



## Loogikasünteesi põhiprobleemid

- **Loogikafunktsiooni esituse optimeerimine**
  - kahe-tasemelise esituse minimeerimine
  - kahend-otsustus diagrammide (BDD – Binary Decision Diagrams) optimeerimine
- **Mitme-tasemeliste kombinatsioonloogikavõrkude (-skeemide) süntees**
  - pindala, viite, võimsustarbe ja/või testitavuse optimeerimine
- **Automaatide optimeerimine**
  - olekute minimeerimine, kodeerimine
- **Mitme-tasemeliste mäluga loogikavõrkude (-skeemide) süntees**
  - pindala, viite, võimsustarbe ja/või testitavuse optimeerimine
- **Sidumine loogikaelementide teegiga**
  - elementide optimaalne valik

# Loogikasünteesi eesmärgid

**Lähteülesanne**  
(tõeväärtustabel)

abc	xyz
000	111
001	011
010	101
011	010
100	000
101	010
110	000
111	101

**Minimeeritult**  
(implikant-kate)

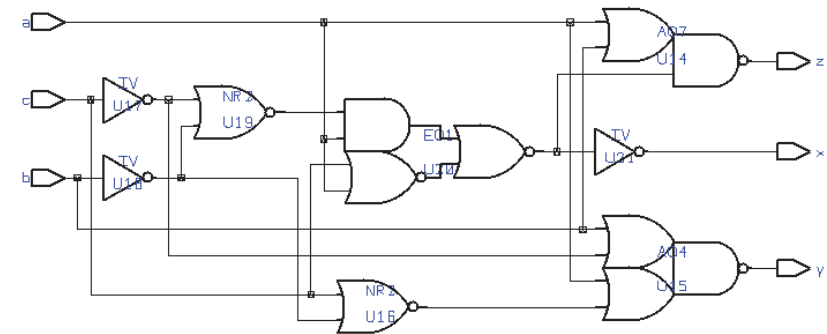
abc	xyz
111	101
-01	010
0-0	101
00-	011
0-1	010

**JA-EI (AND-OR):**

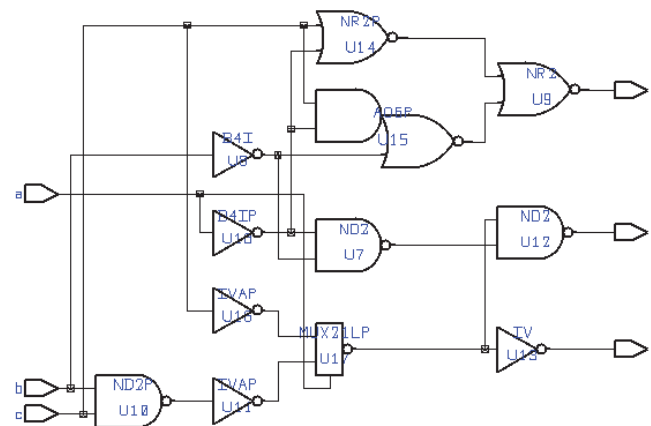
$$x = abc + \bar{a}\bar{c}$$

$$y = \bar{b}c + \bar{a}\bar{b} + \bar{a}c$$

$$z = abc + \bar{a}\bar{c} + \bar{a}\bar{b}$$



**pindala** 12.0  
**viide** 2.73 ns  
**võimsus** 11.3  $\mu$ W



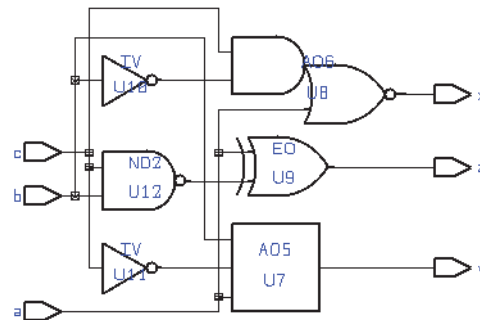
**pindala** 22.0  
**viide** 1.57 ns  
**võimsus** 24.1  $\mu$ W



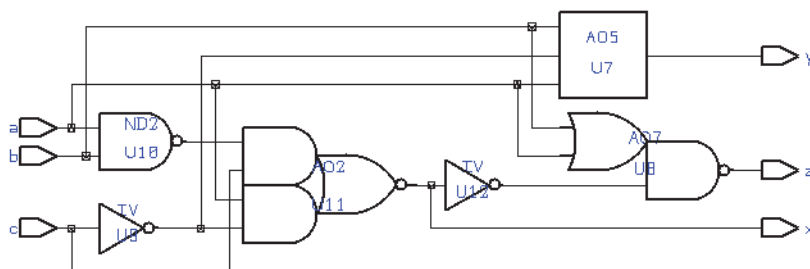
TTÜ1918



## Loogikasünteesi eesmärgid (2)



pindala **11.0**  
viide **1.84 ns**  
võimsus **9.2  $\mu\text{W}$**

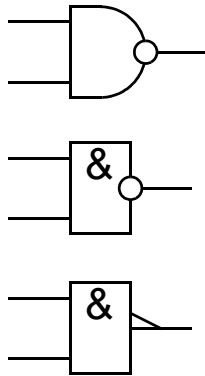


pindala **10.0**  
viide **2.26 ns**  
võimsus **10.2  $\mu\text{W}$**

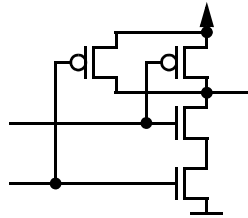


# Baaselemendid – NAND vs NOR

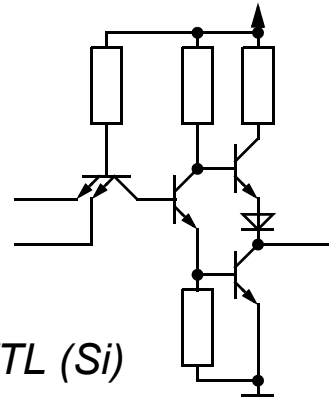
2-NAND



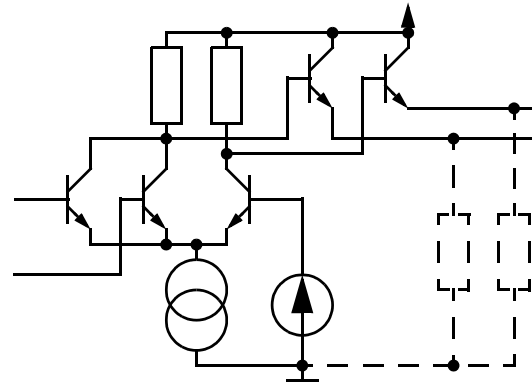
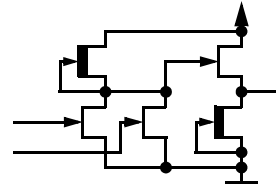
CMOS (Si)



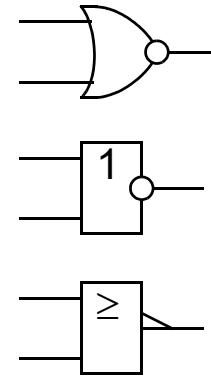
TTL (Si)



DCFL (GaAs)  
Direct-coupled FET



2-NOR



ECL (Si)



## Loogikafunktsioonide süsteem

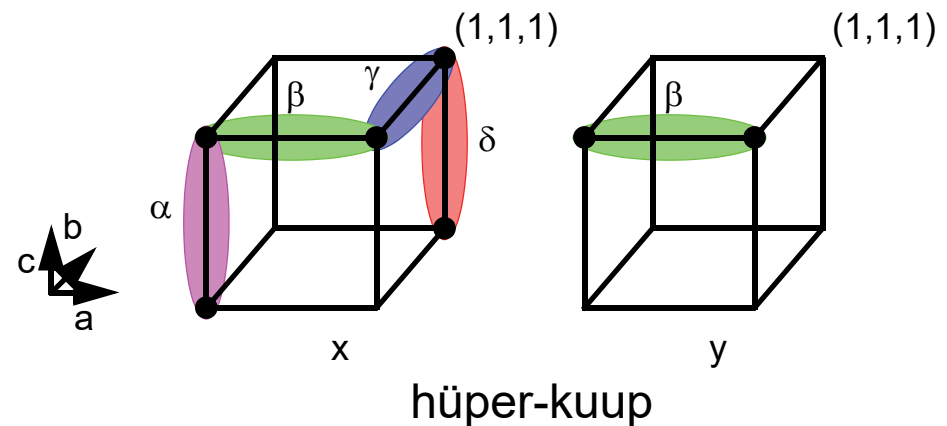
- **Kombinatsiooniskeemi “musta kasti” mudel**
- **Defineeritud Boole'i algebra baasil –  $(B, +, *, \sim), B = \{0, 1\}$**
- **Loogikafunktsioonid võivad olla**
  - **mitme väljundiga (funktsioonide süsteem) –  $f: B^n \rightarrow B$  ja  $f: B^n \rightarrow B^m$**
  - **osaliselt (mittetäielikult) määratud –  $f: B^n \rightarrow \{0, 1, -\}^m$  (ka  $f: B^n \rightarrow \{0, 1, *\}^m$ )**
    - **sõltub funktsiooni kasutamisest, nt. võimatud sisendkombinatsioonid**
  - **ON-set –  $F_f$  – selline funktsiooni määramispiirkonna osa, kus  $f$  on tõene**
  - **OFF-set –  $R_f$  – selline funktsiooni määramispiirkonna osa, kus  $f$  on väär**
  - **DC-set –  $D_f$  – selline funktsiooni määramispiirkonna osa, kus  $f$  on määramata (pole oluline)**
  - **Funktsioonide süsteemis defineeritud iga komponendi jaoks**

## Definitsioonid ja esitusviisid

- *muutuja* (variable)
- *literaal* (literal) ehk *algterm* – muutuja ja selle täiend
- *korrutis* (product) ehk *kuup* (cube) ehk *elementaarkonjunktsioon* – literaalide korrutis
- *implikant* (implicant) ehk intervall – funktsiooni väärtust (tavaliselt 1) määrav konjunktsioon
  - *hüperkuup* (hypercube)
- *minterm* – kõiki sisendmuutujaid sisaldav implikant
  - sõlm hüperkuubis
- *tõeväärtustabel* (truth table)
- funktsiooni kõikide mintermide loetelu
- *implikanttabel* (implicant table) ehk *intervalltabel* ehk *kate* (cover)
  - funktsiooni defineerimiseks piisavate implikantide loetelu

abc	xy
000	10
001	11
101	11
110	10
111	10

abc	xy
00-	10
-01	11
1-1	10
11-	10



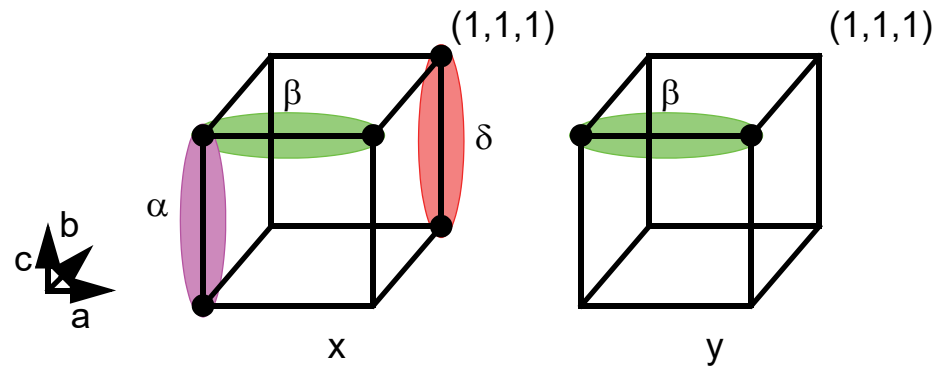


## Definitsioone – kate

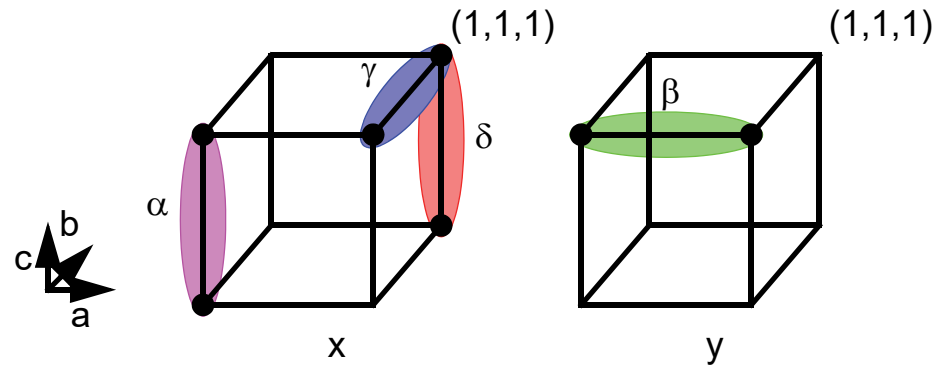
- ***Miimumkate*** (minimum cover)
  - kate vähima implikantide arvuga
  - globaalne optimum
- ***Minimaalne kate*** (minimal cover) ehk ***liiasuseta kate*** (irredundant cover)
  - kate, mis ei sisaldu üheski teises kattes
  - ühtegi implikanti ei saa eemaldada
  - lokaalne optimum
- ***Lihtimplikant*** (prime implicant) – ei sisaldu üheski teises implikandis
- ***Lihtkate*** (prime cover) – kate lihtimplikantidest
- ***Oluline*** (essential) lihtimplikant – leidub minterm, mis on kaetud ainult selle lihtimplikandi poolt



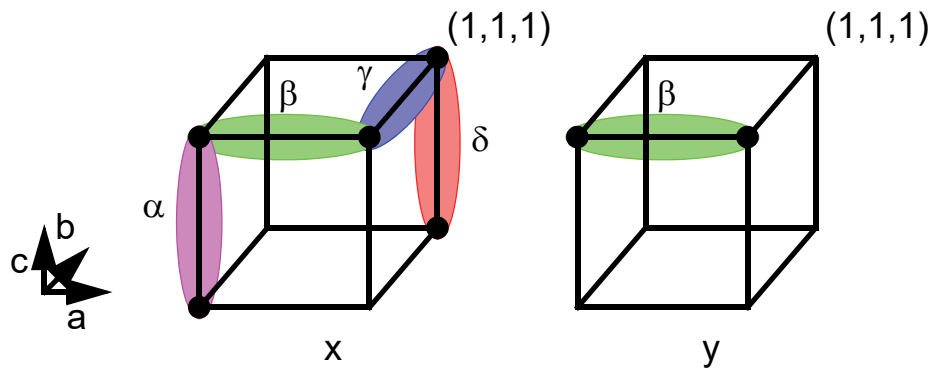
olulised  
lihtimplikandid  
 $x - \alpha$  &  $\delta$   
 $y - \beta$



lihtkate  
miinimumkate



lihtkate  
minimaalne kate



lihtkate



# Loogikafunktsioonide minimeerimine

- **Süntees ja optimeerimine**
  - lähtekirjelduse (tabel, skeem, HDL) teisendamine abstraktseks mudeliks
  - teisendused abstraktsel mudelil (sobivad analüüsiks ja masintöötluks)
  - sidumine elementidega teegist
- **Täpsed meetodid (nt. Quine-McCluskey meetod)**
  - leiavad miinimumkate
  - tihtipeale võimatu suurte funktsioonide korral
- **Heuristilised meetodid (MINI, PRESTO, ESPRESSO, ...)**
  - leiavad minimaalsed katted (miinimumkate leidmine võimalik)
- **Quine'i teoreem – miinimumkate on lihtkate**
  - → miinimumkate otsimisel võib piirduda lihtimplikantidega
- **Quine-McCluskey meetod**
  - põhisammud – [1] leia lihtimplikandid, [2] leia miinimumkate
- **Lihtimplikantide tabel**
  - read – mintermid / veerud – lihtimplikandid [ *või vastupidi... :-)* ]
  - eksponentsiaalne suurus! –  $2^n$  mintermi (mida võib rühmitada)
  - kuni  $3^n/n$  lihtimplikanti (osadel funktsioonidel on neid palju vähem)

## Minimeerimine == implikantide kleepimine

- Erinevus täpselt ühes järgus (vt. kleepimisseadused)
- Võib vaadelda sulgude ette toomisega
  - kleepuvad:  $a b c + a \bar{b} c = a c (b + \bar{b}) = a c (1) = a c$
  - ei kleepu:  $a b c + a \bar{b} \bar{c} = a (b c + \bar{b} \bar{c})$

- $y = \bar{a} \bar{b} \bar{c} + \bar{a} \bar{b} c + \bar{a} b \bar{c} + a \bar{b} c$
- $y = \bar{a} \bar{b} (\bar{c} + c) + \bar{a} b \bar{c} + a \bar{b} c$
- $y = \bar{a} \bar{b} + \bar{a} b \bar{c} + a \bar{b} c$  – pole minimaalne?!

- $y = \bar{a} \bar{b} \bar{c} + \bar{a} \bar{b} c + \bar{a} b \bar{c} + a \bar{b} c$
- $y = \bar{a} \bar{b} \bar{c} + \bar{a} \bar{b} c + \underline{\bar{a} b \bar{c}} + \bar{a} b c + \underline{\bar{a} b c} + a \bar{b} c$
- $y = \bar{a} \bar{b} (\bar{c} + c) + \bar{a} c (\bar{b} + b) + \bar{b} c (\bar{a} + a)$
- $y = \bar{a} \bar{b} + \bar{a} c + \bar{b} c$

- Sarnane dubleerimine töötab ka KNK korral

	c		b
a	1	1	0
	0	1	0

	c		b
a	1	1	0
	0	1	0

## DNK või KNK

- Implikantide kleepimisel erinevust pole
- Katte leidmisel erinevust pole
- Erinevus seisneb tulemuse esitamises
- De Morgani seaduse abil saab ühest esitusviisist teise
  - DNK 1-de ja KNK 0-de järgi on funktsioon
  - DNK 0-de ja KNK 1-de järgi on funktsiooni eitus

		d			
		c			
b		1	0	0	1
		1	1	1	1
a		0	1	0	0
		1	1	1	1

$$f = b \bar{c} d + \bar{a} \bar{d} + \bar{a} b + a \bar{b}$$

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

DNK 1-dest  
KNK 0-dest

$$\bar{f} = \bar{a} \bar{b} d + a b \bar{d} + a b c$$

$$f = (\bar{a} \bar{b} d + a b \bar{d} + a b c)'$$

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

DNK 0-dest  
eitus  
De Morgan!

## Lihtimplikantide leidmine

- Implikantide kleepimine

- erinevus täpselt ühes järgus (vt. kleepimisseadused)

- võib vaadelda sulgude ette toomisega

- kleepuvad:  $a b c + a \bar{b} c = a c (b + \bar{b}) = a c (1) = a c$

- ei kleepu:  $a b c + a \bar{b} \bar{c} = a (b c + \bar{b} \bar{c})$

- Alates kahestest kontuuridest (vähemalt üks ebaoluline muutuja implikandis) leidub alati kaks või enam paari kontuure (implikante), mis moodustava uue kontuuri (implikandi) – paaride arv on võrdne uue implikandi ebaoluliste muutujate arvuga

0	1	0	1
1	1	1	1
-	1	-	1
0	0	0	0

$$\bar{a} b d + a b d = b d$$

$$b \bar{c} d + b c d = b d$$

$$\frac{01-1}{11-1} \quad \frac{-101}{-111}$$

$$\frac{-1-1}{-1-1}$$

0	1	0	1
1	1	1	1
-	1	-	1
0	0	0	0

$$\bar{a} b + a b = b$$

$$b \bar{c} + b c = b$$

$$b \bar{d} + b d = b$$

$$\frac{01--}{11--} \quad \frac{-10-}{-11-} \quad \frac{-1-0}{-1-1}$$

$$\frac{-1--}{-1--} \quad \frac{-1--}{-1--} \quad \frac{-1--}{-1--}$$

# Lihtimplikantide leidmine - Quine meetod

$$f = \bar{a}\bar{d} + \bar{a}b + a\bar{b} + a\bar{c}d$$

$$f(a,b,c,d) = \Sigma(0,2,4,5,6,7,8,9,10,11,13)$$

- mintermid on algsed implikandid
- implikante proovitakse kleepida paarikaupa
- kaetud (ja dubleeritud) implikandid eemaldatakse
- kleepimist ja kaetute eemaldamist korratakse seni kuni enam uusi implikante ei moodustu

algus

abcd
0000
0010
0100
0101
0110
0111
1000
1001
1010
1011
1101

1. etapp

abcd	abcd
<del>0000</del>	-000
<del>0010</del>	0-10
<del>0100</del>	-010
<del>0101</del>	010-
<del>0110</del>	01-0
<del>0111</del>	100-
<del>1000</del>	10-0
<del>1001</del>	01-1
<del>1010</del>	-101
<del>1011</del>	011-
<del>1101</del>	10-1
00-0	1-01
0-00	101-

2. etapp

abcd	abcd
<del>00-0</del>	<del>10-1</del>
<del>0-00</del>	1-01
<del>000</del>	<del>101</del>
<del>0-10</del>	0--0
<del>010</del>	<del>0-0</del>
<del>010</del>	-0-0
<del>01-0</del>	<del>0-0</del>
<del>100</del>	01--
<del>10-0</del>	<del>01</del>
<del>01-1</del>	10--
-101	<del>10</del>
<del>011</del>	

tulemus

abcd
-101
1-01
0--0
-0-0
01--
10--



## Lihtimplikantide leidmine – Quine-McCluskey meetod

- **Kitsendused lihtimplikantide leidmisel**
  - mintermid grupeeritakse 1-de arvu alusel
  - kleepida proovitakse ainult naabergruppide implikante
    - vrdl. erinevust 1-de arvus!
  - ainult määramata väljundtulemus(t/i) kattev implikant on eritähistusega (nt. tärn)
    - kahe sellise implikandi kleepumisel levib tähistus edasi
    - mõte on selles, et kui implikant katab ainult määramatusi, siis pole teda kattesesse vaja
  - implikantide kleepimisel tuleks arvestada ka ebaoluliste muutujate kokkulangevusi
    - kleepuda võivad ainult need, mis sõltuvad samadest muutujatest
  
- **Käsitsi arvutamise lihtustamiseks kasutusel 10-nd kodeering**
  - “01-0”-le vastab “4 (2)”, “4/6” või “4/6 (2)”
  - kleepimisel võrreldakse, kas vahe on 2 aste –  $01\underline{0}0 \leftrightarrow 01\underline{1}0$  vs.  $4 \leftrightarrow 6$



## Lihtimplikantide leidmine (#1)

$$f = \bar{a} \bar{d} + \bar{a} b + a \bar{b} + a \bar{c} d$$

$$f(a,b,c,d) = \Sigma(0,2,4,5,6,7,8,9,10,11,13)$$

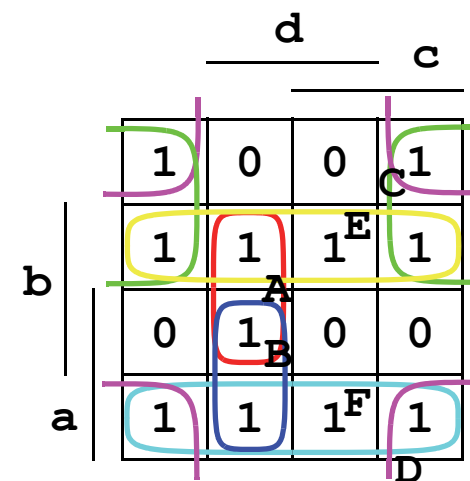
mintermid

1. etapp

2. etapp

gr.	abcd		gr.	abcd		gr.	abcd	
0	0000 *	→	0	00-0 *	→	0	0--0	C
1	0010 *	→		0-00 *	→		-0-0	D
	0100 *	→		-000 *	→	1	01--	E
	1000 *	→	1	0-10 *	→		10--	F
2	0101 *	→		-010 *	→			
	0110 *	→		010- *	→			
	1001 *	→		01-0 *	→			
	1010 *	→		100- *	→			
3	0111 *	→		10-0 *	→			
	1011 *	→	2	01-1 *	→			
	1101 *	→		-101	A			
				011- *				
				10-1 *				
				1-01	B			
				101- *				

\* - on kaetud



## Lihtimplikantide leidmine (#2)

$$f = \bar{a} \bar{d} + \bar{a} b + a \bar{b} + a \bar{c} d$$

$$f(a,b,c,d) = \Sigma(0,2,4,5,6,7,8,9,10,11,13)$$

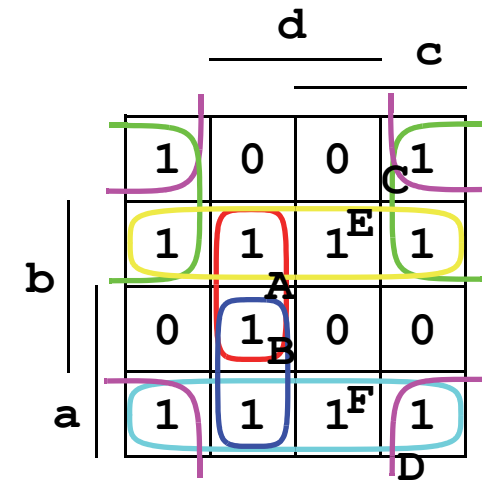
mintermid

1. etapp

2. etapp

gr.	gr.	gr.
0 0 *	0 0 (2) *	0 0 (2,4) <b>C</b>
1 2 *	0 4 *	0 0 (2,8) <b>D</b>
4 *	0 8 *	1 4 (1,2) <b>E</b>
8 *	1 2 (4) *	8 (1,2) <b>F</b>
2 5 *	2 2 (8) *	
6 *	4 (1) *	
9 *	4 (2) *	
10 *	8 (1) *	
3 7 *	8 (2) *	
11 *	2 5 (2) *	
13 *	5 (8) <b>A</b>	
	6 (1) *	
	9 (2) *	
	9 (4) <b>B</b>	
	10 (1) *	

\* - on kaetud



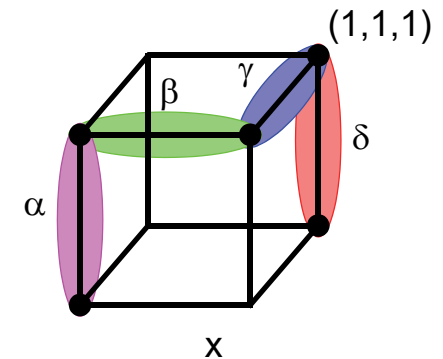
# Lihtimplikantide katte leidmine

$$x = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + a\bar{b}c + a\bar{b}\bar{c} + abc$$

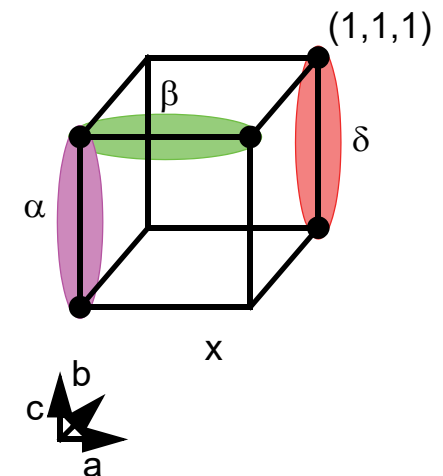
## Varased katte leidmise meetodid

- Tabeli redutseerimine
  - iteratiivne oluliste lihtimplikantide identifitseerimine, märkimine ja tabelist eemaldamine koos kaetud mintermidega
- Petrick'i meetod
  - implikandid summade korrutisena (pos)
  - viia üle korrutiste summaks (sop)
  - valida väikseim korrutis
  - pos -  $(\alpha)(\alpha+\beta)(\beta+\gamma)(\delta)(\gamma+\delta) = 1$
  - sop -  $\alpha\beta\delta + \alpha\gamma\delta = 1$
  - Lahendused -  $\{\alpha, \beta, \delta\}$  või  $\{\alpha, \gamma, \delta\}$

	abc	x
$\alpha$	00-	1
$\beta$	-01	1
$\gamma$	1-1	1
$\delta$	11-	1



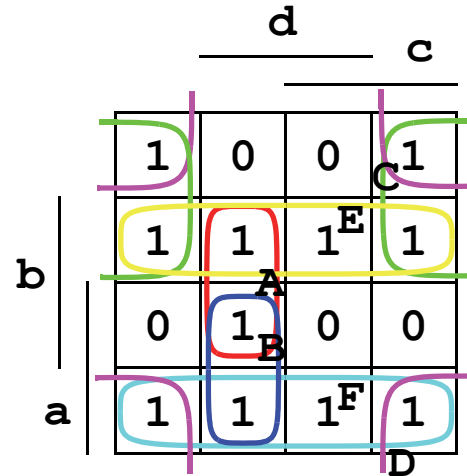
abc	$\alpha$	$\beta$	$\gamma$	$\delta$
000	1	0	0	0
001	1	1	0	0
101	0	1	1	0
110	0	0	0	1
111	0	0	1	1



# Petrick'i meetod – näiteülesanne

lihtimplikantide tabel

abcd	A	B	C	D	E	F
0000	0	0	1	1	0	0
0010	0	0	1	1	0	0
0100	0	0	1	0	1	0
1000	0	0	0	1	0	1
0101	1	0	0	0	1	0
0110	0	0	1	0	1	0
1001	0	1	0	0	0	1
1010	0	0	0	1	0	1
0111	0	0	0	0	1	0
1011	0	0	0	0	0	1
1101	1	1	0	0	0	0



ACEF:

$$f = b \bar{c} d + \bar{a} \bar{d} + \bar{a} b + a \bar{b}$$

ADEF:

$$f = b \bar{c} d + \bar{b} \bar{d} + \bar{a} b + a \bar{b}$$

BCEF:

$$f = a \bar{c} d + \bar{a} \bar{d} + \bar{a} b + a \bar{b}$$

BDEF:

$$f = a \bar{c} d + \bar{b} \bar{d} + \bar{a} b + a \bar{b}$$

$$(C+D)(C+E)(D+F)(A+E)(C+E)(B+F)(D+F)(E)(F)(A+B)=1$$

$$(C+D)(C+E)(D+F)(A+E)(B+F)(E)(F)(A+B)=1$$

$$(CC+CE+DC+DE)(DA+DE+FA+FE)(BE+FE)(FA+FB)=1$$

$$(C+DE)(AD+AF+DE+EF)(BE+EF)(AF+BF)=1$$

$$(CAD+CAF+CDE+CEF+DEAD+DEAF+DEDE+DEEF)(BEAF+BEBF+EFAF+EFBF)=1$$

$$(ACD+ACF+CEF+DE)(BEF+AEF)=1$$

$$(ACDBEF+ACDAEF+ACFB EF+ACFAEF+CEFB EF+CEFAEF+DEBEF+DEAEF)=1$$

$$ACEF+ADEF+BCEF+BDEF=1$$

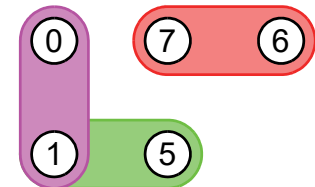
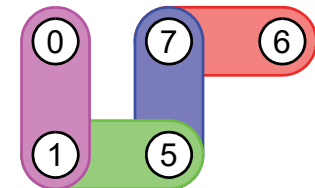
## • Maatriksesitus

- Täisarvuline lineaarplaneerimisülesanne (integer linear programming, ILP)
- Implikantide tabel on kahendmaatriks  $A$
- Valitud implikandid on kahendvektor  $x$
- Leida selline  $x$ , et
  - $Ax \geq 1$
  - valida piisav arv veerge, et kõik read oleksid kaetud
- Minimeerida  $x$  võimsust

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

## • Operatsioonid hulkadega

- Hulga katte leidmine
  - Hulk  $S$  – mintermide hulk
  - Hulk  $C$ , alamhulkade kogu ( $c_i \subseteq S$ ) – lihtimplikantide hulk
  - Leida vähim arv  $C$  elemente, et kõik  $S$  elemendid oleksid kaetud
- Hüpergraafi tipukatte leidmine
  - Sõlmede hulk – mintermide hulk
  - Hüperservade hulk – lihtimplikantide hulk
  - Tuleb leida vähim arv servi, et kõik sõlmed oleksid kaetud





# Kiirendamisvõtted

- Olulised lihtimplikandid peavad kuuluma kattesesse
  - st. need implikandid ja nende poolt kaetud mintermid võib eemaldada edasisest analüüsist
- Sõltumatud rühmad (omavahel mittesidusad alamgraafid) võib lahendada eraldi
- Implikandi domineerimine
  - kui implikant (i) on kaetud mõne domineeriva (j) poolt, siis võib ta eemaldada
  - matriksis -  $a_{ki} \leq a_{kj} \forall k$  ; tekib mittetäielikult määratud funktsioonide korral
- Mintermi domineerimine
  - kui domineeriv minterm (i) on kaetud vähemalt samade implikantide poolt, mis mõni teinegi minterm (j), siis võib ta edasisest analüüsist eemaldada, sest kõik lahendused, mis katavad (j) katavad ka (i)
  - matriksis -  $a_{ik} \geq a_{jk} \forall k$  ; tekib mintermide korral, mis on kaetud rohkem kui ühe implkandi poolt

- oluline lihtimplikant – a [0-01] ja b [-1-1]
- domineeriv implikant – b [-1-1] (> c [-11-])
- domineeriv minterm – [0101] (> [0001])

0	1	0	0
0	1	1	-
0	1	1	-
0	0	0	0

	a	b	c
0001	1	0	0
0101	1	1	0
0111	0	1	1
1101	0	1	0
1111	0	1	1



# Lihtimplikantide katte leidmine

lihtimplikantide tabel

abcd	A	B	C	D	E	F
0000			+	+		
0010			+	+		
<del>0100</del>			+		+	
<del>1000</del>				+		+
<del>0101</del>	+					+
<del>0110</del>			+			+
<del>1001</del>		+				+
<del>1010</del>				+		+
* 0111						*
* 1011						*
1101	+	+				

E, F

\* - olulised

abcd	A	B	C	D
0000			*	+
0010			*	+
1101	*	+		

A, C

	d		c	
b	1	0	0	1 <sup>C</sup>
a	1	1 <sup>A</sup>	1 <sup>E</sup>	1
	0	1	0	0
	1	1	1 <sup>F</sup>	1

$$f = b \bar{c} d + \bar{a} \bar{d} + \bar{a} b + a \bar{b}$$

A                      C                      E                      F



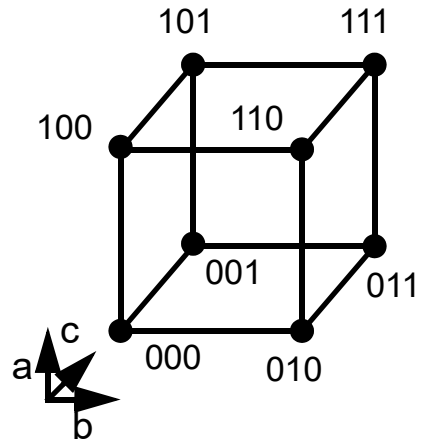
# Heuristiline minimeerimine

- **Täpne minimeerimine on kallis**
  - **Kõikvõimalike lihtimplikantide leidmine nõuab mälu ja aega**
- **Heuristiline minimeerimine**
  - **Väldib täpse minimeerimise kitsaskohti**
  - **“Mõistliku” suurusega liiasuseta katted**
  - **Kiirus ja rakendatav paljudes valdkondades**
  - **Lokaalne miinimumkate**
    - antud on esialgne kate
    - teisendus lihtkatteks
    - liiasuste eemaldamine
  - **Iteratiivne parendamine**
    - suurust parendatakse implikantide “modifitseerimise” teel
    - laiendus/kahandus otsustatakse naaberimplikantide põhjal

# Karnaugh kaart

		c		b	
	000	001	011	010	
a	100	101	111	110	

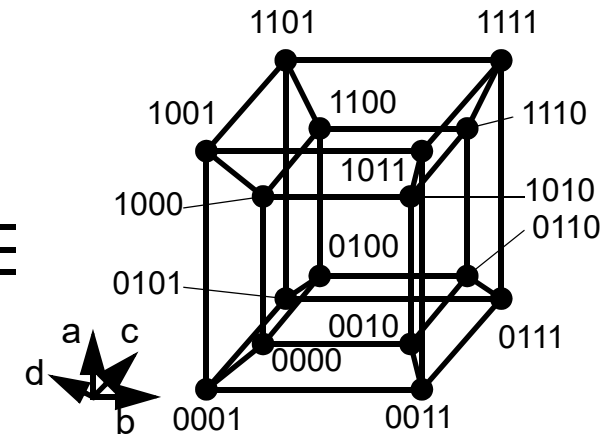
≡



		d		c	
	0000	0001	0011	0010	
b	0100	0101	0111	0110	
a	1000	1001	1011	1010	

		b		c	
	00	01	11	10	
a	0	000	001	011	010
	1	100	101	111	110

≡



		c		a	
	011	111	110	010	
b	001	101	100	000	

		d		b	
	0000	0001	0101	0100	
	0010	0011	0111	0110	
c	1010	1011	1111	1110	
a	1000	1001	1101	1100	

≡

## Karnaugh kaart – näide

$$f = \bar{a} \bar{d} + \bar{a} b + a \bar{b} + a \bar{c} d$$

$$f(a,b,c,d) = \Sigma(0,2,4,5,6,7,8,9,10,11,13)$$

		<u>d</u>		<u>c</u>			
		1	0	0	1		
b		1	1	1	1		
		0	1	0	0		
		a		1	1	1	1
				1	1	1	1

		<u>d</u>		<u>c</u>			
		1	0	0	1		
b		1	1	1	1		
		0	1	0	0		
		a		1	1	1	1
				1	1	1	1

		<u>d</u>		<u>c</u>			
		1	0	0	1		
b		1	1	1	1		
		0	1	0	0		
		a		1	1	1	1
				1	1	1	1

- Implikandi (kontuuri) laiendamine
- Vt. ka <http://tams-www.informatik.uni-hamburg.de/applets/kvd/kvd.html>



## Heuristilised meetodid – näide

$$f = \bar{a} \bar{d} + \bar{a} b + a \bar{b} + a \bar{c} d$$

$$f(a,b,c,d) = \Sigma(0,2,4,5,6,7,8,9,10,11,13)$$

		d		c	
		1	0	0	1
b		1	1	1	1
		0	1	0	0
a		1	1	1	1

		d		c	
		1	0	0	1
b		1	1	1	1
		0	1	0	0
a		1	1	1	1

		d		c	
		1	0	0	1
b		1	1	1	1
		0	1	0	0
a		1	1	1	1

$$f = \underline{b \bar{c} d} + \underline{\bar{b} \bar{d}} + \underline{\bar{a} b} + \underline{a \bar{b}}$$



## Nõrgalt määratud funktsioonide minimeerimine

- $|F_F| + |F_R| \ll |F_D|$
- Funktsiooni argumentide arv on suur
- $|F_F| + |F_R|$  on esitatud intervallidena, st. on olemas esialgne kate
- $F_F$ -i kuuluvaid intervale laiendatakse selliselt, et laiendus jääks  $F_D$  sisse
  - ükski 1-intervall ei tohi omada ühisosa ühegi 0-intervalliga (mittekattuvad)
  - *ortogonaalsusfunktsioon* – näitab, milliste argumentide järgi on intervallide paari teatud argumendi väärtus ühes intervallis 1, teises 0
  - kaks intervalli on mittekattuvad, kui nad on ortogonaalsed vähemalt ühe argumendi järgi
  - ortogonaalsed mitme argumendi järgi → osa argumente võib vabastada
- Näide: 000- # 1011 → 1010, seega võib esimese neist asendada kas 00-- või -00-'ga (eeldusel, säilib ortogonaalsus ka teiste 0-intervallidega)



## Näide #2

	a	b	c	d	e
1	1	1	-	1	-
	-	-	0	1	0
0	0	-	-	0	-
	-	0	0	-	1
	-	0	-	1	1
	0	-	-	1	1

	e		d	e		c	
	0	0	0	1	-	0	0
b	0	0	0	1	-	0	0
	-	-	1	1	1	1	-
a	-	0	0	1	-	0	-

11-1- # 0--0- = 10010  
 11-1- # -00-1 = 01000  
 11-1- # -0-11 = 01000  
 11-1- # 0--11 = 10000

~~-1-1- # 0--0- = 00010  
 -1-1- # -00-1 = 01000  
 -1-1- # -0-11 = 01000  
 -1-1- # 0--11 = 00000~~

11--- # 0--0- = 10000  
 11--- # -00-1 = 01000  
 11--- # -0-11 = 01000  
 11--- # 0--11 = 10000

# Heuristiliste minimeerimiste põhioperaatorid

## Laiendus (Expand)

- implikantide teisendus lihtimplikantideks
- kaetud implikantide eemaldamine

## Kitsendus (Reduce)

- implikantide suuruse vähendamine, hoides katte korrektse

## Ümberkujundus (Reshape)

- implikantide paaride muutmine suurendades üht ja vähendades teist

## Liasusetus (Irredundant)

- liiasuse eemaldamine kattest

ülesanne

0000	1
0010	1
0100	1
0101	1
0110	1
0111	1
1000	1
1001	1
1010	1
1011	1
1101	1

 kõik liht-  
implikandid

0--0	a
-0-0	b
01--	c
10--	d
1-01	e
-101	f

Näide

1	0	0	1
1	1	1	1
0	1	0	0
1	1	1	1

 võimalikud  
lahendused

{a,c,d,e}

{b,c,d,e}



# Näide – laiendus

0000	<i>exp</i>
0010	<i>x</i>
0100	<i>x</i>
0101	
0110	<i>x</i>
0111	
1000	
1001	
1010	
1011	
1101	

0000  
 ↓  
 0--0

0--0  
 katab  
 0010  
 0100  
 0110

0--0	<i>a</i>
0101	<i>exp</i>
0111	<i>x</i>
1000	
1001	
1010	
1011	
1101	

0101  
 ↓  
 01--

01--  
 katab  
 [0100]  
 [0110]  
 0111

0--0	<i>a</i>
01--	<i>c</i>
1000	<i>exp</i>
1001	
1010	<i>x</i>
1011	
1101	

1	0	0	1
1	1	1	1
0	1	0	0
1	1	1	1

## Näide – laiendus – kõik sammud

0000	<i>exp</i>
0010	<i>x</i>
0100	<i>x</i>
0101	
0110	<i>x</i>
0111	
1000	
1001	
1010	
1011	
1101	

0--0	<i>a</i>
0101	<i>exp</i>
0111	<i>x</i>
1000	
1001	
1010	
1011	
1101	

0--0	<i>a</i>
01--	<i>c</i>
1000	<i>exp</i>
1001	
1010	<i>x</i>
1011	
1101	

0--0	<i>a</i>
01--	<i>c</i>
-0-0	<i>b</i>
1001	<i>exp</i>
1011	<i>x</i>
1101	

0--0	<i>a</i>
01--	<i>c</i>
-0-0	<i>b</i>
10--	<i>d</i>
1101	<i>exp</i>

0--0	<i>a</i>
01--	<i>c</i>
-0-0	<i>b</i>
10--	<i>d</i>
1-01	<i>e</i>

{a,b,c,d,e}

1	0	0	1
1	1	1	1
0	1	0	0
1	1	1	1



## Näide – kitsendus

0--0	xxxx
01--	c
-0-0	b
10--	d
1-01	e

0--0  
 ↓  
 00-0  
 ↓  
 0000  
  
 -0-0  
 katab  
~~0000~~

01--	c
-0-0	00-0
10--	d
1-01	e

Kaetuse analüüs:

-0-0 & 0000 = 0000 - katab

Võrdluseks:

-0-0 & 10-- = 10-0 - ei kata

1	0	0	1
1	1	1	1
0	1	0	0
1	1	1	1



## Näide – kitsendus – kõik sammud

0--0	xxxx
01--	c
-0-0	b
10--	d
1-01	e

01--	c
-0-0	00-0
10--	d
1-01	e

01--	c
00-0	b'
10--	d
1-01	1101

01--	c
00-0	b'
10--	d
1101	e'

{ b',c,d,e' }

1	0	0	1
1	1	1	1
0	1	0	0
1	1	1	1



## Näide – ümberkujundus

01--	01-1		01-1	c'
00-0	0--0	→	0--0	a
10--	d		10--	d
1101	e'		1101	e'

{ b',c,d,e' }

1	0	0	1
1	1	1	1
0	1	0	0
1	1	1	1



1	0	0	1
1	1	1	1
0	1	0	0
1	1	1	1

{ a,c',d,e' }



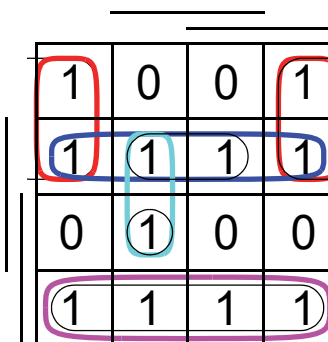
## Näide – laiendus #2

01-1	<i>exp</i>
0--0	<i>a</i>
10--	<i>d</i>
1101	<i>e'</i>

01--	<i>c</i>
0--0	<i>a</i>
10--	<i>d</i>
1101	<i>exp</i>

01--	<i>c</i>
0--0	<i>a</i>
10--	<i>d</i>
-101	<i>f</i>

{ a,c,d,f }





# Kokkuvõte

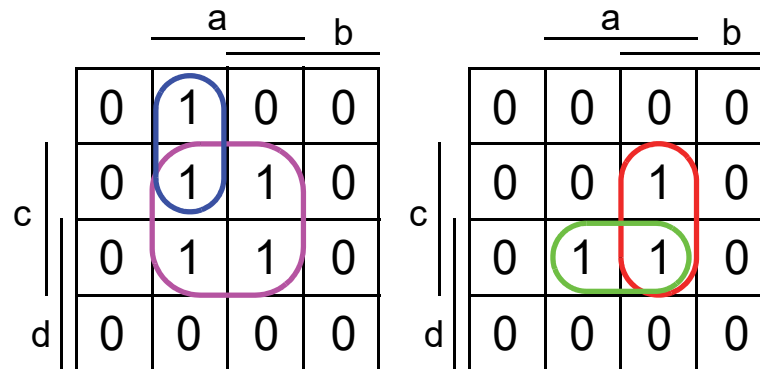
- **MINI**
  - Laiendus: kate – {a,b,c,d,e} – lihtkate, liiasusega (kuid ükski implikant ei sisaldu mõnes teises)
  - Kitsendus: a eemaldatakse;  $b [-0-0] \rightarrow b' [00-0]$ ;  $e [1-01] \rightarrow e' [1101]$ ; kate – (b',c,d,e')
  - Ümberkujundus:  $\{b',c\} [00-0][01--] \rightarrow \{a,c'\} [0--0][01-1]$
  - Laiendus #2: kate – {a,c,d,f}; lihtkate, liiasuseta
- **Intuitiivne laiendus**
  - iga implikandi puhul asenda '0' või '1' võimaluse korral '-'
  - eemalda kõik kaetud implikandid
  - probleemid – õigsuse kontroll & implikantide järjekord
- **Õigsuse kontroll**
  - Espresso, MINI – kontrollitakse laiendatud implikandi *ühisosa* kõigi 0-implikantidega ( $F_R$ ), täienduse leidmine vajalik
  - Presto – kontrollitakse laiendatud implikandi sisaldumist 1- ja \*-implikantide ühendis ( $F_F \cup F_D$ ), taandub nn. rekursiivsele tautoloogia kontrollile



- **Laiendus – heuristilised võtted**
  - Laiendada tuleks esimesena need intervallid, millede katmine teiste poolt on vähetõenäoline
  - Kaalutud intervallid – mida suurem kaal, seda väiksem on võimalik kaetavus (“hõredalt asustatud ümbruskond”)
- **Kitsendus – heuristilised võtted**
  - Kaalutud intervallid – mida väiksem kaal, seda suuremad võimalised kitsendamiseks (“tihedalt asustatud ümbruskond”)
- **Liiasuse eemaldamine**
  - Oluliste intervallide kindlaks tegemine
  - Katte probleem lahendatakse heuristiliselt
- **Espresso**
  - Täiendi leidmine
  - Oluliste intervallide/mintermide määramine (pärast laiendamist ja liiasuse eemaldamist)
  - Iteratsioon – laiendus, liiasusetus, kitsendus
  - Kaalufunktsioonid – katte võimsus & intervallide ja literaalide arvu kaalutud summa

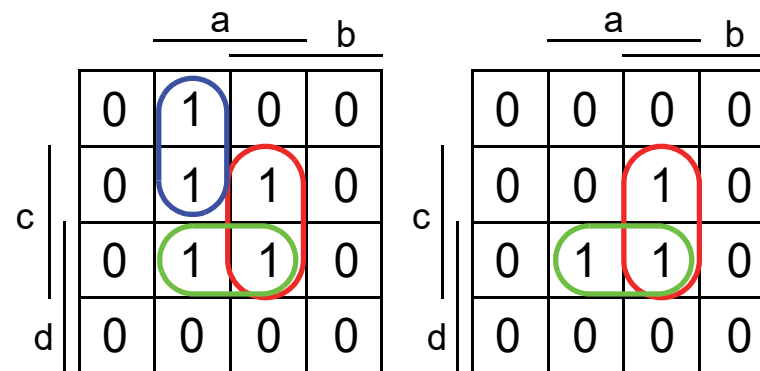
# Funktsioonide süsteemi minimeerimine

abcd	xy
10-0	10
1-1-	10
1-11	01
111-	01



funktsioonid eraldi

abcd	xy
10-0	10
1-11	11
111-	11



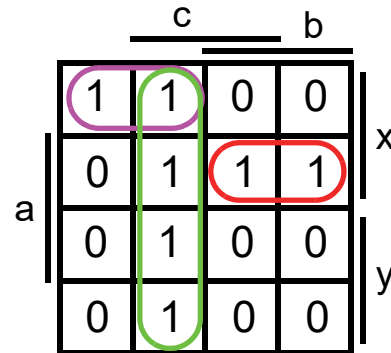
funktsioonid korruga

# Funktsioonide süsteemi minimeerimine

- Väljundite hulka vaadeldakse kui täiendavat mitmevalentset sisendit
- Implikantide leidmisel rakendatakse samu operatsioone

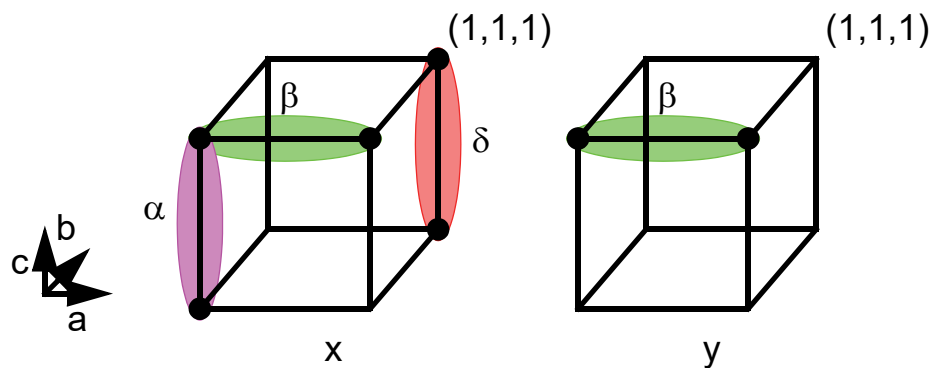
abc	xy
000	10
001	11
101	11
110	10
111	10

abc0	
0001	1
001-	1
101-	1
1101	1
1111	1



abc0	
00-1	1
-01-	1
11-1	1

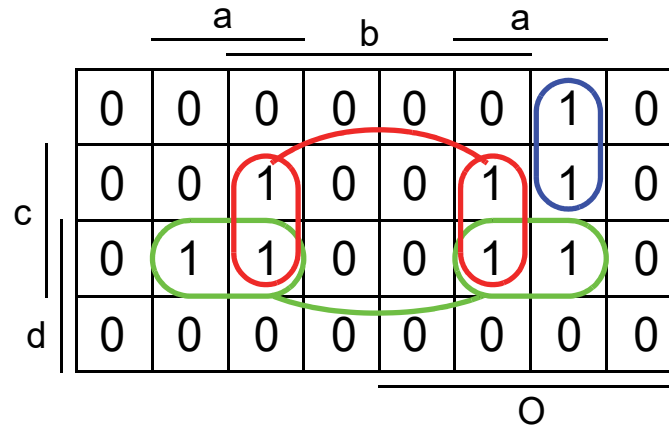
abc	xy
00-	10
-01	11
11-	10



# Funktsioonide süsteemi minimeerimine

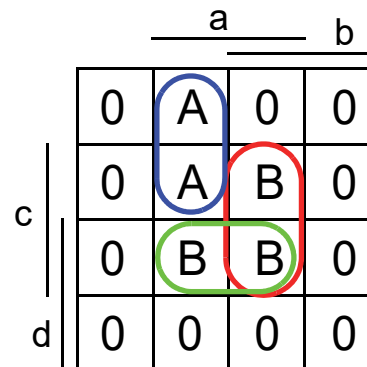
abcdO	z
10-01	1
1-1-1	1
1-110	1
111-0	1

mitme-valentse loogika abil



abcdO	z
10-01	1
1-11-	1
111--	1

sümbol-kodeeringu abil



abcd	xy
10-0	10
1-11	11
111-	11



## Mitme-valentne loogika (Multiple-Valued Logic - MVL)

- **Post'i algebra**
  - Boole'i algebra üldistus
  - Kasutatakse matemaatilise baasina MVL loogikalülide projekteerimisel
  - Post, 1921. a. – esimene mitmeväärtuseline (mitmevalentne) loogika
- **Kahendloogika –  $(B, *, +, \sim), B = \{0, 1\}$** 
  - täielikult määratud funktsioonid –  $f: B^n \rightarrow B$  ja  $f: B^n \rightarrow B^m$
  - mittetäielikult määratud funktsioonid –  $f: B^n \rightarrow \{0, 1, -\}^m$  (ka  $f: B^n \rightarrow \{0, 1, *\}^m$ )
  - AND, OR, NOT – loogikafunktsioonide täielik süsteem
- **MV-loogika –  $(\{P_i\}, \text{MIN}, \text{MAX}, \text{literal}), P_i = \{0, 1, \dots, m_i - 1\}$** 
  - mittetäielikult määratud funktsioonid –  $f: P_1 \times P_2 \times \dots \times P_n \rightarrow P_m$   
või ka  $f: P_1 \times P_2 \times \dots \times P_n \rightarrow \{0, 1, -\}^m$
  - MIN, MAX, literal - MVL-funktsioonide täielik süsteem

## Mitme-valentne loogika – operatsioonid

- **MIN(x, y)** – minimaalne x ja y väärtus [ $\cdot$ ]
  - vrdl. AND kahendloogikas
- **MAX(x, y)** – maksimaalne x ja y väärtus [ $+$ ]
  - vrdl. OR kahendloogikas
- **literaal (literal)** – unaarne operatsioon –  $x_i^{\{c_i\}} = m_i - 1$ , kui  $x_i = c_i$ , muidu 0
  - tähistus –  $x_1^{\{2\}} \equiv x_1^2$  ja  $x_1^{\{2\}} \equiv \bar{x}_1$
- **hulkliteaal (set literal)** –  $x_i^{\{S\}} = m_i - 1$ , kui  $x_i \in S$ , muidu 0
  - tähistus  $x_3^{\{0,2\}} \equiv x_3^{\{0,2\}} \equiv x_3^{0,2}$
  - vrdl. kahendloogikaga –  $x_i^{\{0\}} \equiv \bar{x}_i$ ,  $x_i^{\{1\}} \equiv x_i$ ,  $x_i^{\{0,1\}} \equiv -$  (don't-care)
- **Shannoni arendus**
  - $f(x) = \bar{x}f_{\bar{x}}() + xf_x()$  /  $f(x) = x^0f_{x^0}() + x^1f_{x^1}() + \dots + x^{m-1}f_{x^{m-1}}()$

# Esitusviisid

## Tõeväärtustabel

$x_1$	$x_2$	$f$	
0	0	0	$0 \cdot x_1^{\{0\}} \cdot x_2^{\{0\}}$
0	1	0	$0 \cdot x_1^{\{0\}} \cdot x_2^{\{1\}}$
0	2	2	$2 \cdot x_1^{\{0\}} \cdot x_2^{\{2\}}$
1	0	1	$1 \cdot x_1^{\{1\}} \cdot x_2^{\{0\}}$
1	1	1	$1 \cdot x_1^{\{1\}} \cdot x_2^{\{1\}}$
1	2	0	$0 \cdot x_1^{\{1\}} \cdot x_2^{\{2\}}$
2	0	0	$0 \cdot x_1^{\{2\}} \cdot x_2^{\{0\}}$
2	1	0	$0 \cdot x_1^{\{2\}} \cdot x_2^{\{1\}}$
2	2	2	$2 \cdot x_1^{\{2\}} \cdot x_2^{\{2\}}$

" · " – MIN    "+" – MAX     $x^{\{i\}}$  –  $x$ 'i literal

## Avaldis

$$f(x_1, x_2) = 1x_1^{\{1\}}x_2^{\{0\}} + 1x_1^{\{1\}}x_2^{\{1\}} + 2x_1^{\{0\}}x_2^{\{2\}} + 2x_1^{\{2\}}x_2^{\{2\}}$$

$$f(x_1, x_2) = 1x_1^{\{1\}}x_2^{\{0,1\}} + 2x_1^{\{0,2\}}x_2^{\{2\}}$$

## Karnaugh kaart

$x_1 \backslash x_2$	0	1	2
0	0	1	0
1	0	1	0
2	2	0	2

$x_1 \backslash x_2$	0	1	2
0	0	1	0
1	0	1	0
2	2	0	2



## MV-funktsioonide minimeerimine

- Ei midagi uut!
- Antud funktsiooni  $F$  ühtede ( $f$ ) ja määramata ( $d$ ) (ja nullide ( $r$ )) piirkondade kate
- Leida minimaalne korrutiste-summa kuju funktsioonile  $F$
  
- Genereerida  $f+d$  lihtimplikandid
- Luua implikantide tabel
- Lahendada katte probleem
  - Algoritmid erinevad ainult pisiasjades

## Mitme väljundiga funktsioonid

- $n$ -muutjaga ja  $k$ -väljundiga kahendfunktsioonide süsteem teisendatakse  $n+1$ -muutujaga ja 1-väljundiga funktsiooniks, üks sisendmuutujaist on MV:  
 $\{0,1\}^n \rightarrow \{0,1\}^k \equiv \{0,1\}^n \times \{0, 1, \dots, m-1\} \rightarrow \{0,1\}$
- Hong'i teoreem
  - iga  $n$ -muutuja implikant pluss vastavad väljundid moodustavad ühe implikandi  $n+1$ -ruumis
  - väljundite arv määrab täiendava sisendi valentside arvu
  - implikandi määratud väljundid moodustavad hulk-literaali täiendavas sisendis



## Lihtimplikantide leidmine

- **Sarnane üksiku funktsiooni implikantide leidmisega**
  - erinevus täpselt ühes järgus (vt. kleepimisseadused)
    - võib vaadelda sulgude ette toomisega
  - praktikas tasub eristada, kas erinevus on kahend- või MV-sisendis
- **Erinevus ühes kahendsisendis**
  - täpselt üks kahendsisend on erinev – ühes 0 ja teises 1 (ning MV-osad identsed)
  - kleepuvad:  $a^0b^0c^0e^0 + a^0b^1c^0e^0 = a^0c^0e^0 (b^0+b^1) = a^0b^{\{0,1\}}c^0e^0$
- **Erinevus MV-sisendis**
  - kõik kahendsisendid on identsed
  - kleepuvad:  $a^0b^1c^1e^0 + a^0b^1c^1e^1 = a^0b^1c^1 (e^0+e^1) = a^0b^1c^1e^{\{0,1\}}$
- **Vektoriesitus – kahendosa ‘0’, ‘1’ ja ‘-’; MV-osa positsioonilise kodeeringuga**
  - $a^0b^0c^0e^0 + a^0b^1c^0e^0$  : 000 100 + 010 100 => 0-0 100
  - $a^0b^1c^1e^0 + a^0b^1c^1e^1$  : 011 100 + 011 010 => 011 110



# Näide

abc	xyz
000	111
001	011
010	101
011	110
100	000
101	010
110	000
111	101

1	0	1	1
0	0	1	0

1	1	1	0
0	1	0	0

1	1	0	1
0	0	1	0

abc	xyz
0-0	101
-11	100
00-	011
0-1	010
-01	010
111	001





## Näide (järg)

abc	xyz
000	111
001	011
010	101
011	110
100	000
101	010
110	000
111	101

$$x(a,b,c) = \bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc$$

$$y(a,b,c) = \bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}bc + a\bar{b}\bar{c}$$

$$z(a,b,c) = \bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}bc + abc$$

$$f: \{0,1\} \times \{0,1\} \times \{0,1\} \times \{0,1,2\} \rightarrow \{0,1\}$$

$$\begin{aligned} o(a,b,c,e) = & a^0b^0c^0e^0 + a^0b^1c^0e^0 + a^0b^1c^1e^0 + \\ & + a^1b^1c^1e^0 + a^0b^0c^0e^1 + a^0b^0c^1e^1 + \\ & + a^0b^1c^1e^1 + a^1b^0c^1e^1 + a^0b^0c^0e^2 + \\ & + a^0b^0c^1e^2 + a^0b^1c^0e^2 + a^1b^1c^1e^2 \end{aligned}$$

abc	e	o
000	100	1
010	100	1
011	100	1
111	100	1
000	010	1
001	010	1
011	010	1
101	010	1
000	001	1
001	001	1
010	001	1
111	001	1



## Näide (järg)

abc	e	o	
000	100	1	1
010	100	1	2
011	100	1	3
111	100	1	4
000	010	1	5
001	010	1	6
011	010	1	7
101	010	1	8
000	001	1	9
001	001	1	10
010	001	1	11
111	001	1	12

$$\begin{array}{cc} 3. & 7. \\ a^0b^1c^1e^0 + a^0b^1c^1e^1 = a^0b^1c^1e^{\{0,1\}} \end{array}$$

$$\begin{array}{cccc} 1. & 2. & 9. & 11. \\ a^0b^0c^0e^0 + a^0b^1c^0e^0 + a^0b^0c^0e^2 + a^0b^1c^0e^2 = \dots \\ \dots = a^0b^{\{0,1\}}c^0e^0 + a^0b^{\{0,1\}}c^0e^2 = a^0b^{\{0,1\}}c^0e^{\{0,2\}} = a^0c^0e^{\{0,2\}} \end{array}$$

$$\begin{array}{cccc} 5. & 9. & 6. & 10. \\ a^0b^0c^0e^1 + a^0b^0c^0e^2 + a^0b^0c^1e^1 + a^0b^0c^1e^2 = \dots \\ \dots = a^0b^0c^0e^{\{1,2\}} + a^0b^0c^1e^{\{1,2\}} = a^0b^0c^{\{0,1\}}e^{\{1,2\}} = a^0b^0e^{\{1,2\}} \end{array}$$

# Näide – minimeerimine

1.

2.

9.

11.

$$a^0b^0c^0e^0 + a^0b^1c^0e^0 + a^0b^0c^0e^2 + a^0b^1c^0e^2 = \dots$$

$$\dots = a^0b^{\{0,1\}}c^0e^0 + a^0b^{\{0,1\}}c^0e^2 = a^0b^{\{0,1\}}c^0e^{\{0,2\}} = \underline{a^0c^0e^{\{0,2\}}}$$

abc	e	o	
000	100	1	<b>1</b>
010	100	1	<b>2</b>
011	100	1	<b>3</b>
111	100	1	<b>4</b>
000	010	1	<b>5</b>
001	010	1	<b>6</b>
011	010	1	<b>7</b>
101	010	1	<b>8</b>
000	001	1	<b>9</b>
001	001	1	<b>10</b>
010	001	1	<b>11</b>
111	001	1	<b>12</b>

mintermid

1. etapp

2. etapp

gr.	abc	e
0	000	100
	000	010
	000	001
1	010	100
	001	010
	001	001
	010	001
2	011	100
	011	010
	101	010
3	111	100
	111	001

gr.	abc	e
0	000	110
	000	101
	000	011
0-0	100	100
00-	010	010
00-	001	001
0-0	001	001
1	010	101
	001	011

gr.	abc	e
0	000	111
0-0	101	

	c		b
a	1	0	1
	0	0	1
	1	1	1
	0	1	0
	1	1	0
	0	0	1



# Näide – minimeerimine

abc	e	o	
000	100	1	<b>1</b>
010	100	1	<b>2</b>
011	100	1	<b>3</b>
111	100	1	<b>4</b>
000	010	1	<b>5</b>
001	010	1	<b>6</b>
011	010	1	<b>7</b>
101	010	1	<b>8</b>
000	001	1	<b>9</b>
001	001	1	<b>10</b>
010	001	1	<b>11</b>
111	001	1	<b>12</b>

mintermid

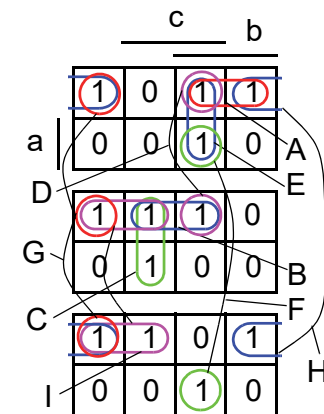
gr.	abc	e	
0	000	100	*
	000	010	*
	000	001	*
1	010	100	*
	001	010	*
	001	001	*
	010	001	*
2	011	100	*
	011	010	*
	101	010	*
3	111	100	*
	111	001	*

1. etapp

gr.	abc	e	
0	000	110	*
	000	101	*
	000	011	*
	0-0	100	*
	00-	010	*
	00-	001	*
	0-0	001	*
1	010	101	*
	001	011	*
	01-	100	<b>A</b>
	0-1	010	<b>B</b>
	-01	010	<b>C</b>
2	011	110	<b>D</b>
	-11	100	<b>E</b>
3	111	101	<b>F</b>

2. etapp

gr.	abc	e	
0	000	111	<b>G</b>
	0-0	101	<b>H</b>
	00-	011	<b>I</b>





# Näide – minimeerimine

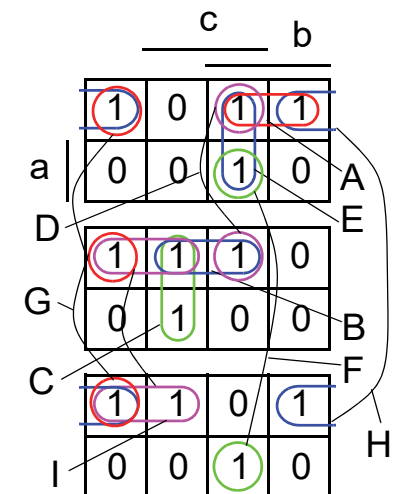
lihtimplikantide tabel

abc	e	A	B	C	D	E	F	G	H	I
<del>000</del>	<del>100</del>									
<del>000</del>	<del>010</del>									
<del>000</del>	<del>001</del>									
<del>010</del>	<del>100</del>									
<del>001</del>	<del>010</del>									
*	001									*
*	010								*	
	011	+			+	+				
	011		+		+					
*	101			*						
<del>111</del>	<del>100</del>									
*	111									*

abc	e	A	B	D	E	G
011	100	+		*	+	
011	010		+	*		

abc	e	o
011	110	1
111	101	1
0-0	101	1
00-	011	1
-01	010	1

D  
F  
H  
I  
C





## Näide (pärast minimeerimist)

$$\begin{aligned}
 o(a,b,c,e) = & a^0b^0c^0e^0 + a^0b^1c^0e^0 + a^0b^1c^1e^0 + \\
 & + a^1b^1c^1e^0 + a^0b^0c^0e^1 + a^0b^0c^1e^1 + \\
 & + a^0b^1c^1e^1 + a^1b^0c^1e^1 + a^0b^0c^0e^2 + \\
 & + a^0b^0c^1e^2 + a^0b^1c^0e^2 + a^1b^1c^1e^2
 \end{aligned}$$

Lihtimplikandid & liiasuseta

$$\begin{aligned}
 o(a,b,c,e) = & a^0b^1c^1e^{\{0,1\}} + a^1b^1c^1e^{\{0,2\}} + \\
 & + a^0b^{\{0,1\}}c^0e^{\{0,2\}} + a^0b^0c^{\{0,1\}}e^{\{1,2\}} + \\
 & + a^{\{0,1\}}b^0c^1e^1
 \end{aligned}$$

$$\begin{aligned}
 o(a,b,c,e) = & a^0b^1c^1e^{\{0,1\}} + a^1b^1c^1e^{\{0,2\}} + \\
 & + a^0c^0e^{\{0,2\}} + a^0b^0e^{\{1,2\}} + b^0c^1e^1
 \end{aligned}$$

abc	e	o
011	110	1
111	101	1
0-0	101	1
00-	011	1
-01	010	1

## Näide (minimeeritud kahendfunktsioonid)

$$o(a,b,c,e) = a^0 b^1 c^1 e^{\{0,1\}} + a^1 b^1 c^1 e^{\{0,2\}} + a^0 c^0 e^{\{0,2\}} + a^0 b^0 e^{\{1,2\}} + b^0 c^1 e^1$$

abc	e	o
011	110	1
111	101	1
0-0	101	1
00-	011	1
-01	010	1

abc	xyz
011	110
111	101
0-0	101
00-	011
-01	010

1	0	1	1
0	0	1	0

1	1	1	0
0	1	0	0

1	1	0	1
0	0	1	0

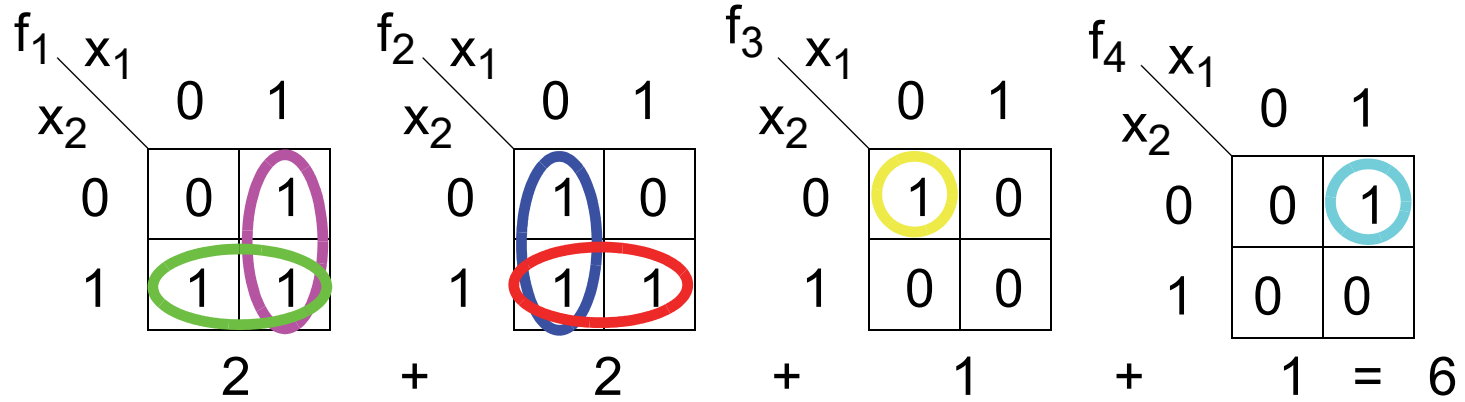
$$x(a,b,c) = \bar{a}bc + abc + \bar{a}\bar{c}$$

$$y(a,b,c) = \bar{a}bc + \bar{a}\bar{b} + \bar{b}c$$

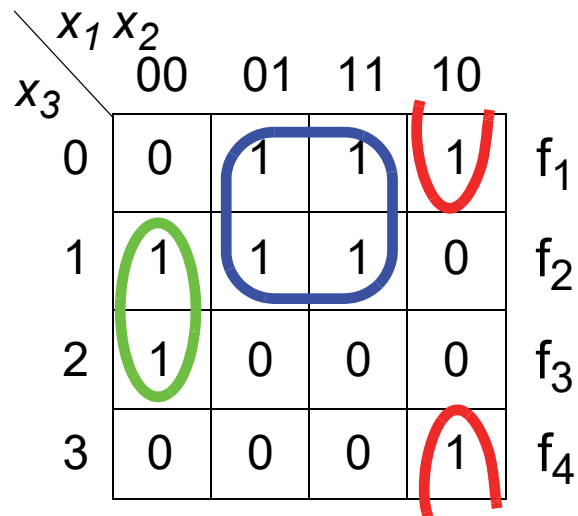
$$z(a,b,c) = abc + \bar{a}\bar{c} + \bar{a}\bar{b}$$



# Näide #2



implikanti



3 implikanti



## Näide #2 (minimeeritud)

$x_2x_1$	$f_1f_2f_3f_4$
0 0	0 1 1 0
0 1	1 0 0 1
1 0	1 1 0 0
1 1	1 1 0 0

mintermid

gr.	21	1234	
0	00	0100	*
	00	0010	*
1	01	1000	*
	01	0001	*
	10	1000	*
	10	0100	*
2	11	1000	*
	11	0100	*

1. etapp

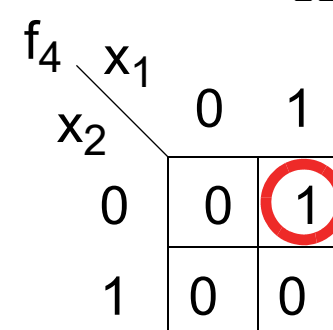
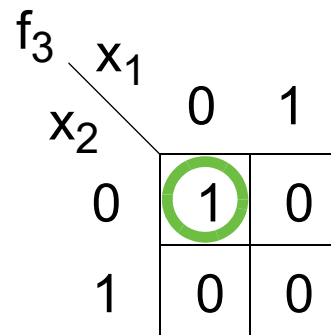
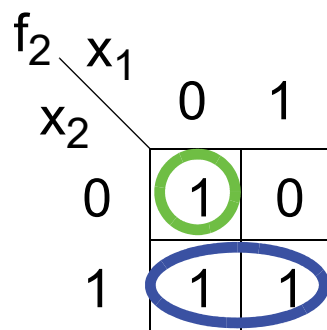
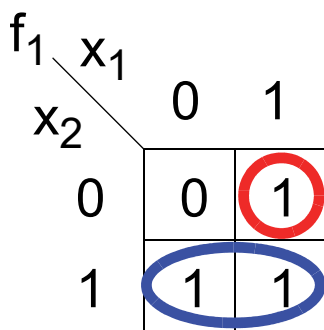
gr.	21	1234	
0	00	0110	A
	-0	0100	B
1	01	1001	C
	10	1100	*
	-1	1000	D
	1-	1000	*
	1-	0100	*
2	11	1100	*

2. etapp

gr.	21	1234	
1	1-	1100	E

lihtimplikantide tabel

21	1234	A	B	C	D	E
<del>00</del>	<del>0100</del>					
*	00 0010	*				
<del>01</del>	<del>1000</del>					
*	01 0001			*		
*	10 1000					*
<del>10</del>	<del>0100</del>					
<del>11</del>	<del>1000</del>					
*	11 0100					*



3 implikanti



## Näide #3

abc	xy
000	10
001	11
101	11
110	10
111	10

abc e	
000 10	1
001 11	1
101 11	1
110 10	1
111 10	1

mintermid

gr.	abc e	*
0	000 10	*
1	001 10	*
	001 01	*
2	101 10	*
	101 01	*
	110 10	*
3	111 10	*

1. etapp

gr.	abc e	
0	00- 10	A
1	001 11	*
	-01 10	*
	-01 01	*
2	101 11	*
	1-1 10	B
	11- 10	C

2. etapp

gr.	abc e	
1	-01 11	D

1	1	0	0
0	1	1	1

0	1	0	0
0	1	0	0

lihtimplikantide tabel

abc e	A	B	C	D
* 000 10	*			
<del>001 10</del>		+		+
* 001 01				*
<del>101 10</del>		+		+
* 101 01				*
* 110 10			*	
<del>111 10</del>		+	+	

abc	e	o	
00-	10	1	A
11-	10	1	C
-01	11	1	D

abc	xy
00-	10
11-	10
-01	11



## Näide #4 – määramatused

abc	xyz
000	110
001	0-1
010	-0-
011	010
100	1-0
101	01-
110	101
111	-00

mintermid

```

gr. abc e
0 000 100 *
   000 010 *
1 *001 010 *
   001 001 *
   *010 100 *
   *010 001 *
   100 100 *
   *100 010 *
2 011 010 *
   101 010 *
   *101 001 *
   110 100 *
   110 001 *
3 *111 100 *

```

1. etapp

```

gr. abc e
0 000 110 *
   0-0 100 *
   -00 100 *
   00- 010 *
   -00 010 *
1 001 011 *
   *010 101 *
   100 110 *
   0-1 010 A
   -01 010 *
   -01 001 *
   -10 100 *
   -10 001 *
   1-0 100 *
   10- 010 *
2 101 011 *
   110 101 *
   11- 100 B

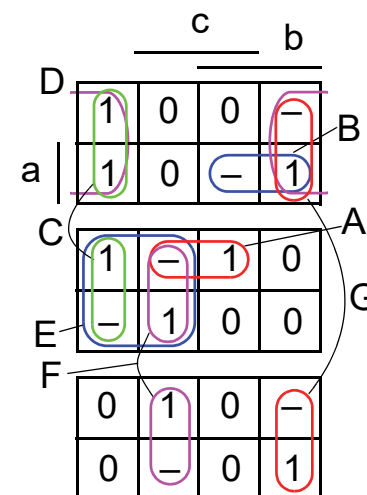
```

2. etapp

```

gr. abc e
0 -00 110 C
   --0 100 D
   -0- 010 E
1 -01 011 F
   -10 101 G

```



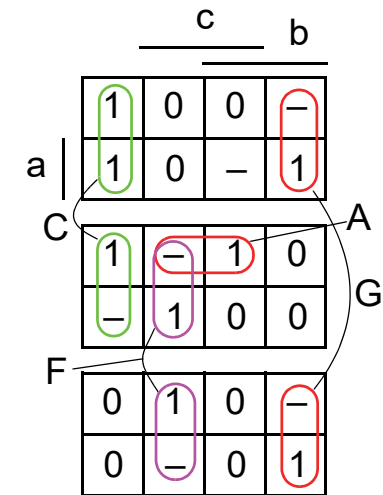


# Näide #4 – määramatused

lihtimplikandidid & tabel

0-1 010 **A**  
 11- 100 **B**  
 -00 110 **C**  
 --0 100 **D**  
 -0- 010 **E**  
 -01 011 **F**  
 -10 101 **G**

abc	e	A	B	C	D	E	F	G
000	100			+	+			
000	010			+		+		
001	001						+	
011	010	+						
100	100			+	+			
101	010					+	+	
110	100		+	+				+
110	001							+



abc	e	A	B	C	D	E	F	G
000	100			+	+			
000	010			+		+		
* 001	001							*
* 011	010	*						
100	100			+	+			
<del>101</del>	<del>010</del>							
<del>110</del>	<del>100</del>							
* 110	001							*

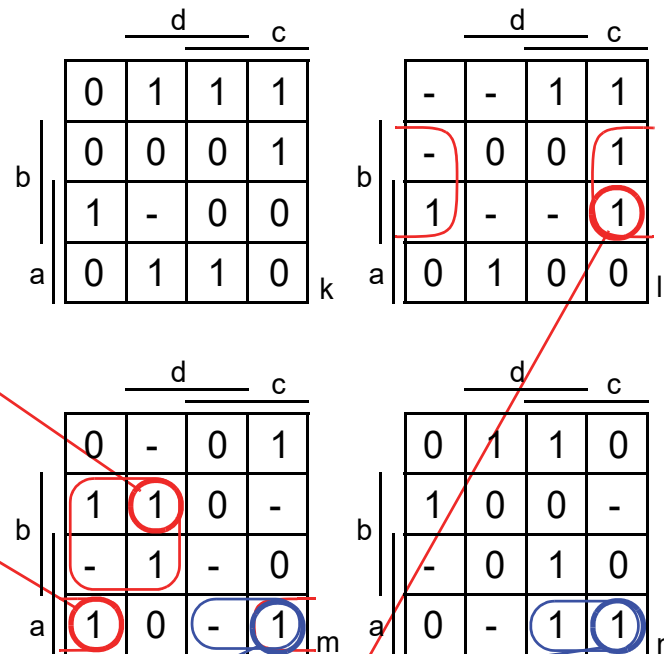
abc	e	B	C	D	E
000	100	+	+		
000	010	+			+
100	100	+	+		

abc	xyz	
0-0	010	<b>A</b>
-00	110	<b>C</b>
-01	011	<b>F</b>
-10	101	<b>G</b>

## Näide #5 – heuristiline minimeerimine

*lähteülesanne...*

abcd	klmn
0000	0-00
0001	1--1
0010	1110
0011	1101
0100	0-11
0101	0010
0110	11--
0111	0000
1000	0010
1001	110-
1010	0011
1011	10-1
1100	11--
1101	--10
1110	0100
1111	0--1



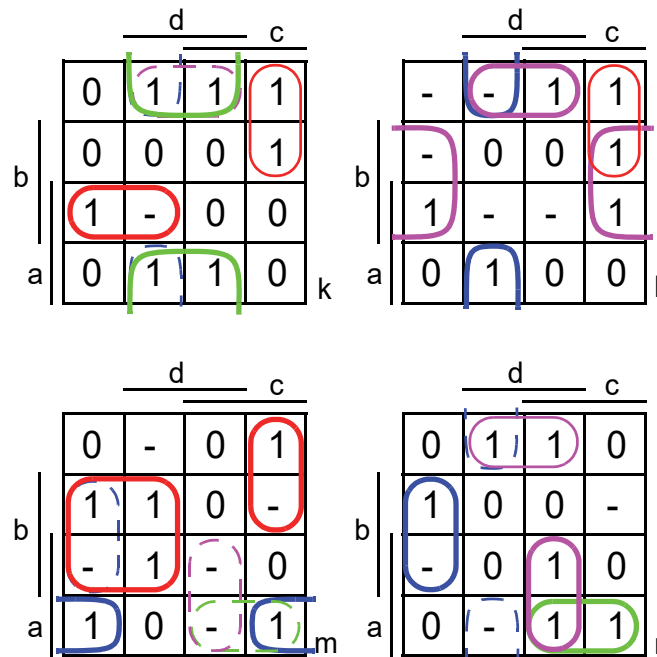
Millest alustada?

Vrdl. olulised lihtimplikandid →  
igal juhul peab olema kaetud →  
“üksikud ühed”, siis “paarid”, ...

- 1) “üksikud ühed” –  
kõige suurem kontuur  
vastava väljundi katmiseks, mis  
kataks võimalikult palju 1-d
- 2) “paarid” –  
kõige suurem kontuur  
mõlema väljundi katmiseks

## Näide #5 – heuristiline minimeerimine (järg)

abcd	klmn
-10-	0010
10-0	0010
-1-0	0100
101-	000 <u>1</u>
<del>0-10</del>	<del>1110</del>
-001	0 <u>1</u> 00
00-1	0 <u>1</u> 01
-0-1	1000
110-	1000
-100	0001
1-11	0001



Edasi need, mis veel katmata...

Jälle alustada sealt, kus vähe ühtesi alles...

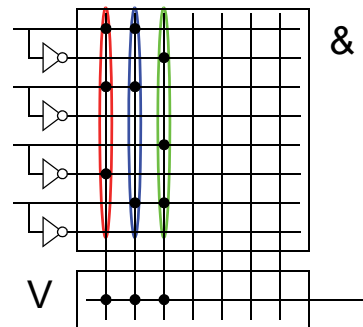
Eemaldada need, mis juba kaetud (punktirjoonega)

Variante on rohkemgi...

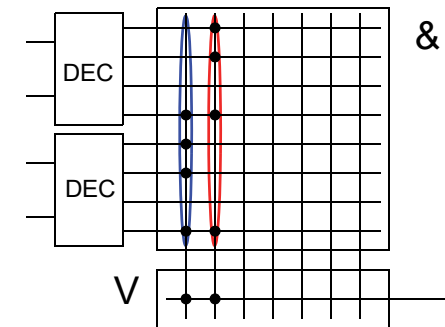
## MVL rakendusi

- Efektivsem kahendloogika probleemide lahendamine
  - kahendfunktsioonide süsteem (mitu väljundit) – väljundeid vaadeldakse kui ühte täiendavat MV sisendit
  - sisendite, väljundite ja olekute kodeerimine (optimeerimine)
  - dekodeeriga PLM – sisendite paari vaadeldakse kui üht 4-valentset sisendit
  - testimine – kolmas väärtus kasutusel vea tähistamiseks

$x_1 x_2$	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	1	1	1	0
10	0	0	0	0



$y_1$	0	1	3	2
0	0	0	1	0
1	0	0	1	0
3	1	1	1	0
2	0	0	0	0



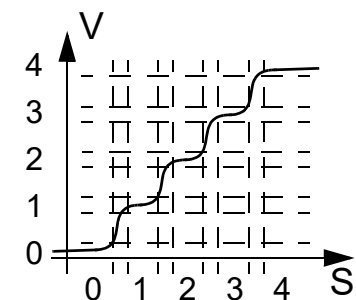
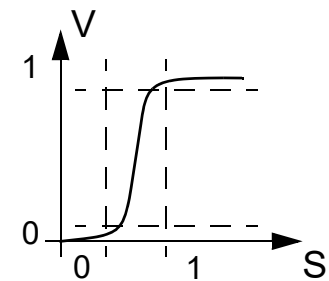
- MV riistvara – rohkem kui kaks signaalinivood
  - rohkem kui kaks diskreetset signaalinivood (pinge või vool)
  - võimalik kasutada olemasolevaid tehnoloogiaid – CMOS jne.

## MVL eelised riistvaras

- Tüüpiline VLSI mikroskeem
  - ~70% pindalast ühendused, ~20% isolatsioon ja ainult ~10% transistorid
- MVL süsteemid
  - traadid kannavad rohkem informatsiooni – kokkuhoid traatide arvus ja nende vahelises isolatsioonis
  - väljaviigud kannavad rohkem informatsiooni – väiksem väljaviikude arv korpuse kohta
- MVL võimaldab kiireid aritmeetikaoperatsioone
  - nt. kolmendaritmeetika

## MVL mälud

- 4-valentsed mälud (flash, DRAM)
  - kahekordne salvestustihedus (transistori kohta)
- Mäluelementide põhiprobleemid
  - salvestamine & lugemine
  - töökindlus - vajalikud kindlad vahed eri nivoode vahel
  - nivoo taastamine registrites





# Kahetasemeline minimeerimine ja mitmetasemeline realisatsioon

3 varianti

1: F, A, B

2: F, A, D

3: F, B, E

Lisaks

üksikult

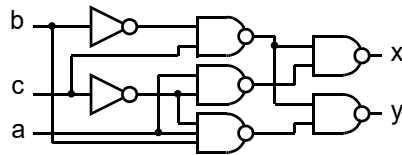
minimeeritud

4: A, E, D, F

$bi=b'$ ;  $ci=c'$ ;  $t1=bi \cdot c$ ;

$t2=a \cdot ci$ ;  $t3=a \cdot b \cdot ci$ ;

$x=t1+t2$ ;  $y=t1+t3$ ;



abc	xy
-01	11
1-0	10
110	<u>01</u>

	c	b	
-	1	0	0
a	1	1	0
	0	1	0
	0	1	-

NOT - 2  
2-NAND - 4  
3-NAND - 1  
26 transistori  
[13 literaali]

abc	xy
-01	11
1-0	10
11-	01

	c	b	
-	1	0	0
a	1	1	0
	0	1	0
	0	1	-

NOT - 2  
2-NAND - 5  
3-NAND - 0  
24 transistori  
[12 literaali]

abc	xy
-01	<u>01</u>
110	11
-0-	10

	c	b	
-	1	0	0
a	1	1	0
	0	1	0
	0	1	-

NOT - 2  
2-NAND - 3  
3-NAND - 1  
22 transistori  
[11 literaali]

abc	xy
1-0	10
-0-	10
11-	01
-01	<u>01</u>

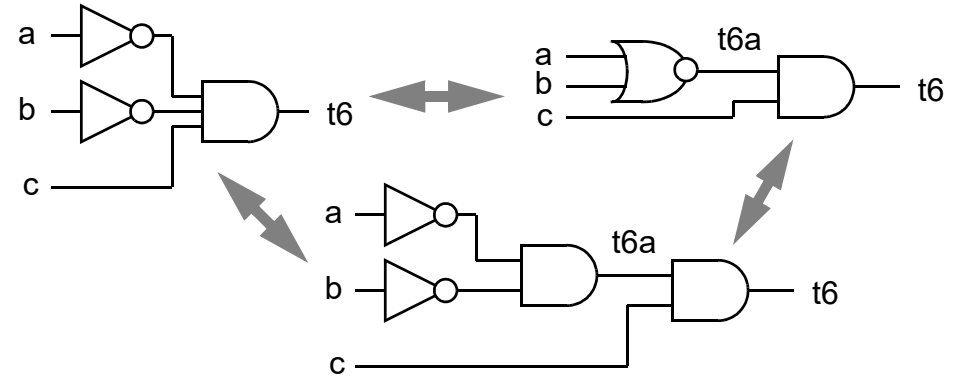
	c	b	
-	1	0	0
a	1	1	0
	0	1	0
	0	1	-

NOT - 2  
2-NAND - 5  
3-NAND - 0  
24 transistori  
[12 literaali]

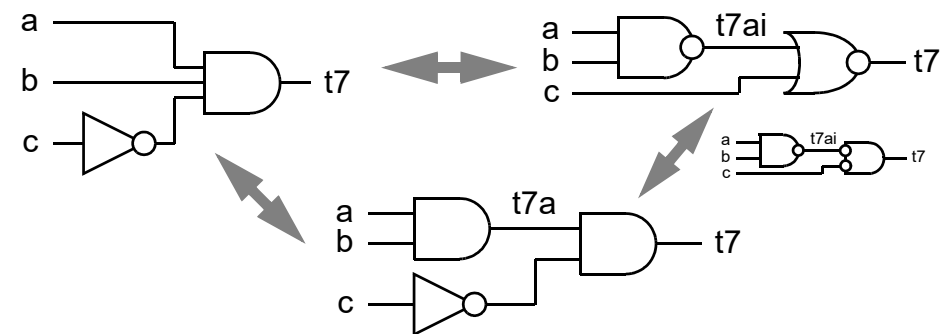
# Teisendused mitmetasemelisel realisatsioonil

## De Morgani & topelteiluse seadused

- $t6 = a' b' c$  ;
- $t6a = a' b'$  ;  $t6 = t6a c$  ;
- $t6a = (a + b)'$  ;  $t6 = t6a c$  ;

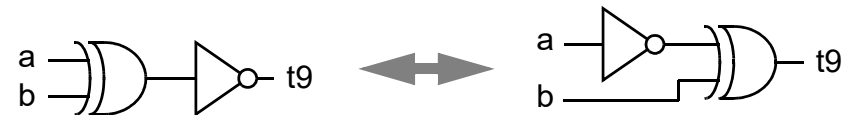


- $t7 = a b c'$  ;
- $t7a = a b$  ;  $t7 = t7a c'$  ;
- $[t7ai = (a b)'$  ;  $t7 = t7ai' c'$  ;]
- $t7ai = (a b)'$  ;  $t7 = (t7ai + c)'$  ;



a	b	$\bar{a} \bar{b}$
0	0	1
0	1	0
1	0	0
1	1	1

a	b	$\bar{a} \bar{b}$
0	0	0
0	1	1
1	0	1
1	1	0



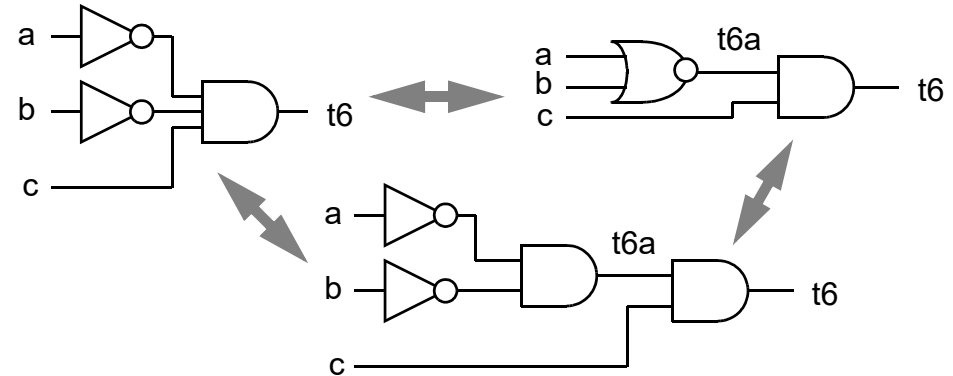
# Modelleerimine, mitmetasemeline realisatsioon, teisendused, signaali hilistumine...

```
entity t6_ver is
  port ( a, b, c: in bit; t6: out bit );
end entity t6_ver;
```

```
architecture yks of t6_ver is
  signal ai, bi: bit;
begin
  ai <= not a;    bi <= not b;
  t6 <= ai and bi and c;
end architecture yks;
```

```
architecture kaks of t6_ver is
  signal ai, bi, t6a: bit;
begin
  ai <= not a;    bi <= not b;
  t6a <= ai and bi;    t6 <= t6a and c;
end architecture kaks;
```

```
architecture kolm of t6_ver is
  signal t6a: bit;
begin
  t6a <= a nor b;    t6 <= t6a and c;
end architecture kolm;
```



Signaalide levimisteed:

yks: a - not - 3-and - t6 / b - not - 3-and - t6 / c - 3-and - t6

kaks: a - not - 2-and - 2-and - t6 /  
b - not - 2-and - 2-and - t6 / c - 2-and - t6

kolm: a - 2-nor - 2-and - t6 / b - 2-nor - 2-and - t6 /  
c - 2-and - t6