

KECSKEMÉTI FŐISKOLA  
GÉPIPARI ÉS AUTOMATIZÁLÁSI MŰSZAKI  
FŐISKOLAI KAR

Dr. Madarász László

MIKROPROCESSZOROK, MIKROSZÁMÍTÓGÉP  
ELEMEEK

Átdolgozott kiadás

KECSKEMÉT  
2009

Szerző:	Dr. Madarász László
Kiadás éve:	2009
	Átdolgozott kiadás
Szakmai lektor:	Bene Sándor okl. villamosmérnök
Nyelvhelyességi lektor:	Várkonyiné Stumpf Anikó

Felelős kiadó:	Dr. Danyi József rektor
A kiadást végző kar:	KFGAMFK
Példányszám:	400
Terjedelem:	13,25 A/5 ív
Azonossági szám:	KF-GAMFK-H-392

---

A nyomdai munkát végezte:	Kecskeméti Főiskola KIK Nyomda
Munkaszám:	2009.147

## Bevezetés az 1998. évi első kiadáshoz

A mikroelektronika születését az első mikroprocesszor bejelentésének időpontja alapján 1971-re szokás dátumozni. Ez a ma huszonnyolc éves technikai ág többszörösen kiforgatta a világot a sarkaiból; átalakította a technikai rendszereket is és a társadalmakat is. A mikroelektronika az információs forradalom technikai háttere is, a kényelmi eszközök alapja, a szórakoztató elektronika bázisa, az ipari irányítási-logisztikai rendszerek központi eleme, a PC-k és a munkaállomások építő elemkészlete. A mikroelektronika nélkül nincs orvosi diagnosztika, repülés, hadászat, űrhajózás, rádiótelefon, pénzgépjegy automata.

A bennünket kiszolgáló elektronikus készülékek többségében a központi irányító szerepet egy vagy több mikroprocesszor, mikrovezérlő látja el. A következő oldalakon megismerjük a mikroprocesszorok felépítését, működését. Az általános bemutatáson túl néhány típussal részletesebben is megismerkedünk majd.

A mikroprocesszorok, mikrovezérlők témakör oktatásakor gondot okozhat az, hogy az általános ismereteket milyen konkrét típusokhoz kössük, melyik áramkörökkel ismerkedjünk meg részletesen is. Vannak szerzők, akik egy elképzelt mikroprocesszorról beszélnek, annak feltételezett utasítás készletét és belső kialakítását ismertetik. Mások csak a legújabb típusok bemutatását tartják helyesnek.

Ez a jegyzet teljes részletességgel néhány nyolc bites mikroprocesszort fog bemutatni. Ezek a kiválasztott típusok igen jó alapismereteket nyújtanak a mikrovezérlőkhöz is és a nagyobb szóhosszúságú processzorok megértéséhez is. A legalaposabban az Intel 8080A mikroprocesszor szerepel a jegyzetben. Ennek már korszerű a belső kialakítása, tömör és hatékony az utasításkészlete, a hardver működés mégis egészében áttekinthető, megérthető, követhető. Megismerése biztos alapot és támpontot nyújt a többi tananyagrészhöz is és a későbbi önálló búvárkodáshoz is.

A mikroszámítógép elemek között sorra kerülnek a különféle memória áramkörök, az I/O elemek, a programozható kiegészítő egységek, az A/D és D/A konverterek. megismerjük a mikrovezérlők általános felépítését, tulajdonságait is.

A mikroprocesszorok, mikrovezérlők, a memóriák és egyéb kiegészítő áramkörök a nagyintegráltságú digitális áramkörök (LSI) csoportjába tartoznak. A jegyzetnek mintegy kereket ad az LSI áramkörök áttekintése az első oldalakon, majd az utolsó fejezetben az egyre nagyobb jelentőségű LSI család, a PLD megismerése.

A jegyzet a mikroelektronika hardvereszközeivel, a hardverépítés technikájával foglalkozik, a programozást, a szoftver fejlesztést csak annyiban érinti, amennyiben a hardver működésének megértéséhez szükséges. Az utasítások szerkezetét, felépítését, az utasításelemek elhelyezkedését a memóriában, a végrehajtás szervezését át kell tekintenünk, mert ezek nélkül a mikroprocesszor sok belső részletének, csatlakozó pontjának a szerepe homályban maradna. De megnyugtadjuk a kedves olvasót, hogy ha egy programozással foglalkozó anyagot vesz a kezébe, az pedig legalább ilyen áttekintő jelleggel a hardvert lesz kénytelen bemutatni - a valóságban ugyanis a két oldal elválaszthatatlan, csak akkor tudja valaki kezelni ezeket az eszközöket, ha a hardver és a szoftver ismereteket egyaránt elsajátította már.

Az egyik kiemelt témánk a címek és a vezérlő jelek kezelése a mikroszámítógépben. Az itt leírtak alapján lehet majd később a PC memória bővítését, kiegészítő egységeinek illesztését, beállítását megérteni, ezek az ismeretek segítenek majd az esetleges hibakeresés, javítás során is.

A mikroelektronika a kívülálló számára rendkívül bonyolult, érthetetlen világnak tűnik. A jegyzet célja az, hogy segítségével ez a világ kezelhetővé, átláthatóvá, érthetővé és barátságossá váljon. Ezt a célt csak közösen tudjuk elérni - szükség van arra is, hogy a tanulás során felvetődő észrevételekkel, megválaszolatlan kérdésekkel megkeressék az előadót, a gyakorlatvezetőt. Ezek a kérdések sokszor az oktatókat is segítik abban, hogy rájöjjenek, hogyan lehetne egy-egy tématerületet hatékonyabban feldolgozni, érthetőbben elmagyarázni. De az olvasó közreműködéséhez is.

désére nemcsak ebben az értelemben van szükség - a jegyzetben feldolgozott ismeretanyag akkor válik hasznossá, hatékonnyá, ha lesznek, akik feldolgozzák, megértik. Egy közös tevékenység indul el, amikor a jegyzet a nyomdába kerül – a szerző továbbra is nyitott a közös munkára, az együtt gondolkodásra.

A szerző már számos főiskolai jegyzetet készített, a lektor pedig évtizedek óta hűséges kritikus. Kívánjuk, hogy munkánk eredménye újabb híveket szerezzen a mikroprocesszoroknak, a mikroelektronikának, olyanokat, akik értik és használni tudják ezeket a csodálatos digitális áramköröket.

Kecskemét, 1997. december

A Szerző és a Lektor

## **Bevezetés a 2009. évi átdolgozott kiadáshoz**

10 év a technika világában jelentős időtartam, még inkább így van ez a mikroelektronikában, ahol a chipekben (Moore előrejelzése alapján) 18 havonta megduplázódik a beintegrálható tranzistorok száma. Ez a jegyzet az alapfogalmakkal foglalkozik, néhány kiválasztott konkrét áramkör példán keresztül, mégsem tehető meg 2009-ben, hogy egy-egy témakörben legalább ne utaljunk a fejlődésre, a mai helyzetre.

Az eredeti jegyzetet az időközben érkezett jelzések alapján kijavítottuk, átdolgoztuk, és kissé kibővítettük. Arra ez a jegyzet nem vállalkozik, hogy a mikroprocesszorok, memóriák, mikroszámítógépek témakör 2009. évi állapotát részleteiben feldolgozza, ez nem is feladata. De az egyes területeken szeretnénk legalább érzékeltetni, hogy egy évtized alatt milyen újdonságok jelentek meg. Ezek a jelzések az eredeti, az 1970-es években kialakított megoldások jobb megértését is segítik. Figyeljük majd meg, hogy a korszerű mikroelektronikai megoldások legtöbbször az alapötlete már az eredeti, korai áramkörökben is megbújt!

Elsősorban a mikroszámítógépek általános felépítése, a mikroprocesszorok működése, a különféle memória-áramkörök területén volt szükség a jegyzet anyagának bővítésére. Bár ez a jegyzet nem a PC felépítésével foglalkozik, szeretnénk, ha a PC összeállításakor, a BIOS „hangolásakor”, a memóriamodulok kiválasztásakor szükséges ismereteket is (legalább alapfokon) tartalmazná a jegyzetünk.

Az átdolgozás során szembesült a szerző is azzal, mekkorát változott a mikroelektronika tíz év alatt. Érdekes elgondolni, hogy tíz év múlva, a következő átdolgozáskor milyen eredményekről számolhat majd be a jegyzet!

Ezekkel a gondolatokkal bocsátjuk útjára az átdolgozott jegyzetet, remélve, hogy újabb híveket szerez a mikroszámítógépek, a mikroelektronika világának, bebizonyítva, hogy ez a világ átlátható, érthető, megismerhető.

Kecskemét, 2009. október

A Szerző

## 1. A digitális áramkörök fejlődése

A digitális áramkörök fejlesztése hatalmas ütemben folyik, a félvezető gyártók egyre újabb és újabb áramkör-típusokkal rukkolnak elő. Ez a fejlesztő munka azonban nemcsak az áramkörök logikai felépítésére terjed ki, hanem gyorsan változik a chip gyártás technológiája, a tokozás és a szerelés is. A környezetvédelem és a gazdasági gondok a tápenergia csökkentésére ösztönzik a gyártókat, ez az igény vezetett el az 5 V-nál kisebb tápfeszültségű digitális áramkörökhöz.

### 1.1. Alapanyag és technológia - néhány szó a chipek gyártásáról

Ebben a jegyzetben sajnos nincs arra mód, hogy a digitális IC-k gyártási folyamatát részletesen bemutassuk. Néhány lépését azonban leírjuk e folyamatnak, azokat, amelyekre a későbbiekben hivatkoznunk kell.

A digitális IC-k alapanyaga általában szilícium (Si), ritkábban gallium-arzenid (GaAs). Az IC gyártás két párhuzamos úton indul:

- Si (vagy GaAs) egykristályt készítenek, majd felszeletelik, így kapjuk a szeletet;
- az áramköri tervek alapján fotomaszkokat állítanak elő, annyit, ahány fotolitografikus lépéssel a chip elkészíthető.

A félvezetőszelet felületét minden gyártási lépés elején fényérzékeny anyaggal vonják be (fotoreziszt), a fotomaszkon keresztül megvilágítják, majd lemosják a fotorezisztet ott, ahol nem érte fény. A lemosott területeken a félvezető felületét szabadon eléri azok az anyagrészecskék, amikkel a kemencében bombázzák – az adalékatomok, molekulák, ionok. A megfelelő mélységű behatolás elérése után letisztítják a felületet majd ismét fotoreziszt kerül a korongra. Az egymást követő ilyen lépések eredményeképpen a félvezető vékony felületi rétegében tranzistorok, diódák, beállított értékű ellenállások alakulnak ki. A chipgyártás utolsó fázisában a félvezető felületére fémet visznek fel, a kialakuló az alumínium (a gyors processzorok esetében vörösréz) vezetékek kötik össze a korábban kialakított alkatrészeket egymással (ez a lépés a fémezés, az ehhez használt fotomaszk a fémező maszk). Bonyolultabb LSI áramkör esetén többretegű fémezést is használhatnak, kereszteződő (de elszigetelt) vezető sávokkal.

A félvezetőszeletet ezután feldarabolják, a chip-eket (lapkákat) tokozzák. A foglalatlapra helyezett chip szélein lévő csatlakozó felületekhez (pad-ek) vékony aranyvezeték csatlakoztatnak (termoemissziós kötéssel), a huzal másik végét a foglalat csatlakozó felületéhez ismét termoemissziós kötéssel csatlakoztatják, majd a tokot lezárják a felső részével.

Egyfázisúnak nevezzük az IC gyártást, ha a félvezetőszelet első megvilágításától kezdve a kész IC tokozásáig minden lépés egyetlen folyamatban készül. Ha a félkész IC-t konzerválják (előgyártmány) és csak később folytatják az elkészítését, a gyártás kétfázisú. Attól függően, hogy hol vágják ketté a gyártási folyamatot, az előgyártmányból több-kevesebb munkával alakítható ki a végleges chip.

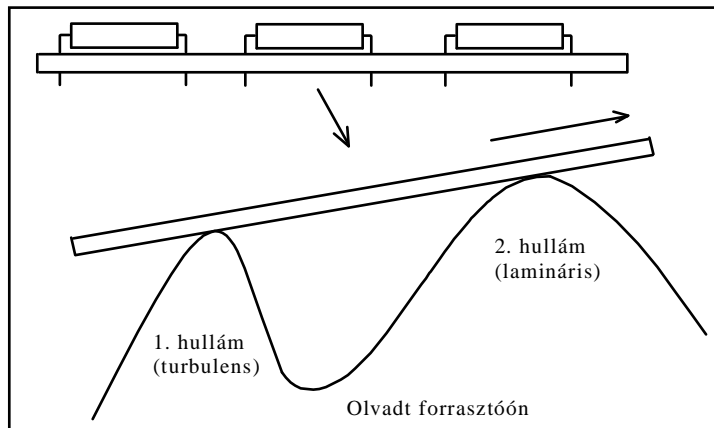
A digitális integrált áramkörök gyártása a 60-as évek közepén a kis- és közepes integráltságú (SSI-MSI) sorozatokkal indult el. A két használatos technológia a TTL és a CMOS. A nagyintegráltságú (LSI) chip-ek eleinte p csatornás MOS-FET alapon készültek (PMOS), majd az NMOS megoldás vált uralkodóvá. Később a kis fogyasztású, szimmetrikus jelalakú, széles tápfeszültségtartományú CMOS technológia fejlesztése révén az LSI elemek alaptechnológiájává vált ez a megoldás.

Az LSI áramkörök fejlődésének egyik mutatója a chip felületén alkalmazott legvékonyabb rajzolat szélessége, ami ma már 0,05  $\mu\text{m}$  alatti; egy másik fontos adat a működési sebesség (jelenleg néhány GHz); és talán a legszemléletesebb a beintegrált tranzistorfunkciók száma (a legnagyobb integráltságú mai chip-eknél ez a szám  $10^9$  feletti).

## 1.2. A tokozás változásai, szerelési megoldások

A digitális IC-ket hosszú időn át kizárólag a nyomtatott huzalozású panel (NYÁK) furatgalvanizált átmenő furataiba beforrasztható dual-in-line (DIL) tokozással gyártották. A 60 feletti kivezetés szám esetén a DIL tok már nehézkes, túl nagy méretű - helyette általában több, egymásba helyezett négyzet mentén elrendezett kivezetéseket alkalmaztak a tok alsó felületén (pin-grid-array, PGA).

Időközben a kisebb méretek, egyszerűbb gyárthatóság miatt mind népszerűbbekké váltak a furat nélküli NYÁK panelek, melyeknél az elektronikus elemek a felületen helyezkednek el

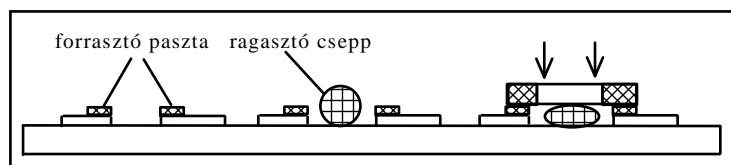


1.1. ábra.

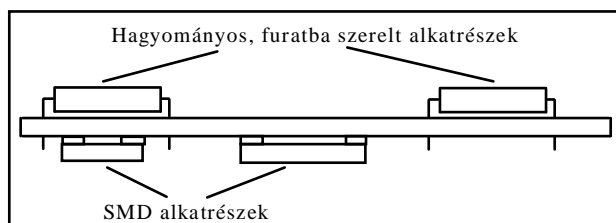
zetéseinek rugózása tartja meg szereléskor a furatokban, míg a forrasztás megtörténik. Többnyire a kettőshullámú forrasztási technikát alkalmazzák, az első, turbulens hullám mindenhol eljuttatja az önt, a második, lamináris hullám tisztára mossa a felületet, a felesleges önt eltávolítja (1.1. ábra).

Az SMD alkatrészek használatához a forrasztási területeket folyasztószerrel kell bevonni, az SMD alkatrészt erős ragasztóval kell a felülethez szorítani (1.2.

ábra). Amíg mindkét tokozási megoldást vegyesen használják, többnyire vegyesen szerelt pane-



1.2. ábra.



1.3. ábra.

lekkal készülnek a berendezések. Ekkor a forrasztási oldalra célszerű elhelyezni az SMD elemeket, megfordítva a panelt, szerelhetők az átmenő furatba ültethető elemek (1.3. ábra) - majd a kettőshullámú forrasztó berendezéssel forrasztható a panel. Az SMD elemek bemeríthetők az olvadt ónfürdőbe.

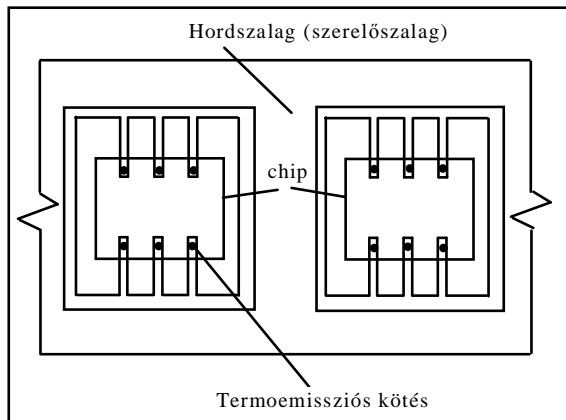
Ha a panelen csak SMD alkatrészeket helyeznek el, SMT forrasztási megoldá-

sokat lehet használni:

- gőzfázisú forrasztás: túlhevített gőztérbe helyezik a szerelt panelt, olyan anyag gőzébe, ami nem okoz korróziót; a forró gőz mindenhol behatol, megolvasztja az SMD alkatrészen lévő önt is és a panelre felvitt pasztát is, így létrejön a kötés;

– infravörös forrasztás: infralámpák közé tolják be a szerelt panelt, az infravörös fény az árnyékolt területeken nem melegít.

Egy különleges szerelési technika a szerelőszalagos (hordszalagos) chip felvitel (1.4. ábra).



1.4. ábra.

Itt a chip tokozás nélkül kerül ki a félvezető gyárból, egy hordszalag fém bajúsz-szálain lógva (a bajúsz-szálok a chip pad felületeihez vannak erősítve, a másik végük a hordszalaghoz van rögzítve). A szereléskor a megfelelő rajzolatú NYÁK mintázatra ráfektetik a chip-et és egy célszerszám elvágja a fém-szálokat - egyúttal azok végét a NYÁK mintázathoz köti, termoemissziós kötessel. Végül a chip-et fekete műgyanta cseppel takarják le, ez helyettesíti a tokozást, biztosítja a chip védelmét. Olyan eszközökben kedvelik ezt a megoldást, ahol az elektronika egyetlen IC, pl. a zsebszámolóban, karórákban.

### 1.3. A tápfeszültség változásai

A Texas Instruments, amikor a 60-as évek közepén megjelent az SN74N sorozatú SSI-MSI TTL digitális áramkörsaláddal, egyszerre több ipari szabványt is teremtett. A +5 V tápfeszültség hosszú ideig kizárólagos volt a digitális rendszerekben, néhány ipari CMOS egységben alkalmaztak csak magasabb tápfeszültséget.

A 90-es évekre jelentős változások történtek az IC gyártásban is és a környezetünkben is. Minden területen megjelent a kisebb energiafogyasztás igénye, így a digitális berendezéseknél is, ezért az IC-k tápenergia-fogyasztását csökkenteni kellett – az egyik megoldás a tápfeszültség csökkentése. Más hatások is a tápfeszültség csökkentését követelik. Így pl. az egyre finomabb rajzolatú áramköri részletek a chip-ben olyan kicsiny méretű tranzisztorokat jelentenek, melyek már nem viselik el az 5 V-ot. Jelenleg az LSI áramkörök jelentős részét 3,3 V ... 1,2 V tápfeszültségre gyártják.

A hordozható, telespes készülékek a 3,3 V-os táplálás révén hosszabb ideig képesek egy elemkészlettel működni, a hálózati egységeknek kisebb a fogyasztásuk, elmaradhat a zajos ventilátor.

A zöld személyi számítógép (Green PC) követelményeinek csak a kis tápfeszültségű áramkörök felelnek meg. A mikroprocesszorokat ma már 1 V körüli tápfeszültségre készítik.

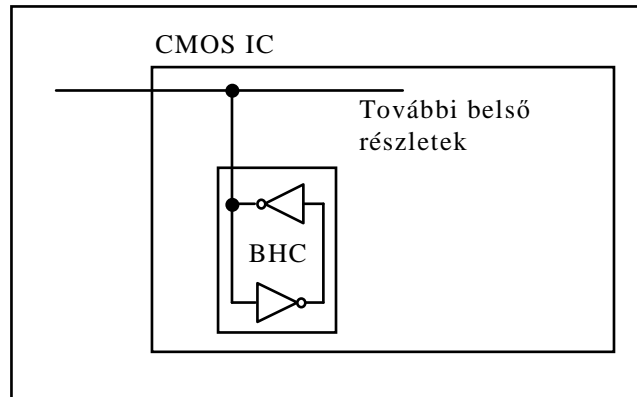
A jövő digitális készülékeiben nem lesz egységes a tápfeszültség, de ez nem okoz majd gondot, mert a mai egyetlen, központi helyzetű tápegység eltűnik a berendezésekből. Ehelyett egy előszűrt tápfeszültséget vezetnek a különféle részegységekhez és az ott telepített DC/DC konverterek helyileg állítják elő a kívánt feszültségértéket.

Az SSI-MSI digitális áramkörsaládok között is megtalálhatók a 3,3 V-os és még kisebb tápfeszültségű sorozatok, a számítógépekben a BUSZ erősítő, a BUSZ kezelő áramkör már ezekből a típusokból kerül a panelre. Az alacsony tápfeszültségű áramkörökben új áramkörtechnikai megoldásokkal is találkozunk. Az újdonságok sokszor az IC fogyasztását is csökkentik, hiszen láttuk, hogy a tápfeszültség csökkentésének ez az igény az egyik alapvető oka. Ilyen új belső részlet a CMOS bemenetekre telepített kis tárolóegység, a Bus Hold Cella (BHC, 1.5. ábra).

Működés közben ennek nincs semmilyen hatása, de ha a bemenetre nem érkezik jel (ha a bemenetre csatlakozó előző áramkör kimeneti fokozata harmadik állapotba kerül), a legutolsó bemeneti értéket őrzi a BHC, és azt küldi az áramkör bemenete felé. Ezáltal ilyen helyzetben sem válik a CMOS bemenet bekötetlenné, ami azért fontos, mert a bekötetlen CMOS bemenet véletlenszerű logikai értékeket érzékel, s ilyen körülmények között az áramkör hibás működést

produkálhat, sőt, akár tönkre is mehet. Mindezt hagyományosan a bemenetre telepített felhúzó vagy lehúzó ellenállásokkal oldották meg, de a kis tápfeszültségű, minimális fogyasztású korszerű áramköröknél ez túl nagy fogyasztású, ezért megengedhetetlen megoldás.

Egy buszvezetékre csatlakozó áramkörsoportban elegendő egyetlen olyan elemet használni, melybe beépítették a BHC egységet, hiszen a cella által tárolt logikai érték a CMOS IC bemeneti pontján is jelen van, azaz a buszvezetéken is, tehát a többi áramkör számára is biztosítja a határozott logikai értéket.



**1.5. ábra.**

Egy másik új belső áramköri részlet a POD kimenetvédő egység. Ez az áramköri részlet figyeli az IC terhelőáramát, s ha az egy meghatározott érték alá csökken, a kimeneti fokozatot harmadik állapotba, azaz lebegésre kapcsolja. Ezáltal jelentősen csökken az áramkör fogyasztása.

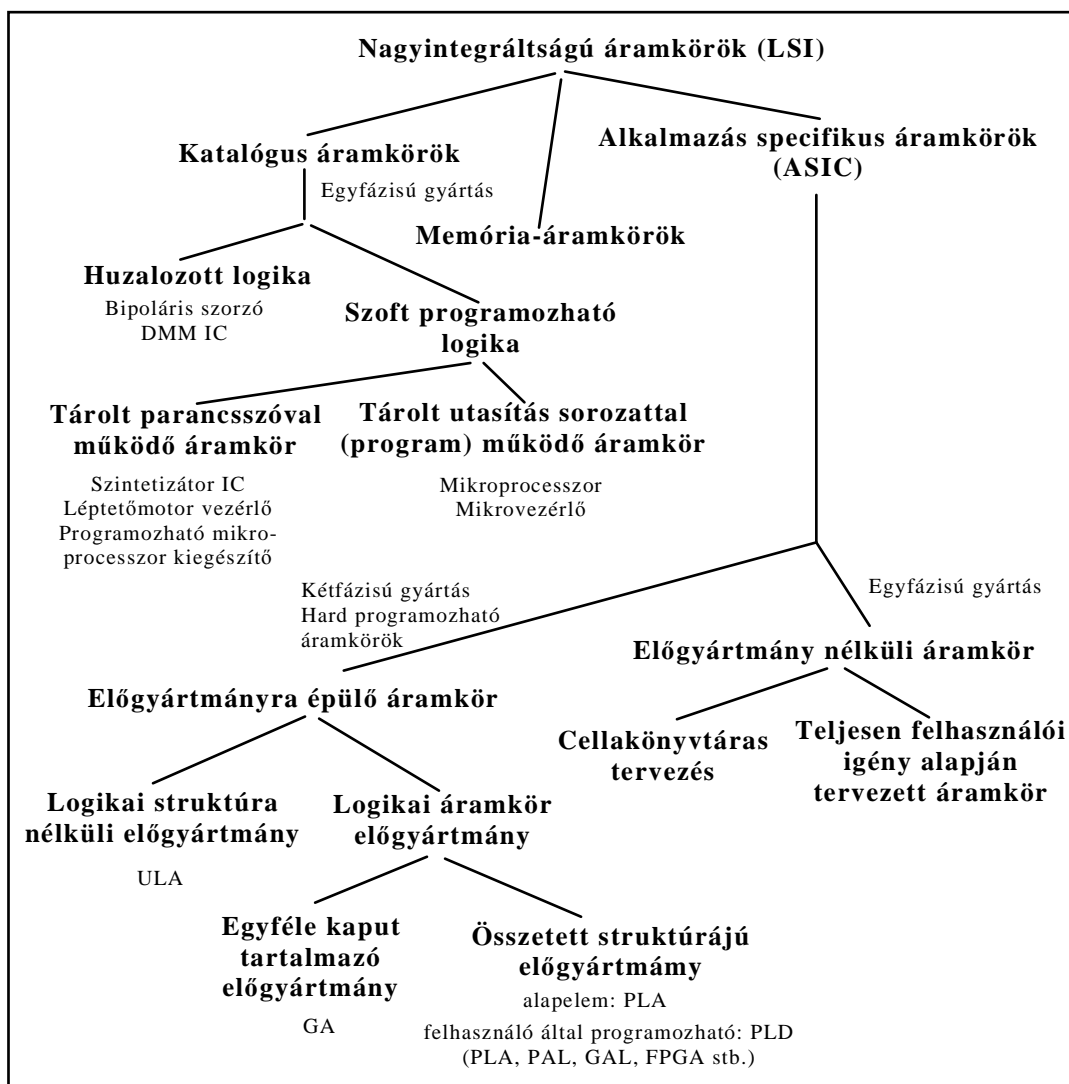
Az áramkörfejlesztés azonban itt sem áll meg, a kutatók folyamatosan keresik a még kisebb tápfeszültség alkalmazásának lehetőségét. Csak így oldható meg ugyanis az, hogy az egyre több tranzisztorfunkciót magába foglaló LSI áramkör tápteljesítmény felvétele elfogadható határok között maradjon.



## 2. Az LSI áramkörök áttekintése

Az LSI (nagyértékben integrált) áramkörök eredeti meghatározásuk szerint olyan digitális chip-ek, melyekre ezer, vagy annál több tranzisztorfunkciót integráltak. Ma ez a meghatározás már idejétmúlt. Egyrészt vannak olyan MSI elemek is, melyek több ezer tartalmaznak, másrészt az LSI áramkörök milliárdos tranzisztorszámuk már feleltette az 1000-es értéket.

1965-ben, a digitális IC megjelenésének évében, amikor analóg integrált áramköröket is csak két-három éve gyártottak, Gordon Moore, az Intel egyik alapítója egy tanulmányban azt a várakozását írta le, hogy az integrált áramkörökben a beintegrált tranzisztorok száma folyamatosan és nagy ütemben fog nőni. Ma „Moore törvényét” úgy szokták idézni, hogy 18 hónaponként megduplázódik a beintegrálható tranzisztorok száma. Ha a mikroprocesszorokat vagy a memória-áramköröket szemügyre vesszük, láthatjuk, hogy a törvény 45 éve pontosan teljesül...



2.1. ábra.

Célszerűbb tehát minőségi meghatározást adni az LSI áramkörökre: olyan IC-k, melyek digitális rendszer-elemeket vagy komplett rendszereket tartalmaznak. Az LSI áramkörök felosztását a 2.1. ábra mutatja be. Az alábbiakban az ábra tételeit vesszük sorra.

A *katalógus áramkörök* esetében az áramkör tervezése, fejlesztése az IC gyártó saját fejlesztői ötletéből indul ki, maga a cég véli úgy, hogy az új áramkör piacképes lesz. Megtörténik a tervezés, elkészül az első sorozat, azt szétküldik a forgalmazóknak, és a cég katalógusaiban megjelennek az új áramköröket bemutató oldalak. A felhasználók tehát a friss katalógusból, a gyártmánytájékoztatóból ismerik meg az IC-t. Ha valóban sikeres a fogadtatás, hosszabb időn át gyártásban maradhat az áramkör, ha nem fogy megfelelő mértékben, leállnak a gyártásával. A katalógus áramkörök egyfázisú gyártással készülnek. Ez azt jelenti, hogy a gyártósor elején behelyezik a „nyers”, azaz áramköri elemeket nem tartalmazó félvezető szeleteket a berendezésbe, s amikor a szeleteket végül kiemelik, a teljesen kész áramkörök találhatók rajta.

A *huzalozott logika* jelentése: az IC egyetlen, változtathatatlan, rögzített logikai működésre alkalmas, logikai felépítése a gyártás során kialakult és nem módosítható. Ilyen LSI áramkörök a digitális multiméterek IC-i (DMM IC), a bipoláris szorozóáramkörök, különféle függvényképző áramkörök stb.

A *programozható logika*, pontosabban ami itt szerepel, az a *szoft-programozható logika*, olyan LSI áramkör, mely szoftver úton állítható be egy konkrét működésre. (A továbbiakban találkozunk azokkal az áramkörökkel, melyek végleges működését az IC fizikai szerkezetének megváltoztatásával lehet beállítani, azok lesznek a hard-programozható megoldások.) A szoft programozható áramkörök egyes részletei többféle működésre, üzemmódra is alkalmasak, ezek egyikét egy vagy több vezérlőbit megfelelő beállításával lehet elérni. Ha a beállítás egy *parancsszó-tároló* (regiszter) betöltésével történik, az újabb parancsszó betöltéséig az IC az előírt működést végzi. Minden beállítható részlethez tartoznak vezérlőbitek, amelyeket parancsszó-regiszterekbe rendeznek a fejlesztők. Ilyen felépítésű áramkörök a mikroprocesszorok programozható kiegészítői (PIO, SIO, CTC, USART, DMA vezérlő stb.); a hangszintetizátor IC-k stb.

Más esetekben a szoft programozható áramkör kötött periódusidővel, ciklikusan beolvas egy-egy parancsszót (utasítást), amit az utasítástárolóba helyez, onnan az utasításdekóderbe továbbítja, ami értelmezi az utasítás biteit. A sorozatos beolvasás érdekében az *utasítás-sorozat* (a *programot*) elő kell készíteni és tárolni kell. A működést vezérlő utasítás-sorozat, a program tárolható az IC-n kívül (*mikroprocesszorok*) vagy az áramkörön belül (*mikrovezérlők*).

Az *alkalmazás specifikus áramkörök* (Application Specific IC, *ASIC*) logikai felépítését a felhasználó igényei szerint alakítják ki. (Ennek az áramkörcsoportnak az elnevezése nem egységes, és az ide sorolható áramköröket is esetenként eltérő nevekkkel illetik.) Két módon valósulhat meg az ASIC áramkör. A szokásosabb megoldás az, hogy egy félkész áramkört, előgyártmányt készít a gyártó (ez az előgyártmány tulajdonképpen így katalógus áramkör jellegű), a végleges logikai struktúrát később, egy elkülönült folyamat során, hard programozással alakítja ki vagy ismét a gyártó, vagy a felhasználó (utóbbi esetben az IC-t *PLD*-nek nevezik).

A vázolt megoldások sokszínűsége mögött anyagi megfontolásokat találunk. A legolcsóbbak a katalógus áramkörök, hiszen többnyire igen nagy darabszámban készülnek. A gyártási lépések, a tokozás, a tesztelés ára természetesen azonos, akár katalógus áramkörrel van szó, akár ASIC elemről. A tervezési folyamat, valamint a gyártás során használt maszkok előállításának ára azonban a katalógus áramköröknél a sokszor milliós darabszám miatt egy-egy áramkörre vetítve csak kis további költséget jelent. A felhasználói igények alapján készülő áramköröknél a darabszám többnyire jóval kisebb, így az egyedi tervezés, fejlesztés árából származó költségösszetevő jóval magasabb. Ugyanakkor a katalógus áramkör sokszor csak többé-kevésbé felel meg a felhasználó igényeinek, az ASIC IC teljesen az alkalmazáshoz illeszkedik.

Az *előgyártmány* (amiből hard programozás után lesz végleges, működő logikai elem), lényegében egy félig-meddig elkészült digitális chip. Már tudjuk, hogy minden IC (akár egy NAND kapu, akár egy mikroprocesszor, akár DRAM legyen is az), a szilícium felületi rétegében kialakított tranzistorokból és az azokat összekötő fémcsíkokból áll. A gyártási folyamat ezért több helyen megszakítható, arról kell gondoskodni, hogy a már elkészült elemek ne károsodjanak hosszabb raktározás alatt sem, majd a gyártás következő lépései később folytathatóak legyenek!

Minél későbbi fázisban kerül ki az előgyártmány a gyártósorról, a tervezési költségek annál kedvezőbben alakulnak, hiszen mind több lépés fejlesztési kiadásai a nagyszámú előgyártmányra oszlanak el. A felhasználói igényeket a végső lépések valósítják meg, azok minél kevesebben vannak, annál olcsóbbak az áramkörök. Ugyanakkor, ha a gyártási folyamat elején emeljük ki az előgyártmányt, szinte tetszőleges végső logikai hálózat kialakítható lesz rajta. Ha már kapukat, vagy összetettebb egységeket képeztek ki és így kerül raktárba az előgyártmány, akkor már kevésbé rugalmasan lehet felhasználni, a már elkészült logikai részleteken utólag nem lehet változtatni. Mindez magyarázza, miért van igen sokféle ASIC áramkör, annak érdekében, hogy a felhasználó mindig tudjon optimális megoldást találni a fejlesztéséhez.

Az előgyártmány lehet *logikai struktúra nélküli* (ULA, Uncommitted Logic Array, elkötelezetlen logikai mező); a „félkész” áramkör ebben az esetben csak ellenállásokat, diódákat, tranzistorokat tartalmaz, egymástól független elemekként (az ULA ezért általában akár digitális, akár analóg, akár vegyes áramkör kialakítására felhasználható). Az alkotóelemek összekötése a felületre felvitt alumínium összekötésekkel történik (a *fémező maszknak* megfelelően), ezért az ULA-t csak a félvezető gyártó képes programozni (maszk-programozás). A digitális áramkörök világában az ULA elemeket ma már nem használják.

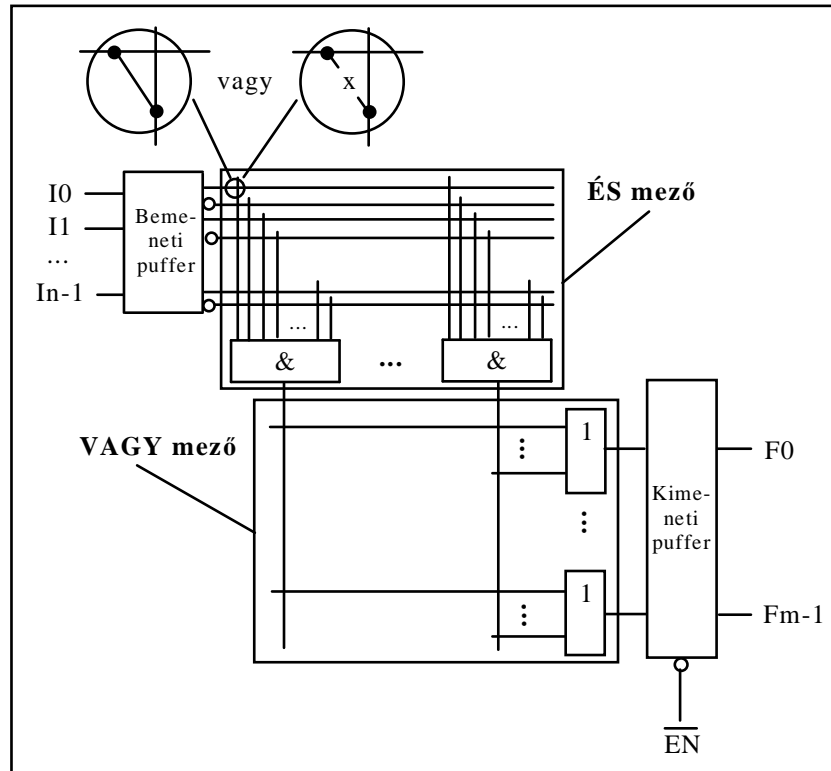
A *logikai struktúrával* rendelkező előgyártmányok legegyszerűbb esetében egy univerzális logikai kapuból alakítanak ki az előgyártmányon egy mátrixot (GA, Gate Array, kapu mező), pl. 1000 db 10 bemenetű NAND kapuból. (Mivel az ASIC áramkörök többségénél valamilyen elem, ebben az esetben kapu, nagyszámban helyezkedik el a felületen, az IC megnevezésében az azonos, nagyszámú elemből létrejövő felületre utaló angol „array” szó gyakran megjelenik. Egyébként, ha egy IC-tokban pl. hat dióda található, az áramkör az angol nyelvű katalógusban így szerepel: „Diode Array”. Ennek a szónak ebben a használatban még nem alakult ki magyar megfelelője, esetenként a szótárból kiolvasható „mező” szóval fordítjuk, vagy a „mátrix” szóval, de mindenképpen tudnunk kell, hogy ez a „mező”, „mátrix” a leírt értelmet hordozza!) A kapuk belső kialakítása már az előgyártmányon is készen van, de egymással vagy a be/kimeneti pontokkal nincsenek összekötve. Ezek a hiányzó kötések ismét alumíniummal, maszk-programozással hozhatók létre a félvezető gyártónál, a felhasználó igényeinek megfelelően.

Az *összetett logikai struktúrájú*, hard programozható előgyártmányok alaptípusa a PLA (Programmable Logic Array), felépítését a **2.2. ábra** szemlélteti. A PLA egy klasszikus kétszintes logikai hálózat, melyben programozható az ÉS mező is és a VAGY mező is. Miután a klasszikus kétszintes logikai hálózat a logikai függvény minterm alakjának a közvetlen ábrázolását jeleníti meg, természetesen bármilyen logikai hálózat kialakítható ilyen módon. Ami korlátot jelenthet: az ÉS kapuk száma, a bemenetek száma, a VAGY kapuk száma, a bemenetek száma, az IC bemeneteinek és kimeneteinek száma. A PLA-ban programozható az áramkörre érkező bemenőjelek és az ÉS kapuk kapcsolata (azaz az ÉS terület) is, és az ÉS kapuk kimeneteinek és a VAGY kapuk bemeneteinek kapcsolata (VAGY terület) is.

Az első PLA-k maszk-programozhatók voltak, csak később jelentek meg a felhasználó által programozható változatok, amiket megjelenésükkor FPLA-nak, Field PLA-nak neveztek el. (Az ASIC áramkörök világában a „Field” szó arra utal, hogy a felhasználónál történik a kezelés, esetünkben a végleges alakra programozás, tehát nem a gyártónál). Az FPLA programozatlan előgyártmány, amit tokozva kiszállít a gyártó, a felhasználó megvásárolja és egy programozó készülékkel maga tölti be a programot. A későbbiekben a PLA maszk-programozható megoldásának gyártása megszűnt, s a felhasználó által programozható kivitel nevéből az F betű elkopott.

Hamarosan bővülni kezdett ez az áramkör csoport, megjelent a rögzített VAGY területű PAL, a kibővített kimeneti képességű GAL, majd az összetettebb struktúrájú áramkörök sorozata is. A felhasználó által hard programozható logikai elemek (PLD, Programmable Logic Device) fő típusaival a jegyzet 11. fejezetében fogunk részletesen megismerkedni. Jelenleg a legnépszerűbb programozható logikai áramkörök az FPGA elemek, melyekkel akár mikroprocesszort is lehet saját igényeink szerint készíteni.

Az ASIC áramkörök is elkészíthetők egyfázisú gyártással. Az egyik ilyen megoldás a *cellakönyvtáras tervezésű* áramkör. Ekkor az egyes logikai alapegységek (számlálók, kódolók, dekódolók stb.), az SSI-MSI áramkörökre jellemző részletek teljes gyártási dokumentációját szoftver



2.2. ábra.

módon előre elkészíti a gyártó, ezek az egységek a cellák. A felhasználó által kért logikai áramkört ezekből a cellákból rakják össze mozaikszerűen és a tárolt gyártási információkból elkészül a tényleges chip gyártási terve. Maga az IC ezután fizikailag egyetlen gyártási folyamat során készül el.

A másik egyfázisú ASIC gyártási megoldás a *teljesen felhasználói igények alapján* történő tervezés, amikor az LSI áramkör minden részletét a felhasználó kérésének megfelelően tervezik meg. Ez a megoldás csak jelentős darabszámok esetén lehet kifizetődő, hiszen ebben az esetben az áramköri tervezés, a gyártáshoz szükséges előkészítő lépések, maszkgyártás mind a felhasználó igényeinek megfelelően lesznek végrehajtva.

A nagy integráltságú logikai áramkörök fontos csoportját alkotják a memória IC-k. Mivel ezek az áramkörök felépítésük alapján a 2.1. ábra különféle pontjain jelennének meg, célszerűbb csak utalni rájuk, ahogyan az ábrán is tettük. A memória-áramkörök típusaival, azok felépítésével és működésével a 4. fejezet foglalkozik.

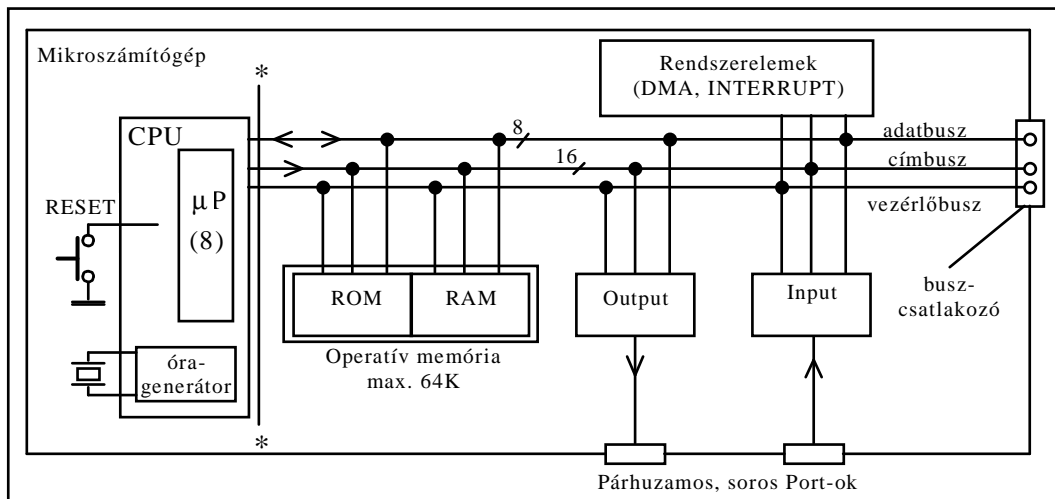
### 3. A mikroszámítógép általános felépítése, működése

Mikroszámítógépnek tekintünk minden olyan digitális készüléket, mely mikroprocesszorra, mikrovezérlőre épül. Mikroszámítógép a PC, de mikroszámítógépet találunk az egérben, a klaviatúrában, a rádiótelefonban, a mosógépben, a telefax készülékben is. A PC-k, a munkaállomások egyre nagyobb szóhosszúságú, egyre gyorsabb mikroprocesszorokkal épülnek. Ezeket Magyarországra készen vagy szerelt egységekként importálják. A mikrovezérlőkkel más a helyzet, ezekkel itthon is folyamatosan fejlesztenek termékeket, eszközöket. A mikrovezérlőkkel ezért további tantárgyakban részletesen megismerkedünk majd.

A beépített (embedded) fedélzeti mikroszámítógépek többsége ma is nyolcbites. Mindez indokolja, hogy a nyolcbites processzorok felépítését, működését részletesen feldolgozzuk. Ezek a mikroprocesszorok még teljesen áttekinthetők, működésük a legapróbb részletig megismerhető, követhető. Felépítésük, utasítás készletük megértése alapot ad a mai 16, 32 bites eszközök megismeréséhez, megértéséhez. Az alapfogalmakat mikroprocesszoros mikroszámítógép vizsgálatán keresztül ismerjük majd meg.

#### 3.1. A mikroprocesszoros mikroszámítógép felépítése

A nyolcbites mikroprocesszorok 16 bites címeket használnak, ami 64 Ki memória közvetlen megcímezését teszi lehetővé. A mikroszámítógép általános felépítését a **3.1. ábra** mutatja be.



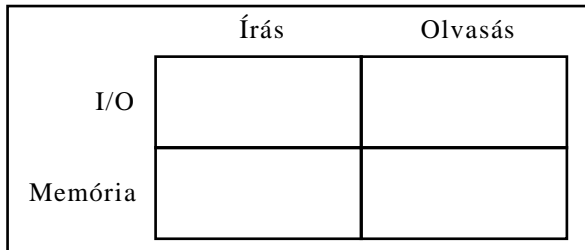
**3.1. ábra.**

A mikroszámítógép központi végrehajtó egysége (Central Processing Unit, CPU) feladata a hármas buszrendszer kezelése (kétirányú nyolc bites adatbusz, egyirányú 16 bites címbusz és vezérlőbusz). A külvilággal a bemeneti (Input) és a kimeneti (Output) egységeken, Portokon át tartja a kapcsolatot a mikroszámítógép (ezeket általában I/O elemeknek, I/O Portoknak hívjuk). A csatlakozó-megoldásaik többnyire szabványosak (soros Port, párhuzamos Port), a mikroszámítógépben belül az I/O elemek a buszrendszerre csatlakoznak. Szintén a három buszhoz kapcsolódik az operatív tár, mely kétféle memória áramkörből épül fel; a csak olvasható (de nem illanó tartalmú) ROM memóriákból és az írható-olvasható, de tápfeszültség nélkül a tartalmukat elvesztő (illanó) RAM áramkörökből. A nyolcbites mikroszámítógépekben a rögzített programokat ROM-ban tárolják (onnan futtatja a processzor a programokat), a RAM programfejlesztésre, adattárolásra szolgál.

A mikroszámítógépben esetenként további rendszerelemek is lehetnek, pl. megszakítás-vezérlő, DMA kezelő stb.

A program, az utasítássorozat az operatív tárban található, onnan olvassa be a CPU egymás után a végrehajtandó utasítás elemeket. A buszrendszer kezelésén, az utasítások beolvasásán és végrehajtásán kívül egyéb feladatai is vannak a CPU-nak. Ez állítja elő a mikroszámítógép órajelét, ez oldja meg az alaphelyzetbe állítást (RESET) a tápfeszültség bekapcsolásakor, esetenként nyomógombos RESET kéréskor illetve logikai RESET jel hatására. A CPU-ban találjuk meg a mikroprocesszort. Van olyan  $\mu P$ , amelyik önmagában komplett CPU, de a legtöbb esetben több-kevesebb külső áramkörrel kell kiegészíteni a mikroprocesszort s csak így alakul ki a működőképes CPU.

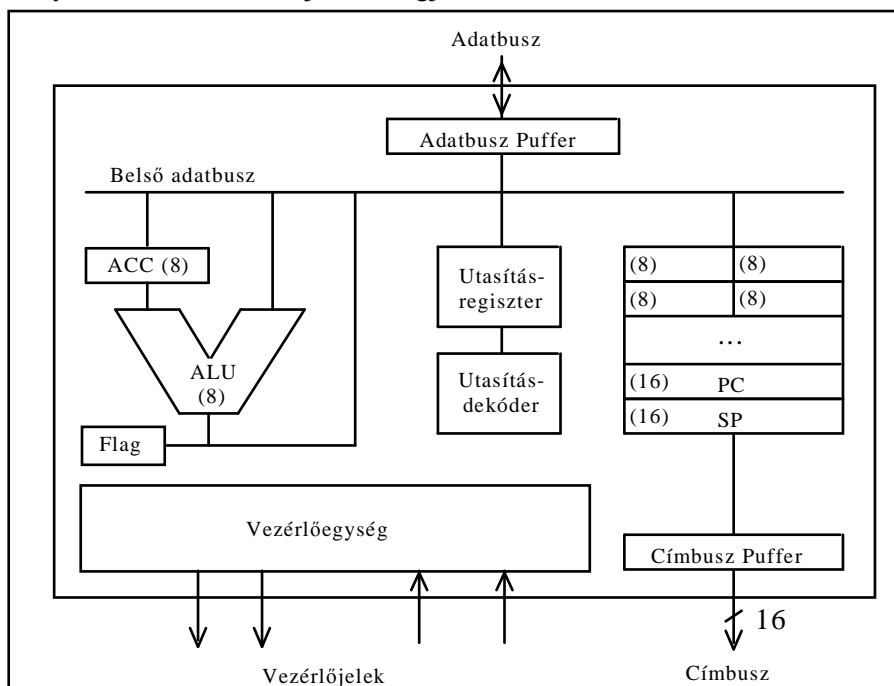
A mikroszámítógépek egy részénél a buszrendszert csatlakozókra kivezetik s így a felhasználó a buszokra csatlakozó elemekkel tudja bővíteni a gépet (nyitott felépítésű mikrogép). Ha nincs buszcsatlakozó, a mikroszámítógép zárt felépítésű.



3.2. ábra.

A mikroszámítógépben a CPU-t és környezetét a 3.1. ábrán egy \*-gal jelölt vonal választja el. Hardver szempontból a nyolcbites mikroszámítógép egy működési alapegysége, ún. gépi ciklusa az a működési esemény, melynek során ezen a felületen át a CPU és környezete között egy adat átadása megtörténik, azaz a CPU egy adatot ki- (írd) vagy fogad (beolvas). Az adat-

mozgás egyik részvevője minden esetben a CPU, a másik fél lehet memória vagy I/O elem. A négy legfontosabb működési eseményt a 3.2. ábra szerint lehet ábrázolni, a rekeszekben az egyes eseményeket kiváltó vezérlő jeleket fogjuk később feltüntetni.



3.3. ábra.

A 3.3. ábrán a mikroprocesszorok legfontosabb belső elemeit láthatjuk. A program utasításai az utasítás regiszterbe, onnan az utasítás dekódolóba jutnak; a dekódolt utasítások alapján működik a vezérlő egység. Az adatok feldolgozását, az aritmetikai és logikai műveleteket az ALU (aritmetikai-logikai egység) végzi el. Az adatokat átmenetileg a  $\mu P$  regisztereiben lehet tárolni, ezek között kitüntetett szerepe van az akkumulátornak (A). Az eredmény egy biten kifejezhető jellemzőit tárolja a Flag regiszter.

Egy-egy flag (jelzőbit) jelezheti hogy az eredmény zérus vagy sem, hogy átvitellet jött-e létre vagy sem, hogy pozitív vagy negatív, hogy páros vagy páratlan paritású stb.

Az operatív tár címzésére a mikroprocesszor a PC-t (Program Counter) használja, ami 16 bites egység. Egy másik címzőegységet is látunk az ábrán, az SP (Stack Pointer) a veremtár címzésére szolgál.

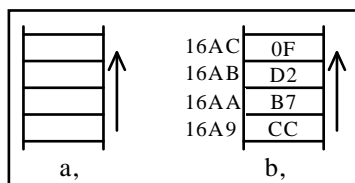
A mikroszámítógép teljesen szinkron működésű digitális egység, a leggyorsabban változó jele az órajel. Az órajel periódusideje az óraciklus, ez a legrövidebb időtartam, a legrövidebb esemény a mikroszámítógép működése folyamán. Az órajel frekvenciájának természetesen van felső határa, az NMOS processzoroknál többnyire alsó határértéket is előír a gyártó, a CMOS elemek esetén már tetszőlegesen lassú órajellel is működőképese az áramkörök.

A mikroszámítógép már említett működési alapegysége a gépi ciklus. Egy gépi ciklus alatt egy adat lép át a 3.1. ábrán csillaggal jelzett felületen, vagyis a CPU és környezete között. A négy legfontosabb gépi ciklus: a memóriaolvasás, a memóriaírás, az I/O olvasás és az I/O írás. Sokszor külön kezelik az utasításkód beolvasására szolgáló memóriaolvasási gépi ciklust (M1). További gépi ciklusok a verem olvasás, verem írás és a megszakítás-elfogadás gépi ciklus.

A mikroszámítógép leghosszabb működési egysége az utasításciklus. Annyi különféle utasításciklusa van a mikrogépnek, ahány utasításból a mikroprocesszor utasítás készlete áll. A nyolcbites processzorok utasítás készlete általában 70 ... 150 utasítást tartalmaz (normál utasításkészletű processzor). Jelentősen kisebb az utasításkészlete a csökkentett utasításkészletű (RISC) processzoroknak. Az utasításciklus magába foglalja az utasítás beolvasását, értelmezését és teljes végrehajtását. Az utasításciklus annyi gépi ciklusból áll, ahány adatmozgás szükséges a végrehajtásához.

### 3.2. A program és az adatok elhelyezkedése a memóriában

A mikroprocesszorra épülő mikroszámítógépek Neumann elrendezésűek a memóriák kialakítása szempontjából - azaz a programelemek és az adatok ugyanabban a memóriában találhatók (a mikrovezérlők a Harvard architektúrát követik, elkülönülő program memóriát és adatmemóriát használva).



3.4. ábra.

A programmemória egy-egy részletét a 3.4.a. ábra szerint lehet ábrázolni, a b, részleten néhány rekesz tartalmát és címét is feltüntettük. A memória címeket és tartalmakat mindig hexadecimális értékként fogjuk kezelni, ezért ezt nem is jelöljük külön. A bemutatott részletek egy nyolcbites mikrogépből valók, ezért egy byte méretűek a rekeszek, a címek pedig négy hexadecimális helyértékűek.

A nyolcbites mikroprocesszorok memóriájában az adatok lehetnek nyolcbitesek (egy Byte) vagy 16 bitesek (két Byte), 16 bites pl. egy cím. Ha egy Byte-os adatok vannak a bemutatott memória-részletben, akkor a tartalom így értelmezhető:

- a 16A9h címen CCh,
- a 16AAh címen B7h,
- a 16ABh címen D2h

stb. A nyolcbites gépek memóriájában 16 bites (két Byte-os) adat is elhelyezhető, ekkor tudni kell, milyen sorrendben helyezi el a Byte-okat a CPU a memóriában. Ez processzorfüggő, pl. az i8080, az i8085, a Z80 mikroprocesszorok a 16 bites adat vagy cím felső Byte-ját helyezik magasabb címre és az alsó Byte kerül a kisebb című rekeszbe. Ha az előbbi ábrán az 16AA címen egy 16 bites értéket tárolnak, a teljes, két Byte-os alak a következő:

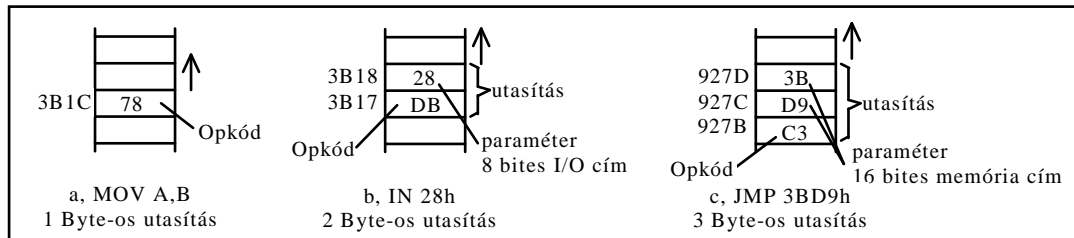
16AA címen D2B7h .

A Motorola 6800 mikroprocesszora a 16 bites érték felső Byte-ját tárolja a kisebb címen, az alsó Byte-ot a magasabb címértéken, így a 16AAh címről ez a processzor ilyen két Byte-os értéket olvas ki:

16AAh címen B7D2h .

Az utasítás a memóriában legalább egy rekeszt elfoglal (egy rekeszben több utasítás nem fordul elő), de az utasítások nagy része egynél több rekeszt igényel. Ahány memóriarekesz szükséges az utasítás tárolásához, az utasítás annyi Byte-os.

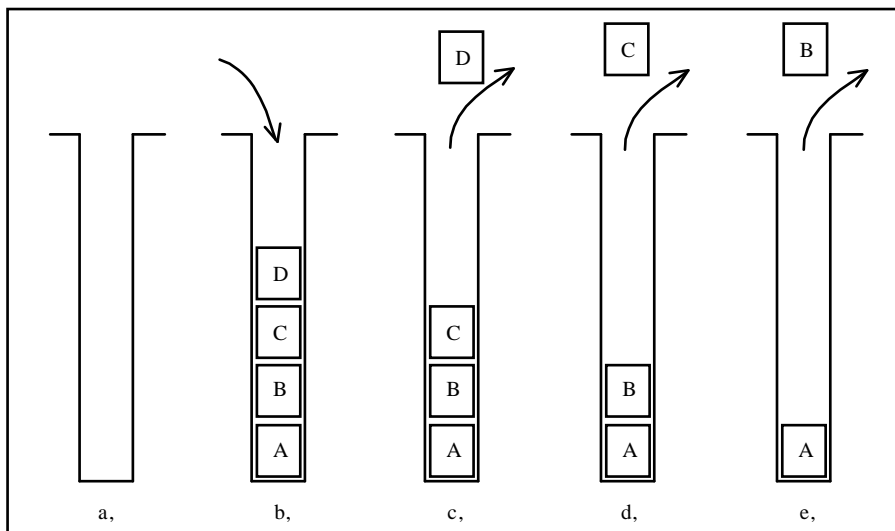
Az utasítás fő részei: a műveleti kód (operációs kód, opkód) és a paraméterek. Sok különféle utasítással találkozunk, vannak, amelyeknek nincs paraméterük. A 8080 minden utasítása egy Byte-os opkóddal rendelkezik, más mikroprocesszornál az opkód több Byte is lehet (pl. Z80). A



3.5. ábra

memóriában az utasítás mindig az opkóddal kezdődik, az esetleges paraméterek mögötte helyezkednek el (ha a paraméter 16 bites szám, a Byte sorrendre az előbb elmondottak érvényesek).

Egy utasítás végrehajtása legalább annyi gépi ciklust igényel, ahány Byte-os, hiszen az utasítás elemeit be kell olvasnia a CPU-nak. A **3.5. ábrán** látható néhány utasítás elhelyezkedése a memóriában (a 8080 utasításkészlete alapján). Az opkód beolvasó gépi ciklus M1, a további utasítás elem beolvasó gépi ciklus már csak memóriaelolvasás. Ha az utasítás végrehajtásához nincs szükség a mikroprocesszor és környezete között további adatmozgásra, az utasítás végrehajtása több gépi ciklust nem is igényel. Ha azonban a végrehajtáshoz adatmozgás is kell, azok



3.6. ábra.

száma határozza meg a további gépi ciklus számot.

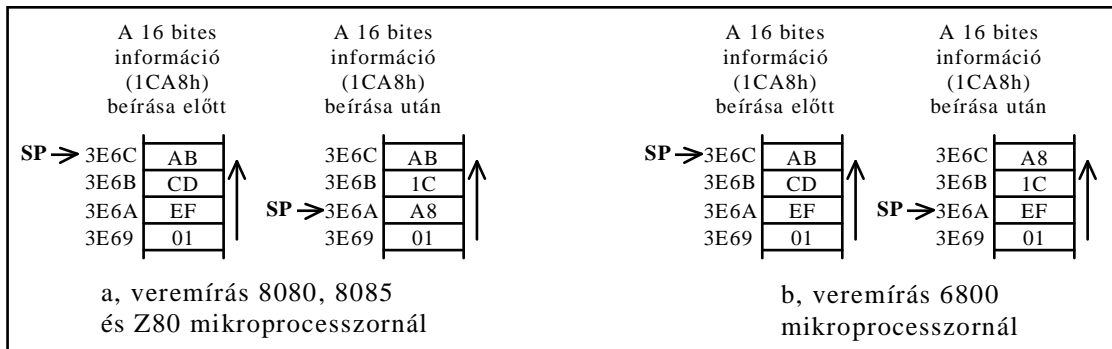
A verem (stack) jellegű memóriakezelés igen sajátos működés. A működés megértéséhez képzeljünk el egy szűk vermet, melybe ládákat engedünk le (**3.6. ábra**). Először az A-t, majd a B-t, azt követően a C-t, végül a D-t. Ha most kiemelünk

egyet, az természetesen a D lesz - egyrészt a D kiválasztásához nincs szükség semmilyen intézkedésre, hiszen legfelül lévén ez adódik; másrészt nincs is mód a D helyett ebben a helyzetben a C-t vagy a B-t kiemelni - csak a D után, szép sorban! Ezt a memória típust LIFO (Last-In-First Out, Utolsóként-Be-Elsőként-Ki) névvel illetjük, a leírt működésre utalva. A mikroszámítógép veremtára is hasonlóképpen működik, a veremtár mutató (Stack Pointer, SP) jelzi, meddig van feltöltve a verem. A mikroszámítógépek a vermet általában a magasabb címek felől az alacsonyabbak felé töltik, és fordított irányba ürítik, az SP a verem írásakor illetve olvasásakor automatikusan kezelődik.



A **3.7.a**, ábrán a 8080, 8085, Z80 veremkezelése látható - ezeknél veremíráskor előbb csökken eggyel az SP (dekrementálódik), majd erre a címre tölti be a processzor az adatot; olvasáskor onnan olvas, ahova az SP éppen mutat és a kiolvasás után növeli SP tartalmát eggyel (inkrementálja). Az ábrán az 1CA8h cím íródott be a verembe.

A **3.7.b**, ábrán a 6800 veremkezelése figyelhető meg - íráskor ez az SP által mutatott címre ír és ezután dekrementálja az SP tartalmát; olvasáskor előbb inkrementálja SP értékét majd a kapott címről olvas. Az ábrán a 6800 is az 1CA8h 16 bites értéket, pl. címet helyezte be a verembe.



3.7. ábra.

### 3.3. Az utasítás-végrehajtás menete

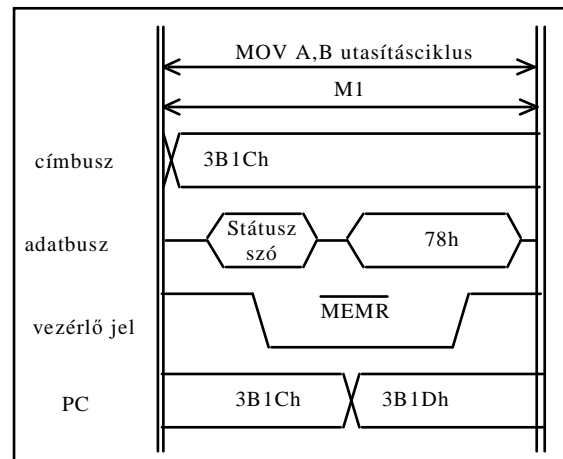
A  $\mu P$  utasításkészletére jellemző, hogy a kétoperandusos működéseket milyen felépítésű utasítás rendeli el. A korai elektronikus számítógépeknél a kétoperandusos utasítás négy címet tartalmazott; szerepelt benne az első és a második operandus címe, az eredményt befogadó regiszter címe és végül a következő utasítást tartalmazó rekesz címe. A programozók azonban a négycímes utasítás lehetőségeit nem használták ki, az utasításokat sorban helyezték el a memóriában, az egyik operandust egy kitüntetett regiszterben helyezték el (az akkumulátorban), s az eredményt is ide kérték. Az utasítás felépítése ezért átalakult, mára az egycímes utasítás a jellemző; a kétoperandusos utasításkor az egyik operandus az akkumulátorban van, a másik címét tartalmazza az utasítás, az eredmény az akkumulátorba kerül, s végül a következő utasítás a programtár következő címéről kerül elő. Ezt az utasítás-végrehajtási sorrendet normál vagy természetes sorrendnek nevezzük.

A *normál sorrendű utasítás-végrehajtást* a mikroprocesszorba beépített automatizmus teszi lehetővé. Minden utasításelemet beolvasó gépi ciklusban automatikusan nő eggyel (inkrementálódik) a PC, a programszámláló, így a gépi ciklus végén minden esetben a következő utasításelemre mutat a PC. (Utasításelem az opkód és az utasítás esetleges többi Byte-ja, a paraméter egy-egy Byte-ja is.) A **3.5.a**, ábrán látható adatmozgató utasítás, a MOV A,B a B regiszter tartalmát tölti át az A-ba. Ez egy Byte-os egyciklusos utasítás, egyetlen gépi ciklus alatt megtörténik az utasítás beolvasása, dekódolása és végrehajtása (**3.8. ábra**).

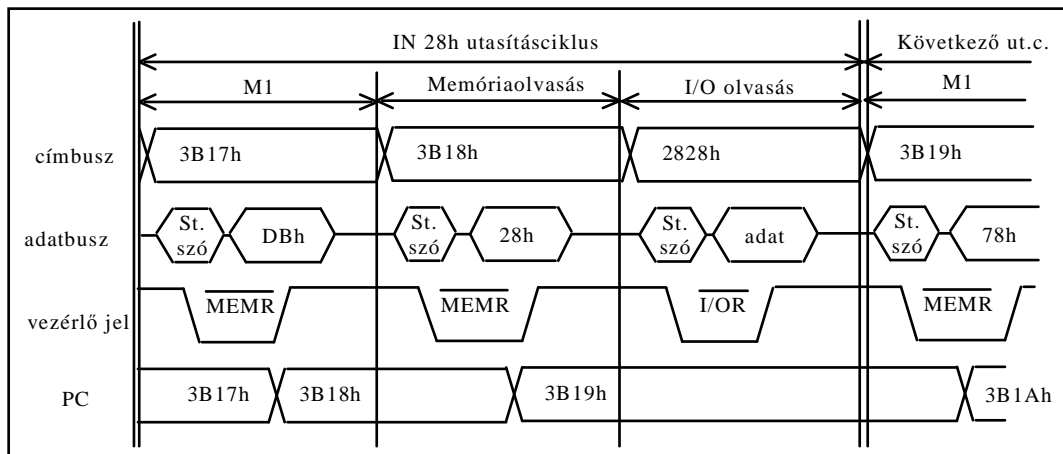
A **3.5.b**, ábrán egy bemeneti elemet (INPUT) olvasó utasítás, az IN 28h látható. Ez két Byte-os, három ciklusos utasítás (**3.9. ábra**). Az utasításelemek beolvasása két gépi ciklust igényel (M1 majd memóriaolvasás), végül még egy adatot kell mozgatni a CPU és egy bemeneti elem között, ez egy I/O olvasás gépi ciklusban megy végbe. A **3.8.** és a **3.9. ábrán** 8080 utasításciklusok láthatók. Ez a  $\mu P$  minden gépi ciklus elején kiküld egy státusz-szót az adatbusz vezetékein, ez is látszik az ábrákon. A státusz-szó informálja a mikroszámítógép többi egységét arról, hogy a 8080 milyen belső működést, milyen gépi ciklust indított el.

(Más mikroprocesszorok közvetlenül, vezérlőjelekkel informálják a mikroszámítógépben lévő többi elemet az éppen futó gépi ciklus jellegéről.)

A programokat nem lehet egyetlen folyamatos, lineáris utasítás-sorozatként megírni, a normál sorrendtől ezért rendszeresen el kell térni. A normál sorrendtől a feladat megoldása érdekében a  $\mu P$  szoftver és hardver úton is eltéríthető. Az eltérítés szoftver eszközei a vezérlésátadó utasítások (ugrás, szubrutinra ugrás, szubrutinból visszatérítés) és a HALT; a hardver eltérítési lehetőségek a mikroprocesszor vezérlőbemeneteire vezetett logikai jelekkel érhetők el (RESET, megszakítás, READY/WAIT működés, HOLD kérés). A következőkben ezekkel a lehetőségekkel sorra megismerkedünk.



3.8. ábra.



3.9. ábra.

Az ugró utasítás (jump, JMP) a 8 bites mikroprocesszoroknál alapesetben (közvetlen címzés esetén) 3 Byte-os, az opkód (8080 esetén pl. C3) és a 16 bites cím található benne (ld. 3.5.c, ábra). Az utasítás beolvasását követően a PC felülíródik az ugrási címmel - a következő M1 ciklusban ez a cím lép ki a címbuszra.

A feltételes ugró utasítás is így működik, ha a benne előírt feltétel teljesül, ha nem, akkor a PC tartalma nem íródik felül, ott a normál inkrementálódás eredménye marad meg. Feltételként a  $\mu P$  egyes flag-jeinek 0 vagy 1 értéke írható elő (pl. JNZ : ugrás, ha Z = 0, JC : ugrás, ha C = 1). Általában a Z (zérus érték), az S (előjel), a C (túlcsordulás) és a P (prioritás) flag használható feltételként.

Ha az ugrás után a most elhagyott programot folytatni kell a kiugrási pont utáni részével, az ugrás végrehajtása előtt biztos helyre el kell menteni a folytatás címét. Ezt automatikusan elvégzi a szubrutinra ugró (szubrutinhívó, CALL) utasítás, a visszatérési címet a verembe mentve. A CALL is három Byte-os (közvetlen címzés esetén), a 8080 esetén CD az opkódja. tartalmazza az utasítás a 16 bites szubrutin kezdőcímet is. A CALL is lehet feltétel nélküli vagy feltételes.

A szubrutin végén csak azt kell közölni a mikroprocesszorral, hogy a PC-be vissza kell tölteni a verem tetején lévő, oda utoljára beírt címet. Ezt a RET (return, visszatérés) utasítással tehetjük meg, ez is lehet feltétel nélküli vagy feltételes.

A HLT (halt, leállítás) utasítás hatására a  $\mu P$  felhagy a szoftver végrehajtásával - ebből az állapotából csak hardver úton mozdítható ki.

A RESET jel hatására a mikroprocesszor egy pontosan definiált alaphelyzetet vesz fel, annak érdekében, hogy a programfuttatás egyértelmű kiindulóponttal kezdődhessen meg. A RESET jel hatása processzoronként változó, a katalógusok precízen meghatározzák a kialakuló alaphelyzetet. A RESET egyik legfontosabb feladata, hogy meghatározza a RESET jel utáni működést (pl. a 8080, a 8085 és a Z80 a RESET után a 0000h címről utasítás beolvasással indulnak).

Minden esetben meg kell valósítani a tápfeszültség hatására automatikusan kialakuló RESET-et (bekapcsolási RESET), többnyire célszerű kézi RESET lehetőséget is biztosítani. Esetenként logikai jel bevezetésével is kiváltható az alaphelyzetbe állítás.

A megszakítás (INTERRUPT) kérő jel hatására (ha elfogadható), a  $\mu P$  a megszakítást kiszolgáló szubrutinra lép, azt végrehajtja, majd visszatér a megszakított programhoz, azt folytatja; a megszakítás tulajdonképpen egy hardveres szubrutinra ugrás. A megszakítás-kérő bemenet lehet maszkolható (utasítással tiltható vagy engedélyezhető az elfogadása) vagy nem maszkolható, amit minden esetben elfogad a processzor. A  $\mu P$  utasításciklust nem hagy félbe a megszakítás miatt, csak két utasítás közé iktatja be a megszakítást kezelő szubrutint. Ezért többnyire csak az utasításciklusok utolsó gépi ciklusában, annak utolsó óraciklusában vizsgálják meg a mikroprocesszorok a megszakítás-kérő bemenetek állapotát.

A megszakításokat a mikroszámítógépben prioritási (fontossági) sorrendbe kell rendezni. Ha egyszerre jelentkeznek, a magasabb prioritású jut érvényre; de akkor is, ha már egy megszakítási rutin fut, és újabb kérés érkezik. Ha az új kérés nem magasabb szintű a már kiszolgáltnál, nem reagál rá a  $\mu P$ . Ha a megszakítási rutinra ugráskor a CALL utasításnál több információ mentődik el, a megszakításból nem a szokásos RET utasítással kell visszatérni. A Z80 esetén is külön lezáró utasítása van a megszakítási rutinnak, de ennek más az oka (később megfjtjük ezt a titkot).

A HOLD, HLD vezérlőjel a mikroprocesszort arra kényszeríti, hogy a buszokra csatlakozó pontjait lebegtesse. Így a buszrendszerre csatlakozó egyéb elemek egymással közvetlenül tudnak kommunikálni, közvetlen memória-hozzáférés (Direct Memory Access, DMA) hozható létre. Ehhez azonban a mikroszámítógépben lennie kell egy DMA-vezérlőnek, amit előzőleg a mikroprocesszor felkészített a DMA vezérlésére.

A READY vagy WAIT jelek segítségével a mikroprocesszor a gépi ciklus belsejében képes várakozni (várakozási óraciklusok beiktatásával) valamilyen külső eseményre. Ezzel a megoldással nyújtják meg a gépi ciklusokat, pl. lassúbb memória használata vagy perifériák lassúbb működése esetén. Ez a lehetőség igen jól használható a mikroprocesszor, mikroszámítógép ellenőrzésére, tesztelésére is.

A nyolcbites mikroprocesszoroknak általában egy belső vezérlőegységük van, ez egy adott pillanatban vagy a buszrendszert kezeli, vagy egy utasítás belső végrehajtásával van elfoglalva. A  $\mu P$  működése során ezért ez a két fázis egymást váltogatva, egymás után jelentkezik. A mai, nagyobb teljesítményű mikroprocesszorok több vezérlő egységgel épülnek, a belső működés közben a memóriából előre beolvashatják a következő utasításelemeket, sőt, azokat elődekódolhatják, elő-végrehajthatják. Ezt a párhuzamosított megoldást nevezik pipe-line módszernek (az utasítás előolvasó tár a queue, az előfeldolgozó vonal a pipe-line). A pipe-line megoldás normál sorrendű programrészletnél jelentősen meggyorsítja a működést, de az eltérésekkor többletfeladatot jelent (a pipe-line regisztereket ki kell üríteni majd újra fel kell tölteni).

### 3.4. A PC, a legnépszerűbb mikroszámítógép

Az IBM 1983-ban jelent meg a PC-vel a számítógép-piacon. Ebben 16 bites mikroprocesszor működött (Intel 8086). A programokat részben ROM memóriák, részben mágneslemez tárolók őrizték, a futtatáshoz a lemezekről kiolvasták és RAM-ba (ez a főmemória, DRAM jellegű, ld. később) töltötték az utasítássorozatot! A PC tehát RAM-ból futtatja a programokat, ezért nagyméretű, gyors RAM területre van szüksége! A ROM csak a gép indulásához szükséges

utasítássorozatot, valamint az operációs rendszer és a konkrét hardver illesztését biztosító programelemeket (BIOS) tárolja.

Az eredeti IBM PC DOS operációs rendszerrel működött (amit ma sokan nem is tekintenek operációs rendszernek). A BIOS-ban csak egy rövid indítóprogram (Boot) található, ami a háttértárolót aktivizálja, ami már az első PC-knél is merevlemez volt. A programok tehát mágneslemezen tárolódnak, de onnan nem futtathatók, hiszen a lemezes egységek olvasási sebessége igen alacsony a processzorok működési sebességéhez képest, ezért kellett a futtatásra kerülő programrészeket a főmemóriába, azaz RAM-ba áttölteni. A multitasking programszervezést is biztosító Windows operációs rendszer bevezetése tovább fokozta a hardverrel szemben támasztott igényeket, az áramköri rendszer, a processzor, a memória-készlet gyorsan változott.

Még egy speciális memóriaelem található a PC-kben, egy olyan RAM, ami tartósan meg tudja őrizni a tartalmát (CMOS SRAM, ld. később). A PC működésének néhány alapadatát a felhasználó tudja beállítani (Setup információk), de ezeket nem lenne kényelmes minden gépindításkor betölteni. Hagyományosan ezeket az adatokat egy RAM tárolja tehát, amelynek azonban egy elem, egy akkumulátor vagy egy kondenzátor folyamatos tápellátást biztosít (háttértáplálás), így ez kikapcsolt állapotban sem veszíti el a beírt tartalmát.

A mai személyi számítógépek sok vonatkozásban már alig emlékeztetnek az eredeti készülékre, a működési sebesség, a tárolókapacitás növelésének (mesterségesen is) felfokozott igénye hihetetlen fejlődést idézett elő. A mai PC, mint mikroszámítógép, sok újdonságot tartalmaz az eredeti mikroszámítógépekkel összehasonlítva.

A mai mikroprocesszorok 32 vagy 64 bites adatokon végeznek műveleteket, többszörös pipe-line rendszert alkalmaznak, viszonylag hosszú programrészeket tudnak előre feldolgozni. A beintegrált tranzisztorok száma több milliárd, a gyártáskor alkalmazott vonalszélesség 50 nm alatti. A tápfeszültség 1 V körüli, sok esetben 1 V alatti, a processzor fogyasztása mégis általában eléri, sok esetben meghaladja a 100 W-ot! Ezért is alapvető fontosságú a PC-kben a processzor aktív hűtése, de egyúttal hűteni kell a tápegységet is, sokszor további ventilátorral a számítógép házát, sőt, esetenként egyéb áramköröket (buszkezelőket, memóriákat) is.

A processzorok működési frekvenciája több nagyságrenddel nőtt, ma már jelentősen meghaladja az 1 GHz-et, sokszor 2 GHz ... 3,4 GHz is használható. A mikroszámítógépben lévő alaposzcillátor nem ilyen frekvenciájú órajelet állít elő. Az alapórajel frekvenciája a processzor órajelénél lassúbb (pl. 600 MHz ... 800 MHz), ezt a processzor belsejében lévő többszöröző áramkör fogadja és az állítja elő a belső órajelet. A többszörözés pontos értékét szoftveresen vagy hardveres eszközökkel a felhasználó is be tudja állítani. A PC-ben azonban nem csak ez a két órafrekvencia található meg, mint rövidesen látjuk majd!

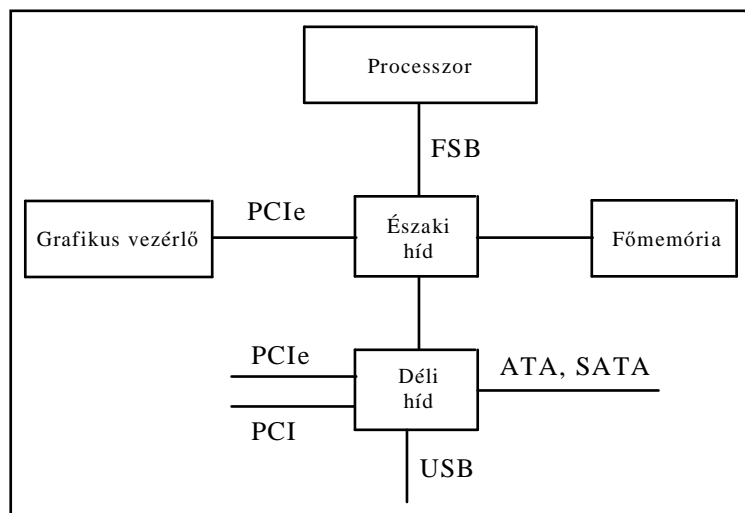
A működési sebesség növelésének egy eszköze, ha párhuzamosan több processzor is működik. Ez a gondolat eredményezte a 2009-ben piacon lévő kétmagos és négymagos processzorok kifejlesztését.

A mikroprocesszorok több GHz-es órajele mellett a memóriák működési sebessége jelentősen elmarad. A főmemória kapacitása ma már több GiByte, de az olvasási ciklus csak néhány-szor 10 ns, ami a mikroprocesszor órajeléhez képest alacsony érték. A sebességkülönbséget úgy hidalják át, hogy a pillanatnyilag használt programrészeket a fő memóriából egy kisebb, de gyorsabb memóriaegységbe, a cache-memóriába töltik át, onnan a processzor sokkal gyorsabban tudja kiolvasni a programkódot. Természetesen ez egy igen bonyolult feladat, folyamatosan figyelni kell, hogy a következő utasítások benn vannak-e a cache-ben, szükség esetén törölni kell a felesleges tartalmakat és újabbakat ismét itt kell elhelyezni. (Általában a RAM jellegű memóriákról elmondható, hogy a nagy kapacitású, lassú változatok olcsóbbak, a gyors változatok viszont kisebb kapacitásúak és drágábbak. A különféle megoldású memória-áramkörökkel a 4. fejezetben fogunk részletesen megismerkedni.) A cache kezelése a rendszerprogram egyik kiemelt fontosságú feladata.

A mai PC többszintű cache-rendszerrel működik. Minden mai processzorban beintegrált részleteként megtalálható az L1 cache, ami rendkívül nagy sebességű, de viszonylag kis kapacitású RAM (8 KiByte ... 64 KiByte). A főmemória és az L1 között található egy nagyobb kapacitású, lassúbb, de a főmemóriánál még mindig jóval gyorsabb cache, az L2 (64 KiByte ... 4 MiByte kapacitással). Ez egyes processzorok esetében szintén a mikroprocesszor beintegrált részlete, más megoldásoknál az alaplapon található. Némely processzorok további fokozatot is használnak (L3 cache). Az L3 akár 6 MiByte ... 24 MiByte méretű is lehet, és a processzorba integrálva jelenik meg. Egyes gyártók a többmagos processzoroknál a magok közötti gyorsított információforgalmat biztosító speciális memóriaterületet nevezik az L3-nak.

A processzor többnyire az L1 cache memóriából dolgozik, a főmemóriát csak a teljes futási idő 5 ... 10 %-ában veszi igénybe. Ez a tény igen fontos, mert arra utal, hogy a főmemóriába beépített memóriamoduloknál nincs túl nagy jelentősége a sebesség esetleges növelésének. Ha a teljes főmemóriát 50%-kal gyorsabb RAM modulokkal cserélik ki, a PC tényleges sebessége alig 2,5 ... 5 %-kal nő meg. (A memóriák már nem beforrasztott integrált áramkörök a mikroszámítógép nyomtatott huzalozású paneljén, az alaplapon (Mother Board), a memória IC-kből modulokat alakítanak ki, és az alaplapon lévő csatlakozókba ezeket kell behelyezni.)

A nyolcbites mikroszámítógépben megismert buszrendszer a PC-kben egy összetett hierarchiává terebélyesedett (**3.10. ábra**). Ez az ábra természetesen csak leegyszerűsítve mutatja be a modern buszrendszert. A mikroprocesszor az alaplapon lévő elsődleges buszhoz (Front Side Bus, FSB) csatlakozik.



**3.10. ábra.**

A PC-ben az alaplapon lévő oszcillátor által előállított alapórajellel működik a mikroprocesszor és az „Északi híd” (North Bridge) közötti busz. Az Északi híd egy buszillesztő áramkör, ez kezeli a főmemóriát és a grafikus egységet, valamint ez csatolja a processzorhoz a többi egységet kezelő „Déli híd” (South Bridge) IC-t. A főmemória sebessége egyre nő, ezért egyes processzorokba beintegrálják a memóriavezérlőt is, így a mikroprocesszor közvetlenül tudja kezelni az alaplapon RAM területét.

A processzor órajele az FSB frekvenciájának többszöröse. A teljes hardver szinkronizált működése érdekében a processzor órajelét az alaplapról származó alapórajel többszörözésével állítják elő, a frekvenciatöbbszöröző áramkör a mikroprocesszoron belül van kialakítva. Ha pl. az FSB órajele 283 MHz, és a mikroprocesszorban kilences szorzó van beállítva, a processzor órajel-frekvenciája 2547 MHz, azaz 2,547 GHz.

Érdekes problémát vet fel a processzorok több GHz-es belső órajele. Többnyire megelégedünk arról, hogy a villamos jelek terjedéséhez is időre van szükség, még az egyszerű vezetékekben is. Igaz, ilyen körülmények között a jelek közel fénysebességgel terjednek, de a ns-os periódusidő már olyan rövid, hogy ez alatt a megtett út már csak cm nagyságrendű. És a mai pro-

cesszor chip-ek is ilyen méretekkel rendelkeznek! Többek között ez is magyarázza, hogy a nagyfrekvenciás órajelet a processzoron belül állítják elő.

Mivel már érzékelhető a jelterjedés sebességének korlátozottsága, a processzor belsejében a különféle távolságokban lévő egységekhez vezetett órajel különböző késéseket szenvedhet, és így végül nem valósulhat meg a tényleges szinkron működés. A helyzetet úgy segítene, hogy az órajel forrásához közeli áramkörökhöz több erősítő, késleltető tagon át vezetik az órajelet, a távolabbihoz már kevesebbet alkalmaznak, a legtávolabbi helyekre pedig csak vezetékekkel kötik be az órajelet. A gyártók szerint egy mai processzorban a beintegrált tranzisztorok 25...30 %-a ezzel a feladattal, a belső szinkronizáció megvalósításával foglalkozik.

Egy mai alaplapon is jelentkezik már a sok 100 MHz-es órajelek miatt hasonló probléma, ezért a nyomtatott huzalozást szemügyre véve a buszvezetékeknél láthatjuk, hogy egyes szakaszokba hurkokat, logikailag feleslegesnek látszó mellékutakat helyeztek el, éppen a szinkronizáció biztosítása érdekében.

Esetenként az alaplapi memóriában alkalmazott RAM modulok is az FSB óráfrekvenciájának többszörösét igénylik, ilyen esetben is az alapórajel többszörözésével oldják meg ezt a feladatot.

A PCI, PCIE buszok a PC-k számára kifejlesztett korszerű megoldások, általános illesztési feladatokra. Az ATA, SATA busz elsősorban a merevlemez egység kezelésére használatos. A Déli híd állítja elő az alaplapon szükséges egyéb illesztőfelületeket is, így pl. az USB csatornákat. (Ezekkel az itt említett buszmegoldásokkal további tantárgyakban fognak találkozni.)

Az Északi híd, a Déli híd alkotják azt az IC-készletet (Chip-Set), amely az alaplap legfontosabb egysége. Ez határozza meg, milyen processzort, milyen memória-modult lehet az alaplapba behelyezni, milyen beállítási lehetőségeket lehet használni.

A mai felhasználók sok esetben nem elégszenek meg a Chip-Set alapbeállítása által biztosított működési sebességgel, a PC-t „felpörgetik” (tunningolják). Ennek során növelhetik az FSB óráfrekvenciáját, a mikroprocesszorban alkalmazott szorzót, rövidíthetik a főmemória kezelésére szánt időtartamokat. Ha mindezt mértékletesen teszik, a számítógép gyorsabban fog működni. Egyes beállítások hatására már megszűnik a működőképesség, pl. az alaplapi memóriát nem tudja kezelni a processzor, ekkor vissza kell állítani a lassúbb működési értékeket. Az alaplapok gyártói ezt a tunningolást annyira elfogadják, hogy többnyire a SetUp beállításoknál lehetőséget is adnak a működési elemek gyorsítására.

A gyorsításnak van egy másik következménye is. A Chip-Set, a processzor, a memóriák mind CMOS áramkörök. A CMOS áramkörök ismert sajátossága, hogy fogyasztásuk (azaz a hőtermelésük) a működési frekvenciájukkal arányosan nő. A gyorsabban működő áramkörök jobban melegsznek. A tunningolás gyakori velejárója a számítógép hűtőrendszerének az átalakítása, a processzor hatékonyabb hűtése, sőt, esetenként az Északi híd és a Déli híd, illetve a memória-modulok hűtésének megoldása is szükségessé válhat.

### **3.5. A szinkron és az aszinkron mikroprocesszorok**

A részletesen megismert, nyolcbites mikroprocesszorok, az azokra épülő mikroszámítógépek szinkron (synchronous) áramkörök, az oszcillátor órajele ütemezi minden részlet működését.

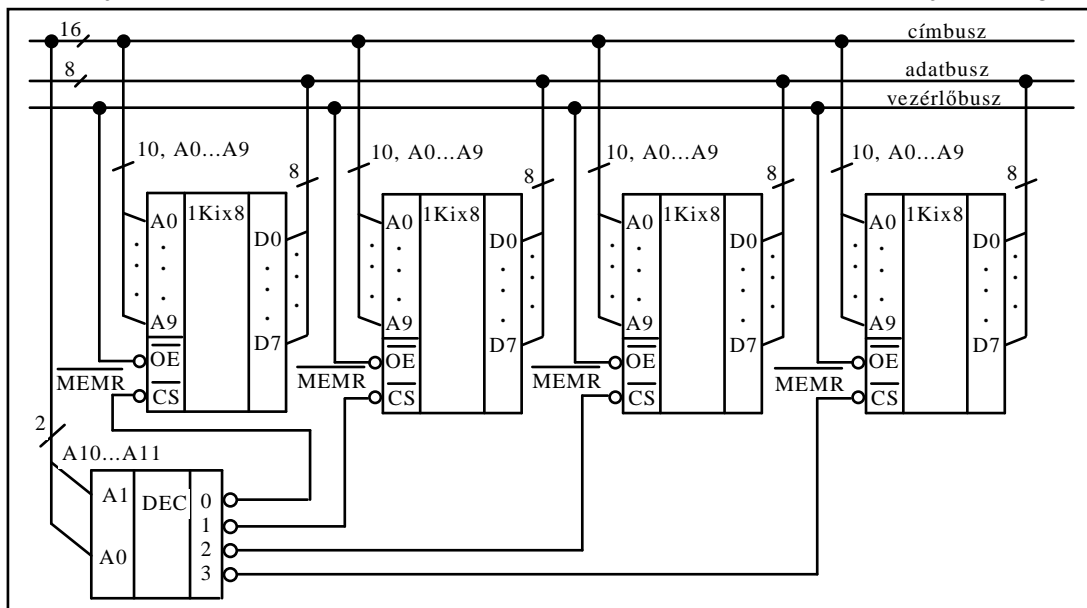
A mai PC is szinkron működésű, az alaplapon lévő oszcillátor határozza meg a működési ütemet, ahol nagyobb óráfrekvencia szükséges, ezt az impulzussorozatot alakítják át nagyobb frekvenciájúvá.

Több gyártó laboratóriumában egy alapvetően eltérő felépítésű processzor típus készül, az aszinkron (asynchronous) mikroprocesszor. Az aszinkron processzor számos előnnyel rendelkezik szinkron társaihoz képest. Az egyes áramköri részek nincsenek ütemezetten működésre készítve, csak az az áramköri részlet igényel tápáramot, ami ténylegesen működik, így a teljes áramkör fogyasztása a megfelelő szinkron megoldás törtrésze lehet!

Javasoljuk, hogy az interneten kövesse figyelemmel ezt az új fejlesztési irányt!

## 4. A memória-áramkörök

A memória-áramkörök olyan félvezető chip-ek, melyek nagy tömegű információ tárolására használhatók fel. Megjelenésükkor PMOS technológiával készültek, később NMOS elemek voltak, végül ezen a területen is uralkodóvá vált a CMOS megoldás. A tárolási egység a rekesz (szó), ennek bitszáma, hosszúsága (szóhossz) a memória IC egyik fő adata. A memória IC címző bemenetei ( $A_i$ ) segítségével lehet a rekeszek egyikét kijelölni. A címdekódolás a memória-áramkörben mindig teljes, minden lehetséges címet használ az áramkör. Ha az áramkörnek n címbemenete van, a rekeszei száma  $2^n$ . Az áramkörön belül a címeket sor- és oszlopcímre bontva használják fel, de a DRAM-ok kivételével ennek a felhasználó számára nincs jelentősége.



4.1. ábra.

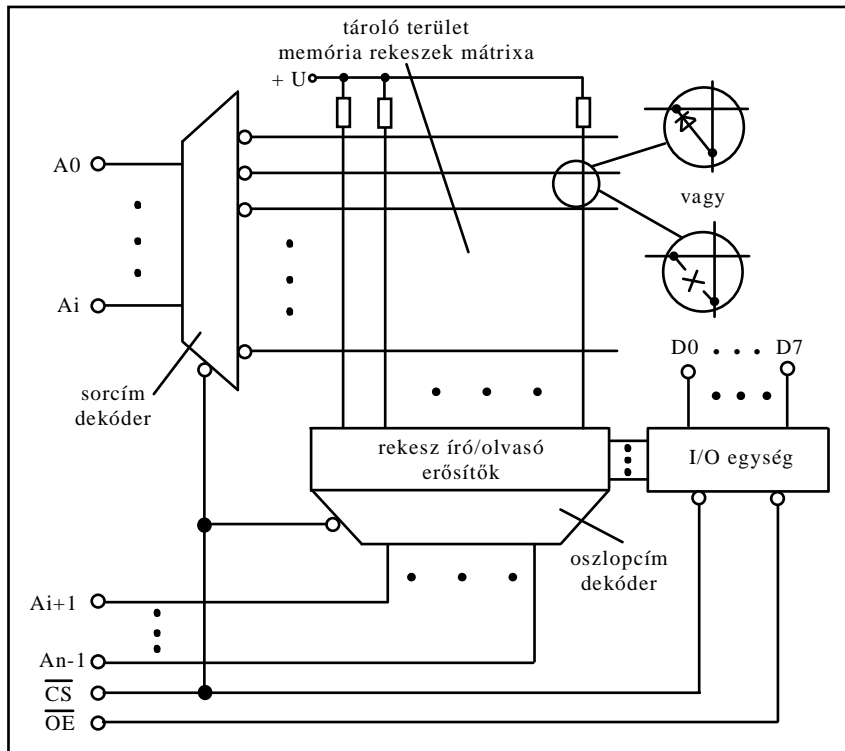
Ha egy memóriának 2048 db 8 bites rekesze van, a kapacitását több módon is megadhatjuk: 2048 x 8 vagy 2 Ki x 8 vagy 2 KiByte vagy 16384 bit vagy 16 Ki bit. A memória IC adatkimenetei mindig három állapotúak, hogy adatbuszra csatlakozhassanak. A kimenet engedélyező jele általában  $\overline{OE}$ . A teljes IC működését engedélyező jel a Chip Select vagy Chip Enable ( $\overline{CS}$  vagy  $\overline{CE}$ ). A  $\overline{CS}$  jeleket használjuk fel akkor is, ha több memória IC-ből épül fel a mikroszámítógép memória-rendszere (4.1. ábra). A memória-áramkörök általában kijelölés nélkül a normális, üzemi tápáram töredékét veszik fel (stand-by állapot).

A mikroszámítógépben az operatív tárat alkotó memória-áramkörök párhuzamos adatkezelésű, címkijelöléssel működő, véletlen hozzáférésű memóriák. A párhuzamos adatkezelés lényege, hogy a cím és az adat bitjei egy időben csatlakoznak a memóriához, azaz annyi címbevezetése van, ahány bites címeket kezel, és annyi adatpontja, ahány bites adatokat tárol egy-egy rekeszben. A címmel történő kijelölés a memória rekeszeinek egyikét azáltal választja ki, hogy a rekesz címét küldjük a memóriához. A véletlen elérés (random access) azt jelenti, hogy egy rekesz elérése ideje független a címtől, független attól, hogy előtte melyik rekeszt kezeltük (ellentétben a soros hozzáférésű megoldásokkal). Az operatív tár céljára írható/olvasható, nem illanó (azaz a tartalmát tápfeszültség nélkül is őrző) memória felelne meg. Sajnos, az IC gyártók nem állítanak elő ilyen megfelelő kapacitással és elfogadható áron. A gyakorlatban a rendszertervezők kétféle memóriával építkeznek:

- kialakítanak egy nem illanó területet, de ezt csak olvasható memóriából (ROM és társai, 4.1. fejezet) tudják megoldani,

– létrehoznak egy írható/olvasható területet RAM memóriákkal (4.2. fejezet), de ezek illanók.

Megoldható a nem illanó, írható/olvasható memória is (4.3. fejezet), de csak kisebb kapacitással és meglehetősen költségesen.



4.2. ábra

A memória IC a címeket fogadó ÉS kapusorozatból (címdekóder) és a tartalmat őrző VAGY kaputerületből áll (memória mátrix, memória terület). Ez egy klasszikus két-szintes logikai hálózat, a PLA-hoz hasonló logikai szerkezet (de a PLA-ban az ÉS mező is, a VAGY mező is programozható, itt az ÉS terület fix kialakítású, hiszen ez a címdekóder). A címdekóder L aktív kimenetei és a VAGY kapuk bemenetei egy vezetékhalózathoz keresztpontjain

köthetők össze egymással (4.2. ábra). Egy címdekóder-kivezetés egy rekeszt, egy szót jelöl ki (ezek a vízszintes vezetékek a szóvezetékek, más névvel a sorvezetékek). Ha ez a szóvezeték a VAGY bemenettel össze van kötve, az 0 értékű tárolt bitet jelent, ha nincs összekötve, az 1-et. A memória-áramkörök abban különböznek egymástól, hogy a tartalmat megvalósító keresztponti kapcsolókat hogyan alakították ki bennük. A keresztpontok összekötését a 4.2. ábrán az átkötésnél látható diódákon át kell megvalósítani, hogy az egyes szavak egymástól függetlenek legyenek. Ha vezeték az átkötő elem, a diódát külön be kell integrálni. Ha az átkötés félvezetőelem, pl. tranzisztor, az a megfelelő dióda-hatást is előállítja. A következőkben áttekintjük az elterjedt megoldásokat, a részletekkel gyakorlatokon, katalógusok alapján fogunk megismerkedni.

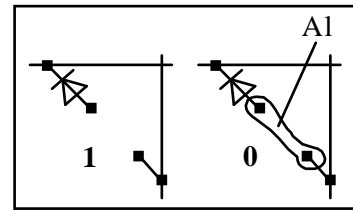
Az integrált memóriák megjelenése előtt már használták a 4.2. ábrán látható kapcsolási elrendezést információátvitelre, a szükséges tartalmat beforrasztott diódákkal állították be a nyomtatott huzalozású dióda-mátrixokban. Tulajdonképpen az IC technológia megjelenése után ezt az ismert kapcsolási elrendezést ültették át integrált kivitelre.

Ha egy memória-áramkör pl. 16 címvezetékkel kezel, 65536 (64Ki) rekeszt tartalmaz. Ha egyetlen címdekóder állítja elő a rekeszkijelölő vezetékeket, az 65536 db 16 bemenetű ÉS kapuból áll. Ha a rekeszeket 256 x 256 elemű mátrixba rendezik, a sorok illetve oszlopok kiválasztásához csak 256 db 8 bemenetű ÉS kapu kell, azaz összesen 512 kapu. Ez egyszerűbb chip kialakítást, kevesebb belső vezeték jelent, ezért kezelik a memória-áramkörök belsejében a címeket sor- és oszlopcímekre bontva. A legtöbb esetben a memória-áramkör bekötésekor a gyártó nem adja meg, melyik biteket használja az áramkör a sorok címzésére, melyikeket az oszlopokhoz, mivel ez a felhasználó szempontjából közömbös.



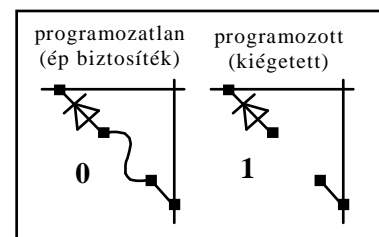
## 4.1. ROM változatok

A ROM (Read Only Memory) csoport minden tagja nem il-  
lanó, csak olvasható memória; a mikroszámítógépben a mikro-  
processzor csak olvassa ezeket. A szűkebben értelmezett ROM  
memória a *maszk-programozott* áramkör. A 4.2. ábrán szereplő  
keresztpontokban fémezéssel (4.3. ábra) teremt kapcsolatot a  
gyártó, a felhasználói igénynek megfelelően (0 értékű bit), vagy  
összekötetlenül hagyja a keresztpontot (1 érték). A maszk-  
programozást csak IC gyártó képes végrehajtani. A ROM in-  
formáció tárolási időtartama korlátlan. A ROM memória rend-  
kívül megbízható tároló, ezért a különleges alkalmazásokban (ürkutatás, repülés) szívesen hasz-  
nálják ma is.

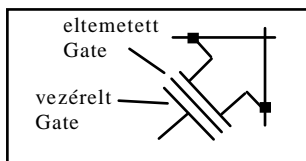


4.3. ábra.

A PROM a felhasználó által programozható, csak olvasha-  
tó memória (Programmable ROM), bipoláris technológiával  
készült. Az üres, tartalom nélküli áramkörben a tartalomőrző  
mátrix minden keresztpontja (4.4. ábra) össze van kötve egy-  
egy apró biztosítékkal (0 értékű bit). Ahol 1 értékre van szük-  
ség, a programozáskor a kis fémhidat ki kell olvasztani egy  
áramimpulzussal. A PROM a beprogramozott tartalmat kor-  
látlan ideig őrzi. Általában bitenként lehet csak programozni,  
mert a biztosíték kiégetése jelentős hőfejlődéssel jár. A  
PROM mára már elavult típus.



4.4. ábra.



4.5. ábra.

Az EPROM (Erasable PROM, eredetileg UV EPROM, azaz  
Ultraviolet Erasable PROM)) törölhető, programozható, csak ol-  
vasható memória. Az információtároló mező keresztpontjaiban  
egy-egy FET tranzisztor csatornája helyezkedik el (4.5. ábra), a  
csatornát kezelő Gate elektródát szigetelő burok veszi körül (elte-  
metett Gate, lebegő Gate). A programozatlan EPROM-ban minden  
eltemetett Gate üres, töltésmentes, a csatorna nem épül fel (a  
Source és a Drain elektródák között a FET közel szakadást mutat),  
minden tárolt bit 1 értékű. Ahol 0 értéket kell tárolni, az IC-re kapcsolt programozó-feszültség  
és a vezérelt Gate segítségével elektronokat juttatnak az eltemetett Gate-re, ami felépíti a csator-  
nát a Source és a Drain pontok között, összeköti a szóvezetékét és a bitvezetékét. A programo-  
zó-feszültség először közel 100 V volt, később 20...22 V lett, a mai EPROM-ok esetében már  
csak 12 V. Az eltemetett Gate kapacitása kb. 0,1 pF, a teljes töltés kb. 30 millió elektron. A  
tranzisztor folyamatosan veszít a töltéséből, 1-2 percenként megszökik egy elektron. A kataló-  
gusok többnyire 10 év információ-megőrzési időt garantálnak.

Az EPROM tokozásán, a chip felett kvarcablakot helyeznek el. Ha ezen át megfelelő intenzi-  
tású ultraibolya fény éri az áramkört, a töltések eláramlása felgyorsul. A katalógusok 253,7 nm-  
es hullámhosszat, 15 WS/cm<sup>2</sup> sugárzási energiát jelölnek meg, ezt UV fénycsővel kb. 2-3 cm  
távolságról lehet elérni. A törléshez így 15 ... 20 perc szükséges. Mivel a teljes felületet világít-  
juk meg, az EPROM egész tartalma törlődik, azaz ismét minden biten 1 lesz a tartalom (ez tehát  
nem szelektív törlés).

Olcsóbb, zárt műanyag tokban is kapható EPROM chip, ezt csak programozni lehet, törölni  
nem, ez az OTP (One Time Programmable) EPROM. Az OTP EPROM tulajdonképpen egy  
MOSFET kivitelű PROM, megjelenésekor tűntek el az eredeti PROM áramkörök.

Sajátos módon sok esetben PROM helyett kvarcablakos EPROM-ot is felhasználtak. Az első  
PC-k alaplapján a BIOS ilyen EPROM-okban helyezkedett el, fekete lappal ragasztották le a  
kvarcablakot, hogy a környezeti fény ne törölje a tartalmat.

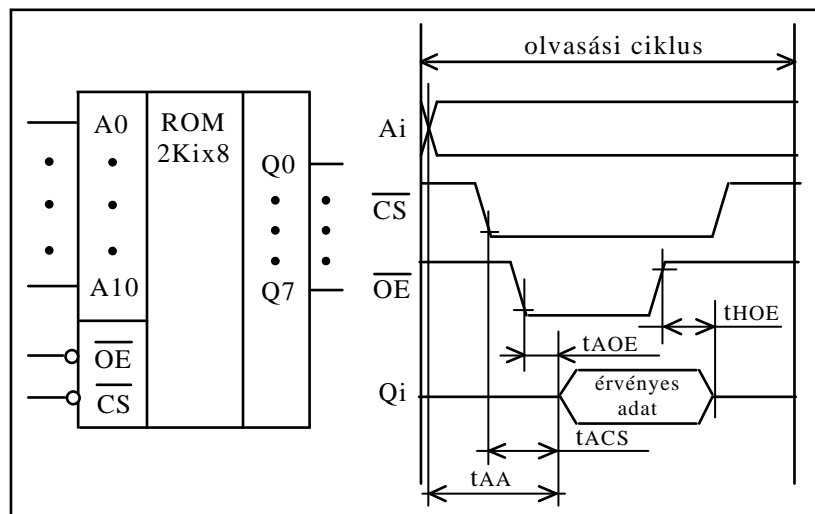
Az *EEPROM* (Electric Erasable PROM) elektromosan törölhető, programozható csak olvasható memória, törlése szelektív. A keresztvezetési pontokban itt is FET tranzisztorok vannak, de ún. szendvics Gate szerkezettel, a csatornát kezelő Gate és a vezérlő Gate közötti félvezető rész vezetőképessége a programozó-feszültséggel vezérelhető. Az EEPROM információ őrzési ideje attól függ, hány törlés/programozás ciklust élt át. Az átlagos EEPROM-ok 10.000 ... 40.000 ciklus után még 10 évig őrzik az információt. A legjobb minőségű EEPROM-ok több millió ciklus után még 40...100 évig garantálják az információ megtartását!

Az EEPROM szavanként törölhető, programozható. Ha a teljes IC-t kell törölni, a szavankénti törlés nagyon időigényes, a gyártók ezért beépítettek egy speciális képességet, a teljes törlés lehetőségét. Ennek időigénye alig haladja meg egyetlen rekesz törlési idejét.

Az első EEPROM-ok az EPROM-hoz hasonlóan külső programozó-feszültséget igényeltek, később a gyártók beépítettek egy feszültség-konvertert, ami a tápfeszültségről működik. A törlés, programozás időzítését is belső áramkörök vezérlik a mai áramkörökben.

A ROM család mai, legelterjedtebb tagja a Flash memória. Ez az EPROM és az EEPROM előnyeit és technológiai megoldásait egyesíti. Az elemi tárolótranzisztorok sorba kapcsolódhatnak (NAND Flash) vagy párhuzamosan (NOR Flash). A NOR Flash általában párhuzamos adatkezelésű, akár programok is futtathatók belőle. A NAND Flash többnyire soros adatelérésű, elsősorban adattárolási célra készítik. A Flash törlése, írása laponként történik, a lap mérete az egyik alapadata az áramkörnek. Elsősorban olyan alkalmazásoknál előnyös, ahol nem rekeszenként, hanem blokkonként kell adatokat tárolni, mozgatni (pl. digitális fényképezőgép).

A Flash áramkörök 100 000 ... 1 000 000 törlés/programozás ciklust viselnek el, 10... 40 évig őrzik az információt. Újszerű alkalmazási formáik a memóriakártyák, a Flash drive (pendrive) egységek. A PC alaplapján ma többnyire Flash tárolja a BIOS-t, amit így interneten át frissíteni, módosítani is lehet.



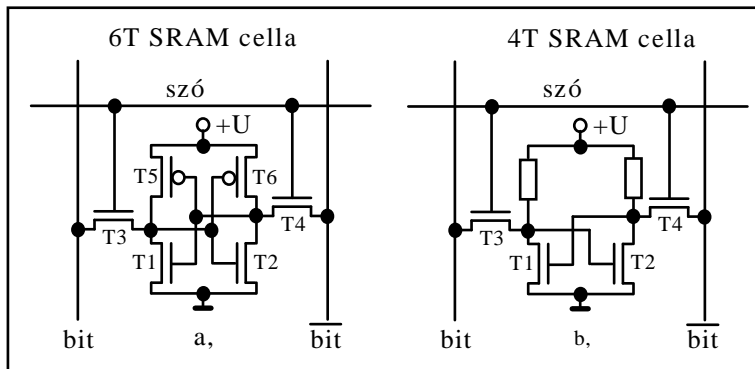
4.6. ábra.

A 4.6. ábrán bemutatjuk a ROM áramkörök tipikus olvasási ciklusát. A  $t_A$  jelű idők az elérési idők, a  $t_H$  adatfenntartási idő. A kijelöletlen ROM áramkör stand-by állapotba kerül. Sok korszerű áramkör már nem vezérlőjelekkel, hanem parancsszavakkal kezelődik.

## 4.2. RAM változatok

Az *írható/olvasható (RAM) memóriák* tápfeszültség hiányában elveszítik a tartalmukat, illanók. Világszerte RAM-nak nevezik ezeket a memória-változatokat, ami nem precíz megjelölés, mivel a RAM a véletlen elérésű memória angol megnevezésének kezdőbetűiből alkotott betűszó, és a megismert ROM változatok, mint tudjuk, szintén mind véletlen elérésűek.

Mindenesetre a három betűs rövidítés (RAM) a gyakorlatban az írható/olvasható memóriákat jelenti. A RAM-áramkörök két nagy csoportja a statikus RAM (SRAM) és a dinamikus RAM (DRAM).



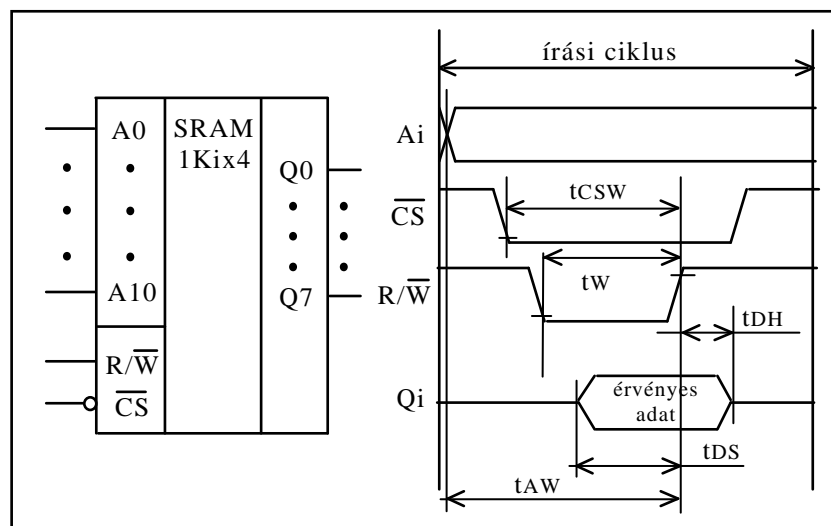
4.7. ábra

A SRAM áramkörben a tároló mátrix minden keresztpontjában egy komplett bistabil multivibrátor van, azaz egy hagyományos egy bites tároló. Egy elemi tárolóegység a mai SRAM áramkörökben (4.7. ábra) általában hat tranzisztorból épül fel (6T cella), de 4 tranzisztorból is kialakítható a tároló (4T cella). A 6T cellával épített SRAM sta-

bil, megbízható működésű, érzéketlen a tápfeszültség ingadozásaira. A 4T megoldás jobban integrálható, de kényesebb áramkör, gondosabb tervezést, kivitelezést igényel, azaz költségsőbb megoldás.

A SRAM olvasási folyamata gyorsítható, ha az olvasási ciklus előtt a bitvezetékeket a tápfeszültség felére feltöltik. Ez az előtöltés (precharge) lépés egy kis időt igényel, de az olvasási ciklust jelentősen lerövidíti, a tényleges bitértékeknek nem kell a teljes tápfeszültségnek megfelelő változást végrehajtani az érzékelő erősítők bemenetén.

A SRAM tápfeszültségre kapcsolásakor a tárolók (flip/flop-ok) véletlenszerűen állnak be 0 vagy 1 értékre, függetlenül az előző tartalomtól, ezért illanó a SRAM áramkör. A korszerű CMOS SRAM stand-by tápárama igen kicsiny, de stand-by állapot mellett, lecsökkentett tápfeszültség esetén is képes a tartalmát megőrizni (általában a tápfeszültség 50%...40% értéke mellett még őrzik a tartalmukat).



4.8. ábra.

A SRAM memória olvasási ciklusa nem különbözik lényegesen a ROM olvasási működésétől, az írási ciklus az újdonság. A 4.8. ábrán két vezérlőjeles ( $\overline{CS}$ ,  $R/\overline{W}$ ) SRAM logikai rajza és írási ciklusa látható.

Az  $R/\overline{W}$  jel H szintje olvasást, L szintje írást rendel el. A mai SRAM áramkörök többnyire három vezérlőjelesek, az olvasást külön  $\overline{OE}$  jel engedélyezi, az írást a  $\overline{WE}$ . A harmadik vezérlőjel a működést engedélyező  $\overline{CS}$ .

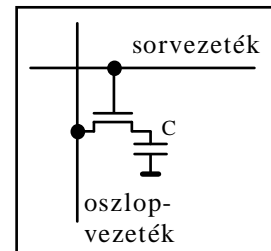
A három vezérlőjeles SRAM könnyebben kezelhető, egyszerűbb beépíteni a mikroszámítógépbe. A két vezérlőjelesnél gondot okozhat, hogy amennyiben a  $\overline{CS}$  aktív, azaz L szintű, a SRAM azonnal működik, hiszen az  $R/\overline{W}$  akár H, akár L szintű, működést ír elő. A három vezérlőjeles áramkör, ha a  $\overline{CS}$  bemenete L szintű, még nem működik, csak ha a további két vezérlőjel ( $\overline{OE}$ ,  $\overline{WE}$ ) valamelyike szintén L szintű.

A SRAM áramkörök korszerű változata a szinkron SRAM (SSRAM), mely a bevezetett órajel éleinél fogadja a bemenőjeleket, illetve hasonló módon küldi ki a kiolvasott értékeket. Így a SRAM és a CPU teljes mértékben szinkronban működhet. További gyorsító megoldásokat is alkalmaznak a SRAM áramköröknél, így az átfedéses (interleaving) és a csomagolt (burst) kezelést, az órajel belső kétszerezését (DDR SSRAM) és négyszerezését (QDR SSRAM), ezekről a technikákról a DRAM áramkörök kapcsán valamivel bővebben szólunk majd.

A PC-ben az indítási adatok (SetUp Information) őrzésére egy akkumulátorral táplált CMOS SRAM áramkört használunk. A cache áramkörök SRAM memóriák, sok esetben a processzor beintegrált részletei.

A PC fő memóriája *dinamikus RAM (DRAM)* áramkörökből épül fel. A DRAM esetén a tárolóterület keresztpontjaiban ismét csak egy-egy FET tranzisztort helyeztek el (4.9. ábra), amely egy parányi kondenzátor feszültségét tudja a bitvezetékre vezetni. Mivel az egyes bitek tárolására a SRAM 6 (esetleg 4) tranzisztora helyett csak egyre van szükség, a DRAM sokkal jobban integrálható, azaz azonos méretű szilíciumfelületen nagyobb kapacitású tárolóterület alakítható ki.

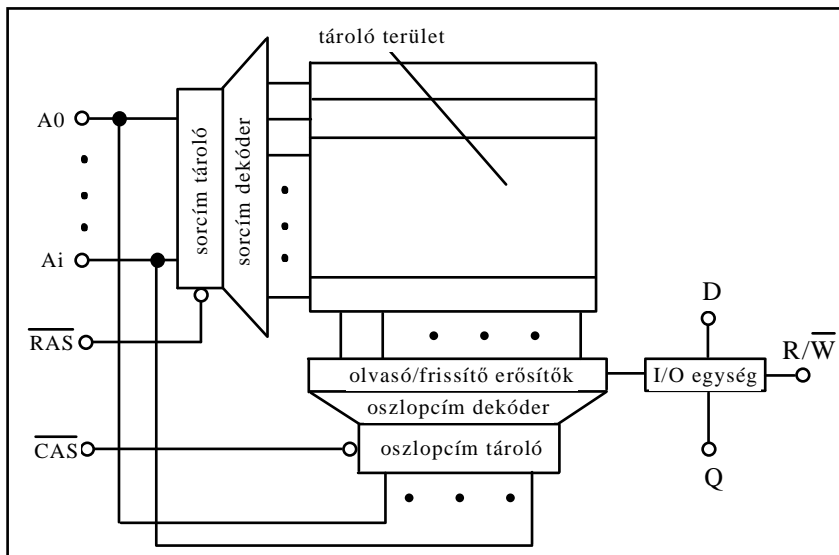
Az oszlopvezeték, más néven bitvezeték az olvasó/frissítő erősítőkhez csatlakozik. Olvasás előtt ezt is a tápfeszültség felére kell beállítani az előtöltési (precharge) fázisban, ezáltal érhető el a teljes olvasási sebesség. A tároló ka-



4.9. ábra

pacitás 30 ... 50 FF, a teljes töltés kb. 1 millió elektron. (Az F decimális prefix, értéke  $10^{-15}$ ). A kis-méretű kondenzátor a feltöltés után előbb-utóbb ki fog sülni.

A folyamatos működés érdekében a DRAM tartalmát rendszeresen, régebbi típusoknál 2 ms-onként, az újabbaknál 32 vagy 64 ms-onként frissíteni kell. (Ez az időérték a frissítési idő, a  $T_F$ ).



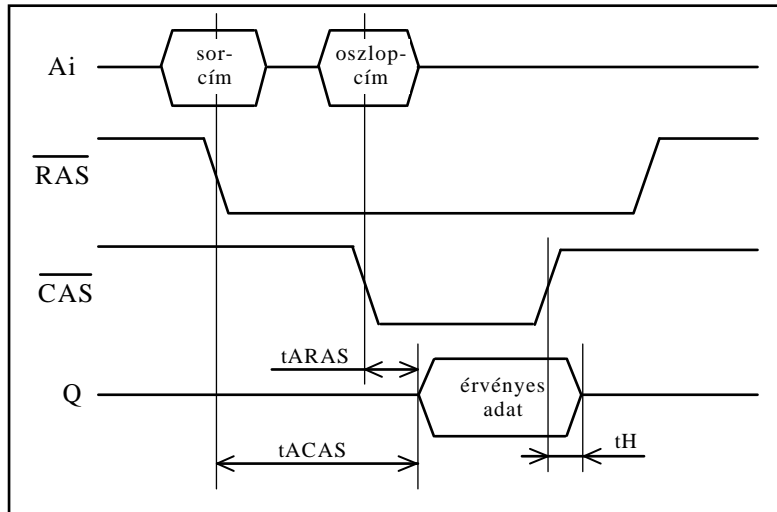
4.10. ábra

Szerencsére ez nem jelenti a DRAM teljes tartalmának ismételt beírását!

A DRAM memóriák címbevezetése általában két lépéses, multiplexelt, egymás után ugyanazokon az IC lábakon át kell bevezetni először a sorcímet (azonosító jele a Row Address Select,  $\overline{RAS}$ ), majd később az oszlopcímet (azonosító jele a Column Address Select,  $\overline{CAS}$ ).

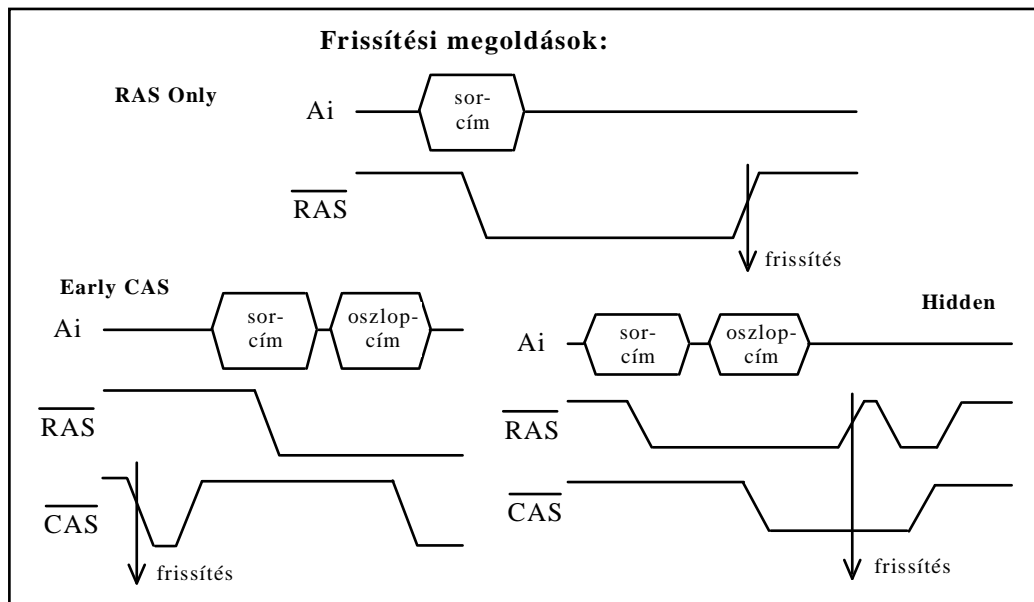
A sorcím és az oszlopcím általában azonos bitszámú. Amikor az  $\overline{\text{RAS}}$  lefutó élekor a sorcímet befogadja a DRAM, az a sorcím tárolóba jut, a sorcím dekóder dekódolja, a tároló mátrixhoz vezeti, onnan a kijelölt sor kilép az olvasó erősítőbe (4.10. ábra). Ha ezután a  $\overline{\text{CAS}}$  jel lefutó élekor az oszlopcím is beolvasódik a tárolójába, az az oszlopcím dekóder útján a sor egy bitjét kiválasztja, ami az I/O egységhez jut.

A működés idődiagramját a 4.11. ábra szemlélteti.



4.11. ábra

A DRAM rekeszei egyetlen bitesek, az I/O egység segítségével valósul meg az egy bit olvasása vagy írása. A  $\overline{\text{CAS}}$  felfutó éle zárja le az olvasási, írási működést; az  $\overline{\text{RAS}}$  felfutó élekor a sor (felfrissülve) visszakerül a helyére a tároló mátrixba. A normál olvasási, írási ciklus tehát az érintett teljes sort frissíti is egyben.



4.12. ábra

Ha a programok futtatására a mikroszámítógép DRAM-ját használjuk, nagy kapacitású memóriaegységekre van szükség. Egy ilyen nagyméretű (2 GiByte, 4 GiByte, vagy még nagyobb) DRAM folyamatos frissítése a CPU, illetve a memóriavezérlő számára igen nagy feladat. Ennek megkönnyítésére a gyártók különböző frissítési eljárásokat dolgoztak ki, ezeket a 4.12. ábrán foglaltuk össze.

A legegyszerűbb frissítés a fentiek alapján a „csak RAS” (RAS Only) megoldás, azaz egy frissítő cím (sorcím) beállítása után egy  $\overline{\text{RAS}}$  impulzus kialakítása a DRAM bemenetén. Ezzel a módszerrel minden DRAM frissíthető, a korai változatok is és a fejlettebbek is.

A korszerű DRAM áramkörökbe beépítették a frissítő számlálót és a frissítő áramkört, a CPU csak a frissítési parancsot adja ki.

A későbbi aszinkron DRAM áramkörök ezért már képesek belső frissítésre, de nincs külön frissítő vezérlőjelük. Az egyik használatos megoldás a „korai CAS” (Early CAS), ahol a  $\overline{\text{RAS}}$  H szintjénél (tehát az olvasási, írási ciklus előtt) a  $\overline{\text{CAS}}$  vezetéken lefutó élt állítanak elő. A ciklus megkezdése előtt H szintre kell állítani a  $\overline{\text{CAS}}$  bemenetet. Ha van rá idő, több  $\overline{\text{CAS}}$  impulzus is beküldhető, így több sor is frissíthető.

Egy másik elterjedt belső frissítési lehetőség a rejtett (Hidden) frissítés. Az olvasási, írási ciklus belsejében lehet frissítést kérni, a  $\overline{\text{CAS}}$  L szintje mellett az  $\overline{\text{RAS}}$  bemeneten kialakított felfutó éllel. Itt is lehet többszörös frissítés is, de a ciklus vége előtt, amikor a  $\overline{\text{CAS}}$  H szintre vált, az  $\overline{\text{RAS}}$  jelnek már L szinten kell állnia.

A kvázistatikus RAM (KSRAM, IRAM) egy frissítést kiváltó vezérlőjellel készül, az  $\overline{\text{RFSH}}$  (refresh) vezérlőbemeneten kialakított lefutó él hatására valósul meg egy sorfrissítés. Ha a  $T_F$  időtartam alatt az összes sor nem frissült fel, a BUSY kimenetén foglaltságot jelez az IC és belül egy komplett burst frissítést valósít meg (ld. alább).

A mai DRAM áramkörök két elterjedt frissítési megoldása az automatikus frissítés (Auto-Refresh) és az önálló frissítés (Self-Refresh). Az Auto-Refresh a memóriavezérlő által kezdeményezett frissítési megoldás, egy frissítő parancs hatására valósul meg (a frissítési címet a DRAM belső számlálója állítja elő). A Self-Refresh üzemmódot is a memóriavezérlő állítja be (és az tudja később megszüntetni is), ebben a beállításban a DRAM belső időzítője méri a frissítési események közötti időt és annak alapján belső frissítéseket hajt végre. Akkor célszerű ezt a megoldást használni, ha az adott DRAM áramkört a CPU átmenetileg nem használja.

A DRAM áramköröket tartalmazó mikroszámítógépek tervezésekor lényeges szempont, hogy a frissítési időpontokat hogyan határozzák meg. Az alkalmazott DRAM frissítési időtartamán, a  $T_F$  időn belül minden sornak frissülnie kell. A frissítés időbeli megoldási lehetőségei:

- csomag (burst) frissítés, a  $T_F$  letelte előtt a CPU program-végrehajtási működése egy ideig megszakad, és maximális sebességgel minden sort felfrissít a DRAM-ban,
- egyenletesen elosztott, a  $T_F$  annyi időszeletre van felosztva, ahány soros a DRAM, minden időszellet legelején egy időzítőáramkörből származó megszakítás hatására a CPU normál működése leáll és egy sor felfrissül,
- eseményhez kötötten, speciális CPU esetén. A Z80 mikroprocesszor pl. frissítő számlálót tartalmaz és minden M1 gépi ciklusban kiküldi a megnövelt frissítő címet és egy frissítő vezérlő jelet.

Az egyre gyorsabb mikroprocesszorok mellett a DRAM ciklusidejének is csökkennie kell, különben nem lehet kihasználni a CPU sebességét. Mindenesetre a mai processzorok jóval gyorsabbak, mint a DRAM-ok, ezért szükséges a már ismertetett cache-rendszer.

A DRAM gyártók különféle áramkör-tervezési fogásokkal tudják lecsökkenteni a DRAM olvasási, írási ciklusidejét. Az is gyorsítja a DRAM kezelést, ha lapokra, bank-ekre osztják, azokon belül egyszerűbb, gyorsabb a címzés.

A gyors lapozásos DRAM (FPM, Fast Page Mode DRAM) a sorcím fogadása után csak oszlop címeket vár, ha a soron belül kell további oszlopokat kezelni. Az EDO (Extended Data Out) DRAM a kiolvasott értéket tárolja, eközben már egy következő ciklus indítható. Továbbfejlesztett változata a BEDO (Burst EDO) DRAM. A „burst” (csomag) jelleg itt azt jelenti, hogy egy cím beírása után a következő 3 címértéket a DRAM belső számlálója állítja elő, s azokat is kiolvasa illetve beírja, külső címzés nélkül (4 elemű burst).

A PC-ben a DRAM működési sebességét többnyire úgy adták meg, hogy felsorolták, az egy-mást követő olvasási, írási ciklusok hány óraciklus időtartam alatt zajlanak le. A felsorolt DRAM áramkörök nem fogadják az órajelet, ezek mind aszinkron memóriák, de az órajellel rendelkező buszra csatlakoznak.

Egy eredeti DRAM esetében a ciklusok időtartama pl. így alakul:

6-6-6-6-6-6- ...

Lapozás esetén az első címzés teljes időtartamú, de azután a soron belüli címek rövidebben elérhetők, pl.

6-4-4-4-4-... .

Természetesen csak a sor végéig lehet így dolgozni, az új sor új sorcímet is igényel, azaz ismét egy 6 óraciklusnak megfelelő időtartamú működés következik.

Az EDO DRAM jellegzetes idősora így alakult:

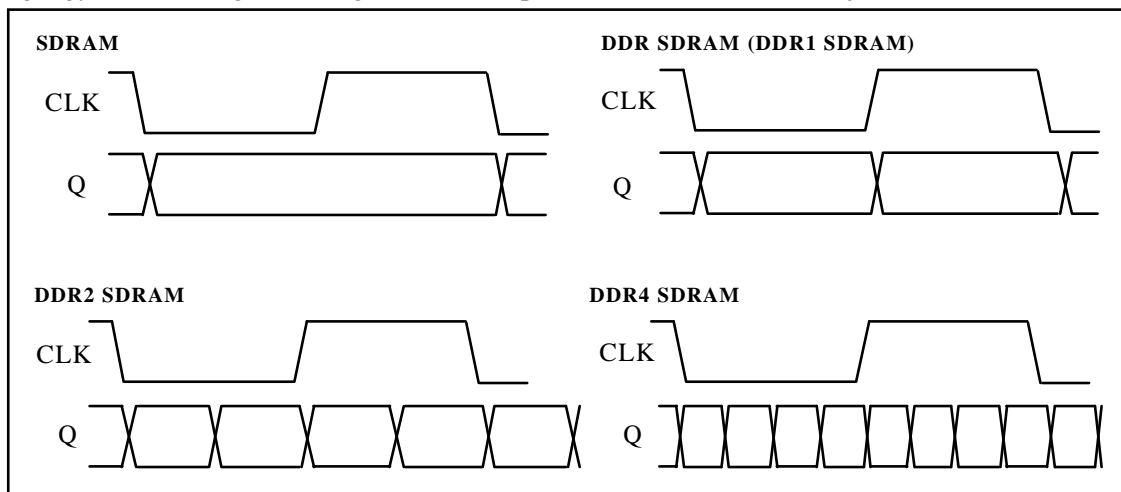
5-2-2-2-2- ...

A BEDO DRAM a címzéskor azonnal nem küldött ki vagy fogadott adatot, mert egy teljes regiszter-rendszere volt, azt kellett feltöltenie előbb. A továbbiakban azonban már nagy sebességgel működött, pl.:

5-1-1-1

jellel.

A további sebességnövekedéshez a DRAM-ot is szinkronná kellett tenni. Az SDRAM, amikor teljes sebességgel működik, minden órajel alatt egy adatot ad ki vagy fogad be. Az SDRAM-ok burst működésre is képesek, a szerelt modulok pedig interleaving működést is végrehajthatnak (felváltva használva a modulokat). Az SDRAM változatok bemenő órajelének és legnagyobb sebességű adatforgalmának a kapcsolatát a **4.13. ábra** vázolja fel.



4.13. ábra

Az eredeti SDRAM a működési mód beállítás után burst módban minden órajelciklus alatt egy adatot ad ki vagy olvas be. A DDR, azaz Double Data Rate (mai jelöléssel DDR1) SDRAM belső áramkörrel kétszeresíti az órajel frekvenciáját, egy külső órajelciklus alatt így két adatot tud kezelni, a DDR2 SDRAM belső frekvencia-négyszeresítéssel már négy adatot olvas vagy ír egy külső óraciklus alatt. A DDR3 SDRAM nyolc adatot kezel a külső órajel egy ciklusa alatt, mivel a frekvenciát nyolcszorozza. Ezek a sebességek természetesen csak a burst működésen belül lépnek fel, hosszabb idő átlagában a memóriák adatsebessége jelentősen kisebb lehet ennél. (Az egyre gyorsabb áramkörök tápfeszültsége egyre kisebb, a DDR-nél 2,5 V; a DDR2-nél 1,8 V; a DDR3-nál 1,5 V. Az Intel már bejelentette a 2012-ben piacra kerülő DDR4 SDRAM-ot, de részleteket még nem árult el róla; a tápfeszültsége 1.2 V lesz, vagy annál is kisebb).

Idő közben egy különleges, szinkron DRAM is megjelent, a Rambus SDRAM (RDRAM), de úgy tűnik, ennek speciális vezérlési igénye az alaplapok tervezőinek nem nyerte el a tetszését, nem vált népszerűvé.

A DDR SDRAM-változatok, az RDRAM már nem úgy készül, hogy minden üzemmódhoz külön vezérlőjele lenne. A hagyományos DRAM csatlakozópontokon kívül azért vannak újak, CKE (órjel engedélyezés), DQM (I/O puffer maszkoló bemenet), BAi (Bank-választó bitek). A CKE (Clock Enable) H szintje mellett a beérkező órjel működteti a DRAM-ot. Az L szint mellett önálló frissítés (Self Refresh) történik, előtöltés végezhető, de nincs adatcsere a DRAM és környezete között. A DQM (DQ Mask) hatására olvasási folyamat esetén két órajellel meghosszabbodik az olvasási ciklus. Írási működés esetén a DQM letiltja a beírást.

Legtöbbször az I/O puffer külön csatlakozópontokon kapja a tápfeszültségét, hogy az áramkör belső működéséből eredő villamos zajok ne zavarják sem az IC működését, sem a hozzá csatlakozó többi áramkört. De mindez nem elegendő az összes működési mód kijelölésére. Ezért a belső üzemmódregiszter (Mode Register) tartalmát a DRAM-ot kezelő vezérlőáramkör egy különleges programozási lépésben tudja betölteni. Ebben a lépésben a vezérlőjeleken kívül a címbemenetek is üzemmód-információt közvetítenek a DRAM-hoz, azaz a címbemenetekről töltődik fel az üzemmódregiszter.

Az SDRAM áramkörök működési időit már a tényleges óraciklusok számával írhatjuk le, hiszen ezek órajelekkel működnek. A címző és vezérlő szakaszban a bemenőjelek segítségével meghatározzuk a sorcímet, az oszlopcímet, a működési módot (az SDRAM-ok esetében pl. beállítható a burst hosszúsága, ami 2, 4, 8 elemű vagy egy teljes lap lehet, más esetben a futó burst működés vezérlőjelekkel megszakítható, kijelölhető a frissítési mód stb.) Ennek a bevezető szakasznak a legfontosabb időértékeit (a látens időket, Latency Times) kötőjeles számsorként szokás megadni, pl. így:

2,5-5-4-1,5 1T.

A kötőjellel elválasztott számok jelentése: CL-RCD-RP-RAS memória-időzítés. Ezek óraciklus-számok, a különféle történések végrehajtásához szükséges időtartamot óraciklus-számmal fejezik ki. CL (CAS Latency): olvasáskor az adatoknak a tárolómátrixban történő kijelöléstől ennyi időre van szükségük, míg a kimeneti pontokon megjelennek. RCD (RAS to CAS delay): a sor és az oszlop kiválasztása között szükséges időtartam. RP (RAS Precharge): új sor kezelése előtt a bitvezetékek előtöltéséhez szükséges idő, azaz két sor közötti váltás közötti minimális idő. RAS (RAS Active Time): az RAS jel aktív L szintjének, az RAS impulzusnak az időtartama.

Az időzítési számsorban a kötőjellel összekötött értékeket egy további elem követi, ami 1T vagy 2T lehet. Esetenként az 1T helyett CR1-et, a 2T helyett CR2-t írnak, a CR: Command Rate. Amikor a memóriavezérlő először fordul a memóriához, engedélyezni kell az IC működését, ki kell választani egy bank-et, majd ebben a sorcímmel, oszlopcímmel elvégezhető a címzés. Az IC kiválasztásához szükséges óraciklusok számát jelöli az 1T/2T, ha egyetlen óraciklus elég, akkor írják elő az 1T-t, ha hosszabb idő szükséges, akkor a két óraciklust biztosító 2T-t.

Mindezeket az értékeket a PC-nél általában a felhasználó is tudja állítani, a BIOS-ból származó alapértékektől így el lehet térni, fel lehet gyorsítani a memóriát (ez a tuningolás egyik alaplehetősége, a további fogások az alapórjel frekvenciájának és a processzor szorzójának a növelése). A látens idők eltelte után már órajelenként 1, 2, 4 vagy 8 adatot kezel az SDRAM (burst üzemmódban), a belső órjel-kezelésnek megfelelően.

A PC alaplajába már régen nem közvetlenül a DRAM IC-ket helyezzük be, hanem szerelt, sok csatlakozólábbal rendelkező modulokat. A chipset-ben az Északi Híd vezérli az alaplaj főmemóriáját, bár néhány újabb mikroprocesszor közvetlenül csatlakozik a DRAM modulokhoz, miután beintegrálták a memóriavezérlő részleteket is a processzorba.

Az SDRAM esetében az IC megjelölésekor is és a modul megjelölésekor is az órjel frekvenciáját szokták használni (MHz-ben megadva), pl. SDRAM-66 (66 MHz-es SDRAM), a modul PC-66. PC-100 modul SDRAM-100, a PC-133 modul SDRAM-133 áramkörök alkotják.

A DDR SDRAM áramköröknél sajátosak a jelölések. A memória-áramköröknél a felszorozott frekvenciát szokás megjelölni, így a DDR-200 kívülről 100 MHz-es órajelet kap, a belső, felszorozott értéke a 200 MHz. A modulok esetében viszont a MByte/s-ban kifejezett legnagyobb adatátviteli sebességet jelöli a típus kódjele!



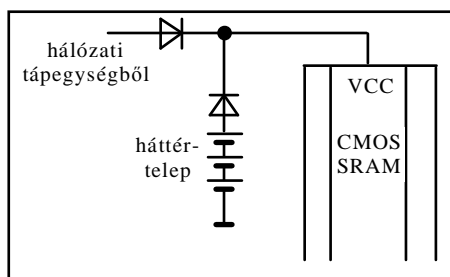
A PC2-6400 modul pl. DDR2 SDRAM áramkörökből épül fel, mégpedig DDR2-800 jellegűekből. Mivel ez az áramkör a külső órajelet négyszerezi, a beérkező külső frekvencia 200 MHz. A modul belső adatútja 64 bites, azaz 8 Byte-os, így másodpercenként  $8 \times 800 \times 10^6$  Byte-ot mozgat, tehát 6400 Mbyte-ot. Így adódik a modulra jellemző 6400-as számérték.

A modulok csatlakozószáma, mérete is változik a beépített áramköröktől függően, ezért egy alaplapba csak a leírásában meghatározott modulokat lehet behelyezni.

### 4.3. Az írható/olvasható, nem illanó félvezetős memória megvalósítása

A mikroszámítógépekben egy-egy kisebb kapacitású, *nem illanó, írható/olvasható memóriát* ki szoktak alakítani. Az *NVRAM* (Non-Volatile RAM, nem illanó RAM) létező IC típus, de sajnos, kis kapacitású és drága elem. Ebben azonos kapacitású SRAM és EEPROM van, a tápfeszültség eltűnésekor a SRAM tartalmát átveszi az EEPROM, megőrzi, a visszatérő tápfeszültség hatására a tartalom ismét visszamasolódik a SRAM rekeszeibe.

A memóriagyártó cégek laboratóriumaiban készülnek írható/olvasható, nem illanó félvezetős memóriák, több gyártó is kísérletezik olyan IC-kkel, melyekben a biteket ferromágneses cellát tárolják (FRAM, FERAM, MRAM). Mivel mágneses alapú az információ őrzése, az elemek nem igényelnek tápfeszültséget az információ megtartásához. A mágneses adattároló részlet egy tranzisztoron belül is elhelyezhető, így ez a memória hasonlóan jól integrálható, mint a DRAM változatok. Sajnos, egyelőre ezek az áramkörök nem kerülnek be a tömeggyártásba.



4.14. ábra

A gyakorlatban elterjedt megoldás inkább az, hogy egy kis stand-by tápáramú CMOS SRAM memória IC-nek nagy kapacitású (F-os) kondenzátorból, elemről vagy akkumulátorról *háttértáplálást* biztosítanak, amikor a tápegység kikapcsolódik (4.14. ábra). A korszerű, alacsony tápáramú CMOS memóriák éveken át nagy megbízhatósággal őrzik így az adatokat, pl. a PC SetUp információit. Egyes gyártók (pl. a Dallas) a CMOS SRAM tokozásán belül helyeznek el egy lítiumelemet, így lényegében egy nemillanó, írható/olvasható integrált áramkört alkotva (10 évre szól az adatőrzési garancia).

### 4.4. Soros hozzáférésű félvezetős memóriák

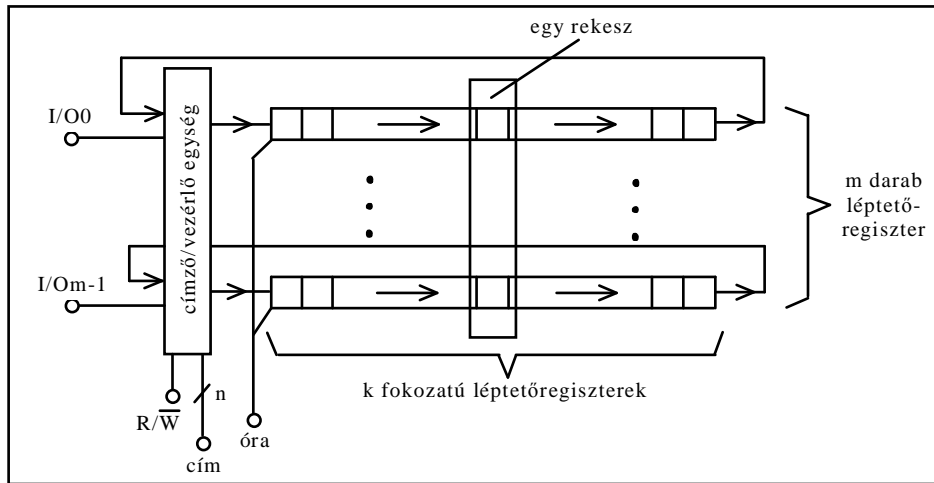
A szakirodalom röviden csak soros memóriákat emleget (Serial Memory) többféle áramköri megoldás esetén is, ami esetleg félvezető lehet. Két fontos csoportot sorolnak ide:

- a véletlen elérésű, de soros adatkezelésű memóriákat,
- a soros hozzáférésű memóriákat.

Az eddig megismert áramkörök mind *véletlen hozzáférésűek* voltak, ebben a fejezetben viszont a *soros hozzáférésű* félvezetős memóriákat mutatjuk be. A működési elvet a léptetőregiszterekből kialakított (4.15 ábra) memórián lehet megfigyelni. Ahány bites a léptetőregiszter, annyi szavas a tároló, a szóhosszúságot pedig a léptetőregiszterek darabszáma határozza meg. Az ábrán szereplő soros hozzáférésű memória  $k$  darab  $m$  bites szót tárol.

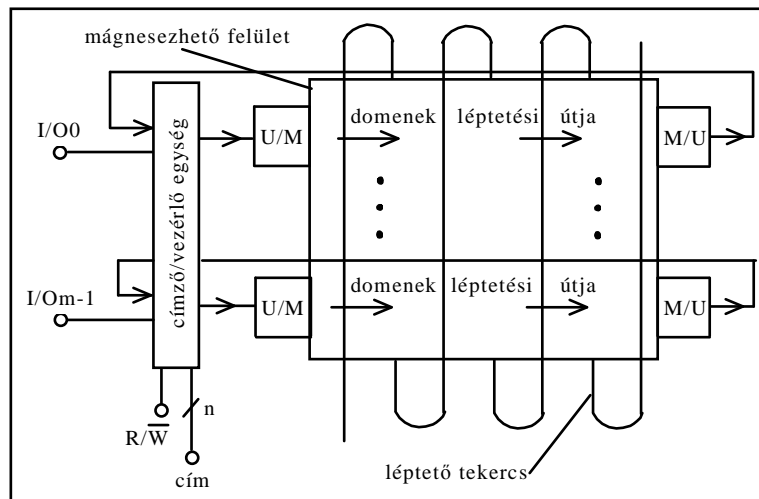
A léptetőregisztereket folyamatosan léptetik, az adatok körben haladnak. A regiszter végére érve az adatok a visszacsatoló vezetékeken a címező/vezérlő egységbe jutnak, ebben a helyzetben lehet az adatszót írni vagy olvasni, majd ismét belépnek a regiszterekbe. A soros hozzáférés jól megfigyelhető. Ha olyan rekeszt címezünk, ami éppen belép a címező/vezérlő egységbe, azonnal rendelkezésre áll; de ha éppen most lépett ki a regiszterekből, meg kell várni, míg végighalad a teljes regiszteren.

A léptetőregiszteres megoldást memória chip-ként nem gyártják, de vannak más félvezető megoldások, melyek tulajdonképpen ezt az elvet alkalmazzák.



4.15. ábra.

A *mágneses buborékmemória* (Magnetic Bubble Memory, *MBM*) esetében a chip felületén mágneses réteget hoznak létre és mágneses csomagocskák (domain, buborék) halad végig a chip körül kialakított vezeték rendszerben alkalmazott léptető áram mágneses terének hatására. A chip bal szélén (4.16. ábra) egy magnetofon írófejhez hasonló egység hozza létre a logikai állapotnak megfelelő mágneses csomagot, amit a jobb szélén a fordított átalakító ismét logikai jellé konvertál vissza. Egyébként a kialakítás és a működés hasonló a léptetőregiszteres áramkörhöz. Ha 262144 lépés alatt jut át a bal szélről a domain a jobb szélre az MBM 256 Ki szót tárol, ha 8 mágneses ösvény van a felületen, 256 KiByte a kapacitása. Mivel a mágneses domain léptető jelek, tápfeszültség nélkül megmarad a felületen, az MBM nem illanó memória.



4.16. ábra.

Eleinte (az 1970-es években) igen nagy jövőt jósoltak az MBM megoldásnak, a mágneslemezes tárolók helyettesítőjét látták benne. MiByte-os kapacitású áramköröket készített a Texas Instruments és több más cég. Nagy előnye, hogy nincs mozgó alkatrésze, igen hosszú életű. Ha folyamatosan ennek a típusnak a fejlesztésére költöttek volna a gyártók, lehet, hogy megtörténik a váltás, de a mágneslemezes, a merevlemezes egységek fejlesztése vonzóbb volt. Mára a buborékmemóriák gyártása teljesen megszűnt.

Egy másik soros hozzáférésű félvezető memória a *CCD* (Charge Coupled Device), a *töltéscsatolt áramkör*. Ennél a léptető regiszterek helyett a chip felületén villamos töltéscsomagokat léptető pályákat alakítanak ki, a töltéscsomagokat a félvezetőt körülvevő vezetékben lévő

feszültség villamos tere lépteti végig a felületen. Az áramkör felépítése hasonló a 4.12. ábrán látotthoz, csak az U/M átalakító helyén U/Q; az M/U konverter helyén Q/U elem szerepel benne. A villamos töltést léptető jel tartja egy csomagban, ha az megszűnik, a töltés eloszlik a felületen. Eszerint a CCD memória léptetés nélkül nem őrzi meg az adatokat, illanó memória.

Bár gyártanak CCD memória IC-eket, a CCD technika igazi alkalmazási területe az optoelektronika. Ha egy CCD chip felületén, minden egyes léptetési pozíción egy-egy fotodiódát vagy fototranzisztort helyeznek el, a rávetülő kép fény-árnyék viszonyainak megfelelő töltéskép alakul ki a chip felületén. Ezt a töltésképet CCD jelleggel ki lehet léptetni az áramkörből – így működik a CCD kamera is. Az első fényérzékelők csak egy sor fotodiódát tartalmaztak, a képet soronként dolgozták fel, de hamarosan megjelentek a teljes képet egyszerre befogadó szenzorok is. A továbbiakban megoldották a színszűrést is, és így a képszenzorok már színes videojelet tudtak előállítani.

#### 4.5. Nem címkijelöléssel működő memóriák

Az eddig megismert memória-áramkörök mindegyikében a rekesz tartalmának beírásához vagy kiolvasásához a rekesz címét kellett az áramkörhöz vezetni (címkijelöléssel működő memóriák).

A 3.2. fejezetben megismertük a veremtárral, melynél a memóriába íráskor illetve kiolvasások száma határozta meg, melyik rekesz kezelődik egy soron következő hozzáforduláskor. A mikroprocesszorok a veremtárat hagyományos *címkijelöléssel működő memóriával* valósítják meg (hiszen az operatív tárból alakul ki a verem is), a verem jellegű működést a veremtár mutató (SP) automatikus dekrementálásával, inkrementálásával érik el. Ugyanakkor LIFO (Last In First Out, az utolsóként beírt elsőként kiolvasható), FIFO (First In First Out, az elsőként beírt elsőként kiolvasható) jellegű, hozzáfordulási sorrend alapján címzhető memória IC-eket gyártanak is. A mikroprocesszoros mikroszámítógépben tehát tulajdonképpen egy LIFO memória emulációja történik meg.

Sajátos feladatok megoldására alkalmasak a *tartalom-címezhető memóriák* (Content Addressable Memory, CAM), melyeket *asszociatív tárnak* is neveznek. Itt a rekeszek beírása, azaz a memória feltöltése hagyományosan történik. Kiolvasáskor azonban egy tartalmat adunk meg, az áramkör válasza pedig arra utal, megtalálható-e benne a keresett adat, s ha igen, melyik rekesze tartalmazza azt. A mikroprocesszorokba integrált cache memóriák asszociatív jelleggel is kezelhetők, így képes a processzor gyorsan megállapítani, hogy a megfelelő programegység a cache-ben van-e.

#### 4.6. A soros adatkezelésű memóriák

A mikroprocesszoros mikroszámítógépek, a PC-k a gyors működés érdekében párhuzamos adatkezelést használnak, természetesen az ezekbe beépített memóriák is párhuzamos adatkezelésűek. Az adatok, a címek párhuzamosan futnak a buszokon, s a memória-áramkörök is így fogadják ezeket, és így küldik ki a kiolvasott adatokat is.

Elsősorban a mikrovezérlők környezetében használják a soros adatkezelésű memóriákat, mivel ott a sebesség nem olyan elsődleges paraméter, továbbá a mikrovezérlők esetében törekedni kell arra, hogy minél kevesebb csatlakozópontot kössön le egy csatlakozó áramkör. A mikrovezérlőnek nincs külső buszrendszere, egy külső egységet ezért perifériaként, párhuzamos portokon át tud csak kezelni.

A soros adatkezelésű áramköröknél egyetlen vezetéken át, bitsorosan érkeznek az adatok, a címek, illetve a kiolvasott adatok is így távoznak. Ezek az IC-k nem rendelkeznek a megismert vezérlőjelekkel, a működési módokat parancsszavakkal lehet kiváltani, amiket szintén bitsorosan lehet beküldeni az IC parancsszó-tárolójába.

A mai soros adatkezelésű áramkörök mind órajelelrel működő, szinkros soros adatátviteli elemek, és a rendszerekben Master vagy Slave szerepűek (a memóriák minden esetben Slave jellegűek). A Master a kapcsolat irányítója, egyben az órajel előállítója, a Slave alárendelten működik, az órajeleket csak fogadni képes.

A soros adatkezelésnek mára több szabványos megoldása is kialakult, a Microwire, SPI, I2C illesztővel rendelkező memóriákról a Mikroelektronikai gyakorlatok V. jegyzetben olvashat.

## 5. A mikroprocesszorok felépítése, működése

A következőkben megismerkedünk néhány mikroprocesszorral. Első példánk, a 8 bites mikroprocesszorok bemutatják a mikroprocesszor-technika alapvető megoldásait, szokásos működési módjait, így pl. a vezérlőjelek státuszó alapján való előállítását (8080), a multiplexelt adat- és címbusz-t (8085), a sokoldalú megszakítási rendszert (Z80), az I/O működéseket nem ismerő processzor-kialakítást (6800). Teljes részletességgel az i8080A mikroprocesszort dolgozzuk majd fel, mert egyrészt egy áttekinthető, világos működésű áramkör, másrészt ennek megértése alapot ad bármelyik mikroprocesszor, mikrovezérlő tanulmányozásához.

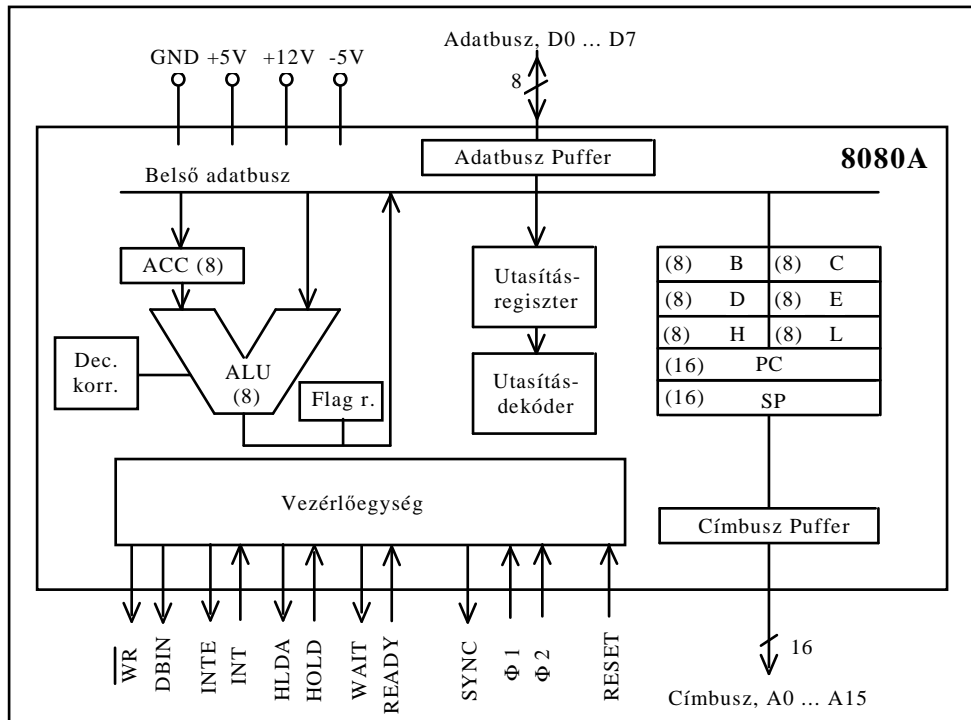
A fejezet második részében a nagyobb szóhosszúságú Intel mikroprocesszorok megjelenéséről olvashatunk.

### 5.1. Az Intel 8080A mikroprocesszor

Az Intel cég a Fairchild-ből vált ki, LSI áramkörök gyártására szakosodott. Főleg memória IC-eket fejlesztettek, de készüléképítést is vállaltak, pl. asztali számológépeket is gyártottak. A mikroprocesszorok történetét 1971-től számítjuk, amikor az Intel egy sorozat asztali számológépre fogadott el egy japán megrendelést. A közel hasonló készülékeket egyenként kellett volna megtervezni, eltérő nyomtatott huzalozású paneleket igényeltek volna, egyedi szerelést. A cég egyik mérnöke, Marcial Edward (Ted) Hoff kezdeményezésére a különféle huzalozott logikai áramkörök helyett a tárolt programot végrehajtó integrált számítógép CPU-t, a 4004-et fejlesztették ki. Rövidesen kiderült, milyen sokoldalú eszközt készítettek, visszavásárolták az áramkör jogait és megindult a mikroprocesszorok fejlesztése. A 16 kivezetéses tokozású 4004, 2250 beintegrált tranzisztor funkciójával 4 bites mikroprocesszor volt, PMOS technológiával készült. 1972-re jelent meg a fejlettebb 4040, majd a 8008, ez már nyolc bites  $\mu P$ , 3300 tranzisztor funkcióval.

1974-ben jelenti be az Intel a 8080-at. Rövidesen elkészül a módosított 8080A változat, s ez minden eddiginél nagyobb sikert aratott az alkalmazóknál. 5500 beintegrált tranzisztor foglalt magába, 40 kivezetésű DIL tokozású áramkör, NMOS technológiával. Sokan a 8080A-t tartják az első igazi mikroprocesszornak. Mivel gyakorlatilag csak a 8080A került forgalomba, sokszor csak 8080-at említenek, de a végleges 8080A típusra gondolnak. A mikroprocesszor utasításkészlete 74 utasításból áll, az utasításokat részletesen gyakorlatokon elemezzük.

A 8080A belső felépítését az <b>5.1. ábra</b> mutatja be. A csatlakozópontok szerepe a következő:	
GND, +5V, +12V, -5V	tápfeszültség csatlakozások,
$\phi 1$ , $\phi 2$	kétfázisú, 12V-os órajel,
SYNC	kimeneti vezérlő jel, a gépi ciklusok elején H szintű,
RESET	alaphelyzet beállító jel ( $PC = 0$ , $IR = 0$ , $INTE = 0$ ),
READY	a gépi cikluson belül, a T2 és T3 óraciklusok közé TW várakozási ciklusokat kérő bemeneti jel,
WAIT	a TW ciklust jelző kimenő jel,
HOLD	a buszokhoz csatlakozó $\mu P$ pontok lebegtetését kérő jel,
HLDA	a HOLD állapotot jelző kimenő jel,
INTE	H szintje azt jelzi, hogy a programban megszakítást engedélyező utasítás szerepelt,
INT	megszakítást kérő bemenő jel (maszkolható),
DBIN	az adatbuszról a mikroprocesszor adatot olvas be, ha H szintű,
WR	az adatbuszra a $\mu P$ adatot küld ki, ha L szintű,
A0 ... A15	a 16 bites címbusz csatlakozó pontjai, 64 Ki címterjedelem,
D0 ... D7	a 8 bites adatbusz csatlakozó pontjai.



5.1. ábra.

A mikroprocesszor belső elemei:

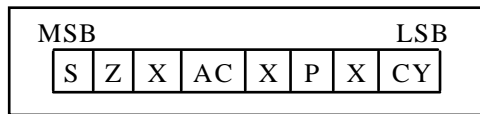
- buszpufferek,
- vezérlőegység,
- utasításregiszter (IR),
- utasításdekóder,
- 8 bites akkumulátor (A vagy ACC),
- 8 bites aritmetikai-logikai egység (ALU), előjel nélküli egész kettes számrendszerbeli számokon végez aritmetikai és logikai műveleteket,
- jelzőbit (Flag) regiszter (F), esetenként az A-val párban szerepel (AF),
- decimális korrekciós áramkör (BCD számok összeadása után a keletkezett hamis eredményt helyes BCD összegre igazítja ki),
- általános célú 8 bites regiszterek (B, C, D, E, H, L), esetenként 16 bites párokban is használhatók (BC, DE, HL),
- 16 bites veremtár mutató (SP),
- 16 bites programszámláló (PC).

RESET után a 8080A a 0h címről, M1 gépi ciklussal indul, a megszakítások tiltottak. A memóriacímzés (akár veremről, akár operatív tárról van szó) a címbuszon valósul meg, 16 bites címmel. Az I/O utasítások (a 8080A-nak egy IN N és egy OUT N utasítása van) 8 bites I/O címet használnak (n), a 16 bites címbuszon ez az I/O gépi ciklusban duplikálva jelenik meg (azaz ekkor a címbusz alsó és felső felének azonos a tartalma, az n érték).

A jelzőbit (Flag) regiszter bitjeit az 5.2. ábra mutatja be, az egyes bitek jelentése a következő:

- |    |   |
|----|---|
| S  | (Sign), előjel, valójában az eredmény D7 bitje,   |
| Z  | (Zero), zérus jelző bit; ha az eredmény 0, Z = 1 lesz,                                      |
| AC | (Auxiliary Carry) másképpen HC (Half Carry); az alsó négy biten keletkezett átvitelt jelzi, |
| P  | (Parity), paritás bit; ha az eredményben az 1-ek száma páros, P = 1,                        |

CY másképpen C (Carry), nyolcbites átvitel, illetve kivonásnál kölcsön.

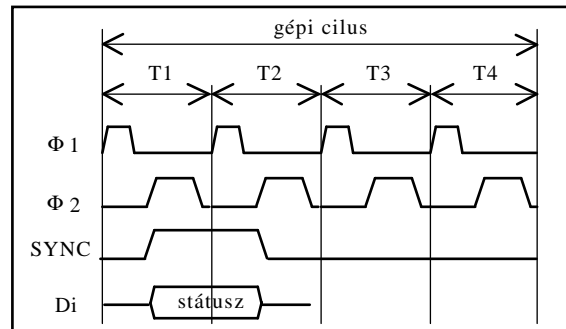


5.2. ábra

A 8080A kétfázisú, + 12 V névleges H szintű órajelet igényel, a két fázis H szintjei nem lapolhatják át egymást. Az órajel frekvenciája 0,5 MHz ... 2 MHz lehet. Az alsó határ magyarázata az, hogy a  $\mu P$  belső tároló elemei DRAM jellegűek és az órajel biztosítja a frissítésüket (ez általában így van az NMOS processzoroknál, a CMOS jellegűeknél többnyire az órajelnek nincs alsó frekvenciahatára).

Az óraciklusban mindkét fázis H szintje megjelenik. 3 ... 5 óraciklus alkot egy gépi ciklust (5.3. ábra). Az adatmozgás a T3 óraciklusban történik meg, s csak akkor tartalmaz a gépi ciklus további óraciklusokat, ha a pillanatnyi belső működések ezt igénylik.

A gépi ciklus elején, az első  $\Phi 2$  felfutó élénél jelenik meg a SYNC jel H szintje, vele egy időben az adatbuszon kilép a gépi ciklus jellegéről informáló státusz szó. A 8080A-nak tízféle státusz szava van, tehát tíz különféle gépi ciklusát lehet megkülönböztetni. A mikroszámítógépekben általában nem szükséges a tízfé-



5.3. ábra

s t á t u s z   s z ó								a gépi ciklus megnevezése	vezérlő jel
D0	D1	D2	D3	D4	D5	D6	D7		
INTA	$\overline{WO}$	STACK	HLTA	OUT	M1	INP	MEMR		
0	1	0	0	0	1	0	1	M1, Fetch, utasítás beolvasás (opkód)	$\overline{MEMR}$
0	1	0	0	0	0	0	1	általános memória olvasás	$\overline{MEMR}$
0	0	0	0	0	0	0	0	általános memória írás	$\overline{MEMW}$
0	1	1	0	0	0	0	1	verem memória olvasás	$\overline{MEMR}$
0	0	1	0	0	0	0	0	verem memória írás	$\overline{MEMW}$
0	1	0	0	0	0	1	0	bemenet olvasás (az IN utasítás miatt)	$\overline{I/OR}$
0	0	0	0	1	0	0	0	kimenet írás (az OUT utasítás miatt)	$\overline{I/OW}$
1	1	0	0	0	1	0	0	elfogadott megszakítás	$\overline{INTA}$
0	1	0	1	0	0	0	1	HALT utasítás miatti leállás	
1	1	0	1	0	1	0	0	megszakítás elfogadás HALT állapotban	$\overline{INTA}$

5.4. ábra

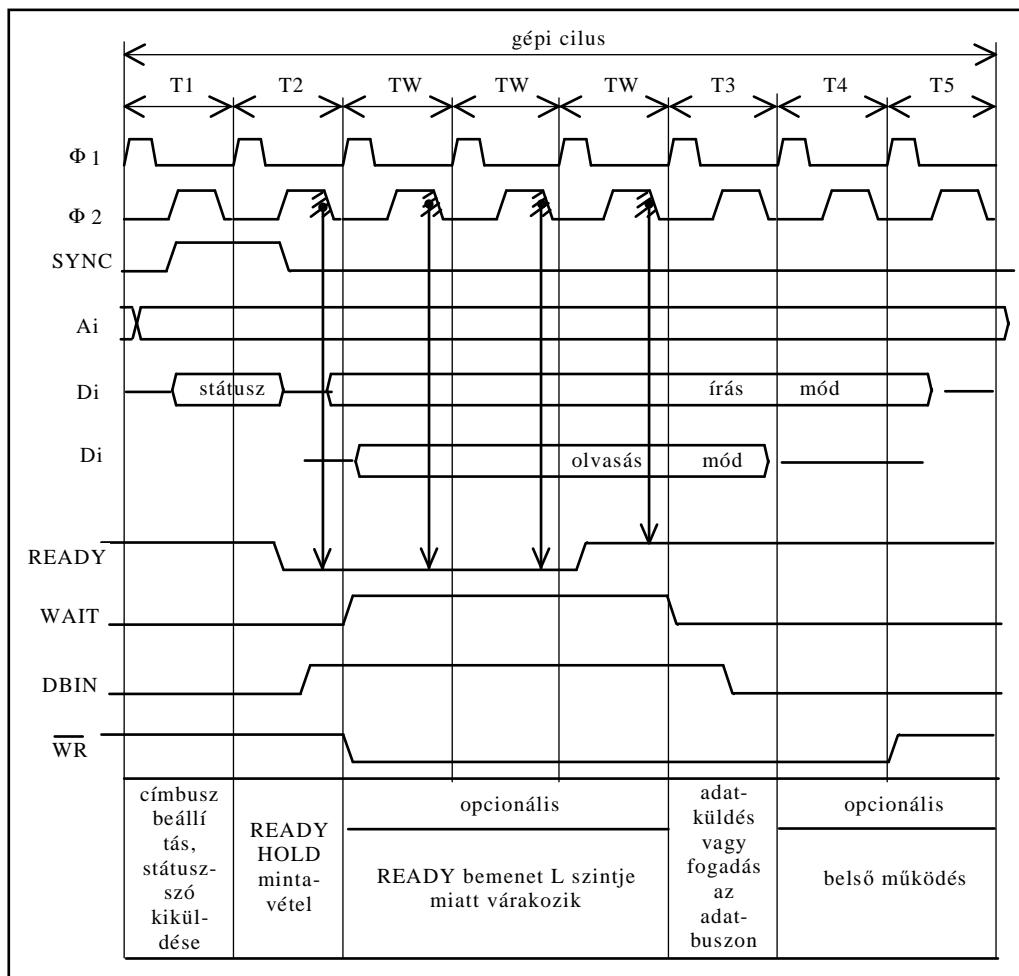
le gépi ciklus megkülönböztetése. A 8080A-ra épülő rendszerekben öt vezérlőjelet szokás használni:  $\overline{MEMR}$ ,  $\overline{MEMW}$ ,  $\overline{I/OR}$ ,  $\overline{I/OW}$ ,  $\overline{INTA}$ . Az 5.4. ábra bemutatja a tízféle gépi ciklust leíró státusz szót, a gépi ciklus neveket, a státusz szóból szokásosan előállított vezérlőjeleket.

	Írás	Olvasás
I/O	$\overline{\text{I/O}\overline{\text{W}}}$	$\overline{\text{I/O}\overline{\text{R}}}$
Memória	$\overline{\text{MEM}\overline{\text{W}}}$	$\overline{\text{MEM}\overline{\text{R}}}$

5.5. ábra.

A négy alpműködéshez egy-egy vezérlőjel tartozik csak, amelyik felveszi L aktív szintjét (5.5. ábra). A vezérlőjeleket a státusz szóból külső hardvernek kell előállítania, ehhez a SYNC jel felhasználásával az adatbuszról le kell olvasni a státusz szót, tárolni kell a gépi ciklus idejére, végül át kell kódolni.

Az 5.6. ábrán a 8080A egy gépi ciklusa látható, az ábrán szerepel az írási és az olvasási működés is. Az első három óraciklus feladatai minden gépi ciklusban azonosak, a 4. és 5. óraciklus (ha van), az éppen végrehajtott utasítástól függ (opcionális).



5.6. ábra

- T1: A címbuszra kilép a gépi ciklusban érvényes cím, megjelenik a SYNC jel és vele egy időben a státusz szó.
- T2: a READY és a HOLD bemenetekről mintát vesz a  $\mu\text{P}$  (az ábrán a READY = L, így a T2 után nem T3 következik, hanem TW).
- TW: várakozási ciklus, T2 után egy vagy több TW írható elő READY = L értékével; minden TW végén ismét ellenőrzi a  $\mu\text{P}$  a READY bemenetét, és csak akkor folytatja a működést a T3 ciklussal, ha a READY = H. A TW alatt a WAIT kimenet H szintű.



T3: az adatkiküldés illetve az adatbefogadás óraciklusa.

T4, T5: utasítástól függően létezhetnek, belső működés zajlik alattuk.

Az **5.6. ábrán** nyomon követhető az írási művelet is és az olvasási is, természetesen a valóságban egy gépi ciklusban csak az egyik mehet végbe.

Ha T2-ben a HOLD = H, az adatmozgást még elvégzi a  $\mu P$ , de rögtön utána lebegteti a csatlakozópontjait. Ha ilyen esetben a gépi ciklus 4 vagy 5 óraciklusos és az utolsó végén a HOLD bemenet 0-ra vált, az utasítás végrehajtás normál ütemben folyhat (nem lassul le), közben egy-egy időszakra a DMA lehetőség is megjelenik (cikluslopás).

A megszakítás-kérő bemenetet (INT) minden utasításciklus utolsó gépi ciklusában, az utolsó óraciklus végén vizsgálja meg a mikroprocesszor. A megszakítás elfogadását utasítással lehet engedélyezni (EI) vagy tiltani (DI). Ha nincs megszakítás-kérés vagy tiltott, a következő gépi ciklus M1 lesz. Ha azonban megszakítást fogad el a  $\mu P$ , egy módosított M1 fut le, egy INTA gépi ciklus (megszakítás-elfogadás). A működés lényege megegyezik az M1-gyel, az eltérések a következők:

- a státusz szó M1 helyett INTA (két bit felcserélődik benne),
- a PC nem inkrementálódik,
- a megszakítások elfogadása letiltódik.

Ahhoz, hogy a 8080A a megszakítást kezelő szubrutinra ugorhasson, a külső hardvernek kell a megszakítás elfogadásakor egy opkódot a  $\mu P$  adatbusz-pontjaira vezetni. Ez célszerűen az RST N, ami egy Byte-os szubrutinhívó utasítás (a CALL három Byte-os utasítás, abból csak az opkódot lehetne beküldeni). Az RST N három bites címrészt tartalmaz  $N = 0 \dots 7$ ), ez a három bit határozza meg a szubrutin kezdőcímét:

0000 0000 00nn n000 ,

ahol nnn az N bináris alakja. Az INTA ciklusban azért kell letiltani a PC inkrementálódását, hogy a hardver által beküldött RST a helyes folytatási címet menthesse a verembe.

A 8080A utasításait gyakorlatokon fogjuk elemezni, összesen 74 különféle utasítása van a  $\mu P$ -nek, ezek összesen 246 féle opkódot használnak.

A 8080A kimenetei az egységnyi TTL terhelést nem viselik el, bemeneti minimális H szintje sem TTL jellegű (3,3 V). A környezetet ezért hagyományos TTL elemekből nem lehet kialakítani, speciális áramkörökkel kell a 8080A-t körülvenni. Ugyancsak speciális áramkört igényel a +12 V-os névleges feszültségű, kétfázisú órajel. Mindezekkel a körülményekkel a 6. fejezetben fogunk még találkozni.

Végezetül emlékeztetünk arra, hogy a **3.9. ábrán** már bemutattuk a 8080A IN N utasításának időbeli lefolyását.

## 5.2. Az Intel 8085A mikroprocesszor

A vezérlőjelek kombinált előállítására és a multiplexelt adat- és címbusz alkalmazására szemléletes példa a 8085A  $\mu P$ . 1976-ban adja ki az Intel azzal a szándékkal, hogy a mikroprocesszor alkalmazókat úgy mentesítse a 8080A hardver nehézségei alól (három tápfeszültség, külső kétfázisú 12V-os óra, külső vezérlőjel előállítás, szegényes megszakítási lehetőség stb.), hogy a 8080A-ra fejlesztett szoftverek változtatás nélkül továbbra is felhasználhatók legyenek, azaz szoftver szempontból a 8085A felülről kompatibilis legyen a 8080A-val. A 8085A a 8080A-nak mind a 74 utasítását bit-azonos módon ismeri és kezeli, mindössze két új utasítása van. 6200 tranzisztort tartalmaz, 40 kivezetéses DIL tokozással gyártják.

A 8085A belső óragenerátorral készült, melyre csak egy frekvencia-meghatározó elemet (általában kvarckristályt) kell csatlakoztatni kívülről (a  $\mu P$  órajele a kvarcfrekvencia fele lesz). Egyetlen, +5 V-os tápfeszültséget igényel, beépített megszakítás-kezelővel és soros adatcsatlakozási lehetőséggel is rendelkezik. A címbusz alsó nyolc bitje ugyanazokon a csatlakozóponton jelenik meg, mint az adatbusz (multiplexelt busz), tehát ezen a nyolc ponton (AD0 ... AD7) felváltva vagy címbitek, vagy adatbitek mozognak.

A vezérlőjeleket maga a mikroprocesszor állítja elő. A négy fő gépi ciklust kezelő vezérlőjelek változtak, a működést három jel kezeli:

		Írás	Olvasás
I/O	Memória	$IO/\overline{M} = H$ $\overline{WR} = L$	$IO/\overline{M} = H$ $\overline{RD} = L$
		$IO/\overline{M} = L$ $\overline{WR} = L$	$IO/\overline{M} = L$ $\overline{RD} = L$

5.7. ábra.

$\overline{RD}$ ,  $\overline{WR}$ ,  $IO/\overline{M}$ .

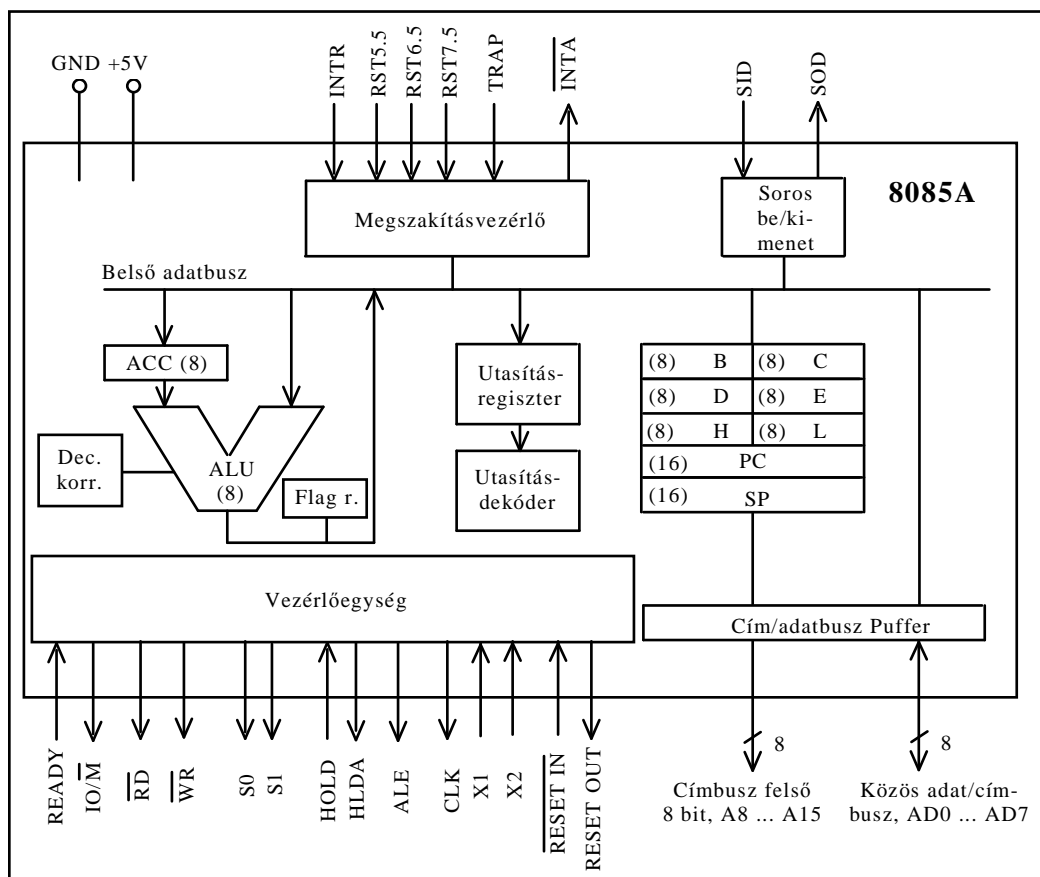
A négy gépi ciklus (memória olvasás, memória írás, I/O olvasás, I/O írás) kijelölését a négy vezérlőjel az **5.7. ábra** szerint oldja meg. Ugyanakkor a 8085A státuszbitet is előállít (S1, S0), bár ezeket többnyire a felhasználók nem vették igénybe. A státuszbit a következő információkat szolgáltatják (az  $IO/\overline{M}$  jel függvényében):

$IO/\overline{M}$	S1	S0	állapot
0	0	1	memória írás
0	1	0	memória olvasás
1	0	1	I/O írás
1	1	0	I/O olvasás
0	1	1	M1, opkód beolvasás
1	1	1	INTA, megszakítás-elfogadás
lebeg	0	0	HALT
lebeg	X	X	HOLD, RESET

A 8085A belső felépítését az **5.8. ábra** mutatja be. Jól látható, hogy sok részletben megegyezik a 8080A belső kialakításával, a bevezetésben megismert változások is megfigyelhetők. A csatlakozópontok közül a READY, a HOLD és a HLDA ugyanolyan szerepet tölt be, mint a 8080A esetében, a további pontok szerepe a következő:

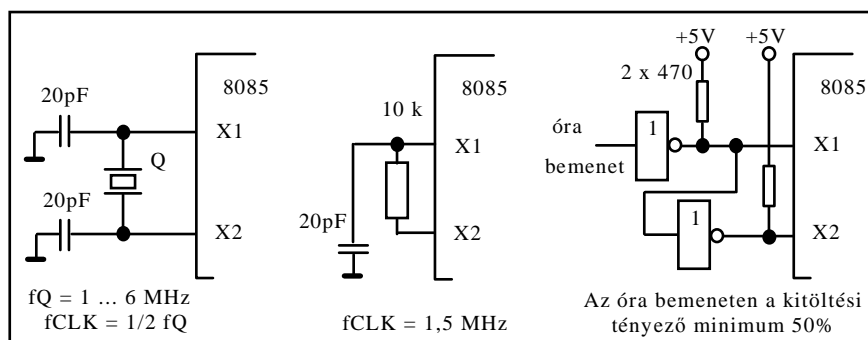
+5V, GND	a tápfeszültség csatlakozásai,
X1, X2	a frekvencia-meghatározó elem csatlakozópontjai, az órajel frekvenciája az itt meghatározott érték fele,
CLK	órajel kimenet,
$\overline{RESETIN}$	RC taggal kiegészíthető bemenet, bekapcsolási és kézi RESET kialakítására,
RESOUT	a RESET folyamat közben H szintű kimenő RESET jel,
ALE	H szintje jelzi, hogy az AD0 ... AD7 pontokon cím áll,
S1, S0	a státuszbitet,
$\overline{RD}$	L szintje jelzi az olvasási jellegű gépi ciklust,
$\overline{WR}$	L szintje jelzi az írási jellegű gépi ciklust,
$IO/\overline{M}$	H szintje jelzi az I/O működést, L szintje a memória aktivizálást,
TRAP	nem maszkolható megszakítás-kérő bemenet,

INTR	maszkolható megszakítás-kérő bemenet (megfelel a 8080A INT bemenetétének),
RST 7.5, 6.5, 5.5	maszkolható megszakítás-kérő bemenetek,
INTA	megszakítás-elfogadás gépi ciklus jelzése,
SID	soros bemenőadat,
SOD	soros kimenőadat.



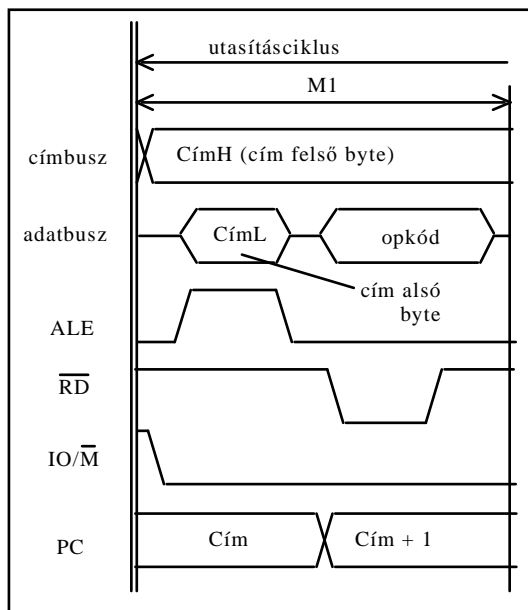
5.8. ábra.

A RIM utasítás az akkumulátor legfelső bitjére helyezi a SID ponton lévő értéket; a SIM utasítás pedig az akkumulátor legfelső bitjét küldi ki a SOD ponton (ez a 8085 két új utasítása, a soros pontokon kívül ezek kezelik az új megszakításokat is).



5.9. ábra

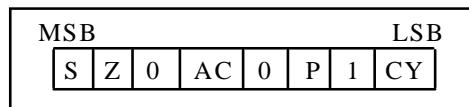
A frekvencia-meghatározó elemek alkalmazási lehetőségeire az 5.9. ábrán látunk példákat, az 5.10. ábra pedig egy M1 gépi ciklus időbeli lefolyását szemlélteti. A 8085A-nál az órajel frekvenciája 3 ... 5 MHz közötti lehet (a rezgőkvarc tehát kétszer ilyen frekvenciájú).



5.10. ábra.

felfutó él alakult ki, ettől kezdve a megszakítási igény folyamatosan fennáll, mindaddig, míg szoftverrel nem töröljük a kérést. Erről akkor sem szabad megfeledkezni, ha a 81085A elfogadja a megszakítás-kérést és lefut a megfelelő szubrutin. Ebben a szubrutinban el kell helyezni az utasítást, amely törli az RST7.5 igényt jelző belső tárolót.

A 8085A jelzőbitjei (Flag-ek) azonosak a 8080A Flag készletével, az F regiszter az **5.11. ábrán** található meg.



5.11. ábra

A RESET folyamat hatása is azonos a 8080A-nál megismerttel, a PC nullázódik, a megszakítás tiltódik és az utasításregiszter kiürül, így RESET után a processzor a 0 címről, M1 gépi ciklussal indul. Az IN N, OUT N utasítások végrehajtása sem különbözik a 8080A működésétől, az I/O cím itt is megduplázva jelenik meg a címbusz az I/O gépi ciklusban.

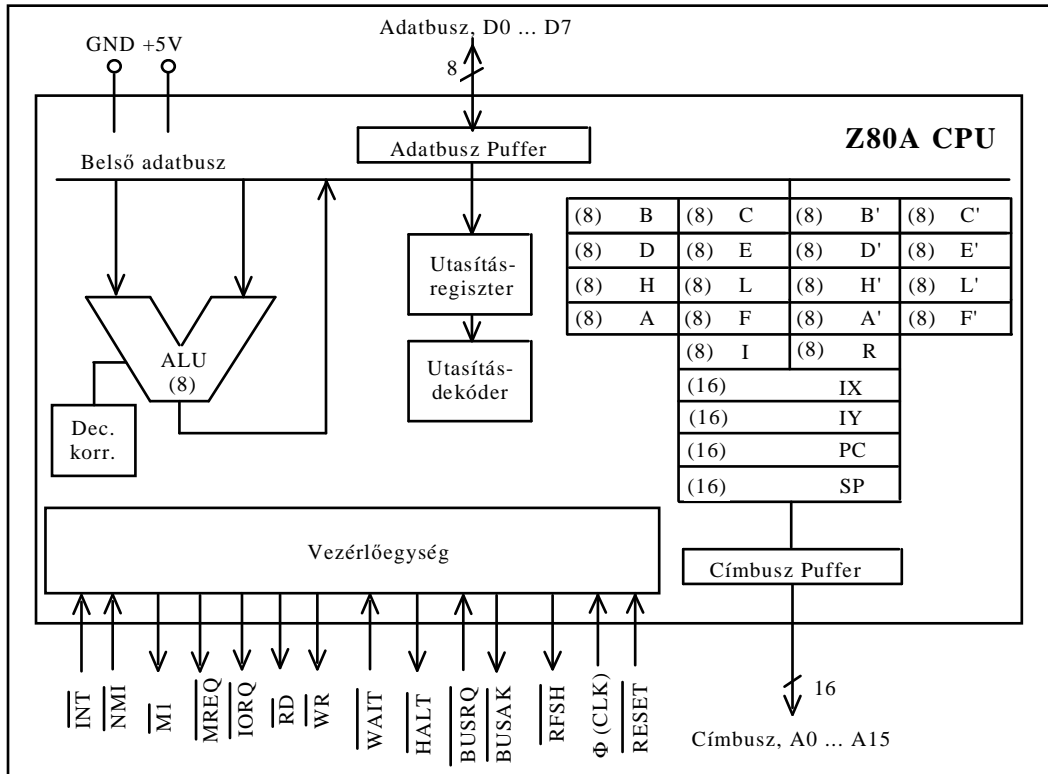
A 8085A megjelenése után is népszerű maradt a 8080A, többnyire azzal tervezték meg a kor nyolcbites kisszámítógépeit. A 8085A megítélése változatos volt, sokan úgy minősítik, hogy az Intel a mikrovezérlők fejlesztésének próbájaként alkotta meg ezt az áramkört. Valóban, sok olyan vonása van, amit később a mikroprocesszoroknál nem találunk meg, de a mikrovezérlőknél igen (pl. belső soros egységek, azaz soros kimenet, soros bemenet, belső óragenerátor, beintegrált megszakítás-kezelő egység nagyszámú megszakítás-kérő bemenettel).

A 8085A később CMOS kivitelben is megjelent.

### 5.3. A Zilog Z80-CPU

A sokoldalú megszakítási rendszerre szép példa a Z80-CPU (amit legtöbbször röviden csak Z80-ként említenek), mely 8000 tranzisztor funkciót tartalmaz és szintén 40 kivezetésű DIL tokozású. 1976-ban készítette el a nemrégén létrejött Zilog cég (melyet egy, az Intel-ből kivált fiatal mérnökcsoport hozott létre). A 8080A-val felülről szoftver kompatibilis, annak utasításait bit-azonosan fogadja és értelmezi; de az utasításkészlet 120 eleműre bővült, az új utasítások jelentős részénél az opkód 2 vagy 3 Byte méretű lett.

Azért volt szükség a több-Byte-os opkódokra, mert a 8080A 246 opkódot használ a 256 lehetséges nyolcbitesből, tehát csak tíz kombináció maradt ki, ezek közül egyet-egyét a Zilog arra használta fel, hogy jelezze: az utasítás opkódja két Byte-os, illetve három Byte-os. Két illetve három Byte-on már igen nagy számú új utasítást lehetett képezni.



5.12. ábra.

A Z80 belső felépítése sok részletben megegyezik a 8080A belső felépítésével, de több új elemet is tartalmaz, az **5.12. ábrán** lehet megvizsgálni. A csatlakozópontok szerepe a következő:

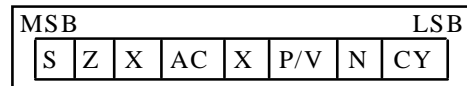
+5V, GND	a tápfeszültség csatlakozó pontjai,
A0 ... A15	16 bites címbusz,
D0 ... D7	8 bites adatbusz,
CLK (φ)	+5V-os órajel, 50 %-os kitöltési tényezővel,
RESET	alaphelyzetbe állító jel, RC taggal bekapcsolási RESET, nyomógombbal kézi RESET hozható itt létre,
RFSH	az M1 gépi ciklus második felében aktív vezérlő jel, L szintje jelzi, hogy a címbuszon a DRAM frissítő cím van jelen,
BUSRQ	HOLD kérés, L aktív szinttel,
BUSAK	HOLD állapot nyugtázó jele, L aktív szinttel,
WAIT	a T2 és T3 közé TW várakozási ciklusokat kérhet ez a bemenő jel, L aktív szinttel,
HALT	a HALT utasítás utáni állapotot jelzi, L szinttel,
MREQ	L szinttel memóriakezelő gépi ciklust jelez,
IORQ	L szinttel I/O-kezelő gépi ciklust jelez,
RD	L szinttel olvasás jellegű gépi ciklust jelez,
WR	L szinttel írás jellegű gépi ciklust jelez,

$\overline{MI}$	önállóan opkód beolvasó gépi ciklust jelez, az $\overline{IORQ}$ -val együtt a jelentése: INTA,
$\overline{NMI}$	nem maszkolható megszakítás-kérő jel, L aktív szinttel,
$\overline{INT}$	maszkolható megszakítás-kérő jel, L aktív szinttel.

A belső elemek között a következő újdonságokat találjuk (a 8080A-val összevetve):

- második regisztercsoport (B', C' stb.), az első és a második közül egyszerre csak az egyik aktív, az aktívat két utasítással lehet kijelölni illetve váltani (pl. szubrutinban)
- megszakítás-vektor felső Byte regiszter (I),
- frissítő regiszter (R),
- index regiszterek (IX, IY).

A Z80 F regisztere kicsit eltér a 8080A-nál látott jelzőbit regisztertől (5.13. ábra). A HC, Z, S megegyezik a 8080A azonos nevű bitjeivel; az N a kivonás műveletet jelzi, mert ebben a  $\mu P$ -ben a decimális korrekció összeadás és kivonás után is működik és helyreállítja a BCD eredményt; a C csak átvitel, a túlsordulást a P/V bit jelzi, ami logikai műveletkor paritásbit szerepet lát el, aritmetikai műveletnél túlsordulás/kölcsön jelző bit. Ez az oka annak, hogy bár a 8080A utasításait bit-azonosan kezeli a Z80, a Flag bitek esetleges eltérő



5.13. ábra

kezelése miatt a 8080A-ra írt szoftvereket át kell vizsgálni, mielőtt azokat Z80-on futtatnánk.

A négy alapvető gépi ciklus vezérlőjeleit az 5.14. ábra mutatja be.

A Z80 egy tápfeszültségű, külső órajellel működő  $\mu P$ , az órajel frekvenciája 2,5 MHz ... 8 MHz közötti lehet, a CMOS változatoké 10 MHz. A CMOS verziókban a belső regisztereket SRAM jelleggel alakították ki, így az órajel frekvenciának nincs alsó határa.

A Z80 ismeri a 8080A IN N/OUT N utasításait, hasonló módon hajtja végre, de az I/O gépi ciklusokban nem duplázva küldi ki az I/O címet,

	Írás	Olvasás
I/O	$\overline{IORQ} = L$ $\overline{WR} = L$	$\overline{IORQ} = L$ $\overline{RD} = L$
Memória	$\overline{MREQ} = L$ $\overline{WR} = L$	$\overline{MREQ} = L$ $\overline{RD} = L$

5.14. ábra.

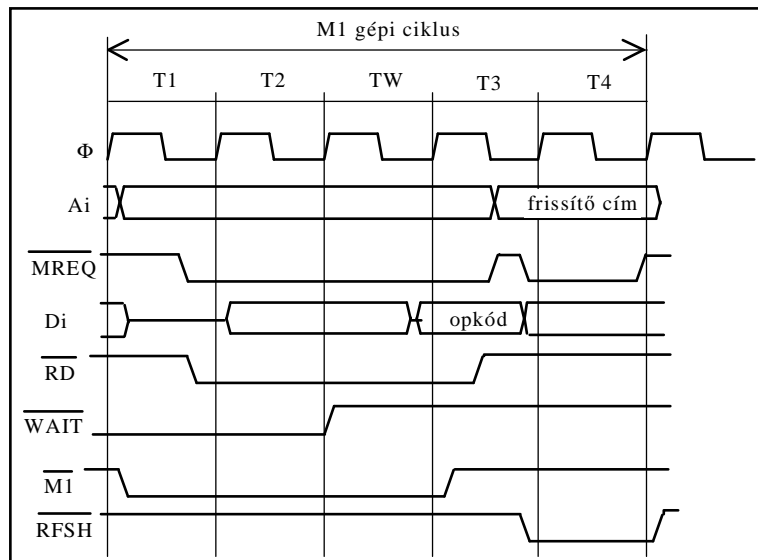
csak a címbusz alsó 8 bitjén, a felső 8 biten az A tartalma lép ki. A Z80 utasításkészletben további I/O utasítások is vannak, még blokkos adatmozgatást megvalósító utasítások is.

A megszakítás elfogadását (INTA gépi ciklus) az  $\overline{IORQ}$  és az  $\overline{MI}$  egyidejű L szintje jelzi. A nem maszkolható megszakítás-kérés hatására a 66h címről indul a szubrutin. A maszkolható megszakítás-kérés hatása attól függ, hogy a három megszakítási módot beállító utasítás (IM0, IM1 és IM2) közül utoljára melyiket hajtotta végre a processzor. A három megszakítási mód működése a következő:

- IM0: a 8080A megszakítási mechanizmusával megegyező (az INTA ciklusban opkódot fogad az adatbuszról),
- IM1: a 38h címről szubrutint indít,
- IM2: az adatbuszról fogadja a megszakítási vektor alsó Byte-ját (a legkisebb bit 0, ha nem az, a mikroprocesszor felülbírálja, és 0-nak tekinti), a vektor felső Byte-ját az I regiszterből veszi elő, a teljes 16 bites vektorral megcímezett külső memória rekeszből olvassa ki a megszakítási rutin kezdőcíme alsó Byte-ját, a következő memória rekeszből a kezdőcím felső Byte-ját, majd a szubrutinra ugrik.

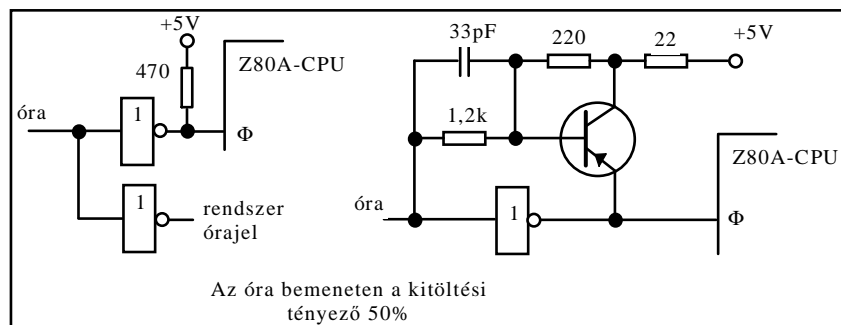
A megszakítást kezelő szubrutint a Z80 esetén nem RET, hanem RETI utasítással szokás lezárni; ez a Z80 számára azonos hatású a RET utasítással, viszont ezt a Z80 programozható kiegészítő áramkörei képesek felismerni és felhasználni (ld. 8.4. alfejezet).

Az **5.15. ábrán** a Z80 M1 gépi ciklusának időbeli lefolyása látható. A Z80 támogatja a DRAM alkalmazásokat, minden M1 gépi ciklusban egy sort frissít a DRAM-ban. A frissítő cím az R regiszter tartalma. Az R regiszter csak hétbites, így 128 soros DRAM memóriák frissítését tudja közvetlenül megoldani a  $\mu P$ . A Z80 fejlesztői törekedtek arra, hogy a DRAM frissítések mindig megtörténjenek a kellő sűrűséggel, ezért pl. a HALT utasítás után a CPU nem áll le a működésekkkel teljesen, hanem NOP ciklusokat hajt végre, mindegyikben elvégezve egy-egy sorfrissítést.



**5.15. ábra**

A Z80-CPU órajele nem TTL szintű, de TTL áramköri elemekkel is előállítható (**5.16. ábra**). Egyszerűbb megoldás +5 V-ról működő CMOS áramkörökkel megépíteni az oszcillátort, akkor az órajel H szintje automatikusan +5 V, L szintje 0 V lesz.

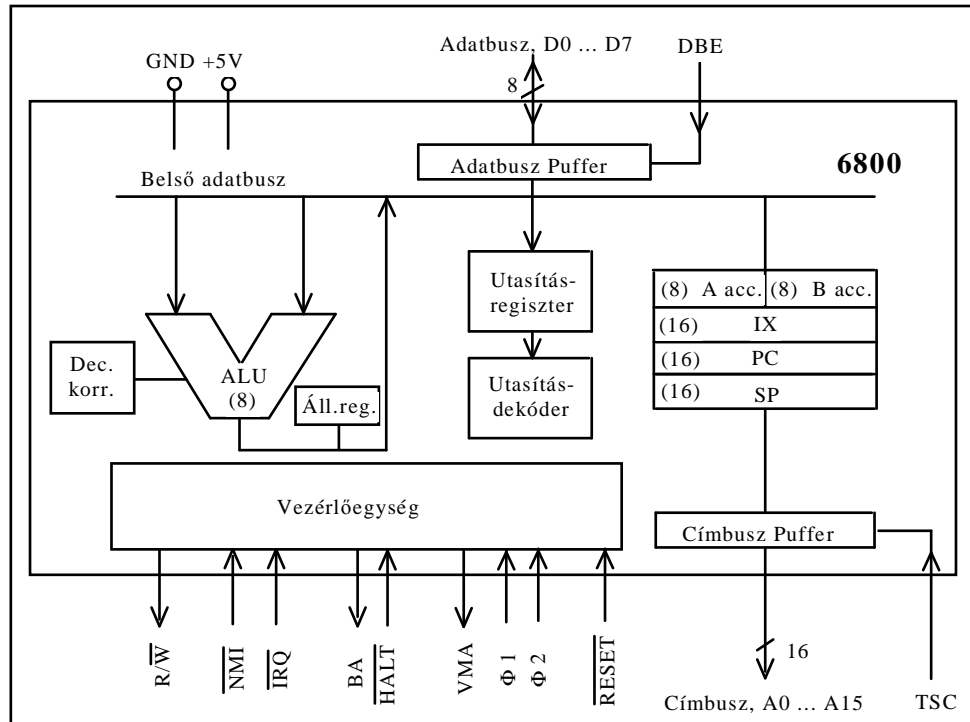


**5.16. ábra**

A bevezetett órajel kitöltési tényezőjének 50 %-nak kell lennie (négyzetöghullám). Ezt úgy lehet elérni, hogy kétszeres frekvenciájú órajelet állítunk elő, amit egy CMOS kivitelű (D vagy JK tárolóból képzett) T tárolón vezetünk át (így egyúttal a kívánt jelszinteket is megvalósítotuk).

## 5.4. A Motorola MC6800 mikroprocesszor

A Motorola kezdetben egy IC-gyártó cég volt, mára már számos különböző terméket (pl. mobiltelefonokat) előállító cég-komplexum. Félvezetőgyárát ma Freescale néven ismerjük.

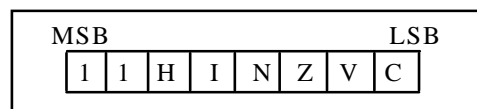


5.17. ábra.

A 6800  $\mu\text{P}$  (amely a gyártója szerint 6800 beépített tranzisztort tartalmaz), az olyan mikroprocesszor-kialakításra példa, amelyiknél nincs I/O működés. Az utasításkészletben nincsenek I/O utasítások, a csatlakozó pontok között nem találunk I/O elemeket aktivizáló kivezetéseket. A 6800 belső felépítése az **5.17. ábrán** látható.

A 6800 egyetlen tápfeszültségű, külső, kétfázisú órajellel működő  $\mu\text{P}$ , melynek adatbusz puffere és címbusz puffere külön vezérlőjellel aktivizálható vagy tiltható. Flag regisztere az állapotregiszter (CCR) nevet viseli, a bitelrendezését az **5.18. ábrán** találjuk meg. Az egyes bitek szerepe a következő:

- C: átvitel a teljes 8 bitről,
- H: átvitel az alsó 4 bitről,
- N: negatív eredmény jelzője,
- Z: zérus eredmény jelzője,
- V: túlcsoordulás jelző,
- I: megszakítás-maszk.



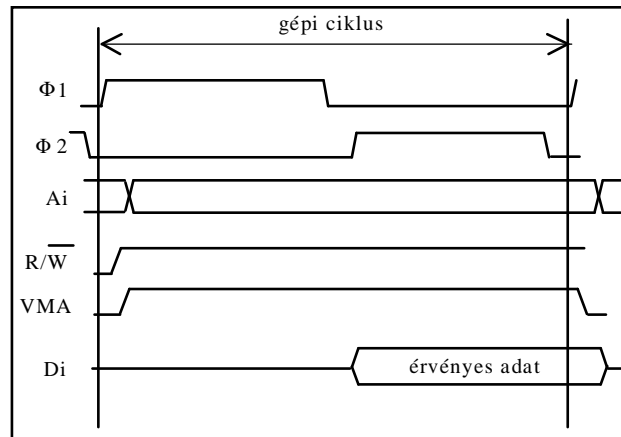
5.18. ábra

A 6800 két általános célú regisztert (A akkumulátor és B akkumulátor) tartalmaz, egy IX indexregisztert, más belső regisztere nincs is. A csatlakozópontok szerepe a következő:

- +5V, GND tápfeszültség csatlakozások,
- D0 ... D7 adatbusz (8 bit),
- DBE adatbusz puffer engedélyező bemenet,
- A0 ... A15 címbusz (16 bit),
- TSC címbusz puffer engedélyező bemenet,
- VMA érvényes memóriacímet jelez a címbuszon,
- $\overline{\text{HALT}}$  HOLD állapotot kérő bemenő jel (L aktív szinttel),



BA	HOLD állapot visszajelzése,
$\phi 1, \phi 2$	kétfázisú órajel (1 ... 10 MHz),
$\overline{R} / \overline{W}$	az olvasási/írási gépi ciklusokat jelző vezérlőjel,
$\overline{NMI}$	nem maszkolható megszakítás-kérő bemenet (L aktív szinttel),
$\overline{IRQ}$	maszkolható megszakítás-kérő bemenet (L aktív szinttel).



5.19. ábra

Az **5.19. ábrán** egy olvasási ciklus látható. Figyeljünk fel arra, hogy egyetlen óraciklus alatt megvalósul egy komplett gépi ciklus.

A 6800-nak, mint említettük már, nincsenek I/O utasításai, így nincsenek I/O kijelölő vezérlő jelei sem, az I/O elemeket a 6800-ra épülő mikroszámítógép memória rekeszekként tudja kezelni.

FFFF RESTART cím alsó byte	FFFE RESTART cím felső byte
FFFD NMI cím alsó byte	FFFC NMI cím felső byte
FFFB SWI cím alsó byte	FFFA SWI cím felső byte
FFF9 IRQ cím alsó byte	FFF8 IRQ cím felső byte

5.20. ábra

Megszakítás elfogadásakor a 6800 automatikusan elmenti a verembe a PC, az index regiszter, a két akkumulátor és a CCR tartalmát s letiltja a további megszakítás-elfogadást. A megszakítások kiszolgálására szolgáló szubrutin kezdőcímet, a RESET utáni indulási címet, a megszakítás-utasítás (szoftver megszakítás) hatására aktivizálódó szubrutin kezdőcímét a 6800 egy táblázatból olvassa ki, ami a lehetséges 64 Ki memóriaterület legtetején helyezkedik el (**5.20. ábra**). (A RESET utáni indulási címet nevezi a Motorola RESTART címnek, a megszakítás-utasítás az SWI.)

## 5.5. A nagyobb szóhosszúságú Intel mikroprocesszorok első változatai

Az 1970-es évek közepén a mikroprocesszorok népszerűsége rohamosan nőtt, de a felhasználók egyre igényesebbek lettek. A három fő követelmény a működési sebesség fokozása, a szóhosszúság növelése és a  $\mu P$  chip komplexebbé tétele volt. Abban az időben a chip gyártás színvonalja nem tette lehetővé a három igény egyidejű teljesítését, ezért több fejlesztési irány is létrejött. Az Intel is három fejlesztési projektet indított el, ezek a következők voltak:

- a bipoláris bitszeletelt mikroprocesszorok, a nagyobb sebesség érdekében (mára ez a típus megszűnt),

– a komplex, teljes mikroszámítógépet tartalmazó chip, a mikrovezérlő (ma is gyorsan fejlődő terület),

– a 16 bites mikroprocesszorok, ezekről lesz szó a következőkben.

A nagyobb szóhosszúságú mikroprocesszorok fejlesztésének első állomásait mutatjuk be a következőkben.

8086 (1976)	16 bites adat, 20 bites cím, 1 Mi fizikai memória, 29.000 beintegrált tranzisztor, 40 kivezetéses DIL tok, 6 byte utasítás előolvasó tár (queue); két független belső vezérlő egység.
8088 (1978)	a 8086-hoz hasonló felépítésű, de 8 bites adatbuszon kommunikál a környezetével.
80186 (1980)	8086 és kiegészítő elemei egy chip-en, 40 kivezetéses DIL tokban.
80286 (1983)	16 bites adat, 24 bites cím (16 Mi fizikai és 1 Gi virtuális memória), 134.000 tranzisztor funkció, 68 kivezetéses tokozások, 6 Byte queue, 3 Byte utasítás elődekódolás.
80386 (1985)	32 bites adat, 32 bites cím, 4Gi fizikai és 64 Ti virtuális memória, 275.000 tranzisztor funkció, 132 kivezetéses tokozás, 16 Byte queue, 2 teljesen dekódolt utasításos pipe-line.
80486 (1989)	32 bites adat, 32 bites cím, 4 Gi fizikai és 64 Ti virtuális memória, 1.180.235 tranzisztor funkció, 168 kivezetéses tokozás, 32 Byte queue, két teljes pipe-line utasítás feldolgozó, beépített aritmetikai társprocesszor, a chip felület 15 %-án egy 386-os található.
PENTIUM (1994)	32 bites adat, 32 bites cím, 4 Gi fizikai és 64 Ti virtuális memória, az első változat 5V tápfeszültségű, 3,1 millió tranzisztort tartalmazó, fogyasztása 16W; a végleges tápfeszültsége 3,3V, beépített tranzisztorai száma 3,3 millió, a fogyasztása 4 W; a külső cache adatútja 64 bites, 2+1 teljes pipe-line működés, a chip felület 4 %-án egy komplett 486-os található; belső 8 Ki utasítás- és 8 Ki adat cache, 238 kivezetéses tokozás.
PENTIUM PRO (1996)	32 bites adat, 32 bites cím, 4 Gi fizikai és 64 Ti virtuális memória, az IC tokban két chip van, egy 256 KiByte kapacitású cache (15,5 millió tranzisztor) és a processzor (5,5 millió tranzisztor); a külső memóriával 64 bites adatúton tartja a kapcsolatot, 2,9 V tápfeszültségen 20 W a fogyasztása, 387 kivezetéses tokozással készül, három utas szuper-skalár, 14 fokozatú a pipe-line sora, az utasításokat belső mikrokódra bontja, dinamikus elágazás jelzésének találati valószínűsége 90 % feletti, a tokon belüli cache találati valószínűsége jobb mint 90 %.

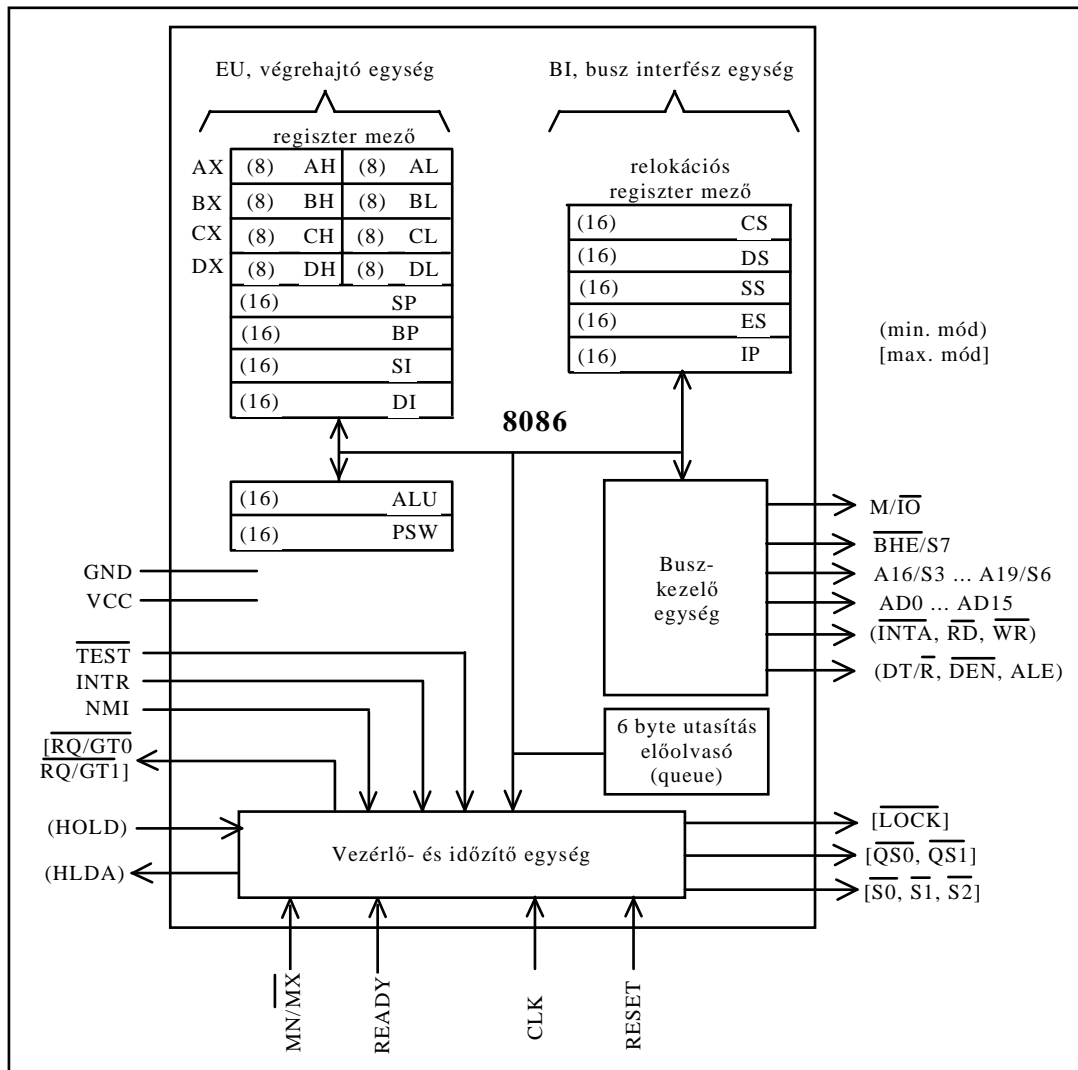
A nagyobb szóhosszúságú mikroprocesszorok közül a 8086/8088-cal és a 80286-tal fogunk részletesebben foglalkozni. Elsősorban az újszerű megoldásokra hívjuk fel a figyelmet, hogy az újabb és újabb fejlesztésű  $\mu P$ -k leírásaiban ezek alapján már önállóan is el lehessen igazodni.

### 5.5.1. Az Intel 8086/8088 mikroprocesszorok

A 8086 és a 8088 tulajdonképpen azonos, 16 bites  $\mu P$ -ok, az eltérés a külső adatbusz-csatlakozásban van. A 8086 16 bites adatbusszal kapcsolódik a környezetéhez, a 8088 csak 8 biteset használ (a 8088 minden 16 bites adat mozgatására két ciklust használ fel). További eltérések a két mikroprocesszor között: a 8086 esetében az utasítás előolvasó tár, a queue 6 Byte méretű, a 8088-ban csak 4 Byte. Mivel csak utasítás előolvasás történik, elő-dekódolás vagy elő-végrehajtás nem, ezt még nem is tekintik valódi pipe-line működésnek.

A 8086/8088 az Intel első 16 bites mikroprocesszor-sorozata, a 8086 belső felépítését az **5.21. ábra** mutatja be. Az  $\overline{MN}/\overline{MX}$  csatlakozón lehet H szinttel a minimum, L szinttel a maximum üzemmódot kijelölni.

Minimum módban a 8086 maga állítja elő a vezérlőjeleket, kisebb mikrogepekben ez a megoldás használatos. Maximum módban a  $\mu P$  státuszbiteket ad ki, amiket egy külső rendszervezérlő dolgoz fel, az állítja elő a vezérlőjeleket is.



5.21. ábra.

A 8086 40 kivezetéses DIL tokban készül, 8 csatlakozópontja üzemmód függő. Az **5.21. ábrán** gömbölyű zárójelben olvasható a minimum módban érvényes jel elnevezés és szögletes zárójel utal a maximum módban érvényes szerepre.

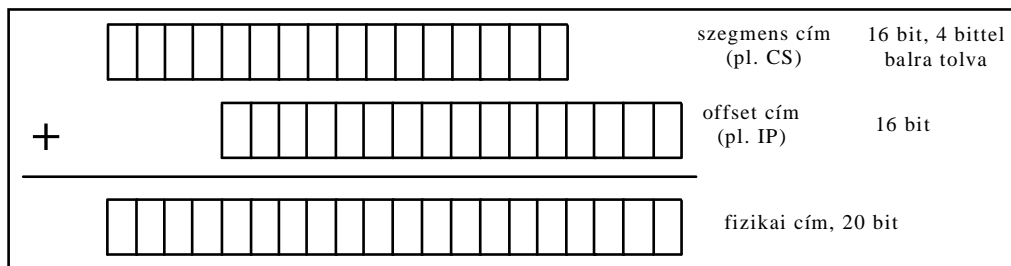
A 8086 két belső vezérlőegysége párhuzamosan, egymástól függetlenül működik, így képes az utasítások végrehajtása közben (ami az EU, Execute Unit feladata) a queue rekeszeibe beolvasni a külső memóriából a következő utasításelemeket (a BI, a Bus Interface használatával). Ha a program normál sorrendű utasítás-végrehajtással fut, az előolvasás jelentős mértékben gyorsítja a működést. Maximum módban a queue állapotáról, működéséről is ad információt a 8086, a  $QSi$  státuszbiteken keresztül.

Négy általános célú, 16 bites regiszterét (AX, BX, CX, DX) 8 bites részletekben (AH, AL stb.) is lehet használni. Az utasítások egy része alap- (default) módon már egy vagy több regiszter szerepét is előírja. Ezeket a default előírásokat a programozó felülbírálhatja, de akkor a programkód terjedelmesebb lesz, a futási idő is megnő.

A 8086 szegmentált módon kezeli a 20 bites címbusszal kezelhető 1 MiByte méretű fizikai memória területét. A cím előkészítése (logikai cím) 32 biten valósul meg, két 16 bites rész útján: az egyik a szegmens cím, a másik az offset (eltolás). A logikai címet a szegmenscím:offset for-

mában szokás felírni. A fizikai cím 20 bites, egy 4 bites eltolás után adódik össze a két 16 bites elem, a legfelső bitről ilyenkor az esetleges átvitel elvész (**5.22. ábra**). A szegmensök egymást így jelentősen átlapolhatják, egy szegmens mérete 64 KiByte. A szegmensregiszterek (CS: kód-szegmens, DS: adatszegmens, ES: extraszegmens, SS: veremszegmens) és az offset regiszterek (IP: utasításszámláló, SP: veremtár-mutató, BP: kisegítő veremtár-mutató, SI és DI - adatszegmens pointerek) általában nem kombinálhatók tetszőlegesen.

Az utasításelemek beolvasásához a CS:IP regiszterek tartalmát használja fel a  $\mu P$ , a verem



**5.22. ábra**

címzésére az SS:SP vagy az SS:BP logikai címeket. Az adatkezelésekhez is default előírások tartoznak, de a DS, ES valamint az SI, DI már szabadabban kezelhetők.

A 16 bites mikroprocesszorban természetesen az ALU is 16 bites és a program állapot szó (PSW, Program Status Word) is, ami a 8 bitesek F regiszterének utódja (**5.23. ábra**). A PSW bitek szerepe a következő:

Flag bitek:	CF	átvitel bit,
	PF	paritás bit,
	AF	fél-átvitel bit,
	ZF	zérus eredményt jelző bit,
	SF	előjel bit, negatív eredményt jelez,
Vezérlő bitek:	OF	túlsordulást jelző bit.
	DF	string műveleteknél a címváltoztatás iránya, DF = 1: csökkenő címek; DF = 0: növekvő címek,
	IF	megszakítás-engedélyező bit,
	TF	utasítás végrehajtás utáni TRAP megszakítás-engedélyező bit,
		TF = 1: minden utasítás után TYPE1 jellegű megszakítás generálódik, az ennek megfelelően előírt címről szubrutin indul; TF=0 esetén mindez elmarad.



**5.23. ábra**

A 8086 csatlakozópontjainak szerepe a következő:

GND, VCC	tápfeszültség-csatlakozások,
AD0 ... AD15	multiplexelt cím/adatbusz, a T1 óraciklus alatt címet tartalmaznak, később itt zajlik az adatmozgás,
A16/S3, A17/S4	multiplexelt cím/státusz bitek, T1 óraciklusban címet tartalmaznak, később a státusz biteket. Az S4, S3 az aktuális szegmenst jelzik: S4, S3 = 00: Extra szegmens (ES) 01: Verem szegmens (SS) 10: kód szegmens (CS) 11: adat szegmens (DS),

A18/S5	multiplexelt cím/státusz bit, T1 óraciklusban címet tartalmaz, később a státusz bitet. Az S5 a megszakítás-engedélyező Flag bit (IF) értékét közvetíti,
A19/S6	multiplexelt cím/státusz bit, T1 óraciklusban címet tartalmaz, később a státusz bitet, az S6 későbbi fejlesztésre van fenntartva,
$\overline{\text{BHE}} / \text{S7}$	multiplexelt vezérlő jel/státusz bit, T1 óraciklusban a páratlan fizikai címeket tartalmazó, felső Byte memória felet kezelő vezérlő jel lép ki rajta, később az S7 állapot bit (az S7 későbbi fejlesztésre van fenntartva),
$\overline{\text{RD}}$	olvasási jellegű gépi ciklust jelző vezérlő jel (L aktív szinttel),
READY	alacsony szintje esetén a T3 és a T4 óraciklusok közé a $\mu\text{P}$ várakozási TW) ciklusokat iktat be,
$\overline{\text{TEST}}$	a WAIT utasítás végrehajtásakor vizsgálja az itt lévő bemenő jelet a processzor, L szint esetén a működés folytatódik; H szint esetén ún, idle (tétlen) módba kerül és várakozik a $\overline{\text{TEST}} = \text{L}$ szint beérkezésére,
INTR	maszkolható megszakítás-kérő bemenet,
NMI	nem maszkolható megszakítás-kérő bemenet,
RESET	alaphelyzetbe állító jel,
CLK	órajel bemenet, a kitöltési tényező 33 %,
$\text{MN} / \overline{\text{MX}}$	a minimum/maximum módot átkapcsoló bemenő jel.

A további csatlakozópontok szerepe üzemmód-függő. Maximum módban az alábbi jelcsatlakozások kezelhetők:

$\overline{\text{S0}} \dots \overline{\text{S2}}$	státusz információ kimenetek ( <b>5.24. ábra</b> ):
---	---

státuszbit			értelmezés
$\overline{\text{S2}}$	$\overline{\text{S1}}$	$\overline{\text{S0}}$	
0	0	0	Megszakítás-elfogadás
0	0	1	I/O olvasás
0	1	0	I/O írás
0	1	1	HALT utasítás végrehajtás
1	0	0	utasításkód beolvasás (M1)
1	0	1	memória olvasás
1	1	0	memória írás
1	1	1	nincs busz ciklus

5.24. ábra

QS0, QS1 a queue állapotáról informáló státusz bitek (**5.25. ábra**):

bitértékek		jelentés
QS0	QS1	
0	0	nincs queue művelet
0	1	opkód első byte a queue-ból
1	0	a queue üres
1	1	a következő byte a queue-ból

5.25. ábra

$\overline{\text{RQ0}} / \overline{\text{GT0}}$  több-processzoros esetben itt lehet elkérni a busz feletti felügyelet jogát, ugyanitt jelzi a 8086, hogy átadja a busz rendszert, de azt is itt tudja később jelezni, ha ismét szüksége lenne a busz-okra.

$\overline{RQ1} / \overline{GT1}$ az  $\overline{RQ0} / \overline{GT0}$ -al azonos működés, de alacsonyabb prioritási szinttel, $\overline{LOCK}$ 

a buszrendszert lezáró jel, hosszú végrehajtási idejű utasítások közben idegen egység buszhasználatát lehet így megelőzni.

Minimum módban a következő jeleket kezeli a 8086:

 $\overline{DEN}$ 

adatmozgás engedélyező jel az adatbusz meghajtók számára,

 $\overline{DT} / \overline{R}$ 

adatmozgás irányának jelzése; olvasáskor L szintű, íráskor H,

 $\overline{M} / \overline{IO}$ 

memória (H) vagy I/O (L) működést jelző vezérlőjel,

 $\overline{INTA}$ 

megszakítás-elfogadás gépi ciklus jelzése,

 $\overline{HOLD}$ 

a busz-ok lebegtetését (HOLD állapot) kérő jel,

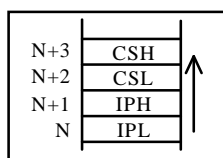
 $\overline{HLDA}$ 

a HOLD állapot visszajelzése,

 $\overline{WR}$ 

az adatbusz-ra kiküldött adat érvényességének jelzése.

A 8086 két címzésmóddal kezel I/O elemeket. A közvetlen címzésű utasítás nyolcbites I/O címet tartalmaz, a közvetett (regiszteres) címzésű I/O utasítás esetén az I/O cím a DX regiszterben van, tehát 16 bites. Az I/O gépi ciklusokban a nem használt címbitek 0 áll.

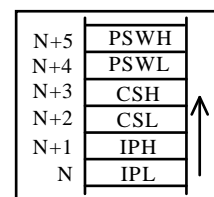


5.26. ábra

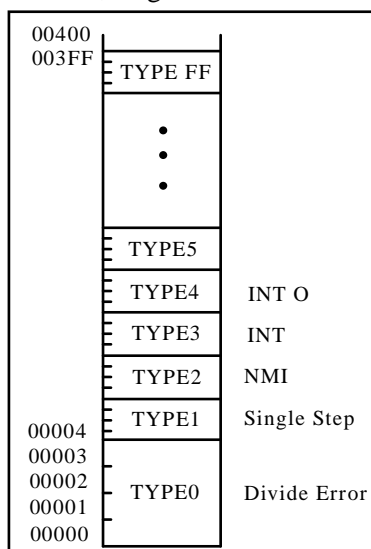
A megszakítások tárgyalását a paraméteres megszakítás-utasítás, az INT N bemutatásával célszerű indítani. Az n a megszakítás típuszáma (TYPE), nyolc bites érték, azaz decimálisan a 0 ... 255 tartományba eshet. A memória induló területén, a legelső 1 Kbyte részen (0h ... 3FFh) van egy megszakítási ugrási tábla. Ebben minden n értékhez négy Byte tartozik, a 4 Byte egy teljes logikai címet tartalmaz, az 5.26. ábra szerinti elrendezésben.

A típuszámot négygyel megszorozva adódik a hozzá tartozó négy rekesz közül az elsőnek a címe.

A megszakítási rutinra ugráskor a CPU automatikusan menti a PSW tartalmát és a visszatérési címet (logikai cím formájában), mindezt a verembe helyezi be (5.27. ábra). A CALL szubrutinra ugró utasítás csak a visszatérési címet menti, a RET visszatérítő utasítás is csak azt tölti vissza. A megszakítást kezelő rutin végén ezért nem a RET áll, hanem egy újabb utasítás, az IRET, ez gondoskodik a PSW helyreállításáról is.



5.27. ábra



5.28. ábra

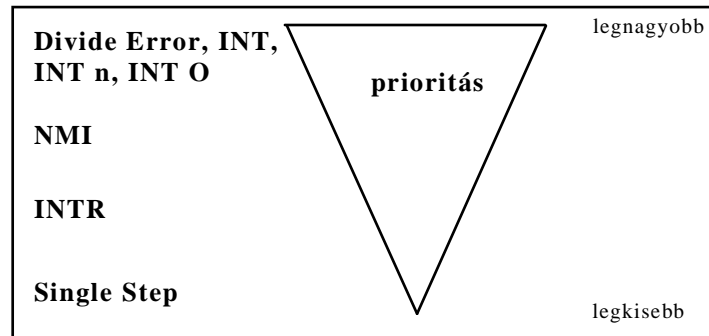
A 8086 utasításkészletében további megszakítás-utasítások is vannak. Az INT O a túlcordulás (OF) értékét feltételként használja, akkor valósul meg a megszakítás, ha OF = 1; az utasítás a TYPE4 típuszámot aktivizálja. A paraméter nélküli megszakítás-utasítás, az INT a TYPE2 típuszámot használja.

A 8086 hardver-megszakítás elfogadásakor törli a PSW-ben a TF és IF biteket. A maszkolható megszakítás elfogadását két INTA ciklus jelzi, a másodikban az adatbusz alsó felén a hardvernek be kell küldenie egy 8 bites vektort - ami a megszakítás típusszámát adja meg. A vektorral a megszakítási tábla bármely eleme ugyanúgy kiválasztható, mint az INT N utasítással. Az NMI is a táblázatra lép, de kötött pontra - a TYPE2 típuszámot aktivizálja, azaz a 08h címnél kezdődik az NMI rutin kezdőcíme.

A 8086 egyes belső működései is megszakítást váltanak ki. Ha osztáskor az eredmény az ábrázolható számtartományánál nagyobb lenne (pl. nullával való osztáskor), a TYPE0

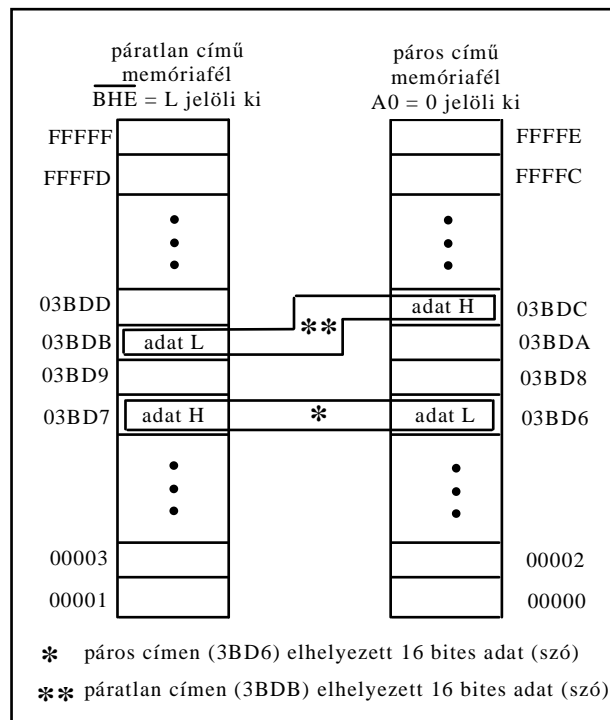
megszakítás jön létre. Ha a TF flag 1 értékű, minden utasítás végrehajtása után TYPE1 típuszámú megszakítás generálódik.

Az **5.28. ábra** mutatja be a megszakítási tábla egyes részleteinek felhasználását. A megszakítási lehetőségek prioritását az **5.29. ábráról** lehet leolvasni.



**5. 29. ábra**

A 8086 mellett a fizikai memóriát nyolcbites memória IC-kból kell felépíteni. Az 1 Mi memória csak 512 Ki 16 bites adat elhelyezését biztosítja így. A 8086 megengedi, hogy a 16 bites adat páros vagy páratlan címen legyen a memóriában elhelyezve (**5.30. ábra**). A páratlan fizikai címek memóriafelét a  $\overline{\text{BHE}} = \text{L}$ , a páros fizikai címek memóriafelét az  $\text{A0} = 0$  aktivizálja. Páros címről az adatot egy ciklusban lehet kezelni (mindkét kijelölő jel aktív), páratlan címről két ciklus szükséges (a vezérlő jelek egyenként, felváltva aktívak).



**5.30. ábra**

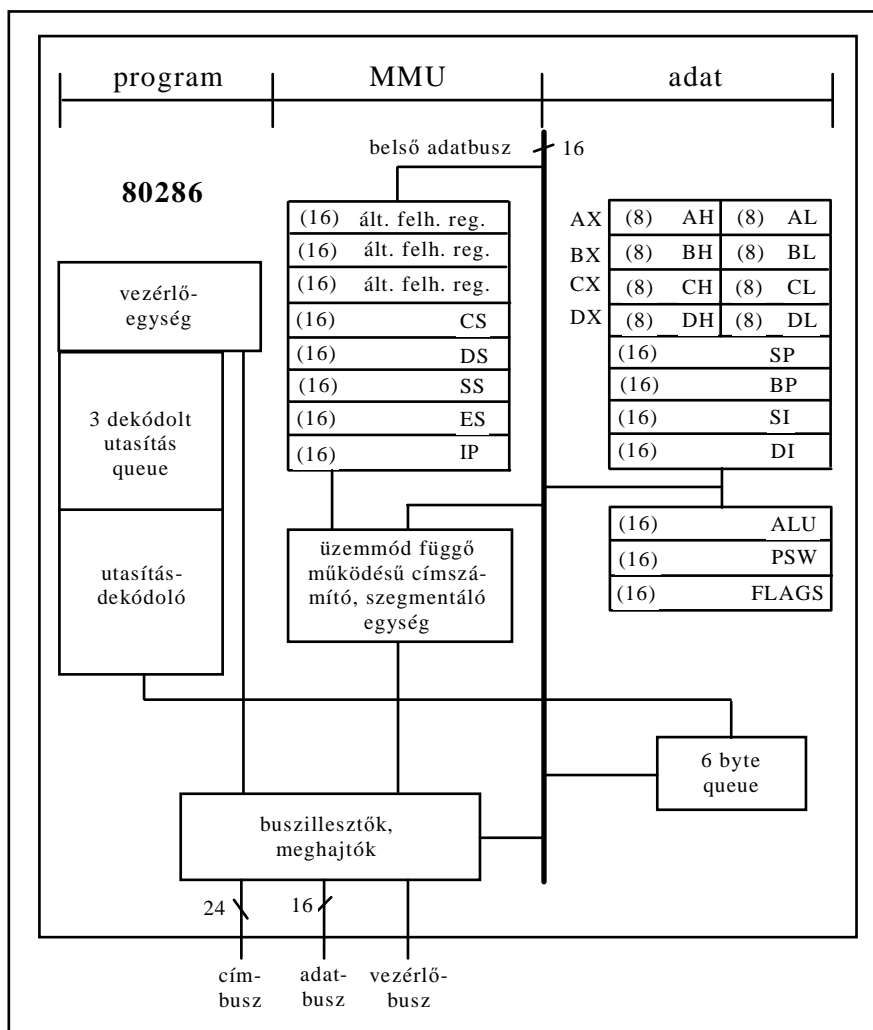
A RESET folyamat hatására kialakuló alapállapot:

CS = FFFFh,  
DS, ES, SS, IP = 0,  
PSW = 0 ,

tehát a processzor RESET utáni kezdőcíme FFFF0h. Mivel ez a fizikai memória tetején van, a 8086 és 8088 melletti memóriában a felső részen, az FFFFh címtől lefelé található a ROM területe, a 00000h címtől, alulról pedig a RAM. Utóbbi ilyen elhelyezését a megszakítási táblázat is indokolja.

### 5.5.2. Az Intel 80286 mikroprocesszor

Az Intel 80286 már multitasking jellegű szoftver kezelésre alkalmas. Felülről kompatibilis a korábbi 16 bites processzorokkal (az Intel a későbbiekben is minden mikroprocesszoránál megvalósította a felülről kompatibilitást). 16 bites  $\mu P$ , a címbusz 24 bites; a fizikai memória így 16 MiByte lehet, a virtuális címezési lehetőség 1 GiByte. A védett virtuális módban a multitasking üzem rugalmas memóriacímzési megoldása, védelmi rendszere és task-menedzselő lehetőségei biztosítják. A memóriacímzést szegmens-leíró táblázatokon keresztül végzi ilyen esetben. A busz átviteli sebessége 12,5 MiByte/s.



5.31. ábra.

A 80286 két üzemmódban használható:

- valós üzemmód (Real Address Mode), a 8086-ot emuláló üzemmód,
- védett virtuális üzemmód (Protected Virtual Mode), ez alkalmas a multitasking működések megvalósítására.

A 80286 fejlesztése még abban a korszakban valósult meg, amikor az LSI gyártók új termékei mutatták a digitális technika fejlesztői számára a továbblépési lehetőségeket. Ekkor jelentette ki az Intel egyik vezetője, hogy egy hónap alatt komolyabb számítástechnikai kapacitást állítanak elő, mint amit az IBM számítógépek formájában egy év alatt kibocsát. És akkor ez így is volt. Megjelent a 80286 a multitasking képességével és az IBM AT gépeken futó operációs rendszer, a DOS nem tudta ezt a lehetőséget kihasználni.

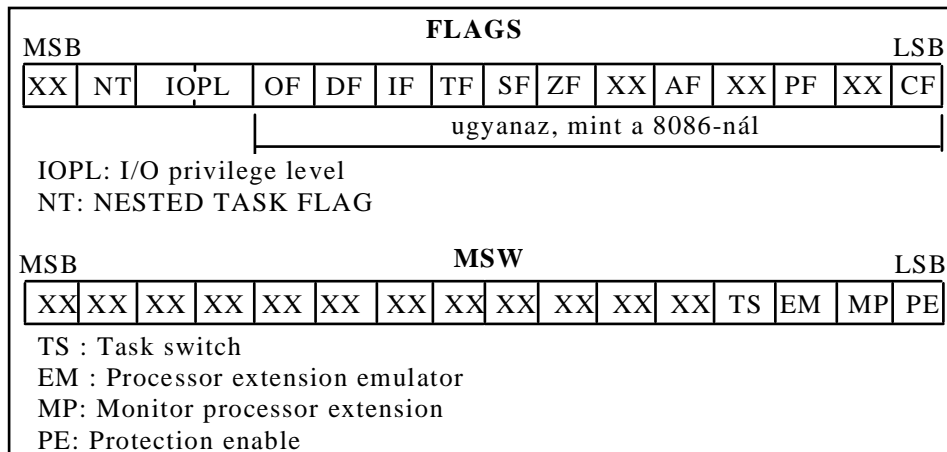


Mire elterjedt a WINDOWS, ami a multitasking működésre épül, az Intel már megjelentette a 80386-os mikroprocesszort, a 286-os fejlesztések lezárultak. Így valószínűleg a 80286-os mikroprocesszorok többsége úgy fejezte be pályafutását, hogy soha nem kapcsolták át a védett virtuális módba.

A 68 kivezetéses tokozás révén már nincsenek multiplexelt, többfunkciós csatlakozópontok, nincs rájuk szükség. A queue itt is 6 Byte, de már egy 3 Byte méretű utasítás-elődekódoló is működik a  $\mu$ P-ben. A 80286 belső felépítését az **5.31. ábra** mutatja be. Ezzel a mikroprocesszorral már nem ismerkedünk meg teljes részletességében, elsősorban a multitasking lehetőségekre koncentrálunk a továbbiakban.

A 8086 állapotregiszteréhez képest sok újdonság látható a 80286 állapotregisztereiben, erre a célra itt két regiszter is szolgál, az F és az MSW (Machine Status Word) regiszter (**5.32. ábra**). Az új bitek szerepe a következő:

- NT: beágyazott task jelző, akkor lesz 1 értékű, ha a taskot egy másikon belül aktivizálták,
- IOPL: I/O védelmi szint jelző (0 ... 3 értékű lehet), a program csak akkor használhatja az IN, OUT, LOCK, STI, CLI utasításokat, ha védelmi szintszáma nem magasabb ennél az értéknél,
- PM: védett módot elrendelő bit (a védett módba belépéshez 1-re kell váltani),
- MP: matematikai társprocesszor jelzője,
- EM: matematikai emuláció jelzés, ha 1 értékű, a társprocesszor működését helyettesítő emuláló szoftver futtat le,
- TS: taszkváltás jelző, 1-re változik minden taskváltásnál.



**5.32. ábra**

A belső regiszterek között megtaláljuk a 8086 teljes regiszterkészletét, beleértve a szegmens-regisztereket, offsetregisztereket is. A multitasking működést négy új regiszter segíti:

- GDIR: globális-tábla regiszter (a GDT globális leíró tábla címe és hossza található meg benne),
- IDTR: megszakítás-tábla leíró regiszter (az IDT megszakítás-leíró tábla címe és hossza van benne),
- LDTR: lokális-tábla regiszter (az LDT lokális leíró tábla címével és hosszával),
- TR: task regiszter, a task állapot szegmensre (TSS) utaló szelektort tartalmazza, a TSS helyére vonatkozó egyéb adatok (cím, méret) a GDT-ben vannak.

(A leíró táblákkal, a szelektor fogalmával rövidesen találkozni fogunk).

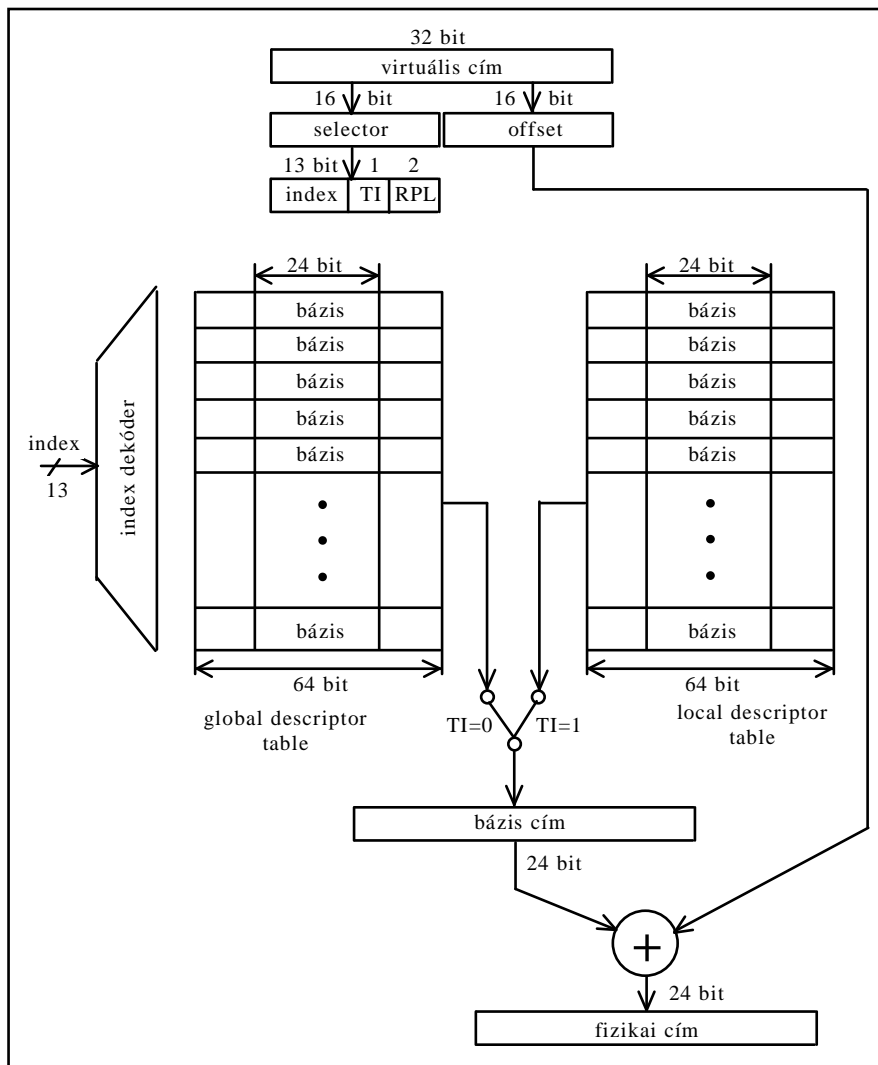
A 80286 RESET után valós üzemmódban indul, a PE bit 1-re állításával lehet védett módba vezérelni. Innen már szoftver úton nem vihető vissza a valós módba, csak RESET folyamattal.

A 80286 megszakítási viselkedése valós módban megegyezik a 8086 megszakítási működésével, védett virtuális módban a megszakítási címek kezelése is szegmensleíró táblázaton keresztül (IDT) történik. Valós módban a 8086-nak megfelelően a 80286 20 bites fizikai címeket használ, ugyanazzal a technikával számítja ki a fizikai címet a szegmens és offset értékből. Védett módban is két 16 bites érték alkotja a logikai címet, de a fizikai cím kialakítása összetettebb úton megy végbe.



5.33. ábra

A két részlet egyike továbbra is az offset, a másik új nevet és szerepet kapott, ez a szelektor.



5.34. ábra

A szelektor (5.33. ábra) részletek szerepe a következő:

RPL: igényelt védelmi szint, 0 ... 3 közötti védelességi szintérték,

TI: leíró tábla jelző (Table Indicator), a task által kezelhető két szegmensleíró tábla egyikét jelöli ki

TI = 0: a globális leíró tábla használata

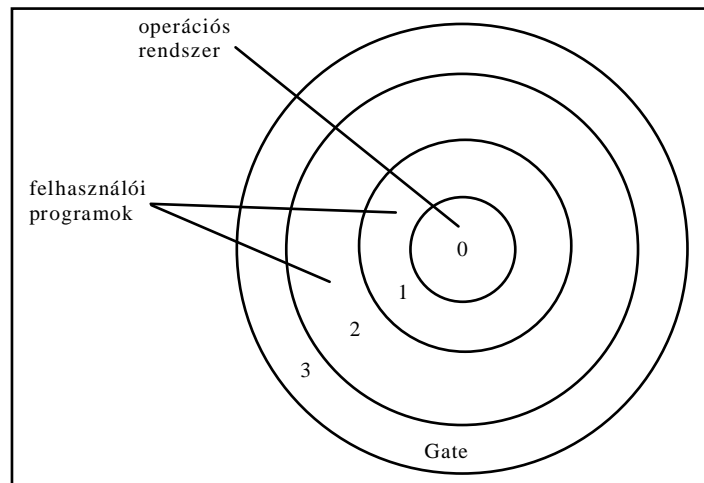
TI = 1: a lokális leíró tábla használata,

INDEX: a szegmensleíró táblán belüli, 13 bites cím.

A címkialakítás szegmensleíró táblán (Segment Description Table) keresztül megy végbe (5.34. ábra). A szoftver rendszer minden task-hoz hozzárendel egy saját, lokális szegmens-leíró táblát, s a rendszernek van egy közös, globális szegmens-leíró táblája, ez utóbbit minden task használhatja.

A szegmens-leíró táblák rekeszei 64 bitesek, ezek egyikét jelöli ki a 13 bites index érték. A szegmensleíró táblából kiválasztott 64 bites szegmens-leíró információk belül található a 24 bites bázis cím, ez, túlcsoportolás képzése nélkül, helyérték-helyesen (eltolás nélkül) összeadásra kerül az offset 16 bitjével, s így jön létre a 24 bites fizikai cím.

A 80286 négy védettségi szintet ismer (0 ... 3), az alacsonyabb szám jelenti a magasabb védettségi szintet (5.35. ábra). Egy task csak a nála kevésbé védett programelemeket érheti el. Az operációs rendszert a 0-s védettségi szinten kell elhelyezni, a felhasználói programok az 1-es, 2-es szintre kerülnek. A 3-as szinten ún. kapu (Gate) leíró tábla is telepíthető, ez szabályozott módon lehetővé teszi a felhasználói programoknak azt, hogy pl. a rendszer egyes rutinjait felhasználják.



5.35. ábra

A 80286, ha egy információt kiolvass a leíró táblából, azt egy árnyékrejiszterben helyezi el. A következőkben, ha a szelektor nem változik, csak az offset módosul, nem kell végigdolgozni a táblázatkezelést, csak az új offset értékkel történő végső összeadást kell ismét elvégezni.

A 80286 megismert képességei többsége nem az Intel találmánya. A 8088/8086 fejlesztésével egy időben pl. a Zilog már kidolgozta a Z8000 sorozatú 16 bites mikroprocesszorait, s azok már a multitasking működéshez egy hasonló táblázatos memória-menedzselő rendszert tartalmaztak. Tulajdonképpen az Intelnek éppen azért kellett a 286-ost megalkotnia, mert a versenytársak abban a pillanatban nagyobb teljesítményű processzorokat kínáltak.

A továbbiakban a mikroprocesszor-gyártó, esetünkben az Intel, a hardver fejlesztője (a PC esetében az IBM) és a szoftver szállítója (az IBM PC-knél elsősorban a Microsoft) együtt alakítja ki a fejlesztési lépéseket, ezen belül többek között a következő mikroprocesszorokkal szembeni igényeket.

Sajátos módon az igénylistán kiemelkedő helyen szerepel az újabb és újabb számítógépes játékok követelménysora. A játékok egyre nagyobb számítási teljesítményt igényelnek a processzortól, s hamarosan szinte önálló mikroszámítógéppé válik a PC-n belül a grafikus kártya, grafikus rendszer. A játékokhoz szükséges grafikus és számítási megoldások azután, ha sikeresek, az egyéb felhasználói szoftverekbe is beépülnek. A PC-n számítógépes játékokat futtatók szinte tesztelik a hardver és a szoftver fejlesztők legfrissebb ötleteit, megoldásait.

Azért választják ezt a megoldást a gyártók, mert ha egy játékprogram lefagy, akadozik, azon legfeljebb a fiatalok bosszankodnak, jelzik a hibát, korrigálható a hardver, a szoftver. Az üzleti

és a tervezői programok esetében az ilyen hibák akár jelentős veszteséget okozhatnak, ezt kívánják elkerülni úgy, hogy ott csak kipróbált, alaposan tesztelt megoldásokat használják.

A mikroprocesszor működési sebessége rohamosan nő, a memória-áramkörök nem tudnak vele lépést tartani, legalábbis a PC főmemóriája, a DRAM. Hiába az újabb és újabb szervezési megoldások sora, közvetlenül a mikroprocesszor nem tudja a programokat a DRAM-ból futtatni. Ahogyan 1980 körül kialakult, hogy a merevlemezről a programot a DRAM-ba kellett tölteni, úgy kellett a program éppen használt részét SRAM-ból kialakított cache-memóriába helyezni, hogy onnan futhasson. Mára ezért alakult ki a többszintű cache rendszer.

A mikroprocesszor ma több GHz-en fut, a DRAM modulok GHz körüli órajellel, amit esetenként többszörözéssel állítanak elő. Ezek az órajelek és a PC egyéb elemeinek órajelei nem lehetnek egymástól függetlenek, ezért minden órajel az alaplapon lévő kvarcoszcillátor jeleiből képződik. Ezt szorozza fel a mikroprocesszor belső áramköre 2...4 GHz-re, ezt többszörözik a DDR DRAM modulok, ezt osztják le a lassúbb buszok, a merevlemez-illesztők és az illesztő áramkörök.

## 6. Nyolcbites CPU kialakítása

A 3. fejezetben megismertük a mikroszámítógépek általános felépítését, láttuk a központi feldolgozó egységet, a CPU feladatait. A CPU működése a mikroszámítógépekben a mikroprocesszorra épül, de általában más elemeket is fel kell használni a kialakításához. A mikroszámítógép CPU egységének feladatait legtöbbször csak több-kevesebb kiegészítő elemmel együttműködve képes ellátni a mikroprocesszor. A következőkben példaként azt vizsgáljuk meg, hogy a megismert Intel és Zilog nyolcbites mikroprocesszorokkal hogyan lehet CPU-t építeni.

A nyolcbites mikroszámítógép CPU kialakítási lehetőségeinek megismerése több szempontból is célszerű. Ezek a CPU egységek jól áttekinthetőek, a működésük minden részlete követhető. A mikrovezérlőkkel történő készüléktervezéshez a mikroprocesszoros CPU tapasztalatai igen jól felhasználhatók. Végül, a mai nagyobb szóhosszúságú mikroprocesszorokból épített CPU működésének számos részlete továbbra is hasonló, mint a 8 bites egységeké volt, így azok tanulmányozásához is hasznosíthatók az itt megszerezhető alapismeretek.

### 6.1. 8080A-ra épülő CPU

A 8080A 20-25 SSI-MSI áramkörrel egészíthető ki működő CPU egységgé. Meg kell oldani a következőket:

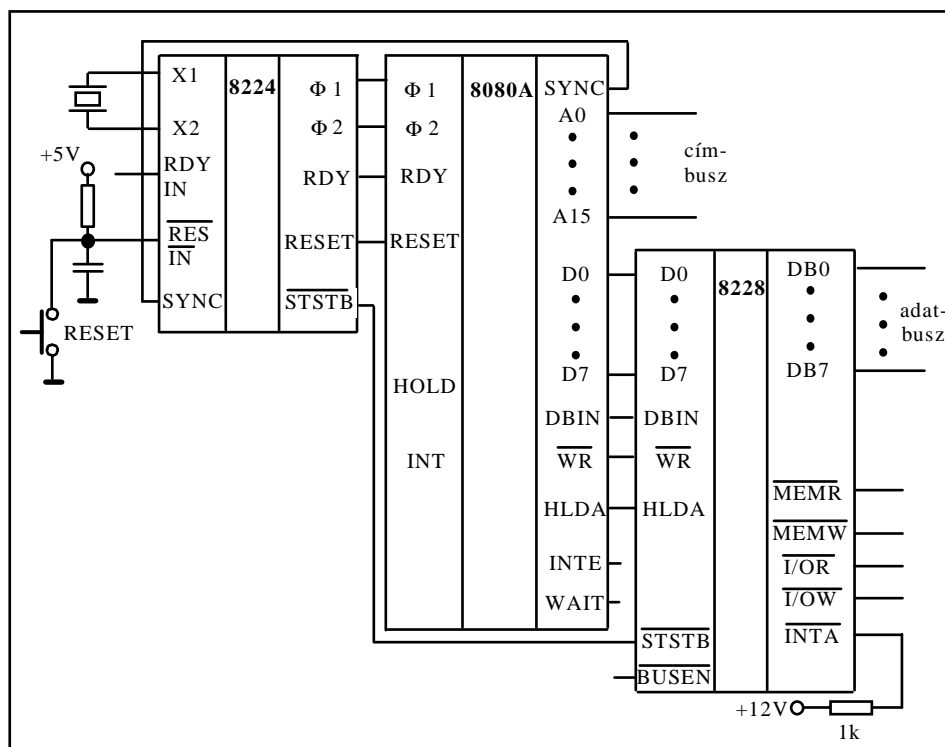
- kétfázisú, +12V-os órajel előállítás,
- RESET funkciók (bekapcsolási, kézi),
- státusz információk lekapuzása, tárolása, átkódolása vezérlő jelekké,
- bemenő jelek szintátalakítása,
- kimenő jelek terhelhetőségének biztosítása.

1974-ben az Intel a 8212 sokfunkciós LS bipoláris latch (D tárolós regiszter) áramkört ajánlotta segítségképpen (később ez az áramkör bekerült a 74-es LS sorozatba is, 74LS412 típusjelzéssel). A 8212 LS bipoláris jellegének köszönhetően kis bemeneti áramot igényel, a kimeneti H szintje magasabb, mint a standard TTL elemeké - így egyszerű illesztőelemnek bizonyult a 8080A és a TTL kompatibilis környezete között. A 8212 segítségével meg lehetett oldani a státusz szó lekapuzását és tárolását, az adatbusz és a címbusz erősítését, a megszakításkor a megfelelő paraméterű RST utasítás beküldését, a bemeneti és a kimeneti Port-ok kialakítását egyaránt. A 8080A kifejlesztésekor egy további LS bipoláris áramkört is kialakított az Intel, a 8205-öt, ami egy 3 bites, nagy sebességű, vezérelhető címdekóder. Ez az áramkör is megtalálható a 74-es TTL sorozatban, típusjele 74S138, 74LS138.

Két évvel később az Intel a 8080A alkalmazások támogatására két új LS bipoláris áramkört fejlesztett ki, a 8224 óragenerátort és a 8228 rendszervezérlő/adatbusz-erősítő áramkört. Az egyik alapvető gondon, a 8080A három tápfeszültségén ezek az áramkörök nem segíthettek, de a többi hardver tervezési probléma megoldásához hathatósan hozzájárultak. A három IC-vel felépített CPU már nem marad el hatékonyságban a Z80-CPU képességeitől sem!

A 8080A esetében meg kell jegyezni, hogy a három tápfeszültség alkalmazása teljesen természetes volt, mivel 1974-ben a PMOS memória-áramkörök többsége ezt a három tápfeszültséget használta, tehát nyugodtan feltételezhetjük a 8080A-nál is, hogy a három tápfeszültség rendelkezésre áll. Érdekességképpen megemlíjtük, hogy a 8085A chip sem +5V tápfeszültséggel működik, de a felhasználók igényei kielégítése érdekében kívülről csak ezt a tápfeszültséget fogadja, az áramkör belsejében helyezett el az Intel feszültség-konvertereket.

A továbbiakban a 8080A-ra épülő mikroszámítógépekben a CPU többnyire ezt az IC-hármaszt használta (8080+8224+8228), a **6.1. ábra** szerint. Ezek az új áramkörök is bekerültek később a 74-es sorozatba (74LS424, 74LS428). Akár azt is mondhatjuk, hogy ezek a kiegészítő integrált áramkörök alkották az első Chip-Set készletet.



6.1. ábra

A 8224 óragenerátor IC, de sok további feladatot is ellát. A 8080A által igényelt, + 12 V-os órajel-szintek előállítására érdekében ez az áramkör a + 5 V-os tápfeszültségen kívül a + 12 V-os tápfeszültséget is megkapja, bár LS TTL IC. Ha a 8224-re a + 12 V-ot nem vezetjük rá, az oszcillátor áramköre nem is működik!

Az IC a következő funkciókat tölti be:

- Előállítja a +12V névleges szintű, kétfázisú órajel-sorozatot, ehhez egy rezgőkvarcot kell csatlakoztatni hozzá. Az órajel frekvenciája a kvarc frekvenciájának egykilenced része. A  $\phi 2$ -t TTL szinten is kiveztették az áramkörből. Szükség esetén az oszcillátor felharmónikus üzemmódban is működtethető, amikor a rezgőkvarc frekvenciája a szükséges frekvencia harmadrésze s az oszcillátort egy LC körrel hangolják a kvarcfrekvencia háromszorosára,
- a SYNC jelnél szűkebb kapuzójelet állít elő a státusz szó érzékeléséhez (Status Strobe,  $\overline{\text{STSTB}}$ ),
- alacsony szinten aktív, Schmitt-trigger bemenetű  $\overline{\text{RESIN}}$  bemeneti pontján kialakítható a bekapcsolási automatikus RESET és a kézi RESET (ld. a 6.1. ábrán),
- a READY jelet időzítve meggátolja, hogy akkor változzék, mikor a  $\mu\text{P}$  éppen beolvassa.

Az  $\overline{\text{STSTB}}$  jel azért szükséges, mert a 8080A nem megfelelően jelzi a státusz szó érvényességét. Mint már tudjuk, a gépi cikluson belül az első  $\phi 2$  felfutó élkor helyezi ki az adatbuszra, majd a második  $\phi 2$  felfutó élnél szünteti meg. A státusz szó jelenlétét a SYNC jel H szintje jelzi, de ez pontosan akkor jelenik meg, amikor a státusz szó, és azzal együtt szűnik meg. A státusz szót egy felfutó vagy egy lefutó éllel kellene tárolóba beírni, de erre a célra a SYNC élei nem alkalmasak, megjelenésükkor a státusz szó még nem, illetve már nem stabil az adatbuszon. Az  $\overline{\text{STSTB}}$  jel aktív L szintje a SYNC jel H szintjének megjelenése után alakul ki, és a lefutó éle a SYNC jel vége előtt már megjelenik. Így ennek a jelnek az éleit fel lehet használni a státusz szót befogadó tároló vezérlésére.

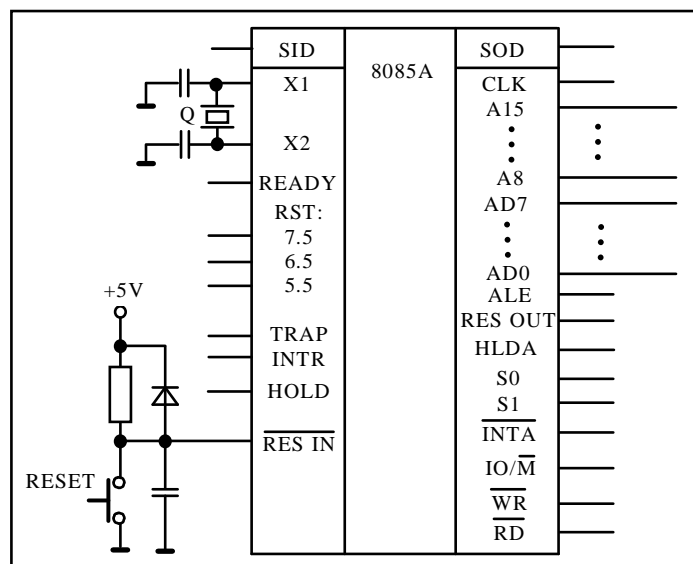
A 8228 rendszervezérlő és adatbuszkezelő áramkör a következő feladatokat látja el:

- a státusz szóból előállítja a vezérlőjeleket, a gépi ciklusig fenntartja azokat,
- kezeli a kétirányú adatbusz-t, elvégzi a külső TTL áramkörök és a 8080A között szükséges szintillesztéseket, a terhelhetőséget megvalósítja (névleges kimeneti árama 16 mA, a 8080A számára adott kimeneti H szintjének minimális értéke 3,8 V),
- a  $\overline{\text{BUSEN}}$  jel L szintjének hatására a rendszeroldali adatbuszt lebegteti,
- lebegteti a rendszeroldali adatbuszt, ha a  $\mu\text{P}$  HOLD állapotba kerül,
- támogatja a megszakításkezelést.

A 8080A megszakításkezelése, mint korábban láttuk, elég szegényes és nehézkes. Mint tudjuk, egy kényelmes megoldás az, ha a megszakításkérés hatására automatikusan szubrutint aktivizál a mikroprocesszor. A 8228 beépítése után ez lehetővé válik a 8080A-nál is! Ha a 8228  $\overline{\text{INTA}}$  kimenetét egy 1 kOhm-os ellenálláson át a +12V tápfeszültségre kötjük, elfogadott megszakítás után automatikusan beküld egy RST7-es utasítást a processzorhoz, azaz megtörténik a 38h kezdőcímű szubrutinra ugrás. Ha az RST7 opkódját megvizsgáljuk, kiderül, hogy az FFh, azaz a 8228 az  $\overline{\text{INTA}}$  gépi ciklusban tiszta 1-t vezet a mikroprocesszor adatbuszpontjaira.

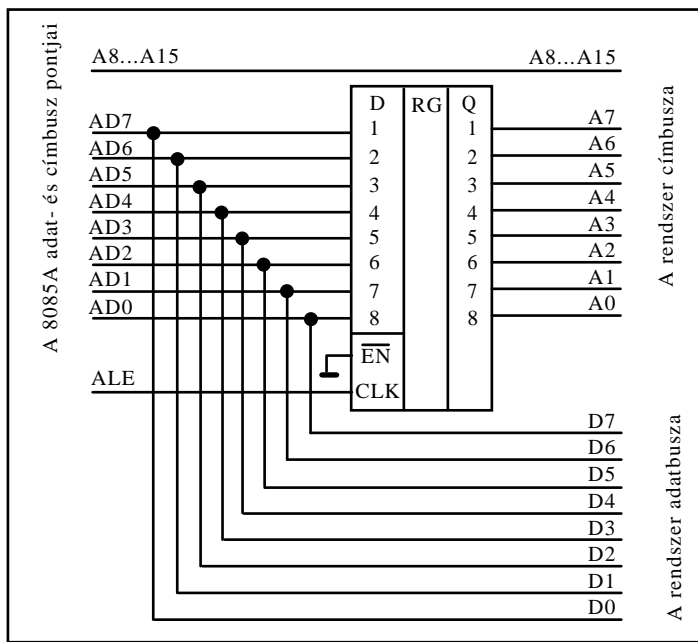
Ha több megszakításforrás is van a rendszerben, a 8228 lehetővé teszi, hogy mindegyik a saját rutinját tudja aktivizálni, egyszerű CALL szubrutinhívó utasítás használatával. Ha a 8228 érzékeli, hogy  $\overline{\text{INTA}}$  gépi ciklusban (rajta át) a külső hardver a CALL opkódját (CDh) küldte be a processzorhoz, a következő két gépi ciklusban (amikor a  $\mu\text{P}$  memória olvasást végezne) felülbírálja a 8080A státusz információját. Ebben a két gépi ciklusban a 8080A memória olvasás státusz szót küld ki, amiből a 8228-nak a  $\overline{\text{MEMR}}$  vezérlő jelet kell előállítania, azaz ennek a vezérlő jelnek kellene L szinten állnia a gépi ciklus folyamán. Ehelyett a 8228 a két gépi ciklusban az  $\overline{\text{INTA}}$  vezérlő jelet állítja ismét L szintre, azaz még két gépi ciklusban  $\overline{\text{INTA}}$  gépi ciklust jelez, ezt a külső hardver felhasználhatja a CALL-hoz tartozó 16 bites cím beküldésére. A 8080A programozható megszakításkezelő áramköre is ezt a lehetőséget használja ki, így tudja a megszakításkérésnek megfelelő szubrutint aktivizálni.

## 6.2. CPU kialakítása a 8085A mikroprocesszorral



6.2. ábra

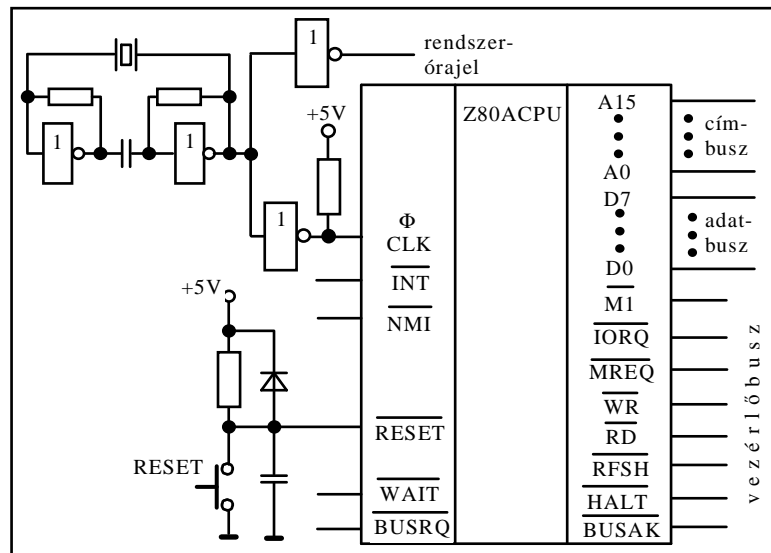
A 8085A gyakorlatilag komplett CPU (sőt, tulajdonképpen kicsit több is, már a mikrovezérlők egyik előfutára a sokoldalú megszakítási rendszerével és a soros bemeneti, kimeneti lehetőségével).



6.3. ábra

lunk, akkor külső tároló felhasználásával tárolni kell a címet, ami a multiplexelt buszon csak egy rövid időre tűnik fel a gépi ciklusok elején (az ALE jel utal a cím jelenlétére), a későbbiekben ezeken a vezetéseken az adatforgalom zajlik (6.3. ábra).

### 6.3. A Z80 CPU



6.4. ábra

A Z80-CPU szinte komplett központi egység (6.4. ábra). A +5 V szint kialakításához pl. az 5.16. ábra megoldásait lehet használni.

Az órajellel kapcsolatos másik igény, hogy a kitöltési tényezője 50 % legyen. Ennek érdekében az RC- vagy kvarc-oszcillátorok módosíthatása helyett gyakran azt az utat választjuk, hogy kétszeres frekvenciájú alaposzcillátort építünk és a kimenőjeleit egy T tárolóra vezetjük. A tároló kimenetén fele frekvenciájú, szabályos négyszög hullám fog megjelenni.



## 7. A memória és az I/O elemek kezelése a mikroszámítógépekben

A mikroszámítógépben a memória és az I/O elemek kezelésére a vezérlőjeleket és a címbuszt használjuk fel, az adatpontok az adatbuszra csatlakoznak.

Tudjuk, hogy a mikroszámítógép működése gépi ciklusokra tagolódik, minden gépi ciklusban egy adatmozgás zajlik le. A gépi ciklus jellege attól függ, milyen utasításciklusban, annak melyik fázisában vagyunk. Mivel a gépi ciklus egy külső elemtől adat fogadását illetve egy külső elemhez adat kivitelét jelenti, tehát külső elemet mindenképpen igénybe vesz, ezért mindig szerepet játszik benne egy cím, egy külső egység címe. A mikroprocesszorok gépi ciklusának első eseménye minden esetben ennek a címnek a kiküldése. A következő esemény a gépi ciklus folyamatában a vezérlőjelek kialakítása, megjelenése, ezek jelzik a gépi ciklus jellegét. A korábbiakban az egyes mikroprocesszoroknál mindig kitértünk arra, hogy a legfontosabb gépi ciklusokat milyen vezérlőjelekkel jelzi, kezeli az áramkör (memóriaolvasás, memóriaírás, I/O írás, I/O olvasás, megszakítás-elfogadás).

Az éppen végrehajtás alatt álló utasítás határozza meg, hogy milyen gépi ciklusokat hajt végre a mikroprocesszor, ami ennek megfelelően aktivizálja a vezérlőjeleit. Az azonban, hogy az aktivizált vezérlőjelek hogyan működtetik a külső elemeket, a mikroszámítógép további áramköreit, az már nem a mikroprocesszoron múlik.

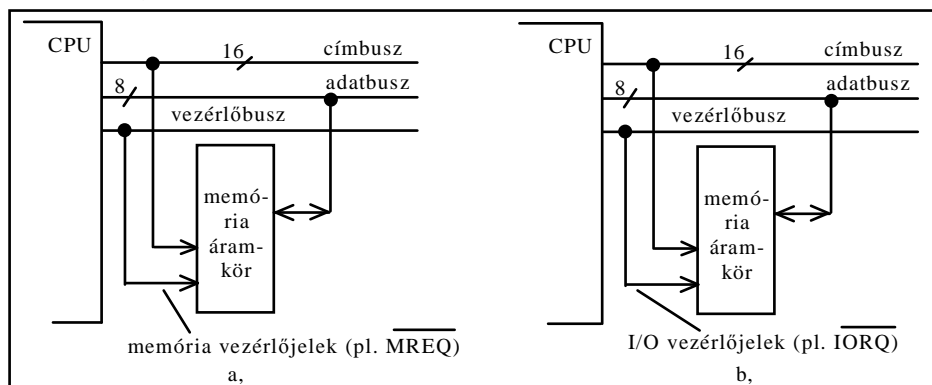
Emlékeztetünk a különféle mikroprocesszorok I/O és memóriavezérlő jeleire:

- a 8080A-ra épülő rendszerek:  $\overline{\text{MEMR}}$ ,  $\overline{\text{MEMW}}$ ,  $\overline{\text{I/O OR}}$ ,  $\overline{\text{I/O OW}}$ ,
- a 8085A esetén:  $\overline{\text{IO/M}}$ ,  $\overline{\text{RD}}$ ,  $\overline{\text{WR}}$ ,
- a Z80 esetében:  $\overline{\text{MREQ}}$ ,  $\overline{\text{IORQ}}$ ,  $\overline{\text{RD}}$ ,  $\overline{\text{WR}}$ .

Mindhárom  $\mu\text{P}$  16 bites címbusszal rendelkezik a memória címezéséhez, az I/O utasítások mindháromnál 8 bites címet kezelnek. Ha memória-referenciás működés zajlik (utasításelem beolvasás, veremkezelés, memóriára vonatkozó utasítás végrehajtása), a memóriát aktivizáló vezérlőjelek lesznek aktívak (a cím pedig 16 bites); I/O kezelő utasítások végrehajtása során az I/O aktivizáló vezérlő jelek hatnak (és csak nyolcbites I/O cím).

### 7.1. Memória kezelése a mikroszámítógépekben

A memória IC-eket a mikroszámítógépben (7.1. ábra) működtethetjük a memória vezérlőjelekkel (memóriába ágyazott memória) vagy az I/O vezérlőjelekkel (I/O-ba ágyazott memória illetve adatmemória).



7.1. ábra

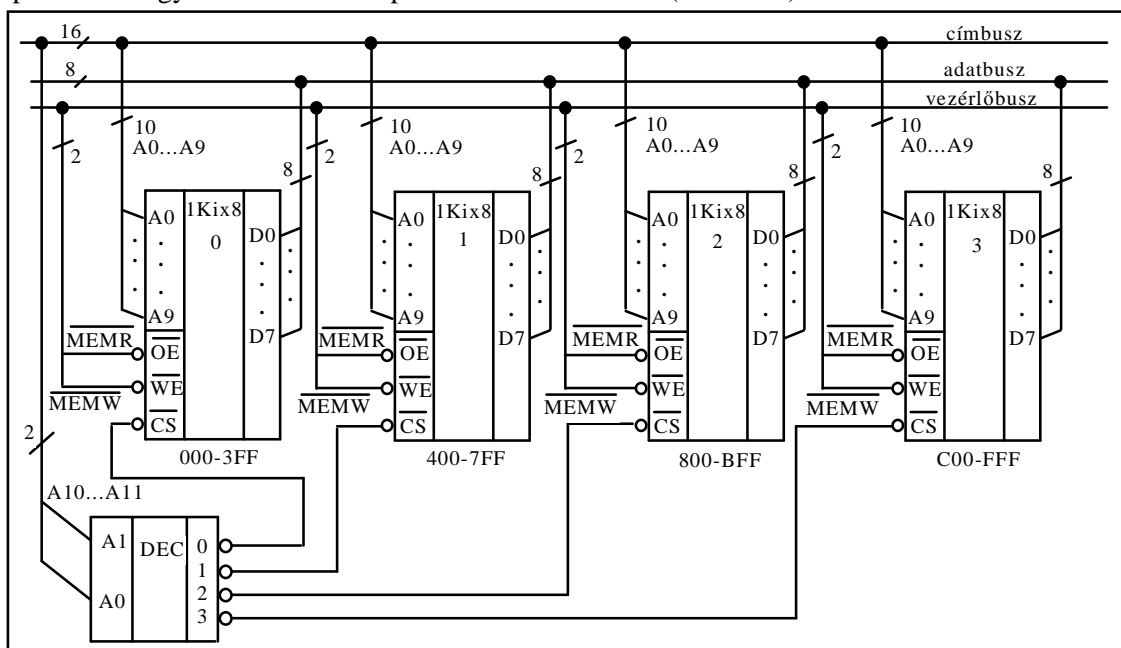
A memóriába ágyazott memóriát az automatizmusok használják (utasításelem beolvasás, veremkezelés), valamint a memória referenciás (memóriára hivatkozó) utasítások, 16 bites címmel.

A mikroprocesszorok utasításkészletében nagyszámú memória referenciás utasítás található. Általában a memóriarekeszek egyszerű olvasásán és írásán kívül műveletvégzési lehetőségekkel is találkozunk (az aritmetikai és logikai műveletek egyik operandusa lehet a külső memóriában). Memóriába ágyazott memóriát mindig ki kell alakítani a mikrogépben, mert az automatikus működések csak ezzel valósulhatnak meg. Ha a  $\mu P$ -nek vannak I/O kezelő utasításai, az I/O-ba ágyazott memória kiépíthető, de ezt csak az I/O utasítások fogják elérni.

Mivel az I/O-ba ágyazott memória-áramkört csak az I/O utasításokkal lehet olvasni vagy írni, a címzésük is szűkebb lehetőségeket nyújt, csak az I/O utasítások által kezelt címmel címezhetők, ami a megismert processzoroknál 8 bites. Az I/O utasítások száma általában jóval kisebb, mint a memória referenciásaké, a működési módok is egyszerűbbek. A 8080A mikroprocesszornak pl. egyetlen bemeneti (IN N) és egyetlen kimeneti (OUT N) utasítása van! Ez a két utasítás is rendkívül kötött működésű, az I/O eszköz és az akkumulátor közötti adatcserét valósítja meg.

Akár memória-, akár I/O kezelő gépi ciklust hajt végre a  $\mu P$ , a processzor látókörében csak a buszvezetékek vannak. Mit értünk ez alatt? Ha egy I/O írás gépi ciklusban vagyunk, a mikroprocesszor a címbuszra beállítja az I/O elem címét, az adatbuszra kihelyezi az akkumulátor tartalmát, a vezérlőjelekkel az I/O írás igényt jelzi. Ennél többet a processzor nem tesz és nem is tehet. Ha a gépi ciklus I/O olvasás, a cím kiküldése és az I/O olvasás vezérlőjeleinek előállítása után az adatbusz tartalmát beolvassa az akkumulátorba. Azt, hogy a mikroszámítógép CPU-n kívüli részén mi történik, hol lesz felhasználva a kiküldött cím, milyen elem reagál a vezérlőjelekre, hova kerül a kiírt adat vagy honnan származik a beolvasott, a mikroprocesszor nem tudja befolyásolni. Ahhoz, hogy valóban megtörténjen egy memória vagy I/O olvasás vagy írás, a külső hardver megfelelő megtervezése, összeállítása szükséges.

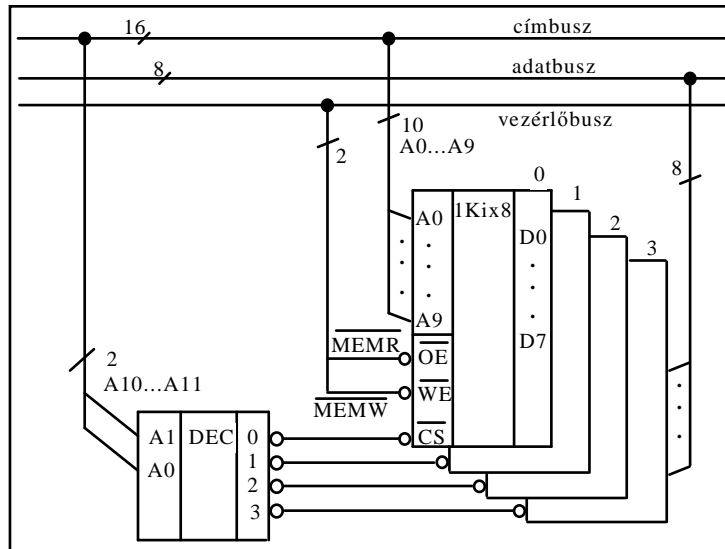
A címkezelés megértéséhez tételezzünk fel egy 8080A-ra épülő mikroszámítógépet s abban egy memóriába ágyazott, 4Ki x 8 kapacitású SRAM memóriablokkot. Ezt a memóriablokkot építsük fel négy darab 1Ki x 8 kapacitású SRAM IC-ből (7.2. ábra).



7.2. ábra.

Minden memória-áramkör saját belső címzőpontjai a címbusz ugyanilyen sorszámú vezetéksorozatára kapcsolódik (itt az A0 .... A9, tíz vezeték). Általában a RAM jellegű memóriáknál megengedhető, hogy a címbusz és a memória címbemenetek között ne tartsuk be a címbitek sorrendjét, de ha a memóriának tíz címvezetéke van, az a címbusz alsó 10 vezetékével lesz összekapcsolva.

Megtehetjük (ha pl. a NYÁK mintázat így egyszerűsödik), hogy a címbusz A0 vezetéket pl. a memória A1 pontjára kötjük, míg a címbusz A1 vezetéke a memória A0 pontjára csatlakozhat. A RAM jellegű memóriáknál az adatpontoknál is megengedett az ilyen csere. Más a helyzet a ROM típusú áramköröknél, azoknál a memória IC-kenél, melyek tartalmát nem a mikroprocesszor állítja elő, a mikroszámítógép működése közben. A ROM áramkörök katalógus szerinti cím- és adatbit sorszámozását mereven figyelembe kell venni és a buszok ugyanolyan sorszámu pontjaival kell összekötnünk mindegyiket.



7.3. ábra

A kitűzött áramköri elképzelésnél a négy memória-áramkör egyikének kiválasztása a következő két címbit felhasználásával oldható meg, az A10 ... A11 összesen négy lehetséges érték-kombinációját egy címdekóder felhasználásával lehet dekódolni és a

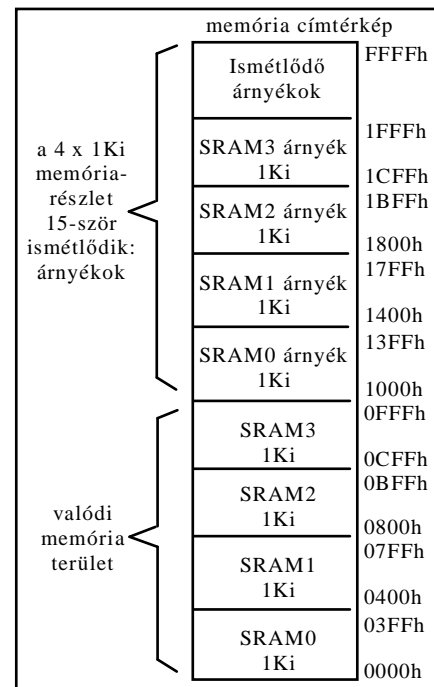
memória címtérkép	
?	FFFFh
	1000h
SRAM3 1Ki	0FFFh
SRAM2 1Ki	0CFFh
SRAM1 1Ki	0BFFh
SRAM0 1Ki	0800h
	07FFh
	0400h
	03FFh
	0000h

7.4. ábra

vagy tiltják.

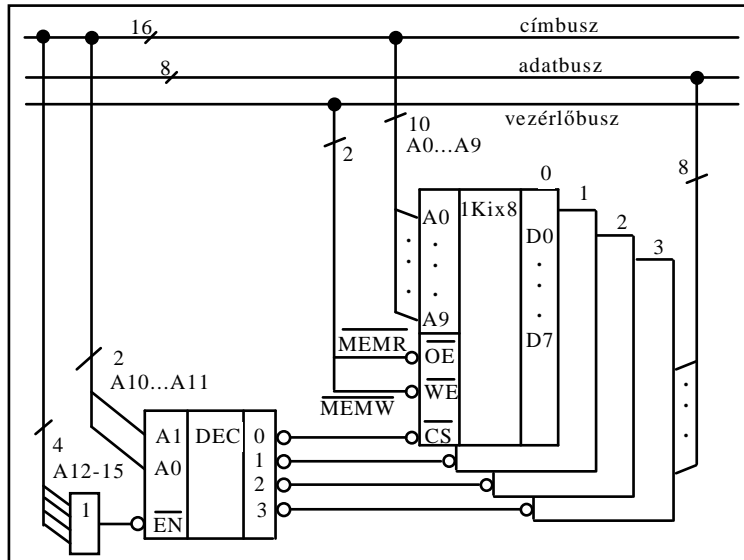
A 7.2. ábrán látható elrendezést egyszerűbben is ábrázolhatjuk, a 7.3. ábra megoldását alkalmazva. A takarásban lévő áramkörök egyformák, az elől lévővel azonos a bekötésük, kivéve azt a részletet, amit minden elemnél megrajzolunk: most a  $\overline{CS}$  jelek ilyenek.

A memória címtérkép a 16 bites címnek megfelelően 64 Ki területet ábrázol (0000h ... FFFFh, 7.4. ábra).



7.5. ábra

Az ábrába már berajzoltuk a négy memória IC által elfoglalt címterületet, azok kezdő- és végértékével együtt. Rövid szemléltetés után kiderül, hogy a magasabb címtartományon ez a négy memória IC újra és újra aktivizálódik, hiszen a felsőbb címbitek értéke nem befolyásolja a működésüket. A 0 sorszámú memória áramkört eszerint elérheti a mikroprocesszor a 0000 ... 03FF (hexadecimális) címtartományban is, de az 1000 ... 13FF, a 2000 ... 23FF stb. címtartományokban is. A **7.2. ábrán** szereplő áramkörök számára ugyanis a 0000h cím teljesen azonos azonos az 1000h címmel, a 2000h címmel.



7.6. ábra

A **7.4. ábrán** még kérdőjellel megjelölt terület tehát nem üres, a 0000h ... 0FFFh tartományban valódi memóriaként megjelenő elemek árnyékai befedik a teljes felső területet, ahogyan erre a **7.5. ábra** utal. Az árnyék oka az, hogy vannak olyan magasabb címbitek, melyek értékétől függetlenül aktivizálódnak a memória elemeink, azaz a címdekódolás nem teljes, csak részleges.

Az árnyékok a mikroszámítógép működésében általában nem okoznak zavart, de

pl. a memória tesztelésekor, a memóriakapacitás automatikus felmérésekor már problémát jelenthetnek. Az igazi gondot akkor jelenti a részleges címdekódolás, a memória-árnyékok jelenléte, ha bővíteni szeretnénk a mikroszámítógép memóriáját, s nem tehetjük, mert a címtérképet az árnyékok lefedik. Ha a **7.2. ábra** szerinti a mikroszámítógép memóriája és újabb memória áramköröket kell beépíteni, ebben a pillanatban ez nem oldható meg, mert nincs olyan címterület, ahol valamelyik már beépített memória áramkör ne aktivizálna.

A **7.6. ábrán** már árnyékmentessé tettük a memóriánkat. A címdekóder működését, és így a memória-áramköröket is csak az A12 ... A15 0 értékei mellett engedélyeztük. Ezáltal a legalsó megjelenésen kívül a címtérképről minden további eltűnt, a teljes címdekódolás árnyékmentessé tette a memóriát (**7.7. ábra**). Természetesen a felsőbb címbitek más kombinációját is felhasználhatjuk a memóriaterület aktivizálására, akkor a memória egységünk a 64 Ki területen belül egy magasabb címtartományban fog megjelenni a címtérképen.

memória címtérkép	
üres	FFFFh
	1000h
SRAM3 1Ki	0FFFh
	0CFFh
SRAM2 1Ki	0BFFh
	0800h
SRAM1 1Ki	07FFh
	0400h
SRAM0 1Ki	03FFh
	0000h

7.7. ábra

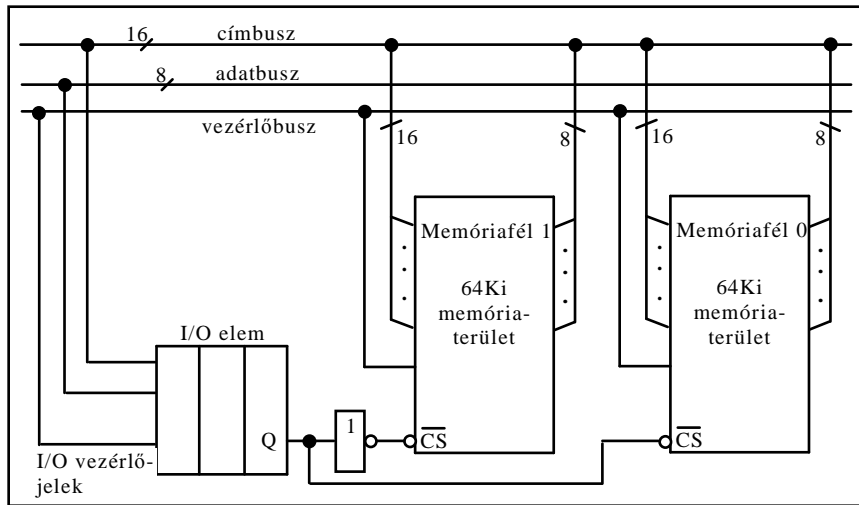
Egy lényeges megjegyzés: ha egy memória IC-t vagy egy kész blokkot kell behelyeznünk a lehetséges címterületre, mindig csak olyan területre helyezzük, amit a teljes terület és az adott memória kapacitása elosztásával ki lehet jelölni. Ha tehát a lehetséges memóriaterület 64 Ki és 8 Ki méretű memória IC-t kell elhelyeznünk, a 64 Ki-t elosztjuk a 8 Ki mérettel és az így kialakuló 8 helyre lehet az IC-t betenni (azaz a 64 Ki-n belül a legalsó 8 Ki-ra, vagy a következőre stb.). Úgy ne tegyük be a memóriát, hogy a fenti határok a memória belsejébe kerüljenek, mivel a címzés megoldása rendkívül bonyolulttá

válhat.

Ha a 64 Ki memóriaterület szűknek bizonyul, kiegészítő címbiteket lehet kialakítani. Megtehetjük, hogy egy kimeneti elem, kimeneti Port egy bitjét egy új, 17. címvezetékre vezetjük, a

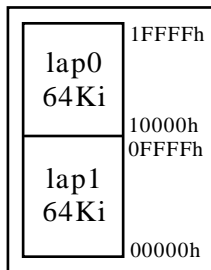
bitet I/O utasításokkal kezeljük, és így két darab 64 Ki méretű memóriaterületet felváltva tudunk használni (7.8. ábra).

Hasznos lehet ez a megoldás, ha a mikroszámítógép egyszerre csak az egyik 64 Ki területet használja. A memória címtérképében az esetben már 128 Ki méretű, amin két 64 Ki kapacitású memória egység osztozik (7.9. ábra). A bemutatott esetben egy adott pillanatban a CPU a teljes memóriát nem tudja



7.8. ábra

kezelni, csak annak egy részét. Ha az egyidejűleg kezelhető részek egymás mellé helyeződve fedik le a teljes memória tartományt, lapozásról beszélünk (itt tehát két memória lapunk van, a 0 és az 1 sorszámú).

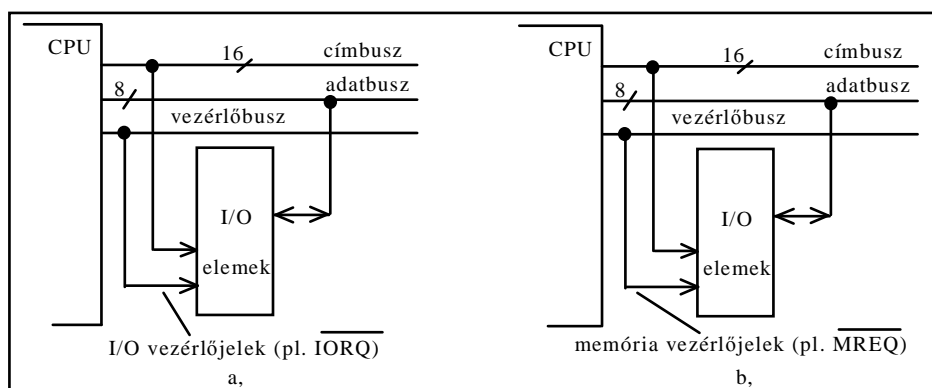


7.9. ábra

Ha az egyszerre kezelhető memóriarészek egymást fedve helyezkednek el a tényleges memóriaterületen, szegmentálásról beszélünk, erre látunk majd példát a 16 bites Intel mikroprocesszorok memóriakezelésénél. Akár lapozós, akár szegmentált a memóriakezelés, ha a szoftver kezelése közben át kell lépni a határokat, lapot illetve szegmenset kell váltani, az már komoly felügyeleti megoldásokat, operációs rendszert igényel.

Az I/O-ba ágyazott memóriákat a I/O címtérképen ábrázoljuk, ami az I/O utasításban szereplő címnek megfelelő méretű, a megismert 8 bites mikroprocesszoroknál csak 256 Byte (00h ... FFh).

## 7.2. I/O elemek kezelése a mikroszámítógépben



7.10. ábra

A bemeneti/kimeneti (I/O) elemeket a mikroszámítógépben (7.10. ábra) ráköthetjük akár a memória vezérlőjelekre (memóriába ágyazott I/O), vagy az I/O vezérlőjelekre (I/O-ba ágyazott I/O illetve utasításos I/O kezelés).

A memóriába ágyazott I/O elemek a sokoldalú memória-referenciás utasításokkal kezelhetők, az I/O-ba ágyazott I/O csak az I/O utasításokkal írható, olvasható. A két esetben a használható címtartomány is jelentősen eltérhet, a 8 bites mikroprocesszorok esetében a memóriába ágyazott I/O-t a 16 bites címekkel lehet kezelni, az I/O-ba ágyazottakat a 8 bitesekkel.

A memóriába ágyazott I/O elemeket a memória címtérképen kell feltüntetni, az I/O-ba ágyazottakat az I/O címtérképen. Sajátos jelenség, hogy azonos címen elhelyezhető egy Input egység és egy Output egység, hiszen egyszerre soha nem lesznek aktivizálva. A párosítási lehetőségeket tovább vizsgálva még azt is felfedezhetjük, hogy azonos címtérületre helyezhetők ROM jellegű memória áramkörök és kimeneti áramkörök, ezek sem zavarják egymást.

Itt célszerű megjegyezni, hogy sajnos, helytelen címmel az áramkörök tönkre is tehető. Ha azonos címen aktivizálódik pl. egy ROM és egy RAM áramkör, a címtartományon belüli memória olvasáskor mindkettő kimeneti egysége az adatbuszra helyezi a megcímzett rekesztartalmát, buszütközés jön létre minden olyan bitnél, amelyiken a két áramkör tartalma eltér egymástól. Ha az egyiknek nagyobb a kimeneti terhelhetősége, annak a kimenő jelértéke érvényesül, s ez a jel a másik kimeneti fokozatot tönkreteheti. Adódhat olyan címelrendezési helyzet is, hogy pl. a memória-áramkörök valódi megjelenése a címtérképen szabályos, nincs káros átfedés, de valamelyik árnyéktartománynál már létrejön az egyidejű káros aktivizálódás. Ez is egy indok lehet arra, hogy lehetőleg teljes címdekódolást alkalmazzunk. Ugyanilyen problémára vezet, ha azonos címen aktivizálódik egy I/O egység és egy RAM memória; vagy egy bemeneti egység és egy ROM áramkör.

A memóriába ágyazott I/O esetében igen gyakori a részleges címdekódolás. Sokszor azt a legegyszerűbb tervezői fogást alkalmazzák, hogy az A15 címvezeték bitértéke választja szét a memória áramköröket az I/O elemektől, pl. az  $A15 = 0$  a memóriaterületet működteti, míg az  $A15 = 1$  az I/O elemeket. Ezt valóban nagyon egyszerűen meg lehet oldani, de így néhány I/O elem miatt 32 Ki memóriaterülettel lettünk szegényebbek! Ha a memóriába ágyazott I/O elemeknél teljes címdekódolást használunk, akkor csak azt a néhány címet foglalják le, amit feltétlenül szükséges, viszont így 16 címbit teljes dekódolását kell elvégezni!

Ejtsünk néhány szót arról, hogy a mikroszámítógépben milyen áramköri elemekkel lehet kivitelezni az I/O elemeket, a bemeneti és kimeneti Port-okat.

A kimeneti (Output) elem, kimeneti Port aktivizálásakor az adatbuszról leolvassa az adatot és megjeleníti a kimeneti pontjain. Az aktivizálás után azonban az adat a kimeneti pontjairól nem tűnhet el, hiszen a kimeneti Port aktivizálása esetenként csak néhány ns-okig tart. Mivel a mikroszámítógép kimenetére csatlakozó elemek számára nem elegendő ilyen rövid időre felvillantani a kimenő adatot, a kimeneti elemben mindig van tároló. A kimenet aktivizálásakor az adatbusz adatát átveszi, a kimenő pontjain megjeleníti és addig ott tartja, míg egy újabb kimenet-aktivizálás során az új adat az előzőt felül nem írja.

A bemeneti elem (Input), bemeneti Port a külvilágból fogad adatot és azt, ha aktivizáljuk, az adatbuszra helyezi. Ehhez nem szükséges tárolónak lennie benne, hiszen az aktivizálásakor egyszerűen átvezetővé, átlátszóvá tehető. Sokszor mégis a bemeneti elemekben is alkalmaznak tárolót, amit az aktivizálás előtt leválasztanak a bemeneti pontokról, így a  $\mu P$  stabil, véletlenül sem változó jeleket lát az adatbuszon. Amikor a bemeneti elem nincs aktivizálva, akkor viszont az adatbusz vezetékeit nem zavarhatja, ezért az adatbuszra csatlakozó oldalon a bemeneti egységnek háromállapotú kimenetekkel kell rendelkeznie, amiket kijelöletlen állapotban lebegtetni tud.

Sok mikroprocesszor utasításkészletéből hiányzanak az I/O utasítások, a vezérlő jelek között ezért nincsenek I/O vezérlő jelek sem. Ezeknél a memória teljes terjedelmét is és az I/O eszközöket is memóriába kell ágyazni. Ilyen  $\mu P$  pl. a MOTOROLA MC6800, az RCA 6500.

## 8. A mikroprocesszorok programozható kiegészítő egységei

A mikroszámítógépekben számos részfeladatot programozható kiegészítő elemek oldanak meg. A rájuk bízott részfeladatok a mikroprocesszor szempontjából rutinfeladatok, melyek – ha közvetlenül a  $\mu P$  végzi el azokat – feleslegesen kötik le a processzor idejét. Sok programozható kiegészítő áramkör maga is mikroprocesszor bonyolultságú, tulajdonképpen egy-egy cél-mikroprocesszor. A mikroprocesszor (programja útmutatásai alapján) felprogramozza ezeket a kiegészítő áramköröket, s azok addig látják el a megkívánt feladatot, míg másikat nem kér tőlük a CPU, parancsszó tárolóik átírása útján.

A következő alfejezetekben néhány egyszerűbb programozható kiegészítőt részletesen bemutatunk, így az alpműveleteket és az alkalmazási fogásokat megismerjük. Ezek birtokában az összetettebb egységek működését, programozását is meg lehet majd érteni. A párhuzamos illesztő, a soros illesztő és az időzítő/számláló a legtöbbször alkalmazott programozható kiegészítő elem. Azért is érdemes velük részletesen megismerkedni, mert a mikrovezérlőkben is ezek a leggyakoribb belső programozható elemek, s ott a felépítésük, kezelésük majdnem tökéletesen megegyezik azzal, amit itt megismerünk.

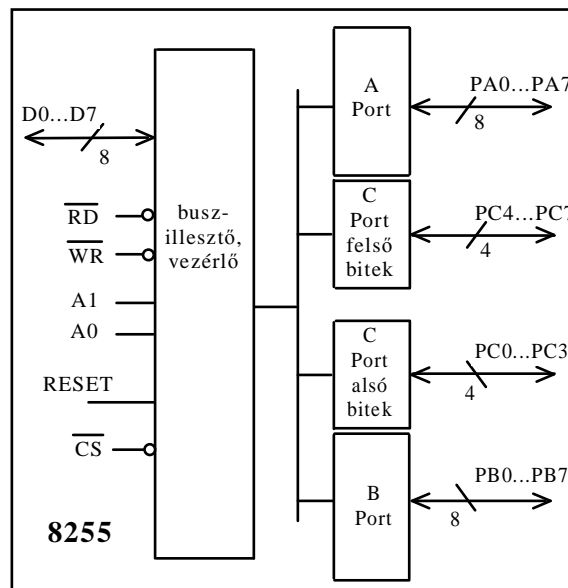
A programozható kiegészítő elemek az LSI áramkörök felosztásánál használt kifejezés szerint szoft-programozható egységek. Egy vagy több parancsszó tároló van bennük, az abban elhelyezett bitminta határozza meg a pillanatnyi működési módot. Ha más viselkedésre van szükség, a megfelelő parancsbiteket kell megváltoztatni. Általában a bekapcsolási RESET után induló programrész feladata, hogy a programozható kiegészítőket az alkalmazásnak megfelelően beprogramozza (inicializálás), a későbbiekben már csak az éppen aktuális kisebb átprogramozási vagy vezérlési feladatokat kell ellátni.

Egyes programozható kiegészítő egységekben 8-10 olyan belső elem is lehet, amit írás vagy olvasás jelleggel a mikroprocesszornak el kell érnie. A belső regiszterek kiválasztására több megoldást is alkalmaznak az áramköröknél, a három legelterjedtebb a következő:

- címző csatlakozó pontok használata (A0, A1 stb.) hasonlóan, mint a memória-áramköröknél; ezt a megoldást azért nem szívesen alkalmazzák a gyártók, mert a programozható kiegészítő áramköröknél általában kivezetés szűkében szoktak lenni, ha még a belső elemek címzése is lefoglal IC lábakat, az sok esetben azt jelentené, hogy az áramkör működése válik megoldhatatlanná
- a beírt adaton belül egyes biteket felhasználva (tartalom alapján történő címzés), a mikroprocesszor azonos vezérlőjel kombinációval kezeli a belső elemeket, a beküldött bitsorozatból egy vagy több bit értéke határozza meg, hova kell az adatnak kerülnie; ez a megoldás nem igényel további IC lábakat, viszont a beküldött adatszó 8 helyett csak kevesebb valóban funkcionáló vezérlőbitet tartalmaz
- a hozzáfordulási sorrend alapján történik a kiválasztás (pl. a RESET utáni első íráskor az R0, a másodikonál az R1 stb. kapja az adatot); ez a megoldás sem külön IC lábakat nem köt le, sem az adatszó hasznos bitjei számát nem csökkenti - viszont nehezebben követhető, az áramkör felprogramozása és későbbi kezelése nagyobb figyelmet kíván a felhasználótól.

### 8.1. A 8255 párhuzamos periféria illesztő

A 8255 a 8080A-hoz kifejlesztett programozható kiegészítő áramkör, máig is népszerű mikroszámítógép kiegészítő elem. 40 kivezetéses DIL tokozású áramkör. 24 programozható I/O pontot kezel, melyeket három 8 bites Port-ba rendeztek (A, B és C Port). A 8255 tömbvázlatát a **8.1. ábrán** láthatjuk. Az A1, A0 címző bemenetek a Port-ok egyikét vagy a vezérlőregisztert választják ki, a táblázat szerint:



8.1. ábra

A1	A0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	működés
0	0	L	H	L	A Port $\rightarrow$ D0 ... D7
0	1	L	H	L	B Port $\rightarrow$ D0 ... D7
1	0	L	H	L	C Port $\rightarrow$ D0 ... D7
0	0	H	L	L	D0 ... D7 $\rightarrow$ A Port
0	1	H	L	L	D0 ... D7 $\rightarrow$ B Port
1	0	H	L	L	D0 ... D7 $\rightarrow$ C Port
1	1	H	L	L	D0 ... D7 $\rightarrow$ vezérlőregiszter
1	1	L	H	L	nincs értelmezve
X	X	X	X	H	nincs kijelölve

Egy Port írás a kimeneti tárolót tölti fel, a Port olvasás vagy a Port külső csatlakozóit olvassa be (0-s mód), vagy a bemeneti puffert (1-es illetve 2-es mód). A vezérlőregiszter nem olvasható, csak írásra érhető el. A RESET jel a 8255-öt alapállapotba viszi, a Port pontok mindegyikét bemeneti állapotba vezérli. A 8255 párhuzamos Port-jai valódi kétirányú Port-ok, az adatirányt külön vezérlőbitek segítségével lehet beállítani.

A lehetséges üzemmódok:

- MODE0** vezérlőjelek nélküli kapcsolat a 8255 és a periféria között, csak a 8 bites adatok mozognak közöttük (bármelyik Port); a 8255 és a periféria nem tudják nyomon követni, hogy a másik fél feldolgozta-e a hozzá küldött adatot illetve kíván-e új adatot átadni, átvenni,
- MODE1** hand-shaking vezérlőjeles kapcsolat a 8255 és a periféria között, megszakításos visszajelzés a mikroprocesszorhoz (a C Port egyes bitjei szolgálnak vezérlőjelekként); az A és a B Port használható 1-es módban; ez a mód már biztosítja a 8255 és a periféria közötti intelligens kapcsolatot, az adó oldal tudni fogja, ha a vevő oldal fogadókész, és azt is, ha a vett adatot már feldolgozta; továbbá a 8255 és a mikroprocesszor között megszakításos kapcsolat alakítható ki, így a mikrogépen belül is hatékonyan oldható meg az adatkezelés,
- MODE2** kétirányú, hand-shaking vezérlőjelekkel működő kapcsolat a 8255 és a periféria között, a Port adatirányát a periféria vezérli; a mikroprocesszorral megszakításos visszajelzéses kapcsolat; csak az A Port működhet 2-es módban.



Ez az üzemmód lényegében megegyezik az 1-es móddal, de egyszerre mind a bemeneti mind a kimeneti adatirány felépül; azt, hogy pillanatnyilag az A Port bemenetként vagy kimenetként működjön-e, a periféria fogja vezérelni.

MSB	8255 üzemmód beállító szó						LSB
1	AM1	AM0	DA	DCH	MB	DB	DCL

D7 = 1 : az üzemmód beállító szó jele  
AM1,AM0 : az A Port üzemmódja: 00 = Mode 0  
01 = Mode 1  
1X = Mode 2  
DA : az A Port adatiránya, 1 = bemenet, 0 = kimenet  
DCH : a C Port felső bitjei adatiránya, 1 = bemenet, 0 = kimenet  
MB : a B Port üzemmódja: 0 = Mode 0  
1 = Mode 1  
DB : a B Port adatiránya, 1 = bemenet, 0 = kimenet  
DCL : a C Port alsó bitjei adatiránya, 1 = bemenet, 0 = kimenet

8.2. ábra

ható, hogy a C Port két 4 bites részletére külön lehet üzemmódot előírni, mert az egyik fele az A Port-hoz, a másik fele a B Port-hoz van kapcsolva.

A C Port bit beállító parancs értelmezését a 8.3. ábra segíti. A C Port bit beállító parancs akkor szükséges, ha az A vagy a B Port nem MODE0 üzemmódban dolgozik. Ekkor a C Port egyes bitjei szolgálnak vezérlő jelekként. Ha ilyenkor a C Port-ot olvassuk, az nem okoz zavart, de ha nyolcbites Portként kimenőadatot írunk bele, az a vezérlőjel funkciójú lábak állapotát is változtatja. Ilyenkor a vezérlő funkciójú lábak átírása elkerülhető, ha csak a fennmaradó biteket használjuk kimenetként s azok logikai értékét bitenként, ezzel a paranccsal állítjuk be.

A MODE0 beállítása után, ha a  $\mu P$  egy Port-ra ír, az adat a kimeneti pufferébe kerül (ha kimeneti üzemmódra van állítva, a csatlakozó pontjain is megjelenik); ha a processzor egy Port-ot olvas, a 8255 Port csatlakozópontjainak logikai állapota kerül be az adatbuszra. MODE0 esetben tehát a kimenet pufferelt, a bemenet pufferetlen, nincsenek a 8255 és a periféria között működő vezérlő jelek. A MODE1 és a MODE2 esetében a bemenet is pufferelt.

MODE1 bemenetként a következő vezérlő jeleket kezeli a 8255:

STB	Strobe, adatérvényesség jel, a periféria ezzel jelzi, ha a Port pontokra adatot helyezett,
IBF	Input Buffer Full, a bemeneti puffer betöltve, a 8255 ezzel jelzi a perifériának, hogy az adatot beolvasta bemeneti pufferébe,
INTR	megszakításkérő jel, a 8255 megszakításkéréssel jelzi a $\mu P$ -nek, hogy új adat van a bemeneti pufferében,
INTE	belső flip/flop a 8255-ben, 1 értéke engedélyezi a megszakítás-kérő jel kiküldését.

A 8.4. ábráról lehet leolvasni, hogy az A Port MODE1 bemenet esetén melyik C Port biteket használja fel vezérlőjelekként, a zárójeles értékek a B Port MODE1 bemenet üzemmódjára vonatkoznak. A megszakítás-tiltás illetve megszakítás-engedélyezés a Port C4 (illetve 2) bitjének bit állító paranccsal történő beírásával vagy törlésével végezhető el.

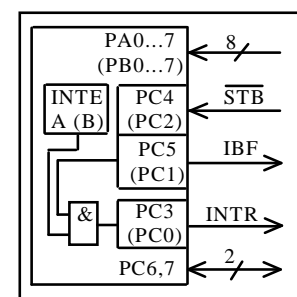
A vezérlőregiszterbe beírt parancsszó legfelső bitje (D7) azt mutatja meg, hogy üzemmód beállító szóként (D7 = 1) vagy C Port bit beállító parancsként (D7 = 0) kell-e felhasználni. (Az eddigiek alapján látható, hogy a 8255-nél a belső részletek kijelölésére címző lábakat is felhasználtak, valamint a beküldött adat tartalma alapján is történik elágasztás.)

Az üzemmód beállító szót a 8.2. ábra mutatja be, ez határozza meg hosszabb távra a 8255 tennivalóit. Lát-

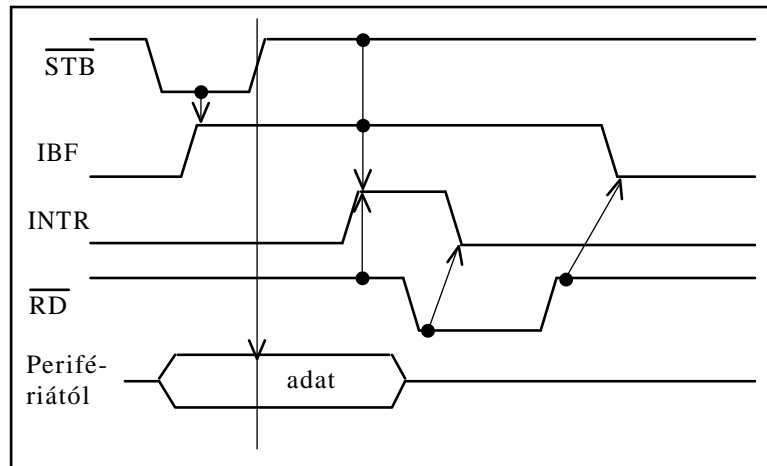
MSB	8255 C Port bit beállító szó						LSB
0	X	X	X	b2	b1	b0	S/R

D7 = 0 : a C Port bit beállító szó jele  
X,X,X : definiálatlan, javasolt érték: 0  
b2,b1,b0 : a kezelt bit sorszáma  
S/R : bit értéke, 0 ill. 1

8.3. ábra



8.4. ábra

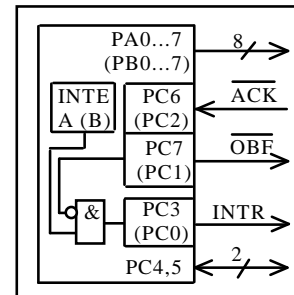


8.5. ábra

A 8.5. ábrán egy Port MODE1 működésének idődiagramja szerepel, az ábrán az is látható, hogy az egyes jelek megjelenése, eltűnése milyen szerepet játszik a folyamatban, hogyan következnek egymás után a jelváltozások.

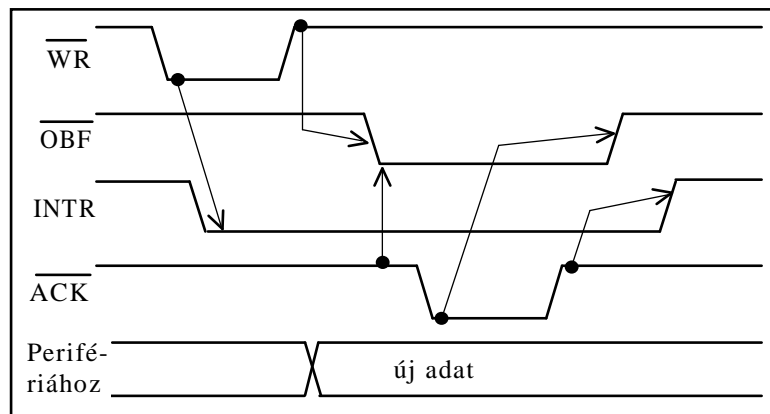
MODE1 kimenetként a 8255 a következő vezérlőjeleket használja:

- $\overline{\text{OBF}}$  Output Buffer Full, a kimeneti puffer betöltve; a 8255 ezzel jelzi a perifériának, hogy új adat érkezett a kimeneti pufferbe, egyidejűleg a Port csatlakozó pontokra is,
- $\overline{\text{ACK}}$  Acknowledge, nyugtázás; a periféria ezzel jelzi a 8255-nek, hogy az új adatot le olvasta a csatlakozó pontokról,
- $\text{INTR}$  a 8255 megszakításkéréssel jelzi a  $\mu\text{P}$ -nek, hogy kiolvasták a kimenő adatát,
- $\text{INTE}$  belső flip/flop a 8255-ben, 1 értéke engedélyezi a megszakításkérő jel kiküldését.



8.6. ábra

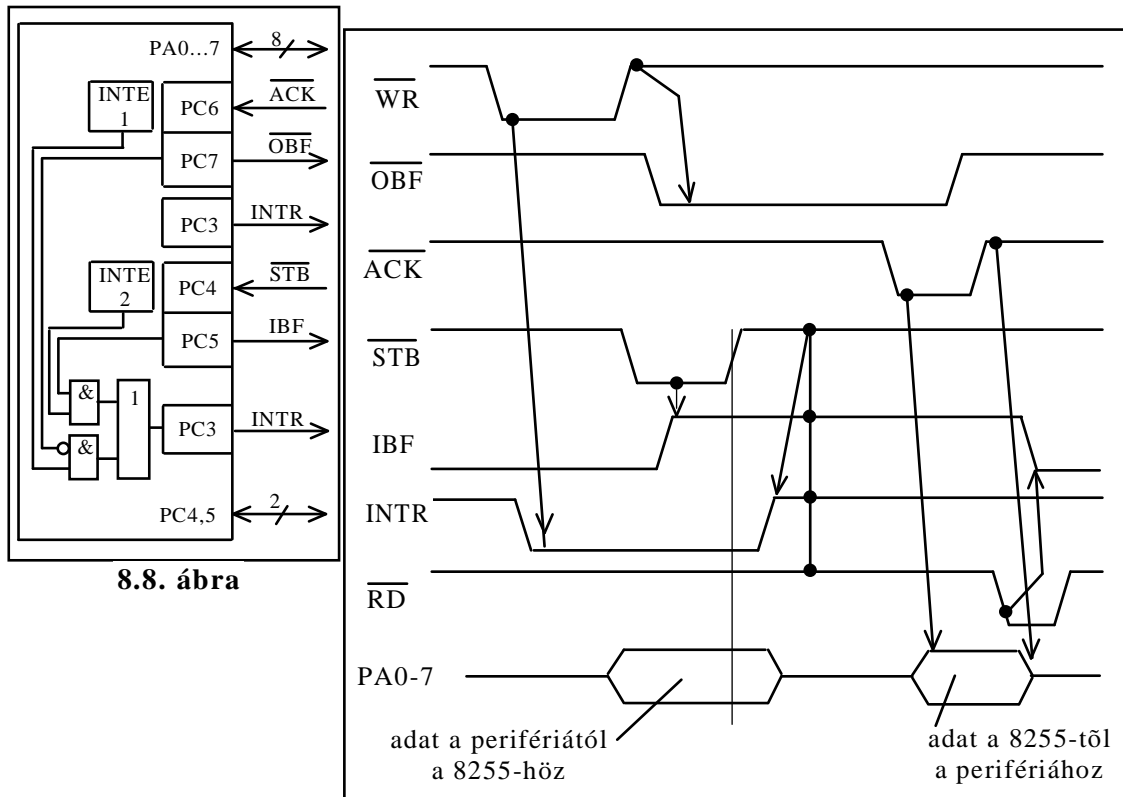
A 8.6. ábrán az A Port MODE1 kimenet vezérlőjeleit lehet megtalálni, zárójellezve pedig a B Port jelei láthatók. A 8.7. ábrán a MODE1 kimenet működésének időbeli alakulása követhető.



8.7. ábra

A MODE2 üzemmód csak az A Port-ra írható elő, ebben az üzemmódban a jelek a 8.8. ábra szerint alakulnak. Megszakítás-kérés csak egy van, így a megszakítást kezelő rutinnak kell megállapítania, hogy a kimeneti vagy a bemeneti irány kezdeményezte-e a megszakítást. A MODE2 kétirányú Port működés, ahol az adatirányt a periféria vezérli.

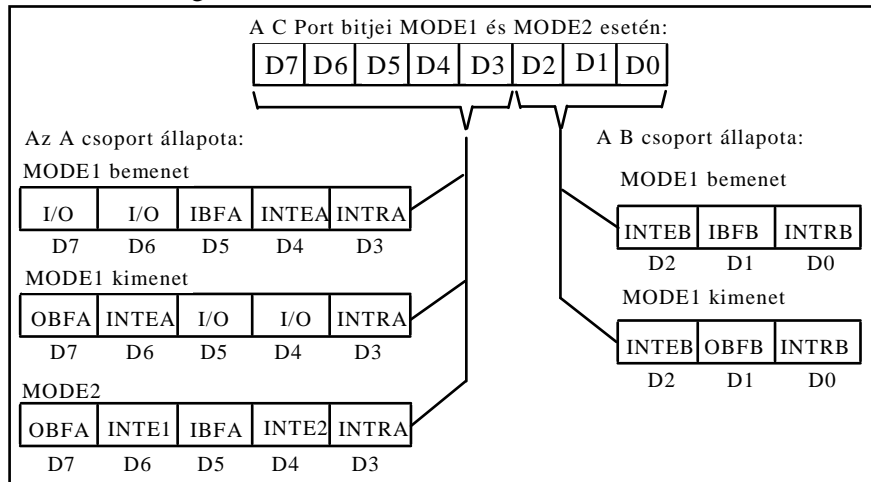
Alapállapotban az A Port bemeneti működésű, kimeneti üzemre a perifériától érkező ACK jel L szintje kapcsolja át (8.9. ábra). Végezetül a 8.10. ábrán azt láthatjuk, hogy különféle üzemmódokban a C Port olvasása milyen információkat ad a 8255 és vezérlőjelei állapotáról.



8.8. ábra

8.9. ábra

A 8255 igen népszerű illesztő egység, az eredeti IBM PC a hangszóró kezelését, a nyomtató illesztését ezzel oldotta meg.

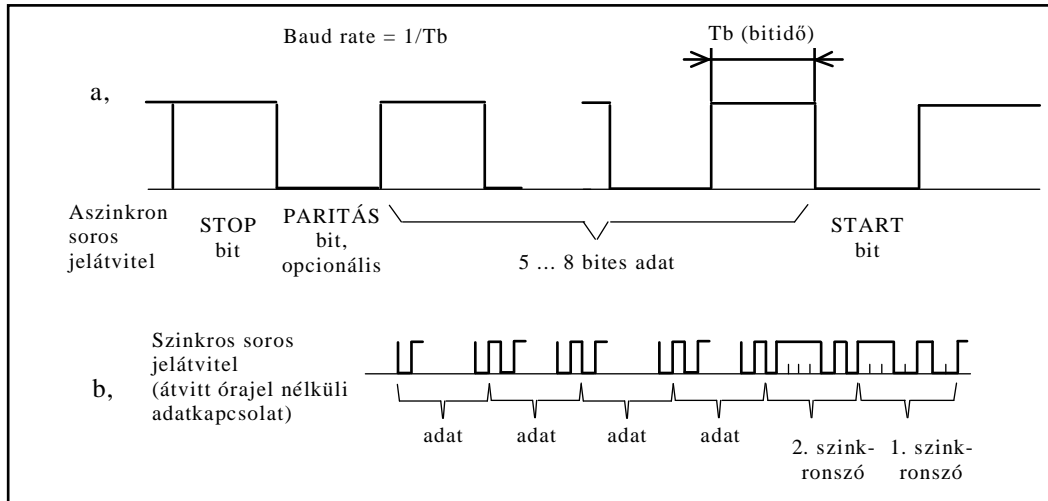


8.10. ábra

## 8.2. A 8251 kommunikációs illesztő áramkör

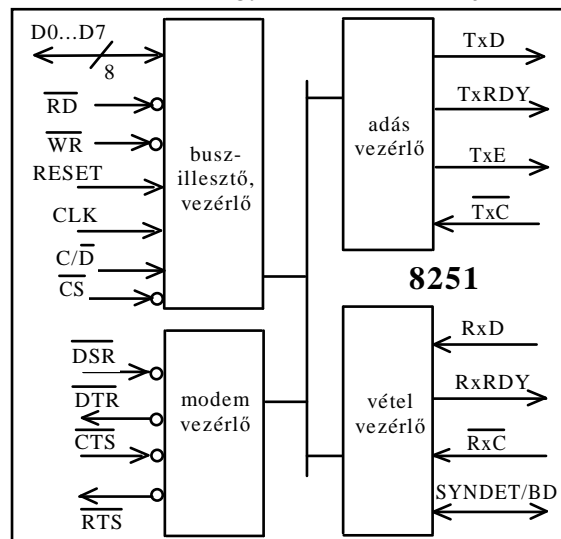
A soros adatátvitelre alkalmas programozható illesztő áramköröket szokás kommunikációs interfésznek nevezni, mivel a nagy távolságú adatátvitel (telekommunikáció) kizárólag soros jelátvitellel valósulhat meg.

A DIL 28 tokozású 8251 aszinkron és szinkron soros adatátvitelt tud megvalósítani, egy teljes duplex üzemi csatornával. Az adatszó hosszúsága lehet 5, 6, 7 vagy 8 bit, alkalmazható paritásbit is az átvitel során. Aszinkron üzemben egy Start bit, 1 vagy 1,5 vagy 2 Stop bit, opcionális paritásbit átvitel történik az adatbiteken kívül, a sebesség 0 ... 9,6 KBaud. Az aszinkron jelátvitel RS 232 C jellegű (8.11. a, ábra). A szinkron átvitel (8.11. b, ábra) a 8251 esetén a szinkronszavas megoldást jelenti, egy vagy két szinkronszavas lehet a működés, az átviteli sebesség 0 ... 56 KBaud.



8.11. ábra

Az aszinkron működés megfelel az RS 232 C szabványnak, kivéve a jelszinteket, mivel a 8251 TTL kompatibilis jelszintekkel dolgozik (az RS 232 C szerint a logikai 1 tartománya - 3 ... - 12 V, a logikai 0 feszültség sávja + 3 ... + 12 V). Az RS 232 C az adatot továbbító vezetékeken kívül egyéb vezérlő jeleket is definiál, azokat a 8251 nem kezeli. Az aszinkron átvitelnél sincs, a szinkron átvitelnél sincs az adóegység és a vevőegység között órajel-átvitel, a bitek szinkronizálását más módszerrel oldják meg. A mai szóhasználatban már általában nem a 8251 szinkronszavas átviteli megoldását értik szinkron soros átvitel alatt, hanem olyan jelkapcsolatokat, ahol az adó és a vevő között az adatokon kívül egy szinkronizáló órajelsorozat is átvitelre kerül.



8.12. ábra

Az aszinkron soros átvitelnél a bitidő nem választható meg tetszőlegesen. Reciprok értékét, a Baud Rate értéket egy szabványos sorból kell választani (75, 150, 300, 600, 1200, 2400, 4800, 9600 stb.). A soros adó és vevő rendszernek az adatkapcsolat felépülése előtt már meg kell állapodni abban, milyen adatsebességet, milyen Baud Rate értéket fognak használni.

A 8251 belső felépítését a **8.12. ábrán** láthatjuk. A buszillesztő, vezérlő áramkör legtöbb jele ismerős, a CLK pontra a mikroszámítógép alaposzcillátorából kell bevezetni az órajelet (ez az adási és vételi órajelnél legalább 40-szer nagyobb frekvenciájú legyen). A  $\overline{C/D}$  a belső elemek közül választ, a  $\overline{C/D} = 1$  parancsszót jelez, a  $\overline{C/D} = 0$  pedig adatot. A 8251 tartalmaz egy látszólagos MODEM vezérlőt, de a modem működését szoftverrel kell megoldani, az IC csak átengedi ezeket a jeleket magán. Valódi funkciója a 8251-en belül csak a  $\overline{CTS}$ -nek van, az adás megkezdéséhez ezen a bemeneten L szintnek kell állnia. A kimenő modem vezérlő jelek értékét a parancsszó egyes bitjei állítják be, a bemenő modemjelek értékét pedig az állapotszó megfelelő bitjeiként lehet megismerni. Tulajdonképpen a 8251 ilyen módon további két soros kimeneti és két soros bemeneti bitet is nyújt, közvetlen szoftveres kezelési lehetőséggel.

Az adó vezérlőegység jelei:

TxD soros kimenő adat,  
TxRDY az adó puffer üres, a 8251 adatot vár a mikroprocesszortól,  
TxE az adó léptetőregiszter üres,  
Tx $\overline{C}$  adó órajel (aszinkron üzemben az adó tényleges működési frekvenciája lehet ez vagy ennek 16-tal, 64-gyel leosztott értéke; szinkron üzemben nincs leosztási lehetőség).

A vevő vezérlő egység jelei:

RxD soros bejövő adat,  
RxRDY a vevő puffer betöltve, fogadta a bejövő adatot,  
Rx $\overline{C}$  vevő órajel (aszinkron üzemben az adó tényleges működési frekvenciája lehet ez vagy ennek 16-tal, 64-gyel leosztott értéke; szinkron üzemben nincs leosztási lehetőség),

SYNDET szinkronjel kezelő vezérlőjel; belső szinkron üzemben a SYNDET, mint kimenet, jelzi, ha a megfelelő szinkron szót beléptette a vevő és a következő bit már adatbit; külső szinkron üzemben ezen a ponton (mint bemeneten) lehet jelezni a 8251 számára, hogy a következő bit már adatbit lesz.

A 8251 belső regiszterei a parancsregiszter (CR), az üzemmód beállító regiszter (MR), az adó pufferregiszter (TX), a vevő pufferregiszter (RX), az 1. és a 2. szinkronszó regiszter (SWR1, SWR2). A belső elemeket a CPU a táblázat szerint tudja elérni:

$\overline{CS}$	$\overline{C/D}$	$\overline{RD}$	$\overline{WR}$	működés
L	L	L	H	$RX \rightarrow D0 \dots D7$
L	L	H	L	$D0 \dots D7 \rightarrow TX$
L	H	L	H	$SR \rightarrow D0 \dots D7$
L	H	H	L	$D0 \dots D7 \rightarrow MR$
L	H	H	L	$D0 \dots D7 \rightarrow SWR1$
L	H	H	L	$D0 \dots D7 \rightarrow SWR2$
L	H	H	L	$D0 \dots D7 \rightarrow CR$
H	X	X	X	nincs kijelölve

A CR, MR, SWR1, SWR2 eléréséhez a CPU ugyanazt a vezérlőjel kombinációt használja. A RESET utáni első esetben az MR-be kerül a belépő 8 bit, s ha aszinkron üzemet ír elő, minden ezután a CR-be jut. Ha az MR-be került 8 bit szinkron üzemet írt elő, a szinkron szavak számának megfelelően a következő egy vagy két 8 bites adat a szinkron szó tároló regiszterekbe kerül s csak azután érhető el a CR, de azután itt is folyamatosan.

Ha az MR betöltése megtörtént, üzemmód módosítást csak RESET után lehet végrehajtani. Azért, hogy ne kelljen a mikroszámítógépet a 8251 kedvéért resetelni, a CR-be küldhető parancsok között van egy ún. szoftver RESET, ami hatásában ugyan olyan, mintha a 8251 RESET lábára érkezett volna alaphelyzetbe állító jel. A 8251-nél eszerint megtaláljuk a címzéses belső elem választást is (hiszen a  $C/\overline{D}$  bemenet lényegében egy címző bemenet), és a hozzáfordulási sorrend alapján történő belső címzést is és rövidesen kiderül, hogy a harmadik megoldást is alkalmazták az IC tervezői, a tartalom alapján történő belső címzést.

MSB 8251 aszinkron üzemmód beállítás LSB								
S1	S2	EP	PEN	L2	L1	B2	B1	
B2,B1 : mivel nem 00, az aszinkron üzemmódot definiálja								
B2,B1 : sebesség tényező, az RxC, TxC leosztása								
mértéke: 01 = 1-szeres								
10 = 16-szoros								
11 = 64-szeres								
L2,L1,PEN,EP : ugyanaz, mint a szinkron esetben								
S2,S1 : a STOP bitek száma, 00 = érvénytelen								
01 = 1								
10 = 1,5								
11 = 2								

8.14. ábra

tartja.

A 8251 állapotregiszterét (8.16. ábra) olvasva néhány belső jel értékét lehet megismerni, valamint három hibabit állapotát.

**PE:** paritáshiba (vételnél észlelt paritáshiba); a 8251 valódi paritáskezelést végez, ha előírjuk a paritásbitet, a megfelelő (páros vagy páratlan) paritásbitet adáskor az áramkör számítja ki, hozzáilleszti a kimenő adathoz, vételnél is kiszámítja a paritásértéket, összehasonlítja a vett paritásbittel s hibát jelez, ha eltérés mutatkozik – a mikroprocesszortól az adásra kerülő adatokat paritásbit nélkül kapja és paritásbit nélkül szolgáltatja a vett adatokat.

**OE:** ráfutás hiba, az előző vett adat még a bemeneti pufferben volt, amikor a bemeneti léptetőregiszterben kialakult egy újabb bemeneti adat, az új adat nem írja felül az előzőt, elvesz.

**FE:** formátum hiba, az első Stop bit helyén 0 értékű bit jelentkezett. A Stop bit, tudjuk, 1 értékű. Mivel a szóhosszúság és a paritásbit léte vagy hiánya ismert, a 8251 „tudja”, hogy mikor kell beérkeznie a Stop bitnek. Ha helyette 1 értéket kap, akkor jelez hibát.

MSB 8251 szinkron üzemmód beállítás LSB							
SCS	ESD	EP	PEN	L2	L1	0	0
D1,D0 = 00 : a szinkron üzemmód definiáló byte jele							
L2,L1 : az adat hossza: 00 = 5 bit							
01 = 6 bit							
10 = 7 bit							
11 = 8 bit							
PEN : paritás ellenőrzés; 1 : engedélyezett							
EP : ha PEN = 1, a paritás jellege; 1 = páros, 0 = páratlan							
ESD : a külső/belső szinkron üzem; 1 : külső szinkron, a SYNDET bemenet; 0 : belső szinkron, a SYNDET kimenet							
SCS : a szinkron karakterek száma, 0 = 1, 1 = 2 szinkron szó							

8.13. ábra

Az MR tartalmának értelmezése a D1-D0 bitek értékétől függ. Ha itt 00 áll, szinkron üzemmódról van szó (8.13. ábra). A legkisebb helyértékű két bit minden más kombinációja aszinkron átvitelt rendel el (8.14. ábra). A parancs Byte a CR-be kerülő bitminta, a közvetlen működtetés a feladata (8.15. ábra), itt található meg a szoftver RESET lehetősége is.

A 8251 képes az adatkapcsolat ideiglenes felfüggesztésére is (Break), ehhez a soros kimenetet folyamatosan 0 értéken

MSB				8251 parancs byte			LSB	
EH	IR	RTS	ER	SBRK	RXE	DTR	TXEN	

TXEN : adás engedélyezés; 1 = engedélyezett

DTR : terminál készenlét, a DTR lábon a negáltja lép ki

RXE : vétel engedélyezés; 1 = engedélyezett

SBRK : break kiküldése; 1 = a TxD folyamatosan 0 lesz

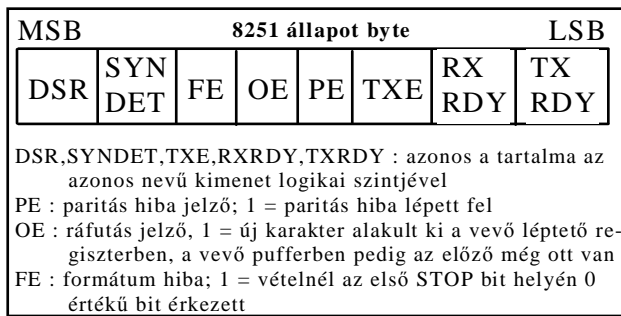
ER : a hibajelzők törlése az SR regiszterben; 1 = törlés

RTS : adás igény, az RTS lábon a negáltja lép ki

IR : belső RESET, az üzemmód átírásához ismét az üzemmód regiszterbe kerül a következő parancs byte (visszakapcsolás az MR regiszterre)

EH : 1 = belépés a szinkronszó figyelő (HUNT) módba

8.15. ábra



8.16. ábra

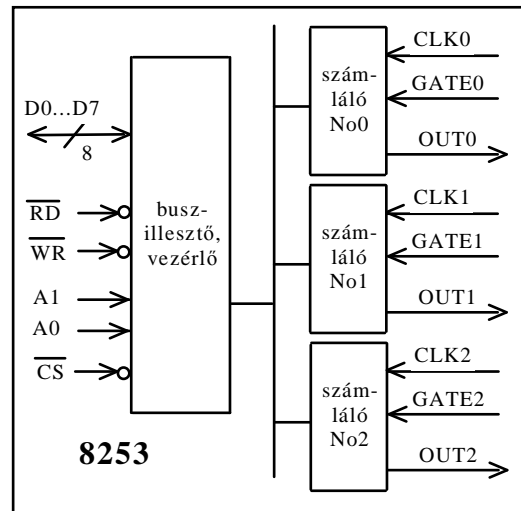
A 8251-ben programozható a Stop bitok száma (lehet 1 vagy 1,5 vagy 2 Stop bit). Vételkor viszont csak az első Stop bitot figyel. Mi lehet az értelme a különféle Stop bit hosszúságoknak? Tulajdonképpen arra használhatjuk fel ezeket, hogy a vevőnek időt biztosítsunk a vett adat feldolgozására. Ha pl. a 2 Stop bit időtartam már elég a vevőnek arra, hogy átadja processzorának a beérkezett adatot, az adás folyamatos lehet.

### 8.3. A 8253/8254 időzítő/számláló áramkör

Az időzítő/számláló áramkör általában ugyanaz az áramkör, a felhasználás két lehetőségére utal a kétféle elnevezés. Időzítőként a mikroszámítógép órajelét számlálja az elem, számlálóként valamilyen külső esemény jeleit. A DIL 24 tokozású 8253 a 8080A-hoz készült programozható kiegészítő áramkör. Korszerűbb változata a 8254, ami a 8253 minden képességét tartalmazza és néhány újjal is kiegészült.

A 8253 belső felépítését a 8.17. ábra mutatja be.

Három számlálóegységet tartalmaz, ezek kaszkádba is köthetők. Az egyes számlálófokozatok 16 bites lefelé számlálók, külső léptető jellel (CLK) dekrementálhatók, a működésük kapuzható is (GATE); a számláló kiürülését egy kimenőjel jelzi (OUT). A 8253 legnagyobb léptetési frekvenciája 2 MHz, alsó frekvenciahatár nincs.



8.17. ábra

A CPU a vezérlő jelek segítségével a táblázat szerint éri el az áramkör belső egységeit:

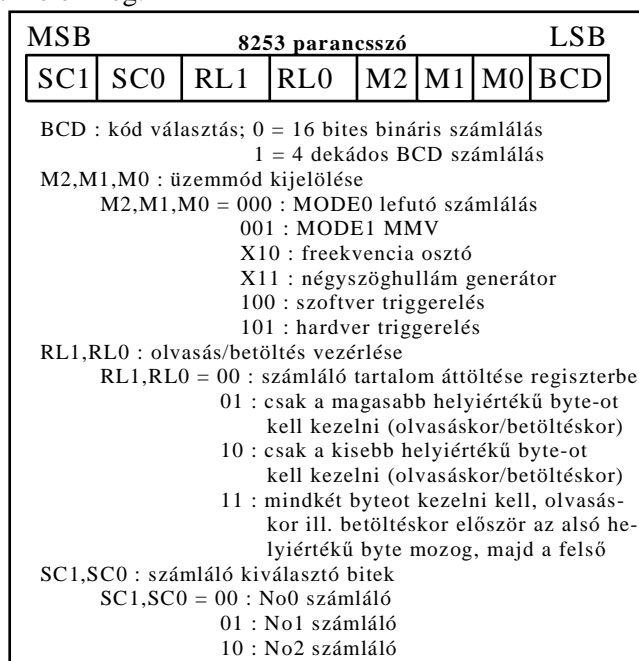
RD	WR	A1	A0	működés
H	L	0	0	D0 ... D7 → #0 számláló
H	L	0	1	D0 ... D7 → #1 számláló
H	L	1	0	D0 ... D7 → #2 számláló
H	L	1	1	D0 ... D7 → CWR
L	H	0	0	#0 számláló → D0 ... D7
L	H	0	1	#1 számláló → D0 ... D7
L	H	1	0	#2 számláló → D0 ... D7
L	H	1	1	D0 ... D7 lebeg

A táblázatból látható, hogy ennél a programozható kiegészítő áramkörnél a címzőbemeneteket használta az Intel a belső elemek megkülönböztetésére – a parancsszavak elemzéséből kiderül majd, hogy ezt ki kellett egészíteni a tartalom alapján történő címzéssel is.

A parancsszóregiszterbe (CWR) kerülő adat-Byte jelentését a 8.18. ábra mutatja be. A számlálók tartalma bármikor olvasható és írható. Egy számláló felépítését a 8.19. ábra szemlélteti. Ha két Byte-os adatkezelést ír elő a parancsszó, olvasáskor is, íráskor is először az alacsony

nyabb helyértékű Byte mozog, majd második alkalommal a magasabb helyértékű. A két Byte-os működés előírása esetén az Intel azt javasolja, hogy minden esetben közvetlen egymás után történjen meg mind a két Byte mozgatása, ellenkező esetben egy későbbi olvasáskor vagy íráskor másik Byte kezelése történhet meg, mint amire számítunk (vagy körülményes módon folyamatosan figyelemmel kell kísérni, és tárolni is kell, melyik Byte kezelése történt meg és melyiké nem).

A számláló-tartalom közvetlen kiolvasása helyett célszerűbb a tartalmat paranccsal átírni a számláló melletti regiszterbe, a számláló olvasási működése ekkor a regiszter tartalmát fogja az adatbuszra helyezni. Ha a számlálót közvetlenül olvassuk, és az alsó Byte olvasásakor a számláló tartalma pl. 0001h, mire a felső Byte kiolvasásra kerül, beérkezhet két léptetőjel, s az akkori FFFFh tartalomnak megfelelően a teljes kiolvasott tartalom FF01h érték, ami nyilvánvalóan hamis adat. Ha a kiolvasást a regiszterbe töltéssel indítjuk, ilyen hiba nem lép fel, igaz viszont, hogy akkor a kiolvasott érték nem a kiolvasás pillanatában érvényes számláló tartalom, hanem az átírás időpontjának felel meg.



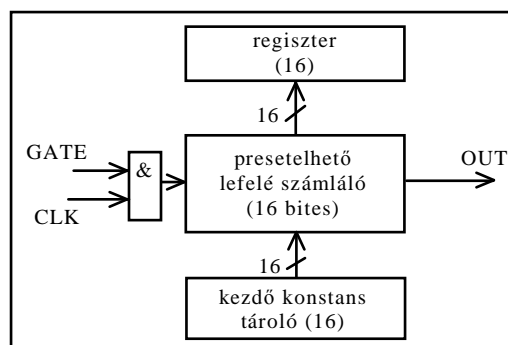
8.18. ábra

A MODE0 működés egyszerű lefelé számlálás, indításakor a számláló kimenete 0 értékű. Ha a számláló kiürül, a kimenete 0-ról 1-re vált.

A MODE1 újraindítható monostabil multivibrátor, a kezdőérték betöltése után a GATE felfutó éle indítja, a kimenet ekkor 0 lesz. Ha a leszámolás előtt ismét kap indító jelet, az eredeti kezdőértékkel kezd ismét lefele számolni. Ha a számláló kiürül, a kimenete 1-re vált.

A MODE2 frekvenciaosztó, minden teljes le-  
számáláskor egy léptetőjel ciklusig 0 értékű a  
kimenet, újratöltődik a kezdőérték és folytatódik a  
működés.

Hasonló a MODE3, de a leszámítás feléig a kimenet 1 értékű, a második felében pedig 0 értékű, így az eredmény négyszöghullám lesz. A MODE3 kapcsán tetten érhető az Intel mérnökeinek precizitása.



**8.19. ábra**



Ha a beállított kezdőérték páros szám, valóban a feléig lesz a kimenet 1 értékű, a feléig 0. Páratlan kezdőérték esetén többféle megoldás is szóba jöhetne, de a 8253 ebben az esetben is a működési idő felében 0, felében 1 kimeneti értéket állít elő. Ezt úgy tudja megtenni, hogy az egyik ciklusban a 0 kimenet lesz páratlan hosszúságú és az 1 páros, a következőben pedig fordítva. Hosszabb idő alatt így valóban pontosan az összes idő feléig 0, feléig 1 értékű a kimenet!

A MODE4 szoftver triggerelt leszámolás, a szoftver betöltésekkel indítható a működés; a MODE5 hardver triggerelt leszámolás, ennél a GATE jel indítja precízen a leszámolást.

MSB		8254 Read-Back Command					LSB
1	1	COUNT	STATUS	CNT2	CNT1	CNT0	0
CNTi : amelyik biten 1 áll, arra a számlálóra vonatkozik (egyszerre több számlálóra is elrendelhető) STATUS : ha itt 0 áll, a számláló állapotbitjeit áttölti az állapotregiszterbe COUNT : ha itt 0 áll, a számláló tartalmát áttölti a regiszterbe							

8.20. ábra

A 8254 már 10 MHz-es CLK jellel léptethető. Egy új parancsszót is kezel, ún. visszaolvasási parancsot (Read-Back Command, RBC). Az RBC parancsszó bitjeit a **8.20. ábra** foglalja össze.

A parancsszó segítségével elrendelhető bármelyik számláló tartalmának regiszterbe töltése, illetve az állapotszavának áttöltése. Az áttöltött információkat fogja olvasási működéskor a mikroprocesszor elérni. Egy-egy biten lehet megjelölni, melyik számlálóra érvényes a parancs, de egyszerre akárhány számlálóra kiadható. A számláló és az állapotregiszter átírása is előírható egyszerre, ekkor az első olvasáskor az állapotregiszter tartalmát kapjuk meg, ezután a számláló tartalmát. A 16 bites tartalom kiolvasásával kapcsolatos problémát korábban bemutattuk, azt a számláló regiszterbe áttöltése megoldotta. Ha azonban kaszkádban működnek számlálók, a 8253-nál nem oldható meg a számlánc teljes, akár 48 bites tartalmának regiszterbe töltése, csak 16 bites részletekben egymás után. A 8254 az új parancs segítségével akár a 48 bites számlánc tartalmát is egyetlen lépésben regiszterbe tudja tölteni.

Az állapotinformációkat a **8.21. ábra** szerint lehet értelmezni.

MSB		8254 állapotszó					LSB
OUT	NULL COUNT	RL1	RL0	M2	M1	M0	BCD
RL1,RL0,M2,M1,M0,BCD : a beállított paraméter értékek NULL COUNT : a kezdőérték beállító egység állapotára utaló bit; 1-re vált amikor a CPU a kezdőértéket beírja a 8254 kezdőérték tárolójába és 0 lesz, ha a kezdőérték áttöltődött a számlálóba OUT : a számláló OUT kimenete logikai állapota							

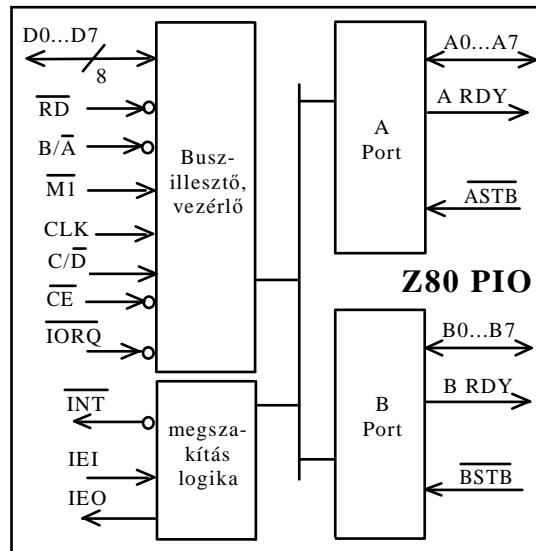
8.21. ábra

#### 8.4. A ZILOG programozható kiegészítő elemeinek megszakításkezelése

A Z80 programozható kiegészítő áramköreinek (a Z80-PIO párhuzamos illesztő, a Z80-SIO soros illesztő, a Z80-CTC időzítő-számláló stb.) mindegyike tartalmaz egy egységes kialakítású megszakítás-kezelő egységet.

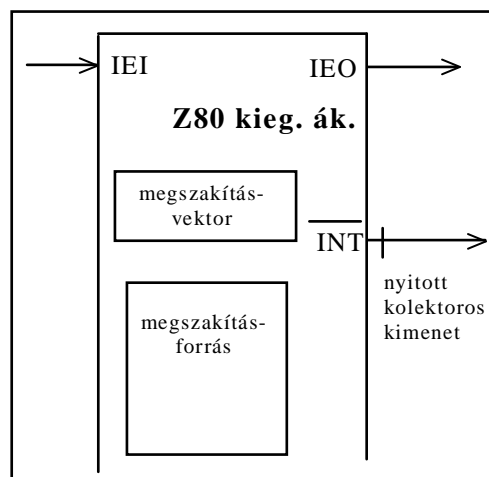
Példaként a **8.22. ábrán** a Z80-PIO belső felépítése látható. A megszakítás-kezelő logika csatlakozó pontjai, az INT, az IEI (megszakítás-engedélyezés bemenet) és az IEO (megszakítás-engedélyezés kimenet) minden Z80 kiegészítőnél megtalálhatók.

Egy pillanatra érdemes elidőzni itt, mert egy érdekes tanúságra bukkanhatunk. Tapasztalatból lehet tudni, hogy a 8080A programozható kiegészítőit, így a megismert áramköröket is szívesen alkalmazzák más mikroprocesszorok mellett – egészen a mai, nagy szóhosszúságú processzorokig. A Z80 programozható kiegészítők nem ilyen népszerűek, más processzor mellett ritkán találkozunk velük. Ennek az a magyarázata, hogy relatíve drágák.



8.22. ábra

Minden kiegészítő áramkör tartalmazza ugyanis a meglehetősen bonyolult (de természetesen nagyon intelligens) megszakítási logikát, amit azonban csak a Z80-CPU mellett lehet használni. A 8080A programozható kiegészítői viszont egyáltalán nem mikroprocesszor-specifikus elemek, ezért mindenhol jól felhasználhatók, minden megfizetett képességet ki lehet aknázni.



8.23. ábra

A 8.23. ábrán látható módon a Z80 kiegészítő áramkörben minden megszakításforráshoz tartozik egy megszakításvektor regiszter, ennek a tartalmát tudja az áramkör az INTA ciklusban az adatbuszon a mikroprocesszorhoz küldeni. A Z80 a kiegészítők megszakítás-kezelő képességét akkor tudja használni, ha IM2 megszakítási állapotban van.

Amikor a programozható kiegészítőnek nincs megszakítási igénye, a megszakítási logika passzív állapotban van, az IEI pont logikailag össze van kötve az IEO ponttal, utóbbin az a jel-érték lép ki, ami az IEI pontra kerül.

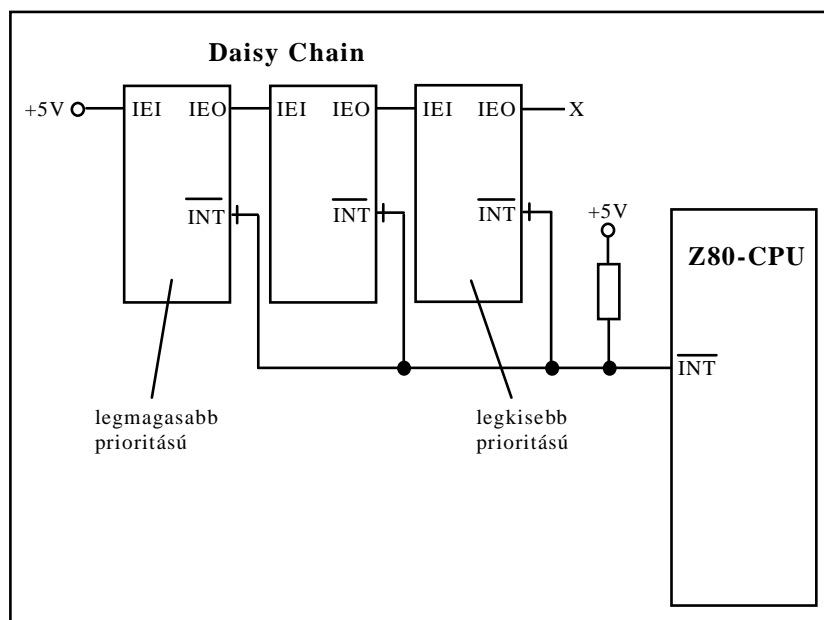
Ha egy megszakításforrás megszakítást kíván elérni és az  $IEI = H$ , a megszakítási logika aktív állapotba kerül. (Ha az  $IEI = L$ , nem történik meg az átkapcsolás, továbbra is passzív lesz a megszakítási logika.)

Az aktív állapotban az  $IEI$  és az  $IEO$  összekapcsolása megszűnik, az  $IEO = L$  értékre vált, s az  $L$  szint megjelenik az  $\overline{INT}$  kimeneten is, a huzalozott VAGY működés révén így a Z80 megszakításkérő bemenetén is. Az áramkör figyelni kezdi az adatbusz-t és a vezérlő jeleket. Ha a CPU elfogadta a megszakítást,  $INTA$  ciklusba kerül.

Az  $INTA$  ciklus kezdetét érzékeli a kiegészítő áramkör (az egyidejű  $\overline{M1}$  és  $\overline{IORQ}$  felismerése útján). Ekkor az  $IM2$ -nek megfelelően az áramkör az adatbuszra helyezi a megszakítási vektor alsó részét (a felső Byte a CPU I regiszteréből kerül ki).

A Z80 a vektor segítségével megkeresi a kezelő szubrutin kezdőcímét, a mentések után rálép a szubrutin első utasítására. Időközben a kiegészítő áramkör megszünteti az  $L$  szintet az  $\overline{INT}$  kimeneten.

A kiegészítő áramkör tovább figyeli az adatbusz-t és a vezérlő jeleket, minden  $M1$  ciklusban elemzi az opkódot. Ha a programban megjelenik a  $RETI$  utasítás, azt a kiegészítő áramkör felismeri, számára ez jelzi a megszakítási folyamat végét. Ekkor a megszakítási logika ismét felveszi passzív állapotát, összekapcsolva az  $IEI$  bemenetet az  $IEO$  kimenettel.



8.24. ábra

Emlékezzünk vissza, hogy a Z80 megszakításrendszerével kapcsolatban tanultuk, hogy a megszakítást kezelő szubrutint  $RETI$ -vel kell lezárni, az egyéb szubrutinokat  $RET$ -tel. Akkor azt is megírtuk, hogy a kétféle  $RET$  a mikroprocesszor számára ugyanazt jelenti.

De eltérő a jelentésük a kiegészítő áramkörök számára! A  $RET$  nem okoz a kiegészítő áramkörben változást, a  $RETI$  hatására viszont beszünteti az aktív állapotát és felszabadítja a Daisy Chain vonalat (összekötve  $IEI$  és  $IEO$  pontjait). Mi ez a Daisy Chain?

A Z80 programozható kiegészítő áramkörök megszakítás-engedélyező jeleit kaszkádba lehet kötni, ezt a felfűzést nevezik Daisy Chain-nek (8.24. ábra). A Daisy Chain-ben elfoglalt pozíció az áramkör számára meghatározza a megszakítási prioritási helyzetét. A bal szélső elem  $IEI$  bemenetén mindig  $H$  szint áll, ez a legmagasabb megszakítási prioritású, kérése mindig érvényre jut. Minél távolabb van innen egy elem, annál kisebb a prioritása, a legkisebb a jobb szélsőé (az csak akkor kérhet megszakítást, ha tőle balra semelyik áramkör nem kér).

A Z80 programozható kiegészítőkből belül is van egy-egy 3-4 fokozatú Daisy Chain fűzér, ezért csak négy áramkör összekapcsolását engedi meg a ZILOG.

Ha több áramkört kell összefűzni, speciális logikai hálózattal kell az IEI/IEO pontok jeleit kezelni, amire a ZILOG ajánlásokat ad katalógusaiban.

A megszakítások kezelésével kapcsolatban láttuk, milyen nagy jelentősége van annak, hogy a megszakítás-kéréseket egyértelműen fontossági, prioritási sorrendbe rendezzük. Megismerve a Daisy Chain megoldást, gondoljuk át, milyen módszerekkel találkoztunk a megszakítások prioritási rendjének kialakítására!

Az egyik megoldást a 8085A mikroprocesszornál láttuk: a mikroprocesszoron belül alakítottak ki nagyszámú megszakítást kérő bemenetet és azokat a belső hardverrel rögzített prioritási sorrendbe állították.

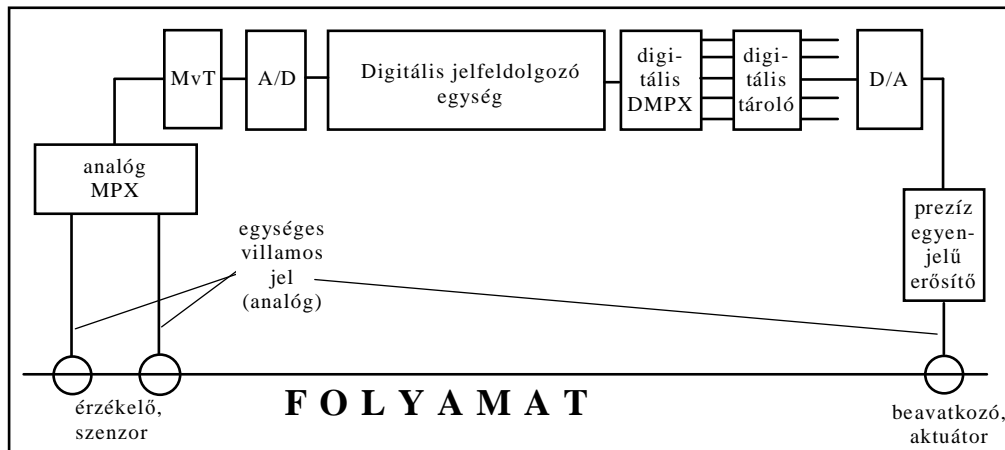
A másik lehetőséget a Zilog alakította ki a Daisy Chain kifejlesztésekor, itt az egyes elemek fizikai helyzete határozza meg a prioritásukat, az, hogy a füzérben hol helyezkednek el.

Végül csak utalunk egy harmadik lehetőségre, a felhasználó által programmal beállítható prioritási sorrendre. A 8080A egyetlen megszakítás-kérő bemenettel rendelkezik, a hatékony megszakítás-kezeléshez programozható megszakítás-kezelő áramkört is fejlesztett az Intel. Az nyolc megszakítás-kérést fogad és a beprogramozásakor azt is elő lehet írni, melyik megszakítás-kérő bemenetnek milyen legyen a prioritási értéke.

Egyébként az itt jelzett megszakítás-kezelő programozott áramkör a 8228-cal együttműködve a memória tetszőleges címéről tud szubrutint indítani, mert INTA gépi ciklusban CD opkódot (CALL utasítás) küld a processzorhoz a 8228-on át. Ebben a helyzetben a 8228, mint tanultuk, lehetővé teszi a CALL 16 bites címének beküldését. A programozható megszakítás-kezelő ezért minden bemenetéhez nemcsak prioritási értéket rendel, hanem egy 16 bites címet is tárol.

## 9. Az A/D és D/A konverterek

A mikroszámítógépek alkalmazásakor igen gyakori, hogy analóg jeleket kell kezelni, fogadni és kiadni. Leginkább az irányítási célra készült mikrogépeknél adódnak ilyen igények (9.1. ábra).



9.1. ábra

Az érzékelők, szenzorok többsége analóg jelet állít elő, a beavatkozók, aktuátorok analóg jelekkel működtethetők. Amikor az irányítástechnikában, kb. 25 éve megjelentek az első számítógépek, már többen arra tippeltek, hogy a szenzorok, aktuátorok is átalakulnak majd digitálissá. Amikor kb. 10 éve gyakorlatilag eltűntek az analóg irányító készülékek, szabályozók, azóta minden irányítási feladatot számítógép old meg, ekkor már minden szakmabeli úgy gondolta, hogy a szenzorok, aktuátorok területén is kizárólagossá válik a digitális megoldás. Nos, a várakozással ellentétben a szenzorok nagy része máig is az egységes villamos analóg jeleket szolgáltatja és a beavatkozók, végrehajtók is majdnem mindannyian az analóg jelekkel működnek. Ha ezek az elemek idővel mégis mind közvetlenül digitális jelekkel működnek, akkor is ismernünk kell az A/D és D/A áramköröket, hiszen csak annyi történik, hogy ezek a konverterek majd bekerülnek az érzékelőkbe, a beavatkozókba. Vegyük sorra, hogyan oldható meg a probléma a mikroszámítógépeknél. A mikroszámítógéphez beérkező analóg jeleket A/D átalakítókkal digitálissá kell konvertálni, s a kimenő oldalon D/A átalakítókkal kell az analóg kimenő jeleket előállítani.

Ha több analóg jelet fogad a mikrogép, azokat multiplexelni kell. A multiplexelés megoldható az analóg oldalon is és a digitális oldalon is. Analóg multiplexer esetén egyetlen A/D elegendő; ha a digitális oldalon multiplexelünk, egyenként kell a jeleket A/D átalakítóval kezelni (viszont a digitális multiplexer olcsó tömegárú, az analóg multiplexer drága és a jel típusától függő elem). Sokszor találkozunk olyan megoldással, hogy néhány analóg bemenő jelet analóg multiplexer fogad és azokat egy A/D-re vezeti, más analóg jeleket saját A/D fogad, s itt a digitális jelet multiplexelik. Ennek az a magyarázata, hogy a beérkező analóg jelek különfélék, és különfélék a feldolgozási igények is. Lehet néhány olyan bemenő analóg jel, amit csak egyedi, speciális A/D tud kezelni.

Az analóg jellel együtt zajok, zavarok is érkezik a mikroszámítógéphez. A zajok ellen szűréssel lehet védekezni. Az analóg jelek lassan változó egyenáramú, egyenfeszültségű jelek, a zajok, zavarok többsége nagyobb frekvenciájú és ezért olcsó RC szűrőkkel leválaszthatók a jeltől. Az RC-szűrőket közvetlenül a jeladó kimenetéhez szokták telepíteni. Egy másik koncepció szerint gyors A/D konverterrel a zajt is alakhűen digitalizálják és a feldolgozó szoftverrel végzik el a szűrést. Végül egyes A/D átalakítók felépítésüknél fogva bizonyos típusú zajokat kiszűrnek a beérkező jeltől (pl. az integráló és a szigma-delta modulátoros konverterek).

Az A/D konverterek egyes változatai megkívánják, hogy a konverzió alatt a bemenetükön állandó legyen a jelérték. Ha a feldolgozott jel a konverziós idő alatt jelentősen változhat, s az átalakító bemenetén a teljes átalakítási idő alatt jelen kell lennie az analóg jelnek, akkor az A/D bemenete elé mintavevő-tartó (MvT) áramkört kell kapcsolni.

Az A/D sebessége, a konverziós idő rövidege vagy hosszúsága relatív fogalom, mindig az adott alkalmazásban fellépő bemenőjel változási sebességéhez kell viszonyítani. Ugyanaz az A/D az egyik alkalmazásban (pl. egy hőmérséklet-szabályozásban) lehet gyors, ami egy másikban (pl. egy nukleáris erőműben) elfogadhatatlanul lassú.

A mintavevő-tartó áramkört is készen lehet beszerezni, nem kell megépíteni. Általában úgy épül fel, hogy mintavételkor egy kondenzátort tölt fel az analóg jelnek megfelelő értékre, s a kondenzátor feszültségét műveleti erősítő jeleníti meg a kimeneten. Annak érdekében, hogy a mintavett érték megmaradjon, az elektronikát FET alapúra készítik. Így sem 100 %-os a feszültségtartás, az MvT pontosságát úgy szokás megadni, hogy mennyi idő alatt hány %-kal csökken a benne tárolt minta értéke.

Az analóg/digitál átalakító a nevének megfelelően analóg jelet fogad és digitális jelet állít elő a kimenetén. Általában a digitális jelminták előállításának szaporaságát, az analóg jelből történő mintavételek gyakoriságát a Shannon-féle mintavételezési törvény szerint választják meg („mintavételezett” jelfeldolgozás). A Shannon-törvény szerint egy jel mintavételezésekor a mintavételeli frekvencia az analóg jel maximális frekvenciájú összetevője frekvenciájának kétszerese legyen, ez biztosítja a helyreállíthatóságot, az így képződő digitális jelből később D/A átalakítással alakhelyesen visszaállítható az eredeti analóg jel. Így tehát a Shannon-törvény szerintinél szaporább mintavétel feleslegesen drágítja a rendszert.

A D/A kimenetén megjelenő jel általában meghatározott időpillanatokban változik, a lehetséges amplitúdó-értékei korlátozott számúak. Ha a D/A 8 bites, 256 féle kimeneti feszültség-szintet tud előállítani. Ezt a kimenőjelet sok osztályozás nem analóg jelnek tekinti, így sokan még a D/A elnevezést is vitatják.

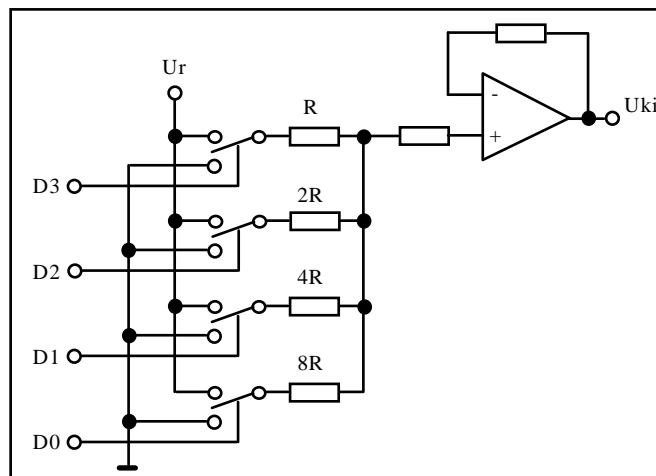
Mivel nemzetközileg elfogadott a D/A megnevezés, és a kialakuló kimenőjelet elterjedten analóg jelnek tekintik, mi is így járunk el. Egyébként egy szűrőtagon átvezetve a kimenőjel lépcsőzöttsége már eltűnik, a jel kisimul, és mindenki által elfogadott analóg jellé válik.

A D/A kvantált analóg jelet állít elő, kis energia szinttel. Ezt a jelet – a végrehajtó, beavatkozó szervek számára – erősíteni szükséges, méghozzá igen magas minőségű egyenáramú erősítővel. Már az egyenáramú erősítés igénye önmagában is drágítja az erősítőt, ráadásul (ha nem akarjuk elveszíteni a 12, 14 vagy 16 bites feldolgozás által szerzett előnyüket) precíziós erősítőkre van szükség, offset hiba nélküli, tökéletesen lineáris karakterisztikájú áramkörökre. Ha az erősítő karakterisztikája görbe, ha a nullpontja változik, hiába van a bemenő oldalon nagy felbontású A/D, hiába dolgozik a digitális központi egységben 16 bites szoftver, hiába alkalmazunk nagy felbontású D/A konvertert, a teljes rendszer pontatlan lesz. Ez az erősítés-igény jelentősen drágítja ezt a megoldást.

A valódi D/A konverzió helyett ezért egyre inkább a kvázi analóg jelek előállítását részesítik előnyben, melynek legkifinomultabb formája az impulzusszélesség moduláció (PWM). A kvázianalóg jeleket, a PWM jelsorozatot a CPU közvetlenül, szoftver úton képes előállítani s a végrehajtóra olcsó kapcsoló elemekkel lehet rávezetni (pl. nagy teljesítményű, kis veszteségű FET kapcsoló tranzisztorokkal), így elmarad a D/A konverter is és a speciális erősítő is.

Ebben a fejezetben nem ismerünk meg konkrét A/D vagy D/A típusokat (gyártó és IC-típuszám szerint). Számos félvezető-gyártó sok ezer különféle változatban készíti ezeket az elemeket. A különféle elvi megoldásokat fogjuk áttekinteni, a fő típusok fontosabb tulajdonságait, így a konkrét áramkörök közötti választáshoz kapunk segítséget, ahhoz, hogy az A/D és D/A áramkörök katalógusait értelmezni tudjuk. Az adott feladatot kielemezve (a jelek jeltartománya, a jelváltozás sebessége stb.) után lehet kiválasztani a konkrét konvertereket.

## 9.1. D/A konverterek és a helyettesítő megoldások

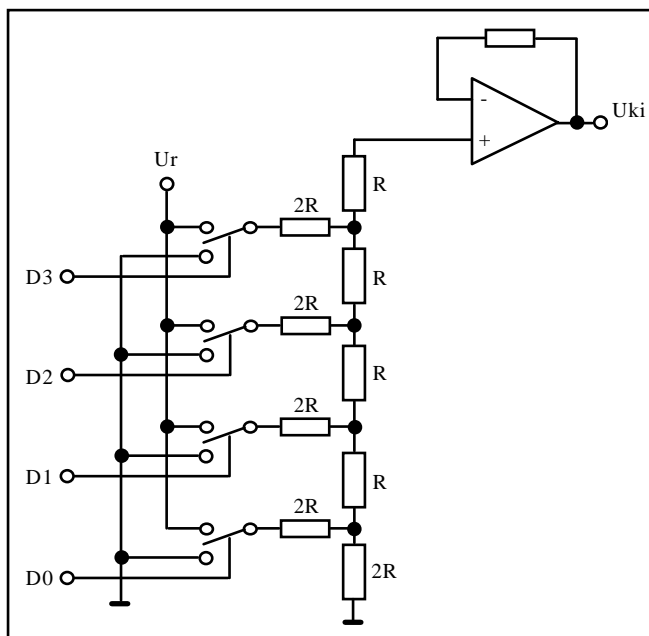


9.2. ábra

A valódi D/A konverterek *kapcsolt ellenálláshálózattal* állítják elő a bemenő kóddal (többnyire kettes számrendszerbeli kóddal) arányos kimenő analóg jelet.

Az ellenállásokat kapcsoló tranzisztorok iktatják be vagy kapcsolják ki, az ellenálláshálózat lehet *súlyozott értékekből* kialakított (9.2. ábra), vagy R/2R jellegű *létrahálózat* (9.3. ábra).

A súlyozott értékekből álló ellenálláshálózatnak az a hátránya, hogy sok különféle ellenállásérték szerepel benne, a gyártása problémás, a behangolása nehézkes. A létrahálózatban csak kétféle ellenállás található, így a jelzett gondok csökkennek. A valódi D/A konverterek többsége áram kimenetű és egy műveleti erősítőre épülő áram/feszültség átalakítóval lehet feszültség jelhez jutni a kimenetükről, de azért a választékban találunk közvetlenül feszültség kimenetű D/A elemeket is.

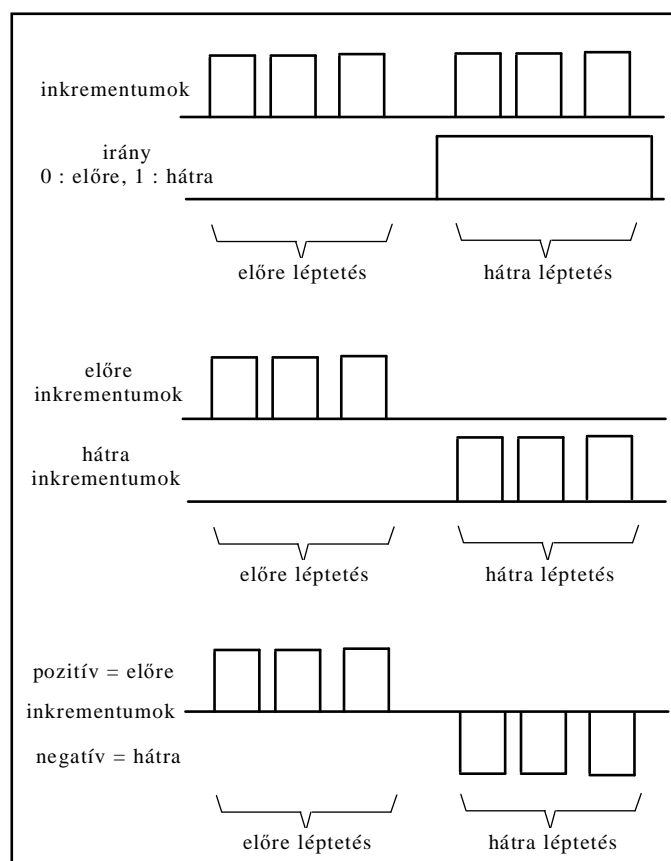


9.3. ábra

A konverterek vagy unipolárisak (0 ... +5V, 0 ... +10V stb.), vagy bipolárisak (- 5V ... + 5V stb.). A kimenő-jel tartománya a konverter tápfeszültségével vagy referencia feszültségével arányos, így igény szerint megválasztható (ha a konverter széles feszültségtartomány alkalmazását engedi meg).

A korai D/A átalakítók hibrid áramkörök voltak, a precíziós ellenállásokat nem a chip felületén alakították ki hanem egyedileg gyártották és úgy szerelték össze. Az integrált áramköri gyártási technológiák fejlődése tette lehetővé, hogy idővel a nagy felbontású, precíziós D/A konverterek is kialakíthatók lettek monolitikus, egy chip-es módon. Korábban a D/A konverterek két tápfeszültséget igényeltek, az analóg kimenő jel teljes

tartománya (pl. 0V ... +5V) alsó értékei, a 0V környezetébe tartozó jelek előállítására érdekében egy negatív tápfeszültséget is igényeltek; ma már többnyire egy tápfeszültségűek ezek az áramkörök.

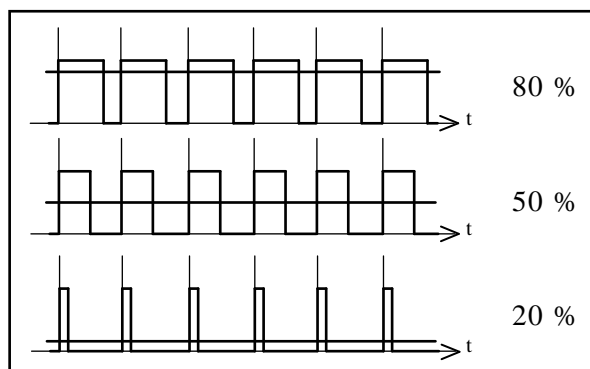


9.4. ábra

Az első kvázi analóg jeleket az inkrementális működtetéshez dolgozták ki. Ha a beavatkozó, végrehajtó számára nem az abszolút pozíciót kell egyenáramú jellel megadni, hanem az utolsó helyzethez képest szükséges elmozdulás mennyiségét (az inkrementumokat), a 9.4. ábra szerinti megoldásokat lehet használni. Az első kettő előnye, hogy csak digitális jeleket használnak, igaz, hogy két kimeneti pontra van szükség az alkalmazásukkor. Az inkrementális jelek előállításához nincs szükség speciális áramkörre, konverterre, ez logikai jel, ami a mikroszámítógép bármelyik kimeneti egységén, szoftver úton képezhető. Ha a beavatkozó léptethető jellegű (számláló mű, léptető gép, léptető motor stb.) vagy integráló működésű, az inkrementális jeleket közvetlenül fel tudja dolgozni, a valódi analóg jelet igénylő elemek számára integráló áramkörrel lehet felhasználhatóvá tenni az inkrementumokat.

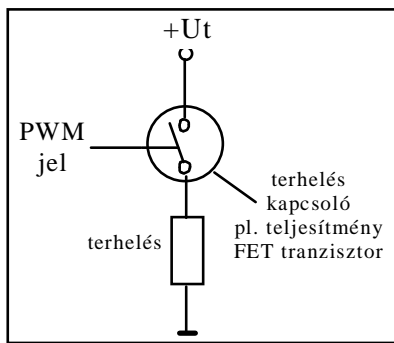
Az *impulzusszélesség modulált (PWM)* jel azonban már bármilyen, egyenfeszültségű vagy egyenáramú működtető jelet igénylő analóg eszköz működtetésére közvetlenül alkalmas. A PWM állandó frekvenciájú, de a digitális jellel arányos kitöltési tényezőjű impulzussorozat (9.5. ábra). A PWM jelet a CPU szoftverrel képes előállítani, nincs speciális áramköri igénye, a Port pontok egyikén kiküldhető.

Egyenáramú, egyenfeszültségű táplálást igénylő terhelést a PWM jellel közvetlenül lehet működtetni, az eszköz az átlagértéknek megfelelően fog viselkedni - így pl. egy egyenáramú motor fordulatszámát lehet beállítani, egy fényforrás fényerejét stb.



9.5. ábra



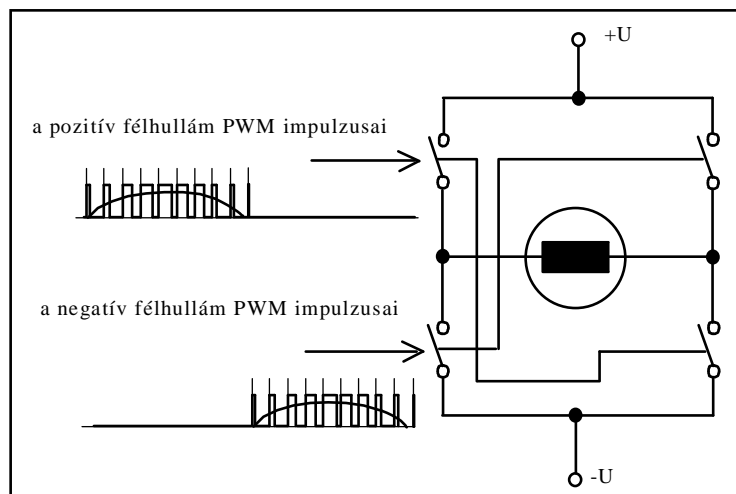


9.6. ábra

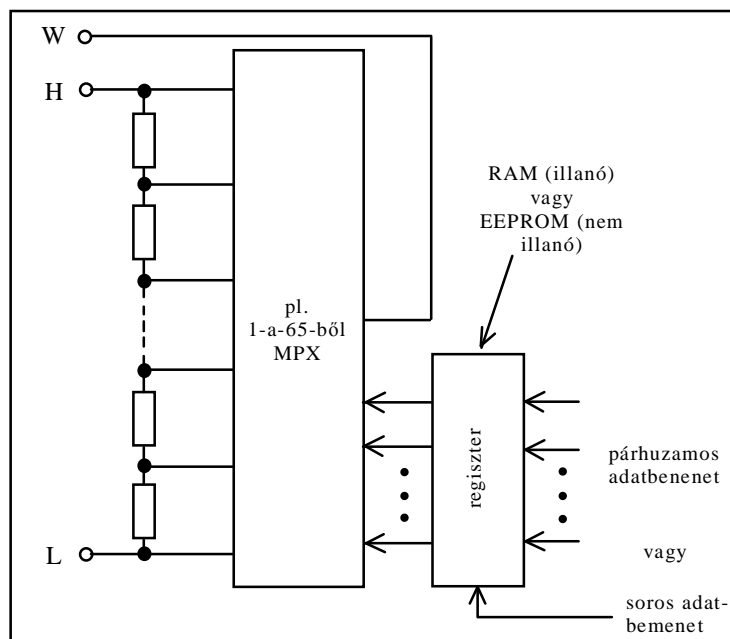
különbféle értéket vehet fel.

Ha a kitöltési tényezőt pl. szinusz függvény szerint változtatják, váltakozóáramú készülék is működtethető PWM jellel. Az áramirány váltását hídkapcsolással lehet elérni (9.7. ábra).

Ez a megoldás megháromszorozva és az egyes kimeneteknél  $120^\circ$ -os fáziseltolást alkalmazva háromfázisú motorok fordulatszám kezelésére is lehetőséget teremt. Ilyen PWM alapú eszközök az egyre népszerűbb frekvenciaváltók, melyeket aszinkron motorok fordulatszámának kezelésére használnak.



9.7. ábra



9.8. ábra

Ma már ezekkel az eszközökkel kezelik a szivattyútelepeken a szivattyú-motorokat (vegyipari üzemekben, víztisztítóknak stb.), általában azokat a háromfázisú motorokat, melyeknél a terhelés a névleges értékhez képest kicsiny, viszont a fordulatszámot pontosan be kell állítani.

A D/A konverzió egy különleges megoldási lehetőségét nyújtják a digitális elektronikus potencióméterek. Ezek a monolitikus integrált áramkörök a csuszka helyzetére vonatkozó információt digitálisan fogadják, a potencióméterüket azonban ugyanúgy lehet bekötni, mint egy hagyományos elektromechanikus potenciómétert (9.8. ábra), a digitális vezérlőoldal és a potencióméter-részletek egymástól galvanikusan függetlenek. Beszerezhetőek lineáris és logaritmusos karakterisztikával is, léteznek egyes és kettős digitális potencióméterek. Vannak közöttük SRAM jellegűek és EEPROM tárolásak is, utóbbiak a csuszka helyzetre vonatkozó információt tápfeszültség nélkül is megőrzik.

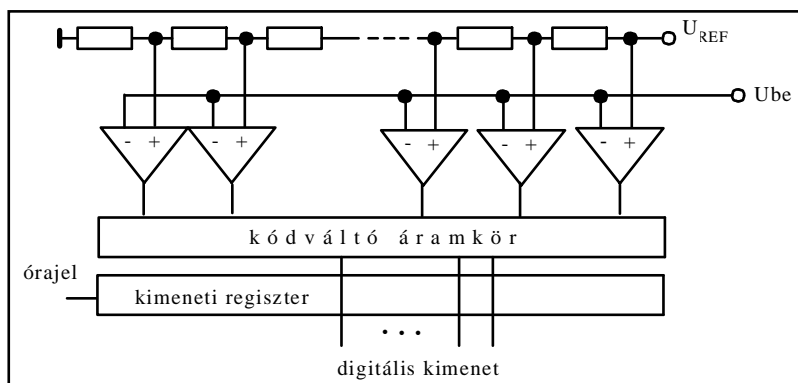
## 9.2. Az A/D átalakítók

Az A/D konverterek a bemenő analóg jel (többnyire egyenfeszültség) egyes sávjaihoz, tartományaihoz valamilyen digitális kódot rendelnek hozzá, többnyire a kimenet bitszáma által meghatározott tartományú kettes számrendszerű értéket. A táblázat bemutat egy lehetséges A/D konverziós adatsort:

Bemenőjel tartomány (V)	Kimenő kód
0 ... 0,0196	00000000
0,0196 ... 0,0392	00000001
...	...
4,9608 ... 4,9804	11111110
4,9804 ... 5	11111111

Az A/D konverter is lehet unipoláris (amilyenre a táblázat vonatkozik) vagy bipoláris, többnyire itt is egy referenciafeszültség határozza meg a bemenő jeltartományt.

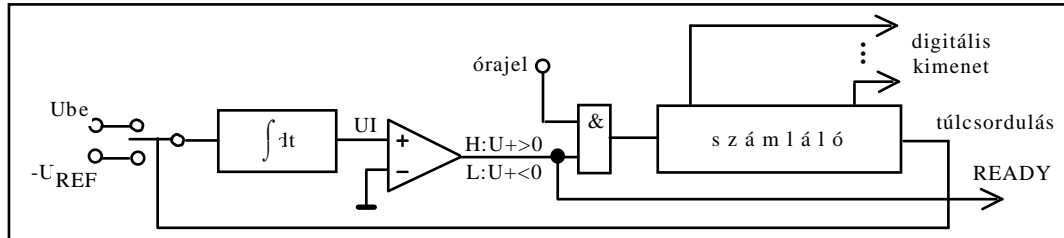
A leggyorsabb A/D konverter a *komparátor sorra épülő, ún. flash konverter*. Ebben tulajdonképpen a bemutatott táblázat valósul meg közvetlenül. Minden bemenőtartomány-határt egy konverter figyel, a bejövő jelet a konverterek összehasonlítják az általuk figyelt küszöbértékkel és így áll elő az elsődleges digitális érték. Annyi komparátort kell beépíteni, ahány tartományt, sávot meg kívánunk különböztetni a bemenő jeltartományon belül (9.9. ábra). A küszöbfeszültségeket a referenciafeszültségből az ellenállásosztó állítja elő. A bemenő jeltől függően a komparátorok egy része a küszöbértékéhez képest kis jelet érzékel, a többiek a küszöbértéküknél nagyobb jeleket. Ez az elsődleges digitális kód nagyon terjedős, ezért kódváltóval pl. kettes számrendszerbeli értékekké szokták alakítani.



9.9. ábra

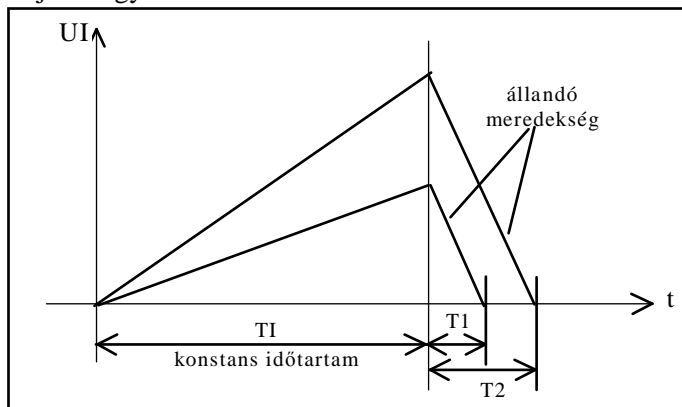
A flash konverter elvét már régen ismerték, építettek is ilyen átalakítókat, de a komparátor IC-kben legfeljebb négy összehasonlító egység volt - így csak kis felbontású átalakítók készülhettek. Mára már az integrálási technika fejlődése lehetővé tette, hogy 8-10-12-14 bites flash konverter chip-eket állítsanak elő.

Sokszor egy kimenőregisztert is kap a konverter, hogy szinkronizált működésre lehessen készíteni, a kimenetén akármikor ne változhasson meg a digitális érték. A flash konverter konverziós ideje ns nagyságrendű, így videojel digitalizálására is alkalmas. Az ipari használatánál mintavevő-tartó áramkör nem szükséges eléje a gyors működése miatt, a zavarok bemérését is megoldhatjuk vele, ezért a digitális zajszűréshez is kiváló megoldás.



9.10. ábra

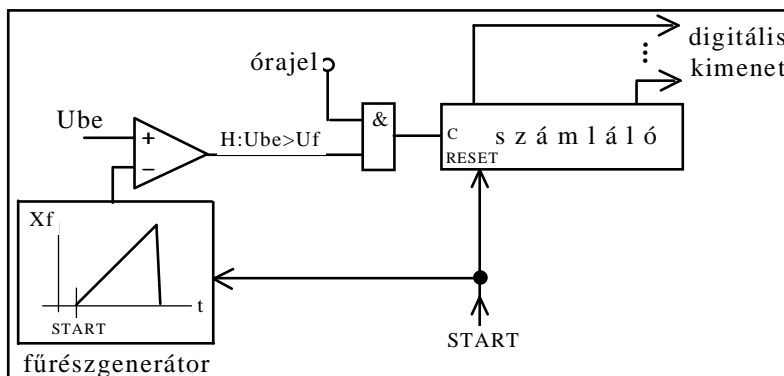
Az integráló (dual slope, két lejtős) A/D átalakító jellegzetes ipari konverter. A bemenőjelet egy meghatározott ideig ( $T_i$ ) integrálja, majd a kapott integrálértéket alakítja át digitális jellé (9.10. ábra). A  $T_i$  időt a számláló első túlszordulása jelzi, ekkor a bemenetre az analóg jel helyére egy negatív referencia feszültség kapcsolódik fel. Az integráló kimenőfeszültsége a negatív jel miatt idővel lineárisan csökken, a beintegrált jel nagyságától függő idő múlva az integráló kimenetén 0V jelentkezik. Az ehhez szükséges időt a számláló tartalma jelzi, ez a digitális kimenő jel is egyben.



9.11. ábra

A 9.11. ábrán követhető a működés. Az első lejtő meredeksége a bemenő jeltől függ, az integrálási időtartam azonban konstans. Ha kisebb a bemenő jel, az integrálási idő végére alacsonyabb feszültség jön létre az integrátor kimenetén, a magasabb bemenő jelhez nagyobb integrált érték tartozik. A második szakasznál a meredekség konstans, így a leintegrálás időtartama a bemenőjel függvénye.

Az integráló A/D-t olyan környezetben használják, ahol az analóg jelen az erősáramú hálózathoz eredő szinuszos alakú zavar jelenhet meg. Ha a  $T_i$  idő a hálózati feszültség periódus ideje, az integrálás a szinuszt kiejtí s



9.12. ábra

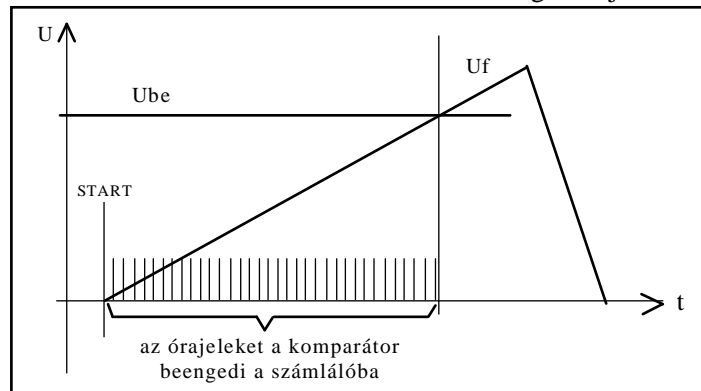
nem lehet.

Egy klasszikus A/D átalakító típus a fűrészgenerátorra épülő változat (9.12. ábra). A Konverzió Start parancs 0-ról indítja a számlálót s egyúttal aktivizálja a fűrészgenerátort is.

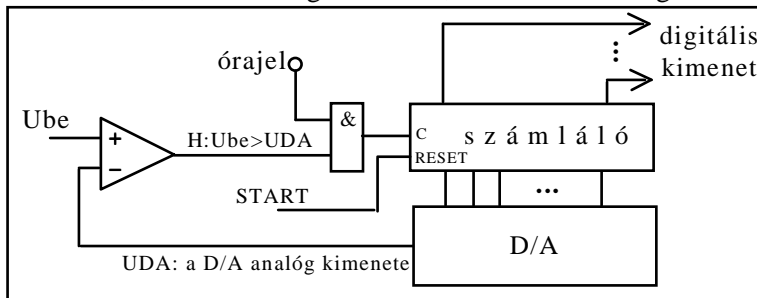
Az órajel frekvenciája és a fűrészgenerátor meredeksége egymással szinkronban vannak, s akkor éri el a fűrészjel a csúcspontját, mikor a számláló minden bitjén 1 áll. Ha a bemenő jel kisebb mint a lehetséges maximális érték, ahogy a fűrészjel meghaladja a bemenő szintet, a számlálás leáll, több órajel nem jut a számlálóba, a számláló tartalma a kimenő digitális jel.

A működést a **9.13. ábra** szemlélteti, ahol  $U_f$  a fűrészgenerátor jele. A konverter pontosságát sok tényező befolyásolja, pl. a fűrészgenerátor linearitása, a szinkronizáció a fűrész emelkedése és az órajelek között, az órajel frekvenciájának stabilitása stb. A konverziós idő általában 0,1 ... 10 ms körüli, a konverter előtt mintavevő-tartó áramkört kell használni.

A D/A-ra épülő A/D (**9.14. ábra**) közel azonos paraméterekkel rendelkezik, mint a fűrészgenerátoros, ez előtt is szükség van mintavevő-tartó áramkörre.



9.13. ábra

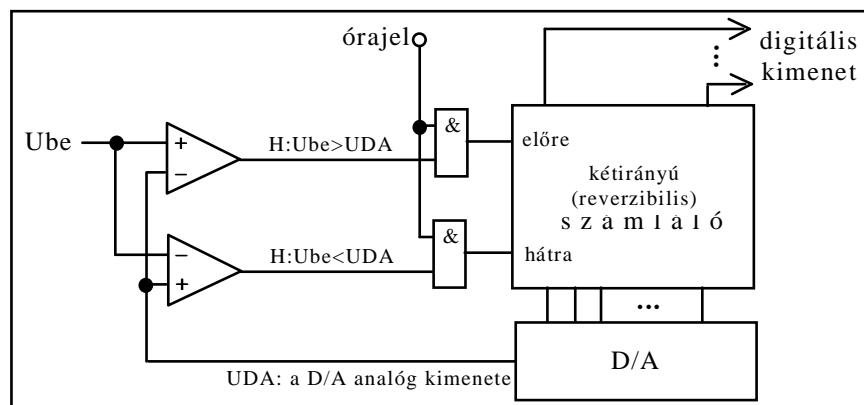


9.14. ábra

A Konverzió Start parancs itt is nulláról indít egy számlálót, de itt a számláló tartalmával arányos analóg jelet vezetjük vissza a bemeneti komparátorra, a beépített D/A segítségével folyamatosan össze lehet hasonlítani a számláló tartalmának megfelelő analóg értéket a bejövő jellel. Ha a számláló tartalma

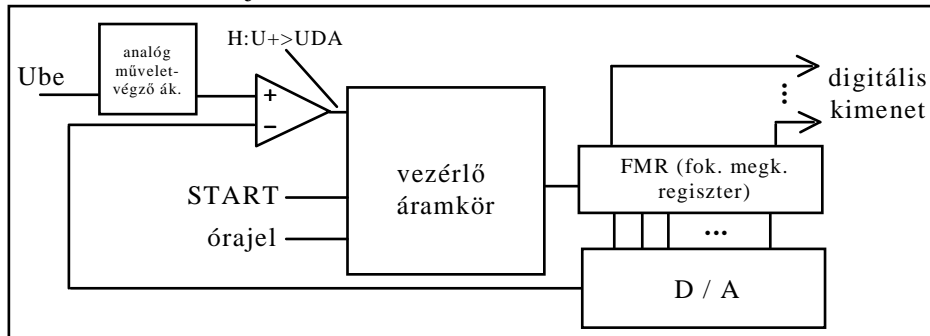
eléri a bemenetnek megfelelő értéket, az órajelek további bejutása tiltódik, a számláló tartalma a kimenő digitális jel. Ennél az átalakítónál sokkal kevesebb a pontosságot befolyásoló tényező, nincs szükség nagy frekvencia pontosságú órajelre sem, ezért kedvelt, olcsó megoldásnak számít.

A D/A-ra épülő A/D konverter átalakítható úgy, hogy két bemeneti komparátor segítségével egy kétirányú (reverzibilis) számlálót a bemenő jel változása esetén automatikusan (konverziós parancs nélkül) felfelé vagy lefelé tudjon léptetni (**9.15. ábra**). Ez elé a konverter elé mintavevő-tartó nem szükséges, viszont a bemeneti komparátort határozott hiszterézissel kell elkészíteni, különben a kisebb helyértékeken állandóan fel-le fog változni a kimenő jel.



9.15. ábra

A **9.14. ábrán** látott, D/A-ra épülő konverter a számlálót egyenként léptetve, inkrementálva jut el a kimeneti érték kialakításáig, ami (ha a bemenőjel a mérési tartomány tetején van) meglehetősen hosszú ideig tart. Ha a konverter 8 bites, a számláló teljes felléptetése 256 óraciklust jelent, ha 16 bites, 65536 órajelet.



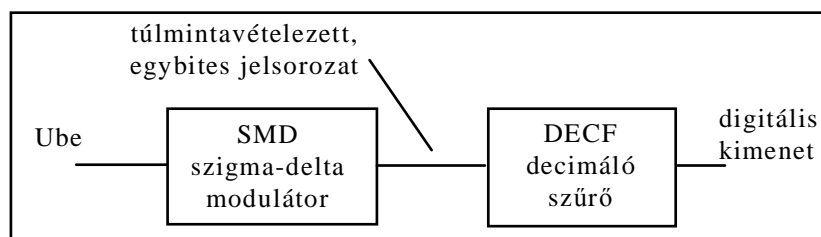
**9.16. ábra**

Ha az egyesével léptethető számláló helyett helyértékenként vezérelhető regisztert alkalmazunk (fokozatos megközelítésű regisztert, Successive Approximation Register, SAR) a konverzió annyi óraciklust igényel csak, ahány bites a konverter! A fokozatos megközelítésű átalakító blokkvázlata a **9.16. ábrán** látható.

A fokozatos megközelítésű konverter első lépésben a legfelső bitet vizsgálja meg, hogy 1-re vagy 0-ra kell-e állítani. A bit 1-re állításával a tartomány fele értékének megfelelő analóg jelet ad ki a D/A konverter - ha a bemenő jel ennél nagyobb, a bit 1 értékű lesz az eredményben, ha kisebb, akkor 0. Ezután a bemenő jelből le kell vonni a most beállított analóg értéket (azaz ha a bit 1 marad, akkor a fél értéket, ha a bit 0, akkor nem kell levonni semmit). Hasonló módon az összes bitet végigpróbálja a konverter, s mire elérkezik a legkisebb helyértékhez, már ismert is a digitális kimenőérték. Minden lépésben fele akkora küszöbszinttel kell összevetni a bemenő jelet, mint előzőleg. A pontosabb működés érdekében sokszor úgy módosítják a konvertert, hogy minden lépés után kétszeresen erősítik a bemenő jelet, így mindig azonos küszöbértékhez kell hasonlítani. Mivel a fokozatos megközelítésű konverter működése sokkal gyorsabb, mint a hagyományos D/A-ra épülő elődjéé, lassan kiszorítja azt az alkalmazásokból. A mintavevő-tartó áramkör azonban ez elé az áramkör elé is szükséges. (Gondoljuk meg, a 16 bites konverter 65536 óraciklus helyett 16 óraciklus alatt elvégzi a konverziót!)

A fokozatos megközelítésű konverterek új generációjában a D/A részletet nem ellenálláshálózattal, hanem kapcsolt kondenzátorokkal oldották meg. A kapcsolt kondenzátoros A/D konverterek CMOS IC-k, kis fogyasztással, nagy sebességgel. A konverzió első lépéseként a kondenzátorcsoport feltöltődik a bemenőjelnek megfelelő feszültségre majd lekapcsolódik a bemenetről, ezért a kondenzátoros konverter elé nem kell mintavevő-tartó áramkör, tulajdonképpen belső mintavételezéssel dolgozik. A CMOS technológia lehetővé teszi extra szolgáltatások beépítését is, pl. a kondenzátorokat apró trimmer kondenzátorokkal is kiegészíthetik, melyeket kalibráció során lehet be- illetve kikapcsolni, így a konverter pontossága fokozható. A kalibrációt sok esetben a konverter maga végzi el (autokalibrációs A/D konverter).

Sajátos működésű A/D konvertereket használ a CD technika - a szigma-delta modulátoros konvertereket ( $\Sigma\Delta$ ). A szigma-delta konverterek túlmintavételezett (oversampling) konverterek, ha a Shannon törvény szerinti előírt mintavételi frekvencia pl. 40 kHz, a szigma-delta modulátoros konverter több MHz-es mintavételezéssel dolgozik. A konverter első fokozata, a szigma-delta modulátor állítja elő ezt a túlmintavételezett, de egyetlen bitre kvantált elsődleges digitális jelet. Egy második fokozat, a decimáló szűrő a mintavételezési értéket a kívánt értékre változtatja meg, esetünkben 40 kHz-re, miközben a digitális jel bitszélessége megnő, általában 16 ... 24 bitre.



9.17. ábra

A szigma-delta modulátorra épülő A/D konverter (9.17. ábra) fő előnye, hogy szinte teljesen mentes a kvantálási zajtól, így Hi-Fi minőségű CD hanganyag készítésére alkalmas. Mivel ez a konverter lényegében egy digitális jelprocesszor, a kalibrációt is kényelmesen meg tudja oldani, többnyire különféle autokalibrációs megoldásokat is beépítenek ezekbe az áramkörökbe.

Az ipari célra gyártott A/D konverterek között is megjelentek már a szigma-delta modulátoros megoldások, s a ma gyártott D/A konverterek egy részében is megjelent a kapcsolt kondenzátor mező az ellenállások helyett.

Ha az analóg érték ellenállásváltozás, a mikroszámítógépekben úgy lehet az értéket digitalizálni, hogy egy pontos feszültségre feltöltött kondenzátort sütnék ki a mérendő ellenálláson keresztül. Komparátor áramkörökkel lehet figyelni a kisülés időtartamát s abból egyszerűen számítható az ellenállás értéke.

Végezetül megemlítnék egy sajátos megoldási lehetőséget az A/D átalakításra. Nagy pontosságú, széles jeltartományban működő feszültség-frekvencia konverterek, feszültség-vezérelt oszcillátorok szerezhetők be az IC piacon. Ezek különféle tartományú egyenfeszültséget fogadnak a bemenetükön, a kimenetük egy impulzus sorozat, melynek frekvenciája a bemenőfeszültséggel arányos. Sok változatot gyártanak, többnyire 0,1 KHz/V ... 10 KHz/V átalakítási tényezővel. Az A/D alkalmazáskor a mikroszámítógép pontos időintervallum alatt számlálja az impulzusokat s a kapott érték alapján az analóg jel, a feszültség értékét ismeri meg.

## 10. A mikrovezérlők

Amikor a 70-es évek második felében a mikroprocesszorok iránti kereslet egyre jelentősebbé vált, különféle továbbfejlesztési utakon próbáltak a gyártók új piacképes elemeket kialakítani. Az egyik próbálkozás célja az volt, hogy komplett mikroszámítógépeket helyezzenek el egy chip-en. A ma mikrovezérlőknek nevezett integrált áramkörök ennek a fejlesztési iránynak az eredményei.

Korábban egy-chip-es mikroszámítógépeknek nevezték ezeket az áramköröket, de a mikrovezérlő találóbb elnevezés. Igaz ugyan, hogy a mikrovezérlő komplett mikroszámítógép, mely a CPU mellett programtárat, adattárat, bemeneti és kimeneti egységeket is tartalmaz, de felépítése is, utasításkészlete is a vezérlési feladatokra optimalizált.

Az első egy-chip-es mikroszámítógépek kialakításakor még nem lehetett pontosan látni, milyen felhasználói kör fogja elsősorban használni ezeket az elemeket, de hamarosan kiderült, hogy a különféle ipari, fogyasztói készülékek beépített (embedded) vezérlőegységeinek kialakítására használhatók fel elsősorban. Mit jelent az, hogy vezérlésre orientált eszközök?

Jellemző a mikrovezérlőre a sokféle bit-orientált művelet, a bit-kezelő utasítások, bit-címezzhető memória területek, mindezek a vezérlési feladatoknál hasznosíthatók. A készülékek, amelybe beépítik ezeket, sokszor a hálózattól függetlenül is üzemelnek, ezért fontos a kis fogyasztás, a stand-by állapot kialakítása, természetesen adatvesztés nélkül. A berendezések karbantartásakor, szervizeléskor, beállításakor célszerű, ha könnyen csatlakoztathatók a vizsgáló, mérő, beállító egységhez, ezért sokoldalú soros és párhuzamos Port megoldásokat alkalmaztak bennük.

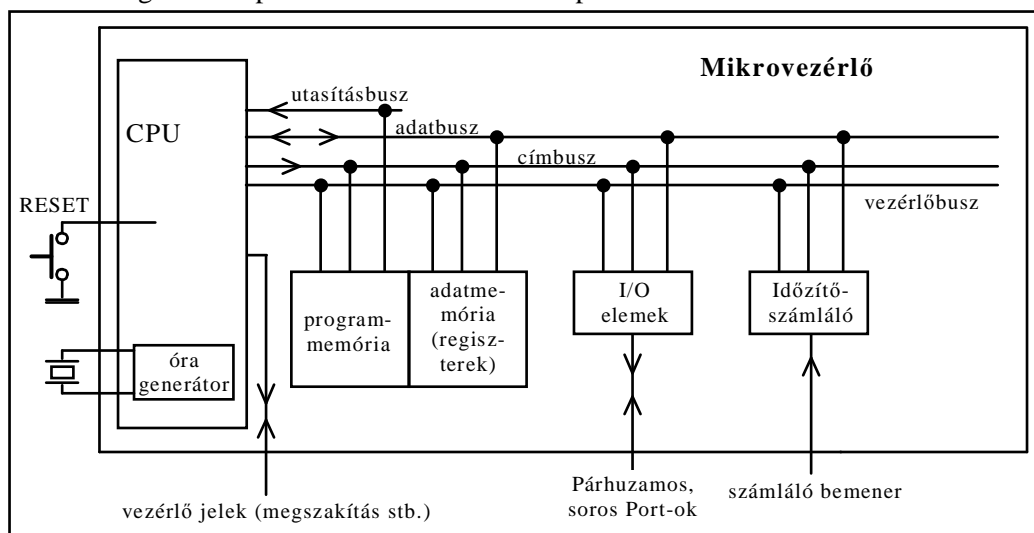
A mikrovezérlő tehát nem általános célú mikroszámítógép, hanem elsősorban beépített, fedélzeti mikroszámítógépek alkotó eleme. Mikrovezérlő működteti az egeret, a billentyűzetet, a mobiltelefont, a telefax készüléket, a mosógépet, a mikrohullámú sütőt, de sokszor a közeledés-érzékelőt és a kézszárítót is. Időközben az általános alkalmazhatóságú mikroprocesszorok is hatalmasat fejlődtek, az egyre kisebb méretű számítógépek (lap-top, notebook, palm-top) egyre komplexebb Chip-Set készletek jelennek meg, a teljes alaplap megvalósítható két vagy három IC-vel, mégis folytatódik a mikrovezérlők gyártása és fejlesztése, mert azok más alkalmazási területre használhatók fel.

Érdemes néhány szót szólni még a már említett egyik mikrovezérlő-alkalmazásról, a PC egerről. Vegyünk kezünkbe egy hagyományos, görgős egeret! Ha szétszedjük, az egerben általában egyetlen mikrovezérlőt találunk, nincsenek külső kiegészítő áramköri elemei. A mikrovezérlőkkel foglalkozó szakemberek szerint akkor hatékony egy mikrovezérlővel megépített eszköz, ha abban a mikrovezérlőn kívül egyéb elektronikus elem nincs! Az egerünkben az egérgolyó két irányú mozgását optikai jeladók érzékelik, ezek jelei közvetlenül a mikrovezérlőbe jutnak, s az állítja elő az RS 232 C jellegű soros jelet a PC számára (természetesen közben az egér gombjaival működtetett kapcsolókat is kezeli). Az egér különlegessége az, hogy látszólag nincs tápellátása! Az egerben nincs elem vagy akkumulátor, az IBM PC soros Port-ról pedig tudni való, hogy nincs rajta tápfeszültség kivezetve a számítógépből! Az egerben mindig kis fogyasztású CMOS mikrovezérlő változatot használnak, s a tápellátását, jellegzetes módon, a soros Port egyik kimenőjének H szintje biztosítja!

A mikrovezérlő általános felépítését a **10.1. ábra** mutatja be. A mikrovezérlő nem a Neumann elv szerint használja a memóriákat, hanem a Harvard architektúra szerint, azaz teljesen elkülönített programmemóriát és adatmemóriát tartalmaz. A programmemóriából csak utasítás-elemeket tud kiolvasni, az adatmemóriát pedig többnyire egyszerű utasításokkal csak adatbeírásra vagy adat-visszaolvasásra tudja használni. A Harvard-kialakítás jelentősen leegyszerűsíti a belső szerkezetet, az egyik memóriaterület kizárólag az utasítástárolóval van kapcsolatban, a másik csak az ALU-val.

A mikroprocesszor esetében az egységes memória mindkét elemmel összekapcsolódhat, amit egy összetett multiplexer/demultiplexer rendszer tesz lehetővé, ez a mikrovezérlőből kimaradhatott.

A mikrovezérlő buszrendszere belső buszrendszer, a felhasználó számára nem érhető el (zárt felépítés). A mikrovezérlők egyes csoportjai képesek külső memória kezelésére, de ezeket csak a Port pontjaikon keresztül lehet hozzájuk kapcsolni. Ennek az a hátrányos következménye, hogy a feladat megoldására felhasználható Port pontok száma jelentősen lecsökken. A mikrovezérlők a külső memóriákat is a Harvard elvnek megfelelően használják, két teljesen elkülönülő vezérlőjel csoporttal kezelik a külső programmemóriát és a külső adatmemóriát. A külső memória által lekötött Port pontok csökkentése érdekében ezek a memória-csatlakoztatások mindig busz-multiplexeléssel történnek, a 8085A-hoz hasonló módon az adatvonalak közösek a címvonalak alsó bitjeivel. Ennek ellenére egy külső memória csatlakoztatása legalább egy 8 bites párhuzamos Port-ot leköt, ha a címtartománya 8 bitnél nagyobb, egy második port pontjaira is szükségünk lesz a használata során. A probléma egyik lehetséges megoldása az, hogy az utóbbi években erőteljes fejlesztéssel kialakított, soros adatelérésű memória IC-eket helyeznek a mikrovezérlők mellé, így csak 2 vagy 3 Port pontot köt le még a legnagyobb kapacitású külső memória is. Más mikrovezérlő-gyártók azt a filozófiát követik, hogy minden vezérlési feladathoz meg kell keresni azt a mikrovezérlőt, amelyik külső áramköri segítség nélkül meg tudja oldani a feladatot. Tulajdonképpen igazuk van, az ezernyi mikrovezérlő változatból valószínűleg minden problémához lehet találni optimális áramkört!



10.1. ábra

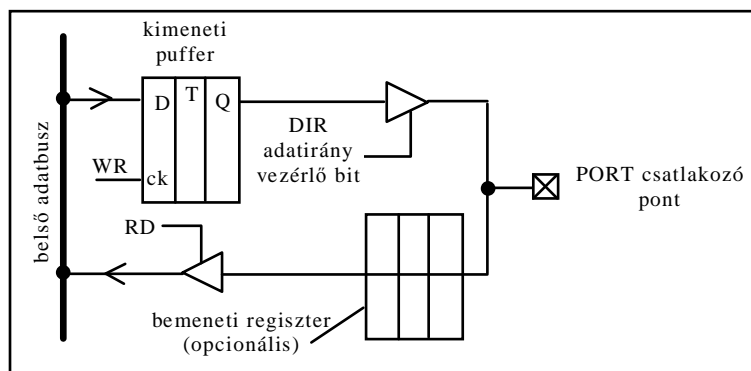
A mikrovezérlőket családokban gyártják. A család tagjainak azonos a CPU kialakítása, de sok változatban készülnek. Különböző memória paraméterekkel lehet vásárolni családtagokat, ezáltal a feladathoz legjobban illeszkedő elemet lehet felhasználni. A különféle programmemória-jelleg pedig a fejlesztés során tesz jó szolgálatot.

A fejlesztés kiszolgálására különféle mikrovezérlő család-stratégiák léteznek. Az Intel a családtagokat EPROM és ROM programtárral gyártja, arra biztatva, hogy a felhasználó a program fejlesztés fázisában használja az EPROM-os mikrovezérlőket, ha a program kész, küldje be az Intelhez, ott maszk-programozással a megfelelő darabszámú, ROM memóriás áramkörbe betöltik ezt a programot. (Azoknak, akik ezt nem tudnák megfizetni, belső programtár nélküli változatokat is készít az Intel, ezek mellett, külső programmemóriaként hagyományos EPROM-ot lehet használni.) Más mikrovezérlő-gyártók is átvették ezt a megoldást, és EPROM-os valamint ROM-os változatokat készítenek. Néha azonban érik meglepetések is a felhasználót, mert a ROM-ot tartalmazó chip mégsem teljesen egyezik meg az EPROM-mal készített áramkörrel! Előfordul, hogy szélsőséges tápfeszültségen, szélsőséges hőmérsékleten, szélsőséges működési frekvencián különböző módon dolgoznak.



A ZILOG az egyik mikrovezérlőjénél Piggy-Back megoldást használ. A mikrovezérlő tokozásának hátoldalán elhelyeztek egy EPROM foglalatot, így hagyományos EPROM-okkal lehet programot fejleszteni, ugyanaz az áramkör a fejlesztő elem és a végleges mikrovezérlő, de ez a megoldás igen költséges!

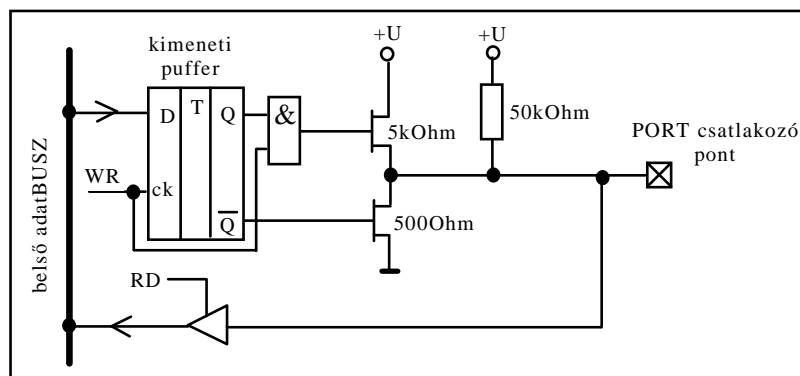
A Microchip a PIC mikrovezérlőket egységes kivitelben készíti. Ezek CMOS EPROM chip-ek, a programfejlesztéshez az ablakos változatot kell megvenni. Ha kész a program, az OTP kivitelű, zárt műanyag tokozású áramkörökbe kell betölteni - ezek ára törtrésze a kvarcablakos, fémkerámia tokozású változatok árának. Mivel ugyanaz a chip van a kvarcablakos és a zárt műanyagból készült tokozásban, az azonos működés is garantált! A Microchipnek volt egy kísérlete, az EEPROM alapú mikrovezérlő! A villamos törlés leegyszerűsítette a programfejlesztési munkát. Mivel egy EEPROM programtárat is kialakítottak benne, ez a mikrovezérlő a működése során szerzett adatokat úgy tudja tárolni, hogy azok a tápenergia teljes megszűnése esetén sem vesznek el! Az alapötlet jó volt, de a továbbiakban Flash jellegű programmemóriát használtak, ma minden gyártó ezt gyártja.



10.2. ábra

A mikrovezérlők néhány típusánál nincs megszakítási lehetőség, de azért többnyire a sokoldalú megszakítási rendszer a jellemző, nagyszámú külső megszakítási lehetőséggel és belső megszakítást kezdeményező eseménnyel. A megszakítások rögzített címről indítanak szubrutint, nem vektoros jellegűek, és többnyire teljes körűen maszkolhatók.

Mivel a buszrendszer nincs kivezetve, minden külső eszköz Portokon át csatlakozik a mikrovezérlőkhöz. A memóriák, A/D és D/A átalakítók, egyéb áramkörök két-három port minden csatlakozópontját is lekötik. A mikrovezérlők párhuzamos Port-jai általában kétirányúak, de nem mindig valódi kétirányúak. A valódi kétirányú Port-ot arról lehet felismerni, hogy adatirány-regiszter tartozik hozzá. A Port adatirányát vagy globálisan, vagy bitenként lehet megadni. Egy bit kialakítása a 10.2. ábrán látható, a DIR az adatirányt vezérlő bit.



10.3. ábra

A kvázi kétirányú Port felépítése jóval egyszerűbb (10.3. ábra). A kimeneti pufferben valamilyen tárolt érték mindig jelen van.

Ha a tárolt érték 0, a kis impedanciás FET egy kemény földeléssel egy erőteljes logikai 0-t állít elő a Port kimenő pontján. Ha a tárolóban 1 áll, a Port csatlakozópont logikai szintjét az 50 kOhm-os felhúzó ellenállás állítja be 1 értékre. Ez a magas szint nagyon könnyen megváltoztatható kívülről – lényegében ez a kvázi kétirányú Port bemeneti állapota. Egy külső áramkör a Port csatlakozó pontot könnyedén a saját logikai értékére tudja változtatni ilyenkor. Ha tehát egy kvázi kétirányú Port-ot bemenetként szeretnénk használni, akkor nem egy adatirány regiszterben kell ezt elrendelni, hanem a Port kimeneti regiszterébe logikai 1-t kell beírni.

A nagy értékű felhúzó ellenállás miatt, ha kimenetként használjuk a Port-ot, a  $0 \rightarrow 1$  átmenet esetén a jelváltozás elnyúlik, a jelalakok legömbölyödnek (a terhelő kapacitások a nagy ellenálláson át lassan töltődnek fel). Ezen segít a kimeneti puffer Q kimeneténél lévő ÉS kapu, az 1 érték beírásakor (a beíró impulzus idejéig) egy kisebb csatorna ellenállású FET segítségével a terhelő kapacitásokat gyorsan áttölti, mielőtt az 50 kOhm-os lágy H szint létrejön.

A kommunikációs lehetőségek terén a mikrovezérlők ma már igen sok lehetőséget nyújtanak. A hagyományos (párhuzamos Port illetve RS 232 C soros Port) megoldások mellett egyre újabb lehetőségeket is beépítenek. Az ipari szabvánnyá vált korszerű szinkron átviteli megoldások (I<sup>2</sup>C, Microwire stb.) mind megjelentek már a mikrovezérlőknél is.

A mikrovezérlők veremtára mindig belső verem, hiszen külső memória nélkül is működniük kell. Emiatt a mikrovezérlő verme többnyire nem túlságosan sok cím befogadására alkalmas. Az utasítás készletben általában nincsenek közvetlen verem felhasználó utasítások, a vermet a legtöbbször csak a CALL, RETURN utasítások használják címkezésre.

A programozható belső elemek kezelésére két megoldást alkalmaznak a mikrovezérlők. Az egyik a közvetlen utasításos kezelés, amikor minden működési eseményt egy-egy utasítás vált ki. Ez egyrészt terjedelmessé teszi az utasítás készletet, másrészt ha a mikrovezérlő chip-et továbbfejlesztik, új elemekkel egészítik ki, mindig változik az utasítás készlet - változik a fejlesztő rendszer is.

A másik lehetőség az, hogy az adatmemóriában kialakítanak egy regiszter mezőt (SFR, speciális funkciójú regiszterek területe). A programozható belső egységeket úgy szervezik meg, mint a mikroprocesszorok programozható kiegészítő elemeit, parancsszó regiszterekkel. Az SFR regiszterek lényegében a programozható elemek parancsszó tárolói. Az utasításkészletben pedig csak két utasítás kell: regiszter írás, regiszter olvasás. Ha bővül a mikrovezérlő, újabb SFR regiszterek jelennek meg - az utasítás készlet nem változik.

Láttuk az előző fejezetben, hogy az ipari irányítási feladatokban gyakori az, hogy analóg jelet kell fogadni illetve analóg kimenő jelet kell előállítani. Eleinte a mikrovezérlők mellett külső konvertereket kellett felhasználni, de az újabb fejlesztésű mikrovezérlő családok egyes tagjaiban már megtalálható az A/D átalakító, általában sokcsatornás analóg multiplexerrel együtt, s az analóg jelek előállítása érdekében a PWM kimeneti egység. Egyik-másik, az analóg jelkezelésre különösen alkalmassá tett mikrovezérlőt egyenesen analóg jelprocesszorként hirdeti gyártója.

A mikrovezérlők a berendezésbe rejtetten, felügyelet, operátori beavatkozási lehetőség nélkül végzik munkájukat. Mivel különféle erősáramú szerkezetek, nagyfeszültségű egységek is működhetnek a közelükben, a leggondosabb kivitelezés ellenére is előfordulhat, hogy egy zavaró jel, egy zajimpulzus olyan módon hat a mikrovezérlővel felépített mikroszámítógépre, hogy az kiesik a normál üzemből, a programja lefagy vagy végtelen ciklusba keveredik. Az ilyen helyzetekre is felkészíthetik a mikrovezérlőket. Az egyik lehetséges megoldás a Watch-Dog Timer (WDT) alkalmazása. Egy felfelé számláló egység alkotja a WDT-t, amit a normálisan futó mikrovezérlő szoftvere rendszeresen töröl. Ha a mikrovezérlő programhiba miatt nem törli az áramkört, az eléri a maximális számértékét s ebben a helyzetben általában Reset folyamatot vált ki. A szoftvert úgy kell elkészíteni, hogy a Reset folyamat után mindenképpen helyesen el tudjon indulni, így a mikrovezérlős áramkör tulajdonképpen öngyógyító funkcióval is felruházható.

A mikrovezérlők oszcillátora, óragenerátora belső áramkör, de a frekvenciáját külső alkatrészekkel (frekvencia-meghatározó elemekkel) lehet beállítani. Egyes mikrovezérlők több különféle módon is lehetővé teszik az órajelek előállítását, pl. RC-tagokkal, rezgőkvarccal, külső órajellel. A mai mikrovezérlőknél sokszor a belső órajel-áramköri részletek át is alakíthatóak az ún. konfigurációs bitek megfelelő beállításával. A konfigurációs biteket a futó program nem tudja elérni, azokat a programozó-készülék, a fejlesztő-rendszer képes beállítani, a program betöltésével egyidejűleg.

A Reset folyamat kialakítására általában nagy hangsúlyt fektetnek a gyártók, mivel gyakran magára hagyottan kell a mikrovezérlőnek a Reset után indulnia. Sokszor az oszcillátor órajelének stabil kialakulásáig nem is engedik működni a mikrovezérlőt, nehogy az első, tökéletlen óraimpulzusok miatt hibásan kezdődjön meg a program végrehajtása. Ha a Reset

Az elemes, akkumulátoros készülékeknél létfontosságú, hogy takarékoskodni lehessen a tápenergiával. A mikrovezérlők ezért (szoftver úton) stand-by állapotba helyezhetők. Abban nagy különbségek vannak, hogy egy-egy típusnak hányféle ilyen csökkentett tápenergiájú állapota van, hogyan lehet ezekbe az állapotokba vezérelni az áramköröket és hogyan lehet onnan kihozni a mikrovezérlőt. Igen jól használható az a megoldás, ahol megszakítással is feléleszthető a mikrovezérlő, vagy ahol a bemenetére kapcsolt billentyűzet megérintése éleszti fel.

A mikrovezérlő hatékonyságát fokozza, ha van olyan stand-by állapota, amelyben nem teljesen tétlen. Megoldható pl. hogy a WDT, a soros Port, egy univerzális időzítő/számláló működjön a stand-by állapotban, s ekkor a minimális tápáram felvétel ellenére egyes műveleteket még el tud végezni a mikrovezérlő. Ennek többnyire az a feltétele, hogy ezek a részegységek saját, független belső oszcillátorral legyenek kialakítva, mert a mikrovezérlő alap oszcillátorát a stand-by üzemben mindenképpen le kell állítani (ha az alap oszcillátor működik, a felvett tápenergia mindig jelentős értékű!).

A mai mikrovezérlőknél már sokoldalú védelmi rendszert alakítanak ki. Változatosak a figyelt hatások és paraméterek, egyes típusok képesek felismerni a túlmelegedést, a tápfeszültség növekedését vagy csökkenését, a verem helytelen használatát, azt, ha olyan területről kellene programkódot kiolvasni, ahova nem töltöttünk be programot stb. Ezeket a helyzeteket megszakítással illetve Reset folyamattal tudja megoldani a mikrovezérlő.

A mikrovezérlő programját fejlesztőrendszerek alkalmazásával lehet kialakítani. Egy-egy gyártó vagy vele szerződéses kapcsolatban lévő partnere a gyártó minden mikrovezérlőjéhez használható univerzális fejlesztőrendszert is ajánlhat, de van olyan cég is, amelyik minden családkhoz más-más fejlesztőegységet készít.

A hagyományos programozási megoldáskor úgy lehet beírni a kódokat a mikrovezérlő programtárolójába, ahogyan a memória-áramkörökbe is, tehát a cím és az adat párhuzamosan rá vezetődik az áramkörre, programozó-feszültség jelenlétében, a betöltést programozó-impulzus végzi el.

A programozás korszerű megoldása a soros, felhasználói környezetben történő megoldás. A mikrovezérlőt beforraszthatjuk a felhasználói áramkörbe, s ebben a helyzetben két-három pontjára rácsatlakozva elvégezhető a programozás.

Mivel ezzel a csatlakozási megoldással a felhasználói rendszer működéséről is információkat lehet szerezni, többnyire univerzális fejlesztő és hibakereső rendszerek készülnek. Ezek használata során a betöltött program helyes működése azonnal le is tesztelhető.

A mikrovezérlő programja általában a mikrovezérlő belső programmemóriájában van. Alapállapotban ezt a program memóriát a mikrovezérlő fejlesztő berendezése képes kiolvasni. A mikrovezérlők egy részénél ezt a kiolvasást a felhasználó meggátolhatja, titkosító bitek beprogramozásával. Különösen nagy a titkosítási lehetőség jelentősége az EPROM alapú mikrovezérlőknél, így pl. a Microchip PIC áramköreinél. Ugyanakkor a titkosítás következtében arra sem lesz mód, hogy a tartalmat összehasonlítsuk egy etalonnal. Sok esetben ezért a titkosítást egy titkosító táblázattal oldják meg. A program-memória tartalma a táblázat tartalmával történő bonyolult módosítások után lép ki az áramkörből, de a felhasználó, aki a titkosító táblát betöltötte, a saját fejlesztő rendszere számára is megadhatja a titkosító tábla tartalmát s ezután a

fejlesztő rendszer már helyre tudja állítani az információt. A titkosító táblát tartalmazó mikrovezérlőkben ez a tábla nem olvasható, így illetéktelen semmiképpen nem ismerheti meg a tartalmát.

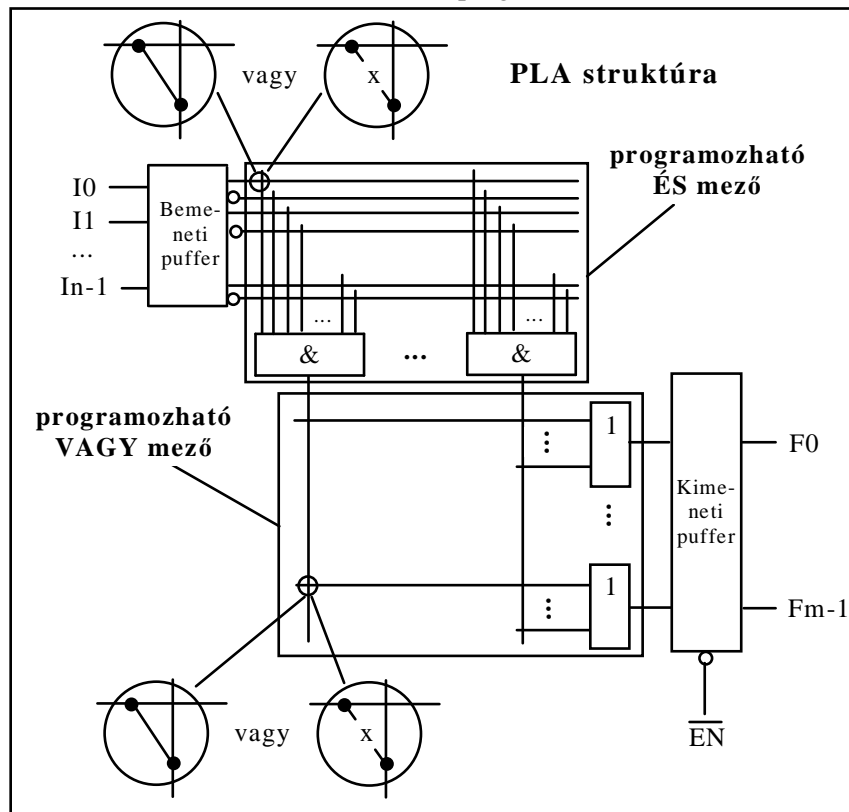
A mikrovezérlők periféria-készletét folyamatosan fejlesztik. Általában ha pl. egy új csatolófelület, buszmegoldás megjelenik a számítógépes világban, az hamarosan a mikrovezérlőkben is megjelenik. Így pl. korábban nem volt USB Port a mikrovezérlőkben, ma pedig a legtöbbnek van ilyen illesztőegysége.

A mikrovezérlők egy-egy népszerű családjával önálló jegyzetekben ismerkedünk majd meg.

## 11. A PLD áramkörök

Az LSI áramkörök felosztásakor már láttuk, hogy a PLD áramkör a felhasználó által beprogramozható logikai eszköz (Programmable Logic Device). A PLD-k előgyártmányra épülő, két-fázisú gyártással készülő logikai áramkörök, a második fázis a beprogramozás, ami a mi szóhasználatunk szerint hard-programozás.

A PLD-k őse a PLA, mely logikai felépítés szempontjából is kiinduló pontnak tekinthető és programozási megoldásai alapján is. A logikai elrendezése: klasszikus kétszintes kapuhálózat, az alsó szinten ÉS kapukkal, a felsőn VAGY kapukkal. A PLA-ban mindkét kaputerület programozható (**11.1. ábra**). Az  $n$  bemenetet a bemeneti puffer fogadja, előállítja a pozitív és negált jeleket, bevezeti azokat az ÉS mezőbe. A bemenő jelek és az ÉS kapuk kapcsolatai programozhatóak. Az ÉS kapuk száma a bemenetek számához viszonyítva igen alacsony (pl. 16 bemenet, 48 ÉS kapu). Az ÉS kapuk kimenetei egy második vezeték hálóra kerülnek, ahol a VAGY kapukra lehet rávezetni ezeket. Ez a VAGY mező is programozható.



11.1. ábra

Az eredeti PLA maszk-programozható elem volt, az ÉS mező és a VAGY mező kialakítására vonatkozó információ alapján a félvezető-gyártó fémmező maszkot készített és a chip felületén az alumínium összekötéseket ennek megfelelően hozta létre. Később jelent meg a felhasználó által programozható PLA, amit akkor FPLA-nak neveztek. Mivel mára már csak ezek a változatok maradtak használatban, az F betű le is kopott a megnevezésből. Ezek az első FPLA elemek bipoláris áramkörök voltak, olvadóbiztosítékos programozási lehetőséggel. A következő években a PLD fejlesztés felgyorsult, egyre újabb programozási eljárások jelentek meg és a logikai struktúrák is változtak.

A **11.1. ábrán** láttuk, hogy a PLD programozás lényegében abból áll, hogy kereszteződő vezetékeknél a kívánt logikai hálózatnak megfelelően vagy összekötjük a vezetékeket vagy nem.

A PLD programozás szempontjából kétféle lehet. Az egyikben a félkész előgyártmányban minden összekötés jelen van, a programozáskor a feleslegeseket meg kell szüntetni, ki kell égetni.

A másik lehetőség az, ha az új áramkörben nincsenek összekötve a keresztpontok, s a programozáskor, ahol szükséges, létre lehet hozni a galvanikus kapcsolatot. A leggyakoribb programozási technikákat a **11.2. ábra** foglalja össze.

	programozatlanul	programozva
a, biztosíték (fuse)	ép biztosíték 	kiégetett 
b, AIM	két, szembekötött dióda 	az egyik dióda Al atomokkal elárasztva 
c, EPROM FET	üres GATE, nincs csatorna eltemetett GATE vezérelt GATE 	feltöltött GATE, felépült csatorna eltemetett GATE vezérelt GATE 
d, EEPROM FET	üres GATE, nincs csatorna szendvics GATE vezérelt GATE 	feltöltött GATE, felépült csatorna szendvics GATE vezérelt GATE 
e, ANTIFUSE	dielektrikum (kb. 100 MOhm) oxid Si oxid 	vezető (néhány 100 Ohm) oxid Si oxid 
f, SRAM vezérlésű	lezárt FET, nincs csatorna 	nyitott FET, felépült csatorna 

11.2. ábra

*a, Olvadó biztosíték (fuse)*

NiCr vagy TiW anyagú, elvékonyított átkötés, jelentős programozó árammal lehet kiolvasztani, csak bipoláris chip-en alkalmazzák, nagy a helyigénye: bitenként programozható, az eredeti állapot nem állítható helyre. Korlátlan ideig őrzi az információt, nem illanó.

*b, Lavinaszzerű indukált anyagáram (AIM, Avalanche Induced Migration)*

A két dióda szembekapcsolása miatt nincs galvanikus kapcsolat, programozáskor az egyiket Al atomokkal árasztják el, megszűnik a pn átmenet, egy dióda marad. Bipoláris chip-en alkalmazzák, kicsi a helyigénye, az eredeti állapot nem állítható helyre. Korlátlan ideig őrzi a beprogramozott információt, nem illanó.

*c, EPROM jellegű, eltemetett GATE-es FET tranzisztor*

Üres GATE esetén nincs galvanikus kapcsolat, ha programozáskor a nagyobb értékű programozó feszültség hatására a szigetelő réteg átüt, a töltések bejutnak az eltemetett GATE elektródára, a csatorna felépül, létrejön a galvanikus kapcsolat. Ha a chip tokozásán van kvarc ablak, törölhető, az eredeti állapot visszaállítható. Az információ megőrzés 10 évre garantált, nem illanó.

*d, EEPROM jellegű, szendvics GATE-es FET tranzisztor*

Üres GATE esetén nincs galvanikus kapcsolat, ha a programozó feszültség hatására a GATE körül a vezetőképesség megnő, a GATE feltölthető, kisüthető, így elektromosan programozható, törölhető az áramkör. Az információ megőrzés ideje a törlési/programozási ciklusok számától függ, nem illanó.

*e, ANTIFUSE*

Programozás nélkül a két vezető sáv között dielektrikum van, szakadás. Programozáskor a dielektrikum átalakul vezető réteggé, létrejön a galvanikus kapcsolat. Kis helyigényű megoldás, az eredeti állapotot nem lehet helyreállítani. Az információ megőrzés időtartama korlátlan, nem illanó.

*f, SRAM vezérlésű keresztfonti kapcsoló FET tranzisztor*

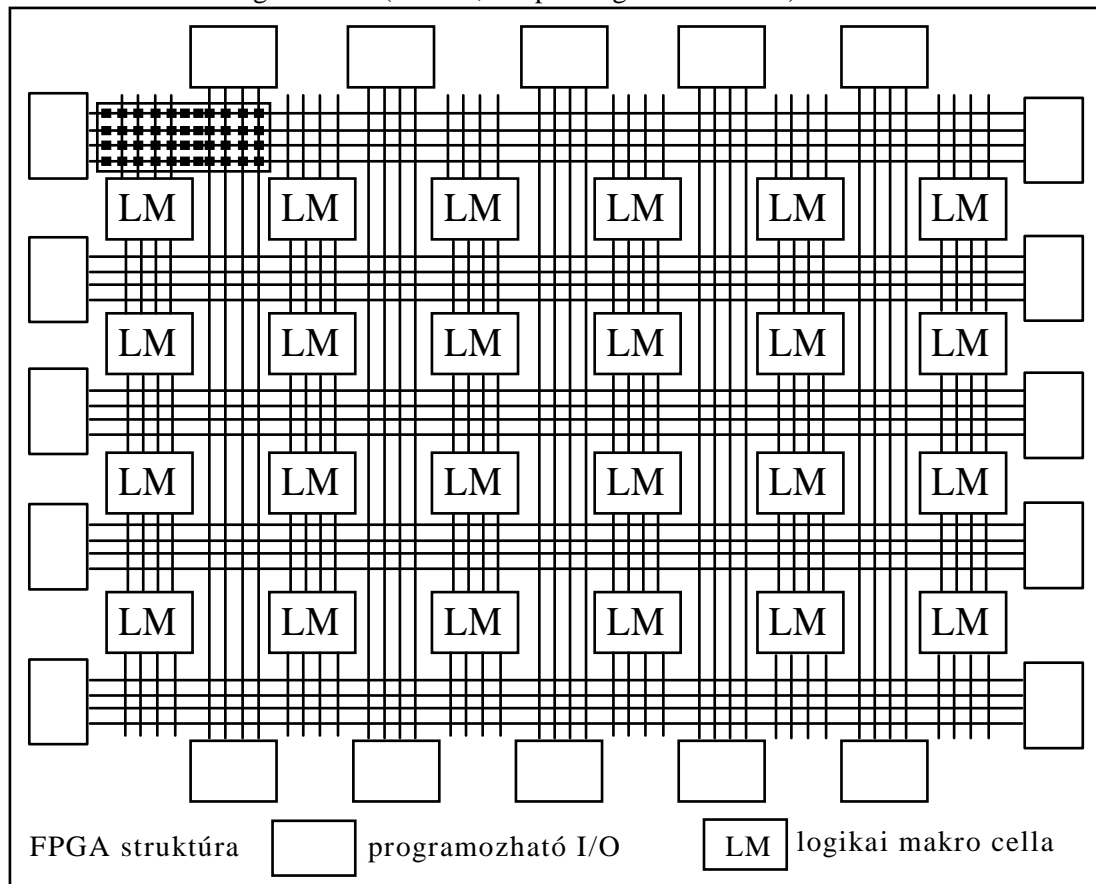
A SRAM tartalma vezérli a kapcsoló tranzisztorokat. Tetszőleges sokszor, tetszőleges gyakorisággal módosítható a tartalom. Illanó megoldás.

A logikai struktúrák is fejlődnek, néhány jelentősebbet bemutatunk az alábbiakban.

PLA programozható ÉS mező, programozható VAGY mező.

PAL programozható ÉS mező, fix VAGY mező; egyszerű kimeneti egységek (ponáló puffer, negáló puffer, D tároló).

GAL programozható ÉS mező, rögzített VAGY mező, összetett programozható kimeneti logikai cella (OLMC, Output Logic Macro Cell).



11.3. ábra

Az összetettebb logikai struktúrájú PLD jellegzetes képviselője a logikai modulokból (LM, Logic Modul) építkező FPGA (11.3. ábra). Az LM logikai modul többszáz tranzisztor-funkcióval kialakított, programozható részlet kapukkal, multiplexerekkel, tárolókkal. A chip felületén vezető sávok láthatók, melyek egy része globális, a teljes felületen áthalad, más részük lokális, a legközelebbi logikai modulnál ered.

A kereszteződések programozottan összeköthetők. Az ábra szaggatott vonallal bekerített részén a programozható keresztpontokat be is rajzoltuk. A globális vezetékek érdekessége, hogy programozottan szakaszolhatóak is, így rendkívül rugalmasan alakítható a logikai struktúra. A bemutatott logikai makrocellás elrendezést a XILINX SRAM programozható változatban kezdte gyártani, a Texas Instruments ANTIFUSE technikával. Mára már a XILINX is ajánl ANTIFUSE FPGA chipeket.



## Tartalomjegyzék

Bevezetés az 1998. évi első kiadáshoz.....	3. o.
Bevezetés a 2009. évi átdolgozott kiadáshoz.....	4. o.
1. A digitális áramkörök felépítése .....	5. o.
1.1. Alapanyag és technológia – néhány szó a chippek gyártásáról .....	5. o.
1.2. A tokozás változásai, szerelési megoldások.....	6. o.
1.3. A tápfeszültség változásai .....	7. o.
2. Az LSI áramkörök áttekintése.....	9. o.
3. A mikroszámítógép általános felépítése, működése.....	13. o.
3.1. A mikroprocesszoros mikroszámítógép felépítése.....	13. o.
3.2. A program és az adatok elhelyezkedése a memóriában .....	15. o.
3.3. Az utasításvégrehajtás menete .....	17. o.
3.4. A PC, a legnépszerűbb mikroszámítógép .....	19. o.
3.5. A szinkron és az aszinkron mikroprocesszorok .....	22. o.
4. Memória áramkörök.....	23. o.
4.1. ROM változatok.....	25. o.
4.2. RAM változatok.....	26. o.
4.3. Az írható/olvasható, nem illanó félvezetős memória megvalósítása.....	33. o.
4.4. Soros hozzáférésű félvezetős memóriák .....	33. o.
4.5. Nem címkijelöléssel működő memóriák.....	35. o.
4.6. A soros adatkezelésű memóriák.....	35. o.
5. A mikroprocesszorok felépítése, működése.....	37. o.
5.1. Az Intel 8080A mikroprocesszor .....	37. o.
5.2. Az Intel 8085A mikroprocesszor .....	41. o.
5.3. A ZILOG Z80-CPU .....	44. o.
5.4. A MOTOROLA MC6800 mikroprocesszor .....	48. o.
5.5. A nagyobb szóhosszúságú Intel mikroprocesszorok első változatai.....	49. o.
5.5.1. Az Intel 8086/8088 mikroprocesszorok .....	50. o.
5.5.2. Az Intel 80286 mikroprocesszor .....	56. o.
6. Nyolcbites CPU kialakítása .....	61. o.
6.1. 8080A-ra épülő CPU.....	61. o.
6.2. CPU kialakítása a 8085A mikroprocesszorral .....	63. o.
6.3. A Z80 CPU .....	64. o.
7. A memória és az I/O elemek kezelése a mikroszámítógépekben.....	65. o.
7.1. Memória kezelése a mikroszámítógépekben.....	65. o.
7.2. I/O elemek kezelése a mikroszámítógépekben .....	69. o.
8. A mikroprocesszorok programozható kiegészítő egységei .....	71. o.
8.1. A 8255 párhuzamos periféria illesztő .....	71. o.
8.2. A 8251 kommunikációs illesztő áramkör.....	75. o.
8.3. A 8253/8254 időzítő/számláló áramkör .....	79. o.
8.4. A ZILOG programozható kiegészítő elemeinek megszakításkezelése .....	81. o.
9. Az A/D és D/A konverterek .....	85. o.
9.1. D/A konverterek és helyettesítő megoldások.....	87. o.
9.2. Az A/D átalakítók .....	90. o.
10. A mikrovezérlők .....	95. o.
11. A PLD áramkörök.....	101. o.

