

4. A mikroprocesszoros mikroszámítógép és fő elemei

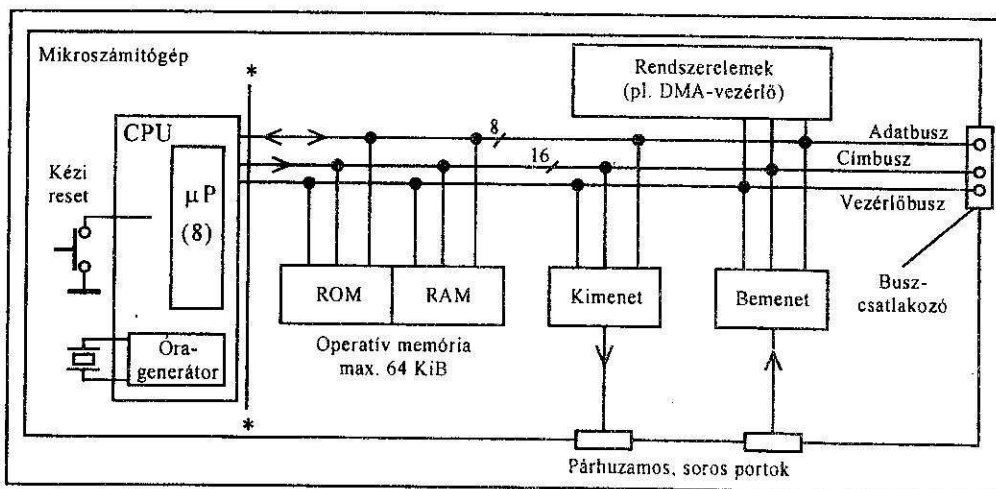
Korábban a mikroszámítógép a számítógépek egy méretbeli osztályát jelölte, mára azonban megváltozott a kifejezés értelmezése. Mikroszámítógépnek tekintünk minden olyan digitális készüléket, amely mikroprocesszorra, mikrovezérlőre épül. Mikroszámítógép a PC, de mikroszámítógépet találunk az egérben, a klaviatúrában, a rádiótelefonban, a mosógépben, a telefaxkészülékben is.

Bár ez a jegyzet a mikrovezérlőkkel foglalkozik, a mikroszámítógépre vonatkozó alapismereteinket a mikroprocesszorra épülő mikroszámítógép bemutatásán keresztül célszerű megszerezni. A PC-k, a munkaállomások egyre nagyobb szóhosszúságú, egyre gyorsabb mikroprocesszorokkal épülnek. Ezeket Magyarországra készen vagy szerelt egységekként importálják, a hazai fejlesztések száma elenyésző. A mikrovezérlőkkel más a helyzet, ezekkel itthon is folyamatosan fejlesztenek termékeket, eszközöket. A mikrovezérlőkkel ezért célszerű részletesebben megismerkedni. Azt is látni fogjuk, hogy az irányítási, automatizálási feladatokhoz elsősorban a mikrovezérlős elektronikák alkalmazhatók.

A mikrovezérlővel kialakított beépített (embedded) fedélzeti mikroszámítógépek többsége ma is nyolcbites. A rendszertechnika megismeréséhez ezért először a nyolcbites processzorok felépítését, működését fogjuk feldolgozni. Ezek a mikroprocesszorok még teljesen áttekinthetők, működésük a legapróbb részletig megismerhető, követhető. A megszerzett alapismeretek birtokában a mikrovezérlős elektronika felépítését, működését már könnyebben meg tudjuk majd érteni.

4.1. A mikroszámítógép felépítése

A nyolcbites mikroprocesszorok egységesen 16 bites címekeket használnak, ami 64 KiB memória közvetlen megcímezését teszi lehetővé (1 Ki = 1024, a B pedig a bájt, a nyolcbites szóhosszúság rövidítése). A mikroszámítógép általános felépítését a 4.1. ábra mutatja be.



4.1. ábra

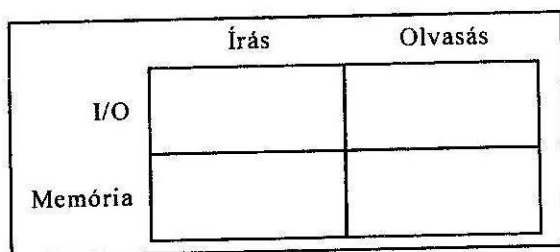
A mikroszámítógép központi végrehajtóegysége (Central Processing Unit, CPU) feladata a hármas buszrendszer kezelése (kétirányú nyolcbites adatbusz, egyirányú 16 bites címbusz és vezérlőbusz). A buszrendszer ábrázolásakor a buszok nevét, bitszámát, adatirányát is feltüntetjük. A vezérlőbusznál azért hiányzanak a paraméterek, mert azt nem egységesítették a gyártók.

A külvilággal a bemeneti (Input) és a kimeneti (Output) egységeken át tartja a kapcsolatot a mikroszámítógép (ezeket általában I/O elemeknek hívjuk). A csatlakozófelületük többnyire szabványos (soros port, párhuzamos port), a mikroszámítógépen belül az I/O elemek a buszrendszerre csatlakoznak. Szintén a három buszhoz kapcsolódik az operatív tár (operatív memória), 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 mikroszámítógépben esetenként további rendszerelemek is lehetnek, pl. megszakításvezérlő, DMA-kezelő stb.

A program, az utasítássorozat az operatív tárban található, onnan olvassa be a CPU egymás után a végrehajtandó utasításelemeket. 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 resetkéréskor, illetve logikai resetjel hatására. A CPU-ban találjuk meg a mikroprocesszort. Van olyan μP , amelyik önmagában komplett CPU, de a legtöbb esetben több-kevesebb külső áramkörrel kell kiegészíteni a mikroprocesszort, s csak így alakul ki a működőképes központi egység.

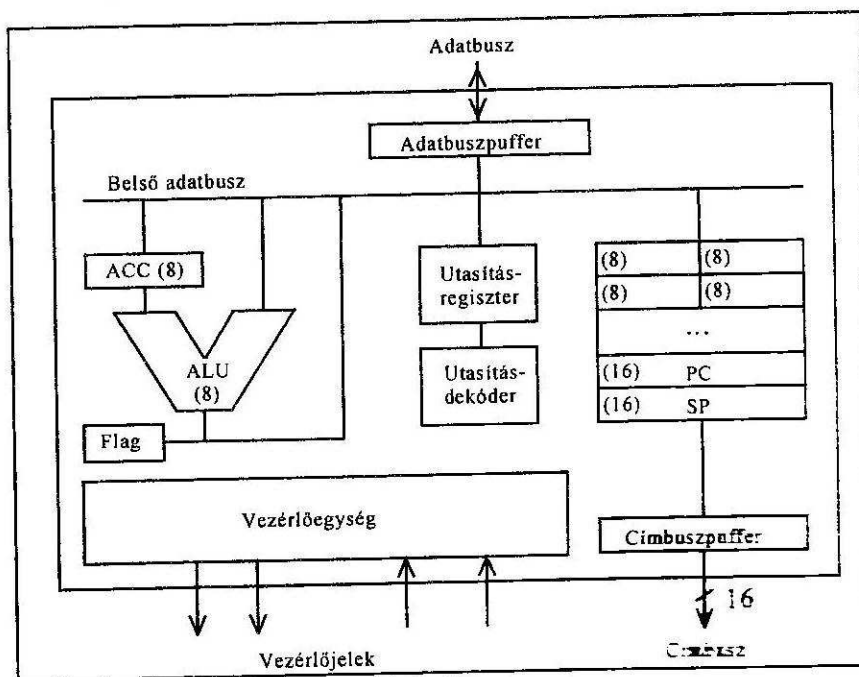
A mikroszámítógépek egy részénél a buszrendszert csatlakozókra is kivezetik, s így a felhasználó a buszrendszerre kapcsolódó elemekkel tudja bővíteni a gépét (nyitott felépítésű mikrogép). Ha nincs buszcsatlakozó, a mikroszámítógép zárt felépítésű.



4.2. ábra

A mikroszámítógépben a CPU-t és környezetét a 4.1. ábrán a két végén csillaggal jelölt vonal választja el. Hardver szempontból a nyolcbites mikroszámítógép egy időbeli 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 kiküld (ír) vagy fogad (beolvas). Az adatmozgás egyik részveője minden esetben a CPU, a másik fél lehet memória vagy I/O elem. A négy legfontosabb

gépi ciklust (I/O írás, I/O olvasás, memóriairás, memórialvasás) a 4.2. ábra szerint lehet ábrázolni, a rekeszekben az egyes eseményeket kiváltó vezérlőjeleket fogjuk feltüntetni. Sokszor külön kezelik az utasításkód beolvasására szolgáló memórialvasási gépi ciklust (M1). További gépi ciklusok a veremolvasás, veremírás és a megszakításelfogadás gépi ciklus.



4.3. ábra

A 4.3. ábrán a mikroprocesszorok legfontosabb belső elemeit láthatjuk. A program utasításai az utasításregiszterbe, onnan az utasításdekódolóba 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 μP regisztereiben lehet tárolni, ezek között kitéüntetett szerepe van az akkumulátornak (A, ACC). Az ALU-ban végzett művelet eredményének egy biten kifejezhető jellemzőit tárolja a flagregiszter. Egy-egy flag (jelzőbit) jelezheti, hogy az eredmény zérus vagy sem, hogy átvitelrel jött-e létre vagy sem, hogy pozitív vagy negatív, hogy páros vagy páratlan paritású stb. A flagbitek értékét a feltételes utasítások végrehajtásakor veszi figyelembe a mikroprocesszor.

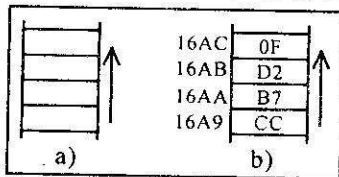
Az operatív tár címzésére a mikroprocesszor a programszámlálót (PC, Program Counter) használja, ami 16 bites egység. Egy másik címzőegységet is látunk az ábrán, az SP (Stack Pointer) a veremtár címzésére szolgál.

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-frekvenciának természetesen van felső határa, ugyanakkor többnyire alsó határértéket is előír a gyártó (a korszerű CMOS-processzorok esetén általában tetszőlegesen lassú órajellel is működőképeselek az áramkörök).

A mikroszámítógép leghosszabb működési egysége az utasításciklus. Annyi különféle utasításciklusa van a mikrogépnek, ahány utasításból a mikroprocesszor utasításkészlete áll. A nyolcbites processzorok utasításkészlete általában 70–150 utasítást tartalmaz. Az utasításciklus magába foglalja az utasítás beolvasását, értelmezését és teljes végrehajtását. Az utasításciklus annyi gépi ciklusból áll, ahány adatmozgás szükséges a végrehajtásához.

4.2. 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 az egységes memóriában (az operatív tárban) találhatóak (a mikrovezérlők a Harvard-architektúrát követik, elkülönülő programmemóriát és



4.4. ábra

adatmemóriát használva). A mikroszámítógép memóriájának egy-egy részletét a 4.4.a) ábra szerint lehet ábrázolni, a b) részleten néhány rekesz tartalmát és címét is feltüntettük. A memóriacímek az ábránkon mindig felfelé nőnek, erre utal a nyíl is. A memóriacímeket és a rekesztartalmakat hexadecimális értéként fogjuk kezelni és feltüntetni. A bemutatott részletek egy nyolcbites mikroszámítógépből valók, ezért a rekeszek egy bájt méretűek, a címek pedig négy hexadecimális helyértékűek (kétbájtosak).

A nyolcbites mikroprocesszorok memóriájában az adatok lehetnek nyolcbitesek (egy bájt) vagy 16 bitesek (két bájt), hiszen pl. címértékeket is el kell ott helyezni. Ha egybájtos adatok vannak a bemutatott memóriarészletben, akkor a tartalom így értelmezhető:

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

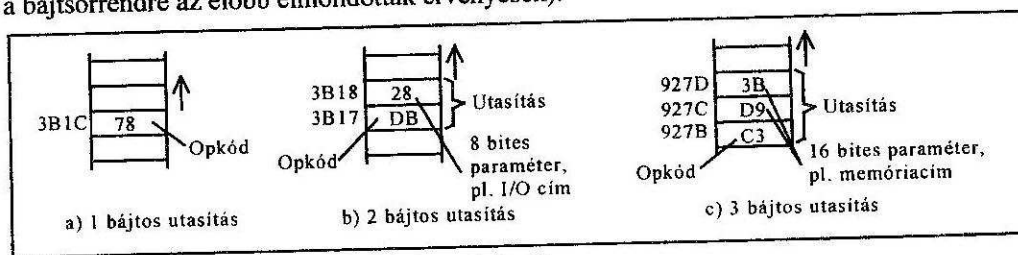
stb. A 16 bites (kétbájtos) adat értelmezéséhez tudni kell, milyen sorrendben helyezi el a bájtokat a CPU a memóriában. Ez processzorfüggő, pl. az Intel és a Zilog mikroprocesszorai a 16 bites adat vagy cím felső bájtját helyezik magasabb címre, és az alsó bájt kerül a kisebb című rekeszbe, a Motorola processzorok fordítva rendezik el a bájtokat. Ha az ábrázolt részlet egy Intel processzorhoz tartozik, és a 16AAh, 16ABh címeken egy 16 bites adat található, akkor azt így kell értelmeznünk:

- a 16AA címen D2B7h .

Az utasítás a memóriában legalább egy rekeszt foglal el (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 bájtos.

Az utasítás fő részei: a műveleti kód (operációs kód, opkód) és a paraméterek. Az opkód lehet egybájtos, de egyes processzoroknál találkozhatunk több bájt hosszúságúakkal is. Egyes utasításoknak van, másoknak nincs paraméterük, ami szintén lehet egybájtos vagy hosszabb. A memóriában az utasítás min-

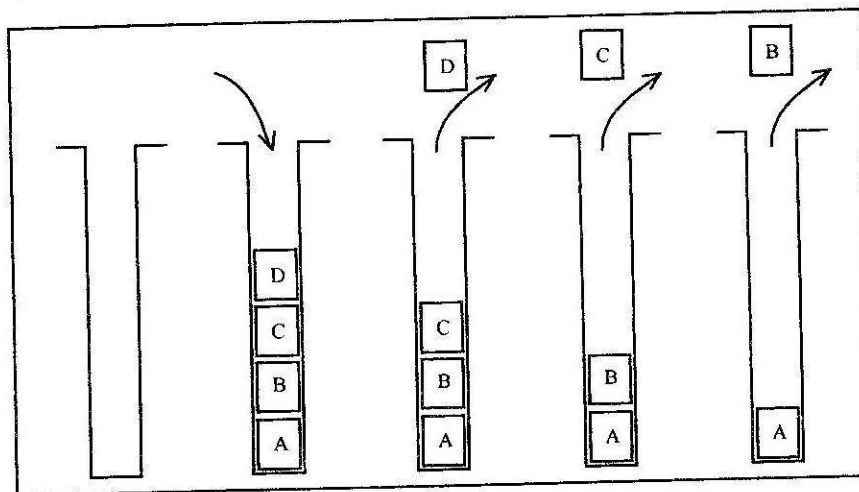
dig az opkóddal kezdődik, az esetleges paraméterek mögötte helyezkednek el (ha a paraméter 16 bites érték, a bájtrendre az előbb elmondottak érvényesek).



4.5. ábra

Egy utasítás végrehajtása legalább annyi gépi ciklust igényel, ahány bájtos, hiszen az utasítás elemeit be kell olvasnia a CPU-nak. A 4.5. ábrán látható néhány utasítás elhelyezkedése a memóriában egy olyan mikroprocesszor esetében, melynél az opkód minden esetben egybájtos. Az opkódot beolvasó gépi ciklus M1, az esetleges további utasításelem-beolvasó gépi ciklus már csak memóriolvasás. Ha az utasítás elemeinek beolvasása után a végrehajtásához már nincs szükség a mikroprocesszor és környezete között további adatmozgásra, az utasítás végrehajtása több gépi ciklust nem is igényel. Ha azonban a végrehaj-

táshoz adatmozgás is kell, annak mennyisége határozza meg a további gépi ciklusok számát.



4.6. ábra

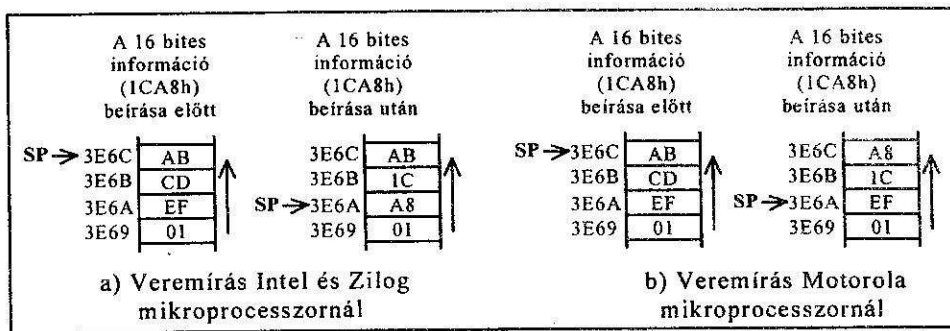
A verem (stack) jellegű memóriakezelés igen sajátos működés. Megértéséhez képzeljünk el egy szűk vermet, melybe ládákat engedünk le (4.6. ábra). Először az A-t, majd a B-t, azt követően a C-t, végül a D-t. Ha most kiemelünk egyet, az természetesen a D lesz. Egyrészt a D kiválasztásához nincs szükség semmilyen intézkedésre, hiszen leg-

felül 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, sorban! Ezt a memóriatípust LIFO (Last-In-First Out, Utolsóként-Be-Elsőként-Ki) névvel illetjük, a leírt működésre utalva. A mikroszámítógép veremtára az operatív memóriában található, a veremtármutatató (Stack Pointer, SP) jelzi, meddig van feltöltve a verem. A mikroprocesszorok 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 4.7.a) ábrán az Intel és a Zilog mikroprocesszorainak veremkezelése látható. Itt íráskor 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 4.7.b) ábrán a Motorola processzorok veremkezelése figyelhető meg. Íráskor ez a mikroprocesszor az SP által mutatott címre ír, és ezután dekrementálja az SP tartalmát; olvasáskor előbb inkrementálja az SP értékét, majd a kapott címről olvas. Az ábrán a Motorola mikroprocesszor is az 1CA8h 16 bites értéket helyezte be a verembe.

A mikrovezérlők belső veremtárat használnak, ami sokszor teljesen elkülönül az egyéb memóriaterületektől.



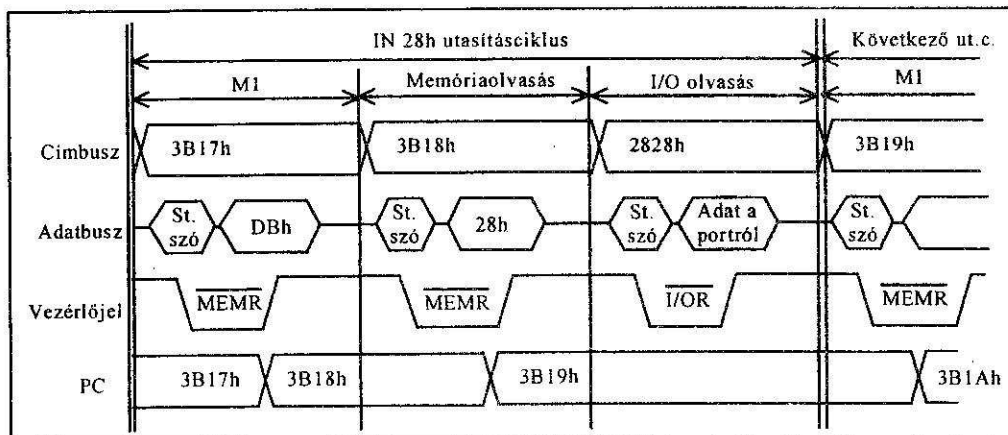
4.7. ábra

4.3. Az utasítás-végrehajtás menete

Egy adott μP utasításkészletére jellemző, hogy a kétoperandusos működéseket milyen felépítésű utasítás rendel el. A mai mikroprocesszorokra az ún. egycímes utasításfelépítés 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 módot, amelynél az utasításokat abban a sorrendben hajtja végre a processzor, ahogyan azok a memóriában elhelyezkednek, normál vagy természetes sorrendűnek 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. (Utasításelem az opkód és az utasítás esetleges többi bájta, a paraméter is.)

A 4.8. ábrán példaként egy bemeneti utasítás lefolyását mutatjuk be, i8080 mikroprocesszor esetén. A 3B17h címen lévő IN28h utasítás a 28h című bemeneti portról olvassa be az adatot, ami az adatbuszon keresztül belép a mikroprocesszorba, és ott az akkumulátorba kerül. Az utasítás kétbájtos, első bájta az opkód (ami ebben az esetben DBh), a második bájta az I/O cím, azaz esetünkben a 28h. Az utasításciklusban az első gépi ciklus M1, ekkor olvasódik be a processzorba az opkód. A második gépi ciklus szolgál a második utasításelem, adott esetben az I/O cím beolvasására, ez egy memóriaolvasás gépi ciklus. Az utasítás végrehajtásához egy külső adatot, a 28h című bemeneti port adatát kell a processzorba beolvasni, ehhez egy új gépi ciklus szükséges (ebben az esetben egy I/O olvasás gépi ciklus). Ezeket az eseményeket mutatja be az ábra. Az i8080 különlegessége, hogy a gépi ciklus jellegére az adatbuszon kiküldött státuszszóval utal, amiből a külső hardver képez vezérlőjeleket. Megfigyelhető, hogy a PC mikor inkrementálódik a gépi ciklus végrehajtása alatt (minden M1-ben és minden utasításelem-beolvasáskor).



4.8. ábra

A normál sorrendtől a feladat megoldása érdekében a μP eltéríthető. Az eltérítés szoftver eszközei a vezérlésadó utasítások (ugrás, szubrutinra ugrás, szubrutinból visszatérítés) és a HALT; a hardver eltérítési lehetőségek a mikroprocesszor vezérlőbemeneteire vezetett logikai jelekkel érhetők el (RESET jel, megszakítás, READY/WAIT működés, HOLD kérés). A következőkben ezekkel a lehetőségekkel sorra megismerkedünk.

Az ugróutasítás (jump, JMP) a 8 bites mikroprocesszoroknál alapesetben (közvetlen címzés esetén) 3 bájtos, az opkód és a 16 bites cím található benne. Az utasítás beolvasását követően a PC felülíródik az ugrási címmel – a következő M1 ciklusban ez a cím lép ki a címbuszon.

A feltételes ugróutasítás is így működik, ha a benne előírt feltétel teljesül; ha viszont nem, akkor a PC tartalma nem íródik felül, ott a normál inkrementálódás eredménye marad meg. Feltételként a μP egyes flagjeinek 0 vagy 1 értéke írható elő (pl. JNZ : ugrás, ha Z = 0, JC : ugrás, ha C = 1). A mikroprocesszorok esetében általában a Z (zérus érték), az S (előjel), a C (túlsordulás) és a P (paritás) flag használható feltételként.

Ha az ugrás után a most elhagyott programot folytatni kell a kiugrasi 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 hárombájtos (közvetlen címzés esetén), az opkódot tartalmazza és a 16 bites szubrutin kezdőcímet. 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 vagy RETURN (visszatérés) utasítással tehetjük meg, ez is lehet feltétel nélküli vagy feltételes.

A HLT (halt, leállítási) utasítás hatására a μP felhagy a szoftver végrehajtásával – ebből az állapotából csak hardver úton mozdítható ki.

A resetfolyamatot a RESET jel váltja ki. 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 resetfolyamat egyik legfontosabb feladata, hogy meghatározza a RESET jel utáni programcímet; a mikroprocesszorok többsége ilyenkor a 0000h címről utasításbeolvasással indul.

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

A megszakításkérő jel (INTERRUPT, INT, I) hatására a μP a megszakítást kiszolgáló szubrutinra lép. A megszakítást kiszolgáló szubrutin végrehajtása után a CPU visszatér a megszakított programhoz, azt folytatja; így a megszakítás tulajdonképpen egy hardver úton kiváltott szubrutinra ugrás. Eppen a megszakított program folytathatósága érdekében szükséges, hogy a megszakítás kiszolgálása szubrutinnal valósuljon meg. Egyes processzorok a megszakítási szubrutinra ugráskor másképpen viselkednek (más információkat mentenek el), mint közönséges szubrutinhíváskor. Emiatt ezeknél a megszakítási rutin végére a „normálistól” eltérő visszatérítő utasítás (pl. RETI vagy IRET) kerül, hogy az adatok visszatöltése is helyesen valósuljon meg.

A megszakításkérő bemenet lehet maszkolható (utasítással tiltható vagy engedélyezhető az elfogadás) vagy nem maszkolható, amit minden esetben elfogad a processzor. A μP 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 megszakításkérő bemenetek állapotát. A mai mikroprocesszoroknál belső működés is kiválthat megszakítást, esetenként az utasításkészletben megszakításutasítás is található.

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 μP .

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 memóriáhozáférés (Direct Memory Access, DMA) hozható létre. Ehhez azonban a mikroszám-

mítógépben lennie kell egy DMA-vezérlőnek, amit előzőleg a mikroprocesszor felkészített a DMA-folyamat kezelésére.

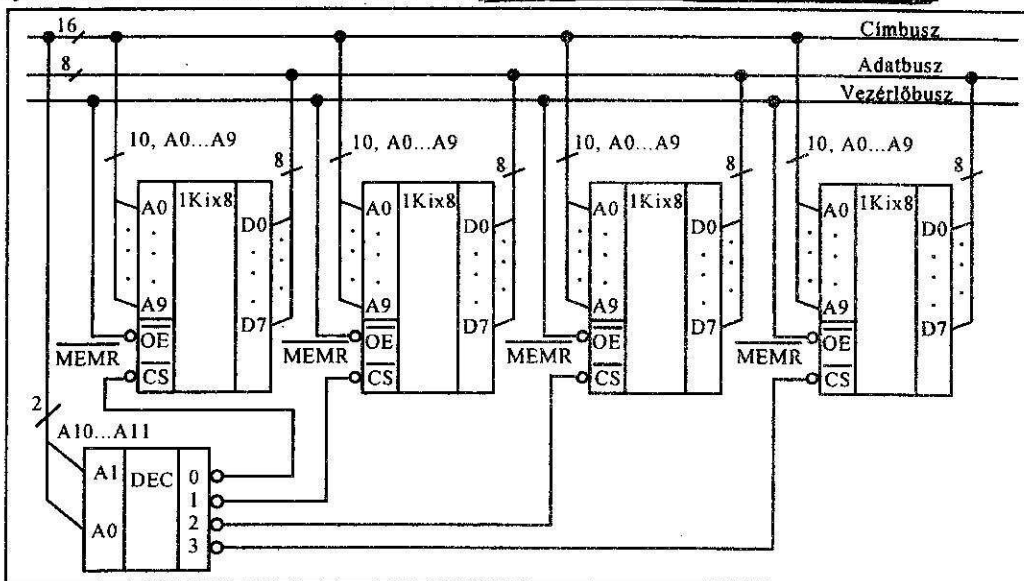
A READY vagy WAIT jelek segítségével a mikroprocesszor a gépi ciklus belsejében képes várakozni (várakozási óraciklusok beiktatásával) valamilyen külső eseményre. Ezzel a megoldással nyújtják meg a gépi ciklusokat pl. lassúbb memória használata vagy perifériák lassúbb működése esetén. Ez a lehetőség 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 μP működése során ezért ez a két fázis egymást váltogatva, egymás után jelentkeznek. A mai, nagyobb teljesítményű mikroprocesszorok több vezérlőegységgel épülnek, a belső működés közben a memóriából előre beolvashatják a következő utasításelemeket, sőt, azokat elő-dekódolhatják, 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).

4.4. A memória-áramkörök

A mikroszámítógép bemeneti elemét (bemeneti portját) olyan áramkörrel lehet megoldani, mely biztosítja a szükséges számú adatutatót, a kimenetein háromállapotú fokozattal rendelkezik (mivel az adatbuszra csatlakozik, amikor nincs kijelölve, lebegtetnie kell a pontjait). A kimeneti portnak is meg kell valósítania a megfelelő számú adatutatót, de tárolókkal kell készülnie (pl. regiszter vagy megfelelő számú D tároló lehet). A CPU beírja ide a kimeneti adatot, amit a kimeneti pontokon mindaddig fenn kell tartani, míg új kimeneti adattal a processzor felül nem írja, nem módosítja azt. A memória-áramköröket nem tudjuk egy hasonló rövid bekezdésben bemutatni, mert számos különféle fajtájuk van, s kezelésük is összetettebb.

A memória-áramkörök olyan félvezető csipek, melyek nagy tömegű információ tárolására használhatók fel. A tárolási egység a rekesz (szó), ennek bitszáma, hosszúsága (szóhosszúság) a memória IC egyik fő adata. A memória IC címzobemenetei (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 címzobemenete van, a rekeszei száma 2^n . Az áramkörön belül a címetek sor- és oszlopcímre bontva használják fel, de a DRAM-ok kivételével ennek a felhasználó számára nincs jelentősége.



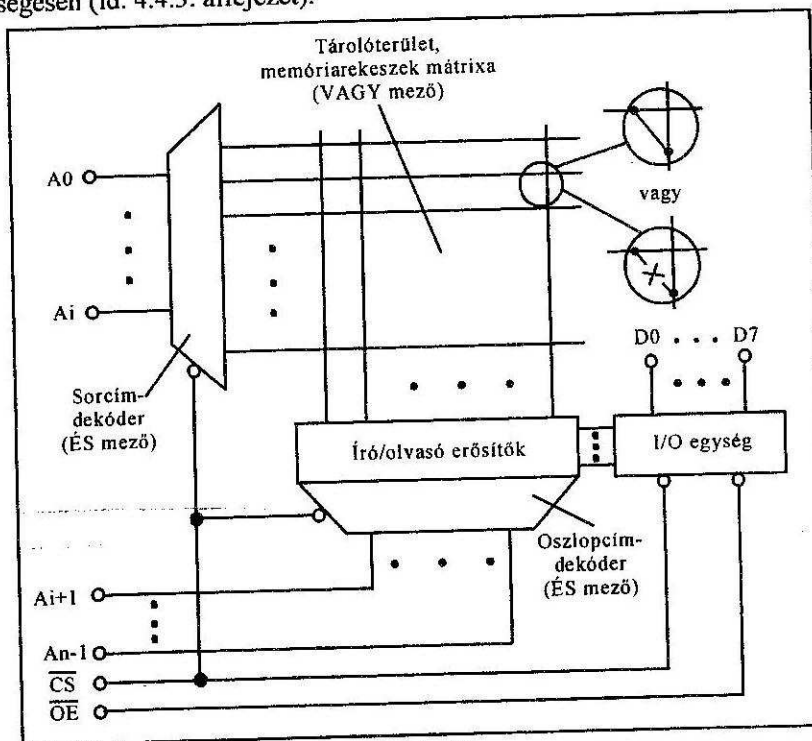
4.9. ábra

Ha egy memóriának 2048 db 8 bites rekesze van, a kapacitását több módon is megadhatjuk. Előnyben részesítjük azokat a felírási formátumokat, amelyek a memória szerkezetét is leírják, azaz szószám x szóhosszúság alakúak, pl. 2048 x 8 vagy 2 Ki x 8. Ugyanezt a memóriakapacitást másképpen is megadhatjuk, bár ezek a kifejezések a memória szerkezetére, a rekeszhosszúságra nem utalnak (pl. 2 KiB, vagy 16384 bit vagy 16 Kibit (az IEEE nemzetközi szervezet javasolja az 1024 bináris prefixumként történő megjelenítésére a szokásos K vagy k helyett a Ki rövidítést, hasonlóan az M helyett az Mi, a G helyett a Gi jelöléseket, egyúttal szükség esetén a bájt rövidítését B betűvel, viszont tiltja a bit rövidítést, utóbbit mindig ki kell írni). A memória IC adatkimenetei mindig háromállapotúak, hogy adatbuszra csatlakozhassanak. A kimenetengedélyező jel általában OE. A teljes IC működését engedélyező jel a chip select vagy chip enable (\overline{CS} vagy \overline{CE}). Ha a memória írható/olvasható, e két működési módot többnyire egy R/W jel vezérli. A \overline{CS} jelet használjuk fel akkor is, ha több memória IC-ből épül fel a mikroszámítógép memóriarendszere (4.9. ábra). A memória-áramkörök általában kijelölés nélkül a névleges üzemi tápáram töredékét veszik fel (stand-by állapot).

A mikroszámítógépben az operatív tárat alkotó memória-áramkörök véletlen hozzáférésű memóriák (random access memory), ami azt jelenti, hogy egy rekesz elérési ideje független a rekesz címétől, 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 csak olvasható memóriából (ROM és társai),
- létrehoznak egy írható/olvasható területet RAM memóriákkal, de ezek illanók.

Megoldható a nem illanó, írható/olvasható memória is, de csak kisebb kapacitással és meglehetősen költségesen (ld. 4.4.3. alfejezet).



4.10. ábra

A memória IC fő részei a címetek fogadó címdekóder és a tartalmat őrző terület (memóriamátrix, memóriaterület). Mivel a címdekóder a bemeneteket közvetlenül fogadó ÉS kapukból áll, a tárolóterület pedig logikailag egy VAGY mező, ez is egy klasszikus kétszintes logikai hálózat, a PLA-hoz hasonló logikai szerkezet (de a PLA-ban az ÉS mező programozható, míg itt fix kialakítású). A címdekóder kimenetei és a VAGY kapuk bemenetei egy vezetékhalózat keresztpontjain köthetők össze egymással (4.10. ábra). Egy címdekóder-kivezetés egy rekeszt, egy szót jelöl ki. Ha ez a rekesz-kijelölő (vízszintesen ábrázolt) vezeték a VAGY bemenettel (függőleges vezeték az

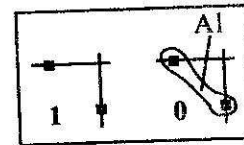
ábrán) össze van kötve, az pl. 0 értékű bitet jelent, ha nincs összekötve, az 1-t. (Van olyan memória is, amelynél a logikai értékek fordítva jelennek meg.). A memória-áramkörök abban különböznek egymástól, hogy a tartalmat megvalósító keresztponti „kapcsolókat” hogyan alakították ki bennük. A követke-

zókben áttekintjük az elterjedt megoldásokat, a részletekkel gyakorlatokon, katalógusok alapján fogunk megismerkedni.

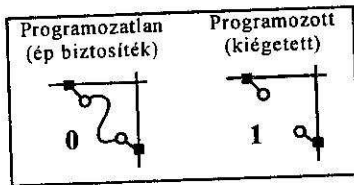
Ha egy memória-áramkör pl. 16 címvezetékkel kezel, 65536 (64Ki) rekeszt tartalmaz. Ha egyetlen címdekóder állítja elő a rekeszkijelölő vezetékeket, az 65536 db 16 bemenetű ÉS kapuból áll. Ha a rekeszeket 256 x 256 elemű mátrixba rendezik, a sorok illetve oszlopok kiválasztásához csak 256 db 8 bemenetű ÉS kapu kell, azaz összesen 512 kapu. Ez egyszerűbb csipkialakítást, kevesebb belső vezetékkel jelent, ezért kezelik a memória-áramkörök belsejében a címeket sor- és oszlopcímekre bontva.

4.4.1. A ROM-változatok

A ROM (Read Only Memory, csak olvasható memória) csoport minden tagja nem illanó, csak olvasható memória. A mikroszámítógépben a mikroprocesszor csak olvassa ezeket (memóriaolvasás ciklusban, memóriairás ciklusban ezek nem kezelhetők). A szűkebben értelmezett ROM-memória az ún. maszkprogramozott áramkör. A 4.10. ábrán szereplő keresztpontokban fémezéssel, tehát szokásosan alumíniummal teremthet kapcsolatot a memória gyártója (4.11. ábra). Elhelyezi a csip felületén az alumínium összekötést, ha a felhasználói igény az adott biten 0 értéket ír elő, vagy összekötetlenül hagyja a keresztpontot (1 érték). A maszkprogramozást csak az IC-gyártó képes végrehajtani, a kívánt tartalmat ezért el kell küldeni hozzá. A ROM információátviteli időtartama korlátlan, de a beprogramozás után nem is törölhető vagy módosítható.



4.11. ábra

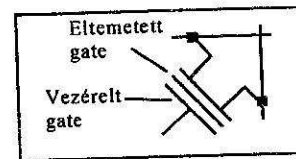


4.12. ábra

A PROM a felhasználó által programozható csak olvasható memória (Programmable ROM, programozható ROM). Az üres, tartalom nélküli áramkörben a tartalomórzó mátrix minden keresztpontja (4.12. ábra) össze van kötve egy-egy apró olvadóbiztosítékkal (0 értékű bit). Ahol 1 értékre van szükség, a programozáskor a kis fémhidat ki kell olvasztani. A PROM a beprogramozott tartalmat korlátlan ideig őrzi, később még kiolvasható egy meghagyott biztosíték, de helyreállítani nem lehet a keresztponti összeköttetéseket, azaz nem törölhető. Általában bitenként

lehet csak programozni, mert a biztosíték kiégetése jelentős hőfejlődéssel jár.

Az EPROM (Erasable PROM, törölhető PROM) törölhető, programozható, csak olvasható memória. Az információátviteli mező keresztpontjaiban egy-egy FET-tranzisztor csatornája helyezkedik el (4.13. ábra), a csatornát kezelő gate-elektrodát szigetelőburok veszi körül (eltemetett gate, lebegő gate). A programozatlan EPROM-ban minden eltemetett gate üres, töltésmentes, a csatorna nem épül fel, ilyenkor minden tárolt bit 1 értékű. Ahol 0 értéket kell tárolni, az IC-re kapcsolt programozófeszültség és a vezérelt gate segítségével elektronokat juttatnak az eltemetett gate-re, ami felépíti a csatornát, létrehozva a keresztpontban a galvanikus kapcsolatot. Az eltemetett gate kapacitása kb. 0,1 pF, a teljes töltés kb. 30 millió elektron. A töltésből folyamatosan veszít a tranzisztor, 1-2 percenként megszökik egy elektron. A katalógusok többnyire 10 év információmegőrzési időt garantálnak, de a gyakorlat ennél hosszabb tárolási időt igazol.



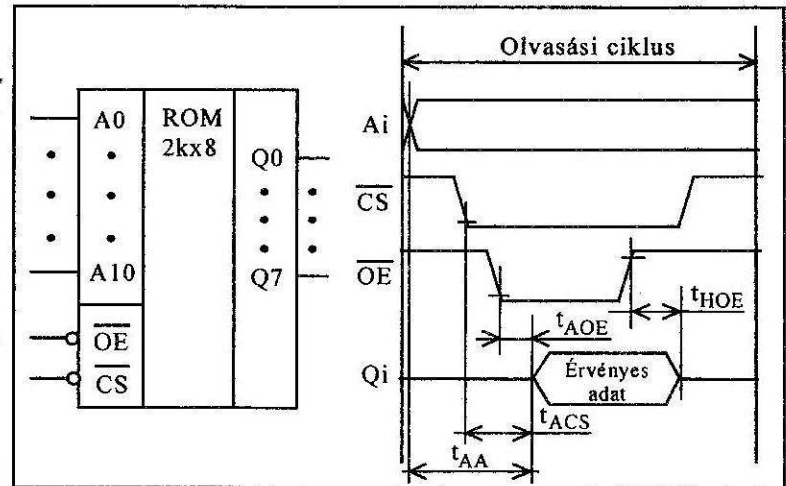
4.13. ábra

Az EPROM tokozásán, a csip felett kvarcablakot helyeznek el. Ha ezen át megfelelő intenzitású ultraribolya (UV) fény éri az áramkört, a töltések szétáramlása felgyorsul, kb. 15-20 perc alatt a teljes áramkör törlődik, ismét minden biten 1 lesz a tartalom (nem szelektív törlés, a teljes IC tartalma törlődik). Az EPROM-ok néhány százszor programozhatók át.

Olcsóbb, zárt műanyag tokban is kapható EPROM-csip, ezt csak programozni lehet, törölni nem, ez az OTP (One Time Programmable, egyszer programozható) EPROM. Az EPROM-ok egy fiatalabb csoportja a Flash-EPROM, mely zárt tokban készül, villamos jellel, nem szelektíven (általában blokkonként vagy teljesen) törölhető. Mivel a Flash-EPROM gyártásakor más technológiák elemeit is hasznosítják (pl. az EEPROM-áramkörökét), ma már többnyire csak egyszerűen Flash-memóriaként említik ezeket.

Az EEPROM (Electric Erasable PROM) elektromosan törölhető, programozható, csak olvasható memória, törlése szelektív. A kereszteződési pontokban itt is FET-tranzisztorok vannak, de ún. szendvics gate szerkezettel. Ebben a típusban a csatornát kezelő gate és a vezérlő gate közötti félvezetőrész vezetőképessége a programozófeszültséggel vezérelhető, így oldható meg az elektromos törlés. Az EEPROM információőrzési ideje attól függ, hogy korábban hány törlés/programozás ciklust élt át. Az átlagos EEPROM-ok 10 000–40 000 ciklus után még 10 évig őrzik az információt, de vannak olyanok is, melyek kedvezőbb adatokkal rendelkeznek.

Végül a 4.14. ábrán bemutatjuk a ROM-áramkörök tipikus olvasási ciklusát. A t_A jelű idők az elérési idők, a t_H az adatfenntartási idő jele. A kijelöletlen ROM-áramkör általában stand-by állapotba kerül. A számítógépekben a ROM-elemeket olyan prog-



4.14. ábra

ramrészeket tárolására használják, amelyeknek bekapcsoláskor mindig rendelkezésre kell állni (set-up, boot, BIOS-elemek), s melyeket a számítógép használata során nem vagy csak igen kivételes esetben kell megváltoztatni. A mikroprocesszoros számítógépek a ROM-elemekben tárolt programokat többnyire nem onnan futtatják, a használathoz a programrészeket előbb RAM-területre másolják. A mai ROM-áramkörök gyakorlatilag mind CMOS-elemek.

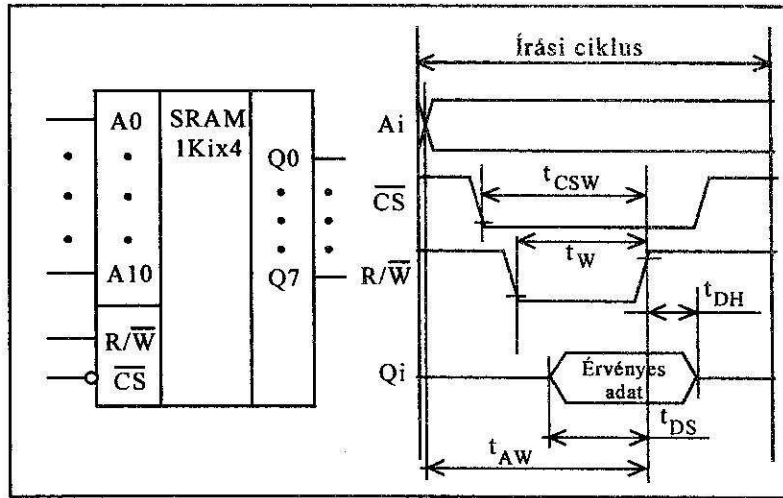
4.4.2. A RAM-változatok

Az írható/olvasható (RAM) memóriák tápfeszültség hiányában elveszítik a tartalmukat, illanók. Világszerte RAM-nak nevezik ezeket a memóriaváltozatokat, ami nem precíz megjelölés, mivel a RAM a véletlen elérésű memória (Random Access Memory) angol kifejezés kezdőbetűiből alkotott betűszó – és a ROM-változatok is mind véletlen elérésűek. Mégis el kell fogadnunk, hogy a hárombetűs megjelölés (RAM) a gyakorlatban ezt a memóriafajtát jelenti. Az áramkörök két nagy csoportja a statikus RAM (SRAM) és a dinamikus RAM (DRAM).

A SRAM-áramkörben minden tárolómátrix-keresztpontban egy komplett bistabil multivibrátor van, egy hagyományos egybites tároló, általában 6 ... 14 tranzisztorból kialakítva. A sor- és oszlopvezetékek aktivizálják a bistabilt, annak kimenőértéke belép a kimeneti VAGY kapuba. A SRAM tápfeszültségre kapcsolásakor ezek a flip/flop-ok véletlenszerűen állnak be 0 vagy 1 értékre, ezért illanó a SRAM-áramkör. A korszerű CMOS SRAM stand-by tápárama igen kicsiny, de lecsökkentett tápfeszültség és stand-by állapot mellett is képes a tartalmát őrizni. Egy +5 V névleges tápfeszültségű SRAM esetén stand-by módban általában akár 2,3–2,2 V-ig lecsökkenthető adatvesztés nélkül a tápfeszültség.

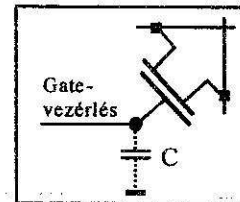
Ma a SRAM-áramkörök kivétel nélkül CMOS IC-k, egyre nagyobb kapacitással és sebességgel. A SRAM-áramkörök a mikroprocesszorokra épülő számítógépekben különleges feladatokat látnak el. Jellemzően SRAM-elemkből alakítják ki a cache gyorsítárakat, melyek a processzorok teljes működési sebességének a kihasználását biztosítják. A GHz feletti órajel-frekvenciák esetén újszerű, órajelet is használó szinkron memóriákat használnak fel (SSRAM).

A SRAM olvasási ciklusa nem különbözik lényegesen a ROM-áramkörökétől, az írás/olvasás vezérlőjével az olvasási módot kell előírni. A 4.15. ábrán egy 1Ki x 4 kapacitású SRAM írási ciklusát mutatjuk be.

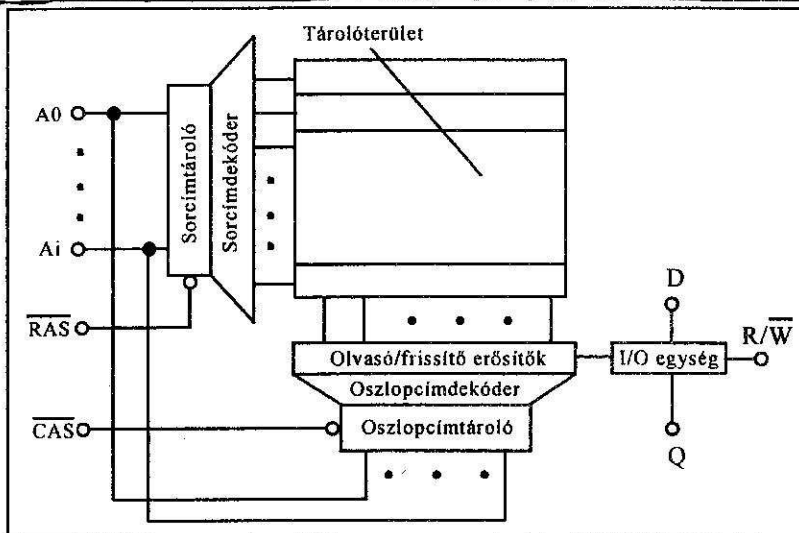


4.15. ábra

A *dinamikus RAM* (DRAM) esetén a tárolóterület keresztpontjaiban ismét csak egy-egy FET-tranzisztort helyeztek el (4.16. ábra). A tranzisztor gate-source kapacitása tárolja az oda vezetett elektrontöltést. (A mai DRAM-elemek többségében már egy source-szubsztrát kapacitás szolgál az információ megőrzésére, de ez a működésen lényegileg nem változtat.) A tároló kapacitás 30–50 pF, a teljes töltés kb. 1 millió elektron. Mivel az előző fokozat is FET-tranzisztoros, a kondenzátor töltése csak lassan fog kisülni, kb. 10 ms-ig a töltés még teljes értékűnek tekinthető, a tranzisztor csatornája összeköti a vezetékeket. A folyamatos működés érdekében a DRAM tartalmát rendszeresen, régebbi típusoknál 2 ms-onként, az újabbaknál 4 ms-onként (esetleg még ritkábban) frissíteni kell.



4.16. ábra

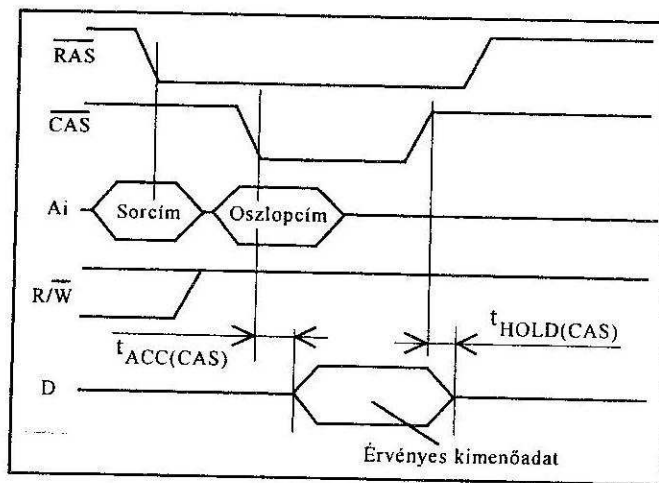


4.17. ábra

A DRAM-memóriák (4.17. ábra) címbevezetése általában multiplexelt, egymás után ugyanazokon az IC-lábakon át kell bevezetni először a sorcímet (azonosító jele a Row Address Select, RAS), majd később az oszlopcímet (azonosító jele a Column Address Select, CAS). Amikor az RAS lefutó élekor a sorcímet befogadja a DRAM, az a sorcímtárolóba jut, a sorcímtárolóba jut, a sorcímtárolóba dekódolja, a tárolómátrixhoz vezet, onnan a kijelölt sor kilép az olvasó-erősítőkbe. Ha ezután a CAS

jel lefutó élekor az oszlopcím is beolvasódik a tárolójába, az az oszlopcímdekóder útján a sor egy bitjét kiválasztja, ami az I/O egységhez jut. A DRAM rekeszei egyetlen bitesek (a szélesség 1 bit), az I/O egység segítségével valósul meg az egy bit olvasása vagy írása. A CAS lefutó éle zárja le az írási működést; az egybites érték visszakerül a megfelelő olvasó/frissítő erősítőbe, az RAS lefutó éle a sor (felfrissítve) visszakerül a helyére a tárolómátrixba.

A DRAM működési ciklusaiban az $\overline{\text{RAS}}$ és a $\overline{\text{CAS}}$ jelek élének igen fontos szerepük van, s ezen jelek kialakulási sorrendje is szigorú rend szerinti. Megfigyelhető mindez a 4.18. ábrán, ahol egy DRAM olvasási ciklusa látható.



4.18. ábra

beállítása után egy $\overline{\text{RAS}}$ impulzus kialakítása a DRAM bemenetén. Ez a ciklus csak a frissítést végzi el, ebben a ciklusban a $\overline{\text{CAS}}$ nem is aktivizálódik. Ezt a frissítési megoldást minden DRAM-változatnál lehet alkalmazni.

Az újabb áramkörök kedvezőbb lehetőségeket is kínálnak, belső frissítőáramkörökkel. Ezek a DRAM-ok normál írási vagy olvasási ciklusok közben oldják meg egy-egy soron következő sor frissítését (nem azért, ami éppen íródik vagy olvasódik!). A frissítendő sort a belső frissítőszámoló jelöli ki. Csak a frissítési időpontot kell kívülről jelezni, amire a vezérlőjelek változási sorrendjének átrendezése alkalmas.

A frissítési ciklusok időbeli elosztására is több lehetőség adódik:

- *csomag (burst) frissítés:* a 2 ms (4 ms vagy hosszabb frissítési idő) leteltekor a CPU normál működése egy ideig megszakad, és a processzor maximális sebességgel minden sort felfrissít a DRAM-ban,
- *egyenletesen elosztott:* a 2 ms (vagy 4 ms) annyi időszelre van felosztva, ahány soros a DRAM, minden időszel letelején a CPU megszakításának hatására a normál működés leáll és egy sor felfrissül,
- *eseményhez kötött:* speciális CPU esetén bizonyos események fellépésekor (pl. a Zilog processzoroknál minden MI gépi ciklusban) megjelenik egy frissítő parancs, esetenként vele együtt a processzor a frissítőcímet is előállítja.

A kvázistatikus RAM (KSRAM, IRAM) egybeintegráltan tartalmazza a DRAM-csipet és a frissítőáramkört. Egy RFSH (refresh) vezérlő bemenet L szintjének hatására valósít meg egy sorfrissítést. Ha 4 ms alatt az összes sor nem frissült fel, a BUSY kimenetén foglaltságot jelez és belül egy komplett burst frissítést valósít meg. Ennek az áramkörtípusnak már befejeződött a gyártása.

A DRAM működését többféleképpen is meg lehet gyorsítani. A sorozatos, egymás utáni címekre történő írás vagy egymás melletti rekeszekből történő olvasás különösképpen meggyorsítható. A lapozásos kezelés a DRAM-elemek esetében azt jelenti, hogy az oszlop cím beküldése után csak a sorcímeket adják meg egymás után, az oszlop cím ismételt beírásai elmaradnak. Tulajdonképpen csak az írási/olvasási technikákban különböznek egymástól a DRAM különféle változatai (FPM, EDO, BEDO). A gyakorlaton, katalógusok alapján ismerkedünk majd meg részletesen is a különféle változatokkal.

A DRAM-áramkörök a félvezetőpiac legkeresettebb elemei. Ennek az az oka, hogy a mikroprocesszorra épülő gépekben (PC-k, munkaadóállomások, szerverek) a szoftver minden esetben DRAM-ból fut. Akár ROM-ban, akár mágneslemezen tárolódik a program, a futtatása előtt (részben vagy egészben) betöltődik a DRAM-ba, a CPU csak a DRAM-ból képes programokat futtatni. Ebből következik, hogy a számítógépek sebességét mennyire meghatározzák ezek az áramkörök. Mivel a processzorok sebessége

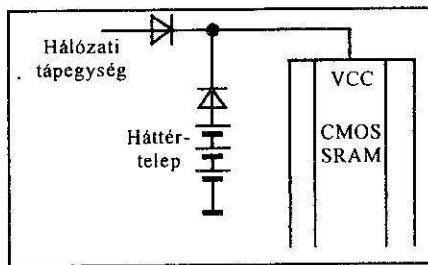
rohamosan nő, azokat közvetlenül nem is tudja a DRAM kiszolgálni, ezért kellett beiktatni a (sokszor többlépcsős) nagy sebességű cache gyorsítótárakat a DRAM és a CPU közé.

A mikroprocesszorok, mikrovezérlők némelyikénél az órajelnek (főleg a régebbi fejlesztésű típusoknál) nemcsak felső, hanem alsó frekvenciahatára is van. Az alsó határnak az a magyarázata, hogy a processzorok belső tárolóterületeit DRAM jelleggel alakították ki, s az órajel biztosítja ezeknek a részleteknek a frissítését. A korszerű CMOS-típusokban többnyire teljesen statikus felépítést használnak, a memóriaelemek is SRAM jellegűek, így bármilyen lassú órajellel is működőképesek az áramkörök. (Mindez egyaránt igaz a mikroprocesszorokra és a mikrovezérlőkre.)

4.4.3. Az írható/olvasható, nem illanó félvezetős memória megvalósítása

A mikroszámítógépekben egy-egy kisebb kapacitású, *nem illanó, írható/olvasható memóriát* ki lehet alakítani. A PC setup információi is egy ilyen memóriában helyezhetők el, így bekapcsoláskor mindig rendelkezésre állnak, ugyanakkor szükség esetén egyszerűen módosíthatók.

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 visszamásolódik a SRAM rekeszeibe. A memóriagyártók fejlesztései között található olyan típus, amelyik SRAM jelleggel írható és olvasható, de tápfeszültség nélkül is őrzi a tartalmát. Lehetséges, hogy néhány év múlva már nagy tömegben gyártják is ezeket.



4.19. ábra

A jelenlegi gyakorlatban elterjedt megoldás az, hogy egy kis stand-by tápáramú CMOS SRAM IC-nek nagy kapacitású (F-os) kondenzátorból, elemről vagy akkumulátorról *háttértáplálást* biztosítanak, amikor a tápegység kikapcsolódik (4.19. ábra). Ezt a megoldást azért lehet nyugodtan alkalmazni, mert a korszerű CMOS SRAM stand-by tápárama néhány μA , a hosszú élettartamú elemek akár tíz évig is benn lehetnek a készülékben, az akkumulátor vagy a kondenzátor pedig, helyesen kivitelezett töltőáramkör esetén szintén tartós és biztonságos megoldást jelent.

4.4.4. Soros elérésű félvezetős memóriák

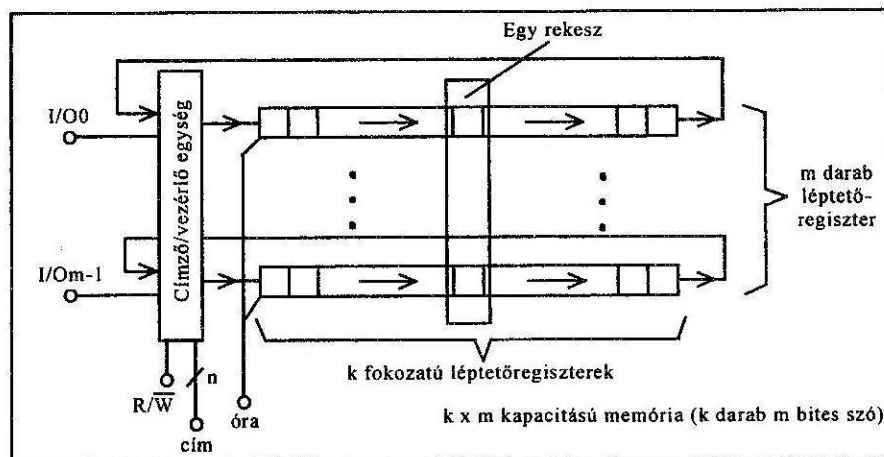
A szakirodalom röviden csak *soros memóriákat* emleget (Serial Memory) többféle áramköri megoldás esetén is, ami esetleg félrevezető lehet. Két fontos csoportot sorolnak ide:

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

Az eddig megismert memória-áramkörök mind *véletlen elérésűek* voltak, emellett, mivel a cím- és az adatbitek párhuzamos bevezetésére voltak felkészítve, *párhuzamos adatkezelésűek*. Az utóbbi időkben, elsősorban a mikrovezérlős készülékekben, egyre népszerűbbek a *véletlen elérésű, de soros adatkezelésű* memória-áramkörök, melyeknél egy vagy két adatvezetéken, egymás után jelennek meg a cím- és az adatbitek.

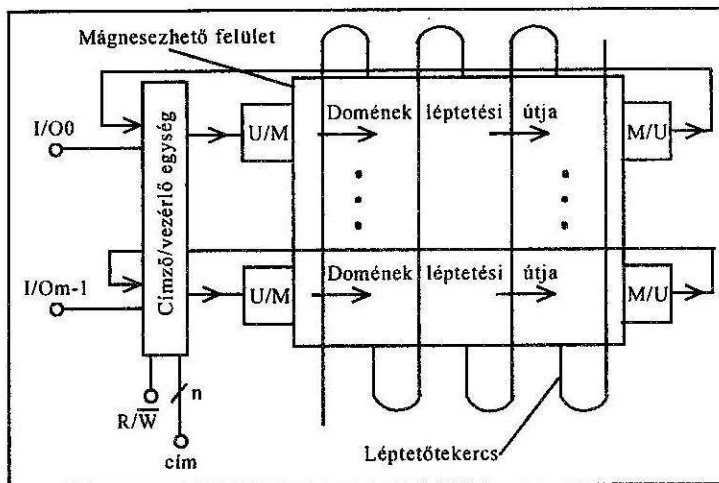
A soros adatkezelés az órajeles szinkron soros átviteli megoldásokra épül. Sok különféle megoldást kidolgoztak ezen a területen, ezek közül néhány szabványossá vált. A mikrovezérlőkbe ezeket a szabványos szinkron soros illesztőket építik be, így a megfelelő típusú soros adatkezelésű elem minden további nélkül hozzákapcsolhatóvá válik. A mikrovezérlő-családokkal együtt részletesen megismerjük majd a szabványos soros adatkezelési megoldásokat.

Ebben az alfejezetben viszont a *soros elérésű* félvezetős memóriákat mutatjuk be. A működési elvet a léptetőregiszterekből kialakított memórián lehet megfigyelni (4.20. ábra). Ahány bites a léptetőregiszter, annyi szavas a tároló, a szóhosszúságot pedig a léptetőregiszterek darabszáma határozza meg.



4.20. ábra

A léptetőregisztereket folyamatosan léptetik, az adatok körben haladnak. A regiszter végére érve az adatok a visszacsatoló vezetéseken a címző/vezérlő egységbe jutnak, ebben a helyzetben lehet az adatot írni vagy olvasni; majd a tárolt bitek ismét belépnek a regiszterekbe. A soros elérés ténye jól megfigyelhető. Ha olyan rekeszt címzünk, ami éppen belép a címző/vezérlő egységbe, azonnal rendelkezésre áll, de ha éppen most lépett be a regiszterekbe, meg kell várni, míg végighalad a teljes regiszteren.



4.21. ábra.

A léptetőregiszteres megoldást memóriacsipként nem gyártják, de vannak más félvezetős megoldások, melyek tulajdonképpen ezt az elvet alkalmazzák, a felépítésük szinte lemásolja a léptetőregiszteres kapcsolást. A *mágneses buborékmemória* (Magnetic Bubble Memory, MBM) esetében a csip felületén mágneses réteget hoznak létre, és mágneses csomagocskák (domén, buborék) halad végig a csip körül kialakított vezetékrendszerben alkalmazott léptetőáram mágneses tere hatására. A csip bal szélén (4.21. ábra) a magnetofon írófejéhez hasonló egység hozza

létre a logikai állapotnak megfelelő mágneses csomagot, amit a jobb szélén a fordított átalakító (az olvasófej) ismét logikai jellé konvertál vissza. Egyébként a kialakítás és a működés hasonló a léptetőregiszteres áramkörhöz. Ha 262 144 lépés alatt jut át a bal szélről a domain a jobb szélre, akkor az MBM 256 KiB szót tárol, ha 8 mágneses ösvény van a felületen, 256 KiB a kapacitása. Mivel a mágneses domain léptetőjelek, tápfeszültség nélkül megmarad a felületen, az MBM nem illanó memória.

Egy másik soros hozzáférésű félvezetős memória a CCD (Charge Coupled Device), a *töltéscsatolt áramkör*. Ennél a léptetőregiszterek helyett a csip felületén villamos töltéscsomagokat léptető pályákat alakítanak ki, a töltéscsomagokat a félvezetőt körülvevő vezetékhalózatra vezetett feszültség villamos tere lépteti végig a felületen. Az áramkör felépítése hasonló a 4.21. ábrán látotthoz, csak az U/M átalakító helyén U/Q; az M/U konverter helyén Q/U elem szerepel benne. A villamos töltést a léptetőjel tartja egy csomagban, ha az megszűnik, a töltés eloszlik a felületen. Eszerint a CCD-memória léptetés nélkül nem őrzi meg az adatokat, illanó memória.