

# Hobbi Elektronika



		y	
		0	1
x	0	0	0
	1	0	1

		y	
		0	1
x	0	0	1
	1	1	1

		y	
		0	1
x	0	0	1
	1	1	0

		y	
		0	1
x	0	0	1
	1	1	0

Figure 1. Truth tables

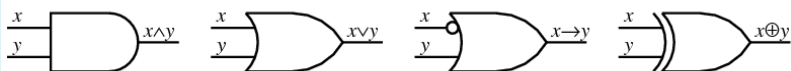


Figure 2. Logic gates

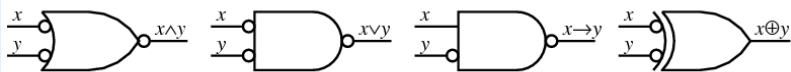


Figure 3. De Morgan equivalents



Figure 4. Venn diagrams



**A digitális elektronika alapjai:**  
**Kombinációs logikai hálózatok – 2. rész**



# Felhasznált anyagok

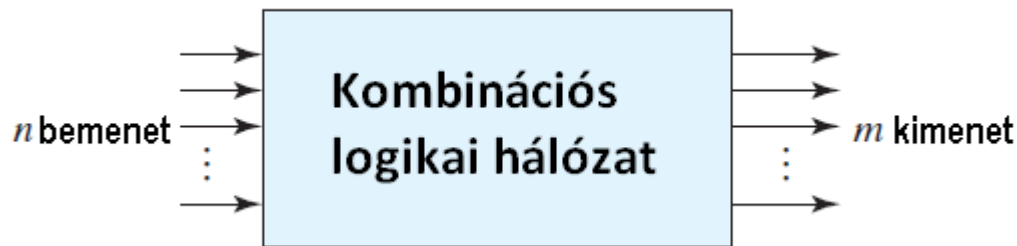
- ❑ M. Morris Mano and Michael D. Ciletti: [Digital Design - With an Introduction to the Verilog HDL, 5th. Edition](#)
- ❑ [Electronics-course.com](#) (Digital Electronics Course I. – II.)
  - [Boolean Algebra Calculator](#)
  - [Karnaugh Map Calculator](#)
- ❑ Végh János: [Ismerkedés a digitális elektronikával](#)
- ❑ Mészáros Miklós: [Logikai algebra alapjai, logikai függvények I.](#)
- ❑ BME FKE: [Logikai áramkörök](#)
- ❑ Logisim szimulátor: [www.cburch.com/logisim/](http://www.cburch.com/logisim/)
- ❑ Falstad.com: [Circuit simulator](#)
- ❑ F-alpha.net: [Boolean Algebra](#)
- ❑ Neuproductions-be: [The Logic Lab](#)



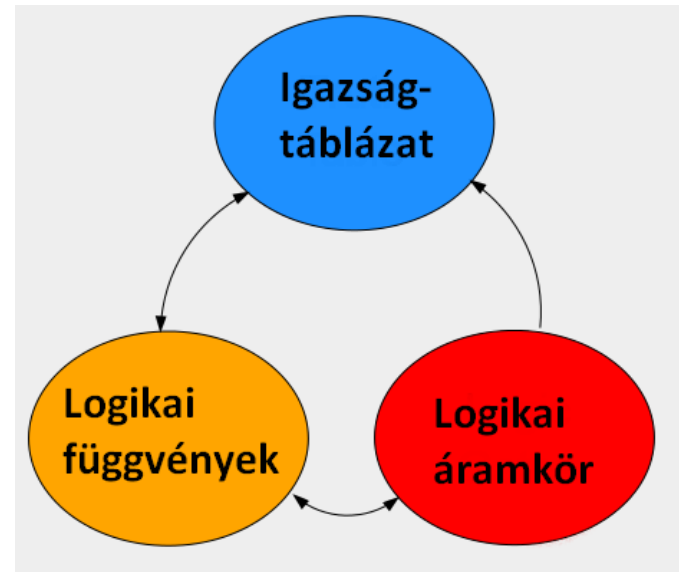
# Kombinációs logikai hálózatok

Emlékeztető:

- ❑ A **kombinációs logikai hálózat** viselkedése logikai függvényekkel leírható. Kimeneteinek állapota csupán a bemenetek pillanatnyi állapotának függvénye.
- ❑ A **kombinációs logikai hálózat** összekapcsolt logikai kapukból áll. Az  $n$  bemenetnek összesen  $2^n$  állapota lehetséges, amelyek mindegyikéhez a kimenetek egy-egy állapota rendelhető. Így a kombinációs logikai hálózat egyértelműen megadható az **igazságtáblázatával**, amely leírja a „viselkedését”.
- ❑ A kombinációs logikai hálózat kimenetei egy-egy  $n$  változós **logikai függvénnyel** is megadhatók.



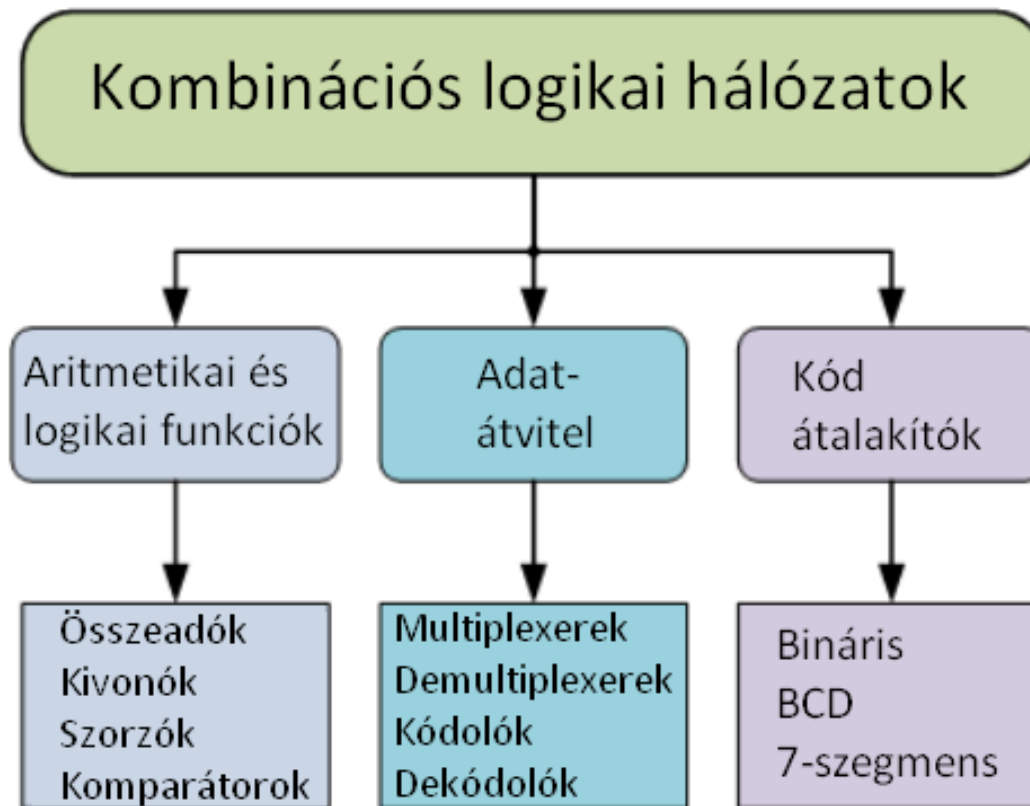
$$Y_i = F_i(x_1, x_2, \dots, x_n)$$

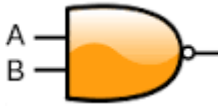




# Kombinációs logikai hálózatok csoportosítása

A kombinációs logikai hálózatok legelterjedtebben használt funkcionális moduljai az alábbiak szerint csoportosíthatók:



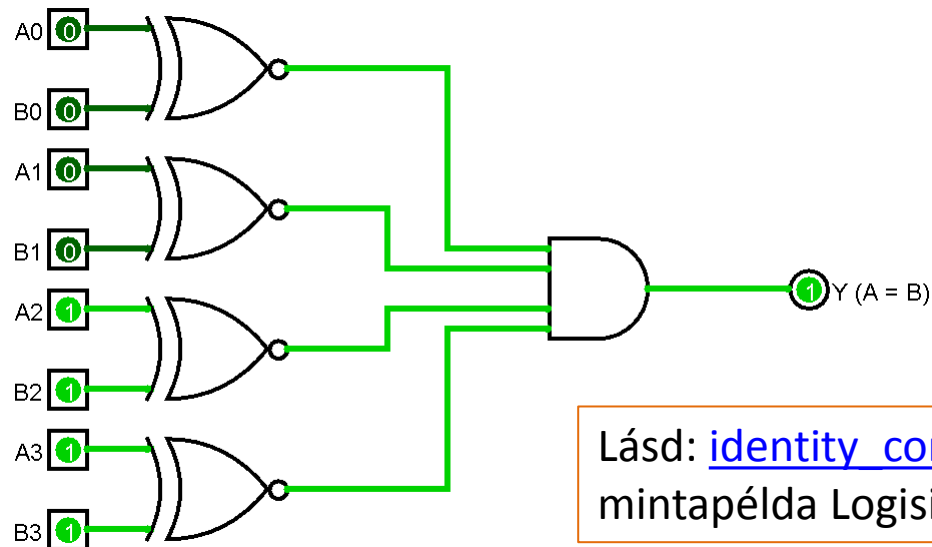


# Digitális összehasonlító áramkör

A komparátor áramkör funkciója a bemenetére adott számok összehasonlítása. Ha csak egyenlőséget jelez, akkor **egyenlőség komparátornak** (*identity comparator*) nevezzük. Ha az áramkör azt is jelzi, hogy az A szám nagyobb, vagy kisebb, mint a B szám, akkor **magnitúdó komparátornak** (*magnitude comparator*) hívjuk.

**Az egyenlőség vizsgálata** az egyszerűbb feladat, **XNOR** (ekvivalencia) kapukkal könnyen megvalósítható.

$$Y = (A_0 \oplus B_0)' \cdot (A_1 \oplus B_1)' \cdot (A_2 \oplus B_2)' \cdot (A_3 \oplus B_3)'$$

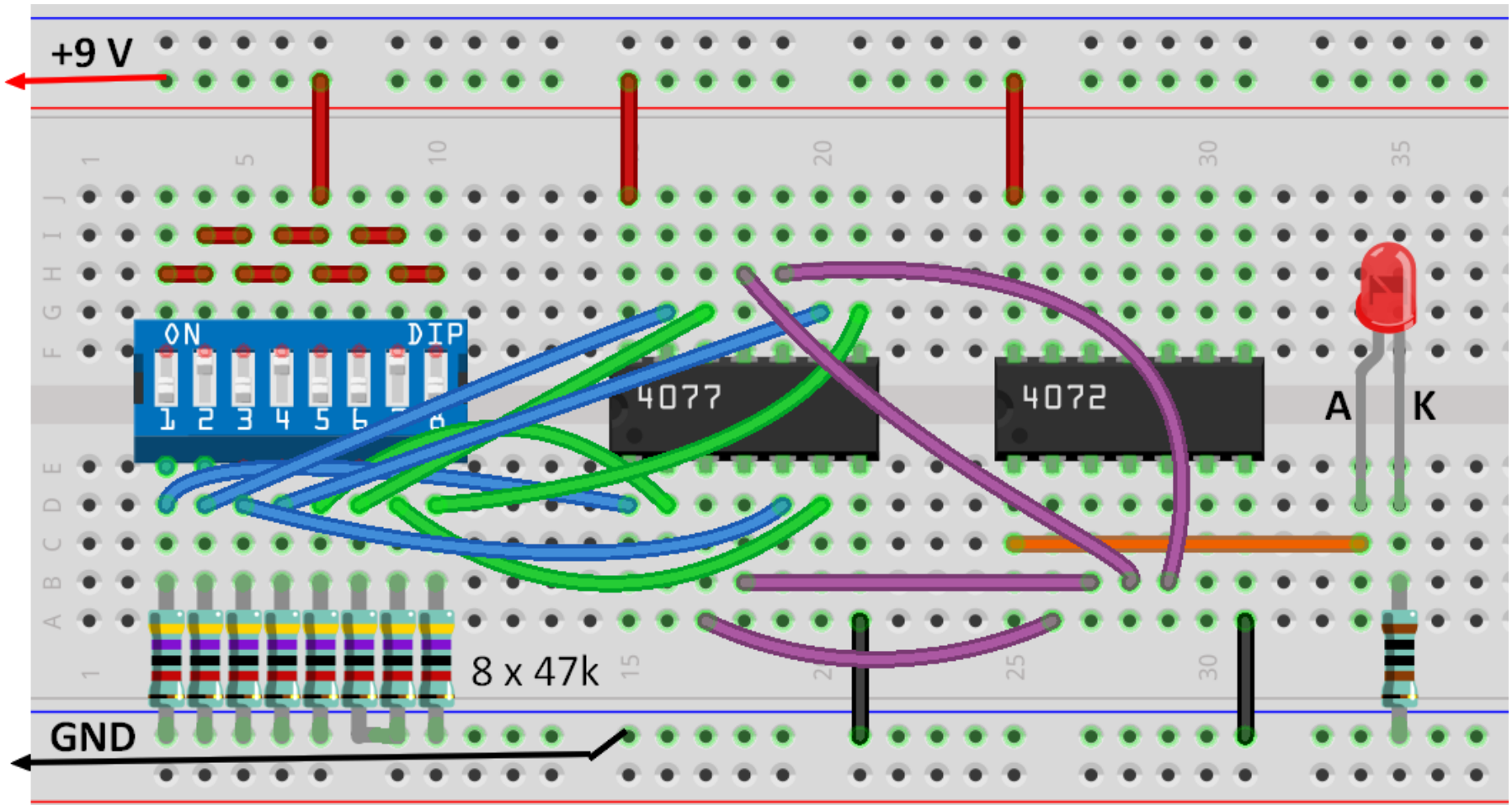


Lásd: [identity\\_comparator.circ](#)  
mintapélda Logisim-hez



# Egyenlőség komparátor

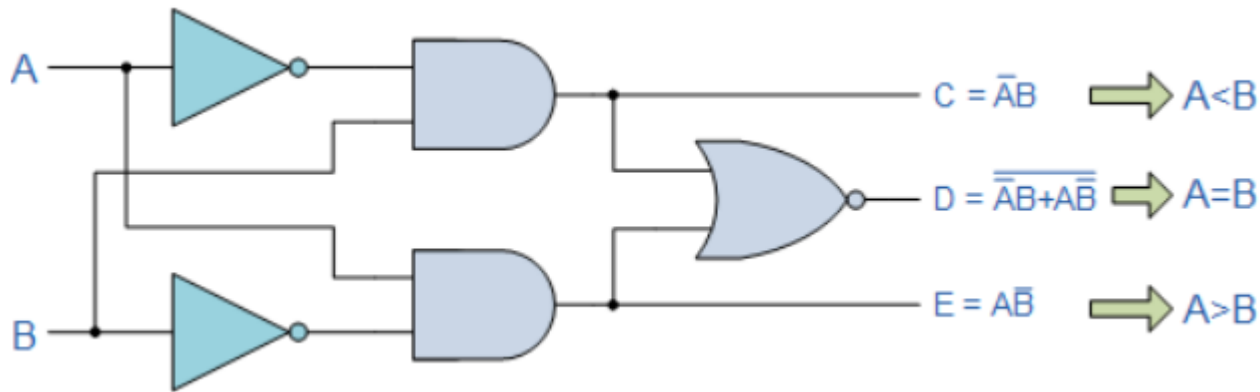
A nyolcas kapcsolósor első fele az **A**, a másik fele pedig a **B**, bemeneteket állítja be. A LED akkor világít, ha az **Y** kimenet '1' állapotban van (egyezés).





# 1-bites magnitúdó komparátor

Az 1 bites alapegység kapcsolása és igazságtáblázata. Itt az X(N)OR kapcsolás eredeti, alpműveletekkel megvalósított kapcsolásához nyúltunk vissza, hogy a „részeredményekhez” is hozzáférhessünk. A felhasznált kapuk így többszörösen kihasználtak...



A	B	A < B	A = B	A > B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0



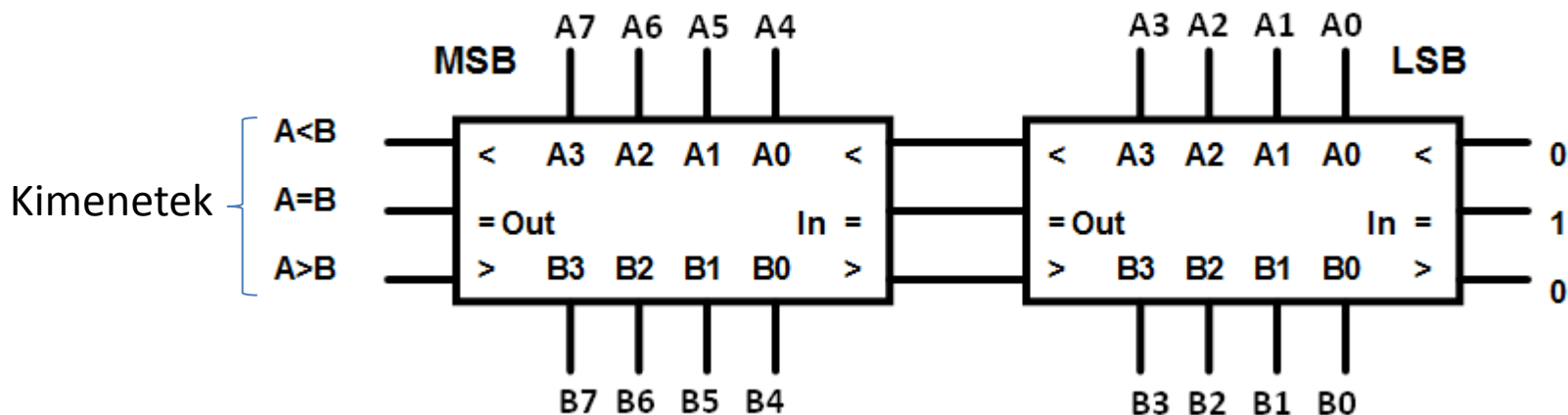


# 8-bites magnitúdó komparátor

A **4063** logikai IC olyan 4-bites magnitúdó komparátort tartalmaz, ami „áthozat” bemeneteket is tartalmaz, tehát kaszkádba köthető.

A legelső komparátor kaszkádosító bemeneteit úgy kell beállítani, mintha egy előző komparátor egyenlőséget jelezne.

Az alábbi ábrán egy 8-bites összehasonlító kapcsolása látható, amit két **4063** IC-vel építhetünk meg. A megépítéshez az [en.f-alpha.net/electronics/digital-electronics/digital-comparator/lets-go/experiment-8-8-bit-magnitude-comparator/](http://en.f-alpha.net/electronics/digital-electronics/digital-comparator/lets-go/experiment-8-8-bit-magnitude-comparator/) oldalon találunk útmutatót.

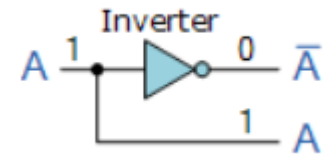




# Bináris dekódolók

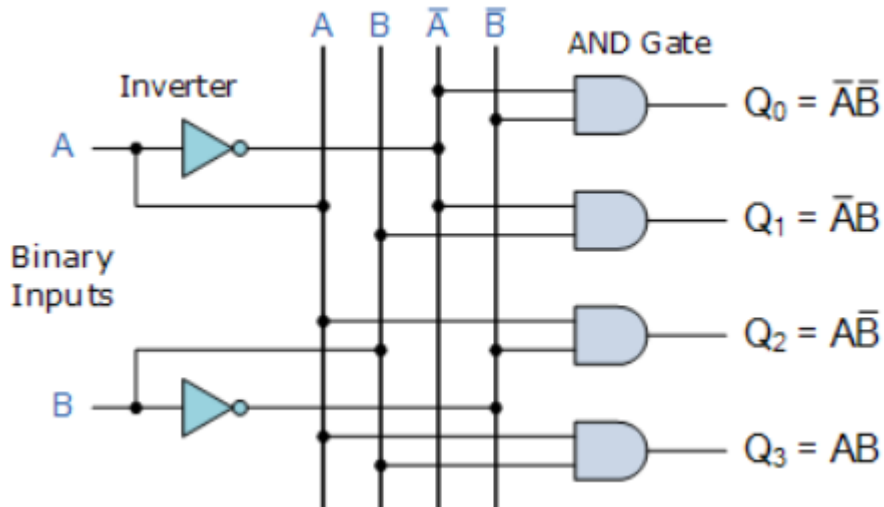
A bináris dekódoló olyan kombinációs logikai áramkör, ami az  $n$  bemenetre érkező bináris információt  $m \leq 2^n$  egyedi kimenő jellé alakítja. Tipikus alkalmazásai: cím- és utasításdekódoló, kijelző vezérlő dekódoló (pl. BCD  $\rightarrow$  7 szegmens dekódoló).

Az inverter (NEM kapu) például 1 bites bináris dekódolónak tekinthető.



Általánosságban az  $n$  bitről  $m$  kimenetre dekódoló áramkör  $m$  darab **mintermet** (a bemeneti változókból vagy azok komplementjéből képzett szorzatok) valósít meg.

## Példa: 2 bitről 4 vonalra dekódoló áramkör



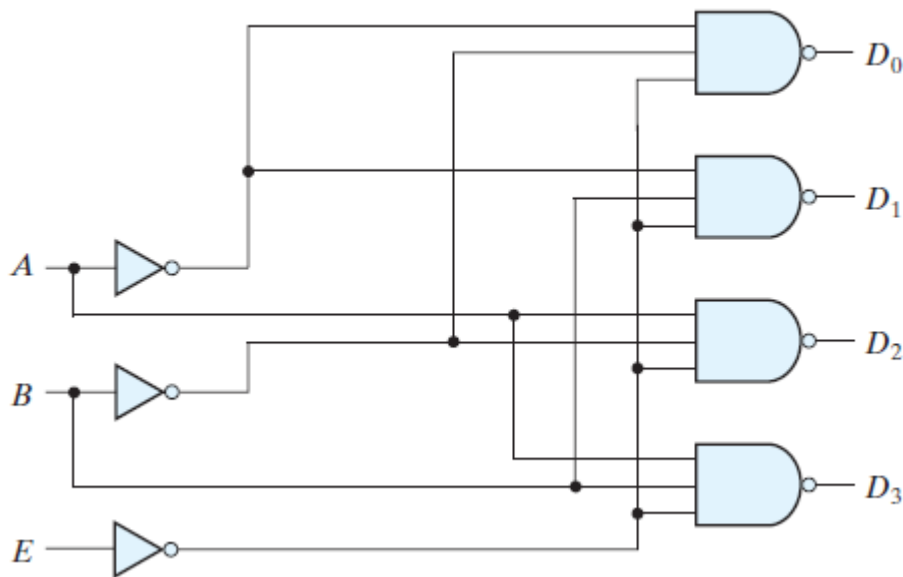
A	B	Q0	Q1	Q2	Q3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



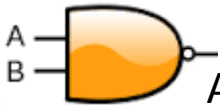
# Negált kimenetű bináris dekódoló

- Némelyik dekódoló NAND (Nem-ÉS) áramkörökkel valósítják meg, ilyenkor a kimenetek aktív állapota alacsony szint (0). Egyidejűleg legfeljebb egy kimenet lehet aktív.
- Ha a kapubemenetek számát bővítjük, akkor egy engedélyező jelet is bevezethetünk. Az alábbi kapcsolásban az engedélyező jel is alacsony szintű jelre aktiválja a dekódolót.

Az ábrán láthatóhoz hasonló áramkör a **74LS138** TTL IC (3 bitről 8 vonalra dekódoló).

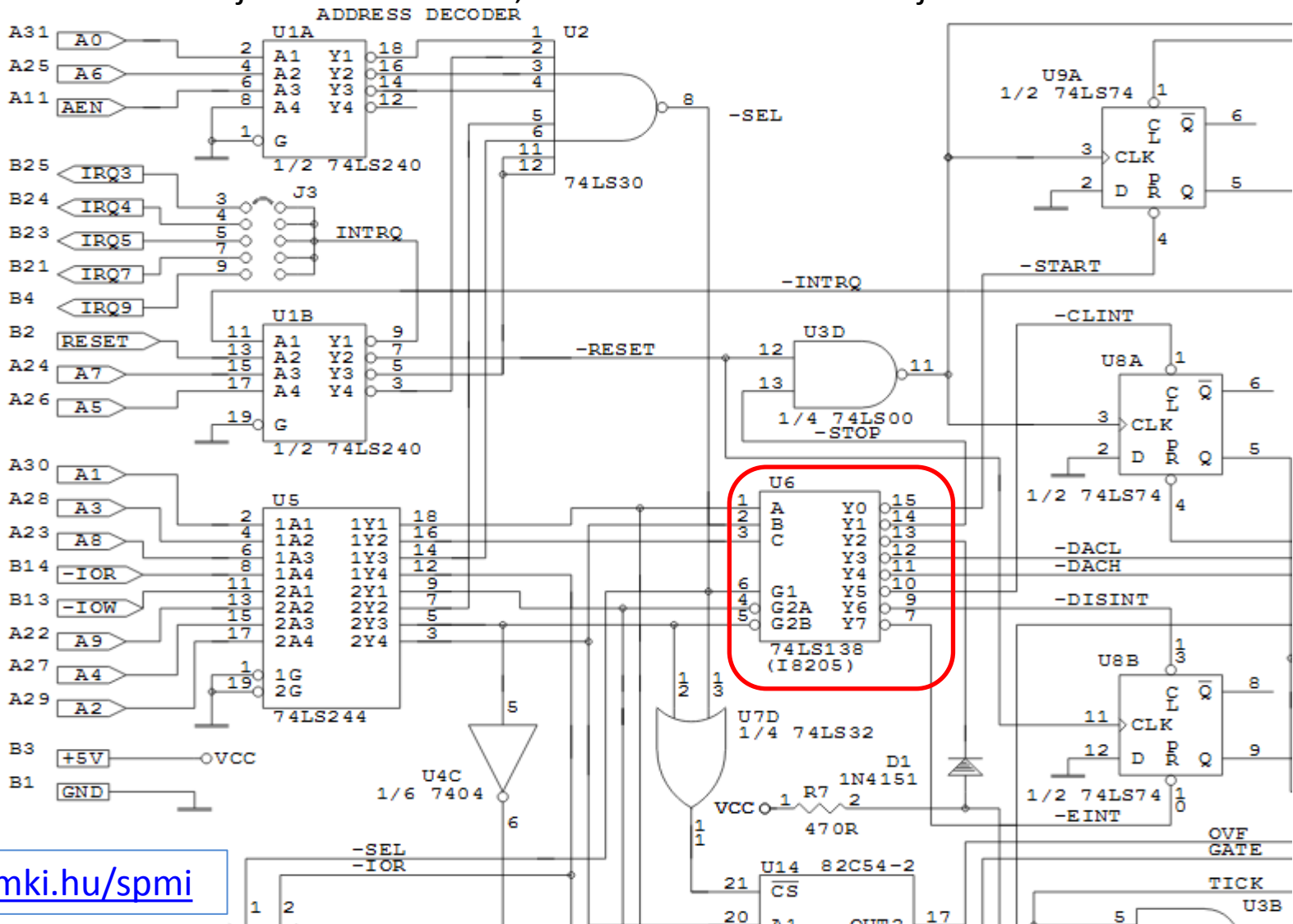


<i>E</i>	<i>A</i>	<i>B</i>	<i>D</i> <sub>0</sub>	<i>D</i> <sub>1</sub>	<i>D</i> <sub>2</sub>	<i>D</i> <sub>3</sub>
1	<i>X</i>	<i>X</i>	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

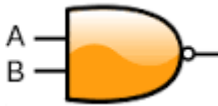


# ISA buszos mérőkártya részlet

A **74LS138** bináris dekódoló itt a periféria cím A1, A2, A3 bitjeit dekódolja az **-IOW** vezérlő jellel szinkronban, s ad 8 különféle vezérlő jelet a címzés szerint.



[desca.atomki.hu/spmi](http://desca.atomki.hu/spmi)

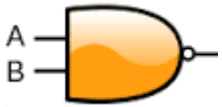


# ISA buszos mérőkártya vezérlő parancsai

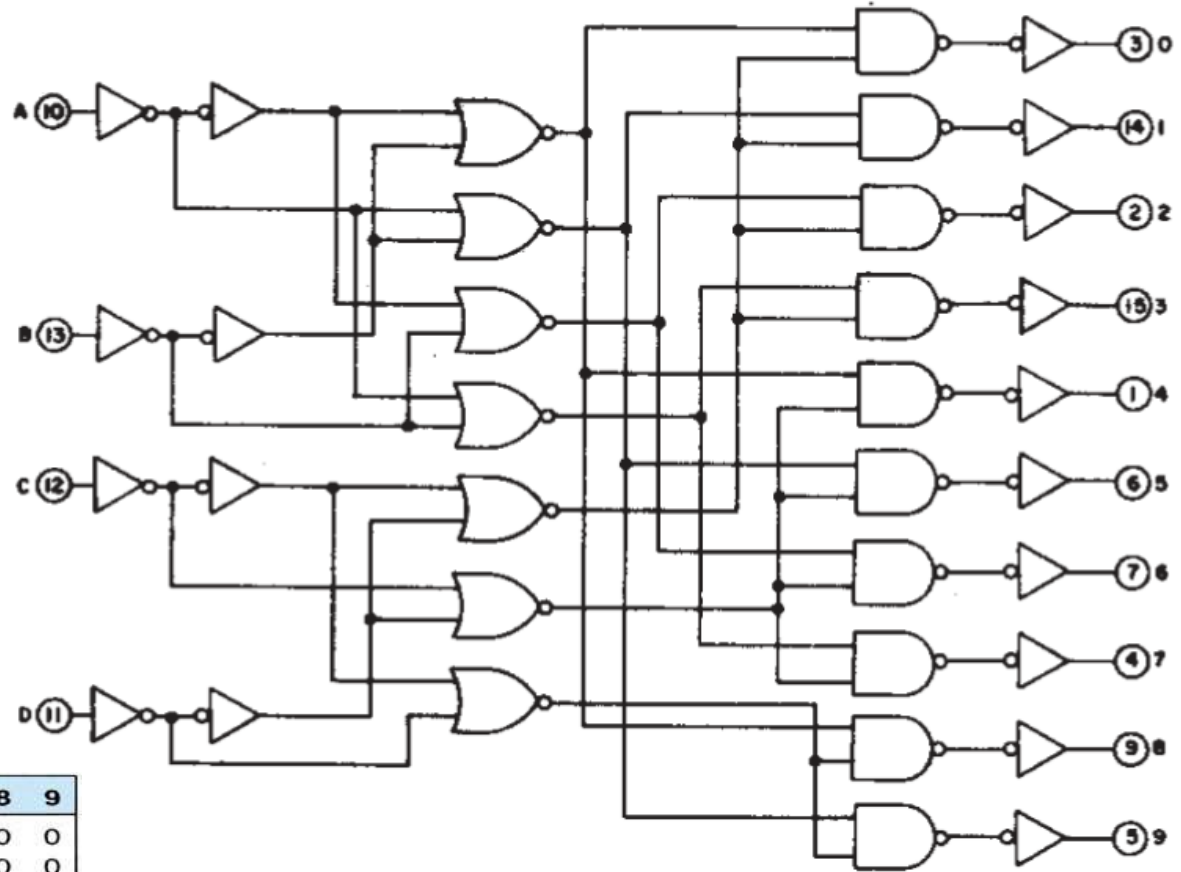
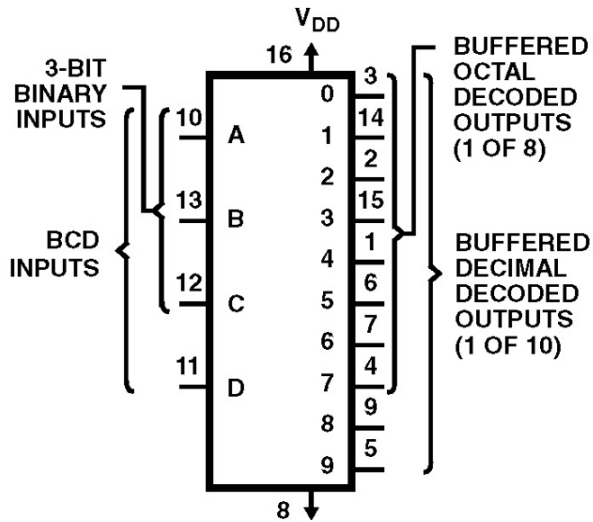
Az előző oldalon kiemelt **74LS138** IC az itt bekeretezett utasításokat dekódolja.

I/O address		Performed function	
Hex	Dec	Read	Write
300	768	Read counter #0	Write counter #0
302	770	Read counter #1	Write counter #1
304	772	Read counter #2	Write counter #2
306	774	Read control reg	Write control register
308	776	Same as 300H	Same as 300H
30A	778	Same as 302H	Same as 302H
30C	780	Same as 304H	Same as 304H
30E	782	Same as 306H	Same as 306H
310	784	Read status reg	Start measurement
312	786	Read status reg	Abort measurement
314	788	Read status reg	Toggle input of CT2
316	790	Read status reg	Write DAC low byte
318	792	Read status reg	Write DAC high byte
31A	794	Read status reg	Clear interrupt flag
31C	796	Read status reg	Disable interrupt
31E	798	Read status reg	Enable interrupt

Link: [desca.atomki.hu/spmi](http://desca.atomki.hu/spmi)



# CD4028: BCD - decimális dekódoló



D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

1 = High Level 0 = Low Level

Lásd: [decoder4-10.circ](#)  
Mintaáramkör Logosim-hez

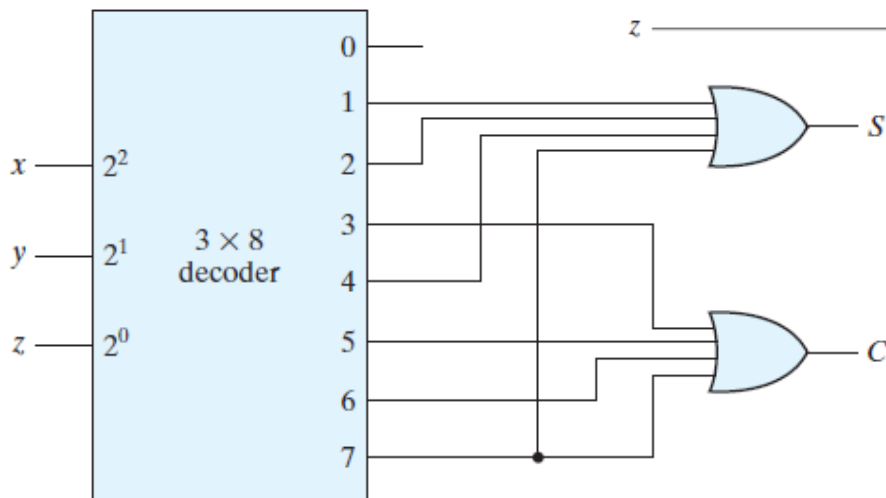
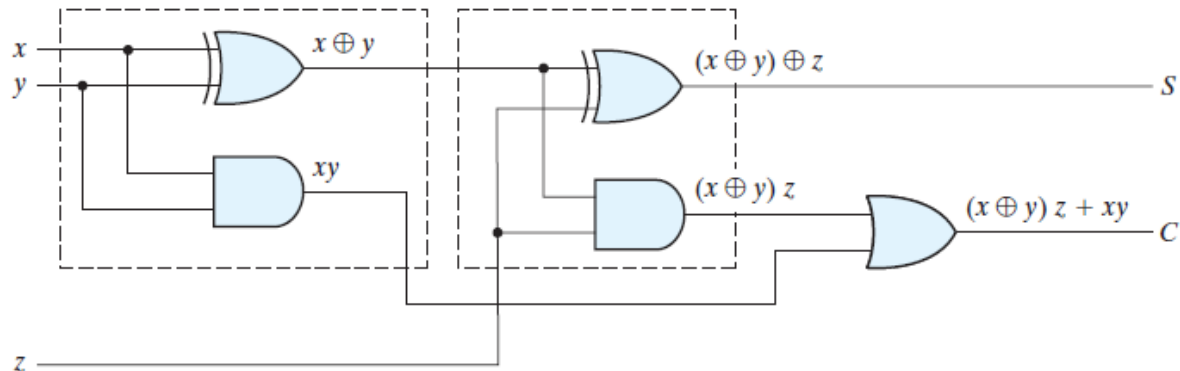
# Kombinációs logikai áramkörök megvalósítása dekódolókkal



Minden kombinációs logikai hálózat felírható a bemenetek ponált vagy negált állapotai szorzatának összegeként (kanonikus alak), a bináris dekódoló pedig éppen ezen szorzatokat állítja elő, így egyetlen külső **VAGY** kapu hozzáadásával tetszés szerinti kombinációs logikai áramkört megépíthetünk. (A **NAND** kapukkal megvalósított negált kimenetű dekódereket pedig a **De Morgan** szabály szerint külső **NAND** kapuval kell kiegészíteni.)

## Példa: teljes összeadó

Az eredeti kapcsolás  
Az új megvalósítás



$x$	$y$	$z$	$C$	$S$	Sorsz.
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	2
0	1	1	1	0	3
1	0	0	0	1	4
1	0	1	1	0	5
1	1	0	1	0	6
1	1	1	1	1	7



# Kódoló áramkörök (encoders)

A kódoló áramkör feladata a fordítottja a dekódolónak, például  $2^n$  bemenő vonal állapotának megfelelően kiad egy  $n$  bites bináris kódot a kimenetén. Egy egyszerű példa a 8 bemenetet három bites bináris számmá alakító kódoló, melynek az igazságtáblázata alább látható. A kimenetek az alábbi formulák szerint egy-egy négybemenetű **VAGY** áramkörrel egyszerűen megépíthetők.

$$z = D_1 + D_3 + D_5 + D_7$$

$$y = D_2 + D_3 + D_6 + D_7$$

$$x = D_4 + D_5 + D_6 + D_7$$

Bemenetek								Kimenetek		
$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$	$x$	$y$	$z$
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

**Ennek az áramkörnek azonban komoly hátrányai vannak:**

- ❖ Egyszerre csak egy bemenet lehet aktív, különben hamis kódot kapunk.
- ❖ Ha egyik bemenet sem aktív, ugyanúgy **000** kódot kapunk, mint amikor a  $D_0$  bemenet aktív.

A következő példánál gondot fordítunk majd ezen hátrányok kiküszöbölésére.

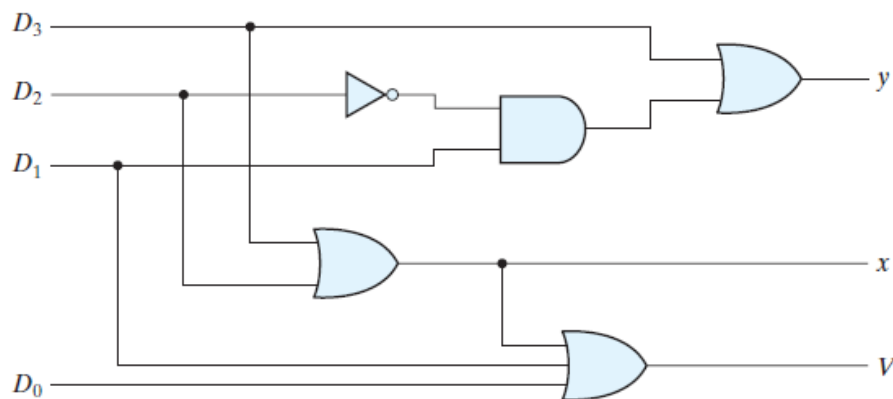
# Prioritás kódoló



- ❑ A **prioritás kódoló** több aktív bemenet esetén csak a legmagasabb prioritásút veszi figyelembe. A prioritás előre behuzalozott, a bemenetek sorszámától függ.
- ❑ A tétlen állapot megkülönböztetésére (amikor egyik bemenet sem aktív) egy külön jelet is előállítunk, ami azt jelzi, hogy van-e aktív bemenet.

A mikroprocesszoros buszon beérkező megszakítási kérelmek prioritizálására is használtak ilyen áramköröket. Az alábbi példában egy 4 bemenetű prioritás kódolót mutatunk be.

Bemenetek				Kimenetek		
$D_0$	$D_1$	$D_2$	$D_3$	$x$	$y$	$V$
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

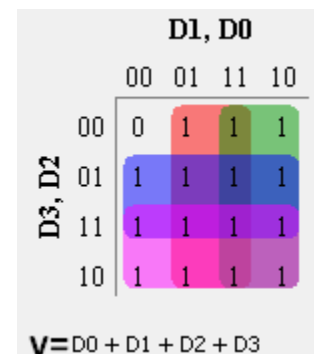
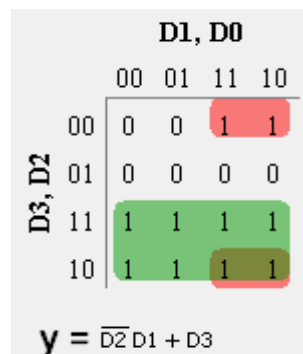
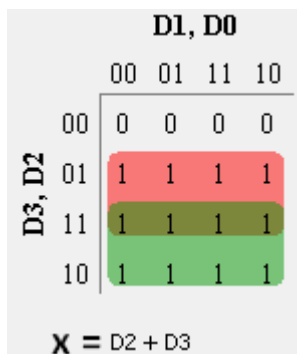


$$x = D_2 + D_3$$

$$y = D_3 + D_1 D_2'$$

$$V = D_0 + D_1 + D_2 + D_3$$

Lásd: [priority\\_encoder.circ](#)  
mintaáramkör Logisim-hez

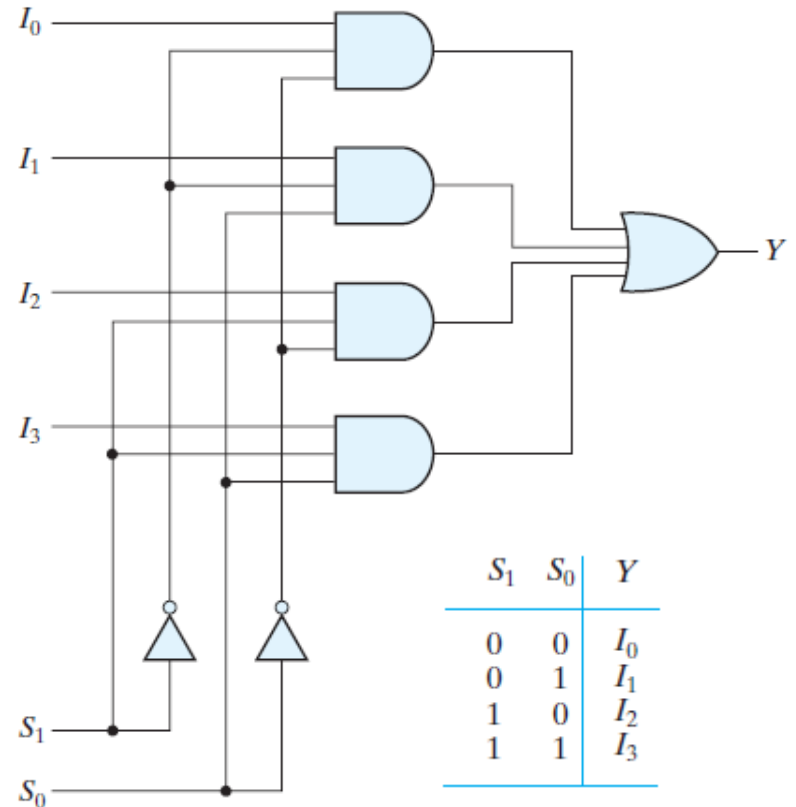
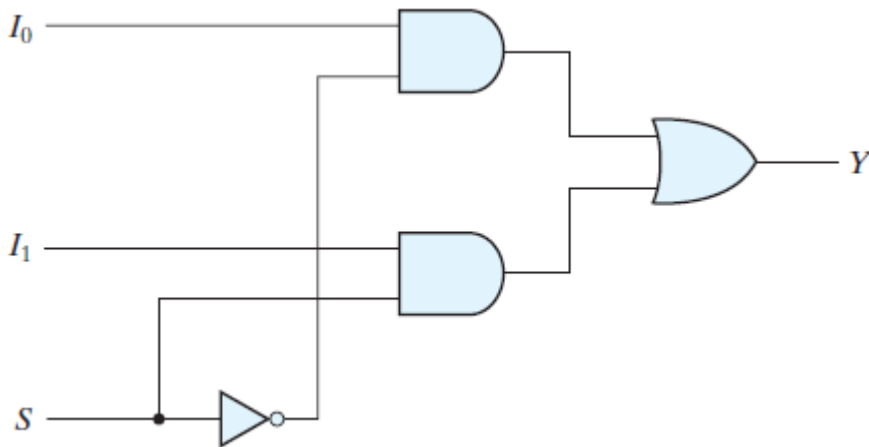




# Multiplexerek

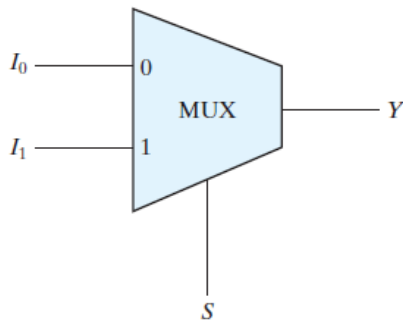
A **multiplexer** olyan kombinációs logikai áramkör, ami több bemenet egyikéről a bináris információt a kimenetre irányítja át. A kiválasztás az úgynevezett kiválasztó bemenetek beállításával történik, s  $2^n$  bemenet esetén  $n$  darab kiválasztó bemenő jel szükséges, melyek bináris kombinációja kiválaszt egy adatbemenetet.

Az alábbi ábrákon egy 2 bemenetű és egy 4 bemenetű multiplexert mutatunk be.



## Blokkvázlat

A mikrovezérlő adatlapokban találkozhatunk ilyen jelöléssel.





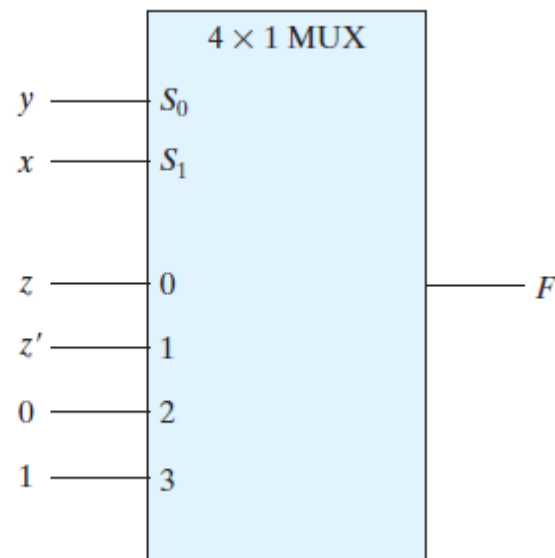
# Alkalmazás: Logikai függvény megvalósítása multiplexerrel

Egy  $n$  változós logikai függvényt egy  $n-1$  bemenetválasztós multiplexerrel is megépíthetünk.

Például:  $F(x, y, z) = \Sigma(1, 2, 6, 7)$  ( $F$  oszlopában az 1., 2., 6. és 7. sorokban áll egyes).

A bemenetválasztók  $x$  és  $y$  lesznek, a négy bemenetre pedig az igazságtáblázat alapján  $z$ ,  $z'$ ,  $0$  és  $1$  kötendő.

$x$	$y$	$z$	$F$	
0	0	0	0	$F = z$
0	0	1	1	
0	1	0	1	$F = z'$
0	1	1	0	
1	0	0	0	$F = 0$
1	0	1	0	
1	1	0	1	$F = 1$
1	1	1	1	



Vegyük észre, hogy az  $F$  oszlopban két-két soronként nézve a függvény értéke vagy megegyezik  $z$ -vel, vagy ellentétes vele, vagy csupa nulla, vagy csupa 1-es, más lehetőség nincs.



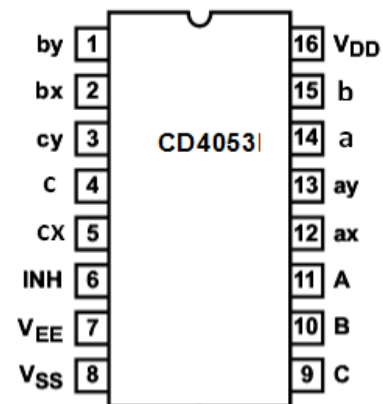
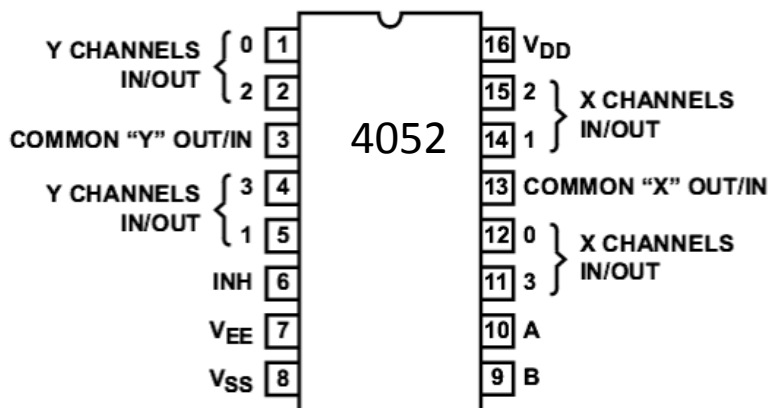
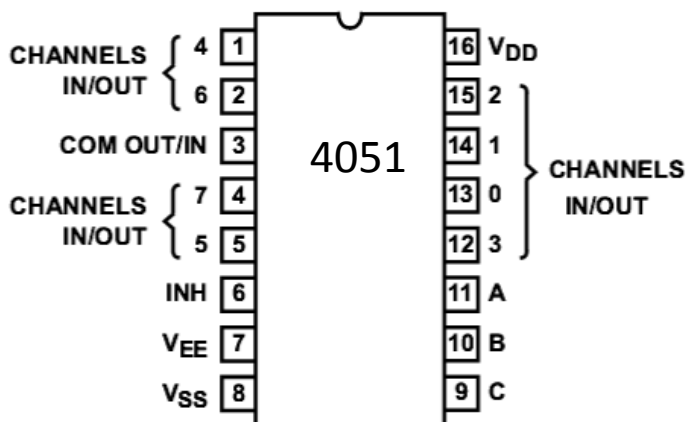
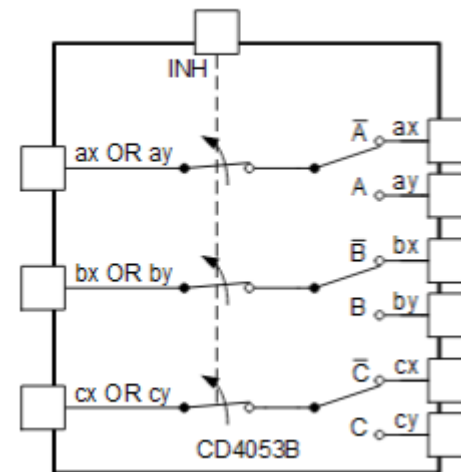
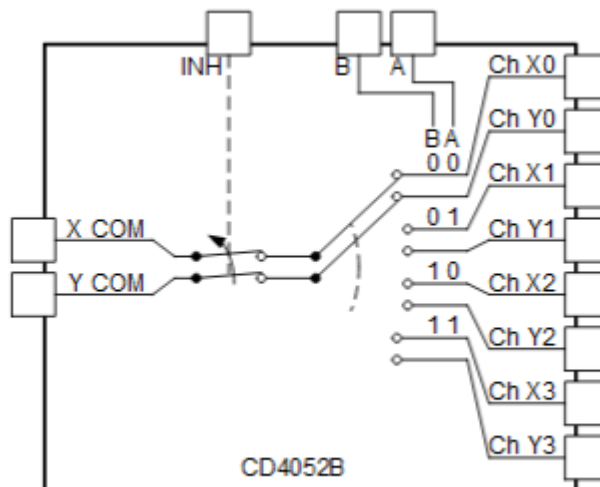
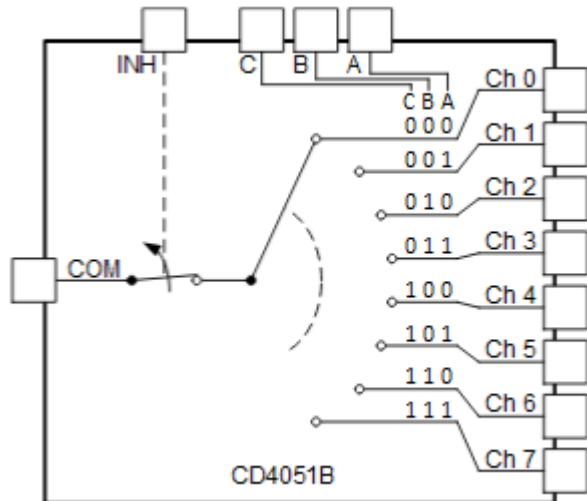
# Analóg multiplexerek

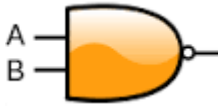
4051 CMOS Single 8-Channel Analog Multiplexer/Demultiplexer with Logic-Level Conversion

4052 CMOS Differential 4-Channel Analog Multiplexer/Demultiplexer with Logic-Level Conversion

4053 CMOS Triple 2-Channel Analog Multiplexer/Demultiplexer with Logic-Level Conversion

A kiválasztás digitális, a csatorna viszont analóg jeleket visz át.

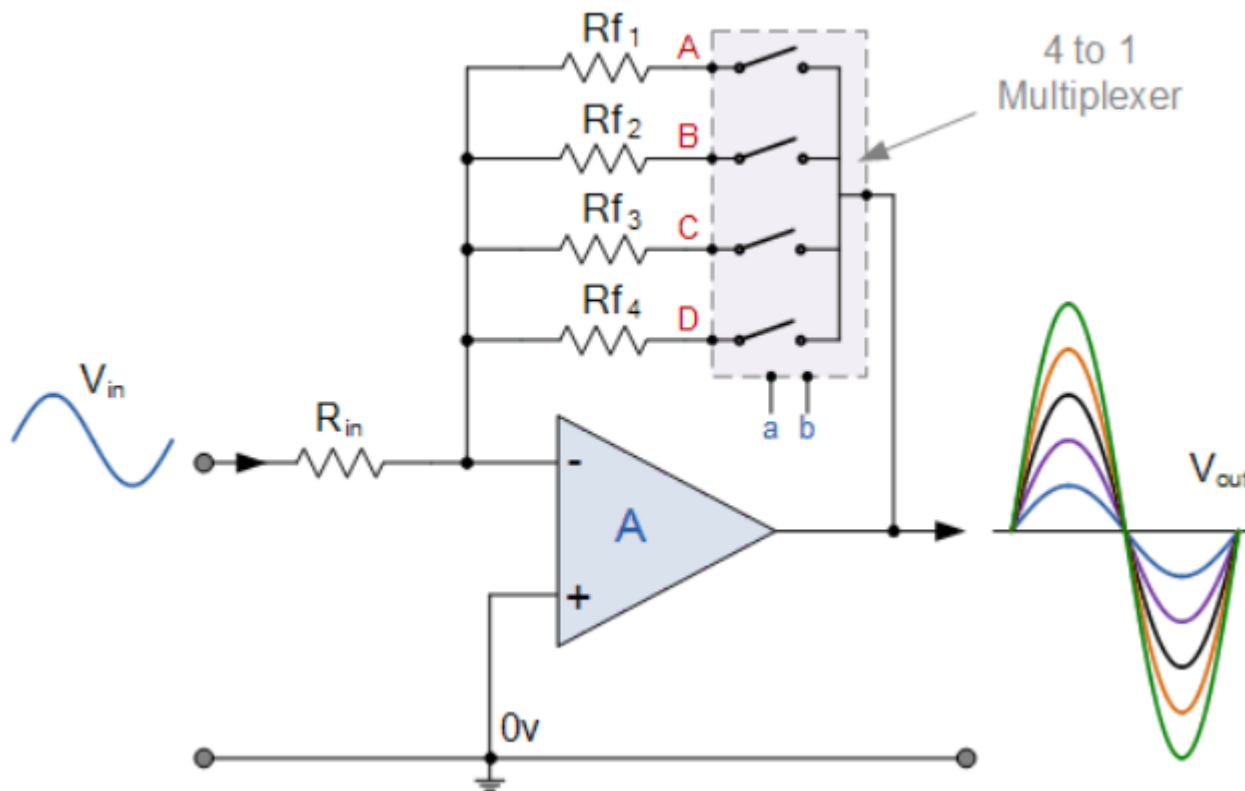




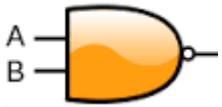
# Analóg multiplexer alkalmazása

Digitálisan változtatható erősítés.

$$\text{Gain (Av)} = \frac{V_{\text{out}}}{V_{\text{in}}} = -\frac{R_f}{R_{\text{in}}}$$



Link: [www.electronics-tutorials.ws/combin/comb\\_2.html](http://www.electronics-tutorials.ws/combin/comb_2.html)

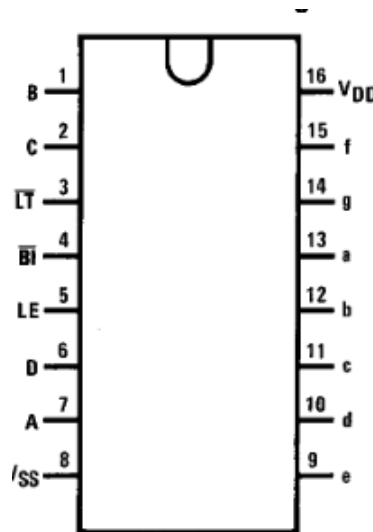
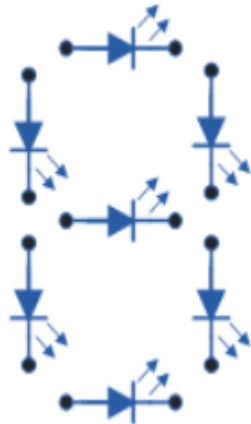
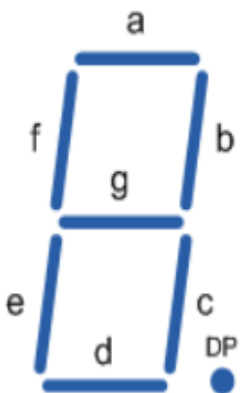


# CD4511: 7-szegmens dekóder

A 7-szegmes dekóder feladata az, hogy a bemenetekre kapcsolt bináris (BCD) kód függvényében a hétszegmenses kijelző egyes Szegmenseinek állapotát meghatározza.

## Alkalmazási példa:

NEMO [KAT120B számkijelző panel](#)

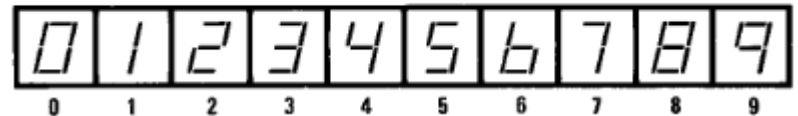


Inputs							Outputs							
LE	$\overline{BI}$	$\overline{LT}$	D	C	B	A	a	b	c	d	e	f	g	Display
X	X	0	X	X	X	X	1	1	1	1	1	1	1	B
X	0	1	X	X	X	X	0	0	0	0	0	0	0	
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	1	1	1	1	0	0	1	2
0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
0	1	1	1	0	0	1	1	1	1	0	0	1	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	
0	1	1	1	0	1	1	0	0	0	0	0	0	0	
0	1	1	1	1	0	0	0	0	0	0	0	0	0	
0	1	1	1	1	1	0	0	0	0	0	0	0	0	
0	1	1	1	1	1	1	0	0	0	0	0	0	0	
0	1	1	1	1	1	1	0	0	0	0	0	0	0	
1	1	1	X	X	X	X				*	*	*	*	*

X = Don't Care

\*Depends upon the BCD code applied during the 0 to 1 transition of LE.

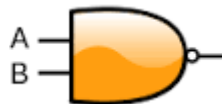
Display





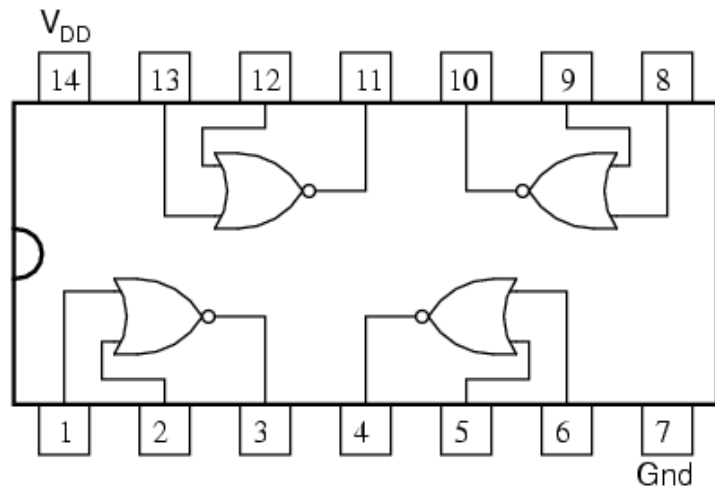
# A 4000-es sorozat tipikus tagjai

- 4001 CMOS Quad 2-Input NOR Gate
- 4011 CMOS Quad 2-Input NAND Gate
- 4013 CMOS Dual D-Type Flip Flop
- 4017 CMOS Decade Counter with 10 Decoded Outputs
- 4021 CMOS 8-Stage Static Shift Register
- 4022 CMOS Octal Counter with 8 Decoded Outputs
- 4023 CMOS Triple 3-Input NAND Gate
- 4025 CMOS Triple 3-Input NOR Gate
- 4026 CMOS Decade Counter/Divider with Decoded 7-Segment Display Outputs and Display Enable
- 4027 CMOS Dual J-K Master-Slave Flip-Flop
- 4028 CMOS BCD-to-Decimal or Binary-to-Octal Decoders/Drivers
- 4043 CMOS Quad NOR R/S Latch with 3-State Outputs
- 4046 CMOS Micropower Phase-Locked Loop
- 4049 CMOS Hex Inverting Buffer/Converter
- 4050 CMOS Hex Non-Inverting Buffer/Converter
- 4051 CMOS Single 8-Channel Analog Multiplexer/Demultiplexer with Logic-Level Conversion
- 4052 CMOS Differential 4-Channel Analog Multiplexer/Demultiplexer with Logic-Level Conversion
- 4053 CMOS Triple 2-Channel Analog Multiplexer/Demultiplexer with Logic-Level Conversion
- 4060 CMOS 14-Stage Ripple-Carry Binary Counter/Divider and Oscillator
- 4066 CMOS Quad Bilateral Switch
- 4069 CMOS Hex Inverter
- 4070 CMOS Quad Exclusive-OR Gate
- 4071 CMOS Quad 2-Input OR Gate
- 4072 CMOS Dual 4-Input OR Gate
- 4073 CMOS Triple 3-Input AND Gate
- 4075 CMOS Triple 3-Input OR Gate
- 4081 CMOS Quad 2-Input AND Gate
- 4082 CMOS Dual 4-Input AND Gate
- 4093 CMOS Quad 2-Input NAND Schmitt Triggers
- 4094 CMOS 8-Stage Shift-and-Store Bus Register

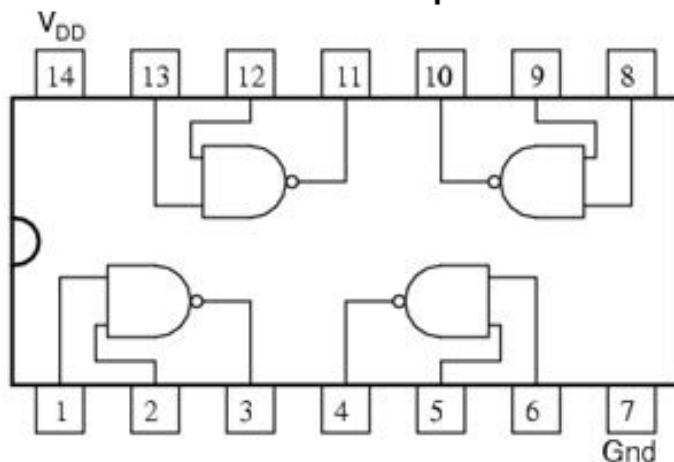


# A 4000-es sorozat tipikus tagjai

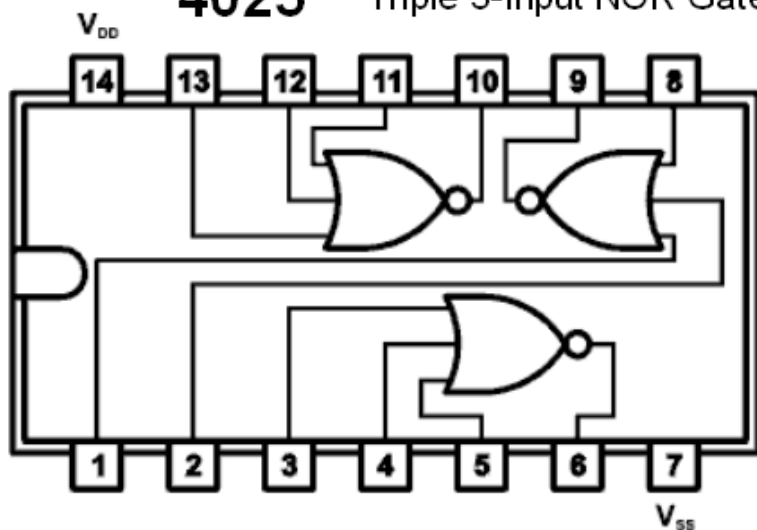
**4001** Quad 2-Input NOR Gate



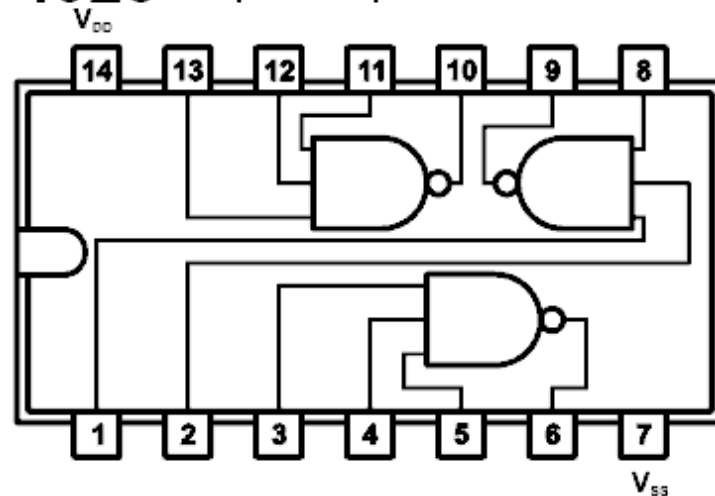
**4011** Quad 2-input NAND



**4025** Triple 3-Input NOR Gate



**4023** Triple 3-Input NAND Gate

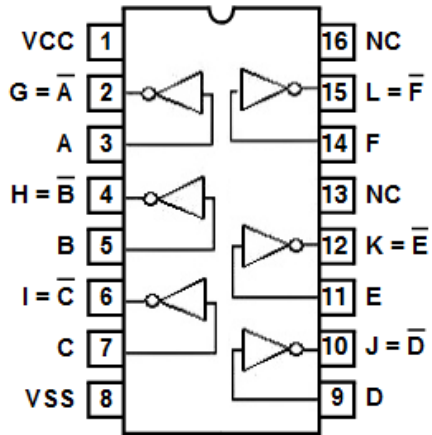




# A 4000-es sorozat tipikus tagjai

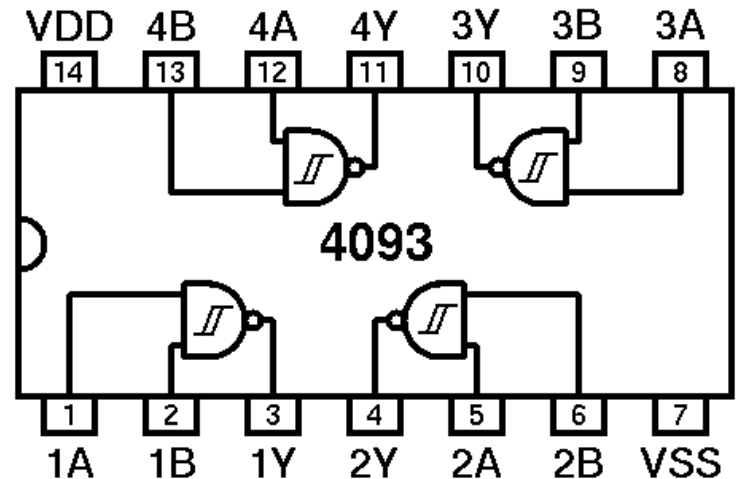
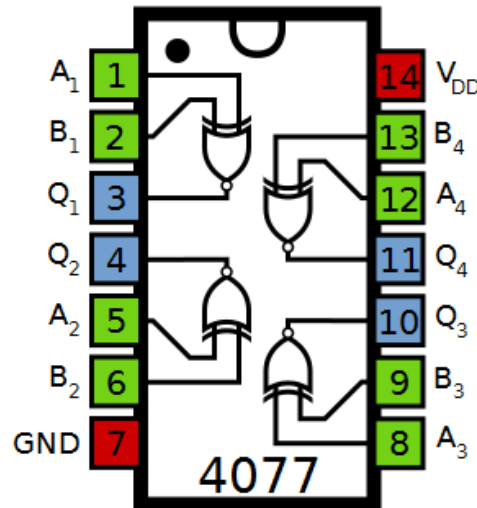
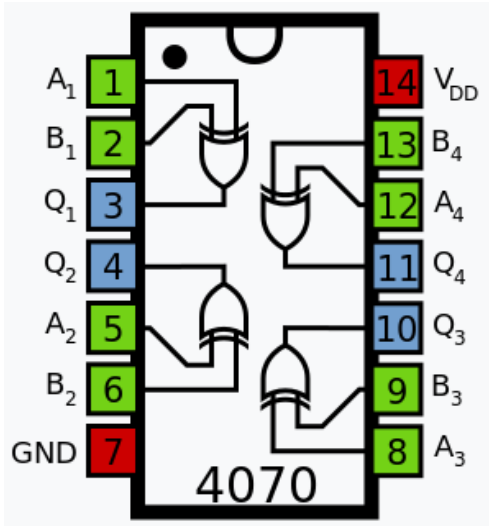
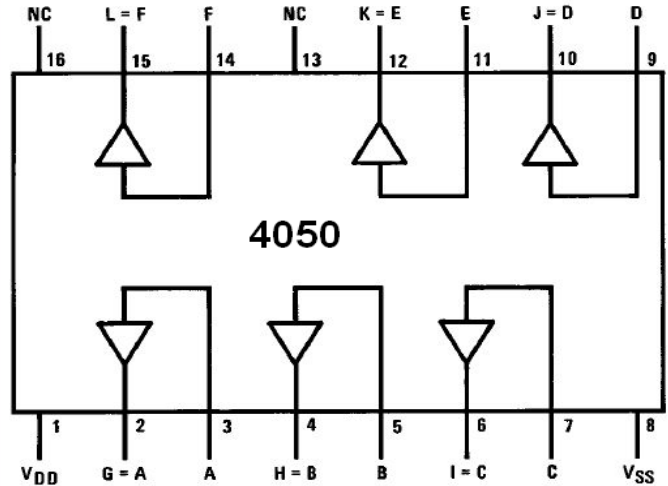
## 4049

Hex Inverting Buffer/Converter



## 4050

Hex Non-Inverting Buffer/Converter





# A 4000-es sorozat tipikus tagjai

