



Számítógép Architektúrák (MIKNB113A)

1. előadás: Ismétlés - Logikai egyenletek leírása
Boole-algebra, igazságtábla, Karnaugh tábla

Előadó: Dr. Vörösházi Zsolt

voroshazi.zsolt@mik.uni-pannon.hu

Jegyzetek, segédanyagok:

- Könyvfejezetek:

- <http://www.virt.vein.hu>

- Oktatás → Tantárgyak → Számítógép
Architektúrák

- (00_chapter, 01_chapter.pdf)

- Fóliák, óravázlatok .ppt (.pdf)

- Frissítésük folyamatosan

További ajánlott irodalom

-  Dr. Holczinger, Dr. Göllei. Dr. Vörösházi:
Digitális Technika I. (TAMOP 4.1.2A - 2012) :
[Digitalis technika I TAMOP](#)
-  Dr. Holczinger, Dr. Göllei. Dr. Vörösházi:
Digitális Technika II. (TAMOP 4.1.2A - 2013) :
[Digitalis technika II TAMOP](#)

Digitális áramköri tervezés

- Logikai konstansok:

- ☐ Logikai állítás: Igaz / Hamis, True / False, '1' / '0', Asserted / Non-Asserted

- Logikai (bináris) változók:

- ☐ Pl: 'A' logikai (független bemeneti) változó esetén legyen,

A:=fotódióda hiba

- ☐ 'A' lehet: T / F (A=F nincs hiba; A=T hiba)

- ☐ Logikai változó neve utaljon a funkciójára

- Logikai operátorok:

- ☐ Felírásuk kanonikus igazság táblázattal (Truth Table) ↔ Karnaugh tábla (K-Map)

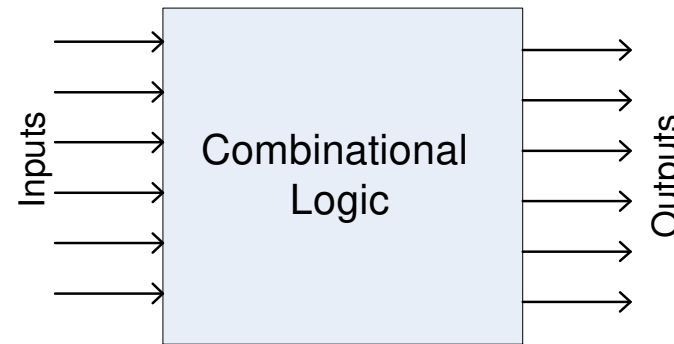
Logikai hálózatok csoportosítása

Ezek alapján kétféle hálózatot különböztetünk meg

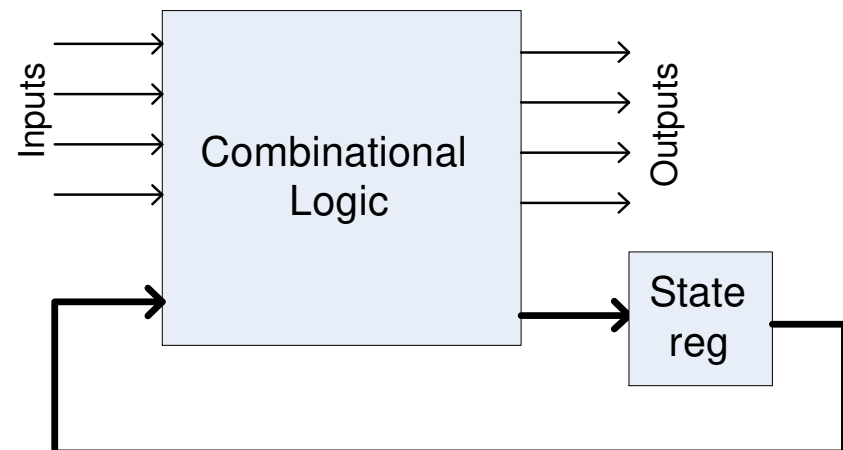
- **(K.H.) Kombinációs logikai hálózat**ról beszélünk: ha a mindenkori kimeneti kombinációk létrehozásához *elég a bemeneti kombinációk* pillanatnyi értéke.
- **(S.H.) Sorrendi (szekvenciális) logikai hálózat**ról beszélünk: ha a mindenkori kimeneti kombinációt, nemcsak *a pillanatnyi* bemeneti kombináció,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

Kombinációs vs. sorrendi hálózatok felépítése:

- Kombinációs hálózat:



- Sorrendi hálózat:



Igazságtábla: logikai operátorok felírása

- 'F' logikai függvény megadása az 'A,C,B' logikai változók összes lehetséges értékétől függően

Jel: $F(A,C,B)$ //3 változó $\rightarrow 2^3 = 8$ sor//

A	C	B	F
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



A	C	B	Fgv
F	F	F	F
F	F	T	T
F	T	F	T
F	T	T	F
T	F	F	T
T	F	T	F
T	T	F	F
T	T	T	F

sor
0.
1.
2.
3.
4.
5.
6.
7.

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

Logikai operátorok és igazság táblázatuk (NOT)

- Jel: $\text{NOT } A = \bar{A}$
- 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 = $B \cdot C$
- Formális definíció igazságtáblával:

B	C	$B \cdot C$
0	0	0
0	1	0
1	0	0
1	1	1

- Def: $B \cdot C$ értéke pontosan akkor 'igaz' ha 'B' és 'C' is egyszerre 'igaz', különben hamis

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

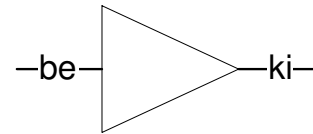


Egy-, ill. kétváltozós logikai függvények bemutatása, és szabványos jelöléseik

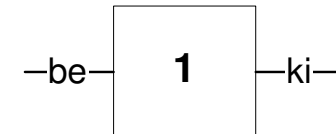
Egyváltozós logikai függvények:

■ Jelmásoló („buffer” - jel-erősítő):

be	ki
0	0
1	1



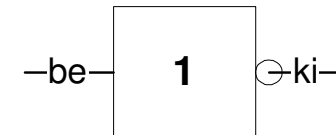
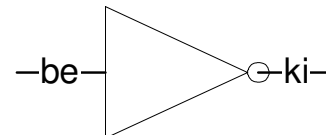
Nemzetközi
szabvány



Magyar
szabvány

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

be	ki
0	1
1	0

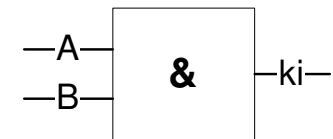
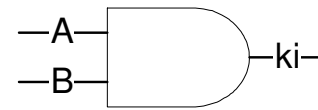


Kétváltozós logikai függvények:

■ ÉS (AND):

$$A \cdot B$$

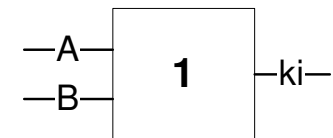
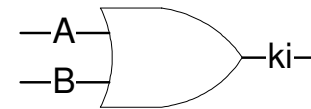
A	B	ki
0	0	0
0	1	0
1	0	0
1	1	1



■ VAGY (OR):

$$A + B$$

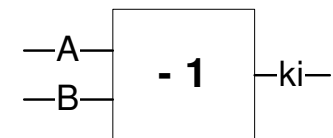
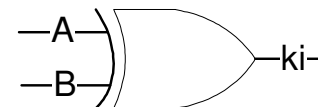
A	B	ki
0	0	0
0	1	1
1	0	1
1	1	1



■ Antivalencia (XOR):

$$A \oplus B$$

A	B	ki
0	0	0
0	1	1
1	0	1
1	1	0

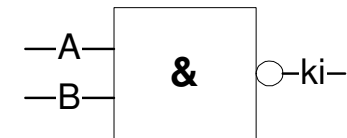
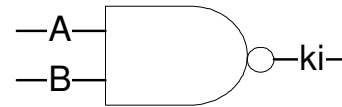


Kétváltozós log.függv. (folyt.):

■ NEM-ÉS (NAND):

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

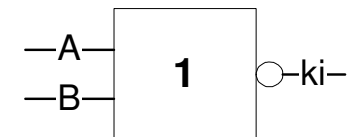
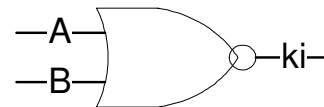
A	B	ki
0	0	1
0	1	1
1	0	1
1	1	0



■ NEM-VAGY (NOR):

$$\overline{A + B}$$

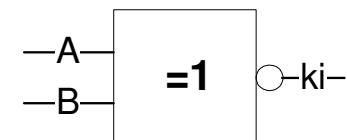
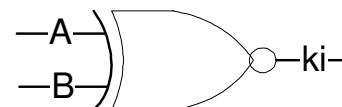
A	B	ki
0	0	1
0	1	0
1	0	0
1	1	0



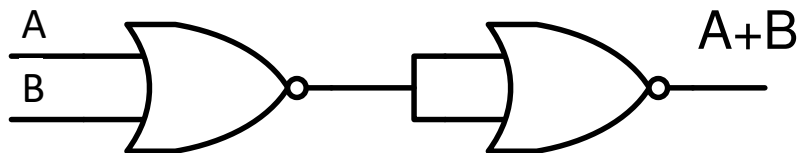
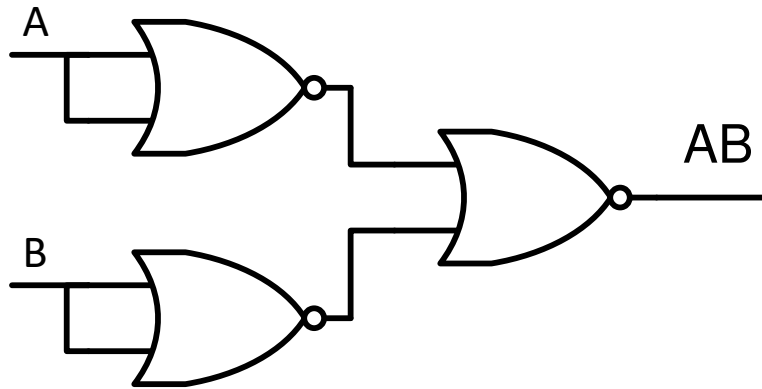
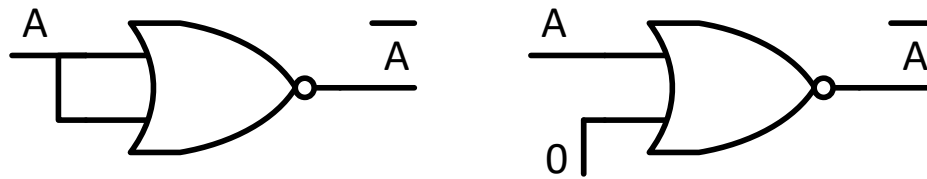
■ Ekvivalencia (NXOR):

$$A \odot B$$

A	B	ki
0	0	1
0	1	0
1	0	0
1	1	1



DEF: Funkcionális teljesség - példák

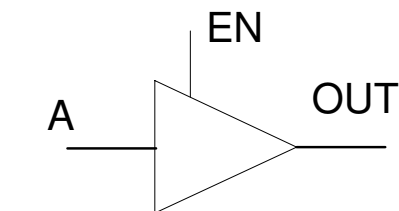


**Funkcionálisan teljes
vagy univerzális
áramköri alapelemek:**
Logikai hálózatok
esetén a CMOS VLSI
technológiában a
NAND, illetve NOR
kapu.

(Aritmetikai egységek
esetében ilyen
univerzális építőelem
az „**összeadó**”.)

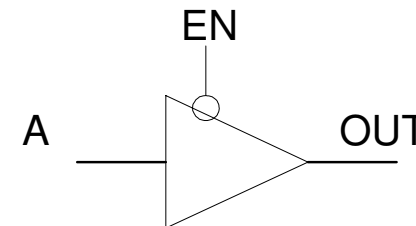
Tri-State Buffer:

- buszok esetén használatos: kommunikációs irány változhat
 - Driver: egyirányú kommunikációra
 - Transceiver: kétirányú kommunikációra
- 3 állapota lehet:
 - magas: '1'
 - alacsony: '0' (normál TTL szintek)
 - nagy impedanciás állapot: 'Z' – mindkét kimeneti tranzisztor zár



High-true enable

A	EN	OUT
0	1	0
1	1	1
X	0	Z



Low-true enable

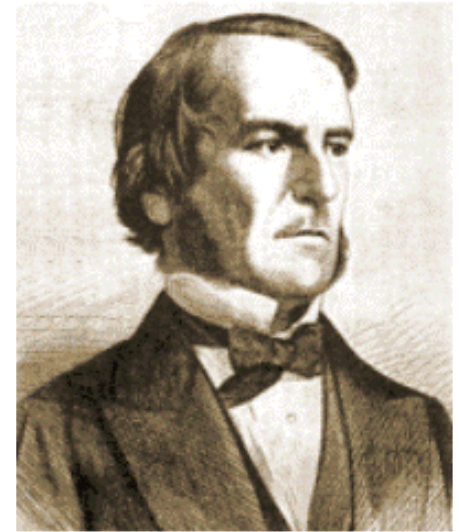
Smart áramkörök ☺





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.
- HW tervezés alacsonyabb absztrakciós szintjén rendkívül fontos szerepe van. (Specifikáció + egyszerűsítés)

Boole algebra elemei:

- A vizsgált 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

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

*

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

*

Elnyelési
tul.

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

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

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

*

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

De-Morgan azonosságok:

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

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

DUAL
ITÁS

Boole-algebrai azonosság igazolása igazságtáblával

■ Pl: De Morgan

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

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

Dualitás elve

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

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

Logikai egyenletek megadása igazság-táblázat alapján

■ P|:

sor	A	B	F
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:

$$F = A \cdot \bar{B}$$

■ P|:

sor	A	B	F
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:

$$1 = F = \bar{A} \cdot \bar{B} + A \cdot \bar{B} + A \cdot B \Rightarrow \bar{B} + A \cdot B \Rightarrow \bar{B} + A$$

$$0 = \bar{F} = \bar{A} \cdot B$$

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

- **Szorzat (AND) termék összeg (OR) kapcsolata**
- Emberi szemléletmódhoz közelebb áll: a táblázat soraiból azokat a függvényértékeket (F) vesszük amelyek ‘1’-esek.
- Def: Triviális forma: ha egy változó egy adott szorzat termében vagy pozitívan, vagy negatívan legfeljebb egyszer szerepel.
Ezt hívják még **mintermnek** (m_i) vagy **kanonikus szorzat termék** is. Tegyük fel hogy $F(A,B,C)$

□ Pl: valós / triviális / kanonikus formulák: $A, \bar{A}, A \cdot B, \bar{A} \cdot B \cdot \bar{C}$

□ Pl: érvénytelen formulák (de ettől még Boole kifejezés), ami jelenti egyben azt is, hogy tovább egyszerűsíthetők:

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

Diszjunktív Normál Forma:

■ Jel:

$$F(DNF) : \sum_{i=0}^{2^n-1} m_i$$

- n változó esetén 2^n lehetséges minterm van.
- Képzésük: az igazságtáblázatból azoknak a **mintermeknek a VAGY kapcsolatát** vesszük, ahol függvényértékek sorában (F) '1' -es szerepel, vagy ahol a függvény komplementensének ($\overline{F}$) értéke '0'.
- minterm: m_i (i. sora a kanonikus táblának, ahol F értéke '1').

Példa: DNF felírása

■ Igazságtábla:

sor	A	B	F
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	1

■ Kapott egyenlet: $1 = F = \overline{A} \cdot \overline{B} + A \cdot \overline{B} + A \cdot B = m_0 + m_2 + m_3$
[0 0] [1 0] [1 1]

■ Komplementens: $0 = \overline{F} = \overline{A} \cdot B = m_1 = \overline{(m_0 + m_2 + m_3)}$
[0 1]

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

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

KNF: Konjunktív Normál Forma

■ Jel:
$$F(KNF) = \prod_{i=0}^{2^n-1} M_i$$

- Képzésük: a kanonikus igazságtábla azon **maxtermjeinek ÉS kapcsolatát** vesszük, ahol a függvény (F) értéke '0', vagy a komplementens függvény ($\overline{F}$) értéke '1'.

■ Pl: $F = \overline{A} + \overline{B}$ vagy

$$\overline{F} = (A + B) \cdot (\overline{A} + B) \cdot (A + \overline{B}) \stackrel{\text{disztributív}}{=} \overline{A \cdot B} = A \cdot B$$

- Maxterm (M_i): az igazságtáblázat i . sora, ahol az F kimeneti függvényérték '0'.

Példa: KNF felírása

- Igazságtábla:

sor	A	B	F
0	0	0	1
1	0	1	1
2	1	0	1
3	1	1	0

- Kapott egyenlet: $0 = F = \overline{A} + \overline{B} = M_3$
[1 1]

- Komplementens:

$$1 = \overline{F} = (A + B) \cdot (\overline{A} + B) \cdot (A + \overline{B}) = M_0 \cdot M_1 \cdot M_2$$

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

Példák: KNF

- Legyen: $M_i = \bar{A} + B + \bar{C}$ ahol a 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 $[101]=5. \rightarrow M_5$ (táblázat 5.sora)
- Legyen: $M_i = A + B + C$ ahol a 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 $[000]=0. \rightarrow$ Így M_0 (táblázat 0.sora)

Példa: KNF/DNF felírása

■ Igazságtábla

sor	J	K	L	F
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:

$$0 = F(KNF) = (J + K + L) \cdot (\bar{J} + K + L) \cdot (\bar{J} + K + \bar{L}) \cdot (\bar{J} + \bar{K} + L) \cdot (\bar{J} + \bar{K} + \bar{L})$$

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

$$1 = F(DNF) = (\bar{J} \cdot \bar{K} \cdot L) + (\bar{J} \cdot K \cdot \bar{L}) + (\bar{J} \cdot K \cdot L)$$

$$F(DNF) = [001] + [010] + [011] = m_1 + m_2 + m_3$$

Igazságtábla felírása logikai kifejezésekből I.

■ a.) **DNF-ből**: felírás egyszerű

- **Kanonikus** mintermből: egy sor képződik (ahol F 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 (F) ad,
- Egy sorhoz **több term** is tartozhat!

Példa: DNF $\rightarrow$ Igazságtábla

■ Eredeti egyenlet:

$$'1' = F(DNF) = J \cdot \bar{K} + \bar{J} \cdot K \cdot L + J \cdot K \cdot \bar{L} + K \cdot L$$

term1

term2

term3

term4

kanonikus (minterm)

■ Kapott igazságtábla:

sor	J	K	L	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

term2 és term4

term1

term1

term3

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 F 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 ($F=0$) ad,
 - Egy sorhoz **több term** is tartozhat!

Igazság táblák tömörebb felírási formája

- Eml: Kanonikus ig. táblánál: n változó $\rightarrow 2^n$ sor (összes lehetséges változó kombináció felírásával)
- Egyszerűsített / tömörebb felírás:
 - „X”: Don't Care változó két értéke: 0 és 1 is lehet.
- Pl: $'1' = F(\text{DNF}) = \underbrace{J \cdot \bar{K}}_{\text{term1}} + \underbrace{\bar{J} \cdot K \cdot L}_{\text{term2}} + \underbrace{J \cdot K \cdot \bar{L}}_{\text{term3}} + \underbrace{K \cdot L}_{\text{term4}}$

J	K	L	F
0	0	0	0
0	0	1	0
0	1	0	0
X	1	1	1
1	0	X	1
1	1	0	1

term2 és term4

term1

term3

kanonikus (minterm)

Term1: L don't care (0 v. 1)

Term4: J don't care (1 v. 0)

NTSH: Nem Teljesen Specifikált Hálózat (Don't Care kimenet)

- Bizonyos bemeneti kombinációkra ugyanazt a kimeneti eredményt kapjuk (irreleváns)
- Jele: „–” Don't care kimeneti értéke F-nek

■ Pl.

A	B	F
0	X	1
1	0	-
1	1	0

$$\text{ha '–'=1, } F = \bar{A} + A \cdot \bar{B} = \bar{A} + \bar{B}$$

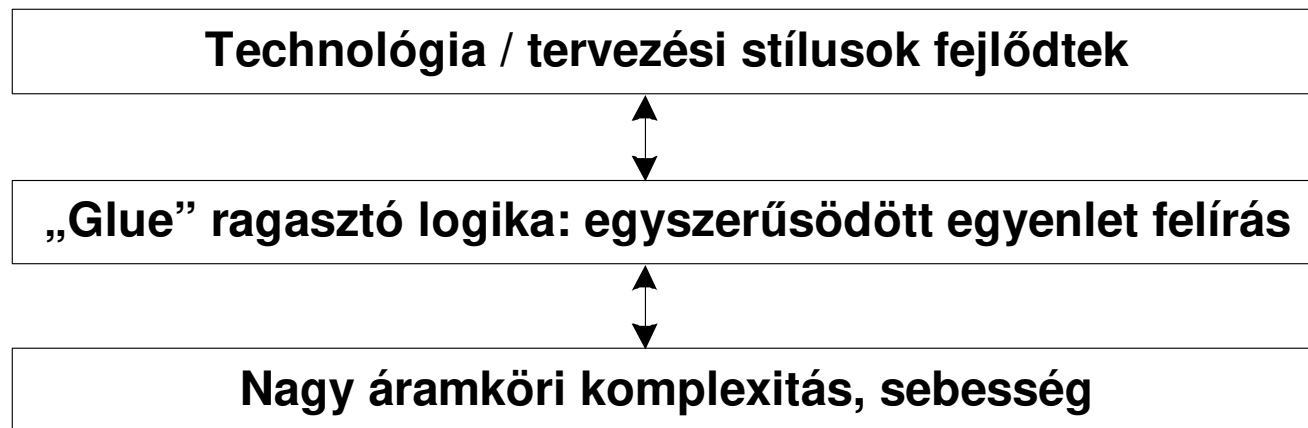
$$\text{ha '–'=0, } F = \bar{A}$$



KARNAUGH TÁBLÁK: grafikus minimalizálás

Karnaugh táblák

- Korai időszakban: logikai elemek hatalmas, nehezen tervezhető, nagy energiát disszipáló eszközökből álltak.
- Logikai kifejezések egyszerűsítése. Ma: HW olcsó elemekből épül fel. Cél: az áramköri minimalizáció (modularitás, egyszerűség).



- **K-Map / Veicht diagram**: grafikus ábrázolási és egyszerűsítési mód, a kanonikus igazságtábla egy újrendezett formája (több forma is létezik, és fontos a betűk, címkék sorrendje)

Karnaugh tábla felírása igazság táblázatból

- Igazságtábla mindenegybes sorának kimeneti függvény értékéhez (Y_i) a Karnaugh tábla egy négyzete (cellája, rovata) feleltethető meg.
- Pl. $n=2$ változó esetén lehetséges táblák:

sor	A	B	Y
0	0	0	Y0
1	0	1	Y1
2	1	0	Y2
3	1	1	Y3

		A	
		0	1
B	0	Y ₀	Y ₂
	1	Y ₁	Y ₃

Egy lehetséges
jelölés

		B	
		0	1
A	0	Y ₀	Y ₁
	1	Y ₂	Y ₃

Általános (standard)
jelölés

Karnaugh táblák

- $n=2, 3, 4$ változóval még könnyű felírni (de $n>4$ változó felett már más technikát érdemes használni. Pl. QM minimalizálás)
- Pl: $n=3$ változó esetén lehetséges táblákra:

		B			
		A			
C	AB	00	01	11	10
0		Y_0	Y_2	Y_6	Y_4
1		Y_1	Y_3	Y_7	Y_5

Egy lehetséges
jelölés

		C			
		B			
A	BC	00	01	11	10
0		Y_0	Y_1	Y_3	Y_2
1		Y_4	Y_5	Y_7	Y_6

Általános (standard)
jelölés

Karnaugh táblák

- Pl: $n=4$ változó esetén lehetséges táblákra:

AB		A			
		00	01	11	10
CD	00	Y_0	Y_4	Y_{12}	Y_8
	01	Y_1	Y_5	Y_{13}	Y_9
	11	Y_3	Y_7	Y_{15}	Y_{11}
	10	Y_2	Y_6	Y_{14}	Y_{10}

B

D

Egy lehetséges
jelölés

CD		C			
		00	01	11	10
AB	00	Y_0	Y_1	Y_3	Y_2
	01	Y_4	Y_5	Y_7	Y_6
	11	Y_{12}	Y_{13}	Y_{15}	Y_{14}
	10	Y_8	Y_9	Y_{11}	Y_{10}

D

B

Általános (standard)
jelölés

Karnaugh táblák

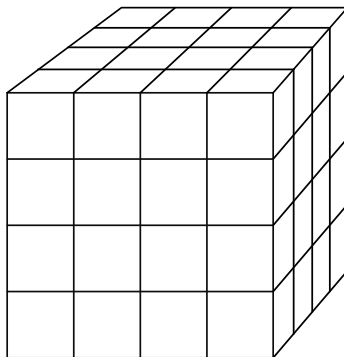
■ $n=5$ változó esetén

CD		D				C			
		00	01	11	10	00	01	11	10
AB	00	Y_0	Y_1	Y_3	Y_2	Y_6	Y_7	Y_5	Y_4
	01	Y_8	Y_9	Y_{11}	Y_{10}	Y_{14}	Y_{15}	Y_{13}	Y_{12}
	11	Y_{24}	Y_{25}	Y_{27}	Y_{26}	Y_{30}	Y_{31}	Y_{29}	Y_{28}
	10	Y_{16}	Y_{17}	Y_{19}	Y_{18}	Y_{22}	Y_{23}	Y_{21}	Y_{20}

CD		E=0		C	
		00	01	11	10
AB	00	Y_0	Y_2	Y_6	Y_4
	01	Y_8	Y_{10}	Y_{14}	Y_{12}
	11	Y_{24}	Y_{26}	Y_{30}	Y_{28}
	10	Y_{16}	Y_{18}	Y_{22}	Y_{20}

CD		E=1		C	
		00	01	11	10
AB	00	Y_1	Y_3	Y_7	Y_5
	01	Y_9	Y_{11}	Y_{15}	Y_{13}
	11	Y_{25}	Y_{27}	Y_{31}	Y_{29}
	10	Y_{17}	Y_{19}	Y_{23}	Y_{21}

■ $n=6$ változó esetén



FED		CBA							
		000	001	011	010	100	101	111	110
	000								
	001								
	011								
	010								
	100								
	101								
	111								
	110								

Boole függvény ábrázolási módjai

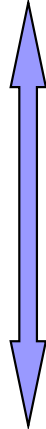
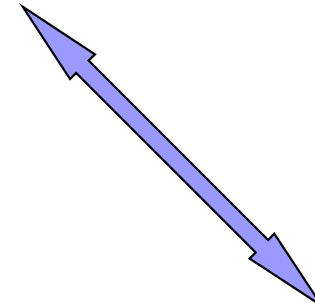
■ Boole-algebrai kifejezés: $Y = \overline{A} \cdot \overline{B} + A \cdot \overline{B}$

■ Igazságtábla:



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

■ Karnaugh tábla:



		B	
		0	1
A	0	1	0
	1	1	0

Szomszédosság – adjacencia

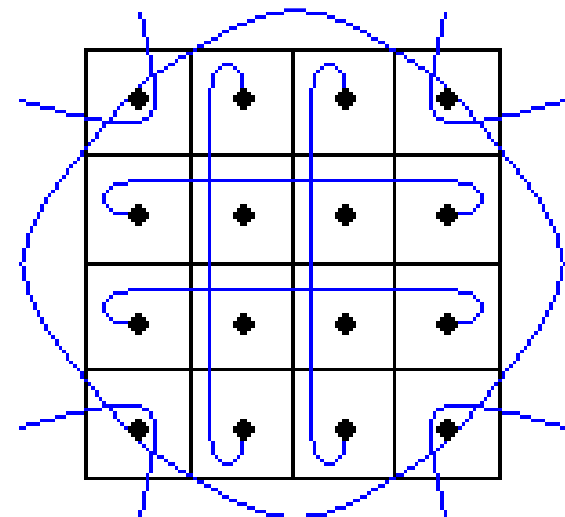
- Def: Ha egy Karnaugh táblában két szomszédos (adjacent) cella csak egy változó értékében különbözik (egységnyi távolság)!

- Pl. $Y_3 = \bar{A} \cdot B \cdot C$ és $Y_7 = A \cdot B \cdot C$

		C			
		B			
A	BC	00	01	11	10
	0	0 ₀	0 ₁	1 ₃	0 ₂
A	1	0 ₄	0 ₅	1 ₇	0 ₆

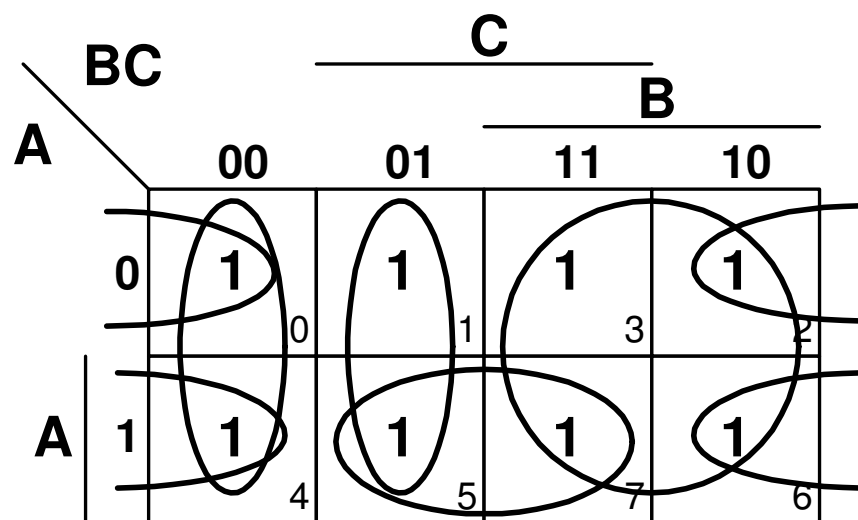
Egyszerűsítés Karnaugh táblákkal

- **Tömörítés („tömbösítés”)** szabályai:
 - 2^n ($n=0,1,2..$) *term* vonható be egy tömbbe,
 - Egyetlen *term* több tömbben is szerepelhet (átlapolódás lehetséges)
 - Egyik tömb, a másikat nem tartalmazhatja teljes mértékben, (redundancia)
 - Mindig a lehető legnagyobb lefedéseket keressük, és haladjunk a legkisebb méretű tömbök/lefedések felé
 - Don't care ('-') kimeneti függvényértékeket a jobb lefedésnek megfelelően kell megválasztani (NTSH)
 - Egymás mellett lévő (adjacens) sorokra és oszlopokra érvényes:



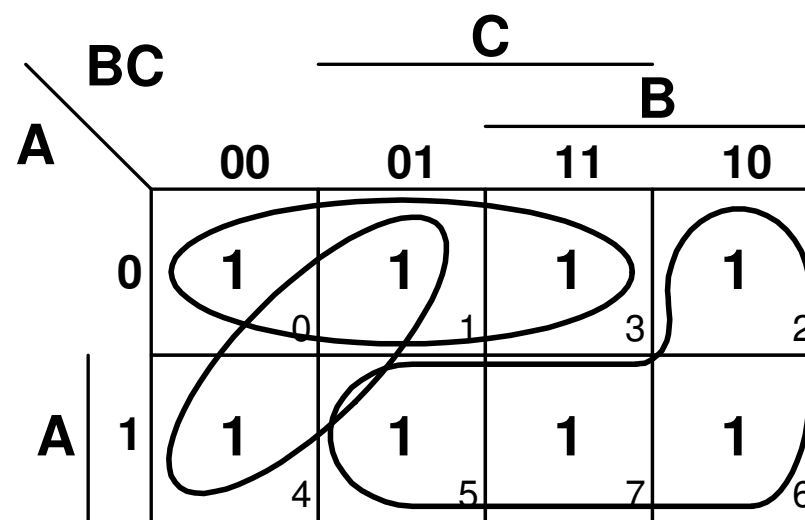
Példa: Karnaugh táblák egyszerűsítése

■ érvényes



Nem összes, de lehetséges egyszerűsítések - érvényes

érvénytelen



Átlós, és nem 2^n számú '1'-es lefedés érvénytelen

Lehetséges módszerek Karnaugh tábla értelmezésére (ism):

- M1: $F(DNF)$ '1'-esek lefedésével képzett (normál, eddig általánosan használt módszer)
- M2: $\overline{F(DNF)}$ '0'-k lefedésével képzett inverz függvény felírás
- M3: $F(KNF)$ '0'-k lefedésével képzett (normál)
- M4: $\overline{F(KNF)}$ '1'-esek lefedésével képzett inverz függvény felírás