Pannon Egyetem
Képfeldolgozás és Neuroszámítógépek Tanszék

Digitális Rendszerek (BSc)

2. előadás: Logikai egyenletek leírása II:
Függvény-egyszerűsítési eljárások

Előadó: Vörösházi Zsolt
voroshazi@vision.vein.hu

Jegyzetek, segédanyagok:

- Könyvfejezetek:
  - http://www.knt.vein.hu -> Oktatás -> Tantárgyak -> Digitális Rendszerek (BSC).

    (01_chapter.pdf)
- Fóliák, óravázlatok .ppt (.pdf)
- Feltöltésük folyamatosan

Függvényminimalizálás

- Általánosan:
  - Függvényminimalizálást a szomszédos mintermek megkeresésével tehetjük meg.
  - A szomszédosság megállapítása után egyszerűsítünk.
  - Minterm → implikáns (egyszerűsíthető) → prímimplikáns (tovább nem egyszerűsíthető)

Függvényegyszerűsítési eljárások

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

1.) Algebrai módszer

- A Boole-algebra azonosságait használjuk fel az egyszerűsítéshez:

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

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

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

2.) Kifejtési módszer:

- Komplexebb függvények esetén egy adott változó értékét először ponáltnak, majd negáltnak definiáljuk, végül pedig az így kiszámított két logikai kifejezést összeadjuk. Ezáltal leegyszerűsödik a függvényminimalizálási feladat.

Példa: kifejtési módszer

- Legyen F_1 függvény a következő:

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

  - Ha A:=1

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

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

  - Ha A:=0

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

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

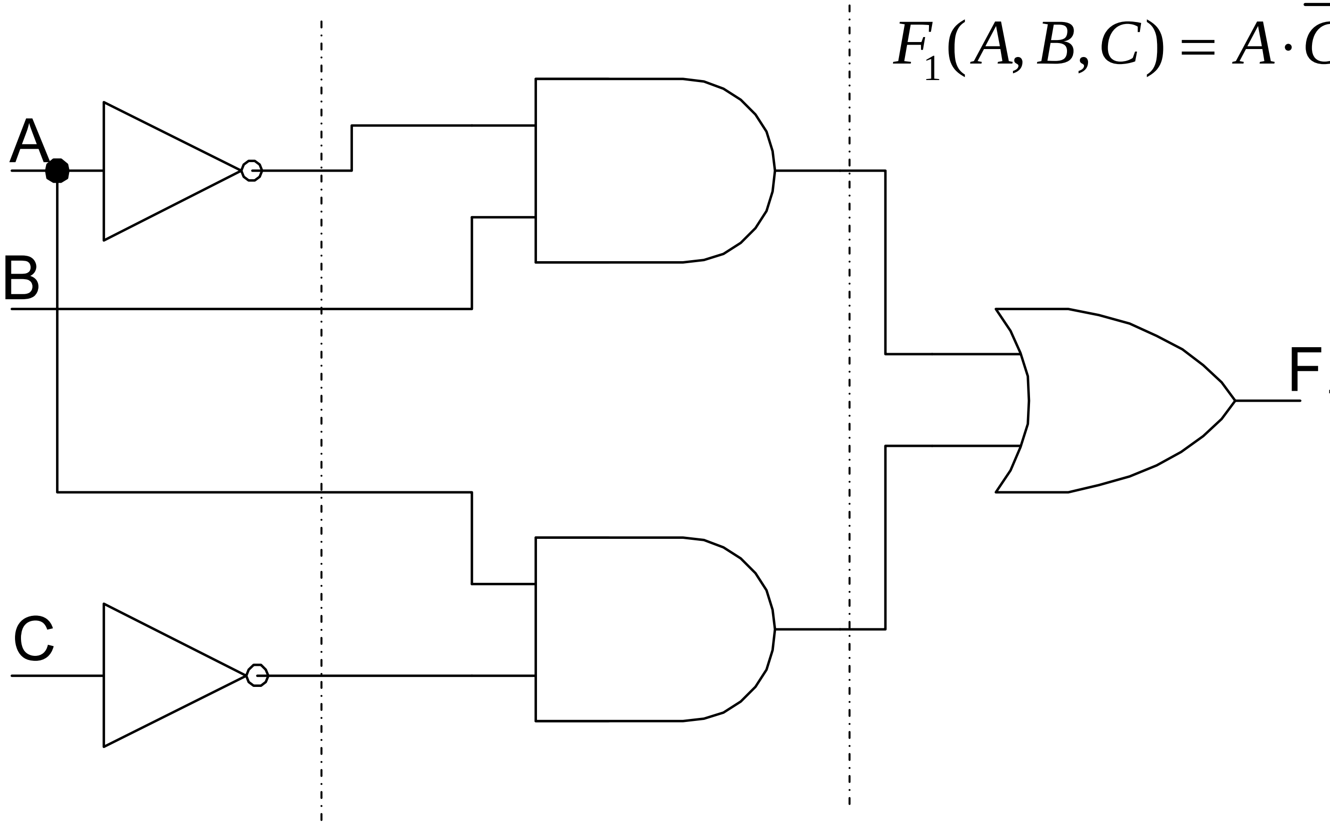
  - Végül összeadjuk a kettőt (egyszerűsített alak):

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

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

Az egyszerűsített függvény logikai áramköri realizációja

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



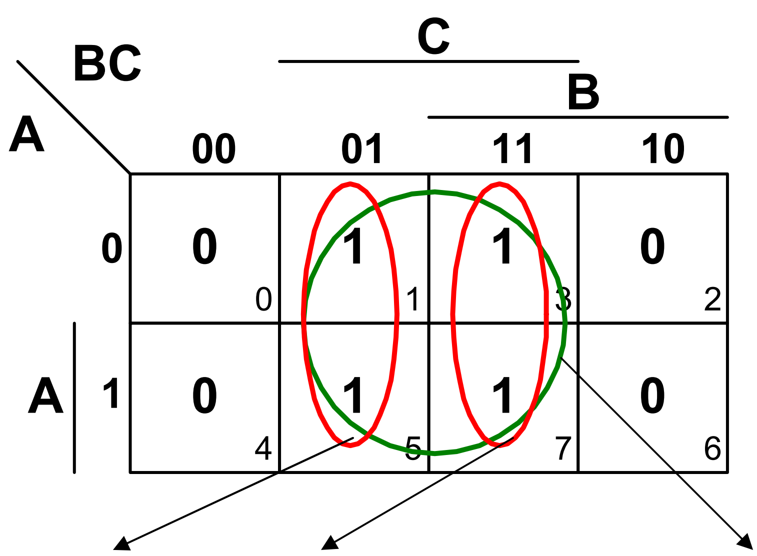
Inverter szint

ÉS kapuk szintje

VAGY kapuk szintje

3.) Grafikus módszer

- Karnough (Veicht) diagramm
  - Tömbösítés szabályainak betartása!
- Példa:

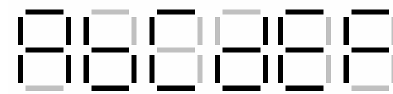


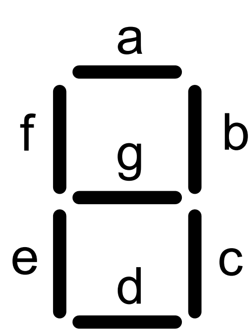
$$\overline{B}\cdot C + B\cdot C = C\cdot(\overline{B}+B) \Leftrightarrow C$$

Lehetséges, de nem tömör összevonások

Legtömörebb összevonás

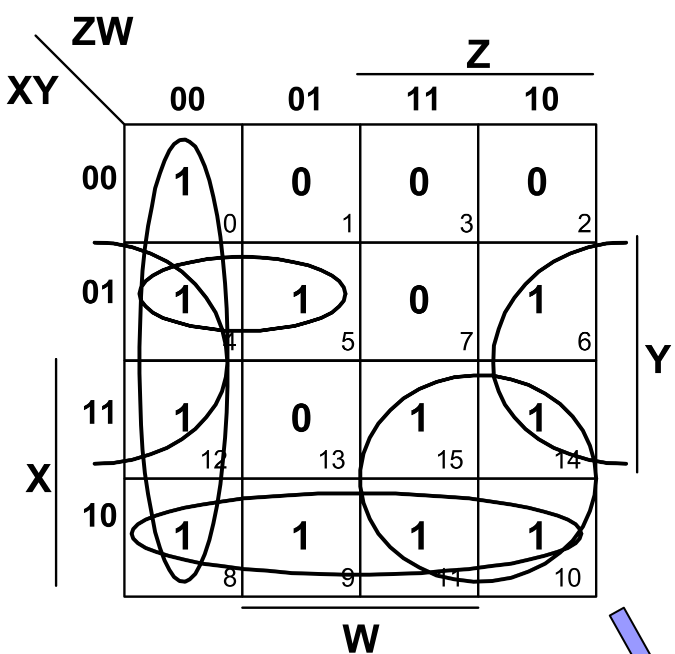
Példa 1: 7-szegmenses dekóder áramkör tervezése

- ***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)
- Karnough tábla:

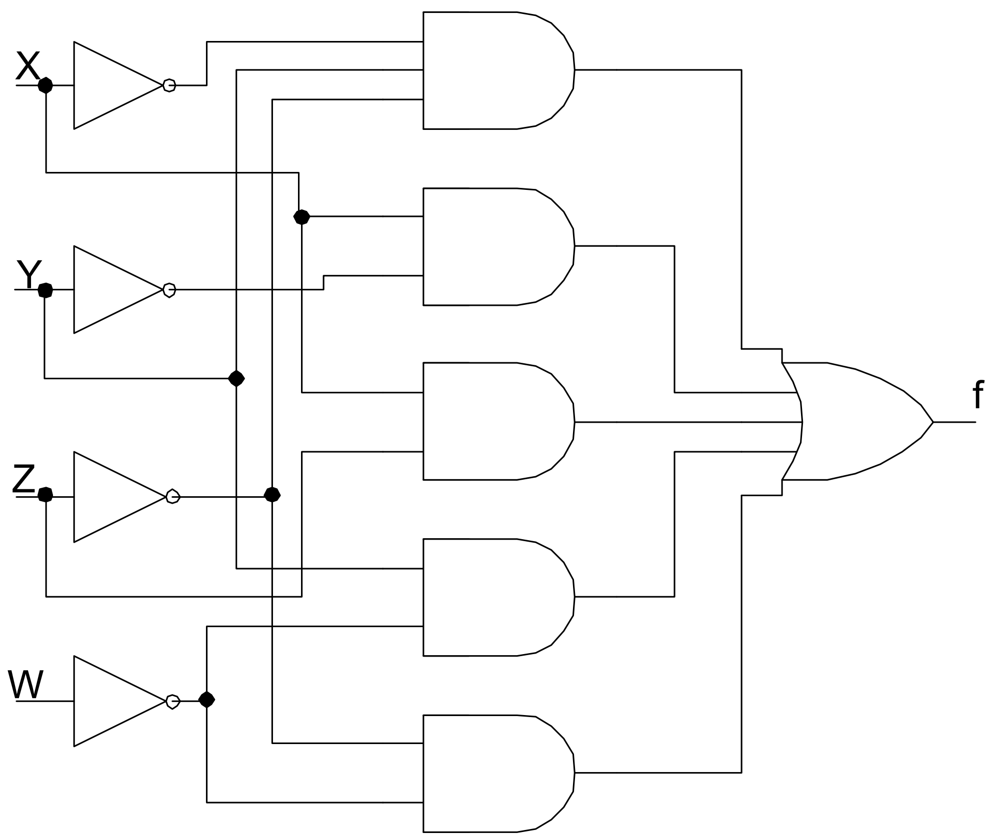


sor	X	Y	Z	W	f
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

- Kapott **f** kimeneti függvény:

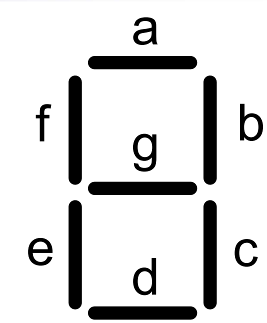
$$f(X,Y,Z,W) = \overline{Z}\cdot\overline{W} + X\cdot\overline{Y} + Y\cdot\overline{W} + X\cdot Z + \overline{X}\cdot Y\cdot\overline{Z}$$

Példa 1: A 7-szegmenses dekóder logikai áramköri realizációja



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

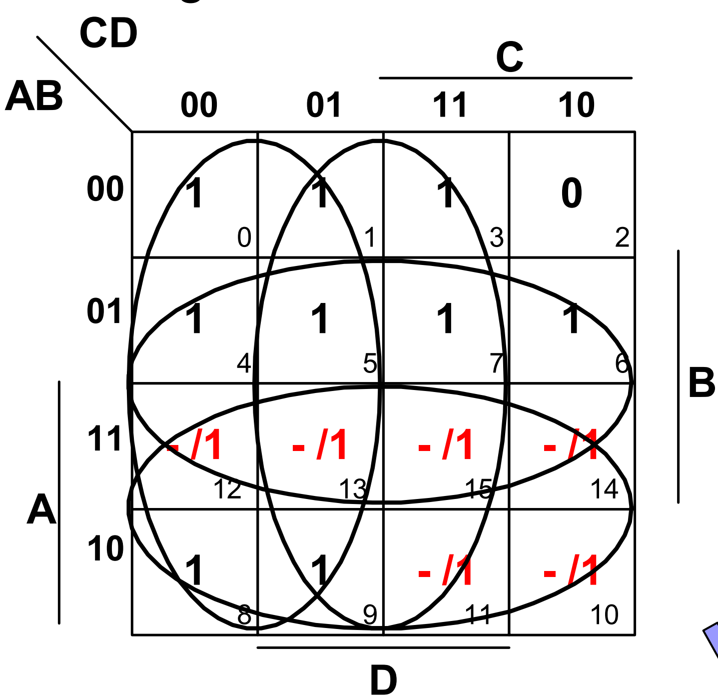
Példa 2: 7-szegmenses dekóder áramkör tervezése



- 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 kimeneti függvényértékeiben lehetnek don’t care ‘-’ definiált állapotok)
  - Feladat: $F^{n=4} = \sum_{i=0}^{2^n-1}(0,1,3,4,5,6,7,8,9) \quad \overbrace{x: 10,11,12,13,14,15}$

Példa 2: 7-szegmenses dekóder tervezése (folyt)

- Igazságtábla (**c** szegmensre)
- Karnough tábla:

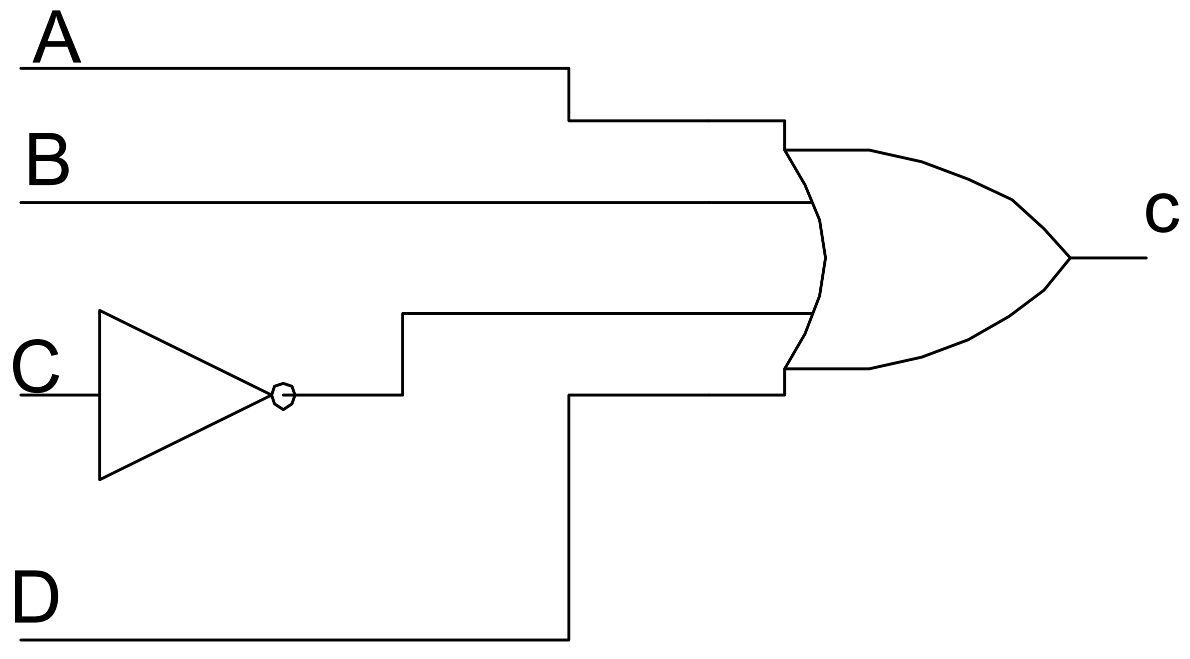


sor	A	B	C	D	c
0	0	0	0	0	**1**
1	0	0	0	1	**1**
2	0	0	1	0	**0**
3	0	0	1	1	**1**
4	0	1	0	0	**1**
5	0	1	0	1	**1**
6	0	1	1	0	**1**
7	0	1	1	1	**1**
8	1	0	0	0	**1**
9	1	0	0	1	**1**
10	1	0	1	0	–
11	1	0	1	1	–
12	1	1	0	0	–
13	1	1	0	1	–
14	1	1	1	0	–
15	1	1	1	1	–

- Kapott **c** kimeneti függvény:

$$c(A, B, C, D) = A + B + \overline{C} + D$$

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



$$c(A,B,C,D) = A + B + \overline{C} + D$$

4.) Normálformák (NF)

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

Példa 1: Diszjunktív Normál Forma

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

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 $_{10}$

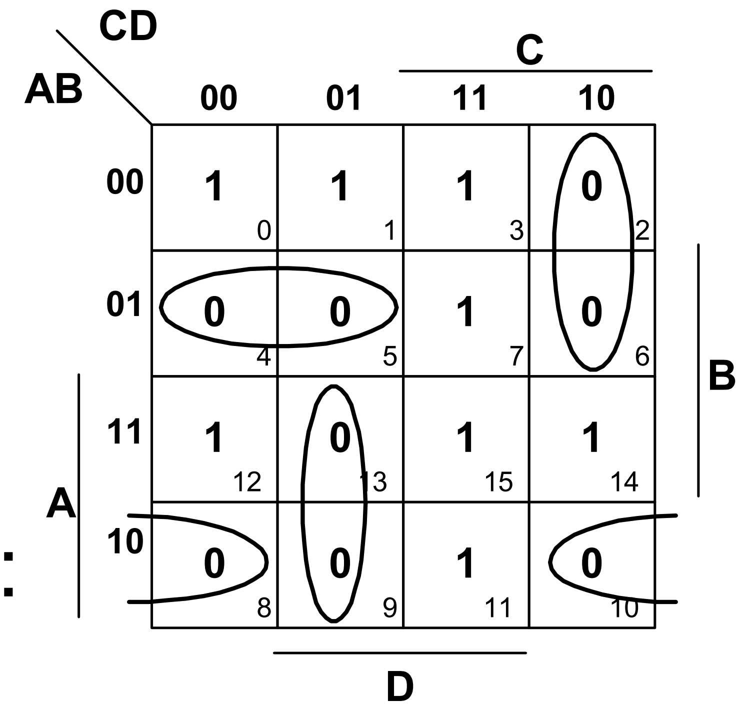
C B A D

- Kapott F függvény:

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

Példa 2: Konjunktív Normál Forma

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



- Kapott F függvény:

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

5.) Számjegyes minimalizálás (**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) vagy 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:

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

- Karnough tábla:

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)

Számjegyes minimalizálás
Quine-McCluskey módszer I.lépés

- Csoportosítás bináris súlyuk szerint:
  - ahol a kimeneti értékük ’1-s’ volt.

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]

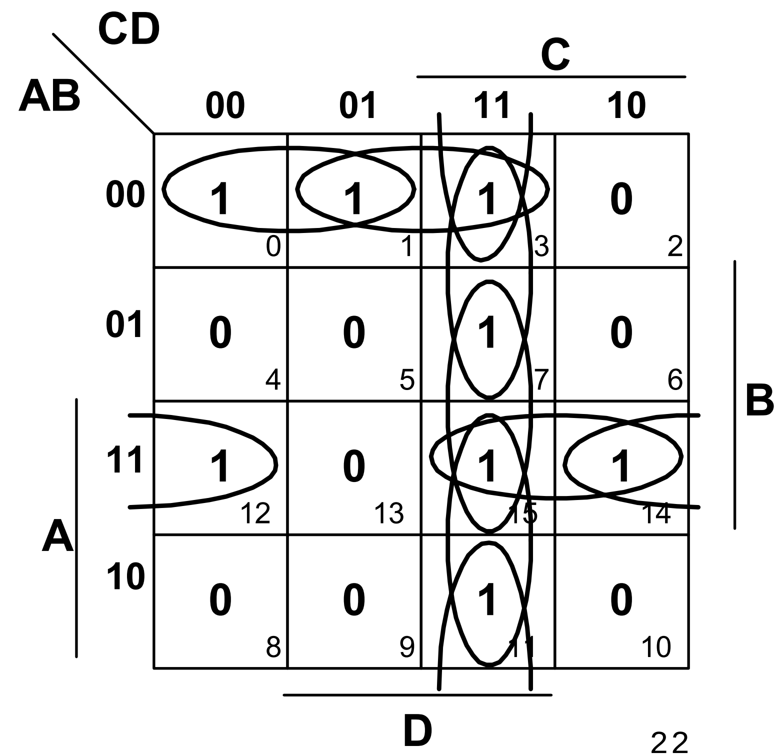
bináris súly szerinti

csoportképzések

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

- II. Összes létező szomszédos **kételemű** lefedő tömb összevonása (Karnough tábla alapján)

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)



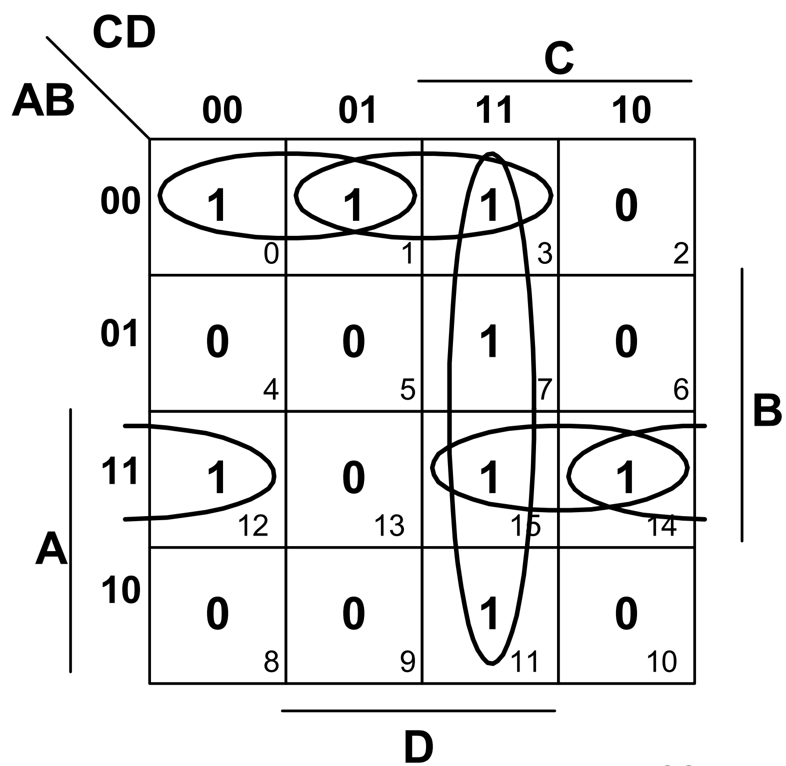
Számjegyes minimalizálás
Quine-McCluskey módszer III.lépés

- III. Összes létező szomszédos kettesekből képzett **négyelemű** lefedő tömb összevonása (Karnough tábla alapján)

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)

Négyes Összevonás

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



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

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

sor		0	1	3	7	11	12	14	15
*	**0,1** (1)	*	*						
	1,3 (2)		*	*					
*	**12,14** (2)						*	*	
	14,15 (1)							*	*
*	**3,7,11,15** (4,8)			*	*	*			*

* : ahol egy adott mintermhez tartozó oszlopban csak egy '*' 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. Prímimplikánsokból képzett kimeneti függvény megadása (IV. lépés alapján):
  - **(0,1):** 0000, 0001 $\Rightarrow$ 000(0/1)
  - **(12,14):** 1100, 1110 $\Rightarrow$ 11(0/1)0
  - **(3,7,11,15):** 0011, 0111, 1011, 1111 $\Rightarrow$ (0/1)(0/1)11

A mintermen belüli egyszerre 0/1 tagok kiesnek!

- Tehát a kimeneti minimalizált F függvény a következő:

$$F = 000(0/1) + 11(0/1)0 + (0/1)(0/1)11 \Rightarrow \quad F = \overline{A}\cdot\overline{B}\cdot\overline{C} + A\cdot B\cdot\overline{D} + C\cdot D$$

- Ajánlott: fejezetek végén a feladatok (Exercises) részek áttekintése.