sor- és oszlopcímeket.

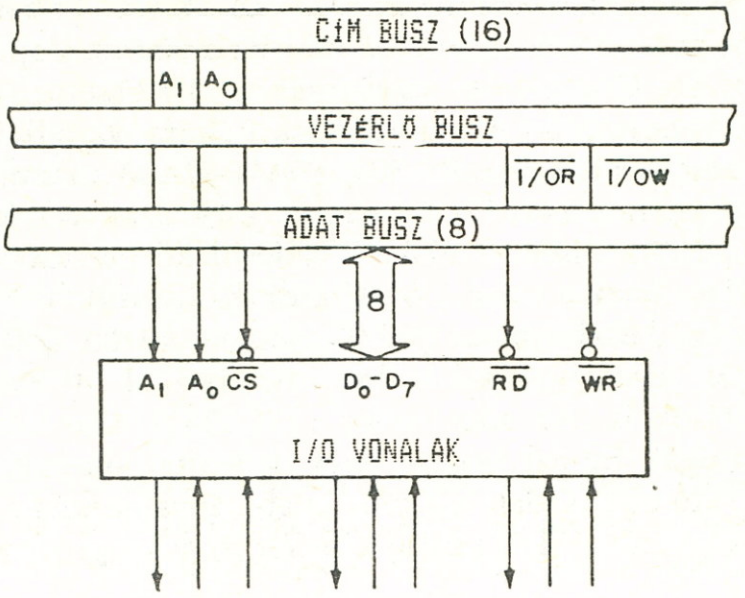
A memória vezérlő áramkörök kapcsolási rajza a 2-5.3 ábrán látható. A memória ciklust vagy egy beíró (-MEMW) vagy egy kiolvasó (-MEMR) jel indíthatja. Ezekből állítják elő a RAS és a késleltetett CAS jeleket, amelyek a vezérlőjelek alapjait képezik. A -MEMW jel ezen kívül leválasztás után közvetlenül is a chip-ek -WE bemeneteire kerül. A címdekóderek feladata, hogy a memória chip-ek összekötött ki-bemeneteihez csatlakozó adat buffer (74LS245) működését engedélyezzék, ha a cím az alaplemez RAM memória-területére esik, és a RAS, CAS jeleket szétosszák a 64 Kbyte-os (256 Kbyte-os) tömbök között a címzésnek megfelelően. A címdekóder néhány kapuból és két 74138 típusú demultiplexerből áll, amelyek a RAS0-RAS3 és CAS0-CAS3 kimeneteket szolgáltatják.
A címmultiplexert két darab 74S157 típusú integrált áramkör képezi. A multiplexelt kimenetek sorosan kapcsolt 33 ohm értékű ellenállások közbeiktatásával jutnak a memória chip-ek címbemeneteire. Ezek a soros ellenállások csökkentik a Shottky TTL kimenetek és a MOS technológiával készült memóriák bemenetei között kialakuló reflexiókat, és a tárolók zaj-védettségét javítják. Ilyen célokat szolgálnak a RAS0-RAS3 vonalakkal sorbakapcsolt ellenállások is. Az adat buffer (74LS245) adat-irányának a kijelölésére a -MEMR olvasó jel szolgál. Ez azt jelenti, hogy nyugalmi állapotban (memória ciklusok között) és a tárolóba való beírás alatt a buffer adat-iránya mindig a memória chip-ek felé mutat, és csak az olvasó jel ideje alatt fordul a buffer iránya a processzor felé.

2.2.4 I/O eszközök csatolása az alapkártyán.

Ebben a fejezetben áttekintésre kerül, hogy hogyan kommunikál a 8088-as processzor az alaplemezen elhelyezett I/O eszközökkel a cím, adat és vezérlő vonalakon keresztül, és közöljük az I/O eszközök címtérképét.

I/O folyamatoknál mindig van két közös mozzanat, amelyek minden esetben megismétlődnek: az adatbusz jeleit regiszterekbe írják és tárolják, és azonosítják a CPU által megcímzett eszközt. A jelek tárolása leginkább OUTPUT műveleteknél szükséges, hogy a buszon gyorsan változó adat és kód információk hosszabb ideig legyenek elérhetők a lassabban működő I/O berendezések részére. INPUT műveleteknél a regiszterek elmaradhatnak, ha kellő hosszú ideig áll a bejövő adat a processzor rendelkezésére. Egy egyszerű példa, a parallel keyboard illesztése valamilyen processzoros rendszerhez. Mivel a leütött billentyűk kódjai viszonylag hosszú ideig megmaradnak, nem szükséges azokat regiszterekbe tárolni, ha a processzor elég gyakran lekérdezi a billentyűzetet. Ilyen esetben elegendő egy egyszerű 3 állapotú (tri-state) adat buffert alkalmazni, mint például a 74LS244-t.

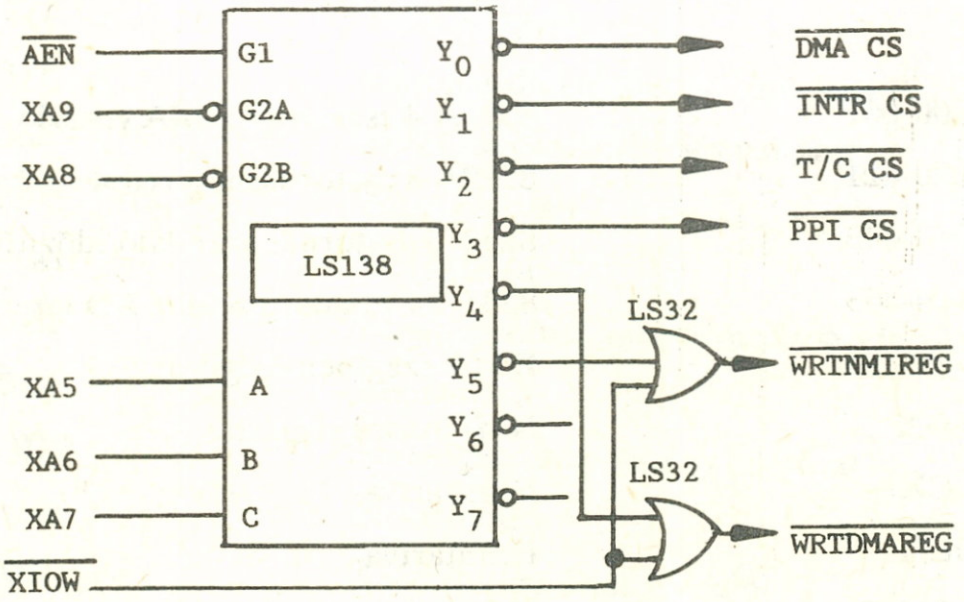
Az I/O folyamatok másik mozzanata, hogy a CPU által megszólított periférikus eszköznek fel kell ismernie, hogy hozzá szóltak. Ezt a mozzanatot nevezzük cím-dekódolásnak, amelynek az a lényege, hogy összehasonlításra kerülnek a címvonalak az I/O eszközön beállított és csak arra jellemző címekkel. Ezenkívül figyelni kell az író vagy olvasó jelek megjelenésére (-IOW, -IOR), amelyek az I/O folyamatot elindítják. Egy 74LS374 regiszterbe például csak a helyes cím és a -IOW vonalon megjelenő pulzus hatására íródik be az adatbusz tartalma. Ugyanúgy az előbbiekben említett keyboard adatpuffer tartalma csak akkor kerül a buszra, ha a rá jellemző cím és a -IOR pulzus megjelenik, egyébként a 74LS244 puffer 'tri-state' állapotban található. Az itt említett regiszterek parallel I/O eszközök, mivel egyszerre mind a nyolc bitjük bemenetként vagy kimenetként szolgálhat. Egy másik csoportba tartoznak a soros eszközök, melyek az adatokat egyetlen vonalon szállítják, egymás utáni bitek formájában.



2-9 ábra.
Intelligens periféria IC interface.

Az I/O eszközök legjelentősebb részét az „intelligens" periféria chip-ek teszik ki. Ezek az integrált áramkörök programozhatók és rendszerint bonyolult műveletek elvégzésére alkalmasak. Az XT alaplemezen a 8237A DMA vezérlő, a 8259A interrupt vezérlő, a 8253 számláló és időzítő és a 8255 parallel interface tartozik ebbe a kategóriába.
Egységesen úgy tekinthetünk egy-egy ilyen intelligens periféria chip-et, mint parallel I/O regiszterek sorozatát, amelyek író és olvasó vezérlőbemenetekkel vannak ellátva. A regiszterek megkülönböztetésére belső és külső logikákat használnak. A 2-9 ábra mutatja egy ilyen chip és az XT belső buszának a kapcsolódását. Bármelyik belső regiszterhez a -CS (Chip Select) és a -RD (Read) vagy -WR (Write) vonalak együttes 0V-ra húzásával férhetünk hozzá. Az aktuális regisztert a periféria címvonalakra adott címekkel választhatjuk ki. Az ábrán az A0 és A1 címvonalak 4 belső regiszter kiválasztását teszik lehetővé. (További címvonalakat bekötve a regiszterek száma növelhető.)

A regiszterek kiválasztása program utján is történhet. Gyakori eset, hogy mindkét megoldást használják egy IC-n belül.
Az I/O címdekóder kimenetét a periféria IC-k CS (Chip Select) bemeneteivel kötik össze. Az XT gépek I/O címdekóderét láthatjuk a 2-10 ábrán. A dekóder lényegében egyetlen 74LS138 típusú integrált áramkör. A bemeneteire az XA5-XA9 címvonalak és az AEN (Address ENable) nevű címbusz engedélyező jel kerülnek. Az XAi címvonalak megegyeznek az Ai vonalakkal, azzal a különbséggel, hogy az Ai-kből kerültek leválasztásra 74LS244 típusú bufferekeren keresztül az alaplemezen történő felhasználás céljából.



2-10 ábra.
I/O címdekóder.

Az IBM PC-k csak az XA0-XA9 címvonalakat használják fel I/O eszközök kiválasztására, az XA10-XA15 vonalakat figyelmen kívül hagyják. Ezért ezeken a felső címvezetékeken megjelenő bármely bitkombináció mellett elérhetők az XA0-XA9 vonalak által megcímzett I/O áramkörök. Ez az úgynevezett redundáns címzési mód egyszerűbb dekódoló logikát biztosít abban az esetben, ha nagyszámú I/O címre van lehetőség.

(A 8088 processzor 65536 bemeneti és ugyanennyi kimeneti I/O címet tud képezni. I/O utasításoknál a CPU az A16-A19 címvonalakat logikai 0 értéken tartja.)
Az XA0-XA4 címvezetékeket a 2-10 ábra szerinti címdekóder szintén figyelmen kívül hagyja, így a dekóder kimenetekkel összekötött IC-k további 32 alcímmel rendelkezhetnek. Például, ha az XA5-XA9 vezetékek logikai 0 értékűek, és a címbusz engedélyezett (AEN = 1), a 74LS138 áramkör Y0 kimenete kiválasztja a 8237 DMA vezérlő chip-et. Bármelyik 00000 és 11111 = 31 közötti I/O cím szintén a DMA vezérlőhöz tartozik. A 8237A azonban nem használja az XA4 címvonalat, mivel csak 16 I/O 'port'-ja van, ezért a 0-15 és 16-31 címek azonosak.
A 74LS138 dekóder további kimenetei a 8259A megszakítás-kérés vezérlő, a 8253 számláló/időzítő és a 8255 parallel interface áramkörök kiválasztását végzik. Az Y4 kimenet a DMA szegmens regiszter feltöltését, az Y5 kimenet pedig az NMI regiszterbe való beírást engedélyezheti.

A 2-4 táblázat az XT gépekkel kapcsolatos I/O címeket összegzi. A 0-FFh címtartományba eső periférikus eszközök az alaplemezen találhatók, a 200-3FFh tartományba esők pedig az I/O csatornára kapcsolódnak.

Hexadecimális címtartomány	**I/O eszköz**
	Alaplemez
000-00F	8237A 4 csatornás DMA vezérlő
020-021	8259A 8 csatornás interrupt vezérlő
040-043	8253 3 csatornás számláló/időzítő
060-063	8255 24 vonalas parallel I/O interface
080-083	DMA szegmens regiszter
0AX *	NMI maszk regiszter
0CX	Fenntartva
0EX	Fenntartva
100-1FF	Nem használható

* Bekapcsoláskor a 8088 CPU NMI bemenete maszkolatlan.
A maszk bit beállítható és törölhető software úton a következőképpen:
Maszk beállítás: 80h kiadása a 0A0h I/O címre
Maszk törlés: 00h kiadása a 0A0h I/O címre.

2-4 táblázat. (I.)
Az XT I/O címtérképe.

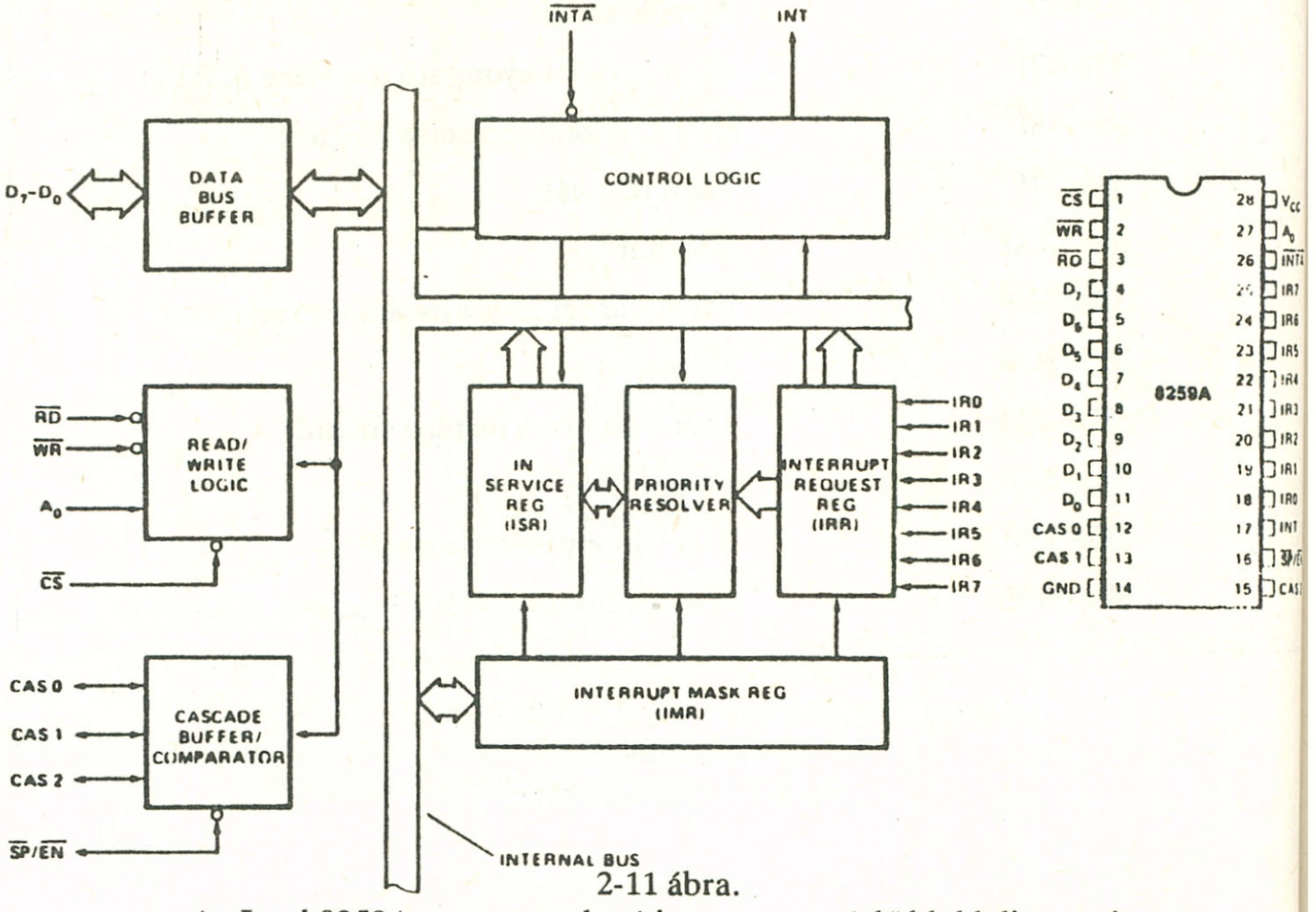
Hexadecimális címtartomány	I/O eszköz
	I/O csatorna
200-20F	Joystick adapter
210-217	I/O buszkiterjesztő egység
220-24F	Fenntartva
250-277	Nem használt
278-27F	Második parallel nyomtató interface (LPT2)
280-2EF	Nem használt
2F0-2F7	Fenntartva
2F8-2FF	Második aszinkron UART interface (COM2)
300-31F	Prototípus kártya
320-32F	Merev lemezes diszk vezérlő
330-377	Nem használt
378-37F	Első parallel nyomtató interface (LPT1)
380-38F	SDLC kommunikációs adapter
390-39F	Nem használt
3A0-3AF	Fenntartva
3B0-3BF	Monochrom display/nyomtató adapter
3C0-3CF	Fenntartva
3D0-3DF	Színes grafikus display adapter
3E0-3EF	Fenntartva
3F0-3F7	5 l/4" floppy diszk vezérlő
3F8-3FF	Első aszinkron UART interface (COM1)

2-4 táblázat. (II.)
Az XT I/O címtérképe.

2.2.5 A 8259A interrupt vezérlő, az XT interrupt rendszere, hardware interrupt lista, software interruptok és az interrupt vektorok elhelyezése.

A számítógép alapvető feladata, hogy adatokat fogadjon el és adjon ki magából és képes legyen az adatok feldolgozására. Miközben a gép egy feldolgozási folyamatban vesz részt vagy arra vár, hogy lassú perifériákkal adatcseréket végezzen, más perifériák adatokat szolgáltathatnak, melyek elvesznek, ha nem fogadják őket. Az egyik lehetséges hardware/software eljárás a probléma megoldására az interrupt rendszer kialakítása. Ha az adat elvihető, a perifériális berendezés megszakítás kéréssel jelentkezik a CPU felé, azt kérve, hogy a processzor hagyja abba addigi tevékenységét, fogadja az adatot, majd folytassa a munkáját ott ahol abbahagyta. Ezeket a megszakításkéréseket úgy tekinthetjük, mint aszinkron CALL utasításokat. Aszinkron azért, mert bármelyik időpillanatban létrejöhet, és CALL utasítás azért, mert a visszatérési címet és a flag-eket a verem regiszterbe (stack) írjuk, majd a program elágazik egy rutin címére. Az interrupt kezelő rutin, amelyet a periféria meghív, IRET utasítással fejeződik be, miután feldolgozta a bejövő adatokat. Ekkor a programszámláló (IP) és a flag regiszter visszaveszi a stack-ben tárolt eredeti értékét, és így ugyanaz a gépi állapot áll elő, mint a megszakítás előtt. (Követelmény, hogy az interrupt kezelő rutin hagyja változatlanul minden regiszter tartalmát.)

A 8259A típusú interrupt vezérlő nyolc független csatornán érkező megszakítás kérést tud koordinálni. A kéréseket sorrendbe rakja, meghatározza melyik eszköznek van a legnagyobb prioritása, majd megszakítás kérelemmel jelentkezik a processzor felé. Az interrupt vezérlő blokk diagramja a 2.11 ábrán látható.



2-11 ábra.

Az Intel 8259A programozható interrupt vezérlő blokkdiagramja.

Az IR0-IR7 bemeneteken TTL logikai 1 szintek jelentik a megszakítás kérési állapotot. A legmagasabb prioritással az IR0 vonal, a legalacsonyabbal az IR7 rendelkezik. Az interrupt maszk regiszter (IMR) egy nyolcbites regiszter, amelynek az n. bitje letilthatja az IRn vonalon megjelenő kérést. Ha az n. bit tartalma 0, az IRn vonal engedélyezett, egyébként tiltott. Ha a 8259A vezérlő azonosított egy bejövő kérelmet, a CPU INT vonalát logikai 1 értékre állítja. Amennyiben a 8088 interrupt flag 1 állapotú (egy előzőleg végrehajtott STI utasítás eredményeként), a CPU nyugtázza az interruptot a -S0, -S1, -S2 vonalakon keresztül, melynek következtében a 8288 busz vezérlő INTA kimenetén logikai 0 jelenik meg. Az interrupt vezérlő ezt észlelve egy 1 byte-os interrupt vektor számot helyez az adatbuszra. Ez a szám azonosítja a RAM memória első 1Kbyte területén elhelyezkedő 4 byte-os interrupt vektort. Pontosabban, ha ez a szám KK, akkor az interrupt vektor a 4*KK címen található.

A CPU ezután végrehajt egy távoli hívást (FAR CALL) az interrupt vektor által mutatott címre (2 byte szegmenscím 2 byte mutató). Ezen a címen az interrupt kezelő rutinnak feltétlenül jelen kell lennie, különben programhiba keletkezik.

A 8259A interrupt vezérlő „intelligens" I/O eszköz, és megfelel a 2-9 ábrán közö általános formának. Tehát vannak író, olvasó, kiválasztó, adat és címvonalai. Ezenfe ul rendelkezik a már ismertetett IR0-IR7 bemenetekkel, és a CPU-val kommunikáló INT, -INTA vonalakkal. A CAS0-CAS2 jelek csak kaszkádba kötött interrupt vezérlőknél használatosak, az XT gépben ezek a vonalak nincsenek bekötve, mivel csak egy darab ilyen vezérlő van az alaplemezen. Az utolsó vonal az SP/EN, mely logikai 0 értéket vesz fel az interrupt vektor számnak az adatbuszra helyezésekor, szintén nincs kihasználva.

A 8259A vezérlőnek számtalan üzemmódja van. Ezeket a katalógusok részletesen leírják. Most azt a módot ismertetjük, amelyet az XT gépekben használnak. Az üzemmód beállításhoz három inicializáló utasítás szót kell az interrupt vezérlőhöz küldeni. A gép bekapcsolásakor a BIOS idevonatkozó inicializáló programja végzi el ezt a műveletet:

```
mov al,13h      ; éltriggerelt int. bemenetek, 1 db 8259A, ICW4 szükséges
out inta00,al   ; az első utasítás byte kiküldése az első vezérlő I/O címre (20h)
mov al,8        ; a 8-0Fh interrupt vektorokat rendeli az IR0-IR7 bemenetekhez
out inta01,al   ; a második byte kiküldése a második vezérlő I/O címre (21h)
mov al,9        ; ICW4: puffer mód, normál EOI, 8088-os mód
out inta01,al   ; a harmadik byte kiküldése a második vezérlő I/O címre (21h)
```

Az inicializálás után a következő állapotok vannak érvényben:

a.) az IR0-IR7 vonalakon a 0 -->1 jelváltások okozhatnak megszakítás kéréseket (éltriggerelt mód)
b.) az IR0-IR7 vonalakhoz sorrendben a 8-0FH interrupt vektorok tartoznak
c.) az 8259A interrupt vezérlő a 8088-as CPU-val tud együttműködni (a másik lehetőség a 8080/85 processzor)
d.) normál EOI (end of interrupt) mód.

Az interrupt rendszer azonnal müködésbe lép, ha a CPU végrehajt egy STI (set interrupt flag) utasítást. Előtte azonban a BIOS feltölti az interrupt maszk regisztert (IMR) és csak az induláshoz szükséges megszakítás kéréseket engedi továbbhaladni. Például:

```
mov al,0BCh     ; engedélyezett interrupt kérések: diszk(6. bit),
                ; billentyűzet (1. bit) és időmérés (0. bit)
out inta01,al   ; a maszk regiszter feltöltése
sti             ; az interruptok engedélyezése
```

Miután egy interrupt előfordult, további interruptok nem kerülnek a CPU-hoz. A 8259A azonban feljegyzi a beérkező kéréseket, de mindaddig nem küldi tovább a processzorhoz, míg egy EOI kódot nem kap. Az interrupt kezelő rutin engedélyezheti a további interruptokat a következő néhány utasítással:

```
sti
mov al,20h      ; EOI kód küldése
out inta00,al   ; a 8259A vezérlőhöz
```

Az IR0-IR7 megszakítás kérő vonalakat az I/O csatorna csatlakozóira is kivezették, így az adapter kártyák tervezői is számolhatnak az interruptos üzemmel. Meg kell azonban jegyezni, hogy az IBM valamennyi IR bemenethez hozzárendelt egy konkrét I/O eszközt (még ha az nincs is beépítve), kivéve az IR2 bemenetet, ezt viszont fenntartja későbbi felhasználás céljára. Ezért nehéz helyzetbe kerülhet az a tervező, aki az XT gépen olyan feladatot kíván megoldani, amely sok interrupt vonal kezelését igényli. Egy lehetséges megoldás a rendszer által éppen nem használt interrupt vektorok (pl. az egyik soros kommunikációs egység, nyomtató) átirányítása a felhasználó által definiált célokra.
Egy másik eljárás szerint több I/O eszközt kapcsolnak egyetlen IR vonalra. Ebben az esetben viszont a software csak egy státusz byte lekérdezése után tudja eldönteni, hogy melyik perifériáról érkezett megszakítás kérés. Az egy IR vonalra kapcsolt I/O perifériákat pedig némi plusz áramkörökkel kell kiegészíteni a helyes működés érdekében.

Vektorszám	Szám	Felhasználás
2h	NMI	Alaplemez RAM paritásfigyelés, I/O csatorna paritásfigyelés, 8087 aritmetikai processzor
8h	0	Rendszer óra
9h	1	Billentyűzet
Ah	2	Fenntartva
Bh	3	Második aszinkron kommunikációs interface (COM2)
Ch	4	Első aszinkron kommunikációs interface (COM1)
Dh	5	Merev lemezes diszk vezérlő
Eh	6	Floppy diszk vezérlő
Fh	7	Parallel nyomtató

2-5 táblázat.
Hardware interrupt lista.

A hardware interruptok egy különleges fajtája a 8088 processzor nem maszkolható interruptja (NMI). Szerepére már a CPU ismertetésénél kitértünk. Az XT gépek érdekessége, hogy az NMI bemenetet egy kiegészítő áramkörrel maszkolhatóvá alakították (2-5.2 ábra). Három I/O eszköz kérhet interruptot ezen a vonalon: a RAM memória paritás ellenőrző áramköre, az I/O csatorna I/O CH CK vonalán keresztül paritáshibát jelző külső eszköz, és az aritmetikai processzor interrupt kimenete. A megszakításkérő azonosítását a software státusz byte beolvasásával végzi el.

A hardware interrupt listát a 2-5 táblázat tartalmazza. A számok az interrupt vektorokat és a 8259A IRn bemeneteit jelölik, a felhasználás oszlop pedig a bemenetekhez rendelt I/O eszközöket sorolja fel.

Az interruptoknak egy nagyon fontos csoportját az úgynevezett software interruptok teszik ki. Az elgondolás lényege, hogy a programoknak lehetnek olyan részei, amelyek más programrészek szolgáltatásait igénylik, és ezeket a szolgáltatásokat a hardware interruptokhoz hasonló módon igényelheti a software az INT n 8088 assembler utasításon keresztül. Tehát ezeket a programrészeket egy-egy interrupt-vektor-számhoz rendelik hozzá, a vektor a programrész kezdőcímére mutat. A software interruptok ezenkívül egyszerű lehetőséget adnak arra, hogy programból is hívhatók legyenek a hardware interrupt kezelő rutinok, ha erre szükség van (pl. hibakeresésnél). A működési mechanizmus teljesen azonos a hardware interruptoknál megismertekkel.
A software interruptok három fő csoportját különböztethetjük meg:

1) ROM-BIOS software interruptok. Ezeket az IBM definiálta és univerzálisnak tekinthetők az egész PC családban. Segítségükkel a ROM-BIOS program egyes részei aktivizálhatók, és lehetővé válik, hogy ezeket a jól használható BIOS rutinokat felhasználói programokból is hívhatjuk. Az 5h és 10h-1Ah számú megszakítások tartoznak ide.
2) DOS software interruptok. Az eddig említett interruptoktól eltérően, ezek az interruptok nincsenek a számítógépbe építve: a software az ami létrehozza őket, jelen esetben a DOS operációs rendszer. Mivel a DOS operációs rendszer szinte mindig jelen van, úgy tűnik, mintha ezek az interruptok is a gépbe építettek lennének. A DOS belső rutinjai, és az alkalmazások is használják ezeket a hívásokat. A 20h-3Fh című interruptok tartoznak ebbe a csoportba.
3) Felhasználói software interruptok. Az éppen futó program definiálja és hozza létre őket (ide sorolhatjuk a BASIC-et is, amely csak néhány speciális saját interrupttal rendelkezik). A felhasználói szoftver interruptok címtartománya: 60h-67h és 80h-F0h (BASIC).

A software interruptokkal kapcsolatban felmerülhet a kérdés, hogy mennyiben nyújtanak többet, mint a szokásos CALL hívások. A válasz erre az, hogy sokkal flexibilisebbek, mint a CALL utasítások. Tételezzük fel, hogy felhasználói programunkba közvetlen hívásokat építettünk be az operációs rendszer számunkra szükséges rutinjainak elérésére. Ez megkívánja a rutinok pontos címeinek az ismeretét. Az operációs rendszerek azonban folyton bővülnek, új verziók jelennek meg, melyekben a rutinok új címekre kerülnek. Tehát egy megváltozott operációs rendszer mellett programunk használhatatlan lesz, hacsak át nem írjuk az összes általunk használt operációs rendszer rutinok címdefinícióit.

Software interruptokat használva közvetlen hívások helyett a rutinok elérésére, ez a probléma nem áll fenn. A felhasználói program a cím ismerete nélkül tudja hívni a kívánt rutint az INT n utasításon keresztül, felhasználva azt a tényt, hogy az operációs rendszer a gép bekapcsolásakor automatikusan feltölti a vektor táblát a megfelelő címekkel.
Egy másik előny abból származik, hogy az interrupt vektor tartalmát könnyen módosíthatjuk a táblában, és így a számunkra nem megfelelő rutinokat helyettesíthetjük az általunk megváltoztatottal, vagy egy rutin helyére például egy egészen más jellegű I/O kezelő programot helyezhetünk.

A 2-6 táblázat az IBM PC-kben használatos interrupt vektorokról tájékoztat. Az első oszlopban az interrupt vektor memória címe, a másodikban a vektor száma, a harmadikban pedig a hozzárendelt funkció található. A típus nevű utolsó oszlop az interrupt jellegéről tájékoztat. A HARDW a hardware interruptot, a BIOS és a DOS a rendszer hívásokat, a TRAP a 8088 CPU által definiált belső interruptokat, a SPEC speciális hívásokat, a USER pedig a felhasználói software interruptokat jelöli.

Memória cím (hex)	Vektor száma	Funkció	Típus
0-3	0	Nullával való osztás	TRAP
4-7	1	Egyes programlépés	TRAP
8-B	2	Nem maszkolható interrupt (NMI)	HARDW
C-F	3	'Break point' utasítás	TRAP
10-13	4	Túlcsordulás	TRAP
14-17	5	A képernyő tartalmának kinyomtatása	BIOS
18-1B	6	Fenntartva	
1C-1F	7	Fenntartva	
20-23	8	'Real time' interrupt (18,2/sec)	HARDW
24-27	9	Billentyűzet	HARDW
28-2B	A	Fenntartva	HARDW
2C-2F	B	Második aszinkron komm. interface	HARDW
30-33	C	Első aszinkron komm. interface	HARDW
34-37	D	Merev lemezes diszk vezérlő	HARDW
38-3B	E	Floppy diszk vezérlő	HARDW
3C-3F	F	Parallel nyomtató	HARDW
40-43	10	Video I/O hívás	BIOS
44-47	11	A gép kiépítettségét ellenőrző hívás	BIOS
48-4B	12	Memória méretét meghatározó hívás	BIOS
4C-4F	13	5 1/4"-os diszk vezérlő I/O hívás	BIOS
50-53	14	RS232 kommunikációs interface I/O hívás	BIOS
54-57	15	Kazetta vezérlő I/O hívás	BIOS
58-5B	16	Billentyűzet I/O hívás	BIOS
5C-5F	17	Nyomtató I/O hívás	BIOS
60-63	18	Rezidens ROM BASIC belépési pont	BIOS
64-67	19	Az operációs rendszer betöltése (boot strap)	BIOS

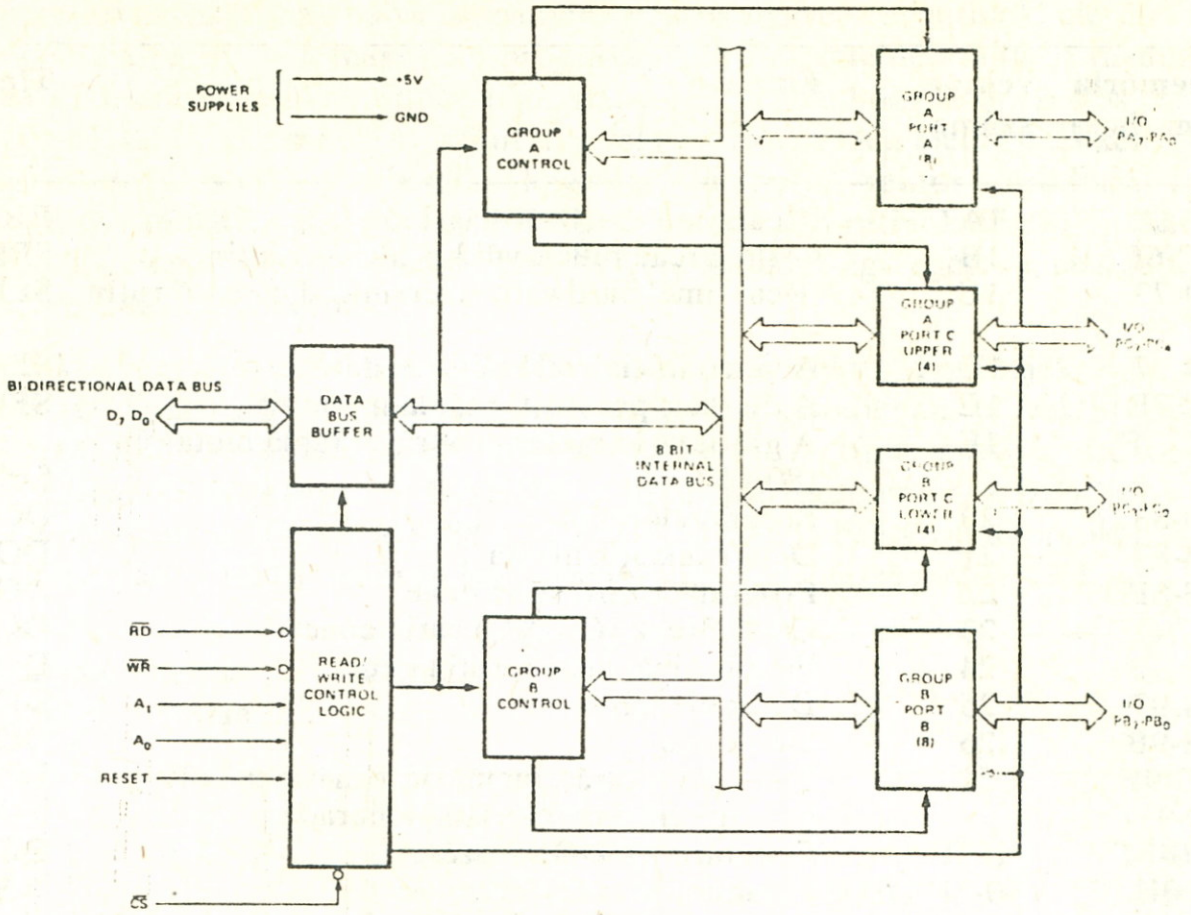
Memória cím (hex)	Vektor száma	Funkció	Típus
68-6B	1A	'Real time' óra beállítás, kiolvasás	BIOS
6C-6F	1B	Ctrl&Break billentyűkkel hívható rutin	SPEC
70-73	1C	'Real time' hardware interrupttal hívható rutin	SPEC
74-77	1D	A video inicializáló tábla mutatója	SPEC
78-7B	1E	A diszkett paraméter tábla mutatója	SPEC
7C-7F	1F	A grafikus karakter generátor tábla mutatója (80-FF)	SPEC
80-83	20	Program lezárás	DOS
84-87	21	DOS funkciók hívása	DOS
88-8B	22	Program lezáró rutin címe	DOS
8C-8F	23	Ctrl&Break (Ctrl&C) rutin címe	DOS
90-93	24	Fatális hibát kezelő rutin címe	DOS
94-97	25	Diszk olvasás	DOS
98-9B	26	Diszk írás	DOS
9C-9F	27	Program lezárás, memória kiürítés nélkül (a program a memóriában marad)	DOS
A0-FF	28-3F	Fenntartva a DOS részére	DOS
100-17F	40-5F	Fenntartva	
180-19F	60-67	Fenntartva a felhasználói software interruptok részére	USER
200-217	80-85	Fenntartva a BASIC részére	USER
218-3C3	86-F0	Fenntartva a BASIC interpreter részére	USER
3C4-3FF	F1-FF	Nem használt	

2-6 táblázat.
Az IBM XT interrupt-vektor táblája.

2.2.6 A 8255 programozható periféria interface és a rendszer konfigurációt beállító DIP kapcsolók.

A 8255 PPI (programmable peripheral interface) vagy másnéven PIO (parallel I/O) a legrégibb intelligens I/O eszköz az Intel mikroprocesszor családokban. Az XT gépekben egy ilyen 40 lábú chip-et találunk, amelynek 16 vonalát bemenetként, 8 vonalát pedig kimenetként használják.

A 8255 valószínűleg a legolcsóbb I/O áramkör ha a költség/bit mutatót vesszük figyelembe. Népszerűségét sokféle üzemmódjának, és a 24 I/O kivezetésének köszönhette. A kivezetések többféleképpen elrendezhetők: bemenetként és kimenetként, vezérlő jelek képezhetők 'handshake' és interrupt-os üzemmódokra. A 2-12. ábra a 8255 PPI blokkvázlatát mutatja. Megtalálhatók mindazok a mikroprocesszor busz interface jelek, melyek egy intelligens periféria eszközre jellemzőek: -RD, -WR, -CS, 8 kétirányú adatvonal és belső regisztereket címző periféria címvonalak (A0 és A1).



2-12 ábra.
A 8255 PPI blokkdiagramja.

Az I/O kivezetéseket három 8-bites csoportba sorolták (A, B, C) és mindegyik csoport külön címmel rendelkezik. Mindegyiket használhatjuk kimenetnek vagy bemenetnek. Az A csoport úgy is kialakítható, mint egy kétirányú adatbusz, a vezérlőjeleket ebben az esetben a C csoport 5 kivezetése szolgáltatja. A chip RESET bemenetére az általános nullázó jelet kötik. Ennek hatására mind a 24 I/O kivezetés INPUT állapotba kerül, azaz bejövő adatok fogadására készül fel. Ez a megoldás megakadályozza, hogy a tápegység bekapcsolásakor helytelen működés, vagy meghibásodás jöjjön létre azért, mert a 8255 valamely kivezetése nem olyan állapotban van, mint amit a hozzákapcsolt külső hardware megkívánna.

A 8255-ben négy belső regiszter található, melyek közül három beírható és kiolvasható, egy pedig csak beírható. A regisztereket az A1 és A0 címvonalak választják ki. Az A1A0 = 00 az A, az A1A0 = 01 a B, az A1A0 = 10 a C regisztert, végül pedig az A1A0 = 11 a csak írható vezérlőregisztert jelöli ki. Az A regiszter kiolvasásához például az szükséges, hogy az A1,A0, -RD és -CS jelek egyidőben logikai 0 állapotot vegyenek fel, ekkor a 8255 PPI az adatbuszra helyezi az A regiszter tartalmát.

Mint ahogy a 2-4 táblázatban közölt címtérképből is kiderül az IBM XT az A, B, C és vezérlő regiszter I/O címeket a 60H-63H tartományban helyezi el egymás után. Ha mind a 24 vonalat bemenetnek használnák, a 8255 nem kívánná meg hogy programozzák: a bekapcsolási nullázójel hatására ebbe az állapotba kerülne.

Ettől eltérő esetekben, a vezérlőregiszterbe töltött utasításszóval kell az üzemmódot kiválasztani. A vezérlőregiszter négy célra használható:

1) Az adatáramlás irányának a kijelölésére az A, B és C 'port'-okon.
2) Az A 'port' három lehetséges üzemmódja közül egynek a kiválasztására.
3) A B 'port' két lehetséges üzemmódja közül egynek a kiválasztására.
4) A C 'port' bitjeinek bitenkénti beállítására vagy törlésére.

Az IBM XT a 0-val jelölt módot valósítja meg. Ebben az üzemmódban az A és B csoportok kimenetként vagy bemenetként konfigurálhatók, a C csoport pedig két négy bites részre bontható, amelyek szintén kimenetek vagy bemenetek lehetnek. A 0 mód a 24 kivezetést I/O vonalként használja. Az 1 és 2 jelű módban a C port bizonyos kivezetései interrupt és 'handshake' vezérlőjelekként működnek.

A 8255 vezérlőregiszterébe a ROM BIOS inicializáláskor a 99h értéket küldi. Az egyes bitek értelmezése a következő:

	0. bit:	a C port C0-C3 vonalainak irányát határozza meg
	1. bit:	a B port B0-B7 vonalainak irányát határozza meg
	2. bit:	a B port üzemmódját adja meg (0 vagy 1)
	3. bit:	a C port C4-C7 vonalainak irányát határozza meg
	4. bit:	az A port A0-A7 vonalainak irányát határozza meg
6. és	5. bit:	az A port üzemmódját adják meg (00 = 0 mód, 01 = 1 mód, 10 = 2 mód)
	7. bit:	1

A vonalak irányát meghatározó vezérlő-bitek 1 értékűek INPUT és 0 értékűek OUTPUT esetén. Ha a 7. bit logikai 0, a 8255 a részére küldött byte-ot nem vezérlő utasításként, hanem a C port egyes bitjeit beállító vagy törlő utasításként értelmezi.
Az 10011001 (99h) utasítást elemezve belátható, hogy a ROM BIOS a 8255 PPI-t inicializáláskor 0 módba állítja, a B0-B7 vonalakat kimeneteknek, az A0-A7 és a C0-C7 vonalakat pedig bemeneteknek definiálja. Az A0-A7 vonalakon a billentyűzetről érkező közvetlen kódokat lehet beolvasni soros parallel átalakítás után (u.n. 'scan' kódok, amelyek a billentyűk elhelyezkedésére utalnak és nem ASCII kódok). Egyes gépeknél diagnosztikai célokra is használják az A0-A7 bemeneteket. Ilyen például az eredeti IBM XT. A billentyűzet működése, az idetartozó interface ismertetése a könyv egy későbbi fejezetében található. Itt csak annyit jegyzünk meg, hogy a billentyű-kódok soros-parallel átalakítását egy 74LS322 típusú IC és a hozzá kapcsolódó áramkörök biztosítják (2-5.6. ábra U53 pozíció). Diagnosztizálásnál a soros-parallel átalakító bemenetére átmenetileg a 8255 B2 tartalék kimenete kerülhet. Így mód nyílik arra, hogy a keyboard-interface-t tesztprogrammal ellenőrizzék. (A közölt XT kapcsolási rajz a diagnosztikai lehetőséget nem tartalmazza.) A billentyűkódok kiadását az A0-A7 vonalakra a B7 kimenet logikai 0 állapota engedélyezi, logikai 1 állapota pedig tiltja. A logikai 1 állapot ezenkívül nullázza a 74LS322 regisztert, így a benne tárolt billentyűkódot törli.

A 8255 B6 kimenete a keyboard I8048 típusú mikroprocesszora által generált órajeleket (KBD CLK) tudja letiltani. A billentyűkódokat ezek az órajelek léptetik be a 74LS322 regiszterbe. (Pontosabban: ezen órajelek 2 rendszer-óra periódussal késleltetett megfelelői.)

A B5 és B4 jelű kimenetek az I/O csatorna és az alaplemezen elhelyezett RAM paritás-ellenőrző áramkörök működését engedélyezik logikai 0 állapottal. A paritás-ellenőrző áramkörök kimeneteit a C7 és C6 bemenő vonalakra kötötték (C7 = alaplemez RAM paritás, C6 = I/O csatorna paritás). A software ezt a két vonalat használja NMI státusz információ megszerzésére. (Ha mindkét vonal logikai 0 állapotú, akkor az NMI megszakítást a 8087 aritmetikai processzor okozhatta.) A 8255 B3 kimenete a SW1 DIP kapcsolón beállított értékeket multiplexeli a C0-C3 bemenetekre. Ha a B3 logikai 1 állapotú, akkor a kapcsoló felső 4 bitje, ha 0 állapotú, akkor a kapcsoló alsó 4 bitje kerül a C0-C3 vonalakra.
A B2 kimenet tartalék célokra szolgál.
A B1 és B0 kimenetek a hangszóró vezérlő áramkörökhöz kapcsolódnak. A B1 az adatjeleket szolgáltatja, a B0 pedig logikai 1 állapotban a 8253 számláló/időzítő 2. csatornájának négyszögjelét engedélyezi a hangszóró számára.
A 24 I/O vonalból már csak a C4-C5 jelűek nem kerültek ismertetésre. A C4 tartalék célokra fenntartott, a C5 pedig a 8253 2. csatornájának kimenetére csatlakozik (ugyanoda, ahová a hanszóró vezérlő egyik bemenete).
A 8255 periféria-interface I/O bit-funkcióit összesíti az XT Technical Reference leírásból átvett 2-7 táblázat. Az alaplemezen található SW1 DIP kapcsolósor beállítási lehetőségeit vizuálisabban mutatja a 2-13 ábra.

A SW1 kapcsolósor 8 db. mikrokapcsolót tartalmaz, melyek az alaplemez RAM területének, a monitor adapter-kártya típusának, a beépített floppy diszkek számának, az aritmetikai processzor interrupt engedélyezésének jelzésére szolgálnak.

A kapcsoló-funkciók ismertetése:

1. számú kapcsoló: normál működésnél OFF állapotban található. ON állapotban bekapcsolási teszt-hurkot hoz létre. (loop POST)
2. számú kapcsoló: a 8087 aritmetikai processzor interrupt kérelmét tiltja ON állapotban. Ez a normál beállítási mód, akár van a gépben 8087 processzor, akár nincs. A 8087-et ugyanis nem használják interrupt-os üzemben. Ha valaki mégis ezt a módot választaná, akkor meg kell írja az ide vonatkozó interrupt-kezelő rutinokat, és a kapcsolót OFF állásba kell helyeznie.
3.-4. számú kapcsolók: az alaplemezen kiépített RAM memóriaterület beállítására szolgálnak.
5.-6. számú kapcsolók: a rendszerbe épített monitor-vezérlőkártya típusa és működési módja adható meg segítségükkel.
7.-8. számú kapcsolók: a rendszerbe épített 5-1/4"-os floppy diszkek számáról adnak felvilágosítást.

A SW1 kapcsolók által nyújtott információk program útján is meghatározhatók (ROM BIOS rutinok), és az így nyert adatokat a BIOS 400h-47Fh RAM területén tárolják. A kapcsolók állásait csak az inicializáló rutinok veszik figyelembe. Mivel működés közben a konfiguráció erősen megváltozhat (pl. RAM diszk installálása), célszerűbb ha mindig a rendszert ellenőrző program által szolgáltatott adatokra támaszkodunk, és soha nem olvassuk ki a kapcsolósort közvetlenül.

Hex Port Number							
0060	INPUT	PA0	+ Keyboard Scan Code	0	vagy	Diagnostic Outputs	0
		1		1			1
		2		2			2
		3		3			3
		4		4			4
		5		5			5
		6		6			6
		7		7			7
0061	OUTPUT	PB0	+ Timer 2 Gate Speaker				
		1	+ Speaker Data				
		2	Nem használt				
		3	High vagy Low kapcsolók olvasása				
		4	-RAM Parity Check engedélyezése				
		5	-I/O Channel Check engedélyezése				
		6	-Keyboard Clock Low szinten tartása				
		7	- (Keyboard engedélyezése) vagy + (Keyboard törlése)				
0062	INPUT	PC0	Loop on POST	Sw-1	vagy	Display 0	**Sw-5
		1	+ Co-Processor Install	Sw-2		Display 1	**Sw-6
		2	+ Planar RAM Size 0	*Sw-3		#5-1/4" Drives 0	***Sw-7
		3	+ Planar RAM Size 1	*Sw-4		#5-1/4" Drives 1	***Sw-8
		4	Nem használt				
		5	+ Timer Channel 2 Out				
		6	+ I/O Channel Check				
		7	+ RAM Parity Check				

0063 Parancs/Mode regiszter

Mode regiszter értékei:

7	6	5	4	3	2	1	0
1	0	0	1	1	0	0	1

Hex 99

*

Sw-4	Sw-3	A memória mérete az alapkártyán
0	0	64K
0	1	128K
1	0	192K
1	1	256K

**

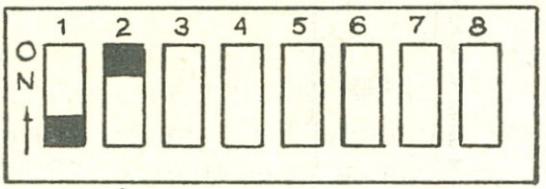
Sw-6	Sw-5	Display módok bekapcsoláskor
0	0	Foglalt
0	1	Colour 40x25 (BW mód)
1	0	Colour 80x25 (BW mód)
1	1	IBM Monochrome 80x25 mód

***

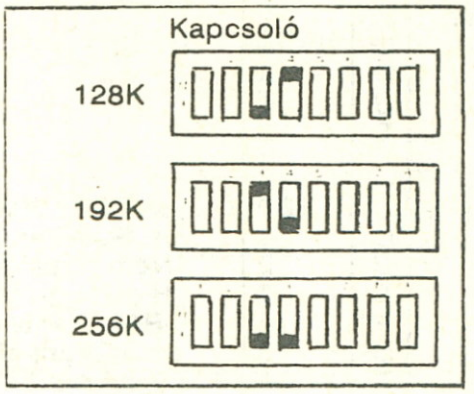
Sw-8	Sw-7	Az 5-1/4" drive-ok száma a rendszerben
0	0	1
0	1	2
1	0	3
1	1	4

Megjegyzés: Plusz (+) jelzi, hogy az 1 értékű bit hajtja végre a megjelölt funkciót
Minusz (-) jelzi, hogy a 0 értékű bit hajtja végre a megjelölt funkciót
PA bit = 0 : kapcsoló "ON", PA bit = 1 : kapcsoló "OFF" állapot

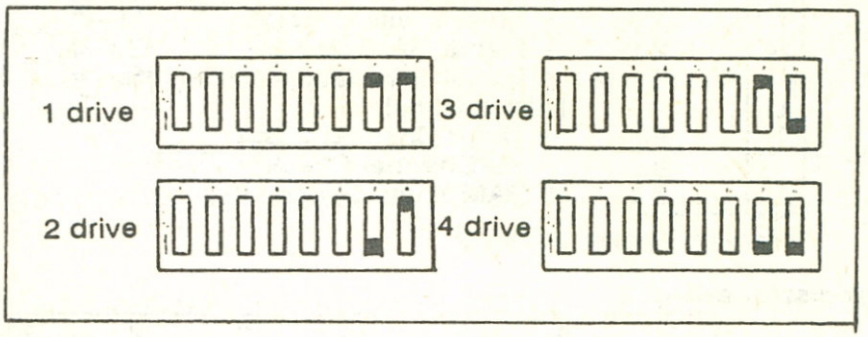
2-7 táblázat.
A 8255A PPI bit-térképe.



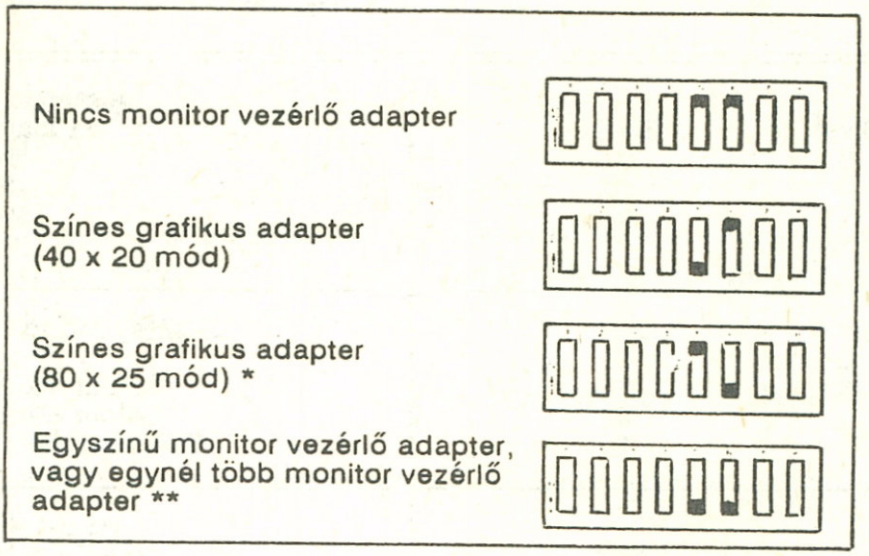
Pozíció	Funkció
1	Normál működés OFF (ON állapotban teszt hurok)
2	8087 interrupt tiltás ON (OFF engedélyezés)
3-4	Az alapkártyán található memória terület
5-6	A monitor vezérlő adapter típusa(i)
7-8	A rendszerhez tartozó floppy diszkek száma



Az alapkártya RAM terület kiválasztása a 3-4 kapcsolókkal



A floppy diszkek számának beállítása a 7-8 kapcsolókkal



* 80x25 mód esetén, színes televíziót, vagy gyengébb minőség monitort használva a karakterek minősége romlik.

** Ebben a beállításban a rendszer 'default'-ja az egyszínű monit vezérlő adapter. Ha 2 display adapter üzemel a gépben, és az egyszí vezérlő adaptert nem monochrom monitorhoz kapcsolták, megfelelő CGA adapter kapcsolóbeállításokat kell használni.

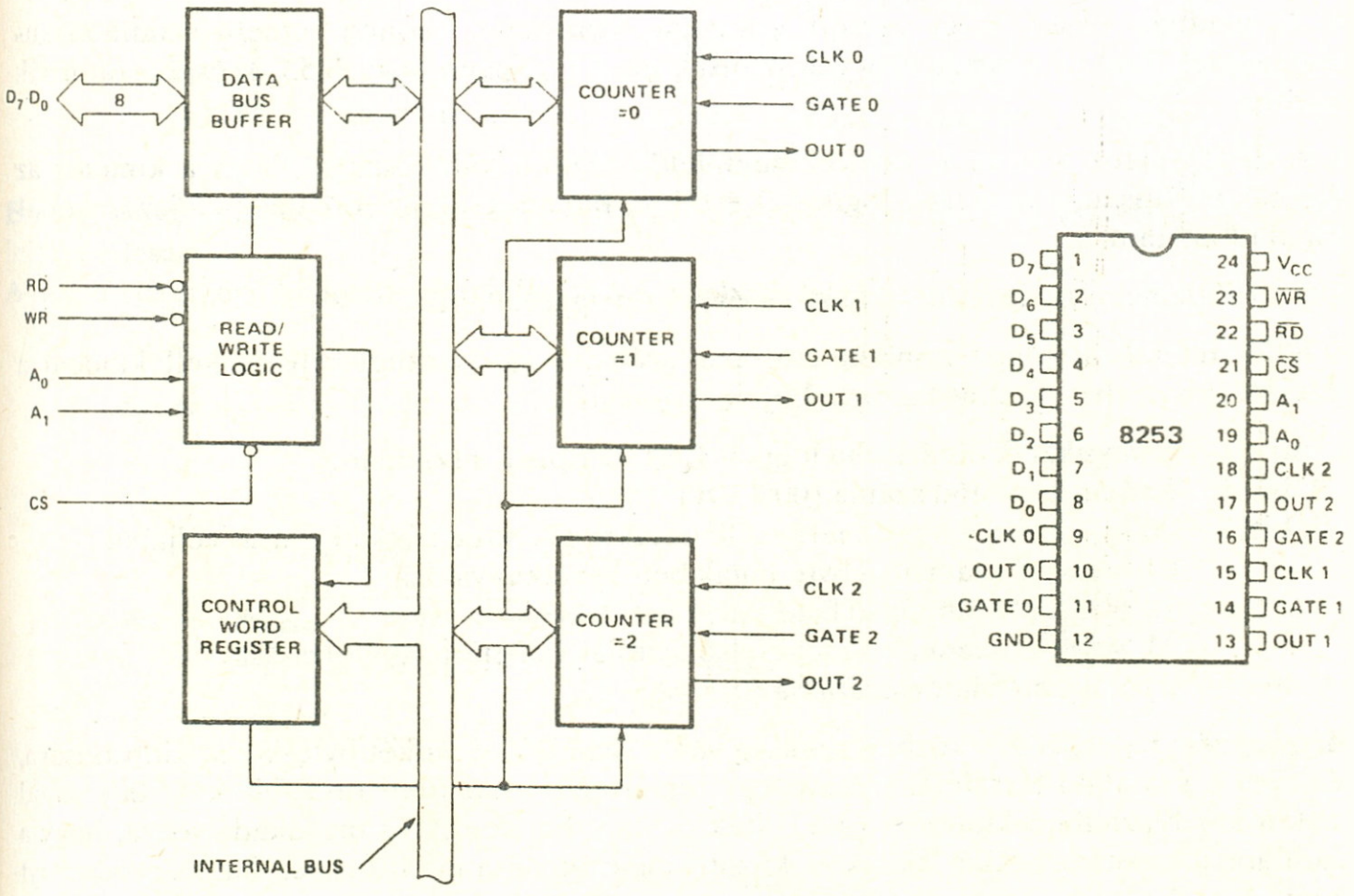
2-13 ábra.
SW1 DIP kapcsoló beállítási lehetőségei.

2.2.7 A 8253 típusú számláló/időzítő és a hangszóró interface

A három-csatornás, 16 bites számláló/időzítő blokk-diagramja a 2-14 ábrán látható. Minden csatornához tartozik egy bemenő órajel (CLK0-CLK2) és egy kimenőjel (OUT0-OUT2), melynek frekvenciáját az órajel és egy programmal beírható, maximum 16 bites szám hányadosa adja. Az órajel-frekvenciák a 0-2MHz közötti sávba eshetnek. Az XT gépeken mindhárom órajel bemenetet a rendszer leosztott órajeléhez kapcsolják. A 8284 óragenerátor PCLK (peripheral clock) kimenete még egy kettes osztó közbeiktatásával kerül a CLK0-CLK2 bemenetekre. Mivel a PCLK jel frekvenciája a rendszer órafrekvenciájának a fele, a 8253 bemeneteire kerülő jelek 4,772727 / 4 = 1,1931817 MHz-es, 50% kitöltésű négyszögjelek.

A 0. csatorna kimenete a 'real time' órajeleket szolgáltatja, melyek a 8259 interrupt vezérlő IR0 megszakítás kérő bemenetére kerülnek. Az 1. csatorna kimenete 15 μsec-ként DMA ciklus kéréssel fordul a 8237A DMA vezérlő 0. csatornájához, RAM frissítési célból. Végül a 2. csatorna kimenete 896 Hz-es négyszögjelet szolgáltat, ami a hangszórót működteti (2-5.5 ábra).

Valamennyi számláló csatorna időkapu bemenettel is rendelkezik (GATE 0-2). Az XT gépeknél csak a 2. csatorna kapubemenetét vezérlik, a többit +5V-ra kötik. A GATE2 bemenet jele vagy engedélyezi (logikai 1) vagy tiltja (logikai 0) a 896Hz-es négyszögjel kiadását.



2-14 ábra.
A 8253 számláló/időzítő blokkdiagramja.

Minden számláló/időzítő csatorna hat különféle módban programozható. Ezek a következők:

0. mód: interrupt az időzítés lejártakor
1. mód: programozható monostabil
2. mód: frekvencia osztó
3. mód: négyszög generátor
4. mód: software triggerelt mintavevő jel előállítása
5. mód: hardware triggerelt mintavevő jel előállítása.

0. mód esetén a csatorna kimenete logikai 0 állapotba kerül a beprogramozott 'n' számú órajel idejére. Az időzítés lejártakor a kimenet újból logikai 1 értéket vesz fel. A 0 -- >1 jelváltás megszakítás kérést hozhat létre. A működést úgy is felfoghatjuk, mintha programmal indítanánk egy monostabil áramkört.
Az **1. mód** nagyon hasonlít az előzőre, azzal a különbséggel, hogy a programozható monostabil indítását most a GATE bemenet 0 -- > 1 jelváltása idézi elő.

A **2. mód** egy 'n'-el osztó számlánc, ahol 'n' a beprogramozott 16 bites szám. A kimenet minden 'n' óraciklus leszámolásakor 1 órajel periódus időre logikai 0-ba vált, egyébként logikai 1. A kimeneti hullámforma asszimetrikus, órajel/n frekvenciájú jel. Az XT gépben a 8253 1. jelű csatornáját használják ebben a módban DMA ciklusok kezdeményezésére DRAM frissítési célból.

A **3. mód**ban olyan 'n'-el osztott számlánc kimenetet kapunk, amely szimmetrikus négyszögjel. Ezt az üzemmódot választották az XT gépben lévő 8253 0. és 2. csatornák részére.

A **4.** és **5. mód**ok a 0. és 1. módok megfelelői, azzal a különbséggel, hogy a kimenet az időzítés indításakor nem vált logikai 0 értékre, hanem csak az időzítés befejezésekor, 1 órajel időtartamra.

A GATE bemenettel indított módok (1. és 5.) retriggerelhetők.

A 8253 inicializálásához minden csatorna számára egy utasítás-byte-ot kell küldeni a vezérlőregiszterbe. Az utasítás-byte bitjeinek jelentése:

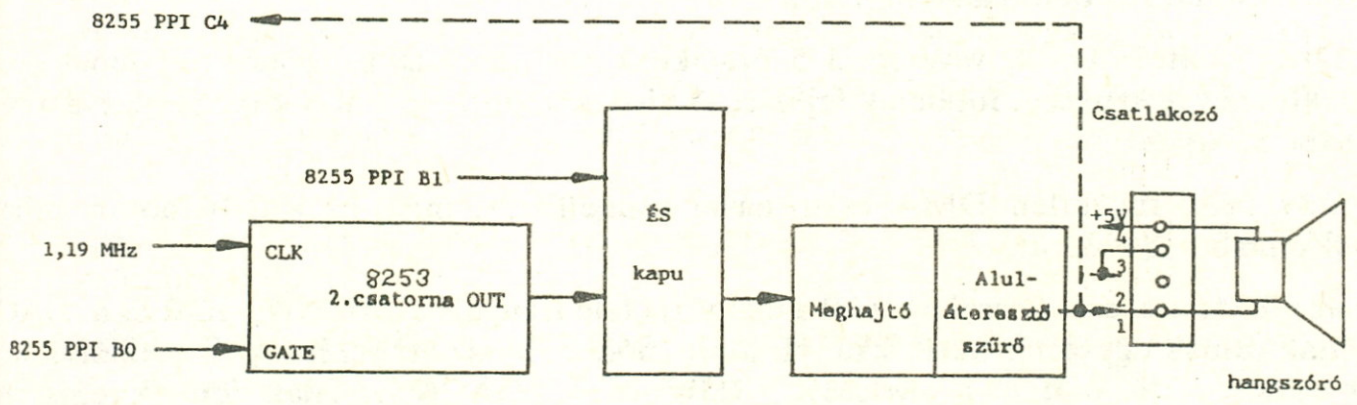
0. bit: ha logikai 0, bináris, ha logikai 1, BCD típusú a számlánc.
1-3. bitek: a választott mód száma (000-101)
4-5. bitek: 00 = a számláló pillanatnyi állapotát regiszterbe írják kiolvasás céljából
01 = az 'n' szám alsó byte-jának betöltése/kiolvasása
10 = az 'n' szám felső byte-jának betöltése/kiolvasása
11 = az 'n' szám alsó majd felső byte-jának betöltése/kiolvasása.
6-7. bitek: a választott csatorna száma (0, 1 vagy 2)

Megjegyzés: ha működés közben szükség van a számláló mindkét byte-jának kiolvasására, célszerű a számláló tartalmát először egy regiszterbe elmenteni (4-5. bitek = 00), majd ezután két részletben kiolvasni (4-5. bitek = 11). Ez az eljárás megakadályozza, hogy a számlánc állapotának a két kiolvasás közötti megváltozása miatt esetleg hamis eredményt kapjunk.

A gép bekapcsolásakor a ROM BIOS a három számláló/időzítő csatorna közül először az 1. jelűt állítja az előzőekben ismertetett 2. üzemmódba, majd egy rövid tesztprogrammal ellenőrzi a számláló működőképességét. A kimenet végleges frekvenciáját a DMA vezérlő inicializálásakor állítja be a 66 kHz-es (15 μs-os) értékre. Ettől kezdve a DRAM-ok frissítése biztosítva van. Ezután következhet a 8253 0. csatornájának az ellenőrzése, majd az időméréshez ('real time' óraütések) a kezdeti értékek beállítása. A következő néhány programsor a számlálók/időzítők beállítását mutatja:

```
TIMER    EQU    40h
TIM-CTL  EQU    43h
TIMER0   EQU    40h
         MOV    AL,54h          ; 1. időzítő beállítása, 2. mód, n = 1 byte (low)
         OUT    TIMER+3,AL      ; vezérlő utasítás kiadása
         MOV    AL,18           ; n megadása
         OUT    TIMER+1,AL      ; utasítás kiadása: n = 0012h
                                ; beállítása az 1. időzítőben
                                ; f = 1,19MHz/18 = 66 kHz
         .
         .
         .
         MOV    AL,36h          ; 0. időzítő beállítása, 3. mód, n = 2 byte
         OUT    TIMER+3,AL      ; vezérlő utasítás kiadása
         MOV    AL,0            ; n megadása: 65536
         OUT    TIMER,AL        ; utasítás kiadása: n alsó byte betöltés
         OUT    TIMER,AL        ; utasítás kiadása: n felső byte betöltés
                                ; f = 18,2 Hz
```

Ezek a programrészek minden ROM BIOS verziónál fellelhetők a számlálók/időzítők inicializálására.
A nagyon egyszerű hangszóró interface-t a 2-15 ábrán láthatjuk.



2-15 ábra.
Hangszóró interface.

A hangszórót két jel vezérelheti. Az egyik a 8253 2. csatornájának a kimenete, a másik pedig a 8255 PPI B 'port'-jának az 1-es bitje (B1). A 8253 2. számláló/időzítő kimenőjelére ezenkívül a 8255 PPI B0 bitje is hatással lehet a GATE bemeneten keresztül. A 2-1/4"-os hangszórót az egyik, másik, vagy mindkét meghajtó jel vezérelheti. A meghajtó fokozat egy aluláteresztő szűrőn keresztül kapcsolódik a hangszóróhoz. A szűrő a négyszögjelek magasabb felharmónikusait levágja. A csatlakozó egy 4 pontos Berg csatlakozó, melynek a 2. pontja kihasználatlan. (Egyes XT gépeknél erre a pontra is +5V tápfeszültséget vezettek, így egy kétpontos csatlakozó ellendarab elegendő a hangszóró bekötéséhez.)

2.2.8 A 8237A DMA (Direct Memory Access) vezérlő, és az IBM XT DMA csatornái

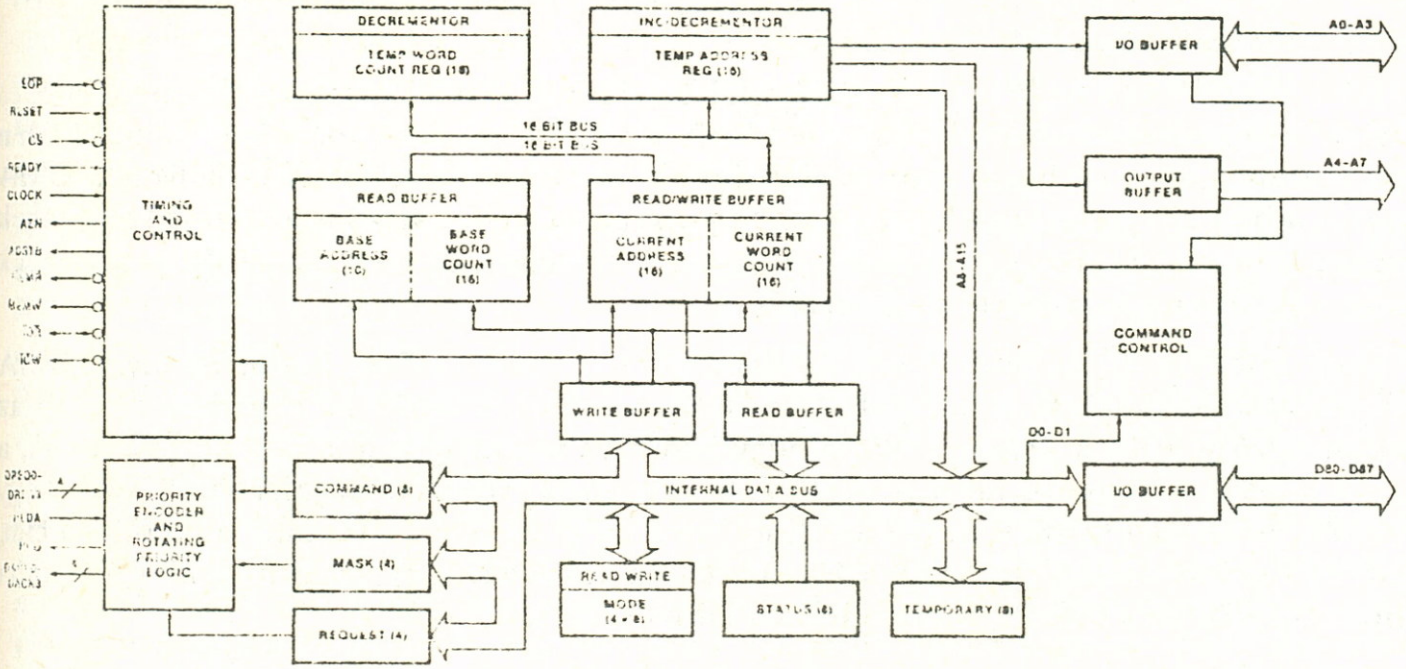
A 8237A típusú DMA vezérlő a legbonyolultabb intelligens I/O eszköz, azok közül, amik az XT alapkártyán fellelhetők. Helyesebb talán speciális célú mikroprocesszorról beszélni, melynek egyetlen feladata, hogy adat-blokkokat szállítson egyik memória helyről a másikra, vagy egy I/O eszköz és a memória között, a lehető legnagyobb sebességgel. Az adat blokkok legnagyobb mérete 64 Kbyte lehet. Összehasonlítva a régebben elterjedt 8257 típusú DMA vezérlővel, két lényeges változásra hívnánk fel a figyelmet: a memória-memória közötti átviteli lehetőségre, amellyel a 8257 nem rendelkezik és a maximális adatblokkok méretére, amely a 8257 esetében 16 Kbyte.

Tekintsük át először egy I/O eszköz és a memória közötti adatátvitelt. Ha ez programozott úton történik, a 8088 mikroprocesszor először váró hurokban figyeli az új adat megjelenését, majd beolvassa azt az I/O 'port'-ról az AL regiszterbe, végül az AL tartalmát beírja valamelyik memória helyre. Mindez DMA vezérlőn keresztül a következőképpen zajlik le: A 8237A egyszerűen vár, és semmit sem csinál mindaddig, míg egy logikai 1 értékű jel meg nem jelenik a DREQ (DMA REQUEST) vonalán. Ez a jel az új adat megjelenéséről informál. Ekkor a 8237A nyugtázó jelet küld az I/O eszköz számára a DACK (DMA ACKNOWLEDGE) vonalon, melynek hatására az az adatot a buszra helyezi. Ezzel egyidőben a 8237A DMA vezérlő a címvonalakra teszi az aktuális memóriarekesz címét, ahol az adatot tárolni akarja, és kiadja a -MEMW jelet, amely a buszon levő adatot beírja a memóriába.

Egy byte átvitele DMA vezérlővel 5 óraciklust igényel, míg programozott módban 29 óraciklusra van szükség. Időben kifejezve, az első esetben 1 μs, a másodikban 6 μs időt vesz igénybe az átvitel.

A 8237 négy független DMA csatornával rendelkezik, mint az a 2-16 ábrán látható blokksémából is kiderül.

Ha az adatátvitel az I/O eszköz felől a memória felé irányul, a -MEMW, -IOR és a -DACK vonalak válnak egyszerre aktívakká. Hasonló módon, ha az átvitel iránya a memóriától az I/O eszköz felé mutat, a -MEMR, -IOW és a -DACK vonalak lépnek egyszerre működésbe. Mivel az I/O eszköz megszólításakor a címvonalakon cím is jelen van, előfordulhatna hogy a periféria esetenként I/O címként értelmezné a memória címet. Ennek megakadályozására az AEN (Address ENable) vonal DMA ciklusok alatt logikai 1 értéket vesz fel. Előírás, hogy valamennyi I/O címdekóder figyelje az AEN vonalat is, és tiltsa az I/O eszköz kiválasztását, ha az AEN vonal logikai 1 állapotban van.



2-16 ábra.
A 8237A DMA vezérlő blokkdiagramja.

Memóriából memóriába irányuló adatátvitel két 8237 DMA csatornát és az előzőekkel ellentétben nem egyidejű, hanem szekvenciális működést igényel, mivel a memóriahelyeket nem lehet kijelölni a DACK vonalakkal (ellentétben az I/O eszközökkel), hanem egymás után címezni kell őket. (Előbb a 'forrást', majd a 'rendeltetést'.) Egy byte átviteléhez nyolc óraciklus szükséges. Programozott módon ez a művelet 17 óraciklust igényel, ha a CPU MOVSB utasítását használjuk. A 8237 0. csatornáját a 'forrás' címzésére az 1. csatornáját pedig a 'rendeltetés' címzésére használhatjuk. A forráscímek növekedhetnek, csökkenhetnek, vagy állandóak maradhatnak, a 8237-hez küldött I/O utasításnak megfelelően. A forráscímet állandónak tartva, kényelmesen feltölthető akár 64 Kbyte-nyi memória is a forráscímen lévő értékkel. Ugyanezt a feladatot a CPU STOSB utasítása is elvégzi közelítőleg ugyanannyi idő alatt.

Az eddigiekből megállapítható, hogy 8086/88-as rendszerekben a memóriából-memóriába irányuló adatátvitel DMA vezérlő segítségével nem sokkal gyorsabb mint a programozott mód.

Az XT gépekben a 8237A DMA vezérlőt kizárólag memória és I/O eszközök közötti adatátvitelre használják.

Egy speciális blokk-átviteli üzemmódban az egymásutáni byte-ok átvitele csak két óraciklust igényel (420 ns). A DMA ciklusok idejére a 8088 CPU felfüggeszti tevékenységét.

A blokkdiagramot szemlélve a 8237A sok kivezetése emlékeztet a processzor hasonló megfelelőire. Ahogy a 8088-nál, az adatbusz itt is 8 címbittel multiplexelt (most a 8-15 címkivezetések közösek a 0-7 adatkivezetesekkel). A cím/adat buszról az ADSTB

(Address Strobe) jel 0 -- > 1 váltása a címinformációt egy 74LS373 típusú regiszterbe írja (kapcsolási rajz 2-5.5 ábra, U15 pozíció).

A -IOR, a -IOW és a négy legalsó címvezeték kétirányú, a CPU leválasztott címbuszától és a 8288 rendszervezérlő által dekódolt -IOR és -IOW jelektől eltérően. A 8237 ezen kivezetésein DMA ciklusok alatt megfordul az adatirány, mivel a CPU helyett a DMA vezérlő állítja elő ezeket a jeleket. A 8237 programozásakor viszont az A0-A3 vezetékek szolgáltatják az I/O címet a 16 belső regiszter eléréséhez, és a -IOR, -IOW vezérlőjelek pedig a 8288 rendszervezérlőtől érkeznek.

A négy DREQn bemenetet lehet azokhoz az I/O eszközökhöz kötni, amelyek DMA szolgáltatást igényelnek. A megfelelő DACKn kimenetek adják az engedélyező jelet az átvitelt kérő periféria számára. Ha a négy bemenet valamelyikén átvitel-kérés érkezik, a 8237A logikai 1 állapotba helyezi a HRQ (hold request) vonalát. Ez jelzi a CPU-nak, hogy a 8237A át kívánja venni a busz vezérlését, és a CPU helyezze 'tri-state' állapotba a cím, adat, és vezérlő vonalait. A 8088 a HLDA (hold acknowledge) vonal logikai 1 állapotba állításával jelzi ha a buszt felszabadította a DMA vezérlő részére.

A 8237A -EOP kivezetése szintén kétirányú. Mint kimenet logikai 0 értéket vesz fel, ha a megvalósított DMA ciklusok száma eléri a beprogramozottat. Bemenetként használva, valamely külső eszköz a folyamatban lévő DMA ciklusokat lezárhatja, ha logikai 0-t helyez erre a vonalra.

Eddig egy általános áttekintést adtunk a DMA vezérlőről és működéséről a blokkséma alapján. Most az inicializálást és a belső működést szeretnénk kicsit részletesebben ismertetni. (Az INTEL katalógusokban mindez pontosabban megtalálható.)

Minden DMA csatornához tartozik egy 6-bites üzemmód utasításszó és négy 16-bites regiszter. A regiszterek a pillanatnyi címet és a pillanatnyi DMA ciklusszámot, valamint a kezdőcímet és a megkívánt ciklusszámot (az átviendő byte-ok számát) tartalmazzák. A DMA működés megkezdése előtt a kezdőcím regisztert és a ciklusszám regisztert az inicializáló program feltölti a szükséges értékekkel. Ezek az értékek automatikusan átíródnak a pillanatnyi címet és a pillanatnyi ciklusszámot tároló regiszterekbe is.
Ahogy a működés kezdetét veszi, minden byte átvitele után a pillanatnyi cím értéke eggyel növekszik vagy csökken (attól függően, hogy az aktuális csatornára vonatkozó üzemmód utasítás 5. bitje 0 vagy 1) a pillanatnyi ciklusszám pedig eggyel csökken. Az átvitelek mindaddíg folytatódnak, amíg a pillanatnyi ciklusszám regiszter értéke 0-ra nem csökken (terminal count), vagy míg a -EOP vonalon egy külső eszköz a működést meg nem szünteti. A kezdőcím és ciklusszám regiszterek továbbra is megtartják beprogramozott értéküket, így a pillanatnyi regiszterek azonnal automatikusan feltöltődnek a kiindulási adatokkal az átviteli ciklusok lezárása után, ha az üzemmód regiszter 4. bitje 1.
Az üzemmód regiszter 3. és 2. bitjei az átvitel irányát határozzák meg: 01 esetén írás (I/O -- > memória), 10 esetén olvasás (memória -- > I/O) ciklusok jönnek létre.

A regiszter 7. és 6. bitjei az átviteli módot definiálják. A négy lehetséges eset: 00 = teszt célokra, 01 = egyes mód, 10 = blokk mód, 11 = kaszkád mód.
Ha az átlagos átviteli sebesség kisebb mint 1 byte/gépi ciklus (a gépi ciklus 4 órajel periódus), akkor az egyes módot célszerű választani. Ebben az üzemmódban minden DREQ kérésre egy byte átvitelére van lehetőség. Ezt a módot használják a DRAM-ok

frissítésére, ahol minden 72 órajel alatt egyszer jön létre DMA kérés, vagy a floppy diszk vezérlőkkel való adatcserére, mikoris 32 mikroszekundumonként ismétlődnek a kérések. A floppy diszk vezérlővel a kommunikáció elég lassú, így programozott úton is megvalósítható lenne. A billentyűzet és a 'REAL-TIME' óra interrupt kérései miatt azonban célszerűbb volt a DMA módot alkalmazni.

A blokk átviteli mód akkor célszerű, ha az I/O eszköz közelítőleg ugyanolyan gyors működésre képes mint a DMA vezérlő. Ha az I/O eszköz valamivel lassúbb mint a 8237, a READY vonalat logikai 1-be téve várakozási óraciklusokkal meghosszabbítható a DMA ciklus, hasonlóan a 8088 CPU-nál megismertekkel. Az első byte átvitele öt órajel periódust, a továbbiak azonban csak három periódust (630 nsec az XT gépen) igényelnek. Amennyiben a rendszer sebessége megengedi, egy speciális, két óraciklust (420 ns) igénylő blokk átviteli mód is választható.
Az előzőekben már volt szó arról, hogy a 8237 DMA vezérlő maximum 64 Kbyte nagyságú memória címterületet tud kezelni. Ez azt jelenti, hogy önmagában nem képes kihasználni a rendelkezésre álló 1 Mbyte-os címtartományt. Azért, hogy a DMA lehetőséget a teljes címtartományban biztosítsák, egy 4x4 bites 74LS670 típusú regiszter-file-t építettek a gépbe, az úgynevezett DMA szegmens-regisztereket, amelyek a DMA csatornák felső 4 címbitjeinek (A16-A19) a tárolására szolgálnak. A 74LS670 regiszter-file egyidejű írás és olvasás műveletek elvégzésére alkalmas, mivel külön adat ki- és bemenetekkel, és külön író és olvasó címvonalakkal rendelkezik. Az egyidejű írás/olvasás sajátosság nincs felhasználva, tekintettel arra, hogy a regiszterek beírására a DMA inicializáláskor kerül sor kiolvasásukra pedig a DMA ciklusok alatt. A külön író és olvasó címvezetékek viszont nagyon kényelmesen kihasználhatók. A WA és WB író címbemenetekre az A0 és A1 jelek mint I/O címbitek kerülnek, a chip beírást engedélyező jelet (-WRT DMA REG) pedig az I/O dekóder állítja elő (kapcsolási rajz 2-2.5 ábra). Az RA és RB olvasó címbemenetekre az egyszerűség kedvéért a DACK3 és DACK2 jeleket vezették, hogy DMA ciklusok alatt az éppen átvitelt kérő csatornához tartozó szegmens-regiszter váljon aktívvá. Ez a megoldás azt is jelenti, hogy a 0. és 1. csatornákhoz ugyanazt a regisztert rendelték hozzá, ami viszont nem okoz problémát, mivel a DRAM frissítést végző 0. csatorna számára nincs jelentősége a hozzátartozó szegmens-regiszter tartalmának.

A DMA szegmensregiszterek I/O címei a következők:

0. csatorna: 83h
1. csatorna: 83h
2. csatorna: 81h
3. csatorna: 82h

Végül összefoglaljuk a DMA csatornák felhasználását az XT gépekben. A 0. csatorna a DRAM-ok frissítésére szolgál. Az 1. csatornát a felhasználó saját célokra használhatja. A 2. csatorna a floppy diszk vezérlővel, a 3. csatorna pedig a winchester diszk vezérlővel való kommunikációra van fenntartva. A DREQn és DACKn DMA vezérlőjeleket a DREQ0 kivételével az I/O csatorna csatlakozóira vezették, hogy az adapter kártyák részére hozzáférhetők legyenek.

Amennyiben 64 Kbyte-nál nagyobb memóriaterületet igénylő DMA átviteleket valósítunk meg, a programnak módosítani kell a DMA szegmensregiszterek értékeit a 64 Kbyte-os sávhatárok átlépésekor.

2.2.9 Az IBM XT I/O csatornája

Az alapkártyán kívül további egységek szükségesek ahhoz, hogy egy XT gép használhatóvá váljon. Még a legszerényebb konfiguráció esetén is szükség van video vezérlő kártyára, floppy diszk vezérlőre és soros/parallel adapterekre. Ilyen egységek csatlakoztatására hozták létre az alaplemezen az I/O csatornát. Lényegét tekintve az I/O csatorna egy 62 pontos parallel busz, amely 8 csatlakozóval rendelkezik. A bedugható kártyák 12" (305 mm) hosszúak és 4,5" (14 mm) magasak lehetnek.

Az I/O csatornát úgy is felfoghatjuk, mint a 8088-as processzor közvetlen buszának a kiterjesztését demultiplexelés után, interrupt bemenetek és DMA lehetőségek hozzáadásával.

A buszvonalak a következőképpen csoportosíthatók:
- 8 kétirányú adatvonal
- 20 címvonal
- 6 megszakítást kérő bemenet
- vezérlőjelek memória és I/O írás/olvasási célokra
- órajelek és időzítő vonalak
- 3 DMA csatorna részére vezérlő jelek, továbbá a memória frissítést időzítő jel
- I/O csatorna ellenőrző vonal
- tápfeszültségeket biztosító vonalak.

Négy tápfeszültség van az I/O csatlakozókra kivezetve: +5Vdc, -5Vdc, +12Vdc és -12Vdc.

Az I/O csatorna 'ready' vonala lehetővé teszi az alaplemez együttműködését lassú perifériákkal vagy memóriákkal. Ha ezt a vonalat nem használják, a CPU által generált memória írás vagy olvasás ciklusok 4 órajelet, azaz 0,84 us-ot igényelnek byte-onként. Valamennyi CPU által létrehozott I/O ciklus részére pedig 5 órajel periódus, azaz 1,05 us szükséges byte-onként. 1 byte átvitele DMA csatornán szintén 5. óraciklust, azaz 1,05 us-ot igényel. A DRAM frissítő ciklusok 4 óraperiódus időtartammal 15 mikroszekundumonként fordulnak elő, és a maximálisan elérhető busz sebességet közelítőleg 7%-al csökkentik.

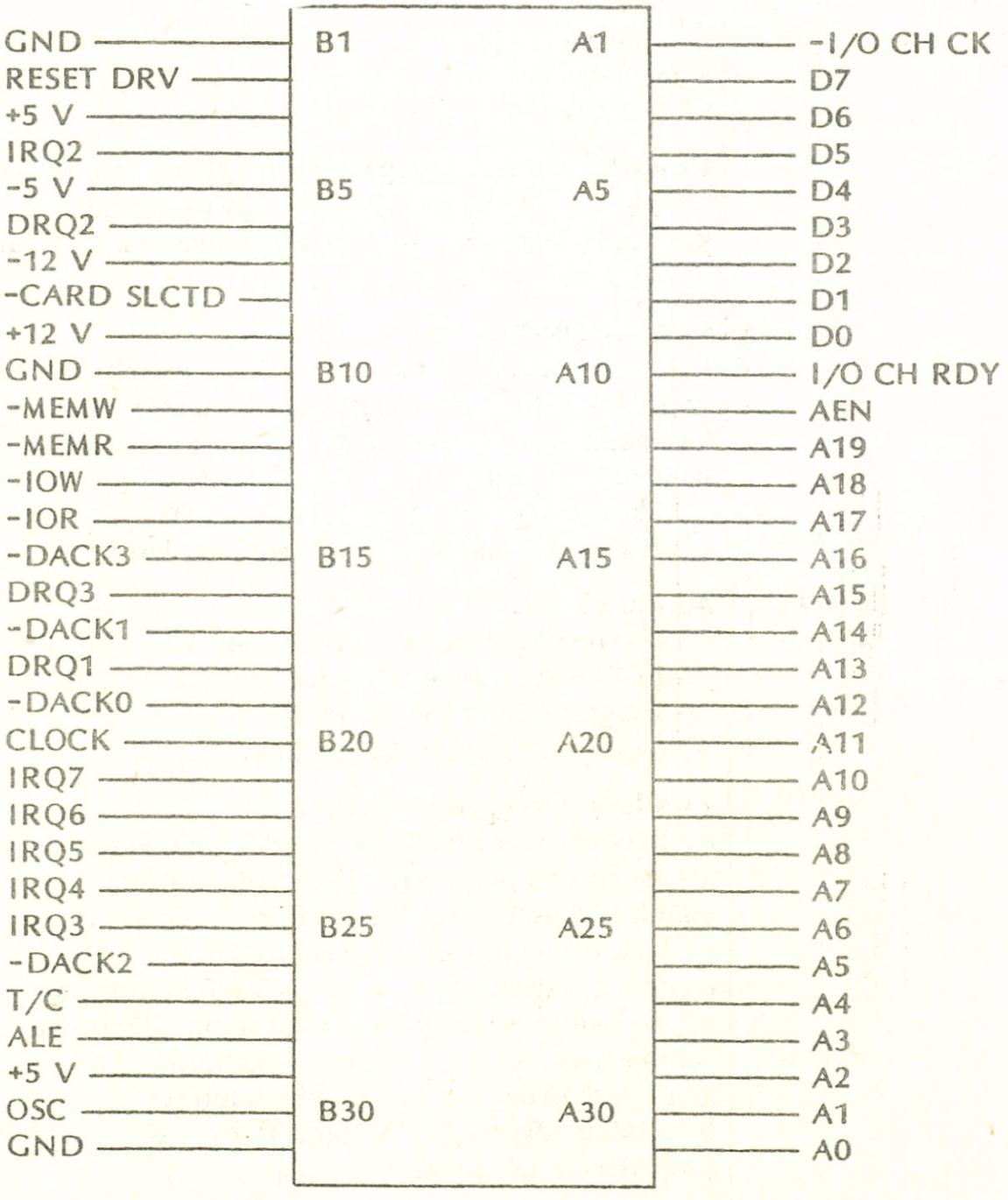
Az I/O csatornát úgy tervezték, hogy 768 I/O cím felhasználására van mód az adapter kártyákon.

Az I/O csatorna felügyeleti vonal ('channel check') a CPU-t informálja valamelyik adapter kártya meghibásodásáról, nem maszkolható interrupt kérést (NMI) okozva. Például a memória bővítő kártya paritás hibát jelezhet ezen a vonalon. Az I/O csatorna bufferelt jelei elegendő teljesítményűek ahhoz, hogy a nyolc csatlakozó vonalait meghajtsák, ha csatlakozó pontonként két LS (Low-power Schottky) terhelést feltételezünk.

A J8 jelű I/O csatlakozón egy olyan jelet is találunk, amelynek a többi hétnél nincs jelentősége: ez a 'kártya kiválasztva' (card selected) vonalon megjelenő jel. Ezt a jelet egyedül a J8 csatlakozóban elhelyezett kártya állíthatja elő, jelezve a rendszer részére, hogy a kártya üzemképes és kiválasztásra került.

Az I/O csatlakozók jelkiosztása a 2-17 ábrán látható. A csatlakozó alkatrészoldalról van ábrázolva, és az A1, B1 jelű pontok esnek közelebb a gép hátlapjához.

A jelek elnevezéseinél a név elé helyezett - jelzés a jel logikai 0 aktív állapotára utal (negált jelek).



2-17 ábra.
Az IBM XT I/O csatorna csatlakozó jelei az alaplemez alkatrészoldala felől nézve.

A következőkben részletesen ismertetésre kerülnek az egyes vonalak funkciói. A felsorolás három oszlopot tartalmaz: az elsőbe a jelek nevei, a másodikba a jelek irányai az alapkártya felöl nézve (I = bemenet, O = kimenet), a harmadikba a jelek leírásai kerülnek.

A jel neve	I/O	A jel leírása
OSC	O	**Oszcillátor:** nagysebességű órajel, 70ns periódusidővel (14,31818 MHz), 50%-os kitöltési tényezővel.
CLK	O	**A rendszer órajele:** a harmadára osztott oszcillátor jel, 210 ns periódusidővel (4,77 MHz), és 33%-os kitöltési tényezővel.
RESET DRV	O	**Ez a jel a rendszer alapállapotba állítására**, inicializálására szolgál a gép bekapcsolásakor, vagy tápfeszültség kimaradáskor. Az órajel 1 -- > 0 éléhez szinkronizált, és logikai 1 az aktív állapota.
A0-A19	O	**Címbitek 0-19:** a rendszer memóriáinak és I/O eszközeinek címzésére szolgálnak. A 20 címvonal 1Mbyte memóriaterület használatát teszi lehetővé. A0 a legalsó bitet (LSB), A19 pedig a legfelső bitet jelöli (MSB). A címeket vagy a processzor, vagy a DMA vezérlő generálja. A vonalak aktív állapota logikai 1.
D0-D7	I/O	**Adatbitek 0-7:** ezek a vonalak képezik a processzor, a memória és az I/O eszközök adat-buszát. D0 a legalsó bitet (LSB), D7 a legfelső bitet (MSB) jelöli. A vonalak aktív állapota logikai 1.
ALE	O	**Cím-leválasztás engedélyezése:** a 8288 busz vezérlő állítja elő ezt a jelet, amely az alapkártyán a processzor által szolgáltatott címek leválasztására, és regiszterekbe írására szolgál. A jel lefutó 1 -- > 0 éle rögzíti a címet a regiszterben. A jel logikai 0 időtartama alatt a címregiszterek kimenetei stabil állapotban vannak, és érvényes címeket szolgáltatnak az I/O csatornán. Ennek jelzésére tették hozzáférhetővé az I/O adapterek részére. (Az AEN jellel együtt használják.)

A jel neve	I/O	A jel leírása
-I/O CH CK	I	**I/O csatorna ellenőrzés:** ez a vonal jelzi a processzor részére ha az I/O csatornára csatlakoztatott memóriákban vagy perifériákban paritás hiba áll elő. A jel logikai 0 állapota paritás hibára utal.
I/O CH RDY	I	**I/O csatorna készenlét:** készenlét esetén a vonal logikai 1 állapotban van. Az adapter kártyák ezt a vonalat logikai 0-ba húzhatják, hogy meghosszabbítsák a memória vagy I/O ciklusokat. E vonal felhasználásával lassú eszközök is hozzákapcsolhatók az I/O csatornához. Bármelyik lassú eszköznek azonnal le kell húznia ezt a vonalat, mihelyt felismerte a címét és az író vagy olvasó vezérlőjeleket. 10 órajel periódus időtartamnál hosszabb ideig nem szabad a vonalat logikai 0 állapotban tartani. A gépi ciklusok (memória vagy I/O) az órajel periódusok egész számú értékeivel kerülnek meghosszabbításra (210 ns).
IRQ2-IRQ7	I	**Interrupt kérések 2-7:** ezek a vonalak informálják a processzort arról, hogy valamelyik I/O eszköz fel szeretné hívni magára a figyelmet. Az interrupt vonalak közül az IRQ2 a legmagasabb, az IRQ7 pedig a legalacsonyabb prioritású. Megszakítás kérés akkor jön létre, ha az IRQ vonal logikai 0-ból logikai 1-be vált. A vonalat mindaddig logikai 1-ben kell tartani, míg a kérést nem nyugtázza a processzor (az interrupt kiszolgáló rutinon keresztül).
-IOR	O	**I/O olvasási parancs:** ez a vonal arra készteti a kiválasztott I/O eszközt, hogy helyezze az adatbuszra az adatot. A -IOR jelet vagy a processzor, vagy a DMA vezérlő állítja elő. A vonal aktív állapota logikai 0.
-IOW	O	**I/O írás parancs:** ez a vonal szolgáltatja egy I/O eszköz részére a vezérlőjelet a CPU által az adatbuszra helyezett információ beolvasásához. A -IOW jelet vagy a processzor, vagy a DMA vezérlő állítja elő. A vonal aktív állapota logikai 0.
-MEMR	O	**Memória olvasás parancs:** memóriát vezérlő vonal, melynek hatására a kiolvasott

A jel neve	I/O	A jel leírása
		memória cella tartalma az adatbuszra kerül. A -MEMR jelet vagy a processzor, vagy a DMA vezérlő állítja elő. A vonal aktív állapotban logikai 0.
-MEMW	O	**Memória írás parancs:** ez a vonal szolgáltatja a memória részére a vezérlőjelet, az adatbuszon lévő információ beírásához. A -MEMW jelet vagy a processzor, vagy a DMA vezérlő állítja elő. A vonal aktív állapotban logikai 0.
DRQ1-DRQ3	I	**DMA kérelem 1-3:** ezeket a vonalakat az I/O eszközök használják aszinkron módon DMA ciklusok kezdeményezésére. A DRQ1 vonal rendelkezik a legmagasabb, a DRQ3 pedig a legalacsonyabb prioritással. A kérést a DRQ vonal logikai 1 aktív állapota idézi elő. A DRQ vonalat mindaddig logikai 1 értéken kell tartani, míg a -DACK vonalon a kérés nyugtázása meg nem érkezik.
-DACK0-DACK3	O	**DMA nyugtázás 0-3:** ezek a vonalak nyugtázzák a DMA kérelmeket (DRQ1-DRQ3), és frissítik a DRAM memóriákat (-DACK0). A vonalak aktív állapota logikai 0.
AEN	O	**Címzés engedélyezés:** ez a jel lekapcsolja a processzort és az egyéb eszközöket az I/O csatornáról (felszabadítja a buszt) DMA átvitel céljából. Amikor a vonal aktív állapotban van (logikai 1), a DMA vezérlő működtetheti a cím- és adatbuszokat, az olvasó és író vonalakat (memória és I/O).
T/C	O	**DMA ciklusok lezárása:** ha bármelyik DMA csatornán az átvitt byte-ok száma eléri a DMA vezérlőbe beprogramozottat, ezen a vonalon egy pozitív impulzus jelenik meg. A jel aktív állapota logikai 1.
-CARD SLCTD	I	**Kártya kiválasztva:** ezt a vonalat a J8 I/O csatlakozóba helyezett kártyák aktivizálhatják. Ez a vonal jelzi az alapkártya részére, hogy a J8 helyre csatlakozó eszköz kiválasztásra került, és erről a helyről adatot lehet beolvasni, vagy erre a helyre adatot lehet küldeni az alapkártya megfelelő bufferein keresztül. Bár a nyolc I/O csatlakozón ez a pont összekötött, csak a J8 csatlakozó esetében értelmezett. A vonalat open kollektoros meghajtóval kell vezérelni.

Az I/O csatorna a felsorolt jeleken kívül tápfeszültség csatlakozási pontokkal rendelkezik. A tápfeszültség kivezetések:

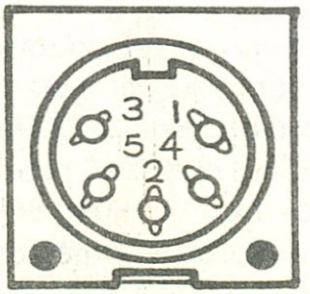
+5Vdc	+5%	a B3 és B29 csatlakozópontokon
-5Vdc	+10%	a B5 csatlakozóponton
+12Vdc	+5%	a B9 csatlakozóponton
-12Vdc	+10%	a B7 csatlakozóponton
GND		a B1, B10 és B31 csatlakozópontokon.

2.2.10 Az alapkártyán található egyéb csatlakozók és beállítási lehetőségek, az XT TURBO alaplemez

A tápfeszültség bevezetésére szolgáló, P1 és P2 jelű csatlakozók bekötését a 2.2.1 fejezetben találjuk (2-2. táblázat). Az I/O csatorna csatlakozóinak a jelkiosztását a 2.17 ábra mutatja (2.2.9 fejezet).
A rendszer konfigurációt meghatározó SW1 kapcsoló beállítási lehetőségeit a 2-13 ábrán láthatjuk (2.2.6 fejezet).

A billentyűzet és a hangszóró interface ugyan említésre került, de a csatlakozók bekötését most közöljük (2-18 ábra). Az IBM XT alapkártyán más csatlakozási lehetőséget nem találunk. Jumper átkötésekkel választhatjuk ki a beépítésre kerülő ROM, EPROM és DRAM memóriák méretét és típusát. (Bővebb információ ezekről az átkötésekről a géppel együtt szállított felhasználói kézikönyvekből nyerhető.)

Csatlakozási pont	I/O	A jel neve	Jelszint
1	I	A billentyűzet órajele (+TTL)	+5Vdc
2	I	A billentyűzet adatjele (+TTL)	+5Vdc
3	O	Általános törlő jel (-TTL)	
		Tápfeszültségek	
4		GND	OV
5		+5V	+5Vdc



5 pontos DIN csatlakozó

A billentyűzet csatlakozója

Csatlakozási pont	A jel neve
1	Adat kimenet
2	-
3	GND
4	+5Vdc

A hangszóró csatlakozója

2-18 ábra.
A keyboard és a hangszóró csatlakozók bekötése.

A 2.1.1 fejezetben már utaltunk az XT gépek úgynevezett TURBO-XT változatára, mely a rendszer órajel-frekvenciájának az átváltásával (4,77 MHz-ről 8 MHz-re) a működési sebességet mintegy 70%-kal megnövelheti. A programok zöme ebben a gyorsított "TURBO" módban is gond nélkül futtatható, de akad néhány amelynél csak a normál mód (4,77 MHz) a megengedett. (Ide tartozik például a COPYWRITE másoló program.) A TURBO-XT alapkártyán a 8088-2 típusú processzort és az I/O chip-ek gyorsabb működésű változatait találjuk.

Az IBM XT-hez viszonyítva további csatlakozó pontok és átkötési lehetőségek kerültek az alaplemezre. (Ezek közül néhány más XT gépeken is fellelhető.) A TURBO/NORMÁL mód vagy hardware, vagy software úton váltható. Ennek kijelölésére szolgál egy jumper beállítási lehetőség az alapkártyán. Ha a software átváltási módot választják, megfelelő ROM BIOS-ról kell gondoskodni (pl. ERSO/DTK TURBO BIOS melynél a Ctrl/Alt/- billentyűk egyidejű lenyomása eredményez átváltást mindkét irányban). Hardware átváltási mód esetén az alaplemez egy két pontos kivezetéséhez csatlakozó kétállású kapcsoló ON állapota a TURBO, OFF állapota a NORMÁL módot állítja elő. Egy további csatlakozóhoz LED kijelzők kapcsolhatók, melyek a gép bekapcsolt állapotát és a TURBO módban való működést mutatják. A billentyűzet működése reteszelhető a gép előlapján található zárral, amely szintén az alaplemezhez csatlakozik. A rendszer nyomógombbal is újraindítható, az óragenerátor -RESET vonalának a lehúzásával, tehát nem szükséges a gépet ki-be kapcsolni az alaphelyzet létrehozásához rendszerhiba esetén.

2.3 Az AT alapkártya ismertetése

Az IBM személyi számítógépek másik nagy ágát az AT gépek képviselik. Mint ismert, ezek a gépek az Intel 80286 processzor család elemeire épülnek. Az óragenerátor a 82284, a rendszervezérlő a 82288 típusszámot viseli, amelyek a 8088/86 család elemeinek itteni megfelelői. A 80286 processzorral együttműködő aritmetikai processzor a 80287. Bár az első részben a 8088/86 és 80286 processzorok és a hozzájuk tartozó aritmetikai társprocesszorok ismertetésre kerültek, az AT gép blokkdiagramjának és működésének részletezése előtt érdemes azokat a speciális jellemzőket felemlíteni, amelyek csak a 80286-os CPU-k sajátjai, és amelyekkel a 8088-as mikroprocesszorok nem rendelkeznek. Az IBM AT mellett az IBM XT 286 és a COMPAQ Deskpro-286 gépek is a 286-os processzor családra épültek.

A 80286 két olyan megjelenési formával, vagy speciális üzemmóddal rendelkezik, amely a működést tekintve lényegbevágó. Az egyik módban a 80286 CPU teljesen úgy viselkedik, mint egy 8088-as processzor, míg a másik mód lehetővé teszi a beépített extra lehetőségek, speciális tulajdonságok teljes kihasználását. A két működési mód "valós címzési mód" (real address mode) és "védett virtuális címzési mód" (protected virtual address mode) néven ismert.
"Valós módban" a 80286 hasonlóan működik mint egy 8088 (pontosabban mint egy 8086), mivel 16 bites adatbusszal rendelkezik és 16 bites szó szervezésű memóriákkal is képes együttműködni. A dolgoknak azonban ez a lényegtelenebb része; a lényeges az, hogy valós módban a programokat ugyanúgy kezeli mint egy szabványos XT gép CPU-ja.) A speciális tulajdonságok, extra képességek ebben az esetben rejtve maradnak, így a 286-os processzorral épített gép teljesen kompatibilis a normál XT-vel. A programok azonban

lényegesebben gyorsabban futnak mint az XT gépek esetében. Ennek két oka van: először, a 286-os CPU belső kialakítása korszerűbb, ami azt jelenti, hogy kevesebb lépésben, kevesebb óraciklus alatt hajt végre egy utasítást, mint a 8088. Például egy szorzást a 8088 120 óraciklus alatt, míg a 80286 20 óraciklus alatt hajt végre. Másodszor, a 286-os CPU gyorsabb órajel frekvenciával működik. Az eredeti IBM AT modellt nézve ez 6 MHz-et jelent, ami 25%-kal több mint a normál XT 4,77 MHz-es órajele. Manapság az AT gépek különböző sebességű változatai vannak forgalomban: az órajelek 6, 8, 10 vagy 12 MHz frekvenciájúak lehetnek, és általában átkapcsolhatók két érték között. Átlagosan az AT gépek 10 MHz-es változatai hatszor, nyolcszor gyorsabban hajtanak végre egy programot, mint a normál XT.

A 80286 CPU utasításkészlete felülről kompatibilis a 8088 utasításaival. Ez azt jelenti, hogy rendelkezik valamennyi 8088 utasítással, de tartalmaz néhány további utasítást is. Ezen új utasítások egy része közvetlenül a 8088 utasításkészletét bővíti néhány jól használható új típussal (pl. PUSHA-POPA: push all, pop all, vagy az új SHIFT/ROTATE utasítások, ahol az elmozdulások bitértéke külön megadható). Másik (nagyobb) részük a "védett virtuális módban" használatos.

A "védett mód" lényege, hogy a 286-os CPU-nak olyan belső tulajdonságai kerülnek előtérbe, amelyek lehetővé teszik, hogy egyidőben több program működjön teljes biztonsággal. Hogy ez teljesüljön, négy lehetőség vagy tulajdonság kihasználására van szükség. Ezek sorra: a védelem, a kiterjesztett memória (extended memory), a virtuális memória, és az egyidőben több feladat elvégzésére való képesség (multi-tasking).

A védelem biztosítja azokat a korlátokat, amelyek megakadályozzák, hogy egy program más programokkal, vagy magával az operációs rendszerrel kerüljön zavaró kapcsolatba (interferencia). Egy XT gépben egy "elkalandozó" program megzavarhatja az operációs rendszer vagy más programok működését, sőt súlyosabb esetben a rendszer "összeomlását" is előidézheti.

A 286-os védett üzemmódja lehetővé teszi az operációs rendszer részére, hogy megakadályozza, ha egy program nem a számára kijelölt memóriaterületet kívánja használni, vagy éppen az operációs rendszer működését akarja megzavarni. Addig amíg egy darab programot futtatnak a gépen, nincs nagy jelentősége annak, ha a program valami vad dolgot művel és blokkolja a gépet. Ha azonban a számítógép egyidőben több feladattal foglalkozik, nagyon megnő annak a jelentősége, hogy megvédjék az ilyen jellegű meghibásodásoktól. A 80286 processzor sajátsága, hogy erre lehetőséget ad.
Az XT gépek 1 millió byte memóriaterületet tudnak kezelni. Bár ez első hallásra soknak tűnhet, néha ennél többre is szükség van. A 80286 CPU "védett módban" megnöveli a címterületet. Először is 16 millió byte címzését tudja elvégezni közvetlenül a címvonalain keresztül, ami azt jelenti, hogy max. 16 MByte nagyságú memória helyezhető el a gépben. Az 1 MByte feletti részt kiterjesztett memóriának (extended memory) nevezték el. Másodszor, egy látszólagos memóriát képes szimulálni (virtual memory), amely jóval nagyobb lehet, mint ami a valóságban fizikailag megvalósítható. A virtuális memória címterülete 1 gigabyte-ig terjedhet (10^9 byte).

Több program egyidejű futtatása esetén a 80286-os processzor gyors és megbízható átkapcsolási lehetőséget biztosít a programok között (multi-tasking). Egy-egy

időpillanatban a processzor csak egy program utasításaival foglalkozik, de az átkapcsolások miatt úgy tűnik, hogy valamennyi aktivizált programot működésben tartja. Tulajdonképpen bármely számítógép, így az XT is megkísérelheti a "multi-task" mód megvalósítását, mint ahogy erre van is példa (MLINK, DESQVIEW programok), de bizonyos hardware jellemzők megléte nélkül ezt a feladatot igazán jól megoldani nem lehet (például memóriavédelem, gyors átkapcsolások). A 80286 "védett módja" az egyetlen az Intel processzor családban, amely biztosítja a hardware feltételeket egy igazán jól működő "multi-task"-os üzemmód megvalósításához.

Bár ezek a feltételek a "védett módban" rendelkezésre állnak, pillanatnyilag mégsem annyira hasznosak, mint hinnénk. Ennek alapvető oka az, hogy azoknak a programoknak, amelyeket ebben a környezetben kívánnak működtetni, figyelembe kell venniük a védett mód bizonyos alapszabályait. A jelentősebb PC programok többsége azonban nem így készült, és nem vették figyelembe, hogy a gép kapacitását esetleg meg kell osztani több program között. Ezenkívül a PC DOS operációs rendszert sem tervezték úgy, hogy védett módban is használható legyen.

Mindaddig, míg a fontosabb programok zöme nem kompatibilis a védett móddal és nem létezik egy ebben a környezetben széleskörűen elterjedt operációs rendszer, a védett mód előnyeinek a széleskörű kihasználására nincsenek meg a feltételek. A nagyobb memória címterületet néhány program kihasználja (pl. memória diszk emulációs programok), de a védett mód lehetőségeit ezek a programok sem használják ki igazán.

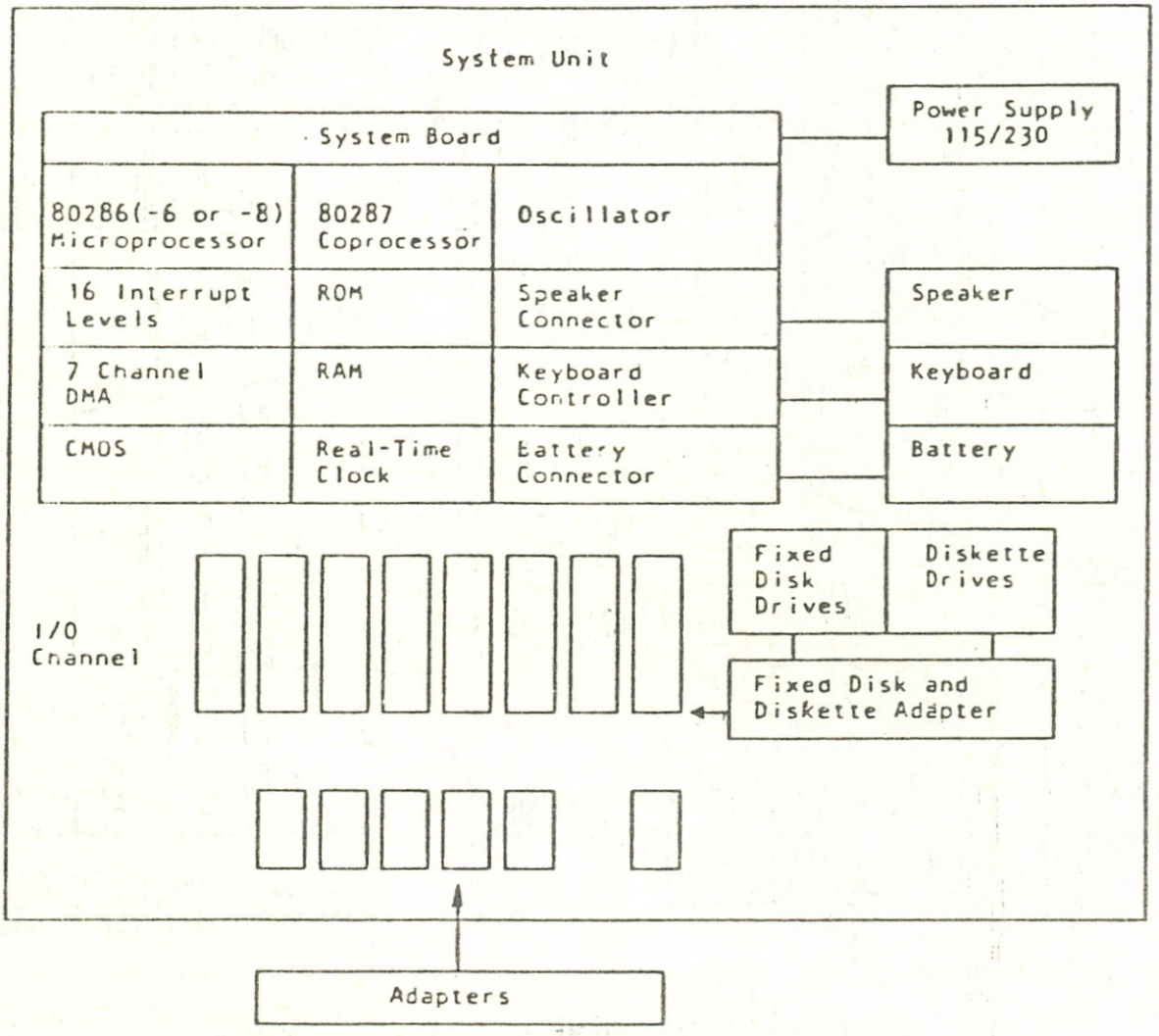
Az AT alapkártya ismertetésénél szeretnénk felhasználni az XT alapkártyánál már leírtakat. Ezért nem írjuk le teljes részletességgel az egyes áramkörök működéseit, hanem csak azokkal foglalkozunk, amelyek az XT-től eltérnek, vagy az ott látottaknál bővebb lehetőségeket nyújtanak. A memóriáknál kitérünk a memória bővítési lehetőségekre, mind a kiterjesztett (extended), mind a kibővített (expanded) memóriákra.
Az IBM AT személyi számítógép központi része a 'system unit', amely alatt az alapkártyát, egy 200 Watt teljesítményű tápegységet, telepet a CMOS memória és óraáramkör részére, floppy és winchester diszk meghajtókat és diszkeket, valamint a billentyűzetet kell érteni. A doboz hátoldalán találhatók a monitor, a billentyűzet és az I/O adapter kártyák hátlap-csatlakozói.

2.3.1 Az AT alaplemez leírása

Az alaprendszer blokk diagramja a 2-19 ábrán, az alkatrészek elhelyezkedését az alaplemezen pedig a 2-20 ábrán láthatjuk. Az alaplemezen a következő funkcionális egységek helyezkednek el:

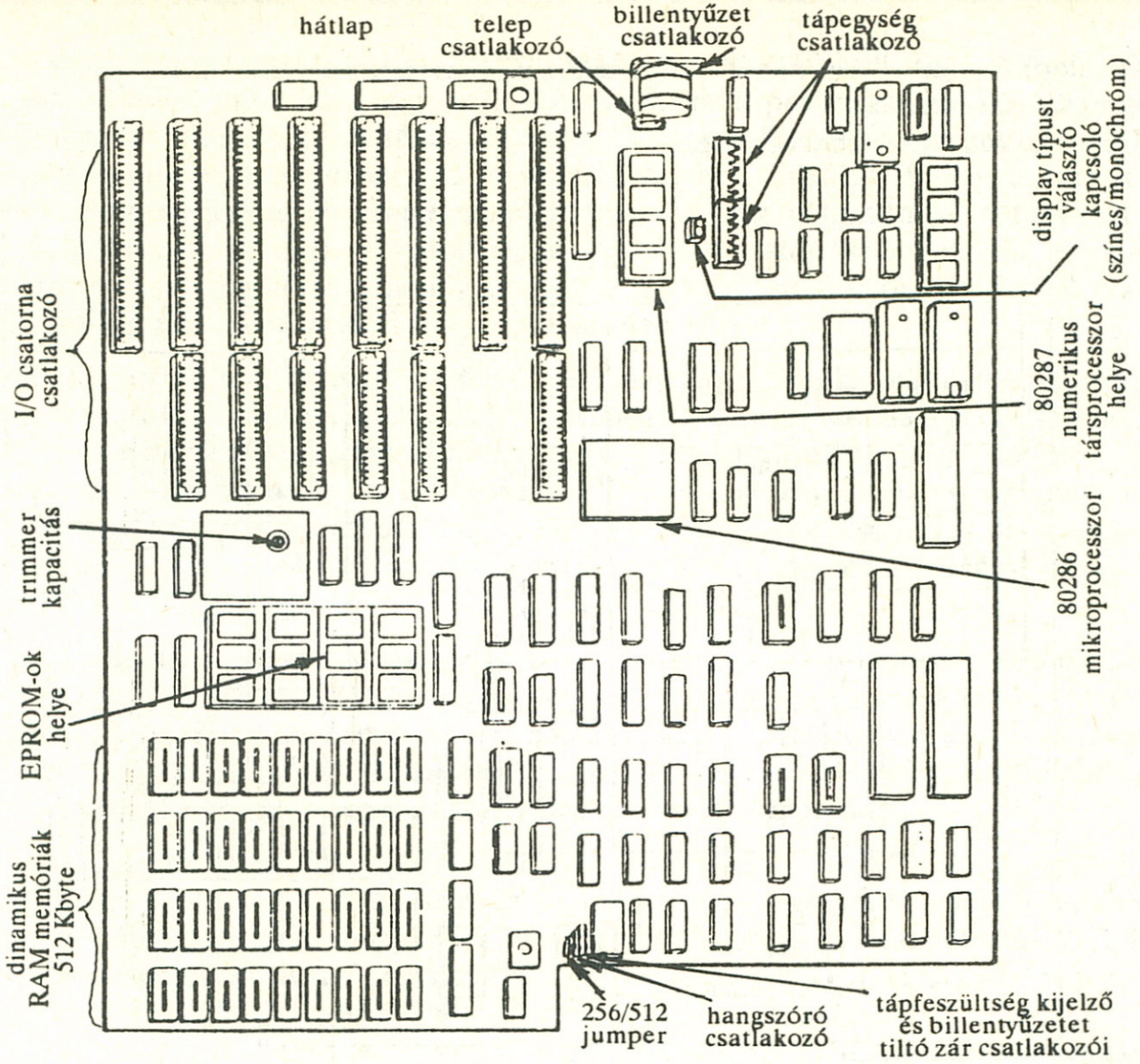
- a 80286 processzor és 80287 aritmetikai processzor, valamint az ezek működését segítő és kiszolgáló áramkörök
- ROM memóriák
- RAM memóriák
- CMOS memória és óraáramkör telepes táplálással
- 7 csatornás DMA áramkörök
- 16 interrupt csatorna

- az alapkártyán elhelyezkedő I/O eszközök (számláló/időzítő, billentyűzet interface, hangszóró interface)
- I/O csatorna 8 csatlakozóhellyel.



2-19 ábra.
Az alaprendszer blokkdiagramja.

Az alaplemez közelítőleg 12" x 13" (305 x 330 mm) méretű és többrétegű nyomtatott áramköri technológiával készült. A tápfeszültségek csatlakozásai megegyeznek az XT gépnél leírtakkal (2-2 táblázat), és szintén 8 csatlakozási helyet találunk az I/O adapter kártyák részére. Az I/O csatlakozási felület azonban eltér az XT-nél megismertekkel, mivel kibővítésre került egy további 2x18 pontos csatlakozóval, míg az eredeti 2x31 pontos csatlakozó megtartotta jelkiosztását. Így lehetővé vált hogy az XT-re tervezett adapter kártyák az AT gépen is működtethetők legyenek. A 2x18 pontos kiegészítő csatlakozókat nem minden I/O helyre szerelték be. A J1 és J7 pozíciójú csatlakozók mellé nem került 2x18 pontos csatlakozó, csak a helye található meg (2-20 ábra). Vannak ugyanis olyan mechanikai méretekkel rendelkező XT adapter kártyák, amelyek nem dughatók be a gépbe, ha a kiegészítő csatlakozó is jelen van. Az XT alaplemez nyolc bites DIP kapcsolósora egyetlen kapcsolóra redukálódott. Ez a monitor típusának a kiválasztására szolgáló kapcsoló. Az egyik állás a színes, a másik az egyszínű monitorra utal. Több kap-



2.20 ábra.
Az AT alapkártya alkatrész elrendezési rajza.

csolóra nincsen szükség, mivel a rendszer kiépítettségének jellemzői software úton meghatározhatók, másrészt teleppel ellátott CMOS RAM-ban tárolják.

A 80286 processzor többféle tokozásban került forgalomba (68 pontos LCC = Leadless Chip Carrier, vagy PGA = Pin Grid Array), és így a különböző gyártók által készített AT-k processzorai tokozás és foglalat tekintetében eltérőek lehetnek. A 80287 aritmetikai társprocesszor itt is opcionális: a helye megvan, és a beültetett foglalatba utólag is behelyezhető.

Az EPROM-ok, ROM-ok részére négy foglalatot alakítottak ki, a RAM-ok pedig négy sorban helyezhetők el. Az XT-től eltérően az alaplemezre építették a külön teleppel ellátott, valós időt mérő óraáramkört és CMOS RAM chip-et (MC 146818). A 64 byte-os memória 14 rekeszét az óra használja, a fennmaradó részben viszont az AT gépek esetén a rendszer konfigurációjára vonatkozó információkat tárolnak. A telepet a dobozhoz erősítik, csatlakozásra az alapkártyán levő 4 pontos csatlakozó szolgál. Hasonlóan helyezik el a hangszórót is. Az I/O csatorna 14,31818 MHz OSC jelét előállító kvarcoszcillátor

frekvenciája finoman hangolható egy trimmer kondenzátor segítségével, hogy a színes televíziók részére szükséges színsegédvívő burst jel pontosan beállítható legyen.

2.3.2 Az AT alapkártya blokkdiagramja

Az eredeti IBM AT blokkdiagram a 2-21 ábrán látható. Az egyes részek ismertetése előtt rövid tájékoztató a rendszer sebesség jellemzőiről 6 és 8 MHz-es órajelet feltételezve (a 8 MHz-es órajelre vonatkozó értékek zárójelben találhatók):

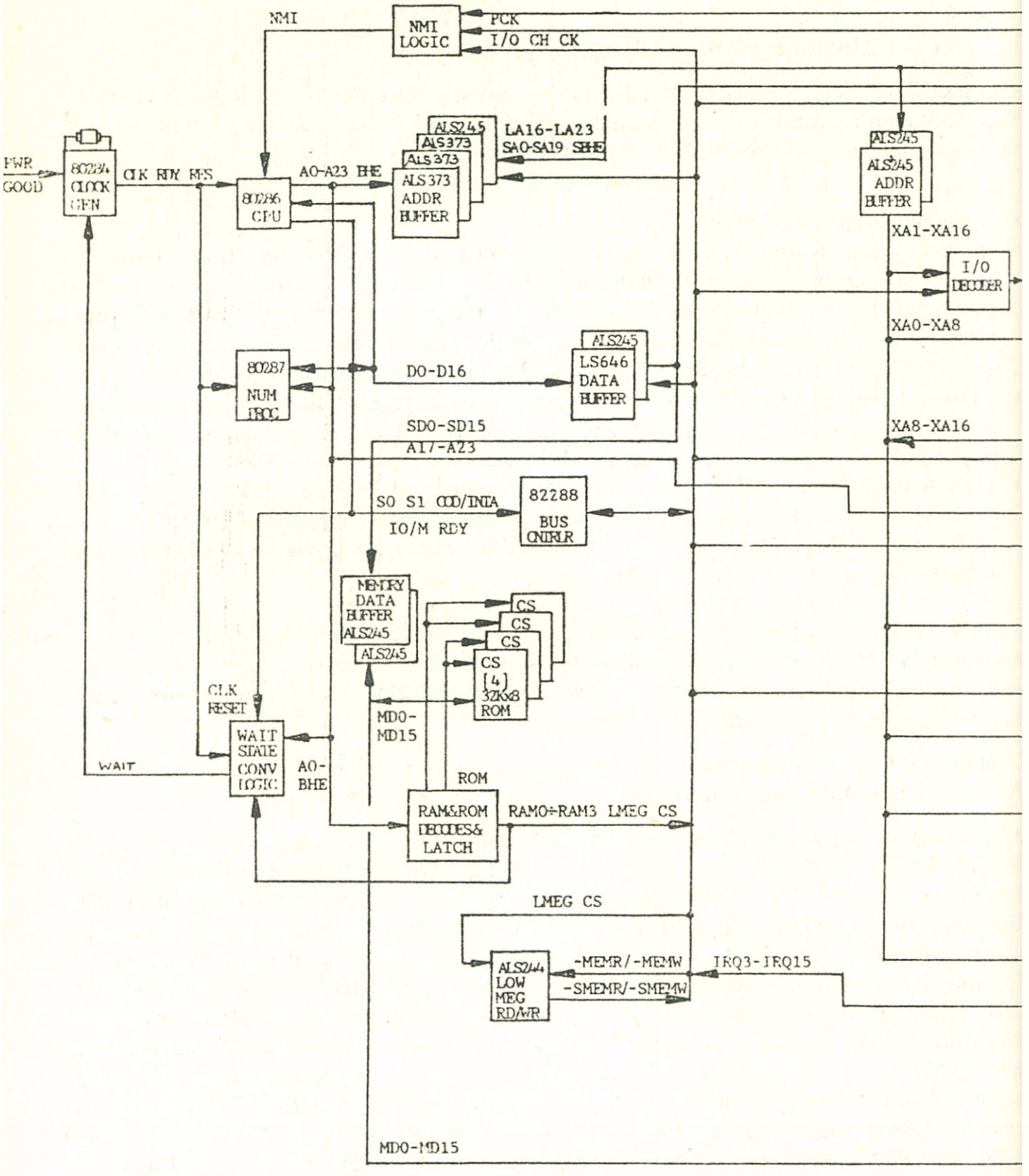
- az óraciklus 167 (125) nsec;
- egy busz ciklus 3 órajel időtartamú (1 várakozási ciklus beiktatásával), így 500 (375) nsec-os 16 bites processzor ciklusidő érhető el. 8 bites busz-műveletek 8 bites eszközökkel 6 órajel periódust igényelnek (4 várakozási ciklus), így a CPU ciklusidő 1000 (750) nsec-ra adódik. 16 bites busz-műveletek 8 bites eszközökkel 12 órajel periódust vesznek igénybe, 2000 (1500) nsec alatt (10 várakozási ciklus);
- a DRAM frissítő áramkör 6 MHz órajellel dolgozik. Egy frissítési ciklushoz 8 órajel periódus szükséges. 4 msec alatt 256 frissítési ciklust kell végrehajtani, amiből az következik, hogy a frissítések a busz sávszélességét 8,7 (6,5) %-kal csökkentik;
- a DMA vezérlő 3 (4) MHz-es órajelet kíván. Valamennyi DMA átviteli ciklus öt DMA órajel alatt megy végbe, így egy átvitelre 5 x 333,3 = 1666 nsec idő adódik (5 x 250 = 1250 nsec). A 0-3 DMA csatornák 8 bites, az 5-7 csatornák pedig 16 bites átvitelekre vannak fenntartva.

A 80286 processzor lokális buszára csatlakoznak a cím-puffer regiszterek és az adat meghajtók, a 82288 típusú buszvezérlő, a 80287 aritmetikai processzor és részben a DMA szegmens regiszter. A processzorok részére az órajelet a 82284 típusszámot viselő óragenerátor közvetlenül állítja elő (12 MHz).

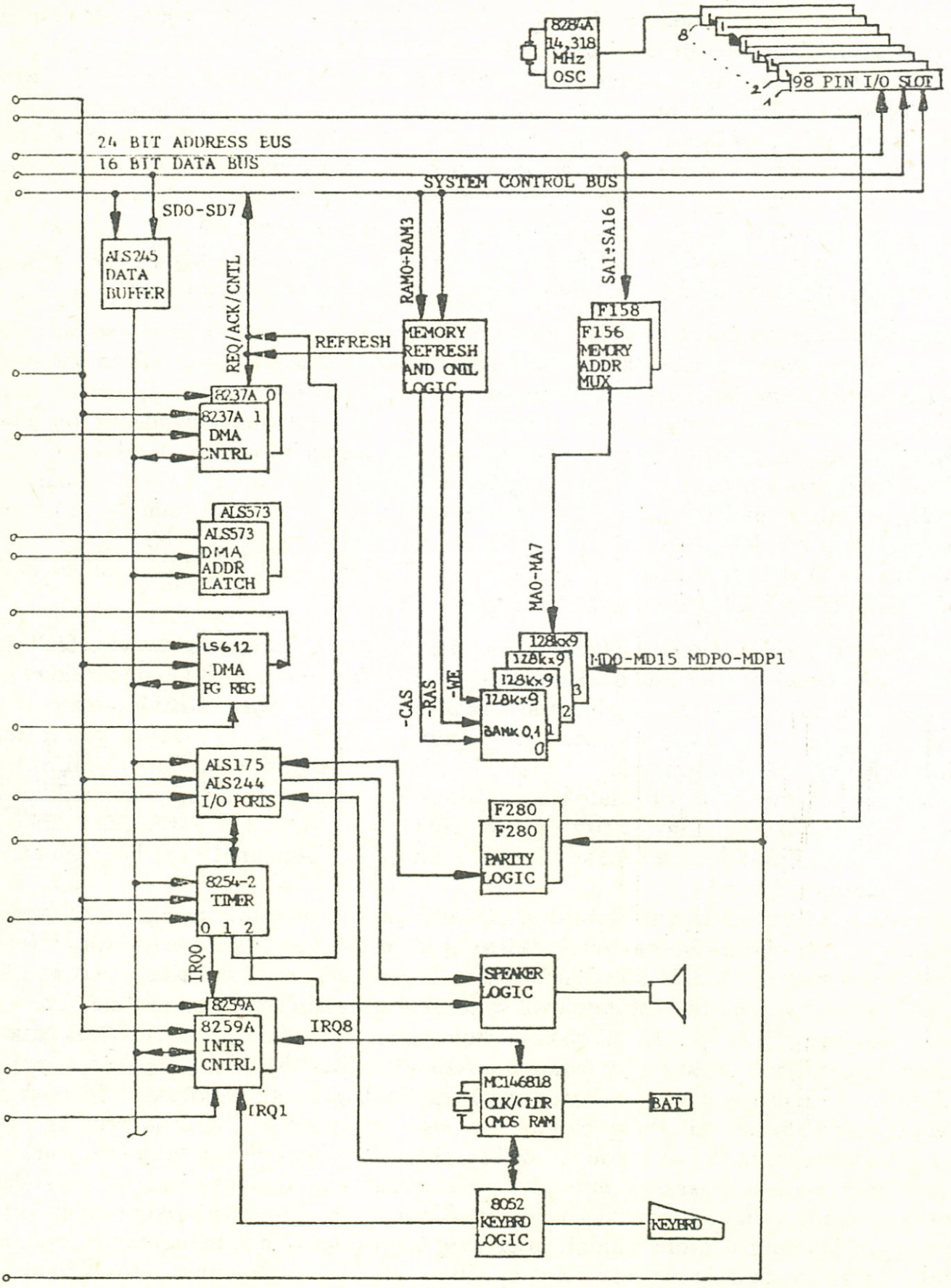
További osztó áramkörök hozzák létre a rendszer-órajelet, a processzor ciklusait mutató PCLK jelet és a DMA vezérlő órajelét.

A CPU-nak és az aritmetikai processzornak különböző sebességű változatai léteznek. A bemenő órajelet a 80286 kettővel, a 80287 pedig hárommal osztja magában a chip belsejében. A belső órajelekre vonatkoztatva a 80286-os processzornak 6, 8, 10 és 12,5 MHz sebességű, a 80287 aritmetikai processzornak pedig 5, 6, 8 és 10 MHz sebességű típusai kaphatók. A 82284 óragenerátort rendszerint 12, 16, 20 vagy 25 MHz-es kvarcokkal látják el a különböző sebességű verziókhoz. Az első IBM AT gépekbe a 6 MHz frekvenciájú processzort építették be, 12 MHz-es kvarcoszcillátorral. Manapság egyre több olyan AT van forgalomban, amely a 10 vagy a 12,5 MHz sebességű CPU-val dolgozik, és ezt figyelembe véve alakították ki a gép egyéb órajeleit is. A gép működtethető kisebb órajel frekvenciával is, ha a konfigurációt beállító SETUP program segítségével, az órafrekvenciát átkapcsoljuk. Ezeknél a gépeknél természetesen törekedtek arra, hogy az egyéb áramkörök, például az I/O eszközök, interrupt vezérlők, stb. is lehetőleg gyors működésű típusokból kerüljenek megvalósításra.

Az alaplemezen található még egy 8284A típusú óragenerátor, amely az I/O csatorna OSC vonalára ad 14,31818 MHz frekvenciájú jeleket.



2-21 ábra.
Az IBM AT alapkártya blokkdiagramja (I.).



2-21 ábra.
Az IBM AT alapkártya blokkdiagramja (II.).

A 82284 óragenerátor nem csak a processzorok alapidőzítését állítja elő, hanem a -READY vonalán keresztül várakozási ciklusokat kényszeríthet a CPU-ra. A várakozási állapotokat egy logikai áramkör generálja (WAIT STATE CONV LOGIC elnevezéssel a blokkdiagramon), melynek bemeneteire a 82288 rendszervezérlő jelei, a ROM, RAM és I/O kiválasztó jelek, valamint a 8 vagy 16 bites átvitelt vezérlő A0 és BHE jelek kerülnek. A logikai áramkör kimenete jut az óragenerátorra, amely az órajellel szinkronizált -READY jelet továbbítja a processzorhoz. A várakozási ciklusokat előállító dekóder a 2.3.2 pont elején leírtak szerint különböző számú várakozási ciklust állít elő a különféle busz-műveleteknél. A 82284-es integrált áramkör utolsó funkciója, hogy órajelhez szinkronizált, általános érvényű törlőjelet hozzon létre a rendszer számára.

A 80286 CPU lokális busza (a CPU kivezetéseire közvetlenül csatlakozó vonalak) a 8088 CPU-val ellentétben nem multiplexelt. Tehát nincsenek a CPU-nak olyan kivezetései, amelyek egyik időpillanatban cím, a másikban adat információkat hordoznak. Mivel a kivezetések száma 68-ra nőtt, a 24 címvonal és a 16 adatvonal egymástól függetlenül elhelyezhető volt. A lokális buszra csatlakozó elemek száma minimális. Ezek közé tartozik az aritmetikai processzor, amely szoros együttműködésben van a CPU-val, és néhány olyan integrált áramkör, melyek számára nagyon fontos, hogy a címinformációt minél előbb megkapják. Ezek a ROM és RAM dekóderek, továbbá a várakozási ciklusokat előállító áramkörök. A DMA szegmens regiszter kimenetei csatlakoznak még ide, amelyek az A17-A23 felső címbiteket állítják elő DMA átvitelek alatt.

A lokális buszt buffer elemek közbeiktatásával megerősítik, és így teszik hozzáférhetővé a rendszer többi részére. A blokkdiagramon a bufferek kimenőjeleit S kezdőbetűkkel látták el, utalva ezzel arra hogy ezek a jelek már a rendszerbuszhoz (System Bus) tartoznak.
Az A0-A19 címbiteket regiszterekbe írják (3 db 74LS373), és így tárolják és erősítik. A címvonalak egy másik csoportját, A17-A23 csak leválasztják és erősítik (74LS245). A processzor címkimenetein az adatok nem állnak rendelkezésre a teljes buszciklus alatt. Egyrészt ez indokolja, hogy regiszterekbe írjuk őket. Másrészt a regiszterekben való tárolás a parallel működés lehetőségét adja, amit az AT gépeknél ki is használnak. Ez a következőképpen értendő:
Tételezzük fel, hogy az n. busz ciklus végrehajtásánál járunk. A címregiszterekben az n. ciklushoz tartozó címek találhatók. A ciklus végefelé azonban a processzor címkimenetein már megjelenik az n + 1. ciklushoz tartozó cím. Habár még az n. ciklusban járunk, a ROM és RAM címdekóderek részére lehetővé válik a következő ciklus dekódolása. Ezért az n. ciklus lezárásakor, az n + 1. ciklushoz szükséges memóriaterületek kiválasztása már rendelkezésre áll, ami feltétlenül gyorsítja a működést, vagy hosszabb memóriahozzáférést engedélyez. Az eljárást szokás "csővonalas címzés"-nek nevezni (pipelined addressing). Az elnevezés onnan származik, hogy az adatokat úgy dolgozzák fel, mintha azok egy csövön haladnának keresztül. A cső bizonyos szakaszainál találunk egy-egy feldolgozó áramkört, amelyek folyamatosan parallel működve részfeladatokat oldanak meg. A működés így sokkal gyorsabbá válhat, mint a hagyományos esetben, amikor tulajdonképpen az történik, hogy megvárják míg a csőbe táplált adat meg nem jelenik a kimenetén, és csak ekkor fognak hozzá a feldolgozáshoz. Ez az itteni esetre úgy lenne lefordítható, hogy megvárnánk míg az n. ciklus lezajlana, majd az n + 1. ciklus címeit a processzor előállítaná, és csak ezután fognánk hozzá a címdekódoláshoz. Bár a parallel működés elve csak minimális mértékben került megvalósításra az AT gépekben, érdemes megemlíteni,

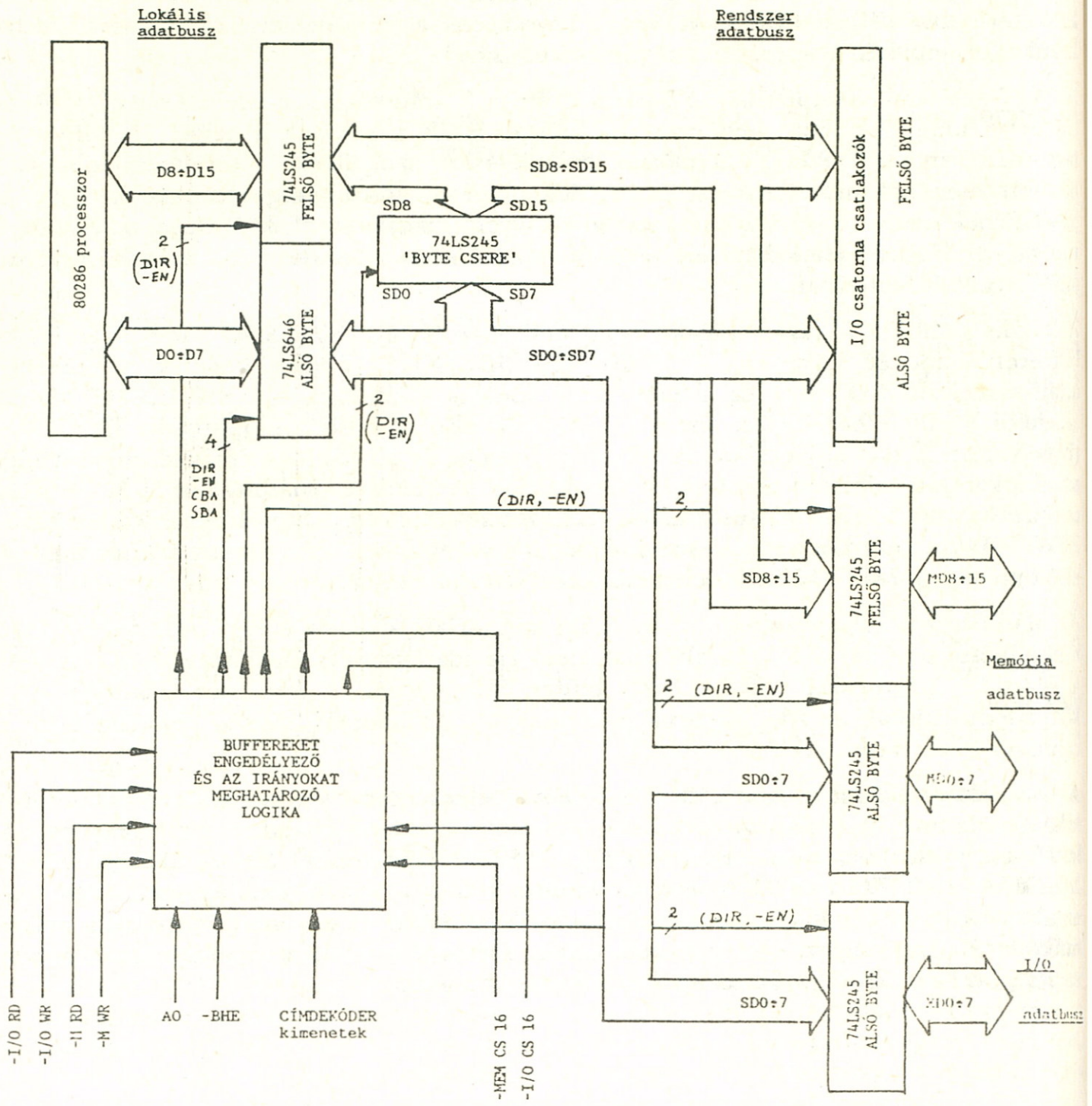
mert ennek eredményeképpen a memóriákhoz való hozzáférési idő úgy nőtt meg, hogy a programok végrehajtási sebessége nem változott. Vagy másképpen fogalmazva, egy adott hozzáférési idejű memóriát feltételezve a gép sebessége nagyobb lett. Mindezt végsősoron a 80286-os processzor belső arhitektúrája teszi lehetővé azáltal, hogy a címkimeneteket úgy időzíti, hogy a következő buszművelet címe még a folyamatban lévő ciklus alatt hozzáférhetővé váljon. Más szavakkal, a következő buszművelet első óraciklusa fedésbe kerül a folyamatban lévő művelet utolsó óraciklusával.

Az A17-A23 vonalakat leválasztó 74LS245 integrált áramkör kimenetein a busz-ciklusnál rövidebb ideig jelennek meg érvényes címek. Ezek a vonalak a teljes 16 Mbyte-os tartományban elhelyezkedő memóriák dekódolására szolgálnak, 1 várakozási ciklussal bővített memória műveletek esetén. A dekóder kimenőjeleit regiszterekbe kell írni a 'BALE' jel lefutó élével. Az A0 legkisebb helyértékű címbitnek különleges jelentősége van: a -BHE (Bus High Enable) vezérlőjellel a 16 bites adatbusz alsó és felső byte-ját jelölik ki az átvitel céljára.

A lokális adatbusz D0-D15 vonalait egy SN74LS646 és egy SN74LS245 integrált áramkör buffereli mindkét irányban. A leválasztott SD0-SD15 vonalak képezik a rendszer adatbuszát. Az SD0-SD7 és a SD8-SD15 vonalak közé még egy SN74LS245 típusú áramkör került elhelyezésre, amely az adatbusz alsó és felső byte-jait tudja felcserélni (byte swap). A blokkdiagramon ez nincs feltüntetve, de a 2-22 ábra részletesen ábrázolja az alapkártyán található adatbuszok kialakítását. A rendszer-adatbuszhoz 8 és 16 bites eszközök csatlakoznak. 8 bitesek az alaplemezen található I/O chip-ek, melyeket az XD0-XD7 I/O adatbusz köt össze egymassal. Az I/O adatbusz és a rendszer adatbuszának az alsó byte-ja egy 74LS245 buffer elemmel kerül szétválasztásra.

Az alaplemezen elhelyezett RAM és ROM memóriák, amelyek 8 vagy 16 bites módban működhetnek, az MD0-MD15 jelű belső memória adatbuszra csatlakoznak. A memória és a rendszer adatbuszát két 74LS245 buffer választja el. A rendszer adatbusz jelei közvetlenül jutnak az I/O csatorna csatlakozóira. Az adapter kártyák feladata hogy a rendszer buszt bufferekkel leválasszák.

A kapcsoló elemként működő "BYTE CSERE" elnevezésű 74LS245 áramkör a 16 bites lokális adatbusz és valamilyen 8 bites adatbusz összekapcsolásakor jut szerephez. A leválasztó buffer elemeket a BUFFEREKET ENGEDÉLYEZŐ ÉS AZ IRÁNYOKAT MEGHATÁROZÓ LOGIKA vezérli. Minden leválasztóhoz tartozik egy vezérlőjel, ami az aktualis adatirány kijelölésére szolgál a bufferen belül, és egy másik, amelyik a buffer működését engedélyezi. A bufferek tehát kapcsoló elemként működnek, amelyek segítségével két adatbusz összekapcsolható vagy elválasztható.



2-22 ábra.
Az AT alapkártya adatbusz kialakítása.

A 8 vagy 16 bites átviteleket az A0 és -BHE vonalak jelölik ki a következő táblázat szerint:

-BHE	A0	A működés leírása
0	0	16 bites szó-átvitel
0	1	8 bites byte-átvitel a lokális adatbusz felső felén (D8-D15)
1	0	8 bites byte-átvitel a lokális adatbusz alsó felén (D0-D7)
1	1	Tiltott állapot, soha nem fordulhat elő.

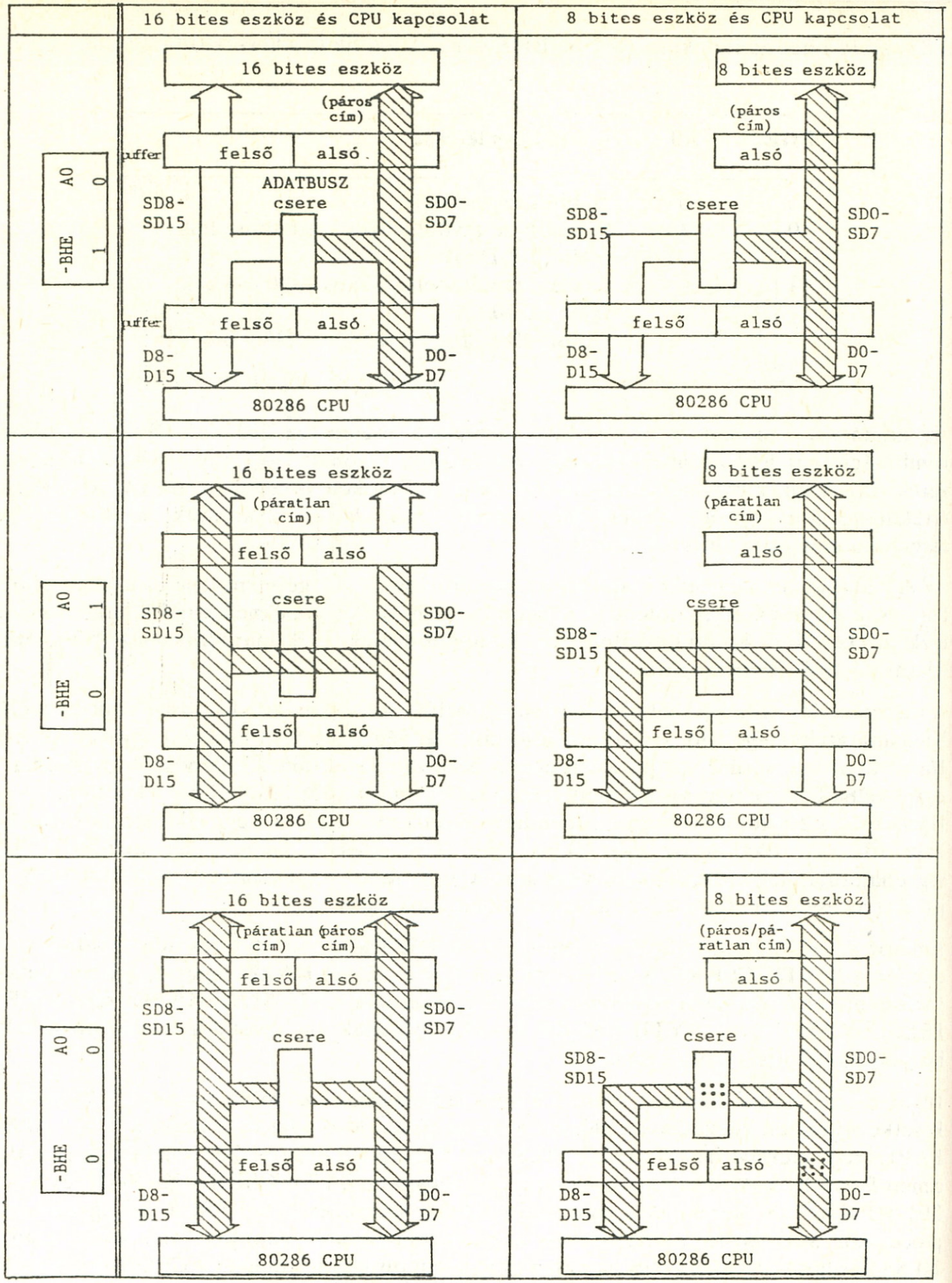
Az adatátviteli utakat 8 és 16 bites eszközök esetén az A0 és -BHE különböző kombinációinál a 2-23 ábra mutatja. Az adatok a vonalkázott csatornákon haladnak keresztül. Ahol a bevonalkázott csatornák a buffereken is áthaladnak, ott a bufferek értelemszerűen nyitott állapotban vannak, míg ha a csatornák a bufferekné megszakadnak ott a bufferek zártak.

Az AT alaplemez adatbusz kialakítása bonyolultabb az XT gépeknél megismerttel, ahol a busz csak 8 bites volt. A dolgot az bonyolítja, hogy az AT gépeknek mind 8, mint 16 bites eszközökkel kell kommunikálnia, és kompatibilisnek kell maradniuk az XT gépek adaptereivel és programjaival.

A buszvezérlés szempontjából a legbonyolultabb eset akkor adódik, ha a processzor 16 bites adatátvitelt igénylő, szó-típusú utasítást hajt végre 8 bites eszközzel együttműködve. Ha az átvitelre kerülő szó páratlan címen helyezkedik el, nincs különösebb probléma: a 8086/80286 CPU-k ugyanis ezt a műveletet két egymásutáni 8 bites ciklusra bontják.
Ha viszont a 2 byte-os szó páros címen helyezkedik el, a CPU ezt egy ciklusban olvassa be, vagy adja ki. Ezalatt az egy ciklus alatt a 8 bites eszköznek két műveletet kell végrehajtania, hogy fogadja a 16 bites adatot, vagy feltöltse az adatbusz alsó és felső byte-ját. Ezt az esetet mutatja a 2-23 ábra jobboldali alsó rajza.

Amikor az átvitel a CPU-tól a 8 bites eszköz felé irányul, a 16 bites adat kiadása után először a 'BYTE CSERE KAPU' zárt és a 'CPU ALSÓ KAPU' nyitott, így az adatszó páros című alsó byte-ja kerül először átvitelre. Ezután a 'CPU ALSÓ KAPU' zárt, a 'CPU FELSŐ KAPU' és a 'BYTE CSERE KAPU' nyitottak, így most a páratlan című felső byte-ot fogadhatja a 8 bites eszköz.

Ha az átvitel iránya fordított, tehát a 8 bites eszköz bocsát ki 16 bites adatot, a folyamat a következőképpen játszódik le: először a páros címen lévő adat-byte kerül a 'CPU ALSÓ KAPU' regiszterébe tárolásra (SN 74LS646). Ezután a 8 bites eszköz a második, páratlan címen fekvő byte-ot teszi az adatbuszra, ami a nyitott 'BYTE CSERE KAPUN' és 'CPU FELSŐ KAPUN' keresztül a processzor felső 8 adatvezetékéhez jut. Ezzel egyidőben a processzor alsó nyolc adatvezetékén a páros címen lévő adat-byte van jelen, amit a 74LS646 regisztere tárolt. Így egyetlen szó-típusú olvasás utasítással a CPU egy ciklusban beolvashatja a 16 adatbitet.



2-23 ábra. Adatátviteli utak.

Visszatérve a 80286-os CPU-hoz, négy olyan kivezetése van, amelyek a 80287-es aritmetikai processzorral való együttműködést szolgálják. Ezek a PEREQ, -PEACK, -BUSY és a -ERROR nevű jelek. (Bővebb ismertetés a 80286, 80287 processzorokról az 1. részben található.) Ugyanezek a kivezetések ugyanilyen nevek alatt az aritmetikai processzoron is megtalálhatók. A 80287 adatátvitelt kezdeményezhet a memóriával úgy is, hogy a PEREQ vonalat logikai 1-be állítva a CPU-tól a memória-kezelés és védett mód kiterjesztését kéri (PEREQ = Processor Extension REQuest). A 80286 a -PEACK vonalon logikai 0-val jelzi, ha a kérést elfogadta. Ettől kezdve a CPU teljes virtuális és 16 Mbyte-os valós címterülete rendelkezésre áll. A 80286/80287 együttes mind 'valós' mind 'védett' módban működtethető. Az aritmetikai processzor a -BUSY vonalon logikai 0 foglaltsági jelet küld a CPU felé amíg valamilyen feladaton dolgozik. Ha eközben valamilyen hibát, vagy kivételt (exception) detektál, a -ERROR vonalat is aktivizálja. A 80287-es aritmetikai processzor három I/O címmel rendelkezik (0F8h, 0FAh és 0FCh), amelyek az utasítások és az adatoperandusok átvitelére szolgálnak. Az I/O címtáblában a 0F8h-0FFh címtartomány a 80287 részére van fenntartva.

A 8088/8087-re és a 80286/80287-re írt programok 'valós' módban áttehetők egyik rendszerről a másikra. A különböző hibakezelési technikák miatt, és a 80287 processzorhoz rendelt hardware interrupt miatt azonban a hibakezelés az AT gépeknél némileg megváltozott.

Míg a 8087 aritmetikai processzor egy interrupt kimenettel is rendelkezik hibajelzési célokra, a 80287 ilyen kimenetet nem tartalmaz. Ehelyett a -ERROR vonal hozhat létre megszakításkérést. XT gépeknél az aritmetikai processzor az NMI interrupton keresztül, AT gépek esetén viszont a 13-as számú hardware interrupton keresztül jelzi a hibát. Hogy az aritmetikai processzor szolgáltatásait is igénybe vevő programok mindkét géptípuson akadálytalanul futhassanak, a 13-as interruptot software úton átirányítják az NMI interrupt kezelő rutin címére.

A bekapcsolási teszt kitölti az interrupt vektortáblában a 13-as interruptot kezelő ROM rutin címét, majd engedélyezi a megszakításkérést. A ROM rutin törli a 80287 foglaltságot jelző belső regiszterét ('busy latch'), majd átadja a vezérlést az NMI rutin címére. Az NMI kezelő először megvizsgálja az aritmetikai processzor státuszregiszterét, hogy azonosítsa a megszakításkérés forrását. Ha az interruptot nem a 80287 okozta, az eredeti NMI rutin kerül végrehajtásra.

A 80287 aritmetikai egység hatféle kivételt (exception) képes detektálni a működés során. Ezek maszkolhatók, vagy a 13-as hardware interrupton keresztül program útján feldolgozhatók. Maszkolás esetén a kivételeket a 80287 'default' belső rutinjai kezelik. Ha a maszk nincs beállítva, a -ERROR hibajel aktív állapotba kerül, ami viszont a 13-as számú megszakításkérést vonja maga után. A '-busy' foglaltsági jel tartósan logikai 0 állapotot vesz fel. A foglaltságot a hex F0 I/O címre küldött írás utasítással lehet törölni, ha a kiadott adat-byte 0 értékű.

Az aritmetikai processzor két üzemmóddal rendelkezik, hasonlóan a 80286 CPU-hoz. Bekapcsoláskor, törléskor, vagy a hex F1 címre kiadott I/O írás utasítás esetén a 80287 'valós címzési módba' kerül. Ez a mód kompatibilis az XT gépek 8087 üzemmódjával. Ha a program végrehajt egy SETPM ESC utasítást, az aritmetikai processzort 'védett,

virtuális módba' kapcsolja. A 'valós' mód a hex F1 címre adott I/O írás utasítással állítható vissza (D0-D7 = 0), amely a 80287-et alaphelyzetbe állítja.

A 80286 CPU a 8088 processzorral megegyező módon két megszakításkérő bemenettel rendelkezik: INTR és NMI. Az INTR maszkolható jelet két kaszkádba kötött 8259A típusú interrupt vezérlő kimenete állítja elő. Az interrupt vezérlők 15 darab megszakításkérő vonalat fogadnak. Az NMI nem-maszkolható interrupt bemenetre érkező jelet egy kapuáramkör szolgáltatja (lásd blokkdiagram NMI LOGIC). A kapu két bemenetére az alapkártya paritáshibát jelző vonala, és az I/O csatorna adapter kártyáinak paritáshibát jelző -I/O CH CK vonala kerül. A kapu működése egy további bemeneten letiltható, így az NMI interruptok az XT gépekhez hasonlóan software úton maszkolhatók.

A rendszer vezérlésére hivatott -S0, -S1, M/-IO, COD/-INTA CPU kimeneteket és a -READY jelet a 82288 jelű buszvezérlő dekódolja. A dekódolt kimeneteken megjelenő -MEMR, -MEMW, -IOR, -IOW, -INTA, ALE, MCE, DEN és DT/R nevű jelek vezérlik a rendszeren belül az adatforgalmat. A 82288 rendszervezérlő funkcióiban megegyezik az XT gépeknél megismert 8288 típusú vezérlővel. A vezérlőjelek a teljes 16 Mbyte-os címtartományban rendelkezésre állnak. A -MEMR és -MEMW mellett előállították ennek a két memóriavezérlő jelnek az első 1 Mbyte-os címtatományra szűkített -SMEMR és -SMEMW nevű változatait is, amelyeket az alapkártyán elhelyezett 0-7FFFh címtartományú memóriák vezérlésére használnak (max. 512 Kbyte).

Az eddigiekben a blokkdiagram alapján bemutatásra került az AT alapkártya "magja": a CPU és közvetlen környezete. A leválasztott rendszer-buszra csatlakozó memóriák, DMA vezérlők és I/O eszközök ismertetésére a következő fejezetekben kerül sor.

2.3.3 RAM és ROM memóriák

Az AT gépek 16 Mbyte-os memória térképét a 2-8 táblázat tartalmazza. Az eredeti IBM AT alaplemezen 512 Kbyte DRAM memória helyezhető el. Az e fölötti memória bővítések az I/O csatorna adapter kártyáin realizálhatók. A memóriabővítő kártyák mind memória kiterjesztésre (extended memory), mind memória bővítésre (expanded memory) használhatók. 'Valós' módban az AT-k maximum 512 Kbyte alap RAM memóriát tudnak kezelni, szemben az XT-k 640 Kbyte-os tartományával. Ha valaki mostanában vásárol egy AT gépet, rendszerint az alaplemezen 1 Mbyte memóriahelyet talál (41256 típusú memóriákból kialakított négy 256 Kbyte-os tömb). Ez nem azt jelenti, hogy az 512 Kbyte-os alap RAM terület nőtt meg, hanem azt, hogy egy 512 Kbyte-os bővítő kártya került az alaplapra. A második 512 Kbyte mind memória kiterjesztésként, mind memória bővítésként konfigurálható.

A ROM terület egy része kettős kijelölésű. Ezen a következő értendő: ugyanaz a ROM memória van hozzárendelve az első és a tizenhatodik (utolsó) 1 Mbyte-os címterület legfelső két 64 Kbyte-os szegmenséhez (0E0000h-0FFFFFh és 0FE0000h-0FFFFFFh). Az alaplemezen a legfelső szegmens (0F0000h-0FFFFFh) elhelyezésére van lehetőség.
Az 100000h-0FDFFFFh közel 15 Mbyte címtartomány az I/O csatornára csatlakoztatható memória bővítő kártyák részére van fenntartva.

Cím [hex]	Megnevezés	Funkció
000000-07FFFF	az alapkártyán 512 Kbyte RAM	Az alapkártyán elhelyezett memória
080000-09FFFF	128 Kbyte RAM	128 Kbyte RAM bővítési lehetőség az I/O csatornán (vagy az alaplemezen)
0A0000-0BFFFF	128 Kbyte video RAM	Grafikus és alfanumerikus képtartalom tárolására fenntartott terület
0C0000-0DFFFF	128 Kbyte I/O ROM	I/O adaptereken elhelyezett ROM-ok számára fenntartva
0E0000-0EFFFF	64 Kbyte fenntartva az alapkártya részére	Kettős címterület kijelölés ugyanerre a memória tartományra. A másik: FE0000-FEFFFF
0F0000-0FFFFF	64 Kbyte ROM az alapkártyán	Kettős címterület kijelölés ugyanerre a memória tartományra. A másik: FF0000-FFFFFF
100000-FDFFFF	≈15 Mbyte memória	Memória bővítő kártyák részére fenntartott terület (I/O csatorna)
FE0000-FEFFFF	64 Kbyte fenntartva az alapkártya részére	Kettős címterület kijelölés ugyanerre a memória tartományra. A másik: 0E0000-0EFFFF
FF0000-FFFFFF	64 Kbyte ROM az alapkártyán	Kettős címterület kijelölés ugyanerre a memória tartományra. A másik: 0F0000-0FFFFF

2-8 táblázat.
Az IBM AT 16 Mbyte-os memória térképe.

2.3.3.1 Memória kiterjesztés és virtuális memória

Az XT gépek 1 Mbyte-os címtartomány limitje a 8088 processzor belső felépítéséből adódott. A 80286-os processzorok kialakítása olyan, hogy "védett" módban 16 Mbyte valós címtartomány elérésére nyílik lehetőség. A CPU által kínált címzési lehetőség azonban csak egy dolog. A másik dolog az, hogy a számítógép mennyit tud használni a rendelkezésre álló címtartományból. Az eredeti IBM AT modellek esetében 3 Mbyte volt a hivatalos limit. Az 1Mbyte feletti részt kiterjesztett memóriának (extended memory) nevezték. A kiterjesztett valós memória mellett a 80286 processzor sajátsága, hogy egy

virtuális memóriát is képes szimulálni, amely jóval nagyobb mint a gépbe épített tényleges memória. A virtuális memória címtartománya egy gigabyte (1024 Mbyte).

A virtuális memória egyfajta bűvészkedésnek tekinthető, amit a mikroprocesszor, a virtuális memóriát vezérlő program és a diszkek közötti gondosan összehangolt működés jellemez.

Amikor a számítógépen egy program futtatásra előkészített állapotba kerül, az operációs rendszer virtuális memória táblát alakít ki. Ez a hely azon memória területeknek és címeknek a modellje, amit a program használni fog. A virtuális memória-címtartományt használó programok fontos jellemzője a szegmentáltság. A program maximum 64 Kbyte területeket magába foglaló szegmensek sokasága, amelyek a teljes virtuális címtartományban elhelyezkedhetnek. A 80286 CPU védett módjában minden szegmensnek van egy memória-táblában rögzített azonosítója, vagy másnéven leírója (segment descriptor). Egy-egy ilyen szegmens leíró 8 byte-ot foglal el. 3 byte a szegmens bázis-címét szolgáltatja, amely ha bekerül a CPU belső szegmens-báziscím regiszterébe, az offszet címmel összegezve a valós fizikai 24 bites címet állítja elő. A szegmens leíró 2 byte-ja a szegmens méretéről tájékoztat (a szegmens utolsó byte-jának az offset értéke), egy további byte pedig a szegmenshez való hozzáférési jogokat definiálja. Ez a byte informál a szegmens típusáról (adat, kód, rendszer), a szegmens fontossági szintjéről, arról hogy a szegmenst az adott időpontban a virtuális memória kezelő program leképezte-e a valós fizikai memóriába, vagy sem, és arról hogy a szegmens éppen kiválasztásra került-e a CPU valamelyik szegmens regiszterén keresztül. (Bővebb információ található a leírtakról az Intel alkatrész katalógusban.) A szegmens leíró 7. és 8. byte-ja későbbi felhasználásra van fenntartva. Egy-egy feladathoz (task) két szegmens leíró táblát rendeltek hozzá: egy globális és egy helyi táblát. Mindkettő maximum 8192 szegmens leírót tartalmazhat.

A virtuális működés elengedhetetlen feltétele a virtuális memória kezelő programmal kibővített operációs rendszer, a szegmens leíró táblák és az azokkal hardware úton együttműködő CPU, továbbá a valós, fizikailag rendelkezésre álló memóriának egy olyan kijelölt része, amelyet a virtuális memória leképzésére használnak.

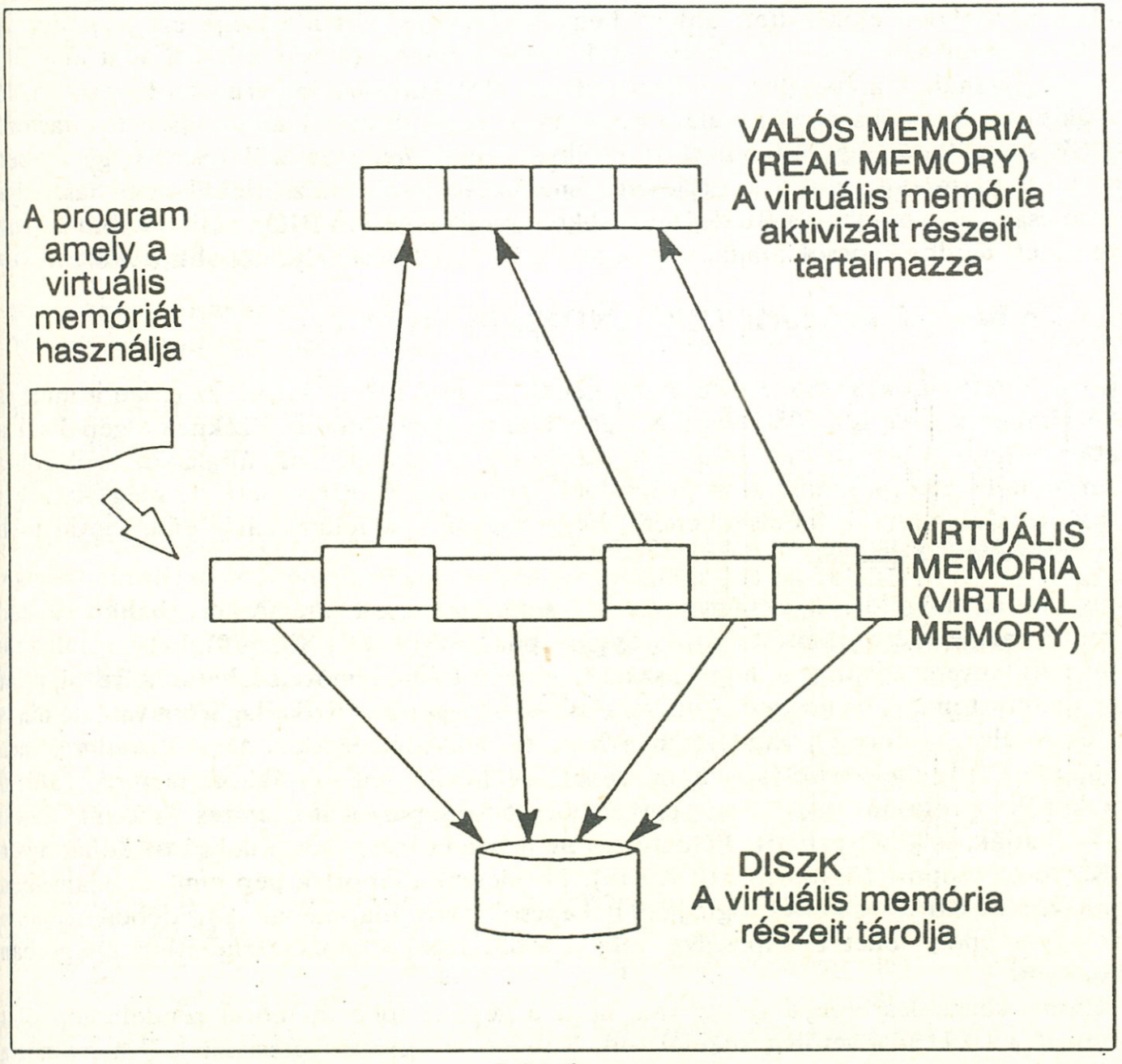
A működés során a virtuális memória kezelő program a 80286 CPU sajátságait felhasználva (memory mapping) arra készteti a valós memória leképzésre kijelölt részét, hogy a saját valós címtartományát elfelejtve, más címeknél, a virtuális címtartományban legyen működőképes.

Mindaddig nincs semmi probléma, amíg a program által elfoglalt virtuális memóriaterület leképezhető a valós memóriába. A helyzet azonban általában nem ez. A virtuális memóriát rendszerint azok a programok használják, melyek címtartománya jóval meghaladja a rendelkezésre álló fizikai memória méreteit. Egy ilyen program indítása úgy történik, hogy a virtuális memória bizonyos részeit, amelyekre az indulásnál szükség van, leképezik a valós memóriába, a többi részét pedig diszken tárolják (2-24 ábra).

Amíg a program a leképzett, aktív memóriában fut minden rendben van. De mi történik, ha a virtuális memória egy olyan részéhez kíván hozzáférni, amelyik nincs benne a valós memóriában? Amikor ez a helyzet előáll, a szegmens leíró tábla jelzi, hogy egy olyan címre történt hivatkozás, amely nem létezik. Minden ilyen esetben a CPU egy kivételt (exception) detektál (page fault), és azonnal a virtuális memória kezelő program veszi át a

vezérlést. Ez a program működését ideiglenesen felfüggeszti, és az előállt probléma megoldásával foglalkozik. A valós memóriába leképzett virtuális memóriarész egy darabját átmenetileg a diszkre viszi át, míg az így felszabaduló helyre a diszkről behozza azt a címtartományt, amelyhez a programnak hozzá kell férnie. Ha a későbbiekben az elmentett memóriarészre újból szükség lesz, a memória kezelő újabb cserét hoz létre a memória és a diszk között.

Programtól függően a virtuális memória kezelése több vagy kevesebb diszk-memória adatcserével járhat. Ha cserék száma túlságosan megnő, a gép működése nagyon lelassul, szinte használhatatlanná válik. Mikrogépeknél különösen körültekintően kell eljárni, mivel a relatív kis memóriakapacitásuk, és lassúságuk korántsem engedi meg ennek az elvnek a nagyobb gépeknél elterjedt széleskörű alkalmazását.



2-24 ábra.
A virtuális memória.

Az AT gépek 16 Mbyte-os címezhetősége, védett módja és a virtuális memória által kínált előnyök kihasználásához az szükséges, hogy létezzen egy olyan operációs rendszer környezet (beleértve ebbe a programokat is), amelyik figyelembe veszi ezeket a sajátosságokat. Mivel a DOS operációs rendszer tervezésekor nem gondoltak a memória kiterjesztésre és a virtuális memória lehetőségére, ezek az előnyök kiaknázatlanok maradtak. Ez az állapot mindaddig megmarad, míg meg nem jelenik az AT gépek és a 80286 CPU által biztosított lehetőségeket maximálisan kihasználó operációs rendszer és felhasználói software.
Ennek ellenére, egy program számára a kiterjesztett memória használata nincsen teljesen kizárva. A szokványos út, hogy a program ilyenkor a ROM-BIOS szolgáltatásait veszi igénybe. Az egyik ilyen szolgáltatás, adatblokkok átvitele a kiterjesztett memória és a konvencionális 1 Mbyte-os alapmemória között. A program átkapcsolhatja a CPU-t védett módba, ha kellően előkészített ahhoz, hogy fennakadás nélkül kezelje ezt a módot. A szokásos eljárás azonban az, hogy elkerülik a védett módot és csak a BIOS adat mozgató rutinjait használják a 16 Mbyte-os címterületen (15h software interrupton keresztül). Jó példák erre a virtuális diszk emuláló programok. A DOS operációs rendszerhez tartozó VDISK szolgáltatás úgy működteti az 1 Mbyte feletti memóriaterületeket, hogy közben sem a védett módot, sem a kiterjesztett memória címzést közvetlenül nem használja, hanem csak valós módban a BIOS rutinnal kommunikál (a 15h BIOS rutin viszont a hívás után védett módba kapcsol, majd kilépéskor a valós kiindulási helyzetet állítja vissza).

2.3.3.2 A bővített memória (EXPANDED MEMORY)

A jövőt feltétlenül a 80286, 80386 alapú gépek jelentik a PC piacon, azonban a múlt és talán részben a jelen is a 8088 alapú XT gépek millióihoz kötődik. Ezeknek a gépeknek a címtartománya 1 Mbyte, de csak 640 Kbyte az a terület, ami általában szóbajöhet programok és adatok számára. A programok fejlesztésénél ezt kellett, de még ma is ezt kell szem előtt tartani, annak ellenére, hogy nagyon sok felhasználás ennél jóval több memória területet igényel.

A probléma megoldását az úgynevezett csoport-kapcsolt memóriák (bank-switched memory) megjelenése jelentette az összes géptípus részére (PC, XT, AT).
Az eljárás lényege abban van, hogy a számítógépben több memóriát lehet elhelyezni, mint amit a címtartomány megenged. A memória többlet a gépben fizikailag jelen van, de nincs hozzárendelve semmilyen címtartományhoz, és hozzáférhetetlen a program számára mindaddig, míg bekapcsolásra nem kerül. A bővítő memóriákhoz tartozó vezérlő áramkörök program útján az egyes memóriacsoportokat tetszés szerint ki-be kapcsolhatják, és áthelyezhetik. Például egy memória bővítő kártyán helyet foglalhat nyolc 64 Kbyte-os csoport (összesen 512 Kbyte). Mindegyik csoport a gép címtartományában ugyanazt a 64 Kbyte-os helyet foglalja el bekapcsolt aktív állapotában. Egyidőben azonban csak egy csoport lehet bekapcsolva, míg a többi passzív, hozzáférhetetlen állapotban tartózkodik.
A csoport-kapcsolás előnye abban van, hogy a géphez több memória rendelhető, mint amennyit a CPU alapvetőleg kezelni tud. Mindössze egyetlen utasítást kell a memória bővítő kártyához küldeni, ha meg kívánják változtatni a csoportok címzését és valamely

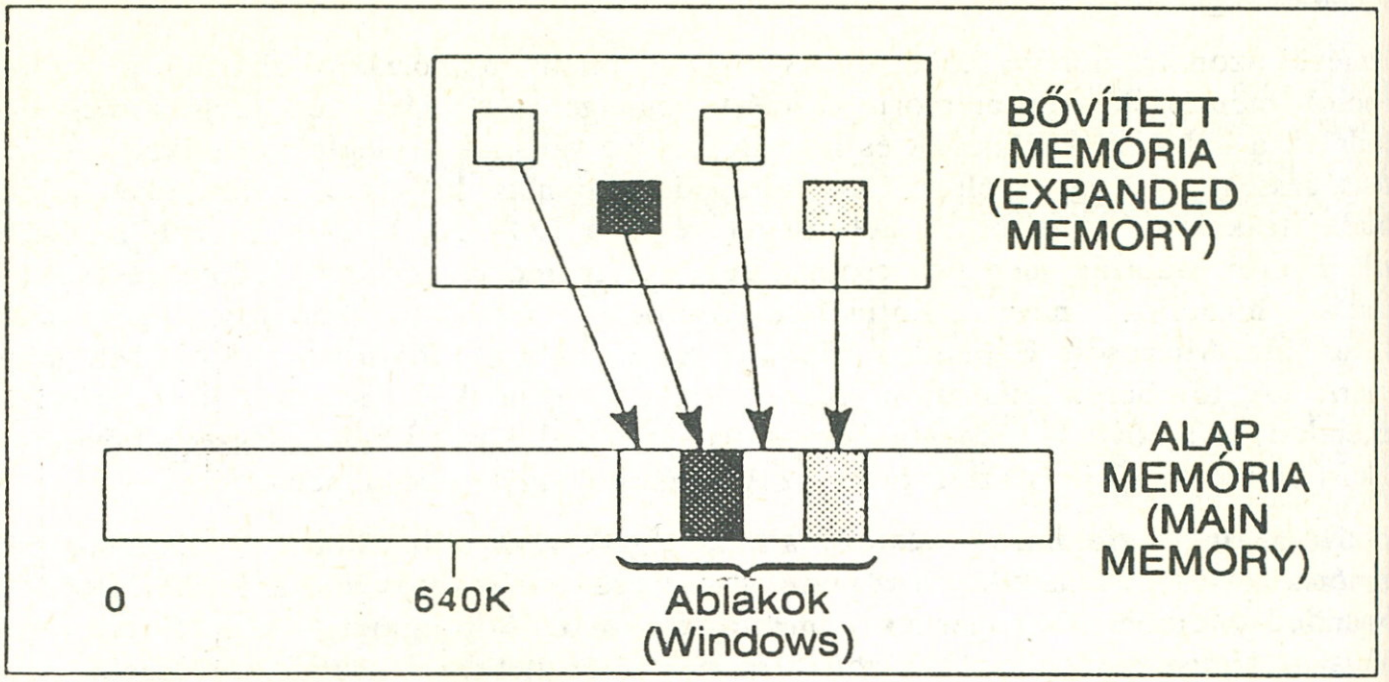
más csoportot akarnak aktív állapotba hozni. Az átkapcsolás késedelem nélkül az utasítás ideje alatt végbemegy.

Felmerül azonban néhány nehézség. A konvencionális memóriáktól eltérően a csoport-kapcsolt memóriák aktív memória-kezelést (management) igényelnek, hogy a megfelelő területek a helyükre kerüljenek és hozzáférhetővé váljanak a megfelelő időkben. Ennek szükségessége, és egy széleskörűen elfogadott ajánlás hiánya a fejlesztőket jóideig elriasztották ettől az úttól. A nehézséget végül a LOTUS, INTEL és MICROSOFT szakemberei oldották meg egy csoport-kapcsolt memória szabvány kidolgozásával. Az ajánlás hivatalos neve: Lotus/Intel/Microsoft bővített memória specifikáció (Lotus/Intel/Microsoft Expanded Memory Specification). Rövidítve LIM EMS néven ismert. (A tévedések elkerülése végett ismét felhívnánk a figyelmet a kiterjesztett (extended) és a bővített (expanded) memória fogalmak közötti különbségre: a bővített az itt leírt memória, a kiterjesztett pedig az AT gépek 1 Mbyte feletti memóriája.)

A bővített memória három részből áll: az egyik rész a hardware (csoport-kapcsolt memória kártya), a másik két rész pedig a software (a bővített memória kezelő: EMM = Expanded Memory Manager és a memóriát használó program). A memória bővítő kártyákat típustól függően különböző memória területekkel árulják, az EMM pedig maximum 8 Mbyte-t képes kezelni. A memóriát 16 Kbyte-os lapokra osztották, ezeket aktivizálhatja a csoport-kapcsoló.

A gép bekapcsolásakor az EMM memória kezelő program működésbe lép és megteremti a bővítés lehetőségét. Legfontosabb feladata, hogy találjon egy olyan nem használt területet a PC címtartományában, ahol egy 64 Kbyte-os munkaterület kialakítható, és ahová a csoport-kapcsolt memóriák leképezhetők. Az XT és AT memória térképből kiderül, hogy a D és E blokkok lehetnek jó választások, de az EMM a C blokkot is használhatja. A pontos helynek ezen belül nincs jelentősége, csak arra kell vigyázni nehogy olyan terület legyen, amit más eszközök is használnak. Lehet olyan megkötés is, hogy a 64 K munkaterületet nem szabad a blokkok határainál elhelyezni. Tehát például a 0C400h szegmens címnél kezdődhet egy ilyen terület és átnyúlhat a D blokk első 16 Kbyte-jába, de nem kezdődhet a 0C000h szegmens címnél. Az EMM miután kijelölte a 64 K munkaterület helyét (page frame), négy 16 K nagyságú ablakot alakít ki. Most már minden előállt a bővített memóriát igénylő felhasználói program indításához. A felhasználói programnak tudnia kell, hogy hogyan veheti igénybe az EMM szolgáltatásait a 16 K-s ablakok adatcseréinél. A program közli az EMM-el, hogy hány ablakot akar használni, és kéri hogy rendeljen egy-egy lapot az ablakokhoz, majd aktivizálja őket. Ha a program különböző 16 K adattömböket igényel, az EMM ezeket egymás mellé, az ablakokba helyezi (2-25 ábra).

A bővített memória használata néhány nyilvánvaló kötöttséggel jár együtt. Az egyik az, hogy a programhoz tartozó adatok tárolására alkalmas, de programrészek tárolására nem. A konvencionális 0-640 K tartományban a DOS operációs rendszer elég helyet talál arra, hogy nagy programokat is elhelyezzen, de ha ezeket a programokat végrehajtják, könnyebbséget jelent ha a bővített memóriában helyezhető el az adatoknak az a része, ami már nem fért ebbe a tartományba.



2-25 ábra.
A bővített memória.

A másik kötöttséget az jelenti, hogy a programnak tudnia kell, hogy hogyan veheti igénybe az EMM szolgáltatásait, és hogyan tudja használni a 16 Kbyte-os lapokra osztott adattartományt. Végeredményben a bővített memória többszörösen megnöveli a gép teljesítőképességét olyan esetekben, amikor nagy mennyiségű adat feldolgozása válik szükségessé.

A memória bővítést figyelembe vevő néhány ismertebb programcsomag a Lotus 1-2-3 Release 2, a Symphony 1.1, Framework II.

A LIM EMS-t alapul véve a kaliforniai AST Kutató Intézet is kidolgozott egy memória bővítő ajánlást EEMS néven (Enhenced Expanded Memory Specification). Az EEMS flexibilisebb lapkezelést biztosít mint a LIM EMS, ami javítja a software jellemzőket, és lehetővé teszi több feladat egyidejű elvégzését (multitasking).

A valós memóriában képzett ablakok (amelyek a 640 Kbyte alatti tartományba is eshetnek) és a memória bővítő kártyák 16 Kbyte-os lapjai közötti kapcsolatot 64 'mapping' regiszter, a leképzést vezérlő regiszter és a lapregiszterek határozzák meg. A memória kezelő program ezenkívül folyamat azonosítókat (Process IDs) rendelhet a bővítőlapokhoz. Minden azonosító hozzákapcsolható valamely programhoz. Ezáltal egy-egy program a memóriabővítő kártyán olyan lapokkal rendelkezhet, amelyek csak a számára hozzáférhetőek. Ez a módszer segíti a 'multitask'-os környezetet kialakító rendszer programok működését.

Az EEMS rendelkezik mindazokkal a lehetőségekkel, amit a LIM EMS kínál, és megteremti a feltételeit több program egyidejű működtetésének. Az AST DESQview programja például a DOS operációs rendszer kibővítésével, AST gyártmányú

bővítőkártyákat feltételezve (RAMpage!, RAMpage AT, vagy SixPakPremium) 9 programot futtathat egyidőben a bővített memóriában, összesen 5,5 Mbyte terjedelemben. A konkurrens programok a LIM EMS-re épülő programok is lehetnek. Ha a bővített memória megtelt, és egy új programrész betöltése válik szükségessé, a DESQview automatikusan elment a diszken egy nem használt programot, és a felszabadított területre betölti az újat, így virtuális memóriakezelésre is mód nyílik.
Az idézett példa, a DESQview program jól mutatja azt a törekvést, hogy a bővített memória használatával, a 80286 CPU valós módjában, jóval szerényebb körülmények között ugyan, de mégiscsak megvalósítsák a védett mód által kínált 'multitask'-os lehetőségeket.

2.3.3.3 ROM memóriák

Az eredeti IBM AT alaplemezen vagy két 32Kx8 bites ROM/EPROM, vagy négy 16Kx8 bites ROM/EPROM modul helyezhető el rendszer ROM célokra (64 Kbyte). A páros és páratlan címeken található memóriahelyek két szeparált memóriacsoportot képeznek (LOW BANK = páros címek, HIGH BANK = páratlan címek). Az alaplemez rendszer-ROM címtartománya az első és utolsó 1 Mbyte-os terület legfelső blokkja (0F0000h-0FFFFFh és 0FF0000h-0FFFFFFh). A ROM-ok paritását nem ellenőrzik. A ROM-ok hozzáférési ideje maximum 260 (190) nsec, a ciklusidejük pedig 480 (360) nsec (a zárójeles értékek a 8 MHz-es órajellel működő típusokra vonatkoznak). Ha két 32Kx8 bites ROM chip-et használnak, két ROM foglalat üresen marad. Ide járulékos ROM/EPROM modulokat lehet elhelyezni a 0E0000h-0EFFFFh vagy a 0FE0000h-0FEFFFFh címtartományban.

Az alaplemezen elhelyezett járulékos ROM memóriát a gép bekapcsolásakor induló teszt program (POST) csak akkor fogja hibátlannak minősíteni, ha a ROM kitöltésénél figyelembe vették a következőket:

0. byte:	Hex 55	(0E0000H)
1. byte:	Hex AA	(0E0001H)
2. byte:	nem használt	(0E0002H)
3. byte:	program belépési pont	(0E0003H)
.	.	
.	.	
.	.	
65536. byte:	checksum	(0EFFFFH)

Helyes checksum érték és hibátlan kiolvasás esetén a járulékos ROM modul valamennyi byte-ját modulo hex 100 szerint összegezve, az utolsó byte (checksum) után 0-t kell kapni. Ha a járulékos ROM nem az előírt formájú, vagy a checksum hibás, a teszt futásakor hibaüzenetet kapunk. Hibátlan ROM esetén a BIOS tesztprogramja egy 'távoli hívással' (CALL FAR) átadja a vezérlést a járulékos ROM modul 3. byte-jának (belépési pont). A visszatérést egy RET FAR utasítás fogja végrehajtani.

Az alaplemezen elhelyezhető járulékos ROM memórián kívül az adapter kártyák is tartalmazhatnak ROM modulokat. Példa lehet erre az AT gépek bizonyos típusú Winchester diszk vezérlője, vagy az EGA grafikus adapter kártya.

A bekapcsolási teszt alatt a ROM-BIOS kitölti a 'default' interrupt vektor címeket. Ezután végignézi hogy vannak-e járulékos ROM chip-eket tartalmazó adapter kártyák. Ha ilyet talál és a ROM formátum megfelelő, átadja a vezérlést a kártya részére. Az adapter kártya ekkor kitöltheti a számára szükséges interrupt vektorokat, RAM munkaterületet inicializálhat, egyszóval hozzákapcsolhatja magát a renszerhez. Ezután a vezérlés visszakerül az alapkártya ROM-BIOS-ához.

A 0C8000h címtől kezdve a 0E0000h címig bezárólag a BIOS 2 Kbyte-os blokkonként vizsgálja meg a memóriaterületet járulékos ROM adaptert keresve. A ROM modul difiníciója nagyon hasonlít az alaplemezen elhelyezett járulékos ROM memóriánál leírtakkal:

0. byte:	Hex 55
1. byte:	Hex AA
2. byte:	a járulékos ROM modulban található 512 byte-os blokkok száma (a modul hossza)
3. byte:	program belépési pont
	.
	.
	.
A modul utolsó byte-ja :	checksum

A BIOS a járulékos ROM adaptert akkor fogja érvényesnek tekinteni, ha mind az itt leírt formátumot, mind a checksum-ra végzett ellenőrzést hibátlannak találja.

A járulékos ROM adapterekre vonatkozó megállapítások az XT és AT típusú gépekre egyaránt igazak.

Mint ismert, a ROM-BIOS címterülete az F jelű 64Kbyte-os blokk felső részére esik. A gép bekapcsolásakor, vagy a RESET gomb megnyomásakor a vezérlés az 0FFFF0h címre kerül. Az itt elhelyezett 5 byte-os JMP FAR utasítás adja át a vezérlést a ROM-BIOS kezdő címére. A BIOS által elfoglalt ROM terület géptípustól és gyártóktól függően változó. A ROM-BIOS végének jelzésére különböző módszerek használatosak. Az IBM például dátumot és egy azonosító byte-ot (ID) helyez a BIOS végére. Az azonosító byte értékei: ID = FF a PC, ID = FE az XT, ID = FD a PCjr és ID = FC az AT típusú IBM gépek esetén. Az azonosító hasznos útmutató lehet a programok számára, ha az egyes típusoknál a program "adjusztálására" van szükség.

2.3.3.4 RAM memóriák

Az eredeti IBM AT alapkártyákat (type 1 system board) 256 Kbyte, vagy 512 Kbyte DRAM memóriával szállították, 128Kx1 bites chip-ekből felépítve. A 256Kx1 bites elemek elterjedésével ez a helyzet megváltozott és egy módosított alaplemez jelent meg 256Kx1

bites memóriákkal (type 2 system board). A memória hozzáférési idő maximum 150 nsec és a ciklusidő 275 nsec.

A frissítés 1 memória ciklust igényel minden 15 mikroszekundum alatt. A frissítést az XT gépektől eltérő módon nem a DMA0 csatornán keresztül valósítják meg. A memória vezérlő önálló frissítő áramkörrel rendelkezik, és csak a frissités időpontját kell kijelölni egy számláló/időzítő kimenet felhasználásával. Az alaplemezen található 8254-2 típusú programozható számláló/időzítő CLK OUT1 kimenete jelzi a frissítési időpontokat. A frissítést a memória chip-ek 256 sorcímén kell végrehajtani max. 4 msec. alatt. A frissítési címeket egy SN74LS590 típusú számláló állítja elő (8 címvonal). A számláló '3-state' kimenetei csak a frissítési ciklusok alatt engedélyezettek, egyébként nagyimpedanciás állapotban találhatók. A frissítés befejezésekor a számláló tartalma 1-el növekszik. Bekapcsolás után a DRAM memória csak 8 frissítési ciklus (vagy olvasás/írás) után üzemképes.

A páros és páratlan címeken található memóriahelyek, akárcsak a ROM memóriáknál, két szeparált 8 bites csoportot képeznek (LOW BANK = páros címek, HIGH BANK = páratlan címek). Mind a két csoporthoz tartozik egy-egy paritás-képző és paritás-ellenőrző áramkör, amelyek azonos módon működnek az XT gépeknél megismerttel.

A memória vezérlő áramkörök kimenetei szolgáltatják a DRAM csoportokat meghajtó -RAS0,1,2,3 -CAS0L,H -CAS1L,H és -WE jeleket. A vezérlő áramkörök címdekódolást és frissítést is végeznek, és megfelelő ütemben kapcsolják a címmultiplexert.

Az AT alaplemezen van egy jumper átkapcsolási lehetőség, amellyel vagy 256K vagy 512K DRAM terület választható. Ez a kapcsoló nem a BIOS-t informálja, mint az XT gépek SW1 DIP kapcsolója, hanem 256K állásában a memória vezérlő címdekóderén keresztül a memóriát hardware úton a 0-3FFFh címtartományra szűkíti. Mire jó ez, miért van erre szükség? Vannak olyan programok (pl. DESKview), amelyek a bővített memóriát akkor tudják igazán jól használni, ha az alapkártya konvencionális RAM területét a lehető legjobban lecsökkentik. A jumper segítségével a terület úgy csökkenthető, hogy nincs szükség a feleslegessé vált DRAM chip-ek eltávolítására.

A RAM és ROM memóriák MD0-MD15 16 bites adatbuszát két 74LS245 buffer elem csatolja az SD0-SD15 rendszer-adatbuszhoz. A 74LS245 áramkörök az átviteli módot (páros/páratlan 8 bites, 16 bites) figyelembe véve kapcsolják össze a két busz alsó és felső 8 vonalát.

Az AT gépek 512K alap-RAM címtérképét a 2-9 táblázat mutatja. A 0-3FFh interrupt címterület 0-1DFh részét a BIOS, DOS és a rendszer hardware interrupt címei foglalják el. A BIOS rutinok RAM munkaterülete 400h-tól 4FFh-ig tart. 400h-407h helyeken az RS232C kommunikációs adapterek, a 408h-40Fh helyeken pedig a nyomtató interface-k báziscímei találhatók. A bekapcsolás utáni inicializálás, és az operációs rendszer 'behúzása' alatt (bootstrap) a 300h-3FFh címtartomány stack területként használatos. A fenntartott memóriahelyeket a táblázat összesíti.

CÍM(hex)	HASZNÁLÓ	FUNKCIÓ
0-1DF	BIOS,DOS, HARDWARE	Interrupt vektor címek
1E0-3FF	BASIC,USER	Interrupt vektor címek
400-4A1	ROM BIOS	ROM BIOS adat terület
4A2-4EF	Fenntartva	
4F0-4FF		Belső alkalmazásokra fenntartott terület
500-5FF		DOS és BASIC számára fenntartott terület
500	DOS	Képernyő tartalom nyomtatás státusz byte 00 = Képernyő tart. nyomtatás nincs aktivizálva, vagy a nyomtatás sikeresen befejeződött. 01 = Képernyő tart.nyomtatás van folyamatban. FF = A nyomtatás alatt valamilyen hiba állt elő.
504	DOS	Floppy diszk státusz byte
510-511	BASIC	A BASIC szegmens címe
512-515	BASIC	Az órajel interrupt szegmens és offset címe
51A-51D	BASIC	Diszk hiba interrupt szegmens és offset címe
600-7FFFF	USER	Programok számára kijelölt RAM terület

2-9 táblázat.
Az IBM AT RAM memória térképe (0-512 Kbyte).

2.3.4 Az AT interrupt rendszere, hardware interruptok, interrupt vektorok listája

Az AT gépek 16 szintű interrupt rendszerrel rendelkeznek. A 8259A interrupt vezérlők számát az XT-hez képest kettőre növelték. A két kaszkádba kötött vezérlőnél a második kimenete az első IRQ2 bemenetére került. Az interrupt vektorokat úgy helyezték el a táblában, hogy az XT gépekkel a kompatibilitás biztosítva legyen. A hardware interruptok listáját a 2-10 táblázat tartalmazza.

VEKTOR	SZÁM		FELHASZNÁLÁS
	NMI		Alaplemez RAM paritáshiba, I/O csatorna paritáshiba (I/O Channel Check)
	CTRL1	CTRL2	
08h	IRQ0		Rendszer óra ('real time' óraütések)
09h	IRQ1		Billentyűzet (kimeneti puffer feltöltve)
0Ah	IRQ2		Interrupt a CTRL2 vezérlőtől (második 8259A)
70h		IRQ8	'Real time' óra
71h		IRQ9	Software úton átirányítva a 0Ah vektorhoz (IRQ2)
72h		IRQ10	Fenntartva
73h		IRQ11	Fenntartva
74h		IRQ12	Fenntartva
75h		IRQ13	80287 aritmetikai processzor hibakezelés
76h		IRQ14	Merev lemezes diszk vezérlő
77h		IRQ15	Fenntartva
0Bh	IRQ3		Második RS232C interface (COM 2) BSC BSC (alternatív) 'Cluster' (elsődleges) PC Network* hálózat adapter PC Network (alternatív)* hálózat adapter SDLC szinkron kommunikációs adapter
0Ch	IRQ4		Első RS232C interface (COM 1) BSC BSC (alternatív) SDLC szinkron kommunikációs adapter
0Dh	IRQ5		Második parallel nyomtató interface
0Eh	IRQ6		Floppy diszk vezérlő

2-10 táblázat (I.).
AT hardware interrupt lista.

VEKTOR SZÁM	FELHASZNÁLÁS
0Fh IRQ7	Első parallel nyomtató interface Adatgyűjtő és vezérlő adapter*** GPIB** IEEE 488 busz interface 'Cluster' (másodlagos)
* A PC Network jumper átkötéssel választható ** A GPIB adapter a 2-7 interruptokhoz kapcsolható *** Az adatgyűjtő adapter a 3-7 interruptokhoz kapcsolható A 'default' érték: 7	

2-10 táblázat (II.).
AT hardware interrupt lista.

A táblázat a megszakítás kérelmeket prioritási sorrendben ábrázolja. Legmagasabb prioritással az NMI interrupt bemenet, a legalacsonyabbal pedig az IRQ7 bemenet rendelkezik. A második 8259A interrupt vezérlő (CTRL2) 8 bemenete prioritási sorrendben megelőzi az első vezérlő IRQ3-IRQ7 bemeneteit. A kaszkádba kapcsolás miatt az első interrupt vezérlő IRQ2 bemenetét nem lehet kihasználni. Hogy ezt a hiányt pótolják (XT gépeknél ez a vonal rendelkezésre áll), az IRQ9 vonalat használják az IRQ2 pótlására. Az IRQ9-hez tartozó interrupt-vektor a 71h. Az IRQ9 kérést kezelő interrupt rutin egyszerűen átadja a vezérlést az IRQ2-t kiszolgáló rutinnak. Hasonló eljárást alkalmaztak az aritmetikai processzor interrupt fogadásánál is. Az XT gépek 8087 megszakítás kérését az NMI rutin dolgozza fel. Hogy ez az analógia megmaradjon, az IRQ13-hoz tartozó interrupt rutin (80287 hibakezelés) átadja a vezérlést az NMI vektor címére. Valamennyi hardware interrupt kérés maszkolható, beleértve az NMI-t is.

A 2-11 táblázat az interrupt vektorokat ábrázolja AT gépek esetén. Összevetve ezt az XT interrupt vektor táblájával (2-6 táblázat), látható, hogy az eltérés minimális. A 0Dh számú interrupt vektor az AT gépeknél a második nyomtatóhoz, XT gépeknél a Winchester diszk vezérlőhöz van hozzárendelve. Az AT-k második 8259A vezérlőjéhez a 70h-77h számú vektorok tartoznak, az XT-k ezt a tartományt nem használják.

Memória cím(hex)	Vektor szám	Megnevezés	BIOS belépési pont
0-3	0	Nullával való osztás	D11
4-7	1	Egyes programlépés	D11

Memória cím(hex)	Vektor szám	Megnevezés	BIOS belépési pont
8-B	2	Nem maszkolható interrupt(NMI)	NMI INT
C-F	3	'Break point' utasítás	D11
10-13	4	Túlcsordulás	D11
14-17	5	Képernyőtartalom kinyomtatása	PRINT SCREEN
18-1B	6	Fenntartva	D11
1C-1F	7	Fenntartva	D11
20-23	8	Rendszer óra interrupt (18,2/sec)	TIMER INT
24-27	9	Billentyűzet	KB INT
28-2B	A	Fenntartva	D11
2C-2F	B	Második RS232C interface	D11
30-33	C	Első RS232C interface	D11
34-37	D	Második nyomtató interface	D11
38-3B	E	Floppy.diszk vezérlő	DISK INT
3C-3F	F	Első nyomtató interface	D11
40-43	10	Video I/O hívás	VIDEO IO
44-47	11	Kiépítettséget ellenőrző hívás	EQUIPMENT
48-4B	12	Memória méretét meghatározó hívás	MEMORY SIZE DETERMINE
4C-4F	13	5 1/4" diszk-vezérlő I/O hívás	DISKETTE IO
50-53	14	RS232C interface I/O hívás	RS232 IO
54-57	15	Kazetta vezérlő I/O hívás/ Rendszer-kiterjesztések	CASSETTE IO/ SYS. EXTENSIONS
58-5B	16	Billentyűzet I/O hívás	KEYBOARD IO
5C-5F	17	Nyomtató I/O hívás	PRINTER IO
60-63	18	Rezidens ROM BASIC belépés	F600:0000
64-67	19	Operációs rendszer betöltése	BOOT STRAP
68-6B	1A	'Real time' óra beállítás, kiolvasás	TIME OF DAY
6C-6F	1B	Ctrl & Break billentyű rutin	DUMMY RETURN
70-73	1C	Rendszer óra rutin	DUMMY RETURN
74-77	1D	Video inicializáló tábla mutatója	VIDEO PARMS
78-7B	1E	Diszkett paraméter tábla mutatója	DISK BASE
7C-7F	1F	Grafikus karakter generátor tábla mutatója (80-FF)	0

2-11 táblázat (I.)
Az IBM AT interrupt-vektor táblája. A BIOS belépési pontok az IBM AT BIOS listában találhatók. Irodalom: IBM AT Technical Reference.

Memória cím(hex)	Vektor szám	Megnevezés
80-83	20	DOS program lezárás
84-87	21	DOS funkciók hívása
88-8B	22	DOS program lezáró rutin címe
8C-8F	23	DOS Ctrl & Break (Ctrl&C) rutin címe
90-93	24	DOS fatális hibát kezelő rutin címe
94-97	25	DOS diszk olvasás
98-9B	26	DOS diszk írás
9C-9F	27	DOS program lezárás memória kiürítés nélkül (a program a memóriában marad).
A0-FF	28-3F	Fenntartva a DOS részére
100-17F	40-5F	Fenntartva
180-19F	60-67	Fenntartva a felhasználó software interruptok részére
1A0-1BF	68-6F	Nem használt
1C0-1C3	70	IRQ8 'Realtime' óra interrupt (BIOS belépési pont: RTC-INT)
1C4-1C7	71	IRQ9 (BIOS: RE-DIRECT)
1C8-1CB	72	IRQ10 (BIOS: D11)
1CC-1CF73		IRQ11 (BIOS: D11)
1D0-1D3	74	IRQ12 (BIOS: D11)
1D4-1D7	75	IRQ13 a BIOS átirányítja az NMI interrupthoz (BIOS: INT-287)
1D8-1DB	76	IRQ14 (BIOS: D11)
1DC-1DF	77	IRQ15 (BIOS: D11)
1E0-1FF	78-7F	Nem használt
200-217	80-85	Fenntartva a BASIC részére
218-3C3	86-F0	Fenntartva a BASIC interpreter részére futási időben
3C4-3FF	F1-FF	Nem használt

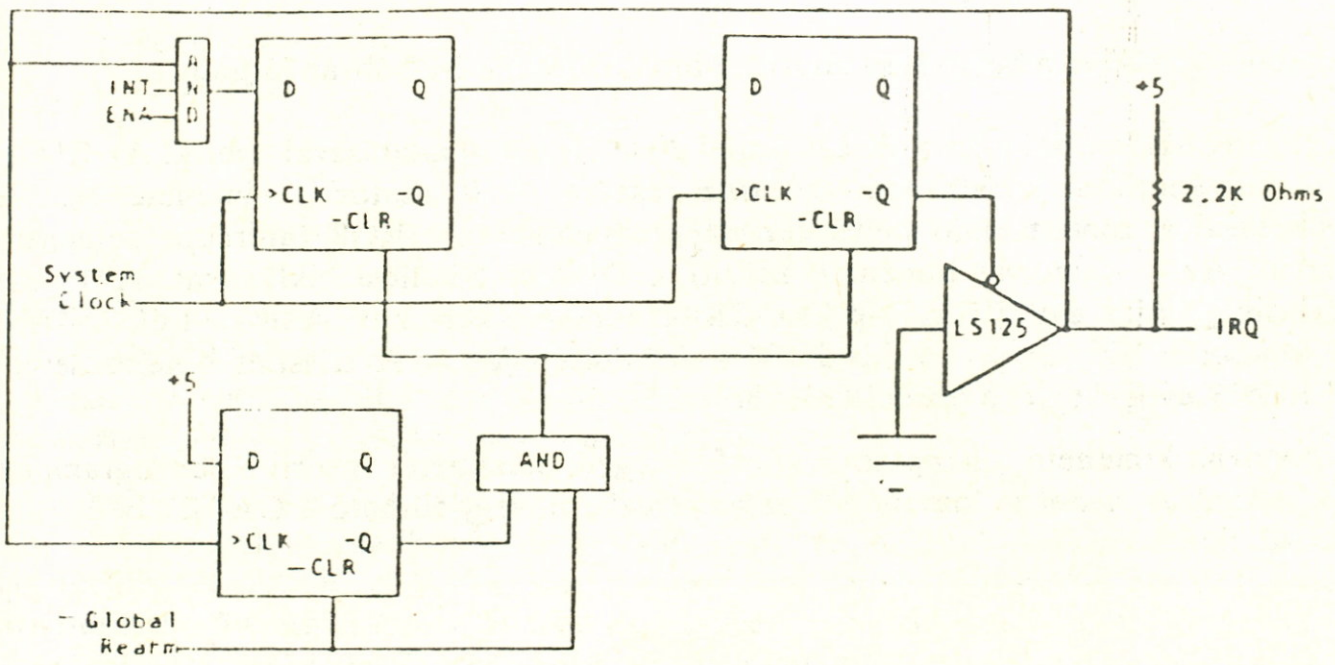
2-11 táblázat (II.).
Az IBM AT interrupt vektor táblája.

Az interruptok használata külső célokra

A hardware interrupt listát áttekintve láthatjuk, hogy szinte minden interrupt bemenet valamilyen rendszer-funkcióhoz kötött. A felhasználói célokra igénybe vehető bemenetek száma erősen korlátozott. A probléma megoldására az IBM ajánl egy eljárást, amely lehetővé teszi, hogy ugyanahhoz az interrupt vonalhoz több adaptert lehessen csatlakoztatni.

A szokásos módszer az, hogy az adapter kártya a megszakításkérő IRQ vonalat TTL0 szinten tartja, és interrupt esetén TTL1 szintre helyezi. Osztott-interrupt kezelés esetén a helyzet ezzel ellentétes. Az IRQ vonalra '3-state' meghajtókon keresztül csatlakoznak a megszakítást kérő modulok (2-26 ábra). Nyugalmi állapotban a meghajtók nagyimpedanciás kimenetekkel rendelkeznek, az IRQ vonalat pedig egy felhúzó ellenállás +5V-os szinten tartja. Bármelyik modul megszakítást kérhet az IRQ vonalra helyezett TTL0 impulzussal. Az impulzus első 1 -->0 éle a 8259A interrupt vezérlőt kezeli. A visszafutó 0 -- > 1 él hozza létre az interruptot. A pulzus időtartama 125 -- 1000 nsec.

A közös IRQ vonalra csatlakozó adapterek mindegyike rendelkezik egy interrupt státusz bittel (INT) és egy interrupt engedélyező bittel (ENA), amelyek software úton vezérelhetők és figyelemmel kísérhetők. Mindegyik adapternek kötelessége figyelni a közös IRQ vonalat. Ha valamelyik közülük kiad egy megszakítás kérést, és a vonalat TTL0 szintre húzza, a többi mindaddig nem kezdeményezhet interruptot, míg a folyamatban lévő interrupt kezelő rutin egy 'global rearm' utasítással ezt nem engedélyezi. Ha egy adapter INT státusz bitje éppen logikai 1 állapotban van amikor a 'global rearm' parancs megérkezik, az adapternek meg kell kísérelnie újból kiadni az interruptot. Ez biztosítja azt, hogy ne vesszen el megszakítás kérés, ha egyidejűleg két adapter szeretné használni az IRQ vonalat.



2-26 ábra.
Osztott-interrupt kezelő áramkör. Logikai diagram.

2.3.5 A 8254-2 típusú számláló/időzítő és a hangszóró meghajtó áramkörök. Programozható parallel interface (PPI).

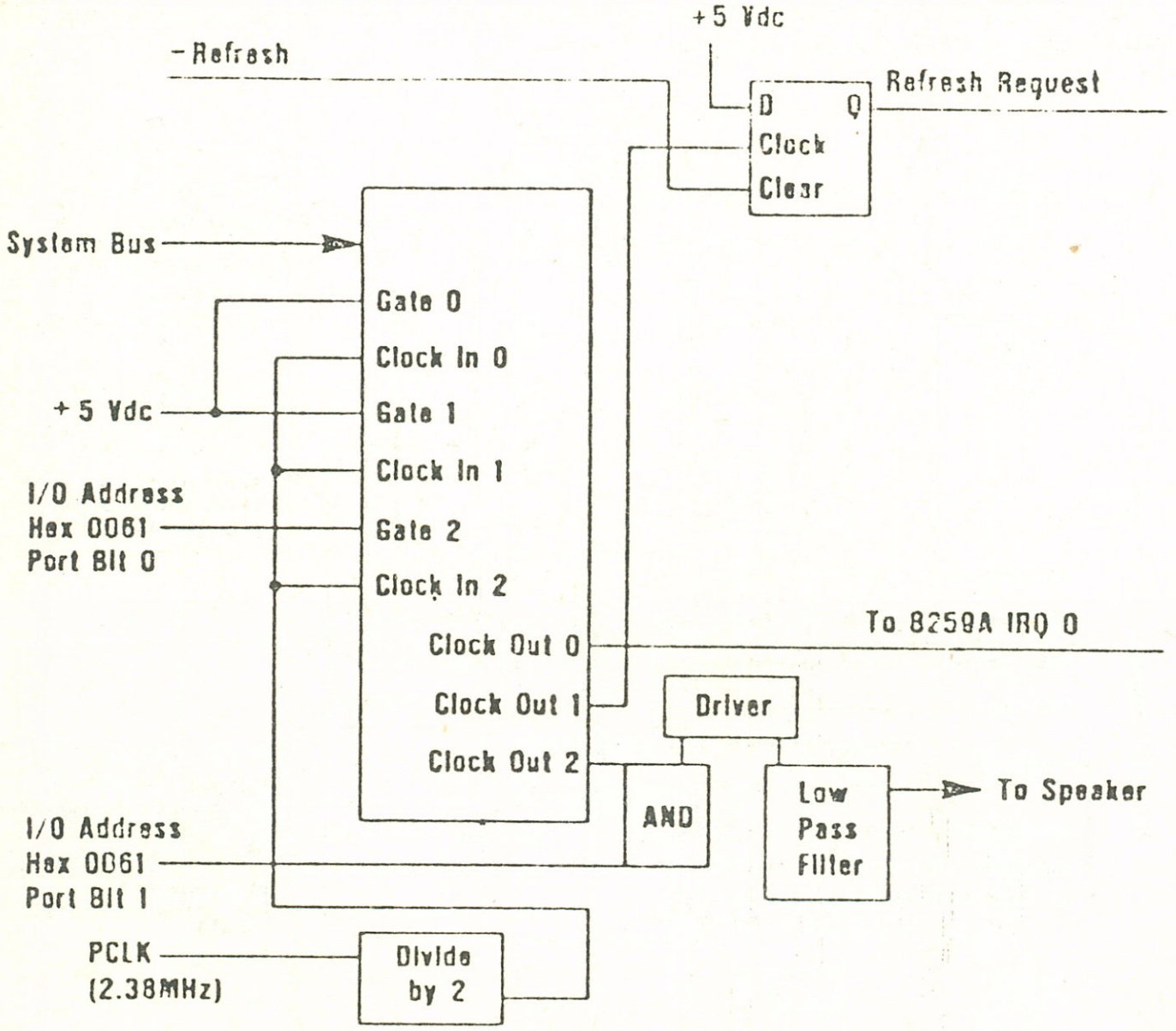
A háromcsatornás 16 bites számláló/időzítő ugyanazokat a feladatokat látja el, mint az XT 8253 típusú időzítője. Négy I/O 'port'-tal rendelkezik. Három ezek közül a számláló/időzítő csatornákhoz tartozik, a negyedik pedig a vezérlő regiszter programozására szolgál. A három csatorna felosztása:

0. csatorna:	**Rendszer óra (system timer)**
GATE0	+5V
CLK IN0	1,193182 MHz OSC
CLK OUT0	8259A IRQ0
1. csatorna:	**RAM frissítést időzítő generátor**
GATE1	+5V
CLK IN1	1,193182 MHz OSC
CLK OUT1	frissítési ciklus kérés 15 μsec-ként
2. csatorna:	**Hangfrekvenciás jelek a hangszóró részére**
GATE2	a 61h I/O című programozható parallel interface 0. bitje vezérli
CLK IN2	1,193182 MHz OSC
CLK OUT2	a hangszóró meghajtó jele

A számláló/időzítő és a hangszóró interface blokkvázlata a 2-27 ábrán látható.

Az 1,193182 MHz OSC jelet a PCLK órajel 2-vel történő leosztásával nyerik. Az OSC jel szolgál bemenetként mindhárom csatorna részére. A 0. csatorna kimenete 18,2 Hz frekvenciával a rendszer-idő méréshez ad ütemjeleket az IRQ0 interrupt bemeneten keresztül. Az 1. csatorna kimenete időzíti a DRAM frissítési ciklusokat. A 15 usec periódusidejű jelek egy D flip-flop CLOCK bemenetére kerülnek. A 0 -- > 1 él a bistabilt log.1 állapotba billenti, és ez idézi elő a frissítési ciklust. A frissítés befejezésével a -REFRESH nyugtázó jel a bistabilt alaphelyzetbe állítja.

A 2. csatorna kimenete a hangszóró interface egyik bemenetét vezérli a beprogramozott frekvenciával. A csatorna kimenőjele engedélyezhető, vagy tiltható a GATE2 bemeneten keresztül.

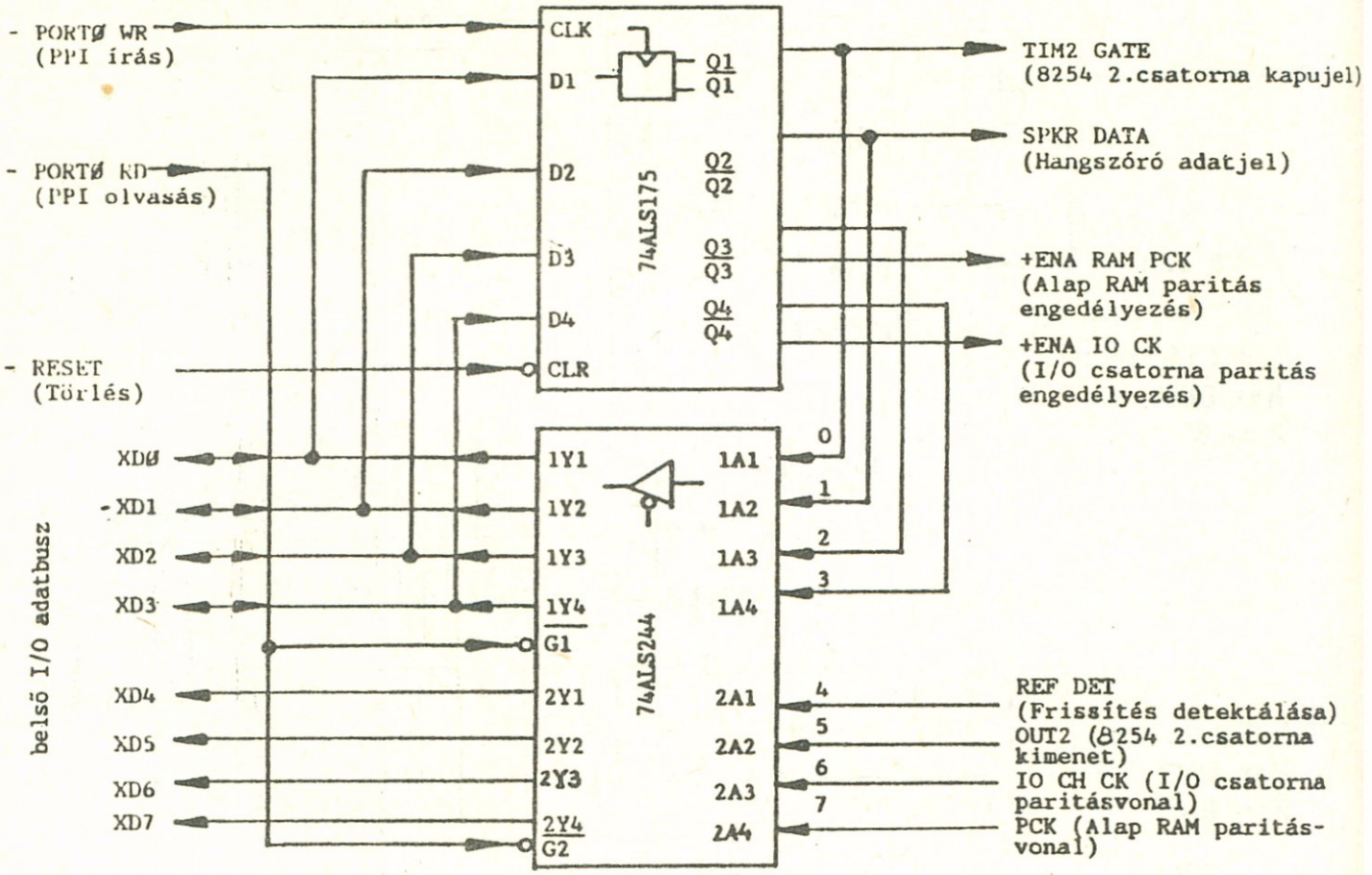


2-27 ábra.
A számláló/időzítő és a hangszóró interface blokkvázlata.

A GATE2 bemenetet a programozható parallel interface (PPI) 0. bitje, a hangszóró vezérlő második bemenetét pedig a PPI 1. bitje vezérli.
A programozható parallel interface-t a 2-28 ábrán láthatjuk: egy 74ALS175 típusú regiszterből és egy 74ALS244 típusú bufferből áll.

A PPI-nek négy kimenő vonala, és nyolc bemenő vonala van. A bemenő vonalak közül négy a kimenetek visszaolvasására szolgál. A 0. és 1. bitek felhasználását az előbb ismertettük. A 2. bit a RAM paritásellenőrzést, a 3. bit pedig az I/O csatorna paritásellenőrző vonalát (I/O CH CK) engedélyezi vagy tiltja. A 74ALS175 regiszter 0-3 kimenetei a 74ALS244 buffereken keresztül az XD0-XD3 adatvonalakra kapcsolhatók. A parallel interface 4,5,6,7 bemenő vonalai sorra a RAM frissitésről, a 8254 számláló/időzítő második csatornájának a kimenetéről, az I/O csatorna paritásellenőrző vonalának

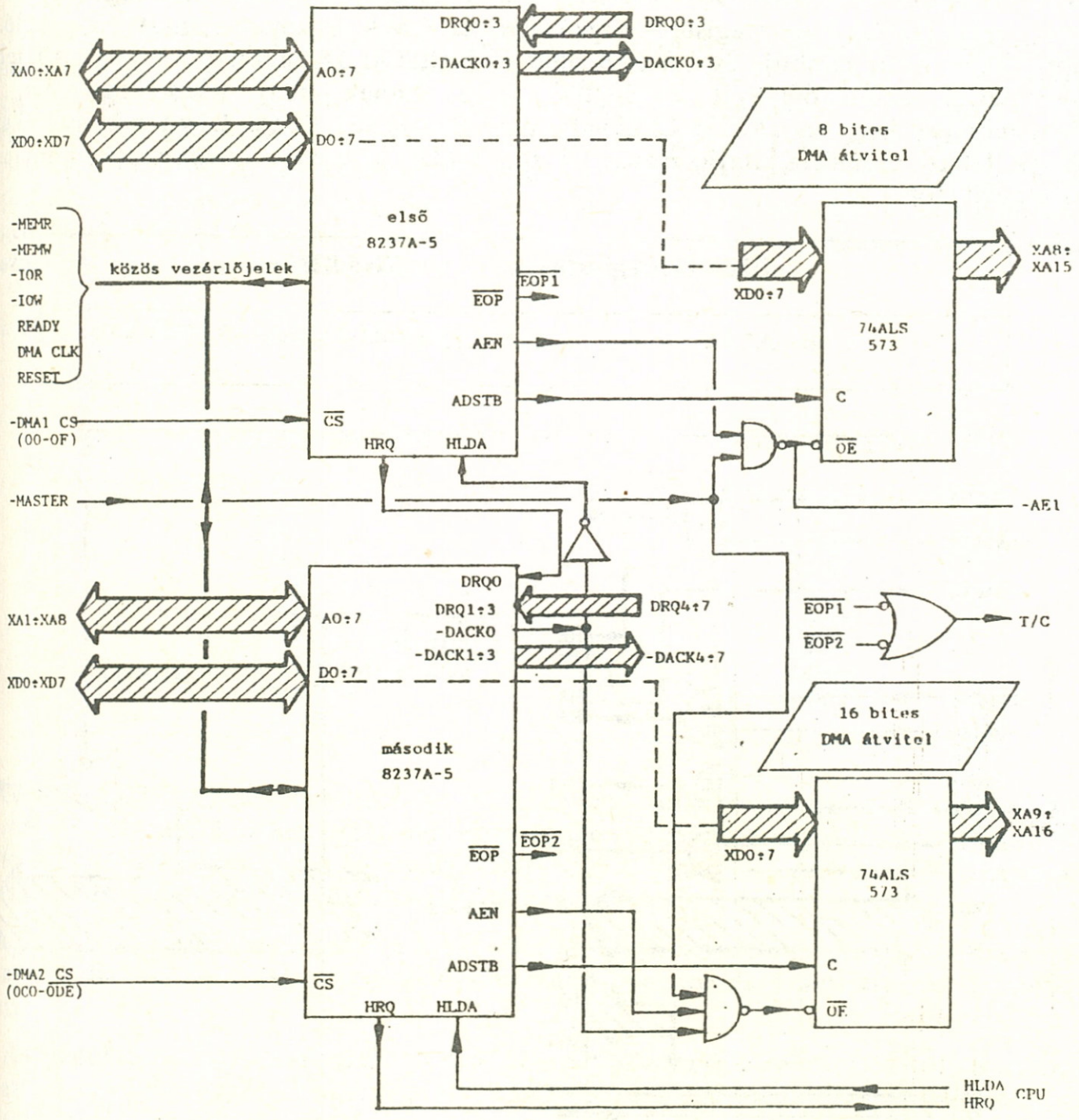
állapotáról és az alapkártya RAM paritás ellenőrző vonaláról adnak információt. A PPI 4-7 bemenő jelei beolvasáskor a belső I/O adatbusz XD4-XD7 vonalain keresztül kerülnek továbbításra. A PPI I/O címe: 0061h.



2-28 ábra.
Programozható parallel interface (PPI).

2.3.6 Az IBM AT DMA csatornái

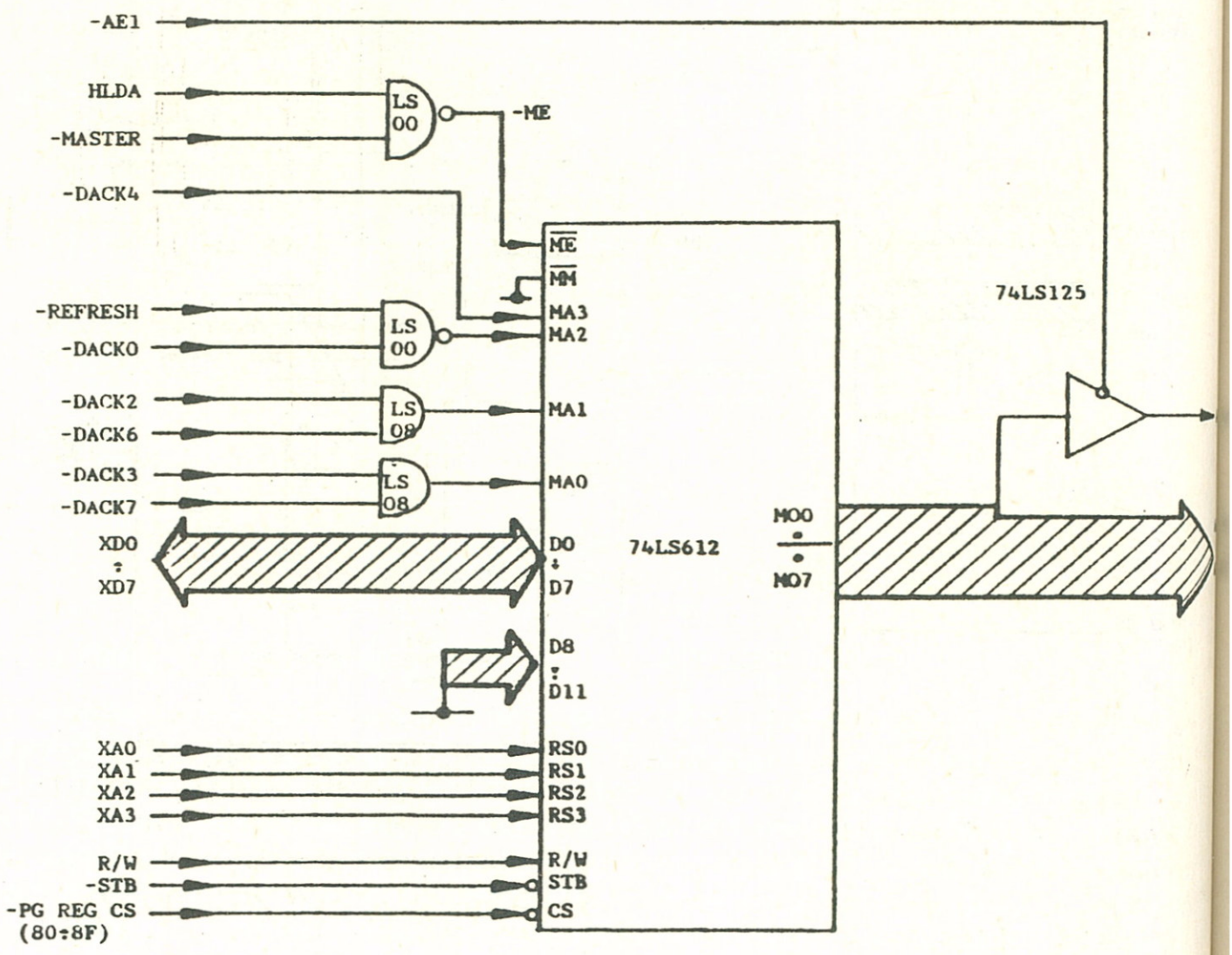
Az AT gépeknél a közvetlen memória hozzáférési (DMA) lehetőségeket megnövelték az XT gépekhez viszonyítva. Két 8237A-5 típusú vezérlő került az alaplemezen beépítésre, amely hét DMA csatorna kiszolgálására alkalmas. A két vezérlő kaszkádba kapcsolt, az első HRQ (Hold Request) kimenete a második DREQ0 bemenetére csatlakozik. Mindkét vezérlő számára a busz-hozzáférést a második 8237A vezérlő HRQ vonala kezdeményezi a processzor felé. Az első vezérlő részére a DMA kérést nyugtázó jelet (HLDA = Hold Acknowledge) a DACK0 kimenet szolgáltatja. A DMA csatornákat működtető áramkörök kapcsolási vázlata a 2-29 ábrán látható.



2-29 ábra (I. rész).
DMA vezérlő áramkörök.

Az első 8237A vezérli a 0,1,2 és 3. csatornákat. Ezeken a DMA csatornákon 8 bite adatátvitelt lehet megvalósítani 8 bites I/O adapterek és 8 vagy 16 bites memóriák között. Valamennyi csatorna a teljes 16 Mbyte-os címtartományban használható maximum 64 Kbyte-os adatblokkok átvitelére. A 16 Mbyte-os tartományban a címezhetőséget egy DMA lapregiszter (DMA Page Register), vagy másnéven DMA szegmens regiszter segíti elő. A lapregiszter nyolc címbitje és a 8237A által szolgáltatott 16 címbit (24 bit) az AT teljes kiterjesztett címtartományát elérhetővé teszi. Ennek nagy jelentősége van, ha meggondoljuk, hogy így a 80286 CPU valós módjában is kezelhetők DMA-n keresztül az 1 Mbyte feletti memória-kiterjesztések. A 0-3 DMA csatornák címképzését mutatja a következő tábla:

DMA lapregiszterek	Első 8237A-5
Cím: A23 < ---------- > A16	A15 < ---------- > A0



2-29 ábra (II. rész).
DMA lapregiszter.

DMA ciklusok alatt a 8237A vezérlő a 0-7 címbiteket az A0-A7 címkivezetéseire, a 8-15 címbiteket pedig a D0-D7 adatkivezetéseire helyezi. Egy 74ALS573 típusú regiszter a DMA műveletek idejére az XD0-XD7 adatvonalakat az XA8-XA15 címvonalakra kapcsolja, így az XA0-XA15 vonalakon a 16 bites cím rendelkezésre áll.

A 24 bites cím felső 8 bitjét a DMA lapregiszter A16-A23 kimenetei szolgáltatják. A lapregiszter egy 74LS612 típusú memória-mapper: 16 darab 12 bites kettős címzésű regisztert tartalmaz. Az RS0-RS3 címvonalakkal a regiszterek az adatbusz felől kiválaszthatók, egyenként beírhatók és kiolvashatók (R/W, -STB, -PG REG CS). Az MA0-MA3 címvonalakkal kiválasztott regiszter a mapper MO0-MO11 kimenetéhez kapcsolódik. A regiszter tartalma csak akkor kerül az A16-A23 vonalakra, ha a kimenetet engedélyező -ME jel (Map Enable) TTL0 állapotot vesz fel. A 16 regiszter közül nyolcat használnak, és a 12 bitből 8 vesz részt a címképzésben.

A nyolc lapregiszter I/O címe a következő:

A regiszter funkciója	I/O cím (hex)
DMA 0. csatorna	0087
DMA 1. csatorna	0083
DMA 2. csatorna	0081
DMA 3. csatorna	0082
DMA 5. csatorna	008B
DMA 6. csatorna	0089
DMA 7. csatorna	008A
Frissítés	008F

A memória-mapper MO0 kimenete egy 74LS125 "kapcsolón" keresztül csatlakozik az A16 címvonalra. Ha a 0-3 DMA csatornák valamelyike működik, a kapcsoló zárt. Ha az 5-7 csatornákon történik átvitel, a kapcsoló nyitott. Ekkor ugyanis az A16 címbitet a második 8237A vezérlő állítja elő. (Erről a következőkben lesz szó.)

A 0-3 csatornák működése esetén a kaszkádba kötött 4. csatorna is aktív állapotba kerül, de a második 8237A a cím- és vezérlő-kimeneteit ilyenkor nagyimpedanciás állapotban tartja.

Az AT gépek 0-3 DMA csatornája felülről kompatibilis az XT gépek DMA csatornáival. Ez azt jelenti, hogy ugyanolyan 8 bites átvitelekre nyílik mód, mint az XT gépek esetén, de a 0. csatorna is kihasználható (a 36 pontos I/O csatlakozón). Ezenkívül a DMA lapregiszter 16 Mbyte címtartományt engedélyez, szemben az XT DMA szegmens regiszterével, amely a 8088 CPU 1 Mbyte-os címtartományát használta. Egy további lehetőség, hogy az I/O adapter kártyákon is elhelyezhetők kaszkádba kötött 8237A vezérlők, így a DMA csatornák száma szükség esetén megnövelhető.

Az XT gépeknél erre nincsen mód. Az I/O kártyára épített kaszkád kapcsolású 8237A, vagy a buszt vezérelni szándékozó CPU valamelyik DMA csatornához csatlakozik. Az I/O

modul DMA kérését nyugtázó -DACK jel megérkezése után az I/O csatorna -MASTER vonalát logikai 0 állapotban kell tartani arra az időre, amíg a külső CPU, vagy a 8237A használja a cím-, adat- és vezérlő-vonalakat. Ez az időtartam 15 μsec-nál nem lehet hosszabb, különben frissítés hiányában a DRAM-ok elvesztik tartalmukat. A 2-29 (I) és 2-29 (II) ábrákon látható, hogy a -MASTER vonalra adott TTL0 szint letiltja mind a lapregiszter, mind a DMA vezérlők (alapkártya) kimeneteinek a címvonalakra való kapcsolódását.

Az eddigiekből megállapíthatjuk, hogy az AT gépek multiprocesszoros környezet kialakítására is alkalmasak, vagyis olyan intelligens I/O modulokkal képesek együttműködni, amelyek ha engedélyt kapnak, ugyanúgy vezérelhetik a rendszer összes buszát, mint a 80286 CPU.

A 0-3 DMA csatornákkal kapcsolatban a következő megjegyzést kell tenni: DMA ciklusok alatt a felső adatbyte-ot kiválasztó -BHE jel (Byte High Enable) mindig az A0 címbit inverze.
A második 8237A vezérli a 4-7 csatornákat. A 4. csatorna kaszkád módban működik. Az 5,6 és 7. csatornákon 16 bites adatátvitelt lehet megvalósítani 16 bites adapterek és 16 bites szószervezésű memóriák között. Ezek a csatornák maximum 128 Kbyte-os adatblokkokat tudnak kezelni a 16 Mbyte-os címtartományban. A címzésnél figyelembe kell venni egy megkötést: a 16 bites szavak csak páros címekre eshetnek.

Az 5-7 DMA csatornák címképzését mutatja a következő tábla:

DMA lapregiszterek	Második 8237A-5
Cím: A23 <---------- > A17	A16 <---------- > A1

A 2-29 ábrán is láthatjuk, hogy a második 8237A vezérlő címkimeneteihez kapcsolódó címvonalak 1-el eltolódtak. Tehát az A0 az XA1-hez, az A7 az XA8-hoz, a D0 az XA9-hez és a D7 az XA16-hoz kapcsolódik. DMA ciklusok alatt a -BHE és A0 jelek logikai 0 állapotot vesznek fel (16 bites szóátvitel). Mivel az A0 mindig 0, ezzel magyarázható, hogy a címek csak párosak lehetnek.

A címvonalak eltolódásából három dolog következik:

- a második 8237A 128 Kbyte címterületet tud kezelni
- a DMA lapregiszter A16 címvonalra kapcsolt kimenetét a 74LS125 kapcsolóval meg kell szakítani
- a második 8237A vezérlőhöz tartozó I/O címek csak párosak lehetnek

A második 8237A I/O címtartománya: 0C0h-0DFh
Az első 8237A I/O címtartománya: 00 - 0Fh

A két DMA vezérlő I/O címeit és a hozzátartozó regiszter funkciókat a 2-12 táblázat tartalmazza.

Cím (hex) Első 8237A	Második 8237A	Regiszter funkciók
00	C0	CH0 bázis cím és pillanatnyi cím tárolás
01	C2	CH0 bázis ciklusszám és pillanatnyi ciklusszám
02	C4	CH1 bázis cím és pillanatnyi cím
03	C6	CH1 bázis ciklusszám és pillanatnyi ciklusszám
04	C8	CH2 bázis cím és pillanatnyi cím
05	CA	CH2 bázis ciklusszám és pillanatnyi ciklusszám
06	CC	CH3 bázis cím és pillanatnyi cím
07	CE	CH3 bázis ciklusszám és pillanatnyi ciklusszám
08	D0	Státusz regiszter kiolvasása/vezérlő regiszter beírása
09	D2	DMA csatornákat engedélyező regiszter beírása
0A	D4	Maszk regiszter bitenkénti beírása
0B	D6	Mód regiszter beírása
0C	D8	A byte-mutató bistabil törlése
0D	DA	Az átmeneti adattároló regiszter kiolvasása/általános törlés
0E	DC	A maszk regiszter törlése
0F	DE	A maszk regiszter összes bitjének a beírása

2-12 táblázat.
Az IBM AT DMA vezérlőinek I/O címei.

Az 5,6 és 7 DMA csatornák programozásakor a következők szerint kell eljárni:

- a báziscím regiszterbe a báziscím kettővel osztott értékét kell beírni
- a bázis ciklusszám regiszterbe beírt érték az átviendő 16 bites szavak számát jelenti
- a DMA lapregiszter programozásakor a D1-D7 adatbitek tartalmazzák a felső A17-A23 címbiteket. A D0 adatbitet nem használják az 5-7 csatornák programozásakor

A tápfeszültség bekapcsolásakor, inicializáláskor, valamennyi belső regisztert fel kell tölteni, még akkor is, ha egyes csatornákat nem használnak.

A hét DMA csatornához rendelt I/O eszközöket mutatja a következő táblázat:

Első vezérlő	Második vezérlő
Ch0 - Fenntartva	Ch4 - Kaszkád csatorna az első vezérlő részére
Ch1 - SDLC szinkron kommunikációs adapter	Ch5 - Fenntartva
Ch2 - Floppy diszk vezérlő	Ch6 - Fenntartva
Ch3 - Fenntartva	Ch7 - Fenntartva

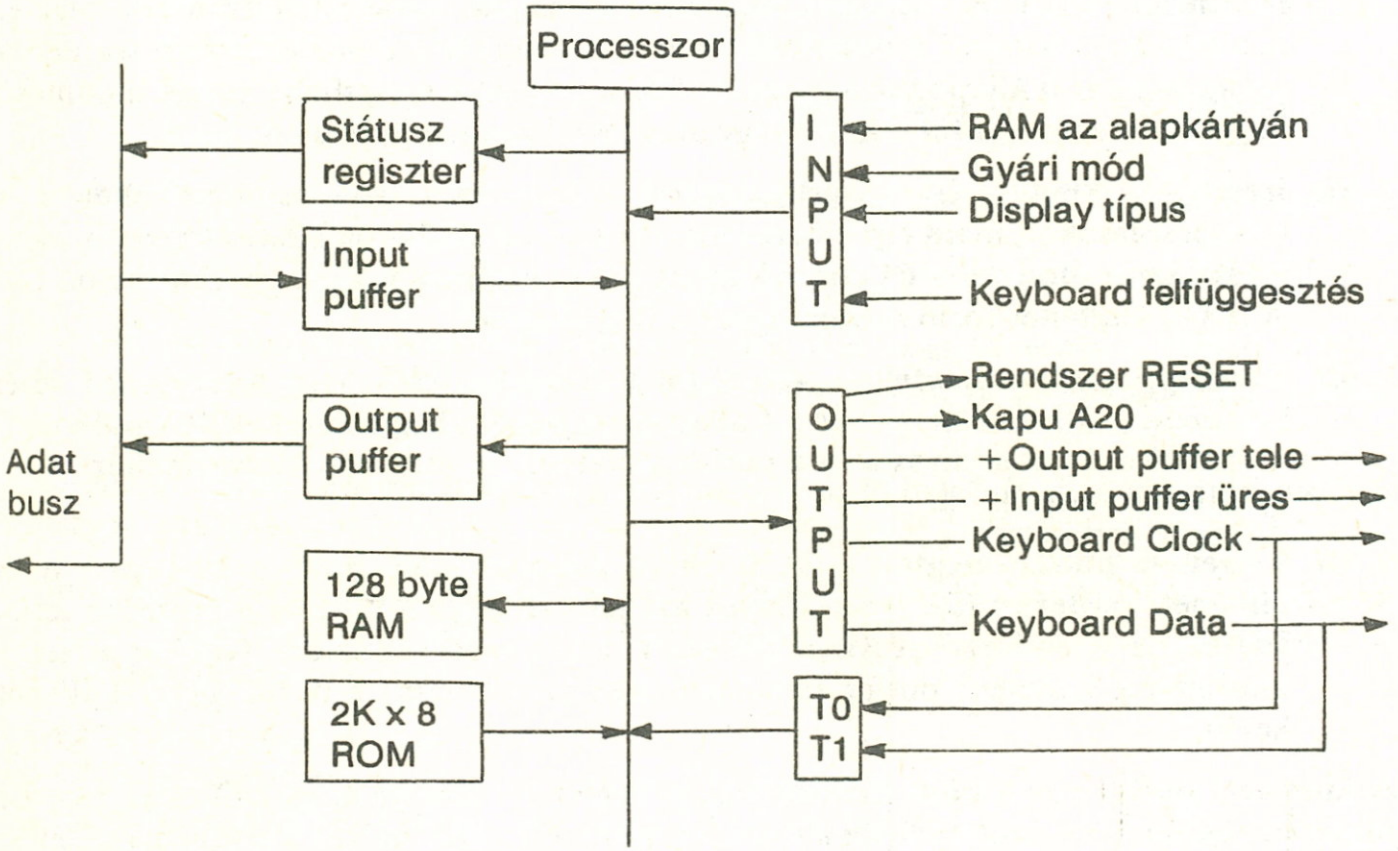
2-13 táblázat.
Az IBM AT DMA csatornáihoz rendelt I/O eszközök.

2.3.7 Interface a billentyűzet részére

Az AT alapkártyán még két olyan I/O modul található, amely eddig nem került ismertetésre: a keyboard interface, és a teleppel ellátott óraáramkörök/CMOS memóriák (2-21. ábra blokkdiagram). A keyboard interface egy Intel 8042 típusú mikroprocesszor, amit úgy programoztak, hogy kommunikálni tudjon a billentyűzet soros jeleivel. Az interface ellenőrzi a billentyűzetről érkező soros adatjelek paritását, értelmezi és átalakítja a 'scan' kódokat és átadja a rendszer részére, mint a kimeneti pufferében tárolt adatbyte-ot. A 8042 vezérlő vagy megszakítás kéréssel fordulhat a rendszerhez, ha a kimeneti pufferébe adat került, vagy megvárja míg a program lekérdezi a státusz regiszterét, hogy van-e érvényes billentyűkód a kimeneti pufferben.

A 8042 interface vezérlőn keresztül adatok is küldhetők a keyboard részére. Ekkor a program először megvizsgálja a státusz regiszter kiolvasásával, hogy a bemeneti puffer kész-e az adatfogadásra, majd ha ez a feltétel teljesül, beírja a pufferbe a kiküldendő információt. A pufferből a billentyűzethez kiküldött soros adatok automatikusan páratlan paritásbittel egészítődnek ki. Az interface-től a keyboard felé történő adatforgalom esetén is szükséges az adatok nyugtázása. A program nem küldhet addig újabb adat byte-ot, míg nem kap visszajelzést, hogy az előző byte már a keyboard-hoz került. Az 'output-buffer-full' interrupt vonal (IRQ1) mind az adást, mind a vételt kezelő rutinokhoz használható.

A 8042 keyboard vezérlő interface blokkdiagramja a 2-30 ábrán látható. A vezérlő a rendszer adatbuszával a státusz regiszteren, és a bemeneti/kimeneti puffereken keresztül tart kapcsolatot. A 8042 processzor, amely egy egy tokban elhelyezett mikrogép, 128 byte RAM, 2Kbyte ROM memóriával, és I/O portokkal rendelkezik.



2-30 ábra.
A 8042 keyboard-vezérlő interface blokkdiagramja.

A státusz regiszter csak olvasható, az I/O címe hex 64. A regiszter bitjeinek definíciója:

7. bit **Paritás hiba.** -- Ha a billentyűzetről küldött utolsó byte páratlan paritású volt, a 7. bit értéke 0. Páros paritás esetén ez a bit 1. Mivel a billentyűzet az adatokat páratlan paritással kell, hogy küldje, a 7. bit 1 értéke hibára utal.

6. bit **'Time-Out' vétel** -- Ha a 6. bit értéke 1, az átvitelt a keyboard megkezdte, de nem fejezte be a beprogramozott 'Time-Out' időn belül.

5. bit **'Time-Out' adás** -- Ha az 5. bit értéke 1, az átvitelt a keyboard vezérlő megkezdte, de nem fejezte be a specifikált idő-határon belül. Ha az adat byte-ot a vezérlő kiküldte a specifikált 'time-out' időn belül, de nyugtázó választ ezalatt nem kapott, a 6. és az 5. státusz bitek logikai 1 értéket vesznek fel.

4. bit **Tiltás kapcsoló állapota** -- A 4. bit a keyboard interface tiltott vagy engedélyezett állapotáról tájékoztat, a bemenő portra csatlakozó keyboard-tiltó előlapi kapcsoló állásának megfelelően. Valahányszor a 8042 kimeneti pufferébe új adat kerül, a 4.

bit az aktuális tiltott/engedélyezett értéket veszi fel. A 0 érték az interface tiltott állapotára utal.

3. bit **Parancs/Adat** -- A keyboard-vezérlő bemeneti pufferét a hex 60 vagy hex 64 I/O címekkel lehet megszólítani. Ha a hex 60 címet használjuk, a pufferbe küldött információ adatként, ha a hex 64 címet használjuk, parancsként van értelmezve. A hex 64 címre történő írás művelet a 3. bitet logikai 1 a hex 60 címre történő írás pedig logikai 0 állapotba helyezi. A 8042 használja ezt a bitet annak eldöntésére, hogy a bemeneti puffer adattal vagy utasítással van-e feltöltve.

2. bit **Rendszer Flag** -- A vezérlő logikai 0-ba állítja a 2. státusz bitet a gép bekapcsolásakor, majd egy sikeres teszt után a bit értékét logikai 1-re változtatja. A bit software úton a hex 64 címre kiadott I/O utasítással beállítható/törölhető. A BIOS általános törlő rutinja figyeli a flag állapotát.

1. bit **Bemeneti puffer megtelt** -- Az 1. bit 0 állapota jelzi, hogy a keyboard vezérlő bemeneti puffere még üres. Az 1 állapot azt jelenti, hogy adat került a pufferbe, de a vezérlő még nem olvasta ki a puffer tartalmát. A tartalom kiolvasásakor az 1. bit újból 0 állapotba kerül.

0. bit **Kimeneti puffer megtelt** -- A 0. bit 0 állapota jelzi, hogy a keyboard vezérlő kimeneti puffere üres. Az 1 állapot azt jelenti, hogy a vezérlő a pufferbe tette az adatot, de a rendszer még nem olvasta ki a kimeneti regisztert. Amikor a rendszer kiolvassa a kimeneti puffer tartalmát (hex 60 I/O cím) a 0. bit újból 0 állapotba kerül.

A kimeneti puffer egy 8 bites csak-olvasható regiszter, amelynek az I/O címe hex 60. A regisztert akkor lehet kiolvasni, ha a státusz regiszter 0. bitje logikai 1 értéket vesz fel.

A bemeneti puffer 8 bites, csak-írható regiszter, amely a hex 60, vagy a hex 64 I/O címen érhető el. A hex 64-es címen küldött byte-ot a vezérlő utasításként, a hex 60-as címen küldöttet pedig adatként értelmezi. A hex 60-as címen küldött adatot az interface a billentyűzethez továbbítja, kivéve azt az esetet, amikor az adat az előzőleg küldött utasításhoz tartozik. Az adatot akkor lehet a pufferbe írni, ha a státusz regiszter 1. bitje logikai 0 állapotban van.

A hex 64 I/O címen küldhető utasítás byte-ok értelmezése a következő:

20h **A keyboard-vezérlő utasítás byte-jának a kiolvasása** -- A vezérlő az aktuális utasítás byte-ját a kimeneti pufferbe helyezi.

60h **A keyboard-vezérlő utasítás byte-jának a beírása** -- Az ezután következő hex 60 I/O címen kiadott adat az utasítás regiszterbe kerül. Az utasítás-adatbitek definíciója a következő:

7. bit **Fenntartva** -- Logikai 0 állapotban kell tartani.

6. bit **IBM PC kompatibilis mód** -- Amennyiben a 6. bit értéke 1, a vezérlő a billentyűzet 'scan' kódjait az IBM PC által megkívánt alakúakra alakítja. A két byte-os 'break' szekvenciákat 1 byte-os IBM PC formátumúvá konvertálja.

5. bit **IBM PC mód** -- Ha ez a bit 1 értékű, a billentyűzet és az interface IBM PC kialakítású. Ebben a módban a vezérlő nem ellenőrzi a paritást és nem változtat a 'scan' kódokon.

4. bit **A billentyűzet működésének tiltása** -- Ha ez a bit 1 értékű, az interface tiltott állapotba kerül a 'clock' vonal logikai 0 szinten való tartásával. Ezalatt nem lehet sem adatot küldeni, sem adatot venni.

3. bit **Tiltás felülírás** -- Ha ez a bit 1 értékű, hatálytalanítja a keyboard tiltott állapotban való tartását.

2. bit **Rendszer Flag** -- Az utasítás 2. bitjének az értéke átkerül a státusz regiszter rendszer flag bitjébe.

1. bit **Fenntartva** -- Logikai 0 állapotban kell tartani.

0. bit **Az 'Output-Buffer-full' interrupt kimenet engedélyezése** -- Ha ez a bit 1 értékű, a vezérlő megszakítás kérést hoz létre minden alkalommal, amikor feltölti adattal a kimeneti puffert.

AAh **Önteszt** -- Ezen utasítás hatására a keyboard-vezérlő belső diagnosztikai tesztet hajt végre. Ha hiba nélkül fut le a tesztprogram, a kimeneti pufferbe hex 55 kerül.

ABh **Interface teszt** -- Ezen utasítás hatására a vezérlő a 'keyboard clock' és a 'keyboard data' vonalakat teszteli. A teszt eredménye a kimeneti pufferben a következőképpen jelenik meg:

00 A tesztprogram nem talált hibát
01 A 'keyboard clock' vonal állandóan TTL0 szinten van
02 A 'keyboard clock' vonal állandóan TTL1 szinten van
03 A 'keyboard data' vonal állandóan TTL0 szinten van
04 A 'keyboard data' vonal állandóan TTL1 szinten van

ACh **Diagnosztikai utasítás** -- A vezérlő RAM memóriájának 16 byte-ja, a bemenő port pillanatnyi állapota, a kimenő port pillanatnyi állapota és a státusz-szó átadásra kerül a rendszer-program részére.

ADh **A billentyűzet működésének tiltása** -- Ez a parancs az utasítás byte 4. bitjét állítja logikai 1 értékre. Ennek hatására a 'clock' vonal TTL0 szintre, a keyboard interface pedig tiltott állapotba kerül.

AEh **A keyboard interface engedélyezése** -- Ez a parancs törli az utasítás byte 4. bitjét, aminek következtében az interface újból üzemképessé válik.

C0h **A bemenő port olvasása** -- Az utasítás hatására a vezérlő beolvassa a bemenő port adatait, majd a kimeneti pufferbe helyezi őket. Az utasítást csak akkor szabad kiadni, ha a kimeneti puffer üres.

D0h **A kimenő port olvasása** -- Ezen utasítás hatására a vezérlő beolvassa a kimenő port adatait, majd a kimeneti pufferbe helyezi őket. Az utasítást csak akkor szabad kiadni, ha a kimeneti puffer üres.

D1h **A kimenő port beírása** -- Az ezután következő hex 60 I/O címen kiadott adat byte a vezérlő kimenő portjára kerül.

Megjegyzés: a kimenő port 0. bitje az általános törlő vonalhoz (system reset) van kapcsolva. Ezt a bitet nem szabad 0-ba állítani, mivel ekkor a processzor alaphelyzetbe kerül. (Kivéve ha program útján kell általános törlést végrehajtani. Pl: a Ctrl/Alt/Del billentyűk lenyomásakor.)

E0h A teszt bemenetek olvasása -- Ennek az utasításnak a hatására a vezérlő beolvassa a T0 és T1 bemenetek jeleit. Az adatot a kimeneti pufferbe helyezi. A T0 bemenetnek a 0. adatbit, a T1 bemenetnek pedig az 1. adatbit felel meg.

F0h-FFh Impulzus kiadása a kimenő porton -- A 8042 kimenő portjának 0-3 bitjei közelítőleg 6 μsec időtartamra logikai 0 állapotba kerülhetnek. Az utasítás 0-3 bitjei jelölik ki, hogy mely port-biteken jelenjen meg impulzus. A 0 érték kijelöli a port-bitet, az 1 érték pedig tiltja.

Megjegyzés: a 0. port bitre adott impulzus általános törlést idéz elő. (Alaphelyzet)

A 8042 keyboard vezérlő egy bemenő és egy kimenő I/O porttal rendelkezik. A két teszt bemeneten keresztül (T0,T1) a 'keyboard clock' és 'keyboard data' jelek kerülnek a rendszerhez továbbításra.
A 2-14 táblázat az I/O portok és a teszt-bemenetek bit-definícióit tartalmazza.

A bemenő port vonalaira csatlakozik a billentyűzet működését tiltó előlapi kulcs kapcsolója, a monitor típusát kiválasztó kapcsoló (színes, vagy egyszínű), az alapkártya RAM területét beállító 'jumper' és egy gyári 'jumper'.
A kimenő port vonalai szolgáltatják a billentyűzet részére az adat és órajeleket, ha a 8042 utasítást küld a keyboard-hoz, vezérlik ay IRQ1 interrupt bemenetet ('output buffer full', vagy 'input buffer empty'), általános törlőjelet generálhatnak és kapuzhatják a 80286 processzor A20 címvonalát. Ha az A20 kapujel logikai 0 értékű, az A20 címvonal tartósan logikai 0 állapotba kerül, függetlenül a 80286 CPU A20 címkivezetésén megjelenő jelektől.

A 8042 keyboard vezérlővel kapcsolatos funkciók a következők:

- a vezérlő inicializálása,
- a billentyűzetről küldött adatok vétele,
- a 'scan' kódok átalakítása rendszer 'scan' kódokká,
- utasítások küldése a billentyűzethez a vezérlőn keresztül,
- a billentyűzet tiltása az előlapon elhelyezett kapcsolóval (zár).

A gép bekapcsolásakor a 8042 vezérlő a rendszer flag bitet nullázza, a 'keyboard clock' vonalat pedig TTL0 szinten tartja. A keyboard interface paraméterek ekkor töltődnek a 8042 RAM memóriájába. Ezután a tiltás-felülírás utasításbitet a keyboard vezérlő logikai 1-be állítja, és hex 55 értéket tölt a kimeneti pufferbe, ha az önteszt hibátlanul lefutott. Bármely más érték a 8042 hibás működésére utal. A 'keyboard data' és 'keyboard clock' vonalak engedélyezett állapotba kerülnek, a vezérlő pedig utasításokat fogadhat a rendszertől, vagy keyboard adatokat olvashat be és továbbíthat.

Bemenő port-bit definíciók.

7. bit Billentyűzetet tiltó kapcsoló
0 = a billentyűzet tiltása
1 = a billentyűzet engedélyezése
6. bit Monitor adapter típust választó kapcsoló
(Elsődleges monitor)
0 = színes/grafikus adapter
1 = egyszínű adapter
5. bit Gyári jumper
0 = a jumper-pontok összekötöttek
1 = az átkötés hiányzik
4. bit Alapkártya RAM terület jumper
0 = 512K RAM memória engedélyezett
1 = 256K RAM memória engedélyezett
3. bit Fenntartva
2. bit Fenntartva
1. bit Fenntartva
0. bit Fenntartva

Kimenő port-bit definíciók.

7. bit Keyboard adat (kimenet)
6. bit Keyboard órajel (kimenet)
5. bit A bemeneti puffer üres
4. bit A kimeneti puffer megtelt
3. bit Fenntartva
2. bit Fenntartva
1. bit A20 címvonal kapujel (Gate A20)
0. bit Általános nullázójel

A bemeneti teszt-bitek definíciói.

T1 Keyboard adat (bemenet)
T0 Keyboard órajel (bemenet)

2-14 táblázat.
Keyboard vezérlő I/O bitek.

A billentyűzet az adatokat 11 bites soros formában küldi a 8042 processzorhoz. Az első start bitet nyolc adat bit, egy páratlan paritásbit és egy stop bit követi. A soros adat a keyboard által szolgáltatott órajelhez szinkronizált. Egy-egy 11 bites adat átvitele után a 8042 vezérlő mindaddig tiltott állapotban tartja az interface-t, míg a rendszer az adatot el nem fogadta. Ha az adat paritás hibával érkezett, egy 'Resend' utasítás az adat ismételt kiadására kéri a keyboard-ot. Ha a vezérlő a ROM BIOS által előírt számú ismétlések

után sem tud hibátlan adatot beolvasni, hex FF értékkel tölti fel a kimeneti puffert, és a státusz regiszter paritás bitjét 1-be állítja. Ha egy adat átvitele 2 msec-on belül nem fejeződik be, hex FF kerül a kimeneti pufferbe, és a státusz regiszter 'time-out' (vétel) bitje logikai 1 értékre változik. 'Time out' hibánál a software nem kísérel meg újabb kiolvasást.

Megjegyzés: amennyiben a 'default' módban (utasítás byte 5, 6 és 7 bitje 0) detektál hibát a vezérlő, a kimeneti pufferbe hex 00 kerül hex FF helyett.

A billentyűzet által szolgáltatott 'scan' kódokat a 8042 processzor átalakítja rendszer 'scan' kódokká. A különböző billentyűzetek (84, 101 és 102 gombos) billentyűkódjaiból képzett 'scan' kódokat (amelyeket a keyboard állít elő) a 8042 mégegyszer konvertálja, és így helyezi a kimeneti puffer-regiszterbe. (Ezek a billentyű kódok még nem ASCII kódok.) Az ASCII kódok, illetve az IBM karakter szet egy ezt követő újabb konverzió eredménye (ROM-BIOS KEYBOARD I/O rutinok és a KEYBxx program. Bővebb információ az XT/AT Technical Reference-ben és a DOS kézikönyvekben található.)
A 8042 processzor az átalakítás során a két tagból álló 'scan' kódokat egytagúakká, a három tagból állókat pedig kéttagúakká változtatja.

Ha a rendszer küld adatokat a keyboard-hoz, azaz az átvitel iránya az előzőekhez képest fordított lesz, ugyanazt a soros 11 bites formát használják mint a 'scan' kódok vételekor. A paritásbitet a 8042 keyboard vezérlő állítja elő és automatikusan elhelyezi minden kiadott adat utolsó bitje után. Ha a billentyűzet nem kezdi meg az adatok vételét a kiadástól számított 15 msec-on belül, vagy ha a vétel megkezdése után 2 msec alatt nem fejezi be azt, a 8042 hex FE értékkel tölti fel a kimeneti pufferét és a státusz regiszter 'time-out' (adás) bitjét logikai 1-be állítja.

A billentyűzet a számára kiadott adatokat a rendszer felé visszaküldött ACK válasszal nyugtázza (hex FA). A billentyűzet köteles valamennyi érvényes utasítást és az azokhoz tartozó paramétereket nyugtázni, kivéve az 'Echo' és 'Resend' utasításokat.

Ha az ACK válasz paritáshibával kerül a 8042 bemenetére, a vezérlő hex FE értéket tesz a kimeneti pufferébe, a státusz regiszter 'time-out' (adás) és 'parity error' bitjeit pedig logikai 1 értékre állítja.

A 8042 vezérlőt úgy programozták, hogy ha a billentyűzetről várt ACK nyugtázás 25 msec alatt nem érkezik meg, a vezérlő a kimeneti puffert hex FE értékkel tölti fel, és a 'time-out' (adás), 'time-out' (vétel) státusz biteket logikai 1-be állítja.
Ha az előbbi hibák bármelyike előfordul, a 8042 vezérlő nem tesz újabb kísérletet az átvitelre.
A gép előlapjára szerelt, kulccsal állítható kapcsoló a 8042 keyboard vezérlő működését befolyásolja. A kapcsoló a 8042 bemenő port 7. bitjéhez csatlakozik. A billentyűzet felé kiadott bármilyen utasítás a kapcsoló állásától függetlenül továbbhalad a vezérlőn. A billentyűzet által szolgáltatott adatokat a 8042 megvizsgálja, és ha az egy utasítás nyugtázása, vagy a keyboard állapotára vonatkozó információ, feltétel nélkül továbbengedi. Ha azonban a billentyűzetről jövő adat egy 'scan' kód, és a kulcs a billentyűzetet tiltó állásban van (bemenő port 7. bit = 0), az adat nem kerül továbbításra.

2.3.8 A teleppel ellátott MC146818 típusú óraáramkörök és CMOS RAM memóriák.

A Motorola MC146818 integrált áramköre 64 byte-os CMOS RAM memóriát és 'real-time' órát foglal magába. A CMOS RAM első 14 rekesze a belső óraáramkörök részére van fenntartva, a fennmaradó részt pedig a rendszer-konfigurációs információk tárolására használják.

A következő táblázat a CMOS RAM címtérképe.

Cím (hex)	Megnevezés
00-0D	* 'Real-time' óra információk
0E	* Diagnosztikai státusz byte
0F	* 'Shutdown' diagnosztikai státusz byte
10	Floppy diszkek típusai az A és B diszkekre vonatkozóan
11	Fenntartva
12	Merevlemezes diszkek típusai (1-14 típus választható)
13	Fenntartva
14	Rendszer konfiguráció
15	Alap memória méret - 'low byte'
16	Alap memória méret - 'high byte'
17	Memória bővítés mérete - 'low byte'
18	Memória bővítés mérete - 'high byte'
19	Merevlemezes diszktípusok 16-255 - C diszk
1A	Merevlemezes diszktípusok 16-255 - D diszk
1B-2D	Fenntartva
2E-2F	2 byte-os CMOS 'checksum'
30	* Memória bővítés mérete - 'low byte'
31	* Memória bővítés mérete - 'high byte'
32	* Az évszázad BCD formában
33	* Információs flag-ek
34-3F	Fenntartva

Megjegyzés: a *-gal jelölt byte-ok figyelmen kívül maradnak a 'checksum' kiszámításánál, és nem részei a konfigurációs rekordnak.

2-15 táblázat.
Az AT CMOS RAM címtérképe.

A címek nem közvetlen címeket jelentenek, hanem a CMOS RAM 'map' címeit. A CMOS RAM írása/olvasása két lépésban történik.

CMOS RAM írás:

1. OUT utasítás kiadása a hex 70 I/O címre. Az adat az a CMOS cím, amelyre az írás vonatkozik.
2. OUT utasítás kiadása a hex 71 I/O címre a CMOS memóriába írandó adattal.

CMOS RAM olvasás:

1. OUT utasítás kiadása a hex 70 I/O címre. Az adat az a CMOS cím ahonnan a memória-tartalmat ki kell olvasni.
2. IN utasítás kiadása a hex 71 I/O címre. Az AL regiszter által szolgáltatott adat a megcímzett CMOS memória-rekesz tartalma.

A CMOS RAM első 14 byte-ja órainformációkat tárol, a következő táblázat szerinti formában:

Byte	Megnevezés	Cím (hex)
0	Másodpercek - 'Real-time'	00
1	Másodpercek - 'Alarm'	01
2	Percek - 'Real-time'	02
3	Percek - 'Alarm'	03
4	Órák - 'Real-time'	04
5	Órák - 'Alarm'	05
6	A hét napja	06
7	A hónap dátum	07
8	A hónap	08
9	Az év	09
10	A státusz regiszter	0A
11	B státusz regiszter	0B
12	C státusz regiszter	0C
13	D státusz regiszter	0D

2-16 táblázat.
'Real-time' óra információk (címek: 00-0Dh).

Megjegyzés: Amikor a 'setup' program segítségével beállítjuk az időt és a dátumot, a program inicializálja az A, B, C és D regisztereket. Az 1Ah című BIOS interrupt az idő és a dátum kiolvasására/beírására szolgál. A státusz regisztereket ugyanúgy inicializálja mint a 'setup' program.

A státusz regiszterek bit-definíciói:

A REGISZTER (0Ah)

7. bit A tárolt adatok felfrissítése van folyamatban (UIP = Update In Progress) -- Az 1 érték a frissítésre utal. A 0 érték jelzi, hogy az aktuális dátum és idő kiolvasható.

4-6. bitek 22 fokozatú osztó (DV0-DV2) -- Ez a három bit az időalap kiválasztását végzi. A rendszer inicializálásakor a bitek a 010 értéket veszik fel, ami a 32768 kHz-es időalapot hozza létre.

0-3. bitek Az osztási arány kiválasztása (RS0-RS3) -- Ezek a bitek az osztó kimenetének a frekvenciáját állítják be. A rendszer inicializálásakor a bitek a 0110 értéket veszik fel, ami 1024 kHz-es négyszögjel kimenetet eredményez, és így az interrupt periódusidő 976,562 usec-ra adódik.

B REGISZTER (0Bh)

7. bit Beállítás -- A 0 érték azt jelzi, hogy a tárolt adatok másodpercenként az aktuális értéknek megfelelően módosulnak. Ha a bit 1 értékű, az 'update' ciklusok szünetelnek, és a program inicializálhatja a CMOS RAM 14 idő-információt tároló rekeszét. Az inicializálás után a 7. bitet 0-ba állítva az órát a software elindítja.

6. bit Periódikus interrupt engedélyezés (PIE = Periodic Interrupt Enable) -- A 6. bit, amely írható/olvasható, engedélyezi vagy tiltja az A regiszter által meghatározott periódusidővel jelentkező interruptot. A bit 1 állapota engedélyezést, 0 állapota tiltást jelent. A rendszer inicializálásakor a 6. bit logikai 0 állapotba kerül.

5. bit 'Alarm' interrupt engedélyezés (AIE = Alarm Interrupt Enable) -- A bit 1 állapota engedélyezi, 0 állapota tiltja az 'alarm' interruptot. A rendszer az 5. bitet inicializáláskor törli (logikai 0).

4. bit 'Update-ended' interrupt engedélyezés(UIE = Update-ended Interrupt Enable) -- A bit 1 állapota engedélyezi, 0 állapota tiltja az 'update-ended' interruptot. A rendszer a 4. bitet inicializáláskor törli (logikai 0).

3. bit Az időosztó kimeneti négyszögjelének engedélyezése (SQWE = Square Wave Enabled) -- A bit 1 állapota engedélyezi az A regiszter bitjei által meghatározott frekvenciájú négyszögjelet az időosztó kimenetén. A bit 0 állapota a kimenetet tiltja. A rendszer a 3. bitet inicializáláskor törli (logikai 0).

2. bit Adatforma (DM = Data Mode) -- Ez a bit azt jelöli, hogy az óra és dátum információknak az aktuális értékekre történő módosítása ('update') bináris, vagy binárisan kódolt decimális (BCD) formában megy-e végbe. A bit 1 állapota bináris, 0 állapota BCD formátumot jelent. A rendszer a 2. bitet inicializáláskor törli (logikai 0).

1. bit 24/12 formátum -- Ez a bit azt jelzi, hogy az óra byte 24 vagy 12 órás módban tárolja-e az információt. A logikai 1 a 24 órás módot, a logikai 0 a 12 órás módot jelöli. A rendszer az 1. bitet inicializáláskor törli (logikai 0).

0. bit A nyári időszámítás engedélyezése (DSE = Daylight Savings Enabled) -- A bit 1 állapotban a nyári időszámítást engedélyezi, 0 állapotban tiltja (standard time). A rendszer a 0. bitet inicializáláskor törli (logikai 0).

C REGISZTER (0Ch)

4-7. bitek UF, AF, PF, IRQF flag bitek -- Ezek a bitek csak olvashatók, és csak a B regiszter UIE, PIE, és AIE bitjeinek 1 állapotában aktívak.

0-3. bitek Fenntartottak -- Logikai 0 állapotban kell tartani mind a négy bitet.

D REGISZTER (0Dh)

7. bit A CMOS RAM-ban tárolt adatok érvényesek (VRB = Valid RAM Bit) -- Ez a bit csak olvasható és a CMOS RAM-hoz kapcsolt telep állapotát jelzi. A bit 1 értéke elfogadható telepfeszültséget jelent, a 0 érték pedig a kimerült, vagy hiányzó telepre utal.

0-6. bitek Fenntartottak -- Logikai 0 állapotban kell tartani mind a négy bitet.

A következő felsorolások a CMOS RAM konfigurációs byte-ok bit-definícióit tartalmazzák.

Diagnosztikai státusz byte (hex 0E)

7. bit A 'real time' óra-chip tápfeszültség státusza -- Az 1 érték azt jelenti, hogy az óra-chip tápfeszültség-ellátása rendben van, a 0 érték pedig hiányzó tápfeszültségre utal.

6. bit 'Checksum' státusz indikálás -- A bit 0 állapota a helyes 'checksum' értéket jelenti. Ha a bit 1 állapotban van, a 'checksum'-érték hibás.

5. bit Helytelen konfigurációs információ -- A konfigurációs byte-ban tárolt információt a bekapcsolási teszt ellenőrzi. Az 5. bit 0 állapota jelzi, hogy a konfigurációs információ megfelel a valóságnak, az 1 állapot pedig eltérésre utal.

A bekapcsolási teszt által megkívánt minimális feltételek:

- Legalább egy floppy diszk legyen a rendszerben (konfigurációs byte 0. bit = 1)
- Az elsődleges display adapter típusa (konfigurációs byte 4,5 bitek) és az alapkártya monitor kapcsolójának a beállítása illeszkedjenek egymáshoz, és a tényleges display adapter kártyához.

4. bit Memória-méret összehasonlítás -- A bit 0 állapota jelzi, hogy a bekapcsolási teszt ugyanakkora memória-méretet állapított meg mint amekkorát a konfigurációs rekord tartalmaz. Az 1 állapot eltérésre utal.

3. bit Merevlemezes diszkvezérlő adapter/C diszk inicializálás utáni státusza -- A bit 0 állapota jelzi, hogy az adapter és a diszk helyesen működik, és meg lehet kísérelni az operációs rendszer beolvasását ('boot up'). Ha a bit 1 állapotban van, az adapter kártya és/vagy a C diszk működésében hiba van. A 3. bit 1 állapota megakadályozza az operációs rendszer 'behúzását'.

2. bit Idő állapot-jelző -- A bit 0 állapota jelzi hogy az idő érvényes, az 1 állapot pedig azt, hogy nem felel meg a valóságnak.

0. és 1.bitek Fenntartottak.

'Shutdown' diagnosztikai státusz byte (hex 0F)

Ennek a státusz byte-nak a bitjei a bekapcsolási diagnosztikai teszt alkalmával kerülnek beállításra. (Bővebb információ található erről a byte-ról az AT Technical Reference BIOS listájában.)
A vezérlés visszaadásakor ('shutdown') a visszaadás okát rögzítik ebben a byte-ban. A byte tartalma felhasználható a visszatérési cím meghatározásához.

Floppy diszkek típusai (hex 10)

7-4. bitek Az első floppy diszk típusa:
0000 Az első floppy diszk hiányzik a rendszerből
0001 Kétoldalas floppy diszk (48TPI)
0010 Nagykapacitású floppy diszk (96TPI)
Megjegyzés: 0100-1111 fenntartott értékek

3-0. bitek A második floppy diszk típusa:
0000 A második floppy diszk hiányzik a rendszerből
0001 Kétoldalas floppy diszk (48TPI)
0010 Nagykapacitású floppy diszk (96TPI)
Megjegyzés: 0100-1111 fenntartott értékek.

A hex 11 címen egy fenntartott byte található.

Merevlemezes diszkek típusai (hex 12)

7-4. bitek Az első merevlemezes diszk típusa (C diszk):
0000 Az első merevlemezes diszk hiányzik a rendszerből
0001 Az első merevlemezes diszk típusa 1-14
.
. A diszk-típusok jellemzőit a 2-17 táblázat mutatja
.
1110
1111 Az első merevlemezes diszk típusa 16-255
A típus a C diszkhez rendelt kiterjesztésben található (hex 19 címen)

3-0. bitek A második merevlemezes diszk típusa (D diszk):
0000 A második merevlemezes diszk hiányzik a rendszerből
0001 A második merevlemezes diszk típusa 1-14
.
. A diszk-típusok jellemzőit a 2-17 táblázat mutatja
.
1110
1111 A második merevlemezes diszk típusa 16-255
A típus a D diszkhez rendelt kiterjesztésben található (hex 1A címen)

Típus	Sávok száma	Fejek száma	Írás elő-kompenzáció	Parkolási zóna
1	306	4	128	305
2	615	4	300	615
3	615	6	300	615
4	940	8	512	940
5	940	6	512	940
6	615	4	nincs	615
7	462	8	256	511
8	733	5	nincs	733
9	900	15	nincs	901
10	820	3	nincs	820
11	855	5	nincs	855
12	855	7	nincs	855
13	306	8	128	319
14	733	7	nincs	733
15	Kiterjesztett paraméterek (hex 19 és 1A)			

2-17 táblázat.
Merevlemezes diszk típusok 1-14. (A jellemzők megegyeznek az AT BIOS diszk-paraméter táblában definiáltakkal.)

A hex 13 címen egy fenntartott byte található

Rendszer konfigurációs byte (hex 14)

7-6. bitek A rendszerhez tartozó floppy diszkek száma:
00 1 floppy diszk
01 2 floppy diszk
10 Fenntartva
11 Fenntartva

5-4. bitek Elsődleges display
00 Az elsődleges monitor egy olyan adapter kártyához kapcsolt, amely saját BIOS-t tartalmaz.
Az adapter kártya a következők egyike lehet:
- Bővített grafikus adapter (EGA)
- Professzionális grafikus adapter (PGA)

01 Az elsődleges monitor 40 oszlopos módban egy színes grafikus adapterhez csatolt (CGA)

10 Az elsődleges monitor 80 oszlopos módban színes grafikus adapterhez csatolt (CGA)
11 Az elsődleges monitor egyszínű display és nyomtató adapterhez csatolt (MGA)

3-2. bitek Nem használtak

1. bit Aritmetikai processzor jelenlétét jelző bit:
0 Nincs aritmetikai processzor a gépben
1 A gép aritmetikai processzorral rendelkezik

0. bit Floppy diszk jelenlétét jelző bit:
0 Nincs floppy diszk a gépben
1 A gép floppy diszkkel rendelkezik.

Megjegyzés: A rendszer konfigurációs byte azt az alapkonfigurációt definiálja, amely a bekapcsolási diagnosztikai teszthez szükséges.

Alap memória méret (hex 15 és hex 16)

7-0. bitek hex 15 cím - Alap memória méret 'low' byte
7-0. bitek hex 16 cím - Alap memória méret 'high' byte
Memória méretek:
0100h 256K - alapkártya RAM
0200h 512K - alapkártya RAM
0280h 640K - 512K alapkártya RAM, IBM AT 128K memória bővítési opció, vagy a 128/640K memória bővítési opció.

Memória bővítés mérete (hex 17 és hex 18)

7-0. bitek hex 17 cím - Memória bővítés mérete 'low' byte
7-0. bitek hex 18 cím - Memória bővítés mérete 'high' byte
Méretek:
0200h 512K - Bővítő memória
0400h 1024K - Bővítő memória
0600h 1536K - Bővítő memória
. . .
. . .
3C00h 15360K - Bővítő memória (15M maximum)

C diszk típus-kiterjesztés (hex 19)

7-0. bitek Az első merevlemezes diszk típusa (C diszk):
00000000b-00001111b fenntartott értékek
00010000b-11111111b diszk típusok 16-255
A diszk típusok jellemzőit a 2.18 táblázat sorolja fel.

D diszk típus-kiterjesztés (hex 1A)

7-0. bitek A második merevlemezes diszk típusa (D diszk):
00000000b-00001111b fenntartott értékek
00010000b-11111111b diszk típusok 16-255

A diszk típusok jellemzőit a 2.18 táblázat sorolja fel.

Típus	Sávok száma	Fejek száma	Írás előkompenzáció	Parkolási zóna
16	612	4	összes sáv	663
17	977	5	300	977
18	977	7	nincs	977
19	1024	7	512	1023
20	733	5	300	732
21	733	7	300	732
22	733	5	300	733
23	306	4	nincs	336
24	Fenntartva			
.	.			
.	.			
255	Fenntartva			

2-18 táblázat.
Merevlemezes diszk típusok 16-23 (24-255 fenntartott típusok).
A diszk típusok azonosak az AT BIOS diszk paraméter táblában definiált típusokkal.

hex 1B-hex 2D Fenntartott címek

'Checksum' (hex 2E és 2F)

7-0. bitek hex 2E cím - 'Checksum low byte'
7-0. bitek hex 2F cím - 'Checksum high byte'

Megjegyzés: A 'checksum'-ot a hex 10-2D címeken található memória tartalmak összegezésével számítják.

Memória bővítés mérete (hex 30 és hex 31)

7-0. bitek hex 30 cím - Memória bővítés mérete 'low' byte
7-0. bitek hex 31 cím - Memória bővítés mérete 'high' byte

Méretek:
0200h 512K - Bővítő memória
0400h 1024K - Bővítő memória
0600h 1536K - Bővítő memória

3C00h 15360K - Bővítő memória (15M maximum)

Megjegyzés: Ezek a byte-ok az 1Mbyte feletti memóriaterületeket mutatják, ahogy a bekapcsolási teszt meghatározta a méretüket. A memória-kiterjesztés mérete a hex 15 számú BIOS interrupton keresztül határozható meg. Az alap-memória méretének meghatározására a hex 12 számú interrupt szolgál.

Évszázad byte (hex 32)

7-0. bitek Az évszázad BCD formában kifejezve. (A BIOS interface állíthatja be és használja a későbbiekben.)

Információs flag-ek (hex 33)

7. bit A bit logikai 1 értéke jelzi, hogy az alap-memória legfelső 128 Kbyte-ja rendelkezésre áll.

6. bit Ha ez a bit logikai 1 értékű, a kezdeti 'setup' után egy felhasználói üzenet kerül kiadásra.

5-0. bitek Fenntartottak.

hex 34 - hex 3F Fenntartott címek

2.3.9 Az IBM AT I/O címtérképe.

A címtérképet a 2-19 táblázat tartalmazza. A hex 00-FF tartományba eső címek az alapkártya I/O eszközeihez vannak hozzárendelve. A hex 100-3FF címeket az I/O csatorna adapterei használhatják.

I/O címek (hex)	Megnevezés
000-01F	Első DMA vezérlő 8237A-5
020-03F	Első interrupt vezérlő, 8259A (Master)
040-05F	Programozható számláló/időzítő, 8254-2
060-06F	Keyboard interface, 8042 mikroprocesszor. Programozható Parallel Interface, PPI (061 cím)
070-07F	'Real-time' óra, NMI maszk regiszter
080-09F	DMA lapregiszter, 74LS612
0A0-0BF	Második interrupt vezérlő, 8259A
0C0-0DF	Második DMA vezérlő, 8237A-5
0F0	Az aritmetikai processzor foglaltság jelének törlése
0F1	Aritmetikai processzor törlése (Reset)
0F8-0FF	Aritmetikai processzor
100-1EF	Nem használható

I/O címek (hex)	Megnevezés
1F0-1F8	Merevlemezes diszk
200-207	'Joystick' I/O
20C-20D	Fenntartva
21F	Fenntartva
278-27F	Második parallel nyomtató
2B0-2DF	Alternatív bővített grafikus adapter (EGA)
2E1	GPIB (0. adapter)
2E2&2E3	Adatgyűjtés (0. adapter)
2F8-2FF	Második RS232 soros interface
300-31F	Prototípus kártya
360-363	PC hálózat ('low address') 'PC Network'
364-367	Fenntartva
368-36B	PC hálózat ('high address') 'PC Network'
36C-36F	Fenntartva
378-37F	Első parallel nyomtató
380-38F	SDLC, második BSC
390-393	'Cluster'
3A0-3AF	Első BSC
3B0-3BF	Egyszínű monitor és nyomtató adapter
3C0-3CF	Bővített grafikus adapter (EGA)
3D0-3DF	Színes grafikus adapter (CGA)
3F0-3F7	Floppy diszk vezérlő
3F8-3FF	Első RS232 soros interface
6E2&6E3	Adatgyűjtés (1. adapter)
790-793	'Cluster' (1. adapter)
AE2&AE3	Adatgyűjtés (2. adapter)
B90-B93	'Cluster' (2. adapter)
EE2&EE3	Adatgyűjtés (3. adapter)
1390-1393	'Cluster' (3. adapter)
22E1	GPIB (1. adapter)
2390-2393	'Cluster' (4. adapter)
42E1	GPIB (2. adapter)
62E1	GPIB (3. adapter)
82E1	GPIB (4. adapter)
A2E1	GPIB (5. adapter)
C2E1	GPIB (6. adapter)
E2E1	GPIB (7. adapter)

2-19 táblázat.
Az IBM AT I/O címtérképe.

Néhány kiegészítés az I/O címekkel kapcsolatban:

- Bekapcsoláskor az NMI interrupt bemenet tiltott állapot kerül (masked off). A maszk bitet a következő utasításokkal lehet beírni, vagy törölni:
  'Mask on ' a hex 70 I/O címre kiadott írás utasítással az NMI interruptot engedélyezhetjük, ha a 7. adatbit logikai 0 értékű.
  'Mask off ' a hex 70 I/O címre kiadott írás utasítással az NMI interruptot tilthatjuk, ha a 7. adatbit logikai 1 értékű.
  A bekapcsolási teszt végén (POST = Power On Self-Test) az NMI interrupt tiltása megszűnik.
- A hex F0 címre küldött 8 bites OUT utasítás törli az aritmetikai processzor tárolt foglaltsági jelét ('busy'). A foglaltsági jel akkor kerül a 80287-ben egy 'latch'-ben tárolásra, ha az aritmetikai processzor 'error' és 'busy' kimenetei egyszerre tartózkodnak aktív állapotban. Az OUT utasítással küldött 8 bites adat 00 értékű kell legyen.
  A hex F1 címre küldött 8 bites OUT utasítás törli az aritmetikai processzort. Az adatbyte 00 értékű kell legyen.
- A hex 80 I/O címet diagnosztikai ellenőrzési célokra használják. A DMA lapregiszterek egyik regisztere taláható ezen a címen. (74LS612).
- Az I/O csatorna ellenőrző jele (-I/O CH CK) a RAM adapter kártyák nem-javítható hibáit jelzi a rendszer részére. Az ilyen típusú hibák NMI interruptokat okoznak, amennyiben az NMI engedélyezett. Bekapcsoláskor a -I/O CH CK jelnek nincs semmiféle hatása, mert az NMI tiltott. Az NMI és az -I/O CH CK engedélyezése a következő lépések szerint megy végbe:

  a.) Az I/O RAM adaptereket feltöltik valamilyen adattal; ez biztosítja a helyes paritást valamennyi memóriahely részére.

  b.) Engedélyezik a -I/O CH CK jelet.

  c.) Engedélyezik az NMI-t.

  Megjegyzés: az a, b, és c lépéseket a bekapcsolási teszt (POST) valósítja meg.

  Ha a -I/O CH CK vonal hibát jelez (logikai 0), NMI megszakításkérés jön létre. Az NMI-t okozó forrás a státusz bitek beolvasásával azonosítható. A hibás adapter helyét úgy lehet megállapítani, hogy az egyes adapterek tetszőleges helyére valamilyen adatot kell beírni. Amikor a beírás abba az adapterbe történik, amelyik a hibát jelezte, a -I/O CH CK vonal inaktív állapotba kerül (logikai 1).

2.3.10 Az IBM AT I/O csatornája.

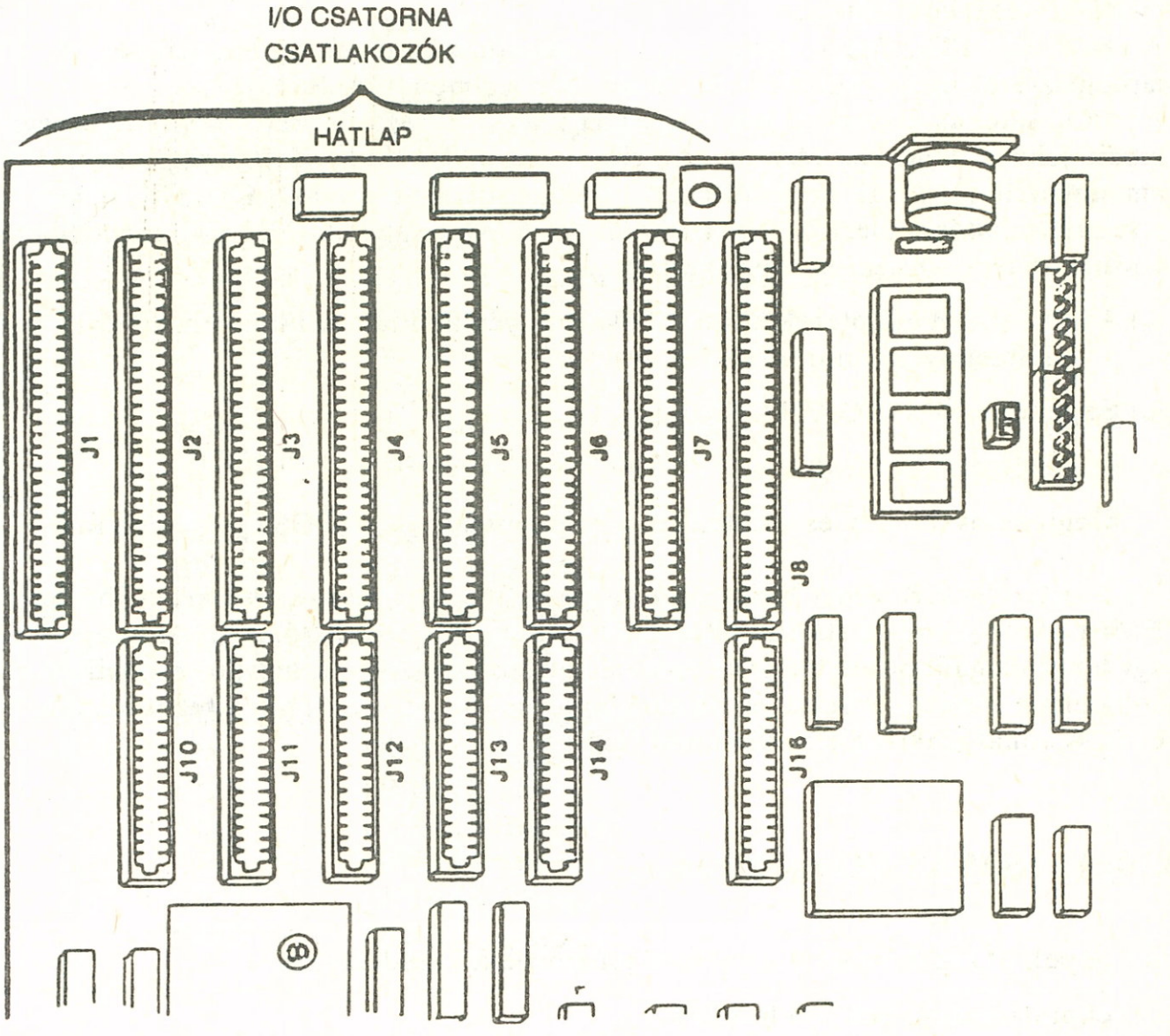
Az AT gépek I/O csatornája a következő jellemzőkkel rendelkezik:

- I/O címtartomány: hex 100-hex 3FF
- 24 bites memória címek (16Mbyte)
- 8 vagy 16 bites adat-hozzáférési lehetőség

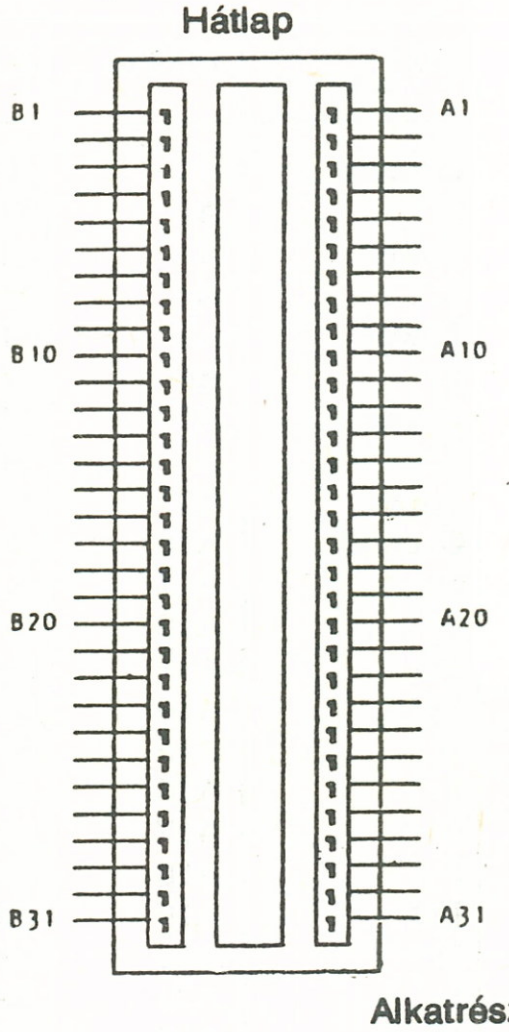
- 11 interrupt bemenet
- 7 DMA csatorna
- I/O várakozási ciklus generálás
- Nyitott busz-struktúra, amely lehetővé teszi, hogy több mikroprocesszor között is megoszthatók legyenek a rendszer 'erőforrásai' (beleértve ebbe a memóriát is)
- A rendszer RAM frissítése az I/O csatornára csatlakozó mikroprocesszorról.

A 2-31 ábra az I/O csatorna csatlakozók elhelyezkedését és számozását mutatja. 8 darab 62 pontos és 6 darab 36 pontos csatlakozó aljzat került beépítésre az alaplemezen. A 36 pontos csatlakozók két helyen hiányoznak az alaplemezről: ezeken a pozíciókon csak 62 pontos I/O adapterek helyezhetők el.

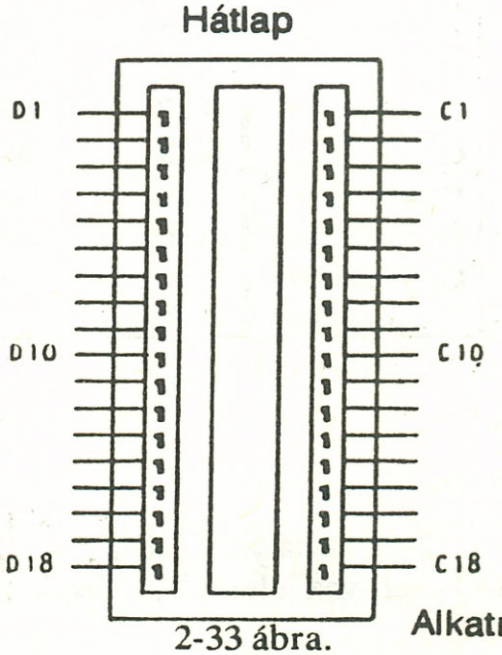
A 2-32 ábrán a 62 pontos, a 2-33 ábrán pedig a 36 pontos csatlakozó aljzatok kivezetéseinek a számozását láthatjuk. A 2-20 táblázat az I/O csatorna jeleit sorolja fel, a csatlakozási pontok sorrendjében. A táblázat három oszlopból áll: az első a csatlakozási pontok számát, a második a jelek neveit, a harmadik pedig a jelek irányát mutatja.



2-31 ábra.
Az I/O csatorna elhelyezkedése az alaplemezen (AT).



2-32 ábra.
Az I/O csatorna csatlakozópontjainak számozása (J1-J8 csatlakozók).



2-33 ábra.
Az I/O csatorna csatlakozópontjainak számozása (J10-J14 és J16 csatlakozók).

I/O Pin	A jel neve	I/O	I/O Pin	A jel neve	I/O
B1	GND	Ground	A1	-I/O CH CK	I
B2	RESET DRV	0	A2	SD7	I/O
B3	+5Vdc	Power	A3	SD6	I/O
B4	IRQ9	I	A4	SD5	I/O
B5	-5Vdc	Power	A5	SD4	I/O
B6	DRQ2	I	A6	SD3	I/O
B7	-12Vdc	Power	A7	SD2	I/O
B8	OWS	I	A8	SD1	I/O
B9	+12Vdc	Power	A9	SD0	I/O
B10	GND	Ground	A10	-I/O CH RDY	I
B11	-SMEMW	O	A11	AEN	O
B12	-SMEMR	O	A12	SA19	I/O
B13	-IOW	I/O	A13	SA18	I/O
B14	-IOR	I/O	A14	SA17	I/O
B15	-DACK3	O	A15	SA16	I/O
B16	DRQ3	I	A16	SA15	I/O
B17	-DACK1	O	A17	SA14	I/O
B18	DRQ1	I	A18	SA13	I/O
B19	-Refresh	I/O	A19	SA12	I/O
B20	CLK	O	A20	SA11	I/O
B21	IRQ7	I	A21	SA10	I/O
B22	IRQ6	I	A22	SA9	I/O
B23	IRQ5	I	A23	SA8	I/O
B24	IRQ4	I	A24	SA7	I/O
B25	IRQ3	I	A25	SA6	I/O
B26	-DACK2	O	A26	SA5	I/O
B27	T/C	O	A27	SA4	I/O
B28	BALE	O	A28	SA3	I/O
B29	+5Vdc	Power	A29	SA2	I/O
B30	OSC	O	A30	SA1	I/O
B31	GND	Ground	A31	SA0	I/O
	I/O csatorna (B oldal J1-től J8-ig)			I/O csatorna (A oldal J1-től J8-ig)	

2-20 táblázat (I. rész).
Az I/O csatorna jeleinek a felsorolása.

I/O Pin	A jel neve	I/O	I/O Pin	A jel neve	I/O
D1	-MEM CS16	I	C1	-SBHE	I/O
D2	-I/O CS16	I	C2	LA23	I/O
D3	IRQ10	I	C3	LA22	I/O
D4	IRQ11	I	C4	LA21	I/O
D5	IRQ12	I	C5	LA20	I/O
D6	IRQ15	I	C6	LA19	I/O
D7	IRQ14	I	C7	LA18	I/O
D8	-DACK0	O	C8	LA17	I/O
D9	DRQ0	I	C9	-MEMR	I/O
D10	-DACK5	O	C10	-MEMW	I/O
D11	DRQ5	I	C11	SD08	I/O
D12	-DACK6	O	C12	SD09	I/O
D13	DRQ6	I	C13	SD10	I/O
D14	-DACK7	O	C14	SD11	I/O
D15	DRQ7	I	C15	SD12	I/O
D16	+5Vdc	Power	C16	SD13	I/O
D17	-MASTER	I	C17	SD14	I/O
D18	GND	Ground	C18	SD15	I/O

I/O csatorna (D oldal J10-től J14-ig és J16)

I/O csatorna (C oldal J10-től J14-ig és J16)

2-20 táblázat (II. rész).
Az I/O csatorna jeleinek a felsorolása.

AZ AT I/O CSATORNA JELEINEK A LEÍRÁSA

Az I/O csatorna valamennyi vonala TTL kompatibilis. Az I/O adapterek max. két 'low-power Schottky (LS)' egységterheléssel csatlakozhatnak az I/O vonalakra.
A jelek elnevezéseinél a név elé helyezett - jelzés a jel logikai 0 aktív állapotára utal (negált jelek).

SA0-SA19 (I/O)

Az SA0-19 címvonalak a rendszer memóriáinak és I/O eszközeinek címzésére szolgálnak. Ez a 20 címvonal az LA17-LA23 vonalakkal kiegészítve 16Mbyte memóriaterület használatát teszi lehetővé. Az SA0-19 jelek akkor veszik fel a mikroprocesszor által küldött aktuális értékeket, amikor a BALE = 'Buffered Address Latch Enable' vezérlőjel TTL1 állapotba kerül. A BALE 1 -- >0 visszafutó éle zárja a cím 'latch'-eket és rögzíti a

címinformációt. A címeket vagy a processzor, vagy a DMA vezérlő generálja. Ezenkívül az I/O csatornára kapcsolódó további mikroprocesszorok is előállíthatják a címeket.

LA17-LA23 (I/O)

Ezek a jelek memóriacímzésre és I/O eszközök címzésére használatosak a rendszeren belül. A jeleket nem írják címregiszterekbe, csak bufferekben vezetik keresztül. Ezek a címvonalak biztosítják a 16Mbyte-os címezhetőséget. A jelek akkor érvényesek, ha a BALE vonal TTL1 állapotú. Mivel az LA17-LA23 biteket nem tárolják regiszterekben, a cím nem érvényes a teljes processzor ciklus alatt. Az LA17-LA23 vonalakat memória-címdekódolásra használják 16 bites, 1 várakozási ciklussal bővített átvitelek esetén. A címdekóder kimeneteket az I/O adapter kártyáknak kell regiszterekben rögzíteniük a BALE jel visszafutó (1 -- >0) élénél.
Az I/O csatornán elhelyezkedő mikroprocesszorok és DMA vezérlők is meghajthatják ezeket a vonalakat.

CLK (O)

Ez a vonal adja a 6 vagy 8 MHz-es rendszer órajelet, amely szinkronban van a mikroprocesszor ciklusjelével. Az órajel periódusideje 167 nsec (125 nsec). A kitöltési tényező 50%. Ez a jel csak szinkronozási célokra használható, és nem használható fix frekvenciák előállítására.

RESET DRV (O)

Ez a jel a rendszer alapállapotba állítására, inicializálására szolgál a gép bekapcsolásakor, vagy a tápfeszültség kimaradásakor. A vonal aktív állapotban logikai 1 értéket vesz fel (TTL1).

SD0-SD15 (I/O)

Ezek a vonalak szállítják a 0-15 adatbiteket a mikroprocesszorok, memóriák és I/O eszközök részére. SD0 a legalsó bitet (LSB), SD15 a legfelső bitet (MSB) jelöli. Az I/O csatornára kapcsolódó 8 bites eszközöknek az SD0-SD7 vonalakat kell használniuk a CPU-val való kommunikáció céljára. A 16 bites eszközök az SD0-SD15 vonalakat használják. Azért hogy a 16 és 8 bites eszközök együttműködhessenek, az SD8-SD15 vonalak adatait az SD0-SD7 vonalakra kapuzzák a 8 bites átvitel idejére. A 16 bites mikroprocesszor átviteli ciklusokat két 8 bites átviteli ciklussá konvertálják, ha a CPU 8 bites eszközzel tart kapcsolatot.

BALE (O) (bufferelt)

A címleválasztást engedélyező BALE = 'Buffered Address Latch Enable' jelet a 82288 busz vezérlő állítja elő. Ez a jel a mikroprocesszor által kiadott címek leválasztására, és regiszterekbe írására szolgál. A jel lefutó 1 -- >0 éle rögzíti az SA0-SA19 biteket a regiszterekben. A jelváltás alatt a CPU vagy a DMA vezérlő által kiadott címek érvényesnek tekintendők. Ezt jelzi a BALE vonal az I/O adapter kártyák részére. (Az I/O

adapterek a BALE és az AEN = 'Address Enable' jeleket együtt használják). DMA ciklusok alatt a BALE vonal logikai 1 aktív állapotot vesz fel.

-I/O CH CK (I)

Ez a vonal az I/O csatornára csatlakozó adapterek működéséről ad információt (-I/O CH CK = -I/O channel check). Jelzi a 80286 CPU részére, ha az I/O csatornára csatlakozó memóriákban vagy perifériákban paritás hiba állt elő. A jel logikai 0 aktív állapota nem-javítható rendszer hibára utal.

I/O CH RDY (I)

I/O csatorna készenléti jel (I/O CH RDY = I/O channel ready). Készenlét esetén a vonal logikai 1 állapotot vesz fel. Az adapter kártyák ezt a vonalat logikai 0-ba húzhatják, hogy meghosszabbítsák a memória vagy I/O ciklusokat. Bármelyik lassú eszköznek azonnal le kell húznia ezt a vonalat, mihelyt felismerte a címét és az író vagy olvasó vezérlőjelet. A gépi ciklusok (memória vagy I/O) az órajel periódusok egész számú értékeivel kerülnek meghosszabbításra (167 vagy 125 nsec). 2,5 μsec-nál hosszabb ideig nem szabad a vonalat logikai 0 állapotban tartani.

IRQ3-IRQ7, IRQ9-IRQ12, IRQ14 és IRQ15 (I)

Ezek a megszakítást kérő vonalak informálják a processzort arról, hogy valamelyik I/O eszköz fel akarja hívni magára a figyelmet. Az interrupt kérések közül az IRQ9-IRQ12, IRQ14 és IRQ15 magasabb prioritásúak (IRQ9 a legmagasabb) mint az IRQ3-IRQ7 (IRQ7 a legalacsonyabb).

Megszakítás kérés akkor jön létre, ha az IRQ vonal állapota logikai 0-ról logikai 1-re vált. A vonalat mindaddig logikai 1-ben kell tartani, amíg a kérést nem nyugtázza a processzor (az interrupt kezelő rutinon keresztül).

Megjegyzés: Az IRQ0,1,2,8,13 interrupt bemenetek nincsenek az I/O csatornára kivezetve. Ezeket az interrupt vonalakat az alapkártyán elhelyezett I/O eszközök és a billentyűzet foglalják le:

IRQ0: rendszer óra
IRQ1: billentyűzet
IRQ2: második interrupt vezérlő
IRQ8: 'real time' óra
IRQ13: aritmetikai processzor

-IOR (I/O)

I/O olvasás vezérlőjele (-I/O read).
A -IOR jel arra készteti a kiválasztott I/O eszközt, hogy helyezze az adatbuszra az adatot. A -IOR jelet vagy az alapkártya mikroprocesszora/DMA vezérlője, vagy az I/O csatornához kapcsolt mikroprocesszor/DMA vezérlő állíthatja elő. A vonal aktív állapota logikai 0.

-IOW (I/O)

I/O írás vezérlőjele (-I/O write).
Egy I/O eszköz részére ez a vonal szolgáltatja a vezérlőjelet az adatbuszra helyezett információ beolvasásához. A -IOW jelet bármely, a rendszerhez tartozó mikroprocesszor vagy DMA vezérlő előállíthatja. A vonal aktív állapota logikai 0.

-SMEMR (O) -MEMR (O)

Memória olvasás vezérlőjelei (-memory read).
Az ezeken a vonalakon megjelenő vezérlőjelek hatására a kiolvasott memória rekesz tartalma az adatbuszra kerül.
A -SMEMR vonal csak akkor aktivizálódik, ha a memóriacím az első 1Mbyte-os címtartományba esik.
A -MEMR vonal valamennyi olvasási ciklus alatt aktív állapotba kerül. A -MEMR jelet bármely, a rendszerhez tartozó mikroprocesszor vagy DMA vezérlő előállíthatja. A -SMEMR jel az első 1Mbyte-os címdekóder kimenetével kapuzott -MEMR jel. Ha egy az I/O csatornához kapcsolt mikroprocesszor a forrása a -MEMR jelnek, a mikroprocesszornak a vonal aktivizálása előtt legalább egy óraciklussal korábban ki kell bocsátania a címvonalakra az érvényes, stabil címet. A -SMEMR és -MEMR vonalak aktív állapotban logikai 0 értéket vesznek fel.

-SMEMW (O) -MEMW (O)

Memória írás vezérlőjelei (-memory write).
Ezek a vonalak szolgáltatják a memória részére a vezérlőjelet az adatbuszon lévő információ beírásához.
A -SMEMW vonal csak akkor aktivizálódik, ha a memóriacím az első 1Mbyte-os címtartományba esik. A -MEMW vonal valamennyi memória írás ciklus alatt aktív állapotba kerül.
A -MEMW jelet bármely, a rendszerhez tartozó mikroprocesszor vagy DMA vezérlő előállíthatja. A -SMEMW jel az első 1Mbyte-os címdekóder kimenetével kapuzott -MEMW jel. Ha egy az I/O csatornához kapcsolt mikroprocesszor a forrása a -MEMW jelnek, a mikroprocesszornak a vonal aktivizálása előtt legalább egy óraciklussal korábban ki kell bocsátania a címvonalakra az érvényes, stabil címet. A -SMEMW és -MEMW vonalak aktív állapotban logikai 0 értéket vesznek fel.

DRQ0-DRQ3 és DRQ5-DRQ7 (I)

DMA kérelmek 0-3 és DMA kérelmek 5-7 (DMA request).
Ezeket a vonalakat az I/O eszközök és mikroprocesszorok használhatják aszinkron módon, DMA ciklusok kezdeményezésére. A DRQ0 vonal rendelkezik a legmagasabb, a DRQ7 pedig a legalacsonyabb prioritással. A kérést a DRQ vonal logikai 1 aktív állapota idézi elő. A DRQ vonalat mindaddig logikai 1 értéken kell tartani, míg a -DACK vonalon (DMA acknowledge) a kérés nyugtázása meg nem érkezik. A DRQ0-DRQ3 vonalak 8 bites, a DRQ5-DRQ7 vonalak pedig 16 bites DMA ciklusokat kezdeményeznek.

A DRQ4 vonalat az alapkártya első 8237A DMA vezérlője használja, és az I/O csatornára ez a vonal nincs kivezetve.

-DACK0 -- -DACK3 és -DACK5 -- -DACK7 (O)

DMA kérelmek nyugtázása 0-3 és 5-7 (-DMA acknowledge).
A vonalak aktív állapota logikai 0.
A DACK4 vonalat az alapkártya első 8237A DMA vezérlője használja, és az I/O csatornára ez a vonal nincsen kivezetve.

AEN (O)

Címzés engedélyezés (address enable).
Ez a jel DMA átvitel céljából lekapcsolja a mikroprocesszort és a többi eszközt az I/O csatornáról (felszabadítja a buszt). Amikor a vonal aktív állapotba kerül, a DMA vezérlő működtetheti a cím- és adatbuszokat, az olvasó és író vonalakat (memória és I/O). Az AEN vonal aktív állapota logikai 1.

T/C (O)

DMA ciklusok lezárása (terminal count).
Ha bármelyik DMA csatornán az átvitt byte-ok (szavak) száma eléri a DMA vezérlőbe beprogramozottat, ezen a vonalon egy pozitív impulzus jelenik meg. A jel aktív állapota logikai 1.

-SBHE (I/O)

A rendszer adatbuszának felső 8 vonala engedélyezett ezen jel aktív állapotában (system bus high enable).
A -SBHE jel az adatbusz felső byte-ján (SD8-SD15) végbemenő adatátvitelt jelzi. A 16 bites eszközök ezt a jelet használják az SD8-SD15 vonalakhoz kapcsolt buffereik engedélyezéséhez/tiltásához. Az SBHE vonal aktív állapotban logikai 0 értéket vesz fel.

-MASTER (I)

Ez a vonal az egyik DRQ vonallal együtt a rendszer-buszok vezérlésének az átvételére szolgál. Az I/O csatornára kapcsolt processzor vagy DMA vezérlő kiad egy DRQ kérést az alapkártya valamelyik DMA csatornájára, amit előzőleg kaszkád módba programoztak. A -DACK nyugtázás vétele után a processzor a -MASTER vonalat logikai 0-ba húzza (aktív állapot), majd átveszi a rendszer-buszok vezérlését (cím-, adat-, vezérlővonalak). A -MASTER jel hatására minden eszköznek a rendszer-buszhoz kapcsolódó 'felülete' nagyimpedanciás állapotba kerül (3-state). Miután a -MASTER vonal aktív állapotba került, a processzor 1 óraciklus eltelte után adhat információt az adat és címvonalakra, és további óraciklus eltelte után adhat ki írás vagy olvasás parancsot.
Ha ez a vonal 15 μsec-nál hosszabb ideig tartózkodik aktív állapotban, a frissítés hiánya miatt a rendszer RAM-ok elveszthetik tartalmukat.

-MEM CS16 (I)

Ez a jel informálja a rendszert arról, hogy az adatátvitel egy 16 bites, 1 várakozási állapottal bővített memória ciklust igényel (-memory 16-bit chip select). A jel az LA17-LA23 címbitek dekódolásával állítható elő. A -MEM CS16 vonal aktív állapota logikai 0, és egy olyan 'open' kollektoros vagy 'tri-state' meghajtó áramkör vezérelheti, amely legalább 20mA áram leadására képes (TTL0 szint esetén).

-I/O CS16 (I)

Ez a jel informálja a rendszert arról, hogy az adatátvitel egy 16 bites, 1 várakozási állapottal meghosszabbított I/O ciklust igényel (-I/O 16-bit chip select). A jelet a címdekóder állíthatja elő.
A -I/O CS16 vonal aktív állapota logikai 0, és egy olyan 'open' kollektoros vagy 'tri-state' meghajtó áramkör vezérelheti, amely legalább 20 mA áram leadására képes (TTL0 szint esetén).

OSC (O)

Az OSC oszcillátor jel egy nagysebességű, 70 nsec periódusidejű (14,31818MHz) órajel. Nincs szinkronban a rendszer órajelével, és 50%-os kitöltési tényezővel rendelkezik.

0WS (I)

Nulla várakozási állapot (zero wait state).
Ez a jel informálja a mikroprocesszort arról, hogy a folyamatban lévő busz-ciklust várakozási állapot beiktatása nélkül fejezheti be.
Ha egy 16 bites eszköz várakozási ciklus nélküli memória-műveletet akar megvalósítani, a címdekódernek az író/olvasó vezérlőjelekkel kapuzott kimenőjelét használhatja a 0WS vonal meghajtására.
Ha egy 8 bites eszköz minimálisan két várakozási ciklussal bővített memória-műveletet akar megvalósítani, az író/olvasó vezérlőjelek megjelenését követő 1 órajel periódus késleltetéssel kell aktivizálnia a címdekóder kimenetével kapuzott 0WS vonalat.
A 8 bites memória írás/olvasás utasítások a rendszer órajelének (CLK) 1 -- > 0 éléhez szinkronizáltak. A 0WS vonal aktív állapota logikai 0, és egy olyan 'open' kollektoros vagy 'tri-state' meghajtó áramkör vezérelheti, amely legalább 20mA áram leadását biztosítja (TTL0 szint esetén).

Összehasonlítva az XT gépek I/O csatornájának jeleit az AT gépek I/O csatorna jeleivel a következők állapíthatók meg:

- Az AT I/O csatorna J1-J8 csatlakozóinak jelei funkcióiban és elrendezését tekintve megegyeznek az XT I/O csatorna J1-J8 csatlakozóinak jeleivel.

- A csatlakozók egyezése lehetővé teszi, hogy az XT gépek adapter kártyáinak nagy része az AT gépekben is használhatók legyenek (pl: monitor vezérlők, RS232 adapterek, nyomtató vezérlők, stb.)

- A jelkiosztásban két olyan helyet találhatunk (B4 és B8), amely első rátekintésre különbözőnek tűnik az XT és az AT esetén. A B4 csatlakozási pont az XT gépeknél az IRQ2, AT gépeknél az IRQ9 megszakításkérő bemenet. Emlékeztetnénk azonban arra, hogy az IRQ9 bemenet interrupt vektora software úton az IRQ2 vektor címére van átirányítva.
  A másik eltérő csatlakozási pont a B8, amelyre az XT esetén a -CARD SLCTD, AT esetén pedig a 0WS jelet vezették. Az XT adapter kártyák zöme a -CARD SLCTD vonalat nem használja, így ezek a modulok mindkét géptípusra nézve kompatibilisek maradnak.

2.3.11 Az IBM AT alapkártya csatlakozói

A könnyebb áttekintés végett itt ismertetjük az AT alaplemezen található összes csatlakozó bekötését, függetlenül attól, hogy más fejezetekben néhányuk már bemutatásra került.

Az alaplemezen a következő csatlakozók találhatók:

- Két tápegység csatlakozó (PS8 és PS9)
- Hangszóró csatlakozó
- LED kijelző és a billentyűzetet tiltó kulcs kapcsolójának a csatlakozója
- CMOS telep csatlakozó
- Billentyűzet csatlakozó

A tápegység csatlakozók bekötését mutatja a következő felsorolás. A doboz hátsó oldalához a PS8 jelű csatlakozó esik közelebb. Ugyancsak a hátsó oldaltól kezdődik a csatlakozó pontok számozása 1-től 6-ig.
A tápegység csatlakozók bekötése:

Csatlakozó	Csatlakozópont száma	Tápfeszültségek
PS8	1	'Power Good' jel: a tápegység hibátlan működését jelzi.
	2	+5 Vdc
	3	+12 Vdc
	4	-12 Vdc
	5	GND
	6	GND

Csatlakozó	Csatlakozópont száma	Tápfeszültségek
PS9	1	GND
	2	GND
	3	-5 Vdc
	4	+5 Vdc
	5	+5 Vdc
	6	+5 Vdc

A hangszóró csatlakozó egy négypontos 'Berg strip' csatlakozó. A számozás a doboz előlapja felől kezdődik:

	Csatlakozási pont	Megnevezés
	1	Adat kimenet
*	2	-
	3	GND
	4	+5 Vdc

A bekapcsolást jelző LED és a billentyűzetet tiltó kulcs kapcsolójának a csatlakozója egy ötpontos 'Berg strip' csatlakozó. A számozás a doboz előlapja felől kezdődik:

	Csatlakozási pont	Megnevezés
	1	LED
*	2	-
	3	GND
	4	Keyboard tiltás
	5	GND

A CMOS telep csatlakozó egy négypontos 'Berg strip' csatlakozó. A számozás a doboz jobb oldalától kezdődik:

Csatlakozási pont	Megnevezés
1	GND
2	Nem használt
* 3	-
4	+6 Vdc

Megjegyzés: a Berg csatlakozók *-gal jelölt csatlakozási pontjait pozicionáló csappal töltötték ki, amely megakadályozza, hogy a csatlakozó ellendarab fordítva is bedugható legyen.

A billentyűzet csatlakozója egy ötpontos, nyomtatott áramköri lapra szerelhető DIN csatlakozó. A bekötés a következő:

Csatlakozási pont	Megnevezés
1	Keyboard órajel
2	Keyboard adatjel
3	Fenntartva
4	GND
5	+5 Vdc

2-21 táblázat.
Az IBM AT alapkártya csatlakozói.

3. RÉSZ

A tápegység.

3.1 Bemeneti specifikáció

3.2 Kimenetek

3.3 A 'Power Good' jel

3.4 A kimeneti csatlakozó jelkiosztása, terhelhetőségek

3.5 A tápegység elektromos kapcsolási vázlata

Az IBM PC tápegysége a gép dobozában helyezkedik el. Feladata a rendszer alaplemeze, a kártyák, a lemezegységek, a Winchester disk, a billentyűzet és az IBM monochrom monitor tápfeszültségének biztosítása.

3.1. Bemeneti specifikáció

A tápegység 50/60Hz +/-3Hz frekvenciájú, 110V (5A) vagy 220/240V (2,5A) hálózatra csatlakoztatható. A megfelelő hálózati feszültség a hálózati kábel csatlakozó mögött, a tápegység hátsó részén található kapcsolóval állítható be. Az alábbi táblázatban összefoglaljuk a tápegység bemeneti specifikációját:

névleges feszültség	fesz. szélső értékek	áram-felvétel
115V~	min. 100V~ max. 125V~	max. 5A
230V~	min. 200V~ max. 240V~	max. 3A

3.2. Kimenetek

A tápegység +5V, -5V, +12V és -12V-os egyenfeszültségeket szolgáltat, továbbá ellátja 115V ill. 230V tápfeszültséggel az IBM monochrom monitort. Az alábbi táblázat a tápegység kimeneteinek terhelhetőségét foglalja össze:

névleges DC kimenet	terhelő áram min.	max.	
+5V	7A	19,8A	-4%
-5V	0A	0,3A	-8%
+12V	2,5A	7,3A	-4%
-12V	0A	0,3A	-9%

3.2.1. Kimeneti védelem

Ha bármelyik kimenet terhelése a megengedett áram fölé emelkedik, a tápegység 20 msec-on belül lekapcsol. A túlterhelés következtében a tápegység nem károsodik.

3.2.2. Be- és kikapcsolási sorrend

Normális körülmények között a hálózati tápfeszültség be- ill. kikapcsolása esetén a tápegység kimenetei 300 msec alatt elérik a névleges feszültséget, ill. lekapcsolódnak. Nem károsodik a tápegység, ha a kimenetek bármelyikén nincs meg a minimálisan előírt terhelés, azonban a tápegység ez esetben kikapcsolódhat. A normális működéshez valamennyi kimeneten meg kell lenni a minimális terhelésnek.

3.3. A 'Power-Good' jel

A tápegység egy 'Power-Good' (tápfeszültség jó) jelet ad normális működés esetén. Ha a tápegységet legalább 1 másodpercre kikapcsoljuk, majd újra bekapcsoljuk, a bekapcsoláskor megjelenik a 'Power-Good' jel, jelezve hogy minden rendben. Ez a jel a DC kimeneti feszültség érzékelő, és az AC bemeneti feszültség érzékelő logikai ÉS kapcsolatából jön létre. TTL magas szintet ad normális működés esetén, és alacsonyat hibás körülmények között. Az AC bemeneti feszültség érzékelő az AC hiánya esetén a kimenetek minimális szintre történő csökkenése előtt legalább 1 msec-mal alacsony szintre állítja a 'Power-Good' jelet. A DC kimeneti feszültség érzékelő a hálózat bekapcsolásakor alacsony szintre állítja a 'Power-Good' jelet, és az összes kimeneti szint megjelenése után legalább 100, de legfeljebb 500 msec-ig lent tartja. A kimenetek alsó érzékelési szintjét mutatjuk be az alábbi táblázatban:

névleges DC feszültség	érzékelési szint
+5V	+4,5V
-5V	-3,75V
+12V	+10,8V
-12V	-10,4V

A 'Power-Good' jel 6 normál TTL bemenetet tud meghajtani.

3.4. A kimeneti csatlakozók jelkiosztása, terhelhetőségek

csatlakozó jelölés	feszültség	max.áram
PS8-1	Power Good	6 TTL bem.
PS8-2	+5V	3,8A
PS8-3	+12V	0,7A
PS8-4	-12V	0,3A
PS8-5	Föld	0,0A
PS8-6	Föld	0,0A
PS9-1	Föld	0,0A
PS9-2	Föld	0,0A
PS9-3	-5V	0,3A
PS9-4	+5V	3,8A
PS9-5	+5V	3,8A
PS9-6	+5V	3,8A
PS10-1	+12V	2,8A
PS10-2	Föld	0,0A
PS10-3	Föld	0,0A
PS10-4	+5V	1,8A
PS11-1	+12V	2,8A
PS11-2	Föld	0,0A
PS11-3	Föld	0,0A
PS11-4	+5V	1,8A
PS12-1	+12V	1,0A
PS12-2	Föld	0,0A
PS12-3	Föld	0,0A
PS12-4	+5V	0,6A

3-1 táblázat.
A PC tápegységek kimeneti csatlakozói.

3.5 A tápegység elektromos kapcsolási vázlata

Tápegységet több cég is forgalmaz különböző teljesítményekkel (150W XT, 200W AT) IBM személyi számítógépek részére. A különféle gyártók termékei felépítésüket tekintve nagyon hasonlóak, így a 3-1 ábrán bemutatott kivitel is rendelkezik mindazokkal a sajátságokkal, amelyek ezt a tápegység típust jellemzik. (A kapcsolási rajzhoz alkatrészlistát nem mellékelünk, mivel itt csak a felépítés ismertetése a cél.)

Valamennyi XT/AT tápegység kapcsoló üzemben működik, mégpedig oly módon hogy a hálózati primer feszültséget kapcsolgatják egy közelítőleg 25 kHz frekvenciájú oszcillátor által vezérelt pulzus-szélesség modulátor kimenőjelével. A terheléstől függően változik a modulátor-kimenőjel kitöltési tényezője, és ezt felhasználva hozzák létre a szabályozást.

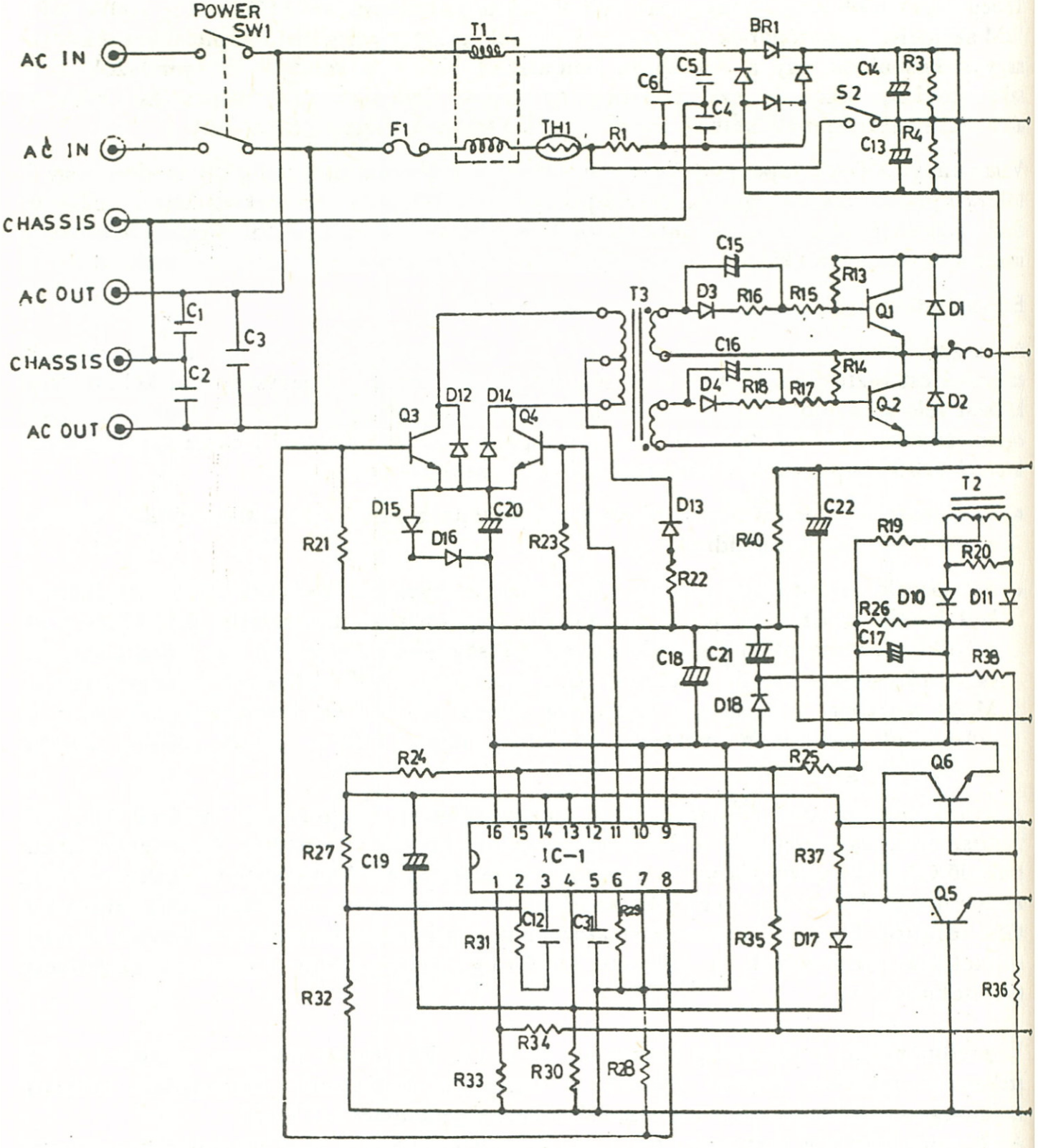
Ezekután tekintsük át a tápegység fontosabb részeit.

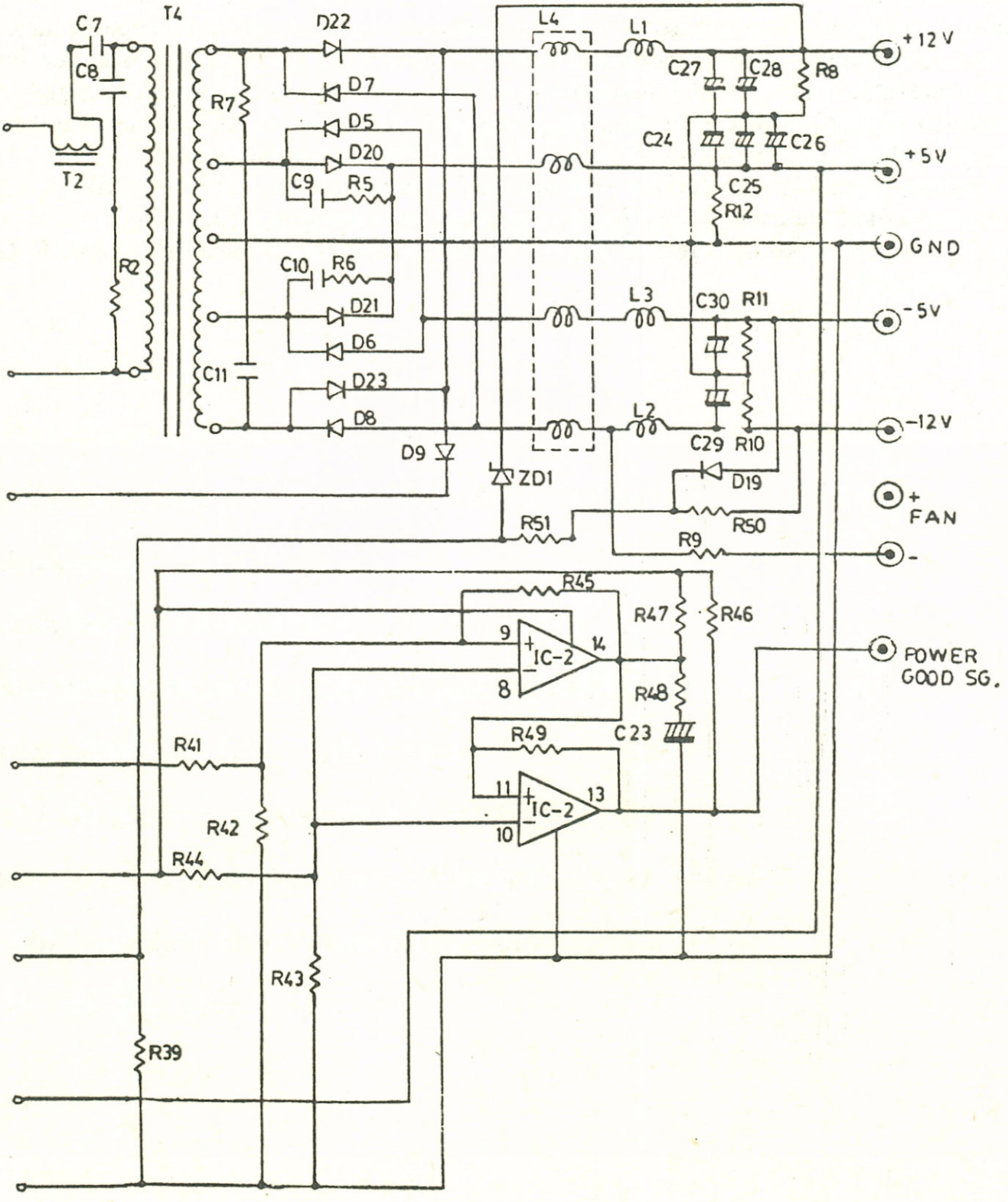
A T1-gyel jelölt fojtótekercsek és a C1, C2, C3 valamint C4, C5, C6 kondenzátorok zavarszűrést biztosítanak mind a hálózatról érkező, mind a tápegység által keltett és a hálózat felé irányuló zavaró-jelek ellen. A BR1 négy diódát tartalmazó tömb a hálózati egyenirányító, és az S2 kapcsoló állásától függően, hol kétutas, hol Graetz kapcsolásban üzemel. Az S2 kapcsolóval a bemeneti hálózati feszültség választható:

- ha a kapcsoló zárt, a bemeneti névleges feszültség 115V~ és a BR1 diódák kétutas egyenirányítóként működnek.
- A kapcsoló nyitott állása mellett, a bemeneti névleges feszültségnek 230V~ értékűnek kell lennie, a BR1 diódák ekkor Graetz kapcsolásban üzemelnek. A C13 és C14 szűrőkondenzátorokon kialakuló csúcsfeszültség ebben az esetben fele lenne a kétutas egyenirányítással kapott értéknek, ha a bemeneti váltófeszültségek megegyeznének. Mivel azonban a bemeneti feszültség ebben az esetben kétszeres (230 V~) a szűrőkondenzátorokon mérhető feszültség mind 115V~ mind 230V~ esetén közelítőleg egyforma.

A szűrőkondenzátorok feszültségeit a pulzus-szélesség modulátorral vezérelt Q1, Q2 tranzisztorok kapcsolgatják a T4 transzformátor primer tekercsére. Az ellenütemben működő Q1, Q2 tranzisztorokat a modulátor kimenetek a T3 leválasztó transzformátoron keresztül vezérlik. A T2, T3 és T4 transzformátorok választják el a hálózatot a tápegység többi részétől. Ezeknek a transzformátoroknak a primer és szekunder tekercsei közötti szigetelések megfelelnek a hálózati feszültség leválasztására vonatkozó szigetelési előírásoknak.

Végeredményben a tápegység bemenete, és a T2, T4 primer tekercse, valamint a T3 szekunder tekercse közötti elemek a tápegység bekapcsolt állapotában hálózati feszültségen találhatók, ezért szervizeléskor erre figyelemmel kell lenni.
A T4 transzformátor szekunder tekercsének megcsapolásaira négy kétutas egyenirányító csatlakozik és ezek állítják elő a +12V, +5V, -5V és -12V tápfeszültségeket.
Az IC1-gyel jelölt integrált áramkör az impulzus-szélesség modulátor. Típusszáma jelen esetben TL494. A modulátor IC két hibajel erősítőt, RC tagokkal beállítható oszcillátort,


POWER
SW1
AC IN
AC IN
CHASSIS
AC OUT
CHASSIS
AC OUT
T1
F1
TH1
R1
BR1
C5
C6
C4
C14
R3
S2
R4
C13
C15
R13
T3
D3
R16
R15
Q1
D1
C16
R14
D4
R18
R17
Q2
D2
C1
C2
C3
D12
D14
Q3
Q4
D15
C20
D16
R21
R23
D13
R22
R40
C22
T2
R19
R20
D10
D11
R26
C17
R38
C18
C21
D18
R24
R25
Q6
16 15 14 13 12 11 10 9
IC-1
1 2 3 4 5 6 7 8
R27
C19
R37
Q5
C12
C31
R29
R31
R35
D17
R32
R36
R34
R33
R30
R28



3-1 ábra.
A PC-k tápegységének kapcsolási vázlata.

+5V-os referencia feszültséget, holtidő vezérlő komparátort és pulzus-szélességet meghatározó elemeket tartalmaz.
Az első hibajel erősítő neminvertáló bemenetére (1) a +5V kimeneti tápfeszültség leosztott jele, az invertáló bemenetére (2) pedig a +5V referencia feszültség R27, R32 ellenállásokkal leosztott jele kerül. Az R32 ellenállás adjusztálásával (R32 helyett potenciométer is elképzelhető) a +5V-os kimeneti feszültség pontosan beállítható.
A második hibajel erősítő neminvertáló bemenete (16) földre van kötve. Az invertáló bemenetére (15) a +5V-os kimeneti feszültség és a bemeneti AC figyelő áramkör negatív kimeneti feszültségének az összegzett jele kerül. A bemeneti AC figyelő áramkört a T2 transzformátor és a D10, D11, C17 elemekből álló egyenirányító alkotja.
A pulzus-szélesség modulátor ellenütemű kimenetei (8, 11) vezérlik a Q1, Q2 tranzisztorokat kapcsoló transzformátoros meghajtófokozatot (Q3, Q4, T3).
Az IC2-vel jelölt komparátor a tápegység helyes működését mutató 'POWER GOOD' jelet állítja elő.

4. RÉSZ

A billentyűzet és az egér (mouse).

4.1 Bekapcsolás utáni funkciók

4.2 AT rendszer-parancsok (84 billentyűs kivitel)

4.3 AT billentyűzet-parancsok (84 billentyűs kivitel)

4.4 Adatátvitel

4.5 A billentyűzet csatlakozója

4.6 IBM billentyűzet kiosztás (84 billentyűs kivitel)

4.7 Scan kódok (84 billentyűs kivitel)

4.8 IBM 101/102 nyomógombos billentyűzet

4.9 IBM 101/102 nyomógombos billentyűzet scan kódjai

4.10 A billentyűzet elektromos kapcsolási vázlata

4.11 Az egér (mouse)

Az eredeti IBM billentyűzet 84 nyomógombos, kívülről csatlakoztatható egység. Megjegyezzük, hogy ettől eltérő számú (általában több) gombos billentyűzetek is forgalomban vannak. Külön AT és XT billentyűzeteket gyártanak, ezek egymással nem kompatibilisek. A különbségeket az alábbiakban részletezzük (A billentyűzetek nagy része olyan, hogy átkapcsolható XT vagy AT üzemmódra).

Az AT billentyűzet és a rendszer között kétirányú soros interface biztosítja az adatforgalmat. Az XT billentyűzet csak egy irányba képes adni. A billentyűzet egység érzékeli az összes billentyű leütést, és a megfelelő sorrendben továbbítja a scan kódokat az interface felé. A billentyűzet kiszolgálása alatti leütések elvesznek.

A billentyűleütések egy 16 karakteres FIFO pufferbe kerülnek arra az időre, amíg az interface fogadóképessé válik. Ha a puffer ürítése előtt érkezik a 17. karakter, akkor a puffernek az erre az esetre fenntarott 17. helyére a puffer túlcsordulás kódja (overrun code), AT esetén hex 00, XT esetén hex FF kerül. Ettől kezdve a puffer ürítésének megkezdése előtti billentyűleütések elvesznek. Amikor az interface újra fogadóképes, a billentyűzet a tárolt karaktereket a normális módon átküldi, és a további új adatok érvényesek lesznek.

Az összes billentyű 'make/break' módon működik. Ennek lényege, hogy ha egy billentyűt lenyomunk, annak 'make' (létrehozási) kódja bekerül a billentyűzet vezérlőbe. A make kód a billentyű scan kódja (4.7), melynek MSB-je 0. Amikor a billentyűt felengedjük, a 'break' (megszakítási) kódot küldi ki. Ez AT esetén két byte: a billentyű 'make' kódját megelőző hex F0, XT esetén a 'make' kód, azzal az eltéréssel, hogy az MSB 1. Az összes billentyű automatikus ismétlési (typematic) funkcióval rendelkezik. Ha egy gombot lenyomva tartunk, annak 'make' kódja kerül a vezérlőbe mindaddig, amíg fel nem engedjük. A kódok kiküldési gyakoriságát ismétlési sebességnek nevezzük. Ha egyszerre kettő vagy több gombot nyomunk le, csak az utolsóként lenyomott gomb kódja ismétlődik a fenti sebességgel. Ha az utolsó gombot felengedjük, az ismétlési funkció megszűnik, függetlenül attól, hogy más billentyű esetleg lenyomva marad. A puffer túlcsordulás megelőzése érdekében nem működik az ismétlési funkció, amíg az interface nem fogadóképes.

4.1 Bekapcsolás utáni funkciók

A billentyűzet logika egy tápfeszültség resetet (POR = Power-On Reset) generál, amikor a tápfeszültség rákapcsolódik a billentyűzet egységre. A POR legalább 300 msec, de legfeljebb 9 sec ideig tart. A POR után a billentyűzet végrehajtja az alapvető biztonsági tesztet (BAT = Basic Assurance Test). Ez a teszt ellenőrző összeget (checksum) képez és ellenőriz a billentyűzet ROM-ban, és teljes RAM tesztet végez az egység memóriájában. Az AT billentyűzeten a működési mód jelzők (3 LED a billentyűzet jobb felső részén) kigyulladnak, majd elalszanak. A BAT végrehajtási ideje 600-900 msec, mely hozzáadódik a POR idejéhez. A BAT végrehajtása után, amikor az interface fogadóképes (az óra és az adatvonal magas szintű), a billentyűzet kiküldi az interface felé az állapotkódot. Ez hibátlan teszt esetén hex AA, hiba esetén pedig hex FC.

4.2 AT rendszerparancsok (84 billentyűs kivitel)

Az alábbi parancsokat csak az AT rendszer küldheti AT billentyűzetnek. A nyugtázást 20 msec-on belül meg kell kapnia a billentyűzettől. A billentyűzet a kapott byte-okat az ACK kóddal nyugtázza.

RESET (alaphelyzetbe állítás, hex FF)

A rendszer ezt a parancsot akkor küldi ki, ha a billentyűzetet alapállapotba kell hozni. Ekkor a billentyűzet a reset programját, és az öntesztet hajtja végre, kiküldi az állapotjelző byte-ot, és alaphelyzetbe áll.

RESEND (újraküldés, hex FE)

A rendszer ezt a parancsot akkor küldi ki, ha a billentyűzet felől érkező átvitel során hibát észlel. A parancsot a billentyűzet nem nyugtázza, hanem újraküldi az utolsó adatot, kivéve, ha az RESEND parancs volt. Ez esetben az azt megelőző utolsó byte-ot küldi újra.

NO-OPERATION (NOP = üres művelet, hex F7-től FD-ig)

Ezek fenntartott kódok, a rendszer nem használja őket. Ha a billentyűzet mégis ilyen kódot kap, nyugtázza, majd folytatja a megszakított tevékenységét.

SET DEFAULT (alapérték beállítás, hex F6)

A nyugtázás után a billentyűzet minden állapotjelzőjét a bekapcsolás utáni helyzetbe állítja (kijelzők, puffer ürítés, ismétlési paraméterek), majd folytatja a gombok figyelését.

DEFAULT DISABLE (alapérték letiltás, hex F5)

Az előző paranccsal egyezik meg, azzal az eltéréssel, hogy utána leállítja a gombok figyelését.

ENABLE (engedélyezés, hex F4)

Nyugtázás után üríti a kimeneti puffert, és megkezdi a gombok figyelését.

SET TYPEMATIC RATE/DELAY (ismétlési sűrűség/késleltetés beállítás, hex F3)

A rendszer ezzel a paranccsal, majd az ezt követő paraméter byte-tal átállítja az ismétlési sűrűséget (RATE) és késleltetést (DELAY). A paraméter byte egyes bitjeinek jelentése a következő:

0. bit sűrűség érték LSB (RATE1)
1. bit sűrűség érték
2. bit sűrűség érték MSB

3. bit sűrűség érték LSB (RATE2)
4. bit sűrűség érték MSB
5. bit késleltetés LSB (DELAY)
6. bit késleltetés MSB
7. bit 0

Megjegyzés: RATE1 = a 0,1 és 2 bitek által előállított érték
RATE2 = a 3 és 4 bitek által előállított érték

A késleltetés idejét így számíthatjuk ki:

$TDELAY = (DELAY + 1) * 250$ [msec] +/- 20%

Az ismétlési sűrűség periódusidejét (két ismétlés közötti idő) így számíthatjuk ki:

$TRATE = (8 + RATE1) * 2^{RATE2} * 0.00417$ [sec]

Az ismétlési sűrűség (karakter/sec) = 1/TRATE. Ennek értékeit a paraméter byte 0-4. bitjeinek függvényében az alábbi táblázatban adjuk meg:

bitek	kar./sec	bitek	kar./sec
00000	30.0	10000	7.5
00001	26.7	10001	6.7
00010	24.0	10010	6.0
00011	21.8	10011	5.5
00100	20.0	10100	5.0
00101	18.5	10101	4.6
00110	17.1	10110	4.3
00111	16.0	10111	4.0
01000	15.0	11000	3.7
01001	13.3	11001	3.3
01010	12.0	11010	3.0
01011	10.9	11011	2.7
01100	10.0	11100	2.5
01101	9.2	11101	2.3
01110	8.6	11110	2.1
01111	8.0	11111	2.0

A paraméter byte érkezése után a billentyűzet a fentieknek megfelelően beállítja az ismétlési paramétereket, majd folytatja a gombok figyelését.

Az alapértékek (default):
RATE = 10 kar./sec
TDELAY = 500 msec +/- 20%

ECHO (visszhang, hex EE)

Diagnosztikai parancs. A billentyűzet válaszként ugyanezt a byte-ot küldi vissza, majd folytatja a gombok figyelését.

SET/RESET MODE INDICATORS (állapotkijelzők beállítása/törlése, hex ED)

Az AT billentyűzeten három kijelző van. Ezek állapotát akkor állítja át a billentyűzet, ha erre parancsot kap a rendszertől. A rendszer feladata, hogy az állapotjelzők állapotát nyilvántartsa. A parancs vételét a nyugtázás, majd egy paraméter byte vétele követi. A paraméter byte egyes bitjeinek jelentése a következő:

0. bit	SCROLL LOCK állapotjelző
1. bit	NUM LOCK állapotjelző
2. bit	CAPS LOCK állapotjelző
3-7. bit	nem használt

4.3 AT billentyűzet parancsok (84 billentyűs kivitel)

Az alábbi parancsokat az AT billentyűzetek küldhetik a rendszer felé.

RESEND (újraküldés, hex FE)

Hibás adatvétel esetén a billentyűzet ezzel a paranccsal újra kérheti a rendszertől az utolsó byte küldését.

ACK (nyugtázás, hex FA)

A billentyűzet ezzel a kóddal nyugtázza a rendszertől vett összes byte-ot, kivéve az ECHO és a RESEND parancsot.

OVERRUN (túlcsordulás, hex 00)

Ez a kód a billentyűzet puffer 17. helyére kerül túlcsordulás esetén. Ez akkor kerül kiküldésre, ha a puffer ürítése során annak tetejére kerül.

Diagnosztikai hiba (hex FD)

A billentyűzet időnként öntesztet hajt végre. Hiba esetén küldi ki ezt a kódot.

BREAK CODE PREFIX (gomb elengedés jelző, hex F0)

Egy gomb elengedését úgy jelzi a billentyűzet, hogy kiküldi ezt a kódot, majd a gomb scan kódját.

BAT hibátlan (hex AA)

A BAT hibátlan lefutása esetén ezt, hiba esetén hex F0-t küld ki a billentyűzet.

ECHO válasz (hex EE)

A rendszer ECHO parancsára válaszol ezzel a kóddal a billentyűzet.

4.4. Adatátvitel

A billentyűzet és a rendszer az óra és az adatvonalon át tartja egymással a kapcsolatot. Ezek a vonalak open kollektorosan vannak meghajtva, hogy mindkét oldalról lehúzhatóak legyenek. Amikor nincs kommunikáció, mindkét vonal magas állapotban van (logikai 1)

4.4.1. Az AT billentyűzet adatátvitele

A kétirányú kapcsolat során az adatok 11 bites sorozatként soros átvitellel kerülnek továbbításra az adatvonalon át. Lássuk az egyes bitek jelentését sorra:

1. bit 0, START bit
2. bit D0 adatbit (LSB)
3. bit D1 adatbit
4. bit D2 adatbit
5. bit D3 adatbit
6. bit D4 adatbit
7. bit D5 adatbit
8. bit D6 adatbit
9. bit D7 adatbit (MSB)
10.bit páratlan paritásbit
11.bit STOP bit

Ha a rendszer adatot kíván küldeni a billentyűzet felé, az adatvonalat lehúzza, az óravonalat 1-ben hagyja.

Az adatátvitel irányától függetlenül mindig a billentyűzet kapuzza a 11 bitet az óravonalon.

4.4.2. Az AT billentyűzet adat küldése (output)

Ha a billentyűzet készen áll az adat küldésére, először ellenőrzi, hogy nincs-e letiltva (inhibit állapot, ha az óra vonal 0-ban van), ill. hogy a rendszer nem akar-e adatot küldeni (request to send: óra = 1, adat = 0). E két esetben a küldésre szánt karaktert a pufferbe teszi, és utóbbi esetben átveszi az adatot.

Ha mind az adat, mind óra vonal magas, a billentyűzet kiküldi a 11 adatbitet az előző pontban leírt sorrendben. Az adat az órajel 0 állapotában érvényes, vagyis a lefutó és a felfutó él között. A billentyűzet az átvitel során 60 msec-onként ellenőrzi, hogy az óra vonal 1-ben van-e. Ha a rendszer az átvitel megkezdése után, de még a paritásbit küldése előtt lehúzza az órajelet, a billentyűzet leállítja az adat küldését, és elengedi az adat- és óravonalakat. Ha a paritásbit órajelének felfutó éle után érkezik a megszakítás, a billentyűzet végrehajtja az adat átküldését.

4.4.3. Az AT billentyűzet adat vétele (input)

Amikor a rendszer adatot akar küldeni a billentyűzet felé, ellenőrzi, hogy nincs-e éppen adásban. Ha adásban van, de még nem ment ki a tizedik órajel, a rendszer megszakíthatja az adást az órajel lehúzásával. A tizedik órajel után a rendszernek át kell vennie az adatot.

Ha minden rendben van, a rendszer 60 mikrosec-nál hosszabb időre lehúzza az óra vonalat, majd elengedi. A billentyűzet ellenőrzi, hogy a lehúzott állapot hosszabb volt-e 60 mikrosec-nál. Ha igen, ezt úgy értelmezi, hogy a rendszer adni akar (request to send), és kikapuzza a 11 órajelet az óravonalon. A tizedik bit után a billentyűzet lehúzza az adatvonalat, és még egyet kapuz (stop bit). A rendszer ebből tudja meg, hogy a billentyűzet átvette az adatot.

Minden rendszer által küldött adatot a billentyűzetnek 20 msec-on belül nyugtáznia kell. Ha a nyugta hibás, vagy elmarad, a rendszer az adatot újraküldi. Ebben az esetben nem kell kiküldeni a RESEND parancsot.

4.4.4. Az XT billentyűzet adatátvitele (output)

Ha a rendszer az óravonalat legalább 40 msec-ig lehúzza, az RESET parancsot jelent.

Normális működés esetén a billentyűzet 9 bites sorozatban küldi át soros módon adatait a rendszer felé. Az egyes bitek jelentése a következő:

1. bit 1, START bit
2. bit D0 adatbit (LSB)
3. bit D1 adatbit
4. bit D2 adatbit
5. bit D3 adatbit
6. bit D4 adatbit
7. bit D5 adatbit
8. bit D6 adatbit
9. bit D7 adatbit (MSB)

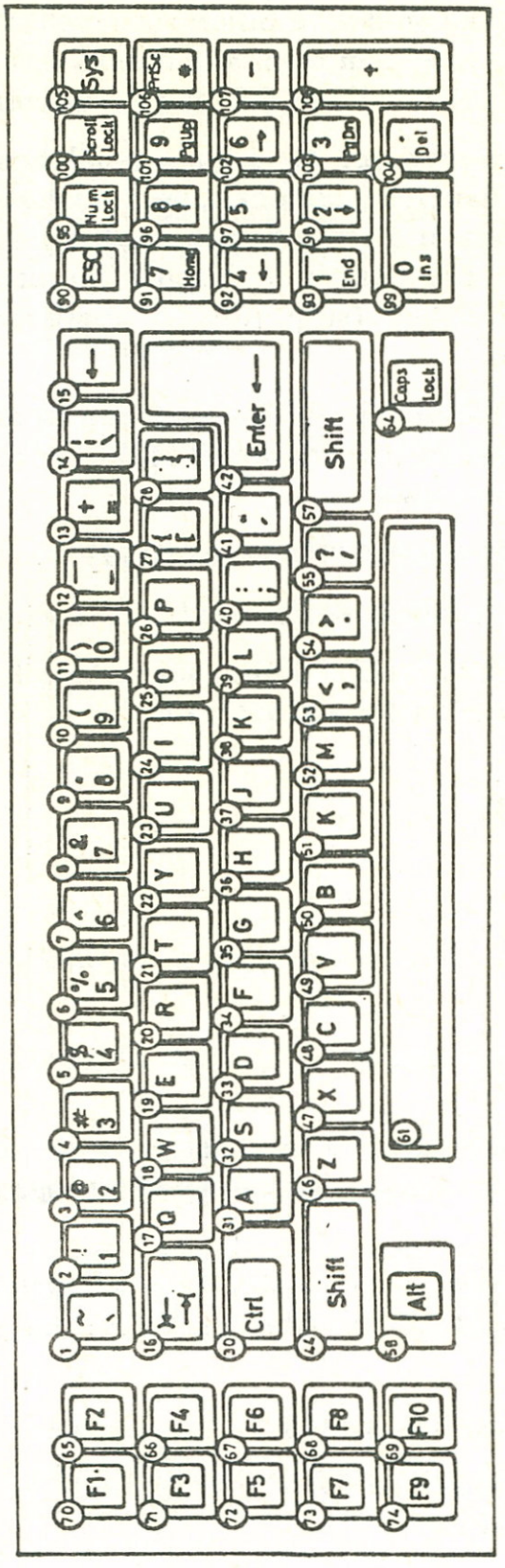
Az átküldés során a billentyűzet 9 órajelet generál az óravonalon az adatbitek kapuzásához. Az adat az órajel alacsony állapotában érvényes, vagyis a lefutó és a felfutó él között.

A billentyűzet az átküldés előtt megvizsgálja, hogy az adatvonal nincs-e lehúzva, mert a rendszer ezzel jelzi foglaltságát (inhibit állapot). Ez esetben a billentyűzet a pufferba teszi a küldendő adatot.

4.5. A billentyűzet csatlakozója

A billentyűzet szabványos ötpólusú DIN csatlakozóval kapcsolódik a rendszerhez. A csatlakozó kiosztása a következő:

1. óravonal
2. adatvonal
3. üres
4. föld
5. +5 Vdc



4-1 ábra.
U.S. English billentyűzet kiosztás és billentyűzet sorszámok.
(84 nyomógombos kivitel)

4.6. IBM billentyűzet kiosztás (84 billentyűs kivitel)

A billentyűzet kezelő a gombokat sorszám szerint azonosítja, és ez a gombok feliratától független. A 4-1 ábra a billentyűzet elrendezését és a sorszámokat mutatja az U.S. English billentyűzet esetén. Az XT billentyűzeten nem szerepel a SYS gomb.

4.7 Scan kódok (84 billentyűs kivitel)

A gombok sorszámához a billentyűzet egy-egy scan kódot rendel, és ezt küldi ki a rendszer felé. A scan kódok és a gombok összerendelése az AT és az XT billentyűzeten jelentősen eltér, ezért a következő táblázatban mindkettőt összefoglaljuk.

sorsz.	scan kód AT	scan kód XT	sorsz.	scan kód AT	scan kód XT
1	0E	29	46	1A	2C
2	16	02	47	22	2D
3	1E	03	48	21	2E
4	26	04	49	2A	2F
5	25	05	50	32	30
6	2E	06	51	31	31
7	36	07	52	3A	32
8	3D	08	53	3C	33
9	3E	09	54	49	34
10	46	0A	55	4A	35
11	45	0B	57	59	36
12	4E	0C	58	11	38
13	55	0D	61	29	39
14	5D	2B	64	58	3A
15	66	0E	65	06	3C
16	0D	0F	66	0C	3E
17	15	10	67	0B	40
18	1D	11	68	0A	42
19	24	12	69	09	44
20	2D	13	70	05	3B
21	2C	14	71	04	3D
22	35	15	72	03	3F
23	3C	16	73	83	41
24	43	17	74	01	43
25	44	18	90	76	01
26	4D	19	91	6C	47
27	54	1A	92	6B	4B
28	5B	1B	93	69	4F
30	14	1D	95	77	45
31	1C	1E	96	75	48
32	1B	1F	97	73	4C
33	23	20	98	72	50
34	2B	21	99	70	52
35	34	22	100	7E	46
36	33	23	101	7D	49
37	3B	24	102	74	4D
38	42	25	103	7A	51
39	4B	26	104	71	53
40	4C	27	105	84	--
41	52	28	106	7C	37
43	5A	1C	107	7B	4A
44	12	2A	108	79	4E

4-1 táblázat.
IBM XT és AT scan kódok.
(84 nyomógombos kivitel)

4.8 IBM 101/102 nyomógombos billentyűzet

A 84 nyomógombos billentyűzet mellett nagyon gyakori a 101/102 gombos változat. 101 nyomógombos az amerikai elrendezésű keyboard (U.S. English), a többi öt nemzetiségi kivitel pedig 102 nyomógombot tartalmaz (angol, francia, német, olasz, spanyol).

A rendszer és a billentyűzet közötti soros adatforgalom néhány eltéréstől eltekintve megegyezik a 84 gombos változatnál leírtakkal. Ezért ebben a részben csak az eltéréseket ismertetjük. A 101 nyomógombos keyboard kivitelét a 4-2 ábra, a 102 nyomógombosét pedig a 4-3 ábra mutatja. Röviden összefoglalható a leglényegesebb két eltérés a 84 gombos és a 101/102 keyboard-ok között:

- Kivitelét tekintve a 101/102 gombos keyboard egy jól elkülönített szövegszerkesztő mezővel bővült az irógépbillentyűzet és a numerikus 'keypad' között. Ezek a szövegszerkesztő funkciókat végző billentyűk a 84 gombos keyboard esetén (és a 101/102-es kivitelnél is) alternatív módon választhatók a 'keypad'-en belül, ami azt jelenti, hogy a 'keypad'-et vagy számbevitelre, vagy szövegszerkesztésre használhatjuk. Ezenkívül megváltozott a függvénygombok elhelyezése is, és számuk 10-ről 12-re növekedett.
- A 101/102 gombos keyboard-ok 3 féle típusú scan kódok előállítására adnak lehetőséget. A scan kódok típusa az AT által küldött rendszerparancsokkal állítható be. A scan kódok kiválasztását és a scan kód táblákat a későbbiekben ismertetjük.

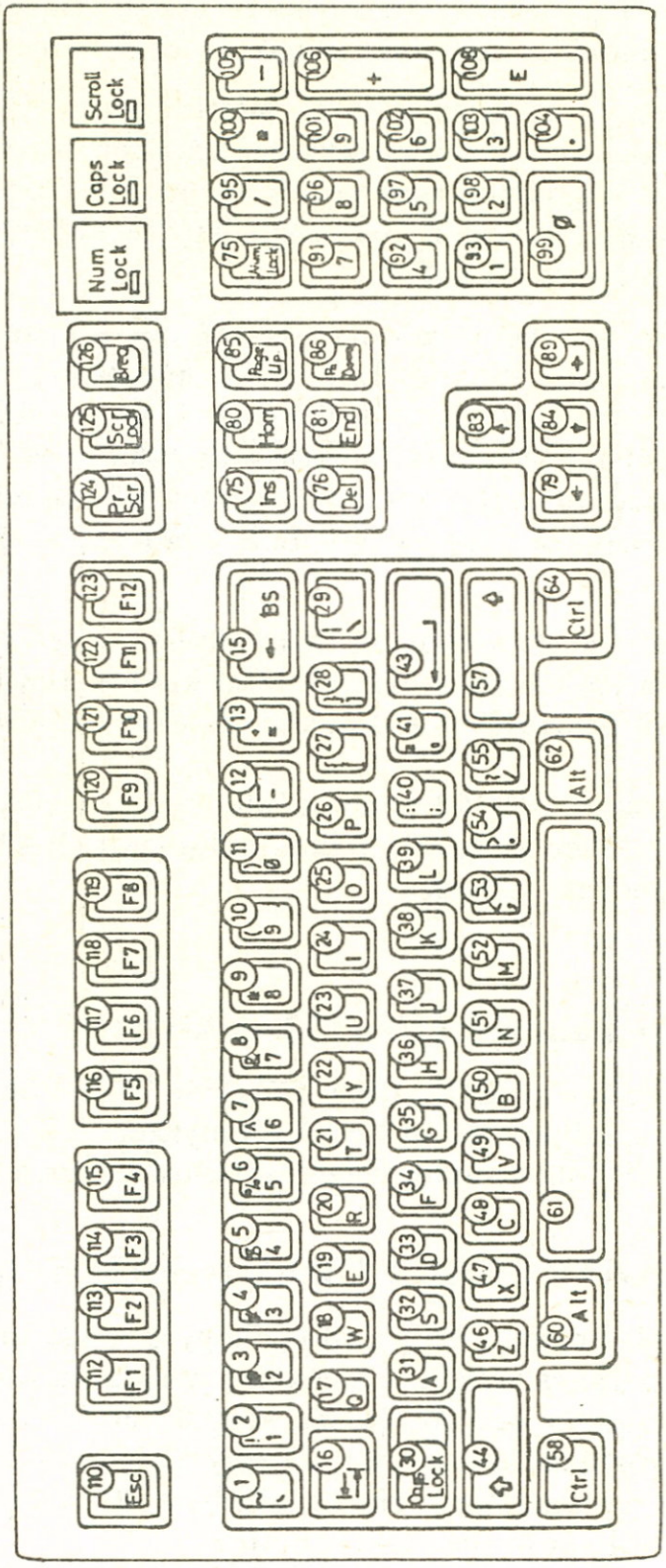
A gép bekapcsolása után a 101/102 gombos billentyűzet tápfeszültség reset jele legalább 150 msec, de legfeljebb 2 sec ideig tart, a biztonsági teszt (BAT = Basic Assurance Test) pedig 300-500 msec időt vesz igénybe.

4.8.1 AT rendszerparancsok (101/102-es kivitel)

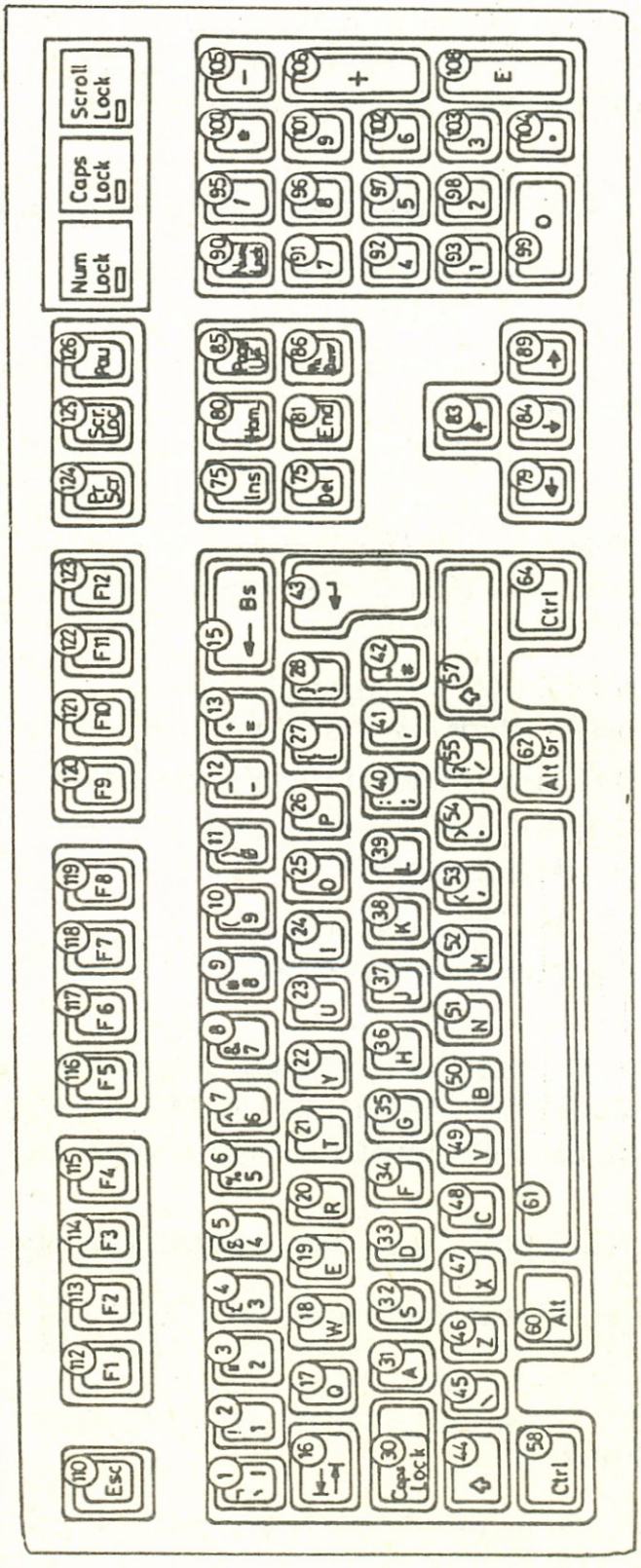
84 gombos keyboard esetén az EDh, EEh, F3h, F4h, F5h, F6h, FEh, FFh kódokkal jelölt rendszerparancsokat küldheti az AT a billentyűzet felé (ld. 4.2 fejezet). Ezek a rendszerparancsok a 101/102 nyomógombos keyboard esetén a következőkkel bővülnek:

SELECT ALTERNATE SCAN CODES (scan kód típusválasztás, hex F0)

Ez a parancs arról informálja a keyboard-ot, hogy a rendszer szeretné megváltoztatni a scan kód típusát. A parancs vételét a keyboard nyugtázza (ACK), és kiüríti a kimeneti pufferét. Ezután a rendszer egy 00, 01, 02, vagy 03 értékű adatbyte-ot küld a keyboard részére, amit újabb nyugtázás követ. A 01, 02 vagy 03 értékek hatására az 1, 2 vagy 3 típusú scan kódok kerülnek kiválasztásra. Ha az adatbyte 00 értékű, a keyboard informálja a rendszert az éppen kiválasztott scan kód típusáról.
Az új scan kód típus kiválasztása után a keyboard folytatja a billentyűk lenyomásának figyelését.



4-2 ábra.
101 nyomógombos U.S. English billentyűzet kiosztás és billentyűzet sorszámok.



4-3 ábra.
102 nyomógombos U.K. English billentyűzet
kiosztás és billentyűzet sorszámok.

READ ID (azonosítási információ olvasása, hex F2)

A rendszer ezt a parancsot kiadva azonosítási információt kér a keyboard-tól. A keyboard nyugtázással (ACK) válaszol, majd a billentyűk figyelését megszakítva 2 byte-os azonosító kódot küld az AT felé. A 2 byte között eltelt idő nem lehet 0.5 msec-nál több. A 2 byte kibocsátása után a keyboard folytatja a billentyűk lekérdezését.

SET ALL KEYS (típusbeállítás az összes billentyűre, hex F7, F8, F9, FA)

Ezen parancsok hatására a keyboard valamennyi billentyűjének a működését a következők szerint állítja be:

Parancs hex érték	Funkció
F7	Valamely billentyű lenyomásakor ismétléses (Typematic) mód kerül megvalósításra
F8	A billentyűk lenyomásakor és visszaengedésekor is scan kódot generál a keyboard (Make/Break)
F9	Csak lenyomáskor keletkezik scan kód (Make)
FA	Az F7 és F8 parancs együtt (Typematic/Make/Break)

A parancs kiadását a keyboard nyugtázza (ACK), kiüríti a kimeneti puffert, beállítja a billentyűk működését a parancsnak megfelelően, majd folytatja a billentyűk állapotának figyelését. Ezek a parancsok csak a 3. típusú scan kódok választása esetén hatásosak.

SET KEY TYPE (típusbeállítás egy adott billentyűre, hex FB, FC, FD)

Ezen parancsok hatására a keyboard egy kiválasztott billentyű működését a következők szerint állítja be:

Parancs hex érték	Funkció
FB	A billentyű lenyomásakor ismétléses (Typematic) mód kerül megvalósításra

Parancs hex érték	Funkció
FC	A billentyű lenyomásakor és visszaengedésekor is scan kódot generál a keyboard (Make/Break)
FD	Csak a billentyű lenyomásakor keletkezik scan kód (Make)

A parancs kiadását a keyboard nyugtázza (ACK), kiüríti a kimeneti puffert, és várja a rendszertől a billentyű-kijelölő információt. A billentyű azonosítása a 3. típusú scan kód készlet szerinti scan kód vételével történik. Ezt a kódot kell a rendszernek kiadnia a keyboard részére a billentyűk kijelöléséhez. Az így felismert billentyűt a keyboard a parancs szerinti üzemmódba állítja.

4.8.2 AT billentyűzet parancsok (101/102-es kivitel)

A 101/102 nyomógombos keyboard a hex 00, 83AB, AA, EE, FA, FC, FE és FF kódú parancsokat küldheti a rendszer felé. A hex AA, EE, FA és FE parancsok megegyeznek a 84 nyomógombos keyboard-nál leírtakkal. A további parancsok:

KEY DETECTION ERROR (billentyű-lenyomás detektálási hiba, hex 00 vagy FF)

Ha a keyboard olyan állapotba kerül, hogy képtelen detektálni valamely leütött billentyűt, hex 00 vagy FF kódú parancsot küld a rendszer felé. 1. típusú scan kódok esetén a 00, 2. és 3. típusú scan kódok esetén a hex FF parancs kerül kiadásra.

KEYBOARD ID (keyboard azonosítás, hex 83AB)

A keyboard azonosító a 2 byte-os hex 83AB. A rendszer által kiadott READ ID (azonosító lekérdezése) parancsra a keyboard nyugtázással válaszol (ACK), megszakítja a billentyűk figyelését, majd kibocsátja a 2 byte-os azonosítót, először az alacsonyabb, majd a magasabb helyértékűt. Ezt követően folytatja a billentyűk lekérdezését.

BAT FAILURE CODE (biztonsági teszt hiba, hex FC)

Ha az alapvető biztonsági teszt (BAT) hibát észlel, a keyboard a hex FC kódot küldi a rendszer felé, megszakítja a billentyűk figyelését, és a rendszer válaszára, vagy általános törlésre (reset) vár.

4.9 IBM 101/102 nyomógombos billentyűzet scan kódjai

A következő táblázatok a 101/102 nyomógombos keyboard-ok billentyű sorszámait és a hozzájuk tartozó scan kódokat tartalmazzák, mind a 3 scan kód készlet esetén. A 'default' scan kód típus a 2.

1. típusú scan kódok

Az 1. típusú kódkészletben az egyes billentyűkhöz egy-egy alap scan kódot rendeltek, és van néhány olyan billentyű, amely a Shift, Ctrl, Alt, Num Lock gombok állapotától függően különböző scan kódokat állít elő. Valamely billentyű hosszabb idejű lenyomásakor az ismételten kiadott kódok megegyeznek az alap scan kóddal. A 4-2.1 és 4-2.2 táblázatok tartalmazzák az 1. típusú scan kód készletet.
A 4-2.1 táblázat azokat a scan kódokat sorolja fel, amelyek nem függenek a Shift, Ctrl, Alt, Num Lock billentyűk állapotától, míg a 4-2.2 táblázat az ezektől függő scan kódokat ismerteti. A 4-2.2 táblázatban szereplő billentyűk alap scan kódjai azonosak más billentyűk alap scan kódjaival, ezért egy hex E0 kiegészítő kóddal bővültek a 4-2.2 táblázatban felsorolt scan kódok.

A billentyű száma	Scan kód lenyomáskor	Scan kód felengedéskor
1	29	A9
2	02	82
3	03	83
4	04	84
5	05	85
6	06	86
7	07	87
8	08	88
9	09	89
10	0A	8A
11	0B	8B
12	0C	8C
13	0D	8D
15	0E	8E
16	0F	8F
17	10	90
18	11	91
19	12	92
20	13	93
21	14	94
22	15	95

A billentyű száma	Scan kód lenyomáskor	Scan kód felengedéskor
23	16	96
24	17	97
25	18	98
26	19	99
27	1A	9A
28	1B	9B
29 *	2B	AB
30	3A	BA
31	1E	9E
32	1F	9F
33	20	A0
34	21	A1
35	22	A2
36	23	A3
37	24	A4
38	25	A5
39	26	A6
40	27	A7
41	28	A8
42**	2B	AB
43	1C	9C
44	2A	AA
45**	56	D6
46	2C	AC
47	2D	AD
48	2E	AE
49	2F	AF
50	30	B0
51	31	B1
52	32	B2
53	33	B3
54	34	B4
55	35	B5
57	36	B6
58	1D	9D
60	38	B8
61	39	B9
62	E0 38	E0 B8
64	E0 1D	E0 9D
90	45	C5

A billentyű száma	Scan kód lenyomáskor	Scan kód felengedéskor
91	47	C7
92	4B	CB
93	4F	CF
96	48	C8
97	4C	CC
98	50	D0
99	52	D2
100	37	B7
101	49	C9
102	4D	CD
103	51	D1
104	53	D3
105	4A	CA
106	4E	CE
108	E0 1C	E0 9C
110	01	81
112	3B	BB
113	3C	BC
114	3D	BD
115	3E	BE
116	3F	BF
117	40	C0
118	41	C1
119	42	C2
120	43	C3
121	44	C4
122	57	D7
123	58	D8
125	46	C6

* - csak 101 gombos billentyűzet esetén
** - csak 102 gombos billentyűzet esetén

4-2.1 táblázat.
101/102 gombos billentyűzet
1. típusú scan kódkészlete (I.)

A billentyű száma	Alapeset, vagy Shift + Num Lock lenyomás /felengedés	Shift lenyomás /felengedés *	Num Lock lenyomás /felengedés
75	E0 52 /E0 D2	E0 AA E0 52 /E0 D2 E0 2A	E0 2A E0 52 /E0 D2 E0 AA
76	E0 53 /E0 D3	E0 AA E0 53 /E0 D3 E0 2A	E0 2A E0 53 /E0 D3 E0 AA
79	E0 4B /E0 CB	E0 AA E0 4B /E0 CB E0 2A	E0 2A E0 4B /E0 CB E0 AA
80	E0 47 /E0 C7	E0 AA E0 47 /E0 C7 E0 2A	E0 2A E0 47 /E0 C7 E0 AA
81	E0 4F /E0 CF	E0 AA E0 4F /E0 CF E0 2A	E0 2A E0 4F /E0 CF E0 AA
83	E0 48 /E0 C8	E0 AA E0 48 /E0 C8 E0 2A	E0 2A E0 48 /E0 C8 E0 AA
84	E0 50 /E0 D0	E0 AA E0 50 /E0 D0 E0 2A	E0 2A E0 50 /E0 D0 E0 AA
85	E0 49 /E0 C9	E0 AA E0 49 /E0 C9 E0 2A	E0 2A E0 49 /E0 C9 E0 AA
86	E0 51 /E0 D1	E0 AA E0 51 /E0 D1 E0 2A	E0 2A E0 51 /E0 D1 E0 AA
89	E0 4D /E0 CD	E0 AA E0 4D /E0 CD E0 2A	E0 2A E0 4D /E0 CD E0 AA

A billentyű száma	Alapeset lenyomás/felengedés	Shift lenyomás/felengedés *
95	E0 35 / E0 B5	E0 AA E0 35 / E0 B5 E0 2A

* - Ha a bal Shift billentyű lenyomott állapotban van, az AA / 2A shift 'make/break' kódok is kiadásra kerülnek az aktuális kóddal. Ha a jobboldali Shift billentyű van lenyomva, a B6 / 36 kódok egészítik ki a scan kódot. Ha mindkét Shift billentyű lenyomott állapotban található, mind a két Shift gomb kódja kiadásra kerül az aktuális kóddal.

A billentyű száma	Alapeset lenyomás /felengedés	Ctrl, Shift lenyomás /felengedés	Alt lenyomás /felengedés
124	E0 2A E0 37 /E0 B7 E0 AA	E0 37 /E0 B7	54 /D4

A billentyű száma	Alapeset lenyomás/felengedés	Ctrl lenyomás/felengedés *
126 **	E1 1D 45 E1 9D C5 / ----------	E0 46 E0 C6 / ----------

** - Ez a billentyű nem működik ismétléses módban. Scan kódok csak a billentyű lenyomásakor kerülnek kiadásra.

4-2.2 táblázat.
101/102 gombos billentyűzet
1. típusú scan kódkészlete (II.).

2. típusú scan kódok

A 2. típusú kódkészletben az egyes billentyűkhöz egyedi, 8 bites scan kódok tartoznak, a billentyűk lenyomásakor. A billentyűk felengedésekor a scan kódok 2 byte-ot tartalmaznak: az első közülük a hex F0 prefix, amely a felengedést jelzi, a második byte megegyezik a lenyomáskor generált scan kóddal.

A 4-3.1 és 4-3.2 táblázatok mutatják be a második típusú scan kódkészletet. A 4-3.1 táblázat azokat a scan kódokat sorolja fel, amelyek nem függenek a Shift, Ctrl, Alt, Num Lock billentyűk állapotától, míg a 4-3.2 táblázat az ezektől függő scan kódokat ismerteti. A 4-3.2 táblázatban szereplő billentyűk alap scan kódjai azonosak más scan billentyűk alap scan kódjaival, ezért egy hex E0 kiegészítő kóddal bővültek a 4-3.2 táblázatban felsorolt scan kódok.

A billentyű száma	Scan kód lenyomáskor	Scan kód felengedéskor
1	0E	F0 0E
2	16	F0 16
3	1E	F0 1E
4	26	F0 26
5	25	F0 25
6	2E	F0 2E
7	36	F0 36
8	3D	F0 3D
9	3E	F0 3E
10	46	F0 46
11	45	F0 45
12	4E	F0 4E
13	55	F0 55
15	66	F0 66
16	0D	F0 0D
17	15	F0 15
18	1D	F0 1D
19	24	F0 24
20	2D	F0 2D
21	2C	F0 2C
22	35	F0 35
23	3C	F0 3C
24	43	F0 43
25	44	F0 44
26	4D	F0 4D
27	54	F0 54
28	5B	F0 5B
29 *	5D	F0 5D
30	58	F0 58
31	1C	F0 1C
32	1B	F0 1B
33	23	F0 23
34	2B	F0 2B
35	34	F0 34
36	33	F0 33
37	3B	F0 3B
38	42	F0 42
39	4B	F0 4B
40	4C	F0 4C
41	52	F0 52

A billentyű száma	Scan kód lenyomáskor	Scan kód felengedéskor
42 **	5D	F0 5D
43	5A	F0 5A
44	12	F0 12
45 **	61	F0 61
46	1A	F0 1A
47	22	F0 22
48	21	F0 21
49	2A	F0 2A
50	32	F0 32
51	31	F0 31
52	3A	F0 3A
53	41	F0 41
54	49	F0 49
55	4A	F0 4A
57	59	F0 59
58	14	F0 14
60	11	F0 11
61	29	F0 29
62	E0 11	E0 F0 11
64	E0 14	E0 F0 14
90	77	F0 77
91	6C	F0 6C
92	6B	F0 6B
93	69	F0 69
96	75	F0 75
97	73	F0 73
98	72	F0 72
99	70	F0 70
100	7C	F0 7C
101	7D	F0 7D
102	74	F0 74
103	7A	F0 7A
104	71	F0 71
105	7B	F0 7B
106	79	F0 79
108	E0 5A	E0 F0 5A
110	76	F0 76
112	05	F0 05
113	06	F0 06
114	04	F0 04

A billentyű száma	Scan kód lenyomáskor	Scan kód felengedéskor
115	0C	F0 0C
116	03	F0 03
117	0B	F0 0B
118	83	F0 83
119	0A	F0 0A
120	01	F0 01
121	09	F0 09
122	78	F0 78
123	07	F0 07
125	7E	F0 7E

* - csak 101 gombos billentyűzet esetén
** - csak 102 gombos billentyűzet esetén

4-3.1 táblázat.
101/102 gombos billentyűzet
2. típusú scan kódkészlete (I.)

A billentyű száma	Alapeset, vagy Shift + Num Lock lenyomás /felengedés	Shift lenyomás /felengedés *	Num Lock lenyomás /felengedés
75	E0 70 /E0 F0 70	E0 F0 12 E0 70 /E0 F0 70 E0 12	E0 12 E0 70 /E0 F0 70 E0 F0 12
76	E0 71 /E0 F0 71	E0 F0 12 E0 71 /E0 F0 71 E0 12	E0 12 E0 71 /E0 F0 71 E0 F0 12
79	E0 6B /E0 F0 6B	E0 F0 12 E0 6B /E0 F0 6B E0 12	E0 12 E0 6B /E0 F0 6B E0 F0 12
80	E0 6C /E0 F0 6C	E0 F0 12 E0 6C /E0 F0 6C E0 12	E0 12 E0 6C /E0 F0 6C E0 F0 12
81	E0 69 /E0 F0 69	E0 F0 12 E0 69 /E0 F0 69 E0 12	E0 12 E0 69 /E0 F0 69 E0 F0 12
83	E0 75 /E0 F0 75	E0 F0 12 E0 75 /E0 F0 75 E0 12	E0 12 E0 75 /E0 F0 75 E0 F0 12

A billentyű száma	Alapeset, vagy Shift + Num Lock lenyomás /felengedés	Shift lenyomás /felengedés *	Num Lock lenyomás /felengedés
84	E0 72 /E0 F0 72	E0 F0 12 E0 72 /E0 F0 72 E0 12	E0 12 E0 72 /E0 F0 72 E0 F0 12
85	E0 7D /E0 F0 7D	E0 F0 12 E0 7D /E0 F0 7D E0 12	E0 12 E0 7D /E0 F0 7D E0 F0 12
86	E0 7A /E0 F0 7A	E0 F0 12 E0 7A /E0 F0 7A E0 12	E0 12 E0 7A /E0 F0 7A E0 F0 12
89	E0 74 /E0 F0 74	E0 F0 12 E0 74 /E0 F0 74 E0 12	E0 12 E0 74 /E0 F0 74 0E F0 12

A billentyű száma	Alapeset lenyomás/felengedés	Shift lenyomás/felengedés *
95	E0 4A / E0 F0 4A	E0 F0 12 4A / E0 12 F0 4A

* - Ha a bal Shift billentyű lenyomott állapotban van, az F0 12/12 shift 'make/break' kódok is kiadásra kerülnek az aktuális kóddal. Ha a jobboldali Shift billentyű van lenyomva, a F0 59/59 kódok egészítik ki a scan kódot. Ha mindkét Shift billentyű lenyomott állapotban található, mind a két Shift gomb kódja kiadásra kerül az aktuális kóddal.

A billentyű száma	Alapeset lenyomás /felengedés	Ctrl, Shift lenyomás /felengedés	Alt lenyomás /felengedés
124	E0 12 E0 7C /E0 F0 7C E0 F0 12	E0 7C /E0 F0 7C	84 /F0 84

A billentyű száma	Alapeset lenyomás/felengedés	Ctrl lenyomás/felengedés *
126 **	E1 14 77 E1 F0 14 F0 77 / ----------	E0 7E E0 F0 7E / ----------

** - Ez a billentyű nem működik ismétléses módban. Scan kódok csak a billentyű lenyomásakor kerülnek kiadásra.

4-3.2 táblázat.
101/102 gombos billentyűzet
2. típusú scan kódkészlete (II.).

3. típusú scan kódok

A 3. típusú kódkészletben az egyes billentyűkhöz egyedi, 8 bites scan kódok tartoznak a billentyűk lenyomásakor. A billentyűk felengedésekor a scan kódok 2 byte-ot tartalmaznak: az első közülük a hex F0 prefix, amely a felengedést jelzi, a második byte megegyezik a lenyomáskor generált scan kóddal. Valamely billentyű hosszabb idejű lenyomásakor az ismételten kiadott kódok megegyeznek a lenyomás scan kódjával.
A 3. típusú kódkészletben minden egyes billentyűhöz csak egy scan kód tartozik, és nincsenek olyan billentyűk, amelyeknek a scan kódjait más billentyűk állapota befolyásolná. A 4-4 táblázat sorolja fel a 3. típusú scan kódokat.

A billentyű száma	Scan kód lenyomáskor	Scan kód felengedéskor	A billentyű 'default' módja
1	0E	F0 0E	Typematic
2	16	F0 16	Typematic
3	1E	F0 1E	Typematic
4	26	F0 26	Typematic
5	25	F0 25	Typematic
6	2E	F0 2E	Typematic
7	36	F0 36	Typematic
8	3D	F0 3D	Typematic
9	3E	F0 3E	Typematic

A billentyű száma	Scan kód lenyomáskor	Scan kód felengedéskor	A billentyű 'default' módja
10	46	F0 46	Typematic
11	45	F0 45	Typematic
12	4E	F0 4E	Typematic
13	55	F0 55	Typematic
15	66	F0 66	Typematic
16	0D	F0 0D	Typematic
17	15	F0 15	Typematic
18	1D	F0 1D	Typematic
19	24	F0 24	Typematic
20	2D	F0 2D	Typematic
21	2C	F0 2C	Typematic
22	35	F0 35	Typematic
23	3C	F0 3C	Typematic
24	43	F0 43	Typematic
25	44	F0 44	Typematic
26	4D	F0 4D	Typematic
27	54	F0 54	Typematic
28	5B	F0 5B	Typematic
29 *	5C	F0 5C	Typematic
30	14	F0 14	Make/Break
31	1C	F0 1C	Typematic
32	1B	F0 1B	Typematic
33	23	F0 23	Typematic
34	2B	F0 2B	Typematic
35	34	F0 34	Typematic
36	33	F0 33	Typematic
37	3B	F0 3B	Typematic
38	42	F0 42	Typematic
39	4B	F0 4B	Typematic
40	4C	F0 4C	Typematic
41	52	F0 52	Typematic
42 **	53	F0 53	Typematic
43	5A	F0 5A	Typematic
44	12	F0 12	Make/Break
45 **	13	F0 13	Typematic
46	1A	F0 1A	Typematic
47	22	F0 22	Typematic
48	21	F0 21	Typematic
49	2A	F0 2A	Typematic
50	32	F0 32	Typematic

A billentyű száma	Scan kód lenyomáskor	Scan kód felengedéskor	A billentyű 'default' módja
51	31	F0 31	Typematic
52	3A	F0 3A	Typematic
53	41	F0 41	Typematic
54	49	F0 49	Typematic
55	4A	F0 4A	Typematic
57	59	F0 59	Make/Break
58	11	F0 11	Make/Break
60	19	F0 19	Make/Break
61	29	F0 29	Typematic
62	39	F0 39	Make only
64	58	F0 58	Make only
75	67	F0 67	Make only
76	64	F0 64	Typematic
79	61	F0 61	Typematic
80	6E	F0 6E	Make only
81	65	F0 65	Make only
83	63	F0 63	Typematic
84	60	F0 60	Typematic
85	6F	F0 6F	Make only
86	6D	F0 6D	Make only
89	6A	F0 6A	Typematic
90	76	F0 76	Make only
91	6C	F0 6C	Make only
92	6B	F0 6B	Make only
93	69	F0 69	Make only
95	77	F0 77	Make only
96	75	F0 75	Make only
97	73	F0 73	Make only
98	72	F0 72	Make only
99	70	F0 70	Make only
100	7E	F0 7E	Make only
101	7D	F0 7D	Make only
102	74	F0 74	Make only
103	7A	F0 7A	Make only
104	71	F0 71	Make only
105	84	F0 84	Make only
106	7C	F0 7C	Typematic
108	79	F0 79	Make only
110	08	F0 08	Make only
112	07	F0 07	Make only

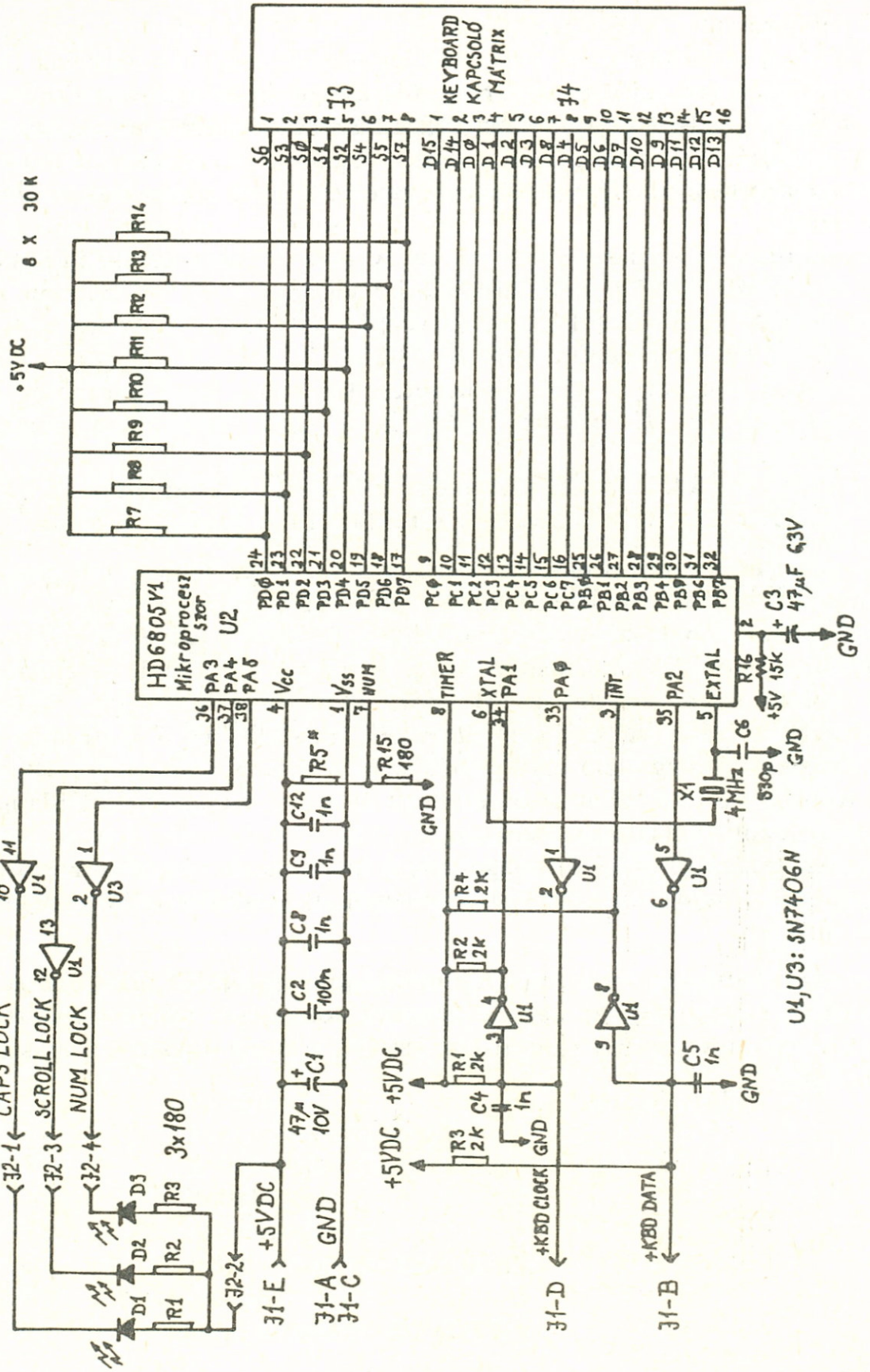
A billentyű száma	Scan kód lenyomáskor	Scan kód felengedéskor	A billentyű 'default' módja
113	0F	F0 0F	Make only
114	17	F0 17	Make only
115	1F	F0 1F	Make only
116	27	F0 27	Make only
117	2F	F0 2F	Make only
118	37	F0 37	Make only
119	3F	F0 3F	Make only
120	47	F0 47	Make only
121	4F	F0 4F	Make only
122	56	F0 56	Make only
123	5E	F0 5E	Make only
124	57	F0 57	Make only
125	5F	F0 5F	Make only
126	62	F0 62	Make only

* - csak 101 gombos keyboard esetén
** - csak 102 gombos keyboard esetén

4-4 táblázat.
101/102 gombos billentyűzet
3. típusú scan kódkészlete.

4.10 A billentyűzet elektromos kapcsolási vázlata

Az IBM 101/102 nyomógombos billentyűzet kapcsolási rajza a 4-4 ábrán látható. A kapcsolás csak néhány elemet tartalmaz, ezek leglényegesebb része a RAM és ROM memóriákat is magába foglaló HD6805V1 típusú mikroprocesszor. A mikroprocesszor két interface felülettel rendelkezik: az egyikkel az AT géphez csatlakozik, a másikkal pedig a keyboard mátrix elrendezésű kapcsolóival és kijelzőivel áll kapcsolatban.
Az órajelet a mikroprocesszor belső óragenerátora állítja elő a kívülről hozzákapcsolt 4 MHz frekvenciájú kvarckristály segítségével.
A mikroprocesszor kivezetései négy 8 bites I/O port-ra (PA0-7 -- PD0-7), időzítő (TIMER), interrupt (-INT), alaphelyzet beállító (RESET), órajel (XTAL, IXTAL) és tápfeszültség (VCC, VSS) vonalakra csoportosíthatók.
Az AT-vel és a kijelzőkkel összekapcsolt jelek 7406 típusú inverterek közbeiktatásával kerülnek vételre ill. kiadásra. A billentyűzet mátrix és a processzor között az összeköttetés közvetlen.



4-4. ábra.
Az IBM 101/102 nyomógombos billentyűzet kapcsolási rajza.

Az AT rendszerrel a +KBD CLOCK és a +KBD DATA jelek tartják a kapcsolatot. Ezek a jelek a szabványosított keyboard "protokoll" miatt kétirányúak (ld. 4.4 fejezet - adatátvitel), ezért volt szükség arra, hogy a leválasztó inverterek open kollektorosok legyenek. A +KBD CLOCK keyboard órajelet a HD6805V1 mikroprocesszor állítja elő, de az AT megszakítást kérhet ezen vonal logikai 0 szintre állításával, ezért a mikroprocesszornak időnként mintavételeznie kell ennek a vonalnak az állapotát (6805V1 PA1 bemenet).
A keyboard órajelet a mikroprocesszor az 'A' port 0. bitjén keresztül küldi ki az AT felé (PA0). A +KBD DATA keyboard adatjel ugyancsak az 'A' port-on keresztül kerül kiadásra (PA2), de a -INT bemenet a vételi pont. Ha a keyboard nem küld adatot a rendszer felé (+KBD CLOCK = 1 és +KBD DATA = 1), az AT az adatvonal logikai 0-ba húzásával jelezheti a -INT bemeneten keresztül, hogy parancsot akar küldeni a billentyűzet részére. A parancs bitek is a -INT bemeneten keresztül jutnak a mikroprocesszorba.
Az 'A' port PA3, PA4, PA5 jelű kivezetései kimenetként definiáltak, és a keyboard CAPS LOCK, SCROLL LOCK és NUM LOCK feliratú LED kijelzőit vezérlik.

A billentyűzet mátrix-hoz a mikroprocesszor a 'B', 'C' és 'D' jelű I/O port-okon keresztül kapcsolódik. A billentyűzet letapogatása a következőképpen történik:
A mátrix 16 oszlopának az elemei oszloponként összekötöttek, és az oszlopokat a 'B' és 'C' port-ok bitjei impulzusokkal hajtják meg. A mátrix 8 sorának az elemei szintén össze vannak kötve, és a sorokat a 'D' port bemenetei érzékelik. Valamely billentyű lenyomásakor a billentyű kapcsoló elemként működik, és a mátrixban elfoglalt helye szerint összeköti a megfelelő oszlopot és sort. A mikroprocesszor a 'B' és 'C' port-okra kiadott információ és a 'D' port-on érzékelt és mintavételezett jelek alapján a lenyomott billentyűt (billentyűket) azonosítani tudja.
A keyboard vezérlésével és letapogatásával kapcsolatos programok a mikroprocesszor ROM-jában találhatók, az adatokat, parancsokat, járulékos információkat a processzor RAM-ja tárolja.

Az ismertetett kapcsolás tipikusnak tekinthető valamennyi AT keyboard-ra. Több más cég által gyártott billentyűzet is nagyon hasonlít felépítését tekintve az itt közölthöz, az eltérések a mikroprocesszor típusát (pl. I8049), a kapcsolómátrix kialakítását és a LED kijelzőket érinti.

4.11 Az egér (mouse)

A billentyűzetet jól kiegészítő adat- és parancsbeviteli eszköz az egér (*mouse*). Az egér az ún. "mutató eszközök" (*pointing devices*) csoportjába tartozik, amelyek elsősorban a képernyő kurzor mozgatására, a képernyő valamely pontjának a kijelölésére, képernyő pozíciók bevitelére szolgálnak, és nem szövegbevitelre. Éppen ezért jó kiegészítője a keyboard-nak, és ezért is került a keyboard-dal együtt ismertetésre.

A "mutató eszközök" csoportjához tartozik többek között a fényceruza és a *joy-stick*. Mindkettő számára (az egértől eltérően) gépbe épített ROM-BIOS rutinok készültek, az egereket a BIOS-t és az operációs rendszert kibővítő, meghívás után rezidenssé váló *device driver*-ekkel szállítják.
Elvont értelemben a fényceruza és a *joy-stick* nem különbözik az egértől: mindegyik kétdimenziós szabad mozgást engedélyez a képernyőn, és nyomógombokkal vihetők be a parancsok. A három eszköz közül valószínűleg az egér használható a legkényelmesebben, és ez illeszkedik legjobban a PC alkalmazási köréhez. Ez lehet az oka, hogy az elmúlt években a "mutató eszközök" közül az egerek tettek szert a legnagyobb népszerűségre.

Az egérnek többféle típusa létezik: leggyakrabban az egérbe épített, gumiszerű anyaggal bevont golyó elmozdulásait detektáló típussal találkozhatunk, de vannak optikai megoldásokat alkalmazó egerek is, amelyeknek mindig tartozéka egy az egér alá helyezendő speciális, optikai osztásokat tartalmazó lap.
Az egér, kivitelét tekintve könnyen kezelhető, külön figyelmet nem igényel, a tetején elhelyezett két vagy három nyomógombbal adhatók jelzések a PC számára. Kétféleképpen csatlakoztatható a géphez:

- Vannak olyan típusok, amelyek RS-232 interface-t tartalmaznak, és egyszerűen az RS-232 soros adapter kártya valamelyik csatlakozójába kell dugaszolni őket. Az esetek többségében az egérben elhelyezett CMOS mikroprocesszor külön tápfeszültség be-

25 pontos csatlakozó	Jelmegnevezések	9 pontos csatlakozó
1	Védőföld	csatlakozóház
2	Vett adat (egér veszi)	3
3	Küldött adat (egér küldi)	2
4	Adáskérés	7
5	Adás nyugtázás	8
6	Adatberendezés kész (DSR)	6
7	Jel-föld	5
20	Adat terminál kész (DTR)	4

4-5 táblázat.
Az RS-232 interface-szel rendelkező
egér csatlakozókiosztása.

vezetést nem igényel, a tápfeszültség ellátást az RS-232 interface DTR (*Data Terminal Ready*) jele biztosítja. A DTR jelet a PC küldi az egér felé (logikai 1 szint a 25 pontos csatlakozó 20-as pontján) ha az RS-232 interface kártya üzemkész állapotba került. Az egér csatlakozójának a bekötését 25 és 9 pontos csatlakozók esetén a 4-5 táblázat mutatja.

- Az egerek egy másik típusát képviselik azok, amelyek önálló, I/O buszra csatlakozó adapter kártyával rendelkeznek. Ezeknek az az előnyük, hogy nem foglalják le a soros kommunikációs csatornát.
  Az adapter kártyán található *jumper*-ek a következő beállításokat teszik lehetővé:
  a.) Interrupt bemenet választás IRQ2, IRQ3, IRQ4 vagy IRQ5 (minden egér-adapter kártyán megtalálható). XT gépeknél az IRQ2, AT gépeknél az IRQ5 az ajánlott.
  b.) Elsődleges és másodlagos adapter I/O báziscím beállítás (Microsoft adapter esetén).
  c.) XT 8. I/O csatlakozó választás (Microsoft adapter esetén). Ez a beállítás akkor szükséges, ha a kártya az XT gép J8 jelű I/O csatlakozójában van elhelyezve (XT esetén a J8 csatlakozó különbözik a többitől: -CARD SLCTD, B8 csatl. pont).

Az első egereket eredetileg a *Xerox Research Laboratory* fejlesztette ki grafikus alkalmazások mutató eszközeként. A software fejlesztők azonban, felfedezve az egér flexibilitását és produktivitását, más típusú alkalmazásoknál is figyelembe vették. Ma már természetes, hogy szövegszerkesztő, adatbáziskezelő, desktop publishing programoktól a telekommunikációs programokig mindenütt megtalálhatók. Ennek a kiadványnak a szerkesztésénél is használtak egeret, az MS Word 4.0 szövegszerkesztővel. Különösen jó hatásfokúak a szövegblokk műveletek, amelyek kényelmesen definiálhatók az egérrel. Ismertebb egér forgalmazók a Microsoft, Logitech, Genius, Mouse Systems Corp.

Az egér működése közben interrupt szinten kommunikál a PC-vel. Megszakítás kérések jönnek létre az egér mozgatásakor, vagy a gombok lenyomásakor. Adapter kártyával rendelkező egér esetében az interrupt bemenet választható (IRQ2, 3, 4, 5), az RS-232 soros interface-hez csatlakozó egerek pedig a COM1 vagy COM2-höz tartozó IR bemeneteket (IRQ4, IRQ3) használják.

Az elmozdulást detektáló áramkörök a változásokat X és Y irányú komponensekre bontják és digitalizálják. Egy-egy elmozdulás digit 0,1-0,2 mm-nek felel meg (pl. *Genius Mouse* esetén kb. 1/200" ≈0,13 mm). Az egérbe épített mikroprocesszor előjelhelyesen számolja az elmozdulás-inkrementumokat. Adott periódusidővel mintavételezi a "számlálók" tartalmát, amely periódusidő (*report rate*) egyes egereknél programozható, míg más soros interface-re csatlakozó egereknél ezt az időt a beállított *baud rate* határozza meg. Mintavételezés után a processzor törli a "számlálói" tartalmát, így minden mintavétel csak az előzőhöz viszonyított változást regisztrálja. Az egér akkor jelentkezik átvitel kérelemmel a PC felé a *report rate* által meghatározott időközönként, ha valóban vannak változások a számlálók tartalmát illetően, tehát ha az egér mozog, vagy ha a nyomógombokat működtetjük. Minden egyes átvitelnél átadásra kerülnek az elmozdulás X és Y összetevőit kifejező előjeles adatok és a nyomógombok állapotát mutató státuszinformációk. Ez néhány byte-ot igényel. Az átviteli formák egér típusonként

különbözőek lehetnek, így többféle "protokoll" terjedt el. Például, a három byte-os tömörített bináris, az öt byte-os tömörített bináris (ilyennel működnek a *Mouse Systems Corp.* egerei), hexadecimális formátum, vagy a leggyakoribb a *Microsoft Mouse* protokoll.

A képernyő megjelenítés vagy alfanumerikus, vagy grafikus. Alfanumerikus módban a legkisebb felbontás az egérrel mozgatott kurzor részére az egy karakterblokk (8x8 képpont CGA, 9x14 HERC, EGA esetén). A karakter blokkon belül nincs látható információ az egér elmozdulásáról. Emiatt előfordulhat, hogy vízszintes irányú elmozdításnál a képernyőn a kijelölendő karaktersorozat feletti, vagy alatti karakter sorra ugrik át a kurzor. Grafikus módban az említett probléma nem áll fenn, ellenben a grafikus megjelenítéssel működő szövegszerkesztő képernyő-kezelése lassúbb lehet. AT gépeknél ez kevésbé érzékelhető.

A BIOS szintű interface-t az egér és a felhasználói program között a MOUSE.COM vagy MOUSE.SYS programok képezik, amelyeket az egérrel együtt szállítanak. Feladatuk közé tartozik az egér mozgásának a nyomonkövetése, jelzéseinek vétele, továbbá a szöveg-kurzor vagy grafikus pointer megjelenítése a képernyőn az egértől és a felhasználói programtól kapott információ alapján. A MOUSE.COM egy szabványos *driver*, amely az AUTOEXEC.BAT file-ba téve a gép minden bekapcsolásakor vagy újraindításakor rezidenssé válik, és működőképes állapotba helyezi az egeret. A MOUSE.SYS az utólag installálható BIOS *driver*-ek csoportjába tartozik, és a CONFIG.SYS file-ba kell helyezni az ismert DEVICE = MOUSE.SYS formában. A gép bekapcsolása után az egér interface az operációs rendszer részévé válik, MOUSE.SYS esetén csak tiltani lehet, a memóriából való eltávolítása csak újraindítással lehetséges.

Az egérrel kapcsolatos működési paraméterek:

- A kommunikációs csatorna száma (COM1, COM2)
- *Baud Rate*
- *Report Rate*
- Az érzékenység
- A nyomógombok száma
- Az adatátviteli protokoll

Az érzékenységet az egér elmozdulását mutató inkrementumok számának és a kurzor-elmozdulás kép-pixelekben mért számának az aránya fejezi ki. Minél nagyobb ez a szám, annál kisebb távolságot tesz meg a kurzor a képernyőn ugyanolyan egér elmozdulásoknál (a kurzor mozgása "lassúbbá" válik).

A működési paraméterek az egér *driver* meghívásakor a file neve után álló kulcsokkal állíthatók be. Kulcsok nélkül a *default* működési paraméterek az érvényesek.

Default értékek általában a COM1, 2400 Baud, és a *Microsoft* vagy a *PC Mouse* (*Mouse Systems Corp.*) részére definiált protokoll.

A felhasználói programok zöme, amely az egeret is használja, feltételezi a BIOS szintű interface-t. Bizonyos esetekben maga a felhasználói program is tartalmazhatja a BIOS rutinokat (pl. IBM Top View), de ez kevésbé általános, és ellentétben áll azzal az elfogadott filozófiával, hogy valamely eszközhöz rendelt alapszintű interface legyen a programoktól elválasztva.

Az egér *driver* hardware, software és grafikus kurzor használatát teszi lehetővé. Hardware kurzor pl. a 6845 CRT vezérlő által előállított villogó, vonal vagy téglalap alakú szöveg-kurzor, amely az egér mozgatásakor karakterenként lép tovább (alfanumerikus mód).
A software kurzor szintén karakterenként mozgatható (alfanumerikus mód), de ebben az esetben a kijelölt karakter valamilyen eltérő attribútummal kerül megjelenítésre. Ezen kurzor-típus előállítása teljesen software úton történik, az éppen kijelölt karakter képmemóriában tárolt kód byte-jának és attribútum byte-jának a kijelölés időtartamára történő módosításával.
A grafikus kurzor egy olyan mutató, amely a monitoron látható képtartalom fölött helyezkedik el, és a grafikus ábrán bárhova elmozdítható. Ahogy a kurzor mozog a képernyőn, az alatta fekvő képpontok jellemzőit megváltoztatja, és így alakul ki a kurzor képe, valamint a hozzá tartozó háttér. A kurzor képe sok esetben egy ferde mutató, amely bal felé, vagy jobb felé dőlhet. A grafikus kurzor formáját és a hátteret két 16x16 bites tömb tartalma adja meg, amely software úton definiálható. Az egyik tömb a képernyő maszk, a másik a kurzor maszk. A kurzor mérete nagyfelbontású módban 16x16 pixel, közepes felbontásnál négy színnel 8x16 pixel, közepes felbontásnál 16 színnel 4x16 pixel lehet.
Általában a kurzor-blokk bal felső sarkát tekintik referencia pontnak, amely meghatározza a képernyő koordinátákat (default).
Az egérhez készített *driver*-ek tartalmaznak egy kurzor *flag*-et. A *flag* állapotától függően a kurzor vagy megjelenik, vagy eltűnik a képernyőről.

Az egér BIOS rutinjait a felhasználói software a 33h számú software interrupton keresztül küldött függvényhívásokkal érheti el. A függvényhívások paramétereit az AX, BX, CX, DX regiszterek tartalmazzák. Az AX regiszterbe a hívott függvény száma kerül. Példaképpen, részletesebb ismertetés nélkül felsoroljuk a *Genius Mouse* függvényhívásait:

A függvény száma	Funkció
0	A *Genius Mouse* alaphelyzetbe állítása
1	A kurzor engedélyezése
2	A kurzor tiltása
3	A kurzor helyének és a *Genius Mouse* nyomógomb állapotának az olvasása
4	A kurzor helyének a beállítása
5	A *Genius Mouse* lenyomott gombjaira vonatkozó státuszinformáció olvasása (az utolsó hívás óta bekövetkezett változások)
6	A *Genius Mouse* felengedett gombjaira vonatkozó státuszinformáció olvasása (az utolsó hívás óta bekövetkezett változások)
7	Vízszintes irányú mozgástartomány definiálása a képernyőn a kurzor részére

A függvény száma	Funkció
8	Függőleges irányú mozgástartomány definiálása a képernyőn a kurzor részére
9	A grafikus kurzor megjelenési formájának a definiálása
10	Az alfanumerikus hardware vagy software kurzor megjelenési formájának a definiálása
11	Az egér X és Y irányú elmozdulásának az olvasása (az utolsó hívás óta megtett út X és Y inkrementumainak a száma)
12	Az egérre vonatkozó eseménykezelő rutin belépési cím megadása (események: az egér elmozdulása, valamely nyomógomb állapotának megváltozása)
13	A fényceruzát emuláló működés engedélyezése
14	A fényceruzét emuláló működés tiltása
15	A *Genius Mouse* érzékenységének a beállítása (egér mozgás-inkrementumok/pixelek száma)
16	A kurzor megjelenítés tiltása a képernyő egy definiált területén
19	Dupla sebességre történő átváltás küszöbértékének a megadása

4-6 táblázat.
Egér BIOS szintű függvény hívások (*Genius Mouse*).

Az egerekkel szállított programcsomagok a meghajtó és kezelő rutinokon kívül több ismert program számára kész, kidolgozott *mouse* menüket kínálnak. A *Microsoft* például a dBASE III PLUS, a Lotus 1-2-3, a Lotus Symphony, a Multimate Advantage II., a PFS: Professional Write és a WordStar 3.3/4.0 programok menüit is mellékeli az egérhez.
A Logitech az Edlin (MS), FrameWork, SuperCalc, Symphony, Turbo Pascal, Turbo Prolog, VP Planner, WordPerfect, WordStar és az XTREE programokhoz mellékel kész menüket.

A manapság forgalmazott software termékek zöme támogatja az egér használatát, a *mouse* menü a software része, a program inicializálásakor pedig választhatunk a különféle egér típusok között.

Szokásos dolog olyan felhasználói segédprogramokat is adni az egérhez (Menu Maker, Logimenu), amelyek segítségével azokhoz a programokhoz is készíthetők *mouse* menük, melyeknél eredetileg nem tervezték az egér használatát.

Ezen kívül speciális, az egér lehetőségeit maximálisan kihasználó programokat is találhatunk az egérrel szállított programcsomagban: ilyenek például a PC Paint Brush, Show Partner (MS), POINT EDITOR (Logitech), stb.

5. RÉSZ

Monitorok és monitor adapterek.

5.1 Az MC6845 CRT vezérlő

5.2 IBM monochrome display és printer adapter

5.3 IBM színes grafikus monitor adapter (CGA)

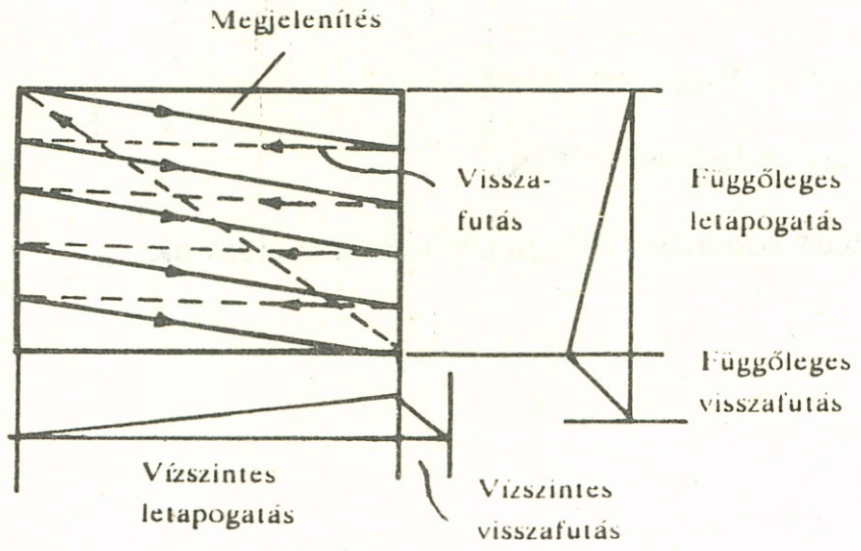
5.4 Nagyobb felbontású grafikus monitor adapterek és monitorok

5.1. Az MC6845 CRT vezérlő

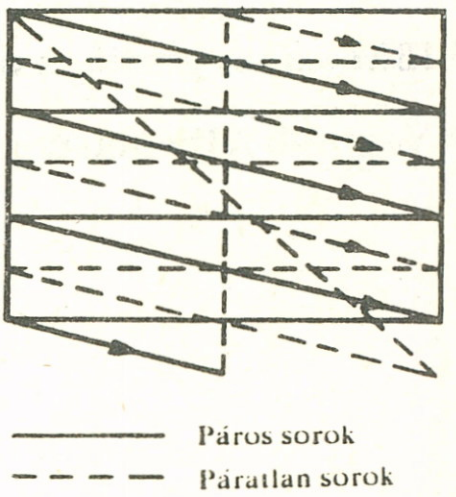
Az IBM személyi számítógépek monitor vezérlő adapter kártyáiban a Motorola MC 6845 típusú intelligens CRT vezérlők kerültek beépítésre. Egyes nagy integráltsági fokú adapter kártyáknál az MC6845 funkcióit VLSI áramkörökkel emulálják. Az MC6845 n-csatornás, szilícium-kapus MOS technológiával készült programozható CRT vezérlő áramkör. A raszter display-k összes szükséges vezérlőjelének az előállítására alkalmas.

A raszteres megjelenítők kétféle letapogatási módban működhetnek. Az első esetben (ld. 5-1.a ábra) az elektronsugár a képernyő bal felső sarkából indulva megjeleníti az első képsort. Ezután az elektronsugarat a vezérlő elektronika kioltja és az eltérítőtekercseket olymódon vezérli, hogy a letapogatás a képernyő bal oldaláról, a következő sor elejéről induljon. Az egész képernyőtartalom megjelenítése után az elektronsugár a jobb alsó sarokból kerül a bal felső sarokba (függőleges visszafutás).

A másik módszer (u.n. váltottsoros letapogatás, 5-1.b ábra) esetében minden egyes kép két félképből tevődik össze. A képernyő bal felső sarkából indulva először a páros számú sorok kerülnek megjelenítésre, majd egy függőleges visszafutási periódus után a képernyő felső szélének közepétől indulva a páratlan számú sorok kirajzolása történik meg. Ez a módszer használatos a televíziós műsorszórásban, valamint a számítástechnikában olyan monitorok esetében, ahol a nagyfelbontású megjelenítés szükséges.



5-1.a ábra



5-1.b ábra

Letapogatási módok

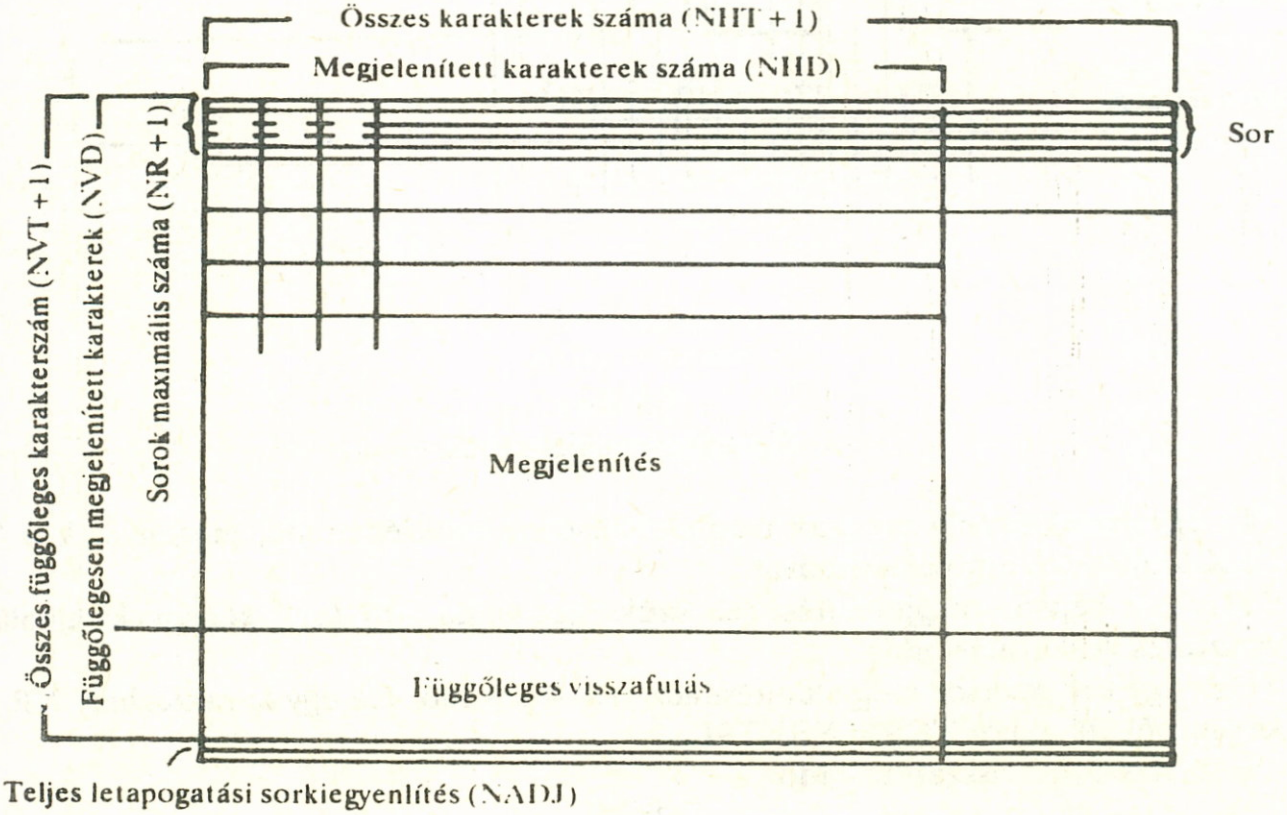
A képernyők rövid utánvilágítási ideje miatt szükséges a képeket bizonyos idő eltelte után újból megjeleníteni (frissíteni). A képfrekvencia (azaz a másodpercenként megjelenített képek száma) növelésével a képernyő bántó villogása csökkenthető, ill. kiküszöbölhető. A kép hullámzásának elkerülése végett a képfrekvenciát a hálózati frekvenciával arányosan szokás megválasztani (műsorszóró TV-nél 25, számítógép megjelenítőnél 50 kép másodpercenként). Az MC6845 mindkét letapogatási módot támogatja.

5.1.1. CRT-vezérlő regiszterkészlet

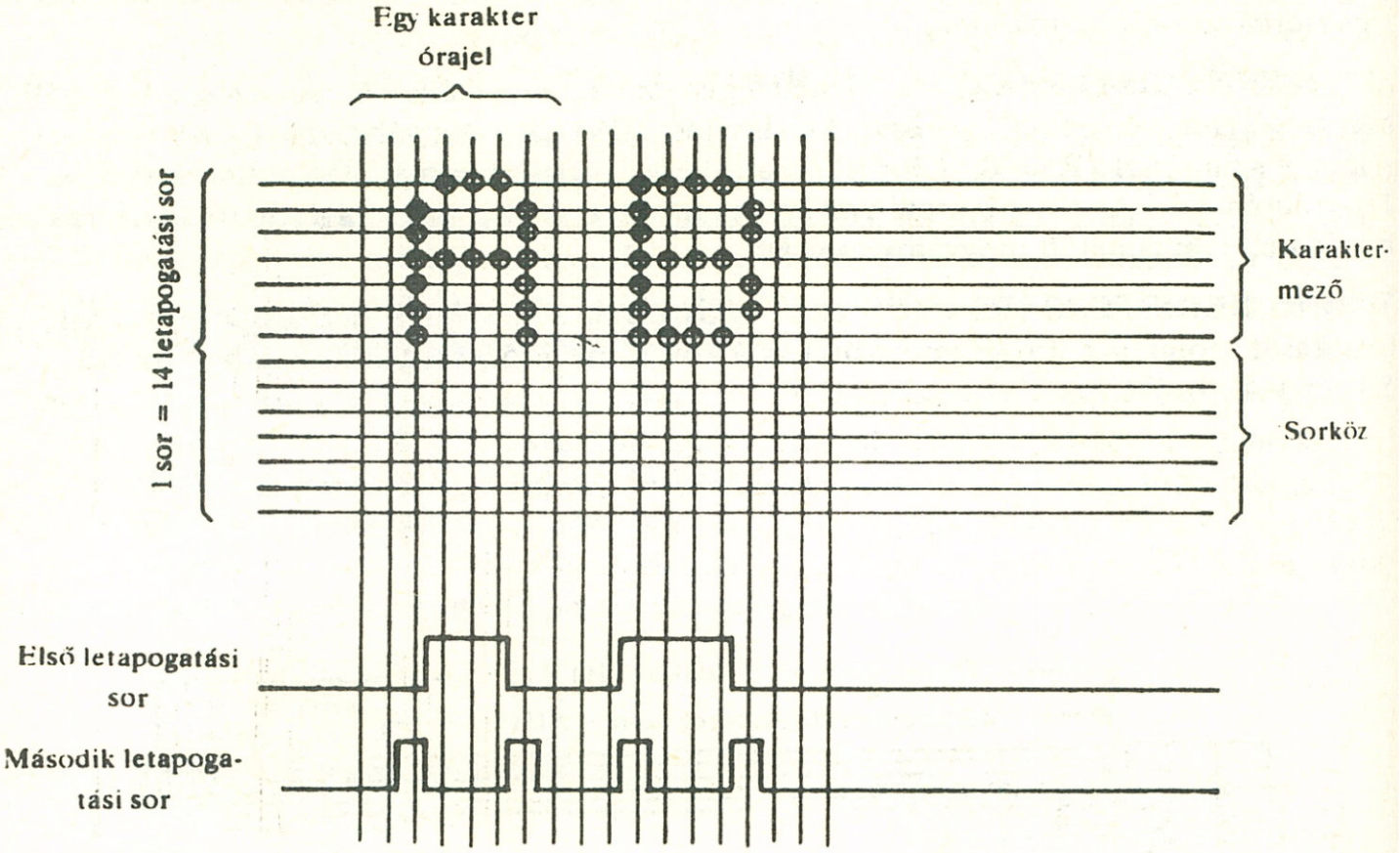
A processzor a vezérlővel annak 19 darab belső regiszterén keresztül kommunikálhat, amelyeket egy 8 bites adatbusz segítségével írhat, ill. olvashat. A következőkben az egyes regiszterek leírását adjuk meg.

ADDRESS REGISTER (AR) -- a 18 adatregiszter valamelyikének megcímzésére szolgáló 5 bites regiszter. A CRTC regiszterkiválasztás (RS) és tokkiválasztás (-CS) vonalainak alacsony szintjénél (RS = 0, -CS = 0) írható be a kiválasztandó adatregiszter sorszáma (0-17). Ezután RS = 1, -CS = 0 esetén az így megcímzett regiszterrel folytatható adatforgalom az AR-beli cím ismételt megváltoztatásáig.

TIMING REGISTERS (R0-R9) -- ezen regiszterek a vízszintes és függőleges időzítések beállítását szolgálják. Programozásukkor a következő időegységek használandók (ld. 5-2., 5-3., és 5-4. ábrák).



5-2 ábra.
Időzítések.



5-3 ábra.
Karakter megjelenítés.

- TC: egy karakterhely sorának megjelenítéséhez szükséges idő, megegyezik a CRTC CLK órajelének egy periódusával.
- TSL: egy képsor megjelenítéséhez szükséges teljes idő (egy képsor kirajzolása + vízszintes visszafutási idő).
- TCR: egy karaktersor megjelenítéséhez szükséges idő. Ha egy karakterhely NR darab képsorból áll, akkor TCR = NR*TSL.
- THR: vízszintes visszafutási idő.

HORIZONTAL TOTAL REGISTER (R0) -- ez a 8 bites regiszter határozza meg a sorszinkron frekvenciát (fHS) az alábbi képlet szerint:

$$fHS = 1/TSL = 1/((R0+1)*TC)$$

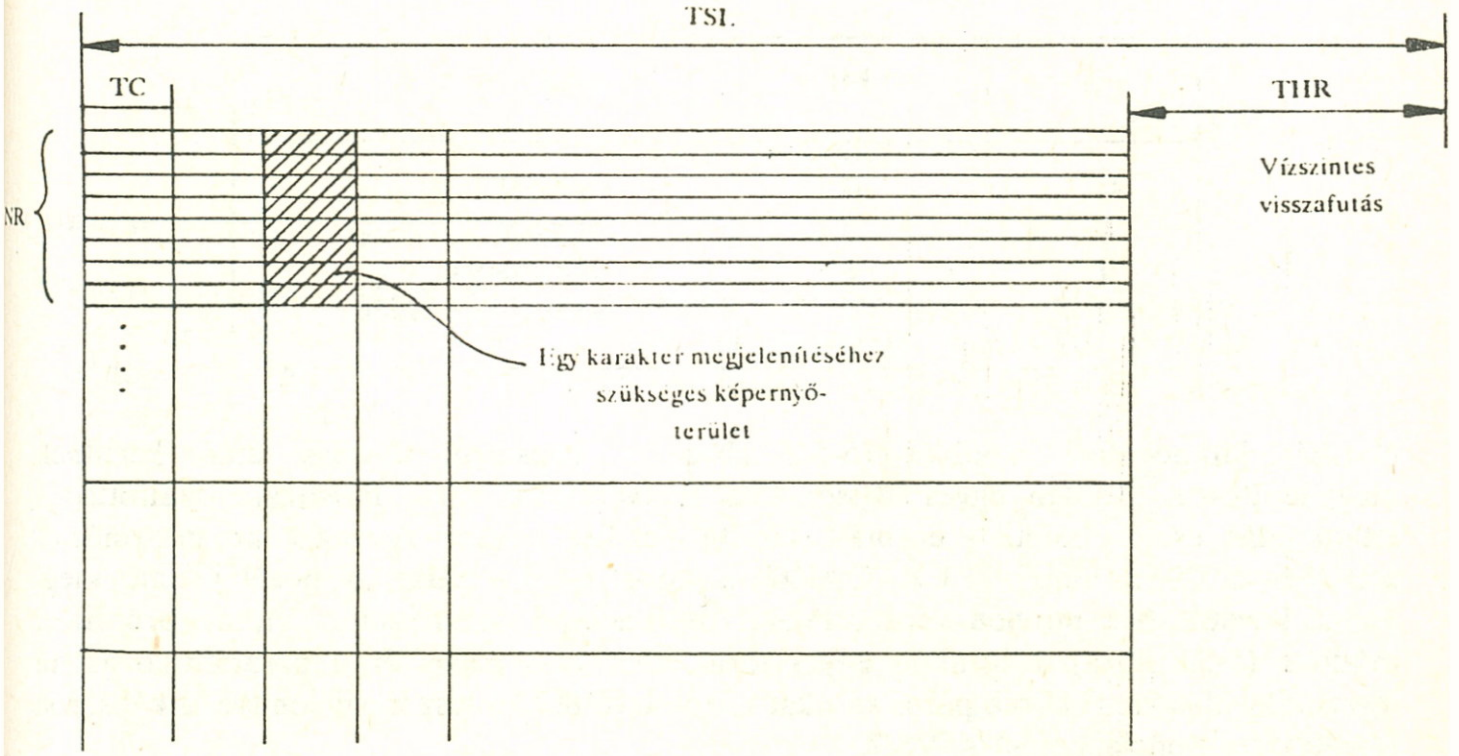
HORIZONTAL DISPLAYED REGISTER (R1) -- 8 bites regiszter, amely az egy sorban megjelenített karakterek számát határozza meg (általában 40 vagy 80 karakter).

HORIZONTAL SYNC POSITION REGISTER (R2) -- a regiszter a sorkioltás kezdete és a sorszinkronjel felfutó éle között eltelt időt határozza meg TC egységekben. A regiszter tartalmának növelése a kép balra-, csökkentése a kép jobbracsúszását eredményezi a képernyőn.

HORIZONTAL SYNC WIDTH REGISTER (R3) -- a regiszter alsó négy bitje a sorszinkronjel szélességét adja meg TC egységekben. R1, R2 és R3 úgy programozandó, hogy tartalmuk összege kisebb legyen R0-nál. R0 és R1 különbsége a sorkioltási idő. A sorkioltási időt (R0-R1) általában a teljes letapogatási idő (R0) 20%-ának szokás választani. A sorkioltási idő három további részre bontható:

- sorszinkron késleltetési idő (R2)
- sorszinkronjel szélesség (R3)
- letapogatás késleltetés (R0-R1-R2-R3)

Általában a fenti három időintervallumot 1:2:2 arány szerint állítják be.



5-4 ábra.
Vízszintes időzítések

VERTICAL TOTAL REGISTER (R4) és

VERTICAL TOTAL ADJUST REGISTER (R5) -- a két regiszter együttesen határozza meg a képszinkron frekvenciáját. Ha két képváltás között eltelt idő TVS, akkor R4-et és

R5-öt az alábbi képlet szerint kell beállítani a kívánt TVS eléréséhez:

$$TVS = (R4+1)*TCR + (R5*TSL)$$

Mivel a képszinkronfrekvenciát (fVS = 1/TVS) a hálózattól függően 50 vagy 60 Hz-re választják a kép hullámzásának elkerülése végett, R4-et és R5-öt úgy állítják be, hogy TVS 20 msec vagy 16,66 msec legyen.

VERTICAL DISPLAYED REGISTER (R6) -- a regiszter alsó 7 bitje a megjelenített karaktersorok számát határozza meg (általában 25). R6 tartalma nem lehet nagyobb R4-énél.

VERTICAL SYNC POSITION REGISTER (R7) -- a regiszter (R2-höz hasonlóan) a képszinkron jel előtti késleltetési időt adja meg TCR egységekben. A regisztertartalom növelése a kép felfelé, csökkentése a kép lefelé csúszását eredményezi a képernyőn. R7 nem lehet nagyobb R4-nél.

INTERLACE MODE REGISTER (R8) -- az MC6845 három letapogatási módot támogat (egy normál és két váltottsorú). A kiválasztás R8 alsó két bitjével történik az alábbi módon:

bit 1	bit 0	mód
0	0	normál
1	0	normál
0	1	váltottsorú
1	1	váltottsorú videó

Normál letapogatási módban (5-5.a ábra) a képsorok egymás után kerülnek megjelenítésre. Minden egyes képsor fVS frekvenciával kerül frissítésre. Váltottsorú letapogatás esetén minden egymást követő két-két képsor ugyanazt az információt tartalmazza (5-5.b ábra). A CRTC előbb a páros, majd a páratlan mezőt jeleníti meg. Ebből következően minden képsor fVS/2 frekvenciával lesz frissítve. Váltottsorú videó módban (5-5.c ábra) a karakter egyes sorai felváltva a páros ill. a páratlan sorokban lesznek kirajzolva. Előbb a páros sorokat tartalmazó félkép lesz megjelenítve, a képsorok frissítési frekvenciája most is fVS/2.

Ha nem normál letapogatási módot használunk, a következő megszorításokat kell figyelembe venni:

- R0 tartalmának páratlannak kell lenni
- váltottsorú videó módban R9 csak páratlan lehet
- váltottsorú videó módban R6-nak párosnak kell lennie és a megjelenítendő karaktersorok számának a felét kell tartalmaznia
- váltottsorú videó módban R10 és R11 tartalmának vagy mind párosnak vagy mind páratlannak kell lennie.

MAXIMUM SCAN LINE ADDRESS REGISTER (R9) -- alsó 5 bitje határozza meg a karaktersoronkénti képsorok számát. A képsorok száma R9 + 1.

```
0 . . . . .
1 * . . . *
2 * . . . *
3 * . . . *
4 * * * * *
5 * . . . *
6 * . . . *
7 * . . . *
```

5-5.a ábra.

```
0 . . . . .
  . . . . . 0
1 * . . . *
  * . . . * 1
2 * . . . *
  * . . . * 2
3 * . . . *
  * . . . * 3
4 * * * * *
  * * * * * 4
5 * . . . *
  * . . . * 5
6 * . . . *
  * . . . * 6
7 * . . . *
  * . . . * 7
```

5-5.b ábra.

```
0 . . . . .
  * . . . * 1
2 * . . . *
  * . . . * 3
4 * * * * *
  * . . . * 5
6 * . . . *
  * . . . * 7
0 . . . . .
  . . . . . 1
2 . . . . .
  . . . . . 3
4 . . . . .
  . . . . . 5
6 . . . . .
  . . . . . 7
```

5-5.c ábra.

Letapogatási módok.

CURSOR START REGISTER (R10) és

CURSOR END REGISTER (R11) -- ezen regiszterek segítségével egy maximum 32 képsor magas kurzorszimbólumot helyezhetünk el valamely karakterblokk alsó sorától kezdődően. Az R10 és R11 regiszterek alsó öt bitje határozza meg a kurzor helyzetét az 5-6. ábrán látható módon.

```
0 . . . . . .
1 . . . . . .
2 . . . . . .
3 . . . . . .
4 . . . . . .
5 . . . . . .
6 . . . . . .
7 . . . . . .
8 * * * * * *
```

R10 = 8 R11 = 8

```
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
* * * * * *
* * * * * *
```

R10 = 7 R11 = 8

```
* * * * * *
* * * * * *
* * * * * *
* * * * * *
* * * * * *
. . . . . .
. . . . . .
. . . . . .
. . . . . .
```

R10 = 0 R11 = 4

5-6 ábra.
Kurzorszimbólum elhelyezése.

Emellett az R10 5. és 6. bitje a kurzor egyéb jellemzőit állítja be az alábbi táblázat szerint:

bit 6	bit 5	kurzormegjelenítés módja
0	0	nem villog
0	1	kurzor nincs megjelenítve
1	0	villog 1/16 képfrekvenciával
1	1	villog 1/32 képfrekvenciával

CURSOR REGISTER (R14-H, R15-L) -- ezen regiszterpár tartalmazza a kurzor címét a 16 kbyte-os képernyő RAM-ban. R15 és 14 bites cím alsó 8 bitjét, R14 a cím felső 6 bitjét adja.

START ADDRESS REGISTER (R12-H, R13-L) -- ezen regiszterpár tartalmát, amely R14, R15-höz hasonlóan egy címet ad meg, arra használja fel a CRTC, hogy meghatározza a 16 kbyte-os képernyő RAM-on belül azt a kezdőcímet, ahonnan a következő megjelenítendő képhez tartozó karakterkódokat el kell kezdeni kiolvasni. A regisztertartalmak megfelelő valtoztatásával megvalósítható a képernyő görgetése (hardware scrolling).

LIGHT PEN REGISTER (R16-H, R17-L) -- a fényceruza (ill. annak illesztője) által az MC6845 LPSTB vonalán generált felfutó él hatására ezen regiszterpárba íródik a CRTC 14 bites címbuszán éppen jelenlévő cím (a 6845 ezen címbuszon címzi folyamatosan a képernyő RAM-ot).

Az alábbi táblázatban összefoglaljuk az MC6845 belső regisztereit és azok funkcióját.

AR 43210	Reg #	Regiszter neve	Progr. egység	READ	WRITE
00000	R0	Horizontal Total	karakter	-	+
00001	R1	Horizontal Displayed	karakter	-	+
00010	R2	H. Sync Position	karakter	-	+
00011	R3	H. Sync Width	--	-	+
00100	R4	Vertical Total	kar.sor	-	+
00101	R5	V. Total Adjust	képsor	-	+
00110	R6	Vertical Displayed	kar.sor	-	+
00111	R7	V. Sync Position	kar.sor	-	+
01000	R8	Interlace Mode	--	-	+
01001	R9	Max Scan Line Address.	képsor	-	+
01010	R10	Cursor Start	képsor	-	+
01011	R11	Cursor End	képsor	-	+

AR 43210	Reg #	Regiszter neve	Progr. egység	READ	WRITE
01100	R12	Start Address High	--	+	+
01101	R13	Start Address Low	--	+	+
01110	R14	Cursor Address High	--	+	+
01111	R15	Cursor Address Low	--	+	+
10000	R16	Light Pen Addr. High	--	+	-
10001	R17	Light Pen Addr. Low	--	+	-

5-1 táblázat.
Az MC6845 belső regiszterei.

A következőkben egy hasznos "programozási ívet" mutatunk be, amely a rajta feltüntetett számítási utasításokkal megkönnyíti az egyes regiszterek tartalmának beállítását. Az ív egy 80*24 karakteres alfanumerikus display-re van kitöltve, de értelemszerűen más formátumokra is felhasználható.

Display formátum programozói ív

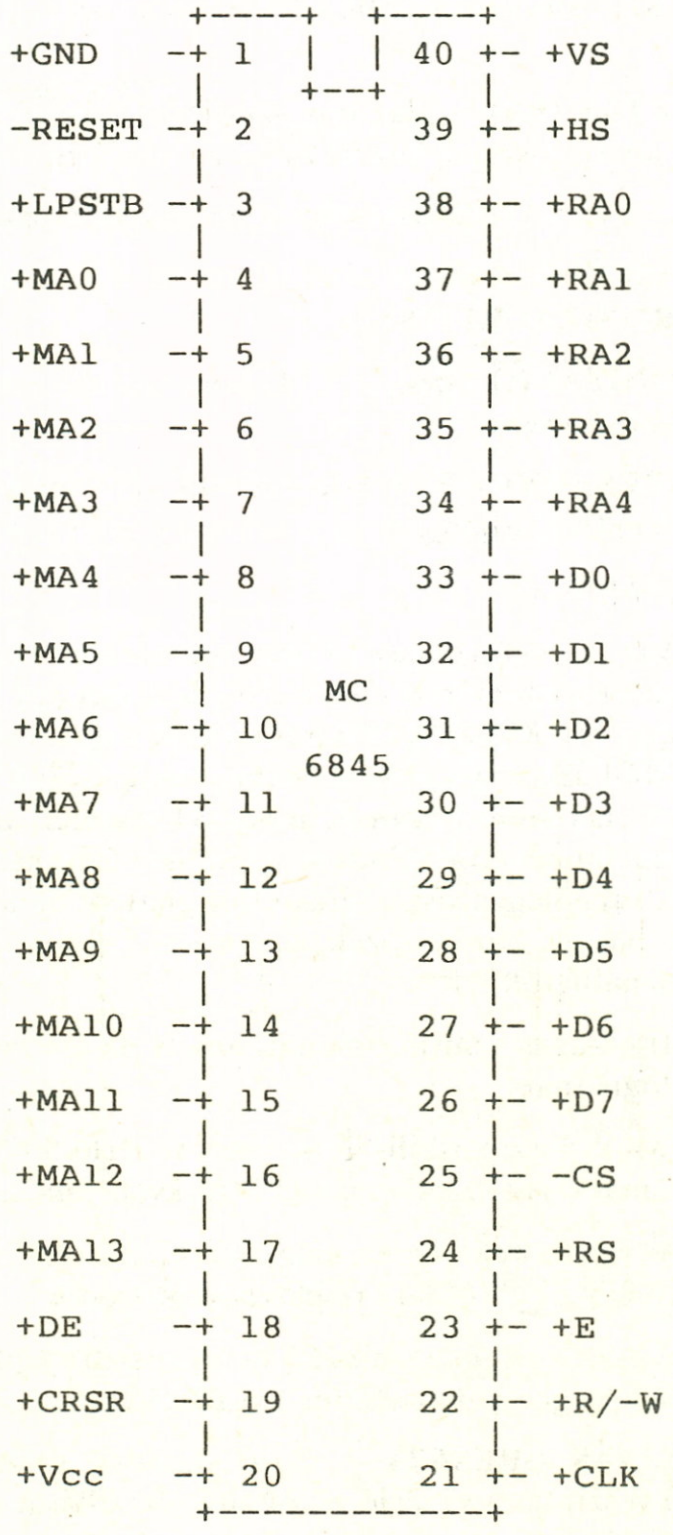
1. soronkénti karakterek száma		*80 kar.*
2. karaktersorok száma a képernyőn		*24 képsor*
3. karakter mátrix	a. oszlop	*7 oszlop*
	b. sor	*9 képsor*
4. karakterblokk	a. oszlop	*9 oszlop*
	b. sor	*11 képsor*
5. képfrekvencia		*60 Hz*
6. oszcillátorfrekvencia		*18600 Hz*
7. képsorok száma aktív (sor2*sor4b)		*264 képsor*
8. képsorok száma teljes (sor6/sor5)		*310 képsor*
9. összes karaktersorok (sor8/sor4b)		*28 kar.sor + 2 képsor*
10. függőleges szinkron késleltetés		*0 kar.sor*
11. függőleges szinkronjel szélesség		*16 képsor*
12. vízszintes szinkron késleltetés		*6 kar.idő*
13. vízszintes szinkronjel szélesség		*9 kar.idő*
14. letapogatás késleltetés		*7 kar.idő*
15. teljes karakteridő (sor1 + 12 + 13 + 14)		*102 kar.idő*
16. karakterfrekvencia (15*sor6)		*1,8972 MHz*
17. képpontfrekvencia (16*sor4a)		*17,0750 MHz*

MC6845 regiszter		Dec	Hex
R0	sor15 - 1	101	65
R1	sor1	80	50
R2	sor1 + sor12	86	56
R3	sor13	9	09
R4	sor9a - 1	27	1B
R5	sor9b	2	02
R6	sor2	24	18
R7	sor2 + sor10	24	18
R8	(ld. a leírást)	0	00
R9	sor4b - 1	10	0A
R10	---	0	00
R11	---	11	0B
R12,R13	--- (high)	128	80
	--- (low)	00	00
R14,R15	--- (high)	128	80
	--- (low)	00	00

A programozói ív adatainak megfelelő
MC 6845 regisztertartalmak.

5.1.2. Processzor - CRTC interface

Ebben a pontban röviden ismertetjük az MC6845 egyes kivezetéseit. A vezérlő lábkiosztását az 5-7. ábrán láthatjuk.



5-7 ábra.
Az MC6845 lábkiosztása.

ADATBUSZ (D0-D7) -- kétirányú busz a processzor és a vezérlő közötti adatforgalom biztosítására.

ADAT ENGEDÉLYEZÉS (E) -- lefutó éle jelzi, hogy az adatbusz vonalak stabilizálódtak és az érvényes adat rendelkezésre áll.

TOKKIVÁLASZTÁS (-CS) -- alacsony szintje kiválasztja a vezérlőt a belső adatregiszterek írása/olvasása céljából.

REGISZTER KIVÁLASZTÁS (RS) -- alacsony szintjénél a vezérlő címregisztere, magas szintje esetén valamelyik adatregisztere között zajlik az adatforgalom.

OLVASÁS/ÍRÁS (R/-W) -- a belső regiszterek és a processzor közötti adatforgalom irányát határozza meg.

FÜGGŐLEGES SZINKRONIZÁCIÓ (VS) és

VÍZSZINTES SZINKRONIZÁCIÓ (HS) -- TTL kompatibilis jelek, melyek a kompozit videójel kialakításához szükségesek.

DISPLAY ENGEDÉLYEZÉS (DE) -- TTL kompatibilis aktív magas szintű jel, amely jelzi, hogy a vezérlő a képernyő aktív (látható) mezője számára ad ki vezérlőjeleket.

MEMÓRIABUSZ (MA0-MA13) és

SORCÍMBUSZ (RA0-RA4) -- a vezérlő egy 14 bites memóriabusszal rendelkezik, amivel a maximum 16 kbyte-os képernyőmemóriát címzi. Emellett van egy 5 bites ún. sorcímbusza is, ezzel a karaktergenerátor ROM-ban levő karakterképek egyes sorait címzi meg. Így egy karakter maximum 32 sorból állhat. Grafikus alkalmazások esetén a memóriacímbusz és a sorcímbusz együttesen egy maximum 19 bites címbuszt képezhet, amivel akár 512 kbyte-os képernyőmemória is megcímezhető. Mindkét címbusz folyamatosan tovább címzi a memóriát a vízszintes, ill. függőleges visszafutás ideje alatt is, de az így kapott adatokat a vezérlő figyelmen kívül hagyja. Ezért ezek a vonalak ilyenkor a dinamikus RAM-memóriák frissítésére használhatók fel.

KURZOR (CRSR) -- aktív magas szintű kimenet, amely az érvényes kurzorcímet jelzi a külső videójelfeldolgozó logikának.

ÓRAJEL (CLK) -- TTL/MOS kompatibilis bemenet a vezérlő 50%-os kitöltési tényezőjű, aktív lefutó élű órajele számára. Az órajel maximális frekvenciája 2,5 MHz.

FÉNYCERUZA (LPSTB) -- ezen bemeneten megjelenő felfutó él hatására az MA0-MA13 vonalakon éppen jelenlevő cím az R16, R17 regiszterekbe íródik.

TÁPFESZÜLTSÉG (Vcc) -- a +5V-os stabilizált és zavarszűrt tápfeszültség bemenete. A névlegestől való legnagyobb megengedett eltérés 5% (0,25V).

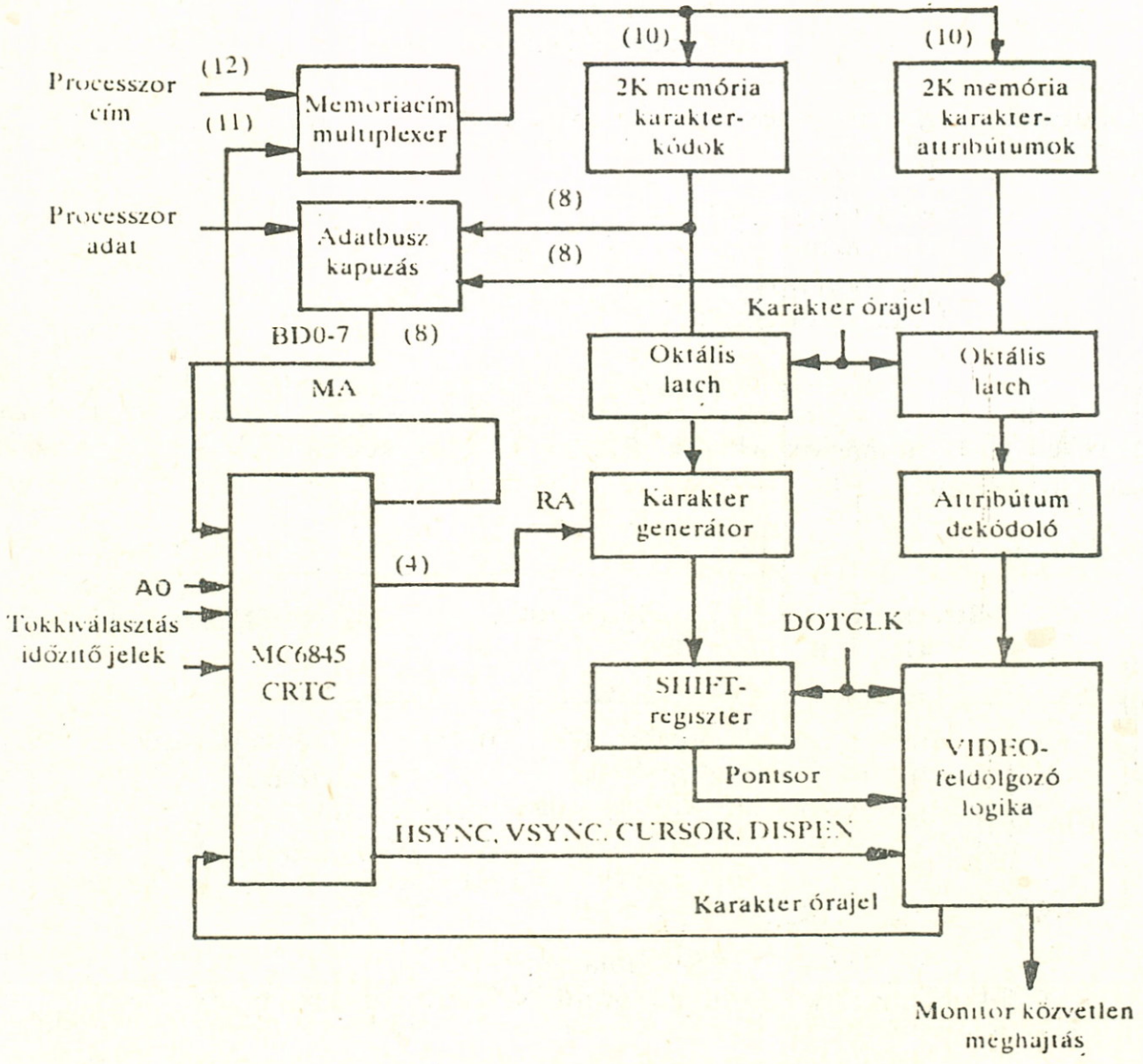
ALAPHELYZETBE ÁLLÍTÁS (-RESET) -- aktív alacsony szintű bemenet, amelynek legalább egy óraciklusnyi 0 szintje a vezérlő alaphelyzetbe állását eredményezi. Hatására az összes belső számláló törlődik és minden kimeneti vonal alacsony szintre kerül.

5.2. IBM monochrome display és printer adapter

Ez a kétfunkciós kártya egyrészt az IBM Monochrome Display-t illeszti a számítógéphez, másrészt párhuzamos nyomtatóinterface-ként szolgál. Ebben az alfejezetben csak a display-illesztő funkciót tárgyaljuk. A kártya az alábbi szolgáltatásokkal, ill. főbb jellemzőkkel rendelkezik:

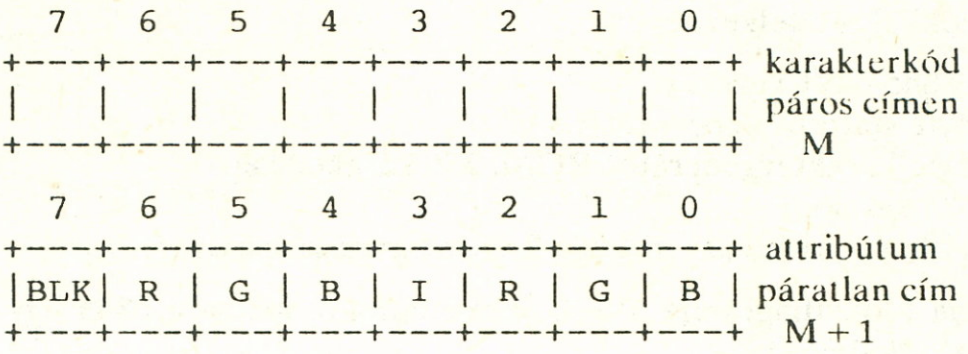
- MC6845 CRT vezérlő
- 4 kbyte statikus képernyő-RAM
- 4 kbyte karaktergenerátor ROM, 256 karakterkép
- 80*25 karakter
- 9*14 képpont karakterblokk, 7*9-es karaktermátrix.

A kártya blokkdiagramja a 5-8. ábrán látható, amely a 6845 egy tipikus (a gyártó által ajánlott) alkalmazása.



5-8 ábra.
Egyszínű adapter kártya blokkdiagramja.

A 4 kbyte-os képernyőmemóriát a számítógép processzora a 0B0000h címtől kezdődően látja. Ebben 80*25 = 2000 darab karakterkód és minden karakterhez egy attribútumkód tárolható. A karakterkódnak mindig páros címen, az ahhoz tartozó attribútumkódnak az azt követő páratlan címen kell elhelyezkedni (ld. a 5-9. ábrát).



5-9 ábra.
Karakterkód és attribútuma.

Az attribútumkód egyes bitjeinek jelentése a következő:

0-2	:	előtér színe
3	:	előtérszín intenzitása
4-6	:	háttér színe
7	:	0 - karakter nem villog, 1 - karakter villog

Az előtérszín, ill. a háttérszín meghatározására szolgáló bitekbe csak bizonyos engedélyezett bitkombinációk írhatók. Ezeket az értékeket és ezek jelentését az alábbi táblázat foglalja össze:

háttérszín			előtérszín			jelentése
R	G	B	R	G	B	
0	0	0	0	0	0	sötét képernyő
0	0	0	0	0	1	aláhúzás
0	0	0	1	1	1	black/white
1	1	1	0	0	0	white/black

5-2 táblázat.
Színkombinációk egyszínű monitor esetén (attribútum byte).

Az egyszínű adapterkártya I/O címtérképe

Az adapter ill. az MC6845 regisztereinek írása vagy olvasása a processzor OUT és IN

utasításain keresztül történik. Az egyes regiszterek az alábbi I/O címeken helyezkednek el:

I/O cím (hex)	Funkció
3B4	6845 címregiszter
3B5	6845 adatregiszter
3B8	üzemmódkiválasztás regiszter
3BA	státuszregiszter
3BC	nyomtató parallel adat 'port'
3BD	nyomtató státusz 'port'
3BE	nyomtató vezérlő 'port'

(A 3BCh -- 3BEh I/O címeken a nyomtatással kapcsolatos funkciók érhetők el.)

5-3 táblázat.
Az egyszínű CRT adapter kártya I/O címtérképe.

5.3 IBM szines grafikus monitor adapter (CGA)

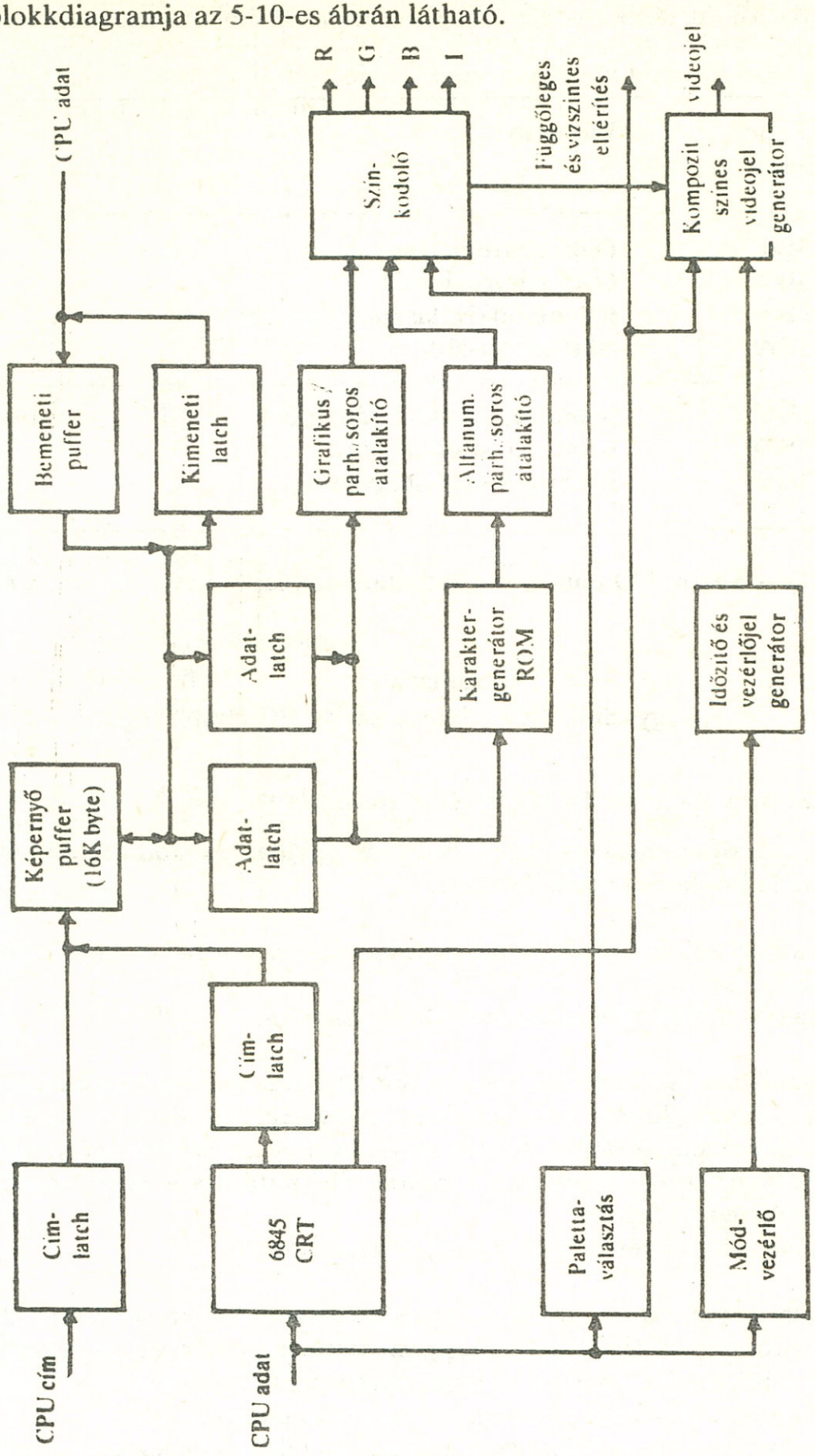
Az adapter mind alfanumerikus, mind grafikus módban működtethető. Alfanumerikus módban kétféle felbontás áll rendelkezésre:

- 40 x 25 karakter, amely kisfelbontású monitorokon és normál otthoni televíziókon is jól olvasható képet ad,
- 80 x 25 karakter a nagyfelbontású monitorok számára.

Mindkét felbontásban egy karakter egy-egy 8x8 képpont méretű karaktermátrixban kerül megjelenítésre. Fekete-fehér üzemmódban lehetőség van inverz, villogó és növelt fényerejű karakterek előállítására. Szines módban minden karakterhelyre 16 háttér és 8 előtérszínből választhatunk, emellett a karakter villogtatására is lehetőség van. A keret a 16 szín bármelyikére beállítható.
Grafikus módban szintén két felbontás közül választhatunk:

- közepes felbontású szines grafikus üzemmódban a képernyő 320 x 200 egyedileg címezhető képpontból áll. Minden képpont színe négy szín valamelyike lehet. A négy színből egy a háttérszín, amely a 16 lehetséges színből választható. A maradék három a két paletta egyike: vagy zöld piros barna, vagy cyan magenta fehér;
- nagyfelbontású grafikus módban csak a fekete és fehér színek állnak rendelkezésre, de a címezhető képpontok száma 640 x 200, tehát a bittérkép elfoglalja az egész 16 kbyte-os képernyő RAM-ot.

Az adapter blokkdiagramja az 5-10-es ábrán látható.

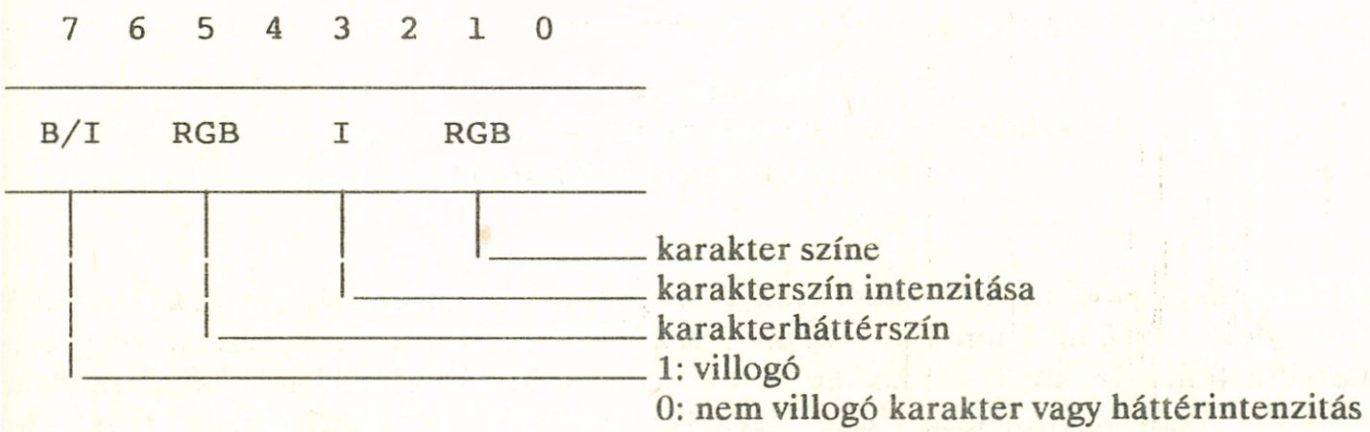


5-10. ábra.
Színes grafikus adapter blokkdiagramja.

A kártya egy MC6845 CRT-vezérlőre épül. A 16 kbyte képernyőpuffer az adapteren van elhelyezve, ezt a számítógép processzora a 0B8000h címtől kezdődően látja. Dual-portos kialakítása révén a memóriát mind a processzor, mind a CRT-vezérlő képes címezni. Nagyfelbontású alfanumerikus mód esetén a vízszintes visszafutási idő alatt csak a processzor érheti el a képernyő RAM-ot. A szükséges változtatásokat ilyenkor célszerű eszközölni a RAM-ban a képernyő villogásának elkerülése érdekében. Ugyancsak az adapterkártyán található a 4 kbyte méretű karaktergenerátor ROM is. Két különböző karakterformát használhatunk, 5 x 7 pont nagyságú sovány ill. 7 x 7 képpont méretű félkövér karaktereket. A betűkészlet a kártyán levő jumper-rel állítható. A félkövér karaktereket a jumper eltávolításával kapjuk.

5.3.1 Alfanumerikus mód

Ezen módban minden egyes megjelenítendő karakterhez egyszónyi információ tartozik a képernyő RAM-ban. A szó páros címen lévő byte-ja a karakterkódot tartalmazza, a páratlan címen lévő ún. attributum byte a karakter megjelenítését szabályozza. Az attributum felépítése a következő:



5-11 ábra.
Attribútum byte (CGA adapter).

A 6,5,4 és 2,1,0 bitek a három alapszín, a vörös (Red), a zöld (Green), és a kék (Blue) megjelenítését szabályozzák. Az alapszínek keverésével, valamint az intenzitás változtatásával összesen 16 színt jeleníthetünk meg.

A későbbiekben ismertetésre kerülő működési mód kiválasztó regiszter segítségével eldönthetjük, hogy az attributum 7. bitje a villogtatást, vagy a háttérszín intenzitását szabályozza. A villogtatásfunkció kiválasztása esetén így csak 8 háttérszín közül választhatunk. A lehetséges színeket a következő táblázat szerint állíthatjuk be:

IRGB	szín
0	fekete
1	kék
2	zöld
3	cián
4	vörös
5	bíbor
6	barna
7	fehér
8	szürke
9	világoskék
10	világoszöld
11	világoscián
12	világospiros
13	világosbíbor
14	sárga
15	fehér

5-4 táblázat.
Színkombinációk CGA monitor adapter esetén
(attribútum byte IRGB bitek).

A kisfelbontású módban egy képernyőnyi információ megjelenítéséhez 40x25x2 = 2000 byte, nagyfelbontású alfanumerikus módban 80x25x2 = 4000 byte szükséges. Az, hogy ezen területek a teljes 16 kbyte memórián belül hol kezdődnek, a 6845 R12, R13 regiszter segítségével határozható meg (lásd MC6845 leírását). A kezdőcímnek kötelezően párosnak kell lennie. A CRT-vezérlő a megjelenítéshez ezen kezdőcímtől kezdve sorban olvassa ki a karakterkódokat ill. attributumbyte-okat, és a karaktereket a bal felső saroktól kezdve balról jobbra és fentről lefelé haladva rajzolja ki.

5.3.2. Grafikus módok

5.3.2.1 Közepes felbontású színes grafikus mód

Ezen módban a képernyő felbontása 320*200 képpont. Az egész képernyőn csak négy színt használhatunk: a beállított háttérszínt és a két, egyenként három színt tartalmazó paletta valamelyikét. A képernyőmemória egy byte-ja négy egymást követő képpontot, és így két bit egy képpontot határoz meg az alábbi módon:

7	6	5	4	3	2	1	0	
C1	C0	C1	C0	C1	C0	C1	C0	
n.		n+1.		n+2.		n+3.		képpont

A C0 és C1 bitek a következőképpen adják meg a képpont színét.

C1	C0	szín
0	0	háttérszín (16 szín valamelyike)
0	1	paletta szín #1
1	0	#2
1	1	#3

A két választható paletta az alábbi színeket tartalmazza:

	Paletta 1	Paletta 2
szín#1	zöld	cián
szín#2	vörös	bíbor
szín#3	barna	fehér

A kép definiálásához a fentiekből következően (320*200)*2/8 = 16000 byte szükséges. Ez a 16 kbyte-os képernyő RAM-ban két, egyenként 8000 byte-os blokkba van szervezve, az egyik blokk a páros, a másik a páratlan sorok megjelenítéséhez szükséges információkat tartalmazza.

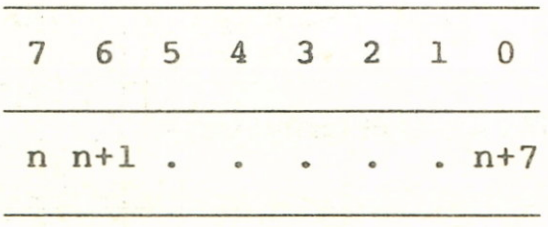
0B8000h	Páros sorok (0,2,4,...198)
0B9F3Fh	8000 byte
0B9F40h 0B9FFFh	nem használt
0BA000h	Páratlan sorok (1,3,5,...199)
0BBF3Fh	8000 byte

0BBF40h 0BBFFFh	nem használt

5-5 táblázat.
A képmemória szervezése CGA 320x200 felbontás esetén.

5.3.2.2 Nagyfelbontású fekete-fehér grafikus mód

Ebben az esetben a felbontás 640*200 pont. Mivel a képernyőmemória minden bitje egy képpontot definiál, így színek kiválasztására nincs lehetőség. Ha a bit 0, a pont színe fekete, ha 1, akkor fehér (bekapcsolt). A képernyőmemória most is két, egyenként 8000 byte-os blokkra van osztva, a blokkok kezdőcíme és szerepe ugyanaz, mint a közepes felbontású módban. Az egyes byte-ok viszont nem 4, hanem 8 képpontot határoznak meg a következőképpen:



5-12 ábra.
Képmemória bitek és képpontok megfeleltetése CGA 640x200 felbontás esetén.

5.3.3 Az adapter programozása

Az adapter ill. az MC6845 regisztereinek írását ill. olvasását a processzor OUT és IN utasításaival végezzük. Az egyes regiszterek az alábbi I/O címeken helyezkednek el:

I/O cím (hex)	Funkció
3D4	6845 címregiszter
3D5	6845 adatregiszter
3D8	üzemmódválasztás regiszter
3D9	színkiválasztás regiszter
3DA	státuszregiszter
3DB	fényceruza latch törlése
3DC	fényceruza latch beállítása

5-6 táblázat.
A CGA adapter I/O címtérképe.

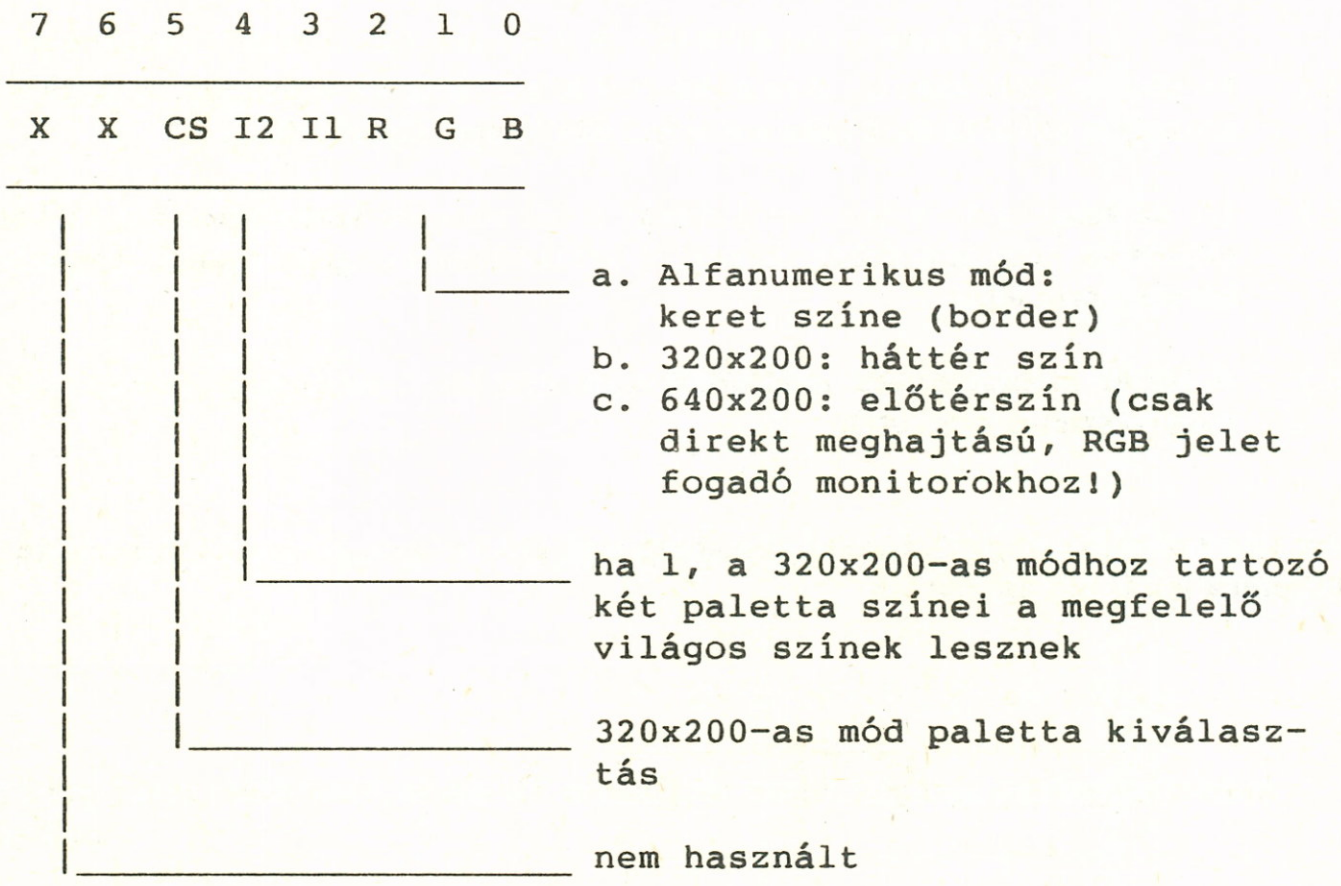
A kártyán lévő MC6845 CRT-vezérlő címregisztere 3D4h, a 18 adatregiszter a 3D5h címen keresztül érhető el.
Az adatregiszterekbe való írás ill. az azokból való olvasás módját és a regiszterek szerepét az MC6845 leírásában már részletesen tárgyaltuk. A következő táblázat az egyes módok helyes működéséhez szükséges értékeket adja meg.

Cím reg.	Reg. szám	Regiszter típus	Egység	I/O	Alfa-num. 40*25	Alfa num. 80*25	Grafikus módok
0	R0	Horizontal Total	Karakter	Csak írható	38	71	38
1	R1	Horizontal Displayed	Karakter	Csak írható	28	50	28
2	R2	Horizontál Sync Position	Karakter	Csak írható	2D	5A	2D
3	R3	Horizontal Sync Width	Karakter	Csak írható	0A	0A	0A
4	R4	Vertical Total	Karakter-sor	Csak írható	1F	1F	7F
5	R5	Vertical Total Adjust	Sor	Csak írható	06	06	06
6	R6	Vertical Displayed	Karakter-sor	Csak írható	19	19	64
7	R7	Vertical Sync Position	Karakter-sor	Csak írható	1C	1C	70
8	R8	Interlace Mode		Csak írható	02	02	02

Cím reg.	Reg. szám	Regiszter típus	Egység	I/O	Alfa-num. 40*25	Alfa num. 80*25	Grafikus módok
9	R9	Maximum Scan Line Address	Sor	Csak írható	07	07	01
A	R10	Cursor Start	Sor	Csak írható	06	06	06
B	R11	Cursor End	Sor	Csak írható	07	07	07
C	R12	Start Address (H)	-	Csak írható	00	00	00
D	R13	Start Address (L)	-	Csak írható	00	00	00
E	R14	Cursor Address (H)	-	Írható/ olvasható	XX	XX	XX
F	R15	Cursor Address (L)	-	Írható/ olvasható	XX	XX	XX
10	R16	Light Pen (H)	-	Csak olvasható	XX	XX	XX
11	R17	Light Pen (L)	-	Csak olvasható	XX	XX	XX

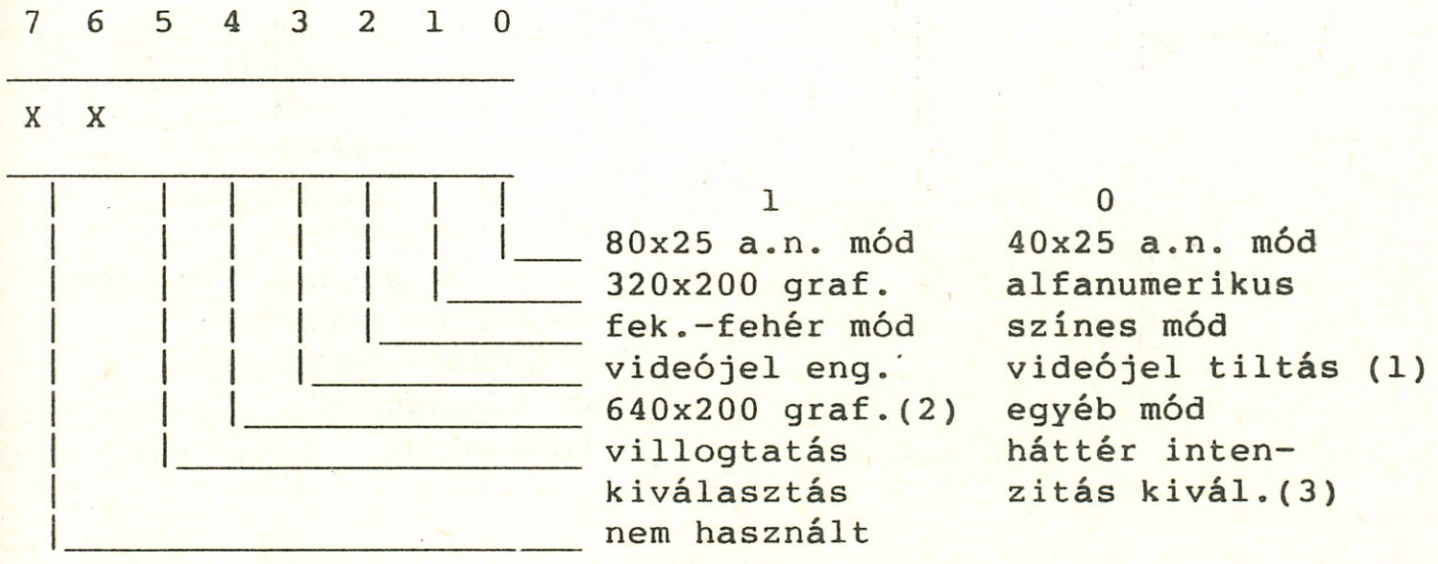
5-7 táblázat.
Az MC6845 CRTC regiszter értékek a CGA adapter alfanumerikus és grafikus módjai esetén.

A színkiválasztás regiszter csak írható, és a 3D9h címen érhető el. A regiszter egyes bitjeihez a következő funkciók vannak rendelve:



5-13 ábra.
A CGA adapter színkiválasztó regisztere.

A módkiválasztás a 3D8h címen levő csak írható regiszter. A regiszter leírása a következő:



5-14 ábra.
A CGA adapter üzemmód kiválasztó regisztere.

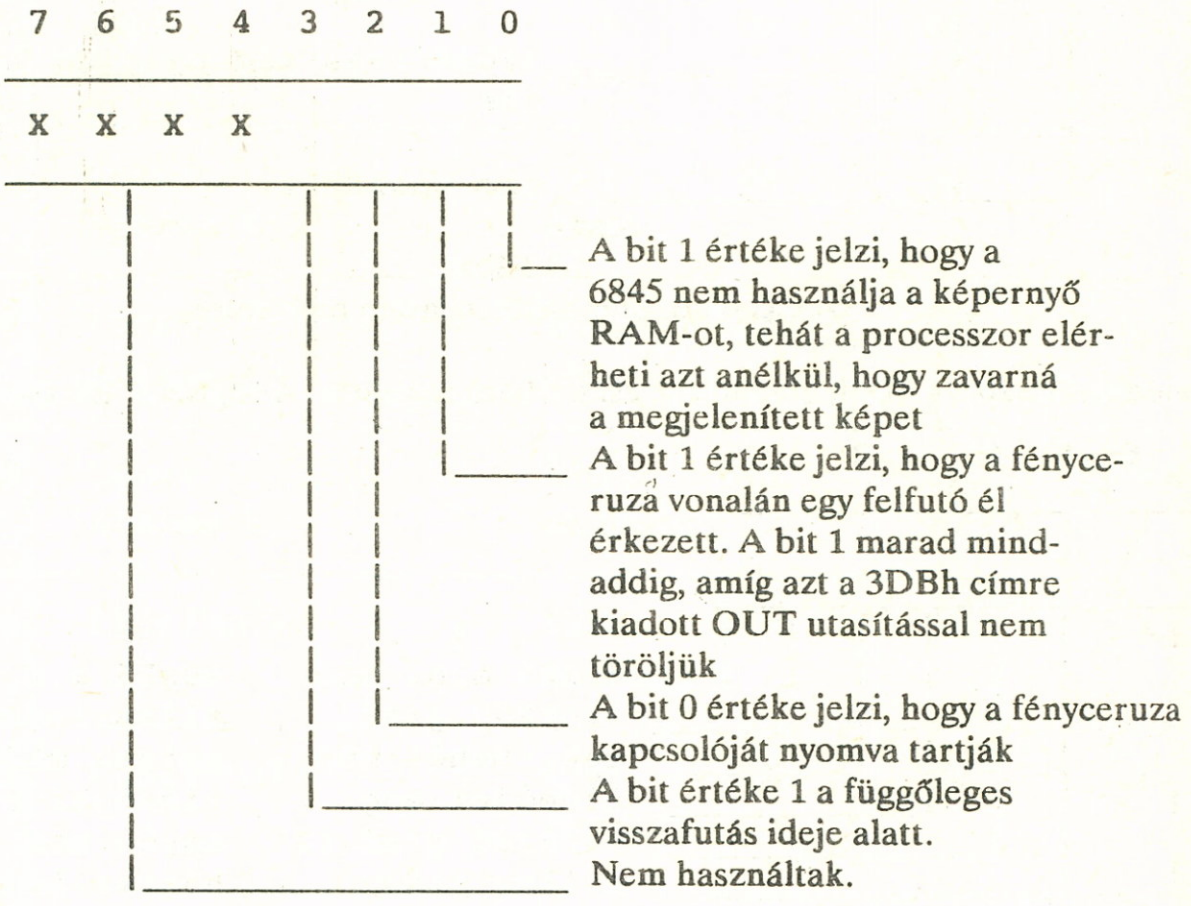
Hivatkozások:

1.) Módváltás előtt ezen bit 0-ra állításával le kell tiltani a videójelet. A módváltáshoz szükséges változtatások után a videójel újból engedélyezhető.

2.) Ha az adapter és a monitor a direkt RGB vonalon keresztül van összekötve, az előtérszín nemcsak fehér, hanem a színkiválasztásregiszter által meghatározott bármely szín lehet.

3.) A bit az alfanumerikus mód attributum byte-jai 7. bitjének jelentését változtatja meg (részletesen ld. ott).

Az adapter státuszregisztere csak olvasható a 0BDAh címen keresztül. A regiszter egyes bitjeinek leírása az alábbi:



5-15 ábra.
A CGA adapter státusz regisztere.

5.3.4 Az adapter csatlakozói

1) 9 tűs apa típusú D-csatlakozó, amelyről direkt RGB jelek vehetők le.

1	föld
2	föld
3	vörös
4	zöld
5	kék
6	intenzitás
7	fenntartott
8	vízszintes elt.
9	függőleges elt.

2) Kompozit videójel (1,5 Vpp) RCA csatlakozója. A videójel NTSC szabványú.

1	videójel
2	föld

3) 4 tűs Berg-Strip csatlakozó, amely RF modulátor meghajtásához szükséges jeleket szolgáltat

1	+12V
2	not used
3	kompozit videojel
4	föld

4) 6 tűs Berg-Strip csatlakozó a fényceruza jeleinek fogadására, ill. a fényceruza áramellátására

1	-fényceruza bemenet (I)
2	nem használt
3	-fényceruza kapcsoló (I)
4	föld
5	+5V (O)
6	+12V (O)

Az (I) bemenetet, az (O) kimenetet jelöl. Az 1. és 2. csatlakozók a kártyára merőlegesen szerelt fém hátlapon, a 3. és 4. csatlakozók magán a kártyán találhatók.

5-8 táblázat.
A CGA adapter csatlakozói.

5.4 Nagyobb felbontású grafikus monitor adapterek és monitorok

Az *IBM Monochrome Adapter* és az *IBM Color Graphics Adapter (CGA)* az első két tagja volt a PC család display vezérlő kártyáinak. A *monochrome adapter*-t professzionális célokra szánták, nagyon jól olvasható karakterkészletet hoztak létre (9x14 képpontos karakterblokk, 7x9 képpontos karaktermező), és csak az egyszínű monitor használatát engedélyezték. Mivel az olcsó árkategóriájú színes monitorok képpont mérete többszöröse az egyszínű monitor képpont méretének, és a sorok száma is kevesebb, az egyszínű monitorok sokkal jobb felbontást, élesebb kontúrt tesznek lehetővé, mint az ilyen színes CGA monitorok. Ennek, és a jó minőségű karakterkészletnek köszönhető, hogy szöveg módban az *IBM Monochrome Adapter* messze felülmúlja a *Color Graphics Adapter*-t. A CGA adapter karakterblokkja 8x8 bites, ezen belül a karaktermező 7x7 bites, a karakterek olvasása hosszabb időn keresztül eléggé fárasztó. Nagy előnye a színválaszték és a grafikus mód, amit az egyszínű IBM adapter kártya nem tett lehetővé. Ezen kívül a CGA adapterek 4 különböző típusú display-vel működhetnek együtt, 7 féle video módban (a video módok felsorolását ld. később az 5-9 táblázatban), szemben a *monochrome adapter* egyféle monitortípusával.
Látható, hogy mindkét alaptípus rendelkezik hiányosságokkal, amelyeket a tervezők később igyekeztek megszüntetni, vagy legalábbis mérsékelni. A Compaq cég pl. olyan CGA kártyát forgalmazott, amely megszüntette szöveg módban a CGA adapterek legnagyobb hibáját, az életlen karaktereket, és ugyanolyan tiszta és finom-rajzolatú karaktermegjelenítést tett lehetővé, mint az IBM egyszínű monitor adapter. Hátránya azonban, hogy ezek a kártyák csak Compaq PC-n üzemelnek.
Az *IBM Monochrome Adapter* alapvető hibája a grafikus mód hiánya. Ezt a hiányosságot pótolta a ma is széles körben használt *Hercules* grafikus adapter kártya. A *Hercules* kártyát úgy tekinthetjük, mint egy *IBM monochrome adapter*-t, kiegészítve nagyfelbontású (720x348), egyszínű grafikus üzemmóddal. A kártya I/O port címei, belső regiszter-címei, a regiszterek elérése és használata teljesen megegyezik az IBM egyszínű adapter ismertetésénél leírtakkal. A *Hercules* grafikus mód az IBM szempontjából nem szabványos, és nem sorolható be az IBM által definiált video módok egyikébe sem. Tekintettel azonban a széles körű használatára, a felsorolásoknál mindig megtalálhatjuk, nem típusszám, hanem *Herc* megjelöléssel.

A legjobban elterjedt nagyfelbontású színes grafikus vezérlő kártya az *IBM Enhanced Graphics Adapter*, vagy röviden EGA adapter. Mielőtt ezt a típust ismertetnénk, felsoroljuk az IBM által a PC-kben szabványosított megjelenítési módokat. Ezeket a módokat számokkal jelölték, és ezekre a számokra hivatkoznak a ROM-BIOS rutinok is.

Az első hét megjelenítési módot (0-6) a CGA, a Compaq CGA, a PCjr adaptere, az EGA, és több más nagyfelbontású monitor vezérlő egyaránt használhatja.
Az első 6 képernyő-mód három párra osztható, ahol minden pár egy színes és egy fekete/fehér változatot tartalmaz ugyanolyan jellemzőkkel. A fekete/fehér módok esetén a szín-attribútumok szürke árnyalatként jelentkeznek a fekete/fehér TV képernyőn, vagy a *composite* video bemenettel rendelkező monitoron. Ezek a módok árnyalatokat kifejező képmegjelenítésre alkalmasak.

Mód #	Típus	Színek?	Sor-szélesség	Megnevezés
0	Szöveg	--	40	40 karakteres szöveg, színek nélkül
1	Szöveg	16	40	40 karakteres szöveg, színekkel
2	Szöveg	--	80	80 karakteres szöveg, színek nélkül
3	Szöveg	16	80	80 karakteres szöveg, színekkel
4	Grafikus	4	320	Közepes felbontású grafikus mód (320 képpont), színekkel
5	Grafikus	--	320	Közepes felbontású grafikus mód (320 képpont), színek nélkül
6	Grafikus	2	640	Nagyfelbontású grafikus mód, fekete/fehér (színek nélkül)
7	Szöveg	Spec.	80	80 karakteres szöveg mód, monochrom
8	Grafikus	16	160	Alacsony felbontású grafikus mód, színekkel
9	Grafikus	16	320	Közepes felbontású grafikus mód, színekkel
10	Grafikus	4	640	Nagyfelbontású grafikus mód, színekkel
11	--	--	--	Nem használt
12	--	--	--	Nem használt
13	Grafikus	16	320	Közepes felbontású grafikus mód, színekkel
14	Grafikus	16	640	Nagyfelbontású grafikus mód, több színnel
15	Grafikus	--	640	Nagyfelbontású grafikus mód, monochrom, 350 sor
16	Grafikus	64	640	Nagyfelbontású grafikus mód, színekkel, 350 sor
Herc	Grafikus	--	720	Nagyfelbontású grafikus mód, monochrom, 348 sor

5-9 táblázat.
Képernyő megjelenítési módok.

A fekete/fehér üzemmódok minden olyan esetben értelmet nyernek, ahol a színek száma 2-nél több. A 6-os számú megjelenítés csak két színt engedélyez, ezért itt nincs értelme külön színes és fekete/fehér változat létrehozásának.
Grafikus módban két fajta felbontás és sorszélesség létezik: a közepes-felbontás 320 képpontot, és a nagy-felbontás 640 képpontot jelent soronként. A sorok száma 200. (Nagyfelbontású adapterkártyák pl. EGA, SPEGA olyan bővített megjelenítési módokat tartalmaznak, ahol a 0-6 módok esetén a sorok száma 350 a szokásos 200 helyett.)

Az *IBM Monochrome Adapter*-hez a 7-es számmal jelölt mód tartozik, amely hasonlít a 80 karakteres 25 soros CGA megjelenítéshez (3). Két lényeges eltérés van azonban: az egyik a karakterkészletre vonatkozik, a másik pedig a színek jelentésére. A színek itt speciális értelmezést kaptak, a karakter megjelenítést befolyásolják: pl. aláhúzás, inverz video, stb.

A 8, 9 és 10-es számú video módok csak a PCjr beépített monitor vezérlőjéhez tartoznak. A 11 és 12 számok hiányoznak a felsorolásból. Ezeket valószínűleg szintén definiálta az IBM, de végül is nem kerültek bevezetésre. Elképzelhető, hogy egy újabb terméknél ezeket is értelmezik.

Az EGA adapterek működtethetők az első 8 video módban, de nem támogatják a PCjr módjait. A 13, 14, 15 és 16-os számú megjelenítési formákat kimondottan az EGA adapterek részére hozták létre. A 13 és 14-es grafikus formában a sorok száma 200, hasonlóan a 4,6 módokhoz, a színek száma azonban megnőtt. A 13 és 14-es mód speciális színes monitort nem igényel. A 15 és 16-os módokban a sorok számát közel kétszeresre, 350-re növelték. A 15-ös szám jelöli az *IBM Monochrome Graphics* módot, amely egyszínű monitort igényel. A 16-os forma csak speciális nagyfelbontású színes monitorral (*Enhanced Color Display*) valósítható meg, eltérően az eddigi CGA színes monitoroktól. A 16-os mód nyújtja a legjobb jellemzőket: 640x350-es felbontás, professzionális karakterkészlet, 64 szín.

Végül külön sorban szerepel a *Hercules* mód, amely nagyon hasonlít a 15-ös módhoz, csak a vízszintes/függőleges képpontok száma különbözik minimális mértékben (720x348), és a vízszintes felbontás 17 %-kal jobb.

Az IBM által definiált video módokhoz különböző típusú és felbontású monitorok használatosak. Az 5-10 táblázat a PC monitorok jellemzőit sorolja fel: a táblázat oszlopai a monitor típusokat, a sorai a jellemzőket tünteti fel. A négyféle típus:

(a) monochrom monitor
(b) közepes felbontású színes monitor
(c) nagyfelbontású színes monitor (16-os mód EGA)
(d) többfrekvenciás nagyfelbontású színes monitor

A speciális többfrekvenciás nagyfelbontású színes monitorok sorfrekvenciája széles tartományban változtatható, így lehetőség nyílik arra, hogy vízszintes irányban (és függőlegesen is) a képpontok számát változtatva sokféle alfanumerikus és grafikus üzemmód realizálható legyen (pl. 132x44 - ld. később).
A monitortípusok közül az első három az IBM és Herc módok valamelyikében működik, a nagyfelbontású többfrekvenciás színes monitorok a *VGA* módokat és az *Advanced* video módokat támogatják (5-12, 5-13 táblázatok).

	Monochrom monitor	Színes monitor	Nagyfelbontású színes monitor	Nagyfelbontású többfrekvenciás színes monitor
Sorfrekvencia	18,432kHz	15,750kHz	21,850kHz	15 - 36kHz
Képfrekvencia	50Hz	60Hz	60Hz	50 - 60Hz
Video sávszélesség	16,257MHz	14,318MHz	16,257MHz	39MHz
Maximális felbontás	720x350	640x200	640x350	800x600

5-10 táblázat.
PC monitorok jellemzői.

A táblázatból kitűnik, hogy minél nagyobb felbontású egy monitor, annál nagyobb sorfrekvenciával üzemel, és a video sávszélesség is jelentősen megnövekszik.

A következő felsorolás a PC display típusokra ad példákat:

1.) **Monochrom monitorok**
   a.) *IBM Monochrome Display*
   b.) *IBM Monochrome Display*-vel kompatibilis monitorok

2.) **Színes monitorok**
   a.) *IBM Color Display*
   b.) *IBM Color Display*-vel kompatibilis monitorok

3.) **Nagyfelbontású színes monitorok**
   a.) *IBM Enhanced Color Display*
   b.) *IBM Enhanced Color Display*-vel kompatibilis monitorok

4.) **Nagyfelbontású többfrekvenciás színes monitorok**
   (Speciális adapterkártyát igényelnek)
   a.) *Magnavox Multimode Display*
   b.) *Mitsubishi AUM 1371A Display*
   c.) *NEC MultiSync Display, JC14-01*
   d.) *Sony Multiscan Display*
   e.) *Taxan 770*
   f.) *Electrohome 19" Vari-Scan Display, ECM 1910*
   g.) *Microvitec 19" Auto-Sync Display Models 1019/SP és 1019/LP*
   h.) *Conrac 19" 7250C19*
   i.) *NANAO 8060S, 8060G1*
   j.) Egyéb, a felsoroltakkal kompatibilis monitorok.

5-11 táblázat.
PC monitorok típusai.

A nagyfelbontású színes monitorok (3) vezérlő kártyája az *Enhanced Graphics Adapter* (EGA) egyesíti magában valamennyi elődjének (CGA, MDA, HERC) tulajdonságait, egységesíti az adaptereket és új video módokat támogat. Hátránya, hogy kb. kétszer annyiba kerül, mint egy CGA adapter, és előnyei csak a nagyfelbontású monitorral együtt mutatkoznak meg igazán, amelynek szintén jóval magasabb az ára az egyszerűbb színes monitorokhoz viszonyítva.
Az EGA adapterek jellemzője a nagyméretű saját memória (64 kbyte -- 256 kbyte). A nagyobb felbontású színes grafikus módok egyre növekvő memóriaméreteket igényelnek. A 16-os számú video mód megvalósításához pl. pixelenként 6 bit szükséges a 64 lehetséges szín kiválasztása miatt. Így egy 640x350 felbontású kép tárolásához 1.344.000 bit, ill. 168 kbyte szükséges. A kártyára épített memóriaméret igen eltérő lehet.
Alaphelyzetben az EGA kártyák memória kezdőcíme egybeesik a 0B000h vagy a 0B800h címekkel. Amikor az EGA adapter valamelyik speciális üzemmódjába kerül, az 'A' memória-szegmens címzését is használja. Ellenben, ha a szokásos video módok egyikét használja, az EGA memória soha nincs az 'A' blokk címeihez kapcsolva.
A PC-k memória címtérképe szerint két szegmens, az 'A' és a 'B' (összesen 128 kbyte) a kijelölt video terület. A 'B' blokkot más display adapterek is használhatják (CGA, MDA, HERC), amelyek mint másodlagos display vezérlők működnek az EGA adapter mellett. Így végeredményben csak az 'A' szegmens maradt az EGA kártya memória-címtartományára. Mivel a memóriaméret a 64 kbyte-os 'A' blokk többszöröse lehet, az EGA kártyára épített memóriáknak mindig csak egy része kerül bekapcsolásra, és válik elérhetővé a PC fizikai címtartományán belül. A memóriarészek program útján az EGA áramköreivel ki- és bekapcsolhatók, így az EGA kártyán elhelyezett memória minden része hozzáférhetővé válik. Azok a programok, amelyek alapszinten kötődnek az EGA adapterhez, a nagyobb felbontású video módokban olyan parancsokat adnak át a kártyának, amelyek a memóriacímzést vezérlik.

A többfrekvenciás nagyfelbontású monitorok speciális adapterkártyát igényelnek. A képernyő közelítőleg 500.000 képpontból állhat, amelyeket a legkülönfélébb csoportosításban használnak a *VGA* és az *Advanced Display* módok megvalósítására. A *VGA* és az *Advanced Display* megjelenítési módok jellemzőit mutatja be az 5-12 és az 5-13 táblázat.

Mód #	Típus	Szöveg formátum	Kar. blokk méret	Felbontás	Színek száma	Max. lapok száma
VGA[0+,1+]	TEXT	40x25	9x16	360x400	16/64	8
VGA[2+,3+]	TEXT	80x25	9x16	720x400	16/64	8
VGA[4+,5+]	GFX	40x25	8x16	320x200D	16/64	8
VGA[6+]	GFX	80x25	8x16	640x200D	16/64	8
VGA[D+]	GFX	40x25	8x16	320x200D	16/64	8
VGA[E+]	GFX	80x25	8x16	640x200D	16/64	4

Mód #	Típus	Szöveg formátum	Kar. blokk méret	Felbontás	Színek száma	Max. lapok száma
VGA[10]	GFX	80x66	8x14	640x350	16/64	2
VGA[11]	GFX	80x30	8x12	640x480	16/64	1
VGA[12]	GFX	80x29	8x12	640x480	16/64	1
EGA	TEXT	132x25	8x14	1056x350	16/64	8
EGA	TEXT	132x29	8x12	1056x348	16/64	8
EGA	TEXT	132x32	8x11	1056x352	16/64	8
EGA	TEXT	132x44	8x8	1056x352	16/64	8
EGA	TEXT	132x60	8x8	1056x480	16/64	8
CGA	T/G	40x25	8x8	320x200	4	8
CGA	T/G	80x25	8x8	640x200	2	8

5-12 táblázat.
VGA megjelenítési módok.

Mód #	Típus	Szöveg formátum	Kar. blokk méret	Felbontás	Színek száma	Max. lapok száma
GA1	T/G	40x25	8x16	320x400	16	8
GA2	T/G	80x25	8x16	640x400	16	8
GA3	BGAME	80x25	8x8	640x400	16/64	8
GA4	TEXT	80x60	8x8	640x480	16/64	8
GA5	GFX	80x60	8x8	640x480	16/64	8
GA6	TEXT	80x66	8x8	640x528	16/64	8
GA7	GFX	80x66	8x8	640x528	16/64	8
GA8	TEXT	94x29	8x12	752x406	16/64	8
GA9	GFX	94x29	8x12	752x406	16/64	8
GA10	TEXT	100x75	8x8	800x600	16/64	8
GA11	GFX	100x75	8x8	800x600	16/64	8
GA12	TEXT	114x60	8x8	912x480	16/64	8
GA13	114x60	8x8	912x480	16/64	8	

5-13 táblázat.
Advanced megjelenítési módok.

A táblázatokból láthatjuk, hogy alfanumerikus módban a legváltozatosabb szövegformátumok választhatók: a soronkénti karakterek száma akár 132 is lehet (40 -- 132 tartomány), a sorok száma pedig elérheti a 66-ot (8x8-as karakterblokk esetén).
Grafikus módban a sor/oszlop arány valamint a soronkénti és oszloponkénti képpontok száma ugyancsak sokféleképpen választható.

A többfrekvenciás nagyfelbontású monitorok PC adapterei meglehetősen bonyolultak, esetenként külön mikroprocesszort, vagy VLSI áramköröket, logikai tömböket tartalmaznak. A GENOA cég *SuperEGA HiRes+* típusú adaptere pl. olyan VLSI tömböket tartalmaz, amelyek nemcsak a *VGA* és *Advanced* módok realizálására képesek, de az IBM által szabványosított 0-16 és Herc módok is tetszőlegesen választhatók a kártya hátlapján hozzáférhető mikrokapcsolók segítségével. A választott video módban az adapter regiszter-szinten is kompatibilis a megfelelő CGA, EGA vagy HERC adapterkártyával.

6. RÉSZ

MULTI I/O kártya.

6.1 A MULTI I/O kártya rövid leírása

6.2 A MULTI I/O kártya alkalmazása

6.3 RS-232 soros illesztő csatlakozás

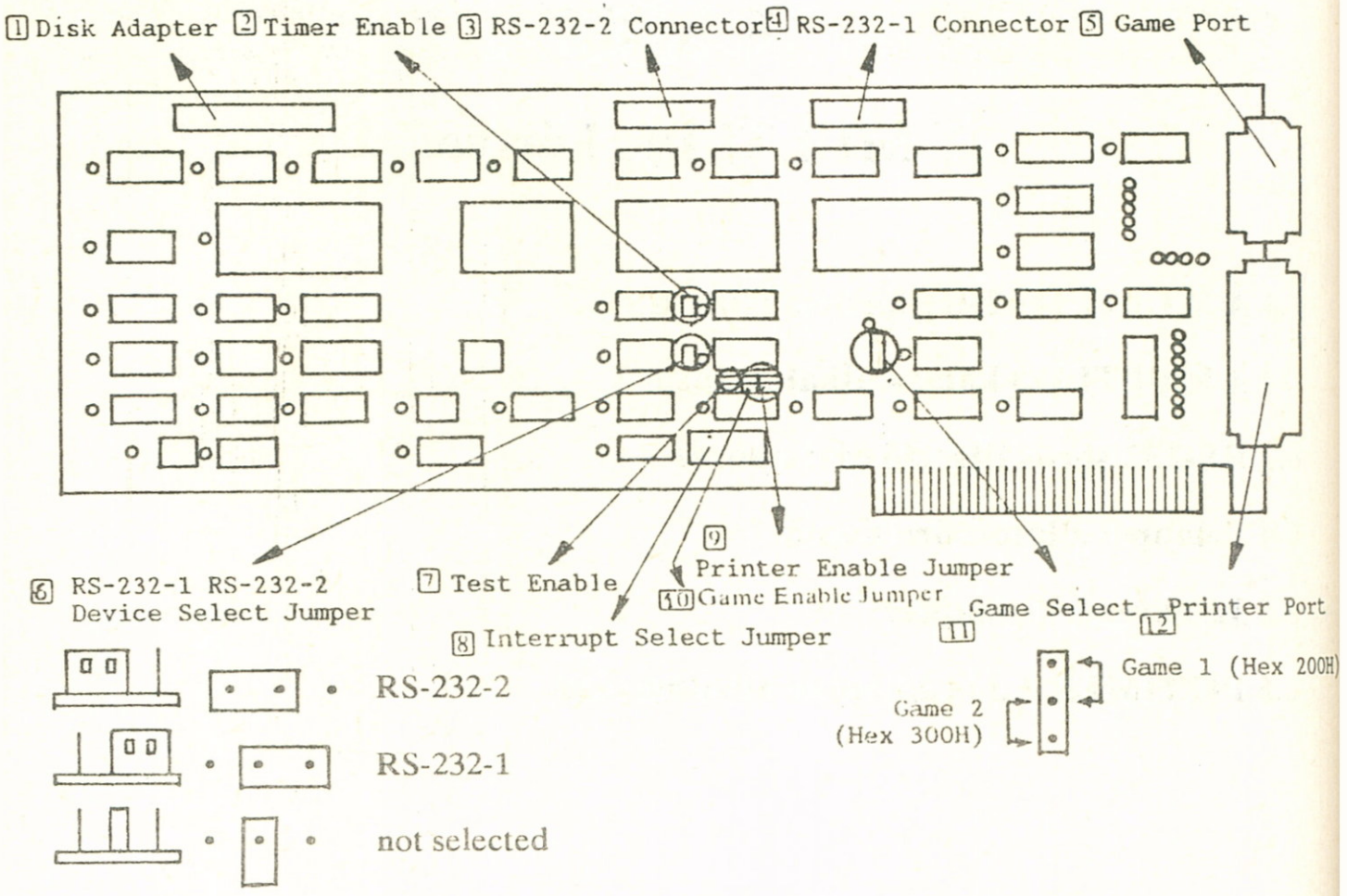
6.4 Teleppel ellátott óra/naptár

6.5 Játékvezérlő adapter

6.6 PC TIMER 1.3 óra/naptár alkalmazása

A PRIPHERALS ENTERPRISE COMPANY által gyártott MULTI I/O kártya alkalmazható minden 16 bites IBM PC kompatibilis személyi számítógéphez, és az alábbi 5 funkcióval rendelkezik:

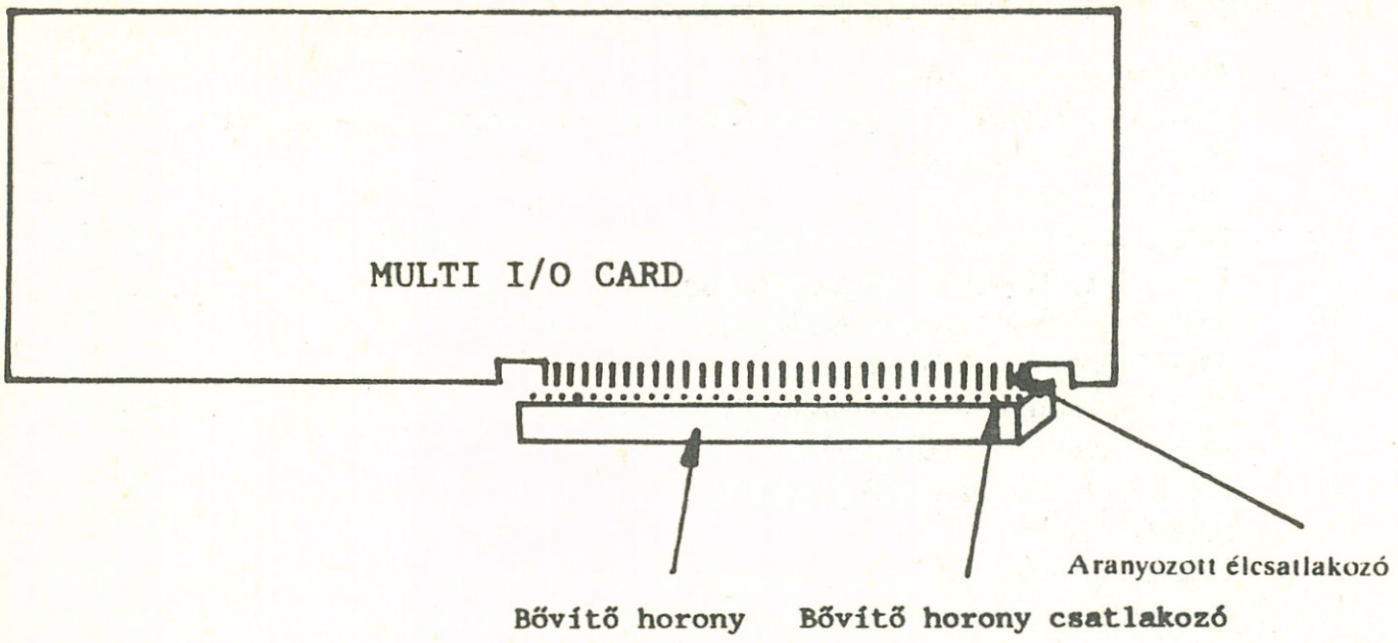
1. Floppy-diszk adapterként, két DS/DD lemezegység vezérlése.
2. Párhuzamos nyomtató-illesztése.
3. Két soros (RS-232-1, RS-232-2) ki/bemenet illesztése.
4. Tápfeszültség-független óra/naptár.
5. Játék adapter.



6-1 ábra.
A MULTI I/O kártya elrendezési rajza.

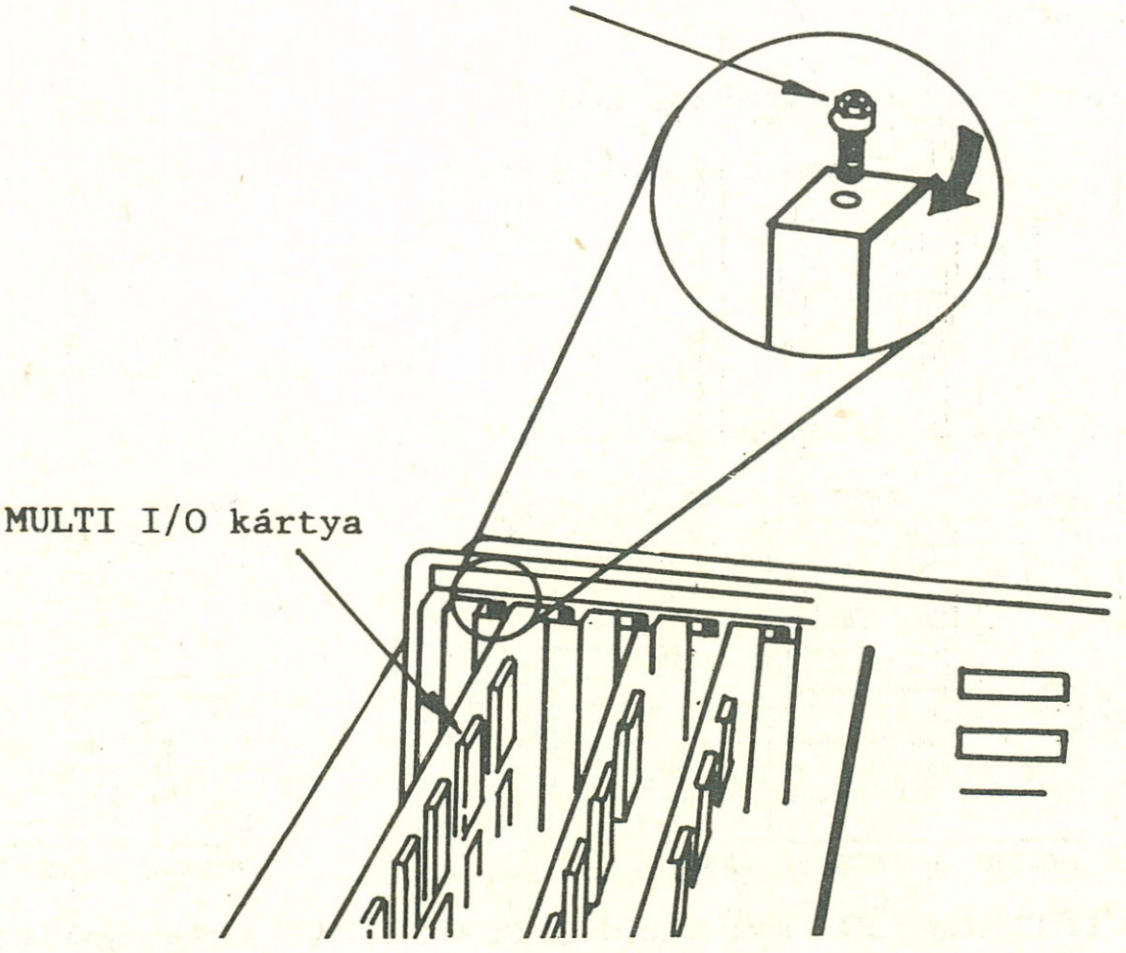
A MULTI I/O kártya üzembehelyezése:

- Kapcsoljuk ki a rendszert.
- Csavarhúzóval lazítsuk meg a kártya rögzítő csavart.
- Tegyük be a kártyát a bővítő horonyba.
- Figyelem! A MULTI I/O kártyát 31 aranyozott élcsatlakozó kapcsolja az alaplemezhez. A kártya üzembehelyezése előtt győződjünk meg a biztos csatlakozásról!
- Húzzuk meg csavarhúzóval a rögzítő csavart!



6-2 ábra.
A kártya behelyezési vázlata.

Így helyezzük be a kártyarögzítő csavart



6-3 ábra.
A kártya rögzítése.

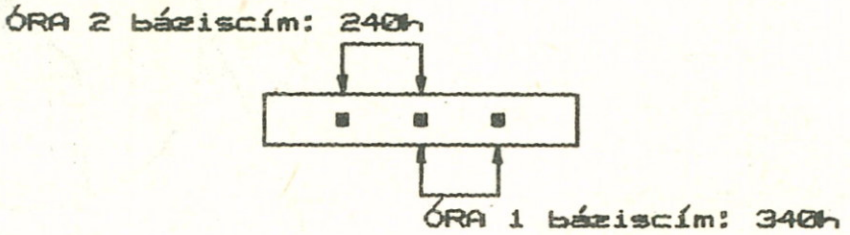
6.1. A MULTI I/O kártya rövid leírása

1/ Mágneslemez adapter
2/ Óra engedélyezés (Timer Enable) (I/O címkiválasztó dugó JP1)
3/ RS-232-2 csatlakozó
4/ RS-232-1 csatlakozó
5/ Játék csatlakozó
6/ RS-232-1, RS-232-2 kiválasztó dugó JP2
7/ Teszt vezérlés - nyomtató kiválasztó dugó JP3
8/ Megszakítás kiválasztó dugó
9/ Nyomtató engedélyező dugó JP5
10/ Játék engedélyező dugó JP4
11/ Játék I/O címkiválasztó dugó JP6
2. játék (300h)
1. játék (200h)
12/ Nyomtató csatlakozó
(A számozás a 6-1 ábra jelöléseinek megfelelő)

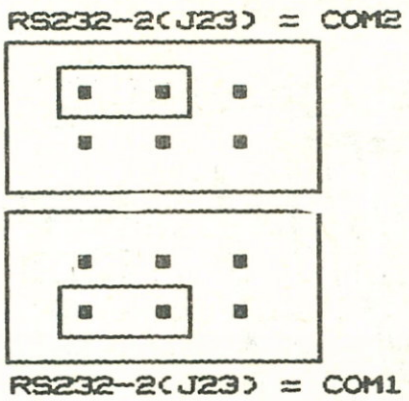
- Mágneslemez adapter: biztosítja két FDD lemezmeghajtó szalagkábelének csatlakoztatását.

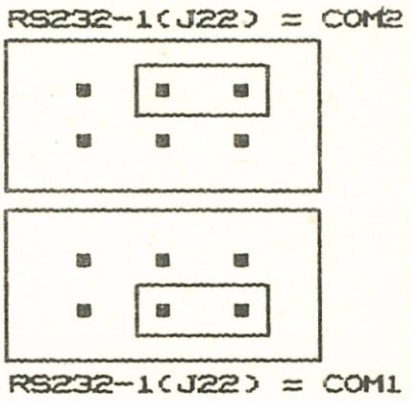
- RS-232-1 (J22), RS-232-2 (J23): biztosítja egy-egy RS-232 szabványú soros adatillesztő csatlakoztatását.

- JP1: óra I/O báziscímét kiválasztó dugó



- JP2: COM1-COM2 (kommunikációs csatorna) kiválasztó dugó





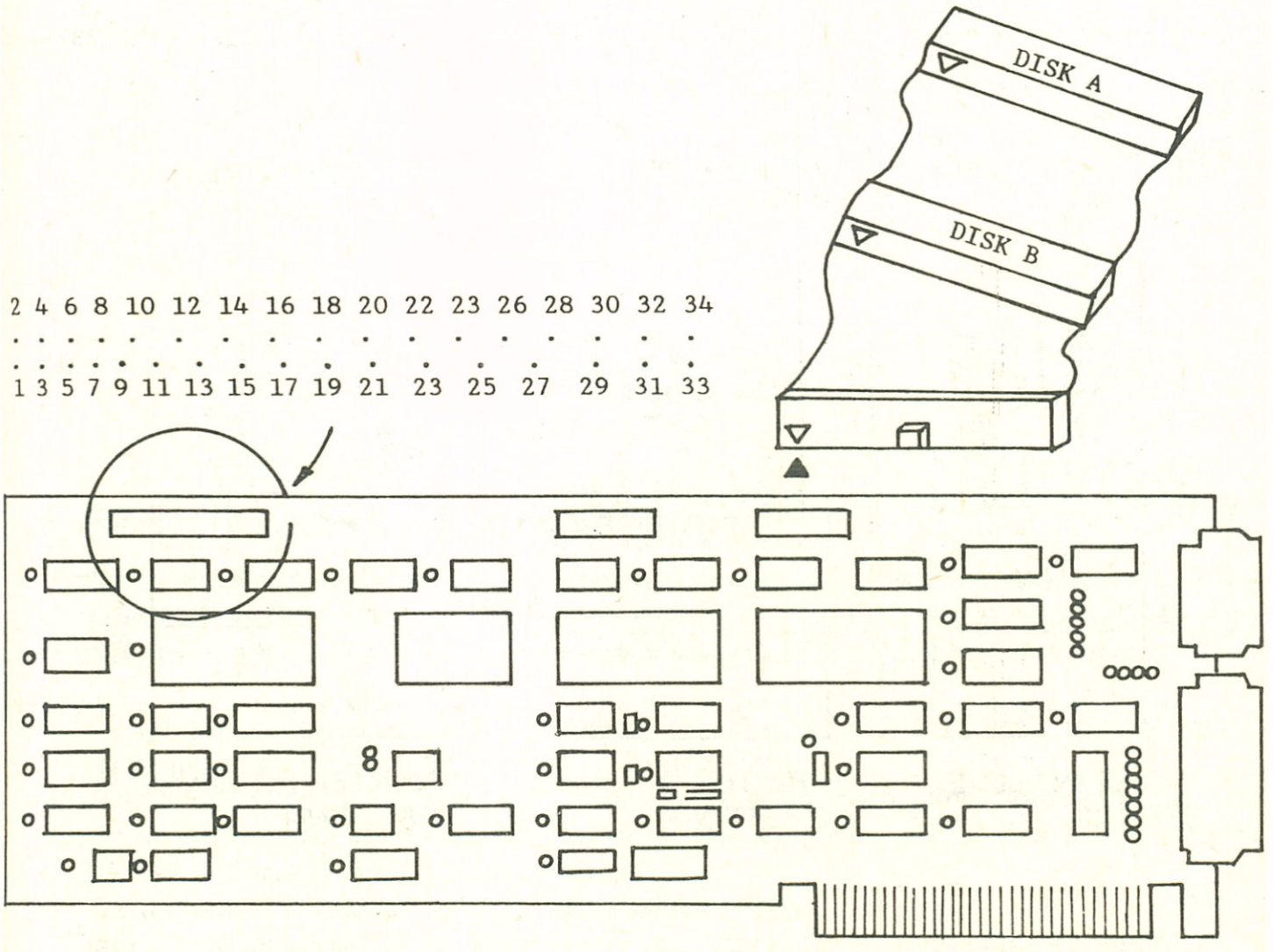
- JP3: LPT1 vagy LPT2 kiválasztó dugó (a MULTI I/O kártya nyomtató interface-t definiálja LPT1-ként, vagy LPT 2-ként)
- JP4: játék engedélyező dugó

- JP5: nyomtató engedélyező dugó
- JP6: játék adapter I/O cím választás
- MEGSZAKÍTÁS: szint kiválasztható
- NYOMTATÓ CSATLAKOZÓ: biztosítja a párhuzamos nyomtató csatlakoztatást
- JÁTÉK CSATLAKOZÓ: biztosítja a botkormány(ok) csatlakoztatását (joy-stick).

6.2 A MULTI I/O kártya alkalmazása

6.2.1 Mágneslemez meghajtó

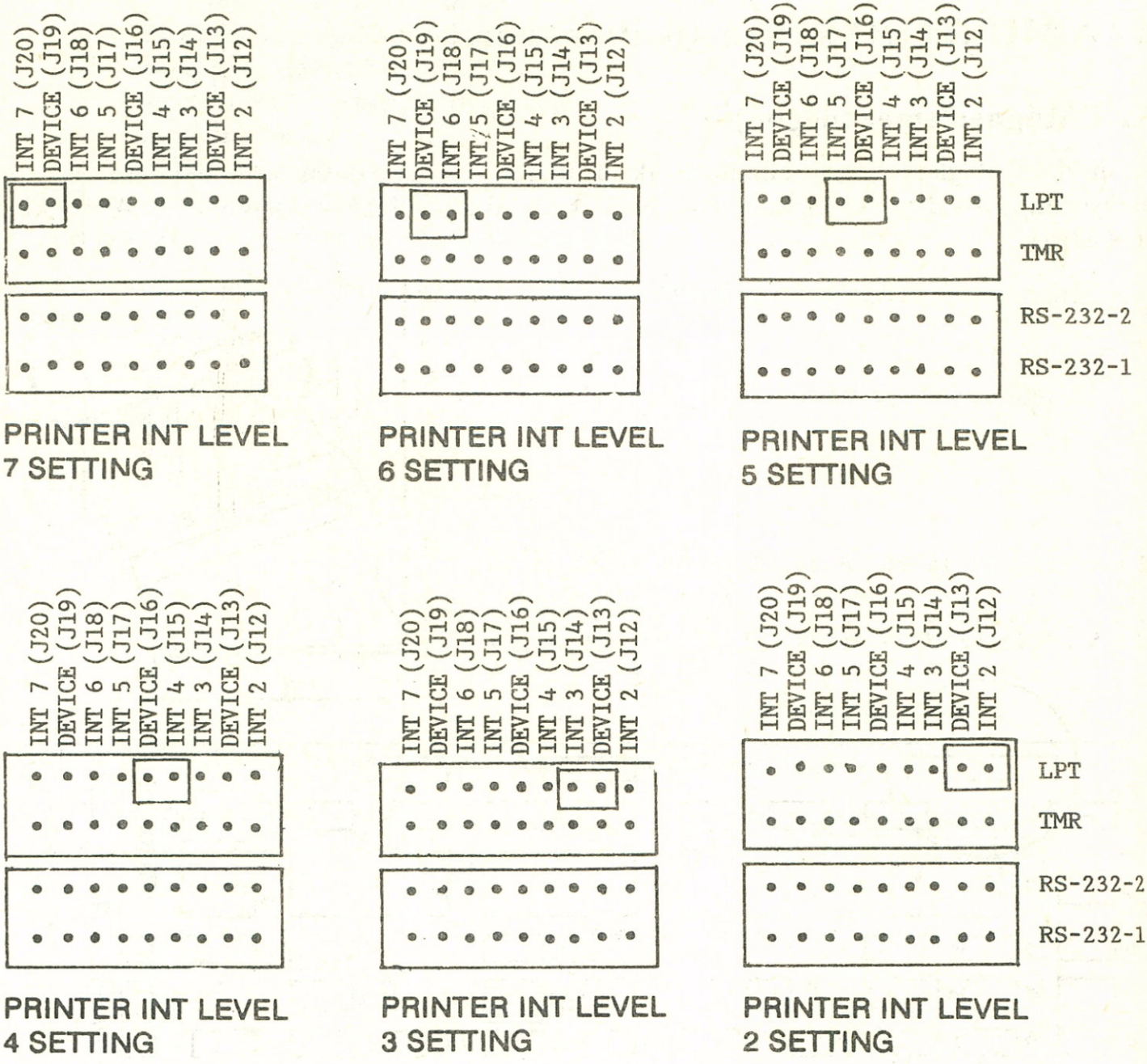
A diszk szalagkábeljénél ellenőrizzük a biztos csatlakoztatást. A kábelen két, 34 aranyérintkezős diszk-csatlakozó van, melyek közül a távolabbi csatlakozik az A diszkhez (6-4 ábra).



6-4 ábra.
A mágneslemez meghajtók csatlakoztatása.

6.2.2 Párhuzamos nyomtatóillesztő

A párhuzamos nyomtatóillesztő printercsatlakozóval van ellátva. A nyomtató engedélyező (JP5) dugóval lehet üzembe helyezni vagy kiiktatni a nyomtatót.
A nyomtató megszakítási szintjének kiválasztása a megszakítás kiválasztó dugók segítségével történik a 6-5. ábra szerint.



6-5 ábra.
A nyomtató megszakítási szintjének kiválasztása.

A nyomtató csatlakozó érintkezőkiosztása

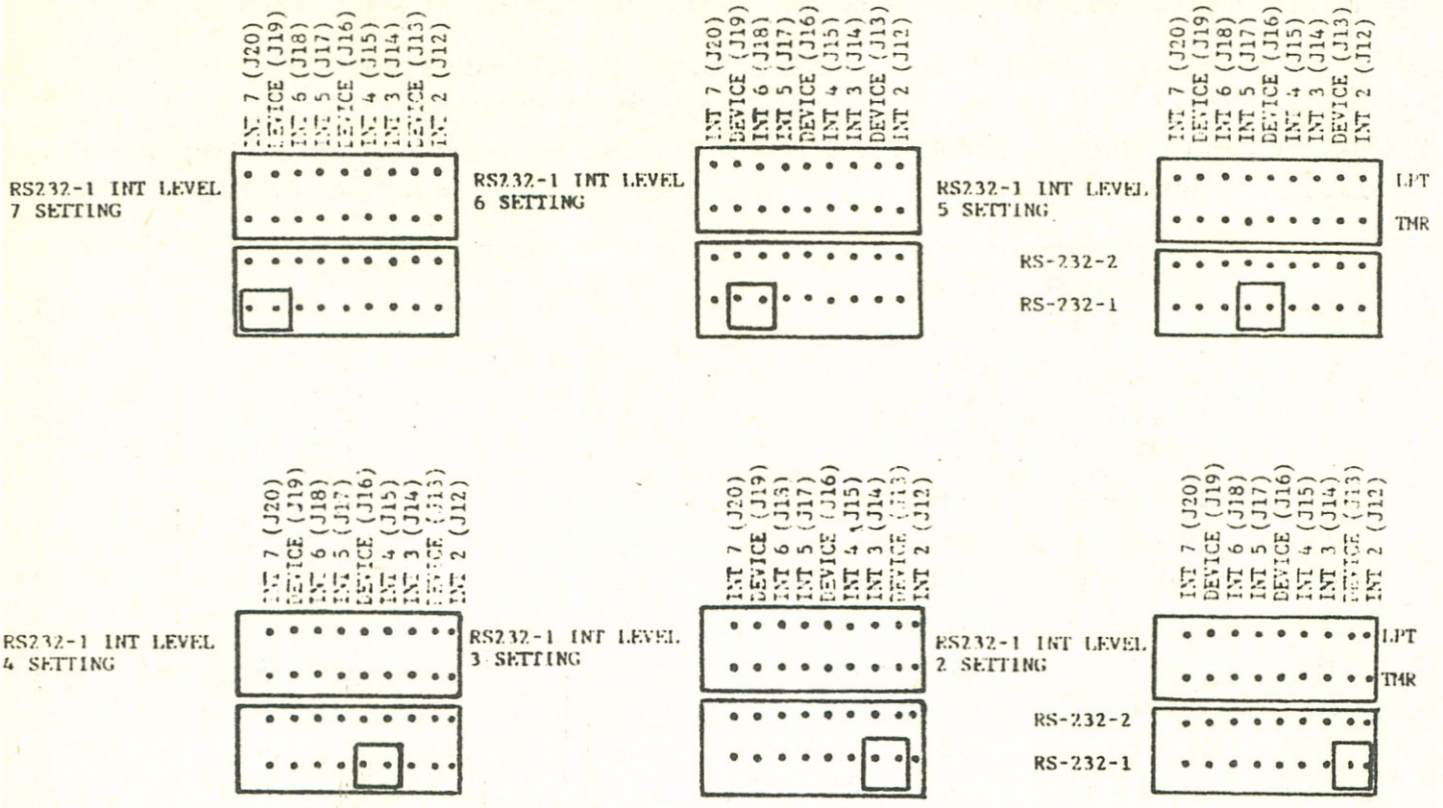
Csatl. szám	Jel neve	Megnevezés	Adatirány adapter	Adatirány nyomtató
1	-STROBE	-kapujel		-->
2	DATA 0	adat-bit		-->
3	DATA 1	adat-bit		-->
4	DATA 2	adat-bit		-->
5	DATA 3	adat-bit		-->
6	DATA 4	adat-bit		-->
7	DATA 5	adat-bit		-->
8	DATA 6	adat-bit		-->
9	DATA 7	adat-bit		-->
10	-ACK	-nyugtázás		<--
11	+BUSY	+foglaltság		<--
12	+PE	+papírhiány		<--
13	+SELECT	+nyomtató kiv.		<--
14	-AUTO. FD	-auto soremelés		-->
15	-ERROR	-hiba		<--
16	-INIT	-inicializálás		-->
17	-SEL IN	-nyomtató kiv.bemenet		-->
18-26	GND	föld		-->

6-6 ábra.
A nyomtató csatlakozó jelkivezetései.

6.3 RS-232 soros illesztő csatlakozás

A soros illesztő az EIA RS-232 szabványnak felel meg. Az RS-232 soros interface programozható, és kielégíti az IBM gépek aszinkron átviteleit megvalósító BIOS függvény hívásokat. Az adatátviteli sebesség 50 és 9600 BAUD (bit/sec) között választható meg.

- A kártyán 2 soros illesztő helyezhető el. Az RS-232-1 illesztő minden elemét tartalmazza minden kártya, az RS-232-2 opcionális.
- Azt, hogy az egyes illesztők melyik kommunikációs csatornához kapcsolódnak, azt a megfelelő dugóval lehet beállítani, de definiálni lehet ezt a géptől is: a COM1 a 3F8h, a COM2 csatorna a 2F8h portról érhető el.
- Az adatátviteli funkció (COM) megszakítási szintjét, a nyomtatóéhoz hasonlóan, a megfelelő dugókkal lehet beállítani a 6-7. ábrának megfelelően. Alaphelyzetben - adatterminál üzemmódban - a COM1 kommunikációs csatorna megszakítási szintje a 4. szint.



6-7 ábra.
A kommunikációs csatorna megszakítási szintjének kiválasztása.

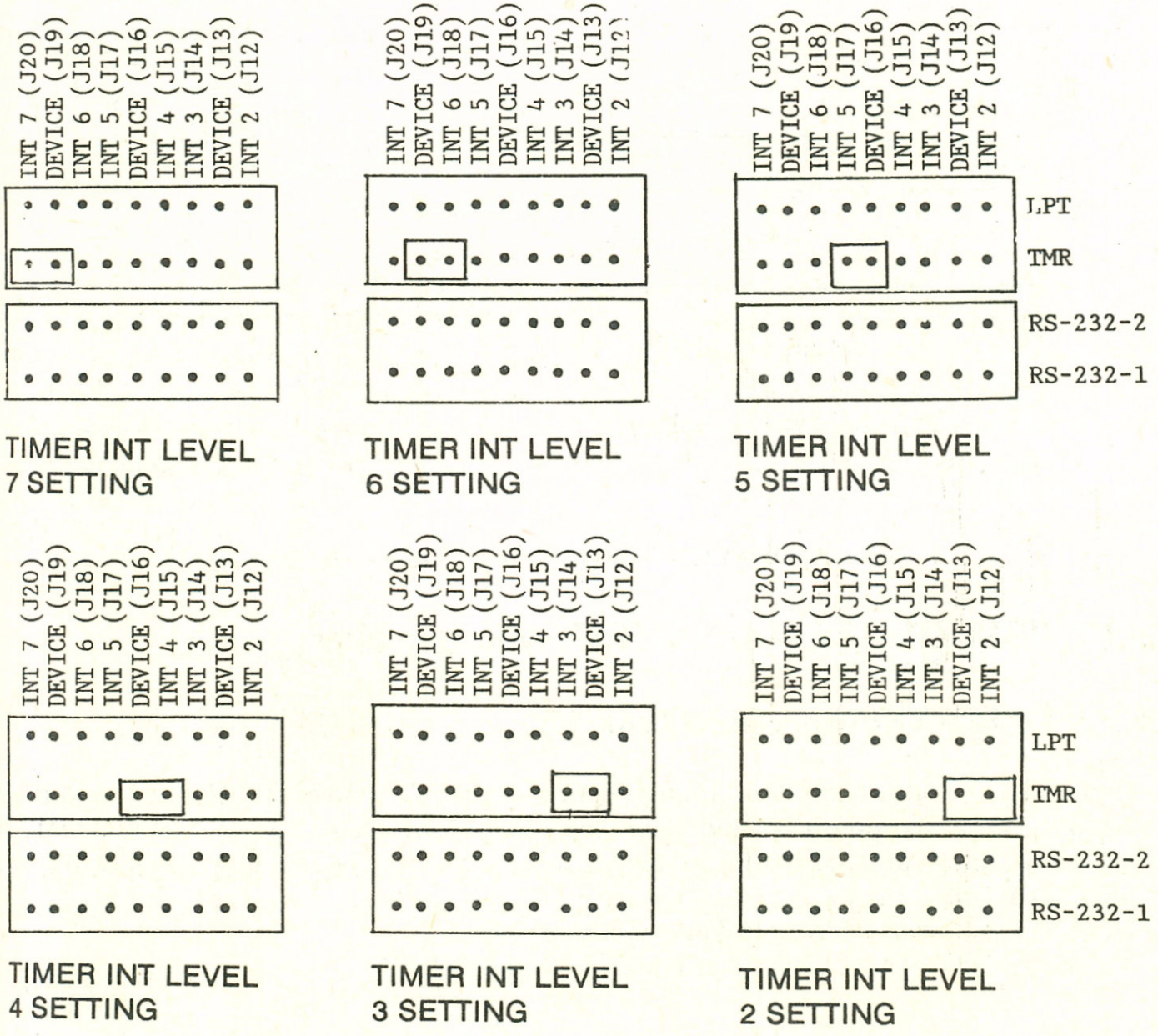
Az RS-232 csatlakozó kiosztása:

Csatl. szám	A jel megnevezése	Adatirány IBM PC	Állomás
1	Védőföld	---	
2	Adat ki	-->	
3	Adat be	<--	
4	Adás kérés	-->	
5	Vételkészség nyugtázó jel	<--	
6	Adatkommunikációs berendezés készenléti jel	<--	
7	Jel föld	---	
20	Adatterminál készenléti jel	-->	
22	Hívás indikátor	<--	

6-8 ábra.
Az RS-232 csatlakozó kivezetései.

6.4 Teleppel ellátott óra/naptár

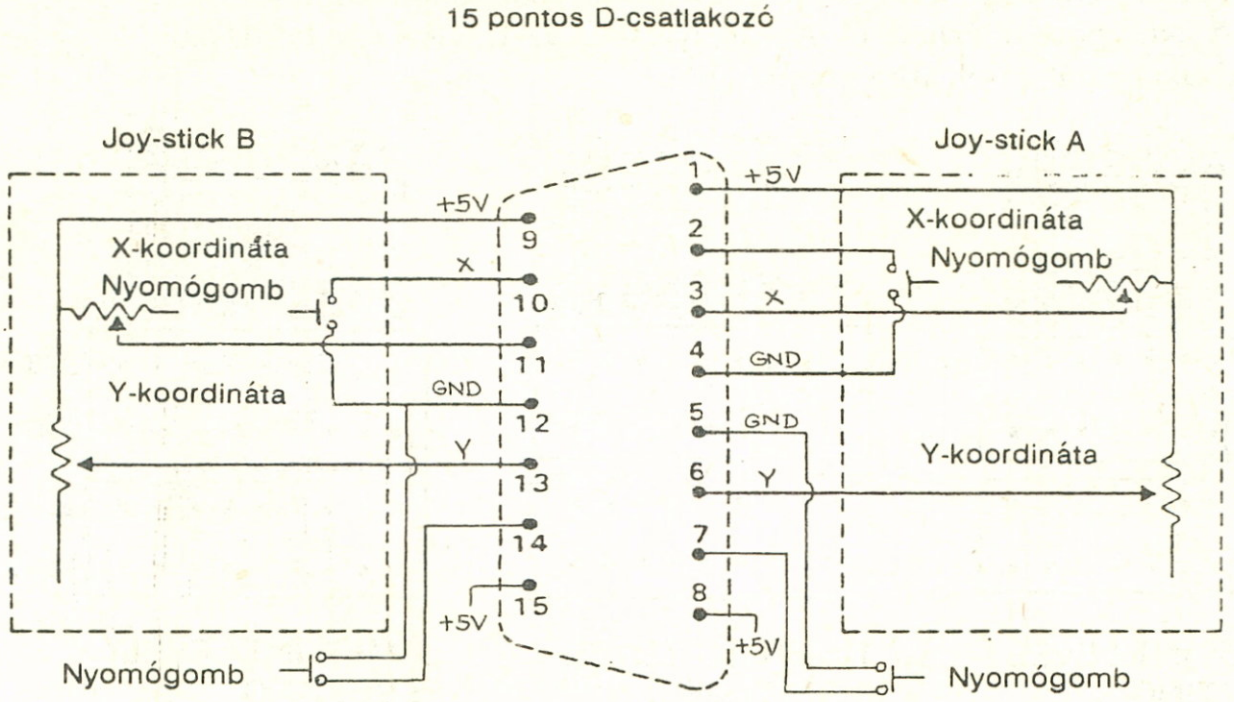
Az óra/naptár táplálása telepről történik, ha a számítógép kikapcsolt állapotban van. Bekapcsolás után a telep nincs terhelve, és a +5V-os tápfeszültségről töltődik.
Az óra/naptárt az MS-DOS LEMEZEN lévő TIMER.COM programmal lehet használni, az aktuális idő és dátum megadásával. Részletesebben lásd az ÓRA/NAPTÁR felhasználói kézikönyvében.
Az óra kiválasztó dugó lehetővé teszi az óra I/O báziscím beállítását. Az 1. ÓRA a 340h, a 2. ÓRA 240h portról érhető el. Alaphelyzetben az 1. ÓRA van kiválasztva. A megszakítási szint kiválasztását a 6-9. ábra szemlélteti.



6-9 ábra.
Az ÓRA megszakítási szintjei.

6.5 Játékvezérlő adapter

A játékvezérlő adapter két botkormány (joy-stick) csatlakoztatását biztosítja. A botkormány két szélső állása 0 -- 100 kohm ellenállás tartományt fog át. Még az IBM PC kompatibilis botkormányoknak is számos változata létezik, és így előfordulhat, hogy a központosító mechanizmust a programhoz kell adjusztálni.
A játékvezérlő botkormány csatlakozó érintkezőinek kiosztását a 6-10 ábra szemlélteti.



6-10 ábra.
A botkormány csatlakozó érintkezőinek kiosztása.

6.6 PC TIMER 1.3 ÓRA/NAPTÁR alkalmazása

6.6.1 A TIMER.COM program funkciói

- Biztosítja az ÓRA/NAPTÁR induló dátum és idő adatainak beállítását. A tölthető telep a gép kikapcsolt állapotában tápfeszültséggel látja el az ÓRA/NAPTÁR áramkört, így az utoljára beállított adatok mindaddig érvényesek, míg egy új beállításra nem kerül sor.
- Biztosítja a dátum és idő adatokat a rendszer számára.
- Tájékoztat az ÓRA/NAPTÁR pillanatnyi állapotáról.

6.6.2 Indítás

A program kétféleképpen indítható:

1) Be kell billentyűzni a TIMER (ill. TIMER/opció) utasítást.

2) Létre kell hozni egy AUTOEXEC.BAT file-t, amely automatikusan indítja a rendszert.

a/ Ha a lemezen nincs AUTOEXEC.BAT file, akkor először billentyűzze be a

COPY CON: AUTOEXEC.BAT

utasítást, nyomja meg az ENTER gombot, majd írja be

TIMER/S [Ctrl + Z]

(vagy, nyomja meg az F6 funkcióbillentyűt) és nyomja meg az ENTER gombot.

Az A > prompt újramegjelenésekor készen van az AUTOEXEC.BAT file.
A képernyő képe:

```
A > COPY CON: AUTOEXEC.BAT
    TIMER/S ^ Z
            1 File(s) Copied
A >
```

A fenti utasítások közül az első parancs sor mondja meg a DOS-nak, hogy másolja be a billentyűzetről bevitt információkat az AUTOEXEC.BAT file-ba. A második parancs-sor viszi be a TIMER/S programot az AUTOEXEC.BAT file-ba. A [Ctrl + Z] vagy az F6 billentyűk lenyomásakor létrejön a file olyan formában, hogy egyéb keyboard információk nem kerülnek rögzítésre.
Ha már létrehoztuk az AUTOEXEC.BAT file-t, akkor a rendszer újraindításakor a TIMER/S az AUTOEXEC.BAT file-ban átviszi az ÓRA/NAPTÁR idő információit a rendszerbe. Ilyenkor a képernyő képe:

```
A > TIMER/S

** PC TIMER 1.3 **
Current time is XX:XX:XX
Current date is XXX XX-XX-XX
```

b/ Ha az AUTOEXEC.BAT file már megvan a lemezen, akkor hozzá kell adni egy sort a file-hoz. Ehhez szükség van a lemezen az EDLIN sorszerkesztő programra.

Írja be:

EDLIN AUTOEXEC.BAT

Nyomja meg az ENTER billentyűt.

Írja be:

li

Nyomja meg az ENTER billentyűt.

Írja be:

TIMER/S

Nyomja meg az ENTER billentyűt.

Írja be:

[Ctrl] C
E

Nyomja meg az ENTER billentyűt.

Az A > prompt újramegjelenésekor létrejött a bővített AUTOEXEC.BAT file, benne a TIMER/S programmal.

A képernyő képe:

```
A > EDLIN AUTOEXEC.BAT
*li
 i:*TIMER/S
 2:* ^C
*E

A >
```

6.6.3 TIMER utasítások

TIMER/? Segítő menü, amely tartalmazza a TIMER utasításokat:

```
** PC TIMER 1.3 **

Timer Help Menu:

TIMER           Read current time & date
                (Aktuális idő és dátum kiolvasása)
TIMER/I         Initial setting current time & date to system clock
                (Az aktuális alapadatok beírása a rendszer-órába)
TIMER/S         Set current time & date to system clock
                (Az aktuális idő és dátum beírása az órába)
TIIMER/?        Help menu (Segítő menü)
TIMER FORMAT  HOUR:MINUTE:SECOND
                (Every item have two digits)
Idő Formátum    Óra:Perc:Másodperc
                (Minden adat két számjegyű)
```

```
DATE FORMAT  MONTH - DAY - YEAR
                (Every item have two digits)
Dátum formátum  Hónap - Nap - Év
                (Minden adat két számjegyű)
```

Tekintsük át az egyes utasításokat a menüből.

TIMER Kiolvassa az aktuális adatokat, és kiírja a képernyőre:

```
** PC TIMER 1.3 **
Current time is XX:XX:XX
Current date is XXX XX-XX-XX
```

TIMER/I Az ÓRA/NAPTÁR alapadatainak beállítása a MULTI I/O kártyán. Az első képen kiírja az aktuális időt, és biztosítja az idő és dátum adatok bevitelét. A program ellenőrzi, hogy az adatokat megfelelő formátumban vittük-e be. Ha a formátum helyes, akkor a TIMER program beírja azokat az ÓRA/NAPTÁR-ba, ellenkező esetben hibaüzenetet küld, hogy ismételjük meg a bevitelt. Ha nem akarunk változtatni az adatokon, akkor csak meg kell nyomni az ENTER billentyűt. A képernyő kép a következő:

```
** PC TIMER 1.3 **
Current time is XX:XX:XX
Enter new time:
```

A helyes adatok bevitele után a képernyő képe:

```
** PC TIMER 1.3 **
Current time is XX:XX:XX
Current date is XXX XX-XX-XX
```

```
TIME FORMAT: XX:XX:XX
               /   /   /
            óra perc másodperc
```

- a TIMER program csak nyolc karaktert fogad el, adatonként két számjegyet és az elválasztó karaktereket;
- az ÓRA tartomány 0-23;
- a PERC tartomány 0-59;
- a MÁSODPERC tartomány 0-59.

DATE FORMAT:HÓNAP-NAP-ÉV

- a HÓNAP tartomány 1-12;
- a NAP tartomány 1-31;
- az ÉV tartomány 0-99 (a 0-79 közötti értékeket a program 2000-2079-ként, a 80-99 közöttieket pedig 1980-1999-ként értelmezi!)
- betűkaraktereket nem használhatunk!
- a TIMER program csak nyolc karaktert fogad el, adatonként két-két számjegyet és az elválasztó karaktereket.

Ha az időt, vagy a dátumot hibásan vittük be, akkor a hibajelzés képe:

```
Invalid time (Hibás időadat)
Enter new time: (Vigyen be új adatot)
```

Illetve:

```
Invalid date (Hibás dátumadat)
Enter new date: (Vigyen be új adatot)
```

Az ENTER billentyű idő előtti megnyomásával megszakíthatja az adatbevitelt. Ez esetben az eredeti érték megmarad.

TIMER/S Ez a képernyő utasítás beállítja a rendszer-órát az aktuális idő és dátum értékére. Egyenértékű azzal, amikor az AUTOEXEC.BAT file automatikusan beállítja a rendszer-órát indításkor illetve újraindításkor.

7. RÉSZ

Floppy-k és Winchester-ek.

7.1 Merevlemezes tárolók

7.2 IBM merevlemezes meghajtó adapter

7.3 IBM PC/AT Winchester és floppy disk adapter

7.1 Merevlemezes tárolók

A személyi számítógépek fejlődésében jelentős előrelépést jelentett a merevlemezes tárolók egyik típusának, a *Winchester* diszkeknek a megjelenése. Bár a floppy diszkek segítségével a szövegszerkesztés, a program- és adattárolás gondtalanul megvalósítható volt, az olyan feladatok azonban, mint nagyobb adatbázisok kezelése, *compiler*-ek használata már sok kényelmetlenséget okozott a tárolási helyhiány miatt. A diszkettek gyakori cseréje nagyon kényelmetlenné tette a munkát. Ezt a problémát végleg megoldotta a *Winchester* diszkek megjelenése.

A merevlemezes diszkek lemezei olyan merev fémlapokból készülnek, melyekre fémoxidrétegeket visznek fel. Ezek a lemezek tipikusan 3600 ford/perc fordulatszámmal forognak, tízszer gyorsabban, mint a floppy lemezek. Az író/olvasó fejek nem érintik a lemezt, távolságuk a lemeztől néhányszor tíz mikron. Még a legkisebb porszem is tönkreteheti a lemez felületét, ha az író/olvasó fej alá kerül. Ezért a merevlemezes tárolókat hermetikusan zárt kivitelben készítik. Olyan légszűrő rendszert alkalmaznak, amely ultra-tisztaságú levegőcserét biztosít egy néhány tized mikron átmérőjű furaton keresztül a belső diszk szerelvények és a külvilág között.

A *Winchester* típusú tárolók kapacitása 5 és 400 Mbyte között változhat. Az 5,25" átmérőjű diszk lemezek által meghatározott külső méretek kb. 40 Mbyte kapacitásig nem nagyobbak egy 5,25"-os floppy meghajtó méreteinél, a nagyobb kapacitásúak magassága nagyobb, és méretük közelítőleg két egymás fölé helyezett 5,25"-os floppy meghajtó méreteivel egyeznek meg.

Az IBM XT-hez csatolt első *Winchester* diszkek 10 Mbyte-osak voltak. Jelenleg a leggyakrabban használt változatok 20, 30, 40, 60 Mbyte-os formattált kapacitással rendelkeznek. Néhány típus jellemző adatait felsorolnánk:

a.) Az IBM 10 Mbyte-os merevlemezes tárolója
(Az első, XT-hez kapcsolt *Winchester*)

A lemezegység két darab fixen beépített 5.25 inch átmérőjű mágneslemezt tartalmaz. Egy lemezfelületre 306 sávot lehet felvinni, mind a négy felülethez tartozik egy-egy mozgatható író/olvasófej. Formázás után az egység kapacitása 10 Mbyte (1224 sáv, 17 szektor/sáv, 512 byte/szektor).

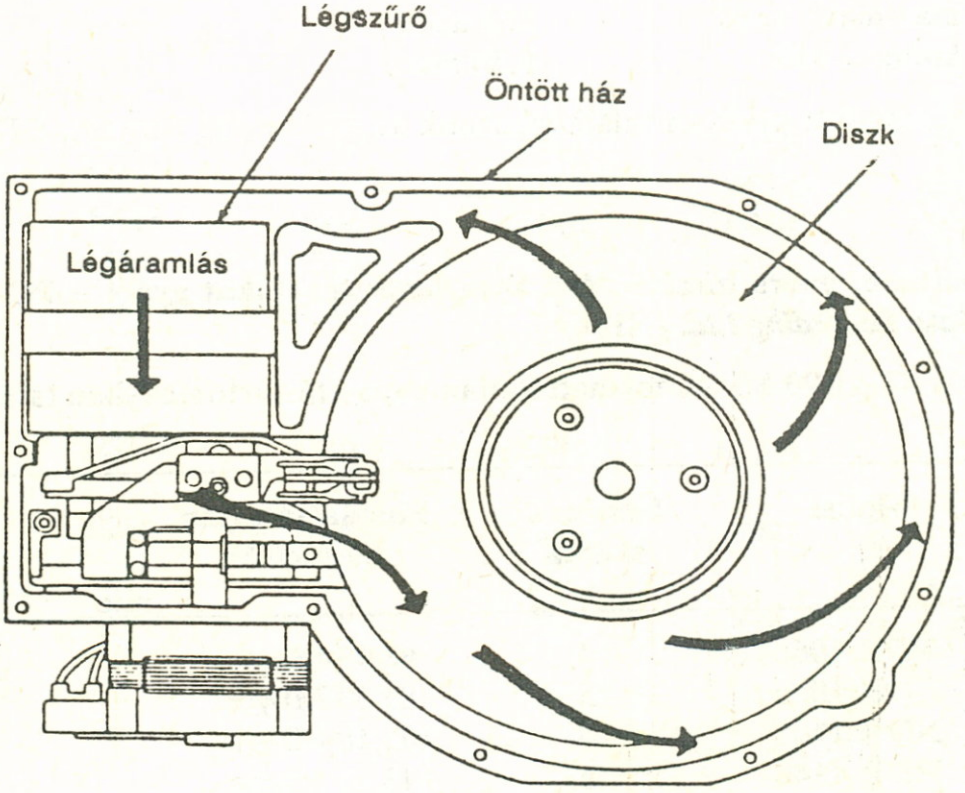
A fejek, a meghajtószerkezet és a lemezek egy ütés - és rázkódásálló fémtokban nyertek elhelyezést. A tokon belül található levegőcserélő rendszer 0.3 mikron lyuknagyságú szűrőn keresztül szívja be a levegőt. A tiszta levegő megfelelő áramoltatása biztosítja az egység hűtését. A lemezforgatómotor és a fejmozgató léptetőmotor hőszigetelő burkolatban van. Így hosszú idejű működés után sem emelkedik számottevően a tokon belüli hőmérséklet. Ezenfelül az egység bekapcsolás után azonnal működőképes, nincs szükség a hőegyensúly beálltának megvárására. (7-1 ábra)

Mechanikus és elektromos specifikációk

Sávok száma:	1224
Elérési idő:	3 msec (sáv-váltási idő)
Lemezforgási sebesség:	3600 fordulat/perc +/- 1%

Adatátviteli sebesség: 5 Mbit/sec
Adatfelírási mód: MFM
Meghibásodás:

- élettartam: kb. 5 év (8000 óra MTF)
- soft error: 1 hiba 10^{10} bitenként
- hard error: 1 hiba 10^{12} bitenként
- fejpozícionálás: 1 pozícionálási hiba 10^{6} pozícionálásonként.



7-1 ábra.
A merevlemezes tároló felépítése.

b.) Seagate ST225 20 Mbyte-os merevlemezes tároló

Ez az egyik leggyakrabban használt típus. Az elektromos és mechanikus specifikációit röviden a következő felsorolás foglalja össze:

Cilinderek száma:	614
Sávok száma:	2460
Diszk lemezek száma:	2
Fejek száma:	4
Adatfelírási mód:	MFM
Lemezforgási sebesség:	3600 ford/perc +/- 1%

Formattálatlan kapacitás: 25,62 Mbyte
Formattálatlan kapacitás sávonként: 10416 byte

Formattált kapacitás: 20,15 Mbyte
Formattált kapacitás
sávonként: 8192 byte
szektoronként: 256 byte
Szektorok száma sávonként: 32

Hozzáférési idők
sáv-váltás: 3 msec
átlagos hozzáférési idő: 65 msec
maximális 'seek': 185 msec
latencia idő: 8,33 msec

Az ST225 típus mellett gyakran találkozhatunk az ST238 (31 Mbyte), ST251 (43 Mbyte) típusokkal is.

c.) Nagykapacitású, gyors hozzáférésű *Winchester* tárolókat gyárt a PC-k részére pl. a *Newbury Data Recording Ltd.* (NDR)

Gyártmányaik a 67 -- 190 Mbyte formattálatlan kapacitástartományban találhatók:

Model	Lemezek száma	Formattálatlan kapacitás
NDR 1065	4	66,93 Mbyte
NDR 1085	5	85,33 Mbyte
NDR 1105	6	105,18 Mbyte
NDR 1140	8	143,43 Mbyte
NDR 2190	8	191,24 Mbyte

Elektromos és mechanikus jellemzőiket mutatja be a következő felsorolás:

Jellemzők	NDR 1065	NDR 1085	NDR 1105	NDR 1140	NDR 2190
Tárolási konfiguráció					
Cilinderek száma	918	1024	918	918	1224
Fejek száma	7	8	11	15	15
Lemezek száma	4	5	6	8	8

Jellemzők	NDR 1065	NDR 1085	NDR 1105	NDR 1140	NDR 2190
Formattálatlan kapacitás					
Mbyte/drive	66,93	85,33	105,18	143,43	191,24
Byte/cilinder	72912	83328	114576	156240	156240
Byte/sáv	10416	10416	10416	10416	10416
Formattált kapacitás (33 szektor/sáv)					
Mbyte/drive	54,3	69,2	85,3	116,3	155,1
Byte/cilinder	59136	67584	92928	126720	126720
Byte/sáv	8448	8448	8448	8448	8448
Byte/szektor	256	256	256	256	256
Szektor/sáv	33	.33	33	33	33
Adatfelírási paraméterek					
Felvételi mód	MFM	MFM	MFM	MFM	MFM
Bitsűrűség (bit/inch)	9280	9934	9280	9280	11155
Sávsűrűség (sáv/inch)	995	995	995	995	1022
Adatátviteli sebesség (MHz)	5	5	5	5	5
Hozzáférési idők (msec)					
Sáv-váltás	3,5	3,5	3,5	3,5	3,5
Átlagos hozzáférési idő	25	26	25	25	28
Maximális 'seek' (beállási idővel)	45	48	45	45	52
Forgatási jellemzők					
Forgási sebesség	-----	3600 ford/perc +/- 0,1 %			-----
Átlagos latencia idő		-----	8,33 msec	-----	
Motor felfutási idő		-----	20 sec	-----	
Motor megállási idő		-----	30 sec	-----	

Jellemzők	NDR 1065	NDR 1085	NDR 1105	NDR 1140	NDR 2190
Megbízhatóság					
Élettartam	----- 16000 óra bekapcsolt állapotban ----- (kb 8 év, naponta 6-7 órás üzemmel számolva)				
Soft error	----- 1 hiba 10^{10} bitenként -----				
Hard error	----- 1 hiba 10^{12} bitenként -----				
Fejpozícionálás	----- 1 pozícionálási hiba ----- 10^{6} pozícionálásonként				

A felsorolt merevlemezes tárolóegységek az MFM (Modified Frequency Modulation) eljárást használják adatfelírási módként. Ez a felírási mód a soros adatjelek hatékony kódolásával állítja elő a lemezre közvetlenül felvitt információt. Az eljárás lényege a következő: bármely sorosan érkező adatmennyiséget bit-cellákra lehet osztani, amelyek rövid egymás utáni időintervallumok, ahol minden intervallumhoz egy bit információ van hozzárendelve. A kódolás szabályai :

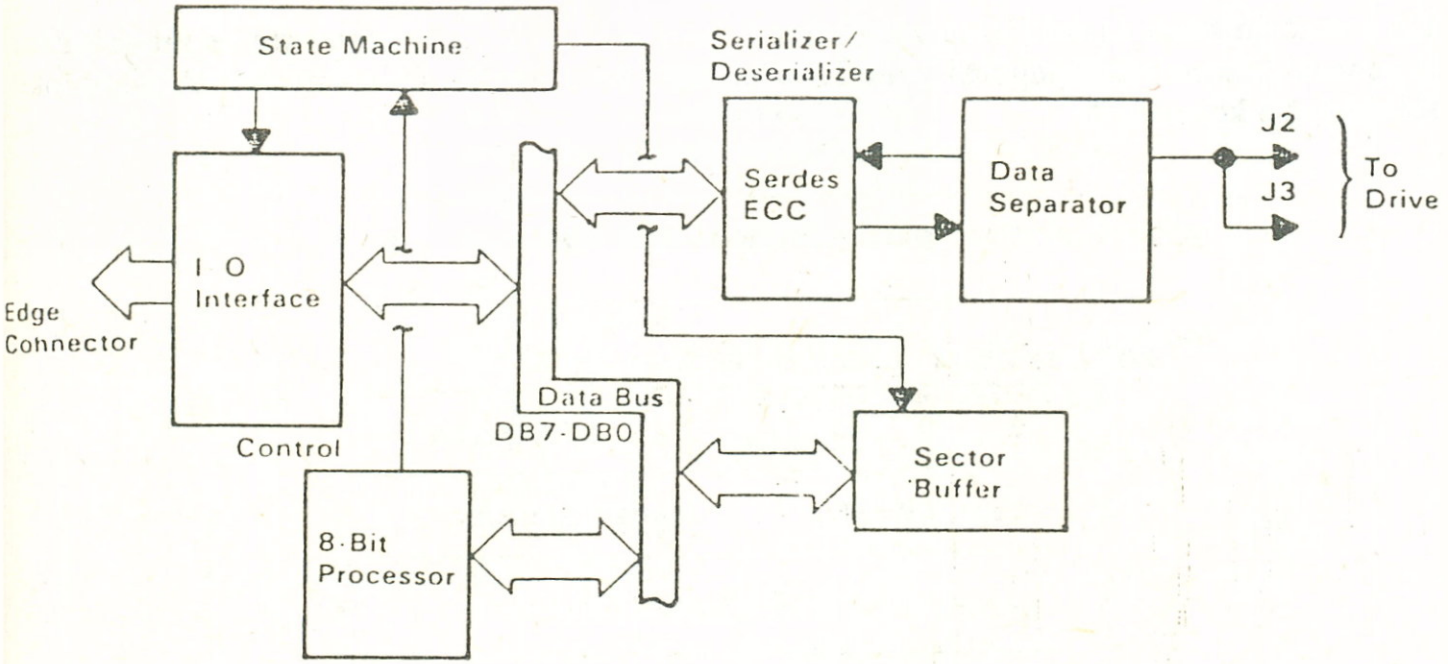
a.) A logikai 1 értékű bitet egy olyan impulzus jelképezi, amelynek az első, induló éle a cella közepén helyezkedik el.

b.) A logikai 0 értékű bitet egy olyan impulzus jelképezi, amelynek az első, induló éle a cella kezdetén helyezkedik el. Ellenben, ha az előző bit-cella logikai 1 értékű volt, és a jelen bit 0, ezt a bitet egy impulzus nélküli, üres cella fogja reprezentálni.

A lemezegység soha nem értelmezi az MFM adatokat, ez a diszk vezérlő adapter feladata. Az MFM kódolási eljárást használják a leggyakrabban a floppy lemezek adatfelírására is.

7.2 IBM merevlemezes meghajtó adapter

Az adapter illesztőként szolgál a számítógép és a merevlemezes tároló között. Egy adapter két lemezegységet képes kiszolgálni. A kártya a rendszermemóriával DMA-n keresztül tart kapcsolatot az adatrekordok átvitele céljából. A processzorral megszakításszinten kommunikál. Az adapter 32 bites ECC kódot használ, ennek segítségével képes felismerni és kijavítani max. 11 bit hosszúságú hibacsomókat is. Az egység blokkdiagramja 7-2 ábrán látható.

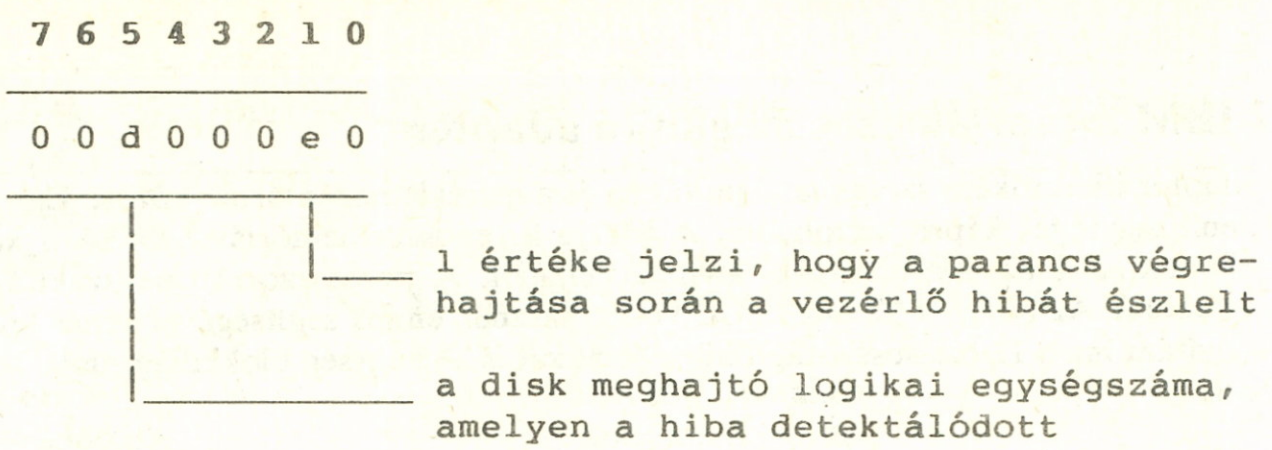


7-2 ábra.
Az IBM merevlemezes meghajtó blokkdiagramja.

7.2.1 A lemezvezérlő programozása

7.2.1.1 Státuszregiszter

A vezérlő minden vett parancs végrehajtása után egy státuszinformációt bocsát a processzor rendelkezésére. Ha a megszakításkérés engedélyezve van, a vezérlő egy IRQ jel generálásával jelzi, hogy kész a státuszbyte küldésére. A státuszregiszter tartalmának a rendszerbuszra kapuzása után történik meg a vezérlő BUSY jelének visszavétele. A státusz felépítése a következő:



7.2.1.2 Hibablokk

Ha a vezérlő hibát észlel (státuszregiszter 1. bitje = 1), akkor egy négy byte-os hibablokkot is generál, aminek segítségével a rendszer képes a hiba pontos azonosítására. A hibablokk felépítése a következő.

Bit	7	6	5	4	3	2	1	0
Byte0	AV	0	hibatípus			hibakód		
1	0	0	d		fej sorszáma			
2	sáv (8-10)				szektorszám			
3	sáv (0-7)							

d: meghajtó logikai egységszáma

AV: ha ez a bit logikai 1. akkor a cím érvényes és a hiba helyét az 1,2 és 3 byte-okban található sáv, szektor és fejsorszám azonosítja. Ha 0, a lemezcímet figyelmen kívül lehet hagyni, mert a hiba nem egy szektorra lokalizálódik.

A következő táblázat a lehetséges hibatípusokat ill. hibakódokat tartalmazza.

Típus	kód	hiba
0	0	nincs hiba
0	1	nincs index-jel
0	2	a meghajtó nem szolgáltat "keresés kész" jelet keresés után
0	3	lemezírás hiba

Típus	kód	hiba
0	4	egy meghajtó kiválasztása után az nem ad READY jelet
0	5	nem használt
0	6	a meghajtó nem generál "0.sáv" jelet annak elérésekor
0	7	nem használt
0	8	A meghajtó még nem érte el a keresett szektort. Ezt az információt kapjuk a Test Drive Ready parancs után, ha a meghajtó még dolgozik.
1	0	ID olvasás hiba: a vezérlő ECC hibát észlelt a lemez azonosítómezőjében
1	1	adathiba: olvasás során javíthatatlan ECC hiba
1	2	a vezérlő nem érzékeli a lemez Address Mark jelét
1	3	nem használt
1	4	A vezérlő nem találja a megcímzett szektort
1	5	Keresési hiba: a meghajtó olyan sávra állt, amely nem a megadott sáv, vagy rossz fejet választott ki.
1	6	nem használt
1	7	nem használt
1	8	a vezérlő javítható ECC hibát észlelt
1	9	a vezérlő "hibás sáv" jelzést észlelt
2	0	a vezérlőnek adott parancs hibás, vagy nem értelmezhető
2	1	a vezérlőnek küldött lemezcím hibás, nem létező fej/sáv/szektor
3	0	a vezérlő hibát észlelt saját szektor-puffer RAM-jában
3	1	a vezérlő saját (ROM-ban levő) programjában ellenőrzőösszeg hibát észlelt
3	2	a vezérlő belső tesztje során saját ECC hibapolinom generátorát hibásnak találta.

7.2.1.3 Adatregiszter

A processzor a vezérlő működését egy 6 byte méretű parancsblokk (Device Controll Block - DCB) segítségével szabályozza. A parancsblokk felépítése a következő:

bit	7	6	5	4	3	2	1	0
byte0	parancsosztály				parancskód			
1	0	0	d	fej sorszám				
2	sáv high			szektorszám				
3	sáv low							
4	blokkszámláló vagy interleave							
5	r	a	0	0	0	s	s	s

d: Meghajtó logikai egységszám.

r: Hiba esetén az ismétlések számát határozza meg. Ha értéke 0, a vezérlő hiba esetén még négyszer megpróbálja az adott parancsot végrehajtani. Ha 1, akkor hiba esetén nincs újrapróbálkozás.

a: Ha 0, akkor a vezérlő ECC hiba esetén automatikusan újraolvassa a hibásnak talált adatblokkot. Ha 1, nincs újraolvasás.

sss: A meghajtó típusát, az alkalmazható lépésidőt jelöli az alábbiak szerint:

2 1 0 s s s	
0 0 0	3ms/lépés (default)
0 0 1	nem használt
0 1 0	nem használt
0 1 1	nem használt
1 0 0	200 us/l
1 0 1	70 us/l
1 1 0	3 ms/l
1 1 1	3 ms/l

7.2.1.4 Parancstáblázat

Meghajtó teszt
00000000
00dxxxxx

Rekalibrálás

00000001
00dxxxxx
r0000sss

Státusz

00000011
00dxxxxx

Lemez formázás

00000100
00d < fej >
< ch > 000000
< cylinder low >
000 < interleave >
r0000sss

Olvasás ellenőrzés

00000101
00d < fej >
< ch > < szektor >
< cylinder low >
< blokkszámláló >
ra000sss

Sáv formázása

00000110
00d < fej >
< ch > 000000
< cylinder low >
000 < interleave >
r0000sss

Hibás sáv formázása

00000111
00d < fej >
< ch > 000000
< cylinder low >
000 < interleave >
r0000sss

Olvasás

00001000
00d < fej >
< ch > < szektor >
< cylinder low >
ra000sss

Írás

00001010
00d < fej >
< ch > < szektor >
< cylinder low >
< blokkszámláló >
r0000sss

Fejmozgatás (seek)

00001011
00d < fej >
< ch > 000000
< cylinder low >
xxxxxxxx
r0000sss

Lemezegység jellemzőinek alaphelyzetbe állítása

00001100
* 8 kiegészítő byte:
maximális cilinder szám (2 byte)
a fejek maximális száma (1 byte)
a csökkentett áramú írás
kezdő cilinder száma (2 byte)
az írás előkompenzáció
kezdő cilinder száma (2 byte)
maximális ECC adat burst hosszúság (1 byte)

ECC hibacsomó hossz olvasás

00001101

Olvasás a szektor-pufferből

00001110

Írás a szektor-pufferba

00001111

RAM diagnosztika

11100000

Meghajtó diagnosztika

11100011
00dxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
r0000sss

Vezérlő belső diagnosztika

11100100

Olvasás 512 byte + 4 byte ECC

11100101
00d < fej >
< ch > < szektor >
< cylinder low >
< blokkszámláló >
r0000sss
** A parancs eredményeként
512 kiolvasott byte-ot (1 szektor)
és 4 byte ECC adatot kapunk

Írás 512 byte + 4 byte ECC

11100110
00d < fej >
< ch > < szektor >
< cylinder low >
< blokkszámláló >
r0000sss
*** A parancs blokk után 512 adatbyte-ot
és 4 byte ECC információt kell elhelyezni (1 szektor)

Az adatregiszterbe írható parancsok táblázatában az alábbi jelöléseket ill. rövidítéseket alkalmaztuk:

d: meghajtó logikai egységszáma (0 v. 1)(1 BIT)

x: tetszőleges értékű bit (don't care)

r: ismétlés vezérlés. Részletes leírást lásd ADATREGISZTER címszó alatt. (1 BIT)

sss: meghajtó típusa. Részletes leírást lásd ADATREGISZTER címszó alatt. (3 BIT)

fej: azon író/olvasó fej sorszáma, amelyre a művelet vonatkozik. (5 BIT)

ch: a cilindert (sávot) azonosító 10 bites szám legmagasabb helyiértékű két bitje (cylinder high) (2 BIT)

cylinder low: a cilindert (sávot) azonosító 10 bites szám alsó 8 bitje (8 BIT)

szektor: azt a szektort azonosítja egy sávon belül, amelyre a művelet vonatkozik (6 BIT)

blokkszámláló: megadja, hogy az adott művelet hány egymást követő adatblokkra vonatkozik (8 BIT)

a: ismétlés-vezérlés ECC hiba esetén. Részletesen lásd ADATREGISZTER címszónál.

7.3 IBM PC/AT Winchester és floppy-disk adapter

A leírt illesztő az általánosan használt szabvány kontrollert írja le. Az egység két darab merev- és két darab hajlékonylemezes meghajtót képes kezelni. Lehetővé teszi egy Winchester és egy floppy egyidejű működtetését is.

A vezérlő összesen 13 darab írható/olvasható regiszterrel rendelkezik, ezek: egy darab 16 bites adatregiszter, 7 darab 8 bites parancsregiszter (ezek alkotják a parancsregiszter blokkot) és 5 darab 8 bites vezérlő ill. státuszregiszter (státuszregiszterblokk).

Minden egyes regiszter egy elsődleges és egy másodlagos I/O címmel is rendelkezik. Azt, hogy a regiszterek az elsődleges vagy a másodlagos címeken érhetők el, egy jumperrel lehet beállítani. Így elméletileg két vezérlő is lehetne egy gépben, de a másodlagos címeket a DOS nem támogatja. A regiszterekhez rendelt I/O címek az alábbiak:

I/O cím elsődleges	I/O cím másodlagos	olvasás	írás
		ADATREGISZTER	
1F0	170	adatregiszter 16 bit	adatregiszter 16 bit
		PARANCSREGISZTER BLOKK	
1F1	171	hibaregiszter	írás pre-comp
1F2	172	blokkszámláló	blokkszámláló
1F3	173	szektorszám	szektorszám
1F4	174	cylinder low	cylinder low
1F5	175	cylinder high	cylinder high
1F6	176	SDH regiszter	SDH regiszter
1F7	177	státuszregiszter	parancsregiszter
		STÁTUSZREGISZTER BLOKK	
3F2	372	---	digitális kimeneti regiszter
3F4	374	floppy regiszter	floppy státuszreg.
3F5	375	floppy adatregiszter	floppy adatregiszter
3F6	376	másodlagos státusz	merevlemez státusz
3F7	377	digitális bemeneti regiszter	floppy vezérlő-regiszter

A parancsregiszterblokk felépítése a következő:

REGISZTERNÉV	7	6	5	4	3	2	1	0
írás pre-comp	((cilinderszám) DIV 4)							
hiba	BB	ECC	0	ID	0	AC	TK	DM
blokkszámláló	(blokkok száma)							
szektorszám	(szektor azonosító szám)							
cylinder low	(cilinder <sáv> szám alsó 8 bit)							
cylinder high	0	0	0	0	0	0	C9	C8
SHD	1	0	1	DS	HS3	HS2	HS1	HS0
parancs	(parancskód)							
státusz	BSY	RDY	WF	SC	DRQ	CRD	IDX	ERR

A státuszregiszter bitkiosztása az alábbi:

REGISZTERNÉV	7	6	5	4	3	2	1	0
digitális kimenet	x	x	MOEN2	MOEN1	FDHA	FRST	x	FDSEL EN
floppy státusz	RQM	DIO	EXM	CB	0	0	D1B	D0B
másodlagos státusz	BSY	RDY	WFT	SKC	DRQ	CRD	IDX	ERR
merevlemez státusz	0	0	0	0	HS3	EN RST	IEN	0
digitális bemenet	DCHG	WTG	HS3	HS2	HS1	HS0	DS2	DS1
floppy vezérlés							SDB1	SDB0

Az előző két táblázatban felsorolt egyes bitek jelentései a következők:

MOEN2

MOEN1: MOTOR ENABLE. A floppy meghajtók motorforgását vezérli. 0-ra állítva az adott meghajtó motorja kikapcsolódik és ezután a meghajtó már nem választható ki.

FDMAEN: FLOPPY DISK INTERRUPT & DMA ENABLE. A bit 0-ra állítása letiltja a vezérlők megszakítás ill. DMA-kérését.

FRST: 0-ra állítva alaphelyzetbe állítódik a floppy-vezérlő. A beállási idő 3,5 usec.

FDSEL: FLOPPY DISK SELECT. 0 értéke az A, 1 értéke a B meghajtót választja ki.

RQM: REQUEST FOR MASTER. 1 értéke jelzi, hogy a floppy adatregiszter kész az adatforgalomra.

DIO: DATA I/O. Adatforgalom irány-kiválasztó bit. 0: géptől a vezérlő felé, 1: vezérlőtől a gép felé.

EXM: EXECUTION MODE. Ha ez a bit 1, a vezérlő nem DMA módban működik.

CB: írás/olvasás műveletek ideje alatt ez a bit 1, egyébként 0.

D1B: 1, ha a B meghajtó SEEK műveletet hajt végre.

D0B: 1, ha az A meghajtó SEEK műveletet hajt végre.

BSY: BUSY. Jelzi, hogy a vezérlő foglalt.

RDY: READY. Jelzi, hogy a vezérlő befejezett egy műveletet és kész egy másik parancs fogadására.

WF: WRITE FAULT. Írási hiba a kiválasztott egységen.

SC: SEEK COMPLETE. A kiválasztott egység végrehajtotta az előzőleg kiadott SEEK műveletet.

DRQ: DATA REQUEST. Adatforgalom kérés bit.

CRD: CORRECTED DATA. A vezérlő adathibát észlelt, amelyet sikeresen kijavított.

IDX: INDEX. A kiválasztott egység indexpulzust észlelt.

ERR: ERROR. A vezérlő hibát észlelt.

HS3EN: 1 értéke a HS3, 0 érték az RWC bitet engedélyezi.

RST: RESET. Alaphelyzetbe állítás. A bitnek minimum 5,0 μsec-ig, 1 értékűnek kell lennie a helyes resetciklus végrehajtása érdekében.

IEN: INTERRUPT ENABLE. Engedélyezi vagy tiltja a megszakításkérést.

DCHG: DISKETTE CHANGE. 1 értéke jelzi, hogy a floppy egységben nincs lemez ill. az egység ajtaja nyitott.

C9,C8: a cilinder (sáv) szám legmagasabb helyiértékű 2 bitje.

A vezérlőnek nyolc különböző parancsot adhatunk. Minden parancs vétele után a vezérlő ellenőrzi a parancs paramétereit, és ha a parancsregiszterblokk tartalmát helyesnek találja, megkísérli azt végrehajtani. Pozicionálja a kiválasztott egység író/olvasófejeit, végrehajtja a megfelelő adatforgalmat, hibakezelést végez és státuszbiteket állít be. A parancs végrehajtása után megszakítást kér a host-géptől, amely megvizsgálhatja a státuszbiteket, kiolvashatja az adatokat, újabb parancsot adhat, stb.
A vezérlőnek a következő 8 utasítást adhatjuk a parancsregiszterblokk parancsregiszterén keresztül.

PARANCS	7	6	5	4	3	2	1	0
Fejállítás a 0.sávra	0	0	0	1	R3	R2	R1	R0
Fejállítás az n.sávra(SEEK)	0	1	1	1	R3	R2	R1	R0
Szektor olvasás	0	0	1	0	0	0	L	T
Szektor írás	0	0	1	1	0	0	L	T
Sáv formázása	0	1	0	1	0	0	0	0
Olvasás ellenőrzés (Verify)	0	1	0	0	0	0	0	T
Belső diagnosztika	1	0	0	1	0	0	0	0
Paraméterállítás	1	0	0	1	0	0	0	1

Az egyes bitek jelentése a következő:

R3,R2,R1,R0 bitek 4 bites bináris számként értelmezve a lépésidőt adja meg, így a vezérlő eltérő minőségű és sebességű merevlemezes meghajtókhoz is használható. A lépésidő kiszámítása: ha (R3,R2,R1,R0) = 0 akkor 35 μsec,
egyébként a lépésidő = (R3,R2,R1,R0)*0,5 msec
Reset v. diagnosztika után a lépésidő 7,5 msec-ra áll be.

L: 1 értéke engedélyezi az ún. hosszú (long) írási és olvasási módot. Ebben a módban minden 512 byte hosszú adatblokk kiegészül további 4 byte-al, amely hibajavító ECC kódot tartalmaz.

T: ha a T bit értéke 0, hiba észlelése esetén a vezérlő újból megkísérli az adott műveletet végrehajtani. Ha a bit 1, ismétlés nincs.

8. RÉSZ

Tervezési és kivitelezési szempontok saját fejlesztésű adapterkártyák esetén

8.1 Az áramköri tervezés

8.2 Az adapter kártya elkészítése, szerelése

8.3 A kártya bemérése, a hibák kijavítása

8.4 Az IBM prototípus kártya

8.5 Az I/O csatorna kiterjesztése, az IBM XT I/O buszkiterjesztő egység

Ha valaki a személyi számítógépet nem önálló egységként, hanem egy rendszer tagjaként akarja felhasználni, szinte azonnal azzal a követelménnyel fog szembekerülni, hogy a PC-t különféle feladatokat ellátó adapterekkel kell felszerelnie. Valószínű, hogy ezeknek az I/O adapter kártyának a nagy része készen kapható, de előfordulhat az is, hogy a gyártó vagy a felhasználó saját maga fejleszt ki egy-egy speciális feladatot ellátó, vagy nehezen megvásárolható I/O modult.

Gyakran előfordul az is, hogy a géppel dolgozó pusztán hobby-ból vagy kíváncsiságból próbál elkészíteni egy számára érdekes kártyát.

Ezekhez a munkákhoz szeretnénk némi segítséget nyújtani, összefoglalva a tervezéshez és a kivitelezéshez szükséges általános szempontokat. Az ajánlások elsősorban azoknak szólnak, akik először fognak hozzá ilyen feladathoz, vagy minimális hardware ismeretekkel rendelkeznek. Néhány rész azonban tapasztaltabb hardware-es szakemberek számára is érdekes lehet.

Mint minden munkának, az I/O adapter kártyák elkészítésének is különböző, egymásutáni fázisai vannak, és az egyes fázisok alatt néhány lényeges szabály betartására ügyelni kell. Ha ezt nem tesszük, a hibákat utólag kell korrigálni, ami már sokkal nehezebb feladat, mintha előzőleg pontosan átgondoltuk volna a teendőket. Egy kártya elkészítése a következő fázisokat igényli:

- áramköri tervezés
- az áramkörök összehuzalozása, a kártya megszerelése
- a kártya bemérése, a hibák kijavítása (debugging).

Mind a három fázis alatt nagyon fontos a dokumentáció pontos elkészítése, a változtatások, javítások utólagos rögzítése a rajzokon. Ha ezt nem tesszük meg, egy későbbi időpontban, amikor a kártyán egy meghibásodást kell megkeresni, vagy valamilyen változtatás szükséges, az alábbi problémák adódhatnak: (1) a dokumentáció olvashatatlan, (2) a kapcsolási rajz nem felel meg a kártyán kivitelezett áramköröknek, (3) dokumentáció egyáltalán nem létezik, vagy bizonyos áramköri részek nincsenek dokumentálva.

8.1. Az áramköri tervezés

Aki nem tapasztalt digitális tervező, valószínűleg tanácstalan lesz, hogy hogyan fogjon hozzá az interface megtervezéséhez. Érdemes ekkor az amatőr tervezés egyik jól bevált módszeréhez folyamodni: "nézzük meg, hogy csinálja valaki más!" Sokat segíthet egy hasonló funkciókat megvalósító adapter kártya kapcsolási rajzának a tanulmányozása. Ha a megtervezendő kártya saját felhasználásra, kísérleti célból készül, legegyszerűbb az azonos részek lemásolása. Ha viszont a kártya eladási célra készül, ez az út nem ajánlott.
A tervezés kiinduló lépése mindig az igények pontos meghatározása. Érdemes egymásután felírni, amilyen részletességgel csak lehet, hogy a kártya egyes áramköreinek milyen feladatokat kell megvalósítaniuk. Az interface általában több áramköri csoportra osztható, amelyek egymástól viszonylag függetlenül üzemelhetnek. Ilyenek a cím-pufferek és adatmeghajtók, a címdekóderek, oszcillátorok, a külső perifériális eszközökkel kapcsolatot tartó nagyteljesítményű meghajtóáramkörök, stb. Ezeknek az áramköröknek a

példák találhatók az IBM XT és IBM AT Technical Reference könyvekben, amelyek több adapter részletes kapcsolási rajzát is közlik, vagy a mikroprocesszor és személyi számítógép orientált BYTE magazinban. Fontos, hogy a másolásra szánt kapcsolási elrendezések működését jól megértsük, és a saját felhasználási igényeknek megfelelően módosítsuk azokat.

Gyakran az interface modul legfontosabb eleme egy LSI chip: egy UART (RS232 adapter), egy IEEE488 vezérlő (GPIB adapter), egy léptetőmotor vezérlő chip, vagy egy PPI chip (programozható parallel interface). Az ilyen intelligens I/O eszközökről már a 2.2.4. fejezetben volt szó. Ha valaki egy ilyen elemet kíván az adapter kártyán elhelyezni, az elem kiválasztásánál körültekintően kell eljárni: csak olyan chip-et szabad felhasználni, amely felépítését, kivezetéseit tekintve, jól illeszkedik az I/O busz struktúrájához. Az IBM Technical Reference által ismertetett kártyákban található valamennyi intelligens I/O chip megfelel ennek a követelménynek. Ezért is ajánlható a Technical Reference tanulmányozása.

Felsorolnánk néhány olyan LSI áramkört, amelynek a használatát árdemes elkerülni: az egyik ilyen típust azok az integrált áramkörök képviselik (pl. soros interface), amelyek multiplexelt cím- és adatvonalakkal rendelkeznek. Mint ismeretes, az IBM I/O csatorna külön adat- és címvonalakat tartalmaz. Egy gyakran használt népszerű integrált áramkör a ZILOG PIO (Parallel Input/Output), amit szintén érdemes elhagyni a választékból (ha interruptokkal használják), mivel a CPU-hoz hasonlóan figyeli az adatbuszon megjelenő utasításokat, és automatikus válaszokat ad ki, ha bizonyos Z80-as kódokat detektál.

Amikor az interface kapcsolási vázlata elkészült, érdemes lépésről lépésre átnézni, megvizsgálni, hogy a logikai áramkörök helyesen működnek-e, és hogy a modul akkor és csak akkor küld-e válaszokat, ha a buszon az előírt utasítások jelennek meg. Érdemes elgondolkodni a logikai áramkörök minimalizálásán, hogy minél kevesebb diszkrét elemből legyenek felépíthetők.

A kártyát úgy célszerű megtervezni, hogy maradjon rajta egy kis szabad hely a későbbi módosításokra és kiegészítésekre. Ha valamelyik áramkör működésével kapcsolatban kétségeink támadnak, célszerű azt külön megépíteni és kipróbálni. Így mód nyílik néhány kritikus értékű, vagy nehezen kiszámítható alkatrész (ellenállás, kondenzátor) végleges értékének meghatározására. Analóg áramkörök, erősítők, oszcillátorok ilyen módon történő előzetes kipróbálása feltétlenül ajánlott, mielőtt a kapcsolást véglegesítenénk.
Az interface tervezésénél gondolni kell arra, hogy az I/O csatornának az adapterek felé irányuló jelei kártyánként maximum két 'low-power Schottky' (LS) egységterheléssel terhelhetők.

Amikor a kártya áramkörei minden apró részletben a végleges állapotot tükrözik, ajánlatos hozzáfogni a pontos dokumentáció elkészítéséhez. Bár sokkal kézenfekvőbb gondolat a kártya szerelését elkezdeni, a későbbiekben kifizetődőbb, ha egy részletes dokumentáció áll rendelkezésre. A kapcsolási rajzot könnyebb nyomon követni és értelmezni, ha a rajzolásnál a mindenki által megszokott, konvencionális logikai jelöléseket alkalmazzuk (kapuk, bistabilok, regiszterek, stb.). A jelek haladási iránya lehetőleg balról jobbra mutasson. A bemeneteket a rajz baloldalán, a kimeneteket a jobboldalán helyezzük el.

A logikailag azonos csoportba tartozó elemeket, áramköröket célszerű egymás közelében és nem szétosztottan elhelyezni. Az egymást keresztező vonalakat csak akkor szabad összekötöttnek tekinteni, ha a keresztezésben egy pont van elhelyezve. Az alkatrészek, integrált áramkörök lábszámozását a huzalozás, vagy az adapter nyomtatott áramköri rajzának az elkészülte után kell kitölteni.

Akinek módja van rá és rendelkezik valamilyen kapcsolási rajz készítést segítő grafikus programcsomaggal, pl. OrCAD/SDT, a dokumentációt a számítógépen is elkészítheti. Ezt a megoldást választva, a későbbi javítások, változtatások könnyen keresztülvihetők, nincs szükség a rajzok, jegyzékek tárolására, a dokumentáció bármikor kinyomtatható és sokszorosítható.
Röviden ismertetnénk az OrCAD/SDT kapcsolási rajz és dokumentáció tervezést-segítő programcsomag jellemzőit, mivel úgy tűnt, hogy ez a program nem csak megkönnyíti a dokumentáció készítést, hanem segít abban is, hogy minél kevesebb hibát kövessünk el a rajzolásnál. Egy hibátlanul elkészített kapcsolási rajzról az OrCAD olyan összekötési listát (netlist) állít elő, amely közvetlenül valamilyen automatikus NYÁK-tervező program bemenő adatait szolgáltathatja (pl. RACALREDAC, CALAY).
A programcsomag a DRAFT rajzoló, a DECOMP könyvtári visszafordító, a COMPOSER könyvtári fordító, a NETLIST összekötési listát előállító, az ERCHECK ellenőrző programot, továbbá számos felhasználói programot foglal magába.
A kapcsolási rajzot elkészíthetjük:

- Több önálló dokumentációs file formájában ('flat file' struktúra), ahol egy-egy file egy-egy „munkalapot" tartalmaz (a kapcsolási rajz valamely részét), és a lapok közös csatlakozó vezetékekkel és jelekkel rendelkeznek.
- Hiarerchikus file struktúrában, ahol a munkalapok blokkos szervezésűek, és minden egyes blokk a kapcsolás egy részét tartalmazza. Egy blokkon belül újabb blokkok helyezhetők el, így a rajz egymás mellé és egymásba épített blokkok sokaságából állítható elő. A hierarchikus struktúra mindig tartalmaz egy kiindulási ('root') munkalapot.
- Egymunkalapos file struktúrában, ahol a rajz minden részét egy munkalap tartalmazza.

A munkalapok mérete A - tól (11,0"x8,5") E - ig (44,0"x34,0") terjedhet. A munkalapokon lévő rajzokat kinyomtathatjuk a konfigurációs programból választott különféle nyomtatókon és rekordereken.

DRAFT.EXE -- A kapcsolási rajzok elkészítését, editálását és a munkalapok file-kezelését teszi lehetővé. A munkát gyorsítja, ha a keyboard mellett valamilyen „egér" (mouse) is rendelkezésre áll.

Főbb jellemzői:

- 3000-nél több szabványos könyvtári elemet tartalmaz (digitális, analóg áramkörök, kapcsolási szerelvények).
- A digitális könyvtári áramköri kapuelemek De Morgan ekvivalensekkel is rendelkeznek.
- A billentyűzettel és / vagy az egérrel különféle típusú összekötések, buszok, csatlakozók, címkék és összekötési pontok realizálhatók.
- Blokkok mozgatásakor az összekötések és buszok követhetik az elmozdított részeket.

Az adapter kártyák címdekóderei összehasonlítják a címvonalakat a kártyán beállított I/O címekkel. A címdekóder kimenetei rendszerint az IC-k CS (Chip Select) bemeneteit vezérlik. A címdekóder kialakítása a kétféle adapter típus esetén eltérő. Memória mappelt I/O kártya címdekódere valamennyi címvonalat (A0-A19) figyeli. Az alsó címbitek elhagyhatók. Ekkor az első címvezetékek (pl. A0-A4) a kártyán belüli címszétválasztásokra használhatók. Nem memória mappelt kártyáknál a címdekóder egyszerűbb lehet, itt ugyanis nincs szükség teljes dekódolásra. A 8088 processzor eleve csak 65536 bemeneti és kimeneti címet tud képezni. (Az A16-A19 címbitek I/O utasításoknál 0 értékűek.)

Ez 16 címvonalat jelent, de az IBM személyi számítógépek csak az A0-A9 címvonalakat használják fel az I/O eszközök kiválasztásához, az A10-A15 vonalakat nem veszik figyelembe. Ezért ezeken a felső címvezetékeken előállt bármely bitkombináció mellett az A0-A9 vonalak által kiválasztott áramkör aktív állapotba kerülhet. Ez az úgynevezett redundáns címzési mód, amely egyszerűbb címdekódoló logikát enged meg. Az A0-A9 címvonalakkal a 0-3FFh tartomány címezhető. Az XT és AT gépek I/O címtérképeit a 2-4 és a 2-19 táblázatok tartalmazzák.

A 0-FFh tartományba eső címek az alapkártya részére fenntartottak, az I/O csatornára kapcsolt eszközök címei az efölötti területet foglalhatják el. Mint a címtérképekből kiderül, a 100-3FFh tartomány nagyrészét különféle adapter kártya típusokhoz rendelték hozzá. Saját tervezésű adapterek a felhasználók számára biztosított 300-31Fh címterületet foglalhatják le elsősorban (prototípus kártya részére fenntartott hely). Ha ez a tartomány szűknek bizonyul, vagy ez a hely már egy másik prototípus kártya által foglalt, a címtérképet átnézve további nem használt címtartományokat választhatunk.
A címdekóder az alsó címvezetékeket rendszerint figyelmen kívül hagyja, így a dekóder kimenetekkel összekötött IC-k további alcímekkel rendelkezhetnek, amelyeket közvetlenül az alsó címvonalak választhatnak ki.

Valamennyi I/O címdekódernek figyelnie kell az AEN (address enable) vonal állapotát. A dekóder nem adhat kimenőjelet, ha az AEN jel logikai 1 értékű. Ekkor valamely DMA ciklus van folyamatban, és mivel az I/O vezérlőjelekkel egyidőben memória címek is jelen vannak, előfordulhatna, hogy a címdekóder tévesen I/O címet detektálna és egy téves I/O átvitel jönne létre.

Memória mappelt I/O adapter kártyákat olyan esetekben célszerű tervezni, ha az I/O eszköznek nagymennyiségű adatot kell begyűjtenie, amit a processzornak még utólag, vagy 'real time' módon fel kell dolgoznia. Ezek a kártyák lefoglalják I/O célokra a rendszer memória címterületének egy részét. Ezen a területen helyezkednek el az I/O címek, és az esetek többségében egy kettős hozzáférésű memória (dual port memory), amit mind az I/O eszköz, mind a CPU használhat. A kártya áramköreit úgy tervezzük, hogy közelítőleg memória-ciklus idejű I/O műveletek jöjjenek létre.

A memória mappelt I/O adapterek tipikus esete néhány nukleáris célú adatgyűjtő kártya. Például a spektroszkópiában használatos sokcsatornás analizátor alapvető funkcióit megvalósító kártya spektrum tárját a kártyán elhelyezett A/D konverter kimenete folyamatosan tölti adatokkal, a CPU pedig esetenként kiolvassa a tár tartalmát megjelenítési és feldolgozási célokból. A spektrum tár címtartománya a rendszer-memória

címterületének egy olyan részét foglalja el, ami az alapkártyán nincs memória chip-ekkel kitöltve. A spektrum tár a rendszer memória részévé válik és a processzor által közvetlenül címezhető.

Memória mappelt adapter választása akkor is indokolt lehet, ha nagymennyiségű, memóriában előkészített adatot kell kiküldeni az I/O eszköz részére. Az esetektől függően kell eldönteni, hogy egy memóriaként kezelt I/O eszköz, vagy egy DMA átvitel megvalósítására alkalmas modul közül melyik a jobb választás.

Az I/O modul és az alapkártya közötti DMA összeköttetés könnyen realizálható. Az alaplemezen elhelyezett 8237A típusú integrált áramkör vezérli és összehangolja a DMA csatornák működését. XT gép esetén az 1. csatorna, AT gép esetén az 1, 5, 6 vagy 7. csatornák jöhetnek számításba. Minden esetben vizsgáljuk meg, hogy a kiszemelt csatorna nem foglalt-e más célokra a rendszeren belül. A DMA átvitelt az I/O kártya a DRQi vonalon kezdeményezheti (i a csatornaszámra utal). A kártya feladata a -DACKi nyugtázó jel vételekor mindössze annyi, hogy az adatbuszra helyezze, vagy beolvassa az aktuális információt. A memória és I/O vonatkozású vezérlőjeleket, valamint az átvitelt nyugtázó jeleket a 8237A típusú vezérlő állítja elő. A DMA vezérlő programozásáért a választott csatornán az I/O modult kezelő software a felelős (device driver).
Ha az adapter kártya interrupt vonalakat is használ, vizsgáljuk meg, hogy az I/O csatorna megszakítást kérő bemenetei közül melyek nincsenek felhasználva a rendszeren belül. Általában kevés ilyen bemenetet fogunk találni. XT gépeknél a legtöbbször az IRQ2,3 AT gépeknél az IRQ3,5,9,10,11,12 vonalak szabadok, ha a második RS232 interface-t nem használják (COM2). Amennyiben a szabad bemenetek száma kevésnek bizonyulna, az interrupt vektorcímek átirányításával az adapter működésének idejére a foglalt bemenetek némelyike ideiglenesen felszabadítható.

Az alaplemezen elhelyezett 8259A interrupt vezérlők programozásáért szintén az I/O modult kezelő software a felelős (az adapter által használt bemenetek vonatkozásában).

8.2. Az adapter kártya elkészítése, szerelése.

Az áramköri tervezés és a dokumentálás után következhet a kártya elkészítése. Ha csak egy példány szükséges, a legjobb megoldás ha beszerzünk egy prototípus kártyát. Prototípus kártyát, vagy elterjedtebb néven 'joker' kártyát több cég árul, és Magyarországon is beszerezhetők. Az IBM prototípus kártya a 8.4 pontban kerül bemutatásra. Ez egy XT I/O csatlakozóval szerelt 'joker' kártya, amelyen néhány interface elemet is elhelyeztek: van rajta cím-, adat- és vezérlőjel-buffer, címdekóder a 300-31Fh tartományra, továbbá egy hátoldali csatlakozó a külső eszközök részére. Ezek az alkatrészek körülbelül a kártya 1/5 részét foglalják el. A többi rész mintegy 4000 darab, forrasztható szegéllyel ellátott lyukgalvanizált furattal van kitöltve. Az alkatrészoldalon föld-sinek, a forrasztási oldalon +5V tápfeszültség-sinek húzódnak.
Bár a lyukgalvanizált furatokba az IC'-k és egyéb alkatrészek beforraszthatók és a huzalozás forrasztott vezetékekkel kivitelezhető, ha egy mód van rá kerüljük el ezt a megoldást, és a 'wire-wrapped' huzalozást alkalmazzuk. Ez ugyan nagyobb felkészültséget igényel (szerszámok, foglalatok), de megbízhatóbb kötési módot ad mint a forrasztás, és a hibalehetőségek száma is csökken. Mivel minden integrált áramkört foglalatba kell

helyezni, ha valamelyik meghibásodik, egyszerűen cserélhető. Az ellenállásokat, kondenzátorokat célszerű huzalozható tüskékre forrasztani.

A foglalatok elhelyezése után először a tápfeszültség és a föld hozzávezetéseket huzalozzuk be, és csak ezután fogjunk hozzá a lábak kapcsolási rajz szerinti összekötéséhez. Célszerű a tápfeszültség hozzávezetésekhez más színű huzalt használni. Ne feledkezzünk meg a tápfeszültség-szűrő kondenzátorok beépítéséről sem. Minden második IC mellett ajánlott egy 22-47nF értékű kerámia kondenzátor elhelyezése, a csatlakozó közelében pedig szűrjük meg a tápfeszültségeket 10-30 µF értékű tantál kondenzátorokkal.

A foglalatokba az alkatrészeket csak a kártya bemérésekor tegyük bele.
Ha az adapter kártya eladási célokra készül, el kell készíteni a nyomtatott áramköri rajzokat. 'Wire-wrap' huzalozott prototípust ebben az esetben is érdemes összeszerelni, és csak a prototípus bemérése és a bemérésnél tapasztalt hibák kijavítása után érdemes hozzáfogni a NYÁK tervezéshez.

A NYÁK tervezés vagy a hagyományos úton (kézzel való megrajzolás, majd filmkészítés), vagy számítógépen történhet. A számítógépes tervezés professzionális 'layout'-ot eredményez. Az IBM gépeken futtatható NYÁK tervező programcsomag például a smARTWORK, a Wintek cég terméke. Röviden ismertetnénk ennek a programnak a lehetőségeit. A gép képernyője ablaknak tekinthető, ami a nyomtatott áramköri lap egy részét mutatja. Az ablakot mozgathatjuk a lap felett. A NYÁK egyik, vagy mindkét oldala megjeleníthető egyidőben, különböző színekkel.

A billentyűzetet használva, összekötési pontokat és összekötési vonalakat helyezhetünk el, vagy szüntethetünk meg. A program felügyel a vonalak közti távolságokra és a vonalak vastagságára. Az összekötési és kivezetési pontokat rögzített raszter helyeken lehet elhelyezni, két raszter pont távolsága 0,05" (1,27 mm). A vonalak vízszintesen, függőlegesen, vagy 45° alatt haladhatnak. Két kijelölt pont összekötését a program automatikusan elvégzi, megkeresve a legrövidebb utat, miközben ügyel arra, hogy más vezetékeket ne keresztezzen ugyanazon az oldalon. Ha egy rövidebb utat is lehet találni más pontból kiindulva, mint amit megadtunk, a program a rövidebb utat választja. Természetesen arra is mód van, hogy az összekötést mi rajzoljuk be az általunk megkívánt úton a két pont közé. Kétféle vonalvastagság között választhatunk. A vonalvastagság utólag is módosítható. Ha két alkatrész között kevés a hely a vonalak számára, a program lehetőséget ad az elrendezés széthúzására mind vízszintes, mind függőleges irányban.
A kész NYÁK rajzot DOS file-ban tároljuk. A későbbi változtatások gyorsan és pontosan keresztülvihetők, miközben az eredeti állapotot tükröző file-t is megtarthatjuk. A NYÁK rajz papírra nyomtatásának két útja van: vagy egy jóminőségű pont-mátrix nyomtatót, vagy X-Y rekordert használunk. A rajz 2:1 vagy 1:1 méretben készíthető el. Fotózáshoz a 2:1 méretet célszerű választani. A tervezési idő lecsökkenthető, ha a cursort nem a billentyűzet gombjaival, hanem "egérrel" (mouse) mozgatjuk. Az egéren lévő két kapcsolóval az összekötések kezdő és végpontjai jelölhetők ki.

Az smARTWORK jellemzőit felülmúlja a RACALREDAC tervező program. Az OrCAD NETLIST program által előállított kimeneti file használható a RACALREDAC bemeneteként. Így mód nyílik arra, hogy az OrCAD segítségével megszerkesztett hibátlan

kapcsolási rajzból a RACALREDAC elkészítse a nyomtatott áramköri lemez rajzát. Az alkatrészek elhelyezését meghatározhatja a tervező, de maga a program is. A RACALREDAC sűrűbb vonalvezetést enged meg mint az smARTWORK (két IC kivezetés között két vonal is átvezethető), és sokrétegű nyomtatott áramköri lapok tervezésére alkalmas.
Használata több gyakorlatot igényel, mint az smARTWORK, ezenkívül az OrCAD pontos ismerete is szükséges ahhoz, hogy hibátlan kapcsolási rajzok készüljenek, ami alapfeltétele a RACALREDAC-kal végzett munkának.

8.3. A kártya bemérése, a hibák kijavítása.

Először ohm-mérővel, ellenőrizzük, hogy nincs-e rövidzár valamelyik tápfeszültség bevezetés és a föld között. Ha ez nem áll fenn, az összehuzalozott prototípus kártyát alkatrészek nélkül helyezzük a gép I/O csatlakozójába. Jó szolgálatot tesz ilyenkor egy kihosszabbító, amit mi magunk is könnyen elkészíthetünk. A gép bekapcsolása után voltmérővel ellenőrizzük azokat a pontokat, ahol a tápfeszültségeknek meg kell jelenniük.
Ha mindent rendben találunk, kapcsoljuk ki a gépet és tegyük be a foglalatokba az integrált áramköröket és a többi alkatrészt. Vigyázzunk, nehogy valamelyik IC fordítva kerüljön a foglalatába. Különös gonddal kell eljárni a MOS (Metal Oxide Semiconductor) eszközök behelyezésénél, ügyelve arra hogy ne legyünk sztatikus elektromossággal feltöltve, amikor ezt a műveletet végezzük. A MOS IC-ket eleve alumínium fóliában, vagy vezető habszivacs lapra szúrva szállítják, ami az összes kivezetésüket rövidre zárja. Kivétel előtt érintsük meg a gép leföldelt fém házát (ott ahol nincs rajta festés), és a vezető fóliát is érintsük hozzá. Ha a kártyáról kiveszünk egy MOS eszközt, a vezető habszivacs lapon tároljuk.

Miután az alkatrészeket a helyükre tettük, kapcsoljuk be a gépet. Ha minden úgy működik ahogy elő van írva, a további részeket nem is kell elolvasni.

Sajnos az esetek többségében azonban nem ez a helyzet. Elég ha egy apró dolog elkerülte a figyelmünket és máris az egész adapter kártya teljesen működésképtelennek látszik.
Mielőtt a hibakereséshez fognánk, vegyük ki a kártyát, és nézzük át gondosan a következők szerint:

- minden IC-t és alkatrészt rátettünk-e?
- nem cseréltünk-e fel két IC-t?
- nem tettünk-e be fordítva egy IC-t?
- nem hajlott-e el valamelyik IC lába a foglalat alá amikor behelyeztük?
- a huzalozási oldalon nem ér-e össze két 'wire-wrap' foglalat kivezetése?

Ha itt nem találunk hibát, tegyük vissza a kártyát és ellenőrizzük a hátoldali csatlakozásokat, hogy a külső perifériális berendezéssel megfelelő-e az összeköttetés. Gondoljunk arra, hogy a hibának nemcsak hardware, hanem software okai is lehetnek. Ellenőrizzük, hogy a kártyán behuzalozott, vagy kapcsolóval beállított I/O port száma megegyezik-e azzal, mint amit a software használ. Annak eldöntésére, hogy az interface egyáltalán reagál-e valamire, legegyszerűbb a DEBUG hibakereső programot futtatni és néhány soros assembler tesztprogramokat írni, felhasználva a DEBUG A (Assembler)

utasítását. Ezek a minimális programok valószínűleg csak néhány OUT és IN utasítást tartalmaznak, de alkalmasak annak megítélésére, hogy a kártya egyes részei működőképesek-e vagy sem.

Ha az adapter nem működik, az első gondolat az lehet, hogy valamelyik IC hibás. Bár ez az eset is előfordulhat, de eléggé ritkán, ezért mielőtt IC cseréhez folyamodnánk, gondosan ellenőrizzük az áramkört. Egy gyors módszer a hibás IC-k felderítésére, ha bekapcsolt állapotban végigtapintjuk őket. Amelyik égeti az ujjunkat, az általában cserére szorul. Ne tévesszen meg azonban, hogy az IC-k különböző hőmérsékletűek. A bufferek, a 74S sorozat elemei a nagyobb áramfelvétel miatt melegebbek a többinél. Ha egy IC-t hibásnak találunk, nem biztos hogy a csere segít. Sokszor előfordul, hogy a hibát valamilyen szerelési, vagy áramköri hiba okozza, és a kicserélt alkatrész is tönkremegy.
Ha az eddigi egyszerű módszerekkel nem sikerült a hibát elhárítani, részletesebb hibakeresésre van szükség, amely néhány műszert is igényel. A következő műszerek jöhetnek számításba: (1) voltmérő, (2) logikai teszter, (3) oszcilloszkóp, (4) logikai analizátor. A logikai teszter két kijelzővel rendelkezik, egyik a TTL0, a másik a TTL1 szint jelzésére szolgál. Impulzus sorozat detektálásakor mindkét kijelző világít. A kijelzők fényerejéből következtethetünk arra, hogy a pozitív vagy a negatív impulzusok vannak-e túlsúlyban. A logikai teszterek fontos szolgáltatása a keskeny pulzusok detektálása és kijelzése: 10 ns-nál szélesebb impulzusokat a teszter megnyújtja és a kijelzőt felvillantja. Segítségével a zavaró 'glitch'-ek is felderíthetők. A logikai teszterek jól használhatók a logikai áramkörök működésének ellenőrzésénél, alapvető hibák felderítésénél. Vannak azonban olyan esetek, amikor oszcilloszkóp nélkül nem boldogulunk: ha egy oszcillátor jelét kell megfigyelni, vagy az időzítését beállítani, ha két digitális jel nem a megfelelő időpontban jelenik meg egymáshoz képest, ha zajproblémákat kell megoldani amelyek megzavarják az áramkörök működését, stb. Az oszcílloszkópnak legalább kétsugarasnak és 50-60 MHz sávszélességűnek kell lennie.

Bonyolultabb logikai hibák felderítésére jól használható a logikai analizátor. A logikai analizátor egyidejűleg nagyszámú digitális vonal működését tudja nyomonkövetni (32-80 bemenet), és a memóriájában rögzíteni a jelkombinációkból képzett triggerjel segítségével. A triggerjel előtti és utáni állapotok is felvételre kerülhetnek a memóriában, így a hibát megelőző okok nyomonkövethetők.

A felsorolt műszerek közül a logikai analizátorhoz a legnehezebb hozzájutni, de az adapter kártyák nagyrésze enélkül is bemérhető.

A részletes hibakereséshez nem lehet megadni pontos útmutatásokat, hogy mit érdemes megvizsgálni, az mindig a hiba jellegétől és az áramköröktől függ. Mindenesetre érdemes az áramköröket logikai blokkokra osztani, és ezeknek a blokkoknak a bemeneteit és a kimeneteit figyelni. Felsorolunk néhány dolgot, amit érdemes ellenőrizni:

1) Ha az adapteren van valamilyen oszcillátor, ellenőrizzük hogy működik-e. A vizsgálathoz elég a logikai teszter. Ugyanígy ellenőrizzük a rendszer órajelének (CLOCK) a jelenlétét, ha az adapter használja azt.

2) Célszerű ellenőrizni a bufferek engedélyező és irányt kijelölő bemeneteit (ha vannak ilyenek) és a címdekódoló áramköröket. Az ellenőrzéshez oszcilloszkóp és

egy programhurkot megvalósító assembler tesztprogram szükséges, amely a folyamatos megfigyelést lehetővé teszi.

3) Ha az adapter kártyán intelligens I/O eszköz került beépítésre és nem működik, két dologról győződjünk meg: (1) valamennyi bemenetéhez eljut-e a kiadott információ, (2) az I/O eszköz kiválasztásra kerül-e (Chip Select). Amennyiben mindez teljesül, valószínűleg software hiba okozza a gondot. Vizsgáljuk meg az I/O eszközt inicializáló programot.

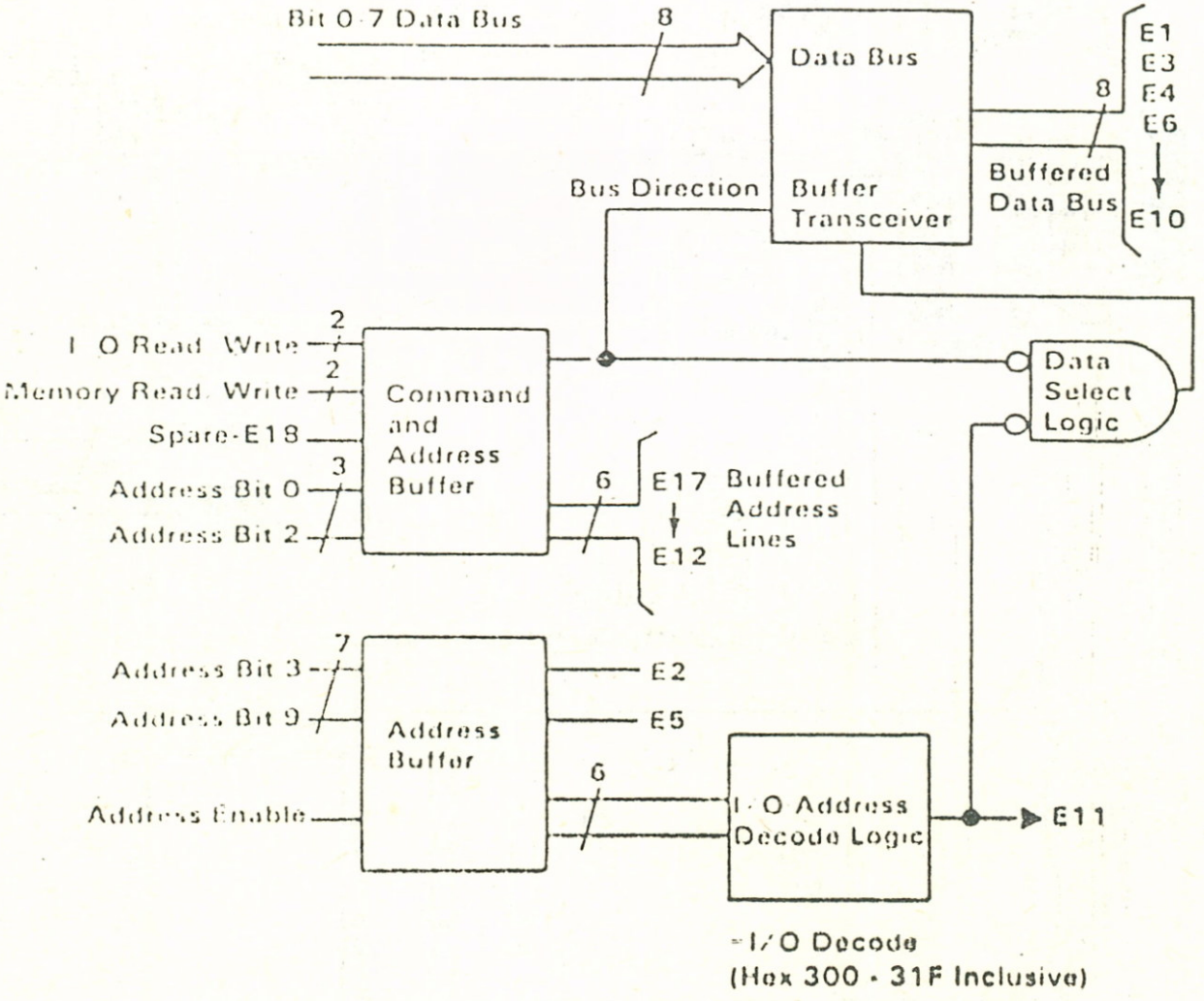
Felhívnánk a figyelmet arra, hogy az adat és címvonalakat megfigyelni nem könnyű dolog a rendszer-óra interrupt kérések és a 15 μsec-ként ismétlődő frissítő ciklusok miatt. Ezek a funkciók az adat és címvonalakat és a -IOR -IOW vonalakat is használják. Az interruptok kikapcsolhatók, de a frissítést nem szabad megszüntetni.

Mind a hardware mind a software hibakeresésnél alapszabály, hogy jól gondoljuk át a problémát, az előidéző okokat. Rendszerint néhány jól választott mérés a kártyán a legjobb út a hiba felderítéséhez. A hardware hibák nagyobb részét a tervezési pontatlanságok okozzák és csak kisebb részük vezethető vissza hibás alkatrészekre. Ha minden hibát elhárítottunk, ne feledkezzünk meg a dokumentáción is feltüntetni a változásokat.

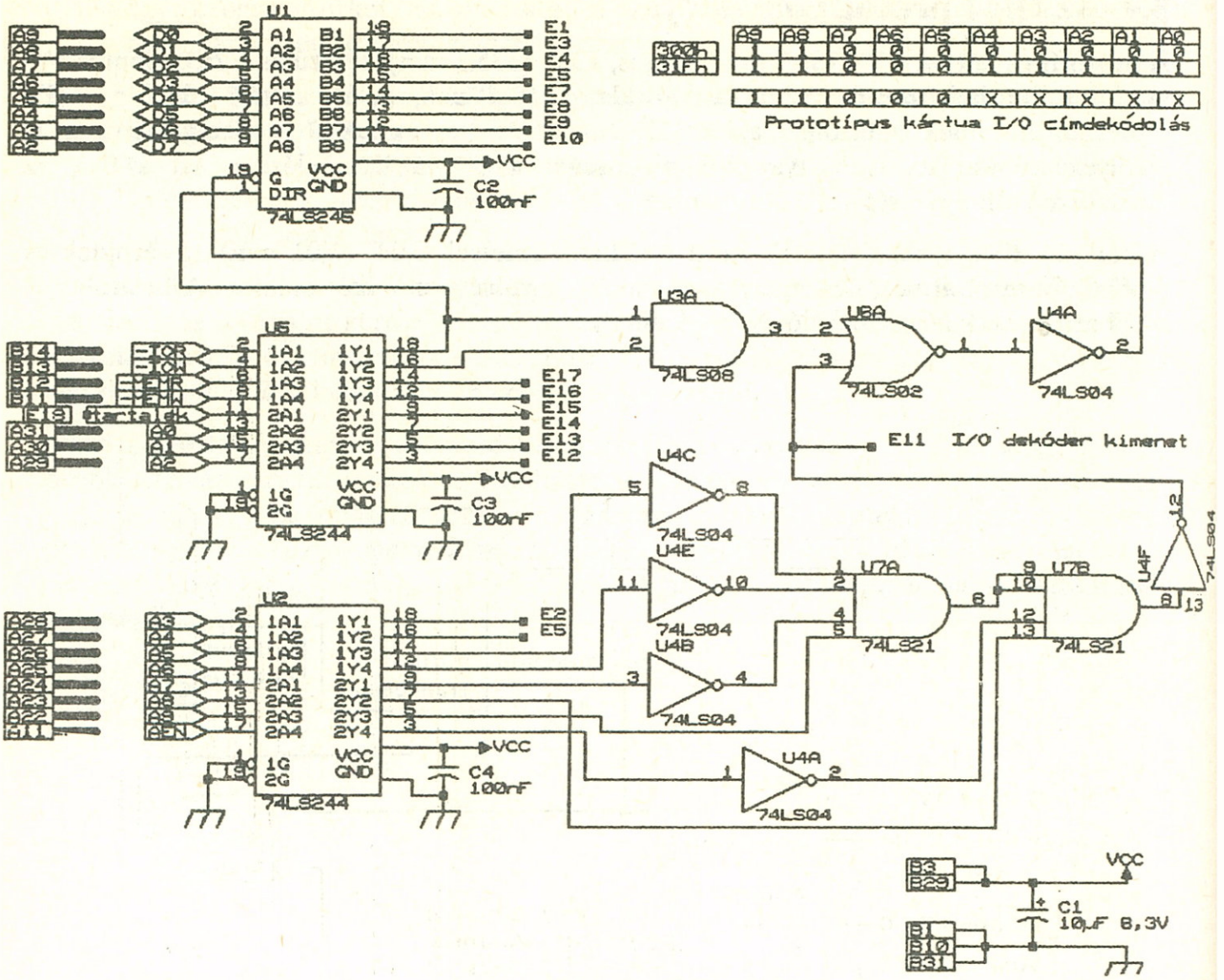
8.4. Az IBM prototípus kártya

A prototípus kártya 4,2" (106,7 mm) magas, 13,2" (335,3 mm) hosszú és a 2x31 pontos I/O csatlakozóba helyezhető. Az alkatrészoldalon földsínek, a forrasztási oldalon +5V tápfeszültség sínek határolják a kártyát. A kártyán egy egyszerű rendszer-interface is elhelyezésre került. A kártya blokkdiagramját a 8-1, a kapcsolási rajzát a 8-2, az alkatrészek elhelyezését a 8-3 ábra mutatja.

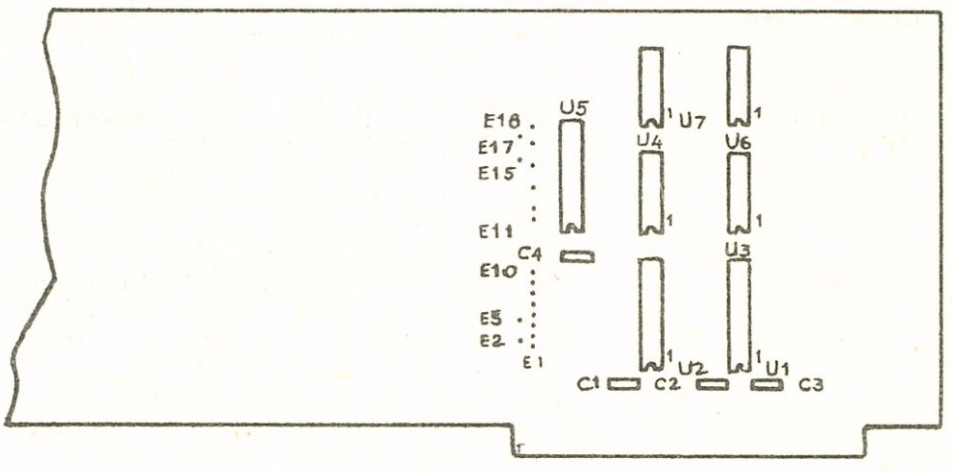
A kártyán 3909 lyukgalvanizált furat található, amelyek 0,04" (1,01 mm) átmérőjűek és 0,06" (1,52 mm) átmérőjű szegélyekkel vannak körülvéve mindkét oldalon. A furatok 0,1" (2,54 mm) osztású raszterhálón vannak elhelyezve.



8-1 ábra.
A prototípus kártya blokkdiagramja.



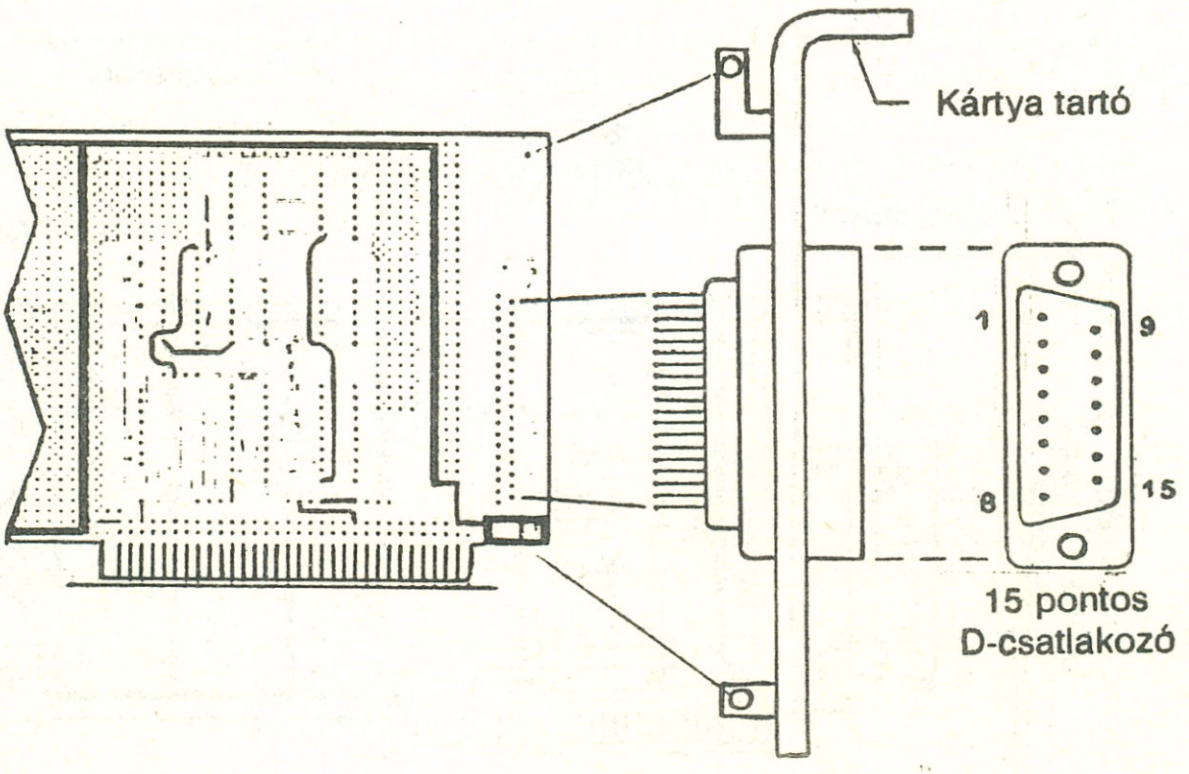
8-2 ábra.
A prototípus kártya kapcsolási rajza.



8-3 ábra.
A prototípus kártya alkatrész beültetési rajza.

A kártyának a hátlaphoz közel eső részén 37 darab 0,048" (1,22 mm) átmérőjű lyukgalvanizált furat található, amelyekbe különböző pontszámú (9-37) 90°-os D típusú csatlakozók forraszthatók. A csatlakozók típusait a következő táblázat, felszerelésüket pedig a 8-4 ábra mutatja.

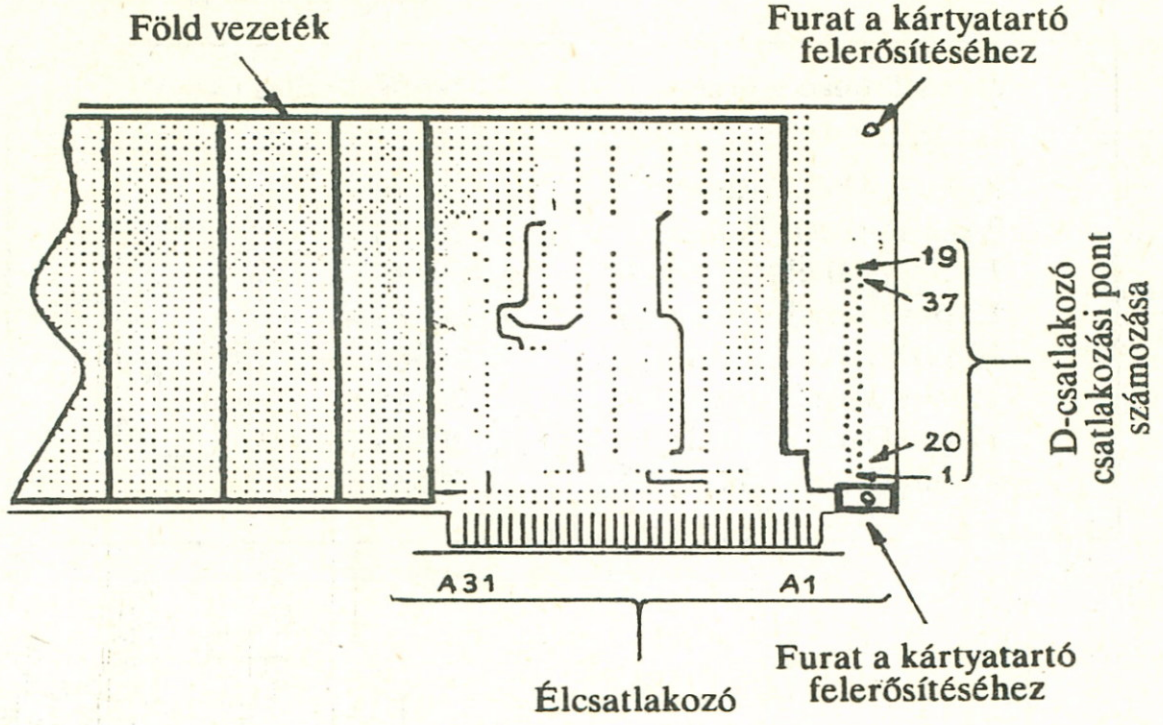
A csatlakozó mérete	Jelölési szám (Amp)
9-pin D-shell (Male)	205865-1
9-pin D-shell (Female)	205866-1
15-pin D-shell (Male)	205867-1
15-pin D-shell (Female)	205868-1
25-pin D-shell (Male)	205857-1
25-pin D-shell (Female)	205858-1
37-pin D-shell (Male)	205859-1
37-pin D-shell (Female)	205860-1



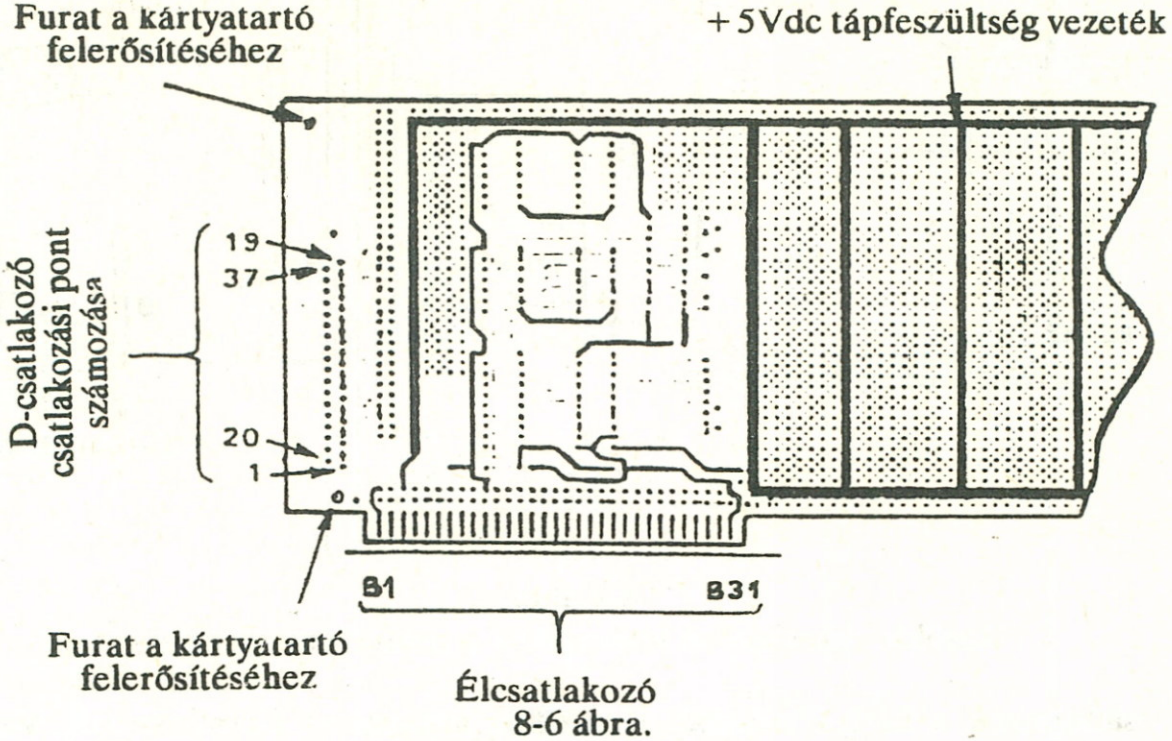
8-4 ábra.
Hátoldali csatlakozó felszerelése a prototípus kártyára.

Az I/O csatlakozók a kártya anyagán kialakított aranyozott élcsatlakozók. Valamennyi csatlakozási pontot egy-egy lyukgalvanizált furathoz vezették, amelyek két sorban helyezkednek el az élcsatlakozók felett.

A prototípus kártya alkatrész oldal felöli nézetét a 8-5, a forrasztási oldal felöli nézetét a 8-6 ábrán láthatjuk. Az interface E1-E17 bufferelt kimenetei köthetők közvetlenül a kártyán megépített áramkörökhöz. Az I/O címdekóder a hex 300-31F címterületet engedélyezi a kártya részére.



8-5 ábra.
Prototípus kártya (alkatrész oldal).



8-6 ábra.
Prototípus kártya (forrasztási oldal).

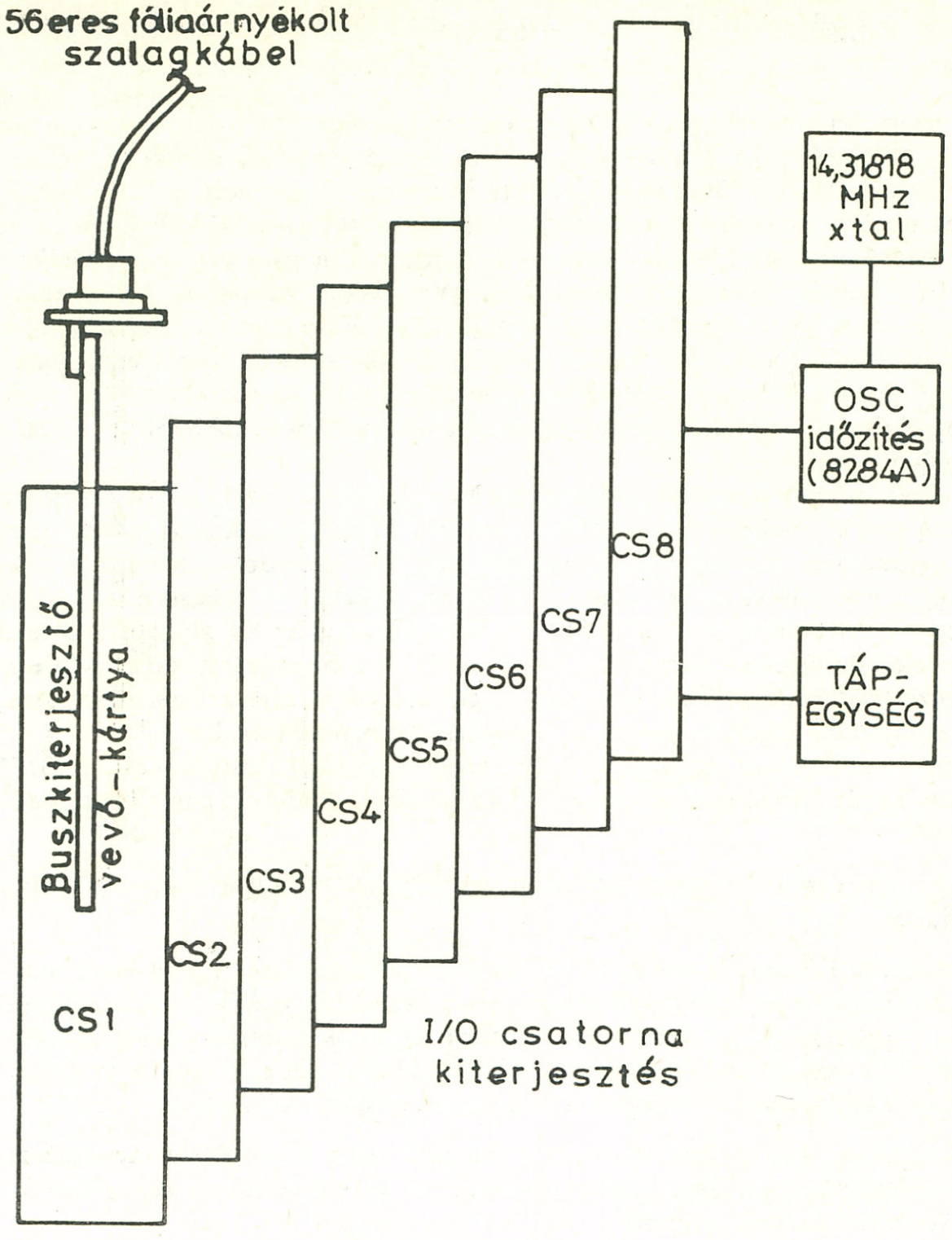
8.5 Az I/O csatorna kiterjesztése, az IBM XT I/O buszkiterjesztő egység.

Az IBM személyi számítógépek I/O buszára csatlakoztatható modulok száma korlátozott (max. 8). Általában 4-5 szabad hely marad egy átlagos kiépítettségű gépben, amely bonyolultabb feladatok elvégzésére épített hardware-t már nem tud kiszolgálni. Ezért szükséges az I/O csatorna-kiterjesztés, amely további modulok befogadására alkalmas. Saját tervezésű adapter kártyák esetén is előfordulhat, hogy a gépben rendelkezésre álló szabad I/O helyek száma kevésnek bizonyul. 'Wire wrapped' huzalozású kártyák szélességük miatt két csatlakozó helyet foglalnak el. Ha a gépben a szabad helyek száma négy, és kettőnél több saját készítésű adaptert szeretnénk elhelyezni, már gondok adódnak. Ezért a buszkiterjesztő egység ismertetése nem haszontalan, főleg azok részére, akik bonyolultabb mérésadatgyűjtő rendszereket kívánnak tervezni az IBM személyi számítógépekhez.
Az XT gépek megjelenésével egyidőben került kibocsátásra az I/O buszkiterjesztő egység (expansion unit), amely az I/O csatorna további 8 hellyel való kibővítésére szolgál. Ez az egység ugyanolyan dobozba került mint az XT alapgép, 130-150 W szabványos tápegységgel van ellátva és Winchester és floppy diszkek beépítésére is lehetőséget ad. (2.1. fejezet). A buszkiterjesztést két kártya valósítja meg: az XT alapgép I/O csatornájára csatlakoztatott kiterjesztő kártya (extender card), és a buszkiterjesztő egység egyik 'slot'-jába helyezett vevő kártya (receiver card). A két kártyát 56-eres, fóliaárnyékolt szalagkábel köti össze, mindkét végén 62 pontos D típusú csatlakozóval szerelve.
A buszkiterjesztő egység vázlatát a 8-7 ábra mutatja. Az I/O csatorna OSC 14,31818 MHz-es jele nem kerül átvitelre az alapgépről, hanem újból előállítják egy 8284 típusú óra-chip segítségével.

A buszkiterjesztéssel előállított I/O csatorna teljesen transzparens az XT alapgép I/O csatornájával.

Az I/O csatorna készenléti vonala (I/O CH RDY) lehetőséget teremt lassú memóriakkal és I/O eszközökkel való együttműködésre. Ha a készenléti jelet a megcímzett eszköz nem használja, a processzor által generált memória ciklusok 5 órajel periódust igényelnek a buszkiterjesztő egységben (A memória és I/O ciklusok késleltetéséhez az I/O CH RDY jelet logikai 0 állapotba kell helyezni a késleltetési óraciklusok idejére).

A 8-1 táblázat az adó és vevő kártyákon áthaladó jelek felsorolását tartalmazza, a jelek mellett feltüntetve a kialakult késleltetéseket. A késleltetésbe az összekötő kábel által okozott késleltetés is beleértendő.



8-7 ábra.
A buszkiterjesztő egység vázlata.

A jel neve	Névleges késleltetés (nsec)	Maximális késleltetés (nsec)	A jel iránya
A0-A19	27	39	XT-->EXT_I/O
AEN	27	39	XT-->EXT_I/O
DACK0-DACK3	27	39	XT-->EXT_I/O
MEMR	27	39	XT-->EXT_I/O
MEMW	51	75	XT-->EXT_I/O
IOR	51	75	XT-->EXT_I/O
IOW	27	39	XT-->EXT_I/O
ALE	27	39	XT-->EXT_I/O
CLK	27	39	XT-->EXT_I/O
T/C	27	39	XT-->EXT_I/O
RESET	27	39	XT-->EXT_I/O
IRQ2-IRQ7	36	(*)	XT<--EXT_I/O
DRQ1-DRQ3	36	(*)	XT<--EXT_I/O
I/O CH RDY	36	51	XT<--EXT_I/O
I/O CH CK	36	51	XT<--EXT_I/O
D0-D7 (olvasás)	84	133	XT<--EXT_I/O
D0-D7 (írás)	19	27	XT-->EXT_I/O

(*) Az interrupt és DMA kérések aszinkron jellege miatt az azonosításig eltelt idő sokkal jobban függ a CPU foglaltságától, mint a késleltetéstől.

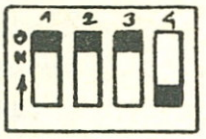
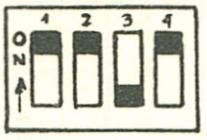
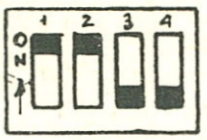
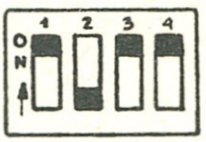
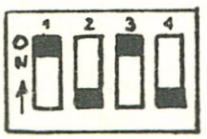
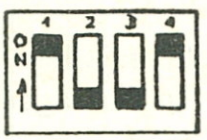
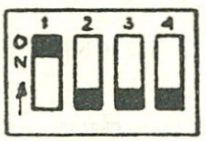
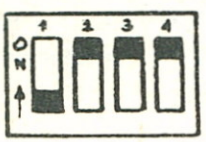
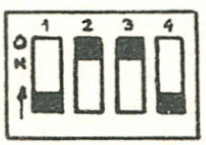
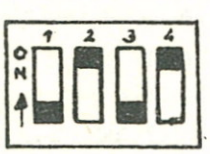
8-1 táblázat.
A buszkiterjesztés jeleinek késleltetései.

8.5.1. Az I/O kiterjesztő kártya (extender card).

A kártyán elhelyezett meghajtó áramkörök elegendő teljesítményűek ahhoz, hogy kiküszöböljék a hosszabbító kábel kapacitív hatásait. A gép I/O csatornájának vonalát a meghajtók 1 LS egységterheléssel terhelik.

Az extender kártyán található 'wait state' generátor egy-egy várakozási állapottal hosszabbítja meg a buszkiterjesztő egységben elhelyezett memóriák írás és olvasás ciklusait (a memória frissítési ciklusok ez alól kivételek). A 'wait state' generátor érvényességi címtartománya mikrokapcsolókkal állítható be. A mikrokapcsolók állásai az XT alaplemezen beépített maximális egybefüggő memória-területet jelölik a 8-8 ábra szerint. Az 1-4 jelölésű kapcsolók az A19-A16 címbiteknek felelnek meg. Várakozási állapotok a kapcsolók által beállított cím és a 0EFFFFh cím közötti tartományban kerülnek beiktatásra, függetlenül a memória típusától (ROM vagy RAM). Az utolsó F jelű szegmensre (0F0000h-0FFFFFh BIOS szegmens) vonatkozó műveletek nem tartalmaznak várakozási ciklusokat.

Kivitelét tekintve, az I/O kiterjesztő kártya négyrétegű nyomtatott áramköri lapra készült.

Rendszer memória		Memória szegmens
16K to 64K		1
96K to 128K		2
160K to 192K		3
224K to 256K		4
288K to 320K		5
352K to 384K		6
416K to 448K		7
480K to 512K		8
544K to 576K		9
608K to 640K		A

8-8 ábra.
Az I/O extender kártya mikrokapcsoló beállításai.

Az extender kártya programozása:

A kártyán található regiszterek többsége diagnosztikai célokat szolgál. A 8-2 táblázat a regisztereket sorolja fel, a hozzájuk tartozó I/O címekkel együtt. A kártyát a gép bekapcsolásakor az XT ROM BIOS inicializáló programja teszteli, majd engedélyezett állapotba helyezi. Az adó és vevő kártyákban a 0FXXXXh című memória írás műveletnél az alsó 16 címbit tárolásra kerül és diagnosztikai célokból visszaolvasható, ha a kártyák nincsenek tiltott állapotban. A visszaolvasás két lépésben történik, külön az alsó és külön a felső byte-ra vonatkozóan.

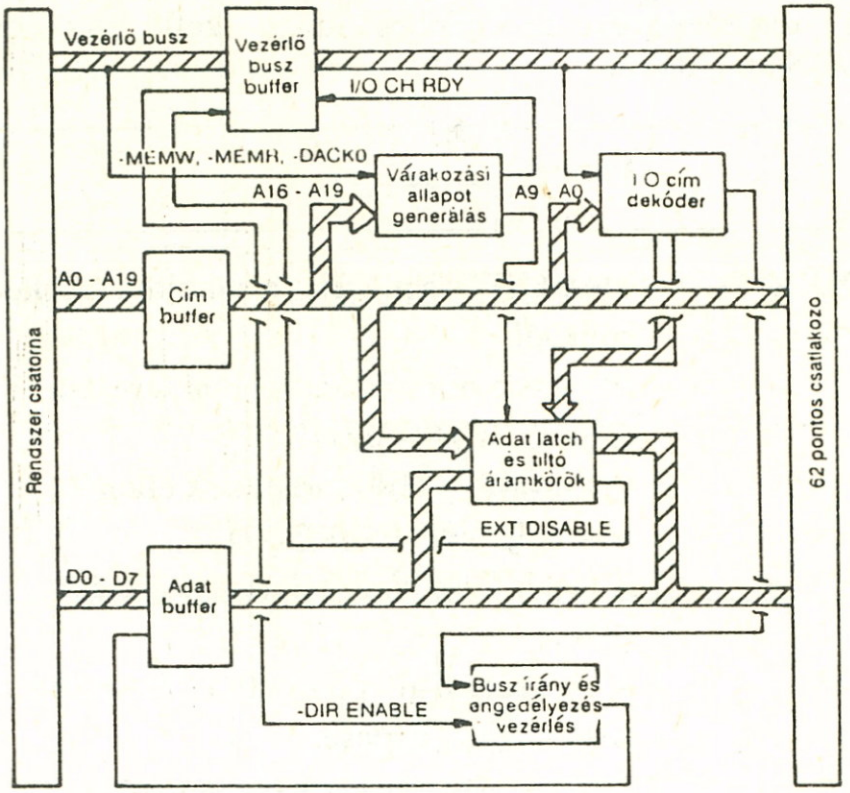
Az AT gépek a buszkiterjesztést nem támogatják. Az AT ROM BIOS programból hiányoznak az adó-vevő kártyákat inicializáló részek. Ha valaki mégis megpróbálná az XT buszkiterjesztő egységet az AT gép 6/8 MHz sebességű változatához illeszteni, a kártyák inicializálásáról és engedélyezéséről gondoskodnia kell. Az extender kártya blokkdiagramja a 8-9 ábrán látható.

Cím (hex)	Funkció
Memória FXXXX(*)	Memória írás művelet a címbitek tárolása céljából.
I/O 210	I/O írás a pufferelt adatbitek tárolása céljából (ED0-ED7).
I/O 210	I/O olvasás a tárolt adatbitek ellenőrzése céljából. (ED0-ED7)
I/O 211	A tárolt felső címbitek visszaolvasása (A8-A15)
I/O 211	A várakozási állapotot (I/O CH RDY) jelző bistabil törlése.
I/O 212	A tárolt alsó címbitek visszaolvasása (A0-A7)
I/O 213	I/O írás művelet D = 00 adat byte kiadással a buszkiterjesztés tiltása céljából.
I/O 213	I/O írás művelet D = 01 adat byte kiadással a buszkiterjesztés engedélyezése céljából.
I/O 213	A buszkiterjesztés státusz-byte kiolvasása D0 = engedélyezés/tiltás D1 = várakozási állapotot kérő flag

D2-D3 = nem használt
D4-D7 = a DIP kapcsoló állásai
(1 = OFF 0 = ON)

(*) Példa: Memória írás művelet 0F123:4 = 00
I/O olvasás a 211 címről = 12h
I/O olvasás a 212 címről = 34h

8-2 táblázat.
Az I/O buszkiterjesztő adó-kártya címei és funkciói.



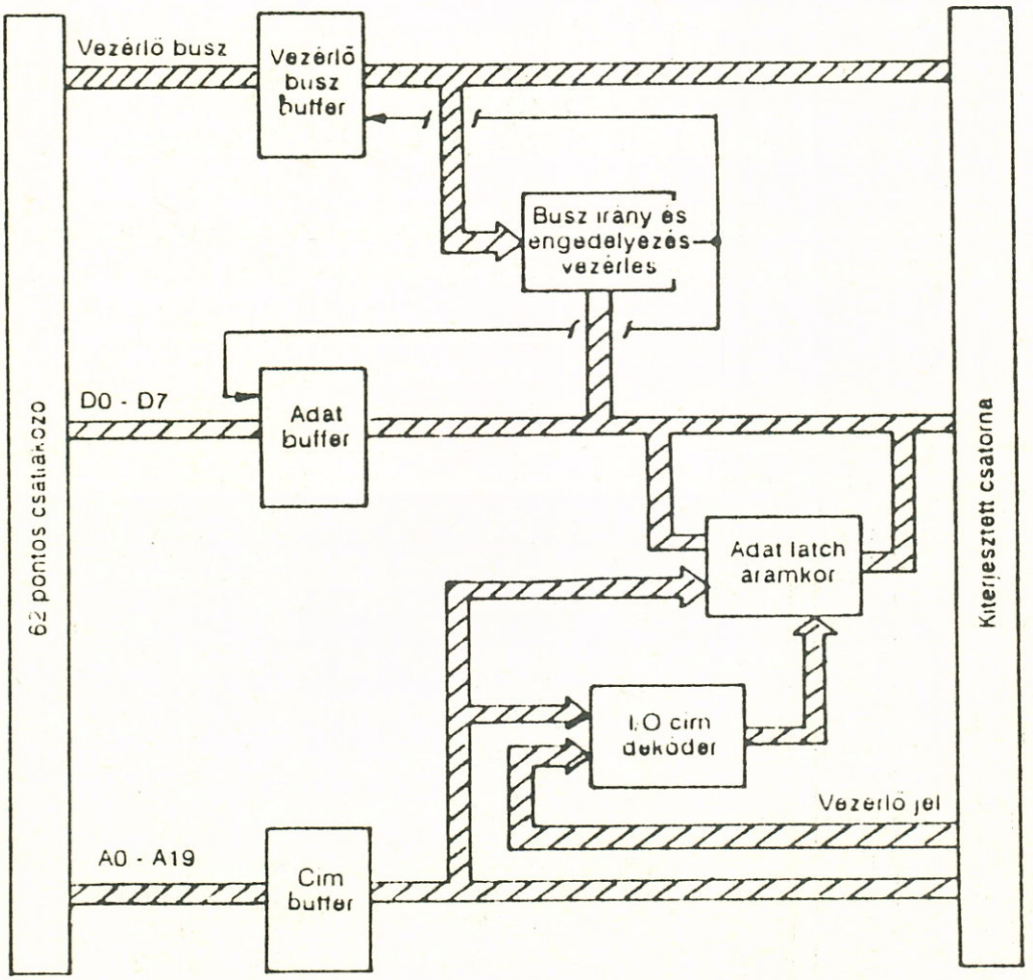
8-9 ábra.
Az I/O buszkiterjesztő adó-kártya blokkdiagramja.

8.5.2. A vevő kártya (receiver card)

A vevő kártya az extender kártyához hasonlóan négyrétegű nyomtatott áramköri lapra készült. A jelforgalom mind a buszkiterjesztés, mind az extender kártya irányában meghajtó áramkörökön keresztül történik. Az adatforgalom irányát meghatározó logikai áramkörök a kétirányú adat-buffert a megfelelő irányba állítják.

A kártyát a gép bekapcsolásakor az XT ROM BIOS inicializáló programja teszteli, majd engedélyezett állapotba helyezi.

A kártyán található regiszterek egy része diagnosztikai célokat szolgál. A vevő kártya blokkdiagramja a 8-10 ábrán látható. A 8-3 táblázat a vevő kártya címeit és a hozzárendelt műveleteket sorolja fel.



8-10. ábra.
Az I/O buszkiterjesztő vevő-kártya blokkdiagramja

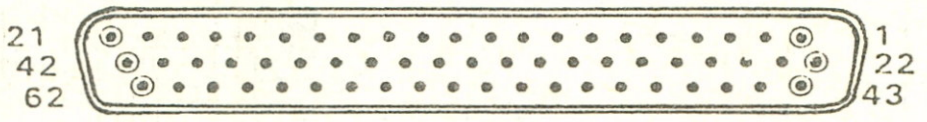
Cím (hex)	Funkció
Memória FXXXX(*)	Memória írás művelet a címbitek tárolása céljából.
I/O 214	I/O írás a kiterjesztett adatbitek tárolása céljából (D0-D7)
I/O 214	I/O olvasás a tárolt adatbitek ellenőrzése céljából (D0-D7)
I/O 215	A tárolt felső címbitek visszaolvasása

	(A8-A15)
I/O 216	A tárolt alsó címbitek visszaolvasása (A0-A7)

(*) Példa: Memória írás művelet 0F123:4 = 00
I/O olvasás a 215 címről = 12h
I/O olvasás a 216 címről = 34h

8-3 táblázat.
Az I/O buszkiterjesztő vevő-kártya címei és funkciói.

Az adó és vevő kártyák hátoldali csatlakozóinak a jelkiosztása megegyezik. A 8-11. ábra mutatja a csatlakozópontokhoz rendelt jelek elrendezését.



Pin	Signal	Pin	Signal	Pin	Signal
1	+E IRQ6	22	+E D5	43	+E IRQ7
2	+E DRQ2	23	+E DRQ1	44	+E D6
3	+E DIR	24	+E DRQ3	45	+E I/O CH RDY
4	+E ENABLE	25	RESERVED	46	+E IRQ3
5	+E CLK	26	+E ALE	47	+E D7
6	-E MEM IN EXP	27	+E T/C	48	+E D1
7	+E A17	28	+E RESET	49	-E I/O CH CK
8	+E A16	29	+E AEN	50	+E IRQ2
9	+E A5	30	+E A19	51	+E D0
10	-E DACK0	31	+E A14	52	+E D2
11	+E A15	32	+E A12	53	+E D4
12	+E A11	33	+E A18	54	+E IRQ5
13	+E A10	34	-E MEMR	55	+E IRQ4
14	+E A9	35	-E MEMW	56	+E D3
15	+E A1	36	+E A0	57	GND
16	+E A3	37	-E DACK3	58	GND
17	-E DACK1	38	+E A6	59	GND
18	+E A4	39	-E IOR	60	GND
19	-E DACK2	40	+E A8	61	GND
20	-E IOW	41	+E A2	62	GND
21	+E A13	42	+E A7		

E = Extended

8-11 ábra.
Az I/O buszkiterjesztő adó- és vevő-kártyák hátoldali csatlakozóinak jelkiosztása.

IRODALOMJEGYZÉK

Sargent, M.III. - Shoemaker, R.L.:
The IBM Personal Computer From the Inside Out
Addison - Wesley Publishing Co., 1984.

Norton, P.:
Inside the IBM PC
Robert J. Brady Co., 1983.

IBM Personal Computer XT
Technical Reference
IBM Corporation, Boca Raton FL, 1983.

IBM Personal Computer AT
Technical Reference
Revised Edition
IBM Corporation, Boca Raton FL, 1986.

MS-DOS 3.1 *Programmer's Reference Manual*
Microsoft Corporation, 1984.

The 8086 *Family User's Manual*
Intel Corporation.

The 8086 *Family User's Manual*
Numerics Supplement
Intel Corporation, 1980.

Stephen P. Morse - Douglas J. Albert
The 80286 architecture
Wiley Press, 1986

iAPX 86,88 Microprocessors
Intel Corporation (Component Data Catalog).

Component Data Catalog
Intel Corporation, 1986.

The TTL Data Book
Texas Instruments, 1987.

Függelék

A1.) 8086-8088-80286 utasításkészlet ABC sorrendben

A2.) 8087-80287 utasításkészlet ABC sorrendben
(A 8086-8088-80286 utasításkészlet kibővítése)

B1.) IBM PC karakterkészlet

B2.) IBM PC karakterkészlet decimális és hexadecimális kódokkal

B3.) ASCII vezérlő karakterek

B4.) Táblázat rajzoló karakterek

8086-8088-80286 utasításkészlet ABC sorrendben

Jelölések, magyarázatok:

A címképzést az opkód *mod* és *r/m* mezője határozza meg a következők szerint:
(*mod* az opkód2 vagy opkód3 legfelső két bitje)

- *mod* = 11 esetben az *r/m* mező *reg* mezőként kezelendő
- *mod* = 00 esetén az eltolás (displacement) = 0
- *mod* = 01 esetén az eltolás alsó byte-ja 16 bites előjel-kiterjesztetté válik, a felső byte hiányzik
- *mod* = 10 esetén az eltolás alsó és felső byte-ja is érvényes

- ha *r/m* = 000, akkor EA = (BX) + (SI) + DISP (EA:= Execution Address)
- ha *r/m* = 001, akkor EA = (BX) + (DI) + DISP
- ha *r/m* = 010, akkor EA = (BP) + (SI) + DISP
- ha *r/m* = 011, akkor EA = (BP) + (DI) + DISP
- ha *r/m* = 100, akkor EA = (SI) + DISP
- ha *r/m* = 101, akkor EA = (DI) + DISP
- ha *r/m* = 110, akkor EA = (BP) + DISP *
- ha *r/m* = 111, akkor EA = (BX) + DISP

* - kivéve, ha *mod* = 00 és *r/m* = 110, akkor EA = eltolás felső-eltolás alsó(16 bit offszet)

A *reg* regiszter mező:

16 bit (w = 1)		8 bit (w = 0)	
000	AX	000	AL
001	CX	001	CL
010	DX	010	DL
011	BX	011	BL
100	SP	100	AH
101	BP	101	CH
110	SI	110	DH
111	DI	111	BH

Az *sr* szegmensregiszter mező:

sr	szegmens
00	ES
01	CS
10	SS
11	DC

A flag-ek kódjai:

kód	funkció
1	flag beállítás
0	flag törlés
X	a flag a művelet eredménye szerint változik
U	határozatlan
R	a memóriából kerül helyettesítésre (pl. SAHF utasítás)
	a flag nem változott

Egyéb jelölések:

r	regisztertartalom
m	memóriatartalom
r\m	regiszter- vagy memóriatartalom
i	konstans
x	az opkód azon bitjei, melyek az utasítást nem befolyásolják

8086-8088-80286 utasításkészlet ABC sorrendben

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
AAA	37	55	00110111	-	-	AL	8	-	U...UUXUX
AAD	D5	213	11010101	00 001 010	-	AX	16	-	U...XXUXU
AAM	D4	212	11010100	00 001 010	-	AL	8	-	U...XXUXU
AAS	3F	63	00111111	-	-	AL	8	-	U...UUXUX
ADC	10	16	00010000	00 reg 110	++	r/m -> r	8	-	X...XXXXX
ADC	10	16	00010000	00 reg r/m	-	r/m -> r	8	-	X...XXXXX
ADC	10	16	00010000	01 reg r/m	+	r/m -> r	8	-	X...XXXXX
ADC	10	16	00010000	10 reg r/m	++	r/m -> r	8	-	X...XXXXX
ADC	10	16	00010000	11 reg r/m	-	r/m -> r	8	-	X...XXXXX
ADC	11	17	00010001	00 reg 110	++	r/m -> r	16	-	X...XXXXX
ADC	11	17	00010001	00 reg r/m	-	r/m -> r	16	-	X...XXXXX
ADC	11	17	00010001	01 reg r/m	+	r/m -> r	16	-	X...XXXXX
ADC	11	17	00010001	10 reg r/m	++	r/m -> r	16	-	X...XXXXX
ADC	11	17	00010001	11 reg r/m	-	r/m -> r	16	-	X...XXXXX
ADC	12	18	00010010	00 reg 110	++	r -> r\m	8	-	X...XXXXX
ADC	12	18	00010010	00 reg r/m	-	r -> r\m	8	-	X...XXXXX
ADC	12	18	00010010	01 reg r/m	+	r -> r\m	8	-	X...XXXXX
ADC	12	18	00010010	10 reg r/m	++	r -> r\m	8	-	X...XXXXX
ADC	12	18	00010010	11 reg r/m	-	r -> r\m	8	-	X...XXXXX
ADC	13	19	00010011	00 reg 110	++	r -> r\m	16	-	X...XXXXX
ADC	13	19	00010011	00 reg r/m	-	r -> r\m	16	-	X...XXXXX
ADC	13	19	00010011	01 reg r/m	+	r -> r\m	16	-	X...XXXXX
ADC	13	19	00010011	10 reg r/m	++	r -> r\m	16	-	X...XXXXX
ADC	13	19	00010011	11 reg r/m	-	r -> r\m	16	-	X...XXXXX
ADC	14	20	00010100	-	+	i -> AL	8	-	X...XXXXX
ADC	15	21	00010101	-	++	i -> AX	16	-	X...XXXXX
ADC	80	128	10000000	00 010 110	+++	i -> r\m	8	-	X...XXXXX
ADC	80	128	10000000	00 010 r/m	+	i -> r\m	8	-	X...XXXXX
ADC	80	128	10000000	01 010 r/m	++	i -> r\m	8	-	X...XXXXX
ADC	80	128	10000000	10 010 r/m	+++	i -> r\m	8	-	X...XXXXX
ADC	80	128	10000000	11 010 r/m	+	i -> r\m	8	-	X...XXXXX
ADC	81	129	10000001	00 010 110	++++	i -> r\m	16	-	X...XXXXX
ADC	81	129	10000001	00 010 r/m	++	i -> r\m	16	-	X...XXXXX
ADC	81	129	10000001	01 010 r/m	+++	i -> r\m	16	-	X...XXXXX
ADC	81	129	10000001	10 010 r/m	++++	i -> r\m	16	-	X...XXXXX
ADC	81	129	10000001	11 010 r/m	++	i -> r\m	16	-	X...XXXXX
ADC	82	130	10000010	00 010 110	+++	i -> r\m	8	-	X...XXXXX
ADC	82	130	10000010	00 010 r/m	+	i -> r\m	8	-	X...XXXXX
ADC	82	130	10000010	01 010 r/m	++	i -> r\m	8	-	X...XXXXX
ADC	82	130	10000010	10 010 r/m	+++	i -> r\m	8	-	X...XXXXX
ADC	82	130	10000010	11 010 r/m	+	i -> r\m	8	-	X...XXXXX
ADC	83	131	10000011	00 010 110	+++	i -> r\m	16	-	X...XXXXX
ADC	83	131	10000011	00 010 r/m	+	i -> r\m	16	-	X...XXXXX

mnemo	opkód1			opkód2	adat byte	adat-irány			bit	286	flag-ek ODITSZAPC
	hex	dec	bináris								
ADC	83	131	10000011	01 010 r/m	++	i	->	r\m	16	-	X...XXXXX
ADC	83	131	10000011	10 010 r/m	+++	i	->	r\m	16	-	X...XXXXX
ADC	83	131	10000011	11 010 r/m	+	i	->	r\m	16	-	X...XXXXX
ADD	00	0	00000000	00 reg 110	++	r/m	->	r	8	-	X...XXXXX
ADD	00	0	00000000	00 reg r/m	-	r/m	->	r	8	-	X...XXXXX
ADD	00	0	00000000	01 reg r/m	+	r/m	->	r	8	-	X...XXXXX
ADD	00	0	00000000	10 reg r/m	++	r/m	->	r	8	-	X...XXXXX
ADD	00	0	00000000	11 reg r/m	-	r/m	->	r	8	-	X...XXXXX
ADD	01	1	00000001	00 reg 110	++	r/m	->	r	16	-	X...XXXXX
ADD	01	1	00000001	00 reg r/m	-	r/m	->	r	16	-	X...XXXXX
ADD	01	1	00000001	01 reg r/m	+	r/m	->	r	16	-	X...XXXXX
ADD	01	1	00000001	10 reg r/m	++	r/m	->	r	16	-	X...XXXXX
ADD	01	1	00000001	11 reg r/m	-	r/m	->	r	16	-	X...XXXXX
ADD	02	2	00000010	00 reg 110	++	r	->	r\m	8	-	X...XXXXX
ADD	02	2	00000010	00 reg r/m	-	r	->	r\m	8	-	X...XXXXX
ADD	02	2	00000010	01 reg r/m	+	r	->	r\m	8	-	X...XXXXX
ADD	02	2	00000010	10 reg r/m	++	r	->	r\m	8	-	X...XXXXX
ADD	02	2	00000010	11 reg r/m	-	r	->	r\m	8	-	X...XXXXX
ADD	03	3	00000011	00 reg 110	++	r	->	r\m	16	-	X...XXXXX
ADD	03	3	00000011	00 reg r/m	-	r	->	r\m	16	-	X...XXXXX
ADD	03	3	00000011	01 reg r/m	+	r	->	r\m	16	-	X...XXXXX
ADD	03	3	00000011	10 reg r/m	++	r	->	r\m	16	-	X...XXXXX
ADD	03	3	00000011	11 reg r/m	-	r	->	r\m	16	-	X...XXXXX
ADD	04	4	00000100	-	+	i	->	AL	8	-	X...XXXXX
ADD	05	5	00000101	-	++	i	->	AX	16	-	X...XXXXX
ADD	80	128	10000000	00 000 110	+++	i	->	r\m	8	-	X...XXXXX
ADD	80	128	10000000	00 000 r/m	+	i	->	r\m	8	-	X...XXXXX
ADD	80	128	10000000	01 000 r/m	++	i	->	r\m	8	-	X...XXXXX
ADD	80	128	10000000	10 000 r/m	+++	i	->	r\m	8	-	X...XXXXX
ADD	80	128	10000000	11 000 r/m	+	i	->	r\m	8	-	X...XXXXX
ADD	81	129	10000001	00 000 110	++++	i	->	r\m	16	-	X...XXXXX
ADD	81	129	10000001	00 000 r/m	++	i	->	r\m	16	-	X...XXXXX
ADD	81	129	10000001	01 000 r/m	+++	i	->	r\m	16	-	X...XXXXX
ADD	81	129	10000001	10 000 r/m	++++	i	->	r\m	16	-	X...XXXXX
ADD	81	129	10000001	11 000 r/m	++	i	->	r\m	16	-	X...XXXXX
ADD	82	130	10000010	00 000 110	+++	i	->	r\m	8	-	X...XXXXX
ADD	82	130	10000010	00 000 r/m	+	i	->	r\m	8	-	X...XXXXX
ADD	82	130	10000010	01 000 r/m	++	i	->	r\m	8	-	X...XXXXX
ADD	82	130	10000010	10 000 r/m	+++	i	->	r\m	8	-	X...XXXXX
ADD	82	130	10000010	11 000 r/m	+	i	->	r\m	8	-	X...XXXXX
ADD	83	131	10000011	00 000 110	+++	i	->	r\m	16	-	X...XXXXX
ADD	83	131	10000011	00 000 r/m	+	i	->	r\m	16	-	X...XXXXX
ADD	83	131	10000011	01 000 r/m	++	i	->	r\m	16	-	X...XXXXX
ADD	83	131	10000011	10 000 r/m	+++	i	->	r\m	16	-	X...XXXXX
ADD	83	131	10000011	11 000 r/m	+	i	->	r\m	16	-	X...XXXXX
AND	20	32	00100000	00 reg 110	++	r/m	->	r	8	-	0...XXUX0
AND	20	32	00100000	00 reg r/m	-	r/m	->	r	8	-	0...XXUX0
AND	20	32	00100000	01 reg r/m	+	r/m	->	r	8	-	0...XXUX0
AND	20	32	00100000	10 reg r/m	++	r/m	->	r	8	-	0...XXUX0

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
AND	20	32	00100000	11 reg r/m	-	r/m -> r	8	-	0...XXUX0
AND	21	33	00100001	00 reg 110	++	r/m -> r	16	-	0...XXUX0
AND	21	33	00100001	00 reg r/m	-	r/m -> r	16	-	0...XXUX0
AND	21	33	00100001	01 reg r/m	+	r/m -> r	16	-	0...XXUX0
AND	21	33	00100001	10 reg r/m	++	r/m -> r	16	-	0...XXUX0
AND	21	33	00100001	11 reg r/m	-	r/m -> r	16	-	0...XXUX0
AND	22	34	00100010	00 reg 110	++	r -> r\m	8	-	0...XXUX0
AND	22	34	00100010	00 reg r/m	-	r -> r\m	8	-	0...XXUX0
AND	22	34	00100010	01 reg r/m	+	r -> r\m	8	-	0...XXUX0
AND	22	34	00100010	10 reg r/m	++	r -> r\m	8	-	0...XXUX0
AND	22	34	00100010	11 reg r/m	-	r -> r\m	8	-	0...XXUX0
AND	23	35	00100011	00 reg 110	++	r -> r\m	16	-	0...XXUX0
AND	23	35	00100011	00 reg r/m	-	r -> r\m	16	-	0...XXUX0
AND	23	35	00100011	01 reg r/m	+	r -> r\m	16	-	0...XXUX0
AND	23	35	00100011	10 reg r/m	++	r -> r\m	16	-	0...XXUX0
AND	23	35	00100011	11 reg r/m	-	r -> r\m	16	-	0...XXUX0
AND	24	36	00100100	-	+	i -> AL	8	-	0...XXUX0
AND	25	37	00100101	-	++	i -> AX	16	-	0...XXUX0
AND	80	128	10000000	00 100 110	+++	i -> r\m	8	-	0...XXUX0
AND	80	128	10000000	00 100 r/m	+	i -> r\m	8	-	0...XXUX0
AND	80	128	10000000	01 100 r/m	++	i -> r\m	8	-	0...XXUX0
AND	80	128	10000000	10 100 r/m	+++	i -> r\m	8	-	0...XXUX0
AND	80	128	10000000	11 100 r/m	+	i -> r\m	8	-	0...XXUX0
AND	81	129	10000001	00 100 110	++++	i -> r\m	16	-	0...XXUX0
AND	81	129	10000001	00 100 r/m	++	i -> r\m	16	-	0...XXUX0
AND	81	129	10000001	01 100 r/m	+++	i -> r\m	16	-	0...XXUX0
AND	81	129	10000001	10 100 r/m	++++	i -> r\m	16	-	0...XXUX0
AND	81	129	10000001	11 100 r/m	++	i -> r\m	16	-	0...XXUX0

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	bit	286	flag-ek ODITSZAPC
ARPL	63	99	01100011	00 reg 110	++	16	+	.....X...
ARPL	63	99	01100011	00 reg r/m	-	16	+	.....X...
ARPL	63	99	01100011	01 reg r/m	+	16	+	.....X...
ARPL	63	99	01100011	10 reg r/m	++	16	+	.....X...
ARPL	63	99	01100011	11 reg r/m	-	16	+	.....X...

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	intra szegmentális	inter szegmentális	dir.	286	flag-ek ODITSZAPC
BOUND	62	98	01100010	00 reg 110	++	-	+	-	+	.....
BOUND	62	98	01100010	00 reg r/m	-	-	+	-	+	.....
BOUND	62	98	01100010	01 reg r/m	+	-	+	-	+	.....
BOUND	62	98	01100010	10 reg r/m	++	-	+	-	+	.....
CALL	9A	154	10011010	-	++++	-	+	+	-	.....
CALL	E8	232	11101000	-	++	+	-	+	-	.....

mnemo	opkód1 hex	dec	bináris	opkód2	adat byte	intra szegmentális	inter szegmentális	dir.	286	flag-ek ODITSZAPC
CALL	FF	255	11111111	00 010 110	++	+	-	-	-	.....
CALL	FF	255	11111111	00 010 r/m	-	+	-	-	-	.....
CALL	FF	255	11111111	00 011 110	++	-	+	-	-	.....
CALL	FF	255	11111111	00 011 r/m	-	-	+	-	-	.....
CALL	FF	255	11111111	01 010 r/m	+	+	-	-	-	.....
CALL	FF	255	11111111	01 011 r/m	+	-	+	-	-	.....
CALL	FF	255	11111111	10 010 r/m	++	+	-	-	-	.....
CALL	FF	255	11111111	10 011 r/m	++	-	+	-	-	.....
CALL	FF	255	11111111	11 010 r/m	-	+	-	-	-	.....

mnemo	opkód1 hex	dec	bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
CBW	98	152	10011000	-	-	AL -> AX	16	-	.....
CLC	F8	248	11111000	-	-	-	-	-	0
CLD	FC	252	11111100	-	-	-	-	-	.0.....
CLI	FA	250	11111010	-	-	-	-	-	.0.....
CLTS	0F	15	00001111	00 000 110	-	-	-	+	.....
CMC	F5	245	11110101	-	-	-	-	-	X
CMP	38	56	00111000	00 reg 110	++	r/m <-> r	8	-	X...XXXX
CMP	38	56	00111000	00 reg r/m	-	r/m <-> r	8	-	X...XXXX
CMP	38	56	00111000	01 reg r/m	+	r/m <-> r	8	-	X...XXXX
CMP	38	56	00111000	10 reg r/m	++	r/m <-> r	8	-	X...XXXX
CMP	38	56	00111000	11 reg r/m	-	r/m <-> r	8	-	X...XXXX
CMP	39	57	00111001	00 reg 110	++	r/m <-> r	16	-	X...XXXX
CMP	39	57	00111001	00 reg r/m	-	r/m <-> r	16	-	X...XXXX
CMP	39	57	00111001	01 reg r/m	+	r/m <-> r	16	-	X...XXXX
CMP	39	57	00111001	10 reg r/m	++	r/m <-> r	16	-	X...XXXX
CMP	39	57	00111001	11 reg r/m	-	r/m <-> r	16	-	X...XXXX
CMP	3A	58	00111010	00 reg 110	++	r <-> r\m	8	-	X...XXXX
CMP	3A	58	00111010	00 reg r/m	-	r <-> r\m	8	-	X...XXXX
CMP	3A	58	00111010	01 reg r/m	+	r <-> r\m	8	-	X...XXXX
CMP	3A	58	00111010	10 reg r/m	++	r <-> r\m	8	-	X...XXXX
CMP	3A	58	00111010	11 reg r/m	-	r <-> r\m	8	-	X...XXXX
CMP	3B	59	00111011	00 reg 110	++	r <-> r\m	16	-	X...XXXX
CMP	3B	59	00111011	00 reg r/m	-	r <-> r\m	16	-	X...XXXX
CMP	3B	59	00111011	01 reg r/m	+	r <-> r\m	16	-	X...XXXX
CMP	3B	59	00111011	10 reg r/m	++	r <-> r\m	16	-	X...XXXX
CMP	3B	59	00111011	11 reg r/m	-	r <-> r\m	16	-	X...XXXX
CMP	3C	60	00111100	-	+	i <-> AL	8	-	X...XXXX
CMP	3D	61	00111101	-	++	i <-> AX	16	-	X...XXXX
CMP	80	128	10000000	00 111 110	+++	i <-> r\m	8	-	X...XXXX
CMP	80	128	10000000	00 111 r/m	+	i <-> r\m	8	-	X...XXXX
CMP	80	128	10000000	01 111 r/m	++	i <-> r\m	8	-	X...XXXX
CMP	80	128	10000000	10 111 r/m	+++	i <-> r\m	8	-	X...XXXX
CMP	80	128	10000000	11 111 r/m	+	i <-> r\m	8	-	X...XXXX

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
CMP	81	129	10000001	00 111 110	+++	i <-> r\m	16	-	X...XXXXX
CMP	81	129	10000001	00 111 r/m	+	i <-> r\m	16	-	X...XXXXX
CMP	81	129	10000001	01 111 r/m	+++	i <-> r\m	16	-	X...XXXXX
CMP	81	129	10000001	10 111 r/m	++++	i <-> r\m	16	-	X...XXXXX
CMP	81	129	10000001	11 111 r/m	++	i <-> r\m	16	-	X...XXXXX
CMP	82	130	10000010	00 111 110	+++	i <-> r\m	8	-	X...XXXXX
CMP	82	130	10000010	00 111 r/m	+	i <-> r\m	8	-	X...XXXXX
CMP	82	130	10000010	01 111 r/m	++	i <-> r\m	8	-	X...XXXXX
CMP	82	130	10000010	10 111 r/m	+++	i <-> r\m	8	-	X...XXXXX
CMP	82	130	10000010	11 111 r/m	+	i <-> r\m	8	-	X...XXXXX
CMP	83	131	10000011	00 111 110	+++	i <-> r\m	16	-	X...XXXXX
CMP	83	131	10000011	00 111 r/m	+	i <-> r\m	16	-	X...XXXXX
CMP	83	131	10000011	01 111 r/m	++	i <-> r\m	16	-	X...XXXXX
CMP	83	131	10000011	10 111 r/m	+++	i <-> r\m	16	-	X...XXXXX
CMP	83	131	10000011	11 111 r/m	+	i <-> r\m	16	-	X...XXXXX

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat-irány forrás	adat-irány cél	bit	286	flag-ek ODITSZAPC
CMPSB	A6	166	10100110	-	DS:SI	ES:DI	8	-	X...XXXXX
CMPSW	A7	167	10100111	-	DS:SI	ES:DI	16	-	X...XXXXX

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
CWD	99	153	10011001	-	-	AX -> AX,DX	32	-	.....
DAA	27	39	00100111	-	-	AL	8	-	X...XXXXX
DAS	2F	47	00101111	-	-	AL	8	-	U...XXXXX
DEC	48+x		01001reg	-	-	reg	16	-	X...XXXX.
DEC	FE	254	11111110	00 001 110	++	r\m	8	-	X...XXXX.
DEC	FE	254	11111110	00 001 r/m	-	r\m	8	-	X...XXXX.
DEC	FE	254	11111110	01 001 r/m	+	r\m	8	-	X...XXXX.
DEC	FE	254	11111110	10 001 r/m	++	r\m	8	-	X...XXXX.
DEC	FE	254	11111110	11 001 r/m	-	r\m	8	-	X...XXXX.
DEC	FF	255	11111111	00 001 110	++	r\m	16	-	X...XXXX.
DEC	FF	255	11111111	00 001 r/m	-	r\m	16	-	X...XXXX.
DEC	FF	255	11111111	01 001 r/m	+	r\m	16	-	X...XXXX.
DEC	FF	255	11111111	10 001 r/m	++	r\m	16	-	X...XXXX.
DEC	FF	255	11111111	11 001 r/m	-	r\m	16	-	X...XXXX.
DIV	F6	246	11110110	00 110 110	++	r\m -> AL	16	-	U...UUUUU
DIV	F6	246	11110110	00 110 r/m	-	r\m -> AL	16	-	U...UUUUU
DIV	F6	246	11110110	01 110 r/m	+	r\m -> AL	16	-	U...UUUUU
DIV	F6	246	11110110	10 110 r/m	++	r\m -> AL	16	-	U...UUUUU
DIV	F6	246	11110110	11 110 r/m	-	r\m -> AL	16	-	U...UUUUU
DIV	F7	247	11110111	00 110 110	++	r\m -> AX	32	-	U...UUUUU
DIV	F7	247	11110111	00 110 r/m	-	r\m -> AX	32	-	U...UUUUU
DIV	F7	247	11110111	01 110 r/m	+	r\m -> AX	32	-	U...UUUUU

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
DIV	F7	247	11110111	10 110 r/m	++	r\m -> AX	32	-	U...UUUUU
DIV	F7	247	11110111	11 110 r/m	-	r\m -> AX	32	-	U...UUUUU
ENTER	C8	200	11001000	-	+++			+	.....
ESC	D8+x		11011xxx	00 xxx 110	++	-	-	-	.....
ESC	D8+x		11011xxx	00 xxx r/m	-	-	-	-	.....
ESC	D8+x		11011xxx	01 xxx r/m	+	-	-	-	.....
ESC	D8+x		11011xxx	10 xxx r/m	++	-	-	-	.....
ESC	D8+x		11011xxx	11 xxx r/m	-	-	-	-	.....
HLT	F4	244	11110100	-	-	-	-	-	.....
IDIV	F6	246	11110110	00 111 110	++	r\m -> AL	16	-	U...UUUUU
IDIV	F6	246	11110110	00 111 r/m	-	r\m -> AL	16	-	U...UUUUU
IDIV	F6	246	11110110	01 111 r/m	+	r\m -> AL	16	-	U...UUUUU
IDIV	F6	246	11110110	10 111 r/m	++	r\m -> AL	16	-	U...UUUUU
IDIV	F6	246	11110110	11 111 r/m	-	r\m -> AL	16	-	U...UUUUU
IDIV	F7	247	11110111	00 111 110	++	r\m -> AX	32	-	U...UUUUU
IDIV	F7	247	11110111	00 111 r/m	-	r\m -> AX	32	-	U...UUUUU
IDIV	F7	247	11110111	01 111 r/m	+	r\m -> AX	32	-	U...UUUUU
IDIV	F7	247	11110111	10 111 r/m	++	r\m -> AX	32	-	U...UUUUU
IDIV	F7	247	11110111	11 111 r/m	-	r\m -> AX	32	-	U...UUUUU
IIMUL	69	105	01101001	00 reg 110	+++	r\m -> r	16	+	X...UUUUX
IIMUL	69	105	01101001	00 reg r/m	+	r\m -> r	16	+	X...UUUUX
IIMUL	69	105	01101001	01 reg r/m	++	r\m -> r	16	+	X...UUUUX
IIMUL	69	105	01101001	10 reg r/m	+++	r\m -> r	16	+	X...UUUUX
IIMUL	69	105	01101001	11 reg r/m	+	r\m -> r	16	+	X...UUUUX
IIMUL	6B	107	01101011	00 reg 110	++++	r\m -> r	16	+	X...UUUUX
IIMUL	6B	107	01101011	00 reg r/m	++	r\m -> r	16	+	X...UUUUX
IIMUL	6B	107	01101011	01 reg r/m	+++	r\m -> r	16	+	X...UUUUX
IIMUL	6B	107	01101011	10 reg r/m	++++	r\m -> r	16	+	X...UUUUX
IIMUL	6B	107	01101011	11 reg r/m	++	r\m -> r	16	+	X...UUUUX
IMUL	F6	246	11110110	00 101 110	++	r\m -> AL	8	-	X...UUUUX
IMUL	F6	246	11110110	00 101 r/m	-	r\m -> AL	8	-	X...UUUUX
IMUL	F6	246	11110110	01 101 r/m	+	r\m -> AL	8	-	X...UUUUX
IMUL	F6	246	11110110	10 101 r/m	++	r\m -> AL	8	-	X...UUUUX
IMUL	F6	246	11110110	11 101 r/m	-	r\m -> AL	8	-	X...UUUUX
IMUL	F7	247	11110111	00 101 110	++	r\m -> AX	16	-	X...UUUUX
IMUL	F7	247	11110111	00 101 r/m	-	r\m -> AX	16	-	X...UUUUX
IMUL	F7	247	11110111	01 101 r/m	+	r\m -> AX	16	-	X...UUUUX
IMUL	F7	247	11110111	10 101 r/m	++	r\m -> AX	16	-	X...UUUUX
IMUL	F7	247	11110111	11 101 r/m	-	r\m -> AX	16	-	X...UUUUX
IN	E4	228	11100100	-	+	port -> AL	8	-	.....
IN	E5	229	11100101	-	+	port -> AX	16	-	.....
IN	EC	236	11101100	-	-	port -> AL	8	-	.....
IN	ED	237	11101101	-	-	port -> AX	16	-	.....
INC	40+x		01000reg	-	-	reg	16	-	X...XXXX.
INC	FE	254	11111110	00 000 110	++	r\m	8	-	X...XXXX.

mnemo	opkód1 hex	dec	bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
INC	FE	254	11111110	00 000 r/m	-	r\m	8	-	X...XXXX.
INC	FE	254	11111110	01 000 r/m	+	r\m	8	-	X...XXXX.
INC	FE	254	11111110	10 000 r/m	++	r\m	8	-	X...XXXX.
INC	FE	254	11111110	11 000 r/m	-	r\m	8	-	X...XXXX.
INC	FF	255	11111111	00 000 110	++	r\m	16	-	X...XXXX.
INC	FF	255	11111111	00 000 r/m	-	r\m	16	-	X...XXXX.
INC	FF	255	11111111	01 000 r/m	+	r\m	16	-	X...XXXX.
INC	FF	255	11111111	10 000 r/m	++	r\m	16	-	X...XXXX.
INC	FF	255	11111111	11 000 r/m	-	r\m	16	-	X...XXXX.

mnemo	opkód1 hex	dec	bináris	opkód2	adat-irány forrás	adat-irány cél	bit	286	flag-ek ODITSZAPC
INSB	66	102	01100110	-	0:DX	ES:DI	8	+	.....
INSW	67	103	01100111	-	0:DX	ES:DI	16	+	.....

mnemo	opkód1 hex	dec	bináris	opkód2	adat byte	intra szegmentális	inter	dir.	286	flag-ek ODITSZAPC
INT	CD	205	11001101	-	+	-	+	-	-	..00.....
INTR	CC	204	11001100	-	-	-	+	-	-	..00.....
INTO	CE	206	11001110	-	-	-	+	-	-	..00.....
IRET	CF	207	11001111	-	-	-	+	-	-	RRRRRRRRR
JB	72	114	01110010	-	+	+	-	+	-	.....
JBE	76	118	01110110	-	+	+	-	+	-	.....
JCXZ	E3	227	11100011	-	+	+	-	+	-	.....
JE	74	116	01110100	-	+	+	-	+	-	.....
JL	7C	124	01111100	-	+	+	-	+	-	.....
JLE	7E	126	01111110	-	+	+	-	+	-	.....
JMP	E9	233	11101001	-	++	-	+	+	-	.....
JMP	EA	234	11101010	-	++++	-	+	+	-	.....
JMP	EB	235	11101011	-	+	-	+	+	-	.....
JMP	FF	255	11111111	00 100 110	++	+	-	-	-	.....
JMP	FF	255	11111111	00 100 r/m	-	+	-	-	-	.....
JMP	FF	255	11111111	00 101 110	++	-	+	-	-	.....
JMP	FF	255	11111111	00 101 r/m	-	-	+	-	-	.....
JMP	FF	255	11111111	01 100 r/m	+	+	-	-	-	.....
JMP	FF	255	11111111	01 101 r/m	+	-	+	-	-	.....
JMP	FF	255	11111111	10 100 r/m	++	+	-	-	-	.....
JMP	FF	255	11111111	10 101 r/m	++	-	+	-	-	.....
JMP	FF	255	11111111	11 100 r/m	-	+	-	-	-	.....

mnemo	opkód1 hex	dec	bináris	opkód2	adat byte	intra szegmentális	inter	286 dir.		flag-ek ODITSZAPC
JNB	73	115	01110011	-	+	+	-	+	-	.....
JNBE	77	119	01110111	-	+	+	-	+	-	.....
JNE	75	117	01110101	-	+	+	-	+	-	.....
JNL	7D	125	01111101	-	+	+	-	+	-	.....
JNLE	7F	127	01111111	-	+	+	-	+	-	.....
JNO	71	113	01110001	-	+	+	-	+	-	.....
JNP	7B	123	01111011	-	+	+	-	+	-	.....
JNS	79	121	01111001	-	+	+	-	+	-	.....
JO	70	112	01110000	-	+	+	-	+	-	.....
JP	7A	122	01111010	-	+	+	-	+	-	.....
JS	78	120	01111000	-	+	+	-	+	-	.....
LAHF	9F	159	10001111	-	-	flag-> AH		8	-	.....

mnemo	opkód1 hex	dec	bináris	opkód2	opkód3	adat byte	bit	286	flag-ek ODITSZAPC
LAR	0F	15	00001111	00000010	00 reg 110	++	16	+	.....X...
LAR	0F	15	00001111	00000010	00 reg r/m	-	16	+	.....X...
LAR	0F	15	00001111	00000010	01 reg r/m	+	16	+	.....X...
LAR	0F	15	00001111	00000010	10 reg r/m	++	16	+	.....X...
LAR	0F	15	00001111	00000010	11 reg r/m	-	16	+	.....X...

mnemo	opkód1 hex	dec	bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
LDS	C5	197	11000101	00 reg 110	++	m -> r	32	-	.....
LDS	C5	197	11000101	00 reg r/m	-	m -> r	32	-	.....
LDS	C5	197	11000101	01 reg r/m	+	m -> r	32	-	.....
LDS	C5	197	11000101	10 reg r/m	++	m -> r	32	-	.....
LEA	8D	141	10001101	00 reg 110	++	m -> r	16	-	.....
LEA	8D	141	10001101	00 reg r/m	-	m -> r	16	-	.....
LEA	8D	141	10001101	01 reg r/m	+	m -> r	16	-	.....
LEA	8D	141	10001101	10 reg r/m	++	m -> r	16	-	.....
LEA	8D	141	10001101	11 reg r/m	-	m -> r	16	-	.....
LEAVE	C9	201	11001001	-	-			+	.....
LES	C4	196	11000100	00 reg 110	++	m -> r	32	-	.....
LES	C4	196	11000100	00 reg r/m	-	m -> r	32	-	.....
LES	C4	196	11000100	01 reg r/m	+	m -> r	32	-	.....
LES	C4	196	11000100	10 reg r/m	++	m -> r	32	-	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	opkód3	adat byte	bit	286	flag-ek ODITSZAPC
LGDT	0F	15	00001111	00000001	00 010 110	++	48	+	.....
LGDT	0F	15	00001111	00000001	00 010 r/m	-	48	+	.....
LGDT	0F	15	00001111	00000001	01 010 r/m	+	48	+	.....
LGDT	0F	15	00001111	00000001	10 010 r/m	++	48	+	.....
LIDT	0F	15	00001111	00000001	00 011 110	++	48	+	.....
LIDT	0F	15	00001111	00000001	00 011 r/m	-	48	+	.....
LIDT	0F	15	00001111	00000001	01 011 r/m	+	48	+	.....
LIDT	0F	15	00001111	00000001	10 011 r/m	++	48	+	.....
LLDT	0F	15	00001111	00000000	00 010 110	++	16	+	.....
LLDT	0F	15	00001111	00000000	00 010 r/m	-	16	+	.....
LLDT	0F	15	00001111	00000000	01 010 r/m	+	16	+	.....
LLDT	0F	15	00001111	00000000	10 010 r/m	++	16	+	.....
LLDT	0F	15	00001111	00000000	11 010 r/m	-	16	+	.....
LMSW	0F	15	00001111	00000001	00 110 110	++	16	+	.....
LMSW	0F	15	00001111	00000001	00 110 r/m	-	16	+	.....
LMSW	0F	15	00001111	00000001	01 110 r/m	+	16	+	.....
LMSW	0F	15	00001111	00000001	10 110 r/m	++	16	+	.....
LMSW	0F	15	00001111	00000001	11 110 r/m	-	16	+	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
LOCK	F0	240	11110000	-	-	-	-	-	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat-irány forrás	adat-irány cél	bit	286	flag-ek ODITSZAPC
LODSB	AC	172	10101100	-	DS:SI	AL	8	-	.....
LODSW	AD	173	10101101	-	DS:SI	AX	16	-	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	intra szegmentális	inter dir.	286	flag-ek ODITSZAPC
LOOP	E2	226	11100010	-	+	+	-	+ -	.....
LOOPNZ	E0	224	11100000	-	+	+	-	+ -	.....
LOOPZ	E1	225	11100001	-	+	+	-	+ -	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	opkód3	adat byte	bit	286	flag-ek ODITSZAPC
LSL	0F	15	00001111	00000011	00 reg 110	++	16	+	.....X...
LSL	0F	15	00001111	00000011	00 reg r/m	-	16	+	.....X...
LSL	0F	15	00001111	00000011	01 reg r/m	+	16	+	.....X...
LSL	0F	15	00001111	00000011	10 reg r/m	++	16	+	.....X...
LSL	0F	15	00001111	00000011	11 reg r/m	-	16	+	.....X...
LTR	0F	15	00001111	00000000	00 011 110	++	16	+	.....X...

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	opkód3	adat byte	bit	286	flag-ek ODITSZAPC
LTR	0F	15	00001111	00000000	00 011 r/m	-	16	+	.....X...
LTR	0F	15	00001111	00000000	01 011 r/m	+	16	+	.....X...
LTR	0F	15	00001111	00000000	10 011 r/m	++	16	+	.....X...
LTR	0F	15	00001111	00000000	11 011 r/m	-	16	+	.....X...

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
MOV	88	136	10001000	00 reg 110	++	r -> r\m	8	-	.....
MOV	88	136	10001000	00 reg r/m	-	r -> r\m	8	-	.....
MOV	88	136	10001000	01 reg r/m	+	r -> r\m	8	-	.....
MOV	88	136	10001000	10 reg r/m	++	r -> r\m	8	-	.....
MOV	88	136	10001000	11 reg r/m	-	r -> r\m	8	-	.....
MOV	89	137	10001001	00 reg 110	++	r -> r\m	16	-	.....
MOV	89	137	10001001	00 reg r/m	-	r -> r\m	16	-	.....
MOV	89	137	10001001	01 reg r/m	+	r -> r\m	16	-	.....
MOV	89	137	10001001	10 reg r/m	++	r -> r\m	16	-	.....
MOV	89	137	10001001	11 reg r/m	-	r -> r\m	16	-	.....
MOV	8A	138	10001010	00 reg 110	++	r/m -> r	8	-	.....
MOV	8A	138	10001010	00 reg r/m	-	r/m -> r	8	-	.....
MOV	8A	138	10001010	01 reg r/m	+	r/m -> r	8	-	.....
MOV	8A	138	10001010	10 reg r/m	++	r/m -> r	8	-	.....
MOV	8A	138	10001010	11 reg r/m	-	r/m -> r	8	-	.....
MOV	8B	139	10001011	00 reg 110	++	r/m -> r	16	-	.....
MOV	8B	139	10001011	00 reg r/m	-	r/m -> r	16	-	.....
MOV	8B	139	10001011	01 reg r/m	+	r/m -> r	16	-	.....
MOV	8B	139	10001011	10 reg r/m	++	r/m -> r	16	-	.....
MOV	8B	139	10001011	11 reg r/m	-	r/m -> r	16	-	.....
MOV	8C	140	10001100	00 0sr 110	++	sr -> r\m	16	-	.....
MOV	8C	140	10001100	00 0sr r/m	-	sr -> r\m	16	-	.....
MOV	8C	140	10001100	01 0sr r/m	+	sr -> r\m	16	-	.....
MOV	8C	140	10001100	10 0sr r/m	++	sr -> r\m	16	-	.....
MOV	8C	140	10001100	11 0sr r/m	-	sr -> r\m	16	-	.....
MOV	8E	142	10001110	00 0sr 110	++	r/m -> sr	16	-	.....
MOV	8E	142	10001110	00 0sr r/m	-	r/m -> sr	16	-	.....
MOV	8E	142	10001110	01 0sr r/m	+	r/m -> sr	16	-	.....
MOV	8E	142	10001110	10 0sr r/m	++	r/m -> sr	16	-	.....
MOV	8E	142	10001110	11 0sr r/m	-	r/m -> sr	16	-	.....
MOV	A0	160	10100000	-	++	m -> AL	8	-	.....
MOV	A1	161	10100001	-	++	m -> AX	16	-	.....
MOV	A2	162	10100010	-	+	AL -> m	8	-	.....
MOV	A3	163	10100011	-	++	AX -> m	16	-	.....
MOV	B0+x		1011reg	-	+	i -> r	8	-	.....
MOV	B8+x		1011reg	-	++	i -> r	16	-	.....
MOV	C6	198	11000110	00 reg 110	+++	i -> r\m	8	-	.....
MOV	C6	198	11000110	00 reg r/m	+	i -> r\m	8	-	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
MOV	C6	198	11000110	01 reg r/m	++	i -> r\m	8	-	.....
MOV	C6	198	11000110	10 reg r/m	+++	i -> r\m	8	-	.....
MOV	C6	198	11000110	11 reg r/m	+	i -> r\m	8	-	.....
MOV	C7	199	11000111	00 reg 110	++++	i -> r\m	16	-	.....
MOV	C7	199	11000111	00 reg r/m	++	i -> r\m	16	-	.....
MOV	C7	199	11000111	01 reg r/m	+++	i -> r\m	16	-	.....
MOV	C7	199	11000111	10 reg r/m	++++	i -> r\m	16	-	.....
MOV	C7	199	11000111	11 reg r/m	++	i -> r\m	16	-	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat-irány forrás	adat-irány cél	bit	286	flag-ek ODITSZAPC
MOVSB	A4	164	10100100	-	DS:SI	ES:DI	8	-	.....
MOVSW	A5	165	10100101	-	DS:SI	ES:DI	16	-	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
MUL	F6	246	11110110	00 100 110	++	r\m -> AL	8	-	X...UUUUX
MUL	F6	246	11110110	00 100 r/m	-	r\m -> AL	8	-	X...UUUUX
MUL	F6	246	11110110	01 100 r/m	+	r\m -> AL	8	-	X...UUUUX
MUL	F6	246	11110110	10 100 r/m	++	r\m -> AL	8	-	X...UUUUX
MUL	F6	246	11110110	11 100 r/m	-	r\m -> AL	8	-	X...UUUUX
MUL	F7	247	11110111	00 100 110	++	r\m -> AX	16	-	X...UUUUX
MUL	F7	247	11110111	00 100 r/m	-	r\m -> AX	16	-	X...UUUUX
MUL	F7	247	11110111	01 100 r/m	+	r\m -> AX	16	-	X...UUUUX
MUL	F7	247	11110111	10 100 r/m	++	r\m -> AX	16	-	X...UUUUX
MUL	F7	247	11110111	11 100 r/m	-	r\m -> AX	16	-	X...UUUUX
NEG	F6	246	11110110	00 011 110	+++	r\m	8	-	X...XXXX1
NEG	F6	246	11110110	00 011 r/m	+	r\m	8	-	X...XXXX1
NEG	F6	246	11110110	01 011 r/m	++	r\m	8	-	X...XXXX1
NEG	F6	246	11110110	10 011 r/m	+++	r\m	8	-	X...XXXX1
NEG	F6	246	11110110	11 011 r/m	+	r\m	8	-	X...XXXX1
NEG	F7	247	11110111	00 011 110	+++	r\m	16	-	X...XXXX1
NEG	F7	247	11110111	00 011 r/m	+	r\m	16	-	X...XXXX1
NEG	F7	247	11110111	01 011 r/m	++	r\m	16	-	X...XXXX1
NEG	F7	247	11110111	10 011 r/m	+++	r\m	16	-	X...XXXX1
NEG	F7	247	11110111	11 011 r/m	+	r\m	16	-	X...XXXX1
NOT	F6	246	11110110	00 010 110	+++	r\m	8	-	.....
NOT	F6	246	11110110	00 010 r/m	+	r\m	8	-	.....
NOT	F6	246	11110110	01 010 r/m	++	r\m	8	-	.....
NOT	F6	246	11110110	10 010 r/m	+++	r\m	8	-	.....
NOT	F6	246	11110110	11 010 r/m	+	r\m	8	-	.....
NOT	F7	247	11110111	00 010 110	+++	r\m	16	-	.....
NOT	F7	247	11110111	00 010 r/m	+	r\m	16	-	.....
NOT	F7	247	11110111	01 010 r/m	++	r\m	16	-	.....
NOT	F7	247	11110111	10 010 r/m	+++	r\m	16	-	.....
NOT	F7	247	11110111	11 010 r/m	+	r\m	16	-	.....
OR	08	8	00001000	00 reg 110	++	r/m -> r	8	-	0...XXUX0

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
OR	08	8	00001000	00 reg r/m	-	r/m -> r	8	-	0...XXUX0
OR	08	8	00001000	01 reg r/m	+	r/m -> r	8	-	0...XXUX0
OR	08	8	00001000	10 reg r/m	++	r/m -> r	8	-	0...XXUX0
OR	08	8	00001000	11 reg r/m	-	r/m -> r	8	-	0...XXUX0
OR	09	9	00001001	00 reg 110	++	r/m -> r	16	-	0...XXUX0
OR	09	9	00001001	00 reg r/m	-	r/m -> r	16	-	0...XXUX0
OR	09	9	00001001	01 reg r/m	+	r/m -> r	16	-	0...XXUX0
OR	09	9	00001001	10 reg r/m	++	r/m -> r	16	-	0...XXUX0
OR	09	9	00001001	11 reg r/m	-	r/m -> r	16	-	0...XXUX0
OR	0A	10	00001010	00 reg 110	++	r -> r\m	8	-	0...XXUX0
OR	0A	10	00001010	00 reg r/m	-	r -> r\m	8	-	0...XXUX0
OR	0A	10	00001010	01 reg r/m	+	r -> r\m	8	-	0...XXUX0
OR	0A	10	00001010	10 reg r/m	++	r -> r\m	8	-	0...XXUX0
OR	0A	10	00001010	11 reg r/m	-	r -> r\m	8	-	0...XXUX0
OR	0B	11	00001011	00 reg 110	++	r -> r\m	16	-	0...XXUX0
OR	0B	11	00001011	00 reg r/m	-	r -> r\m	16	-	0...XXUX0
OR	0B	11	00001011	01 reg r/m	+	r -> r\m	16	-	0...XXUX0
OR	0B	11	00001011	10 reg r/m	++	r -> r\m	16	-	0...XXUX0
OR	0B	11	00001011	11 reg r/m	-	r -> r\m	16	-	0...XXUX0
OR	0C	12	00001100	-	+	i -> AL	8	-	0...XXUX0
OR	0D	13	00001101	-	++	i -> AX	16	-	0...XXUX0
OR	80	128	10000000	00 001 110	+++	i -> r\m	8	-	0...XXUX0
OR	80	128	10000000	00 001 r/m	+	i -> r\m	8	-	0...XXUX0
OR	80	128	10000000	01 001 r/m	++	i -> r\m	8	-	0...XXUX0
OR	80	128	10000000	10 001 r/m	+++	i -> r\m	8	-	0...XXUX0
OR	80	128	10000000	11 001 r/m	+	i -> r\m	8	-	0...XXUX0
OR	81	129	10000001	00 001 110	++++	i -> r\m	16	-	0...XXUX0
OR	81	129	10000001	00 001 r/m	++	i -> r\m	16	-	0...XXUX0
OR	81	129	10000001	01 001 r/m	+++	i -> r\m	16	-	0...XXUX0
OR	81	129	10000001	10 001 r/m	++++	i -> r\m	16	-	0...XXUX0
OR	81	129	10000001	11 001 r/m	++	i -> r\m	16	-	0...XXUX0
OUT	E6	230	11100110	-	+	AL <- port	8	-	.....
OUT	E7	231	11100111	-	+	AX <- port	16	-	.....
OUT	EE	238	11101110	-	+	AL <- port	8	-	.....
OUT	EF	239	11101111	-	+	AX <- port	16	-	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat-irány forrás	adat-irány cél	bit	286	flag-ek ODITSZAPC
OUTSB	6E	110	01101110	-	DS:SI	0:DX	8	+	.....
OUTSW	6F	111	01101111	-	DS:SI	0:DX	16	+	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
POP			000sr111	-	-	stack -> sr	16	-	.....
POP	58+x		01011reg	-	-	stack -> r	16	-	.....
POP	8F	143	10001111	00 000 110	++	stack -> m	16	-	.....
POP	8F	143	10001111	00 000 r/m	-	stack -> m	16	-	.....
POP	8F	143	10001111	01 000 r/m	+	stack -> m	16	-	.....
POP	8F	143	10001111	10 000 r/m	++	stack -> m	16	-	.....
POP	8F	143	10001111	11 000 r/m	-	stack -> m	16	-	.....
POPA	61	97	01100001	-	-	stack -> sr	256	+	.....
POPF	9D	157	10001101	-	-	stack->flag	16	-	RRRRRRRRR
PUSH			000sr110	-	-	sr -> stack	16	-	.....
PUSH	50+x		01010reg	-	-	r -> stack	16	-	.....
PUSH	6A	106	01101000	-	++	i -> stack	16	-	.....
PUSH	6C	108	01101010	-	+	i -> stack	16	-	.....
PUSH	FF	255	11111111	00 110 110	++	m -> stack	16	-	.....
PUSH	FF	255	11111111	00 110 r/m	-	m -> stack	16	-	.....
PUSH	FF	255	11111111	01 110 r/m	+	m -> stack	16	-	.....
PUSH	FF	255	11111111	10 110 r/m	++	m -> stack	16	-	.....
PUSH	FF	255	11111111	11 110 r/m	-	m -> stack	16	-	.....
PUSHA	60	96	01100000	-	-	r -> stack	256	+	.....
PUSHF	9C	156	10001100	-	-	flag-> stack	16	-	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	forgatás 1	forgatás CL	bit	286	flag-ek ODITSZAPC
RCL	C0	192	11000000	00 010 110	++		+	8	+	.X.....X
RCL	C0	192	11000000	00 010 r/m	-		+	8	+	.X.....X
RCL	C0	192	11000000	01 010 r/m	+		+	8	+	.X.....X
RCL	C0	192	11000000	10 010 r/m	++		+	8	+	.X.....X
RCL	C0	192	11000000	11 010 r/m	-		+	8	+	.X.....X
RCL	C1	193	11000001	00 010 110	++		+	16	+	.X.....X
RCL	C1	193	11000001	00 010 r/m	-		+	16	+	.X.....X
RCL	C1	193	11000001	01 010 r/m	+		+	16	+	.X.....X
RCL	C1	193	11000001	10 010 r/m	++		+	16	+	.X.....X
RCL	C1	193	11000001	11 010 r/m	-		+	16	+	.X.....X
RCL	D0	208	11010000	00 010 110	++	+		8	-	X.....X
RCL	D0	208	11010000	00 010 r/m	-	+		8	-	X.....X
RCL	D0	208	11010000	01 010 r/m	+	+		8	-	X.....X
RCL	D0	208	11010000	10 010 r/m	++	+		8	-	X.....X
RCL	D0	208	11010000	11 010 r/m	-	+		8	-	X.....X
RCL	D1	209	11010001	00 010 110	++	+		16	-	X.....X
RCL	D1	209	11010001	00 010 r/m	-	+		16	-	X.....X
RCL	D1	209	11010001	01 010 r/m	+	+		16	-	X.....X
RCL	D1	209	11010001	10 010 r/m	++	+		16	-	X.....X
RCL	D1	209	11010001	11 010 r/m	-	+		16	-	X.....X

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	forgatás 1	forgatás CL	bit	286	flag-ek ODITSZAPC
RCL	D2	210	11010010	00 010 110	++		+	8	-	X.....X
RCL	D2	210	11010010	00 010 r/m	-		+	8	-	X.....X
RCL	D2	210	11010010	01 010 r/m	+		+	8	-	X.....X
RCL	D2	210	11010010	10 010 r/m	++		+	8	-	X.....X
RCL	D2	210	11010010	11 010 r/m	-		+	8	-	X.....X
RCL	D3	211	11010011	00 010 110	++		+	16	-	X.....X
RCL	D3	211	11010011	00 010 r/m	-		+	16	-	X.....X
RCL	D3	211	11010011	01 010 r/m	+		+	16	-	X.....X
RCL	D3	211	11010011	10 010 r/m	++		+	16	-	X.....X
RCL	D3	211	11010011	11 010 r/m	-		+	16	-	X.....X

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat-irány forrás	adat-irány cél	bit	286	flag-ek ODITSZAPC
REPNZ CMPSB	F2	242	11110010	10 100 110	DS:SI	ES:DI	8	-	.....
REPZ CMPSB	F3	243	11110011	10 100 110	DS:SI	ES:DI	8	-	.....
REPNZ COMPW	F2	242	11110010	10 100 111	DS:SI	ES:DI	16	-	.....
REPZ COMPW	F3	243	11110011	10 100 111	DS:SI	ES:DI	16	-	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	forgatás 1	forgatás CL	bit	286	flag-ek ODITSZAPC
RCR	C0	192	11000000	00 011 110	++		+	8	+	.X.....X
RCR	C0	192	11000000	00 011 r/m	-		+	8	+	.X.....X
RCR	C0	192	11000000	01 011 r/m	+		+	8	+	.X.....X
RCR	C0	192	11000000	10 011 r/m	++		+	8	+	.X.....X
RCR	C0	192	11000000	11 011 r/m	-		+	8	+	.X.....X
X.....X										
RCR	C1	193	11000001	00 011 110	++		+	16	+	.X.....X
RCR	C1	193	11000001	00 011 r/m	-		+	16	+	.X.....X
RCR	C1	193	11000001	01 011 r/m	+		+	16	+	.X.....X
RCR	C1	193	11000001	10 011 r/m	++		+	16	+	.X.....X
RCR	C1	193	11000001	11 011 r/m	-		+	16	+	.X.....X
RCR	D0	208	11010000	00 011 110	++	+		8	-	X.....X
RCR	D0	208	11010000	00 011 r/m	-	+		8	-	X.....X
RCR	D0	208	11010000	01 011 r/m	+	+		8	-	X.....X
RCR	D0	208	11010000	10 011 r/m	++	+		8	-	X.....X
RCR	D0	208	11010000	11 011 r/m	-	+		8	-	X.....X
RCR	D1	209	11010001	00 011 110	++	+		16	-	X.....X
RCR	D1	209	11010001	00 011 r/m	-	+		16	-	X.....X
RCR	D1	209	11010001	01 011 r/m	+	+		16	-	X.....X
RCR	D1	209	11010001	10 011 r/m	++	+		16	-	X.....X
RCR	D1	209	11010001	11 011 r/m	-	+		16	-	X.....X
RCR	D2	210	11010010	00 011 110	++		+	8	-	X.....X
RCR	D2	210	11010010	00 011 r/m	-		+	8	-	X.....X
RCR	D2	210	11010010	01 011 r/m	+		+	8	-	X.....X

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	forgatás 1	forgatás CL	bit	286	flag-ek ODITSZAPC
RCR	D2	210	11010010	10 011 r/m	++		+	8	-	X.....X
RCR	D2	210	11010010	11 011 r/m	-		+	8	-	X.....X
RCR	D3	211	11010011	00 011 110	++		+	16	-	X.....X
RCR	D3	211	11010011	00 011 r/m	-		+	16	-	X.....X
RCR	D3	211	11010011	01 011 r/m	+		+	16	-	X.....X
RCR	D3	211	11010011	10 011 r/m	++		+	16	-	X.....X
RCR	D3	211	11010011	11 011 r/m	-		+	16	-	X.....X
RET	C2	194	11000010	-	++		+	-	-	.....
RET	C3	195	11000011	-	+		-		-	.....
RET	CA	202	11001010	-	++	-	+		-	.....
RET	CB	203	11001011	-	-	-	+		-	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat-irány forrás	adat-irány cél	286 bit		flag-ek ODITSZAPC
REP INSB	F3	243	11110011	01 101 100	0:DX	ES:DI	8	+	.....
REP INSW	F3	243	11110011	01 101 101	0:DX	ES:DI	16	+	.....
REP LODSB	F3	243	11110011	10 101 100	DS:SI	AL	8	-	.....
REP LODSW	F3	243	11110011	10 101 101	DS:SI	AX	16	-	.....
REP MOVSB	F3	243	11110011	10 100 100	DS:SI	ES:DI	8	-	.....
REP MOVSW	F3	243	11110011	10 100 101	DS:SI	ES:DI	16	-	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	forgatás 1	forgatás CL	bit	286	flag-ek ODITSZAPC
ROL	C0	192	11000000	00 000 110	++		+	8	+	.X.....X
ROL	C0	192	11000000	00 000 r/m	-		+	8	+	.X.....X
ROL	C0	192	11000000	01 000 r/m	+		+	8	+	.X.....X
ROL	C0	192	11000000	10 000 r/m	++		+	8	+	.X.....X
ROL	C0	192	11000000	11 000 r/m	-		+	8	+	.X.....X
ROL	C1	193	11000001	00 000 110	++		+	16	+	.X.....X
ROL	C1	193	11000001	00 000 r/m	-		+	16	+	.X.....X
ROL	C1	193	11000001	01 000 r/m	+		+	16	+	.X.....X
ROL	C1	193	11000001	10 000 r/m	++		+	16	+	.X.....X
ROL	C1	193	11000001	11 000 r/m	-		+	16	+	.X.....X
ROL	D0	208	11010000	00 000 110	++	+		8	-	X.....X
ROL	D0	208	11010000	00 000 r/m	-	+		8	-	X.....X
ROL	D0	208	11010000	01 000 r/m	+	+		8	-	X.....X
ROL	D0	208	11010000	10 000 r/m	++	+		8	-	X.....X
ROL	D0	208	11010000	11 000 r/m	-	+		8	-	X.....X
ROL	D1	209	11010001	00 000 110	++	+		16	-	X.....X

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	forgatás 1	forgatás CL	bit	286	flag-ek ODITSZAPC
ROL	D1	209	11010001	00 000 r/m	-	+		16	-	X.....X
ROL	D1	209	11010001	01 000 r/m	+	+		16	-	X.....X
ROL	D1	209	11010001	10 000 r/m	++	+		16	-	X.....X
ROL	D1	209	11010001	11 000 r/m	-	+		16	-	X.....X
ROL	D2	210	11010010	00 000 110	++		+	8	-	X.....X
ROL	D2	210	11010010	00 000 r/m	-		+	8	-	X.....X
ROL	D2	210	11010010	01 000 r/m	+		+	8	-	X.....X
ROL	D2	210	11010010	10 000 r/m	++		+	8	-	X.....X
ROL	D2	210	11010010	11 000 r/m	-		+	8	-	X.....X
ROL	D3	211	11010011	00 000 110	++		+	16	-	X.....X
ROL	D3	211	11010011	00 000 r/m	-		+	16	-	X.....X
ROL	D3	211	11010011	01 000 r/m	+		+	16	-	X.....X
ROL	D3	211	11010011	10 000 r/m	++		+	16	-	X.....X
ROL	D3	211	11010011	11 000 r/m	-		+	16	-	X.....X
ROR	C0	192	11000000	00 001 110	++		+	8	+	.X.....X
ROR	C0	192	11000000	00 001 r/m	-		+	8	+	.X.....X
ROR	C0	192	11000000	01 001 r/m	+		+	8	+	.X.....X
ROR	C0	192	11000000	10 001 r/m	++		+	8	+	.X.....X
ROR	C0	192	11000000	11 001 r/m	-		+	8	+	.X.....X
ROR	C1	193	11000001	00 001 110	++		+	16	+	.X.....X
ROR	C1	193	11000001	00 001 r/m	-		+	16	+	.X.....X
ROR	C1	193	11000001	01 001 r/m	+		+	16	+	.X.....X
ROR	C1	193	11000001	10 001 r/m	++		+	16	+	.X.....X
ROR	C1	193	11000001	11 001 r/m	-		+	16	+	.X.....X
ROR	D0	208	11010000	00 001 110	++	+		8	-	X.....X
ROR	D0	208	11010000	00 001 r/m	-	+		8	-	X.....X
ROR	D0	208	11010000	01 001 r/m	+	+		8	-	X.....X
ROR	D0	208	11010000	10 001 r/m	++	+		8	-	X.....X
ROR	D0	208	11010000	11 001 r/m	-	+		8	-	X.....X
ROR	D1	209	11010001	00 001 110	++	+		16	-	X.....X
ROR	D1	209	11010001	00 001 r/m	-	+		16	-	X.....X
ROR	D1	209	11010001	01 001 r/m	+	+		16	-	X.....X
ROR	D1	209	11010001	10 001 r/m	++	+		16	-	X.....X
ROR	D1	209	11010001	11 001 r/m	-	+		16	-	X.....X
ROR	D2	210	11010010	00 001 110	++		+	8	-	X.....X
ROR	D2	210	11010010	00 001 r/m	-		+	8	-	X.....X
ROR	D2	210	11010010	01 001 r/m	+		+	8	-	X.....X
ROR	D2	210	11010010	10 001 r/m	++		+	8	-	X.....X
ROR	D2	210	11010010	11 001 r/m	-		+	8	-	X.....X
ROR	D3	211	11010011	00 001 110	++		+	16	-	X.....X
ROR	D3	211	11010011	00 001 r/m	-		+	16	-	X.....X
ROR	D3	211	11010011	01 001 r/m	+		+	16	-	X.....X
ROR	D3	211	11010011	10 001 r/m	++		+	16	-	X.....X
ROR	D3	211	11010011	11 001 r/m	-		+	16	-	X.....X

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat-irány forrás	adat-irány cél	bit	286	flag-ek ODITSZAPC
REP OUTSB	F3	243	11110011	01 011 110	DS:SI	0:DX	8	+	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat-irány forrás	adat-irány cél	bit	286	flag-ek ODITSZAPC
REP									
OUTSW	F3	243	11110011	01 011 111	DS:SI	0:DX	16	+	.....
REPNZ									
SCASB	F2	242	11110010	10 101 110	DS:SI	ES:DI	8	-	.....
REPZ									
SCASB	F3	243	11110011	10 101 110	DS:SI	ES:DI	8	-	.....
REPNZ									
SCASW	F2	242	11110010	10 101 111	DS:SI	ES:DI	16	-	.....
REPZ									
SCASW	F3	243	11110011	10 101 111	DS:SI	ES:DI	16	-	.....
REP									
STOSB	F3	243	11110011	10 101 010	AL	ES:DI	8	-	.....
REP									
STOSW	F3	243	11110011	10 101 011	AX	ES:DI	16	-	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
SAHF	9E	158	10001110	-	-	AH -> flag	8	-	....RRRRR

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	forgatás 1	forgatás CL	bit	286	flag-ek ODITSZAPC
SAR	C0	192	11000000	00 111 110	++		+	8	+	....XXUXX
SAR	C0	192	11000000	00 111 r/m	-		+	8	+	....XXUXX
SAR	C0	192	11000000	01 111 r/m	+		+	8	+	....XXUXX
SAR	C0	192	11000000	10 111 r/m	++		+	8	+	....XXUXX
SAR	C0	192	11000000	11 111 r/m	-		+	8	+	....XXUXX
SAR	C1	193	11000001	00 111 110	++		+	16	+	....XXUXX
SAR	C1	193	11000001	00 111 r/m	-		+	16	+	....XXUXX
SAR	C1	193	11000001	01 111 r/m	+		+	16	+	....XXUXX
SAR	C1	193	11000001	10 111 r/m	++		+	16	+	....XXUXX
SAR	C1	193	11000001	11 111 r/m	-		+	16	+	....XXUXX
SAR	D0	208	11010000	00 111 110	++	+		8	-	X...XXUXX
SAR	D0	208	11010000	00 111 r/m	-	+		8	-	X...XXUXX
SAR	D0	208	11010000	01 111 r/m	+	+		8	-	X...XXUXX
SAR	D0	208	11010000	10 111 r/m	++	+		8	-	X...XXUXX
SAR	D0	208	11010000	11 111 r/m	-	+		8	-	X...XXUXX
SAR	D1	209	11010001	00 111 110	++	+		16	-	X...XXUXX
SAR	D1	209	11010001	00 111 r/m	-	+		16	-	X...XXUXX
SAR	D1	209	11010001	01 111 r/m	+	+		16	-	X...XXUXX
SAR	D1	209	11010001	10 111 r/m	++	+		16	-	X...XXUXX
SAR	D1	209	11010001	11 111 r/m	-	+		16	-	X...XXUXX
SAR	D2	210	11010010	00 111 110	++		+	8	-	X...XXUXX
SAR	D2	210	11010010	00 111 r/m	-		+	8	-	X...XXUXX
SAR	D2	210	11010010	01 111 r/m	+		+	8	-	X...XXUXX

mnemo	opkód1			opkód2	adat	forgatás		bit	286	flag-ek
	hex	dec	bináris		byte	1	CL			ODITSZAPC
SAR	D2	210	11010010	10 111 r/m	++		+	8	-	X...XXUXX
SAR	D2	210	11010010	11 111 r/m	-		+	8	-	X...XXUXX
SAR	D3	211	11010011	00 111 110	++		+	16	-	X...XXUXX
SAR	D3	211	11010011	00 111 r/m	-		+	16	-	X...XXUXX
SAR	D3	211	11010011	01 111 r/m	+		+	16	-	X...XXUXX
SAR	D3	211	11010011	10 111 r/m	++		+	16	-	X...XXUXX
SAR	D3	211	11010011	11 111 r/m	-		+	16	-	X...XXUXX

mnemo	opkód1			opkód2	adat	adat-irány		bit	286	flag-ek
	hex	dec	bináris		byte					ODITSZAPC
SBB	18	24	00011000	00 reg 110	++	r/m	-> r	8	-	X...XXXXX
SBB	18	24	00011000	00 reg r/m	-	r/m	-> r	8	-	X...XXXXX
SBB	18	24	00011000	01 reg r/m	+	r/m	-> r	8	-	X...XXXXX
SBB	18	24	00011000	10 reg r/m	++	r/m	-> r	8	-	X...XXXXX
SBB	18	24	00011000	11 reg r/m	-	r/m	-> r	8	-	X...XXXXX
SBB	19	23	00011001	00 reg 110	++	r/m	-> r	16	-	X...XXXXX
SBB	19	23	00011001	00 reg r/m	-	r/m	-> r	16	-	X...XXXXX
SBB	19	23	00011001	01 reg r/m	+	r/m	-> r	16	-	X...XXXXX
SBB	19	23	00011001	10 reg r/m	++	r/m	-> r	16	-	X...XXXXX
SBB	19	23	00011001	11 reg r/m	-	r/m	-> r	16	-	X...XXXXX
SBB	1A	26	00011010	00 reg 110	++	r	-> r\m	8	-	X...XXXXX
SBB	1A	26	00011010	00 reg r/m	-	r	-> r\m	8	-	X...XXXXX
SBB	1A	26	00011010	01 reg r/m	+	r	-> r\m	8	-	X...XXXXX
SBB	1A	26	00011010	10 reg r/m	++	r	-> r\m	8	-	X...XXXXX
SBB	1A	26	00011010	11 reg r/m	-	r	-> r\m	8	-	X...XXXXX
SBB	1B	27	00011011	00 reg 110	++	r	-> r\m	16	-	X...XXXXX
SBB	1B	27	00011011	00 reg r/m	-	r	-> r\m	16	-	X...XXXXX
SBB	1B	27	00011011	01 reg r/m	+	r	-> r\m	16	-	X...XXXXX
SBB	1B	27	00011011	10 reg r/m	++	r	-> r\m	16	-	X...XXXXX
SBB	1B	27	00011011	11 reg r/m	-	r	-> r\m	16	-	X...XXXXX
SBB	1C	28	00011100	-	+	i	-> AL	8	-	X...XXXXX
SBB	1D	29	00011101	-	++	i	-> AX	16	-	X...XXXXX
SBB	80	128	10000000	00 011 110	+++	i	-> r\m	8	-	X...XXXXX
SBB	80	128	10000000	00 011 r/m	+	i	-> r\m	8	-	X...XXXXX
SBB	80	128	10000000	01 011 r/m	++	i	-> r\m	8	-	X...XXXXX
SBB	80	128	10000000	10 011 r/m	+++	i	-> r\m	8	-	X...XXXXX
SBB	80	128	10000000	11 011 r/m	+	i	-> r\m	8	-	X...XXXXX
SBB	81	129	10000001	00 011 110	++++	i	-> r\m	16	-	X...XXXXX
SBB	81	129	10000001	00 011 r/m	++	i	-> r\m	16	-	X...XXXXX
SBB	81	129	10000001	01 011 r/m	+++	i	-> r\m	16	-	X...XXXXX
SBB	81	129	10000001	10 011 r/m	++++	i	-> r\m	16	-	X...XXXXX
SBB	81	129	10000001	11 011 r/m	++	i	-> r\m	16	-	X...XXXXX
SBB	82	130	10000010	00 011 110	+++	i	-> r\m	8	-	X...XXXXX
SBB	82	130	10000010	00 011 r/m	+	i	-> r\m	8	-	X...XXXXX
SBB	82	130	10000010	01 011 r/m	++	i	-> r\m	8	-	X...XXXXX
SBB	82	130	10000010	10 011 r/m	+++	i	-> r\m	8	-	X...XXXXX
SBB	82	130	10000010	11 011 r/m	+	i	-> r\m	8	-	X...XXXXX

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
SBB	83	131	10000011	00 011 110	+++	i -> r\m	16	-	X...XXXXX
SBB	83	131	10000011	00 011 r/m	+	i -> r\m	16	-	X...XXXXX
SBB	83	131	10000011	01 011 r/m	++	i -> r\m	16	-	X...XXXXX
SBB	83	131	10000011	10 011 r/m	+++	i -> r\m	16	-	X...XXXXX
SBB	83	131	10000011	11 011 r/m	+	i -> r\m	16	-	X...XXXXX

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat-irány forrás	adat-irány cél	bit	286	flag-ek ODITSZAPC
SCASB	AE	174	10101110	-	DS:SI	ES:DI	8	-	X...XXXXX
SCASW	AF	175	10101111	-	DS:SI	ES:DI	16	-	X...XXXXX

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	opkód3	adat byte	bit	286	flag-ek ODITSZAPC
SGDT	0F	15	00001111	00000001	00 000 110	++	48	+	.....
SGDT	0F	15	00001111	00000001	00 000 r/m	-	48	+	.....
SGDT	0F	15	00001111	00000001	01 000 r/m	+	48	+	.....
SGDT	0F	15	00001111	00000001	10 000 r/m	++	48	+	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	forgatás 1	forgatás CL	bit	286	flag-ek ODITSZAPC
SHL	C0	192	11000000	00 100 110	++		+	8	+	....XX.XX
SHL	C0	192	11000000	00 100 r/m	-		+	8	+	....XX.XX
SHL	C0	192	11000000	01 100 r/m	+		+	8	+	....XX.XX
SHL	C0	192	11000000	10 100 r/m	++		+	8	+	....XX.XX
SHL	C0	192	11000000	11 100 r/m	-		+	8	+	....XX.XX
SHL	C1	193	11000001	00 100 110	++		+	16	+	....XX.XX
SHL	C1	193	11000001	00 100 r/m	-		+	16	+	....XX.XX
SHL	C1	193	11000001	01 100 r/m	+		+	16	+	....XX.XX
SHL	C1	193	11000001	10 100 r/m	++		+	16	+	....XX.XX
SHL	C1	193	11000001	11 100 r/m	-		+	16	+	....XX.XX
SHL	D0	208	11010000	00 100 110	++	+		8	-	X.......X
SHL	D0	208	11010000	00 100 r/m	-	+		8	-	X.......X
SHL	D0	208	11010000	01 100 r/m	+	+		8	-	X.......X
SHL	D0	208	11010000	10 100 r/m	++	+		8	-	X.......X
SHL	D0	208	11010000	11 100 r/m	-	+		8	-	X.......X
SHL	D1	209	11010001	00 100 110	++	+		16	-	X.......X
SHL	D1	209	11010001	00 100 r/m	-	+		16	-	X.......X
SHL	D1	209	11010001	01 100 r/m	+	+		16	-	X.......X
SHL	D1	209	11010001	10 100 r/m	++	+		16	-	X.......X
SHL	D1	209	11010001	11 100 r/m	-	+		16	-	X.......X
SHL	D2	210	11010010	00 100 110	++		+	8	-	X.......X
SHL	D2	210	11010010	00 100 r/m	-		+	8	-	X.......X
SHL	D2	210	11010010	01 100 r/m	+		+	8	-	X.......X
SHL	D2	210	11010010	10 100 r/m	++		+	8	-	X.......X
SHL	D2	210	11010010	11 100 r/m	-		+	8	-	X.......X
SHL	D3	211	11010011	00 100 110	++		+	16	-	X.......X

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	forgatás 1	forgatás CL	bit	286	flag-ek ODITSZAPC
SHL	D3	211	11010011	00 100 r/m	-		+	16	-	X.....X
SHL	D3	211	11010011	01 100 r/m	+		+	16	-	X.....X
SHL	D3	211	11010011	10 100 r/m	++		+	16	-	X.....X
SHL	D3	211	11010011	11 100 r/m	-		+	16	-	X.....X
SHR	C0	192	11000000	00 101 110	++		+	8	+	....XX.XX
SHR	C0	192	11000000	00 101 r/m	-		+	8	+	....XX.XX
SHR	C0	192	11000000	01 101 r/m	+		+	8	+	....XX.XX
SHR	C0	192	11000000	10 101 r/m	++		+	8	+	....XX.XX
SHR	C0	192	11000000	11 101 r/m	-		+	8	+	....XX.XX
SHR	C1	193	11000001	00 101 110	++		+	16	+	....XX.XX
SHR	C1	193	11000001	00 101 r/m	-		+	16	+	....XX.XX
SHR	C1	193	11000001	01 101 r/m	+		+	16	+	....XX.XX
SHR	C1	193	11000001	10 101 r/m	++		+	16	+	....XX.XX
SHR	C1	193	11000001	11 101 r/m	-		+	16	+	....XX.XX
SHR	D0	208	11010000	00 101 110	++	+		8	-	X.....X
SHR	D0	208	11010000	00 101 r/m	-	+		8	-	X.....X
SHR	D0	208	11010000	01 101 r/m	+	+		8	-	X.....X
SHR	D0	208	11010000	10 101 r/m	++	+		8	-	X.....X
SHR	D0	208	11010000	11 101 r/m	-	+		8	-	X.....X
SHR	D1	209	11010001	00 101 110	++	+		16	-	X.....X
SHR	D1	209	11010001	00 101 r/m	-	+		16	-	X.....X
SHR	D1	209	11010001	01 101 r/m	+	+		16	-	X.....X
SHR	D1	209	11010001	10 101 r/m	++	+		16	-	X.....X
SHR	D1	209	11010001	11 101 r/m	-	+		16	-	X.....X
SHR	D2	210	11010010	00 101 110	++		+	8	-	X.....X
SHR	D2	210	11010010	00 101 r/m	-		+	8	-	X.....X
SHR	D2	210	11010010	01 101 r/m	+		+	8	-	X.....X
SHR	D2	210	11010010	10 101 r/m	++		+	8	-	X.....X
SHR	D2	210	11010010	11 101 r/m	-		+	8	-	X.....X
SHR	D3	211	11010011	00 101 110	++		+	16	-	X.....X
SHR	D3	211	11010011	00 101 r/m	-		+	16	-	X.....X
SHR	D3	211	11010011	01 101 r/m	+		+	16	-	X.....X
SHR	D3	211	11010011	10 101 r/m	++		+	16	-	X.....X
SHR	D3	211	11010011	11 101 r/m	-		+	16	-	X.....X

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	opkód3	adat byte	bit	286	flag-ek ODITSZAPC
SIDT	0F	15	00001111	00000001	00 001 110	++	48	+	.....
SIDT	0F	15	00001111	00000001	00 001 r/m	-	48	+	.....
SIDT	0F	15	00001111	00000001	01 001 r/m	+	48	+	.....
SIDT	0F	15	00001111	00000001	10 001 r/m	++	48	+	.....
SLDT	0F	15	00001111	00000000	00 000 110	++	16	+	.....
SLDT	0F	15	00001111	00000000	00 000 r/m	-	16	+	.....
SLDT	0F	15	00001111	00000000	01 000 r/m	+	16	+	.....
SLDT	0F	15	00001111	00000000	10 000 r/m	++	16	+	.....
SLDT	0F	15	00001111	00000000	11 000 r/m	-	16	+	.....
SMSW	0F	15	00001111	00000001	00 100 110	++	16	+	.....
SMSW	0F	15	00001111	00000001	00 100 r/m	-	16	+	.....
SMSW	0F	15	00001111	00000001	01 100 r/m	+	16	+	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	opkód3	adat byte	bit	286	flag-ek ODITSZAPC
SMSW	0F	15	00001111	00000001	10 100 r/m	++	16	+	.....
SMSW	0F	15	00001111	00000001	11 100 r/m	-	16	+	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
STC	F9	249	11111001	-	-	-	-	-	1
STD	FD	253	11111101	-	-	-	-	-	.1.....
STI	FB	251	11111011	-	-	-	-	-	..1.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat-irány forrás	adat-irány cél	bit	286	flag-ek ODITSZAPC
STOSB	AA	170	10101010	-	AL	ES:DI	8	-	.....
STOSW	AB	171	10101011	-	AX	ES:DI	16	-	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	opkód3	adat byte	bit	286	flag-ek ODITSZAPC
STR	0F	15	00001111	00000000	00 001 110	++	16	+	.....
STR	0F	15	00001111	00000000	00 001 r/m	-	16	+	.....
STR	0F	15	00001111	00000000	01 001 r/m	+	16	+	.....
STR	0F	15	00001111	00000000	10 001 r/m	++	16	+	.....
STR	0F	15	00001111	00000000	11 001 r/m	-	16	+	.....

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
SUB	28	40	00101000	00 reg 110	++	r/m -> r	8	-	X...XXXXX
SUB	28	40	00101000	00 reg r/m	-	r/m -> r	8	-	X...XXXXX
SUB	28	40	00101000	01 reg r/m	+	r/m -> r	8	-	X...XXXXX
SUB	28	40	00101000	10 reg r/m	++	r/m -> r	8	-	X...XXXXX
SUB	28	40	00101000	11 reg r/m	-	r/m -> r	8	-	X...XXXXX
SUB	29	41	00101001	00 reg 110	++	r/m -> r	16	-	X...XXXXX
SUB	29	41	00101001	00 reg r/m	-	r/m -> r	16	-	X...XXXXX
SUB	29	41	00101001	01 reg r/m	+	r/m -> r	16	-	X...XXXXX
SUB	29	41	00101001	10 reg r/m	++	r/m -> r	16	-	X...XXXXX
SUB	29	41	00101001	11 reg r/m	-	r/m -> r	16	-	X...XXXXX
SUB	2A	42	00101010	00 reg 110	++	r -> r\m	8	-	X...XXXXX
SUB	2A	42	00101010	00 reg r/m	-	r -> r\m	8	-	X...XXXXX
SUB	2A	42	00101010	01 reg r/m	+	r -> r\m	8	-	X...XXXXX
SUB	2A	42	00101010	10 reg r/m	++	r -> r\m	8	-	X...XXXXX
SUB	2A	42	00101010	11 reg r/m	-	r -> r\m	8	-	X...XXXXX
SUB	2B	43	00101011	00 reg 110	++	r -> r\m	16	-	X...XXXXX
SUB	2B	43	00101011	00 reg r/m	-	r -> r\m	16	-	X...XXXXX
SUB	2B	43	00101011	01 reg r/m	+	r -> r\m	16	-	X...XXXXX
SUB	2B	43	00101011	10 reg r/m	++	r -> r\m	16	-	X...XXXXX

mnemo	opkód1			opkód2	adat	adat-irány		bit	286	flag-ek
	hex	dec	bináris		byte					ODITSZAPC
SUB	2B	43	00101011	11 reg r/m	-	r	-> r\m	16	-	X...XXXXX
SUB	2C	44	00101100	-	+	i	-> AL	8	-	X...XXXXX
SUB	2D	45	00101101	-	++	i	-> AX	16	-	X...XXXXX
SUB	80	128	10000000	00 101 110	+++	i	-> r\m	8	-	X...XXXXX
SUB	80	128	10000000	00 101 r/m	+	i	-> r\m	8	-	X...XXXXX
SUB	80	128	10000000	01 101 r/m	++	i	-> r\m	8	-	X...XXXXX
SUB	80	128	10000000	10 101 r/m	+++	i	-> r\m	8	-	X...XXXXX
SUB	80	128	10000000	11 101 r/m	+	i	-> r\m	8	-	X...XXXXX
SUB	81	129	10000001	00 101 110	++++	i	-> r\m	16	-	X...XXXXX
SUB	81	129	10000001	00 101 r/m	++	i	-> r\m	16	-	X...XXXXX
SUB	81	129	10000001	01 101 r/m	+++	i	-> r\m	16	-	X...XXXXX
SUB	81	129	10000001	10 101 r/m	++++	i	-> r\m	16	-	X...XXXXX
SUB	81	129	10000001	11 101 r/m	++	i	-> r\m	16	-	X...XXXXX
SUB	82	130	10000010	00 101 110	+++	i	-> r\m	8	-	X...XXXXX
SUB	82	130	10000010	00 101 r/m	+	i	-> r\m	8	-	X...XXXXX
SUB	82	130	10000010	01 101 r/m	++	i	-> r\m	8	-	X...XXXXX
SUB	82	130	10000010	10 101 r/m	+++	i	-> r\m	8	-	X...XXXXX
SUB	82	130	10000010	11 101 r/m	+	i	-> r\m	8	-	X...XXXXX
SUB	83	131	10000011	00 101 110	+++	i	-> r\m	16	-	X...XXXXX
SUB	83	131	10000011	00 101 r/m	+	i	-> r\m	16	-	X...XXXXX
SUB	83	131	10000011	01 101 r/m	++	i	-> r\m	16	-	X...XXXXX
SUB	83	131	10000011	10 101 r/m	+++	i	-> r\m	16	-	X...XXXXX
SUB	83	131	10000011	11 101 r/m	+	i	-> r\m	16	-	X...XXXXX
TEST	0C	12	00001100	-	+	i	-> AL	8	-	0...XXUX0
TEST	0D	13	00001101	-	++	i	-> AX	16	-	0...XXUX0
TEST	84	132	10000100	00 reg 110	++	r/m	-> r	8	-	0...XXUX0
TEST	84	132	10000100	00 reg r/m	-	r/m	-> r	8	-	0...XXUX0
TEST	84	132	10000100	01 reg r/m	+	r/m	-> r	8	-	0...XXUX0
TEST	84	132	10000100	10 reg r/m	++	r/m	-> r	8	-	0...XXUX0
TEST	84	132	10000100	11 reg r/m	-	r/m	-> r	8	-	0...XXUX0
TEST	85	133	10000101	00 reg 110	++	r/m	-> r	16	-	0...XXUX0
TEST	85	133	10000101	00 reg r/m	-	r/m	-> r	16	-	0...XXUX0
TEST	85	133	10000101	01 reg r/m	+	r/m	-> r	16	-	0...XXUX0
TEST	85	133	10000101	10 reg r/m	++	r/m	-> r	16	-	0...XXUX0
TEST	85	133	10000101	11 reg r/m	-	r/m	-> r	16	-	0...XXUX0
TEST	F6	246	11110110	00 000 110	+++	i	-> r\m	8	-	0...XXUX0
TEST	F6	246	11110110	00 000 r/m	+	i	-> r\m	8	-	0...XXUX0
TEST	F6	246	11110110	01 000 r/m	++	i	-> r\m	8	-	0...XXUX0
TEST	F6	246	11110110	10 000 r/m	+++	i	-> r\m	8	-	0...XXUX0
TEST	F6	246	11110110	11 000 r/m	+	i	-> r\m	8	-	0...XXUX0
TEST	F7	247	11110111	00 000 110	++++	i	-> r\m	16	-	0...XXUX0
TEST	F7	247	11110111	00 000 r/m	++	i	-> r\m	16	-	0...XXUX0
TEST	F7	247	11110111	01 000 r/m	+++	i	-> r\m	16	-	0...XXUX0
TEST	F7	247	11110111	10 000 r/m	++++	i	-> r\m	16	-	0...XXUX0
TEST	F7	247	11110111	11 000 r/m	++	i	-> r\m	16	-	0...XXUX0

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	opkód3	adat byte	bit	286	flag-ek ODITSZAPC
VERR	0F	15	00001111	00000000	00 100 110	++	16	+	.....X...
VERR	0F	15	00001111	00000000	00 100 r/m	-	16	+	.....X...
VERR	0F	15	00001111	00000000	01 100 r/m	+	16	+	.....X...
VERR	0F	15	00001111	00000000	10 100 r/m	++	16	+	.....X...
VERR	0F	15	00001111	00000000	11 100 r/m	-	16	+	.....X...
VERW	0F	15	00001111	00000000	00 101 110	++	16	+	.....X...
VERW	0F	15	00001111	00000000	00 101 r/m	-	16	+	.....X...
VERW	0F	15	00001111	00000000	01 101 r/m	+	16	+	.....X...
VERW	0F	15	00001111	00000000	10 101 r/m	++	16	+	.....X...
VERW	0F	15	00001111	00000000	11 101 r/m	-	16	+	.....X...

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
WAIT	9B	155	10011101	-	-	-	-	-	.....
XCHG	86	134	10000110	00 000 110	++	r/m <-> r	8	-	.....
XCHG	86	134	10000110	mod 000 r/ m	-	r/m <-> r	8	-	.....
XCHG	87	135	10000111	00 000 110	++	r/m <-> r	16	-	.....
XCHG	87	135	10000111	mod 000 r/ m	-	r/m <-> r	16	-	.....
XCHG	90+x		10010reg	-	-	r <-> AX	16	-	.....
XLAT	D7	215	11010111	-	-	m -> AL	8	-	0...XXUX0
XOR	24	36	00100100	-	+	i -> AL	8	-	0...XXUX0
XOR	25	37	00100101	-	++	i -> AX	16	-	0...XXUX0
XOR	30	48	00110000	00 reg 110	++	r/m -> r	8	-	0...XXUX0
XOR	30	48	00110000	00 reg r/m	-	r/m -> r	8	-	0...XXUX0
XOR	30	48	00110000	01 reg r/m	+	r/m -> r	8	-	0...XXUX0
XOR	30	48	00110000	10 reg r/m	++	r/m -> r	8	-	0...XXUX0
XOR	30	48	00110000	11 reg r/m	-	r/m -> r	8	-	0...XXUX0
XOR	31	49	00110001	00 reg 110	++	r/m -> r	16	-	0...XXUX0
XOR	31	49	00110001	00 reg r/m	-	r/m -> r	16	-	0...XXUX0
XOR	31	49	00110001	01 reg r/m	+	r/m -> r	16	-	0...XXUX0
XOR	31	49	00110001	10 reg r/m	++	r/m -> r	16	-	0...XXUX0
XOR	31	49	00110001	11 reg r/m	-	r/m -> r	16	-	0...XXUX0
XOR	32	50	00110010	00 reg 110	++	r -> r\m	8	-	0...XXUX0
XOR	32	50	00110010	00 reg r/m	-	r -> r\m	8	-	0...XXUX0
XOR	32	50	00110010	01 reg r/m	+	r -> r\m	8	-	0...XXUX0
XOR	32	50	00110010	10 reg r/m	++	r -> r\m	8	-	0...XXUX0
XOR	32	50	00110010	11 reg r/m	-	r -> r\m	8	-	0...XXUX0
XOR	33	51	00110011	00 reg 110	++	r -> r\m	16	-	0...XXUX0
XOR	33	51	00110011	00 reg r/m	-	r -> r\m	16	-	0...XXUX0
XOR	33	51	00110011	01 reg r/m	+	r -> r\m	16	-	0...XXUX0
XOR	33	51	00110011	10 reg r/m	++	r -> r\m	16	-	0...XXUX0
XOR	33	51	00110011	11 reg r/m	-	r -> r\m	16	-	0...XXUX0
XOR	80	128	10000000	00 110 110	+++	i -> r\m	8	-	0...XXUX0
XOR	80	128	10000000	00 110 r/m	+	i -> r\m	8	-	0...XXUX0
XOR	80	128	10000000	01 110 r/m	++	i -> r\m	8	-	0...XXUX0

mnemo	opkód1 hex	opkód1 dec	opkód1 bináris	opkód2	adat byte	adat-irány	bit	286	flag-ek ODITSZAPC
XOR	80	128	10000000	10 110 r/m	+++	i -> r\m	8	-	0...XXUX0
XOR	80	128	10000000	11 110 r/m	+	i -> r\m	8	-	0...XXUX0
XOR	81	129	10000001	00 110 110	++++	i -> r\m	16	-	0...XXUX0
XOR	81	129	10000001	00 110 r/m	++	i -> r\m	16	-	0...XXUX0
XOR	81	129	10000001	01 110 r/m	+++	i -> r\m	16	-	0...XXUX0
XOR	81	129	10000001	10 110 r/m	++++	i -> r\m	16	-	0...XXUX0
XOR	81	129	10000001	11 110 r/m	++	i -> r\m	16	-	0...XXUX0

8087-80287 utasításkészlet ABC sorrendben (A 8086-8088-80286 utasításkészlet kibővítése)

Jelölések, magyarázatok:

A címképzést az opkód *mod* és *r/m* mezője határozza meg a következők szerint:
(*mod* az opkód2 legfelső két bitje)

- *mod* = 11 esetben az *r/m* mező ST(i) mezőként kezelendő
- *mod* = 00 esetén az eltolás (displacement) = 0
- *mod* = 01 esetén az eltolás alsó byte-ja 16 bites előjel-kiterjesztetté válik, a felső byte hiányzik
- *mod* = 10 esetén az eltolás alsó és felső byte-ja is érvényes

- ha *r/m* = 000, akkor EA = (BX) + (SI) + DISP (EA:= Execution Address)
- ha *r/m* = 001, akkor EA = (BX) + (DI) + DISP
- ha *r/m* = 010, akkor EA = (BP) + (SI) + DISP
- ha *r/m* = 011, akkor EA = (BP) + (DI) + DISP
- ha *r/m* = 100, akkor EA = (SI) + DISP
- ha *r/m* = 101, akkor EA = (DI) + DISP
- ha *r/m* = 110, akkor EA = (BP) + DISP *
- ha *r/m* = 111, akkor EA = (BX) + DISP

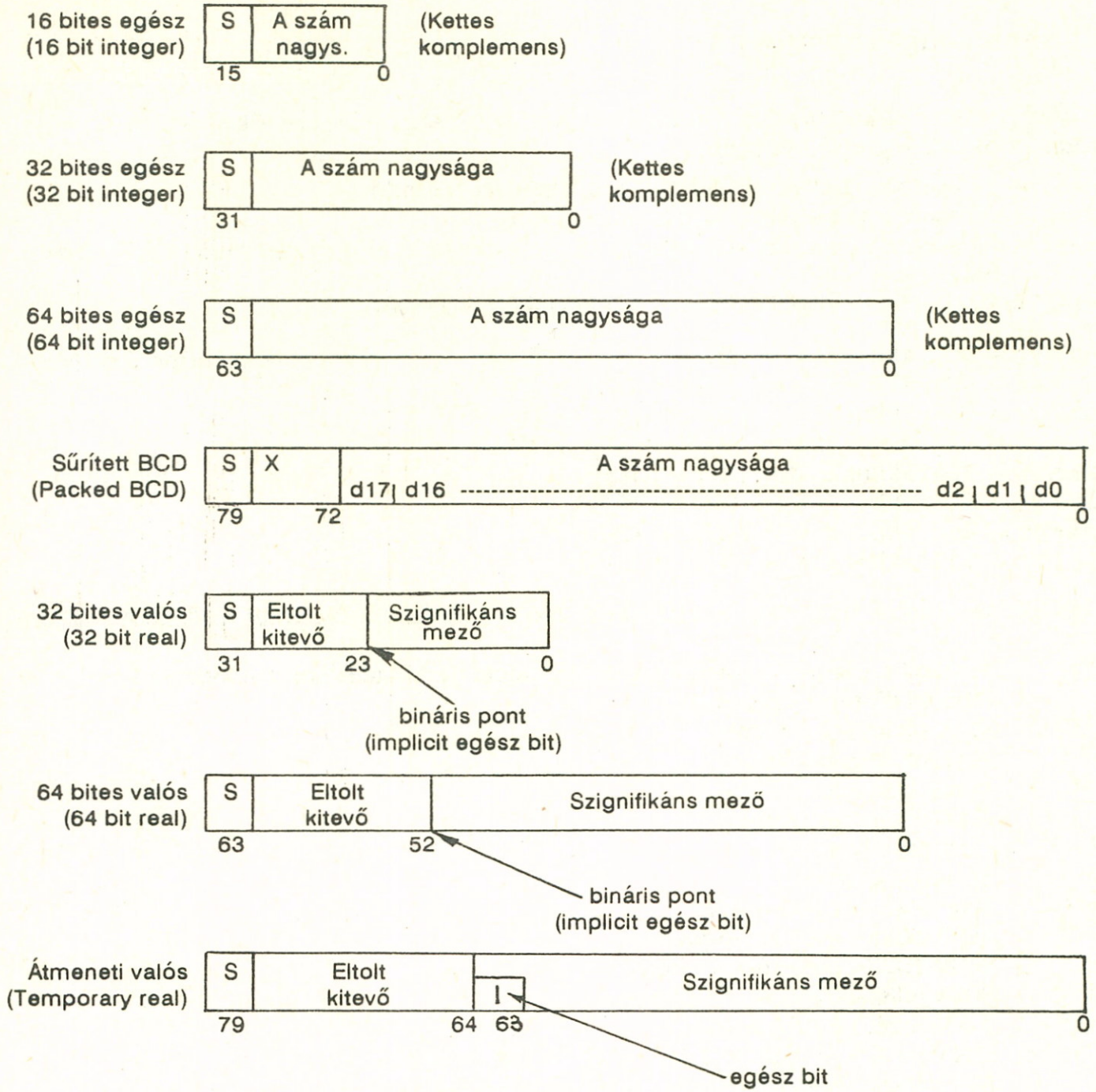
* - kivéve, ha *mod* = 00 és *r/m* = 110, akkor EA = eltolás felső-eltolás alsó(16 bit offszet)

ST(i)	A stack i. eleme (a stack 8 elem mélységű: az aritmetikai processzor 8 belső 80 bites regisztere)
ST(0)	A stack legfelső eleme
ST(7)	A stack legalsó eleme
STP	Stack Pointer
ST	A stack összes eleme
CR	Control regiszter
SR	Status regiszter
TAG	Tag regiszter (toldalék)
IP	8087-80287 instrukció pointer
OP	8087-80287 operandus pointer
AX	8086-8088-80286 AX regisztere
ENV	Környezet (Environment) ENV = CR + SR + TAG + IP + OP + opkód
ic	8087-80287 belső konstans (internal constant)
R	operandusok felcserélése 0 - cél (operátor) forrás 1 - forrás (operátor) cél

16-i: 16 bites egész
32-i: 32 bites egész
64-i: 64 bites egész
32-r: 32 bites valós
64-r: 64 bites valós
80-t: Átmeneti valós
80-d: Sűrített decimális

8087-80287 utasításkészlet ABC sorrendben (A 8086-8088-80286 utasításkészlet kibővítése)

Jelölések, magyarázatok:

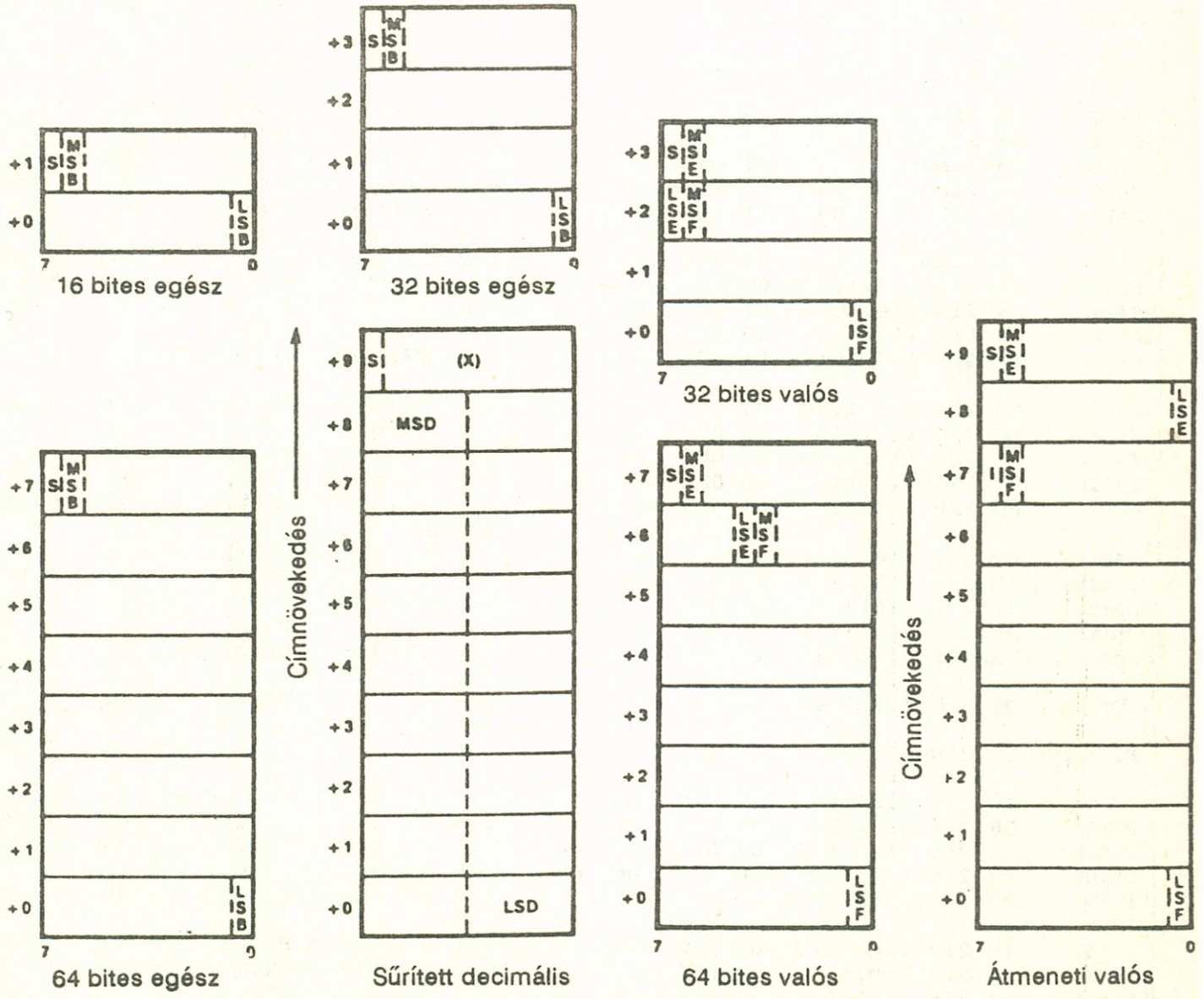


S előjel: 0 - pozitív, 1 - negatív
di decimális digit
X ezeknek a biteknek nincs jelentésük
Eltolások: 32 bites valós - 127 (7Fh)
64 bites valós - 1023 (3FFh)
Átmeneti valós - 16383 (3FFFh)

8087-80287 adattípusok

8087-80287 utasításkészlet ABC sorrendben
(A 8086-8088-80286 utasításkészlet kibővítése)

Jelölések, magyarázatok:



S: előjel bit
MSB/LSB: legnagyobb/legkisebb helyértékű bit
MSD/LSD: legnagyobb/legkisebb helyértékű decimális digit
(x): ezeknek a biteknek nincs jelentésük

S: előjel bit
MSE/LSE: legnagyobb/legkisebb helyértékű exponens bit
MSF/LSF: legnagyobb/legkisebb helyértékű fraction bit
I: integer bit

8087-80287 adattípusok elhelyezkedése a memóriában

8087-80287 utasításkészlet ABC sorrendben
(A 8086-8088-80286 utasításkészlet kibővítése)

mnemo	opkód1 hex	dec	bináris	opkód2	elt. byte	mem. adat	adat-irány forrás --> cél	bit műv.	87	287
F2XM1	D9	217	11011001	11 110 000	-	-	ST(0)	80	+	+
FABS	D9	217	11011001	11 100 001	-	-	ST(0)	80	+	+
FADD	D8	216	11011000	00 000 110	++	32-r	r/m --> ST(0)	80	+	+
FADD	D8	216	11011000	00 000 r/m	-	32-r	r/m --> ST(0)	80	+	+
FADD	D8	216	11011000	01 000 r/m	+	32-r	r/m --> ST(0)	80	+	+
FADD	D8	216	11011000	10 000 r/m	++	32-r	r/m --> ST(0)	80	+	+
FADD	D8	216	11011000	11 000 ST(i)	-	-	ST(i)--> ST(0)	80	+	+
FADD	DA	218	11011010	00 000 110	++	32-i	r/m --> ST(0)	80	+	+
FADD	DA	218	11011010	00 000 r/m	-	32-i	r/m --> ST(0)	80	+	+
FADD	DA	218	11011010	01 000 r/m	+	32-i	r/m --> ST(0)	80	+	+
FADD	DA	218	11011010	10 000 r/m	++	32-i	r/m --> ST(0)	80	+	+
FADD	DA	218	11011010	11 000 ST(i)	-	-	ST(i)--> ST(0),pop	80	+	+
FADD	DC	220	11011100	00 000 110	++	64-r	r/m --> ST(0)	80	+	+
FADD	DC	220	11011100	00 000 r/m	-	64-r	r/m --> ST(0)	80	+	+
FADD	DC	220	11011100	01 000 r/m	+	64-r	r/m --> ST(0)	80	+	+
FADD	DC	220	11011100	10 000 r/m	++	64-r	r/m --> ST(0)	80	+	+
FADD	DC	220	11011100	11 000 ST(i)	-	-	ST(0)--> ST(i)	80	+	+
FADD	DE	222	11011110	00 000 110	++	16-i	r/m --> ST(0)	80	+	+
FADD	DE	222	11011110	00 000 r/m	-	16-i	r/m --> ST(0)	80	+	+
FADD	DE	222	11011110	01 000 r/m	+	16-i	r/m --> ST(0)	80	+	+
FADD	DE	222	11011110	10 000 r/m	++	16-i	r/m --> ST(0)	80	+	+
FADD	DE	222	11011110	11 000 ST(i)	-	-	ST(0)--> ST(i),pop	80	+	+
FCHS	D9	217	11011001	11 100 000	-	-	ST(0)	80	+	+
FCLEX	DB	219	11011011	11 100 010	-	-	-	-	+	+
FCOM	D8	216	11011000	00 010 110	++	32-r	r/m --> ST(0)	80	+	+
FCOM	D8	216	11011000	00 010 r/m	-	32-r	r/m --> ST(0)	80	+	+
FCOM	D8	216	11011000	01 010 r/m	+	32-r	r/m --> ST(0)	80	+	+
FCOM	D8	216	11011000	10 010 r/m	++	32-r	r/m --> ST(0)	80	+	+
FCOM	D8	216	11011000	11 010 ST(i)	-	-	ST(i)--> ST(0)	80	+	+
FCOM	DA	218	11011010	00 010 110	++	32-i	r/m --> ST(0)	80	+	+
FCOM	DA	218	11011010	00 010 r/m	-	32-i	r/m --> ST(0)	80	+	+
FCOM	DA	218	11011010	01 010 r/m	+	32-i	r/m --> ST(0)	80	+	+
FCOM	DA	218	11011010	10 010 r/m	++	32-i	r/m --> ST(0)	80	+	+
FCOM	DC	220	11011100	00 010 110	++	64-r	r/m --> ST(0)	80	+	+
FCOM	DC	220	11011100	00 010 r/m	-	64-r	r/m --> ST(0)	80	+	+
FCOM	DC	220	11011100	01 010 r/m	+	64-r	r/m --> ST(0)	80	+	+
FCOM	DC	220	11011100	10 010 r/m	++	64-r	r/m --> ST(0)	80	+	+
FCOM	DE	222	11011110	00 010 110	++	16-i	r/m --> ST(0)	80	+	+
FCOM	DE	222	11011110	00 010 r/m	-	16-i	r/m --> ST(0)	80	+	+
FCOM	DE	222	11011110	01 010 r/m	+	16-i	r/m --> ST(0)	80	+	+
FCOM	DE	222	11011110	10 010 r/m	++	16-i	r/m --> ST(0)	80	+	+
FCOMP	D8	216	11011000	00 011 110	++	32-r	r/m --> ST(0),pop	80	+	+

mnemo	opkód1 hex	dec	bináris	opkód2	elt. byte	mem. adat	adat-irány forrás --> cél	bit műv.	87	287
FCOMP	D8	216	11011000	00 011 r/m	-	32-r	r/m --> ST(0),pop	80	+	+
FCOMP	D8	216	11011000	01 011 r/m	+	32-r	r/m --> ST(0),pop	80	+	+
FCOMP	D8	216	11011000	10 011 r/m	++	32-r	r/m --> ST(0),pop	80	+	+
FCOMP	D8	216	11011000	11 011 ST(i)	-	-	ST(i)--> ST(0),pop	80	+	+
FCOMP	DA	218	11011010	00 011 110	++	32-i	r/m --> ST(0),pop	80	+	+
FCOMP	DA	218	11011010	00 011 r/m	-	32-i	r/m --> ST(0),pop	80	+	+
FCOMP	DA	218	11011010	01 011 r/m	+	32-i	r/m --> ST(0),pop	80	+	+
FCOMP	DA	218	11011010	10 011 r/m	++	32-i	r/m --> ST(0),pop	80	+	+
FCOMP	DC	220	11011100	00 011 110	++	64-r	r/m --> ST(0),pop	80	+	+
FCOMP	DC	220	11011100	00 011 r/m	-	64-r	r/m --> ST(0),pop	80	+	+
FCOMP	DC	220	11011100	01 011 r/m	+	64-r	r/m --> ST(0),pop	80	+	+
FCOMP	DC	220	11011100	10 011 r/m	++	64-r	r/m --> ST(0),pop	80	+	+
FCOMP	DE	222	11011110	00 011 110	++	16-i	r/m --> ST(0),pop	80	+	+
FCOMP	DE	222	11011110	00 011 r/m	-	16-i	r/m --> ST(0),pop	80	+	+
FCOMP	DE	222	11011110	01 011 r/m	+	16-i	r/m --> ST(0),pop	80	+	+
FCOMP	DE	222	11011110	10 011 r/m	++	16-i	r/m --> ST(0),pop	80	+	+
FCOMPP	DE	222	11011110	11 011 001	-	-	ST(1)--> ST(0),2pop	80	+	+
FDECSTP	D9	217	11011001	11 110 110	-	-	STP	-	+	+
FDISI	DB	219	11011011	11 100 001	-	-	-	-	+	-
FDIV	D8	216	11011000	00 11R 110	++	32-r	r/m --> ST(0)	80	+	+
FDIV	D8	216	11011000	00 11R r/m	-	32-r	r/m --> ST(0)	80	+	+
FDIV	D8	216	11011000	01 11R r/m	+	32-r	r/m --> ST(0)	80	+	+
FDIV	D8	216	11011000	10 11R r/m	++	32-r	r/m --> ST(0)	80	+	+
FDIV	D8	216	11011000	11 11R ST(i)	-	-	ST(i)--> ST(0)	80	+	+
FDIV	DA	218	11011010	00 11R 110	++	32-i	r/m --> ST(0)	80	+	+
FDIV	DA	218	11011010	00 11R r/m	-	32-i	r/m --> ST(0)	80	+	+
FDIV	DA	218	11011010	01 11R r/m	+	32-i	r/m --> ST(0)	80	+	+
FDIV	DA	218	11011010	10 11R r/m	++	32-i	r/m --> ST(0)	80	+	+
FDIV	DA	218	11011010	11 11R ST(i)	-	-	ST(i)--> ST(0),pop	80	+	+
FDIV	DC	220	11011100	00 11R 110	++	64-r	r/m --> ST(0)	80	+	+
FDIV	DC	220	11011100	00 11R r/m	-	64-r	r/m --> ST(0)	80	+	+
FDIV	DC	220	11011100	01 11R r/m	+	64-r	r/m --> ST(0)	80	+	+
FDIV	DC	220	11011100	10 11R r/m	++	64-r	r/m --> ST(0)	80	+	+
FDIV	DC	220	11011100	11 11R ST(i)	-	-	ST(0)--> ST(i)	80	+	+
FDIV	DE	222	11011110	00 11R 110	++	16-i	r/m --> ST(0)	80	+	+
FDIV	DE	222	11011110	00 11R r/m	-	16-i	r/m --> ST(0)	80	+	+
FDIV	DE	222	11011110	01 11R r/m	+	16-i	r/m --> ST(0)	80	+	+
FDIV	DE	222	11011110	10 11R r/m	++	16-i	r/m --> ST(0)	80	+	+
FDIV	DE	222	11011110	11 11R ST(i)	-	-	ST(0)--> ST(i),pop	80	+	+
FENI	DB	219	11011011	11 100 000	-	-	-	-	+	-
FFREE	DD	221	11011101	11 000 ST(i)	-	-	TAG ST(i)	2	+	+
FINCSTP	D9	217	11011001	11 110 111	-	-	STP	-	+	+
FINIT	DB	219	11011011	11 100 011	-	-	-	-	+	+
FLD	D9	217	11011001	00 000 110	++	32-r	r/m --> ST(0)	80	+	+
FLD	D9	217	11011001	00 000 r/m	-	32-r	r/m --> ST(0)	80	+	+

mnemo	opkód1 hex	dec	bináris	opkód2	elt. byte	mem. adat	adat-irány forrás --> cél	bit műv.	87	287
FLD	D9	217	11011001	01 000 r/m	+	32-r	r/m --> ST(0)	80	+	+
FLD	D9	217	11011001	10 000 r/m	++	32-r	r/m --> ST(0)	80	+	+
FLD	D9	217	11011001	11 000 ST(i)	-	-	ST(i)--> ST(0)	80	+	+
FLD	DB	219	11011011	00 000 110	++	32-i	r/m --> ST(0)	80	+	+
FLD	DB	219	11011011	00 000 r/m	-	32-i	r/m --> ST(0)	80	+	+
FLD	DB	219	11011011	01 000 r/m	+	32-i	r/m --> ST(0)	80	+	+
FLD	DB	219	11011011	10 000 r/m	++	32-i	r/m --> ST(0)	80	+	+
FLD	DD	221	11011101	00 000 110	++	64-r	r/m --> ST(0)	80	+	+
FLD	DD	221	11011101	00 000 r/m	-	64-r	r/m --> ST(0)	80	+	+
FLD	DD	221	11011101	01 000 r/m	+	64-r	r/m --> ST(0)	80	+	+
FLD	DD	221	11011101	10 000 r/m	++	64-r	r/m --> ST(0)	80	+	+
FLD	DF	223	11011111	00 000 110	++	16-i	r/m --> ST(0)	80	+	+
FLD	DF	223	11011111	00 000 r/m	-	16-i	r/m --> ST(0)	80	+	+
FLD	DF	223	11011111	01 000 r/m	+	16-i	r/m --> ST(0)	80	+	+
FLD	DF	223	11011111	10 000 r/m	++	16-i	r/m --> ST(0)	80	+	+
FLD	DF	223	11011111	00 101 110	++	64-i	r/m --> ST(0)	80	+	+
FLD	DF	223	11011111	00 101 r/m	-	64-i	r/m --> ST(0)	80	+	+
FLD	DF	223	11011111	01 101 r/m	+	64-i	r/m --> ST(0)	80	+	+
FLD	DF	223	11011111	10 101 r/m	++	64-i	r/m --> ST(0)	80	+	+
FLD	DB	219	11011011	00 101 110	++	80-t	r/m --> ST(0)	80	+	+
FLD	DB	219	11011011	00 101 r/m	-	80-t	r/m --> ST(0)	80	+	+
FLD	DB	219	11011011	01 101 r/m	+	80-t	r/m --> ST(0)	80	+	+
FLD	DB	219	11011011	10 101 r/m	++	80-t	r/m --> ST(0)	80	+	+
FLD	DF	223	11011111	00 100 110	++	80-d	r/m --> ST(0)	80	+	+
FLD	DF	223	11011111	00 100 r/m	-	80-d	r/m --> ST(0)	80	+	+
FLD	DF	223	11011111	01 100 r/m	+	80-d	r/m --> ST(0)	80	+	+
FLD	DF	223	11011111	10 100 r/m	++	80-d	r/m --> ST(0)	80	+	+
FLD1	D9	217	11011001	11 101 000	-	-	ic --> ST(0)	80	+	+
FLDCW	D9	217	11011001	00 101 110	++	16bit	r/m --> CR	16	+	+
FLDCW	D9	217	11011001	00 101 r/m	-	16bit	r/m --> CR	16	+	+
FLDCW	D9	217	11011001	01 101 r/m	+	16bit	r/m --> CR	16	+	+
FLDCW	D9	217	11011001	10 101 r/m	++	16bit	r/m --> CR	16	+	+
FLDENV	D9	217	11011001	00 100 110	++	14byte	r/m --> ENV	7x16	+	+
FLDENV	D9	217	11011001	00 100 r/m	-	14byte	r/m --> ENV	7x16	+	+
FLDENV	D9	217	11011001	01 100 r/m	+	14byte	r/m --> ENV	7x16	+	+
FLDENV	D9	217	11011001	10 100 r/m	++	14byte	r/m --> ENV	7x16	+	+
FLDL2E	D9	217	11011001	11 101 010	-	-	ic --> ST(0)	80	+	+
FLDL2T	D9	217	11011001	11 101 001	-	-	ic --> ST(0)	80	+	+
FLDLG2	D9	217	11011001	11 101 100	-	-	ic --> ST(0)	80	+	+
FLDLN2	D9	217	11011001	11 101 101	-	-	ic --> ST(0)	80	+	+
FLDPI	D9	217	11011001	11 101 011	-	-	ic --> ST(0)	80	+	+
FLDZ	D9	217	11011001	11 101 110	-	-	ic --> ST(0)	80	+	+
FMUL	D8	216	11011000	00 001 110	++	32-r	r/m --> ST(0)	80	+	+
FMUL	D8	216	11011000	00 001 r/m	-	32-r	r/m --> ST(0)	80	+	+

mnemo	opkód1 hex	dec	bináris	opkód2	elt. byte	mem. adat	adat-irány forrás --> cél	bit műv.	87	287
FMUL	D8	216	11011000	01 001 r/m	+	32-r	r/m --> ST(0)	80	+	+
FMUL	D8	216	11011000	10 001 r/m	++	32-r	r/m --> ST(0)	80	+	+
FMUL	D8	216	11011000	11 001 ST(i)	-	-	ST(i)--> ST(0)	80	+	+
FMUL	DA	218	11011010	00 001 110	++	32-i	r/m --> ST(0)	80	+	+
FMUL	DA	218	11011010	00 001 r/m	-	32-i	r/m --> ST(0)	80	+	+
FMUL	DA	218	11011010	01 001 r/m	+	32-i	r/m --> ST(0)	80	+	+
FMUL	DA	218	11011010	10 001 r/m	++	32-i	r/m --> ST(0)	80	+	+
FMUL	DA	218	11011010	11 001 ST(i)	-	-	ST(i)--> ST(0),pop	80	+	+
FMUL	DC	220	11011100	00 001 110	++	64-r	r/m --> ST(0)	80	+	+
FMUL	DC	220	11011100	00 001 r/m	-	64-r	r/m --> ST(0)	80	+	+
FMUL	DC	220	11011100	01 001 r/m	+	64-r	r/m --> ST(0)	80	+	+
FMUL	DC	220	11011100	10 001 r/m	++	64-r	r/m --> ST(0)	80	+	+
FMUL	DC	220	11011100	11 001 ST(i)	-	-	ST(0)--> ST(i)	80	+	+
FMUL	DE	222	11011110	00 001 110	++	16-i	r/m --> ST(0)	80	+	+
FMUL	DE	222	11011110	00 001 r/m	-	16-i	r/m --> ST(0)	80	+	+
FMUL	DE	222	11011110	01 001 r/m	+	16-i	r/m --> ST(0)	80	+	+
FMUL	DE	222	11011110	10 001 r/m	++	16-i	r/m --> ST(0)	80	+	+
FMUL	DE	222	11011110	11 001 ST(i)	-	-	ST(0)--> ST(i),pop	80	+	+
FNOP	D9	217	11011001	11 010 000	-	-	-	-	+	+
FPATAN	D9	217	11011001	11 110 011	-	-	ST(0)-->ST(0),push	160	+	+
FPREM	D9	217	11011001	11 111 000	-	-	ST(1)--> ST(0)	-	+	+
FPTAN	D9	217	11011001	11 110 010	-	-	- ST(0)	-	+	+
FRNDINT	D9	217	11011001	11 111 100	-	-	- ST(0)	-	+	+
FRSTOR	DD	221	11011101	00 100 110	++	94byte	r/m -->ENV+ST	7x16+8x80	+	+
FRSTOR	DD	221	11011101	00 100 r/m	-	94byte	r/m -->ENV+ST	7x16+8x80	+	+
FRSTOR	DD	221	11011101	01 100 r/m	+	94byte	r/m -->ENV+ST	7x16+8x80	+	+
FRSTOR	DD	221	11011101	10 100 r/m	++	94byte	r/m -->ENV+ST	7x16+8x80	+	+
FSAVE	DD	221	11011101	00 110 110	++	94byte	ENV+ST--> r/m	7x16+8x80	+	+
FSAVE	DD	221	11011101	00 110 r/m	-	94byte	ENV+ST--> r/m	7x16+8x80	+	+
FSAVE	DD	221	11011101	01 110 r/m	+	94byte	ENV+ST--> r/m	7x16+8x80	+	+
FSAVE	DD	221	11011101	10 110 r/m	++	94byte	ENV+ST--> r/m	7x16+8x80	+	+
FSCALE	D9	217	11011001	11 111 101	-	-	ST(1)--> ST(0)	80	+	+
FSETPM	DB	219	11011011	11 100 100	-	-	-	-	-	+
FSQRT	D9	217	11011001	11 111 010	-	-	ST(0)	80	+	+
FST	D9	217	11011001	00 010 110	++	32-r	ST(0)--> r/m	32	+	+
FST	D9	217	11011001	00 010 r/m	-	32-r	ST(0)--> r/m	32	+	+
FST	D9	217	11011001	01 010 r/m	+	32-r	ST(0)--> r/m	32	+	+
FST	D9	217	11011001	10 010 r/m	++	32-r	ST(0)--> r/m	32	+	+
FST	DB	219	11011011	00 010 110	++	32-i	ST(0)--> r/m	32	+	+
FST	DB	219	11011011	00 010 r/m	-	32-i	ST(0)--> r/m	32	+	+
FST	DB	219	11011011	01 010 r/m	+	32-i	ST(0)--> r/m	32	+	+
FST	DB	219	11011011	10 010 r/m	++	32-i	ST(0)--> r/m	32	+	+
FST	DD	221	11011101	00 010 110	++	64-r	ST(0)--> r/m	64	+	+
FST	DD	221	11011101	00 010 r/m	-	64-r	ST(0)--> r/m	64	+	+

mnemo	opkód1 hex	dec	bináris	opkód2	elt. byte	mem. adat	adat-irány forrás --> cél	bit műv.	87	287
FST	DD	221	11011101	01 010 r/m	+	64-r	ST(0)--> r/m	64	+	+
FST	DD	221	11011101	10 010 r/m	++	64-r	ST(0)--> r/m	64	+	+
FST	DD	221	11011101	11 010 ST(i)	-	-	ST(0)--> ST(i)	80	+	+
FST	DF	223	11011111	00 010 110	++	16-i	ST(0)--> r/m	16	+	+
FST	DF	223	11011111	00 010 r/m	-	16-i	ST(0)--> r/m	16	+	+
FST	DF	223	11011111	01 010 r/m	+	16-i	ST(0)--> r/m	16	+	+
FST	DF	223	11011111	10 010 r/m	++	16-i	ST(0)--> r/m	16	+	+
FSTCW	D9	217	11011001	00 111 110	++	16bit	CR --> r/m	16	+	+
FSTCW	D9	217	11011001	00 111 r/m	-	16bit	CR --> r/m	16	+	+
FSTCW	D9	217	11011001	01 111 r/m	+	16bit	CR --> r/m	16	+	+
FSTCW	D9	217	11011001	10 111 r/m	++	16bit	CR --> r/m	16	+	+
FSTENV	D9	217	11011001	00 110 110	++	14byte	ENV --> r/m	7x16	+	+
FSTENV	D9	217	11011001	00 110 r/m	-	14byte	ENV --> r/m	7x16	+	+
FSTENV	D9	217	11011001	01 110 r/m	+	14byte	ENV --> r/m	7x16	+	+
FSTENV	D9	217	11011001	10 110 r/m	++	14byte	ENV --> r/m	7x16	+	+
FSTP	D9	217	11011001	00 011 110	++	32-r	ST(0)--> r/m,pop	32	+	+
FSTP	D9	217	11011001	00 011 r/m	-	32-r	ST(0)--> r/m,pop	32	+	+
FSTP	D9	217	11011001	01 011 r/m	+	32-r	ST(0)--> r/m,pop	32	+	+
FSTP	D9	217	11011001	10 011 r/m	++	32-r	ST(0)--> r/m,pop	32	+	+
FSTP	DB	219	11011011	00 011 110	++	32-i	ST(0)--> r/m,pop	32	+	+
FSTP	DB	219	11011011	00 011 r/m	-	32-i	ST(0)--> r/m,pop	32	+	+
FSTP	DB	219	11011011	01 011 r/m	+	32-i	ST(0)--> r/m,pop	32	+	+
FSTP	DB	219	11011011	10 011 r/m	++	32-i	ST(0)--> r/m,pop	32	+	+
FSTP	DD	221	11011101	00 011 110	++	64-r	ST(0)--> r/m,pop	64	+	+
FSTP	DD	221	11011101	00 011 r/m	-	64-r	ST(0)--> r/m,pop	64	+	+
FSTP	DD	221	11011101	01 011 r/m	+	64-r	ST(0)--> r/m,pop	64	+	+
FSTP	DD	221	11011101	10 011 r/m	++	64-r	ST(0)--> r/m,pop	64	+	+
FSTP	DD	221	11011101	11 011 ST(i)	-	-	ST(0)-->ST(i),pop	80	+	+
FSTP	DF	223	11011111	00 011 110	++	16-i	ST(0)--> r/m,pop	16	+	+
FSTP	DF	223	11011111	00 011 r/m	-	16-i	ST(0)--> r/m,pop	16	+	+
FSTP	DF	223	11011111	01 011 r/m	+	16-i	ST(0)--> r/m,pop	16	+	+
FSTP	DF	223	11011111	10 011 r/m	++	16-i	ST(0)--> r/m,pop	16	+	+
FSTP	DF	223	11011111	00 111 110	++	64-i	ST(0)--> r/m,pop	64	+	+
FSTP	DF	223	11011111	00 111 r/m	-	64-i	ST(0)--> r/m,pop	64	+	+
FSTP	DF	223	11011111	01 111 r/m	+	64-i	ST(0)--> r/m,pop	64	+	+
FSTP	DF	223	11011111	10 111 r/m	++	64-i	ST(0)--> r/m,pop	64	+	+
FSTP	DB	219	11011011	00 111 110	++	80-t	ST(0)--> r/m,pop	80	+	+
FSTP	DB	219	11011011	00 111 r/m	-	80-t	ST(0)--> r/m,pop	80	+	+
FSTP	DB	219	11011011	01 111 r/m	+	80-t	ST(0)--> r/m,pop	80	+	+
FSTP	DB	219	11011011	10 111 r/m	++	80-t	ST(0)--> r/m,pop	80	+	+
FSTP	DF	223	11011111	00 110 110	++	80-d	ST(0)--> r/m,pop	80	+	+
FSTP	DF	223	11011111	00 110 r/m	-	80-d	ST(0)--> r/m,pop	80	+	+
FSTP	DF	223	11011111	01 110 r/m	+	80-d	ST(0)--> r/m,pop	80	+	+
FSTP	DF	223	11011111	10 110 r/m	++	80-d	ST(0)--> r/m,pop	80	+	+
FSTSW	DD	221	11011101	00 111 110	++	16bit	RM --> r/m	16	+	+
FSTSW	DD	221	11011101	00 111 r/m	-	16bit	RM --> r/m	16	+	+
FSTSW	DD	221	11011101	01 111 r/m	+	16bit	RM --> r/m	16	+	+
FSTSW	DD	221	11011101	10 111 r/m	++	16bit	RM --> r/m	16	+	+

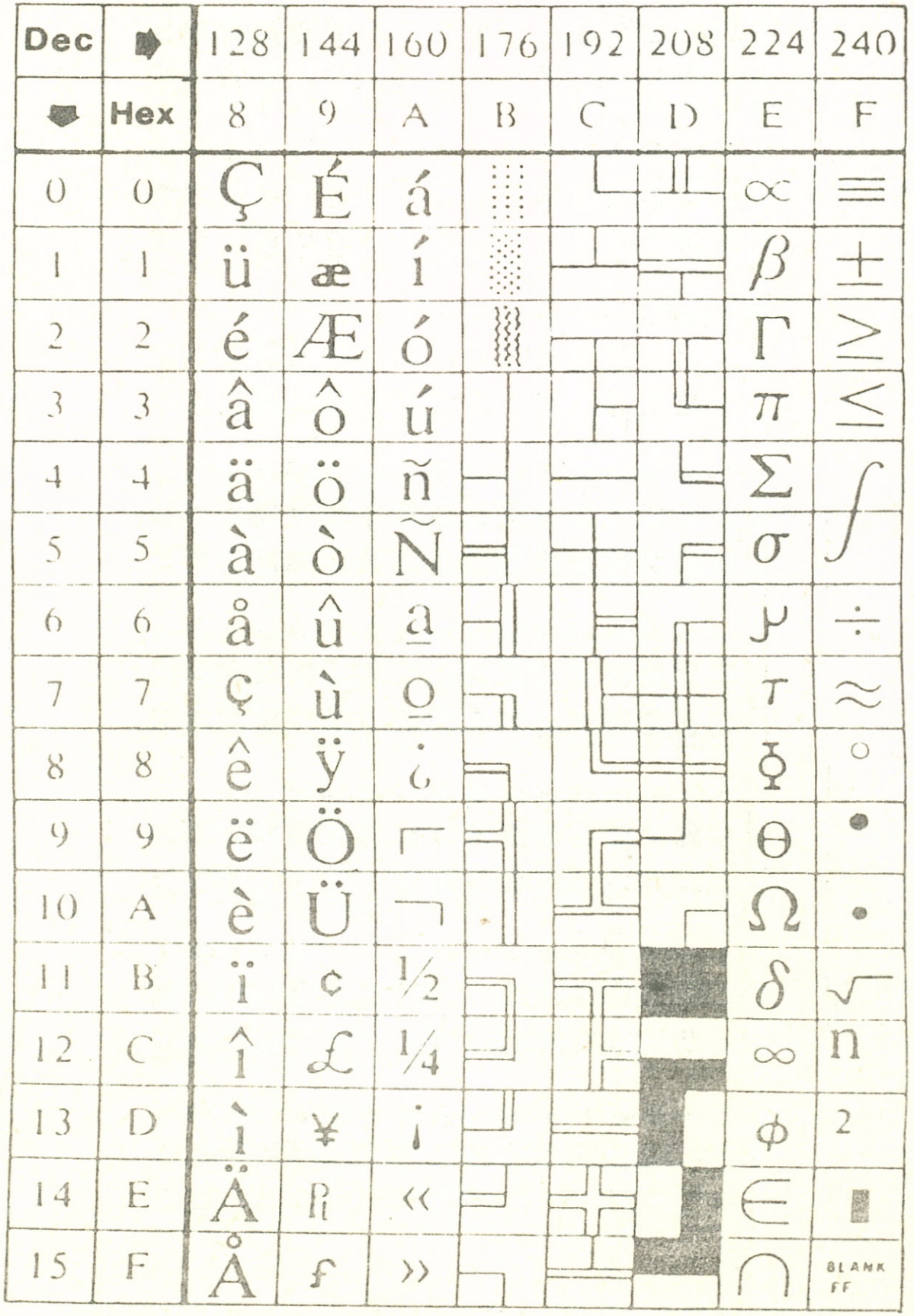
mnemo	opkód1 hex	dec	bináris	opkód2	elt. byte	mem. adat	adat-irány forrás --> cél	bit műv.	87	287
FSTSW AX	DF	223	11011111	11 100 000	-	-	SR --> AX	16	-	+
FSUB	D8	216	11011000	00 10R 110	++	32-r	r/m --> ST(0)	80	+	+
FSUB	D8	216	11011000	00 10R r/m	-	32-r	r/m --> ST(0)	80	+	+
FSUB	D8	216	11011000	01 10R r/m	+	32-r	r/m --> ST(0)	80	+	+
FSUB	D8	216	11011000	10 10R r/m	++	32-r	r/m --> ST(0)	80	+	+
FSUB	D8	216	11011000	11 10R ST(i)	-	-	ST(i)--> ST(0)	80	+	+
FSUB	DA	218	11011010	00 10R 110	++	32-i	r/m --> ST(0)	80	+	+
FSUB	DA	218	11011010	00 10R r/m	-	32-i	r/m --> ST(0)	80	+	+
FSUB	DA	218	11011010	01 10R r/m	+	32-i	r/m --> ST(0)	80	+	+
FSUB	DA	218	11011010	10 10R r/m	++	32-i	r/m --> ST(0)	80	+	+
FSUB	DA	218	11011010	11 10R ST(i)	-	-	ST(i)--> ST(0),pop	80	+	+
FSUB	DC	220	11011100	00 10R 110	++	64-r	r/m --> ST(0)	80	+	+
FSUB	DC	220	11011100	00 10R r/m	-	64-r	r/m --> ST(0)	80	+	+
FSUB	DC	220	11011100	01 10R r/m	+	64-r	r/m --> ST(0)	80	+	+
FSUB	DC	220	11011100	10 10R r/m	++	64-r	r/m --> ST(0)	80	+	+
FSUB	DC	220	11011100	11 10R ST(i)	-	-	ST(0)--> ST(i)	80	+	+
FSUB	DE	222	11011110	00 10R 110	++	16-i	r/m --> ST(0)	80	+	+
FSUB	DE	222	11011110	00 10R r/m	-	16-i	r/m --> ST(0)	80	+	+
FSUB	DE	222	11011110	01 10R r/m	+	16-i	r/m --> ST(0)	80	+	+
FSUB	DE	222	11011110	10 10R r/m	++	16-i	r/m --> ST(0)	80	+	+
FSUB	DE	222	11011110	11 10R ST(i)	-	-	ST(0)--> ST(i),pop	80	+	+
FTST	D9	217	11011001	11 100 100	-	-	ST(0)	80	+	+
FXAM	D9	217	11011001	11 100 101	-	-	ST(0)	80	+	+
FXCH	D9	217	11011001	11 001 ST(i)	-	-	ST(i)<-->ST(0)	80	+	+
FXTRACT	D9	217	11011001	11 110 100	-	-	ST(0)-->ST(0),push	160	+	+
FYL2X	D9	217	11011001	11 110 001	-	-	ST(0),ST(1)-->ST(1), pop	80	+	+
FYL2XP1	D9	217	11011001	11 111 001	-	-	ST(0),ST(1)-->ST(1), pop	80	+	+
FWAIT	9B	155	10011011	-	-	-	-	-	+	+

00H - 7FH

Dec ➡	0	16	32	48	64	80	96	112
⬇ Hex	0	1	2	3	4	5	6	7
0	0	BLANK (NULL)	►	BLANK (SPACE)	0	@	P	`	p
1	1	☺	◄	!	1	A	Q	a	q
2	2	☻	↕	"	2	B	R	b	r
3	3	♥	!!	#	3	C	S	c	s
4	4	♦	¶	$	4	D	T	d	t
5	5	♣	§	%	5	E	U	e	u
6	6	♠	▬	&	6	F	V	f	v
7	7	•	↨	'	7	G	W	g	w
8	8	◘	↑	(	8	H	X	h	x
9	9	○	↓	)	9	I	Y	i	y
10	A	◙	→	*	:	J	Z	j	z
11	B	♂	←	+	;	K	[	k	{
12	C	♀	∟	,	<	L	\	l	¦
13	D	♪	↔	—	=	M	]	m	}
14	E	♫	▲	.	>	N	^	n	~
15	F	☼	▼	/	?	O	_	o	△

IBM PC karakterkészlet (I.)

80H - FFH



IBM PC karakterkészlet (II.)

0	00		16	10	▶	32	20		48	30	0	64	40	@	80	50	P	96	60	`	112	70	p
1	01	☺	17	11	◀	33	21	!	49	31	1	65	41	A	81	51	Q	97	61	a	113	71	q
2	02	☻	18	12	↕	34	22	"	50	32	2	66	42	B	82	52	R	98	62	b	114	72	r
3	03	♥	19	13	‼	35	23	#	51	33	3	67	43	C	83	53	S	99	63	c	115	73	s
4	04	♦	20	14	¶	36	24	$	52	34	4	68	44	D	84	54	T	100	64	d	116	74	t
5	05	♣	21	15	§	37	25	%	53	35	5	69	45	E	85	55	U	101	65	e	117	75	u
6	06	♠	22	16	▬	38	26	&	54	36	6	70	46	F	86	56	V	102	66	f	118	76	v
7	07	•	23	17	↨	39	27	'	55	37	7	71	47	G	87	57	W	103	67	g	119	77	w
8	08	◘	24	18	↑	40	28	(	56	38	8	72	48	H	88	58	X	104	68	h	120	78	x
9	09	○	25	19	↓	41	29	)	57	39	9	73	49	I	89	59	Y	105	69	i	121	79	y
10	0A	◙	26	1A	→	42	2A	*	58	3A	:	74	4A	J	90	5A	Z	106	6A	j	122	7A	z
11	0B	♂	27	1B	←	43	2B	+	59	3B	;	75	4B	K	91	5B	[	107	6B	k	123	7B	{
12	0C	♀	28	1C	∟	44	2C	,	60	3C	<	76	4C	L	92	5C	\	108	6C	l	124	7C	¦
13	0D	♪	29	1D	↔	45	2D	-	61	3D	=	77	4D	M	93	5D	]	109	6D	m	125	7D	}
14	0E	♫	30	1E	▲	46	2E	.	62	3E	>	78	4E	N	94	5E	^	110	6E	n	126	7E	~
15	0F	☼	31	1F	▼	47	2F	/	63	3F	?	79	4F	O	95	5F	_	111	6F	o	127	7F	⌂

IBM PC karakterkészlet decimális és hexadecimális kódokkal (I.)

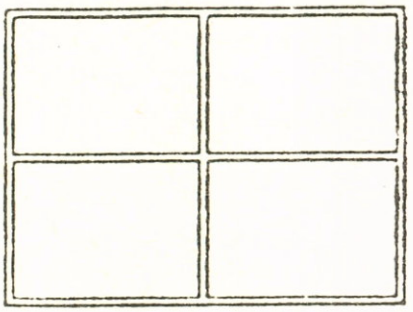
128	80	Ç	144	90	É	160	A0	á	176	B0	░	192	C0	└	208	D0	╨	224	E0	α	240	F0	≡
129	81	ü	145	91	æ	161	A1	í	177	B1	▒	193	C1	┴	209	D1	╤	225	E1	β	241	F1	±
130	82	é	146	92	Æ	162	A2	ó	178	B2	▓	194	C2	┬	210	D2	╥	226	E2	Γ	242	F2	≥
131	83	â	147	93	ô	163	A3	ú	179	B3	│	195	C3	├	211	D3	╙	227	E3	π	243	F3	≤
132	84	ä	148	94	ö	164	A4	ñ	180	B4	┤	196	C4	─	212	D4	╘	228	E4	Σ	244	F4	⌠
133	85	à	149	95	ò	165	A5	Ñ	181	B5	╡	197	C5	┼	213	D5	╒	229	E5	σ	245	F5	⌡
134	86	å	150	96	û	166	A6	ª	182	B6	╢	198	C6	╞	214	D6	╓	230	E6	μ	246	F6	÷
135	87	ç	151	97	ù	167	A7	º	183	B7	╖	199	C7	╟	215	D7	╫	231	E7	τ	247	F7	≈
136	88	ê	152	98	ÿ	168	A8	¿	184	B8	╕	200	C8	╚	216	D8	╪	232	E8	Φ	248	F8	°
137	89	ë	153	99	Ö	169	A9	⌐	185	B9	╣	201	C9	╔	217	D9	┘	233	E9	Θ	249	F9	·
138	8A	è	154	9A	Ü	170	AA	¬	186	BA	║	202	CA	╩	218	DA	┌	234	EA	Ω	250	FA	·
139	8B	ï	155	9B	¢	171	AB	½	187	BB	╗	203	CB	╦	219	DB	█	235	EB	δ	251	FB	√
140	8C	î	156	9C	£	172	AC	¼	188	BC	╝	204	CC	╠	220	DC	▄	236	EC	∞	252	FC	ⁿ
141	8D	ì	157	9D	¥	173	AD	¡	189	BD	╜	205	CD	═	221	DD	▌	237	ED	φ	253	FD	²
142	8E	Ä	158	9E	₧	174	AE	«	190	BE	╛	206	CE	╬	222	DE	▐	238	EE	ε	254	FE	■
143	8F	Å	159	9F	ƒ	175	AF	»	191	BF	┐	207	CF	╧	223	DF	▀	239	EF	∩	255	FF	

IBM PC karakterkészlet decimális és hexadecimális kódokkal (II.)

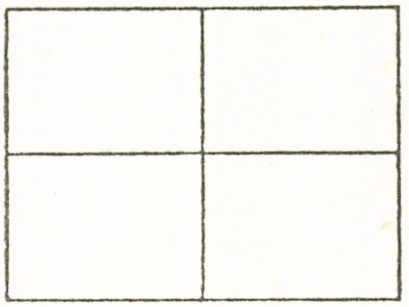
Dec. kód	Hex. kód	Ctrl + bill.	Név	Megnevezés
0	00	^@	NUL	null karakter
1	01	^A	SOH	fejrész kezdete (*start of header*)
2	02	^B	STX	szöveg kezdete (*start of text*)
3	03	^C	ETX	szöveg vége (*end of text*)
4	04	^D	EOT	átvitel vége (*end of transmission*)
5	05	^E	ENQ	kérés (*enquire*)
6	06	^F	ACK	nyugtázás (*acknowledge*)
7	07	^G	BEL	csengő (*bell*)
8	08	^H	BS	*backspace*
9	09	^I	HT	vízszintes tab (*horizontal tab*)
10	0A	^J	LF	soremelés (*line feed*)
11	0B	^K	VT	függőleges tab (*vertical tab*)
12	0C	^L	FF	lapdobás (*form feed*)
13	0D	^M	CR	kocsi vissza (*carriage return*)
14	0E	^N	SO	*Shift Out*
15	0F	^O	SI	*Shift In*
16	10	^P	DEL	törlés (*delete*)
17	11	^Q	DC1	*device control 1*
18	12	^R	DC2	*device control 2*
19	13	^S	DC3	*device control 3*
20	14	^T	DC4	*device control 4*
21	15	^U	NAK	negatív nyugt. (*negative ack*)
22	16	^V	SYN	szinkronizálás (*syncronize*)
23	17	^W	ETB	szövegblokk vége (*end of text block*)
24	18	^X	CAN	érvénytelenítés (*cancel*)
25	19	^Y	EM	közeg vége (*end of medium*)
26	1A	^Z	SUB	helyettesítés (*substitute*)
27	1B	^[	ESC	megszakítás (*escape*)
28	1C	^/	FS	file szeparátor
29	1D	^]	GS	csoport szeparátor
30	1E	^^	RS	rekord szeparátor
31	1F	^_	US	egység szeparátor

ASCII vezérlő karakterek.

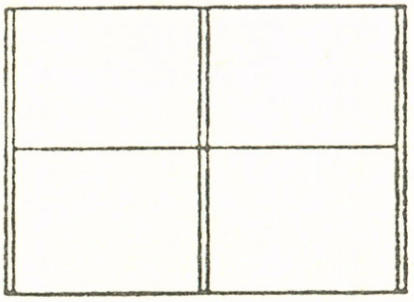
Kettős vonalak:



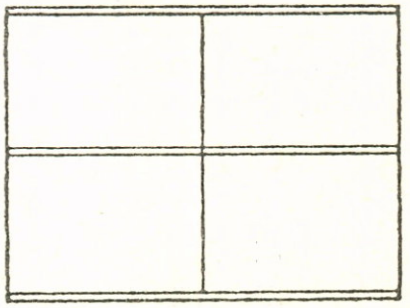
Egyes vonalak:



Kettős függőleges vonalak:



Kettős vízszintes vonalak:



Kettős vonalak:

╔	═	╦	═	╗
201	205	203	205	187
‖		‖		‖
186		186		186
╠	═	╬	═	╣
204	205	206	205	185
‖		‖		‖
186		186		186
╚	═	╩	═	╝
200	205	202	205	188

Egyes vonalak:

┌	─	┬	─	┐
218	196	194	196	191
│		│		│
179		179		179
├	─	┼	─	┤
195	196	197	196	180
│		│		│
179		179		179
└	─	┴	─	┘
192	196	193	196	217

Kettős függőleges vonalak:

╓	─	╥	─	╖
214	196	210	196	183
‖		‖		‖
186		186		186
╟	─	╫	─	╢
199	196	215	196	182
‖		‖		‖
186		186		186
╙	─	╨	─	╜
211	196	208	196	189

Kettős vízszintes vonalak:

╒	═	╤	═	╕
213	205	209	205	184
│		│		│
179		179		179
╞	═	╪	═	╡
198	205	216	205	181
│		│		│
179		179		179
╘	═	╧	═	╛
212	205	207	205	190

Táblázat rajzoló karakterek.

Tárgymutató

2764 / 88
27128 / 88
4164 / 89
41256 / 89
6845 / 234-244
8042 / 158-164
8048 / 103
8086-8088-80286 / 15-27, 64, 71-87, 120-122, 125-134
- 8088-8086 kivezetései / 71-87
- 80286 jellemzők / 120-122, 125-134
- regiszterek / 18, 19
- szegmentáció / 16, 17
- utasítások felépítése és címzés / 20-27 ld. még: címzési módok
8087-80287 / 28-61, 66, 85, 86, 125, 133
- egész (integer) formátumok / 35
- kivételek (exceptions) / 30, 133
- lebegőpontos számábrázolás / 28
- regiszterkészlet / 36
számregiszterek / 37
vezérlő regiszter / 38
státusz regiszter / 39
toldalék (tag) regiszter / 40
kivétel mutató (exception pointer) / 41
- utasítások / 41-61
8089 / 85
8237A / 84, 93, 110-113, 154, 155
8253 / 93, 107-110
8254-2 / 150-152
8255 / 93, 101-106
8257 / 110
8259A / 83, 93, 96-98, 134
8284A / 71, 85, 125
8288 / 71
82284 / 120, 125
82288 / 120, 125, 134
MC 146818 / 165-173

A

adatátviteli utak (AT alapkártya) / 131, 132
adat buffer / 81, 91, 92
alapkártya csatlakozók
- AT / 185-187
- XT / 69, 119
ALE (Address Latch Enable) / 80

alfanumerikus mód / 247-250
AT alapkártya / 120
AT alapkártya blokkdiagram / 125-127
AT alaplemez leírás / 122
AT alaprendszer blokkdiagram / 123
attribútumkód / 246, 249
áramköri tervezés (I/O adapterek) / 298-304

B

bank-switched memory (csoport-kapcsolt memória) / 138
billentyűzet ld.: keyboard
billentyűzet interface ld.: keyboard interface
BIOS / 230 ld. még: ROM BIOS
blokkdiagram
- AT alapkártya / 125-127
- AT alaprendszer / 123
- XT alapkártya / 71-73
- XT rendszer vázlata / 68
bővített memória ld.: Expanded Memory
bővített memória kezelő ld.: EMM
byte csere (byte swap) / 129-131

C

CALL / 99
CGA grafikus monitor adapter kártya / 71, 88
címvonalak / 80
címzési módok (8086-8088-80286)
- indirekt / 23
- közvetlen (immediate) / 25
- közvetlen memória / 24
- regiszter / 20, 22
CMOS RAM és óraáramkörök (MC 146818) / 165-173
- CMOS RAM címtérkép / 165
CRT vezérlő ld.: display vezérlő
csoport-kapcsolt memória ld.: bank-switched memory
"csővonalas" címzés ld.: pipelined addressing

D

DESQview / 140, 141, 143
DIP kapcsolósor (XT alaplemez rendszerkonfiguráció) / 104

DIP kapcsoló beállítási lehetőségei / 106
display adapter kártyák / 245-264
- IBM monochrome display- és printer adapter / 245-247
- IBM színes grafikus monitor adapter (CGA) / 247-257
a CGA adapter csatlakozói / 257
a CGA adapter regiszterei / 252-256
- nagyobb felbontású monitor adapterek (Herc, EGA, VGA) / 258-264
display megjelenítési módok / 259, 262, 263
display típusok / 260, 261

E

EEMS (Enhanced Expanded Memory Specification) / 140
EGA grafikus kártya ld.: display vezérlők
egér (mouse) / 227-231
- adapter kártyára csatlakozó egerek / 228
- egér BIOS függvényhívások / 230, 231
- működési paraméterek / 229
- RS-232 soros egér csatlakozó kiosztás / 227
EMM (Expanded Memory Manager) bővített memória kezelő / 139
EMS (Expanded Memory Specification) / 139
EPROM / 88, 141
Exception (kivétel) ld.: 8087-80287
Expanded Memory / 138, 139
Extended Memory / 121, 135, 138

F

FIFO / 86
floppy meghajtó adapter / 292-295
fordító programok / 86
frissítés / 90, 143

G

grafikus adapterek ld.: display adapterek
grafikus módok / 247, 250-252

H

handshake / 101
hangszóró interface / 109, 110, 150, 151

hardware interrupt lista
- AT / 145
- XT / 98
hibakeresés (debug) / 306-308

I

IBM PC alapkártya / 67
IBM PC rendszer-komponensek / 65
IBM protípus kártya / 309-312
intelligens periféria IC chip / 92, 93, 101
interrupt kezelés / 83
(nem maszkolható interrupt ld.: NMI)
interrupt rendszer (AT) / 145
- hardware interrupt lista / 145
- interrupt vektor tábla / 146-148
- osztott-interrupt kezelés / 149
interrupt rendszer (XT) / 96
- hardware interrupt lista / 98
- interrupt vektor / 99, 100, 101
I/O adapter kártyák tervezése / 298-304
I/O CH READY / 82
I/O címdekódolás / 92, 93, 303
I/O címtérkép (AT) / 173, 174
I/O címtérkép (XT) / 94, 95
I/O csatorna
- AT / 175-185
- XT / 114-119
I/O csatorna kiterjesztése / 68, 313-320
- extender card / 315-318
- receiver card / 318-320
I/O eszközök csatolása az alapkártyán / 92

J

játékvezérlő (joy-stick) / 274
joy-stick csatlakozó / 274

K

kapcsolási rajz
- keyboard / 225
- prototípus kártya / 310

- tápegység / 194
- XT alapkártya / 74-79
karakter attribútum kód / 246, 249
"kártya kiválasztva" (card selected) jel / 115, 118
keyboard / 103, 197-226
- 84 gombos billentyűzet sorszámok és scan kódok / 204-206
- 101/102 gombos billentyűzet / 207-211
- 101/102 gombos billentyűzet scan kódok / 212-224
- adatátvitel / 202-203
- AT rendszerparancsok (84 billentyűs kivitel) / 199-201
- AT billentyűzet parancsok (84 billentyűs kivitel) / 201
- BAT (Basic Assurance Test) / 198
- kapcsolási vázlat / 224-226
- POR (Power-On Reset) / 198
keyboard interface / 158-164
képernyő letapogatási módok / 234
kiterjesztett memória ld.: Extended Memory
konkurrens működés / 55
közvetlen memória hozzáférés ld.: DMA

L

letapogatási módok / 234
LIM-EMS (Lotus/Intel/Microsoft Expanded Memory Specification) / 139
lokális busz (80286) / 128

M

"maximum" mód (8086-8088) / 71, 83
mágneslemez-meghajtó interface (floppy) / 269
megszakítás kérések ld.: interrupt
memóriák
- AT memória címtérkép / 134, 135, 144
- AT RAM / 142, 143, 144
- AT ROM / 134, 141
- bővített memória ld.: Expanded Memory
- CMOS RAM ld.: MC 146818 és CMOS RAM
- csoport-kapcsolt memória ld.: bank-switched memory
- kiterjesztett memória ld.: Extended Memory
- XT memória címtérkép / 87
- XT memória vezérlő / 91
- XT RAM / 87, 88, 89, 90, 91
- XT ROM / 87, 88, 89
memory mapped I/O eszközök / 81, 303

memory mapping / 136
merevlemezes és floppy meghajtó adapter (PC/AT) / 292-295
merevlemezes meghajtó adapter (IBM) / 285-292
merevlemezes tárolók típusai / 169, 170, 280-284
monitor adapterek ld. display adapterek
monitorok ld. display-k
mouse ld. egér
Multi I/O kártya / 265-274
Multi-tasking / 121, 122, 140

N

NMI (Nonmaskable Interrupt) / 84, 91, 94, 99, 104, 114, 134, 175
NYÁK tervezés / 305, 306
- tervező programok smARTWORK, RACALREDAC / 305, 306
nyomtató ld.: printer

O

operációs rendszer / 64, 138
- CP/M-86 / 64
- DOS / 67, 68, 88, 99, 138
- PC DOS / 64, 66, 122
óra/naptár / 273
óra/naptár program / 274-278
OrCAD/SDT / 300-302
osztott-interrupt kezelés / 149

P

paritás bit / 64, 89, 91
PC DOS / 64, 66, 122
PCjr / 83
pipelined addressing / 128
'Power Good' jel / 84, 191
PPI (Programmable Parallel Interface) / 101
PPI bittérkép / 105, 151
printer adapter / 245, 247, 270
prototípus kártya / 309-312

R

RAM ld.: memóriák
'real time' órajelek / 107, 109
redundáns címzési mód / 93
regiszterek (8086-8088-80286) / 15, 18
regiszterek (8087-80287) / 36
rendszer konfigurációt beállító DIP kapcsolók ld. DIP kapcsolósor
RESET jel / 71, 84, 102
ROM ld.: memóriák
ROM BIOS / 88, 91, 97, 98, 103, 109, 141, 142
RS-232 soros illesztő / 271, 272

S

scan kódok / 103, 162, 164, 204-206, 212-224
SETUP program / 125, 166
software interruptok / 99
system unit (AT) / 122
számláló/időzítő ld.: 8253, 8254-2
szegmens leíró (segment descriptor) / 136
szegmentáció / 16, 136
színkiválasztás / 255
színkombináció / 250, 251
színpaletta / 250, 251

T

tápegység / 189-195
- csatlakozó, terhelhetőség / 192
- kapcsolási rajz / 193-196
tervezési szempontok / 297-304
TIMER.COM / 274-278
TURBO-640 / 71, 82, 120

U

utasítások (8086-8088-80286) / 20-27, 324-348
utasítások (8087-80287) / 41-61, 349-357
- adatmozgató utasítások / 42
- adminisztratív utasítások / 58
- aritmetikai utasítások / 45
- komparáló utasítások / 48

- regiszter felcserélő utasítások / 44
- tároló utasítások / 44
- töltő utasítások / 43, 45
- transzcendentális utasítások / 51

V, W

valós címzési mód / 15, 120
vezérlő busz (8088) / 81
video adapterek ld.: display adapterek
virtuális memória / 135, 136
védett virtuális mód / 15, 120-122, 136
Winchester diszkek ld.: merevlemezes tárolók

X

XT alapkártya alkatrész elrendezés / 70
XT alapkártya blokkdiagram / 71
XT alapkártya leírás / 69
XT személyi számítógép / 67

1077 Majakovszkij u. 91.

Telefon: 221-076

Mikroszámítógépek alkalmazása:

- szakkönyvek
- Software
- Hardware
- tanácsadás
- ...

MEGRENDELEM AZ LSI ATSz KIADVÁNYAI KÖZÜL AZ ALÁBBIAKAT:

.... **db REXLIB** + melléklet
(Képernyő- és file-kezelő rezidens rutin és utility csomag MS-DOS és NOVELL Netware 286 környezetben (Turbo Pascal 4.0 alatt is) fejlesztők számára) 205.-Ft
.... db Orosz Judit: **IBM PC-HÁLÓZATOK ÁTTEKINTÉSE**
(LAN RENDSZEREK: Novell, NetWare, PC Network, 3Com3 +) 123.-Ft
.... db Erdős-Rusznák: **IBM PC XT/AT INFORMÁCIÓS KÁRTYA** 126.-Ft
.... db Dr.Pajor G.: **IBM PC-RŐL KEZDŐ FELHASZNÁLÓKNAK I. A HARDVER** 66.-Ft
.... db Dr.Pajor G.: **IBM PC-RŐL KEZDŐ FELHASZNÁLÓKNAK II. A SZOFTVER** ... 70.-Ft
.... db Kápolnai A.-Geszti P.: **IBM PC BASIC FEJLESZTŐRENDSZER** 236.-Ft
.... db Pirkó J.: **3D PERSPEKTÍVIKUS GRAFIKA TURBO PASCAL-BAN** 149.-Ft
.... db Dr.Úry-Bernáth-Darvas-Orosz: **IBM PC DOS I-II-III.**
(3.30 verzióval bővített kiadás) 692.-Ft
.... db Dr.Úry László: **SYMPHONY I.** 160.-Ft
.... db Dr.Úry László: **SYMPHONY II.** 133.-Ft
.... db Dr.Barakonyi: **IBM PC TÁBLÁZATKEZELŐ RENDSZEREK**
(LOTUS 1-2-3, FRAMEWORK, CALCSTAR) 314.-Ft
.... db Ráth-Éltető: **PROGRAM- ÉS ADATVÉDELEM IBM PC-RE** 97.-Ft
.... db Sipos-Soltész: **SZÖVEGSZERKESZTŐK IBM PC-re I.**
(WORDSTAR, CHIWRITER) 248.-Ft
.... db Bartha Attila: **NORTON**
(UTILITEIS, INTEGRATOR, EDITOR, COMMANDER, GUIDE) 150.-Ft
.... db Pirkó J.: **TURBO PASCAL** kezdőknek - haladóknak **4.0 verzióig** 192.-Ft
.... db Dóra és trsai.: **AZ IBM PC BELSŐ FELÉPÍTÉSE** 399.-Ft
.... db Mondovics és trsai.: **DBASE III PLUS, FOXBASE + , CLIPPER** I-II.k.... Irányár 450.-Ft

.... db Dr.Kovács Magda: **ANGOL-MAGYAR MIKROELEKTRONIKAI SZÓTÁR** 197.-Ft
.... db Berkes - Balogh - Kovács: **A SZÁMÍTÓGÉPES TÁVKÖZLÉS TELEMATIKAI SZOLGÁLATAI: TELETEX, FAKSZIMILE, VIDEOTEX**
(A könyvet a klasszikus hírközlés területén dolgozó szakembereknek és azoknak az informatikával foglalkozó olvasóknak ajánljuk, akik a számítógépes információcsere szabványokon alapuló megvalósulási formái iránt érdeklődnek.) 463.-Ft
.... db Dr.Arz-Dr.Lipóth-Dr.Merksz: **ROBOTMANIPULÁTOROK** 350.-Ft
.... db Dr.Siegler András: **ROBOTIRÁNYÍTÁSI MODELLEK** 350.-Ft

Név: .....

Cím: .....

Dátum: .....

Levélcím: LSI ATSz, Budapest 1300 Postafiók 114

Jegyzetek

AZ LSI ATSZ KÖNYVAJÁNLATA

SZÖVEGSZERKESZTŐK AZ IBM PC-re 1. kötete
Soltész-Sipos: **WORDSTAR, CHIWRITER**

Ezzel a kötettel megkezdjük az IBM PC számítógépeken futó szövegszerkesztőket ismertető sorozatunkat. A kiadványokat – szokásunk szerint – egyaránt szánjuk kezdőknek és haladóknak.
E könyv első része alapismereteket tartalmaz. Ezután fejtjük ki mind az első részben már tárgyalt funkciók további lehetőségeit, valamint azokat az ismereteket, amelyek ugyan nem nélkülözhetetlenek, de – a gyors és hatékony munkához – nagyon hasznosak.
Előbb a WordStar-t, majd a ChiWriter-t ismertetjük. Mindkét anyagban külön részben foglalkozunk az elemi szövegfelviteli- és nyomtatási tudnivalókkal, hogy e – rövidebb – rész végére érve már az Olvasó akár "üzemszerűen" is használhassa az addig tanultakat.
Ára: 248.-Ft. 290 oldal

Ráth Gy.-Éltető L.: **PROGRAM– ÉS ADATVÉDELEM** IBM PC-re

Ez a kiadványunk átdolgozott, javított, és az adatvédelemmel kibővített kiadás.
A könyv hasznos lehet azoknak, akik megprobálkoznak a program- és adatvédelem készítésével, mivel nemcsak hasznos ötleteket kaphatnak munkájukhoz, de a könyv elolvasása igen sok buktatótól is megkímélheti őket.
Ára: 97.-Ft. 65 oldal

LEVÉLCÍM: LSI ATSZ
1300 BUDAPEST, Pf. 114

Tordai Attila: **VAX BASIC** programozási segédlet

A könyv hatásos segítséget nyújthat mindazoknak, akik a TPA-11 számítógépek 500-as sorozatán vagy Mikrosztár-31,32,33 gépeken BASIC nyelven programoznak.
A VMS operációs rendszer sokkal több szolgáltatást nyújt a fejlesztőknek és felhasználóknak, mint a kötegelt feldolgozást biztosító vagy a korábban használt egyszerűbb interaktív rendszerek, vagy akár a széles körben elterjedt személyi számítógépek.
A VAX BASIC kielégít szinte minden, a jól kezelhető magas szintű nyelvekkel szemben támasztott igényt, és széles körű alkalmazást tesz lehetővé.
Ára: 207.-Ft. 128 oldal

Pirkó József: **TURBO PASCAL** kezdőknek, haladóknak 4.0 verzióig

A szerző a PASCAL nyelv IBM PC és vele kompatibilis számítógépeken való megvalósításával foglalkozik. A nyelvet nem ismerők ebből képet kaphatnak az utasítások fajtáiról, használatuk miként jéről.
A következő fejezet a szöveg- és a filekezeléshez szükséges ismereteket tartalmazza. Nagyon hasznos a programok kapcsolódási lehetőségeinek bemutatása, már meglévő alprogramok új programba való beillesztésének leírása.Végül a TURBO PASCAL 4.0-ás verzió eltéréseit olvashatjuk az eddigiekhez képest.
A szerzőnek nagy tapasztalata van a nyelv használatában, és ezeket a példákon keresztül át is adja.
A könyv hiányt pótol, tankönyvnek is használható.
Ára:192.-Ft 245 oldal

EltSECRET

Ára: 399,– Ft

EltSECRET adatvédelmi rendszer

Termékünk, az EltSECRET, a SW'88 díjazott programjának, az EltGUARD-nak adatvédelemre továbbfejlesztett változata.

IBM PC-n működő adatvédelmi rendszer, mely nemcsak az adatokhoz és programokhoz való hozzáférést szabályozza, hanem adattitkosításra (kódolásra) is lehetőséget ad.

Az EltSECRET rendszer jellemzői:

- 64 hierarchia-szint a fájlok védelmére
- 127 alkönyvtárszint
- 2500 jogosult felhasználó
- 3 különböző bonyolultságú kódolási eljárás
- együttműködés lokális hálózati rendszerekkel
- felhasználói menürendszerek kialakítása
- kiegészítő hardverrel fokozott biztonság

Hardverkiegészítő használata esetén a felhasználók fizikai személyi azonosítót kapnak, jelszavuk mellett ez is kell a bejelentkezéshez. A herdver letiltja a floppyról történő rendszerbetöltést is.

A teljes program ára: **76.000,– Ft** (+ ÁFA).
A DEMO változat ára: **8.000,– Ft** (+ ÁFA).

Egy telephely összes számítógépéhez elegendő egy példány EltSECRET rendszert vásárolni.

Bővebb felvilágosítással, bemutatóval állunk rendelkezésére:

eltsoft

EltSOFT
Számítástechnikai és Reklámszervező GMK
1122 Budapest, Városmajor u. 7–9.
Telefon: 550-962 1536 Budapest, Pf. 390.