

Dienes Zoltán

# Funkcionális áramkörök vizsgálata



A követelménymodul megnevezése:  
**Elektronikai áramkörök tervezése, dokumentálása**

A követelménymodul száma: 0917-06 A tartalomlelem azonosító száma és célcsoportja: SzT-026-50



## FUNKCIONÁLIS ÁRAMKÖRÖK VIZSGÁLATA

### ESETFELVETÉS – MUNKAHELYZET

Ön egy elektronikai szervizben dolgozik. A munkahelyére egy digitális áramkörök tartalmazó berendezést (fényújság) hoztak szervizelésre. A fényújság egy 1kbyte-os ROM-ban tárolt ASCII kódolású szöveget jelenít meg ciklikusan. A fényújságnak semmiféle dokumentációja nem áll rendelkezésre. A főnöke Önt bírta meg azzal, hogy a berendezést vizsgálja meg és tervezze meg a fényújság funkcionális blokkvázlatát.

A feladat végrehajtásához elengedhetetlen a digitális áramkörök alapvető ismerete.

Feltételezzük, hogy a tananyag feldolgozása előtt az alapvető digitális technikai anyag és eszközismeret, valamint a digitális berendezések tervezésével kapcsolatos alapvető ismeretek már rendelkezésre állnak.

### SZAKMAI INFORMÁCIÓTARTALOM

A digitális áramkörök tervezését és megvalósítását nagymértékben megkönnyíti a funkcionális áramkörök alkalmazása. A gyártók már a kezdeti időkben felfigyeltek arra, a forgalomban lévő SSI (Small Scale Integrated ) áramkörökből bizonyos típusú funkcionális áramköröket építettek a felhasználók. A technológiai fejlődés lehetővé tette, hogy azokat az áramköröket, amiket a felhasználók a kis integráltságú SSI (Small Scale Integrated ) IC-kből leggyakrabban megépítettek beletegyék egy IC-be. Ezekből, a több tipikus SSI áramkört tartalmazó funkcionális integrált egységekből jöttek létre az MSI IC-k (Middle Scale Integrated Circuit) .

Funkcionális elem nem csak valamely funkciót egy IC -ben megvalósító áramkört értünk, hanem a tervező számára egy egységként kezelhető blokkot. A funkcionális IC részletes belső felépítését nem szükséges ismerni az alkalmazáshoz. Az MSI áramkörök olyan célorientált hálózatok , amelyek előnyösen felhasználhatók SSI IC-ből készült áramkörök realizálásánál . A bonyolult LSI (Large Scale IC), VLSI (Very LSI) IC belsejének tervezésénél is ilyen formában használjuk a funkcionális elemeket.

A funkcionális elemeket hagyományosan két csoportra osztjuk aszerint, hogy kombinációs hálózatot vagy sorrendi hálózatot valósítanak meg.

## KOMBINÁCIÓS FUNKCIONÁLIS ELEMELK

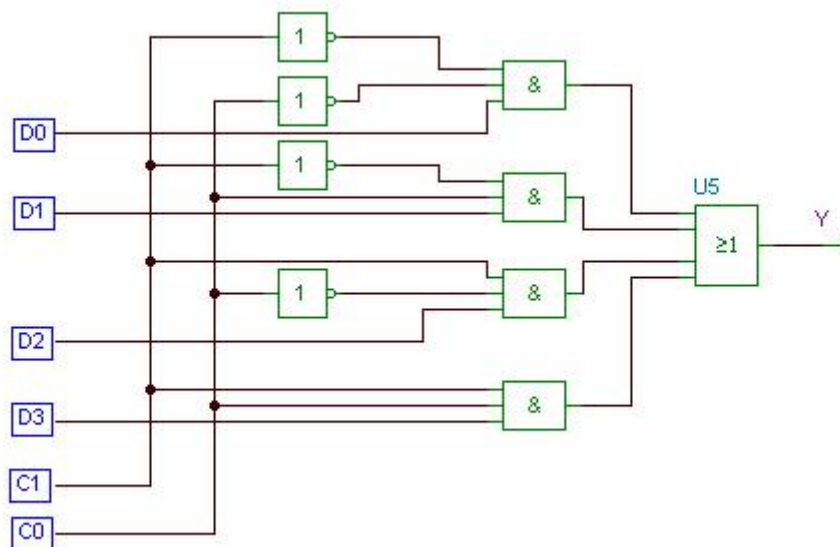
## 1. Multiplexer

A multiplexer feladata az, hogy a több bemenetere érkező jelből egyet vezessen a kimenetere. A kiválasztást a beérkező a címbitek végzik el.

A multiplexer címzése mindig kettes számrendszerben történik és mindig annyi bemenő csatornája van, amennyi különféle értéket a címbitek felvehetnek. A címbitekre adott értékeket bináris számként értelmezve, megkapjuk a kimenetre jutó bemeneti jel címét. Ha a címbemenetek száma  $n$ , az adatbemenetek száma  $2^n$ .

A multiplexer tartalmaz egy címdekódoló fokozatot. A címdekódoló kimenetei engedélyezik egy-egy csatorna jelének a közös kimenetre jutását. A multiplexerek digitális és analóg kivitelben készülnek. A digitális multiplexerekben (1. ábra) az ÉS kapuk bemenetere érkező címbitek kombinációi végzik az adatbemenetek engedélyezését illetve tiltását (egy címkombináció egy-egy kaput engedélyez). Az ÉS kapuk kimenő jele egy VAGY kapun keresztül jut a kimenetre. A digitális multiplexereket TTL és CMOS kivitelben egyaránt készítik. A jelterjedés iránya csak egyféle lehet. (Bemenet→kimenet)

Az 1. ábrán egy 4 bemenetű multiplexer logikai vázlata látható.



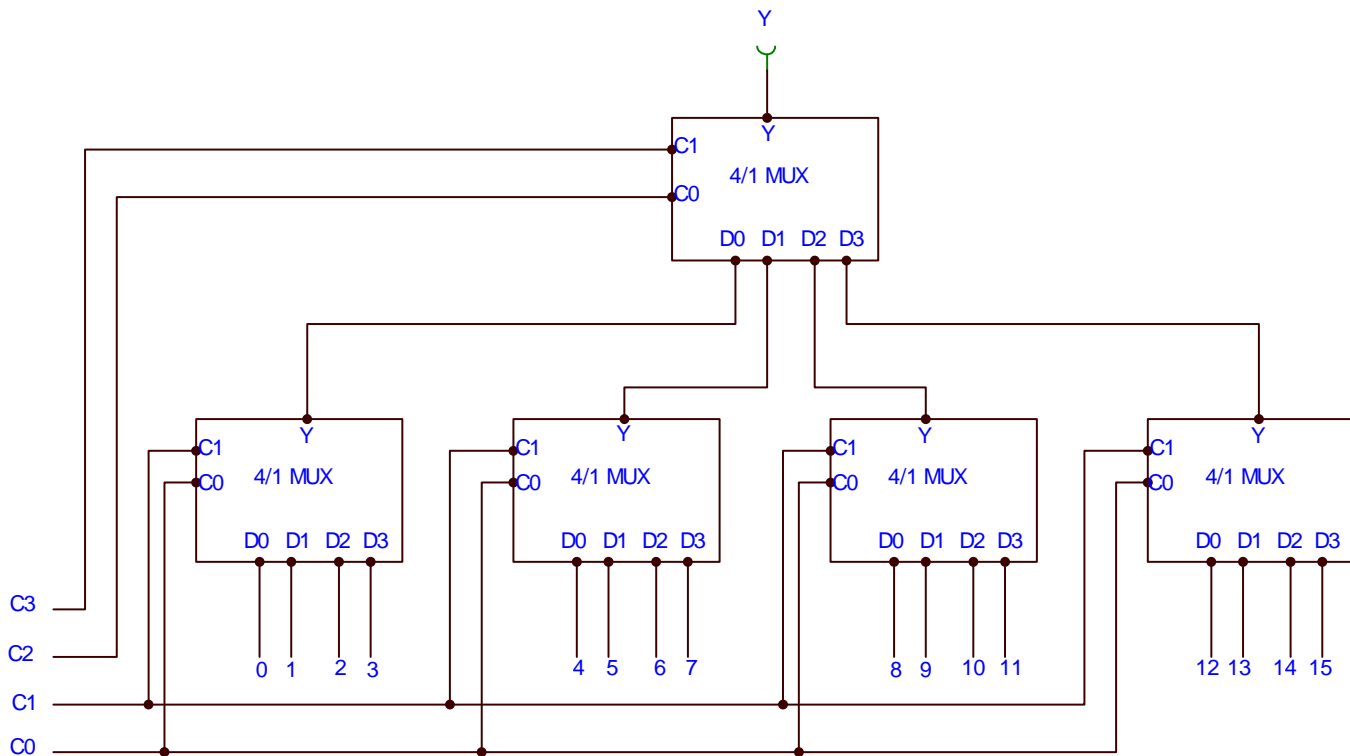
1. ábra. Digitális multiplexer

Az analóg multiplexerekben a címbitek egy-egy analóg kapcsolót zárnak. Az analóg multiplexereket csak CMOS áramkörökkel lehet megvalósítani és az ilyen típusú áramkörök kétféle jelterjedési irányt engednek meg. (Bemenet → kimenet, Kimenet→ bemenet).

**Multiplexer csatornaszámának növelése:**

Ha a multiplexer bemeneteinek száma nem elegendő, akkor több multiplexer alkalmazásával megnövelhetjük azt.

Az 2. ábrán egy 4/1-es multiplexerekből készített 16/1 multiplexer látható.



2. ábra. Multiplexer bemenet szám bővítés

Az azonos szinten lévő multiplexerek címbemeneteit közösítjük. A multiplexerek kimenő jeleit egy multiplexer vezeti a kimenetre.

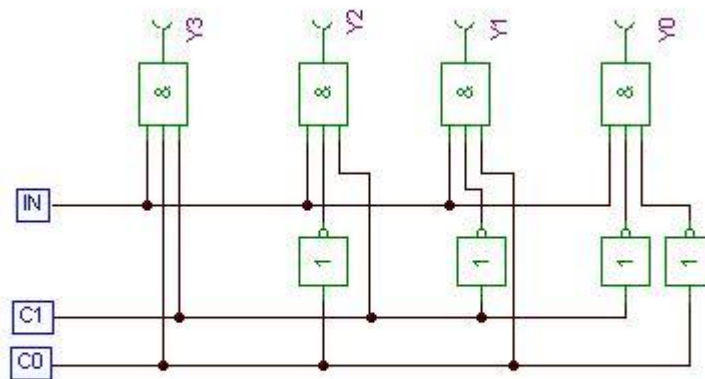
Az első szinten lévő multiplexer címbemenetei a bemeneten lévő multiplexerek kimeneteit engedélyezik. A bemeneti multiplexerek közösített címbemenetei pedig kiválasztják a megfelelő bemenetet.

A szintek száma elvileg tetszőlegesen növelhető.

A tervezésnél figyelembe kell venni, hogy minden egyes szint növeli az áramkör késleltetését.

## 2. Demultiplexer

A Demultiplexer a multiplexerrel ellentétes funkciót valósít meg. Feladata, hogy az egyetlen bemenet jelét a címbemenetek által kiválasztott kimenetre kapcsolja. A multiplexerhez hasonlóan analóg és digitális kivitelben készítik. Az *analóg demultiplexerek* csak CMOS áramkörökkel lehet megvalósítani és felépítése megegyezik az analóg multiplexerével. A *digitális demultiplexerek* 3. ábra annyi ÉS kaput tartalmaznak, ahány kimenete van az áramkörnek.



3. ábra. Demultiplexer

Az ÉS kapuk egy-egy bemenetét közösítve kapjuk a demultiplexer bemenetét. A fennmaradó bemeneteket a Cím bemenetekre kötve engedélyezzük a demultiplexer kimeneteit. A digitális demultiplexereket TTL és CMOS áramkörökkel valósíthatjuk meg. Ilyenkor a jelterjedés csak egyirányú lehet. A digitális demultiplexereket sok változatban gyártják. 2, 4, 8 és 16 csatornás áramkörök is vannak közöttük. Néhány közkedvelt típus:

### Digitális multiplexer:

- 8 csatornás            74LS151
- 16 csatornás        74LS150 illetve 4067
- Két 4 csatornás     74LS153
- Négy 2 csatornás   74LS157

### Digitális demultiplexer:

- 16 csatornás        74LS154
- Két 4 csatornás    74LS139

### Analóg multiplexer/demultiplexer áramkörök

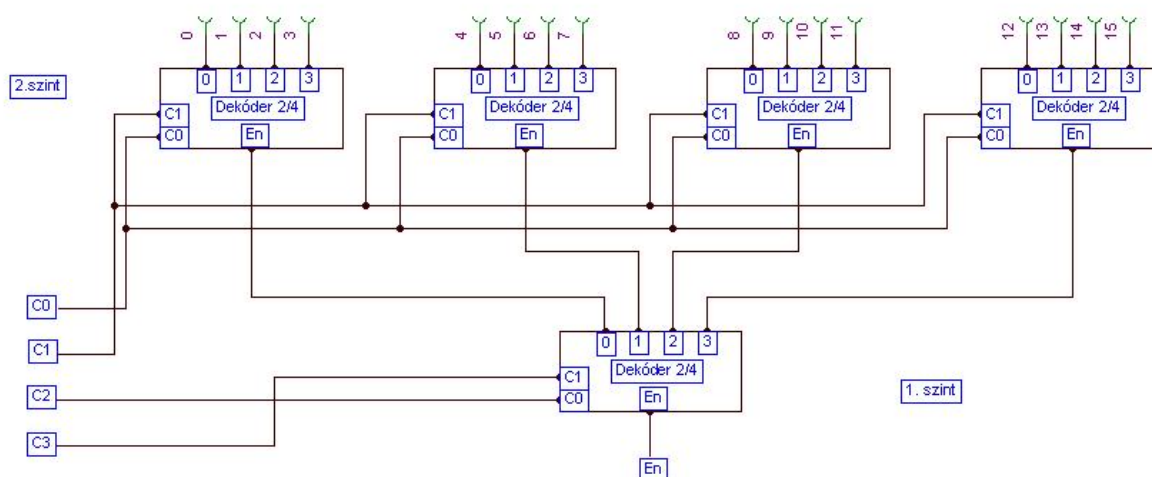
- 8 csatornás                    4067
- 16 csatornás                 4051
- Két 4 csatornás               4052

### 3. Dekóder

A dekóder tulajdonképpen egy olyan demultiplexer áramkör, ahol az áramkör bemenetét fixen aktív szintre kötjük. Ilyenkor az aktív szintre kötött bemenet az áramkör engedélyezését végzi. Ha az engedélyező bemenet inaktív, az MSI áramkör összes kimenete is inaktív. A dekóder kimenetén mindig csak az az egy kimenet aktív, amit a cím bemenetek segítségével kiválasztunk. Az aktív kimenet sorszáma a cím bemenetek bináris kódját dekódolja..

### 4. Dekóderek bővítése

Ha a dekóder kimeneteinek a számát növelni szeretnénk, akkor azt a dekóder engedélyező bemenetei segítségével megtehetjük. Az 4. ábrán 2/4 dekóderek segítségével készítünk 4/16 -os dekódert.



4. ábra Dekóder bővítése

Az 1. szinten lévő dekóder kimenetei engedélyezik a 2. szinten lévő dekódereket. Azt, hogy az 1. szinten lévő dekóder kimenetei melyik 2. szinten lévő dekódert engedélyezik a C2 , C3 címbitek segítségével lehet kiválasztani. A kiválasztott (engedélyezett) dekóder kimenetei közül a C1 , C0 címbemenetek segítségével választhatunk.

#### Népszerű dekóder típusok:

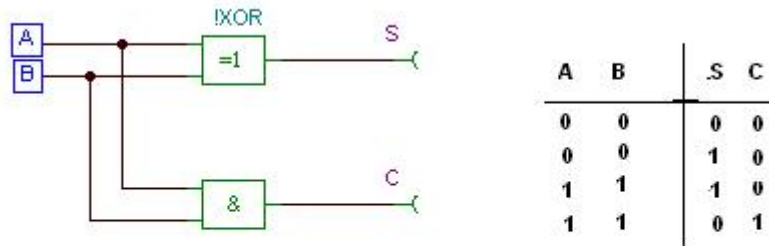
- 74138 (3/8)
- 74154 (4/16)

## 5. Aritmetikai áramkörök

### Összeadó

Azokat a digitális áramköröket, amelyek számtani műveletek végzésére alkalmasak aritmetikai áramköröknek nevezzük. Az aritmetikai áramkörök bemenetét képező számokat megfelelő bináris kódban kifejezve kell megadni. Az eredményként kapott számok ugyan abban a kódban adódnak.

A számítási jellegű műveletek alapeleme az összeadó áramkör. Az áramkör működésének megértéséhez először a félösszeadó áramkör működését kell megértenünk. Az áramkör kapcsolási rajza és igazságtáblázata.



5. ábra. Félösszeadó *S* – Összeg *C*–Maradék ,átvitel

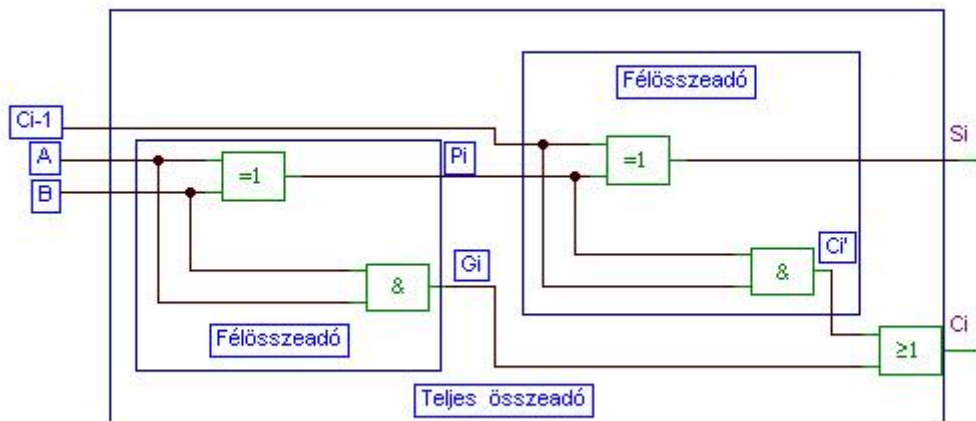
Két bit maradék nélküli összeadására a kizáró vagy kapu alkalmas, mivel bemeneti változók 0–1 értékeire pontosan olyan "választ" ad ami a bit értékek összeadásának felel meg. A maradék (átvitel) akkor keletkezik, ha mindkét bemeneti változó 1 értékű. Ezt valósítja meg az ÉS kapu. A félösszeadó önmagában nem alkalmas több bites számok helyiértékenkénti összeadására, mivel nem veszi figyelembe az előző helyiértéken keletkező átvitelt. Az átvitelt is figyelembe vevő összeadó az úgynevezett **teljes összeadó**. A teljes összeadónak ezek szerint 3 bemenetűnek kell lenni.  $A_i$ ,  $B_i$  a két összeadandó bit.  $C_{i-1}$  az előző helyiértéken keletkező átvitel. Működése az alábbi táblázatból követhető nyomon.

$C_{i-1}$	$A_i$	$B_i$	$S_i$	$C_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1

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

$A_i$  ,  $B_i$  az  $i$ . oszlop bitjei  $C_{i-1}$  az  $(i-1)$ . oszlop összeadása után keletkező átvitel.

A teljes összeadó két félösszeadóból és egy VAGY kapuból alakítható ki az 6. ábrának megfelelően.



6. ábra. Teljes összeadó

A teljes összeadót az alábbi logikai függvények írják le:

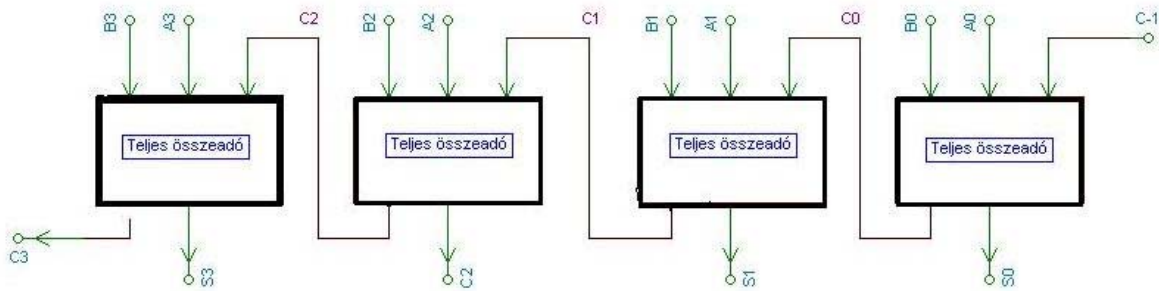
$S_i = C_{i-1} \oplus P_i$ , ahol  $P_i$  a félösszeadó kimenetén keletkező összeg.

$C_i = C_i' + G_i$ , ahol  $G_i$  és  $C_i'$  a félösszeadók kimenetén keletkező átvitel.

Teljes összeadó felhasználásával az összeadást több bitre is kiterjeszthetjük.

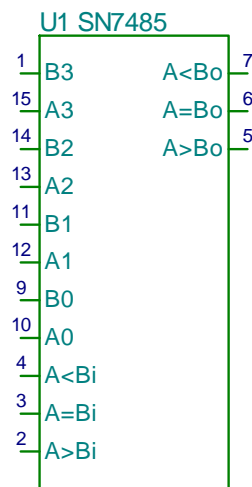
Az 7. ábrán egy 4-bites teljes összeadó látható.





7. ábra Négybites teljes összeadó

**Komparátor:** A digitális komparátor egy sokoldalúan felhasználható aritmetikai áramkör. Feladata, hogy két  $n$  bites abszolútértékes ábrázolású szám között jelezze a relációt. A komparátornak általában két  $n$  bites adatbemenete ( $A_{n-1} \dots A_0$  és  $B_{n-1} \dots B_0$ ),  $A=B$ ,  $A < B$  és  $A > B$  kimenete valamin az áramkör bővítésére (kaszkádosítás) használható  $A=B$ ,  $A < B$  és  $A > B$  bemenete van. A TTL áramköröknél a 7485 a használatos típus. Ez két 4 bites bináris (vagy BCD) számot hasonlít össze 8. ábra.

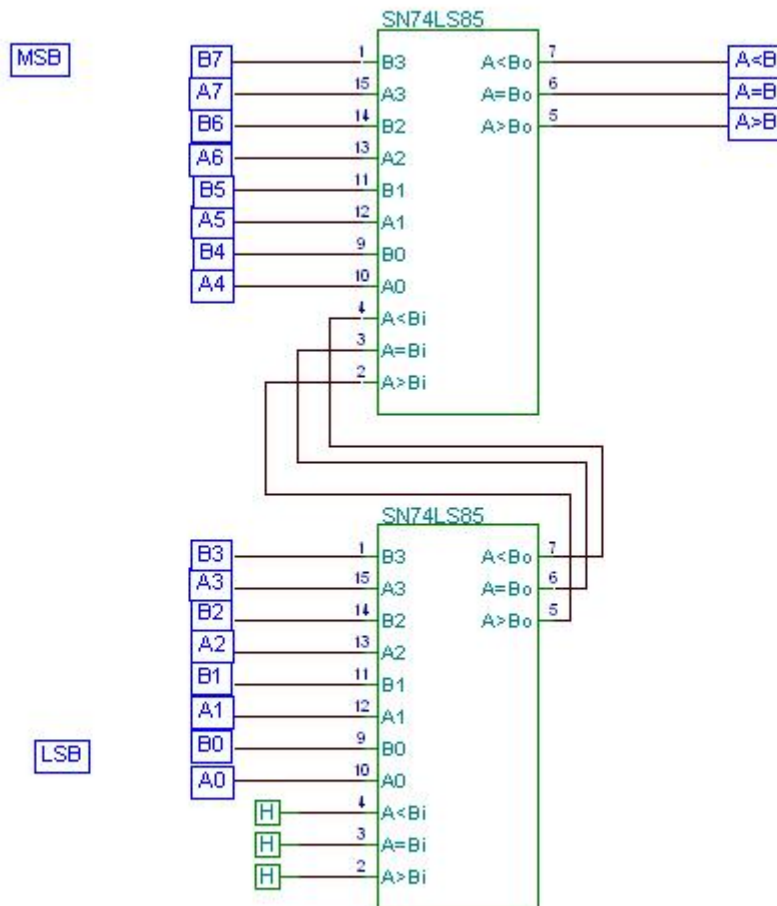


8. ábra. Komparátor IC

Ha az összehasonlítást 4 bitnél nagyobb bináris számokra is ki szeretnénk terjeszteni, lehetőségünk van rá a komparátorok kaszkádosításával.

Erre mutat példát az 9. ábra.

Az összehasonlítást mindig az LSB-nél (legkisebb helyi értékű bit) kell kezdeni. Az e biteket összehasonlító áramkör  $A < B$ ,  $A = B$ ,  $A > B$  kimeneteit kell a magasabb helyi éréken lévő komparátor  $A < B$ ,  $A = B$ ,  $A > B$  bemeneteihez kötni. Az összehasonlítás eredménye az MSB (legnagyobb helyi értékű bit) helyi éréken lévő komparátor  $A < B$ ,  $A = B$ ,  $A > B$  kimenetein fog megjelenni.



9. ábra. 8 bites komparátor

## SORRENDI FUNKCIONÁLIS ÁRAMKÖRÖK

### SZÁMLÁLÓK

A számlálók (COUNTER) olyan sorrendi hálózatok, amelyek az impulzusok számlálására és a számlált érték tárolására alkalmasak. A bemenetükre adott órajel impulzusokat adott feltételek között megszámlolják és a következő órajel impulzus megérkezéséig emlékeznek a számlálás eredményére. A számlálók építőelemei a flip-flopok. A számlálókat többféle szempont alapján csoportosíthatjuk.

## 1. A számlálót alkotó flip-flopok működése alapján:

- *Aszinkron számláló:* A beérkező órajel nem jut el mindegyik flip-flophoz. A flip-flopok egymást billentik.
- *Szinkronszámláló:* A flip-flopok billenése egyidejűleg történik és az állapotváltozásokat a flip-flopok előző értéke és az őket vezérlő kombinációs hálózat határozza meg.

## 2. A számlálás iránya alapján:

- *Előre számláló (UP COUNTER):* A beérkező órajel impulzusok hatására a számláló eggyel növeli az értékét.
- *Vissza számláló (DOWN COUNTER):* A beérkező órajel hatására a számláló eggyel csökkenti az értékét.
- *Kétirányú számláló (UP/DOWN COUNTER):* A számlálás irányát vezérlő jeltől függően, a számláló előre vagy visszafele számol.

## 3. A számlálás rendszere alapján:

- Különböző kódrendszerek szerint működő számlálókat különböztetünk meg. Pl. Bináris, NBCD, GRAY stb.

## 4. A számlálás kezdő értékének programozhatósága alapján:

- Vannak olyan számlálók, amelyeknél a tárolók bemenetei is ki vannak vezetve, így a kezdőérték párhuzamosan betölthető. Az ilyen számlálókat *programozható (PRESET) számlálóknak* nevezzük

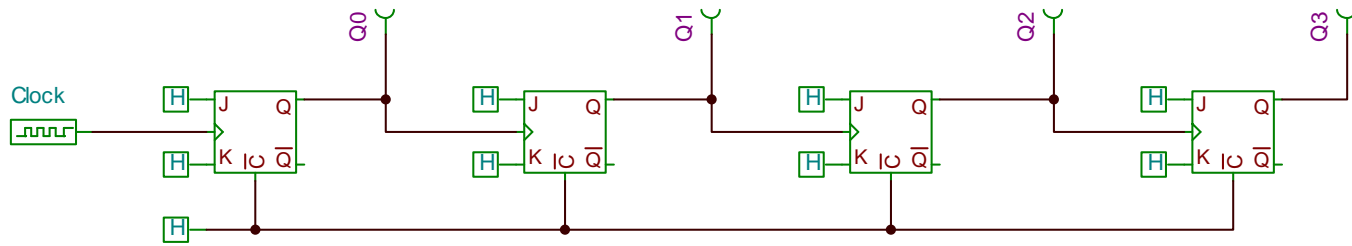
## ASZINKRON SZÁMLÁLÓK

Aszinkron számlálóknál biztosítani kell, hogy a flip-flop minden órajelre ellentétes állapotba billenjen. Ezt JK flip-flopnál  $J=K=1$ , T flip-flopnál  $T=1$ , D flip-flopnál  $D=Q_{negált}$  visszacsatolással lehet biztosítani.

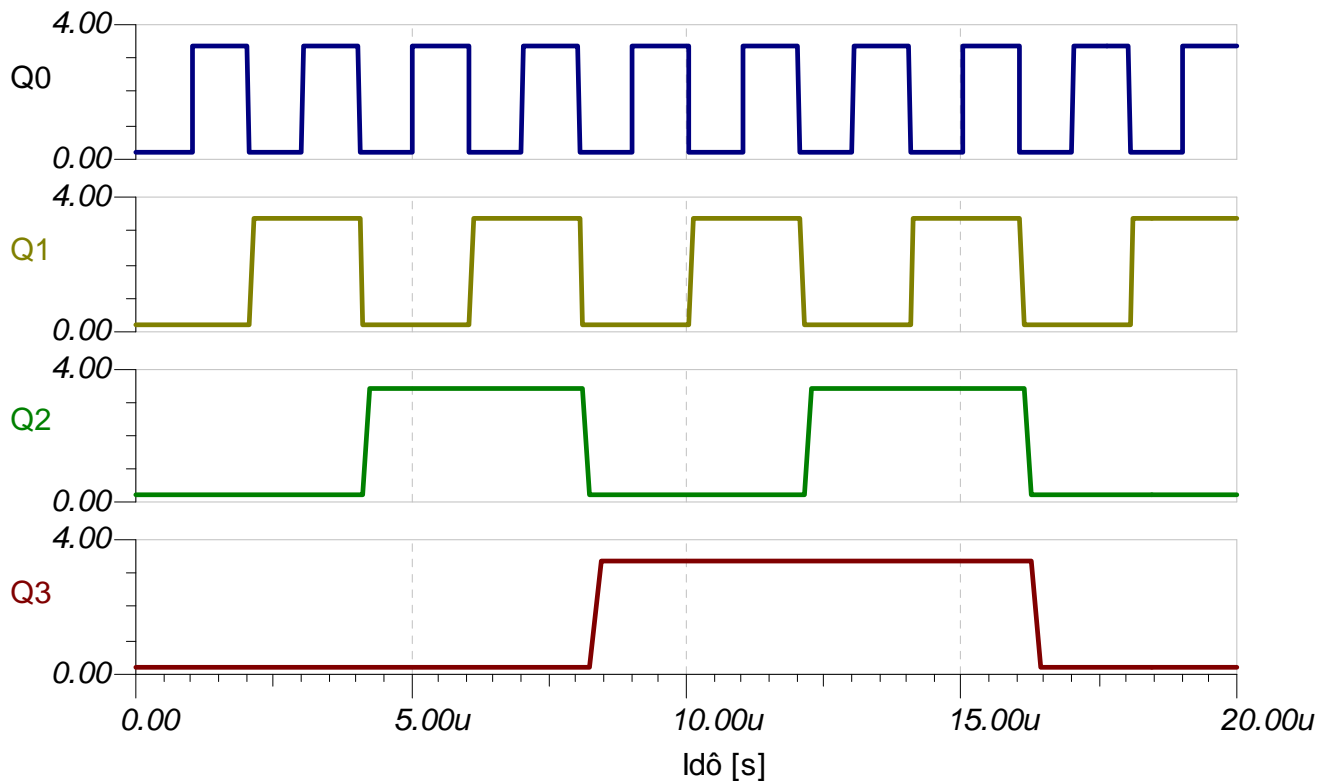
A számlálás iránya két dologtól függ:

- A számlálót felépítő flip-flop az órajel milyen (0-1 → felfutó él vagy 1-0 → lefutó él) átmenetére billen.
- A billentést az előző flip-flop Q vagy  $Q_{negált}$  kimenete végzi.

Az aszinkron számláló működését egy JK flip-flop-al megvalósított bináris előreszámlálón szemléltetjük.



10. ábra. Aszinkron számláló

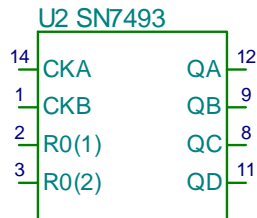


11. ábra. Aszinkron számláló idődiagram

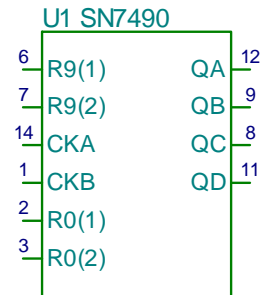
A működés az idődiagram alapján értelmezhető. A számláló 0-tól 15-ig számol bináris kódban. A JK flip-flop az órajel lefutó élére (1-0 átmenet) billen. A számlálás iránya megváltoztatható ha flip-flop órajel bemenetét ( $C_{ki}$ ) az előző flip-flop negált ( $\bar{Q}_{i-1}$ ) kimenetére kötjük.

**Aszinkron MSI számlálók:** Az aszinkron számlálókat nagyon gyakran alkalmazzák digitális áramkörökben. Ennek főleg egyszerű felépítésük és olcsóságuk az oka. Integrált kivitelben nagyon sok MSI áramkör áll rendelkezésre. A 74-es TTL sorozatok legnépszerűbb aszinkron számlálói a 7493-as és a 7490-es típus 12. ábra.

- 7493 TTL aszinkron bináris előreszámláló: A számláló egy 1 bites és egy 3 bites részből áll. A 4 bites működéshez a két egységet kaszkádosítani kell (az A flip-flop kimenetét,  $Q_A$  a B flip-flop bemenetével  $Ck_B$  össze kell kötni. A számláló törölhető az  $R0(1)$  és  $R0(2)$  Reset bemenetek segítségével. A Reset bemenetek ÉS kapcsolata hozza létre a nullázó állapotot. a.)ábra



a.,



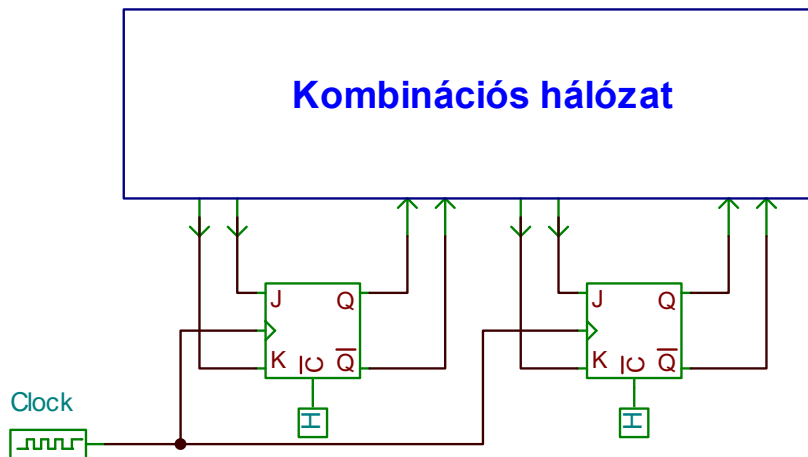
b.,

12. ábra. Aszinkron MSI számlálók

- **7490** TTL aszinkron bináris dekádszámláló: A számláló itt is két részből áll és kaszkádosítani kell ahhoz, hogy végigszámolja a teljes tartományt 0000-tól 1001-ig (4 bites BCD kód). A számlálónak kétféle törlési lehetősége van az  $R0$  bemenetek a számláló nullázását (0000 állapot), az  $R9$  bemenetek a számláló 9-be (1001 állapot) állítását végzik.

## SZINKRON SZÁMLÁLÓK

Szinkron számlálóokban az állapotváltozások egyidejűleg mennek végbe. A flip-flopok egyszerre, az órajellel szinkronizálva billenek így mentesek mindazoktól a problémáktól, amelyeket az aszinkron számlálók flip-flopjainak nem egyidejű billenése okoz. 13. ábra



13. ábra. Szinkronszámláló blokkvázlat

Az ábra alapján a szinkron számláló tervezése lényegében a flip-flopokat vezérlő kombinációs hálózat tervezésére vezethető vissza. A tervezés lépései a következők:

- A számlálási állapotoknak megfelelő állapot átmeneti tábla felvétele
- A flip-flopok vezérlési függvényének meghatározása.
- Realizálás

A szinkronszámlálók tervezését egy 3 bites előre számlálón szemléltetjük.

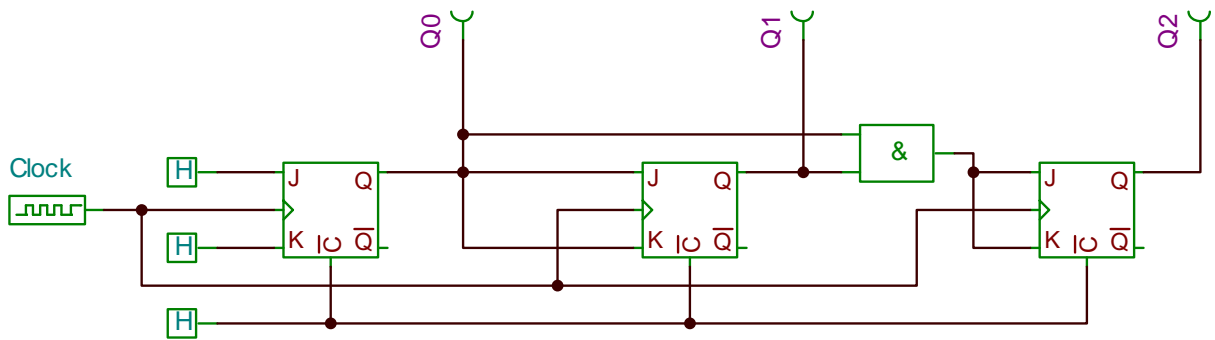
A számláló állapot átmeneti táblája:

n-dik állapot			(n+1) állapot			Flip-flopok					
Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	J <sub>2</sub>	K <sub>2</sub>	J <sub>1</sub>	K <sub>1</sub>	J <sub>0</sub>	K <sub>0</sub>
0	0	0	0	0	1	0	H	0	H	1	H
0	0	1	0	1	0	0	H	1	H	H	1
0	1	0	0	1	1	0	H	H	0	1	H
0	1	1	1	0	0	1	H	H	1	H	1
1	0	0	1	0	1	H	0	0	H	1	H
1	0	1	1	1	0	H	0	1	H	H	1
1	1	0	1	1	1	H	0	H	0	1	H
1	1	1	1	1	0	H	0	H	0	H	1

Az állapot átmeneti táblából V-K táblák (grafikus egyszerűsítés) segítségével kijövő vezérlési függvények:

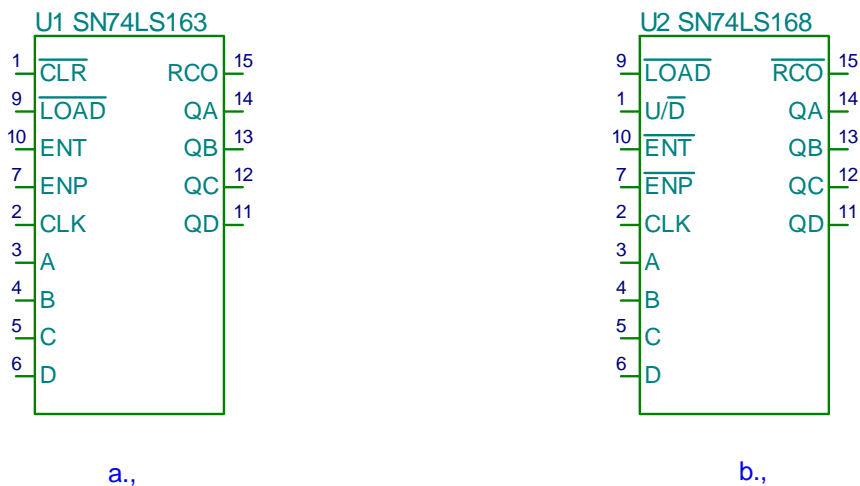
$$\begin{aligned}
 J_2 &= Q_1 * Q_0 & J_1 &= Q_0 & J_0 &= 1 \\
 K_2 &= Q_1 * Q_0 & K_1 &= Q_0 & K_0 &= 1
 \end{aligned}$$

A számlálást megvalósító szinkronszámláló kapcsolási rajza:



14. ábra 3 biteszinkron előre számláló

**Szinkron MSI számlálók:** Az aszinkron számlálókhoz hasonlóan 2 MSI szinkron számlálót mutatunk be. A 74163 bináris a 74168-as pedig decimális számláló 15. ábra.



15. ábra MSI szinkronszámlálók

Mindkét áramkör alkalmas párhuzamos beírásra. A párhuzamos beírás engedélyezését a LOAD bemenetek végzik. Az ENP és az ENT jelek a számlálást vezérlik. Az RCO kimeneteken megjelenő jelek az átviteli értékek, aminek a segítségével lehet egy másik számlálót az IC-hez kapcsolni (kaszkádosítani).

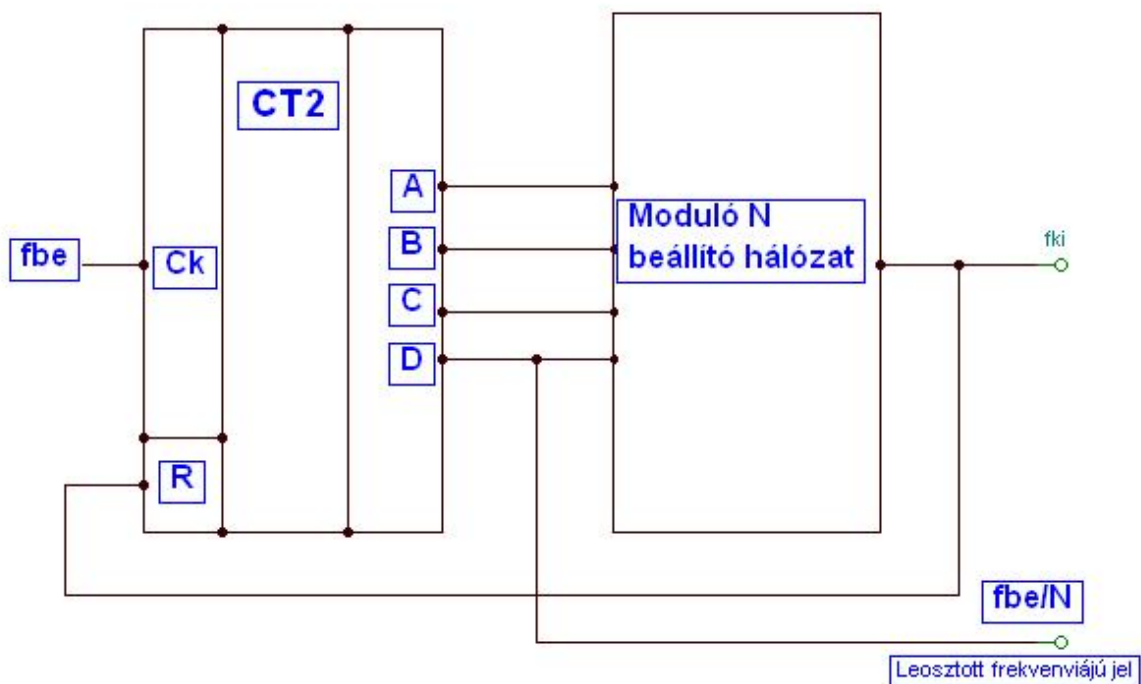
**Moduló fogalma:**

A számláló áramkörök mindegyike periodikus működésű, a pillanatnyi állapotukat bizonyos számú léptetés után ismét felveszik. Azt az órajel számot, ami után a pillanatnyi állapot ismétlődik a számláló *moduló* értékének nevezzük. A moduló érték azt mutatja meg, hogy hány különféle állapotot tud felvenni a számláló. Egy 3 bites számláló  $2^3$  állapotot vehet fel, így a számláló modulusa 8, 4 bites számlálónál ez az érték  $2^4=16$ .

Lehetőség van a számláló modulusának csökkentésére és növelésére is.

#### Számláló modulusának csökkentése:

Ezt általában akkor alkalmazzuk, ha csökkenteni szeretnénk a számláló állapotainak számát. Pl. 4 bites aszinkron BCD számláló esetén a számláló állapotait (modulusát) 16 állapotról 10-re csökkentjük. Ilyenkor egy áramkörrel figyeljük az számláló kimeneteit. Az áramkör kimenő jele Reseteli (nullázza) a számlálót. Ez a módszer **frekvenciaosztásra** használható. A leosztandó frekvenciát a számláló órajel bemenetére vezetik. A frekvenciaosztás mértékét a számláló állapotainak száma (modulusa) határozza meg. A leosztott frekvenciájú jel a legnagyobb helyi értéken áll elő 16. ábra.



16. ábra. Modulo N számláló

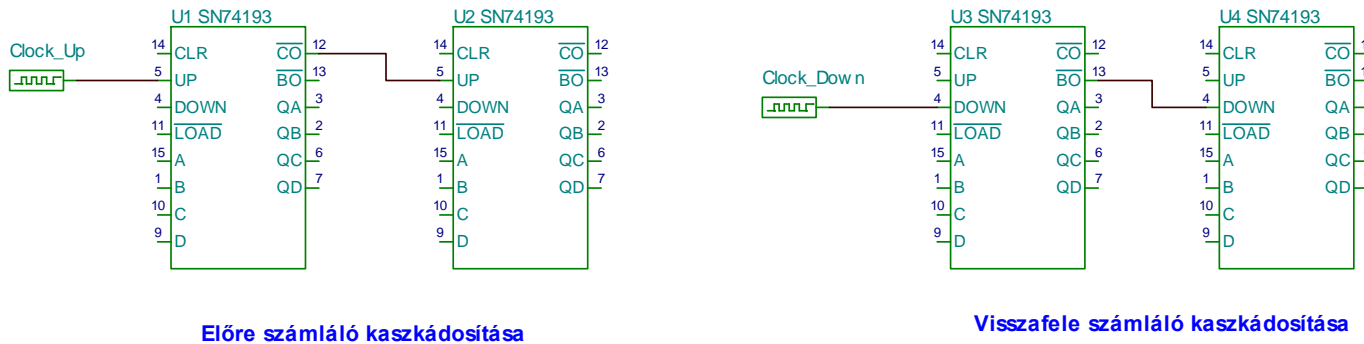
#### Számláló modulusának növelése:

Ezt akkor alkalmazzuk, ha a számláló állapotainak számát növelni szeretnénk. Ehhez több számlálót kell egymás után kapcsolnunk(kaszkádosítani kell a számlálót).

- Aszinkron számlálók kaszkádosításánál az egyik számláló legmagasabb helyi értékű kimenetét rávezetjük a másik bemenetére.



- Szinkronszámlálók – ha kaszkádosíthatók – külön kaszkádósító bemenettel és kimenettel rendelkeznek. Az 17. ábrán egy TTL szinkron, bináris UP-DOWN , PRESET számláló kaszkádósítása látható (74193)



17. ábra. MSI szinkronszámláló kaszkádósítása

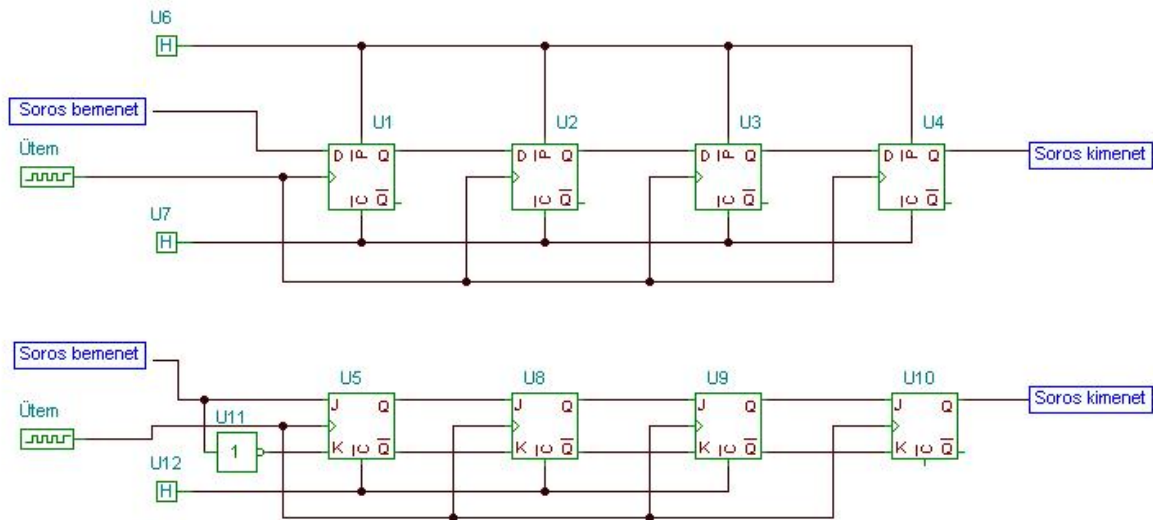
Külön órajel bemenete van a felfele (UP), és külön órajel bemenete van a visszafele (DOWN) számolásnak. Előre számlálásnál az átvitel a CO (Carry Output) kimeneten keletkezik . Előre számláló kaszkádósításánál a CO kimenetet kell a következő fokozat UP órajel bemenetéhez kötni. Visszafele számlálásnál az áthozat a BO (Borrow Output) kimeneten keletkezik. Az ilyen számláló kaszkádósításánál a BO kimenetet kell a következő fokozat Down órajel bemenetére kötni.

## LÉPTETŐREGISZTER (SHIFT REGISZTER)

### Működési elve

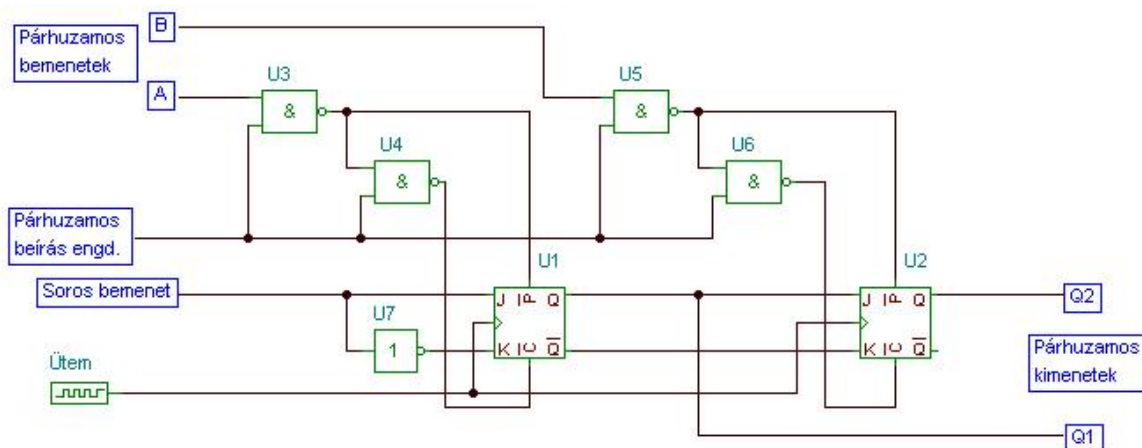
A Shift-regiszterek egymással oly módon összekapcsolt flip-flopok, hogy mindegyik kimenete a következő bemenetéhez csatlakozik. Az ütemimpulzus egyszerre jut el az összes flip-flopra. Legegyszerűbben közös órajellel vezérelt D vagy JK flip-flopok sorba kapcsolásával alakíthatók ki (18. ábra).

Az így kialakított lánc első tagjának bemenetére vezetett információt az első ütemimpulzus beviszi a tárolásra az első flip-flopba. A következő ütemimpulzus átlépteti (eltolja) a tárolt információt a Shift regiszter második flip-flopjába. Az első flip-flopba pedig új információ kerül. "n" ütemimpulzus elteltével a "n" lépést tesz meg a beírt információ a léptetőregiszterben. Egyes típusoknál a léptetés iránya is változtatható.



18. ábra. Soros beírású léptetőregiszter

A léptetőregiszterbe sorosan vagy párhuzamosan lehet beírni az adatokat. Soros beírás az első fokozat bemenetére vezetett jellel történik. Párhuzamos beírásra a flip-flopok aszinkron vezérlő bemeneteit használják (19. ábra).



19. ábra. Párhuzamos beírású léptetőregiszter

Ezek segítségével egyetlen lépésben feltölthető a regiszter tartalma. Regiszter kimeneteinek a kialakítása is kétféle lehet. Soros kimenet esetén a fokozat utolsó flip-flopjának kimenete alkotja a soros kimenetet. A párhuzamos kimenet az összes fokozat kimeneteinek egyidejű kezelésével valósul meg. A shift regiszter bemeneti és kimeneti megoldási lehetőségeiből adódik a shift-regiszterek alkalmazási lehetősége. A digitális rendszerekben az adatok sorosan és párhuzamosan állnak rendelkezésre. Előfordulhat, hogy szükség van a két adatformátum konvertálására. A shift regiszterek segítségével a párhuzamos – soros és a soros – párhuzamos kódátalakítás egyszerűen megoldható.

**Soros-párhuzamos átalakítás:** Az adat a shift-regiszter első bemeneti fokozatán keresztül (soros bemenet) lép be a regiszterbe. A teljes regiszter feltöltődése után az adat a párhuzamos kimenetekről egyszerre kiolvasható.

**Párhuzamos-soros kódátalakítás:** A párhuzamos bemenetek segítségével az adatokat egy lépésben vihetjük be majd az utolsó fokozat kimenetéről sorosan vehetjük ki.

A soros /párhuzamos átalakításon kívül a shift\_regisztereket felhasználhatják digitális jelek késleltetésére is és gyakran vezérlőregisztert alkotnak ezek az áramkörök.

**MSI SHIFT regiszter:** Mivel az MSI Shift regisztereket is 14, 16 kivezetéses tokokban gyártják csak néhány fokozatot lehet kialakítani. Hosszabb fokozat esetén a fokozatok egy részét nem vezetik ki az áramkörből.

A **74LS91 SISO** típusú (Soros bemenetű – Soros kimenetű) Shift regiszter. A soros Bemenet két jel **ÉS** kapcsolataként lett kialakítva.

A **74LS164 SIPO** (Soros bemenetű – Párhuzamos kimenetű) Shift regiszter. A soros bemenet itt is két bemenet **ÉS** kapcsolataként lett kialakítva.

A **74LS165 PISO** (Párhuzamos bemenetű-soros kimenetű) Shift regiszter. Külön vezérlő jele van a Soros vagy Párhuzamos bemenetek engedélyezésének SH/LD.

## MEMÓRIÁK

A memória sok egyidejű adat tárolására és gyors kiolvasásra alkalmas elemek. Lényegében regiszterekből épülnek fel. Legkisebb egysége a memóriacella ,ami egy bit tárolására szolgál. Több memóriacella alkot egy memóriarekeszt.

A memóriák olyan tárolóelemek, ahol az Address (címvezetékek) segítségével kiválasztott memóriarekeszek tartalmát a C (Control) vezérlőjeltől függően a DATA (adat) adatkimenetre juttathatjuk (olvasás), vagy írás esetén a DATA adatvonalak tartalmát a kiválasztott rekeszbe tölti.

**A memóriákon előforduló tipikus jelek:**

**Adat és cím:**

- Cím bemenetek ( $A_n-A_0$ ): Az adat címének kijelölésére
- Adat Ki/bemenetek ( $D_m-D_0$ ): Az adatok beírására, kiolvasására

**Vezérlőjelek:**

- **CS** (chip select) IC kiválasztó bemenet a memória engedélyezésére.
- **OE**(output enable) vagy **RD** (read) az olvasás engedélyezésére.
- **WR** (write) írás engedélyező bemenet.

**Memóriák jellemzői:**

**Memóriakapacitás:** A memóriarekeszek számát jelenti és lényegében és lényegében a memóriában tárolható adatok mennyiségét határozza meg.

A rekeszek számát a címvonalak száma határozza meg. Ha a címvonalak száma  $n$ , a megcímezhető memóriarekeszek száma  $2^n$ .

**Kapacitás** = memóriarekeszek száma \* memóriarekeszek szélessége.

Ez a megadási mód utal a memória szervezésére is.

Pl.  $1K*8bit$  jelentése : A memória egy sora (rekesze) 8 bites és  $1kByte = 1024$  rekesze van.

A rekeszek méretére az adatvonalak száma utal.

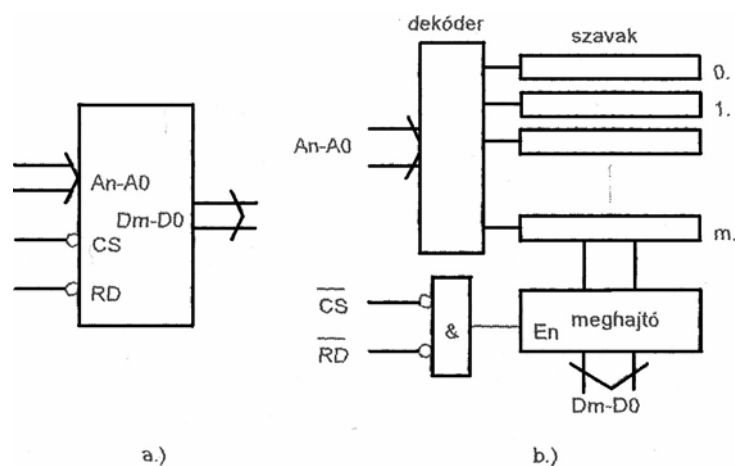
**Elérési idő (access time)** :A cím kiadásától az adat rendelkezésre állásáig eltel időt értjük.

Ez az áramkör gyorsaságára utal. Megmutatja, hogy mennyi időt vesz igénybe egy adat kiolvasása.

A memóriákat az alapján, hogy csak írható vagy írható és olvasható is **ROM –ra** (Read Only Memory) és **RAM-ra** (Random Access Memory) oszthatjuk.

**ROM**

A ROM-ok un. csak olvasható memóriák.



20. ábra. ROM memória<sup>1</sup>

<sup>1</sup> Forrás : Benesóczky Zoltán: Digitális tervezés funkcionális elemekkel és mikroprocesszorral Műegyetemi kiadó, 2008

Leggyakrabban mikroprocesszoros rendszerekben nem megváltoztatható programok és konstans adatok tárolására ill. univerzális kombinációs hálózatként használják. A ROM-ok jelölése és egyszerűsített blokk vázlata a 20. ábrán látható. Az An–A0 címbemenetek segítségével választható ki a memória egy-egy rekesze. A megcímzett memóriarekesz tartalom akkor jelenik meg a kimeneten, ha a CS bemenet és a RD bemenet egyszerre aktív.

### ROM, min univerzális kombinációs hálózat

ROM segítségével tetszőleges kombinációs hálózatot is megvalósíthatunk. Ilyenkor a ROM címbemenetei a kombinációs hálózat bemeneteinek felelnek meg. A kombinációs hálózat kimeneteit a ROM adat kimenetein kapjuk. A ROM-ba a kombinációs hálózat igazságtábláját kell beégetni.

Az alábbi táblázat egy 3 bites szavazó áramkör megvalósítását mutatja.

A megvalósított logikai függvény  $F = A_{negált}BC + AB_{negált}C + ABC_{negált} + ABC$

Az alábbi memóriával 4db 4 változós logikai függvényt lehetne megvalósítani. A példa esetében a 3 változó a memória A2, A1 és A0 címbemeneteit címzi. Jelen esetben A3 felesleges így értéke A3=0. Ezzel a jelen esetben a memória alsó 8 sorát használjuk csak ki.

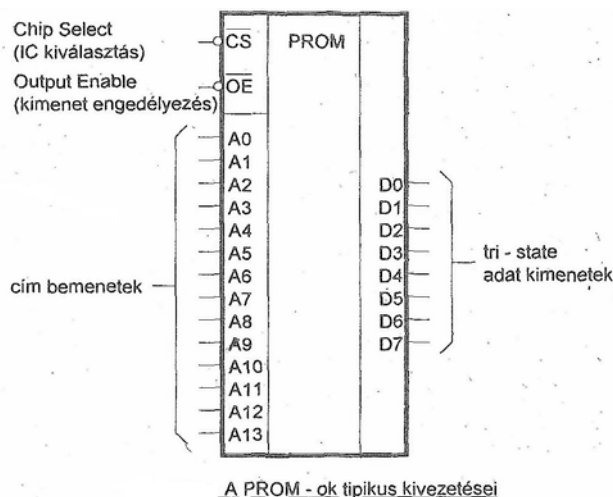
A függvény kimenő jelét a DO adatkimenet adja.

Cím	D3	D2	D1	D0	A3
0.	X	X	X	0	A2
					A1
1.	X	X	X	0	A0
2.	X	X	X	0	
3.	X	X	X	1	
4.	X	X	X	0	
5.	X	X	X	1	
6.	X	X	X	1	
7.	X	X	X	1	
8.	X	X	X	X	
.	.	.	.	.	
.	.	.	.	.	OE
					CS
15.	X	X	X	X	

**PROM** (*programozható ROM*)

Olyan ROM típusú memória ami a felhasználó által egyszer programozható. A PROM elem olyan mátrixot tartalmaz aminél a mátrix minden egyes keresztpontjában a diódával sorba egy olvadó biztosítékot építenek. Programozáskor ezeket a biztosítékokat megfelelő nagyságú árammal átégetik. A folyamat nem reverzibilis így a PROM-ot csak egyszer lehet programozni.

A PROM-ok tipikus jelei a 21. ábrán láthatóak.



21. ábra. PROM IC<sup>2</sup>

Működését az alábbi igazságtáblázat alapján tanulmányozhatjuk

**Cs OE működés**

0	0	
		olvasás
0	1	kiválasztás
1	X	nyugalmi állapot

**EPROM** (*Electrically Prormmable ROM*)

---

<sup>2</sup> Forrás: Szűcs László: Digitális számítógépek példatár Székesfehérvár,1996

Elektromosan programozható és speciális hullámhosszú UV fényel törölhető. Ezeket kizárólag MOS technológiával gyártják. A tartalom bevitele töltések injektálásával történik, ez a folyamat megfordítható. Ez ad lehetőséget tartalmuk törlésére, majd újraprogramozásukra. Csak a teljes tartalom törölhető. Külön sorokat (memóriarekeszeket) nem tudunk törölni. Az EPROM-ok programozását külön erre a célra kifejlesztett EPROM programozóval lehet elvégezni. A programozáshoz a TTL áramköröknél megszokottnál nagyobb (12.5V–21V) feszültségre van szükség.

Az EPROM-ok tipikus jelei megegyeznek a PROM-okéval. Nyugalmi állapotban (standby) az áramkörök fogyasztása normál működéshez képest töredékére csökken. Ez különösen a CMOS típusokra igaz. Pl. a 27C256 EPROM 40 mW helyett mindössze 0,5mW-t fogyaszt nyugalmi állapotban.

### **EEPROM** (*Elektromosan törölhető PROM*)

Adatai elektromosan törölhetőek. Lehetővé válik hogy egyszerre ne csak az egész tartalma, hanem csak egyes rekeszeket töröljünk. A RaM-októl abban térnek el, hogy a programozás és a törlés körülményes folyamat (több típusnál külön feszültséget igényel) az olvasásánál sokkal lassabb (10–ms–50 ms) és csak korlátozott számban ismételhető ( $10^4$ – $10^5$ ). A beírási vagy törlési folyamatot egy az IC-be épített vezérlő végzi. Mivel a beírás és a törlés lassú és újabb adatot csak az előző beégetése után lehet beírni, ezért a beírás vagy törlés végét egy READY jellel jelzi az áramkör. Ezt a jelet kell figyelni a beégetést végző egységnek.

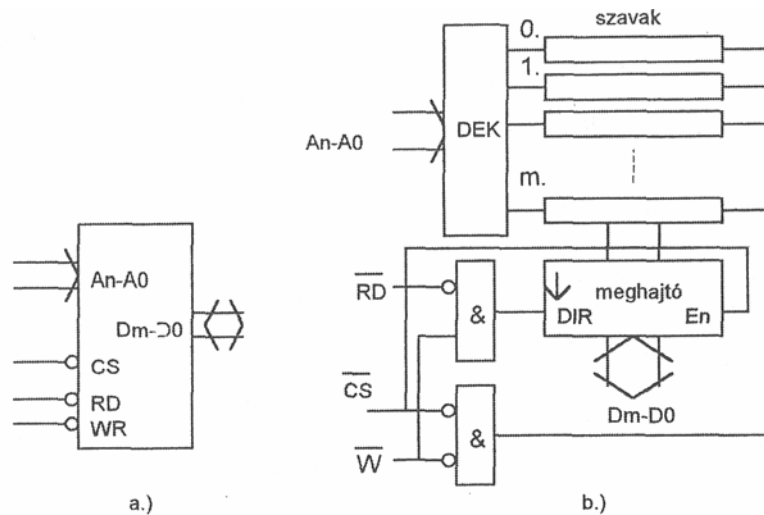
## **RAM**

A RAM (Random Access Memory)– véletlen elérésű táruk vagy más néven írható – olvasható táruk. Nevüket onnan kapták, hogy egy adat kiolvasásának és beírásának ideje nem függ sem a művelet helyétől sem az időpontjától. Legfontosabb jellemzőjük, hogy írhatók és olvashatók. Tartalmukat a tápfeszültség megszűnésekor elvesztik. Két alapvető típusa van az SRAM és a DRAM.

### **Statikus RAM**

Az SRAM-ban az elemi cellákat mátrix alakzatba kötött elemi flip-flopokkal valósítják meg. Működési elvéből adódóan nincs szükség frissítő áramkörre, mint a DRAM-nál. A nagyobb Chip felület igény miatt méretük kisebb mint a DRAM-é. Az SRAM-ok mérete 4Mbites nagyságrendű. Hozzáférési idejük kb. 5–150 nsec. Fogyasztásuk rendkívül kicsi.

Az SRAM IC-k belső funkcionális felépítése látható a 22. ábrán



22. ábra. SRAM IC felépítése<sup>3</sup>

Az  $A_n-A_0$  címbemenetek segítségével választható ki a memória egy-egy rekesze. A megcímzett memóriarekesz tartalom akkor jelenik meg a kimeneten, ha a CS bemenet és a RD bemenet alacsony, a WR magas szintű. Írni a Cs és WR bemenet alacsony és a RD bemenet magas szintjénél lehet.

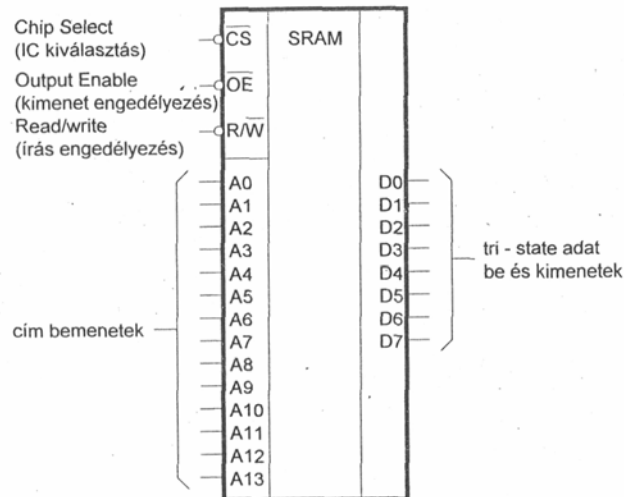
A vezérlő jeleinek működését az igazság tábla tartalmazza.

CS	OE	R/ $\bar{W}$	Működés
0	0	0	Tiltott
0	0	1	Olvasás
0	1	0	Írás
0	1	1	Nincs művelet
1	X	X	Nyugalmi állapot

Egy SRAM IC tipikus kivezetései láthatók a 23. ábrán

<sup>3</sup> Forrás: Benesóczky Zoltán: Digitális tervezés funkcionális elemekkel és mikroprocesszorral Műegyeremi kiadó, 2008





23. ábra. SRAM IC<sup>4</sup>

Nyugalmi állapotban az áramkör fogyasztása a normál működéshez képest töredékére csökken.

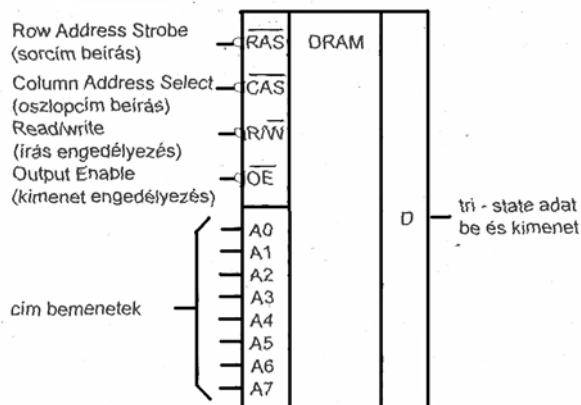
### Dinamikus RAM

A DRAM-okban az egyes memóriacellákat mátrix alakzatba kötött MOS tranzisztorokkal valósítják meg.

A MOS tranzisztor tulajdonképpen egy kapacitást képvisel, ahol az adat csak korlátozott ideig tárolható. A DRAM-ok tartalmát időről időre frissíteni kell. Ez a DRAM-ok használatát bonyolultabbá teszi. Frissítéskor általában 2ms-on belül minden sorhoz tartozó vezetékot aktiválni kell. Előnye az SRAM-al szemben, hogy ugyan olyan chip felületen, ugyanakkora áramfelvétel mellett kb. négyszeres kapacitás érhető el.

Egy DRAM IC tipikus kivezetései az 24. ábrán láthatók.

<sup>4</sup> Forrás: Szűcs László: Digitális számítógépek példatár

24. ábra. DRAM IC<sup>5</sup>

A DRAM-ok címzése két részben történik. Ugyan azokra a címbemenetekre kell ráadni a cím alsó (sor cím) és a cím felső (oszlop cím) részét. A sor cím a RAS (Row Address Select), az oszlop cím a CAS (Column Address Select) jel hatására íródik be a DRAM címregiszterébe.

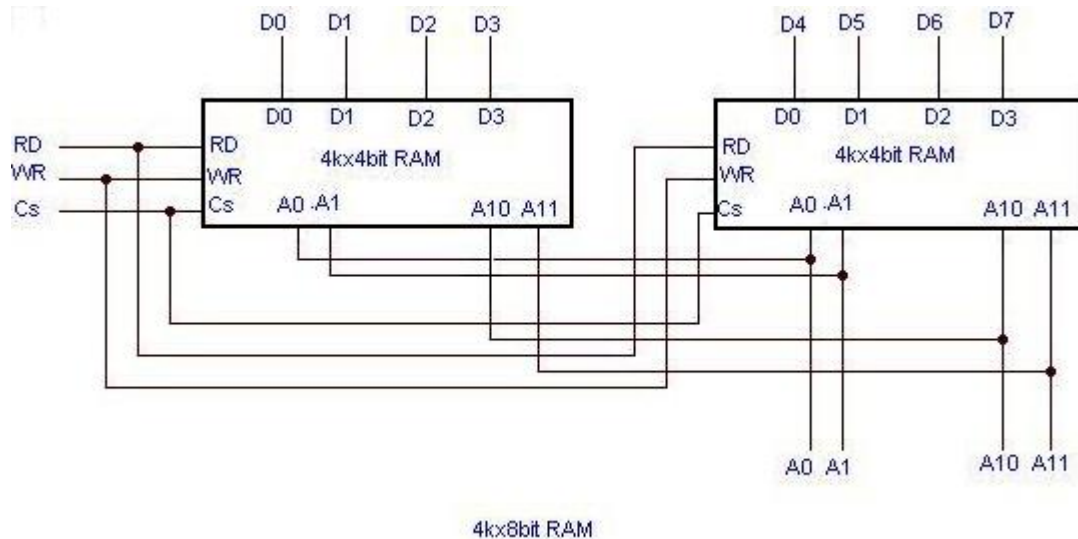
### Memóriák szélességének növelése

Ha a memóriarekeszek hossza nem megfelelő, akkor egyszerűen megnövelhetjük azt. Ilyenkor a memória cím, vezérlő és engedélyező bemeneteit párhuzamosan kötjük. Az adat kimenetek pedig sorosan kapcsolódnak az adatbuszra. A 25. ábrán 2db 4kx4bit-es RAM-ból készítünk 1db 4kx8bit-es RAM-ot.

Az A0... A11 címbemeneteket valamint a RD, WR vezérlő és Cs engedélyező jeleket mindkét RAM egyszerre megkapja. A D0... D3 adatvezetékek külön-külön kapcsolódnak a nyolcbites adatbuszra.

A RAM a nyolcbites adatot két részben tárolja el. Az egyik 4 bitet az egyik RAM a másik 4 bitet a másik RAM tárolja ugyan azon a címen. Mivel mindkét memória ugyan azt a címet kapja és az engedélyezés is egyszerre történik, a memóriák kimenetén egyszerre jelenik meg nyolcbites adatként a két négybit-es adat.

<sup>5</sup> Forrás: Szűcs László: Digitális számítógépek példatár



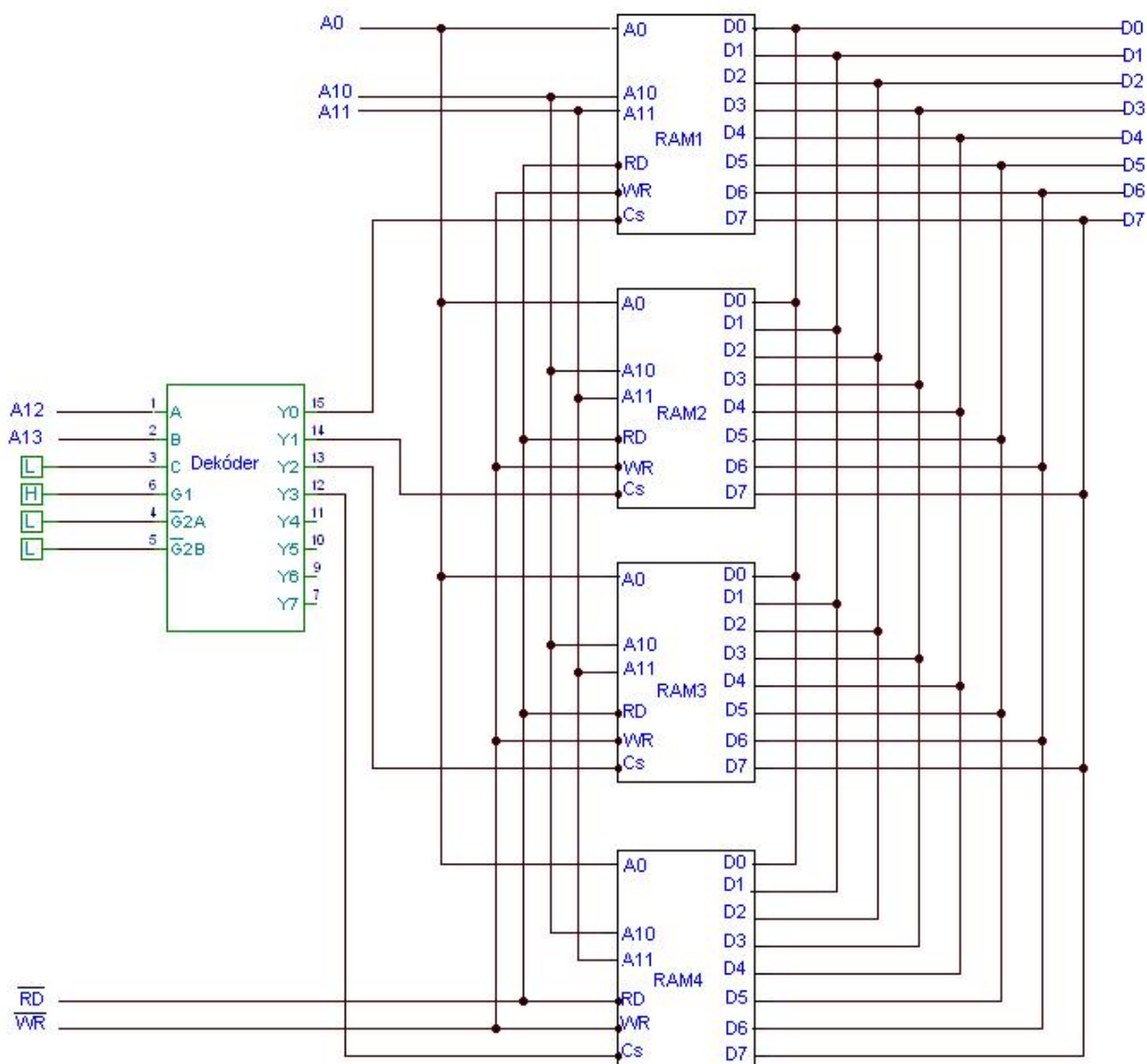
25. ábra. Memória szélesség növelés

### Memóriák kapacitásának növelése

Ha a memóriák kapacitása (mérete) nem elég, akkor megnövelhetjük azt több memóriamodul használatával. Ilyenkor mindegyik memóriamodul ugyan azt a címet kapja és az adatvezetékeik is ugyan arra az adatbuszra kapcsolódnak. Azt, hogy melyik memóriára vonatkozik a cím, egy dekóder segítségével lehet kiválasztani. A dekóder kimenete engedélyezi a kiválasztott memória működését. A dekóder kimenetek kiválasztását és engedélyezését azok a címbitek végzik, amiket a processzor a memória tokok címzéséhez nem használ.

Az 26. ábrán egy 16kx8bit RAM-ot készítünk 4kx8bit RAM-ból. A szükséges memória kapacitást 4db memória modul alkalmazásával érhetjük el. Mindegyik RAM megkapja a processzor A0-A11 címvezetékét. A memória tokok kiválasztását a dekóder q0-g3 kimenetei végzik a RAM-ok CS IC kiválasztó bemeneteinek segítségével. A dekódernek a 8 kimenete közül csak négyet használunk. Így a jelen memória rendszer kialakítás még 4db RAM bővítést tesz lehetővé.

Mivel az alsó négy kimenetét használjuk csak a dekódernek, a kimenetek kiválasztásához elég két címvezeték A12, A13. A dekóder harmadik kiválasztó bementét log. 0 szintre köthetjük. A dekóder engedélyezését a jelen esetben log. 0 illetve log.1 szintekkel oldottuk meg. Ha a processzornak még van olyan címbemenete, ami a címrésben nem vett részt, akkor azt a dekóder engedélyezésére használjuk. (Teljes címdekódolás).

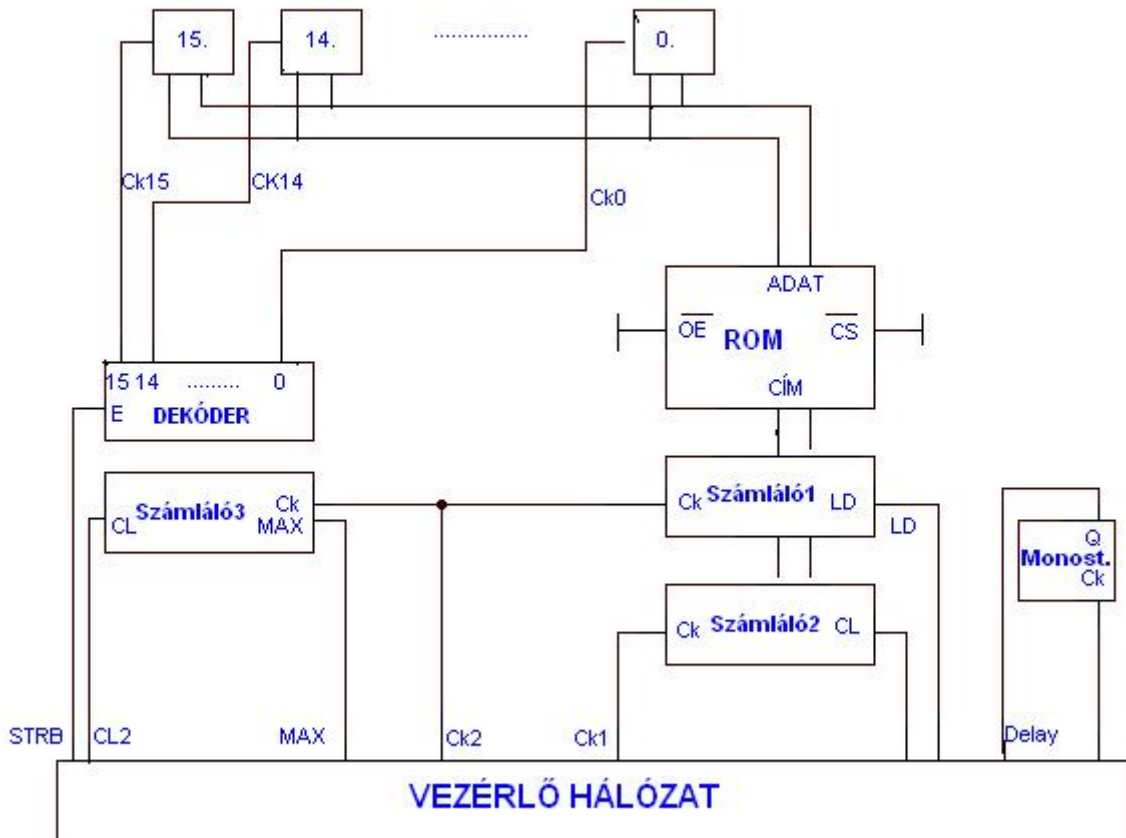


26. ábra. Memóriakapacitás növelése

## A FÉNYŰJSÁG FUNKCIONÁLIS BLOKKVÁZLATA

A fényűjság egy 1kbyte-os ROM-ban tárolt ASCII kódolású szöveget jelenít meg ciklikusan. A ROM egy-egy sora (memóriarekesze) a fényűjság egy-egy karakterét tárolja. A megjelenítendő ASCII karakterek a ROM egy-egy sorában, 8 biten vannak tárolva. (Az ASCII kód 8 bites.) Ezeket a karaktereket a 16 db ASCII dekódert és felfutó élre aktív tároló regiszttert tartalmazó LED kijelzőn jeleníti meg. A szöveget másodpercenként 1-gyel balra lépteti.

## KIJELZŐK



27. ábra. Fényújság blokkvázlata<sup>6</sup>

A 2-es számláló jelöli ki a 16 byte hosszú megjelenítendő szöveg kezdőcímét az 1kbyte-os ROM memóriában. Ez a számláló 1 sec-onként lép, az időzítés leteltét egy monostabil multivibrátor jelzi. A 2-es számláló által tárolt kezdőcímtől kezdve a memóriát az 1-es számláló címzi, 16 lépésben. Az 1-es számláló jelöli ki a memóriának azokat a rekeszeit a melyek a fényújság megjelenítendő ASCII karaktereit tárolják. Az 1-es számlálással párhuzamosan működő 3-as számláló választja ki azt a kijelzőt, amelynek megfelelő szöveg-ablakon belüli karakter az adatbuszon megjelenik. Ez az érték íródik át az STRB impulzus hatására a kijelzőbe. A 16-os kijelzési ciklus végén, amit a max jel jelez, a monostabil újraindul. Az időzítés leteltekor a 2-es számláló lép és az egész előről kezdődik.

<sup>6</sup> Forrás: Dr. Selényi Endre-Benesóczky Zoltán: Digitális technika példatár BME, Budapest, 1991

## TANULÁSIRÁNYÍTÓ

A témakörhöz tartozó ismeretek gyakorlati alkalmazásához szükséges az alábbi készségek, képességek fejlesztése:

- írott szakmai szöveg megértése
- katalógus használat,
- elemző képesség,

A témakörhöz tartozó ismeretek gyakorlati alkalmazásához szükséges az alábbi személyes (Sze), társas (Tá), módszer (Mó) kompetenciák fejlesztése:

- logikus gondolkodás (Mó)
- áttekintő képesség (Mó)

### Javasolt tanulói tevékenységforma az ismeretek feldolgozásához

- Ismerje meg a logikai függvények építőelemeit!
- Szerezzen jártasságot a kombinációs hálózatok tervezése területén. Ehhez olvassa el Szűcs László Digitális áramkörök c. könyvének ide tartozó fejezetét (47–75 oldal)!
- Szerezzen jártasságot a szekvenciális hálózatok tervezése területén. Ehhez olvassa el Szűcs László: Digitális áramkörök c. könyvének ide tartozó fejezetét(109–175.oldal )!
- Tanulmányozza át a tananyagot!
- Elemezze az egyes funkcionális egységek működését valósítsa meg azokat SSI elemek segítségével!
- Tanári irányítással funkcionális áramkörök segítségével valósítson meg egyszerű áramköri kapcsolásokat!
- Oldja meg az önellenőrző feladatokat! Ellenőrizze megoldását a megoldási útmutató alapján!

## ÖNELLENŐRZŐ FELADATOK

### 1.feladat

Funkcionális elemek felhasználásával tervezzen olyan áramkört, ami egy 4x4-es billentyűzetmátrixban elhelyezkedő nyomógomb megnyomásának hatására egy a regiszterbe beír egy a billentyűre jellemző kódot és, és egy kimeneten jelzi, hogy a regiszterben érvényes adat van! A jelzés egy RD jellel törölhető. Rajzolja meg az áramkör blokkvázlatát! Magyarázza a működést!

---

---

---

---

---

**2.feladat**

Tervezze meg egy programozható számú, 50 %-os kitöltési tényezőjű impulzussorozat előállító generátor funkcionális blokkvázlatát! Az impulzusszám 1–256. Az impulzussorozatot egy külső jel programozhatóan fel, vagy lefutó éle indítsa!

---

---

---

---

---

**3.feladat**

Tervezze meg egy olyan digitális háromszöggenerátor funkcionális blokkvázlatát, amelynek 4 bites kvantálással programozható az amplitúdója!



---



---



---



---



---



---



---



---



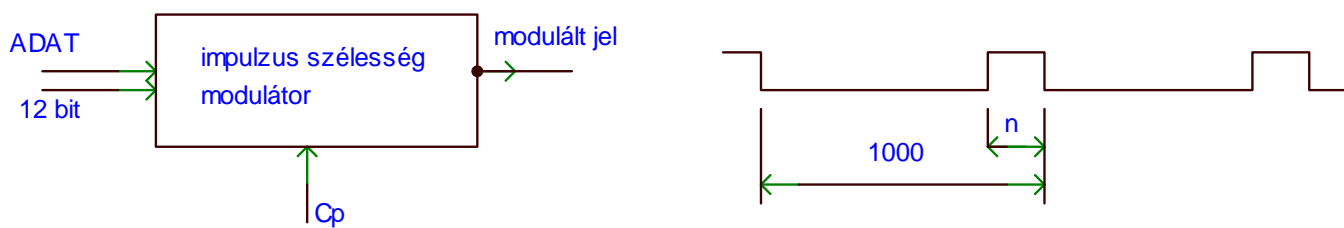
---



---

**4.feladat**

Tervezzen programozható impulzus szélesség modulátort!



28. ábra. Impulzus szélesség modulátor

A bemeneti adatot 3 NBCD számjeggyel adjuk meg.



### 5. feladat

Tervezze meg egy olyan beállítható jelformájú függvény generátor blokkvázlatát, aminek hullámformáját saját magunk programozhatjuk (állíthatjuk) be! Írja le a működést!



A large empty rectangular box with a yellow border, intended for drawing or writing. It contains five horizontal lines near the bottom.

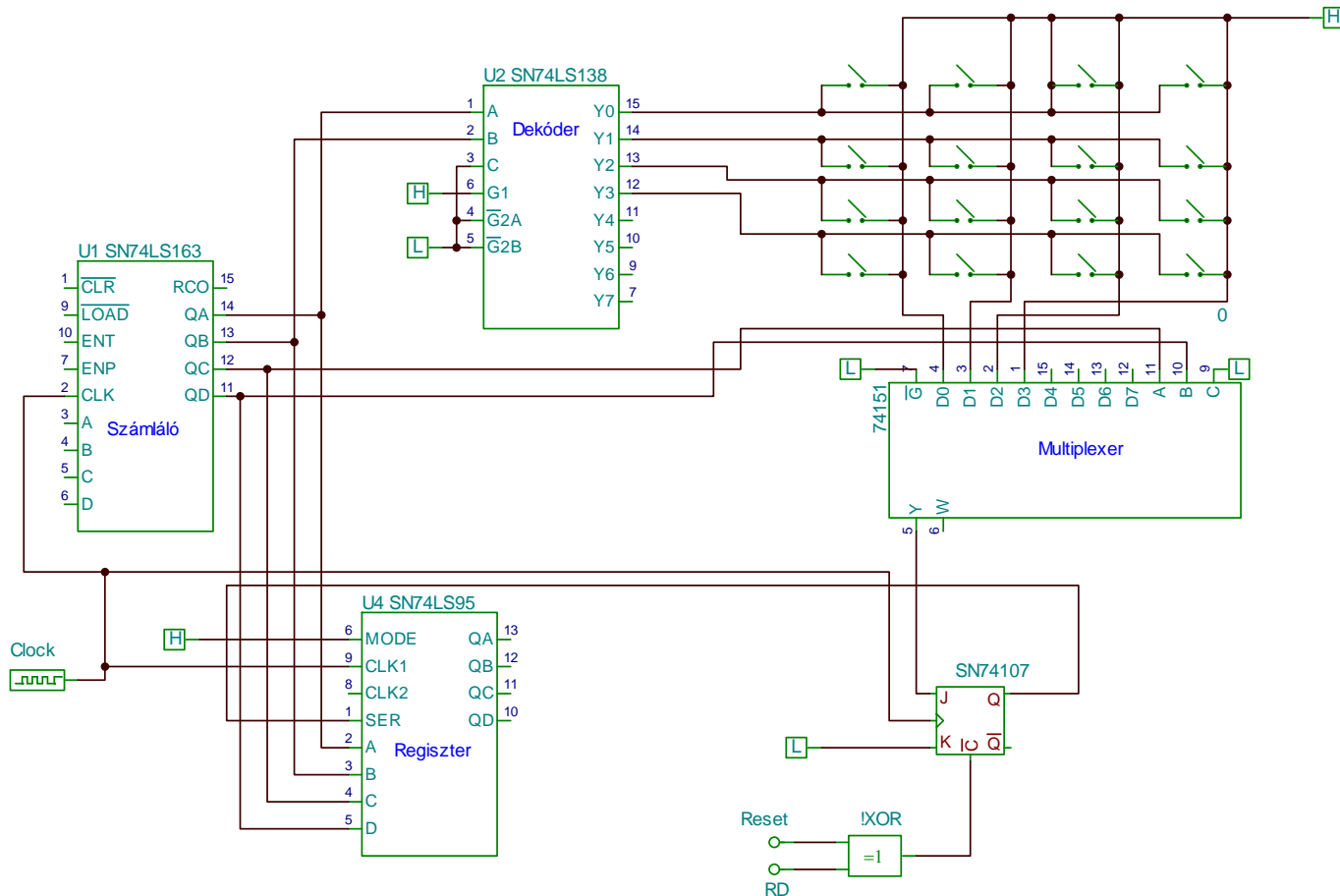
MEGOLDÁSOK

1.feladat

A billentyűzetmátrix minden egyes keresztezési pontjában van egy-egy kapcsoló.

A 4-bites bináris számláló kimenetei egy multiplexert és egy dekódert vezérelnek.

A számláló Qc, Qd kimenetei a multiplexert címzik. A kiválasztott multiplexer bemenetek egy felhúzó ellenálláson keresztül tápfeszültségre kapcsolódnak. Így a kiválasztott bemenet log1 szintet kap.



29. ábra. Billentyűzetmátrix

A számláló  $Q_a$ ,  $Q_b$  kimenetei egy dekódert címeznek. A dekóder kimenetek a billentyűzetmátrix sorait választják ki. A billentyűzetmátrix kiválasztott sora log.0-ra kerül. Azokon a helyeken ahol a billentyű lenyomásra került a kapcsoló zár és logikai 0 szintre viszi a multiplexer bemenetét. A multiplexer kimenő jelét egy JK tároló (flip-flop) mintavételezi.

A billentyű lenyomásának hatására a flip-flop kimenetén log.1 jelenik meg amivel egy regiszter beírást engedélyezünk.

A regiszter a számláló aktuális állapotát írja be, ami a lenyomott billentyű bináris kódja. A regiszter kimenő jele jelzi, hogy a regiszterben érvényes adat van. Az RD jel hatására a Flip-flop Q kimenete nullázódik és ezzel a regiszter kimenetén megjelenő jelzés törölhető.

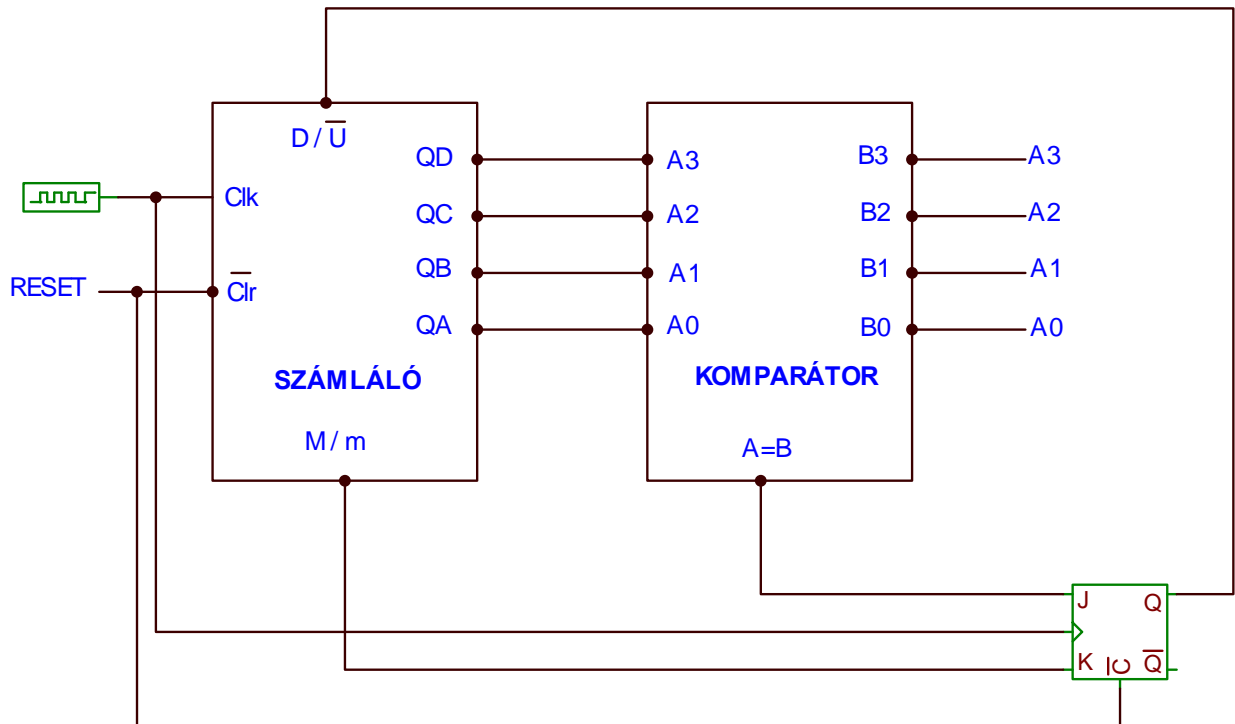
---

### 2.feladat

A nyolcbites számláló kimenő jelét egy komparátor figyeli aminek a másik bemeneteire beállítjuk a szükséges impulzusszám-1- t. A komparátor kimenő jele egy JK flip-flopot vezérel. Amikor a számláló kimenete eléri a beállított szintet a komparátor a JK flip-flopon keresztül letiltja az impulzus sorozat engedélyezését a kimeneten. Az impulzus sorozat engedélyezése a JK flip-flop J (beíró bemenete segítségével) történik.

A JK flip-flop engedélyezését egy VAGY kapun keresztül a D flip-flopok végzik. A külső jel programozható fel- vagy lefutó élének indítását egy kizáró VAGY (XOR) kapu végzi.





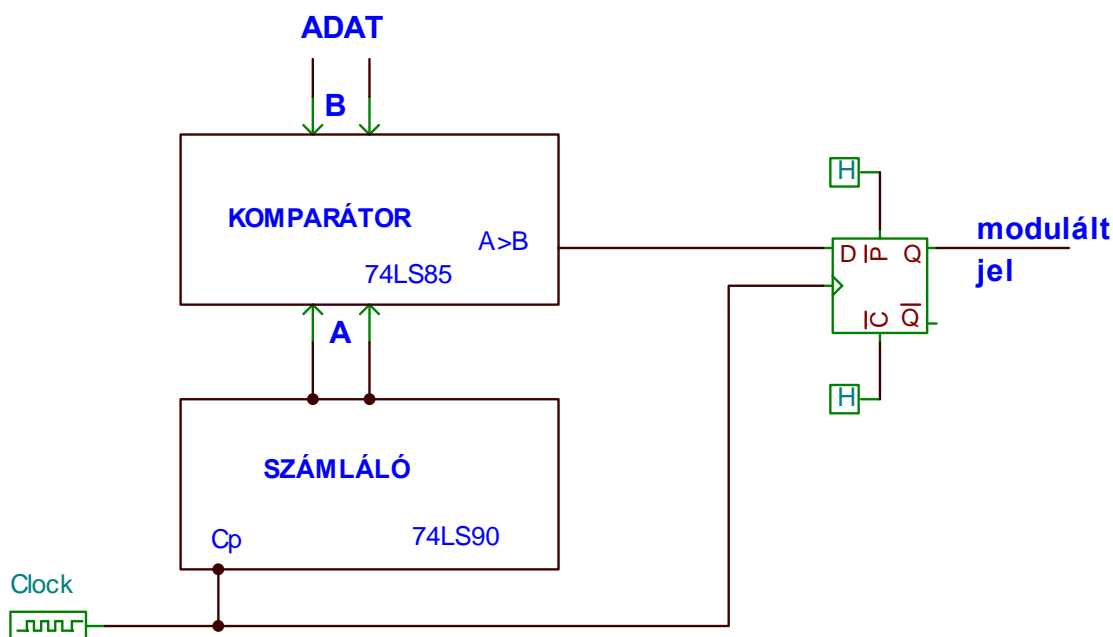
31. ábra. Háromszög generátor<sup>8</sup>

#### 4.feladat

A három dekádos NBCD számláló 0–999-ig számol. A számláló kimenő jelét összehasonlítjuk egy komparátor bemenetén beállított értékkel. A komparátor kimenetén megjelenő jel egy D flip-flopot vezérel. Abban a pillanatban, amikor a számláló kimenő jele meghaladja a komparátor bemenetén beállított számértéket a komparátor A>B kimenetén megjelenik egy log.1 szint. Ez a D flip-flop kimenetén is log1 szintet eredményez. A kimenet addig marad log.1-ben, amíg a számláló értéke 999 nem lesz. Ekkor a számlálás kezdődik előről. A számláló kimenete kisebb lesz a beállított értéknél és a komparátor A>B kimenetén log. 0 szint jelenik meg és így a flip-flop kimenő jele is log.0 szintű lesz.

Ha a komparátor bemenetén beállított szám  $n$ , akkor a impulzus  $1000-n$  értékű lesz. Az impulzus idejét megkapjuk, ha ezt a számot megszorozzuk a számláló órajel frekvenciájának periódusidejével.  $T_i = T_{clock} * (1000-n)$

<sup>8</sup> Forrás: Dr. Selényi Endre–Benesóczky Zoltán: Digitális technika példatár BME,Budapest,1991



32. ábra. Programozható impulzus szélesség modulátor<sup>9</sup>

## 5. feladat

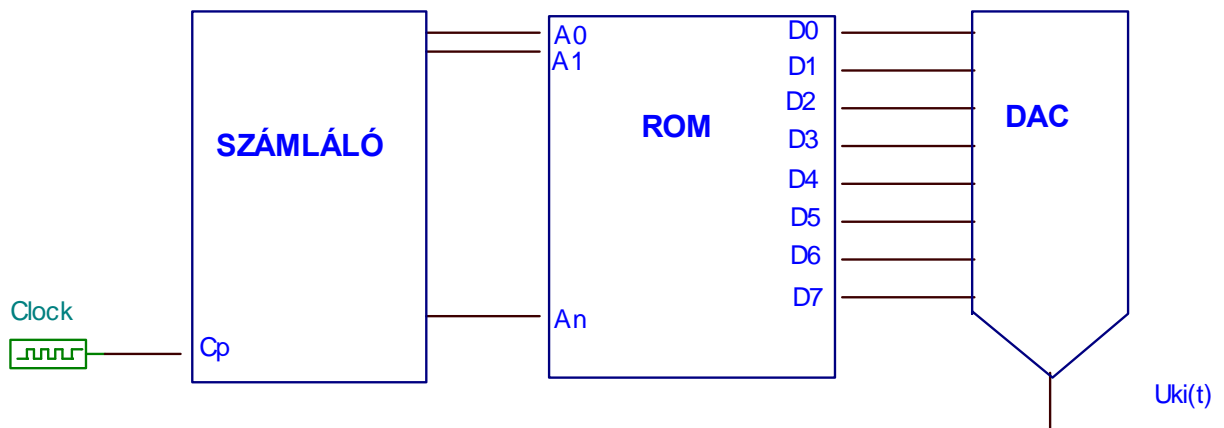
Programozható jelformájú függvénygenerátort a legegyszerűbben digitális módszerekkel állíthatunk elő. A legegyszerűbben egy EPROM-ba programozhatjuk (égethetjük be) a kívánt jelformát. A programozás a jel függvénytáblázatának megfelelően történik.

Az EPROM kiolvasása egy számláló segítségével történik. Az órajel generátor hatására a számláló egyel tovább lép megnövelve ezzel a kiolvasott memória rekesz címét. Az adott címeken tárolt digitális függvényértékek az adatkimeneteken jelennek meg egymás után.

Ezekből a digitál – analóg átalakító (DAC) állít elő analóg hullámformát. Ez a kvantálás miatt bizonyos mértékig "lépcsős". A kimenő jel lépcsőzetességét csökkenthetjük, ha megnöveljük a lépcsők számát. Minél több bites a DAC, minél több bitre bővítjük fel a ROM-ot, annál kevésbé zavaró a lépcsőzetes kimenő jel.

<sup>9</sup> Forrás: Dr. Selényi Endre–Benesóczky Zoltán: Digitális technika példatár BME, Budapest, 1991





33. ábra. Programozható jelformájú függvénygenerátor

## IRODALOMJEGYZÉK

### FELHASZNÁLT IRODALOM

Zsom Gyula :Digitális technika I. Műszaki könyvkiadó, Budapest,1989

Zsom Gyula: Digitális technika II. Műszaki könyvkiadó, Budapest,1989

Ajtonyi István: Digitális rendszerek Miskolci Egyetemi Kiadó ,2006

Benesóczy Zoltán: Digitális tervezés funkcionális elemekkel és mikroprocesszorral Műegyetemi kiadó, 2006

Dr Selényi Endre– Benesóczy Zoltán: Digitális technika példatár BME, Budapest, 1991

Dr. Madarász László : A digitális áramkörök alkalmazástechnikájának alapjai KF GAMF Kar Kecskemét, 2008 (Főiskolai jegyzet)

Szűcs László: Digitális technika példatár Székesfehérvár ,1995

A(z) 0917-06 modul 026-os szakmai tankönyvi tartalomeleme felhasználható az alábbi szakképesítésekhez:

A szakképesítés OKJ azonosító száma:	A szakképesítés megnevezése
54 523 01 0000 00 00	Elektronikai technikus

A szakmai tankönyvi tartalomelem feldolgozásához ajánlott óraszám:  
25 óra

A kiadvány az Új Magyarország Fejlesztési Terv  
TÁMOP 2.2.1 08/1–2008–0002 „A képzés minőségének és tartalmának  
fejlesztése” keretében készült.

A projekt az Európai Unió támogatásával, az Európai Szociális Alap  
társfinanszírozásával valósul meg.

Kiadja a Nemzeti Szakképzési és Felnőttképzési Intézet  
1085 Budapest, Baross u. 52.

Telefon: (1) 210–1065, Fax: (1) 210–1063

Felelős kiadó:  
Nagy László főigazgató