Tantárgy: DIGITÁLIS ELEKTRONIKA
Tanár: Dr. Burány Nándor

4. félév
2+2 óra

1

II. RÉSZ
DIGITÁLIS TERVEZÉS SSI ÉS MSI FUNKCIONÁLIS EGYSÉGEKKEL (HAGYOMÁNYOS TERVEZÉS)

- **Kombinációs** hálózatok
- **Sorrendi** hálózatok
- **Vegyes** hálózatok

2

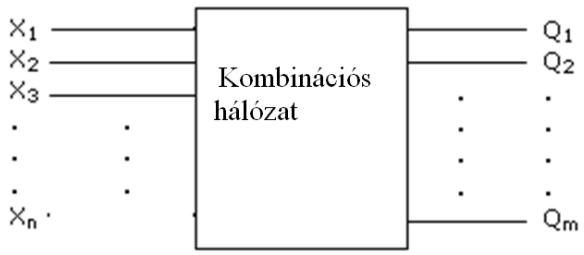
2.a DIGITÁLIS HÁLÓZATOK FELOSZTÁSA

- A digitális áramkörök **összetett feladatot** megoldó együttesét **digitális hálózatnak** nevezzük.
- Két csoportot szokás megkülönböztetni, a tisztán **kombinációs** hálózatot és a **sorrendi, vagy szekvenciális** hálózatot.
- A két hálózattípus kombinációját nevezhetjük **vegyes** hálózatoknak.

3

2.b KOMBINÁCIÓS HÁLÓZATOK JELLEMZŐI

- Olyan digitális áramkörök, amelyeknél **a kimenetek logikai állapota (Q_i) csak a bemeneti jelek (X_i) pillanatnyi értékektől függ**, megfelelő logikai függvény szerint.
- A **kimeneti értékek nem függnek** a bemenetre érkező jelek **sorrend**jétől, értékváltásuk **irány**ától, a megelőző időpontokban kialakult logikai értékeiktől.



4

2.c KOMBINÁCIÓS HÁLÓZATOK JELLEMZŐI

- A 3. fejezetben tárgyalandó sorrendi hálózatok viselkedése lényegesen eltér a kombinációs hálózatok viselkedésétől.
- Gyártanak **SSI és MSI kombinációs áramköröket** szinte minden áramkörcsaládban. Ezekkel ma is építhetők összetettebb digitális rendszerek, de ez a módszer jórészt elavult.
- **A VLSI áramkörök** (mikrovezérlők, PLD-k) **belső felépítése is tartalmazza ezeket az elemeket**. A működés leírásakor az itt ismertetendő alapfogalmakat használják.

5

2.d KOMBINÁCIÓS FUNKCIONÁLIS EGYSÉGEK

Jellemző kombinációs hálózatok:

1. Illesztő áramkörök
2. Logikai kapuk
3. Dekóderek
4. Kóderek
5. Kódátalakítók
6. Multiplexerek
7. Demultiplexerek

6

2.1 ILLESZTŐ ÁRAMKÖRÖK

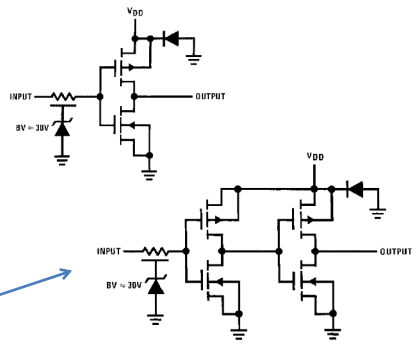
A következő **feladatokat** látják el:

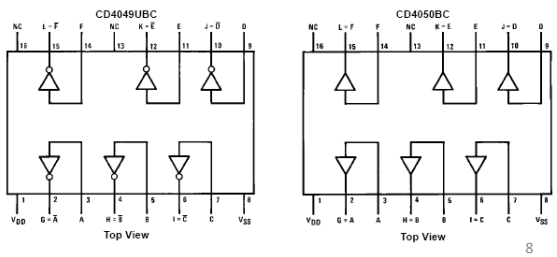
1. **Impedancia illesztés** (áram erősítés)
2. **Szintillesztés** (feszültségszintek növelése vagy csökkentése)
3. **Invertálás**
4. **Közös vezetékek használata**.

7

2.1.1a IMPEDANCIA ILLESZTÉS

- **Nagy bemeneti/kis kimeneti ellenállás**
- Kis bemeneti áram, nagy kimeneti terhelhetőség
- Invertáló illesztő (logikai inverter) **belső szerkezete** (CMOS kivitel)
- **Neminvertáló illesztő** (buffer) (CMOS kivitel)
- Példa: **CD4049/4050** - hat invertáló/neminvertáló illesztő egy DIL16 tokozásban (lábkiosztás)

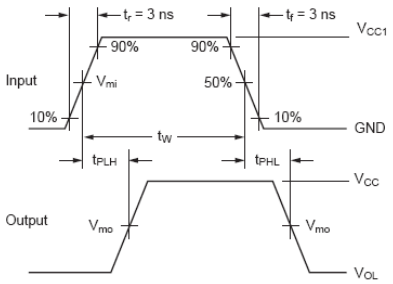




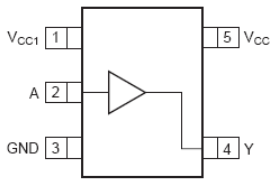
8

2.1.1.b SZINTILLESZTÉS

- **Két különböző tápfeszültségen (V_{CC1}, V_{CC}) működő digitális rendszer** között viszi át a jeleket.
- Összehangolja a logikai szinteket az adott oldalra alkalmazott tápfeszültséggel.



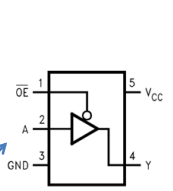
- Példa: Fairchild FXLP34
- $1V<V_{CC}, V_{CC1}<3,6V$
- $0<V_{IL}<0,35V_{CC1}$
- $0,65V_{CC1}<V_{IH}<V_{CC1}$
- $V_{OL}\approx 0V$
- $V_{OH}\approx V_{CC}$

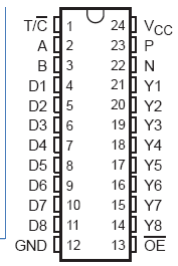


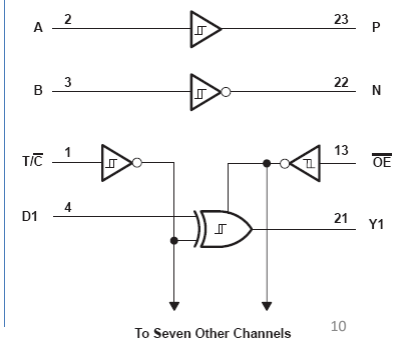
(Top View)

2.1.2 HÁROMÁLLAPOTÚ ILLESZTŐK

- A bemenetről (A) a logikai jel csak akkor jut a kimenetre (Y) ha **/OE=0**.
- Ha **/OE=1**, a kimenet a harmadik állapotban van (nagyimpedanciás állapot)
- 1. Példa: **NC7SZ125** TinyLogic UHS Buffer with 3-STATE Output
- A vezérlés (OE) történhet alacsony vagy magas logikai szinttel.
- Igény szerint választható invertáló illesztő vagy lehet univerzális.
- Beépíthető hiszterézis az átviteli jellegörbébe.
- 2. Példa: **SN74LV8151** 10-bit universal Schmitt-trigger buffer with 3-state output
- Vezérlőjeltől (T/C) függően invertál vagy nem invertál (ezért univerzális).

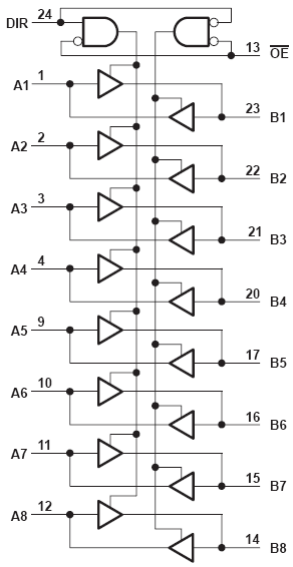






2.1.3 KÉTIRÁNYÚ ILLESZTŐK

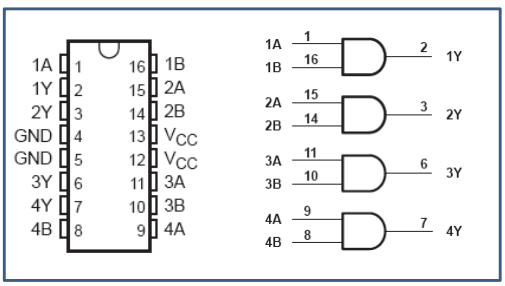
- Két jelvonalat két drb. háromállapotú illesztővel kötünk össze.
- A DIR bemenetre vezetett logikai szinttől függően a **jeleket vagy egyik vagy másik irányba viszi át** (A→B vagy B→A).
- Ha $\overline{OE} = 1$, mindkét oldalon (A is és B is) magas impedanciát mutat.
- Példa: **74AC11245** Octal bus transceiver with three state outputs (Texas Instruments)

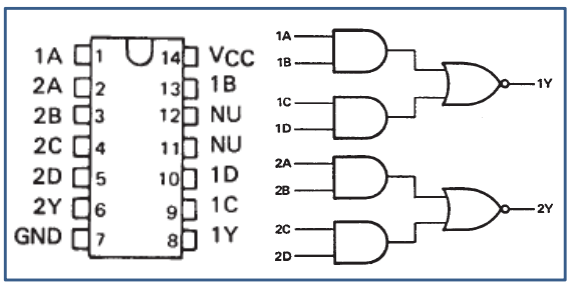


11

2.2 LOGIKAI KAPUK

- Egyszerű logikai függvényeket valósítanak meg: ÉS, VAGY, NEM-ÉS, NEM-VAGY, kizáró VAGY...
- 1. Példa: **74AC11008**, **négy darab ÉS kapu egy tokozásban**.
- Vannak egy kapus tokozások is (**LittleLogic**-Texas Instrumants, **TinyLogic**-Fairchild)
- Vannak IC-k kapuk kombinációjával is.
- 2. Példa: **SN74LS51 AND-OR-INVERT** gates.



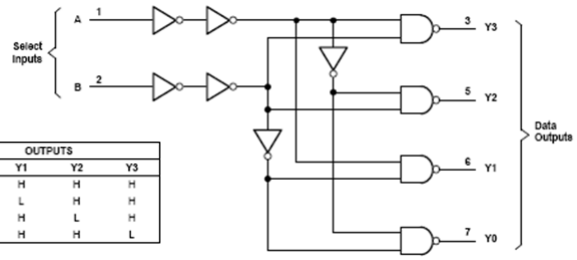


12

2.3. DEKÓDEREK

- **Több bemenet (n) és több kimenet (≤2^n).**
- A bemenetek **binárisan kódolt számok**.
- Minden egyes számkód megjelenésekor **másik kimeneti vonal aktivizálódik.**

- Példa: **SN74LVC1G139**, közönséges (teljes) 2/4-es dekóder:



SELECT INPUTS		OUTPUTS			
B	A	Y0	Y1	Y2	Y3
L	L	L	H	H	H
L	H	H	L	H	H
H	L	H	H	L	H
H	H	H	H	H	L

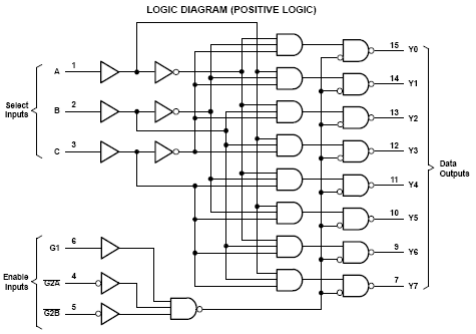
13

2.3.1 TELJES DEKÓDER

- Minden lehetséges bemeneti variációra aktivizálódik egy kimenet, de csak egy kimenet.
- Példa: **SN54LVC138A**, 3/8-as teljes dekóder.
- Az engedélyező bemenetek megfelelő használatával a kapacitás bővíthető vagy demultiplexer is megvalósítható (2.7 alfejezet).

FUNCTION TABLE

ENABLE INPUTS			SELECT INPUTS			OUTPUTS							
G1	$\overline{G2A}$	$\overline{G2B}$	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	L	H	L	H	H	H	H	H	H	L	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

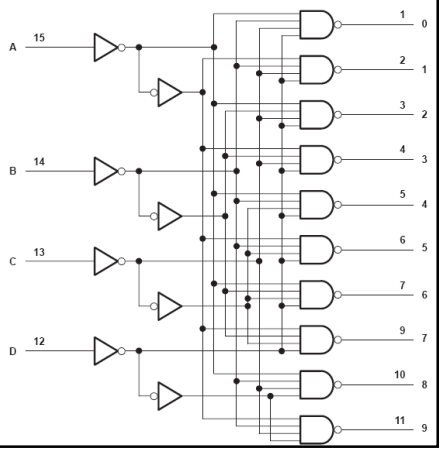


2.3.2 NEM TELJES DEKÓDER

- Ha **nincs szükség mind a 2^n kimenetre**...
- Lehetőség van minimalizálásra.
- Leggyakoribb eset: 4/10-es nem teljes dekóder.
- Példa: **SN74HC42**
  (nincs minimalizálva)

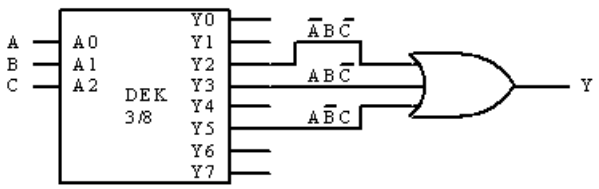
FUNCTION TABLE

NO.	INPUTS				OUTPUTS									
	D	C	B	A	0	1	2	3	4	5	6	7	8	9
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
Invalid	H	L	H	L	H	H	H	H	H	H	H	H	H	H
	H	L	H	H	H	H	H	H	H	H	H	H	H	H
	H	H	L	L	H	H	H	H	H	H	H	H	H	H
	H	H	L	H	H	H	H	H	H	H	H	H	H	H
	H	H	H	L	H	H	H	H	H	H	H	H	H	H
	H	H	H	H	H	H	H	H	H	H	H	H	H	H



2.3.3 LOGIKAI FÜGGVÉNYEK MEGVALÓSÍTÁSA DEKÓDERREL

- A dekóder minden egyes kimenete a bemenő változók egy **logikai szorzatának** felel meg.
- A megfelelő szorzatokat összegezve egy VAGY kapuval tetszőleges logikai függvény állítható elő.
- Példa: $Y = \overline{A}B\overline{C} + AB\overline{C} + A\overline{B}C$



16

2.4 KÓDER

- A számítógépek és más digitális berendezések **binárisan kódolt információkat** dolgoznak fel.
- A kód bizonyos számú (n) logikai jelből (bit) áll.
- n bittel legtöbb 2^n bemeneti jelet lehet kódolni.

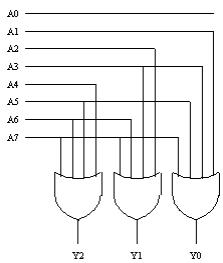
17

2.4.1 TELJES KÓDER

- **2^n bemenet, n kimenet**
- **Nem igazán használható**, mert ha a bemeneteken egyszerre több egyes jelenik meg, a kimenet téves kódot ad.
- Nem gyártanak ilyen alkatrészt külön!
- Példa: 3/8-as teljes kóder.

A7	A6	A5	A4	A3	A2	A1	A0	Y2	Y1	Y0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Y0=A1+A3+A5+A7
Y1=A2+A3+A6+A7
Y2=A4+A5+A6+A7



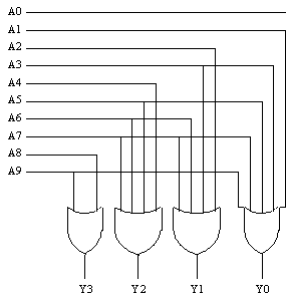
18

2.4.2 NEM TELJES KÓDER

- $<2^n$ bemenet, n kimenet
- **Ez sem igazán használható**, mert ha a bemeneten egyszerre több egyes jelenik meg, a kimenet téves kódot ad.
- Nem gyártanak ilyen alkatrészt külön!
- Példa: 10/4-es kóder.

Ai	Y3	Y2	Y1	Y0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Y0=A1+A3+A5+A7+A9
Y1=A2+A3+A6+A7
Y2=A4+A5+A6+A7
Y3=A8+A9



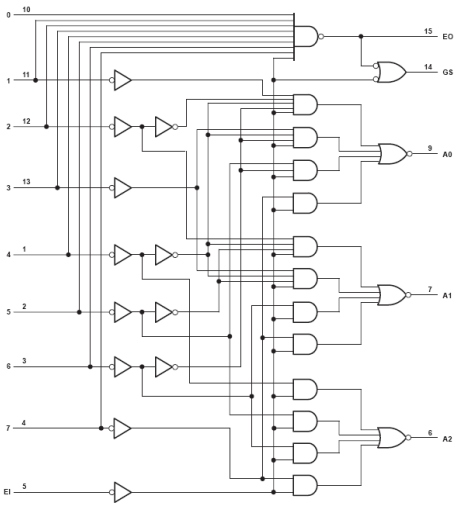
19

2.4.3.a PRIORITÁSOS KÓDER

- Nincs gond ha egyszerre **több bemenet aktív**, mindig csak a legnagyobb prioritásút (itt a legnagyobb sorszámú) veszi figyelembe.
- Példa: **SN74HC148**, 8/3-as prioritásos kóder

Input									Output				
E_I	D7	D6	D5	D4	D3	D2	D1	D0	GS	Q2	Q1	Q0	E_O
0	X	X	X	X	X	X	X	X	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	X	X	X	X	X	X	X	1	1	1	1	0
1	0	1	X	X	X	X	X	X	1	1	1	0	0
1	0	0	1	X	X	X	X	X	1	1	0	1	0
1	0	0	0	1	X	X	X	X	1	1	0	0	0
1	0	0	0	0	1	X	X	X	1	0	1	1	0
1	0	0	0	0	0	1	X	X	1	0	1	0	0
1	0	0	0	0	0	0	1	X	1	0	0	1	0
1	0	0	0	0	0	0	0	1	1	0	0	0	0

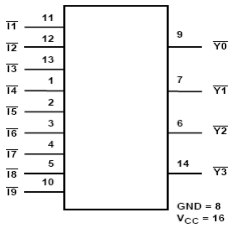
X = Don't Care Logic 1 ≡ High Logic 0 ≡ Low



20

2.4.3.b PRIORITÁSOS KÓDER

- Prioritásos kódernél is lehet a bemenetek száma kisebb mint 2^n.
- Példa: **SN74HC147**, 10/4-es prioritásos kóder.



INPUTS									OUTPUTS			
$\overline{I1}$	$\overline{I2}$	$\overline{I3}$	$\overline{I4}$	$\overline{I5}$	$\overline{I6}$	$\overline{I7}$	$\overline{I8}$	$\overline{I9}$	$\overline{Y3}$	$\overline{Y2}$	$\overline{Y1}$	$\overline{Y0}$
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

H = High Logic Level, L = Low Logic Level, X = Don't Care

21

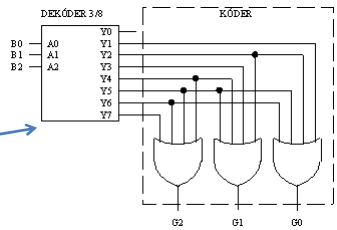
2.5 KÓDÁTALAKÍTÓ

- **Egyik kódrendszerből a másikba** alakítja az információt.
- **Standard megoldás**: dekóder és kóder kaszkád kötése.
- Rendszerint van ennél **egyszerűbb megoldás**:
  1. a logikai függvények minimalizációjával kapott hardver
  2. szoftver, kiolvasás táblázatból.

22

2.5.1 TERMÉSZETES BINÁRIS/GRAY-FÉLE KÓDÁTALAKÍTÓ

- **Dekóder-kóder kaszkád kötésével** a következő megoldást kapjuk:
- A táblázat alapján a logikai egyenletek:



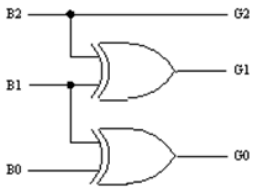
Természetes bináris kód B2B1B0	Gray-féle kód G2G1G0
000	000
001	001
010	011
011	010
100	110
101	111
110	101
111	100

$$G_2 = B_2\overline{B}_1\overline{B}_0 + B_2\overline{B}_1B_0 + B_2B_1\overline{B}_0 + B_2B_1B_0$$
$$G_1 = \overline{B}_2B_1\overline{B}_0 + \overline{B}_2B_1B_0 + B_2\overline{B}_1\overline{B}_0 + B_2\overline{B}_1B_0$$
$$G_0 = \overline{B}_2\overline{B}_1B_0 + \overline{B}_2B_1\overline{B}_0 + B_2\overline{B}_1B_0 + B_2B_1\overline{B}_0$$

- A függvények **minimalizációjával** egyszerűbb kifejezéseket kapunk:

$$G_2 = B_2$$
$$G_1 = B_2 \oplus B_1$$
$$G_0 = B_1 \oplus B_0$$

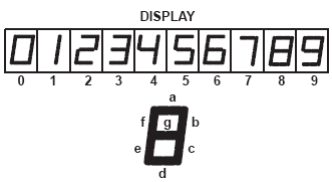
- A minimalizált függvények alapján megszerkeszett egyszerűbb hálózat:



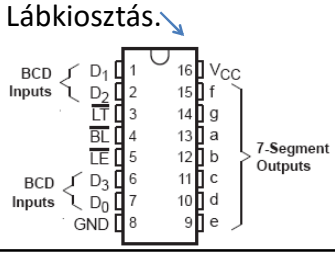
23

2.5.2 BCD/7 SZEGMENSES KÓDÁTALAKÍTÓ

- A hétszegmenses kijelző a tizes számrendszer **számjegyeinek megjelenítésére** szolgál.
- Példa: **SN74HCT4511** BCD-to-7 segment latch/decoder/driver.
- Táblázat (vannak vezérlő bemenetek is).
- Lehetne minimalizálni.
- Lábkiosztás.

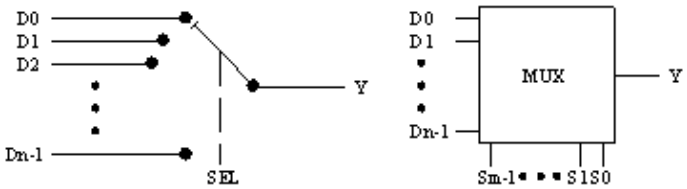


INPUTS							OUTPUTS							
$\overline{LE}$	$\overline{BL}$	$\overline{LT}$	D_3	D_2	D_1	D_0	a	b	c	d	e	f	g	DISPLAY
X	X	L	X	X	X	X	H	H	H	H	H	H	H	8
X	L	H	X	X	X	X	L	L	L	L	L	L	L	Blank
L	H	H	L	L	L	L	H	H	H	H	H	H	L	0
L	H	H	L	L	L	H	L	H	H	L	L	L	L	1
L	H	H	L	L	H	L	H	H	L	H	H	L	H	2
L	H	H	L	L	H	H	H	H	H	H	L	L	H	3
L	H	H	L	H	L	L	L	H	H	L	L	H	H	4
L	H	H	L	H	L	H	H	L	H	H	L	H	H	5
L	H	H	L	H	H	L	L	L	H	H	H	H	H	6
L	H	H	L	H	H	H	H	H	H	L	L	L	L	7
L	H	H	H	L	L	L	H	H	H	H	H	H	H	8
L	H	H	H	L	L	H	H	H	H	L	L	H	H	9
L	H	H	H	L	H	L	L	L	L	L	L	L	L	Blank
L	H	H	H	L	H	H	L	L	L	L	L	L	L	Blank
L	H	H	H	H	L	L	L	L	L	L	L	L	L	Blank
L	H	H	H	H	L	H	L	L	L	L	L	L	L	Blank
L	H	H	H	H	H	L	L	L	L	L	L	L	L	Blank
L	H	H	H	H	H	H	L	L	L	L	L	L	L	Blank
H	H	H	X	X	X	X	†	†	†	†	†	†	†	†



2.6. MULTIPLEXER

- Digitális jelek továbbítása **több bemeneti vonalról** (jelbemenetek, D_0, D_1...D_{n-1}) **egy kimeneti vonalra** (Y).
- Olyan, mint egy egypólusú, többállású kapcsoló.
- Egyszerre természetesen csak egy jelet továbbíthat - **időmultiplex**.
- Hogy melyik jelet továbbítja a kimenetre, azt a választó bemenetek (S_0, S_1, S_{m-1}) határozzák meg.
- Rendszerint $n=2^m$.



25

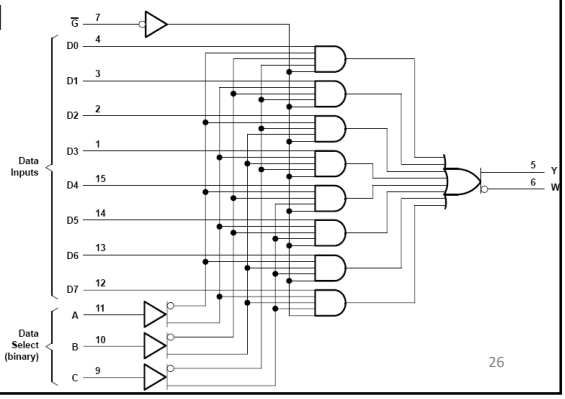
2.6.1 DIGITÁLIS MULTIPLEXER FELÉPÍTÉSE

- A kombinációs táblázat (**CD74AC151**):
- A logikai függvény alakja 8/1-es egyszerű multiplexer esetére:

$$Y = D_0\overline{S}_2\overline{S}_1\overline{S}_0 + D_1\overline{S}_2\overline{S}_1S_0 + \ldots + D_7S_2S_1S_0$$

- A táblázatban szerepel egy vezérlő bemenet is (STROBE), amely minden mástól függetlenül nullát eredményez a kimeneten.
- Van invertált kimenet is.

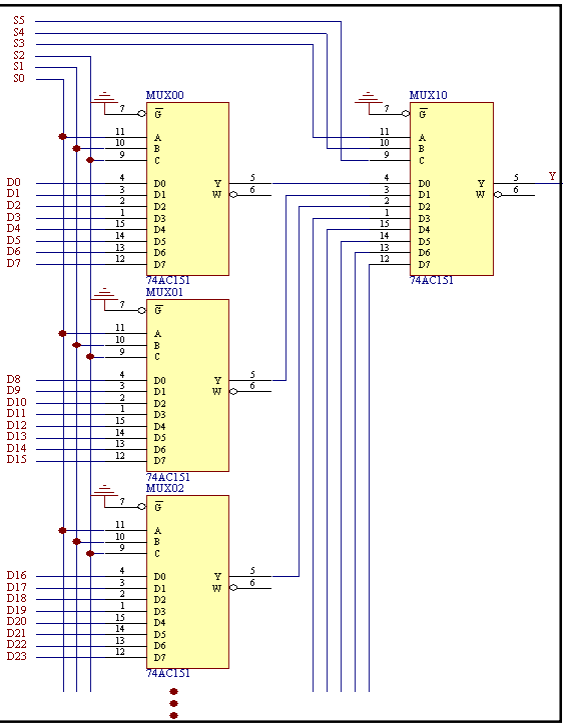
INPUTS				OUTPUTS	
SELECT			STROBE $\overline{G}$		
C	B	A		Y	W
X	X	X	H	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$



26

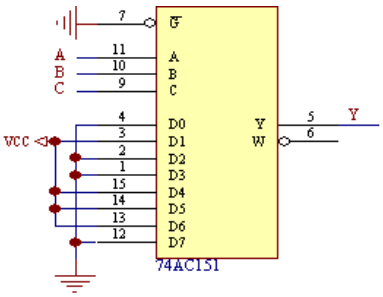
2.6.2 MULTIPLEXER BŐVÍTÉSE

- Nem gyártanak multiplexert tizenhatnál több bemenettel.
- Ha több jelet kell multiplexálni, az megoldható **multiplexerek összekapcsolásával**.
- Példa: 8x8=64 bemenő csatornával rendelkező multiplexer.



2.6.3 LOGIKAI FÜGGVÉNYEK MEGVALÓSÍTÁSA MULTIPLEXERREL

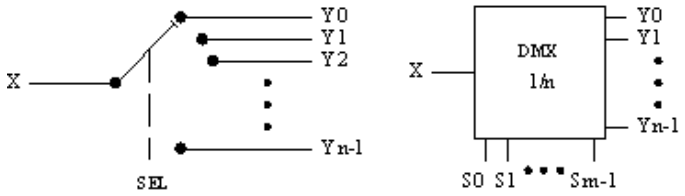
- **A logikai változókat a multiplexer választó bemeneteire kötjük**.
- Az **adatbemenetre** azokat a logikai szinteket hozzuk, amelye az adott függvényre jellemzőek logikai változók adott variációja mellett.
- Példa: $Y = \overline{C}\,\overline{B}A + C\overline{B} + CB\overline{A}$
- A függvényt normál alakra kell hozni: $Y = \overline{C}\,\overline{B}A + C\overline{B}\,\overline{A} + C\overline{B}A + CB\overline{A}$
- Van hatékonyabb módszer is (négyváltozós függvény is megvalósítható ugyanezzel az áramkörrel)!



28

2.7 DEMULTIPLEXER

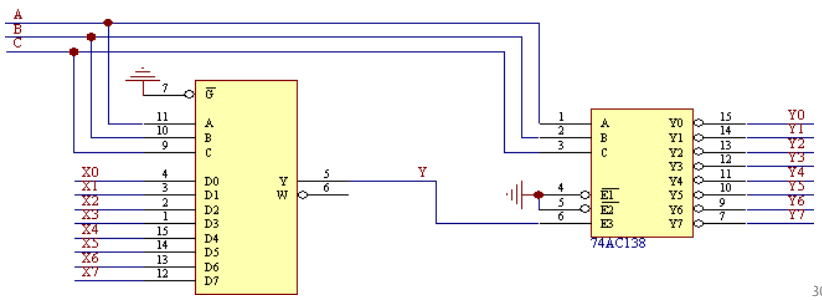
- Digitális jelek továbbítása **egy bemeneti vonalról** (X) **több kimeneti vonalra** (Y_0, Y_1...Y_{n-1}).
- Olyan, mint egy egypólusú, többállású kapcsoló.
- Egyszerre csak egyik kimenet felé továbbíthatja a bemenő jelet - **időmultiplexben érkeznek a jelek**.
- Hogy **melyik jelet továbbítja** a kimenetre, azt a választó bemenetek (S_0, S_1, S_{m-1}) határozzák meg.
- Rendszerint $n=2^m$.
- A katalógusok rendszerint ugyanazt az alkatrészt kínálják dekódernek és demultiplexernek.



29

2.7.1 TÖBB ADAT ÁTVITELE KÖZÖS CSATORNÁN

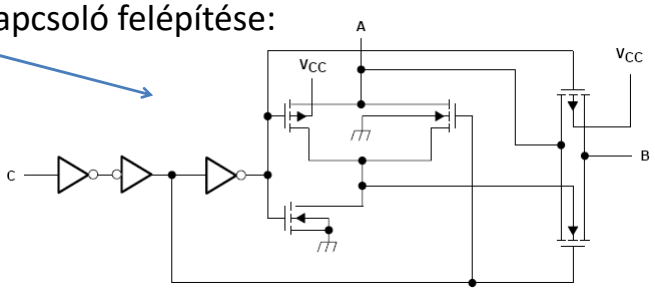
- Multiplexer és demultiplexer kaszkád kötésével **nagyobb számú jel átvihető egy közös jelvonalon** (időmultiplex).
- A multiplexált jel mellett szükséges a **választó bemeneteket** is összekötni, hogy a multiplexer és a demultiplexer szinkronban működjönek.
- A példában nyolc átviteli vezeték helyett négy elegendő. Általános esetben m+1(+1) vezeték kell, ahol $n=2^m$ az átvitelre szánt jelek száma.



30

2.7.2.a ANALÓG MULTIPLEXER/DEMULTIPLEXER

- **Alkalmas analóg jelek átvitelére** is (digitálisra is).
- Ugyanaz az alkatrész **viszi át a jelet mindkét irányban**.
- Analóg kapcsolókat és dekódert tartalmaz.
- Az analóg kapcsoló felépítése:

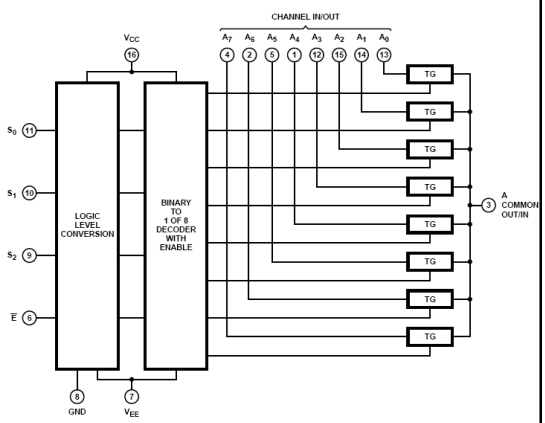


- A és B pontok között teremt kapcsolatot (kis ellenállás).
- C - vezérlőbemenet, C=1 - vezet a kapcsoló.

31

2.7.2.b ANALÓG MULTIPLEXER/DEMULTIPLEXER

- Példa: **CD74HC4051**, nyolccsatornás analóg multiplexer-demultiplexer
- 3/8-as dekóder vezérli az analóg kapcsolókat
- Vezérlőjelek (S0, S1, S2): 0V és 5V.
- Analóg jeltartomány: -5V...+5V. (V_{CC}=5V, V_{EE}=-5V).



32

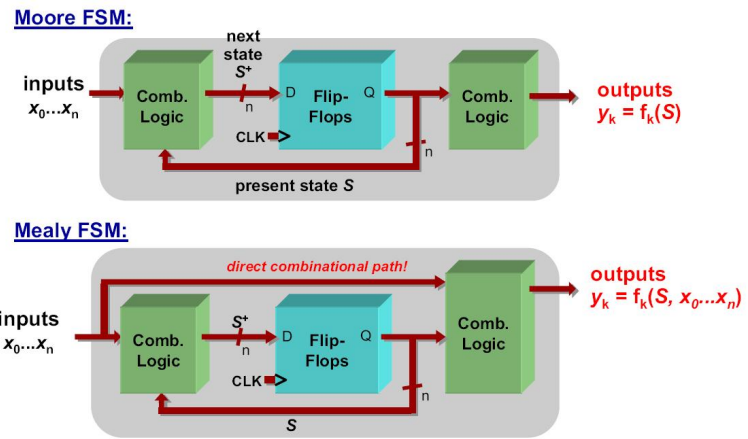
3.a SORRENDI HÁLÓZATOK - BEVEZETŐ

- Két elnevezés: sorrendi vagy szekvenciális hálózatok.
- Digitális **áramkörök memóriával** (információ-tároló képesség).
- A **jelenlegi kimenetek függenek a jelenlegi bementektől is, de az előzményektől is** (korábbi időszakokban alkalmazott bemenetek, amelyek meghatározzák a tárolt információt - az állapotot).
- A bemeneten történő változás hatására a hálózattól függően a belső változókban (állapot) is változás jön létre. A belső változók a visszacsatolás révén visszahatnak, így később is kihatnak a működésre.
- **Logikai automaták**: egyes sorrendi hálózatokat így neveznek, mert automatikus vezérlést végeznek.
- Angol elnevezés a logikai automatákra: (finite) state machine.
- Rendszerint a **memória kicsi** - néhány bit, mivel *n* bittel 2^n (sok!) állapot kódolható, tehát bonyolult viselkedésű hálózat építhető.

33

3.b SORRENDI HÁLÓZATOK - SZERKEZET

Két fajta szerkezet használatos a sorrendi hálózatok szerkesztésénél:



A Moore-féle hálózat rendszerint egyszerűbb, a Mealy féle viszont általában gyorsabb.

34

3.c SORRENDI HÁLÓZATOK - ÓRAJEL

- Az **állapotváltozás pillanatát** rendszerint órajellel (CLK) szinkronizáljuk.
- A szinkronizáló jel a memória-elemekre van kötve.
- A szinkronizáció **nem kötelező**, **de** a mai digitális berendezések nagy része szinkronizált hálózatokkal működik.
- Szinkronizációval sok **hazárdjelenség kiküszöbölhető**.
- A szinkronizált hálózatok működése jobban **áttekinthető**.
- A Mealy-féle automata hajlamosabb hazárdra, mert a kimeneti kombinációs hálózat bemeneteire közvetlenül rá vannak kötve a bemeneti jelek.

35

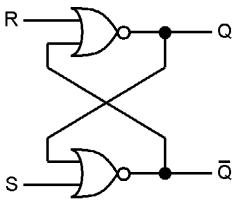
3.1 ELEMI MEMÓRIÁK

- A **sorrendi hálózatok megépítéséhez** elemi memóriák szükségesek.
- Maguk is elemi, nem szinkronizált sorrendi hálózatok.
- Információ rögzítése **pozitív visszacsatolással** - addig tartják az információt, amíg fennáll a tápfeszültség.
- Az információ **beírása** történhet a **szinkronizáló jel** megfelelő **szintjénél** (latch-ekre jellemző) vagy **élénél** (flip-flop-okra jellemző).

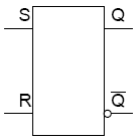
36

3.1.1.a LATCH-EK - SR LATCH

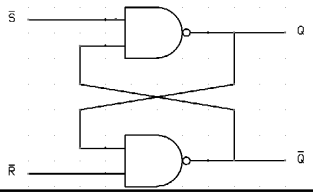
- **Szintvezérléssel** működő elemi memóriák.
- SR latch **NEM-VAGY (NOR) kapukkal**



S	R	Q	$\overline{Q}$
0	0	latch	latch
0	1	0	1
1	0	1	0
1	1	0	0



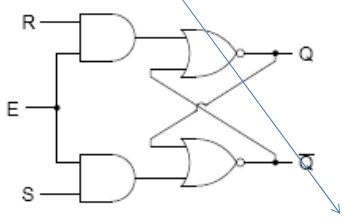
- A **két egyes** megszünte után nem lehet tudni, hogy melyik állapot fog kialakulni - ezért tilos ilyen vezérlést adni.
- Hasonló megoldás **NEM-ÉS (NAND) kapukkal**, vezérlés logikai nullával.
- Itt az okoz határozatlan viselkedést, ha mindkét bemenetre egyszerre nulla kerül.



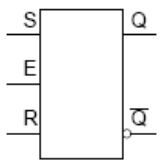
37

3.1.1.b LATCH-EK - SR LATCH SZINKRONIZÁLÁSA

- Az E (enable - engedélyez) jellel határozzuk meg, hogy mikor történjen az állapotváltozás a latch-ben.
- Előkészítjük az SR bemeneteket, majd alkalmazzuk az E jelet (szinkronizáló jel).
- Ha változás történik az SR bemeneteken E=1 mellett, az folyamatosan kihat a kimenetekre (**transzparens latch**).
- Két egyes a bemeneten (SR) itt **is határozatlan esethez** vezet.



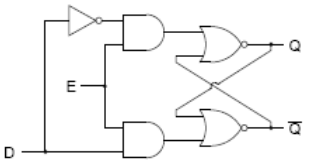
E	S	R	Q	$\overline{Q}$
0	0	0	latch	latch
0	0	1	latch	latch
0	1	0	latch	latch
0	1	1	latch	latch
1	0	0	latch	latch
1	0	1	0	1
1	1	0	1	0
1	1	1	0	0



38

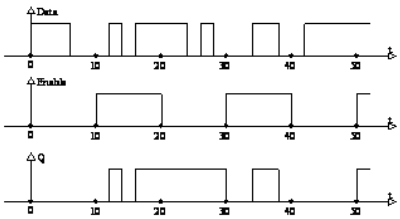
3.1.1.c LATCH-EK - D LATCH

- Az SR latch-nél jelentkező **határozatlan viselkedés** megszüntethető az ábrán bemutatott módon.
- Az így megszerkesztett latch-nek ténylegesen csak **egy (adat)bemenete van** (D) (amely a tárolt információt meghatározza).



E	D	Q	$\overline{Q}$
0	0	latch	latch
0	1	latch	latch
1	0	0	1
1	1	1	0

- A D latch is **transzparens** viselkedésű.

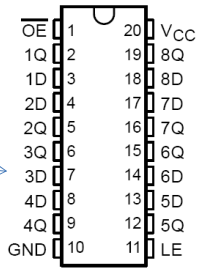


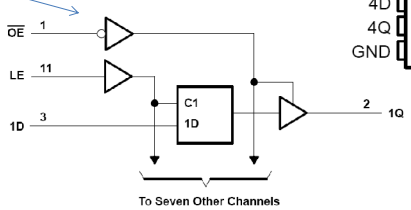
39

3.1.1.d LATCH-EK - PÉLDA

- **SN74AHCT373** - OCTAL TRANSPARENT D-TYPE LATCHES WITH 3-STATE OUTPUTS
- Viselkedési táblázat,
- lábkiosztás
- logikai diagram.

INPUTS			OUTPUT
$\overline{OE}$	LE	D	Q
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

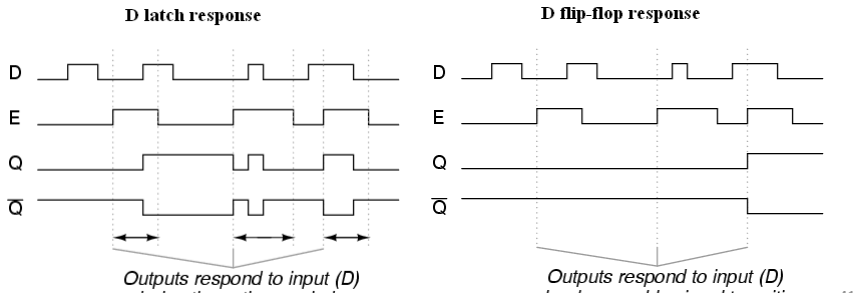




40

3.1.2.a FLIP-FLOP-OK

- Ezek is **elemi memóriák**.
- Szintvezérlés helyett **élvezérlés**.
- Állapotváltozás (adat beírása) az órajel felfutó (pozitív) vagy lefutó (negatív) élénél.
- **Különbségek** a D latch és a D flip-flop között:

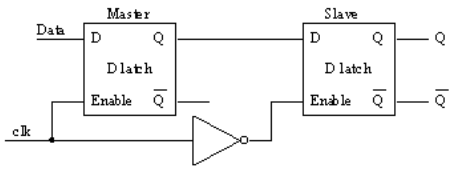


41

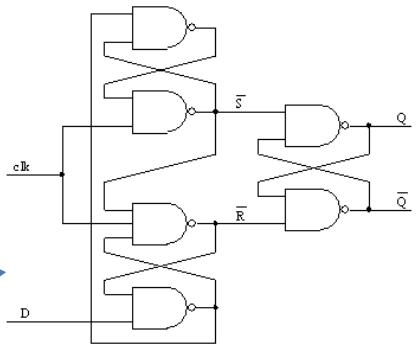
3.1.2.a FLIP-FLOP-OK - ÉLVEZÉRLÉS MEGVALÓSÍTÁSA

- D flip-flop megvalósításának módjai:

1. **Master-slave**



2. **Igazi élvezérlés**



- A vezérlési táblázat mindkét esetben azonos

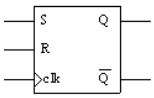
D	Qn	Qn+1
0	0	0
0	1	0
1	0	1
1	1	1

42

3.1.2.b FLIP-FLOP-OK - MÁS TÍPUSOK

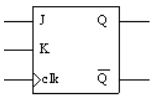
- Különbségek a vezérlési táblázatokban.

- SR flip-flop
  $Q_{n+1} = S + \overline{R}Q_n$



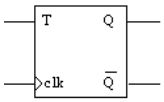
S	R	Qn	Qn+1
0	0	Q	Q
0	1	Q	0
1	0	Q	1
1	1	Q	nem def.

- JK flip-flop
  $Q_{n+1} = J \cdot \overline{Q}_n + \overline{K} \cdot Q_n$



J	K	Qn	Qn+1
0	0	Q	Q
0	1	Q	0
1	0	Q	1
1	1	Q	Q̄

- T flip-flop
  $Q_{n+1} = \overline{T} \cdot Q_n + T \cdot \overline{Q}_n$



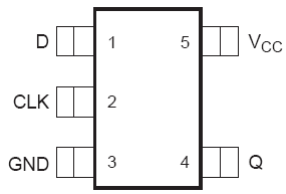
T	Qn	Qn+1
0	Q	Q
1	Q	Q̄

Minden sorrendi hálózat megépíthető minden típusú flip-flop-pal. Ma leginkább csak D flip-flop-okat alkalmaznak, különösen a VLSI technikában. A korábbi tarkaság oka, hogy adott típusú logikai automaták megvalósítása egyszerűbb volt bizonyos típusú flip-flop-okkal.

43

3.1.2.c FLIP-FLOP-OK - PÉLDA

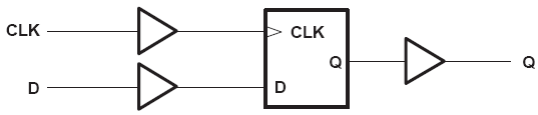
- **SN74AUP1G79** LOW-POWER SINGLE POSITIVE-EDGE-TRIGGERED D-TYPE FLIP-FLOP
- A kimenetek egyes IC-knél három állapotúak vagy invertáltak.
- A harmadik állapot megvalósításához megfelelő vezérlőbemenet áll rendelkezésre.



FUNCTION TABLE

INPUTS		OUTPUT
CLK	D	Q
↑	H	H
↑	L	L
L or H	X	Q_0

LOGIC DIAGRAM (POSITIVE LOGIC)



44

3.2 LOGIKAI AUTOMATÁK LEÍRÁSA ÉS SZERKESZTÉSE

- Általában valami automatizációs feladat megoldására gondolunk, de ugyanezekkel a módszerekkel szerkeszthető bármilyen sorrendi hálózat.

Az eljárás:

1. **Szóbeli leírásból** indulunk ki.
2. Létrehozzuk az **állapotgráfot** vagy a vele egyenértékű **állapottáblázatot**.
3. A táblázatból nyert **logikai egyenletek** alapján megszerkesztjük a kimeneti és a bemeneti kombinációs hálózatot.
4. A **kombinációs hálózatokat összekapcsoljuk a megfelelő számú és típusú flip-flop-pal**.
5. Máris **kész a logikai automata**!

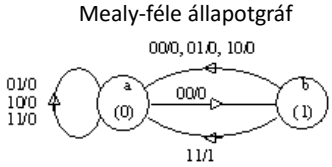
45

3.2.1 AZ ÁLLAPOTGRÁF

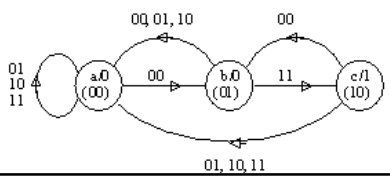
- Precíz módszer a sorrendi hálózat (logikai automata) leírására.
- **Feladat**:

  Az A és B bementi vonalakat kell figyelni. Ha először két logikai nulla, majd két logikai egyes jelenik meg, a kimenetet egy órajel ciklusra magas logikai szintre kell emelni. Minden más esetben a kimenet legyen alacsony logikai szinten.
- **Jelölések a Mealy-féle automatánál**: a körben az állapot jele, zárójelben az állapot kódja, a nyíl mellett a bemeneti kombináció/kimeneti kombináció.
- **Jelölések a Moore-féle automatánál**: a körben az állapot jele/kimenet, zárójelben az állapot kódja, a nyíl mellett a bemeneti kombináció.

Mealy-féle állapotgráf



Moore-féle állapotgráf



46

3.2.2 AZ ÁLLAPOTTÁBLÁZAT

- **Az állapottáblázat ugyanazt az információt tartalmazza mint az állapotgráf**, de alkalmasabb a kombinációs hálózatok egyenleteinek felírására.
- A táblázatban szerepelnie kell **minden egyes állapotnak és bemeneti variációnak**.
- Minden egyes esetre meg kell adni a **következő állapotot és a kimenet értékét**.

Mealy-féle állapottáblázat

Jelenlegi állapot	Következő állapot BA= 00	01	10	11	Kimenet Y BA= 00	01	10	11
a (0)	b	a	a	a	0	0	0	0
b (1)	a	a	a	a	0	0	0	1

Moore-féle állapottáblázat

Jelenlegi állapot	Következő állapot BA= 00	01	10	11	Kimenet Y
a (00)	b	a	a	a	0
b (01)	a	a	a	c	0
c (10)	b	a	a	a	1

47

3.2.3 AZ ÁLLAPOTOK KÓDOLÁSA

- **n darab flip-flop - ≤2^n állapot** kódolható.
- **Régebben** (hagyományos tervezés): minimális számú flip-flop a cél.
- **Manapság** (tervezés PLD-vel): nem cél a minimális számú flip-flop, **gyakran minden állapotra külön flip-flop-ot** alkalmaznak.
- Nem mindegy, **hogyan kódoljuk** az egyes állapotokat (kihat a kombinációs hálózatok összetettségére), de nincs szisztematikus módszer az optimalizációra.
- **Bármely típusú flip-flop-ot** használhatunk (D, SR, JK, T), de nem lehet előre tudni, melyik ad egyszerűbb megoldást.

48

3.2.4.a A FLIP-FLOP-OK VEZÉRLÉSI EGYENLETEI

- Az adott típusú flip-flop vezérlési táblázatából meg lehet határozni, hogy **mit kell a bemenetre hozni** ahhoz, hogy a kimenet a kívánt módon reagáljon.
- A tervező feladata, hogy olyan bemeneti kombinációs hálózatot hozzon létre, amely a flip-flop-okat a megfelelő módon vezérli.
- Egyes esetekben ez **konkrét logikai szintek** létrehozását jelenti (D flip-flopnál mindig így van), de vannak esetek, amikor egyes flip-flop bemenetek **tetszőleges értékek** lehetnek (pl. az SR flip-flop logikai nullát ad, ha már addig is nullát adott, akár reszeteljük, akár nem, tehát R tetszőleges, annyi csak a fontos, hogy ne legyen S=1).

49

3.2.4.b A FLIP-FLOP-OK VEZÉRLÉSI EGYENLETEI - KONKRÉT PÉLDA

- A korábbi logikai automata **Mealy**-féle megvalósítása esetén a **vezérlési táblázat és az egyeletek**:

$$D = \overline{Q}\,\overline{B}A$$

Q_n	BA	$Q_{n+1}=D$	Y
0	00	1	0
	01	0	0
	10	0	0
	11	0	0
1	00	0	0
	01	0	0
	10	0	0
	11	0	1

- **Moore**-féle megvalósítás **táblázata** ugyanarra az automatára és az **egyenletek**:

$$D1 = \overline{Q}\,\overline{1}Q0BA$$
$$D0 = Q\,\overline{1}Q\,\overline{0}\,\overline{B}\,\overline{A} + Q1\overline{Q}\,\overline{0}\,\overline{B}\,\overline{A} = \overline{Q}\,\overline{0}\,\overline{B}\,\overline{A}$$

$Q1_nQ0_n$	BA	$Q1_{n+1}=D1,\ Q0_{n+1}=D0$	Y
00	00	01	0
	01	00	0
	10	00	0
	11	00	0
01	00	00	0
	01	00	0
	10	00	0
	11	10	0
10	00	01	1
	01	00	1
	10	00	1
	11	00	1
11	00	00	0
	01	00	0
	10	00	0
	11	00	0

50

3.2.5 A KIMENETEK KÉPZÉSE

- **Mealy**-féle megvalósítás: a **kimeneti kombinációs hálózat** a bemenetek és a pillanatnyi állapotok alapján képezi a kimeneteket. A konkrét példánál:

$$Y = QBA$$

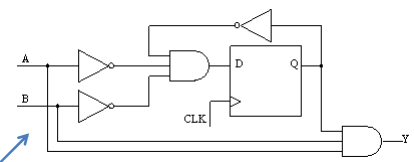
- **Moore**-féle megvalósítás: a kimenetek csak a jelenlegi állapotoktól függenek, ezeket kell a **kimeneti kombinációs hálózat** bemeneteire vezetnünk. A konkrét példánál:

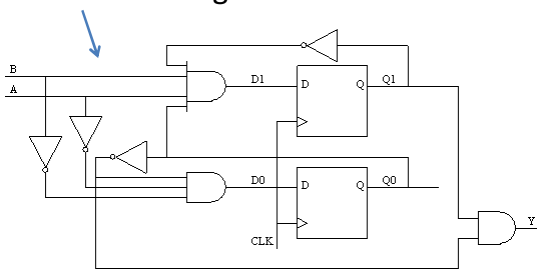
$$Y = Q1\overline{Q}\,\overline{0}$$

51

3.2.6.a A TELJES HÁLÓZAT KIALAKÍTÁSA

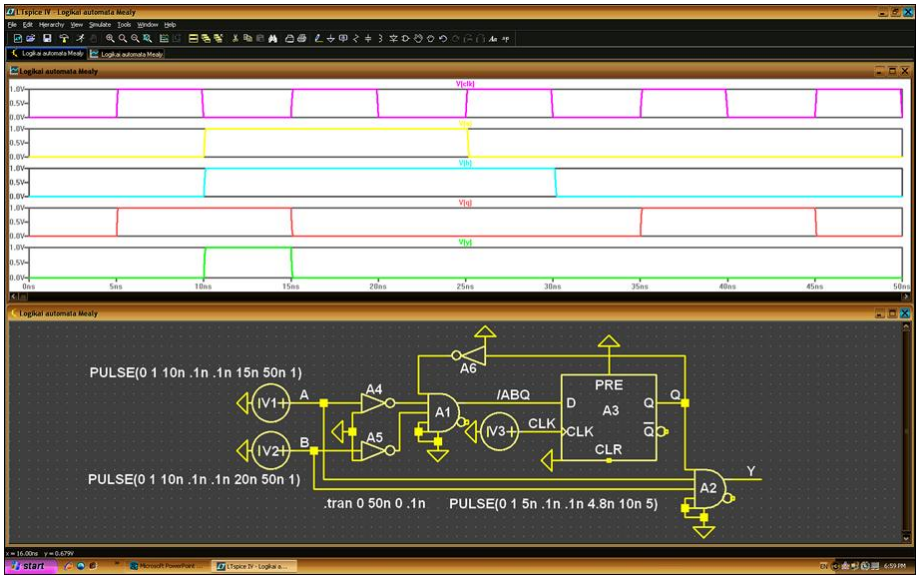
- A megvalósított sorrendi hálózatok (automaták) tartalmazzák a bemeneti kombinációs hálózatot, a szüséges számú és típusú flip-flop-ot és a kimeneti kombináiós hálózatot.
- **Mealy**-féle automata logikai rajza.
- **Moore**-féle automata logikai rajza.



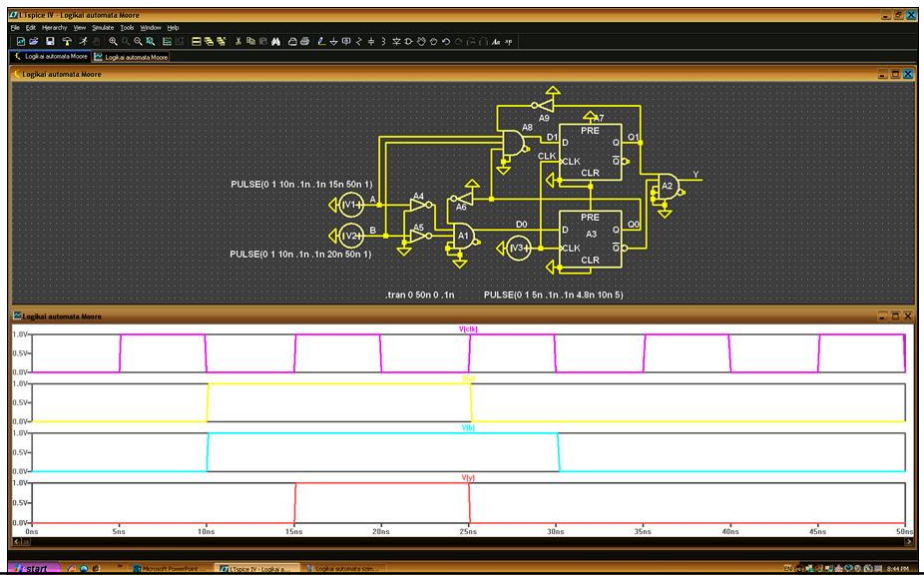


52

3.2.7.a A MEGÉPÍTETT HÁLÓZAT MŰKÖDÉSE - **MEALY**-FÉLE AUTOMATA



3.2.7.b A MEGÉPÍTETT HÁLÓZAT MŰKÖDÉSE - **MOORE**-FÉLE AUTOMATA



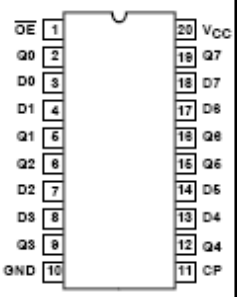
3.3 REGISZTEREK

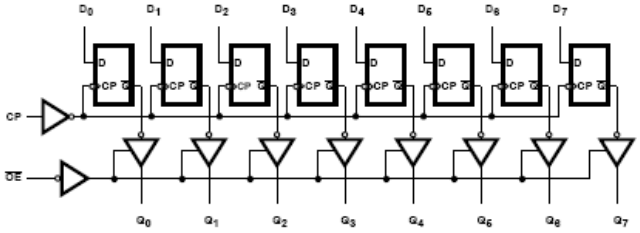
- **Kis mennyiségű információ** (néhány bit) ideiglenes **tárolására** alkalmasak.
- Szerkezet: **latch-ek vagy flip-flop-ok** felsorakoztatva, egy tokozásban, közös vezérléssel.
- **Típusok**:
  1. Közönséges (stacionáris) regiszterek
  2. Léptető (shift) regiszterek
  3. Gyűrűs regiszterek (gyűrűs számlálók)

55

3.3.1 KÖZÖNSÉGES (STACIONÁRIS) REGISZTEREK

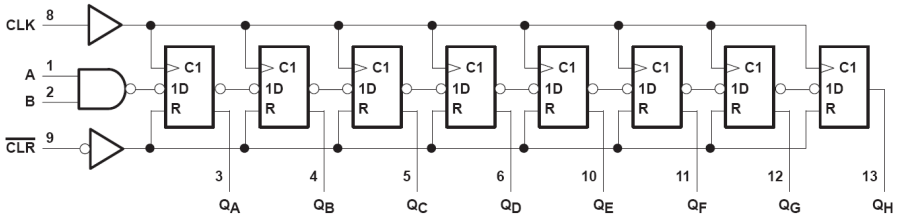
- Rendszerint **közös órajel** (CLK) vagy engedélyező jel (LE).
- **Párhuzamos beírás és kiolvasás** (minden bit egyszerre).
- A kimenetek lehetnek három állapotúak vagy invertáltak.
- Bitek száma 2...32. Több IC összekapcsolásával bővíthető.
- Példa: **CD54HC374** High-Speed CMOS Logic Octal D-Type Flip-Flop, 3-State Positive-Edge Triggered



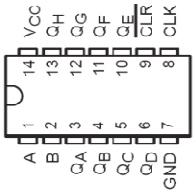


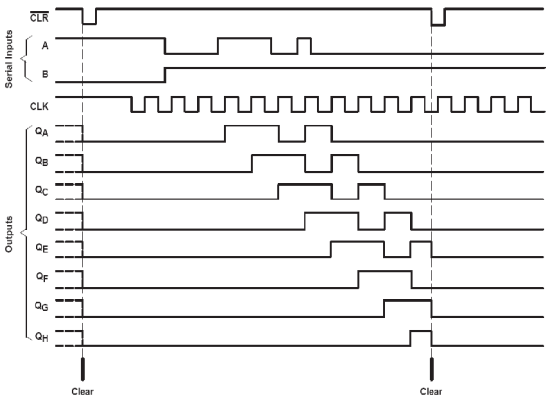
56

3.3.2 LÉPTETŐ (SHIFT) REGISZTEREK



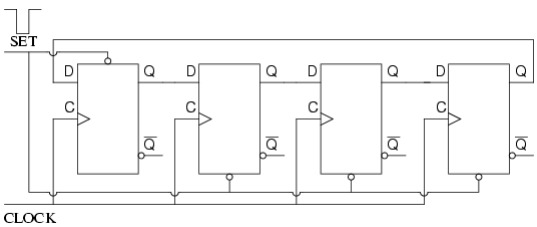
- Az elemi tárolók (flip-flop-ok) tartalma **egyikből a másikba** íródik át.
- **Egy bemenet, egy kimenet**, de lehet párhuzamos beírás és kiolvasás is.
- Példa: **SN74HC164** 8-bit parallel-out serial shift register: közös törlés, párhuzamos vagy soros kiolvasás.

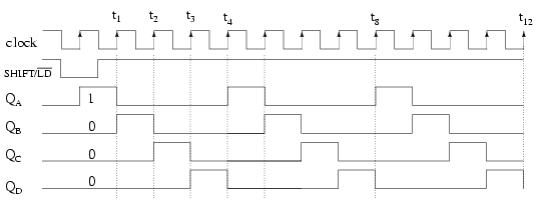




3.3.3.a GYŰRŰS REGISZTEREK (GYŰRŰS SZÁMLÁLÓK)

- **Visszacsatolás** a kimenetről a bemenetre.
- **n számú órajel ciklus** alatt csinál a tartalom egy kört.
- Valahogy "**be kell indítani**": a SET jel egy flip-flop-ba egyest ír, a többibe nullát (lehet más kombináció is).

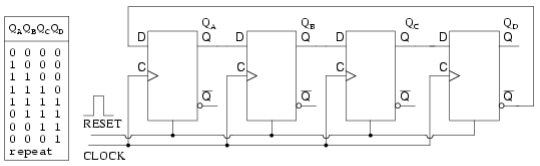




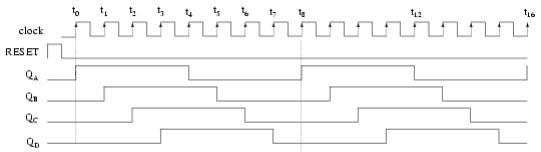
58

3.3.3.b GYŰRŰS REGISZTEREK (GYŰRŰS SZÁMLÁLÓK) - JOHNSON-FÉLE SZÁMLÁLÓ

- Az utolsó flip-flop invertált kimenete van **visszacsatolva** az első flip-flop-ra.
- A tartalom az órajel **2n számú ciklusa** alatt csinál egy kört (n a flip-flop-ok száma).
- Nem szükséges kezdeti beállítás.



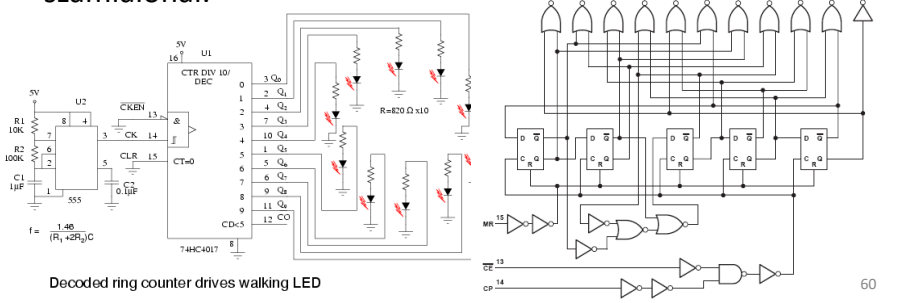
Johnson counter (note the $\overline{Q}_D$ to D_A feedback connection)



Four stage Johnson counter waveforms

3.3.3.c GYŰRŰS REGISZTEREK (GYŰRŰS SZÁMLÁLÓK) - PÉLDA

- **74HC4017** - DECADE COUNTER/DIVIDER WITH TEN DECODED OUTPUTS
- Szerkezetileg Johnson- féle számláló, de a **kimenetek dekódolva vannak**, úgy viselkednek mint egy gyűrűs számlálónál.



Decoded ring counter drives walking LED

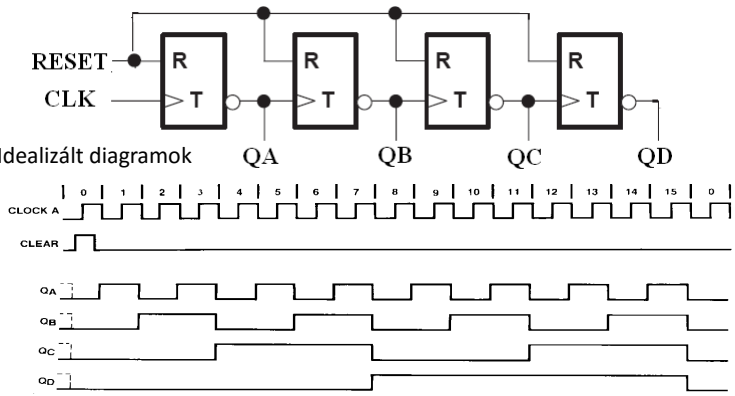
3.4 SZÁMLÁLÓK

- Olyan regiszterek, amelyek **előre meghatározott állapotokon haladnak keresztül** (természetes bináris kód vagy más).
- A haladás (léptetés) az **órajel** hatására történik.
- Az új tartalom beírását az egyes flip-flop-okba megfelelő **kombinációs hálózat** végzi.
- Rendszerint nem szükséges kimeneti kombinációs hálózat: a **flip-flop-ok kimenetei képezik a számláló kimeneteit**.
- Az állapotok számát a számláló **modulusának** nevezzük. A modulus lehet 2^n (bináris számlálók) vagy annál kisebb (decimális és más számlálók).
- **Aszinkron** (soros) számlálók: az egyes flip-flop-ok állapotváltozása nem pontosan egyidőben történik.
- **Szinkron** (párhuzamos) számlálók: az állapotváltozások egyidőben történnek, köszönve a közös órajelnek.

61

3.4.1.a ASZINKRON (SOROS) SZÁMLÁLÓK

- Rendszerint **T flip-flop-okból** épül fel.
- Egyszerű kaszkád kötés, **nincs sem bemeneti- sem kimeneti kombinációs hálózat**.
- Az **órajelet** csak az **első flip-flop-ra** vezetjük, a többiek egymástól kapják a vezérlést.
- A flip-flop-ok **állapotváltozásai késne**k a bemeneti órajelhez képest - rövid ideig, az órajel élét követően, a kimenetek nem érvényesek.

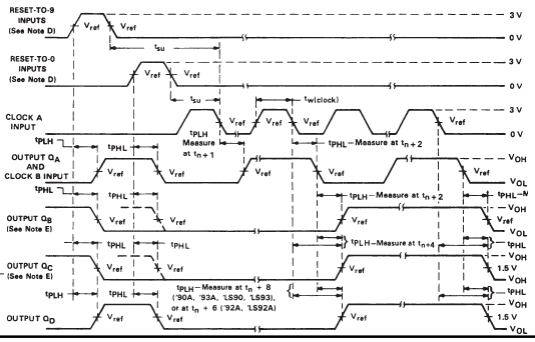


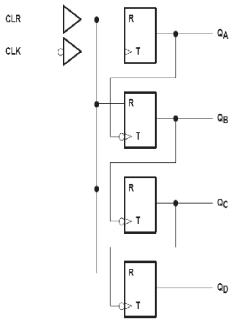
COUNT	OUTPUTS			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H

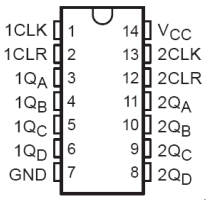
62

3.4.1.b ASZINKRON (SOROS) SZÁMLÁLÓK - PÉLDA

- Példa: **SN74HC393** Dual four-bit asynchronous binary counters.
- Modulus: 2^4=16.
- A diagramon láthatók a **késések**.







63

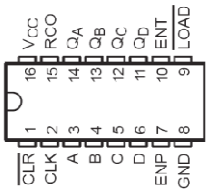
3.4.2.a SZINKRON (PÁRHUZAMOS) SZÁMLÁLÓK

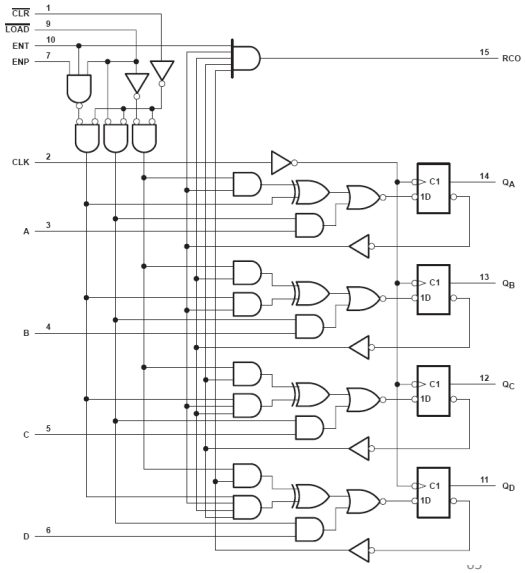
- Az egyes flip-flop-ok **ugyanazt az órajelet** kapják, így egyidőben írják be az új tartalmat (magasabb frekvenciás órajel mellett is szabályos a működés).
- **Szükséges bemeneti kombinációs hálózat** az új tartalom (állapot) előkészítésére.
- A bemeneti kombinációs hálózatot a **logikai automaták szintézisénél** bemutatott módszerrel végzik.
- Kimeneti kombinációs hálózat általában nincs.
- Modulus ≤2^n.

64

3.4.2.b SZINKRON (PÁRHUZAMOS) SZÁMLÁLÓK - PÉLDA **BINÁRIS** SZÁMLÁLÓRA

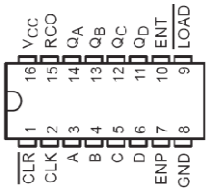
- **SN74ALS161B** - SYNCHRONOUS 4-BIT BINARY COUNTER
- Párhuzamos beírási lehetőség (A,B,C,D, LOAD)
- Minden flip-flop törlése egyszerre (CLR).
- ENT, ENP, RCO - kaszkád kötéshez szükséges (modulus növelés).

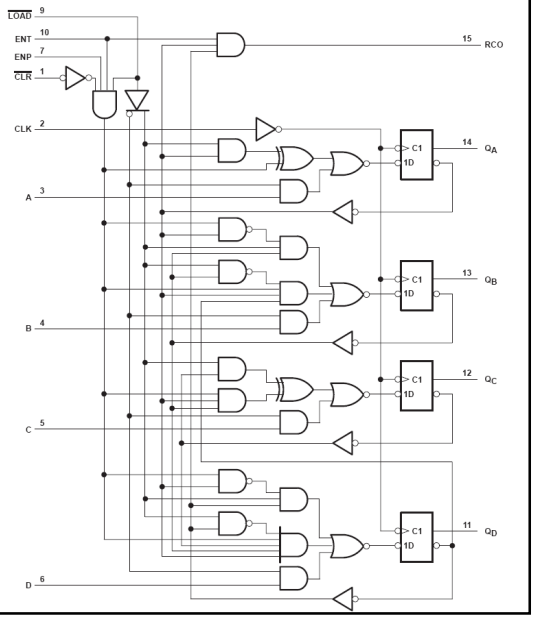




3.4.2.c SZINKRON (PÁRHUZAMOS) SZÁMLÁLÓK - PÉLDA **DECIMÁLIS** SZÁMLÁLÓRA

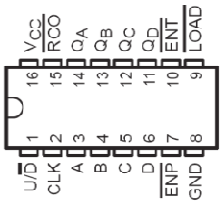
- **SN74ALS162B** - SYNCHRONOUS 4-BIT DECIMAL COUNTER
- Párhuzamos beírási lehetőség (A,B,C,D, LOAD).
- Minden flip-flop törlése egyszerre (CLR).
- ENT, ENP, RCO - kaszkád kötéshez szükséges (modulus növelés).

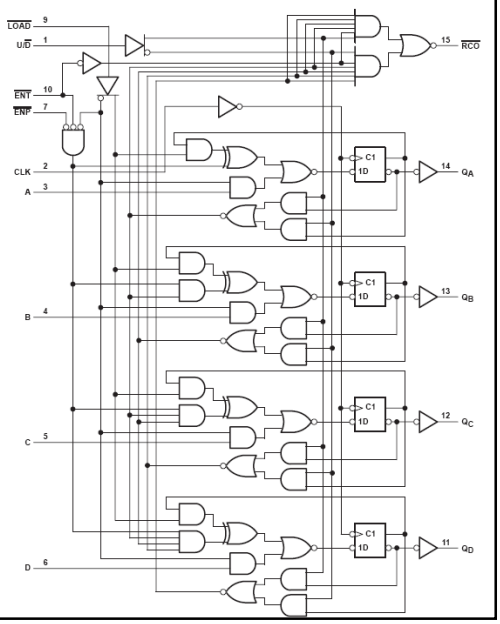




3.4.2.d SZINKRON (PÁRHUZAMOS) SZÁMLÁLÓK - **ELŐRE/HÁTRA** SZÁMLÁLÓ

- **SN74ALS169B** - SYNCHRONOUS 4-BIT UP/DOWN BINARY COUNTER
- U/D - a számlálás irányát határozza meg
- Párhuzamos beírási lehetőség (A,B,C,D, LOAD)
- ENT, ENP, RCO - kaszkád kötéshez szükséges (modulus növelés).





4. VEGYES HÁLÓZATOK

Két lehetőség:

1. Digitális áramkörök, amelyek tartalmaznak kombinációs és sorrendi elemeket is. A hangsúly lehet egyik vagy másik részen.
2. Digitális és analóg áramkörök együtt, egy tokozásban.

Felosztás:

1. Memória áramkörök
2. Aritmetikai egységek
3. D/A átalakítók
4. A/D átalakítók

68

4.1. MEMÓRIÁK

- Nagyobb mennyiségű adat tartós vagy ideiglenes tárolására alkalmasak.
- A nagy kapacitás sok tárolóhelyet és jó szervezést feltételez.
- Vannak más tárolási módszerek is (optikai, mágneses...), de mi itt csak a félvezető alapú megoldásokkal foglalkozunk.
- Memória áramkör tömbvázlata:

C - címvonalak

V - vezérlő bemenetek

A - adatvonalak

69

4.1.1.a A MEMÓRIÁK FELOSZTÁSA ÉS JELLEMZŐIK

Felosztási elvek:

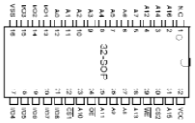
- **Gyors** vagy **lassú** hozzáférés (jelentős kölönbség lehet a beírás- vagy kiolvasás sebessége és gyakorisága között ugyanannál a memóriánál),
- **Statikus** (flip-flop-okban történő-) vagy **dinamikus** (parazita kapacitásokban történő) tárolás,
- Hozzáférés az adatokhoz **sorban** vagy **tetszőleges sorrendben**,
- **Egybites** vagy **többites** adatok,
- Gyártástechnológia: **CMOS** vagy **bipoláris**.

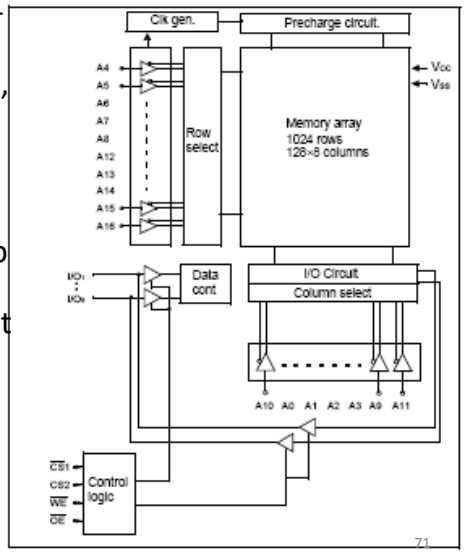
70

4.1.1.b A MEMÓRIÁK FELOSZTÁSA ÉS JELLEMZŐIK - KERESKEDELMI TÍPUSOK

RAM (random access memory) -

- Tetszőleges sorrendben lehet beírni és kiolvasni az adatokat, operatív memóriaként használják.
- Lehet statikus (SRAM) vagy dinamikus (DRAM) tárolás. A dinamikus megoldás kevesebb alkatrészt igényel cellánként.
- Példa: **K6T1008V2C** 128Kx8 bit Low Power and Low Voltage CMOS Static RAM

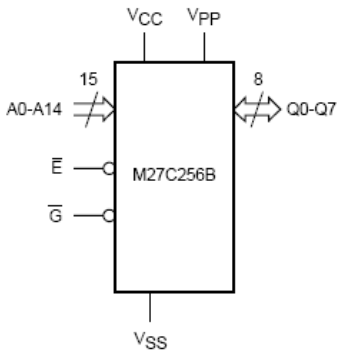




4.1.1.c A MEMÓRIÁK FELOSZTÁSA ÉS JELLEMZŐIK - KERESKEDELMI TÍPUSOK

ROM (read only memory)

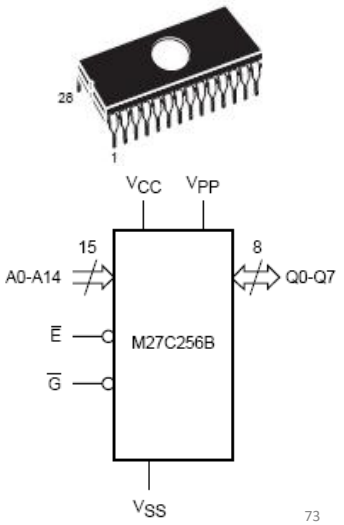
- az egyszer beírt tartalmat nem lehet megváltoztatni,
- gyorsan és tetszőleges számszor kiolvasható,
- szerkezetileg kombinációs hálózat (kódátalakító),
- OTP ROM - a felhasználó programozza,
- mask programmable ROM - gyártó programozza
- nem alkalmas fejlesztési munkákhoz
- Példa: **M27C256B** 256 Kbit (32Kb x 8) UV EPROM and OTP EPROM



4.1.1.d A MEMÓRIÁK FELOSZTÁSA ÉS JELLEMZŐIK - KERESKEDELMI TÍPUSOK

EPROM - electrically programmable ROM

- elektromos jelekkel programozható - MOSFET gate-jére villamos töltést juttatunk - nem tud távozni a szigetelt térből,
- törlés ibolyántúli fénnyel - viszonylag lassú eljárás,
- **ablak** van a tokozás tetején.
- Példa: **M27C256B** 256 Kbit (32Kb x 8) UV EPROM and OTP EPROM

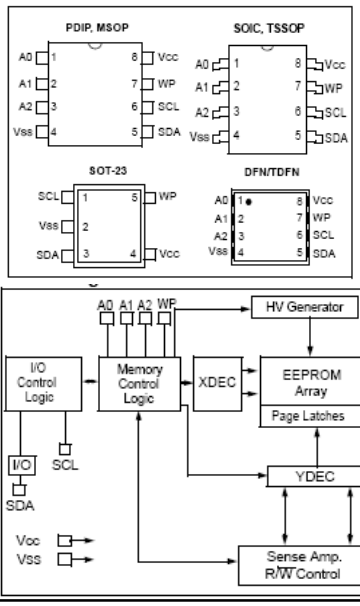


73

4.1.1.e A MEMÓRIÁK FELOSZTÁSA ÉS JELLEMZŐIK - KERESKEDELMI TÍPUSOK

EEPROM - electrically erasable PROM

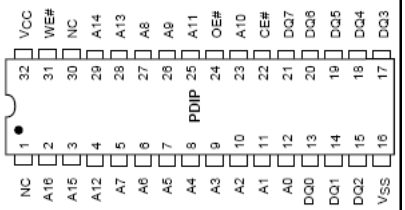
- törlés és újraprogramozás villamos jelekkel nem kell ibolyántúli fény - nincs ablak
- olvasás gyorsan, törlés és újraírás lassabban
- gyakori a soros írás-olvasás (kis lábszámú tokozásban elfér nagy kapacitás)
- Példa: **24AA32A/24LC32A** 32K I2C™ Serial EEPROM

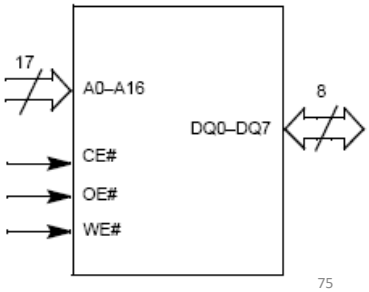


4.1.1.f A MEMÓRIÁK FELOSZTÁSA ÉS JELLEMZŐIK - KERESKEDELMI TÍPUSOK

flash EEPROM (flash memória)

- viszonylag gyors törlés és újraírás villamos jelekkel,
- adatok megőrzése - tápfeszültség nélkül is megőrzik a tartalmat,
- az újraírások száma korlátozott (pl. 100,000),
- törlés rendszerint csak szektoronként lehetséges,
- **Am29F010** 1 Megabit (128 K x 8-bit) CMOS 5.0 Volt-only, Uniform Sector Flash Memory

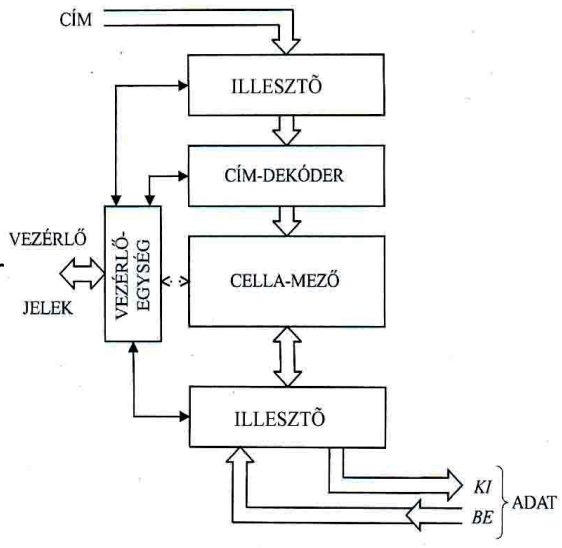




75

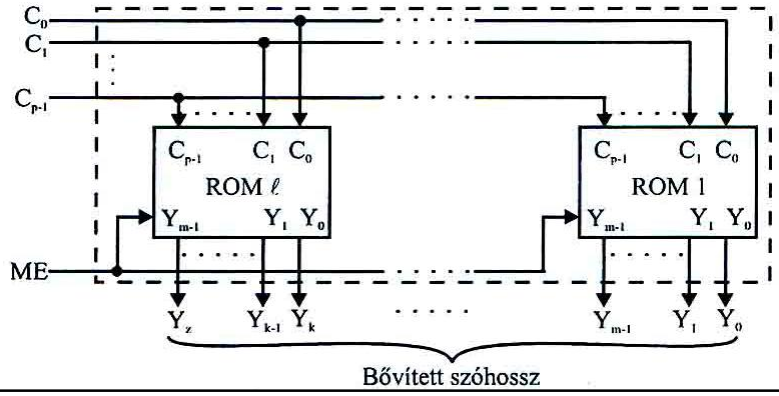
4.1.2 A MEMÓRIA ÁRAMKÖRÖK BELSŐ SZERKEZETE

- Nagy számú adat - hatékony szervezést igényel.
- Központi rész: cella mező, ez tárolja az adatokat.
- Dekóder: kiválasztja a megfelelő cellát.
- Rendszerint két dekóder (oszlop és sor) a dekóder egyszerűsítése végett.
- Adat kimenet/bemenet azonos vonalakon, kétirányú illesztővel
- Vezérlőjelek : WE, OE, CS



4.1.3.a A KAPACITÁS BŐVÍTÉSE

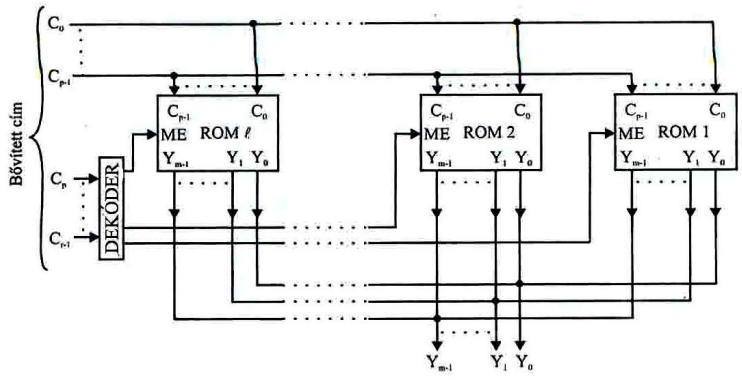
- Milyen lehetőségek vannak, ha az egy tokozásban beszerezhető memória kapacitása nem elegendő?
- **Szóhossz bővítése**: közös címvonalak, adatok párhuzamosan.



77

4.1.3.b A KAPACITÁS BŐVÍTÉSE

- Milyen lehetőségek vannak, ha az egy tokozásban beszerezhető memória kapacitása nem elegendő?
- A címezhető **szavak számának növelése**: dekóderrel választjuk a megfelelő tokozást, a kimenetek/bemenetek közös adatsinre csatlakoznak.



78

4.2 ARITMETIKAI EGYSÉGEK

Ezek a funkciók inkább a mikrovezérlők és FPGA eszközök belső egységeiként fordulnak elő, de van egy-két ilyen MSI áramkör is.

A funkciók:

- Összeadók
- Szorzók
- Aritmetikai komparátorok
- Párosság ellenőrzésére szolgáló áramkörök

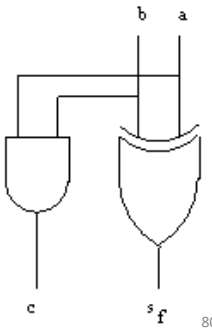
79

4.2.1.a ÖSSZEADÓ ÁRAMKÖRÖK

- Az alap a **félösszeadó**: két bitet ad össze, meghatározza az összeget és az átvitelt a magasabb helyi értékre.
- Nem alkalmas kaszkád kötésre (több bit összeadása)
- Az egyenletek:

$$s_f = \overline{a}b + a\overline{b} = a \oplus b$$
$$c = ab$$

a	b	c	s_f
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

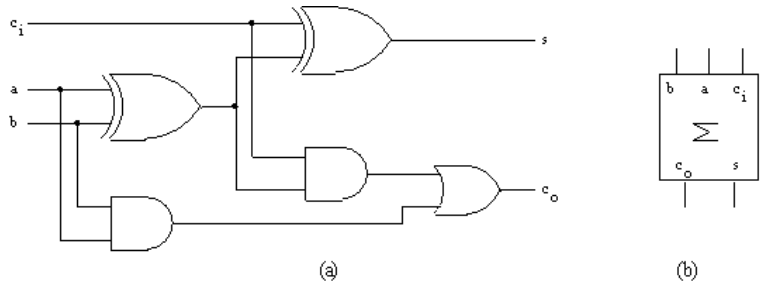


80

4.2.1.b ÖSSZEADÓ ÁRAMKÖRÖK

- A **teljes összeadó** alkalmas kaszkád kötésre: fogadja az átvitelt a kisebb helyi értékről.
- Az egyenletek:

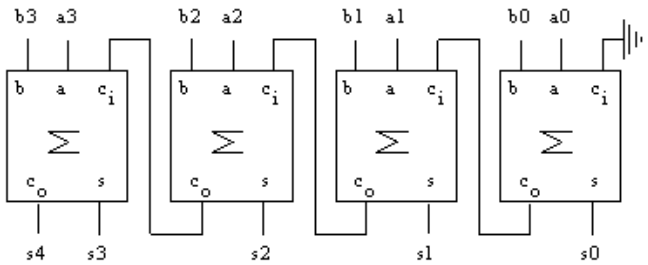
$$s = a \oplus b \oplus c_i$$
$$c_o = (a \oplus b)c_i + ab$$



81

4.2.1.c ÖSSZEADÓ ÁRAMKÖRÖK

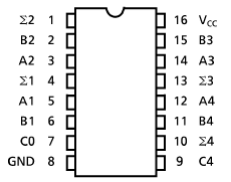
- Több bites számok összeadása.
- Egybites összeadók kaszkád kötése.
- Az átvitel párhuzamos kiértékelésével gyorsítható az összeadás (külön hálózat szükséges).

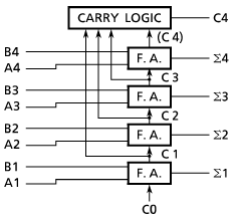


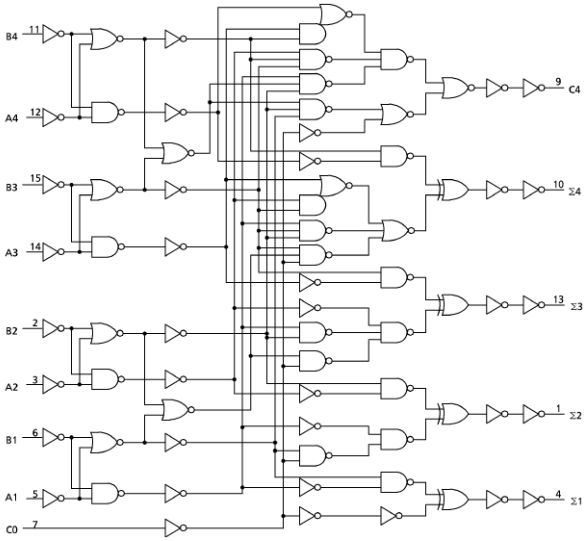
82

4.2.1.d ÖSSZEADÓ ÁRAMKÖRÖK

- Példa: **CD54/74ACT283** 4-Bit Binary Full Adder With Fast Carry

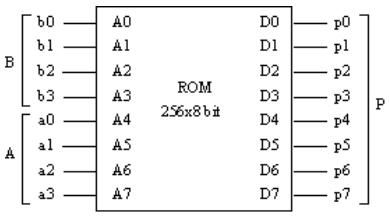


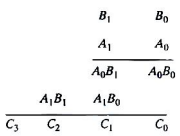


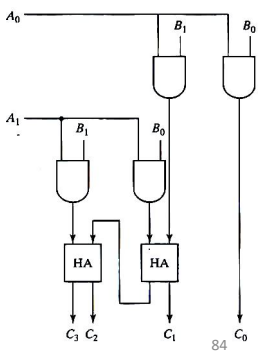


4.2.2.a SZORZÓ ÁRAMKÖRÖK

- Tisztán **kombinációs hálózattal** - táblázat szerint - minden egyes bemeneti értékkombinációra kiértékeli a kimenetet. A bitek számával a bonyolultság meredeken nől (célszerű **ROM**-ot használni).
- Szintén kombinációs hálózattal: bitenkénti **szorzás ÉS kapukkal**, a részeredmények **összeadása teljes összeadókkal**.



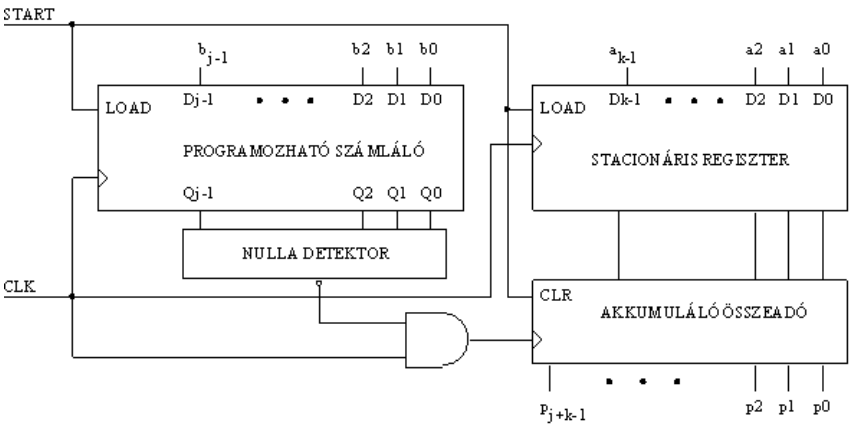




4.2.2.b SZORZÓ ÁRAMKÖRÖK

Sorrendi hálózattal:

1. Annyiszor adjuk össze a szorzandót, amennyi a szorzó számértéke.

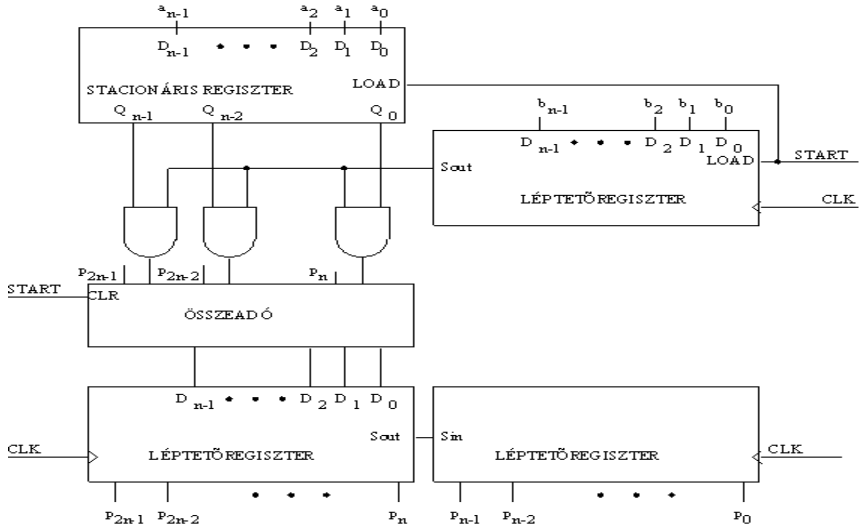


85

4.2.2.c SZORZÓ ÁRAMKÖRÖK

Sorrendi hálózattal:

2. Bitenkénti szorzás és a részeredmények összeadása.



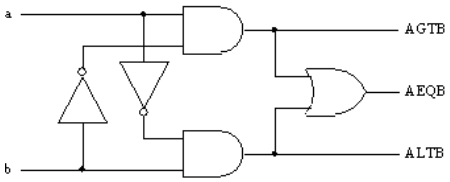
4.2.3.a ARITMETIKAI (DIGITÁLIS) KOMPARÁTOR

- Bináris számok összehasonlítása.
- Az eredemény: kisebb, nagyobb, egyenlő.
- Egy bit esetére a kifejezések a következők:

$$AGTB = a\bar{b}$$
$$AEQB = ab + \bar{a}\bar{b} = \bar{a} \overline{\oplus} \bar{b}$$
$$ALTB = \bar{a}b$$

- Kombinációs táblázat és logikai rajz:

A	B	AGTB	AEQB	ALTB
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

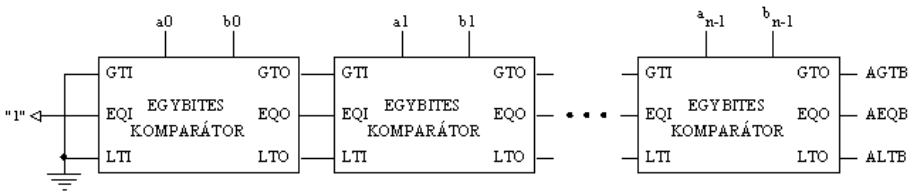


87

4.2.3.b ARITMETIKAI (DIGITÁLIS) KOMPARÁTOR

Több-bites számok összehasonlítása

1. Egybites komparátorok kaszkád kötése:



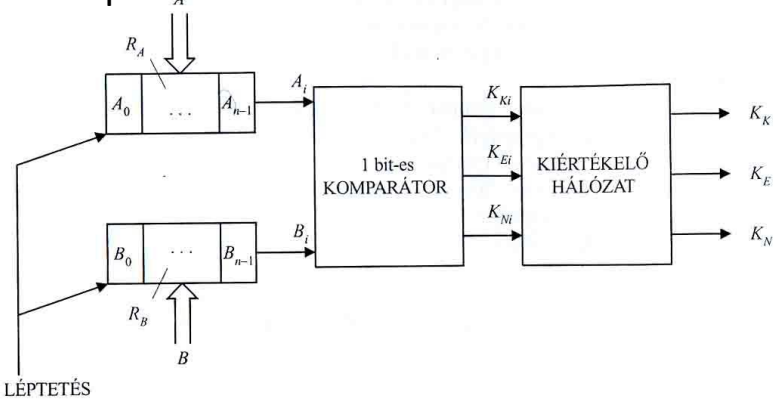
- Biztosítani kell a kimenenteket és a bemeneteket a kaszkád kötéshez.
- Egyszerű, de viszonylag lassú a soros átvitel miatt.

88

4.2.3.c ARITMETIKAI (DIGITÁLIS) KOMPARÁTOR

Több-bites számok komparálása

2. Komparálás bitenként:

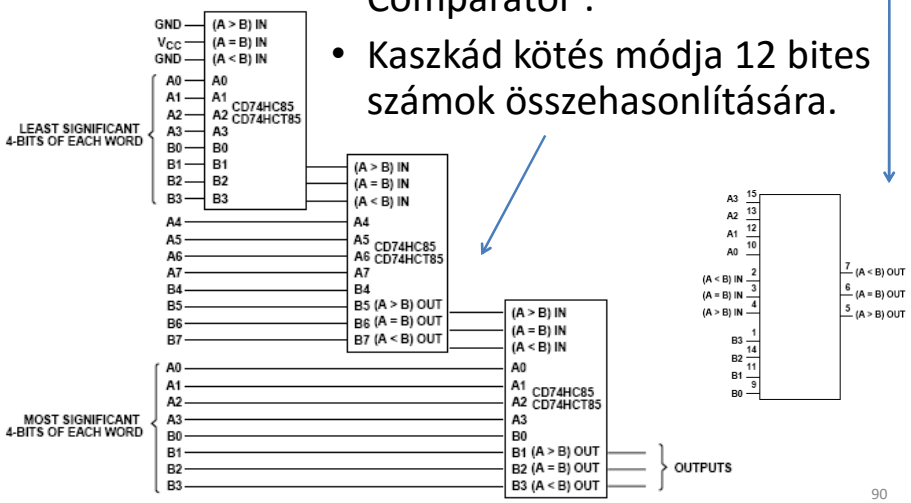


- Az összehasonlítást célszerű a nagyobb helyi értékű bitek felől kezdeni.

89

4.2.3.d ARITMETIKAI (DIGITÁLIS) KOMPARÁTOR

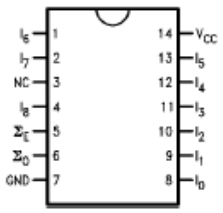
- Példa: **CD74HCT85** High-Speed CMOS Logic 4-Bit Magnitude Comparator .
- Kaszkád kötés módja 12 bites számok összehasonlítására.



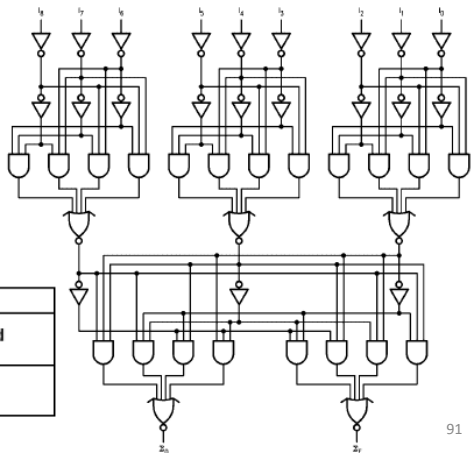
90

4.2.4 PÁROSSÁGOT ELLENŐRZŐ ÁRAMKÖRÖK

- Megállapítja, hogy páros- vagy páratlan számú bit van-e logikai egyesen.
- Példa: **74F280** 9-Bit Parity Generator/Checker



Number of HIGH Inputs I_0–I_8	Outputs	
	Σ Even	Σ Odd
0, 2, 4, 6, 8	H	L
1, 3, 5, 7, 9	L	H



91

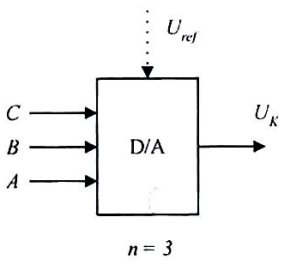
4.3 D/A ÁTALAKÍTÓK

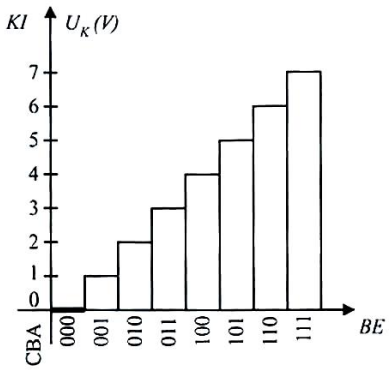
- Számokból analóg jeleket (feszültségértéket) képeznek.
- A feszültségérték rendszerint arányos a számértékkel.
- A kapott feszültségértékek diszkrét skálán helyezkednek el.

92

4.3.1 D/A ÁTALAKÍTÓ MŰKÖDÉSI ELVE

- n - bites bemeneti szám (kód) esetén 2^n feszültségérték lehetséges a kimeneten.
- Szükséges egy alapjel (V_{REF}), a kapott feszültségértékek ezzel arányosak.

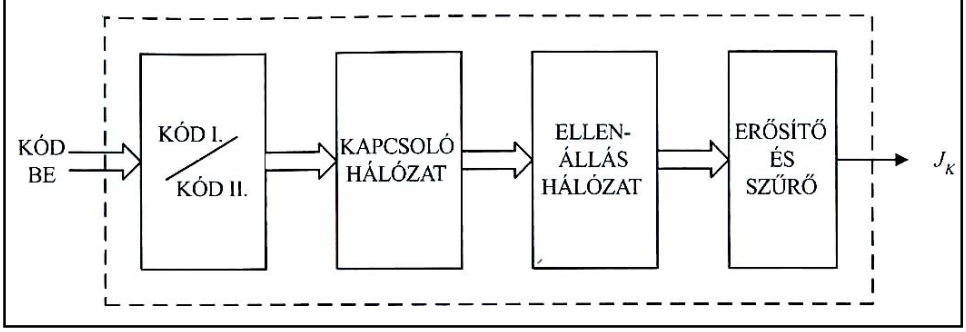




93

4.3.2.a D/A ÁTALAKÍTÓK FELÉPÍTÉSE

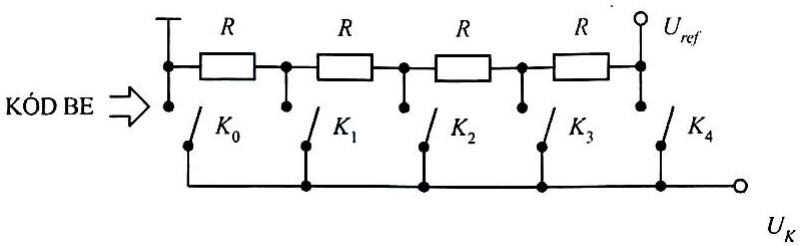
- Az átalakító fő része egy ellenálláshálózat.
- Analóg kapcsolókkal variáljuk a referens feszültség leosztását.
- Az analóg kapcsolókat a bemeneti számkód bitjei vezérlik. Egyes esetekben szükséges kódátalakítást végezni.
- A kimeneten rendszerint szükséges bizonyos analóg jelformálás (erősítés, szűrés).



4.3.2.b D/A ÁTALAKÍTÓK FELÉPÍTÉSE

Direkt típusú átalakító

- Azonos értékű ellenállások soros kötése.
- A lecsapolásokon előállítjuk a skálának megfelelő összes analóg feszültségértéket.
- Az analóg kapcsolók vezérlése rendszerint dekódert tesz szükségessé.



95

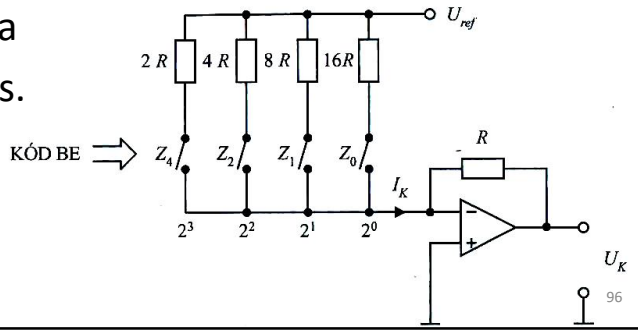
4.3.2.c D/A ÁTALAKÍTÓK FELÉPÍTÉSE

Súlyozott ellenálláshálózattal működő átalakító

- Az ellenállásértékek és a rajtuk keresztülfolyó áramok 1:2:4: . . . 2^n arányban vannak egymással.
- A kimeneti feszültség képlete:

$$V_O = -R_f V_{REF} \frac{1}{R}\left(2^0 Q_0 + 2^1 Q_1 + 2^2 Q_2 + \ldots 2^{n-1} Q_{n-1}\right)$$

- Integrálásra nem alkalmas.



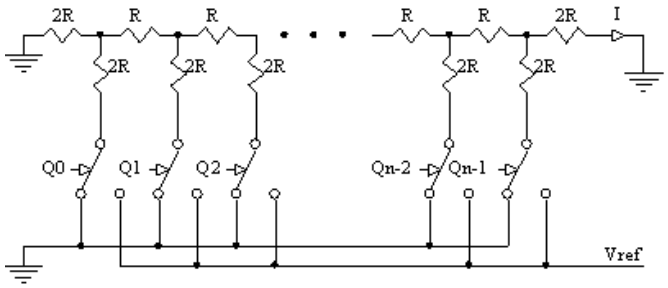
96

4.3.2.d D/A ÁTALAKÍTÓK FELÉPÍTÉSE

R-2R létrahálózattal működő átalakító

- Integrált formában általában ezeket az átalakítókat gyártják - csak két ellenállásértéket kell pontosan reprodukálni.
- A kimeneti áram (a feszültség vele arányos) képlete:

$$I = \frac{V_{REF}}{6R} \cdot \frac{1}{2^{n-1}} \left(2^{n-1} Q_{n-1} + 2^{n-2} Q_{n-2} + \ldots + 2^1 Q_1 + 2^0 Q_0 \right)$$



97

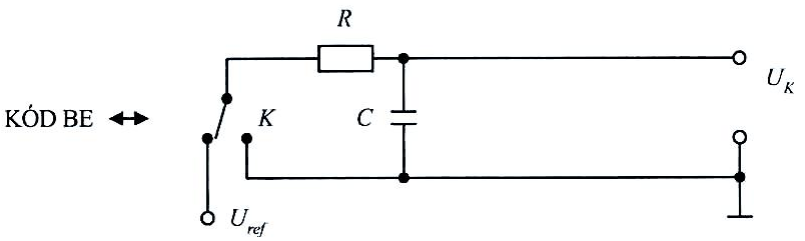
4.3.2.e D/A ÁTALAKÍTÓK FELÉPÍTÉSE

Impulzus-szélesség modulációval működő átalakító

- Segéd megoldás, kevés alkatrészt igényel.
- Analóg kimenet nélküli mikrovezérlővel is megoldható.
- A kimeneti feszültség képlete:

$$V_O = D \cdot V_{REF} \qquad D = f\left(Q_0, Q_1 \ldots Q_{n-1}\right)$$

- A szűrő késése miatt lassan működik.

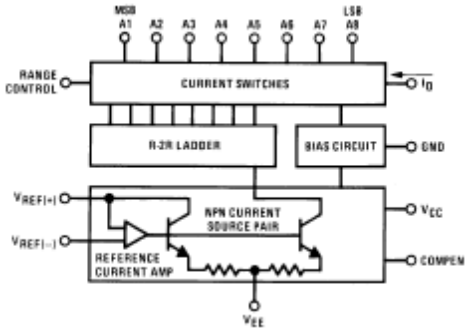


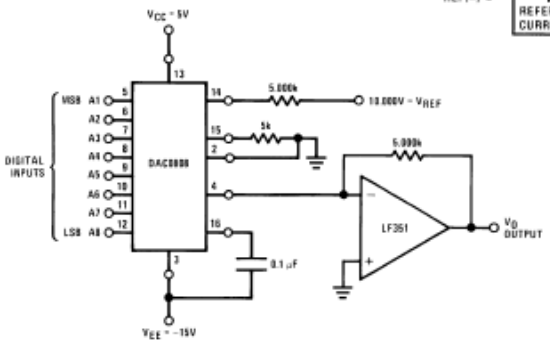
98

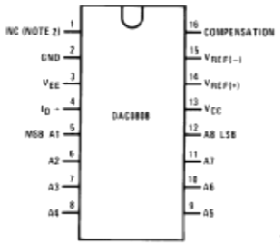
4.3.2.f D/A ÁTALAKÍTÓK FELÉPÍTÉSE

Példa: **DAC0808** 8-Bit D/A Converter.

- R-2R típusú átalakító
- A kimeneti áramot feszültséggé kell alakítani (műveleti erősítővel).







99

4.3.3 D/A ÁTALAKÍTÓK JELLEMZŐI

Felbontás

- a bemeneti bitek száma
- meghatározza a pontosságot is, mivel felelni kell az átviteli jelleggörbe monotonitásáért.

Sebesség

- az ellenállás-hálózatokkal működő átalakítók általában gyorsak, a beállási idő μs alatti (az analóg kapcsolók és a műveleti erősítő késései)
- az impulzus szélesség modulációval működő átalakítók lassúak: a beállási idő az impulzusok periódusának sokszorosa.

100

4.4 A/D ÁTALAKÍTÓK

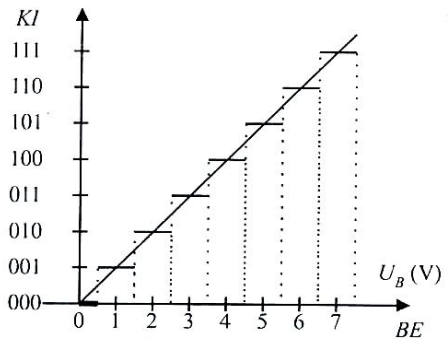
1. Analóg jel (feszültségérték) átalakítása digitális jelre (szám).
2. Célok:
   - digitális tárolás
   - digitális jelfeldolgozás
   - digitális jeltovábbítás
   - digitális megjelenítés (kijelzés)

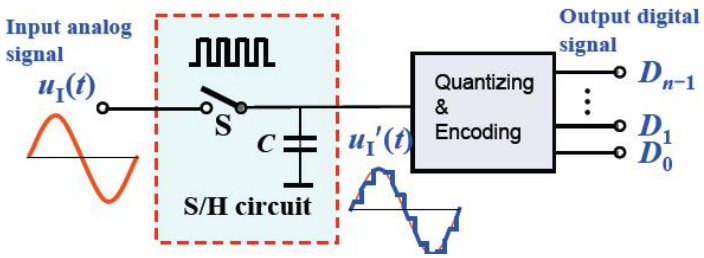
101

4.4.1 AZ A/D ÁTALAKÍTÁS ELVI MEGOLDÁSA

A megoldandó **feladatok**:

1. mintavételezés (diszkretizálás idő szerint),
2. diszkretizálás amplitúdó szerint (összehasonlítás egy diszkrét érték-skálával),
3. kódolás (minden diszkrét értékhez egy kódot rendelünk).





102

4.4.2.a AZ A/D ÁTALAKÍTÓK FELÉPÍTÉSE

Négy megoldás fordul elő a gyakorlatban:

1. **Direkt** (flash típusú) átalakítók
2. **Fokozatos közelítéses** (szukcesszív aprokszimációs - bitenkénti) átalakítók
3. **Számlálós** (integráló) megoldás.
4. **Sigma-delta** A/D konverter

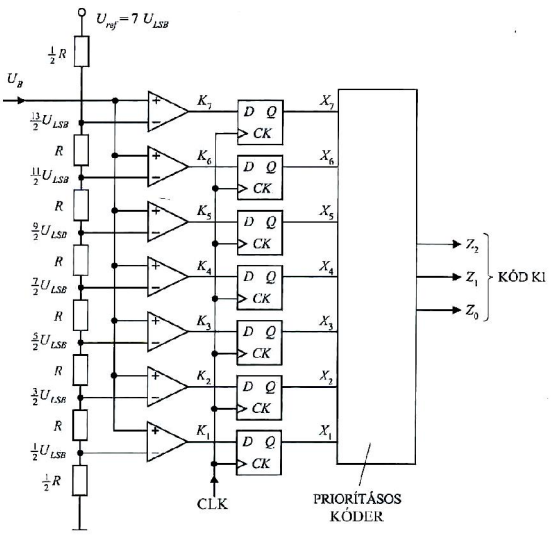
Lényeges különbségek sebességben és árban.

103

4.4.2.b AZ A/D ÁTALAKÍTÓK FELÉPÍTÉSE

1. **Direkt** (flash) típusú átalakító felépítése

- n-bites átalakítóhoz 2^n-1 komparátor végzi az amplitúdó szerinti diszkretizálást.
- Kódolás prioritásos kóderrel.
- Az órajellel történő szinkronizálás biztosítja, hogy csak érvényes kódokat olvassunk ki.
- Bonyolult, költséges hardver.
- Nagy sebesség.

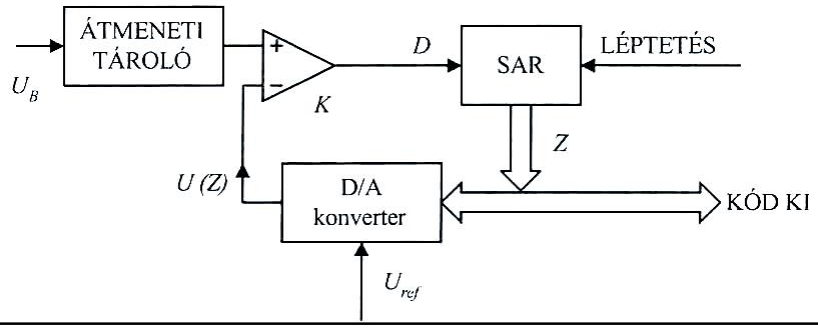


104

4.4.2.c AZ A/D ÁTALAKÍTÓK FELÉPÍTÉSE

2. **Fokozatos közelítéses** átalakító felépítése.

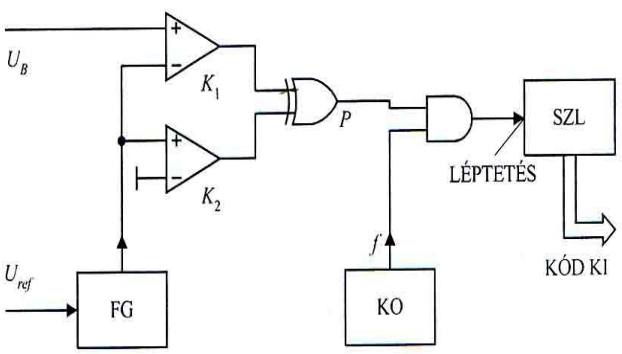
- A SAR (regiszter) tartalmát bitenkét egyesre állítjuk (kezdve a legnagyobb helyi értéktől).
- A D/A átalakító létrehozza a megfelelő analóg jelet.
- A komparátor eldönti, hogy szükséges volt-e az illető bitet egyesre állítani.



4.4.2.d AZ A/D ÁTALAKÍTÓK FELÉPÍTÉSE

3.a **Számlálós típusú** átalakító felépítése - **egyszeri integrálás**

- Az analóg jelet egy fűrészgenerátor feszültségével hasonlítjuk össze - ezzel a feszültséget időtartammá alakítjuk.
- Nagyobb analóg jel esetén a számláló arányosan tovább számol.
- Gyönge pontok: a fűrészgenerátor meredeksége nem szabad hogy változzon, úgyszintén az órajel frekvenciája nagyon stabil kell hogy legyen.

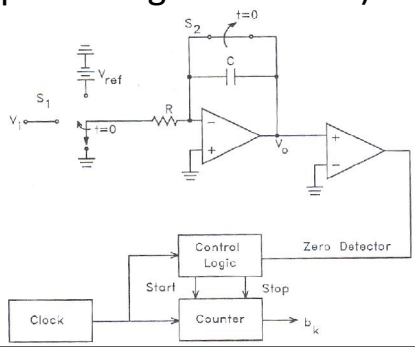


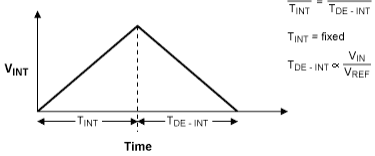
106

4.4.2.e AZ A/D ÁTALAKÍTÓK FELÉPÍTÉSE

3.b **Számlálós típusú** átalakító felépítése - **kettős integrálás**

- Csak a V_{REF} kell, hogy pontos legyen.
- Az órajel frekvenciája és az integrátor elemei elég, ha rövid távon (egy átalakítás alatt) stabilak (pontosság nem fontos).





$\frac{V_{IN}}{T_{INT}} = \frac{V_{REF}}{T_{DE-INT}}$

T_{INT} = fixed

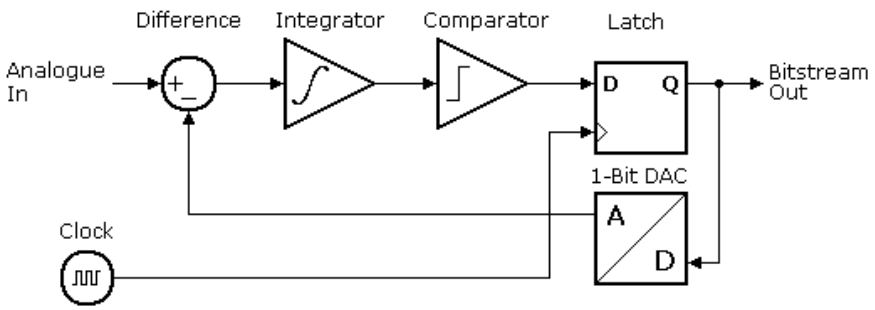
$T_{DE-INT} \propto \frac{V_{IN}}{V_{REF}}$

107

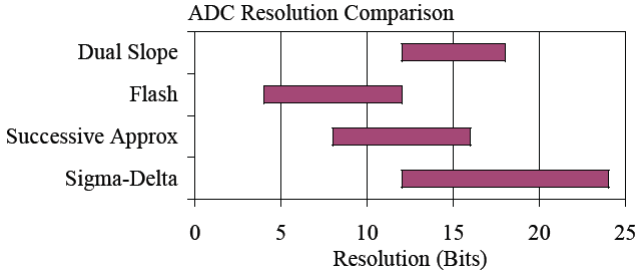
4.4.2.e AZ A/D ÁTALAKÍTÓK FELÉPÍTÉSE

4. Sigma-delta átalakító

- Az integrátor az analóg jel és a D/A átalakító kimeneti feszültségének különbségét integrálja.
- Az integrál előjelét a komparátor határozza meg.
- A komparátor kimenete beíródik a flip-flopba.
- Az időegység alatt kapott impulzusok száma a flip-flop kimenetén arányos az analóg jellel.



4.4.3 AZ A/D ÁTALAKÍTÓK ÖSSZEHASONLÍTÁSA

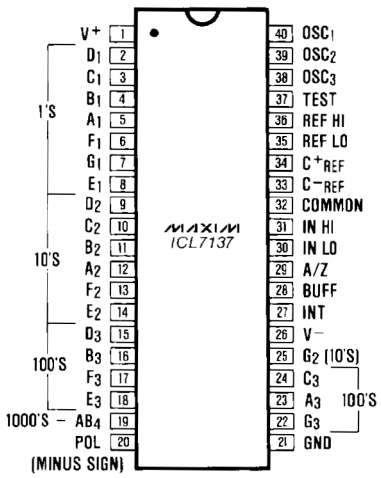


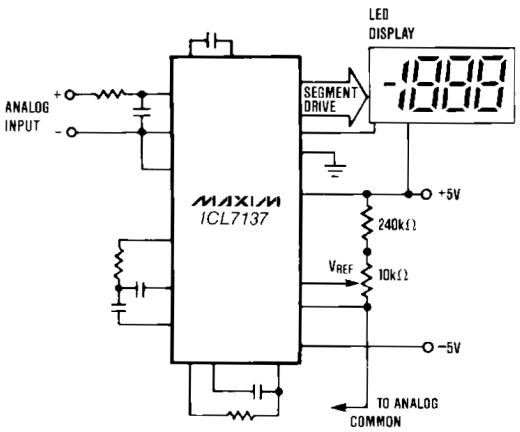
Type	Speed (relative)	Cost (relative)
Dual Slope	Slow	Med
Flash	Very Fast	High
Successive Appox	Medium – Fast	Low
Sigma-Delta	Slow	Low

109

4.4.4.a PÉLDÁK A/D ÁTALAKÍTÓKRA

- **ICL7137** 3 ½ számjegyű (decimális számok) A/D átalakító (kettős integrálású) (digitális voltmérő)

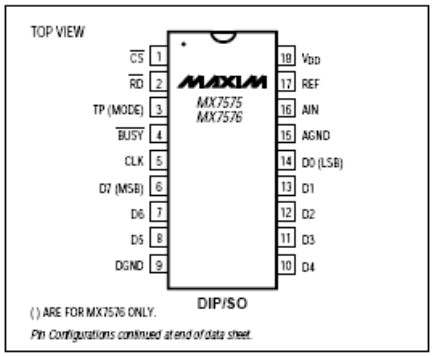


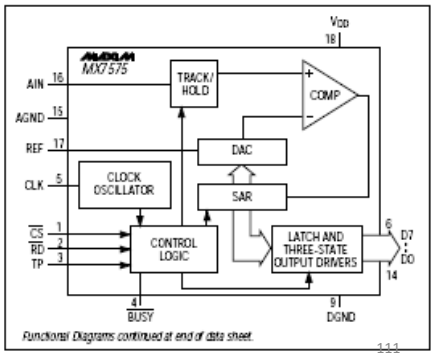


110

4.4.4.b PÉLDÁK A/D ÁTALAKÍTÓKRA

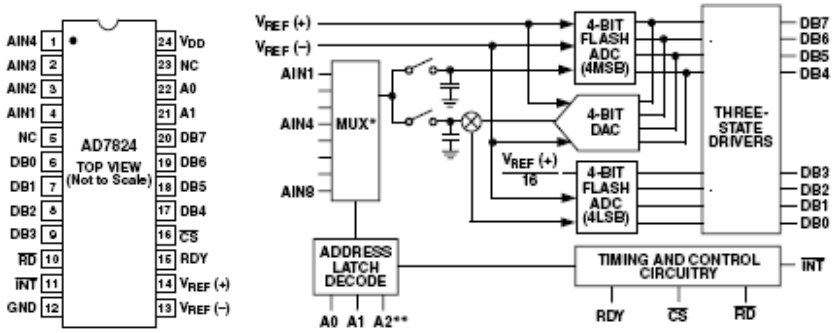
- **MX7575/MX7576** CMOS, µP-Compatible, 5µs/10µs, 8-Bit ADCs (fokozatos közelítéses)
- 5V táp, beépített mintavételezés, analóg jel határfrekvenciája 50 kHz.





4.4.4.c PÉLDÁK A/D ÁTALAKÍTÓKRA

- **AD7824/AD7828** LC^{2}MOS High Speed 4- and 8-Channel 8-Bit ADCs
- Külön négybites direkt (flash típusú) A/D konverter a felső négy bitre és másik négybites átalakító a maradékra. 2,5µs átalakítási idő.



Vége a II. résznek

Digitális tervezés SSI és MSI funkcionális egységekkel (hagyományos tervezés)

113