



# **DIGITÁLIS TECHNIKA ALAPJAI**

## **LABORGYAKORLAT SEGÉDLET**

**Debrecen, 2003**

Debreceni Egyetem  
Műszaki Főiskolai Kar  
Villamosmérnöki Tanszék

## DIGITÁLIS TECHNIKA ALAPJAI

LABORGYAKORLAT

SEGÉDLET

Összeállította:

**Szász Csaba**  
főiskolai adjunktus

Debrecen, 2003.

**Felelős kiadó:**

A Debreceni Egyetem Műszaki Főiskolai Kar Főigazgatója

**Kiadó:**

STUDIUM KKT  
4400, Nyíregyháza, Szent-Györgyi Albert út 17  
ISBN

Készült az OP-ART Nyomdában

## 1. mérés: Logikai alapáramkörök

Napjainkban az automatizálás, a mérés technika, a számítástechnika területén és számos egyéb műszaki feladat megoldására alkalmaznak digitális elektronikus berendezéseket. E berendezések alapáramkörei az integrált technológiás félvezető alapú logikai áramkörök.

A digitális technika azon alapszik, hogy az információt - mely elektronikus formában van jelen - kétértékű változók képviselik. Fizikailag egy adott áramköri ponton az adott pont feszültsége vagy a ponton (keresztmetszeten) átfolyó áramerősség képviselheti az információt. Ez a feszültség ill. áramerősség az analóg áramkörökkel ellentétben csak két állapot megkülönböztetésére szolgál, azaz két értéket vehet fel. Ez a két érték általában az adott áramkör családra jellemző feszültség szint szokott lenni, ahol az egyik szint pozitívabb, a másik negatívabb. A gyakorlatban mindkét értékre egy-egy feszültség-tartomány van megadva adott tűréssel. A két értéket jelölhetjük az abszolút értelemben vett „magas” és „alacsony” (angolul „HIGH” és „LOW”) kifejezésekkel, de lehet a logikai algebrában használatos „1”-el és „0”-val is jelölni a két értéket, melyek a logikai „igaz” és „hamis” értékeknek felelnek meg.

**Pozitív logika:** H = logikai „1”, „igaz”      **Negatív logika:** H = logikai „0”, „hamis”  
L = logikai „0”, „hamis”                              L = logikai „1”, „igaz”

A digitális technikában az információ alapegysége ezáltal adott: egy **bit**-nyi (bit = **b**inary **d**igit) információt a gyakorlatban egy áramköri elem (változó) képvisel és két értéket vehet fel, azaz két választási lehetőség közül adja meg az aktuális értéket, variációt. Nagyobb információ mennyiség ábrázolásához több bitet vonunk össze, amit bináris szónak nevezünk. Például nyolc bit együttesen egy **байт**-nyi információt képvisel és  $2^8$ , azaz 256 variáció, lehetőség közül választ ki egyet.

A mérések során egy bit értékének meghatározásakor nem mérjük közvetlenül a feszültséget, hanem az adott bitet képviselő áramköri elem állapotát egy fényemittáló dióda (LED) fogja jelezni, azaz vagy világít, vagy nem.

A kétértékű logikai változókkal (bitekkel) logikai műveleteket lehet végezni, melyeket tudományosan a logikai algebra ír le. Ilyen logikai műveletek egymásutánjából tevődik össze általában a digitális szavakból álló egyedi információk feldolgozásának döntő többsége.

A logikai változók lehetnek állítások (független változók) és következtetések (függő változók), melyek jelölésére a latin ABC nagybetűit használjuk, és értéként az igaz (1) vagy a hamis (0) értékeket vehetik fel.

Az alapvető logikai műveletek az alábbiak (erre a három alpműveletre minden logikai kapcsolási feladat visszavezethető, illetve velük realizálható):

- **ÉS (AND) konjunkció** – logikai szorzás
- **VAGY (OR) diszjunkció** – logikai összeadás
- **NEM (NOT) invertálás, negálás** – tagadás

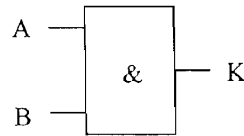
Amennyiben két vagy több állítást logikai ÉS művelet kapcsol össze, akkor a következtetés csak minden állítás egyidejű IGAZ értékénél lesz IGAZ. Ennek algebrai leírása:

$$A \cdot B = K$$

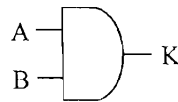
ahol A, B az állítások és K a következtetés. A művelet igazságtáblázatát (értéktáblázat) és logikai szimbólumát az 1. ábra mutatja. [Igazságtáblázatnak nevezzük azt a táblázatot, ahol a bemeneti információk (állítások vagy események) összes lehetséges kombinációjához megadjuk a logikai függvény által definiált kimeneti értéket. A logikai

[függvény általában alapszervelet kombinációjából épül fel, és meghatározza a független és a függő változók közötti kapcsolatot]

A	B	K
0	0	0
0	1	0
1	0	0
1	1	1



a) IEC szabvány



b) régi angolszász

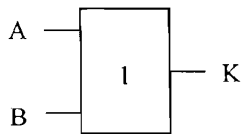
1. ábra : ÉS művelet igazságtáblázata és szimbolikus jelölése

Két vagy több állítás közötti VAGY műveletnél a következtetés akkor igaz, ha az állítások közül legalább az egyik igaz. Algebrai leírása:

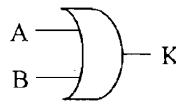
$$A + B = K$$

ahol A, B az állítások és K a következtetés. A művelet igazságtáblázatát és logikai szimbólumát az 2. ábra mutatja.

A	B	K
0	0	0
0	1	1
1	0	1
1	1	1



a) IEC szabvány



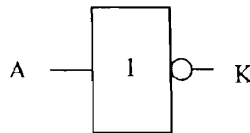
b) régi angolszász

2. ábra: VAGY művelet igazságtáblázata és szimbolikus jelölése

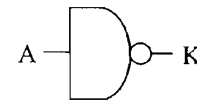
A logikai tagadás – NEM –művelet (negálás, invertálás) az állítás logikai értékét változtatja az ellenkezőjére. Algebrai leírása:

$$\bar{A} = K$$

A	K
0	1
1	0



a) IEC szabvány



b) régi angolszász

3. ábra: A negálás művelet igazságtáblázata és logikai szimbóluma

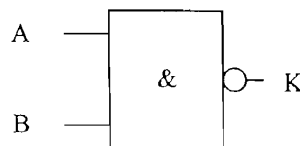
A felsorolt logikai alapfüggvényeket ill. az ezeket realizáló alapkapsolásokat másféleképpen kapuknak, kapuáramköröknek is nevezik.

A fenti három alapszervelet kombinálásával további, összetett logikai műveletek illetve logikai függvények állíthatók össze. A többváltozós logikai függvény esetében a következtetés, azaz a függő változó (egyenlet jobb oldala) egy - a független változókon elvégzett - meghatározott logikai műveletsorozat eredményeként kap értéket. Pl.:  $K = A \cdot B + C$

Két logikai változót ÉS művelettel összekapcsolva és az így kapott eredményt negálva kapjuk az ún. **NAND** (NEM-ÉS) kaput, ami az előbb említett kapukkal ellentétben integrált áramköri megvalósítást tekintve a legelterjedtebb. Szimbolikus áramköri jelölése és igazságtáblázata a korábbiakban leírtakból logikusan eredeztethető, de a 4. ábrán külön feltüntettük. Hasonló módon értelmezett a **NOR** (NEM-VAGY) kapu is.

$$K = \overline{A \cdot B}$$

A	B	K
0	0	1
0	1	1
1	0	1
1	1	0

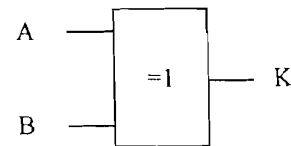


4. ábra: NAND függvény algebrai leírása, igazságtáblázata és szimbolikus áramköri jelölése

A NAND, NOR, AND, OR és a negáció logikai alapfüggvények mellett a digitális technikában két további alapfüggvény is gyakorta előfordul, melyeknek ezért külön elnevezést is adtak.

- **Kizáró VAGY** (antivalencia) XOR

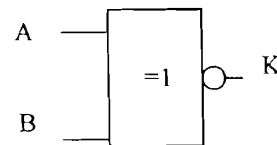
$$K = \bar{A} \cdot B + A \cdot \bar{B} = A \oplus B$$



- **Kizáró NOR** (ekvivalencia) XNOR

$$K = \bar{A} \cdot \bar{B} + A \cdot B = \overline{\bar{A} \cdot B + A \cdot \bar{B}}$$

$$K = \overline{A \oplus B} = A \otimes B$$



Mindkettő fontos és gyakran használt függvény. Kettő vagy több bemenet összehasonlítására szolgálnak és azt jelzik ki (1-el vagy 0-val), ha mindegyik bemenet azonos potenciálon van. A kizáró VAGY kimenete logikai 0 értéket vesz fel, ha a bemenetek azonos potenciálon vannak, ugyanezt az ekvivalencia-kimenet logikai 1-el jelzi. A két kapuáramkör csak egy kimeneti negálásban különbözik egymástól.

#### ***Általános tudnivalók a mérések során:***

A mérések során CMOS technológiájú integrált áramköröket használunk a TTL áramkörökkel szemben felmutatott néhány előnyös tulajdonságuk miatt:

- a tápfeszültség szélesebb tartományban helyezkedhet el (viszont 74HC sorozatnál vigyázni kell arra, hogy a **tápfeszültség nem lehet nagyobb 7 V-nál**, tehát a V5 (7,5 V) és V6-os (9 V) jelölésű tápfeszültségeket tilos használni!)
- kisebb a bemenő (vezérlő) teljesítményük, ezért több bemenet csatlakoztatható egy kimenetre
- jóval kevesebb áramot vesznek fel a tápellátó hálózathoz

A CMOS IC-k hiányossága viszont az, hogy rendkívül érzékenyek az elektrosztatikus hatásokra. Ezért minden használaton kívüli IC-t a tárolásukra szolgáló, sztatikus hatásoknak ellenálló **szivacsban kell tárolni**. Az IC-t a már előzőleg összeállított kapcsolásba legutoljára helyezzük be, (a lábait **kézzel lehetőleg ne érintsük!**) és csak ezután adjuk rá az áramkörre a tápfeszültséget. Az áramkör szétszedését, változtatásait csak feszültségmentes állapotban végezzük!

Az IC-k és tranzisztorok láb kiosztása, az ellenállásértékek és a kerámiakondenzátorok kódjai minden mérőhelyen külön fel vannak tüntetve.

Elektrolit kondenzátorok csatlakoztatásánál és az IC-k tápfeszültség-bekötésénél **figyeljünk a helyes polaritásra!**

#### **Fontos!**

Minden mérést a szükséges áramköri elemek kiválasztásával kezdjük! Az összeszerelési rajz alapján, annak megfelelően állítsuk össze az elemekből a kapcsolást. A polaritásra külön figyeljünk, helytelen bekötés esetén az áramköri elemek tönkremehetnek! Az áramkör bekapcsolására (feszültség alá helyezésére) csak megegyezési ellenőrzés után és a mérésvezető engedélyével kerülhet sor.

**Feladat, 1. mérés:**

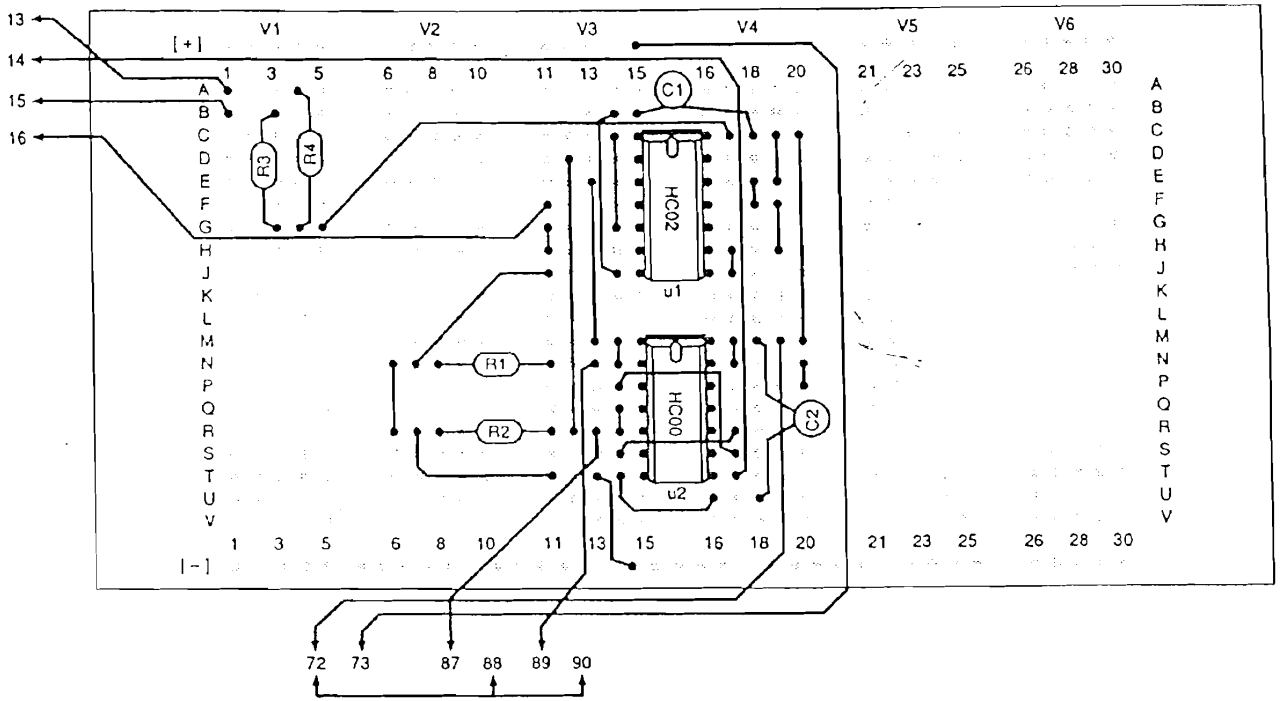
Alapvető logikai kapuáramkörök integrált technológiás megvalósításának, működésének megismerése.

- Állítsa össze az alábbi rajz szerint a megadott kapcsolást!

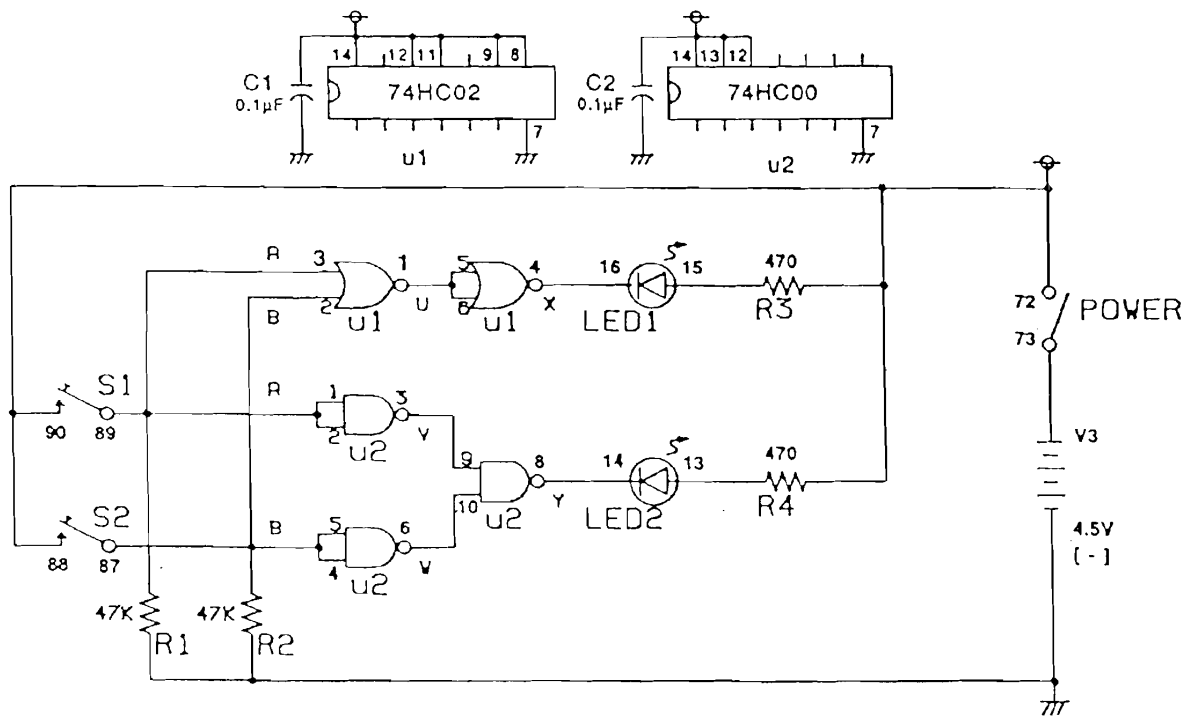
Az összeállított kapcsolás tulajdonképpen két logikai függvényt megvalósító áramkör, melyek mindegyike két bemenettel rendelkezik. S1 és S2 kapcsolók segítségével a digitális bemeneti jeleket lehet a kapuáramkörök megfelelő bemeneteire ráadni. A két áramkör kimeneti állapotait a LED-ek mutatják.

*A kapcsoló bekapcsolásakor (benyomásakor) H szint, azaz „1” kerül a bemenetre. A kapcsoló kikapcsolt (elengedett) állapotában a bemenet értéke „0”. Amikor az adott kimenet „0”, a LED világít és amikor „1”, nem világít.*

- A tápfeszültség ráadása után hasonlítsa össze a két áramkör működését (a két áramkör logikai működésének egzakt leírása, igazságtáblázat felvétele, a közbülső, multiméterrel megmért logikai állapotokkal együtt) és indokolja a tapasztaltakat!
- Ezután cserélje fel U1 5-6-4-es NOR kapuját U2 8-9-10-es NAND kapujával a megfelelő vezetékek átkötésével! Írja le az így előállt logikai függvényeket táblázattal és algebrai módon is!



U1	74HC02	R1	47KΩ	R3	470Ω	C1	0.1μF
U2	74HC00	R2	47KΩ	R4	470Ω	C2	0.1μF



Forrás: [1]



## 2. mérés: Kombinációs hálózat

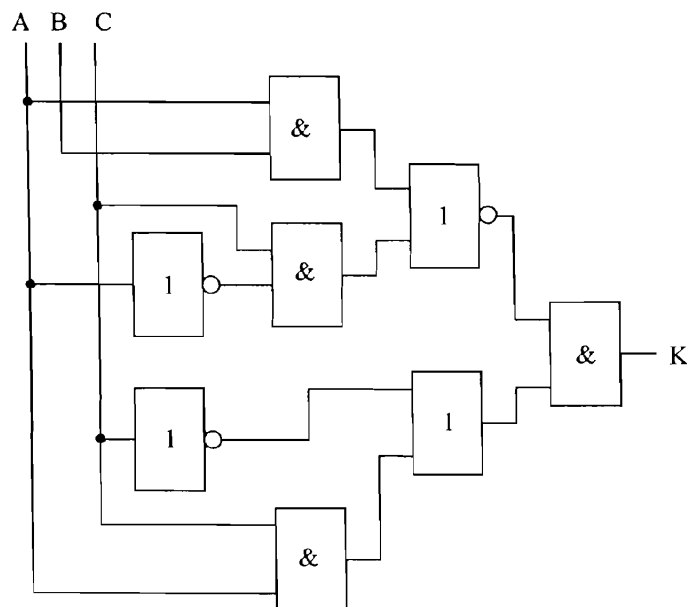
A **kombinációs hálózatok** olyan alpműveletekből álló összetett digitális hálózatok, amelyeknél a kimeneti változó mindenkor az adott időpillanatban a bemenetekben meglévő változók aktuális értékeitől függ. Másképpen fogalmazva a kombinációs hálózatok logikai függvényeket valósítanak meg. A digitális áramkörök másik nagy csoportját a **szekvenciális hálózatok** képezik, melyeknél a kimeneti változók aktuális értékei nem csak az aktuális bemeneti jel-állapot(ok)tól függenek, hanem már az időtényező is közrejátszik: például tároló-elemek révén korábbi állapotok is „megőrződnek” és befolyásolhatják az aktuális kimeneti értéket. A szekvenciális hálózatoknál ellentétben a kombinációs hálózatokkal visszavezetések is előfordulnak (kimenetet visszavezetjük az előtte lévő bemenetre), ezáltal itt már két jelhaladási irány létezik.

A kombinációs hálózatok, azaz logikai függvények megadása alpműveletekkel történhet algebrai úton, igazságtáblázattal, grafikusan vagy szimbólumokkal. Az algebrai alakban az egyenlőségjel bal oldalára írjuk a logikai műveletekkel összekapcsolt állításokat és a jobb oldalára a következtetést.

Például vizsgáljuk meg az alábbi, három bemenettel rendelkező logikai függvényt:

**algebrai leírás:**  $(A \cdot B + \bar{A} \cdot C) \cdot (\bar{C} + A \cdot C) = K$

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



1. ábra. Logikai függvény megadása igazság táblázattal és logikai vázlattal

A vezérléstechnikában vagy egyéb logikai feladatok megoldásánál a gyakorlatban gyakran fordul elő, hogy összetett, bonyolult logikai függvényt kell megvalósítani.

Bármilyen logikai hálózat megvalósításánál alapvető célkitűzés, hogy megkeressük a minimális elemű elrendezést. Egy adott logikai függvény minimális alakját egyszerűsítéssel érhetjük el, amire többféle eljárás létezik. Az algebrai eljárásnál a logikai alpműveletek algebrai tulajdonságait (kommutativitás, asszociativitás és disztributivitás) logikai tulajdonságait és a logikai tagadásra vonatkozó - de Morgan féle - tételt használjuk fel.

Kitüntetett elemekkel	$1 \cdot 1 = 1$	$0 \cdot 0 = 0$
(1 és 0) végzett	$1 \cdot A = A$	$0 \cdot A = 0$
műveletek:	$1 + 1 = 1$	$0 + 0 = 0$
	$1 + A = 1$	$0 + A = A$

Azonos állításokkal végzett műveletek:

$$A \cdot A = A \quad A \cdot \tilde{A} = 0$$

$$A + A = A \quad A + \tilde{A} = 1$$

A logikai tagadásra vonatkozó – de Morgan – tételek:

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

A fenti tételek felhasználásával egyszerűsítsük algebrai úton a példában szereplő logikai függvényt!

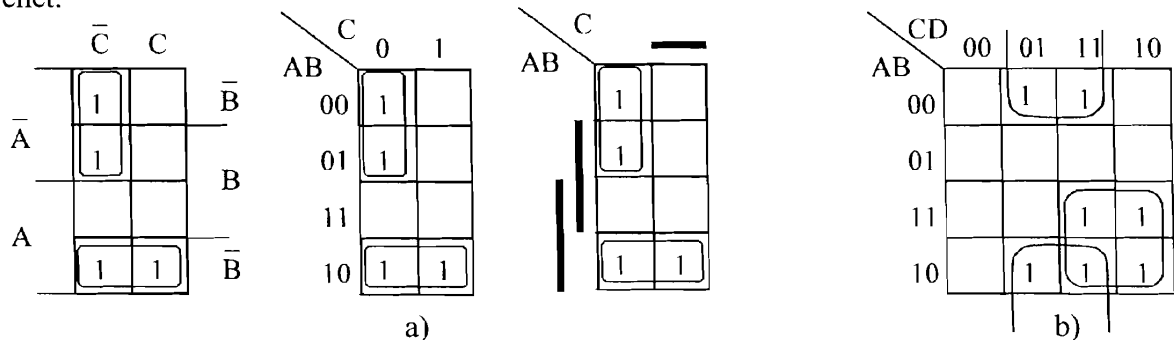
$$\begin{aligned} (A \cdot B + \overline{A} \cdot C) \cdot (\overline{C} + A \cdot C) &= (\overline{A} \cdot \overline{B} \cdot \overline{A} \cdot C) \cdot (\overline{C} + AC) = (\overline{A} + \overline{B})(A + \overline{C})(\overline{C} + AC) = \\ &= (\overline{A}\overline{A} + \overline{A}\overline{C} + \overline{A}B + \overline{B}C)(\overline{C} + AC) = \overline{A}\overline{C}\overline{C} + \overline{A}\overline{C}AC + \overline{A}B\overline{C} + \overline{A}BAC + \overline{B}C\overline{C} + \overline{B}CAC = \\ &= \overline{A}\overline{C} + \overline{A}B\overline{C} + \overline{A}B\overline{C} + \overline{B}C = \overline{A}\overline{C} + \overline{A}B(\overline{C} + C) + \overline{B}C = \overline{A}\overline{C} + \overline{A}B + \overline{B}C = \\ &= \overline{A}\overline{C} + \overline{A}B + \overline{B}C(\overline{A} + A) = \overline{A}\overline{C} + \overline{A}B + \overline{A}B\overline{C} + \overline{A}B\overline{C} = \overline{A}\overline{C}(1 + \overline{B}) + \overline{A}B(1 + \overline{C}) = \overline{A}\overline{C} + \overline{A}B \end{aligned}$$

### Grafikus ábrázolás – Karnaugh – tábla (Veitch-diagramm)

A logikai függvények algebrai úton való minimalizálása nem mindig egyszerű, és rendszerint bizonyos intuíció vagy szerencse kell hozzá.

A logikai függvények egyszerűsítésének másik módja, amikor a függvényt az igazságtáblázathoz hasonlóan egy táblázatban (grafikus módon) ábrázoljuk, és abból olvassuk ki a legegyszerűbb alakot. A bemeneti változók összes előfordulási lehetőségét itt is fel kell sorolni úgy, hogy a táblázat egyes rácsmezői fogják az egyes variációkhoz tartozó kimeneti értéket tartalmazni. Ha a bemeneti változók száma  $n$ , akkor az összes bemeneti variáció és ezzel a rácsmezők darabszáma  $2^n$ . Amelyik bemeneti variációnál a függvényérték egy, a hozzá tartozó rácsmezőbe (cellába) 1-es jel kerül. Arra kell figyelni, hogy a táblázat minden egyes szomszédos sora ill. oszlopa egymástól csak egy bemeneti változó értékében különbözhet. A szomszédos cellákat jelölő bináris kombinációk mindig csak egyetlen változóban (jelben) különböznek egymástól.

A példánkban szereplő függvény Karnaugh-táblája például a 2. ábra a) pontjában megadott lehet.



2. ábra: Példák Karnaugh táblákra a) 3 bemenet, b) 4 bemenet

A Karnaugh féle ábrázolásmóddal lehetőség adódik egy igen praktikus és áttekinthető minimalizálás megvalósítására. A logikai függvény minimális alakját úgy kapjuk meg, hogy a táblázat 1-el jelölt celláit, azaz a cellák bináris kódjait a táblából kiolvassuk és VAGY kapcsolattal összekapcsolva írjuk be az illető logikai függvény kifejezésébe. A minimalizálás érdekében a szomszédos, logikailag összevonható cellákat össze kell vonni. Az összevonható cellákat a 2. ábra két példáján hurkokkal jelöltük. Ha egy cella több hurkokban is szerepel, nem baj, az a lényeg, hogy lehetőleg minél több szomszédos cellát fogjunk össze.

A 2. ábrából kiolvasott függvények:

a)  $K = \overline{A}\overline{C} + \overline{A}B$

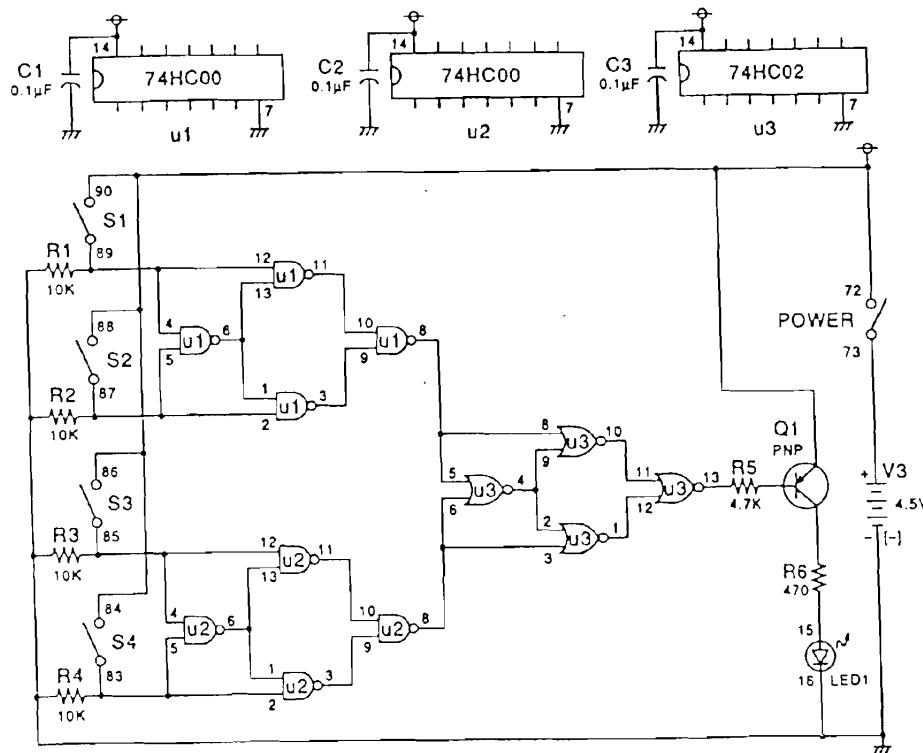
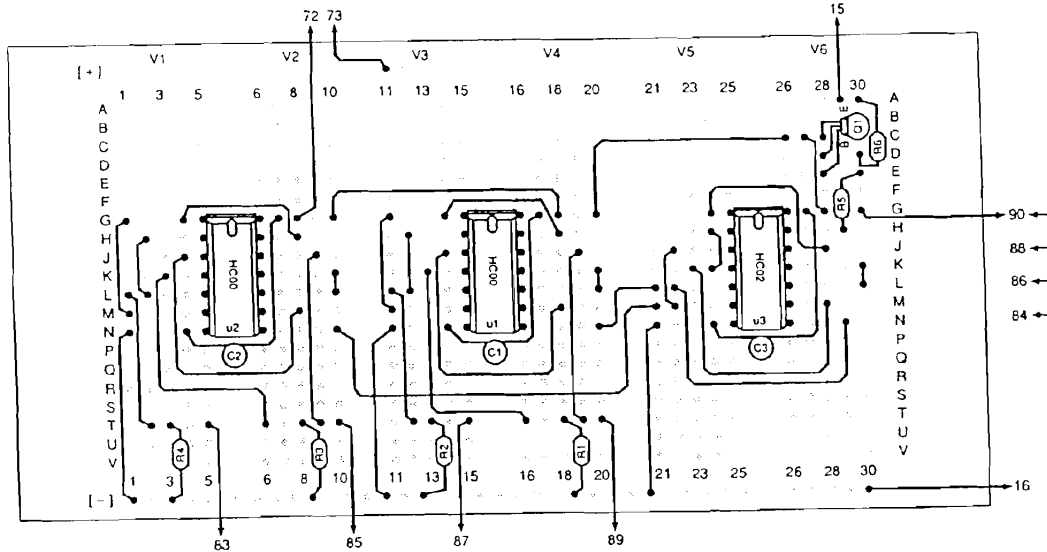
b)  $K = \overline{B}D + AC$

**Feladat, 2. mérés:** Állítsa össze az alábbi kapcsolást: (356)

Az összeállított kombinációs hálózat négy bemeneti változó (S1,...,S4) aktuális értékének függvényében kapcsolja be illetve ki a kimeneten lévő LED-et.

- Vizsgálja meg a megvalósított logikai függvényt a bemenetek változtatásával és vegye fel az igazságtáblázatot!
- Írja fel a megvalósított függvényt a kapcsolási rajz alapján algebrai alakban és a mérési eredmények alapján Karnaugh-táblával is! Hogyan írható le a kapcsolat működése pár mondatban?
- Milyen három logikai részfüggvényre, "alap" funkcióra bontható fel a megvalósított függvény (U1, U2, U3)? Bizonyítsa be, hogy a gyakorlatban tapasztalt működés az elméletben elvártak megfelelő!

U1	74HC00	Q1	PNP	R3	10 kOhm	R6	470 Ohm	C3	0,1 μF
U2	74HC00	R1	10 kOhm	R4	10 kOhm	C1	0,1 μF		
U3	74HC02	R2	10 kOhm	R5	4,7 kOhm	C2	0,1 μF		



Forrás: [1]

### 3. mérés: Kódoló és dekódoló áramkörök

A digitális technika az információ megjelenítésére fizikailag két állapotot különböztet meg, ezért a digitális áramköri funkciók elméleti tárgyalásakor célszerű a szintén két számjegyet használó kettős (bináris) számrendszert alkalmazni. Általában bináris ábrázolásnak nevezünk minden olyan rendszert, amelyik két jellel operál.

#### Számrendszerek

A számrendszerek feladata az impulzusszámlálás, a számolt értékek ábrázolása, számokkal történő műveletvégzés. A kettős számrendszer elnevezésében a kettő azt jelenti, hogy egy helyi-értéken kétféle számjegy (0, 1) fordulhat elő, a kettő a számrendszer alapja. A tízes számrendszerben 10-féle számjegy létezik (0,1,2,3,4,5,6,7,8,9), itt a számrendszer alapja tíz.

Egy szám egy vagy több helyi értékből áll, azaz egymás mellé helyezett számjegyekből. A helyiértékek súlyozva vannak. A jobb oldalt legszámban álló számjegy (helyi érték) súlya a legkisebb, balra haladva a soron következő mindig nagyobb. Ha az egyes egymás mellett álló számjegyeket jobbról, 0-val kezdve sorszámozzuk, akkor a számjegy-helyek (helyi értékek) súlya megegyezik azzal az értékkel, amit akkor kapunk, ha a számrendszer alapját a hely sorszámára emeljük. A helyiértékhez súlyként tehát mindig a számrendszer alapjának valamelyik hatványa van hozzárendelve. A szám értékét az egyes helyeken álló számjegyek és a hozzájuk tartozó helyiérték-súlyok szorzatának összege adja.

Pl. a 13 tízes számrendszerbeli szám :  $1 \cdot 10^1 + 3 \cdot 10^0 = 13$

ugyanaz kettős számrendszerben:

$$1101: \quad 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8+4+1=13$$

Ha a számunk n darab helyi-értékből áll, akkor a számrendszer alapját az n-dikre emelve megkapjuk az összes előfordulható variációk számát, azaz például két helyiértéken

- a tízes számrendszerben  $10^2 = 100$ , azaz száz féle különböző állapot van (0 – 99)
- a kettős számrendszerben  $2^2 = 4$  (0 – 3), tehát itt csak négyig tudunk számolni.
- a 16-os számrendszerben 2 helyiértéken  $16^2 = 256$  db állapot fordulhat elő.

A 16-os számrendszer gyakran használatos a számítástechnikában, itt 16 féle „számjegy” különböztethető meg: 0,1,...,9, A,B,C,D,E,F ( $A_H=10_D$ ,  $B_H=11_D$ ,  $C_H=12_D$ ,...,  $F_H=15_D$ ) A 2-es számrendszert (és számokat) bináris (B), a 10-es számrendszert decimális (D) (vagy dekadikus), a 16-ost pedig hexadecimális (H) számrendszernek is nevezik. A számrendszert úgy jelölik, hogy a szám után alsó indexként odaírják a számrendszer alapját.

Pl.  $1A_{16} = 26_{10} = 11010_2$

vagy  $1A_H = 26_D = 11010_B$

A **digitális számlálók** a számokat szintén több helyi-értéken jelenítik meg, melyek mindegyike egy-egy bit-nek (binary digit) felel meg. Több bitet egymás mellett egy bináris szónak, vagy kód-szónak nevezünk

Minden digitális számláló egy bizonyos **kód** szerint dolgozik. Ez a kód abból a bináris szó-sorozatból adódik, amelyeken valamilyen esemény (pl. a számláló bemenetére érkező impulzusok) számolása során a számláló keresztülmegy. A számláló egyes állapotai és az addig beérkezett impulzusok száma közötti egyértelmű hozzárendelést kódnak, kódolásnak nevezzük.

A számokon végzett műveleteknél, számlálásnál a kettős számrendszeren kívül különböző kódokat használhatunk, melyek előnyös-hátrányos tulajdonságai, kiválasztásuk szempontjai különbözőek lehetnek:

- számokon végzett műveletek könnyű kivitelezhetősége
- hibák felismerése
- a rendelkezésre álló átviteli csatorna optimális kihasználása
- biztonság a túl könnyű, jogosulatlan dekódolás ellen
- esetleg előre meghatározott az egyes helyiértékek súlya

Amennyiben csak a számláló legegyszerűbb felépítése (kevés elem, minimális huzalozás) vagy a billenési folyamatok gyorsasága, azaz a maximális megengedett frekvencia az egyetlen kritérium, akkor a kódolás, és annak tulajdonságai háttérbe szorulnak. A gyakorlatban a többféle követelmény kielégítésére kompromisszumos megoldást kell találni.

A kódokat az alábbiak szerint csoportosíthatjuk:

**Tetrádikus kódok**-nak nevezzük összefoglaló néven azokat a kódokat, amelyek 4 helyiértékű kód-szavakkal, un. *tetrádokkal* dolgoznak. Többnyire decimális számjegyek (és -számok) kódolására használatosak.

**BCD-kódoknak** vagy **dekádikus kódoknak** nevezzük az összes olyan kódot, amelyeket csak a 0...9 decimális számjegyek kódolására használunk. (BCD: binárisan kódolt decimális számjegy) A leggyakrabban használt BCD-kód a 8-4-2-1 kód, mely egyben tetrádikus kód is.

(BCD-kód alatt sokan csak a 8-4-2-1 kódot értik.)

#### Bináris kód (1. ábra)

Az előzőekben ismertetett kettes számrendszert szokták **bináris kód**-nak is nevezni. A kódszó egyes helyiértékeinek súlya, jobbról kezdve, 2 egész kitevőjű hatványaival egyezik meg.

A kettes számrendszer szerinti kód a legegyszerűbb kód. Ennek oka, hogy a digitális technikában használatos más kódokhoz képest adott számú állapot megkülönböztetéséhez, (esemény megszámlálásához) egy bizonyos számolási kapacitáson belül a legkevesebb helyiértékre a kettes számrendszernek van szüksége, azaz a bináris kódú számlálóknál van a legkevesebb FlipFlop-ra szükség.

decimális szám (órajel impulzus)	bináris kód				decimális szám (órajel impulzusok)	bináris kód			
	D	C	B	A		D	C	B	A
0	0	0	0	0	8	1	0	0	0
1	0	0	0	1	9	1	0	0	1
2	0	0	1	0	10	1	0	1	0
3	0	0	1	1	11	1	0	1	1
4	0	1	0	0	12	1	1	0	0
5	0	1	0	1	13	1	1	0	1
6	0	1	1	0	14	1	1	1	0
7	0	1	1	1	15	1	1	1	1
helyiérték súlya	8	4	2	1	helyiérték súlya	8	4	2	1

1. ábra: Bináris kódok 0...15-ig

#### 8-4-2-1 kód (2. ábra)

A 8-4-2-1 kód a dekádikus (BCD) kódokhoz tartozik. A 0...9 számjegyek kódolására szolgál, négy helyiértéken. Egy tetrád egy számjegynek felel meg. A megfeleltetés a 9-es számjegyig megegyezik a kettes számrendszerbeli, bináris kóddal.

A 16 lehetséges kódszóból 6 nincs felhasználva, ezeket pszeudotetrádoknak nevezik.

A 8-4-2-1 elnevezés egy tetrádon belül az egyes helyiértékek súlyát jelenti. Használatos még a BCD, vagy NBCD (természetes binárisan kódolt decimális szám) vagy az 1-2-4-8 elnevezés is. Gyakran használják a rövidebb BCD-kód megnevezést, habár a BCD-kódokhoz még más kódok is tartoznak.

A 8-4-2-1 kód nagy hátránya, hogy számolásra nehezen használható. Ennek oka az, hogy az összeadás, ha az összeg nagyobb mint 9, pszeudotetrádokat eredményez és nem keletkezik átvitel - ellentétben a bináris kóddal. Ezért ez a kód számolásra csak bonyolult korrekciókkal alkalmas.

Deci- mális szám	BCD-kód (binárisan kódolt decimális szám)																		
	8-4-2-1 kód				Gray-kód				Aiken-kód (2-4-2-1)				Excess-3-kód (X-3)						
	D	C	B	A	D	C	B	A	D	C	B	A	D	C	B	A			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	
1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	1	0	0
2	0	0	1	0	0	0	1	1	0	0	1	0	0	1	0	1	0	1	
3	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	0	0	1	0	0	0	1	0	1	1	1	1
5	0	1	0	1	0	1	1	1	1	0	1	1	1	1	0	0	0	0	0
6	0	1	1	0	0	1	0	1	1	1	0	0	0	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0	1	1	0	1	1	0	1	1	0	1	0
8	1	0	0	0	1	1	0	0	1	1	1	1	0	1	0	1	1	1	1
9	1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0
helyiérték súlya	8	4	2	1	nincs				2	4	2	1	nincs						

2. ábra: Dekádikus (BCD) kódok

Bár a bináris kód előnyösebb lenne a számolásra, viszont nagyobb számok ábrázolásához binárisan kódolva nagyon sok helyiértékre lenne szükség, ami a számolást és a hibakeresést is nehezkesse teszi. Ezenkívül a leglényegesebb, hogy a sok helyiértékből álló, hosszú bináris kódszavak dekódolása nagymennyiségű áramköri elemet, bonyolult kapcsolást igényel. Ezért nagy számokon végzett műveleteknél előnyösebb a decimális számaábrázolást használni, ahol az egyes helyiértékeken lévő számjegyek binárisan vannak kódolva (BCD számaábrázolás). A 0,...,9 számjegyeket kódoló tetrádok súlyát a helyiértéktől függően 10-nek a hatványai fogják adni. Ezzel a módszerrel a bináris kódot hozzáigazítottuk a decimális rendszerhez.

### Gray-kód

A Gray-kódnál az egyes egymást követő kódszavak csak egy bitben különböznek egymástól. A számlálók esetében gyakorlatilag ez azt jelenti, hogy minden bejövő impulzusnál egyszerre csak egy tároló billen át. Az ilyen kódokat *egylépéses* kódoknak nevezzük.

A Gray-kód jellemzője, hogy a helyiértékek – bár helyük meghatározott – nincsenek a számrendszerekben szokásos módon súlyozva. A négy-helyiértékű kódszavak decimális számjegyekhez való hozzárendelésénél csak az egylépésesség volt az irányelv. A kódszavak nem csak tetrádokból állhatnak, ki lehet bővíteni tetszőleges hosszúságú kódszavakká is. Négy helyiértéken például 16 féle kódszó 16 féle számjegyet is jelölhet, viszont BCD kódról csak az első 10 (0...9) esetében beszélhetünk.

### Aiken-kód

2-4-2-1 kódnak is nevezik, amely a helyiértékek súlyának sorrendjére utal. Csak tetradikus kódként és csak a 0...9 számjegyek kódolására szolgál és *szimmetrikus*. A szimmetrikusság azt jelenti, hogy pl. a 9-es számjegy kódszója a 0 számjegy kódszójából úgy képezhető, hogy az egyes biteket negáljuk. Ugyanez érvényes az 1 és 8, a 2 és 7 stb. számjegykódokra is. Az összetartozó számjegy-párok kódszavai egymásnak az inverzei. Másképpen kifejezve: egy adott kódszó kilences-komplementjét az egyes bitek egyszerű invertálásával megkaphatjuk. Egy szám kilences-komplementje az a szám, amelyik azt kilencre kiegészíti. Használata a dekadikus kódokkal dolgozó összeadó és kivonó áramköröknél biztosít jelentős számítási és kapcsolástechnikai könnyebbséget.

Az Aiken-kód hátránya, hogy a 0-t és a 9-est a 0000 és az 1111 kódszavak jelölik és ezek az állapotok zavar (pl. áramkimaradás) esetén könnyen előadódhatnak (anélkül, hogy a

hibát fel lehetne ismerni). Ezt a hátrányt küszöböli ki az **Excess-3-kód**, ami szintén szimmetrikus és az egyes bitek negálásával szintén a kilences-komplementet adja. A helyiértékek itt sem súlyozottak. A kódot legkönnyebben úgy kapjuk meg, hogy a kérdéses decimális szám bináris kódjához hozzáadunk binárisan 3-at (0011).

### Hexadecimális kód

Tulajdonképpen nem más, mint a 16-os számrendszeren alapuló tetradikus kód. A számrendszer „számjegyeit” - azaz a 0...9 számjegyeket és A...F betűjeleket - 4 helyiértékű bináris kódszavak kódolják. A 8-4-2-1 kód által ki nem használt pszeudotetrádokat (10...15 decimális számok kódjait) a hexadecimális kód kihasználja. A tetrádokon belül a helyiértékek a bináris számrendszer szerint súlyozottak (8-4-2-1).

Lehetőség van a nagyobb számokat a BCD kódokhoz hasonlóan úgy képezni, hogy a hexadecimális kód tetrádjaival helyettesítjük a 16-os számrendszer egyes helyiértékein álló „számjegyeket”, és a tetrádokat a sorban elfoglalt helyüktől függően a 16-os számrendszer szerint súlyozzuk.

A hexadecimális kóddal tehát egy binárisan kódolt 16-os számrendszerbeli számot lehet ábrázolni.

### 5 vagy annál több helyiértékű kódok

Redundánsnak nevezzük azokat a kódokat, amelyek több helyiértékkel rendelkeznek mint amennyire a bináris kód szerint szükség lenne. (pl. 10 számjegyet akarunk kódolni 5, vagy annál több helyiértéken). A redundáns kódok elsődleges célja a bithibák felismerése és esetleges kijavítása.

Az **5-ből 2 kód** egy dekádikus (azaz a 0...9 számjegyeket kódoló), 5 helyiértékes, egy lépéses kód, melynek helyiértékei nem súlyozottak. Jellegzetessége, hogy mindegyik kódszóban az öt helyiértékből kettő az 1-es (H) értéket veszi fel (innen az 5-ből 2 elnevezés). Ezzel a szabályszerűséggel lehetővé válik a kódolási hibák felismerése.

A **„10-ből 1” kód** esetében minden kódszó csak egy db „1”-es (H) bináris értéket tartalmaz, a többi bit értéke „0”. Az „1”-es értéket mindig az a helyiérték veszi fel, amelyiknek a sorban elfoglalt helye jobbról számolva megegyezik a kódolt decimális számmal. A kód kétlépéses (két egymásra következő kódszó 2 bitben tér el egymástól), tetszőlegesen kiterjeszhető „m-ből 1” kódra és nincs szüksége dekódolásra. Tulajdonképpen nem is nevezhető kódnak, mert az m db összes lehetséges variációt m db helyiértéken ábrázoljuk. Ha n ( $n \leq m$ ) a kódolandó szám, akkor az n-dik helyiérték értéke lesz 1-es.

### Dekódoló áramkörök

feladata a digitális adatfeldolgozás során használt kódok visszaalakítása többnyire a tízes (decimális) számrendszerbe, ezért ezek kimenete már nem bináris ábrázolás. Elsősorban a digitális jelfeldolgozó egységek kimeneti fokozataként (a perifériák felé) használatosak.

A dekódolóra szemléletes példa a hétszeggmenses kijelző dekódolója, melynek egy bináris kódot (kettes számrendszerben ábrázolt számot) kell úgy átalakítani, hogy a kimeneti jelével az egyes szeggmensek közvetlenül vezérelhetők legyenek.

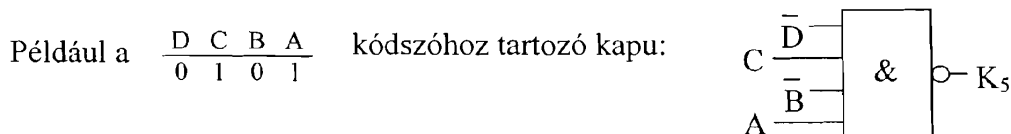
A dekódoló áramköröknek mindig párhuzamos formában rendelkezésre álló kódszavakra van szükségük. A soros továbbítású információkat ezért előbb párhuzamos formára kell hozni.

Ha a dekódolásra váró kódszó n db helyiértékkel (bittel) rendelkezik, akkor maximum  $m=2^n$  számú egymástól különböző kódszó (bitsorozat) fordulhat elő. Ebből a kódszóból például „m-ből 1” típusú dekódolás lehetséges.

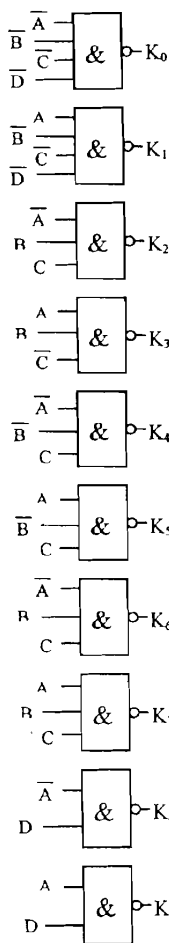
Egyfokozatú a dekódoló, ha a beérkező jel a kimenetig csak egy kapuáramkörön kell, hogy áthaladjon. Ahány kapuáramkörön kell a beérkező jelnek a kimenetig áthatolnia, annyi fokozatú a dekódoló áramkörünk.

”m-ből 1” típusú **egyfokozatú** dekódolóknál a beérkező n db jelvezeték (negálva vagy nem negálva) mindegyike egy-egy NAND vagy NOR kapura megy rá. A kapuk tehát legalább n bemenettel kell, hogy rendelkezzenek. Ahány féle dekódolandó kódszó van (max. 2<sup>n</sup> lehet), annyi kapura van szükség. Mivel a gyakorlatban maximum 8 bemenettel rendelkező kapuáramkörök léteznek, ezért egyfokozatú dekódolók csak maximum 8 bites kódszók dekódolására alkalmasak. Ha a kódszó több mint 8 helyiértéket (bitet) tartalmaz, többfokozatú dekódolót kell használni.

Ha pl. a négy helyiértékes kódszó összes lehetséges 16 változatából egy adott kód (pl. BCD) csak 10-et használ ki, akkor az egyfokozatú dekódolóhoz (leszámítva az invertálásához esetlegesen szükséges kapukat) csak 10 db 4 bemenetes NAND (vagy NOR) kapura van szükség. A NAND kapukra a bemeneteket úgy kell bekötni, hogy az adott kódszó-hoz tartozó kapu kimenete csak akkor legyen L, ha az adott kódszó jelenik meg a bemeneti négy helyiértéken, egyébként a kapu H szinten legyen.



Fenti példa esetében egy olyan „10-ből 1” (decimális) kimeneti „kódot” kapunk, melynél a 10 db kimeneti jelből (K<sub>0</sub> – K<sub>9</sub>) csak a bemeneti bináris kód decimális értékének megfelelő sorszámú veszi fel az "L" értéket, az összes többi kimenet "H" szinten van.



Decimális szám	Kódszó 8-4-2-1 kódban				Kimenetek (10-ből 1 kód)									
	D	C	B	A	K <sub>0</sub>	K <sub>1</sub>	K <sub>2</sub>	K <sub>3</sub>	K <sub>4</sub>	K <sub>5</sub>	K <sub>6</sub>	K <sub>7</sub>	K <sub>8</sub>	K <sub>9</sub>
0	L	L	L	L	L	H	H	H	H	H	H	H	H	H
1	L	L	L	H	H	L	H	H	H	H	H	H	H	H
2	L	L	H	L	H	H	L	H	H	H	H	H	H	H
3	L	L	H	H	H	H	H	L	H	H	H	H	H	H
4	L	H	L	L	H	H	H	H	L	H	H	H	H	H
5	L	H	L	H	H	H	H	H	H	L	H	H	H	H
6	L	H	H	L	H	H	H	H	H	H	L	H	H	H
7	L	H	H	H	H	H	H	H	H	H	H	L	H	H
8	H	L	L	L	H	H	H	H	H	H	H	H	L	H
9	H	L	L	H	H	H	H	H	H	H	H	H	H	L

3. ábra: 8-4-2-1 kód dekódolója, kapcsolása és igazságtáblázata (csak a megengedett kódszavak vannak beírva)

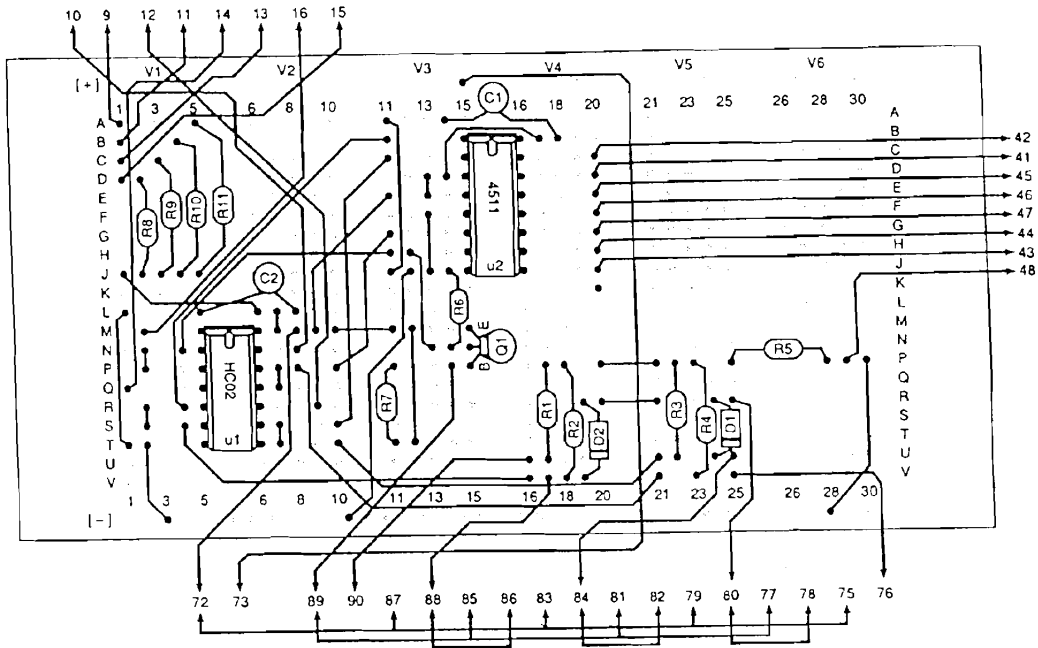
Az ábrából látható, hogy mivel nincs minden lehetséges kódszó kihasználva, nem szükséges minden kapura mind a négy bemenetet bekötni.

[A karnaugh-táblázatban ez azt jelentené, hogy minél több rácscella összefogása érdekében azokat a rácscellákat is kiolvassuk, melyek elvileg nem fordulhatnak elő, ezáltal ezeken a cellákon (kódszavaknál) is L értéket kapunk, de ez jelen esetben nem zavar.]

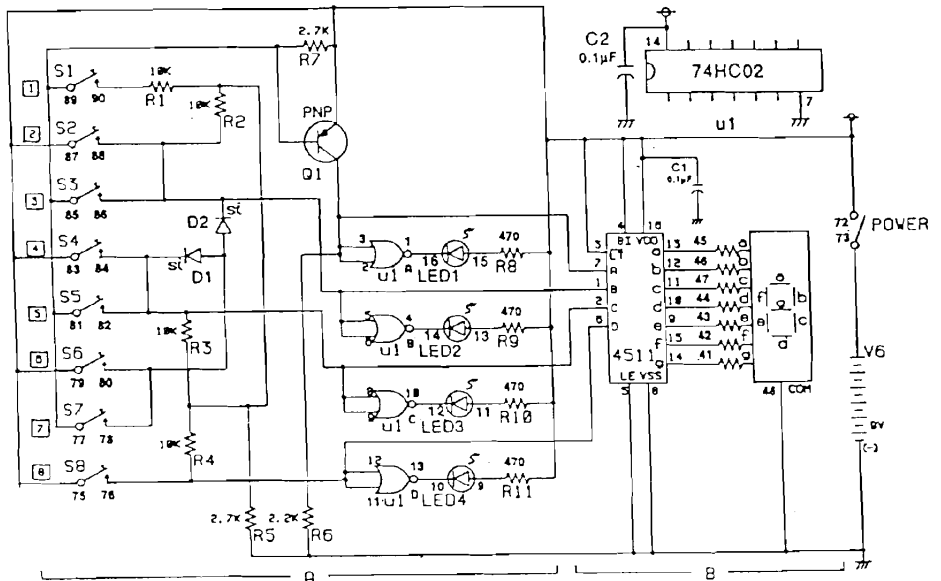
**Feladat, 3. mérés:** Állítsa össze az alábbi kapcsolást:

(341)





U1	74HC02	R1	10KΩ	R5	2.7KΩ	R9	470Ω	C1	0.1μF
U2	74HC4511	R2	10KΩ	R6	2.2KΩ	R10	470Ω	C2	0.1μF
Q1	PNP	R3	10KΩ	R7	2.7KΩ	R11	470Ω		
		R4	10KΩ	R8	470Ω				



Forrás: [1]

Az áramkör a hétszégmenses LED kijelzőn megjeleníti a lenyomott (S1-S8) gombok sorszámát. A vázlatból látható, hogy ez az áramkör egyrészt egy kódolóból áll, mely átalakítja az 1-8 decimális számokat bináris számokká. A kódolt adatok az 1- 4-es LED-eken jelennek meg. Az áramkör második része egy dekódoló IC, mely a hétszégmenses kijelző számára dekódolja a bináris adatokat.

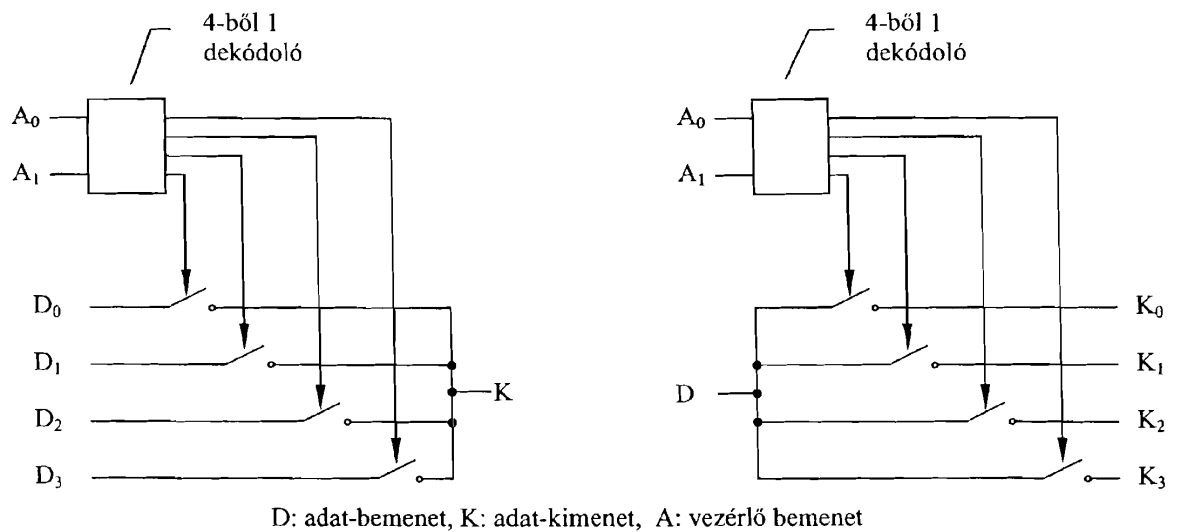
- Ellenőrizze az áramkör működését, táblázatban ábrázolja a bemeneti és kimeneti jelek közötti összefüggést!
- A kódoló helyes működésének feltétele, hogy egyidőben csak egy gomb lehet benyomva. Mi történik, ha egyszerre két gombot nyomunk le? Jellemezze néhány példán keresztül!
- Írja fel a dekódoló négy logikai függvényét algebrai alakban! Hogyan valósítja meg ezeket az áramkör?

#### 4. mérés: Multiplexer, demultiplexer

A digitális jelfeldolgozás során gyakran előfordul, hogy több vezetéken bejövő információt kell továbbítani egy vezetékre, időbeli eltolással. Ilyenkor egy olyan áramkörre van szükség, mely megfelelő vezérlő információ alapján a kívánt bejövő vezeték kiválasztja és csak azt kapcsolja rá a kimenetre. Ugyanennek a fordítottja is előfordul, amikor egy bejövő vezeték kell adott információ alapján a több lehetséges kimenővezeték egyikére kapcsolni. Szemléletes gyakorlati példa erre a vasúti állomás váltója, mely az egy irányból beérkező vonatokat irányítja a megfelelő vágányra, ill. induláskor a váltó állása szabja meg, hogy melyik vágányról indulhat a kifelé vezető irányba vonat. A vezeték itt a sínek helyettesítik, a továbbítandó adatot (információt) a vonatok, az irányítási információt pedig a menetrend tartalmazza.

##### Multiplexer

A multiplexer lényegében egy sor kapcsoló, amelyek több bemenetet (adatbemenetek) kötnek össze egy kimenettel, mégpedig úgy, hogy egy megfelelő vezérlés alapján időbelileg meghatározva egyszerre csak egy bemeneti vezeték kapcsolnak a kimenetre.



1. ábra. multiplexer és demultiplexer működési vázlat

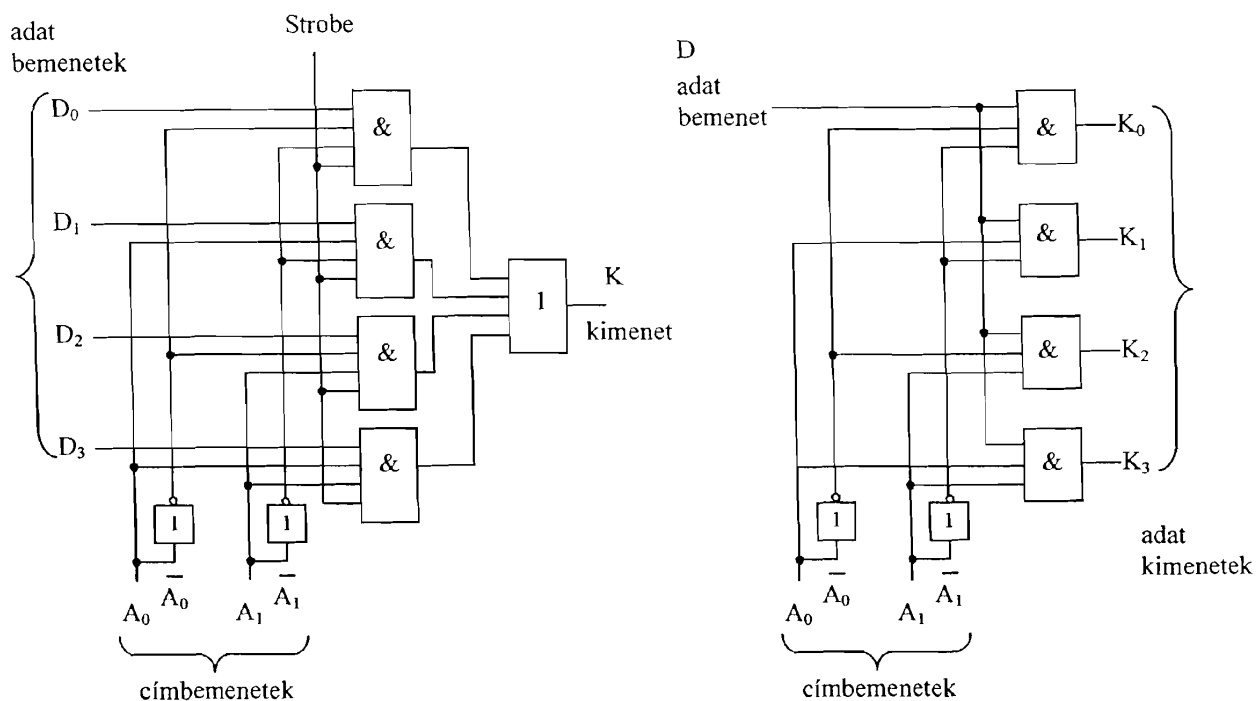
A multiplexer működése egy  $m$  állású fokozatkapcsoló működésének felel meg. Segítségével lehetővé válik  $m$  db különböző jelet, melyek párhuzamosan  $m$  db vezetéken vannak jelen, időmultiplex módon - azaz időben egymás után - egy db kimeneti vezetékre kapcsolni. Az egyes bemenetek kimenetre kapcsolásának időrendje tetszőleges is lehet. Többnyire azonban a címbemenetekre adott vezérlő jel egy számláló által szolgáltatott kódszó-sorozat szokott lenni és ebben az esetben az adatbemenetek sorrend szerint ciklikusan kapcsolódnak rá a kimenetre.

A digitális jeleknél "kapcsoló"-ként egyszerű AND kapuáramköröket használunk (2. ábra). Mindegyik bemeneti adatvezetékét rávezetjük egy AND kapu egyik bemenetére.  $m$  db bemeneti jelvezeték esetén tehát  $m$  db AND kapura van szükség. Az AND kapuk kimeneteit egy utánuk következő OR kapu fogja össze. Az AND kapukat kell vezérelni, mégpedig úgy, hogy egyidőben csak egy kapu nyisson ki. A kinyitást (vezérlést) egy adott időintervallumban a kapu adatbemeneten kívüli bemeneteire adott egyidejű H szintű jelekkel érhetjük el.

A vezérlést általában az  $un$ . címbemenetekeken ( $A = \text{Adress}$ ) keresztül binárisan kódolva valósítjuk meg. Ez azt jelenti, hogy az  $m$  db állapot megkülönböztetéséhez legalább  $n$  db

cím-bemenet szükséges, ahol  $m=2^n$ . A vezérléshez az  $m$  darabszámú AND kapuk mindegyikének rendelkeznie kell  $n$  db ( $m \leq 2^n$ ) cím-bemenettel. Ezenkívül egy közös un. strobe-bemenet (tiltó bemenet) segítségével minden bemeneti vezeték le lehet választani a kimenetről, a cím-bemenetektől függetlenül.

Ezzel a módszerrel pl. mintavételező kapcsolások vagy párhuzamos-soros-átalakítók valósíthatók meg.



A <sub>1</sub>	A <sub>0</sub>	K=
0	0	D <sub>0</sub>
0	1	D <sub>1</sub>
1	0	D <sub>2</sub>
1	1	D <sub>3</sub>

A <sub>1</sub>	A <sub>0</sub>	D=
0	0	K <sub>0</sub>
0	1	K <sub>1</sub>
1	0	K <sub>2</sub>
1	1	K <sub>3</sub>

2. ábra A multiplexer és demultiplexer kapcsolása, működését leíró táblázat

a) Multiplexer

b) Demultiplexer

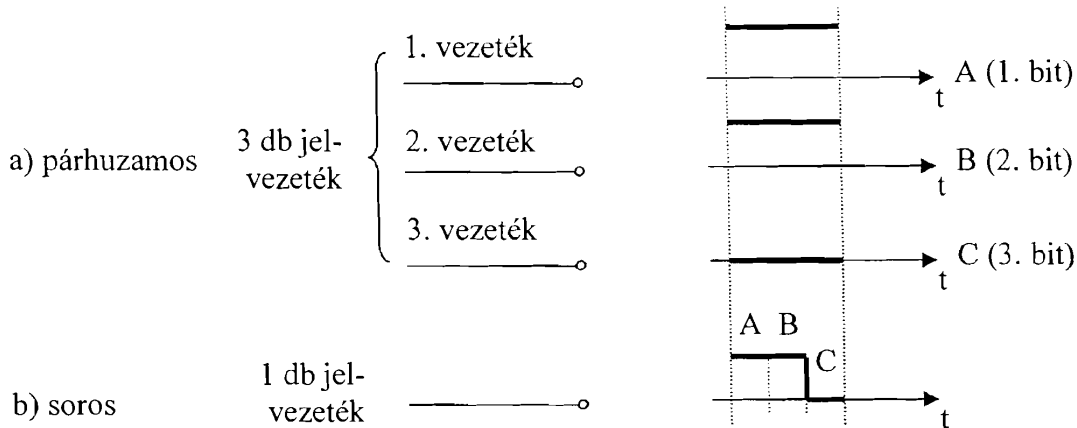
A multiplexert egy egyfokozatú dekódolónak is fel lehet fogni, amennyiben minden egyes dekódoló kapuhoz pluszban egy adatvezeték is becsatlakozik és a dekódoló minden kimenete egy OR kapuval össze van fogva.

A **demultiplexer** a multiplexer fordítottját végzi. Feladata általában az szokott lenni, hogy az egy db vezetéken időben egymás után érkező adatokat több kimeneti vezetékre ossza el. Minden dekódoló áramkört egyben demultiplexerként is lehet használni, amennyiben minden egyes dekódoló-kapuhoz egy közös adatbemenetet csatlakoztatunk.

Léteznek olyan multiplexerek, melyek nem csak egyesével választhatnak ki és kapcsolhatnak tovább beérkező adatvezetékeket egyetlen kimenetre, hanem több vezeték (bitet) összefogva, csoportosával is képesek ugyanezt végrehajtani. A mikroprocesszoros technikában jelentős a 4-szer 8-ból 8 (4x8-ból 8) multiplexer, mely egyszerre 8 vezeték kapcsol és 4 db 8 bites szó egyikét tudja vezérlés alapján (négyállású fokozatkapcsolóhoz hasonlóan) a 8 bites kimenetre adni.

**Párhuzamos és soros adatábrázolás, adattovábbítás**

Egy bináris jel (bit) kétféle állapotot vehet fel, az általa képviselt információ az információ alapegysége. Több bináris jel (bit) összefogásával nagyobb a variációk száma, ezzel a képviselt információ mennyiség is. Több bit-es elrendezéseket kódszónak nevezzük. Az egyes biteknek kétféle ábrázolási ill. továbbítási módja lehetséges: párhuzamos és soros, lásd 3. ábrát.



3. ábra: 3 bitből álló kódszó a) párhuzamos, b) soros ábrázolása

A soros adattovábbításnál csak 1 db jelvezetékre van szükség. Az egyes bitek időben eltolva,  $n$  bit esetében a párhuzamos ábrázolásmóddhoz képest  $n$ -szeres órajel-frekvenciával jelennek meg a vezetéken.

Ha adatokat akarunk továbbítani az egyik digitális egységtől a másikig, nem mindegy, hogy hány vezetékkel kell összekötnünk őket, illetve milyen gyorsan lehet egy adott számú bitet az adott jelvezeték(ke)ken továbbítani (átadó egység a jelet a kimenetén ráadja a vezetékre, mely az átvevő bemenetére kapcsolódik). A párhuzamos adattovábbítás gyors, viszont a több vezeték miatt igényesebb kialakítású és a külső elektromágneses zavarokra is érzékenyebb. A soros adattovábbítás hátránya a lassúság.

**Feladat, 4. mérés:**

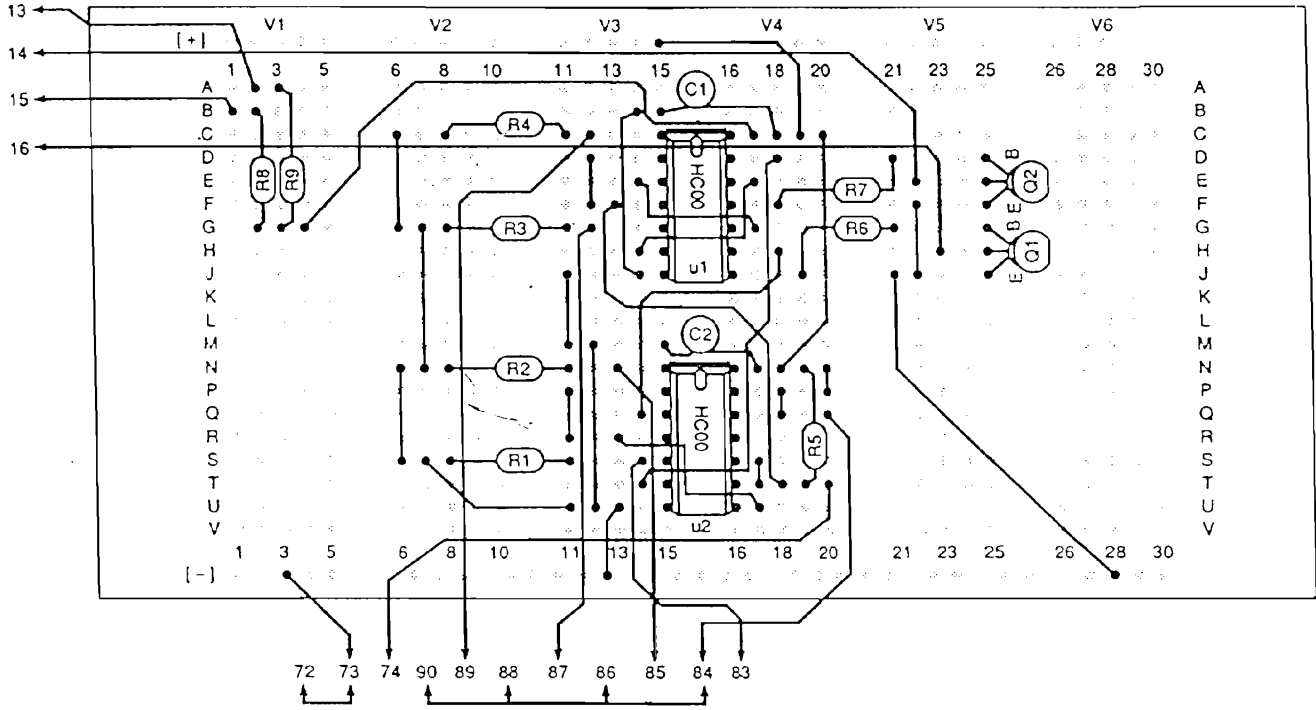
(179)

- Az alábbi kapcsolási rajz alapján egy 2-bemenetű kettős multiplexert lehet összeállítani. Az áramkör egy bementi információ alapján a bemenő négy adatvezetékéből kettőt-kettőt kapcsol egyszerre a két kimenő adatvezetékre. Mivel itt két lehetőségből kell egyet kiválasztani címként elegendő egyetlen jelbemenet (1 bit), amit a kétállású kapcsoló valósít meg. Vizsgálja meg az áramkör működését az összes előfordulható esetre kiterjedően és rögzítse a jegyzőkönyvben alkalmas módon.
- Alakítsa át a kapcsolást dekódoló nélküli 4:1 multiplexerré, azaz a bemenetválasztást **további 4 kapcsoló** végezze !

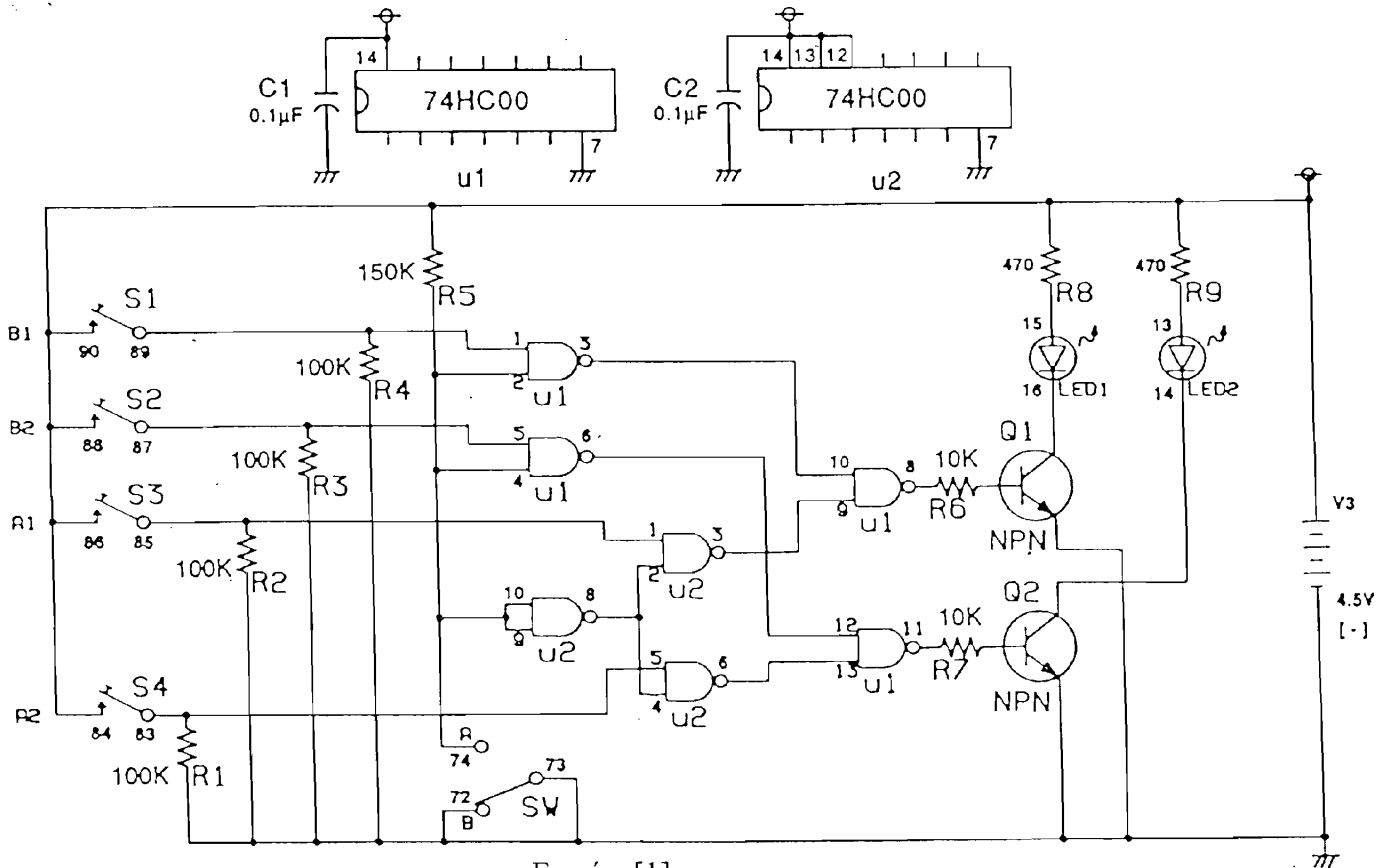
**Változtatások:**

- Ki kell iktatni: R5, SW, U2 8-as kimenete, LED2, R9
- A 4 bemenetválasztó kapcsoló mindegyikénél:
  - egyik pólus a tápfeszültségre,
  - a másik pólus az adott NAND kapura (U1 és U2 2-es és 4-es lábai) és egyúttal egy ellenálláson keresztül GND-re kapcsolódik.
- Q2 kollektorát kössük párhuzamosan Q1 kollektorával (OR függvénykapcsolat) !

Írja le az áramkör működését!



U1	74HC00	R1	100KΩ	R4	100KΩ	R7	10KΩ	C1	0.1μF
U2	74HC00	R2	100KΩ	R5	150KΩ	R8	470Ω	C2	0.1μF
Q1	NPN	R3	100KΩ	R6	10KΩ	R9	470Ω		
Q2	NPN								



Forrás: [1]

## 5. mérés: SR-tároló

A bináris tárolók a szekvenciális hálózatok jellemző elemei. A tárolók a nevükből eredően digitális, jellemzően egy elemi bináris adat, azaz egy bit tárolására alkalmasak. Másképpen bistabil (két stabil állapottal rendelkező) billenő-kapcsolásnak, vagy FlipFlop-nak is nevezik őket.

A bináris tárolók egy vagy két jelbemenettel valamint rendszerint egy órajel-bemenettel és két kimenettel ( $Q$  és  $\bar{Q}$ ) rendelkeznek.

Sok bináris tároló rendelkezik még egy beállító és egy nullázó bemenettel is, melyeken keresztül a kimeneti állapotot bármikor és az órajeltől függetlenül közvetlenül be lehet állítani (Clear és Preset). Ezesetben TTL áramköröknél az alábbi jelszintek érvényesek:

Clear = L (és Preset = H)  $\rightarrow$   $Q = L$ ,

Preset = L (és Clear = H)  $\rightarrow$   $Q = H$ .

Clear = Preset = H esetén ezek a bemenetek hatástalanok. A Clear = Preset = L állapot a FlipFlop típusától függően eredményez  $Q = \bar{Q} = H$  vagy  $Q = \bar{Q} = L$  kimeneteket és ezért ez az állapot nincs megengedve.

A bináris tárolók jelbemenetük szerint csoportosítva lehetnek

- SR tároló
- JK tároló
- D tároló
- T tároló

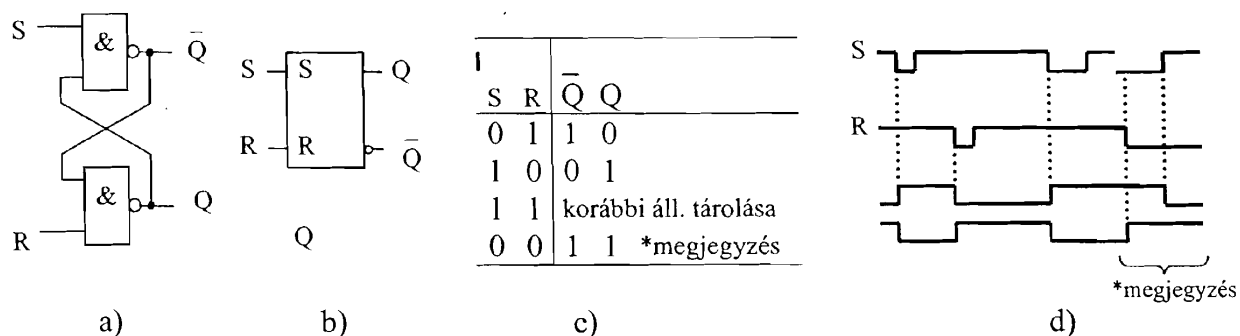
illetve az órajel-vezérlés szerint csoportosítva

- órajel-vezérlés nélküli tárolók (nincs órajel bemenetük)
- szintvezérelt tárolók, Latch-ek (az "órajel" szintje, állapota által vezérelt)
- élvezérelt tárolók (edge-triggered)
- Master-Slave-tárolók

Az órajel- (vagy más néven ütem-) bemenet arra szolgál, hogy a tároló kapcsolását, átbillenését egy adott időponthoz kössük. Egyes áramköröknél az órajel felfutó, másoknál a lefutó éle váltja ki a FlipFlop kapcsolását. A többnyire T-vel, vagy C-vel (clock) jelölt órajel-bemenetek lehetővé teszik több FlipFlop egyidejű kapcsolását, ezáltal pl. szinkron léptetőregiszterek, számlálók valósíthatók meg velük.

**SR-tároló**

Az órajel-vezérlés nélküli SR-tároló a legegyszerűbb bináris tároló. Egyik megvalósítási formája a 2 db NAND kapuból felépített tároló, melyek visszacsatolásokkal vannak egymással összekötve (lásd 1. ábrát).



1. ábra: A két NAND kapuból álló SR-FlipFlop kapcsolása, rajzjele, igazságtáblázata és impulzusdiagramja

A bemeneteket jelölő S és R a „set” (beállítás, azaz a tárolóba „1” beírása) és a „reset” (visszaállítás, azaz törlés) műveletét jelenti.

Ha (pozitív logikát feltételezve)  $S=1$  és  $R=0$ , akkor a FlipFlop, azaz a Q kimenet 1 állapotba kerül és  $\bar{Q}=0$  értékű lesz. Ez a beállítás, vagy másképpen beírás. Ha  $S=0$  és  $R=1$ , akkor a törlés funkció működik, és  $Q=0$ ,  $\bar{Q}=1$  lesz. Ha  $S=R=1$ , ekkor mindkét NAND kapunál a visszacsatoló ág dominál: a FlipFlop ilyenkor megtartja korábbi állapotát, azaz 1 bit információt tárol

\* Ha az S és az R bemeneten is "L" potenciál van, akkor mindkét NAND kapu „zár” és ezáltal mindkét kimenet "H" szintre kerül. Ekkor a két kimenet már nem egymásnak a negáltja, ahogy azt eredetileg elváránk. Ilyen esetben set és reset egyszerre aktív, ami bár logikailag ésszerűtlen, viszont a kapcsolás működése eddig egyértelmű és meghatározott. Gond akkor keletkezik, ha ebben az állapotban mindkét bemenet *egyidőben, egyszerre* vált át "H" szintre: ekkor a FlipFlop elveszti tároló tulajdonságát és véletlenszerűen áll be a kimenet a két állapot ( $Q = H / \bar{Q} = L$  vagy  $Q = L / \bar{Q} = H$ ) valamelyikére. Emiatt ennél a kapcsolásnál nincs megengedve az az eset, amikor mindkét bemenet egyidejűleg "L" potenciálon van. *(Ha külön gondoskodunk arról, hogy egyidejűleg ne válthasson mindkét bemenet "H" szintre, akkor biztosítva van a FlipFlop mindenkor definiált, egyértelmű állapota és ez az eset is megengedett.)*

A fentiekben ismertetett SR-FlipFlop-ot például kapcsolók pergésmentesítésére, vagy olyan áramkörökben használják, ahol egyszeri lefolyású, rövid impulzusokat (zavarjeleket) kell regisztrálni.

### SR-Latch

Az SR-FlipFlopok órajel-bemenettel, tehát órajel-vezérléssel nem rendelkeznek. Létezik viszont egy olyan SR-FlipFlop-változat, amelyiknél az S és R bemenetek elé egy-egy NAND kapu van bekötve és mindkét kaput egy közös enable- azaz engedélyező-bemenet vezérli. Az engedélyező bemenetére adott H szint nyitja a bemeneti kapukat, ezen idő alatt az SR-FlipFlop a szokásos módon működik (pontosabban ilyenkor az SR bemenetek "H" szintje aktív). Az engedélyező bemenet L szintje lezárja a Latch-t, ezen idő alatt az SR bemenetek változásaitól függetlenül a lezárás időpontjában fennálló állapot tárolódik. Az SR-en kívül létezik D-Latch is. Latch-FlipFlopok csak adatok ütemezés nélküli felfogására, átmeneti tárolására alkalmasak. Léptetőregisztert vagy számlálót nem lehet ezekből felépíteni. SR-Latch-al viszont megvalósítható a Master-Slave működési mód. A "Latch"-vezérlés és az órajelvezérlés között az a különbség, hogy a Latchnál a FlipFlop-ot közvetlenül az információ-bemenetek billentik be, míg az az élvezérlésnél az órajel hatására billen.

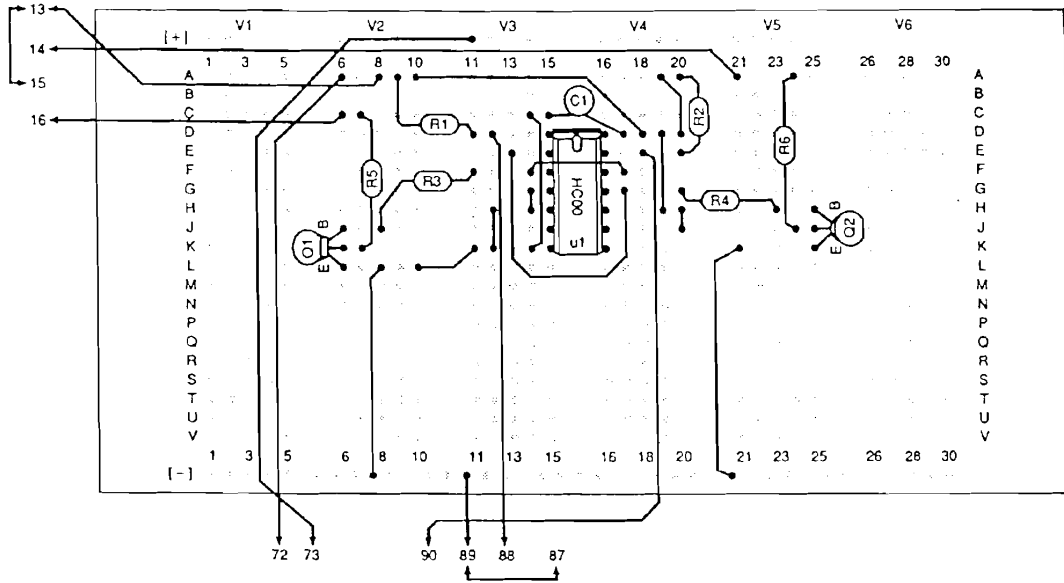
Integrált áramköri formában SR-FlipFlop többnyire csak Latch változatban fordul elő.

**Feladat, 5 mérés:** SR-FlipFlop vizsgálata

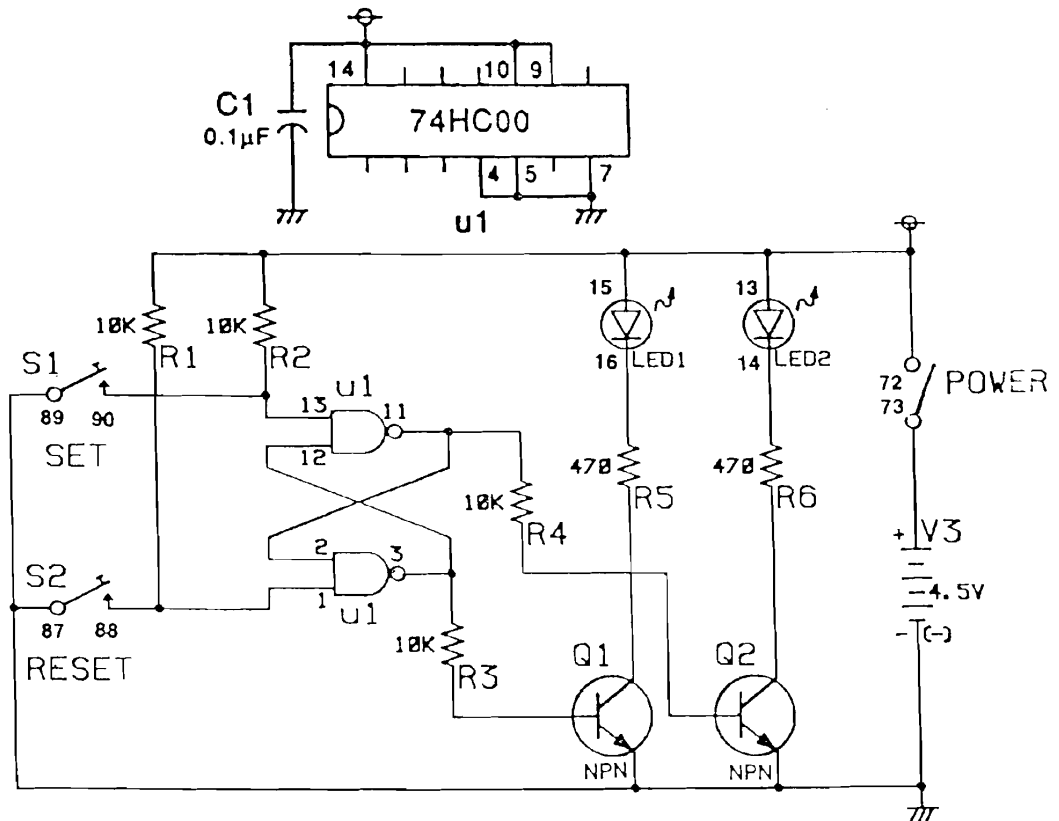
(158)

Állítsa össze az alábbi kapcsolást! A tápfeszültség bekapcsolása után adja rá a FlipFlop bemeneteire kapcsolók segítségével az összes jelkombinációt és figyelje a kimeneteket jelző LED-eket!

- Az igazságtáblázat és egy idődiagram felvételével igazolja az elméletileg várt működést! Mitől függ, hogy milyen állapotot vesz fel a FlipFlop a tápfeszültség bekapcsolásakor?
- Készítsen SR-Latch FlipFlop-ot egy harmadik kapcsoló (és ellenállás) és az IC két, eddig nem használt NAND kapujának felhasználásával! Írja, vagy rajzolja le a szükséges változtatásokat és elemezze az áramkör gyakorlati működését! A két engedélyező NAND kapu közül melyik kapcsol gyorsabban?



U1	74HC00	R1	10KΩ	R4	10KΩ	C1	0.1μF
Q1	NPN	R2	10KΩ	R5	470Ω		
Q2	NPN	R3	10KΩ	R6	470Ω		



Forrás: [1]



## 6. mérés: JK-tároló

## JK-tároló

A JK-tárolók az SR-tárolókkal ellentétben mindig ütemfüggő (órajel-vezérelt) tárolók és a két kimeneten túlmenően mindig van

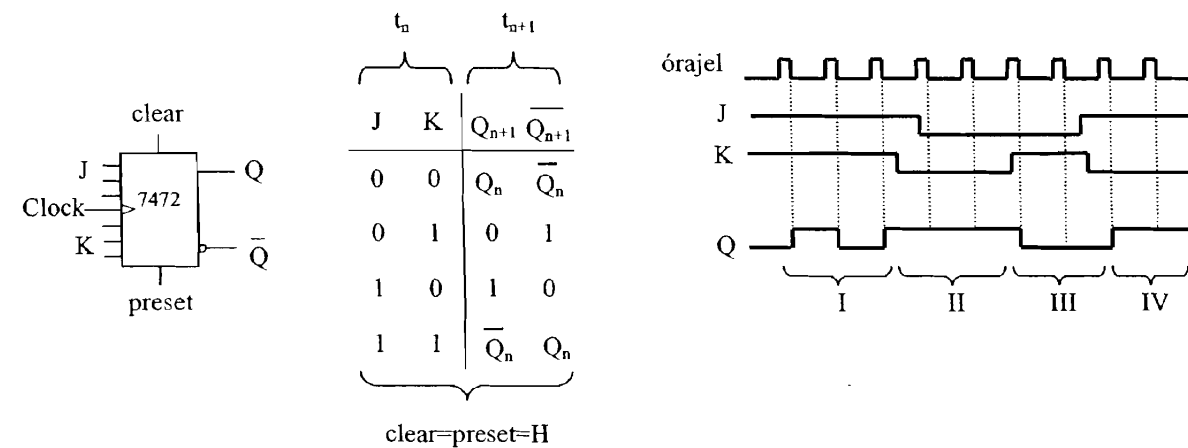
- 1 órajel-bemenetük
- 1 vagy 2 adatbemenetük (J,K) és
- 0 – 2 közvetlen állító-bemenetük (clear, preset)

Az ütemvezérlés esetében az adatátvétel típusától függően beszélünk

- Master-Slave tárolókról és
- élvezérelt tárolókról.

Az ütemjel-vezérlés miatt a J és K bemenetekre adott jelek meghatározzák („előkészítik”) azt az állapotot, amelyikbe a FlipFlop a következő vezérlő élnél át fog billenni.

Az RS FlipFlop előzőekben tárgyalt változatához képest a  $J=K=1$  bemeneti állapot minden korlátozás nélkül megengedett.



$t_n$ -nel jelöljük a hatásos órajel-él előtti időt,  $t_{n+1}$ -el a hatásos órajel-él utáni időt (állapotot)

1. ábra: Az élvezérelt JK-tároló rajzjele, igazságtáblázata és impulzusdiagramja

Az 1. ábrán általános JK-tároló rajzjele, igazságtáblázata és impulzusdiagramja látható. A három J bemenet belül AND kapuval van összekapcsolva ugyanígy a három K bemenet is. Ez TTL áramkörök esetében azt jelenti, hogy a be nem kötött, nyitott bemenetek H-szinten vannak és ezért nem befolyásolják a működést.

Mivel a tároló élvezérelt (jelen esetben a lefutó él a hatásos), a tároló kimenete mindig csak az órajel lefutó élének időpillanatában válthat állapotot. A tároló a  $J=K=0$  esetben nem vált állapotot, hanem megmarad előző állapotában. Ha a bemenetek H szintűek, azaz  $J=K=1$ , akkor az ütemjel minden egyes hatásos éle a tároló átbillenését eredményezi. (Ez a második eset egyébként egy 2:1 arányú frekvenciaosztót valósít meg.)

A 1. ábra impulzusdiagramja is leírja a tároló működését:

- I. A JK bemenetek nyitottak vagy H potenciálon vannak, a tároló (Q kimenet) minden egyes hatásos órajel-él-re állapotot vált.
- II.  $J=K=0$  esetén a FlipFlop az éppen meglévő állapotot tárolja, a lefutó él nem tudja átbillenteni.

- III.  $J=0$  és  $K=1$  esetén a Q kimenet a következő órajel-él hatására "L" szintre (0-ra) vált és változatlan bemenetek mellett itt is marad.
- IV.  $J=1$  és  $K=0$  esetén a Q kimenet a következő órajel-él hatására "H" szintre (1-re) vált és itt is marad.

### JK Master-Slave tároló

#### Master-slave működési elv:

A master-slave tárolók két db, sorba kötött élvezérelt tárolóból állnak, melyek ugyanazt az - de egymáshoz képest negált - órajelet kapják. Az első a master (mester), ez már az órajel aktív felfutó élével (vagy szintvezérlés esetén azon idő alatt, amikor az órajel H szinten van) átveszi a J és K bemeneteken lévő külső információt és azt tárolja. Az órajel LH (felfutó) éle tehát kinyitja a master-t és egyidejűleg lezárja a master után kötött második tárolót, a slave-t (szolgát). A slave tároló csak az órajel HL élével veszi át a master-től az információt, mikor is a master bemenete lezáródik és a slave bemenete nyílik ki. A slave tároló tartalma egyben a Q kimeneten is megjelenik. A külső JK (adat)bemeneteket egyébként előkészítő bemeneteknek is szokták nevezni.

A master-slave elv előnye, hogy az információ felvétele és továbbítása különböző ütemeleknél következik be, azaz egyidejűleg új információt lehet a bemenetre adni (és felvetetni) és a kimeneten ugyanakkor még a régi információt le lehet kérdezni. Hátránya szintvezérelt flipflop-ok esetében, hogy az ütemjel H szintjének ideje alatt a kapcsolás a JK bemeneteken megjelenő pozitív zavarjelekre érzékeny. A TTL áramkörsládban léteznek olyan megoldások is, melyek ezt a zavarérzékenységet egy ún. JK-zárral (data lockout) kiküszöbölik, és csak a felfutó ütemél idejében tudja a bemeneteken meglévő jel a master állapotát befolyásolni. A csak simán élvezérelt JK (és D) tárolók esetében ez a probléma nem áll fenn, mivel ezek eleve csak a hatásos él időpontjában billenek be.

#### *Feladat, 6 mérés:* JK-FlipFlop vizsgálata

(167)

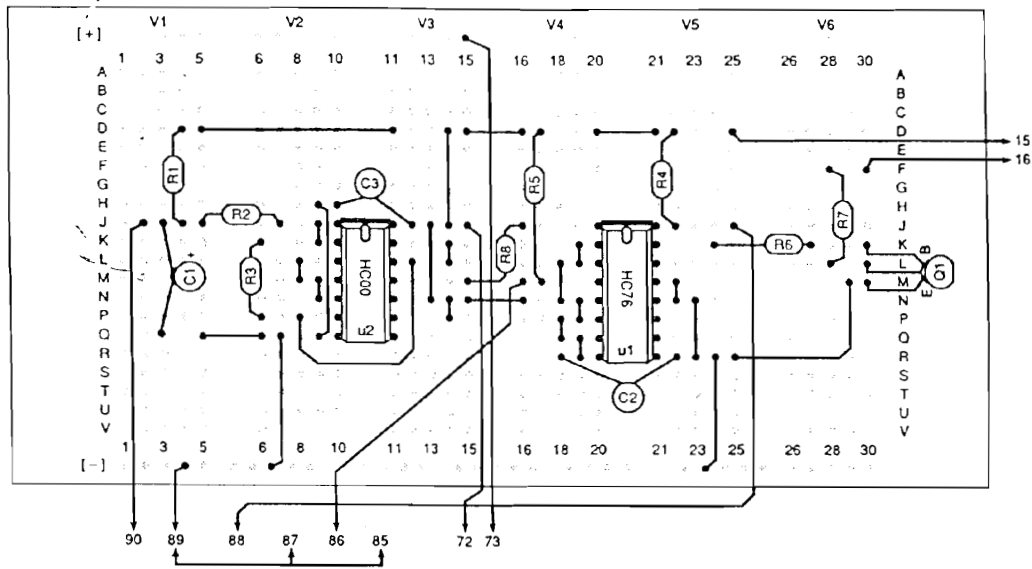
- Állítsa össze az alábbi kapcsolást! A tápfeszültség bekapcsolása után adja rá a FlipFlop bemeneteire kapcsolók (S2, S3) segítségével az összes JK jelkombinációt és az S1 kapcsoló nyomogatásával biztosítsa az órajelet! A kimenetet jelző LED figyelésével igazolja az elméletileg várt működést (igazságtáblázat és egy idődiagram felvétele)!

[A C1 kondenzátorral és az U2 IC-vel megvalósított kapcsolás az órajelet adó mechanikus kapcsoló pergésmentesítését és az órajel alakjavítását (élmeredekség növelését) végzi. Miért van erre szükség az órajel-bemeneteknél?]

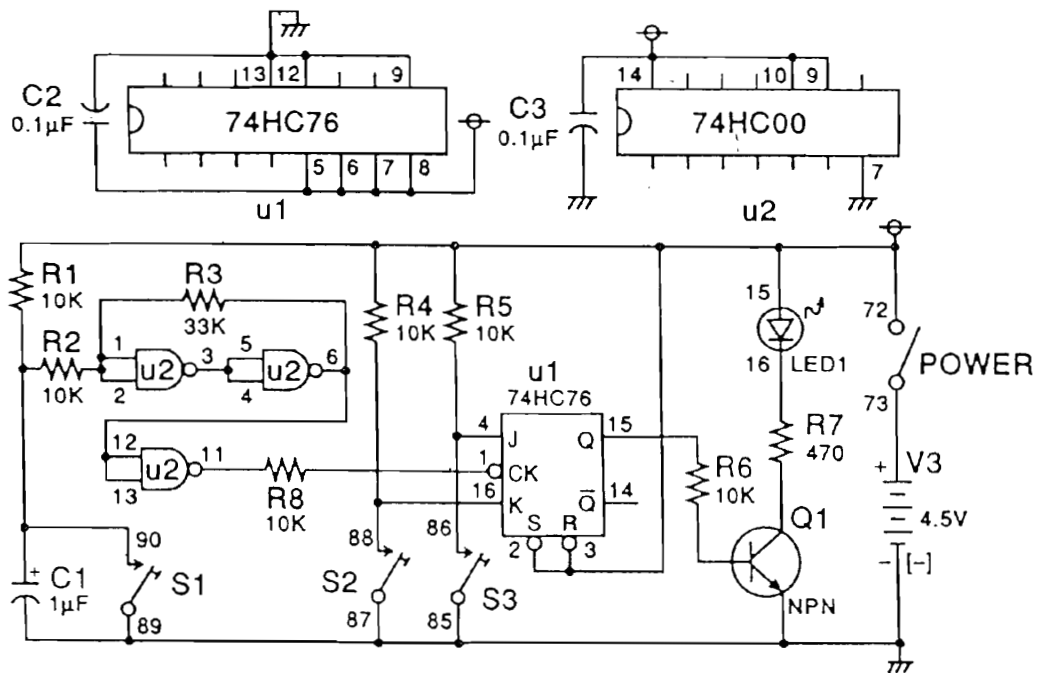
- Az S (preset) és R (clear) jelű bemenetek funkcióit ellenőrizze közvetlen tápfeszültség ill. GND ráadásával!
- Két mérőbőrönd felhasználásával állítson elő Master-Slave JK-flipflop-ot!
- A master felfutó órajellel veszi át a külső információt.
- A slave az órajelét a master-mérőbőrönd U2 12-es lábáról kapja saját R8-án keresztül.
- A master-flipflop Q-ját a slave J bemenetére, a  $\overline{Q}$ -ját a slave K bemenetére kösse! (R4-et és R5-öt a tápfeszültségről leválasztani)

Vizsgálja meg és rajzolja le idődiagramban, hogyan megy át a master és slave

1	1	állapotból	0	0	állapotba, ill.
0	0	állapotból	1	1	állapotba!



U1	74HC76	R1	10KΩ	R4	10KΩ	R7	470Ω	C1	1μF
U2	74HC00	R2	10KΩ	R5	10KΩ	R8	10KΩ	C2	0.1μF
Q1	NPN	R3	33KΩ	R6	10KΩ			C3	0.1μF

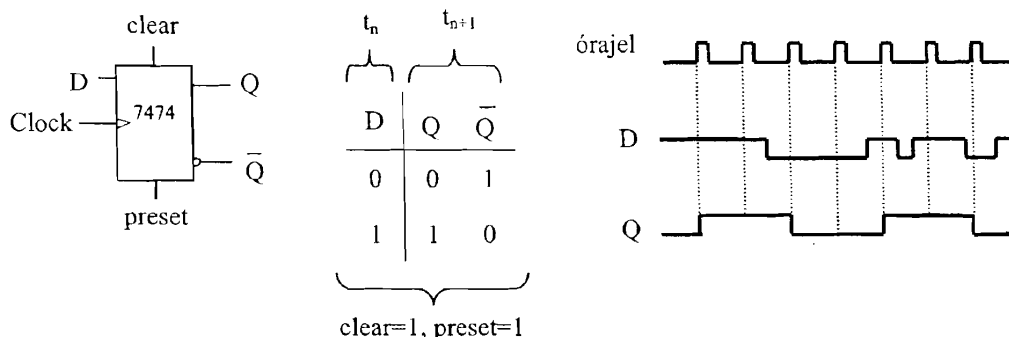


Forrás: [1]

## 7. mérés: D-tárolók, T-tárolók

**D-tároló**

Az órajel-vezérelt D-tárolónak csak egy, D-vel jelölt adatbemenete (információ-bemenete) van. Az élvezérelt D-tároló rajzjele (például 7474-es TTL IC), igazságtáblázata és impulzusdiagramja az 1. ábrán látható.



$t_n$  a hatásos órajel-él előtti,  $t_{n+1}$  a hatásos órajel-él utáni időt (állapotot) jelenti

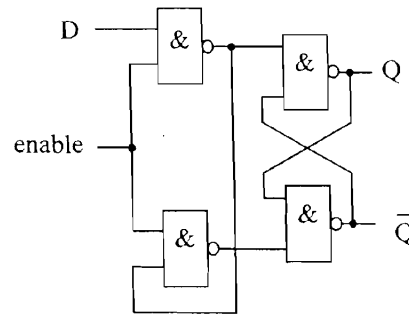
1. ábra: Az élvezérelt D-tároló rajzjele, igazságtáblázata és impulzusdiagramja

Működése a következő: Minden hatásos órajel-élnél (a 7474-esnél ez a felfutó él) a tároló átveszi az ebben az időpillanatban a D bemeneten lévő információt és az megjelenik a Q kimeneten (mégpedig attól függetlenül, hogy azelőtt milyen jel volt rajta) és azt a következő hatásos élig tárolja. Az adatbemenet két hatásos órajel-él közötti változásai nincsenek befolyással a tároló állapotára.

- Az impulzusdiagramból kitűnik, hogy ha  $t_n$  időpontban adunk egy jelet a D bemenetre (mely aztán nem változik), az a kimeneten csak a következő hatásos órajel-él bekövetkeztekor, a  $t_{n+1}$  időpontban fog megjelenni. Ez egyfajta késleltetést jelent. A D az angol „delay” kifejezés rövidítése. E tulajdonsága miatt a D-tároló különböző folyamatok szinkronizálásra, léptetőregiszterek felépítésére különösen alkalmas.
- A jelvezérelt D tároló működése azon alapszik, hogy a kapcsoláson belül az órajel egy késleltetést szenved el. A felfutó órajel a bemenet és a kimenet közötti utat rövid időre (amíg az adatjel a kimenetig eljut) felszabadítja és ez az út ezután a késleltetett órajel-él visszacsatolása révén ismét lezárul.

**D-Latch**

A 2. ábra a D-Latch egyik megvalósítási módját ábrázolja NAND kapus SR-Latch-ból kialakítva. Az engedélyező bemenetére adott H szint nyitja a bemeneti kapukat, ezen idő alatt a FlipFlop úgy működik, hogy a D-bemenetre adott információ (H vagy L szint) közvetlenül megjelenik a kimeneten, azaz ekkor a tároló „permanens”. Az engedélyező bemenet L szintje lezárja a Latch-t, ezen idő alatt a lezárás időpontjában fennálló állapot tárolódik, a flipflop „befagy”. A D-bemenet időközbeni változásai hatástalanok. Latch-FlipFlopok csak adatok ütemezés nélküli felfogására, átmeneti tárolására alkalmasak. Léptetőregisztert vagy számlálót nem lehet ezekből felépíteni (l. 8. mérés). A "Latch"-vezérlés és az órajelvezérlés között az átbillenés időpontjában van a különbség: a Latchnál a FlipFlop-ot közvetlenül az információ-bemenet változása billenti be, míg az élvezérlésnél a FlipFlop csak a hatásos órajel-él megjelenésekor billen.

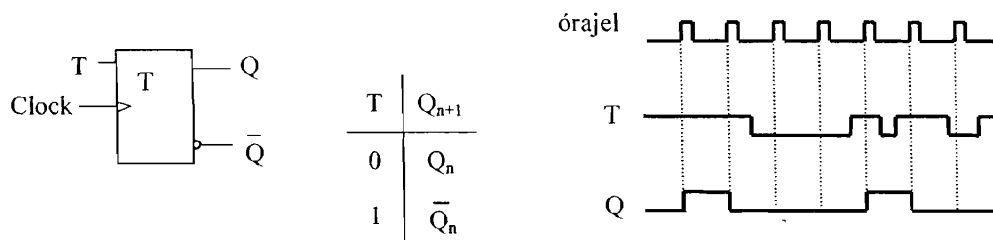


2. ábra: D-Latch felépítése NAND kapukból

### T-tároló

A T-tárolónak az órajel-bemeneten kívül egy T-vel jelölt vezérlő-bemenete van, információ-bemenete nincs. A D tárolóhoz hasonlóan élvezérelt, azaz csak az órajel megváltozásának pillanatában (felfutó vagy lefutó él) lehetséges a tartalmának megváltozása. Működésének lényege, hogy ha a T-bemenet H szintű, a tároló minden egyes hatásos órajel-élre állapotot vált. Ha a T-bemenet L szintű, a tároló megőrzi korábbi állapotát. A T betű a trigger elnevezésre utal. Rajzjele, igazságtáblázata és impulzusdiagramja a 3. ábrán látható.

A T-FlipFlop-ot elsősorban frekvenciaosztókban és számláló áramkörökben alkalmazzák Integrált áramköri kialakításban ezt a tároló változatot önállóan nem gyártják, miután a JK típusból külső kötéssel ( $J=K="H"$ ) kialakítható.



$Q_n$  a hatásos órajel-él előtti,  $Q_{n+1}$  a hatásos órajel-él utáni tároló-állapotot jelenti

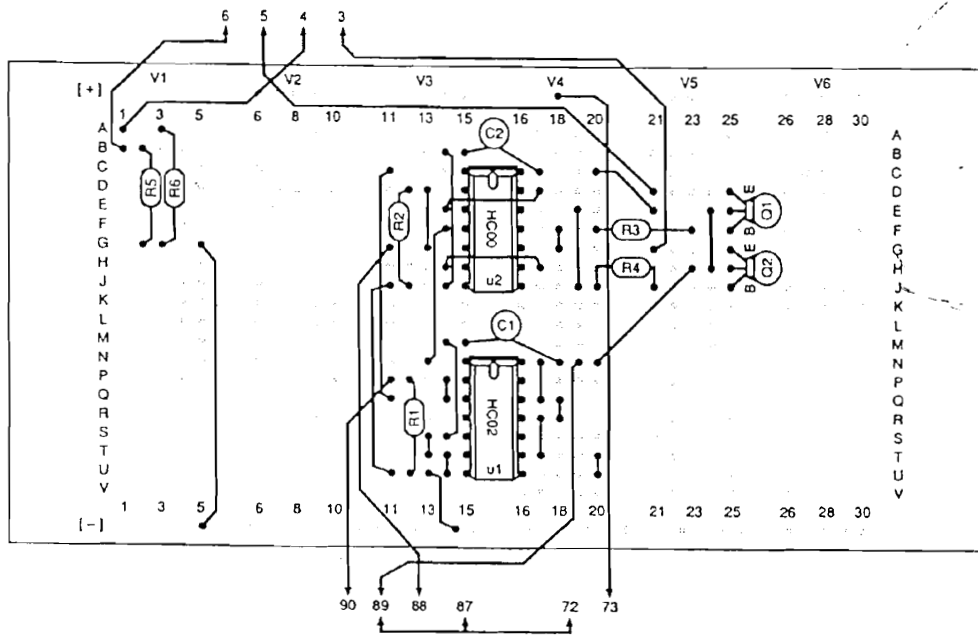
3. ábra: Az élvezérelt T-tároló rajzjele, igazságtáblázata és impulzusdiagramja

Az eddig tárgyalt tároló-típusok egymásból, (főleg az alapnak számító SR-ből és az integrált formában kapható JK-ból) felépíthetők.

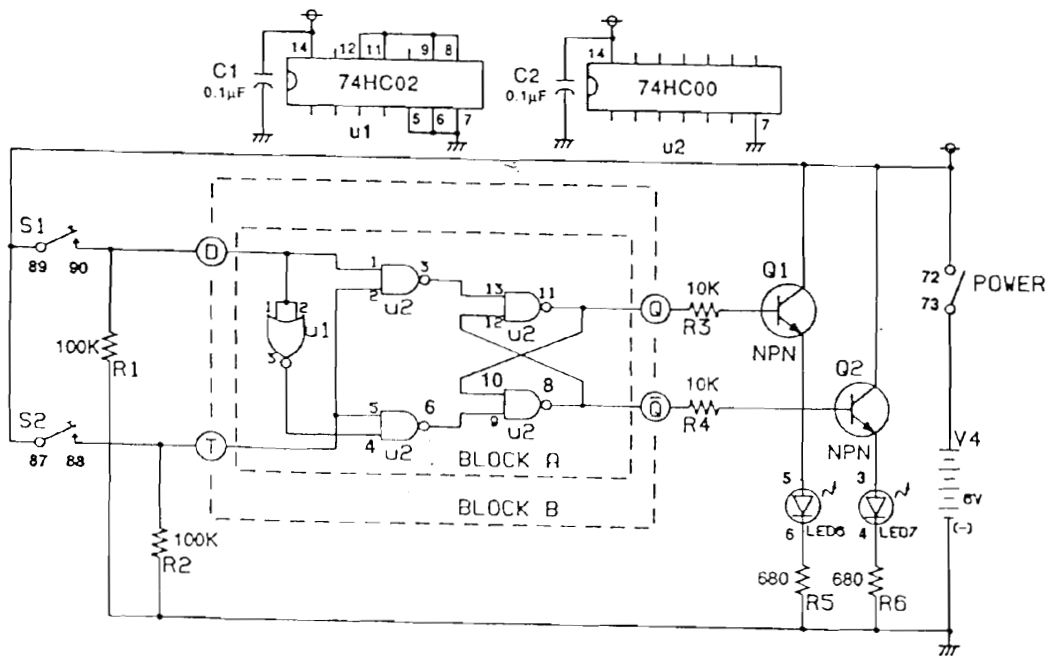
### Feladat, 7. mérés: D-Latch tároló

(176)

- NAND kapuk segítségével D-Latch tároló felépítése az alábbi kapcsolási rajz szerint. Az S1 kapcsolóval a D-bemenetet tudjuk beállítani, S2-vel pedig az engedélyező bemenetet. A kimenetek értékeit LED-ek jelenítik meg. Igazolja a D-Latch működését egy alkalmas idődiagram felvételével!
- Két mérőkapcsolás felhasználásával készítsen Master-Slave Flip-Flop-ot (azaz élvezérelt D FlipFlop-ot) és alkalmas idődiagram felvételével írja le működését! (Két D-Latch sorbakötése, közös GND, egy adat-bemenet, egy kimenet, közös engedélyező órajel invertálással)
- Vizsgálja meg, mely esetben és milyen helytelen működést okozhat a kimeneten, ha két hatásos órajel-él között a bemeneten rövididejű H- vagy L-szintű hibaimpulzus keletkezik? (hibaimpulzus szimulálása, idődiagram felvétele)



U1	74HC02	Q1	NPN	R1	100KΩ	R4	10KΩ	C1	0.1μF
U2	74HC00	Q2	NPN	R2	100KΩ	R5	680Ω	C2	0.1μF
				R3	10KΩ	R6	680Ω		



Forrás: [1]

## 8. mérés: Regiszterek, léptetőregiszter

A több-bites **regiszterek** és tárolók (memóriák) bináris jelek tárolására szolgálnak.

A regiszter az elemi tárolók (Flipflopok) olyan szervezett összessége, amelynél a memóriatárolókkal ellentétben nincsenek címek, a tárolandó információ egyes bitjei egy adott időrendi sorrendben kerülnek be a regiszterbe (beírás) és kerülnek ki onnan (kiolvasás). Az egyes bitek címét (azaz a helyét) a regisztereknél a beírásnál kialakult időrendi sorrend helyettesíti.

A regiszter egyes tárolóelemei egymás mellett, sorban helyezkednek el. Gyakorlatilag **léptetőregiszternek** nevezünk minden olyan FlipFlop elrendezést, ahol a FlipFlop-ok egymás után, sorba vannak kötve és egy *közös órajel* vezérli őket. A sorba kötés azt jelenti, hogy minden FlipFlop (tároló) kimenete a sorban következő FlipFlop bemenetével van összekötve. A léptetőregiszterek főbb felhasználási területei:

- kicsi, de gyors tároló (számolási műveleteknél átmeneti tároló)
- adatok késleltetése és aszinkron jelek szinkronizálása
- soros/párhuzamos és párhuzamos/soros átalakítás

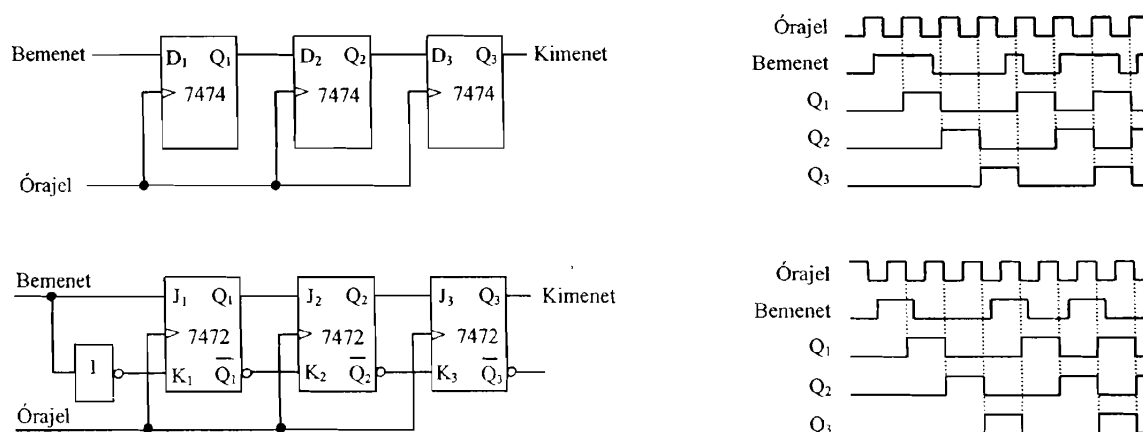
### Működés

A léptetőregiszter működése azon alapszik, hogy a FlipFlopok egyazon órajel-él hatására veszik fel (az előző FlipFlop-tól) és adják tovább (a következő FlipFlop-nak) az információt. A FlipFlop késleltetési ideje ebben az esetben az átmeneti tárolás szerepét tölti be: Az az idő, amíg a FlipFlop kimenete az órajel-él hatására az új értékre beáll elegendő ahhoz, hogy a következő FlipFlop ugyanezzel az órajel-éllel átvegye a régi, a hatásos órajel megjelenésének időpontjáig fennálló információt.

Ezáltal minden hatásos órajel-éllel az információ a léptetőregiszterben egy FlipFloppal arréb tolódik. (Ha az utolsó FlipFlop kimenetét visszavezetjük az első bemenetére, egy gyűrűregisztert, körszámlálót kapunk)

### Felépítés

Az 1. ábra D és JK FlipFlopokból felépített léptetőregiszterekre mutat példát.



1. ábra: 7474-es D-FlipFlopból és 7472-es JK-FlipFlopokból felépített léptetőregiszter és a hozzájuk tartozó impulzusdiagramok

Az impulzusdiagramból látható, hogyan tolódik jobbra minden egyes ütemjel-impulzusra (hatásos órajel-élre) az információ. A K bemenet előtti negátor gondoskodik arról, hogy a J- és K-bemeneteken mindig ellentétes jel legyen.

Az integrált áramkörökben többnyire SR-master-slave FlipFlopok vannak. A tárolók késleltetési ideje sok egyéb alkalmazásban nemkívánatos, a léptetőregiszternél viszont a működés elengedhetetlen feltétele. Ez vonatkozik az élvezérelt D tárolókból felépített léptetőregiszterre, ahol a késleltetési időnek meg kell haladnia egy minimális értéket. A JK tárolónál a master és a slave közötti átmenetei tárolás megoldja ezt a problémát. (A JK-master-slave tárolóknál viszont figyelni kell arra, hogy míg az órajel H-szinten van, a JK bemeneteknek nem szabad megváltozniuk.)

A léptetőregiszterek lehetnek *statikusak*, melyeknél az információ tetszés szerinti ideig megmarad órajel nélkül, és létezik *dinamikus* léptetőregiszter is, amelyeket csak egy bizonyos alsó frekvenciahatár fölötti órajel-frekvenciával szabad üzemeltetni és ezért mindig „mozgásban” kell lenniük. Létezik olyan léptetőregiszter is, melynek hosszát elektromosan be lehet állítani ill. változtatni.

Az egy bemenettel rendelkező léptetőregiszterek az információt csak soros formában vehetik fel, azaz egy n-bites kódszó beírásához n db órajel-impulzus szükséges. Ugyanez érvényes a kiolvasásra: Ha csak egy kimenet van, az információ csak sorosan, azaz az egyes bitek időben eltolva, egymásután adhatók ki.

A soros adatbemeneten és –kimeneten kívül a léptetőregisztert el lehet látni **párhuzamos be- illetve kimenetekkel** is. A bemenetek, ill. kimenetek számából megállapítható, hogy egy ismeretlen léptetőregiszternek van-e párhuzamos beírási ill. kiolvasási lehetősége. A csak soros be- és kimenettel rendelkező léptetőregiszterek felhasználási területe igen korlátozott, a gyakorlati esetek többségében szükség van párhuzamos bemenetre vagy kimenetre is.

A párhuzamos kimenettel rendelkező léptetőregiszterek segítségével a sorosan rendelkezésre álló kódszavakat át lehet alakítani párhuzamos formájúra és a párhuzamos bemenettel rendelkező léptetőregiszterekkel a párhuzamos kódszavakat lehet sorosan kiléptetni.

Párhuzamos üzemmódban minden egyes FlipFlop-ot el kell látni kívülről kivezetett bemenetekkel vagy kimenetekkel vagy mindkettővel.

### **Párhuzamos bemenetek**

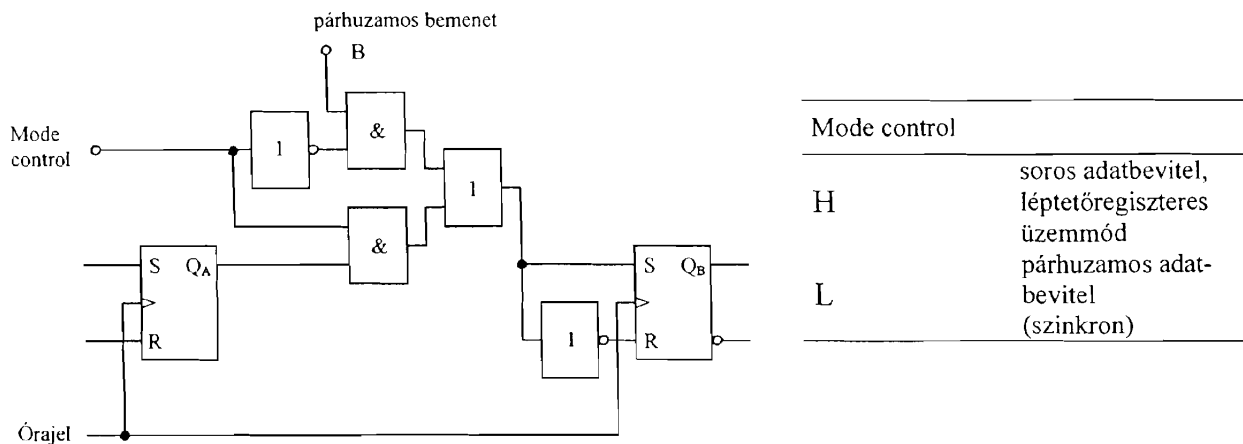
A 2. ábrán látható a párhuzamos bemenetek kialakításának egyik elve, a *szinkron párhuzamos bemenet*. A mode-control-bemenetre adott H vagy L szint szabja meg, hogy a megelőző FlipFlop vagy a kívülről csatlakoztatott párhuzamos bemenet információja kerüljön-e be a FlipFlop-ba a *következő hatásos órajellel*.

Az *aszinkron párhuzamos adatbevitel* a FlipFlop clear és preset bemeneteinek segítségével valósítható meg *az órajeltől függetlenül*. Az aszinkron párhuzamos adatbevitel történhet egy lépésben (ahol kapuk biztosítják a clear és preset bemenetek megfelelő jelszintjeit) illetve két lépésben, mikoris először minden FlipFlop-ot a clear bemeneten keresztül nullázni kell és a második lépésben lehet csak a megfelelő tárolókat a preseten keresztül a H állapotba vinni.

Az információk **soros/párhuzamos és párhuzamos/soros átalakítása** a léptetőregiszterek egyik legfontosabb felhasználási területe. Pl. egy 8 bitből álló kódszót a léptetőregiszter sorosan 8 órajel-impulzussal vesz fel, ami aztán a 8 FlipFlop-kimeneten párhuzamos



formában áll rendelkezésre. Fordított irányban: a 8 bites kódszót a FlipFlopok párhuzamos bemenetein keresztül beírjuk és 8 órajel-impulzussal sorosan kiléptetjük



2. ábra: A léptetőregiszter szinkron párhuzamos bemenetének kapcsolása

### Jobbra és jobbra/balra léptetőregiszter

Alapesetben egy órajel-impulzussal minden egyes FlipFlop információja a következő, tőle jobbra eső FlipFlopba kerül át. A léptetés iránya tehát jobbra mutat. Néhány számítástechnikai alkalmazás viszont az ellentétes információáramlási irányt követeli meg. Két-két szomszédos FlipFlop-kimenet megfelelő kapuáramkörös összekapcsolásával elérhető, hogy egy vezérlő-jel segítségével be lehet állítani a léptetés irányát (ill. a párhuzamos beírást).

### Feladat, 8. mérés:

(345)

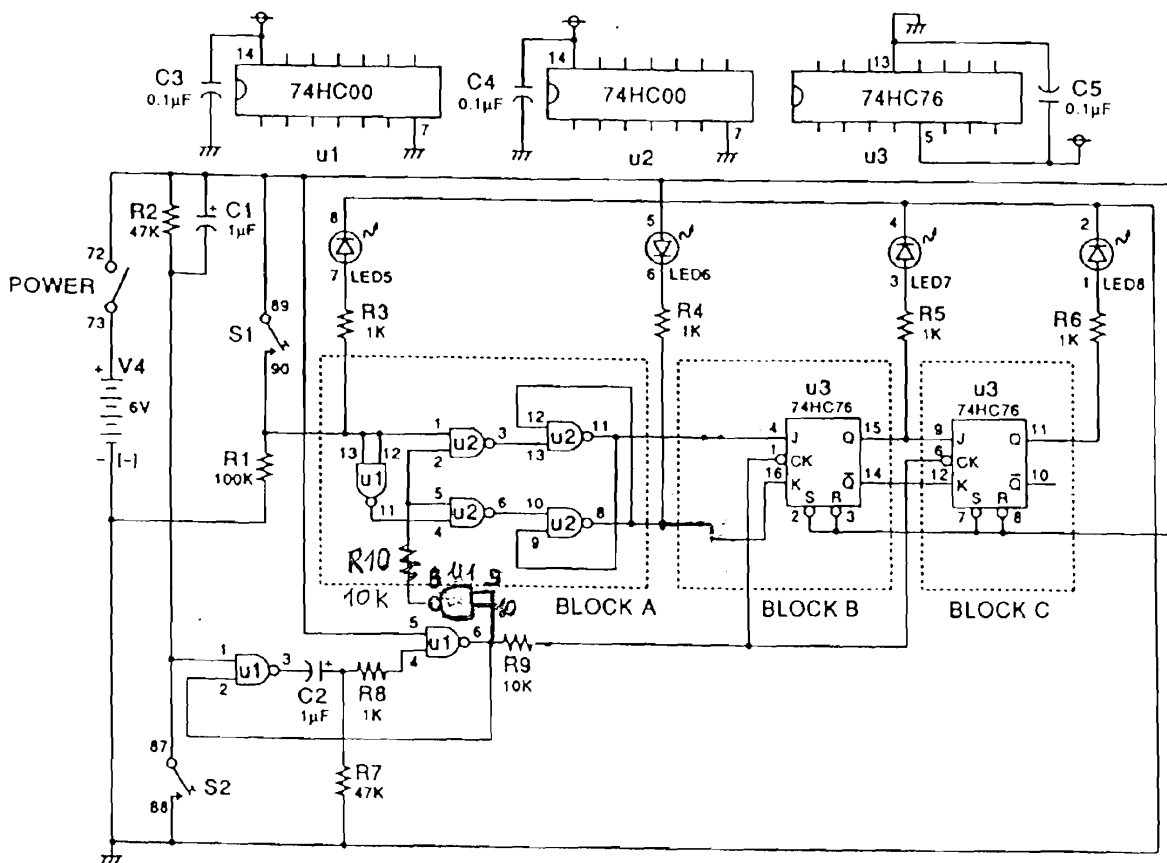
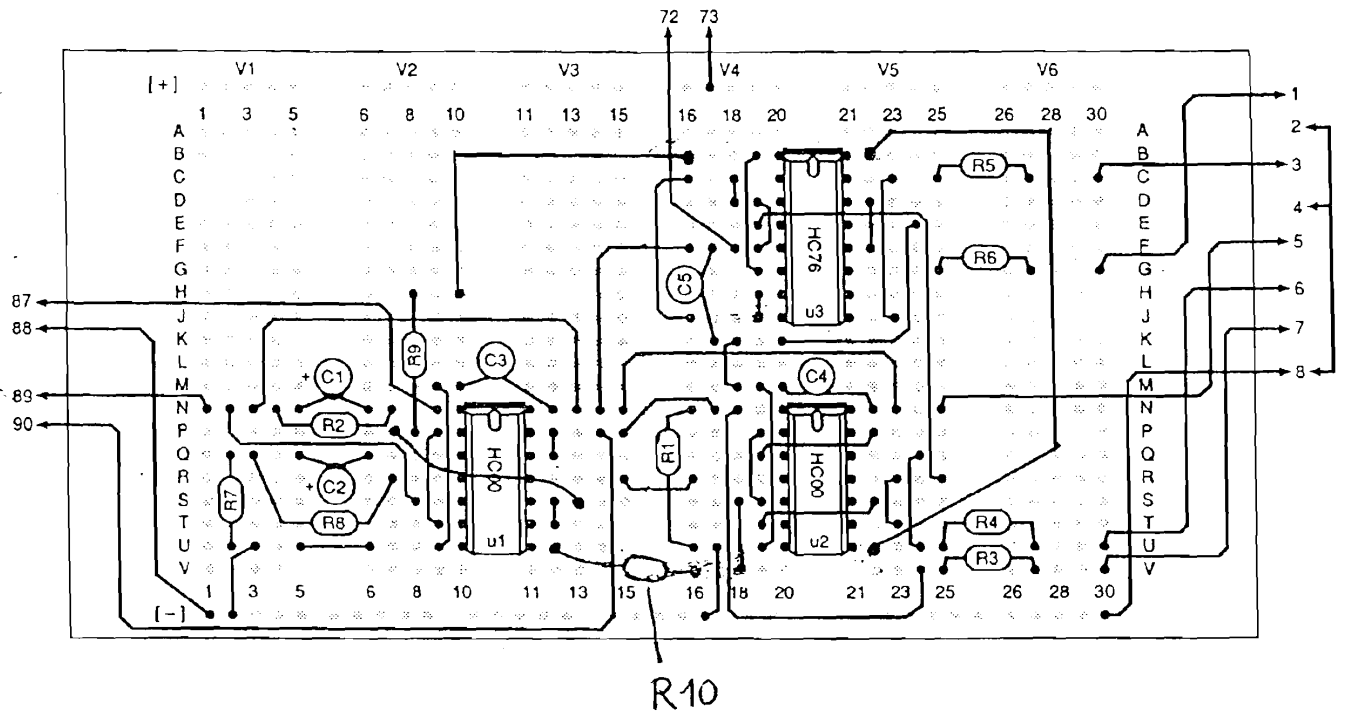
Az alábbi kapcsolás egy három bites léptetőregisztert valósít meg. A szükséges három FlipFlopból kettőt a 74HC76-os IC szolgáltat, egyet kapuáramkörökből állítunk össze.

- Vizsgálja meg az áramkör működését (bemenetek-kimenetek közötti kapcsolatot) és ábrázolja a jegyzőkönyvben alkalmas módon! Írja le a tapasztaltakat! (táblázat, idődiagram három különböző jel-mintára)
- Az órajelet egy monostabil multivibrátor (lásd 9.mérés) állítja elő, mely S2 megnyomására indul. Minden egyes gombnyomásra egy kb. 40 ms-os, L-szintű órajel-impulzus keletkezik.

Miért van szükség az első tároló esetében az órajel invertálására?

Próbálja ki, hogyan működik az áramkör pergésmentesítés nélkül?

U1	74HC00	R1	100 k $\Omega$	R5	1 k $\Omega$	R9	10 k $\Omega$	C3	0,1 $\mu$ F
U2	74HC00	R2	47 k $\Omega$	R6	1 k $\Omega$	R10	10 k $\Omega$	C4	0,1 $\mu$ F
U3	74HC76	R3	1 k $\Omega$	R7	47 k $\Omega$	C1	1 $\mu$ F	C5	0,1 $\mu$ F
		R4	1 k $\Omega$	R8	1 k $\Omega$	C2	1 $\mu$ F		



Forrás: [1]

## 9. mérés: Multivibrátorok

A multivibrátorok (más néven billenő áramkörök) kétféle állapotot vehetnek fel, azaz 1 bit információ tárolására alkalmasak. Az állapotnak a megváltozása gyors lefolyású folyamat, amit átbillenésnek nevezünk. Ehhez az áramkör belül pozitív visszacsatolással rendelkezik. Háromféle változatuk létezik:

- bistabil multivibrátorok (FlipFlop-ok)
- astabil multivibrátorok (impulzusgenerátorok)
- monostabil multivibrátorok (monoflop-ok)

A **bistabil multivibrátor** két stabil állapottal rendelkezik (0,1). Átbillenteni egyik állapotból a másikba egy vezérlő bemeneti jellel lehet. Egy bit információ tárolására alkalmasak. Újabb vezérlő jelig a régi információt tárolják. Más néven FlipFlopnak is nevezik.

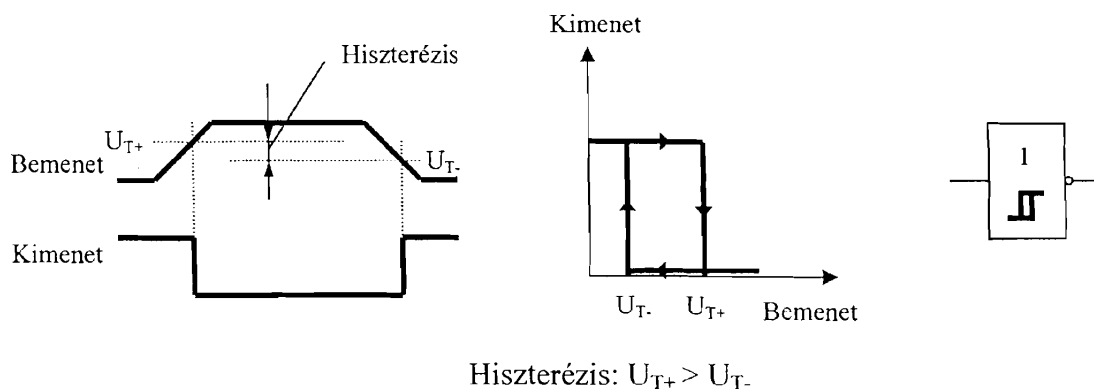
Az **astabil multivibrátornak** nincs stabil állapota. A tápfeszültség rákapcsolásakor magától, azaz vezérlő jel nélkül változtatja az állapotát. Mivel nincs stabil állapota, csak egy meghatározott ideig marad meg egyik állapotában utána átbillen a másikba. A két állapot egymást váltja. Gyakorlatilag egy meghatározott frekvenciájú négyszögjel-impulzussorozatot állít elő.

A **monostabil multivibrátornak** egy stabil állapota van. Alapesetben ebben az állapotban tartózkodik. Ebből kibillenteni egy vezérlő jellel lehet. A másik, instabil állapot viszont csak egy (a méretezés által) meghatározott ideig marad fenn. Ezen idő letelte után a monostabil multivibrátor visszabillen stabil állapotába, ezért időzítő kapcsolásnak is használják.

A mono- és astabil multivibrátorok megvalósításának egyik legegyszerűbb módja, amikor a billenésre Schmitt-triggeres bemenetű invertert használunk és monostabilnál a vezérlő jelet differenciáljuk, ill. astabilnál a visszacsatolt jelet késleltetjük.

A billenő-kapcsolások tárgyalásánál kell megemlíteni tehát a **Schmitt trigger** is, ami lényegében szintén két állapotba billenthető be, mégpedig a bemenetre adott feszültség nagyságától függően. Nem invertáló Schmitt trigger esetében az L→H átbillenéshez tartozó bemeneti feszültség szintet felső (vagy pozitív irányú) küszöbfeszültségnek, a H→L átbillenéshez tartozó bemeneti feszültség szintet pedig alsó (negatív irányú) küszöbfeszültségnek nevezzük.

A Schmitt trigger egy olyan komparátor (feszültség-összehasonlító áramkör), melynek a különböző változási irányokhoz tartozó billenési küszöbfeszültségei (trigger szintek) nem esnek egybe, hanem egy ún. hiszterézisfeszültséggel különböznek ( $U_H = U_{T+} - U_{T-}$ ).



1. ábra: Schmitt triggeres bemenetű inverter működése, átmeneti karakterisztikája és rajzi jelölése

A Schmitt triggeres inverter (1. ábra) a bemeneti feszültség nagyságától függően billen be a két állapot valamelyikébe. Az invertálás miatt, ha a bemeneti feszültség nagy, a kimenet L szintre, ha alacsony, H szintre fog beállni. Az átbillenés akkor következik be, ha a bemeneten a feszültség eléri és meghaladja (pozitív vagy negatív irányban) a billenési küszöbfeszültség értékét. Ha a bemeneten a feszültség növekszik és az inverter L szintre billen, akkor a pozitív irányú küszöbfeszültség ( $U_{T+}$ ) érvényes, ha a feszültség csökken és a kimenet H-ba billen, akkor a negatív ( $U_{T-}$ ).

A bemeneti jel lehet analóg jel is, a kimenet viszont mindig digitális (tehát a bemeneten bármilyen lassú jelváltozási sebesség megengedett, ellentétben a kimenettel).

*Az invertáláson kívül más logikai műveletet végző kapuáramkört is el lehet látni Schmitt triggeres bemenettel, melynek lényege a kapcsolási hiszterézis. A hiszterézisgörbe alakját használják a Schmitt triggeres bemenet rajzi jelölésére is.*

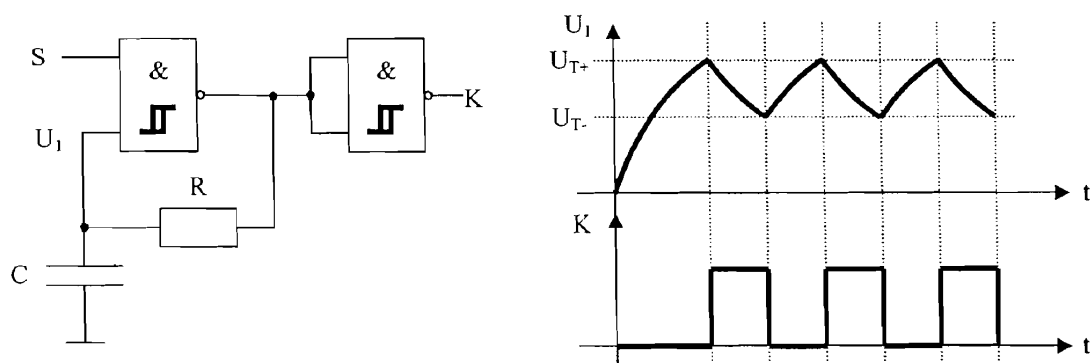
*A kapcsolási hiszterézisre (azaz arra a tényre, hogy a „visszabillenés” alacsonyabb feszültségen következik be mint az „odabillenés”) stabilitási okokból van szükség. Ha a hiszterézis túl kicsi, az oda- és visszabillenés nemkívánatos módon már a hasznos jelen üllő zavarjelek hatására is bekövetkezik. A túl nagy hiszterézis viszont az impulzus hosszát befolyásolja.*

Az astabil multivibrátort a frekvencia jellemzi, a monostabil multivibrátort pedig az instabil (tartási) idő. A billenő kapcsolások lényeges időzítőelemként ellenállásokat és kondenzátorokat tartalmaznak és nem tisztán digitális áramkörök.

Az astabil billenőáramkörök működése többnyire olyan RC-tagok feltöltődési és kisülési folyamatain alapul, melyeket NAND kapuáramkörök visszacsatoló ágába késleltetőelemként kötünk be. A periódusidőt az RC-tag időállandója szabja meg.

Ha egy Schmitt triggeres inverter kimenetét egy késleltető tagon keresztül visszakötjük a bemenetre a legegyszerűbb astabil multivibrátort kapjuk. A negálás miatt az inverter kimenete a bemenetre visszavezetve mindig ellentétes állapotba való billenést vált ki, kicsit késleltetve.

A 2. ábrán két, Schmitt triggeres bemeneti fokozattal rendelkező NAND kapuból felépített astabil multivibrátort láthatunk. C (és R) megválasztásával a frekvenciát 1 Hz és 10 MHz között állíthatjuk be. A kapcsolás egy db Schmitt triggeres inverterrel is működik. A második inverter csak a kimeneti jelformát javítja. Az S bemeneten keresztül az impulzusgenerátort meg lehet állítani.

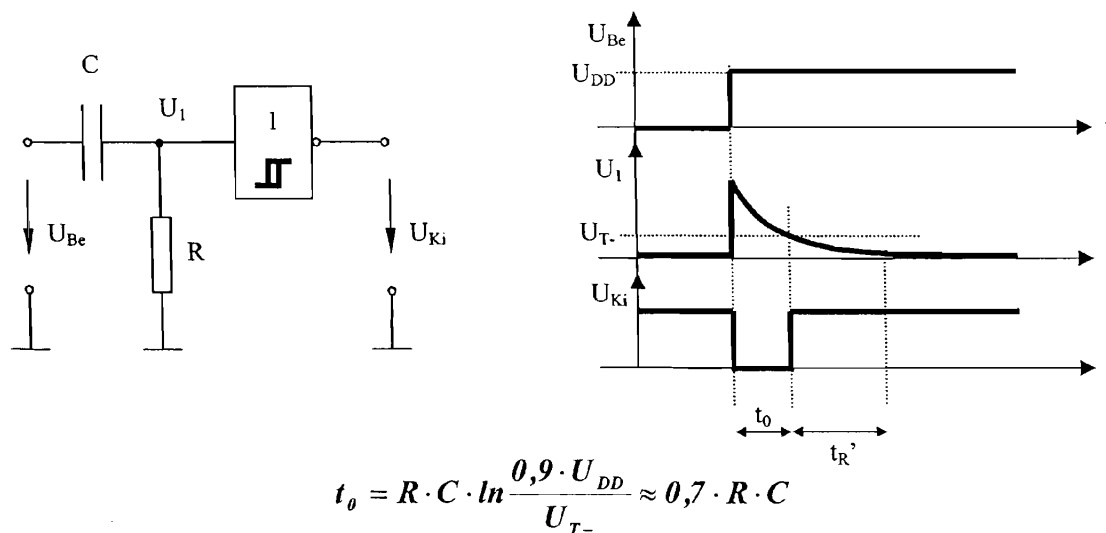


2.ábra: Astabil multivibrátor felépítése, működése

A **monostabil multivibrátor**  $t_0$  tartási idejét (instabil állapot) is a (belső vagy külső) RC-tag szabja meg. Az instabil állapotba való átbillenést a bejövő vezérlő jel felfutó vagy lefutó éle váltja ki. Visszabillenés után az újabb átbillenés csak egy ún. regenerálódási idő ( $t_R$ ) eltelte után következhet be - azaz miután a kondenzátor a nyugalmi feszültségére feltöltődött ill. kisült - mert ellenkező esetben a  $t_0$  tartási idő megrövidül.

Alapesetben a monoflop *nem triggerelhető újra* a tartási idő alatt, azaz az ekkor beérkező újabb vezérlő él hatástalan, nem indítja újra a tartásidőt. Az *újratriggerelhető* monoflopok tartási idejét a tartási időn belül beérkező újabb vezérlő impulzusok újraindítják.

Ha a Schmitt triggeres inverter bemenetére a vezérlő jelet egy differenciáló tagon keresztül vezetjük, akkor a bemeneti jel felfutó élével indított monostabil multivibrátort kapunk, melynek kapcsolása és működési impulzusdiagramja a 3. ábrán látható.



3. ábra: Egyszerű monostabil multivibrátor felépítése, működése

A fenti kapcsolás hátránya, hogy a bemenetnek  $t_0$  teljes ideje alatt H szinten kell maradni, egyébként a  $t_0$  megrövidül. (Ez kiküszöbölhető, ha a bemenet elé bekötünk egy NAND kaput, melynek másik bemenetére a multivibrátor kimenetét vezetjük vissza.)

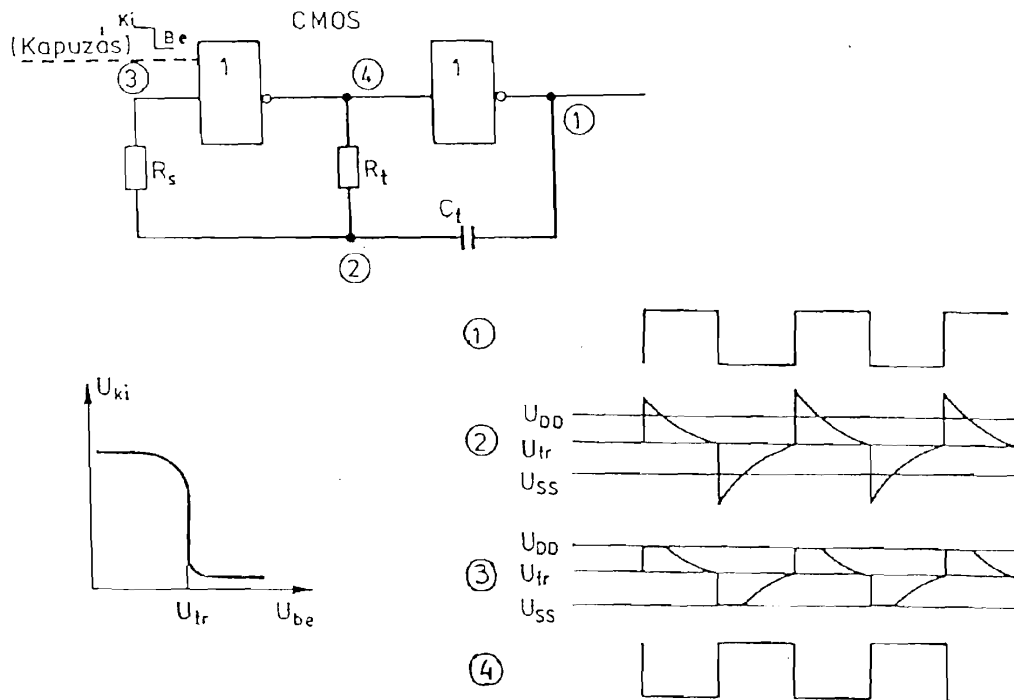
A bemeneti jel L szintre kapcsolásával az  $U_1$  ponton a feszültség  $-U_{DD}$ -vel hirtelen lejjebb ugrik, ami adott esetben a kapu bemenetén negatív feszültségcsúcsot eredményezhet. Az előírt megengedett legnagyobb negatív bemeneti feszültség túllépése ellen a test felé bekötött diódával védekezhetünk. A monoflopot újraindítani a regenerálódási idő eltelte után szabad, azaz csak akkor, ha az  $U_1$  ponton a feszültség gyakorlatilag nulla (a kondenzátor kisült). A regenerálódási időbe beleszámít a bemeneti lefutó él hatására létrejövő,  $U_1$  pontbeli negatív feszültség lecsengése is.

Fenti hátrányok kiküszöbölhetők, ha a kondenzátor elé egy NAND kaput kapcsolunk és ennek másik bemenetére visszavezetjük a kimenetet. Ez pozitív visszacsatolást jelent, és az  $U_{T-}$  küszöbfeszültség elérésekor gyorsítja az átbillenési folyamatot, a kondenzátor kisülését.

**Feladat, 9. mérés: CMOS inverterekből álló astabil multivibrátor vizsgálata (53)**

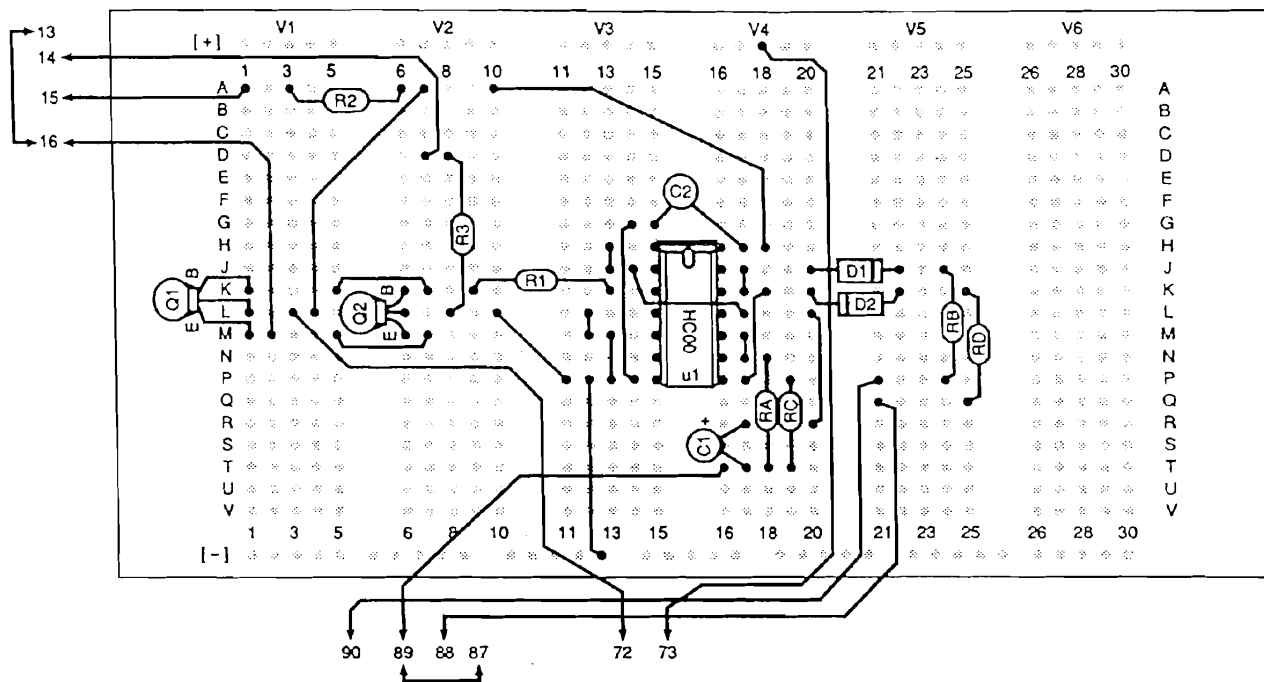
Az áramkör a 4. ábrából kiolvasható elven működik. Ha a 3. pontban lévő feszültség növekedésekor vagy csökkenésekor eléri a triggerszintet, az első kapu kimenete átbillen, ami az 1-es pontra is továbbterjed. A 4-es és 1-es pont közötti potenciálkülönbség a kondenzátor feltöltődési ill. kisülési folyamatát indítja be,  $T=R_t \cdot C_t$  időállandóval.

$R_s$  (a mérőkapcsolásnál  $R_A$ ) csak arra szolgál, hogy a CMOS-bemenet túlfeszültség elleni bemeneti diódás védelme ne korlátozza le a 2-es ponton létrejövő feszültséget, ezzel az 1-es (és 4-es) ponton előálló négyszögjel-frekvenciát függetlenné tegye a tápfeszültségtől.

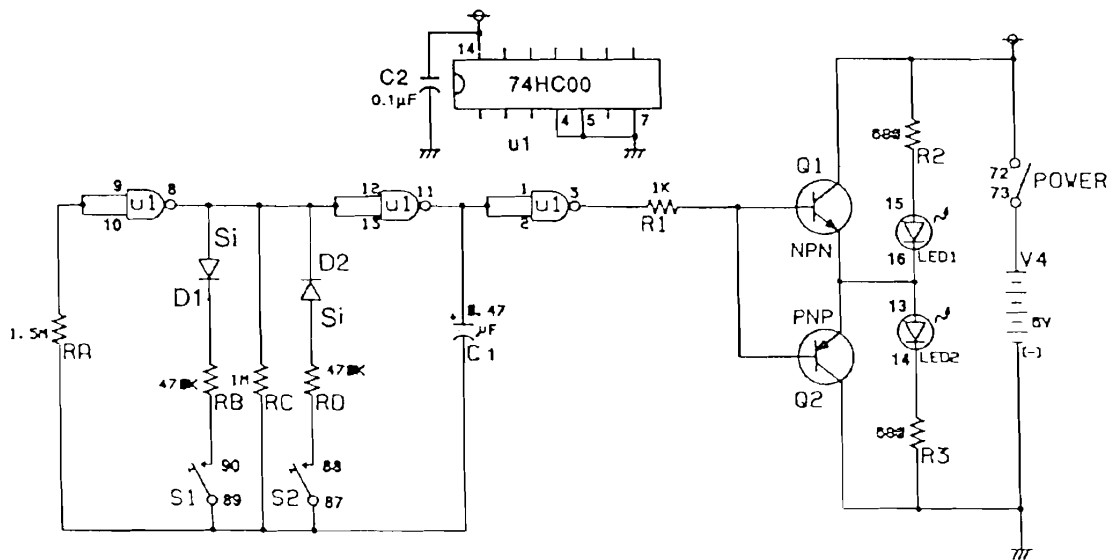


4. ábra: CMOS astabil multivibrátor kapcsolási rajz, idődiagram és az inverter transzfer karakterisztikája

- A mérőkapcsolás egy változtatható kitöltési tényezőjű astabil multivibrátort valósít meg NAND kapuáramkörök és RC elem felhasználásával. Az astabil multivibrátor frekvenciáját  $C_1$  és  $R_C$  határozzák meg ( $T \approx 2 \cdot R_C C_1$ ).  $S_1$  vagy  $S_2$  lenyomásakor a kitöltési tényező megváltozik. A frekvenciát és a kitöltési tényezőt a LED-ek *világít - nem világít* idejéből becsülje meg!
- A tápfeszültség kikapcsolása után  $R_C$  ellenállást és  $C_1$  kondenzátort cserélje ki a mérésvezető által megadottra! A multivibrátor kimenetére (K12-es pont) kapcsolt oszcilloszkóppal mérje meg  $S_1$  és  $S_2$  mind a négy lehetséges variációja esetében a kimeneten a H és L szintek idejét! A mért periódusidőt hasonlítsa össze az elméleti képlet szerint számítottal!
- Az oszcilloszkóp második csatornájára kapcsolja rá a  $C_1$ -en lévő feszültséget (T20-as pont) és rajzolja le léptékhelyesen az oszcilloszkópon látható két idődiagramot mind a négy esetben! Az oszcilloszkóp mindkét csatornája DC-állásban legyen!



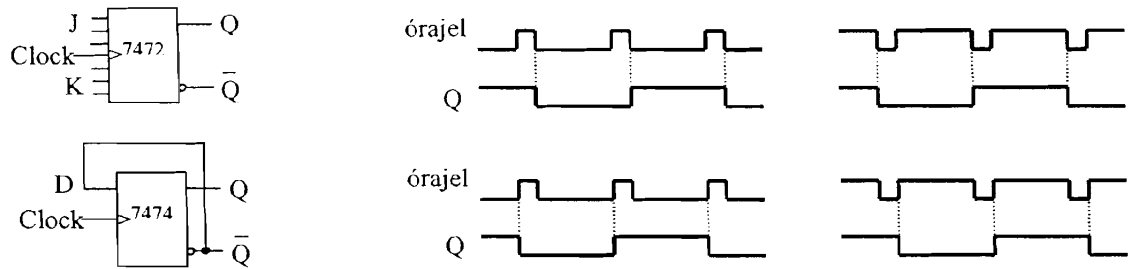
U1	74HC00	RA	1.5MΩ	RD	470KΩ	R3	680Ω	D1	Si
Q1	NPN	RB	470KΩ	R1	1KΩ	C1	0.47μF	D2	Si
Q2	PNP	RC	1MΩ	R2	680Ω	C2	0.1μF		



Forrás: [1]

## 10. mérés: Frekvenciaosztók

A frekvenciaosztók egy adott frekvenciájú impulzussorozatból (négyszögjelből) állítanak elő olyan impulzussorozatot, melynek frekvenciája a bemeneti frekvencia törtrésze. A legegyszerűbb frekvenciaosztó (és egyben számláló) a JK-FlipFlop, melynek mindkét vezérlő bemenete H szinten van (vagy nincs bekötve). Az órajel bemenet frekvenciáját 2:1 arányban leosztja, azaz a kimeneten egy impulzussorozat jelenik meg, melynek frekvenciája az órajel-frekvencia fele. Ugyanezt a funkciót lehet egy D FlipFloppal is megvalósítani, ha a  $\bar{Q}$  kimenetet visszavezetjük a D bemenetre. (A T tároló is 2:1 arányú frekvenciaosztóként működik, amennyiben a T bemenet H szinten van.)



1. ábra: Nyitott JK bemenetű JK tároló (7472) és visszacsatolt D tárló (7474); a legegyszerűbb 2:1 frekvenciaosztók

A 2:1 arányú frekvenciaosztó egyben a bináris kódú számláló alapeleme is. A kettes számrendszerben való előre ill. visszaszámlálásnál ugyanis két szomszédos helyiérték „frekvenciája” egymáshoz 2:1 arányban viszonyul.

A frekvenciaosztó általánosan megfogalmazva  $m$  db óraütem-impulzus után ad le egy db kimeneti impulzust. Frekvenciáról akkor beszélhetünk, ha az impulzusok periodikusan, azaz ugyanazon időközönként érkeznek be, ekkor a kimeneti impulzussorozat frekvenciája a bemeneti impulzusok frekvenciájának az  $m$ -ed része. A frekvenciaosztó  $m:1$  arányban osztja le a frekvenciát, de az  $1:m$  elnevezés is használatos.

Léteznek olyan frekvenciaosztók is, melyek egy  $m$  db bejövő impulzusból álló ciklus során nem csak egy, hanem több kimeneti impulzust adnak le és ezek száma vagy előre meghatározott, de programozható is lehet. A 7497-es programozható frekvenciaosztó például 64 állapotot vehet fel ( $m=64$ ) és mindegyik ilyen 64-es sorozatra választhatóan 1...63 kimeneti impulzust tud leadni. A „frekvencialeosztás aránya” 64:1-től 64:63-ig terjedhet, ami (mivel az impulzusok távolsága cikluson belül nem egyenletes) csak egy impulzus-arányt jelent és nem kimondottan frekvenciát.

### A frekvenciaosztók és a számlálók közötti különbség

A számláló feladata a bemenetére érkező impulzusok darabszámának (melyek érkehetnek szabálytalan időközökben vagy periodikusan) számlálása és a beérkezett impulzusdarabszám megfelelő kód segítségével történő tárolása és kijelzése. A számlálók lényegében egymás után kapcsolt FlipFlopokból (tárolókból) állnak, melyek kimenetei együttesen adják a számláló állapotát. A számláló állapota a kódolás szerint a beérkezett impulzusok darabszámának felel meg. Minden egyes újabb bejövő impulzussal a számláló egy következő állapotba jut, amely az utána következő impulzusig megőrződik. (Szigorúan véve azonban a számláló és a frekvenciaosztó nem az egész impulzusra, hanem annak csak egyik élére reagál.)



A számlálók a frekvenciaosztókhoz hasonlóan ciklusosan dolgoznak és egy cikluson belül szintén  $m$  db állapotot vehetnek fel. Szintén egy bemenettel rendelkeznek. A fő különbség abban rejlik, hogy a számlálók minden egyes állapotra egy **kimeneti kódot** adnak, mégpedig annyi helyiértéken, ahány Flip-Flop a kapcsolásban szerepel. A frekvenciaosztóknál nincs mindegyik tároló kimenete kivezetve és a kódoknak sincs szerepe, mivel a lényeg csak a cikluson belül a kimeneten leadott impulzusok darabszámában rejlik.

A számlálók esetében minden egyes bejövő impulzus után felismerhetőnek kell lenni (a számláló állapotáról), hogy az adott pillanatig hány db órajel-impulzus érkezett be. Az összerendelést a beérkezett impulzus-darabszám és a számláló-állapot között az alkalmazott kódrendszer adja meg.

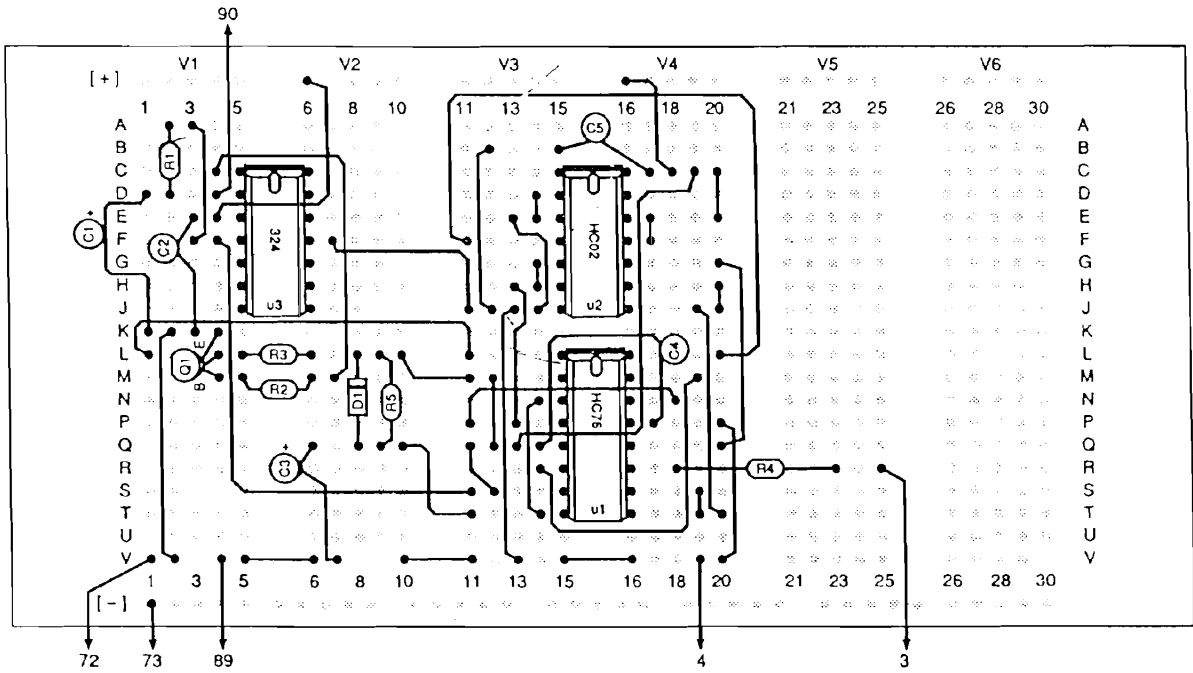
Fentiekből következően az összes számlálót lehet frekvenciaosztóként használni, viszont fordítva ez nem érvényes.

**Feladat, 10. mérés:** 4:1 arányú frekvenciaosztó (számláló)

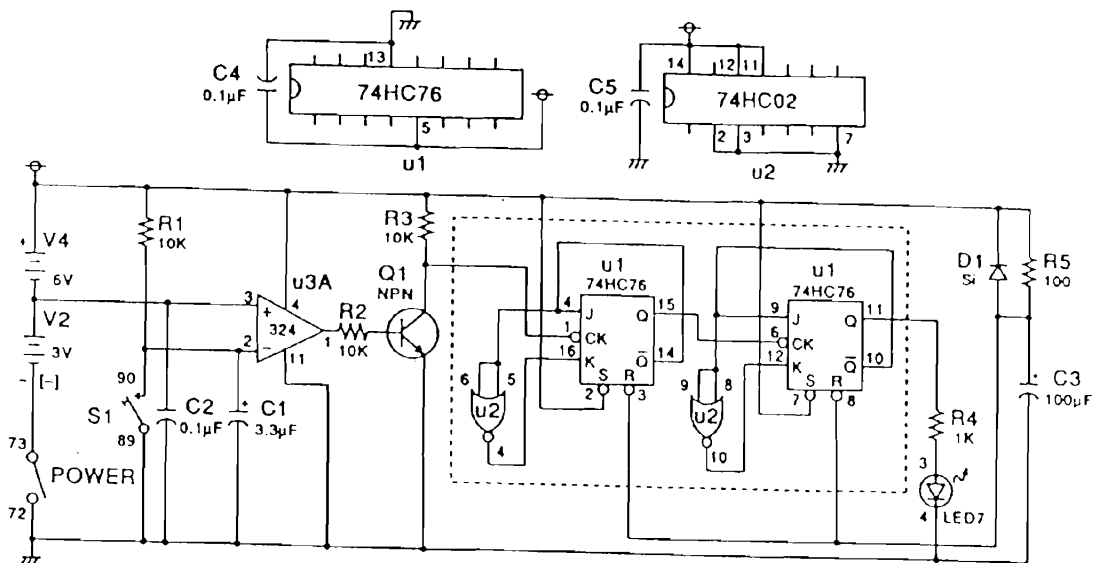
(180)

Állítsa össze az alábbi kapcsolást! A kapcsolási rajzon a bekeretezett rész egy 4:1 arányú frekvenciaosztót valósít meg két db visszacsatolt JK FlipFlop (azaz D FlipFlop) segítségével. A bemeneti impulzusokat az S1 kapcsolóval adjuk rá (a pergesmentesítést RC tag és komparátor végzi), a kimenetet a LED jelzi.

- Figyelje meg a bemeneti és kimeneti impulzusok közötti összefüggést és idődiagramban ábrázolja! U2 4-es lába (F12) és a test közé egy 1 k $\Omega$ -os ellenálláson keresztül egy LED-et kapcsolva figyelje meg a két FlipFlop közötti jelet is. Vegye fel az így előállt számláló igazságtáblázatát! Figyelje meg, mi történik, ha a kapcsoló jelét közvetlenül, pergesmentesítés nélkül adjuk rá az osztó bemenetére!
- Szakítsa meg az osztó órajel-bemenetét a vezetékét az L1-es pontból kihúzva és adjon erre a vezetékre (U1 1-es láb) TTL szintű 1 kHz-es négyszögjelet jelgenerátorral! Oszilloszkóppal ellenőrizze a működést, rögzítse idődiagramban! (bemenet, köztes jel, kimenet)



Q1	NPN	R3	10KΩ	C1	3.3μF	C4	0.1μF	D1	Si
R1	10KΩ	R4	1KΩ	C2	0.1μF	C5	0.1μF		
R2	10KΩ	R5	100Ω	C3	100μF				



Forrás: [1]

## 11. mérés: Számlálók I.

### Számlálási kapacitás, $m$ [0 . . (m-1)]

Mivel minden FlipFlop két értéket (0, 1) vehet fel, az  $n$  db FlipFlop-ból álló számlálónál az összes előfordulható állapotok száma maximum  $2^n$ . Ez a számláló - amennyiben minden lehetséges állapotát kihasználja - 0-tól ( $2^n-1$ )-ig tud számolni. (A  $2^n$ -dik impulzus után újra a kiindulási állapot tér vissza.)

A számlálók, mint említettük mindig ciklikusan dolgoznak, azaz egy bizonyos számú beérkezett impulzus után ( $m \leq 2^n$ ) minden egyes számlálóállapot ismétlődik. Az  $m$  (modulo) a számláló kapacitása. Egy számlálónak nem szükséges minden lehetséges állapotot kihasználni. Ez az adott kódrendszerrel függ.

Ha több számlálót (számlálófokozatot) egymás után sorba kapcsolunk, akkor az elől lévő számláló  $m$ -dik impulzusa (amellett, hogy az adott számlálófokozatot az alaphelyzetbe viszi vissza) ezzel egyidejűleg egy átvitelt (azaz hatásos órajel-élt) is szolgáltat a sorban következő számlálófokozat számára.

A beérkezett impulzusok száma és a kimeneti kódszó közötti összefüggést, azaz a számláló működésének lényegét meg lehet adni az impulzusdiagrammal, vagy az ún. igazságtáblázattal, melyekkel a tulajdonképpeni kapcsolás ekvivalens.

Minden számláló legfontosabb jellemzője a számlálási kapacitás ( $m$ ) és az a kód, amivel a számláló dolgozik. A számláló kódja megadja, hogy hányadik beérkezett impulzus milyen kimentti kódnak felel meg (egyértelműen).

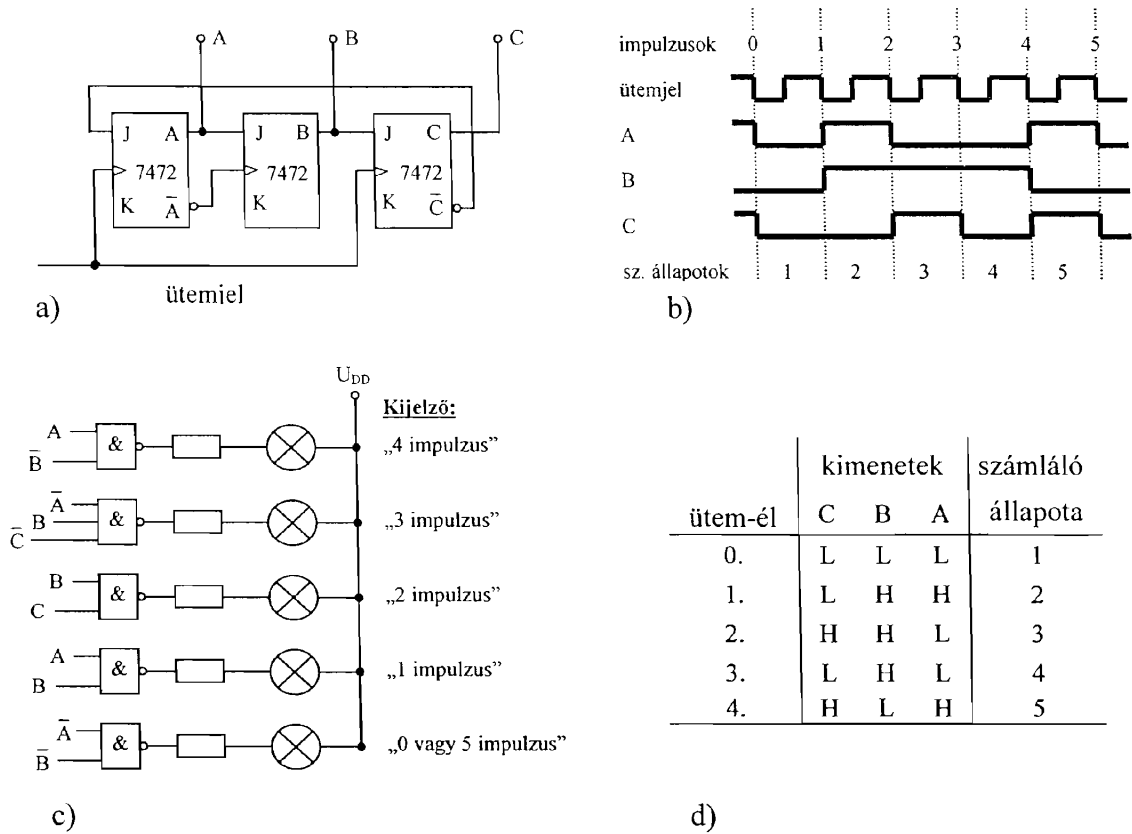
**A számlálók, és ezzel együtt kódjuk megadási módjaira** láthatunk egy általános példát az 1. ábrán. Itt a számlálási kapacitás  $m=5$ , és ez mivel nem 2-nek egész kitevőjű hatványa, ezért ebben az esetben a szükséges  $n=3$  db FlipFlop-ból álló számláló összes lehetséges  $2^3=8$  állapotából 3 állapot kihasználatlan, ezeket a számláló nem veszi fel, azaz átugorja. Ebből a szempontból ez a kód a számlálási kapacitást illetően nem a leggazdaságosabb.

Az 1. ábrán látható számláló 0-tól 4-ig tud számolni.

A 0-dik és első hatásos órajelimpulzus-él közötti időben a számláló az első, vagy másnéven kiindulási állapotban van. Az első órajel-élnél a második állapotba kerül, a másodikkal a harmadikba és így tovább. Az  $(m-1)$ -dik impulzus hatásos éle viszi a számlálót az  $m$ -dik állapotba. Az  $m$ -dik impulzus révén jut a számláló vissza az első, másnéven kiindulási állapotba.

A példánkban az 5-dik beérkező impulzus a számlálót az alaphelyzetbe (000) billenti vissza.

A kód egyedi, nem ekvivalens a kódolók-dekódolók tárgykörében említett kódok egyikével sem, ezért külön dekódoló hálózatot kell a számlálóhoz építeni. A bemutatott aszinkron számláló egy adott séma alapján tervezhető nagyobb kapacitásúra is és a felépítése is viszonylag egyszerű.



### 1. ábra: Általános példa az $m=5$ számlálóra

a) kapcsolási rajz, b) impulzusdiagram, c) dekódoló kapcsolás a beérkezett impulzusdarabszám decimális kijelzésére, d) igazságtáblázat

Az irodalomban nagyon sok, többféle kapacitású és kódot használó számláló ismert, melyek a gyakorlatban felmerülő igényeket túlnyomórészt kielégítik. A speciális igényeket (pl. speciális kód, átváltható kapacitás) kielégítő számlálókat egyedileg kell megtervezni, amihez többféle tervezési módszer áll rendelkezésre. A tervezés lényege, hogy az egyes FlipFlopok be- és kimeneteit úgy kössük össze ill. csatoljuk vissza, hogy a kívánt kapacitás és kód álljon elő; az összekötő vezetékek és logikai kapuáramkörök darabszáma és ezáltal a rajtuk fellépő késleltetési idők (a felső határfrekvencia növelése érdekében) minimálisak legyenek.

Mivel a példában szereplő számláló nem használja ki az összes lehetséges állapotát, elvileg ezek a ki nem használt állapotok nem fordulhatnak elő. A gyakorlatban azonban zavarjelek hatására véletlenszerűen mégis előállhat egy meg nem engedett állapot. Erre az esetre kapcsolástechnikailag kell biztosítani, hogy ne történjen blokkolás, azaz egy bizonyos tiltott állapotból adott darabszámú impulzus után a számláló visszatérjen a normális ciklusába.

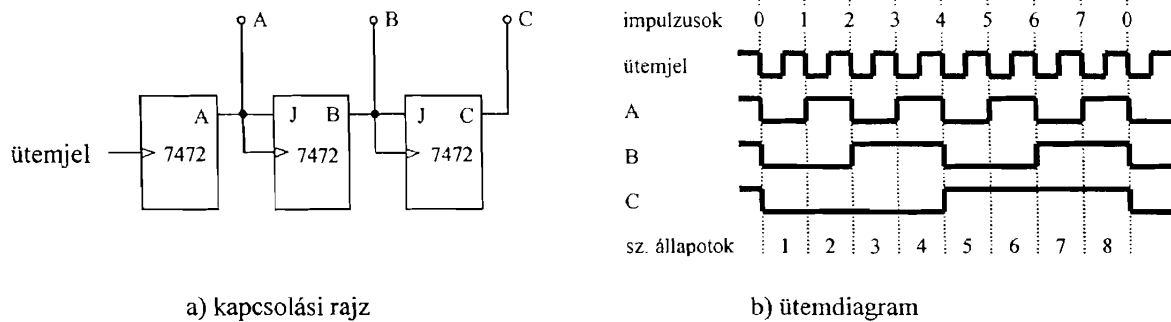
### Aszinkron számlálók

Aszinkron számlálóknál minden FlipFlopot az előtte lévő FlipFlop kimeneti jele vezérli. A bemeneti ütemjel csak a sorban első helyen lévő FlipFlopra kerül rá (esetleg még egyre, lásd 1. ábra).

Előnyük: aszinkron számlálóknak a szinkronszámlálókkal ellentétben az egyes számlálófokozatokat jelentő FlipFlopok között nincs szükségük összekötő kapukra és köztes összekötésekre, vagy csak nagyon kevésre.

Hátrányuk: alacsonyabb a felső határfrekvencia

A megengedett maximális frekvenciát az határozza meg, hogy a FlipFlopok nem egyidőben, hanem egymás után, a megelőző FlipFlop késleltetési idejével késleltetve billennek át. A számláló állapotának kiértékelése minden ütemimpulzus után csak az összes FlipFlop billenési folyamatának befejeződése után lehetséges. Minden FlipFlop a késleltetési idejével növeli a jel terjedési idejét, míg az az első FlipFloptól az utolsóig eljut.



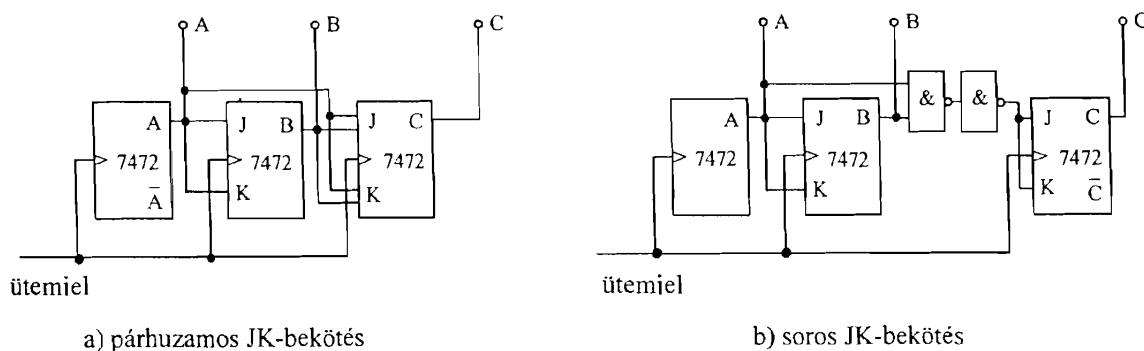
2. ábra: Aszinkron számláló ( $m=8$ , bináris kód)

### Szinkron számlálók

Szinkron számlálóknál mindegyik FlipFlop ütemjel-bemenete párhuzamosan van kapcsolva, az ütemjel egyszerre vezérli őket. Ezáltal az összes átbillenés egyidőben történik. Előnye az aszinkron számlálókkal szemben az, hogy mindegyik FlipFlop-kimeneten a jel csak egy FlipFlop késleltetési idejével késik a hatásos órajel-élhez képest, a számláló nagyságától függetlenül. Mivel nem minden tárolónak kell a hatásos órajel-élnél billennie, ezért a JK-bemenetek és kimenetek megfelelő összekötésével kell a blokkolást elvégezni. Erre két példa látható a 3. ábrán

Az ütemjel felső határfrekvenciája függ a FlipFlop kapcsolási (késleltetési) idejétől, a Q kimenet és JK bemenet között kapcsolt kapuáramkörök legnagyobb előforduló késleltetési idejétől és természetesen attól az időtől, amire a dekódoló áramkörnek az információ átvételéhez szüksége van.

A szinkron számláló leglényegesebb hátránya abból ered, hogy a blokkoláshoz kapuk és összekötések szükségesek, valamint, hogy ezek többletterhelést jelentenek a FlipFlop-kimeneteken.



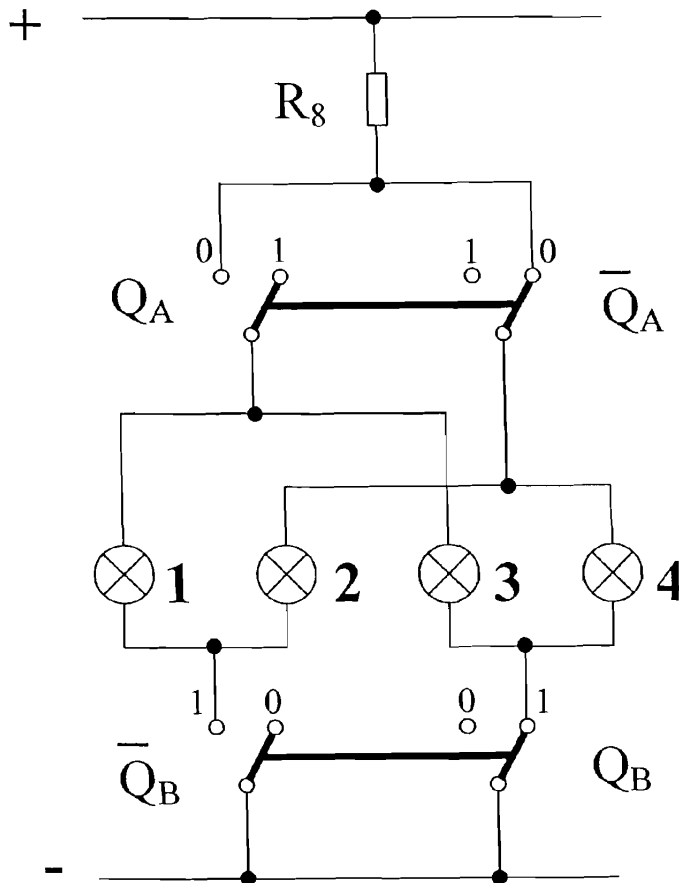
3. ábra Szinkron számláló ( $m=8$ , bináris kód) megvalósítása JK FlipFloppal

**Feladat, 11. mérés:** változtatható modulusú szinkron számláló

(188)

Az alábbi kapcsolás 2 db JK tárolóból álló szinkron számláló, mely egy 555-ös időzítő IC-ből kialakított impulzusgenerátortól kapja az órajelet. Bekapcsolás után az órajel-impulzusok beérkezésének megfelelően a számláló állapotai, azaz két kimeneti helyiértéke ( $Q_B$ ,  $Q_A$ ) ciklikusan változnak. Attól függően, hogy az S2, S3 vagy S4 kapcsolók közül melyiket kapcsoljuk be, változtatható az egy cikluson belül előforduló állapotok száma.

Az egyes állapotokat négy LED-en tudjuk ellenőrizni, melyeket egy "dekódoló" vezérel kapcsoló-tranzisztoros megoldással, az alábbi séma szerint:

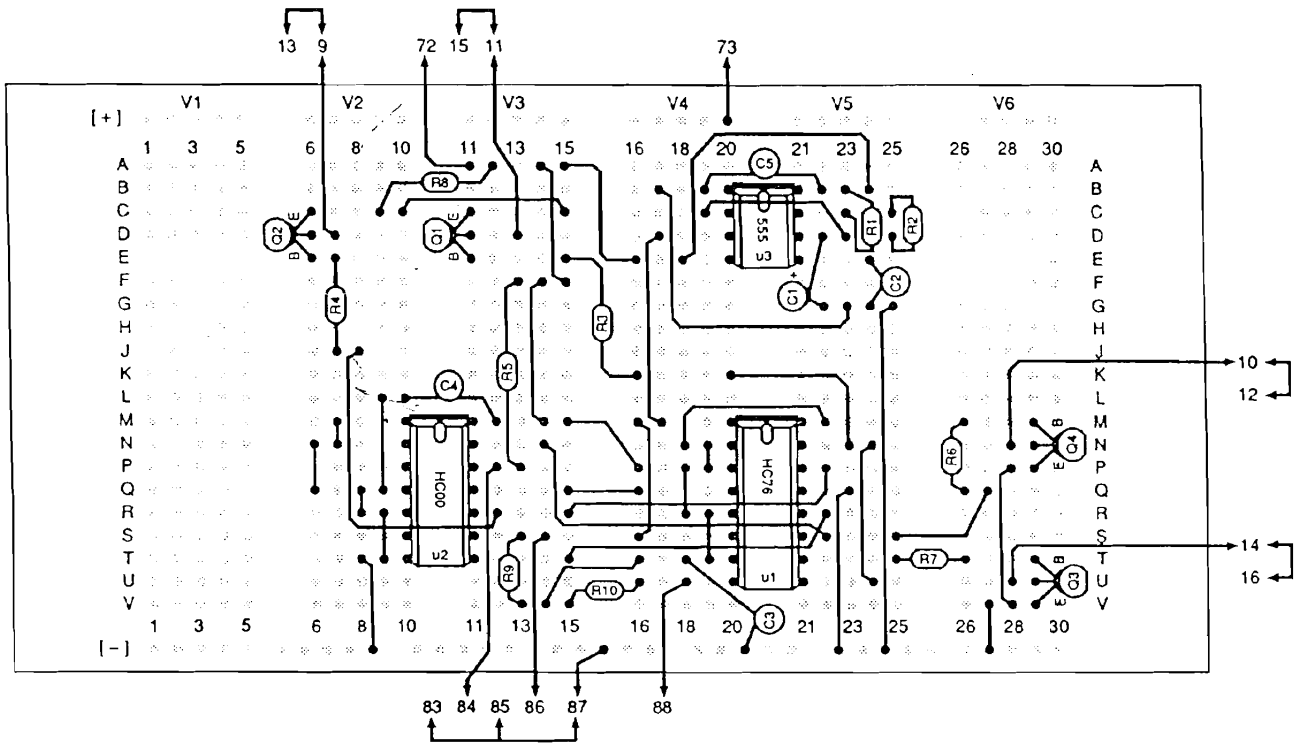


A és B flipflop  $Q$ , ill.  $\bar{Q}$  kimenetei vezérlik a négy, kapcsolóüzemben működő tranzisztor bázisait úgy, hogy a négy lehetséges kódszónak megfelelően mindig csak egy LED világít.

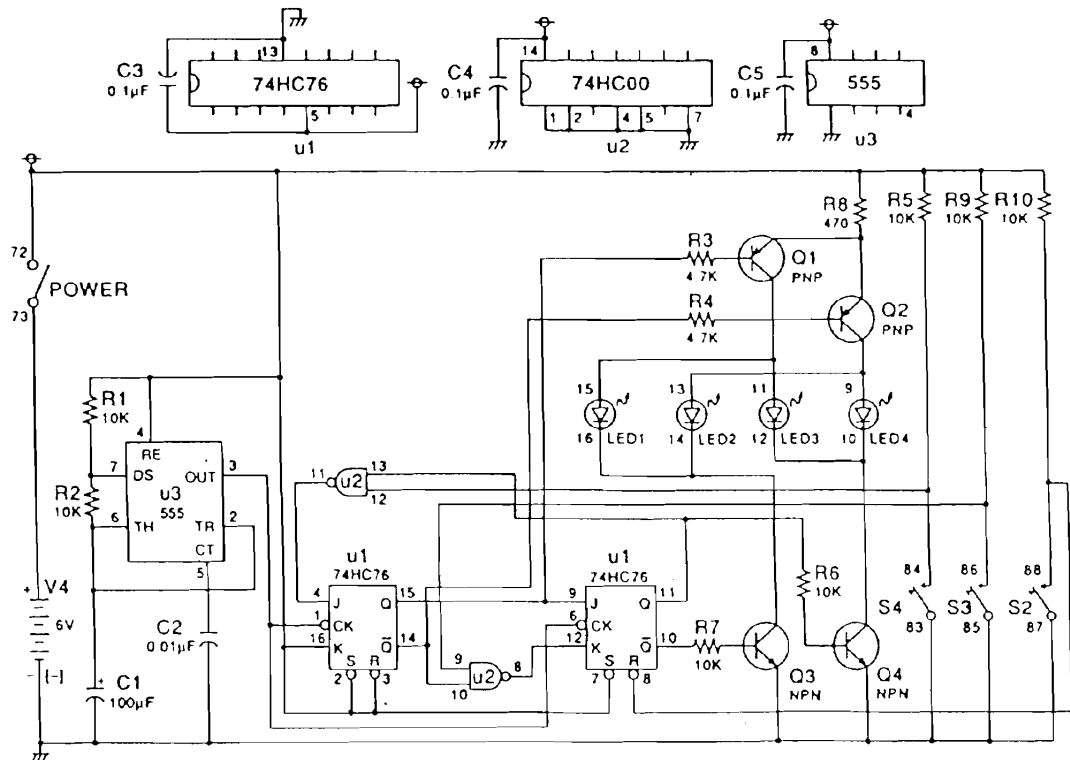
$R_8$  úgy van megválasztva, hogy a három sorban lévő LED ne világítson, csak a velük párhuzamos egy.

$Q_B$	$Q_A$	ez a LED világít
0	0	1
0	1	2
1	0	3
1	1	4

- A három lehetséges kapcsolóállás (S2, S3, S4) mindegyikéhez határozza meg az aktuális számlálási kapacitást és a számlálás kódját, ábrázolja igazságtáblázattal és idődiagrammal!  
(A számláló kódját  $U_1$  14-es és 10-es lábain mérje oszcilloszkóppal!  $C_1$ -et cserélje ki kisebb értékűre!)
- Hogyan működik a dekódoló? Mivel magyarázható a számlálási kapacitás tapasztaltak szerinti változása?



U1	74HC76	Q1	PNP	R1	10KΩ	R6	10KΩ	C1	100μF
U2	74HC00	Q2	PNP	R2	10KΩ	R7	10KΩ	C2	0.01μF
U3	555	Q3	NPN	R3	4.7KΩ	R8	470Ω	C3	0.1μF
		Q4	NPN	R4	4.7KΩ	R9	10KΩ	C4	0.1μF
				R5	10KΩ	R10	10KΩ	C5	0.1μF



Forrás: [1]

## 12. mérés: Számlálók II.

### Dekád számláló hétszegmenses kijelzéssel

A mérés során felépítendő számláló alapja egy 74HC191-es IC, mely egy 4 bites, szinkron, bináris kódú előre/hátra számláló, mely rendelkezik egy kezdeti-érték beállítási lehetőséggel, egy kimenet szolgál az átvitel kijelzésére és egy engedélyező bemenettel lehet a számlálást engedélyezni ill. tiltani. Más hasonló felépítésű számlálókkal együtt több számlálóáramkör egymás után kapcsolására, és ezzel nagyobb kapacitású számlálók felépítésére használható anélkül, hogy az összekapcsoláshoz egyéb kapcsolástechnikai megoldásokra lenne szükség.

#### Bemenetek:

- A,B,C,D - párhuzamos, aszinkron kezdőérték-beírás négy helyiértéken  
 LOAD - "L" szintje az aszinkron beírást aktiválja, beíráskor az A-D bemeneteken meglévő jelek rákerülnek a QA -QD kimenetekre, CK és G aktuális értékétől függetlenül. A LOAD-bemenetre adott (periodikus) jellel (amit például az adott számlálóállapot dekódolásával állítunk elő) bármelyik állapotából a számlálót a kiindulási állapotába lehet vinni, miáltal a ciklus tetszőlegesen lerövidíthető.  
 G - "L" szinttel engedélyezi a számlálást, "H" szint esetén a számlálás megáll és a további órajel-impulzusok hatástalanok  
 D/Ū - "H" szint: számlálás le (down); "L" szint: számlálás fel (up)

#### Kimenetek:

- QA,..QD - adat kimenetek  
 MX/MN - a számlálási ciklus utolsó állapotát (felfele számlálásnál ez az 1111, lefele pedig a 0000) "H" szinttel jelzi, egyébként értéke "L".  
 RCO - "L" szinttel jelzi, amikor a MX/MN "H" szinten van és az órajel pedig "L" szinten

Az utolsó két kimenet teszi lehetővé több számláló egymás után kapcsolását, mely történhet aszinkron és szinkron módon is. (Szinkron felépítésnél az engedélyező bemenet felhasználására is szükség van)

#### Feladat, 12. mérés:

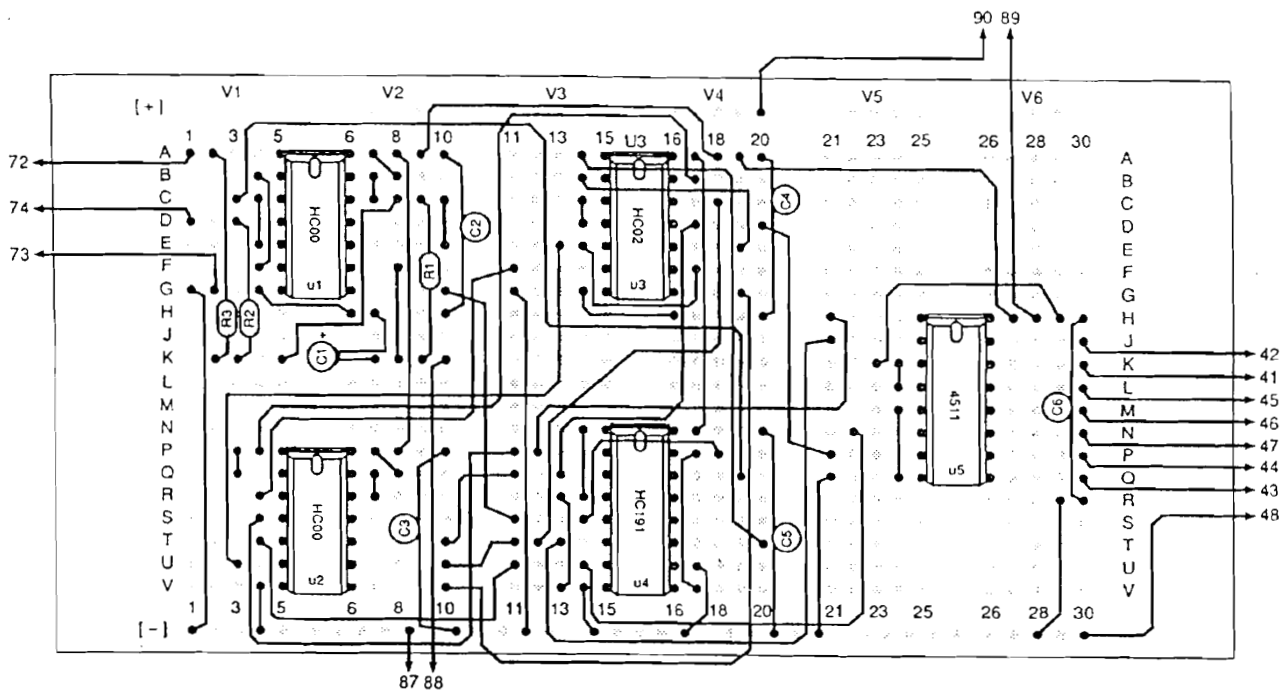
(381)

Az alábbi kapcsolás egy 8421 kódú BCD előre-hátra számláló és hétszegmenses dekódolás megvalósítása integrált áramkörökkel.

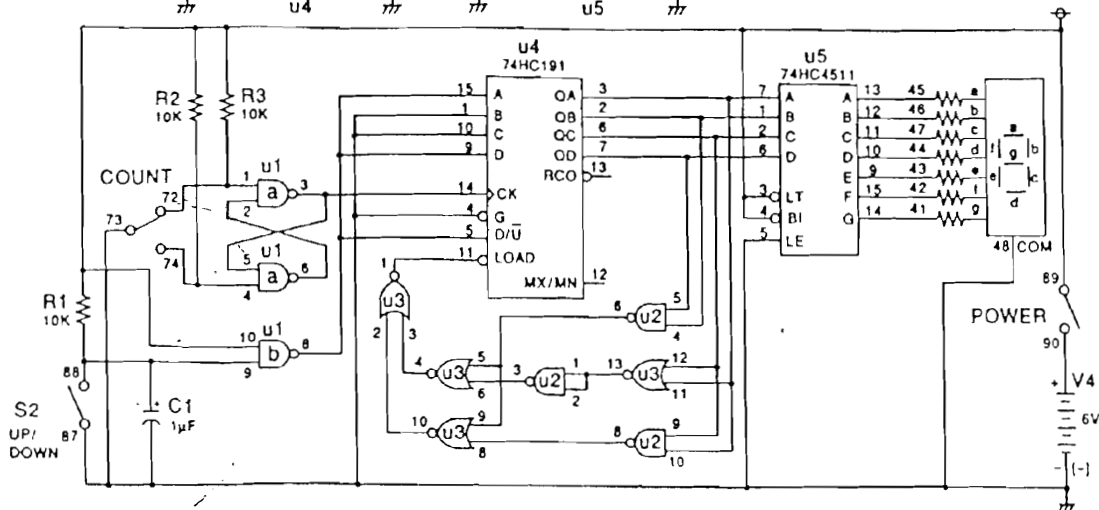
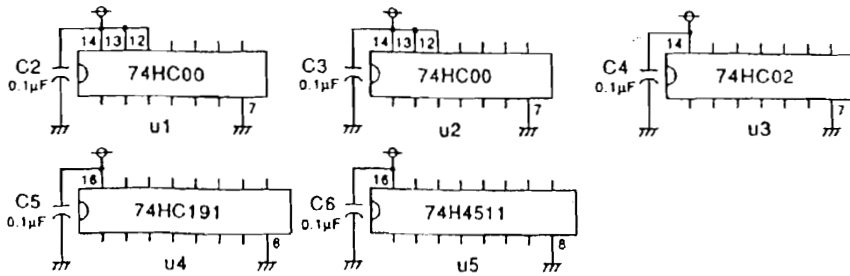
S1-el lehet az áramkört feszültség alá helyezni. S2 kapcsoló a felfele/lefele számlálás átváltását vezérli. A választókapcsoló egy oda-vissza váltásával egy órajelimpulzust adunk a számlálóra.

- A kapcsolás összeállítása után ellenőrizze annak működését: számláltassa körbe a számlálót a kétféle üzemmódban és táblázatban rögzítse az egymás után következő egyes állapotokat! A táblázatba a bináris kódokat is írja be, amit a megfelelő pontokon 4 db LED és 4 db 1 k $\Omega$ -os ellenállás soros bekötésével ellenőrizzen!
- Elemezze a bináris számláló BCD számlálóvá alakításának módját! Mire szolgálnak U1, U2 és U3 kapuáramkörei?
- Készítsen kétszámjegyű aszinkron BCD számlálót két mérőpanel összekapcsolásával! Órajelként használjon TTL négyszögjelet!





U1	74HC00	U4	74HC191	R1	10KΩ	C1	1μF	C4	0.1μF
U2	74HC00	U5	74HC4511	R2	10KΩ	C2	0.1μF	C5	0.1μF
U3	74HC02			R3	10KΩ	C3	0.1μF	C6	0.1μF



Forrás: [1]

## 13. Gyakorló feladatok

## KOMBINÁCIÓS HÁLÓZATOK, BOOLE-ALGEBRA

## 1. feladat

Valósítsa meg minimális számú NAND kapuval a következő kétváltozós  $f(a,b)$  függvényeket:

Adja meg az algebrai átalakítást és a kapcsolási rajzot is!

OR, NOR, XOR (antivalencia), XNOR (ekvivalencia)

## 2. feladat

Valósítsa meg minimális számú NAND kapuval a következő logikai függvényeket, feltételezve, hogy a változók negált értékei nem állnak rendelkezésre:

- $f = \bar{b}c(a + \bar{d}) + bd(a + \bar{c})$
- $f = \bar{a} + b(\bar{c} + de)$
- $f = abd + b\bar{d} + ac\bar{d}$

## 3. feladat

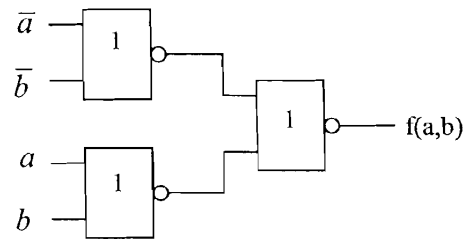
Ismételje meg a 2. feladatot NOR kapuk felhasználásával! Most a negált bemenő jelek is rendelkezésre állnak.

## 4. feladat

Valósítsa meg minimális számú NOR kapuval a kétváltozós  $f(a,b) = a \cdot b$  AND függvényt! Adja meg az algebrai átalakítást és a kapcsolási rajzot is!

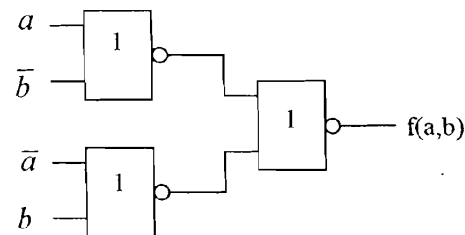
## 5. feladat

Az alábbi kapcsolással adott az  $f(a,b)$  logikai függvény. Írja fel algebrai alakban és egyszerűsítse a Boole-tételek felhasználásával! (diszjunktív normálalakba)



## 6. feladat

Az alábbi kapcsolással adott az  $f(a,b)$  logikai függvény. Írja fel algebrai alakban és egyszerűsítse a Boole-tételek felhasználásával! (diszjunktív normálalakba)



## 7. feladat

Igazságtábla segítségével bizonyítsuk be a következő logikai kifejezés helyességét:

$$\bar{a}b + ac = (a + d)(\bar{b} + c)(a + \bar{d})$$

## 8. feladat

Igazolja a következő azonosságokat a Boole-tételek alkalmazásával:

1.  $ab\bar{d} + a\bar{b}\bar{d} + ab\bar{c} = a\bar{d} + ab\bar{c}$
2.  $acd + a\bar{c}d + \bar{a}d + bc + b\bar{c} = b + d$
3.  $wy\bar{z} + \bar{w}x\bar{z} + \bar{x}y\bar{z} = \bar{z}(w + x + \bar{y})(\bar{w} + \bar{x} + y)$
4.  $(y + \bar{z})(w + x)(\bar{y} + z)(y + z) = yz(w + x)$

## 9. feladat

Egyszerűsítse a következő kifejezéseket a Boole-tételek alkalmazásával:

1.  $\bar{a}bcd + cd + ab\bar{c}d$
2.  $a(\bar{b}c + \bar{a}d) + a(b\bar{c} + ad)$

- 3.  $\bar{a}\bar{b}e + \bar{c}e(b\bar{e} + a\bar{c}\bar{e}) + a\bar{e} + a\bar{e}c$
- 4.  $\bar{v}z(v + \bar{z}) + vwy + yz(w + y) + vw\bar{y}$

**10. feladat**

Írja fel az alábbi kifejezés egyszerűsített inverzét a Boole-tételek alkalmazásával:

$$b(a\bar{d} + c)(c + d)(a + \bar{b})$$

**11. feladat**

Végezze el a 8., 9. és 10. feladatokat KV-diagram segítségével!

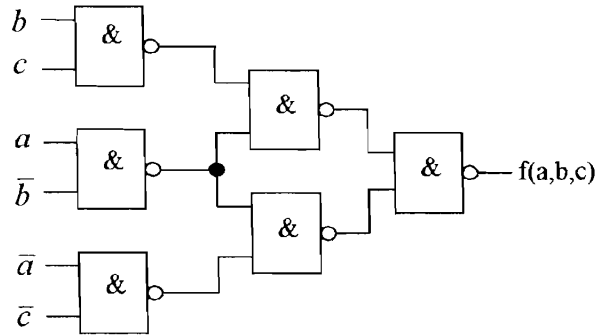
**12. feladat**

Minimalizálja a következő függvényeket KV-diagram felhasználásával:

- 1.  $\bar{a}b\bar{c} + ab\bar{c} + \bar{a}\bar{b}c + abc + a\bar{b}c$
- 2.  $abc + \bar{a}b\bar{c} + \bar{a}\bar{b}c + \bar{a}\bar{b}\bar{c} + \bar{a}bc + a\bar{b}c$
- 3.  $a\bar{b} + a\bar{c} + \bar{a}b + \bar{a}c + \bar{b}\bar{c} + \bar{a}\bar{b}\bar{c}$
- 4.  $c(a\bar{b} + ab\bar{c} + a\bar{b}c) + a\bar{c}$

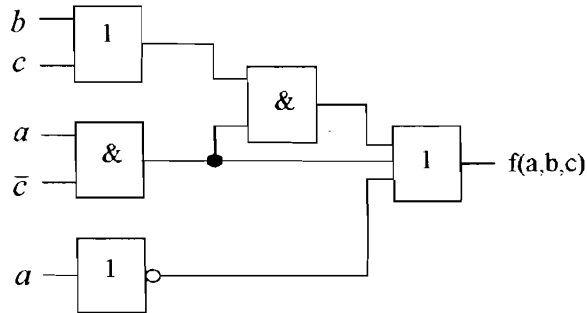
**13. feladat**

Elemezze az alábbi kapcsolást (algebrai alak, KV-diagram) és írja fel a függvényt minimalizált diszjunktív normálalakban:



**14. feladat**

Elemezze az alábbi kapcsolást (algebrai alak, KV-diagram) és írja fel a függvényt minimalizált diszjunktív normálalakban:



**15. feladat**

Az alábbi KV-diagramban adott az  $f(a,b,c,d)$  függvény.

- 1. Határozza meg a legegyszerűbb diszjunktív normálalakot!
- 2. Határozza meg a legegyszerűbb konjunktív normálalakot!
- 3. Valósítsa meg a függvényt NAND kapuk és inverterek felhasználásával!
- 4. Valósítsa meg a függvényt NOR kapuk és inverterek felhasználásával!

		a	
		b	
		0	1
c	d	0	0
		0	0
		1	0
		1	0

**16. feladat**

Adott a következő függvény:  $f(a,b,c,d) = \bar{a}bcd + abc + dc + d\bar{c}b + \bar{a}bc$

1. Készítse el a függvény KV-diagramját!
2. Minimalizálja a függvényt konjunktív normálalakban!

**17. feladat**

Adott az  $f(a,b,c,d)$  logikai függvény igazságtáblázata.

Készítse el a KV-diagramot és

1. határozza meg a függvény minimalizált konjunktív normálalakját
2. határozza meg a függvény minimalizált diszjunktív normálalakját!

d	c	b	a	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

**18. feladat**

Adott az  $f(a,b,c)$  függvény KV-diagramja.

3. határozza meg a függvény minimalizált konjunktív normálalakját
  4. határozza meg a függvény minimalizált diszjunktív normálalakját!
- d: don't care

	a		
	b		
	d	1	d
c	d	0	1
	d	d	0

**19. feladat**

Az alábbiak szerint definiálva van egy 3 bemenetű  $f(a,b,c)$  logikai függvény:

A kimenet akkor lesz "1", ha a bemenetek többsége (a három közül kettő vagy mindhárom) "1"-es szinten van (majoritás függvény).

Adja meg a függvényt

- igazságtáblázattal
- KV-diagramban
- kanonikus diszjunktív és kanonikus konjunktív normálalakban
- minimális diszjunktív normálalakban!

**20. feladat**

Az  $f(a,b,c)$  logikai függvény a mintermjeivel van megadva:  $m_1, m_2, m_3, m_4, m_5, m_7$  (6 db) ahol a=LSB, c=MSB.

Adja meg

1. a KV-diagramot és ebből a minimális diszjunktív normálalakot (DNA)
2. a DNA megvalósítás kapcsolási rajzát kétszintes ÉS-VAGY logikai hálózattal
3. a megvalósítást kizárólag NAND kapukkal (algebrai átalakítás DeMorgan tétélekkel)!

**21. feladat**

Az  $f(a,b,c)$  logikai függvény a maxtermjeivel van megadva:  $M_0, M_1, M_4$  (3 db) ahol  $a=LSB, c=MSB$ .

Adja meg

4. a KV-diagramot és ebből a minimális konjunktív normálalakot (KNA)
5. a KNA megvalósítás kapcsolási rajzát kétszintes VAGY-ÉS logikai hálózattal
6. a megvalósítást kizárólag NOR kapukkal (algebrai átalakítás DeMorgan tételekkel)!

**22. feladat**

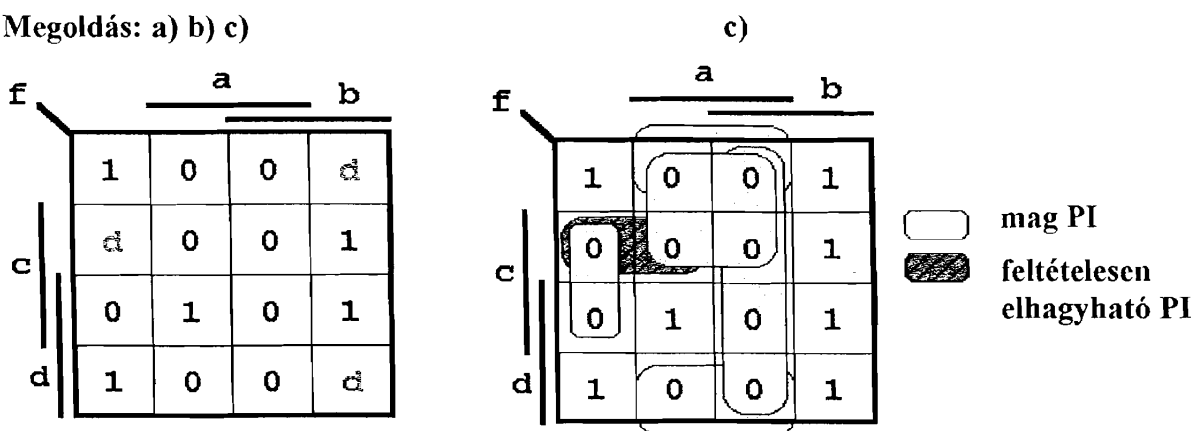
Adott egy logikai függvény  $f(a,b,c,d)$ , melyet az alábbi mintermek és don't care cellák határoznak meg:

mintermek (5 cella):	$m_0, m_6, m_8, m_{13}, m_{14}$
don't-care cellák (3 db):	$mM_2, mM_4, mM_{10}$

[Indexjelölés:  $d = MSB$  és  $a = LSB$ , pl.  $(d,c,b,a)=(1,0,1,0)=10$ ]

- a) Rajzolja fel a függvény KV-Diagramját!
- b) Adjon értéket a "don't-care"-celláknak úgy, hogy a függvényt a minimális (legegyszerűbb) konjunktív normálalakban alakban lehessen megvalósítani!
- c) Jelölje be a primimplikánsokat (PI)! Melyik ezek közül mag-PI, abszolút elhagyható és feltételesen elhagyható PI?
- d) Adja meg a függvény minimális (legegyszerűbb) konjunktív normálalakját!

Megoldás: a) b) c)



d)  $f = (\bar{a} + c) \cdot (a + b + \bar{c}) \cdot (\bar{a} + \bar{b}) \cdot (\bar{a} + d)$

**SZÁMRENDSZEREK ÉS KÓDOK**

**1. feladat**

Fejezze ki a 315,79 decimális számot a következő számrendszerekben (a megoldás lépései legyenek felismerhetőek)

- bináris (tizedesvessző után 12 helyiértékig)
- Oktális
- Hexadecimális
- BCD

**Megoldás:**

**Bináris:** 1 0011 1011 . 1100 1010 0011

**Oktális:** 473 . 6243

**Hexadecimális:** 13B . CA3

**BCD:** 0011 0001 0101 . 0111 1001

Alakítsa át a megadott decimális számokat a megadott alapú számrendszerbe!

7562,45<sub>10</sub> → Alap 8  
 1938,257<sub>10</sub> → Alap 16  
 175,175<sub>10</sub> → Alap 2

**Megoldás:**

7562,45<sub>10</sub> = 16612,3463...<sub>8</sub>  
 1938,257<sub>10</sub> = 792,41CA...<sub>16</sub>  
 175,175<sub>10</sub> = 10101111,001011...<sub>2</sub>

**2. feladat**

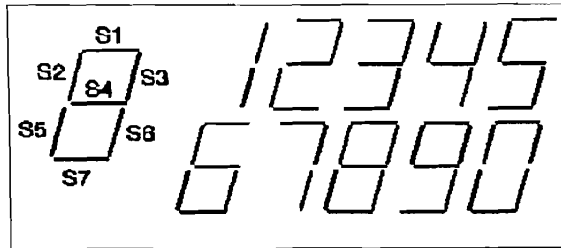
Adott egy memóriablokk legelső és legutolsó tároló-regiszterének címe: 00FF, 41AB.  
 Hány tároló-regisztere van a memóriablokknak?

**Megoldás:**

41AB - 00FF = 40AC<sub>16</sub> = 16556<sub>10</sub>

**3. feladat**

A 0 ... 9 decimális számjegyek 8-4-2-1BCD-kódban állnak rendelkezésre. A számjegyeket 7 szegmenses kijelzővel kell ábrázolni. Az egyes szegmenseket S1 .. S7 jelöléssel láttuk el. Állítsa elő az S3, S4 és S5 szegmensek legegyszerűbb logikai függvényét (D C B A, bináris kód, A=LSB)!



**Megoldás:**

S3 = ( /C + /B + A ) ( /C + B + /A )  
 S4 = ( /C + /B + /A ) (D+ C + B)  
 S5 = /A ( B + /C )  
 S6 = C + /B + A

**4. feladat**

Tervezzen egy dekódolót, mely az Aiken-kódbeli (2-4-2-1-kód) számokat BCD-kódba (8-4-2-1-kód) alakítja át!

**Megoldás:**

Aiken → BCD

B<sub>1</sub> = A<sub>1</sub>  
 B<sub>2</sub> = A<sub>4</sub>A<sub>2</sub> + A<sub>4</sub> $\overline{A_2}$   
 B<sub>3</sub> = A<sub>3</sub>A<sub>2</sub> + A<sub>4</sub>A<sub>3</sub>  
 B<sub>4</sub> = A<sub>3</sub>A<sub>2</sub>

**5. feladat**

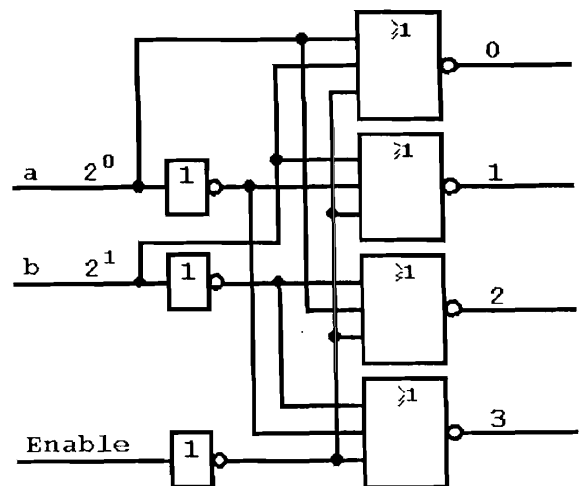
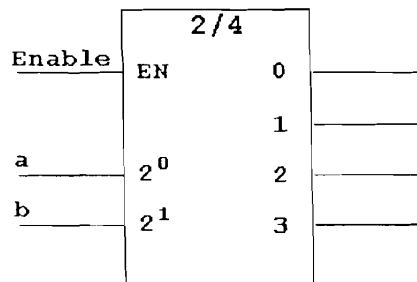
Tervezzen egy dekódolót, mely a Gray-kódban megadott számokat BCD-kódba (8-4-2-1-kód) alakítja át!

**Megoldás:** Gray -> BCD

B1 = A4 A3 A1 + A3 A2 A1 + /A3 /A2 A1 + /A3 A2 /A1 + /A4 A3 /A2 /A1  
 B2 = A3 A2 + A4 A3 A2  
 B3 = A4 A3  
 B4 = A4

**6. feladat**

Tervezzen meg egy 2/4 dekódoló kapcsolást Enable-bemenettel és kizárólag NOR és NOT kapuk felhasználásával!



**Megoldás:**

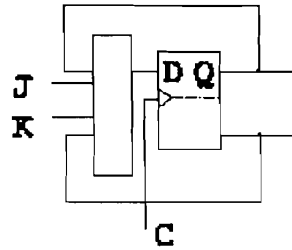
### SZEKVENCIÁLIS HÁLÓZATOK

**1. feladat**

Építsen fel D-flipflop-ból JK-flipflop-ot!  
Adja meg a kiegészítő kombinációs hálózatot és rajzolja meg a kapcsolást!

**Megoldás:**

$$D = \overline{K}n (Qn + Jn) / Qn$$



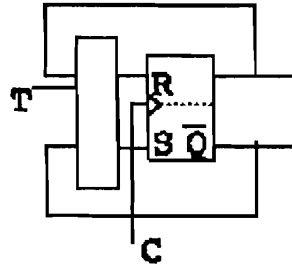
**2. feladat**

Építsen fel RS-flipflop-ból T-flipflop-ot! Adja meg a kiegészítő kombinációs hálózatot és rajzolja meg a kapcsolást!

**Megoldás:**

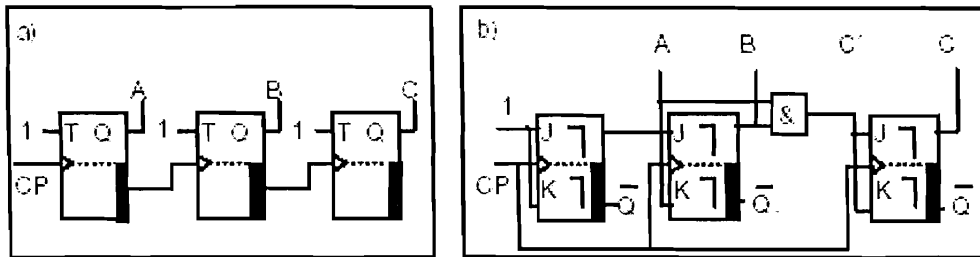
$$S = T / Qn$$

$$R = T (Qn)$$



**3. feladat**

Adja meg a következő kapcsolásokhoz a számlálási (állapotváltási) sorrendet, kezdve 000-



val! Miben különbözik egymástól a két kapcsolás?

**Megoldás:**

a) Aszinkron bináris számláló, sorrend : 0,1,2,3,4,5,6,7,0,..., Modulo 8 számláló

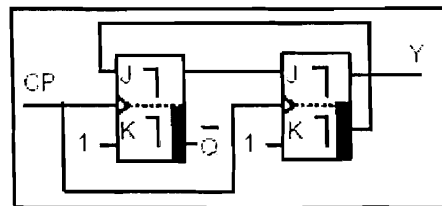
b) Szinkron bináris számláló, sorrend 0,1,2,3,4,5,6,7,0,..., Modulo 8 számláló

**4. feladat**

Milyen frekvencialeosztási arányt valósít meg a következő kapcsolás?

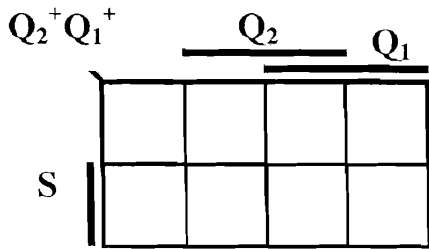
**Megoldás:**

$$f_{CP} : f_Y = 3 : 1$$



**5. feladat**

Tervezen meg két olyan szinkron előre- és hátraszámlálót, amelyek bináris kódot használva ciklikusan számolnak 0-tól 3-ig és 3-tól 0-ig valamint 0-tól 4-ig és 4-től 0-ig! Az S vezérlő bemenet „0” szintje az előreszámlálást, az „1” szintje a hátraszámlálást vezérelje. Építse fel a számlálót JK-flipflop-okból! (Q<sub>i</sub> : FF-kimenetek, Q<sub>1</sub> : LSB)



Megoldás: (0 .. 3 és 3 .. 0)

$$J_2 = Q_1 \bar{S} + \bar{Q}_1 S$$

$$J_1 = 1$$

$$K_2 = Q_1 \bar{S} + \bar{Q}_1 S$$

$$K_1 = 1$$

(0 .. 4 és 4 .. 0)

$$J_3 = Q_1 Q_2 \bar{S} + \bar{Q}_1 \bar{Q}_2 S$$

$$J_2 = Q_1 \bar{S} + Q_3 S$$

$$J_1 = Q_2 + \bar{Q}_3 \bar{S} + Q_3 S$$

$$K_3 = 1$$

$$K_2 = Q_1 \bar{S} + \bar{Q}_1 S$$

$$K_1 = 1$$

**6. feladat**

Tervezzen meg egy szinkron, Aiken-kódú (2-4-2-1) BCD számlálót JK-flipflop-okból!

Megoldás:

4 db JK-FF (Q1 = LSB, Q4 = MSB)

$$J_1 = 1$$

$$J_2 = Q_1 + Q_3 \quad \bar{Q}_4$$

$$J_3 = Q_1 Q_2$$

$$J_4 = Q_3$$

$$K_1 = 1$$

$$K_2 = Q_1$$

$$K_3 = Q_1 Q_2 + \bar{Q}_4$$

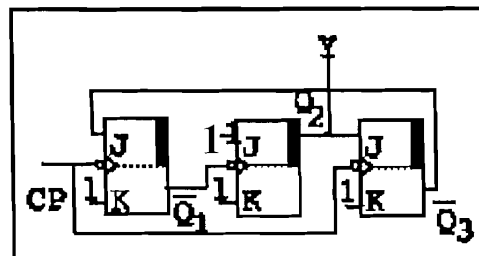
$$K_4 = Q_1 Q_2 Q_3$$

**7. feladat**

Tervezzen meg egy 1: 5 szinkron frekvenciaosztót D-flipflop-okból!

**8. feladat**

Adott az alábbi kapcsolás:



$$Y = Q_2$$

- a) Rajzolja fel az impulzusdiagramot!
- b) Milyen típusú számlálóról van szó?
- c) Mennyi a frekvencia-osztási arány?
- d) Adja meg a kitöltési tényező értékét!

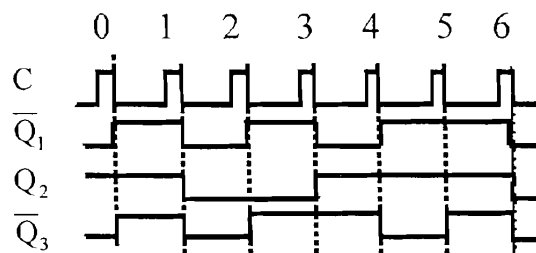
Megoldás:

a)

b) aszinkron frekvenciaosztó

c)  $T_C : T_Y = 1 : 5$

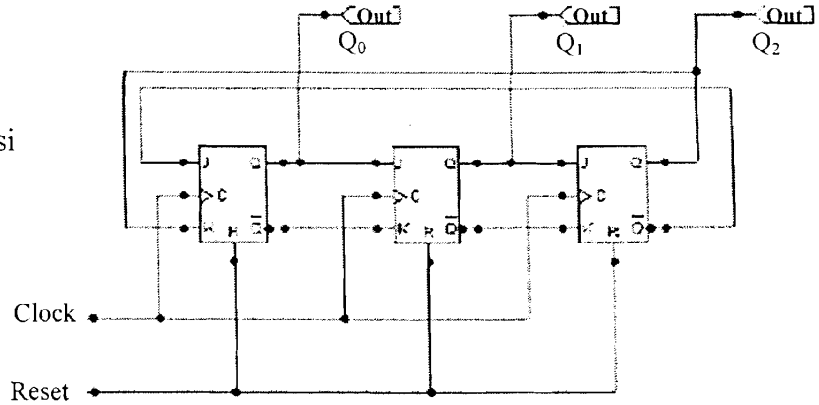
d) 3/5





**9. feladat**

Adott a következő szekven-  
ciális hálózat.  
Mindegyik Flipflop kiindulási  
állapota „0”.



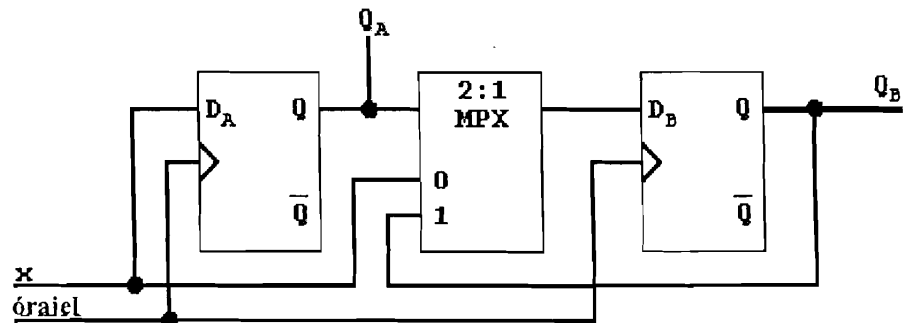
a) Mi a kapcsolás funkciója?

b) Írja be a FF-kimenetek értékeit az alábbi táblázatba! (CP=0 az első órajel előtti állapot)

CP	Q0	Q1	Q2
0	0	0	0
1			
2			
3			
4			
5			
6			
7			

**10. feladat**

Adott a következő  
szekvenciális hálózat:



1. Milyen hálózat ez?

- aszinkron
- kvázi-aszinkron
- szinkron

Miért?.

2. Határozza meg, hogy B Flipflop D-bemenetére (DB) milyen logikai függvény kerül?

3. Adja meg az állapotdiagramot!

4. Az x-bemenetre az alábbi sorrendű jelet adva milyen sorrendiség adódik az  $y = Q_B$  kimeneten? A kiindulási állapot legyen  $Q_A Q_B = 00$ .

x: 0 0 0 1 1 0 1 0 0

y: 0

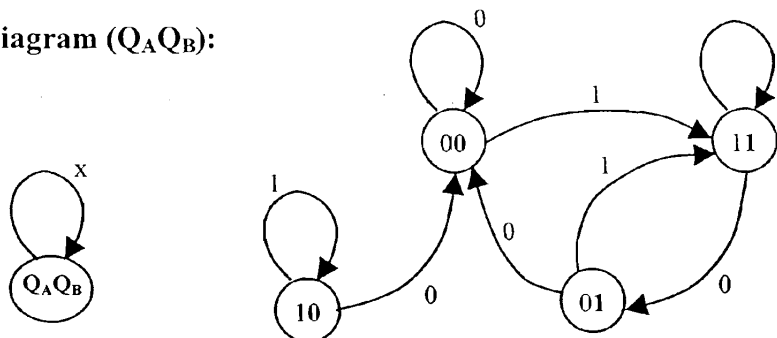
Megoldás:

1. Szinkron, mivel FF<sub>A</sub> és FF<sub>B</sub> egyidejűleg, szinkron kapcsolnak
2. D<sub>B</sub>

$$D_B = \overline{Q_A}x + Q_A Q_B$$

	Q <sub>A</sub>		Q <sub>B</sub>	
D <sub>B</sub>	0	0	1	0
x	1	0	1	1

3. Állapotdiagram (Q<sub>A</sub>Q<sub>B</sub>):



4. Kimeneti válaszjel y = Q<sub>B</sub>:

x: 0 0 0 1 1 0 1 0 0

y: 0 0 0 1 1 1 1 1 0

**11. feladat**

Egy szinkron sorrendi hálózat (automata) két bemenettel x,y és egy kimenettel z rendelkezik.. Ha a bemeneti jelek órajelről órajelre az alábbi sorrend szerint változnak, xy=00, 01, 11, akkor a kimeneti jelet aktivizálni kell (z=1) addig, amíg xy=11 fennáll. A fent definiált jeleken túlmenően rendelkezik a kapcsolás egy órajel-bemenettel valamint egy RESET-bemenettel, melynek aktivizálásával az automatát definiált, kiindulási állapotba (00) lehet vinni órajeltől függetlenül.

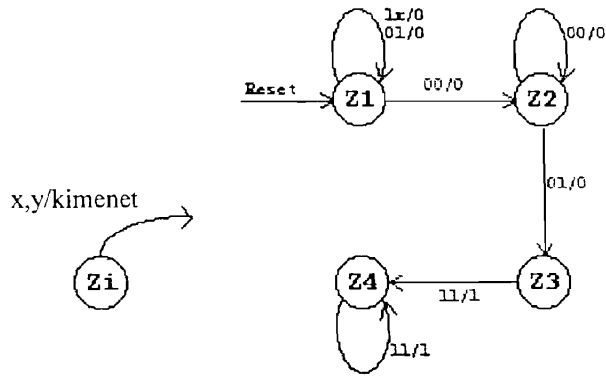
1. Határozza meg az állapot-diagram azon részét, mely az adott bemeneti jelsorozat felismerését (azaz a kívánt funkciót) tartalmazza!
2. Határozza meg a teljes állapotdiagramot!
3. Adja meg az állapotváltási diagramot!
4. Milyen automatáról van szó (Moore, Mealy)?

megoldás:

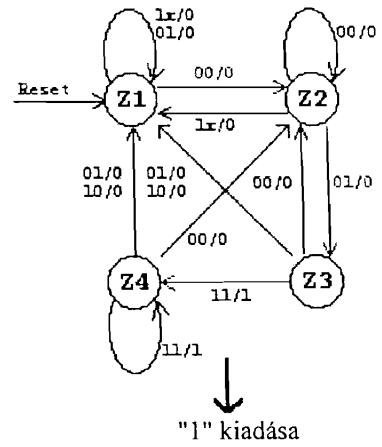
A szükséges állapotok definiálása:

- Z1: Kiindulási állapot (Reset után) Q<sub>B</sub>Q<sub>A</sub>=00
- Z2: (x,y) = (00) ( ) bemenő jelre Clk-hatására beálló állapot Q<sub>B</sub>Q<sub>A</sub>=01
- Z3: (x,y) = (00,01) ( ) bemenő jelsorozatra Clk-hatására beálló állapot Q<sub>B</sub>Q<sub>A</sub>=10
- Z4: (x,y) = (00,01,11) ( ) bemenő jelsorozatra Clk-hatására beálló állapot Q<sub>B</sub>Q<sub>A</sub>=11

**1. Rész-állapotdiagram  
állapotdiagram**



**2. Teljes**



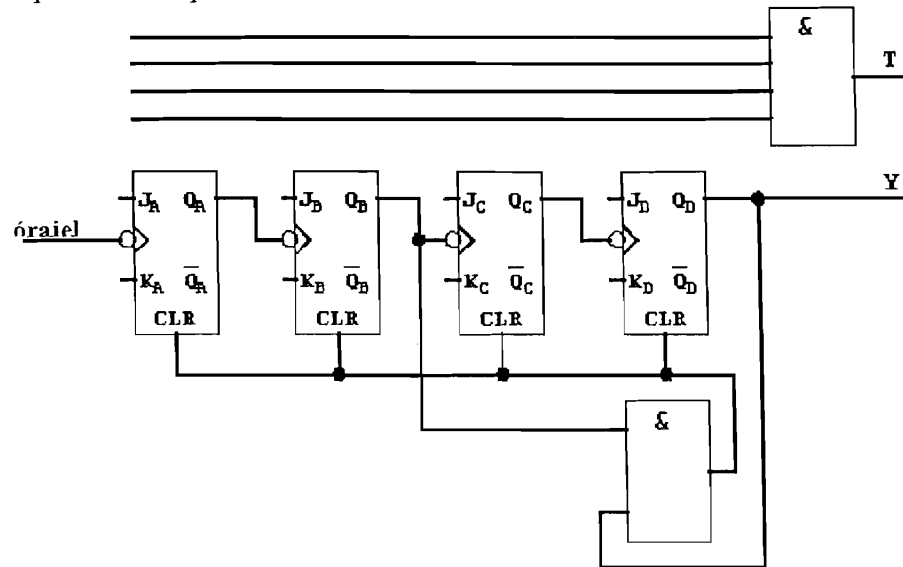
**3. Állapotváltási diagram**

		$Q_a$		$Q_b$
$Z_i$		$Z_2$	$Z_2$	$Z_2$
	$x$	$Z_1$	$Z_1$	$Z_1$
	$y$	$Z_1$	$Z_1$	$Z_4$
	$Y$	$Z_1$	$Z_3$	$Z_1$
		$Z_1$	$Z_2$	$Z_4$

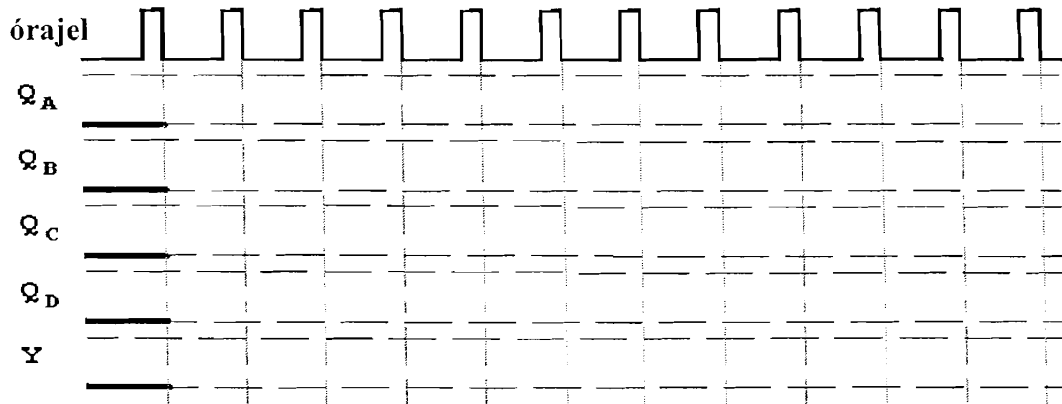
**4. Automata típusa:  
Mealy-Automata**

**12. feladat**

Adott egy JK-flipflop-okból felépülő számláló



a) Határozza meg az impulzusdiagramot 10 órajel-ütemre!

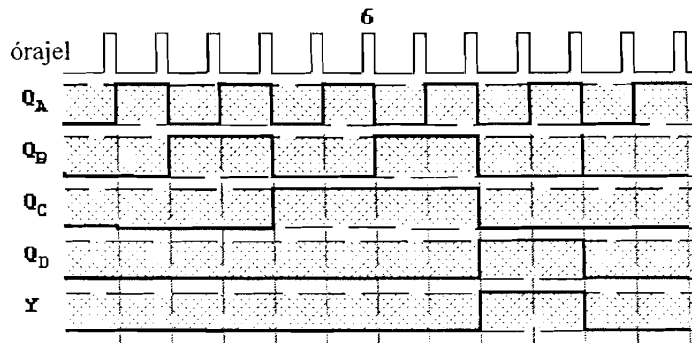


b) Melyik Flipflop-kimeneteket kell a kiegészítésként berajzolt AND-kapu négy bemenetére kötni ahhoz, hogy ennek a kapunak kimenete 6 órajel-impulzus után „H” szintre menjen fel?

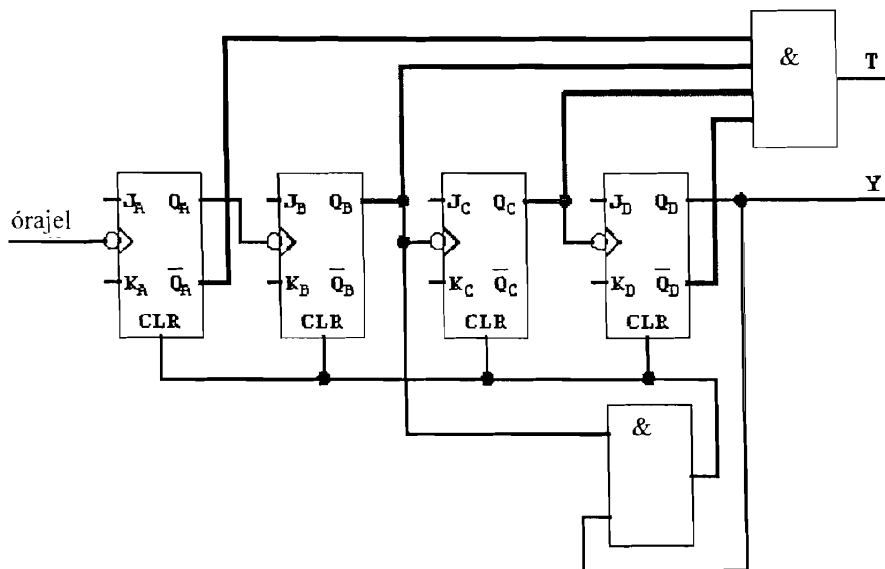
(A be nem kötött JK-bemenetek „H” szintnek felelnek meg (TTL). Minden Q-kimenet kiindulási állapota „0”.)

**Megoldás:**

a) Impulzusdiagram



b)



**13. feladat**

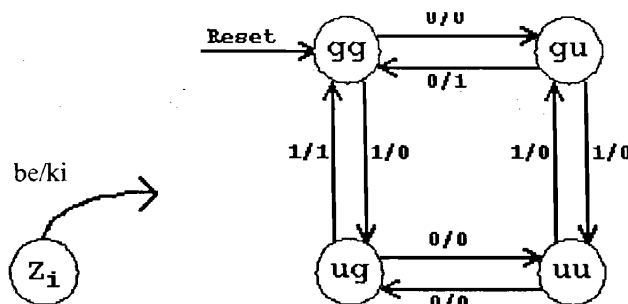
Egy szinkron automata feladata egy jelsorozat ellenőrzése. Ha az órajellel szinkron bemeneti jelsorozat páros számú „0”-ból és páros számú „1”-ből áll, akkor a kimenet ezt egy órajelperiódusnyi ideig „1”-es szinttel jelezzé! A „0”-k és „1”-esek számlálása az alapállapotból kiindulva történik, melybe az automatát egy RESET bemenettel bármikor, órajeltől függetlenül bevihetjük.

(Definiálja az összes lehetséges állapotot, például legyen 00 (uu: páratla, páratlan), 01 (ug: páratlan, páros), 10 (gu: páros, páratlan), 11 (gg: páros, páros).

- a) Határozza meg a teljes állapotdiagramot!
- b) Adja meg a hozzátartozó állapotváltási diagramot!
- c) Hogyan lehet az automatát D-Flipflop-ok segítségével megvalósítani?.

**Megoldás:**

a) Állapotdiagram



állapot-definíció:

- gg: "1"esek száma páros, "0"-k száma páros
- gu: "1"esek száma páros, "0"-k száma páratlan
- ug: "1"esek száma páratlan, "0"-k száma páros
- uu: "1"esek száma páratlan, "0"-k száma páratlan

b) Állapotváltási diagram (Reset nélkül)

$$D_1 = \overline{Q_1} \text{ be} + Q_1 \overline{\text{be}}$$

$$D_0 = \overline{Q_0} \text{ be} + Q_0 \overline{\text{be}}$$

	$Q_1 Q_0^+$		
	$Q_1$		$Q_0$
	01	11	10
	00	01	11
be	10	00	01
	00	10	11
	01	11	00

**14. feladat**

Egy szinkron szekvenciális hálózat két Flipflop-ból áll: FF<sub>A</sub> és FF<sub>B</sub>. Két bemenete van, x és y, valamint egy kimenete, z.

D<sub>A</sub> és D<sub>B</sub> FF-bemenetekre érvényesek az alábbi logikai függvények:

$$D_A = \bar{x} \cdot y + x \cdot Q_A$$

$$D_B = \bar{x} \cdot Q_B + x \cdot Q_A$$

$$Z = Q_B$$

a) Rajzolja meg a logikai kapcsolást!

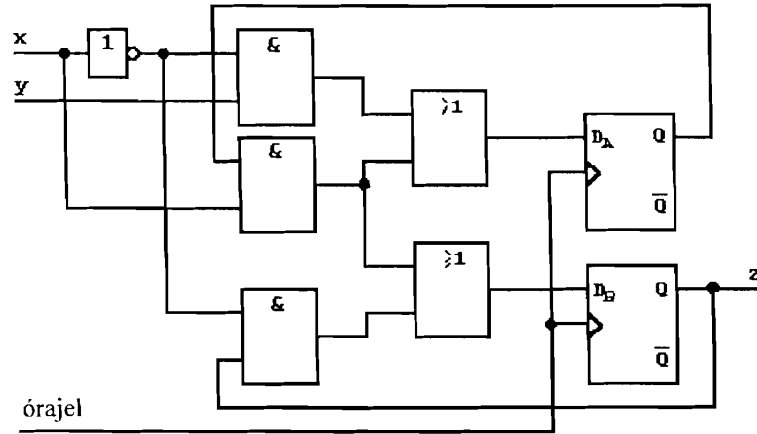
b) Határozza meg az állapotváltási táblázatot az alábbi minta alapján:

Aktuális állapot		Bemenetek		Új állapot		Kimenet
$Q_A$	$Q_B$	x	y	$Q_A^+$	$Q_B^+$	z
0	0	0	0			
.	.	.	.			
.	.	.	.			
.	.	.	.			

c) Adja meg az állapotdiagramot!

Megoldás:

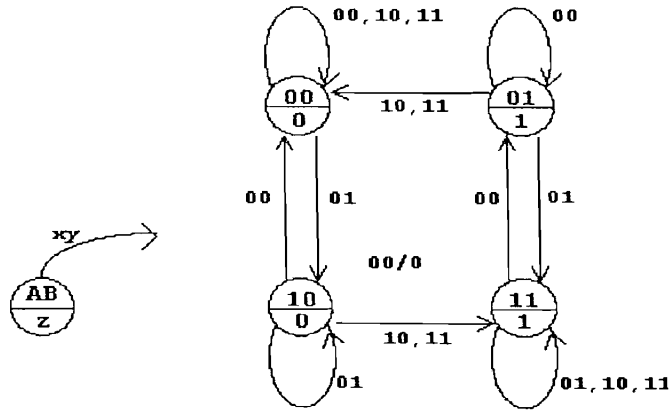
a)



b)

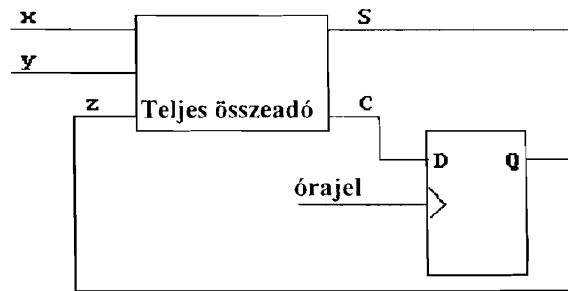
Aktuális állapot		Bemenetek		Új állapot		Kimenet
$Q_A$	$Q_B$	x	y	$Q_A^+$	$Q_B^+$	z
0	0	0	0	0	0	0
0	0	0	1	1	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	1	1
0	1	0	1	1	1	1
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	0	0	0
1	0	0	1	1	0	0
1	0	1	0	1	1	0
1	0	1	1	1	1	0
1	1	0	0	0	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

c)



**15. feladat**

Adott egy automata, mely egy teljes összeadóból és egy D-Flipflop-ból áll:



a) Milyen típusú automatáról van szó?

b) Adja meg az automatára jellemző állapotdiagramot az állapotváltái táblázat segítségével!

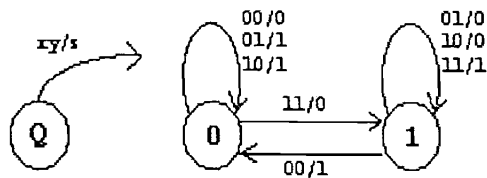
**Megoldás:**

a) Mealy-Automata.

b)

Aktuális állapot	Bemenetek		Új állapot	Kimenet
Q	x	y	Q <sup>+</sup>	z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

c)



## 16. feladat

Tervezen egy szinkron "Modulo-5"-számlálót a következő állapot-sorrenddel::

'100' → '001' → '011' → '101' → '111' → '100' ...

A megvalósításhoz használjon D-Flipflop-ot! Az állapot- és a kimenetei-kód legyenek azonosak.

- a) Hány FF-ra van szükség? Határozza meg az állapot-diagramot, az állapotváltási diagramot és a KV-diagramokat!  
 b) Határozza meg a D.-bemenetek vezérléséhez szükséges logikai függvényeket!

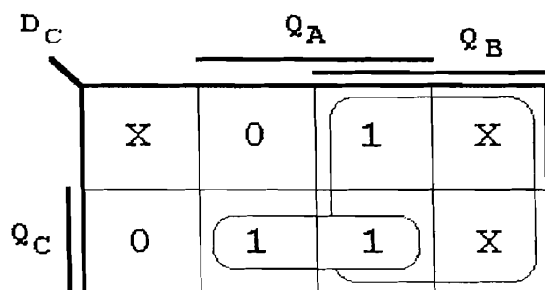
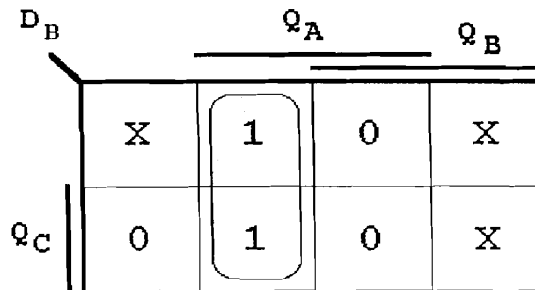
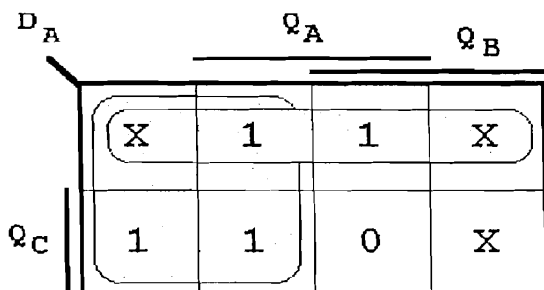
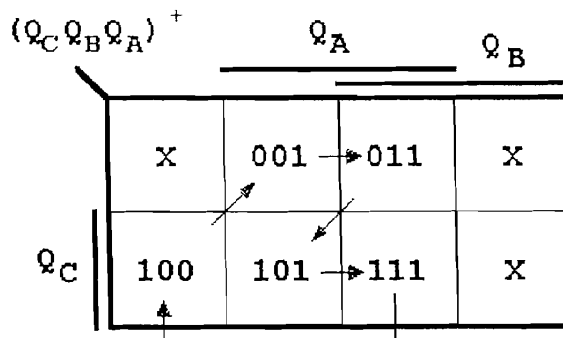
Megoldás:

' $Q_C Q_B Q_A$ ': '100' → '001' → '011' → '101' → '111' → '100' ...

a) szükséges FF-darabszám::

$$n = 3 \quad (2^n = 8 > 5)$$

Állapotváltási diagram:



b)

$$D_A = \overline{Q_B} + \overline{Q_C}$$

$$D_B = Q_A \cdot \overline{Q_B}$$

$$D_C = Q_B + Q_A \cdot Q_C$$



**Tartalomjegyzék**

1. mérés: Logikai alapáramkörök	1
2. mérés: Kombinációs hálózat	6
3. mérés: Kódoló és dekódoló áramkörök	9
4. mérés: Multiplexer, demultiplexer	15
5. mérés: SR-tároló	19
6. mérés: JK-tároló	22
7. mérés: D-tárolók, T-tárolók	25
8. mérés: Regiszterek, léptetőregiszter	28
9. mérés: Multivibrátorok	32
10. mérés: Frekvenciaosztók	37
11. mérés: Számlálók I.	40
12. mérés: Számlálók II.	45
13. Gyakorló feladatok	47

## Irodalomjegyzék

- [1] KAKEN Corporation: 500 in ONE Electronic Lab.  
Owner's Manual, 2001.
- [2] Römer M.: Digitális technikai laboratóriumi útmutató  
BMF KKVKF, 2001.
- [3] Zsom Gyula: Digitális technika I.  
Bp. Műszaki KVK, 2000.
- [4] Tietze, Schenk: Analóg és digitális áramkörök  
Bp. Műszaki KVK, 1990.
- [5] K. Beuth, O. Beuth: Az elektronika aljai 3.  
Bp. Műszaki KVK, 1994.
- [6] Selényi, Benesóczky: Digitális technikai példatár  
Bp. Műegyetem K, 1999.
- [7] Szűcs, Zimányi: Digitális integrált áramkörök: Mérési segédlet  
Bp. Műegyetem K, 1994.
- [8] Eberhard Kühn: Handbuch TTL- und CMOS- Schaltkreise  
VEB Verlag Technik, Berlin, 1986.

