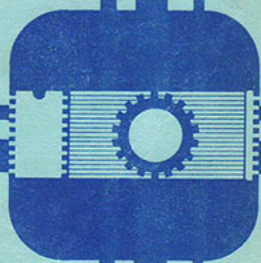


BORBOLA ISTVÁN
A SZÁMÍTÓGÉP
ÉS KÖRNYEZETE

ORSZÁGOS PEDAGÓGIAI INTÉZET



VORKER®



A számítógép és környezete

**Egyszerű interfészek C 16-os
és C Plus 4-es házi számítógépekhez**

BORBOLA ISTVÁN

Főszerkesztő:
dr. Szűcs Barna

Szerkesztő:
Borbola István

A kiadvány szerzője:
Borbola István

Bírálták:
Dr. Sára Attila
Dr. Szűcs Ervin

ISBN: 963 682 160 7 összkiadás

ISBN: 963 632 162 3

Kiadja az Országos Pedagógiai Intézet Számítástechnikai Programirodája
Felelős kiadó: Szabolcsi Miklós főigazgató

TARTALOMJEGYZÉK

A SOROZAT TAGJAI	5
Előszó	9
Bevezető	11
1. Amit a kapcsolatteremtés előtt tudni kell	13
1.1. Kódok az adatsínen	13
1.2. Játsszunk a kódokkal	16
1.3. A mikorszámítógép működésének lényege	18
1.4. Kapuk a külvilágra	21
1.5. Szemlélődünk a kapcsolási rajzon	24
1.6. Építsünk logikai tesztet	30
1.7. A NYÁK-készítés legegyszerűbb módszere	33
2. Kapcsolat a processzor-porton	36
2.1. Logikai értékek és feszültség szintek érzékelése	36
2.2. 8 csatornás demultiplexer a SERIAL PORT-on	41
2.3. A SERIAL PORT védelme illesztő áramkörrel	47
3. „Ráülünk” a belső buszra	53
3.1. Mélyebben elemezzük a számítógép lelki világát	53
3.2. Egyszerű interfészt tervezünk	57
3.3. Az I/O eszközt is az interfész választja ki	63
3.4. I/O áramkörök a rendszersínen	65
3.5. Programozható periféria illesztő (I 8255) a rendszersínen	70
4. Az eszközök fényképei	79
5. Az eszközök NYÁK-jainak beültetési rajzai	83
Hivatkozott irodalom	93

A sorozat tagjai

Az Országos Pedagógiai Intézet Számítástechnikai Programirodájának gondozásában 1987/88-ban 12 számítástechnikai témakörhöz készült oktatási anyagok C 16-os és C Plus/4-es Commodore házi számítógépekre:

1. Algoritmusok, játékok:

A kiadvány tömören, de közérthetően szemléletes példákon keresztül világít rá a probléma-algoritmus-program kapcsolatra, elvi és gyakorlati tanácsokat ad a jól felépített programok készítéséhez. A hozzá tartozó kazetta tizenkét, önmagában is alkalmazható, számítástechnikai és matematikai tanulságokkal teli programot tartalmaz. A programok jórészt listázhatók, így példát mutatnak egy színvonalas programozási stílus kialakítására, a gyakorlottabb felhasználóknak lehetőséget adnak arra, hogy más programok készítésénél is hasznosítsák az itt bemutatott programozási fogásokat.

Ajánlott: ált. isk. 7., 8. osztályosainak és középiskolásoknak.

2. A számítógép és környezete:

Mindazoknak készült, akik szeretnék összekötni saját számítógépüket a külvilággal. 8 db különböző interfész és periféria tervezése kapcsán ismerteti meg az olvasót az interfész tervezés és illesztés fogásaival. Útmutató és 8 kapcsolási rajz, kész nyomtatott áramköri tervrajz. *Ajánlott:* ált. isk. és középiskolai tanároknak, számítástechnikai amatőröknek, 8. osztályos és középiskolai tanulóknak, tanárjelölt főiskolai és egyetemi hallgatóknak.

3. Programmodulok:

A felhasználó 55 db szubrutint, programmodult, ill. önálló felhasználói programot kap. Ezek egyrésze gépi kódú rutin, melyet az átlagos felhasználó nem tud megírni, de beépítheti munkájába. A felhasználói programok ötletet adhatnak hasonló programok megírásához. Az egyes rutinok, programok alkalmazási lehetőségeinek, működésének leírását az útmutató tartalmazza. *Ajánlott:* ált. iskolai tanároknak, tanulóknak, kezdő amatőr programozóknak.

4. Adatkezelés számítógéppel C 16-ra és a C Plus/4-re:

A témakör feldolgozását segítő útmutató az adatkezelés, adatfeldolgozás alapfogalmaival és folyamatával foglalkozik. Megismerteti az adatfeldolgozó rendszerek fejlesztésével és e rendszerek alkalmazásbavételével. A feldolgozást 5 program és 2 minta-adatállomány segíti. *Ajánlott:* ált. és középiskolai tanároknak, fakultációs tananyagként, továbbképzési anyagként.

5. Szövegfeldolgozás számítógéppel:

A témakör feldolgozásához a felhasználó megkapja a Nemzetközi ABC szövegszerkesztő programot, amely magyar, cirill, német, lengyel, görög karakterkészletet és különleges matematikai jeleket (gyökjel, integráljel, alsó és felső index stb.) is tartalmaz. A szövegek és különleges karakterek nyomtathatók — akár fűzött kézírást imitálva is. A kézírásos nyomtatás főleg alsótagozatban számíthat sikerre. A szövegszerkesztő minden olyan szolgáltatást tud, amely a tudományos munkához, vagy a tanári munkához kell. A feldolgozást útmutató és bemutató programok is segítik. *Ajánlott:* tanároknak, kutatóknak, íróknak, adminisztrátoroknak, ill. fakultációk anyagaként.

6. Számítógépes grafika C 16-on és C Plus/4-en:

Az útmutató és a mintaprogramok a felhasználói kézikönyvből meg nem tanulható különleges grafikus lehetőségekkel ismertetik meg az olvasót. A többszínű háttérszín üzemmód mellett a bittérképes színes grafikák elkészítésének fortélyait is megtanulhatja a felhasználó. Az útmutató és a mintaprogramok áttanulmányozása után az olvasó saját maga is képes lesz a játékprogramokból ismert színes és mozgó figurák „szellemecskék” programozására. *Ajánlott:* ált. isk. 7., 8. osztályosainak, középiskolásoknak és a grafika programozását most kezdő programozóknak.

7. Hangkeltés C 16-os (és C Plus/4-es) számítógéppel:

Az anyaghoz az útmutató mellett bemutató programok is tartoznak. Ezek áttanulmányozása alkalmassá teszi a felhasználót egy és többszólamú dallamok programozására, az interrupt alatti programozási lehetőségek kihasználására. A programok és dallamok az ált. iskolai ének-zene oktatáshoz igazodnak, így felhasználhatók számítógépes motivációra. *Ajánlott:* ált. isk. 7., 8. osztályosoknak, középiskolásoknak, valamint ált. iskolai énektanároknak, kezdő programozóknak.

8. Számítógépes fogások, trükkök C 16-ra és C Plus/4-re:

Az útmutató és a hozzá tartozó 16 program azok számára mutatja be a C 16-on és a C Plus/4-en alkalmazható fogásokat, akik a BASIC programozás alapjain túljutottak. A közölt rendszerváltozók, memóriacímek és gépi kódú programok megismertetik az olvasót az alkalmazások különleges lehetőségeivel. *Ajánlott:* kezdő programozóknak, általános és középiskolásoknak, tanároknak.

9. Számítógép és video:

Az anyaghoz útmutató, videofilm és bemutató programok tartoznak. A videofilm a hozzá tartozó számítógépes programokkal egy megvalósítható rendszert és egy példát mutat be a képmagnó és a házi számítógép összekapcsolására, az oktatásban történő alkalmazására. Az anyag az elveken túl egy interfész elkészítésével is megismerteti az olvasót. Elsősorban tanároknak, oktatástechnikai szakembereknek ajánlott!

10. Számítógéppel vezérelt mérések:

Az útmutató megismerteti az olvasót a legfontosabb méréstani fogalmakkal, a mérések tervezésének, elvégzésének, értékelésének elméleti és műszaki-, méréstani alapjaival. A mérőberendezések és mérési összeállítások jelátvivő funkcionális egységeként a Tudományszervezési és Informatikai Intézetnél kapható TechnoMIR moduláris interfészrendszer berendezéseit mutatja be és használja fel az anyag. A témakörhöz tartozó 21 db professzionális program egyrészt az oktatómunkában és a bemutató kísérletekben alkalmazható számítógép-vezérlésű intelligens műszert szimulál (pl. tároló oszcilloszkópot, 8 csatornás logikai analízátort, digitális multimétert, stb.) másrészt az iskolai kísérletekhez szükséges mérési összeállításokat vezérli. *Ajánlott:* általános és középiskolai tanároknak, speciális szakképző intézeteknek, pedagógusképző intézeteknek bemutatói és oktatási célra, pedagógiai intézeteknek a továbbképzéshez.

11. Számítógépes irányítástechnika:

Az anyaghoz útmutató, az útmutatóban ismertetett berendezésekhez 11 db mintaprogram tartozik. Ezek áttanulmányozása megismerteti az olvasót a számítógépes vezérlések és szabályozások alapfogalmaival, a mechatronikus modelleket működtető berendezések (kapcsolómodulok, interfészek) elkészítésével. A mintaprogramok elemzése példát mutat az irányítási algoritmusok programozására. Az irányító rendszerek jelátvivő funkcionális egységeként a Tudományszervezési és Informatikai Intézetnél kapható TechnoMIR interfészrendszer elemeit, érzékelő-, erősítő beren-

dezéseknek és beavatkozó szerveknek pedig saját készítésű, a felhasználó által is reprodukálható eszközöket használ az anyag. Ezek leírásait és elkészítési módját is tartalmazza. *Ajánlott:* az általános és középiskolai technika tárgy oktatásához, fakultációkhoz, szakköri feldolgozáshoz.

12. Számítógépek összekapcsolása, helyi oktatóhálózatok:

Az anyag egy hazánkban eddig nyomtatásban még meg nem jelent területtel, a Commodore házi számítógépek (és a VIDEOTON TV COMPUTER) összekapcsolásának gyakorlati megvalósításával és e számítógépekből összeállított helyi oktatóhálózatok programozásával foglalkozik. Megismerteti a felhasználóval a számítógépek közötti kapcsolat hardver és szoftver eszközeit, a kapcsolatteremtés és az adatátvitel alapfogalmait. Megépített és reprodukálható kapcsolásokkal és kész felhasználói programokkal segíti, hogy az anyagot feldolgozó olvasó saját maga is összekapcsolhasson számítógépeket. A megszerzett ismeretek segítségével a minimális elektronikai ismerettel és készüléképítési gyakorlattal rendelkező felhasználó is meg tud valósítani számítógépek közötti adatátvitelt. Az útmutató második része az oktatásban bevált és sikerrel alkalmazott, a szegedi VORKER Kiszövetkezet által kifejlesztett TC-NET oktatóhálózatokkal, azok programozási fogásaival foglalkozik. Az anyag 6 db kapcsolást és 12 db professzionális felhasználói programot tartalmaz. *Ajánlott:* általános és középiskolai tanároknak, 3., 4. éves középiskolai tanulóknak, számítástechnikai amatőröknek, középfokú szakképző intézeteknek, tanárképző intézeteknek, pedagógiai intézeteknek, továbbképző intézeteknek stb.

Valamennyi itt felsorolt anyag (útmutató és program), valamint a TC-NET megrendelhető a szegedi VORKER Kiszövetkezettől.

Cím: VORKER Kiszövetkezet
Mezeyné Bánfalvi Marianna
6701 SZEGED, Pf.: 711.
Tel.: 06-62-25-435

A VORKER Kiszövetkezet vállalja — megrendelés alapján — a felsorolt témakörökben bemutatók, alap- és továbbképzések tartását, szervezését is.

Az olvasó a szegedi Tarjánvárosi IV. Számú Általános Iskolában fejlesztett 12 pedagógiai program (tanári kézikönyv, szoftver és/vagy beültetési rajzok) egyikét tartja a kezében. A munkában több mint hatvanan vettek részt: általános és középiskolai tanárok, főiskolai, egyetemi oktatók, tudományos intézetek munkatársai).

Az Országos Pedagógiai Intézet Számítástechnikai Programirodája és a szegedi iskola közös vállalkozása összefügg a számítástechnikai alpműveltség körülhatárolására irányuló erőfeszítésekkel. Kétségtelen, hogy a számítástechnikai-informatikai alapismeretek és az alpműveltség kapcsolatának feltárása elsősorban teoretikus jellegű munkát igényel. Az 1970-es években induló tartalmi korszerűsítés tapasztalatai azonban arra hívják fel a figyelmünket, hogy az iskolák nélkül, az iskolai hagyományok és a pedagógia öntörvényeinek mellőzésével nem hozhatók létre olyan dokumentációk, amelyeket a gyakorlat magáénak érez, s amelyekkel a pedagógusok többsége közösséget vállal.

Az alpműveltség témakörében már eddig is több tanulmány látott napvilágot. Viszonylag kevesebb szó esett azonban arról, hogy miként lehet pedagógiaileg szervezett ismeretrendszerre formálni az itthon is és külföldön is már megfogalmazott elképzeléseket (amelyek között meglepően szoros összefüggés lehetséges).

Az elképzelésekről szóló következő rövid ismertetés nem tekinthető állásfoglalásnak abban a kérdésben, hogy mi a jobb: ha a számítástechnika-informatikai alapismeretekhez a diákok az *órateremben*, különféle tantárgyakban jutnak hozzá; *szakköri vagy fakultációs* programok keretében, esetleg *külön tantárgyként* biztosítják ezt számukra.

A fejlesztők munkáját a következő feltételezések irányították:

1. A számítástechnikai-informatikai ismeretek és az ez iránti érdeklődés nehezen tagolható az eddig megszokott módon, vagyis életkor és iskolatípus szerint. Az elkészült anyagoknak tehát egyaránt szolgálniuk kell az általános iskolákat és a középfokú iskolákat.
2. Az iskolák fogadókészsége eltérő (pl. más-más a felszerelés, a tanárok felkészültsége, a szoftverellátottság stb.). Ez aligha teszi lehetővé az eddig megszokott, a mindenki számára kötelező tantervi előírásokat, sémákat. Az igényekhez tehát csak a többféle lehetőség egyidejű bemutatásával kerülhetünk közelebb.
3. A 12 pedagógiai program és ezek járulékaik alapul szolgálhatnak egy majdan országos érvényű informatika tantárgy kidolgozásához. A témák ilyen rendszere a már ma megvalósítható iskolai tevékenységeket tükrözi. Az informatika több más

témájáról az iskolák jórésze számára ma még csupán leírás adható (pl. országos, nemzetközi hálózatok). A későbbiekben elképzelhető, hogy az iskolákban is lehetőség lesz az informatika teljes skálájának bemutatására.

4. A 12 rész között biztosíthatók „átjárások”: a részek egymáshoz illeszthetők és kombinálhatók egymással.
5. A pedagógiai programokban megfogalmazott elképzelések érvényesítése érdekében az elinduláshoz szükséges a megfelelő segédletek elkészítése.
6. A pedagógiai programokhoz készült járulékok (beültetési rajzok, szoftverkészlet) egyik részének tantárgyi alkalmazásokra kell épülnie (pl. matematika, mérések), másik részük viszont az informatikának a mindennapi életben megjelenő gyakorlatát tükrözze (pl. adatkezelés, szövegszerkesztés).

Reméljük, hogy munkánk segíteni fogja a pedagógiai gyakorlatot és ezáltal hozzájárul e téma elméleti kérdéseinek tisztázásához.

Örömmel és tisztelettel fogadjuk munkánk olvasójának, használójának észrevételeit.

Dr. Szűcs Barna

Az elmúlt időszakban tetemes mennyiségű mikroszámítógép került az oktatási intézményekbe és „új családtagként” a háztartásokba. Ezek többségét játékautomatának használják, jobb esetben BASIC programokat futtatnak rajtuk, és nem is sejtik, hogy nemcsak egy „számrágcsálójuk”, hanem egy hatékony, környezetük berendezéseit irányítani képes csodálatos kis elektronikus eszközük is van.

Útmutatónk mindazoknak készült, akik már tudják, hogy a számítógép milyen hatékonyan alkalmazható folyamatirányításra az irányító berendezés részeként, és most szeretnék saját gépüket is összekötni a külvilággal. Szeretnének megismerkedni azokkal az alapismeretekkel, amelyekkel maguk is tervezhetnek egyszerű illesztőegységeket, áramköröket.

A leírt ismeretanyag támogatja a számítástechnika tanórán kívüli oktatását, egyes részei alkalmasak iskolai fakultációs vagy szakköri feldolgozásra, a jól felkészült, a mikroelektronikát hobbiként kedvelő tanulóknak önálló ismeretszerzésre, de sok új információt meríthetnek belőle elektronikai amatőrök is.

Az anyag konkrét házi számítógépek, a COMMODORE 16-os és a C Plus/4-es kapcsán tárgyal olyan általános fogalmakat, elveket, eljárásokat, amelyek minden hasonló kategóriájú mikrogépre megközelítően igazak, illetve az illesztési eljárásoknál, a tervezésnél alkalmazhatók.

Ha kiadványunkat diák olvassa, három dologra hívjuk fel figyelmét:

- Ezzel a témakörrel csak az tud önállóan megbírkózni, akinek van elektronikai alapismerete és épített már egyszerű elektronikus eszközöket. (Tapasztalatunk szerint ezt a szintet 16 éves kor előtt kevesen érik el.)
- A leírt eszközök olcsó áramköri elemeket tartalmaznak, azok zsebpénzből is megvásárolhatók. Viszont az építéshez kell egy pontos és nagy fordulatszámú precíziós fűrő, valamint egy hőfokszabályozós Weller páka, amik igencsak drága játékszerek.
- Azt is feltételezzük, hogy aki egyedül belevág a következőkben leírt ismeretek elsajátításába, rendelkezik C 16-os (C Plus/4-es) számítógéppel, már dolgozott is vele és lehetőségeit nagyjából ismeri.

Természetesen az anyag feldolgozásához leírt feltételek közben is megteremthetők, az alapismeretek megszerezhetők.

Ha az elektronikát hobbiként művelő amatőr vette kezébe kiadványunkat, mire vár? Kezdjen azonnal az áttanulmányozásához! Az anyag a számunkra legjobban bevált szakirodalomból tartalmaz hivatkozásokat. Reméljük ez és jószándékunk, mellyel a számítástechnikai hardverrel szeretnénk megismertetni, az egész anyag feldolgozása során segíteni fogja.

Az útmutatóban szereplő valamennyi kapcsolást már több példányban megépítettük, azok garantáltan működnek.

A leírások és az eszközök saját termékeink. Ezennel **felhatalmazunk mindenkit** ezek további felhasználására, hasznosítására, ha azt oktatási céllal teszik.

Ugyanakkor felhívjuk a figyelmet arra, hogy **a kapcsolások megépítésénél az elektromos balesetek megelőzésének szabályait be kell tartani**, az eszközöket elektromosan biztonságos kivitelben kell elkészíteni (függetlenül attól, hogy a leírt elektronikák mindegyike csak 5 V-ról működik)!

A „Számítógép és környezete” témakör feldolgozásához, amely tulajdonképpen interfészkészítési és I/O perifériaillesztési feladatok megoldásával foglalkozik, sok sikert kíván a szerző:

Borbola István

1. AMIT A KAPCSOLATTEREMTÉS ELŐTT TUDNI KELL

1.1. Kódok az adatsínen

A C 16-os házi számítógép — a többi ma használatos mikroszámítógép társához hasonlóan — a saját működését *tárolt program* segítségével irányító kódfeldolgozó automata.

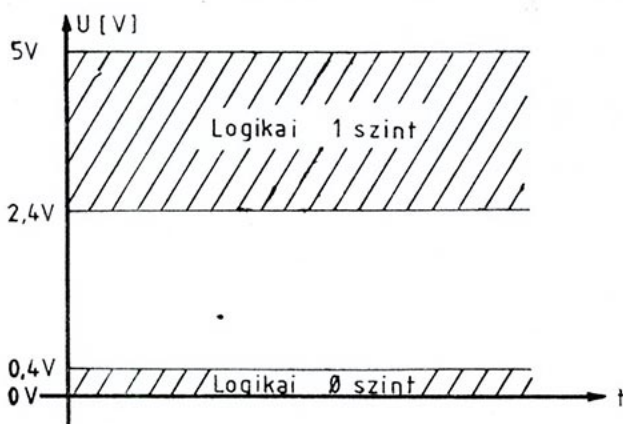
A *kód* tetszőleges jelkészlet elemeiből kiválasztott, végszámú jelből összeállított ismétléses variáció, amelyet egy adott halmaz elemeihez hozzárendelünk.

Például kód a gépjárművek rendszáma. Ennek a *kódszisztemnek* a *jelkészletét* az abc betűi és a számok alakú értékei adják. Ugyanaz a jel egy rendszámban többször is előfordulhat, s a gépjárművek kölcsönösen egyértelműen hozzá vannak rendelve rendszámaikhoz. Kódokra láthatunk példákat az 1.1. ábrán:

Rendszám:	Telefonszám.	Személyi szám:	Gyártmányok típusjele:
IX-22-16	06-62-51-449	1 481231 5999	DS 1521-212-4

1.1. ábra Néhány közismert kód fajta

A C 16-os házi számítógép elektronikája a gép *információs jelvezetékein* csak két stabil működési állapotot tud létrehozni (1.2. ábra). A jelvezetéken vagy 2,4 V fölött van a feszültség (ezt az állapotot 1-gyel jelöljük), vagy 0,4 V alatt (ezt az állapotot



1.2. ábra. Feszültség szintek a C 16-os számítógép belső információs jelvezetékein

\emptyset -val jelöljük). Ezért az információs jelvezetékeken megjelenő kódok \emptyset , vagy 1 feszültségértékű állapotok ismétléses variációi lehetnek. Az egyes áramkörök szempontjából a kivezetéseknek lehet egy harmadik, úgynevezett **nagyimpedanciás** állapota is, de erről a későbbiekben lesz szó.

A C 16-os közvetlen programozásához [1.], [2.], környezetével való összekötéséhez ennek alapján kételemű, ún. bináris jelkészlettel rendelkező kódrendszert kell használni [3.].

A *bináris kód* nullák és egyesek ismétléses variációja, tulajdonképpen egy kettes számrendszerbeli szám. Példa bináris kódszóra az 10110111 jelsorozat. Ha a bináris kódszó hossza egy (pl.: \emptyset) akkor azzal csak egy állítás (logikai ítélet) igazságértékét [4.], [36.] tudjuk megadni. Úgy is fogalmazhatunk, hogy egy olyan egyelemű bináris kód, amelyik egy logikai ítélet igazságértékét adja, egységnyi ún. 1 bit-nyi információt tud közvetíteni. Az információ egysége az *1 BIT*. (A BIT az angol **B**asic **I**ndis-soluble information uni**T** kifejezésből (felbonthatatlan információ alapegység) származó betűszó [6.]. Vannak, akik e rövidítést a **B**inary uni**T** (kettes egység) kifejezésből vezetik le.)

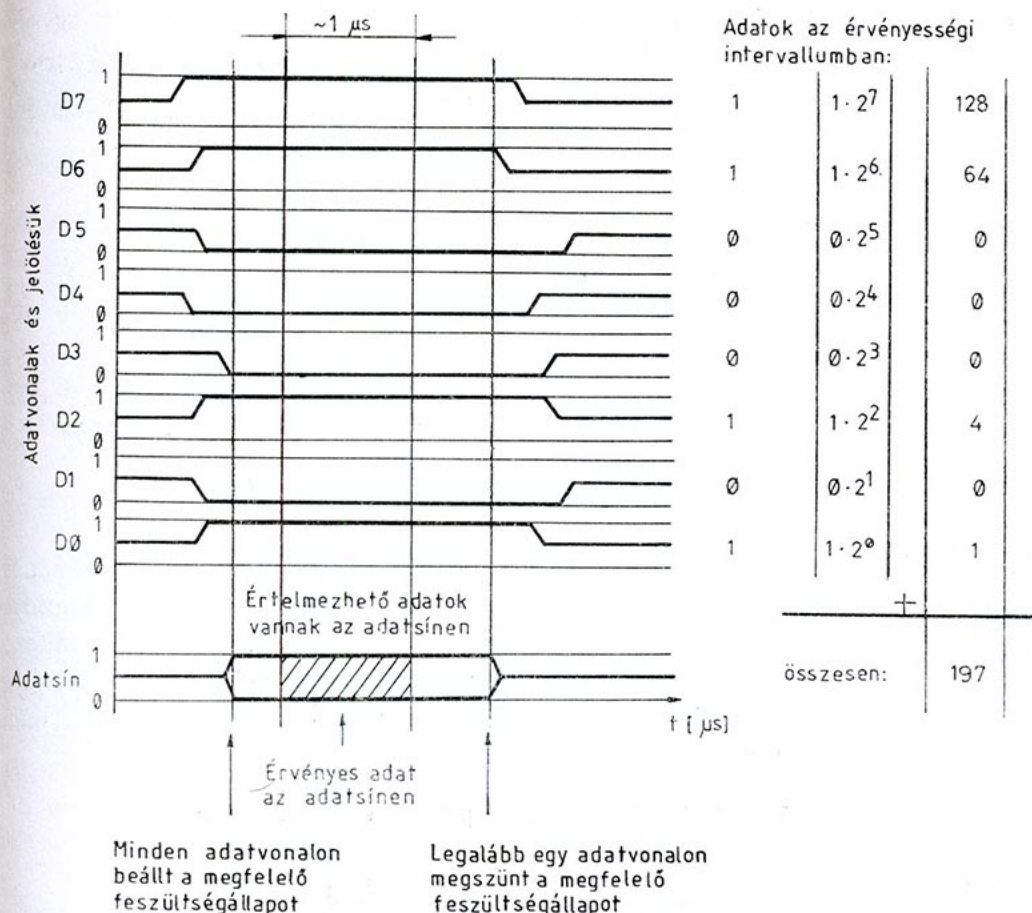
A C 16-os házi számítógépekben az adatok kódolására bináris jelkészletből (\emptyset , 1) álló 8 elemű (8 bit hosszúságú) kódszót használnak. Ezt az együtt értelmezett 8 bit hosszúságú kódszót *1 byte-nak* (bájtnak) nevezzük. A kételemű jelkészletből 8 elemet $2^8=256$ féle módon lehet (ismétlést is megengedve) egymás mellé tenni. Egy bájtal tehát 256 féle információt kódolhatunk közvetlenül. [7.], [8.]

A C 16-os számítógépet vezérlő központi egységnek, a MOS 8501 típusjelű mikroprocesszornak 8 darab adatvezetéke van, ezért lehet 1 bájtnál hosszúságú a kódszó, s ezért lehet az adatvezetékeken egy adott időpontban párhuzamosan és egyszerre értelmezni a rajtuk lévő információt. A 8 adatvezeték együtt az *adatsín* alkotja, amelynek vezetékei külön-külön áramkörökhöz tartoznak elektronikusan, de az információ áramlása szempontjából együtt kezeljük azokat. (Lásd az 1.7. ábrát is!)

Az adatsínen egy adott időpontban lévő kód **bináris szám**, amelyet nagyon könnyű tizenhatos (**hexadecimális**) számrendszerbeli számként kezelni, vagy tízes számrendszerbeli (**decimális**) számmá átírni. [4.], [8.]. A későbbiekhez elképzelhetetlen, hogy a számokat egyik számrendszerből a másikba ne tudjuk átírni (konvertálni), ezért ezt nagyon sokat kell gyakorolni [9].

Az 1.3. ábrán az adatsínen kódolt információ átalakítását és értelmezését láthatjuk a szokásos jelölésekkel. Felhívjuk a figyelmet arra, hogy — a számítógép sajátosságos működése miatt, ami majd később kerül ismertetésre — az adatsínen az adatokat mindig csak egy adott időpontban tekinti a számítógép érvényesnek! A C 16-os házi számítógép esetén az *adatok érvényessége* az adatsínen átlagosan 1 μ s ideig áll fenn!

A C 16-os számítógép az információkat az **interpreter** [2.] segítségével a mindennapi életben használhoz közeli módon kéri, illetve adja [3.], [1.].



1.3. ábra. Kódolt információ az adatsínen, a bináris kódszó decimális megfelelője

Például a számok esetében az interpreter tízes számrendszerben megadott számokat alakít át (kódol) a működéshez kettes számrendszerűvé, majd a feldolgozott kódot (ha az számhoz van rendelve) visszalakítja (dekódolja) ismét „érthetővé”, tízes számrendszerű számmá [4.], [9.]. Arról, hogy a gép kettes számrendszerben dolgozik, könnyen meggyőződhetünk az alábbi kis program segítségével:

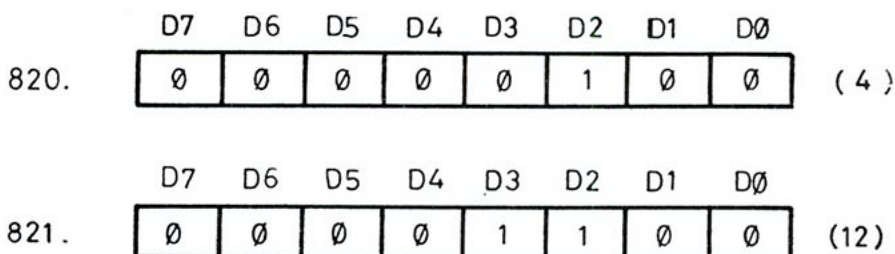
```

10 REM ** 1. mintaprogram **
11 :
12 :
15 POKE 820,4:POKE 821,12
20 A=PEEK(820) OR PEEK(821)
30 PRINT "a=";A
40 B=PEEK(820) AND PEEK(821)
50 PRINT "b=";B
60 END

```

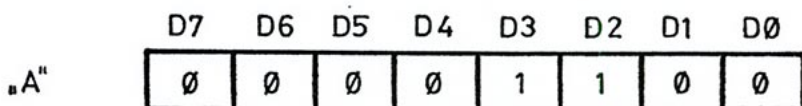
A program magyarázata:

10. sor a memória 820. sorszámú (című) rekeszébe 4-et, a 821. sorszámú rekeszébe pedig 12-t írunk. Ezeket a gép a memóriában 8 elemű bináris kódszóként helyezi el. (1.4. ábra.)



1.4. ábra. Rekeszekben tárolt bináris kódszavak

20. sor logikai VAGY művelettel bitenként összehasonlítja a két rekesz tartalmát és az eredményt beírja az A változóba. Mivel a VAGY művelet eredménye akkor és csak akkor 0, ha mindkét logikai változó 0, ezért az A változó rekeszében minden olyan helyiértéken 1-es fog szerepelni, ahol a 820. és a 821. rekeszben legalább az egyik helyen 1-es volt. Így a VAGY művelet után az A változó rekesze:



1.5. ábra. Az „A” rekesz tartalma a 820. és a 821. rekesz VAGY műveletes összehasonlítása után

30. sor megtörténik az „A” rekesz tartalmának decimálissá alakítása $1 \cdot 2^3 + 1 \cdot 2^2 = 12$ és $A=12$ formátumú kiírása.

A fentiek alapján — a logikai ÉS művelet igazságtáblájának ismeretében [4.], [10.] — mindenki magától kitalálhatja, hogy a $B=4$ értéket miért írja ki a gép!

1.2. Játsszunk a kódokkal

A későbbiekben sokszor lesz szükségünk arra, hogy egy rekesz tartalmát úgy változtassuk meg, hogy csak egy helyiértéken, vagy bizonyos helyiértékeken történjen változás, s közben a többi helyiérték bitjei változatlanok maradjanak. Aki az előzőek alapján a rekeszek logikai művelettel való bitenkénti összehasonlítását megértette és gyakorolta, annak nem okoz majd gondot a következő kis program működésének megértése sem.

A C 16-os számítógép 65286. rekeszében (ez a címe) a 2^4 helyiértékre 0-t írva a képernyő kikapcsolódik, és az éppen futó program futásideje csökken (a munkavégzés felgyorsul). Ezután a 2^4 helyiértékre 1-et írva a képernyőszerkesztés ismét folyamatos lesz.

Próbáljuk ki először az alábbi programot:

```
10 REM ** 2. mintaprogram **
11 :
12 :
15 TI$="000000"
20 FOR I=1 TO 10000:A=I:NEXT I
30 PRINT TI$:PRINT TI
40 END
```

A program ≈ 32 mp-ig fut. Majd futtassuk a következő programot:

```
10 REM ** 3. mintaprogram **
11 :
12 :
13 TI$="000000"
15 POKE 65286,PEEK(65286) AND (255-16)
20 FOR I=1 TO 10000:A=I:NEXT I
25 POKE 65286,PEEK(65286) OR 16
30 PRINT TI$:PRINT TI
40 END
```

A program ≈ 21 mp-ig fut. Ez a jelentős időmegtakarítás annak köszönhető, hogy egy alkalmas bájton belül a 2^4 helyiértékű bit értékét megváltoztattuk.

15. sor a 65286. rekeszbe beírtuk a rekesz eredeti tartalma és egy 239-et tartalmazó átmeneti rekesz bitenkénti „ÉS” kapcsolatának eredményét. Az ÉS művelet eredménye akkor és csak akkor 1, ha mindkét logikai változó értéke 1.

25. sor elemzésével hasonlóan érthető meg, hogyan kerül a 65286. rekesz 2^4 helyiértékű bitjébe ismét 1, visszakapcsolva a képernyőt.

Nézzük meg mi történik, ha a következő parancsok valamelyikét adjuk számítógépünknek:

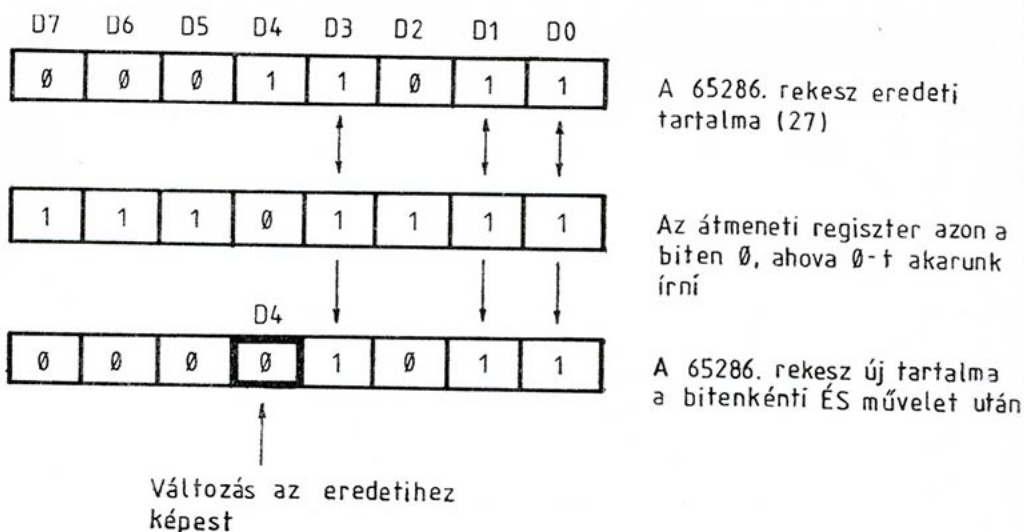
```
POKE 65286,PEEK(65286) AND 247 [RETURN]
```

vagy

```
POKE 65287,PEEK(65287) AND 247 [RETURN]
```

Az adott rekeszek melyik bitjét módosítottuk, és hogyan állítható vissza az eredeti állapot?

$(255-247=8; 8=2^3$, tehát a 4. bitet, azaz a 4. helyiértékűt módosítottuk. Az eredeti állapot pl. a POKE 65287, PEEK(65287) OR 8 paranccsal állítható vissza.)



16. ábra. Rekeszen belüli bitmódosítás (képernyő kikapcsolásához)

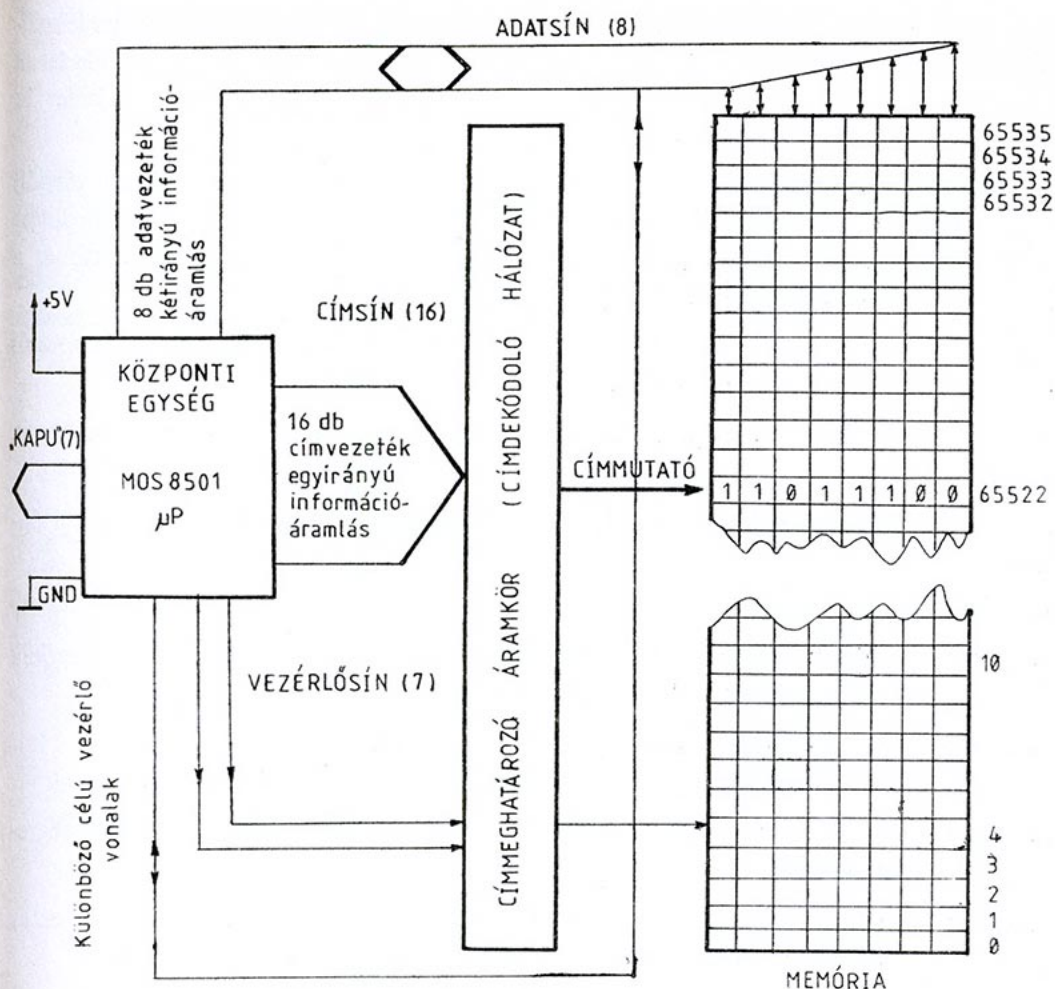
1.3. A mikroszámítógép működésének lényege

A C 16-os házi számítógép működése bonyolult. Ahhoz, hogy egyszerű eszközöket hozzá tudjunk kapcsolni, első közelítésben elegendő egy 8 bites kódszóval működő számítógép *működési modelljét* megérteni.

A számítógép működésének lényege a *memória* és a *központi egység* összehangolt együttműködése.

A C 16-os memóriáját egy hatalmas polcsornak lehet elképzelni, amelynek minden sorában 8 fiók van. Minden polcsornak van egy sorszáma, ez a *címe*. A polcsoron lévő fiókok tartalmát együtt kezeljük. Egy-egy ilyen polcsort egy *memóriarekesznek* nevezünk. A memóriarekeszekben tehát 8 bitnyi, azaz 1 bajtnyi információ tárolódik *8 elemű bináris kódszó* formájában. A tárrekeszekben elhelyezett *kódok* tartalmazzák mindazt az információt, amelyre a számítógép működése során a központi egységnek szüksége van. (1.7. ábra.) Ezek lehetnek a központi egység számára utasítások, címek adatai, egyéb feldolgozandó adatok. [3.], [11.]

A központi egység a kódolt információk áramoltatására, mint már említettük, 8 adatvezetékkel használ. Az *adatsín kétirányú adatmozgást* tesz lehetővé. Meghetnek adatok a processzortól a memóriába, vagy a memóriából a processzorba. A memóriából beolvasott kód alapján a központi egység előállítja a következő feldolgozandó kódnak a címét egy 16 bites, a processzoron belüli regiszter: a „*programszámláló*” (PC) segítségével. Az adatvonalakhoz hasonlóan a *címvonalak* is *két állapotúak* lehetnek (lásd 1.2. ábrát), ezért a 16 címvonalon $2^{16} = 65536$ féle ismétléses variációt lehet előállítani, azaz 65536 darab memóriarekeszt lehet kiválasztani. Erről egy



1.7. ábra. Egy 8 bites mikroszámítógép modell

címdekódoló hálózat gondoskodik. A 16 cím vonal mindegyike külön-külön áramkörhöz tartozik, de információs szempontból együtt kezeljük azokat, s *címsínnek* nevezük. A címsínen a központi egységtől a memória felé egyirányú az információ áramlása. A mikroprocesszor (központi egység), a memória és a többi integrált áramkör időzítve és összehangolva működik együtt. Ezt *vezérlő vonalaival* a mikroprocesszor állandóan ellenőrzi, illetve befolyásolja. Egyes vezérlő vonalakon az áramköröktől a processzor irányába, másokon a processzortól a berendezés irányába halad a jel. A C 16-os házi számítógépnek 7 ilyen vezérlő vonala van, amelyeket információs szempontból együtt kezelve *vezérlősínnek* nevezünk.

A nagybonyolultságú elektronika precíz együttműködését egy *órajel*hez szinkronizálva végzi a processzor. Minden működési állapotot egy előző működési állapot határoz meg, és minden óraütemhez egy új állapot tartozik. A számítógép **programozható szinkron sorrendi hálózat** [3.], [5.].

A C 16-os házi számítógép időzítései egy speciális IC (a TED chip) különleges lehetőségei miatt nagyon bonyolultak. Első közelítésben — egyszerű illesztési feladatok megoldásához — most is elegendő, ha modellünkön vizsgálódva (1.7. ábra) ismerjük meg az alapfogalmakat. [3.], [11.]

A központi egység minden tárművelete (adat írása a tárba, vagy olvasás a tárból) egy, a gép szempontjából lényeges alapidőtartamot igényel. Ezt a tárhozfordulást *gépi ciklusnak*, időtartamát pedig a gépi ciklus idejének nevezzük. Ez a tár és a processzor működését tekintve alapidőegység!

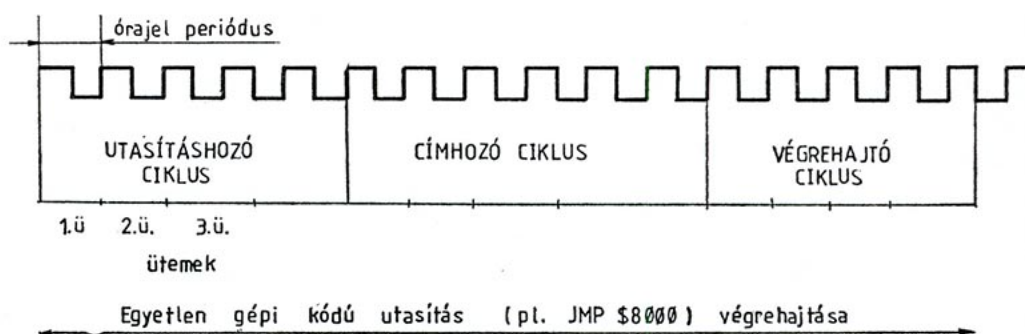
A C 16-os időzítésének bonyolultságát éppen az okozza, hogy a TED chip állapotától függően változik a gépi ciklusidő. [3.]

A számítógép az egyes gépi ciklusokat különböző *ütemekből* összeállítva hajtja végre.

Pl. egy utasításhozó gépi ciklus (FETCH) végrehajtása a következő módon történhet:

1. Gépi ciklus (fetch)

1. ütem: — Az utasításszámláló segítségével a memória egy rekesze megcímeződik
— Az utasításszámláló értéke 1-gyel megnő
2. ütem: — A memóriából bejön az utasítás kódja.
— A processzor azonosítja, hogy milyen utasításról van szó.
3. ütem: — A processzor előkészíti az időzítőt és az időzítőjeleket a későbbi végrehajtáshoz.
4. ütem: — Ebben az ütemben történne a végrehajtás és a címképzés, ezért ez az ütem az utasításhozó ciklusban kihasználatlan.

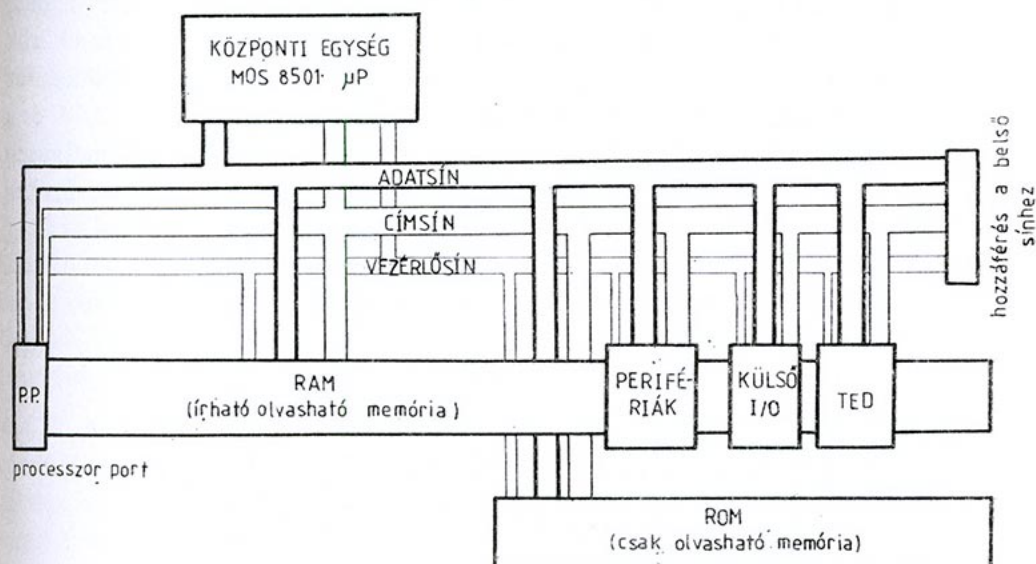


1.8. ábra. A bekapcsolás utáni első utasítás végrehajtása

Programozói szempontból az a lényeges, hogy gépünk mennyi idő alatt hajt végre egy gépi kódú utasítást [12.], hány órajel időtartamig; hardver szempontból pedig az a fontos, milyen gyorsak az órajelek, hogyan lehet az időzítéseket biztosítani. [3.]

1.4. Kapuk a külvilágra

Akármilyen szépen is dolgozik együtt a központi egység és a memória, ha a kódfeldolgozás eredményéről nem tudhatnánk meg semmit, az egész folyamat közömbös lenne számunkra. A számítógépek a kezelőt kiszolgáló perifériák (monitor, billentyűzet) és feladatcentrikus perifériák (diszk, kazettás tár, nyomtató, manipulátor kar stb.) segítségével tartják a kapcsolatot környezetükkel. A külvilághoz való csatlakoztatás legegyszerűbb áramkörei a *portok* vagy magyarul *kapuk*. Ezek az *információs kapuk* a mikroszámítógép és a külső eszközök között.



1.9. ábra.. A TED rendszer vázlatja

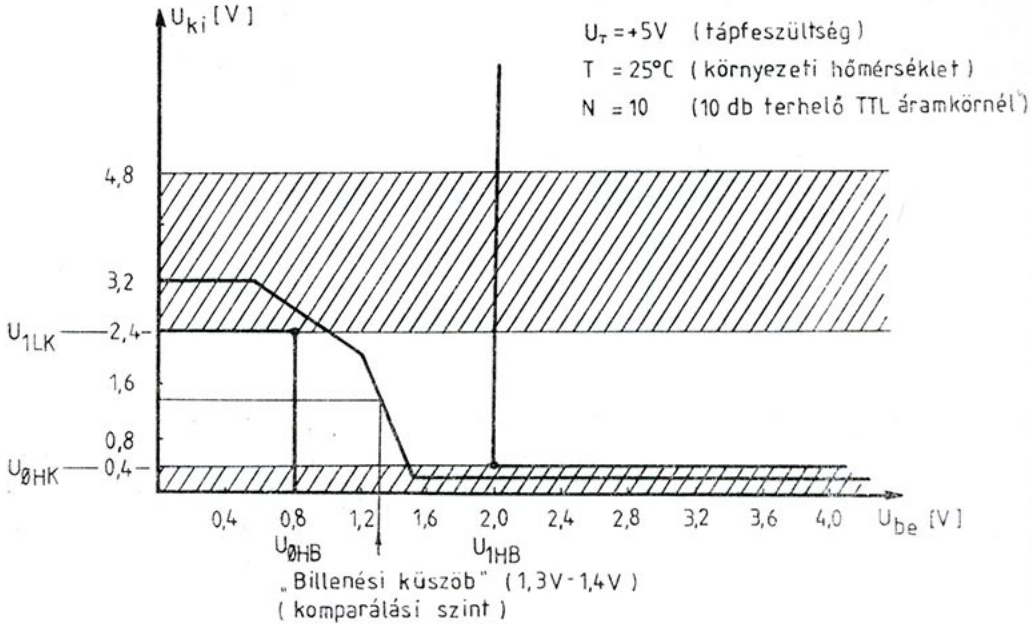
A porták alapvető sajátossága, hogy az adatsínen lévő információból **mintát vesznek** az adat érvényességének pillanatában (lásd 1.3. ábrát) és azt rögzítve mindaddig tárolják, amíg újabb mintavételre nem kényszerítjük őket. Vannak csak bemeneti (*input*) portok, vagy csak kimeneti (*output*) portok, de olyanok is léteznek, amelyek egy parancstól függően kimenetek és bemenetek is lehetnek.

A C 16-os, C Plus 4-es számítógépek alapvetően azonos belső szerkezetűek, azonos rendszertechnikai megoldásúak. A rendszer működésének megértése kezdő hardveres számára reménytelen vállalkozás, de egyszerű illesztési feladatok megoldásához ez szükségtelen is. Elegendő, ha tudjuk, hogy a C 16-os és a C Plusz 4-es házi számítógépek — leegyszerűsítve a tényleges működést — *a címezhető memóriaterületbe ágyazottan, virtuális tárrekeszként* kezelik a perifériákat, a portákat és az egész I/O (input/output; bemeneti/kimeneti) terület [3.], [2.]. A rendszer egy sajátos berendezésorientált áramkörrel (BOÁK), a TED chipről kapta a nevét (1.9. ábra). A virtuális tárrekeszként kezelt I/O (input/output) azt jelenti, hogy a perifériák vagy a portok külön erre a célra fenntartott tárcímeken vannak és úgy kezelhetők, mintha

tárrekeszek lennének. Ugyanúgy címezhetők, mint bármelyik memóriarekesz, és az adatokat az adatsínről veszik, vagy az adatsínre adják. Az *adatkivitel* (írás) és az *adatbehozatal* (olvasás) ugyanazok a vezérlőjelek irányítják vagy mintavételezik, mint a memória esetében [2.], [3.].

A C 16, C Plus/4 portá **inormál TTL** kompatibilisek. A továbbiakban a felhasználó számára — a számítógép „illesztés” utáni életben maradása szempontjából — néhány fontos ismeretet kell pontosítani, bár feltételezzük, hogy az alapvető elektronikai fogalmakat ismeri.

A **TTL rövidítés** az angol **Transistor-Transistor Logic** (tranzisztor-tranzisztor-logika) kifejezésből alkotott betűszó, az áramkör belső felépítésére utal. A TTL áramkörök az egész világra kiterjedően **szabványosítottak**. Ezekben a logikai áramkörökben az **alacsony szint** (L=Low) 0V és 0,4 V között, a **magas szint** (H=High) 2,4 V és a **szabványos tápfeszültség** ($5V \pm 5\%$) között van. A TTL áramkörök biztonságosan 5 V-os **stabilizált tápegységről** [13.] működtethetők! A TTL áramkörök átviteli



1.10. ábra. TTL áramkör átviteli karakterisztikája (NOT kapu)

karakterisztikája (a bemenet függvényében hogyan változik meg a kimenet) az 1.10. ábrán látható.

Az egész karakterisztika és ezzel együtt a **névleges logikai feszültségszintek** is függenek a hőmérséklettől, a tápfeszültségtől és a terheléstől! A karakterisztikát vizsgálva megállapíthatjuk, hogy a kimeneti alacsony (L) szint nem haladhatja meg az $U_{0H} = 0,4V$ értéket, ha a bemeneti magas szint nem csökken $U_{1H} = 2V$ alá és a

kimenetet legfeljebb 10 normál TTL kapu bemenete terheli. A kimeneti magas (H) állapot nem csökkenhet $U_{1L}=2,4\text{ V}$ alá, ha a bemenő feszültség nem haladja meg az $U_{oH}=0,8\text{ V}$ értéket, és a kimenetet legfeljebb 10 normál TTL kapu bemenete terheli.

Sokan szeretnének számítógépükkel például villanymotorokat működtetni. A motorok **kommutátorszikrái** vagy a **kikapcsolási indukció** komoly *zavarjeleket* eredményezhet a jelvezetéken. A TTL áramkörök statikus zavarvédeltsége a legrosszabb esetet feltételezve az áramkör bemenetének L állapotában:

$$U_{ZHL} = U_{oHB} - U_{oHK} = 0,8\text{V} - 0,4\text{V} = 0,4\text{V}$$

A *zavarhatár* az áramkör bemenetének H állapotában:

$$U_{ZHH} = U_{1LK} - U_{1HB} = 2,4\text{V} - 2,0\text{V} = 0,4\text{V}.$$

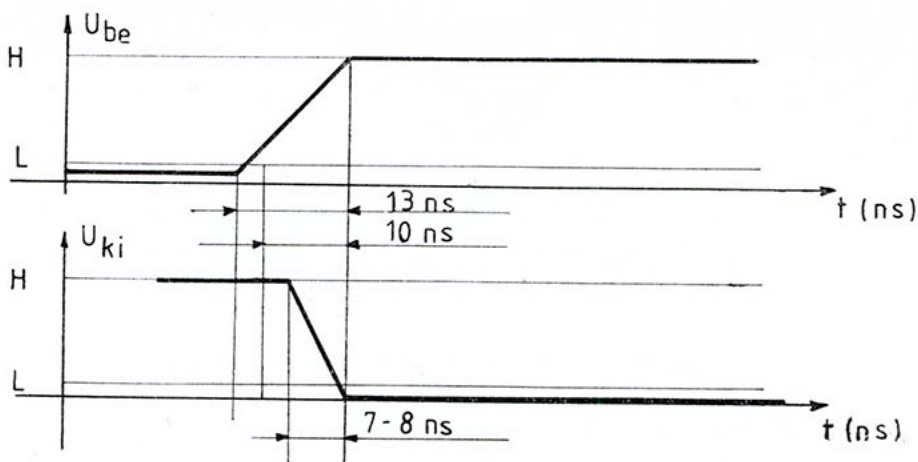
A lehető legrosszabb esetben (amit rossz méretezés vagy nagy gyártási szórás is előidézhet) már $U_z > 0,4\text{ V}$ zavarfeszültség elegendő a hibás működéshez. A gyakorlatban a *tipikus zavarvédeltség* ennél jóval nagyobb a megengedett terhelésnél, ui.: $U_{1LK} \approx 3,3\text{ V}$, $U_{oHK} \approx 0,2\text{ V}$ és a **komparálás** 1,3—1,4 V körül történik. Így a tipikus zavarvédeltség L állapotban kb. 1 V, H állapotban pedig 2 V. [14.]

A TTL áramkörök tipikus bementi árama kb. 1 mA L szinten, de ez az érték a legrosszabb esetben sem haladja meg az 1,6 mA-t. Ezt szokták szabványos *TTL egységterhelésnek* nevezni. Logikai H szinten a bemenet kb. 5 μA —40 μA -rel terheli a meghajtó áramkört. [14.]

A TTL áramkörök bemenetének megvalósításából következik (az elvi fejtegetéseket itt mellőzzük) [14.], [15.], hogy az áramkörök bemenetére nem kapcsolhatunk 1—2 V-nál nagyobb negatív feszültséget, a bemenő feszültség egyetlen bemeneten sem lehet nagyobb, mint +5,5 V, és két bemenet között a potenciálkülönbség szintén nem lehet +5,5 V-nál nagyobb!

A TTL áramkörök kimenetei kb. 16 mA áramot képesek „nyelni” kimeneti L szinten $R_{kiL} \approx 10\text{ Ohm}$ kimeneti ellenállás mellett. H állapotban a kimeneti ellenállás $\approx 70\text{ Ohm}$ és 150 Ohm között változhat a terheléstől függően, és az áramkör ≈ 10 —20 darab 40 μA terhelésű bemenetet tud 1-be vezérelni. Az SN74XX sorozat specifikált rövidzárási árama 15 mA és 55 mA között lehet, ha a kimenet H szinten van.

Nagyon lényeges, hogy a TTL áramkörök a bemenet megváltozása esetén a kimeneten nem veszik fel azonnal a várt logikai értéket. A jeleknek van egy $t_{pL \rightarrow H} = 11 \dots 13\text{ ns}$ *felfutási* és egy $t_{pH \rightarrow L} = 7 \dots 8\text{ ns}$ *lefutási késleltetése*. Így a TTL körökben az **átlagos jelterjedési idő**, ha a vezérelt bemenetek száma maximum 10, megközelítően 10 ns (1.11. ábra).



1.11. ábra. TTL áramkörök jelterjedési idői

A kimeneten jelentkező nagyobb terhelő kapacitás a le- és felfutási időket jelentősen megnövelheti, így a jelterjedés lassul, a jelterjedési idő megnő.

A TTL áramkörök szabadon hagyott bemenetei a kapuáramkörök működését lassítják, ezért ezeket valamilyen stabil logikai feszültség szintre kell kötni, ha nem használjuk azokat!

Végül nagyon lényeges kiemelni, hogy a TTL logikai áramkörök tápáramfelvétele a kimenet L és H állapotában általában eltérő. Az $L \rightarrow H$ átmenetnél jelentős tápáram-ingadozás lép fel, így a tápvezeték soros impedanciáján zavaró tápfeszültség-lökések keletkeznek. Ezért a tápfeszültségvezeték-rendszert nagyon kicsi impedanciájúra kell kialakítani és a tápvezetéseket az IC tokok közvetlen közelében nagyfrekvenciásan hidegíteni kell (pl. 100 nF -os kondenzátorral). [14.]

Az átkapcsolási áramcsúcsok miatt az áramkörök átlagos áramfelvétele és így a disszipáció is függ a működési frekvenciától, így a frekvencia növekedésével az áramfelvétel és a disszipáció is nő. [14.]

1.5. Szemlélődünk a kapcsolási rajzon:

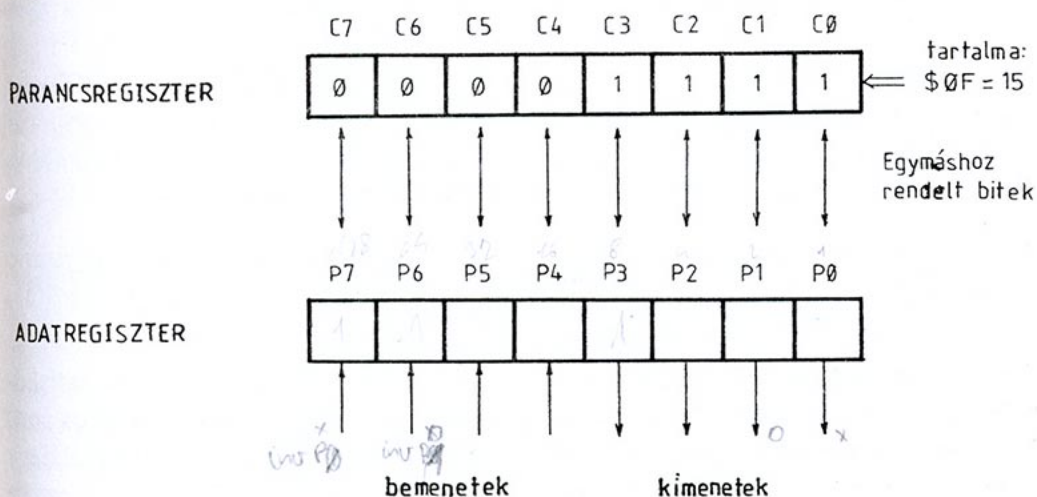
A C 16-os, C Plus/4-es házi számítógépekhez megjelent irodalmak [1.], [2.] mellékleteként a gépek kapcsolási rajzát is közlik. Vizsgáljuk meg a processzort is tartalmazó rajzot! Látható, hogy a processzornak van egy 8 bitesként kezelhető portja, amelyen a P5 helyiértékű bitet a lábak behatárolt száma (40 lábú IC) miatt nem tudták kivezetni. A bitek megjelenése az egyes lábakon:

LÁB SZÁMOZÁS	BIT	FELHASZNÁLÁS	MELYIK CSATLAKOZÓN JÖN KI A SZÁMÍTÓGÉPBŐL?
30.	P0	DISK egység DATA kimenete, a processzor port 0. bitje	Soros port 5. láb
29.	P1	DISK egység CLK kimenete, a kazettás egység írójele (WRT), a processzor port 1. bitje	Soros port 4. láb Cassette port 5. láb
28.	P2	DISK egység ATN kimenete, a processzor port 2. bitje	Soros port 3. láb User port (Plus/4) 9. láb
27.	P3	Kazettás egység MOTOR bekapcsolás kimenet	Cassette port 3. láb (tranzisztorral meghajtott)
26.	P4	Kazettás egység OLVASÓ (RD) bemenete	Cassette port 4. láb
25.	P6	DISK egység CLK bemenete, a processzor port 6. bitje	Soros (serial) port 4. láb
24.	P7	DISK egység DATA bemenete, C-16-nál a kazettás egység billentyűzet lenyomás érzékelés bemenőjele, a processzor port 7. bitje	Soros port 5. láb Cassette port 6. láb

1.12. táblázat. A processzor port kivezetései

Vizsgálódásunk nyomán „megtaláltuk” a C 16, C Plus/4 számítógép egyik kívülágra nyitott kapuját, a *processzor portot*.

A processzor portnak két, memóriába ágyazott regisztere van, melyek memóriarekeszként kezelhetők. A 0. címen a *parancsregiszter*, az 1. címen az *adatregiszter* található. A parancsregiszter itt tulajdonképpen adatrányt meghatározó regiszter. A port minden olyan kivezetésre kimenetként fog viselkedni, amelyikhez tartozó parancsregiszter-beli biten egyes van. Nézzünk egy példát:



1.13. ábra. A processzor port vezérlése

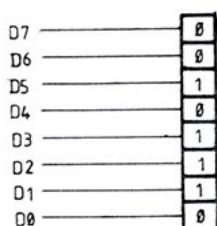
A C0. biten 1 van, ezért a processzorport legkisebb helyiértékű vezetéke kimenet lesz. (Mint egy normál TTL kapu kimenete.) Pl. ha az adatregiszterbe POKE1,1 utasítással 1-et írunk, akkor az 1-nek megfelelő feszültség szint meg fog jelenni a P0. biten. Természetesen a kimenetet nem szabad a 0 potenciálú földdel (GND) rövidre zárni, mert az 50 mA körüli rövidzárási áram azonnal tönkreteszi a port meghajtó áramkörét! A C5. biten 0 van a parancsregiszterben, ezért az adatregiszter P5. bitje bemenetként viselkedik. (Mint egy normál TTL kapu bemenete.) Ha most PRINT PEEK(1) utasítással megnéznénk az adatregiszter tartalmát, azt tapasztalnánk, hogy a P5. bitre kívülről adott logikai értéket beolvasta a számítógép. A processzorportra kötött meghajtó IC és az összetett felhasználás miatt egyelőre a port parancsregiszterének írásával ne foglalkozzunk, mert ha rosszul állítjuk be az adatregiszter bitjeinek adatrányát, az egész processzor pillanatok alatt tönkremehet! A rendszer bekapcsolás után saját maga megfelelően beállítja a processzor portot. Fogadjuk el ezt és dolgozzunk egyelőre a kapott lehetőséggel!

Nézzük meg a PRINT PEEK(0) utasítással, hogyan állítja be kezdetben (hogyan *inicializálja*) az interpreter a portot! Láthatjuk, hogy a parancsregiszterben 15 (\$0F) van, tehát a beállítás az 1.13. ábrának megfelelő. [2.], [1.]

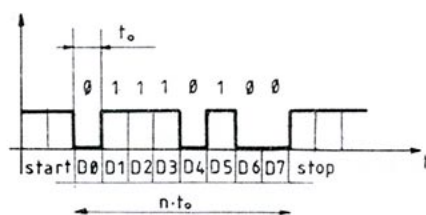
A \$ jel tizenhatos számrendszerbeli számot jelöl [4.], [9.]

Ha az információt hordozó kódszó egyszerre jelenik meg az adatsínen és azt egy adott időpontban együtt értelmezzük, akkor *párhuzamos információátvitelről* beszélünk. Amikor a kódszó egyes bitjeinek feszültségértékei sorban, egymás után jelennek meg egy vezetéken, akkor *soros az információ* adása. (1.14. ábra.) [3.], [16.]

Párhuzamos kódátvitel



Soros kódátvitel



1.14. ábra. Párhuzamos és soros kódátvitel

A párhuzamos átvitel sokkal gyorsabb, mint a soros, mert a jelvezetékekről egyszerre leolvasható az információ. Soros átvitelnél az egy bit átvitelére szánt idő többszöröse kell és még a **szinkronizáló start és stop bitek** is nyújtják az átviteli időt.

Annak ellenére, hogy a C 16, C Plus/4-es házi számítógépek processzor port csatlakozójának egyikére a „SERIAL” (soros) megjelölés van felírva, nem valódi soros port. Ugyanis a párhuzamos processzor port P0 vezetékén a rendszerszoftver szimulál egy soros adatmegjelenítést (DATA) a hozzá tartozó szinkronizáló jelekkel együtt (CLK, ATN). [17.], [18.]

A processzor port működését az eddigiek alapján könnyen ki is próbálhatjuk, hiszen a kazettás egység villanymotorját ennek a portnak a P3-as bitje vezérli, illetve C 16-nál a gép ennek a portnak a segítségével figyeli a kazettás egység billentyűzetének lenyomását.

Normális működésnél a *rendszer szoftver* vezérli a kazettás egységet. Ha az adatmagnó kikapcsolt állapotában PRINT PEEK(1) utasítással lekérdezzük a processzor portot, akkor a *kezdeti beállítási értéket (default érték)*, a 200-at (\$C8) kapjuk. Nyomjuk le a [PLAY] billentyűt, majd ismét kérdezzük le a PRINT PEEK(1) utasítással az adatregiszter tartalmát. Most 192-t (\$C0) kapunk. Miért? Azért, mert a rendszer a *CST SENSE* jel segítségével érzékelte a billentyű lenyomását és a P3 bit (ez a $2^3=8$ helyiérték) alacsonyra állításával bekapcsolta a magnó villanymotorját. A [REWIND] és az [F. WIND] billentyűk lenyomásakor is ez történik. Nyomjuk le a [RECORD] és a [PLAY] billentyűket együtt! Lekérdezve a processzorport adatregiszterének tartalmát most 208-at (\$D0) kapunk.

Ez azért van így, mert e két billentyű együttes lenyomásakor a szalagra írás történik, így a P4 biten ($2^4=16$) a $\overline{RD}=1$ miatt *H* logikai szint lesz, és a $192+16=208$ jelenik meg az adatregiszter kiolvasásakor.

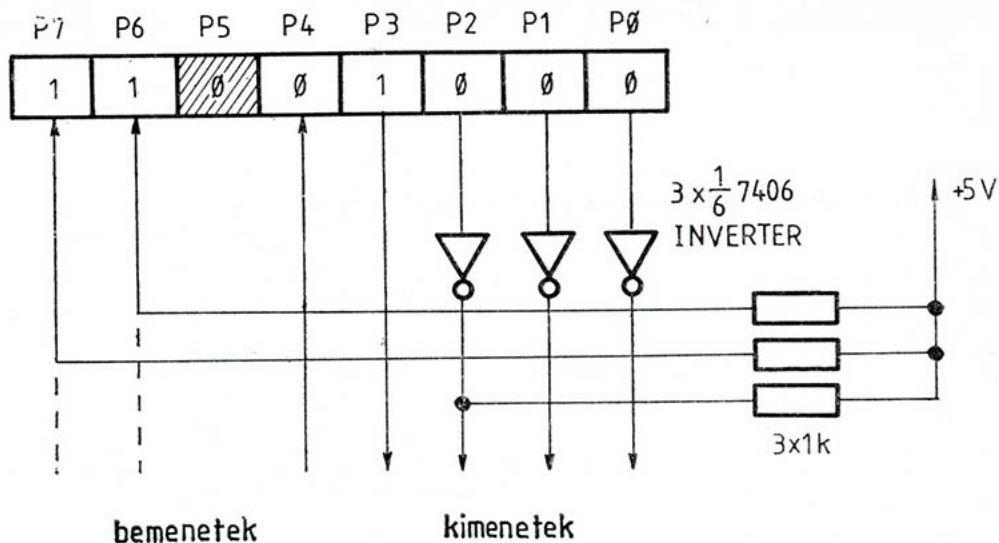
A rendszer szoftver kikerülésével magunk is kapcsolgathatjuk pl. BASIC programból az adatmagnó motorját. Ezt a szándékunkat a 2044. címen (\$07FC) lévő 54 érték 128-ra (\$80) változtatásával és a processzorport P3. bitjének 0-ra állításával érhetjük el. A [PLAY] billentyűvel kapcsoljuk be az adatmagnót! Adjuk ki a következő parancsot:

POKE 2044,128

Ezután a POKE 1,PEEK(1) OR 8 paranccsal (P3. bitre 1 kerül) leállíthatjuk, a POKE 1,PEEK(1) AND (255—8) paranccsal elindíthatjuk adatmagnónkat. Ha ismét a rendszernek akarjuk visszaadni az adatmagnó vezérlését, írjunk a 2044. címen lévő bajtba 4-et.

Aki idáig eljutott a processzor port tanulmányozásával, annak megéri, hogy egy kis időt még rászánjon! Figyeljük meg a kapcsolási rajzon, hogy a P0, P1, P2 bitek egy **openkollektoros inverteres meghajtófokozaton** [19.], [20.] keresztül kapcsolódnak a soros kimenetre! Ezért a csatlakozó lábain a port kimenőjelének *negáltja* jelenik meg.

A rendszer inicializálásakor (indítás utáni kezdeti értékekkel való feltöltésekor) a processzor port alsó 4 bitje kimenetre állítódik és az adatregiszter P3-as ($2^3=8$) bitjébe 1 íródik. (Lásd ismét az 1.13. ábrát.) Mivel a kapcsolásból következően a kimenet P0. bitjének invertáltja a P7. bitre, a P1. bitnek az invertáltja pedig a P6 bitre visszajut, ezért olvasható kezdetben az adatregiszterből 200 (\$C8). Ne feledjük el, az áramkör kialakítása olyan, hogy a P5 bit nincs kivezetve, és az integrált áramkörön belül fixen 0 logikai szinten van, a bemenetek pedig „szabadon hagyva” szintén 0 logika szintet adnak az olvasásakor! (1.15. ábra.)

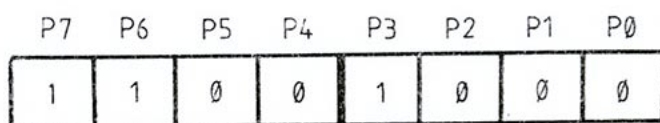


1.15. ábra. A processzor port adatregiszterének állapota inicializálás után

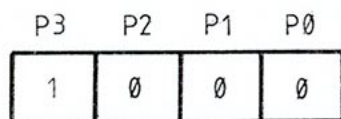
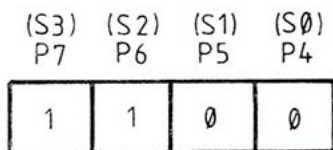
Írjunk az adatregiszter alsó 4 bitjére különböző értékeket és figyeljük meg a felső 4 bit változását! (1.16. ábra.)

A feladatot könnyen elvégezhetjük, ha az egyes bitkombinációkhoz tartozó hexadecimális számokat meg tudjuk határozni. Erre mutatunk be (elvi bizonyítás nélkül) egy gyors módszert: [4.], [20.]

Tegyük fel, hogy a processzor adatregiszterében a következő bináris kódszó van:



Bontsuk ezt két négyes csoportra:



A négyes csoportokra bontott bináris számokat könnyen átalakíthatjuk tízes számrendszerűvé (és mindkettő kisebb lesz mint 16). Az egyes biteket nem eredeti helyiértéküknek megfelelően, hanem a négyes csoportban elfoglalt helyük szerint vesszük figyelembe.

$$\underline{1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 12}$$

$$\underline{1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 8}$$

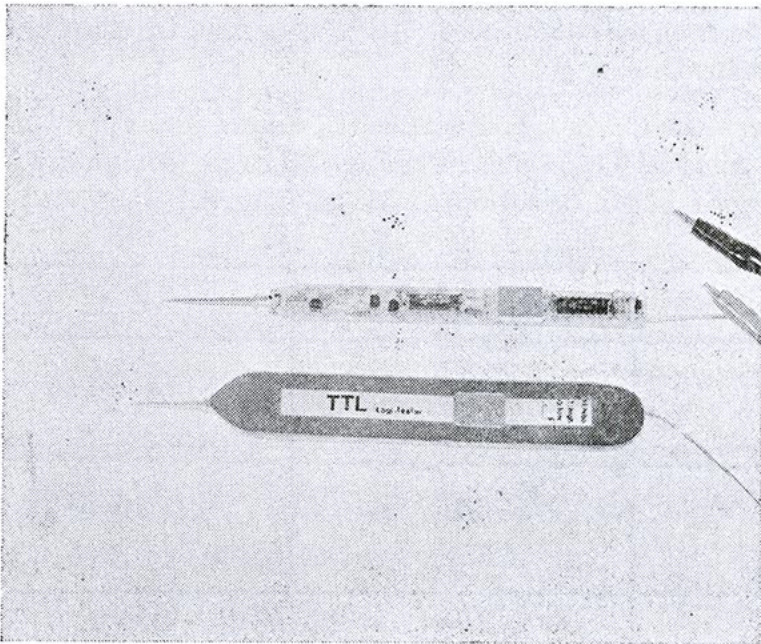
(Az új értékeket monitorból, az alacsonyabb helyiérték felülírásával lehet beírni az adatregiszterbe.)

Az 1.16. ábra elemzéséből és a kapcsolási rajzból most már egyszerűen meg lehet állapítani a „TED-rendszer” processzor portjának lehetőségeit. Ha jól megértettük a leírtakat, könnyen kapcsolatot teremthetünk a külvilág és házi számítógépünk között.

Ehhez azonban szükségünk van egy jó, **stabilizált tápegységre** és egy **impulzusokat is érzékelő logikai teszterre**. [13.], [21.], [22.]. Mivel tápegység készítésével nagyon sok szakirodalom foglalkozik [23.], [24.] ezért itt egy másik nélkülözhetetlen eszköz, a logikai állapotvizsgáló elkészítését mutatjuk be és feltételezzük, hogy a felhasználónak már van egy 5 V-os stabilizált, legalább 500 mA terhelhetőségű tápegysége, amelynek kivezetési **leválasztottak** és **földfüggetlenek**. [19.], [13.].

1.6. Építsünk logikai tesztert!

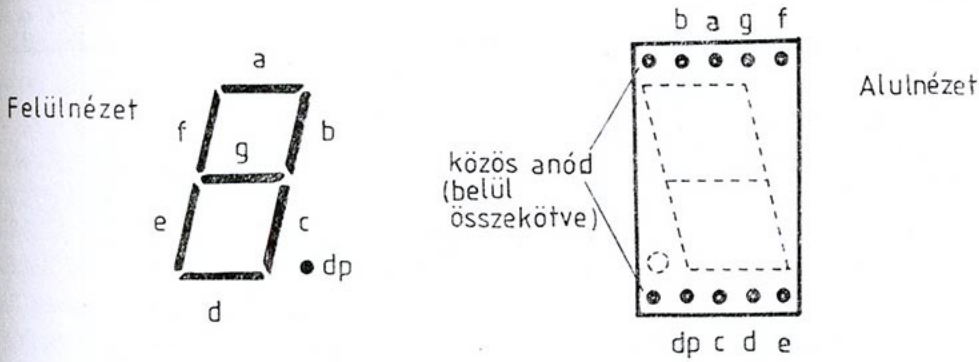
Az általunk építendő logikai teszter a szabványos TTL áramkörök kimeneti szintjeinek érzékelésére alkalmas ún. *dinamikus állapotvizsgáló*. Érzékelhető segítségével a 0 és az 1 logikai állapot, a kettő közötti „határozatlan” feszültség szint, a szakadás és az alacsony szintről magasra futó impulzus. Táplálása a vizsgált berendezés tápegységéről történik. Megépítésének formája az 1.17. ábrán látható (fénykép).



1.17. ábra. A logi-teszter és a szerelt panel

Szintjelzőnk a logikai állapotokat egy HA 1141 vagy HD 1141 típusú **közös anódos hétszegmentes kijelzőn** közli [25.]. (A kijelző bekötése az 1.18. ábrán látható.)

A két típus között csak annyi az eltérés, hogy borításuk színe különböző, de mindkét fajta pirosan világít.)



1.18. ábra. A HA1141 (HD1141) hétszegmenses kijelző szegmenseinek jelölése és a kijelző bekötése

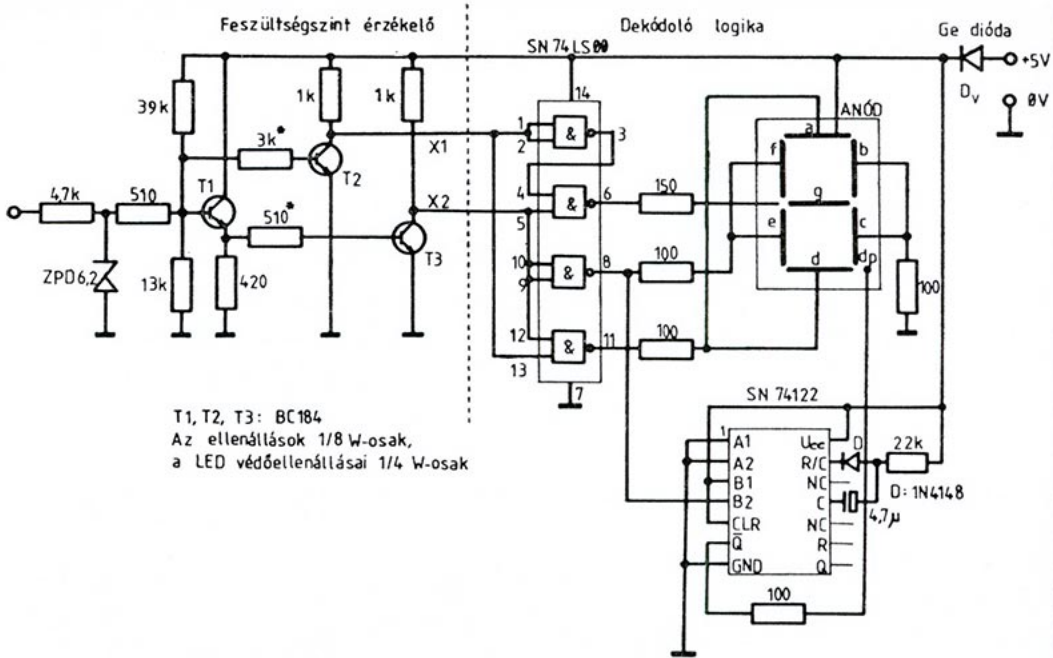
Ha a tapogató csúcsán a földhöz képest logikai 0 feszültség szint van, akkor a kijelzőn is 0 világít. Logikai 1 esetén teszterünk kijelzője 1-et mutat. A TTL szempontból hibás feszültség szintet (valamelyik logikai IC bemenetén vagy kimenetén a feszültség szint (U_v) ($U_{LH}=0,4\text{ V}$; $U_{HL}=2,4\text{ V}$) $0,4\text{ V}$ és $2,4\text{ V}$ között van ($U_{LH} \leq U_v \leq U_{HL}$) H betű kivilágításával közli és ugyancsak H világít, ha a vizsgált ponton nagyimpedanciás állapot [3.] vagy szakadás jelentkezik.

Megjegyezzük, hogy a számítógépekben is alkalmazott integrált áramkörök egyes típusainak kivezetései *tree-state* (TS) állapotúak. Ez azt jelenti, hogy az IC-k bizonyos kivezetései nemcsak 0 vagy 1 logikai feszültség szint megjelenítésére alkalmasak, hanem egy vezérlőjel hatására (amit \overline{EN} -nel vagy G-vel szoktak jelölni) ún. *nagyimpedanciás állapotba* hozhatók. A kivezetések ekkor úgy viselkednek, mintha szakadások lennének, mintha levájasztanák („levágnák”) magukat az áramkörrel. [3.], [28.] Ebben az esetben olyan kis áramok folynak az áramkörben, hogy logi-teszterünk ezt szakadásnak észleli.

A $0 \rightarrow 1$ átmenettel dinamikus változó jelek esetén a hétszegmenses kijelző digitális pontot megjelenítő LED-je [26.] felvillan, gyorsan változó, vagy periodikus jelek esetén folyamatosan világít. (*Dinamikus működés.*)

A teszter kapcsolási rajzából (1.19. ábra) látható, hogy az áramkör feszültség szintérzékelő és dekódoló logikai részekből áll.

A szintérzékelő áramkör bemenetét a 39 KOhmos és a 13 KOhmos ellenállásokból kialakított feszültségosztóval úgy feszítjük elő, hogy a T_2 bekapcsolt állapotú, így az X1 ponton 0 szintnek megfelelő a feszültség. A T_1, T_2 bázisán $\approx 0,8\text{ V}$ mérhető. Vigyázzunk, mert ez erősen függ a D_v védődiodán eső feszültségtől! A T_3 bázisán $\approx 0,2\text{ V}$ van, így X2 ponton logikai 1 szintű a feszültség. Ezt az állapotot $I_{BE}=0$ vagy $0,4\text{ V} < U_{BE} < 2,4\text{ V}$ -nál tapasztalhatjuk. A tranzisztorok *bétájától* függően, a (*)-gal megjelölt ellenállások változtatásával lehet beállítani a billenési értékeket. Az



1.19. ábra. A TTL logi-teszter elvi kapcsolási rajza

510 Ohmos ellenállás megváltoztatásával az $U_{BE}=2,4$ V-hoz tartozó billenést, a 3 KOhmos ellenállással pedig az $U_{BE}=0,4$ V-hoz tartozó küszöböt lehet befolyásolni.

Az 1.20. ábrán látható táblázat a feszültség-szint-érzékelő áramköri rész működésének megértését segíti.

kimenetek	$U_{be} < 0,4V$	$U_{be} > 2,4V$	$I_{be} = 0$	$0,4V \leq U_{be} \leq 2,4V$
X1	1	0	0	0
X2	1	0	1	1
kijelző	0	1	H	H

1.20. ábra. A logi-teszter feszültség-szint érzékelőjének kimenetei statikus állapotban

A dekódoló logika egyetlen 74LS00 IC. Ez egyben a hétszegnemes kijelző meghajtó áramköre is. A szegnemesek védő ellenállásait tapasztalati úton állítottuk be. Működése a NAND kapu igazságtáblája és az 1.20. ábra alapján azonnal megérthető, ha figyelembe vesszük, hogy az egyes állapotokhoz tartozóan mikor, melyik szegnemset

kell kigyújtani. A közös anódos működés miatt mindig azok a szegmensek világítanak, amelyekhez tartozó meghajtó áramköri kimenetén \emptyset logikai feszültség szint van.

A dinamikus működést lehetővé tevő, a $L \rightarrow H$ billenést kijelző digitális pontbegyújtás „trükkje” is rendkívül egyszerű. A hétszegmenses kijelző e , f szegmensei a \emptyset és a H állapotban világítanak. Ekkor a meghajtó IC 8. kivezetésén \emptyset logikai feszültség szint van. Abban a pillanatban, amikor a bemeneti feszültség áthalad a komparálási ponton, a 8. lábön egy felfutó impulzus jelenik meg és beindítja az SN74122 típusú újra indítható monostabil multivibrátort. Ez az időzítésnek megfelelően (esetünkben kb. 100 ms hosszú), egy villanásnyi időre bekapcsolja a digitális pontot a kijelzőn. A felvillanásból lehet tudni, hogy egy $\emptyset \rightarrow 1$ átmenetű impulzus volt a bemeneten. Bár az eszközt az áramköri elemek működésének ismerete nélkül is meg lehet építeni, azért javasoljuk, hogy az Olvasó a szakirodalomban nézzen utána a **tranzisztor kapcsolóüzemű működésének** [19.], [20.], **a LED-ek és a hétszegmenses kijelzők leírásának** [25.], az SN74LS00 NAND kapu [20.], [15.] és az SN74122 monoflopp [20.], [15.] és általában a **multivibrátorok** [29.] működésének [30.].

Az áramkörből már tucatnyi készült. Tapasztalataink szerint az érdeklődő 13—14 éves gyerekek is megértik működését és kb. 6—8 óra alatt „utánépítik”. Mielőtt (minden elvi magyarázat nélkül) leírjuk a **NYÁK-készítés Alfaset-projektor-vasklorid** technológiáját, ismételten felhívjuk a figyelmet arra, hogy a bonyolultabb logikai áramkörök megépítése előtt célszerű a logi-tesztet megépíteni! Ez az „élesztések” során helyettesíti az oszcilloszkópot és a multimétert is.

17. A NYÁK-készítés legegyszerűbb módszere

Az Alfaset-projektor-vasklorid „nyáktechnológia algoritmus”.

1. lépés: Az elvi kapcsolási rajz alapján milliméterpapírra készítsük el NYÁK-lapunk *huzalozási tervét* úgy, hogy az egyik oldalon a *forrasztási oldal*, a másik oldalon az *alkatrészbeültetési-oldal* hálózata legyen a papíron. Ilyen kétoldalas rajzolatú milliméterpapírt átvilágított üveglap (vagy nappal az ablaküveg) segítségével gyorsan készíthetünk. Az M 1:1 méretű rajzot vágjuk a tervezett pontos NYÁK-méretre!
2. lépés: Vágjuk pontos méretre *kétoldalas nyáklapunkat* is! Tisztítsuk meg a vágási éleket a sorjától, majd celluxszal feszítsük rá a NYÁK-tervet a lap egyik oldalára! Pontozóval (pontosan!) jelöljük át a furatok helyét a NYÁK-ra, majd óvatosan vegyük le a rajzot!
3. lépés: Nagy fordulatszámú fúróval (mi erre a célra MINIPLEX MK II. típusút használunk a hozzá tartozó tápegységgel) fúrjuk ki a lyukakat! (\emptyset ,6 mm-es lyuk a legtöbb alkatrésznek elég.)

4. lépés: Óvatosan, θ -ás, ill. polírozó csiszolópapírral sorjátlanítsuk a furatokat, egyenletesen (és finoman!) csiszolva szedjük le a lemezre rakódott sötét színű *oxidréteget*! Tiszta benzines, puha ronggyal zsírtalanítsuk a lemezt! Rövid ideig szárítsuk, hogy a benzin elpárologjon róla!
5. lépés: Ezután ALFASET fóliáról puha ceruzával nyomjuk rá a lemezre az IC-k, alkatrészek forrasztási pontjait. (Mi az ALFASET IC 413, IC 415, IC 417 jelű fóliákat használjuk erre a célra, amelyek Budapesten a Művészellátó és Grafikai Eszközök Boltban, a Bajcsy-Zs. út 18. sz. alatt szerezhetők be!) Amikor mindkét oldallal elkészültünk, az ALFASET fóliák hátoldalán található védő zsírpapíron keresztül kézzel, erősebben nyomkodjuk át a forrpontokat, hogy biztosan tapadjanak!.
6. lépés: Most következik a vezetősávok felrajzolása *acetonos filccel* (projektorral). Ajánlani tudjuk a STAEDTLER LUMOCOLOR 313 WP4-es készletből az S jelűt (szuperfinom) és az F jelűt (finom), a STABILO OHPen 9604 készletből az S és az F jelűt. Ezek közül a feketék és a kék színűek váltak be igazán. Figyeljünk rá, hogy mindig olyan projektor tollat használjunk, amelyik nem a kimerülés határán van, vonlázó mellett húzzuk a vonalakat és száradás után ugyanarra a vonalra ne menjünk rá ismét!
7. lépés: Hagyjuk 2—6 óráig száradni munkánkat! Ha erre nincs időnk, akkor pl. íróasztali lámpa alatt 15—20 percig tartsuk 50—60 °C-on NYÁK-lapunkat! Ez nagyon lényeges fázis a maratás előtt! Tapasztalatunk szerint a jól megszáritott NYÁK-ról még hosszú maratás után sem jön le a projektor, ill. az Alfaset.
8. lépés: A maratófürdőt a maratás előtti napon célszerű elkészíteni úgy, hogy telített vaskloridos oldatot kapjunk. Annyi vaskloridot kell feloldani egy befőttes üvegben (szobahőmérsékletű vízben), hogy az alján még úszkáljon néhány darab. (Vaskloridot a gyógyszertárakban vagy a REANAL boltokban, vagy az Elektron Áruház hálózat alkatrész boltjaiban nagyon olcsón lehet kapni.) A maratás gyorsabb, ha az oldat 40—45 °C hőmérsékletű. Azt tanácsoljuk, hogy e fölé ne menjünk, inkább várjuk ki az 5—10 perccel többet, mert az Alfaset különben leolvad a NYÁK-ról! A lemezt az oldatba helyezve (pl. felfüggesztve) úgy végezzük a maratást, hogy a folyadék körüljárja a NYÁK-lapot. A lemezt ki-kimeelve ellenőrizzük a maródást! A jól előkészített NYÁK ezzel a módszerrel kb. 20—30 perc alatt tökéletesen maratva készül.
9. lépés: Maratás után mossuk le a lemezt, majd SUPERDOL súrolószerrel távolítsuk el róla az Alfaset és a projektor maradékokat! Minden olyan helyen, ahol Alfaset vagy projektor rajz volt, megmarad a rézfólia, a többi lemaródik.
10. lépés: Erős mosás és szárítás után N—35-ös, az Elektron Boltokban kapható

forrasztóspray-vel vonjuk be a NYÁK-ot, majd 50 W-os Weller (5-ös lapos fejjel) pákával melegen ónozzuk be a vezető sávokat!

Ezzel a módszerrel — kis gyakorlattal — olyan NYÁK-ok is készíthetők, amelyek vezetősávjai vékonyabbak mint 1/2 mm. Az egész NYÁK-készítési technológiát mélyebben a szakirodalomból tanulmányozhatjuk át. [31.], [32.]

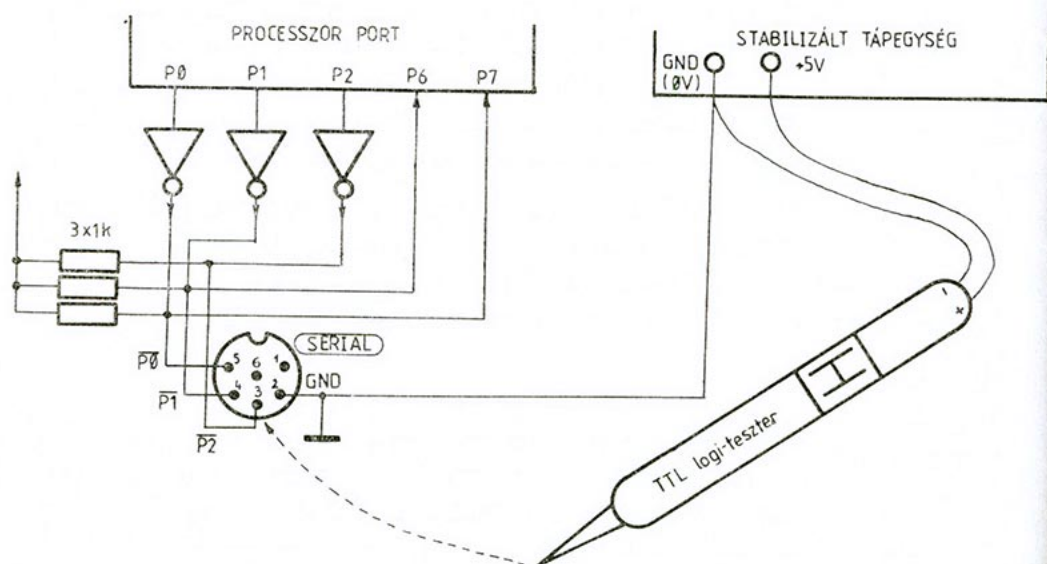
Ha elsőre nem sikerül, ne keseredjünk el! Gyakorlat teszi a mestert!

A célszerűség és a működési biztonság alapján döntünk, hogy mennyi a nyákra maradt vezetősáv vagy furat, és mennyi a huzalozott kötés egy kapcsolás megvalósításakor. Az eszközök működési biztonsága és az esztétikus, pontos kivitel a tapasztalatok szerint egymással összefügg. A gondos és precíz munka eredményeként előállított eszköz működési valószínűsége mindig nagyobb, mint a hevenyészve „össze lapátolté”.

2. KAPCSOLAT A PROCESSZOR PORTON

2.1. Logikai értékek és feszültségszintek érzékelése

A processzor port egy *latch* [19.], (amely az adatregiszterébe írt adatot mindaddig tartja, amíg azt felül nem írjuk) a memória 0. és 1. címén, a címterületbe ágyazva. (Bővebben lásd az 1.4. fejezetben.) A processzor port P0., P1., P2., P6. és P7. adatvonalát a gyártók a számítógép SERIAL feliratú csatlakozóján kivezették az alábbiak szerint: (2.1. ábra)



2.1. ábra. A processzorport kivezetései a SERIAL csatlakozón

Kikapcsolt állapotban kössük össze a SERIAL csatlakozón kivezetett földpotenciálra lévő GND pontot a stabilizált tápegység 0 V-os kivezetésével, a Logi-tesztet pedig tápláljuk a tápegységről. Kapcsoljuk be a tápegységet, majd a számítógépet, és a tesztterrel vizsgáljuk meg a kivezetéseket! Az egyes kivezetések feszültségszintjeit kis műszerünk a közös földpotenciálhoz viszonyítva mutatja. Adjuk ki a következő parancsokat, és írjuk táblázatba a mért értékeket! (2.2. ábra.)

A táblázat 1. sorában az inicializálás utáni kezdeti ún. default értékek láthatók. A második sorban megváltoztattuk a 2^1 helyiértéken szereplő bitet 1 állapotúra. Mivel a kimeneten az openkollektoros invertáló meghajtó fokozat miatt ennek negáltja, a 0 jelenik meg a P1-hez tartozó negyedik lábón, ezért a processzorport P6 bitjének és a

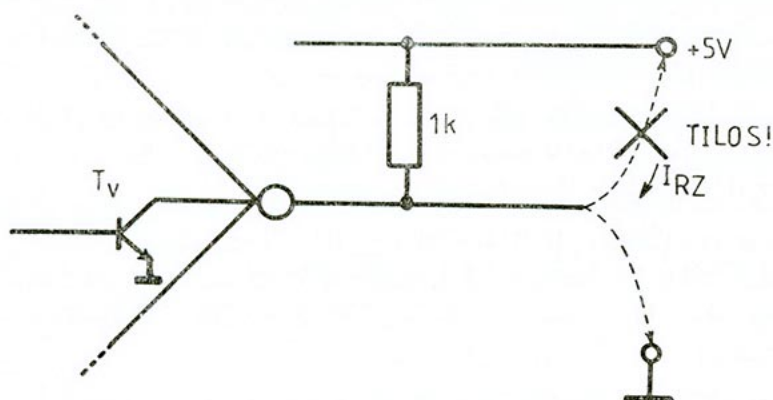
PARANCS	MÉRT ÉRTÉK			ADATREGISZTER		
	P2	P1	P0	BINÁRISAN	HEX.	DEC.
1. közvetlenül bekapcsolás után	1	1	1	1 1 0 0 1 0 0 0	C8	200
2. POKE1, PEEK (1) or 2	1	0	1	1 0 0 0 1 0 1 0	8A	138
3. POKE1, PEEK (1) or 6	0	0	1	1 0 0 0 1 1 1 0	8E	142
4. POKE1, PEEK (1) or 15	1	1	1	0 0 0 0 1 1 1 1	0F	15
5. POKE1, PEEK (1) and (255-3)				1 1 0 0 1 1 0 0		204
6. POKE1, PEEK (1) and (255-4)						
7. POKE1, PEEK (1) or 192	1	1	1	1 1 0 0 1 0 0 0	C8	200

2.2. ábra. Mérések a processzorporton

csatlakozó 4. lábának galvanikus kapcsolata miatt ez a 0 érték állandóan visszajut a bemenetként kapcsolt P6 bitre. Ezért olvashatunk ilyenkor az adatregiszterből 88A-t. Hasonlóan fejthető meg a többi sor tartalma is. Talán érdekes lehet a 7. sorhoz fűzött megjegyzés. Ebben a sorban a processzorport adatregiszterének felső két bitjére írtunk 1-et. Sem a mért értékben, sem a beolvasott értékekben nem történt változás. Ez azért van, mert a bemenetként működtetett portok az ellentétes irányú adatáramlás elkerülése miatt a számítógép belső sínrendszeréről nem tudnak adatot fogadni (az elektronikájukat úgy alakították ki). A sínrendszert *busznak* (információ-tömegszállító) is szokták nevezni.

Próbáljuk most az előző eljárás fordítottját végigkövetni! Változtassuk meg a SERIAL-port (csatlakozó) bemenetein a feszültség szinteket és olvassuk be a kapott értékeket!

Nagyon vigyázzunk! A környezettel „érintkező” U7 jelű meghajtó IC szabad kollektoros kimenetű. Ezért, ha a processzorport P0. kimenetén 1 van, ne próbálkozzunk a SERIAL-csatlakozó 5. kivezetésén a 0-t egy vezetékkel „felhúzni” logikai 1-re, mert a végfokozat tönkremegy! (2.3. ábra.)



2.3. ábra. Nyitott kollektoros végfokozat kimenetét soha ne kössük tápfeszültségre!

Ekkor nagy rövidzárási áram folyik keresztül a T_v végtranzisztoron. [19.] Természetesen, ha a kimeneten 1 szint van, akkor a kimenet és a föld rövidre zárásával ez „lehúzható” 0 szintre. Ilyenkor a külső munkaellenálláson ≈ 5 mA áram folyik, amely nem okoz kárt.

Kísérleteink során tehát csak 0 logikai állapotot olvassunk be a portról! Kössük össze egy vezetékkel a SERIAL-port 2. kivezetését (GND) a 6. számú kivezetés kivételével bármelyikkel, vagy bármelyik kettővel, földpotenciálra tartva azokat! Készítsünk táblázatot a beolvasott értékekről! (2.4. ábra.)

SERIAL PORT			ADATREGISZTER								HEX.	DEC.
FÖLDPOTENCIÁLON LEVŐ BIT	BINARISAN		P7	P6	P5	P4	P3	P2	P1	P0		
\overline{P}_2 (3)	\overline{P}_1 (4)	\overline{P}_0 (5)										
Bekapcsolás utáni alapállapot			1	1	0	0	1	0	0	0	C8	200
1.		GND	0	1	0	0	1	0	0	0	48	72
2.	GND		1	0	0	0	1	0	0	0	88	136
3.	GND		1	1	0	0	1	0	0	0	C8	200
4.	GND	GND			0							

Beolvasó parancs
 BASIC-ben:
 ? PEEK (1)
 MONITOR-ból:
 >0

2.4. ábra. Adatbevitel a processzor portról

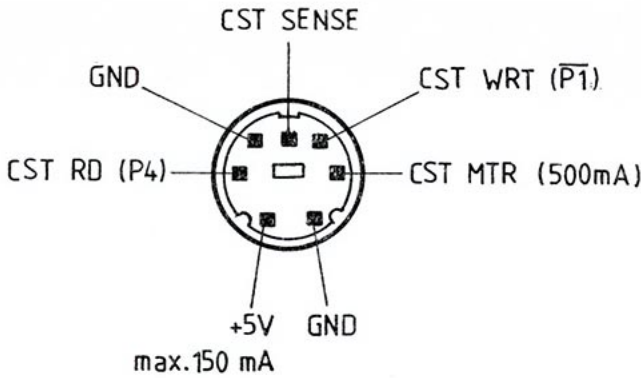
Bekapcsolás után a default érték \$C8 az adatregiszterben, így a $\overline{P0}$ (5), $\overline{P1}$ (4), $\overline{P2}$ (3) adatbiteknek megfelelő kivezetésen a SERIAL-porton csupa 1 szint van. Az 1. sorban földpotenciálra kötöttük a $\overline{P0}$ bitet. Ez tulajdonképpen a P7. processzorport bit földelését jelenti, hiszen a SERIAL-port 5. kivezetése ezzel a bittel van összekötve közvetlenül. A nyitott kollektoros meghajtófokozaton a kimenet földelése — természetesen — nem hat vissza a bemenet állapotára (különben is a bemenet 0 értéken van a $P0=0$ miatt). Így kapunk a beolvasás után az adatregiszterben \$48 értéket. A többi sorban is hasonló módon értelmezhető a változás.

A 3. sorban a $\overline{P2}$ (3) bitet földeltük a SERIAL-porton. Az adatregiszterben azért nem történt változás a beolvasott adatban, mert ahogy azt a C 16-os kapcsolási rajzán [2.], [1.] is megfigyelhetjük, ez a SERIAL-porton egyik bemenetnek kapcsolt bittel sincs galvanikus kapcsolatban.

Hasonló változást észlelünk, ha a Cassette-port 5. kivezetését (CST WRT, P1) kötjük földpotenciálra, és olvassuk az adatregiszter tartalmát.

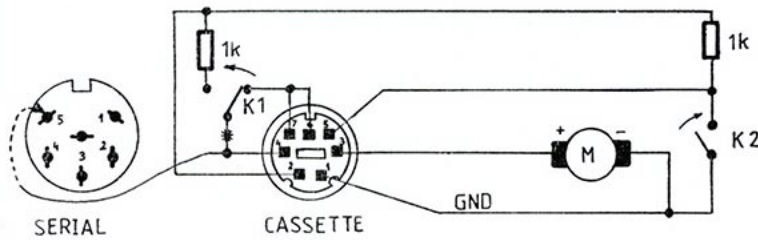
A C 16-os és a C Plus/4-es eltérően érzékeli a CST SENSE jelet. Ez a jel akkor 0 szintű, ha a magnón a [PLAY], [REWIND], vagy [F. WIND] beillentyűk valamelyikét lenyomták. A C 16 a processzorport 7. bitjén (P7) érzékeli a lenyomott állapotot, a C Plus/4 nem itt. Hogy hol, erre később a C Plus/4 USER-portjának ismertetésekor kitérünk.

Ha a P4, biten 1 logikai állapotot szeretnénk beolvasni, akkor kössük össze a Cassette-port 5. kivezetésével ($\overline{P0}$)! Ezt azért szabad megtenni, mert a P0 biten lévő szabad



2.5. ábra. A CASSETTE-PORT csatlakozójának bekötése

kollektoros meghajtó munkaellenállása (1 kOhm) megvédi a P4 vonalat még abban az esetben is, ha a processzorport parancsregiszterében (a 0. bájton) véletlenül ki-menetnek van kapcsolva. (2.6. ábra a) része.)



Kötés nélkül: ? PEEK (1) → 200
 Összekötve: ? PEEK (1) → 216
 (Logikai 1 szint jelent meg a P4 biten)

A cassette port 4. kivezetését csak akkor szabad a serial port 5. kivezetésével összekötni, ha a 4. kivezetés a közölt kapcsolásban nincs földpotenciálton!
 Szakítsuk meg * helyen az áramkört!

a)

b)

2.6. ábra. A CASSETTE-PORT egyes csatlakoztatási lehetőségei

Könnyen vezérelhetünk egy kis, max. 500 mA áramot felvevő egyenáramú villanymotort a Cassette-port 3. kivezetéséről. A 2.6. ábra b) részén szereplő kapcsolásunkat és a C 16 házi számítógép kapcsolását együtt szemlélve látható, hogy ebben a bekötésben villanymotorunk egy impedanciaváltó, földelt kollektoros tranzisztor (emitterkövető) [19.], [20.] munkaellenállása. Mivel az interpreter [2.] állandóan figyeli a SENSE jel állapotát, ezért a motor bekapcsolhatóságának biztosításához két dolgot kell előre megtenni:

1. A SENSE jelvezetékét földpotenciálra (0 logikai szint) kötni (pl. a Cassette-port 6., 7. kivezetéseinek összekötésével).
2. Beállítani a magnó motorbekapcsolást engedélyező bájtot. POKE 2044,128.

A következő kis program ≈ 10 mp-es szünetekkel bekapcsolja a villanymotort ≈ 10 mp-re. A „V” betű lenyomására a folyamat leáll.

```

10 REM ** 4. mintaprogram **
11 :
12 :
14 POKE 2044,128:REM kapcsoló állítás
20 POKE 1,208:GOSUB 100:REM késleltetés
30 POKE 1,200:GOSUB 100
40 END
50 :
100 FOR I=1 TO 1800
110 GET A$
120 IF A$="v" THEN POKE1,PEEK(1) OR 8:PRINT "vége !":END
130 NEXT I
140 RETURN

```

A processzor port alapállapotában (parancsregiszterben \$0F, adatregiszterben \$C8) a K1 kapcsoló zárásával érzékelhetünk a port P4 bitjén logikai 1-et, a K2 kapcsoló zárásával pedig a P6 biten 0 szintet. Kapcsolásunkat kis fantáziával egyszerű játékok irányítására [33.], [37.] is felhasználhatjuk.

Az alábbi program bekapcsolja a K1 lenyomására a motort, K2 zárására pedig leállítja. Figyeljük meg, hogy nem a kapcsolók zárták, ill. szakították meg a villanymotor áramkörét, hanem a figyelőprogram hatására a processzor port elektronikája! A Cassette-port 4. kivezetése közvetlenül rákapcsolódik a processzor port P4 bitjére. Ez rendkívül zavarérzékeny! Kikapcsolt állapotát a 0 potenciálon tartás jelentsel!

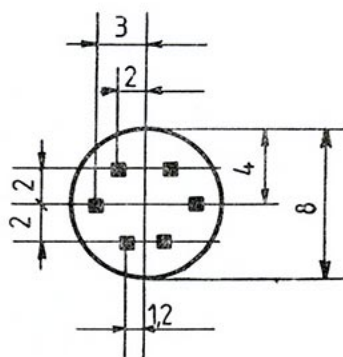
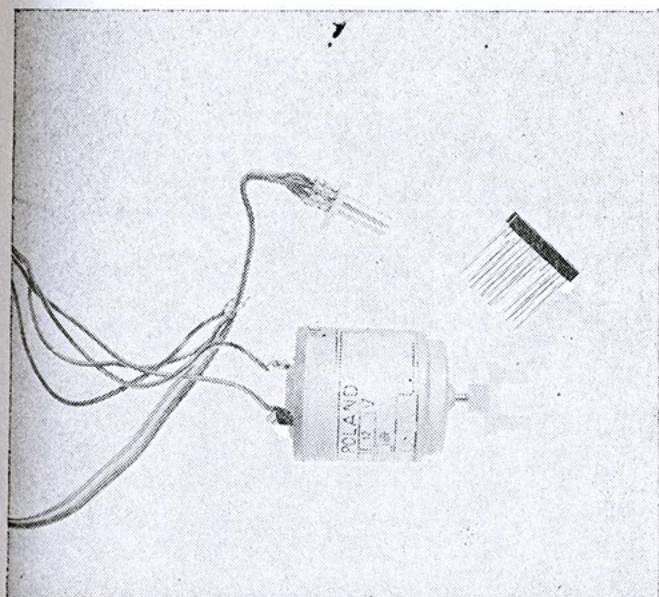
```

10 REM ** 5. mintaprogram **
11 :
12 :
14 REM processzor port inicializálása
15 REM motor engedélyezés
16 :
20 POKE 0,15:POKE 1,200:POKE 2044,128
30 REM k1 kapcsoló figyelése
40 A=PEEK(1):REM az adatregiszter beolvasása
50 IF A=216 THEN 100:REM k1 kapcsolót zárták
60 IF A=136 THEN PRINT "mit akarsz megállítani ?":GOTO 30
65 IF A=192 THEN 120
70 GOTO 30
80 :
100 POKE 1,208:REM a motor bekapcsolása
110 REM k2 kapcsoló figyelése
120 A=PEEK(1):REM az adatregiszter beolvasása
130 IF A=216 THEN PRINT "leállítani csak k2-vel !":GOTO 30
140 IF A=128 THEN POKE 1,200:GOTO 30
150 GOTO 30

```

Sajnos kísérleteinkhez eredeti gyári mini-DIN csatlakozót reménytelen vállalkozás a kereskedelemből beszerezni. Egy kis ügyességgel kis darab NYÁK-ból és tekerhető

kötésre alkalmas hosszúlábú (wire-wrap) IC foglalatból házilag készíthetünk a 2.7. ábrán (fénykép) látható módon.



2.7. ábra. Házi mini-DIN csatlakozó és mérethálózata

Egyszerűen elkészíthetjük csatlakozónkat úgy, hogy az aljzat fölé egy papírlapot teszünk és átszúrjuk a lyukak helyeit gombostűvel, majd a lyukakat pontozóval kijelöljük a NYÁK-lemezen.

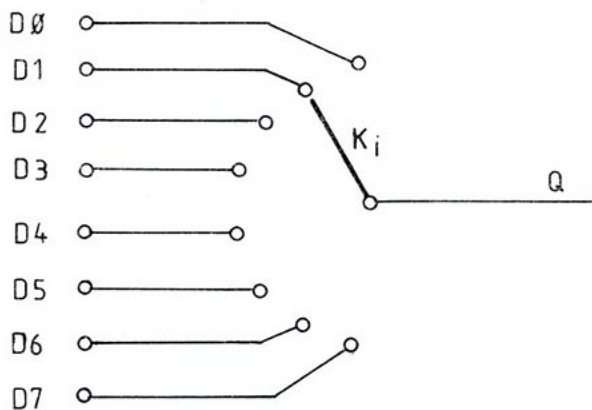
Vigyázzunk, az érintkezők a forrasztáskor nehogy összeérjenek!

2.2. 8 csatornás demultiplexer a SERIAL-porton

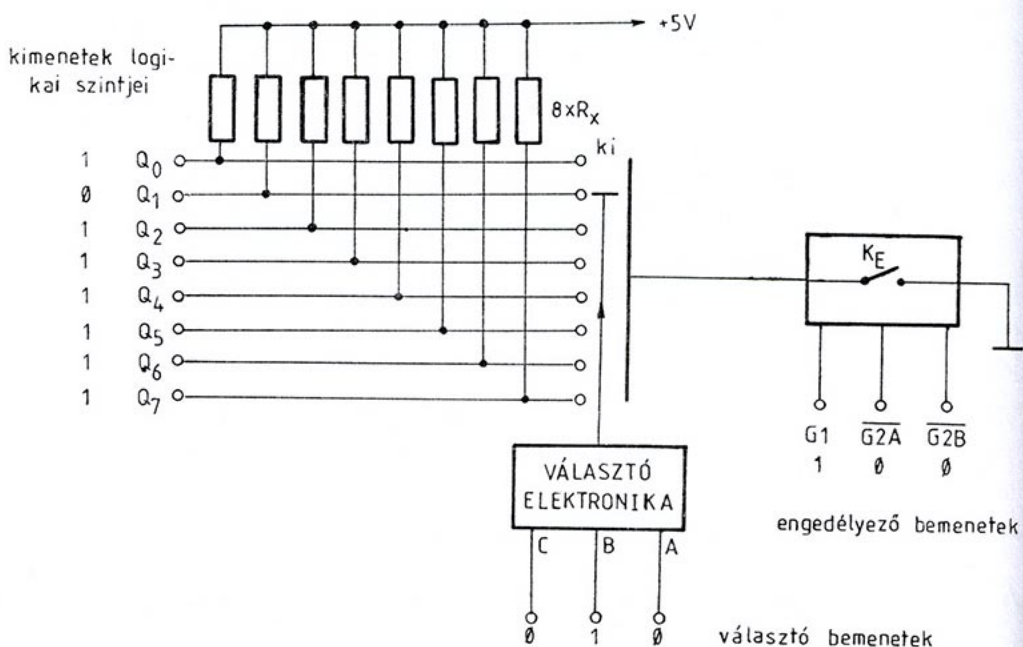
A *multiplexer* tulajdonképpen egy többállású kapcsoló. (Lásd a 2.8. ábrát.)

Bármelyik kapcsolt vonalról továbbíthatunk információt a Q vonalra, ha azt a K_1 kapcsolóval összekötjük. Ebben az esetben az információ a Di vonalokról jut a Q-ra, így azt mondjuk, hogy *8-ról 1-re multiplexálunk*. Fordított esetben az információ Q-ról a vonalakra juttatásakor 1-ről 8-ra demultiplexálunk. Természetesen elvileg akárhány kapcsolt vonal lehet. [14.], [15.] Nézzünk utána a szakirodalomban a multiplexereknek! [19.], [20.]

Mi az alábbiakban egy speciális dekóder [15.] (*demultiplexer*), az SN 74LS138 működésével ismerkedünk meg. Működésének logikája a 2.9. ábra alapján követhető. Abban az esetben, ha az *engedélyező bemenetek* valamelyikén nem az ábrán jelöltnek megfelelő a *bemeneti kombináció*, a földpotenciál nem jut el a K_i kapcsolóra. Így a kimenetek 1 szinten vannak. A $\overline{G2A}$ jelölésben a felülhúzás azt jelenti, hogy a jel



2.8. ábra. 8 csatornás multiplexer (8 állású kapcsoló)



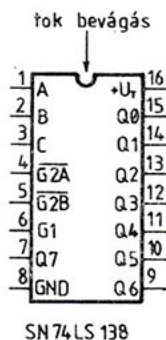
2.9. ábra. Az SN74LS138 működési modellje

aktív állapota a logikai 0 szint, azaz 0 állapota esetén fejt ki az áramkörben működő hatását. A felülvonás nélküli jelek logikai 1 szinten aktívak (pl. G1). Ezért, ha a $G1=1$, $\overline{G2A}=0$, $\overline{G2B}=0$ bemeneti kombináció van az ún. engedélyező bemeneteken, akkor a K_E kapcsoló zárt, a földpotenciál átjut K_i -re.

A választó logikával „mondhatjuk meg” a K_i kapcsolónak, hogy melyik kimeneti vonalat húzza le 0 logikai szintre. Mivel ezek a *választójelek* is kétállapotúak lehetnek csak, így a 3 választó bemenettel (A, B, C) $2^3=8$ kimeneti vonalat lehet kiválasztani (megcímezni). Az áramkör működési táblázatát és bekötését a 2.10. ábra mutat-

ja. Összefoglalva: az SN 74LS138 3-ról 8-ra típusú dekóder (demultiplexer), TTL áramkör. (Csak a tájékozottabbaknak: a kimenetek meghajtása TP, teljesítmény felvétele $P=21\text{ mW}$, átlagos jelterjedési idő $t_R=22\text{ ns}$.)

		BEMENETEK					KIMENETEK								
		engedélyezők			választók										
		G1	$\overline{G2A}$	$\overline{G2B}$	C	B	A	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0
1.	X	1	X	X	X	X	1	1	1	1	1	1	1	1	1
2.	\emptyset	X	X	X	X	X	1	1	1	1	1	1	1	1	1
3.	1	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	1	1	1	1	1	1	1	1	\emptyset
4.	1	\emptyset	\emptyset	\emptyset	\emptyset	1	1	1	1	1	1	1	\emptyset	1	1
5.	1	\emptyset	\emptyset	\emptyset	1	\emptyset	1	1	1	1	1	\emptyset	1	1	1
6.	1	\emptyset	\emptyset	\emptyset	1	1	1	1	1	1	\emptyset	1	1	1	1
7.	1	\emptyset	\emptyset	1	\emptyset	\emptyset	1	1	1	\emptyset	1	1	1	1	1
8.	1	\emptyset	\emptyset	1	\emptyset	1	1	1	\emptyset	1	1	1	1	1	1
9.	1	\emptyset	\emptyset	1	1	\emptyset	1	\emptyset	1	1	1	1	1	1	1
10.	1	\emptyset	\emptyset	1	1	1	\emptyset	1	1	1	1	1	1	1	1



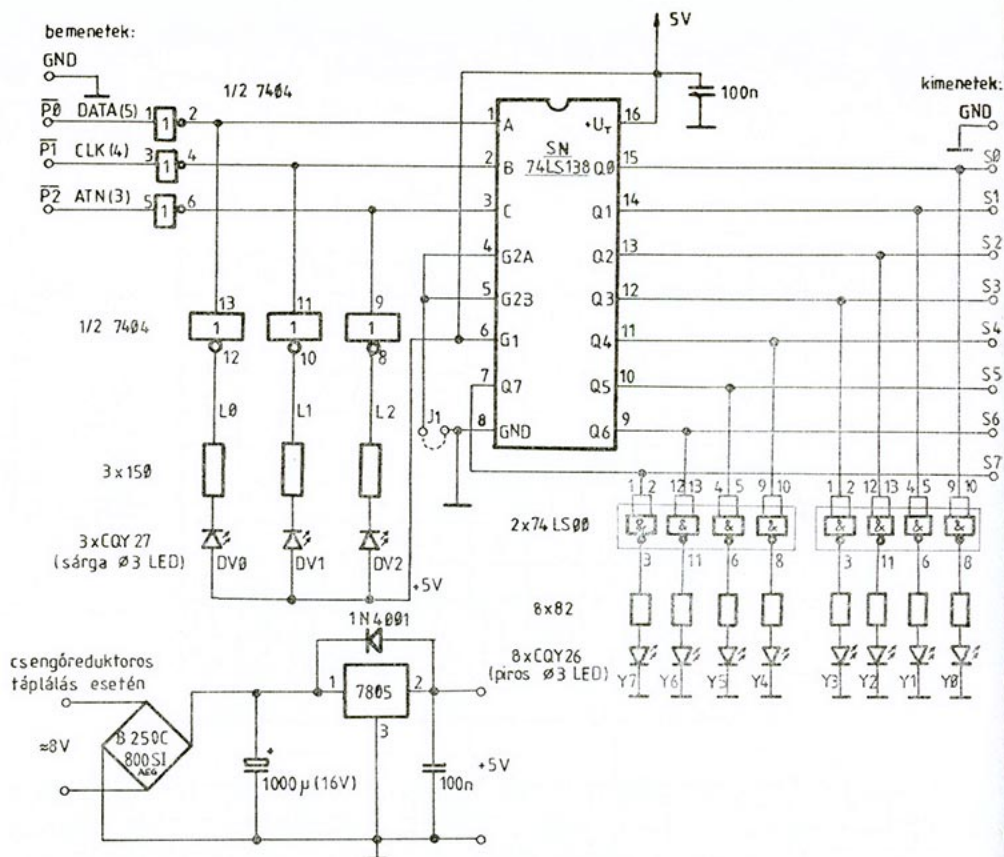
2.10. ábra. Az SN74LS138 működési táblázata és bekötése

Az 1. és a 2. sorokban alkalmazott *X* jelölés azt jelenti, hogy az adott jelek állapota teljesen közömbös, ha a $G1=\emptyset$, vagy a $\overline{G2}=1$, mert ekkor az összes kimenet 1 szintű, legyen akármilyen is a választójelek kombinációja (ez a tulajdonképpeni nem engedélyező állapot).

Felhívjuk a kezdők figyelmét, hogy az IC bekötése felülnézetből van megadva. Az 1. láb helyét a tokon legtöbbször egy kis ponttal megjelölik. Ha ilyen nincs, akkor felülről nézve a tok bevágása melletti bal oldali első láb az 1. sorszámú.

Ezek után könnyen megépíthetünk egy olyan 8 csatornás demultiplexert, amelyik a SERIAL-portról kapja a választójeleket, tápellátását pedig a CASSETTE-portról (vagy tápegységről, vagy egyszerű csengőreduktorról) biztosítjuk. Demultiplexe-rünk egyébként univerzális. Minden olyan számítógép tudja működtetni, amelyek rendelkezik *reteszelt kimeneti kapukkal (latchelt)*.

A bemenetekre érkező $\overline{P0}$, $\overline{P1}$, $\overline{P2}$ jeleket az SN 7404 (jó az SN74LS04 is) három invertere *ponált* [10.], eredeti állapotának megfelelő szintű jelként adja az SN 74LS138 választó bemeneteire [10.], (lásd az invertálás szabályait: pl. ha $A=1$, akkor $\overline{A}=\emptyset$, $\overline{\overline{A}}=1$; negált érték negáltja ponált!). $P1$. Az adatregiszterben $P0=1$, $P1=1$, $P2=\emptyset$, akkor a Q3 kimenet címződik meg. (Lásd 2.10. ábrát.) A választott kimenet \emptyset szintre esését a J1 pontok rövidre zárásával lehet bemutatni, ill. elérni. Az SN 7404 másik fele hajtja a DV0, DV1, DV2 LED-eket. Az invertált meghajtás miatt mindig az a LED világít, amelyikhez tartozó vonalon logikai 1 feszültség szint van. A LED-ek anódjára közvetlenül a tápfeszültség jut, a katódon a \emptyset logikai feszültség szint jelenne



2.11. ábra. 3-ról 8-ra demultiplexer

meg az inverterek bekapcsolásakor. Ezért a LED tönkremenne, ha a 150 Ohmos ellenállással nem védenénk. (Ez kb 12 mA-re korlátozza a LED-en átfolyó áramot.)

A kimenetek állapotát jelző LED-ek 74LS00 NAND kapukból [15.] kialakított inverterekkel vannak meghajtva. Mivel itt a Qi vonalak 0 állapotakor a LED-ek anódjára $\approx 3,2$ V feszültség jutna, katódjaik pedig földpotenciálon vannak, ezért itt is áramkorlátozó ellenállásokkal kell védeni a LED-eket (82 Ohm).

Az áramkör NYÁK rajzát a melléklet tartalmazza. Ezen egy *Greutz-egyenirányító* (B250C800 Si) és egy stabilizátor (μ A 7805) is van egy $I_{max}=500$ mA tápáramot leadni képes tápegység részeként. (Lásd 2.12. ábra fényképét.) A stabilizátor IC-t hűtőfelületre kell szerelni. Működését a szakirodalomból tanulmányozzuk át! [22.], [13.]

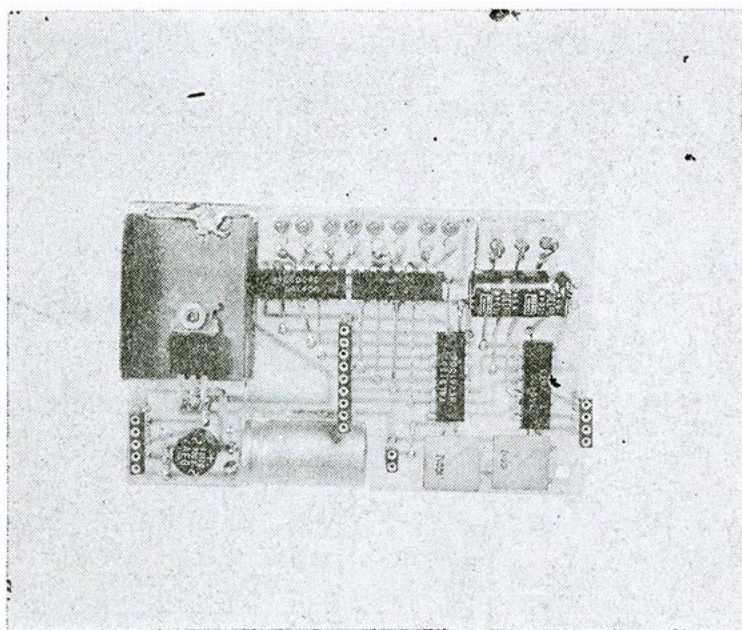
Mire használható kis eszközünk?

- Az alábbi program a LED-ek kapcsolgatásával demonstrálja az SN 74LS138 demultiplexer működését. A sárga LED-ek az éppen aktuális bemeneti (választó) kombinációt mutatják a piros LED-ek pedig (futófényként) az éppen logikai 0-ra kapcsolt kimeneten jeleznek.

```

10 REM ** 6. mintaprogram **
11 :
12 :
14 INPUT "v=";V
15 POKE 0,15:POKE 1,200:REM processzor port init.
20 FOR I=0 TO 7:FOR Z=0 TO V:NEXT Z:REM késleltetés.
30 POKE 1,200+I:REM bit allitas
40 NEXT I
50 GOTO 20

```



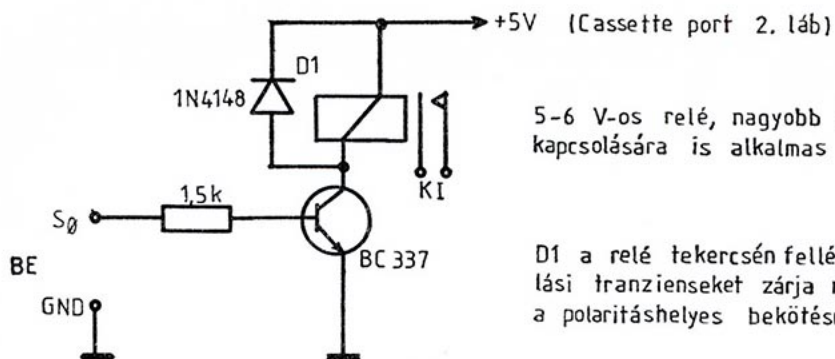
2.12. ábra. A megépített demultiplexer

Szemléletesen bemutatható vele a dekódolás folyamata.

b) A 2.13. ábrán látható módon az eszköz alkalmas például relés vagy tranzisztoros kapcsoló áramkörök, optocsatolás áramkörök működtetésére: (Megépítésüket haladóknak ajánljuk!)

Ha 3 darab vagy kevesebb, a 2.13. ábrán látható áramkörtípust szeretnénk működtetni, ezeket közvetlenül a SERIAL-portról is vezérelhetjük. Ebben az esetben — port reteszeltége miatt — egy adott időpontban egyszerre 1, 2, vagy 3 áramkör is üzemelhet. A demultiplexelt vezérlés nagy hátránya, hogy az áramköröket csak egymás után lehet vezérelni, így egy adott időpontban mindig csak egy áramkör „élhet”.

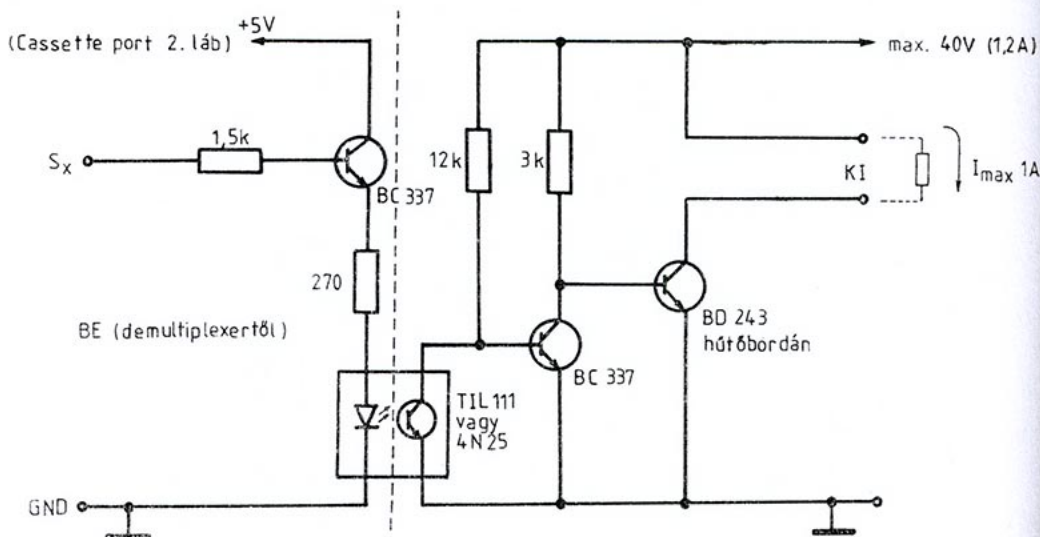
Az áramkörök működése nagyon egyszerű, leírásukat azért nem közöljük, mert működési elvüket a felhasználó által ismertnek, megfejthetőnek, vagy a szakirodalomból elsajátíthatónak feltételezzük.



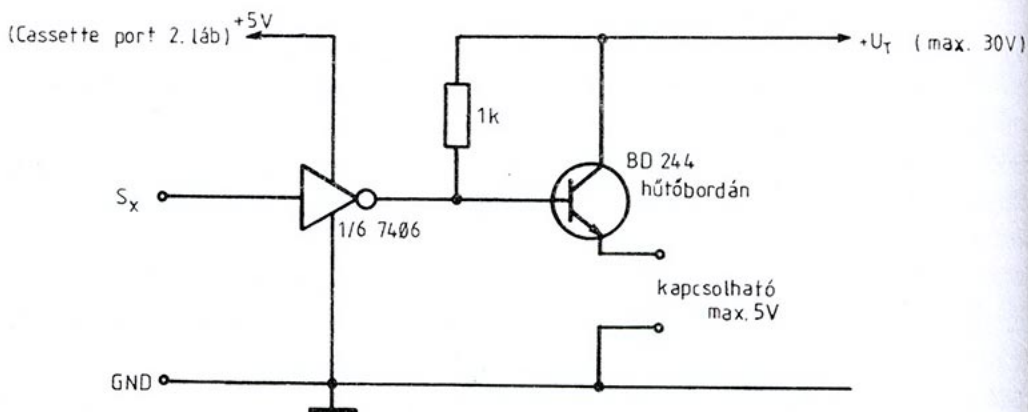
5-6 V-os relé, nagyobb áramok kapcsolására is alkalmas

D1 a relé tekercsén fellépő ki-be kapcsolási tranzienseket zárja rövidre. Vigyázzunk a polaritáshelyes bekötésre!

2.13. a ábra. Demultiplexerrel vezérelhető relés áramkör [20.]



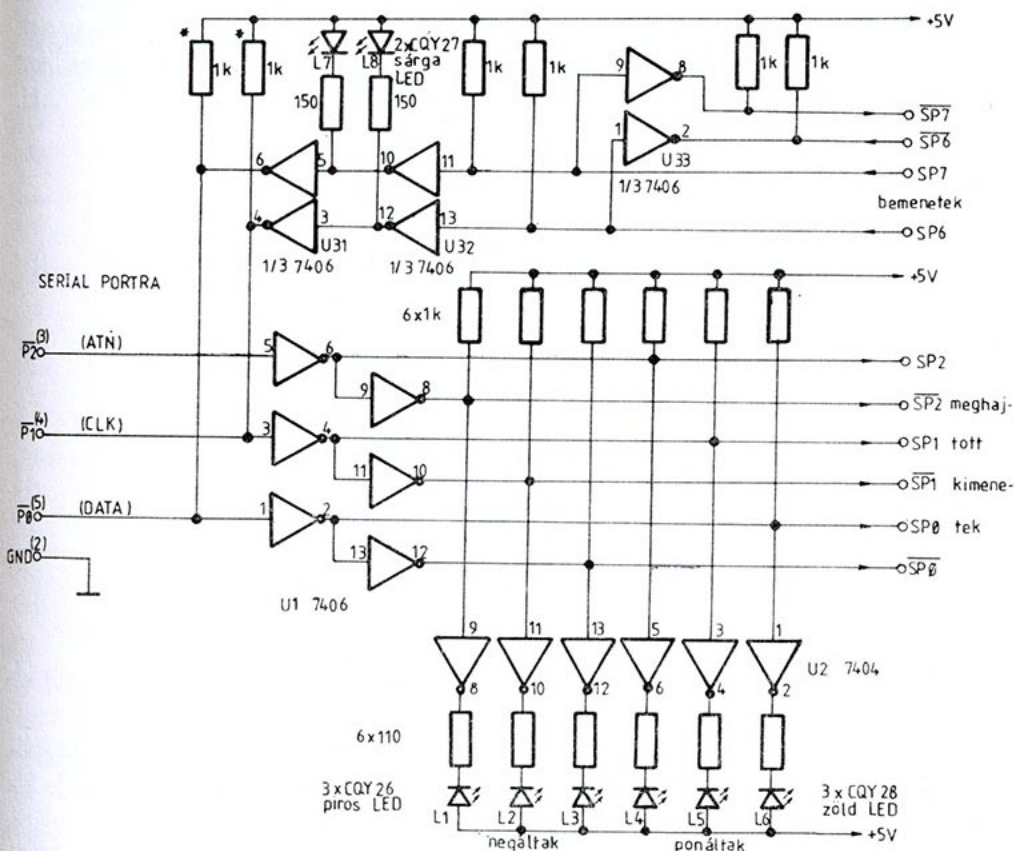
2.13. b ábra. TIL 111 optocsatolóval galvanikusan leválasztott külső kapcsolókör [26.], [19.]



2.13. c ábra. Tranzisztoros kapcsoló [19.], [20.]

2.3. A SERIAL-PORT védelme illesztő áramkörrel (3 párhuzamos kimenet és 2 párhuzamos bemenet)

A 2.14. ábrán egy, a SERIAL-portra csatlakoztatható meghajtó és adatvevő áramkör látható.



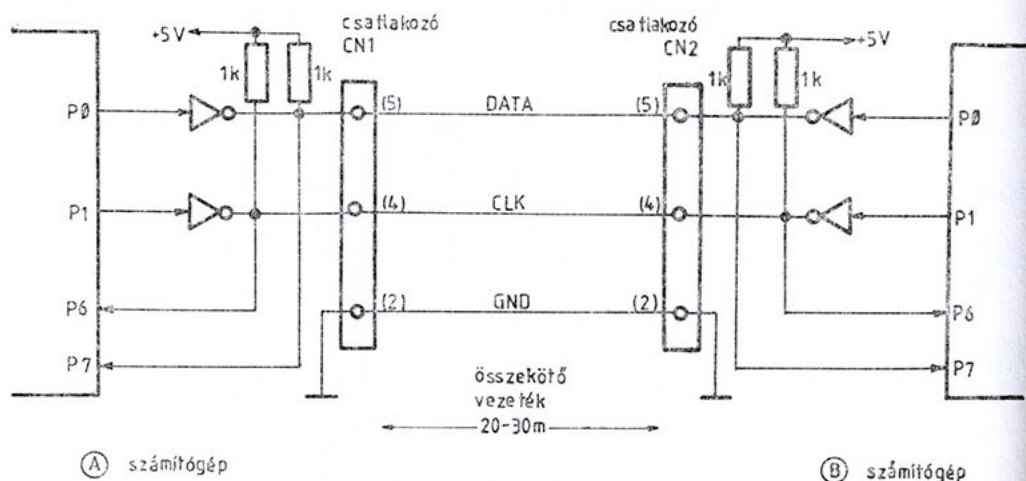
2.14. ábra. Meghajtott párhuzamos kimenetek, védett párhuzamos bemenetek a SERIAL-portra

Az U1 IC (7406) nyitott kollektoros meghajtók a kapcsolás kimeneteit közvetlenül táplálják. Külső 1 kOhmos munkaellenállásaik a stabil kimeneti szinteket abban az esetben is biztosítják, ha a $\overline{P0}$, $\overline{P1}$, $\overline{P2}$ bemeneteket nem csatlakoztatjuk a SERIAL-portra. Az áramkör SPi kimenő vonalainak állapotát az U2 (7404) inverterekkel meghajtott L_1, L_2, \dots, L_6 LED-ekkel lehet figyelni. A 110 Ohmos áramkorlátozó ellenállások a megadott LED típusok esetében még elegendő áramot engednek át a megfelelő fényerőhöz. Más LED típusoknál ezeket változtatni kell.

Az SP6, SP7 bemenetek U₃₁ jelű meghajtói és a SERIAL-port kimeneti meghajtói közös munkaellenállásokra dolgoznak. A (*)-gal megjelölt ellenállásokra azért volt

szükség, hogy az áramkör abban az esetben is rendelkezzen munkaellenállással, ha bemenete nincs a SERIAL-portra csatlakoztatva. L7, L8 LED-ek a bemenetek állapotának kijelzői. Mindig az a LED világít, amelyikhez tartozó vonalon logikai 1 szint van.

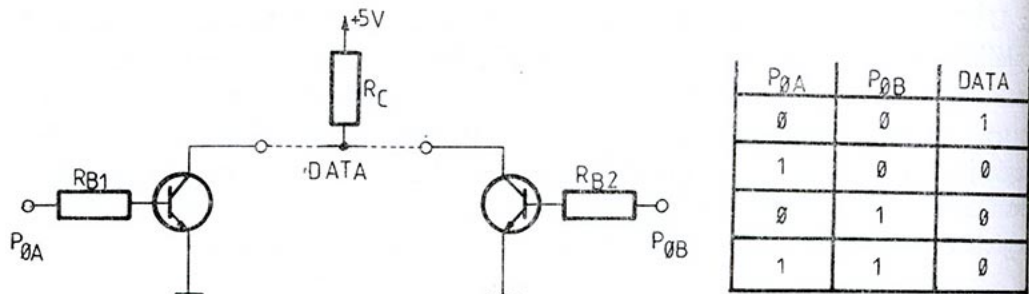
Az áramkör *soros buszra* csatlakozását célszerű jobban szemügyre venni, hiszen a nyitott kollektoros vonali meghajtást a gyakorlatban sokszor alkalmazzák. (Hálódóknak ajánljuk áttanulmányozásra a következő részt.) Nemcsak az itt ismertetett eljárással lehet a hasonló vonalakon információt továbbítani, de az általunk ajánlott mód egyszerű. Nézzük előbb az áramkört és az abból adódó lehetőségeket! (2.15. ábra.)



2.15. ábra. Nyitott kollektoros meghajtóval táplált soros vonal

Az adatátvitel lehetőségét a C 16-os, C Plus/4-es házi számítógépek vonalain vizsgáljuk! Bekapcsolás után a SERIAL-buszra kivezetett processzor port alsó 4 bitje kimenetként, felső 4 bitje bemenetként viselkedik.

A P0, P1 biteken lévő logikai 0 szint az invertáló meghajtók kimenetein 1 logikai állapotot eredményez. Ha a számítógépeket a 2.15. ábrán feltüntetett módon összekötjük, akkor a meghajtó IC-k közös (1 kOhmos) munkaellenállásokra dolgoznak.

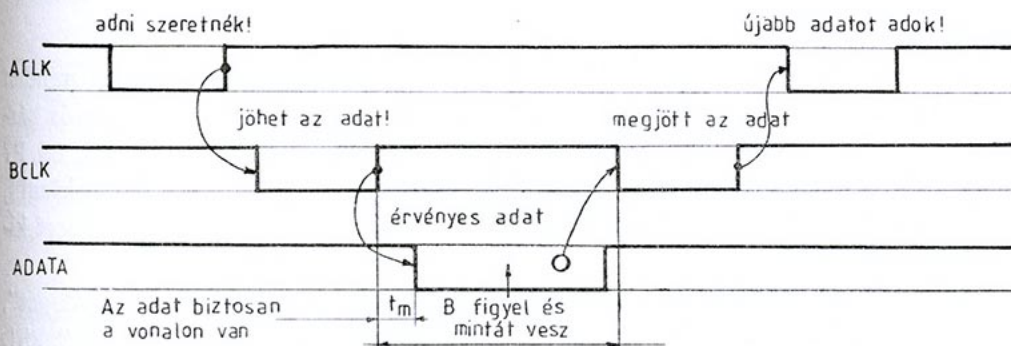


2.16. ábra. A nyitott kollektoros vonali meghajtók és működési táblázatuk

Ezért a szabadkollektoros meghajtás miatt a külső adatvonalak \emptyset állapota attól függ, hogy melyik gép húzza le a vonalat. A vonalak a meghajtók bemeneteiről nézve NOR [14.] kapcsolatban vannak.

Nézzük meg, hogyan küldhetnek egymásnak bitsorozatot a gépek! Mindkét gép a CLK (P6. bit) vonalat figyeli. Amelyik először lehúzza, az kíván a másiktól jelet adni. Ha a vonal már logikai \emptyset szinten van, akkor valamelyik adást kezdeményezett, így a másiktól vevő állapotba kell kerülni, és mindaddig, amíg a beérkező karakterkódok között nem jött az adás végét jelző, figyelnie kell a másik adását.

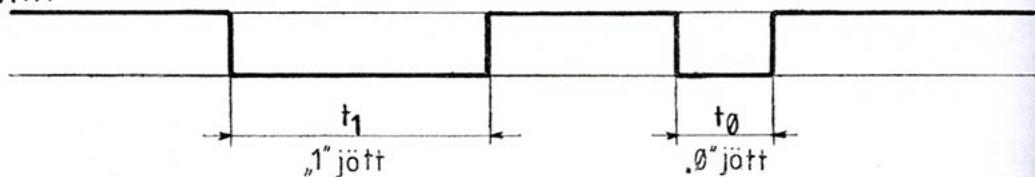
Tíh. az A gép ad, a B gép a vevő. Jelölje ACLK és ADATA az A, BCLK a B gép jeleit! (Lásd 2.17. ábrát is!)



2.17. ábra. „Kézfogásos” soros adatátvitel egy megoldási lehetősége

Az A adó egy rövid időre lehúzza az ACLK jelet, mivel a B vevő figyeli a P6 vonalat, érzékeli, hogy adás fog következni. Ezt a felismerését B a BCLK jel (ugyanazon a vonalon megy) rövid lehúzásával közli A-nak „Jöhét az adat!” Ezután A a DATA vonalra teszi az adatbitet (vagy megváltoztatja a DATA vonal állapotát, vagy nem). Ezt B mintavételezi, majd a BCLK ismételt lehúzásával közli A-nak: „Megjött az adat!” Ha újabb bit megy, A ACLK lehúzásával jelez: „Adni szeretnék!”, B BCLK-val visszajelzi: „Jöhét az adat!”. A gép az adatvonalra teszi az aktuális bitet, majd B mintavételezi a DATA vonal állapotát, s utána a CLK vonalon jelzi: „Az adat megérkezett!”. B gép A adásának megkezdésétől nyolcasával csoportosítja a biteket, majd bájtonként elraktározza. Mivel az egész folyamat a kézfogáshoz hasonlít (én nyújtom a kezem, te fogadod, de egyúttal te is nyújtsz és én is fogadom), az ilyen információközlési módot „kézfogásos” adattovábbításnak (*hand shake*) nevezik. Az adat azonosításának többféle módja képzelhető el. Lehet pl. úgy azonosítani a \emptyset vagy 1 logikai szinteket, hogy az 1-nek egy hosszabb, a \emptyset -nak egy rövidebb \emptyset szintű impulzus feleljen meg az adatvonalon (2.18. ábra), de ekkor a változást és időtartamát is figyelni kell. Adott esetben elegendő lehet az adat azonosítására, hogy fel-tételezünk egy időt a BCLK jel felfutásától, ami alatt az adat már biztosan meg-érkezik (t_m), (aktuális a DATA vonalon), majd mintavételezzük a DATA vonalat. (2.17. ábra.)

DATA



2.18. ábra. az „1” és a „0” állapot egy megkülönböztetési lehetősége egyetlen adatvonalon

Az adattovábbítás csak gépi kódból [12.], [2.] képzelhető el hatékonyan. Természetesen az egész folyamatot célszoftverek irányítják.

Térjünk vissza a fejezet elején ismertetett eszközünkhöz!

Mire használható?

- a) Konstrukciója olyan, hogy a SERIAL-portot teljes egészében megvédi minden külső káros behatástól, így még abszolút kezdő és figyelmetlen hardveres is károkozás nélkül garázdálkodhat segítségével a processzor port P0, P1, P2, P6, P7 vonalain! A vonalak állapotának kijelzésén túl azok negált állapotát is megjeleníti, ezért segítségével arról is szemléletes képet kapunk, mi van a processzorport kimeneti adatbitjein.

Az alkalmazások kezdetén, amíg nem ismerjük teljesen eszközünk „lelkivilágát”, nagyon vigyázzunk, mert azt SP6, SP7 bemenetek visszahatnak az SP0, PS1, SP2, SP0, SP1, SP2 kimenetekre! Ezért kezdetben az eszközt vagy csak bemenetként, vagy csak kimenetként alkalmazzuk! Számítógépünk hibás alkalmazás esetén sem megy tönkre, a kimenetre kapcsolt eszközök esetleg nem úgy működnek, ahogyan szeretnénk. Berendezésünk tehát C 16-os, C Plus/4-es gépünket védő jó demonstrációs eszköz.

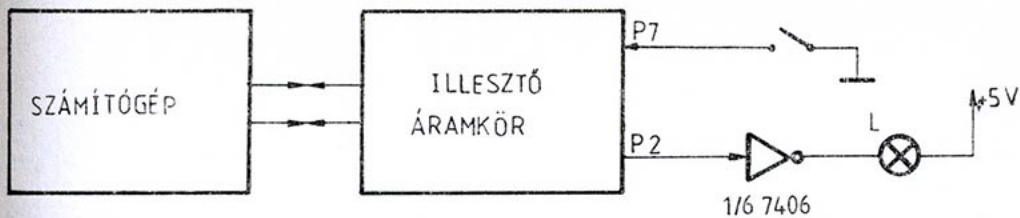
Az alábbi kis program figyelni az SP7 bit állapotát és, ha az 0, akkor SP2 bitet 0-ra kapcsolja (2.19. ábra.)

```

10 REM ** 7. mintaprogram **
11 :
12 :
15 POKE 0,15:POKE 1,200:REM processzor port init.
20 A=PEEK(1):REM adatregiszter beolvasása
30 IF A=72 THEN POKE1,PEEK(1) OR 4:REM izzó bekapcsolása
40 GET B$
50 IF B$="v" THEN PRINT "vége !":END
60 GOTO 20

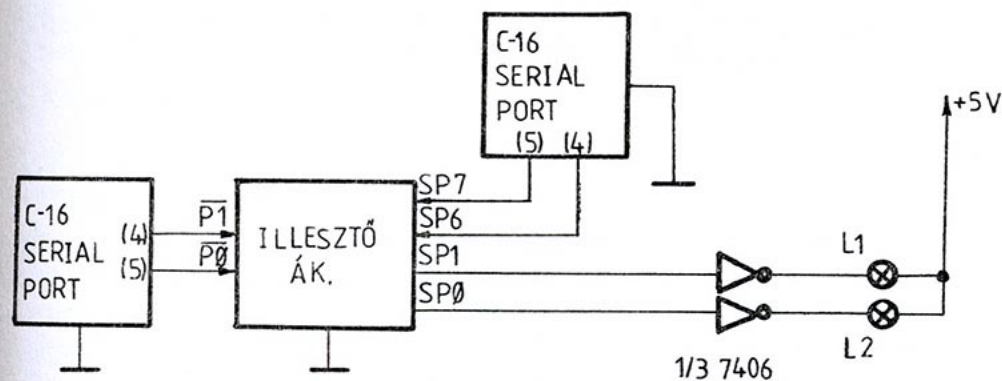
```

- b) Illesztő áramkörünk abból a szempontból már fejlettebb, hogy soros és párhuzamos adatátviteli eszközként egyaránt használható. Kimondottan „haladó” hardvereknek ajánljuk a 2.21. ábrán közölt kapcsolások megépítését, melyek mindegyike működtethető illesztőnkkel. Működésük könnyen megérthető, vagy a szakirodalomból áttanulmányozható [19.], [20.]. Ezért külön leírást ezekhez sem közlünk.

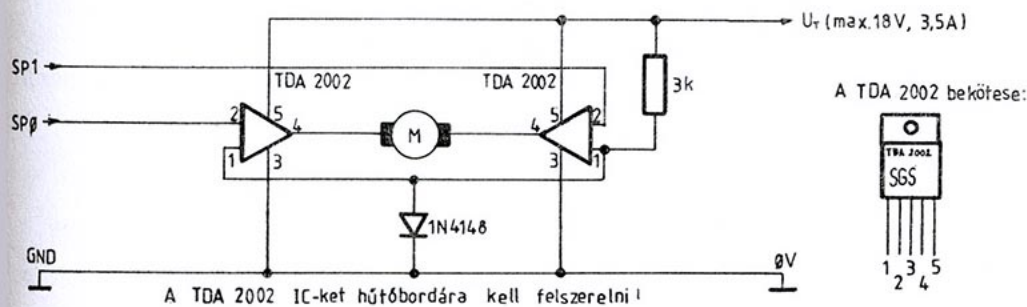


2.19. ábra. Izzó meghajtása illesztő áramkörünkkel [20]

- c) Észrevehetjük, hogy az SP6, SP7 bemeneteket az egyik, a $\overline{P0}$, $\overline{P1}$ bemeneteket egy másik C 16-os számítógépről vezérelve a két számítógép VAGY kapcsolatban vezérelheti az SP0, SP1 kimenetekre kötött eszközt. (Pl. bármelyik számítógép bekapcsolhatja ugyanazokat az izzókat. 2.20. ábra.)

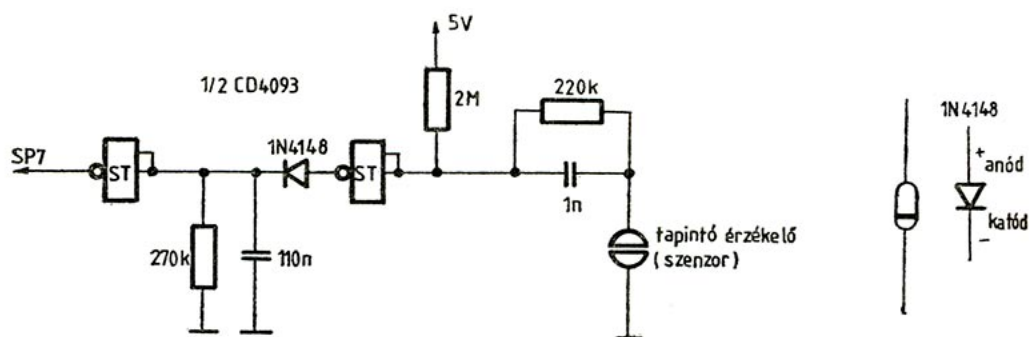


2.20. ábra. Két számítógép közös eszközt vezérel (VAGY kapcsolatban)

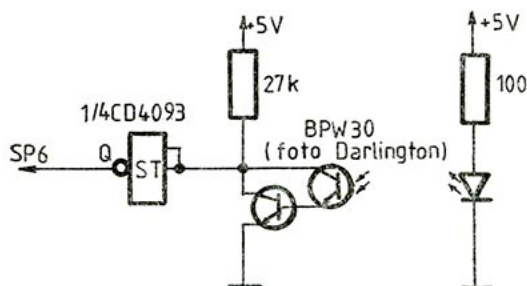


2.21. a ábra. Egyenáramú motor forgásirány megváltoztatási lehetőséggel

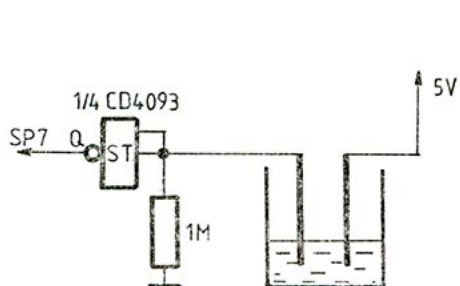
CMOS áramkörökkel rendkívül egyszerű érzékelők és jeladók építhetők, ezért javasoljuk a szakirodalom áttanulmányozását [34.]!



2.21. b ábra. „Brumm” ellen védett tapintóérezkelő [34.]

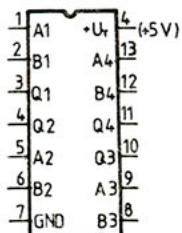


2.21. c ábra. Érzékeny és gyors optokapu [34.]



2.21. d ábra. Nedvesség érzékelése [34.]

A CD4093 négy kétbemenetű „NAND” SCHMITT trigger bekötése:



2.22. ábra. A CD4093 IC lábkiosztása

3. „RÁÜLÜNK” A BELSŐ BUSZRA

3.1. Mélyebben elemezzük a számítógép lelkivilágát

Az 1.9. ábrán a C 16 (C Plus/4) házi számítógép, mint rendszer felépítési vázlatát láthatjuk. Ezt figyelmesen szemlélve megállapíthatjuk, hogy tulajdonképpen 6 olyan fontos funkcionális eleme van eszközünknek, amelyek nélkül működése elképzelhetetlen lenne. Ezek:

1. A központi egység (MOS 8501 mikroprocesszor)
2. A rendszertár (ROM és RAM speciálisan kapcsolgatva)
3. A rendszer *beviteli* (Input) és *kiviteli* (Output) eszközei (I/O), (amelyek itt speciálisan, virtuális tárrekeszként kezelhetők)
4. Rendszercíműsín
5. Rendszeradatsín
6. Rendszervezérlősín

Aki már olvasott a C 16 rendszerről, az most megkérdezheti: „...és a TED chip?”. A kérdezőt megnyugtathatjuk. Rendszertechnikai szempontból — első közelítésben — a TED chip is egy memóriába ágyazott I/O eszköz, amely hardverből végez el olyan feladatokat, amelyeket egyébként szoftverből a mikroprocesszor is elvégezne. Mindazonáltal rendkívül hatékony, a konkrét rendszer működését döntően meghatározó BOÁK [3]. Most „csak” az a célunk, hogy a rendszersínre egy illesztő egységet tervezünk. Ehhez nem szükséges a TED-del foglalkozni!

Mint ahogyan azt már az 1.3. fejezetben megállapítottuk, a számítógép működésének lényege a kódfeldolgozás szempontjából a központi egység és a rendszertár összehangolt együttműködése. A központi egységnek 5 fontos feladata van:

1. Adatok (kódszók) beírása a memóriába
2. Adatok olvasása a memóriából
3. Adatok továbbítása (írása) a rendszer kiviteli eszközébe
4. Adatok olvasása (behozása) a rendszer beviteli eszközéből.
5. A beolvasott kódok által meghatározott belső regiszterműveletek elvégzése.

Az I/O eszközök virtuális tárrekeszként kezelése miatt a C 16-os számítógép rendszerében az 1. és a 3., valamint a 2. és a 4. funkciók azonosak, így leegyszerűsödnek a mikroprocesszor (hardver és szoftver) feladatai.

A címsínnak csupán egyetlen feladata van. A mikroprocesszor ennek vezetékein adja azt a kódot, amellyel kiválasztja az információáramlás útját. (Melyik eszköztől jöjjön, vagy melyik eszközbe menjen a feldolgozandó kód?) Természetesen az eszközválasztást egy dekódoló elektronika segíti (hasonló a 74LS138-hoz). A C 16-os számítógépnek 16 darab címvezetéke van és ezek, mint már említettük, kétállapotú jeleket adnak. Ezért $2^{16} = 65536$ információs út lehet. (Ez a 64 kbájtnyi memória egyes rekeszeinek a címe.) Ha egy újabb I/O eszközt szeretnénk a buszra ültetni, akkor ilyen dekódoló elektronikáról nekünk is gondoskodni kell! Mint az 1.3. fejezetből tudjuk, a címsínen az információ a processzortól a dekódoló eszköz felé egy irányban áramlik.

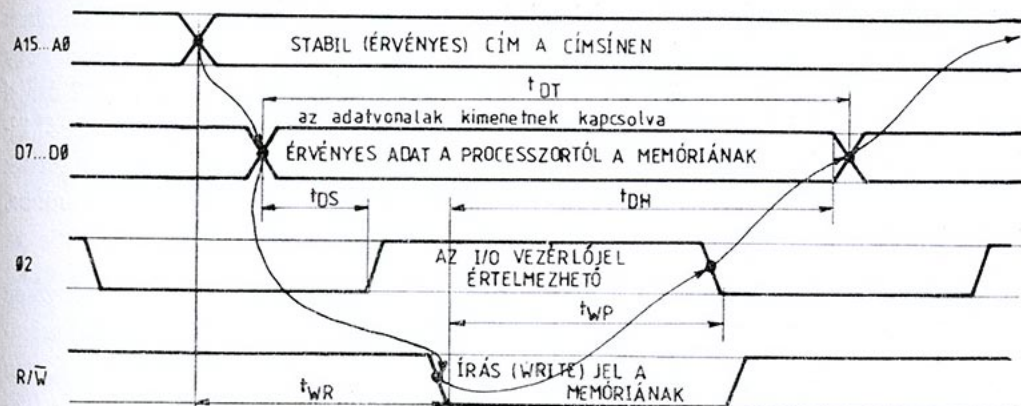
Az adatsín feladata, hogy a címmel kiválasztott eszköz (vagy memóriarekesz) és a processzor között kicserélendő információt továbbítsa. Az adatsín feladatait is egy elektronika valósítja meg. Az adatsín funkójából következik, hogy az elektronikának (egy vezérlőjel által beállíthatóan) kétirányú adatáramlást kell biztosítania. Természetesen egyszerre mindig csak az egyik irányba áramolhat információ. A C 16-os mikroprocesszorának 8 adatvezetéke van. Ezért, ha a belső adatsínhez kívülről hozzá akarunk férni, célszerű ilyen 8 bites, vezérlőjellel beállítható kétirányú sínmeghajtót nekünk is alkalmazni. Egyrészt védenünk kell a belső sítet az esetleges „szembekapcsolás” ellen, másrészt az ilyen sínmeghajtók 10–20 db TTL egységterhelés meghajtására is alkalmasak, s így komoly további külső berendezéseket kapcsolhatunk a rendszersínre.

Ha megnézzük a központi egység előzőekben leírt 5 fontos munkafeladatát, beláthatjuk, azok közül semelyik két esemény nem történhet meg egyszerre, azok közül bármelyik végrehajtása egy adott időben ki kell, hogy zárja a többit. Ennek biztosítása, valamint a jelek időbeni áramlásának pontos időzítése a rendszervezérlősín feladata. Ezek a mikroprocesszor olyan kivezetései, amelyeken a működés adott időpontjában TTL kompatibilis digitális jelek jelennek meg, parancsokat adva az adatáramlást biztosító elektronikáknak. Ilyen, a vezérlőjeleket értelmezni, parancsaikat végrehajtani képes elektronikáról is gondoskodnunk kell, ha a rendszer-sínre illeszteni akarunk egy áramkört.

Összefoglalva: szükségünk van egy címdekódoló hálózatra, egy kétirányú sínmeghajtóra és egy, a vezérlőjeleket értelmező hálózatra, ha a belső sínrendszerre egy eszközt szeretnénk illeszteni. A belső sinned kapcsolatot teremteni képes hálózatokat interfészeknek (vagy csatlakozó felületeknek) nevezzük.

A C 16-os számítógép központi egységének előzőekben felsorolt feladatai közül most nekünk csak kettőt, a memóriába írást és a memóriából olvasást kell elemeznünk ahhoz, hogy a siker reményében hozzáfoghassunk az interfész tervezéséhez, ugyanis a C 16 minden I/O eszközt memóriába ágyazva, virtuálisan kezel. Természetesen e folyamatoknak is vannak olyan, a C 16-ra jellemző „finomságai”, amelyek az I/O működés lényegének megértéséhez nem kellenek. (Pl. a dinamikus RAM multiplex címzése, a frissítő jelek időzítései, a rendszerbusz TED vezérlése stb.)

a) **Adatok beírása a tárba** (vagy a virtuális output eszközbe) A 3.1. ábrán egy tipikus memóriairás időzítési diagramját látjuk.



3.1. ábra. Egyszerűsített memóriairás időzítési diagram

A folyamat:

1. Egy utasítás (pl. STA \$F2) értelmezése után a mikroprocesszor megkezdí annak végrehajtását. Először *stabilizálja* a rendszer címsínén a címkódot. Ez azt jelenti, hogy az egyes címvonalakon nem történik változás mindaddig, amíg a memóriába írást a processzor végre nem hajtja.
2. Röviddel a címsín stabilizálása után a processzor ráteszi a rendszeradatsínre az *érvényes* (kiírandó) adatot, majd a küldendő kódnak megfelelően stabilizálja az adatsín állapotát.
3. Amikor már biztosan stabil az adat és a megcímzett memóriarekesz is kiválasztódott, egy R/\overline{W} vezérlőjellel jelzi a processzor a memóriának, hogy az érvényes adat megérkezett adatvezetésekre és azt megcímzett rekeszébe bekapuzhatja (beolvashatja). A C 16-os az I/O vezérlőjelek időzítésének pontosítására egy $\Phi 2$ *segédórakelet* is használ, amelyet egy dekódoló logika (MOS 7700-as IC) állít elő. Az I/O-t vezérlő jeleket az I/O eszközök csak akkor tekinthetik érvényesnek, ha a $\Phi 2$ logikai 1 szinten van (ez az aktív állapota). $\Phi 2$ nem azonos a rendszer $\Phi 0$ órajelével, amelyet a TED chip állít elő (furfangos módon!) a CLK IN jeltől.

A 3.1. ábrát tanulmányozva az interfésztervezésre vonatkozóan néhány fontos megállapítást kell tenni. A cím stabilizálódása (az írási folyamat megkezdése) és az R/\overline{W} aktív állapota között egy t_{WR} időtartam telik el. Ez a C 16-nál $\approx 500-600$ ns. Az interfész dekódoló logikájának tehát csak ennyi ideje van, hogy „megtalálja”, ill. felismerje a címezett eszközt. Ezért többlépcsős dekódoló hálózatnál vizsgálni szükséges a jelterjedési időket és az egyes szinteken összeadva az eredő időt kell figyelembe venni! A gyakorlatban ez az idő legtöbbször elég, ha TTL dekódoló áramköröket használunk, mert ezek átlagos jelterjedési ideje 10–12 ns. Több szempontból is más

a helyzet a CMOS áramkörökkel. Ezek dekódolásra használatát kezdőknek nem ajánljuk!

Az érvényes adat megjelenése azt jelenti, hogy az adatbuszon a jelvezetékeken nincs logikai átmenet, és nem azt, hogy a logikai szintek már stabilak. Ennek beállítására egy T_{DS} idő szükséges. Ez a C 16-nál ≈ 100 ns és megközelítően egybeesik a $\Phi 2$ órajel egyik felfutó élével. Ezt a T_{DS} időt ki kell várni, mert ha a címbusz tartalmából kikapuzott engedélyező jellel mintavételezzük az adatbuszt (azaz túl gyorsan jelenik meg a dekódolt jel az engedélyező bemeneten), akkor hibás adatokat olvashatunk be. Ezért is célszerű az I/O eszközknél a $\Phi 2$ és a R/\bar{W} jelet együtt figyelni.

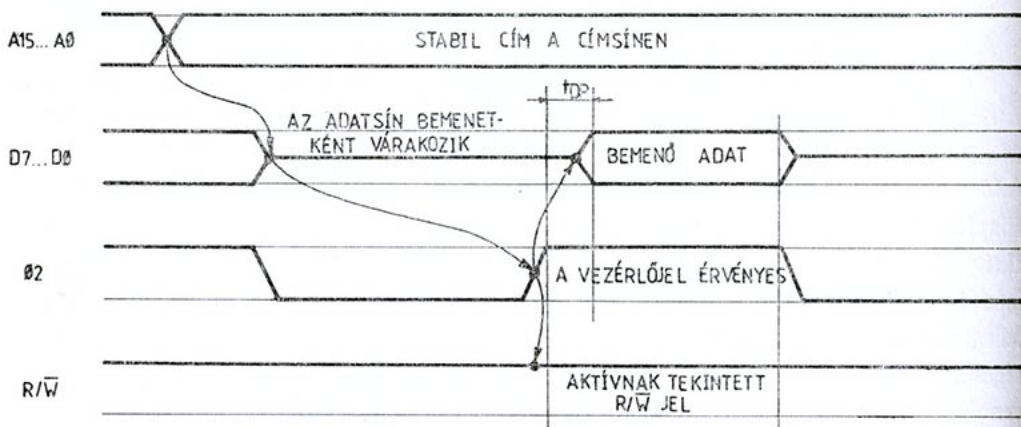
A processzor az R/\bar{W} kapuzó jel aktív állapotától számítva, még T_{DH} ideig érvényesen tartja az adatot a rendszeradatsínen. Az I/O eszköz (memória, vagy periféria) szempontjából azonban nem ennek, hanem a T_{WP} (C 16-nál ≈ 600 – 800 ns) időnek van jelentősége. Ui. ennyi ideje van, hogy az adatvonalaira érkező jeleket beolvassa (bekapuzza). Ez is elég szokott lenni, ha a $\Phi 2$ és az R/\bar{W} jeleket tovább már nem nagyon kódolgatjuk, mert a leglassúbb TTL eszközök bekapuzási ideje is 200 – 300 ns körül van.

b) Adatok beolvasása a tárból (vagy a virtuális inputeszközből)

Elemezzük a 3.2. ábrán látható beolvasási időzítési diagramot!

A folyamat:

1. Egy utasítás (pl. LDA \$FE00) értelmezése után a központi egység megkezdí annak végrehajtását. Először stabilizálja a címsínen a címkódot meghatározva, hogy melyik rekeszből kíván olvasni. Ez az adatút kijelölése.
2. Következő lépés az adatsín adatrányának beállítása. Külön kiemeljük, hogy olvasásnál a memóriától a processzor irányába áramlik az adat, és ez pontosan ellentétes az írási adatránnyal. A processzor adatvezetékeit bemeneteknek állítja (automatikusan és hasonlóan a processzor portnál leírtakhoz), majd az adatvonalakon várja a megérkező kódszót.



3.2. ábra. Egyszerűsített memóriaelolvasás időzítési diagram

3. R/\overline{W} jel magas szintje és a $\Phi 2$ segédjel magas szintje alatt a processzor engedélyezi a memóriának (vagy az I/O eszköznek), hogy adatait az adatsínre tegye és a t_{DP} idő eltelte után (C 16-nál $t_{DP} \approx 50-100$ ns) azonnal be is olvassa az adatokat. Ez az eljárás feltételezi, hogy a tárolvasás vagy az inputeszköz-olvasás megkezdésekor az adott bemeneten már stabilak az adatok. Az I/O adatok stabilitását adott esetben az I/O eszköz jelezni is tudja a CPU-nak (RDY-jellel) és az csak ezután kezdi az olvasási műveletet. A mikroszámítógépes rendszer ilyen várakozásos időzítési diagramja a C 16-nál bonyolult, megtárgyalása most nem feladatunk. [3.]

Az interfésztervezés szempontjából nagyon lényeges kiemelni, hogy *a tár vagy külső adatforrás adatait mindig el kell választani a rendszeradatsíntól*, ha azokat a mikroprocesszor nem kéri. Ezt maguk a memória IC-k vagy I/O megvalósítására alkalmas IC-k megteszik, mert ha engedélyező bemenetük nincs aktivizálva, akkor adatvonalait ún. nagyimpedanciás állapotba vezérlik, mintegy „levágják” magukat az adatsínról. A nagyimpedanciás állapot a vonalakon szakadást jelent! Ezek után fogjunk hozzá az interfésztervezéshez!

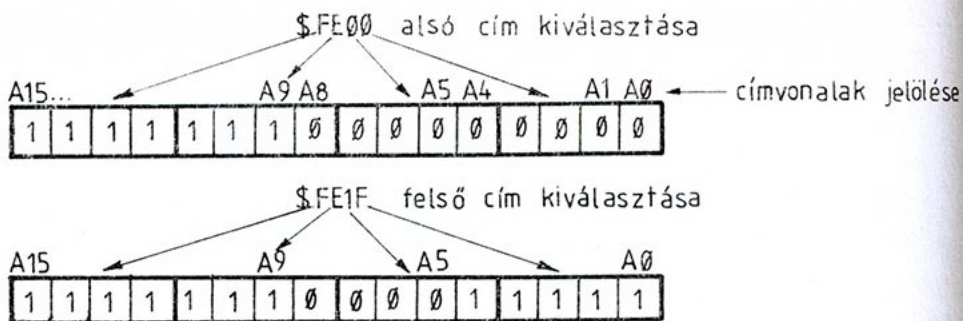
3.2. Egyszerű interfészt tervezünk

A C 16-os és a C Plus/4-es házi számítógépek címezhető memóriaterületen belüli I/O területe a $\$FD00$ -tól (64768) a $\$FEFF$ -ig (65279) terjed. A C 16 és a C Plus/4 I/O felosztása annyiban tér el egymástól, hogy a Plus 4-ben a $\$FD00$ — $\$FD0F$ (64768—64783) címeken egy soros aszinkron kommunikáció megvalósítására alkalmas interfész IC [3.], a $\$FD10$ — $\$FD1F$ (64784—64799) címeken pedig egy párhuzamos, kétirányú port található, amik a C 16-ban nincsenek. Mindkét számítógépben szabad a $\$FE00$ — $\$FEFF$ (65024—65279) terület (256 bájt), ha a belső sínrendszert nem használjuk pl. a VC 1551-es floppy driver a rendszersínre csatlakozik és 10 bájtot lefoglal $\$FE00$ -tól (65024) a kommunikációra).

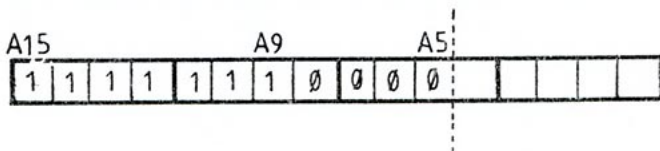
Az interfésztervezés intuitív feladat, de fő lépései megadhatók:

1. lépés: *Elemezzük a számítógép tártérképét és megállapítjuk az I/O terület helyét* ($\$FD00$ — $\$FEFF$). Az I/O területen belül megkeressük azt a részterületet, amelyiket egyetlen eszköz sem használ kommunikációra, s *kijelöljük a mi területünket* ($\$FE00$ — $\$FE1F$, 32 bájt)!
2. lépés: *Megnézzük, milyen címkombinációkkal lehet az adott tárterületet kiválasztani* (3.3. ábra)! *Meghatározzuk az \overline{EN} engedélyező jelhez tartozó címkombinációt!*

A kiválasztott területünkhöz tartozó alsó és felső címkombinációban megkeressük azt a részt, ami a címzés során nem változik. Ez lesz az adott területhez tartozó \overline{EN} engedélyező jel címkombinációja. Ugyanis, ha e kombináció megjelenik a címsínen a címkombináció részeként, akkor a címzés



Mindkét címnél változatlan címkombináció rész:



Az engedélyező jel
címkombinációja

3.3. ábra. Az engedélyező jelhez tartozó címkombináció meghatározása

kiválasztott területünkre mutat és csak akkor jelenik meg a címkombináció részeként a kombináció, ha címterületünket szólítja a CPU (központi egység).

3. lépés: *Megtervezük a címdekódoló hálózatot:* Erre mi két IC-t, egy 8 bemenetű kaput (74LS30) és egy 2×4 bemenetű NOR kapu (CD4002) egyik felét használjuk! (3.4. ábra)

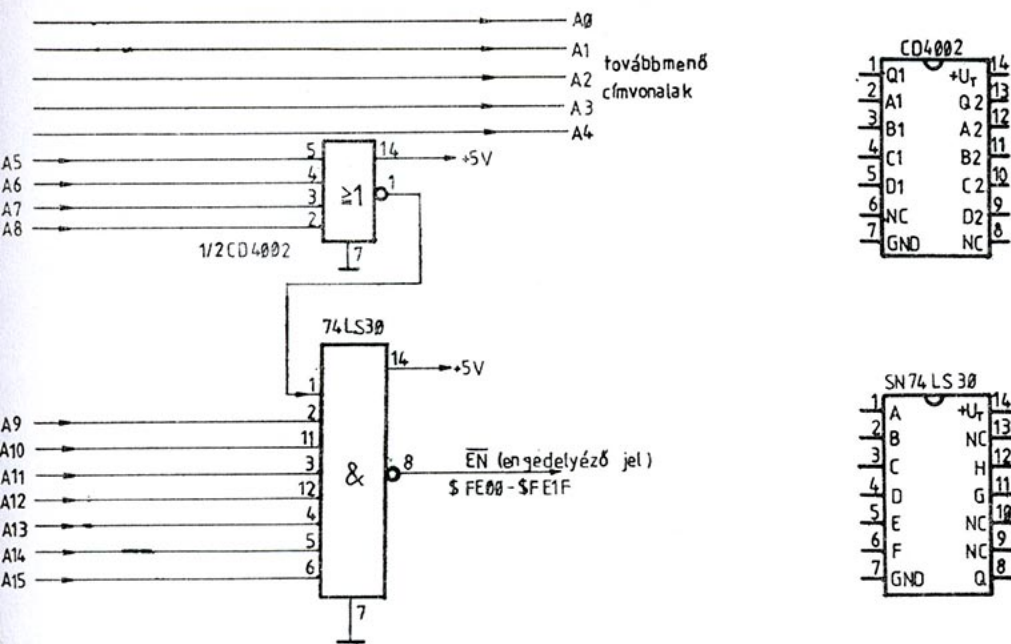
A NOR kapu kimenetén akkor és csak akkor lesz 1 logikai szint, ha az összes bemenetén 0 van. (Azaz az $A5=A6=A7=A8=0$.) A NAND kapu kimenetén pedig a 0 feszültségésszint akkor és csak akkor jelenik meg, ha összes bemenetén 1 szintű a jel.

Összevetve a 3.3. ábrán megjelölt kombinációval látható, hogy az \overline{EN} címterület-engedélyező jel akkor lesz 0, ha az \$FE00—\$FE1F közötti kombinációk valamelyike fennáll a címsínen.

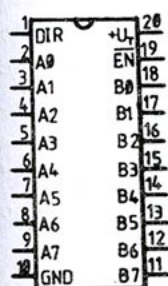
4. lépés: *Megtervezük az adatáramlást.*

Kiválasztunk egy 8 bites vezérelhető kétirányú sínmeghajtó IC-t és illesztjük az adatsínre. Vizsgáljuk meg az erre a célra kiválóan alkalmas SN74LS245 áramkört, melynek bekötése és működési táblázata a 3.5. ábrán látható!

Az áramkör kivezetései, ha az $\overline{EN}=1$, azaz nem a mi általunk kívánt címterület címződik, nagyimpedanciás állapotban vannak, így megvalósítják a leválasztást. Minden olyan esetben, amikor $\overline{EN}=0$ adat áramolhat



3.4. ábra. Címdekódoló hálózat az \$FE00- FE1F címterületre



MŰKÖDÉSI TÁBLÁZAT			
engedélyezés	adatirány vezérlés	adatutak	
		A oldal	B oldal
$\bar{0}$	$\bar{0}$	kimenet (output)	bemenet (input)
$\bar{0}$	1	bemenet (input)	kimenet (output)
1	X	nagyimpedanciás állapot	

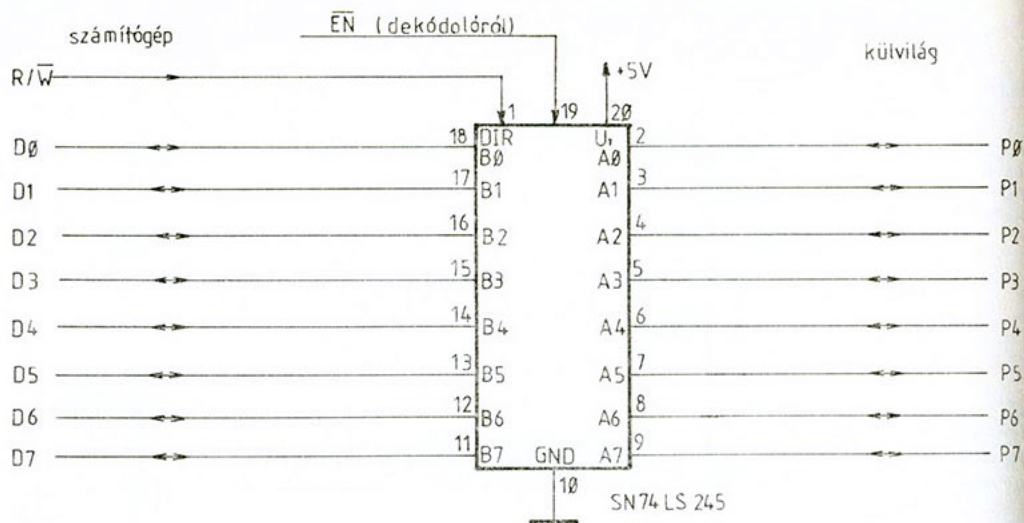
$t_r = 8$ ns (átlagos terjedési idő)
 $N = 15$ TTL egység terhelést képes meghajtani.

X: közömbös jelszint

3.5. ábra. Az SN74LS245 kétirányú sínmeghajtó jellemzői

az adatsínen. Az adatáramlás iránya a DIR vezérlőjeltől függ! A C 16-os számítógépben az adatáramlás irányát a R/\bar{W} (olvasás/írás; Read/Write) jel vezérli. Ezt a DIR kivezetéssel összekötve már az adatsín számítógépes vezérlését is megoldottuk! (Lásd 3.6. ábrát.)

5. lépés: *A továbbhaladó jelek kiválasztása, vonalaik meghajtásának tervezése.*
 Az A0, A1, A2, A3, A4 (összesen 5 db) címvonalakat tovább kell vinnünk, mert ezek segítségével lehetséges a \$FE00—\$FE1F címterületen 32 db eszköz kiválasztása ($2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 31 + 1$ a \$FE00-on). A vonalakat meg kell hajtani, mert a mikroprocesszor címvonalai csak 1 TTL egységterhelést tudnak táplálni, így ez nem lenne elegendő az interfészhez csatlakozó 1 m hosszú szalagkábel és az interfésszel illesztett eszköz címdekó-

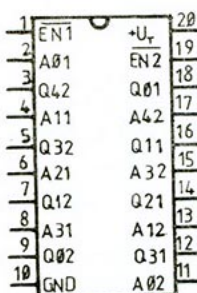


3.6. ábra. Az adatsín meghajtása, a sínmeghajtó vezérlése

doló áramkörének meghajtására. Az I/O perifériák pontos működéséhez szükséges a $\Phi 2$ jelet, az \overline{EN} jelet és a R/\overline{W} jelet meghajtva tovább kell vinni. Egyes perifériák alapállapotba hozásához a hardver RESET jel (\overline{RES}) is szükséges. Ezt is át kell vezetni a szalagkábelben, de nem szükséges külön meghajtani a \overline{RES} vonalat, mert ez a C 16-os számítógépnél 10 TTL egységterhelést bír el. A feladat megoldására alkalmas az SN74LS244 egyirányú sínmeghajtó. (Lásd 3.7. ábrát.)

Az interfész csatlakozóján célszerű lehetne a rendszer működését kívülről is befolyásolni képes jeleket behozni (pl. \overline{IRQ} , BA, AEC stb.). Ezek

SN74LS244 bekötése:



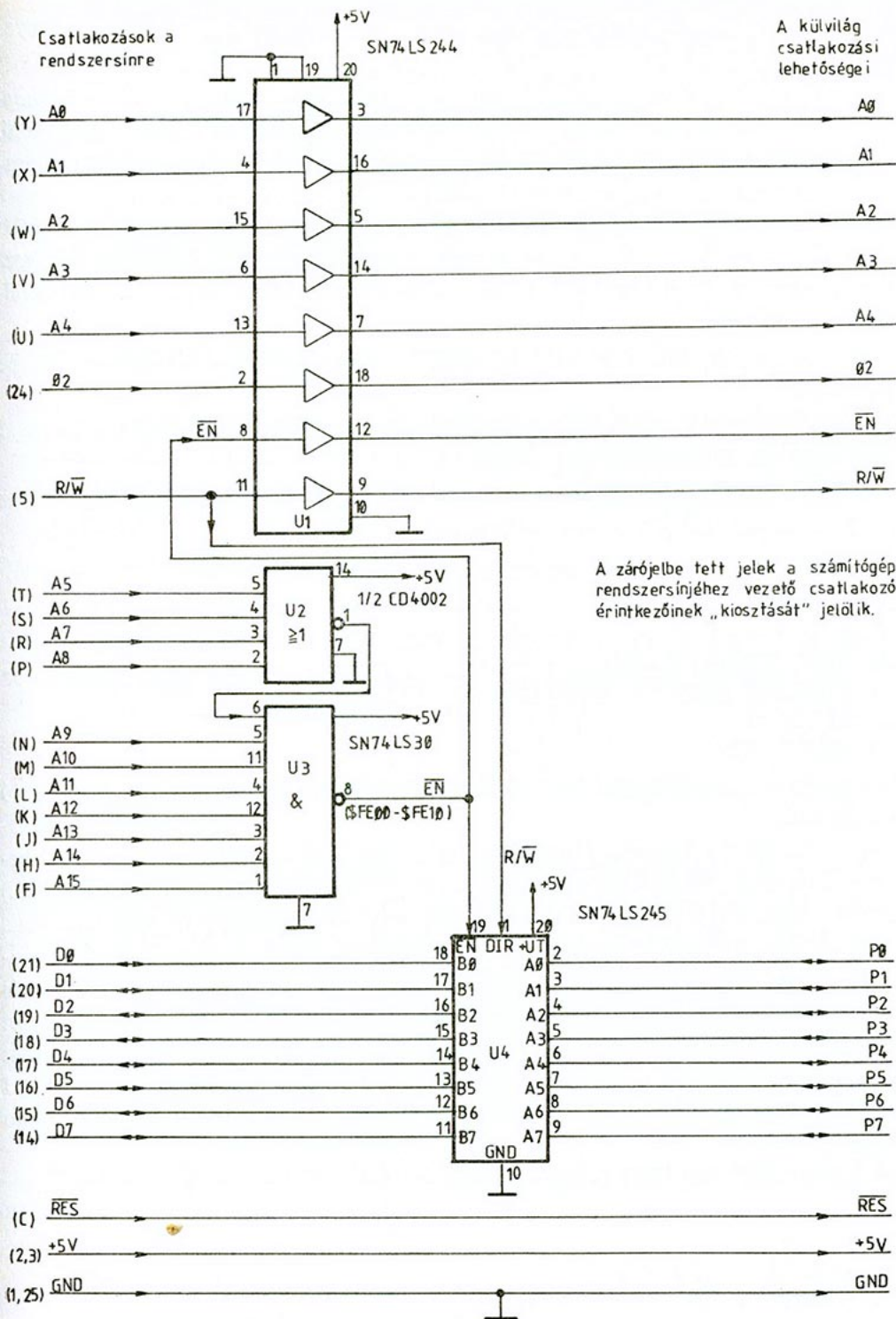
Az azonos indexű kivezetések tartoznak össze. A két négyesből álló meghajtónak két engedélyező jele van.

MŰKÖDÉSI TÁBLÁZAT		
engedélyező bemenet	meghajtandó vonal	meghajtott kimenet
\overline{EN}	A	Q
1	X	nagyimpedanciás
\emptyset	\emptyset	\emptyset
\emptyset	1	1

X: közömbös jelszint

A két négyvonalas meghajtó működése teljesen azonos.
 $t_p = 12\text{ns}$ (átlagos jelterjedési idő)
 $N=15$ TTL egységterhelést képes meghajtani

3.7. ábra. Az SN74LS244 egyirányú neminvertálható sínmeghajtó jellemzői



3.8. ábra. Egyszerű interfész az \$FE00-\$FE1F címtartományra

„lekezelése” a C 16-os számítógépben rendszertechnikailag is elég bonyolult, ezért nem képezheti e bevezető jellegű írás tartalmát.

6. lépés: *Az egyes megtervezett részekységek alapján összeállítjuk az interfész kapcsolási rajzát.* (Lásd 3.8. ábrát.) Ennek működése a leírtakból megérthető. A megépített eszköz NYÁK-hálózatát és fényképét a melléklet tartalmazza.

Első interfészünket célszerű úgy elkészíteni, hogy az integrált áramköröket foglalatba helyezzük. Ennek kettős előnye van. Egyrészt az eszköz élesztését fokozatosan végezhetjük, másrészt, ha a használat során valamelyik IC tönkremegy, könnyen cserélhetjük.

Először a címdekódoló hálózatot helyezzük üzembe az U2, U3, IC-k foglalatba helyezésével.

AZ INTERFÉSZT A SZÁMÍTÓGÉP KIKAPCSOLT ÁLLAPOTÁBAN CSATLAKOZTASSUK A RENDSZERSÍNRE!

Az alábbi kis program futtatása közben vizsgáljuk meg Logi-teszterünkkel az U3 IC 8. lábán ($\overline{E_n}$ jel) a feszültség szintet!

```
10 REM ** B. mintaprogram **
11 :
12 :
15 FOR I=0 TO 40
20 POKE 65024+I,255:PRINT I:FOR Z=0 TO 500:NEXT Z
30 NEXT I
40 GOTO 15
```

Ha a teszteren a digitális pont $I=0$ és $I=31$ értékek között villog, akkor a dekódoló jól működik.

Ezután a C 16-os kikapcsolt állapotában helyezzük foglalatba az U4 IC-t, majd bekapcsolás után a Logi-teszterrel figyeljük a $P_0...P_7$ kimenő vonalakat! (Sínmeghajtó.) A POKE 65024,255 parancs kiadásakor mindegyik P_i vonalon egy villanással jelzi a teszter, hogy az adat kimenet. A POKE 65024,3 parancsnál csak a P_0 és P_1 vonalakon szabad felvillannia. Nagyon vigyázzunk POKE parancs kiadásakor! Mérés közben véletlenül se zárjuk földpotenciálra a kimenő P_i vonalakat! Más a helyzet PEEK parancs esetében. A következő kis program $I=0$ és $I=31$ értékei között minden I -re 254-et kell, hogy beolvasson, ha a P_0 vonalat földpotenciálra kötjük, I minden más értékére pedig „össze-vissza értékeket” az adatvonalak határozatlan állapota miatt.

```
10 REM ** 9. mintaprogram **
11 :
12 :
15 FOR I=0 TO 60
20 PRINT PEEK(65024+I)
30 NEXT I
40 GOTO 15
```

Végezetül az U1 IC-t foglalatba helyezve elkészültünk első interfészünkkel. (Címsín és vezérlősín meghajtó.)

Interfészünk nem I/O periféria! Csak arra alkalmas, hogy egy eszközt a rendszeradat-sínhez illesszen (illetve leválssa azt). Nem végez átmeneti adattárolást, időzítést, vezérlést, adatátalakítást stb. Mint a későbbiekben látni fogjuk, segítségével mégis nagyon sokféle periféria ültethető az adatsínre. Eszközünk jelei a C 16-os számítógép belső rendszersínjén változó jelekkel szinkron változnak. A jelek időzítéséről, összehangolásáról maga a számítógép gondoskodik.

3.3. Az I/O eszközt is az interfész választja ki

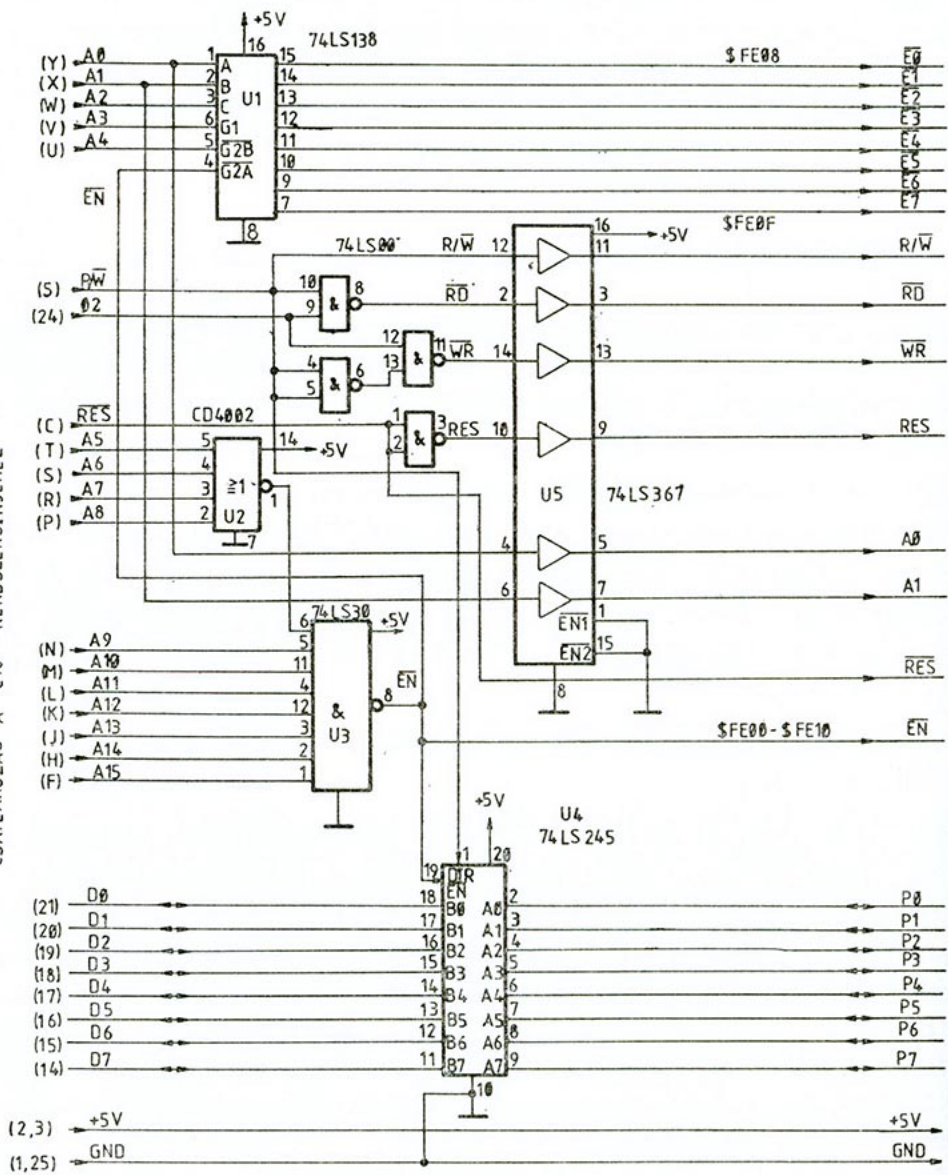
Sok esetben célszerűbb megoldás, ha az I/O berendezések aktivizálásához szükséges engedélyező jeleket is az interfész áramkör részeként, a rendszersínre csatlakoztatott NYÁK-lapon állítjuk elő. (Pl. egyszerűbb lesz az illeszthető I/O eszköz.) Egy ilyen elrendezést láthatunk a 3.9. ábrán.

Vizsgáljuk meg a kapcsolási rajzot!

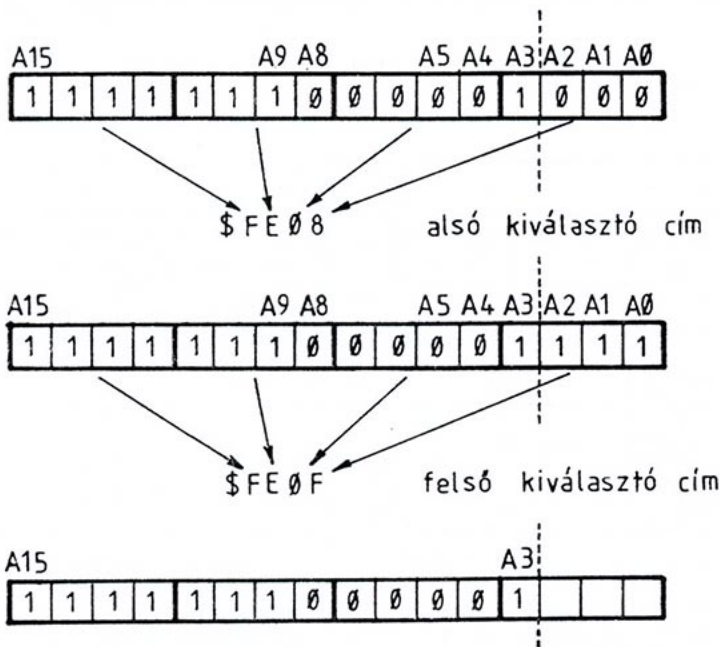
Az SN74LS138 demultiplexer most része a címdekódó hálózatnak. Ugyanis az \overline{E}_i vonalak valamelyike akkor és csak akkor választódik ki, ha az $\overline{E}_N = 0$ és $A_4 = 0$ $A_3 = 1$, mert ezeket a jeleket rávezettük az U1 IC engedélyező logikájára. Az alsó cím, amelynél az $\overline{E}_0 = 0$ (aktív) $\$FE08$, a felső cím pedig amelynél az $\overline{E}_7 = 0$ (aktív) $\$FE0F$. (3.10. ábra.)

Az áramkör adatsínmeghajtó része és működési elve az előzőekben tárgyalt interfészével teljesen azonos.

Bizonyos I/O egységek (pl. az I 8255 PPI) külön engedélyező jeleket követelnek az olvasáshoz (Read, \overline{RD}) és az íráshoz (Write, \overline{WR}). Ezeket a R/\overline{W} és a Φ_2 jelekből állítjuk elő egy SN74LS00 áramkörrel. Mivel a C 16-os és C Plus/4-es $\Phi_2 = 1$ esetén tekinti érvényesnek a R/\overline{W} jel írást vagy olvasást engedélyező állapotát, ezért lehetséges a R/\overline{W} jelet a Φ_2 -vel „kettéválasztani”. Az említett I 8255 a hardver RESET jelet is ponáltan (RES) fogadja, ezért a 74LS00 kimaradó kapuját a RES jel előállítására használtuk fel. A R/\overline{W} , \overline{RD} , \overline{WR} , RES, A_0 , A_1 továbbmenő jelek megfelelő szintjét az SN74LS367 típusú egyirányú sínmeghajtóval biztosítjuk, melynek működése az előzőekben tárgyalt 74LS244-hez hasonlít. Bekötése a 3.9. ábrán közölt kapcsolási rajzról leolvasható. Az eszköz NYÁK-hálózatát és fényképét a mellékletben lehet megtalálni.



3.9. ábra. I/O eszközök engedélyező jeleinek dekódolása az interfész áramkörben



Az engedélyezéshez szükséges címkombináció (a kiválasztási határok közötti állandó címrész)

3.10. ábra. I/O engedélyezést is végző interfész címkódolása

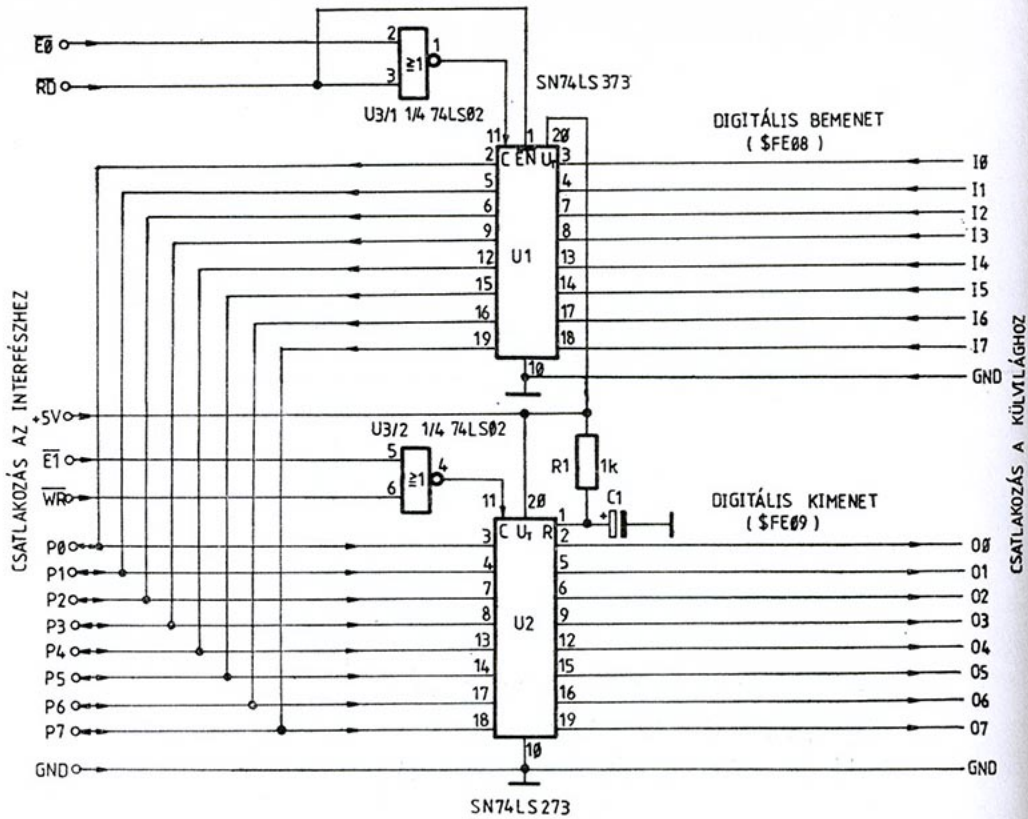
3.4. I/O áramkörök a rendszersínen

Mire használható az interfészünk?

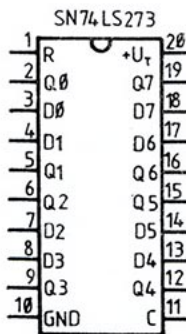
Például segítségével az engedélyezett címterületre helyezhetünk egy digitális I/O eszközt. Tekintsük a 3.11. ábrán szereplő kapcsolást!

A kapcsolat működésének megértéséhez először az integrált áramkörök működését és szerepét kell megérteni.

A 3.12. ábrán láthatjuk az SN74LS273 típusjelű IC bekötését és működési táblázatát. Az áramkör 8 bites latch (D-típusú tároló regiszter [5.]), Tp ([19.] totem-pole, „totem oszlop”) kimenetekkel [14.]. Ha \bar{R} törlő bemenetén logikai 0 szint van, akkor függetlenül attól, hogy milyen a többi bemenet, az összes kimeneten 0 szint jelenik meg. Ezt használjuk ki a kimenetek alapállapotba ($Q_i = 0$) hozásához. Az \bar{R} bemenet a C1 elektrolit kondenzátor pozitív pólusára kapcsolódik. C1 a bekapcsolás után az R1 ellenálláson keresztül elkezd töltődni. Mivel töltődése exponenciális, így az \bar{R} bemenet feszültsége sokkal később éri el a komparálási szintet, mint az áramkör többi része a működési állapotnak megfelelő feszültség szinteket. Ez a késleltetés ($\approx 0,5$ ms) az áramkör (be-) kapcsolási idejéhez mérten ($t_p \approx 18$ ns) nagyon hosszú,



3.11. ábra. Digitális input és digitális output periféria

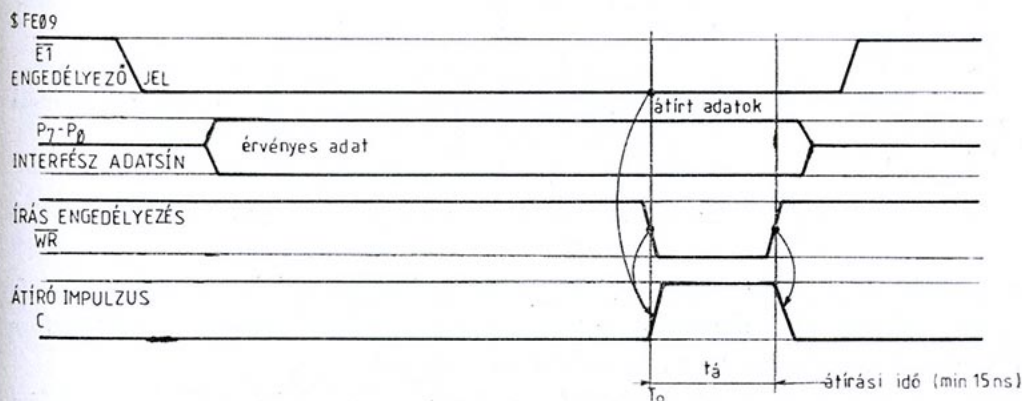


MŰKÖDÉSI TÁBLÁZAT				
BEMENETEK		KIMENETEK		
törlő	átíró	adat	adat	
R	C	D _i	Q _i	
0	X	X	0	← Törlési állapot
1	↑	1	1	
1	↑	0	0	} A bemeneti érték átírása a kimenetre
1	0	X	Q _u	

3.12. ábra. Az SN74LS273 8 bites D regiszter és működési táblázata

s elegendő a kimenetek törléséhez (0-ra állításához). Fixen bekapcsolt állapotban R bemenetet R1 (= 1 KOhm) stabil 1 szinten tartja, így a D tároló működése engedélyezett. Ha a C bemenetre egy 0 → 1 logikai átmenet, egy felfutó élű impulzus kerül, akkor a Di bemeneteken pillanatnyilag jelen lévő szinteket a tároló átírja a kimenetekre. C = 0 esetén pedig reteszelve tartja a kimeneteken az utoljára átírt Qu értéke-

ket. C átírójel (\uparrow) előállításához felhasználjuk az \overline{EI} eszközválasztó jelet és a \overline{WR} írás-engedélyező jelet. U3/2 NOR kapu kimenetén akkor és csak akkor lesz 1 logikai szint, ha mindkét bemenetén 0 van. Az átírás időzítése a 3.13. ábráról követhető

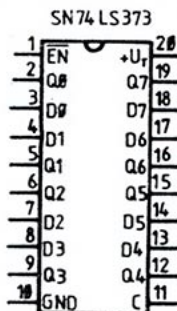


3.13. ábra. A digitális kimenet időzítési diagramja 74LS273-hoz (az időlépték torzított)

(vessük össze a 3.1. ábrán látható egyszerűsített memóriairási időzítési diagrammal!). \overline{EI} a rendszercímsín tartalmából dekódolt jel. Mivel csak 3 IC-n ($t_p \approx 100$ ns) és a szalagkábelén keresztül jut a digitális kimenet logikájához, ezért ez a jel lesz először aktív. E folyamattal kb. egy időben az interfész kapcsolatot teremt a rendszeradatsínnel, majd megjelenik az SN74LS273 bemenetein az aktuális adat. \overline{WR} a $\Phi 2$ és R/\overline{W} jelekből kikapuzott beírójel, így jól időzíthető jelzi a digitális outputként használt áramkörnek, hogy bemeneteire érvényes és mintavételezendő adat jött. Ebben a T_0 időpillanatban U3/2 0-ból 1-be billen és szolgáltatja U2-nek az átíró impulzust ($0 \rightarrow 1, \uparrow$). Természetesen U2 Di bemeneteiről átírja Qi kimeneteire az adatsín tartalmát. $\overline{WR} = 1$ esetén $C = 0$ lesz, ezért az SN74LS273 mindaddig őrzi kimenetein az adatsín utolsó, számára küldött jelkombinációját, amíg a rendszer ismét meg nem szólítja és más adatot nem küld neki. Egy POKE 65033,195 parancs után a digitális kimeneten az alábbi kimeneti kombináció (kódszó tartalom) jelenik meg: (Ellenőrizzük a logi-teszterrel!)

07	06	05	04	03	02	01	00
1	1	0	0	0	0	1	1

A kimenőjelek TTL kompatibilisek és bármilyen digitális elektronika kapcsolására alkalmasak [34.], [14.]. (Felhívjuk a figyelmet arra, hogy a kimeneti meghajtók áramköröit nem szabad közvetlenül a tápfeszültségre vagy földpotenciálra kötni! [19.]) Digitális bemenetként egy hasonló IC-t használunk (az SN74LS373 típusút). (3.14. ábra.) Ha az IC kimenetvezérlő bemenete $\overline{EN} = 1$, akkor kimenetei az interfész sínrendszere felől nézve szakadást mutatnak, az áramkör mintegy „levágja magát”



MŰKÖDÉSI TÁBLÁZAT			
BEMENETEK			KIMENETEK
kimenet-vezérlő	átíró	adat	adat
EN	C	D_i	Q_i
1	X	X	nagyimpedancia állapot
0	1	1	1
0	1	0	0
0	0	X	Q_u

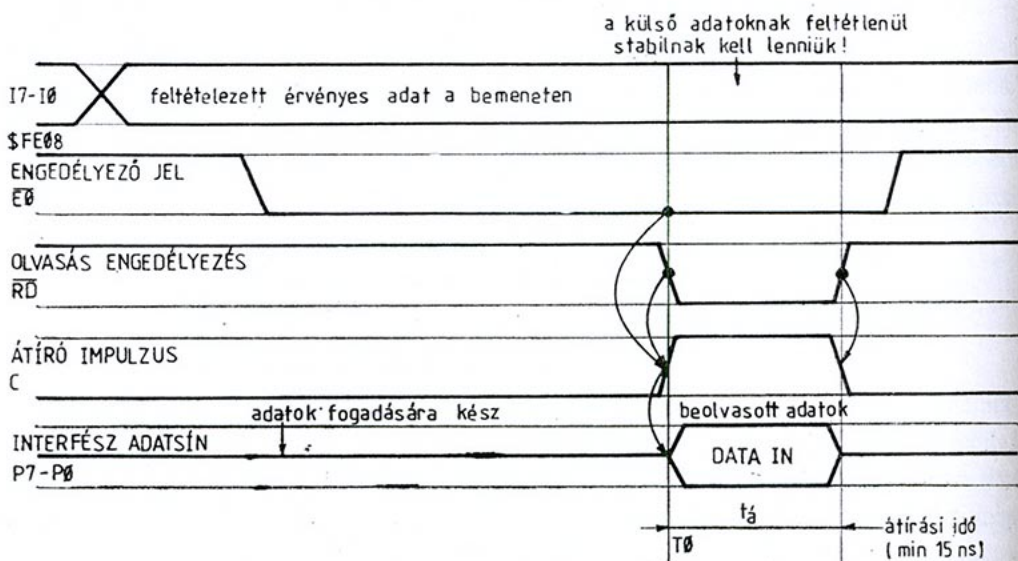
← TS (tree state) állapot
 A bemeneti értékek átírása a kimenetre
 ← Az átírt értékek őrzése

3.14. ábra. Az SN74LS373 8 bites TS kimenetű D regiszter és működési táblázata

az adatsínről. E miatt lehet digitális bemenetként használni, ugyanis a külvilág adatait, változásait mindaddig elválasztja az adatsíntől, ameddig adatközlésre ($\overline{EN}=0$) a számítógép (vagy interfész) fel nem szólította.

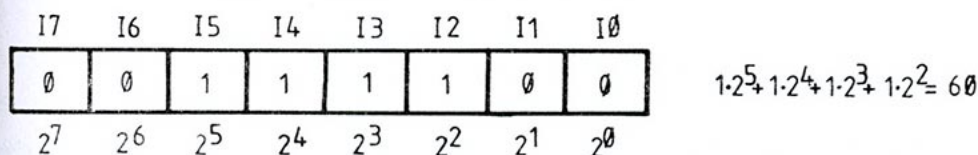
Vizsgáljuk meg digitális bemenetünk (input) működését a 3.15. ábrán látható időzí-tési diagram alapján!

A bemeneti adatok olvasását csak azután kezdjük meg, miután feltételezzük azok érvényességét. (Eszközünk a bemenet állandósulását nem tudja jelezni a központi egységnek!) Egy adatbeolvasó parancs (pl. PRINT PEEK(65032)) nyomán először az $\overline{E0}=0$ állapot áll be (lásd a címzésnél és a címdekódolásnál leírtakat), majd ezzel kb. egy időben a mikroprocesszor előkészíti adatok fogadására a rendszeradatsínt, ill. az interfészadatsínt. Az \overline{RD} olvasójel a $\Phi 2$ és R/\overline{W} jelekből lett kikapuzva, így az $\overline{E0}$ jellel együtt alkalmas az SN74LS373-at bekapuzási állapotba hozni. A $\overline{RD}=0$ a közvetlen összekötés miatt $\overline{EN}=0$ állapotot idéz elő rákapcsolva a D regiszter ki-



3.15. ábra. A digitális bemenet időzí-tési diagramja 74LS373-hoz (az időlépték torzított)

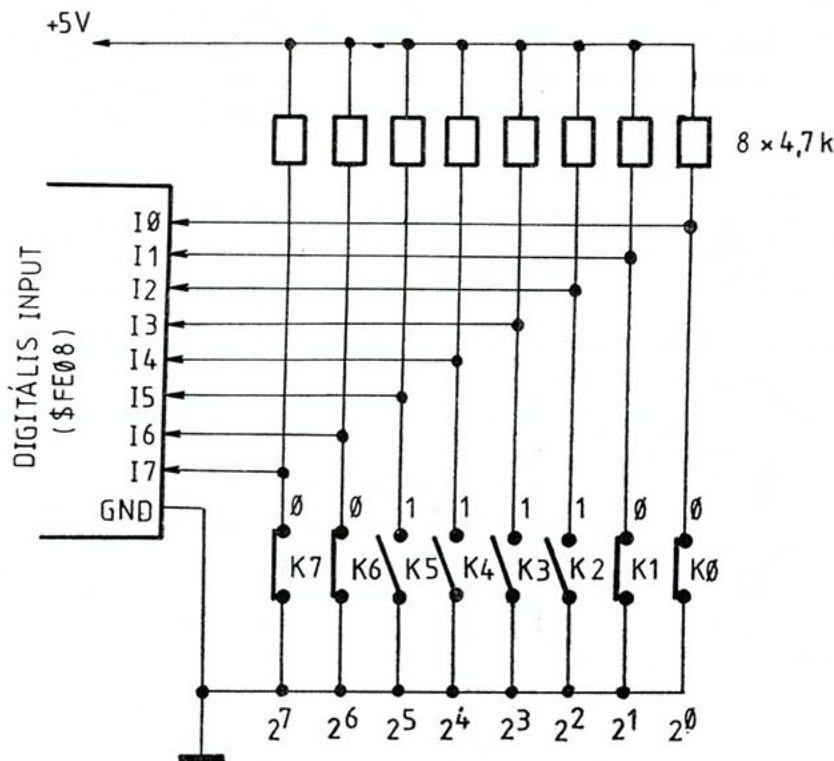
meneti meghajtóit az adatsínre, majd 10–20 ns múlva az U3/1 IC-n $E0=0$ -val egy $0 \rightarrow 1$ átmenetet létrehozva az interfészen keresztül a C 16-os belső adatsíjére teszi az adatot. Mivel \overline{RD} csak a processzor olvasási fázisában aktív, ezért az adatsínen megjelenő bitkombinációt a központi egység mindjárt be is olvassa. A 3.16. ábrán



vázolt módon kapcsolókkal beállítva az bemeneti kombinációt, a PRINT PEEK(65032) parancs után a 60-as szám jelenik meg a képernyőre írva.

A 3.11. ábrán bemutatott kapcsolás tehát már egy „igazi kis I/O berendezés”. Az I/O eszközöktől általában az alábbi feladatok részbeni ellátását szokták megkövetelni:

- adatok átmeneti tárolása, továbbítása
- adatátalakítás
- időzítések, vezérlések
- adattárolás, adatrögzítés
- szintillesztés, vonali meghajtás.



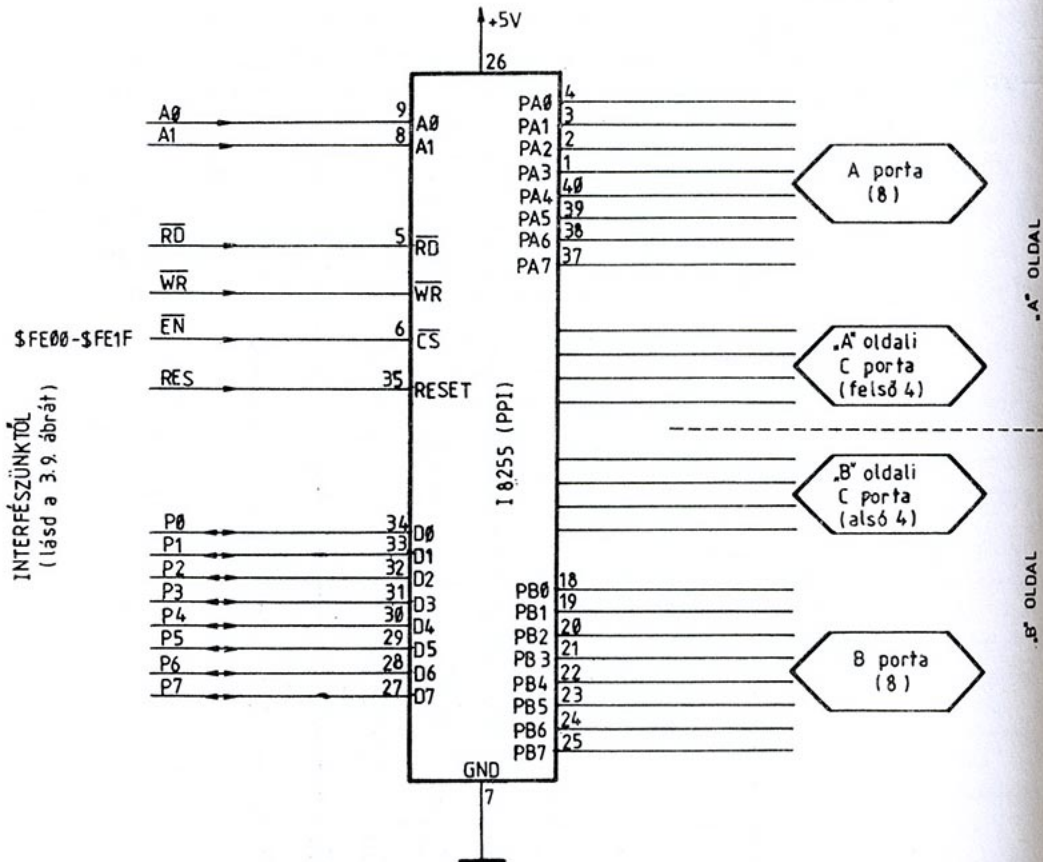
3.16. ábra. 80386 a digitális inputon (kapcsolós input periféria)

A mi kis perifériaillesztőnk az interfésszel összekapcsolva képes adatok átmeneti tárolására, továbbítására, időzítésekre, vezérlésekre. Így minden digitális változás figyelhető a segítségével, ill. digitális jelekkel kapcsolható berendezéseket tud működtetni. Az eszköz ki- és bemenetei egyaránt TTL kompatibilisek!

3.5. Programozható perifériaillesztő (I 8255) a rendszersínen

Aki nem elégszik meg az előzőekben vázolt digitális I/O egyszerű szolgáltatásaival, az professzionális, programozható I/O berendezést is „ráültethet” kis C 16-os házi számítógépe rendszersínjére, megsokszorozva ezzel külvilághoz kapcsolódásának lehetőségeit. Mint látni fogjuk, hardver szempontból — a 3.3. fejezetben megtárgyalt interfész segítségével — könnyű a dolgunk (lásd 3.17. ábrát), az I 8255 különböző üzemmódjainak programozása azonban eléggé nehéz, ill. összetett.

Mivel előrelátóan terveztük interfészünket, az I 8255 illesztése csupán a megfelelő vezetékek összekötését jelenti.



3.17. ábra. Programozható perifériavezérlő (PPI) a C16-os rendszerében (a 3.9. ábrán bemutatott interfésszel illesztve)

Az I 8255 programozásának ismerete nagyon fontos. Egyrészt azért, mert az olcsón beszerezhető technoMIR (moduláris interfészrendszer) egyik lényeges eleme, s e perifériavezérlő ismeretét feltételezi egyes egységeinek (A/D, PROTO-BOARD stb.) használata. Másrészt az I 8255 segítségével a perifériák széles skálája illeszthető a C 16-hoz.

Fentiek miatt ismertetjük az I 8255 felépítését, működését és \emptyset -ás (alap) üzemmódjának programozását.

Az I 8255 áramkört eredetileg az I 8080 mikroprocesszoros rendszerek I/O perifériakezelő eszközének fejlesztették ki. Szolgáltatásai azonban olyan sokrétűek, hogy alkalmazása ma már más rendszerekben is megszokott. 40 lábú DIP tokozásban [19.], [20.] (lásd 3.18.b ábrát) forgalmazott nagy bonyolultságú (LSI) áramkör, a rendszersínen a párhuzamos port szerepét látja el. A C 16 rendszeréhez közvetlenül nem illeszkedik, ezért rendszersínire ültetését külön áramkörrel kell biztosítani (lásd 3.9. ábrán az interfészt).

Az áramkör kimeneti oldala rendszertechnikai szempontból két félre, az „A” oldalra és a „B” oldalra osztható. Ennek akkor van jelentősége, amikor speciális üzemmódválasztás után a PC port felső 4 bitje inkább az A porthoz („A” oldalhoz), alsó 4 bitje pedig inkább a B porthoz („B” oldalhoz) tartozik. (Lásd 3.18.a ábrát.) Ebben az üzemmódban a PC port vezetékai vezérlő funkciókat látnak el az I/O perifériák irányába. Az „inkább” szó nem a határozatlan megfogalmazás következménye, hanem arra utal, hogy az eszköz felépítése nagyon moduláris és programozásától függően az egyes üzemmódoknál egymástól mintegy „kölcön vesznek” bizonyos jelvezetéseket az egyes porták.

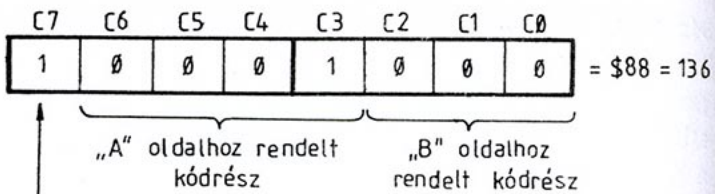
Az I 8255 mindhárom portja lehet kimenet és/vagy bemenet. Az „A” és a „B” port kimenetei és/vagy bemenetei egyaránt reteszelték, míg a „C” porta esetében csak a kimeneti állapotban reteszeli az eszköz az adatokat, a bemeneti állapotban nem.

A PPI üzemmódjainak kialakítását, belső rendszerszerveződését teljes egészében a *parancsregiszterbe (CR)* írt kód határozza meg. Külön *adatregiszterek* tartoznak az egyes portákhoz is. A 4 db regiszter címzése a 3.19. ábrán látható a rendszer alapfunkcióival együtt.

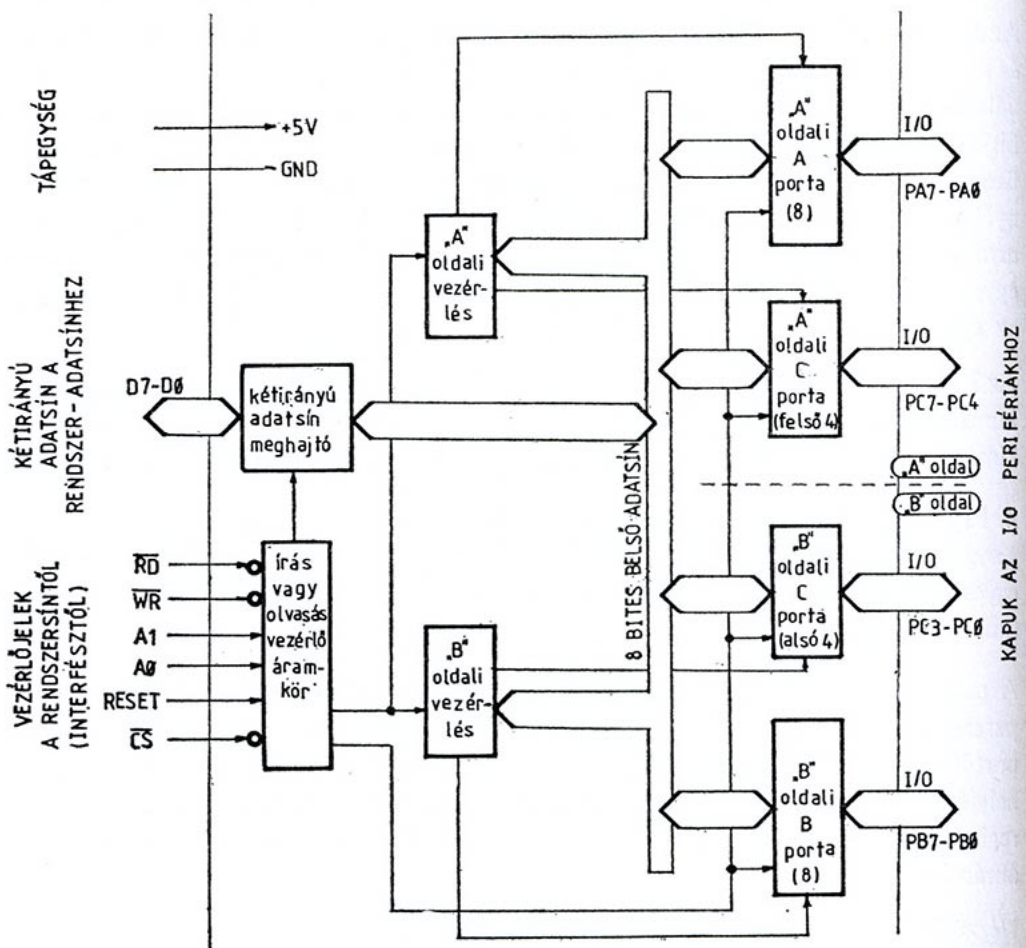
A parancsregiszter kiértékelését az I 8255 attól függően végzi el, hogy mi van a parancsregiszter C7., ún. legnagyobb helyiértékű bitjén. Ha itt \emptyset található, akkor a C port bit beírasi/törlési módja szerint értékeli, ha pedig 1 a parancsregiszter legnagyobb helyiértékű bitje, akkor a regisztertartalmat üzemmód-kijelölésként elemzi. A parancsregiszter programozása, ha a kiértékelés üzemmódválasztás szerint történik, a 3.20. ábrán látható.

Nézzünk két példát a parancsregiszter beállítására:

CR \$FE03 = 65027



A parancsregiszter tartalmát üzemmód-beállító parancsként értékeli a PPI



3.18. a ábra. Az I 8255 programozható perifériavezérlő tömbvázlata

1	PA3	PA4	40
2	PA2	PA5	39
3	PA1	PA6	38
4	PA0	PA7	37
5	R0	WR	36
6	CS	RESET	35
7	GND	D0	34
8	A1	D1	33
9	A0	D2	32
10	PC7	D3	31
11	PC6	D4	30
12	PC5	D5	29
13	PC4	D6	28
14	PC0	D7	27
15	PC1	U _T (V _{CC})	26
16	PC2	PB7	25
17	PC3	PB6	24
18	PB0	PB5	23
19	PB1	PB4	22
20	PB2	PB3	21

JELÖLÉSEK	FUNKCIÓ MEGNEVEZÉSE
D7-D0	kétirányú adatsín (a rendszerhez)
RESET	alaphelyzetbe állító bemenet
CS	IC választó (engedélyező) bemenet
R0	olvasás engedélyezés bemenet
WR	írás engedélyezés bemenet
A0, A1	a port regisztereit címző bemenetek
PA7-PA0	„A” port (8 bit)
PB7-PB0	„B” port (8 bit)
PC7-PC0	„C” port (8 bit)
V _{CC}	+5V tápfeszültség
GND	0V tápfeszültség

A felülvonással jelölt jelek aktív állapota 0.

3.18. b ábra. Az I 8255 PPI lábrendezése, az egyes kivezetések megnevezései

PARANCS A MI INTERFÉSZÜNKKEL ILLESZTETT C-16-OS RENDSZERBEN	CÍMDEKÓDOLÁS			MŰVELETVEZÉRLŐ JELEK		VÉGREHAJTOTT MŰVELET
	interfész engedélyezés			R0	WR	
	A1	A0	CS			
ADATOK BEOLVASÁSA A PORTOKRÓL						
? PEEK(65024)	0	0	0	0	1	„A” portról adat az adatsínre (D _i)
? PEEK(65025)	0	1	0	0	1	„B” portról adat az adatsínre (D _i)
? PEEK(65026)	1	0	0	0	1	„C” portról adat az adatsínre (D _i)
ADATOK KIÍRÁSA A PORTÁKRA						
POKE 65024,V	0	0	0	1	0	V kiírása az adatsínről „A” portra
POKE 65025,V	0	1	0	1	0	V kiírása az adatsínről „B” portra
POKE 65026,V	1	0	0	1	0	V kiírása az adatsínről „C” portra
PARANCS ÍRÁSA A PARANCSREGISZTERBE						
POKE 65027,P	1	1	0	1	0	A parancsregiszterbe a parancs kódja
LEKAPCSOLÓDÁS A RENDSZERSÍN RŐL						
	X	X	1	X	X	Az adatsín felől nézve a PPI nagyimpedanciás állapotban van
	X	X	X	1	1	
NEM ÉRTELMEZHETŐ BEMENETI KOMBINÁCIÓ						
	1	1	0	0	1	A parancsregiszter nem olvasható

3.19. ábra. Az I 8255 parancs- és adatregisztereinek címei és alapvető működési funkciói

A kódszó hatására (lásd 3.20. ábrát is) az „A” oldali A port kimenet lesz, a C port „A” oldali része (a felső 4 bit) nem reteszelt bemenet, a C port „B” oldali része (alsó 4 bit) reteszelt kimenetként, a „B” oldali B port pedig szintén kimenetként fog viselkedni. (3.21. ábra.)

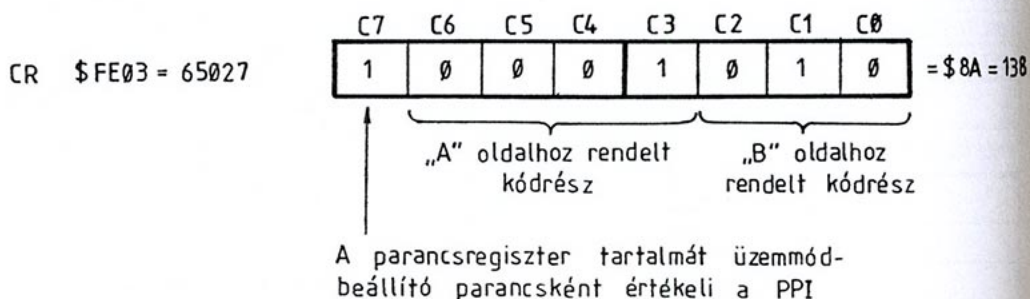
Természetesen, ha a PPI-hez kapcsolt külvilággal kommunikálni akarunk, akkor az első összeköttetés megteremtése előtt a PPI-t be kell állítani. A PPI parancsregisztere a beírt vezérlő kódszót memóriarekeszként őrzi mindaddig, amíg felül nem írjuk azt, vagy az eszközt ki nem kapcsoljuk. Az alábbi kis program a 3.21. ábrán közölt kimeneti konfiguráció beállítása után az A port összes bitjét 0-ra, a B port bitjeit 1-re állítja. A C port első 4 bitjén bitenként felváltva állít be 0-t, 1-et; majd figyel a C port 7. bitjét. Ha PC7=1, akkor hangjelzést ad és leáll.

```

10 REM ** 10. mintaprogram **
11 :
12 :
15 POKE 65027,136:REM a PPI inicializálása
20 POKE 65024,0:REM a port kimeneteinek 0-ra állítása
30 POKE 65025,255:REM a B port minden bitje 1
40 POKE 65026,10:REM a C port alján 1010 kimenet
50 A=PEEK(65026)
60 IF A AND 128 GOTO 80:REM PC7 figyelése
70 GOTO 50
80 SOUND 1,600,5:PRINT "A PC7 biten 1 szint volt"
90 END

```

Legyen a parancsregiszter tartalma: \$8A=138. Hogyan alakul a konfiguráció?



A \$8A-t könnyen átalakíthatjuk bináris kódszóvá. Ezt beírva a CR parancsregiszterbe elemezzük a kialakuló állapotot!

$C0=0$ így a „B” oldali C port rész (alsó 4 bit) kimenet lesz,
 $C1=1$ ezért a „B” oldali B port reteszelt bemenetként fog működni,
 $C2=0$ a „B” oldal 0-ás üzemmódú.

VEZÉRLŐ KÓDSZÓ A PARANCSREGISZTERBEN

C7	C6	C5	C4	C3	C2	C1	C0
----	----	----	----	----	----	----	----

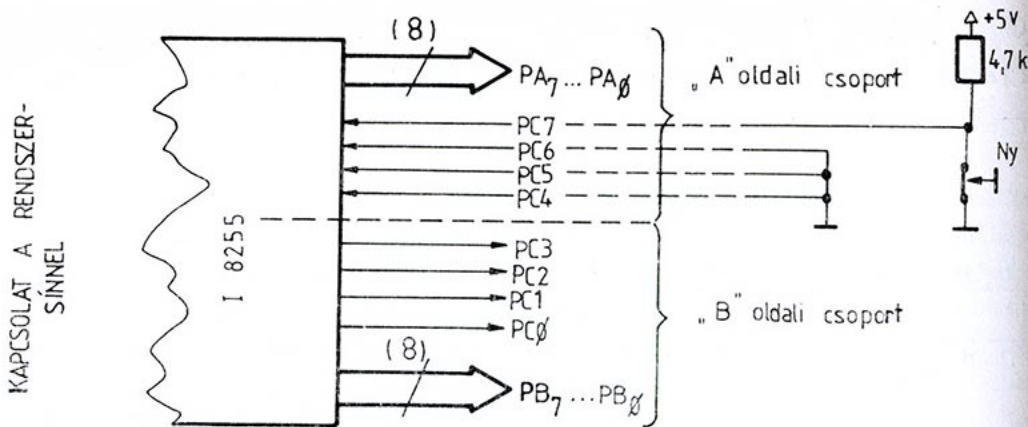
(a parancsregiszter címe 65027 = \$FE03)

a „B” oldal vezérlése	
A „B” oldali C port rész, a C porta alsó 4 bitje: bemenet, ha $C0=1$ (nem reteszelt) kimenet, ha $C0=0$	
A „B” oldali B port rész: bemenet, ha $C1=1$ kimenet, ha $C1=0$	
A „B” oldalnak két üzemmódja adható meg: Ha $C2=0$, akkor ez 0-ás típusú ún. alap input/output üzemmód. Ha $C2=1$, akkor ez 1-es típusú ún. ütemezett, kapuzott üzemmód. Ebben az esetben a „B” oldali C port rész I/O, vagy vezérlőjel funkciót lát el.	

az „A” oldal vezérlése			
Az „A” oldali C port rész, a C porta felső 4 bitje: bemenet, ha $C3=1$ (nem reteszelt) kimenet, ha $C3=0$			
Az „A” oldali A port rész: bemenet, ha $C4=1$ kimenet, ha $C4=0$			
Az „A” oldalnak három üzemmódja adható meg:			
Ha	C6	C5	akkor ez 0-ás típusú ún. alap input/output üzemmód.
	0	0	
Ha	0	1	akkor ez 1-es típusú ún. ütemezett, kapuzott üzemmód. Ebben az esetben az „A” oldali C port rész I/O, vagy vezérlőjel funkciót lát el.
Ha	1	X	akkor ez 2-es típusú, az „A” port kétirányú üzemmódja, az „A” oldali C port és a „B” oldali C port legnagyobb helyiértékű bitje vezérlési feladatokat lát el, a többi C bit I/O. A B port 0-ás üzemben programozható.

A parancsregiszter üzemmód választásként van kiértékelve, ha $C7=1$

3.20. ábra. Az I 8255 üzemmódjának beállítása a parancsregiszterrel



3.21. ábra. Az I 8255 és a külvilág kapcsolata, ha a vezérlő-kódszó 136=888

C3=1 az „A” oldali C port rész (felső 4 bit) nem reteszelt bemenet

C4=0 az „A” oldali A port kimenet lesz

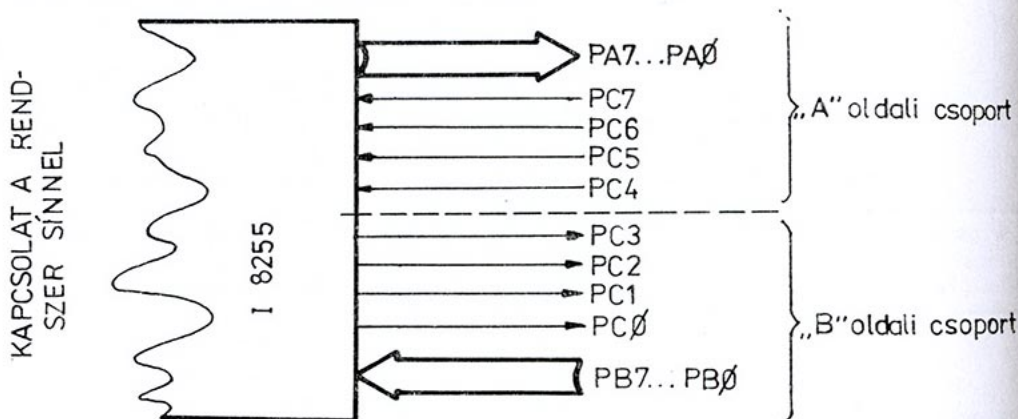
C5=C6=0 így az „A” oldal is nullás üzemmódú

C7=0 a parancsregiszterben üzemmód-beállító parancs van!

A porták beállítása szemléletesen a 3.22. ábrán követhető:

Mivel a PPI-nek 2 db 8 bites, plusz 2 db 4 bites (összesen 4 db) kapuja van és 0-ás üzemmódban mindegyik egyaránt lehet bemenet és kimenet is, ezért a külvilággal kapcsolatot tartó portok beállításának 16 lehetséges változata képzelhető el. E leírás keretében nem foglalkozunk a PPI 1-es, ill. 2-es üzemmódjának programozásával, mert ez olyan szakmai ismereteket feltételez, amelyeket egy következő kiadványban tervezünk közölni.

Aki nem elégszik meg ennyivel, az a [3.] szakirodalom 185—192. oldalán utána nézhet a témakörnek (ill. [38.] 23—26. o., 46—47. o.).

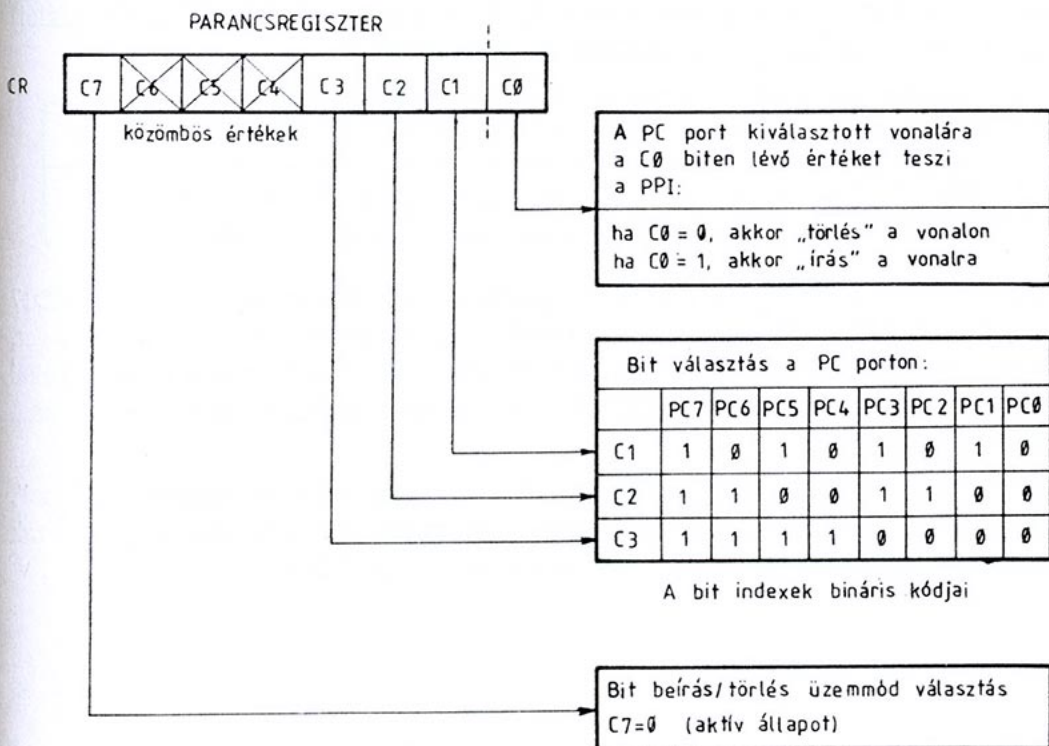


3.22. ábra. Az I 8255 és a külvilág kapcsolata, ha a vezérlő-kódszó 138=88A

A bemenet reteszeltsége azt jelenti, hogy a beolvasással egy időben az adott port adatai beíródnak egy bemeneti adatregiszterbe. Ez tárolja az adatokat akkor is, ha már a bemeneten azok nincsenek jelen. Így a bemeneti adatregiszterből az adatok ismét kiolvashatók.

Ha a PPI parancsregiszterének legnagyobb helyiértékű bitje $C7=0$, akkor az eszköz nem működési módot beállító parancsnak tekinti a parancsregiszter tartalmát, hanem a tartalomnak megfelelő módon bitenként írja vagy törli a PC porta kimeneti állapota kapcsolt bitvonalait.

Értelmezzük a PPI ilyen működését a 3.23. ábra alapján!



3.23. ábra. Az 18255 bit beírás/törlés üzemmódja a PC porton

A PC port kimeneti meghajtói ebben az üzemmódban is reteszelik és tartják a bitre írt állapotot.

A következő kis program egy $\approx 50\%$ -os kitöltési tényezőjű [19.], [20.] négyzögjel-sorozatot ad a PC0 biten:

```

10 REM ** 11. mintaprogram **
11 :
12 :
15 POKE 65027,0:REM a PPI parancsregiszter init.
20 POKE 65027,PEEK(65027) OR 1:REM a PC0 bit írása
30 POKE 65027,PEEK(65027) AND 254:REM a PC0 bit törlése
40 GOTO 20
    
```

Gépi kóddal gyorsabb négyszögimpulzusok generálhatók! És különböző időzítések, frekvenciák állíthatók be.

Az 1.4. fejezetben említettük, hogy a C 16 és a C Plus/4 I/O lehetőségei nem teljesen azonosak. A C Plus 4-es rendelkezik egy **kétirányú párhuzamos I/O portával** a \$FDA0-\$FD1F címeiken, és egy **soros vonali kommunikáció** megvalósítására alkalmas programozható perifériavezérlővel a \$FD00-\$FD0F címterületen. Ez utóbbi működése újabb hardver és szoftver előtanulmányokat igényel, s talán a tervezett folytatásban lesz módunk foglalkozni vele.

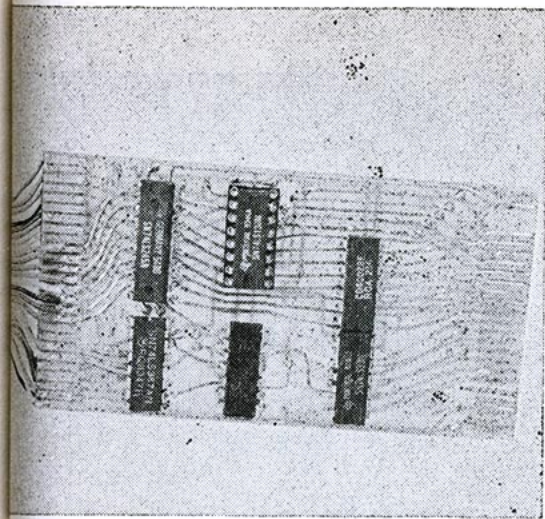
A MOS 6529B párhuzamos 8 bites porta felderítése pedig nem jelenthet gondot annak az olvasónak, aki eddig eljutott és tudja, hogy a port az ún. USER (felhasználói) csatlakozón van kivezetve az alábbiak szerint:

Port bitjei :	P7	P6	P5	P4	P3	P2	P1	P0
Csatlakozón a láb jele :	F	J	7	6	5	4	K	B

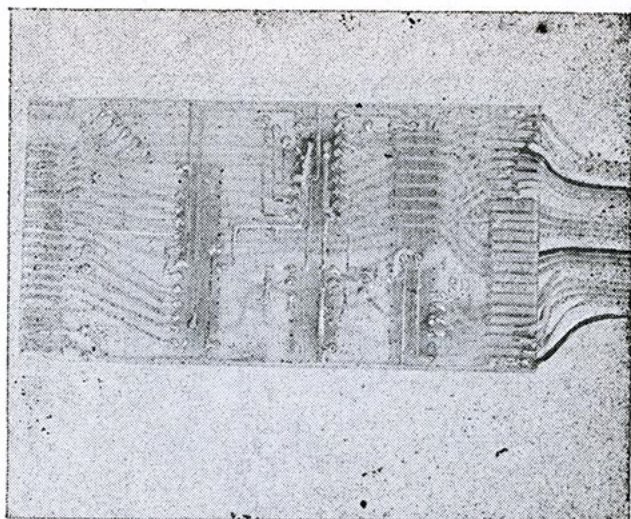
Alapállapotban minden bit 1 feszültség szinten van. Olvasáskor a port a \$FD17 rekeszbe előzőleg beírt érték megváltozását tudja érzékelni. Felhívjuk a figyelmet, hogy a port egyetlen kivezetése sem köthető közvetlenül tápfeszültségre vagy logikai 1 szintre. A bemeneteken legalább 3 kOhmos áramkorlátozó ellenállást célszerű alkalmazni.

Reméljük, az olvasó ennyi ismerettel már hozzá mer nyúlni számítógépe „lelkéhez”, a rendszersínhez és különböző „misztikus” kivezetéseihez. Ezek után az interfészek tervezése nem ördögös feladat, következhetnek a perifériák!

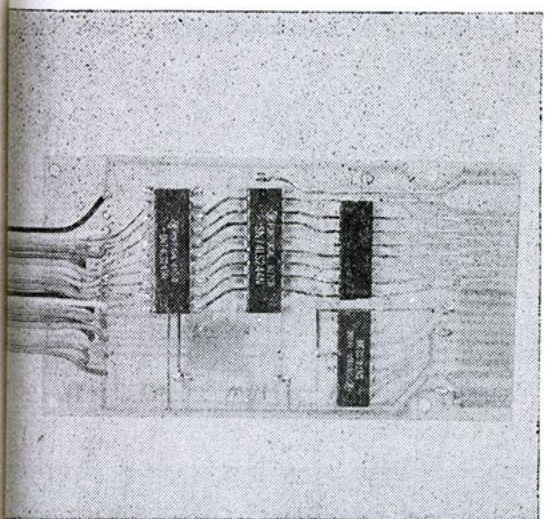
4. Az eszközök fényképei



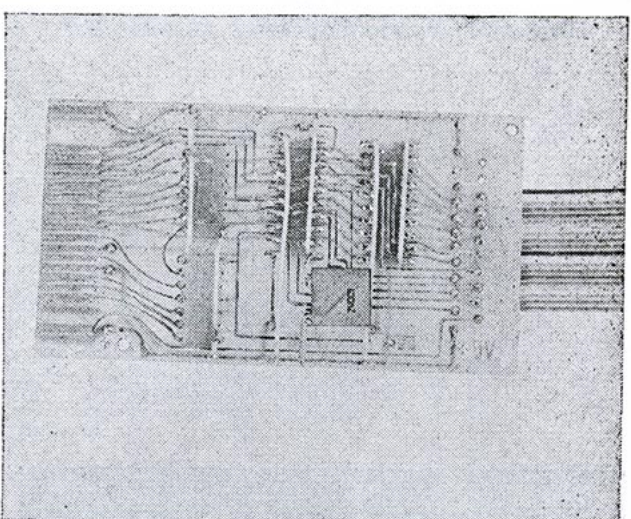
8 csatornás interfész alkatrész oldal



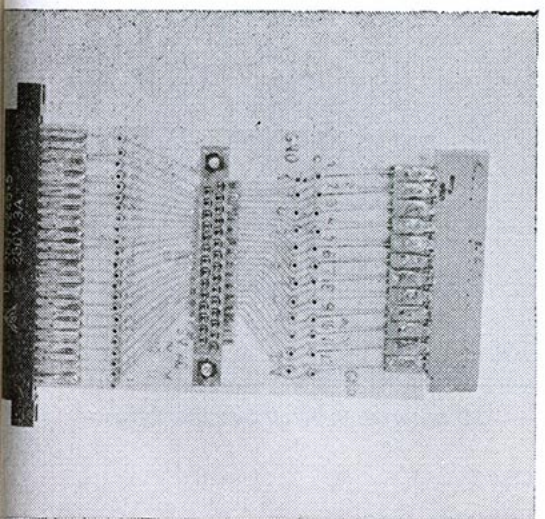
8 csatornás interfész forrasztási oldal



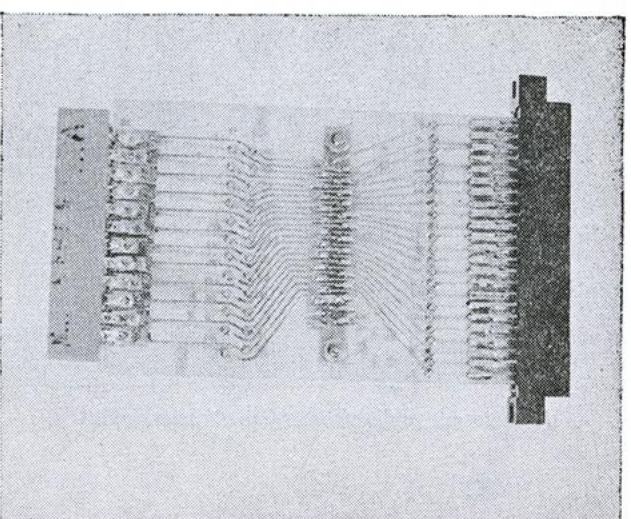
Egyszerű buszillesztő interfész alkatrész oldal



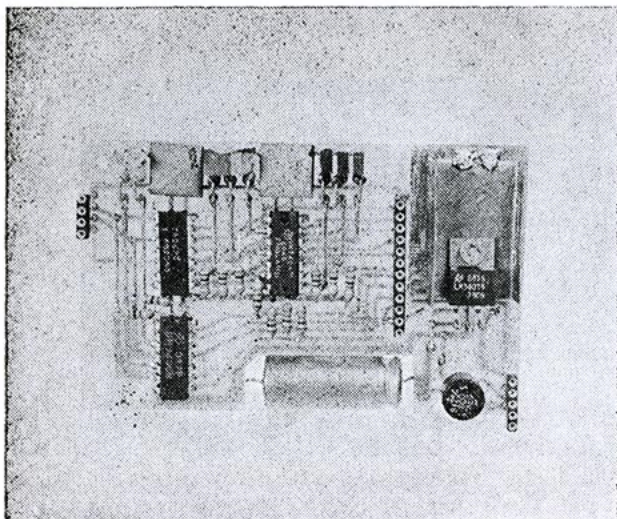
Egyszerű buszillesztő interfész forrasztási oldal



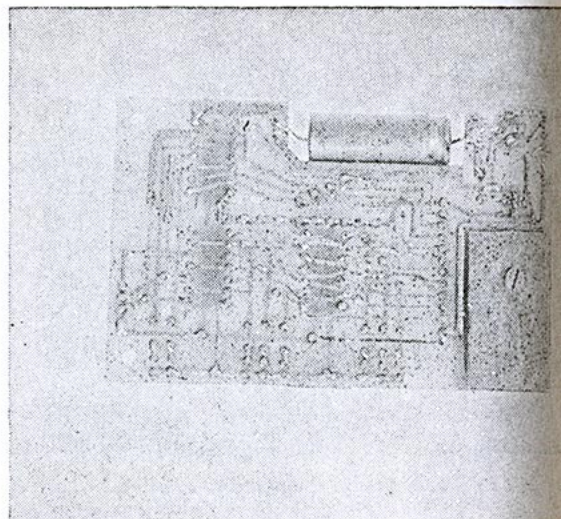
C+4 proto-board NYÁK alkatrész oldal



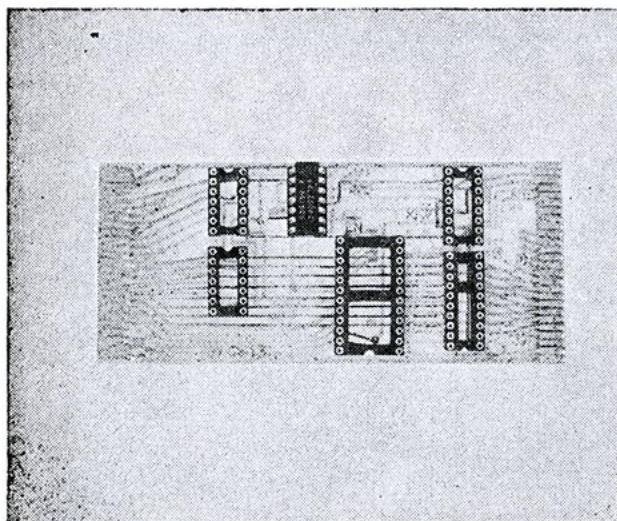
C+4 proto-board NYÁK forrasztási oldal



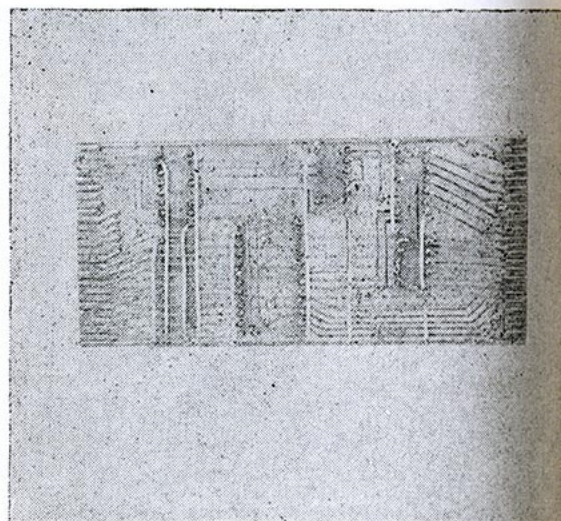
Meghajtott párhuzamos kimenetek alkatrész oldal



Meghajtott párhuzamos kimenetek forrasztási oldal



16 csatornás pufferelt interfész alkatrész oldal



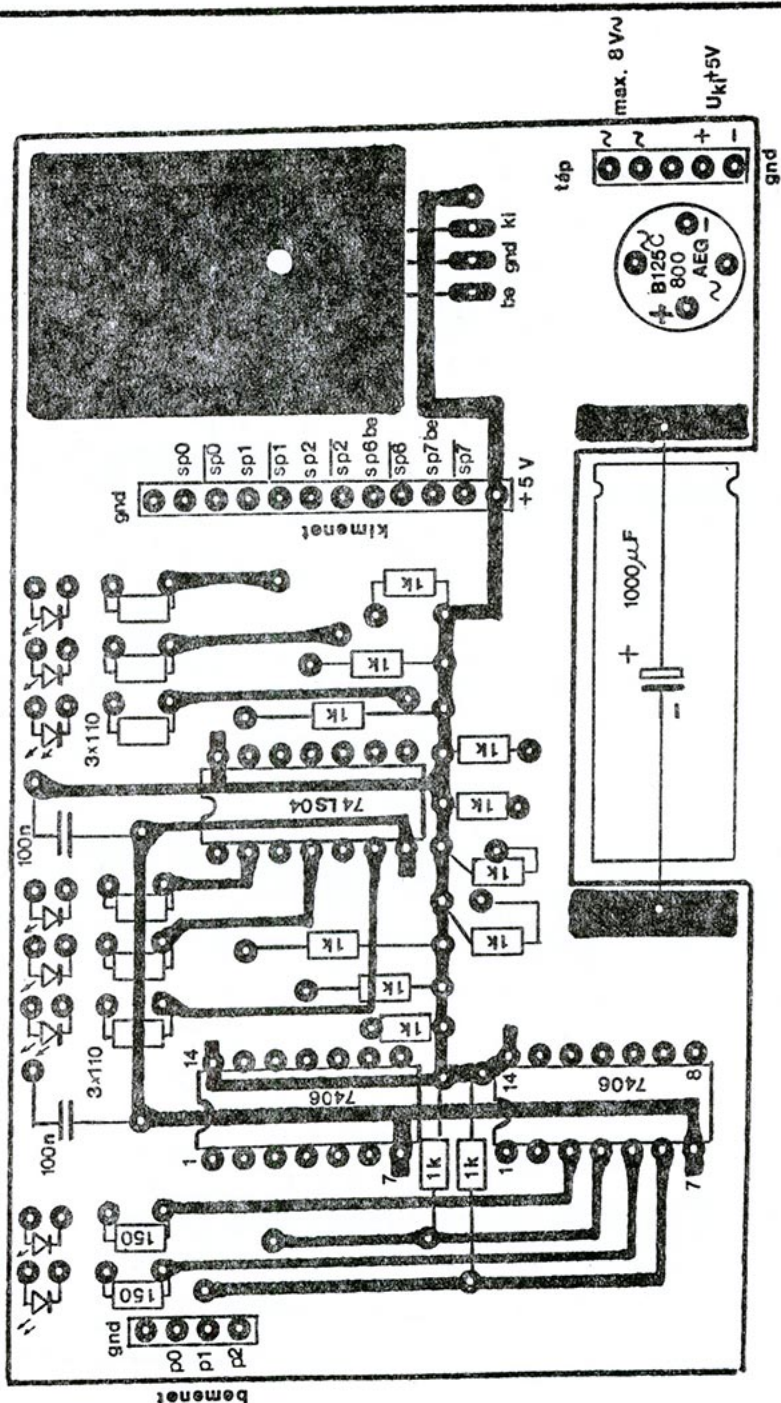
16 csatornás pufferelt interfész forrasztási oldal

5. Az eszközök NYÁK-jainak beültetési rajzai

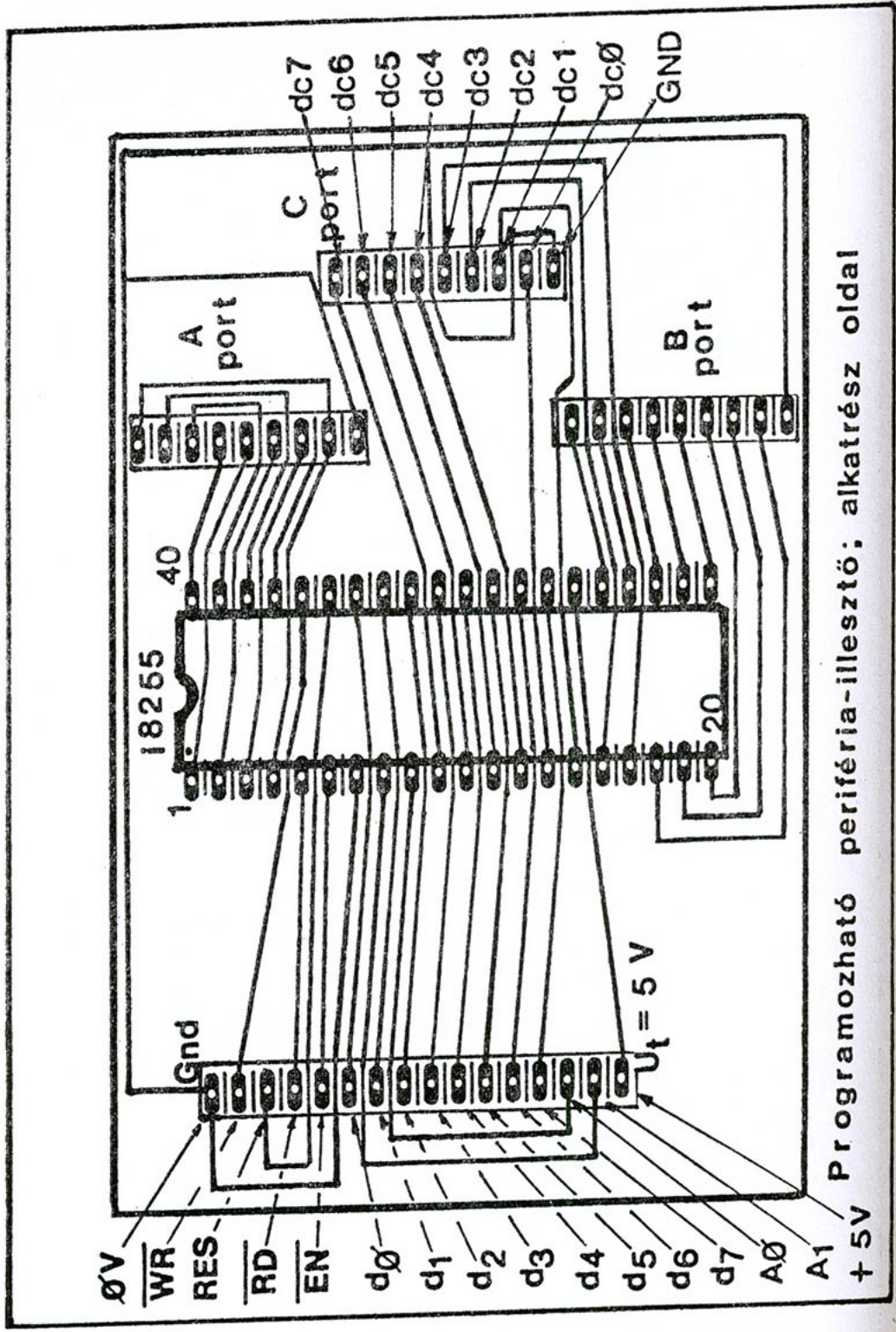
2 CQY 27

3 CQY 26

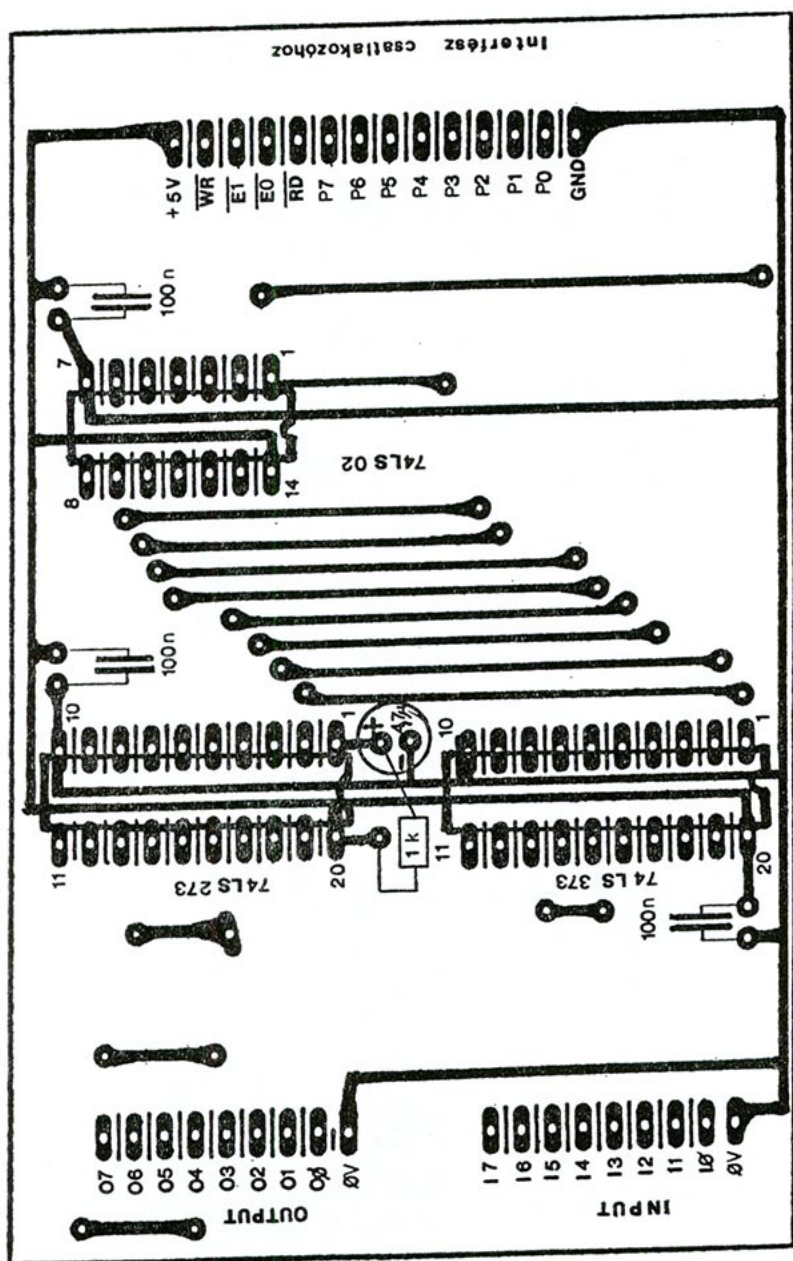
3 CQY 28



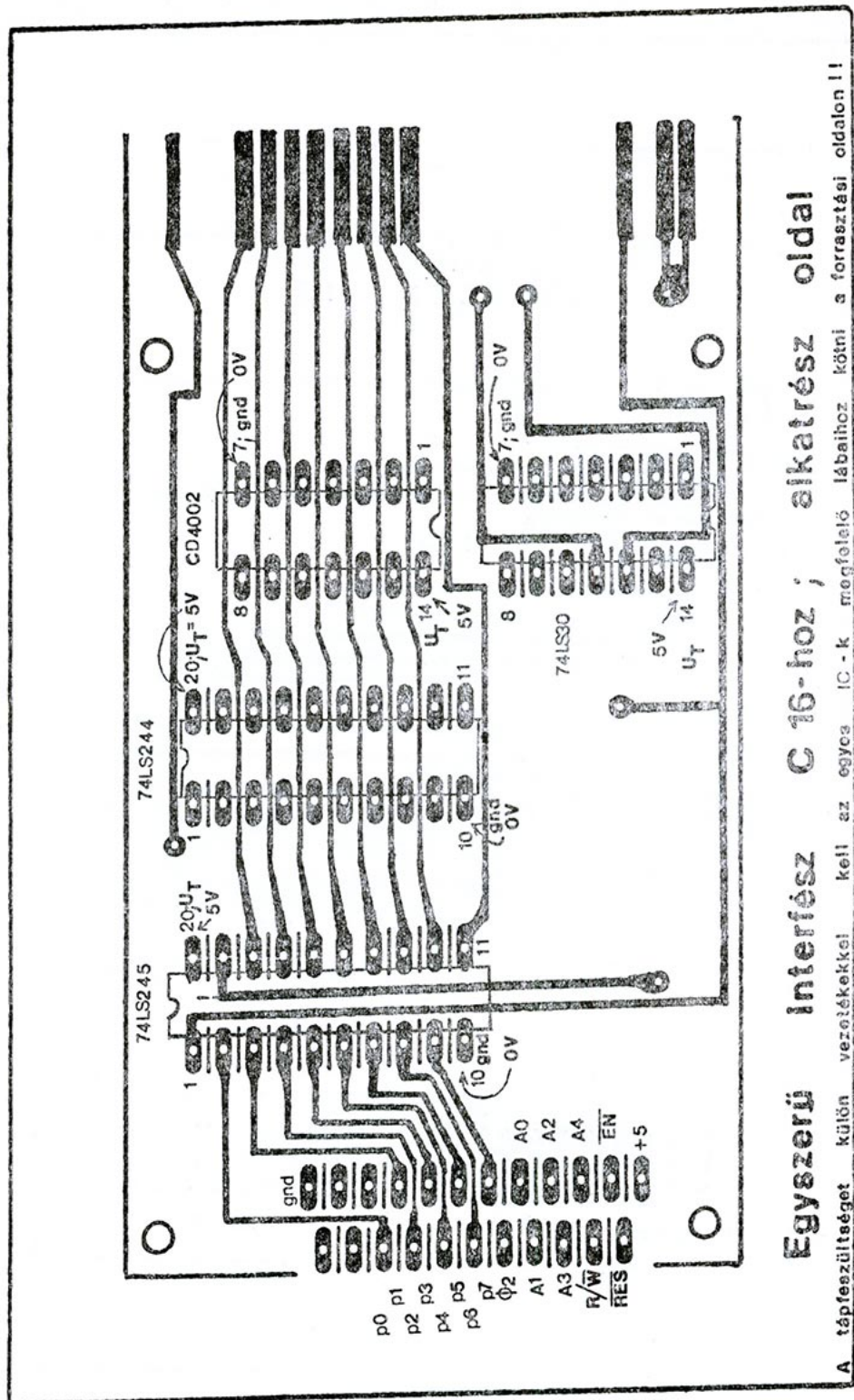
C 16 párhuzamos I/O alkatrész oldal



Programozható periféria-illesztő; alkatrész oldal



Digitális I/O periféria; alkatrész oldal

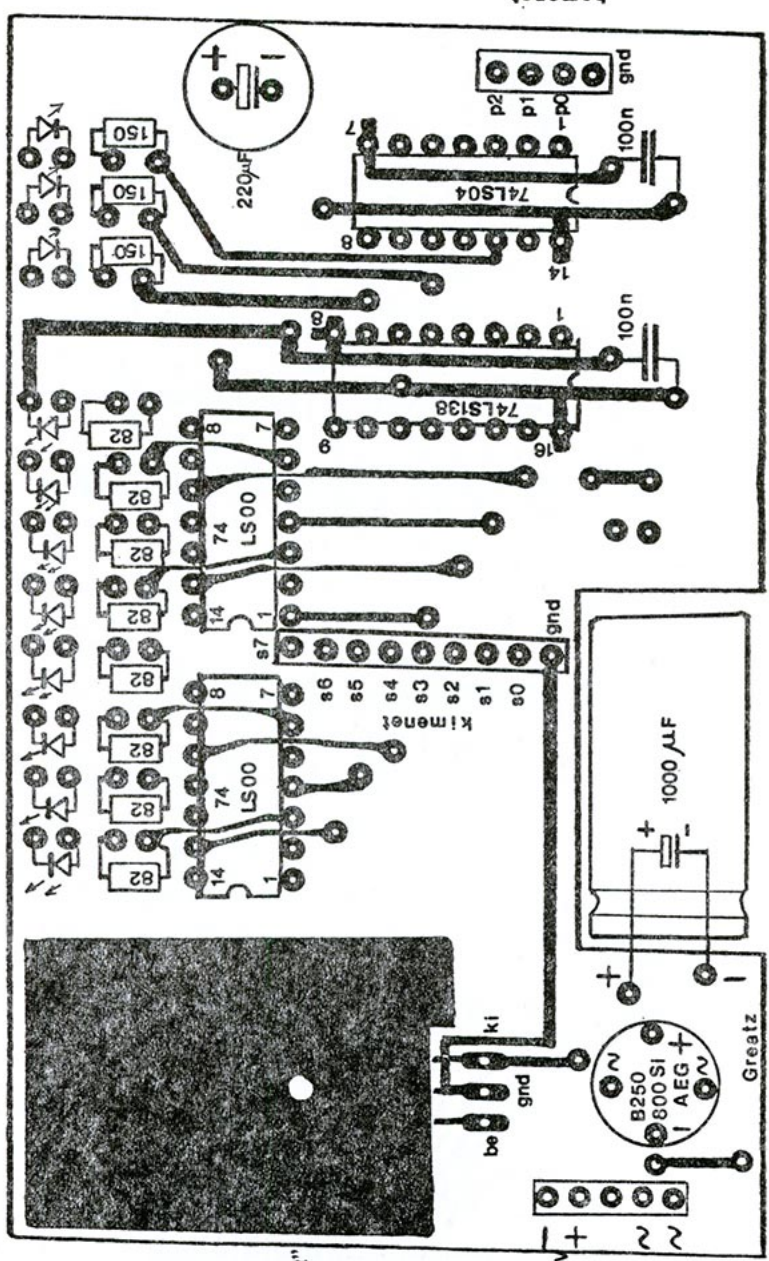


Egyszerű interfész C 16-hoz ; alkatrész oldal

A tápfeszültséget külön vezetékkel kell az egyes IC-k megfelelő lábaihoz kötni a forrasztási oldalon !!

3XCQY 27 sárga LED

8XCQY 26 piros LED

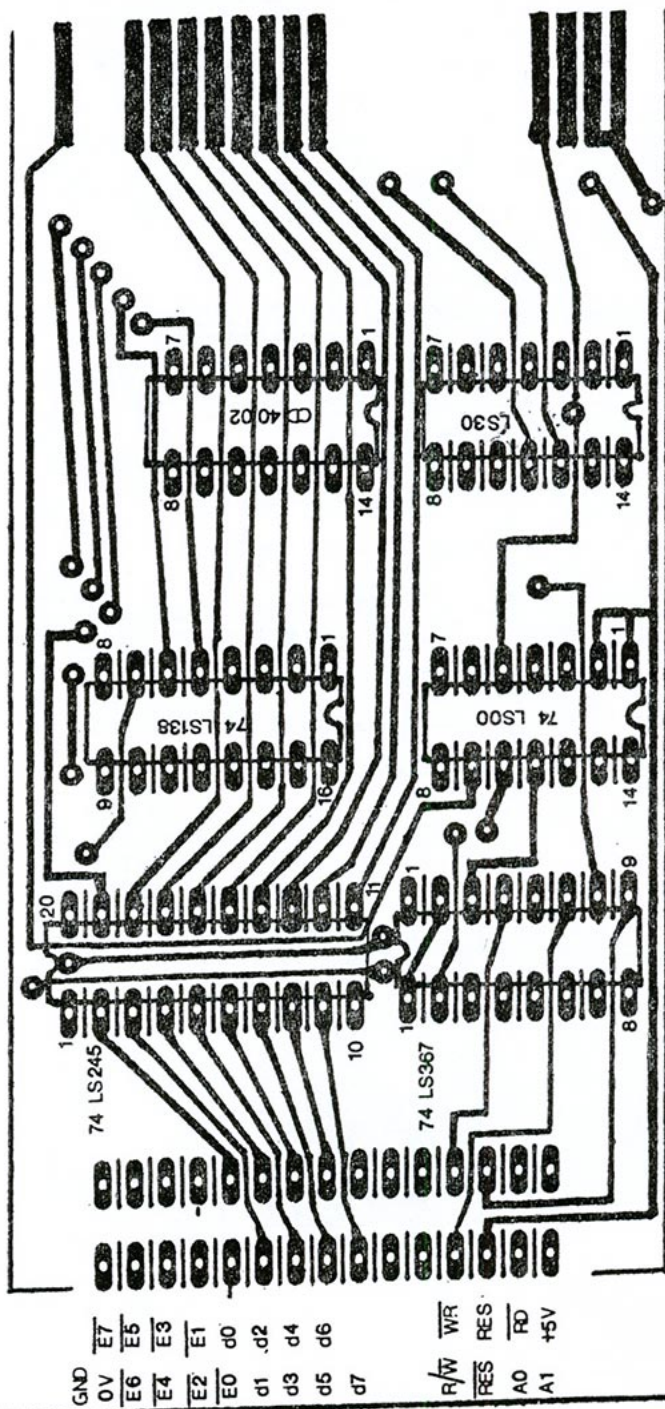


μA 7805
"hűtőfelület"

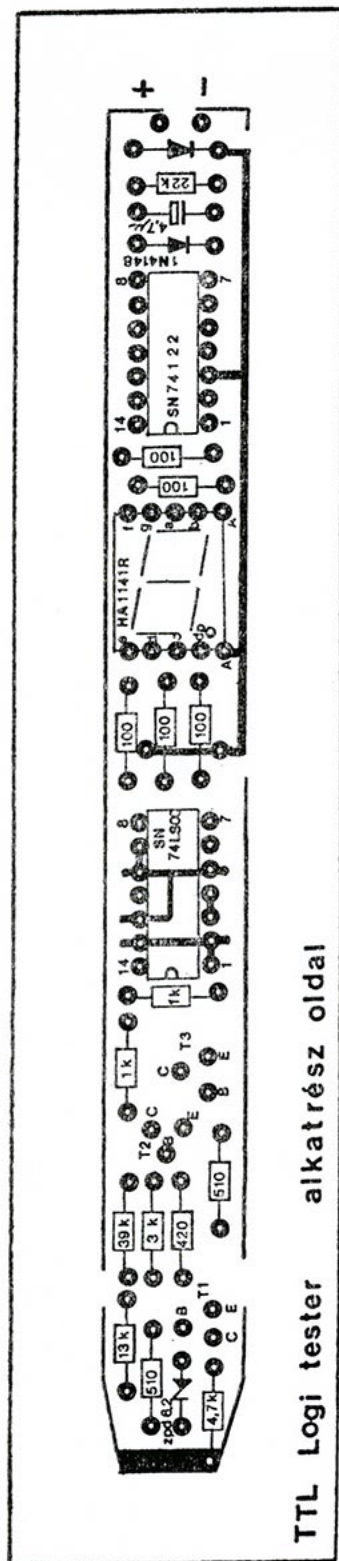
$U_{ki}(\text{stab})=5V$

$U_{be \text{ max}} = 8V \sim$

Multiplexer (3-ról 8-ra) alkatrész oldal



Az \overline{EN}_j -ket is előállító interfész C16-hoz; alkatrész oldal



HIVATKOZOTT IRODALOM JEGYZÉKE

- [1] C 16, C Plus/4 Programozói útmutató, NOVOTRADE RT 1987., Bp.
- [2] Tóth Viktor; A COMMODORE 16-os belső felépítése, NOVOTRADE, 1986. Bp.
- [3] Vancsó Gyula: Mikroszámítógép-elemek a tervezéshez, Műszaki Könyvkiadó, Bp. 1981.
- [4] Dr. Csáki Frigyes: Bevezetés a digitális technikába, Akadémiai Kiadó, Bp. 1977.
- [5] Dr. Arató Péter: Logikai rendszerek tervezése, Tankönyvkiadó, Bp. 1985.
- [6] P. Müller: Számítástechnikai kislexikon, Műszaki Könyvkiadó, Bp. 1973.
- [7] R. Bradbeer: Műsoron a számítógép, Műszaki Könyvkiadó, Bp. 1984.
- [8] N. M. Morris: Digitális áramkörök és rendszerek, Műszaki Könyvkiadó, 1977.
- [9] Matematika, tankönyv az ált. iskolák 5. és 6. osztályai számára, Tankönyvkiadó, Bp.
- [10] Urbán János: Matematikai logika (példatár), Műszaki Könyvkiadó, 1983. (Bólyai J. sorozat)
- [11] James W. Coffron: Mikroprocesszoros rendszerek gyakorlati hibakeresése, Műszaki Könyvkiadó, Bp., 1984.
- [12] L. Englisch: Gépi kódú programozás a COMMODORE 64-esen, Data Becker-NOVOTRADE, Bp. 1986.
- [13] Ferenczi Ödön: Feszültségstabilizátorok, akkumulátor töltők, Műszaki Könyvkiadó, Bp., 1976.
- [14] TEXAS TTL RECEPTEK, Műszaki Könyvkiadó, Bp., 1976.
- [15] Magyarai: Digitális IC atlasz, Műszaki Könyvkiadó, Bp., 1982.
- [16] Dr. Izsák M.: Távközléstechnikai kézikönyv.
- [17] Englisch—Szczepanowski: A VC-1541-es lemezegység programozása, Data Becker-NOVOTRADE, Bp., 1986.
- [18] Dr. Schnell L.: Jelek és jelrendszerek mérés technikája, Műszaki Könyvkiadó, Bp., 1985.
- [19] U. Tietze—Ch. Schenk: Analóg és digitális áramkörök, Műszaki Könyvkiadó, Bp., 1981.
- [20] V. H. Grinich—H. G. Jackson: Példák integrált áramkörök alkalmazására, Műszaki Könyvkiadó, Bp., 1980.
- [21] Lambert M.: Teljesítményszabályozók integrált áramkörei (Elektronika sorozat), Műszaki Könyvkiadó, Bp., 1980.
- [22] Ferenczi Ö.: Tápegységek amatőröknek (Elektronika sorozat), Műszaki Könyvkiadó, Bp., 1980.
- [23] Ferenczi Ö.: Félvezetős feszültségátalakítók, Műszaki Könyvkiadó, Bp., 1979.
- [24] Ferenczi Ö.: Kapcsolóüzemű tápegységek, Műszaki Könyvkiadó, Bp., 1978.
- [25] K. Halas—Mészáros—Szentiday: Optoelektronikai kijelzők és megjelenítők, Műszaki Könyvkiadó, Bp., 1984.
- [26] Lambert Mikós: Optoelektronikai hobby, Műszaki Könyvkiadó, Bp., 1982.
- [27] TEXAS optoelektronikai receptek, Műszaki Könyvkiadó, Bp., 1979.
- [28] M. Andrews: Mikroprocesszorok és illesztőegységek, Műszaki Könyvkiadó, Bp., 1985.
- [29] Lambert M.: RC időzítésű billenőkörök, Műszaki Könyvkiadó, Bp., 1984.
- [30] A. E. Zelgyn: Bistabil áramkörök, Műszaki Könyvkiadó, Bp., 1986.
- [31] Dr. Szalay M.: Nyomatott áramköri lapok amatőröknek, Műszaki Könyvkiadó, Bp., 1984.
- [32] Dr. Szalay M.: Nyomatott áramkörök tervezése, Műszaki Könyvkiadó, Bp., 1982.
- [33] Rigó Béla: Járműmodellek elektronikus távvezérlése, Műszaki Könyvkiadó, Bp., 1979.

- [34] Madarász L.: Digitális CMOS kapcsolásgyűjtemény, Műszaki Könyvkiadó, Bp., 1987.
- [35] I. E. Shepherd: Műveleti erősítők, Műszaki Könyvkiadó, Bp., 1985.
- [36] N. Hesselmann: Digitális jelfeldolgozás, Műszaki Könyvkiadó, Bp., 1985.
- [37] J. Wojciechowski: Elektronikai játékok építése, Műszaki Könyvkiadó, Bp., 1980.
- [38] W. Link: BASIC a mérés-, a vezérlés- és a szabályozástechnikában, Műszaki Könyvkiadó, Bp. 1987.

88-3024 — Szegedi Nyomda

Felelős vezető: Surányi Tibor igazgató

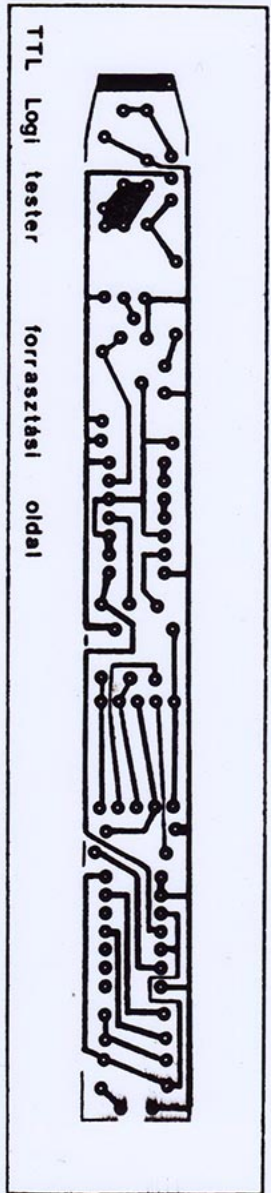
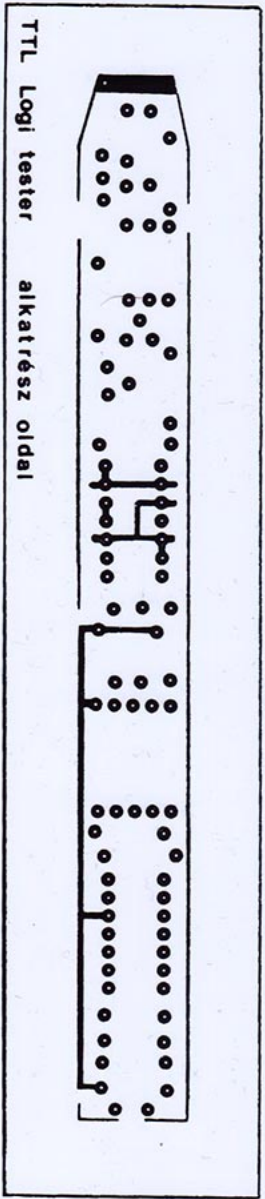
A VORKER Kiszövetkezet ajánlja a Tisztelt Megrendelők figyelmébe az alábbi szoftver termékeket és — szolgáltatásokat.

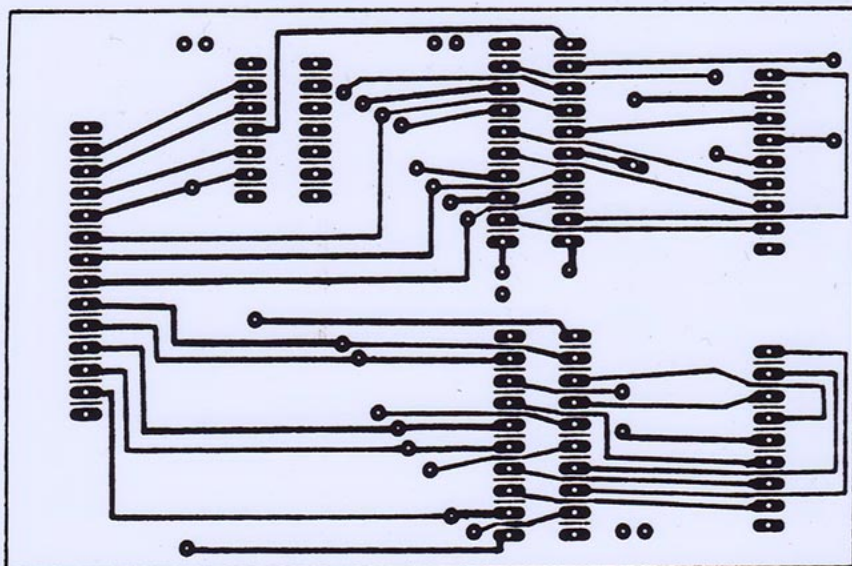
1. C 16, C Plus/4, C 64 és Spectrum gépekre kifejlesztett oktató és játékprogramokat, amelyek Magyarországon a legolcsóbbak. Ezek közül az idegen és magyar nyelv, a matematika, a kémia, a biológia és a fizika tantárgyi oktatóprogramok segítik a pedagógusok és a tanulók munkáját.
2. Egy- és kétszemélyes játékok, valamint totóprogramok segítik a szabadidő jobb eltöltését.
3. IBM kompatibilis gépekre vállaljuk a munkaügyi és bérszámfejtő rendszerünk adaptálását.

Különleges hardver ajánlatunk: a világon egyedülálló fejlesztésünk a TC-NET+4 számítógép-interfész, amellyel 16 db C 16 illetve C Plus/4 számítógépet kapcsolhatunk lokális hálózatba. Ez két fontos célt szolgál: egyrészt az összes gép ilyen módon használhat egyetlen mágneslemezt és nyomtató egységet, másrészt megvalósítja számítástechnikai vonalon azokat az előnyöket, amelyeket a nyelvoktatás területén egy nyelvi labor biztosít. Ily módon számítástechnikai kabinet hozható létre kis anyagi ráfordítással.

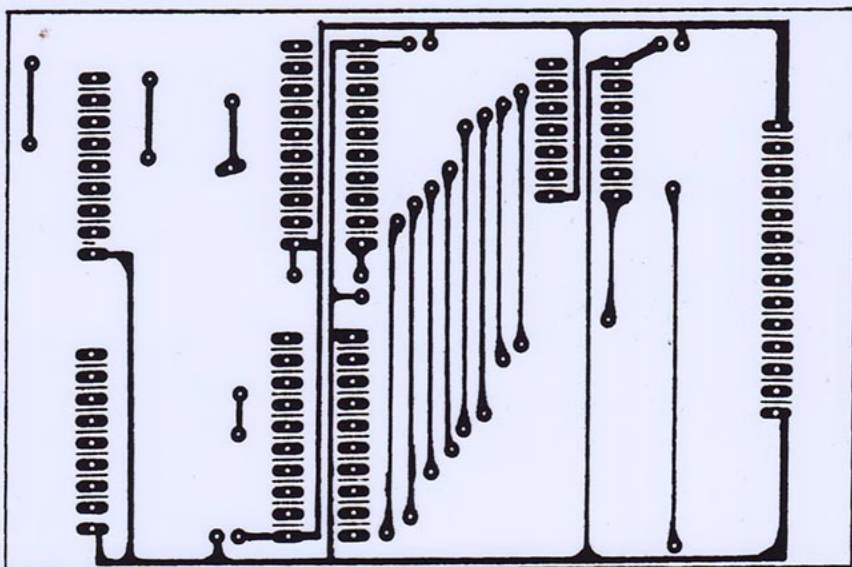
Várjuk érdeklődésüket!

VORKER® Vállalatközi
Organizációs Ipari Szolgáltató
és Kereskedelmi Kiszövetkezet
Szeged, Pf: 711.
Telex: 82-688
Telefon: (62) 26-144
(62) 25-479

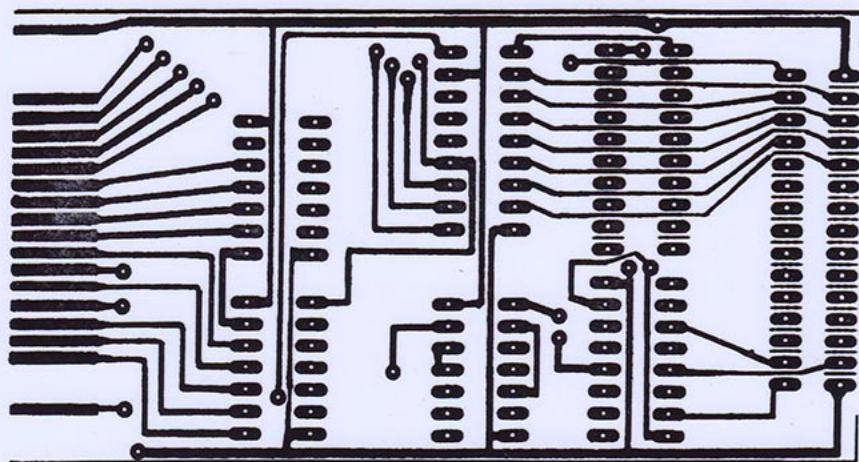




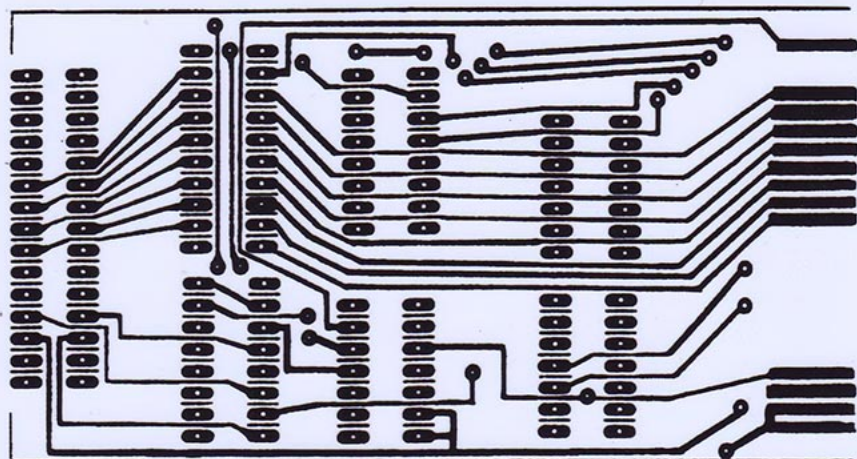
Digitális I/O periféria C16-hoz; forrasztási oldal



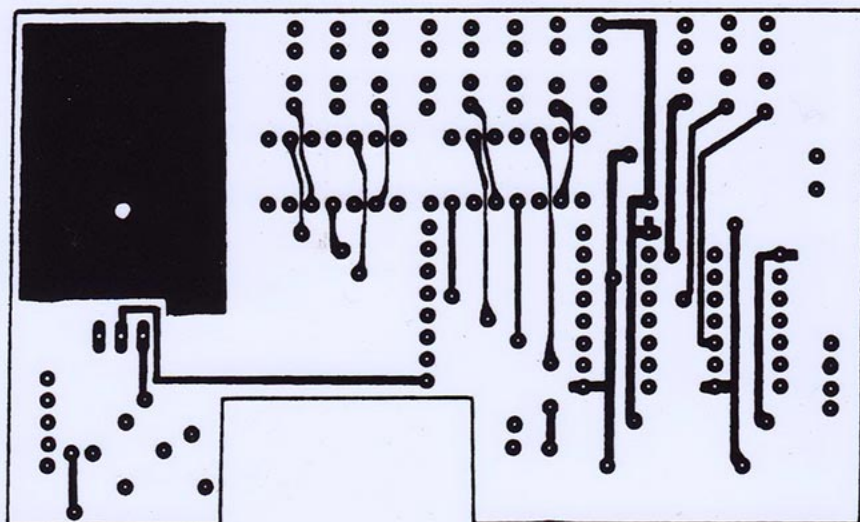
Digitális I/O periféria; alkatrész oldal



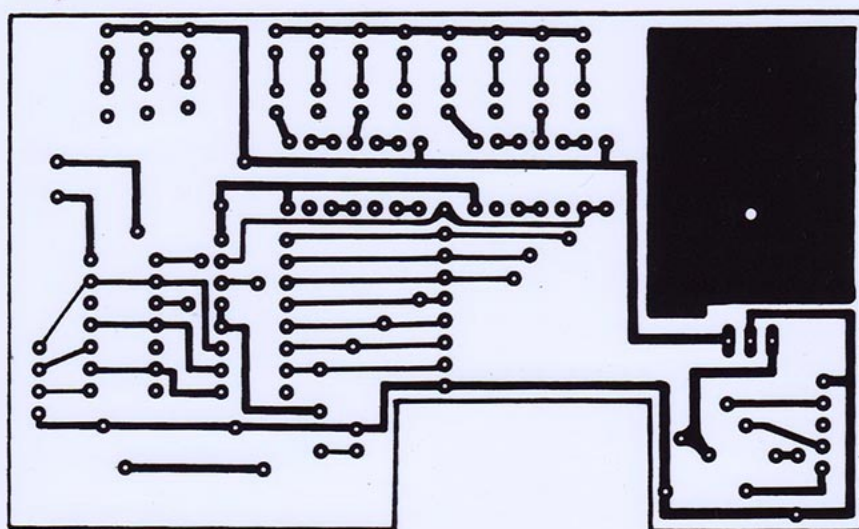
Az \overline{EN}_1 -ket is előállító interfész C16-hoz; forrasztási oldal



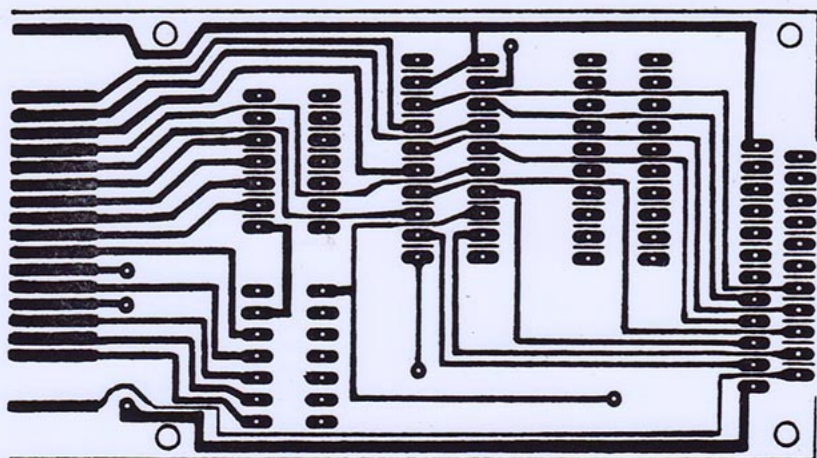
Az \overline{EN}_i -ket is előállító interfész C16-hoz; alkatrész oldal



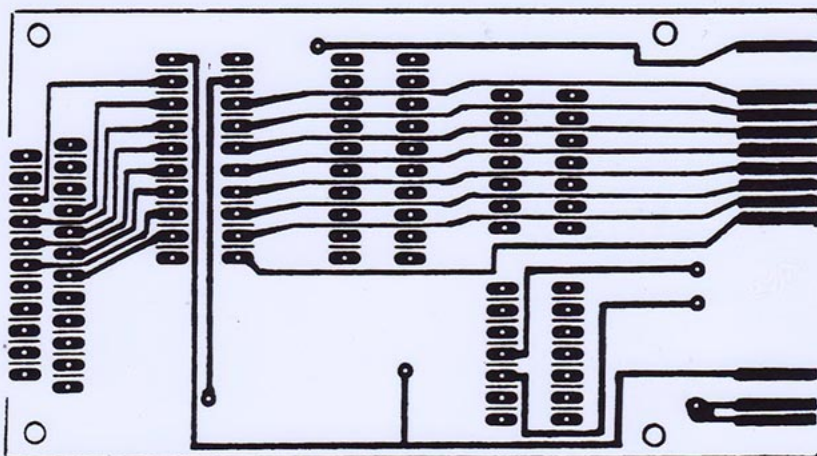
Multiplexer (3-ról 8-ra) alkatrész oldal



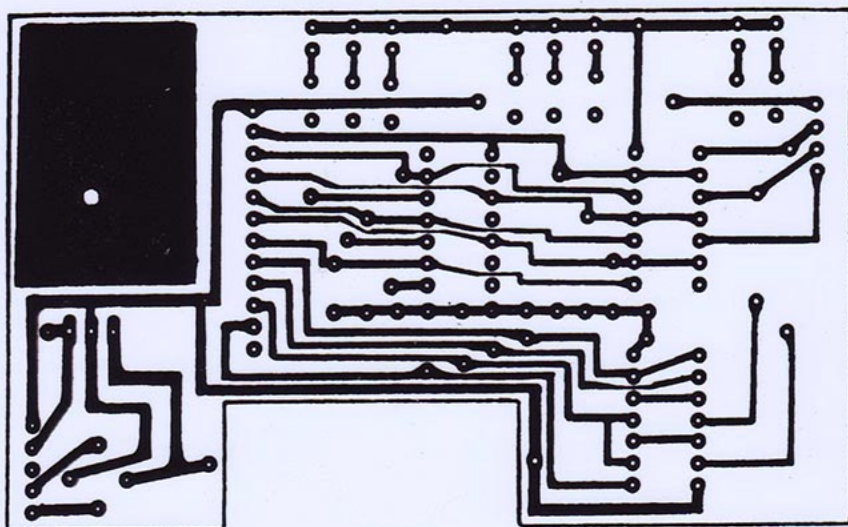
MULTIPLEXER (3-ról 8-ra) forrasztási oldal



Egyszerű interfész C 16-hoz ; forrasztási oldal

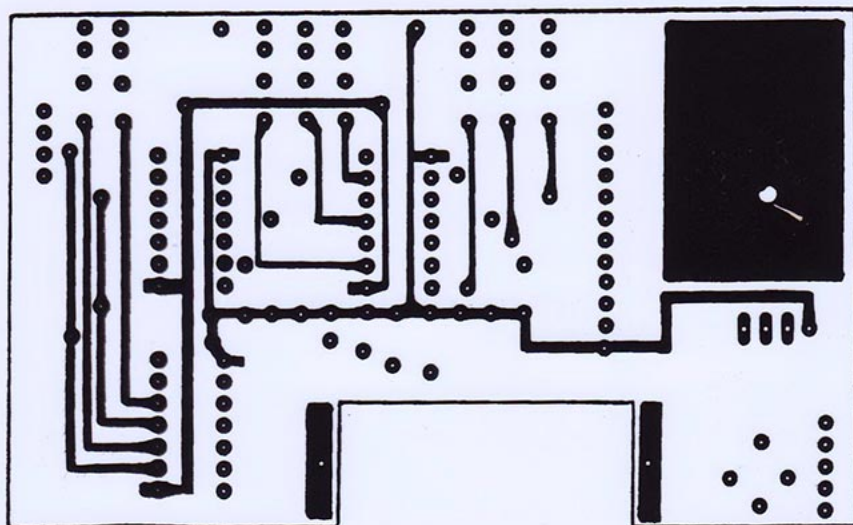


Egyszerű interfész C 16-hoz alkatrész oldal



C 16 párhuzamos I/O

forrasztási oldal



C 16 párhuzamos I/O

alkatrész oldal

