Pannon Egyetem
Képfeldolgozás és Neuroszámítógépek Tanszék

Digitális Rendszerek (BSc)

1. előadás: Logikai egyenletek leírása I.
Boole-algebra axiómái és tételei

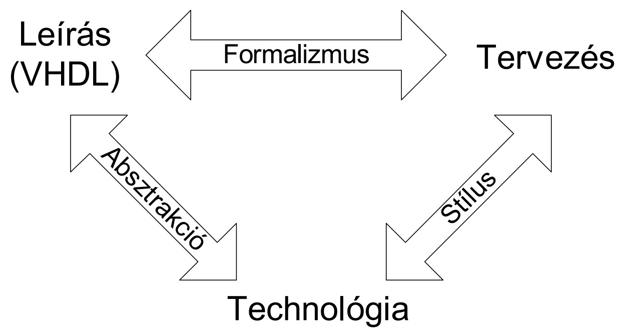
Előadó: Vörösházi Zsolt
voroshazi@vision.vein.hu

Jegyzetek, segédanyagok:

- Könyvfejezetek:
  - http://www.knt.vein.hu
    -> Oktatás -> Tantárgyak -> Digitális Rendszerek (BSC).
    (00_chapter, 01_chapter.pdf)
- Fóliák, óravázlatok .ppt (.pdf)
- Feltöltésük folyamatosan

I. Logikai egyenletek leírása

- Stílus – Absztrakció – Formalizmus



Stílus

- Komplex feladat $\Rightarrow$ egyszerűbb, kezelhető részfeladatokra bontása
  - Szisztematikus
  - Érthető módszerek kellenek

  Pl: programozási stílusok (Top-down, strukturált)

- Jó stílus kialakításának szabályai:
  - Top-down módszer szerinti tervezés
  - Csak kiforrott, biztos technikákat szabad alkalmazni
  - Fontos a dokumentálás!

Absztrakció

- Digitális tervezés „elvi-fogalmi” szintje
  - Kezdeti absztrakció a tervezés során meghatározó, kritikus pont!
    - 1. koncepcionális modell (elvi elgondolás)
    - 2. megvalósítható, realizálható modell (HW)
  - Magas-szintű absztrakció ⇒ elvi modell szintenkénti finomítása és felépítése

Formalizmus

- A rendszer viselkedésének leírására szolgál
  - Szisztematikus szabályok és eljárások
  - Minden absztrakciós szinten fontos a használatuk
    - Pl: alapvető formalizmus a Boole-algebra (bináris logika elmélete) – de csak alsóbb szinteken használható

(felsőbb-, rendszer-szint) (alsóbb-, áramköri szint)

Absztrakció ⇔ Boole algebra (konkretizálás)

Digitális tervezés

- Logikai konstansok:
  - Logikai állítás: Igaz / Hamis, True / False, 1 / 0
- Logikai (bináris) változók:
  - Pl: ‘A’ logikai 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 igazság táblázattal (Truth Table)

Igazságtábla: logikai operátorok felírása

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

A	C	B	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

⇔

A	C	B	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: $\mathrm{NOT}\ A = \overline{A}$
- Formális definíció igazságtáblával:

A	NOT A
0	1
1	0

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

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

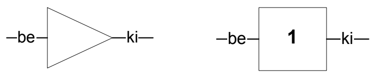
- Def: B**+**C értéke pontosan akkor ‘igaz’ ,ha ‘B’ és ‘C’ közül legalább az egyik ‘igaz’, különben hamis

Egy ill. kétváltozós logikai függvények bemutatása és szabványos jelöléseik

Egyváltozós logikai függvények:

- Jelmásoló (jel-erősítés):

be	ki
0	0
1	1

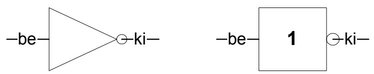


Nemzetközi szabvány

Magyar szabvány

- Negálás - Inverter (NOT):

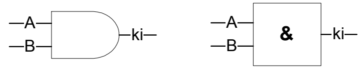
be	ki
0	1
1	0



Kétváltozós logikai függvények:

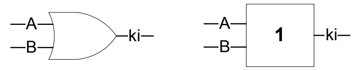
- ÉS (AND):

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



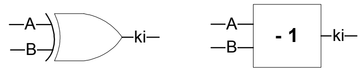
- VAGY (OR):

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



- Antivalencia (XOR):

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



Kétváltozós log.függv. (folyt.):

- NEM-ÉS (NAND):

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

- NEM-VAGY (NOR):

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

- Ekvivalencia (NXOR):

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

Tri-State Buffer:

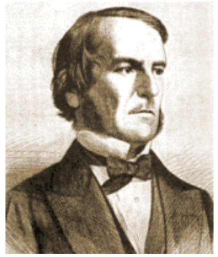
- buszok esetén használatos: kommunikációs irány változhat
  - Driver: egyirányú kommunikációra
  - Transceiver: kétirányú kommunikációra
- 3 állapota lehet:
  - magas: ‘1’
  - alacsony: ‘0’ (normál TTL szintek)
  - nagy impedanciás áll: ‘Z’ – mindkét kimeneti tranzisztor zár

High-true enable

Low-true enable

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 · B=B · A
  - Asszociatív: A+(B+C)=(A+B)+C=A+B+C
    A ·(B · C)=(A · B) · C=A ·B ·C
  - Disztributív: A·(B+C)=A · B+A · C,
    A+(B · C)=(A+B)·(A+C)
- Operátor precedencia (csökkenő sorrendben):
  - NOT
  - AND
  - OR
  - átzárójelezhetőség!

Boole algebrai azonosságok!

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

2.) $A+0=A$

3.) $A+1=1$

4.) $A+A=A$

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

6.) $A\cdot 1=A$

7.) $A\cdot 0=0$

8.) $A\cdot A=A$

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

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

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

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

16.) $A+B\cdot C=(A+B)\cdot(A+C)$

17.) $A\cdot(B+C)=A\cdot B+A\cdot C$

De-Morgan azonosságok:

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

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

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

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

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:

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

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

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

- Szorzat (AND) termek összeg (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.
  - 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: $Y = \overline{A}\cdot\overline{B} + A\cdot\overline{B} + A\cdot B = m_0 + m_2 + m_3$

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

- Komplemens: $\overline{Y} = \overline{A}\cdot B = m_1$

  [0 1]

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

- összeg (OR) termek szorzat (AND) kapcsolata
- **Maxterm** (**M$_i$**): olyan kanonikus összeg term, amelyben mindegyik logikai változó pontosan egyszer fordul elő, ponált, vagy negált alakban.
  - Valós maxterm: $\overline{A} + B + \overline{C}$, de nem valós: $\overline{A} + \overline{C}$
  - 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 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. → 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. → Így M_0 (táblázat 0.sora)

Példa: KNF felírása

- Igazságtábla

sor	J	K	L	W
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:

$$W(KNF) = (J+K+L)\cdot(\overline{J}+K+L)\cdot(\overline{J}+K+\overline{L})\cdot(\overline{J}+\overline{K}+L)\cdot(\overline{J}+\overline{K}+\overline{L})$$

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

$$W(DNF) = (\overline{J}\cdot\overline{K}\cdot L) + (\overline{J}\cdot K\cdot\overline{L}) + (\overline{J}\cdot K\cdot L)$$

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

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) ad,
  - Egy sorhoz több term is tartozhat!

Példa: DNF -> Igazságtábla

- Eredeti egyenlet:

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

kanonikus (minterm)

- Kapott igazságtábla:

sor	J	K	L	W	
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) ad,
  - Egy sorhoz több term is tartozhat!

Példa: KNF -> Igazságtábla

- Eredeti egyenlet:

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

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: $Y = J\cdot\overline{K} + \overline{J}\cdot K\cdot L + J\cdot K\cdot\overline{L} + K\cdot L$

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)

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

- Bizonyos bemeneti kombinációkra ugyanazt a kimeneti eredményt kapjuk (irreleváns)
- Jele: „–” Don’t care kimeneti állapot
- Pl.

A	B	Y
0	X	1
1	0	-
1	1	0

ha ‘–’=1, $Y = \overline{A} + A\cdot\overline{B} = \overline{A} + \overline{B}$

ha ‘–’=0, $Y = \overline{A}$

KARNOUGH TÁBLÁK

Karnough 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

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

Karnough tábla felírása igazság táblázatból

- Igazságtábla mindenegyes sorának kimeneti értékéhez (Y_i) a Karnough tábla egy négyzete 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

B \ A	0	1
0	Y_0	Y_2
1	Y_1	Y_3

könyvbeli jelölés

A \ B	0	1
0	Y_0	Y_1
1	Y_2	Y_3

általános jelölés

Karnough táblák

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

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

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

Karnough táblák

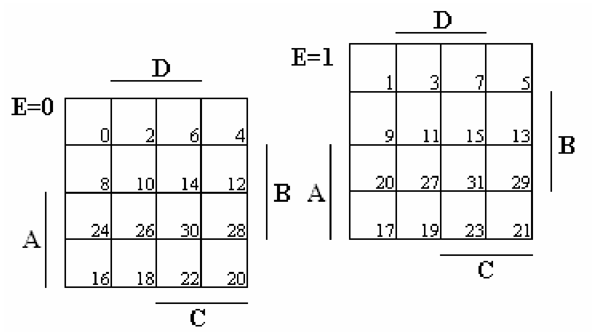
- Pl: n=4 változó esetén lehetséges táblákra:

CD \ AB	00	01	11	10
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}

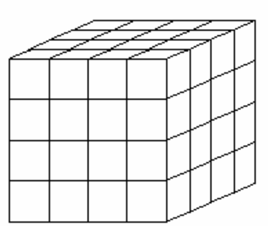
AB \ CD	00	01	11	10
00	Y_0	Y_1	Y_3	Y_2
01	Y_4	Y_5	Y_7	Y_6
11	Y_{12}	Y_{13}	Y_{15}	Y_{14}
10	Y_8	Y_9	Y_{11}	Y_{10}

Karnough táblák

- n= 5 változó esetén



- n=6 változó esetén



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

- Karnough tábla:

A \ B	0	1
0	1	0
1	1	0

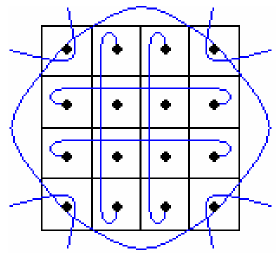
Szomszédosság – adjacencia

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

A \ BC	00	01	11	10
0	0 (0)	0 (1)	1 (3)	0 (2)
1	0 (4)	0 (5)	1 (7)	0 (6)

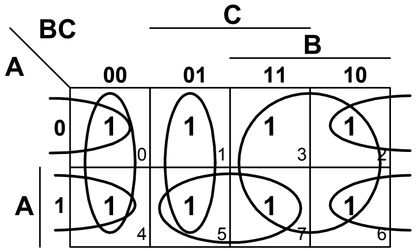
Egyszerűsítés Karnough táblákkal

- 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 lefedésnek megfelelően kell megválasztani (NTSH)
  - Egymás mellett lévő (adjacens) sorokra és oszlopokra érvényes:

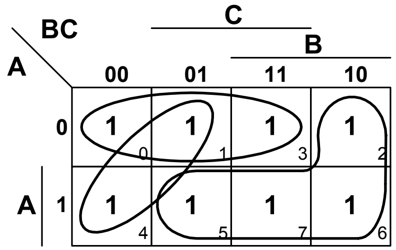


Példa: Karnough táblák egyszerűsítése

- érvényes   érvénytelen



Nem összes, de lehetséges egyszerűsítések - érvényes



Átlós, és nem 2^{n} számú ‘1’-es lefedés érvénytelen

Lehetséges módszerek Karnough tábla értelmezésére:

- M1: $Y(DNF)$ ‘1’-esek lefedésével képzett (<u>normál</u>, 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’-esek lefedésével képzett inverz függvény felírás

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