Kombinációs hálózatok
Karnaugh-Veitch-diagram

Egy általános logikai függvény ábrázolása igazságtáblázattal:

a	b	c	d	..	x	f(a,b,c,d,...,x)	
0	0	0	0	0	0	1	**=> aktív**
0	0	0	0	0	1	1	
:	:	:	:	:	:	:	
1	1	1	1	1	0	0	**=> passzív**
1	1	1	1	1	1	1	

A függvény kimenetként minden bemeneti változó-variáció esetében általában két értéket vehet fel: „0” vagy „1”.
Emellett a gyakorlatban előfordul még két további különleges eset:
- a függvényérték tetszőleges lehet (”0” vagy „1”)
- a bemenet irreleváns, a függvényérték viszont meghatározott.

a	b	c	d	..	x	f(a,b,c,d,...,x)	
1	1	1	1	..	0	?	**=> tetszőleges '0' ill. '1'**
?	?	?	?	..	?	1	**=> irreleváns**

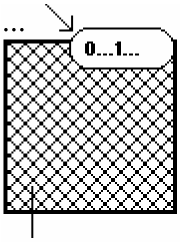
A Boole-függvények ábrázolására alkalmas fenti táblázatos módszer a gépi feldolgozásban használható. Hátránya, hogy nehezen áttekinthető és belőle az adott függvény jellegzetességei nehezen felismerhetőek.

Az ember számára áttekinthetőbb, az adott függvény jellemző tulajdonságainak jobban felismerhetővé tételét elősegítő grafikus megadási módot dolgozott ki M. Karnaugh és E. W. Veitch, mely ma **Karnaugh-Veitch-diagramként** (KV-diagram) ismert.
(Az ábrázolásmód korlátja, hogy növekvő változószámmal a komplexitása is jelentősen növekszik. 6-nál több operandusú függvények ábrázolására már más módszereket kell alkalmazni.)

Kombinációs hálózatok
Karnaugh-Veitch-diagram

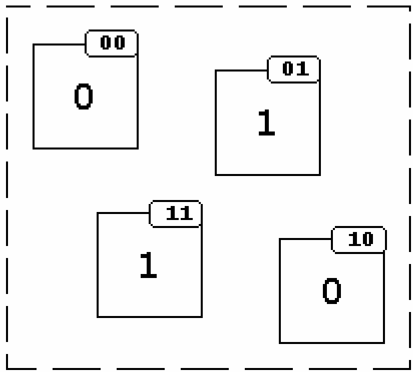
Egy függvény egyértelmű definiálásához meg kell adni a bementi változók összes lehetséges variációjához tartozó függvényértékeket.

Bemeneti változók érték-kombinációi: 0…1...



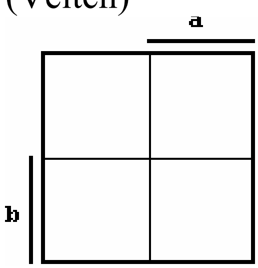
Tartalom: függvényérték 0 ill. 1

A függvény definíciója (példa)

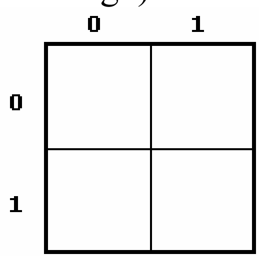


Az egyes kereteket célszerűbben, rendezetten elhelyezve egy áttekinthetőbb, egységesebb rendszert kapunk:

Halmaz-ábrázolás (Veitch)



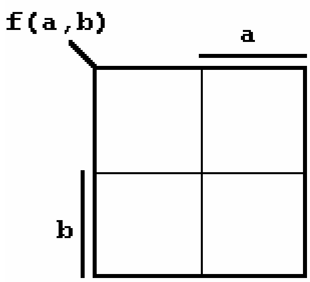
Koordináta-ábrázolás (Karnaugh)



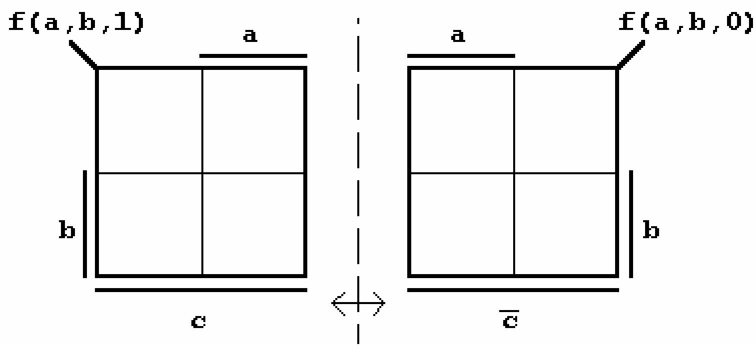
f(a,b) kétváltozós függvény grafikus ábrázolása

Fontos szabály:
A (függőleges ill. vízszintes irányban) szomszédos mezők egymástól csak egy változóértékben különbözhetnek. → szomszéd-kapcsolat

Kétváltozós függvény:

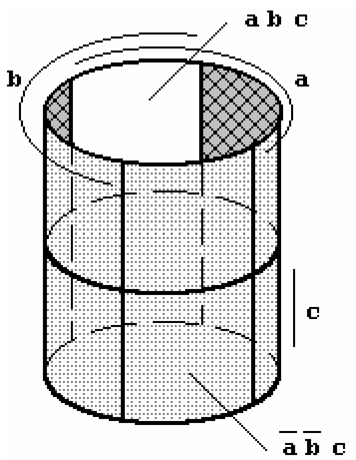


Kétváltozós függvény továbbfejlesztése háromváltozóssá duplázással:

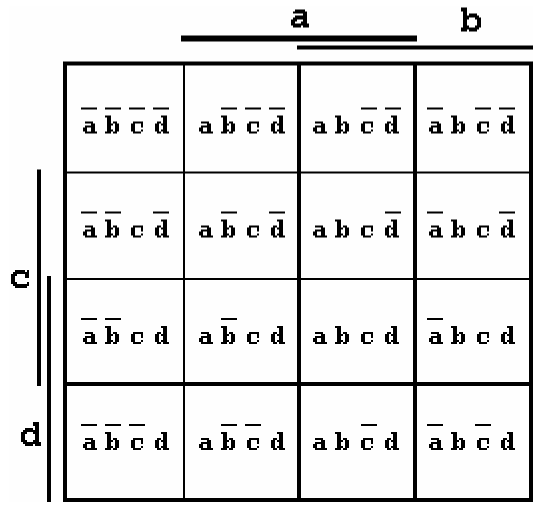


Kombinációs hálózatok
Karnaugh-Veitch-diagram

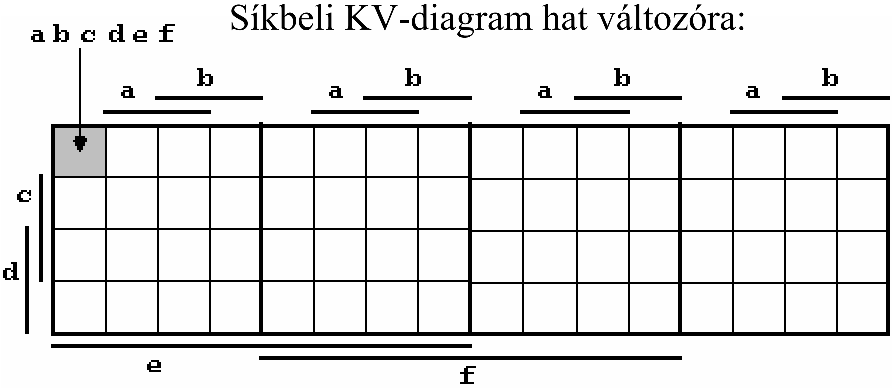
Szomszéd-kapcsolat a KV-diagram szélein:

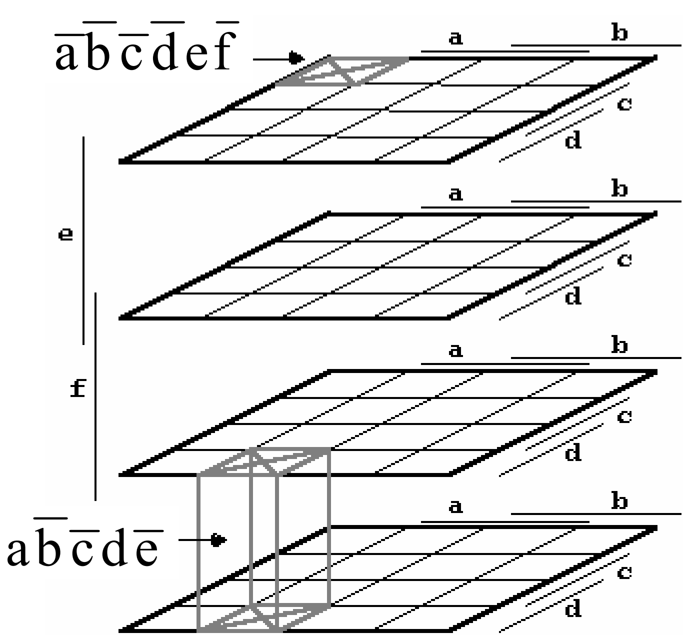


KV-diagram négy változóra:



Síkbeli KV-diagram hat változóra:

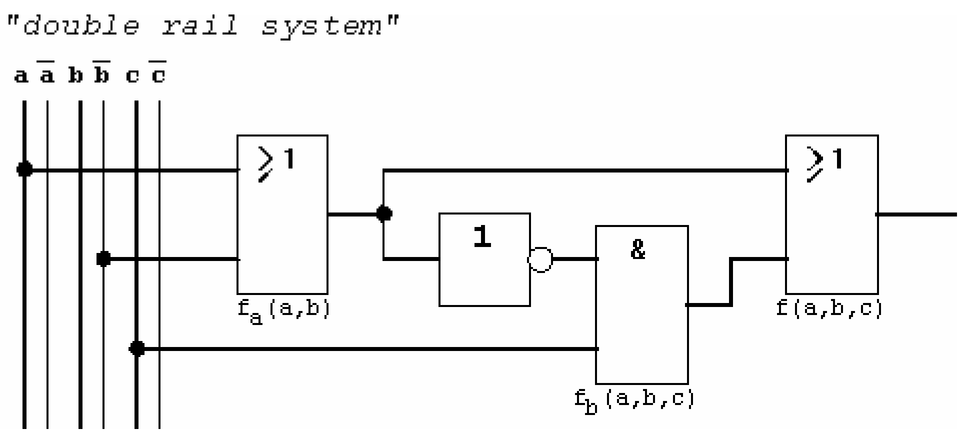


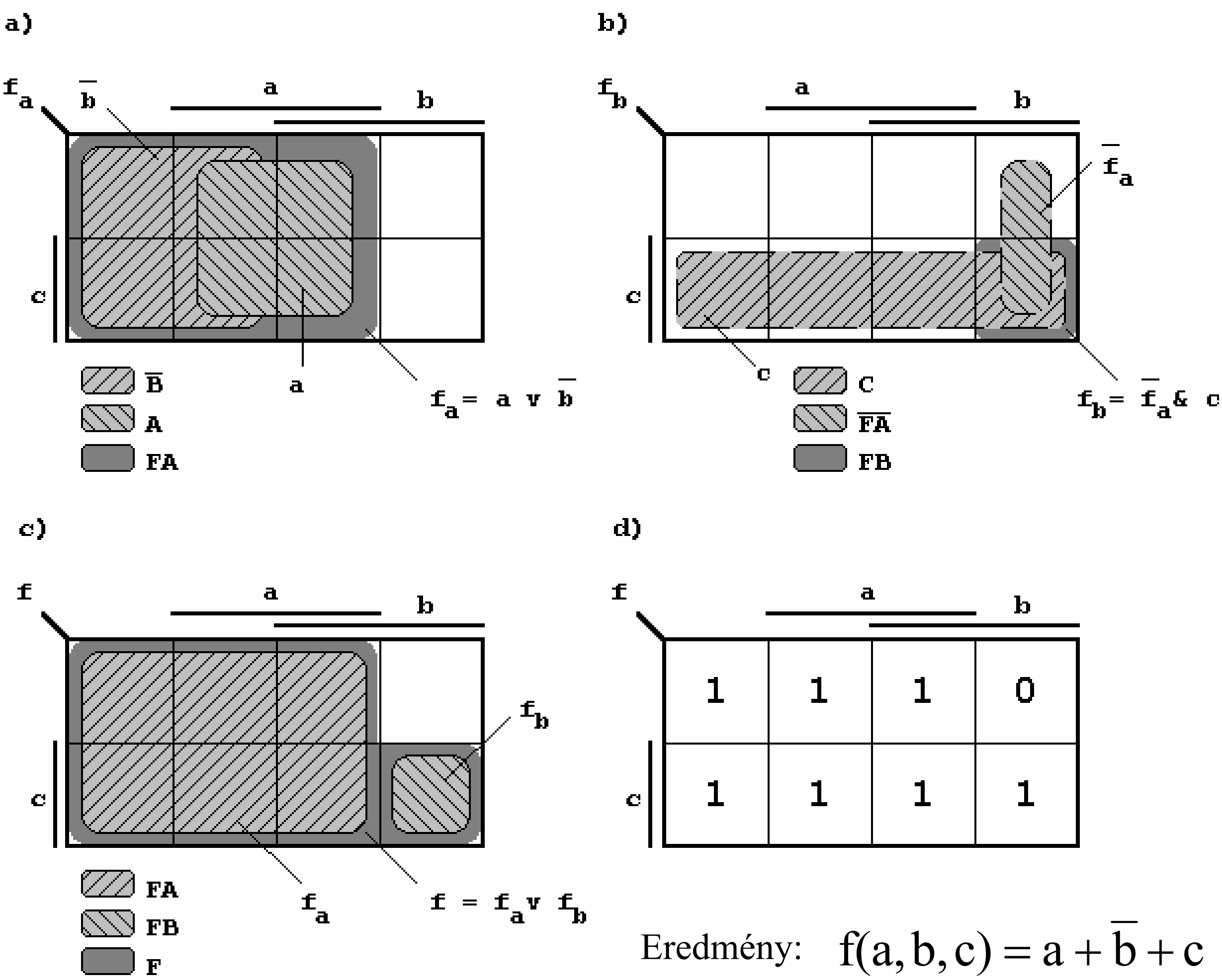


Háromdimenziós KV-elrendezés hat változóra

Példa:

f(a,b,c) Boole-függvény adott az alábbi kapcsolás révén:
Feladat: KV-diagram segítségével egyszerűsíteni





Eredmény: $f(a,b,c) = a + \overline{b} + c$

Kombinációs hálózatok
Mintermek és maxtermek

A KV-diagram jól használható komplex függvények legegyszerűbb algebrai alakjának kiolvasására.
A legegyszerűbb függvények azok, melyek a KV-diagramban csak egy „0”-t vagy „1”-et tartalmaznak. Azok a logikai függvények, melyek csak egyetlen „1”-est tartalmaznak „minimális”, melyek egyetlen „0”-t „maximális” függvényeknek tekinthetők, függetlenül a bemeneti változók számától.

Ezeket **minterm-függvényeknek** ill. **maxterm-függvényeknek** nevezzük. (röviden minterm ill. maxterm)

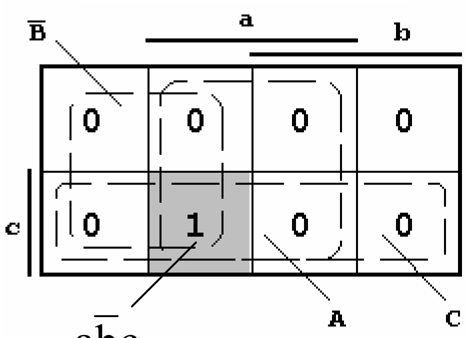
Mintermek - jelölése: $\mathbf{m_i}$, ahol i index a mintermet jellemző cella (értéke: ”1”) koordinátáit (x,…,c,b,a) jelenti kettes számrendszerbeli értelmezéssel (koordináta: változók vektora)

- algebrai leírása:a mintermet jellemző cella koordinátái ÉS műveletettel összekapcsolva

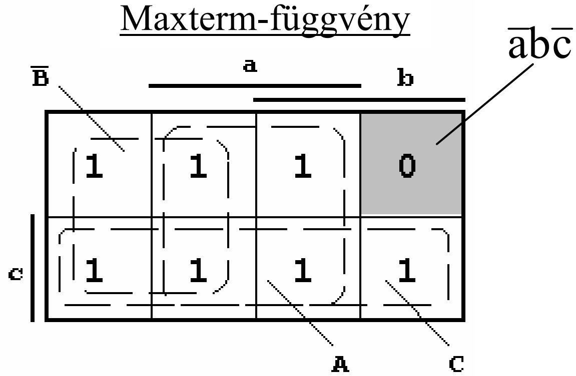
Maxtermek - jelölése: $\mathbf{M_i}$, ahol i index a maxtermet jellemző cella („0”) koordinátáit (x,…,c,b,a) jelenti kettes számrendszerbeli értelmezéssel

- algebrai leírása:a maxtermet jellemző cella negált koordinátái VAGY műveletettel összekapcsolva

Minterm-függvény



Maxterm-függvény



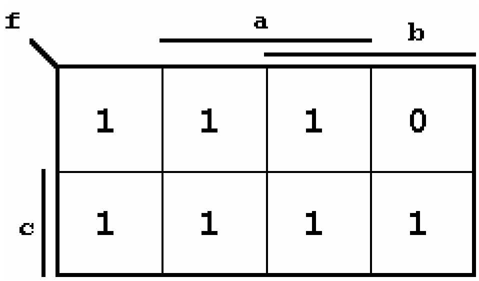
Minterm-függvény algebrai alakja: $m_5(a,b,c) = a \cdot \overline{b} \cdot c$

Maxterm- függvény algebrai alakja: $M_2(a,b,c) = a + \overline{b} + c$

Kombinációs hálózatok
Mintermek és maxtermek

Egy általános függvény felírható bizonyos számú minterm-függvény diszjunkciójaként, vagy bizonyos számú maxterm-függvény konjunkciójaként is.
Ha a KV-diagramban az „1"-esből van kevesebb, célszerű a mintermek diszjunkcióját választani, ha a „0"-kból van kevesebb, akkor pedig a maxtermek konjunkcióját.

Példa egy adott függvény minterm- ill. maxterm-ábrázolására:



7 minterm: $f(a,b,c) = \overline{a}\,\overline{b}\,\overline{c} \vee a\overline{b}\,\overline{c} \vee ab\overline{c} \vee \overline{a}\,\overline{b}c \vee a\overline{b}c \vee \overline{a}bc \vee abc$

1 maxterm: $f(a,b,c) = (a \vee \overline{b} \vee c)$

Egyszerűbben kifejezve:

$$f(a,b,c) = m_0 \vee m_1 \vee m_3 \vee m_4 \vee m_5 \vee m_6 \vee m_7$$

$$f(a,b,c) = M_2$$

Kombinációs hálózatok
Függvények normálalakjai

Definíció:

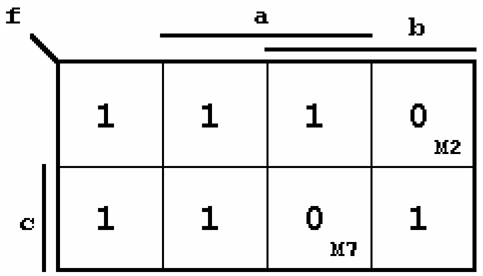
- **Kanonikus diszjunktív normálalak-**ban van a logikai függvény ábrázolva, ha az $\mathbf{m_i}$ mintermjei diszjunktív (VAGY) művelettel vannak összekötve.

- **Kanonikus konjunktív normálalak-**ban van a logikai függvény, ha az $\mathbf{M_i}$ maxtermjei konjunktív (ÉS) művelettel vannak összekötve

Kanonikus: az egyes termek az összes változót tartalmazzák, azaz teljes konjunkció ill. diszjunkció áll fenn.A függvényalak ekkor kétlépcsős kapuhálózat, és ez bár logikailag egyszerű, de felépítését tekintve nem a legoptimálisabb. Különböző egyszerűsítéseket alkalmazva elérhető, hogy az első lépcsőben már nem mindig szerepel az összes változó, így ezek a függvények már nem kanonikusak, és ekkor csak egyszerűen

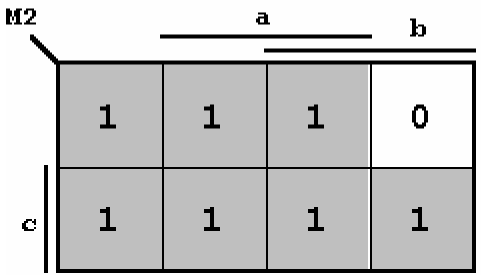
- **diszjunktív normálalak**-ban, ill.
- **konjunktív normálalak**-ban vannak.

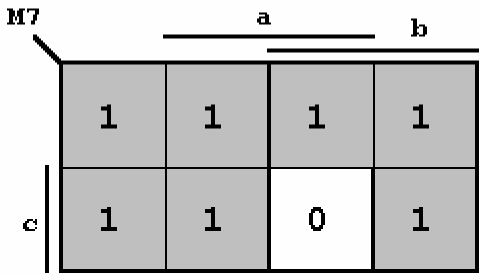
Angolul:**"*sum of products*" (*SOP*)** ill. **"*product of sums*" (*POS*)**



Példa:
KV-diagramban adott f(a,b,c) függvény kanonikus konjugált normálalakjának meghatározása:

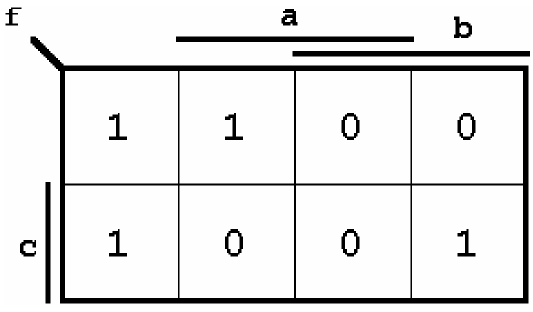
$$f(a,b,c) = (a + \overline{b} + c) \cdot (\overline{a} + \overline{b} + \overline{c})$$





a	b	c	f(a,b,c)
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Példa egy háromváltozós függvény normálalakjainak meghatározására



•Kanonikus diszjunktív normálalak: $f_D = \overline{a}\,\overline{b}\,\overline{c} + \overline{a}\,\overline{b}\,c + \overline{a}bc + a\overline{b}\,\overline{c}$

Ugyanez mintermekkel kifejezve: $f_D = m_0 + m_1 + m_4 + m_6 = \bigvee m_{0,1,4,6}$

•Kanonikus konjunktív normálalak:

$$f_K = (a + \overline{b} + c)(\overline{a} + b + \overline{c})(\overline{a} + \overline{b} + c)(\overline{a} + \overline{b} + \overline{c})$$

Ugyanez maxtermekkel kifejezve: $f_D = M_2 M_3 M_5 M_7 = \bigwedge M_{2,3,5,7}$

Összefoglalás:

adott $f(x_1,\ldots,x_n)$ függvény, melynek kanonikus normálalakjai az alábbiak:

$$f_D = \bigvee_{i=0}^{2^n-1} k_i \cdot m_i$$ ahol: $k_i = 0$ vagy 1

$$f_K = \bigwedge_{i=0}^{2^n-1} (k_i \vee \overline{m}_i) = \bigwedge_{i=0}^{2^n-1} (k_i \vee M_i)$$ ahol: $k_i = 0$ vagy 1

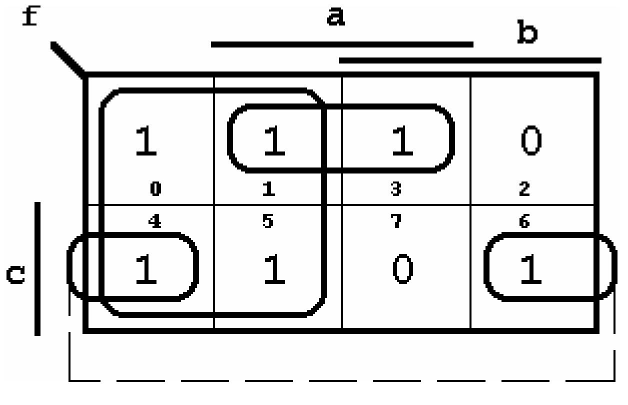
érvényes továbbá: $m_i = \overline{M}_i$ ahol m_i = minterm (teljes konjunkció), és M_i = maxterm (teljes diszjunkció)

Egy kanonikus kifejezésből akkor lesz nem kanonikus, ha legalább egy (konjunktív ill. diszjunktív) term már nem felel meg többé a minterm ill. maxterm definíciójának.

- A KV-diagramban két szomszédos mezőt csak egy vektorelem (változó), x_i különböztet meg egymástól. Amennyiben ezen szomszédos cellák függvényértéke megegyezik, akkor ettől az x_i változótól a két cella értéke nem függ és ezen szomszédos cellákat össze lehet vonni, és egy közös - a 2 cellát átfogó - vektorral leírni. Az összevont cellákat leíró vektor x_i-t már nem tartalmazza. Ez a módszer egyszerűsítésre ad lehetőséget.
- Nemcsak az egyes cellákat, hanem magukat a fenti módon összevont cellatartományokat is össze lehet vonni. (2→4, 4→8, stb.) A z összevont cellák számának minden egyes megduplázásakor kiesik egy további változó. Az összevont cellák száma 2^n lehet, ahol *n* a kiesett változók száma.

A „grafikus" összevonásnál algebrailag tulajdonképpen az alábbi összefüggést alkalmazzuk:

$$ab \vee a\overline{b} = a$$



<u>Példa cellák összevonására</u>:

4-es mező: m_0, m_1, m_4, m_5

2-ős mező: m_1, m_3

m_4, m_6

A függvény megvalósítása a fenti összevonásokkal:

$$f(a,b,c) = \overline{b} \vee a\overline{c} \vee \overline{a}c$$

A XOR-függvény bevezetésével:

$$f(a,b,c) = \overline{b} \vee (a \oplus c)$$

Ebben a példában nem lehetséges „0"-cellák összevonása, ezért a konjunktív normálalakban csak a kanonikus változat létezik:

$$f(a,b,c) = M_2 \wedge M_7$$

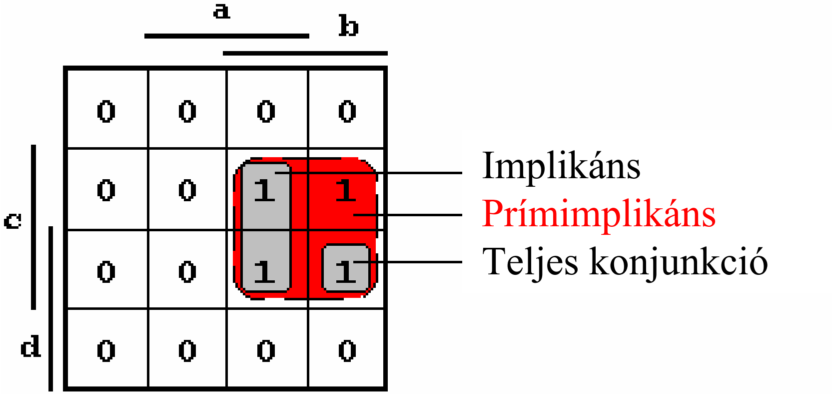
Kombinációs hálózatok
Prímimplikánsok

Az összevont cellatartományokkal jellemezhető függvény-tulajdonságok tárgyalásához a cellatartományokat az alábbi csoportokba soroljuk:

Implikánsoknak nevezzük általában a cellatartományokat. (Melyeket egyébként vagy a grafikai ábrázolásmóddal, vagy pedig az ezeknek megfelelő algebrai kifejezéssel, termekkel határozhatunk meg.)

Prímimplikánsoknak nevezzük egy logikai függvény azon implikánsait, melyek teljes egészükben semelyik másik implikánsban sem foglaltatnak benne.

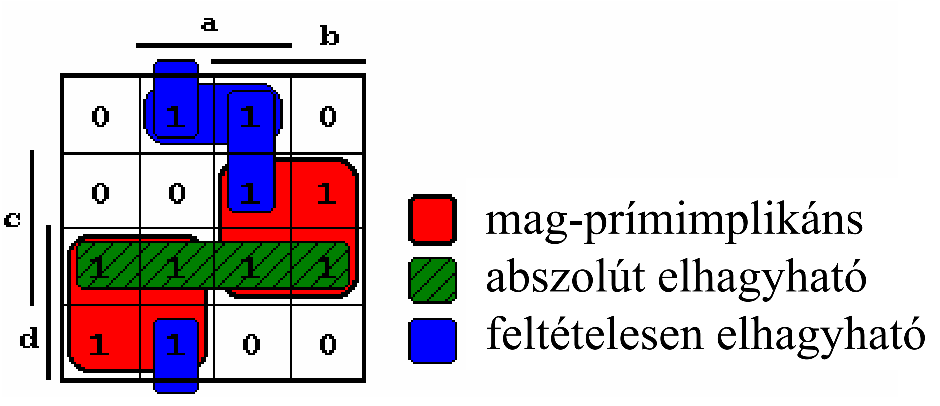
Mag- prímimplikánsoknak nevezzük azokat a prímimplikánsokat, melyek legalább egy olyan teljes konjunkciót (cellát) tartalmaznak, amely nem található meg semelyik másik prímimplikánsban sem.



- Mag-prímimplikánsoknak a kapcsolásban mindig szerepelniük kell
- A prímimplikánsok a megvalósítás során gyakran elhagyhatóak, ez viszont további tulajdonságoktól függ.
  - -Egy prímimplikáns akkor hagyható el minden további nélkül a kapcsolásból, ha az adott függvénynél létezik olyan diszjunkciója mag-prímimplikánsoknak, amely diszjunkció ezt a prímimplikánst teljes egészében tartalmazza.
  - -Azon prímimplikánsok, melyek a fentieknek nem felelnek meg, feltételesen elhagyhatóak

Kombinációs hálózatok
Prímimplikánsok

Példa: cellatartományok osztálybasorolása



A logikai függvény leírására két lehetséges minimális alak adódik:

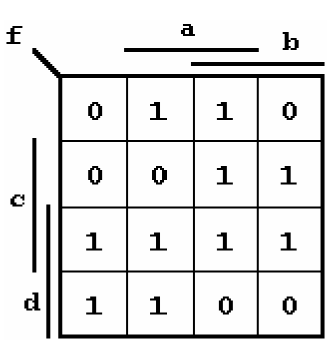
$$f(a,b,c,d) = bc \vee \bar{b}d \vee a\bar{c}\bar{d}$$

$$f(a,b,c,d) = bc \vee \bar{b}d \vee ab\bar{d} \vee a\bar{b}\bar{c}$$

Az összes „1”-es cellatartományokra megfogalmazott szabály érvényes a „0”-cellatartományokra is.

Kombinációs hálózatok
Quine-McCluskey-féle (QC) táblázatos egyszerűsítés

A Karnaugh-Veitch.módszer hátránya, hogy a grafikai ábrázolásmód miatt csak maximum 5-6 változós függvények egyszerűsíthetőek.



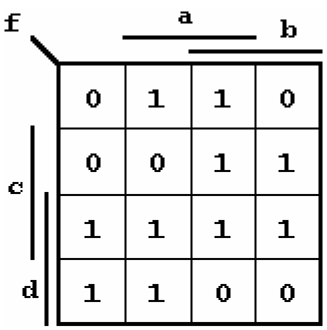
A QC módszer kiindulási alapja az egyik kanonikus normálalak.Az alábbi példában a kanonikus diszjunktív normálalakból indulunk ki.

A KV-diagramban használt példa f (d,c,b,a) négyváltozós függvényének teljes konjunkciói (mintermjei):

1000, 1100, 1110, 0110, 0011
1011, 1111, 0111, 0001, 1001

Kombinációs hálózatok
Quine-McCluskey-féle (QC) táblázatos egyszerűsítés

A Karnaugh-Veitch.módszer hátránya, hogy a grafikai ábrázolásmód miatt csak maximum 5-6 változós függvények egyszerűsíthetőek.



A QC módszer kiindulási alapja az egyik kanonikus normálalak.Az alábbi példában a kanonikus diszjunktív normálalakból indulunk ki.

A KV-diagramban használt példa f (d,c,b,a) négyváltozós függvényének teljes konjunkciói (mintermjei):

1000, 1100, 1110, 0110, 0011
1011, 1111, 0111, 0001, 1001

Kombinációs hálózatok
Quine-féle táblázatos prímcellatartomány-meghatározás

csoport	teljes konjunkció		állapot
	dcba	érték	
0	nincs ilyen		
1	0001	1	✓
	1000	8	✓
2	0011	3	✓
	0110	6	✓
	1001	9	✓
	1100	12	✓
3	0111	7	✓
	1101	13	✓
	1110	14	✓
4	1111	15	✓

Quine-eljárással az **összes** prímterm (prímimplikáns) meghatározható egy egzakt táblázatos eljárással, ahol az egyszerűsítéshez a már ismert

összefüggést használjuk.

Az állapot oszlopban azon mintermek vannak megjelölve, melyek valamelyik **szomszédos csoport** valamelyik mintermjével összevonhatók, azaz csak egy változóban különböznek.

Kombinációs hálózatok
Quine-féle táblázatos prímcellatartomány-meghatározás

csoport	dcba	decimális	állapot
1-2	00-1	1- 3	
	-001	1- 9	
	100-	8- 9	✓
	1-00	8-12	✓
2-3	0-11	3- 7	
	011-	6- 7	✓
	-110	6-14	✓
	1-01	9-13	✓
	110-	12-13	✓
	11-0	12-14	✓
3-4	-111	7-15	✓
	11-1	13-15	✓
	111-	14-15	✓

Az összevonási lehetőségek figyelembevételével újabb táblázatot készítünk, a kiesett, azaz irreleváns változók helyét kötőjellel jelölve.
Ezt megismételve újabb táblázat keletkezik (amelyben a kiesett változók száma eggyel növekszik). A folyamatot addig folytatjuk, míg további összevonás már többé nem lehetséges.
A csoportok megnevezésénél megtartjuk a legelső táblázat csoportneveit, jelölve az összevonást.

csoport	dcba	decimális	állapot
1-2,2-3	1-0-	8- 9,12-13	
	1-0-	8-12, 9-13	
2-3,3-4	-11-	6- 7,14-15	
	-11-	6-14, 7-15	
	11--	12-13,14-15	
	11--	12-14,13-15	

Végeredményként olyan táblázatot kapunk, amelyben további összevonások már nem lehetségesek.

A táblázatokban **nem megjelölt termek** lesznek az adott függvény prímimplikánsai:

$$f(a,b,c,d) = bc \vee \overline{b}d \vee cd \vee a\overline{c}\overline{d} \vee a\overline{b}\overline{c} \vee ab\overline{d}$$

Mivel fenti egyenlet az összes prímimplikánst tartalmazza, kell még egy módszer, mellyel ki lehet választani a függvény megvalósításához feltétlenül szükséges prímimplikánsokat.

McCluskey-féle táblázatos prímtartomány-kiválasztás

Mintermek		**Prímimplikánsok**					
		bc	$\bar{b}d$	cd	$a\bar{c}\bar{d}$	$ab\bar{c}$	$ab\bar{d}$
m_0	$\bar{a}\bar{b}\bar{c}\bar{d}$						
m_1	$a\bar{b}\bar{c}\bar{d}$				x	x	
m_2	$\bar{a}b\bar{c}\bar{d}$						
m_3	$ab\bar{c}\bar{d}$				x		x
m_4	$\bar{a}\bar{b}c\bar{d}$						
m_5	$a\bar{b}c\bar{d}$						
m_6	$\bar{a}bc\bar{d}$	x*					
m_7	$abc\bar{d}$	x					x
m_8	$\bar{a}\bar{b}\bar{c}d$		x*				
m_9	$a\bar{b}\bar{c}d$		x			x	
m_{10}	$\bar{a}b\bar{c}d$						
m_{11}	$ab\bar{c}d$						
m_{12}	$\bar{a}\bar{b}cd$		x	x			
m_{13}	$a\bar{b}cd$		x	x			
m_{14}	$\bar{a}bcd$	x		x			
m_{15}	$abcd$	x		x			

Minterm		**Prímimplikánsok**			
		cd	$a\bar{c}\bar{d}$	$ab\bar{c}$	$ab\bar{d}$
m_0	$\bar{a}\bar{b}\bar{c}\bar{d}$				
m_1	$a\bar{b}\bar{c}\bar{d}$		x	x	
m_2	$\bar{a}b\bar{c}\bar{d}$				
m_3	$ab\bar{c}\bar{d}$		x		x
m_4	$\bar{a}\bar{b}c\bar{d}$				
m_5	$a\bar{b}c\bar{d}$				
m_{10}	$\bar{a}b\bar{c}d$				
m_{11}	$ab\bar{c}d$				

Az X*-el bejelölt mintermek csak egy prímimplikánsban fordulnak elő, ezért ezek mag-prímimplikánsnak számítanak és a kapcsolásban feltétlenül szerepelniük kell.

A maradék ($a\bar{c}\bar{d}, ab\bar{c}, ab\bar{d}$) prímimplikánsok feltételesen elhagyhatóak: vagy az $a\bar{c}\bar{d}$, vagy pedig az $ab\bar{c}, ab\bar{d}$ együttesen szükségesek az m_1 és m_3 mintermek megvalósításához.

A példában szereplő logikai függvény legegyszerűbb (legoptimálisabb) alakja:

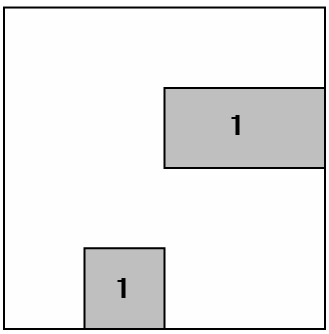
$$f(a,b,c,d) = bc \vee \bar{b}d \vee a\bar{c}\bar{d}$$

Kombinációs hálózatok
Elemi megadási módok, összefoglalás

Logikai függvények egyszerűsítése KV-diagrammal:

a) „1”-es cellák mezőinek összevonása

•Logikai függvény leírása az „1”-es mezőkkel



$$f = \ldots \vee \ldots \vee \ldots$$

Diszjunktív normálalak

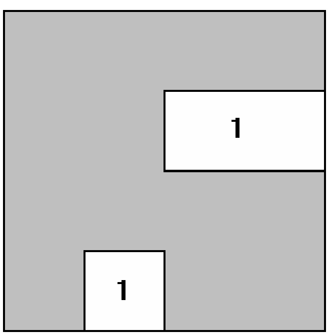
(AND/OR)

Ugyanez „de Morgan”-átalakítással:

$$f = \overline{\overline{\ldots} \wedge \overline{\ldots} \wedge \overline{\ldots}}$$

(NAND)

•Logikai függvény leírása az „1”-es mezők komplemenseivel



$$\overline{f} = (\ldots \vee \ldots \vee \ldots) \wedge (\ldots \vee \ldots$$

(OR/AND)

A „de Morgan”-tétellel átalakítva:

$$f = \overline{(\ldots \vee \ldots \vee \ldots)} \vee \overline{(\ldots \vee \ldots}$$

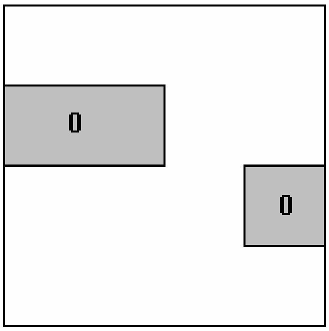
(NOR/OR)

Kombinációs hálózatok
Elemi megadási módok, összefoglalás

Logikai függvények egyszerűsítése KV-diagrammal:

b) „0”-ás cellák mezőinek összevonása

•Logikai függvény leírása a „0”-ás mezőkkel

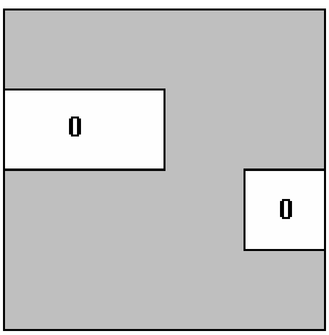


$$\overline{f} = \ldots \vee \ldots \vee \ldots$$ (AND/OR)

A „de Morgan”-tétellel átalakítva:

$$f = \overline{\ldots} \wedge \overline{\ldots} \wedge \overline{\ldots}$$ (NAND/AND)

•Logikai függvény leírása a „0”-ás mezők komplemenseivel



$$f = (\ldots \vee \ldots \vee \ldots) \wedge (\ldots \vee \ldots$$ Konjunktív normálalak (OR/AND)

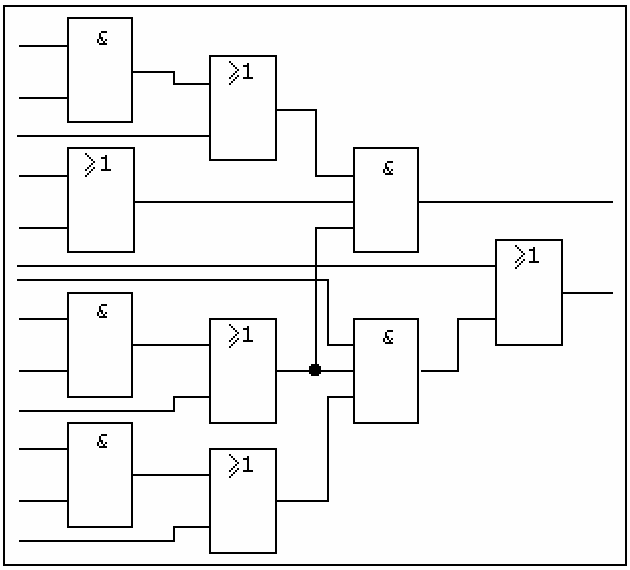
A „de Morgan”-tétellel átalakítva:

$$f = \overline{\overline{(\ldots \vee \ldots \vee \ldots)} \vee \overline{(\ldots \vee \ldots}}$$ (NOR)

A fenti kétfokozatú kombinációs hálózatok közül az AND/OR (diszjunktív normálalak) és az OR/AND (konjunktív normálalak) áttekinthetőségüknél fogva kiemelt jelentőségűek.

Kombinációs hálózatok
AND/OR hálózatok átalakítása

AND/OR hálózatnak nevezzük azokat a többfokozatú , több be- és kimenettel rendelkező kombinációs hálózatokat, amelyeknél a jel az összes útvonalon AND és OR függvényeken felváltva halad keresztül.



A gyakorlatban a NAND és NOR kapuáramkörök terjedtek el előnyösebb technológiai megvalósíthatóságuk miatt.

→ átalakítás szükséges

Az átalakítás egy egyszerű háromlépéses eljárás, mely a „De Morgan”-féle tételen alapszik.

Példa kétfokozatú hálózatok átalakítására:

Diszjunktív normálalak:

$$a + \overline{bc} + ad + abcd \qquad \rightarrow \qquad \overline{\overline{a} \cdot bc \cdot \overline{ad} \cdot \overline{abcd}}$$

Konjunktív normálalak:

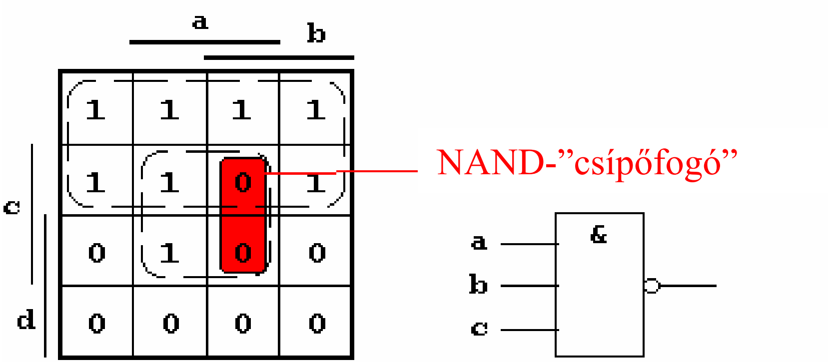
$$\overline{a} \cdot (b + c) \cdot (a + b + c + d) \qquad \rightarrow \qquad \overline{a + \overline{b + c} + \overline{a + b + c + d}}$$

1. NAND-”csípőfogó

Logikai függvények megvalósítása és egyszerűsítése során előfordulhat a következő helyzet:

- A kapcsolás diszjunktív normálalakban lett kifejlesztve
- Csak NAND kapukat lehet felhasználni
- A KV-diagramban nagy cellatartományokat (prímimplikánsokat) lehet képezni, amelyekben viszont néhány „zavaró” „0”-mező is van

Példa:



A szaggatott vonallal bejelölt cellatartományok az alábbi diszjunktív normálalakot adják:

$$f(a,b,c,d) = \bar{d} \vee ac$$

Az eredeti függvény megvalósításához szükség van egy NAND-”csípőfogóra”, ami az előbbi függvényből a két nem kívánt „1”-est kicsípi.

$$f(a,b,c,d) = (\bar{d}\,\overline{(abc)}) \vee (ac\overline{(abc)})$$

$$\overline{(abc)}$$ —-”csípőfogó”

NAND megvalósítás:

$$f(a,b,c,d) = \overline{\overline{(\bar{d}\,\overline{(abc)})} \cdot \overline{(\overline{\overline{ac}}\,\overline{(abc)})}}$$

2. Egyszerűsítés „don't care”-cellákkal

Valós problémák megoldása során előadódhatnak olyan függvények, melyeknél bizonyos bemeneti variációkhoz nincs egyértelmű függvényérték specifikálva. → „don't care” eset (többnyire akkor, ha a bemeneti változók adott értékei a probléma jellegéből adódóan nem fordulhatnak elő.)

Az adott cella értékének tetszőleges megválasztása lehetővé teszi a nagyobb prímimplikánsok kialakítását és ezzel az egyszerűbb függvénymegvalósítást.

Példa: $f(a,b,c,d) = m_0 \vee m_1 \vee m_2 \vee m_5 \vee m_6 \vee m_7 \vee m_{11}$

valamint m_9 - „don'care”

f(a,b,c,d) — a=2^0, b=2^1, c=2^2, d=2^3

0	1	3	2
4	5	7	6
12	13	15	14
8	9	11	10

f(a,b,c,d) — a, b, c, d

1	1	0	1
0	1	1	1
0	0	0	0
0	d	1	0

Megoldás I: m_9 (d =„1”)

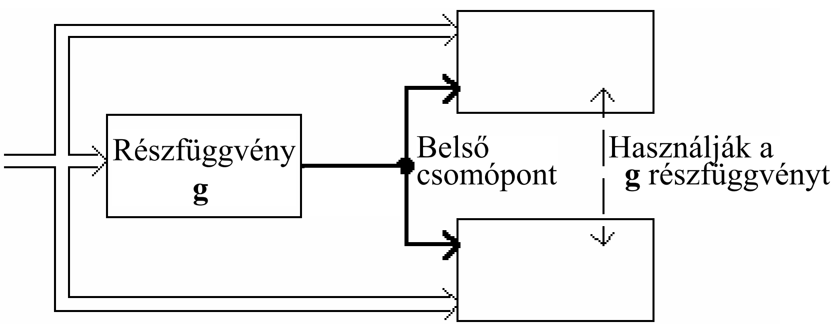
$$f(a,b,c,d) = a\bar{c}d \vee \bar{a}\bar{c}\bar{d} \vee a\bar{b}\bar{d} \vee bc\bar{d}$$

Megoldás II: M_9 (d =„0”)

$$f(a,b,c,d) = (\bar{c} \vee \bar{d})(b \vee \bar{d})(a \vee \bar{d})(a \vee b \vee \bar{c})(\bar{a} \vee \bar{b} \vee c \vee d)$$

Megvalósítás	*pin count*
kDNA (összehasonlítás)	c = 7·4 + 7 = 35
DNA	c = 4·3 + 4 = 16
KNA	c = 3·2 + 3 + 4 + 5 = 18

Ha több függvény közös adottságokkal (pl. azonos bemenő változókkal) rendelkezik, lehetőség van ezeket egy közös hálózatban megvalósítani, ami egyszerűsítésekre ad lehetőséget.
A függvénycsoportot megvalósító közös kombinációs hálózatnak ekkor természetesen több kimenete van.
Az egyszerűsítés alapja, hogy olyan részfüggvények alakíthatók ki, melyeket aztán több függvény is használni tud.



A jelelosztáshoz szükséges belső csomópontok hátránya, hogy
• növelik a kimeneti terhelhetőséget (fan out) és
• növelik a jelterjedési időt

Példa:

f(a,b,c,d)

1	1	0	1
0	1	1	1
0	0	0	0
0	d	1	0

g(a,b,c,d)

1	0	0	1
0	1	1	0
0	0	1	0
1	d	1	1

A példában van közös részfüggvény, viszont a „don't care" esetet célszerű a két függvénynél eltérő módon megvalósítani:

$$f(1,0,0,1) = 1$$
$$g(1,0,0,1) = 0$$

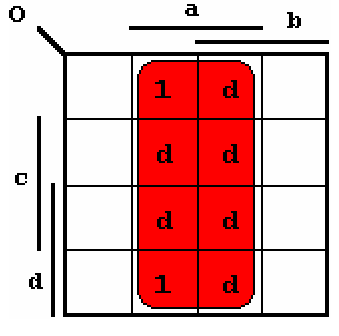
1. Példa:

BCD→GRAY
kódoló hálózat tervezése

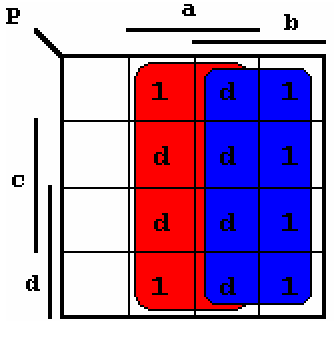
Feladat:
Minden kimenő változóhoz felállítani a KV-diagramot, ebből egyszerűsítés után az adott kimenet függvényét megvalósító kapcsolási lehetőségeket meghatározni.

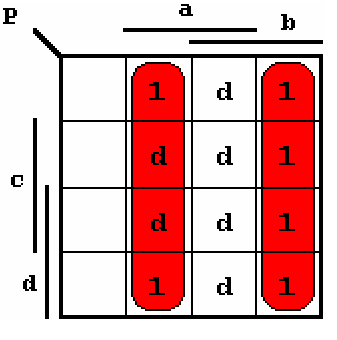
Az összes lehetséges 16 cellából a KV-diagramban csak az „1”-eket és „don't care” cellákat jelöltük be (d: pszeudotetrádok).

BCD-kód				Gray-kód			
bemenő-változók				kimenő-változók			
A	**B**	**C**	**D**	**O**	**P**	**Q**	**R**
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1



$O = A$





$P = A + B$

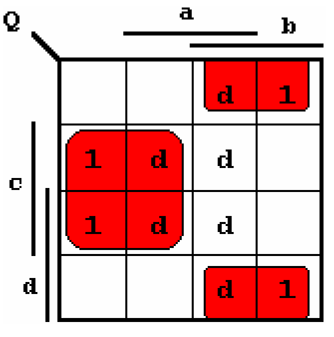
vagy

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

$P = A \oplus B$

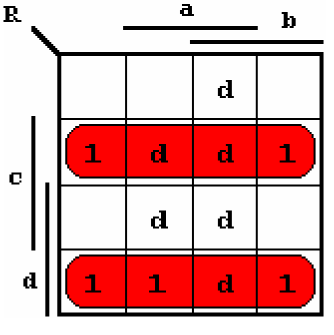
1. Példa:

BCD→GRAY kódoló hálózat tervezése (folytatás)



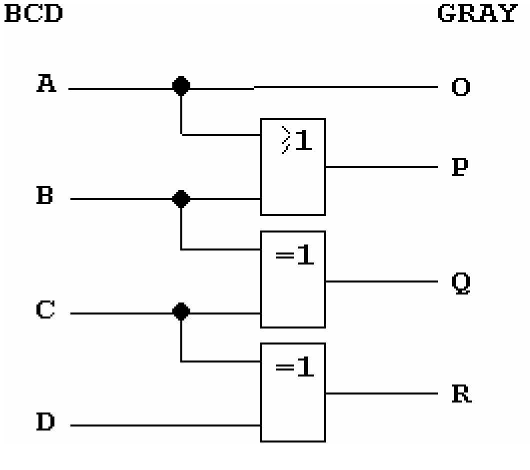
$$Q = (C \cdot \overline{B}) + (\overline{C} \cdot B)$$

$$Q = C \oplus B$$

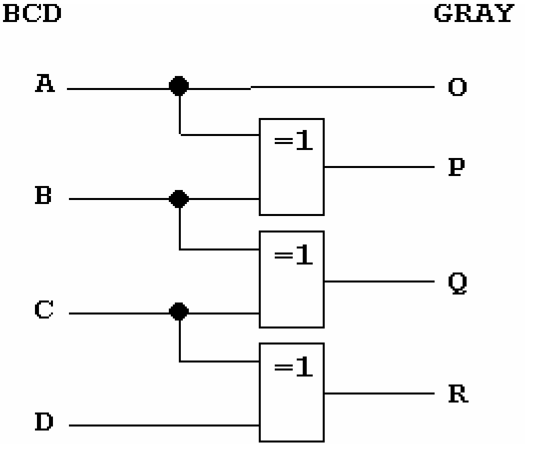


$$R = (C \cdot \overline{D}) + (\overline{C} \cdot D)$$

$$R = C \oplus D$$



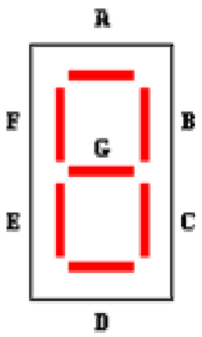
vagy



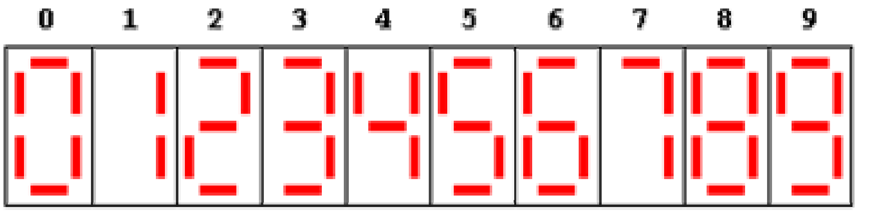
Mindkét megvalósítás működését tekintve egyenértékű, viszont a második változatnak az azaz előnye, hogy csak egyfajta kapuáramkörre (XOR) van szükség.

2. Példa:

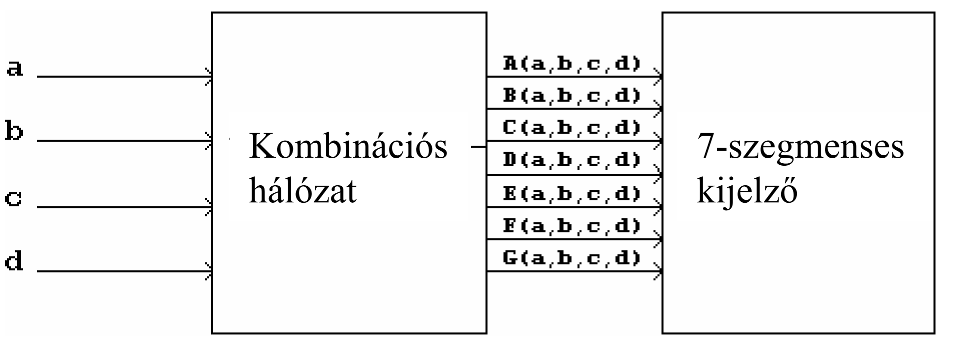
Kombinációs hálózat tervezése hétszegmenses decimális kijelzőhöz



Összerendelés: Szegmensek ⇔ decimális szám



A kapcsolás felépítése:



Igazságtáblázat:

BCD-kód			
Bemeneti változók			
d	**c**	**b**	**a**
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1

Hétszegmenses-kód						
Kimeneti változók						
A	**B**	**C**	**D**	**E**	**F**	**G**
1	1	1	1	1	1	0
0	1	1	0	0	0	0
1	1	0	1	1	0	1
1	1	1	1	0	0	1
0	1	1	0	0	1	1
1	0	1	1	0	1	1
1	0	1	1	1	1	1
1	1	1	0	0	0	0
1	1	1	1	1	1	1
1	1	1	1	0	1	1

KV-diagramok:

2. Példa:

Kombinációs hálózat tervezése hétszegmenses decimális kijelzőhöz (folytatás)

B

1	1	1	1
1	0	1	0
d	d	d	d
1	1	d	d

C

1	1	1	0
1	1	1	1
d	d	d	d
1	1	d	d

D

1	0	1	1
0	1	0	1
d	d	d	d
1	1	d	d

E

1	0	0	1
0	0	0	1
d	d	d	d
1	0	d	d

F

1	0	0	0
1	1	0	1
d	d	d	d
1	1	d	d

G

0	0	1	1
1	1	0	1
d	d	d	d
1	1	d	d

Kombinációs hálózatok
Függvények feldarabolása

Függvények részfüggvényekre bontásával is elérhető optimalizálás, különösen komplex, áttekinthetetlen hálózatok esetében.

A szétdarabolásnál az alábbi vezérelveket kell figyelembe venni:
- A részfüggvények kevesebb bemeneti változóval rendelkezzenek
- A részfüggvények (-hálózatok) többszörösen felhasználhatók legyenek

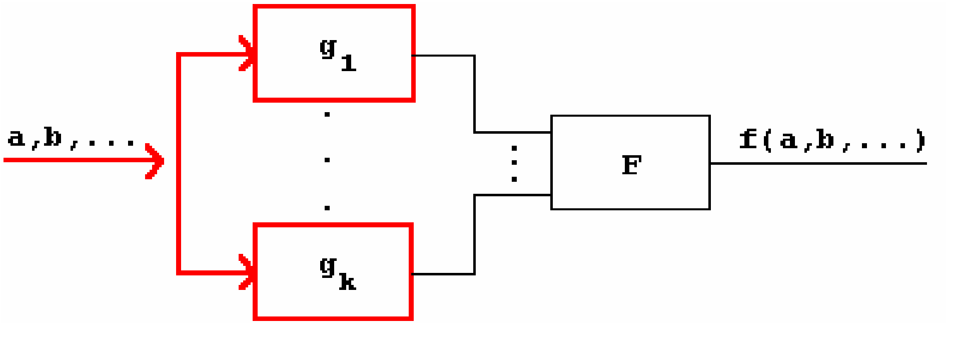
$f(a,b,...) = F(g_1(Q_1),...g_k(Q_k))$ ahol $g_i(Q_i)$ az i-dik részfüggvény
$i=1,\dots,k$ és $Q_i \subseteq \{a,b,\dots\}$

a) Függvényfeldarabolási módszer I.:

Diszjunktív függvényszétbontás

Az $f(a,b,...) = F(g_1(Q_1),...g_k(Q_k))$ függvény akkor **diszjunktív feldarabolható**, ha az összes i,j= 1,…,k és i≠j megfelel az alábbi feltételnek:

$$Q_i \cap Q_j = 0, \qquad \bigcup_{i=1}^{k} Q_i = \{a,b,...\}$$



b) Függvényfeldarabolási módszer II.:

Iteratív függvényszétbontás

Egy logikai függvény akkor **iteratívan szétdarabolható** m db részfüggvényre, ha létezik egy olyan $g_i(Q_i)$ logikai függvény (i=1,…,m), melyre igaz az alábbi feltétel:

$$f(a,b,...) = g_m\left(...g_2\left(g_1(Q_1),Q_2\right)...,Q_m\right)$$

ahol $g_i(Q_i), i = 1,...,m$

b) Függvényfeldarabolási módszer II. (folytatás):

Iteratív függvényszétbontás

Egy logikai függvény akkor **iteratívan szétdarabolható** m db részfüggvényre, ha létezik egy olyan $g_i(Q_i)$ logikai függvény (i=1,…,m), melyre igaz az alábbi feltétel:

$$f(a,b,...) = g_m\left(...g_2\left(g_1(Q_1),Q_2\right)...,Q_m\right)$$

ahol $g_i(Q_i), i = 1,...,m$

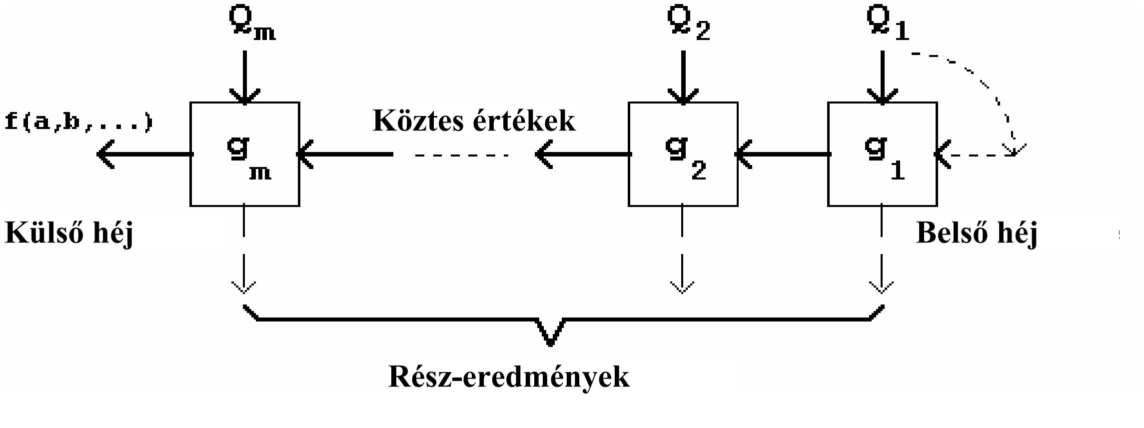
Példa: iteratív függvényszétbontás 2 részfüggvényre:

$$f(a,b,...) = g_2(g_1(Q_1),Q_2)$$

Példa: iteratív függvényszétbontás 3 részfüggvényre:

$$f(a,b,...) = g_3(g_2(g_1(Q_1),Q_2),Q_3)$$

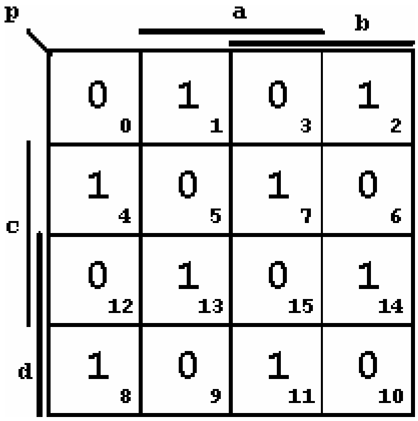
Ezen kapcsolások lánc-szerkezetűek *(pipeline architecture)*:



Kombinációs hálózatok
Függvények feldarabolása, példák

1. Példa: Paritásvizsgálat (diszjunktív feldarabolás)

Kimutatja, hogy egy adott számú jelből az „1”-es értékűek száma páros *(even parity)* vagy páratlan *(odd parity)*.



Az „1”-esek azokat a cellákat jelölik, amelyeknél az „1”-es értékű bementi változók darabszáma páratlan.

DNA:

$$f(a,b,c,d) =$$
$$= m_1 \vee m_2 \vee m_4 \vee m_7 \vee m_8 \vee m_{11} \vee m_{13} \vee m_{14}$$

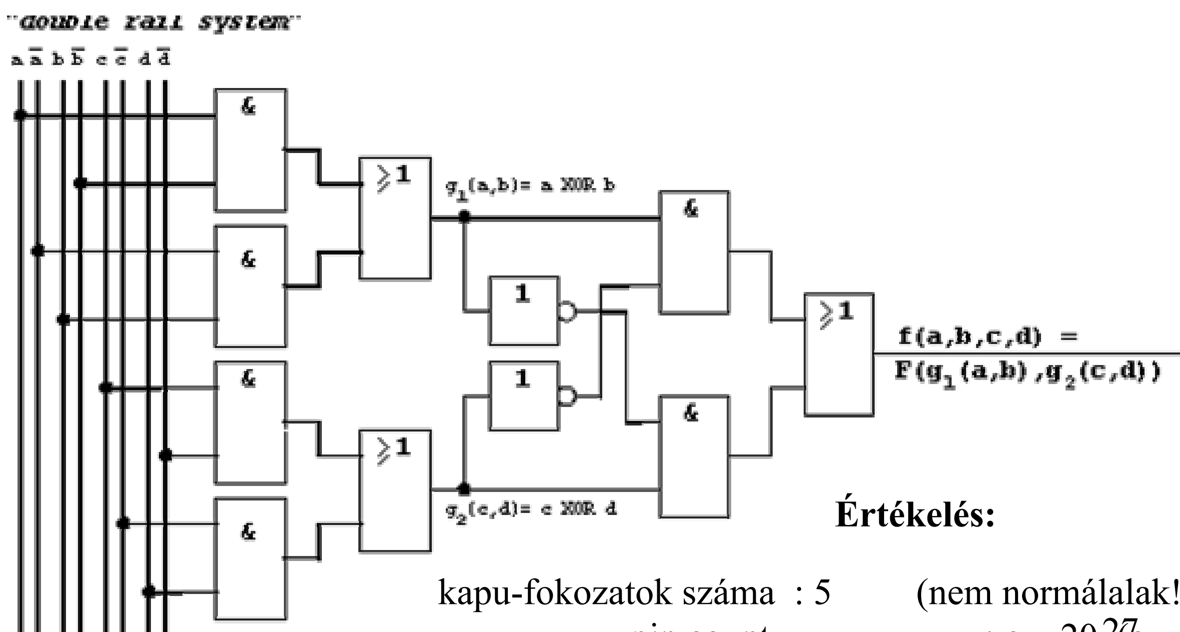
Függvénymegvalósítás értékelése:

kapu-fokozatok száma (jelterjedési idő) : 2 (normálalak!)
pin count : c = 40 (c = 8·4 + 8)

Egyszerűbb megoldás adódik, ha a XOR függvényt használjuk:

$$f(a,b,c,d) = a \oplus b \oplus c \oplus d$$

és az asszociativitás miatt: $f\,)a,b,c,d) = (a \oplus b) \oplus (c \oplus d)$



Értékelés:

kapu-fokozatok száma : 5 (nem normálalak!)
pin count : c = 20 (c = 9·2 + 2·1 stb.)

2. Példa: Összeadó kapcsolások (iteratív feldarabolás)

Két bináris számot összeadó kapcsolást lánc-szerkezetben lehet megvalósítani.

c_{IN}	b	a	s	c_{OUT}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

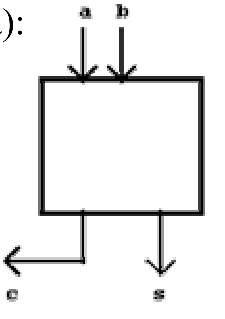
Teljes-összeadó áramkör igazság-táblázata.

a,b - összeadandó bitek

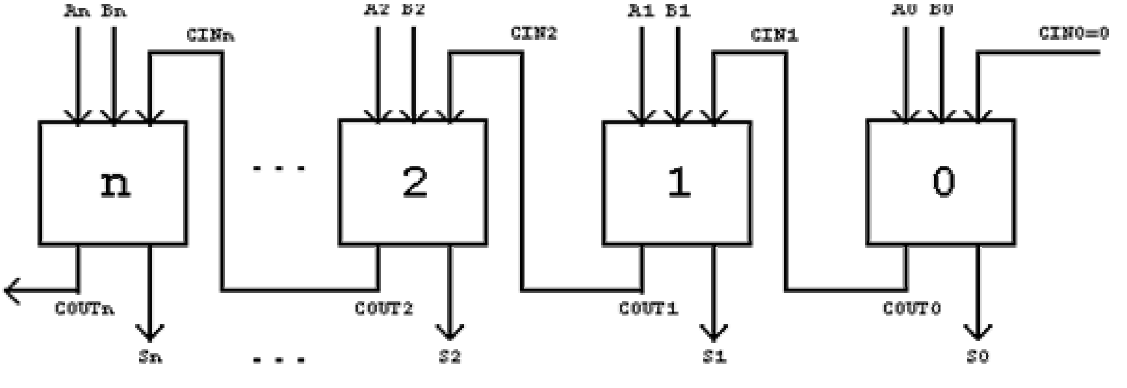
s - eredmény

c - átvitel-bit (carry bit)

A zárójellel jelölt tartomány a fél-összeadó áramkört jelöli (nincs carry in = megelőző fokozat c_{out}-ja):



n-helyiértékes teljes-összeadó lánc-szerkezete:



$$s_i = a_i\overline{b_i}\overline{c_i} \vee \overline{a_i}b_i\overline{c_i} \vee \overline{a_i}\overline{b_i}c_i \vee a_ib_ic_i \qquad \text{ahol } (c_i = c_{ini})$$

$$c_{outi} = a_ib_i \vee a_ic_i \vee b_ic_i$$

Fenti függvényekkel egyenértékű: $s_i = a_i \oplus b_i \oplus c_i$

$$c_{outi} = \text{majoritás}(a_i, b_i, c_i)$$

c) Függvényfeldarabolási módszer III.:

Shannon féle függvényszétbontás

Minden logikai függvény felbontható az alábbi módon:

$$f(a,...,x,...) = x \cdot f(a,...,1,...) \vee \overline{x} \cdot f(a,...,0,...)$$

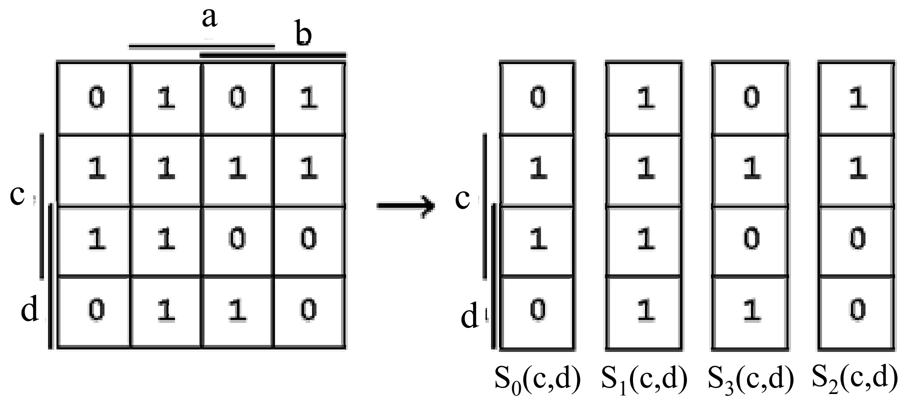
f(a,…,1,…) függvény szolgáltatja a kimeneti értéket ha x=1
f(a,…,0,…) függvény szolgáltatja a kimeneti értéket ha x=0

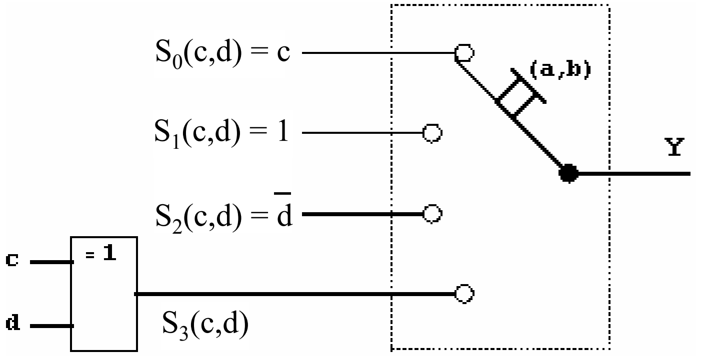
Mivel a módszer minden függvényre alkalmazható, a felbontott (kettéválasztott) függvény részfüggvényei ezzel a módszerrel tovább bonthatóak.

$$\begin{aligned} f(a,b) &= a \cdot f(1,b) \vee \overline{a} \cdot f(0,b) \\ &= a \cdot b \cdot f(1,1) \vee a \cdot \overline{b} \cdot f(1,0) \vee \overline{a} \cdot b \cdot f(0,1) \vee \overline{a} \cdot \overline{b} \cdot f(0,0) \end{aligned}$$

Háromváltozós függvény szétbontása:

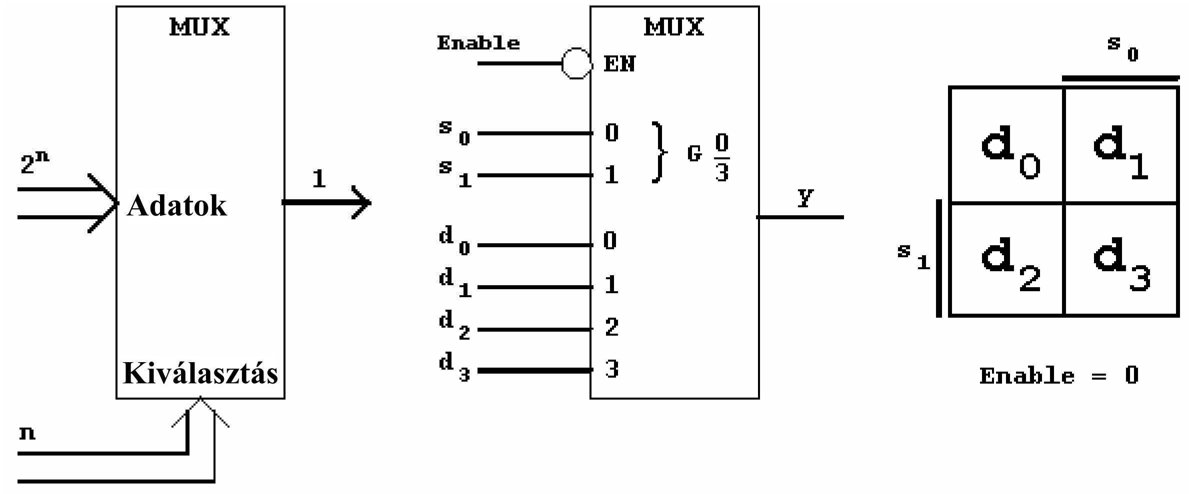
$$\begin{aligned} f(a,b,c) &= a \cdot f(1,b,c) \vee \overline{a} \cdot f(0,b,c) \\ &= a \cdot b \cdot f(1,1,c) \vee a \cdot \overline{b} \cdot f(1,0,c) \vee \overline{a} \cdot b \cdot f(0,1,c) \vee \overline{a} \cdot \overline{b} \cdot f(0,0,c) \\ &= a \cdot b \cdot S_3(c) \vee a \cdot \overline{b} \cdot S_1(c) \vee \overline{a} \cdot b \cdot S_2(c) \vee \overline{a} \cdot \overline{b} \cdot S_0(c) \end{aligned}$$





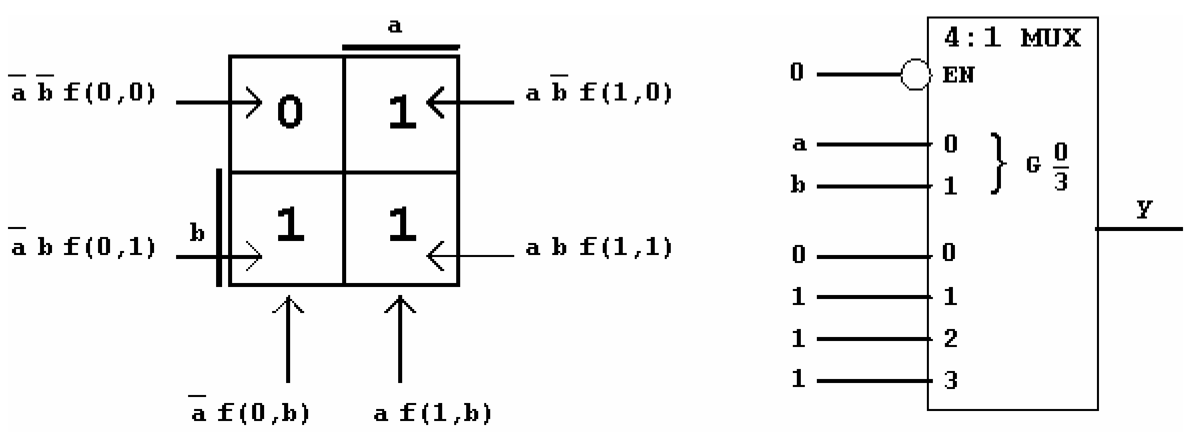
„Digitális kapcsoló”

A multiplexer jelvezetékek közötti kapcsolási feladatot tölt be.
Alapfeladatához szükséges: 1 db kimenet, y
n db kiválasztó (cím)bemenet, s_i
2^n db adatbemenet, d_k

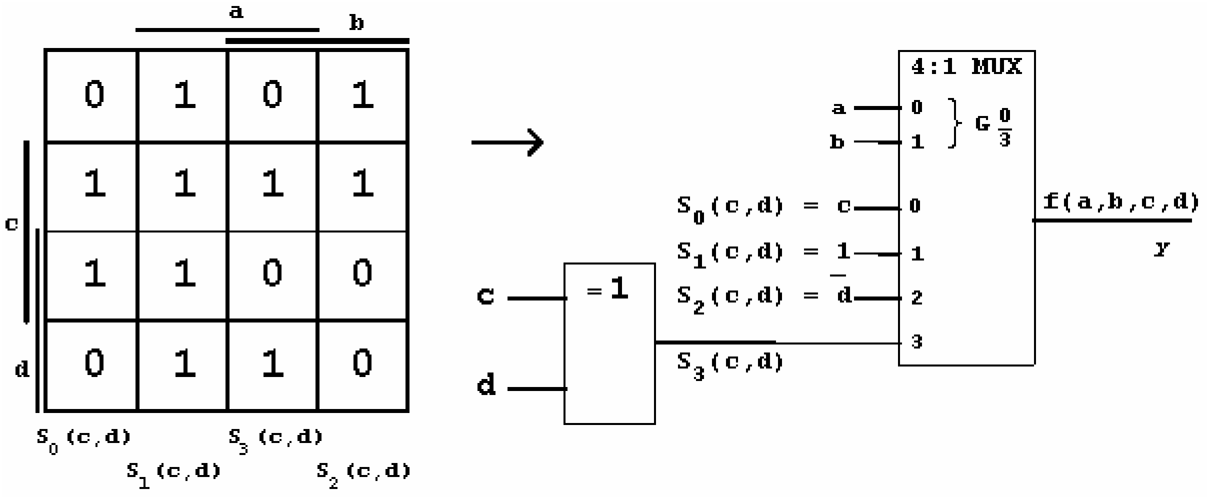


a) „adat-kapcsoló” b) DIN-jelölés c) KV-diagram

Shannon-függvénybontással (és multiplexerrel) megvalósított OR függvény:



Példafüggvény megvalósítása multiplexerrel



A Shannon-függvénybontás kétlépcsős alkalmazása: