DIGITÁLIS TECHNIKA I

Dr. Lovassy Rita
Dr. Pődör Bálint

Óbudai Egyetem KVK
Mikroelektronikai és Technológia Intézet

4. ELŐADÁS

1

4. ELŐADÁS

1. Logikai függvények kanonikus algebrai alakjai, diszjunktív és konjunktív normálalakok
2. Logikai függvények egyszerűsítése
3. Logikai függvények grafikus ábrázolása
4. Logikai függvények minimalizálási módszerei.

Irodalom a 3. és 4. előadáshoz

Arató könyve	29-60 oldalak
Rőmer könyve	17-31 oldalak
Zsom könyve (I)	70-95, 101-129 oldalak

Az előadások ezen könyvek megfelelő fejezetein alapulnak.

2

A KÉTVÁLTOZÓS FÜGGVÉNYEK ILLETVE BOOLE MŰVELETEK ÖSSZEFOGLALÁSA

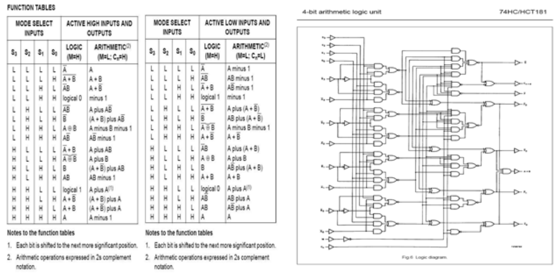
A 16 lehetséges, kétargumentumos Boole művelet illetve kétváltozós függvény közül
6 triviálisnak tekinthető, amelyek közül
kettő állandó (logikai konstans),
négy pedig egyváltozós.
Az utóbbiak közül kettő a negáció.

A 10 nem triviális függvény közül elsősorban kettőt (ÉS, VAGY), illetve negáltjaikat (NEM-ÉS,NAND és NEM-VAGY,NOR), valamint az ANTIVALENCIA (XOR) és EKVIVALENCIA (XNOR) függvényeket tárgyaltuk.

3

IC Implementation: Philips 74HC/HCT181

Total 16 arithmetic operations (add, subtract, plus, shift, plus 12 others)
- Total 16 logic operations (XOR, AND, NAND, NOR, OR, plus 11 others)
- Capable of active-high and active-low operation



ARITMETIKAI LOGIKAI EGYSÉG

FUNCTION TABLES

MODE SELECT INPUTS				ACTIVE HIGH INPUTS AND OUTPUTS	
S_3	S_2	S_1	S_0	LOGIC (M=H)	ARITHMETIC[2] (M=L; C_n=H)
L	L	L	L	$\overline{A}$	A
L	L	L	H	$\overline{A + B}$	A + B
L	L	H	L	$\overline{A}B$	$A + \overline{B}$
L	L	H	H	logical 0	minus 1
L	H	L	L	$\overline{AB}$	A plus $A\overline{B}$
L	H	L	H	$\overline{B}$	(A + B) plus $A\overline{B}$
L	H	H	L	$A \oplus B$	A minus B minus 1
L	H	H	H	$A\overline{B}$	$A\overline{B}$ minus 1
H	L	L	L	$\overline{A} + B$	A plus AB
H	L	L	H	$\overline{A \oplus B}$	A plus B
H	L	H	L	B	$(A + \overline{B})$ plus AB
H	L	H	H	AB	AB minus 1
H	H	L	L	logical 1	A plus A[1]
H	H	L	H	$A + \overline{B}$	(A + B) plus A
H	H	H	L	A + B	$(A + \overline{B})$ plus A
H	H	H	H	A	A minus 1

Notes to the function tables
1. Each bit is shifted to the next more significant position.
2. Arithmetic operations expressed in 2s complement notation.

MODE SELECT INPUTS				ACTIVE LOW INPUTS AND OUTPUTS	
S_3	S_2	S_1	S_0	LOGIC (M=H)	ARITHMETIC[2] (M=L; C_n=L)
L	L	L	L	$\overline{A}$	A minus 1
L	L	L	H	$\overline{AB}$	AB minus 1
L	L	H	L	$\overline{A} + B$	$A\overline{B}$ minus 1
L	L	H	H	logical 1	minus 1
L	H	L	L	$\overline{A + B}$	A plus $(A + \overline{B})$
L	H	L	H	$\overline{B}$	AB plus $(A + \overline{B})$
L	H	H	L	$\overline{A \oplus B}$	A minus B minus 1
L	H	H	H	$A + \overline{B}$	$A + \overline{B}$
H	L	L	L	$\overline{A}B$	A plus (A + B)
H	L	L	H	$A \oplus B$	A plus B
H	L	H	L	B	$A\overline{B}$ plus (A + B)
H	L	H	H	A + B	A + B
H	H	L	L	logical 0	A plus A[1]
H	H	L	H	$A\overline{B}$	AB plus A
H	H	H	L	AB	$A\overline{B}$ plus A
H	H	H	H	A	A

Notes to the function tables
1. Each bit is shifted to the next more significant position.
2. Arithmetic operations expressed in 2s complement notation.

ARITMETIKAI LOGIKAI EGYSÉG (ARITHMETIC LOGIC UNIT, ALU) FŐBB JELLEMZŐI

Tipikus jellemzők (pl. 74181 típus):

4-bit szóhosszúság (be: A, B, ki: F)
16 féle aritmetikai és 16 féle logikai művelet
4 kiválasztó bemenet, 1 módváltó bemenet
Átvitelbemenet és kimenet
Reláció (A = B) kimenet
Gyorsátviteli bitek (G, P) kimenet

6

LOGIKAI FÜGGVÉNYEK MEGADÁSI MÓDJAI

A logikai függvény többféle módon megadható, illetve ábrázolható. Az alábbi módokat fogjuk alkalmazni.

1. Igazságtáblázat
2. Algebrai alak
3. Grafikus ábrázolás
4. Szimbolikus jelölés

LOGIKAI FÜGGVÉNY MEGADÁSA IGAZSÁGTÁBLÁZATTAL ILLETVE ALGEBRAI ALAKBAN

i	A	B	C	Y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

$$Y(A,B,C) = \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + AB\overline{C}$$

Egy logikai függvény egyértelmű megadása: a független változók összes lehetséges kombinációjához megadjuk a függvénykapcsolat által előírt függvényértéket. Ez az ún. igazságtáblázat.
Az ilyen függvénydefiníció egyértelmű.

FÜGGVÉNY MEGADÁSA ALGEBRAI ALAKBAN: KANONIKUS ALAK

A logikai függvény megadható oly módon, hogy a logikai műveletek szimbólumaival (ÉS, VAGY és NEM) definiáljuk a logikai függvényt.

A kombinációs hálózatok tervezésének kiindulási lépése a logikai függvény felírása a megoldandó feladat alapján. Általában az algebrai alak használatos.

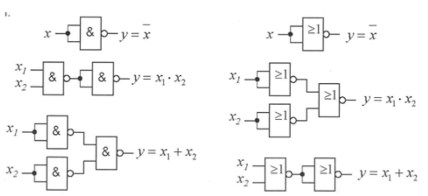
Egyazon függvény többféle algebrai alakban is megadható. Közöttük kitüntetett szerepük van az ún. normalizált vagy kanonikus alakoknak.

GRAFIKUS ÁBRÁZOLÁS

A logikai függvény értékei pl. az igazságtáblázat segítségével grafikusan is szemléltethetők ún. függvénytáblázatokon (*Karnaugh*-táblázatok, ill. *Veitch*-diagramok). Ez az ábrázolási mód korlátozott független változó szám esetén igen szemléletes, és az ún. függvény-minimalizáláskor jól felhasználható.

SZIMBÓLIKUS JELÖLÉS

Az egyes logikai műveleteket szimbólumok jelölhetik (pl. a rajzjelek). Elektronikus logikai rendszerekben minden szimbólum egy-egy, a logikai műveletet realizáló áramkörre utal. Így a logikai függvények szimbolikus megadása egyúttal az áramköri megvalósításról is információt nyújt.



LOGIKAI FÜGGVÉNYEK KANONIKUS ALAKJAI

A most következő anyagot egy háromváltozós, teljesen határozott logikai függvény igazságtáblázata fogja illusztrálni, az új fogalmakat ennek alapján vezetjük be, és az elméletet is ez alapján építjük fel.

A tárgyalásmód Arató Péter (ajánlott) könyvét (Logikai rendszerek tervezése) követi.

TELJESEN HATÁROZOTT LOGIKAI FÜGGVÉNY (LOGIKAI FELADAT)

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Teljesen határozott logikai függvény megadható azon változókombinációk felsorolásával,amelyekhez F = 1, vagy azokéval amihez F = 0 tartozik.

$F(ABC) = \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\,\overline{C} + AB\overline{C}$

vagy

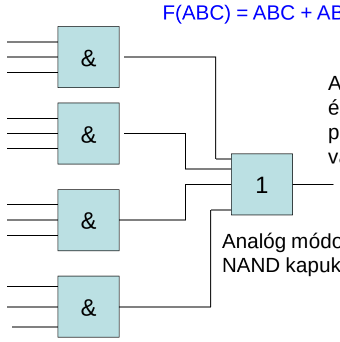
$\overline{F(ABC)} = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + A\overline{B}C + ABC$

F és $\overline{F}$ egyaránt logikai szorzatok VAGY kapcsolataként adható meg.

13

KÉTSZINTŰ ÉS-VAGY KAPUS REALIZÁLÁS

$F(ABC) = \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\,\overline{C} + AB\overline{C}$



Az egyes bementekre értelemszerűen az egyes ponált vagy negált változók kerülnek.

Analóg módon alakítható ki a csak NAND kapukból álló hálózat is.

14

NEM TELJESEN HATÁROZOTT LOGIKAI FÜGGVÉNY (FELADAT)

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	-
1	0	1	0
1	1	0	-
1	1	1	1

Nem teljesen határozott logikai függvény, van olyan változókombináció, amelyekhez nincs előírt függvényérték.

A táblázat négy olyan egymástól különböző de teljesen határozott logikai függvényt definiál, melyekkel megvalósított kombinációs hálózatok mindegyike megoldja az adott logikai feladatot. A két közömbös F érték négyféleképpen tehető határozottá. A tervezés során kell kiválasztani a négy hálózat közül a legkedvezőbbet.

15

NEM TELJESEN HATÁROZOTT LOGIKAI FÜGGVÉNY

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	-
1	0	1	0
1	1	0	-
1	1	1	1

A nem teljesen határozott logikai függvények algebrai alakjában a közömbös kombinációknak megfelelő tagokat általában zárójelben írják fel.

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

$+ (A\overline{B}\,\overline{C}) + (AB\overline{C})$

16

LOGIKAI FÜGGVÉNYEK KANONIKUS ALAKJAI

A kombinációs hálózatok tervezésénél célszerű az algebrai alakból kiindulni. Mivel egy logikai függvénynek több algebrai alakja is van, olyan speciális tulajdonságú alakot kell keresni, mely olyan, hogy a logikai függvényhez más, ilyen tulajdonságú algebrai alak nem rendelhető. Az ilyen alak a logikai függvény normál vagy kanonikus alakja.

17

FÜGGVÉNYEK KANONIKUS ALAKJAI

A kombinációs hálózatok tervezésénél célszerű az algebrai alakból kiindulni. Mivel egy logikai függvénynek több algebrai alakja is lehet, a tervezés kiindulási alapja célszerűen a logikai függvény valamelyik kanonikus vagy normál alakja.

A diszjunktív kanonikus alak (extended sum-of-product, SOP) konjunktív tagok azaz mintermek összegéből áll.

A konjunktív kanonikus alak (extended product-of-sum, POS) diszjunktív tényezők azaz maxtermek szorzatából áll.

18

DISZJUNKTÍV KANONIKUS ALAK (EXTENDED SUM-OF-PRODUCTS)

A logikai szorzatok logikai összegeként (ÉS-VAGY) képzett algebrai alak kanonikus vagy normál alak.
A példa szerinti függvényalak tulajdonságai:

$$F(ABC) = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C}$$

- mindegyik szorzat egy olyan független-változó kombináció, melyhez F = 1 tartozik;
- mindegyik szorzatban az összes független változó szerepel ponált vagy negált alakban.

A teljesen határozott függvénynek csak egy ilyen algebrai alakja van a diszjunktív kanonikus alak.

19

DISZJUNKTÍV KANONIKUS ALAK

A függvény értékei legyenek x_i (az x_i lehetséges értékei 0 vagy 1.
Egy n-változós logikai függvény diszjunktív kanonikus alakja (a mintermek m_i^n, és $k = 2^n-1$)

$$F(A,B,C...) = x_0 m_0^n + x_1 m_1^n + ... + x_k m_k^n = \sum_{i=0}^{k} x_i m_i^n$$

A diszjunktív kanonikus alaknál azokat a tagokat kell figyelembe venni, ahol a függvény értéke 1.

20

MINTERM (DEFINICIÓ)

A diszjunktív kanonikus alak tagjai neve minterm

m_i^n itt n a független változók száma,

i a független változók adott kombinációjának megfelelő bináris szám decimális értéke.

$$F(ABC) = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C}$$

(*emlékeztető*: (2) (3) (4) (6))

$$F(ABC) = m_2^3 + m_3^3 + m_4^3 + m_6^3$$

más jelölés:

$$F = \sum^{3} (2,3,4,6)$$

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

21

KONJUNKTÍV KANONIKUS ALAK

A negált függvény

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

$$\overline{F(ABC)} = m_0^3 + m_1^3 + m_5^3 + m_7^3$$

A függvény negáltja azokat a mintermeket tartalmazza, melyeket a függvény nem tartalmaz (ez csak a teljesen határozott logikai függvényekre igaz!)

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

22

KONJUNKTÍV KANONIKUS ALAK (2)

A De Morgan képletek felhasználásával elvégzett átalakításokkal előállítható az F függvény konjunktív kanonikus alakja, mely az un. maxterm-ek szorzataként adható meg

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

$$(A + B + C)\,(A + B + \bar{C})\,(\bar{A} + B + \bar{C})\,(\bar{A} + \bar{B} + \bar{C})$$

$$F(ABC) = M_7^3 M_6^3 M_2^3 M_0^3$$

23

MINTERMEK ÉS MAXTERMEK KAPCSOLATA

Eredeti függvény, diszjunktív normálalak

$$F(ABC) = m_2^3 + m_3^3 + m_4^3 + m_6^3$$

Negált függvény, diszjunktív normálalak (index i)

$$\overline{F(ABC)} = m_0^3 + m_1^3 + m_5^3 + m_7^3$$

Eredeti függvény, konjunktív normálalak (index I)

$$F(ABC) = M_7^3 M_6^3 M_2^3 M_0^3 = \prod^{3} (0,2,6,7)$$

24

MINTERMEK ÉS MAXTERMEK

A negált függvény mintermjeinek i indexe és a ponált függvény maxtermjeinek l indexei közötti összefüggés (esetünkben három változós a függvény)

$$i + l = 7 = 2^3 - 1$$

n-változós függvényre

$$i + l = 2^n - 1$$

MINTERMEK ÉS MAXTERMEK KAPCSOLATA

Minden minterm egy maxterm inverze, és fordítva, minden maxterm egy minterm inverze. A $k = 2^n-1$ jelöléssel

$$m_i^n = \overline{M_{k-i}^n}$$

és

$$M_i^n = \overline{m_{k-i}^n}$$

A mintermek és maxtermek indexei, i és 2^n-1-i egymás komplemensei. Bináris alakjukban az 1 és 0 számjegyek fel vannak cserélve. Összegük páronként 2^n-1, mely binárisan csupa 1-est tartalmaz.

MINTERMEK ÉS MAXTERMEK

Az n-változós függvények összes mintermjeinek összege a logikai 1 értéket, összes maxtermjeinek szorzata pedig a logikai 0 értéket adja adja

$$\sum_{i=0}^{k} m_i^n = 1 \quad \text{és} \quad \prod_{i=0}^{k} M_{k-i}^n = 0$$

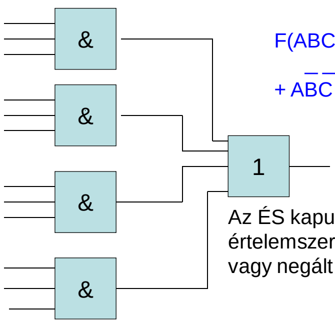
($k = 2^n-1$)

ELVI LOGIKAI RAJZ

Minden logikai függvény megadható ÉS, VAGY, NEM műveletekkel. Eltekintve a bemeneti változók negáltját adó NEM kapuktól (INVERTER), a két kanonikus alak kétszintű ÉS-VAGY illetve VAGY-ÉS kapuhálózattal realizálható.

Mivel az ÉS, VAGY és NEM műveletek akár kizárólag NAND akár kizárólag NOR kapukkal megvalósíthatók, ezért a diszjunktív normálalakból kiindulva bármely logikai függvény realizálható kétszintű NAND kapus vagy NOR kapus hálózattal.

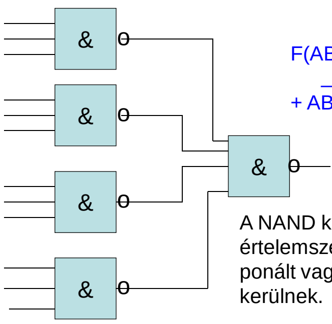
AZ F(A,B,C) KÉTSZINTŰ ÉS-VAGY KAPUS REALIZÁLÁSA



$$F(ABC) = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C}$$

Az ÉS kapuk bementeire értelemszerűen az egyes ponált vagy negált változók kerülnek.

AZ F(A,B,C) KÉTSZINTŰ NAND KAPUS REALIZÁLÁSA

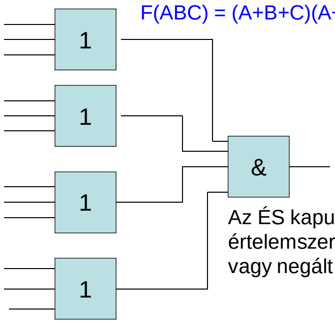


$$F(ABC) = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C}$$

A NAND kapuk bemeneteire értelemszerűen az egyes ponált vagy negált változók kerülnek.

AZ F(A,B,C) KÉTSZINTŰ VAGY-ÉS KAPUS REALIZÁLÁSA

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

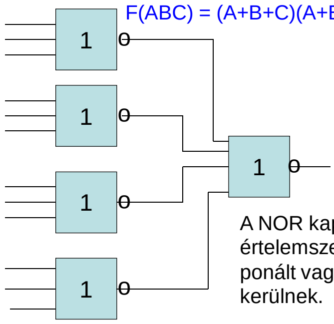


Az ÉS kapuk bemeneteire értelemszerűen az egyes ponált vagy negált változók kerülnek.

31

AZ F(A,B,C) KÉTSZINTŰ NOR KAPUS REALIZÁLÁSA

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



A NOR kapuk bemeneteire értelemszerűen az egyes ponált vagy negált változók kerülnek.

32

Kombinációs hálózatok kétszintű (pl. ÉS-VAGY) megvalósítása

A (diszjunktív) kanonikus alak közvetlenül ilyen kétszintű megoldást ad (ÉS kapukkal realizált mintermek összegét, vagyis VAGY kapcsolatát).

A minimalizálás összevonásai egyszerűbb, de ugyancsak kétszintű ÉS–VAGY hálózatra vezetek.

A működési sebesség ma kritikus kérdés, ezért a szintek számát lehetőleg nem szabad növelni! Minden kapu-áramkörnek véges működési ideje (propagation delay) van, és sorba kötött kapuknál ezek összeadódnak.

További fontos kérdés a dinamikus viselkedés, az ún. hazárdok, és azok kiküszöbölése.

33

LOGIKAI FÜGGVÉNY EGYSZERŰSÍTÉSE: MINIMALIZÁLÁS

Logikai hálózat tervének gazdaságossága:

1. Felhasznált kapuk számának csökkentése;
2. Összekötetések számának csökkentése;
3. Kapukat megvalósító építőelem-fajták optimális kiválasztása.

A 3. pontbeli feladat megoldására nincs általános módszer, az 1. és 2. pontbeli feladatokra adhatók használható eljárások.

Az optimalizálás (minimalizálás) bizonyos korlátokkal, mint a kapubementek számának csökkentése (minimalizálása) is megfogalmazható.

34

LOGIKAI FÜGGVÉNY EGYSZERŰSÍTÉSE: ILLUSZTRÁCIÓ

$$F(ABC) = \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\overline{C} + AB\overline{C}$$

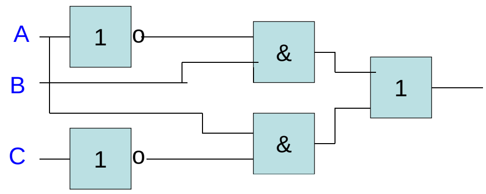
”kiemelések” alkalmazásával

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

Az eredmény egy jóval egyszerűbb algebrai alak melyet realizálva mind a kapuk, mind az összekötetések száma jelentősen csökken.

35

EGYSZERŰSÍTETT FÜGGVÉNY ÉS-VAGY KAPUS REALIZÁLÁSA



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

36

LOGIKAI FÜGGVÉNYEK MINIMALIZÁLÁSA

$F(ABC) = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C}$

mintermek $m_2^3 + m_3^3 + m_4^3 + m_6^3$

$= \bar{A}B(\bar{C} + C) + A\bar{C}(\bar{B} + B) = \bar{A}B + A\bar{C}$

Az összevonható mintermek egy helyértékben és csakis egyben térnek el egymástól:

(010) és (011), illetve (100) és (110)

Ez az összevonások és a minimalizálás kulcsa!

37

SZOMSZÉDOS MINTERMEK, MINIMALIZÁLÁS

Szomszédos mintermek: egy logikai változó ponált illetve negált, a többi azonos.

A minimalizálás menete:

1. A szomszédos mintermeket összevonják, a megfelelő változó kiesik.
2. Az új alakban az esetleges szomszédos termeket megint összevonják.
3. Az eljárást addig folytatják míg olyan szorzatok összegét kapjuk, melyekből már egy változó sem hagyható el.

Az így kapott szorzatok, termek a prímimplikáns-ok.

38

MINIMALIZÁLÁS, PRÍMIMPLIKÁNSOK

A logikai függvény minimalizált (diszjunktív) alakja tehát prímimplikáns-ok összege.

$F(ABC) = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C}$

$F(ABC) = \bar{A}B + A\bar{C}$

itt a prímimplikánsok $\bar{A}B$ és $A\bar{C}$

A logikai függvényt egyszerűsítő eljárások célja a prímimplikánsok megkeresése. Egy függvénynek több ekvivalens legegyszerűbb alakja is lehet!

39

TOVÁBBI ILLUSZTRÁCIÓ

$F = m_1^3 + m_3^3 + m_5^3 + m_7^3 = F = \Sigma^3\,(1,3,5,7)$

az algebrai alak

$F(A,B,C) = \bar{A}\,\bar{B}C + \bar{A}BC + A\bar{B}C + ABC$

$= \bar{A}C\underbrace{(\bar{B} + B)}_{(1)} + AC\underbrace{(\bar{B} + B)}_{(1)}$

$= \bar{A}C + AC = C\underbrace{(\bar{A} + A)}_{(1)} = C$

40

MINIMALIZÁLÁS KONJUNKTÍV ALAKBAN

Hasonló gondolatmenet érvényes a függvények konjunktív vagy maxtermes alakjára is. Két szomszédos maxterm szintén összevonható egyetlen összeggé, mely nem tartalmazza a két maxtermet megkülönböztető változót.

$(\bar{A} + \bar{B} + \bar{C})(A + \bar{B} + \bar{C}) = (\bar{A} + (\bar{B} + \bar{C}))(A + (\bar{B} + \bar{C}))$

$= A\bar{A} + (A + \bar{A})(\bar{B} + \bar{C}) + (\bar{B} + \bar{C}) = \bar{B} + \bar{C}$

41

FÜGGVÉNYMINIMALIZÁLÁSI ELJÁRÁSOK

1. Grafikus (táblázatos) minimalizálás: Karnaugh-táblázat alapján.

2. Számjegyes minimalizálás (Quine-McCluskey-módszer).

3. ”Egzakt” algebrai, számítógépre adaptálható algoritmusok, pl. irredundás lefedési-algoritmus csoport .

4. Egyéb (többnyire heurisztikus) eljárások.

42

MINIMALIZÁLÁS

Valamikor régen a kombinációs hálózatot megvalósító kapuáramkörök bemenet-számát kellett csökkenteni az alkatrészek számának csökkentése céljából.

Ma tulajdonképpen *sebesség-növelési* és *megbízhatósági célból* kell a (logikai) hálózatot minimalizálni.

„A legkevesebb hibát az az alkatrész tudja okozni, amelyik nincs is beépítve”. (Idézet Dr. Tóth Mihály székesfehérvári főiskolai tanártól.)

KARNAUGH TÁBLA, K-MAP

A Karnaugh tábla (Karnaugh map) mint Veitch diagram is ismert (röviden K-map, vagy KV-map). Első leírója Maurice Karnaugh (Bell Labs), illetve Edward W. Veitch.

Edward W. Veitch, *A chart method for simplifying truth functions*, May 1952, Proc. Assoc. for Computing Machinery, Pittsburgh
Maurice Karnaugh, *The map method for synthesis of combinational logic circuits*, Trans. AIEE, pt. I, 72(9), 553-599, November 1953.

A nem túl nagy változószámú függvények gyors szemléletes minimalizálása mellett alkalmas a potenciális hazárd-jelenségek azonosítására és eliminálására, melyeket csupán a Boole algebrai egyenletek alapján nem lehet elvégezni.

A KARNAUGH TÁBLA (K TÁBLA)

A Karnaugh tábla lényegében az igazságtábla célszerű, táblázatos elrendezése.

A táblázat peremezése az 1-es Hamming-távolságú, ún. Gray-kódon alapul.

A Karnaugh tábla minden egyes cellája egy-egy meghatározott változó-kombinációhoz tartozó függvényértéket tartalmaz.

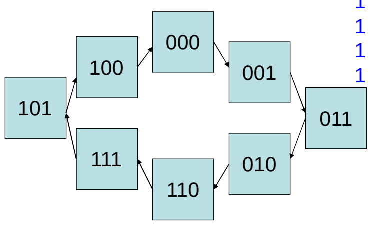
Ha a változók súlyozása rögzített, akkor a K tábla celláihoz indexek rendelhetők hozzá.

n változó esetén a cellák száma 2^n.

GRAY-KÓD, HAMMING-TÁVOLSÁG

A Gray-kód 2n számú n-bites bites kód-szavak olyan sorrendben, hogy bármelyik két szomszédos kódszó csak egyetlen bitben különbözik. Ez áll az első és utolsó kódszóra is (ciklikusság). Ha n = 3, a kódszavak sorrendje a K tábla peremén:

A	B	C
0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0



HAMMING TÁVOLSÁG

Két kódszó Hamming távolságát úgy határozzák meg, hogy a két kódszó azonos helyen álló elemeit összehasonlítják, és megállapítják hány helyen áll különböző bit. Az így kapott szám a Hamming távolság.

A Gray kód bármely két szomszédos kódszava csak egy bitben különbözik.

A	B	C
0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

BEVEZETŐ PÉLDA: 3 VÁLTOZÓS K-TÁBLA

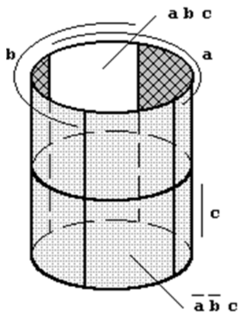
A változók súlyozása és a cellaindexek

A B	0	1	C
0 0	0	1	(0)
0 1	2	3	(2)
1 1	6	7	(6)
1 0	4	5	(4)
(4) (2)	(0)	(1)	

A szomszédos sorok és oszlopok Hamming távolsága 1.
A legalsó és a legfelső sorok is szomszédosak.
A balszélső és a jobbszélső oszlopok is szomszédosak, bár itt, a 3 változós K táblánál ez triviális.
Nem lesz triviális az oszlopok szomszédossága pl. a 4 változós tábláknál.
Az 5 és több változós K táblákon a „szomszédosság” még komplexebb.

SZOMSZÉDOSSÁG A 3-VÁLTOZÓS KARNAUGH TÁBLÁN

A B	0	1	C
0 0	0	1	(0)
0 1	2	3	(2)
1 1	6	7	(6)
1 0	4	5	(4)
(4) (2)	(0)	(1)	



49

PÉLDA: FÜGGVÉNYÁBRÁZOLÁS ÉS MINIMALIZÁLÁS K-TÁBLÁN

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

AB \ C	0	1
0 0	0	0
0 1	1	1
1 1	1	0
1 0	1	0

A jelölt szomszédos cellák (színkód jelöli) összevonhatók, így a legegyszerűbb diszjunktív alak:

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

Az összevonás után maradó két tag a két prímimplikáns.

50

A „szomszédosság" előnyei és a mintermek összevonhatósága

A szomszédos cellák csak ugyanannak az egy változónak a ponált és negált értékében különböznek.

$$X \bullet (\text{valami}) + \overline{X} \bullet (\text{valami})$$

$$= (X + \overline{X}) \bullet (\text{valami}) = 1 \bullet (\text{valami}) = (\text{valami})$$

A K tábla sorainak és oszlopainak szomszédossága nagyon könnyen felismerhetővé teszi az ilyen összevonási lehetőségeket.

51

AZ ÖSSZEVONÁS SZABÁLYAI A KARNAUGH TÁBLÁN

1. A szomszédos cellák összevonhatók, ezek a termek az egyik változót negált, illetve ponált formában is tartalmazzák.

2. Összevonhatók a diagram átellenes szélén lévő cellák is (ciklikusság).

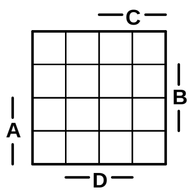
3. 2, 4, 8 ... 2^n, azaz 2 egész számú hatványainak megfelelő szomszédos (területileg összefüggő) cella összevonható.

52

NÉGYVÁLTOZÓS KARNAUGH TÁBLA

C D	0 0	0 1	1 1	1 0	
0 0	0	1	3	2	(0)
0 1	4	5	7	6	(4)
1 1	12	13	15	14	(12)
1 0	8	9	11	10	(8)
AB	(0)	(1)	(3)	(2)	

Akkor, és csak akkor ha A→8; B→4; C→2; D→1;



Akkor, és csak akkor ha A→8; B→4; C→2; D→1;

A K tábla peremezése a változók binárisérték-kombinációival vagy az oldalt elhelyezett vonalakkal adható meg.

53