

Bevezetés az elektronikába

	y	
x	0	1
0	0	0
1	0	1

	y	
x	0	1
0	0	1
1	1	1

	y	
x	0	1
0	1	1
1	0	1

	y	
x	0	1
0	0	1
1	1	0

Figure 1. Truth tables



Figure 2. Logic gates



Figure 3. De Morgan equivalents

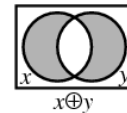
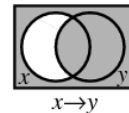
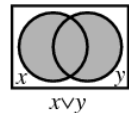
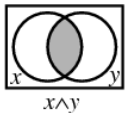


Figure 4. Venn diagrams



5. Számrendszerek, Boole-algebra, Logikai függvények

Felhasznált irodalom

- ❑ Ránkyné Szalay Rita: A 2-es számrendszer tanítása interaktív módon
- ❑ M. Morris Mano and Michael D. Ciletti:
Digital Design - With an Introduction to the Verilog HDL, 5th. Edition
- ❑ Mészáros Miklós: Logikai algebra alapjai, logikai függvények I.
- ❑ BME FKE: Logikai áramkörök
- ❑ Logisim szimulátor: www.cburch.com/logisim/
- ❑ Falstad.com: Circuit simulator
- ❑ F-alpha.net: Boolean Algebra
- ❑ Neuproductions-be: The Logic Lab

A kettes számrendszer

- Hófehérke hét törpéje a kibányászott gyémántokat csomagolja. Ha már két gyémánt, vagy csomag van Kuka kezében, akkor összecsomagolja, és továbbadja Hapcinak.

Tudor



Vidor



Morgó



Szende



Szundi



Hapci



Kuka



A kettes számrendszer

- Hány gyémánt után jutunk el az alábbi ábrán látható helyzetig?

Szende



Szundi



Hapci



Kuka



$$8 + 4 + 2 + 0 = 14_{10}$$

Kettes számrendszerben csak azt írjuk le, hogy az **1110₂**
az adott helyiértéken van-e „csomag” (1), vagy nincs (0)

A kettes számrendszer

- Példa: 125 gyémánt hogyan oszlik meg a törpék kezében?



$125 > 64$	(1)	$125 - 64 = 61$
$61 > 32$	(1)	$61 - 32 = 29$
$29 > 16$	(1)	$29 - 16 = 13$
$13 > 8$	(1)	$13 - 8 = 5$
$5 > 4$	(1)	$5 - 4 = 1$
$1 < 2$	(0)	$1 - 0 = 1$
$1 = 1$	(1)	

$111\ 1101_2$

A tizenhatos számrendszer

- Példa: 125 gyémánt hogyan oszlik meg a törpék kezében?



A bináris (kettes) számrendszerben felírt szám jegyeit 4-es csoportokra osztjuk, és csoportonként egy-egy hexadecimális számjegyként értelmezzük, felhasználva, hogy $2 \cdot 2 \cdot 2 \cdot 2 = 16$

0000 = 0	1000 = 8
0001 = 1	1001 = 9
0010 = 2	1010 = A
0011 = 3	1011 = B
0100 = 4	1100 = C
0101 = 5	1101 = D
0110 = 6	1110 = E
0111 = 7	1111 = F

$111 \mid 1101_2$
 7 D₁₆

Online átváltást segítő honlap

- Link: convertworld.com/hu/szamrendszerek/tizenhatos-szamrendszer.html

Tizenhatos számrendszer (Helyiérték rendszerek), számrendszerek



Írja be a konvertálni kívánt **Tizenhatos számrendszer** számot a szövegdobozba, hogy megjelenítse az eredményeket a táblázatban.

125 Tízes számrendszer

egyenlő

7D

Helyiérték rendszerek

Kettes számrendszer	1111101
Hármas számrendszer	11122
Négyes számrendszer	1331
Ötös számrendszer	1000
Hatos számrendszer	325
Hetes számrendszer	236
Nyolcas számrendszer	175
Kilences számrendszer	148
Tízes számrendszer	125
Tizenegyes számrendszer	104
Tizenkettes számrendszer	A5
Tienhármasszámrendszer	98

Húszas számrendszer 65

Számrendszerek

Arab számok 125
Római számok CXXV

Nyelvek

Angol one hundred twenty five
Német hundertfünfundzwanzig
Spanyol ciento veinticinco
Svéd etthundratjugofem

A Boole-algebra és alkalmazásai

- **A Boole-algebra** egy kétműveletes algebrai struktúra amely a halmazműveletek, a logikai műveletek és az eseményalgebra műveleteinek közös tulajdonságaival rendelkezik. (George Boole 1854)
- **A kijelentéslogika** pl. olyan kijelentésekkel dolgozik, amelyek vagy "igazak", vagy "hamisak", és keressük az olyan kijelentések valóságtartalmát, amelyek igaz, vagy hamis elemi kijelentésekből tevődnek össze.
- **A kapcsolási algebra** azt vizsgálja, hogy a logikai kapuáramkörökből összeállított hálózat kimenetén a lehetséges két állapot melyike valósul meg, ha a bemenetek az egyik vagy másik lehetséges állapotban vannak. Ezért a Boole-algebra az **elektronikus digitális számítógép** konstruálásának nélkülözhetetlen elméleti alapja.
- **Logikai alapműveletek** az **ÉS**, **VAGY** és a **NEGÁCIÓ**. Minden további művelet összetett, ezekből levezethető.

A kétértékű Boole-algebra definiálása

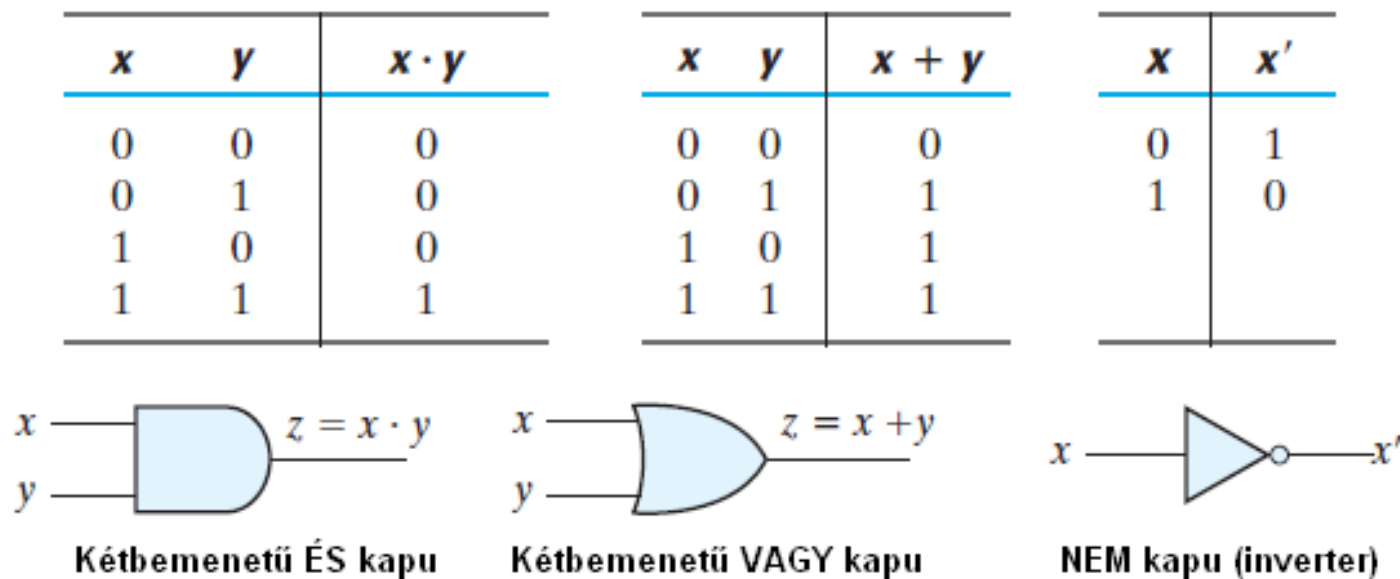
- A kétértékű **Boole-algebra** megadható $B = \{0, 1\}$ halmazával és a halmaz elemein végezhető $+$ (szóban: **VAGY**) és $\cdot$ (szóban: **ÉS**) bináris műveletekkel, amelyekre teljesülnek **E. V. Huntington** 1904-ben megfogalmazott posztulátumai:

- (a) A B halmaz zárt a $+$ bináris műveletre nézve, azaz bármely $x, y \in B$ esetén a $z = x + y$ művelet eredménye is eleme B -nek ($z \in B$).
(b) A B halmaz zárt a $\cdot$ bináris műveletre nézve, azaz bármely $x, y \in B$ esetén a $z = x \cdot y$ művelet eredménye is eleme B -nek ($z \in B$).
- (a) A 0 zéruselem a $+$ műveletre nézve, vagyis $x + 0 = 0 + x = x$.
(b) Az 1 egységelem a $\cdot$ bináris műveletre nézve, vagyis $x \cdot 1 = 1 \cdot x = x$.
- (a) A struktúra kommutatív a $+$ bináris műveletre nézve, azaz $x + y = y + x$.
(b) A struktúra kommutatív a $\cdot$ bináris műveletre nézve, azaz $x \cdot y = y \cdot x$.
- (a) A $\cdot$ művelet disztributív a $+$ műveletre nézve, azaz $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$.
(b) A $+$ művelet disztributív a $\cdot$ műveletre nézve, azaz $x + (y \cdot z) = (x + y) \cdot (x + z)$.
- bármely $x \in B$ esetén létezik olyan $x' \in B$ elem (amelyet x komplementének nevezünk), amelyre teljesül, hogy: (a) $x + x' = 1$ és (b) $x \cdot x' = 0$.
- A B halmaznak létezik legalább két különböző $x, y \in B$ eleme, amelyre teljesül, hogy $x \neq y$.

Megjegyzés: a $B = \{0, 1\}$ halmaz elemei nem számok, hanem szimbólumok! Az x, y, z logikai változók pedig olyan szimbólumok, melyek értéke vagy a 0 vagy az 1 halmazelem.

A Boole-algebra műveletei

- A kétértékű Boole-algebra bináris (kétváltozós) műveleteinek szabályait táblázatos formában is megadhatjuk. Az ábrán a harmadik művelet a komplementis előállítására szolgál.



- A logikai áramkörök építőkövei az úgynevezett kapuáramkörök, amelyek egy-egy elemi logikai művelet (NEM, ÉS, VAGY kapcsolat) elvégzésére képesek. A logikai áramköröknél a „0” pl. alacsony jelszinttel, az „1” pedig magas jelszinttel reprezentálható.

Tételek és tulajdonságok

A posztulátumokból és műveleti táblázatokból újabb összefüggéseket vezethetünk le:

Dualitás elve: A posztulátumokat (a) és (b) párok formájában adtuk meg a $+$ és $\cdot$ műveletekre. Egyikből megkaphatjuk a másikat, ha a $+$ és $\cdot$ műveleteket felcseréljük, valamint a 0 helyébe 1-et, illetve az 1 helyébe 0-t írunk. Ez egy általános tulajdonsága a Boole-algebrának, a levezetett tételekre is igaz.

Alaptételek

- Idempotencia: (a) $x + x = x$ és (b) $x \cdot x = x$
- Korlátosság: (a) $x + 1 = 1$ és (b) $x \cdot 0 = 0$
- Involúció: $(x')' = x$
- Asszociativitás: (a) $x + (y + z) = (x + y) + z$ és (b) $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
- De Morgan azonosságok: (a) $(x + y)' = x' \cdot y'$ és (b) $(x \cdot y)' = x' + y'$
- Abszorpció: (a) $x + (x \cdot y) = x$ és (b) $x \cdot (x + y) = x$

A Boole-algebra azonosságai

- A Boole-algebrában az azonos átalakításokat az alábbi táblázatban összefoglalt azonosságok szerint végezhetjük.

Postulates and Theorems of Boolean Algebra

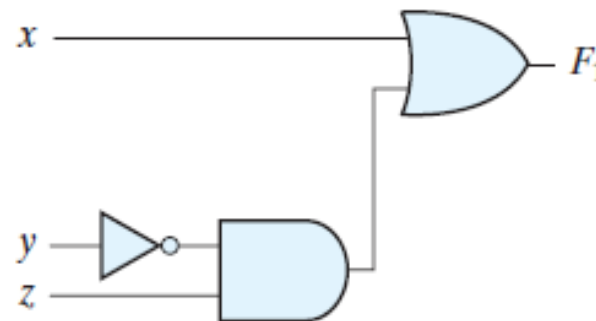
Postulate 2	(a)	$x + 0 = x$	(b)	$x \cdot 1 = x$
Postulate 5	(a)	$x + x' = 1$	(b)	$x \cdot x' = 0$
Theorem 1	(a)	$x + x = x$	(b)	$x \cdot x = x$
Theorem 2	(a)	$x + 1 = 1$	(b)	$x \cdot 0 = 0$
Theorem 3, involution		$(x')' = x$		
Postulate 3, commutative	(a)	$x + y = y + x$	(b)	$xy = yx$
Theorem 4, associative	(a)	$x + (y + z) = (x + y) + z$	(b)	$x(yz) = (xy)z$
Postulate 4, distributive	(a)	$x(y + z) = xy + xz$	(b)	$x + yz = (x + y)(x + z)$
Theorem 5, DeMorgan	(a)	$(x + y)' = x'y'$	(b)	$(xy)' = x' + y'$
Theorem 6, absorption	(a)	$x + xy = x$	(b)	$x(x + y) = x$

- Megjegyzés: Ahol nem zavarja a megértést, a szorzásjelet elhagytuk.

Logikai függvények

- **Logikai függvények:** kifejezések, melyek bináris (kétértékű) változókból, a 0 és 1 konstansokból és logikai műveleti jelekből állnak. A változók adott értékénél a függvény értéke 0 vagy 1 lesz.
- Például : $F1 = x + y' \cdot z$

Az **F1** értéke 1 lesz, ha $x = 1$, vagy ha y' és z egyaránt 1. Minden más esetben **F1** értéke 0 lesz. (y' akkor 1, ha $y = 0$)
- Egy logikai függvény megadható az **igazságtáblázatával** is, illetve összekapcsolt **logikai kapuk hálózatával** is
- A logikai áramköröknél a függvény változói bemenetként szerepelnek, **F1** értékét pedig az áramkör kimenete jelenti.



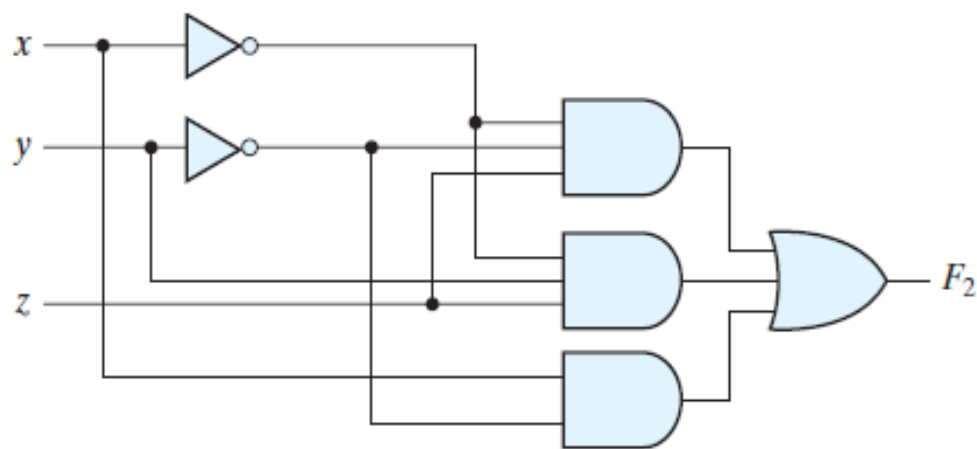
<i>x</i>	<i>y</i>	<i>z</i>	<i>F₁</i>
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Logikai függvények egyszerűsítése

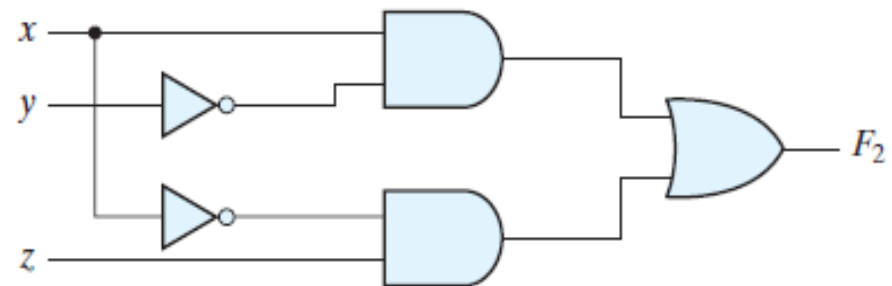
- Az **igazságtáblázattal** egy logikai függvényt **csak egyféleképpen** fejezhetünk ki. **Algebrai formulával** azonban **többféle módon** is megadhatjuk a függvényt úgy, hogy minden alak ugyanarra az eredményre vezessen. Minden egyes formula megfelel egy-egy logikai kapukból kialakított kapcsolásnak.
- A logikai függvények Boole-algebra szabályait követő átalakítása hozzásegít bennünket, hogy egyszerűbb kifejezéssel írjuk fel ugyanazt a függvényt, így kevesebb kapuval vagy kevesebb bemenetű kapuáramkörökkel valósíthatjuk meg.
- Tekintsük például a következő logikai függvényt: $F_2 = x'y'z + x'yz + xy'$
- Egyszerűsítsük a kifejezést a **Boole-algebra** azonosságai alapján:
$$F_2 = x'y'z + x'yz + xy' = x'z(y' + y) + xy' = x'z + xy'$$
- A függvényt leíró kifejezés két tagra egyszerűsödött, ami egyszerűbb áramköri megvalósítást is jelent.

Logikai függvények egyszerűsítése

- Az előző oldalon bemutatott függvény eredeti és egyszerűsített alakjának áramköri megvalósítása az alábbi ábrákon látható.
- Nyilvánvalóan az a kedvezőbb megvalósítás, amelyik kevesebb kaput és kevesebb kapubemenetet igényel, mivel az kevesebb alkatrészből és kevesebb vezetékezéssel kivitelezhető. Általában egy-egy logikai függvénynek sokféle megvalósítása lehet. A leggazdaságosabb megvalósítás megtalálása fontos része a tervezésnek.



(a) $F_2 = x'y'z + x'yz + xy'$



(b) $F_2 = xy' + x'z$

További logikai műveletek

- A kétváltozós logikai függvények igazságtáblázatát az eddig tárgyalt **ÉS** és **VAGY** logikai műveleten kívül másképpen (összesen 16-féleképpen) is kitölthetjük. Ezeket, vagy ezek némelyikét újabb logikai műveletnek is tekinthetjük.

Kétváltozós logikai függvények igazságtáblázatai

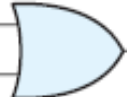
<i>x</i>	<i>y</i>	<i>F</i> ₀	<i>F</i> ₁	<i>F</i> ₂	<i>F</i> ₃	<i>F</i> ₄	<i>F</i> ₅	<i>F</i> ₆	<i>F</i> ₇	<i>F</i> ₈	<i>F</i> ₉	<i>F</i> ₁₀	<i>F</i> ₁₁	<i>F</i> ₁₂	<i>F</i> ₁₃	<i>F</i> ₁₄	<i>F</i> ₁₅
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

- Vegyük észre, hogy az *F_i* függvények sorszáma a megfelelő oszlop felülről lefelé bináris számként összeolvasott számértékének felel meg! Például: *F***5** = 0 1 0 1, vagy *F***13** = 1 1 0 1.

Logikai kapuk

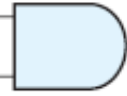
- Az előző oldalon felsorolt 16 függvény közül elsősorban az alábbi nyolc használatos a digitális tervezés gyakorlatában

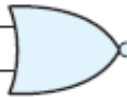
AND ÉS		$F = x \cdot y$	x	y	F
			0	0	0
			0	1	0
			1	0	0
			1	1	1

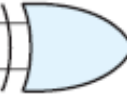
OR VAGY		$F = x + y$	x	y	F
			0	0	0
			0	1	1
			1	0	1
			1	1	1

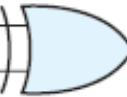
Inverter		$F = x'$	x	F
			0	1
			1	0

Buffer		$F = x$	x	F
			0	0
			1	1

NAND nem-ÉS		$F = (xy)'$	x	y	F
			0	0	1
			0	1	1
			1	0	1
			1	1	0

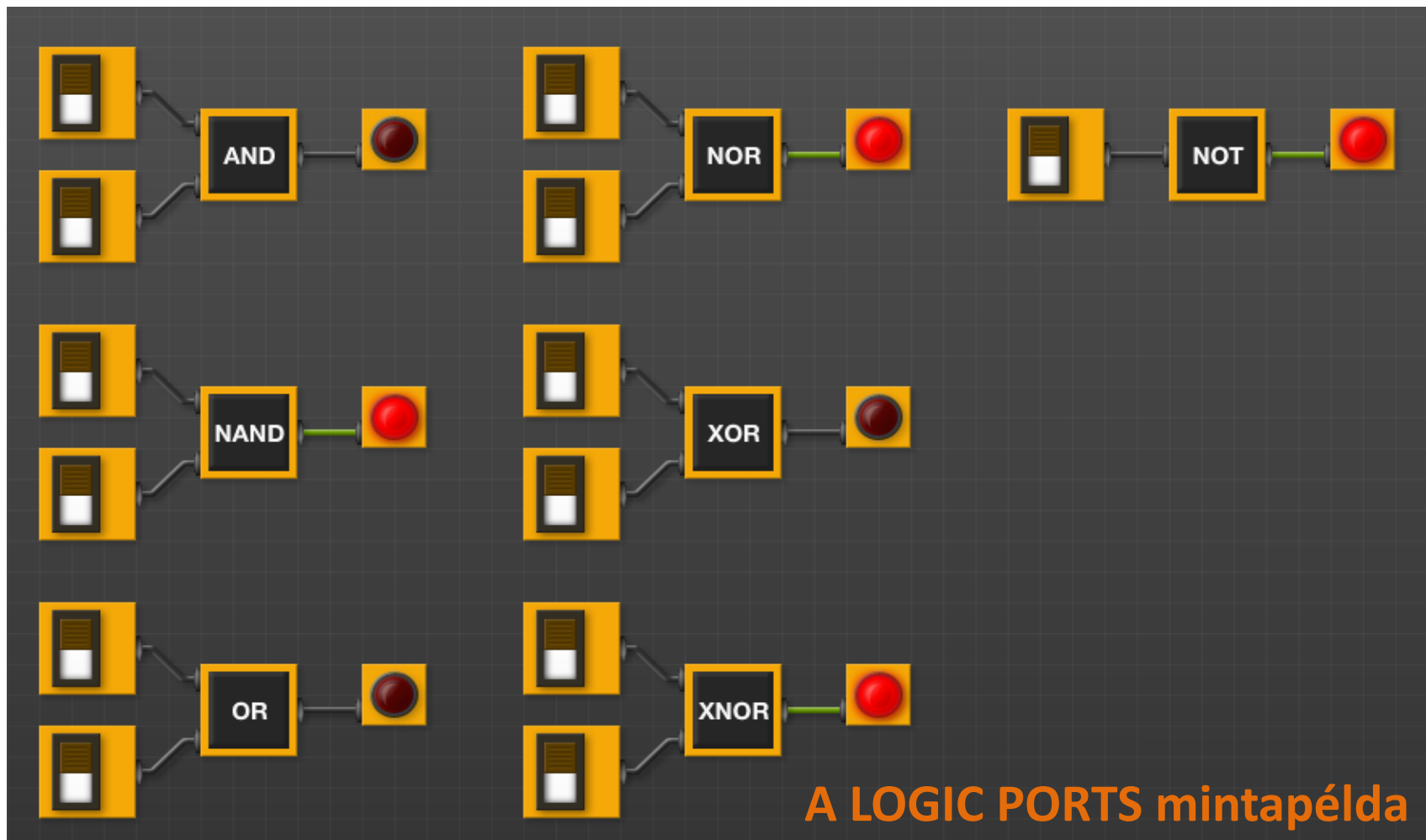
NOR nem-VAGY		$F = (x + y)'$	x	y	F
			0	0	1
			0	1	0
			1	0	0
			1	1	0

XOR kizáró VAGY		$F = xy' + x'y$ $= x \oplus y$	x	y	F
			0	0	0
			0	1	1
			1	0	1
			1	1	0

equivalence ekvivalencia		$F = xy + x'y'$ $= (x \oplus y)'$	x	y	F
			0	0	1
			0	1	0
			1	0	0
			1	1	1

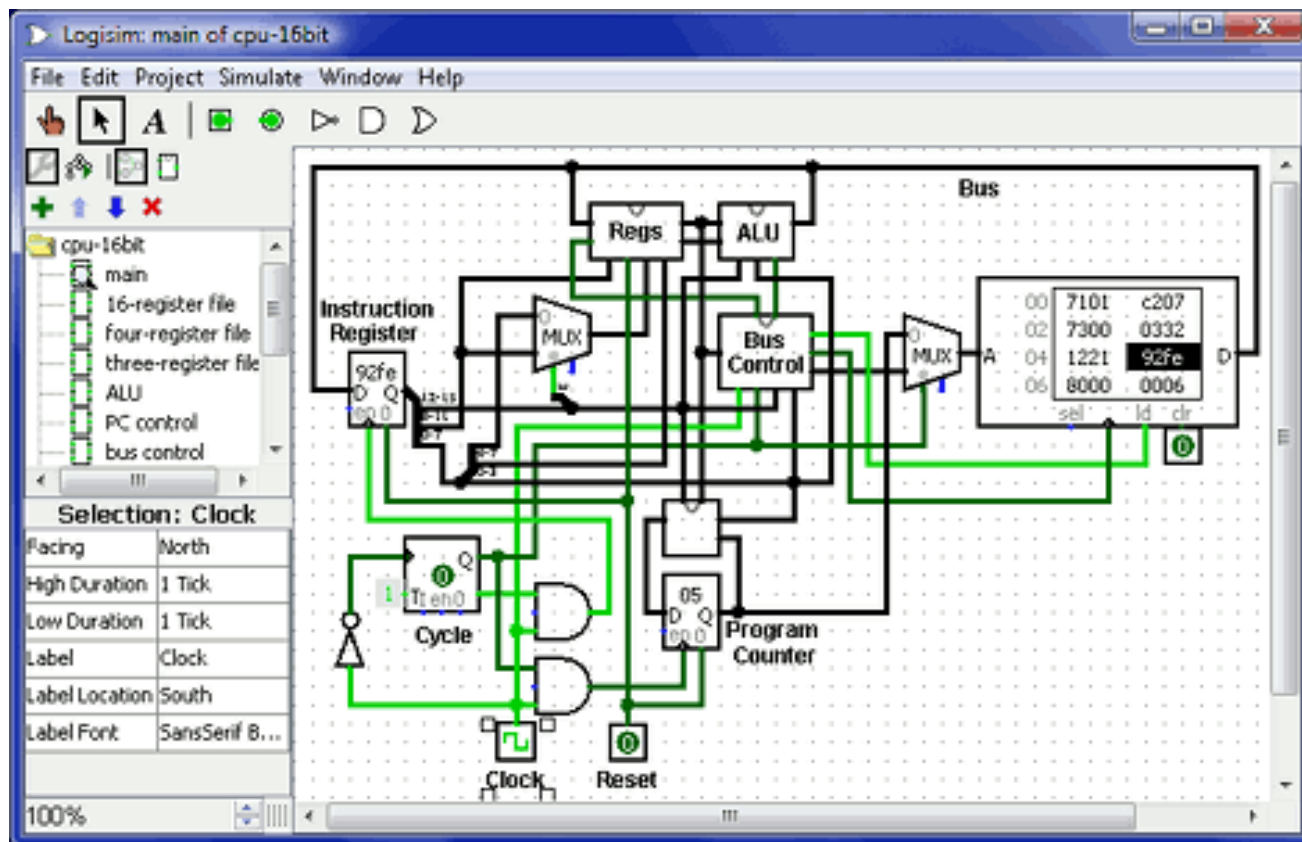
Szimuláció: The Logic Lab

- A logikai áramkörök kipróbálását a THE LOGIC LAB (Logikai Laboratórium) segítségével is végezhetjük! **(Flashplayer szükséges!)**
Link: <http://www.neuroproductions.be/logic-lab/>

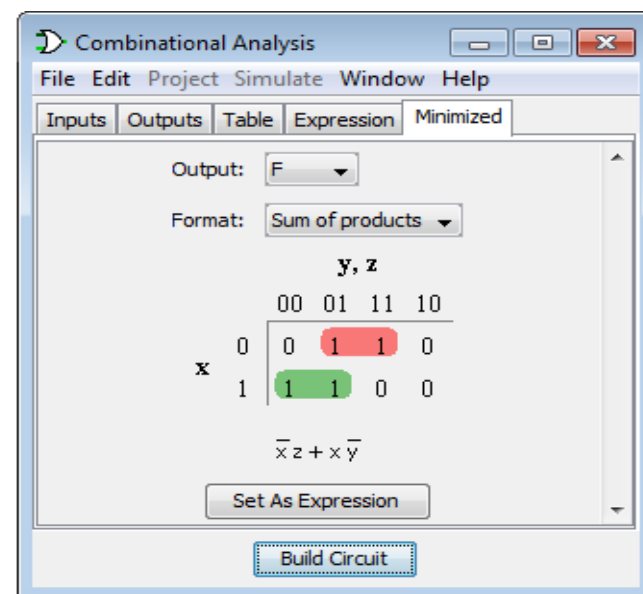
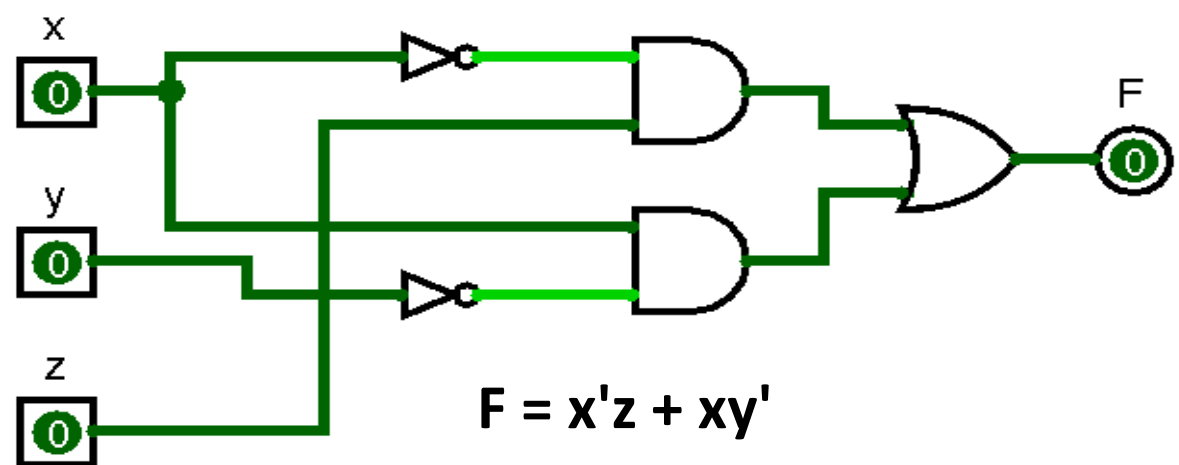
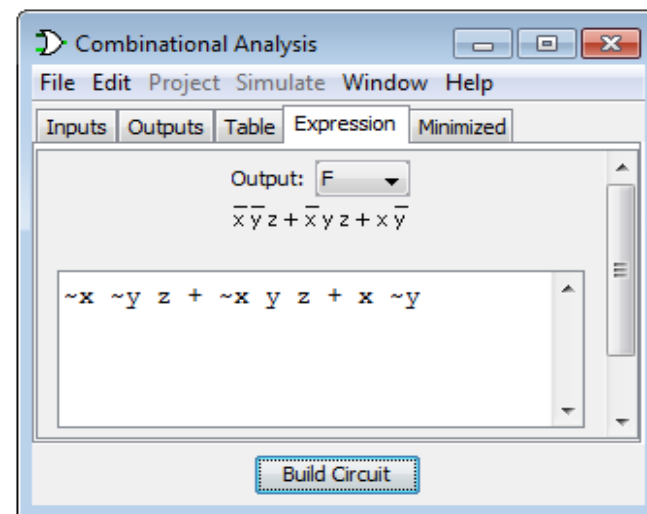
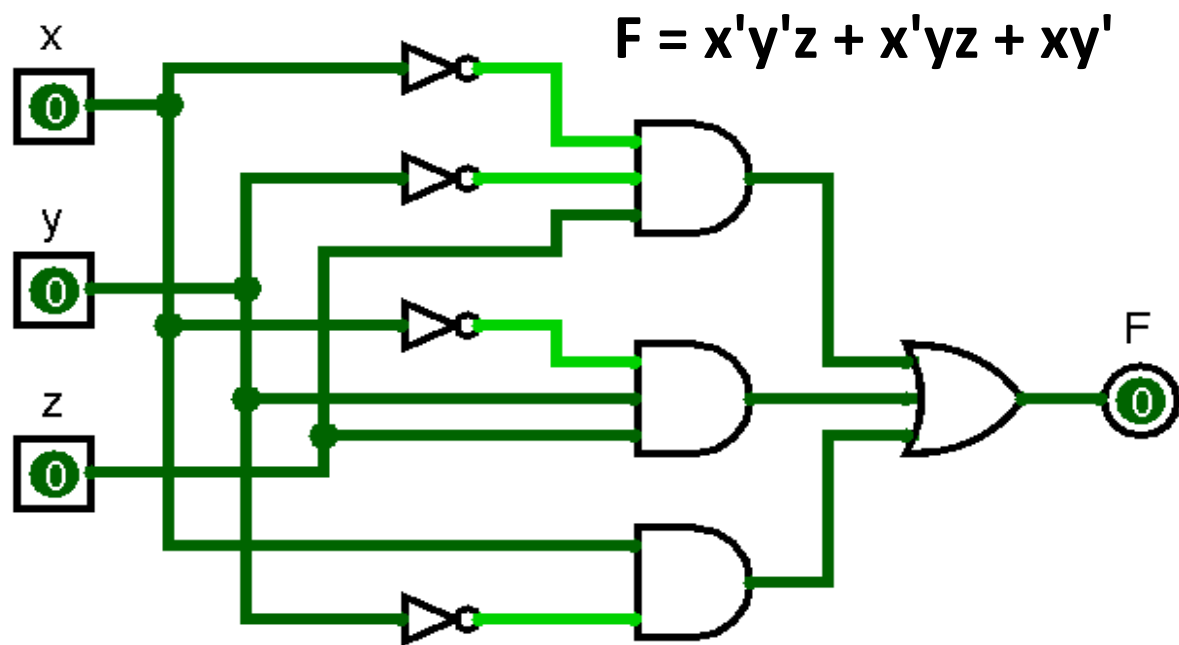


Szimuláció és tervezés: LogiSim

- A Logisim program egy oktatási segédeszköz digitális logikai áramkörök tervezéséhez és szimulációjához (Java kell hozzá!)



A 9 – 10. oldal példája LogiSim-ben



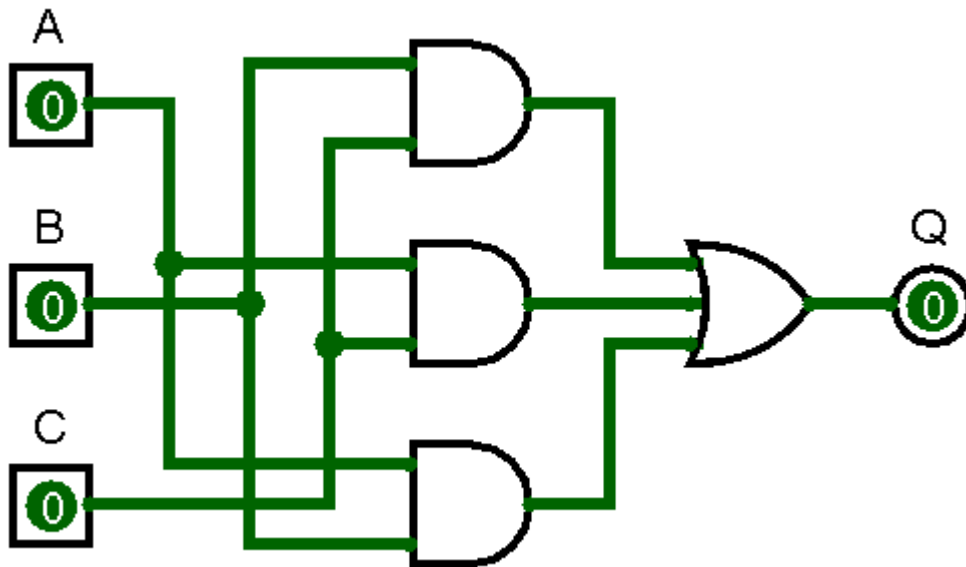
Többségi szavazat kiértékelő áramkör

- Tervezzünk olyan áramkört, melynek kimenete 1, ha legalább két bemenete 1 (igen) állapotban van!

- Logikai függvénnnyel:

$$Q = A \cdot B + A \cdot C + B \cdot C$$

- Logikai hálózattal:



- Használjuk a feladathoz a LogiSim programot!

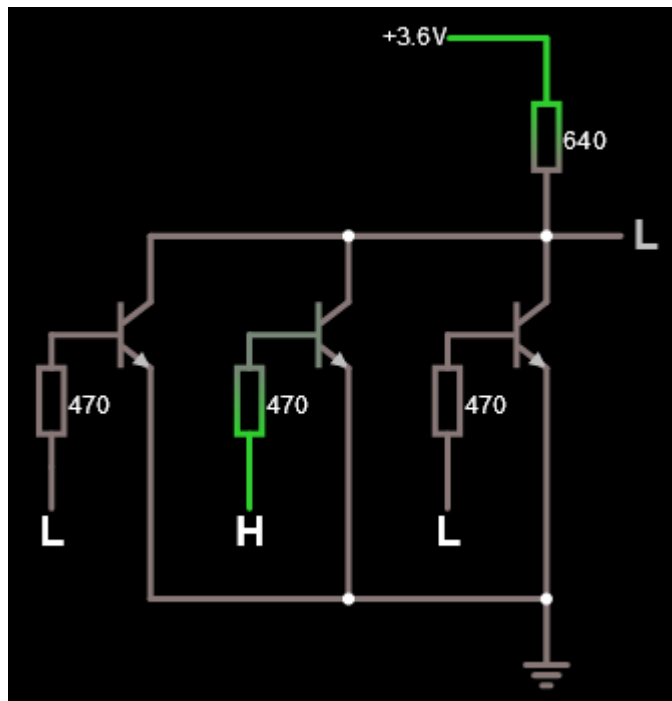
Igazságtáblával:

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

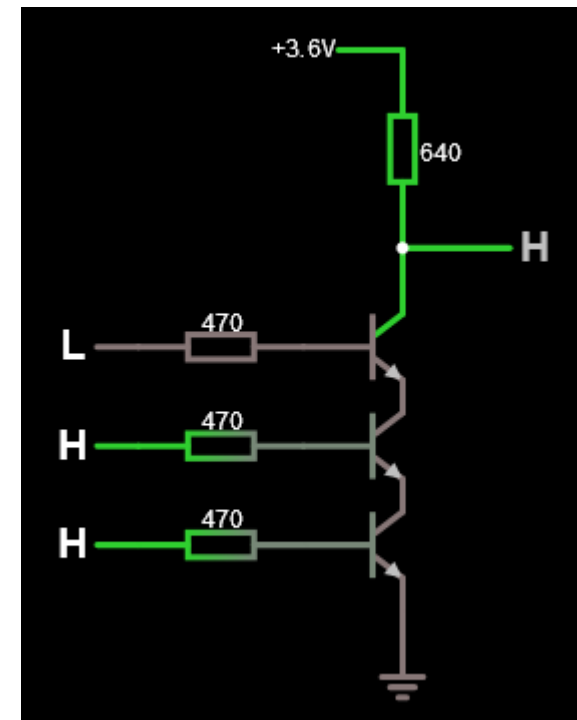
P. Falstad: Circuit Simulator

- Paul Falstad programja nemcsak logikai, hanem analóg áramkörök szimulálására is jól használható. Link: falstad.com/circuit/

A program számos beépített mintapéldát tartalmaz



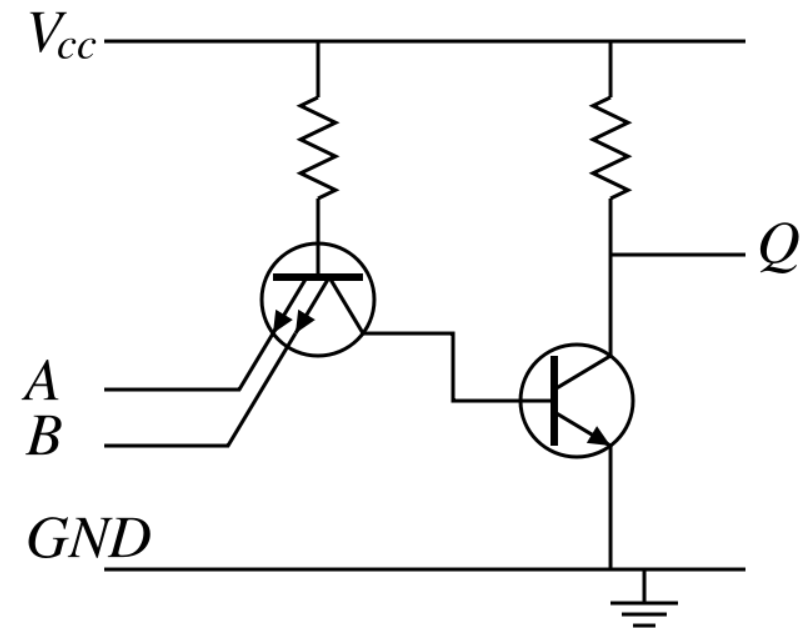
Circuits/Logic families/RTL/RTL_NOR



Circuits/Logic families/RTL/RTL_NAND

TTL Logikai kapuáramkörök

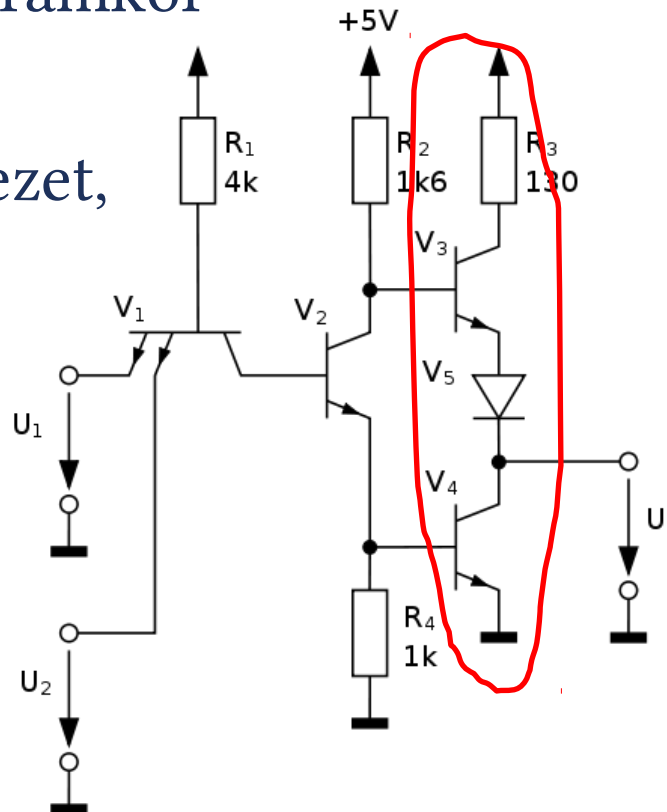
- A TTL (Transistor-Transistor Logic) kapcsolásban bipoláris tranzisztorok vannak, s a logikai bemenetek többemitteres tranzisztorok emitter kivezetései.
- A több emitterrel rendelkező tranzisztorok működése egyenértékű azzal, ha több tranzisztor bázisát és kollektorát összekötjük.
- Az ábrán látható kétbemenetű NEM-ÉS (NAND) áramkör ebben a leegyszerűsített formában nem mentes az RTL áramkörök ismert problémáitól.
- A probléma a következő oldalon bemutatott totem-pole kimenettel orvosolható.



Két bemenetű NAND áramkör

TTL kapu totem-pole kimenettel

- Totem-pole = totem oszlop. Fából faragott oszlop, figurái egymás fölött helyezkednek el, „egymás fején ülnek”... A TTL kimenet elnevezése onnan ered, hogy itt is egymás tetejére ültetett alkatrészeket látunk (az ábrán V3, V5 és V4).
- Kétbemenetű TTL NAND áramkör totem-pole kimenettel
- Alaphelyzetben V2 és V4 vezet, a kimenet alacsony szintű.
- Ha valamelyik bemenetet lehúzzuk, V2 és V4 nem vezet, V3 vezet: magas kimeneti szint.

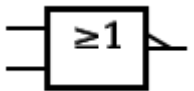


TTL NOR kapu szimulációja

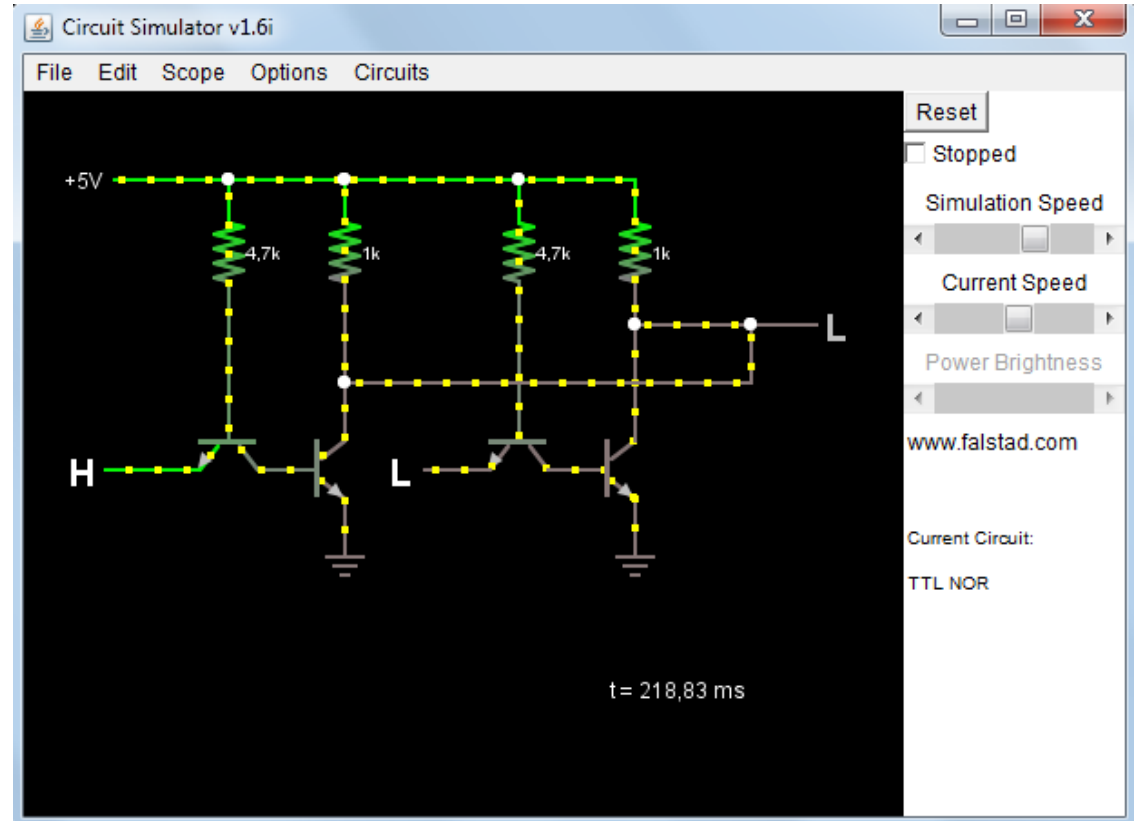
- P. Falstad **Circuit Simulator** programjával vizsgáljuk meg a Circuits/Logic Families/TTL/TTL_NOR példa működését!
- Itt most egy kétbemenetű NOR kaput láthatunk. A kimenet csak akkor magas, ha minden bemenet alacsony szinten van



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



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



2 bemenetű TTL NOR (NEM-VAGY)

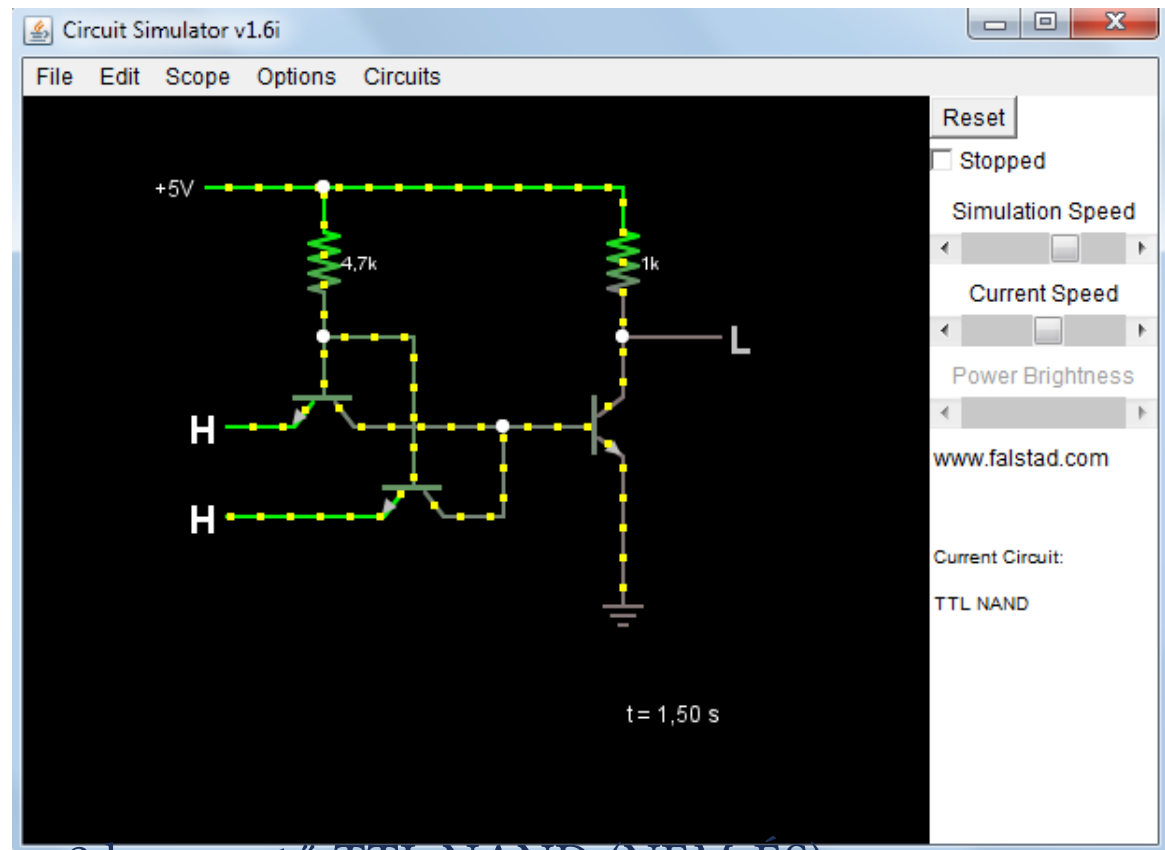
TTL NAND kapu szimulációja

- Most a Circuits/Logic Families/TTL/TTL NAND mintapélda működését vizsgáljuk meg!
- A kimenet csak akkor alacsony, ha minden bemenet magas



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

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

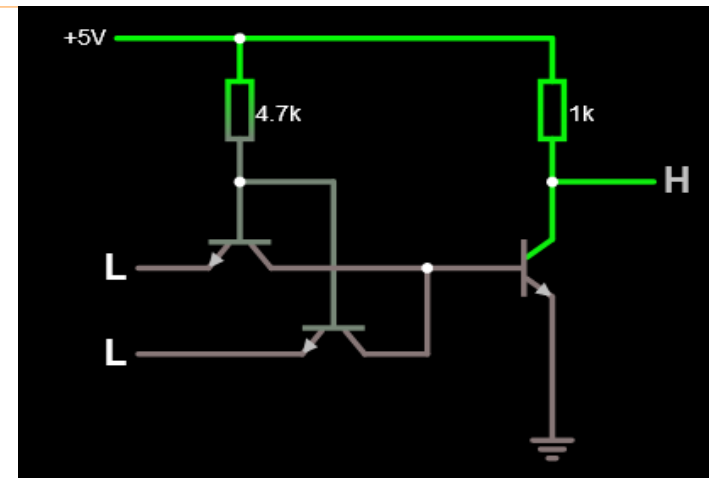


2 bemenetű TTL NAND (NEM-ÉS)

Építsük meg a TTL NAND kaput!

- ❖ 3 db BC 337 NPN tranzisztor
- ❖ 1 db LED
- ❖ 2 db 1 k Ω ellenállás
- ❖ 1 db 4,7 k Ω ellenállás
- ❖ 2 db nyomógomb

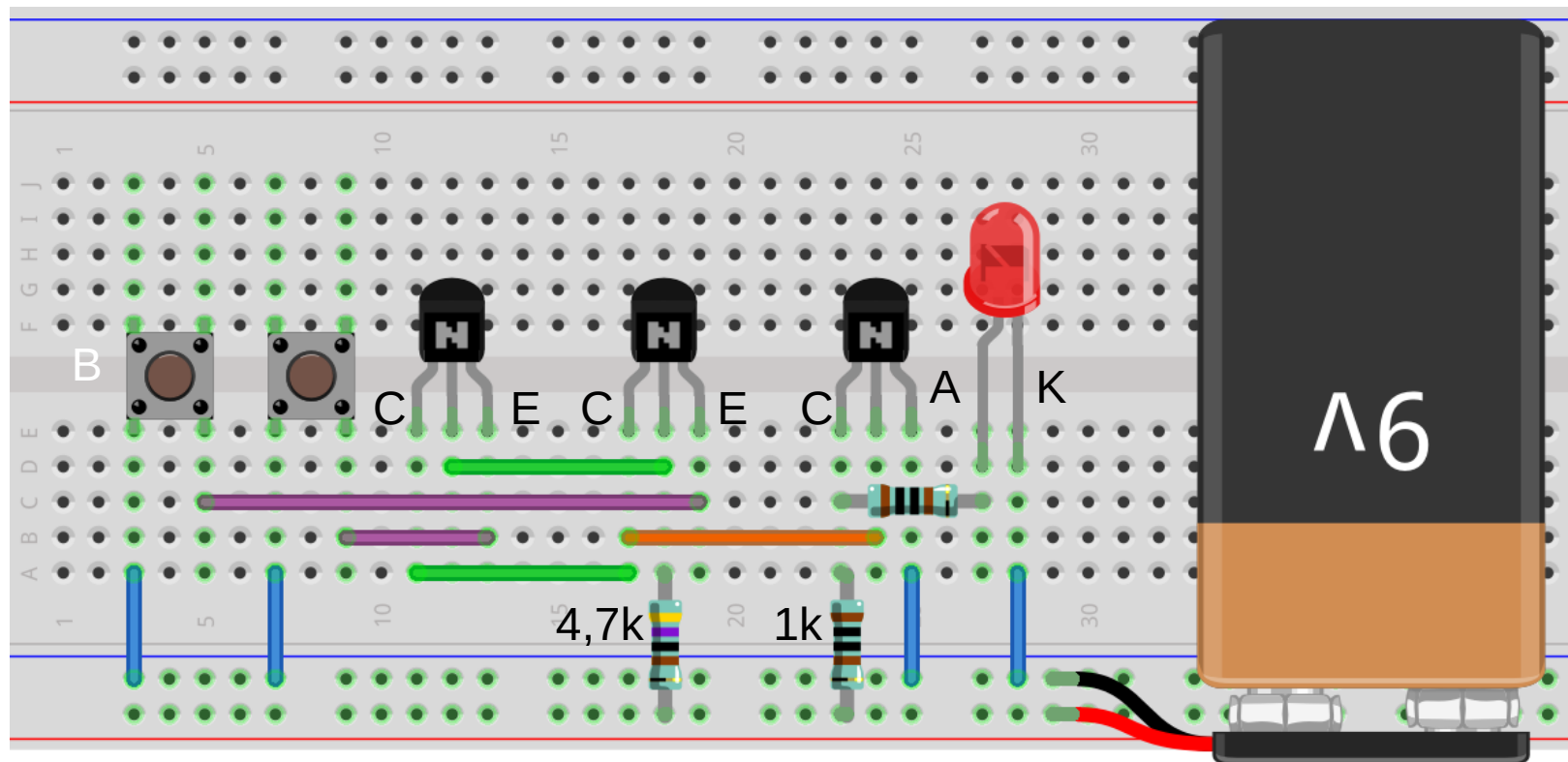
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



Gomblenyomás
alacsony szint (0),

Gomb
elengedéskor
magas szint (1)

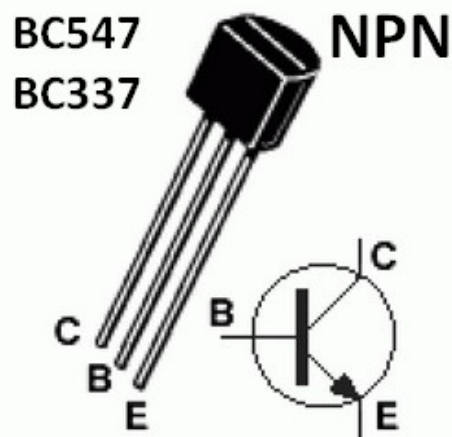
A LED ég, ha a
kimeneten magas
szint van (1)



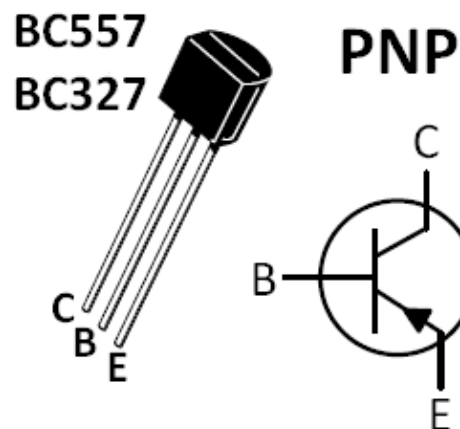
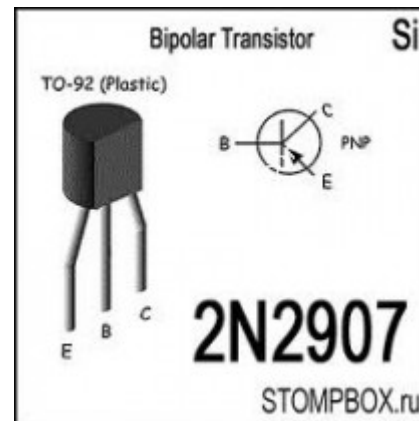
Tranzisztorok lábkiosztása

Ügyeljünk a polaritásra és a tokozás eltéréseire!

NPN tranzisztorok



PNP tranzisztorok



Ellenállás színkódok

