Pannon Egyetem
Villamosmérnöki és Információs Tanszék

Digitális Áramkörök

(Villamosmérnök BSc / Mechatronikai mérnök MSc)

3. hét - Grafikus minimalizálás. Quine-McCluskey féle számjegyes minimalizálás

Előadó: Dr. Vörösházi Zsolt
voroshazi.zsolt@virt.uni-pannon.hu

Kapcsolódó jegyzet, segédanyag:

- http://www.virt.uni-pannon.hu → Oktatás → Tantárgyak → Digitális Áramkörök (Villamosmérnöki BSc / Mechatronikai mérnöki BSc/MSc).
  - Fóliák, óravázlatok (.ppt)
  - Frissítésük folyamatosan

Ismétlés: észrevétel

- Fontos megjegyzés: az Arató P. könyv illetve a nemzetközi szakirodalom eltérő módon indexeli a Maxterm-eket KNF-esetén:
  - Arató könyv: $\overline{m_i} \rightarrow M_k$: ahol k=(2^n–1 – i)
    - Pl: n=3 esetén $\overline{m_1} \rightarrow M_6$

$$Y(DNF) = \overline{A} \cdot \overline{B} \cdot C \Rightarrow \overline{Y(KNF)} = A + B + \overline{C}$$

  - **Nemzetközi szakirodalom: $m_i \rightarrow M_i$**
    - Pl: n=3 esetén $\overline{m_1} \rightarrow M_1$

$$Y(DNF) = \overline{A} \cdot \overline{B} \cdot C \Rightarrow \overline{Y(KNF)} = A + B + \overline{C}$$

[0 0 1] = 1=i

Logikai függvények minimalizálása

Függvényminimalizálás általánosan

- Függvényminimalizálás a szomszédos mintermek megkeresésével, párba válogatásával tehető meg:
  - **Szomszédos** = ha van egy olyan logikai *változó*, amely az egyik mintermben ponált, a másikban negált értékével szerepel (a többi független változó azonos értéken szerepel)
- A szomszédosság megállapítása után **egyszerűsítünk**.
- Minterm → implikáns (egyszerűsíthető) → prímimplikáns (tovább nem egyszerűsíthető)
  - **Prímimplikáns**: a szomszédos összevonásokat mindaddig folytatni kell, amíg a logikai függvény olyan alakú nem lesz, amelyben egyetlen log. változó (betű) sem hagyható el anélkül, hogy a logikai függvény ne változna! Ezek a *szorzatok* a prímimplikánsok.
  - Tehát: a logikai függvény legegyszerűbb DNF alakja a **prímimplikánsok összege.**

Függvényegyszerűsítési eljárások

- 1.) Algebrai módszer (Boole algebrai azonosságokkal)
- 2.) Kifejtési módszer
- 3.) Grafikus módszer: (Karnaugh tábla, igazság tábla)
- 4.) Normálformák:
  - DNF: Diszjunktív Normál Forma
  - KNF: Konjunktív Normál Forma
- 5.) Számjegyes minimalizálás: Quine-McCluskey (prímimplikánsok módszere)

1.) Algebrai módszer

- A Boole-algebra azonosságait használjuk fel az egyszerűsítéshez. Legyen:

$$F^{n=3}(A,B,C) := \sum_{i=0}^{7}(1,3,5,7) = m_1^3 + m_3^3 + m_5^3 + m_7^3 \; //DNF$$

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

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

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

2.) Kifejtési módszer*:

- Komplexebb függvények esetén egy adott változó értékét először ponáltnak, majd negáltnak definiáljuk, végül pedig az így kiszámított két logikai kifejezést „összeadjuk”. Leegyszerűsödik a függvényminimalizálási feladat. Két mód:

I.) $$F^n(x_1, x_2, \ldots, x_n) = x_1 \cdot F(1, x_2, \ldots, x_n) + \overline{x_1} \cdot F(0, x_2, \ldots, x_n)$$

II.) $$F^n(x_1, x_2, \ldots, x_n) = \overline{\left[x_1 + \overline{F(0, x_2, \ldots, x_n)}\right] \cdot \left[\overline{x_1} + \overline{F(1, x_2, \ldots, x_n)}\right]}$$

Példa: kifejtési tétel alkalmazása

- Legyen F_1 függvény a következő (**módszer I.**):

$$F_1^3(A,B,C) = m_2 + m_3 + m_4 + m_6 = \overline{A}\cdot B\cdot\overline{C} + \overline{A}\cdot B\cdot C + A\cdot\overline{B}\cdot\overline{C} + A\cdot B\cdot\overline{C}$$

  - Ha A:=1

$$F_1^3(\textcolor{red}{1},B,C) = \cancel{0\cdot B\cdot\overline{C}} + \cancel{0\cdot B\cdot C} + 1\cdot\overline{B}\cdot\overline{C} + 1\cdot B\cdot\overline{C}$$

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

  - Ha A:=0

$$F_1^3(\textcolor{red}{0},B,C) = 1\cdot B\cdot\overline{C} + 1\cdot B\cdot C + \cancel{0\cdot\overline{B}\cdot\overline{C}} + \cancel{0\cdot B\cdot\overline{C}}$$

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

  - Végül „összeadjuk” a kettőt (egyszerűsített alak):

$$F_1^3(\textcolor{red}{A},B,C) = A\cdot F_1(\textcolor{red}{1},B,C) + \overline{A}\cdot F_1(\textcolor{red}{0},B,C) =$$

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

Példa: kifejtési tétel alkalmazása

- Legyen F_1 függvény a következő (**módszer II.**):

$$F_1^3(A,B,C)=m_2+m_3+m_4+m_6=\overline{A}\cdot B\cdot\overline{C}+\overline{A}\cdot B\cdot C+A\cdot\overline{B}\cdot\overline{C}+A\cdot B\cdot\overline{C}$$

  - Ha A:=1

$$F_1^3(1,B,C)=\cancel{0\cdot B\cdot\overline{C}}+\cancel{0\cdot B\cdot C}+1\cdot\overline{B}\cdot\overline{C}+1\cdot B\cdot\overline{C}$$

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

  - Ha A:=0

$$F_1^3(0,B,C)=1\cdot B\cdot\overline{C}+1\cdot B\cdot C+\cancel{0\cdot\overline{B}\cdot\overline{C}}+\cancel{0\cdot B\cdot\overline{C}}$$

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

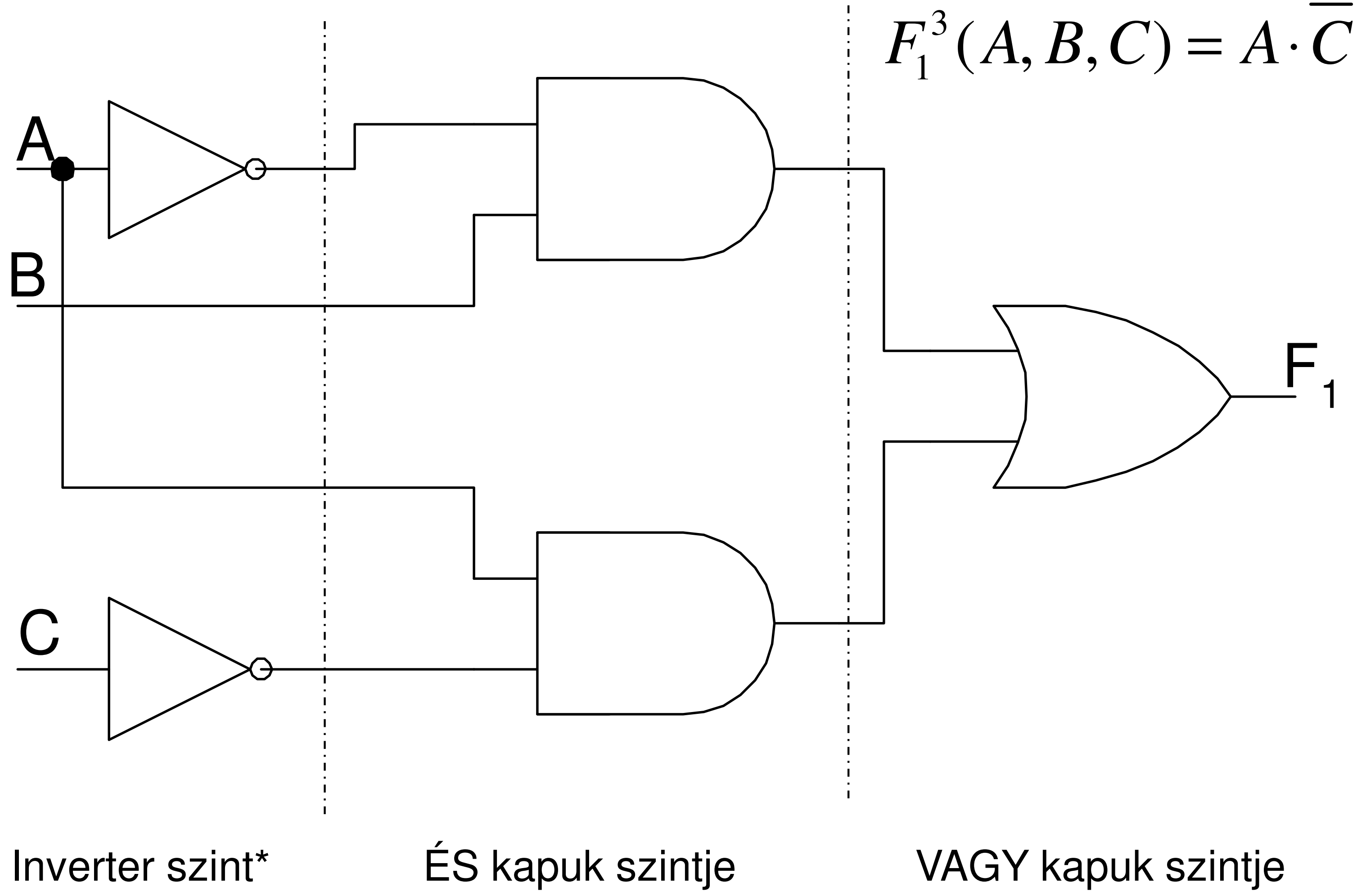
  - Végül „összeszorozzuk” a kettőt (egyszerűsített alak):

$$F_1^3(A,B,C)=\overline{\left[A+\overline{F_1(0,B,C)}\right]\cdot\left[\overline{A}+\overline{F_1(1,B,C)}\right]}=$$

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

Az egyszerűsített F függvény logikai áramköri realizációja:

$$F_1^3(A,B,C) = A\cdot\overline{C} + \overline{A}\cdot B$$



*Arató könyv: 2-szintű elvi kombinációs logikai hálózat (inverter szintet nem számolva!)

Grafikus minimalizálás (Karnaugh tábla)

3.) 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)

Technológia / tervezési stílusok fejlődtek

↕

„Glue” ragasztó logika: egyszerűsödött egyenlet felírás

↕

Nagy áramköri komplexitás, sebesség

- **<u>K-Map / Veitch (KV) diagram</u>**: grafikus ábrázolási és egyszerűsítési mód, a kanonikus igazságtábla egy újrarendezett formája
  - Bell Labs: 1952-54 – Edward *Veitch*, Maurice *Karnaugh*
  - több formája is létezik (de fontos a változók / címkék sorrendje!)

Karnaugh tábla felírása igazság táblázatból

- Igazságtábla mindenegyes sorának kimeneti értékéhez (Y_i) a Karnaugh tábla egy négyzete (cella) feleltethető meg.
- Pl. n=2 változó esetén lehetséges táblák (**peremezési** szabályok):

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

B \ A	0	1
0	Y_0	Y_2
1	Y_1	Y_3

Lehetséges könyvbeli jelölés

A \ B	0	1
0	Y_0	Y_1
1	Y_2	Y_3

Általánosan elfogadott jelölés

Karnaugh táblák

- n=2, 3, 4 változóval még könnyű felírni (>4 változó felett már más technikát érdemes használunk)
- Pl: n=3 változó esetén lehetséges táblákra:

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

Lehetséges könyvbeli jelölés(ek)

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

Általánosan elfogadott jelölés

Karnaugh táblák

- Pl: n=4 változó esetén lehetséges táblákra:

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

Lehetséges könyvbeli jelölés(ek)

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

Általánosan elfogadott jelölés

Karnaugh táblák

- n= 5 változó esetén

AB \ CD	00	01	11	10	00	01	11	10
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}

D, C, B, A, E, E

E=0

AB \ CD	00	01	11	10
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}

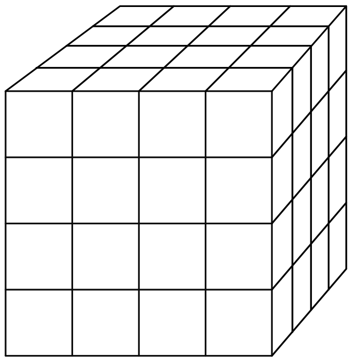
C, B, A, D

E=1

AB \ CD	00	01	11	10
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}

C, B, A, D

- n=6 változó esetén



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

Boole függvény ekvivalens 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:

A \ B	0	1
0	1	0
1	1	0

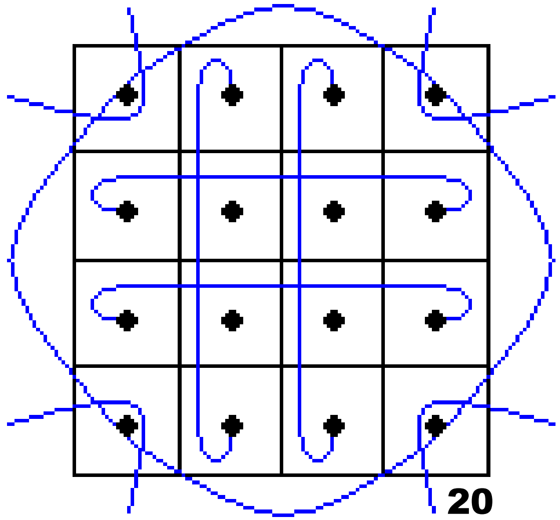
Szomszédosság – adjacencia

- **Def:** Ha egy Karnaugh táblában két szomszédos (adjacent) cella csak *egyetlen* változó értékében különbözik (egységnyi távolság, $d_{Hamming}=1$)!
- Pl. $Y_3 = \overline{A} \cdot B \cdot C$ és $Y_7 = A \cdot B \cdot C$

			C	C	
				B	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

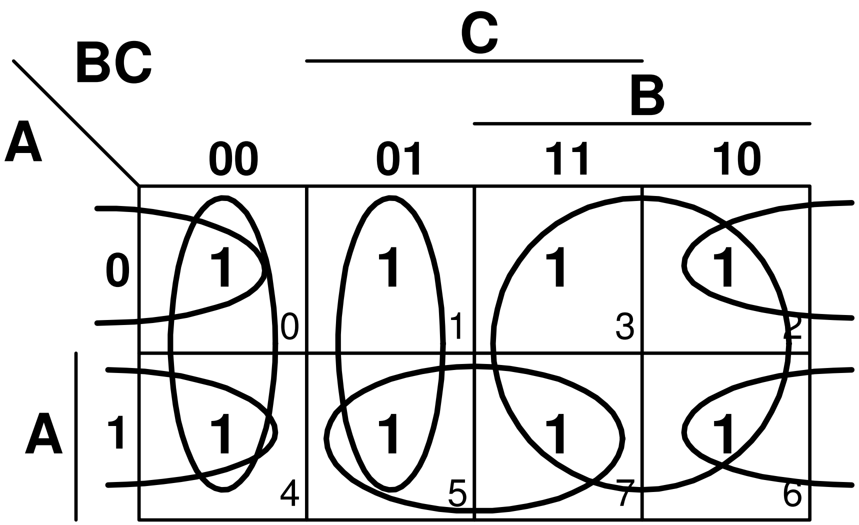
- **Tömbösítés (~tömörítés) szabályai:**
  - **2^n** (n=0,1,2..) term vonható be egy tömbbe,
  - Egyetlen term több tömbben is szerepelhet (*átlapolódás* lehetséges)
  - Egyik tömb, a másikat nem tartalmazhatja teljes mértékben, (redundancia)
  - Mindig a *lehető legnagyobb lefedéseket* keressük, és haladjunk a legkisebb méretű tömbök/lefedések felé
  - *Don't care* ('-') kimeneti függvényértékeket a jobb (*optimálisabb*) lefedésnek megfelelően kell megválasztani (NTSH)
  - Egymás mellett lévő (*adjacens*) sorokra és oszlopokra érvényes.
  - A csak egyetlen hurokban lévő '1'-eseket (DNF) *megkülönböztetett minterm-nek* nevezzük
  - *Lényeges prímimplikáns*: amely legalább egy megkülönböztetett mintermet helyettesít (DNF)



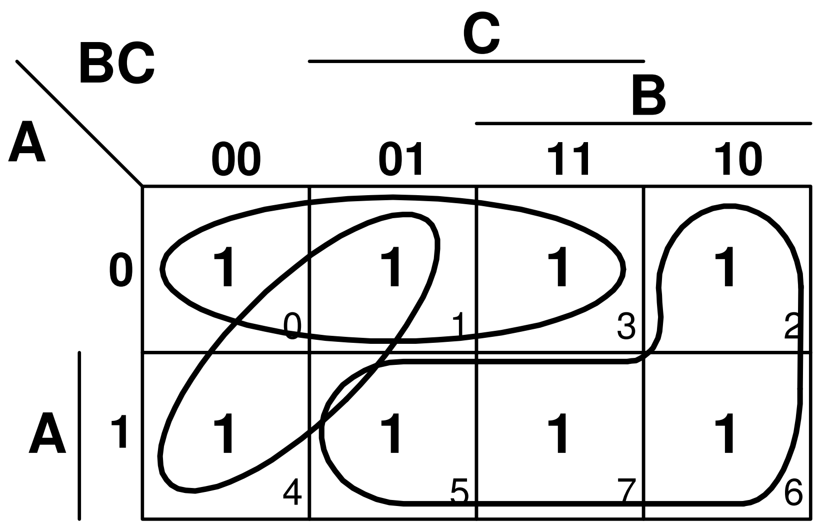
Példa: Karnaugh táblák egyszerűsítése - tömbösítések

- érvényes

érvénytelen



Nem összes, de lehetséges egyszerűsítések - érvényes



Átlós, és nem 2^n számú ‘1’-es lefedés (DNF) érvénytelen

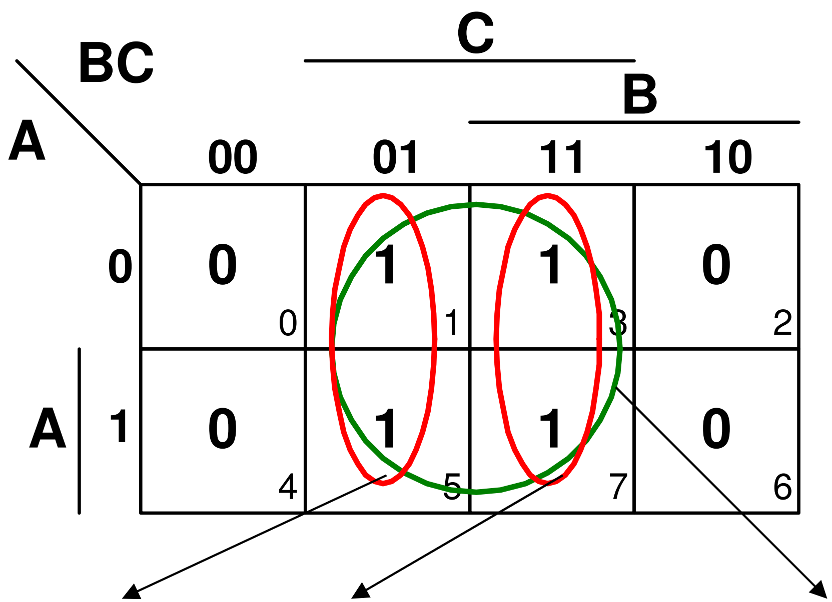
Lehetséges módszerek a Karnaugh tábla értelmezésére:

- M1: $Y(DNF)$ '1'-k lefedésével képzett (**normál**, eddig használt ált. módszer)
- M2: $\overline{Y}(DNF)$ '0'-k lefedésével képzett inverz függvény felírás

- M3: $Y(KNF)$ '0'-k lefedésével képzett
- M4: $\overline{Y}(KNF)$ '1'-k lefedésével képzett inverz függvény felírás

3.1) Karnaugh - grafikus módszer: példa **DNF** szerint

- Karnaugh/Veitch diagram
  - Tömbösítés szabályainak betartása!
- Példa:



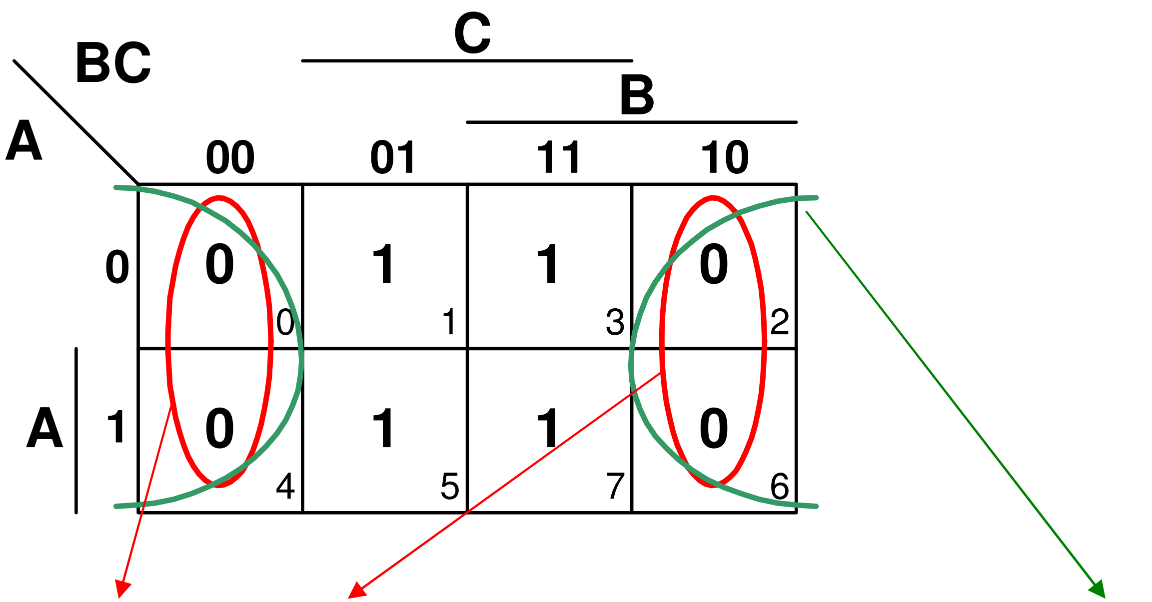
$$F = \overline{B} \cdot C + B \cdot C = C \cdot (\overline{B} + B) \Leftrightarrow C$$

Lehetséges, de nem tömör összevonások

Legtömörebb összevonás

3.2.) Karnaugh - grafikus módszer: példa **KNF** szerint

- Karnaugh/Veitch diagram
  - Tömbösítés szabályainak betartása!
- Példa:

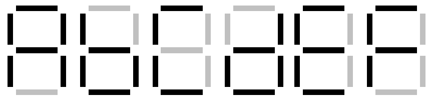


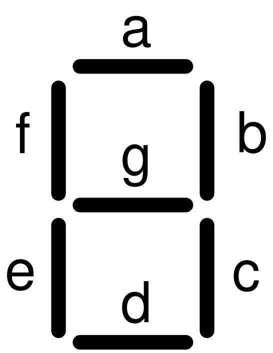
$$F = (B+C)\cdot(\overline{B}+C) = \overline{B}B + BC + \overline{B}C + CC \Leftrightarrow C$$

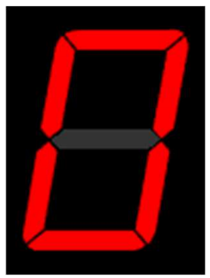
Lehetséges, de nem tömör összevonások

Legtömörebb összevonás

Példa 1: 7-szegmenses dekóder áramkör tervezése (DNF szerint)

- ***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)
- Karnaugh tábla: **TSH!**

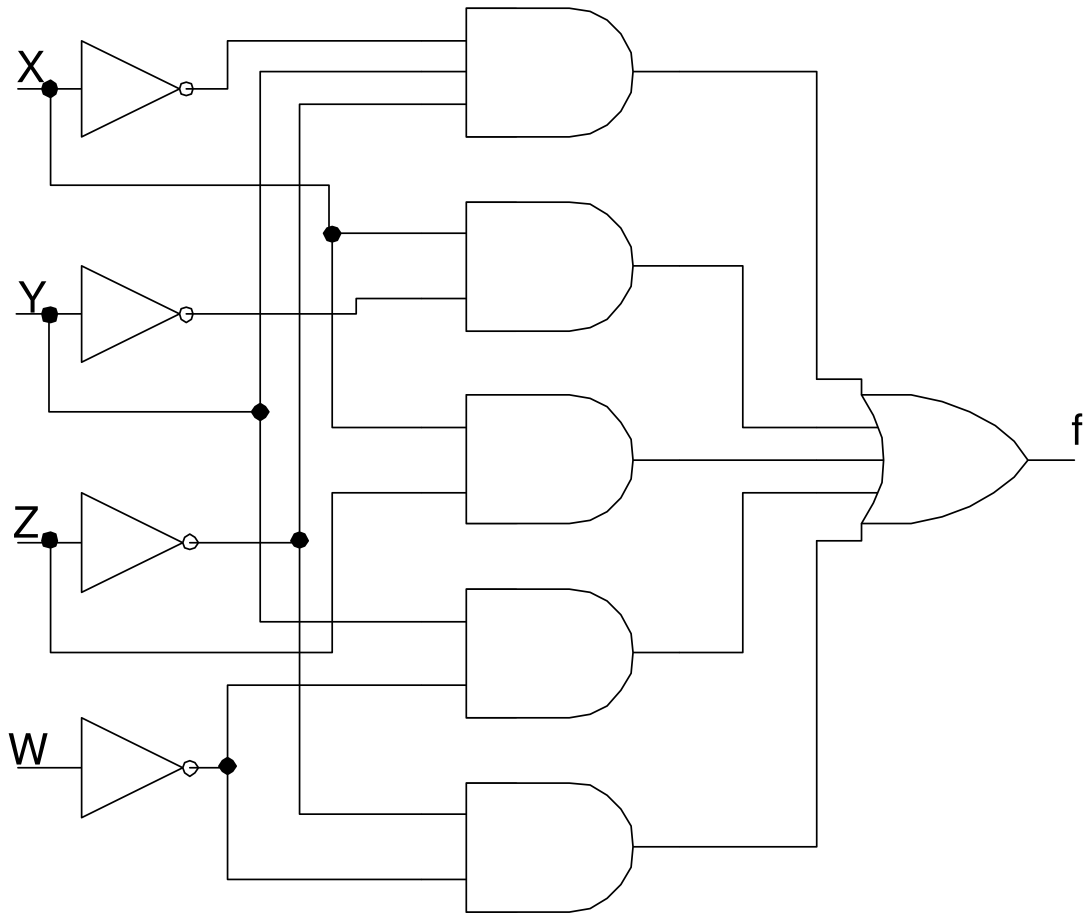
XY \ ZW	00	01	11 (Z)	10 (Z)	
00	**1** [0]	**0** [1]	**0** [3]	**0** [2]	
01	**1** [4]	**1** [5]	**0** [7]	**1** [6]	Y
11 (X)	**1** [12]	**0** [13]	**1** [15]	**1** [14]	Y
10 (X)	**1** [8]	**1** [9]	**1** [11]	**1** [10]	
		W	W		

sor	X	Y	Z	W	f
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 **f** kimeneti függvény:

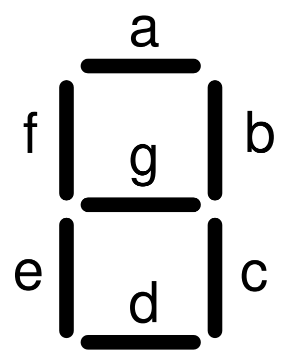
$$f(X,Y,Z,W) = \overline{Z}\cdot\overline{W} + X\cdot\overline{Y} + Y\cdot\overline{W} + X\cdot Z + \overline{X}\cdot Y\cdot\overline{Z}$$

Példa 1: A 7-szegmenses dekóder logikai áramköri realizációja (folyt)



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

Példa 2: 7-szegmenses dekóder áramkör tervezése

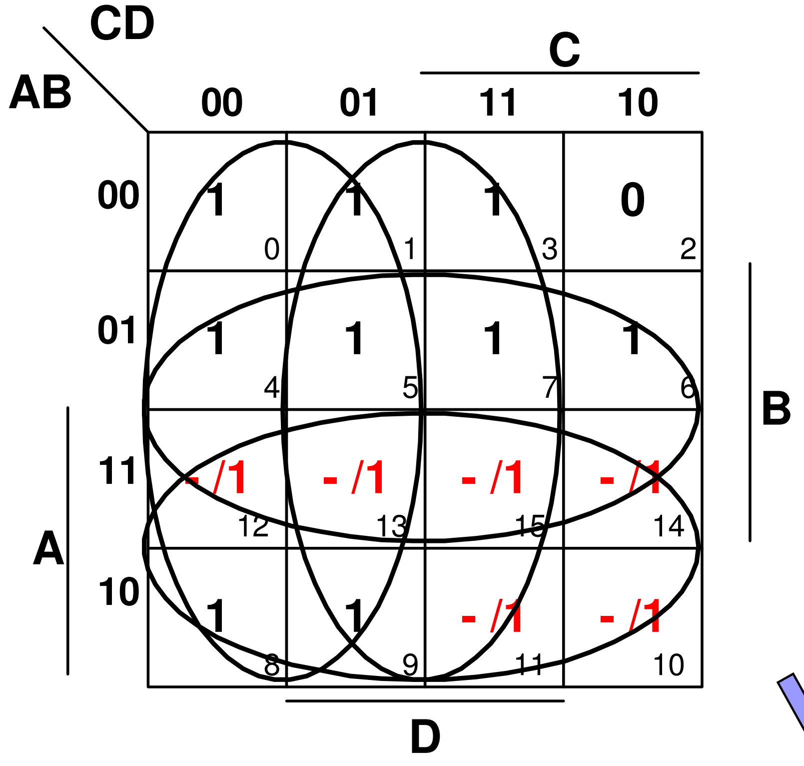


- Csak számjegyeket (0-9) megjelenítésére
  - BCD: Binárisan kódolt decimális számokra
- Nemzetközi elnevezései a szegmenseknek: (a, b, c, d, e, f, g)
  - 10 érték (4 biten ábrázolható): F(A,B,C,D)
- <u>**NTSH**</u>: használjunk <u>Nem Teljesen Specifikált Hálózatot</u>
  - (igazságtábla F kimeneti függvényértékeiben **don't care** '_' nem definiált állapotok is lehetnek)
  - Feladat:

$$F^{n=4} = \sum_{i=0}^{2^n-1}(0,1,3,4,5,6,7,8,9)\ ,\ \overbrace{(10,11,12,13,14,15)}$$

Példa 2: 7-szegmenses dekóder tervezése (folyt)

- Igazságtábla (**c** szegmensre)
- Karnaugh tábla: **NTSH!**



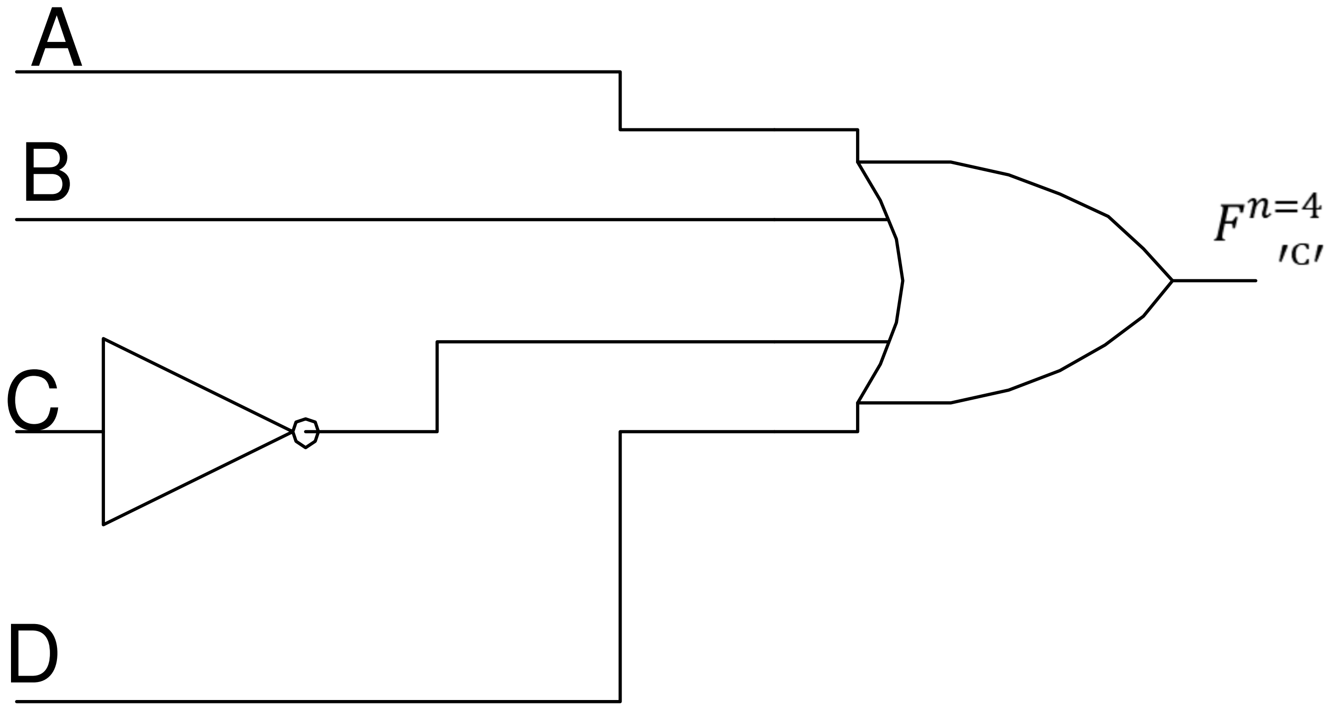
sor	A	B	C	D	c
0	0	0	0	0	**1**
1	0	0	0	1	**1**
2	0	0	1	0	**0**
3	0	0	1	1	**1**
4	0	1	0	0	**1**
5	0	1	0	1	**1**
6	0	1	1	0	**1**
7	0	1	1	1	**1**
8	1	0	0	0	**1**
9	1	0	0	1	**1**
10	1	0	1	0	–
11	1	0	1	1	–
12	1	1	0	0	–
13	1	1	0	1	–
14	1	1	1	0	–
15	1	1	1	1	–

- Kapott **F** kimeneti függvény:

$$F^{n=4}_{'c'}(A,B,C,D) = A + B + \overline{C} + D$$

Példa 2: 7-szegmenses dekóder logikai áramköri realizációja (BCD)

(c szegmensre)



$$F^{n=4}_{'c'}(A,B,C,D)=A+B+\overline{C}+D$$

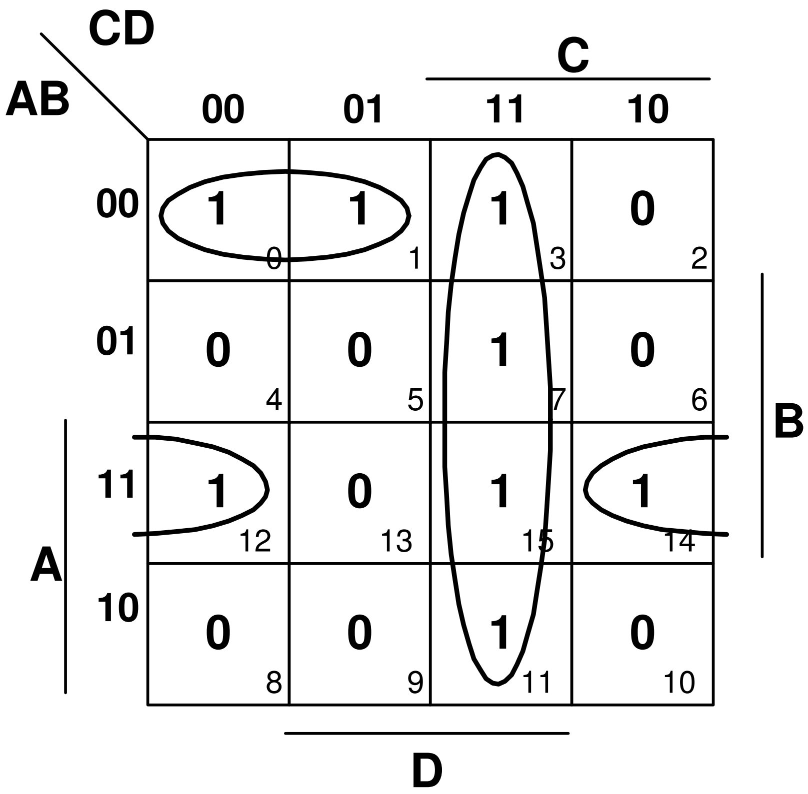
3.3.) Normálformák (NF) + Karnaugh táblák

Ismétlés:

- DNF: Diszjunktív Normál Forma
  - mintermek (szorzattermek) *VAGY* kapcsolata
- KNF: Konjunktív Normál Forma
  - Maxtermek (összegtermek) *ÉS* kapcsolata

Példa 1: Diszjunktív Normál Forma

- Legyen: $\overset{n=4}{F} = \sum_{i=0}^{2^n-1}(0,1,3,7,11,12,14,15)$ **TSH!**
- Karnaugh tábla:



- Kapott F függvény:

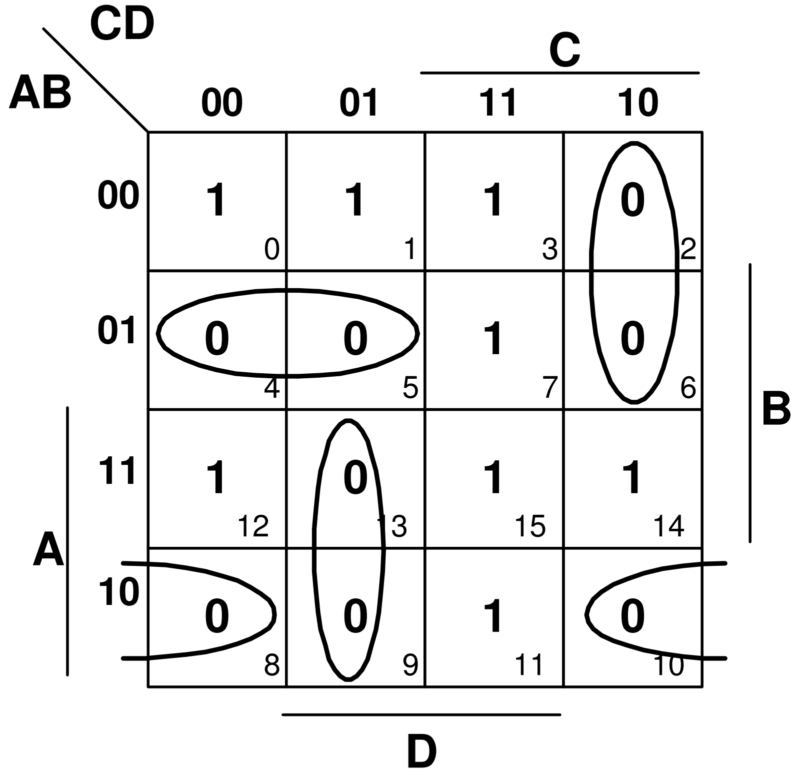
$$'1' = F^4(A,B,C,D) = C \cdot D + \overline{A} \cdot \overline{B} \cdot \overline{C} + A \cdot B \cdot \overline{D}$$

Példa 2: Konjunktív Normál Forma

- Legyen: $\overset{n=4}{F} = \prod_{i=0}^{2^n-1}(2,4,5,6,8,9,10,13)$

TSH!

- Karnaugh tábla:

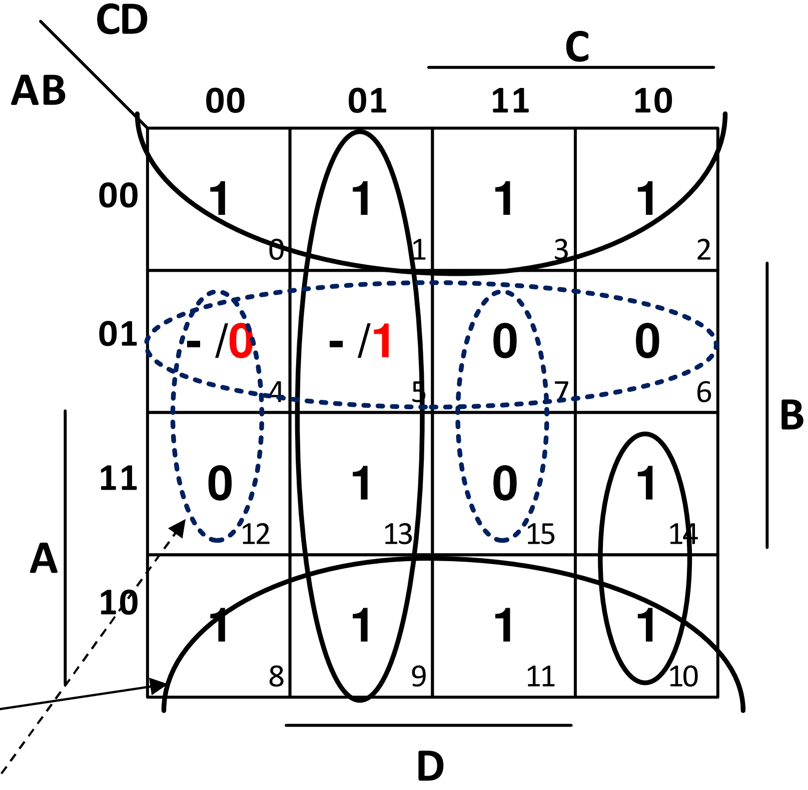


- Kapott F függvény:

$$'0' = F^4(A,B,C,D) = (A+\overline{C}+D)\cdot(A+\overline{B}+C)\cdot(\overline{A}+C+\overline{D})\cdot(\overline{A}+B+D)$$

Példa: NTSH

- Legyen: $\overset{n=4}{F} = \sum_{i=0}^{2^n-1}(0,1,2,3,8,9,10,11,13,14)+(4,5)$ **NTSH!**
- Karnaugh tábla:



- Kapott F_d függvény (DNF) / F_k függvény (KNF):

$$'1' = F_d = \overline{B} + \overline{C}D + AC\overline{D}$$

$$'0' = F_k = (A+\overline{B})\cdot(\overline{B}+C+D)\cdot(\overline{B}+\overline{C}+\overline{D})$$

F_d itt egyszerűbb alakot és kapcsolást realizál

Számjegyes minimalizálás (Quine-McCluskey módszer)

4.) Számjegyes minimalizálás (**Quine-McCluskey módszer)**

- Ha az egyszerűsítés során a mintermeket a Karnaugh táblás ábrázolás helyett az alsó ***indexekkel*** helyettesítünk és segítségükkel számolunk, akkor olyan minimalizáló eljáráshoz juthatunk, amelynek végrehajthatósága nem függ a logikai változók számától.
- *Index*: decimális szám (bináris változó-kombinációk decimális értéke) segítségével:
  - Szomszédosság vizsgálat (3 feltétel!), majd
  - Prímimplikáns képzés.

A.) Szomszédosság: 2^n hatvány (*szükséges, de nem elégséges feltétel!*)

- Két term szomszédos, ha a két m_i minterm különbsége 2-egész hatványa (2^n)
  - $$\begin{array}{rl} & 0110 \quad (\ \ 6) \\ - & 0010 \quad (-2) \\ \hline & 0100 \quad (\mathbf{4=2\hat{}2}) \end{array} \qquad \left.\begin{array}{l} m_6^4 = \overline{A}BC\overline{D} \\ m_2^4 = \overline{A}\,\overline{B}C\overline{D} \end{array}\right\} \rightarrow \overline{A}C\overline{D} \qquad \text{szomszédosak}$$
  - $$\begin{array}{rl} & 0100 \quad (\ \ 4) \\ - & 0010 \quad (-2) \\ \hline & 0010 \quad (\mathbf{2=2\hat{}1}) \end{array} \qquad \left.\begin{array}{l} m_4^4 = \overline{A}B\overline{C}\,\overline{D} \\ m_2^4 = \overline{A}\,\overline{B}C\overline{D} \end{array}\right\} \rightarrow \begin{array}{l} \text{2\^{}n feltétel teljesül, de} \\ \text{nem szomszédosak} \end{array}$$

B.) Szomszédosság: Bináris súly (*szükséges, de nem elégséges feltétel!*)

- Ha két minterm szomszédos, akkor az egyiknek megfelelő bináris szám eggyel és csakis eggyel több '1'-est tartalmaz, mint a másiké.
  - ```
     0110  ( 6)
    -0010  (-2)
     0100  (4=2^2)
    ```
    $$\left.\begin{array}{l} m_6^4 = \overline{A}BC\overline{D} \\ m_2^4 = \overline{A}\,\overline{B}C\overline{D} \end{array}\right\} \rightarrow \overline{A}C\overline{D}$$
    '1'-esek száma eggyel nagyobb
- Tehát ha a mintermek *szomszédosak*, akkor a *bináris súlyaik* különbsége 1.
  - Megj: előző $m_4 - m_2$ mintermek esetén pont ez nem teljesült!
- Azonban a szomszédosság A.) és B.) teljesülése esetén még nem egyértelmű:
  - ```
     1001  ( 9)
    -0111  (-7)
     0010  (2=2^1)
    ```
    $$\left.\begin{array}{l} m_9^4 = A\overline{B}\,\overline{C}D \\ m_7^4 = \overline{A}BCD \end{array}\right\} \rightarrow$$
    Nem szomszédosak!

C.) Szomszédosság: nagyobb bináris súly decimális indexe is nagyobb

(*szükséges, de nem elégséges feltétel!*)

- A.)-ban az $m_6 - m_2$ feltételre ez még igaz.
  - 0**11**0 (6) #'1'=2
−00**1**0 (−2) #'1'=1
0**1**00 (4=2^2)

$$\left.\begin{array}{l} m_6^4 = \overline{A}BC\overline{D} \\ m_2^4 = \overline{A}\,\overline{B}C\overline{D} \end{array}\right\} \rightarrow \overline{A}C\overline{D}$$

- Azonban a B.) pontban $m_9 - m_7$ feltételre ez az állítás már hamis.
  - **1**00**1** (9) #'1'=2
−0**111** (−7) #'1'=3
0010 (2=2^1)

$$\left.\begin{array}{l} m_9^4 = A\overline{B}\,\overline{C}D \\ m_7^4 = \overline{A}BCD \end{array}\right\}$$

Szomszédosság: 3-feltétel együttes teljesülése

- Bizonyítható, hogy az A.), B.) és C.) (**szükséges, de nem elégséges**) feltételek együttes teljesülése esetén lesz pontosan a **két minterm szomszédos**:
  - A.) indexek különbsége 2^n hatványa, és
  - B.) bináris súlyuk különbsége 1, és
  - C.) a nagyobb bináris súlyú minterm decimális indexe is nagyobb!

Prímimplikáns-képzés lépései:

- **I. oszlop**: felsorolt decimális minterm indexek csoportosítása bináris súlyonként a páronkénti szomszédosság vizsgálathoz (a különböző bin. súlyú csoportokat aláhúzással választjuk el.)
  - \+ Kevesebb összehasonlítás a párba válogatáskor
- **II. oszlop**: a párba válogatást úgy végezhetjük el, hogy a bináris „súly” csoportok minden egyes számjegyét *kivonjuk* a *következő* egyel nagyobb súlyú csoport minden egyes számjegyéből.
  - Ha találunk két olyan számot, amelyek különbsége 2^n oda pipát teszünk √. (mintermet már tartalmazza a pár).
  - Összevont számpár elemeit növekvő sorrendben írjuk fel, (zárójelben a decimális különbségüket).
  - A decimális különbség 2-es alapú logaritmusa jelöli ki az elhagyható változó helyiértékét
- **III. illetve további oszlop(ok):** kialakítását a II. oszlopéval azonosan kell végezni!
  - minden elemet összehasonlítunk a következő csoport minden elemével
  - Két egyszerűsített szorzat akkor lesz szomszédos, ha a decimális különbségeik páronként megegyeznek.
- **Végül**: a nem egyszerűsíthető / primimplikáns elemeket *betűkkel jelöljük* meg → prímimplikáns tábla és/vagy segédfüggvény felírása

Egyszerűsített alak lehetséges megadási módjai

- **Prímimplikáns tábla**: ha ránézésre megállapíthatók melyek a lényeges prímimplikánsok (melyek az összes mintermet lefedik)
- **Segédfüggvény (S)**: ha ránézésre nem állapítható meg a prímimplikáns tábla alapján, vagy többváltozós bonyolult függvényt kell minimalizálni. (NTSH-nál az összes lehetséges optimális megoldást megadja.)

Prímimplikáns tábla felírása

- Az optimális lefedést decimális indexek alapján kell elvégezni prímimplikáns tábla segítségével:
  - az egyes mintermeket mely (megbetűzött) prímimplikánsok tartalmazzák, vagy „fedik le”.
  - Táblázat kitöltésekor egy-egy prímimplikánssal kijelölt sornak abba a sorába cellájába kell ‘*’-ot tenni, amelyhez tartozó mintermet az illető prímimplikáns tartalmazza → **lényeges prímimplikáns(ok) (nem elhagyható(k))**
    - van olyan minterm, amely oszlopa alatt csak egyetlen ‘**x**’ szerepel.

Példa:

Lényeges prímimplikáns ok

sor	minterm / Prímimplik.	0	1	3	7	11	12	14	15
*	a	x	x						
	b		x	x					
*	c						x	x	
	d							x	x
*	e			x	x	x			x

Segédfüggvény (S)

- Bonyolultabb (sokváltozós) prímimplikáns táblázatok esetén nehéz lehet felírni (vagy ránézésre nem állapítható meg) a legegyszerűbb végleges alak, tehát nem állapíthatóak meg egyértelműen mely lényeges prímimplikánsok szerepelnek a függvényben. Ekkor:
  - **Segédfüggvényt** lehet használni a felíráshoz, ahol **S=1** a prímimplikánsok *ÉS kapcsolatát* kell képezni (prímimplikáns tábla *oszlopában* lévő prímimplikáns tagok *VAGY kapcsolatban* vannak).
  - „Beszorzás” után meg kell keresni a legkevesebb tényezőt tartalmazó szorzatot (azaz a betűvel jelölt prímimplikáns tago(ka)t) az ‘S’ segédfüggvényben, és ez(ek) segítségével kell felírni az egyszerűsítendő függvény DNF alakját.
  - Végül azokat a **(*lehető legkevesebb számú*) *prímimplikánsokat* kell VAGY kapcsolatba hozni a legegyszerűbb DNF alakban**, amelyeknek megfelelő változók ebben a kapott szorzatban szerepelnek (hiszen ezek együttesen jelentik S=1 -et). ***A lényeges prímimplikánsok logikai összege*** a logikai F függvényben szereplő összes mintermet lefedi, tehát felírható segítségükkel.

Quine-McCluskey módszer

- Szomszédosság szükséges feltételei:
  - Decimális indexek különbsége 2^n kell legyen (*szükséges, de nem elégséges feltétel*!)
    - Pl: i: 6-2=4 (szomszédos), de i:10-6=4 (nem szomszédos)
  - Bináris súlyuk különbsége =1. (Hamming távolság)
    - Pl:

```
0111 (7)   v.   1001 (9)
0011 (3)        0111 (7)
--------        --------
0x00            xxx0
```

jó rossz

(szükséges, de nem elégséges feltétel!)

  - A nagyobb decimális indexűnek kell nagyobb bináris súllyal szerepelnie! (szükséges, de nem elégséges feltétel!)

	00	01	11	10
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}

Példa: Számjegyes minimalizálásra **(Quine-McCluskey módszer)**

- Oldjuk meg a következő feladatot a Quine-McCluskey módszerrel
- Ha adott az F függvény DNF alakban:

$$\overset{n=4}{F}(A,B,C,D)=\sum_{i=0}^{2^n-1}(0,1,3,7,11,12,14,15)$$

TSH!

- Karnaugh tábla:
  - csak szemléltetés végett

AB \ CD	00	01	11	10
00	1 (0)	1 (1)	1 (3)	0 (2)
01	0 (4)	0 (5)	1 (7)	0 (6)
11	1 (12)	0 (13)	1 (15)	1 (14)
10	0 (8)	0 (9)	1 (11)	0 (10)

C: CD = 11, 10; B: AB = 01, 11; A: AB = 11, 10; D: CD = 01, 11

Számjegyes minimalizálás **Quine-McCluskey módszer I.lépés**

- **I. oszlop:** Csoportosítás bináris súly szerint:
  - ahol a kimeneti értékük ’1-s’ volt.

$$F^{n=4} = \sum_{i=0}^{2^n-1}(0,1,3,7,11,12,14,15)$$

Minterm	Bináris alak	
0	0000	[#0 bináris súly]
1	0001	[#1 bináris súly]
3	0011	[#2 bináris súly]
12	1100	
7	0111	[#3 bináris súly]
11	1011	
14	1110	
15	1111	[#4 bináris súly]

bináris súly szerinti

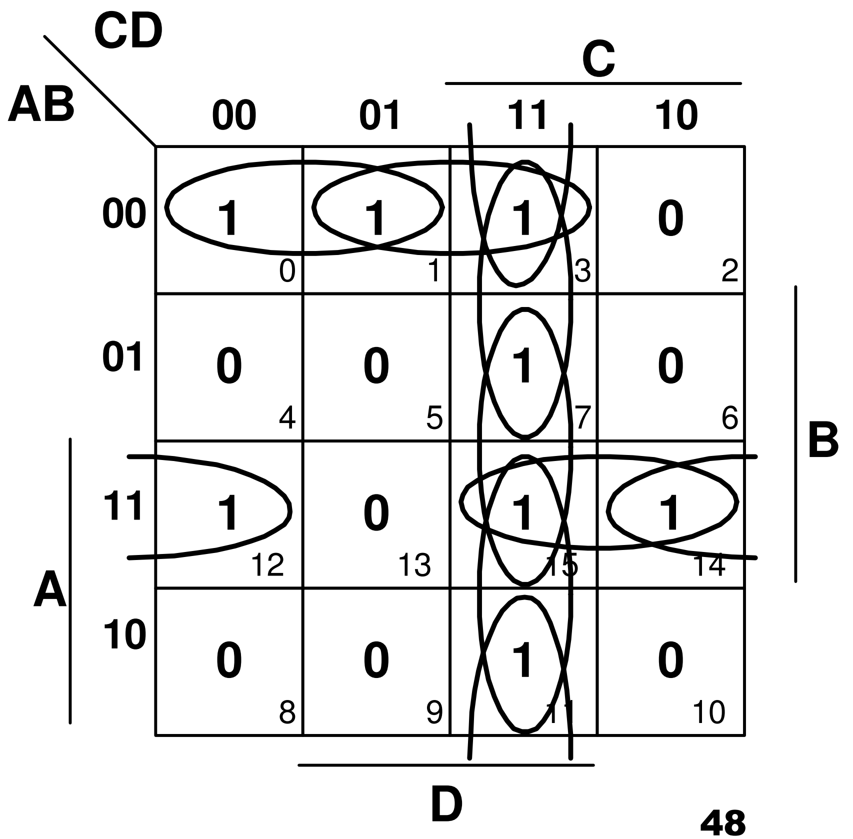
csoportképzések = vonallal elválasztva

Számjegyes minimalizálás
Quine-McCluskey módszer II.lépés

- II. Összes létező szomszédos **kételemű** lefedő tömb (hurok) összevonása (Karnaugh tábla csak szemléltetés végett)

Minterm	Decimális különbség
0,1	(1)
1,3	(2)
3,7	(**4**)
3,11	(8)
12,14	(2)
7,15	(8)
11,15	(**4**)
14,15	(1)

II. oszlop



Számjegyes minimalizálás
Quine-McCluskey módszer III.lépés

- III. Összes létező szomszédos kettesekből képzett **négyelemű** lefedő tömb összevonása
(Karnaugh tábla csak szemléltetés végett)

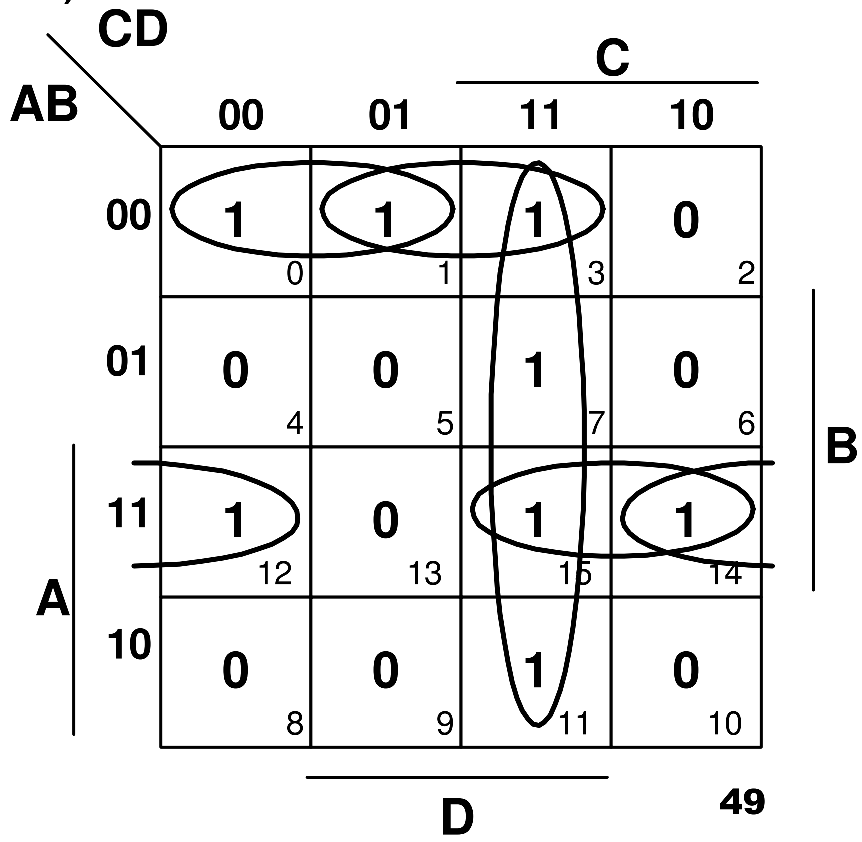
Minterm	Decimális különbség	
0,1	(1)	a
1,3	(2)	b
3,7	(4)	√
3,11	(8)	√
12,14	(2)	c
7,15	(8)	√
11,15	(4)	√
14,15	(1)	d

III. oszlop

Négyes Összevonás

3,7,11,15 **(4,8)** e

prímimplikáns betűzések



Számjegyes minimalizálás
Quine-McCluskey módszer IV. lépés

- IV. Prímimplikáns tábla felírása a megmaradt összevonásokkal (III. lépés alapján)

sor	minterm / Prímimplik.	0	1	3	7	11	12	14	15
*	a **0,1** (1)	x	x						
	b **1,3** (2)		x	x					
*	c **12,14** (2)						x	x	
	d **14,15** (1)							x	x
*	e **3,7,11,15** (4,8)			x	x	x			x

* : ahol egy adott mintermhez tartozó oszlopban csak egy ‘**x**’ van, az a sor jelöli a **lényeges prímimplikánst** (ahol az implikáns tovább már nem egyszerűsíthető!). Az a sor nem elhagyható!

Számjegyes minimalizálás
Quine-McCluskey módszer V.lépés

- V. Lényeges prímimplikánsokból képzett kimeneti függvény megadása (IV. lépés alapján):

(0,1): a	0000 0001	⇨	0001
(12,14): c	1100 1110	⇨	1100
(3,7,11,15): e	0011 0111 1011 1111	⇨	1011

A mintermen belüli egyszerre 0/1 tagok kiesnek!

- Tehát a kimeneti minimalizált F függvény a következő:

$$F = 0001 + 1100 + 1011 \Rightarrow \quad F = \overline{A}\cdot\overline{B}\cdot\overline{C} + A\cdot B\cdot\overline{D} + C\cdot D$$

Prímimplikáns tábla alapján a segédfüggvény (S) felírása:

- S = 1 pontosan akkor, ha
  - *(m_0 lefedéséhez) a* prímplikáns ÉS,
  - *(m_1 lefedéséhez) a* VAGY *b* prímimplikáns ÉS,
  - *(m_3 lefedéséhez) b* VAGY *e* prímimplikáns ÉS,
  - *(m_7 lefedéséhez) e* prímimplikáns ÉS,
  - *(m_{11} lefedéséhez) e* prímimplikáns ÉS,
  - *(m_{12} lefedéséhez) c* prímimplikáns ÉS,
  - *(m_{14} lefedéséhez) c* VAGY *d* prímimplikáns ÉS,
  - *(m_{15} lefedéséhez) d* VAGY *e* prímimplikáns.

$$S = a \cdot (a + b) \cdot (b + e) \cdot e \cdot e \cdot c \cdot (c + d) \cdot (d + e) = 1$$

Segédfüggvény felírása

- Ebben a feladatban *ránézésre megállapítható* volt a prímimplikáns tábla alapján, ahogy a segédfüggvénnyel felírt alakban is:

Legegyszerűbb alak a prímimplikánsból „lefedi” a tagot

Beszorzás elvégzése

$$S = a \cdot (a+b) \cdot (b+e) \cdot e \cdot e \cdot c \cdot (c+d) \cdot (d+e) =$$

$$= abecd + aecd + abec + \boxed{aec} \Rightarrow aec \Rightarrow$$ Legegyszerűbb DNF alak

$$F = a + c + e = \overline{A}\,\overline{B}\,\overline{C} + AB\overline{D} + CD$$

VAGY kapcsolat

a DNF alakban

(Ugyanazt kaptuk itt, mint a prímimplikáns tábla alapján.)

Quine-McCluskey: NTSH hálózatok esetén

- **NTSH: A közömbös dont'care** függvényértékek megadásakor
  - az összevonásoknál (I.-II.-III. stb. oszlopok felírásánál) a **dont'care értékeket '1'-nek tekintjük**, továbbá
  - a közömbös mintermeket **nem kell figyelembe venni** a *prímimplikáns tábla felírásakor* (hiszen azok lefedéséről nem kell gondoskodnunk!)
  - végül, a legtöbb esetben a primimplikáns tábla alapján felírt **S** segédfüggvény adhat jó megoldást.