

Budapesti Műszaki Egyetem
Híradástechnikai Elektronika Intézet

dr. Jagudits László

FÉLVEZETŐS MEMÓRIÁK

Félvezetős memóriák

1. A memóriák áttekintése
 - 1.1 A félvezetős memóriák jellemzése
 - 1.2 A félvezetős memóriák felosztása
 - 1.3 Alkalmazási területek
2. Véletlen hozzáférésű memóriák
 - 2.1 Memória morzsák szervezése
 - 2.2 Memóriák kialakítási szempontjai
3. Soros hozzáférésű memóriák
 - 3.1 Soros hozzáférésű memória morzsák kialakítása
 - 3.2 Soros hozzáférésű memóriák kialakításának szempontjai
4. Permanens memóriák
 - 4.1 Permanens memória morzsák kialakítása

Félvezetős memóriák

1. A memóriák áttekintése

A digitális információ tárolására alkalmazott memóriák számos típusa alakult ki attól függően, hogy milyen mennyiségű információ tárolására van szükség (memória kapacitása), mekkora idő alatt lehet elérni a keresett információt (hozzáférési idő), mennyire megbízható a tárolás és nem utolsósorban lényeges az információ egységre jutó ár. Ezek a követelmények általában ellentmondóak, azért a memóriák kialakítása során megfelelő kompromisszumra kell törekedni.

A digitális információ legelemibb egysége a bit, több bitből álló csoport a byte (8 bit), valamint a szó (ennek hossza számítógépenként változhat, gyakori a 16 ill. 32 bites hosszúságú szó). A memória kapacitását az előző egységek valamelyikében szokás megadni, pl.:

1 Kbyte=1 kilo byte= 2^{10} byte=1024 byte,

1 Mbyte=1 megabyte= 2^{10} Kbyte

A memóriában az információt sorszámozott rekeszekben helyezünk el azért, hogy az a sorszám alapján visszakereshető legyen. Ezt a sorszámot címnek, azt az időt pedig ami a visszakeresési parancs kiadása és az információ megjelenése között eltelik hozzáférési időnek nevezzük.

A különböző tulajdonságú, mind kapacitásban mind hozzáférési időben széles skálán mozgó memóriák döntő többsége vagy mágneses anyagot vagy valamilyen elektronikus áramkört használ az információ tárolás megvalósítására. Az előzőek ún. négy-szöghiszterézisű lágymágneses anyagok két stabil mágneses állapot segítségével rögzítik az információt. Két alapvető típusuk alakult ki: az álló mágneses tárolók kisebb kapacitás mellett nagyobb sebességet biztosítanak. Legfontosabb típusuk a ferritgyűrűs memória volt. A mozgó mágneses memóriák nagy kapacitásúak, sebességük viszont lényegesen kisebb. Legfontosabb típusai a mágneslemez és mágnesszalag memória. A ferromágnességet hasznosító memóriák közös sajátossága, hogy az anyag egy belső tulajdonságát használják fel tárolásra; ez-

zel szemben a memóriák másik nagy csoportja aktív és passzív áramköri elemekből felépített elektronikus áramkörökkel biztosítja az információ megőrzését. Az áramkörökben alkalmazott aktív és passzív elemek is félvezetőkől készülnek valamilyen integrált áramköri technológia szerint, ezért ezeket félvezetős memóriáknak nevezzük.

1.1 A félvezetős memóriák jellemzése

A félvezetős memóriák számos előnyös tulajdonsággal rendelkeznek.

Nagy sebesség, azaz kis hozzáférési idő érhető el velük. Tovább növeli a sebességet az a tény, hogy a félvezetős memóriák egy része nem törölődő kiolvasású (ez azt jelenti hogy egy adott címen lévő információ kiolvasásakor az információ a rekeszből nem törölődik. A memóriák egy részénél ez nem így van, ezért ott az olvasást követően egy visszaírással van szükség, hogy az információ megmaradjon), ezért az elmaradó visszaírási ciklus további sebesség növekedést jelenthet. A megfelelő integrált áramköri technológia is segít a hozzáférési idő csökkentésében, de ugyanakkor biztosítja a tö-

meggyárthatóságot, ami az ár csökkentésének és a megbízhatóság növelésének fontos feltétele. A digitális berendezések többi áramkörét is hasonló vagy azonos technológiával készítik és így a kompatibilitás miatt egyszerű a memóriákhoz való illesztés, de a tervezés, üzemeltetés és javítás szempontjából is azonosak a követelmények. Az integrált áramköri megvalósítás kis méretet eredményez, azaz nagy a tárolási információ sűrűség. Kis méretű félvezető mozsán csak akkor lehet nagy áramkör sűrűséget elérni, ha a hődisszipáció kicsi, ami végeredményben kis energia igényt jelent.

Természetesen a fenti előnyös tulajdonságok mellett néhány hátrányos jellemzőt is figyelembe kell venni. Ezek közül a legfontosabb a tápfeszültség kimaradásakor fellépő információvesztés (ez nem lép fel a mágneses elven működő memóriáknál), melyen csak akkumulátorból ill. telepekből álló segédáramforrással lehet segíteni, amelyek biztosítják a hálózat kimaradásakor a folyamatos energiaellátást.

A félvezetős memóriát tartalmazó integrált áramköri tokok a szokásos nagy bonyolultságú integrált

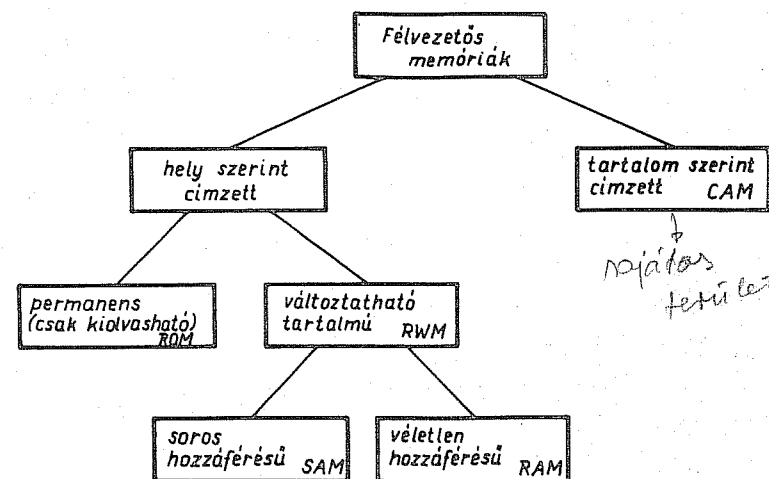
áramkörök (LSI) megbízhatóságával rendelkeznek, amely egy token belüli kapura vagy funkcióra vetítve igen kedvező érték összevetve a kis illetve közepes bonyolultságú integrált áramkörök azonos jellemzőjével. A jelenlegi megbízhatóság mégis kevésnek bizonyul nagy kapacitású (10^7 bit felett) memóriák kialakításánál. Ilyen esetekben hibajavító kódok alkalmazása célravezető, ami természetesen redundáns bitek tárolását és a hibajavításhoz szükséges többlet áramkörök beépítését jelenti, megnövelve ezzel a memória árát. A félvezetős memóriák ára állandóan csökken, azonban az ezekből felépített memóriák áránál természetesen figyelembe kell venni a gyártás során fellépő összes költséget:

- a./ a memória és kiegészítő áramkörök ára
- b./ a memória és kiegészítő áramkörök (IC tokok) vizsgálatának költsége
- c./ a nyomtatott áramköri lap előállításának és vizsgálatának ára
- d./ a szerelés költsége
- e./ a szerelt kártya bemérésének és esetleges javításának (hibás IC, hibás nyomtatás stb.) költségei

1.2 A félvezetős memóriák felosztása

A címzés módja szerint hely és tartalom szerint címzett memóriákat különböztetünk meg (1. ábra). Amikor egy információt az azt tartalmazó rekesz sorszáma (címe), azaz az információ helye szerint keresünk vissza, akkor hely szerint címzett memóriáról beszélünk. Tartalom szerint címzett (CAM: Content Addressable Memory) a memória akkor, ha az információt a tartalma vagy tartalmának egy része alapján keressük vissza. Ebben az esetben az a legfontosabb, hogy megtalálható-e ez az információ a tárolóban. Erre a kérdésre a hely szerint címzett memóriánál csak úgy válaszolhatunk, ha minden rekesz tartalmát visszakeressük és azt összehasonlítjuk a keresett információval, tehát a feladat elvégzése nagyon időigényes. Ezzel szemben a tartalom szerint címzett memóriák rendszerteknikai felépítése olyan, hogy a tárolt információk visszakeresése nélkül, egy lépésben választ kapunk a fenti kérdésre.

A tárolás jellege alapján permanens (csak kiolvasható; ROM: Read Only Memory) valamint változtatható tartalmú memóriákról beszélhetünk. A per-



1. ábra

Félvezetős memóriák felosztása

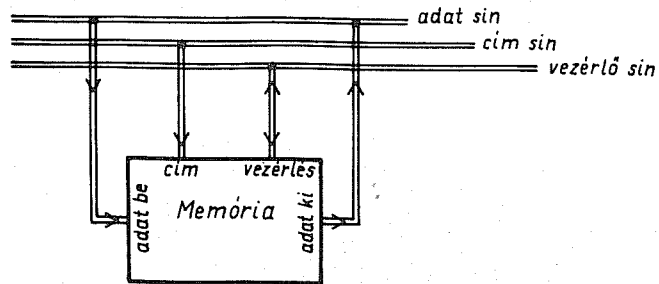
manens memóriák információ tartalma rögzített (működés közben nem változtatható), így üzemszerű állapotban az információt csak kiolvasni lehet belőlük. Ezzel szemben a változtatható tartalmú memóriák üzemszerűen az információ beírására és kiolvasására (író-olvasó memória; RWM: Read Write Memory) is alkalmasak. A hozzáférés módja szerint véletlen hozzáférésű és soros hozzáférésű memóriák léteznek. A véletlen hozzáférésű memóriák (RAM: Random Access Memory) esetében bármely rekesz információ tartalma átlagosan ugyanannyi idő alatt érhető el, míg a soros hozzáférésű memóriáknál (SAM: Serial Access Memory) a hozzáférési idő különböző attól függően, hogy hol helyezkedik el a keresett információ, mert az egyes rekeszek tartalmához csak sorban egymás után férhetünk hozzá.

A megvalósításhoz felhasznált félvezető eszközök szerint beszélhetünk bipoláris, unipoláris (MOS), amorf és optoelektronikus félvezető eszközökről. Az első kettőnek nagy a gyakorlati jelentősége, az utóbbiakkal készített memóriák a fejlesztés stádiumában vannak.

1.3 Alkalmazási területek

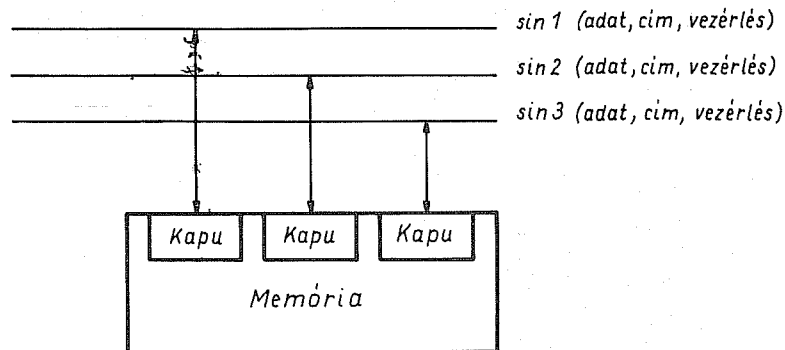
A memóriákat a digitális technika minden területén elterjedten alkalmazzák. Legnagyobb felhasználó a számítógép ipar, de a digitális mérő és adatgyűjtő, adatátviteli és egyéb célberendezésekben is alkalmazásra kerülnek. A számítógép különböző részeiben a legkülönbözőbb kapacitású és hozzáférési idejű memóriákat használnak, vonatkozik ez a tartalom címezhetőségű, permanens, véletlen hozzáférésű és soros hozzáférésű memóriákra egyaránt, attól is függően, hogy nagy-, közepes vagy kis univerzális vagy folyamatirányító vagy ügyviteli számítógépről van szó.

A hely szerint címzett memóriák sín szervezésű rendszerben a 2. ábrán látható formában jelennek meg, eszerint rendelkeznek cím-bemenetekkel, amely megjelöli a keresett információ helyét; adat-bemenetekkel és adat-kimenetekkel (értelemszerűen a ROM csak kimenettel), amelyeken át a rendszer közli a beírandó ill. megkapja a kiolvasott digitális információt; vezérlő be- és kimenetekkel, melyek kijelölik az információcsere irá-



2. ábra

Memória kapcsolódása sinszervezésű berendezésben



3. ábra

Többkapus memória

nyát (írás v. olvasás), jelzik a kiszolgálás kérését stb. ill. a memória foglaltságot (kiszolgálásra készséget). A vezérlőjelek száma és iránya a berendezésben kialakított információcsere módjától függ, ami többféle lehet. Az adat, cím és vezérlő vezetékek elektromos kialakításának leggyakoribb formája a sín amelyhez a memória kapcsolódik.

Előfordul, hogy egy memória egyszerre több sínrendszerrel is kapcsolatban áll és mindegyik felől érkező kéréseket ki kell szolgálnia. Ebben az esetben többkapus memóriáról (multiport memory) beszélünk (3. ábra)

2. Véletlen hozzáférésű memóriák

A véletlen hozzáférésű memóriák a változtatható tartalmú memóriák azon csoportját alkotják, amelyeknél az információ hozzáférési ideje annak fizikai helyétől (azaz a címtől) független.

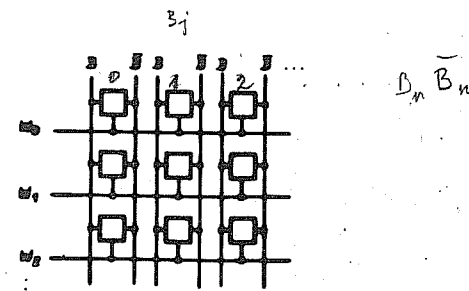
A véletlen hozzáférésű memóriák készülnek mind bipoláris, mind MOS technológiával. Az előbbieken a biteket tároló cellák bistabil multivibrátorok,

az utóbbiakban vagy bistabil áramkörök (statikus MOS RAM), vagy kapacitásban töltésként tárolják az információt (dinamikus MOS RAM). Ez a kapacitás a szivárgási áramok miatt kisül, ezért az információ megőrzése érdekében adott időközökben a kapacitás töltésének pótlása, az információ felfrissítése szükséges.

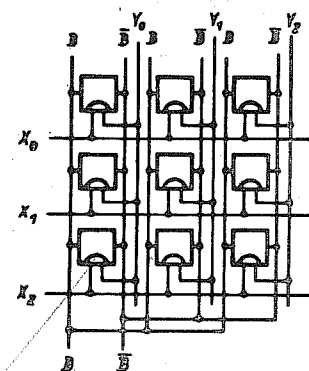
2.1 Memória morzsák szervezése

A memória morzsák (chip) kialakításánál fontos szempont a cellák megfelelő elrendezése, amely kihatással van a hozzáférési időre, a szükséges kiszolgáló áramkörök számára és így végül a memória árára.

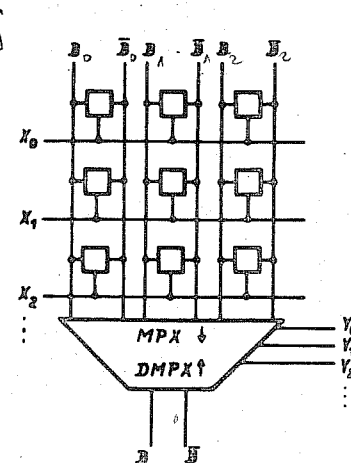
A cellák összetartozó csoportjai tárolják a véletlen hozzáférésű memória szavait, amelyeknek kiválasztásakor az információ beírható ill. kiolvasható. Két alapvető szervezési elv szerint alakíthatók ki a cellákból a memória szavai. A 4. ábra mutatja az un. szószervezésű memória elvi felépítését. A W_i szóvezeték aktivizálásakor a hozzákapcsolódó, az egy szóhoz tartozó cellákat választjuk ki, és ekkor ezen cellák tartalma a



4. ábra
Szószervezésű memória



5. ábra
Bitszervezésű memória
egy síkja



7. ábra
Módosított bitszervezés
elvi vázlata

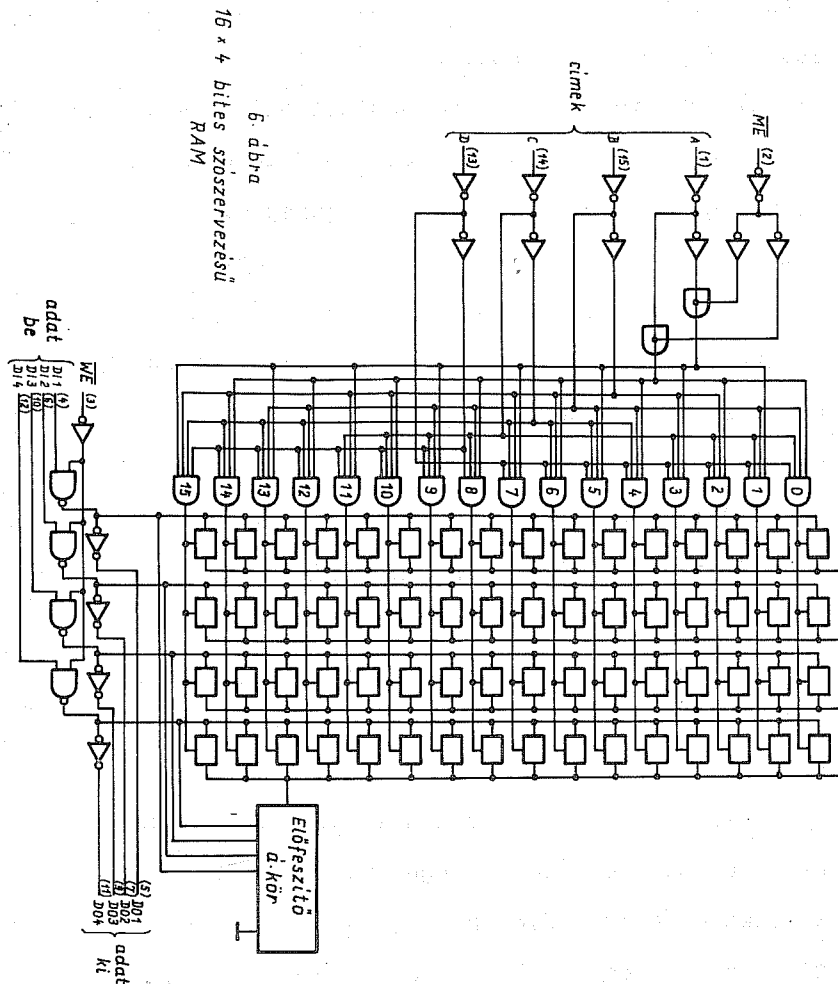
B_j és $\overline{B}_j$ bitvezetékeken párhuzamosan kiolasható ill. beírható.

Az 5. ábrán az ún. bitszervezésű memória egy síkja látható. Egy-egy sík a szavaknak csak egy bitjét tartalmazza, tehát a memória annyi síkból áll, ahány bites a szó. A szavak kiválasztása az X_i és Y_j vezetékek egyidejű aktivizálásával történik valamennyi síkon. A cellákba az ábrán be-rajzolt ÉS kapu gondoskodik a sík egyetlen cellájának kiválasztásáról. A cellánkénti ÉS kapcsolat megvalósítása és az Y vezeték felvitele többlet hely-igénnyel jár, ezért a technológiai okokból korlátozott morzsa méret biztosította félvezető felület rosszabb kihasználását teszi lehetővé. A morzsa méret korlátozottsága miatt a nagyobb kapacitású memóriákat meg kell osztani és több morzsából kell összeállítani. A morzsákat tokozzák és belül a kivezetéseket elkészítik. A morzsán lévő csatlakozási pont és a kivezető láb összekötése és maga a tokozás is költséges (az összes gyártási költség fele), ezen kívül a lábszám csökkentése a megbízhatóságot és a végső kihozataalt is javítja, ezért célszerű a szükséges kivezetések (lábak) számát minimális értéken tartani.

A félvezetős memóriák fejlesztésének első szakaszában csak a cellák (16-64 db) kerültek egy morzsára. Így memória készítésekor külső áramkörökkel kellett biztosítani a dekódolást (kiválasztást), az érzékelést és a szükséges időzítéseket. A viszonylag hosszú vezetékek meghajtása miatt lassult a működés és nagy meghajtó teljesítmény volt szükséges.

A morzsaméret növelése, a cella területének csökkentése és a technológiai fejlődés fokozatosan oda vezetett, hogy ma már egy morzsára 64K-4 Mbitnyi cella is elfér. A cellák számának növelése jelentősen növelné a kivezetések számát, ezért szükségessé vált a dekódoló áramköröknek a cellákkal egy morzsán történő elkészítése. Ugyancsak célszerű az érzékelő áramkört és az időzítő áramköröket is integráltan ugyanarra a morzsára elkészíteni.

Kis kapacitású morzsáknál előnyösen alkalmazható a szószervezésű megoldás anélkül, hogy a lábszám megnőne, ilyen például a 6. ábrán látható 16x4 bites TTL RAM. A cím bemeneteket két inverter követi részint a negáltak előállítására, részint a belső terhelés leválasztására. A memória engedélyező ($\overline{ME}$) bemenet az egész egység működését engedélyezi ill. letiltja, így több morzsából

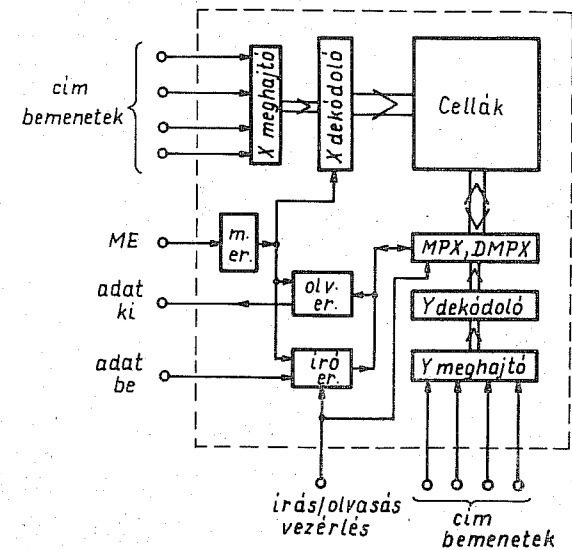


(tokból) álló memória tömbnél csak azokat a morzsákat kell működtetni, amelyekben lévő információt keressük (különösen a MOS RAM cellák esetében jelentősen különbözik a nyugalmi ill. a működés alatti disszipáció, ezért $\overline{ME}$ bemenet alkalmazásával biztosítható a rendszer minimális disszipációja.) A kimenet nyitott kollektoros kapcsolású, ami biztosítja több ilyen kimenet párhuzamos kapcsolhatóságát, melyre nagyobb kapacitású memória tömb kialakításánál feltétlen szükség van (más tokoknál a három állapotú kimenet ugyanezt szolgálja). Az írás engedélyezés ($\overline{WE}$: Write Enable) vezérlő bemenet segítségével választhatjuk ki, hogy írni v. olvasni akarunk a megcímzett szóból.

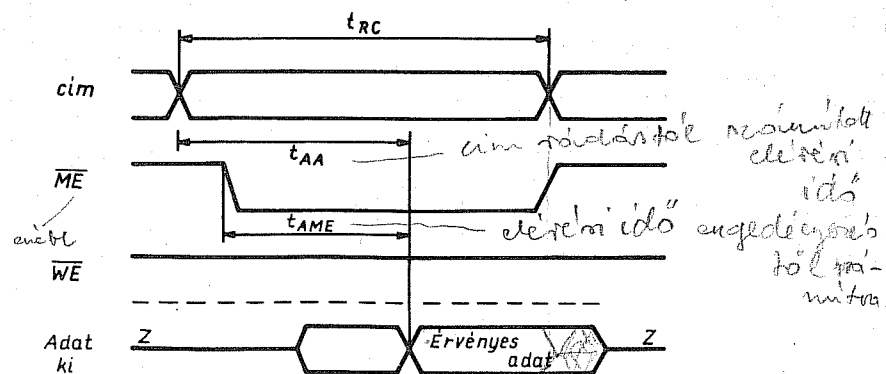
Nagyobb kapacitású morzsák kialakításánál a szöszervezés már nem megfelelő. A klasszikus bitszervezés az előzőekben említett hátrányok miatt csak kis kapacitások esetén jó, ezért nagyobb kapacitású morzsák esetén módosított bitszervezést célszerű alkalmazni, melynek elvét a 7. ábra mutatja. A kiolvasás tulajdonképpen úgy történik, mint a szöszervezésű memóriánál (X_i), de a bitvezetékekre elhelyezett kiválasztó áramkör (mul-

tipler) csak annak az egy bitnek az értékét engedélyezi az olvasó erősítőre, amelyiket az Ycím engedélyezi. Természetesen a beíráshoz biztosítani kell az információ áramlását az ellentétes irányba egy szétosztó áramkör (demultiplexer) segítségével. Az eddigiek alapján egy memória morzsa (tok) rendszertechnikai felépítését felrajzolhatjuk módosított bitszervezés esetére (8. ábra). Láthatóan a lábszámot így alacsony értéken lehet tartani, egy 1 Kbites lapka esetén szükséges 10 cím bemenet, 1 memória engedélyező, 1 adat bemenet, 1 adat kimenet, 1 írás/olvasás vezérlő bemenet és a tápfeszültség bemenetek, így 16-18 lábú tok alkalmazható. A módosított bitszervezés általánosan elterjedt, a memóriamorzsa döntő többsége ezt a felépítést követi.

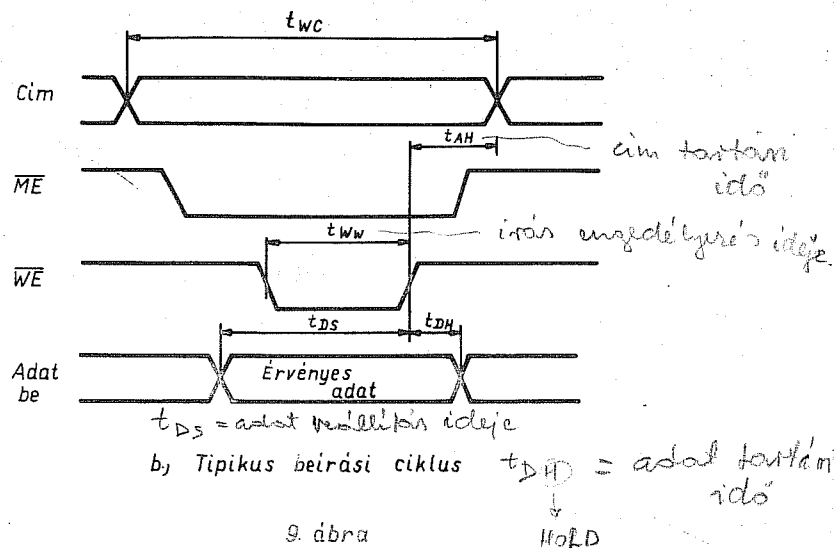
Olvasásuk, beírásuk tipikus idődiagramjait mutatja be a 9. a, b ábra. Az $\overline{ME}$ jel a címmel együtt v. ezt követően válthat aktívá. Ennek hatására a kimenet engedélyeződik, de a helyes adat csak a hozzáférési idő letelte után jelenik meg az adat kimeneten (cím hozzáférési idő t_{AA} : address acces time; $\overline{ME}$ hozzáférési idő t_{AME} ; olvasási



8. ábra
Módosított bitszervezésű memória morzsa
rendszertechnikai felépítése



a, Tipikus olvasási ciklus
 $t_{AME} < t_{AA}$



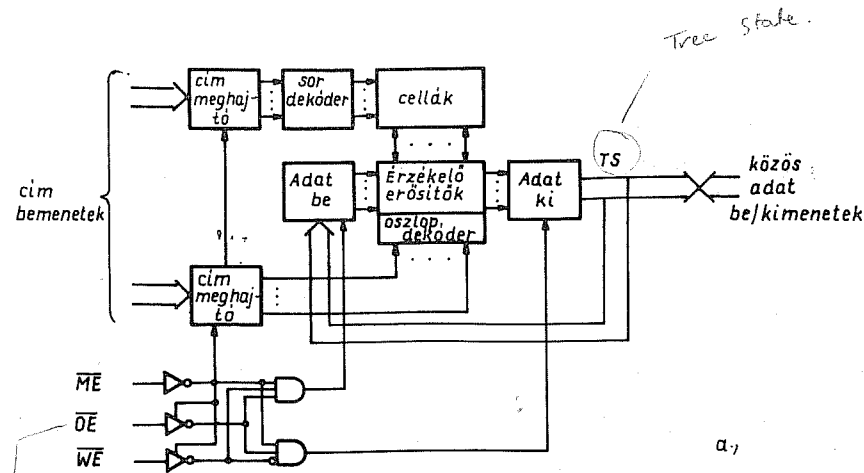
9. ábra
 Tipikus olvasási, beírási ciklus

ciklusidő t_{RC} : read cycle time). Az érvényes adat mindaddig a kimeneten marad amíg $\overline{ME}$ aktiv és a cím változatlan.

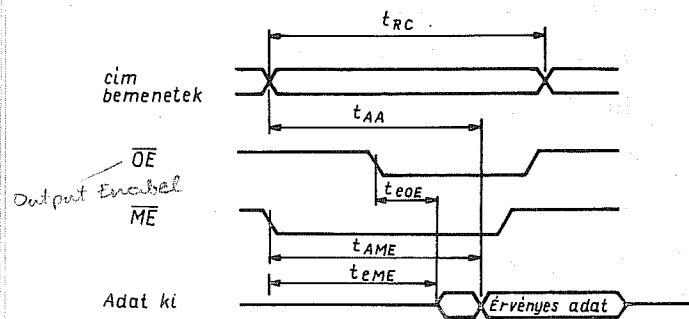
A beírás vezérlési diagramjából leolvasható, hogy a beírás engedélyezése ($\overline{WE}$ aktiv) $\overline{ME}$ aktiv értéke alatt történik. Az adat beírását $\overline{WE}$ felfutó éle végzi, ezért ehhez az élhez időzítetten kell a beírandó információt az adatbemenetre kényszeríteni (adat előkészítési idő t_{DS} : data set-up time; adat tartási idő t_{DH} : data hold time; írási ciklusidő t_{WC} : write cycle time). A lábszámot csökkentheti, ha közös adat be/kimenetet használunk, hiszen az írás/olvasás vezérlőjel eldönti a morzsán belül az információ áramlás irányát. Ilyen típusú morzsák sínezett adatvezeték esetén jól használhatóak. Ilyen morzsa felépítésére mutat példát a 10. ábra. Ezen külön kimenet engedélyező vezérlő jelet ($\overline{OE}$: Output Enable) is találunk, amely a kimeneti három állapotú (TS: Three-State) meghajtókat engedélyezi. Ezen meghajtók engedélyezési ideje (t_{eOE} : engedélyezési idő $\overline{OE}$ vezérlőbemenetről; t_{eME} : engedélyezési idő $\overline{ME}$ vezérlő bemenetről) értelemszerűen jóval rövidebb a hozzáféré-

si időknél (t_{AA} ; t_{AME}), mint az a 10.b. ábrán látható.

64 Kbytes-1Mbytes dinamikus MOS RAM morzsák esetében a cím bemenetek száma tovább növekszik. A lábak számának jelentős csökkentését ilyenkor cím multiplexeléssel lehet megoldani, ez azt jelenti, hogy a cím bemenetek számát a felére csökkentjük (pl. 256K esetén 9 címbemenet), és a morzsán az X ill. Y meghajtók előtt egy-egy átmeneti tárolót (két 9 bites címregiszter) alakítunk ki, amelyeket külső kapuzójellel működtetünk (plusz 2 láb), ezeket $\overline{ME}$ helyett is használhatjuk: így végül a 256 Kbit is elhelyezhető egy 16 lábú tokban. Természetesen a tokra vonatkoztatott írás/olvasási idő megnő, mivel a címet csak két lépésben írhatjuk be az átmeneti tárolókba. Erre a felépítésre jellemző példát mutat a 11. ábra, amely 256 Kbytes dinamikus RAM. A tároló cellák nyolc 32 kbytes (64 sor x 256 oszlop) tömbben helyezkednek el. Két-két 32 Kbytes tömbhöz tartozik közösen 256 érzékelő erősítő, tehát négy 64 Kbytes tömbből lehet egyszerre egy-egy bitet kiolvasni; a legmagasabb két címbit (kétszer A8) választja ki végül a kimenetre kerülő egyetlen adatbitet.



lineneti TREE STATE változó

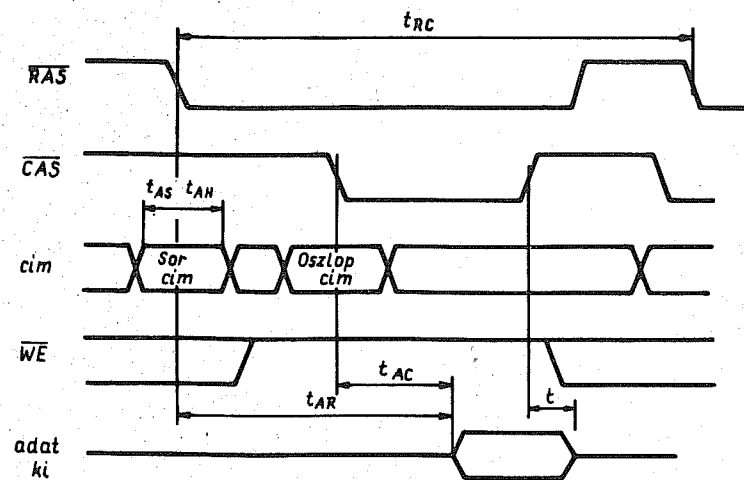


10. ábra

Közös adat be/kimenetű memória

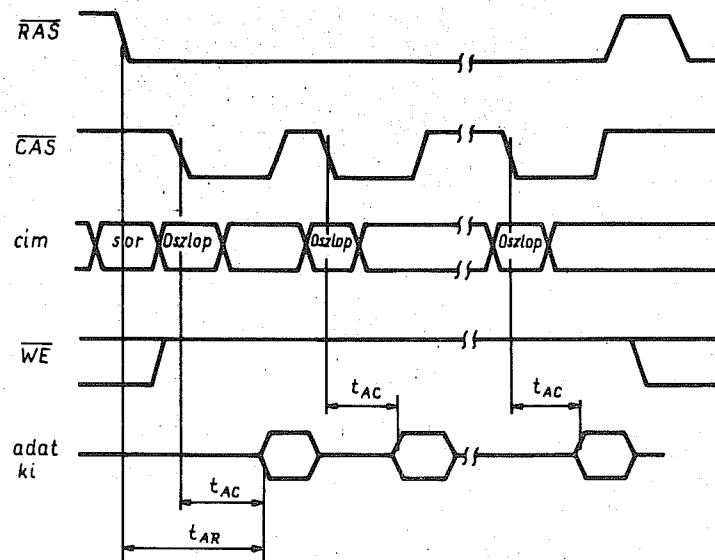
a, rendszerteknikai felépítése

b, olvasási ciklusa



12. ábra

Címultiplexelt dinamikus RAM olvasási ciklusa

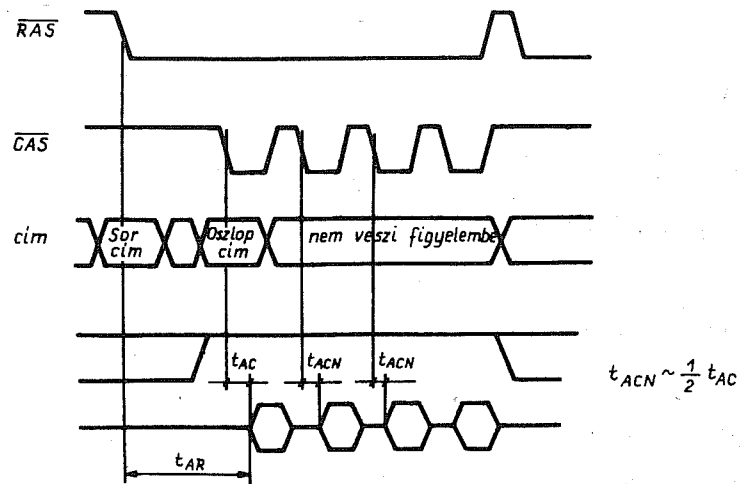


13. ábra

Azonos lapról olvasás

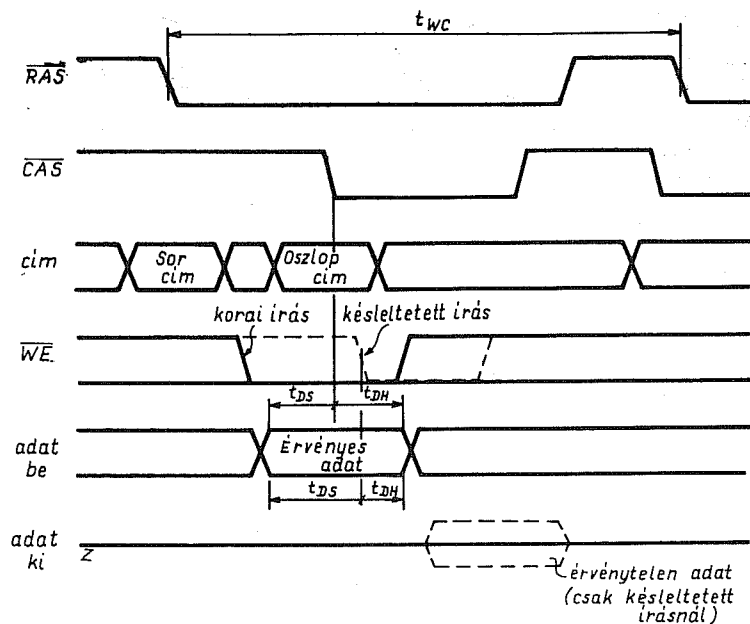
Egy speciális gyors soros hozzáférés (nibble mode) is lehetséges (14. ábra). Ezt az teszi lehetővé, hogy egyszerre négy bit kiolvasása történik meg és ezek egy kimeneti regiszterbe kerülnek, amelyből egyet a kétszerre beolvasott A8 bit választ ki. Az első bit kiolvasása ugyanúgy történik, mint rendes olvasásnál, majd ezt követően $\overline{RAS}=L$ alatt a $\overline{CAS}$ jel mozgatásával kiléptethetjük a többi három bitet is. A kiléptetés A8R, A8C: 00, 01, 10, 11, 00 stb. szekvenciában történik, a kezdőértéket az első bit olvasásakor állíthatjuk be. (Ugyanazon a morzsán vagy az azonos lapról olvasás vagy a gyors soros hozzáférés valósítható meg - ez gyártott típusonként változhat!)

Írási ciklusban a címek beírása ugyanúgy történik, mint olvasásnál. (15. ábra) $\overline{WE}$ vezérlőjelet aktiválhatjuk $\overline{CAS}$ aktív éle előtt (korai írás: early write; az ábrán folytonos vonal), vagy után (késleltetett írás). Az előbbinek előnye, hogy az adat kimenet mindvégig nagy impedanciájú állapotban marad és ezért az adat be- és kimenet összeköthető, azaz egy sínvezetékre köthető. A beírandó adatot most is megfelelő előkészítési és



14. ábra

Olvasás gyors soros hozzáféréssel (nibble mode)

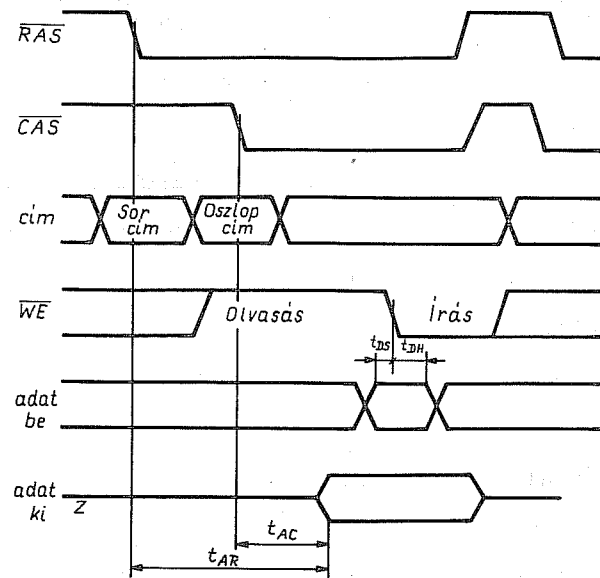


15. ábra

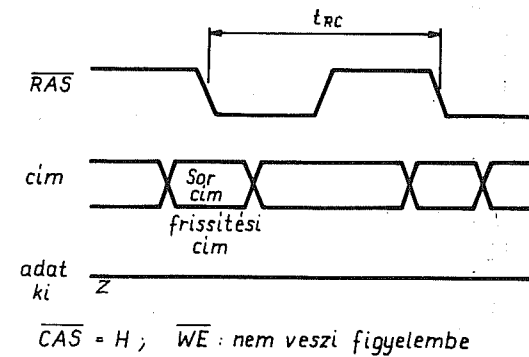
Címultiplexelt dinamikus RAM írási ciklusai

tartási idő betartásával kell a bemenetre kapcsolni. Korai írásnál a $\overline{\text{CAS}}$, míg késleltetett írásnál $\overline{\text{WE}}$ lefutó éle kapuzza be az adatot. Speciális írási ciklusok: azonos lapra írás (page mode), írás gyors soros hozzáférésben (nibble mode) valamint az olvasással megegyező címre történő visszaírás (read-modify-write: olvasás-módosítás-visszaírás). Mint azt a 16. ábra is mutatja, a címbeírás és olvasást közvetlenül egy késleltetett beírás követ változatlan címre. Ilyenkor lehetőség van arra, hogy módosítsuk (megváltoztassuk) az információt és ezt írjuk vissza (pl: kiolvasás, hibadetektálás, ha kell hibajavítás, visszaírás).

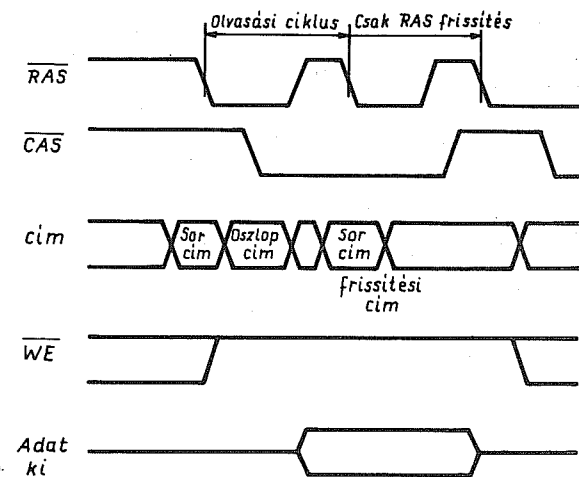
Mint már korábban említettük a dinamikus RAM tartalmát állandóan frissíteni kell, hogy az információ ne vesszen el. Ez azt jelenti, hogy minden bitet legalább 2(4; technológiától függ) ms-onként frissíteni kell. Vizsgált mintaáramkörünkben (11. ábra) a 64 kbit-es tömbök ($2 \times 128 = 256$ sor; 256 oszlop, oszloponként egy érzékelő erősítővel, amely a frissítést is elvégzi) egyszerre frissítődnek, minden tömbben egyszerre egy teljes sor



16. ábra
Olvasás - módosítás - visszairás



17. ábra
Dinamikus RAM frissítése
(csak $\overline{RAS}$ frissítés)



18. ábra
Dinamikus RAM rejtett frissítése

(256 cella); azaz a morzsa valamennyi cellája 256 lépésben frissíthető fel úgy, hogy kívülről kell kijelölni a frissítendő sor címét (A0-A7). (Vannak olyan morzsák is, amelyekre felviszik a frissítési címszámlálót is). Valamennyi olvasási, írási művelet a megcímzett sorokat egyben frissíti is. A frissítési ciklus időbeli lefutását a 17 ábrán láthatjuk (csak $\overline{\text{RAS}}$ jellel vezérelt frissítés; $\overline{\text{RAS}}$ -only refresh). A 18. ábra rejtett frissítésre (hidden refresh), mutat példát. Itt egy memória ciklus után, miközben az adatot tartósan a kimeneten tartjuk ($\overline{\text{CAS}}=\text{L}$ értékkel) végzünk frissítést a $\overline{\text{RAS}}$ vezérlőjel segítségével. $\overline{\text{RAS}}$ lefutó élei írják be a frissítési címeket.

2.2 Memóriák kialakítási szempontjai

Mint azt az előző fejezetben már láttuk a félvezető morzsák mérete miatt egy token belül csak korlátozott számú bit valósítható meg, ezért általában egy memória - a szükséges kapacitástól függően - csak több tokból állítható össze, valamint ezeket ki kell egészíteni a kiszolgáló áramkörökkel (címkezelő, adatkezelő, vezérlő áramkö-

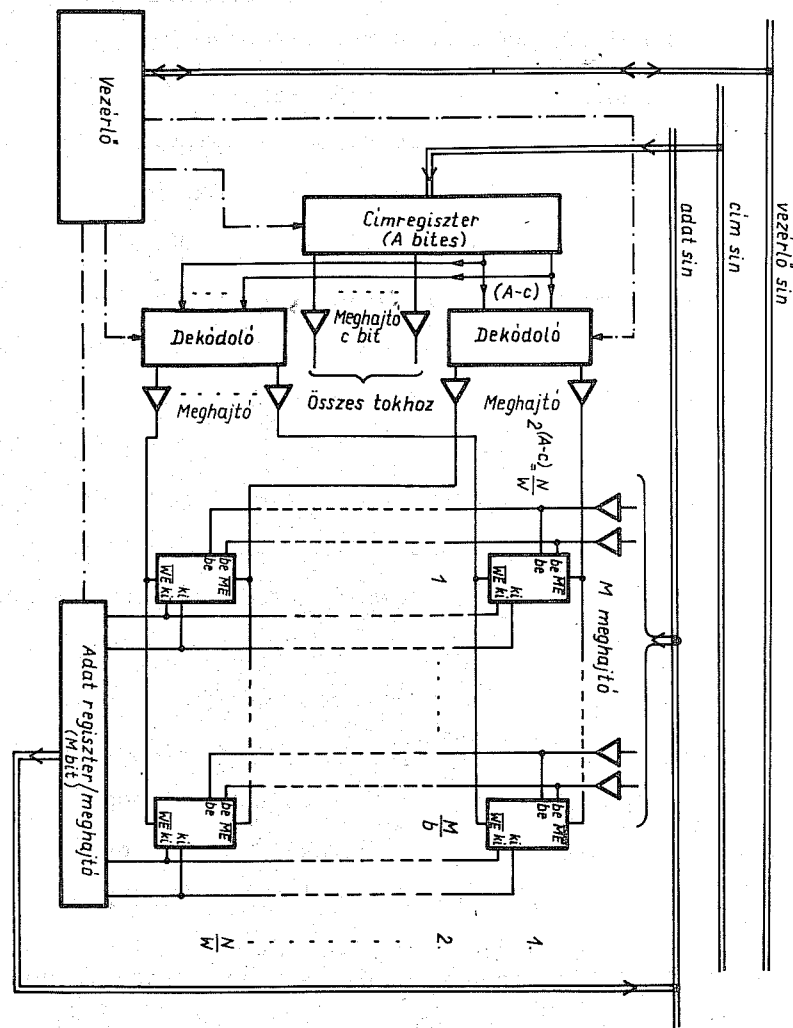
rök), amelyek biztosítják az illesztést a berendezés többi részével és a memória áramkörök megfelelő meghajtását az előírt időzítésekkel.

Az általában szükséges kiszolgáló áramköröket statikus RAM-ok esetén a 19 ábrán láthatjuk. A memória tokoktól (TTL, ECL, MOS) és kiszolgáló áramköröktől függően a meghajtó és szintillesztő fokozatok el is maradhatnak, ezt mindig a konkrét viszonyok határozzák meg.

Ha egy N szavas, szavanként M bites memóriára van szükségünk, akkor egy olyan w szavas, szavanként b bites memória tokot célszerű választani ahol N w -nek ill. M b -nek egészszámu többszöröse, mert így nem lesznek kihasználatlan bitek. A tokokban lévő szavak számára mindig igaz, hogy $2^C=w$ (C : egy tok címezőbemeneteinek száma), de N értékét is célszerű úgy megválasztani, hogy $N=2^A$ (A : a memória cím bitjeinek száma), így végül a memória tokokat egy N sorból és M oszlopból álló tömbben kell elhelyezni.

Terhelés szempontjából a legkedvezőtlenebb a helyzet azoknál a cím vezetékeknél, amelyek az összes tokhoz vezetnek, hiszen ezeknek egyenként

19. ábra
Memória rendszertechnikai felépítése (statikus RAM)



$\left(\frac{N}{W} \cdot \frac{M}{b}\right)$ bemenetet kell meghajtaniok mind statikusan, mind dinamikusan (a bemenetek kapacitív terheléssel is rendelkeznek, a fel- és lefutási idők a legtöbb bemenetnél előírtak a helyes működtetés biztosítása érdekében). A dekódolt cím vezeték - melyek egy-egy sor kiválasztását biztosítják $\overline{ME}$ -n keresztül - csak $\frac{M}{b}$ tokot hajtanak meg egyszerre. A $\overline{WE}$ vezérlőjelet vagy elvezetjük minden tokhoz, vagy ezt is dekódoljuk, így egyrészt kisebbek a követelmények a meghajtó áramkörrel szemben, másrészt kisebb a teljesítmény felhasználás.

A kimenetek terhelhetősége szempontjából érdekes a tokok párhuzamosan kötött kimeneteinek száma, amely $\frac{N}{W}$ (a nyitott kollektoros és a háromállapotú kimeneteknél is fontos a méretezéshez).

A memória hozzáférési idejét a kiszolgáló áramkörök is befolyásolják, hozzáférési idő alatt a cím és az adat megjelenése közti időt értve. Két úton járhatunk a címregisztertől a kimenetig, ezek mentén a késleltetés: (1. a 20. ábrát)

$$t_{HA} = t_{mA} + t_{AA} + t_d$$

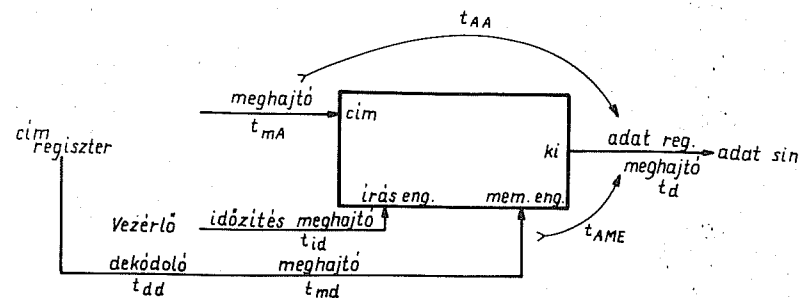
$$t_{HME} = t_{dd} + t_{md} + t_{AME} + t_d$$

A kettő közül a nagyobbik adja a memória hozzáférési idejét.

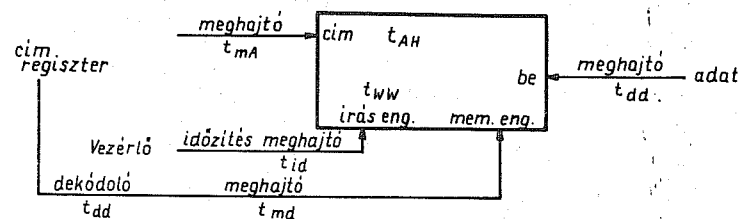
A memória beírási idejét szintén befolyásolják a kiszolgáló áramkörök, mint az a 21. ábrán látható (beírási idő: Írás vezérlőjeltől a cellába történő beírásig eltelt idő). A megbízható beírás megköveteli, hogy az Írás engedélyezés ($\overline{WE}$) adott ideig aktív legyen (t_{ww} közben ME aktív, adat és cím rögzített), és a cím az Írás engedélyezés megszűnte után is stabil maradjon egy ideig (cím tartási idő: t_{AH} ; erre igen gyakran nincs szükség, mert a token belül létrehozott átmeneti tároló a címet stabil értéken tartja). Így a memória írási ideje az ábra alapján:

$$t_{IME} = t_{dd} + t_{md} + t_{ww} + t_{AH}$$

$$t_{IWE} = t_{id} + t_{ww} + t_{AH}$$



20. ábra
Memória hozzáférési idejének összetevői

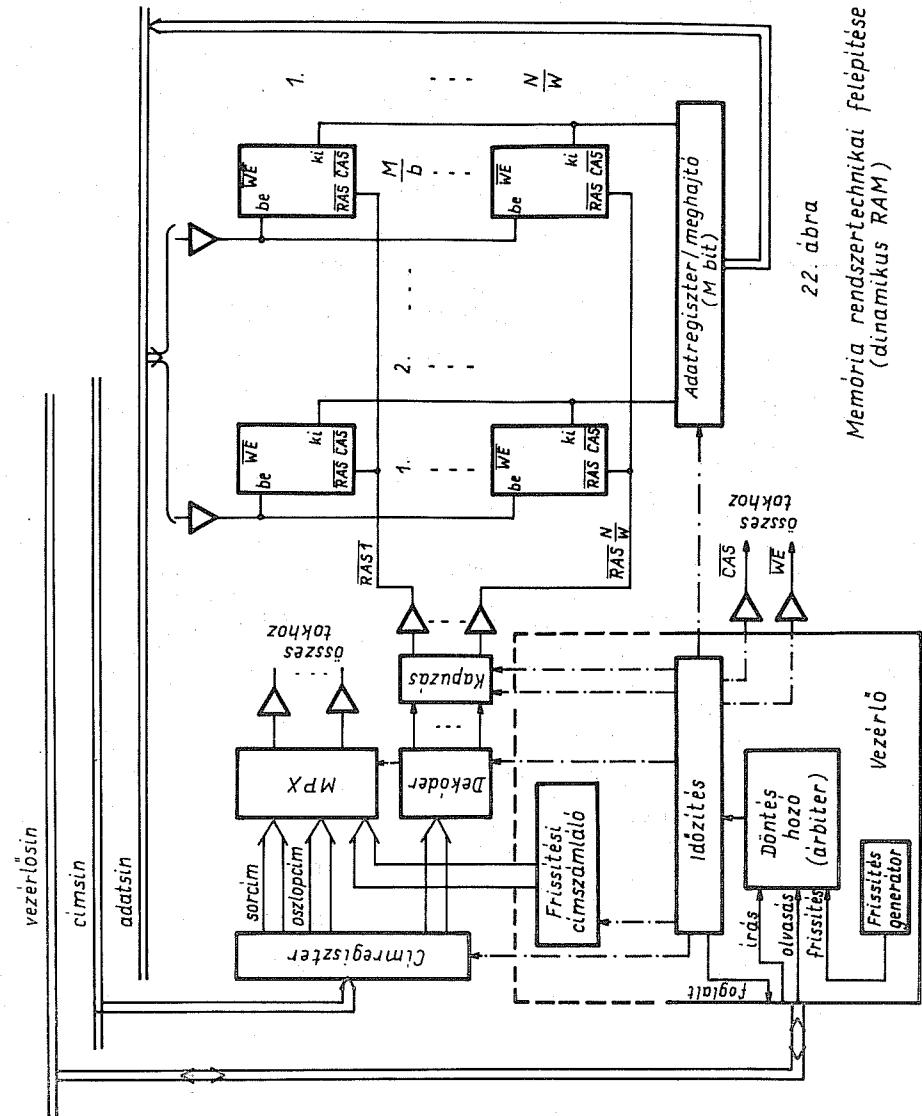


21. ábra
Memória beírási idejének összetevői

A kettő közül a nagyobbikat kell figyelembe venni (elvileg a címhez, ill. az adat bemenethez vezető utak is meghatározhatják a beírás idejét, de az ott lévő késleltetések rendszerint kisebbek).

Ciklusidőnek nevezzük azt az időtartamot amely bármelyik memória művelet teljes befejezéséhez szükséges, tehát hosszabb a memória olvasási, írási idejénél, mert általában bizonyos tartalékokat v. egyéb áramkörök (pl. hibajavító) működéséhez szükséges időt is tartalmazza.

Dinamikus RAM-okból felépített memória rendszertechnikai kialakítása (22. ábra) lényegesen eltér a statikus RAM-okból felépítettektől (19. ábra). Az itt szükséges frissítés, valamint a bonyolultabb (címmultiplexelt) vezérlés miatt a vezérlő rész kialakítása összetettebb. A döntéshozó (arbiter) oldja fel az egyidejű frissítés ill. írás/olvasás cikluskérés közötti versenyhelyzetet - a frissítés prioritásával. Már folyó ciklust újabb kérés nem szakíthat meg. Frissítési ciklus alatt v. azzal egyidejűleg beérkező kérésre a memória foglaltságot jelez, ezért a memória a frissítés alatt nem hozzáférhető.



22. ábra

Memória rendszertechnikai felépítése
(dinamikus RAM)

Normál írás/olvasási ciklus a korábban bemutatott idődiagramoknak a 12 és a 15 ábra) megfelelően zajlik le. A vezérlő először a MPX-en keresztül valamennyi tokhoz eljuttatja a sorcímet, majd generálja a $\overline{RAS}$ vezérlőjelet, beállítja a $\overline{WE}$ jelet, átállítja a MPX-et és generálja a $\overline{CAS}$ vezérlőjelet, majd a kiolvasott adatot a $\overline{CAS}$ felfutó élével írhatjuk az adatregiszterbe (v. $\overline{CAS}=L$ értéken tartásával külön regiszter nélkül, de meghajtókon keresztül a sínre kapuzzuk az adatot). Hasonlóan zajlik a beírás is. Figyeljük meg az ábrán, hogy a $\overline{RAS}_i$ jellel csak a kívánt sort gerjesztjük, így a többi tok nyugalmi (stand-by) állapotban alig fogyaszt energiát (a $\overline{RAS}$ felel meg az $\overline{ME}$ -nek). A $\overline{CAS}$ és $\overline{WE}$ vezérlőjelet minden tokhoz elvezetjük ill. igen sok memória tok esetén ezek is dekódolhatóak. Frissítési ciklus (17. ábra) esetén az MPX a frissítési címszámláló tartalmát kapuzza az összes tokra és most valamennyi sorra egyszerre generál $\overline{RAS}$ jelet ($\overline{CAS}=H$). Így elérjük, hogy valamennyi tokban a kijelölt sor(ok) egyszerre frissítődik fel, ezért a teljes memória frissítése ugyanannyi cikluson át tart mint egyetlen morzsáé. (64 Kbites

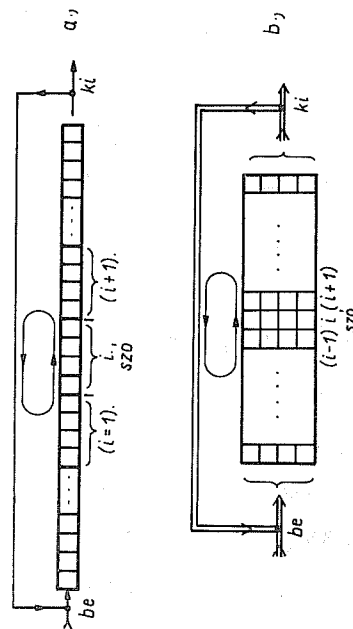
morzsáknál 256 sorcím 2 vagy 4 msec-onként; 256 Kbites morzsáknál a 11. ábra rendszertechnikája mellett ugyanezek az értékek adódnak. Ez 200 nsec olvasási/frissítési ciklusidő feltételezéssel 2 ms-onkénti frissítésnél 2,5% "veszteséget" jelent). A frissítési ciklusok generálása történhet elosztottan (előző számadataink mellett kb. 8 μs -onként egy frissítési ciklus) vagy összevontan (burst), amikor egymás után végezzük el az összes bit frissítését (2ms-onként 256 egymást követő frissítési ciklus ami kb. 51 μs -ig tart). A frissítés módja mindig a konkrét alkalmazástól függ. Különösen szerencsés eset az, amikor a frissítést úgy tudjuk hozzászinkronizálni a berendezéshez, hogy az azt egyáltalán nem veszi észre.

A 22. ábra Vezérlő része és a hozzá tartozó MPX-szel és dekóderrel egy morzsára integrálva kész áramkörként kapható különböző méretű memóriákhoz (dinamikus RAM vezérlő), ami nagymértékben egyszerűsíti a dinamikus RAM-ok használatát.

3. Soros hozzáférésű memóriák

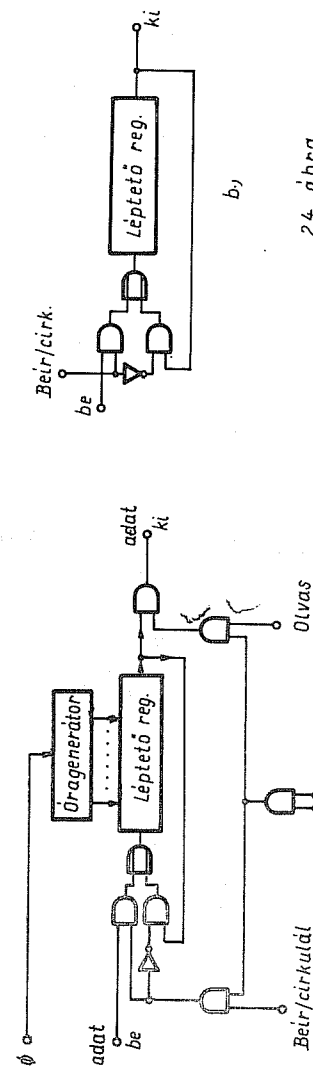
A soros hozzáférésű (v. kiváró) memóriák jellemzője, hogy az egyes információegységeket (pl. szavakat) csak sorban egymás után érhetjük el a

23.a és b. ábrán látható módon a kimeneten. Ezek a memóriák kétféleképpen szervezhetők: szó soros-bit soros (ilyen a 23a. ábra) vagy szó soros-bit párhuzamos (ilyen a 23b. ábra), a megvalósított memóriák egy része az utóbbi (pl. mágnesszalag), egy része pedig az előbbi szervezésű (mágneslemez); a félvezetős soros hozzáférésű memória pedig a szükségletnek megfelelően alakítható ki. Ezen memóriatípusok hozzáférési ideje felépítésükből adódóan nagy (lásd: a 23 ábrát), átlagosan megegyezik a be- és kimenet közti információ futási idő felével. A jelenlegi félvezetős soros hozzáférésű memóriák nagy hátránya, hogy az információ megőrzése energiát igényel, szemben a mágneses elven működő hasonló típusokkal, előnyük viszont a kisebb hozzáférési időben rejlik.



23. ábra

a) Soros hozzáférésű (szó-soros, bit-soros) memória
b) Soros hozzáférésű (szó-soros, bit-párhuzamos) memória



24. ábra

Soros hozzáférésű memóriamozsák szervezése

A bipoláris technológiával készülő soros hozzáférésű memóriák tulajdonképpen SR tárolókból felépített léptető regiszterek és ezek nagy helyigénye miatt általában 8-16 bit fér fel egy morzsára a szükséges lábszámot is figyelembe véve. Legtöbbször a tároláson túl más funkciót is ellátnak.

A MOS tranzisztorokból - hasonlóan a RAM-hoz - statikus és dinamikus léptető regiszter készíthető. A jelenlegi technológiai színvonal Kbit-Mbites morzsák készítését biztosítja.

3.1 Soros hozzáférésű memória morzsák kialakítása.

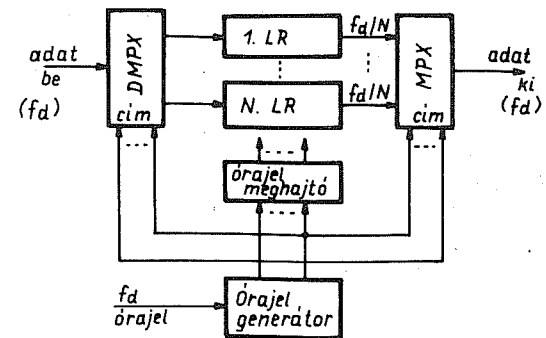
A léptető regiszteren túl célszerű a morzsán integrálva elkészíteni a kiszolgáló áramköröket, amelyek biztosítják az adatáramlást a bemeneten (új adat beírás vagy cirkuláltatás) és kimeneten, a lapka engedélyezését (bővíthetőség érdekében), valamint ha lehetséges az órajel előállító áramköröket is. Ennek megfelelően egy morzsa logikai vázlata a 24a. ábrán látható. Ha egy lapkán több, rövidebb léptető regisztert helyezünk el,

akkor a lábszám korlátozás miatt a szervezés a 24b. ábrán feltüntetettig egyszerűsödhet, sőt még a recirkulációs logika is elmaradhat. A felhasznált léptető regiszterek hossza változó, az alkalmazástól is függ (pl. nyomtatók átmeneti tárolójának hossza megegyezik az egy sorban levő karakterek számával, ez néhány szabványos érték), ezért sokszor van arra lehetőség, hogy a felhasználó válassza meg - adott határok között - a számára szükséges regiszter hosszúságot.

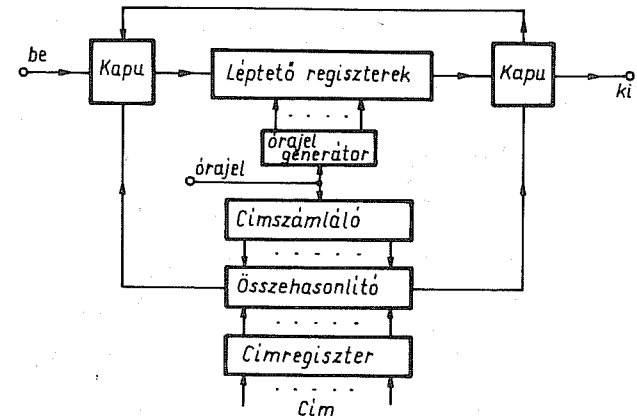
Az adatok a bemeneten ill. a kimeneten meghatározott órajelfázisban vannak jelen, tehát az adatokat az órajelhez képest időzíteni kell. A léptető regiszterek fontos jellemzője az, hogy mennyi adatot (hány bitet) lehet egy másodperc alatt beírni v. kiolvasni, ezt adatátviteli sebességnek nevezzük.

3.2 Soros hozzáférésű memóriák kialakításának szempontjai

Soros memóriák kialakításánál egyik legnagyobb probléma a megfelelő órajel meghajtó kiválasztása, ugyanis a kapacitív terhelésre dolgozó meghajtó áramkörnek nagy csúcsáramokat kell biztosítania a tranziensek idején ($I = C \Delta V / \Delta t$), esetleg több MHz frekvencia esetén is, ami nagyon megnöveli a zajokat. Ezért olyan megoldást kell keresnünk, ami az előző problémákat megoldja. Célszerű az órajel frekvenciát csökkenteni, ez ugyanis kisebb teljesítményt követel a meghajtóktól, de a tokok disszipációja is csökken, viszont a memória átviteli sebessége is ugyanígy romlik. Ezen az adatok multiplex kezelésével lehet segíteni. Ha f_d átviteli sebességre van szükségünk és N léptető regisztert párhuzamosan kapcsolunk, akkor egy léptető regisztert csak f_d/N frekvenciával kell igénybevennünk (25. ábra). Ugyanez a módszer akkor is nyilván jó, ha f_d nagyobb, mint az egyedi léptető regiszterek átviteli sebessége.



25. ábra
Soros adatok multiplex kezelése



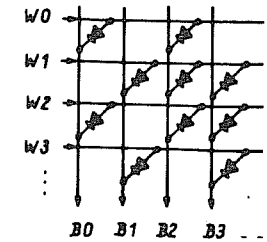
26. ábra
Soros memória címzése

Az adatok helyének nyilvántartására egy szám-
láló szolgálhat, így az adatok beírása, kiolvasása
az adott címről a 26. ábra szerint történik.

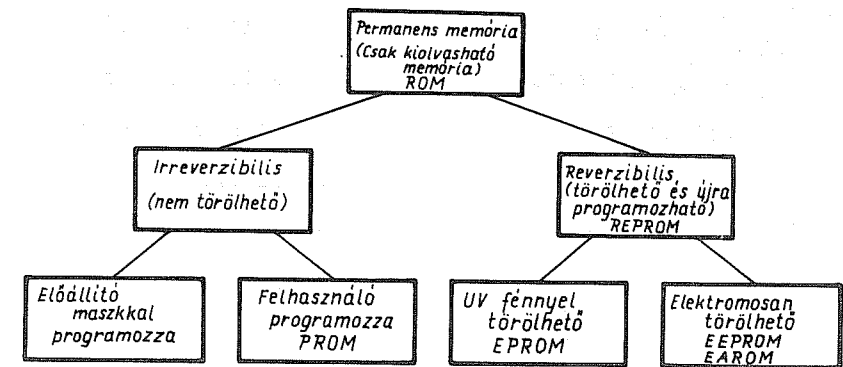
A kialakított soros memóriák kapacitása néhány
száz bittől egészen a több millió bit nagyságren-
dig változhat, az utóbbiakat "álló dob"nak is ne-
vezik, de a félvezetős eszközökből kialakított
"dob" hozzáférési ideje kb. egy nagyságrenddel ki-
sebb, mint a mágnesdoboké. Erre a célra jelenleg a
töltés-csatolt eszközök (CCD: Charge Coupled Devi-
ce) és a mágnesbuborék memóriák a legalkalmasab-
bak.

4. Permanens memóriák

A permanens (csak kiolvasható) memóriák sa-
játsága, hogy üzemszerűen csak egy előzőleg már
rögzített információ kiolvasására alkalmasak. Mű-
ködésüket a 27. ábra alapján érthetjük meg: vá-
lasszuk ki az i-ik szövezetékét olyan feszültség
szinttel, amely a hozzá tartozó diódákat kinyitja,
ugyanakkor az összes többi dióda zárva marad,
ezért azokon a bitvezetéseken, amelyekhez dióda
csatlakozik a kiválasztott szövezetékéről, megjele-



27. ábra
Permanens memória elve



28. ábra
Permanens memóriák felosztása

nik ez a feszültségszint a diódák nyitó irányú feszültségesésétől eltekintve, míg a többi bitvezeték feszültsége nem változik. A bitvezetésekre kapcsolódó olvasó erősítőkkel ez a feszültségváltozás érzékelhető. Mint látható, egy címhez tartozó információ rögzítése diódák elhelyezését jelenti azokban a rácspontokban, ahol 1-et akarunk tárolni, és 0-át kapunk a kimeneten, ahol a dióda hiányzik.

A rácspontokban elhelyezett csatolóelem nemcsak dióda lehet, hanem ellenállás, kapacitás, induktivitás vagy aktív (bipoláris ill. MOS tranzistor), de optikai csatolás is lehetséges. A kiszolgáló áramkörök a csatoló elemhez igazodva bipoláris ill. MOS eszközöket alkalmaznak. Mivel egy tároló cella most többnyire egy elemből áll, ezért nagyobb kapacitású mőzszak készíthetőek, mint az író-olvasó memóriáknál.

A permanens memóriák lehetnek reverzibilisek vagy irreverzibilisek attól függően (28. ábra), hogy a már egyszer beírt információ kitörölhető és új információ beírható-e.

Ha ez megtehető, akkor reverzibilis permanens memóriáról beszélünk. Az új információ beírását programozásnak nevezzük, ezért újra programozható csak kiolvasható memória (RePROM: Reprogrammable ROM) elnevezés is elterjedt. Ezeknél a memóriáknál a kitörlési és újra programozási idő lényegesen nagyobb, mint a kiolvasási idő, és gyakran bonyolult eszközöket is kíván. Nagy előnyük, hogy a programzást a felhasználó végzi, módosítás szükségése esetén a tartalom törölhető és az eszköz újra felhasználható. Két nagy csoportjukat különböztetjük meg: az egyiket ultraibolya sugárzással (UV fény) törölhetjük (UV EPROM: Erasable PROM), majd az üzemi feszültségnél (+5V) nagyobb feszültséggel (kb. +20V) programozhatjuk újra (beégetés); míg a másikat elektromosan törölhetjük (EEPROM: Electrically Erasable PROM) és ugyancsak nagyobb feszültséggel programozhatjuk újra. Ennek a megoldásnak előnye, hogy a törlő és programozó áramkörök is beépíthetők a nyomtatott áramkörti lapra és így egy tápfeszültség kimaradásakor nem felejtő memóriát kapunk, míg az UV EPROM-okat törléshez (kb. 15-20 perc) ki kell venni a helyükből és a törlőköszülékbe kell helyezni azokat.

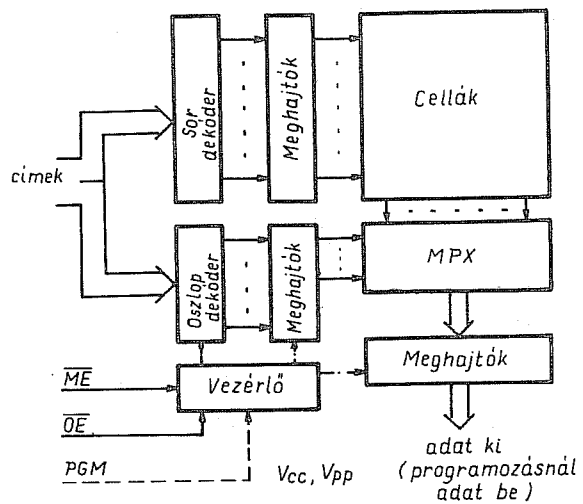
A permanens memóriák jelentős hányada irreverzibilis, azaz az egyszer beírt információ többet nem változtatható meg. A programozás történhet gyárilag, amikor a felhasználó igényének megfelelő információt a gyártás közben maszk segítségével rögzítik úgy, hogy az adott rácspontokban a csatlóelemet "bekötik" a szó és bitvezeték közé, míg a többi rácspontban ez nem történik meg. Ennek a megoldásnak hátránya, hogy a maszk elkészítése drága és időigényes (gyártásba vitellel együtt 4-6 hét), ezért csak nagy sorozatszám esetén kifizetődő. Vannak olyan típusú megoldások, amikor a programozást a felhasználó végzi (PROM : Programmable ROM). Ilyenkor gyártás során minden rácspontban létrehozzák a csatlóelemet és a felhasználó utólag köti be vagy szünteti meg a bekötést az alkalmazott megoldástól függően. Fémes érintkezés kialakítása ill. megszüntetése történik az üzemszerűtől eltérő (lényegesen nagyobb) feszültség és áramértékkel a kimenet felől, és e programozó impulzusok időzítése nagy pontosságot igényel.

4.1 Permanens memória morzsák kialakítása

A gyártók azonos lábkiosztású, kapacitású, hozzáférési idejű ROM-EPROM (EEPROM) családokat fejlesztenek, amelyekhez illeszkedő statikus RAM-ok is készülnek. Így a fejlesztés - kissorozatú - nagysorozatú gyártás során ezeket rugalmasan lehet egymással kicserélni.

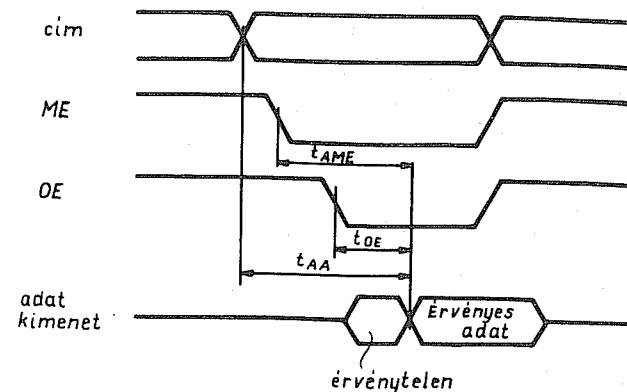
Ezeknek a családoknak a rendszerteknikai felépítése megegyezik, de a tárolócellák kialakításában alapvetően eltérnek egymástól. Ezek a morzsák is a módosított bitszervezést követik, amint az a 29 ábrán látható. Kimeneteik nyitott kollektorosak vagy három állapotúak a bővíthetőség érdekében és az utóbbiak egy része külön kimenet engedélyező (Output Enable: $\overline{OE}$) vezérlőjelet is használ. Kiolvasásnál vezérlésük megegyezik a statikus RAM-ok kiolvasásával (30. ábra). Ugyancsak két hozzáférési időt definiálhatunk: cím hozzáférési időt (t_{AA}); ME hozzáférési időt (t_{AME})

A kimeneti három állapotú meghajtók engedélyezési ideje (t_{OE}).



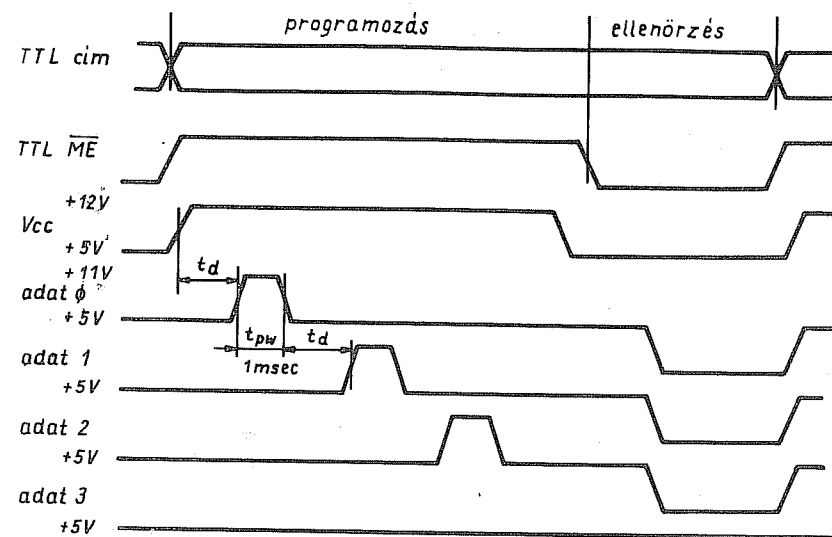
29. ábra

Permanens memória rendszerteknikai felépítése



30. ábra

Permanens memória olvasása



31. ábra

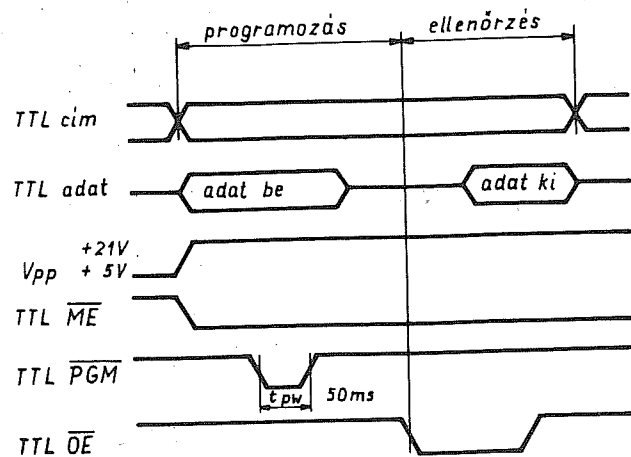
PROM programozási ciklus

A reverzibilis permanens memóriák programozási, beírási vezérlése, az alkalmazott feszültségek ill. áramok valamint ezen ciklusok időzítése egyrészt az alkalmazott tárolócellától (PROM, EPROM, EEPROM), és ezen belül is gyártó által alkalmazott technológiától függ, ezért konkrét típusok esetében csak a katalógusok gondos tanulmányozása segíthet. Célszerű azonban néhány jellegzetes programozási ciklust megvizsgálni. PROM-ok programozásának néhány jellemzője olvasható le a 31. ábráról. A programozandó szó címét (TTL szinteken) beállítva, majd a morzsát letiltva ($\overline{ME}=H$) és a tápfeszültséget megemelve (+12 V-ra) a szó egyes bitjeit külön-külön programozzuk t_p időnként. Arra az adat kimenetre, amelyet programozni akarunk kívülről t_{pw} széles (msec nagyságrendű) a tápfeszültségnél nagyobb feszültségű (itt 5V $\rightarrow$ 11V) impulzust kényszerítünk, melynek hatására az eredetileg H szintű (szinte valamennyi reverzibilis permanens memóriát úgy szállítanak, hogy összes bitje magas szintű; tehát a programozás alacsony szint beírását jelenti) kimenet alacsony szintűvé válik. Ahol megfelel a H szintű kimenet, ott nem kell semmit

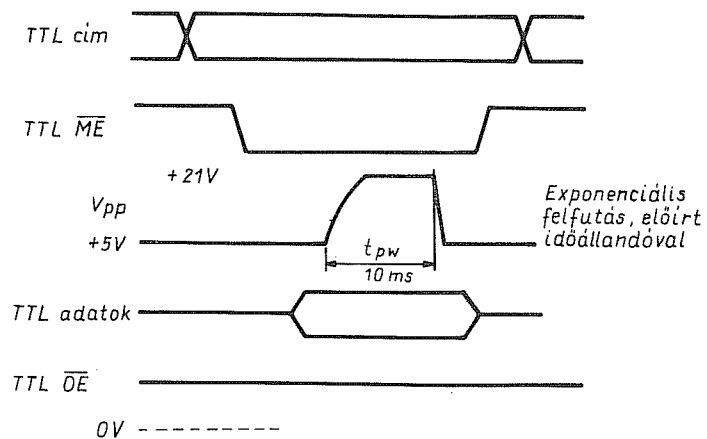
csinálni (adat 3). A programozást ellenőrzés követi, amikor kiolvassuk az előbb programozott szót és meggyőződünk róla, hogy tényleg az előírt értéket tartalmazza-e. Egy cím beégetését általában többször is el kell végezni, többnyire egyre növekvő t_{pw} -vel.

Az EPROM-ok programozásában (32. ábra) a legfontosabb eltérés az, hogy egy szó valamennyi bitjét egyszerre beégethetjük, valamint t_{pw} egy nagyságrenddel nagyobb (~ 50 msec).

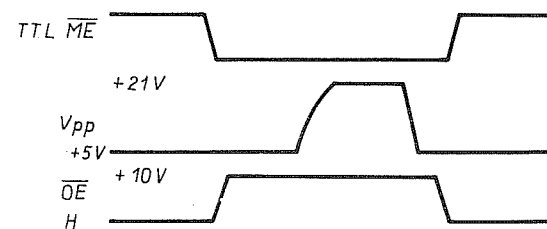
Az EEPROM-ok esetében is byte-onként beírhatunk (FF beírása a byte törlését jelenti), amit a 33. ábra mutat. Programozó impulzus itt t_{pw} (~ 10 msec) szélességű és előírt időállandójú exponenciális felfutású, amit külön áramkörrel kell előállítani. Hasonlóan kell vezérelni az egész morzsa törlését is (valamennyi byte=FF), azzal a különbséggel, hogy az $\overline{OE}$ jelet TTL magas szintről +10 V feszültség szintre kell emelni (34. ábra). EEPROM-ok üzemi törléséhez, a beíráshoz több tápfeszültségre és speciális meghajtó áramkörökre van szükség, amelyeket célszerű egyetlen morzsára ráintegrálni.



32. ábra
Tipikus EPROM programozási ciklus



33. ábra
Byte beírása (törlése) EEPROM-ba



34. ábra
EEPROM morzsa törlése

Nagyobb kapacitású permanens memóriatömbök kialakítása gyakorlatilag megegyezik a statikus RAM-ok bővítésével.