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

TTÜ1918

Loogikasünteesi eesmärgid

Lähteülesanne

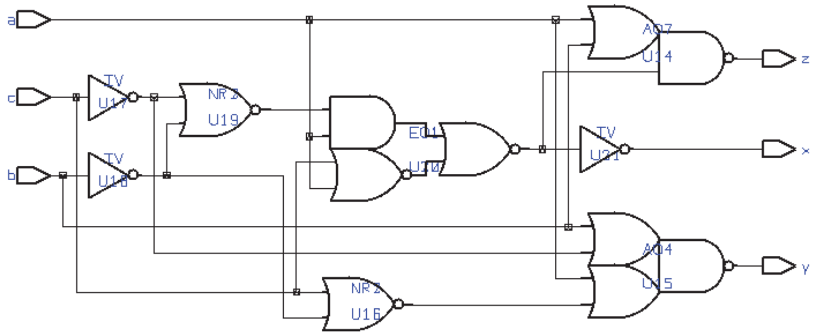
(tõeväärtustabel)

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



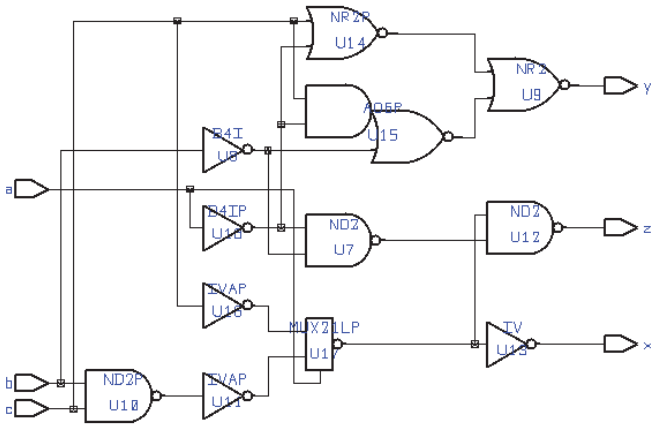
pindala 12.0
viide 2.73 ns
võimsus 11.3 μW

JA-EI (AND-OR):

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

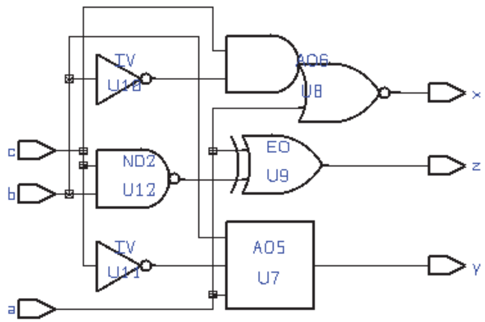
$$y = \bar{b}\,c + \bar{a}\,\bar{b} + \bar{a}\,c$$

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

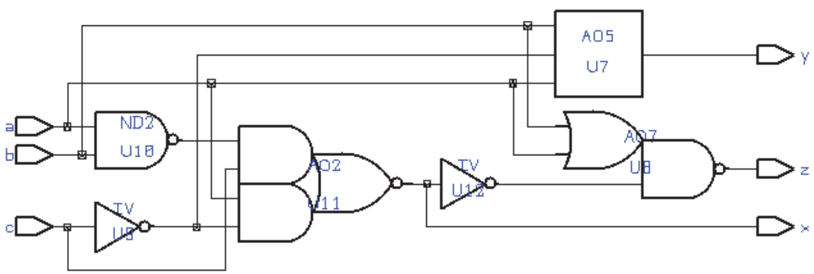


pindala 22.0
viide <u>1.57</u> ns
võimsus 24.1 μW

Loogikasünteesi eesmärgid (2)

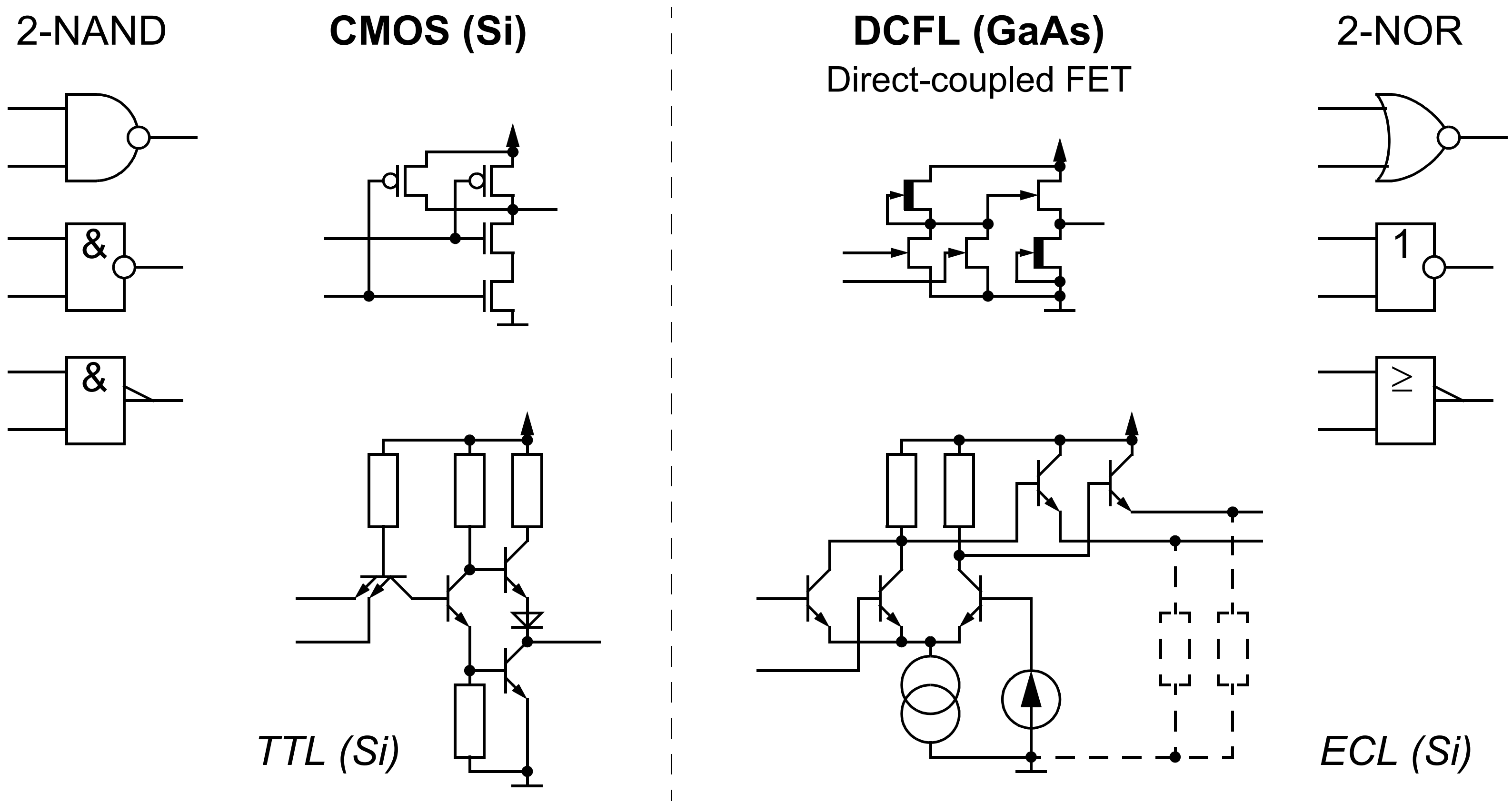


pindala 11.0
viide 1.84 ns
võimsus 9.2 μW



pindala 10.0
viide 2.26 ns
võimsus 10.2 μW

Baaselemendid – NAND vs NOR



Loogikafunktsioonide süsteem

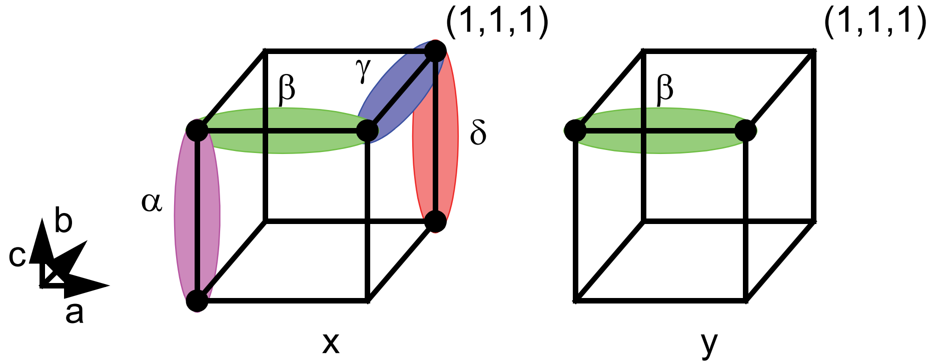
- **Kombinatsioonskeemi “musta kasti” mudel**
- **Defineeritud Boole’i algebra baasil – (B,+,*,~), B={0,1}**
- **Loogikafunktsioonid võivad olla**
  - **mitme väljundiga (funktsioonide süsteem) –** $f : B^n \rightarrow B$ **ja** $f : B^n \rightarrow B^m$
  - **osaliselt (mittetäielikult) määratud –** $f : B^n \rightarrow \{0,1,-\}^m$ **(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** ***elementaarkonjuktsioon*** **– 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



hüper-kuup

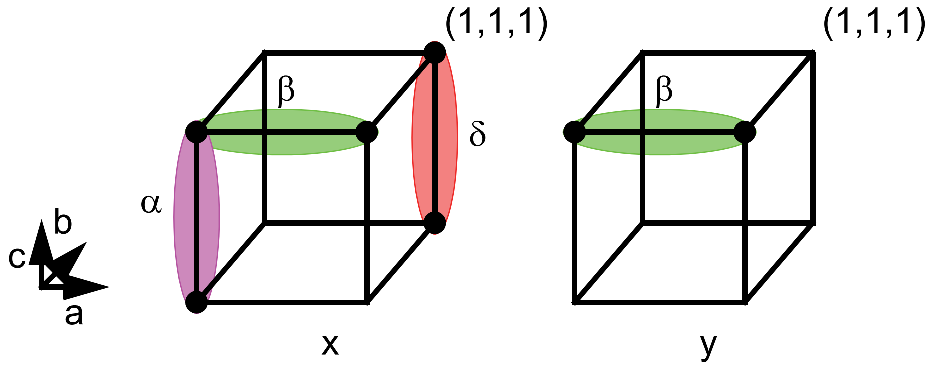
Definitsioone – kate

- *Miinimumkate* (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

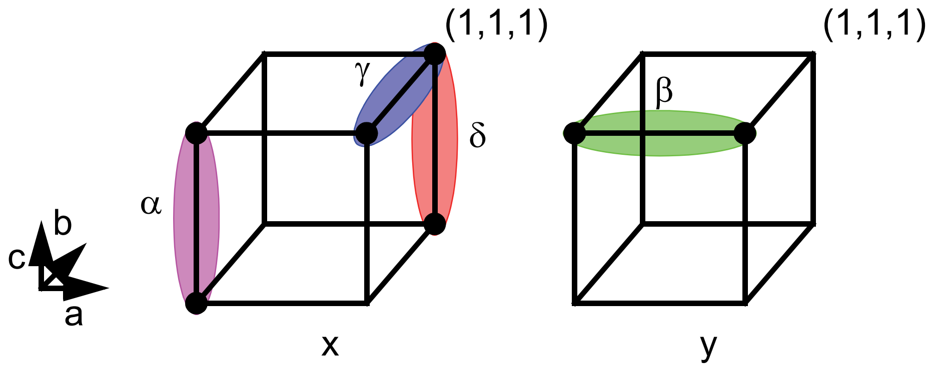
$x - \alpha$ & δ

$y - \beta$



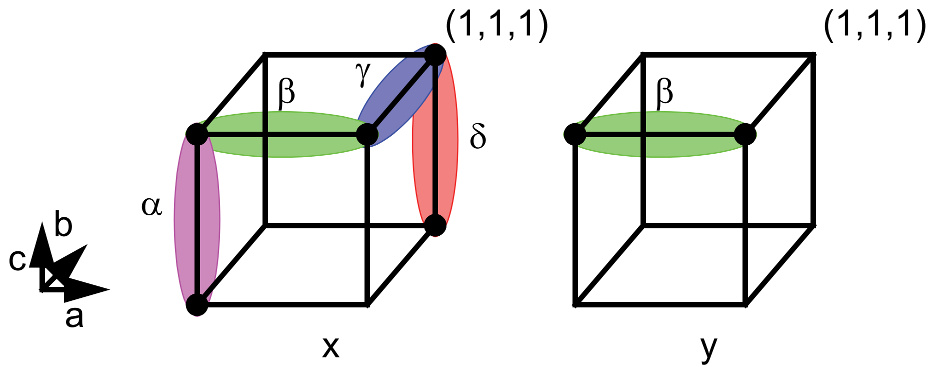
lihtkate

miinimumkate



lihtkate

minimaalne kate



lihtkate

Loogikafunktsioonide minimeerimine

- Süntees ja optimeerimine
  - lähtekirjelduse (tabel, skeem, HDL) teisendamine abstraktseks mudeliks
  - teisendused abstraktsel mudelil (sobivad analüüsiks ja masintöötluseks)
  - sidumine elementidega teegist
- Täpsed meetodid (nt. Quine-McCluskey meetod)
  - leiavad miinimumkatte
  - tihtipeale võimatu suurte funktsioonide korral
- Heuristilised meetodid (MINI, PRESTO, ESPRESSO, ...)
  - leiavad minimaalsed katted (miinimumkatte leidmine võimalik)
- Quine'i teoreem – miinimumkate on lihtkate
  - $\rightarrow$ miinimumkatte 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 <u>täpselt ühes</u> järgus (vt. kleepimisseadused)
- Võib vaadelda sulgude ette toomisega
  - kleepuvad: $a\,b\,c + a\,\overline{b}\,c = a\,c\,(\,b + \overline{b}\,) = a\,c\,(\,1\,) = a\,c$
  - ei kleepu: $a\,b\,c + a\,\overline{b}\,\overline{c} = a\,(\,b\,c + \overline{b}\,\overline{c}\,)$

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

	$\overline{b}\,\overline{c}$	$\overline{b}\,c$	$b\,c$	$b\,\overline{c}$
$\overline{a}$	1	1	1	0
a	0	1	0	0

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

	$\overline{b}\,\overline{c}$	$\overline{b}\,c$	$b\,c$	$b\,\overline{c}$
$\overline{a}$	1	1	1	0
a	0	1	0	0

- Sarnane dubleerimine töötab ka KNK korral

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

	1	0	0	1
b	1	1	1	1
b, a	0	1	0	0
a	1	1	1	1

$f = b\,\bar{c}\,d + \bar{a}\,\bar{d} + \bar{a}\,b + a\,\bar{b}$ DNK 1-dest

$f = (a+b+\bar{d})\ (\bar{a}+\bar{b}+d)\ (\bar{a}+\bar{b}+\bar{c})$ KNK 0-dest

$\bar{f} = \bar{a}\,\bar{b}\,d + a\,b\,\bar{d} + a\,b\,c$ DNK 0-dest

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

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

Lihtimplikantide leidmine

- **Implikantide kleepimine**
  - **erinevus <u>täpselt ühes</u> järgus (vt. kleepimisseadused)**
  - **võib vaadelda sulgude ette toomisega**
    - **kleepuvad:** $a\,b\,c + a\,\overline{b}\,c = a\,c\,(b + \overline{b}) = a\,c\,(1) = a\,c$
    - **ei kleepu:** $a\,b\,c + a\,\overline{b}\,\overline{c} = a\,(b\,c + \overline{b}\,\overline{c})$

- **Alates kahestest kontuuridest (vähemalt üks ebaoluline muutuja implikandis) leidub alati <u>kaks või enam paari</u> 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

$\overline{a}\,b\,d + a\,b\,d = b\,d$

$b\,\overline{c}\,d + b\,c\,d = b\,d$

01–1	–101
11–1	–111
–1–1	–1–1

0	1	0	1
1	1	1	1
–	1	–	1
0	0	0	0

$\overline{a}\,b + a\,b = b$

$b\,\overline{c} + b\,c = b$

$b\,\overline{d} + b\,d = b$

01––	–10–	–1–0
11––	–11–	–1–1
–1––	–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
~~0000~~	-000
~~0010~~	0-10
~~0100~~	-010
~~0101~~	010-
~~0110~~	01-0
~~0111~~	100-
~~1000~~	10-0
~~1001~~	01-1
~~1010~~	-101
~~1011~~	011-
~~1101~~	10-1
00-0	1-01
0-00	101-

2. etapp

abcd	abcd
~~00-0~~	~~10-1~~
~~0-00~~	1-01
~~-000~~	~~101-~~
~~0-10~~	0--0
~~-010~~	~~0--0~~
~~010-~~	-0-0
~~01-0~~	~~-0-0~~
~~100-~~	01--
~~10-0~~	~~01--~~
~~01-1~~	10--
-101	~~10--~~
~~011-~~	

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 kattesse 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 – 0100 <> 0110 vs. 4 <> 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

gr.	abcd	
0	0000	*
1	0010	*
	0100	*
	1000	*
2	0101	*
	0110	*
	1001	*
	1010	*
3	0111	*
	1011	*
	1101	*

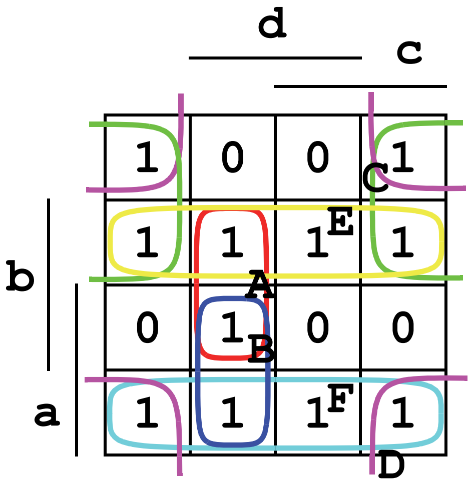
1. etapp

gr.	abcd	
0	00-0	*
	0-00	*
	-000	*
1	0-10	*
	-010	*
	010-	*
	01-0	*
	100-	*
	10-0	*
2	01-1	*
	-101	A
	011-	*
	10-1	*
	1-01	B
	101-	*

2. etapp

gr.	abcd	
0	0--0	C
	-0-0	D
1	01--	E
	10--	F

* - 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

gr.		
0	0	*
1	2	*
	4	*
	8	*
2	5	*
	6	*
	9	*
	10	*
3	7	*
	11	*
	13	*

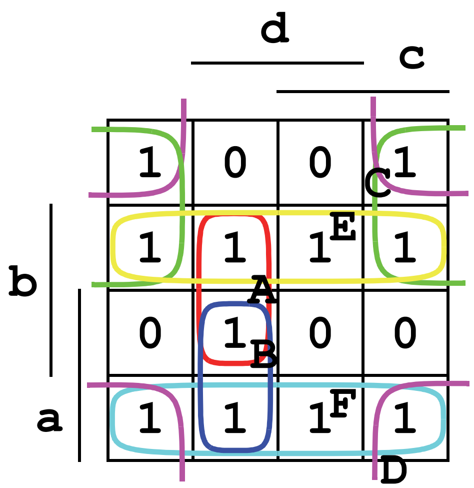
1. etapp

gr.			
0	0	(2)	*
	0	(4)	*
	0	(8)	*
1	2	(4)	*
	2	(8)	*
	4	(1)	*
	4	(2)	*
	8	(1)	*
	8	(2)	*
2	5	(2)	*
	5	(8)	A
	6	(1)	*
	9	(2)	*
	9	(4)	B
	10	(1)	*

2. etapp

gr.			
0	0	(2,4)	C
	0	(2,8)	D
1	4	(1,2)	E
	8	(1,2)	F

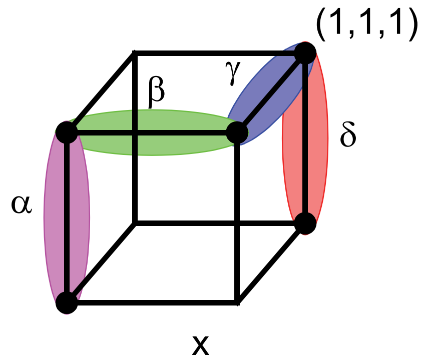
* - on kaetud



Lihtimplikantide katte leidmine

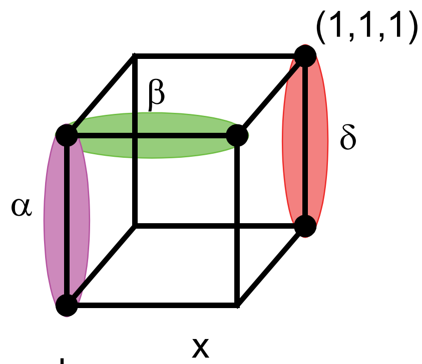
- $x = \overline{a}\,\overline{b}\,\overline{c} + \overline{a}\,\overline{b}\,c + a\,\overline{b}\,c + a\,b\,\overline{c} + a\,b\,c$

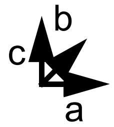
	abc x
α	00- 1
β	-01 1
γ	1-1 1
δ	11- 1



- **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	α	β	γ	δ
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

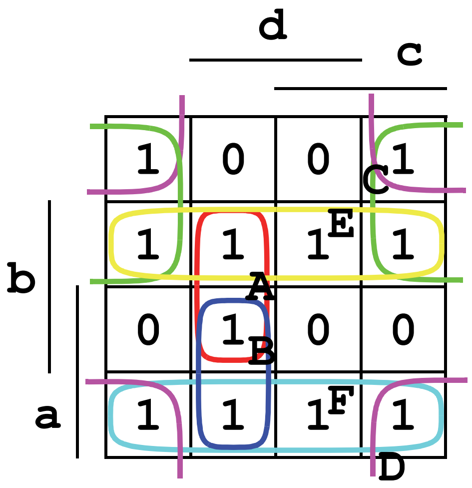




Petrick'i meetod – näiteülesanne

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+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

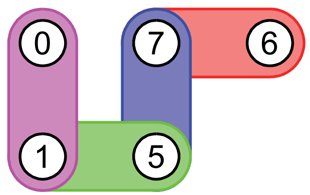
(ACDBEF+ACDAEF+ACFBEF+ACFAEF+CEFBEF+CEFAEF+DEBEF+DEAEF)=1

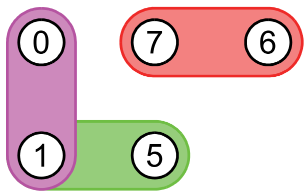
ACEF+ADEF+BCEF+BDEF=1

- **Maatriksesitus**
  - **Täisarvuline lineaarplaneerimisülesanne (integer linear programming, ILP)**
  - **Implikantide tabel on kahendmaatriks *A***
  - **Valitud implikandid on kahendvektor *x***
  - **Leida selline *x*, et**
    - $A\,x \geq 1$
    - **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 kattesse**
  - **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**
  - **maatriksis - $a_{ki} \le a_{kj}$ $\forall k$; 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)**
  - **maatriksis - $a_{ik} \ge a_{jk}$ $\forall k$; 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

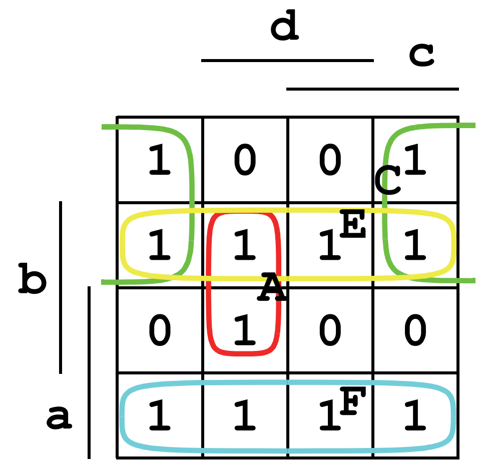
	abcd	A	B	C	D	E	F
	0000			+	+		
	0010			+	+		
	~~0100~~			~~+~~		~~+~~	
	~~1000~~				~~+~~		~~+~~
	~~0101~~	~~+~~				~~+~~	
	~~0110~~			~~+~~		~~+~~	
	~~1001~~		~~+~~				~~+~~
	~~1010~~				~~+~~		~~+~~
*	0111					*	
*	1011						*
	1101	+	+				

E, F

* - olulised

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

A, C



$$f = b\,\overline{c}\,d + \overline{a}\,\overline{d} + \overline{a}\,b + a\,\overline{b}$$

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

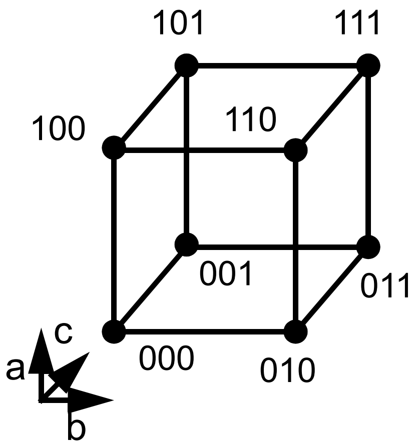
TTÜ1918

Karnaugh kaart

c b

	000	001	011	010
a	100	101	111	110

≡



≡

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

≡

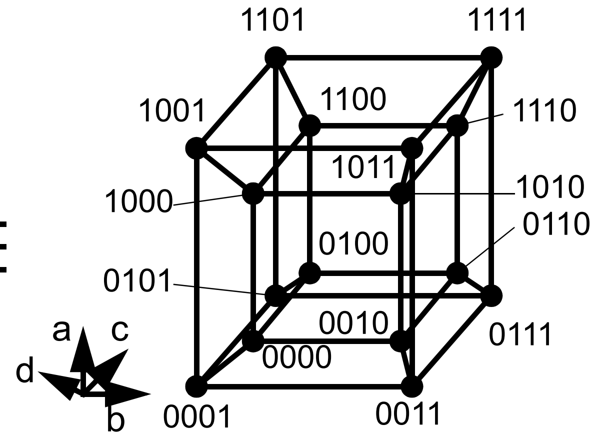
c a

b	011	111	110	010
	001	101	100	000

d c

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

≡



d b

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

≡

Karnaugh kaart – näide

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

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

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

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

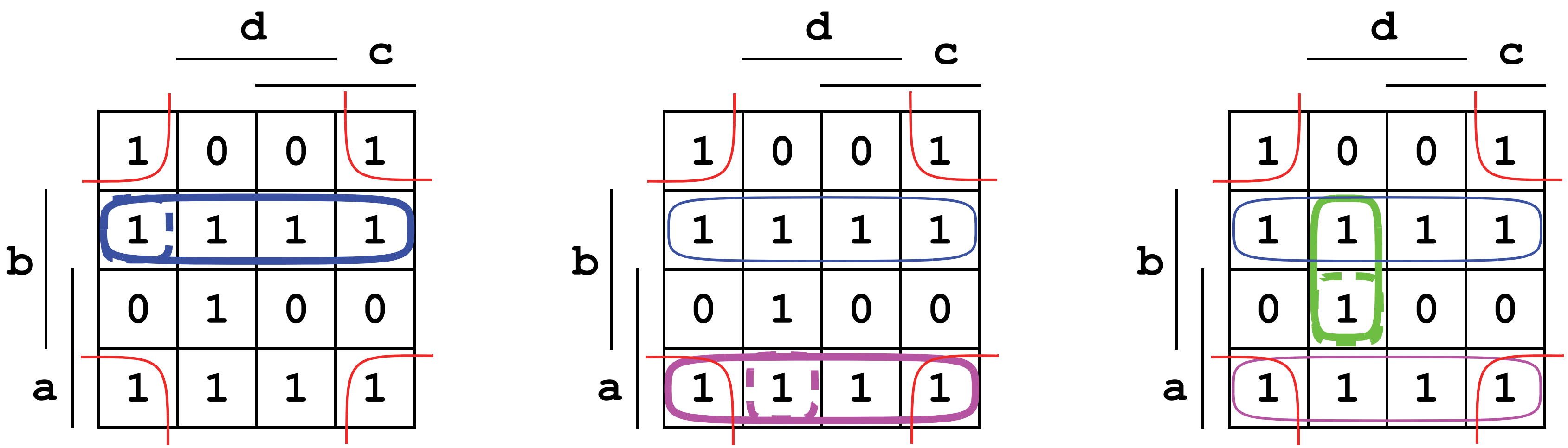
- Implikandi (kontuuri) laiendamine
- Vt. ka http://tams-www.informatik.uni-hamburg.de/applets/kvd/kvd.html

TTÜ1918

Heuristilised meetodid – näide

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

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



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

Nõrgalt määratud funktsioonide minimeerimine

- $|F_F| + |F_R| << |F_D|$
- Funktsiooni argumentide arv on suur
- $|F_F| + |F_R|$ on esitatud intervallidena, st. on olemas esialgne kate

- F_F-i kuuluvaid intervalle 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 $\rightarrow$ 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

Column labels e, d, e, c; row labels b, a.

0	0	0	1	-	0	0	0
0	0	0	1	-	0	0	0
-	-	1	1	1	1	-	-
-	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
```

TTÜ1918

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**
- **Liiasusetus (Irredundant)**
  - **liiasuse eemaldamine kattest**

Näide

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

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	*exp*
0010	x
0100	x
0101	
0110	x
0111	
1000	
1001	
1010	
1011	
1101	

0000
↓
0--0

0--0
katab
0010
0100
0110

0--0	a
0101	*exp*
0111	x
1000	
1001	
1010	
1011	
1101	

0101
↓
01--

01--
katab
[0100]
[0110]
0111

0--0	a
01--	c
1000	*exp*
1001	
1010	x
1011	
1101	

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

Näide – laiendus – kõik sammud

0000	exp
0010	x
0100	x
0101	
0110	x
0111	
1000	
1001	
1010	
1011	
1101	

0--0	a
0101	exp
0111	x
1000	
1001	
1010	
1011	
1101	

0--0	a
01--	c
1000	exp
1001	
1010	x
1011	
1101	

0--0	a
01--	c
-0-0	b
1001	exp
1011	x
1101	

0--0	a
01--	c
-0-0	b
10--	d
1101	exp

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

{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

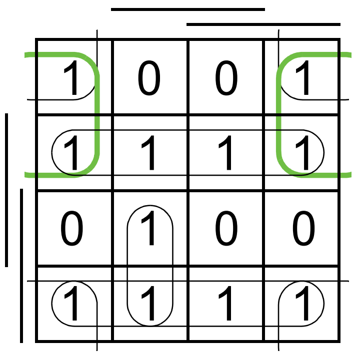
0--0

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

Näide – kitsendus – kõik sammud

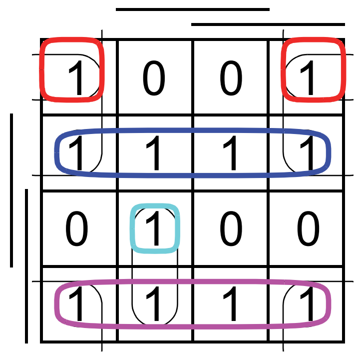
0--0	xxxx
01--	c
-0-0	b
10--	d
1-01	e

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’ }



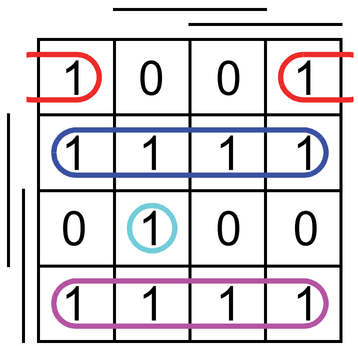
Näide – ümberkujundus

01--	01-1
00-0	0--0
10--	d
1101	e′

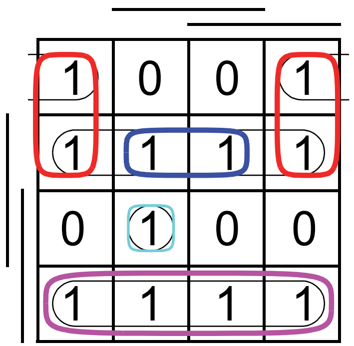
→

01-1	c′
0--0	a
10--	d
1101	e′

{ b’,c,d,e’ }



→



{ a,c’,d,e’ }

Näide – laiendus #2

01-1	*exp*
0--0	a
10--	d
1101	e′

01--	c
0--0	a
10--	d
1101	*exp*

01--	c
0--0	a
10--	d
-101	f

{ a,c,d,f }

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

Kokkuvõte

- **MINI**
  - **Laiendus: kate – {a,b,c,d,e} – lihtkate, liiasusega (kuid ükski implikant ei sisaldu mõnes teises)**
  - **Kitsendus: a eemaldatakse; b [-0-0] → b' [00-0] ; e [1-01] → e' [1101] ; kate – (b',c,d,e')**
  - **Ümberkujundus: {b',c} [00-0][01--] → {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 eemaldmist)
  - Iteratsioon – laiendus, liiasusetus, kitsendus
  - Kaalufunktsioonid – katte võimsus & intervallide ja literaalide arvu kaalutud summa

TTÜ1918

Funktsioonide süsteemi minimeerimine

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

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

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

funktsioonid eraldi

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

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

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

funktsioonid korraga

Funktsioonide süsteemi minimeerimine

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

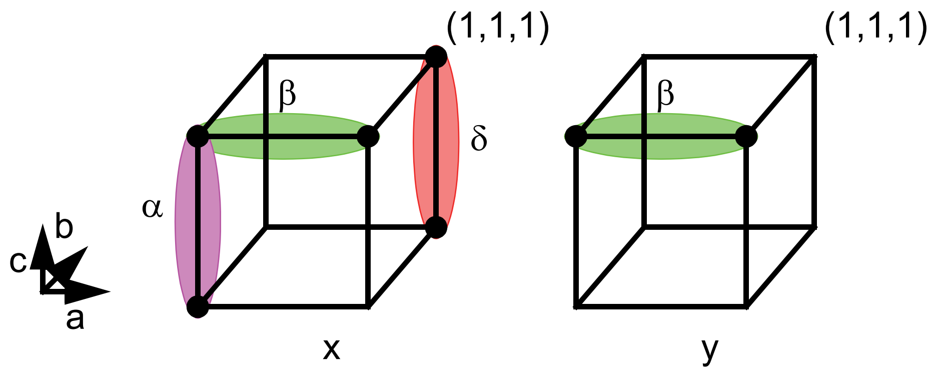
abc	xy
000	10
001	11
101	11
110	10
111	10

abcO	
0001	1
001-	1
101-	1
1101	1
1111	1

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

abcO	
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

a b a

c d

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

O

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

mitme-valentse loogika abil

a b

c d

0	A	0	0
0	A	B	0
0	B	B	0
0	0	0	0

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

sümbol-kodeeringu abil

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,*,+,~), 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 – ($\{P_i\}$,MIN,MAX,literal), $P_i=\{0,1,...,m_i-1\}$**
  - **mittetäielikult määratud funktsioonid –** $f : P_1xP_2x...xP_n \rightarrow P_m$
    või ka $f : P_1xP_2x...xP_n \rightarrow \{0,1,-\}^m$
  - **MIN, MAX, literal - MVL-funktsioonide täielik süsteem**

TTÜ1918

Mitme-valentne loogika – operatsioonid

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

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_2 \ x_1	0	1	2
0	0	1	0
1	0	1	0
2	2	0	2

x_2 \ x_1	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:** $\{0,1\}^n \rightarrow \{0,1\}^k \equiv \{0,1\}^n \times \{0, 1, ..., m\text{-}1\} \rightarrow \{0,1\}$
- **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: $a^0b^1c^1e^0 + a^0b^1c^1e^1 = a^0b^1c^1\ (e^0+e^1) = a^0b^1c^1e^{\{0,1\}}$
- Vektoresitus – 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

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

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

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

f: {0,1}x{0,1}x{0,1}x{0,1,2} -> {0,1}

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

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*

3. *7.*

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

1. *2.* *9.* *11.*

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

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

5. *9.* *6.* *10.*

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

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

TTÜ1918

Näide – minimeerimine

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*

1. *2.* *9.* *11.*

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

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

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
	. . .	

2. etapp

gr.	abc	e
0	000	111
	0-0	101
	. . .	

c b

a

1	0	1	1
0	0	1	0

1	1	1	0
0	1	0	0

1	1	0	1
0	0	1	0

Näide – minimeerimine

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*

mintermid

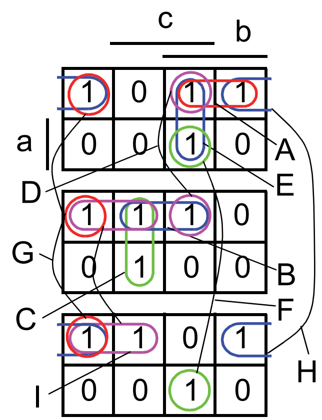
gr.	abc	e	
0	000	100	*
	000	010	*
	000	001	*
1	010	100	*
	001	010	*
	001	001	*
	010	001	*
2	011	100	*
	011	010	*
	101	010	*
3	111	100	*
	111	001	*

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	A
	0-1	010	B
	-01	010	C
2	011	110	D
	-11	100	E
3	111	101	F

2. etapp

gr.	abc	e	
0	000	111	G
	0-0	101	H
	00-	011	I



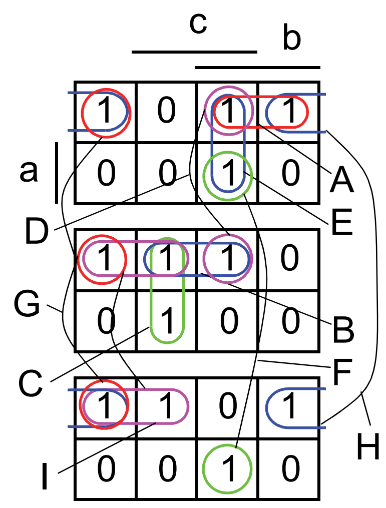
Näide – minimeerimine

lihtimplikantide tabel

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

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

abc	e	o	
011	110	1	D
111	101	1	F
0-0	101	1	H
00-	011	1	I
-01	010	1	C



Näide (pärast minimeerimist)

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

Lihtimplikandid & liiasuseta

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

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

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

Näide (minimeeritud kahendfunktsioonid)

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

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

$$x(a,b,c)=\overline{a}bc+abc+\overline{a}\,\overline{c}$$
$$y(a,b,c)=\overline{a}bc+\overline{a}\,\overline{b}+\overline{b}c$$
$$z(a,b,c)=abc+\overline{a}\,\overline{c}+\overline{a}\,\overline{b}$$

Näide #2

f_1: x_2 \ x_1	0	1
0	0	1
1	1	1

f_2: x_2 \ x_1	0	1
0	1	0
1	1	1

f_3: x_2 \ x_1	0	1
0	1	0
1	0	0

f_4: x_2 \ x_1	0	1
0	0	1
1	0	0

2 + 2 + 1 + 1 = 6

implikanti

x_3 \ $x_1 x_2$	00	01	11	10	
0	0	1	1	1	f_1
1	1	1	1	0	f_2
2	1	0	0	0	f_3
3	0	0	0	1	f_4

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
	~~00~~	~~0100~~	+	+			
*	00	0010	*				
	~~01~~	~~1000~~			+	+	
*	01	0001			*		
*	10	1000					*
	~~10~~	~~0100~~		+			+
	~~11~~	~~1000~~				+	+
*	11	0100					*

f_1: x_2 \ x_1	0	1
0	0	1
1	1	1

f_2: x_2 \ x_1	0	1
0	1	0
1	1	1

f_3: x_2 \ x_1	0	1
0	1	0
1	0	0

f_4: x_2 \ x_1	0	1
0	0	1
1	0	0

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	*			
	~~001 10~~	+			+
*	001 01				*
	~~101 10~~		+		+
*	101 01				*
*	110 10			*	
	~~111 10~~		+	+	

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

abc	xy
00-	10
11-	10
-01	11

TTÜ1918

Näide #4 – määramatused

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

mintermid

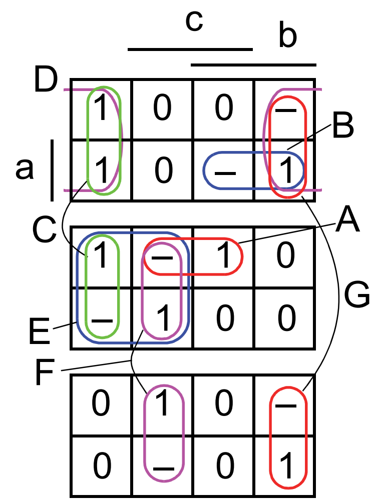
gr.	abc	e	
0	000	100	*
	000	010	*
1	*001	010	*
	001	001	*
	*010	100	*
	*010	001	*
	100	100	*
	*100	010	*
2	011	010	*
	101	010	*
	*101	001	*
	110	100	*
	110	001	*
3	*111	100	*

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

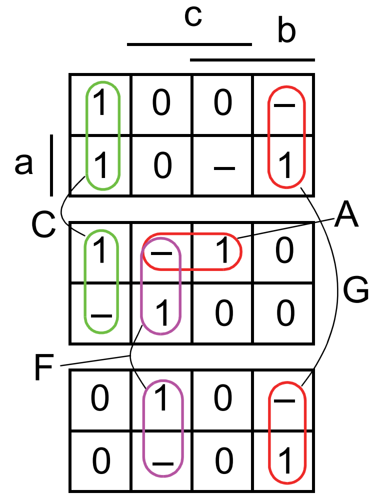


Näide #4 – määramatused

lihtimplikandid & 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			+	+			
	~~101~~	~~010~~					~~+~~	~~+~~	
	~~110~~	~~100~~		~~+~~		~~+~~			~~+~~
*	110	001							*

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

abc	xyz	
0-0	010	A
-00	110	C
-01	011	F
-10	101	G

Näide #5 – heuristiline minimeerimine

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

lähteülesanne...

k:

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

l:

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

m:

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

n:

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

Millest alustada?

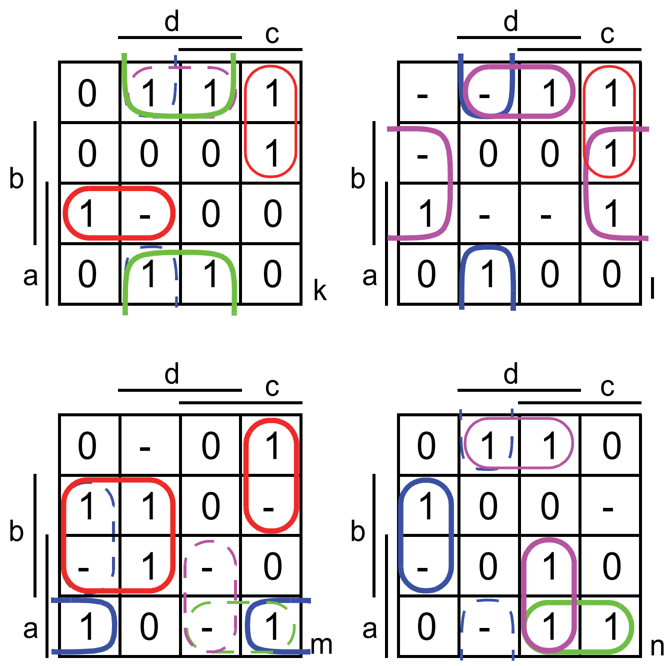
Vrdl. olulised lihtimplikandid –>
igal juhul peab olema kaetud –>
“üksikud ühed”, siis “paarid”, ...

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-	0001
0-10	1110
-001	0100
00-1	0101
-0-1	1000
110-	1000
-100	0001
1-11	0001



Edasi need, mis veel katmata...

Jälle alustada sealt, kus vähe ühtesi alles...

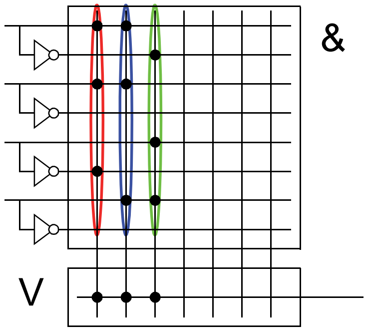
Eemaldada need, mis juba kaetud (punktiirjoonega)

Variante on rohkemgi...

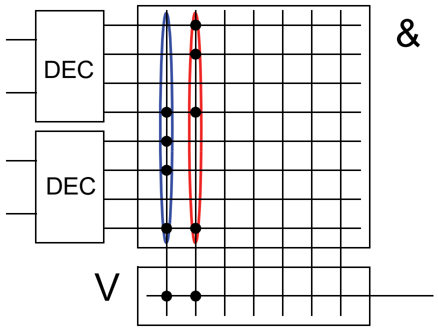
MVL rakendusi

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

x_3x_4 \ x_1x_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



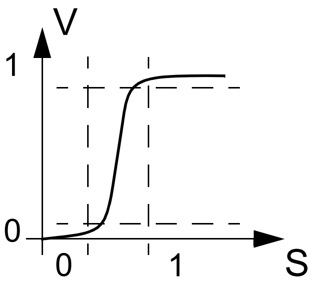
y_2 \ y_1	0	1	3	2
0	0	0	1	0
1	0	0	1	0
3	1	1	1	0
2	0	0	0	0



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

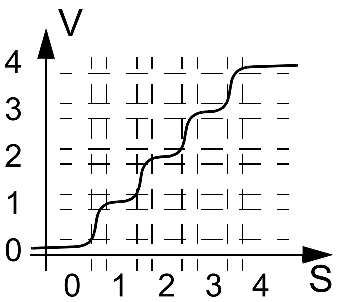
MVL eelised riistvaras

- **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**

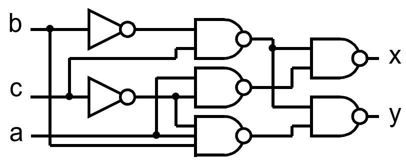


Kahetasemeline minimeerimine ja mitmetasemeline realisatsioon

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·c;
t2=a·ci; t3=a·b·ci;
x=t1+t2; y=t1+t3;



abc	xy
-01	11
1-0	10
110	01

c b a

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

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

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

c b a

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

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

abc	xy
-01	01
110	11
-0-	10

c b a

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

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

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

c b a

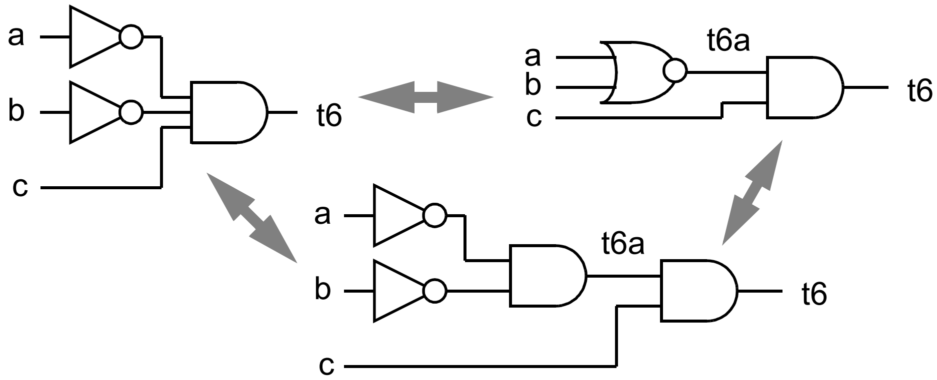
-	1	0	0
1	1	0	1
0	1	0	0
0	1	-	1

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

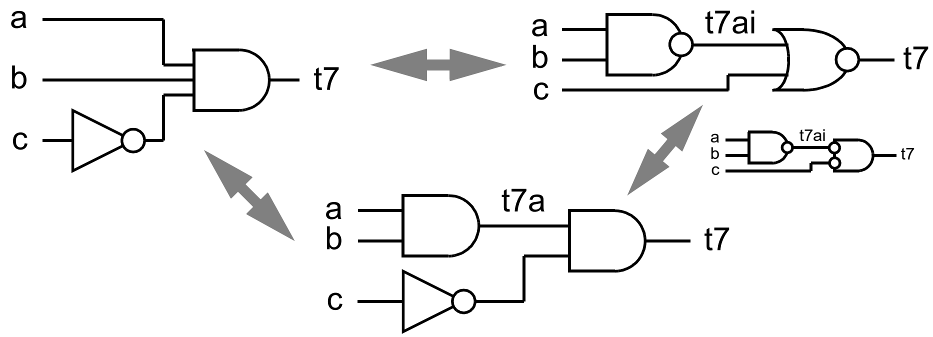
Teisendused mitmetasemelisel realisatsioonil

- **De Morgani & topelteituse seadused**

  - **t6 = a’ b’ c ;**
  - **t6a = a’ b’ ;   t6 = t6a c ;**
  - **t6a = (a + b)’ ;   t6 = t6a c ;**

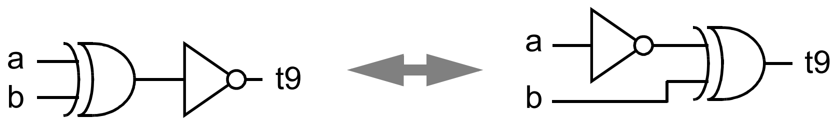


  - **t7 = a b c’ ;**
  - **t7a = a b ;   t7 = t7a c’ ;**
  - **[t7ai = (a b)’ ;   t7 = t7ai’ c’ ;]**
  - **t7ai = (a b)’ ;   t7 = (t7ai + c)’ ;**



a	b	$\overline{\oplus}$
0	0	1
0	1	0
1	0	0
1	1	1

a	b	$\oplus$
0	0	0
0	1	1
1	0	1
1	1	0

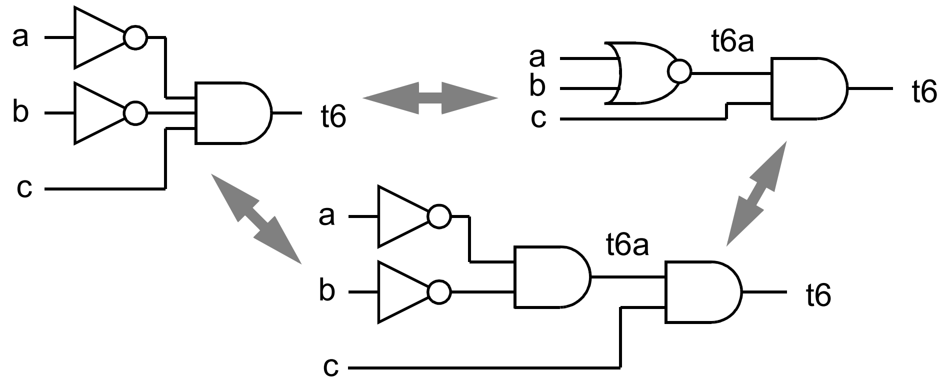


Modelleerimine, mitmetasemeline realisatsioon, teisendused, signaali hilistumine...

```
module t6_ver1 ( input a, b, c, output t6 );
  logic ai, bi;
  assign ai = !a;
  assign bi = !b;
  assign t6 = ai & bi & c;
endmodule

module t6_ver2 ( input a, b, c, output t6 );
  logic ai, bi, t6a;
  assign ai = !a;
  assign bi = !b;
  assign t6a = ai & bi;
  assign t6 = t6 & c;
endmodule

module t6_ver3 ( input a, b, c, output t6 );
  logic t6a;
  assign t6a = !(a | b);
  assign t6 = t6a & c;
endmodule
```



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