**Pannon Egyetem**
**Villamosmérnöki és Információs Rendszerek Tanszék**

# Digitális Technika

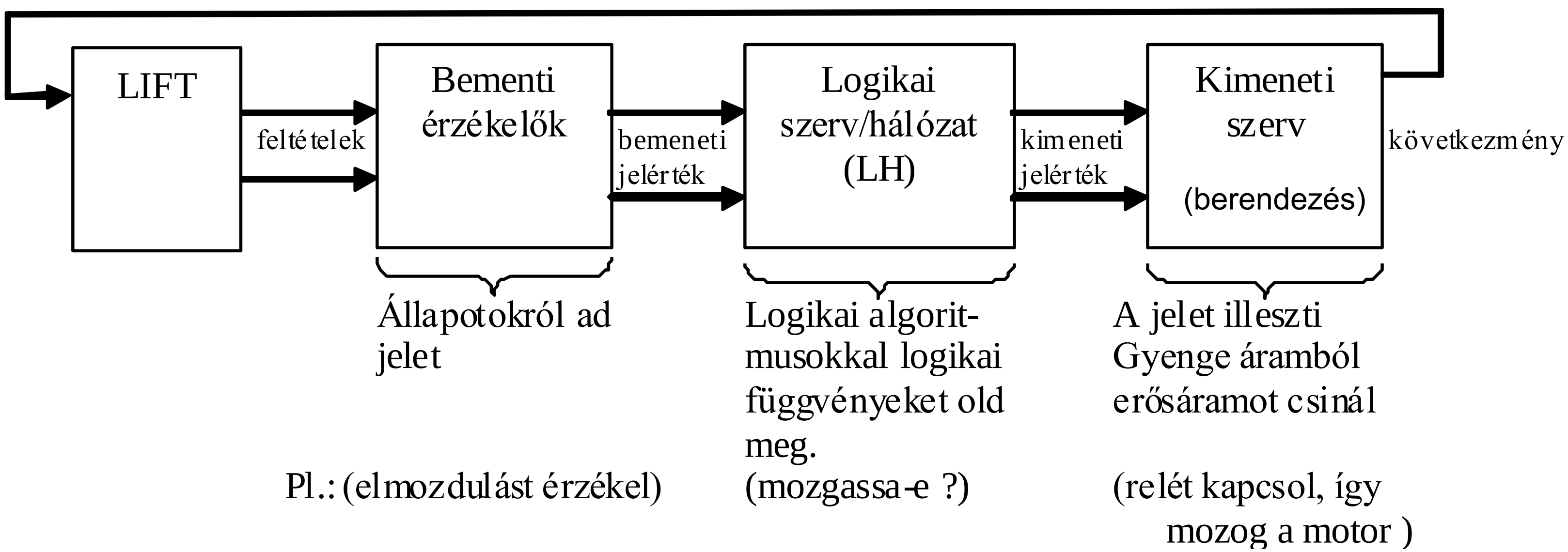
Boole algebra (függvény, igazságtábla, kanonikus alak).
Kombinációs Hálózatok tervezése

# Logikai tervezés alapjai

# A logikai tervezés kritériumai

- Tervdokumentációt (tervezési specifikációt) kell készíteni a hálózatról
  - Egységes, szabványos jelölésrendszer szükséges
- A berendezésnek könnyen vizsgálhatónak, megbízhatónak kell lennie
  - Tagolt legyen (könnyen átlátható, javítható)
  - Moduláris, strukturált felépítés: hiba esetén csak adott modult kelljen cserélni
- Gyárthatónak kell lennie
  - Mérőpontokat, tesztelő helyeket kell kialakítani
  - Építőelem készletet figyelembe kell venni
- Gazdaságosság: lehetőleg minimális építőelem felhasználás

# Példa: LIFT vezérlése




**Liftmotor mozgatás**

Ebben a feladatban azt definiáljuk (az úgynevezett igazságtábla segítségével), hogy adott bemeneti értékek mellett mikor induljon el a lift. Jelen esetben az igazságtáblánknak $2^4=16$ sora kell, hogy legyen, mert 4 bemeneti változónk van.

- Bemeneti változó:
  - nyomtak-e gombot (GNY)
  - liftajtó be van-e csukva (LZ)
  - van-e valaki a liftben (VL)
  - túlterhelt-e (TT)
- Kimeneti változó:
  - Bekapcsolja-e a motort (0 v. 1)

# Igazságtábla: Lift vezérlése

| Automatikus feltételek | | | | következmények | |
|---|---|---|---|---|---|
| **Gombot nyomtak** | **Liftajtó Zárva** | **Valaki Liftben** | **Túlterhe lt** | **Liftmotor** | |
| 0 | 0 | 0 | 0 | 0 | nem indul |
| 0 | 0 | 1 | 0 | 0 | nem indul |
| 1 | 0 | 1 | 0 | 0 | nem indul |
| 1 | 1 | 1 | 0 | 1 | indul |
| 1 | 1 | 1 | 1 | 0 | nem indul |
| 1 | 1 | 0 | 1 | - | NTSH* |

NTSH*: Nem Teljesen Specifikált Hálózat, don’t care „közömbös ‘állapotokkal’

# Logikai hálózatok csoportosítása a megoldandó logikai feladatok szerint

- **Logikai hálózat:** ált. készen kapható logikai építőelemekből állítható össze
- **Logikai rendszer**: logikai hálózatoknak egy adott feladat megoldása céljából együttműködő összességét logikai rendszernek nevezzük.

- Mivel a logikai feladat megoldása szempontjából, nem mindig elég a mindenkori fennálló pillanatnyi bemeneti kombinációk figyelembevétele a kimeneti kombinációk előállításához, szükségünk lehet ***másodlagos*** (ún. ***szekunder*****) feltételeknek** megfelelő szekunder kombinációkra is.

  - **Pl: a lift vezérlése:** nem elég, ha a liftmotor a feltételek teljesülése után, csak úgy elindul, mert nem mindegy *melyik irányba* forog (fel v. le). Hiszen, ha az 1.-ről akarunk eljutni a 3.-ra, akkor más irányba kell forognia, mintha az 5.-ről akarnánk eljutni a 3.-ra. Tehát a szekunder változó ebben az esetben a *felvonó mindenkori helyzetéről* ad felvilágosítást érzékelők segítségével. (A szekunder változó feladata a hálózat belső állapotának figyelése.)

# Logikai hálózatok csoportosítása

Ezek alapján kétféle hálózatot különböztetünk meg

- **(K.H.) Kombinációs logikai hálózat**ról beszélünk: ha a mindenkori kimeneti kombinációk létrehozásához *elég a bemeneti kombinációk* pillanatnyi értéke

- **(S.H.) Sorrendi (szekvenciális) logikai hálózat**ról beszélünk**:** ha a mindenkori kimeneti kombinációt, nemcsak *a pillanatnyi* bemeneti kombináció, hanem *a korábban fennállt bementi kombinációk* és azok sorrendje is befolyásolja. (A *szekunder kombinációk* segítségével az ilyen hálózatok képessé vállnak arra, hogy az ugyanolyan bemeneti kombinációkhoz **más-más kimeneti** kombinációt szolgáltassanak, attól függően, hogy a bemeneti kombináció fellépésekor, milyen értékű a szekunder kombináció)

# Logikai változók

- A logikai változók az egyes események absztrakt leírására szolgálnak. Két értéket vehetnek fel:
- IGAZ vagy HAMIS, attól függően, hogy az esemény bekövetkezik vagy sem. Ha az esemény bekövetkezik, akkor a logikai változó értéke IGAZ. Ha az esemény nem következik be, akkor a logikai változó értéke HAMIS.
- Értékkészlet, jelölések:

| 1 | 0 |
|---|---|
| IGAZ (I) | HAMIS (H) |
| TRUE (T) | FALSE (F) |
| HIGH (H) | LOW (L) |

- Az 1 és 0 itt nem számjegy, nincs numerikus értékük. Jelentésük szimbolikus.
- A HIGH/LOW jelentése a logikai értékek elektromos megjelenéséhez kapcsolódik, alacsony és magas feszültségszintnek felel meg.

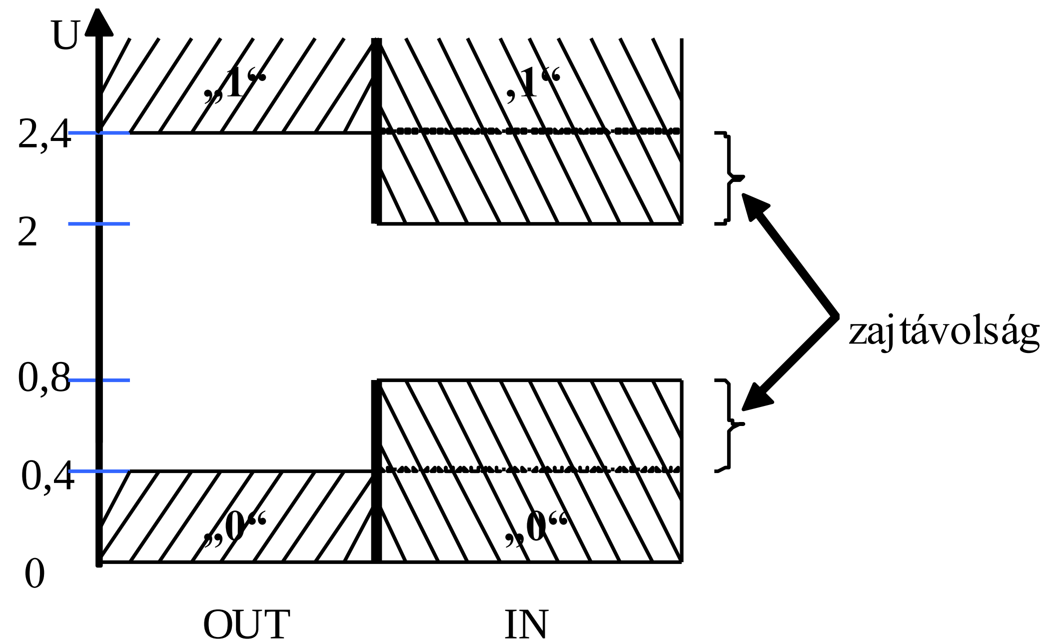
# LOGIKAI VÁLTOZÓK A GYAKORLATBAN

- A két legelterjedtebb logikai áramkörcsalád:
  - CMOS (Complementary Metal Oxide Semiconductor)
  - bipoláris technológia, TTL (Transistor Transistor Logic),
- Az alacsony logikai szint névlegesen 0 Volt,
- A magas logikai szint a pozitív tápfeszültség által meghatározott feszültség.
- CMOS:
  - $U(1) = U_{táp} = +3 \ldots +15$ V, $U(0) = 0$ V
- TTL:
  - $U(1) \approx +3{,}5$ V, $U_{táp} = +5$ V, $U(0) = 0$ V

# Digitális jelértékek, Zajtávolság: standard 0-5V feszültség tartomány esetén,TTL

**Feszültség**

0-5 V-os tartományban
„0“ = 0-0,8V-ig
„1“ = 2-5 V-ig
0,8-2-ig nem értelmezzük




A zajtávolság azért kell, mert ha a feszültség pont a határon van akkor a különféle külső környezeti hatások (zajok) miatt rossz értéket kaphatunk.
Ilyen zajok Pl.:
a termikus zajok -- a vezeték hőmozgása;
az induktív zajok -- külső elektromágneses tér, aminek a hatása akár pár V is lehet az eltérés;

# Logikai operátorok és igazságtáblájuk

# Logikai operátorok

- Fajtáik:
  - Egy-változós
  - Kettő-, vagy több-változós
- Három alapművelet:
  - NOT (NEM)
  - AND (ÉS)
  - OR (VAGY)
- **Funkcionális / Univerzális teljesség:** bizonyos logikai műveletek használatával bármely tetszőleges más logikai függvény megadható.
  - ilyenek a NAND, NOR kapuk!

# Igazságtábla: logikai operátorok felírása

- ‘X’ logikai függvény megadása az ‘A,B,C’ logikai változók összes lehetséges értékétől függően
  Jel: X(A, B,C,)  //3 változó -> 2^3 = 8 sor//

| A | B | C | X |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

$\Leftrightarrow$

| A | B | C | X | **sor** |
|---|---|---|---|---|
| F | F | F | F | **0.** |
| F | F | T | T | **1.** |
| F | T | F | T | **2.** |
| F | T | T | F | **3.** |
| T | F | F | T | **4.** |
| T | F | T | F | **5.** |
| T | T | F | F | **6.** |
| T | T | T | F | **7.** |

- Kanonikus „standard” igazság tábla:
  000 – 111 -ig (3 változó esetén)

# Logikai operátorok és igazság táblázatuk (NOT)

- Jel: $\text{NOT A} = \overline{A}$
- Formális definíció igazságtáblával:

| A | NOT A |
|---|---|
| 0 | 1 |
| 1 | 0 |

- Def:
  - ha A hamis, NOT A igaz
  - ha A igaz, NOT A hamis

# Logikai operátorok és igazság táblázatuk (AND)

- Jel: B **AND** C = B·C
- Formális definíció igazságtáblával:

| B | C | B·C |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

- Def: B·C értéke pontosan akkor ‘igaz’ ha ‘B’ és ‘C’ is egyszerre ‘igaz’, különben hamis
- kommutatív

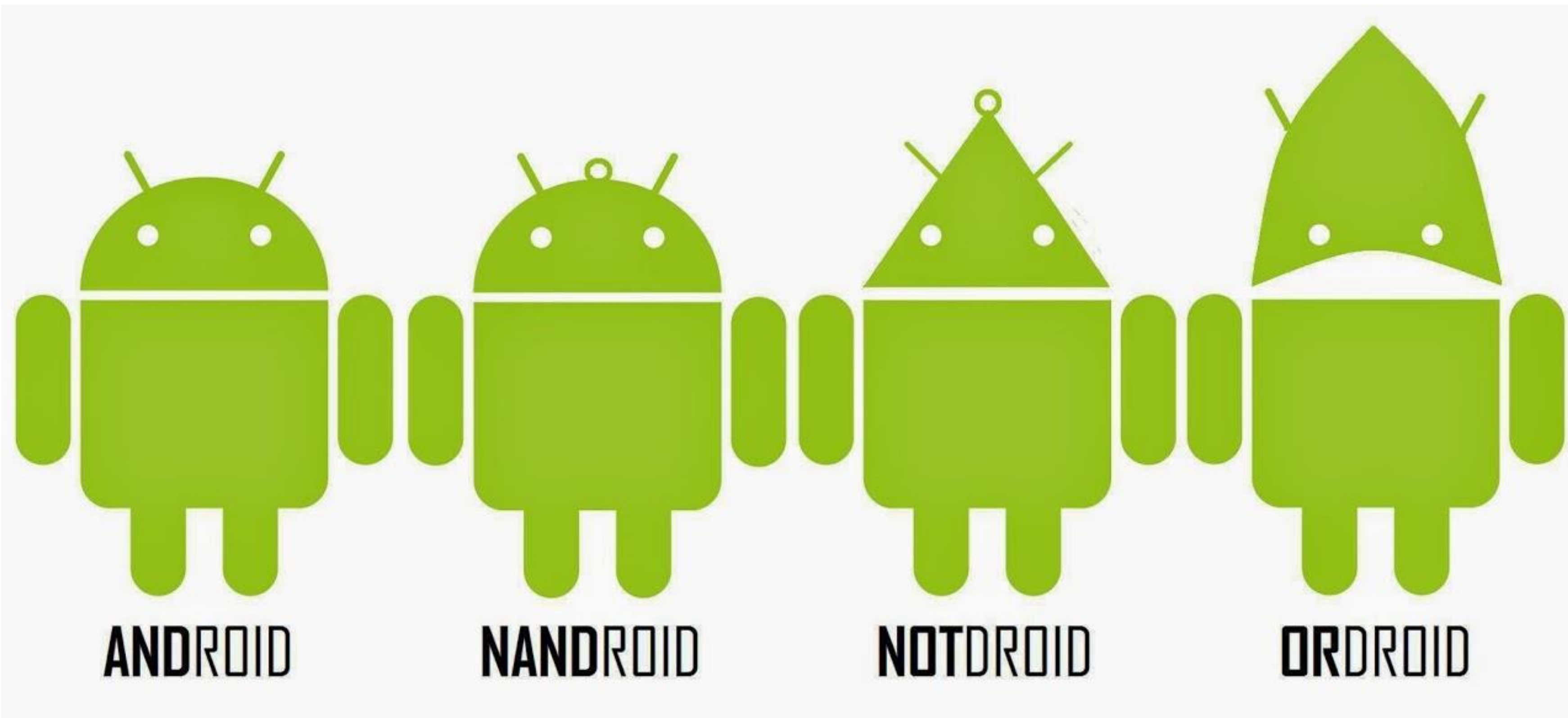
# Logikai operátorok és igazság táblázatuk (OR)

- Jel: B **OR** C = B**+**C
- Formális definíció igazságtáblával:

| B | C | B+C |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

- Def: B**+**C értéke pontosan akkor ‘igaz’ ,ha ‘B’ és ‘C’ közül legalább az egyik ‘igaz’, különben hamis
- kommutatív

# Smart áramkörök ☺

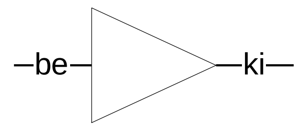


# Egy- ill. két-változós logikai függvények bemutatása és szabványos jelöléseik

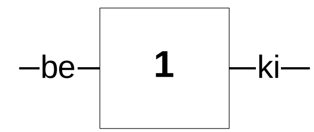
# Egyváltozós logikai függvények:

- Jelmásoló („buffer” - jel-erősítő):

| be | ki |
|---|---|
| 0 | 0 |
| 1 | 1 |



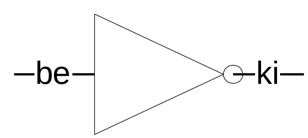

Nemzetközi szabvány

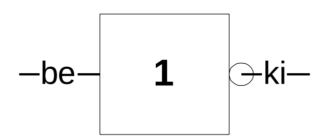



Magyar szabvány

- Negálás - Inverter (NOT): $\overline{A}$

| be | ki |
|---|---|
| 0 | 1 |
| 1 | 0 |



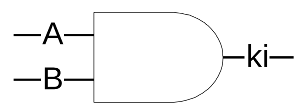



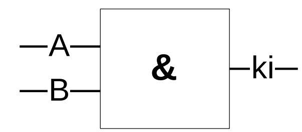
# Kétváltozós logikai függvények:

- ÉS (AND):

$$A \cdot B$$

| A | B | ki |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

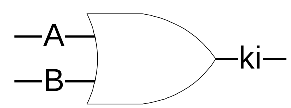


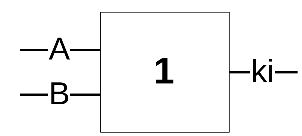




- VAGY (OR):

$$A + B$$

| A | B | ki |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

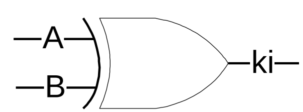


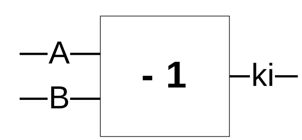




- Antivalencia (XOR):

$$A \oplus B$$

| A | B | ki |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



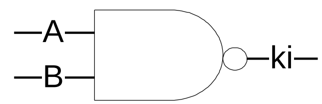



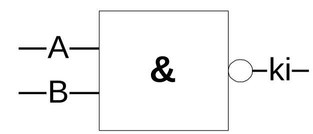
# Kétváltozós log.függv. (folyt.):

- NEM-ÉS (NAND):

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

| A | B | ki |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

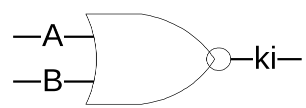


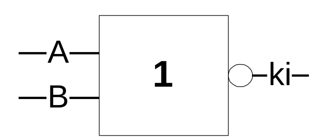




- NEM-VAGY (NOR):

$$\overline{A + B}$$

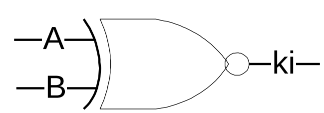
| A | B | ki |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

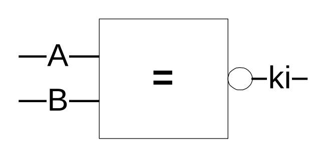



- Ekvivalencia (NXOR):
- $A \odot B$

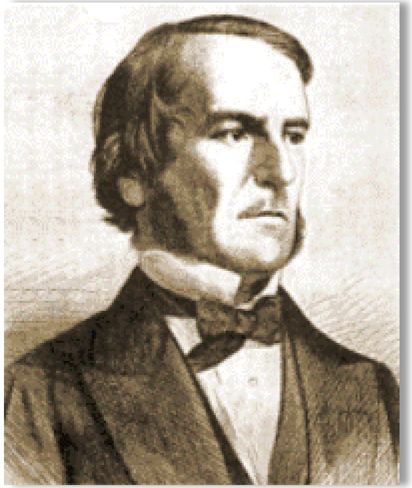
| A | B | ki |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



# Boole algebra

# Boole-algebra



(1815-1864)

- Logikai operátorok algebrája
- George Boole: először mutatott **hasonlóságot** az általa vizsgált ***logikai operátorok*** és a már jól ismert ***aritmetikai operátorok*** **között**.

# Boole algebra elemei:

- 3 alapművelet: AND, OR, NOT
- Tulajdonságaik (AND, OR esetén):
  - Kommutatív: $A+B=B+A$, $A \cdot B=B \cdot A$
  - Asszociatív: $A+(B+C)=(A+B)+C=A+B+C$
    $A \cdot (B \cdot C)=(A \cdot B) \cdot C=A \cdot B \cdot C$
  - Disztributív: $A \cdot (B+C)=A \cdot B+A \cdot C$,
    $A+(B \cdot C)=(A+B) \cdot (A+C)$
- Operátor **precedencia** (csökkenő sorrendben):
  - NOT
  - AND
  - OR
  - átzárójelezhetőség!

# Boole algebrai azonosságok!

$$1.)\ \overline{\overline{A}} = A$$

NOT

$$2.)\ A + 0 = A$$

$$3.)\ A + 1 = 1$$

$$4.)\ A + A = A$$

$$5.)\ A + \overline{A} = 1$$

OR

$$6.)\ A \cdot 1 = A$$

$$7.)\ A \cdot 0 = 0$$

$$8.)\ A \cdot A = A$$

$$9.)\ A \cdot \overline{A} = 0$$

AND

$$10.)\ A + A \cdot B = A \quad *$$

$$11.)\ A \cdot (A + B) = A \quad *$$

Elnyelési tul.

$$12.)\ A \cdot B + A \cdot \overline{B} = A$$

$$13.)\ (A + B) \cdot (A + \overline{B}) = A$$

$$14.)\ A + \overline{A} \cdot B = A + B \quad *$$

$$15.)\ A \cdot (\overline{A} + B) = A \cdot B$$

## De-Morgan azonosságok:

$$18.)\ \overline{A + B} = \overline{A} \cdot \overline{B}$$

$$19.)\ \overline{A \cdot B} = \overline{A} + \overline{B}$$

DUAL ITÁS

# Boole-algebrai azonosság igazolása igazságtáblával

- Pl: De Morgan

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

Dualitás elve

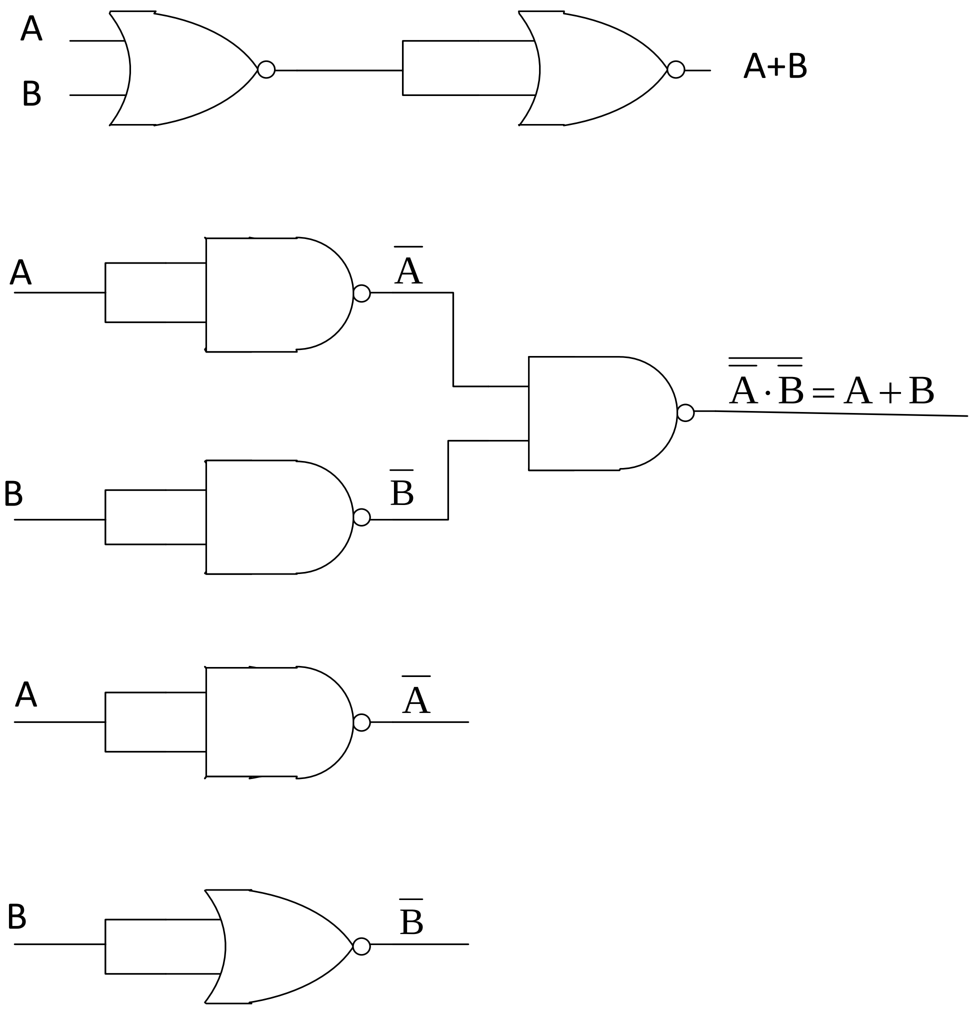
| A | B | A·B | NOT (A·B) |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

| A | B | NOT A | NOT B | NOT(A) + NOT(B) |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |

- Példa: egyszerűsítésre

$$\overline{A \cdot (B + C \cdot (B + \overline{A}))} \stackrel{?}{=} \overline{A} + \overline{B}$$

# Funkcionális teljesség: példák




**Funkcionálisan teljes, vagy univerzális áramköri alapkapuk:** VLSI CMOS **l**ogikai hálózatok esetén a **NAND**, illetve **NOR** kapu.

# Logikai egyenletek megadása igazság-táblázatokból

- Pl:

| sor | A | B | W |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 1 | 0 |

W pontosan akkor lesz **igaz**, ha A igaz és B hamis, egyébként W **hamis** lesz. Vagyis egyenletként kifejezve:

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

- Pl:

| sor | A | B | Y |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 1 | 1 |

Y pontosan akkor lesz **igaz**, ha A és B is hamis, vagy A igaz és B hamis, vagy A és B is igaz, egyébként W **hamis** lesz. Vagyis egyenletként kifejezve:

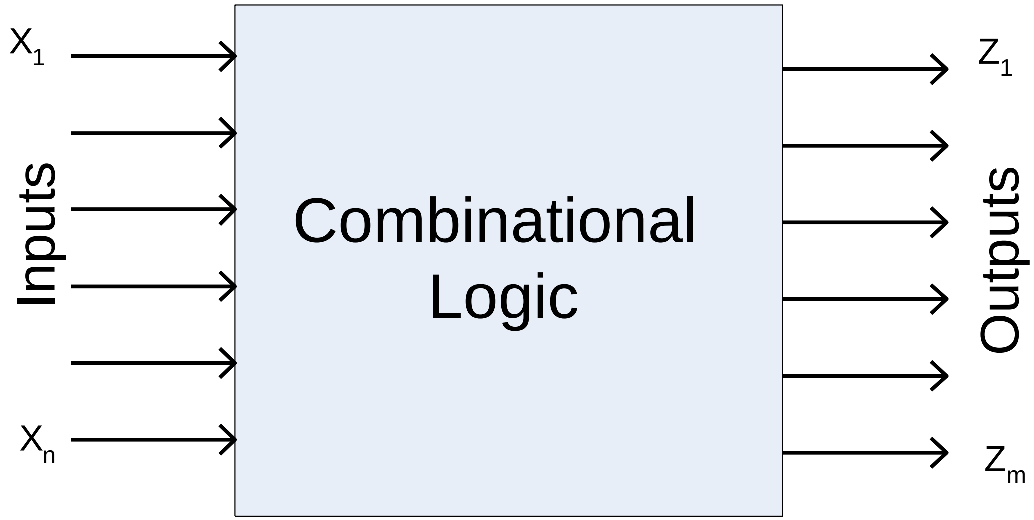
$$Y = \overline{A} \cdot \overline{B} + A \cdot \overline{B} + A \cdot B \Rightarrow \overline{B} + A \cdot B \Rightarrow \overline{B} + A$$

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

# Def: Logikai függvény

- Független változó: bemeneti változó(k)
- Függő változó: kimeneti változó(k)
- **Logikai függvény**:
  - minden kimeneti változó értéke a *<u>függvénykapcsolat</u>* alapján határozható meg (minden egyes bemeneti kombinációhoz meghatározható a kimeneti változó).
  - Egyértelmű hozzárendelés (de nem kölcsönösen egyértelmű!)

# Teljesen határozott logikai hálózat hány különböző függvénnyel írható le összesen?



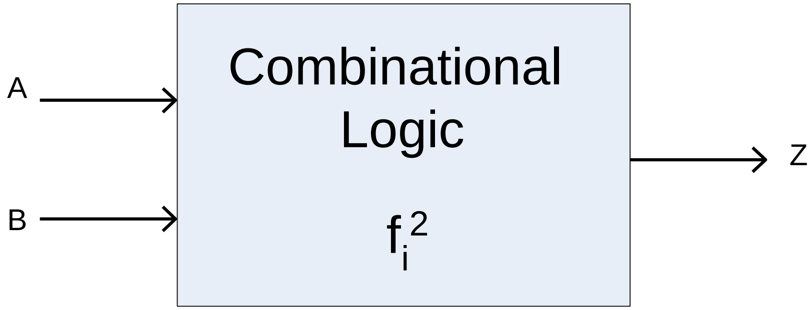

$N_{Be} = 2^n$ — $N_{Ki} = 2^m$

bemeneti változó | bemeneti változó kombináció | kimeneti változó kombináció | kimeneti változó

$$N_f = N_K^{N_B} = (2^m)^{2^n}$$

**PÉLDA: 4 bemenet, 2 kimenet esetén a függvények lehetséges száma:**
**$N_f$** $=(2^2)$^$(2^4)=4^{16}$ logikai függvényt lehet definiálni (teljesen határozott LH)

# Pl. Összes lehetséges két-bemenetű / egy-kimenetű logikai függvény igazságtáblája




$$N_f = N_K^{N_B} = (2^m)^{2^n}$$

$$= (2^1)^{2^2} = 2^4 = 16$$

| A | B | $f_0^2$ | $f_1^2$ | $f_2^2$ | $f_3^2$ | $f_4^2$ | $f_5^2$ | $f_6^2$ | $f_7^2$ | $f_8^2$ | $f_9^2$ | $f_{10}^2$ | $f_{11}^2$ | $f_{12}^2$ | $f_{13}^2$ | $f_{14}^2$ | $f_{15}^2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

Egymás negáltjai/komplemensei

$$Z = f_0^2(A,B) \equiv 0$$

$$Z = f_1^2(A,B) = A \cdot B$$

$$Z = f_2^2(A,B) = A \cdot \overline{B}$$

$$Z = f_3^2(A,B) = A \cdot \overline{B} + A \cdot B \equiv A$$

$$Z = f_4^2(A,B) = \overline{A} \cdot B$$

$$Z = f_5^2(A,B) = \overline{A} \cdot B + A \cdot B \equiv B$$

$$Z = f_6^2(A,B) = \overline{A} \cdot B + A \cdot \overline{B} \equiv A \oplus B$$

$$Z = f_7^2(A,B) = \overline{A} \cdot B + A \cdot \overline{B} + A \cdot B =$$

$$= \overline{A} \cdot B + A \cdot (\overline{B} + B) = \overline{A} \cdot B + A \equiv A + B$$

$$Z = f_8^2(A,B) = \overline{A} \cdot \overline{B} \Leftrightarrow \overline{f_7^2(A,B)}$$

$$Z = f_9^2(A,B) = \overline{A} \cdot \overline{B} + A \cdot B \equiv A \odot B \Leftrightarrow \overline{f_6^2(A,B)}$$

$$Z = f_{10}^2(A,B) = \overline{A} \cdot \overline{B} + A \cdot \overline{B} \equiv \overline{B} \Leftrightarrow \overline{f_5^2(A,B)}$$

$$Z = f_{11}^2(A,B) = \overline{A} \cdot \overline{B} + A \cdot \overline{B} + A \cdot B =$$

$$= \overline{B} \cdot (\overline{A} + A) + A \cdot B = A + \overline{B} \Leftrightarrow \overline{f_4^2(A,B)}$$

$$Z = f_{12}^2(A,B) = \overline{A} \cdot \overline{B} + \overline{A} \cdot B = \overline{A} \cdot (\overline{B} + B) = \overline{A}$$

$$\Leftrightarrow \overline{f_3^2(A,B)}$$

$$Z = f_{13}^2(A,B) = \overline{A} \cdot \overline{B} + \overline{A} \cdot B + A \cdot B =$$

$$= \overline{A} \cdot (\overline{B} + B) + A \cdot B = \overline{A} + B \Leftrightarrow \overline{f_2^2(A,B)}$$

$$Z = f_{14}^2(A,B) = \overline{A} \cdot \overline{B} + \overline{A} \cdot B + A \cdot \overline{B} =$$

$$= \overline{A} \cdot (\overline{B} + B) + A \cdot \overline{B} = \overline{A} + \overline{B} \Leftrightarrow \overline{f_1^2(A,B)}$$

$$Z = f_{15}^2(A,B) \equiv 1 \Leftrightarrow \overline{f_0^2(A,B)}$$

# Logikai függvények kanonikus (normál) alakjai

# 1.) Sum-of-Products (szorzat„termek” összege)

- **Szorzat (AND) termek összege (OR kapcsolata)**
- Emberi szemléletmódhoz közelebb áll: a táblázat soraiból azokat a függvényértékeket (Y) vesszük amelyek ‘1’-esek
- Def: Triviális forma: ha egy változó egy adott szorzat termben vagy ponáltan, vagy negáltan legfeljebb egyszer szerepel.
  Ezt hívják még **mintermnek ($m_i$)** vagy **kanonikus szorzat termnek** is. Tegyük fel hogy F(A,B,C)
  - Pl: *valós / triviális /* kanonikus formulák: $A, \overline{A}, A \cdot B, \overline{A} \cdot B \cdot \overline{C}$
  - Pl: érvénytelen formulák (de ettől még Boole kifejezés), ami jelenti azt is, hogy tovább egyszerűsíthetők:

$$A \cdot \overline{A}, \overline{A} \cdot B \cdot B \cdot \overline{C}$$

# Diszjunktív Normál Forma:

- Jel: $Y(DNF): \sum_{i=0}^{2^n-1} m_i$
- n változó esetén 2^n lehetséges minterm van.
- Képzésük: az igazságtáblázatból azoknak a mintermeknek a VAGY kapcsolatát vesszük, ahol függvényértékek sorában ($Y$) ‘1’ -es szerepel, vagy ahol a függvény komplemensének ($\overline{Y}$) értéke ‘0’.
- minterm: $m_i$ (i. sora a kanonikus táblának, ahol Y értéke ‘1’).

# Példa: DNF felírása

- Igazságtábla:

| sor | A | B | Y |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 1 | 1 |

- Kapott egyenlet: $'1' = Y = \overline{A} \cdot \overline{B} + A \cdot \overline{B} + A \cdot B = m_0 + m_2 + m_3$

  [0 0] [1 0] [1 1]

- Komplemens: $'0' = \overline{Y} = \overline{A} \cdot B = m_1 = (\overline{m_0 + m_2 + m_3})$

  [0 1]

# 2.) Product-of-Sums: összeg„termek” szorzata

- **összeg (OR) termek szorzata (AND kapcsolata)**
- **Maxterm** (**M**$_i$): olyan kanonikus összeg term, amelyben mindegyik logikai változó pontosan egyszer fordul elő, ponált, vagy negált alakban.
  - Tegyük fel hogy W(P,Q,R)
  - Valós maxterm: $P+\overline{Q}+\overline{R}$ , de nem valós: $P+Q$
  - Kanonikus forma: $W=(P+Q+R)(P+\overline{Q}+\overline{R})$
  - Nem kanonikus forma: $W=(P+Q)(P+\overline{Q}+\overline{R})$
- Gyakorlatban kevésbé használt forma.

# KNF: Konjunktív Normál Forma

- Jel: $W(KNF) = \prod_{i=0}^{2^n-1} M_i$
- Képzésük: a kanonikus igazságtábla azon maxtermjeinek ÉS kapcsolatát vesszük, ahol a függvény ($W$) értéke ‘0’, vagy a komplemens függvény ($\overline{W}$) értéke ‘1’.
- Pl: $W = \overline{A} + \overline{B}$ vagy

$$\overline{W} = (A+B)\cdot(\overline{A}+B)\cdot(A+\overline{B}) \overset{disztributív}{=} \overline{\overline{A\cdot B}} = A\cdot B$$

- Maxterm ($M_i$): az igazságtáblázat i. sora, ahol a kimeneti függvényérték ‘0’.

# Példák: KNF

- Legyen: $M_i = \overline{A} + B + \overline{C}$ ahol az Y kimeneti függvényérték *hamis* volt. Ez az $M_i$ maxterm igaz A,B,C változók értékének kombinációjára, kivéve egyet, ahol A=1, B=0, C=1. Tehát i=[101]=5. → $M_5$ (táblázat 5.sora)

- Legyen: $M_i = A + B + C$ ahol az Y kimeneti függvényérték *hamis* volt. Ez az $M_i$ maxterm igaz A,B,C változók értékének kombinációjára, kivéve egyet, ahol A=0, B=0, C=0. Tehát i=[000]=0. → Így $M_0$ (táblázat 0.sora)

# Példa: KNF felírása (hagyományosan)

- Igazságtábla

| sor | J | K | L | Y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 0 |

- Igazságtáblából kapjuk, hogy:

$$'0' = Y(KNF) = (J+K+L)\cdot(\overline{J}+K+L)\cdot(\overline{J}+K+\overline{L})\cdot(\overline{J}+\overline{K}+L)\cdot(\overline{J}+\overline{K}+\overline{L})$$

$$'0' = Y(KNF) = [000]\cdot[100]\cdot[101]\cdot[110]\cdot[111] = M_0 \cdot M_4 \cdot M_5 \cdot M_6 \cdot M_7$$

$$'1' = \mathrm{Y}(DNF) = (\overline{J}\cdot\overline{K}\cdot L) + (\overline{J}\cdot K\cdot\overline{L}) + (\overline{J}\cdot K\cdot L)$$

$$'1' = Y(DNF) = [001]+[010]+[011] = m_1 + m_2 + m_3$$

# Igazságtábla felírása logikai kifejezésekből I.

- a.) **DNF-ből**: felírás egyszerű
  - **Kanonikus mintermből**: egy sor képződik (ahol Y igaz),
  - **Nem kanonikus**, kevesebb változót tartalmazó termből: **több sor** is képződhet, mivel egy ilyen term *egy adott logikai változó* ponált és negált értékére is '*igaz*' kimeneti eredményt (Y=1) ad,
  - Egy sorhoz **több term** is tartozhat!

# Példa: DNF → Igazságtábla

- Eredeti egyenlet:

$$'1' = Y(DNF) = \underbrace{J \cdot \overline{K}}_{\text{term1}} + \underbrace{\overline{J} \cdot K \cdot L}_{\text{term2}} + \underbrace{J \cdot K \cdot \overline{L}}_{\text{term3}} + \underbrace{K \cdot L}_{\text{term4}}$$

kanonikus (minterm)

- Kapott igazságtábla:

| sor | J | K | L | Y | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | |
| 1 | 0 | 0 | 1 | 0 | |
| 2 | 0 | 1 | 0 | 0 | |
| 3 | 0 | 1 | 1 | 1 | term2 és term4 |
| 4 | 1 | 0 | 0 | 1 | term1 |
| 5 | 1 | 0 | 1 | 1 | term1 |
| 6 | 1 | 1 | 0 | 1 | term3 |
| 7 | 1 | 1 | 1 | 1 | term4 |

# Igazságtábla felírása logikai kifejezésekből II.

- b.) **KNF-ből**: felírás „nehezebb” (az egyes logikai változók negált értékeit kell venni)
  - **Kanonikus maxtermből**: egy sor képződik (ahol Y hamis),
  - **Nem kanonikus,** kevesebb változót tartalmazó termből: több sor is képződhet, mivel egy ilyen term egy adott logikai változó ponált és negált értékére is '*hamis*' kimeneti eredményt (Y=0) ad,
  - Egy sorhoz **több term** is tartozhat!

# Példa: KNF → Igazságtábla

- Eredeti egyenlet:

$$'0' = G(KNF) = (\overline{A} + B + C) \cdot (\overline{A} + B) \cdot (\overline{A} + \overline{B} + \overline{C})$$

term1 term2 term3

kanonikus (maxterm)

- Kapott igazságtábla:

| sor | A | B | C | G | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | |
| 1 | 0 | 0 | 1 | 1 | |
| 2 | 0 | 1 | 0 | 1 | |
| 3 | 0 | 1 | 1 | 1 | |
| 4 | 1 | 0 | 0 | 0 | **term1 és term2** |
| 5 | 1 | 0 | 1 | 0 | **term2** |
| 6 | 1 | 1 | 0 | 1 | |
| 7 | 1 | 1 | 1 | 0 | **term3** |

# Igazság táblák tömörebb felírási formája

- Eml: Kanonikus ig. táblánál: n változó → 2^n sor (összes lehetséges változó kombináció felírásával)
- Egyszerűsített / tömörebb felírás:
  - „X”: Don’t Care változó két értéke: 0 és 1 is lehet.
- Pl: $'1' = Y(\mathrm{DNF}) = J \cdot \overline{K} + \overline{J} \cdot K \cdot L + J \cdot K \cdot \overline{L} + K \cdot L$

term1 term2 term3 term4

kanonikus (minterm)

| J | K | L | Y | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | |
| 0 | 0 | 1 | 0 | |
| 0 | 1 | 0 | 0 | |
| X | 1 | 1 | 1 | term2 és term4 |
| 1 | 0 | X | 1 | term1 |
| 1 | 1 | 0 | 1 | term3 |

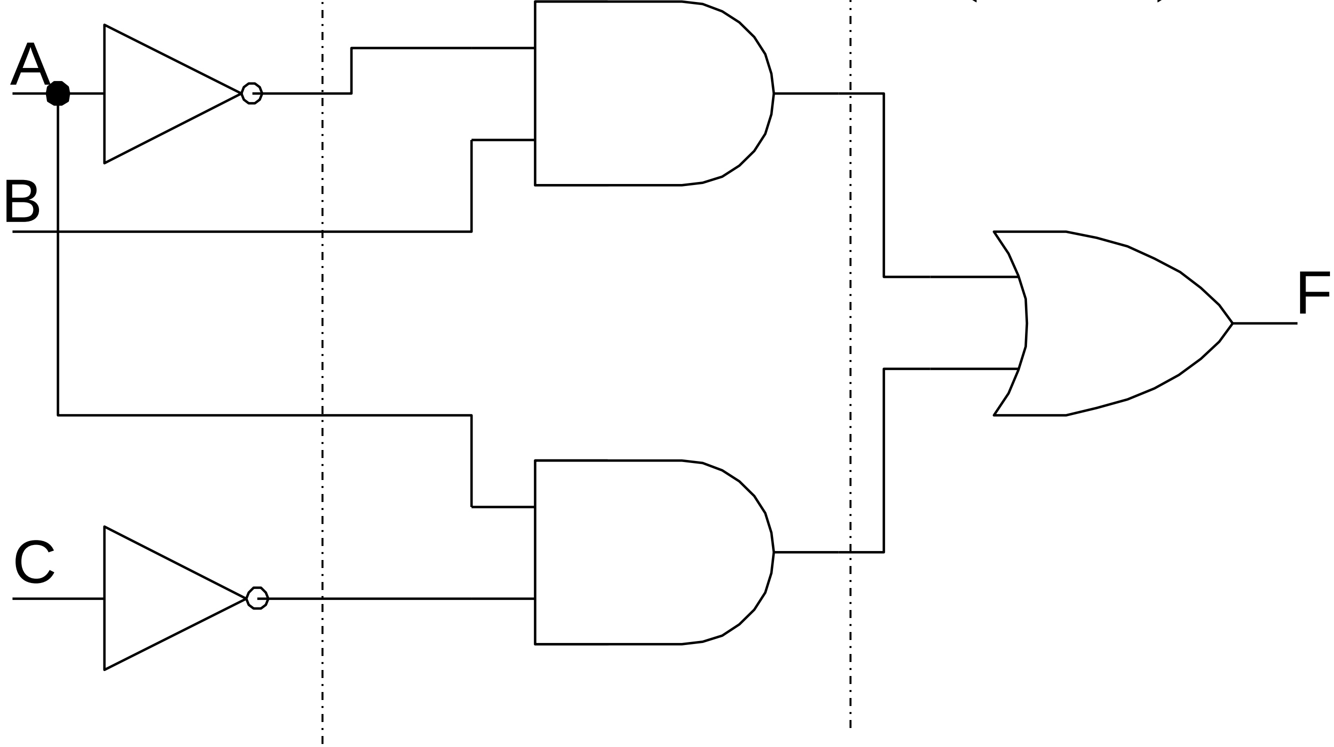
Term1: L don’t care (0 v. 1)

Term4: J dont’care (1 v. 0)

# Elvi logikai rajz

# Az egyszerűsített függvény logikai áramköri realizációja

$$F^3(A,B,C) = A \cdot \overline{C} + \overline{A} \cdot B$$

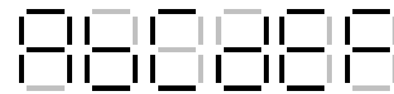


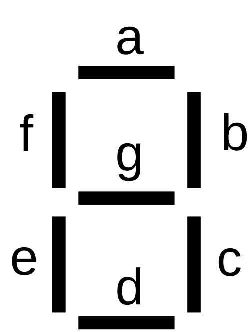

Inverter szint

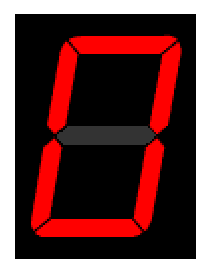
ÉS kapuk szintje

VAGY kapuk szintje

# Példa 1: 7-szegmenses dekóder áramkör tervezése

- ***Számjegyek*** (0-9) és spec. ***hexadecimális karakterek*** megjelenítésére (  )
- nemzetközi elnevezései a szegmenseknek: (a, b, c, d, e, f, g)
  - 16 érték (4 biten ábrázolható): F(X,Y,Z,W)



# Példa: 7-szegmenses dekóder tervezése (folyt)

- Igazságtábla (**f** szegmensre)
- Adott bementi értékre világítani kell-e az f szegmensnek

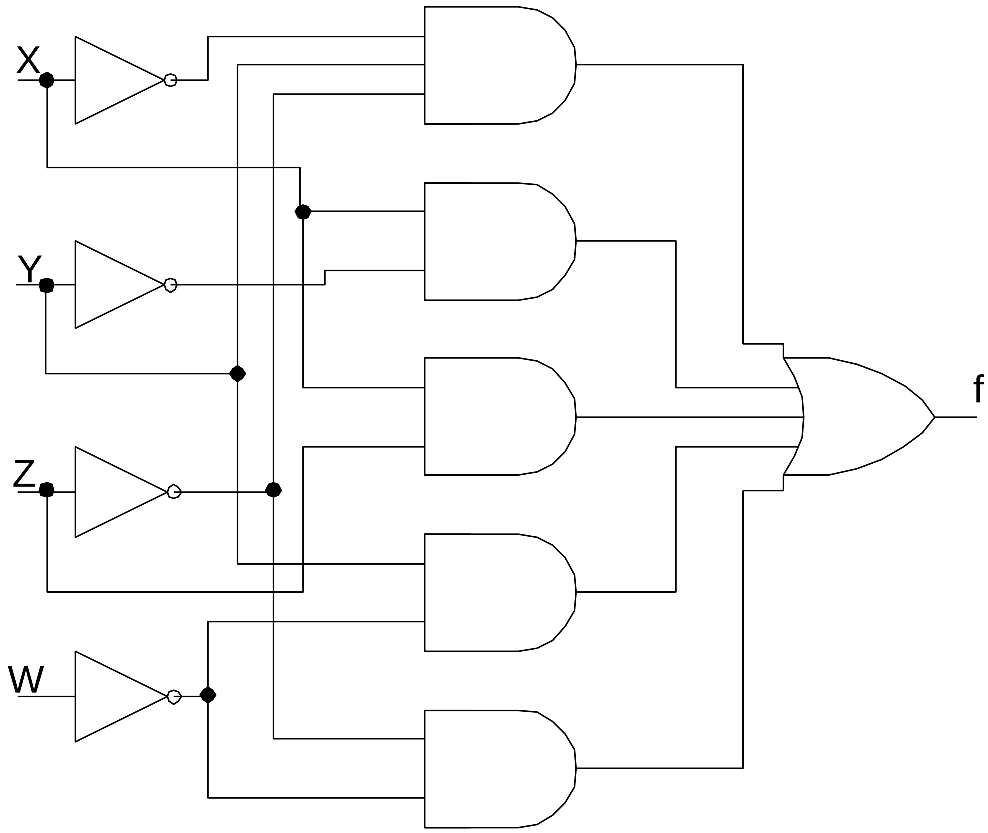
| sor | X | Y | Z | W | Ff |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | **1** |
| 1 | 0 | 0 | 0 | 1 | **0** |
| 2 | 0 | 0 | 1 | 0 | **0** |
| 3 | 0 | 0 | 1 | 1 | **0** |
| 4 | 0 | 1 | 0 | 0 | **1** |
| 5 | 0 | 1 | 0 | 1 | **1** |
| 6 | 0 | 1 | 1 | 0 | **1** |
| 7 | 0 | 1 | 1 | 1 | **0** |
| 8 | 1 | 0 | 0 | 0 | **1** |
| 9 | 1 | 0 | 0 | 1 | **1** |
| 10 | 1 | 0 | 1 | 0 | **1** |
| 11 | 1 | 0 | 1 | 1 | **1** |
| 12 | 1 | 1 | 0 | 0 | **1** |
| 13 | 1 | 1 | 0 | 1 | **0** |
| 14 | 1 | 1 | 1 | 0 | **1** |
| 15 | 1 | 1 | 1 | 1 | **1** |

- Kapott optimalizált F kimeneti függvény **f** szegmensre:

$$F^{n=4}_{'f'}(X,Y,Z,W) = \overline{Z}\cdot\overline{W} + X\cdot\overline{Y} + Y\cdot\overline{W} +$$
$$+ X\cdot Z + \overline{X}\cdot Y\cdot\overline{Z}$$

# Példa 1: A 7-szegmenses dekóder logikai áramköri realizációja

(f  szegmensre)




$$F_f(X,Y,Z,W) = \overline{Z}\cdot\overline{W} + X\cdot\overline{Y} + Y\cdot\overline{W} + X\cdot Z + \overline{X}\cdot Y\cdot\overline{Z}$$