Eszterházy Károly Tanárképző Főiskola

Garamhegyi Gábor

A mikroszámítógépek rendszertechnikai alapjai

Kézirat

Eszterházy Károly Tanárképző Főiskola

Garamhegyi Gábor

A mikroszámítógépek rendszertechnikai alapjai

Kézirat

Eger, 1992

Lektorálta

Dr. Szabó József

tanszékvezető egyetemi adjunktus a
matematika tudományok kandidátusa

A kiadásért felelős:
az Eszterházy Károly Tanárképző Főiskola főigazgatója
Megjelent a Főiskola műszaki gondozásában
Felelős: Dr. Csőke Lajos jegyzetmenedzser
Műszaki szerkesztő: Garamhegyi Gábor

Készült az Eszterházy Károly Tanárképző Főiskola Nyomdájában
Felelős vezető: Budavári Sándor

1. Matematikai logika

A digitális készülékek első pillanatra igen bonyolultnak tűnnek, pedig néhány logikai kapu többszöri felhasználásával felépíthetők. Az alapkapcsolásokból a szükséges rendszert formális módszerekkel alakíthatjuk ki, amelyhez igen nagy segítséget nyújt a logikai algebra vagy más néven Boole - algebra. Ismerkedjünk meg ennek alapjaival, illetve azokkal a módszerekkel, amelyek nélkülözhetetlenek a számítógépek belső felépítésének megértéséhez.

1.1. Logikai algebra

Minden olyan állítást, ami vagy igaz vagy hamis, **logikai kijelentésnek** nevezünk. (Egy kijelentés logikai értéke a két lehetséges közül szigorúan csak az egyik lehet,ezek egymást kölcsönösen kizárják.) Az igaz vagy hamis tulajdonságot **logikai értéknek** szokás nevezni. A továbbiakban jelöljük 1 -el a kijelentés igaz, 0 -val a kijelentés hamis értékét. Ha az E = { 0, 1 } halmazon műveleteket definiálunk, akkor algebrai struktúrát kapunk. (A definiált művelet nem vezet ki a halmazból.) Ez a logikai kijelentések által felvehető értékek algebrája vagy más néven logikai algebra, vagy bináris algebra.

Georg Boole (1815 - 1864) 1854 -ben kiadott Bevezetés a gondolkodás törvényeibe (Introduction to the Lows os Thought) című művében fektette le a bináris algebra alapjait. A logikai változók között három alapvető [1]logikai műveletet definiált:

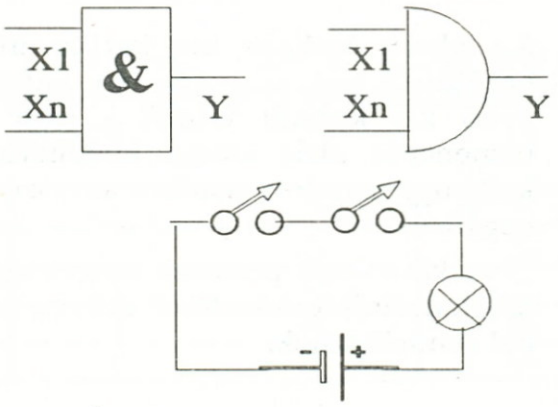
a **konjunkciót** - a logikai SZORZÁS vagy ÉS művelet -

A 1.1.táblázatból látható, hogy a művelet eredménye akkor és csakis akkor igaz, ha a komponensek logikai értéke is igaz.

X1	x2	y
0	0	0
0	1	0
1	0	0
1	1	1

1.1. táblázat

/ A művelet akkor egyértelműen meghatározott,ha a komponensek minden lehetséges értékéhez megadjuk a művelet eredményét. Mivel a most definiált művelet két változós, ezért összesen 4 lehetséges értékkombinációt adhatunk meg. - Minden logikai változó két értéket vehet fel ezért a lehetséges értékkombinációk száma 2^2. Általánosan N változó esetén a lehetséges kombinációk száma 2^N . / A következő ábrán (1.1 ábra) az ÉS kapcsolat szimbolikus jelölését és elemi áramköri megvalósítását láthatjuk.



1.1. ábra

Az És-kapcsolat rajzlejei és elemi áramköri megvalósítása.

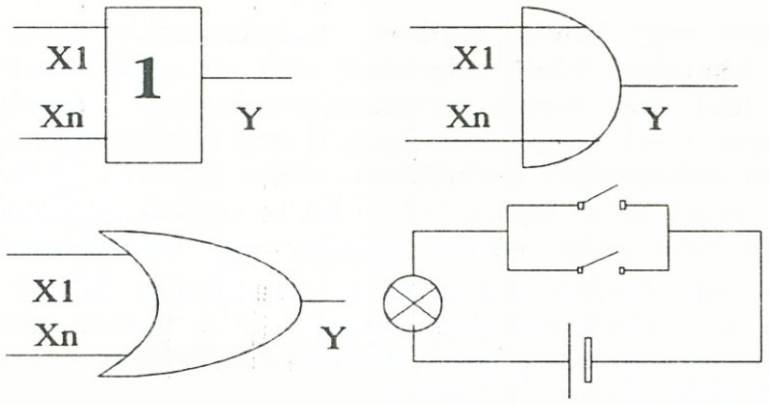
[1] Logikai műveletről akkor beszélünk, ha a művelet eredménye csak a komponenseinek logikai értékétől függ.

A **diszjunkció** vagy logikai ÖSSZEADÁS vagy VAGY művelet

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	1

1.2. táblázat

Mint ahogyan a 1.2. táblázatból látható a művelet eredménye akkor igaz,ha valamelyik bemeneti változó értéke igaz.
Jelölését és egyszerű megvalósítását a 1.2. ábrán láthatjuk.



1.2. ábra

AVagy-kapcsolat rajzjelei és elemi áramköri megvalósítása.

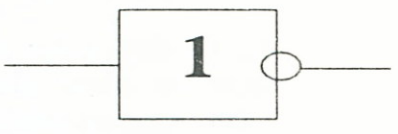
Az utolsó elemi művelet a **negáció ,** a tagadás művelete

Mint látható ez a logikai művelet nem két változós mint az eddigiek voltak,hanem csak egy.

X1	Y
0	1
1	0

1.3. táblázat

A művelet eredménye a logikai érték ellentetje,ha a bemeneti változó hamis volt, akkor a művelet éredménye igaz és fordítva.
Az áramkör jelölését a 1.3. ábrán láthatjuk. Az áramkör megvalósítását azonban nem, mert egyszerű kapcsolók segítségével a feladatot nem lehet megoldani. A későbbiek során majd néhány példát látunk arra, hogy ennek a logikai műveletnek milyen elektromos megfelelői vannak.



1.3. ábra

Negációrajzjele.

Az ábrán látható kis karika utal arra,hogy negálásról van szó. (Ha egy kapcsolási rajzon ilyen kis karikát látunk - akár az áramkör bemenetén akár annak kimenetén - tudnunk kell,hogy az adott ponthoz tartozó logikai érték negált.)

A következő pontban megvizsgáljuk,hogy az előbb definiált műveletek milyen tulajdonságokkal rendelkeznek.

1.2. A bináris algebra tulajdonságai

A logikai műveletek tulajdonságai sokban hasonlítanak az elemi matematikai műveletek tulajdonságaihoz. Igy például megtaláljuk a kommutativitás, asszociativitás, disztributivitás tulajdonságait. Ezeken kívül azonban sok olyan tulajdonságot találunk, amelyek nem

szokványosak. Vegyük tehát sorra a logikai algebra műveleteit.

Érdemes észrevenni, hogy a szokásos algebrával ellentétben, itt nem fordul elő a 2x illetve x^2 kifejezés, ami a tautológia azonosság következménye.

Felcserélve az egyenletpárokat (1.4. táblázat) látható a **dualitás szabálya.** Ugyanis ,ha valamennyi azonosságban felcseréljük a diszjunkciókat a konjunkciókkal és egyidejűleg a 0 -át az 1 -el,akkor szintén azonosságot kapunk.

1.Kommutativitás	x1 * x2 = x2 * x1 x1 + x2 = x2 + x1
2.Asszociativitás	x1 * (x2 * x3) = (x1 * x2) * x3 x1 + (x2 + x3) = (x1 + x2) + x3
3.Disztributivitás	x1 * (x2 + x3) = (x1 * x2) + (x1 * x3) x1 + (x2 * x3) = (x1 + x2) * (x1 + x3)
4.Abszorbció	x1 * (x1 + x2) = x1 x1 + (x1 * x2) = x1
5.Tautológia	x1 * x1 = x1 x1 + x1 = x1
6.Negáció	x1 * $\overline{x1}$ = 0 x1 + $\overline{x1}$ = 1
7.kettős negáció	$\overline{\overline{x1}} = x1$
8.De Morgan szabály	$\overline{x1 * x2} = \overline{x1} + \overline{x2}$ $\overline{x1 + x2} = \overline{x1} * \overline{x2}$
9.műveletek 1 -el és 0 - val	x1 * 1 = x1 x1 * 0 = 0 x1 + 1 = 1 x1 + 0 = x1 $\overline{1}$ = 0 $\overline{0}$ = 1

1.4. táblázat

Ezeket a szabályokat egyszerűen igazolhatjuk ,ha elkészítjük az egyenletek bal és jobb oldalán szereplő kifejezések értéktáblázatait és összehasonlítjuk őket.Nézzük meg például a **De Morgan azonosságok** közül az elsőt.

Látható, (1.5.táblázat) hogy a táblázat negyedik és hetedik oszlopában kapott értékek a bemeneti változók bármely értékkombinációja mellett megegyeznek. Ez azt jelenti, hogy az azonosság teljesül.

x1	x2	x1 *x2	$\overline{x1 * x2}$	$\overline{x1}$	$\overline{x2}$	$\overline{x1} + \overline{x2}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

1.5. táblázat

1.3. Logikai függvények

A logikai algebrában függvényeket is definiálhatunk. Vezessük be ehhez, először a logikai változó fogalmát. Ha egy x változó 1 és 0 értéket vehet fel, akkor azt mondjuk, hogy X bináris változó, vagy — **logikai változó** —. Ha Y logikai változó értéke az X logikai változó értékétől függ, akkor X és Y megfelelő értékpárjai egy függvényt definiálnak.

$F : \{ 0,1 \} \longrightarrow \{ 0,1 \}$ vagy $X \longmapsto Y=F(X)$

Természetesen a kialakított fogalmat általánosíthatjuk is : ha Y logikai változó értéke az X1, X2, . . . Xn logikai változók értékeitől függenek, akkor **n változós logikai függvény** ről beszélünk.

Ezen függvényeket megadhatjuk :

- a.) **táblázattal** / Ezzel a hozzárendelési módszerrel egyértelműen leírhatjuk a függvényt, hiszen az értelmezési tartományt is és értékkészletet is a {0 , 1 } kételemű véges halmaz alkotja/,
- b.) **képlettel,**
- c.) **hozzárendelési utasítással.**

A továbbiakban vizsgáljuk meg, hogy hány darab n változós függvény létezik.

Változók száma	A létező függvények száma
0	2
1	4
2	16
3	256
4	65536

1.6. táblázat

Ezt a feladatot gyorsan meg tudjuk oldani,hiszen tudjuk, hogy n logikai változónak összesen 2^n lehetséges értékkombinációja van és minden ilyen értékkombinációhoz két különféle függvényérték tartozik.

Ez összesen $_2 2^n$.

Mint ahogyan a 1.6. táblázatból látható, a lehetséges logikai függvények száma rohamosan nő.

A kétváltozós függvényeket érdemes kissé közelebről megvizsgálni, és a fontosabbakat elnevezni hiszen ezeket elég gyakran fogjuk használni. (1.7. táblázat)

Sorszám	Logikai érték	elnevezés
	x1 1 1 0 0	
	x2 1 0 1 0	
0	y 0 0 0 0	soha
1	y 0 0 0 1	nem - vagy NOR
2	y 0 0 1 0	x1 gátolja x2 -t
3	y 0 0 1 1	negált x1
4	y 0 1 0 0	x2 gátolja x1 -t
5	y 0 1 0 1	negált x2
6	y 0 1 1 0	kizárólagvagy XOR
7	y 0 1 1 1	nem és NAND

1.7/a. táblázat

A táblázat folytatása.

Sorszám	Logikai érték	Elnevezés
	x1 1 1 0 0	
	x2 1 0 1 0	
8	y 1 0 0 0	és AND
9	y 1 0 0 1	ekvivalencia
10	y 1 0 1 0	x2 ismétlő
11	y 1 0 1 1	ha x1 akkor x2
12	y 1 1 0 0	x1 ismétlő
13	y 1 1 0 1	ha x2 akkor x1
14	y 1 1 1 0	diszjunkció OR
15	y 1 1 1 1	mindig

1.7/b. táblázat

A digitális technikában a feladatokat legtöbbször függvénytáblázattal adjuk meg. A kapcsolások kialakításához azonban szükségünk van a logikai függvények lehető legegyszerűbb alakjára, hiszen ez mint látni fogjuk azt jelenti, hogy ekkor kell a legkevesebb alkatrészt felhasználni.
Az áramkörök kialakításánál tehát a feladataink a következők:

- felírni a logikai kapcsolatokat táblázat formájában,
- a táblázatból megalkotni a logikai függvényeket,
- a függvényeket valamilyen módszerrel a legegyszerűbb alakra hozni,
- majd létrehozni a kívánt kapcsolást.

A függvények táblázatos megadása egyszerű feladat, hiszen a bemeneti változók és a kimeneti értékek ismeretében a hozzárendelés megtehető.

A következőkben a második feladattal fogunk foglalkozni, azaz azt vizsgáljuk, hogyan lehet megadni a függvény algebrai alakját a táblázat ismeretében.

1.4. A logikai függvények létrehozása

A függvény analitikus (képlettel megadott) alakját akkor kapjuk meg, ha a független logikai változókat, véges számú logikai művelettel kapcsolunk össze (konjunkció, diszjunkció, negáció).

1.4.1. A diszjunktív normál alak

Probáljuk meg felírni a következő, 1.8. táblázattal megadott függvény analitikus alakját.Vizsgáljuk meg azokat az eseteket,amikor a függvény sorai igaz értékeket vesznek fel. (Ezeket a táblában * jelöli.) Az előállítandó függvény nyilván ott igaz, ahol ezen sorok valamelyike igaz.Ez pedig a sorok diszjunktív kapcsolatára utal. A függvényérték — az adott bemeneti változók mellett— akkor és csakis akkor lesz igaz, ha az ezekből képezett konjunkció igaz. Ez úgy állítható elő, hogy vesszük azokat a bemeneti változókat, amelyek igaz értéket vesznek fel és azok negáltját amelyek hamisat, majd képezzük ezek konjunkcióját. (A konjunkció akkor és csakis akkor igaz, ha minden tagja igaz. ez pedig az adott esetre nézve csak a fenti módon teljesül.)

x1	x2	y
0	0	1 *
0	1	0
1	0	1 *
1	1	0

1.8. táblázat

Összefoglava **diszjunktív normál alakról** akkor beszélünk, ha

» minden konjunkciós tényező vagy egy változó, vagy egy változó negáltja, és minden diszjunkciós tagban minden változó szerepel (vagy negálva vagy negálatlanul)

» minden változó egy konjunkcióban pontosan egyszer szerepel,

» nincs két olyan diszjunkciós tag amelyben csak a változók sorrendje különbözik.

Az előzőek alapján a függvényünket a következő alakban írahtjuk fel.Az ilyen függvény előállítást akkor célszerű alkalmazni, ha a táblázatban az 1 -es függvényértékek száma kevés.

$$F(x1, x2) = (\overline{x1} * \overline{x2}) + (x1 * \overline{x2}) \qquad 1.1.$$

1.4.2. A konjunktív normál alak

Az előzőek alapján könnyen megfogalmazhatjuk a másik módszert is a függvények előállítására. Míg az előbb a táblázatból azokat az eseteket emeltük ki, amelyekben a függvényértékek igazak voltak, most a hamis (0) értékekkel tesszük ugyanazt. A bemeneti változók adott értékei mellett a függvényérték akkor és csakis akkor hamis,ha valamelyik tényezője hamis. Ez a konjunkciós kifejezésre utal. A tényezők akkor lesznek hamisak,ha minden tagjuk hamis.Ez pedig úgy állítható elő, hogy vesszük azokat a változókat, amelyek hamisak és azok negáltját, amelyek igazak.

Összefoglalva: **konjunktív normál alakról** akkor beszélünk, ha

» minden diszjunkciós tag vagy egy változó, illetve egy változó negáltja, és minden konjunkciós tényezőben minden változó szerepel,

» minden változó a diszjunkciós tagban pontosan egyszer szerepel,

» nincs két olyan konjunkciós tényező amelyben csak a változók sorrendje különbözne.

$$F(x1, x2) = (x1 + \overline{x2}) * (\overline{x1} + \overline{x2}) \qquad 1.2.$$

Természetesen megmutatható, hogy a két függvény ekvivalens, azonos alakra hozható. Az 1.2. egyenletre alkalmazzuk a De Morgan azonosságot. Egyszerű átrendezéssel látható, hogy az 1.1. illetve az 1.2. egyenletek azonos végeredményre vezetnek

$$F(x1, x2) = (x1 + \overline{x2}) * (\overline{x1} + \overline{x2}) =$$
$$= \overline{\overline{(x1 + \overline{x2}) * (\overline{x1} + \overline{x2})}} =$$
$$= \overline{\overline{(x1 + \overline{x2})} + \overline{(\overline{x1} + \overline{x2})}} =$$
$$= \overline{(\overline{x1} * x2) + (x1 * x2)} =$$
$$= \overline{x2 * (\overline{x1} + x1)} = \overline{x2}$$

$$F(x1, x2) = (\overline{x1} * \overline{x2}) + (x1 * \overline{x2}) =$$
$$= \overline{x2} * (\overline{x1} + x1) = \overline{x2}$$

1.4.3. Részben határozott függvények

Sokszor előfordul olyan eset, hogy az igazságtáblázatban nincsen megadva az összes bementeti változóhoz tartozó függvényérték. Ilyenkor beszélünk részben határozott függvényekről. Ebben az esetben szabadon választhatjuk meg az adott sorban a függvény értékét. Természetesen ezt úgy célszerű megtenni, hogy minél könnyebben egyszerűsithető legyen a függvény.

x1	x2	x3	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	*
1	1	0	1
1	1	1	1

1.9. táblázat

Az 1.9. táblázat * -gal jelölt sorában a függvényérték tetszőlegesen választható, akár 1, akár 0 írható a sorba.

$x1*\overline{x2*x3} + x1*x2*\overline{x3} + x1*x2*x3 = x1*\overline{x3} + x1*x2$ adódik 0 választása mellett,

$x1*\overline{x2*x3} + x1*\overline{x2}*x3 + x1*x2*\overline{x3} + x1*x2*x3 = x1$ adódik az 1 választása során.

1.5. Függvényekegyszerűsítése

A logikai függvények egyszerűsítésére számtalan módszert dolgoztak ki: algebrai, grafikus, numerikus eljárásokat. A feladat jellege alapján dönthetjük el, hogy melyik eljárás használata az előnyös.

Az algebrai módszer alkalmazása (az azonosságok felhasználásával) a legtöbb esetben bonyolult és összetett.

Ha viszonylag kevés változónk van, akkor áttekinthető megoldást szolgáltat az úgynevezett **Karnaugh - tábla.** Ez a megoldás a grafikus eljárások családjába tartozik. A továbbiakban ezzel fogunk megismerkedni. Ez a táblázat a független változók mátrix alakban történő elhelyezésén alapul, a függvénytáblázat átrendezésének is tekinthető.Hogy érthetőbb legyen, lássunk egy példát. Mivel mindegyik változó igaz, vagy hamis értéket vehet fel, ezért négy lehetséges kombináció van. Ennek megfelelően a táblázat a következőképpen alakul.

X2 \ X1	$\overline{X1}$	X1
$\overline{X2}$	$\overline{X1}\overline{X2}$	$X1\overline{X2}$
X2	$\overline{X1}X2$	X1X2

Vagy

X2 \ X1	0	1
$\overline{0}$	$\overline{X1}\overline{X2}$	$X1\overline{X2}$
1	$\overline{X1}X2$	X1X2

A két változó lehetséges értékkombinációit a táblázat belső mezői tartalmazzák.Írjuk most fel az $y = \overline{x_1}\,\overline{x_2} + x_1\,x_2$ kétváltozós függvény Karnaugh-tábláját. A fenti

táblázatnak megfelelően az $\overline{x_1}\ \overline{x_2}$ és $\overline{x_1}\ x_2$ négyzetekbe 1-et írunk, a többi mezőbe pedig 0-t. Ezért azt is mondhatjuk, hogy minden négyzetben a változókat AND, a négyzeteket pedig OR művelet kapcsolja össze. A tábla tehát kitűnően alkalmazható a diszjunktív normál alakban megadott függvények ábrázolására.

Több változó használata esetén, ha a számuk páros, négyzet alakú táblázatot kapunk. Ellenkező esetben pedig olyan téglalaphoz jutunk, amelynek az egyik oldala dupla olyan hosszú, mint a másik. Például a háromváltozós Karnaugh-táblázat így alakul:

X2,X3 / X1	00	01	11	10
0				
1				

A táblázat felírásánál látható,hogy a változók értékkombinációi olyan sorrendben követik egymást,hogy a felírt csoportok csak egyetlen számjegyben térnek el egymástól. (Az utolsó illetve utolsó előtti oszlop sorrendje megváltozik a binárisan növekvő sorrendhez képest.)

Az ilyen elrendezés könnyen egyszerűsíthetővé teszi a táblázatban felírt függvényt. Az egyszerűsítés azért tehető meg könnyen, mert ha két szomszédos négyzetben 1 szerepel, akkor az azt jelenti, hogy a két négyzethez tartozó változók szorzata csak az egyik változó értékében különbözik.

X2,X3 / X1	00	01	11	10
0	0	1	1	0
1	0	0	0	0

$$y = \overline{x_1}\ \overline{x_2}\ x_3 + \overline{x_1}\ x_2\ x_3 = \overline{x_1}\ x_3$$

mivel tudjuk,hogy $\overline{x_2} + x_2 = 1$.

Látható, hogy az egyszerűsítés után csak a közös szorzótényezők maradnak meg. A közös szorzótényezőket a táblázatból könnyen leolvashatjuk.

Természetesen több négyzet is összevonható. Az alábbi táblában adott esetben például az egyszerűsített függvény $y = x_2$ alakban írható fel.

X2,X3 / X1	00	01	11	10
0	1	1	0	0
1	1	1	0	0

Általánosan az igaz, hogy ha n változót akarunk kiküszöbölni, akkor 2^n négyzetből álló csoportot kell készítenünk (ha ez lehetséges).

A táblázat szélein lévő négyzetek is összevonhatók, ugyanis a változók fent említett elrendezése mellett, ha csak egy változó értékét módosítjuk, a táblázat szélein lévő oszlopban illetve sorban, akkor a tábla másik végén található oszlop illetve sor változóihoz jutunk. (10 módosításával 00-hoz jutunk.) A táblázat tehát úgy is felfogható, mintha a széleit henger formájában egymáshoz ragasztottuk volna.

X2,X3 / X1	00	01	11	10
0	1	0	0	1
1	0	0	0	0

$$y = \overline{x_1}\ \overline{x_2}\ \overline{x_3} + \overline{x_1}\ x_2\ \overline{x_3} =$$
$$= \overline{x_1}\ \overline{x_3}\ (\overline{x_2} + x_2) = \overline{x_1}\ \overline{x_3}$$

Mint látható, az $\overline{x_1}\ \overline{x_3}$ kifejezés mind a két négyzetben megtalálható. A négyváltozós esetben az egyszerűsítés algebrai módon már elég hosszadalmas. Legyen például az

$$y = \overline{x_1}\ \overline{x_2}\ \overline{x_3}\ \overline{x_4} + \overline{x_1}\ \overline{x_2}\ x_3\ \overline{x_4} + x_1\ \overline{x_2}\ \overline{x_3}\ \overline{x_4} + x_1\ \overline{x_2}\ x_3\ \overline{x_4} =$$

$$= \overline{x_1}\ \overline{x_2}\ \overline{x_4}(\overline{x_3} + x_3) + x_1\ \overline{x_2}\ \overline{x_4}(\overline{x_3} + x_3) = \overline{x_1}\ \overline{x_2}\ \overline{x_4} + x_1\ \overline{x_2}\ \overline{x_4} =$$

$= \overline{x_2}\ \overline{x_4}(\overline{x_1} + x_1) = \overline{x_2}\ \overline{x_4}$.

Nézzük meg ezek után, hogy a Karnaugh-tábla segítségével milyen eredményre jutunk. Rajzoljuk fel a négyváltozós táblát. A közös tényezők a táblázatból könnyen kiemelhetők.

X3,X4 / X1,X2	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

Nem csak különálló csoportokat alakíthatunk ki, hanem átfedett, vagy részben átfedett csoportok is képezhetők.

Az egyszerűsített függvény létrehozásának lépései ebben az esetben a következők: képezzük a létrehozott csoportok minimálizált alakjait és ezeket az OR művelettel kapcsoljuk össze. Befejezésül tekintsünk meg egy ilyen esetet:

X3,X4 / X1,X2	00	01	11	10
00	1	1	0	1
01	1	1	1	1
11	1	1	0	0
10	1	1	1	0

1.csop. 2.csop. 3.csop.

az első csoport $\overline{x_3}$,

a második csoport $x_1\ \overline{x_2}\ x_4$,

a harmadik csoport $\overline{x_1}\ x_2$.

A függvény minimalizált alakja a következő:

$y = \overline{x_3} + x_1\ \overline{x_2}\ x_4 + \overline{x_1}\ x_2$.

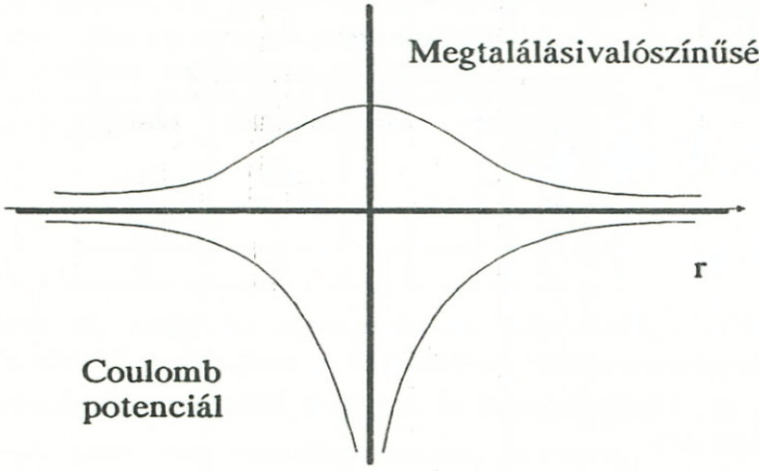
2. A félvezetők fizikai tulajdonságai

2.1. A kovalens kötés

A középiskolai fizikai tanulmányaink során megtanultuk, hogy az anyag a minimális energiájú állapot elérésére törekszik. Ennek alapján jönnek létre az anyag bonyolultabb struktúrái a molekulák, a kristályos anyagok stb.

Idézzük fel ezeket az ismereteket, hogy a későbbiek során megértsük az áramköri elemek — dióda, tranzistor — működését.

Ha például egy hidrogén (H) atomot vizsgálunk azt látjuk, hogy a proton körül keringő elektront a Coulomb vonzás tartja fogva. A klasszikus fizikával ellentétben az elektronnak a helyét és sebességét nem tudjuk pontosan megadni, csak a tartózkodási valószínűségét ismerjük. Ennek menetét ábrázolhatjuk egy koordináta rendszerben. (2.1. ábra) A görbe ott tér el leginkább a vízszintestől, ahol az elektront a legnagyobb valószínűséggel találjuk meg. Látható az is, hogy ekkor a Coulomb potenciál is a legnagyobb.



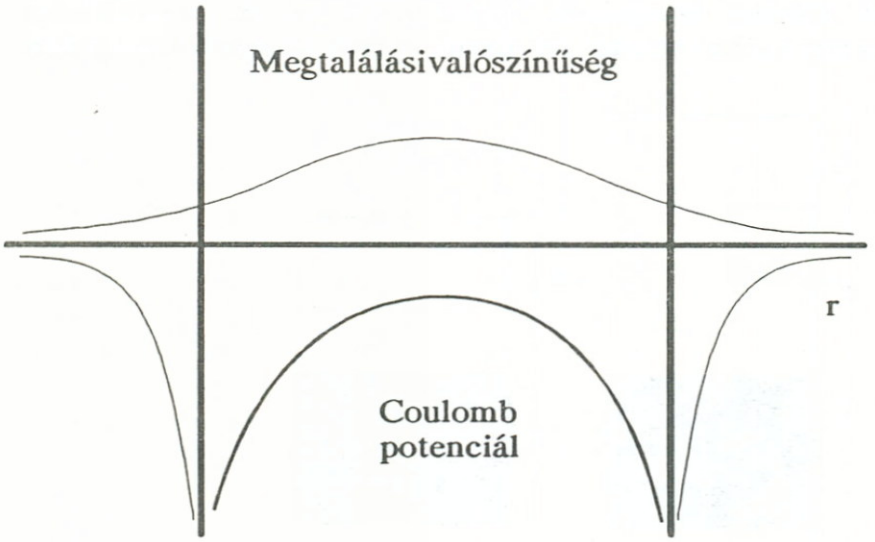
2.1. ábra

A abszcissza tengely fölött az elektron atommag körüli megtalálási valószínűségi eloszlását, alatta a mag által létrehozott potenciált láthatjuk.

Ezt úgy is megfogalmazhatjuk, hogy a Coulomb vonzás szorítását az elektron nyüzsgése ott egyenlíti ki.

Ha most két H magot közelítünk egymáshoz és egy elektront adunk ebbe a rendszerbe, azt tapasztaljuk, hogy az elektron ekkor a két mag között fog legtöbbször elhelyezkedni, azaz itt lesz a legnagyobb az elektron megtalálási valószínűsége. Az elektron számára a két proton (H mag) potenciáltere áll rendelkezésre, mintegy kisebb nyüzsgést, nagyobb szabadságot biztosítva az elektron számára. Így az elektron mozgási energiája (nyüzsgése) csökken, ami az összenergia csökkenéséhez is vezet. A rendszer tehát alacsonyabb energiájú állapotba került, teljesítve ezzel az energiaminimumra törekvés elvét.Ha újabb elektront adunk a rendszerbe, akkor kialakíthatjuk a H_2 molekulát. Ez az elektron a [1]**Pauli elv** szerint ugyannabba az elektronhéjba kerülhet. Mivel a két elektron a két proton terében együttesen mozoghat, úgy tekintjük, mintha kölcsönösen használnák a protonok az elektronokat. De úgy is megfogalmazhatjuk, hogy az elektronok kölcsönösen kicserélődnek az atomok terében. A fizikusok ezt a tényt úgy fogalmazzák meg, hogy a protonokat a kicserélődési erők tartják össze. Ekkor **kovalens kötésről** beszélünk.

1 Egy atomon belül két vagy több elektron nem kerülhet ugyanabba az elektronállapotba. Egy kvantumszám eltérés már más állapotot jelent. Ha az elektron perdületétől eltekintünk — két ellentétes „forgásirány" —, akkor a Pauli elvet átfogalmazhatjuk: egy elektronpályán maximum két elektron tartózkodhat.



2.2. ábra

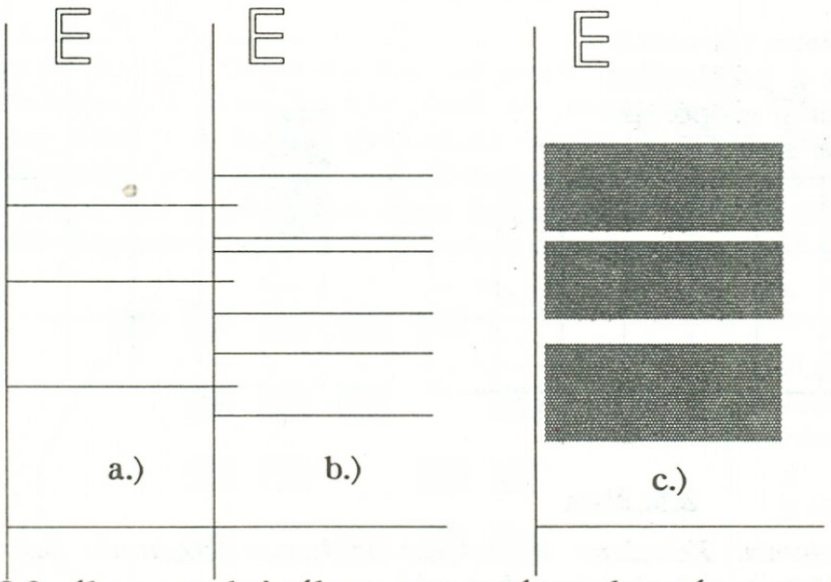
Két proton terében mozgó elektron megtalálási valószínűsége, és az ezt kialakító potenciál. A felrajzolt potenciál két H atom potenciálterének

Az elektron mint "ragasztó" működik, összefogja a két protont.

Tudjuk, hogy az atomok energiaszintjei jól meghatározható, diszkrét értékeket vesznek fel. Ha két egyforma azonos állapotú atomot hozunk közel egymáshoz (a fenti példában a H atomok), akkor az azonos energiaszintek a két elektron kölcsönhatása folytán (taszítás) eltolódnak egymáshoz képest.

2.2. A szilárdtestek sávelmélete

Az atom elektronjaira érvényes Pauli elv lényegében kiterjeszthető a több atomos rendszerekre, így a kristályos anyagokra is. Ha n atomot egyesítünk, akkor az elektronok kölcsönhatása folytán az eredetileg egyetlen energiaszint n darab, egymáshoz igen közeli szintre hasad. A szintfelhasadás mértéke a kölcsönhatás jellegétől függ, az atomok közelítésekor a szintkülönbségek nőnek. Ha n értéke igen nagy, akkor a szintek olyan közel kerülnek egymáshoz, hogy egy - egy sávvá folynak össze.



2.3. ábra

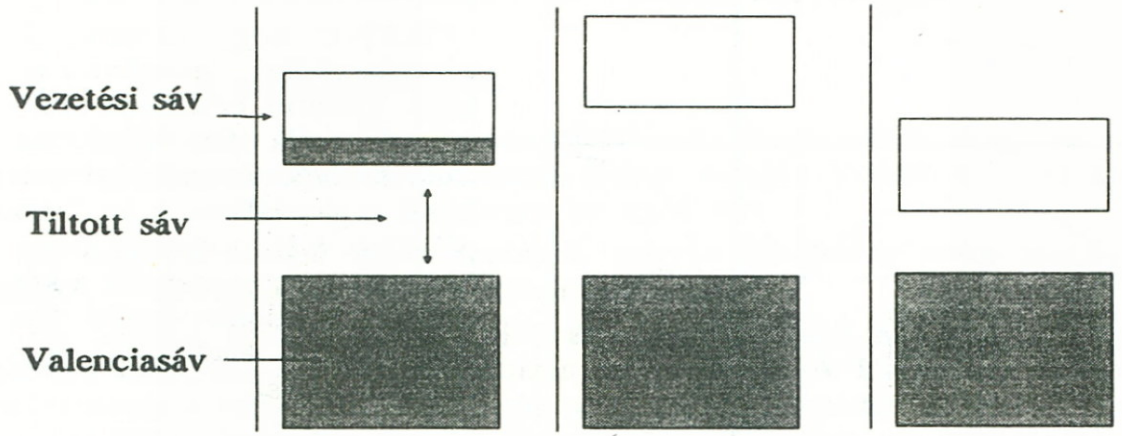
Az a.) esetben egy atom energiaszintjeit,

a b.) esetben ugyanezen atomból felépülő kétatomos molekula energiaszintjeit látjuk.

A c.) ábra n atomból felépülő molekula energiaszerkezetét mutatja.

A 2.3. ábra egy kristályos anyag sávszerkezetét mutatja. Az elektronok megengedett energiaszintjei által kialakított sávokat energiasávoknak, az őket elválasztó tartományokat **tiltott sáv**oknak nevezzük. Az elektronokkal teljesen betöltött energiasávokat **valenci - sáv** oknak, a részlegesen betöltöttet **vezetési sáv**nak nevezzük. A vezetési sáv elnevezés abból adódik, hogy csak azok az elektronok vesznek fel könnyen energiát, vesznek részt a vezetésben, amelyek a saját energiaszintjükhöz közeli üres, betöltetlen, vagy részlegesen betöltött energiaszinteket találnak.

Az energiasávok és tiltott sávok közötti távolságok, valamint a sávok betöltöttsége alapján csoportosíthatjuk a kristályos anyagokat vezető, félvezető, illetve szigetelőanyagokra.



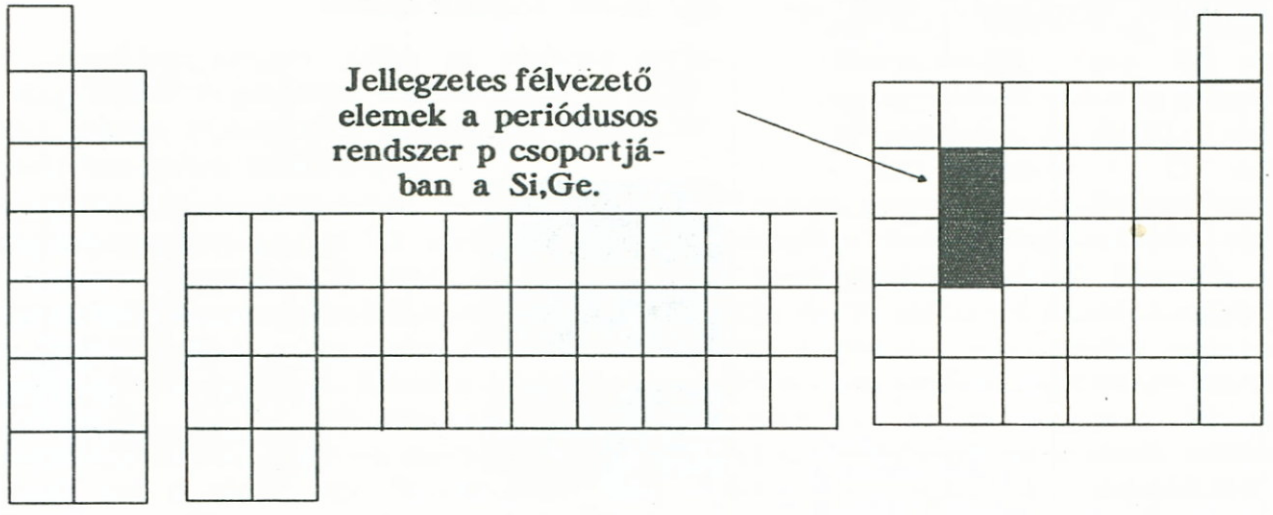
2.4. ábra

Fémek, szigetelők és félvezető anyagok jellegzetes energiaszerkezete.

A fémek esetében a vezetési sáv részben betöltött, a szigetelők esetében üres vezetési sávot és széles tiltott sávot találunk, mig a félvezetők esetén üres a vezetési sáv, de viszonylag keskeny a tiltott sáv szélessége. Nyilvánvaló, hogy a külső erőtér vagy a termikus kölcsönhatás a félvezetők esetén képes a valenciasávból elektronokat a vezetési sávba emelni.

2.3. A félvezetők fizikája

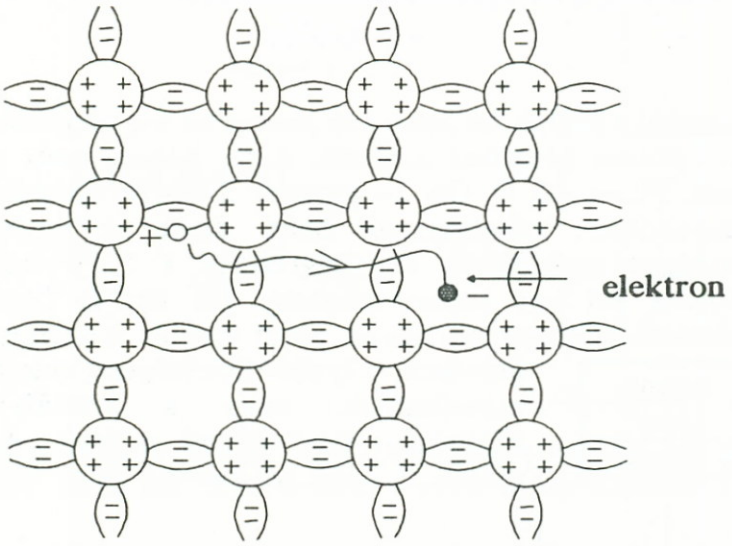
A továbbiakban a félvezető anyagokat tekintjük át. Félvezető anyagon olyan kristályos szilárd testeket értünk, amelyek vezetőképessége a majdnem szabad elektron jelenlétével kapcsolatos és a vezetőképessége a hőmérséklet növekedésével nő.



2.5. ábra

A p csoport 4 vegyértékű elemei. Kovalens kötéssel kristályokat alkotnak. Szerkezetük a gyémántéhoz hasonló.

Ilyen tulajdonságú elemeket a periódusos rendszer p csoportjában találunk.Tegyük fel, hogy a félvezető kristályunk Si atomokból áll. Ezeket kovalens kötés fogja össze. (Azaz a külső elektronhéj felvesz a szomszédos atomoktól egy vegyértékelektront. Így kiegészül az utolsó elektronhéj, szerkezete a nemesgázokéhoz lesz hasonló.) Alapállapotban (-273°C) ezen elektronok kötésben vannak, ezért a külső tér energiáját nem tudják felvenni,



2.6. ábra

A kovalens kötésű félvezetőkristály vázlatos szerkezete valamint a termikus töltéspár látható az ábrán.

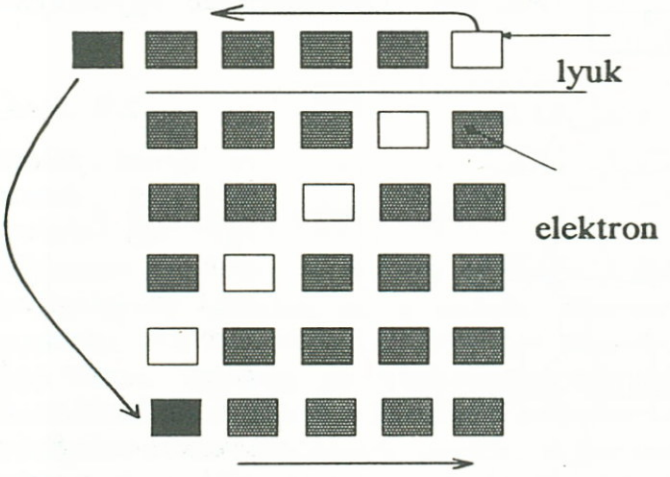
szigetelőként viselkednek, hiszen a tiltott sávba nem léphetnek (a tiltott sáv szélességénél nagyobb energiaközléssel — 1,12 eV— természetesen ezt az „utat" megtehetik).Szobahőmérsékleten a termikus energia elegendő ahhoz, hogy a kötések felszakadjanak és ezzel megteremtődjenek a „szabad" töltéshordozók, azaz a kötésből kiszakadt szabad elektronok. Ezek az elektronok természetesen a vezetési sávba lépnek és felveszik a külső tér energiáját, azaz részt vesznek a vezetésben. A févezetőkben nemcsak elektronok vesznek részt a vezetésben, hanem ezek kilépési helyén képződött lyukak is. A keletkezett töltéshordozók, lyukak és elektronok mindig **töltéspároknak** is nevezzük.

egyenlő számban vannak jelen. Ezért őket **termikus**

Amennyiben feszültséget kapcsolunk a kristályra, akkor a töltéshordozók áramlása megindul. Az elektronok a pozitív, a lyukak a negatív pólus felé vándorolnak. A folyamat hasonló a mozi széksorainak feltöltéséhez. Ha egy későnjövő néző le akar ülni, akkor ez megoldható oly módon, hogy felállítja az egész sort és bemegy a helyére, de oly módon is, ahogyan az elektronok teszik, az üres hely mellett ülő nézők átülnek a szabad helyre. Ekkor a szabad hely — lyuk — kifelé vándorol és a végén a sor szélén jelenik meg, ahová a néző már leülhet. Az elektronok a rácslyukakba ugranak be, majd a hőmérséklet hatására továbblépve vihetik az elektromos áramot. Az áram két összetevőből jön létre, a lyukak és elektronok hordozta áramból.

$$I = I_e + I_{ly}$$

A lyuk és az elektronáram azonban nem egyenlő nagyságú, ugyanis a lyukak **mozgékonyság** a jóval kisebb, mint az elektronoké. (Mozgékonyság alatt az egységnyi térerősség által létrehozott gyorsulást értjük.)A hőmérséklet változása során a félvezető vezetőképessége megváltozik, minél magasabb a hőmérséklet, annál magasabb a kristályon átfolyó áram. Ezt a jelenséget lehet felhasználni hőmérséklet érzékelő eszközök készítésére, amelynek alapvető eleme a termisztor, a tiszta félvezető kristály.



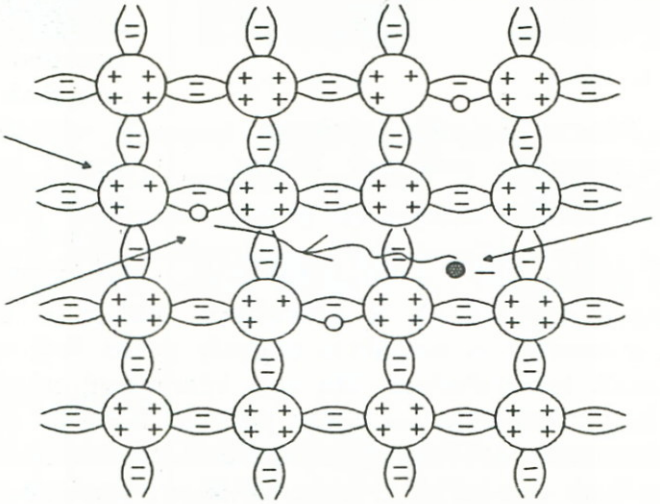
2.7. ábra

Az üres helyek a lyukakat képviselik, a betöltöttek az elektronokat. Az ábrán látható módon a lyukak balra, az elektronok jobbra vándorolnak.

2.3.1. A p típusú vezetés

Ha a félvezető kristályba olyan atomokat viszünk be, amelyek kevesebb vegyértékelektront tartalmaznak mint a szilícium, akkor benne lyukakat hoztunk létre, hiszen nem minden vegyértékkart fog két eleltron betölteni. Pl: az Al, B, Ga, In atomok 3 vegyértékelektronnal rendelkeznek ezért, p típusú szennyezőként alkalmazzuk őket. Ezek az úgynevezett **akceptor** , befogadó atomok.A tényleges szennyezés nagyságrendje 1 : 10^8-hoz, azaz minden százmilliomodik félvezetőatomra jut egy szennyezőatom. Ha azt is figyelembe vesszük, hogy a hőmozgás során kialakuló töltéspárok száma jóval kisebb a szennyezéssel keletkezett lyukakhoz képest, akkor joggal mondhatjuk, hogy a kristályban a többségi töltéshordozó a lyuk. A táblázatból látható módon mintegy két nagyságrend különbség adódik.

Anyag	A lyukak száma
tiszta félvezető	2,5 10 ^{13} 1/cm^3
p típusu félvezető	3,6 10 15 1/cm^3



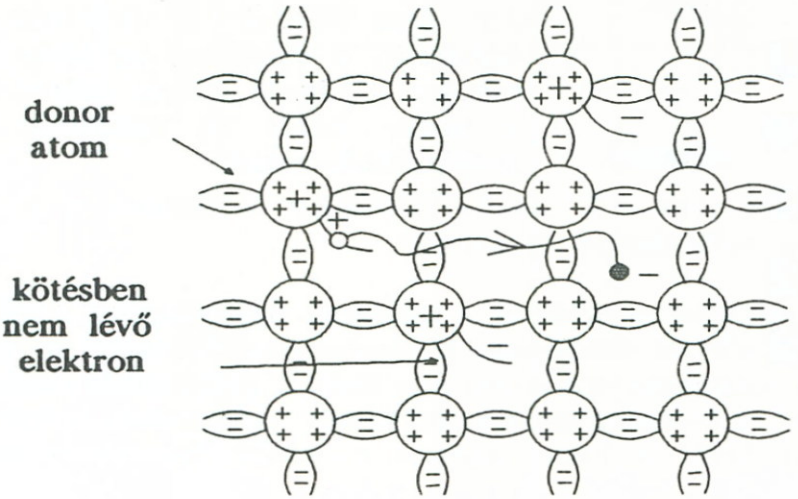
2.8. ábra

Az ábrán a p típusú szennyezés vázlatát láthatjuk.

2.3.2. Az n típusú vezetés

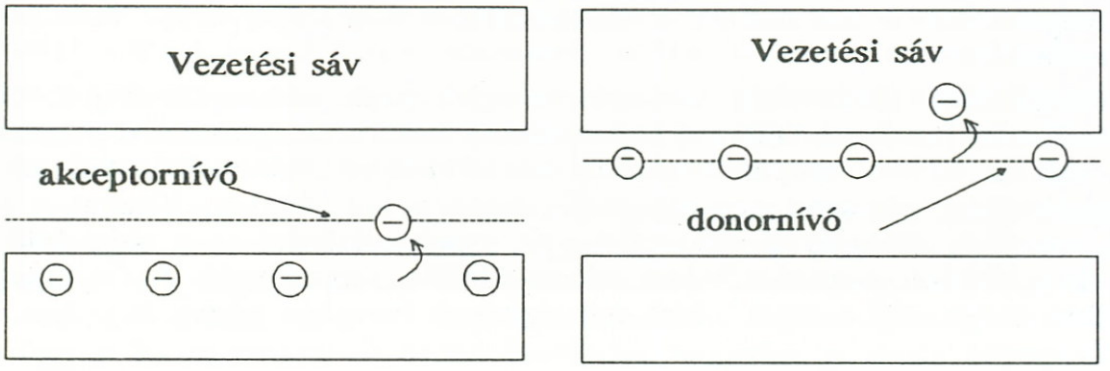
Az n típusú félvezető anyag előállításához 5 vegyértékű elemeket használnak. Ezek az úgynevezett **donor** atomok, amelyek „elektronokat" adnak a kristályhoz. Pl: a P, As, Sb. Természetesen ugyanolyan okokból

Anyag	Az elektronszám
tiszta félvezető	2,5 10 ^{13} 1/cm^3
n típusú félvezető	1,75 10 ^{15} 1/cm^3



2.9. ábra

N típusú félvezetőkristály. A donoratomok egy- egy elektront adnak a vezetéshez.Mivel ezek a kristályrács kötésében nem vesznek részt ,ezért könnyen leszakíthatók, könnyen gyorsíthatók.Energiaszintjük a vezetési sávhoz kerül közel.

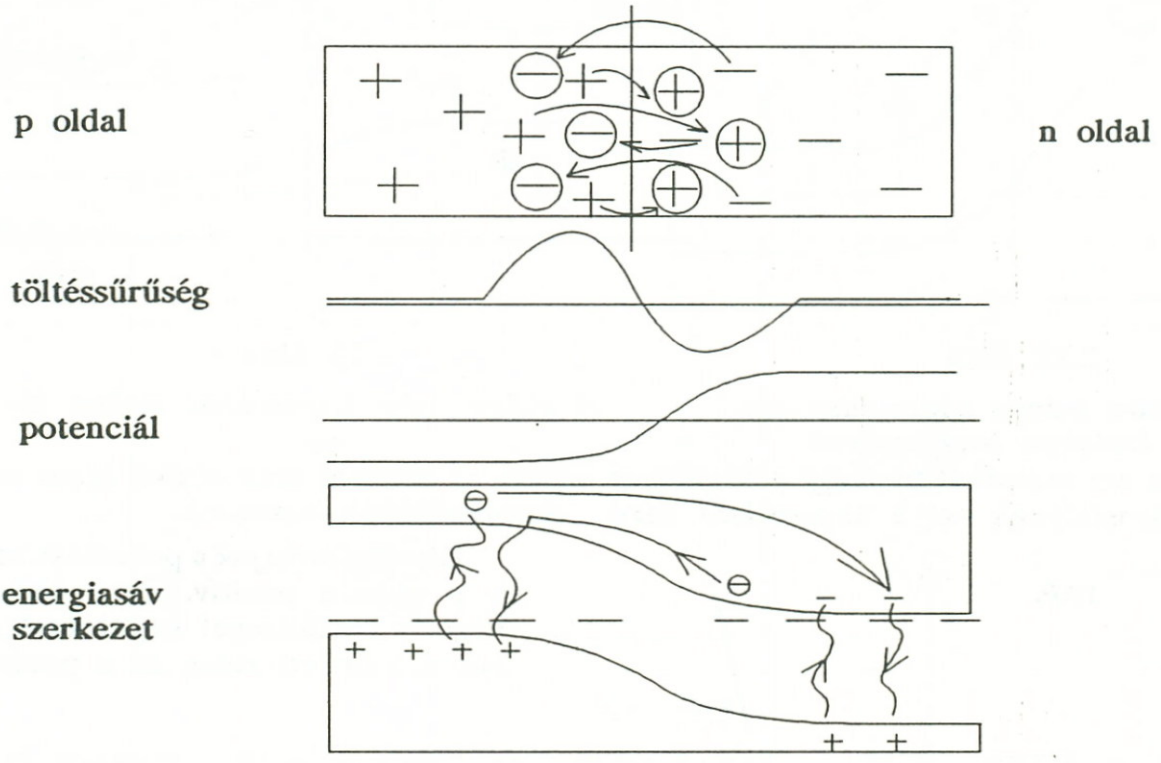


2.10. ábra

A szennyezés során kialakított energianívók. P típusú szennyezés esetén akceptor, n típusú szennyezés esetén donornívókat illesztettünk be a tiltott sávba.

mint az előbb most az elektronok lesznek a többségi töltéshordozók.

2.4. A p -n átmenet



2.11. ábra

2.4.1. A termikus egyensúly

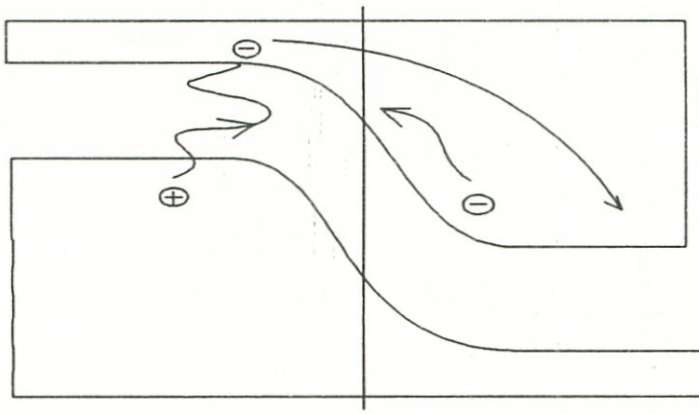
Kapcsoljunk össze egy n és egy p típusú félvezető anyagot. (Ugyanazon félvezető lapkának az egyik oldalán n, a másik oldalán p típusú szennyezést alakítunk ki.) Összekapcsoláskor a két réteg határán egy semleges, úgynevezett kiűrített réteg jön létre. A diffúzió révén ugyanis az elektronok átkerülnek a p oldalra és ott a lyukakkal rekombinálódnak. (A diffúzió a két oldal koncentrációkülönbsége miatt jön létre.) Helyükön a saját rétegükben az elektronok pozitív, a lyukak negatív atomtörzseket hagynak hátra. Ez a diffúziós folyamat addig tart, amíg ki nem alakul a hátrahagyott atomtörzsek miatt

egy potenciálgát, ami e folyamatot a továbbiakban megakadályozza. Természetesen a folyamat dinamikus.

Ha az energiasávos ábrázolását tekintjük a kristálynak, akkor azt látjuk, hogy az n oldali potenciálszint lesüllyed. Diffúzió útján néhány elektron megmássza a potenciálhegyet, átjut a p oldalra és ott beépül a szabad kötési helyre, rekombinálódik az ott lévő lyukkal. Így látszólag elveszett egy-egy töltés mind a két oldalon. Viszont a p oldalon termikusan létrejövő töltéspárok elektron tagját a potenciállejtő az n oldal felé gyorsítja, ezért az n és p oldalon maradó töltések száma időátlagban egyenlő. A folyamat teljesen analóg módon megy végbe az n oldal felé igyekvő lyukakra nézve is.

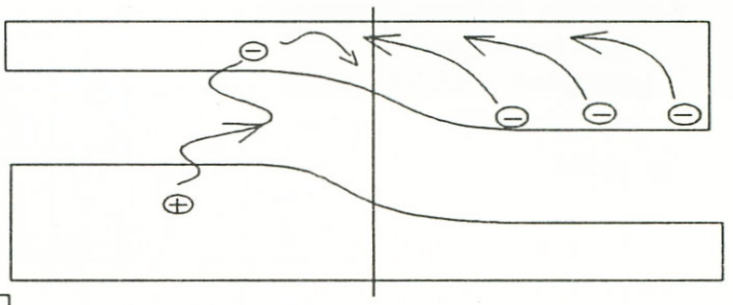
2.4.2. A p - n átmenet feszültség alatt (Dióda)

Amennyiben az előbbiekben tárgyalt kristály p oldalára a feszültségforrás negatív, az n oldalára pozitív kapcsait erősítjük, akkor a helyzet megváltozik. A kiűrített réteg kiszélesedik, a potenciálhegy magasabbá válik, egyre kevesebb elektron tud átjutni rajta. A diffúzió útján átjutó elektronok száma a potenciálhegy magasságával exponenciálisan



2.12. ábra

A dióda záró irányu kapcsolása esetén kialakult energiasávok

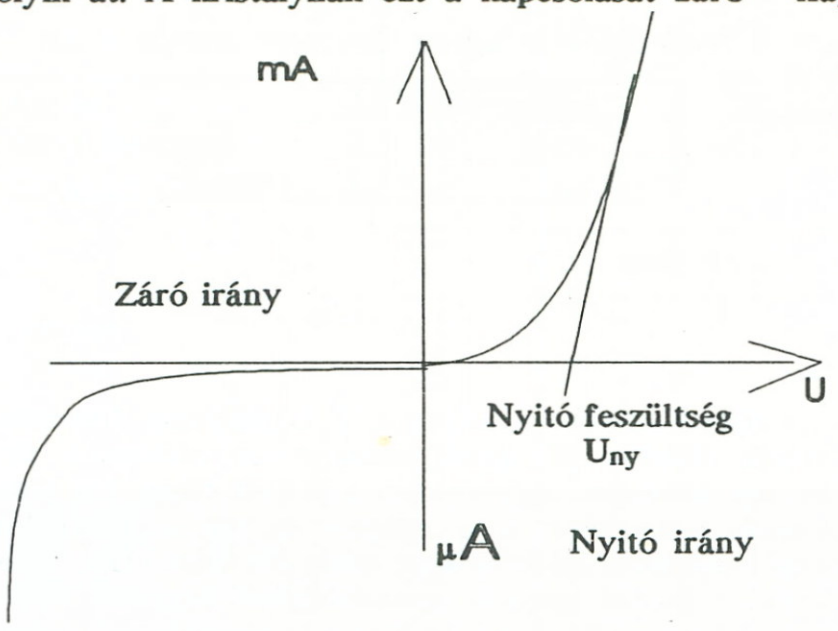


2.13. ábra

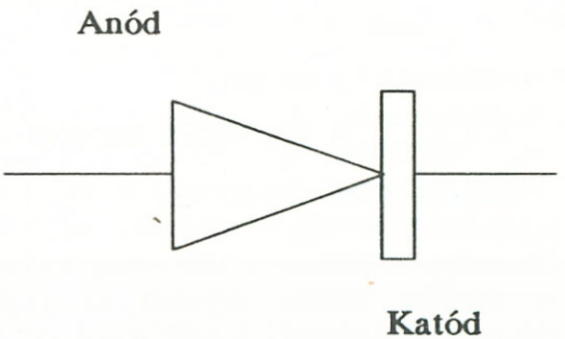
A dióda nyitó kapcsolása mellett kialakult energiasávok

csökken. Azaz azt mondhatjuk,hogy a kristályon ebben az esetben számottevő áram nem folyik át. A kristálynak ezt a kapcsolását **záró kapcsolásnak** nevezzük.

Ha megfordítjuk a polaritást, azaz a p oldalra pozitív, az n oldalra negatív feszültséget kapcsolunk,akkor a kiűrített réteg és a potenci-



2.14. ábra



2.15. ábra

álgát lecsökken és egyre több elektron képes megmászni a potenciállejtőt. Exponenciálisan megnő a töltéshordozók árama. Ezt a kapcsolást **nyitó irányú kapcsolásnak** nevezzük.

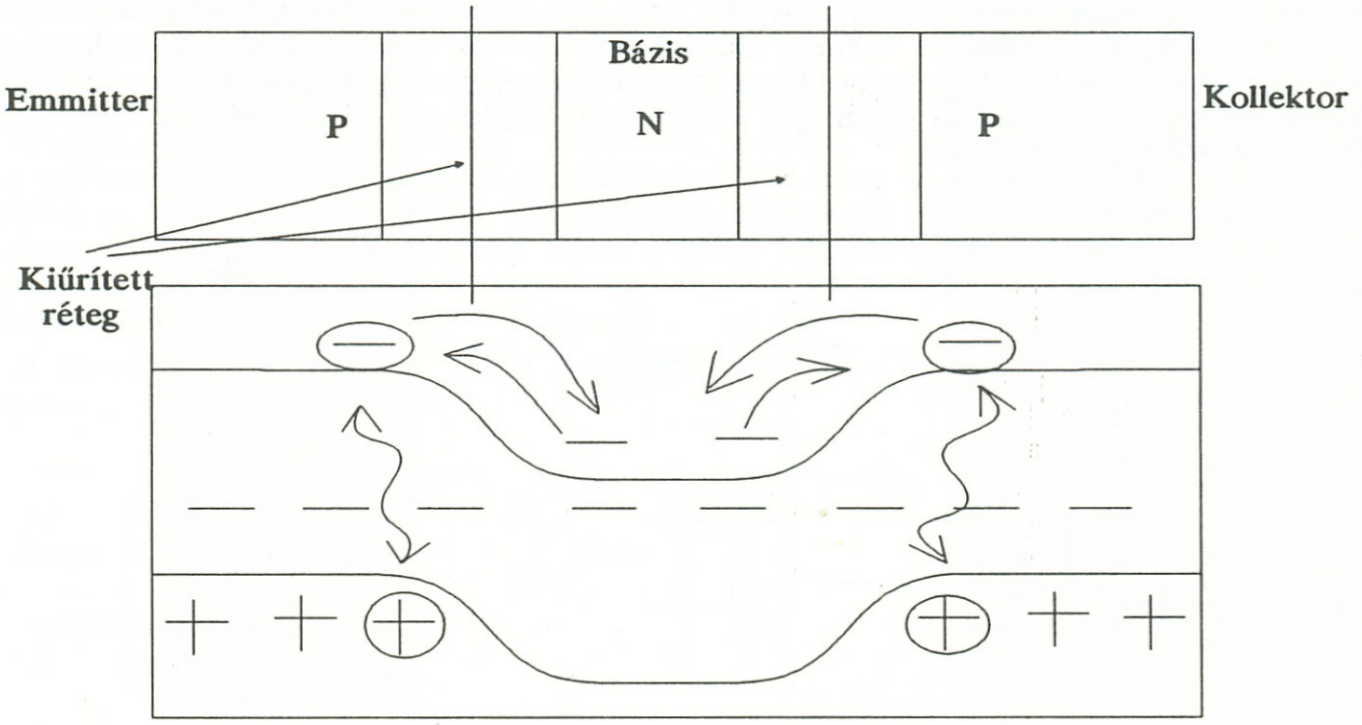
Az eszköz viselkedése tehát igen figyelemreméltó, ugyanis az egyik irányú polaritás esetén vezeti, a másik irányú polaritás esetén pedig nem vezeti az áramot. Szelepként működik. Az ilyen tulajdonságú félvezető eszközt diódának nevezzük.

A dióda karakterisztikáját látva egy fontos megállapítást tehetünk, mely szerint a dióda csak egy jól meghatározott feszültségnél — **nyitó feszültség** — nyit ki, azaz enged át a számottevő nagyságú áramot. Ez egyben azt is jelenti,hogy vezetés közben rajta ekkora feszültség fog esni. (Ezt a megállapításunkat a logikai áramkörök működésének leírásakor fogjuk hasznosítani.) A nyitó feszültség nagysága a dióda anyagától függ, Si dióda esetén kb. 0,5 ... 0,8 V

2.5. A rétegtranzisztor

2.5.1. A feszültségmentes állapot

Ha olyan félvezető kristályt készítünk, amelynek két vége azonos típusú, közepe



2.16. ábra

A tranzisztor termikus egyensúlyát mutatja ábra. A dióda egyensúlyához hasonló viselkedést találunk mind a két P - N átmenetnél.

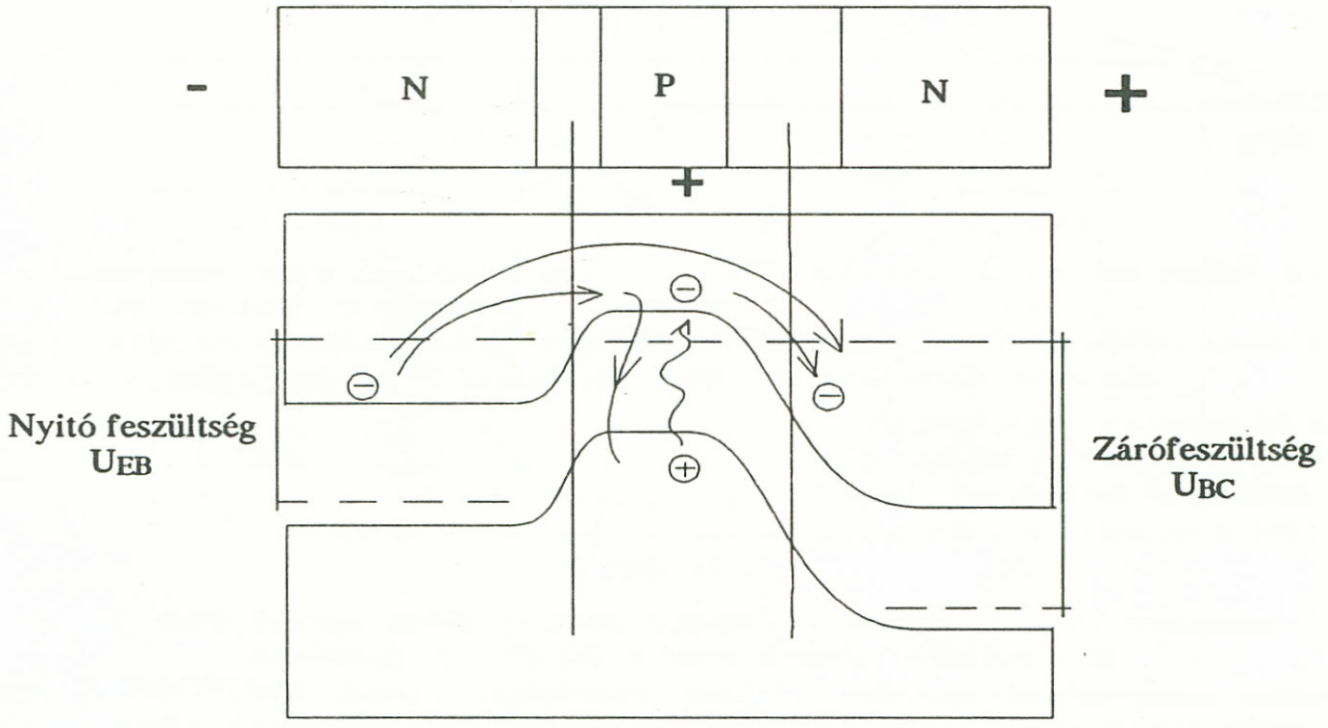
ellentétes szennyezéssel rendelkezik, akkor tranzisztort kapunk. Ilyen módon kétféle tranzisztor készíthető: n-p-n, illetve p-n-p . A rétegek elnevezése balról jobbra haladva emmitter, bázis, kollektor. A termikus egyensúly esetén mind a két átmenetnél kialakul a diódánál már megismert jelenség.Az energiaszintek ennek megfelelően tolódnak el. Kezdetben diffúzió révén, n-ből p-be haladnak az elektronok. A p rétegben rekombinálódnak, kialakul egy mély potenciálgát, amely megakadályozza az elektronok további átjutását. A p oldalon elveszett pozitív töltést a termikusan keletkezett töltéspár pozitív (lyuk) tagja pótolja, az n oldalon hiányzó elektront a p oldalon keletkező töltéspár elektron tagja egyenlíti ki, „legurulva" a potenciállejtőn.

A tranzisztorok működése szempontjából fontos,hogy a középső réteget kissebb töltéstöbblettel lássuk el a gyártás során. Természetesen a folyamatok iránya az n-p-n tranzisztornál éppen ellenkező irányú.

2.5.2. Az áramkörbe kapcsolt tranzisztor

Ha az emitter - bázis diódára nyitó irányú, a bázis kollektor diódára az előbbi feszültség értéknél jóval nagyobb záró irányú feszültséget kapcsolunk, akkor az elöbbi energiaviszonyok lényegesen megváltoznak.

Vizsgáljuk most meg az n-p-n tranzisztor viselkedését a fent leírt kapcsolás mellett.Az emitterből a nyitó irányú kapcsolás miatt megnő a diffúzió útján a bázis felé tartó elektronok árama.(A potenciállejtő szintje lecsökkent.) Az elektronok egy része a bázisban rekombinálódik az ott talált lyukakkal.Mivel a bázisréteget úgy készítik el,hogy a vastagsága és az akceptoratomok koncentrációja jóval kisebb legyen, mint a szomszédos rétegeké, az elektronoknak csak egy része semmisül meg.Az elektronáram megmaradó része folytatja az útját a kollektorréteg felé. Az ott található külső elektromos erőtér az elektront innen gyorsan kiszippantja. Észrevehetjük,hogy ha az emitter-bázis feszültséget megváltoztatjuk, akkor változtatjuk egyúttal az áramba bekerülő elektronok számát, tehát az elelktromos áram nagyságát is. A kollektorrétegre kapcsolt feszültséggel pedig a bázisban rekombinálódni készülő elektronok sebességét állíthatjuk be,így befolyásolatjuk a rekombináció nagyságát, azaz az átjutó áram erősségét.



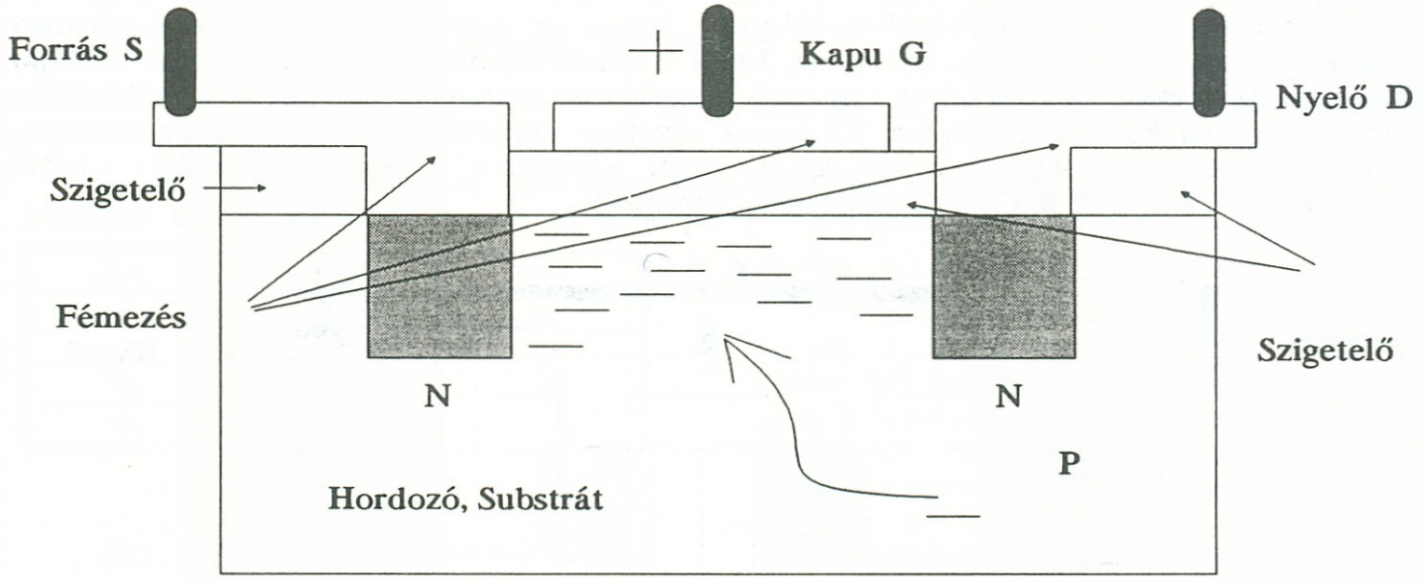
2.17. ábra

A tranzisztor müködése feszültség alatt.

2.6. A MOS tranzisztor

A tranzisztorok másik nagy családja az úgynevezett fém- oxid- félvezető (Metal - Oxid - Semiconductor -Field - Effect - Tranzisztor ,MOSFET). Jelentősége az utóbbi években nagymértékben megntőtt, hiszen kedvező áramfelvétele és teljesítményviszonyai

fontos szerephez jutottak a mikroprocesszorok és nagykapacitású memóriák világában. Vizsgáljuk meg a továbbiakban egy n csatornás MOS tranzisztor felépítését és működését.

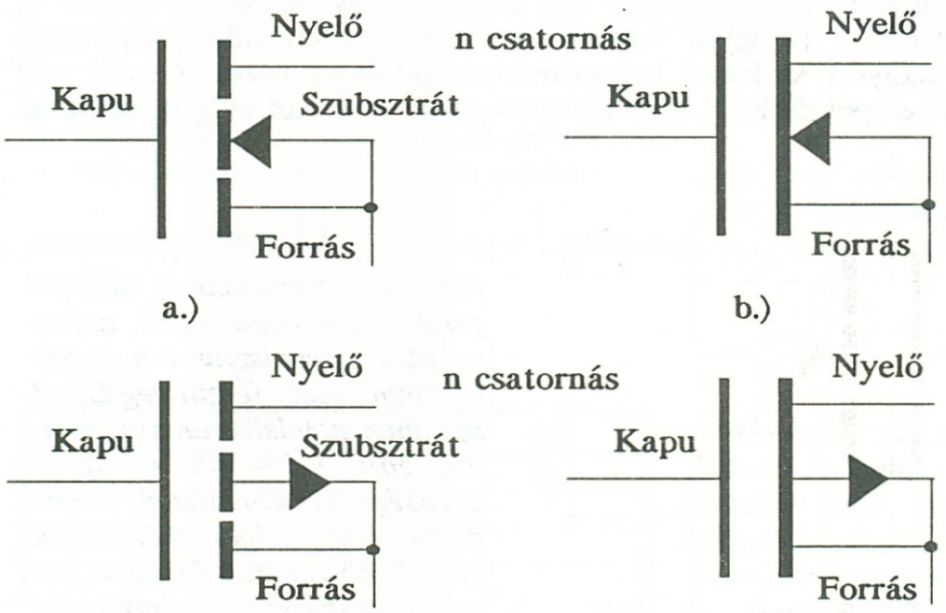


2.18. ábra

MOS tranzisztor felépítése és elnevezései

A szubsztrát vagy hordozó nagy ellenállású p típusu szilícium kristály.Ebbe a rétegbe két különálló n típusú szigetet diffundáltatnak be. Ezek a forrás (Source) és a nyelő (Drain). A struktúra felületét ez után szilíciumoxidréteggel vonják be, (ez igen jó szigetelő anyag) majd a forrás és a nyelő felett ablakot vágnak rajta. Ide kerül majd a fém kivezetés. Az n típusú rétegek között kialakuló csatorna felett szintén készítenek egy fémezést. Ez lesz a kapu (Gate) elektróda.

A hordozó és a csatorna típusát megváltoztatva a p csatornás MOS tranzisztorhoz jutunk.

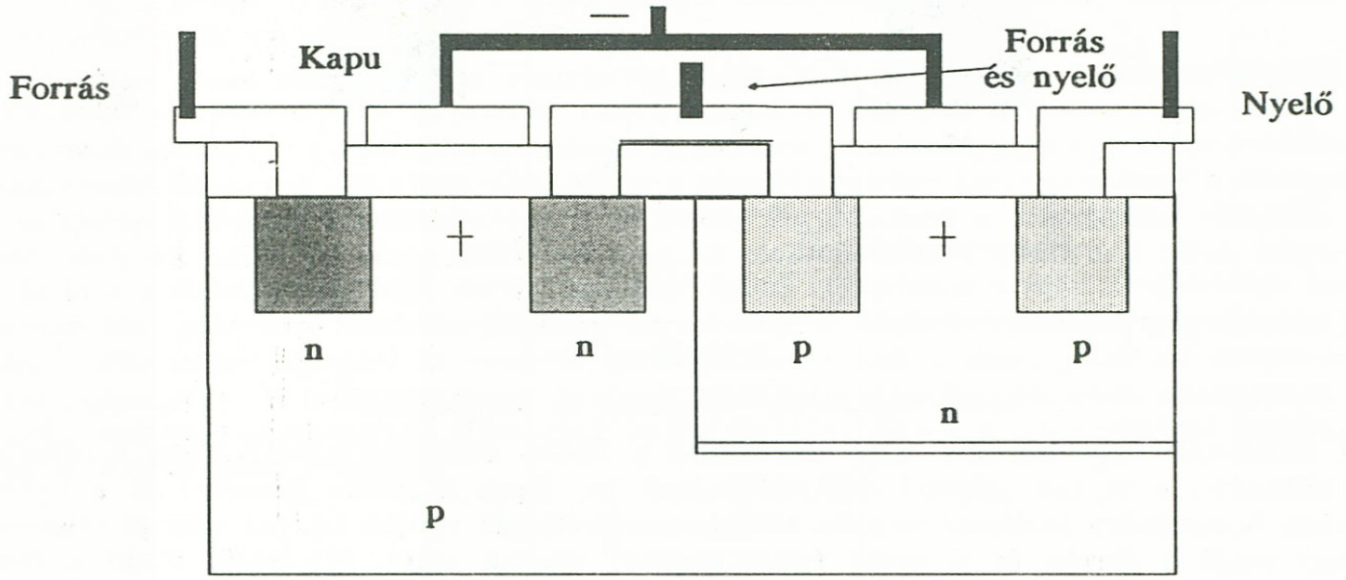


2.19. ábra

Az a.) ábra a növekményes, a b.) ábra a kiürítéses MOS tranzisztor rajzjelét mutatja.

Tekintsük át a tranzisztor működését. Ha a tranzisztor kapujára pozitív feszültséget kapcsolunk, akkor a kialakított fém - oxid -félvezető kapacitás miatt a csatorna negatív töltésű lesz.(Elektromos megosztás jelensége) A p mezőben tehát **indukált n réteg** alakul ki a forrás és a nyelő között, amelynek ellenállása igen kicsiny. Ha feszültségforrás

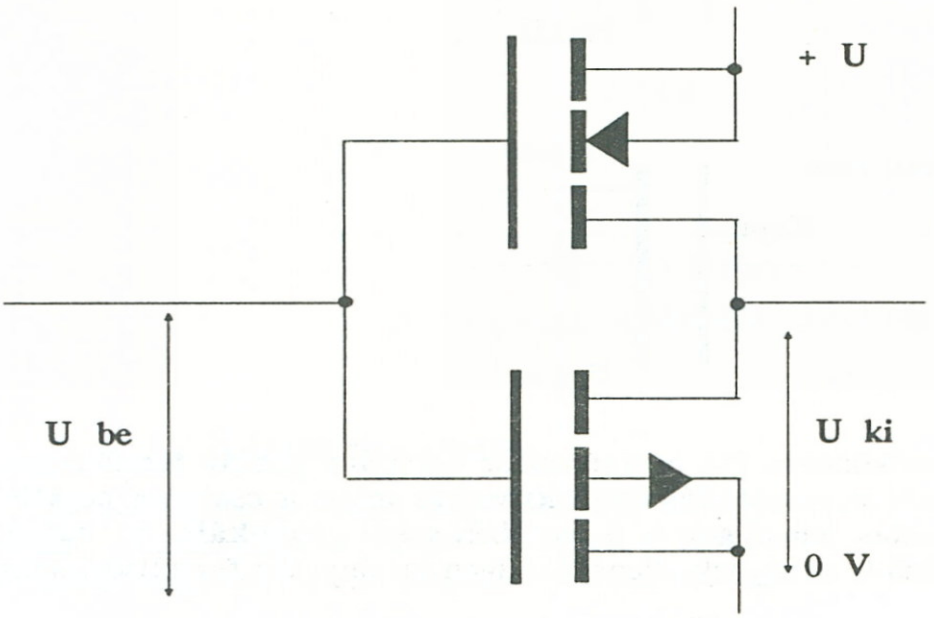
sarkait rákötjük az elektródákra, akkor megindulhat az elektromos áram. A kapu ellentétes irányú (negatív) feszültsége esetén a csatorna ellenállása végtelen naggyá válik. Az esetlegesen termikusan keletkező elektronokat a kapu alatt kialakuló negatív potenciál elhajtja a csatornától távolabbi helyekre.Vagyis ebben az esetben elektromos áram biztosan nem folyik az eszközön. A p csatornás MOS teljesen hasonlóan működik, csak a kapu ellentétes polaritása esetén vezet (negatív feszültsége mellett). Az alkatrész elkészítése nem túlságosan bonyolult feladat. Ráadásul egyetlen szubsztráton létre tudjuk hozni az n és p csatornás MOS tranzisztorokat, ezzel kialakítva a **komplementer MOS** tranzisztort, amelynek a CMOS elnevezést adhatjuk.



2.20. ábra

Egy szubsztráton kialakított P és N csatornás MOS tranzisztor

A CMOS tranzisztor jellegzetessége abban mutatkozik meg, hogy feszültség alatt csak az egyik tranzisztort találjuk vezető állapotban. Ez azt jelenti, hogy kitűnő kapcsoló és invertáló eszközhöz jutunk, hiszen ha az egyik tranzisztora mindig zárt, akkor rajta csak minimális áram folyhat, ami nagyon kedvező teljesítményfelvételhez vezet. (Ezért vált például a kvarcórák elektronikai alapeszközévé. Egy gombelemmel akár két évig is működik az óra.)



2.21. ábra

Ha a bemenetre pozitív feszültséget kapcsolunk, akkor az alsó tranzisztor kinyit,rajta nem esik feszültség.Ezzel szemben a felső tranzisztoron, ami zárt, mérhetjük a tápfeszültséget.A kimeneten ekkor nulla feszültséget mérhetünk Ellenkező esetben fordítva.Tehát a tranzisztor inverterként dolgozik.

3. Logikai áramkörök megvalósítása

3.1. A pozitív és negatív logika

Ebben a fejezetben megmutatjuk, hogyan állítjuk elő a legfontosabb logikai elemek elektromos megfelelőit.

A logikai áramkörökben minden egyes kapu bemeneti állapotához egy jól meghatározott kimeneti állapot tartozik. Ezeket H és L jellel különböztetjük meg.

U1	U2	Uki
L	L	H
L	H	L
H	L	L
H	H	L

3.1. táblázat

Egy lehetséges hozzárendelés.

» H, ha a feszültség egy előre megadott Umax - nál nagyobb.

» L ,ha a feszültség egy előre megadott Umin- nál kisebb.Valamely kapu működése a szinttáblázatával adható meg. A kapu feszültsége által meghatározott függvény logikai szempontból nem teljesen egyértelmű, ugyanis a feszültség és logikai szint hozzárendelésekre két lehetőségünk adódik.

» A H szinthez rendeljük a logikai 1 és az L szinthez a logikai 0 szintet, vagy

» a H szinthez rendeljük a 0 és az L szinthez az 1 logikai értékeket.

Ennek megfelelően kétféle logikáról beszélhetünk.

3.1.1. Pozitív logika

X1	X2	Yki
0	0	1
0	1	0
1	0	0
1	1	0

3.2. táblázat

A feszültségszinteknek megfelelő hozzárendelés pozitív logika esetén.

Ha az első hozzárendelési módot választjuk, akkor a fenti 3.1. táblázatunk a következő alakot ölti. A diszjunktív normál alak alkalmazásával a táblázattal megadott függvényt kifejezhetjük. Látható , hogy a De Morgan szabályt alkalmazva a hozzárendelés alapján NEM-VAGY NOR

$$Yki = \overline{X1} * \overline{X2} = \overline{X1 + X2}$$

kapcsolat adódik. Összefogalva: **pozitív logikáról** beszélünk akkor, ha a magas logikai értéknek pozítívabb feszültséget feleltetünk meg,mint az alacsony logikai értékhez rendelt feszültség.

3.1.2. Negatív logika

X1	X2	Yki
1	1	0
1	0	1
0	1	1
0	0	1

3.3. táblázat

A feszültségszinteknek megfelelő hozzárendelés negatív logika esetén.

Ha tekintjük a második hozzárendelési módot, akkor nyerjük a 3.3. táblázatot.A konjunktív normál alak segítségével ismét kifejthetjük a függvény alakját.
Látható , hogy a De Morgan szabályt alkalmazva a hozzárendelés alapján NEM-ÉS NAND

$$Yki = \overline{X1} + \overline{X2} = \overline{X1 * X2}$$

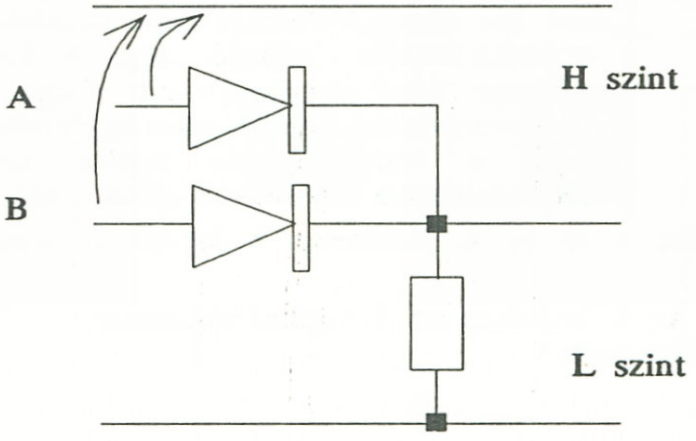
kapcsolat adódik. Összefogalva: **negatív logi-**

káról beszélünk akkor,ha a magas logikai értéknek negatívabb feszültséget feleltetünk meg,mint az alacsony logikai értékhez rendelt feszültség.

Ennek alapján a továbbiakban kialakított áramkörök különböző logikai kapuknak használhatók attól függően ,hogy pozitív vagy negatív logikában gondolkodunk. A feladataink szempontjából azonban ajánlott, hogy mielőtt hozzálátunk azok kivitelezéséhez döntsük el,hogy melyik logikát használjuk és ne térjünk el attól.

3.2. A dióda-tranzisztor logika

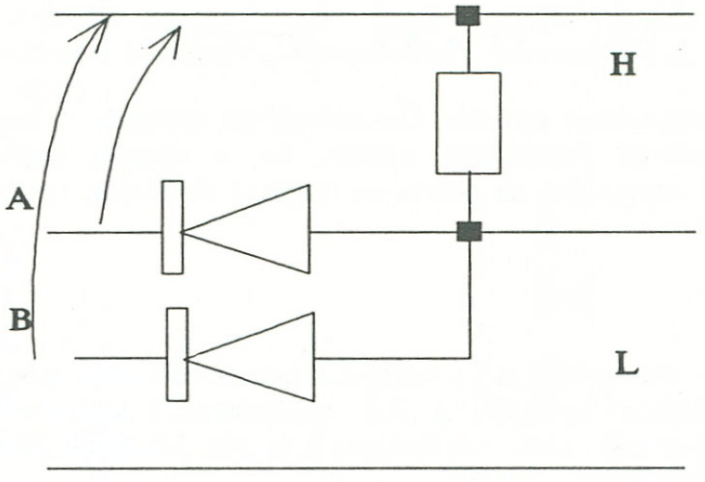
A dióda működésének leírásakor megtudtuk,hogy az eszközt nyitó irányú feszültségre kapcsolva (anódja pozitív, katódja negatív feszültséget kap) rajta számottevő áram akkor folyik, ha a nyitófeszültségnél nagyobbat kapcsoltunk rá. Záróirányban kötve nem vezet,csak minimális mikroamperek folynak rajta keresztül. Ezeket a tulajdonságait kihasználva elkészíthetjük az alapkapukat. Feladatunk tehát az,hogy a matematikailogikából ismert minimális elemkészletet létrehozzunk,amelyekkel tetszőleges logikai függvény előállítható.Ennek egyfajta megoldása az, ha kialakítjuk a VAGY, az ÉS és a NEM kaput. Nézzük sorra ezek megvalósítását. Állapodjunk meg abban, az egyszerűség kedvéért, hogy H szinten a továbbiakban 5 V, L szint alatt 0 V feszültséget értünk.



3.1. ábra

Vagy kapu megvalósítása

A **VAGY logikai kapu** kialakítása pozitív logika esetén a következő módon lehetséges. (3.1. ábra) Kétbemenetü eszközünk A pontját H szintre kötve a rákapcsolt dióda vezet, hiszen nyitóirányban van kapcsolva. Az ellenálláson áram folyik, ami ezen potenciálkülönbséget hoz létre. Ez a feszültség H szintnél a dióda nyitó feszültségével alacsonyabb. (A diódán ekkora feszültség esett).



3.2. ábra

És kapu megvalosítása

Ha a B pontot húzzuk fel H szintre, akkor ugyanilyen megállapítást tehetünk.Nem változik lényegesen a helyzet az A és B pontok H szintre kötése esetén sem.

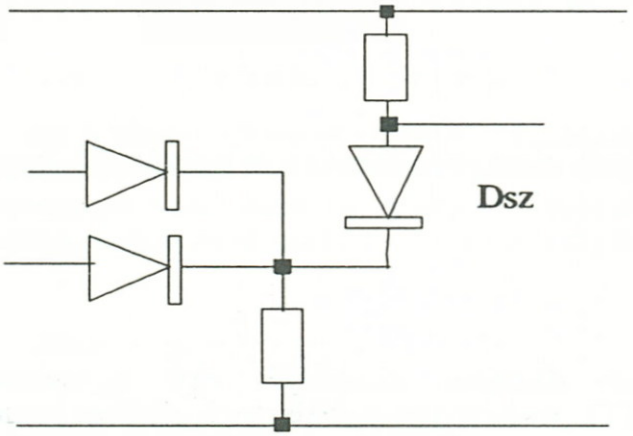
Ha viszont mind a két bemenetet L szintre kapcsoljuk,akkor mivel a dióda anódja és katódja is azonos potenciálra kerül, rajta nem folyik áram, ezért a kimeneten L szint mérhető.Ezzel tehát előállítottuk a VAGY kaput, mert csak A és B alacsony szintje mellett kaptunk alacsony kimenetet, azaz 0 logikai értéket. Minden más esetben magas szinthez - 1 - jutottunk.

Az **ÉS kapu** kialakítása pozitív logika esetén a következő módon lehetséges. (3.2. ábra)

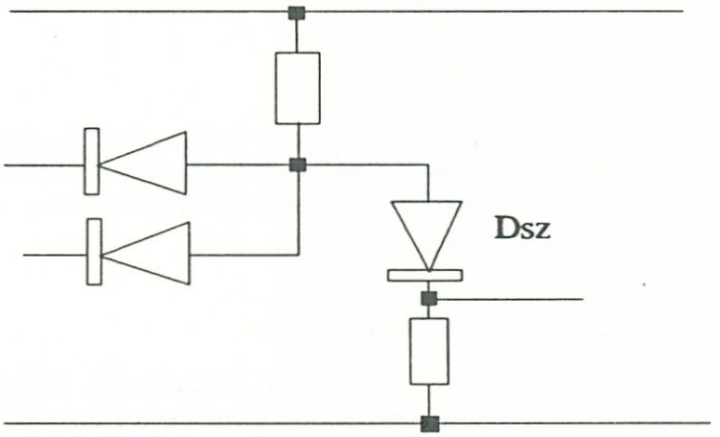
Az eszköz A pontját H szintre kötve, mivel a hozzá tartozó dióda anódja és katódja ugyanarra a potenciálra kerül,nem vezet. Eszközünk állapota csak a B pont logikai értékétől függ. A B pontot lekötve az L szintre a rákapcsolt dióda kinyit, megindul

az ellenálláson az áram. A kimeneten csak a diódán eső feszültség mérhető.A helyzet akkor sem változik, ha megcseréljük a bemenetek feszültségszintjét. Ha az eszköznek mind a két bemenete a H szintre kerül, természetesen egyik dióda sem vezet, ami azt jelenti, hogy a kimeneten H szintet mérünk.

Ha most mindkét bemenetet L szintre kapcsoljuk, akkor mindkét dióda vezet, a kimeneten ekkor a diódán eső feszültség merhető, azaz alacsony logikai szint. Ezzel



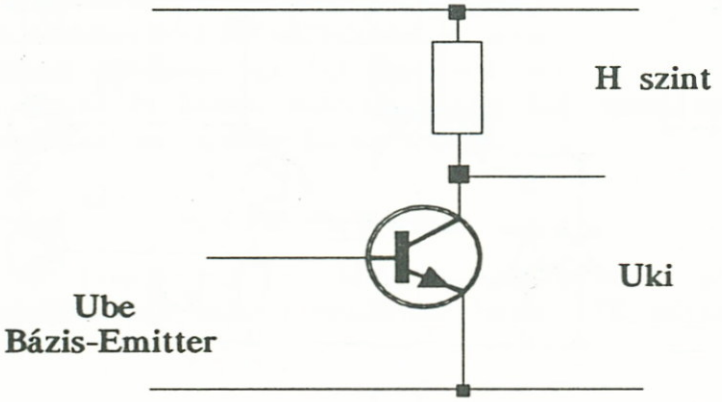
3.3. ábra



3.4. ábra

igazoltuk,hogy az áramkör ÉS- kaput valósít meg, mert csak mindkét bemenet magas szintje esetén kapunk a kimeneten magas logikai értéket.

Mindkét kapu működése során megfigyelhetjük,hogy a feszültségszintek eltolódnak a dióda nyitó feszültségének értékével. Ez azonban komoly problémához vezethet, ha több logikai kaput kell egymás után kapcsolnunk. Ugyanis egyre alacsonyabb bemeneti feszültségeket fogunk mérni. Végül a bemeneti H szint teljesen lecsökken és L szintként érzékelhető. Ennek megakadályozására szinteltoló áramkört kell beiktatnunk. Ez elvileg lehetne egy olyan telep is, amely éppen a dióda nyitófeszültségének nagyságát szolgáltatná.De elképzelhető,hogy ez a megoldás is lehetetlen helyzetet teremt méreteinél fogva. Így nincs más lehetőségünk, hogy a közismert közmondás alapján „ kutya harapását szőrivel" dióda segítségével oldjuk meg a problémát.



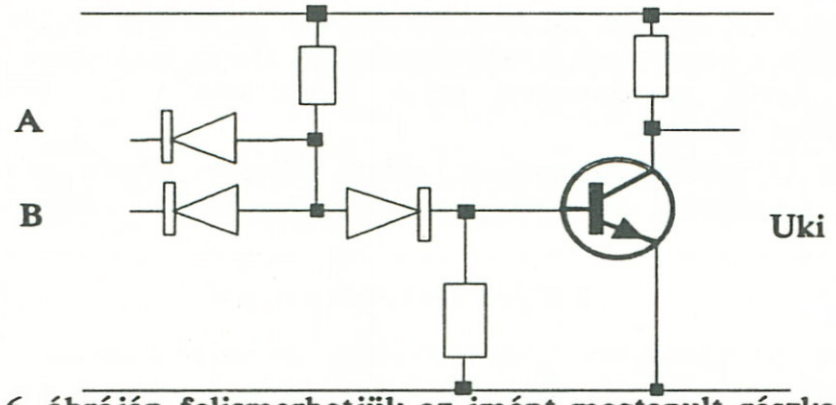
3.5. ábra

A tranzisztor alapkapcsolása

(3.3—3.4.ábra)

A következő áramköri elem, amire szükségünk van az **inverter** . Ez diódás eszközökkel nem alakítható ki, de tranzisztor segítségével igen. A tranzisztor működésének leírásakor láttuk, hogy ha a bázis-emitter diódájára nyitó feszültséget kapcsolunk, akkor az emitter-kollektor kapcsokon áram folyik.Ez természetesen azt is jelenti, hogy ekkor a tranzisztorunk emitter-kollektor ellenállása kicsiny, ezért rajta kis feszültség esik. Ha tehát a bázis-emitter feszültséget H szintre emeljük, (a nyitó feszültségnél jóval nagyobb) akkor a tranzisztor emitter-kollektor kapcsai között alacsony feszültséget mérhetünk. Ha viszont a bemeneti kapcsokat L szintre kötjük, akkor a tranzisztor lezár, a teljes feszültség rajta fog esni, így kimenete H szintre kerül.

Ezek után könnyedén kialakíthatunk a DTL kapukból összetettebb áramköröket is.Példának okáért nézzük meg, hogyan lehet megoldani egy NAND kapu kialakítását.



3.6. ábra

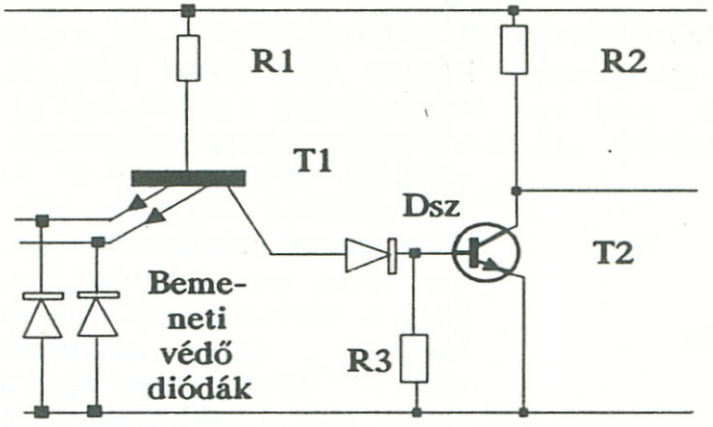
DTL logikában működő NAND kapu. Az ábrán jól elkülöníthető az ÉS, a szinteltoló és invetáló áramkör.

A 3.6. ábráján felismerhetjük az imént megtanult részkapcsolásokat, a diódás ÉS kaput, a szinteltoló áramkört és az invertert.

3.3. A TTL logika

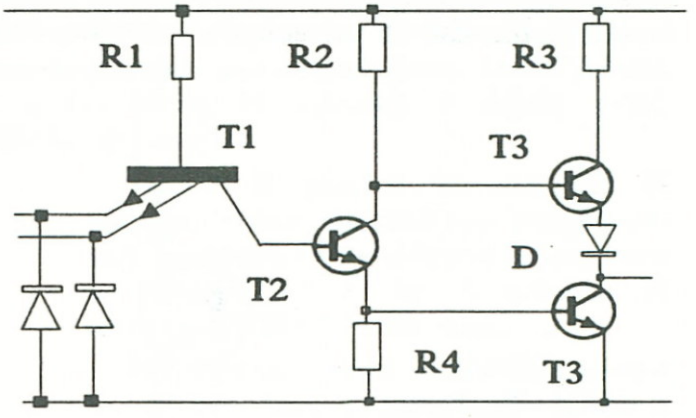
Ha a DTL kapukban szereplő bemeneti diódákat kicseréljük egy úgynevezett többemitteres tranzisztorra,akkor az úgynevezett TTL, tranzisztor-tranzisztor logikához jutunk. A többemitteres tranzisztorokat csak az integrált áramkörökben használjuk,ezeknek diszkrét megvalósításuk nincs.

Vizsgáljuk meg példaként a TTL NAND kapu működését. (3.7. ábra) Induljunk ki onnan,hogy a tranzisztor szimmetrikus áramköri elem - az emittere és kollektora felcserélhető. (Ebben az esetben a bázis-kollektor dióda nyitó, a bázis-emitter dióda záró irányban van előfeszítve.) Ezt a működési módot inverz üzemmódnak nevezzük,



3.7. ábra

TTL NAND kapu felépítése



3.8. ábra

Ellenütemü TTL NAND kapu

amely a normálistól csak kisebb áramerősítési tényezőjében tér el. (Áramerősítési tényezőn a kollektor és bázisáram viszonyát értjük.)

Ha a bemeneti tranzisztornak az összes emittere H szinten van, a bázis pozitívabb, mint a kollektor, így a tranzisztor inverz üzemmódban dolgozik. A bázis-kolektor diódája nyitó irányban van előfeszíve, ezért a T2 tranzisztor bázisába folyó áram azt kinyitja.(Az R3 ellenálláson a T2 nyitásához elegendő feszültség esik.)Ez egyúttal azt is jelenti,hogy a kollektorkörben áram folyik, a kimeneten csak a tranzisztoron eső feszültség jelenik meg (kb. 0,6 V), ami L szintnek felel meg.

Ha a tranzisztor bármely bemenetére L szintet kapcsolunk, akkor a megfelelő bázis-emitter dióda kinyit, áram folyik a meghajtó fokozatba, amelynek értékét az R1 ellenállás korlátozza.A T1 tranzisztor telítésbe megy át,azaz rajta csak kb 0,2 V esik, ami nem elegendő a Dsz dióda kinyitásához, így a T2 tranzisztor nyitásához sem.Tehát annak kimenetén H szint jelenik meg. Ez természetesen a NAND kapu vázlata, a valóságos megjelenése jóval bonyolultabb,precíz müködéséhez több tranzisztort kell felhasználni. Például biztosítani kell az áramkör be és kimenetének rövidzárvédelmét. (3.8.ábra)

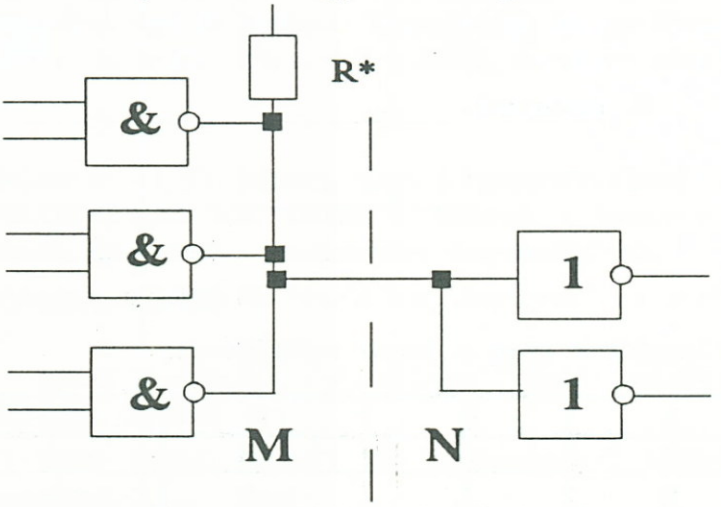
3.4. Huzalozott logika

Ha a kimenet nem rövidzárvédett, úgynevezett nyitott kollektoros,akkor huzalozott logikát is készíthetünk.Erre akkor kerülhet sor,ha olyan áramkört akarunk készíteni,amelynek több bemenete van, mint az integrált áramköri sorozatban megtalálható alkatrésznek. Ekkor külön kollektorellenállásra van szükségünk, amely értékét úgy kell megválasztani, hogy

$$Rmin = \frac{4{,}6K}{(16 - 1{,}6 * N)}$$

$$Rmax = \frac{2{,}6K}{(0{,}25 * M + 0{,}04 * N)}$$

minden körülmények között garantálni lehessen a H szintet,illetve azt,hogy a kapubemenetek terhelőárama nem lépheti túl az áramköri egységre megadott legnagyobb 16 mA kollektoráramot. Az alábbi Rmin és Rmax képletekben szereplő N a meghajtott bemenetek számát, M a meghajtó fokozatokkimemeteinekszámátjelenti. Az alkalmazott R* elelnállásnak nem szabad túllépnie az 1,5 Kohmot és az Rmin és Rmax határok között kell maradnia. Az 1,5 K túllépése esetén lényegesen megnőhet az eszköz futási ideje.

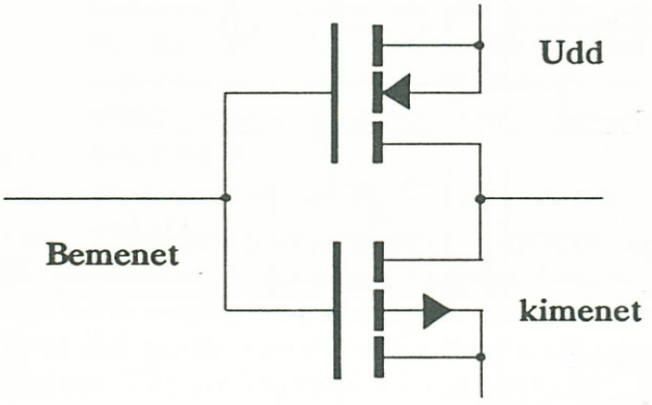


3.1. ábra

Huzalozott logikai kapuk.

3.5. A CMOS logika

A komplementer MOS áramkörökből készített logikai kapcsolások lényegesen egyszerűbbek, mint a nekik megfelelő TTL áramkörök. Vegyük sorra ezeket az eszközöket.



3.2. ábra

A CMOS inverter

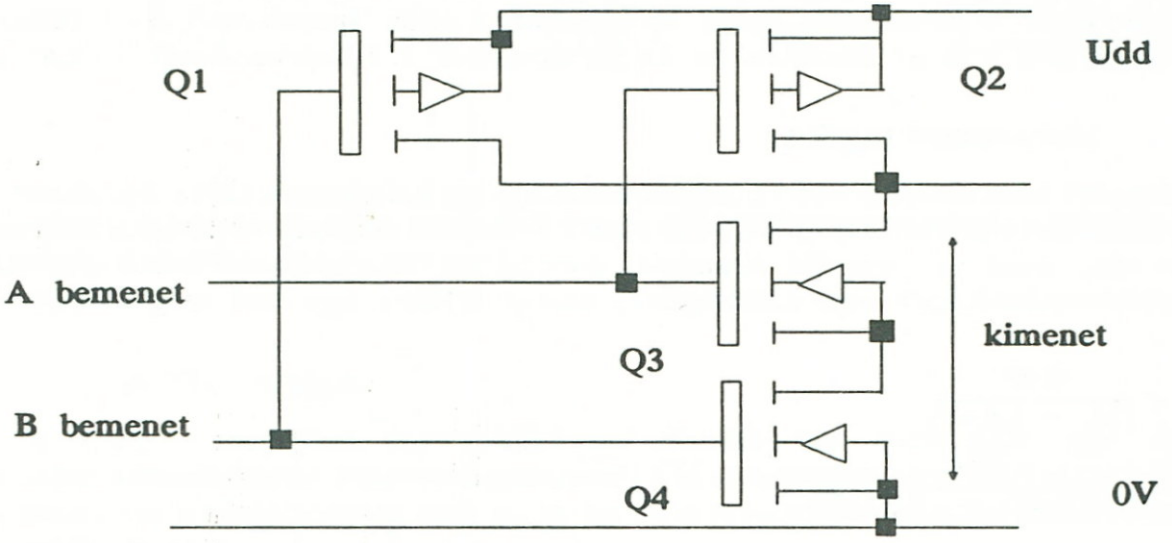
Az inverter.

A kapu (gate) bemenetre adott logikai 0 szint (föld potenciál) esetén a p csatornás MOS vezet,ellenállása kicsiny,mig az n csatornás párja zárt állapotban van.Ez azt jelenti,hogy a kimenet logikai 1 szintre kerül,a kimeneten a tápfeszültség jelenik meg.

A kapu bemenetre adott logikai 1 szint esetén a p csatornás MOS lezár,míg az n csatornás nyit. Ez azt jelenti, hogy a kimenet a föld potenciálra kerül,hiszen az n csatornás MOS tranzisztoron nem esik feszültség. A kapcsolásunk tehát inverterként működik.

CMOS NAND kapu

Működésének megértéséhez vegyük észre,hogy az ábrán felrajzolt kapcsolásban szereplő n csatornás MOS tranzisztorok párhuzamosan,míg a megfelelő p csatornások sorosan vannak kapcsolva.



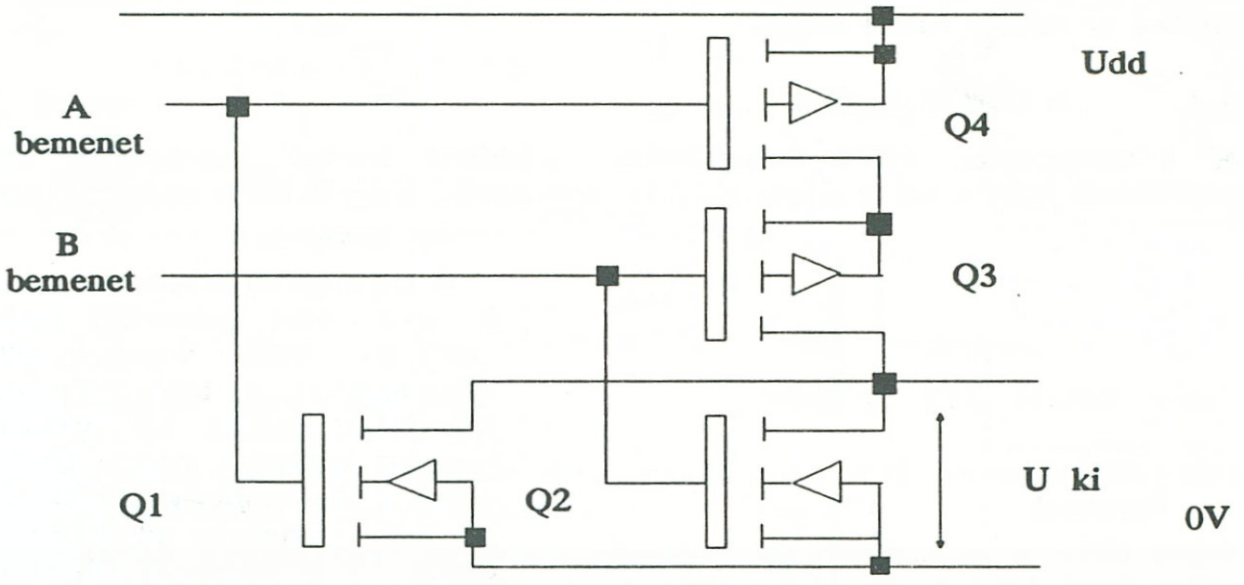
3.3. ábra

CMOS NAND kapu

Vizsgáljuk meg a kapu működését.

A	B	Q1	Q2	Q3	Q4	Kimenet
0	0	nyit	nyit	zár	zár	1
0	1	nyit	zár	zár	nyit	1
1	0	zár	nyit	nyit	zár	1
1	1	zár	zár	nyit	nyit	0

Megállapíthatjuk a táblázat alapján ,hogy valóban NANd kapuval van dolgunk.



3.4. ábra

CMOS NOR kapu

A CMOS NOR áramkör felépítése nagyon hasonló az előző kapcsoláshoz,ugyanúgy

megtaláljuk a sorosan és párhuzamosan kapcsolt tranzisztorokat, csak ezek típusa változott meg. A működés magyarázata a következő táblázat alapján már egyszerű.

A	B	Q1	Q2	Q3	Q4	Kimenet
0	0	zár	zár	nyit	nyit	1
0	1	zár	nyit	nyit	zár	0
1	0	nyit	zár	zár	nyit	0
1	1	nyit	nyit	zár	zár	0

Ezzel kialakítottuk a logikai áramkörcsaládok alapelemeit.Természetesen ezeken kívül a gyártó cégek igen sokféle logikai kaput készítenek az egyszerűbb alkalmazás végett. De mint a logikai függvények előállításánál megismertük, lényegében ezekből az alap-kapcsolásokból épülnek fel.

Manapság a számítógépek belsejében két áramkörcsalád elemeit találjuk meg, a TTL és CMOS elemeket.Ezeket az áramköri elemeket egy Si lapkára integrálják és így hozzák őket forgalomba. A felhasználásukhoz fontos tudnunk néhány fogalmat, amelyet most a teljesség igénye nélkül mutatunk be.

Digitális IC paraméterek

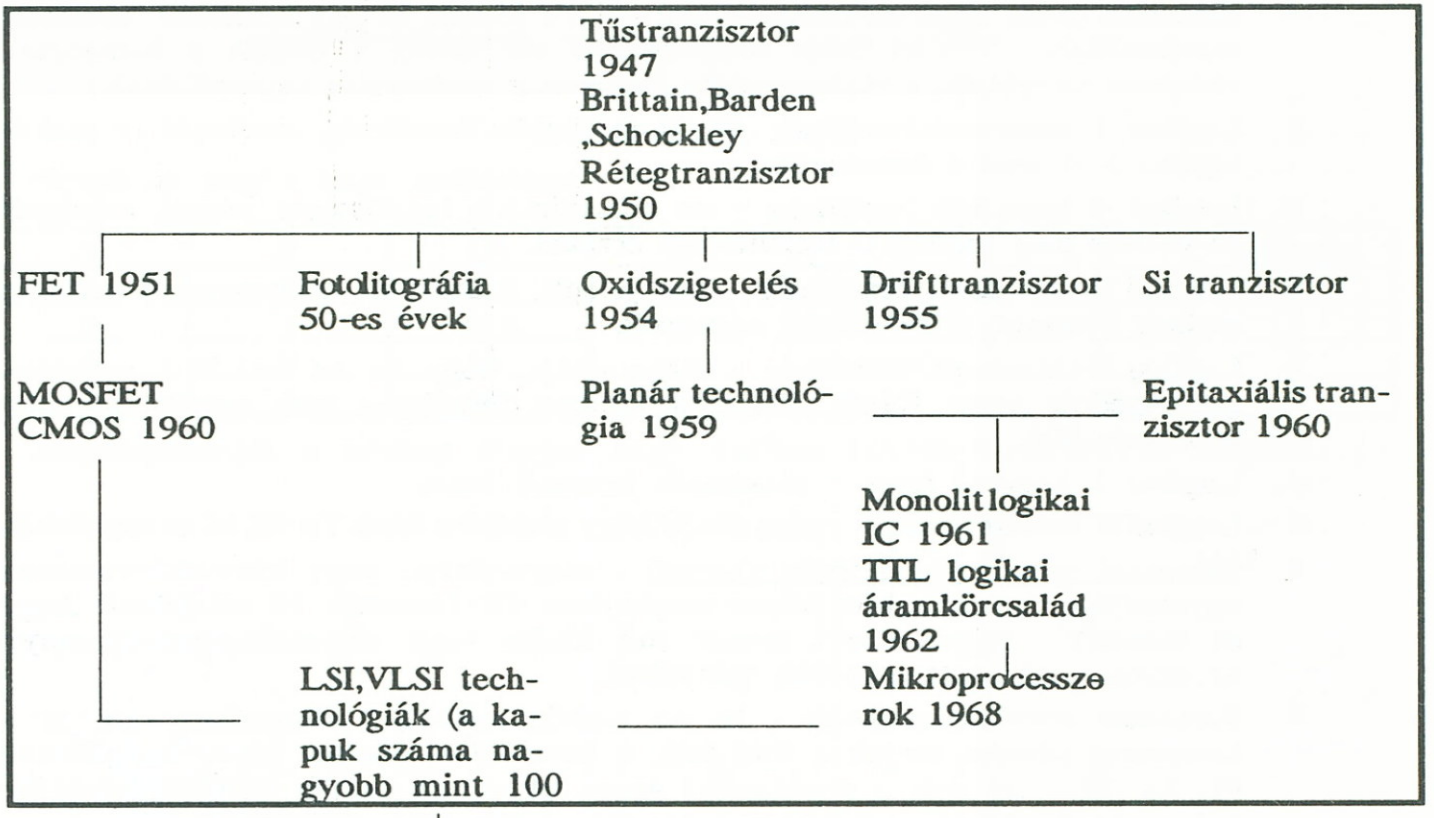
- Bemeneti dióda vágófeszültsége - max —1,5V jelenti, hogy a bemeneti feszültség maximálisan 1,5 V-al eshet földpotenciál alá. (Ezek a diódák a bemenetek védelmét szolgálják, a többemitteres tranzisztor emittereire kapcsolódnak.)
- Logikai 1 bemeneti feszültség - az a minimális feszültség, amelynél az eszköz logikai 1-et érez a bemenetén.
- Logikai 0 bemeneti feszültség - azt a maximális feszültséget jelenti, amelynél az eszköz még logikai 0 feszültséget érzékel.
- Logikai 1 kimeneti feszültség - azt jelenti, amely alá megengedett terhelés mellett a kimeneti feszültség nem esik.
- Logikai 0 kimeneti feszültség - megmutatja, hogy ha az eszköz kimenetébe megengedett áram folyik, akkor a kimenő feszültség nem emelkedik ezen feszültség fölé.
- Logikai 1 bemenő áram - maximális bemenő áram.
- Logikai 0 bemenő áram - megmutatja,hogy mekkora áram folyik ki az eszközből.
- Kimeneti egységterhelhetőség (fanout) - megmutatja, hogy hány,elektromosan ugyanolyan típusú eszközt képes meghajtani. Pl.: fanout = 10 azt jelenti ,hogy az áramkör tízszer annyi áramot tud kiadni vagy elnyelni, mint amennyit az egyes meghajtott eszközök igényelnek.
- Kimeneti rövidzárási áram - ha az eszköz logikai 1 állapotban van, és a kimenetet rövidre zárjuk a föld felé, a kapu adott áramot képes szolgáltatni. Pl.: ha 20 - 55 mA a rövidzárási áram, akkor az eszköz rövidzárvédett.Hibakeresésnél előnyös lehet,hogy valamely eszköz bemenetét leföldeljük, de ez tönkre teheti az előtte lévő fokozatot.Ha rövidzárvédett az eszközünk, akkor ez megtehető.
- Tápáramfelvétel logiakai 1 állapotban
- Tápáramfelvétel logikai 0 állapotban.

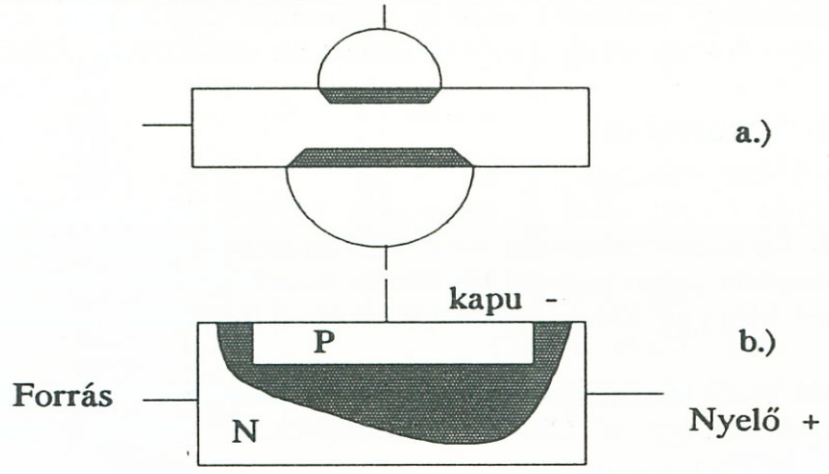
Ezen paraméterek ismerete nagyon fontos azok számára,akik egyszerűbb áramköröket kívánnak tervezni,illetve kivitelezni. Természetesen nem lehet általánosan megadni a sokféle kapu gyártási paraméterét, ezek megtalálhatók a legkülönfélébb katalógusokba. Például a leggyakoribb TTL áramkörök leírását Magyari Béla: Digitális IC-k című kötetében találhatjuk meg.

A továbbiakban összehasonlítjuk a különböző digitális integrált áramköri családok néhány jellemző adatát.A táblázatból kitűnik,hogy a CMOS áramköröknek nagyon kicsiny

	DTL	TTL	CMOS
Elemi függvény pozitív logika esetén	NAND	NAND	NOR
Tápfeszültség	5V	5V	10V
Kapunkéntiteljesítményfelvétel mW-okban	1 - 33	1 -10	0,8 mikroW/kHz
Késleltetési idő mikrosecundumban	30	10 -33	25
Kimenetiegységterhelhetőség	8	10	20

teljesítményfelvételük van. Felépítésük is egyszerü - integrált áramköri lapkán könnyen elkészíthető, ezért kitűnő építőköve az olyan nagybonyolultságú eszközöknek, mint a processzor és memória áramkörök. Az elektronikus alapáramkörök ismertetését lezárva tekintsük át nagyvonalakban a félvezető technika fejlődését.





3.5. ábra

Az „a" ábrán az első tranzisztor vázlata látható. A Ge kristályba két oldalról Indium golyót nyomtak be.

A „b" ábra az első FET - térvezérlésű tranzisztort mutatja. A satírozott terület a kiürített réteget mutatja. Ennek változtatásával lehetet befolyásolni a tranzisztoron átfolyó áramot.

4. A számítógépek logikai áramkörei

4.1. Az összeadó egységek

Az adatokkal végzett műveletek az aritmetikai logikai egységben (ALU) hajtódnak végre. A továbbiakban megvizsgáljuk, hogyan lehet olyan áramköröket tervezni, amelyek az alapvető műveletek elvégzésére képesek. A fejezetben feltételezzük a decimális és bináris számok egymásba alakításának ismeretét.

4.1.1. A félösszeadó egység

Az olyan áramköröket, amelyek ugyanazon bináris helyiértékek összegét képezik ($Ö_i$) és kialakítják a szükséges esetben az átvitelt a magasabb helyiértékek felé (C_i),de nem veszik figyelembe az előző helyiértéken képződött átvitelt, **félösszeadó** áramköröknek nevezzük. Az ilyen áramkörök kialakításához, mint a logikai tervezés alapfeladatánál láttuk fel kell írni először a művelet igazságtáblázatát, majd ki kell alakítanunk a függvény algebrai alakját, egyszerűsíteni kell azt és végül fel lehet rajzolni a kívánt kapcsolás logikai vázlatát. Haladjunk sorban ezeken a lépéseken!

X_i	Y_i	$Ö_i$	C_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

4.1. táblázat

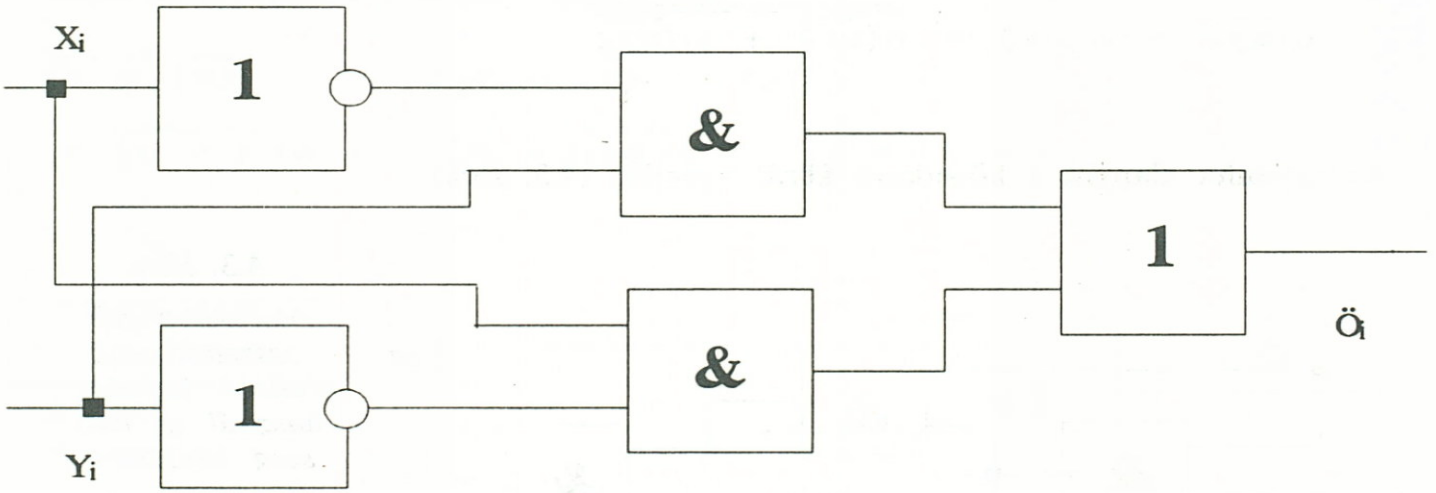
Félösszeadóigazságtáblázata

A bináris összeadást lényegében ugyanúgy végezzük, mint a decimális számok körében. Ha lehetséges a számokat az adott helyiértéken összegezzük,

(Pl: 1 + 0 = 1) ha nem, képezzük az átvitelt a magasabb helyiértékek felé. (Pl: 1 + 1 = 0 és 1-et átvisszük a magasabb helyiértékre.) Ezt ugyanúgy tesszük, mint a decimális számok esetén.

(Pl: 6 + 7 = 13 azaz, van 3 egyesünk és egy tízes, azaz 1 átvitel a magasabb helyiérték, a tízesek felé.)

Írjuk fel most a táblázat alapján a függvény algebrai alakját! A diszjunktív normál alakot felhasználva kapjuk az összegre: $Ö_i = x_i y_i + x_i y_i$,az átvitelre $C_i = x_i y_i$ .Az



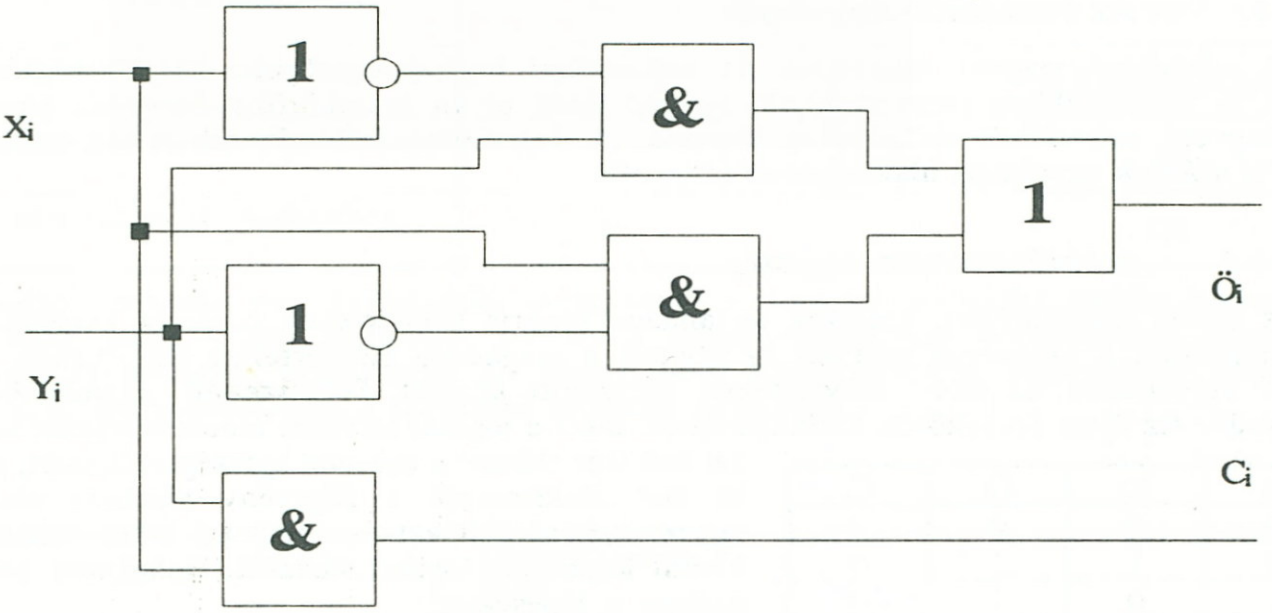
4.1. ábra

Félösszeadó összegképző áramkörének logikai vázlata

összeg alakjából látható, hogy az nem más, mint a bemeneti változók kizáró vagy kapcsolata. Ezt a továbbiakban $x_i \neq y_i$ módon fogjuk jelölni.

A függvény nem túl bonyolult, tovább egyszerűsíteni már nem tudjuk, ezért rajzoljuk fel az áramköri tervet.

A keletkezett átvitel kialakítása sem túl bonyolult feladat, hiszen csak a bemeneti



4.2. ábra

A félösszeadó áramkör logikai vázlata

változók ÉS kapcsolatát kell kialakítanunk. Ezt hozzáfűzve az előző ábrához kaphatjuk az áramkör szerkezetét. (4.2 ábra)

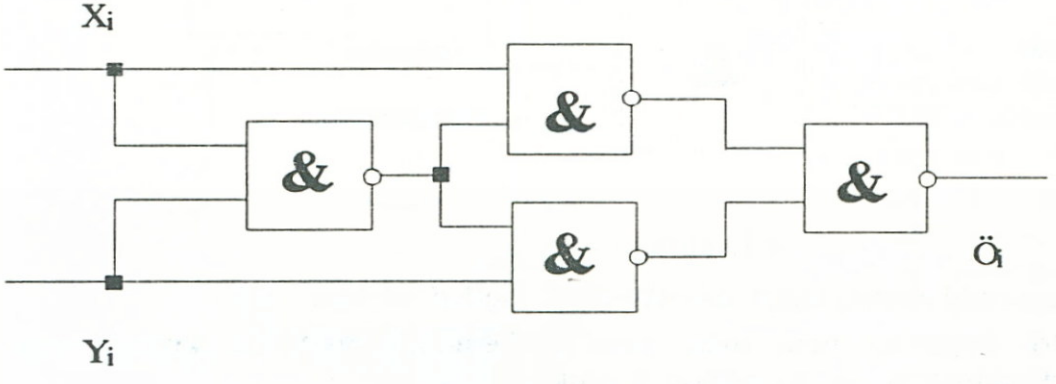
Sok esetben előfordulhat, hogy egyfajta kapu segítségével kell kialakítanunk az áramkörünket. Próbáljuk meg mi is ezt megtenni, alakítsuk át úgy a fent szereplő kapcsolást,hogy benne csak NAND kapuk szerepeljenek. Természetesen ehhez a logikai függvényt kell átalakítanunk a megfelelő alakra.

$$\overline{x_i}y_i + x_i\overline{y_i} = \overline{x_i}x_i + \overline{x_i}y_i + x_i\overline{y_i} + y_i\overline{y_i} = x_i(\overline{x_i} + \overline{y_i}) + x_i(\overline{x_i} + \overline{y_i})$$

$$= x_i\,\overline{\overline{(\overline{x_i} + \overline{y_i})}} + x_i\,\overline{\overline{(y_i + \overline{y_i})}} = x_i\,\overline{\overline{x_i}\,\overline{y_i}} + y_i\,\overline{\overline{x_i}\,\overline{y_i}} =$$

$$= x_i(\overline{x_i y_i}) + y_i(\overline{x_i y_i}) = \overline{x_i(\overline{x_i y_i}) + y_i(\overline{x_i y_i})}$$

Az egyenlet alapján a következő ábrát nyerjük. (4.3. ábra)



4.3. ábra

Átalakított félösszeadó csak NANd kapukat használ az összeg képzésére.

4.1.2. A teljes összeadó áramkör

A félösszeadó egységet elvben csak a legkisebb helyiértréken tudjuk felhasználni, hiszen nem veszi figyelembe az előző fokozatok átvitelét. A későbbiek során azonban azt is látni fogjuk, hogy itt sem alkalmazzuk, ha áramkörünket kivonásra is fel kívánjuk használni. Ezért olyan áramkört kell tervezni, amely a fenti problémát is megoldja, azaz figyelembe veszi az előző fokozatok átvitelét is. Erre szolgál a **teljes összeadó** egység!

A diszjunktív normál alakok felhasználásával megkaphatjuk a logikai függvényeket.

X_i	Y_i	C_{i-1}	$Ö_i$	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

4.2. táblázat

A teljes összeadó igazságtáblázata

$$Ö_i = \overline{x_i}\,\overline{y_i}\,c_{i-1} + \overline{x_i}\,y_i\,\overline{c_{i-1}} + x_i\,\overline{y_i}\,\overline{c_{i-1}} + x_i y_i c_{i-1}$$

$$Ö_i = (\overline{x_i}\,\overline{y_i} + x_i y_i)\,c_{i-1} + (\overline{x_i}\,y_i + x_i\,\overline{y_i})\,\overline{c_{i-1}} \quad \text{adódik.}$$

Az egyenlet első tagjának zárójelben foglalt részét tovább alakítjuk. A második tag első tényezőjéről viszont azonnal látható, hogy az egy kizáró vagy kapcsolat,így ezzel további tennivalónk nincs.

Az átviteli függvényt átalakítva nyerjük,hogy:

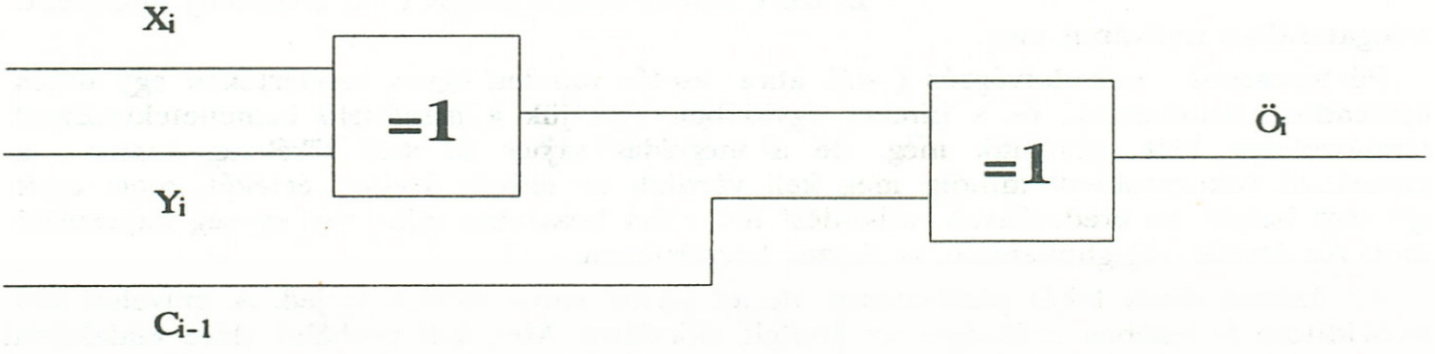
$$= \overline{\overline{x_i}\,\overline{y_i} + x_i y_i} = \overline{\overline{x_i}\,\overline{y_i}}\;\overline{x_i y_i} = \overline{((x_i + y_i)\,(\overline{x_i} + \overline{y_i})} =$$

$$= \overline{(x_i\overline{x_i} + y_i\overline{x_i} + x_i\overline{y_i} + y_i\overline{y_i})} = \overline{(\overline{x_i}\,y_i + x_i\,\overline{y_i})} = \overline{x_i \neq y_i}$$

Ezzel az egyenlet a következő alakot ölti:

$$Ö_i = (\overline{x_i}\,\overline{y_i} + x_i y_i)\,c_{i-1} + (\overline{x_i}\,y_i + x_i y_i)\,c_{i-1} =$$

$$= (\overline{x_i \neq y_i})\,c_{i-1} + (x_i \neq y_i)\,\overline{c_{i-1}} = c_{i-1} \neq (x_i \neq y_i)$$



4.4. ábra

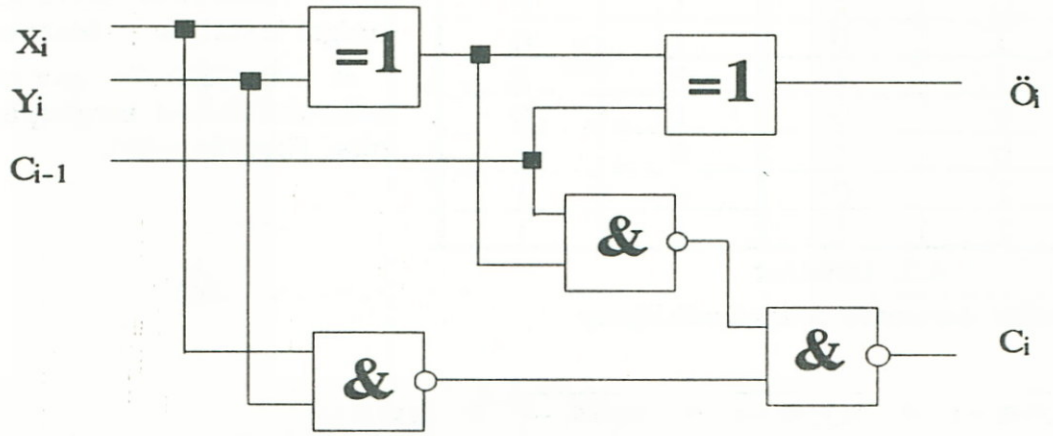
A teljes összeadó áramkör összegképzésének logikai rajza.

$$c_i = \overline{x_i} y_i c_{i-1} + x_i \overline{y_i} c_{i-1} + x_i y_i \overline{c_{i-1}} + x_i y_i c_{i-1} =$$

$$= c_{i-1}(\overline{x_i} y_i + x_i \overline{y_i}) + x_i y_i (c_{i-1} + \overline{c_{i-1}}) = c_{i-1}(x_i \neq y_i) + x_i y_i =$$

$$= \overline{\overline{c_{i-1}(x_i \neq y_i) + x_i y_i}} = \overline{\overline{c_{i-1}(x_i \neq y_i)}\ \overline{(x_i y_i)}}$$

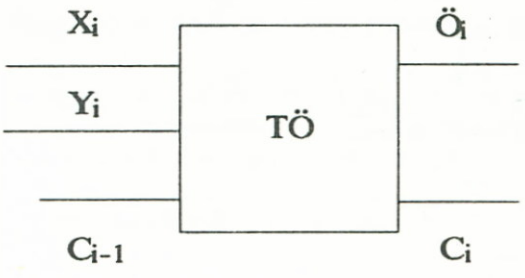
Ezzel az áramkör teljes kapcsolási rajza:



4.5. ábra

A teljes összeadó egység logikai kapcsolási vázlata.

A fent megrajzolt áramkör természetesen egy helyiértéken, azaz egy biten képes elvégezni az összeadást.Nagyobb szóhosszúság esetén kétféle módon járhatunk el. Az összeadás soros vagy párhuzamos módját választhatjuk.



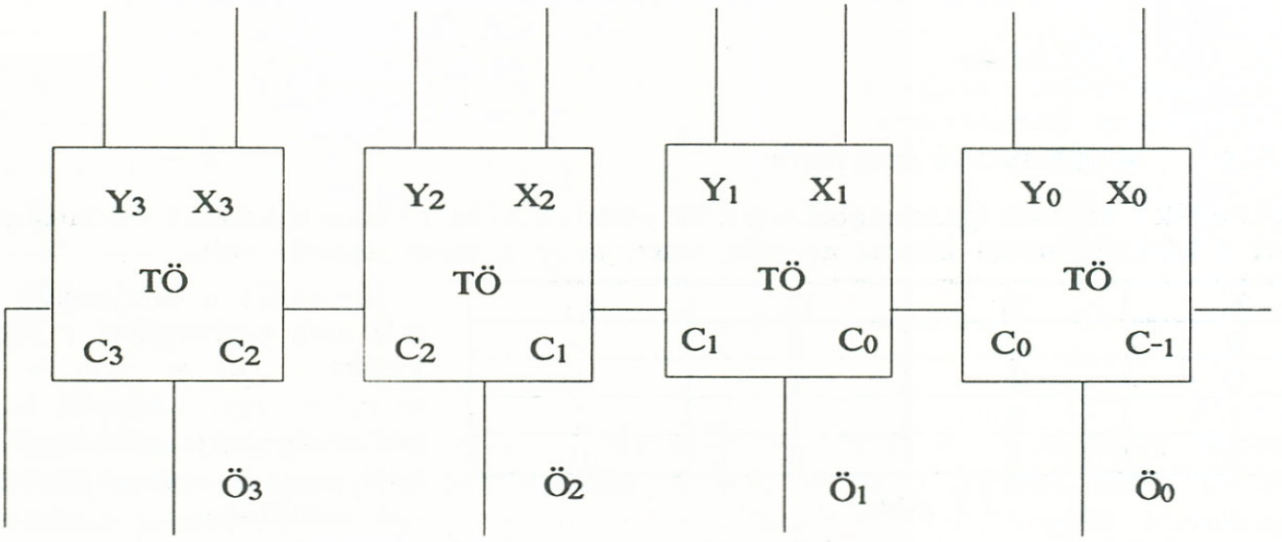
4.6. ábra

A teljes összeadó jelölése

Soros összeadás esetén a biteket sorban az összeadó áramkör bemenetére vezetjük. A keletkezett átvitelt megőrizzük és továbbadjuk a következő helyiértéken érkező biteknek. Az eredményt bitenként vezetjük ki az összegképző áramkörből. Természetesen ehhez egy speciális eszközre van szükségünk,amely képes a bitek léptetésére. Ezek az úgynevezett léptetőtárolók, amellyekkel a későbbiek során meg fogunk ismerkedni. Ezzel megspóroltunk egy sor összeadó elemet, de jelentős időt kell fizetnünk ezért az árért, ami az összeadandók és az eredmény bitenkénti tologatásában nyilvánul meg.

Párhuzamos müveletvégzés (4.7. ábra) esetén minden egyes helyiértéken egy teljes összeadót alkalmazunk, és a biteket egyidőben vezetjük a megfelelő bemenetekre.Ezzel természetesen időt spórolunk meg, de a megoldás sajnos itt sem tökéletes, hiszen a következő fokozatoknak mindig meg kell várniuk az előzők átviteli értékét, mert csak így lesz helyes az eredmény.A működési idő ezért hosszabb, mint egy egység kapcsolási ideje.Az átvitel végighullámzik az összes helyiértéken.

Az üzemmódunk tehát párhuzamos, de az átvitel soros módon terjed. A műveleti idő lerövidítése érdekében szükséges az átvitelt előrelátni. Meg kell próbálni előre kialakítani az átviteleket, a bemeneti értékek alapján. Ez az **átvitel előrelátási** technika. (Paralel carry lookahead) Mivel az átvitel értéke az adott helyiértéken az összes előtte álló változótól függ, ennek kialakítása egyáltalán nem lehetetlen.



4.7. ábra

A párhuzámos összeadó egység elvi logikai vázlata.

Az átvitelről tudjuk, hogy $C_i = x_i y_i + c_{i-1}(x_i \neq y_i)$. Legyen $g_i = x_i y_i$ az átvitelt **generáló függvény**. Ugyanis ha x_i és y_i egyszerre igaz, akkor az átvitel mindig generálódik. Továbbá $p_i = (x_i \neq y_i)$ az átvitelt **propagáló függvény**,mert értéke 1, ha c_{i-1} és valamelyik bemeneti változó igaz. Ha x_i vagy y_i értéke 1, akkor átviszi, továbbadja a c_{i-1} értékét a következő fokozatnak.

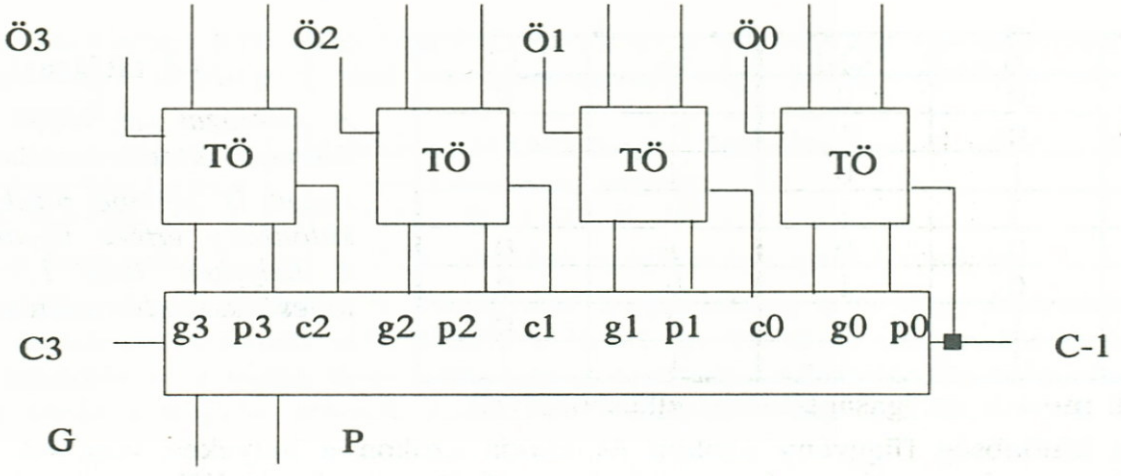
Egyenletünk tehát a következő alakban írható fel:

$c = g_i + c_{i-1}p_i$.

Fejezzük ki rekurzív módon a soron következő átviteleket.

Látható, hogy a fenti összefüggés g és p segédváltozókból megkapható. Az áramkör nem nagyon bonyolult,mert kellő számú ÉS és VAGY kapuból előállítható.

$$c_0 = g_0 + c_{-1}p_0$$
$$c_1 = g_1 + c_0p_1 = g_1 + (g_0 + c_{-1}p_0)p_1 = g_1 + g_0p_1 + c_{-1}p_0p_1$$
$$c_2 = g_2 + c_1g_2 = g_2 + g_1p_2 + g_0p_1p_2 + c_{-1}p_0p_1p_0$$
$$c_3 = g_3 + c_2p_3 = g_3 + g_2p_3 + g_1p_2p_3 + g_0p_1p_2p_3 + c_{-1}p_0p_1p_2p_3 .$$



4.8. ábra

A továbbfejlesztett teljesösszeadó áramkör. G és P a további fokozatokhoz tartozó generáló és propagáló függvények.

4.2. A kivonás

4.2.1. A félkivonó áramkör

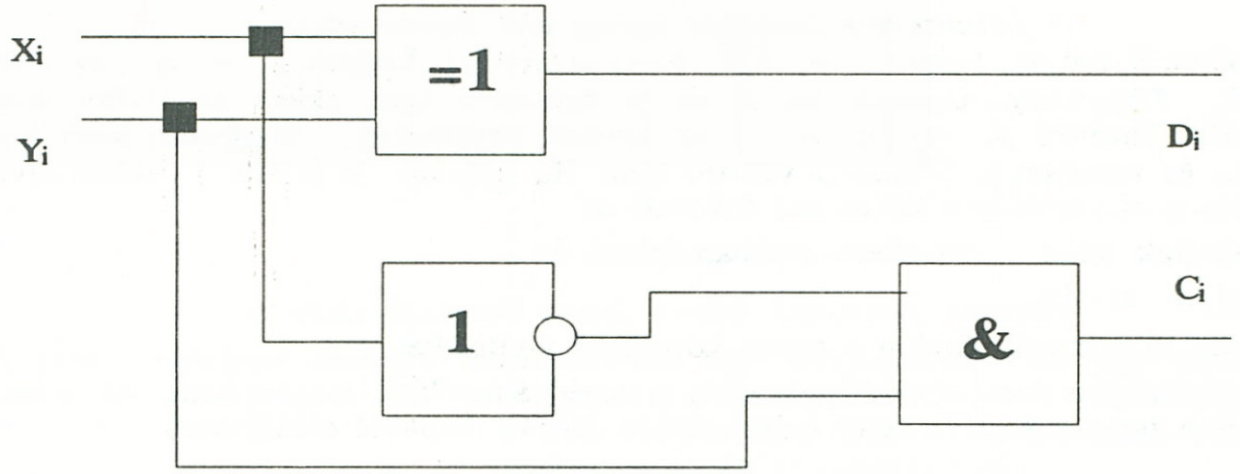
Tekintsük a kivonás igazságtáblázatát! Di jelenti a Xi és Yi bináris számok különbségét,Ci jelenti a kivonás során kapott átvitelt, azaz, hogy a szám negatív volt.

X_i	Y_i	D_i	C_i
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

4.3. táblázat

Félkivonó áramkör igazságtáblázata.

Képezzük a diszjunktív normál alak segitségével a függvényeket: $d_i = \bar{x}_i y_i + x_i \bar{y}_i$ és $c_i = \bar{x}_i y_i$ .Látható, hogy a különbség alakja az összeadásnak felel meg, és csak az átvitelben van különbség.



4.9. ábra

A félkivonó áramkör logikai vázlata.

4.2.2. A teljes kivonó áramkör

X_i	Y_i	C_{i-1}	D_i	C_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

4.4. táblázat

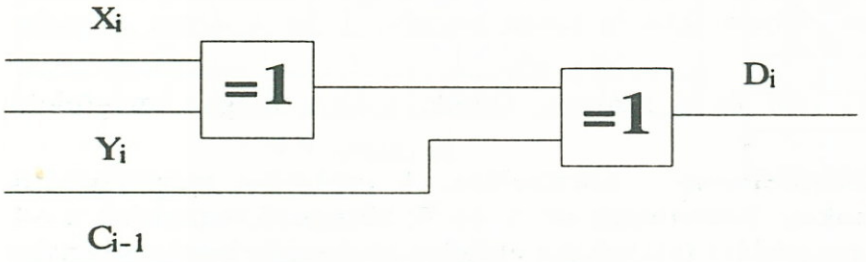
A táblázat a teljes kivonó áramkör adatait tartalmazza.

Innen is szembe tűnik,hogy a különbség értéke ugyanazokon a helyeken lesz 1, mint a teljes összeadás esetén.

Kezdjük most is az igaságtáblázat elkészítésével!

Mivel a különbség függvény azokon és csakis azokon a helyeken vesz fel 1 logikai értéket, ahol az összeadás, ezért ennek levezetését nem ismételjük meg,csak a végső egyenletet közöljük $d_i = c_{i-1} \neq (x_i \neq y_i)$ Természetesen a kivonás logikai áramköre is megegyezik az összeadásnál találttal.

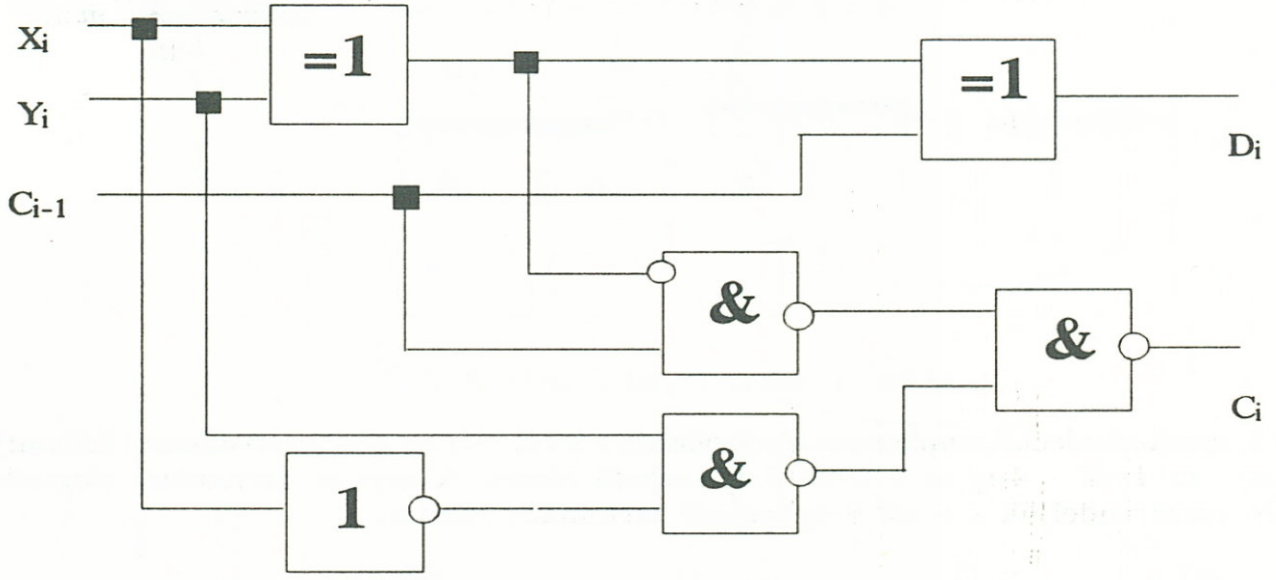
A kivonás átvitelére kapott alakot szükséges csupán közelebbről szemügyre venni.



4.10. ábra

A teljeskivonó áramkör különbségképző részegysége

Az átvitelre az $c_i = \overline{x_i}\,\overline{y_i}\,c_{i-1} + \overline{x_i}\,y_i\,\overline{c_{i-1}} + x_i\,\overline{y_i}\,c_{i-1} + x_i\,y_i\,c_{i-1}$ egyenletet kapjuk.Az átvitel megállapításához emeljük ki $\overline{x_i}\,y_i$ és c_{i-1} tagokat. Ezzel az alábbi egyenlethez jutunk. $\overline{x_i}\,y_i\,(\overline{c_{i-1}} + c_{i-1}) + c_{i-1}\,(\overline{x_i}\,\overline{y_i} + x_i\,y_i)$ .Vegyük észre,hogy az



4.11. ábra

A teljes kivonó áramkör logikai rajza

első tag zárójelben lévő kifejezése 1 és azt, hogy a második tag zárójelbe tett kifejezése egy kizáró vagy negáltja (lásd az összeadásnál).Így az egyenletünkre a következő formulát nyerjük :

$\overline{x_i\,y_i + c_{i-1}\,(x_i \neq y_i)}$,átalakitva NAND kapuk felhasználásával adódik, hogy

$$\overline{\overline{x_i\,y_i + c_{i-1}\,(x_i \neq y_i)}} = \overline{\overline{x_i\,y_i}\;\overline{c_{i-1}\,(x_i \neq y_i)}}$$

Ebből a kapcsolási rajzot a 4.11. ábrának megfelelően alakíthatjuk ki.

Mint a függvények felírása és átalakítása során láttuk, igen nagy hasonlóság van az összeadó és kivonó áramkör működésében. Ezért merül fel az igénye annak, hogy talán egyetlen áramkör is el tudná látni a kivonás és összeadás müveletét. Ez valóban lehetséges, de ennek során a negatív számok kezelésére vonatkozó ismereteinket át kell alakítani.

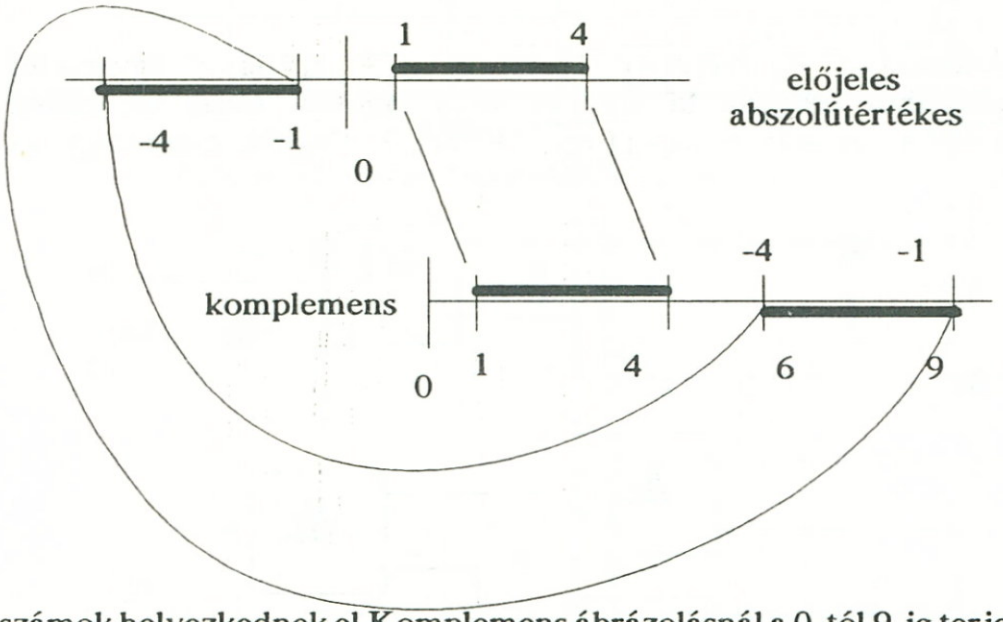
Például: decimálisan $N_{10} = -21$ binárisan $N_2 = 1\ 10101$

Előjel Abszolútérték

A negatív szám ábrázolására két lehetőségünk van.Az első az **előjeles abszolútértékes** ábrázolás, amelynél egy előjel és a szám abszolútértéke határozza meg a számot.Kihasználtuk a bináris ábrázolásnál, hogy a szám előjele 0,ha a szám pozitív, 1 ha a szám negatív.

Az ilyen megoldásnál a műveleti kód és a számok előjele szabja meg a megfelelő műveleti szabály kiválsztását.

A másik lehetőségünk a szám **komplemens ábrázolása.** A probléma megvilágítása céljából használjunk egyjegyű számokat. Szeretnénk az 1 és 4 közé eső, valamint a -4 és -1 közé eső számok ábrázolására megoldást találni. Az előjeles ábrázolásra a szokványos számegyenest használhatjuk , a nullától balra a negatív, tőle jobbra pedig a pozitív



4.12. ábra

A negatív számok ábrázolásaelőjeles abszolútértékes és komplemens módban.

számok helyezkednek el.Komplemens ábrázolásnál a 0-tól 9-ig terjedő intervellumot felbontjuk két részre, az 1-től - 4-ig és a 6-tól 9-ig terjedő részre. A negatív tartomány elemeihez (-4, -10) hozzárendeljük a 6-tól 9-ig terjedő tartomány elemeit.

Ezek után nézzük meg, hogyan vezethető vissza a kivonás összeadásra. Legyenek A_n és B_n n helyiértékes bináris számok. Ekkor ezek különbsége $D= A_n - B_n$,amit úgy is felírhatunk,hogy $D = A_n + (-B_n) = A_n + (C - B_n) - C$,ahol $C = 2^n$ vagy $C = 2^n - 1$ alakú pozitív szám. A C megfelelő választásával a kettes illetve egyes komplemens kódhoz jutunk.Jelölje a $B_n^{(2)} = 2^n - B_n$; $B_n^{(1)} = (2^n - 1) - B_n$ a B_n -nek a komplemenseit $B_n^{(1)} = (2^n - 1) + B_n^{(2)} - 2^n = B_n^{(2)} - 1$;A B_n egyes és kettes komplemense közötti kapcsolat kialakításánál felhasználtuk, hogy $B_n^{(2)} - 2^n = -B_n$. A különbségre kialakított egyenletünk tehát a következő:

$$D = A_n + ((2^n - 1) - B_n) - (2^n - 1) = A_n + B_n^{(1)} - 2^n + 1$$
$$= A_n + B_n^{(2)} - 2^n.$$

Ha meggondoljuk, hogy a $2^n - 1 - B_n$ bitenkénti komplementálást jelent, a 2^n kivonása pedig a túlcsordulási hely negálását, akkor készen is vagyunk.A megoldás tehát az, hogy az összeadó áramkört minden biten, az elsőt is beleértve, teljesösszeadóból építjük meg és hozzáveszünk egy minden biten negálást végző áramkört.

	1	1	1	1	$2^4 - 1$ dec. 15
–	0	1	0	1	B_n dec. 5
	1	0	1	0	

4.5. táblázat

Példa arra,hogy a $2^n - 1 - B_n$ bitenkénti komplementálást jelent.

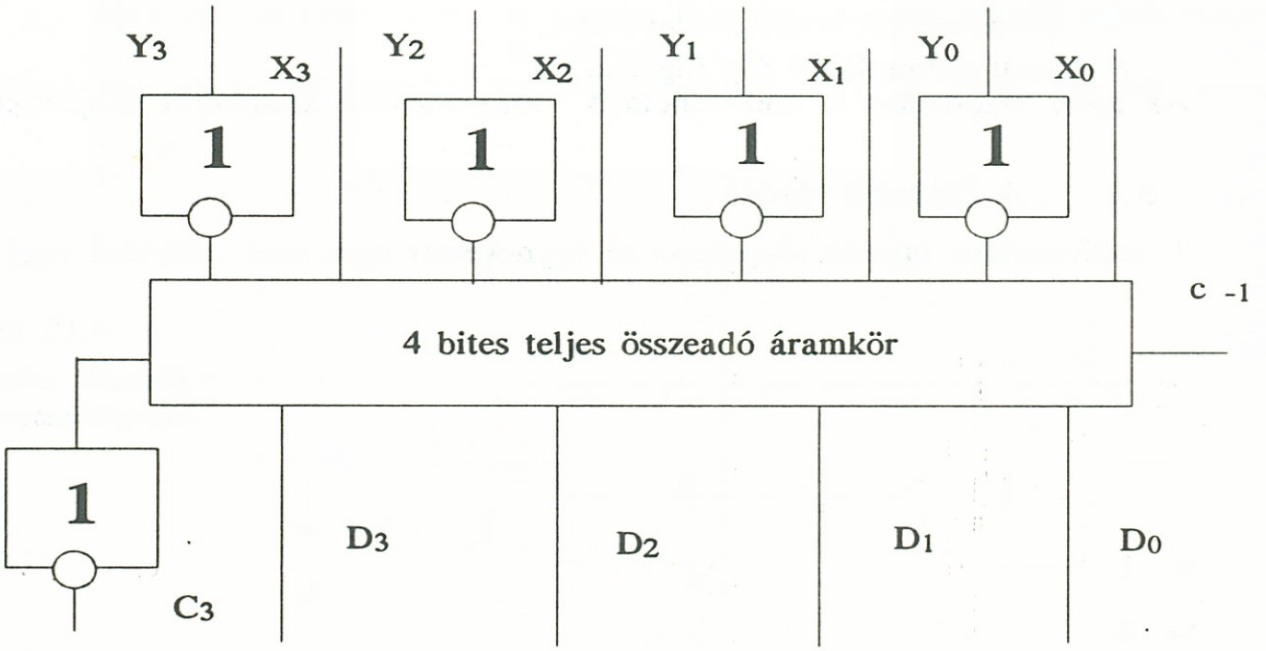
túlcsordulási hely	2^3	2^2	2^1	2^0
0	1	0	1	0
1	0	0	0	0
1	1	0	1	0

4.6. táblázat

Példa arra, hogy a 2^n kivonása a túlcsordulási hely negálását jelenti

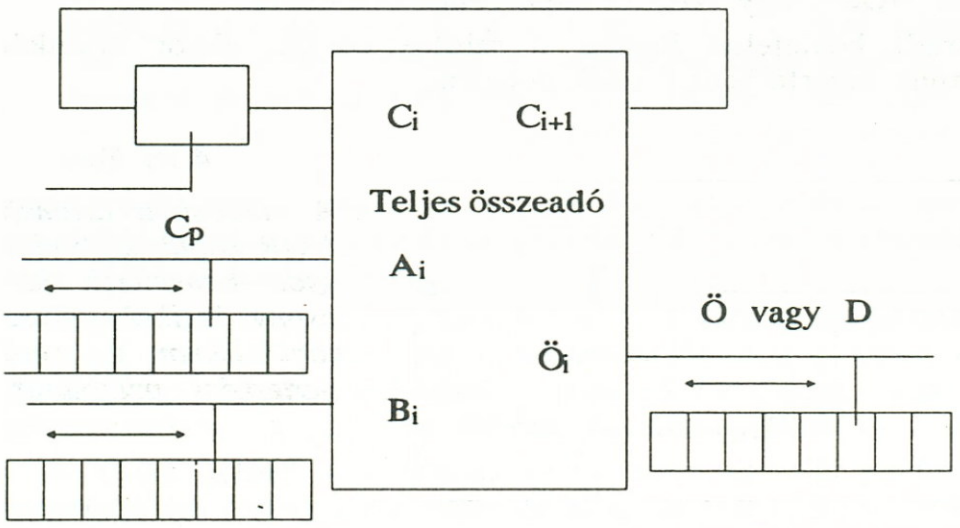
A kivonást ezek után úgy végezzük, hogy a 0 -dik bitre 1 átvitelt adunk (1 hozzáadása a különbség képzésénél), komplementáljuk a kivonandót, elvégezzük az összeadást,majd képezzük a túlcsordulási hely negáltját.

Innen már egyszerűen elkészíthetjük az áramköri tervet.



4.13. ábra

Párhuzamos összeadó és kivonó áramkör logikai vázlata.



4.14. ábra

Soros műveletvégző egység. Legyen a B léptethető regiszter képes a bitenkénti negációra. Cp beállítható 1 -be a kivonás kezdetén. illetve onnan elvihető negálva a műveletsor végén. A beállítások függvényeként kivonást vagy összeadást végezhetünk.A műveleti idő hosszúsága miatt (léptetések) ma már ezt az áramkört ritkán alkalmazzák.

4.3. Az elemi tároló áramkörök

Mint a két alapművelet végrehajtása során láttuk, hogy az összeadó áramkörünk csak abban az esetben működik helyesen, ha a megfelelő időben rendelkezésére állnak a szükséges adatok.Hogy ez ne jelentsen problémát, szükségünk van olyan eszközökre, amelyek a megfelelő adatokat tárolják, esetleg még léptetik is. (Pl:soros összeadó áramkör,soros párhuzamos átalakítás stb.)

Tároló áramköröknek azokat az eszközöket nevezzük, amelyek képesek a beléjük táplált információt tetszés szerinti ideig megőrizni. Az ilyen jellegű feladatot a visszacsatolt áramkörök látják el. Az áramkör kimenetén megjelenő jelet fázishelyesen visszavezetjük az eszköz bemenetére (pozitív visszacsatolás). Ezzel a megoldással elérjük,hogy a kimenő jel értéke

- » a bemenő jel értékétől,
- » a kimenő jel korábbi állapotától,
- » ezek sorrendjétől fog függeni.

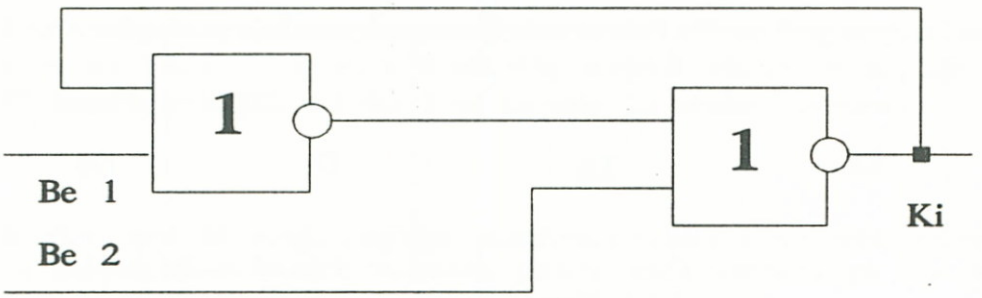
Az ilyen eszközöket a **szekvenciális** vagy sorrendi áramkörök csoportjába soroljuk.

4.3.1. A bistabil tároló

A szekvenciális tárolók alapeleme az úgynevezett együtemű vezérlésű vagy **aszinkron**

4.15. ábra

Bistabil tároló NOR kapu felhasználásával

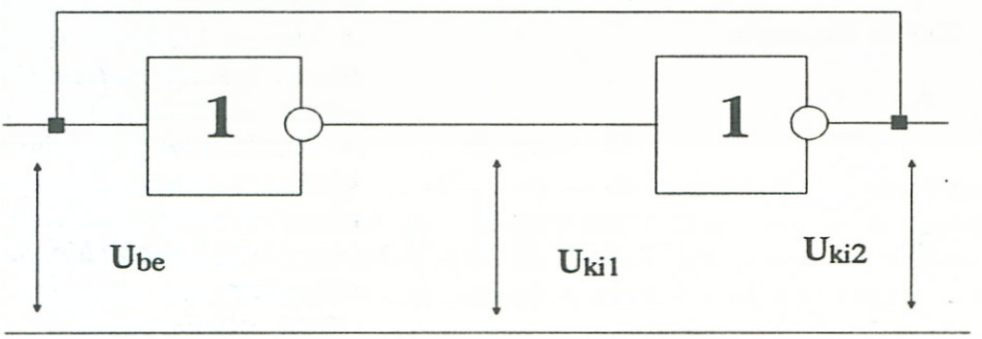


tároló.Az ilyen tárolót **bistabil tárolónak** is nevezzük.Könnyedén készíthetüink ilyen eszközt két kétbemenetű NOR ,vagy NAND kapu felhasználásával.

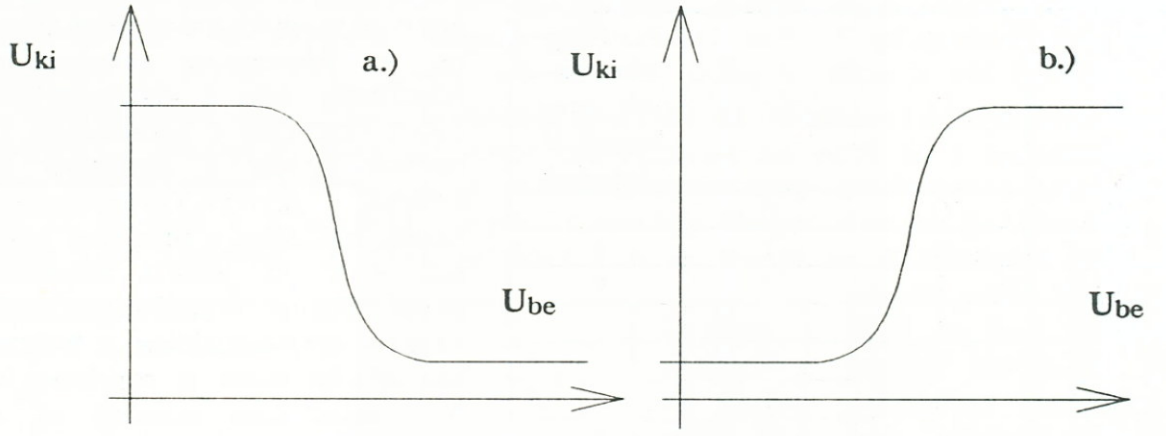
Ha a szabadon maradt bemeteket logikai 0 szintre kötjük, akkor áramkörünket átrajzolhatjuk,mert a kapuk inverterként fognak dolgozni.

4.16. ábra

A módosított áramkör. A NOR kapuk szabadon hagyott bemeneteit az alacsony logikai szintre kötöttük.Ekkor egyszerű inverterként működnek.



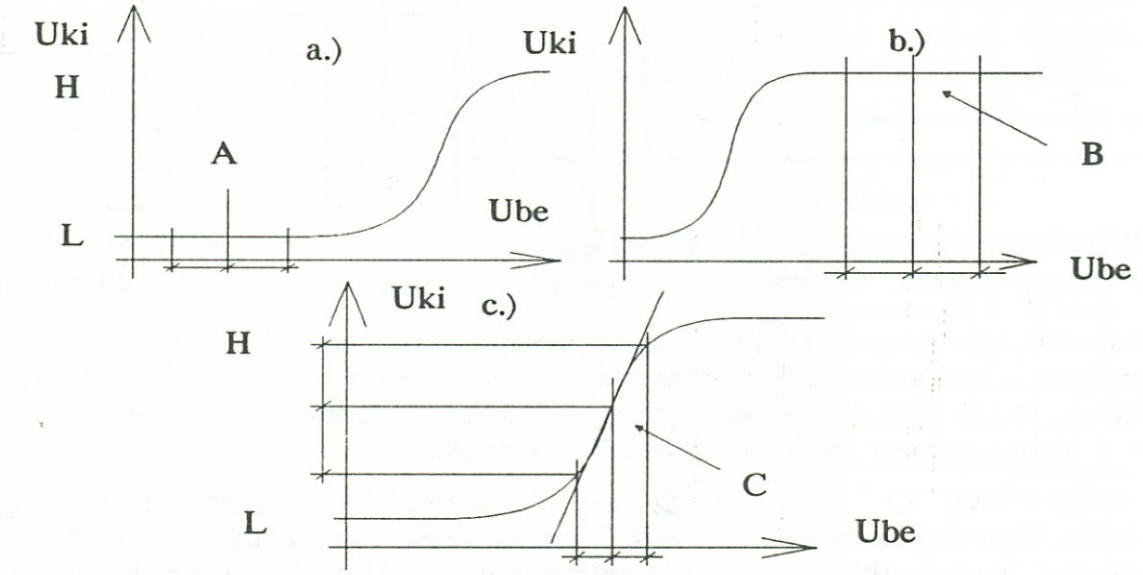
Ha a bemenetre H szintű jelet kapcsolunk,akkor a kétszeri invertálás miatt a kimenet is az, ezt a jelet vezetjük vissza a bemenetre. (A jel tehát fázishelyes lesz.)



4.17. ábra

Az a,) ábra egy inverter, a b.) két sorba kapcsolt inverter átviteli karakterisztikáját mutatja,

Emlékezzünk vissza az inverter karakterisztikájára. (4.17./a ábra) Ha két invertert



4.18. ábra

Az ábrán két sorba kapcsolt inverter karakterisztikáinak felnagyított részletei láthatók.Az a.) ábra az L, a b-n az L és H szint között, a c-n pedig H kimnetei szint esetén ábrázolja a feszültségváltozás hatását. Az a.) és b.) esetben a kimeneti feszültség nem, míg a c.) esetben erőteljesen megváltozik. A fázishhelyes (pozitív) visszacsatolás miatt az áramkör állapotot vált.

kapcsolunk össze, akkor a 4.17./b ábrának megfelelő karakterisztikát kapunk. Ennek megfelelően tehát az átviteli karakterisztikát három elkülönülő részre bonthatjuk.Vizsgáljuk meg ezeket a szakaszokat!

Tegyük fel ,hogy a rendszerünk a 4.18. ábra A-val jelzett pontjában tartózkodik.Nézzük meg mi történik akkor, ha a bemeneti feszültség kicsit megváltozik.Mivel a rendszer jelenleg egy platón tartózkodik, kimenő feszültsége nem változik meg a visszacsatolás következtében. (A platón a kimenő feszültség állandó.)

A C-vel jelzett állapotban, az előzővel egyező gondolatmenet alapján, ugyanezt a megállapítást tehetjük.Az áramkör állapota tehát itt is stabil marad.

A B-vel jelzett állapotban viszont a rendszer másként működik. Feltesszük,hogy az eszközünk karakterisztikája olyan,hogy a B állapothoz tartozó grafikon meredeksége 1-nél nagyobb. A feszültség pozitív értelmü megváltozásának hatására a kimeneten megnő a feszültség, ami a visszacsatolás miatt visszajut a bemenetre. Ez tovább növeli a kimeneti feszültség szintjét.Természetesen ennek hatására a rendszer gyorsan a C-vel jelzett állapotba kerül. Ha a feszültségváltozás negatív értelmű,akkor a helyzet hasonló,de most a rendszer az A állapot felé mozdul el, és szinte azonnal oda is kerül.

Ennek alapján tehát azt mondhatjuk, hogy a rendszernek két stabil állapota van, ezért az

R	S	Q	Q^*	A függvényeknek megfelelő állapotok
0	0	0	0	
0	0	0	1	*
0	0	1	0	*
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	*
0	1	1	1	
1	0	0	0	
1	0	0	1	*
1	0	1	0	
1	0	1	1	
1	1	0	0	*
1	1	0	1	
1	1	1	0	
1	1	1	1	

4.7. táblázat

*A bistabil áramkör igazságtáblázata.A táblában * jelöli a függvényeknek mengfelelő kombináci-*

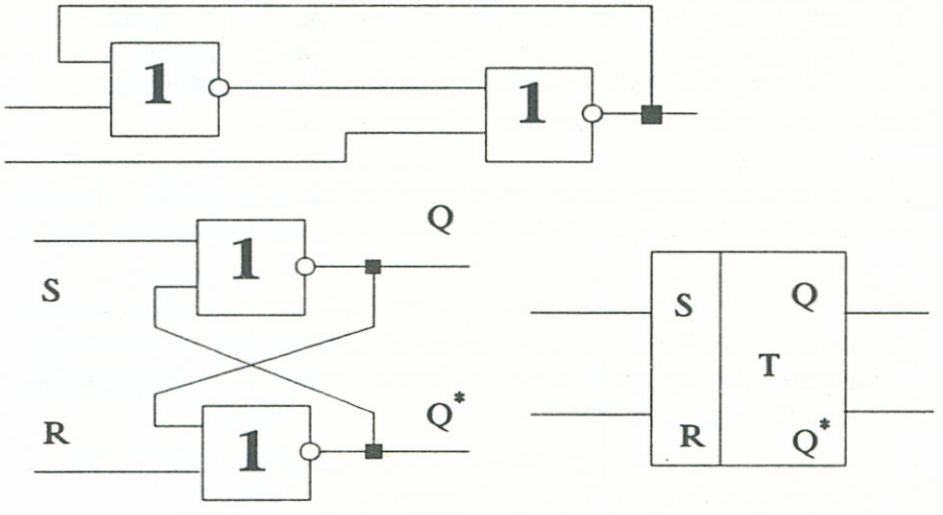
R	S	Q	Q*
0	0	0	1
0	0	1	0
0	1	1	0
1	0	0	1
1	1	0	0

4.8. táblázat

A tároló logikai állapotai

eszközt kétállapotú,vagy idegen szóval bistabil áramkörnek nevezhetjük.A továbbiakban nézzük meg, milyen a berendezésünk logikai működése. Ennek megértése céljából rajzoljuk át az ábránkat. Az 4.19. ábra szépen megmutatja az áramkör szimmentriáját,ezértkönnyen kialakíthatjuk a kimenetekhez tartozó logikai függvényeket is.

$Q = \overline{R + Q}^*$ és $Q^* = \overline{S + Q}$ .Az összefüggések alapján látható,hogy a kimenetekre felírt függvények értéke a másik kimenet logikai értékétől is függ. Készítsük most el a rendszer igazságtáblázatát és vizsgáljuk meg a változók milyen kombinációja felel meg a felírt összefüggéseknek. Mivel a kimenetek értéke a bemenetektől és a rendszer előző állapotától is függ, a táblában rendre szerepeltetni kell az R, S, Q, Q^*



4.19. ábra

A R-S tároló módosított logikai rajza és a tárolórajzjele.

változókat.A megfelelő sorok kiválasztását úgy tehetjuk meg, hogy vesszük először az egyik kimeneti függvényt, és megvizsgáljuk,hogy mely sorok esetén teljesül.Ezeket megtartjuk a táblában, a többit kihúzzuk belőle.Ugyanígy járunk el a másik függvény esetén is, csak most már, az előzőleg megmaradt sorokat kell vizsgálnunk.Ennek megfelelően a táblázatban öt olyan sort fogunk találni, ami kielégíti mind a két feltételt.

A feltételeknek eleget tevő kombinációkat az 4.8. táblázat foglalja össze.Ha közelebbről megvizsgájuk ezeket az eseteket,akkor látható,hogy ha az R és S bemenetek is 0 logikai szinten vannak,akkor a kimenentek értéke bármely állapotot felvehet, ami azt jelenti,hogy ebben az esetben a kimenetek megőrzik eredeti állapotukat.Az S=1 és R=0 esetén a Q kimenet 1-be állítódik, Q szempontjából S tehát beírja az információt (set), S=0 és R=1 esetén pedig törli azt (reset).Az R=1 és S=1 állapot logikailag ugyan helyes,de az áramkör működése szempontjából azonban tiltott,az eszköz tönkremenetelét okozza, egyszerre nem lehet mind a két kimenet azonos szinten.

Hasonló megoldást NAND kapukkal is kialakíthatunk.

A logikai függvények a következő alakot öltik:

$$Q = \overline{Q^* + R} \; ;$$

$$Q^* = \overline{Q + S} \; .$$

R	S	Q	Q^*	A függvényeknek megfelelő állapotok
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	*
0	1	0	0	
0	1	0	1	
0	1	1	0	*
0	1	1	1	
1	0	0	0	
1	0	0	1	*
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	*
1	1	1	0	*
1	1	1	1	

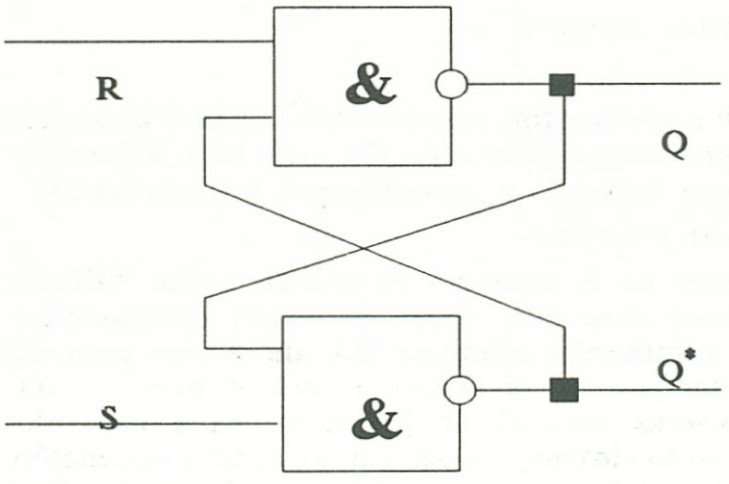
4.9. táblázat

A NAND kapuból álló bistabil áramkör igazságtáblázata.

R	S	Q	Q*
0	0	1	1
0	1	1	0
1	0	0	1
1	1	0	1
1	1	1	0

4.10. táblázat

A tároló logikai állapotai



4.20. ábra

A NAND kapukból kialakított tároló logikai rajza.

n.ütem		(n+1). ütem
R	S	Qn+1
0	0	?
0	1	1
1	0	0
1	1	Qn

4.11. táblázat

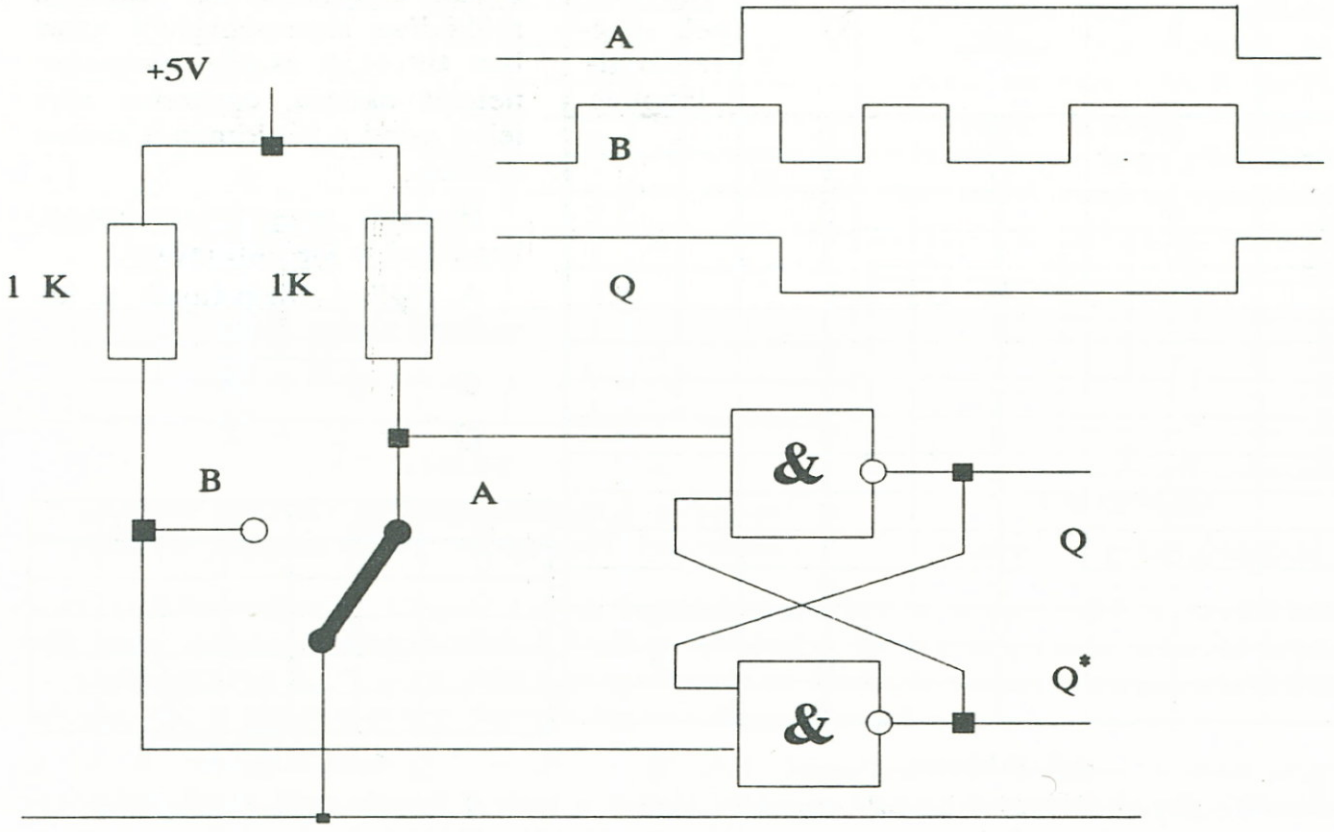
A működési táblázat szokásos alakja. A bal oldalon a bemenetekre adott vezérlést, a jobb oldalon ennek hatására kialakult állapotot tüntetjük fel. Qn jelenti, hogy a tároló állapota megőrizte az előző, n-edik periódusban felvett értékét. A „?" a tiltott állapototjelöli.

A táblázat megfelelő sorait pontosan úgy képezzük, mint az előbb. Ennek megfelelően itt is öt olyan sor marad, amelyik megfelel a logikai függvényeinknek.

A 4.10. tábla alapján itt is megfogalmazhatjuk a müködtetéshez szükséges információkat. Tiltott állapota a rendszernek az, ha mindkét bemenetére 0 kapcsolódik.Az áramkör megörzi az állapotát, ha a bemenetire logikai 1-et kötünk.A Q kimenet szempontjából S-re adott logikai 1 beírja, R-re adott logikai 1 törli a tároló tartalmát. A NOR és NAND kapukból felépített eszközök működése tehát nagyon hasonló A tiltott és tároló állapotok cserélődtek csak fel.Tekintsünk meg a továbbiakban a bistabil áramkörnek egy egyszerű és igen fontos alkalmazását.

A pergésmentes kapcsoló.

A pergés jelensége a mechanikus kapcsolók felépítéséből adódik.A mozgó érintkezőt ugyanis egy rugalmas anyag (rugó) billenti át a megfelelő állapotba. Ennek következtében



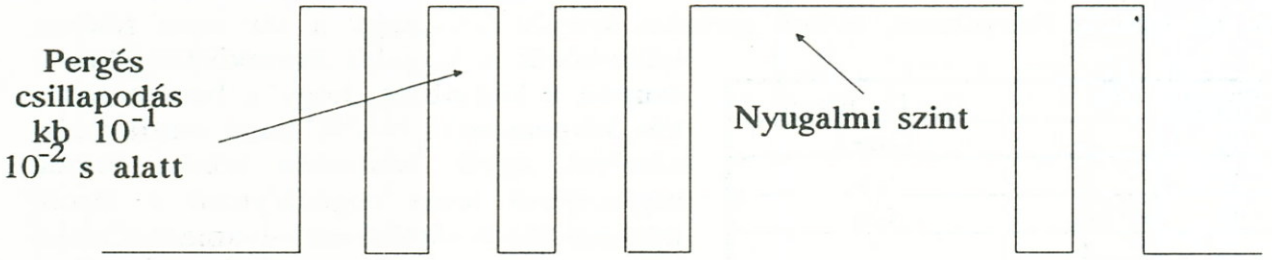
4.21. ábra

A pergés mentes kapcsoló

a kontaktus csillapodó rezgéssel csatlakozik a kiválasztott érintkezőhöz.Tehát a kapcsolás nem egy pillanatszerű jelenség,hanem több apró megszakítás után jön csak létre.Képzeljük csak el milyen bonyodalmakat okozna az ilyen kapcsoló a számítógépek billentyűzetén.

Nézzük meg, hogyan lehet ezt a problémát feloldani!

A kapcsoló érintkezője elhagyja A-t, ekkor az R bemenet H szintre kerül. Miután létrejött az első érintkezés B-vel a Q kimenet 0-ra esik vissza (a tároló tulajdonsága miatt). Ha most a kapcsoló pereg, azaz az érintkezője elhagyja B-t, de A-hoz nem ér vissza (a kapcsoló jó), akkor a tároló megtartja eredeti állapotát, hiszen mind a két bemenete 1 logikai szintre került.Visszakapcsolás csak akkor jön létre, ha a kapcsoló mozgó érintkezője visszakerül A-ba. Tehát a tároló elnyomja a pergést, az eszköz kimenetén azt nem fogjuk észrevenni. Természetesen egysarkú pergésmentesítő áramkör is létezik.

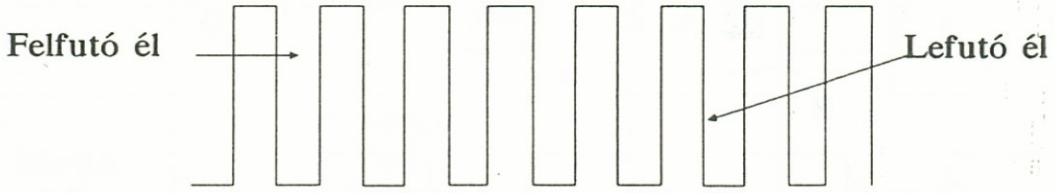


4.22. ábra

A kapcsoló pergésének a jelensége. A mozgó érintkező csillapodó rezgőmozgása

4.3.2. Kapuzott tárolók

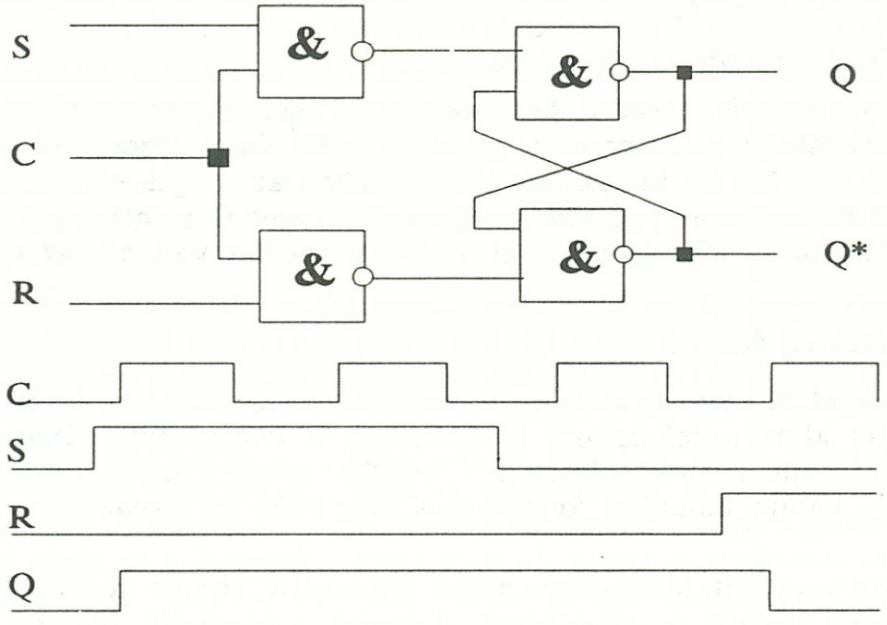
A továbbiakban olyan elemi tároló eszközöket fogunk megtekinteni, amelyekbe az adatokat csak engedélyező jel hatására lehet beírni. Ezek az áramkörök biztosítják majd,hogy a tárolást a megfelelő időben lehessen végrehajtani,megteremtve azt a lehetőséget.hogy a különböző időpillanatokban kapott adatokat a legalkalmasabb időpontban tudjuk a tárakba írni.Ez azért szükséges, mert az információ véges sebességgel (kb. 0,33 m/ns) halad át az áramkörökön, és a kapuk maguk is késleltetést okoznak. Ha bármily csekély is ez a késleltetési idő, nagy számú kapu sorba kapcsolása esetén és a különböző jelterjedési utakon haladó információ miatt, ez a késleletetés jelentős idejű,ezért az áramkörök működésében hibát okozhat.Az olyan áramköröket, amelyekben a kapukra érkező jelek nem egyidőben érkeznek **aszinkron** , míg az ettől eltérő működésűeket **szinkron** logikai hálózatoknak nevezzük.A szinkron logikai hálózatokban az áramkörök mindig egy időben



4.24. ábra

Az órajel alakja

váltanak állapotot.Ez csak úgy lehetséges, ha valamilyen szinkronizáló ütemjellel, úgynevezet órajellel informáljuk a kapukat a jelváltás időpillanatáról. Az órajel nem más mint egy periódikus négyszögjel, amelynek mondjuk H szintjére nyitnak a kapuk, L szintjére pedig zárnak.Az első eszköz, amelyet közelebbről megvizsgálunk a **C típusú tároló.** Ha



4.23. ábra

C típusú tároló és működési diagrammja.

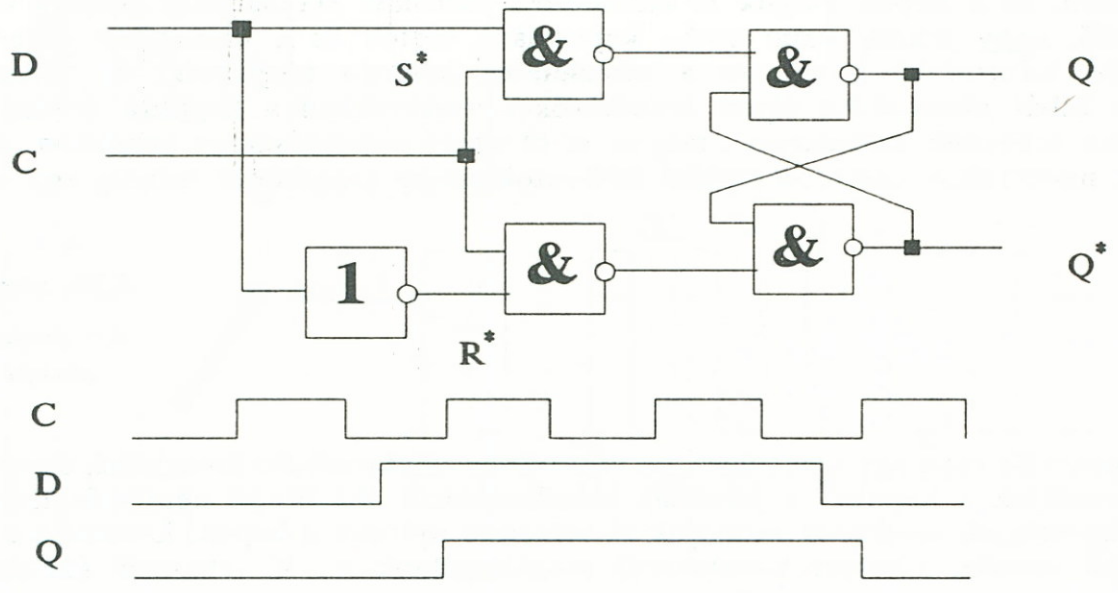
megfigyeljük az eszköz felépítését, akkor azonnal szembetűnik,hogy a tár nem sokban különbözik a bistabil áramkörtől. Annyi csupán a különbség, hogy a bemenetekre két, kétbemenetű NAND kapu csatlakozik, amelyek egyik bemenete közös. Ennek segítségével lehet engedélyezni a tároló írásánakilletvetörlésénekfolyamatát.Órajel hiányában az R illetve S bemeneteken lévő jel nem jut be a tárolóba. Amennyiben a C (clock) bemeneten logikai 1 szint jelenik meg, akkor a NAND kapuk kimeneti állapota csak a szabad bemenetük (R, S) logikai értékétől függ. Mivel a bemeneti kapukon invertálás történik,ezért a tároló átbillenéséhez C=1 és az egyik bemenet 1 szintje szükséges.

n. ütem			(n+1). ütem
C	R	S	Qn+1
0	x	x	Qn
1	0	0	Qn
1	0	1	1
1	1	0	0
1	1	1	?

4.12. táblázat

A C tároló értelmezési táblázata.Az x jelölés arra utal,hogy C=0 mellett a bemenetek értékei bármilyenek lehetnek.

Ha a kívánalmunk úgy alakul, hogy az R-S tároló két bemenetére egyszerre nem kerülhet ugyanolyan jelszint, és a tárolási idő egy órajel, akkor az úgynevezett **D típusú tároló** elemet alkalmazzuk.Elnevezése az angol Delay= késleltetés szóból származik,utalva



4.25. ábra

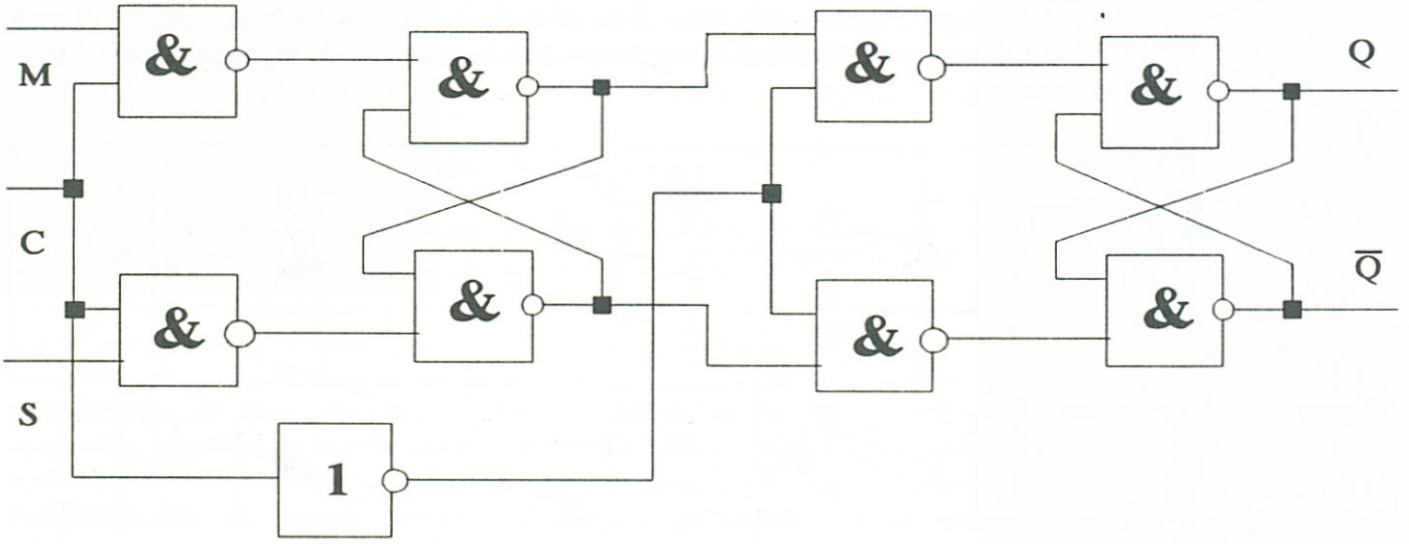
A D tipusú tároló felépítése.

arra,hogy a kimenőjel éppen egy ütemnyit késik a bemenethez képest. Látható, hogy az előző áramkörünkhöz képest a különbség csak annyi, hogy most az R^* és S^* bementeket egy inverteren keresztül kötöttük össze. Az így kialakított kapcsolásban az órajel 1 szintje alatt a kimeneten megjelenik a D-re kapcsolt jel, amennyiben az órajel 0 szintre vált, akkor a tároló megőrzi állapotát és csak a következő órajel 1 értéke mellett vált állapotot.

4.3.3. Az M-S (mester-szolga) tároló

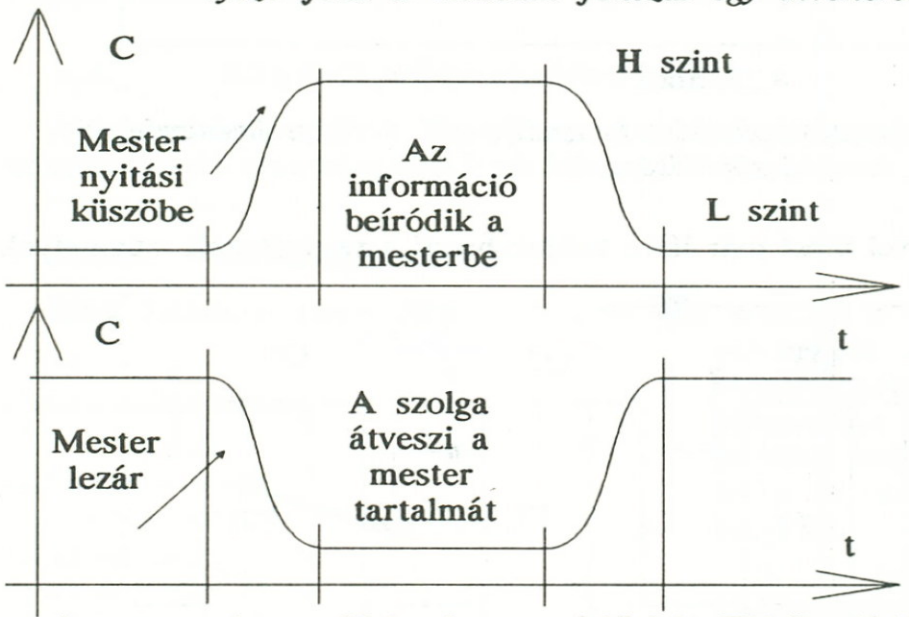
Ha a szinkron tárolókban az órajel H szintje alatt a bemenő adat átvált,akkor hibás működés következhet be. Az órajel alatt a jel egynél több kapun is keresztülhaladhat. Ez például a léptető tárolókban, számláló áramkörökben stb. okozhat gondot,ugyanis egy órajel alatt az információ csak egy kapun haladhat keresztül.A megoldás az úgynevezett **M-S tároló.**

Felépítését megvizsgálva láthatjuk,hogy két, lényegében sorba kapcsolt C típusú tárolóból épül fel, hiszen csak a C bemeneteik vezérlése különbözik. Ugyanis a második fokozat



4.26. ábra

Az M-S tároló felépítése. Jól látszik,hogy két C típusú tárolóból áll. Az engedélyező jelet a második fokozat egy inverteren keresztül kapja.



4.27. ábra

A C órajel H szintje alatt a mester felveszi a bemenetére jutó információt.A szolga csak az órajel L szintje alatt kapja meg.

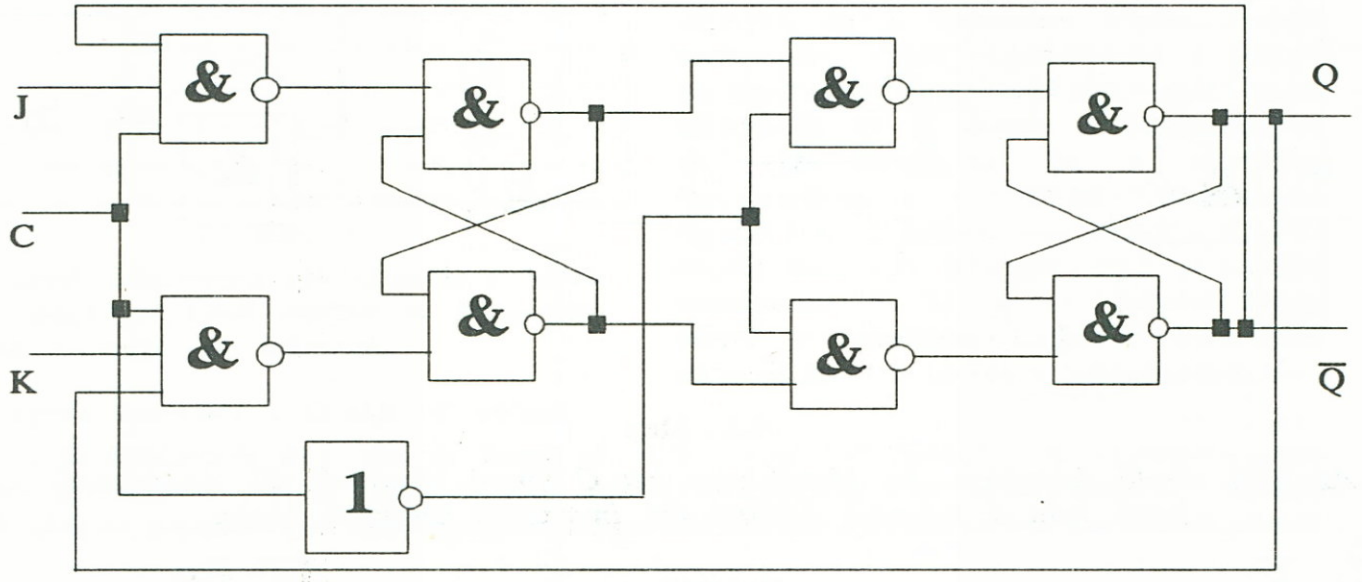
egy inverteren keresztül kapja a vezérlő jelet.Ebből adódóan az órajel H szintje alatt az információ a Mesterbe, míg L szintje alatt a Szolgába íródik át.

Részletes működése: Az órajel megérkezése előtt a bemeneti kapuk zártak, a mester az előző ütem állapotát őrzi.A szolga bemenő kapui, mivel 1 szintű órajelet érzékel, nyitottak.A nyitottság miatt a mester kimenetén meglévő értékek beíródnak a szolgába.Az órajel átváltásakor a szolga kapuja lezáródik és a mesteré kinyit, lehetőséget teremtve ezzel arra, hogy a mester felvegye a bemenetére kapcsolt információt.

Logikai áramkörök működtetése során előfordul, hogy nem tudjuk garantálni,hogy az R és S bemenet egyszerre nem kerül 1 logikai szintre (a C tároló tiltott üzemmódja).Olyan tárolókra van szükségünk,amelyek tetszőleges jelkombinációk esetén is működhetnek.Ezt a problémát visszacsatolással lehet megoldani. Ezeket a tárolókat **J-K táraknak** nevezik.

A következőkben egy M-S típusú J-K tárolóval ismerkedünk meg.Lényegében ez a tároló abban különbözik csak a szokványos M-S táraktól,hogy a Q és Q* kimenetei keresztben visszacsatoltak.A bemeneti kapuk megválasztása miatt (3 bemenetű NAND kapu) csak az a bemeneti kapu nyit ki, amelyik a visszacsatolás folytán H szintet kap.Ha a bemenetek mindegyike H szintre kerül, akkor a kimenetek az óraimpulzusoknak

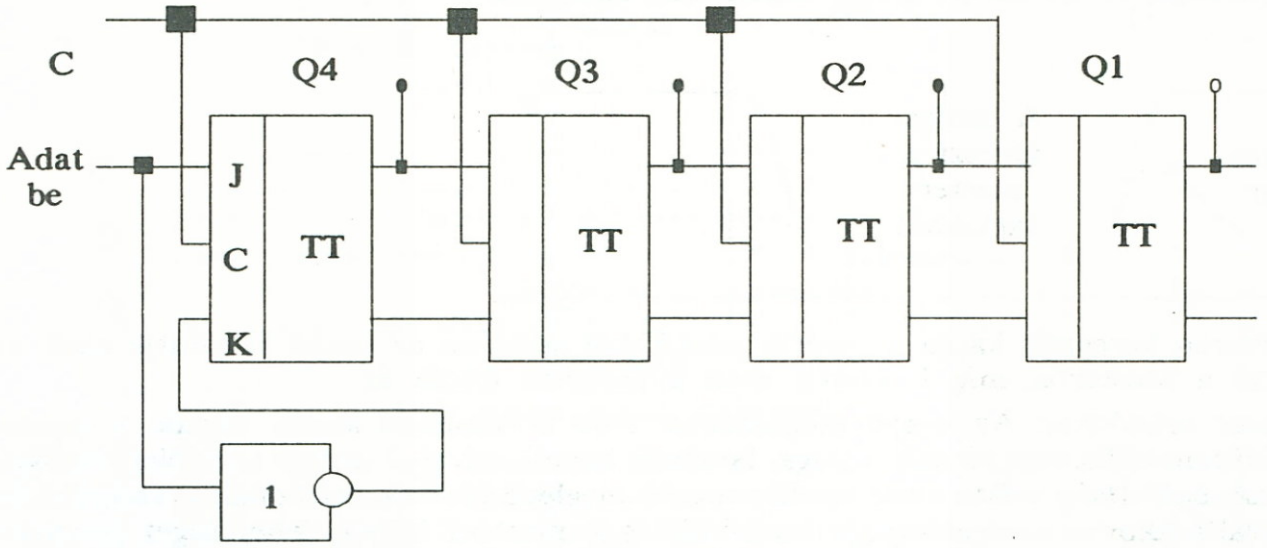
megfelelően billegnek. Az ipar nagyon sok féle J-K tárelemet gyárt.Néhány példa a sok közül SN 7476 két J-K/M-S egységet, SN76102, egy hárombemenetű J-K egységet tartalmaz stb.



4.28. ábra

A J-K tároló a kimenetről visszacsatolt jelet használja fel a tiltott üzemmód kiküszöbölésére.

A J-K (M-S) tárolók segítségével most már létre tudjuk hozni a **regiszterek** et,amelyek



4.27. ábra

Léptető tároló (Balról jobbra léptet)

képesek elemi adatok tárolására és léptetésére.Az „adat be" vezetéken soros formában lépnek be az adatok és minden egyes órajelnél eggyel tovább lépnek.

A beírás előtt a tárolót nullázni kell. (Ez rendszerint közvetlenül is megtehető,ugyanis a J-K tárcellákat úgy alakítják ki, hogy az R-S fokozat NAND kapuit három bemenetűre választják és megfelelő visszacsatolásokkal kialakítják a beíró és törlő bemeneteket,amelyek aszinkron módban működnek.)

A léptetést a legmagasabb biten kezdjük.Ide léptetjük be a legkisebb helyiértékű bitet. Az órajel H szintje alatt az első mester átveszi a bemenetén lévő adatot és a jel L szintje alatt adja át a szolgának.A következő órajel periódusban már a második tároló

Ütemszám	Bemenet	Kimenetek			
		Q4	Q3	Q2	Q1
1	1	1	0	0	0
2	1	1	1	0	0
3	0	0	1	1	0
4	1	1	0	1	1

4.13. táblázat

Léptető tároló működése a decimális $11_{dec}=1011_{bin}$ beírása során.

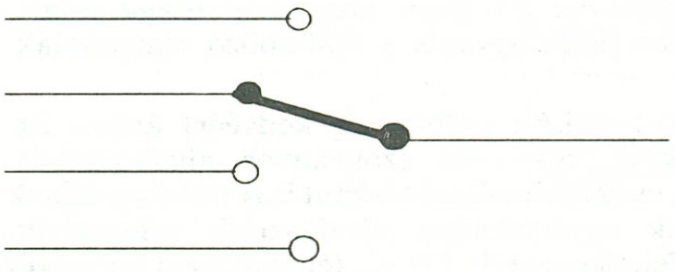
bemenetén is megjelenik a hasznos információ, így a második ütemben az beíródik a második mesterbe is.Az órajel második félperiódusában pedig átveszi annak szolgája. Így tovább, a teljes tároló hosszúságán keresztül.A léptetéseket a kívánt számban végezve feltölthetjük az egész tárolót. Ebben a példában az információ sorosan íródott be és párhuzamosan került kiolvasásra.Természetesen ettől eltérő módon is történhet a dolog.Például a tárolók közvetlen bemeneteire vezetjük a szám bitjeit a helyiértéket figyelembe véve és onnan sorosan olvassuk ki.Nyilvánvalóan ekkor a kiolvasás csak a negyedik ütem végére valósul meg teljesen, és a legkisebb helyiértékű bit hagyja el először a tárolót.

4.4. Az adatkiválasztó áramkörök

A számítógép belső felépítését tanulmányozva gyakran találkozunk úgynevezett adatkiválasztó áramkörökkel.Ezek lehetnek multiplexerek illetve demultiplexerek.

4.4.1. A multiplexerek

Ezek feladata, mint ahogyan a nevük is mutatja a különböző logikai változók közül valamelyik kiválasztása és rákapcsolása az eszköz kimenetére.Természetesen minden időpillanatban csak egy adatbemenet értéke jelenhet meg a kimeneten.Mechanikus megfelelője is van, ami jól megmutatja az eszköz lényegét.



4.29. ábra

Multiplexer, mint kapcsoló.

A K kapcsoló mozgatásával lehetséges kiválasztani a szükséges bemenetek közül azt, amelyiket a kimenetre akarunk kapcsolni.

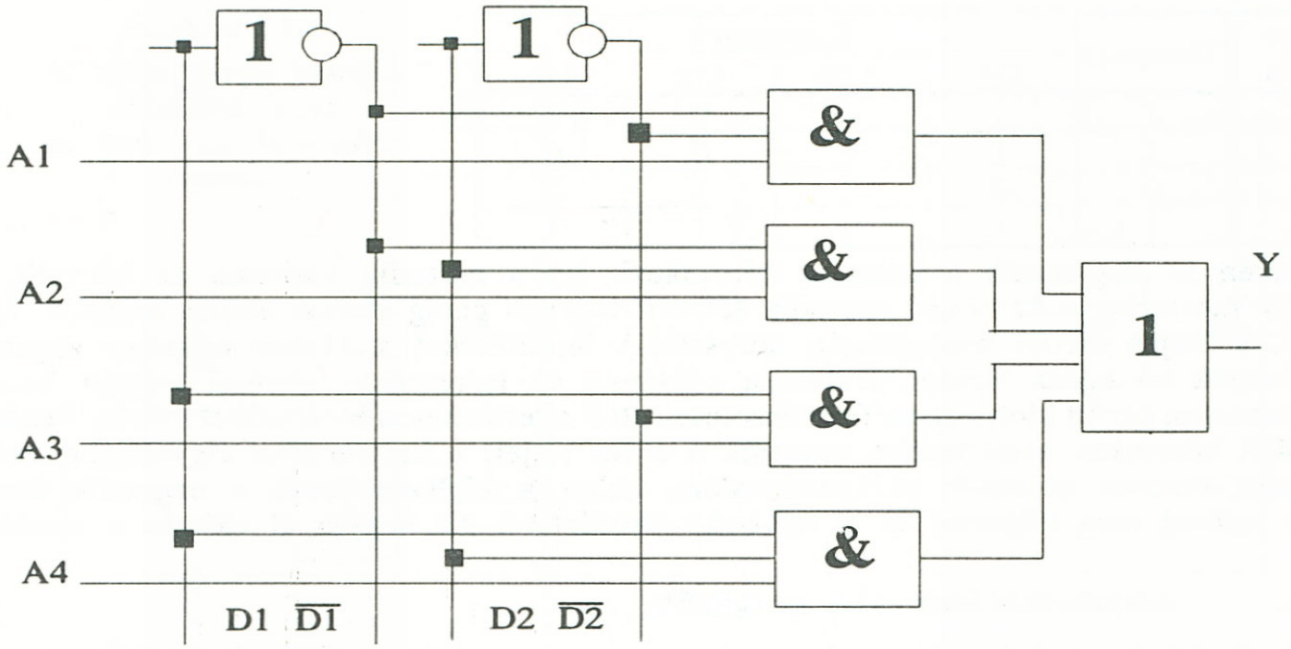
Az elektronikus megoldásnál a helyzet hasonló a D1, D2, D3, D4,... Dn címvezetékek meghatározzák a kapcsolók helyzetét.Ezek függvényében válsztódik ki az átvinni kívánt adat.

D1	D2	A1	A2	A3	A4	Y
0	0	0	x	x	x	0
0	0	1	x	x	x	1
0	1	x	0	x	x	0
0	1	x	1	x	x	1
1	0	x	x	0	x	0
1	0	x	x	1	x	1
1	1	x	x	x	0	0
1	1	x	x	x	1	1

4.14. táblázat

A multiplexer logikai táblázata.A D1,D2 az adatkiválasztó címvonal, az A1,A2,A3,A4 a megfelelő adatvonal.

Példánkban egy négybemenetű multiplexer felépítését fogjuk megvizsgálni.A négy bemenet címzéséhez elegendő két címvonal (D1,D2). Készítsük el az áramkör logikai táblázatát. (4.14. táblázat)



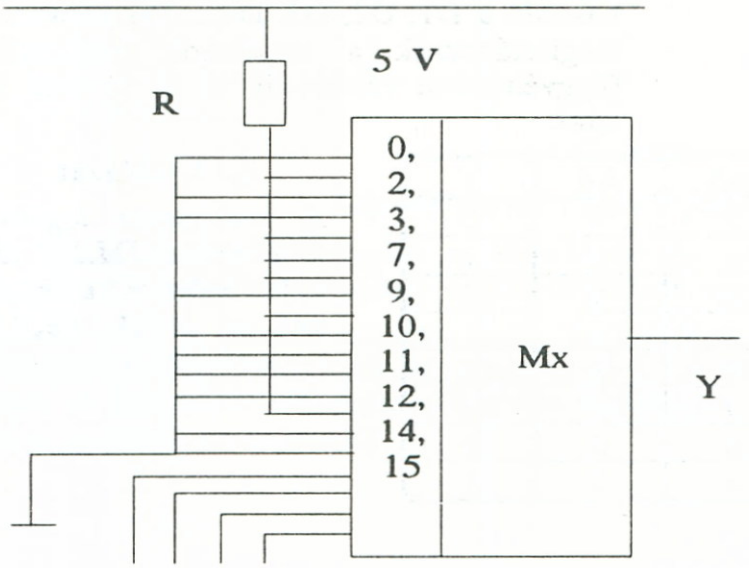
4.30. ábra

Példa multiplexerre. A D címvonala segítségével kapcsolhatjuk a kimenetre az A adatvonalakat.

A rendszer logikai függvénye ekkor:

$Y = \overline{D_1}\overline{D_2}A_1 + \overline{D_1}D_2\overline{A_2} + \overline{D_1}D_2A_3 + D_1D_2A_4$ alakban írható fel.A függvény alapján könnyen felrajzolhatjuk a kapcsolást.Az ÉS kapu alaptulajdonsága miatt mindig csak az egyik adatbemenet kerülhet a kimenetre,ugyanis a különböző címvonalak logikai értékei egymást kizárják.

A multiplexerek felhasználási területe igen sokrétű.Az érdekesség kedvéért álljon itt egy olyan példa, amely ezeknek a kapcsolóknak egy nem szokványos alkalmazását mutatja.Felhasználhatók a multiplexerek egyszerű, csak kiolvasható tárként is.A felhasználásuk azon az ötleten alapszik,hogy a címbemenetek egyértelműen kiválasztják valamelyik adatbemenetet.Ha most ezen adatbemenetek megfelelő vonalait 5 V-ra, (ellenálláson keresztül



4.31. ábra

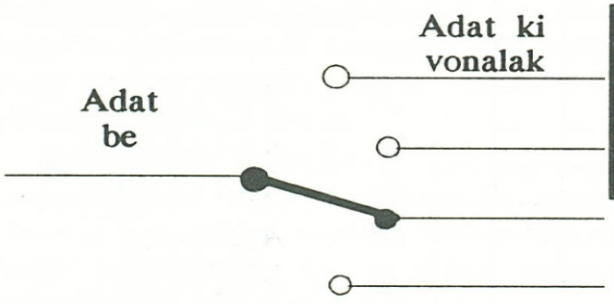
Multiplexer, mint csak kiolvasható tár.

Látható hogy a 0,2,3,7,9,10, 11,12,14,15adatvonalak0-ra (L logikai szint), a többi pedig 5 V-ra, (azaz H szintre) van kötve. A címvonalak seg1tségével a 16 bit valamelyike egyértelműen kiválasztható.

az áramkorlátozás miatt) a többit 0 V-ra kötjük, akkor a cím megjelenése után azonnal a kívánt logikai értéket kapjuk meg.

4.4.2. A demultiplexerek

Gyakran fordul elő a logikai kapcsolásoknál az előző probléma ellentéte is.Azaz a bemenő információt különböző címekre kell szétosztani. Ezt a feladatot oldják meg a



4.32. ábra

Demultiplexer, mint kapcsoló

D1	D2	A	Y1	Y2	Y3	Y4
0	0	1	1	x	x	x
0	1	1	x	1	x	x
1	0	1	x	x	1	x
1	1	1	x	x	x	1

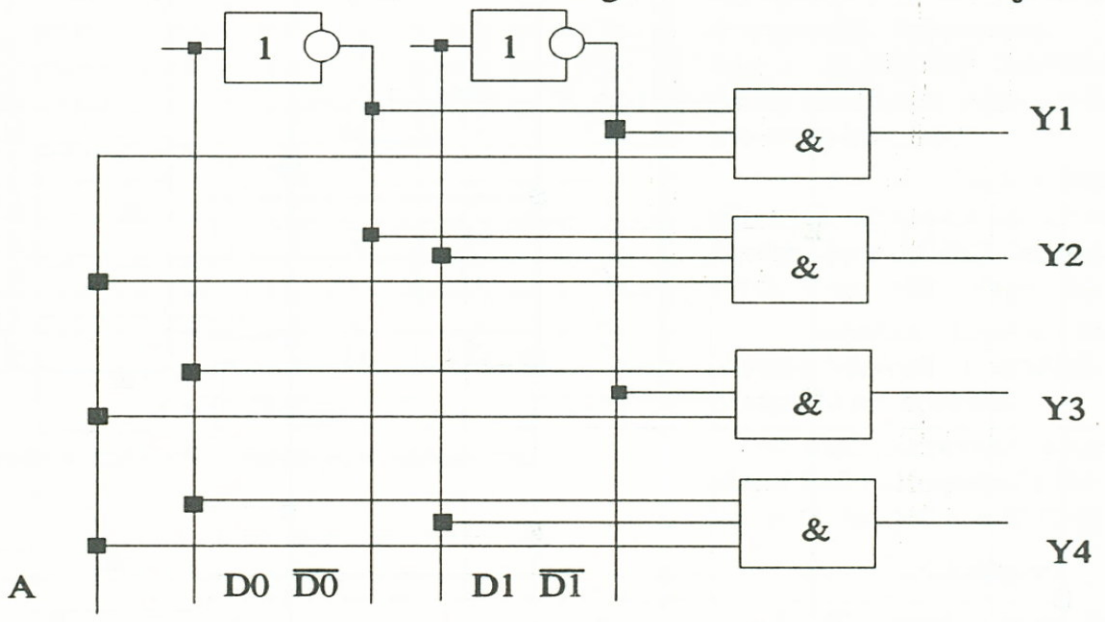
4.15. táblázat

A táblázat a demultiplexer logikai függvényét tartalmazza.Itt nem szerepeltettük az A adat 0 értékeit.Ezt azért tehetjük meg, mert a függény kialakításánál a diszjunktív normál alakot alkalmazva az úgysem játszik szerepet.

demultiplexerek.Természetesen itt is az a feladat,hogy a bemenet mindig csak egyetlen kimeneti vonallal kerülhet kapcsolatba.Készítsünk példaként egy olyan demultiplexert, amely a bemeneti adatvonalat négy kimeneti vonalra kapcsolhatja. A függvény logikai táblázatát elkészítve könnyen felírhatjuk az áramkör függvényeinek kifejtett alakját.

$Y_1 = \overline{D_1}\,\overline{D_2}A$; $Y_2 = \overline{D_1}D_2A$

$Y_3 = D_1\overline{D_2}A$; $Y_4 = D_1D_2A$ ennek megfelelően az áramkör alakja a következő.



4.33. ábra

Az egy adatvonalról négyre átkapcsoló demultiplexer.

4.4.3. Akódátalakítók

Igen sok logikai feladatban merül fel igényként,hogy valamilyen kódból egy másik fajta kódot állítsunk elő. (Pl: bináris kódból decimálisat a számológépek kijelzéséhez, a tár oszlop és sorkijelölése során stb.) Ezt a feladatot a kódátalakítók látják el. Tervezésükhöz

pontosan kell tudnunk,hogy milyen kódot milyen kódba kívánunk átalakítani,hiszen ez alapján a logikai táblázat könnyen előállítható.Példaként lássunk egy olyan átalakítót, amely közönséges számológépek kijelzőjének vezérlését oldja meg,bináris számból decimálisba alakít.Készítsük el az átalakításhoz szükséges táblázatot. A logikai függvények alakja tehát a következő.

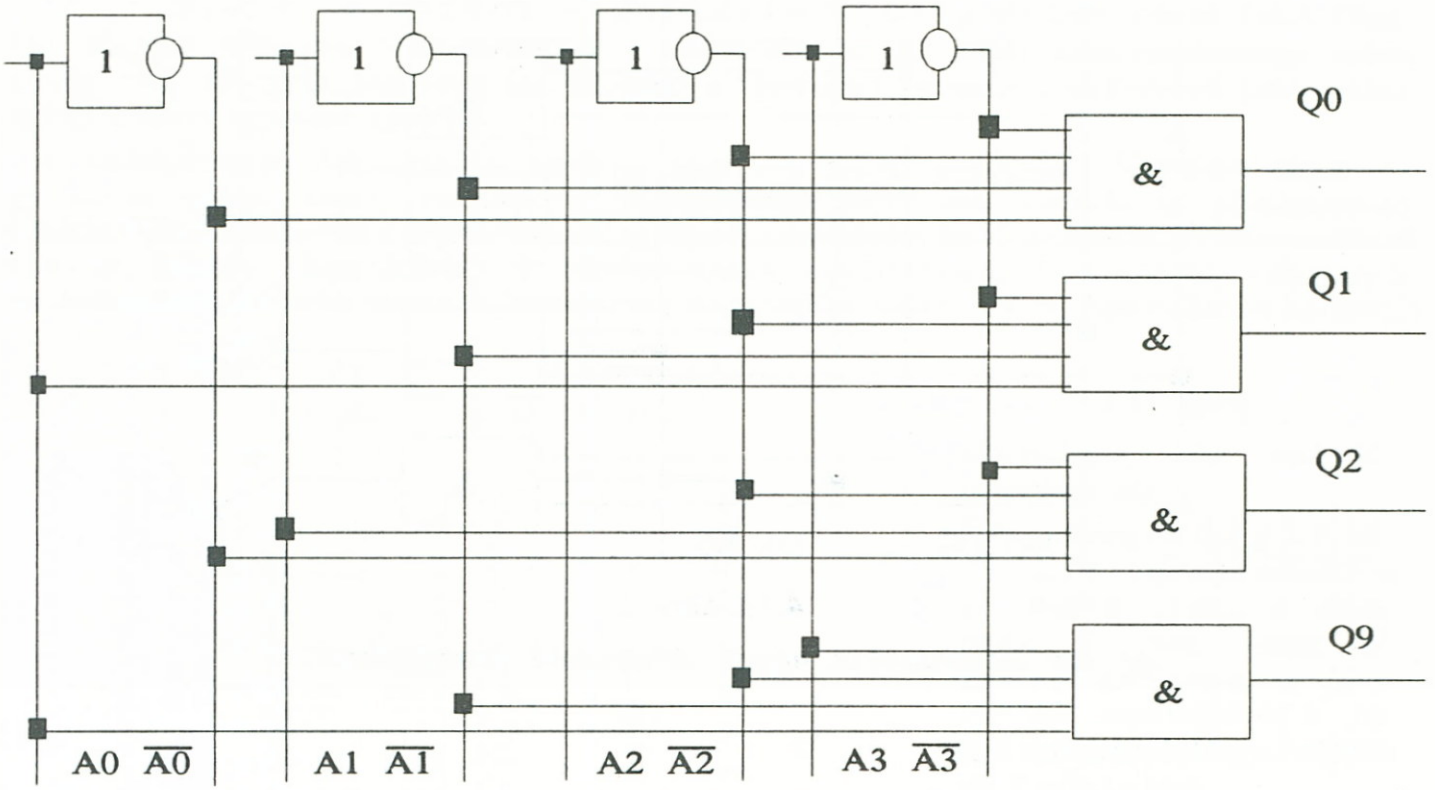
A3	A2	A1	A0	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
0	0	0	0	1									
0	0	0	1		1								
0	0	1	0			1							
0	0	1	1				1						
0	1	0	0					1					
0	1	0	1						1				
0	1	1	0							1			
0	1	1	1								1		
1	0	0	0									1	
1	0	0	1										1
1	0	1	0	1									
1	0	1	1		1								
1	1	0	0			1							
1	1	0	1				1						
1	1	1	0					1					
1	1	1	1						1				

4.16. táblázat

Binárisról- decimálisra átalakító táblázat.

$$Q_0 = \overline{A_0}\,\overline{A_1}\,\overline{A_2}\,\overline{A_3}$$
$$Q_1 = A_0\overline{A_1}\,\overline{A_2}\,\overline{A_3}$$
$$Q_2 = \overline{A_0}A_1\overline{A_2}\,\overline{A_3}$$
$$Q_3 = A_0A_1\overline{A_2}\,\overline{A_3}$$
$$Q_4 = \overline{A_0}\,\overline{A_1}A_2\overline{A_3}$$
$$Q_5 = A_0\overline{A_1}A_2\overline{A_3}$$
$$Q_6 = \overline{A_0}A_1A_2\overline{A_3}$$
$$Q_7 = A_0A_1A_2\overline{A_3}$$
$$Q_8 = \overline{A_0}\,\overline{A_1}\,\overline{A_2}A_3$$
$$Q_9 = A_0\overline{A_1}\,\overline{A_2}A_3.$$

Ezen egyenleteknek megfelelően a kódátalakító logikai rajza a következőképpen alakul.



4.34. ábra

Bináris- decimális kódátalakító logikai rajza.

4.5. Számláló áramkörök

Feladatuk, mint ahogyan a nevük is mutatja, bizonyos események (impulzusok) bekövetkezésének megszámlálása. (Pl: számítógép programlépéseinek — címképzés — és óraimpulzusainak számlálása, időzítések végrehajtása, stb.) Az elektronikus számláló áramköröknek igen sok fajtája van. Alapvetően azonban két fő csoportot szoktunk elkülöníteni, az aszinkron és szinkron számlálókat. Az aszinkront nagyon sok helyen használják, például minden olyan esetben, amikor frekvencia leosztást kell végezni, azaz nagy rezgésszámból kisebbet kell előállítani. A számítógépek belsejében azonban fontos, hogy melyik időpillanatban váltanak állapotot az áramkörök (időzítés), ezért ott szinkron számlálókathasználunk.

Készítsünk példaként, néhány aszinkron illetve szinkron számlálót, amelyek az óraimpulzusok számát határozzák meg. Hogy a számláló felépítése ne legyen bonyolult, ezért a számlálást bináris rendszerben végezzük és korlátozzuk a feladatot arra az esetre, hogy a számláló 16 impulzus beérkezése után kezdje előről a számlálást. Az olyan számlálókat, amelyek egy bizonyos számérték elérése után a számlálást előről kezdik, modulo számlálóknak nevezzük. Példánk tehát moduló 16-os vagy egyszerűen MOD 16 számláló. Előljáróban gondoljuk meg, milyen áramkörökre lesz szükségünk.Természetesen, mivel adott ideig meg kell őrizni a számláló állapotát, tárakra lesz szükségünk, és mivel óraimpulzusok számlálását végezzük JK vagy D típusú tárakat fogunk használni. Ezután gondolkodjunk el a számláló felépítésén. Tekintsük meg a következő táblázatot. (4.26. táblázat) Ennek alapján két dolgot ismerhetünk fel:

— a bi bemenetek akkor váltanak állapotot,ha az előtte lévő kisebb helyiértékű bináris változó 1-ről 0-ra vált, vagy ha

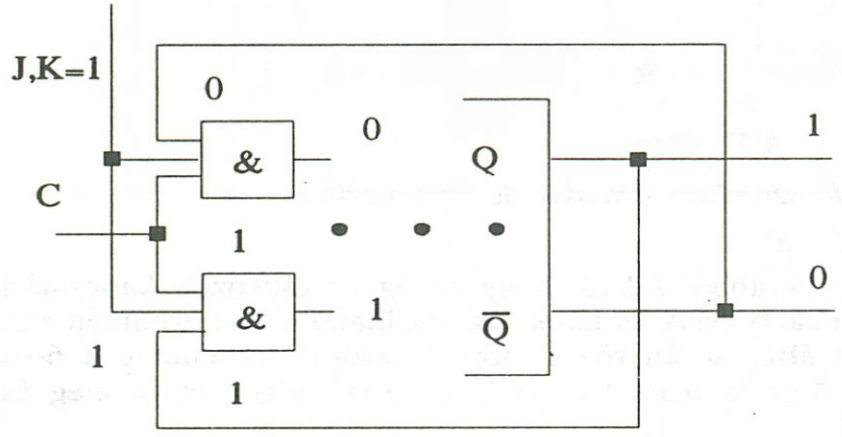
— minden kisebb helyiértékű bináris változó 1 értékű és újabb óraimpulzus érkezik.

Az első észrevétel alapján tehát akkor kell változtatni a tár állapotát, ha a C órajel 1-ről 0-ra vált. Ez

Beérkezett óraimpulzusok száma	b3	b2	b1	b0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0

4.17. táblázat

Bináris számláló működési táblázata.



4.35. ábra

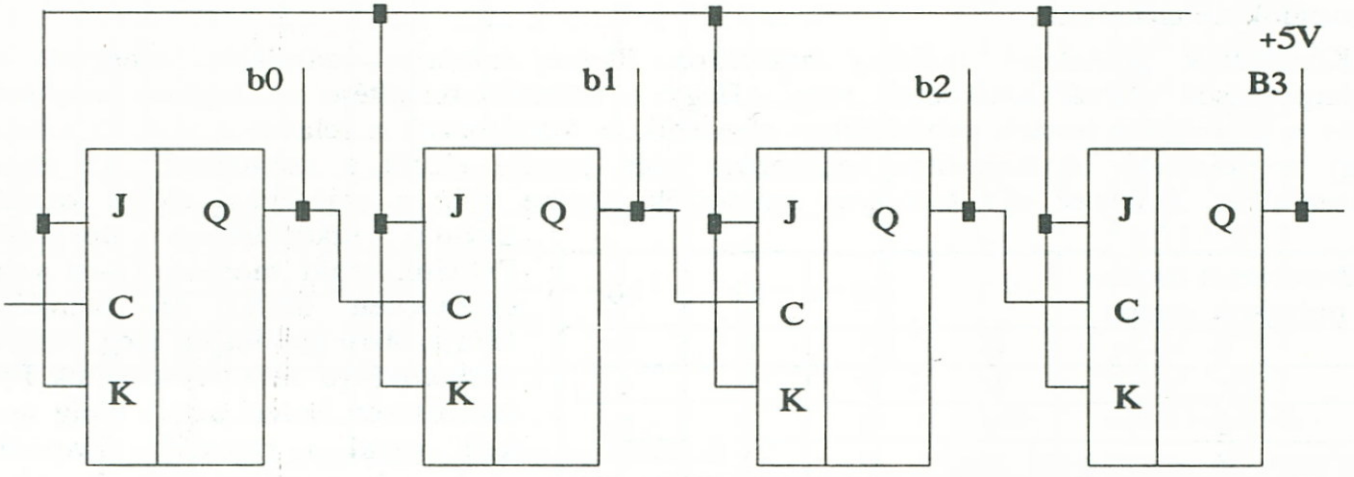
A JK tároló azon bemenete kap engedélyezést J és K logikai 1 állapota mellett, amelyik hatására a kimenet állapotot vált.

Jelen esetben a Q kimenet törlődik K törlő bemenet kap engedélyezést.

a meggondolás élvezérelt tárolót igényel. (Negatív élvezérelt tárolóról van szó, hiszen lefutó élre vált állapotot.)

Kezdjük tehát a feladat megoldását! Próbálkozzunk JK tároló felhasználásával. Tudjuk, hogy az ilyen táraknak, ha mindkét bemenetüket logikai 1-re állítjuk, a kimenete, az órajel hatására billeg.

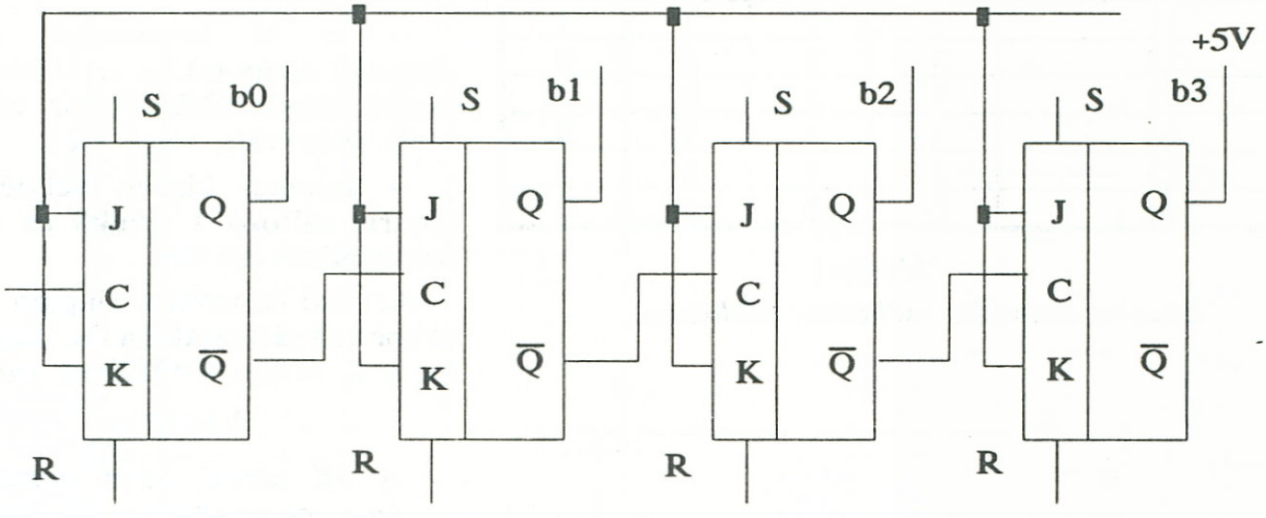
(A JK tárolót azért készítettük, hogy az MS tárolók tiltott állapotát — 1, 1 — megszüntessük. Ezért a J,K bementekre adható logikai 1 érték. Ha most órajelet kapcsolunk a C bemenetre, akkor az a bemenet kap engedélyezést, amelynek hatására a kimenet állapotot vált. — Például ha Q=1, akkor a K törlő bemenet érvényesül, ugyanis mindhárom bemenő vezeték 1 szintű.) (4.35. ábra)



4.36. ábra

Aszinkron előreszámláló áramkör logikai vázlata.

Ha sorba kapcsoljuk a tárolókat, akkor a feladatot megoldottuk. (4.36. ábra) Visszafelé számláló áramköröket is könnyedén tudunk készíteni, ha a C órabemeneteket nem a Q, hanem a $\overline{Q}$ kimenetekkel kötjük össze, az adatkimenetek változatlanul hagyása mellett. Természetesen az R és S bemeneteket is alkalmazni kell, hogy a számláló kezdeti,



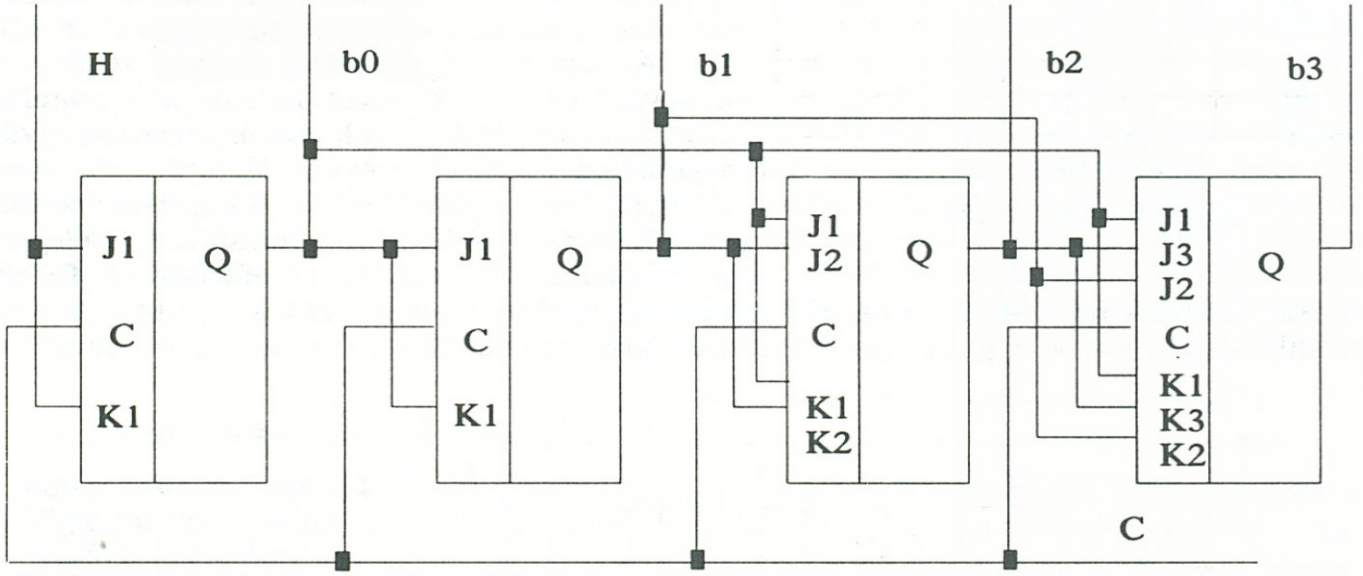
4.37. ábra

Aszinkron visszafelé számláló áramkör logikai vázlata.

indulási értékét be tudjuk írni. (4.37. ábra)

Ezekkel az áramkörökkel a gondunk abból fakad, hogy az egyes eszközök kapcsolási ideje összeadódik, ezért a jel egyre inkább késik és hamis kódok kialakulását eredményezi. Például a bináris 7-ből 8-ba való átváltás során rövid időre először a kimeneten a 6-os (az első tároló változik 0-ra) majd 4-es (a második tároló is 0-ra vált) jelenik meg és

csak ez után alakul ki a bináris 8 a kimeneteken. Ezt a versenyfutási helyzetet oldják meg a szinkron számlálók. Kialakításuk olyan, hogy minden tároló egyszerre kapja meg a számláló impulzust a bemeneten, ezért egyszerre billennnek, de hogy mégsem minden órajelre váltanak állapotot az a J,K vezérléseinek köszönhető. (Csak az első tár J,K bemenete kerül logikai 1-re.) Használjuk fel a bináris számváltásokra tett második



4.38. ábra

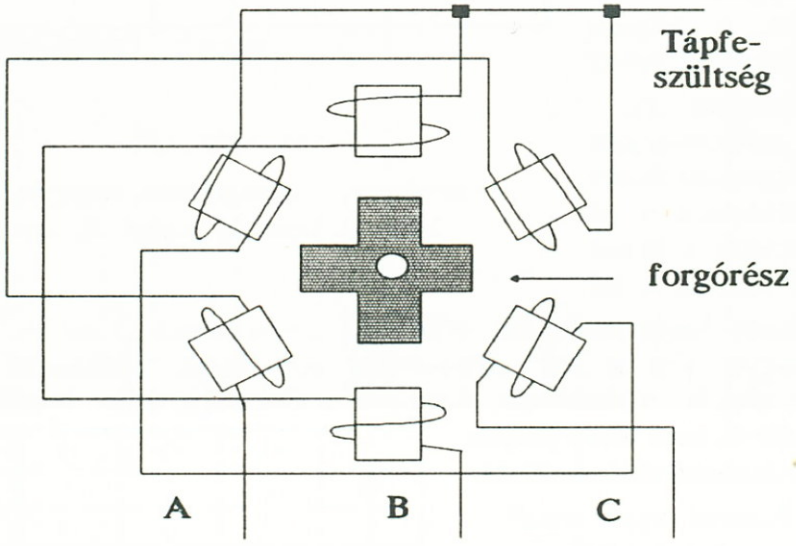
Szinkron előre számláló logikai vázlata.

megfigyelésünket, vagyis azt, hogy az adott helyiértéken akkor változik a szám érétke, ha minden előző helyiérték 1-es és újabb impulzus érkezik. (4.38. ábra)

Ez azt jelenti, hogy a tárolóknak az őt megelőző tárak számával megegyező J illetve K bemenetekkel kell rendelkeznie. Ez a probléma könnyedén megoldható, egyrészt léteznek gyárilag ilyen típusok, másrészt több bemenetű ÉS kapu felhasználásával segíthetünk a gondon.

4.6. A léptető motor és vezérlése

Sok perifériában nemcsak elektronikus, hanem mechanikus egységeket is találunk. (Például lemezes egység fejvezérlése, nyomtató papírtovábbítása, fejvezérlése, plotterek tollmeghajtása stb.) Ezek nagyon precíz, kismértékű elmozdulásokat igényelnek. A mozgatást nagyon sok esetben léptetőmotorokkal oldják meg. Ezek olyan digitális motorok,amelyeknek a forgórésze a vezérlőimpulzusok számával arányos szögelfordulást végez. (Irányváltásra is képesek.) Megkülönböztetünk aktív és reaktív forgórészű motorokat. Az aktív forgórész permanens, vagy gerjesztett mágnesből, a reaktív forgórész lágymágneses anyagokból (pl dinamólemez) készül. A legfontosabb jellemzőjük a léptetési határfrekvencia, amely azt a legnagyobb frekvenciát



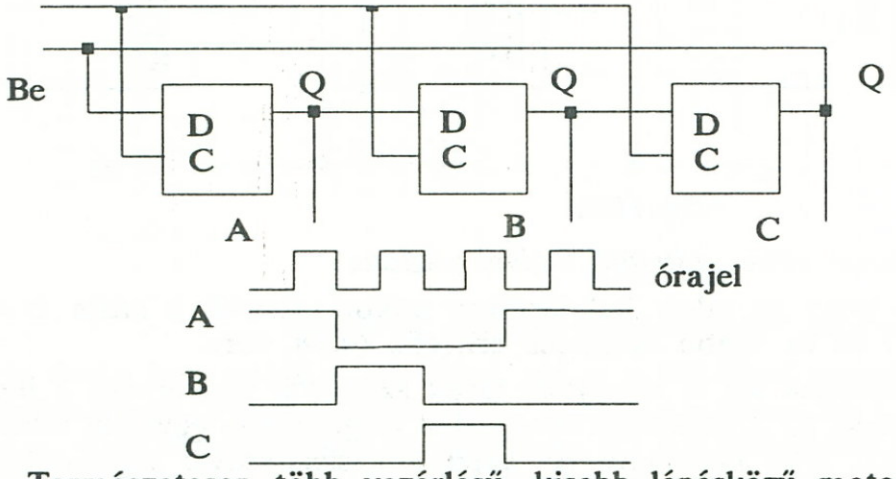
4.39. ábra

jelenti, amelynél a motor forgórésze, álló helyzetből indulva veszteség nélkül szinkronba ugrik. Elvi felépítésüket a 4.39. ábra mutatja. Ha ABC ABC sorrendben adunk gerjesztést a bemenetekre, akkor 30 fokos szöggel az óramutató járásával ellentétes, míg ACB ACB sorrend esetén vele egyező forgásirányt állíthatunk elő. Összefoglalva a léptetés és irányváltás csak az impulzusok sorrendjétől függ. A motor sebessége természetesen az impulzussorozat ütemében változik. A vezérlést egyszerűen megoldhatjuk, mint ahogyan a 4.18. táblázat indokolja a számlálók egy speciális családjával, az úgynevezett gyűrűszámlálókkal. Ezek olyan léptető regiszterek, amelyek soros kimenetét a soros bemenetükkel kötjük össze. A feladat megvalósításához elegendő a feltételnek megfelelő, D elemekből felépülő regisztert kialakítani. (4.40. ábra) Ha ennek bemenetére 1-et adunk és ezt az áramkörbe cirkuláltatjuk, akkor a motor az óramutató járásával ellentétes forgást fog végezni.

Vezérlés	A	B	C
0	1	0	0
1	0	1	0
2	0	0	1
3	1	0	0

4.18. táblázat

Léptetőmotor vezérlése.



4.40. ábra

Léptető motorok vezérlésének vázlata gyűrűs számlánc felhasználásával.

Természetesen több vezérlésű, kisebb lépésközű motorok vezérlését hasonló módon végezhetjük, de ezek hosszab számláló láncot igényelnek. Ügyeljünk arra, hogy a léptető motorok a számlálóhoz közvetlenül nem csatlakoztathatók (a motorok nagyobb áramfelvétele miatt), csak megfelelő tranzisztoros teljesítményfokozat közbeiktatásával.

5. Számítógépek funkcionális áramkörei

5.1. Az aritmetikai egység

Mielőtt hozzákezdenénk az aritmetikai egység felépítéséhez, tekintsük át milyen alapvető feladatokat kell teljesítenie.Végre kell hajtania az összeadás, kivonás, szorzás, osztásaritmetikai műveletét.Az első kettő feladat megoldását már ismerjük,hiszen az előző fejezetben megismerkedtünk ezek végrehajtásával, illetve a nekik megfelelő áramkörök kialakításával. Feladatunk tehát most az, hogy a szorzás, osztás műveletére alkalmas áramkört találjunk. Felmerülhet a gyanú, hogy ez nem lesz nehéz, hiszen a szorzást felfoghatjuk úgy, mint ismételt összeadás, az osztást pedig úgy, mint ismételt kivonást.Ami ezek szerint azt jelenti, hogy az előző fejezetben kialakított komplex összeadó és kivonó egység alkalmas a feladat elvégzésére,csak megfelelő vezérlő illetve segéd áramkörökkel kell ellátnunk.A továbbiakban megvizsgáljuk ezen műveletek menetét,hogy a kiegészítő áramkörökről fogalmat alkothassunk.

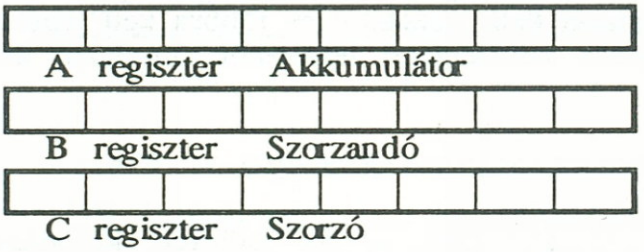
5.1.1. A szorzás művelete

A szorzás tehát visszavezethető összeadásra és léptetések sorozatára. Léptetés alatt azt értjük,hogy a szorzó számjegyeinek helyiértéke szerint eltoljuk a részletösszegeket.

10 * 3 = 30 decimális számokat binárisan összeszorozva kapjuk,hogy

```
 1010 * 0011
 ----
  1010
 1010
 ----
 11110
```

ami valóban decimálisan 30.



5.1. táblázat

Szorzásra alkalmazott regiszterek 8 bites számok esetén.

Észre kell vennünk,hogy a szorzás során a tényezők hosszánál nagyobb lesz a szorzat hossza, de ez mindig el fog férni N helyiértékű bináris számok esetén 2N biten. (Példaként tekintsük a négy biten ábrázolható legnagyobb számot a 15-öt és szorozzuk meg önmagával, ekkor 255-öt kapunk. A szorzat értéke mint ismeretes pontosan 8, azaz 2*4 biten tárolható.)

Ennek a megállapításnak természetesen az a következménye, hogy ha két azonos hosszúságon ábrázolható számot összeszorzunk, akkor ezek és az eredmény tárolására 3 darab ilyen hosszúságú tároló elemre, regiszterre lesz szükségünk.

Azt is felfedezhetjük,hogy a szorzás során ha a bináris számot 1-el szoroztuk meg, akkor magát a számot kaptuk, ha 0-val akkor természetesen 0-t.

A **szorzás** során használt regiszterek: az akkumulátor, (itt göngyöljük, halmozzuk fel a részletösszegeket,innen származik az elnevezése is. akkumulál - összegyüjt) a B regiszter, (ide kerül a szorzandó) és a C regiszter (ide tesszük be a szorzót).

Az A és C regiszterek speciális tulajdonsággal rendelkeznek,ugyanis az adatok A-ból C-be bitenként egymásba eltolhatók. Ezt a két regisztert használjuk fel arra,hogy az eredményt eltároljuk.Mint fentebb említettük, dupla olyan hosszúságú regiszterre van szükségünk ennek tárolásához, mint amelyet a tényezők tárolásánálfelhasználtunk.

A	0	0	0	0
B	1	0	1	0
C	0	0	1	1

Ezek után nézzük meg, milyen módon hajtja végre a műveletvégző a regiszterek felhasználásával a szorzást. Maradjunk a fenti példánál és nézzük meg lépésenként a 10 * 3 = 30 szorzás

végrehajtását. Az egyszerűség kedvéért most négybites regisztereket használunk.Kiindulási állapotunk tehát a következő: az A regiszter üres, B regiszterben foglal helyet a szorzandó, most 1010, a C regiszterben a szorzó 0011.A szorzás első lépése az,hogy a C regiszter legkisebb bitjének megfelelően elvégezzük a beszorzást, azaz B regiszter tartalmát hozzáadjuk az A regiszter tartalmához, ekkor a regisztereink a következőképpen alakulnak.

Regiszter név	regiszter tartalma összeadás után				Regiszter név	regiszter tartalma léptetés után			
A	1	0	1	0	A	0	*1	0	1
B	1	0	1	0	B	1	0	1	0
C	0	0	1	1	C	0*	0	0	1

A léptetés után ismét 1-et találunk a C regiszter utolsó bitjén, ezért ismét összeadást végzünk a fenti módon.

Regiszter név	regiszter tartalma összeadás után				Regiszter név	regiszter tartalma léptetés után			
A	1	1	1	1	A	0	*1	1	1
B	1	0	1	0	B	1	0	1	0
C	0*	0	0	1	C	1	0*	0	1

Mivel a továbbiakban 0-val kell a szorzást végezni, ezért 0 részletszorzatot kell az akkumulátorhoz hozzáadni. Ennek a műveletnek az elvégzése felesleges, igy tehát csak a léptetéseket kell a továbbiakban végrehajtani.

Regiszter név	regiszter tartalma léptetés után				Regiszter név	regiszter tartalma léptetés után			
A	0	0	*1	1	A	0	0	0	*1
B	1	0	1	0	B	1	0	1	0
C	1	1	0*	0	C	1	1	1	0*

Összefoglalva tehát a szorzó utolsó bitje határozza meg a művelet végrehajtását, ha ez 1, akkor a szorzandó és az akkumulátor tartalmát kell összeadni és jobbra kell léptetni őket eggyel, ha 0, akkor csak egy jobbra léptetés szükséges.A lépéseket annyiszor kell elvégezni, amennyi az operanduszok szóhossza.

5.1.2. Az osztás művelete

0	0	0	0	1	0	1	0	dec 10
1	0	1	0	0	0	0	0	normált dec 10

5.2. táblázat

Bináris szám normálása.

Az osztás ismételt kivonásra épül. Mielőtt azonban ennek elvégzéséhez hozzáfognánk, el kell végezni az osztó és az osztandó normálását.Ez azt jelenti, hogy addig toljuk őket balra, hogy a legmagasabb helyiértékű bit 1 legyen. Nézzük meg példaként a 15 osztását 3-al. A számokat normalizáljuk: 15 binárisan $1111=1{,}111*2^3$, illetve a 3 binárisan $11=1{,}1*2^1$. Természetesen ebben az esetben a hányados nagyságát külön ki kell számolni: $\frac{1{,}111*2^3}{1{,}1*2^1} = \ldots 2^{3-1} = \ldots 2^2$. Próbáljuk elvégezni a műveletet a szorzásnál megismert három regiszteres rendszerben. Töltsük normalizálva az A regiszterbe az osztandót és a B regiszterbe az osztót.Vizsgáljuk meg, hogy az A

```
1111 : 11 = 101
-11          hányados
---          5
001
-00
---
0011
 -11
 ---
  00         maradék 0
```

regiszter tartalma nagyobb egyenlő-e a B regiszter tartalmánál. Ha ez teljesül, akkor képezzük az A—B különbséget. Ezt vigyük egy hellyel balra eltolva az A regiszterbe. A C regiszter balra léptetése után, 1-et írunk a legalacsonyabb helyiértékre. Ha A< B akkor az A—0 különbséget képezzük, majd ismét 1 hellyel balra léptetjük A-t. A C regiszter balra léptetett tartalmához, a legalacsonyabb helyiértékre, most 0-t Írunk. Az

Regiszter név	regiszterek eredeti tartalma				Regiszter név	regiszter tartalma kivonás után			
A	1	1	1	1	A	0	0	1	1
B	1	1	0	0	B	1	1	0	0
C	0	0	0	0	C	0	0	0	0

Regiszter név	regiszter tartalma léptetés után				Regiszter név	regiszter tartalma kivonás után			
A	0	1	1	0	A	0	1	1	0
B	1	1	0	0	B	1	1	0	0
C	0	0	0*	1	C	0	0	0*	1

Regiszter név	regiszter tartalma léptetés után				Regiszter név	regiszter tartalma kivonás után			
A	1	1	0	0	A	0	0	0	0
B	1	1	0	0	B	1	1	0	0
C	0	0*	1	0	C	0	0*	1	0

Regiszter név	regiszter tartalma léptetés után			
A	0	0	0	0
B	1	1	0	0
C	0*	1	0	1

eljárást tovább folytatjuk, az utolsó (4.) lépés után C-ben megkapjuk a hányados normalizált alakját. 1,010 * 2^2 = 0101, ami nem más mint 5.

5.1.3. Egy akkumulátoros aritmetikai egység vázlata

Az algebrai műveletek megvalósításának birtokában most már elkészíthetünk egy egyszerű aritmetikai egységet. Előtte foglaljuk össze a szükséges ismereteket!

Szükségünk van tehát három regiszterre az operanduszok és művelet eredményeik tárolásához.Ezek közül kettő egymásba léptethető, jobbra illetve balra eltolható. A harmadik regiszter komplementálható (kivonás miatt). Ezeken kívül szükséges egy párhuzamos összeadást megvalósító áramkör, valamint adatutak a regiszterek feltöltéséhez, illetve az adatok összegzőbe juttatásához.

A műveletek végrahajtását ennek megfelelően a következő módon végezhetjük:

Összeadás

- A regiszterbe töltjük a nagyobbítandót,
- B regiszterbe a hozzáadandót.
- Az eredmény az A regiszterbe kerül.

Kivonás

- A regiszter tartalmazza a kisebbítendőt,
- B regiszterbe kerül a kivonandó komplementálva.

» A különbség az A regiszterbe kerül, az eredmény előjelét az átvitelbit tartalmazza.

Szorzás

» B regiszterbe kerül a szorzandó,

» C regiszterbe töltjük a szorzót.

» Az eredmény az A és C regiszterbe kerül tárolásra.

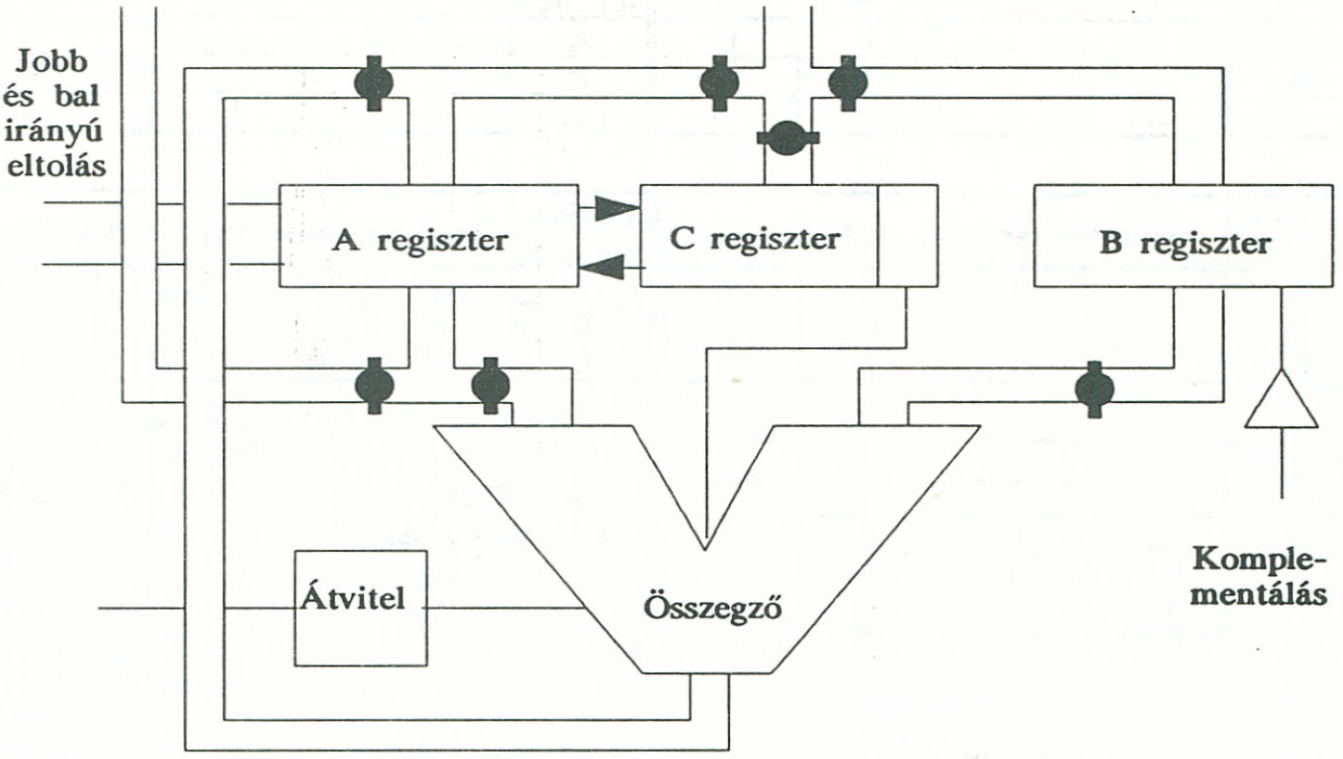
Osztás

» Az A és C regiszter tartalmazza az osztandót normalizált alakban,

» B-be kerül az osztó.

» Az eredményt az A és C regiszter tartalmazza, C a hányadost, A pedig a maradékot.

Ezek után nézzük meg, hogyan is néz ki az aritmetikai egységünk.



5.1. ábra

Az egyszerű aritmetikai egység felépítése.

Ha bináris lebegőpontos műveletet végzünk, akkor a mantissza kiszámítását a fen leírt módon végezhetjük, a kitevők kiszámításához külön blokkra van szükségünk.

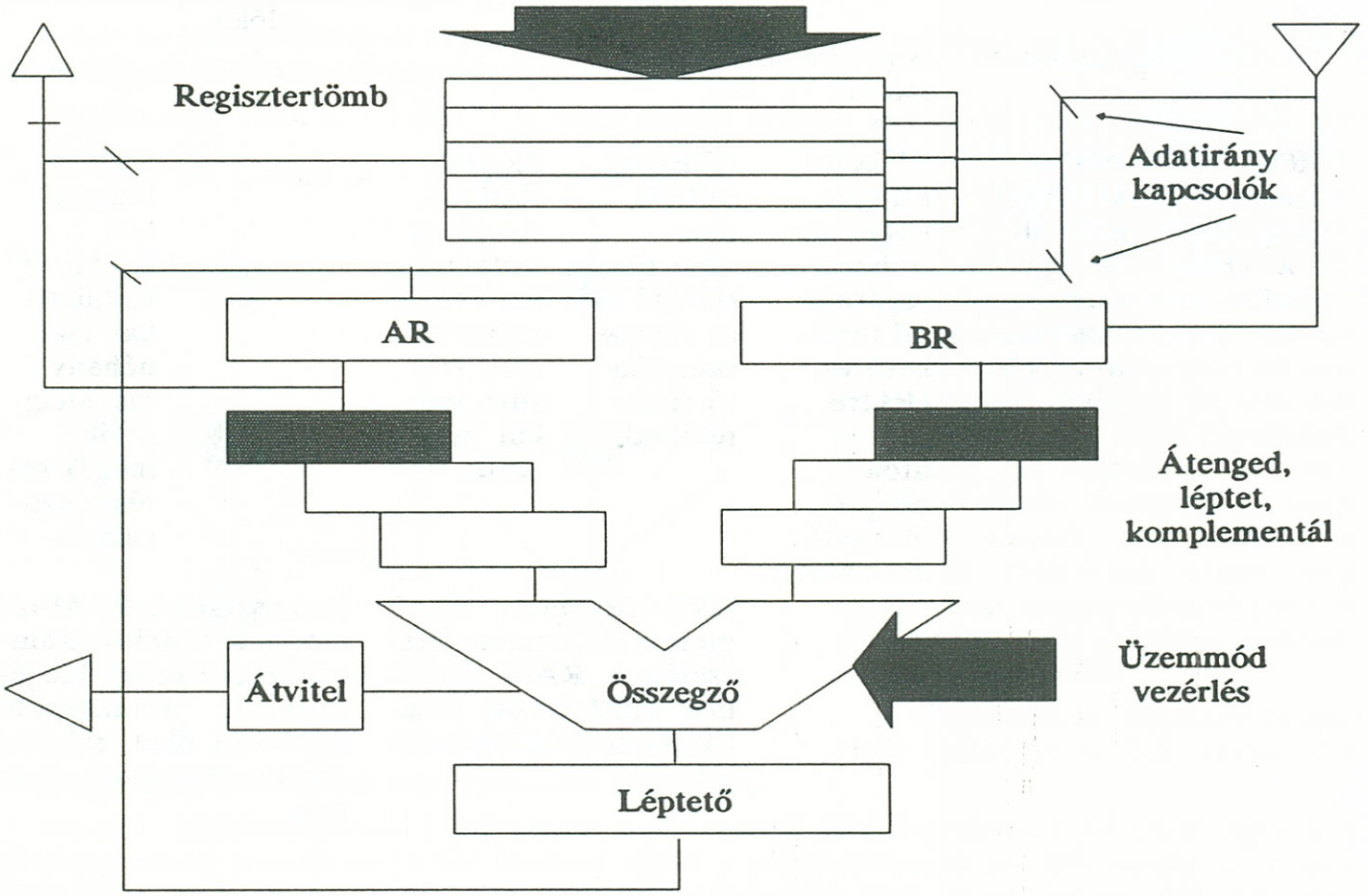
5.1.4. A több akkumulátoros aritmetikai egység

A harmadik generációs számítógépekben megjelenik az úgynevezett regisztertár, a gyors **előmemória** (scratch - pad), amely átalakította az aritmetikai egység szerkezetét.Lényeges változás, hogy az aritmetikai műveletek kiértékelése során nem szükséges az adatokat a központi tárba visszaírni, hanem lehetőségünk van a regisztertömbben való tárolásra,ami lényegesen gyorsabb elérési és ezzel együtt gyorsabb műveletvégzést jelent.

A regiszterek nem végeznek léptetést és komplementálást, hanem az adatutakon lévő áramkörök felelősek ezek végrehajtásáért,ami szintén növeli a művelet végrehajtási sebességet.

Az áramkör kialakítása során gondot kell fordítani az adatutak vezérlésére, hiszen ezek határozzák meg, hogy honnan kell az adatokat elővenni és hova kell küldeni őket Ezeket

a kapcsolókat különálló vezérlő egység irányítja, amely esetleg még programozható is.(Lásd később a mikroprogramozott vezérlő működésénél!) Ezek után tekintsük meg az aritmetikai egység felépítését!



5.2. ábra

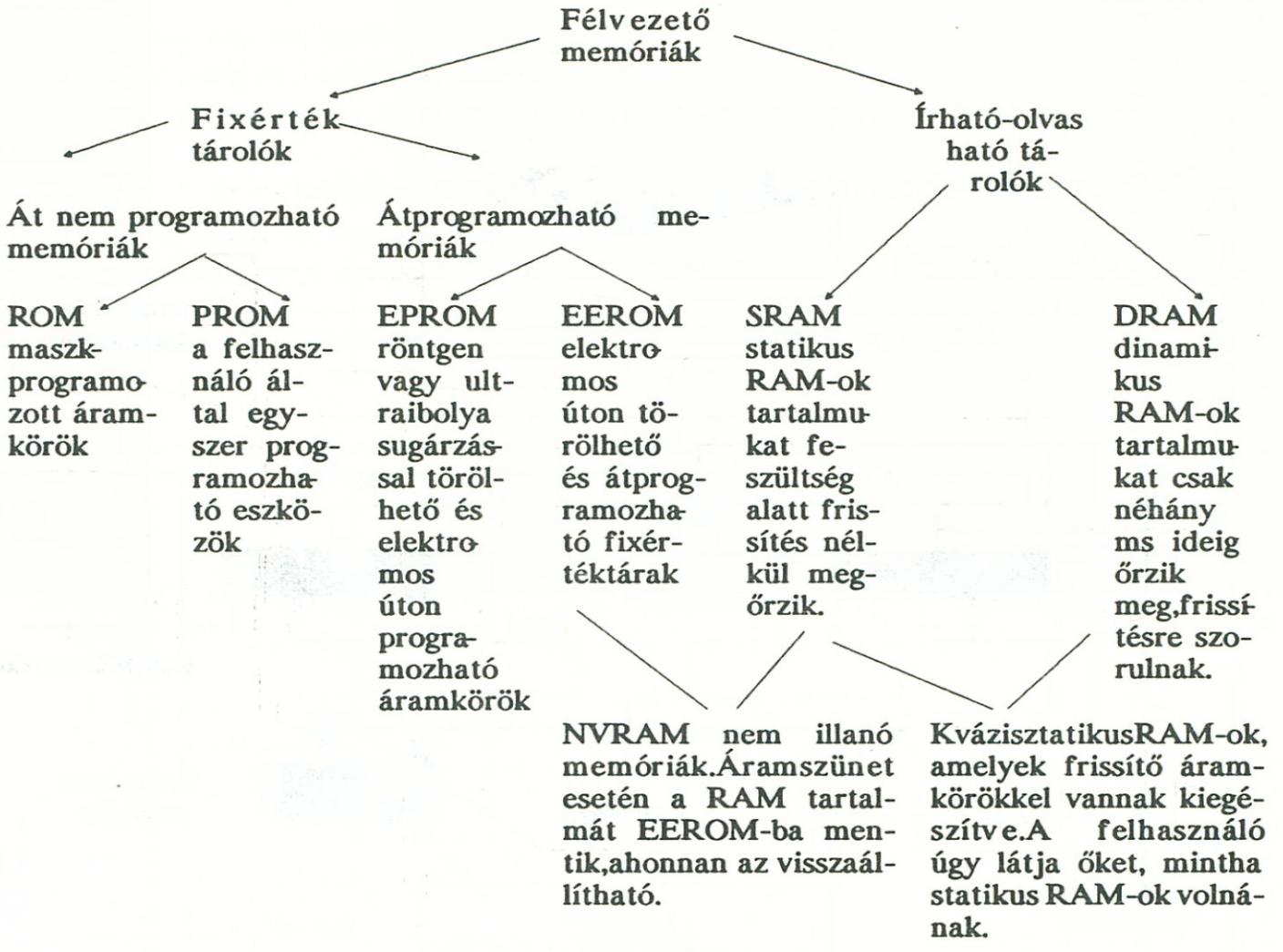
A harmadik generációs számítógépek aritmetikai egységének elvi felépítése.

5.2. Félvezetőmemóriák

5.2.1. Általános bevezető

A félvezető memóriák nagy családjában számos, a számítástechnika legkülönbözőbb területein használt eszközt találunk.Ezen eszközök kialakulásának előfeltétele volt a nagybonyolultságú integrált áramkörök megjelenése (LSI , VLSI technológiák). A számítógépek felhasználóinak igénye egyre nagyobb memóriakapacitást sürgetett.A gépek üzemeltetése során szinte azonnal el lehetett különíteni a memóriák két, talán legfontosabb csoportját, az úgynevezett fixértéktárakat és az írható-olvasható memóriákat. Azonban a memóriakat igen sokféle módon csoportosíthatjuk, például a hozzáférés módja, a cella

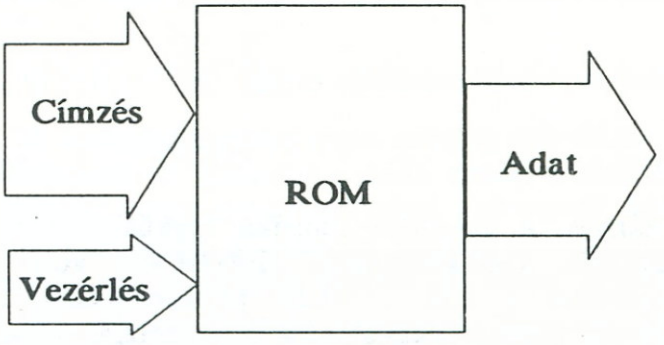
típusa, előállítási technológiája stb. szerint. Mi a továbbiakban az alkalmazás módja szerint fogjuk a memóriákat rendszerezni.



A következőkben sorra vesszük ezeket az áramköröket és részletesen megvizsgáljuk felépítésüket.

5.2.2. A fixértéktárolók általános felépítése

Nézzük meg közelebről ezt a tároló alkatrészt. Első pillanatban kiderül,hogy jól elkülöníthető vezétékcsoportok kötik össze a külvilággal. Ilyen a cím, a vezérlő illetve az adatveztékek csoportja. (5.3. ábra) A kérdés csupán az,hogy hol találjuk meg az áramkörön belül az adatainkat.Nos ezek az úgynevezett **tármátrix**ban foglalnak helyet.A mátrix elhelyezés azért indokolt, mert így a legkevesebb vezeték segítségével lehetséges az adat helyének megjelölése.



5.3. ábra

A ROM tár kapcsolata a külvilággal.

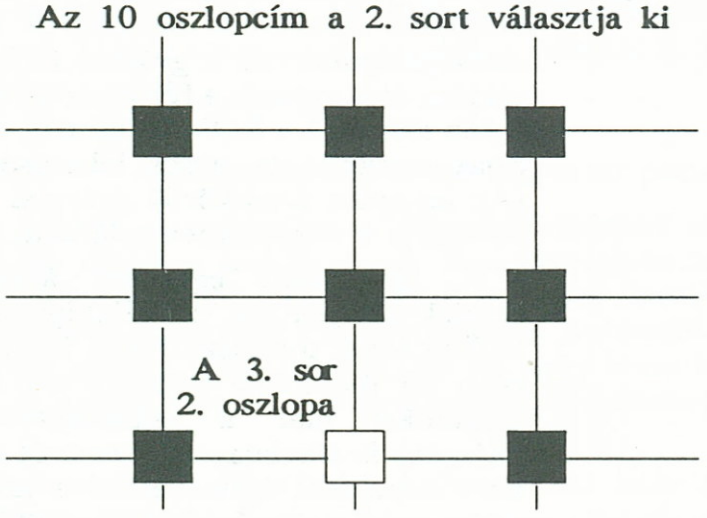
Példának okáért tekintsünk egy 9 cellából álló tárat.Ha minden cellát külön akarunk megcímezni,kiválasztani,akkornyilvánvalóanehhez 9 darab egymástól független vezetékre van szükségünk. Ha a mátrix elhelyezésnél

maradunk,akkor elegendő 3 + 3 = 6 darab.Természetesen minél nagyobb a tároló kapacitása, annál jobban jelentkezik a mátrix elhelyezés előnye. (Pl: 100 cella esetén nem 100, hanem csak 20 vezeték elegendő a kijelöléshez.)

A **címvonal** akon érkező bináris információ kódolva határozza meg a tárelem helyét, ezért a sor és oszlopkijelölő vezetékekhez tartozó információt ebből kell dekódolni. Például 4 bites cím esetén 16 rekesz címezhető. Mondjuk az alsó két bit a tár sorait, a felső két bit a tár oszlopait azonosítja. Ekkor az 1011 címinformáció a 3. sor és 2. oszlop találkozásánál fellelhető elemet határozza meg.

A sor és oszlopvezetékek kijelöléséhez tehát dekódoló áramkörre van szükségünk. (Lásd előző fejezet kódátalakító című alfejezetét.)

Tudjuk,hogy ezek az áramkörök nagy számú azonos jellegű kapuból épülnek fel, így tehát ügyelni kell a kapuk megengedett terhelésére, illetve a megfelelő jelszint biztosítására.Erre szolgálnak a **címmeghajtó** áramkörök.



5.4. ábra

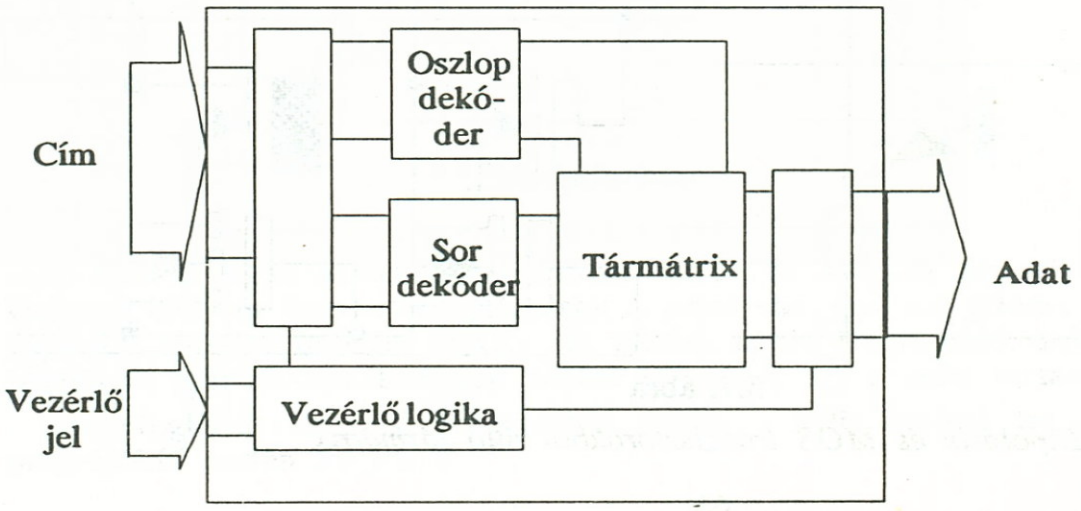
A mátrix elvi felépítése

Természetesen az sem mindegy, hogy a számítógép központi egysége mikor érheti el a kiválasztott adatot. Ezért vezérlő egységet is találunk az áramköri tokon belül. Gondoljuk csak meg ,ha számítógépünk memóriája olyan kapacitású, hogy egyetlen csipből nem tudjuk a memóriát előállítani, akkor még úgynevezett csippkiválasztó jelre is szükségünk van a helyes működtetésszempontjából.

A fixértéktárak általános felépítését láthatjuk a 5.5. ábrán. Jól megfigyelhetjük az előbb leírt áramköri blokkokat.

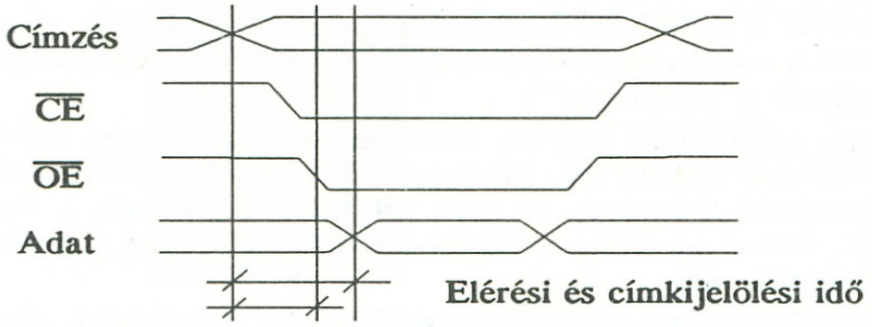
Az adat kiválasztásának a folyamata a következő! Első lépésként a központi egységtől címinformáció érkezik,majd kis késéssel előáll a csippkiválasztó jel CE, amely L szinten aktív.Ezek után a memória olvasás engedélyezését kap. Nem sokkal ezt követően rendelkezésre áll a tár kimenetén a kiválasztott adat.

A cím kijelölés és az olvasás engedélyező jel megjelenése között eltelt időt teljes címkijelölésnek,míg a cím információ és az kiválasztott tárcella tartalmának memória adatvonalára helyezése között eltelt időt elérési időnek nevezzük.



5.5. ábra

ROM áramkör blokkdiagrammja.

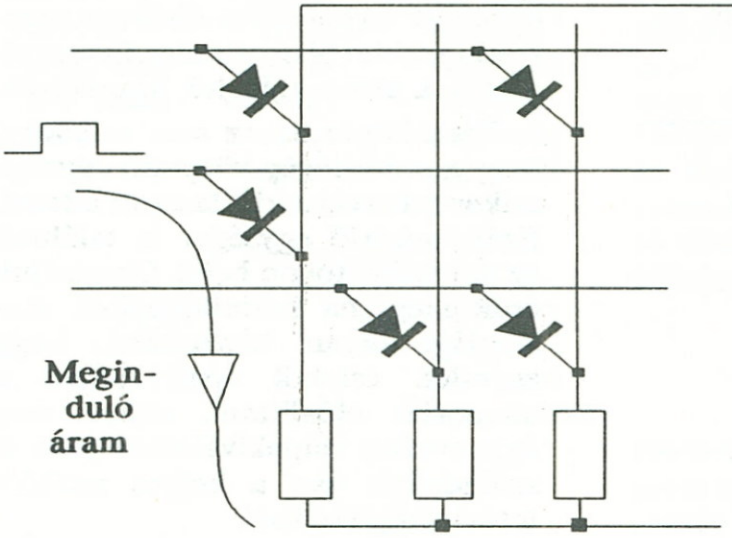


5.8. ábra

A ROM olvasási idődiagrammja. A CE a csipp kiválasztó jel, az OE az olvasás engedélyezése.Mind a két jel alacsony szinten aktív.

5.2.3. A ROM áramkör felépítése

Tekintsük először a maszkprogramozott ROM-okat és nézzük meg,hogyan és milyen áramköri elemekből alakíthatók ki. A maszkprogramozás a gyártás technológiájára utal,ugyanis a félvezető gyártása során alakítják a ROM tartalmát, ennek átprogramozására nincs lehetőségünk. (Az integrált áramkörök gyártási technológiája a függelékben található.)

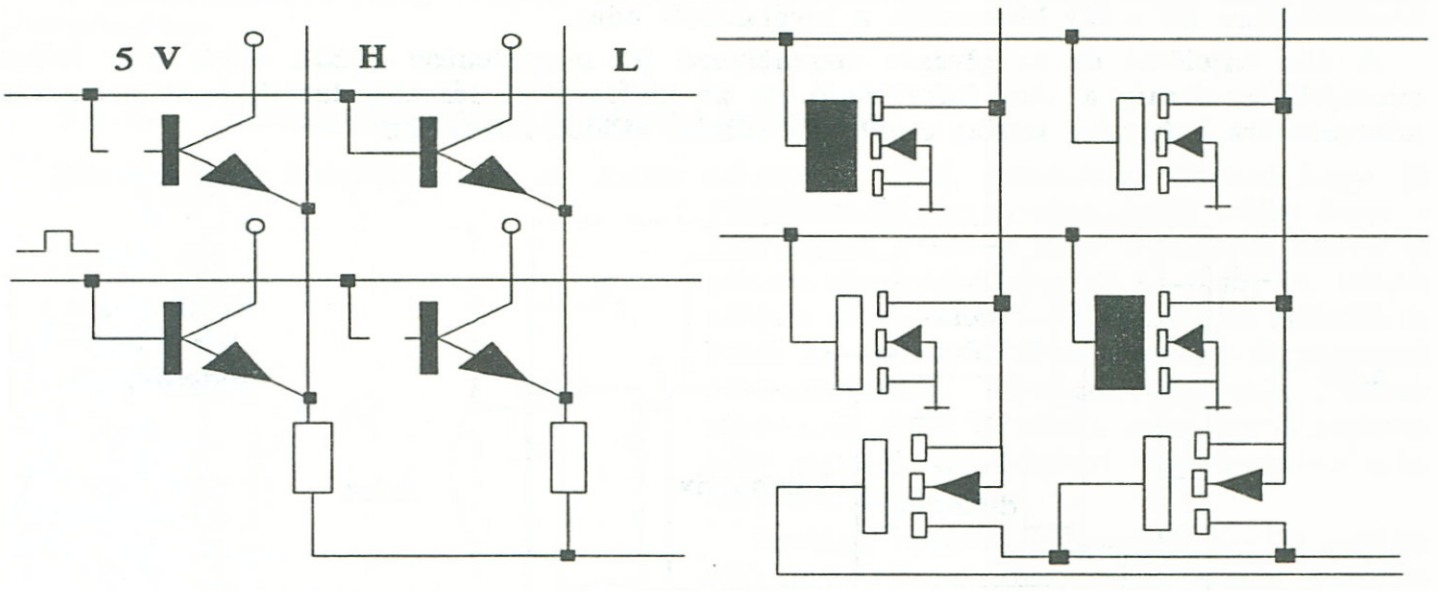


5.6. ábra

Diódás tármátrix

A tárcellákat egyszerű áramköri elemek alkothatják, diódák,bipoláris illetőleg MOS tranzisztorok.A legegyszerűbb tár diódákból állítható elő. Ezt a megoldást már a számítástechnika hajnalán is alkalmazták, diszkrét áramköri elemekből építették fel a tárakat. Ma természetesen integrált áramköröket alakalmaznak, de az elv ugyanaz maradt.Tekintsük meg az 5.6.ábrát,ahol a **diódás tármátrix** elvi kapcsolási rajza látható.

A vízszintes rekeszkiválasztó vezetékek egyikén (a címnek megfeleően) H szint jelenik meg. Ez azokban a függőleges vezetékekben, amelyeket dióda köt össze a vízszintes



5.7. ábra

Bipoláris és MOS tranzisztorokból álló tármátrix

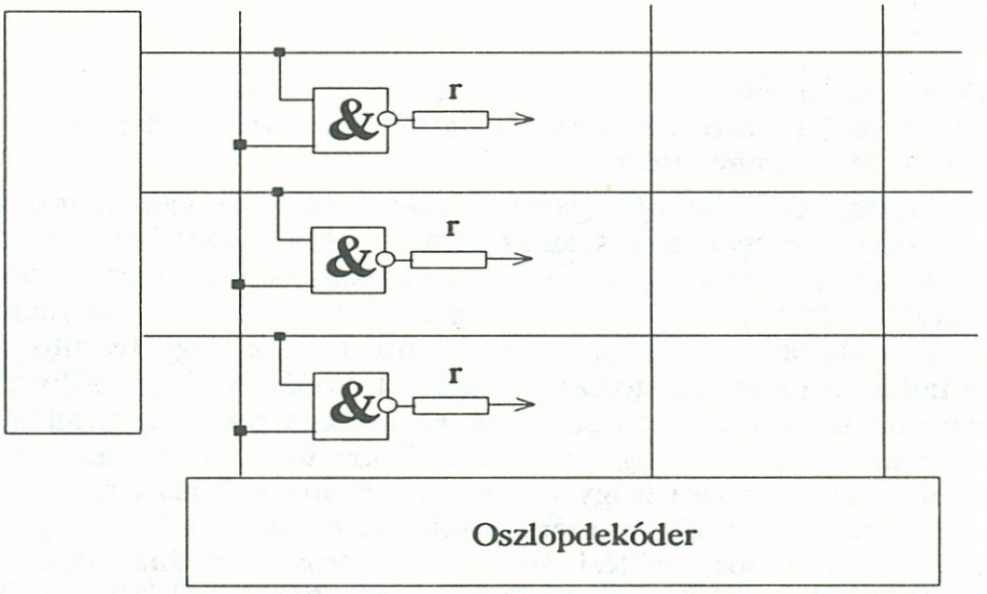
vezetékkel H szintet eredményez, (a dióda vezet, áram indul meg rajta, a teljes feszültség a diódán illetőleg az ellenálláson esik) azokban, amelyeket nem köt össze dióda, L szintet kapunk.

Hasonló elrendezések láthatók az 5.7. ábrán is, ahol bipoláris illetve MOS tranzisztorok alkotják az összeköttetést a rekeszkiválasztó és adatvezetékek között. Manapság ez a leggyakoribb formája a ROM-ok kialakításának,ugyanis a tranzisztorok erősítési tulajdonsága miatt kisebb meghajtóteljesítményt igényelnek, mint diódás társaik. Ha jól megfigyeljük az ábrákat,akkor különbséget fedezhetünk fel a két áramköri elem elhelyezésében. Míg a bipoláris tranzisztor esetén a programozást úgy oldják meg, hogy a tranzisztor bázisát összekötik,vagy sem a kijelölő vezetékkel, addig a MOS tranzisztoros megoldásnál mindegyik áramköri elem ugyanolyan módon van összekötve a megfelelő vezetékekkel. A programozást úgy végzik, hogy a tranzisztor csatornáját és kapuját elválasztó szigetelő réteg vastagságát változtatják meg. Vastag szigetelő esetén a rekeszkiválasztó vezetékre adott H szint ellenére sem nyit ki a tranzisztor.A nyitáshoz magasabb küszöbfeszültség tartozik, mint a normál esetben.

5.2.4. A felhasználó által egyszer programozható memóriák.

A **PROM** áramkörök annyiban különböznek csupán az előbb ismertetett társaiktól,hogy a diódák illetőleg tranzisztorok összeköttetésének kialakítását a felhasználó végezheti el. Eredendően minden dióda vagy tranzisztor össze van kötve a rekeszkiválasztó és kiolvasó vezetékekkel, mégpedig egy ellenálláson keresztül. Ennek az ellenállásnak olyan a szerepe, mint egy olvadó biztosítéknak. Ha túl nagy áram halad át rajta, akkor kiolvad, megszakítva ezzel az összeköttetést. A programozást tehát ennek az ellenállásnak a kiégetésével oldhatjuk meg.

Részletesebben felrajzolva a tárcellákat láthatjuk, hogy a sor és oszlopdekóderek által előállított rekeszkiválasztó jelek hogyan csatlakoznak a **programhid** akhoz A vízszintes és függőleges vezetékeket összekötő nyitott kollektoros NAND-kapuk a rekesz kiválasztását oldják meg.Mint tudjuk, ha mindkét bementük H szintű, akkor a kimeneten L szint

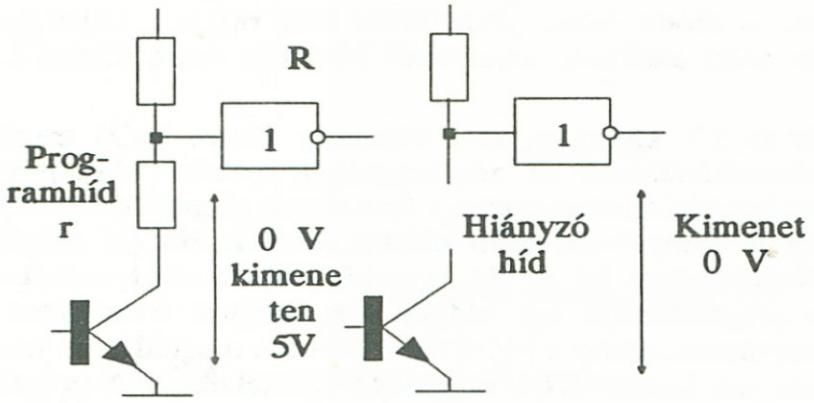


5.9. ábra

Az egyszer programozható memóriák rekeszkiválasztó logikája

van, azaz a kapu kimenő tranzisztora zárt. Ha megvan az „r" programhíd, akkor a kiolvasó inverter bementén ugyancsak L szint van, ami azt jelenti, hogy a cella tartalma logikai 1 lesz (az inverter miatt). Ha viszont nincs programhíd, akkor a kiolvasó inverter bemenete minden körülmények között H szintű, így a cella tartalma logikai 0 lesz.

Összefoglalva: meglévő programhíd esetén a cella logikai tartalma 1, míg kiégetett programhíd esetén logikai 0.



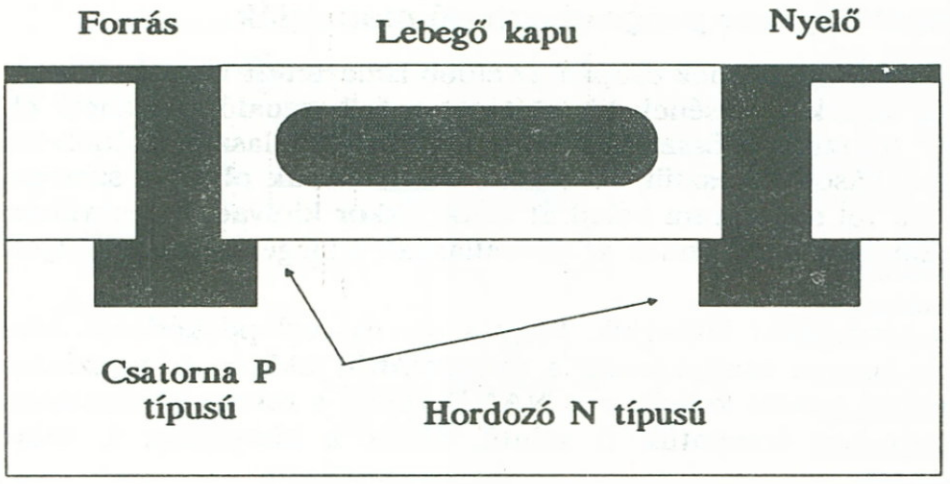
5.10. ábra

A programhíd kiégethető, megléte esetén az inverter bemenete L kimenete H szintű.

Kiégetett programhíd esetén az inverter bemenete H szin- tű kimenete L.

5.2.5. Az átprogramozható ROM-ok

Az átprogramozható ROM-ok alatt olyan ROM áramköröket értünk, amelyekben vissza tudja a felhasználó állítani a kezdeti állapotot, és az eszközt újból programozhatja.Az



5.11. ábra

A FAMOS tranzisztor felépítése. A kaput a lavina effektus segítségével tölthetjük fel.

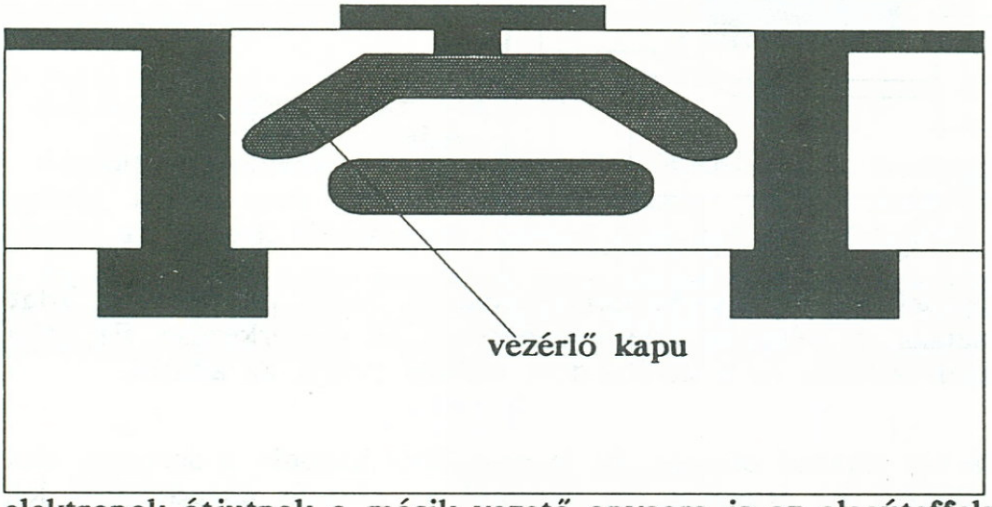
áramkör kialakítása hasonló a már megismertekhez, de a MOS tranzisztor felépítését átalakították, előállítva ezzel az úgynevezett **FAMOS** (Floating-gate avalanche-junction MOS) lebegőkapus,lavinaletöréses MOS tranzisztort.

Az ilyen típusú MOS tranzisztornak a kapuja (gate) szigetelő anyagba van ágyazva és nincs kivezetése (ezért nevezik lebegőkapus tranzisztornak). Alaphelyzetben ez a tranzisztor zárt, a csatornában nincs töltéshordozó. Ha elegendően nagy,de igen rövid ideig tartó feszültséget kapcsolunk a forrás és nyelő elektróda közé, akkor a záró irányban előfeszített diódán létrejön a lavinaletörés jelensége.Ez azt jelenti, hogy a nagy feszültség hatására elektronok áramlása indul meg a záró irányban előfeszített diódán.A töltéshordozók olyan nagy sebebsséggel mozognak,hogy magukkal sodorják az ütközés miatt az útjukba kerülő elektronokat.(Ez a folyamat ha hosszú ideig tart, akkor tönkre teheti a tranzisztort.) A gyorsan repülő elektronok eljutnak a kapura is,így rövid idő (néhány ms) alatt feltöltik azt. Mivel a kaput jó szigetelő réteg veszi körül, ezért a felületére került töltések igen hosszú ideig ott maradnak. (A folyamatos kisütési áram, - elektronok elvándorlása a gate-ről - 10^{-40} A, ami azt jelenti, hogy 10% feszültségcsökkenés körülbelül 80 - 100 év alatt következik be. Addigra a ROM-ban tárolt információ jelentős része elavul!)

Ha a bevitt töltéseket törölni akarjuk,akkor sugárzással ionizált csatornát kell létrehozni a szigetelő anyagban,ami miatt a kapu elveszti töltését. Az ionizáló sugárzás legtöbbször ultraibolya sugárzás,de lehet röntgen sugárzás is. Nagyszerű ultraibolya sugárforrásként használható a közönséges kvarclámpa.Az áramkör tartalma néhány órás „napoztatással"

törölhető. Az EPROM cellák felépítését és a tárcella szerkezetét mutatja be az 5.11. ábra.

A következőkben egy különleges, manapság egyre jobban terjedő ROM tárat mutatunk be. Ez az áramkör az EEPROM , (Electronically Erasable PROM) az elektromosan törölhető ROM. Működési elve a kvantummechanika alagút effektusára épül. Tegyük fel, hogy két vezetőanyagot vékony szigetelő anyag választ el.Az egyikre vigyünk fel töltéseket. Ekkor a kvantummechanika szerint véges és nem nulla annak a valószínűsége, hogy az



5.12. ábra

Az EEPROM áramkör felépítése hasonlít a FA-MOS elemekhez,azzal a különbséggel,hogy a lebegő kapu fölé egy vezérlő kaput is elhelyeztek.

elektronok átjutnak a másik vezető anyagra is az alagúteffektus révén (mintegy átfúrva a kialakuló potenciálgátba).

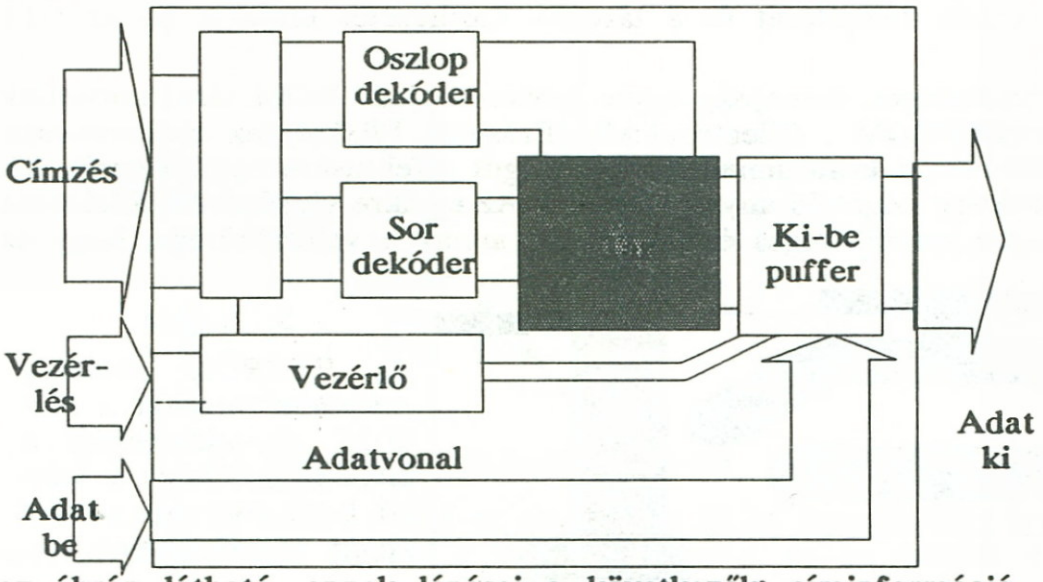
Az EEPROM elem egy elektromosan szigetelő szilíciumdioxid egykristályban épül fel, amelyet egy p típusúra szennyezett szilíciumhordozón helyeznek el. A kristály felülete alatt található az úgynevezett vezérlőkapu, amely elektromosan érintkezik a külvilággal és ez alatt a lebegőkapu, amelyet már megismertünk. Mindkét elem poliszilíciumból készül. A p típusú hordozóban kialakitanak két n szennyezésű szigetet.

A már említett alagúthatásban lévő elektronok úgy képződnek, hogy a nyelőre kicsiny, a vezérlőkapura pedig nagy pozitív feszültséget kapcsolunk. A forrás, a nyelő és a vezérlőkapu most már úgy viselkedik, mint egy közönséges MOS tranzisztor. A meginduló elektronáram néhány elektronja feltölti a lebegő kaput.

A lebegőkapuról törléskor úgy távolíthatjuk el az elektronokat,hogy a nyelőre kis pozitív feszültséget, a vezérlőelektródára meg nagy negatív feszültséget kapcsolunk. Ezzel mintegy letaszíthatjuk a lebegő kapun lévő elektronokat. Sajnos ezek az eszközök nem programozhatók át tetszőleges sokszor,ugyanis a feltöltéskor és kisütéskor az alagúteffektusban résztvevő elektronok közül néhányat a rácshibák magukba zárnak,ezzel úgynevezett szivárgóáram csatornákat hozva létre, amelyek a további felhasználást lehetetlenné teszik. Az áramkörök élettartama körölbelül 10000 átprogramozási ciklus. A kutatók azonban jelentős szerepet szánnak ezen celláknak, hiszen lehetőséget teremtenek például arra, hogy áramszünet esetén a RAM áramkörök tartalmát automatikusan elmenthessük az EEPROM-okba és visszaírhassuk onnan az áramszünet után.

5.2.6. Az írható olvasható memóriák (RAM)

Az általános felépítésük vizsgálatával gyorsan végezhetünk,hiszen nagyon sok elemüket a ROM-ok tárgyalásánál már megismertük. Ilyen áramköri egységek a tármátrix, a sor illetve oszlop dekodoló, a címmeghajtó és a vezérlő áramkör. Akkor mégis mi a különbség a két áramkör között? Az egyiket a vezérlő áramkör felépítésénél találjuk meg, amely szerint nemcsak olvasás engedélyező és csipp kiválasztó vezérlőjeleket találunk, hanem irás engedélyezés vezérlőjelet is tartalmaznia kell. Nyilván funkciójánál fogva két részt különíthetünk el az áramkör vezérlésében is. Az első legyen a tárba írás folyamata. Mint

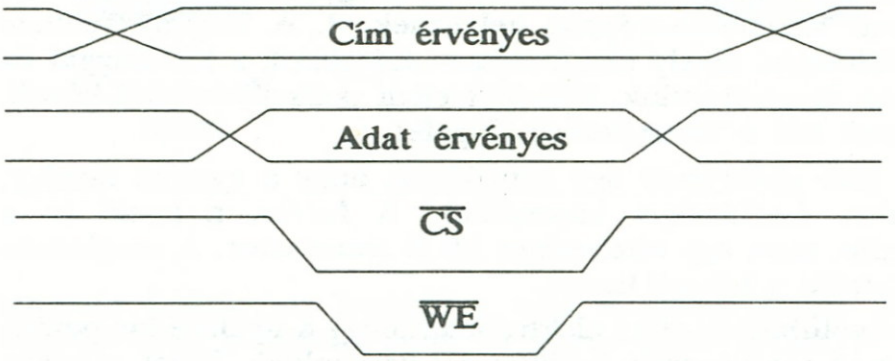


5.13. ábra

A RAM áramkörök blokkdiagramm-ja.

az ábrán látható, ennek lépései a következők: cím információ, majd ez után az adat megjelenése, a csipp kiválasztása, és végül az írás engedélyező jel megérkezése. Ez után a megfelelő RAM egység aktivizálódik és a kiválasztott cellába beírja az adatot.

A második lehetőség a tárból történő olvasás. Itt is nagyjából hasonló a sorrend, cím információ majd csipp kiválasztó jel érkezése, és legvégül az olvasás engedélyező jel, melynek hatására az adat kiolvasása megtörténik a megfelelő áramkörből. Vizsgáljuk meg a másik lényeges különbséget, amely a tárcellák felépítéséből adódik!

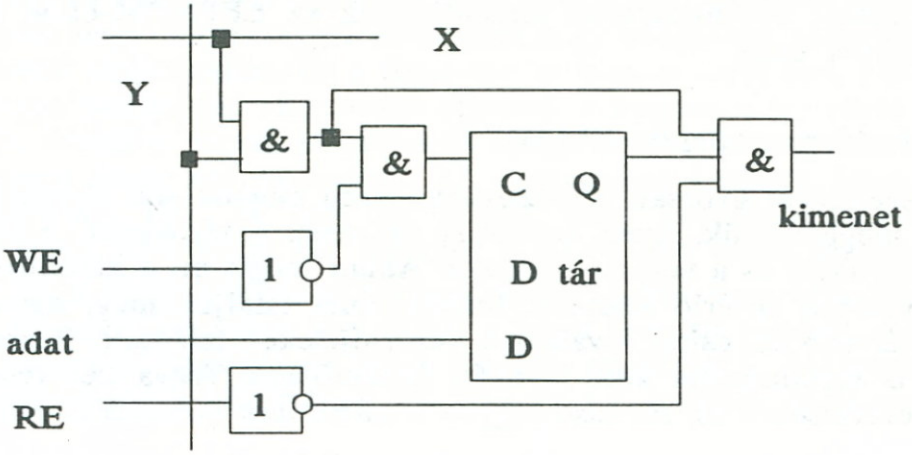


5.14. ábra

A RAM írás idő-diagrammja.Az olvasás diagramm megegyezik a ROM-nál leírtakkal.

5.2.7. A statikus RAM áramkör

A tárolórekesz tulajdonképpen egy D típusú tároló egység, ezt látjuk el a címzéshez, az olvasás és írás vezérléséhez szükséges logikával.



5.15. ábra

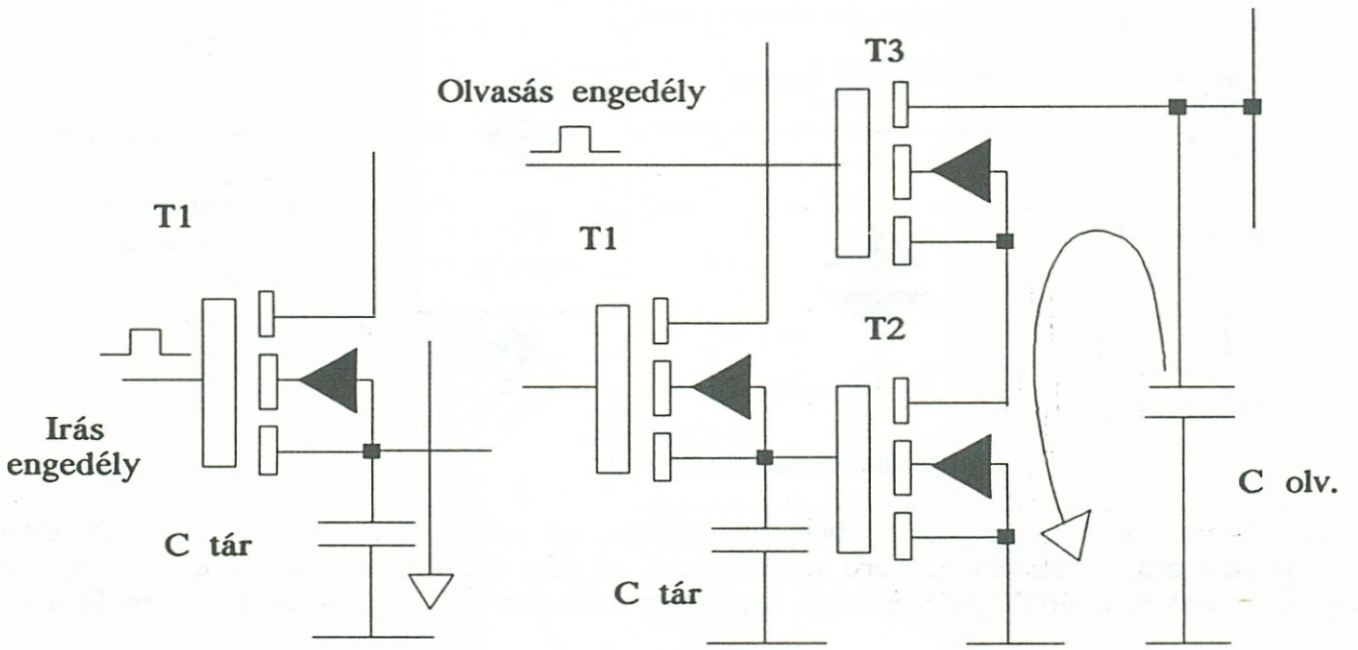
Statikus tárcella megvalósítása D tároló segítségével.

Az X és Y rekeszkiválasztó jelek magas szintje esetén a bemenő ÉS kapun megkapjuk a cella kijelölő jelet, ahonnan tudhatja a tárolócella, hogy neki szól az üzenet. Ha az alacsony szintre aktív $\overline{WE}$ jelet is kiadjuk, akkor az is kiderül, hogy írásra kell felkészülnie.Ezt a jelet invertáljuk és a cella kijelölő jellel együtt egy újabb ÉS kapura vezetjük, amelynek kimenete a D tár C órajel bementéhez csatlakozik. Ha most az adat megjelenik az adatvonalon (mint tudjuk H szintű órajel esetén), akkor az beíródik a tárba.Az olvasás során az alacsony szintre aktív $\overline{OE}$ invertáljuk, majd a D tár Q kimenetével és a cellakiválasztó jellel együtt egy hárombemenetű ÉS kapura csatlakoztatjuk, melynek kimenetéről az információ kiolvasható.

5.2.8. A dinamikus RAM-ok

Azokat a tárolókat nevezzük dinamikus RAM-oknak, amelyek a tárolt információt külső segítség nélkül csak néhány ms időtartamig tudják megőrizni. Az ilyen eszközökben mielőtt az adatok elvesznének, azokat újra kell írni, frissíteni kell a tár tartalmát.

Az eszköz működési alapötlete azon alapul, hogy ha egy kondenzátort feltöltünk,akkor az megőrzi a töltését külső behatás nélkül is A felvitt töltés feszültséget hoz létre a kondenzátor kapcsain, amit megfeleltethetünk a logikai jelszinteknek.Ezen elem



5.16. ábra

Dinamikus tárolócella kialakítása három MOS tranzisztor segítségével.

felhasználásához azonban két problémát kell megoldanunk: a kondenzátor megfelelő időben történő feltöltését (beírás), illetőleg annak vesztesség nélkül történő kiolvasását.

A problémák megoldása a következő! Ha a C kondenzátorral sorba kötünk egy MOS tranzisztort, mégpedig úgy,hogy a kapura vezetjük az írás engedélyező jelet, a nyelőre pedig a beírójelet, akkor a kondenzátort a megfelelő időben feltölthetjük.

Az írás engedélyező jel hatására nyílik a tranzisztor és a beírójel segítségével a kondenzártorfeltölthető.

A második probléma megoldásához az eddigi áramkört kissé át kell alakítanunk.Ehhez két, sorba kapcsolt MOS tranzisztorra és egy **olvasó kondenzátorra** van szükségünk az ábrának mefelelően.

Az olvasáshoz először fel kell töltenünk az olvasó kondenzátort. Az 3. számú MOS tranzisztor bemenetére csatlakozó olvasás engedélyezés hatására, ha $C_{tár}$ kondenzátor fel van töltve kinyit a 2. MOS tranzisztor, amelynek segítségével módosulhat az olvasó

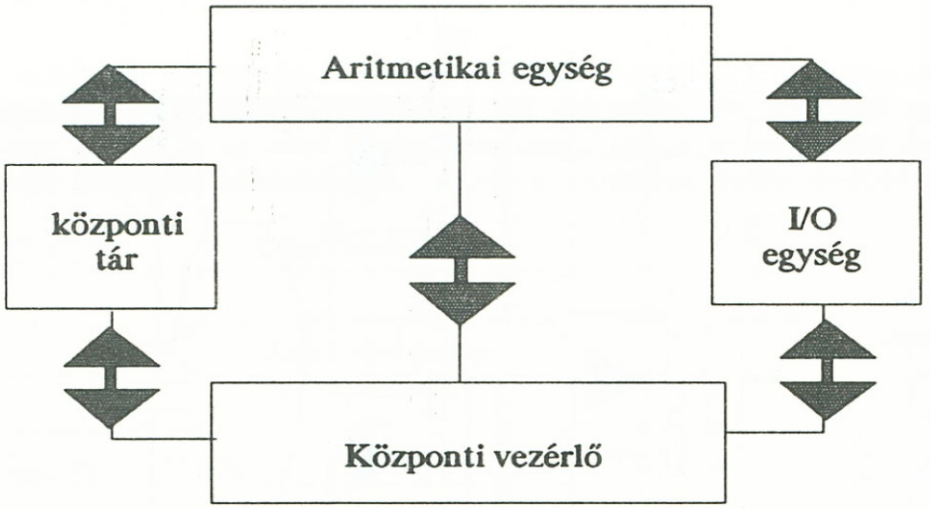
kondenzátor állapota. Ha a $C_{tár}$ fel volt töltve, akkor az olvasó kondenzátor kisül a tranzisztorokon keresztül, feszültségszintje leesik, ha viszont a $C_{tár}$ üres volt, akkor a 2. tranzisztor nem nyit ki,tehát az olvasó tranzisztor állapota változatlan marad.

Mint látható az olvasó kondenzátor állapota éppen ellentétesen alakul, mint a tárolást végző kondenzátor tartalma. Ezen egy inverter közbeiktatásával segíthetünk.

Az eddig elmondottak alapján még nem derült ki, hogy miért kell az áramkört frissíteni,hiszen működése meglehetősen stabilnak látszik.Nos ennek az az oka, hogy nem különálló kondenzátorokat alkalmazunk valójában, hanem a MOS tranzisztorok járulékos kapu - nyelő, kapu - forrás kapacitásaikban történik az információ tárolása,amelyekben a tárolt energia a szivárgási áramok miatt csak rövid ideig őrizhető meg. Ezzel jelentős hely takarítható meg a szilíciumlapka felületén, ami megnöveli a memória kapacitását.

5.3. A központi egység

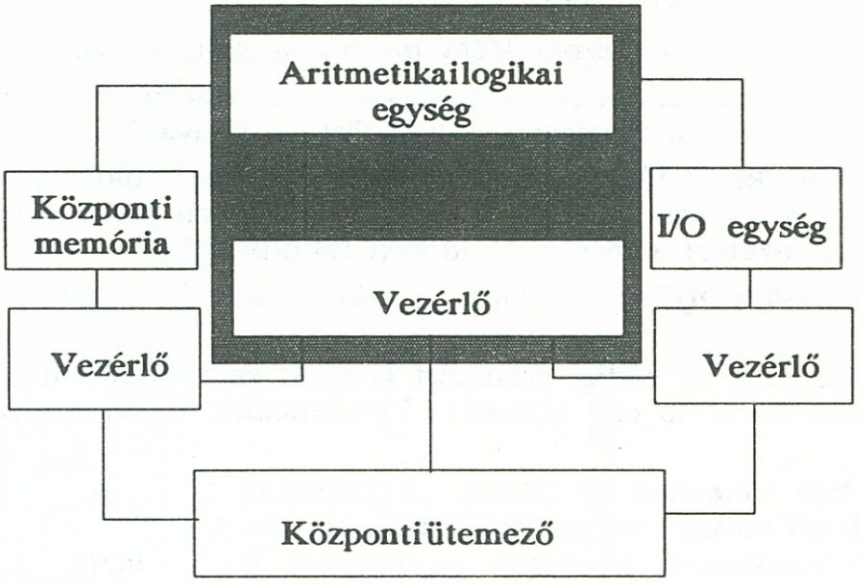
A központi egység (mikroprocesszor) fogalmának kialaulása az ötvenes évek végére nyúlik vissza. Az 1950-es évek elején a számítógépek felépítése egyetlen programvezérelt gépen alapult.Ennek lényeges jellemzője volt,hogy csupán egyetlen vezérlő alrendszert tartalmazott,amelynek az volt a feladata,hogy vezérelje és időben összehangolja a rendszer



5.17. ábra

Az első számítógépek blokkvázlata.Egyetlen vezérlő áramkör hangolta össze a rendszer működését.

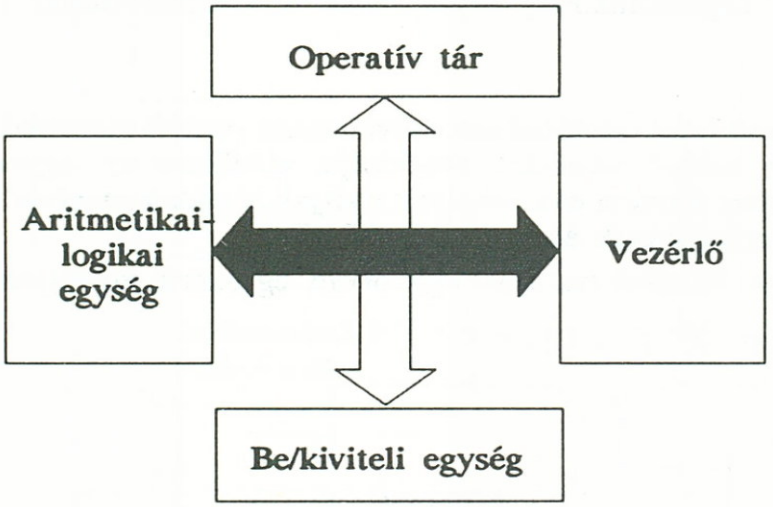
tagjai között folyó adatcserét.Ebben az időben az aritmetikai egység csak az elemi matematikai műveletek elvégzésére szorítkozott. (A gép ilyen felépítésének az volt az oka, hogy a regiszterek előállítási költsége igen nagy volt,ezért a részegységeket nem láthatták el vele.)



5.18. ábra

A központi egység az aritmetikai-logikai egység és a vezérlő egybeolvadásával született meg.

Az 50-es évek végén a vezérlő egységek decentralizálása volt megfigyelhető. Ez azt jelentette,hogy minden részrendszert különálló vezérlővel láttak el. Ezen a ponton alakul ki a központi egység, a CPU (Central processor unit) fogalma. Ez első közelítésben vezérlővel ellátott aritmetikai logikai egységet jelent.



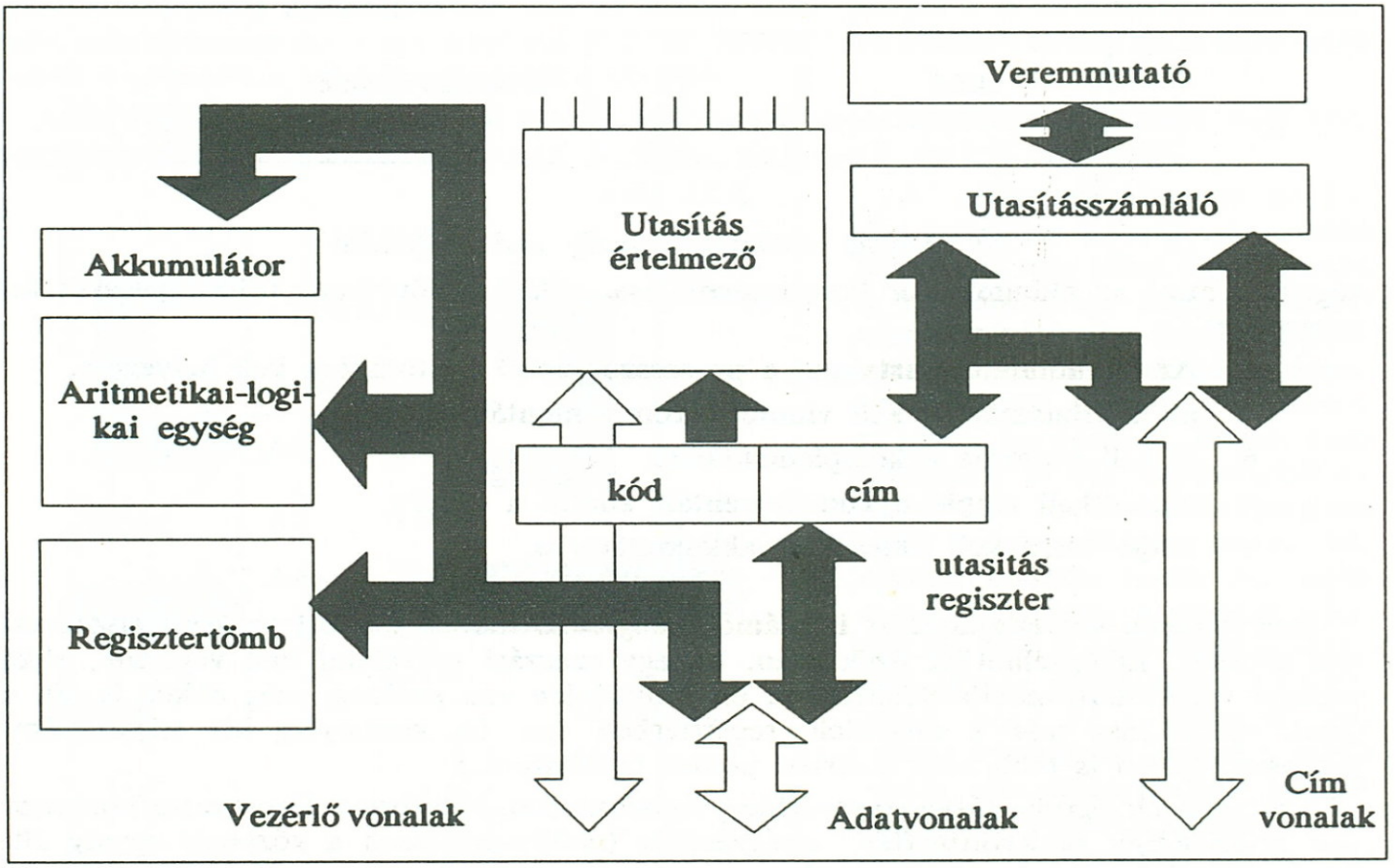
5.19. ábra

A központi egység vázlata

Ma, ha közelebbről megvizsgálunk egy központi egységet észre kell vennünk,hogy a felépítés az idők folyamán megváltozott.Jelenleg központi egységen a számolómű, a vezérlő, az operatív tár és a be/kimeneti áramkörök célszerűen együttműködő rendszerét értjük.

- A **számolómű** (ALU) végzi az aritmetikai logikai műveletek végrehajtását.
- Az **operatív tár** az adatok gyors elérését teszi lehetővé.

A **be/kiviteli egység** tartja fenn a kapcsolatot a processzor és a külvilág között.



5.20. ábra

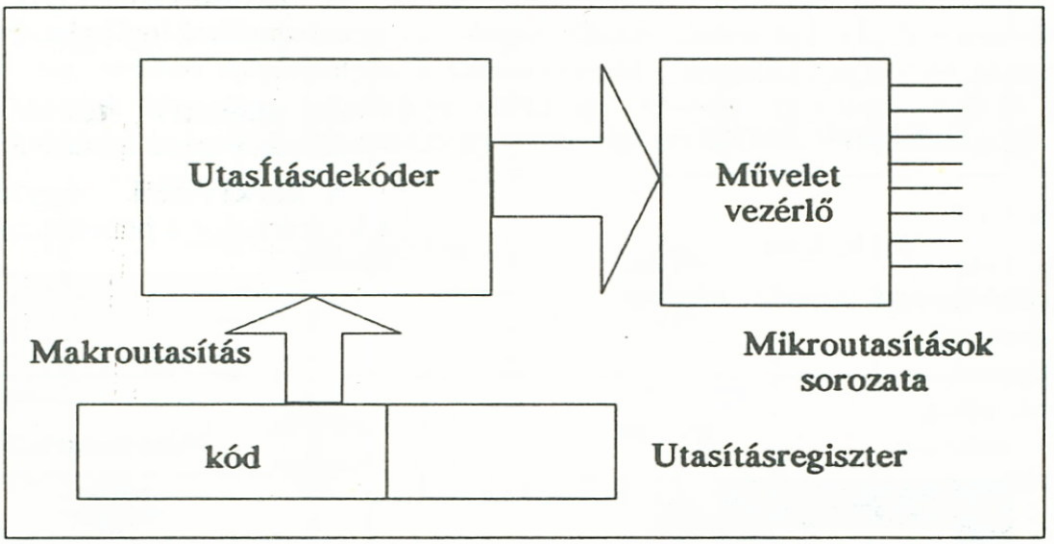
A központi egység felépítése. A sötét nyilak a belső adatutakat, a fehérek a külvilág felé haladó információ szállítást mutatják.

» A vezérlőmű értelmezi az utasításokat és előállítja a központi egység összehangolt működését kiváltó jeleket.

Az egyetlen áramköri morzsán kialakított központi egységet szokás mikroprocesszornak is nevezni.Vizsgáljuk meg részletesebben a központi egység felépítését! Az aritmetikai egység és a tár felépítéséről már van fogalmunk.Folytassuk most a vizsgálatainkat a vezérlő áramkör tulajdonságaival.

A **vezérlőmű** utasításregiszterből, utasításdekódolóból és a műveleteket vezérlő részekből áll.Feladata, hogy az utasításregiszterbe belépő utasítást dekódolja, előállítsa az egyes műveleti kódszavakat és olyan vezérlőjeleket állítson elő, amelyek nyitják,zárják a megfelelő adatutakat, működtetik és ellenőrzik az irányításuk alatt álló egységeket.

A vezérlőmű feladata erősen összetett. Például ha csak egy olyan egyszerű műveletet



5.21. ábra

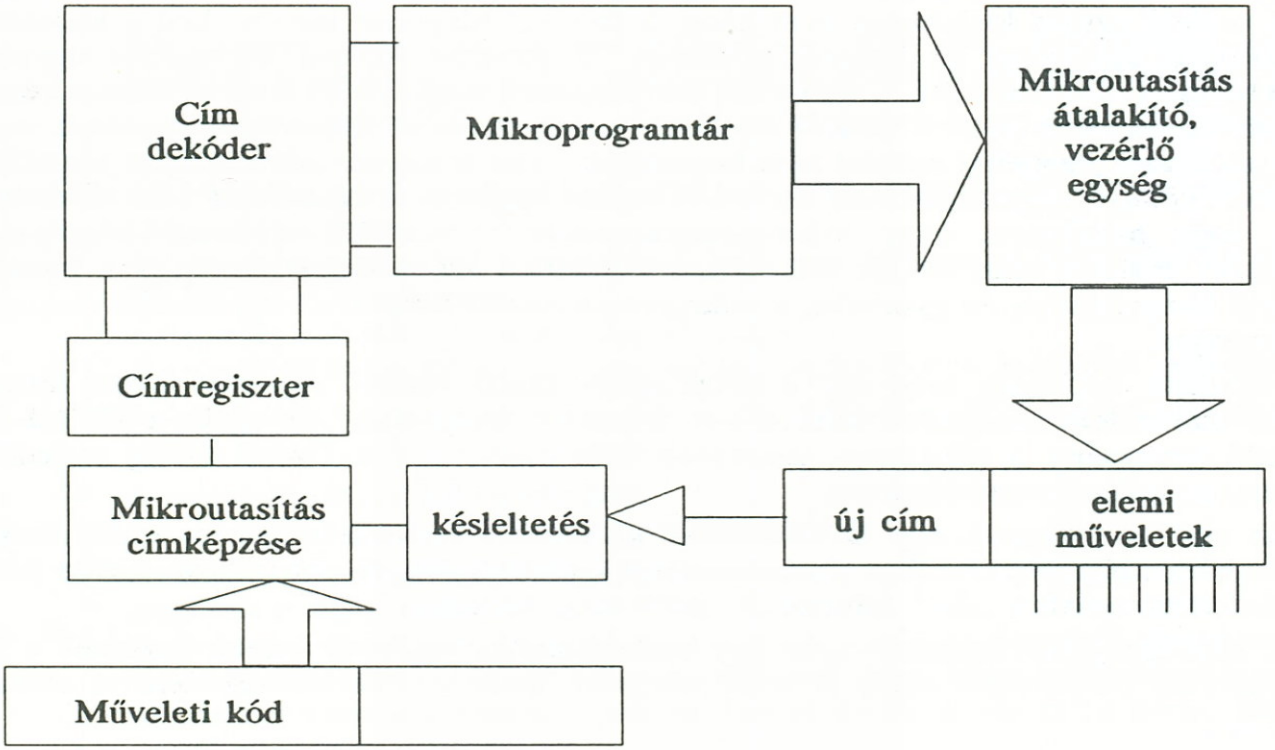
Az utasítás értelmező egység elvi felépítése

végzünk, mint az akkumulátor komplementálása, akkor a következő főbb lépéseket kell vezérelnie:

» Az akkumulátor tartalmát a processzor belső adatbuszára kell helyeznie,
» az adatbuszról be kell vinnie a komplementálóba,
» el kell végeznie a komplementálást,
» vissza kell vinnie a komplementált adatot a buszra,
» majd vissza kell töltenie az akkumulátorba.

Ezen lépések végrehajtásához is számos, megfelelő időben kiadott vezérlő jelsorozatra van szükség. Elképzelhetjük ezek után, ha egy szorzási műveletet kell végeznie, akkor mennyi vezérlőinformáció előállítására és kiküldésére van szükség, még akkor is, ha az egyik operandusz már a megfelelő regiszterben van. (A viszonylag kis teljesítményű processzorokban is több száz vezérlési pontot találhatunk.)

Azutasításokvégrehajtásához a vezérlőegységben kialakított információsorozatra (bitsorozat) van szükség.Egy **makroutasítás** elvégzéséhez (makroutasításon a központi egység által értelmezhető, hagyományos értelemben vett utasítást értünk, amely az utasításregiszterbe kerül és azt a vezérlő feldolgozza. Pl: összeadás, kivonás, komplementálás, mozgatás, betöltés stb.) nemcsak egy bitsorozat kiküldését, hanem az utasítástól függően több óraciklusón keresztül biztosított különböző vezérlő jelsorozat kiadását kell biztosítani. Ezt a bitsorozatokból álló információt **mikroprogramnak** , a mikroprogram elemeit jelentő

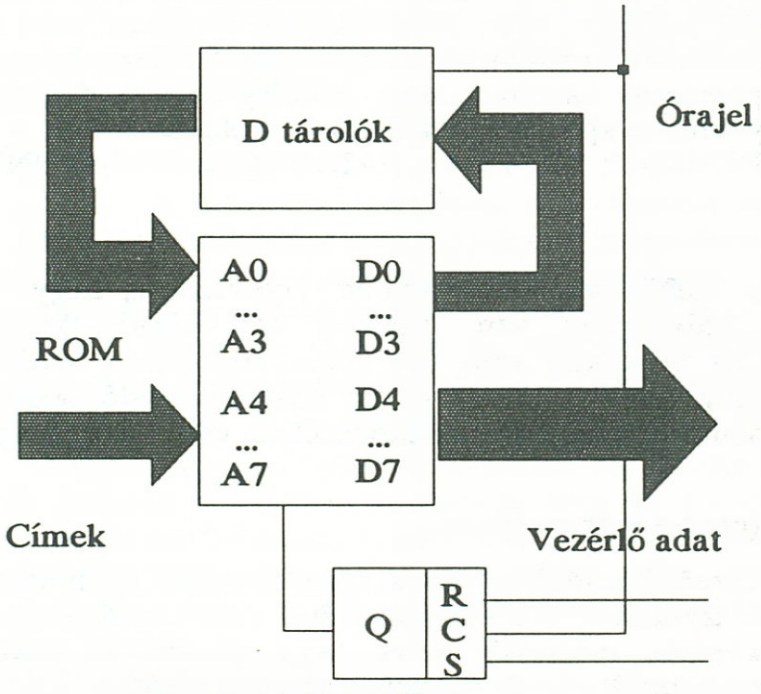


5.22. ábra

Mikroprogramozott vezérlő egység elvi felépítése.

bitsorozatot pedig **mikroutasítás** nak nevezzük. Ezek szerint a makroutasítás nem más, mint mikroutasításoknak egy meghatározott sorrendben összeállított csoportja.A mikroprogramot a processzor vezérlő egységében tároljuk.

Attól függően, hogy a vezérlő egység mikroprogramjai megváltoztathatók-e vagy sem, beszélünk mikroprogramozható, vagy fix illetve huzalozott vezérlő egységről.



5.23. ábra

Mikroprogramozott vezérlő

A mikroprogramozott vezérlő elvi felépítése látható a következő ábrán. Jól meg lehet figyelni rajta a mikroprogram működését. Az utasításregiszterbe érkező makroutasítás alapján a címképző áramkör elkészíti a megfelelő mikroprogram kezdőcímét. Ennek megfelelően a mikroprogramtárból kiválasztásra kerül a mikroprogram első mikroutasítása, amely két részből tevődik össze. Az egyik része szolgáltatja a vezérlő információt, amit a mikroutasítás átalakító egység alakít olyan formátumúvá, hogy az a vezérlési pontokhoz kiküldhető legyen.A másik újabb címinformációval szolgál, mégpedig megmutatja,hogy a kiindulási báziscímhez képest hol folytatódjon a mikroprogram végrehajtása, illetve végjelet szolgáltat a mikroprogram befejezéséhez.

A vezérlő egység kialakítása nem könnyű feladat. Alaposan ismerni kell a kialakítandó számítógép utasításait és hardverfelépítését. A tervezés lépései lényegében ugyanazok, mint bármely más áramköré. A választott műveleteknek megfelelően értéktáblázatot készítünk a vezérlőjelekből. Azokat a bináris értékeket, amelyeknek az adott ütemben nincs szerepe, vagy az általuk megjelölt eszközt nem használjuk olyan értékeket adunk,hogy a kialakítandó áramkör könnyen egyszerűsíthető legyen. A logikai függvények kialakítása után választanunk kell, hogy huzalozott, vagy **mikroprogramozott vezérlőt** kívánunk készíteni. Az áramkört csak ez után tudjuk kialakítani. Általában azt mondhatjuk, hogy a huzalozott vezérlő bonyolultabb, de gyorsabb, a mikroprogramozott lassabb de egyszerűbb elektronikát eredményez.

Példaként tekintsük meg egy mikroprogramozható vezérlő részletét. Mint látható a ROM kimenetein megjelenő adatok két feladatot szolgálnak. Az egyik a D0-tól D3-ig terjedő tartomány a következő címzésben vesz részt, míg a D4-től D7-ig terjedő tartomány szolgáltatja vezérlőjeleket.

Az első adatcsoport egy D tárolókból kialakított áramkörbe jut és ezért csak egy óraimpulzussal később kerül a tár bemenetére a címinformáció. Az mikroprogram kezdetét a felső címvonalakra adott információ jelöli meg A4-től A7-ig.

A mikroprogram indulását azaz egy makroutasítás végrehajtásának kezdetét a ROM engedélyező bemenetére adott H szint váltja ki, amit a C tároló vezérlésével oldhatunk meg.

A mikroprogramozhatóság elve egyre inkább teret hódít, hiszen így nagyobb szabadságot kap a felhasználó elképzeléseinek megvalósításában. Persze nemcsak ez indokolja a vezérlő áramkör ilyen jellegű kialakítását, hanem az az igény is, hogy a felhasználó saját maga igazíthassa a feladatnak megfelelően a processzor kapacitását.A gyártók az úgynevezett bitszelet processzorok kialakításával segítettek ezen a gondon. Ezek nagy sebességű LSI áramkörök, amelyek a processzornak csupán csak egy szeletét képviselik. Általában a CPU regisztereinek és aritmetikai-logikai egységeinek egy 2 vagy 4 bites szeletét integrálják egy áramköri lapkára.A vezérlő egység ugyancsak egy vagy több áramköri lapon helyezkedik el. Ezért ezen elemekből, őket megfelelő módon összekapcsolva, tetszőleges szóhosszúságú processzorok alakíthatók ki. Egy ilyen megoldáshoz azonban már csak mikroprogramozott vezérlő illeszthető. Hiszen huzalozott vezérlőlogikával tetszőleges igényeket kielégíteni nem lehet. (A bitszelet processzorok hátránya, hogy jelenleg drágák, de jelentős szoftverköltségeket takaríthatunk meg velük a speciális feladatok megoldása során.
Ilyen bitszelet processzorok TEXAS SN74S842; SN74S841; FAIRCHILD F 9400; INTEL I 3000.)

Ezek után nézzük meg a processzor működését részletesebben. Nyilvánvaló, hogy egy elemi számítógép megvalósításához szükségünk van legalább egy külső tárra,ahol elhelyezhetjük az adatainkat, illetve az adatokon való operációk végzéséhez szükséges utasítássorozatokat (programok). A számítógép feladata — az adatokon való operációk végrehajtása — csak a tár és a központi egység együttműködésében valósulhat meg.

5.3.1. A tár és a központi egység kapcsolata

A rendszer működését egy utasítás végrehajtásának módjával szemléltetjük.Olyan utasítást választottunk, amely címkiegészítést is tartalmaz. Ha meggondoljuk a CPU-nak ebben az esetben háromszor kell a tárhoz fordulnia, mégpedig azért, hogy elhozza az utasítást, a címkiegészítést, illetve az operanduszt.Nézzük ennek részletes végrahajtását.

Az első ciklusban történik az utasítás felhozása. Ennek lépései a következök:

» Az utasításszámláló tartalma rákerül a rendszer címvonalaira és innen a memória megfelelőcímregiszterébe.
A tárban dekódolódik és ezzel kiválasztásra kerül a megfelelő tárrekesz.

Kis késéssel ezek után olvasás engedélyező jelet küld a processzor, aminek hatására a memória kimeneti adatpufferébe íródik a rekesz tartalma.

» A memória pufferéből az adat rákerül az adatvezetékekre és elindul a központi egység felé.
A CPU-ban, mivel az első ciklus mindig utasítás felhozás, az irányító elektronika az utasításregiszterbe tereli.
Az utasításregiszterből a vezérlőmű kiolvassa és azonnal megindul az utasítás értelmezése,visszafejtése.
Az utasításszámláló tartalma eggyel megnő.
A vezérlőjelek beállítják az utasításnak megfelelően a hátralévő menetrendet.

» A harmadik ütemben történik meg a tár frissítése. (Pl:Az adat visszaíródik a memória pufferéből a tárrekeszbe.)

» A negyedik ütem kihasználatlan.

A második ciklusban kerül behívásra a címkiegészítés.

» Az utasításszámláló tartalma rákerül a rendszer címvonalaira és innen a memória megfelelőcímregiszterébe.
A tárban dekódolódik és ezzel kiválasztásra kerül a megfelelő tárrekesz.
Kis késéssel ezek után olvasás engedélyező jelet küld a processzor, aminek hatására a memória kimeneti adatpufferébe íródik a rekesz tartalma, ami most a címkiegészítést tartalmazza.

» A tárból érkező adatot a CPU címként értelmezi és az eltolási regiszterbe továbbítja. (Kisebb processzorok esetén az eltolási regiszter és az utasításregiszter összeolvad.)

» A harmadik ütem a memória frissítésére fordítódik.

» A negyedik ütemben ismét címgenerálás történik, az utasításszámláló tartalma eggyel megnövekszik.

A harmadik gépi ciklus a végrahajtó ciklus.

» Az utasításszámláló tartalma az eltolási címmel megnövekszik és ez kerül a címvonalra. Erről a címről kerül fel a tárból az adat, az előző két ciklusnak megfelelően.

» Mivel a felhozott adat most egy szám, a vezérlő az akkumulátorba fogja továbbítani.

» A következő két ütem a processzor szempontjából üres.

Mint látjuk több lépésben tudjuk csak megoldani az adatfelhozását a memóriából,különösen akkor,ha a feladat címgenerálást is tartalmaz. Természetesen ahány utasítás, annyiféle elemi lépésre bontható a feladat.Gondoljunk csak arra az esetre, ha például programunkba szubrutin hívásokat iktatunk be. Ekkor szerephez jut a veremmutató is. Hiszen ez a regiszter fogja meghatározni,hogy a rendszer működtetéséhez szükséges regisztertartalmakat hova mentettük el és honnan kell folytatni a program végrehajtását.

A számítógépes rendszerek működése szempontjából fontos a gép műveleti sebessége. Ezt jelentős mértékben határozza meg az az időtartam, ami a processzor két egymást követő tárhoz fordulása között telik el. Ezt az időtartamot a továbbiakban **gépi ciklusnak** nevezzük. Mint láttuk a gépi ciklus több elemibb ütemre bontható. Általában azt mondhatjuk, hogy a mikroszámítógépek világában a gépi ciklus időtartama egy rendszerre nézve állandó.

Az előbbiekben nagyvonalakban láttuk, hogy a központi egység hogyan kommunikál a tárral, hogyan hajtja végre az utasításokat. Viszont arról ez ideig nem szóltunk, hogy a CPU hogyan szerez tudomást az utasítások alapján végrehajtandó feladatokról. Ahhoz, hogy ez kiderüljön szólni kell az utasítás gépi megjelenési formájáról.

Kezdjük ennek ismertetését egy kis történeti áttekintéssel. Az 1960-as években, amikor a tranzisztoros egységek alkalmazása elkezdődött, a processzor és egyben a memóriarekesz

szóhossza 18 - 64 bitig terjedt. A különböző cégek géptípusainak utasításkészlete, s egyben ezek kódolása is gépről-gépre változott. Ez meglehetősen kaotikus állapotot jelentett. Lényeges változást 1965 hozott, amikor az IBM kibocsátotta a 360-as gépcsaládot. Megállapotdtak abban, hogy a processzor és a memória szóhossza is 32 bit szélességű legyen. Definiáltak 150 féle utasítást. (adatmozgató,vezérlésátadó, logikai, aritmetikai, stb.) Lerögzítették ezek 8 bites kódját (Ez időtől datálódik a bájt elnevezés is.) Ezen gépek ciklusideje 0,5 mikrosecundum volt, ami azt jelentette, hogy két tárhozfordulás 1 mikrosecundum alatt képes volt műveleteket végezni. (Tegyük fel ugyanis,hogy az egyik operandusz már a tárban van, ekkor szükséges még egy utasítás és a második operandusz memóriából történő felhozása, aminek ideje 0,5 - 0,5 mikrosecundum.) Vagyis tömören fogalmazva egy másodperc alatt egymillió műveletet volt képes végrehajtani. Szerencsére igen drága eszköz volt és így kevés példányt tudtak belőle eladni. A szerencsét persze úgy értettük,hogy a kis forgalom arra ösztönözte az IBM kutatóit, hogy kifejlesszék a gép kistestvéreit,amelyek már csak 8 bites szóhosszal dolgoztak. A hangsúly viszont azon van,hogy ugyanakkor megtartották ugyanazt az utasításkészletet, mint amit a nagy elöd használt. Természetesen ez azt is jelentette, hogy ha a gép aritmetikai művelet végrahajtására kap parancsot, akkor az utasítás kihozáson kívül még három memóriaciklusra van szükség a művelet végrehajtásához. (4 * 8 bit). Így tehát lassabbá vált a gép működése, de. olcsóbb is lett. Ezt követően a legtöbb gép átvette az öteltet, legalább is azt, ami az utasítások méretét illeti.

Géptípus	Év	Szóhossz bitben	ciklusidő mikroszekundumban	Memória Kbájtban
IBM 360				
/30	1965	8	0,75	
/40	1965	16	0,63	
/50	1965	32	0,5	
/65	1967	63	0,2	
EC				
R-20	1971	8		
R-30	1972	16		
R-40	1973	32		
R-50	1973	64		
IBM 370				
/135	1972	16	0,46	512
/145	1971	32	0,27	2048
/158	1973	32	0,20	2048
/168	1973	64	0,8	6000
Mikroprocesszorok				
INTEL 8080	1974	8	2,00	64
INTEL 8086	1978	16	0,5	1024
INTEL 80286	1984	24		16 Mbájat
INTEL 80386Sx	1988	32		4 Gbájt
Intel 80486Sx	1991	32		4 Gbájt
INTEL 80586	1992 ?	?		?

5.3. táblázat

Központi egységek megjelenése a piacon.

Az utasítások gépi megjelenési formája

8 bites számítógépet feltételezve nyilván az lenne a leggazdaságosabb, ha az utasítások hossza egy bájt lenne, azonban a műveletek operanduszcímére való hivatkozás miatt a számítógépek valamennyi típusában találkozunk 2, illetve 3 bájt hosszú utasítással. A nyolc bit mindegyikéhez való hozzárendelés igen változatos képet mutat. Szemléltetésül nézzünk meg néhányat!

» Akkumulátor hivatkozású, egyoperandusos utasítás.

Ha csak egy akkumulátor van a rendszerben, akkor külön hivatkozás csak az átvitelbit értékének figyelembe vételére, vagy annak figyelmen kívül hagyására van.

» Regiszter hivatkozású műveletek.

Ha az egyik operanduszt az egyik regiszter,a másik operanduszt az akkumulátor tárolja, akkor alkalmazzuk ezt a típust. Például 8 regiszter esetén az utasítás kódja ilyen formában tárolódik:

*	*	*					

ahol *-gal jelöltük meg a regiszterek címzésére szolgáló biteket.A szabadon hagyott mező az utasítás kódját tartalmazza.

*	*	+	+				

A regiszterek közötti adatmozgást a következő címzési módon oldják meg:
ahol *-gal jelöltük meg a forrás, +szal a célregiszterek címzésére szolgáló biteket.

» Tárhivatkozásúutasítások.

*	*	+	+				

x	x	x	x	x	x	x	x

Az esetek többségében ezek két bájtosak.A második bájt egy relatív címet tartalmaz valamely bázisregiszterhez képest. Az első bájt felső két bitjét (*) a közvetett illetve közvetlen hivatkozás jelzésére,a második két bitet (+) a báziscímre történő utalásra tartjuk fenn.Az első bájt maradék négy bitje hordozza az utasítást.

» Teljes címzésű utasítások.

					*	*	*

x	x	x	x	x	x	x	x

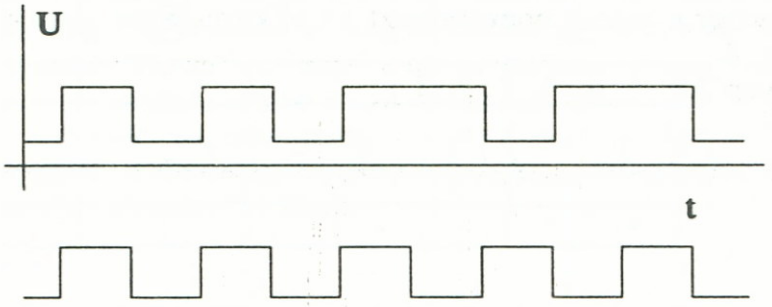
Ezek segítségével általában a teljes címezhető terültre tudunk hivatkozni.Általában 3 bájt hosszúak,amelyből az első az utasítás kódját, a második kettő pedig a teljes címrészt tartalmazza. Az első bájt 5 bitje hordozza az utasítás kódját, a maradék három feltételek előírására szolgál.

x	x	x	x	x	x	x	x

6. Sínszervezésű rendszerek

Az előző fejezetben megismerkedtünk a processzorral és a memóriával. Kérdés most már az, hogy ezen egységek között milyen fizikai elemek segítségével bonyolódik le az adatforgalom, és hogyan lehet ezen egységeket úgy kibővíteni, hogy használható számítógép váljon belőlük. Elsőként ismerkedjünk meg azokkal az elemekkel, amelyeken keresztül az adatforgalombonyolódik.

6.1. A rendszersínek

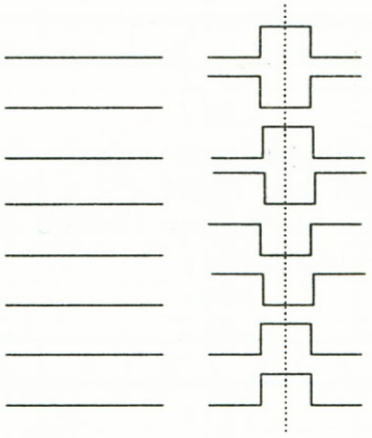


6.1. ábra

Órajellel szinkronizál soros adatok

A számítógépek belsejében az információ, mint tudjuk, bináris formában továbbítódik.Ezek az adatok két fajta úton juthatnak el céljukhoz: vagy soros, vagy párhuzamos vezetékkötegen.

A **soros** adattovábbításnak az a lényege, (6.1. ábra) hogy a biteket időben egymás után, egyetlen vezetéken szállítjuk.Ez jelentős időráfordítással jár, azonban lényegesen kevesebb vezetéket kell felhasználnunk hozzá. (Egyetlen vezetékpár elegendő.)

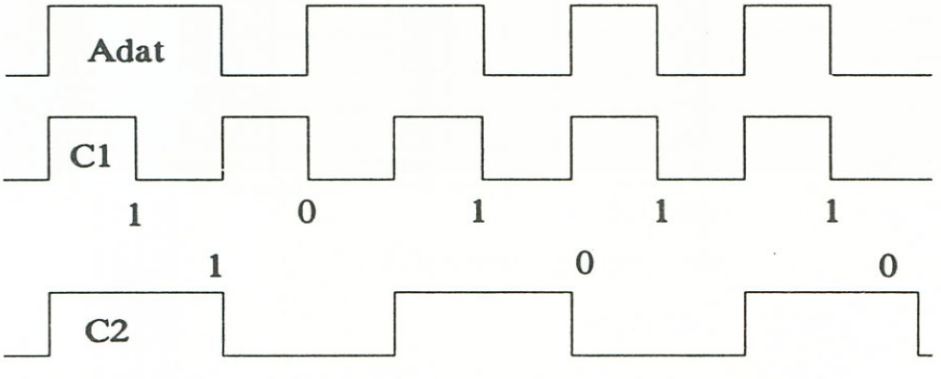


6.2. ábra

Párhuzamosan továbbított adatok, egy időben egyszerre jelennek meg.

A **párhuzamos** továbbítás esetén (6.2. ábra) minden bitet egyszerre, egy erre a célra szolgáló párhuzamos vezetékkötegen keresztül szállítunk. Ez természetesen azt jelenti, hogy időben meggyorsul az adatok átadása, de annyi vezetékre, pontosabban annál eggyel többre van szükség a feladat megoldásához, mint amennyi bitet egyszerre kívánunk szállítani.

Az információt átvivő párhuzamos vezetékek összességét szokás **busznak** vagy sínnek nevezni. A sínen egyetlen órajel periódus alatt jelenik meg az

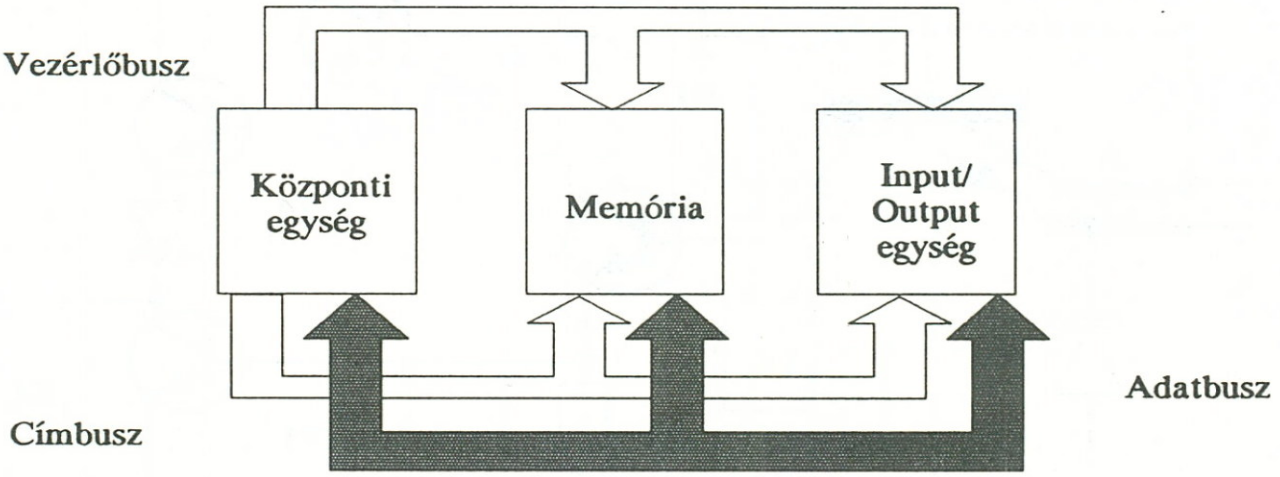


6.3. ábra

Különböző órajel esetén ugyanaz a jelsorozat más és más értéket vesz fel.

ábrázolni,átvinni kívánt adat. Mind a két vázolt megoldásnál gondoskodni kell arról,hogy időbeni ütemezést készítsünk,szinkronizáljuk a rendszer működését. Ez az ütemjel nagyon fontos, hiszen e nélkül nem tudnánk azonosítani, milyen kód tartozik a jelsorozathoz. Mint a mellékelt ábrán (6.3. ábra) látható, soros esetben a szaporább óraimpulzusok megváltoztatják a jelek számát és értelmét,de párhuzamos esetben sem lehetnénk biztosak a bitsorozat jóságában, hiszen ütemezés nélkül nem tudnánk, hogy az adott vonalon nem változott-e meg az információ. Ez az oka többek között annak ,hogy a számítógépek belsejében mindig találunk egy ütemjel, vagy más néven **órajelgenerátort.** Összefoglalva: az adatok szállítása időben szinkronizált párhuzamos vagy soros vonalakon történik. (A működési sebességét vizsgálva hasonlítsuk össze például egy XT, egy AT 286, vagy egy AT 386-os számítógépet. Mi az oka a sebességek különbözőségének?)

A nagy átviteli sebesség elérése érdekében a számítógépekben a sínszervezést valósítják meg, bár ez bonyolultabbá teszi az elemek kapcsolatát. Nézzük meg, hogyan is néz ki egy ilyen számítógépes rendszer! (6.4. ábra)



6.4. ábra

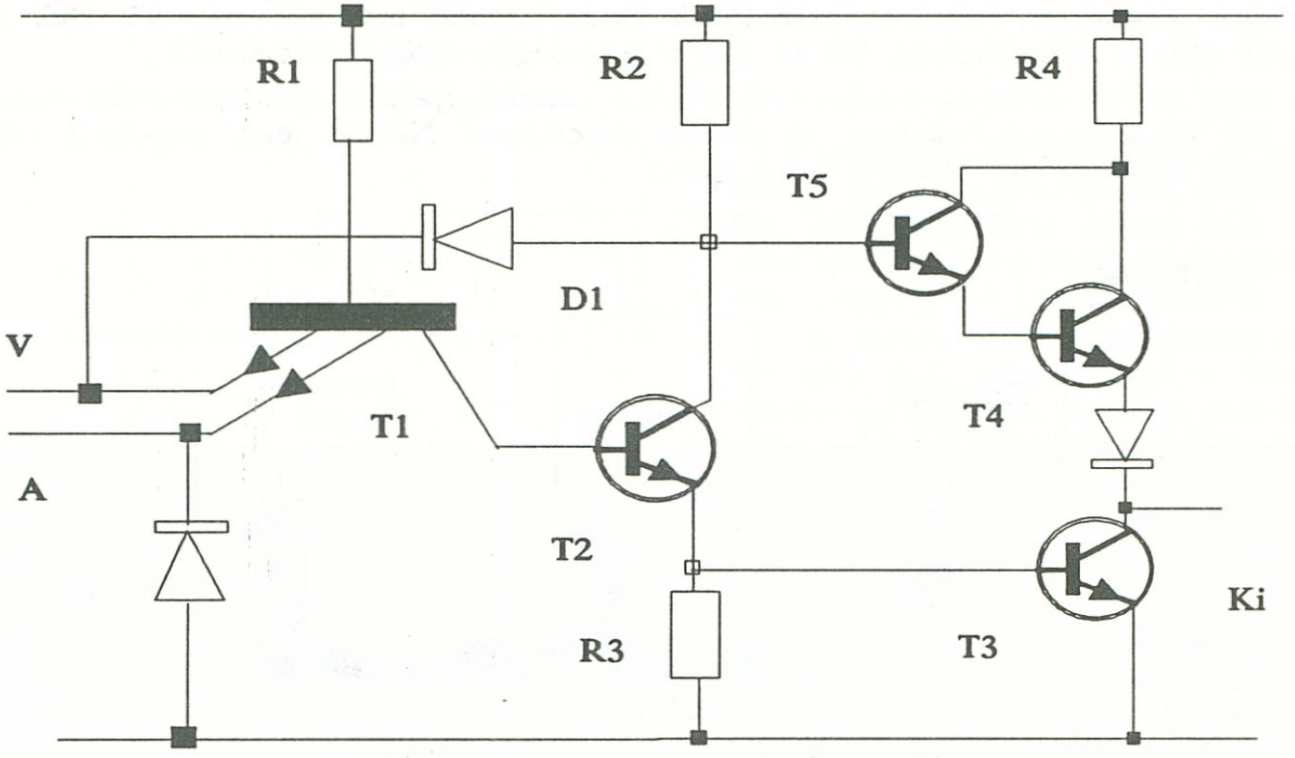
A sínszervézésű mikroszámítógépek modellje.

Mint látható három különálló vezetékköteget különböztethetünk meg benne. Ezek a következők:

- A **címsín** - egyirányú jelvezetékek csoportja.Feladata meghatározni a kommunikáció útját, azaz megnevezni azt az eszközt, amelyikre az adatátvitel vonatkozik. A vezetékek számát a címezhető egységek száma szabja meg.
- Az **adatsín** - kétirányú jelvezetékek csoportja. Feladata, hogy a processzor által engedélyezett és a címsín által kiválasztott külső hardver elemek között kicserélésre kerüljön az információ. Tehát ez a vezeték szállítja az adatokat a rendszer elemei között.A vezetékek számát az egyszerre átvinni kívánt adat vagy másként a számítógép szóhossza határozza meg.
- A **vezérlősín** - egyirányú jelvezetékek csoportja. Feladatának meghatározásához ismernünk kel az adatáramlás lehetséges eseteit.
  a.) A CPU adatokat fogad a tárból.
  b.) A CPU adatokat ír a tárba.
  c.) A CPU adatokat fogad egy külső eszközről.
  d.) A CPU adatokat küld egy külső eszköznek. (Külső eszköz alatt egy input/output egységet értünk.)
  A vezérlősín feladata tehát az, hogy megmondja a lehetséges esetek közül melyik az, amelyik éppen most zajlik.

6.2. A háromállapotú logikai kapuk

Az előbb felsorolt rendszersínek logikai kapukban végződnek, amelyeken keresztül tart kapcsolatot a sín az egyes egységekkel. A problémát az jelenti,hogy a sínre egyszerre több egység is kapcsolódhat, de általában ezek logikai szintje különböző. Ez nagyon könnyen bonyodalmakhoz vezet,mert a rendszer (sín) nem tudja eldönteni,hogy milyen állapotot vegyen fel. Ezért különleges, úgynevezett **háromállapotú kapukra** van szükség, hogy ezt a problémát megoldjuk.



6.5. ábra

A háromállapotú logikai kapu felépítése.

A feladat tehát az, hogy csak az az eszköz kapjon meg minden információt, amelyre az adatcsere tartozik, a többi választódjék le a megfelelő sínről. Az 6.5. ábrán ilyen logikai kaput láthatunk.Vizsgáljuk meg közelebbről az áramkör működését! A rajzon látható,hogy egy kibővített NAND kapuval van dolgunk. A D1 dióda illetve a T5 tranzisztor szerepel új elemként. Nos ezen módósítás segítségével, amint azonnal látni fogjuk elérjük célunkat.A kapu tehát úgy fog viselkedni, hogy a V vezérlő bemenet 0 logikai szintje esetén a kapu kimenete áramot nem képes adni, de elnyelni sem tud. Úgy is fogalmazhatnánk,hogy a kapu kimenete lóg a levegőben, nincs egyetlen logikai állapotban sem.

Legyen V=0. Ekkor a D1 dióda kinyit (nyitóirányban van előfeszítve az R2 ellenálláson keresztül), rajta csak a nyitófeszültség, kb. 0,6 V esik. Ezért a T5 tranzisztor bázisfeszültsége kicsiny, így ez a tranzisztor lezárt állapotban van, ami miatt T4 is zárt állapotba kerül.

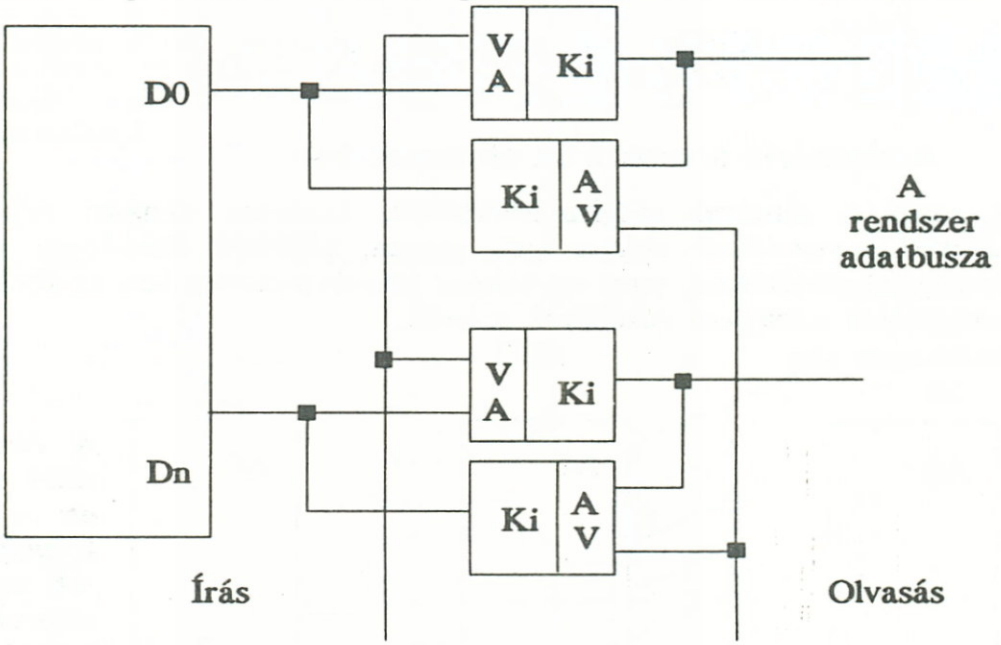
A D1 dióda nyitott állapota miatt a T2 tranzisztor is lezárt állapotba kerül, ugyanis a kollektora közel földpotenciálon van. Tehát az R3 ellenálláson nem esik feszültség (rajta nem folyik áram, mert T2 zárt), ami azt is jelenti, hogy a T3 tranzisztor is zárt állapotba kerül. Összefoglalva sem a T4, sem a T3 tranzisztor nem vezet, ami azt jelenti, hogy az áramkör kimenete a „levegőben lóg".

Ha most a vezérlő bemenetre 5 V-ot adunk, akkor a dióda zár állapotba kerül. Azt is mondhatnánk mintha ott sem lenne, így tehát a kapu közönséges inverterként

működik.(Kettőnél több emitteres bemeneti tranzisztor esetén háromállapotú NOR kapuval van dolgunk.)

6.3. A cím és adatbusz kezelése

A cím és adatbuszt a terhelő áramkörök nagy száma miatt erősíteni kell. (Lásd az integrált áramkörök fan out-ját.) Ha a számítógép nem zárt, akkor a címbuszt mindig erősítik. A feladatot lényegesen egyszerűsíti, hogy a busz egyirányú. E feladat elvégzésére az előbb tárgyalt kapukat alkalmazunk. Az adatbusz kezelése lényegesen bonyolultabb feladat, hiszen az adatok két irányban áramlanak, az egyik időpillanatban az egyik, a másik időpillanatban a másik irányba szállítják az információt. (A kétirányú erősitést tehát különböző időpillanatban kell megoldani. Ha a feladatot tovább elemezzük, akkor kiderül,hogy a két irány összhangban van azzal, hogy éppen milyen folyamatot hajt végre a számítógép. Ugyanis írás esetén a processzor felől (a processzor felől nézve), olvasás esetén a



6.6. ábra

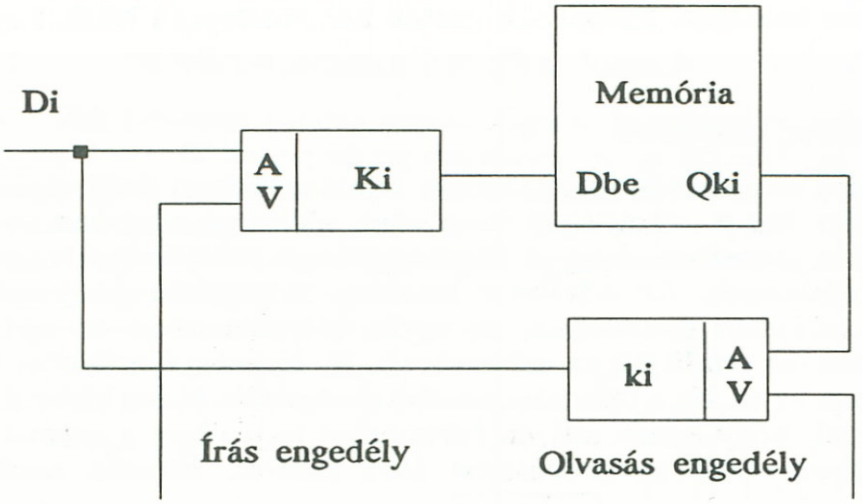
A kétirányú adatbusz kezelése.

processzor felé áramlanak az adatok. Ez azonnal megoldja a helyzetet, ugyanis írás illetve olvasás vezérlőjellel rendelkezik a processzor, amelyeket felhasználhatjuk az adatirány meghatározásához, hiszen ezek a jelek kölcsönösen kizárják egymást. Ha most a háromállapotú kapuk vezérléséhez ezeket a jeleket alkalmazzuk, akkor, amíg az egyik kapu vezetni fog, addig a másik lebegő állapotban van. A 6.6. ábrán láthatjuk, a probléma egy megoldását.

Ha a memóriánk dinamikus RAM, akkor az adat be- illetve kimenet különválasztása jelent problémát. (Ezen memóriák esetén különválasztják a beíró és kiolvasó vonalakat.)Ezek a vonalak természetesen nem köthetők össze, mivel előállhat olyan eset, hogy a bemeneten adat fogadása történik, ugyanakkor a kimeneten még ott van a régi is. Ez tönkre teheti a rendszert. A megoldást itt is a háromállapotú kapu alkalmazása kínálja. (6.7. ábra)

A busz erősítésére használható logikai áramkörök SN74LS245 illetve SN74LS244. Ezek 8 darab egyirányú, háromállapotú meghajtó kaput tartalmaznak. Léteznek kétirányú meghajtók is. Ilyen például az INTEL 8216.

Vannak olyan memóriaáramkörök is, amelyek időmultiplexelten kezelik a cím és adatvonalakat. (Erre takarékossági okokból van szükség, jelentősen csökkenthető az áramkör kivezetéseinek a száma. Az **időmultiplexelés** azt jelenti,hogy az adott kivezetés az egyik időpontban címvonalként, a másikban adatvonalként üzemel.) Ezekben az esetekben



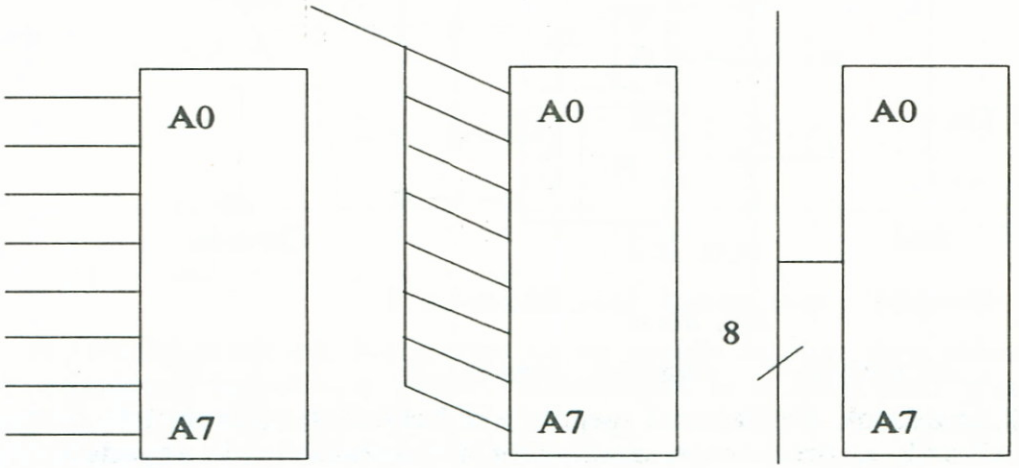
6.7. ábra

Dinamikus RAM különálló adat be illetve kimenetének a kezelése.

az olyan erősítő áramköröket kell alkalmazni, amelyek tároló elemet is tartalmaznak. Például az INTEL 8212 áramköre 8 bites tárolót is tartalmaz.

6.4. A memória áramkörök címkezelése

Mielőtt ezzel a témával megismerkednénk, tanácsos néhány rajztechnikai fogást bevezetni.Mivel a vezetékek száma igen magas, célszerű összefogni a vezetékeket és megfelelő rajzjelet alkalmazni, mert egyébként áttekinthetetlen lesz az ábránk.A vezetékeken célszerű megjelölni a leágazó vezetékek számát.



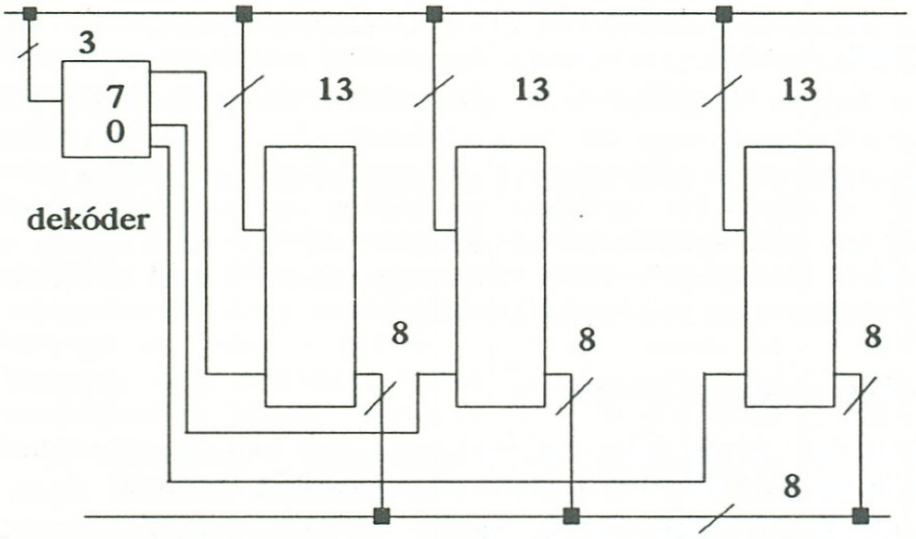
6.8. ábra

A cím- és adatvezetékek nagy száma miatt a rajzolást megkönnyíti, ha egybefogjuk ezeket a vezetékkötegeket és csak egy vonalat rajzolunk, amelyen felüntetjük a vezetékek darabszámát.

Mivel a memóriaáramkörök kapacitása igen változatos (1K*8, 2K*8, 16K*8, 32K*8 stb), a memória területet általában több integrált áramköri tok segítségével tudjukelőállítani. Ekkor azonnal felmerül a kérdés, hogyan lehet kiválasztani a kívánt memóriacellát, hogyan jelölhetjük ki pont azt a tokot, amelyikben majd helyetfoglal a kiválasztott cella. Legyen például az a feladatunk,hogy egy 64K-s területet állítsunk össze 8Kbájtos egységekből.

Ezen feladat megoldásához 8 darab memória áramkörre van szükség. Egy integrált áramköri tok megcímzéséhez 13 jelvezetékre van szükségünk, ugyanis a 13 vezetéken összesen 2^{13} jelkombinácíó alakulhat ki, ami pontosan a tár rekeszeinek számával egyenlő, ami egyben azt is jelenti, hogy minden cellát meg tudunk címezni a tokon belül. Ha azt is tudjuk,hogy a 64K-s terület címzéséhez 16 jelvezetékre van szükség,akkor azonnal kiderül,hogy 3 címvezeték felhasználható a memória IC-k azonosítására.

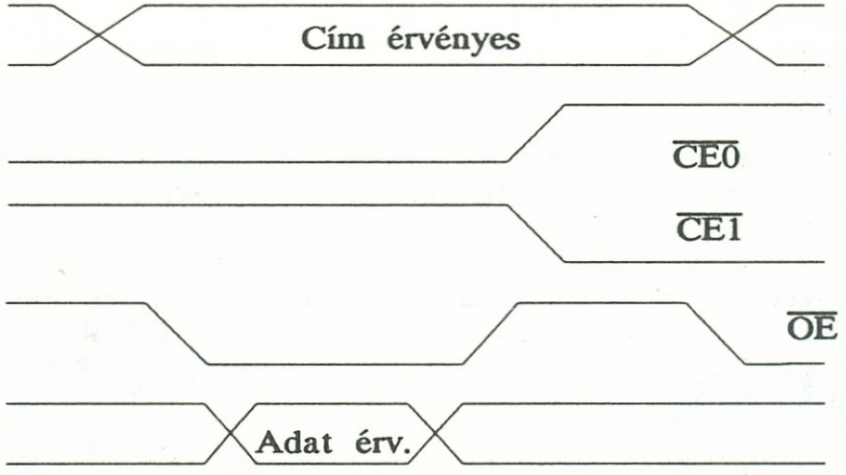
Az azonosítást egy 3-ról 8-ra dekóder segítségével oldhatjuk meg. A dekóder által létrehozott 8 vezeték az áramköri tokok CE csipp engedélyező bemenetéhez csatlakozik.Ez a megoldás azonban nem kielégítő,ugyanis ha a jelvezetékek váltanak előfordulhat,hogy az új memória már aktivizálja a kimenetét,mielőtt a másik lebegő állapotba váltana.Ez



6.9. ábra

Példa memória áramkörök címzésének megvalósítására.

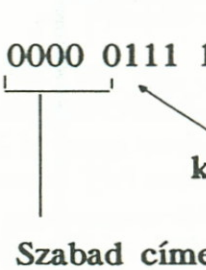
azt jelenti,hogy a busz vezetékeit két memória is igénybe akarja venni, ami miatt a kimeneti fokozataik meghibásodhatnak.Ez a veszély az olvasás engedélyező jelek alkalmazásávalelkerülhető.

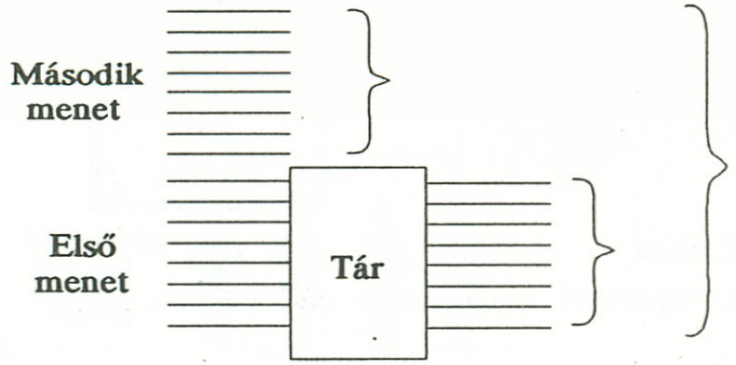


6.10. ábra

A teljes címdekódolás kiegészítése olvasás engedélyező jellel.

Az előző példában úgynevezett **teljes címdekódolást** végeztünk, vagyis minden címvezetéket felhasználtunk. Amennyiben kisebb memóriaterületet kell összeraknunk és nem használunk fel minden címvezetéket, akkor **részleges dekódolás**ról beszélünk. Ebben az esetben, mivel nem minden címvonalat veszünk figyelembe, több cím esetén is megszólalhat az áramkörünk. Például, ha 1K-s memóriákból akarunk összerakni 2 K-s memóriát, akkor ennek címzéséhez elegendő 11 címvezeték. Ebből 10 azonosítja a tokban lévő cellákat, a 11. pedig a két tok kiválasztására ad lehetőséget. Ha a processzorunk mondjuk 16 címvonalat tud kezelni, akkor a részleges címzés a 6.10. ábrának megfelelően





6.11. ábra

Időmultiplexelt címzés megoldásának vázlata. Első menetben csak az alsó, a második menetben csak a felső címrészt állítjuk elő. A teljes cím kialakításához az alsó címet tároljuk a felső megjelenéséig.

történik. Mivel a felső címbitek szabadon választhatók,ezért az áramkör például a 0010 0111 1111 1111 cím kiküldése esetén is kiválasztásra kerül a memória terület.

16 címvonal segítségével, mint tudjuk kiválasztásra kerülhet 64K-s memóriaterület. Ha ez nem elegendő, akkor is megvalósítható nagyobb terület kezelése,ha a címvonalakból több nem áll rendelkezésre. Ez esetben az úgynevezett időmultiplexelt megoldás visz célhoz. Ennek az a lényege,hogy a címet két lépésben (az időben egymásután hozzuk létre), úgy, hogy a felső címet az első időpillanatban állítjuk elő, az alsót pedig a másodikban. Természetesen ez azzal jár, hogy a felső címet egy tárolóba kell beírnunk ahhoz, hogy a teljes címzés befejezése után rendelkezésre álljon.

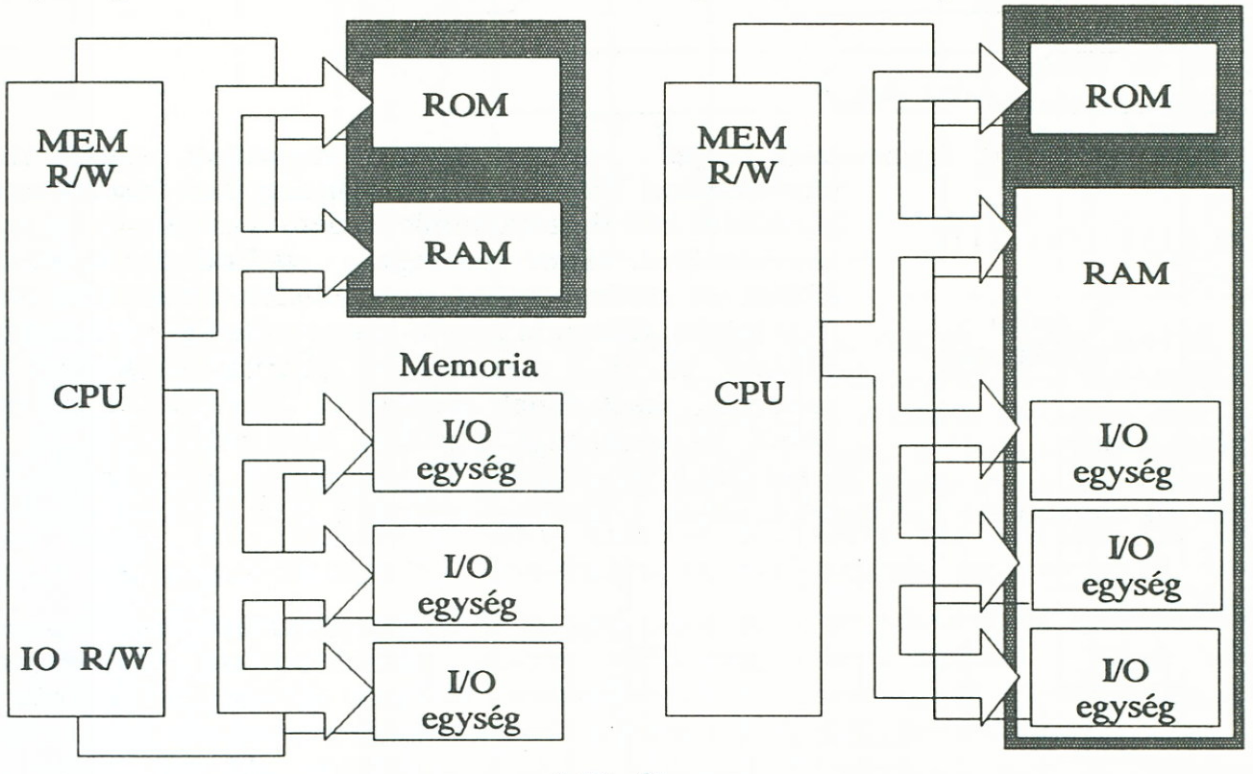
6.5. Eszközök illesztése sínrendszerekhez

Mielőtt belekezdenénk az illesztések kérdésének tárgyalásába, meg kell ismerkednünk bizonyos alapfogalmakkal. Ilyen a Be-Kiviteli cím, a port, és az eszköz fogalma.

- **Be-Kiviteli címen** azt a címet értjük, amely arra az eszközre mutat, amellyel a processzor kommunikálni akar.
- A **port** olyan érvényes be illetve kiviteli cím, amellyel adatforgalom bonyolítható le.
- Az **eszköz** egyetlen áramkör vagy tok, amelyet a CPU vezérel.Az eszköznek több portja is lehet.

6.5.1. Perifériák illesztése

A perifériák kezelésében alapvetően két nagy csoportot különíthetünk el, az úgynevezett **független I/O** (Input/Output, be illetve kimeneti egység) és a **tártérképes I/O** egységeket. A független I/O eszközök esetén külön vezérlőjel szolgál a perifériák címzésére. (I/O R periféria olvasás és I/O $\overline{W}$ periféria irás vezérlőjele) Ezzel a megoldással a teljes memóriaterület hasznosítható. Szemben a tárba ágyazott perifériakezeléssel, amelynél engedélyező jelként a memória írás illetve olvasás jelét használhatjuk fel (MEMR/MEMW).



6.12. ábra

A független és tárba ágyazott I/O egységek vezérlésének elvi vázlata.

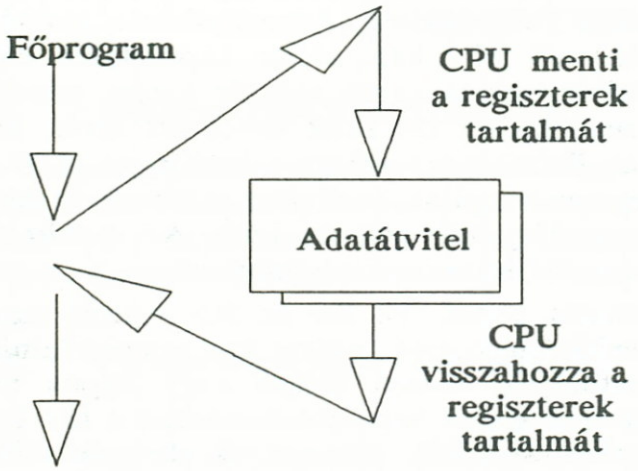
Ez utóbbi megoldást, akkor alkalmazhatjuk előnyösen, ha az eszköznek nagyon sok tárkezelő müveletet kell végrehajtania. (A processzoroknak lényegesen több memóriakezelő utasításuk van, mint ahány perifériakezelő.)

6.5.2. A be/kivitell eszközök (I/O) programozási osztályai

A perifériakezelés során különböző kapcsolattartási módokat fejlesztettek ki, hogy a periféria és a központi egység minél gazdaságosabban használja ki egymás erőforrásait.A legegyszerűbb ilyen kapcsolattartási forma az úgynevezett **feltétel nélkülikapcsolat**.Ennek lényege az, hogy a periféria mindig készen áll az adatok küldésére vagy fogadására. Vezérlő, vagy státusz információ nem szükséges, az adatok egy regiszterben állandóan rendelkezésre állnak. Hátránya, hogy a perifériának állandóan készenlétben kell állnia.

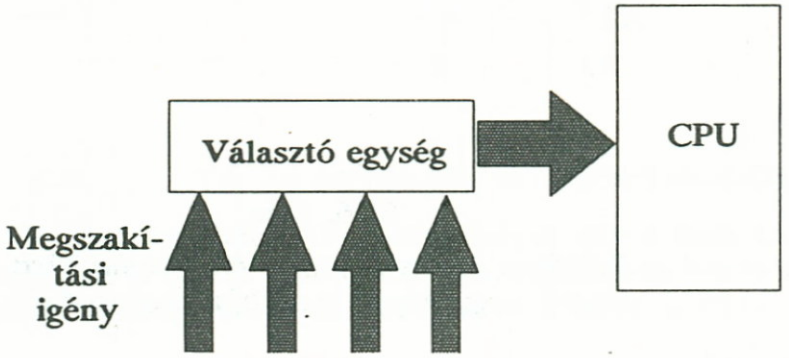
A **feltételes kapcsolat** során ellenőrizni kell a periféria valamilyen státuszinformációját. Ekkor a kapcsolattartás folyamata a következőképpen történik: a processzor figyeli a periféria státuszát és mihelyt az megfelelő, létrejön az adatátvitel. Mint látható ez egyben azt is jelenti,hogy a CPU állandóan a periféria figyelésével van elfoglalva, tehát addig várakoznia kell, amíg a periféria nem kész a kommunikációra.A folyamat tehát egyáltalán nem mondható gazdaságosnak.

Ezen a problémán segített a **programmegszakításos** átvitel kidolgozása.Ennek lényege,hogy a periféria mihelyt kész az adatok fogadására illetve adatot kíván küldeni egy vezérlő vonalon, értesíti a központi egységet a szándékáról. Ekkor a CPU abbahagyja az éppen végzett munkáját és a periféria kiszolgálásával foglakozik. Olyan ez, mint amikor az ember vendégeket vár, de nem tudja mikor érkeznek. Ekkor vagy azt csinálja,hogy állandóan az ablaknál áll és várja a vendégek érkezését (státuszinformáció,feltételes kapcsolat) vagy pedig folytatja a munkáját és vár a csengő megszólalására. Ekkor abbahagyja a munkáját és a vendégeivel foglalkozik.Amikor azok elmennek, ismét ott folytatja a munkát, ahol annak idején abba hagyta. A megszakítási rendszerek között különféle csoportokat alkothatunk. A **megszakításcsatolásos** átvitel esetén a rendszer megőrzi a megszakítás előtti állapotot és a megszakítás befejezése után vissza tölti azt. A rendszer ilyen esetben automatikusan ment. Előfordulhat olyan eset, hogy azonos időben több eszköz is kéri a megszakítást a processzortól.Ekkor a **megszakításprioritási** rendszereket alkalmazzuk. Ezek lehetnek **belső és külső prioritásúak**.A belső prioritású rendszereken belül beszélhetünk egy és több prioritású rendszerekről. Az egyprioritású rendszer esetén a processzornak csak



6.13. ábra

A programmegszakításos átvitel vázlata.



6.14. ábra

Külső megszakítási rendszer felépítésének vázlata.

egyetlen kapcsolója van, pl: IRQ. A több prioritású rendszer esetén a CPU több megszakítási lehetőséggel rendelkezik.Ilyenek a normál megszakítás IRQ, a gyorsmegszakítás FIRQ és a nem **maszkolható megszakítás** NMI. Az utóbb felsorolt lehetőségek közül a legmagasabb prioritással az NMI rendelkezik.A maszkolás számára nem látszik.El lehet takarni, mintegy maszkot húzva rá. Nyilvánvalóan ha egy megszakítás nem ilyen, akkor a CPU először azt fogja végrehajtani,hiszen az nem takarható el. A gyorsmegszakítások során a processzor csak a visszatérési címet és a feltételregiszter tartalmát menti el, juttatja a verembe,majd letiltja a további gyors és normál megszakításokat.A normál megszakítás esetén a gép minden lényeges információt lement a program állapotáról, így nem csonkul annak megszakítás előtti állapota.

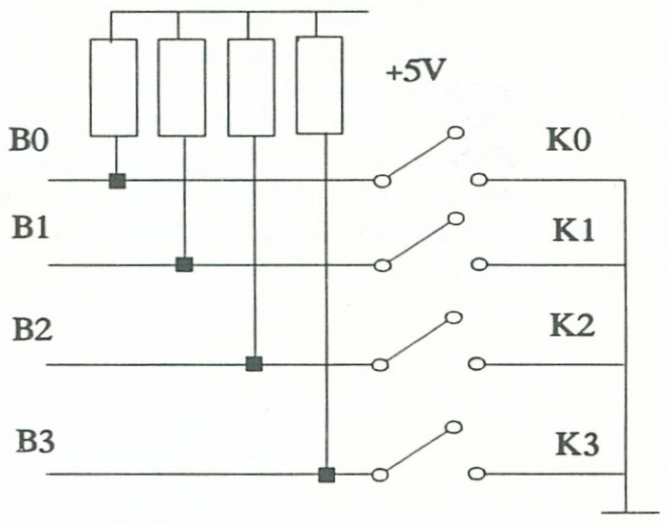
Külső megszakításprioritású rendszerek esetén egy külső megszakításvezérlő egység látja el a beérkező igények kezelését és szabja meg ezek sorrendjét.

A továbbiakban áttekintjük az alapvető be illetve kiviteli struktúrákat.

6.5.3. Címzett port típusú perifériák

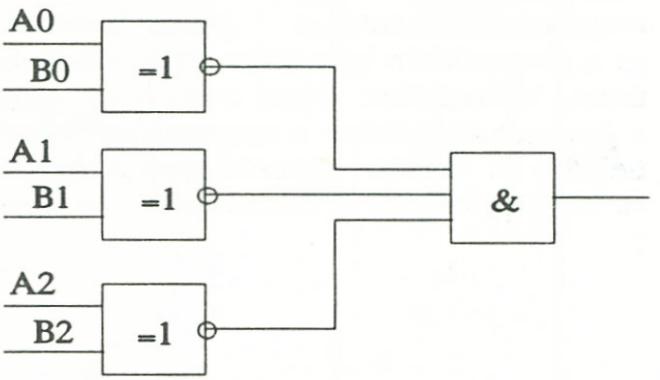
Az ilyen típusú megoldások lényege, hogy minden egyes porthoz hozzá van rendelve egy egyedi bináris azonosító kód. (Általában a processzorok többsége 8 bites azonosítót használ a perifériák kijelölésére.) Ez a bináris kód rögzített. Amennyiben a periféria ezt a bináris jelet „látja" és megfelelő engedélyező jelet kap, akkor kapcsolatba léphet a központi egységgel. A be/kiviteli művelet lebonyolítása során először a cím jelenik meg a címbuszon, majd az adat érvényes jel segítségével szerzünk tudomást arról, hogy az adat az adatsínen van.Ezek után a CPU az összes port számára kiadja az IOW illetve IOR jelet, de csak az a port fog erre ténylegesen reagálni, amelyikre a címzés vonatkozott. Ezt a három műveletet érvényes **portengedélyezésnek** nevezzük. Az eszköz bináris kódját könnyen előállíthatjuk kapcsolók, vagy fix huzalozás segítségével.

A kapcsolók állása határozza meg a bináris kódot. Pl: K0 és K3 nyitott kapcsolók esetén az azonosító kód 1001 lesz. A perifériának csak akkor kell megszólalnia,ha a címzésnek szól. Ezt is könnyen megvalósíthatjuk, hiszen kizáró vagy kapuk negáltja pontosan ezt a feladatot oldja meg. (Akkor lesz igaz a kimenet, ha mind a két bemenet azonos értékű.) Természetesen léteznek olyan áramkörök is, amelyek elvégzik több bitre is az összehasonlítást. (Pl: SN7485 4 bitre oldja meg az összehasonlítást.)



6.15. ábra

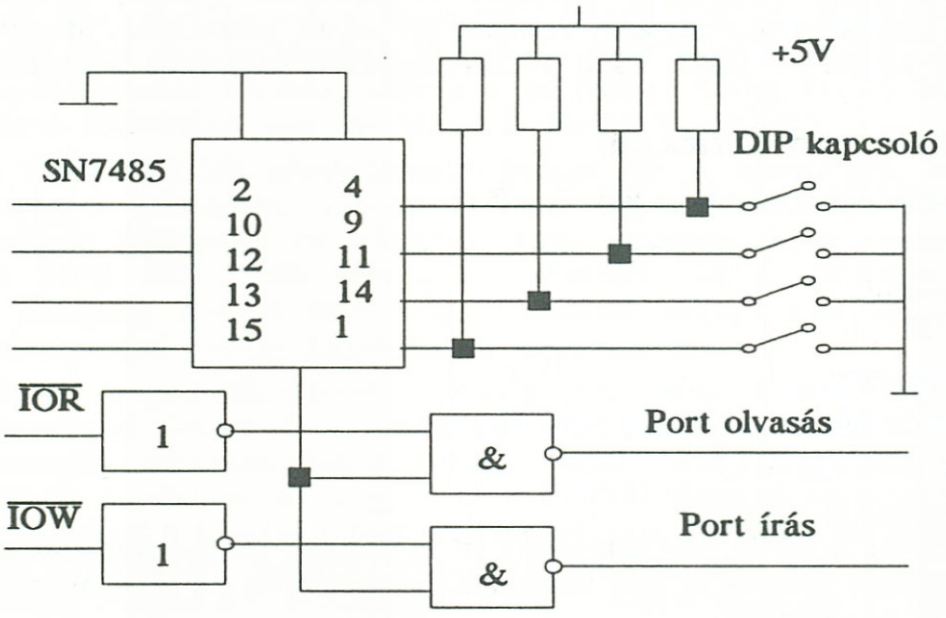
Perifériák portcímének kialakítása.



6.16. ábra

Periféria engedélyező jel kialakítása kizáró vagy kapuk negáltja segítségével.

A példának okáért nézzünk meg most egy olyan kapcsolást, amely megvalósítja egy perifériának a kiválasztását.Látható,hogy eszközünk 8 bites címzés esetén többször is megszólal.



6.17. ábra

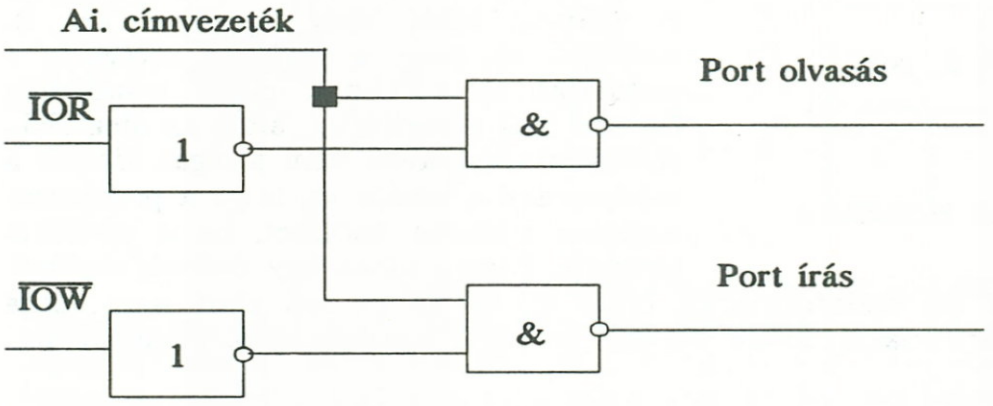
A periféria port olvasó és iró jelének előállítása.

6.5.4. Eszköz-port típusú bevitel

Ilyen megoldást akkor alkalmazunk, ha egy eszköznek több portja is van, amelyikhez hozzá kell férni. Például nyolc bites adatbuszon 16 bites adatokat akarunk szállítani.Ekkor a címbitek csoportját kettéválasztjuk, az egyikkel az eszközt, a másikkal a portokat azonosítjuk. Ezeket portkiválasztó vezetékeknek is nevezik.

6.5.5. Lineáris kiválasztású átvitel

Ennek a módszernek az a lényege, hogy annyi periféria címezhető, amennyi címvezetékünk van. A megoldás igen egyszerű, de csak erősen korlátozott számú periféria esetén használható. Például 16 bites portcímzés esetén 16 periféria.



6.18. ábra

A lineáris kiválasztású rendszerekben annyi periféria használható, amennyi címvezetékünk van.

6.6. Tárba ágyazott be/kiviteli eszköz

A tár címzésére adott lehetőséget ekkor nem használjuk ki teljesen,hanem a címterület valamelyik részét a perifériák azonosításásra használjuk. Például az A0 . . A15 címvezetékek közül néhány, mondjuk A0 . . A12 a tárat, a többi a perifériákat azonosítja. Természetesen

itt nem használhatunk be illetve kiviteli ,hanem csak a memóriára érvényes utasításokat A módszer előnye,hogy igen sokrétű utasítás használható,de lényegesen csökkentheti a tárkapacitását.

A független és tárba ágyazott megoldást nemcsak a perifériák igénye, hanem a processzor adottságai is meghatározzák. Ugyanis van néhány olyan processzor, amelynek nincs külön perifériakezelő kimenete. Ekkor csak a tárba ágyazott megoldás használható.

6.7. Kommunikáció a perifériákkal

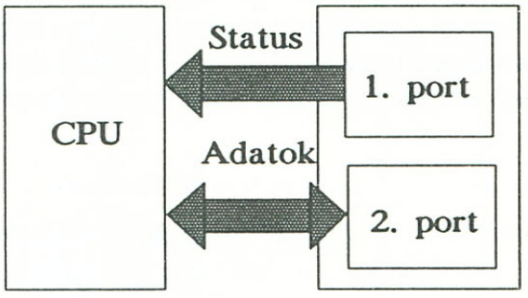
Általánosan a kommunikáció folyamata a következő :

- a cím megjelenése,
- adatvezeték előkészül,
- előáll az IO $R/\overline{W}$,
- aktív portengedélyezés,
- a háromállapotú kapuk engedélyező jelet kapnak,
- az adat rákerül az adatsínre,
- az adat a processzorba érkezik.

Az írás folyamata hasonlóan zajlik le. (Az adatbuszt itt is hasonló módon kezeljük,mint a memeóriáknál — a megfelelő időben le kell választani a perifériát az adatbuszról — ezért háromállapotú kapukat használunk.)

6.7.1. A kézfogásos adatátvitel HANDSHAKE

Ezt a megoldás azokban az esetekben alkalmazzuk,ha a működtetett periféria lényegesen lassúbb, mint a processzor adatfeldolgozó sebessége (Pl: nyomtató). Ebben az esetben a CPU-nak tudnia kell, hogy az illető eszköz mikor végzett a feladatával, mikor tudja fogadni, vagy küldeni a következő adatot. Így tehát a perifériának értesítenie kell a processzort a munka befejezéséről, azaz periféria kész jelet kell szolgáltatnia.



6.19. ábra

Kézfogásos átvitel portok kialakításával

Egy lehetséges megvalósítása a rendszernek, ha az eszközön két portot alakítunk ki, amelyikből az egyik csak az adatok fogadásával, a másik pedig az állapotinformáció (státusz) cseréjével lesz elfoglalva. A státusz lehet akár egyetlen bit is, mondjuk az, hogy a periféria elvégezte a munkáját. A CPU-nak ekkor mindaddig figyelni kell a perifériát, amíg az megfelelő státuszinformációval nem szolgál. Ennek a módszernek a hibája az, hogy a processzor végtelen ciklusba kerülhet, ha a periféria elromlik. Ezen a hibán úgy szoktak segíteni, hogy a CPU csak néhány státuszinformációt olvas le, és ha ez idő alatt nem tudja felvenni a kapcsolatot a perifériával, akkor lemond annak a használatáról, elengedi azt.

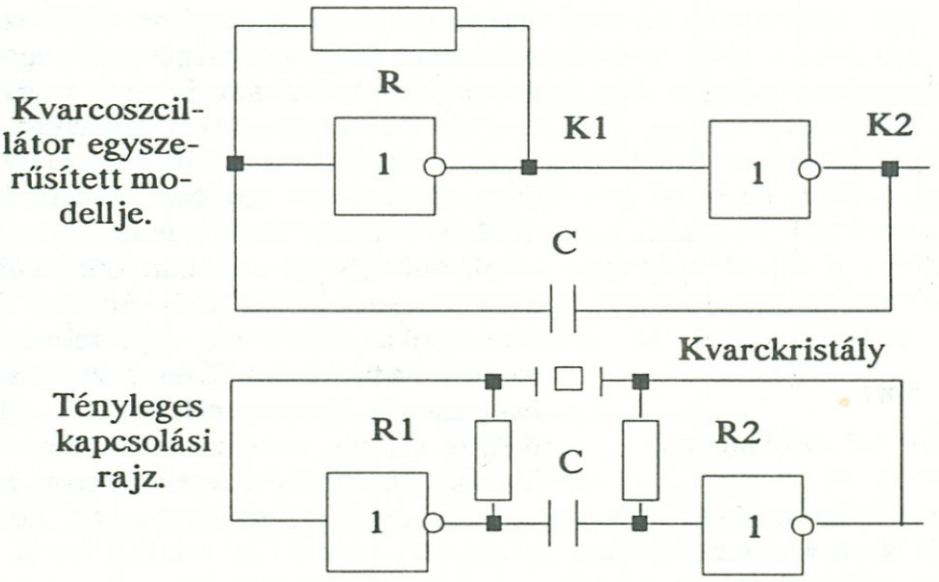
6.7.2. A soros adatátvitel

Sok esetben nem lehet megoldani azt, hogy buszokon oldjuk meg adataink szállítását. (Például hosszú vezetékek esetén a felvett zavarok elnyomnák a hasznos jeleket, vagy igen költségesen lenne megvalósítható.) Ilyenkor a soros adattovábbítás segít a gondon. Ennek a lényege —mint korábban már megismertük— az, hogy az információs biteket egyetlen jelvezetéken keresztül, egymás után továbbítjuk az adóból a vevő felé. Az adatátvitelt természetesen csak úgy tudjuk pontosan megvalósítani, ha mind az adó, mind

pedig a vevő oldalon rendelkezésre áll egy időetalon, órajel, amelynek segítségével szinkronizálhatók az adó és a vevő oldalán is az átvitt jelsorozatok. Az órajel üteme természetesen nem lehet akármilyen. Ha nagyon hosszú ideig H szintű egy periódusban, akkor előfordulhat olyan eset, hogy az általa szinkronizált kapu sokáig lesz nyitva és ez idő alatt megváltozik a bemenetre érkező adat. Nyilvánvalóan ekkor az előbbi adat elveszik. Ezt úgy igyekeznek kikerülni, hogy nem az órajel H vagy L szintjéhez kötik a kapuk nyitását, zárását, hanem a jel felfutó illetve felutó éléhez rögzitik azt.(Ez jóval rövidebb időtartam mint az órajel H illetve L szintje.)

A szinkronizálás követelménye persze azt is megszabja, hogy az adatbitek milyen sebességgel áramlanak a vezetékekben, ugyanis nagyobb frekvenciájú órajel nagyobb sebességet kényszerít az adatokra. Ezen sebesség szabványokban meghatározott értéke 1200, 2400, 4800, 9600 bit/s lehet. (Például ez azt jelenti,hogy másodpercenként 9600 bit/s sebesség mellett 9600 bit információ áramlott át az adóból a vevőbe. A bit/s mértékegységet szokás Baud-nak is nevezni.)

Ha jól megnézzük ezeket a sebességeket az is kiderül, hogy a processzor működését szinkronizáló órajel üteméhez képest ez lényegesen különböző, hiszen a CPU MHz frekvenciájú tartományban dolgozik, az átvitt jelek tartománya pedig 1,2-től — 9,6 KHz-ig terjed. Ez az oka annak, hogy a soros átviteli egységek sziknronizálását külön óragenerátor látja el. Mivel a generátor frekvenciájának igen stabilnak kel lennie (a pontos szinkronizáció miatt) úgynevezett kvarcoszcillátorokat alkalmaznak erre a célra. Az ilyen oszcillátor a fizikában megismert rezonancia — a kristály saját és a kényszerrezgés frekvenciája megegyezik— jelenségére épül. A kvarc rezgése, mint atomi oszcillátor nagyon pontos. Ha egy külső rezgőkör megrezgeti, akkor annak frekvenciáját a saját rezésének ütemében stabilizálja. Az 6.20. ábra alapján tekintsük át a működését.

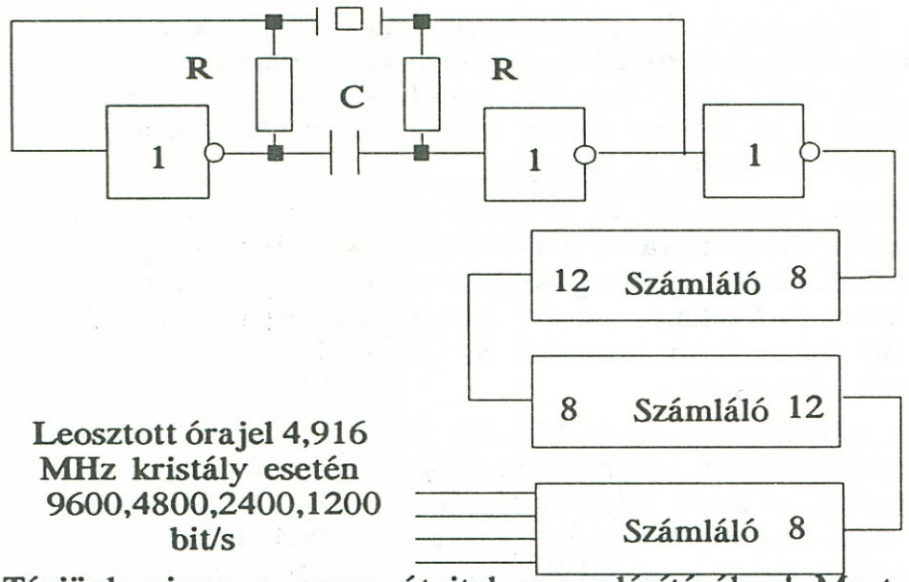


6.20. ábra

Kvarcpontosságú órajelgenerátor felépítése.

Tegyük fel, hogy a két kapu ellentétes állapotban van. (A két sorba kapcsolt inverter kimenete biztosan teljesíti a feltételt.) Mondjuk a K1-el jelzett magas, a K2 alacsony szinten tartózkodik. Ekkor a C kondenzátor az R ellenálláson keresztül feltöltődik (a kialakult pontenciálkülönbség miatt). Ez egyben azt is jelenti, ahogyan nő C töltöttsége és ezzel együtt a feszültsége is, az első kapu bemenete H szintre emelkedik. Ekkor ez a kapu átbillen és magával rántja a második kaput is. Tehát a potenciálviszonyok megváltoztak. Ennek egyenes következménye, hogy a kondenzátor kisül, töltöttsége és ezzel együtt a kapcsain mérhető feszültség csökken. (Az R ellenálláson keresztül áram indul meg a K1 pont felé.) A kapuk tehát visszabillennek. A folyamat folytonosan ismétlődik, a rezgés beindul. Mivel a kvarckristályok frekvenciája nem tetszőleges, a

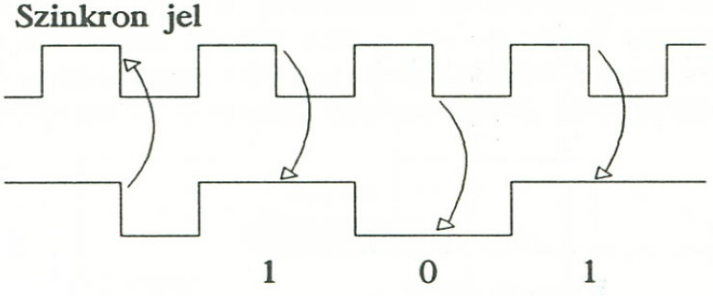
kapott rezgés frekvenciáját általában még le kell osztani olyan üteműre, amely a kívánságnak megfelel. Erre a célra számláló áramköröket használnak. Lásd az alábbi ábrát.



6.21. ábra

Leosztott órajel előállításaszámlálók segítségével. (A számláló áramkör SN74197-es az R ellenállások 330 ohmosak, C 200 pF, az inverter SN7406.)

Térjünk vissza a soros átvitel megvalósításához! Most már tudjuk, hogy megfelelő szinkronizáló ütemimpulzus esetén könnyedén megkaphatjuk az adó által kibocsájtott adatokat. Azonban ha jól meggondoljuk, ennek megvalósítása két jelvezetéket jelent, egy adat és egy szinkronizáló vezetéket. Nem lehetne csak egyetlen jelvezetéket alkalmazni a feladat megoldására, ha már úgyis az a célunk, hogy minél kevesebb vonalat használjunk fel? Természetesen erre is van lehetőség, ha mind az adó és mind a vevő meghatározott frekvencián rezeg„ ekkor egy, az átvitelt szinkronizáló jel segítségével is megoldhatjuk a problémát. Ez azt jelenti, hogy a vevő oldalán beérkező adatok előtt egy szinkronizáló impulzust küldünk, amely időben összerendezi a vevőben meglévő és az adóból érkező impulzusok menetét. Egy szinkronizáló jel azonban önmagában még nem elegendő. Fel kell állítani bizonyos szabályokat, amelyek egyértelművé teszik az adatok tartalmát. Ezen átviteli szabályok összességét **kommunikációs protokol**nak is szokás nevezni. Lehetséges megkülönböztetni úgynevezett szinkron és aszinkron átvitelt. **Szinkron átvitelről** abban az esetben beszélünk, ha az adatfolyam pontosan alkalmazkodik az órajelhez. Minden óraimpulzus alatt egy adatbit kerül átvitelre. A bájthatár felismerése céljából szinkronizáló impulzust küldünk.Amikor a vevő várja az adatot, akkor figyelő állapotban van és csak akkor kezdi el az adat dekódolását, ha előzőleg szinkron impulzus érkezett.



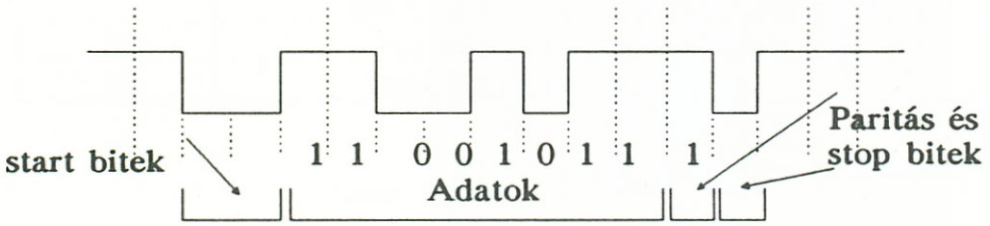
6.22. ábra

Adatátvitel szinkron jel segítségével.

Aszinkron átvitel esetén az adó csak akkor továbbít karaktert,ha az már készen áll.A karakterek között szünet — általában H szintű — megszakításjel van. Ebben az esetben minden adategység tartalmazza a saját szinkronizálásához szükséges információkat, az úgynevezett keretbiteket.Ezek a START-, a STOP- és a PARITÁSbitek.

- A **STARTbit** azért szükséges,hogy tudja a rendszer, hogy hol kezdődik az adat. Ez a bit az alapállápottal mindig ellentétes.
- A **STOPbit** azért szükséges,mert a vevő innen tudja meg, hogy a szónak vége van. A beálló megszakításjel már nem tartozik az adathoz.

» A **PARITÁSbit** re az esetleges átviteli hiba felismerése végett van szükség. Ez természetesen igen egyszerű, de jól alkalmazható megoldás.A lényege az, hogy az átvitt adatban lévő egyesek számát páros paritás esetén párosra, páratlan paritás esetén páratlanra egészítjük ki. Ami egyben azt is jelenti,hogy 8 bites adatok esetén nem 8 adatjelet, hanem 9-et küldünk.(Ha a hibát ki is szeretnénk javítani, akkor üzenetet kell küldeni az adónak,hogy ismételje meg az előző bájt adását. Természetesen a módszer igen primitív.Léteznek ennél bonyolultabb eljárások is, mint például a meghatározott bináris számmal történő osztás módszere, amelynek segítségével összetettebb hibák is felderíthetők és javíthatók.



6.23. ábra

Aszinkron adatátvitel keretbitekkel

Eddigi ismereteink birtokában szedjük össze mindazokat, amelyeknek segítségével lehetővé válik a soros adattovábbítás megvalósítása. A számítógép belsejében a nagyobb működési sebesség miatt párhuzamos síneken történik az adatok szállítása.Ha meg akarjuk változtatni az adat szállítás módját, akkor szükségünk lesz az adóoldalon egy soros- párhuzamos, a vevő oldalon egy párhuzamos- soros átalakítóra. Ezt fizikailag, mint tudjuk léptető regiszterekkel valósíthatjuk meg, amelyek az adó oldalán párhuzamosan írhatók, a vevő oldalán pedig párhuzamosan olvashatók ki. A szinkronizáláshoz szükségünk van egy kvarc-stabil óragenerátorra, az adatok ellenőrzéséhez egy paritás vizsgáló áramkörre, valamint olyan csatornára, amely a lehető legkevesebb zaj felvétele mellett átszállítja az adatainkat az adóból a vevőbe. Mint látható meglehetősen összetett feladatról van szó. Szerencsére létezik egyetlen áramköri egységbe integrál megvalósítása is a feladatnak, az úgynevezett USART (Universal synchronous asynchronous receiver transmiiter — univerzális szinkron-aszinkron adó-vevő áramkör). Ilyen például az Intel 8251-es áramköre. Példaként ismerkedjünk meg ennek felépítésével!

Adat	Paritás		Átvitel	Paritás		megjegyzés
	páros	páratlan		páros	páratlan	
101110	0	1	101100	0	1	felfedezhető
101110	0	1	101110	1	0	felfedezhető
101110	0	1	101011	0	1	nem fedezhető fel

6.1. táblázat

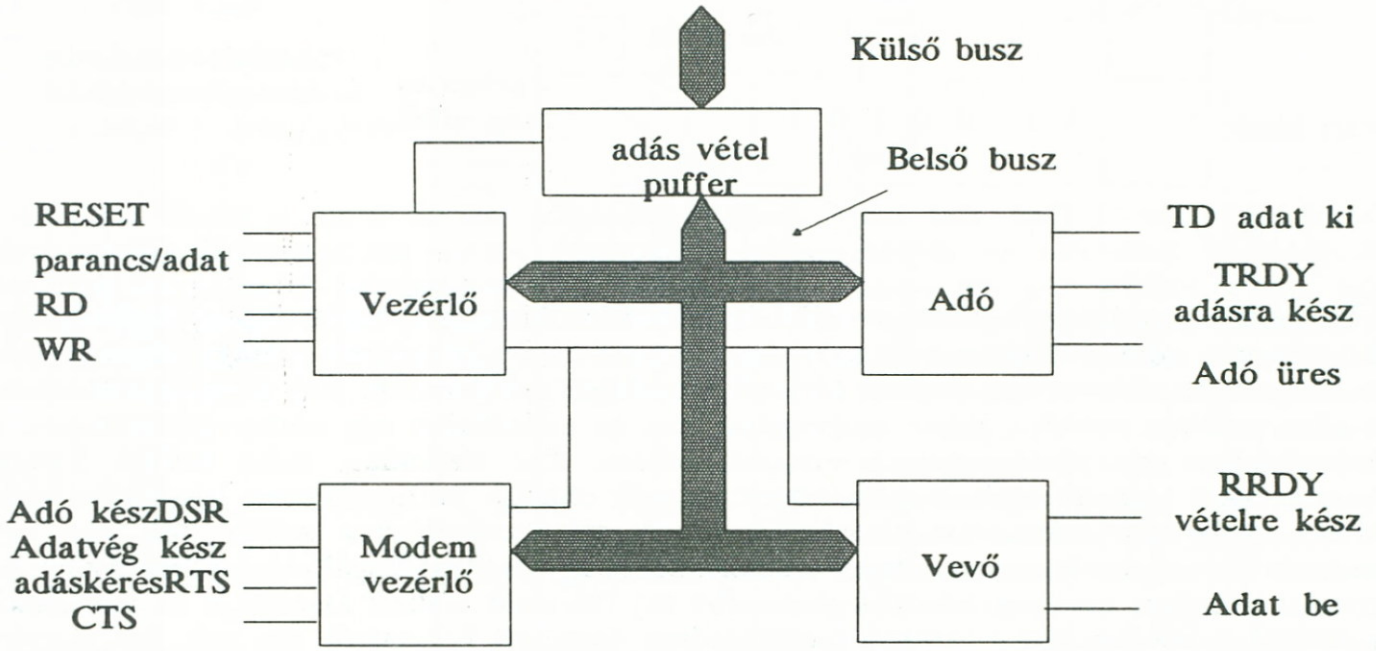
A paritásbit segítségével felfedezhető hibák azok, ahol az átvinni kívánt adat páratlan számú bitje sérült. Páros számú bit sérülése esetén nem fedezhető fel a sérülés.

Az áramkörnek külön óragenerátora van az átvitel ütemezéséhez. Az adásra kész jel hatására a processzor elhelyezi az adatvonalon az átvinni kívánt jelet. Az adáskész jel akkor generálódik,ha az átviteli puffer üres. Az adó soros formába átalakítja az adatokat,

ellátja a keretbitekkel (start, stop, paritás), majd elhelyezi az adó regiszterében. Az adat a TD adatvonalon kerül továbbításra,majd a TE vonalon jelzi a központi egységnek,hogy az adó üres, az átvitel befejeződött.

A USART az adatokat az RD adatbe vezetéken olvassa be és az RRDY vezetéken tudatja,hogy jelenleg adatátvitel zajlik.

Az áramkör működését a vezérlő irányítja a processzor felöl érkező jelek segítségével.A parancs/adat vonalon érkező jel alapján dönti el, hogy most adatot, vagy parancsot küldött-e a processzor. Az érkező információt ennek megfelelő regiszterbe tölti. Az áramkör számára érkező első parancsot talán célszerűbb üzemmódszónak nevezni, hiszen az átvitel módját fogja meghatározni a regiszterbe írt adat. Ilyen adatok például az adatátvitel sebességére, a karakterhossz beállítására, a paritás engedélyezésre, előállítására, illetve



6.24. ábra

Universalis szinkron aszinkron adó-vevő blokk vázlata. (USART)

ellenőrzésére, a stopbitek számára vonatkozó információk. A második parancs az adással és vétellel kapcsolatos információkat határozza meg. (Adás engedélyezés, adaterminál kész, vétel engedélyezés, adásmegszakítás, adáskérés, stb.) A további vezetékek az RD, a WR illetve a CE a szokásos értelemben használatosak.

(Az áramkör ezen funkciókon kívül képes adatátviteli berendezéssel MODEM-mel való kommunikálásra is.)

Az áramkörből kilépő adatok TTL kompatibilisek. Sok esetben azonban nem ezekkel a jelszintekkel találkozunk az átvitel megvalósítása során,hanem ettől lényegesen különböző, úgynevezett RS 232 szabvány alapján felépülő jelszintekkel. Ez a feszültségszint kiosztás a logikai 1-hez 8-tól 15V-ig, a logikai 0 szinthez -8 -tól -15 V-ig terjedő feszültséget rendel. A gyakorlatban igen sokszor 12 és -12 V-os szinteket valósítanak meg. Az áramkör megvalósítására úgynevezett szinteltoló áramkörök szolgálnak, amelyek működésének magyarázata túlhaladja a könyv kereteit. Aki kíváncsi ezek kialakítására, az irodalomból bőséges információ birtokába juthat.Kiegészítésül csupán szimbólum és funkció táblázatát közöljük.

A csatlakozó száma	A jel megnevezése	Adatirány PC ből az állomás felé
1	Védőföld. A modem vagy csatoló vázát köti össze, ha közös föld szükséges	—
2	Adat ki	PC -ből
3	Adat be	PC -be
4	Adáskérés	PC -ből
5	Vételkészség nyugtázása	PC -be
6	Adatkommunikációs berendezés készenléti jele	PC -be
7	Jelföld. A vezérlőegységhez csatlakozó valamennyi áramkör közös referenciapontja.	—
20	Az adatterminál készenléti jele.	PC -ből
22	Hívás indikátor	PC -be

6.2. táblázat

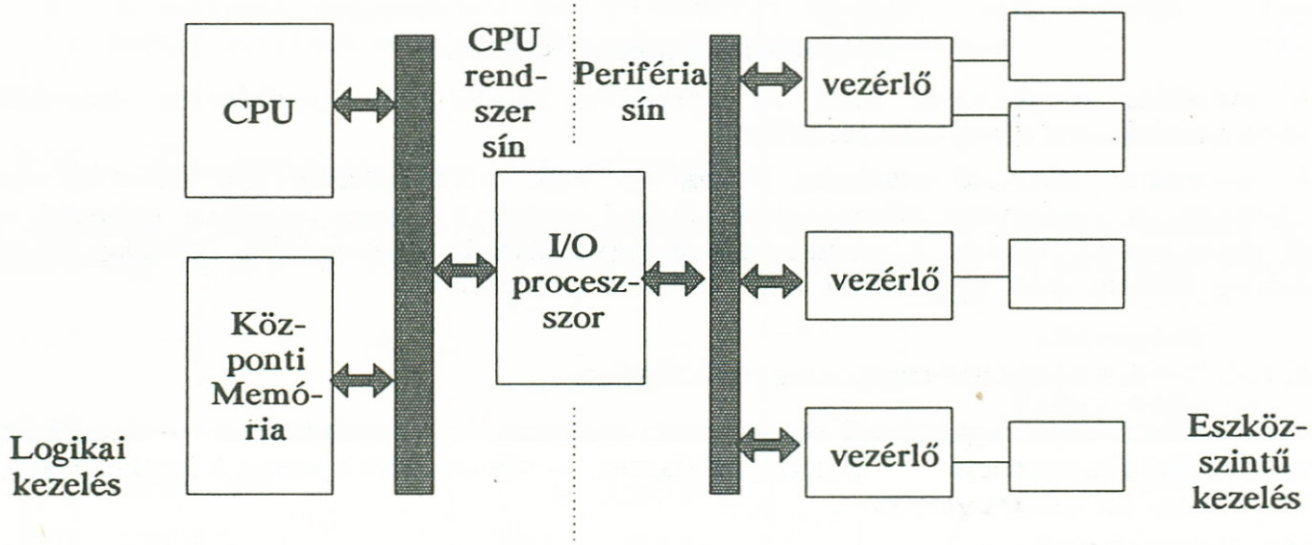
RS — 232 csatlakozó kivezetései.

6.7.3. Perifériák logikai kezelése

A perifériákhoz illeszkedő perifériák sokféle volta felvetette, hogy ne kelljen minden egyes berendezéshez külön illesztő felületet biztosítani,hanem általános eszközöktől független csatlakoztatással oldjuk meg a problémát.

Az eljárás lényege, hogy az adatcsere végrehajtása nem veszi igénybe a központi processzort, hanem önálló, úgymond autonóm módon történik. A probléma megoldására az úgynevezett **I/O processzor**okat fejlesztették ki. Ezek olyan eszközök, amelyek speciális be illetve kiviteli műveletekkel rendelkeznek, működésük erre optimalizált.Mindezek mellett a központi egység felé szabványos illesztő felületet biztosítanak. A folyamatokat három alapvető utasítástípus kezeli, ezek a

- » vezérlő,
- » státusz lekérdező,
- » információátvivőutasítások.



6.25. ábra

Az I/O proceszor helye a logikai és eszköz-szintű vezérlés között.

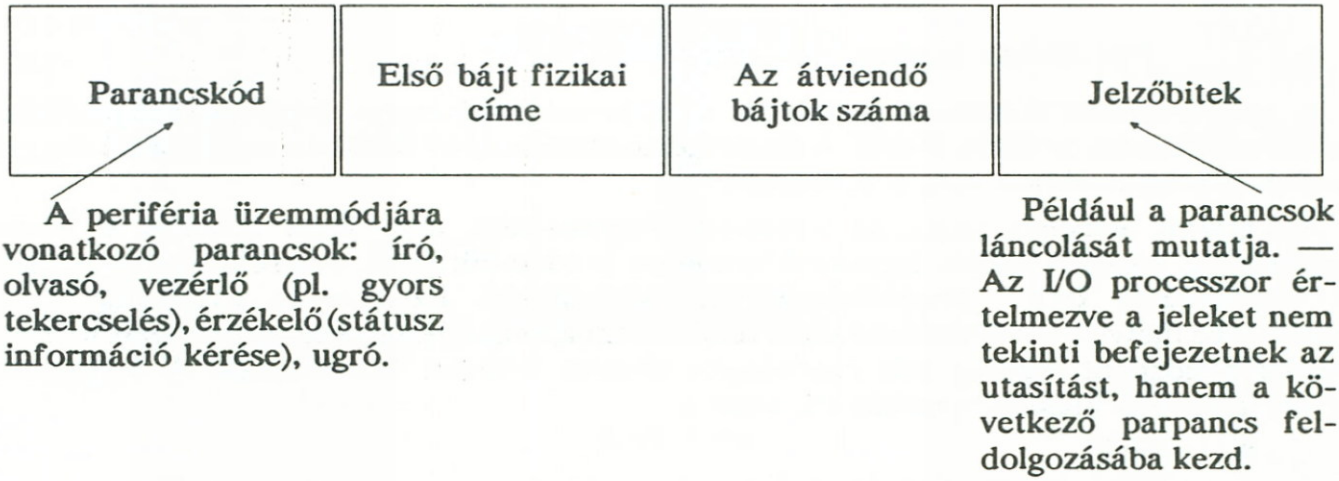
Ezek megfelelő sorozatát, az I/O processzor által végrehajtandó programot, a CPU helyezi el a tárban. Ugyanakkor az I/O processzor is az átvitt adatokat itt helyezi el és innen veszi a periférák számára küldendő adatokat. Ha az I/O processzor csak a be/kiviteli műveletek autonóm vezérlésére szorítkozik, akkor úgynevezett **autonóm csatornák** ról beszélünk.

Ha megfigyeljük a periférák működését, akkor azokat sebességük alapján két csoportba rendezhetjük el.

» Az adatátvitel szempontjából lassúbb periférák, nyomtatók, billentyűzet, (lyukkártyás és szalagos egységek). Ezek működtetése alapján észrevehető,hogy az egymást követő két átvitel közé tetszőleges számú szünet iktatható.

» A gyors periférák háttértárak, lemezes egységek. Az adatok csak sorozatokban továbbíthatók és ezek átvitele a periféria által megkívánt ütemben történik.

Ennek megfelelően két alapvető csatornatípust különböztetünk meg — a szelektor és multiplexor csatornákat. A **szelektor csatorna** esetén az egész átvitel időtartama alatt egyetlen kiválasztott (szelektált) periféria veszi igénybe az I/O processzor szolgáltatásait. Ez csak gyors nagy átviteli sebességgel rendelkező perifériák esetén használható előnyösen. (Ezek összeségében nem foglalják le jelentősen az I/O processzor idejét.) A **multiplexor csatornát** természetesen a lassú perifériák számára fejlesztették ki.



6.26. ábra

A csatornautasítások szerkezete.

A működés megkezdése előtt az operációs rendszer a megfelelően összeállított csatornautasításokat elhelyezi a tárolóban.

A csatornára küldhető eszközcím hossza egy bájt, ezért fizikailag 256 különféle eszközt tud kezelni. A csatornára rákapcsolódó fizikai eszközök száma azonban műszaki okok miatt alatta marad ennek. A csatornára mindegyik eszköz rákapcsolódik fizikailag, azonban logikailag mindig csak eggyel tart kapcsolatot a processzor.

6.7.4. A szelektor csatorna működése

A csatorna három regiszterrel rendelkezik: **csatorna címregiszeter** — utasításszámlálóként működik, **csatorna parancsregiszter** — ide töltődik a soron következő utasítás kódja, illetve az **adatregiszter** .

Működésének lépései:

» A végleges csatorna és eszközcím kialakítása.

» Az eszközcím a megcímzett csatornába kerül

» A szelekciós vezeték beállításával felszólítja az eszközvezérlőket a cím megvizsgálására.

Ezzel indul be a kezdeti kiválasztási folyamat.

» A megcímzett eszköz blokkolja a címet. (A soros felfűzés miatt nem engedi tovább.)

» Az eszközvezérlő válaszjelet és saját címét visszaküldi a csatornába.

Ezzel létrejön a logikai összekapcsolódás folyamata.

» A csatorna címszó bekerül a megfelelő csatorna címregiszterbe, majd ennek alapján kiválasztódik az első csatornaparancs és betöltődik a parancsregiszterbe.

» A csatorna címregisztere inkrementálódik és ezzel a második címre mutat.

» A csatorna az első parancs kódját kiküldi a kiválasztott vezérlőnek.

» Az eszköz egy állapotbájt beküldésével reagál, ezzel jelzi, hogy a kódszó elfogadásra került.

A kezdeti kiválasztási folyamat ezzel a lépéssel ér véget.

» A szelekciós vezeték jele megmarad, ami jelzi, hogy logikai összekapcsolás áll fenn és ez az átvitel végéig fenn is marad. (Szelektor típusú a csatorna.)

» 1.) A csatorna írás parancs esetén a parancsregiszterbe töltött szó címrésze alapján kiküldi a rendszer szóhosszának megfelelő bájtot, majd inkrementálódik a szó hosszának megfelelően.

» Az eszköz által kért ütemben átíródik a meóriatartalom.

» Az átviendő bájtok száma természetesen e folyamat közben csökken.

» 2.) Olvasás parancs esetén, az eszköz küldi az adatokat,melyet a csatorna adatregisztere fogad. Betelte után a csatorna átírja a központi memória megfelelő rekeszébe ennek tartalmát.

» Az adatátvitel végén (a bájtszámláló tartalma nulla) a csatorna közli az eszközzel,hogy több adat átvitelére nem tart igényt.

» Az eszközvezérlő állapotbájt küldésével reagál.

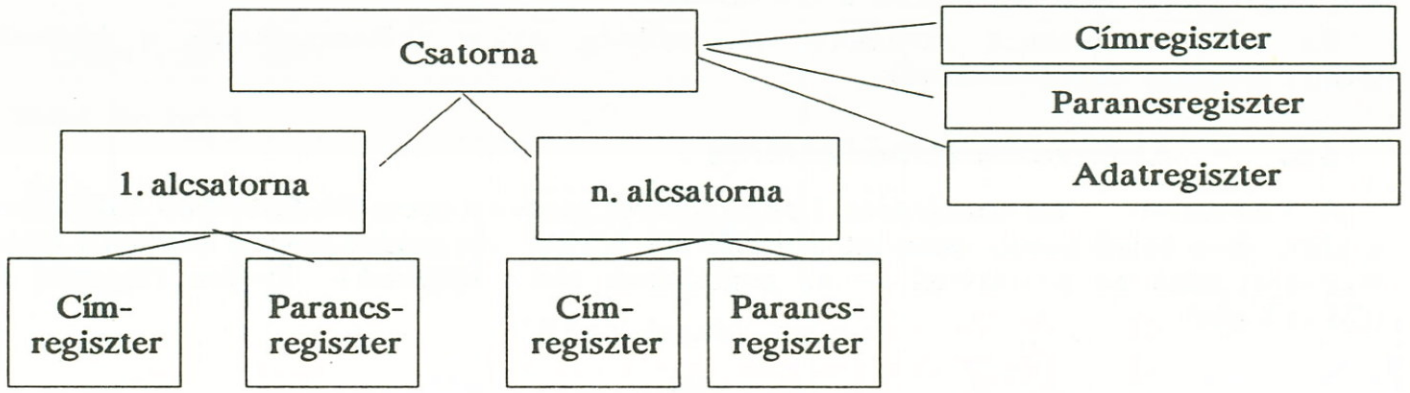
» Ha az átvitel mindkét oldalról (csatorna, eszközvezérlő) befejeződött, akkor a rendszer csatorna állapotszót generál.

Ezzel a logikai kapcsolat megszűnik.

» A csatorna megszakítást kér a központi egységtől, hogy közölje a munka befejeződésének okát (sikeres vagy sikertelen végrehajtás).

6.7.5. A multiplexor csatorna működése

A csatorna a szelektor csatornánál összetettebb képet mutat, ugyanis minden egyes eszközhöz egy alcsatornát rendelünk. Erre azért van szükség, mert a multiplex működés



miatt az eszközvezérlők átmenetileg leválnak a csatornáról, átadva a helyet más eszközvezélőknek. Ez pedig, mivel felfüggesztett munkájukat folytatni akarják, csak úgy lehetséges, ha elmentik a működésük pillanatnyi fázisát tartalmazó adatokat. Ezért a regiszter struktúra a következő képet mutatja:

Az átvitel folyamata.

» Az operatív memóriából a csatornacím átkerül a csatorna-címregiszterbe.

» Ennek alapján a kijelölt végrehajtandó parancs a csatorna parancsregiszterébe töltődik.

» Beindul a kezdeti kiválasztás folyamata.

» Létrejön a logikai összekapcsolódás. (Hasonlóan a szeketor csatonához.)

» Az eszközvezérlő állapotbájt beküldésével jelzi a parancs elfogadását.

» A szelekciós vezeték jele megszűnik, (a csatorna multiplex típusú) lehetőséget adva az eszköznek a leválásra. Ezzel biztosítja, hogy fel illetve elkészüljön a következő utasítás illetve adat vételére.

» Leválás esetén a csatorna az eszközhöz rendelt alcsatorna regisztereibe tölti a csatornaregiszterektartalmát.

» A szabad csatornára a kezdeti kiválasztási folyamat befejeződése után, az eszközvezérlők csatorna kérelmet nyújthatnak be.
Az egyidőben érkező kérelem jelzések kiszolgálására prioritási sorrendet kell felállítani. (A csatornához fizikailag közelebb lévő eszköznek van nagyobb prioritása.)

» A csatorna válaszjelet küld.

» A válaszjelhez legközelebb eső eszközvezérlő (a prioritás miatt, feltéve ha kérte a kiszolgálását) blokkolja azt.

» Az eszköz visszaküldi a válaszjelet és a címét. (A címre azért van szükség, hogy a csatorna azonosítani tudja melyik alcsatorna regiszterből kell kivennie és a csatornaregiszterbe töltenie az értékeket.)

» A cím alapján a megfelelő alcsatornaregiszterek visszatöltődnek a csatornaregiszterekbe.

» Megtörténik az átvitel folytatása, az adatcsere.A regiszterek ennek megfelelően inkrementálódnak illetve dekrementálódnak.

» Ha az utasítás-bájtszámláló értéke nullára csökken, vagy betelik, a csatorna jelzi az átvitel végét.

» Az eszközvezérlő végfolyamatot kér a csatornától és megtörténik a logikai leválás.

» A csatorna állapotszót generál, amelyet a CPU-nak küld be értesítve a folyamat sikeres vagy sikertelen voltáról.

A bemutatott megoldás mind bájtos, mind blokkos üzemmódban dolgozhat. Az eszközvezérlők szabják meg igényük alapján a szükséges folyamatot, pontosabban szólva azt, hogy mikor akarnak leválni a csatornáról.

Ha csak adatblokkok átvitelére van szükség, akkor felhasználhatók a közvetlen tárhozzáférésen alapuló eszközök.

6.8. A közvetlen tárhozzáférés

A közvetlen tárhozzáférés DMA, (Direct memory access) adatátvitelt valósít meg a külső és a belső tárak, valamint a perifériák között, oly módon,hogy a központi egység munkáját csak az adatátvitel — az adatelemek tárba írásának— idejére függeszti fel. (Ciklusloпás)

A DMA célja a központi egység munkájának megkönnyítése. A közvetlen vezérléssel nagy sebességet és nagy mennyiségű adat átvitelét valósítja meg. Az egyszerűbb megoldásoknál központi egység, az I/O egységek közös buszon osztozkodnak.Természetesen ez azt is jelenti, hogy azonos ciklusban nem férhetnek a memóriához. A közvetlen tárhozzáférés során a DMA-t megvalósító eszköznek különféle műveleteket kell elvégeznie az adatiránynak megfelelően.

A DMA műveletei:

- Az I/O egység adatának átvitele I/O egységre.
- Az I/O egység adatának átvitele a memóriába.
- A memória megfelelő adatának kivitele az I/O egységre.
- A memória memória közötti adatátvitel.

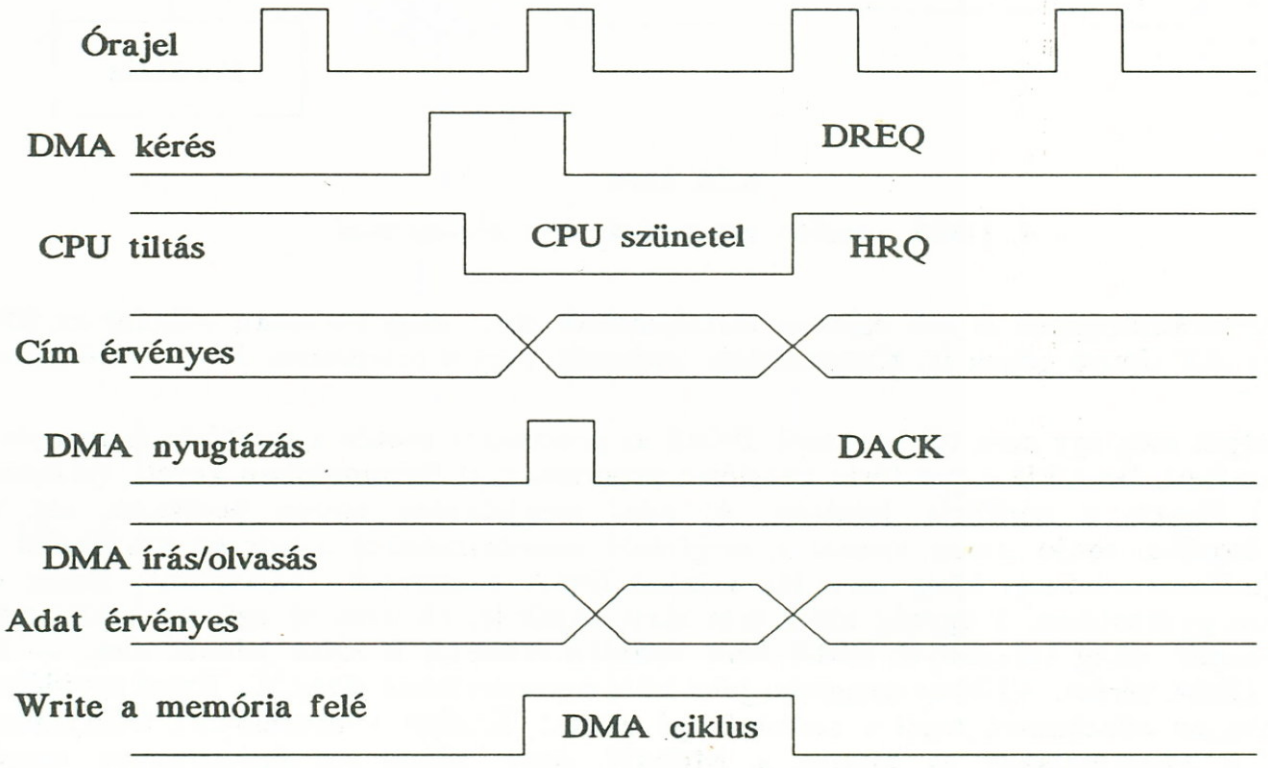
Az adatátvitel megvalósítására több különféle megoldás létezik.

A központi egység leállítása. — A CPU az adatátvitel alatt lekapcsolódik a buszról, ezzel lassítja a központi egység működését.

A memória időszelet megoldás. — A memóriaciklust két időszeletre bontja.Az elsőben a központi egység, a másodikban a DMA hozzáférését biztosítja.

A **cikluslopás.** — Ha a központi egység és a DMA vezérlő egyszerre jelent be tárigényt, akkor a közvetlen tárhozzáférést kiszolgáló eszköznek prioritása van a központi egységgel szemben. Ami tehát azt jelenti, hogy a CPU vár a DMA kiszolgálására, egy ciklus erejéig.Az eszköz feladatából eredően az intelligens, programozható perifériák közé sorolható.A műveletek végzését a DMA regisztereibe töltött értékek határozzák meg. A rendszer feladatából adódóan három regisztert tartalmaz.

- Címregisztert — a memóriarekesz címét tartalmazza, amelyre a következő írás/olvasás művelet vonatkozik.
- Számlálóregiszter — az átvinni kívánt szavak számát tartalmazza.
- Állapotregiszter — tükrözi az adatáramlás irányát,üzemmódját.



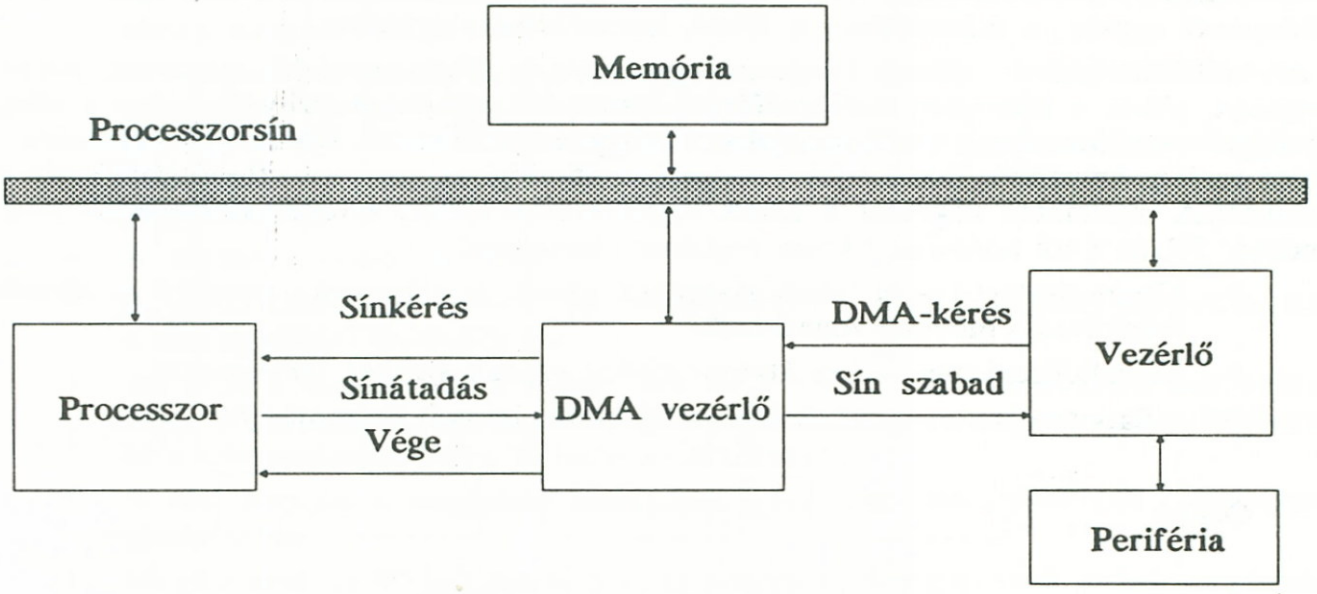
6.27. ábra

DMA időzítési ciklusai.

A központi egység szempontjából a regiszterek egy I/O egység regisztereinek feleltethetők meg. (A CPU így látja.) A rendszer programozásának lépéseit is a regiszterek megfelelő értékekel való feltöltése jelenti.

A DMA ciklus végrehajtása több lépésből tevődik össze. A következőkben egy külső eszköz adatainak tárolása példáján mutatjuk be.

- A DMA vezérlő adatszót fogad az I/O interface-től.
- A központi egységhez rendszerbusz kéréssel fordul.
- Ha megkapja a buszt, akkor kiküldi a megfelelő memóriarekesz címét,majd kiküldi a megfelelő adatszót.
- A memória kész jel vételekor a DMA megvizsgálja a szószámot, ha ez nem egyenlő zérussal, akkor a számlálóregiszter tartalmát eggyel csökkenti, a címet pedig az ávitt szónak megfelelően növeli.
  Ha a szószámláló értéke egyenlő zérussal, befejezi az átvitelt és közli a központi egységgel az adatátvitel végét.



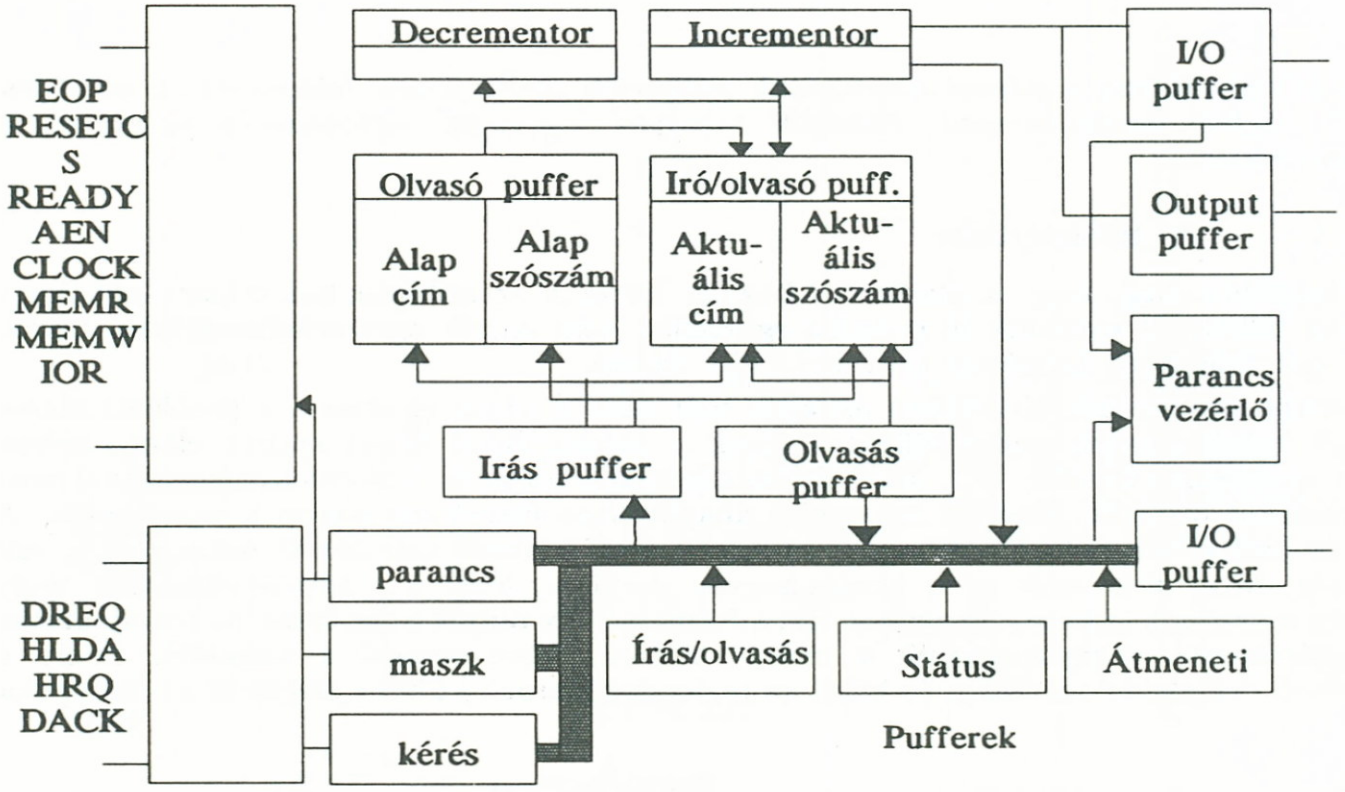
6.28. ábra

A DMA vezérlő kapcsolata a környezetével.

A mikroszámítógépek is sok esetben tartalmaznak ilyen megoldásokat, például az IBM XT illetve AT típusú gépek is. Kiegészítésül tekintsük meg e rendszerek DMA vezérlőinek felépítését.

Vizsgáljuk meg egy nem túl korszerű, I8088-as processzor esetén a periféria és memória közötti átvitelt. Ha CPU a periféria kezelését programozott üzemmódban kezeli, (várakozó hurokben figyeli a periféria jelzéseit, új adat megjelenése esetén beolvassa azt az akkumulátorába, majd innen kiteszi a megfelelő memóriacímre) akkor ez körülbelül 6 egység (mikrosecundum) ideig tart. Ha viszont DMA üzemmódot választunk, akkor ez lényegesen gyorsabban, 1 egység időtartam alatt zajlik le. (A vezérlő egészen addig nem csinál semmit, amíg valamelyik DMA kérő vonalán (DREQ) H szint jelenik meg. — Ez jelenti a DMA kérést.— Ekkor nyugtázó jelet küld a perifériának (DACK). Ezzel egyidőben a periféria az adatbuszra teszi a szóbanforgó adatot. Ezalatt a processzort letiltja,majd beállítja a címvonalakat és kiadja a MEMW jelet. Mivel az input/output egység megszólításakor a memóriacím is jelen van, a perifériának tudnia kell, hogy ez nem

neki szól, ezért kezelnie kell az AEN (adres enable) vonalat is. Természetesen ebben az esetben az AEN letiltja a perifériát.



6.29. ábra

A 8237A jelü DMA vezérlő blokkdiagramja.

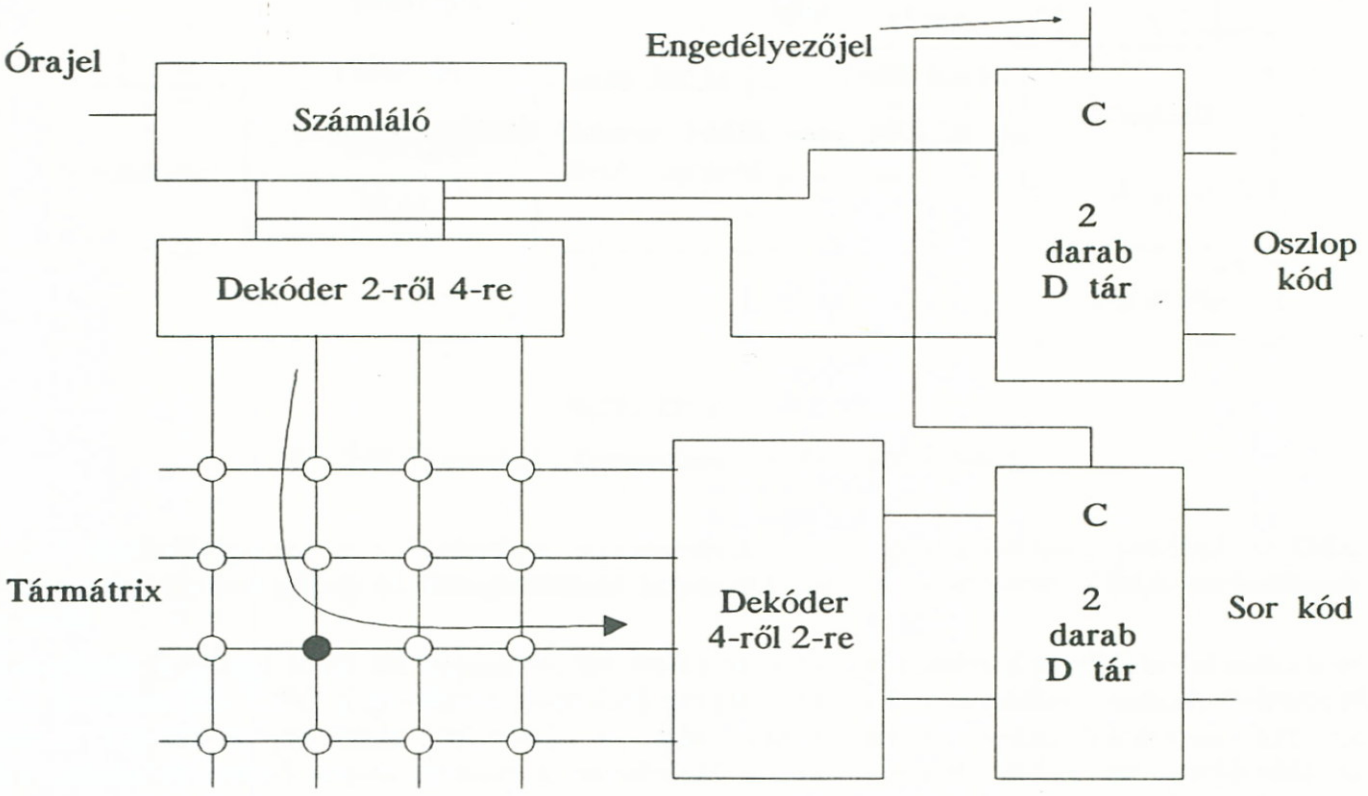
7. Számítógép perifériák

A következőkben a mikroszámítógépek gyakoribb perifériáinak felépítését és működési elveit fogjuk tanulmányozni. Kezdjük mindjárt az egyik legfontosabb egységgel, a billentyűzettel.

7.1. A billentyűzet

Vizsgáljuk meg, hogy a klaviatúra hogyan hozza a számítógép tudomására azt, hogy melyik billentyűt nyomtuk le rajta.Ez egyáltalán nem látszik egyszerű feladatnak, hiszen igen sok billentyűt tartalmaz (84, 101, 102 darab).

Eddigi ismereteink birtokában azonban már nem is olyan elrettentő a probléma,hiszen ha jól meggondoljuk ennyi billentyűt csak a memóriáknál látott mátrix elhelyezésben lehet egyszerűen kezelni. (7.1. ábra) Természetesen a sorok és oszlopok találkozásánál most kapcsolókat találunk, amelyek lenyomott állapotukban összekötik ezeket a vezetékeket. A sor és oszlop meghatározásához ugyancsak a memóriáknál már látott dekóderekre van szükség. Most már csak az a kérdés,hogyan derül ki, hogy a **billentyűmátrix** mely helyén nyomtunk le kapcsológombot. Ezt a feladatot úgy tudjuk megoldani, ha folyamatosan végigkérdezzük, végigtapogatjuk a mátrix elemeit. (Erre szolgál a számláló, amely a beérkező órajeleket számlálja és küldi az oszlopdekódernek.) Ehhez elegendő az oszlopokat



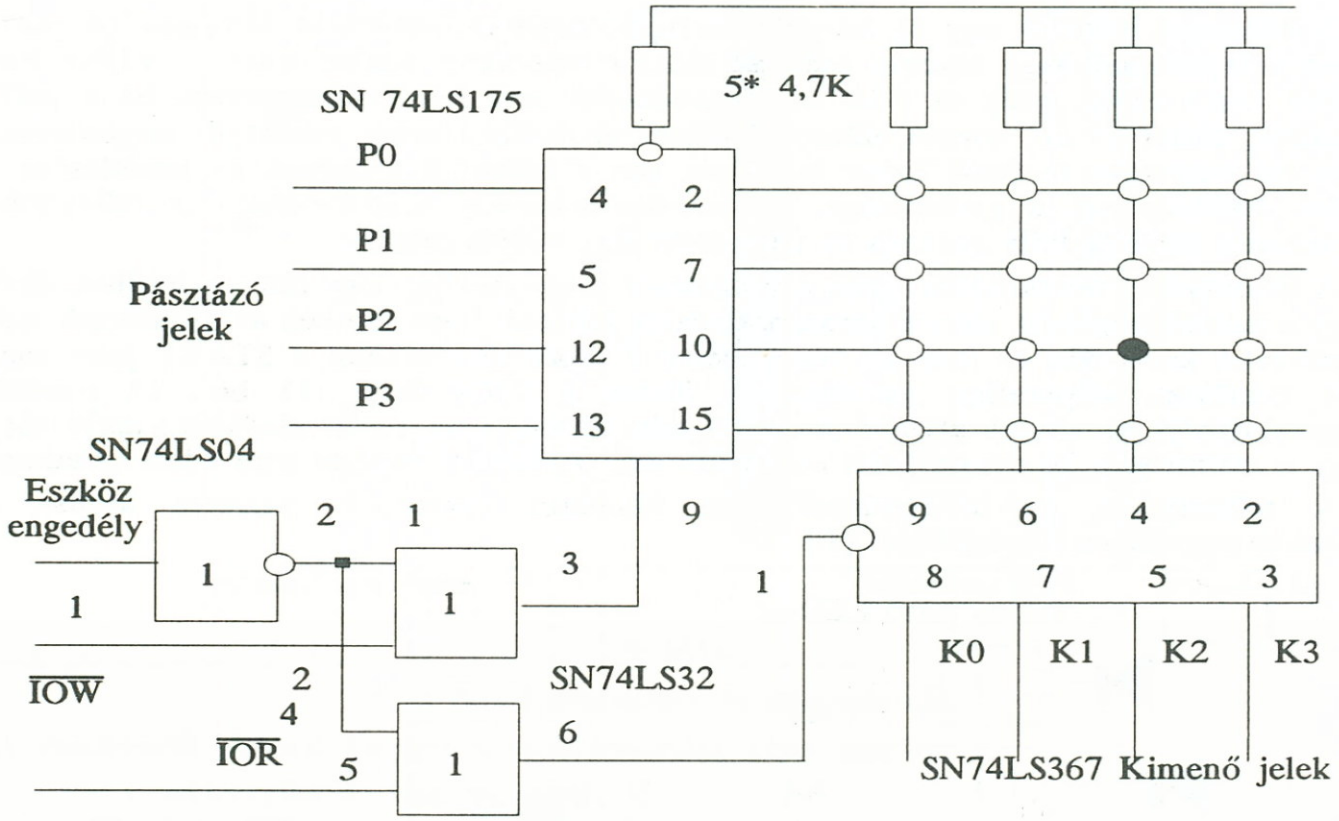
7.1. ábra

A billentyűzet elvi felépítése.

végigpásztázni egy impulzus segítségével, hiszen ha a megfelelő kapcsoló le van nyomva, akkor az impulzus az oszlopról átszalad a sorra és az itt található dekóder meghatározhatja, most már bináris formában, hogy mely sorban volt lenyomva a kapcsoló.

Ez most már egyértelműen meghatározza a kapcsoló helyét a mátrixban,hiszen az oszlop helyét, amelyre az impulzust adtuk ismerjük, a sor amelyben volt lenyomott

kapcsoló dekodolásra került, vagyis rendelkezésünkre áll a lenyomott kapcsoló oszlop illetve sorcíme. Innen már csak a számítógép tudomására kell hozni a D tárolók tartalmát.Ez akár párhuzamos, akár soros formában történhet.



7.2. ábra

Egyszerű 16 gombos billentyűzet felépítése. Az integrált áramkörök melletti számok azok bekötésére utalnak.

Egy egyszerű 16 gombos billentyűzet felépítését mutatja a 7.2 ábra.

A számlálót a CPU által kiadott jelsorozat helyettesíti (pásztázó jelek). A megfelelő portkiválasztó (eszköz engedélyező jel) és $\overline{\text{IOW}}$ jelek hatására a D tárolóba (4 darab egyetlen tokban SN74LS175) beíródik a küldött adat, amely a sorokat fogja lekérdezni.Ez a következő adatküldésig a tárolóban marad. Az eszköz kimenetén akkor is jelen lesz, amikor a következő $\overline{\text{IOR}}$ jel érkezik. Ennek hatására a processzor leolvassa a kapcsolók állapotát. A háromállapotú meghajtókapukon keresztül (SN74LS367) az adat rákerül az adatbuszra. A processzor által kiadott sorkiválasztó és a leolvasott oszlopinformáció birtokában a lenyomott kapcsoló meghatározható. Például ha a kiadott adat 1101, (P0 — P3) akkor ez azt jelenti,hogy a 2. sort kérdeztük le. Amennyiben visszaolvasáskor 1101 (K0 —K3) jelet kapunk, akkor a lenyomott billentyű a 2. oszlop 2. sorában van.

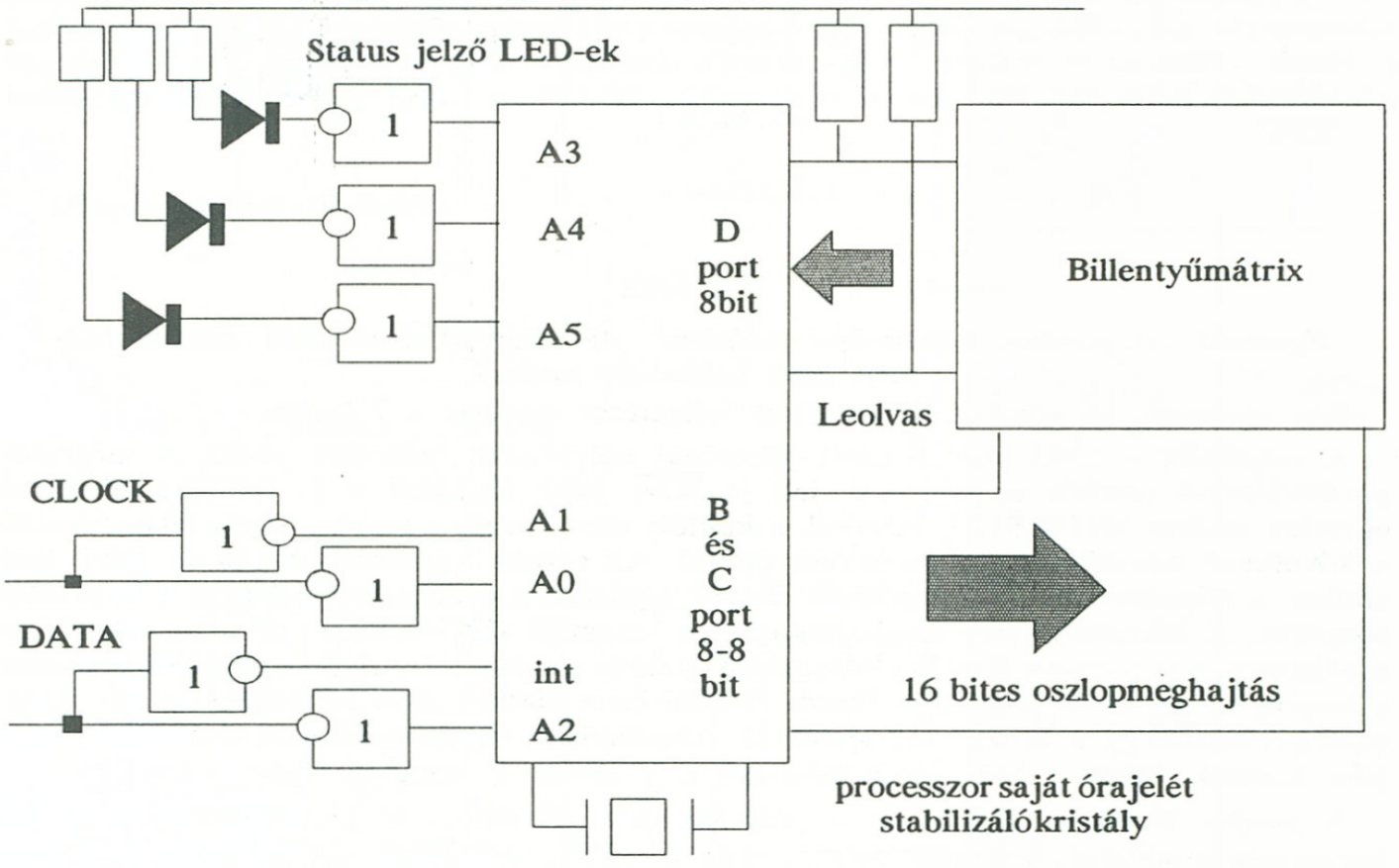
A pontos leolvasás érdekében a processzornak állandóan és periódikusan végig kell tapogatnia a sorokat, hogy észrevegye, hogy a mátrix tetszőleges helyén van-e lenyomott billentyű. Nyilvánvaló,hogy ez meglehetősen időigényes feladat, amire a központi egység nem vehető igénybe. Hiszen ha állandóan a billentyűzet lekérdezésével van elfoglalva, akkor mikor foglalkozik a többi feladatával? Ezért az intelligensebb billentyűzetet saját processzorral szokták ellátni, amely csak ennek a kiszolgálásával van elfoglalva. Az AT számítógépekben például egy HD6805V1 ,vagy I8049 jelű processzor gondoskodik a feladat ellátásáról.(7.3. ábra) Itt a tármátrixot 16 oszlopra és 8 sorra bontották. E processzor a B illetve C portján küldött oszlopmeghajtó jeleket. A D jelzésű portján olvassa vissza, kérdezi le a kapcsolómátrix sorainak az állapotát. A lenyomott billentyűt ennek alapján azonosítja a processzor. A számítógép felé csatlakozó vonalak (CLOCK,DATA) soros adattovábbítást tesznek lehetővé. A megfelelő keretbiteket is előállítja ez a processzor. A

CLOCK jel az átküldött adatok szinkronizálását biztosítja, előállításáról a processzor külön órajelgenerátora gondoskodik (4MHz). Mind a két vonal kétirányú, hogy a számítógép is adatot/parancsot tudjon küldeni a klaviatúra felé. Egyúttal ez azt is jelenti, hogy ezen vonalak állapotát a processzornak mintavételezni kell.

A billentyű leütések egy 16 karakter befogadóképességű pufferbe kerülnek (A puffer olyan tulajdonságú,hogy kivenni csak az először belerakott adatot lehet — **FIFO** First In,Firs Out.) addig, amíg az interface fogadóképes nem lesz.Természetesen, ha a puffer megtelik, akkor az azt követő billenytűleütések elvesznek.Ha egy billentyűt meghatározott időn túl lenyomva tartunk, akkor lehetőség van a billentyű kódjának az ismétlésére. A kódok kiküldésének a gyakorisága, az ismétlés sebessége szabályozható (a billentyűzet portcímére küldött F3H parancs és paraméter bájt segítségével).

A billentyűzet adatküldése során a processzor megvizsgálja, hogy nincs-e letiltva, illetve hogy a számítógép nem akar-e parancsot/adatot küldeni. Ilyen esetben az átküldendő adat a pufferba kerül. Ha ez nem áll fenn, akkor a processzor kiküldi a START jelet, majd a 8 adatbitet, a páratlan paritáskódot, illetve a STOP bitet (11 bit), (A rendszer megszakíthatja az adatok átvitelét,ha a 10. adatjel még nem került kiadásra.) majd várja, hogy a számítógép 20 ms-on belül nyugtázza azt. (Ellenkező esetben újra küldi az adatot.)

A billentyűzet és a rendszer által kiadható parancsok részletes leírása az irodalomjegyzékben megtalálható. [3.].

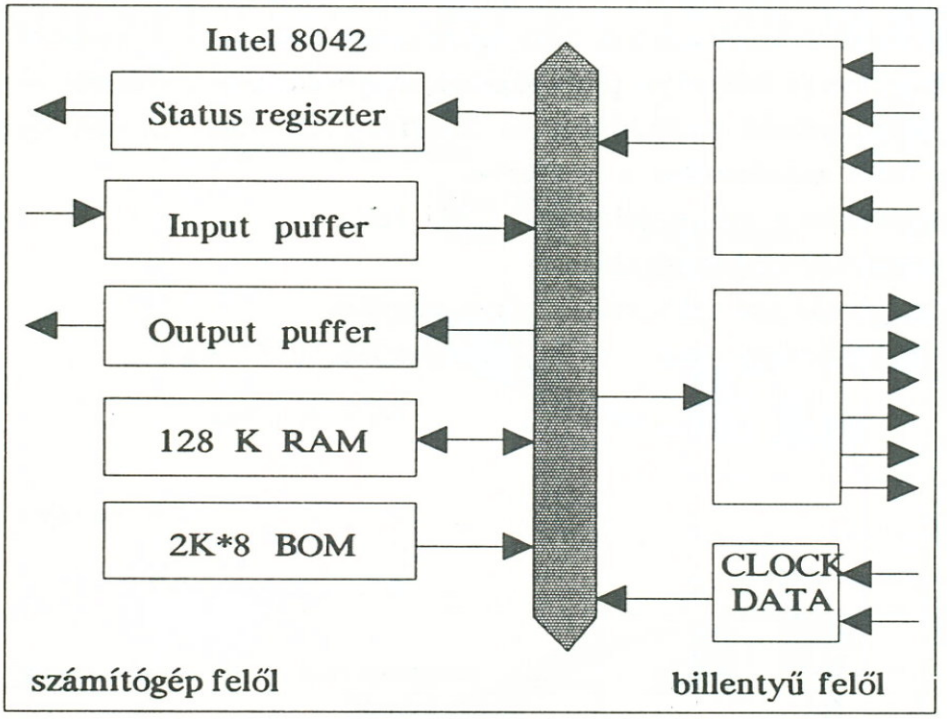


7.3. ábra

Az AT típusú számítógépek billentyű processzorának portjai.

A számítógép rendszer részéről az adatok fogadását a I8042 billentyűzetvezérlő interfacen keresztül történik. Az áramkör feladata, hogy vegye a billentyűzet felől érkező jeleket, ellenőrizze ezek paritását, értelmezze és a rendszer számára feldolgozható formára alakítsa az adatbájtot. Küldjön utasításokat a billentyűzet számára, illetve tiltsa le annak működését a gép előlapján található kapcsoló zárt állása esetén (billentyű zár).

Az áramkör elvi felépítése a következő. (7.4. ábra)



7.4. ábra

Az I 8042 processzor blokkdiagramja.

A rendszerbe érkező (Input) adatok kiosztása bitek szerint:

- 7. billentyűzetet tiltó kapcsoló,
- 6. monitor adatpter típusa,
- 5. gyári jumper,
- 4. az alaplapon elhelyezett RAM jumper, 512K esetén 0, 256K esetén 1,
- 3. 2. 1. 0. bitek fentartottak.

A kimeneti (Output) bitek kiosztása:

- 7. billentyűzet adat (kimenet),
- 6. billentyűzet órajel (kimenet),
- 5. a bemeneti puffer üres,
- 4. a kimeneti puffer megtelt,
- 3. és 2. bit fentartva,
- 1. az A20 címvonal kapujele,
- 0. általános nullázójel.

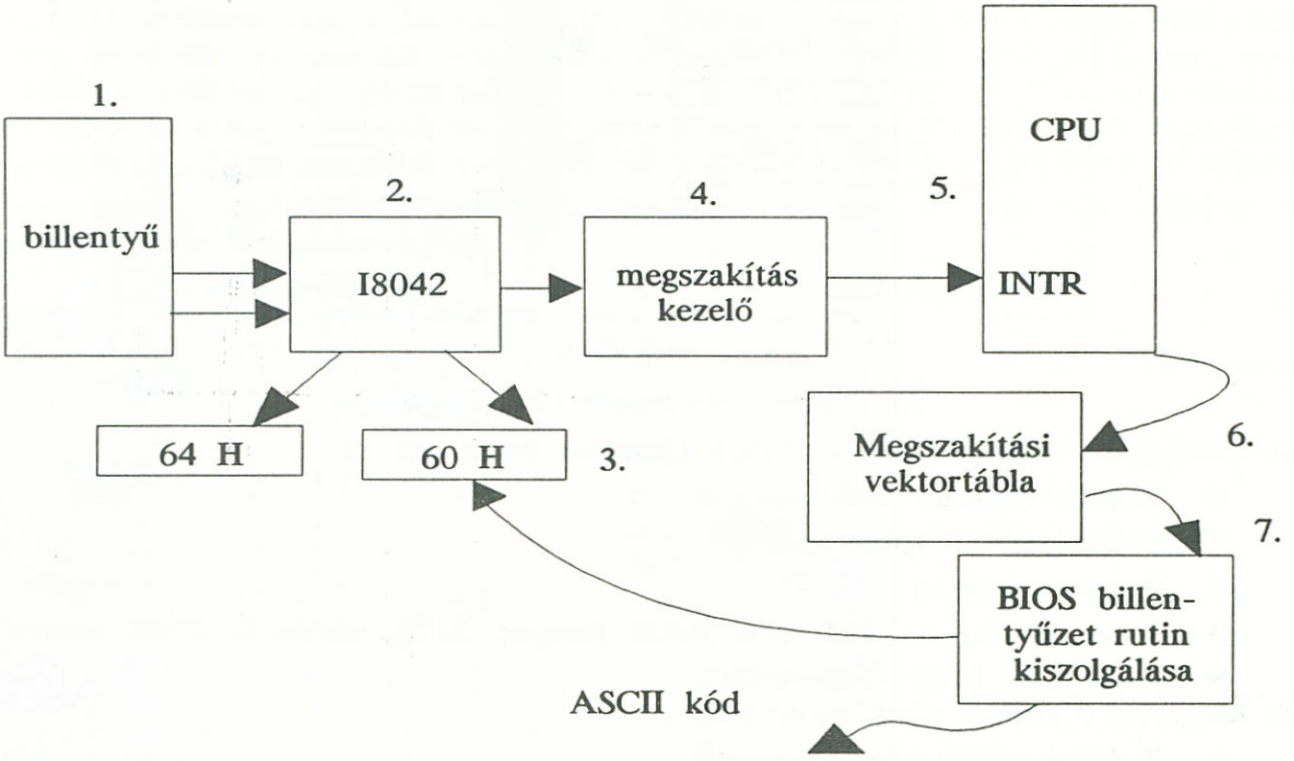
A billentyűzet felé küldött utasításokat a processzor (ha a bemeneti puffer üres) beírja a pufferbe,felszereli keretbitekkkel, majd teszteli a billentyűzet fogadókészségét, (nem továbbít-e adatot, illetve azt, hogy az adattovábbítás megszakítható-e) majd átküldi őket és várja a küldött adatok nyugtázását. A vezérlő a számára 64 H címre elhelyezett bájtot vezérlő utasításként, a 60 H címre érkezőt adatként értelmezi. Ezt beírni csak a státuszregiszter 1 bitjének 0 értéke esetén lehet.

Ha a billentyű felől adat érkezik a pufferba és az üres, akkor az áramkör ellenőrzi annak paritását. Amennyiben ez hibás,úgy ismétlő jelet küld (Resend) a billentyűzet felé. Ha az adat vétele során ilyen hibát nem tapasztal, akkor átalakítja a rendszer számára

értelmes scan kóddá. Megszakításkéréssel fordul a CPU felé, hogy az dolgozza fel a küldött adatot. (A kimeneti puffer a 60 H címen található.)

A billentyűzet felől érkező adatok feldolgozásának menete a következő:

» 1. a lenyomott billentyű pozíciójának meghatározása, (billentyűzetben)
» 2. a billentyűkód továbbítása az interface-nak, (ha az lehetséges)
» 3. az adat elhelyezése a pufferbe,
» 4. megszakítási igény jelzése a CPU felé,
» 5. a megszakítás kiszolgálása,
» 6. a megszakítási vektortábla címe alapján,
» 7. a BIOS (vagy saját) rutin meghívása. (7.5. ábra)



7.5. ábra

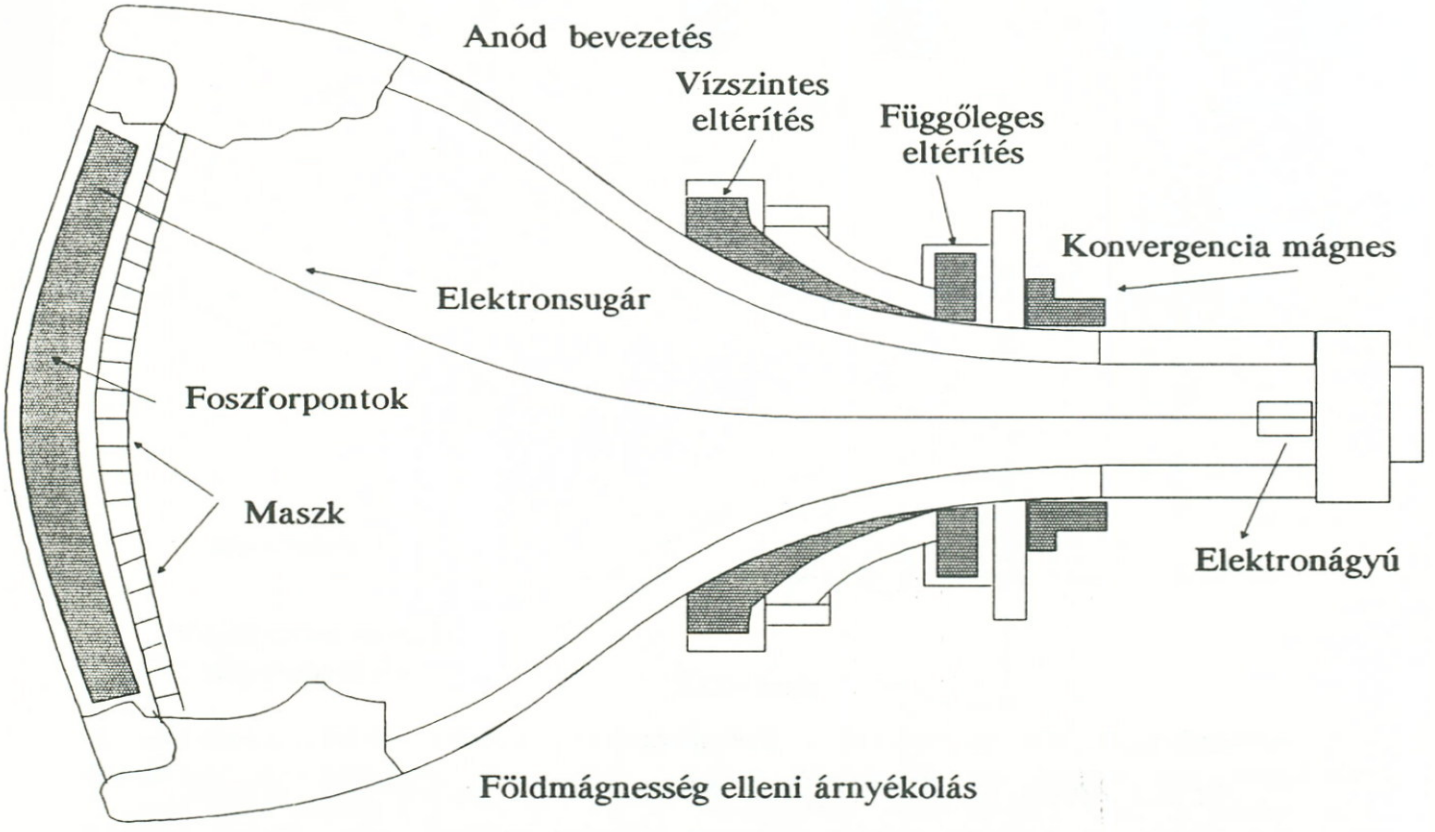
A billentyűzet kiszolgálásának a folyamata.

7.2. A monitorok

Mielőt belekezdenénk annak ismertetésébe, hogy hogyan alakítják ki a képet a monitorok és ezek vezérlő kártyái, idézzük fel a képátalakító katódsugárcső fizikáját.

A kép lényegében egy speciálisan kialakított katódsugárcsőben képződik. Ennek felépítését mutatja a 7.6. ábra. Az elektronágyúból kilépő elektronsugár az ernyő felületére becsapódva utánvilágítást okoz a látható tartományban. Ha az elektronsugarat elmozdítjuk, akkor az nyomot hagy a képernyőn. A sugár mozgatása mind elektromos, mind pedig mágneses úton megoldható. (Az $F = EQ$ és az $F = BQv$ erőtörvények értelmében, a töltésre — így a mozgó elektronra is— erő hat. E az elektromos tér erőssége, B a mágneses indukció, v a mozgó elektron sebessége.) A szokásos megoldás esetén mágneses eltérítést használnak. (Ekkor az eltérítő erő merőleges mind az indukcióra, mind pedig az elektron sebességvektorára is.) Az elektronsugárnak a képernyő teljes felületének bejárásához két irányú, vízszintes és függőleges eltérítő mágneses erőtérre — tekercsre — van szüksége.

Az elektronsugár által gerjesztettt képernyő színéért az ernyőt beborító anyag a felelős.A monochrom monitorok esetén ez egyféle (zöld, sárga, vagy papírhehér színt kibocsájtó vegyület), míg színes monitorok esetén ez háromféle (vörös, zöld, kék alapszínt kibocsátó) anyagot jelent.



7.6. ábra

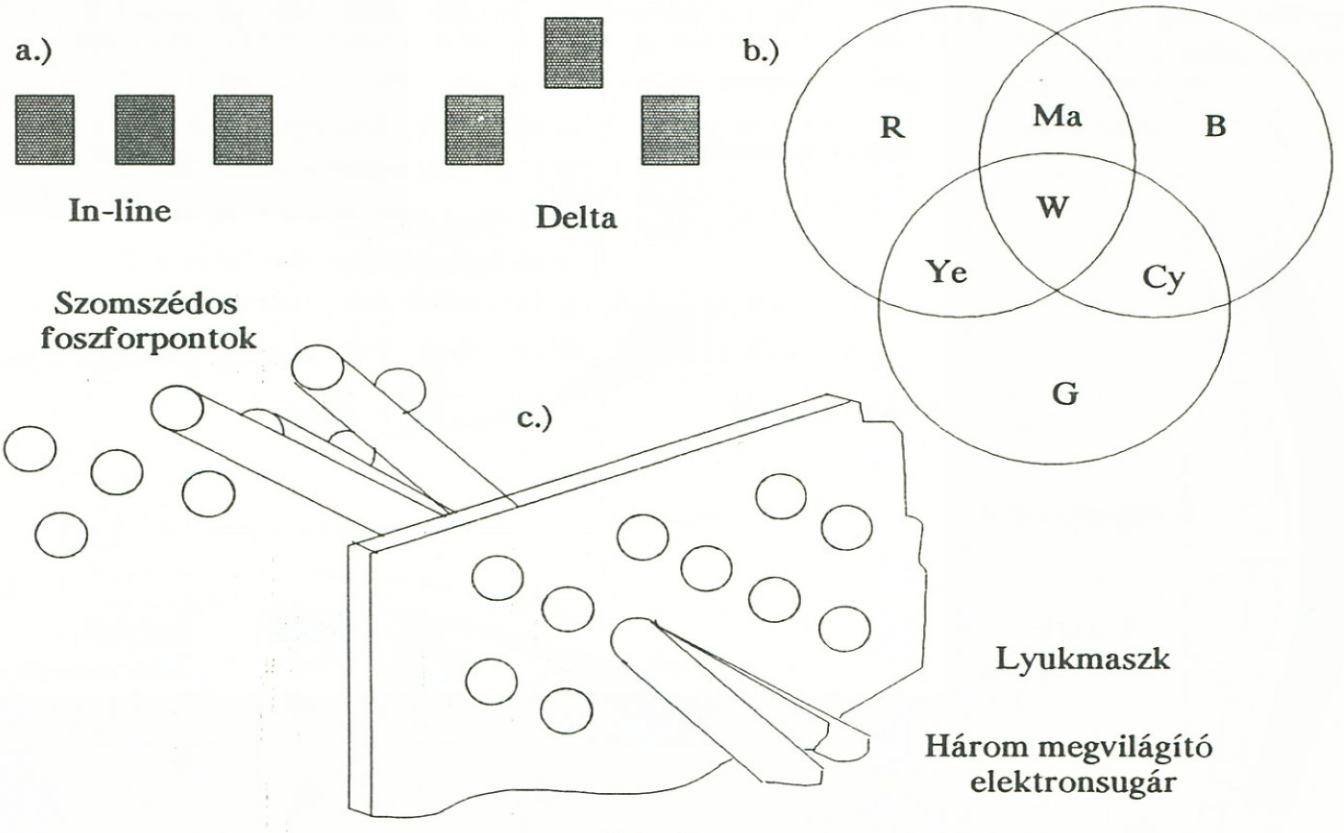
A képcső felépítése.

A színes képcsövekben minden alapszínhez egy - egy elektronsugár tartozik. A kibocsátó elektronágyúk elhelyezése alapján beszélünk in-line és delta képcsövekről. A színek kikeverését az alapszínekből additív módon oldják meg. Az a terület, amelyen két vagy több alapszínt gerjesztünk más színben fog látszani. Például a vörös és kék szín gerjesztése esetén bíbor, míg a zöld és vörös együttes „megvilágítása" esetén sárga színt tudunk kikeverni. Az összetevők erősségének állításával ezek színárnyalata is megváltozik. A kérdés csupán az,hogy hogyan valósíthatjuk meg fizikailag,hogy minden képpont bármilyen színben világítson? Ehhez a képpontok megfelelő elhelyezése és ezek alkalmas „megvilágítása" szükséges. A színpontokat az ernyőn sorban egymás mellé és egy ponttal eltolva egymás alá helyezik el. Ha a színek jelölésére az angol nevük kezdőbetűjét használjuk,akkor az 7.7. ábrának megfelelő színpont elhelyezést találjuk. Az alkalmas „megvilágításról" az ernyő elé elhelyezett lyukmaszk gondoskodik. (Annyi furat van rajta, amennyi képpont az ernyőn.) Ennek ernyőtől mért távolságát a következő módon választják meg. Ha mindhárom elektronsugár egyetlen maszkpontot „világít" meg, akkor azok azon áthaladva éppen a három alapszínt kibocsájtó, egymáshoz legközelebb elhelyezkedő színpontokba esnek.

B R G B R G B R G
R G B R G B R G B
B R G B R G B R G

7.7. ábra

R= red, R=blue, G=green képernyőpontok elhelyezkedése

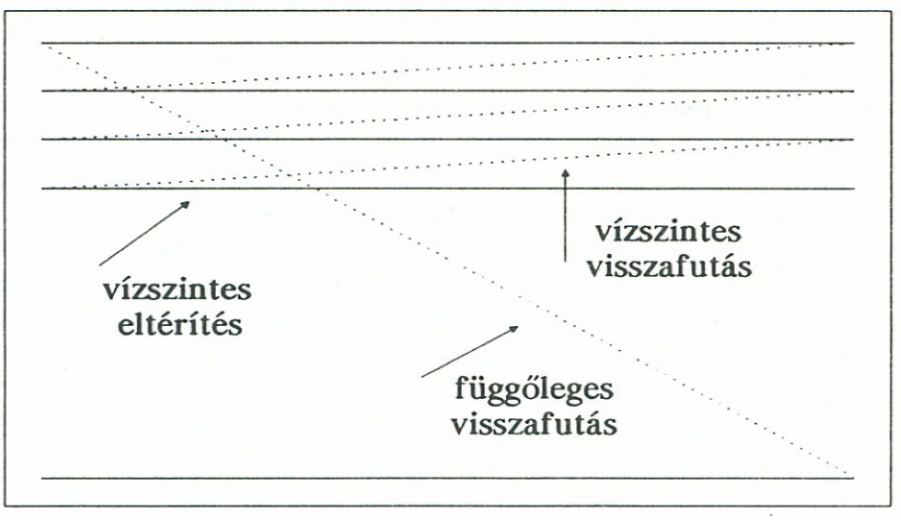


7.8. ábra

Az a.) ábrán az elektronágyuk elhelyezésének geometriája látható.

A b.) ábrán az addittív módon kikevert színeket találjuk meg. B=kék (Blue), R= vörös (Red), G=zöld (Green), Ye=sárga (Yellow), Ma=bíbor (Magenta), Cy=zöldeskék (Cyan), W=fehér (White).

A c.) ábrán a lyukmaszk és a rajta áthaladó elektronsugarak által megvilágított szomszédos foszforpontokat láthatjuk.

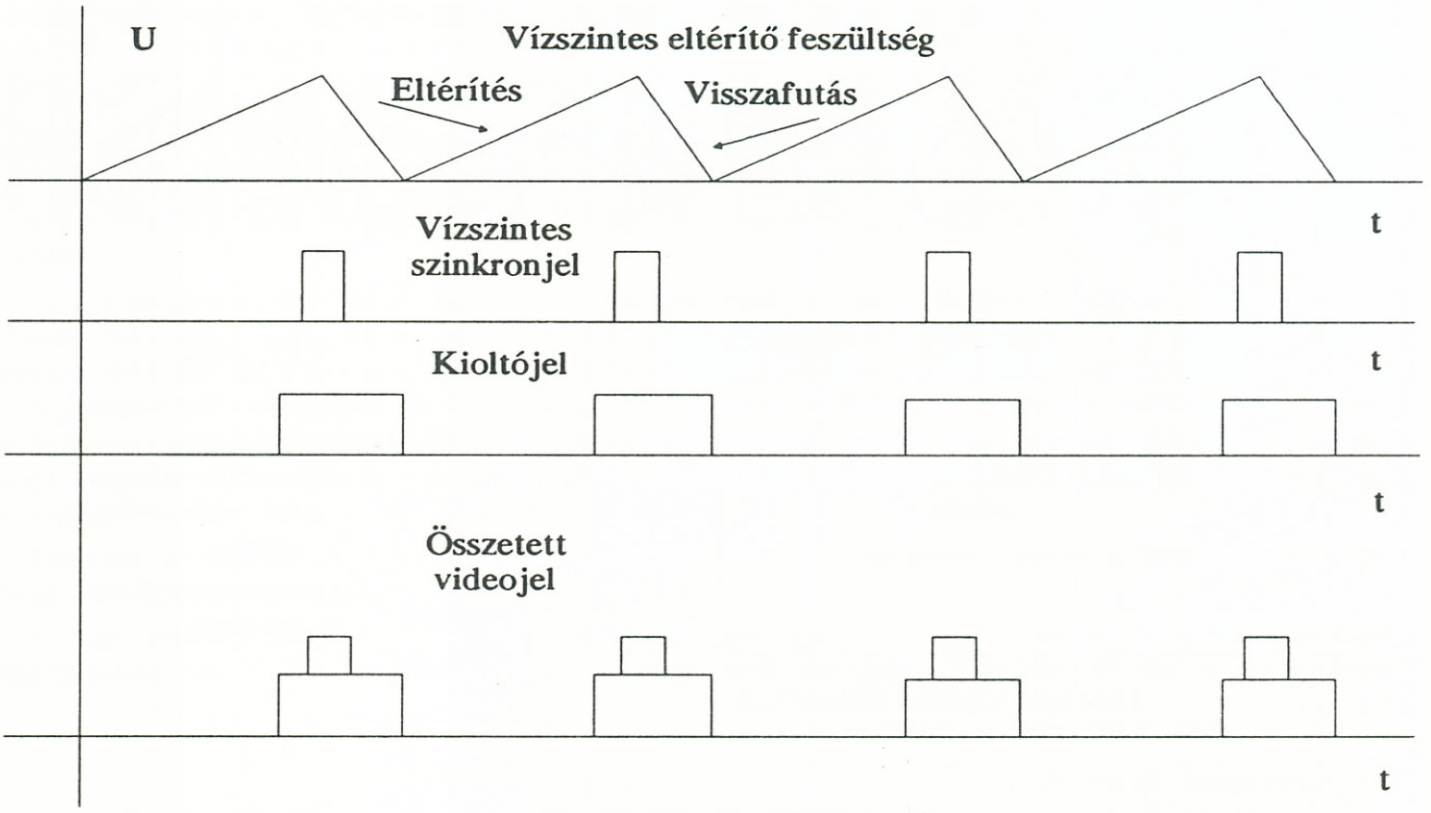


7.9. ábra

Elektronsugár pályája

Ezek után vizsgáljuk meg közelebbről a kép kialakításának módját.Kezdjük vizsgálatunkat az elektronsugár mozgatásának vizsgálatával. A fénypont mozgását (az elektronsugár pályáját) úgy vezéreljük, hogy az elektronsugár a képernyő bal felső pontjából indulva vízsszintesen befutja az első sort, majd visszafut a következő sor elejére, innen eljut a második sor végére és így tovább, egészen a képernyő aljáig, ahonnan visszaugrik a bal felső sarokba.

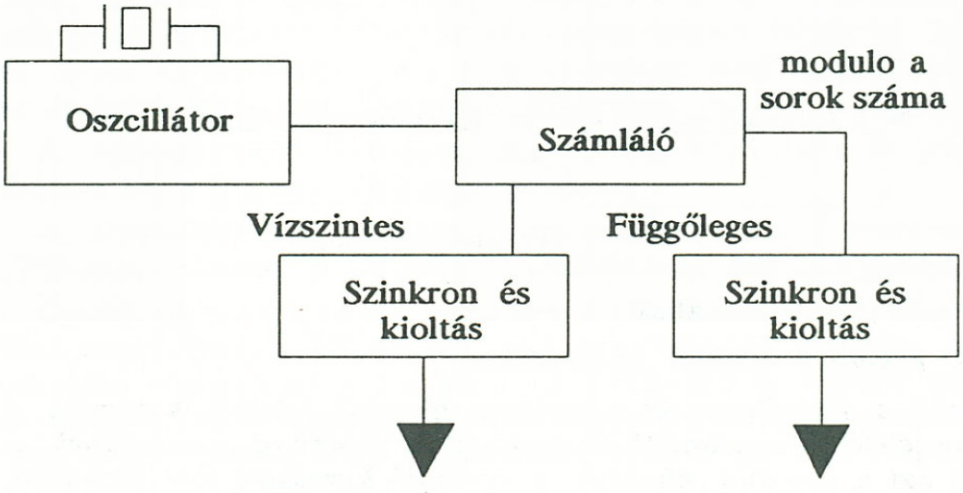
A megoldás csak akkor tökéletes, ha a visszafutáskor az elektronsugár nem hagy nyomot, azaz arra az időre az elektronsugarat kikapcsoljuk. (7.9. ábra)



7.10. ábra

A vízszintes eltérítő tekercs feszültségének változása az idő függvényében. Látható,hogy a visszafutás nem akár mikor következik be, hanem egy meghatározott szinkronimpulzus váltja ki. A visszafutás alatt kioltás van, a kioltó jel mintegy átfogja a szinkronimpulzust.

A kép összeállításához a vízsszintes és függőleges letapogatási frekvencia pontos beállítása szükséges ugyanis, kisebb vagy nagyobb értéke a kép összetöréséhez vezet. Éppen ezért a vezérlő oszcillátorok (rezgőkörök) nem a saját, hanem a **szinkronjel** segítségével beállított frekvencián rezegnek. (7.10. ábra) A visszafutás alatt, mint tudjuk az elektronsugarat ki kell oltani, erre szolgál a vízsszintes kioltóimpulzus, ami tulajdonképpen közrefogja a szinkronjelet. (7.11. ábra) A függőleges kioltó és szinkronimpulzus hasonló



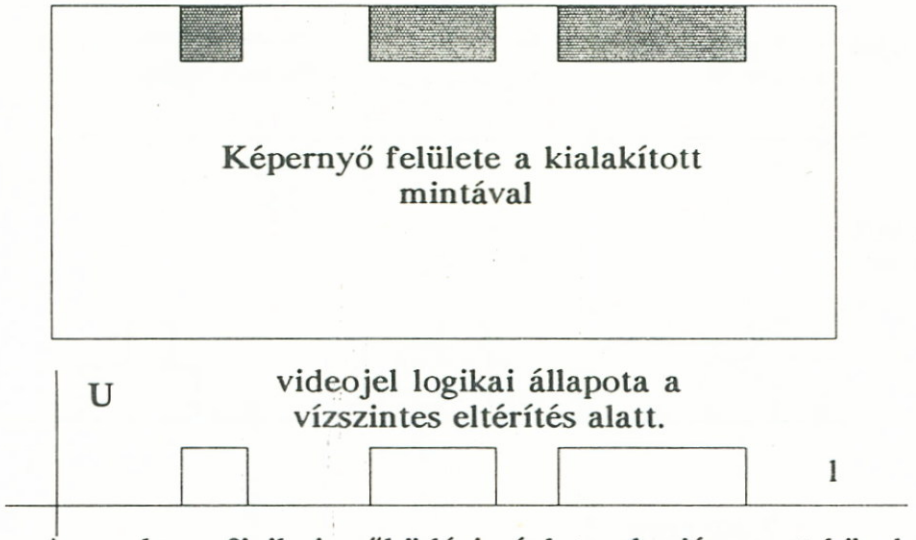
7.11. ábra

A vízszintes és függőleges szinkron és kioltójelek előállításának blokk vázlata.

A függőleges kioltást az utolsó vízszintes kioltóimpulzus kezdeményezi. Olyan számláló áramkört kell készíteni amely a megfelelő sorszám után előről kezdi a számlálást — modulo a sorok száma — és függőleges kioltó impulzust ad.

a vízszinteshez,de észre kell vennünk, hogy ennek bekövetkezését az utolsó vízsszintes kioltójel fogja indítani.

Ezt követően már csak az elektronsugár megfelelő modulációja van hátra, vagyis a vízsszintes eltérítés alatt olyan módon kell kapcsolgatnunk az elektronsugarat ki-be, hogy a kép mintázata kialakuljon. Ezt úgy szokás megtenni, hogy az elektronsugár bekapcsolt állapotához rendeljük a videojel logikai 1, kikapcsolt állapotához a videoljel logikai 0 értékét. (7.12. ábra)

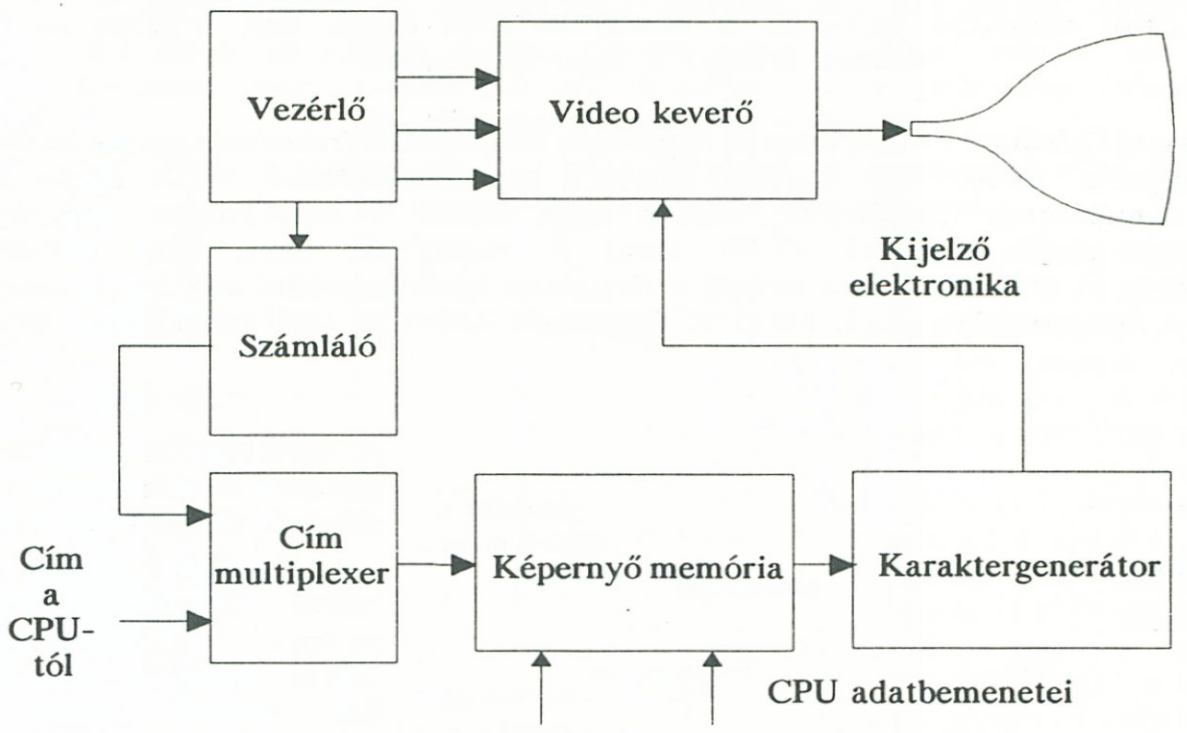


7.12. ábra

A videojel kialakítása. Logikai 1 állapot tartozik a képernyő világos, és 0 a sötét tartományaihoz.

(Negatív modulációnál ez fordított.)

A rendszer fizikai működési vázlata alapján most közelebbről megvizsgálhatjuk a vezérlő elvi vázlatát. (7.13. ábra)



7.13. ábra

A képernyő vezérlő blokkvázlata.

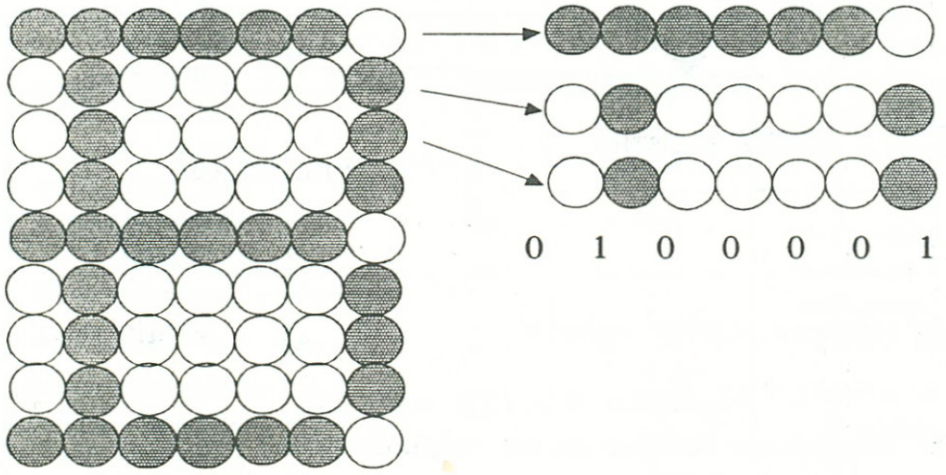
A vezérlő egység állítja elő a függőleges és vízszintes eltérítő jeleket, valamint a működéshez szükséges szinkronjeleket. A számláló szolgáltatja a képernyő memóriának a soron következő címét, vagyis azt a pozíciót, ahonnan a kiírandó karaktert kell kiolvasni.

A képernyő memória tárolja az általunk bevitt és megjeleníteni kívánt karaktereket,pontosabban ezek kódjait. Ezek a kódok úgy helyezkednek el, hogy minden egyes képernyőpozícióhoz tartozik egy tárcím. Például 25 sor és 80 oszlopos kijelzés esetén a bal felső képernyőpozícióhoz az első, a sor végéhez a 80.,a következő sor elejéhez a 81. és így tovább a 25. sor végéig. (Komolyabb megoldás esetén nemcsak a karakterek kódja tárolódik, hanem ezek attribútumai is, például az egymást követő bájtokon a kód, majd a saját attributuma helyezkedik el.) Úgy is mondhatjuk, hogy a kijelzés sorfolytonosan történik.

A **karaktergenerátor** ból vett információsorozat alapján tudjuk előállítani a képernyőmemóriában elhelyezett kódok geometriai képét. Mégpedig oly módon,hogy a képernyőtárba beírt karakterkódok relatív címként szolgálnak a karaktergenerátor elemeinek kiválasztásához. A generátor felépítésétől függően a karakterek képei különböző méretű bitmátrixban tárolódnak. (7*9, 8*8 stb.) A bitmátrix elemei úgy határozzák meg a képet, hogy ahol a mátrixban 1 szerepel ott a képernyő megfelelő pozíciójában bekapcsol egy fénypontot.

Például az MCM6575 sorszámú generátor 7*9-es formában tárolja a karakterek képét (lásd az 7.14. ábrát).

E tár címzéséhez, mivel csak 128 karaktert tárol, 7 címző vezeték szükséges.Ezek segítségével jelölhető meg a tárban a megfelelő karakter kezdete. Mivel a karakterek



7.14. ábra

*A B betű bitmátrixa a 7 * 9-es rendszerben.*

A betű sorai egymást követő bájtokon tárolódnak. A bájt értékét a „világító pontok" — 1-esek — elhelyezkedése határozza meg.

képe csak azok sorainak letapogatásával áll elő, így az áramkört a karaktersorok kereséséhez szükséges 4 (RS0 . . . RS3) vezérlővezetékkel látták el. (A négy vezérlővezetékkel 16 sor lenne kiválasztható. Ha ezen vezetékek által meghatározott szám 9-nél nagyobb, úgy az áramkör kimenetei logikai 0 állapotban vannak, a képernyő sötét.)

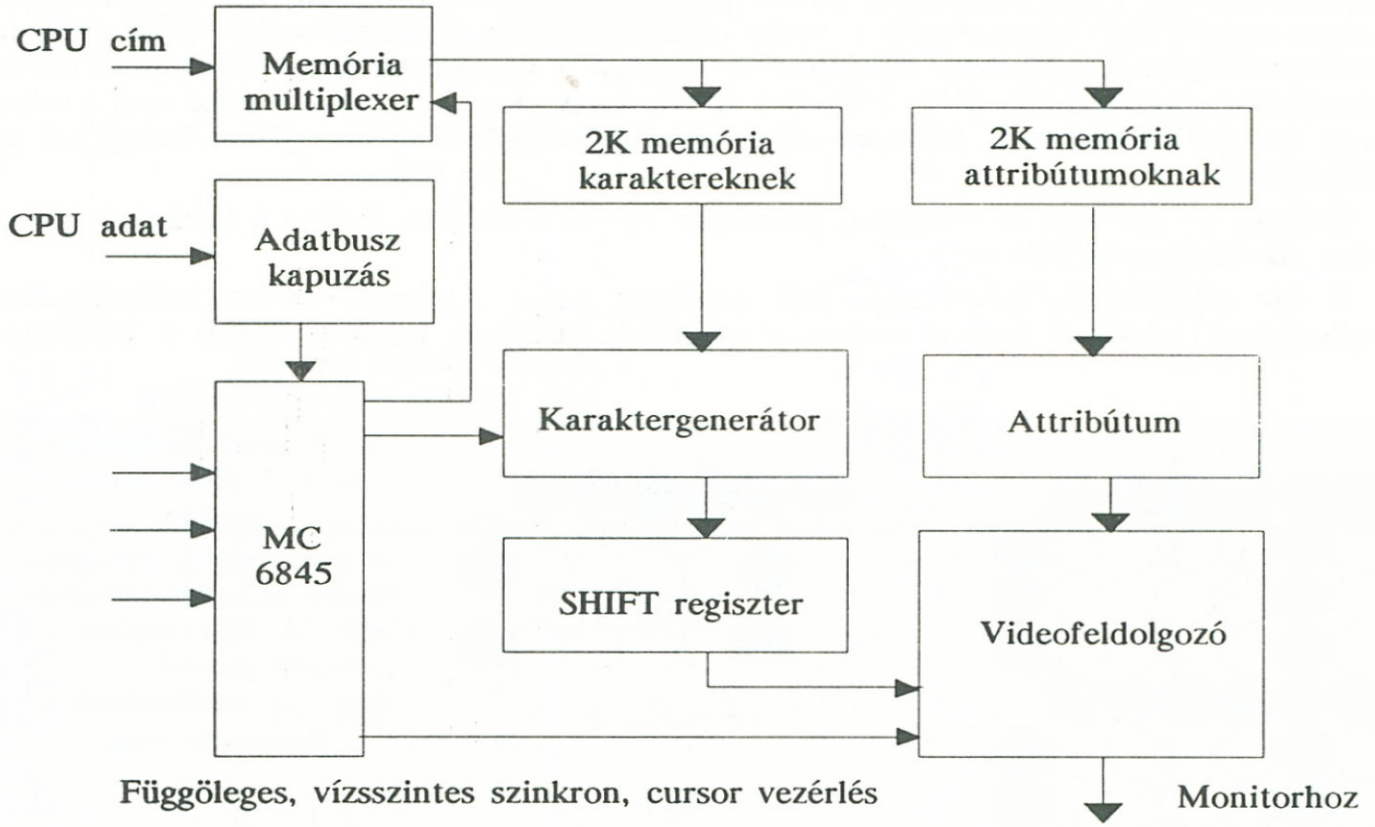
A **videokeverő** feladata, hogy a beérkező digitális jelekből fekete-fehér videójelet állítson elő a kijelző elektronika számára.

A címmultiplexer feladata, hogy megoldja azt a problémát, mely szerint időnként a CPU-nak, időnként a letapogató számlálónak kell a kijelzőtár elemeihez hozzáférni.

Összefoglalva a kijelzés folyamata a következő módon megy végbe. A CPU a multiplexer által engedélyezett időpontban beír egy karaktert a tárba. A következő időpillanatban a számláló végigszalad a tár teljes tartományán és amikor az előbb felírt karakterkódhoz ér, akkor a karaktergenerátor tábla megfelelő helyéről kiemeli a kiválasztott karakter képét. Ami annyit jelent, hogy a generátor vezérlővezetékei segítségével letapogatja a karakter

sorait. Ez a folyamat az elektronsugár mozgásával együtt zajlik. A teljes karakter megjelenítése a jel kezdetétől számított 9. vízszintes sor kirajzolása után fejeződik be.

Felmerülhet a kérdés, hogy nem zavarja-e a kép letapogatását és kiírását az, hogy időnként a CPU megzavarja ezt a folyamatot azzal, hogy időnként a tárhoz nyúl — karakter kódot ír bele. Nos ez egyáltalán nem zavaró, hiszen a CPU felül érkező adatok nagyon gyorsan foglalják el a helyüket a tárban a kijelzéshez képest. Amit úgyis mondhatnánk a kijelzés folyamata a memórába írás számára átlátszó, a folyamatot nem zavarja.Konkrét példa kapcsán tekintsük meg egy monitor adapterkártya felépítését. (7.15. ábra) Ennek elemzését ki-ki önállóan elvégezheti.



7.15. ábra

Az IBM monochrom adapterkártya vázlata.

A monitorok kapcsán szót kell még ejteni néhány káros tényezőről is, amelyek a képernyők előtt hosszú ideig ülőket veszélyezteti.

- Porártalom. Ez az üzemelő képernyő felülete és az emberi test között fellépő potenciálkülönbségnek a következménye. A képernyő magas (több 1000 V-os) feszültsége a levegő molekuláit polarizálja és így magához vonza őket. Ezek a képernyő felületét elérve feltöltődnek — az ernyővel azonos töltésre — és ezért gyorsan távolodnak, (az azonos töltésű testek taszítják egymást) mégpedig az alacsonyabb potenciálú hely felé haladnak. (Az ember teste,szeme felé.)
  Az áramló molekulák magukkal sodorják a porszemcséket, amelyek ha a szembe jutnak gyulladást okozhatnak.
- A képernyő csillogásának káros hatásai. (A szemmozgató izmok kifáradását okozhatják.)
- Elektromágneses és a másodlagos elektronsugárzás következményeivel is számolnunk kell. Ez a szervezetet alkotó molekulákat ionizálhatja illetve polarizálhatja.

E káros hatások jelentős része erőteljesen leszűkíthető alkalmas monitorszűrők használata segítségével. A földelt polárszűrő eddig a legkielégítőbb megoldást adja, ugyanis az eszköz földelése jelentősen csökkenti a porártalmat és az elektromágneses sugárzásra is árnyékoló

hatást fejt ki. A polárszűrő réteg pedig csökkenti, vagy teljesen megszünteti a képernyők csillogását. (pl: Polariod gyártmányú szűrők.)

7.3. Mágneses adathordozók és adattárolók

A tárolóegységek felépítésének megismerése előtt tekintsük át a mágneses adattárolás fizikai alapjait. Középiskolai tanulmányainkból tudjuk, hogy a mágneses teret mozgó töltések, áramok keltik. Ez egyben azt is jelenti, hogy az anyagok mágneses tulajdonságait elemi áramok határozzák meg. (Amper) A kvantummechanika pontosan megmutatta, hogy ez így van, az elektronok mag körüli (pályamomentum), és saját mozgása (spinje) okozza az atomok mágneses momentumát, amely egyben forrása az anyagok mágneses jellemzőinek is.

Attól függően ,hogy a spinből és a pályamomentumból származó mágneses térjellemző (indukcióvektor) milyen viszonyban áll egymással, csoportosítjuk az anyagokat dia, para és ferromágneses anyagokra. (A diamágneses atomoknál nincs eredő erőtér — a spinből és a pályamumentumból származó mágenese indukcióvektor egymás hatását lerontja. — A paramágneseknél minden atom hordoz eredő mágneses teret, azaz dipólusként viselkedik. Az ilyen atomokból felépülő anyag azonban nem mutat mágneses tulajdonságot, mert az atomok mágneses irányultsága rendezetlen, így a terek egymás hatását megsemmisítik.)

Bennünket a továbbiakban a ferromágneses anyagok érdekelnek, hiszen bennük nagyszámú szomszédos atom mutat egyező mágneses irányítást. Az egyirányban mágnesezett tartományokat Weiss- celláknak szokás nevezni. (Méretük 0,001 — 0,1 cm^3 -ig terjed.) Az egész anyag kifelé itt sem mágneses, mert statisztikailag ezek a tartományok is rendezetlen irányításúak, azonban külső mágneses tér hatására beállnak a tér irányába és megőrzik azt.

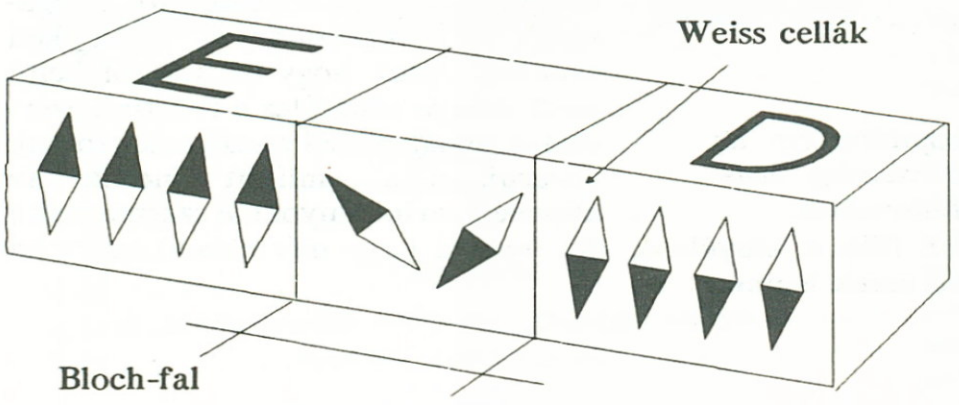
Az anyag megnevezése	Diamágneses	Paramágneses	Ferromágneses
Mágneses tere a spin miatt	→	→	→
Mágneses tere a pályamozgás miatt	←	←	←
Eredő mágneses tere	0	$\neq 0$	$\neq 0$
Szomszédos atomok terének irányultsága	Különböző	Különböző	Azonos,vagy közel azonos

7.1. táblázat

Az anyagok csoportosítása mágneses tulajdonságuk szerint.

A tartományok közötti átmenet nem ugrásszerű, hanem folytonos és a szomszédos rétegeket elválasztó átmeneti tartományokban (Bloch-fal) megy végbe.Az átmeneti tartományok mérete kisebb, mint 10^{-4} mm. A tartományok fokozatos átmenetét illusztrálja a 7.16 ábra.

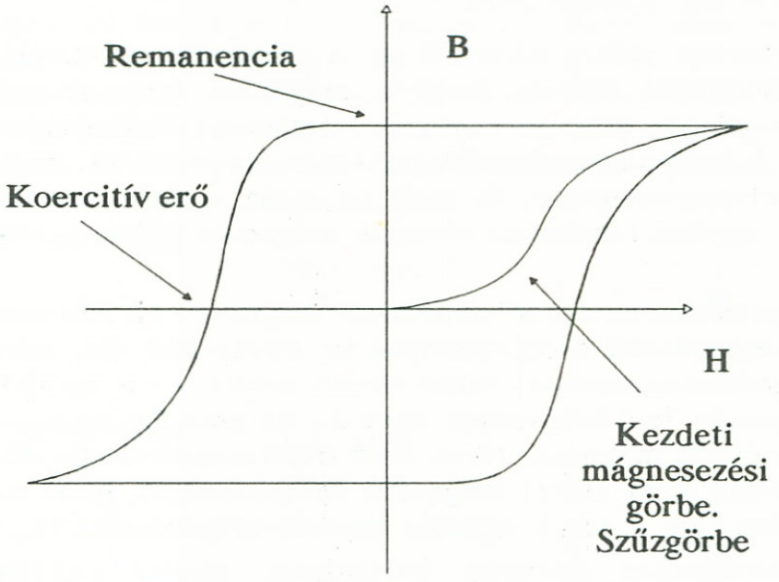
Ha külső mágneses tér hatásának tesszük ki ezeket az anyagokat, azaz mágnesezzük őket, akkor a gerjesztő erőtér változatatása mellett, a terek ábrázolásakor egy zárt görbét, úgynevezett hiszterézis görbét kapunk.(7.17 ábra)



7.16. ábra

A ferromágneses anyagok szerkezete az azonos irányítású Weiss cellákkal és az őket elválasztóBloch-fallal.

Megfigyelhetjük, hogy a gerjesztés megszüntével (H = 0) az anyag nem áll vissza a semleges állapotába, hanem visszamaradó **remanens** mágnességet mutat, megőrzi a gerjesztő tér irányát. (Ez a jelenség használható fel az adatok rögzítésénél.) Ezt a visszamaradó teret megszüntetni csak ellentétes irányultságú külső mágneses mezővel lehet. Ennek a lemágnesező erőtérnek a nagyságát **koercitív erő**nek szokás nevezni.

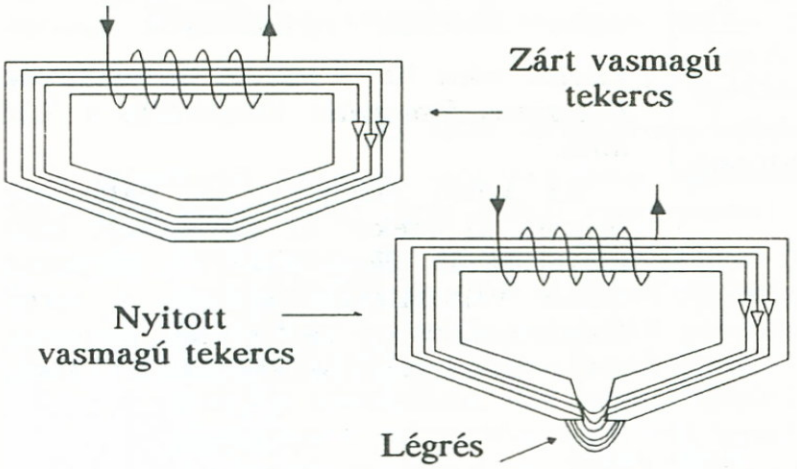


7.17. ábra

A ferromágneses anyagok felmágnesezési görbéje, a hiszterézis görbe.

Mind a két jellemző (remanencia és koercitív erő) alapvetően meghatározza a mágneses adathordozók tulajdonságait.

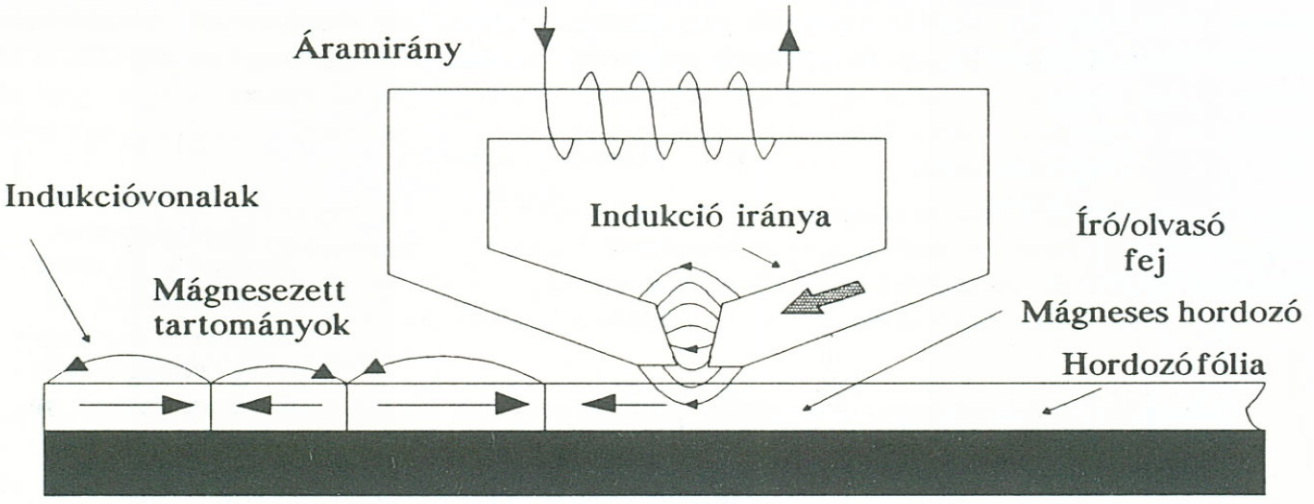
Ennyi előismeret birtokában most már bátran rátérhetünk az adatrögzítés módjának ismertetésére. Első közelítésben a bináris adatok tárolása azt jelenti, függetlenül a hordozó jellegétől (szalag illetve lemez), hogy egy mágneses mintázatot rögzítünk a mágnesezhető réteggel bevont anyagba. Ezt elektromos úton tehetjük meg a legkönnyebben. Ennek alapgondolata arra a jelenségre vezethető vissza, hogy az áramjárta tekercs maga körül mágneses mezőt kelt. ($B = \frac{IN}{l}$,ahol I a tekercsen átfolyó áram N a menetszáma l a tekercshossza.) A mező erővonalai (indukcióvonalak) csapdába ejthetők zárt mágneses kör alkalmazásával. Ez azt jelenti, hogy a zárt, mondjuk kör alakú vasmagon kívül indukcióvonalakat és ezért erőteret nem találunk, az összes benne halad. Ha a mágneses kört nyitjuk, akkor az indukcióvonalak azonnal „kitüremkednek" és a vasmagon kívül is erőteret alakítanak ki. (7.18. ábra)



7.18. ábra

Zárt vasmagos tekercs csapdába ejti az erővonalakat, míg nyitott vasmag mellett a légrésen kidudorodnak.

Ha most a mágnesező fejet (áramjárta, majdnem zárt vasmagos tekercs) a mágneses adathordozó fölé helyezzük, akkor a fejen átfutó áram irányától függően mágneses mintázatot írhatunk fel rá. (7.19. ábra) Visszaolvasáskor az előzőekben beállított mágneses tartományt (Weiss-cellák) mágneses terét használjuk fel, mégpedig a mozgási indukció alapján. (A változó mágneses mező zárt tekercsben áramot indít. A mező változása az adathordozó mozgásának a következménye). Észre kell azonban venni, hogy az áramot keltő mező iránya ellentétes a felírttal, ezért ezeket lemágnesező mezőknek szoktuk nevezni. (Az említett mezők az átmeneti tartományokból származnak.) Mivel az adathordozók felszíne fölé is kinyúlnak, (ha nem is nagy mértékben) ezért ha keresztezik a fejet, áramot indítanak benne.



7.19. ábra

Mágneses jelrögzítés alapelve.

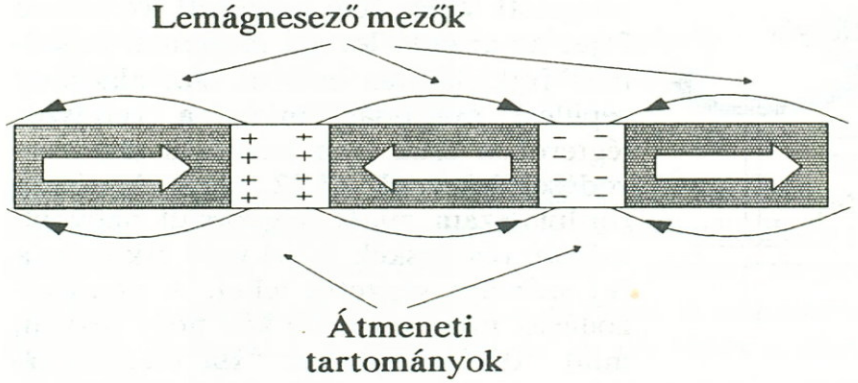
Az adatbitek és ezért a mágneses tartományok is sorban követik egymást. Ezért az adathordozó kapacitását elsősorban a **lineáris bitsűrűség** — egységnyi hosszúságra rögzíthető bitek száma — határozza meg.

Az adathordozók és adattárolók fejlesztésére vonatkozó törekvéseket a lineáris bitsűrűség növelése határozza meg. Kövessük végig ezeket a tendenciákat.

Az adathordozók tulajdonságainak javítása.

A mágnesezett tartományok sűrűsége azon múlik, hogy milyen közel tudjuk elhelyezni őket egymáshoz. Ez azt jelenti, hogy le kell csökkenteni az átmeneti tartományok (Bloch-fal) méretét. Ez azonban nem csökkenthető akár milyen mértékben, ennek fizikai korlátai vannak. (Szemléletesen: az azonos pólusú mágnesek taszítják egymást, és minél közelebb helyezzük őket egymáshoz, annál nagyobb erőt kell kifejteni.)

A korlátok szabta keretek között azt tehetjük, hogy a mágnesezett tartományok méretét csökkentjük. (Gyengébb mágneseket alkalmazunk.) Ekkor viszont számolnunk kell azzal, hogy a lemágnesező erők átfordítják a mágneses tartomány irányát és ezzel adatvesztést okoznak. Az egyik lehetséges megoldást a nagy koercitív erejű mágnesezhető anyagok alkalmazása jelenti. (Ezek viszont nehezen mágnesezhetők. Az íráshoz, a tartomány átfordításához nagyobb íróáramra van szükség.) A másik megoldás az, hogy a mágnesezhető réteg vastagságát csökkentjük. (Itt azt az észrevételt használjuk fel, hogy az átmeneti tartományokból származnak a lemágnesező erők. Így a réteg csökkentésével ezek is gyengülnek.) Ezzel sajnos az is együtt jár, hogy a mágnesezett tartomány térerőssége is gyengül. Segíteni csak erős, nagy remanenciájú mágneses anyagok felhasználásával lehet. (7.20. ábra)

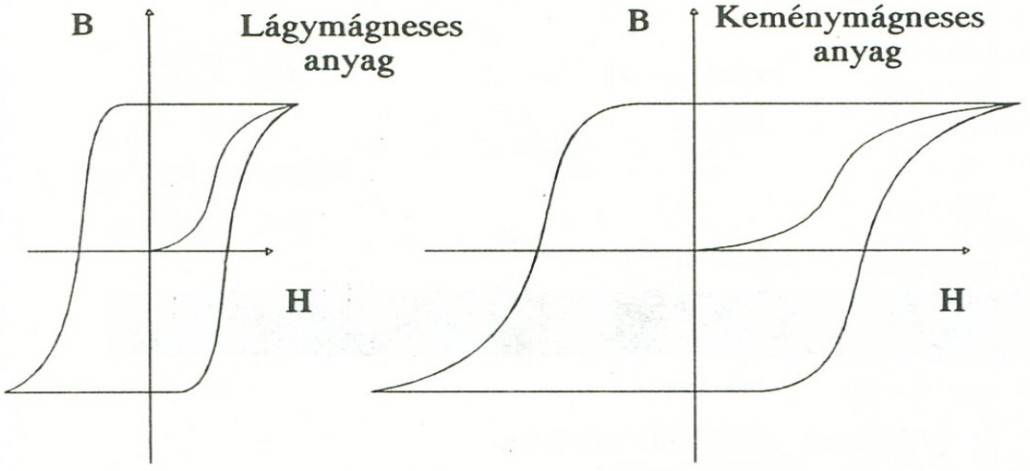


7.20. ábra

A lemágnesező erők az átmeneti tartományokból származnak.

Az előző két igénynek megfelelő anyagot találni nem könnyű feladat, de azért lehetséges. A korábban használatos vas-oxid alapanyagú hordozók tulajdonságai javulnak, ha a kristályok kobalt bevonatot kapnak, de újabban a króm-dioxidalapanyagúhordozókat

részesítik előnyben. Ez utóbbi anyagok alkalmazása jelentősen meggyorsul, hiszen sokkal egyenletesebben teríthetők el a hordozó műanyag felszínen, ugyanis kristályaik sokkal



7.21. ábra

A kemény és lágymágneses anyagok hiszterézis görbéje. A mágneses hordozóréteg anyagválasztásában játszik jelentős szerepet.

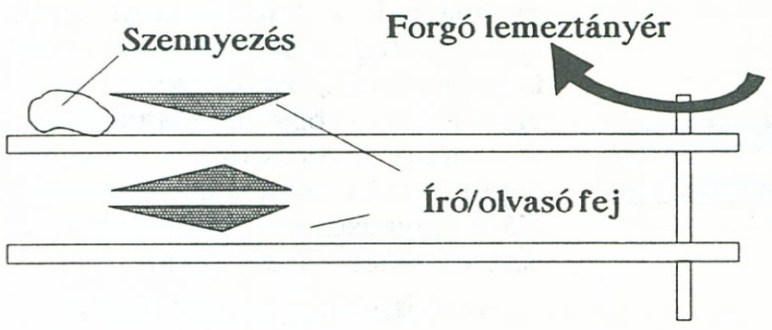
szabályosabbak, mint a tűszerű vas-oxid kristályok.

Az eszközök tökéletesítési tendenciái.

Az adatok visszaolvasási pontossága megnövelhető, még gyenge térerősség esetén is, (vékony adathordozó réteg) ha az író/olvasó fejet elég közel visszük a hordozóhoz. Ez az észrevétel azt sugalja, hogy tegyük a fejet közvetlenül az adathordozóra és ezzel máris megoldottuk a problémát. (Hajlékony lemezek és szalagos egységek megoldása.) Valóban javul a visszaolvasás pontossága nagyobb adatsűrűség esetén is, de ezért komoly árat kell fizetni, hiszen a visszaolvasás sebességét csökkenteni kell.

Az alkalmazott fordulatszám 300 — 360 fordulat/perc között lehet, ugyanis a súrlódás miatt számolni kell a hordozó vagy a fej kopásával. (Lágy adathordozó esetén az, míg kemény adathordozó, például króm-dioxid, alkalmazása esetén a fej kopik.) A hordozó kopását csökkenthetjük keményebb anyag alkalmazásával vagy megfelelő csúsztató réteg kialakításával. Sajnos nem kielégítő megoldás, hiszen a keményebb anyagok koptatják a fejet, ami azt jelenti, hogy a légrés kinyílik, ami megnöveli a mágnesezett tartományok méretét, azaz csökken az adatsűrűség.

Próbálkozhatunk olyan megoldással is, hogy a fejet nem közvetlenül, hanem légpárna alkalmazásával visszük megfelelő közel a hordozóhoz. (Itt nyilván nem kell számolnunk a fej kopásával.) Ezt az elképzelést műszaki okok miatt csak lemezek esetén tudjuk megvalósítani. A nagy sebességre felpörgetett lemez fölé helyezett író/olvasó fejet az áramló levegő megemeli (alkalmas fejkialakítás esetén). Az alacsony repülési magasság miatt a rendszer légterében csak igen kismértű szennyeződések lehetnek. (7.22. ábra) A magas fordulatszám miatt felgyorsult nagyobb méretű részecskék fejjel való ütközése a fej számára végzetes lehet. A szennyeződések mérete kisebb kell hogy legyen, mint 0,5 mikrométer. Ha meggondoljuk, hogy a cserélhető lemezkötegek mindig jelentős szennyezéssel rendelkeznek és ezeket a munka elkezdése előtt meg kell szüntetni, akkor kiderül, hogy



7.22. ábra

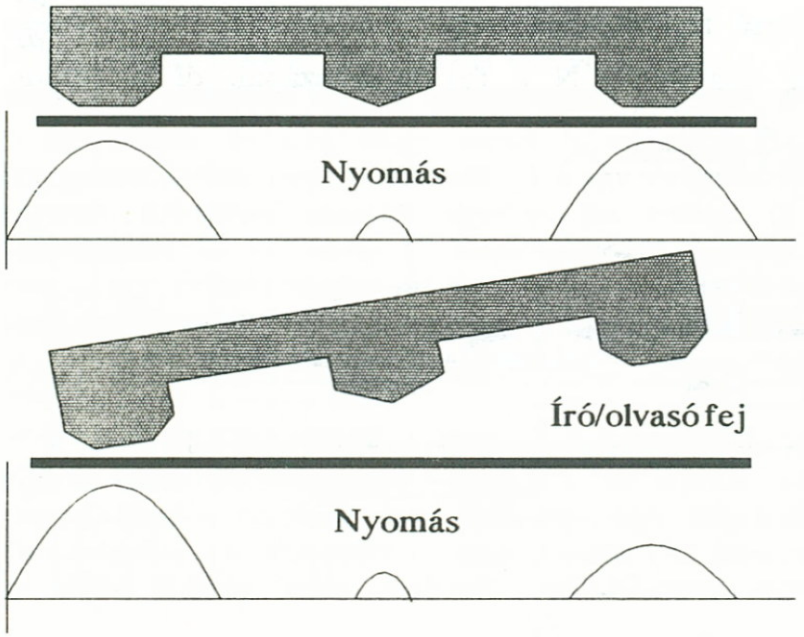
A nagy sebességgel forgó szennyezés és az író olvasó fej ütközése végzetes lehet.

ez a feladat meglehetősen bonyolult műszaki problémákat jelent.

Az előző két megoldás ötvözéséből egy újabb rendszerű tárolót fejlesztettek ki, amely manapság a mikroszámítógépek legelterjedtebb háttértárolója. Ez nem más, mint a **winchester** (1973). Tekintsük át azokat az elemeket, amelyeket e rendszer az elődeitől vett át.

- A csúszóréteggel bevont mágneslemez. (Ez biztosítja azt a lehetőséget, hogy a fej addig csúszhasson a lemez felületén, amig a disk el nem érte a megfelelő fordulatszámot, amely a légpárna kialkaításához szükséges. A lemez leállásakor is ez biztosítja a megfelelő landolást — ez a parkolópálya.)
- Légpárnán csúszó fej.
- A fej emelésének és süllyesztésének elhagyása (egyszerűbb mechanika)
- Nagy pontosságú fejmozgatás és sávontartás (szervorendszerek)
- Nagy fordulatszámú lemezforgató rendszer.

A fejek repülési magasságának állítása automatikus, ami a megfelelő geometriai kialakítás következménye. A repülési magasság jelentősen csökken (0,3 — 0,4 mikrométer). Ez teszi egyben azt is lehetővé, hogy vékonyréteg technológiával előállított fejeket alkalmazzanak, ami a légrés jelentős csökkenését eredményezi. Ez a technológia 50 %-al megnövelte a felírási jelsűrűséget a korábban alkalmazott ferritfejekhez képest (10000 bit/inch helyett 15000 bit/inch).



7.23. ábra

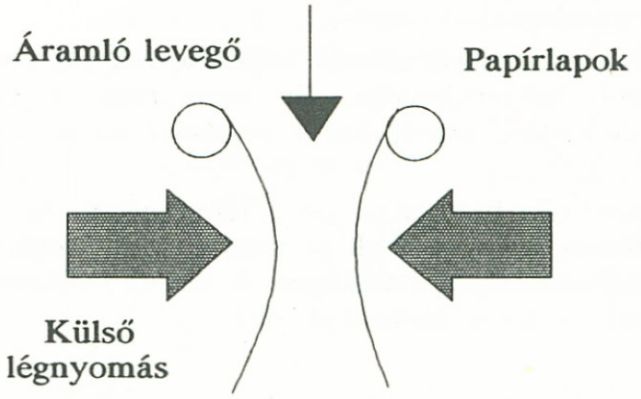
A winchester fej egyensúlyának megbomlását a fej alatt kialakuló nyomásviszony visszaállítja.

Természetesen nem csak horizontálisan (lineáris bitsűrűség) növekszik a jel felírási sűrűsége, hanem vertikálisan is lecsökken a sáv szélessége is.

A lemezek cserélhetőségének jellege megszűnik. (Nem a lemez, hanem a zárt box cserélhető.) Hiszen az alacsonyan, a lemezhez képest nagy sebességgel mozgó fej (a mozgás viszonylagos tulajdonsága miatt a forgó lemezhez rögzített vonyatkoztatási rendszerben a fej látszik mozogni) biztonságos repülése nagy tisztaságú levegőt igényel. Maximum 3 darab 0,5 mikronnál kisebb szennyező részecske tartózkodhat literenként a lemezt körülvevő levegőben. Ez a tér nem hermetikusan zárt, ezért speciális **mikroszűrők** biztosítják a boxba áramló levegő szennyezetségének kordában tartását. (A változó külső nyomás és hőmérséklet megváltoztatná hermetikusan zárt doboz esetén a lemez geometriai jellemzőit.)

A fejek mozgatását szervorendszer végzi. A sávontartást állandóan ellenőrzi és pillanatról pillanatra korrigálja a fejek helyzetét. (A lépés a sávok között 40 — 80 mikron.)

A cserélhetőség igénye azonban továbbra is nagy fontossággal bír. Ennek köszönhetően 1983-ban megalkották az úgynevezett **Bernoulli-boxot**. Ez a lemezegység a hajlékonylemezek cserélhetőségének és a winchesterek nagy írássűrűségének,nagy kapacitásának a házasításából jött létre. A hajlékony lemezt egy merev sík lap fölött forgatjuk nagy sebességgel. (rögzített távolságban). Az író/olvasó fej a fémlapból emelkedik



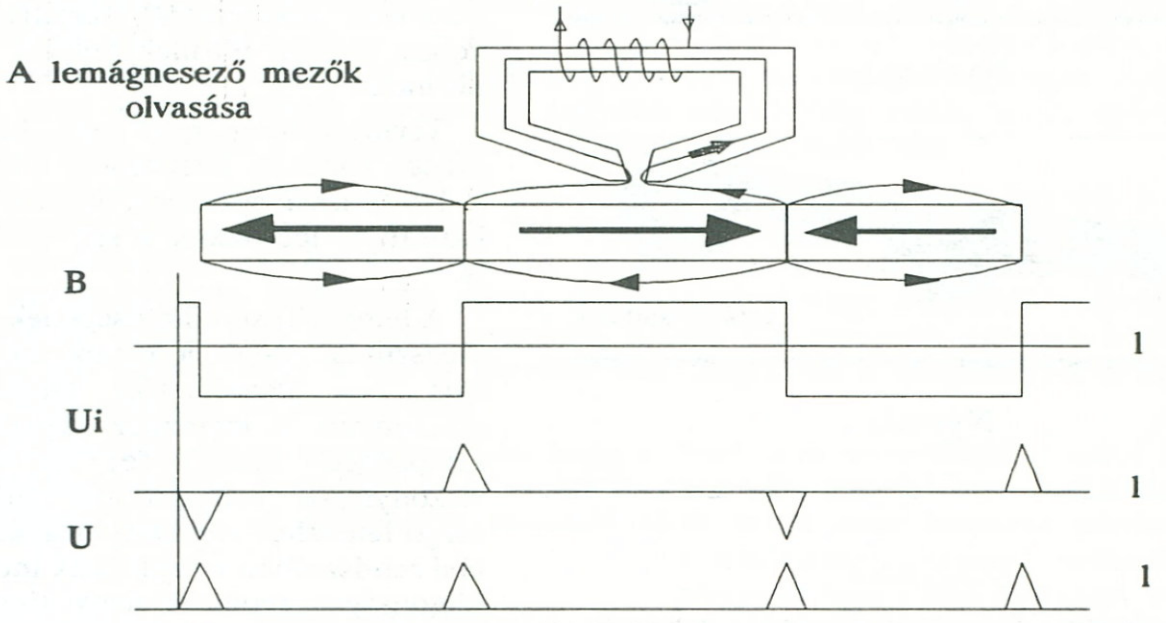
7.25. ábra

A papírlapok között áramló levegő nyomáscsökkenést hoz létre ezért a külső légynomás összenyomja a lapokat.

ki. (Bernoulli-lap.) A forgó hajlékonylemez keltette légáram a fejhez szívja azt, és ezzel csökkenti a lemez és a fej távolságát. (Ha a forgás csökken, akkor a szívóhatás is csökken, ezért a lemez és a fej eltávolodik egymástól.) (7.24. ábra)

Felmerül a kérdés, hogy ezekkel a műszaki megoldásokkal elértük-e a legnagyobb adatsűrűség megvalósítását? A válasz természetesen az, hogy nem. Ugyanis megfelelő kódolási technikák esetén lényegesen növelhető az adatbitek sűrűsége. A megoldások ismertetése előtt azonban tekintsük át az adat visszaolvasásának a folyamatát! (7.25. ábra)

Az író/olvasó fejen áthaladó áram vele arányos mágneses teret hoz létre a hordozón. Olvasáskor, mint már tudjuk, a lemágnesező erőket használjuk fel. (Ezek terének iránya ellentétes a mágnesezett tartományok terének irányával.) A tér változása a az író/olvasó fejben feszültséget indukál ($U_i = -N \frac{df}{dt}$ ahol U_i jelenti az indukált feszültség nagyságát, N a fej menetszáma, df a fluxus



7.24. ábra

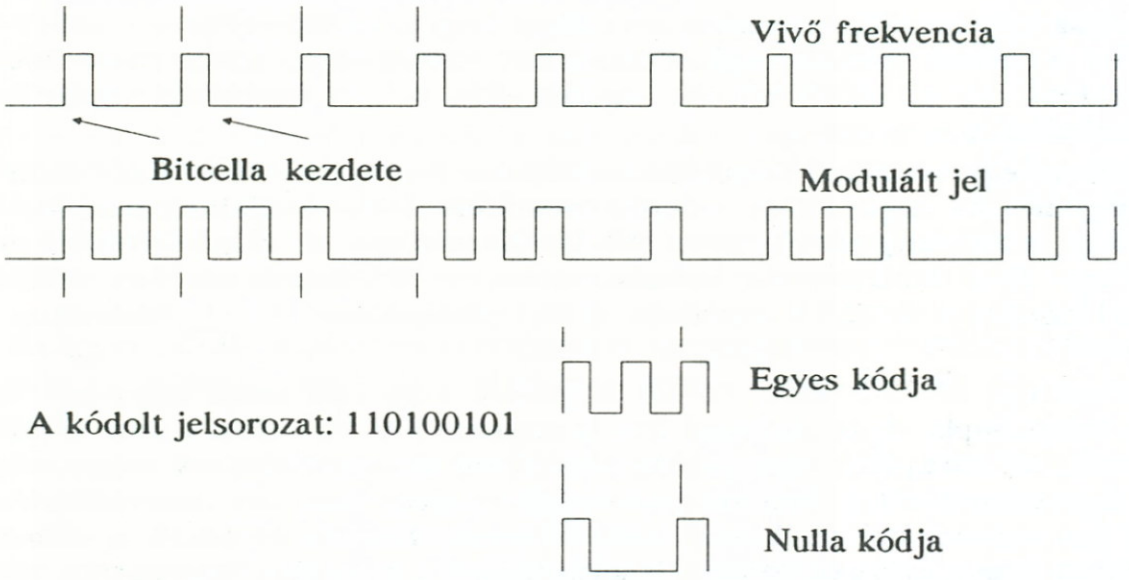
A lemágnesező mezők a fejben indukált feszültséget hoznak létre, (Ui) amelyet egyenirányítva kapjuk a kívánt bitsorozatot (U)

megváltozása, dt a megváltozáshoz szükséges időtartam.) Végül ezt a jelsorozatot egyenirányítjuk és így megkapjuk a feldolgozásra váró adatsorozatot.

Ez maga után vonja azt a kérdést, hogyan lesz ebből az impulzussorozatból bináris jelsorozat? Ez csupán kódolási konvenciók kérdése. A kódolást úgy kell elvégezni, hogy minél ritkábban kelljen megváltoztatni a tér irányát a hordozón.

A különböző eljárások ezt a feladatot oldják meg.Tekintsünk meg néhány, a gyakorlatban ma használatos példát az egyszerűtől a bonyolult eljárás felé haladva.

Vizsgáljuk meg először az úgynevezett FM **frekvencia-moduláció** s eljárást.Az eljárás lényege az, hogy minden bitcella kezdetet órajellel jelölünk meg. Ez a vivőfrekvencia, amelyet változtatunk (modulálunk) a felírandó információ függvényében. A moduláció



7.26. ábra

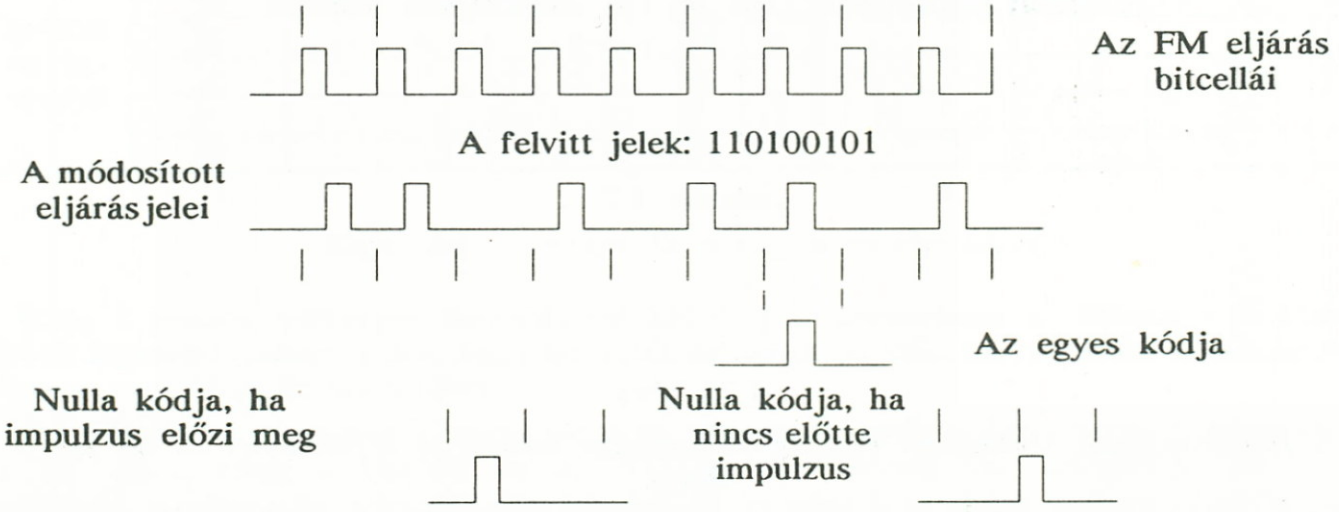
A frekvencia-modulációs kódolás.

most annyit jelent, hogy megnöveljük az impulzusok számát, ha egyest kódolunk. Pontosabban bináris 1 kódolása esetén a bitcella közepén is létrehozunk egy impulzust,míg bináris 0 kódolása esetén változatlanul hagyjuk a cella tartalmát. (7.26. ábra)

Számoljuk ki azt, hogy ennél a megoldásnál milyen a felírt adatok sűrűsége! A legegyszerűbben úgy járhatunk el hogy megnézzük, hogy 10 jel kódolásához, amelyben egyenlő számban vannak egyesek és nullák (5 — 5) hány darab impulzusra van szükségünk. Ez összesen 15 impulzust jelent, mert az 1 kódolásának kétóraimpulzus felel meg, (egy cellakezdeten és egy cellaközépen lévő impulzus) a nulla kódolásához pedig csak egyre vanszükség. Ezért az egységnyi adathoz, egy bithez, tartozó impulzusok száma

$$1{,}5.\ \left(\frac{5 * 2\,impulzus\ +\ 5 * 1\,impulzus}{10\,bit} = 1{,}5\frac{impulzus}{bit}\right)$$

A második eljárásunk a frekvencia-modulációs eljárás javított változata. Ez az **MFM** eljárás (modified-frequency-method). Az eljárás megértéséhez osszuk fel a bitcellákat két részre.Minden részben vagy szünetet,vagy impulzust írhatunk. Ennek megfelelően az 1 ábrázolásához a szünetből és impulzusból álló jelsorozatot rendeljük. (Ez pontosan megfelel az előző eljárás során említett „cellaközépen egyimpulzust kódolunk" megoldásnak.) A



7.27. ábra

A módosított frekvencia-modulációs eljárás.

nulla kódolása egy kissé összetettebb, attól függ, hogy a kódolás előtt mit kellet felírnunk. Ugyanis, ha az előző bitcella végénél impulzust rögzítettünk, akkor két szünet fogja a nullát jelenteni, ha az előző bitcella végénél szünet volt, akkor egy impulzus és egy szünet szimbolizálja a nullát.

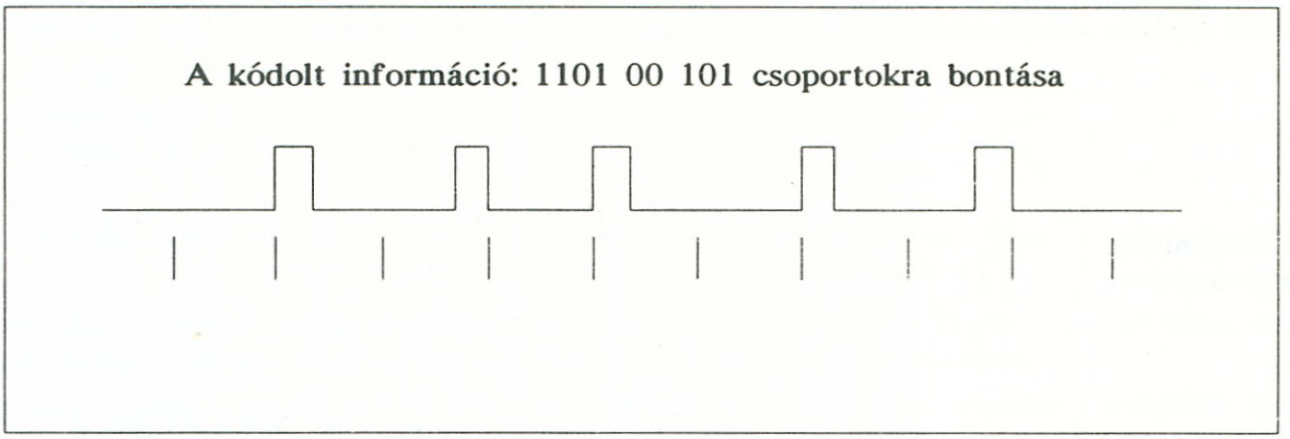
Ezen eljárás egy bitre eső impulzusainak száma lényegesen kedvezőbb, mint az előző esetben, ugyanis ha egyenlőnek vesszük a kétféle nulla kódjainak számát, (ez nagy bitmennyiség esetén megtehető) akkor 10 bit tárolásához, az előző feltételek mellett — 5 bináris egy és 5 bináris nulla kodolása esetén — 7,5 impulzusra van szükségünk. (Tehát $\frac{5 * 1\,impulzus + 2{,}5 * 1\,impulzus + 2{,}5 * 0\,impulzus}{10\,bit} = 0{,}75\ \frac{impulzus}{bit}$) Ezt a megoldást mutatja az 7.27. ábra. Ez azt is jelenti, hogy fele annyi impulzust kell felírni a lemezre, tehát dupla olyan sűrűségű írás is megengedett. Ez valóban szép teljesítmény,de tudjuk e még túlszárnyalni, még jobban tömöríteni az adatainkat. A válasz megint csak igen. Az úgynevezett **RLL** kódólás (run-length-limited), bár jóval bonyolultabb, mint az előző, de azt az eredményt hozza, amit vártunk. Körülbelül 50 %-al növeli meg a tárolható adatok mennyiségét. Mielőtt belekezdenénk a megoldás tárgyalásába vezessük be a **futási hossz** fogalmát. Ez a rögzített jelekben egymást követő legrövidebb és leghosszabb szünetek számát jelenti. Nézzük meg az elöbb bemutatott eseteknél mekkorák ezek az értékek! Az FM eljárásnál a minimális futáshossz 0, hiszen az egyes tárolásához a bitcellában két impulzusra volt szükségünk, és ez azt jelenti egyúttal, hogy a cellában szünet nem lehet. A maximális futáshossz pedig 1, hiszen a nullát ábrázoló cella elején ott van az óraimpulzus, ami annyit jelent, hogy impulzus és szünet van a cellán belül. Ezért azt mondhatjuk hogy az FM eljárás 0,1 RLL-nek felel meg.

A képezhető bitcsoportok	A megfelelő kódolt értékek
00	1000
01	0100
100	001000
101	100100
111	000100
1100	00001000
1101	00100100

7.2. táblázat

Az RLL kódolás hozzárendelési utasítása.

Az MFM eljárásnál a legrövidebb futáshosszhoz egy szünet tartozik, ami két egymást követő 1 tárolásának felel meg. A legnagyobb futási hossznak 3 szünet felel meg.Ez az eset akkor áll elő, ha a kódolandó bitsorozat . . 101. . , mert a nullák tárolásához két szünetre van szükségünk, és az utolsó egyest szünet és impulzus állítja elő. Ezért az MFM futási hosszban kifejezve 1,3 RLL.



7.28. ábra

A futáshosszban korlátozott eljárás csoportokra bontja az információt és így kódolja.

Most már megfogalmazhatjuk, mit is rejt tulajdonképpen az eddig csak emlegetett futáshosszban korlátozott eljárás! Ez nem más, mint a 2,7 RLL értéknek megfelelő kódolási eljárás. Vagyis ez azt jelenti, hogy a minimális futáshossz 2, azaz mindig legalább két

szünet választja el egymástól az impulzusokat. A maximális futáshossz pedig 7, ami a legkedvezőbb esetben két szomszédos impulzus távolsága szünetekben kifejezve. Ezek után tekintsük meg a kódtáblázatot! (7.2. táblázat)

A kódolás során 2 - től 4 hosszúságú csoportokat képzünk a kódolási szabályoknak megfelelően, majd kiválasztjuk a nekik megfelelő kódot. (7.28. ábra) A kódok felépítéséből azonnal látszik,hogy a legkedvezőtlenebb esetben is teljesül, hogy legalább két szünet van az impulzusok között. A legkedvezőbb, azaz a legnagyobb futáshossz pedig az 100 és 1100 bitekből álló és egymást követő csoport kódolása során következik be. A mintegy 50 %-os nyereséget az is mutatja,hogy amíg az MFM 17 szektort ír fel egy sávra, addig az RLL 26-ot. Az eljárás továbbfejleszése az úgynevezett ARLL (Advanced-Run-Length-Limited) eljárás, ami az MFM-hez képest mintegy 90 %-os növekedést eredményezett.

A kódolás egyre bonyolultabb jellege és az egyre nagyobb jelek közti szünetek azonnal magukkal hozták azt az igényt,hogy megfelelő vezérlőket fejlesszenek ki a lemezek kezelésére. (Gondoljuk meg az impulzusok közötti szünetek időtartamára is kell biztosítani az órajeleket a szinkronizáció miatt. Órajelek közvetlenül csak az FM eljárás során kerülnek a lemezre. Ezt a problémát csak úgy tudjuk megoldani, ha nagyon stabil fordulatszámmal rendelkező motorokat használunk a lemezek meghajtásához.)

Az előbb elmondottak miatt, mint láttuk a CPU és a mágneses periféria közvetlenül nem tud kommunikálni, ezért szükséges olyan átalakító létrehozása (kontroller — vezérlő) amely megoldja ezt a problémát. A mikroszámítógépek világában 4 féle winchester szabályozó terjedt el. Ezek legrégebbi példánya a SEAGATE cég által kifejlesztett ST506/412 típusú vezérlő. A lemezt két szalagkábel köti össze a vezérlővel, az egyiken csak az adatforgalom, a másikon csak a vezérlő utasítás küldése zajlik. A vezérlő kábel segítségével lehet kiválasztani a megfelelő meghajtót (1 — 2), a lemezekhez tartozó fejeket. ennek segítségével lehet a fejek léptetését vezérelni, az adatirányokat kiválasztani stb. Az adatvezetékeken keresztül soros formában történik az adatok írása olvasása MFM vagy RLL módban. A vezetékeken átáramló információ mennyisége nagyon nagy fontossággal bír, hiszen ez is meghatározza a számítógépes rendszer teljesítőképességét. A szabvány előírása szerint a vezérlő MFM módban 5 Mbit/s, azaz 510 Kbajt/s sebességgel képes az adatokat átvinni. Az RLL eljárás szerint üzemeltetve ez kb. 7,5 Mbit/s sebességet jelent.

		Fizikai szektor																
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Interlave tényező	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	2	0	2	4	6	8	10	12	14	16	1	3	5	7	9	11	13	15
	3	0	3	6	9	12	15	1	4	7	10	13	16	2	5	8	11	14
	4	0	4	8	12	16	1	5	9	13	2	6	10	14	3	7	11	15
	5	0	5	10	15	1	6	11	16	2	7	12	3	8	13	4	9	14

7.3. táblázat

Kapcsolat a logikai és fizikai szektorok között.

Ezen a ponton szükséges megemlíteni két dolgot, nevezetesen az interlave és a cache tároló fogalmát. Mind a két fogalom által deklarált eljárás az adatforgalom gyorsítását célozza meg. Vegyük sorra őket!

Az **interlave** fogalma. Gondoljuk meg, a nagy sebességre kényszerített mágneslemez az idő alatt, amíg a vezérlő és a CPU az adatcserével van elfoglalva, (az adatok megfelelő tárelemekbe töltése) elfordul. Ha az új adatot az előbb kiolvasotthoz képest a rákövetkező szektorból kellene kiolvasnia, akkor ez azt jelentené, hogy meg kell várnia, hogy a lemez egy teljes fordulatot tegyen (ugyanis a fej már túlhaladta ezt a pozíciót) és ekkor lenne csak képes az adat kiolvasására, ami jelentős idővesztességet jelentene.

Ezért a logikai és fizikai szektorszámok nem azonosak. A logikait úgy választják meg,hogy a következő szektort akkor is olvasni lehessen, amikor a merevlemez már elfordult. Tehát ezek szerint az interlave nem más, mint a fizikai és logikai szektorok közötti eltolás értéke. Nyilvánvaló, hogy ha kicsi az interlave érték, akkor az átviteli sebesség nagy a számítógép és a lemezegység között. Ha ez nagy, akkor fordítva. Egy alacsony teljesítményű XT esetén körülbelül 5, még egy gyors vezérlővel ellátott gép esetén 1 az interlave értéke, azaz nincs eltolás a fizikai és logikai szektorok között.

A jel jelentése	A vezeték száma	A jel jelentése	A vezeték száma
A 3-as fej kiválasztása	2	A 3-as fej kiválasztása	2
A 2-es fej kiválasztása	4	A 2-es fej kiválasztása	4
Íráskapuzás	6	Íráskapuzás	6
Keresés kész	8	Konfiguráció/statusadat	8
A 0. sáv	10	Az átvitel visszaigazolása	10
Íráshiba	12	Figyelem	12
A 0-s fej kiválasztása	14	A 0-s fej kiválasztása	14
Fenntartva	16	Szektor	16
Az 1-es fej kiválasztása	18	Az 1-es fej kiválasztása	18
Index	20	Index	20
Kész	22	Kész	22
Lépés	24	Átvitelkérés	24
Az 1-es meghajtó kiválasztása	26	Az 1-es készülék kiválasztása	26
A 2-es meghajtó kiválasztása	28	A 2-es készülék kiválasztása	28
Fenntartva	30	A 3-as készülék kiválasztása	30
Fenntartva	32	Olvasáskapuzás	32
Irányválasztás	34	Utasításadat.	34

7.5. táblázat

Az ST506/412 interface vezérlő kábeleinek kiosztása. A páratlan csatlakozószámot nem tüntettük fel, mert ezek mindegyike földvezeték. (Árnyékolás miatt.)

7.4. táblázat

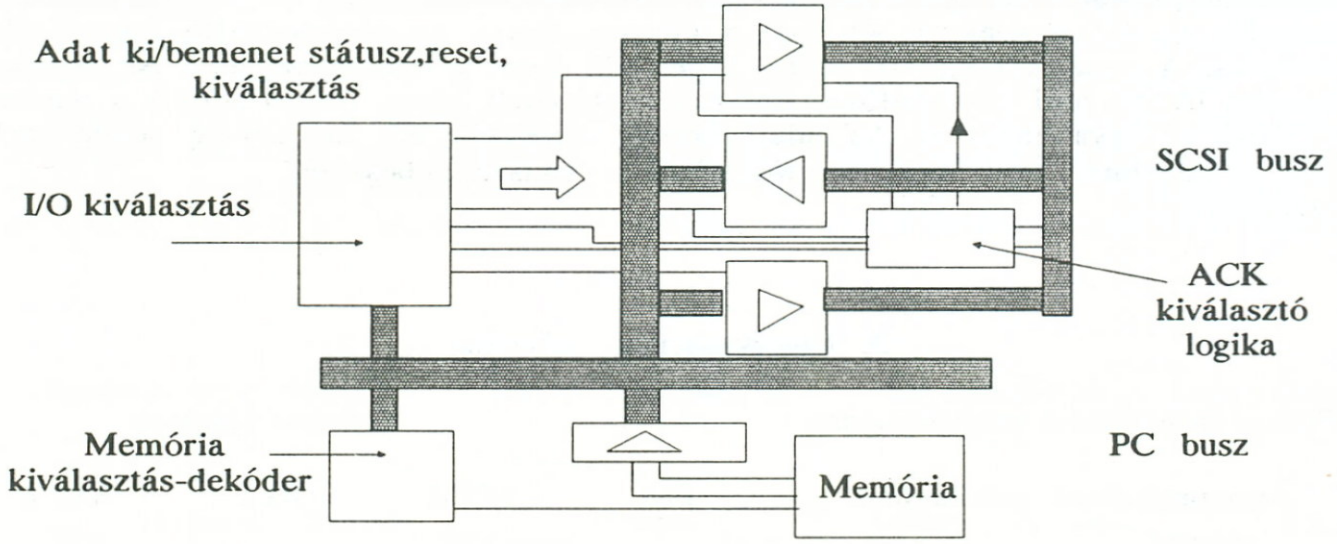
Az ESDI interface vezérlő kábeleinek kiosztása. A páratlan csatlakozószámot nem tüntettük fel, mert ezek mindegyike földvezeték. (Árnyékolás miatt.)

A **cache** tulajdonképpen egy gyorsító tár. Mint ismeretes, minél nagyobb a tárolók kapacitása, annál nagyobb a ciklusideje, annál nehezebb feladat megkeresni a megfelelő elem helyét.(A nagyobb kapuszám miatt lassúbb lesz az elektronika.) Ezért ha egy viszonylag kis kapacitású, de gyors tárolót egy lassú, de nagy kapacitású háttértárolóval kötünk össze, akkor ezek megfelelő együttműködése során egy, a felhasználó számára átlátszó, nagy kapacitású és gyors tárolót jelentenek. A cache tároló effektív sebessége attól függ, hogy milyen gyakorisággal elégíti ki a cache tároló a tárigényeket az operatív tár helyett. Ez általában két tényezőtől függ: egyrészt a tároló kapacitásától, másrészt a tártartalom szervezésétől. Általában a szervezését úgy oldják meg, hogy a jelenleg kért adatok az előzőleg kért adatok fizikai közelében legyenek.

Ennyi kitérő után folytassuk a meghajtók tárgyalását! A második vezérlő, amely elterjedőben van a MAXTOR cég által kezdeményezett **ESDI interface**. (Enhanced Small Device Interface) Kábelezése igen hasonló az ST506/412 -hez. Ha megtekintjük a két rendszer vezérlő kábeleinek felosztását, akkor kiderül, hogy a hasonlóság nem véletlen, az ESDI interface a növekvő sebességigény követelményéből fakadt és továbbfejlesztése a SEAGET cég meghajtójának.

Mivel a lemezen helyezték el a vezérlő és adatkódoló áramkörök zömét, maga a meghajtó jelentősen egyszerűsödött. A rendszernek külön utasításkészlete van (Utasításadat 34. lábon.) az író/olvasó fejek vezérlésére. A rendszer kialakításának köszönhetően az adatátvitel sebessége elérheti a 2 Mbájt/s értéket is.

A bemutatásra kerülő 3. vezérlő az előzőekhez képest lényegesen eltérő felépítéssel és tulajdonságokkal rendelkezik. Ez a **SCSI** (Small Computer System Interface). Egyetlen 50 szálas vezetékszalag köti össze a lemezzel. A kialakított interface nemcsak lemezek,hanem

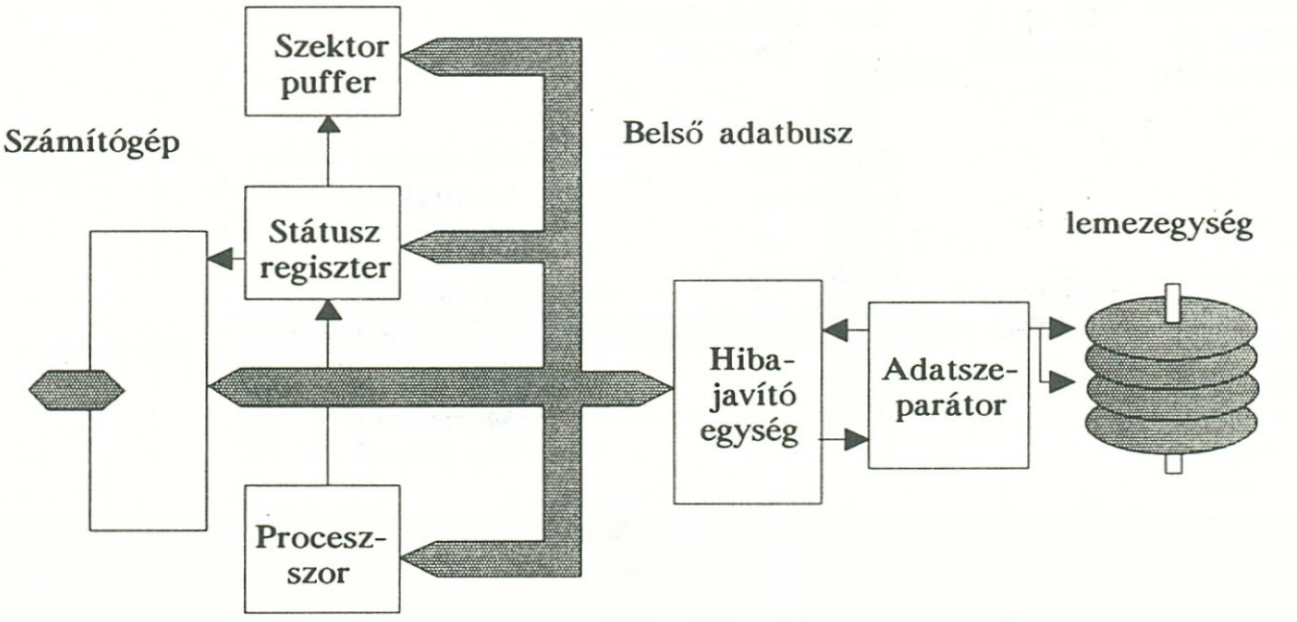


7.29. ábra

Az SCSI interface blokkdiagramja.

más perifériás egységek kezelését is ellátja. A rendszerhez akár 8 különféle berendezés is csatlakoztatható. Minden egység külön azonosítót kap 0 -tól 7-ig. Ez a szám kevésnek tűnhet, bár egyáltalán nem az, hiszen a vezérlőt úgy alakították ki, hogy mindegyik egységhez újabb 8 periféria csatlakozhat, ami már a második szinten 256 logikai egységet jelent.

Az SCSI szabvány szóhasználatával élve minden egység inicializátorként vagy targetként működik. (Inicializátornak — Host — azt az egységet nevezzük, amely küldi illetve készíti

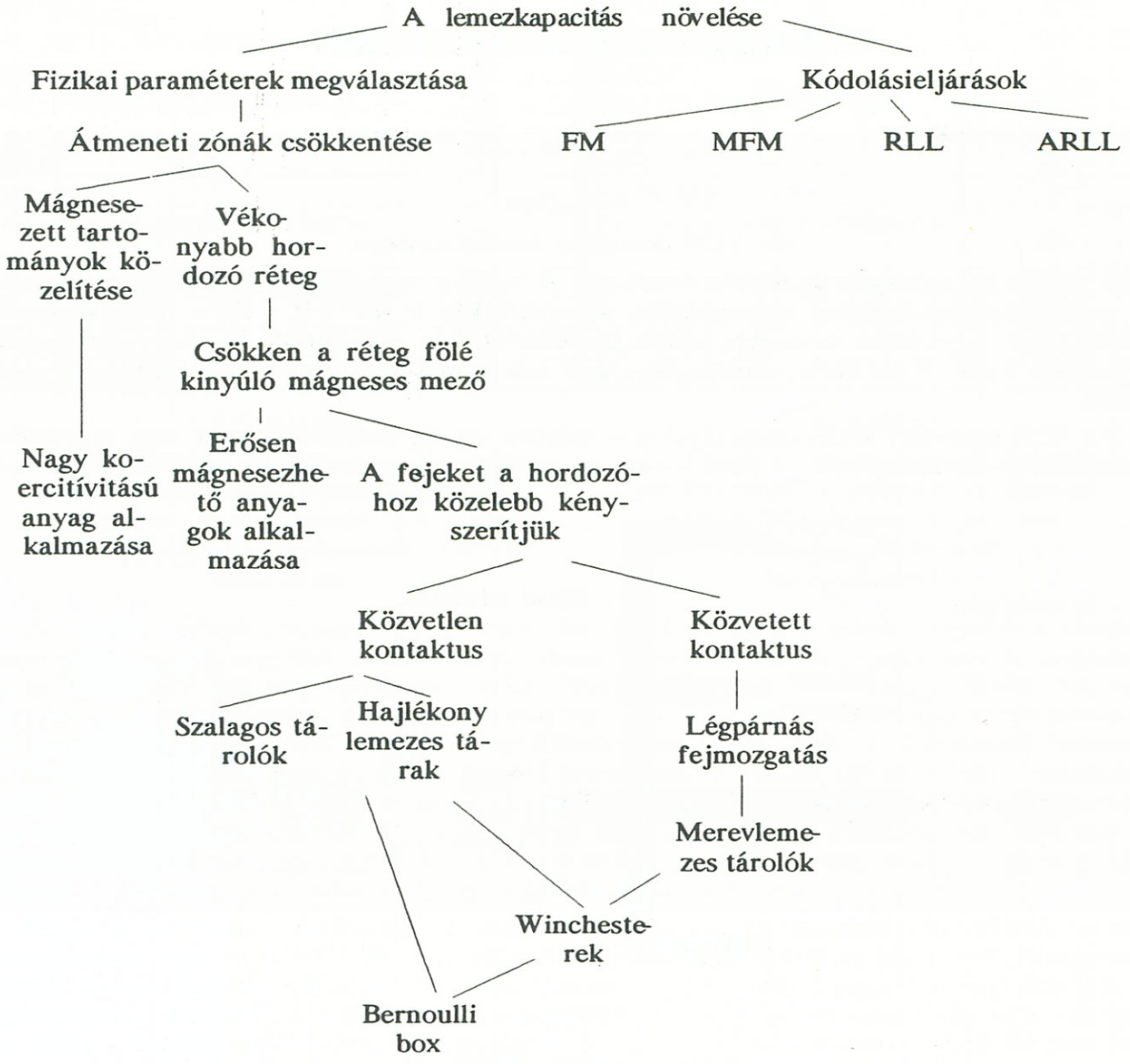


7.30. ábra

Az AT busz felépítése

a parancsokat, target-nak pedig azt, amely fogadja őket.)A rendszer tömbvázlatát az 7.29. ábrán láthatjuk.

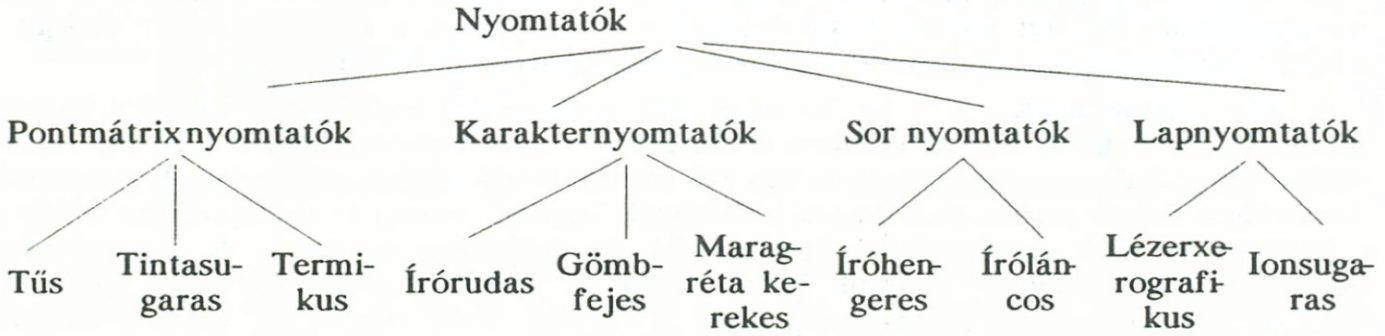
Az utolsó lemezvezérlő szabvány az SCSI-nek a leegyszerűsített, olcsóbb változata, az úgynevezett **AT-busz** vezérlő. (Ezt az egységet több kölönböző néven is ismerhetjük: IDE Integrated Device Equipment, vagy CDI Cluster Disk Interface.) A vezérlőt a lemezegységre építették hasonlóan az előző két esethez. (Ez által kevesebb zaj és nagyobb sebesség érhető el.) Itt foglal helyet maga a kontroller, és a fejemozgatáshoz, a motormeghajtáshoz szükséges fokozatok is itt kerültek elhelyezésre. Az interface csak (ez illeszkedik az AT csatlakozó sínjébe) a címzést illetve az adatmeghajtó áramköröket tartalmazza. A rendszert egy 40 szálas kábel köti össze a lemezegységgel. Az adatbusz szélessége 16 bit, ami meglehetősen kedvező adatátvitelt jelent. (Innen adódik a rendszer elnevezése is, ugyanennyi az AT adatbuszának szélessége is.) Sok egység önálló cache tárolót is tartalmaz, ami jelentősen megnöveli a rendszer sebességét.
Összefoglalva:



7.4. A nyomtató egységek

A nyomtató olyan periféria, amelynek segítségével a számítógép által szolgáltatott bináris információkat, megfelelő átkódolás után maradandó, nyomtatott formában papíron megkapjuk. A számítógépekhez csatlakoztatható nyomtatók meglehetősen tarka képet mutatnak, mind felépítésükben, mind pedig nyomtatási módjukban. Az 7.5. táblázat áttekintést ad a nyomtatók családfájáról, az egyszerre kinyomtatható karakterek száma alapján.

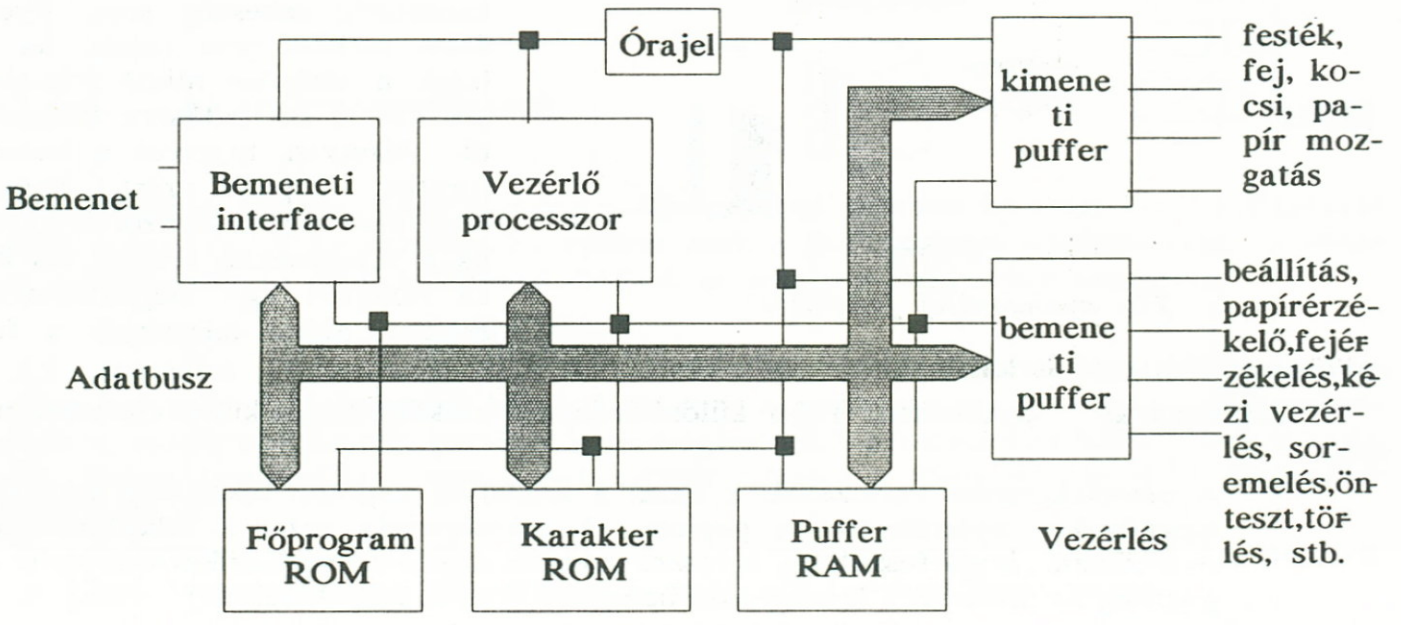
Mielőtt rátérnénk a táblázatban szereplő nyomtatók rövid ismertetésére tekintsük át a nyomtatók általános, elvi felépítését. (7.31. ábra) Mint ahogyan az ábrából kitűnik a nyomtatók intelligens perifériák, hiszen saját vezérlő processzorral rendelkeznek. Ez a processzor vezérli az adatforgalmat, valamint az egyes motorok működési módját. A



7.6. táblázat

A nyomtatók csoportosítása az egyszerre nyomtatott karakterek száma szerint.

rendszer működéséhez szükséges programokat a ROM tartalmazza. A nyomtató karakterkészletére vonatkozó információkat a karakter ROM-ban tároljuk. A belépő adatokat



7.31. ábra

A nyomtató blokkvázlata.

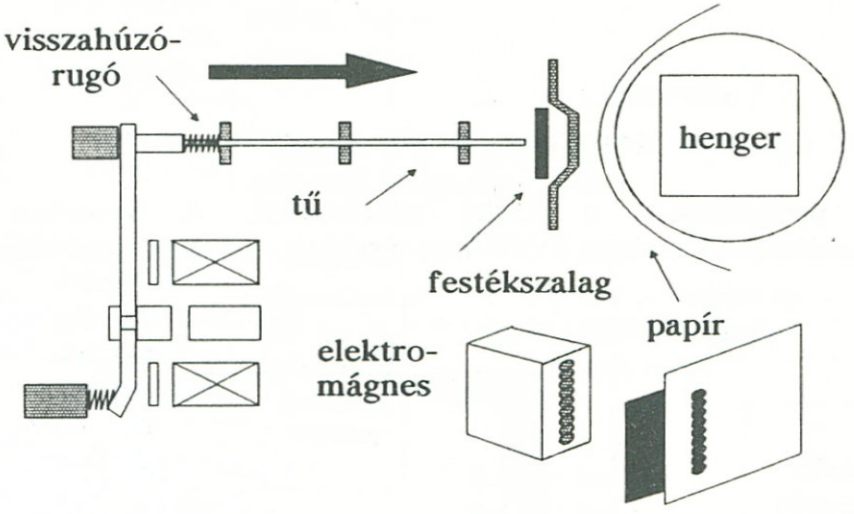
felhasználásig a RAM tárolja. Ez a nyomtató működésének gyorsításához szükséges, ugyanis a számítógép felöl érkező adatok jóval nagyobb sebességgel érkeznek, mint ahogyan a mechanika ki tudja őket nyomtatni. Megoldás lehetne a karakterenként történő leválás a központi egységről, de ez nem jó, hiszen ekkor a processzor munkaidejét vesszük nagyon igénybe. Ezért amennyi adatot lehetséges és értelmes egyszerre küldünk át ebbe a tárba.

A kimeneti pufferek tárolják a különböző motorok és áramkörök vezérléséhez közvetlenül szükséges információkat. A bemeneti pufferek pedig a nyomtató számára érkező információkat tárolják. Ezek tartalmától függ a főprogram végrehajtása.

7.4.1. A pontmátrix nyomtatók

A karakterek képét pontokból rakjuk össze. Ez a megoldás lehetővé teszi, hogy tetszőleges betűtípust és grafikus ábrát kinyomtassunk. A tetszőleges talán túlzásnak tűnhet, hiszen a beépített karakterkészlet ugyancsak szűk terjedelmű, de ezen szoftver úton segíthetünk. A legtöbb nyomtató rendelkezik olyan tárterülettel, ahová letölthető a kívánt karakterkészlet. Ez azt jelenti, hogy átadjuk a nyomtatónak a kívánt karakter alakját és méretét tartalmazó információt (firmware), azaz a karakter pontmátrixát.

A **tűs nyomtatók.** A 9 és 24 tűből álló nyomtatófej segítségével áll elő a karakter mintázata. A 9 tűs nyomtatófejekben a tűk egymás alatt helyezkednek el. (24 tűs változat esetén 12 - 12 tű egymás mellett de egy fél ponttal lefelé eltolva található.) A nyomtatófej vízszintesen balról-jobbra (oda-vissza beálítástól függően) mozog és oszloponként állítja elő a jeleket. A fejek mechanikája igen precíz, de felépítése egyszerű. A tűk számának megfelelő elektromágnest találunk benne. Ezek az áramimpulzusok hatására lövik ki a tűket.Az áramimpulzus megszünése után a fejben lévő rugók az eredeti helyükre visszahúzzák a tűket. A tűk átmérője igen kicsiny, alig haladja meg a fél millimétert. Nagyon rugalmasnak, edzetnek kell lenniük,ha meggondoljuk, hogy milyen mechanikai igénybevételnek vannak kitéve. A nagyteljesítményű nyomtatókban nem ritka a 800 karakter/s sebesség sem. Ilyen ütem mellett nem csoda, ha a fejek a súrlódás miatt felmelegednek. A hő hatására kitáguló tű könnyen beszorul a vezető furatba, éppen ezért, ennek elkerülése végett, szenzorok figyelik a hőmérséklet értékét, és ha az túllépett egy meghatározott értéket, akkor lelassítják a fej működését illetve ki is kapcsolhatják azt.



7.32. ábra

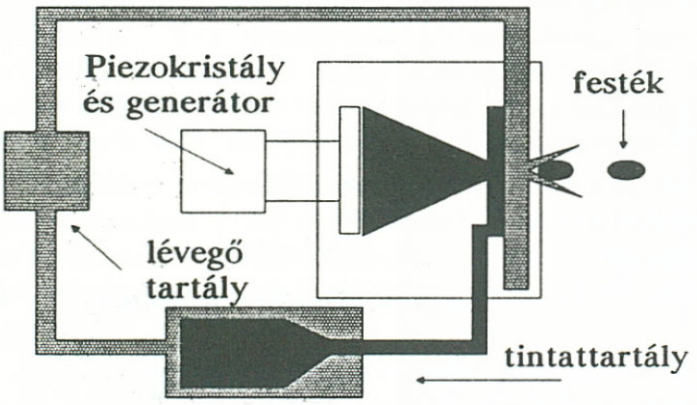
Tűs nyomtatófej felépítése.

A **tintasugaras nyomtató.** Több különböző elven működű típust különböztethetünk meg.

- A pizoelektromos változatnál a tintát a kristályra kapcsolt feszültség hatására megnövekvő nyomás lövi a papírra. (A piezokristály azzal a tulajdonsággal rendelkezik, hogy feszültség hatására megváltoztatja a mechanikai méreteit. A jelenség fordítottját a lemezjátszó hangszedőiben is felhasználják.)
- Elektromos megoldásnál a tintát felhevítik és a keletkező túlnyomás lövi a festékcseppet a papírra.Mind a két említett megoldásnál koncentrikusan áramló levegő („füstkarika") irányítja a tintacsepp útját.

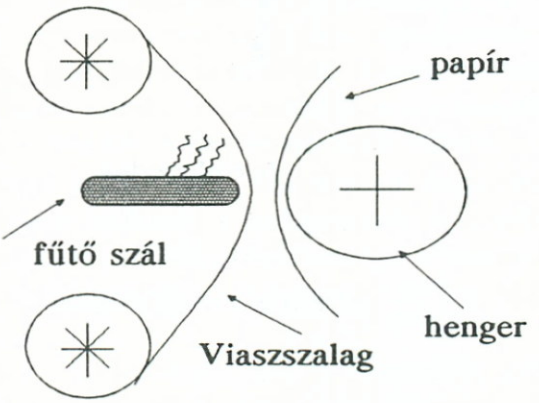
» Az elektrosztatikus változatnál a kilövés előtt sztatikusan feltöltött tintacseppet mágneses mező irányítja a papír megfelelő helyére.

Természetesen ezen megoldás esetén is egy-egy jel kialakításához több festékcsepp szükséges. A nyomtatófejek nem érintkeznek közvetlenül a papírral, ezért a működésük igen csendes. A gépek használatához megfelelő nedvszívó képességű papír használata ajánlott, hiszen ellenkező esetben a tintacsepp szétfolyása következik be.



7.33. ábra

Tintasugaras nyomtatófej.



7.34. ábra

Hőnyomtató működési elve.

A **hőnyomtató**kban a papír szélességének megfelel, színezett viaszréteg felületéről apró fűtőszálak távolítják el, illetve olvasztják a papírra a festégréteget. Ezen eljárás során speciális papírt kell alkalmaznunk. Az eljárás előnye, hogy nagyon szép színes nyomatok készíthetők vele, de a lapok könnyen sérülnek. A nyomtatás lassú, hiszen a színek előállítása négy menetű nyomtatást igényel. (Egyet-egyet az alapszínek, egyet pedig a fekete kikeveréséhez.)

Az újabb megoldások során már műanyag festéket alkalmaznak, amelyet a nyomtató felmelegít és a tintasugaras nyomtatókhoz hasonlóan porlasztja a papírra. A műanyag színezék nem szívódik be a papírba, hanem annak felületén szilárdul meg, így kellemes domborhatás érhető el.

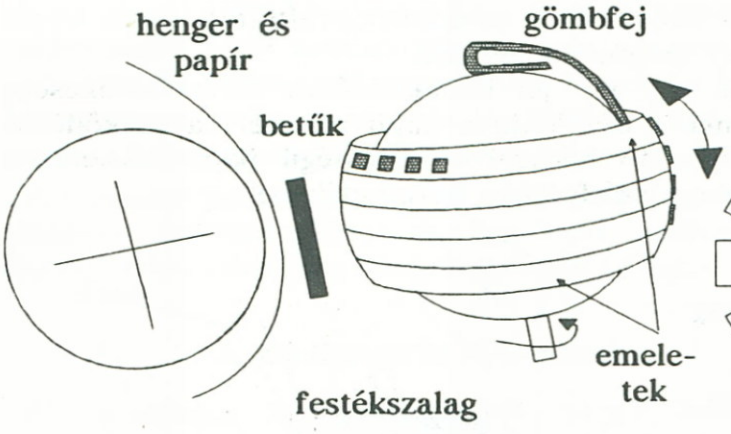
7.4.2. Akarakternyomtatók

Működésének lényege, hogy egyetlen kalapácsütésre egyetlen karakter kerül a megfelelő időben a papírra. A karakterkészlet váltása csak a fej cseréjével oldható meg. Grafikus ábra nem készíthető velük.Működési módjuk az írógép működéséhez hasonlatos.

Az **írórudas nyomtató**knál a karakterkészlet negatív formáit egy rúdon fésűfog-szerűen helyezik el. A kalapács hátulról üti meg a lemezt és így nyomja a festékszalagot a papírra, amellyen a leütött karakter képe megjelenik.A nyomtatórúd hosszirányú mozgást végez. Ennek megfelelően minden karakterkép elhalad a kiíró pozíció előtt. (Sebessége 200 sor/perc.)

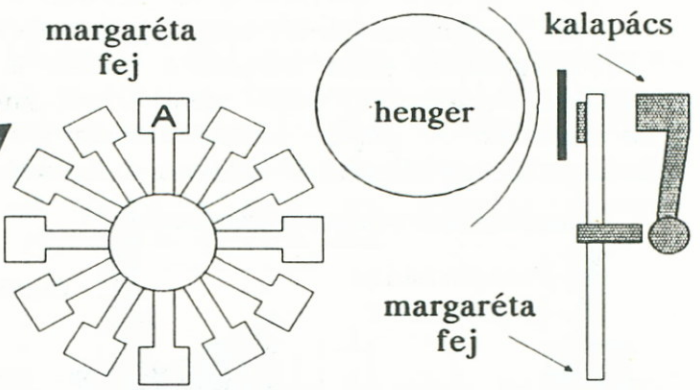
A **gömbfejes** megoldásnál a karakterek negatív képét egy gömb palástjára helyezik el. A gömb forgatásával és billentésével a megfelelő karakter kiválasztható.

A **margarétakerekes** eljárás során a karaktereket egy tárcsa külső körgyűrűjén helyezik el. A tárcsa kialakítása hasonló a margaréta szirmának felépítéséhez. A tárcsa elfordulásával a megfelelő karakter a kívánt pozícióba helyezhető.



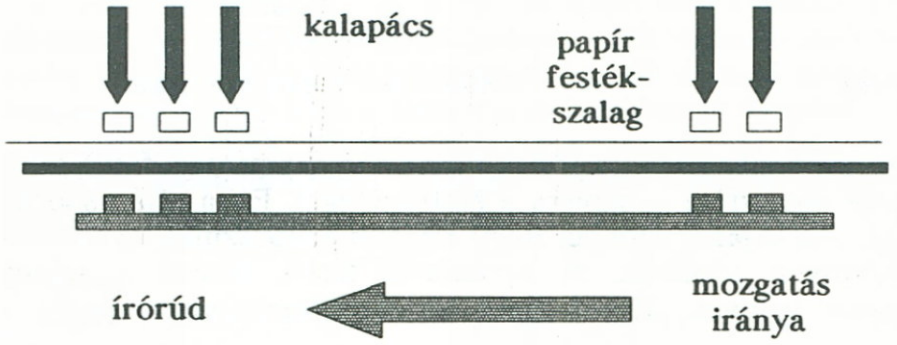
7.36. ábra

Gömbfejes nyomtató



7.37. ábra

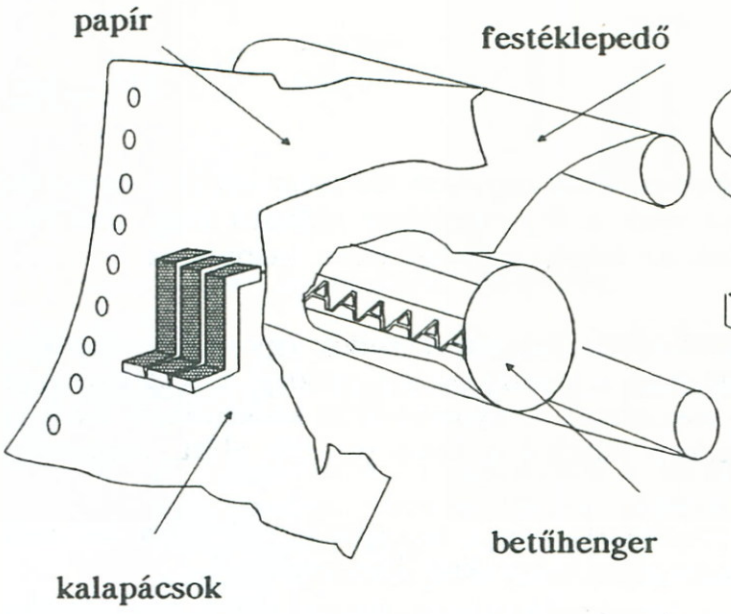
Margarétakerekes nyomtató



7.38. ábra

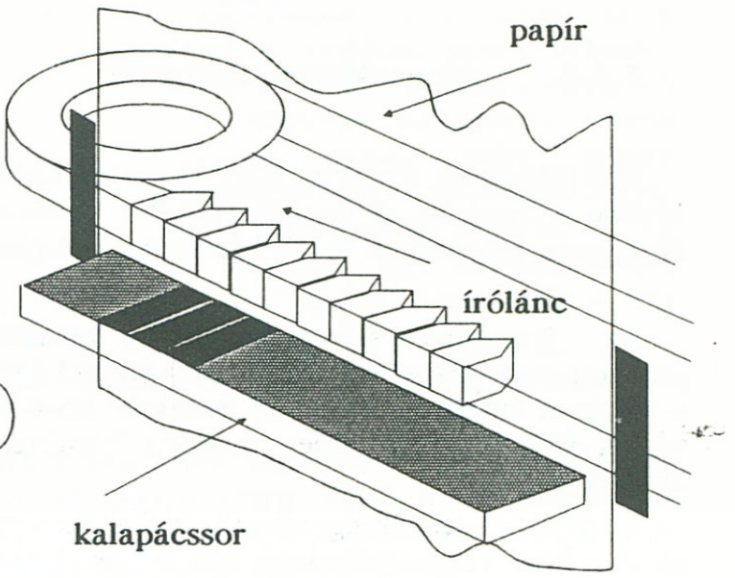
Írórudas nyomtató elvi vázlata. A rúdon egymás mellett, sorban elhelyezett karakterek negatív képét a kalapácsleütés nyomja a papírra.

7.4.3. Sornyomtatók



7.35. ábra

Íróhengeres nyomtató működési elve.



7.39. ábra

Íróláncos nyomtató működési elve.

Működésük lényege, hogy a karaktertároló egyszeri körbefordulása alatt egy teljes sor kerül kiírásra. A karakterkészlet csak a jelhordozó cseréjével változtatható. A működési sebességük szervezésük folytán igen nagy.

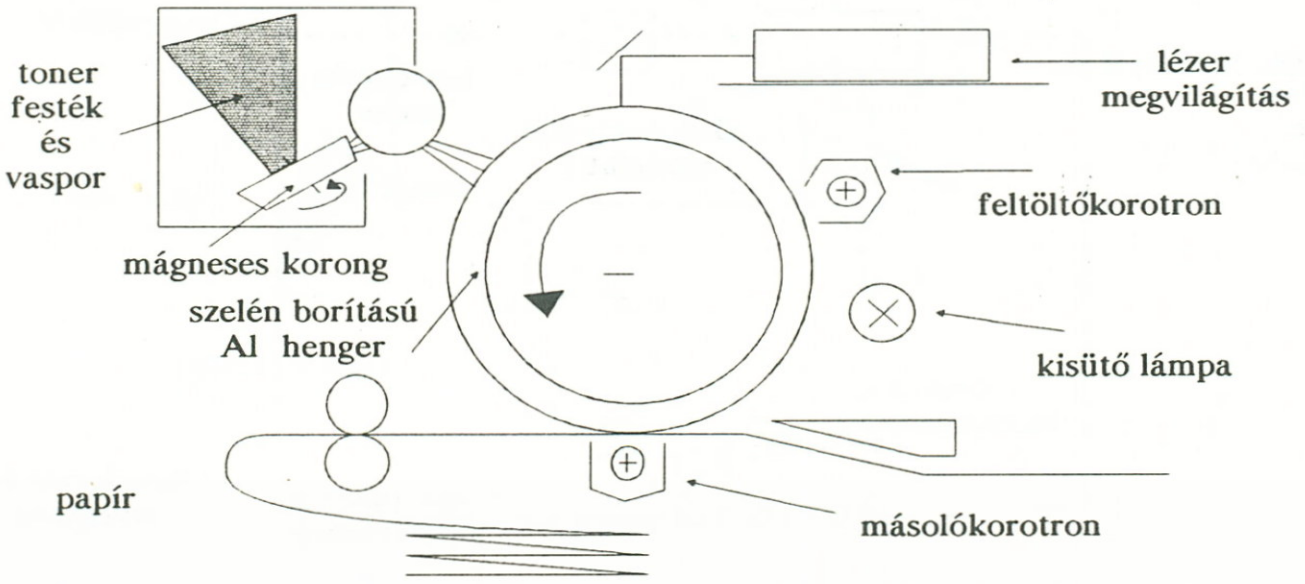
Az **íróhengeres nyomtatók** esetén a karakterek negatív jelét egy henger palástján helyezik el, mégpedig oly módon, hogy a henger alkotója mentén mindig ugyanaz a karakter legyen. A henger kerülete mentén pedig megtaláljuk a jelkészletben szereplő összes karaktert. A henger egyetlen körülfordulás alatt bármely pozícióban mindegyik jel nyomtatható. Természetesen ahhoz, hogy egyszerre egy sort le tudjunk nyomtatni a vezérlőnek tudnia kell, hogy a sorban melyik karakter hol helyezkedik el. Ezért minden karakter leütése során szükség van egy kódkereső eljárásra, amely megmondja a kiválasztott karakterek soron belüli helyét, azaz megmutatja, hogy melyik kalapácsot kell leütni.

A rendszer felépítése pontos mechanikai vezérléseket kíván meg, hiszen az állandóan forgó hengernek 1/20 millimétert sem szabad elmozdulnia mialatt a kalapácsok a kijelölt karaktereket leütik. (Ellenkező esetben a karakter elmosódik.) A nagy sebességű nyomtatás miatt a gumihengeres továbbítás nem alkalmazható. (A nyomtatás sebessége 1500 - 2000 sor /perc.)

Az **íróláncos nyomtatók** szerkezeti felépítése az íróhengeresére hasonlít, csak henger helyett egy vízszintesen körbefutó szalagra helyezik el a karakterkészletet. Ez kiküszöböli az íróhengeres nyomtatók betüelcsúszási hibáját, amely a nyomtatandó sor és henger egymásra merőleges mozgásából adódik. (Természetesen a kalapács leütések pontatlanságakor következik ez a jelenség be.) Mivel a láncszemek ténylegesen nincsenek összefűzve,hanem egy megfelelő sínrendszerbe futnak, ennél a megoldásnál megoldható a karakterek egyenként történő cseréje. (A nyomtatás sebessége eléri az 1100 sor/perc -et.)

7.4.4. Lapnyomtatók

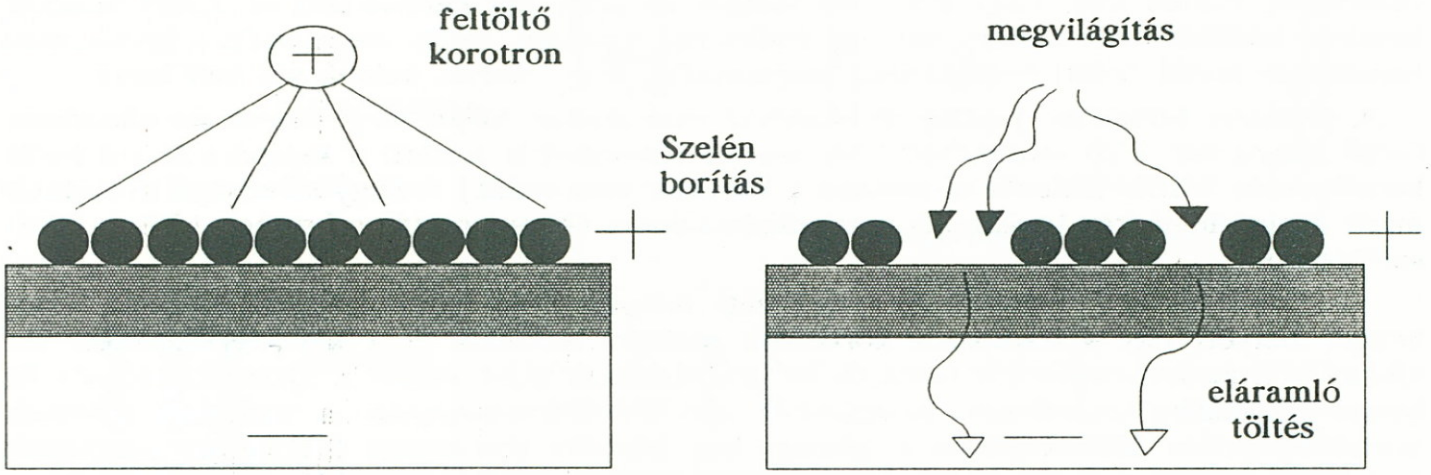
A **lézerxerografikus nyomtató.** Működési elve azon alapszik, hogy vannak olyan anyagok (pl szelén), amelyek fény hatására vezetővé válnak. A nyomtatók legfontosabb eleme egy negatív töltésű alumínium henger. (Palástjának mérete éppen a nyomtatandó lapnak felel meg.) Ennek felületét fényérzékeny szelénnel borítják. Egy vékony, nagy feszültségre töltött fémszálon (korotron) keresztül a koronakisülés jelensége alapján a henger felszínére töltést visznek. (A magas potenciálú korotron a levegő molekuláit ionizálja. Az alumínium henger negatív töltése miatt annak szelénnel borított felületén a pozitív töltések



7.40. ábra

Lézernyomtató elvi felépítése.

megtapadnak.) Megvilágítás hatására a szelén vezetővé válik, ellenállása csökken, ezért a pozitív töltések elszivárognak róla. Ha a megvilágítást pontszerű fényforrással végezzük (lézer), akkor a henger felületén töltött illetve töltés mentes pontokat hozhatunk létre. Ha most a töltött területekre alkalmas töltésű műanyag festéket szórunk, akkor azok ott megtapadnak, megmutatva ezáltal a henger feltöltött részeit. Ezek után egy alkalmas eljárással csupán át kell helyezni a papírra a festékszemcséket és máris megkapjuk a

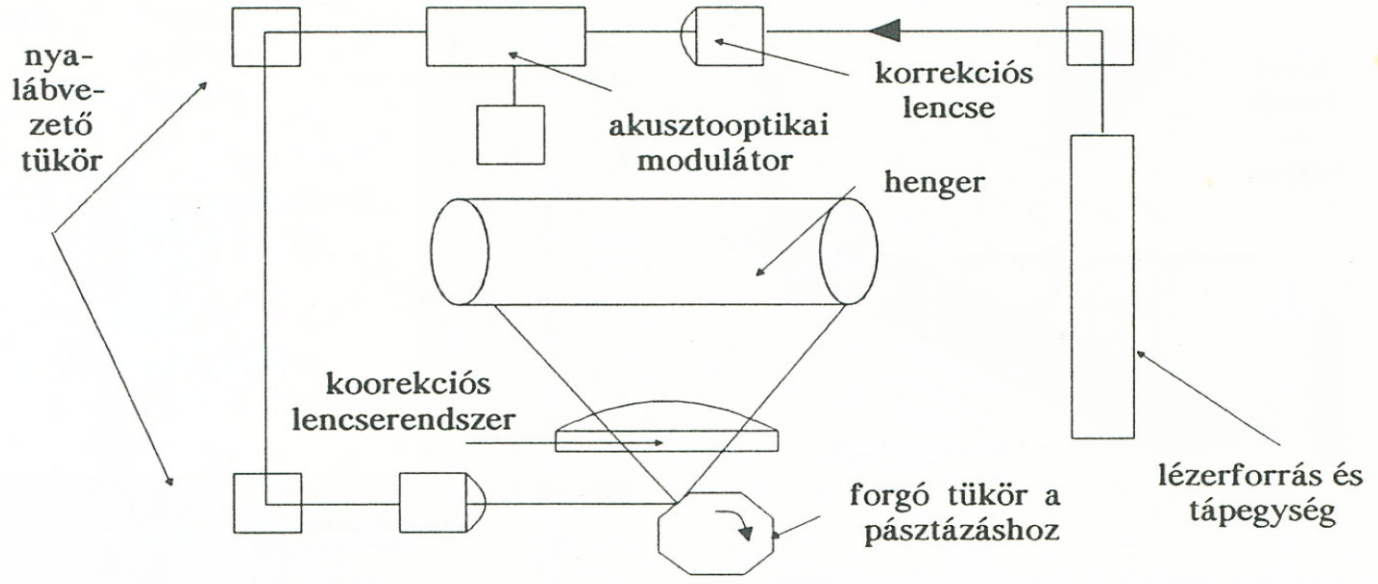


7.41. ábra

A látens kép kialakítása. Feltöltés és megfelelő helyen történő kisütés megvilágítás segítségével. (A fény hatására a szelén vezetővé válik.)

kívánt képet.

A festékfelhordás technológiája is érdekes, ugyanis mágneses úton történik. A finom vasporral elegyített festéket egy forgó kerék szórja ki a festékkazettából. A súrlódás során a műanyag festék feltöltődik és ezzel alkalmassá válik a kép „előhívására". A vasrészecskék az egyenletes kiszórást biztosítják,amelyeket egy forgó mágneskorong választ ki a festékből, ezért a hengerre csak festékszemcsék kerülnek.



7.42. ábra

Lézernyomtató optikai elrendezésének vázlata.

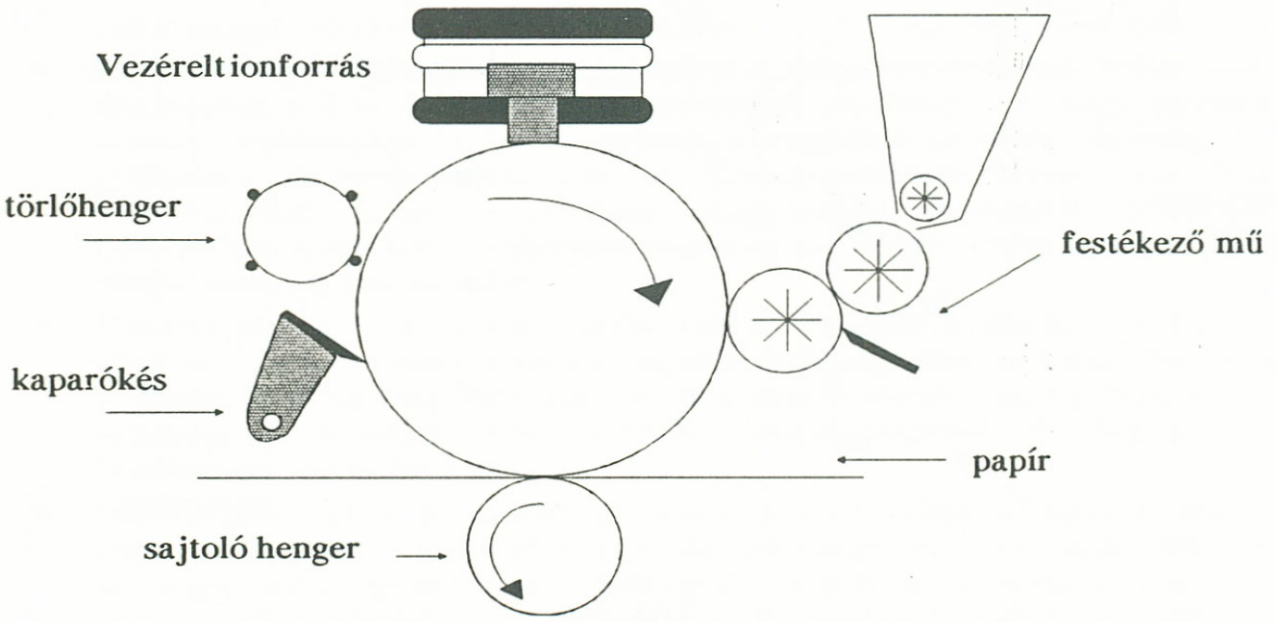
A kép papírra való áthelyezéséhez sem kell mást csinálnunk, mint a leporelló alá, ahol a legközelebb halad a hengerhez egy másoló korotront kell elhelyeznünk. Ennek magas pozitív töltése átszívja a festékrészecskéket a hengerről. Ez után már csak rögzíteni kell a festéket. Ezt melegítéssel oldjuk meg, beleolvasztjuk a festéket a papírba.

Az eljárásnak egyetlen érzékeny pontja van. Hogyan tudjuk a fényt a kialakítandó képnek megfelelően modulálni? Ez valóban nehéz kérdéseket vet fel. Ugyanis a pontszerű lézersugarat nem lehet folyton ki-be kapcsolgatni, mert a lézereffektus kialakitásához időre van szükség. Éppen ezért más eljárást kell kitalálni, például úgynevezett akuszto-optikai úton kell a fénysugár útját a henger felszínéről eltéríteni, ha ott töltést akarunk létrehozni.Ennek az eljárásnak az a lényege, hogy a nyomás alatt lévő folyadékok megváltoztatják a fény polarizációjának a síkját. A középiskolából ismert polarizáció jelensége alapján megfelelő beállítású polárszűrőkkel a fény teljesen kioltható (egymásra merőleges polárszűrők). Ha most a folyadék nyomásviszonyait változtatjuk, akkor mintegy ki/be kapcsolgatjuk a fénysugár útját. Most már csak azt a problémát kell megoldani,hogy hogyan pásztázzuk végig a fotohengert. Erre egy egyszerű megoldás kínálkozik: egy tükröt kell csupán alkalmasan megválasztott sebességgel forgatnunk és máris megoldottuk a problémát.

Ionsugaras nyomtatók működési elve a lézersugaras nyomtatókéhoz hasonló. Lényeges különbség abból adódik, hogy a lézersugár helyet ionsugarat használunk a keményalumínium henger feltöltésére. Az ionsugarat megfelelő módon vezéreljük, hogy csak oda jusson belőle, ahová szántuk. A festékezés alapelve teljesen hasonló, de itt nem elektrosztatikus úton visszük a henger felületéről át a festéket a papírra, hanem sajtoló eljárással (egy másik forgó henger nyomja a papírt a festékezett hengerhez). A hengeren maradó festéket egy kaparókés szedi le, ezért igen jó a festékhasznosításuk.

Előnyük, hogy lényegesen kevesebb energiát fogyasztanak, jó a festékfelhasználás, lényegesen nagyobb a működési sebesség (30 oldal/perc) és jóval hosszabb a nyomtató élettartama.

Hátrányuk, hogy meglehetősen drágák.



7.43. ábra

Ionsugaras nyomtatás elvi vázlata.

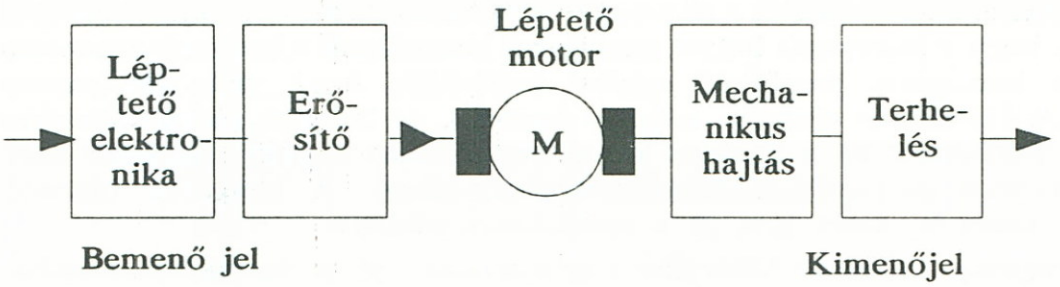
7.5. A plotter vagy rajzgép

A számítógéppel segített műszaki tervezés (CAD) során előállított rajzok papíron történő megjelenítésére alkalmas eszköz. Alapvetően két típusát különböztetjük meg: ezek a tábla és a dobplotterek.

A tábla típusú plottereknél az írószerkezet $\pm X$; $\pm Y$ irányban végez egységnyi vagy félegységnyi mozgást a vízszintesen rögzített papír felett.

A **dobplotter**eknél két, úgynevezett csévélő hengeren helyezik el a papírt, amelyek a léptetőmotorral vezérelt rajzolóhengerre feszítik azt. A toll csak a dob tengelyével párhuzamosan képes mozogni (X irányú elmozdulás). A másik Y irányú elmozdulást a dob illetve a csévélőhengerek megfelelő irányú elmozdulása pótolja.

A kétirányú mozgás lehetővé teszi tetszőleges görbék megrajzolását, függetlenül attól, hogy digitális vagy szervovezérlést alkalmaznak az X illetve Y irányú mozgást biztosító

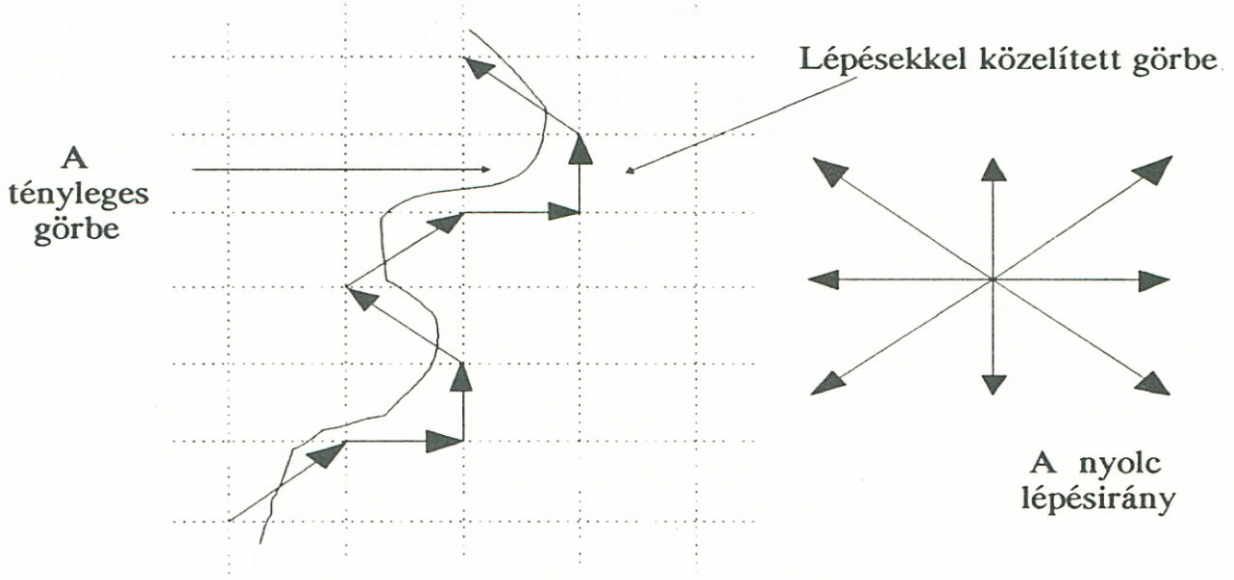


7.45. ábra

Digitális motor-vezérlés hatás-vázlata.

motorokmeghajtására.

Digitális hajtás melett, (7.44. ábra) ha mindkét irányban egységnyi elmozdulást tesz meg az írófej akkor 8 elemi vektorral közelíthetjük a görbét. Ha féllépéseket is megengedünk



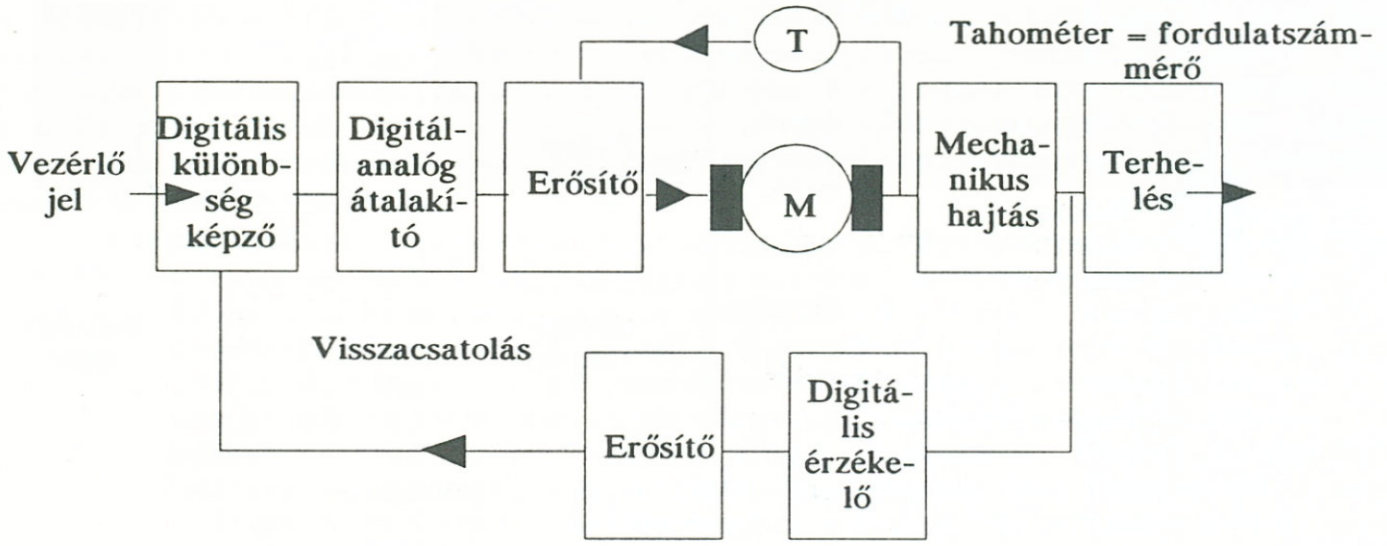
7.44. ábra

Digitális fejmozgatásnál alkalmazott 8 lépéses elrendezés.

akkor összesen 24 irányvektort kapunk, ami nyilvánvalóan precízebb közelítést, jobb felbontást takar. (7.45. ábra)

Szervohajtás esetén folyamatos kiírást valósítunk meg. Ebben az esetben a digitális bemenetet alkalmas interpolátor áramkör közbeiktatásásval egy szervovezérlésű analóg tollmozgató motorhoz. Ezzel a megoldással akár 0,01 mm felbontási fínomságot érhetünk

el. Az 7.46. ábrán látható módon a digitális érzékelő állandó időközönként (néhány ms-onként) jelenti a különbségképzőnek a kimenet (tollpozíció) állapotát. A különbségképző



7.46. ábra

Szervorendszerű hajtás blokkvázlata.

ennek megfelelően működteti az analóg motort. Ez a megoldás bonyolultabb és drágább a digitális hajtáshoz képest, de igen jó minőségű rajzok készíthetők ezekkel a berendezésekkel.

A plotterek általános jellemzői:

» A rajzfelület mérete — általában csak síkbeli gépek esetén definiálják A3-as méretig terjed. Dobplotterek esetén leginkább csak a papír szélességét emelik ki, hiszen hosszirányban tetszőleges lehet. A tényleges méretet a tekercselő hengerekre felrakható papír mennyisége határozza meg.

» A felbontóképesség — megmutatja, hogy egy tetszőleges görbe milyen pontossággal közelíthető. Értéke általában 0,01 és 0,1 mm közé esik.

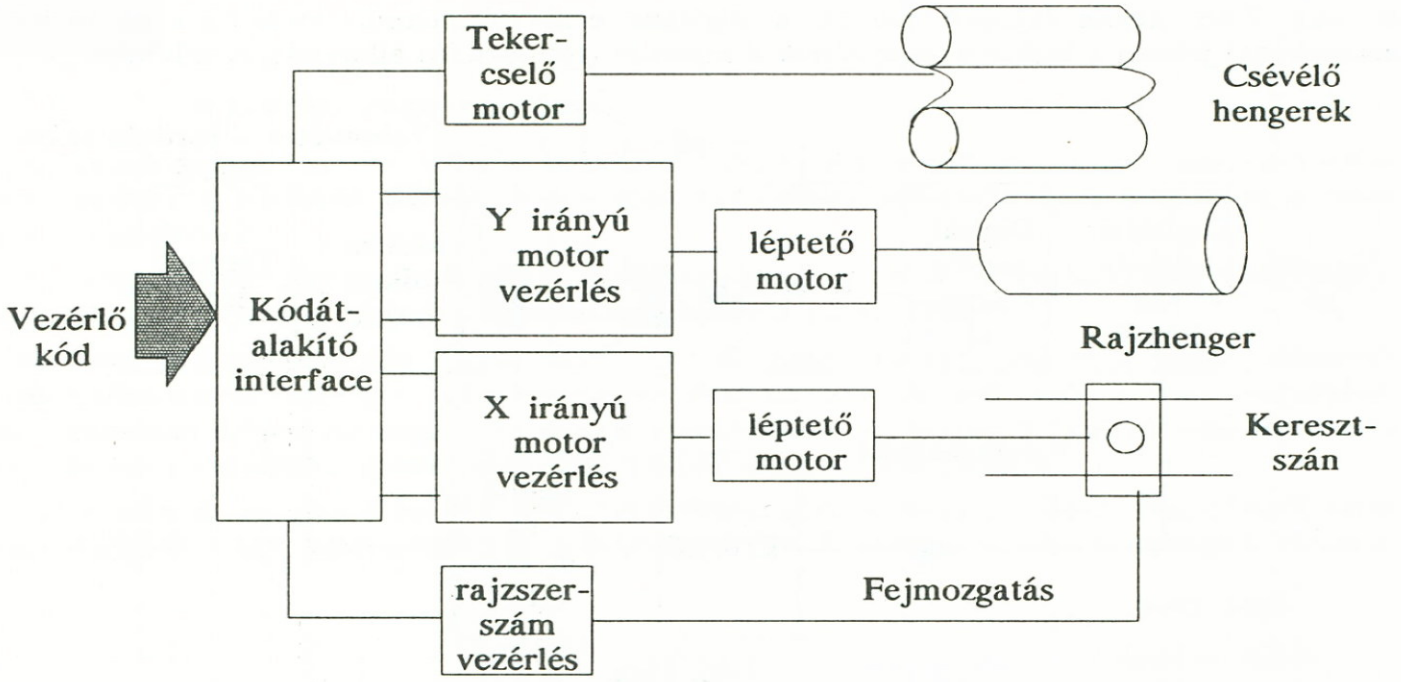
» Rajzsebesség — egyértelműen meghatározza a rajz elkészítésének időtartamát.Ma általánosak a 300 és 800 mm/sec sebességű rendszerek. A nagy rajzsebesség komoly problémákat is felvet, ugyanis a megfelelő sebesség eléréséig fel kell gyorsulnia a rendszernek, ami az üzemi értékhez képest más rajzolási paramétereket jelent. A sebesség és gyorsulás mindenkori értékének az ismeretében megfelelő hibakompenzáló program írható, amely a felhasználó elől elrejti ezeket a problémákat.

» Vonalvastagság — a rajzoló eszköz (csőtoll) függvénye, általában 0,1 és 2 mm közé esik. Természetesen a vonalak egymás mellé rajzolásával bármilyen vastagság is elérhető. Adott rajzszerszám esetén a kapott vonalvastagság függ a rajzolás sebességétől, a rajzhordozó nedvszívó tulajdonságaitól illetőleg a használt festékanyag jelemzőitől is.

» Fedettség — ez a paraméter is, hasonlóan az előzőhöz, nagyon sok egyéb tényező függvénye (például a rajzolás sebessége, vonalvastagság stb). A cél az, hogy minél egyenletesebb fedettséget érjünk el, ugyanis sokszor szükség van a rajz másolására is, ez pedig nem megfelelő fedettség esetén szinte lehetetlen.

» Ismétlési pontosság — megmutatja, hogy ugyanazt az ábrát újból megrajzolva legfeljebb mekkora eltérésre számíthatunk. Ez általában 0,1 mm-es érték vagy ennél kisebb.

Végezetül tekintsük meg a rajzgép blokkvázlatát (7.47. ábra)



7.47. ábra

A dobplotter blokkvázlata.

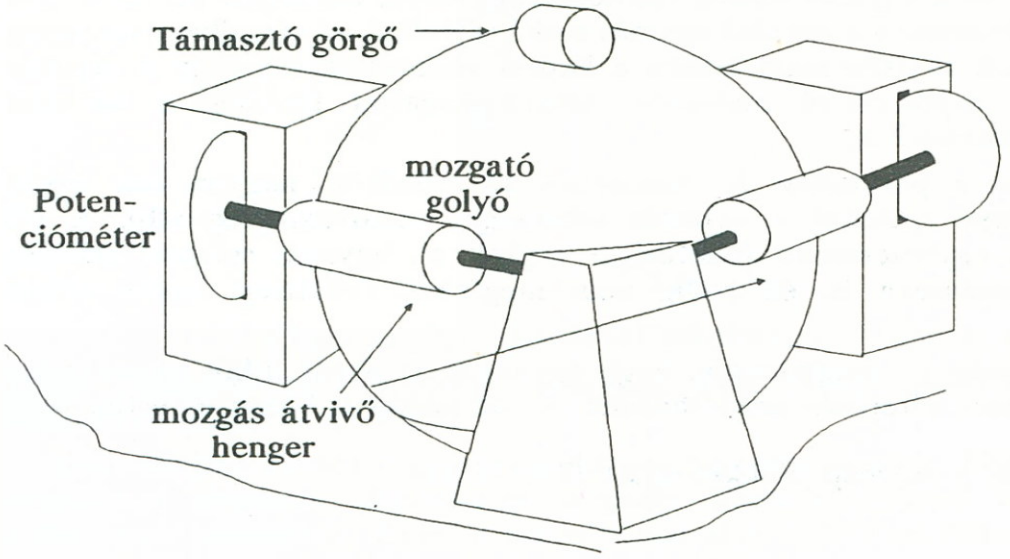
7.6. Az egér

Olyan adatbeviteli eszköz, amelyet síklapon mozgatva, a kurzort irányíthatjuk a képernyő felületén. Alapvető jellemzője a feloldóképesség ez általában 40 — 800 impulzus/cm tartományban mozog.

Az egerek csoprtosítása.

» A **mechanikus egér** belsejében a mozgást egy szilikon, vagy gumigolyó veszi fel. Ez adja át a mozgást két, egymásra merőleges hengernek, amelyek tengelyei kis potenciométereket forgatnak, vagy kapcsolókat kapcsolgatnak. Ezzel elektromos impulzusokat váltanak ki. Ezeket leszámolva kaphatjuk meg az egér X illetve Y irányú elmozdulását.

» Az **optomechanikus egér** felépítése nagyon hasonlít a mechanikusra, de itt a kódátalakító egy megfelelően fogazott tárcsa (maratott üveglap), amelynek fogai egy fototranzisztorra eső fénysugár útját szaggatják meg. Ezzel elektromos



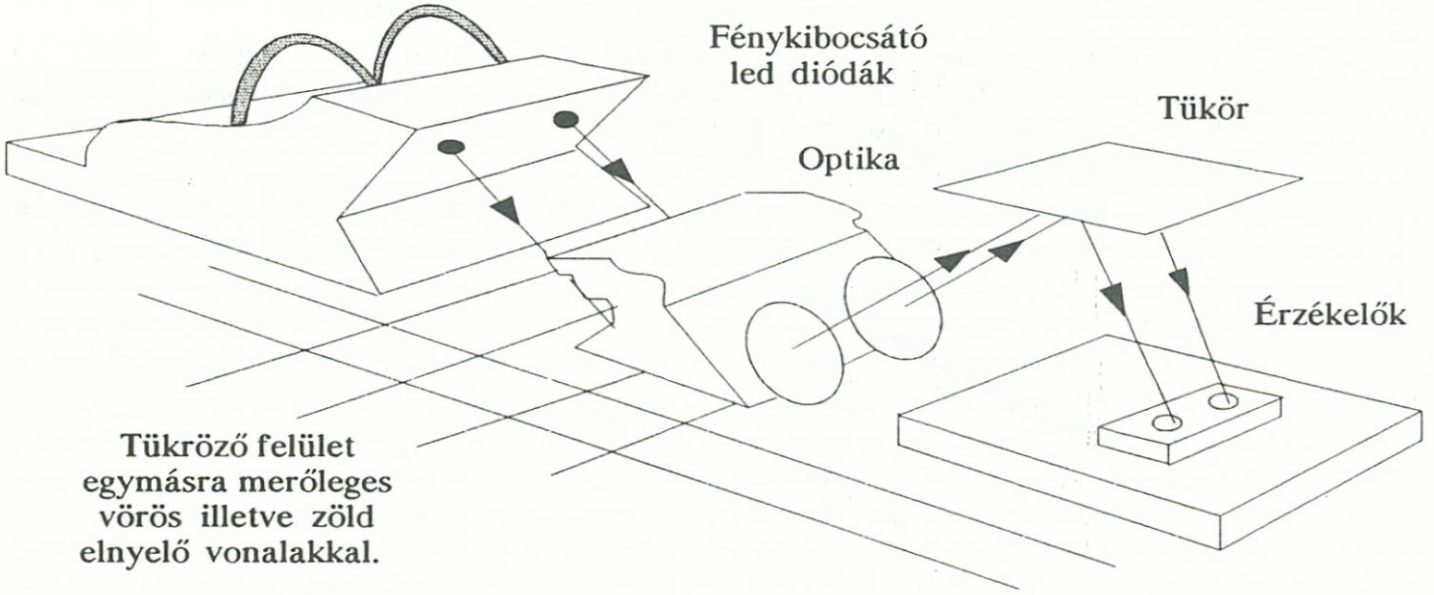
7.48. ábra

A mechanikus egerek erőátviteli szerkezete.

impulzusokat váltanak ki. Az előre-hátra mozgás megállapításához két fénykapu szükséges.

Mindkét fent ismertetett megoldás hátránya,hogy mozgó alkatrészeket tartalmaz, amelyek súrlódásos kapcsolatban vannak egymással, illetőleg az egér alatti felülettel. Ez óhatatlanul kopáshoz vezet, amely az egér felbontóképességének, érzékenységének romlásához vezet. (Az egerek felbontása megmutatja, hogy egységnyi hosszúságon hány impulzust állítanak elő. Ez egyértelműen befolyásolja a kurzor mozgásának finomságát.) A vezető görgő kopásának és zsírosodásának elkerülése végett speciális keményműanyag lapokat használunk egéralátétként (mouse-pad).

» Az **optikai egerekben** nem találunk mozgó alkatrészeket. Speciális mintázatú lemez felett mozognak. Legtöbbször két fénykibocsátó diódát tartalmaznak. A dióda által kibocsátott fény az alaplapról visszaverődik, amelyet egy tükör a detektorra (fototranzisztor) juttat. A fényérzékelő a fénymintát impulzussorozattá alakítja. A mozgás irányára a detektorok által kialakított impulzusok fázisviszonya alapján következtethetünk. A két főirány (X,Y) meghatározásához általában két, különböző színű fényt kibocsátó diódát alkalmaznak. Például a függőleges irányban az alapon vörös, a vízszintes irányban zöld fényelnyelő anyagot találunk. A vörös illetve zöld fénysugárra érzékeny fototranzisztor a rácsvonalak



7.49. ábra

Az optikai egér felépítése.

felett elhaladva kioltást érzékel. Ebből megállapíthatjuk, hogy milyen irányban történt az elmozdulás.

» **A rádió egerek** lényegében a fent ismertetett megoldások valamelyikéhez tartoznak azzal a különbséggel, hogy itt nem vezetékkel csatlakoznak a számítógéphez,hanem a kapott impulzusorozatokat rádiófrekvenciás jellé alakítják és kisugározzák. Ezt az jelet egy vevő áramkör érzékeli a számítógép belsejében. A sugárzás frekvenciája úgy van beállítva, hogy ne legyen kényes az egyéb gép keltette zavarokra. A megoldás előnye az, hogy a géptől szinte tetszőleges távolságban (néhány méter) és helyen használhatjuk az egeret, nem vagyunk a géphez láncolva.

Az egerek családjában az X illetve Y irányú impulzussorozatokat soros formára konvertálja az egérbe épített áramkör (TTL vagy RS-232 szintű kivezetések) és ilyen módon csatlakozunk a számítógéphez. A gépbe illesztett csatolókártya ezt a jelsorozatot

állítja vissza. Megállapítja az impulzusok számát illetve fázisviszonyait, amelynek segítségével a kurzor a képernyőn pozícionálható.

7.7. A scannerek

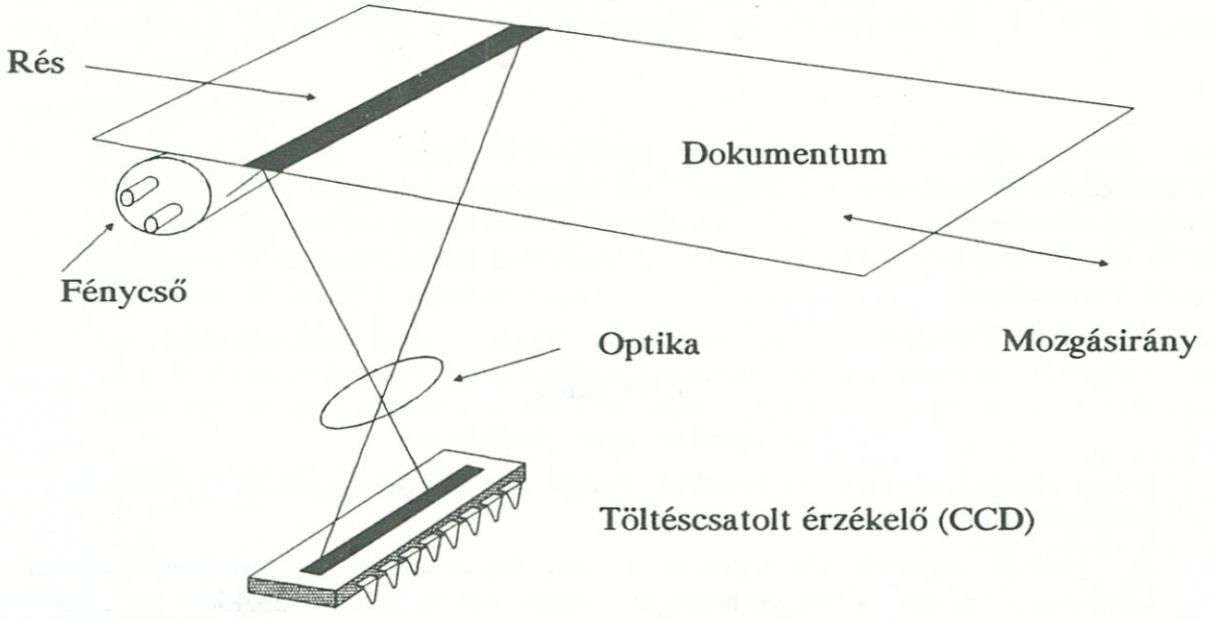
A **scanner** ek vagy más néven optikai letapogató egységek feladata: az imformációhordozóra (bizonylat, kép stb.) eső fény segítségével annak letapogatása, majd a reflektált fény elektromos jelsorozattá alakítása.

A jel átalakítás módját illetően megkülönböztetünk: raszter-, rés- és mátrix típusú scannereket.

» A raster típusú scannerekben egy pontszerű fénysugár tapogatja végig a képet. A reflektált fény erőssége változik. (A képen lévő festék a megvilágítás hatására gerjesztődik — energiát nyel el — és közben kibocsájtja a megfelelő rezgésszámú fényt. Gondoljuk meg, teljesen sötét helyiségben sem a tárgyak alakját sem a színét nem látjuk. Csak akkor jelennek meg számunkra, ha valamilyen fénysugár megvilágítja őket. A tárgyakról visszaverődő és szemünkbe jutó fénysugár intenzitása és rezgésszáma hordozza a tárgyak alakjáról az információt.) A változó intenzitású fénysugár fototranzisztorra esvén benne a kép tónusának megfelelő feszültséget kelt. Ezt megfelelő áramkörrel impulzussorozattáalakíthatjuk.(Feszültségvezéreltoszcillátorok.)

» A rés típusú scannerek esetén kiterjedt fénnyalábot alkalmazunk. Ekkor az információhordozó letapogatását mozgó rés segítségével végezhetjük el. (A mozgás relatív volta miatt a hordozó is mozoghat a rés előtt. Hasonlóan működik a fénymásoló letapogató szerkezete is.)

» Mátrix típusú scannerekben a reflektált fény pontszerű fényérzékelők kétdimenziós tömbjére esik, amelyek villamos jelekké alakítják azt.



7.50. ábra

A fény letapogatása rés típusú scannerekben.

Manapság a fény letapogatására speciális CCD (Charged Coupled Device — töltéscsatolt félvezető eszköz) áramköri egységeket használnak. A fény energiája apró kondenzátorokat tölt fel. A töltés hatására kialakuló feszültséget egy vezérlő elektronika pontról-pontra tapogatja le.

A képeket szürkeségi fokozatokban tároljuk (Az emberi szem 190 szürkeségi fokozatot tud megkülönböztetni.) A letapogatott képpontok szürkeségi fokozatainak számát 4,6, 8 bites formában tároljuk. Ábrázolásukra többféle módszert dolgoztak ki, például a pont és

csíkraszter technika. A **pontraszter technika** estén a képelelmek sűrűségét az egymás melletti, különböző vastagságú képpontok adják. (A képpontok ábrázolására egy $n * n$- -es mátrixot használnak fel, amelyben a mező egyes pontjai 1-esek illetve 0-k.A képernyőn megjelenő képpontok ennek megfelelően lesznek különböző vastagságúak.)

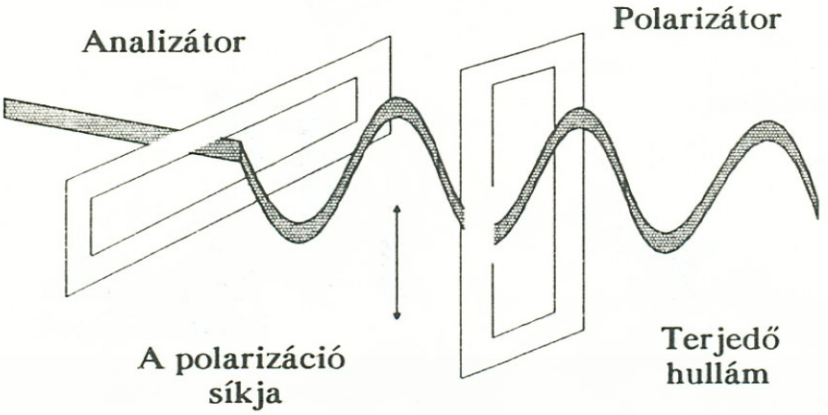
A csíkraszter eljárásnál a képelem szürkeségét különböző hosszúságú ferde vonalakkal állíthatjukelő.

7.8. Speciális perifériák

7.8.1. A magnetooptikai lemezegység

A hagyományos mágneses jelrögzítés esetén a mágnesezett rétegek iránya a hordozó felületével párhuzamosan helyezkedik el. Tudjuk, e fejezetben megismertek alapján, hogy milyen gondokat jelent ez a tény a lineáris bitsűrűség növeléséért folytatott harcban. Felmerül tehát a kérdés mi lenne, ha megváltozatanánk a tér irányultságát és a hordozó felületére merőlegesen rögzítenénk az információt. Úgy néz ki, ez a megoldás nem is lehetetlen, ugyanis vannak olyan anyagok (a porlasztással felvitt kobalt illetve krómötvözetek), amelyekben az eredő mágneses mező szívesebben áll be a hordozó felületére merőlegesen. — A haszon nyilvánvaló. — A lemágnesező erők gyengébbek, vastagabb mágneses hordozóréteg készíthető, ami nagyobb jelsűrűséget és nagyobb adatbiztonságot jelent. A probléma azonban abból adódik, hogy az ilyen módon felvitt jelek visszaolvasása csak úgy lehetséges, ha a fejet a hordózóra helyezzük és azon csúsztatjuk. Ez viszont, ilyen anyagok mellett a felület jelentős kopásához és ezzel az adatbiztonság jelentős gyengüléséhez vezet. Mit tegyünk tehát? A hagyományos mágneses jelrögzítési és visszaolvasási eljárás nem látszik sikeresnek. A nehézségeken azonban átsegít az a tapasztalat, hogy vannak olyan anyagok (gadolinium—vas, terbium—kobalt), amelyek koercitivitása magas hőmérsékleten jóval kisebb, mint szobahőmérsékleten. (Az úgynevezett Curie pont felett az anyagok elveszítik a mágneses tulajdonságaikat. A hőmozgás szétrázza a rendezett doméneket. A vas és kobalt Curie pontjai rendre: 768, 1075 °C.)

Azt a problémát kell csak megoldanunk, hogyan lehet a lemez felületét pontszerűen felmelegíteni. Ekkor ugyanis a külső, gyenge mágneses mező könnyen beállíthatja a megfelelő irányba a hordozó bitcelláját, hiszen ekkor — mivel jóval kisebb a koercitivitás — ehhez lényegesen kisebb mágneses mezőre van szükség. A probléma megoldását a pontszerűre fókuszálható lézersugár alkalmazása adja. Ezzel azonban még korántsem vagyunk készen, hiszen a hordozó felületéről az információt ki is kell olvasni és mint láttuk ez közvetlen indukciós úton nem megoldható. A megoldás kulcsa ismét csak a lézer. A fizikából tudjuk, hogy a mágnesezett anyag elfordítja a ráeső fény polarizációjának a síkját. (7.51. ábra) Ez a jelenség az úgynevezett Kerr-effektus. Attól függően, hogy felfelé vagy lefelé mutat a mágnesezett mező indukcióvektora az óramutató járásával egyező, illetve azzal ellentétes lesz a polarizált fény síkjának elfordulása. A kiolvasásnál legyengített lézersugarat használunk, és mindezt úgy, hogy közben nem érünk az adathordozó felszínéhez, vagyis nem koptatjuk azt.

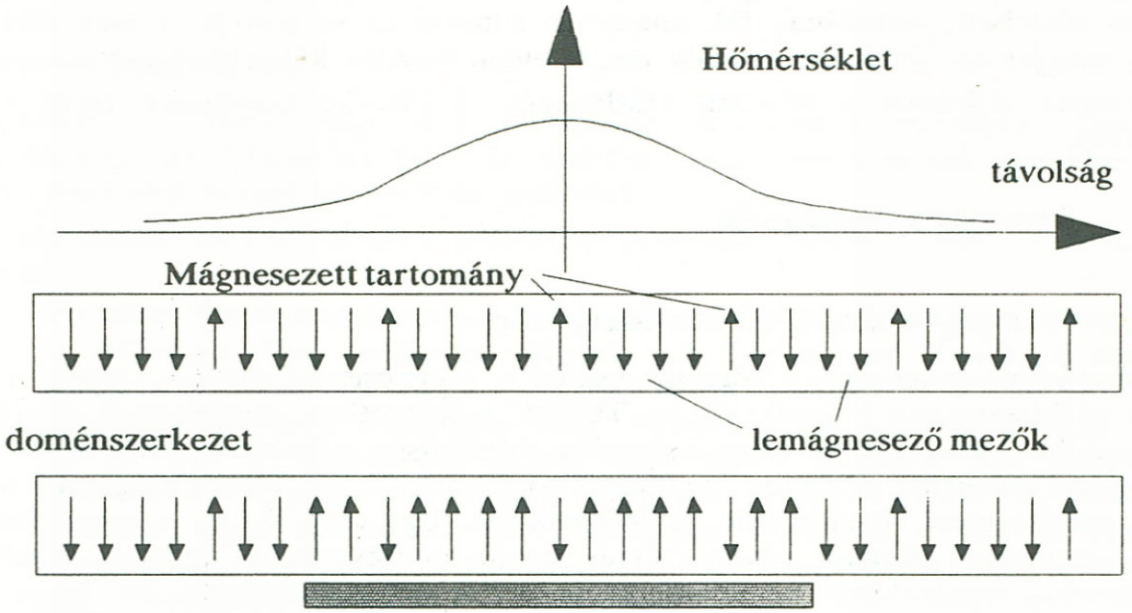


7.51. ábra

A polarizáció jelensége.

Ezek a lemezek természetesen törölhetők is. Ugyanis ha a hőmér-

séklet növekedtével a koercitivitás csökken, akkor a lemágnesező mezők, ha eredendően



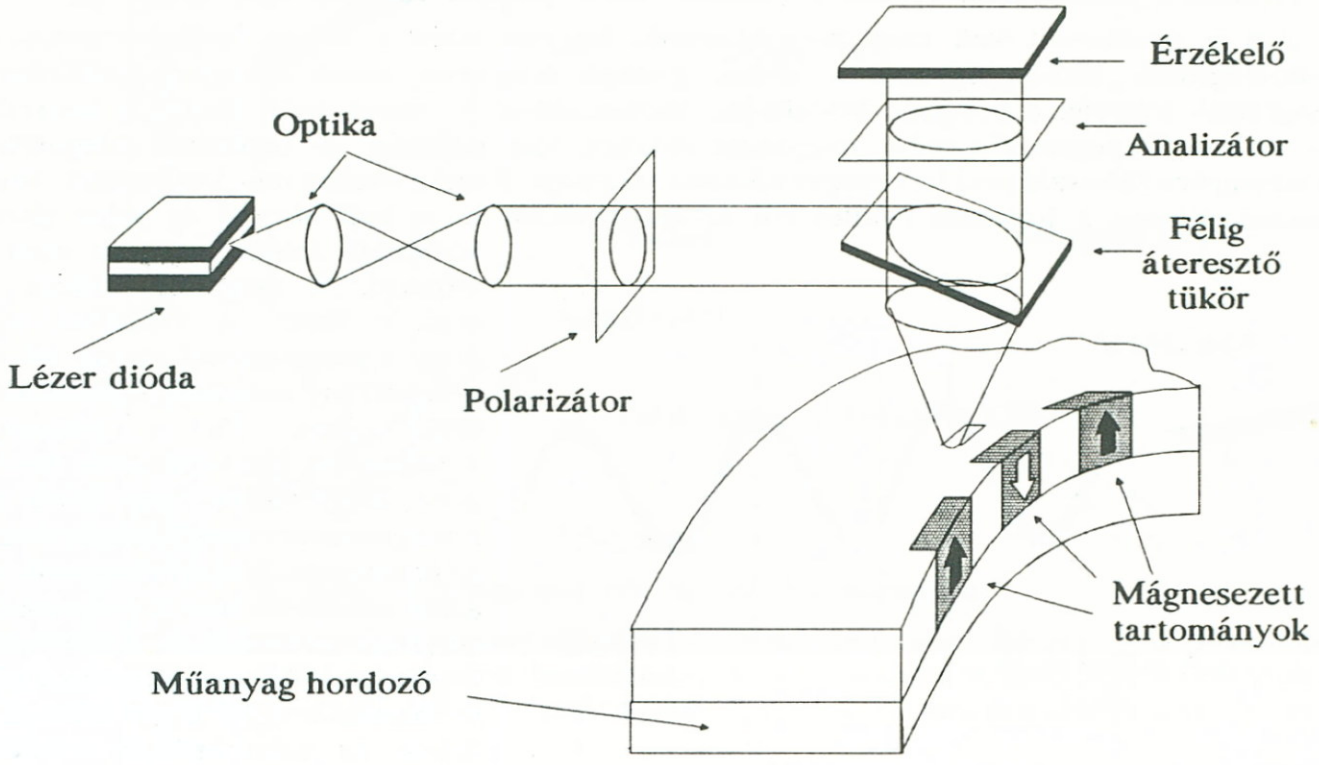
7.52. ábra

A magnetooptikai lemez törlése. A lemágnesező mezők ott fordítják át a mágnesezett tartományok irányát, ahol nagyobb a hőmérséklet és kisebb a koercitivitás.

erősek, képesek átfordítani a mágnesezett tartományok indukcióvektorának az irányát.

Mivel a lézersugár körülbelül 0,5 mikrométer körüli felületre fókuszálható és a sávot 0,1 mikrométer pontossággal követi ezért, igen nagy adatsűrűség érhető el.

Ennek ellenére a kutatók komoly problémával küzdenek. Ugyanis a fej tekintélyes tömegű (150 gramm) a hagyományos iró/olvasó fejekhez képest. A viszonylag nagy



7.53. ábra

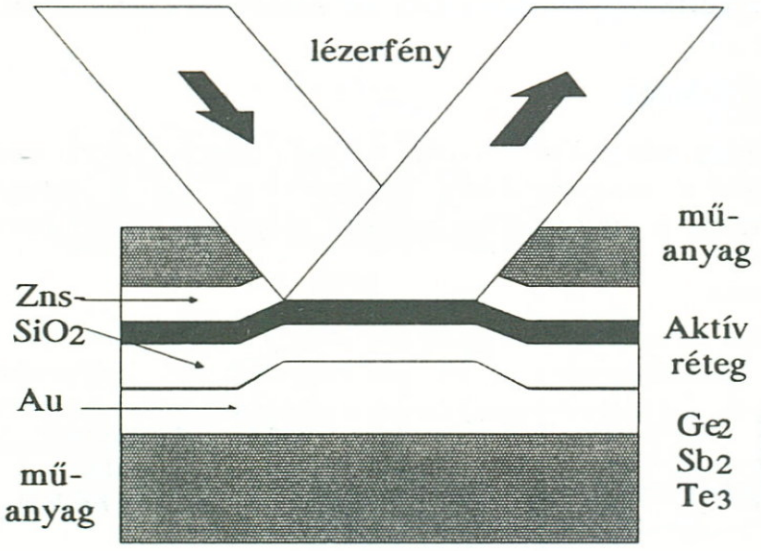
A magnetooptikai lemezegység iró/olvasó fejének felépítése

tehetetlenség miatt nem gyorsíthatók olyan jól, ezért a hozzáférési idő viszonylag nagy.Valószínűleg száloptikai megoldásokkal csökkenthető a fej tömege, ami lényegesen lecsökkentheti a jelenlegi hozzáférési időket (0,1 s).

7.8.2. Az optikai lemezek

A magnetooptikai eljárás használhatóságában kételkedők más utakat választottak és úgy tűnik sikerül a közeljövőben tisztán optikai úton működő Írható/olvasható lemezeket is piacra dobniuk.

Ezen eljárás kulcsa a változtatható reflexiós képességű anyagok alkalmazásában van. A rendezett kristályos anyag felszínére eső fénysugár sokkal inkább visszaverődik, intenzívebb, mint amelyik rendezetlen kristályrácsra esik. (A rendezetlen amorf anyag molekulái minden irányban szórják a fénysugarat. Ezért egy tetszőlegesen kiválasztott irányban csak jóval kisebb intenzitást észlelünk, mint a szabályos kristályszerkezetről visszaverődő fény esetén.)



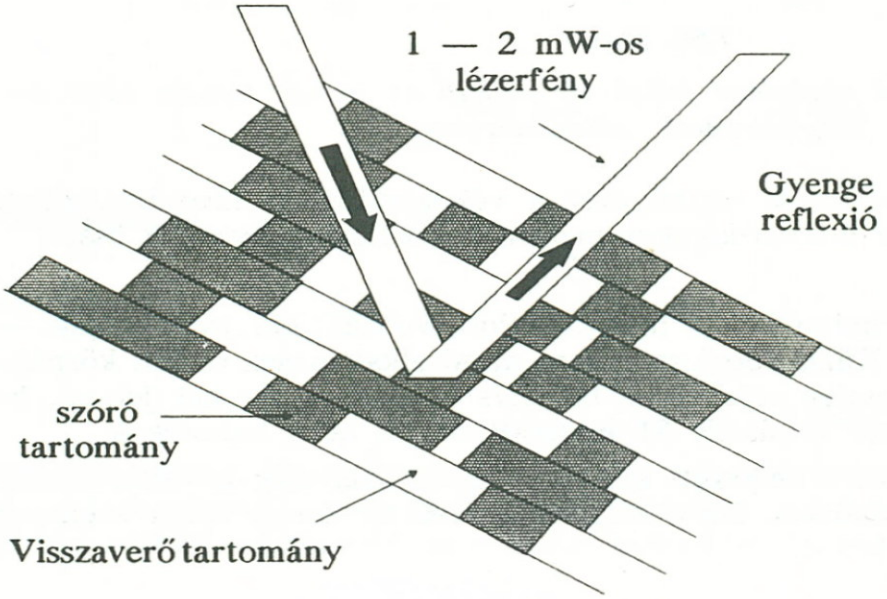
7.54. ábra

Optikai lemez szerkezete.

A kérdés csupán az, hogyan lehet e jelenséget adatrögzítésre felhasználni. A Panasonic cég mérnökei által kialakított eljárás alapján ez a következő módon történhet. Egy műanyag hordozóra felvitt körülbelül 0,2 mikron vastagságú arany réteg képezi a rendszer alapját. Ezután két szilíciumdioxid réteg közé zárt aktív réteget találunk, amely az információt tárolja.

A lemez működése a következő. A logikai 0-hoz erős, míg a logikai 1-hez gyenge reflexiót rendelünk. Ezt úgy valósíthatjuk meg, hogy logikai 0 esetén nem változtatjuk meg az anyag szerkezetét, a szabályos rendet, ezért intenzív visszavert fénysugarat kapunk.

Logikai 1 esetén a cellára érkező erős lézersugár által szolgáltatott energia szétzilálja az aktív réteg atomjait. A beírási folyamat végén az anyag gyorsan lehűl, a hő elvezetéséről a sziliciumdioxid réteg gondoskodik. A zilált amorf szerkezetről minden irányban visszaverődő fényből a megfigyelő



7.55. ábra

Az optikai lemezegység olvasási mechanizmusa.

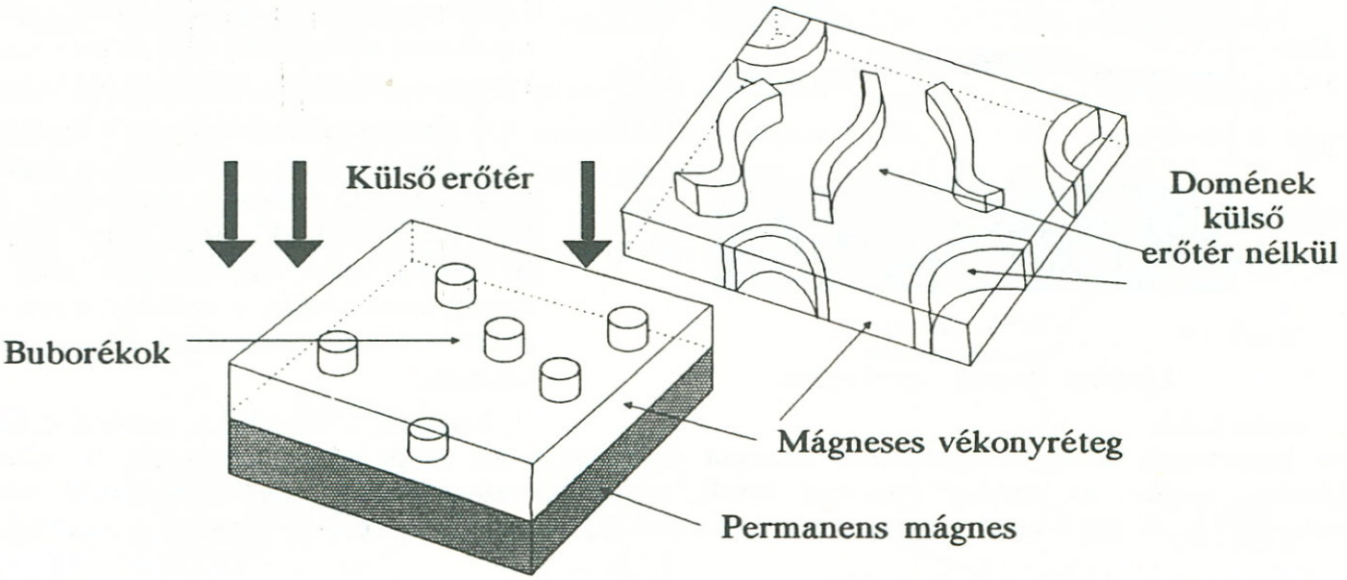
felé gyenge nyaláb érkezik, ami megfelel elvárásainknak. Az aktív réteg törlése (1-ből 0-ba) is viszonylag egyszerűen megoldható. A szétziláló lézersugár energiáját körülbelül felére legyengítjük. Ekkor az aktív réteg nem olvad meg, de a hőmozgás akkora, hogy az anyag összetétele folytán az újrakristályosodás megindulhat.

A kiolvasás természetesen a reflektált fény intenzitásának mérésén alapul. Sajnos ez a megoldás sem kielégítő. Ugyanis a fejnek, a magnetooptikai lemezegységhez hasonlóan, nagy a tömege. Ezért nehezen gyorsítható, ami tetemes körülbelül 90 ms-os hozzáférési időt jelent. A nagyobb adatbiztonság érdekében a lemez fordulatszáma is alacsonyabb, durván negyede a hagyományos lemeznek. (Az optikai lemez 1000, a hagyományos lemezegység pedig 3600 fordulatot tesz meg percenként.) Ez a tény szintén növeli a hozzáférési időt.

Az előzőekben említett megoldások jelenleg is fejlesztés alatt állnak, de remélhetően hamarosan sikerül ezeken a problémákon úrrá lenni. Ha ez valóban bekövetkezik, úgy nagyon valószínű, hogy gyorsan kiszorítják a hagyományos lemezegységeket, hiszen a jelenlegi változatban is 1—2 gigabájt kapacitással rendelkeznek és ezenfelül cserélhetők.

7.8.3. A mágneses buborékmemóriák

A memóriák, illetőleg a lemezes egységek jellemzőit ötvözik. Mozgó alkatrészeket nem tartalmaznak. Információhordozó közegük a mágnesezhető vékonyréteg. Azt a réteget paramágneses hordozóra viszik fel. Így tehát a vékonyréteg állandó mágneses mező felett



7.56. ábra

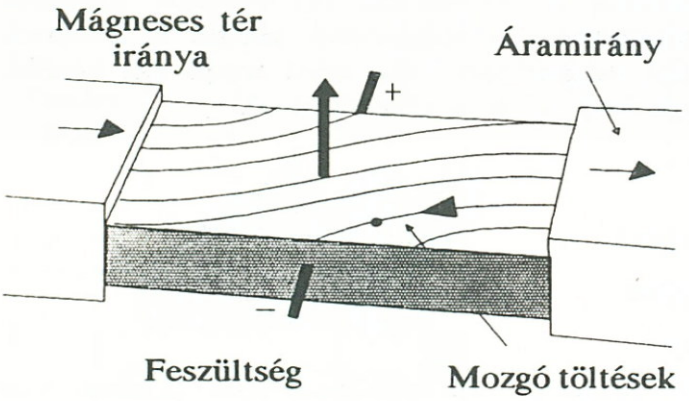
A mágneses doménszerkezet alakulása külső tér nélküli és annak hatása alatt álló mágnesezhető vékonyrétegben.

helyezkedik el. Ha áramjárta hurok segítségével a vékonyrétegben megváltoztatjuk a mágneses tér irányát, akkor az mikroszkópikus méretű buborékká zsugorodik össze.

Kis mágnesekkel ezeket a buborékokat mozgathatjuk, vonzhatjuk, taszíthatjuk — a buborék vándorol a közegben. Külső forgó mágneses mező alkalmazása esetén körpályára kényszerülnek. A mozgás sebessége elérheti a 100 m/s értéket is. Ez azt jelenti, hogy a mondjuk 10 cm átmérőjü kör kerületét 31 ms alatt futják be a buborékok.

A körmozgás irányát a lemezre helyezett kis paramágnesekkel meg is változtathatjuk. Ezek a mágnesek T illetve Y alakúak. Ha a külső mágneses tér forog, akkor ezek szinte adogatják egymásnak a buborékokat.

A buborékok detektálására apró vezetőhurkot vagy úgynevezett Hall-effektuson alapuló érzékelőt alkalmazhatunk. (7.57. ábra) Ezzel a megoldással memóriák készíthetők, ha a logikai értékeknek a következő hozzárandelési szabállyal adunk értelmet. Logikai 1-nek a buborékot tekintjük, logikai 0-nak pedig azt az esetet, amikor nincs buborék. Ennek megfelelően a körben bináris értékek vándorolnak. Alkalmasabbá tehetjük a rendszerünket, ha a már említett paramágnesekkel több apró hurkot hozunk létre. Ezekben természetesen ugyanúgy mozognak a buborékok, mintha csak egyetlen körpályát alakítottunk volna ki. A hurkokba vezérelhető mágneses elemekkel terelhetjük a buborékokat. (7.58. ábra)



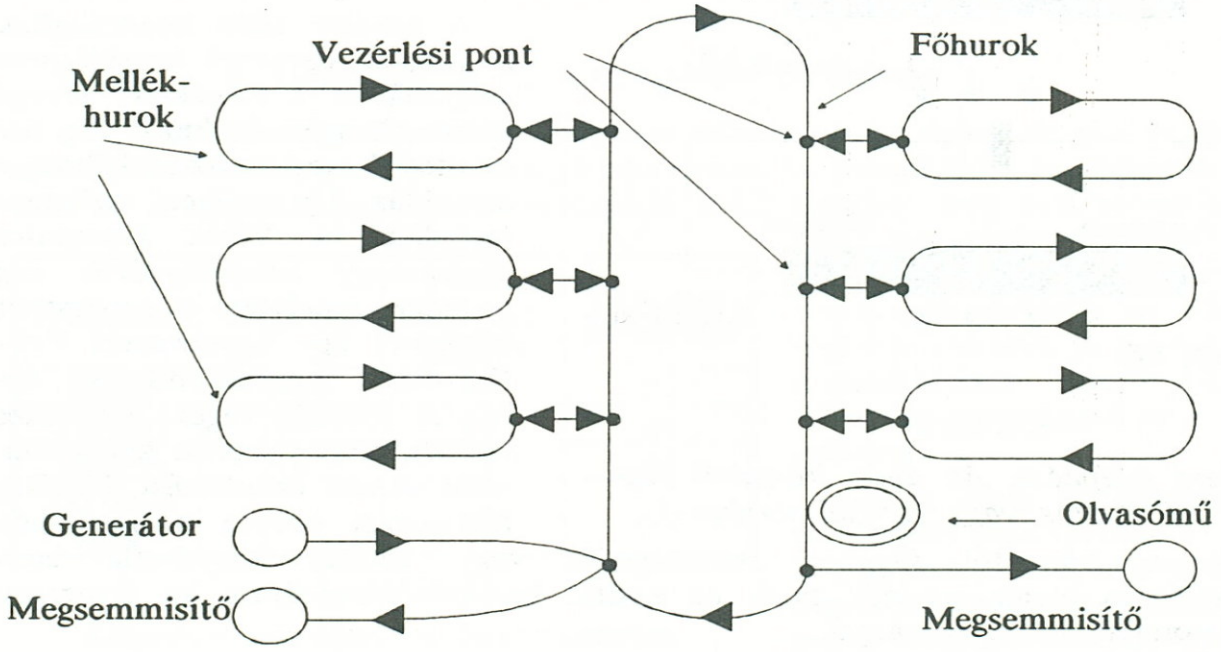
7.58. ábra

A Hall-effektus lényege, hogy a vezetőben mozogó töltéseket a mágneses mező eltéríti és ezzel a vezető oldalán potenciálkülönbséget hoz létre.

Ha a vékonyrétegben eltárolt információt ki szeretnénk olvasni, akkor a vezérelhető pontban át kell állítani a tér irányát és ezzel a középső hurokba (fő hurok, major hurok) kerül a mellékhurok tartalma (minor—hurok). A főhurok felett elhelyezzük a kiolvasó eszközt és amikor elhalad alatta a buborék, akkor benne áramlökés keletkezik, amely feldolgozható.

Az információ (buborék) megsemmisítésére külön megsemmisítő hurkot alakítunk ki. A buborékot ide beléptetjük, és a megsemmisítő tekercsre kapcsolt áram segitségével a buborék indukciójának irányát megváltoztatjuk. Ezzel a buborék „szétpattan", a környezetéhez képest megsemmisül.

Ez az eszköz is fejlesztés alatt van, de előnyei máris megmutatkoznak. Nincs szükség



7.57. ábra

Buborékmemóriák elvi felépítése.

mozgó alkatrészre, áramkimaradás esetén is megörzi tartalmát (permanens mágnes tartja fenn a buborékokat), az adatok egyik helyről a másikra könnyen mozgathatók, (CPU segítsége nélkül), egyszerűbb aritmetikai műveletek is elvégezhetők benne.

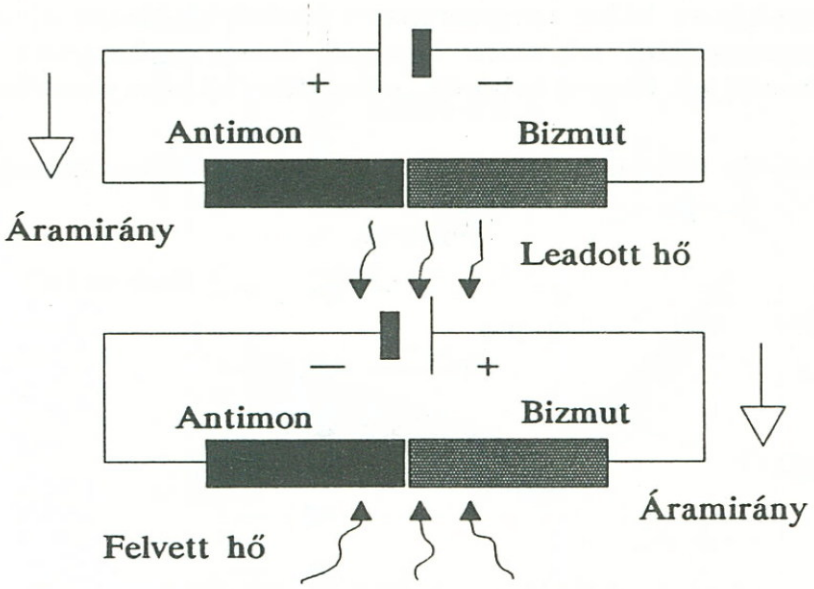
8. A teljesítőképesség növelése

A számítógépek fejlesztési törekvései általában a műveletek végrehajtási idejének csökkentését célozzák meg. A hagyományos sínszervezésű rendszerek esetén is léteznek olyan módszerek, amelyek segítségével e feladat megoldható, bár mint majd azt később látni fogjuk, a gyorsítás érdekében szakítanunk kell ezzel a struktúrával.

8.1. Alapvető eljárások

8.1.1. A rendszer órajelének növelése

Mint tudjuk a gép helyes működéséhez ütemező és szinkronizáló jelekre van szükség.Kézenfekvőnek látszik az a gondolat, hogy ha megnöveljük a rendszer órajelének sebességét, akkor gyorsabb, nagyobb teljesítményű gépet kapunk. Ez nyilvánvalóan igaz, de számolnunk kell bizonyos korlátokkal, amelyek határt szabnak a frekvencia korlátlan növelésének. Ugyanis ha növeljük az ütemezés rezgésszámát, akkor az integrált áramköri tok disszipált teljesítménye megnő, ami a rendszer üzemi hőmérsékletének emelkedéséhez vezet. Ez sajnos befolyásolja a műveletek végrehajtásának biztonságát és műveleti hibákhoz, szélső esetben az áramkör tönkremeneteléhez vezet.



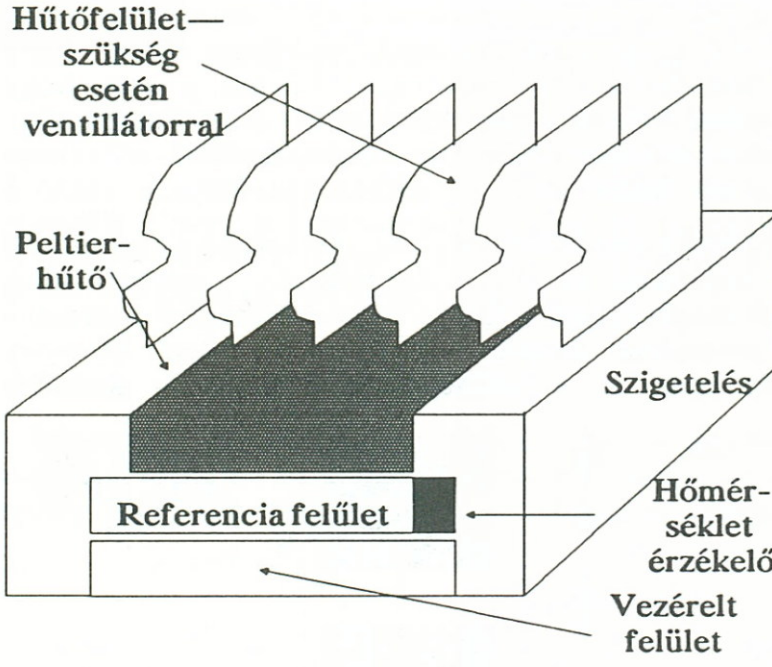
8.1. ábra

Peltier-elem felépítése. Az áram irányától függően hő leadás vagy felvétel történik.

Ha gondoskodunk a rendszer hűtéséről, akkor valóban jelentős sebességnövekedésre tehetünk szert. A rendszer működési frekvenciája ekkor közel 50 százalékkal is növelhető.

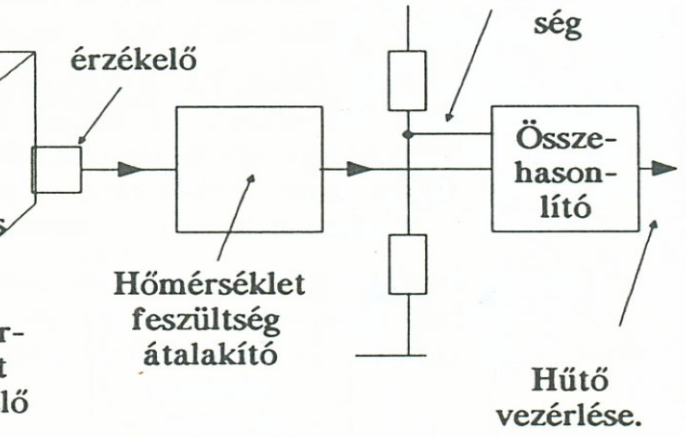
A hűtésre több lehetőségünk is van, az egyszerű ventillátoros megoldástól a folyékony levegő illetve nitrogén alkalmazásáig, bár ez utóbbi meglehetősen költséges megoldás. Új és igen szellemes megoldás a Velox Computer Technology hardvergyártó cég terméke, amelyben a processzor hűtéséről egy úgynevezett **Peltier-elem** gondoskodik (8.1. ábra). A jelenség régóta ismeretes (1834), amely szerint két különböző vezető érintkezési, illetőleg forrasztási helyén a Joule-hőn kívül az áram irányától függően felmelegedés vagy lehűlés következik be.A hőmérsékletváltozás olyan értelmű, hogy az ezáltal kiváltott termoáram az átvezetett elektromos árammal ellentétes irányú.

Részecske szempontból a fém találkozási helyén kialakuló érintkezési feszültség (Galvani feszültség) az áram irányától függően az elektronokat vagy gyorsítja, vagy lassítja. Az elektronok előálló energiatöbblete illetve hiánya a fémrács atomjaival való ütközések révén az érintkezési helyet felmelegíti (energiát ad a rácsnak), vagy hűti (energiát kap a rácstól).



8.2. ábra

Peltier hűtő szerkezeti felépítése.



8.3. ábra

A vezérlő a megengedett hőmérséklethatár túllépésekor beállítja az elemen átfolyó áramot.

Érdekesség, hogy például a 80486- os mikroprocesszor hűtés nélkül 85 °C-on éri el üzemi hőmérsékletét. Ha 10 °C-kal csökkentjük ezt az értéket, akkor közel másfélszeresére nő a félvezető élettartama és az áramkörön 25 %-kal nagyobb áram folyhat. Ha a Peltier-elem segítségével végezük a hűtést, akkor a 486-os processzorból, amely 25 MHz-n dolgozik akár 40 MHz-est is készíthetünk. (8.2. ábra)

8.1.2. A sínrendszer szélességének növelése

Az alkotó elemek teljesítőképességének növelése szép dolog, de mit ér, ha nem tudjuk kihasználni? Mit ér a nagy sebességű processzor, ha nem érkezik be időben a feldolgozandó adat és a központi egységnek várakoznia kell.? Ilyenkor meg kell keresnünk a rendszer szűk keresztmetszetét és változtatnunk kell ennek szerkezetén. A sínszervezésű rendszerek esetén a sín áteresztő képességének mértéke is gyenge pontja lehet a rendszernek. A mikrogépek világában jól nyomonkövethető ez a fejlesztés (8.1. táblázat)

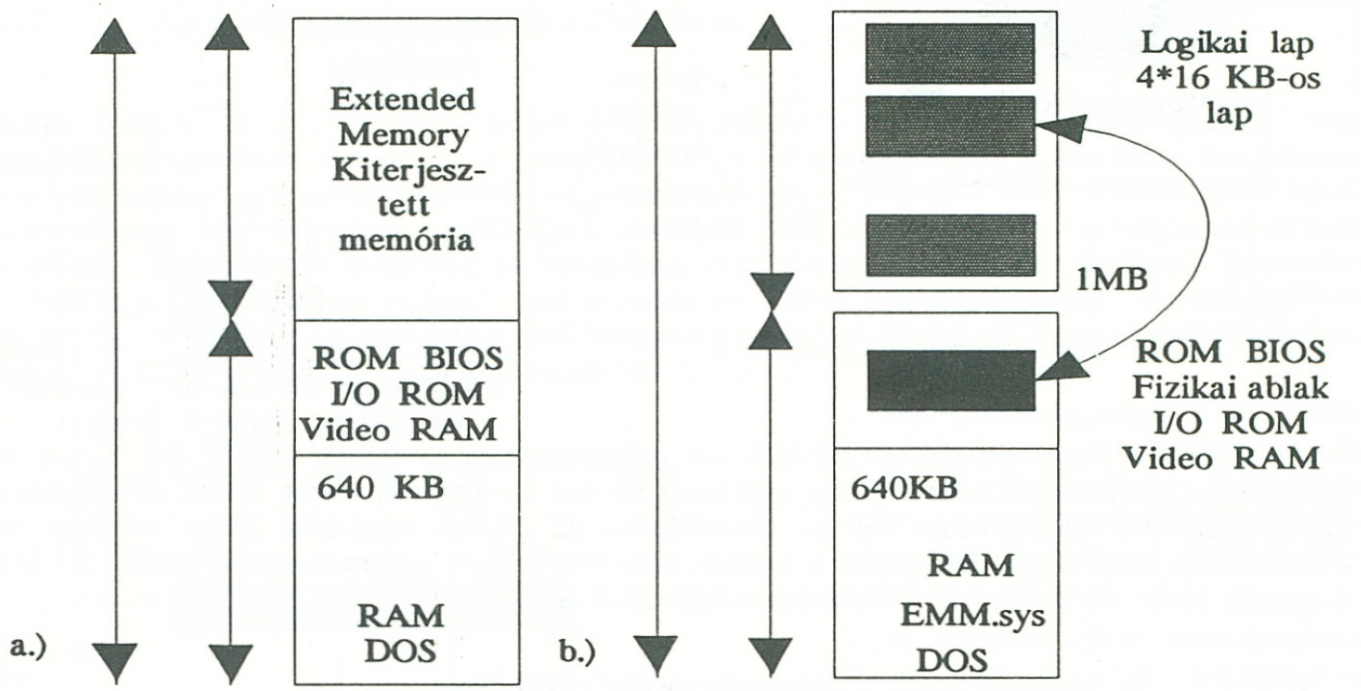
Processzor	Géptíous	Adat-busz szélessége	Címbusz szélessége
I8086	XT	8	20
I80286	AT	16	24
I80386	AT	32	32
D21064 Alfa		64	64

8.1. táblázat

Közismert processzorok buszszélessége.

Addig, amíg az adatbusz mérete az információ szállításának az idejét rövidíti le (egyszerre több bájt mozgatható — a gép szóhosszának megfelelően), a címbusz mérete a tárolókapacitás lehetséges értékét határozza meg és egyúttal a rendszer erőforrásainak növelését is jelenti. Mint tudjuk, ha a címbusz szélessége CS, akkor a fizikailag címezhető tároló kapacitás 2^{CS} . Ennek megfelelően 20 bites címbusz esetén 1MBájt, 32 bites címbusz esetén pedig 4 Gbájt kapacitású memória csatolható a rendszerhez. Ennek kezeléséhez természetesen megfelelő programra van szükség. (DOS alatt ez közvetlenül nem lehetséges.) Tudjuk, hogy a magasabb számozású processzorokat, illetve az általuk

meghajtott számítógépeket úgy készítik, hogy az előző rendszerekkel kompatibilisek legyenek, ami azt jelenti például, hogy a 286-os AT is úgynevezett **reál módban** ébred, hogy a 20 bites XT-re készült programok is futatthatók legyenek rajta. Ez azt jelenti, hogy közvetlenül csak 1MBájt RAM-ot tud kezelni, hiszen a buszméretét 20 bitre korlátozták. A 24 bites **védett módba** természetesen átkapcsolható, de ezt visszafelé már nem tudjuk megtenni. Akkor mit lehet kezdeni ezzel a nagy tárolókapacitással, amit a $2^{20} = 16 * 10^6$ bájt jelent. Két lehetőség között választhatunk: vagy szakítunk az operációs rendszerrel, hogy közvetlenül ki tudjuk használni az AT **kiterjesztett memóriáját,** (Extended Memory) vagy egy ügyes trükkel megkerüljük a fizikai címezhetőségből származó korlátot és úgynevezett **kibővített memória** kezelést EMS (Expanded Memory Specification) alkalmazunk. (XT-k esetén csak ez a járható útja a nagyobb memória



8.4. ábra

Az a.) ábra a 286-os AT memória felépítését mutatja. DOS alatt ezt a területet emulátor segítségével kibővített memóriaként kezeljük és a 64 KBájtos fizikai ablak segítségével nézhetjük végig ennek tartalmát. (b. ábra)

alkalmazásának.) Az 1 Mbájt fölé nyúló területet speciális módon kezeljük, ami azt jelenti, hogy az alsó memóriaszelet (1MBájt) tetején (a felső 382 Kbájtos tartományban) egy 64 Kbájtos ablakot nyitunk. Ez az ablak néz a bővitett memóriaterület megfelelő részére. Az EMS gondoskodik arról, hogy nyilvántartsa adatainkat hol helyeztük el ebben a tartományban, és honnan kell őket visszakeresni. Röviden a felhasználó a teljes memória területet bejárhatja anélkül, hogy érezné a fizikai címzés határait.

8.1.3. A processzor fő feladatának biztosítása

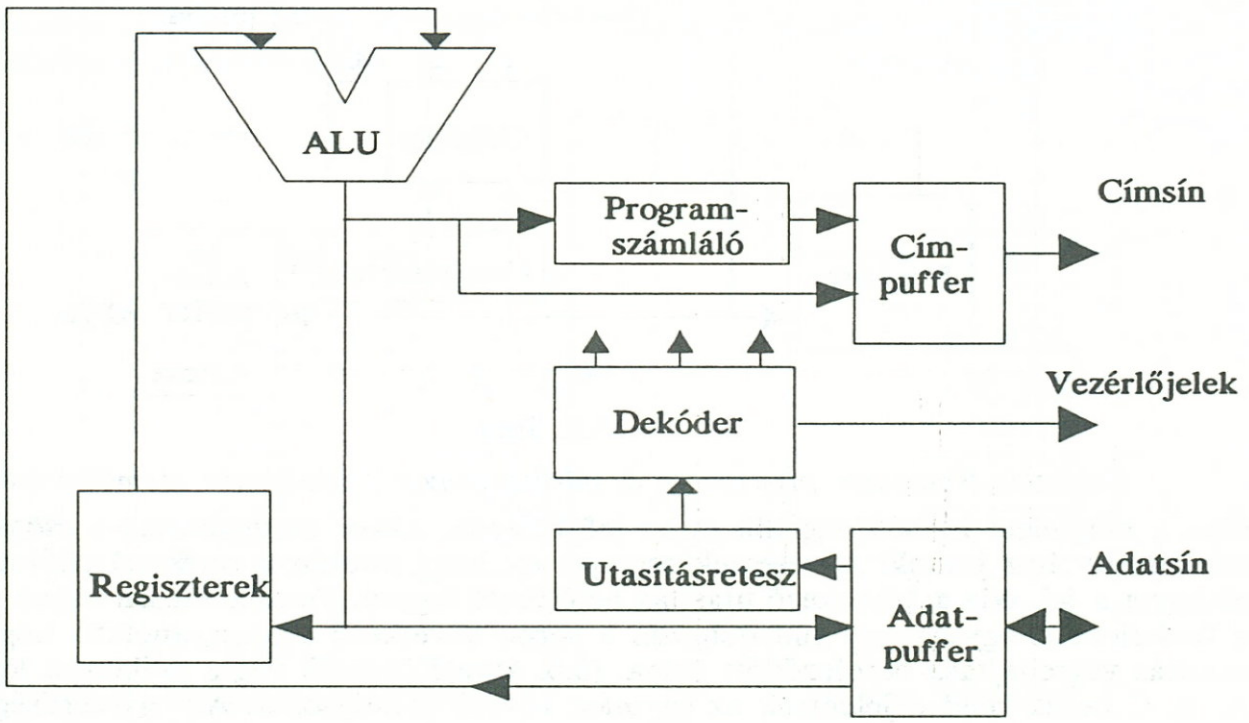
Ez azt jelenti, hogy mentesítjük a processzort bizonyos időigényes feladatoktól, amely miatt a feldolgozási feladatait szüneteltetnie kellene. A sínszervezésű rendszerek fejezetben már szóltunk arról a problémáról, hogy milyen gondot jelent, ha a központi egység állandóan a perifériát szolgálja ki. Mint tudjuk ezen feladat megoldására fejlesztették ki az I/O processzorokat, illetőleg a DMA vezérlő áramköröket. Az említett fejezetben már bemutattuk ezen módszereket.

8.1.4. Gyorsítótárakalkalmazása

A nagyméretű, nagy kapacitású tárak alkalmazása a tárak hozzáférési idejét növeli, ami a számítógép adatfeldolgozási képességét korlátozza. Ezért alkalmazzuk a gyorsító tárakat, amelyek viszonylag kis méretűek és nagy elérési sebességet biztosítanak. A gyorsabb elérés érdekében ezek tartalom szerint címezhetők. Ez azt jelenti, hogy a tárolt információ keresési kulcs alapján olvasható ki. A keresési kulccsal való összehasonlítás az összes rekeszre egyidejűleg történik.

8.1.5. Csővonal technika

Másnéven **pipelin** (instruction overlap, instruction lookahead, functional parallelism) technika azon az észrevételen alapszik, hogy az utasítások végrehajtása során a központi egység általában több hardver egységgel is kapcsolatba lép, azaz az utasítás több fázisra



8.5. ábra

Hagyományos processzor szerkezete.

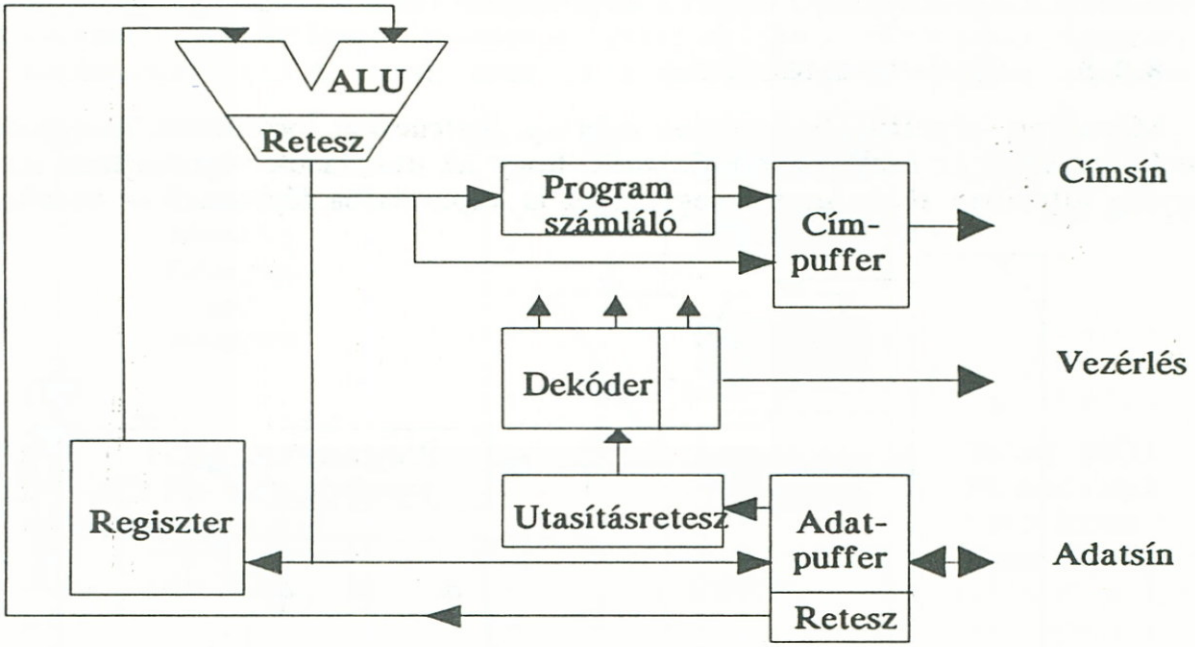
bontható. Ez lehetőséget teremt arra, hogy a műveletek némelyikét egyidőben, egymás mellett hajtsa végre a gép. Hogy ez mennyire gazdaságos megoldás azt talán az autógyárak példájával szemléltethetnénk. Ugyanis az új autógyárakban az első gépkocsi elkészítése igényli a legtöbb időt. Az ez után legördülő kocsik a miatt,hogy a szerelést részfolyamatokra bontják és átlapoltan egymás mellett végzik futószalagon, jóval kevesebb időt igényelnek.

Ha megnézzük a 8.5. ábrát, amely hagyományos processzorstruktúrát mutat, akkor kiderül, hogy a soron következő utasítás felhozatalára csak akkor kerülhet sor, ha az előző végrehajtása már befejeződött.Tekintsük át ebben a rendszerben egy utasítás végrehajtási folyamatát! Ez mint tudjuk a következő lépésekből tevődik össze:

- az első utasításcím kiadása,
- az utasítás felhozása,
- az utasítás dekódolása,
- az operanduszcím kiküldése,

» az operandusz behozatala,
» a kijelölt művelet elvégzése,
» a művelet tárolása a pufferben,
» az eredmény címének kijelölése,
» az eredmény tárolása a megfelelő helyen,
» és a következő utasítás felhozatala.

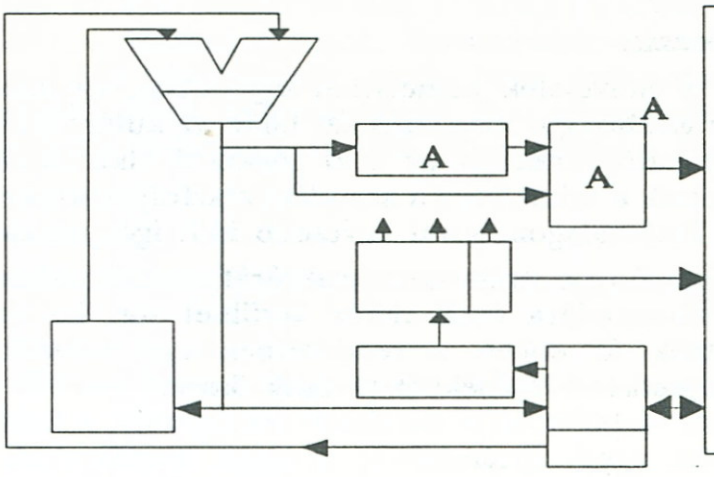
Ha most ezt a processzor szerkezetet a megfelelő helyeken reteszekkel látjuk el, amelyek



8.6. ábra

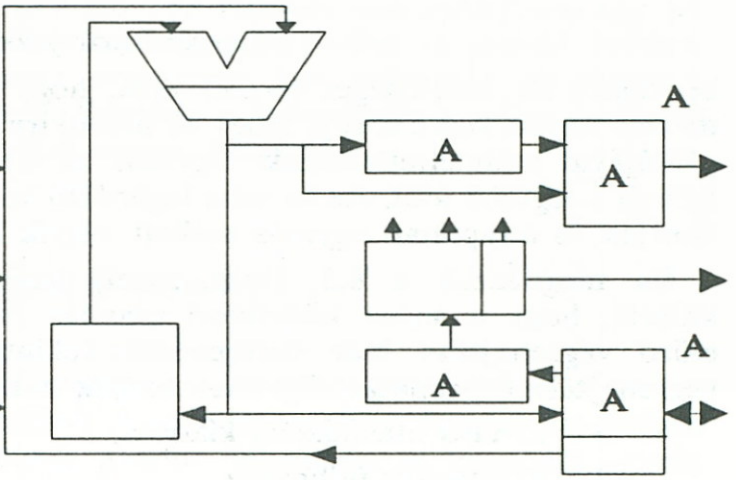
Reteszelt Neumann processzor a párhuzammos feldolgozás szemléltetésére.

csak a megfelelő időben engedik át az információt, akkor az úgynevezett **csővezetékes processzor** hoz jutunk. A reteszek szerepe az, hogy miután a retesszel ellátott funkciót elhagyta a jel, oda a következő utasítás betölthető legyen. Természetesen ebben az esetben a funkcionális egység már tud dolgozni a soron következő adaton, anélkül, hogy az előző utasítás végrehajtása befejeződött volna. (8.6. ábra)Kövessük végig működési lépéseit! Az A, B, C betűk rendre jelentsék az egymást követő utasításokat. Az egyszerűség kedvéért.



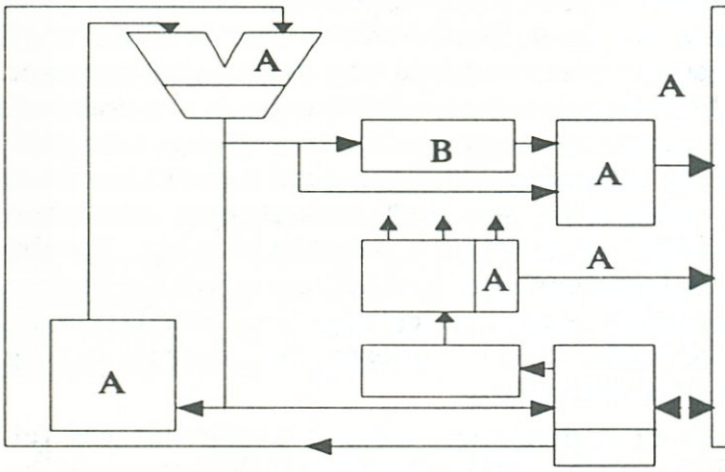
8.7. ábra

Az első utasításcím kiküldése.



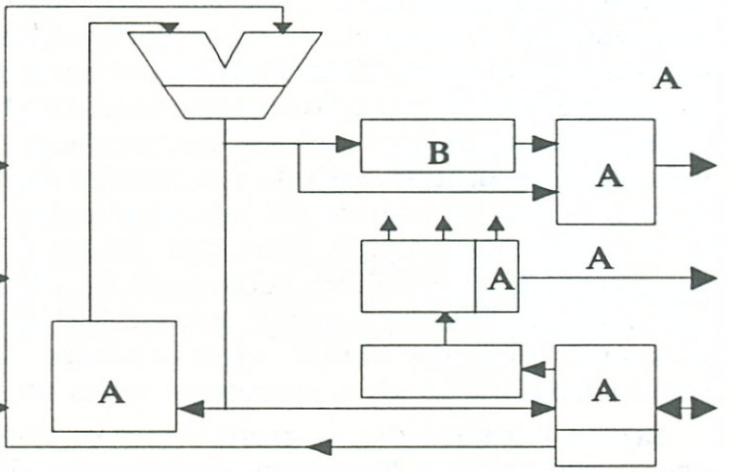
8.8. ábra

Az első utasítás felhozása.



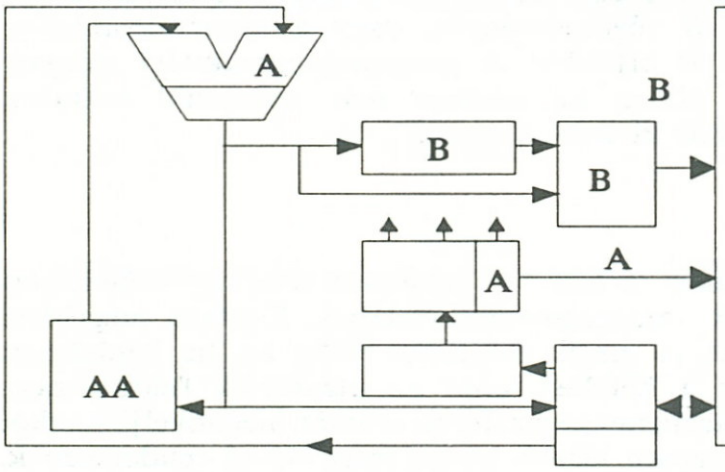
8.9. ábra

Az utasítás dekódolása, és az utasításszámláló inkrementálása. (B)



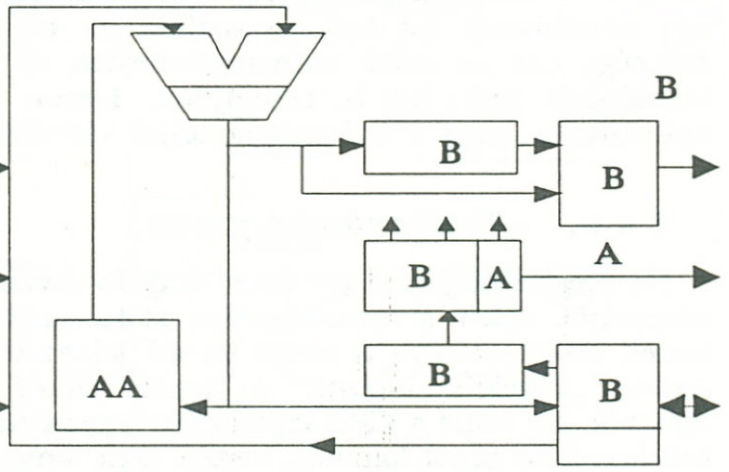
8.10. ábra

Az első utasítás operanduszának behozatala.



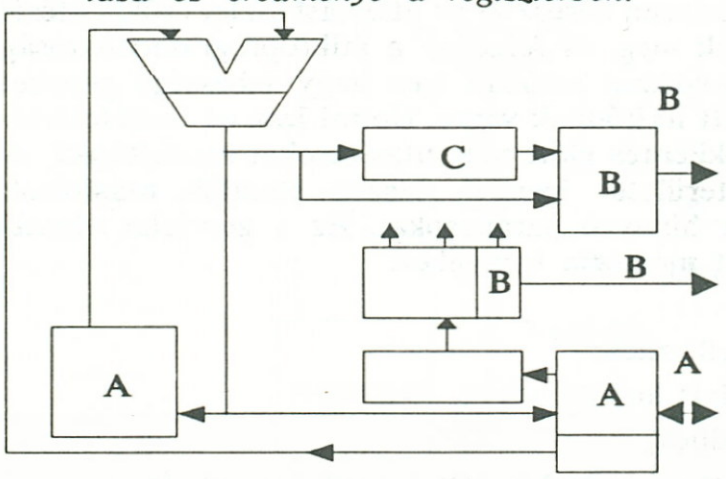
8.11. ábra

A B utasítás kiküldése, A végrehajtása és eredménye a regiszterben.



8.12 ábra

B utasítás behozása.



8.13. ábra

B utasítás dekódolása, A eredményének kiküldése a memóriába.

ha az A betű a címpufferben szerepel, akkor az első utasítás címét, ha az ALU-ban szerepel, akkor a neki megfelelő műveletet, illetve ha az adatpufferban találjuk meg, akkor az adat címére vonatkozó információt jelöli.

A három utasításból álló művelet végrehajtási módjából jól kitűnik a reteszek szerepe. Látható, hogy az A utasításnak megfelelő operanduszon való művelettel egyidőben már felhozatalra került a B utasítás, és megkezdődik ennek dekódolása. Mint kiderült, tehát akkor kezdődhet meg a B behívása, amikor az utasításlehívó egység felszabadul. Ez a példa egy kétszakaszoscsővezetékesrendszertmutatott be, egyidejűleg két utasítás bizonyos

részeivel foglalkozott a gép. Ma már léteznek ennél jóval fejlettebb kilencszakaszos csővezetékes rendszerek. is. Ha megfigyeljük a rendszer szerkezetét, akkor kiderül, hogy alapvetően két részt különíthetünk el benne, egy utasításlehívó és egy végrehajtó egységet. Az utasításlehívó egység feladata az utasítás behívása, a művelet dekódolása, az operandusz címszámítása, az operandusz beolvasása. A végrehajtó egység feladata pedig a kijelölt művelet végrehajtása. Mivel a végrehajtó egység sebessége általában kisebb, mint a lehívó egységé, ezért a végrehajtó egységre nézve várakozási sort kell beiktatnunk, amelyben az utasításlehívó egység által már kezelt adatok várakoznak a végrehajtásukig. Tovább gyorsíthatjuk a feldolgozás sebességét, ha a feldolgozó egységet további, egymással egyszerre használható részekre bontjuk (aritmetikai csatorna). Az egyes egységek az utasításkészlet diszjunkt részhalmazának végrehajtására képesek. (Természetesen az összes egység utasításhalmaza lefedi a processzor teljes utasításkészletét.)

Sajnos bármennyire is gazdaságosnak tűnik ez a megoldás, számos problémát vet fel, amely elsősorban a Neumanni architektúra örökségének a következménye. Nevezetesen ha a gép elágazásokhoz érkezik, akkor a cső tartalmától meg kell szabadulni és az elágazásnak megfelelően kell felszednie a további utasításokat. Újabb problémát jelent az utasítások egymásra hatása is. Előfordulhat ugyanis olyan eset, hogy két utasítás ugyanazt a funkcionális egységet igényli, szükségképpen a később jövőnek várakoznia kell mindaddig, amíg az előző le nem mondott annak használatáról. Hasonló problémát vet fel az is, ha egy utasításnak fel kell használnia az előző végeredményét, vagy a címmódosításhoz szüksége van az előző indexregiszterben tárolt értékére. A programmegszakítási igények kezeléséről már ne is beszéljünk, hiszen ebben az esetben már mindenki számára nyilvánvaló, hogy a csővezeték teljes tartalmát ki kell „öntenie".

8.1.6. RISC számítógépek

Ha meggondoljuk, hogy adott órajelferekvencia mellett mi határozza meg egy számítógép sebességét, akkor a memóriaciklus időtartamát meghatározónak találjuk. Egyfajta megoldást ennek csökkentésére a cache-tároló jelentett. A másik megoldás pedig az, ha képletesen szólva, „közelebb hozzuk" a tárolót ahhoz a helyhez, ahol az utasítások feldolgozása történik. Ha tehát a CPU lapkán lévő memóriaelemek,regiszterek számát megnöveljük,akkor hatékonyabb gépet kapunk, hiszen ezek lényegesen kisebb hozzáférési idővel rendelkeznek. Ez különösen akkor nagyon szembetűnő, ha külön írási- olvasási busszal látjuk el őket, megteremtve a többszörös hozzáférés lehetőségét.

A másik megoldási út az, hogy egyszerűsítjük a vezérlő logikát, hiszen a kapuk száma alapvetően meghatározza a jel áthaladási idejét. A vezérlő logika csökkentésének igénye azonnal maga után vonja a mikroprogramok és ezen keresztül az utasításkészlet csökkentését is. Csak a legalapvetőbb utasításokat hagyjuk meg, és feladjuk a mikroprogramozhatóság elvét is a gyorsítás érdekében. Ezzel a megoldással valóban igen nagy sebességű gépeket kapunk. Az utasítások egyetlen óraciklus alatt hajtódnak végre. De mi lesz az összetettebb feladatok végrehajtásával, amelyekhez a csökkentés előtt még utasításokat rendeltünk? A feladat megoldása nyilván több memóriaterületet igényel, hiszen elemibb utasítások sorozatával, szoftver úton kell pótolnunk a hiányzó parancsokat. Ez a gondolat visszavezet azonnal a processzorlapkán elhelyezett memória igényéhez.

Azokat a processzorokat, amelyek:

- » csökkentett utasításkészlettel rendelkeznek,
- » utasításaikat egyetlen óraciklus alatt hajtják végre,
- » huzalozott vezérlőegységgel működnek,
- » jelentős méretű, lapkán elhelyezett regiszterkészlettel rendelkeznek és
- » szoftver biztosítja a hiányzó, bonyolultabb utasításokat,

RISC (Reduced instruction set computer — redukált utasításkészletű) processzoroknak nevezzük.

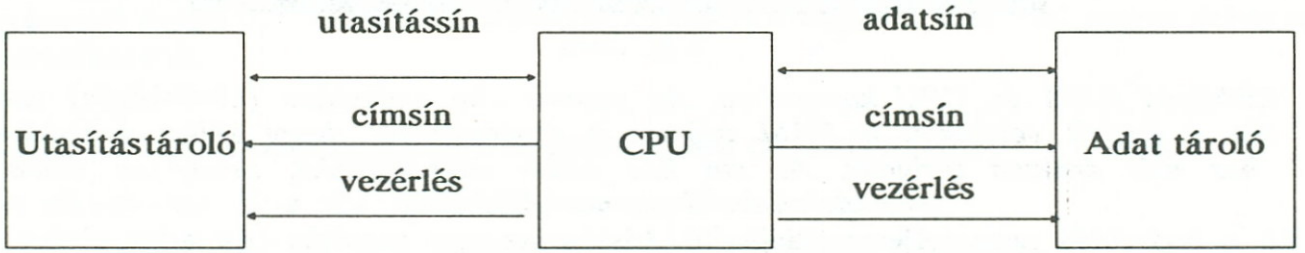
Mivel az utasítások egyetlen óraciklus alatt mennek végbe, jól ütemezhető csővezetékes (pipelin) elvű processzorok készíthetők belőlük. (8.14. ábra)

Az utasítások végrehajtása során a processzor először behozza az első utasítást, ezt követően a CPU dekódolja azt és kiküldi az adatcímet. Az adatmegszerzése után elvégzi a kijelölt utasítást, az eredményt tárolja a regisztereiben és ezzel egyidőben kiküldi a következő utasítás címét. Az eredmény kivitele után a következő utasításbehozása és dekódolása kezdődik, majd a folyamat innen ismétlődik.

A gyorsítás mellett további előnyei is vannak a RISC processzornak. Ugyanis a processzorlapkán felszabaduló helyre (mivel egyszerűbb a vezérlő és végrehajtó egység), más fontos kiegészítő áramköröket helyezhetünk el. (Lebegőpontos aritmetikai- és memóriakezelő egységek stb.)

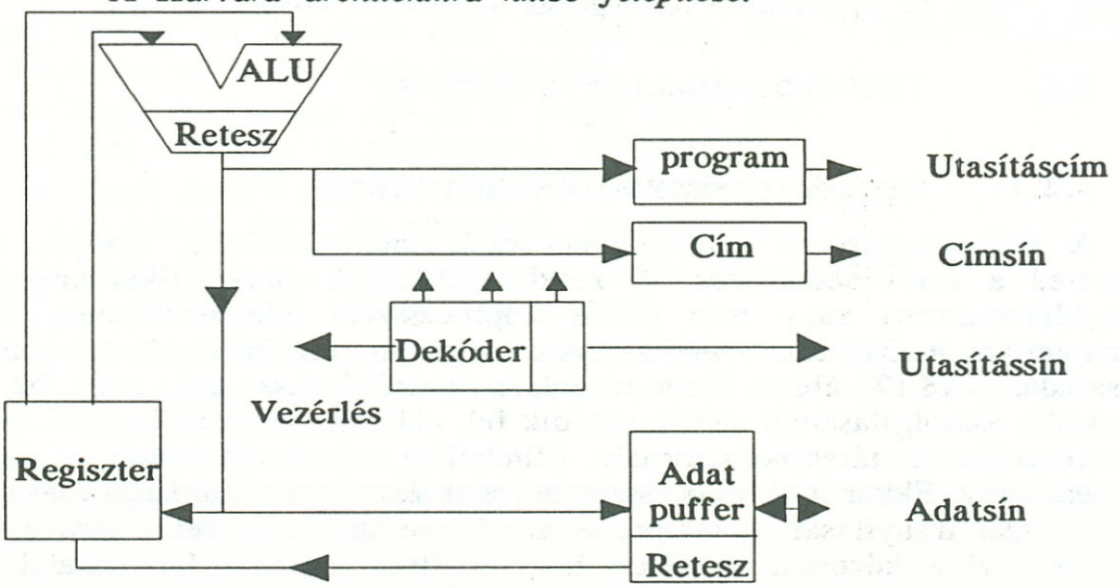
8.1.7. A Harvard architektúra

Az előbb említett eljárások lényegében a Neumann-elvre épültek, ezért nem tudták megkerülni annak korlátait. Az utasítások és a hozzájuk tartozó adatok szekvenciális kezelése egyáltalán nem természetes, semmi sem kényszerít bennünket erre. Ugyanis az utasítással együtt az adatok is felhozhatók a processzorba egyetlen ütemben, ha megváltoztatjuk annak a felépítését. Így megspórolunk egy tárhozfordulást.



8.14. ábra

A Harvard architektúra külső felépítése.

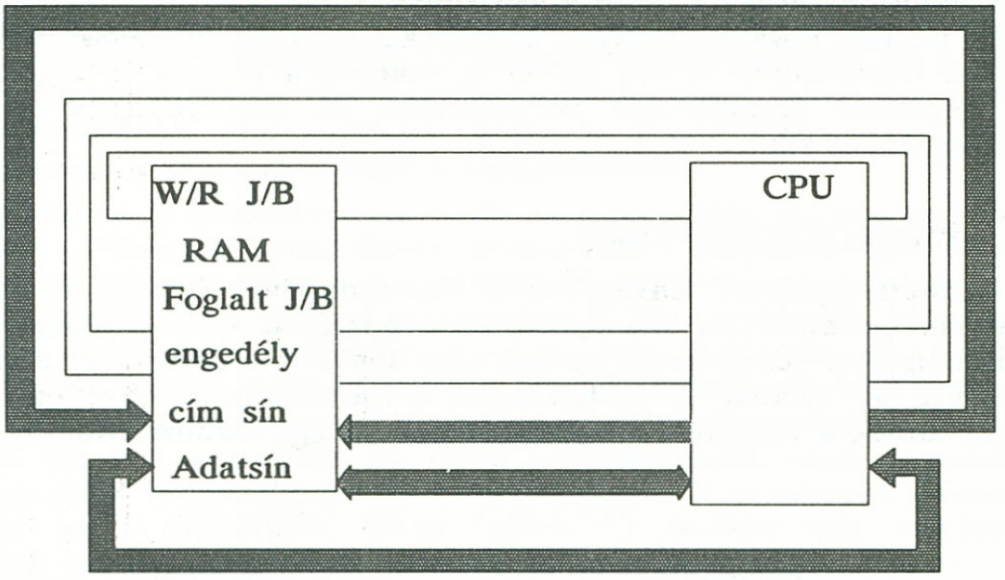


8.15. ábra

A Harvard architektúra szerkezete. Az utasítássín különválik.

A megvalósításért persze árat kell fizetni, dupla sínrendszert kell készíteni, egyet az utasítások, egyet pedig az adatok kezeléséhez.

A tárat nem kell feltétlenül fizikailag is kettészakítani. Léteznek **kétkapus memóriák**, amelyeket ilyen célra lehet felhasználni. Alkalmazásuk azzal az előnnyel jár, hogy az alkalmazói program önmagát is módosíthatja. (A Neumann-elvű gépek esetén ez mindig megoldható.)



8.16 ábra

Kétkapus RAM és CPU kapcsolata. Az azonos cím megadása (címütközés) esetén a foglalt vezetéken a RAM tudatja a processzorral, hogy ehhez a területhez már érkezett tárigény. Az sín kap előbb engedélyezést, amelyiken előbb alakultak ki azonos feltételek.

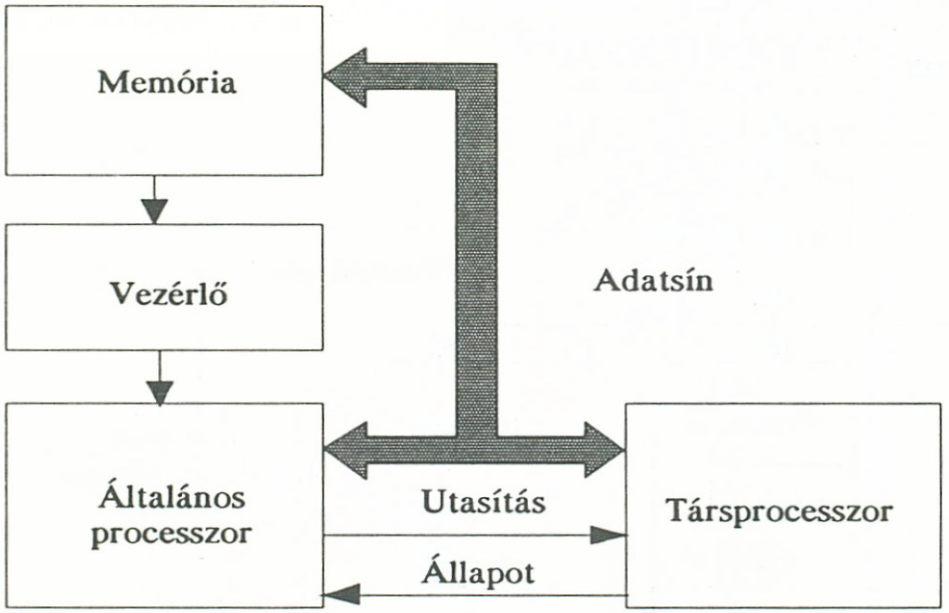
Ez a megoldás sem teljesen kielégítő, hiszen az egy utasítás egy adat elvére épül, hasonlóan a korábban ismertetett rendszerekhez (SISD Single instruction stream, single data stream — egy utasítás folyam és egy adatfolyam). Ennek velejárója, hogy egyetlen aktív áramköri elemet tartalmaz — egy központi processzort.

A tovább vezető út inenn már világosan látszik. Több processzort kell felhasználnunk ahhoz, hogy az információ feldolgozás sebességét növelni tudjuk.

8.2. Többprocesszoros rendszerek

8.2.1. Segédprocesszorok alkalmazása

A többprocesszoros rendszerekkel való ismerkedésünket kezdjük egy igen speciális esette,l a **társprocesszoros rendszerekkel**. A matematikai műveletek gyorsítására, segédprocesszort vagy más néven koprocesszort alkalmazhatunk. Ezek lebegőpontos aritmetikai műveletek végrehajtására orientált eszközök. A központi egység sínjére csatlakoznak(8.17. ábra) Ez a megoldás lehetővé teszi azt, hogy ha egy matematikai művelet kiszolgálásának igénye vetődik fel, akkor a CPU felfüggeszti működését és átadja a vezérlést a társprocesszornak. Mindaddig várakozik, amíg a művelet eredménye megszületik. Ekkor a társprocesszor művelet kész jelzést ad. Ettől fogva a CPU visszaveszi a rendszer irányítását. A társprocesszor közvetlenül nem fér a tárhoz. Ha tárigénye van, akkor kéri a központi egységet, hogy segítsen az adat felhozatalában. (Vagy a CPU hozza fel az adatot vagy a kiszámított címet adja át a koprocesszornak.) A rendszer annyival gyorsabb a hagyományos egyprocesszoros rendszereknél, amennyivel a



8.17. ábra

Társprocesszor rendszerbe illesztése. Figyeljük meg,hogy a memória a központi egység alá van rendelve.

segédprocesszor gyorsabban hajtja végre a lebegőpontos utasításokat, mintha azt a CPU szoftveresen tette volna. A központi egység csak egész számokkal számol, minden műveletet erre vezet vissza. Ugyanakkor a társprocesszor utasításkészletében magasabb rendű függvények is vannak (Pl. szögfüggvények. A 80237-es koprocesszor 68 numerikus függvénnyel dolgozik.), ezért az alkalmazói szoftvertől függően akár 100-szoros sebességgel is számolhatunk.

8.2.2. A többprocesszoros rendszerek osztályozása

Az alkalmazás célja szerint a többprocesszoros rendszereket

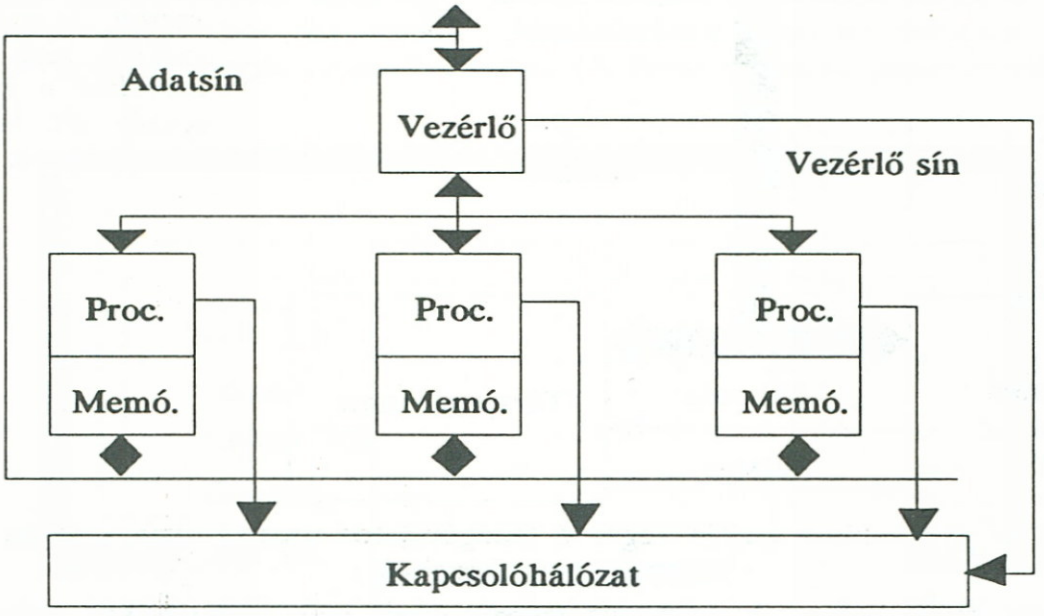
- » általános célú — a működési sebesség növelése érdekében,
- » redundáns — a megbízhatóság növelése érdekében,
- » számítógéphálózatok — a nagy távolságban elhelyezett elosztott információ feldolgozása érdekében alakítják ki.

A továbbiakban először az általános célú rendszereket vesszük kissé közelebbről szemügyre. Ezek a rendszerek tovább csoportosíthatók.

- » SISD — Single Instruction stream on single Data stream,
  (hagyományos egyprocesszoros rendszerek)
- » SIMD — Single Instruction stream on Multiple Data stream,
  (vektor, tömb, és asszociatív processzorok)
- » MISD — Multiple Instruction stream on Single DAta stream,
  (tulajdonság válogató rendszerek)
- » MIMD — Multiple Instruction stream on Multiple Data stream.
  — Lazán csatolt rendszerek. A processzorok közötti kapcsolat üzenetorientált.
  — Szorosan csatolt rendszerek, amelyeknél a processzorok kapcsolata valamilyen közös erőforrás felhasználásával van megvalósítva.(pl. memoria, I/O, stb.)

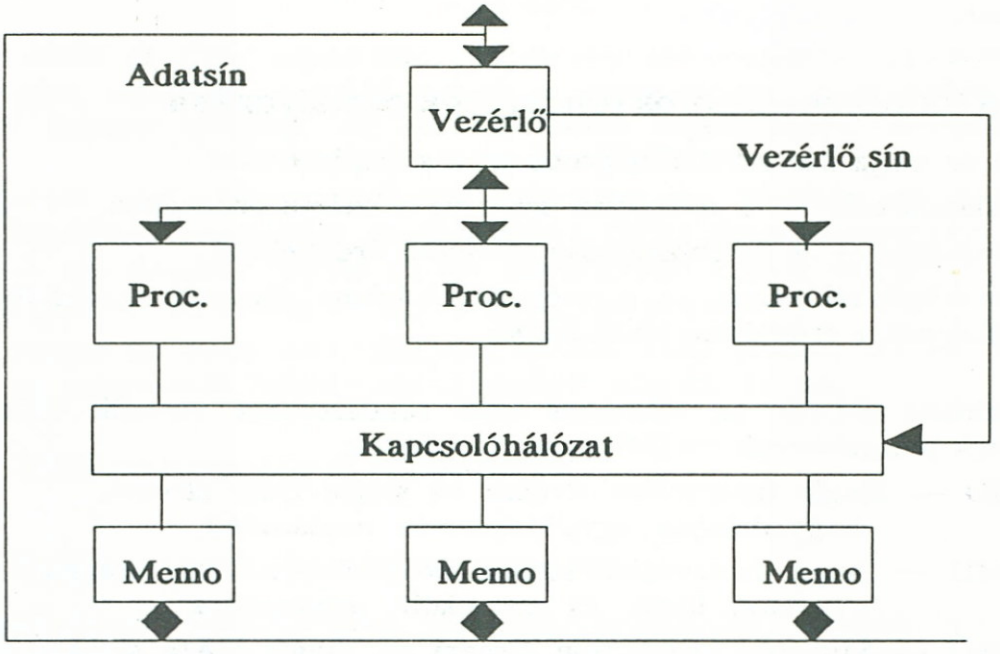
Lényegében az utolsó három csoport sorolható a szorosabb értelemben vett többprocesszoros rendszerekhez. Vegyük őket sorra.

8.2.3. SIMD rendszerek



8.18. ábra

Önálló memóriaelemekkel rendelkező vektorprocesszorok.

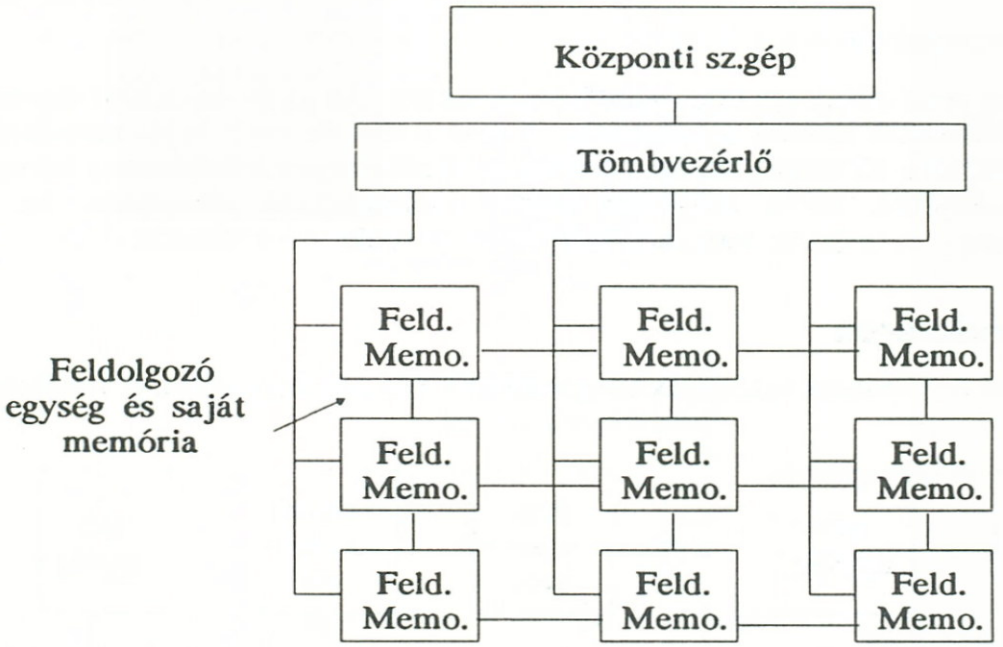


8.19. ábra

Osztott memóriával rendelkező vektorprocesszorok.

Ezt a rendszert elsősorban akkor érdemes használni, ha adatainkon ugyanazokat a műveleteket kell alkalmazni. Úgy is fogalmazhatnánk, hogy a műveleteket vektorokon hajtjuk végre. Ez egyprocesszoros esetben nyilván azt jelenti, hogy annyiszor történik meg az utasítás lehívása , dekódolása, ahány adaton azt végre kell hajtani. Nyilvánvaló, hogy lényegesen gyorsulhat a művelet végzése, ha minden egyes vektorkomponenshez egy processzort rendelünk (8.18.—19. ábra), amelyek ugyanazt az utasítást végzik el. (**Vektorprocesszor** ok)A rendszer hatékonysága akkor nő meg kedvezően, ha ezek a vektorok sok komponensből tevődnek össze, és a feldolgozásra váró vektorok száma nagy. Ebben az esetben adatpipeline-t is alkalmazhatunk, amely tovább növeli a rendszer

hatékonyságát. A művelet, utasítás megváltoztatása természetesen az „adatcső" kényszerű



8.20. ábra

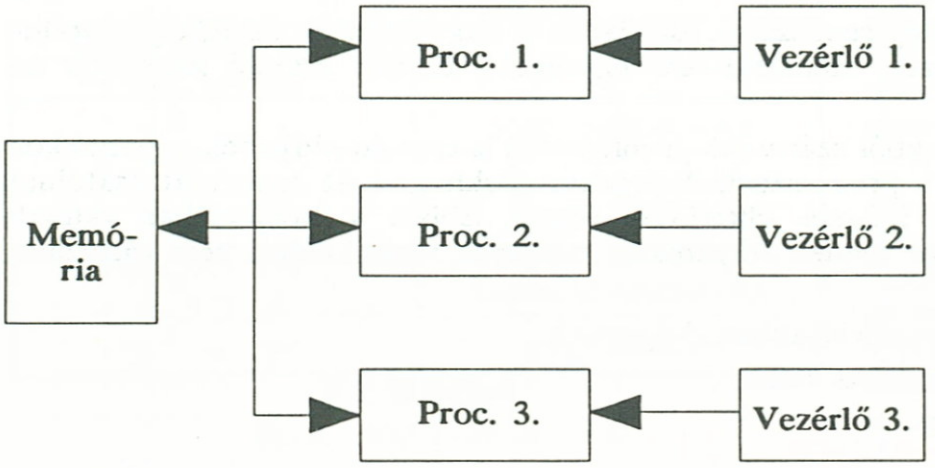
Tömbprocesszorok

kiűrítéséhez vezet.

A rendszer még tovább javítható, ha a vektorprocesszorokat párhuzamosan kapcsoljuk. A párhuzamosan kapcsolt egységek ugyanazt az utasítást, de különböző vektorokon hajtják végre (**tömbporcesszor**). A tömbvezérlő egység hívja le a végrehajtásra kerülő utasítást, dekódolja, majd közli a processzorokkal. (8.20. ábra)

Az adatokat a processzorok a hozzájuk tartozó regiszterekből veszik elő. A rendszer működése még hatékonyabbá tehető, ha a szomszédos feldolgozó egységeket egymással csatlakoztatjuk. (Ilyen rendszereket például a mesterséges intelligencia kutatásában alkalmaznak.)

A SIMD típusú rendszerek egy érdekes változatát az **aszociatív processzor**ok jelentik. Ezek a tároló típusában különböznek az előzőekben bemutatott megoldásoktól. Ugyanis itt úgynevezett asszociatív tárolókat alklamazunk. (Ezek lényegében olyan eszközök, amelyek egy tárolóelemet és egy összehasonlító egységet tartalmaznak. A hozzájuk tartozó vezérlő áramkörök feladata, hogy feloldják a többszörös hozzáférésekből adódó problémákat, valamint irányítja a adatok elhelyezését a tárban. A rendszer nagy előnye, hogy a tároló egészén történik az adatok keresése, ezért nagyon gyors.)



8.21. ábra

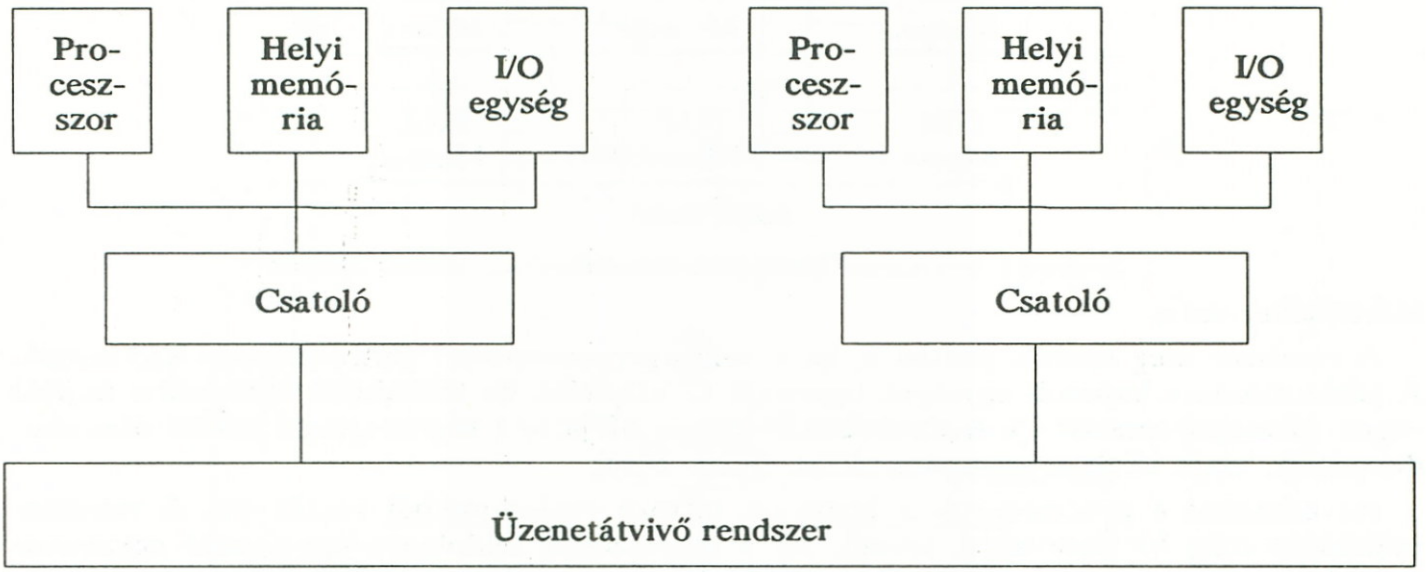
A többszörös vezérlést megvalósító rendszerek.

8.2.4. MISD rendszerek

A többszörös vezérlést realizálják egyetlen adatfolyam esetén. Az aktív áramköri elemeket a vezérlők különböző utasítások szerint működtetik. A processzorok vagy saját memóriából kapják az „n" példányban eltárolt utasítást, vagy kommunikációs kapcsolóhálózaton keresztül a rendszer memóriatömbjéből. Ez a megoldás akkor alkalmazható előnyösen, ha azt szeretnénk megtudni, hogy az adatok halamza milyen osztályokba sorolható.

8.2.5. MIMD rendszerek

A többszörös vezérlési folyamatot valósítják meg a többszörös adatfolyamon. A rendszerek



8.22. ábra

Lazán csatolt többprocesszoros rendszerek.

feladat, eljárás, vagy szubrutinszerű párhuzamosítást valósítanak meg. A processzorok kommunikálnak egymással, adatokat cserélnek és szinkronizációt végeznek.

A rendszer összehangolt tevékenységét speciális operációs rendszer végzi. Feladata, hogy elossza a működés során felmerülő feladatokat az erőforrások között és biztosítsa a felhasználó számára a csatlakozófelületet oly módon, hogy a felhasználó elől eltakarja a párhuzamos processzorok működését.

A **lazán csatolt** számítógép rendszerek alapjában a hagyományos számítógépmodulokból tevődnek össze (processzor, memória, I/O egységek). Ezeket csatoló kapcsolja az üzenetátvivő rendszerekhez.

A többprocesszoros rendszerekből származó problémákat a csatoló oldja fel, ugyanakkor a belső tárolója tehermentesíti a processzort. A processzorokhoz saját memóriát csatolnak a nagymértékű buszhozzáférési igények elkerülése végett. Ebben a memóriában vannak elhelyezve az adott processzorhoz rendelt folyamatok, amelyek futás közben nem változnak meg.A rendszer előnyei, hogy:

- » futás közben saját memóriájukban dolgoznak,
- » a közös busz felhasználása ritka,
- » a rendszer jól tagolt,

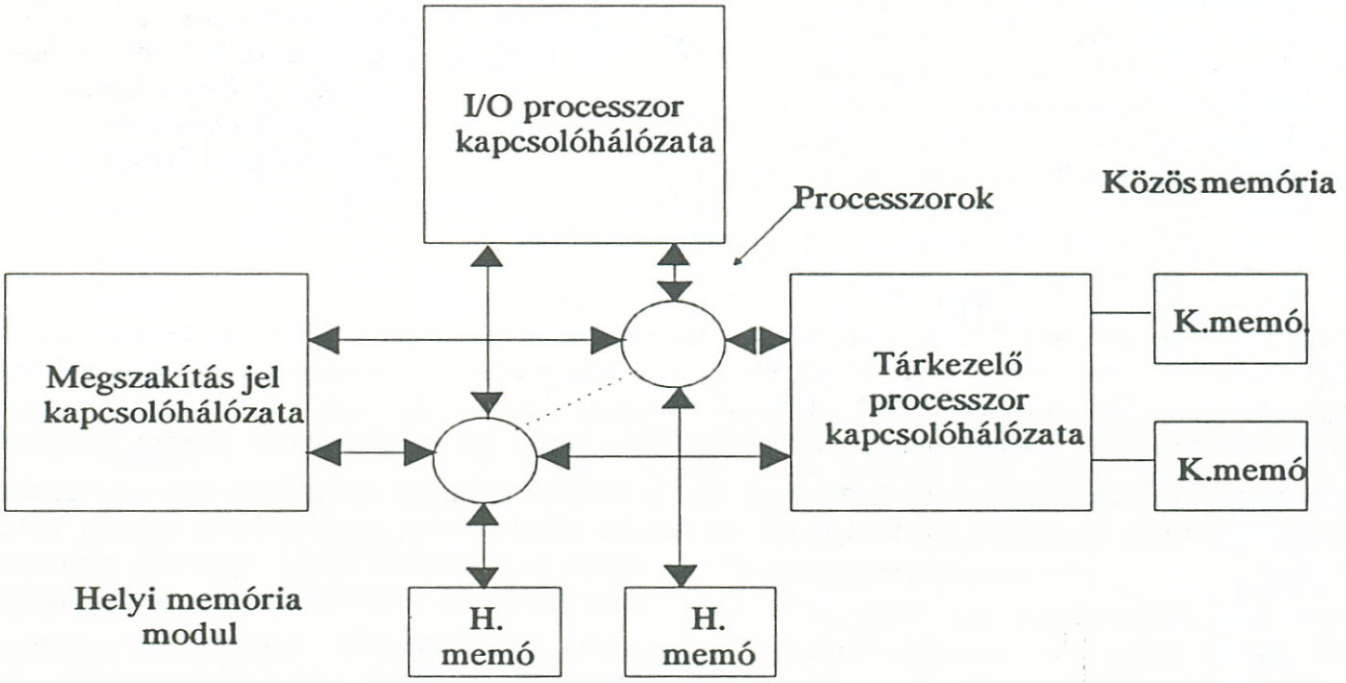
» külön—különegyszerűek.

Hátrányai hogy:

» nehéz megvalósítani a feladatok olyan kiosztását, hogy egyenletesen terhelje a processzorokat,

» a rendszer bővítése felborítja a kiegyensúlyozott terhelési viszonyokat,

» a rendszer egyensúlyát a fellépő hibák könnyen felboríthatják. (Hiba esetén az adott processzorhoz rendelt feladat — task — kiesik, ezért a rendszer nem fejezheti be sikeresen a működését.)



8.23. ábra

Szorosan csatolt többprocesszoros rendszerek.

A **szorosan csatolt** rendszerek a gyakori kommunikáció igényéből fakadnak. A rendszer felépítése olyan, hogy bármely processzor bármely memóriával kapcsolatba léphet. Ez a megoldás feloldja a hagyományos számítógépeknek azon korlátját, hogy a tár legnagyobb része a feldolgozás egész ideje alatt tétlenül „csücsül". (Gondoljuk meg, a számítógép szilíciumfelületének közel 90 %- át a memória teszi ki!)

A rendszer a szokásos modulokon kívül (processzor, memória, I/O egység) kiegészül a kapcsolóhálózattal, amelyen keresztül a processzorok kommunikálnak. A memóriához való többszörös hozzáférés a rendszer hatékonyságát rontja, hiszen meg kell várniuk amíg a kapcsolóhálózat engedélyezi a felhasználást. Ezt kikerülendő a processzorokhoz kisebb méretű memóriát rendelnek (**elosztott memória**), amelyekben az operációs rendszer magját és a szükséges táblázatokat helyezik el.

A Boole kocka csúcsainak száma	A kocka elnevezése
$2^0 = 1$	pont, nulláskocka
$2^1 = 2$	szakasz, egyeskocka
$2^2 = 4$	négyzet, ketteskocka
$2^3 = 8$	kocka,hármaskocka
$2^4 = 16$	négyeskocka

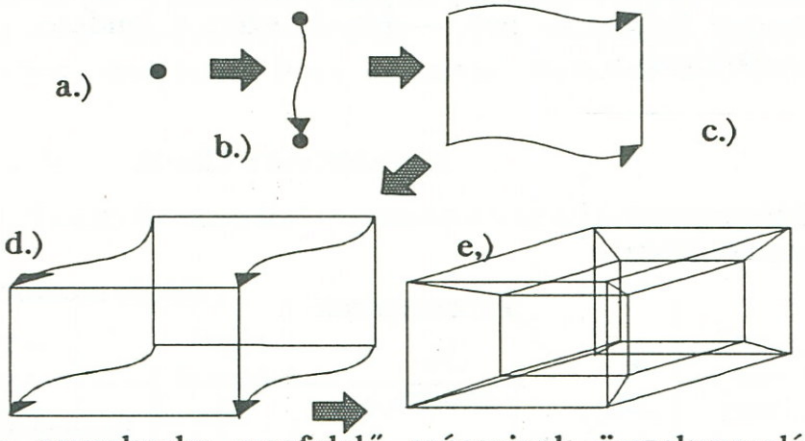
8.2. táblázat

A Boole féle n-es kockák.

Ha a rendszer nagyon nagy számú processzort használ (pl. Connection Machine 65536 darab processzor), akkor nagy gond a rendszerek összekapcsolása, sőt fizikailag a pont-pont összeköttetés nem megvalósítható. (Az előbb említett példában szereplő processzorszám mel-

lett körülbelül 2 milliárd összeköttetést kellene elkészíteni a processzorok pont-pontkapcsolata miatt. Ha egy összeköttetést, mondjuk egy másodperc alatt tudunk kialakítani, akkor e szám megvalósítása több mint egy emberöltőig tartana.)

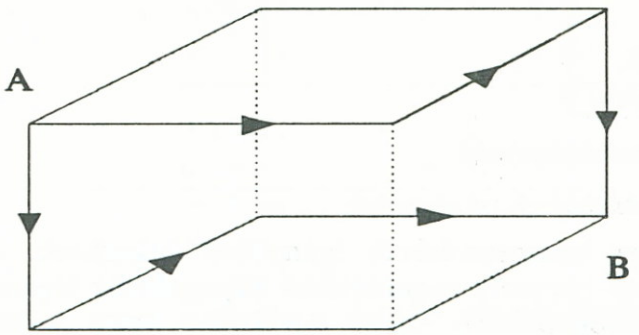
A megoldást a Boole féle n-es kocka jelenti. (8.2. táblázat) A közönséges kockát ebben a rendszerben a különféle dimenzióknak megfelelő kockák sorozatának tagjaként értelmezzük. A pont jelképezi a nullás kockát, az egységnyi hosszúságú szakasz az egyeskockát. Két



8.24. ábra

Boole féle n-es kockák néhány esete. a.) nullás-kocka, b.) egyes-kocka, c.) kettes-kocka, d.) hármas-kocka, e.) négyes kocka.

ilyen egyeskocka megfelelő csúcsainak összekapcsolása hozza létre a ketteskockát, a hagyományos értelemben vett négyzetet. Két ketteskocka összekötése a hármaskockát, hagyományos értelemben vett kockát hozza létre, és így tovább a nagyobb dimenziószámokig. Látható az 8.1. táblázatból, hogy az n dimenziós kocka csúcsainak számát a 2^n összefüggés határozza meg. Ha a kockák minden csúcsában egy-egy processzort képzelünk el, akkor például a 12 -es kocka alkalmazása esetén 4096 egység köthető össze oly módon, hogy egyetlen processzor sem kerül 12 huzalnál távolabb egymástól.



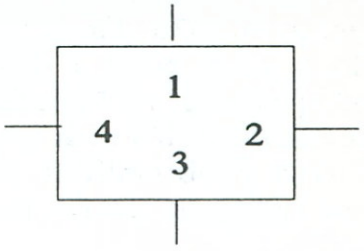
8.25. ábra

Alternatív utak a Boole-féle hár-

A processzorok azonosítása sem túl bonyolult feladat, ha észrevesszük, hogy minden kocka két alkockát tartalmaz. Ezeket rendre 0, 1 számokkal azonosítjuk. Ilyen módon egy 12 bites jelsorozat rendelhető minden egyes processzorhoz, amely megmondja a processzor helyét a rendszerben. Ugyanis a sorozat első bitje megmutatja, hogy a kocka melyik 11-es kockában van, a második bitje meghatározza, hogy melyik 10-es alkockában van és így tovább. Ezért tehát az üzenet általában 12 lépés után megtalálja a gazdáját.A rendszerben párhuzamos átvitel is könnyen megvalósítható, ugyanis az információ több alternatív úton is továbbítható. Ha tehát az egyik utat valamelyik átküldött információ lefoglalta, akkor a másikon még átküldhető a címzettnek.

8.2.6. A transputerek

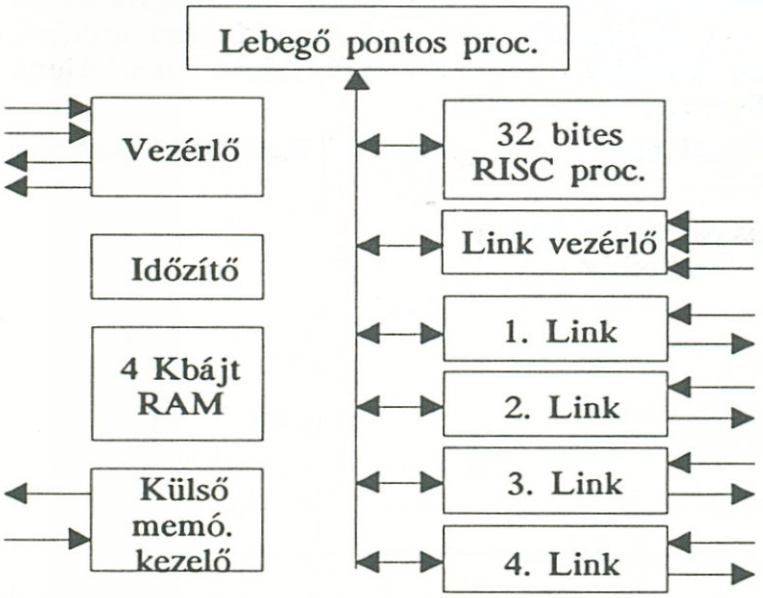
A párhuzamos működési elvű számítógépek egy lehetséges építőeleme, amely nem más, mint egy nagy teljesítményű **RISC** processzor, memóriákkal és I/O egységgel kiegészítve. Általában technikai okokból 4 fizikai csatornát tartalmaz, azaz környezetében négy szomszéddal képes közvetlenül kommunikálni. A hardver fejlesztés érdekes eleme, hiszen eddig a hardver határozta



8.26. ábra

A transputer kommunikációs csatornái.

meg a szoftverek fejlődését. Most viszont a konkurrens folyamatok végrehajtására kifejlesztett OCCAM nyelv igényeinek megfelelően alakították ki ezt a csippet. (Konkurens folyamatok alatt azokat a folyamatokat értjük, amelyek közös erőforrásokért versenyeznek, ezeket egymást kizáró módon használják.) Mivel a transputer egy-egy folyamat végrehajtásáért felel, az eszköznek tartalmaznia kell egy hatékony feldolgozó egységet (RISC processzor), gyors kommunikációs egységet a környezetével való kapcsolattartás miatt., gyors elérésű tárat, ami a folyamatokhoz tartozó utasításokat és adatokat tárolja, illetve hatékony ütemezőt. Ezeken a fő egységeken kívül szükséges egy külső memóriakezelő áramkör, valamint a lebegőpontos műveletek végrehajtása miatt egy lebegőpontos aritmatikai egység. A VLSI technológia lehetővé teszi, hogy egyetlen lapkára integráljuk ezeket az egységeket és 32 bites belső buszrendszerrel kössük össze őket.



8.27. ábra

A transputer belső felépítése

A processzornak már több változata is elkészült, például a T400-as 32 bites belső buszrendszert tartalmaz és 2 Kbájtos belső memóriát, a T800 annyiban különbözik ettől, hogy 4 kbájtra növelték a belső memória kapacitását. T9000 transputerben már 16 Kbájtos táratalkalmaznak.

A CPU hat regiszterből álló regisztertömböt tartalmaz. A felső három hardver stack-et biztosít az egész szám és a címaritmetikához. Az alsó három rendre a munkaterületre és a következő utasítás címére mutató pointert, illetve az aktuális operandust tartalmazza. A segédprocesszor is tartalmaz egy háromelemű regisztertömböt. A regisztertömböket csak veremműveletekkel lehet elérni, kivétel a processzor alsó három regisztere.

A lebegőpontos processzor regiszterei két példányban állnak rendelkezésre, hogyha egy magasabb prioritású folyamat megkapja a vezérlést, ne kelljen a regisztereit a memóriába menteni.A rendszerben lévő folyamatok alapvetően aktívak (végrehajtás alatt, vagy várakozók listáján vannak) vagy inaktívak (inputra, outputra várók, vagy késletetettek) lehetnek. A rendszer ütemezőjének felépítése miatt az inaktív folyamatok nem fogyasztanak processzoridőt. Az aktív folyamatokat az ütemező két regisztere segitségével listába fűzi. (A regiszterek az első és utolsó folyamat kijelölésére szolgálnak.)

A folyamatok addig használják a processzort, ameddig működni képesek, ezek után felkerülnek a várakozók listájára. Ezt követően a következő listában tárolt folyamat kapja meg a vezérlést. A kommunikáció csatornákon keresztül történik, szinkronizált és pufferelés nélküli. Külső csatornán történő üzenettovábbítás igénye esetén a processzor átadja a vezérlést a „Link" interface-nak , miközben leveszi a folyamatot az aktív listáról. A Link három regisztert tartalmaz, amelyek mutatókat tárolnak ,egyet a folyamat munkaterületére, egyet az üzenetre. A harmadik regiszter az üzenet méretének számlálására szolgál.

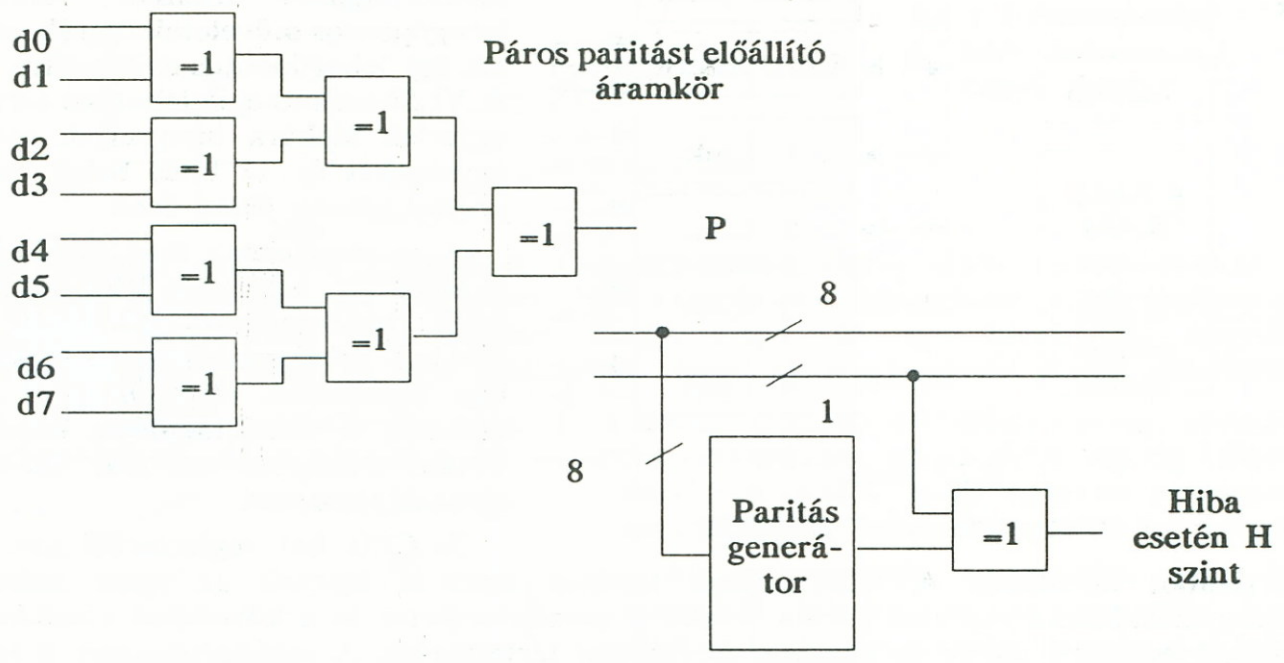
Ha a címzett és a küldő is kész, akkor létrejön a kommunikáció. Ez idő alatt a következő aktív folyamat fut, hiszen a jelenlegi (inputra/outputra) várakozik. Az üzenet végén ismét a régi folyamatot aktivizáljuk. A **transputer** Linkek között két egyirányú jelvezeték biztosít összeköttetést. Ezek szekvenciális módon kommunikálnak egymással. Az adatátviteli protokoll előírja, hogy minden bájt átvitelét nyugtázni kell, amely egy start illetve egy stop bit küldéséből áll. Az átviteli sebesség 0,8 Mbájt/sec -től 1 Gbájt/s-ig

terjed (T9000 transputer), attól függően, hogy az átvitel és nyugtázás egymással átlapolva történik-e.

8.2.7. Redundáns rendszerek

A nagy megbízhatóság és folyamatos működtetés igénye vetette fel a **redundáns rendszer** ek kialakítását. A számítógépekben fellépő hibák elleni védekezés alapja a minél gyorsabb észlelés. Ennek megvalósítására a rendszerekbe ellenőrző áramköröket építünk, amelyek megfelelő módon és adott időben tesztelik a rendszer működését. Gondoljunk csak a mindenki által közismert paritás ellenőrző áramkörökre.

Ha az átvitel során az áramkör hibát észlel akkor, a megszakítási rendszeren kersztül



8.28. ábra

Paritás ellenörző áramkör megvalósítása kizáró vagy kapukkal.

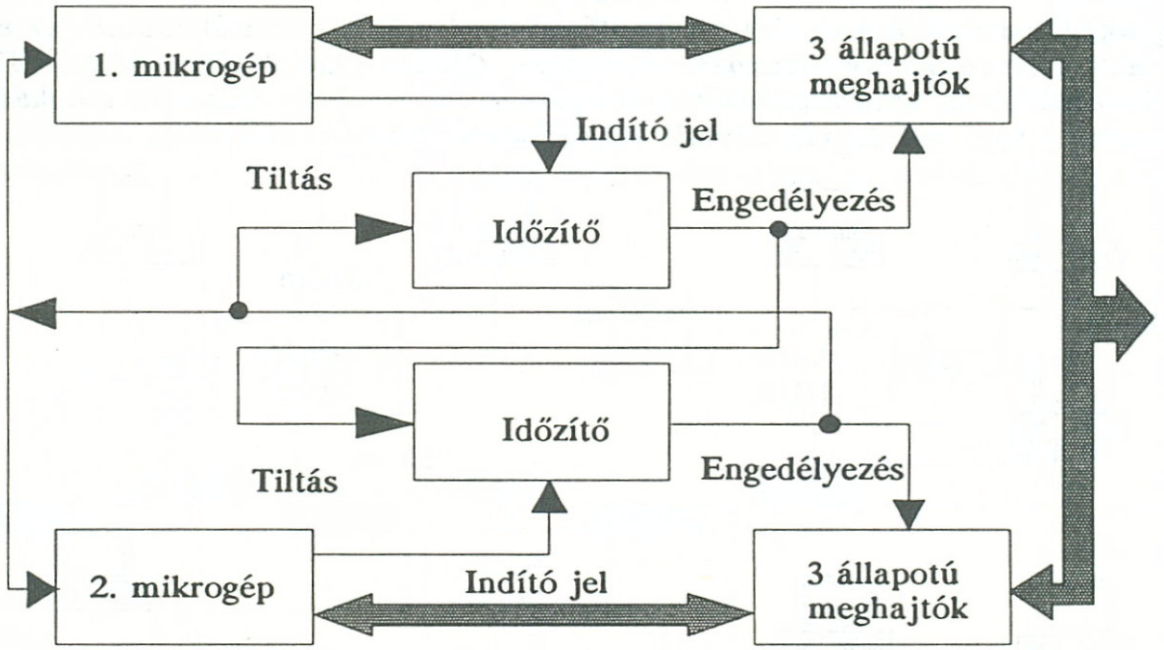
olyan rutinra kényszeríti a processzort, amely a hiba kijavítását vagy megismétlését kezdeményezi.

A hibaellenörzés magasabb szintjén a diagnosztizáló áramkör olyan folyamatot indít el, amely megakadályozza a rendszer elemeinek további károsodását, illetve bekapcsolja a tartalék áramköröket azért, hogy az üzemeltetés folyamatos legyen.

Az önellenőrzésnek vannak természetesen korlátai, a legfontosabb talán az, hogy az áramkörök valamely részének biztosan jól kell működnie ahhoz, hogy a hiba felderíthető legyen (diagnosztizáló rendszer). Ugyanakkor fizikai és gazdaságossági okokból a rendszer a teljes ellenőrzést nem tudja megvalósítani. De arról a tényről sem feledkezzünk meg, hogy a diagnosztizáló program lefutásához is időre van szükség, ami alatt az ellenörzött rendszer állapotában káros változások állhatnak be.

A sínszervezésű rendszerek esetén a sín vagy az arra csatlakozó áramkör hibája az egész rendszer működését felboríthatja. A hiba tényének felismerésre legegyszerűbben időzítő áramköröket használhatunk. Enek a lényege az, hogy a processzor a bekapcsolás időpontjában egy számlálót indít. A program futását a számláló újraindítását kezdeményező kiviteli utasítások szakítják meg. Amennyiben a rendszer valamilyen ok miatt meghibásodik, úgy az újraindító utasítás nem jut el időben a számlálóhoz, így az az időzítés lejártakor hibajelet fog szolgáltatni. Esetleg bekapcsolja a tartalék rendszert. Ennek az elvnek az egyszerű megvalósítását mutatja a 8.29. ábra.

A bemutatott megoldásnál normál esetben az első gép időzítőjének kimenőjele az 1-es



8.29. ábra

Időzítéses hibavédelem.

háromállapotú kapukat nyitva tartja, az átvitel ezen keresztül zajlik. Ugyanakkor a 2-es számlálón keresztül lezárja a 2-es rendszer háromállapotú kapuit, aminek következtében az nem fér a külső buszhoz.

Ha a számlálót újraindító jel nem érkezik az időzítés lejárta előtt (hiba áll fenn a rendszerben), akkor az egyes rendszer háromállapotú kapui az időzítő vezérlő jelének megváltozása következtében lezárnak, ezzel elvágják az 1-es rendszer külső buszhoz való hozzáférési esélyeit. Ennek hatására a hiba nem kerülhet ki a külső buszrendszerbe. Az 1-es háromállapotú kapuk zárásával egyidőben nyitja a 2-es kapukat, ami miatt most a 2. rendszer fog a külső buszrendszerhez férni.

Mindezek mellett megszakításjelet is generál, ami elindíthat egy alkalmas hibajavító programot. Sajnos ez a megoldás csak látszólag kielégítő, ugyanis semmi biztosat nem tudunk arról, hogy a 2-es rendszer hogyan működött az átkapcsolás előtt. Annak a kérdésnek a megoldása sem könnyű feladat, hogy hogyan lehet leválasztani a meghibásodott rendszert ebből a szerkezetből és hogyan lehet megjavítani azt.

Ezen problémák megoldására számos megoldás született, de nem célunk ezek részletes tárgyalása.

8.2.8. Számítógéphálózatok

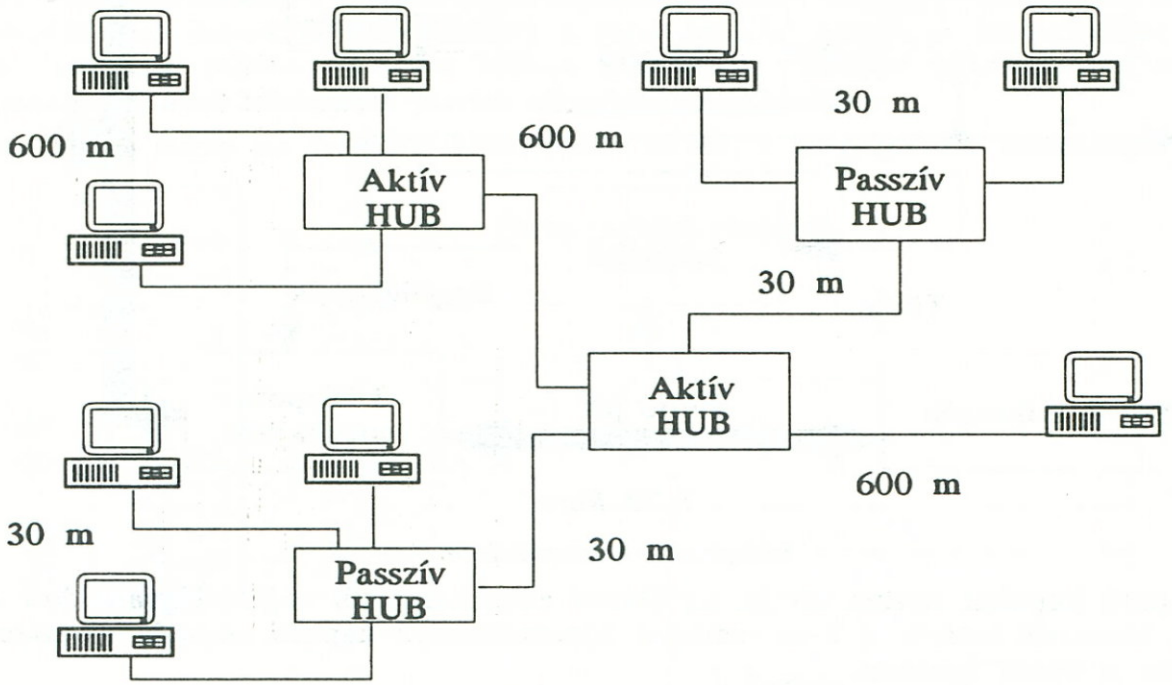
A nagyon lazán csatolt többprocesszoros rendszerekhez sorolhatók. Az összeköttetések ellenére ez a topológia is az egy utasítás egy adatfolyam elvére épül (SISD), de lehetőséget teremt több felhasználó különböző programjának konkurrens futtatására. Ilyen szempontból ez teljesítménynövekedést jelent, de egyetlen felhasználó programjának időigényét ez nem csökkenti. A következőkben néhány hálózati topológiát mutatunk be.

Az ARCNET hálózat

Alapelvét tekintve a munkaállomásokat logikailag egy gyűrűben helyezik el, amelyen egy vezérlő jelet cirkuláltatnak. Az adást csak az a gép kezdeményezheti, amelyik ezzel a vezérlőjellel rendelkezik. A vezérlőjelet az állomások fizikai azonosítóinak megfelelően küldjük ki. Ennek vételét követően a számítógép a sínre helyezheti az üzenetét. A

rendszer működését illesztőkártya biztosítja, amelynek a feladata a kommunikáció megvalósítása. (Ezek a kártyák topológiafüggők!)

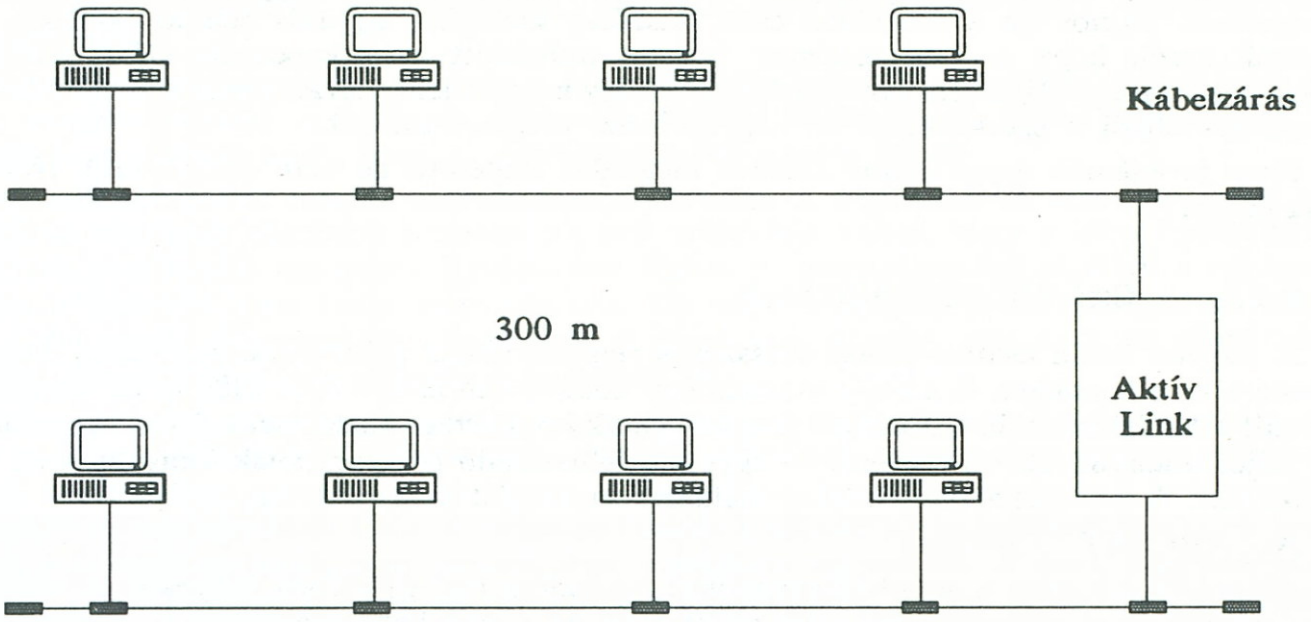
Csillag topológia esetén a rendszer felépítése kötetlen fa-struktúrát mutat. Az elágazási pontokon aktív illetve passzív áramköri elemeket (HUB) találunk (Az aktiv HUB olyan speciális kapcsolóelem, amely erősítőt is tartalmaz.) A rendszer bármely munkaállomása által kibocsátott jelet mindegyik állomás egyidőben érzékeli (radió mód).



8.30. ábra

Csillag topológiájú ARCNET hálózat.

BUS topológia esetén a munkaállomások sorosan kapcsolódanak a kábelre. Ezeket a szegmenseket aktív összekapcsolást végző elemek kötik össze (Link-ek). A szabad BUS végeket az impedancia illesztések miatt megfelelő ellenállással kell lezárni. (Ennek hiánya a rendszer üzemképtelenségét okozza!) Egy sínre maximum 8 munkaállomás csatlakozhat.

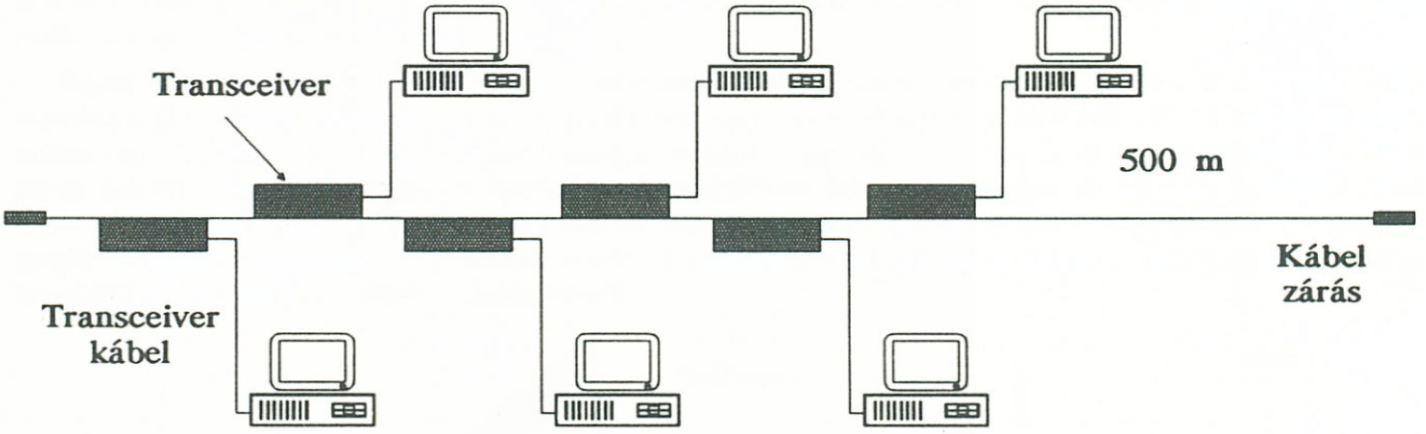


8.31. ábra

Busz topologiájú ARCNET hálózat.

ETERNET hálózatok

A rendszer működése során a munkaállomások mindegyike egyenlő eséllyel kezdeményezhet adást a sín felé. Ha a sín foglalt, akkor a rendszernek várakoznia kell. A sín nyugalmi állapotát érzékelve adást kezdeményezhet. Ha egyidőben kéri két munkaállomás a sín használatát, akkor az erre a célra kifejlesztett áramkörök ezt érzékelik és megszakítják az adást. Ezzel egyidőben indul el mindkét gépben egy véletlen számláló ciklus, amelynek lejárta után a gép újra kezdeményezheti az adást. (Mivel a számlálás

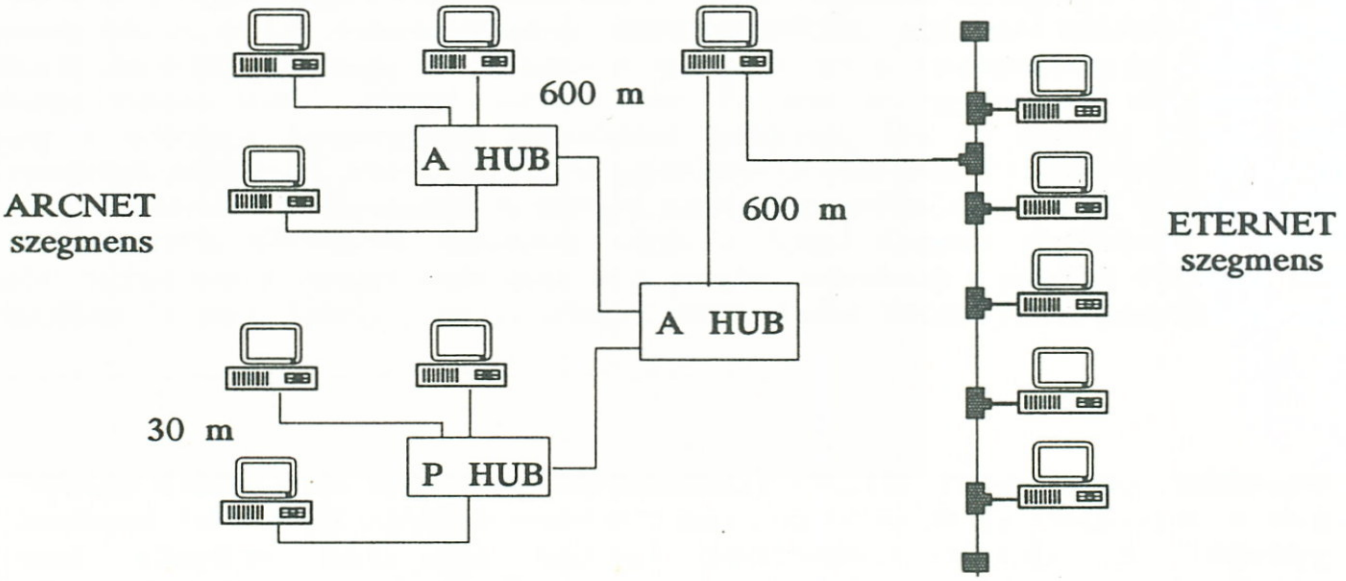


8.32. ábra

Vastag ETERNET hálózat.

véletleszerű, annak az esélye, hogy a két gép ismét egyszerre jelent be sín hozzáférési igényt igen kicsiny.)

Az ETERNET hálózatoknak alapvetően két fajtája van, az úgynevezett vékony illetve vastag kábeles megoldás. Az előbbit általában korlátozott kiterjedésű esetekben alkalmazzák. Ezek szegmeneseit (30 munkaállomás) REPEATER-ek kötik össze. Az utóbbit nagy kiterjedésű hálózatoknál alklamazzák (2500 m). A rendszer rövidebb, legfeljebb 500 m-es szegmensekre tagolódik. A munkaállomások TRANSCEIVER-eken keresztül kapcsolódnak



8.33. ábra

Kevert ARCNET és ETERNET rendszer.

a hálózat kábeléhez. (Érdekesség, hogy a kábelt magát nem szakítják meg, hanem egy tűt szúrnak bele annak kábelén keresztül, aminek segítségével kommunikálnak.

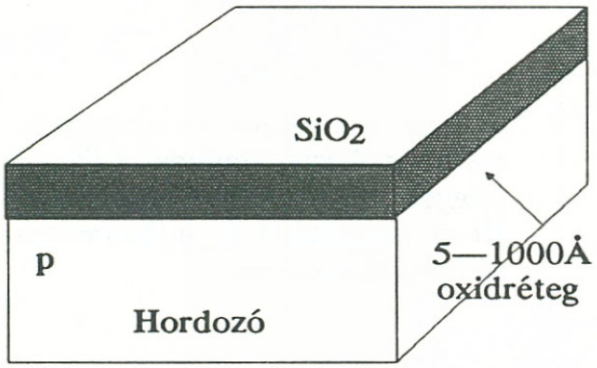
A hálózati elemeket össze is lehet kapcsolni. Keverhetjük egy rendszeren belül az ETERNET és ARCNET rendszereket az interal BRIDGE (file serverek esetén) illetve external BRIDGE (külső munkaállomás esetén) áramkörök segítségével.

1. Függelék

1.1. Integrált áramkörök gyártása

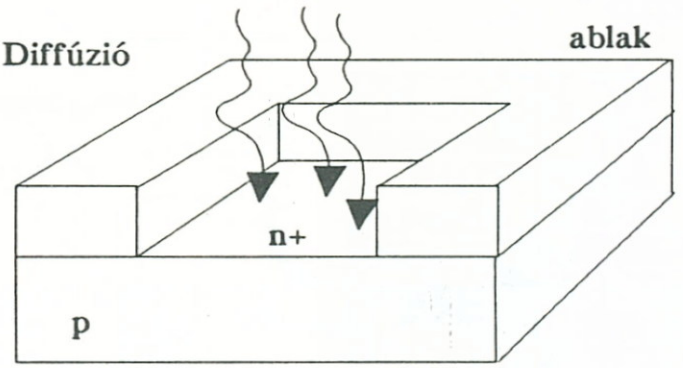
Az integrált áramkörök készítésének alapanyaga nagy tisztaságú,körülbelül 40 cm hosszúságú, 5—7 cm átmérőjű szilícium egykristály, amelyet a gyártó általában p vagy n típúsra szennyez.Az így előállított félvezető csak az áramkörök hordozójául szolgál —szubsztrát—, nem vesz részt a tranzisztorok,diódák felépítésében. A rudat néhány 100 mikron vastag szeletekre vágják. Ezek durva vágási felületét csiszolással, illetve kémiai polírozással tökéletesen simává alakítják.

Ezzel párhuzámosan készítik a fotomaszkot. Ehhez számítógéppel megtervezik az áramkör topológiáját, majd szintén a gép segítségével egy speciális, nagy méretű (2*"2 m) kétrétegű műanyag fóliába 50 mikron pontosságú rajzot vágnak. A nem kívánt részekről a felső piros takaró réteget lefejtik. A komplett áramkörhöz több (4—6) maszk készül. A maszkokat több lépcsőben kicsinyítik és egymás mellé léptetéssel megsokszorozzák. (Ezzel a gyártás hatékonyságát növelik.) A kicsinyített filmnegatívról krómbevonatú üveglapra pontos, kopásálló kontaktmásolatot készítenek.



1. ábra

*Oxidált hordozó—egy tranzisztornyi darabkája 300*300 mikron.*



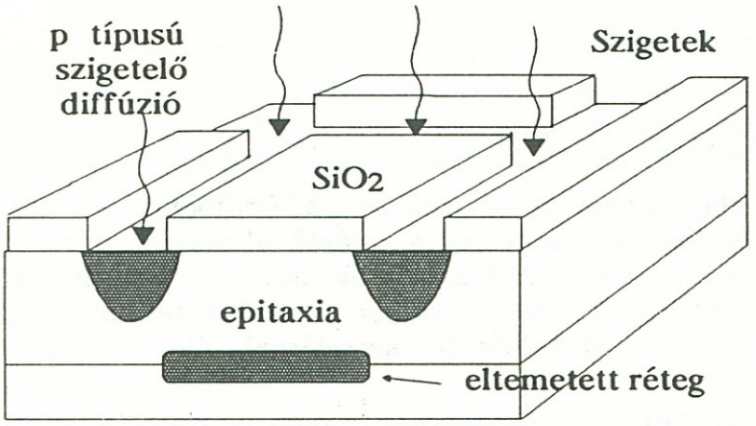
2. ábra

Diffúzióval n+ réteget csapatnak az ablakra. Ebből lesz a későbbiekben az eltemetett réteg.

Ez után a hordozólapra csőkemencében 5—1000 angström vastagságú SiO_2 oxidréteget visznek fel. A maszk felhasználásával [1]**fotolitográfiás** eljárással ablakot nyitnak. Az ablakon keresztül Sb vagy P diffúzióval n+ (a + jel a szennyezőanyag koncentráció erőteljes voltára utal) réteget hoznak létre. Ez lesz az úgynevezett eltemetett réteg, amely a kollektor hozzávezetés ellenállását csökkenti. Ezt az epitaxia követi, amely rá-rendezett növesztést jelent, vagyis az egykristályos szerkezetet továbbépítik Si tartalmú gázelegyből történő lecsapatással. A művelet során eltemetődik az n+ réteg. Ezután oxidáció és fotolitográfia következik, amelynek végén n típusú szigetek képződnek. A szigetek között diffúzióval p réteget alakítanak ki.A további műveletek e szigetek felületén illetve belsejében folynak. Sekély p és n+ rétegek diffúziójával kialakítják a tranzisztorokat.

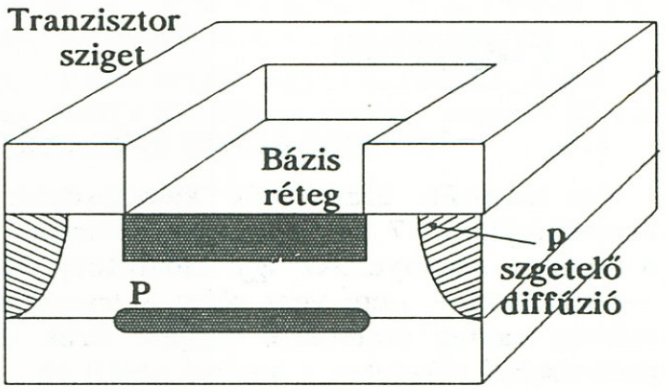
---

1 Néhány tized mikron vastagságú marószereknek ellenálló, fényérzékeny lakkréteget hordanak fel a SiO_2 felületre, amelyet a maszkon keresztül megvilágítanak. A meg nem világított lakkréteget szerves oldószerekkel leoldják. A lakkréteg ellenállóképességének növelésére az anyagot 100—200 °C közötti hőmérsékleten beégetik. A kiszabadított SiO_2 felületet HF tartalmú marószerrel „kivágják".



3. ábra

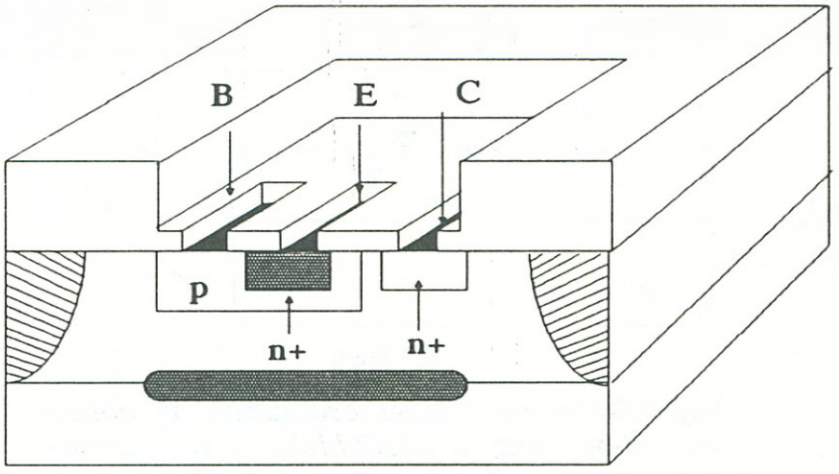
Mély, az epitaxián át az alaprétegig hatoló szigetelő rácsmintázat.



4. ábra

A tranzisztor-szigeten kialakított bázis diffúzió.

Az utolsó folyamatban a kontaktálást végzik el. Az E,B,C rétegek felett ablakot nyitnak,majd Al vezetékeket csapatnak le bennük. Ezt az áramkörök elektoromos ellenőrzése



5. ábra

A kész tranzisztor kontaktálása során a bázis, emitter, kolletor rétegek fölött nyitott ablakokba Al-ot csapatnak le a kontaktusok kialakításához.

követi. A hibás áramköröket mágneses festékkel megjelölik a későbbi kiválogatás miatt. Ezután a határfelületek mentén morzsákra tördelik a hordozó szeleteket. A morzsák kivezetéseit termokompressziós módon rögzítik az áramkörök fémezéséhez. Ehhez rendszerint vékony Au szálat használnak. A folyamat végén műanyag fröccssajtolással kialakítják az alkatrész házát.

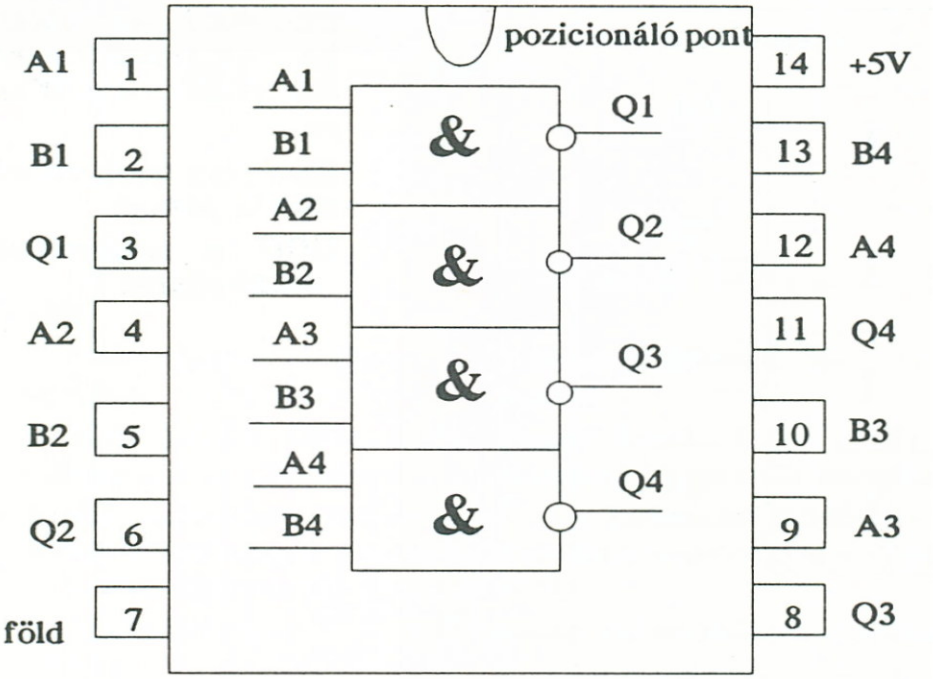
A tokozást követően ismét mérés következik.A technológia minden lépésénél 100 %-nál kisebb a kihozatal. Ezért a gazdaságosság növelése érdekében minden folyamatnak és felhasznált anyagnak nagy tisztaságúnak kell lennie.

1.2. Néhány integrált áramkör bekötése

Az alábbiakban néhány TTL és CMOS integrált áramköri alapkapu bekötési rajzát mutatjuk be.

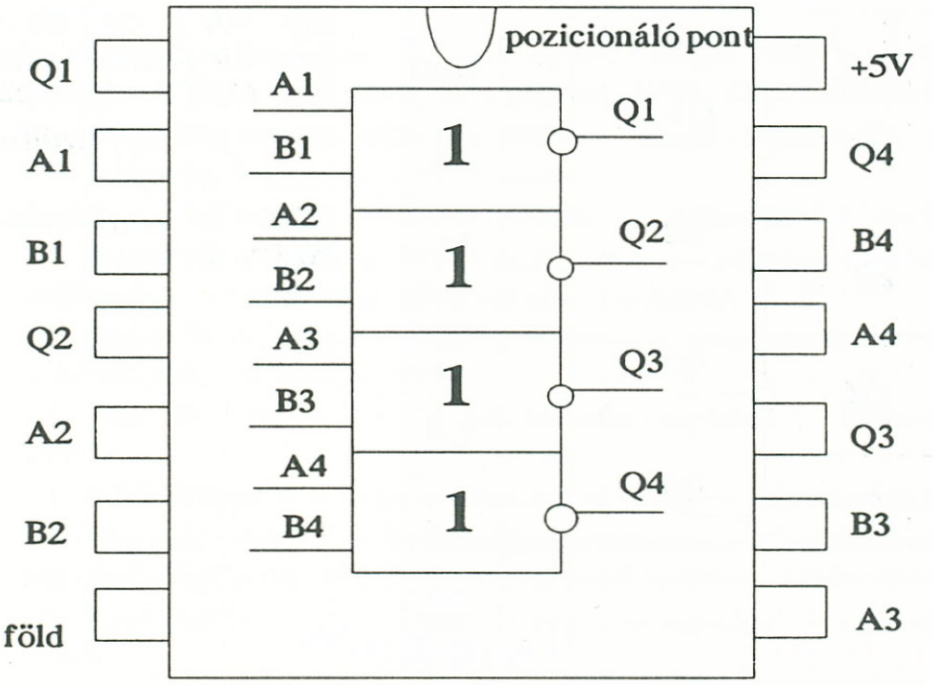
Az SN 74xx-es TTL sorozat standard paraméterei:

pozitív bemeneti feszültség	5,5 V	max tápfeszültség	5,25 V
negatív bemeneti feszültség	—0,5 V	min tápfeszültség	4,75 V
pozitív bemenő áram	1 mA	bemeneti Umax L-nél	0,8 V
negatív bemenő áram	—12 mA	bemeneti Umin H-nál	2 V
pozitív kimeneti feszültség	Ucc	forrasztásihőmérséklet	265°C
negatív kimeneti feszültség	—0,5 V	üzemi hőmérséklet	0...70°C



TTL SN7400
Megnevezés: ÉS—NEM
Bemenet: 4 x 2
Kimenet TP
Logikai Sheffer-függvény
$Q = \overline{A*B}$
P = 10 mW/kapu
tp = 10 ns

Bemenetek		Kimenetek
H	H	L
Bármilyen más kombináció		H

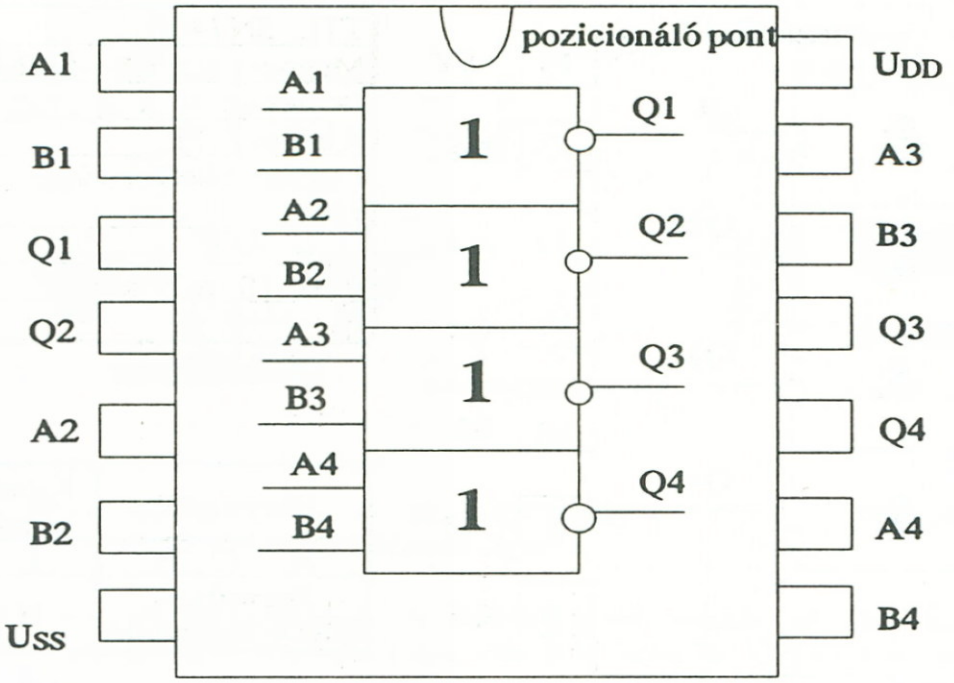


TTL SN7402
Megnevezés: VAGY—NEM
Bemenet: 4 x 2
Kimenet TP
Logikai Pierce-függvény
$Q = \overline{A+B}$
P = 14 mW/kapu
tp = 10 ns

Bemenetek		Kimenetek
L	L	H
Bármilyen más kombináció		L

A CMOS áramköri család CD 4001 AE sorszámú tagjának adatai:

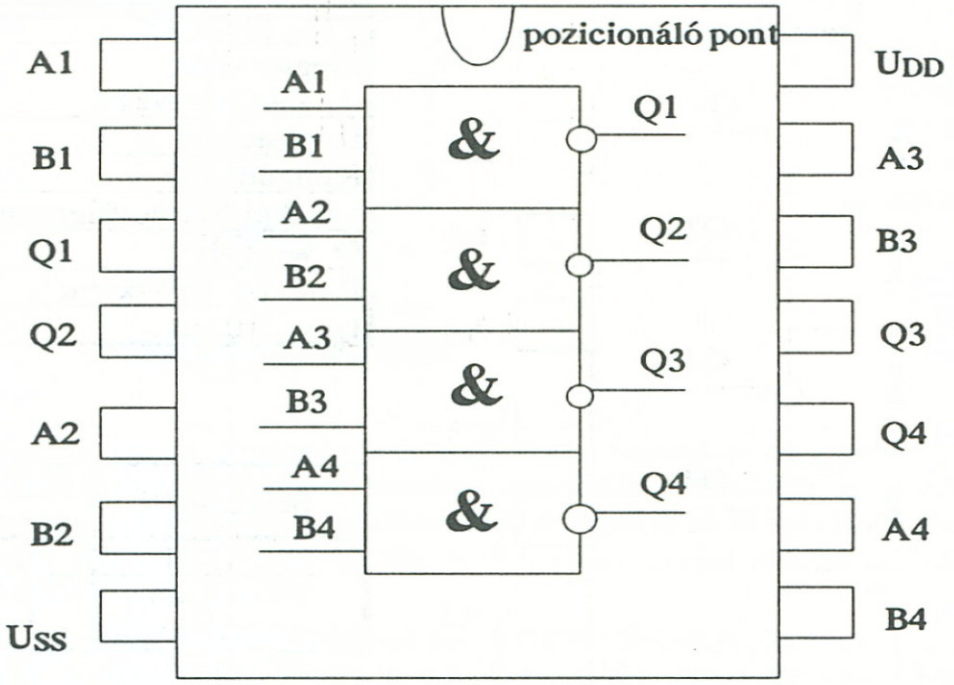
Jellemző	Jelölés	Feltételek:	U_{ki}	U_{DD}	Min.	Névleges	Max.	Egység
Nyugalmi tápáram	I_{DD}			5...10		0,005	0,5	mikroA
Kimenő L áram	I_{kiL}	$U_{Be}=U_{DD}$	0,4...0,5	5...10	0,3...0,6	1...2,5		mA
Kimenő H áram	I_{kiH}	$U_{Be}=U_{SS}$	2,5...9,5	5...10	-0,3... -0,25	-2...-1		mA
Kimeneti L szint	U_{kiL}	$I_{Ki}=0$ $U_{Be}=U_{DD}$		5...10		0	0,01	V
Kimeneti H szint	U_{kiH}	$I_{Ki}=0$ $U_{Be}=U_{DD}$		5...10	4,99... 9,99	5...10		V



CD 4001 -es CMOS integrált áramkör bekötése. (NEM—VAGY kapu)

USS a tápfeszültség negatív pontja.

UDD a tápfeszültség pozitív pontja.



CD 4011 -es CMOS integrált áramkör bekötése. (NEM—ÉS kapu)

USS a tápfeszültség negatív pontja.

UDD a tápfeszültség pozitív pontja.

1.1. IRODALOMJEGYZÉK

1.1.1. Könyvek

1. Dr.Arató Péter — Logikai rendszerek tervezése — Tankönyvkiadó, Budapest, 1990.

2. Bódi Sándor — Elektromosságtan és elektronika — Kossuth Lajos Tudományegyetem, Debrecen, 1987. (jegyzet)

3. Boér László Dóra Gyula Fenyő László Seres Attila — Az IBM PC-k belső felépítése — LSI Alkalmazástechnikai Tanácsadó szolgálat, Budapest, 1989.

4. Brückner Huba Dobrovolni Tibor Lohonzai Miklós — Display,rajzgép,bizonylatolvasó,konzolírógép,lyukszalaglyukasztó,lyukszalagolvasó perifériák — SZAMOK KSH oktató és tájékoztató központ, 1975.

5. Dr.Csáki Frigyes — Irányítástechnikai kézikönyv — Műszaki könyvkiadó, Budapest, 1977.

6. Dr.Cserny László — Számítástechnika I. II. — Tankönyvkiadó, Budapest, 1981. (jegyzet)

7. Ferenczi Szabolcs — Transputer alapú nagyteljesítményű számítógépek és programozásuk — Tanfolyami jegyzet 1991. (MULTILOGIC Számítástechnika Kft.)

8. G.B.Williams — Mikroprocesszor alapú rendszerek hibakeresése — Műszaki könyvkiadó, Budapest, 1989.

9. Gerd Thilele — Félvezetős tárak — Műszaki könyvkiadó, Budapest, 1990.

10. James W. Coffron William E.Long — Mikroprocesszoros rendszerek illesztési technikája — Műszaki könyvkiadó, Budapest, 1989.

11. James W. Coffron — Mikroprocesszor alapú rendszerek hibakeresése — Műszaki könyvkiadó, Budapest, 1989.

12. John D. Lenk — MOS alkalmazási segédlet — Műszaki könyvkiadó, Budapest, 1980.

13. K.M. Rübsam — Merevlemezes tárolók — Műszaki könyvkiadó, Budapest, 1991.

14. Kacsuk Peter — Multimikroprocesszoros rendszerek tervezése — Budapesti Műszaki Egyetem Mérnöktovábbképző Intézet, Budapest, 1979.

15. Karl Reichel — Gyakorlati mágnestechnika — Műszaki könyvkiadó, Budapest, 1985.

16. Kelemen Gáspár Golenzki István Tamás Péter Tóth Bertalan — Novell Netware felhasználói ismeretek I. II. — Computerbooks, Budapest, 1991.

17. Dr.Kónya László — PC - elektronika — Műszaki könyvkiadó, Budapest, 1991.

18. Kovács Győző — A számítógépek technikája — Tankönyvkiadó, Budapest, 1974.

19. Dr.Kovács Magda — 32-bites mikroprocesszorok 80386 — LSI oktatóközpont A mikroelektronika Alkalmazásának Kultúrájáért Alapítvány, Budapest, 1991.

20. Dr.Kovács Magda — Egyszerűen a mikroszámítógépekről — LSI Alkalmazástechnikai Tanácsadó Szolgálat, Budapest, 1985.

21. Dr.Kovács Magda — Mikroszámítógépek alkalmazása értelmező szótár 1-3.— Műszaki könyvkiadó, Budapest, 1991.

22. Madarász László — Digitális CMOS kapcsolásgyűjetemény — Műszaki könyvkiadó, Budapest, 1986.

23. Magyari Béla — Digitális IC-k (atlasz) — Műszaki könyvkiadó, Budapest, 1982.

24. Michael Andrews — Mikroprocesszorok és illesztőegységek — Műszaki könyvkiadó, Budapest, 1985.

25. Molnár Ferenc — Integrált áramkörös berendezések szerelése — Műszaki könyvkiadó, Budapest, 1984.

26. Németh Gábor Horváth László — Számítógép architektúrák — Akadémiai Kiadó, Budapest, 1991.

27. P.F. Lister — Egytokos mikroszámítógépek — Műszaki könyvkiadó, Budapest, 1988.

28. Dr.Sára Attila — A digitális számítógépek rendszertechnikai felépítése — Tankönyvkiadó, Budapest, 1988.

29. Dr.Szittya Otto — Mikroszámítógépek alkalmazásához ELEKTRONIKA 1-4. — Nyitott rendszerű oktatási segédlet, LSI Alkalmazástechnikai Tanácsadó Szolgálat, Budapest, 1989.

30. Sztaricskai Tibor — Elektronika II.rész — Kossuth Lajos Tudományegyetem Kisérleti Fizikai Tanszék, Debrecen, 1977. (jegyzet)

31. Texas — TTL receptek — Műszaki könyvkiadó, Budapest, 1978.

32. Texas — Analóg és illesztő integrált áramkörök — Műszaki könyvkiadó, Budapest, 1979.

33. Texas — Bevezetés a mikroprocesszor technikába — Műszaki könyvkiadó, Budapest, 1982.

34. U. Tietze. Ch. Schenk — Analóg és digitális áramkörök — Műszaki könyvkiadó, Budapest, 1981.

35. Urbán János — Matematikai logika — Műszaki könyvkiadó, Budapest, 1983.

36. V.H.Grinich — H.G.Jackson Példák integrált áramkörök alkalmazására — Műszaki könyvkiadó, Budapest, 1980.

37. Vancsó Gyula — Mikroszámítógép - elemek a tervezéshez — Műszaki könyvkiadó, Budapest, 1983.

38. Vaszenkov Sahnov — Szovjet mikroprocesszorok alkalmazása — Műszaki könyvkiadó, Budapest, 1984.

39. Wolfgang Link — BASIC a mérés-, a vezérlés- és a szabályozástechnikában Műszaki könyvkiadó, Budapest, 1987.

1.1.2. Folyóiratcikkek

1. Computer Panoráma — Egerek, 2, 6-7, 8, számok 1990

2. Computer Panoráma — 32 bites architektúrák 2. szám 1990

3. Computer Panoráma — EISA sínrendszer 1. szám 1990.

4. Computer Panoráma — ISA kontra mikrocsatorna 1. szám 1990.

5. Computer Panoráma — Winchesterek installálása 1. szám 1990.

6. Computer Panoráma — Szkennerek 10. szám 1990.

7. Computer Panoráma — Monitorlexikon 9. szám 1990.

8. Computer Panoráma — Boncasztalon a merevlemez 3. szám 1991.

9. Computer Panoráma — Félvezetős hűtés 9.szám 1991.

10.Computer Panoráma — Matematikai koprocesszorok 2. szám 1991.

11.Computer Panoráma — Tuningolás 9,10. számok 1991.
12.Computer Panoráma — Tárolóbővítés 8.szám 1992.
13.Computer Panoráma — Alpha mikroprocesszor 7.szám 1992.
14.Computer Panoráma — Optikai lemezek 5.szám 1992.
15.Computer Panoráma — Ionsugaras nyomtatók 2.szám 1992.
16.Magyar Elektronika — Vezérlő és adatgyűjtő rendszerek tervezése IBM PC-re VI. évf. 9.sz.
17.Magyar Elektronika — Alternatív számítógépstruktúrák V. évf. 5.sz.
18.Magyar Elektronika — SCSI- adapter a merevlemezekhez VII. évf. 1.sz.
19.Magyar Elektronika — Winchester fejlesztés a KFKI-ban /Egri Béla/ V.évf. 3.sz.
20.Magyar Elektronika — Lézernyomtató a Videotonnál /Baráth István- Kádár Ferenc- Paulinszky Károly- Rákóczi Ferenc/ III.évf. 2.sz.
21.Magyar Elektronika — Mikroelektrónikai szereléstechnika /Laczkó Béla/ I.évf. 1.sz.
22.Magyar Elektronika — Törölhető optikai tárolólemezek III. évf. 1.sz.
23.Magyar Elektronika — Mikroprocesszor katalógus 1.sz. III. évf. 4.sz.
24.Magyar Elektronika — Új rendszerű repülő lemezmeghajtó II. évf. 5.sz.
25.Magyar Elektronika — 8 bites mikroprocesszorstruktúrák /Laczkó Béla/ I.évf. 3.sz.
26.Magyar Elektronika — 32 bites mikroprocesszorok /Dr.Kovács Magda/ I.évf.3.sz.
27.Tudomány — Egy párhuzamos számítógép a Connection Machine 8.szám 1987.
28.Tudomány — Korszerű számítógép-hálózatok 12.szám 1987.
29.Tudomány — A korszerű számítógépek felépítése 12.szám 1987.
30.Tudomány — A korszerű számítógépek adattárolói 12.szám 1987.
31.Tudomány — Látványos szuperszámítógépek 11. szám 1988.
32.Tudomány — A hálózatszervezés alapelvei 11. szám. 1991.
33.Tudomány — A jövő század számítógéprendszerei 11. szám 1991.

Tárgymutató

L

M

N

O

Ó

P

R

S

T

V

W

Z

Tartalomjegyzék