

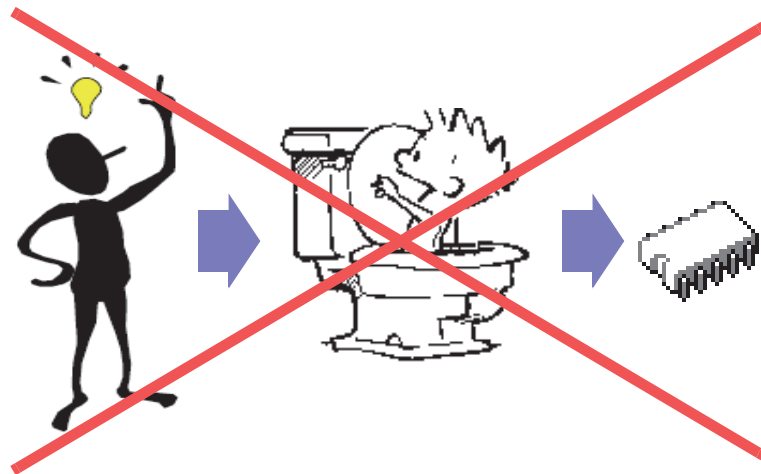


# Digitaalsüsteem

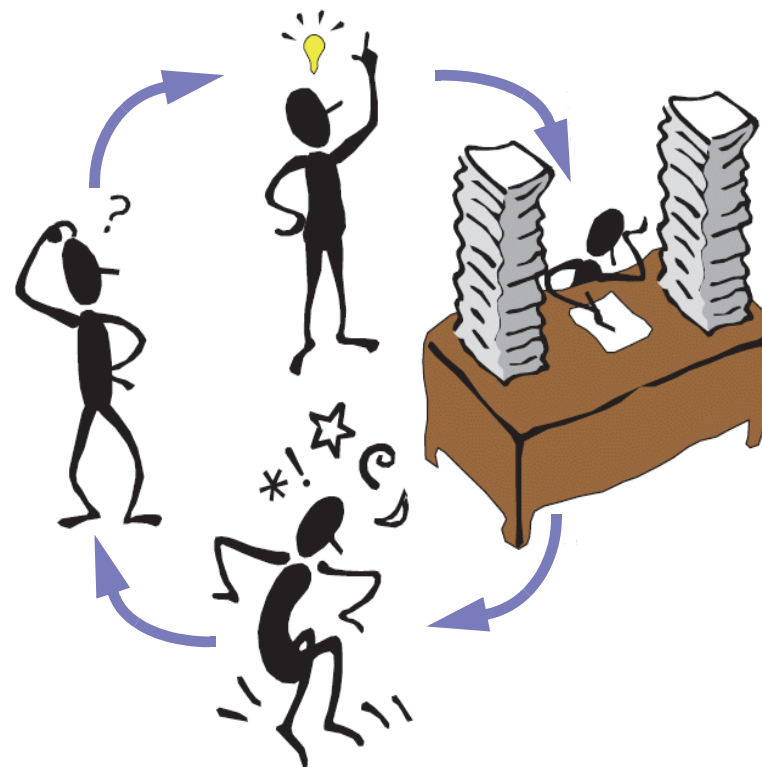
- **Süsteemid**
  - *NB! Piirid pole täpselt paigas...*
  - **Mehhaanikasüsteem** – *liikumine*
  - **Elektrisüsteem** – *elektrienergia*
  - **Elektroonikasüsteem** – *infotöötlus*
    - **Analoogsüsteem** – signaalide esitamine ja töötlus pidevate suurustena  
signaalide väärtused: 0...5 V, -10...+10 mA, jne.
    - **Digitaalsüsteem** – signaalide esitamine ja töötlus diskreetsete suurustena  
signaalide väärtused: 0/1, tõene/väär, true/false, high/low, jne.
- **Sardsüsteem (embedded system)**
  - **Kaasajal peamiselt (hajutatud) digitaalsüsteem**, mis sisaldab nii analoog-alamsüsteeme aga ka mehhaanilisi ja elektrilisi komponente
  - **Suvaline digitaalsüsteem** sisaldab alati analoog, elektrilisi ja mehhaanilisi komponente – nt. nivoomuundurid, toide, lülitid, ...

## Digitaalsüsteemide projekteerimine e. disain

- **Müüt – kõrgtaseme projekteerimine on ainult üks samm**



- **Vajalikud on iteratsioonid**
  - funktsionaalsus
  - disaini eesmärgid



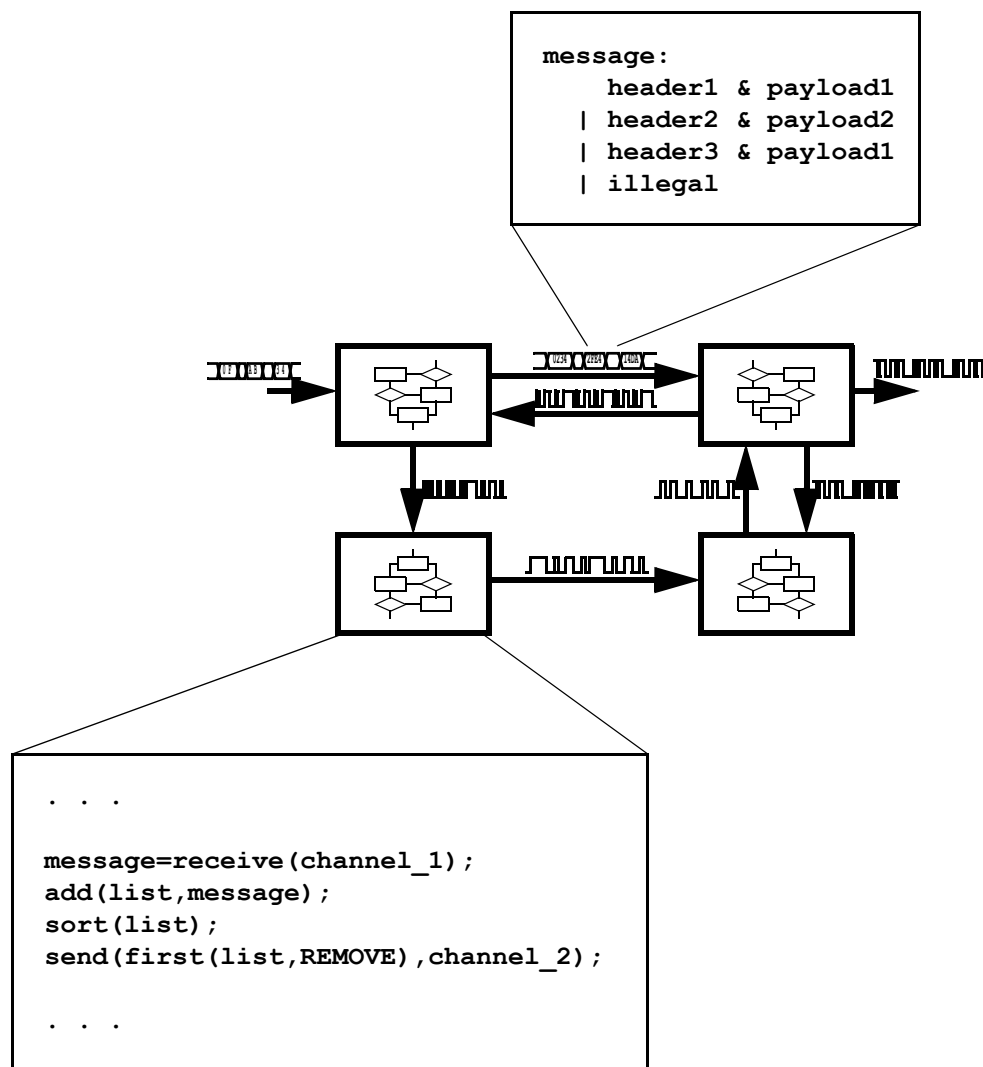


## Disaini põhietapid

- **Süsteemi disain – System design**  
*a.k.a. Arhitektuuri süntees – Architectural-level synthesis*
  - kirjeldus / spetsifikatsioon → plokk-skeem
  - makroskoopilise struktuuri määramine *ehk*  
kuidas on peamised ühendusplokid omavahel ühendatud
- **Loogikadisain – Logic Design**
  - plokk-skeem → loogikalülid
  - mikroskoopilise struktuuri määramine *ehk*  
kuidas on loogikalülid omavahel ühendatud
- **Füüsiline disain – Physical design**  
*a.k.a. Geomeetria süntees – Geometrical-level synthesis*
  - loogikalülid → transistorid, ühendusjuhtmed, mikroskeem

# Abstraktsioonitasemed

- **Süsteemi tase**
  - moodulid / meetodid
  - kanalid / protokollid
  
- **Süsteemi taseme süntees**
  - klasterdamine / tükeldamine
  - liideste süntees

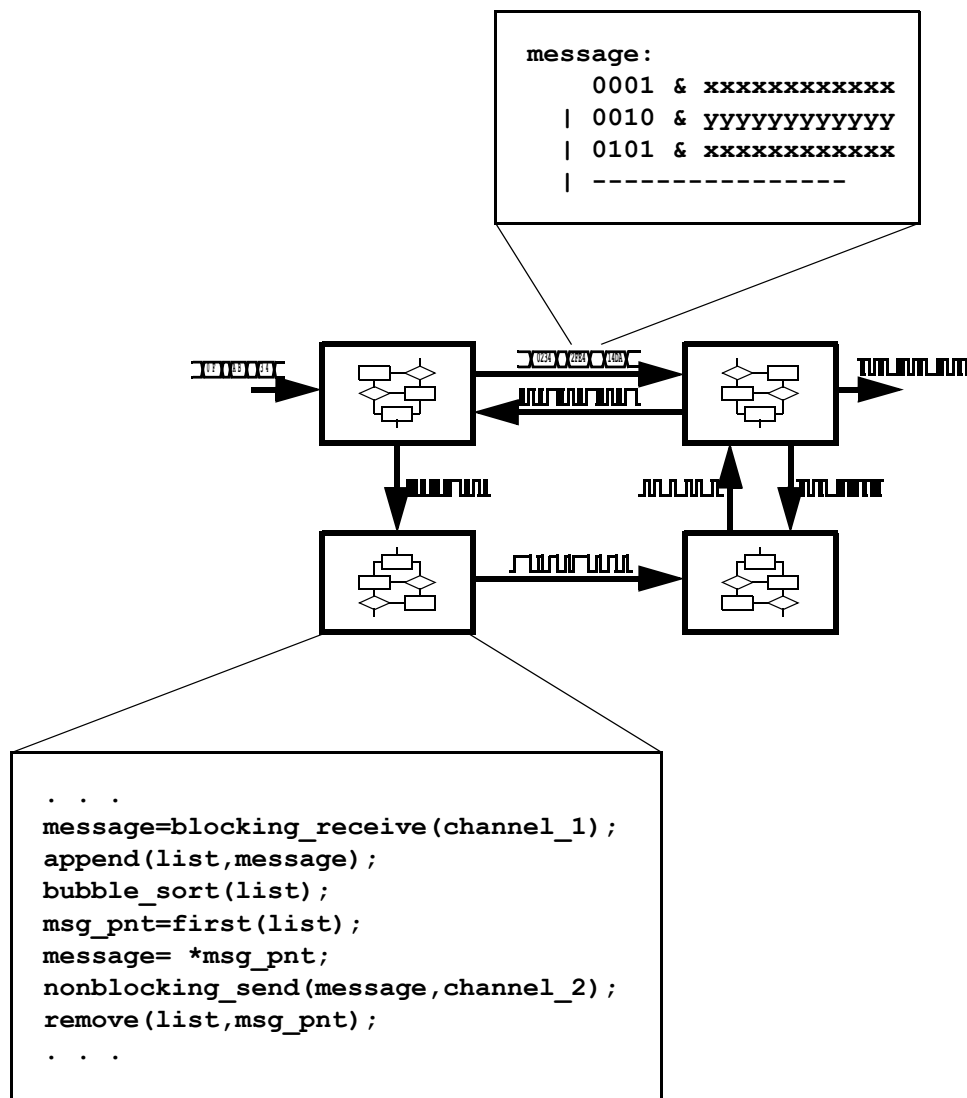


# Abstraktsioonitasemed

- **Algoritmi tase**
  - (alam)moodulid / algoritmid
  - siinid / protokollid

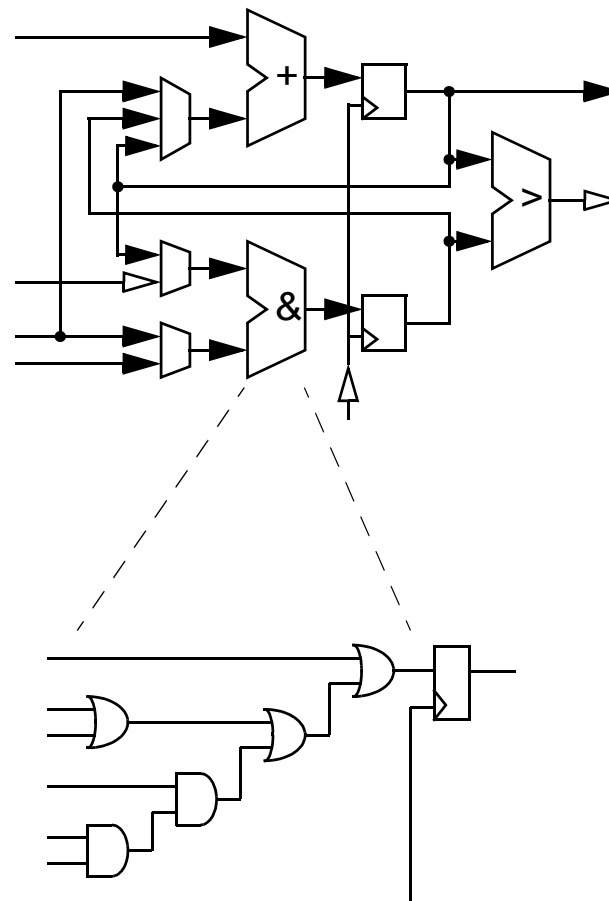
- **Kõrgtaseme süntees**

- ressurs- või aegpiiratud planeerimine
- ressursside eraldamine
- operationide sidumine



# Abstraktsioonitasemed

- **Register-siirete (RT) tase**
  - plokid / loogikatehted
  - siinid / sõnad
- **Registersiirete taseme süntees**
  - andmeosa süntees
  - juhtosa süntees
- **Loogika tase**
  - loogikaelemendid / loogikatehted
  - ahelad / bitid
- **Loogikataseme süntees**
  - loogika minimeerimine
  - optimeerimine, liiasuste eemaldamine

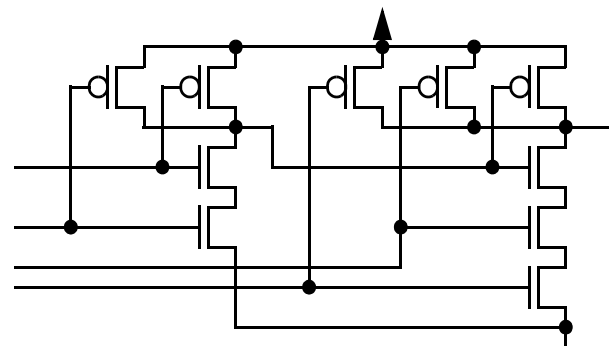
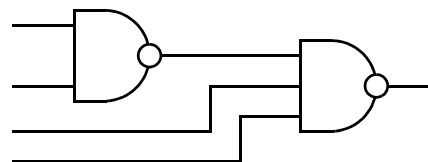


# Abstraktsioonitasemed

- **Füüsikaline tase**

- transistorid / traadid

- polügonid

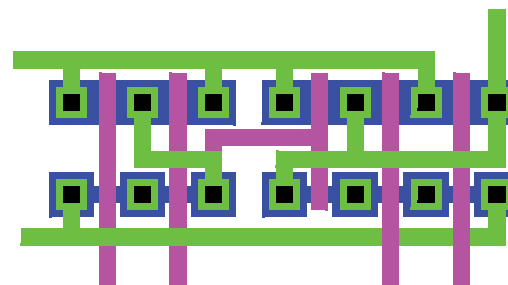


- **Füüsikalise taseme süntees**

- teisendus loogikaelementideks

- paigaldamine

- trasseerimine



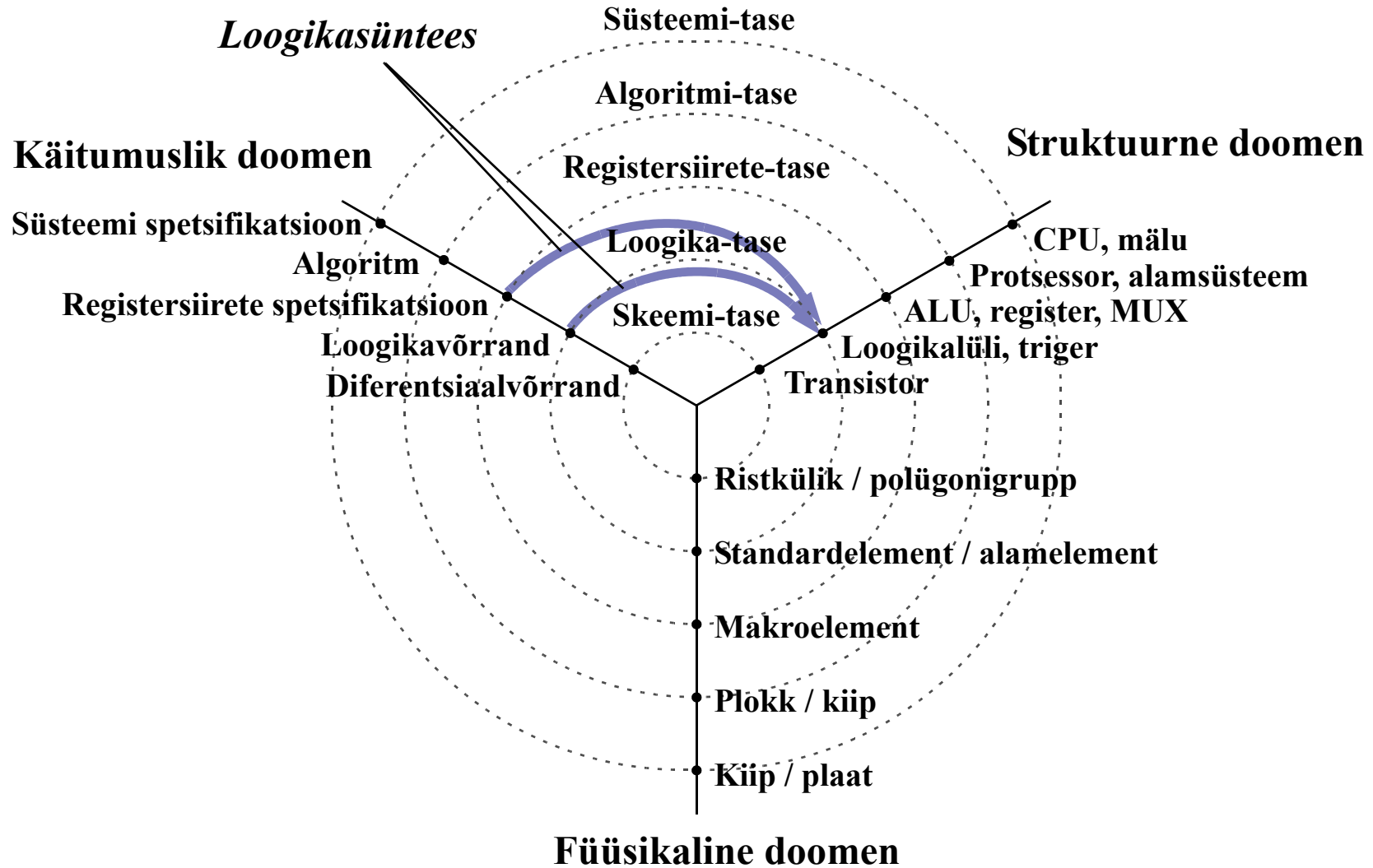


## Abstraktsioonitasemed

Tase	Abstraktsioon	Töövahendid
süsteem	käitumine ruumis ja ajas kui suuniste, ajastuse ja s/v spetsifikatsioonid	plokk-skeemid, diagrammid, kõrgtaseme (programmeerimis) keeled
arhitektuur	funktsionaalsete olemite üldine süsteem (organisatsioon)	riistvara kirjelduskeeled, moodulite paigalduse planeerimine takt-sageduse ja pinna ennustamiseks
registersiirded	andmevoo funktsionaalsete moodulite ja mikrokäskude sidumine	süntees, simuleerimine, verifitseerimine, testi analüüs, ressursside vajaduse hinnang
funktsionaalsed moodulid	primitiivsed operatsioonid ja juhtimisviisid	teegid, mooduli generaatorid, skeemisisestus, testimine
loogika	loogikalülide Boole' funktsioonid	skeemisisestus, süntees ja simuleerimine, verifitseerimine, PLA vahendid
lülitused	transistorahelate elektrilised omadused	RC leidmine, ajastuse verifitseerimine, elektriline analüüs
kristalli pind	geomeetrilised piirangud	pinna redaktor, ahelate ekstraheerimine, DRC, paigaldus ja trasseerimine (ruutimine)



# Loogikasüntees – Logic Synthesis





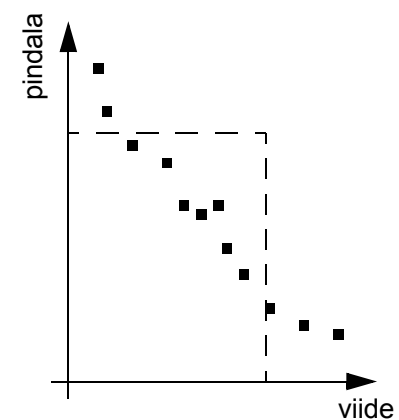
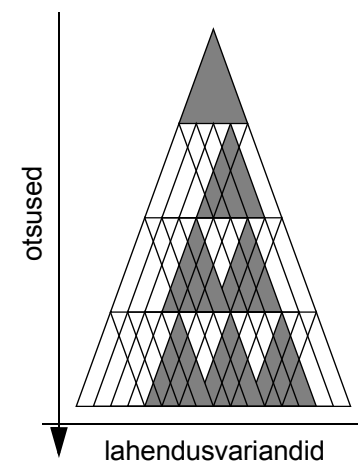
## Automatiseerimise ajalugu (natuke idealistlik vaade)

- 1990 – 4 K gates / year / designer
- 1993 – firmasisene füüsikatase – 5.6K
- 1995 – insener teeb kõik (RTL→GDSII) – 9.1K
- 1997 – väikeste plokkide korduvkasutus (2.5K-75K) – 40K
- 1999 – suurte plokkide korduvkasutus (75K-1M) – 56K
- 2001 – süntees (RTL→GDSII) – 91K
- 2003 – intelligentne testkeskkond – 125K
- 2005 – käitumuslik ja arhitektuurne tase, riist- ja tarkvara (koos)disain – 200K
- 2007 – väga suurte plokkide korduvkasutus (>1M, IP tuumad) – 600K
- 2009 – homogeenne paralleel-töötlus (multi-tuumad) – 1200K
- *Tänapäev ja tulevik – riist- ja tarkvara koosverifitseerimine, täidetav spetsifikatsioon jne.*

# Loogikasüntees – kriteeriumid

- **Jõudlus – Performance**
  - viide & taktsagedus – delay & cycle-time
  - hilistumine – latency
  - tootlus – throughput (konveier-rakendused)
- **Võimsustarve – Power consumption**
- **Pindala – Area**
  - mikroskeemi tootmise kasutegur (*yield*)
  - pakendamise maksumus
- **Testitavus – Testability**

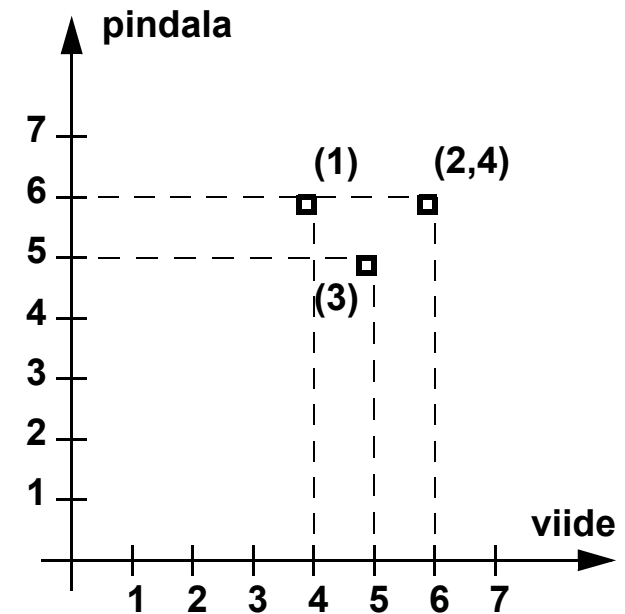
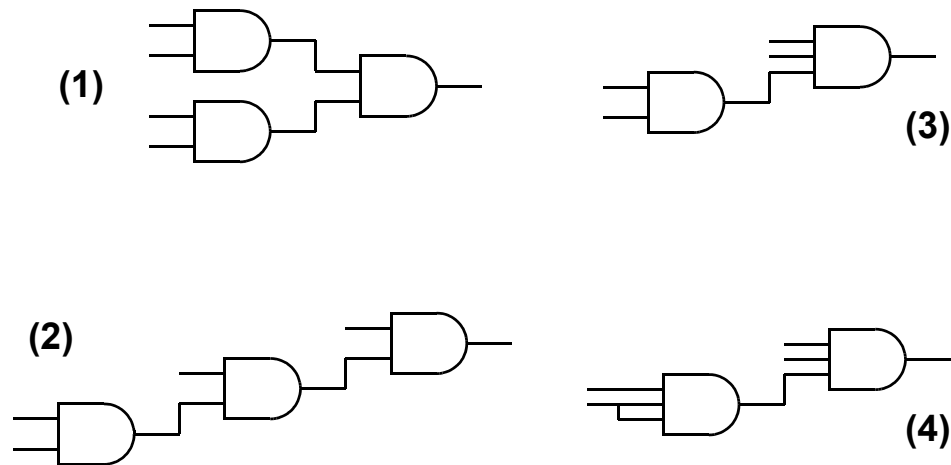
Valikuvõimalused  
(*Design Space Exploration*)





## Näide sünteesivõimalustest

- funktsioon –  $x = a b c d$
- loogikalülid – 2-AND & 3-AND ,  
pindala & viide võrdeline sisendite arvuga





# 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)

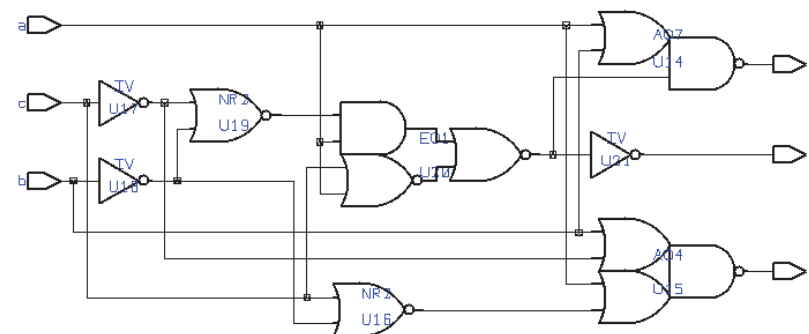
abc	xyz
111	101
-01	010
0-0	101
00-	011
0-1	010

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

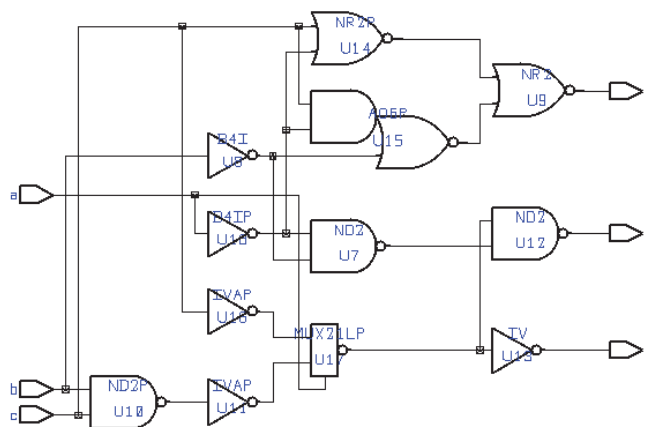
$$x = abc + \bar{a}\bar{c}$$

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

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

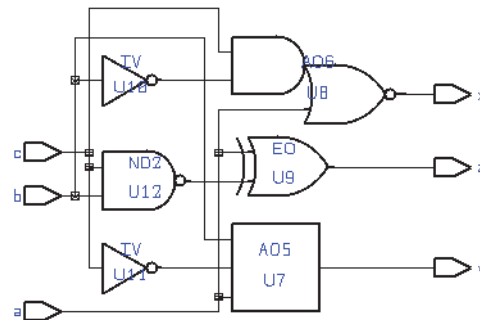


pindala viide  
12.0  
2.73 ns  
võimsus 11.3  $\mu$ W

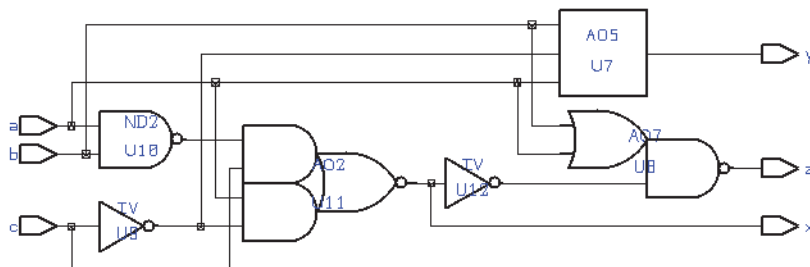


pindala viide  
22.0  
1.57 ns  
võimsus 24.1  $\mu$ W

## Loogikasünteesi eesmärgid (2)



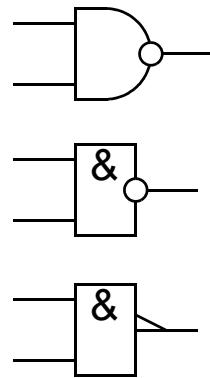
**pindala**      **11.0**  
**viide**        **1.84 ns**  
**võimsus**     **9.2  $\mu\text{W}$**



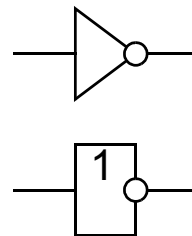
**pindala**      **10.0**  
**viide**        **2.26 ns**  
**võimsus**     **10.2  $\mu\text{W}$**

# Baas-element NAND (JA-EI)

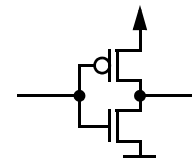
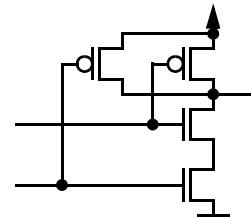
2-NAND



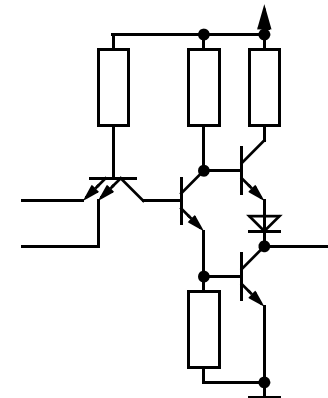
NOT



CMOS (Si)



TTL (Si)

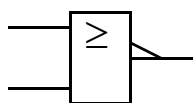
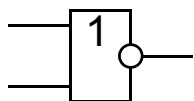
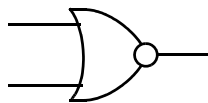




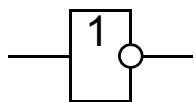
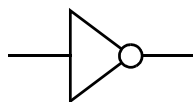


# Baas-element NOR (VÕI-EI)

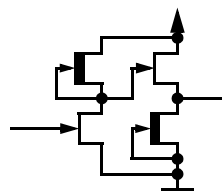
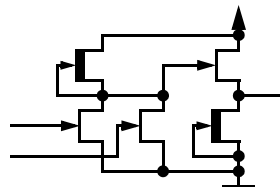
2-NOR



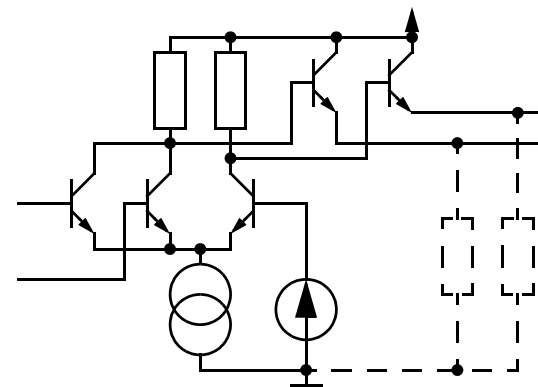
NOT



**DCFL (GaAs)**  
Direct-coupled FET



*ECL (Si)*

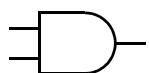




# Loogikaelemendid

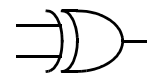
- **Lihtelemendid**

- **sisendite arv varieerub – 2 ... 4 (8), v.a. invertor (NOT) ja puhver (BUFF)**
- **koormatuvus – kuni ~10 (erijuhtudel rohkem)**
- **CMOS – 2 transistori sisendi kohta (n&p), vajadusel ka väljundpuhver (-inverter)**



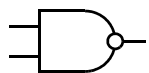
AND

$$o = a \cdot b$$



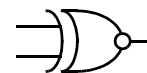
XOR

$$o = a \oplus b$$



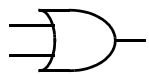
NAND

$$o = \overline{a \cdot b}$$



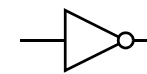
XNOR

$$o = a \oplus \bar{b}$$



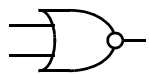
OR

$$o = a + b$$



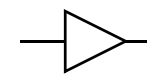
NOT

$$o = \bar{a}$$



NOR

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



BUFF

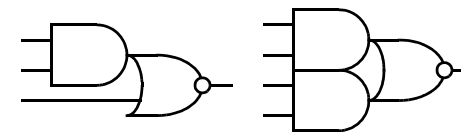
$$o = a$$

# Loogikaelemendid

- **Liitelemendid**

- tehnoloogiast sõltuv

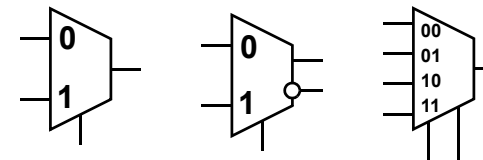
- CMOS – AND: n-tr. järjestikku, p-tr. paralleelselt; VÕI: vastupidi



- **MUX - multipleksor**

- Shannon'i arendus ühe või enama muutuja järgi

- $o = \bar{a}.i_0 + a.i_1$  /  $o = \bar{a}_1.\bar{a}_0.i_{00} + \bar{a}_1.a_0.i_{01} + a_1.\bar{a}_0.i_{10} + a_1.a_0.i_{11}$



- **Programmeeritav loogika**

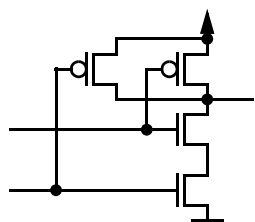
- FPGA – tõeväärtustabelid (4-6 sisendit & 1-2 väljundit)

- PLD / PLA – matriks-struktuur (implikandid)

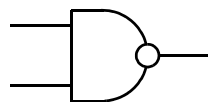
- **Vt. ka “Simulation of basic CMOS gates with explanatory text”**

- <http://tams-www.informatik.uni-hamburg.de/applets/cmos/cmosdemo.html>

# Kahendloogika pööratavus



2-NAND



AND

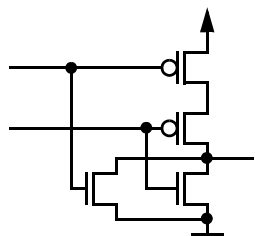
a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

 $0 \rightarrow 1 / 1 \rightarrow 0$ 

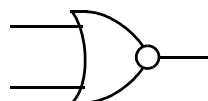
a	b	y
1	1	1
1	0	1
0	1	1
0	0	0

OR

a	b	y
0	0	0
0	1	1
1	0	1
1	1	1



2-NOR



NAND

a	b	y
0	0	1
0	1	1
1	0	1
1	1	0

 $0 \rightarrow 1 / 1 \rightarrow 0$ 

a	b	y
1	1	0
1	0	0
0	1	0
0	0	1

NOR

a	b	y
0	0	1
0	1	0
1	0	0
1	1	0



## Loogikafunktsioonide teisendamine

- **Loogikafunktsioonide minimeerimine / optimeerimine / valideerimine**
  - Eesmärk sobivaima esitusviisi, realisatsiooni jne. leidmine
- **Lähtekirjeldus – tabel, skeem, HDL**
- **Süntees ja optimeerimine**
  - lähtekirjelduse teisendamine abstraktseks mudeliks
  - teisendused abstraktsel mudelil
  - sidumine elementidega teegist
- **Abstraktsed mudelid**
  - põhinevad graafidel, sobivad analüüsiks ja masintöötluks
  - tuletatud lähtekirjeldusest kompileerimise teel
- **Optimeerimine**
  - esituse / realisatsiooni vähendamine / lihtsustamine
- **Valideerimine**
  - teisenduste korrektsuse kontroll – simuleerimine / verifitseerimine

## Esitusviisid ja teisendamine

- **Tõeväärtustabel**

- **1-piirkond**

- $f = \Sigma_{a,b,c} (0,3,6,7)_1 (4)_-$
- $f(a,b,c) = \Sigma (0,3,6,7)_1 (4)_-$

- **0-piirkond**

- $f = \Pi_{a,b,c} (1,2,5)_0 (4)_-$
- $f(a,b,c) = \Pi (1,2,5)_0 (4)_-$

- **Määramatused – väärtus on ebaoluline (-,\* )**

abc	f
000	1
001	0
010	0
011	1
100	-
101	0
110	1
111	1

abc	f
000	1
001	0
010	0
011	1
100	1
101	0
110	1
111	1

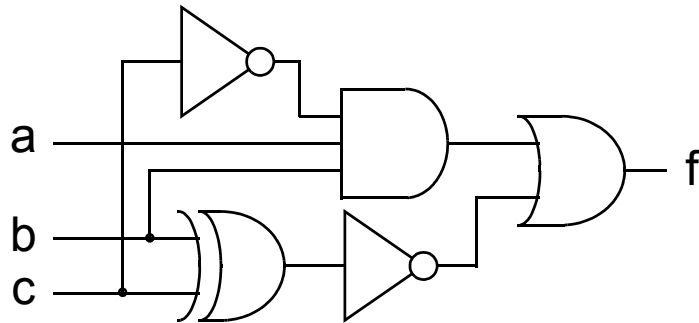
- **Loogika-avaldis ja normaalkujud**

- $f = \overline{(b \oplus c)} + abc$ ;  $f = b(a + c) + \overline{b}\overline{c}$ ;  $f = bc + \overline{c}(a + \overline{b})$ ;
- DNK:  $f = ab + bc + \overline{b}\overline{c}$ ;  $f = bc + a\overline{c} + \overline{b}\overline{c}$ ;  $f = ab + bc + a\overline{c} + \overline{b}\overline{c}$ ;
- KNK:  $f = (b + \overline{c})(a + \overline{b} + c)$ ;

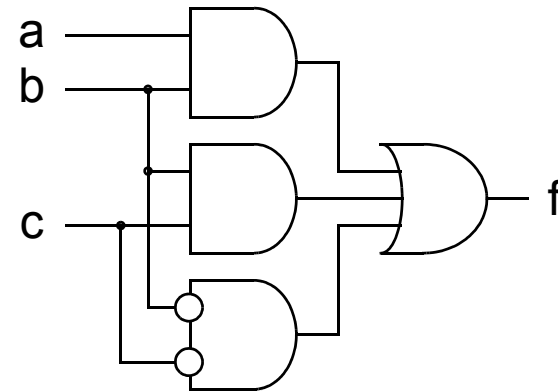


## Avaldis ja skeem

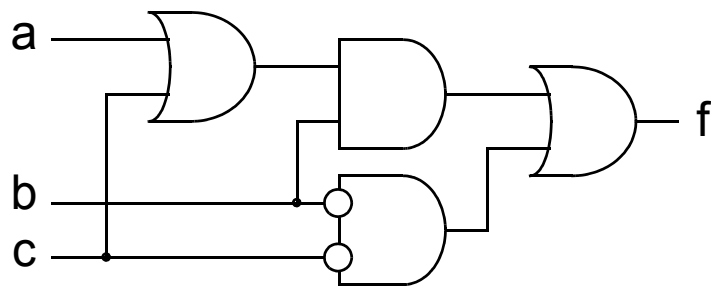
$$f = \overline{(b \oplus c)} + abc$$



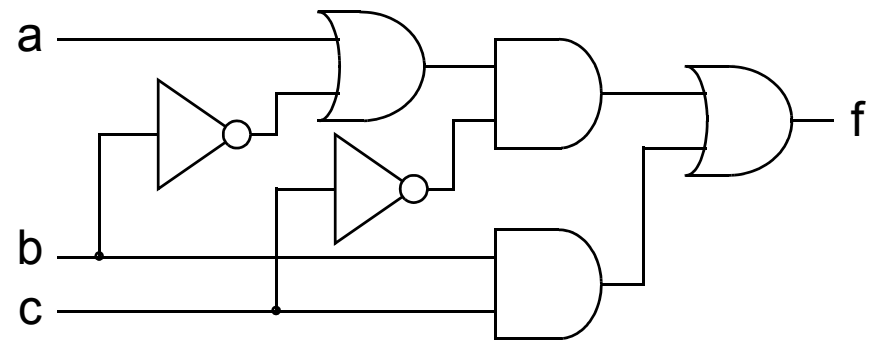
$$f = ab + bc + \bar{b}\bar{c}$$



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

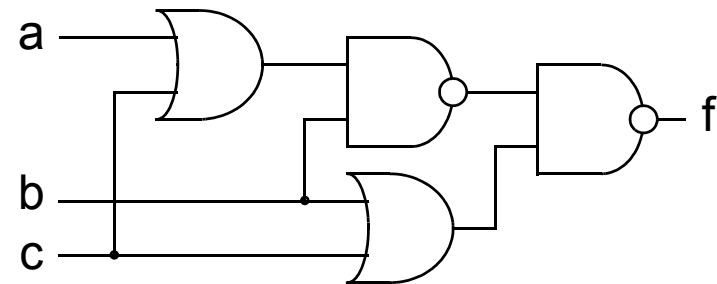
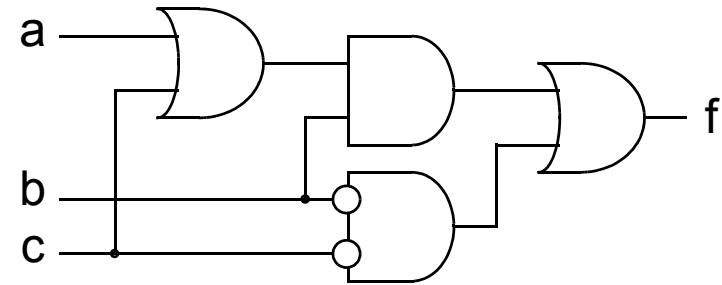
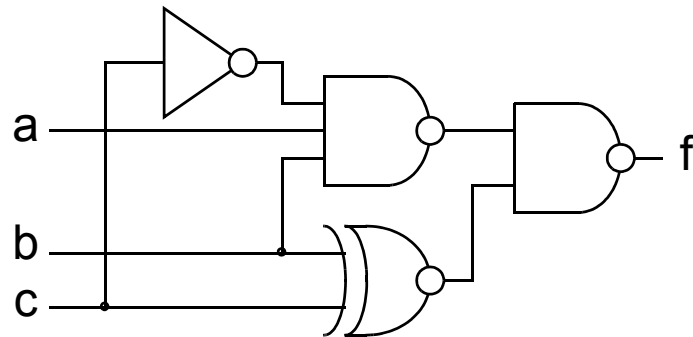
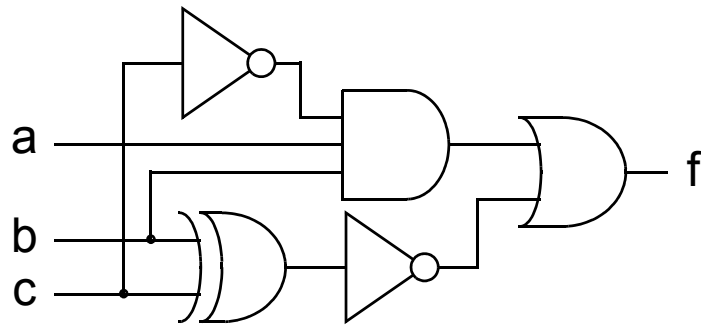


$$f = bc + \bar{c}(a + \bar{b})$$



## Skeemi optimaalsus?

- Kuidas alustada ja mida kasutada?





## 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

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

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

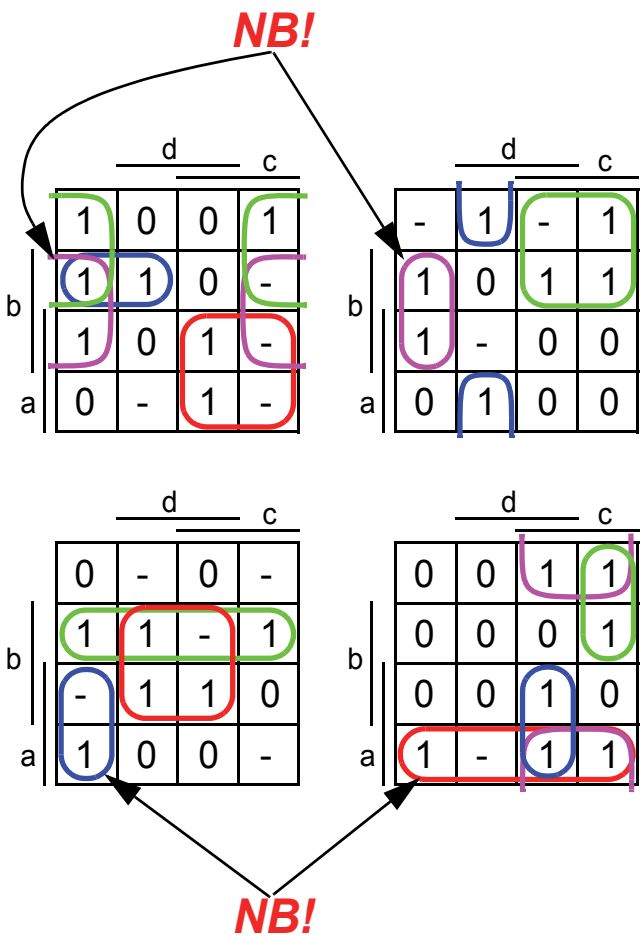
$$f = (a+b+\bar{d}) (\bar{a}+\bar{b}+d) (\bar{a}+\bar{b}+\bar{c})$$

DNK 0-dest  
eitus  
De Morgan!

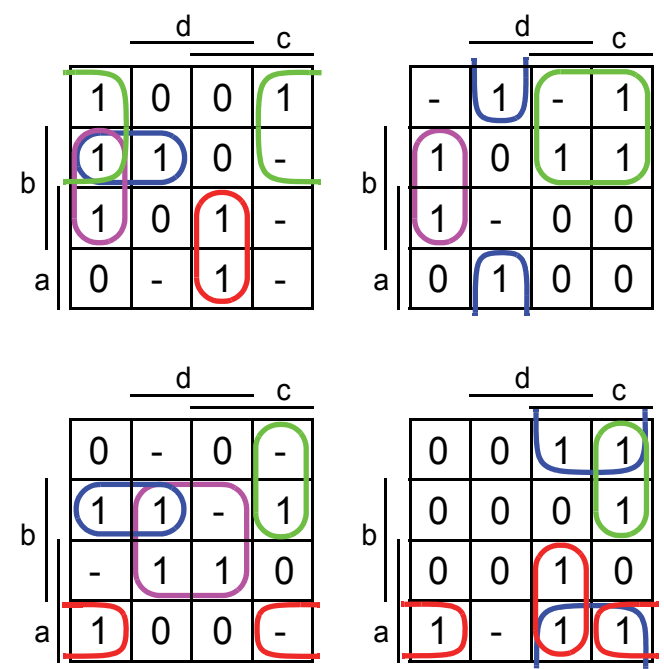


# Ülesanne – funktsioonide süsteemi minimeerimine

abcd	klmn
0--0	1000
010-	1000
-1-0	1000
1-1-	1000
0-1-	0100
-001	0100
-100	0100
-1-1	0010
01--	0010
1-00	0010
-01-	0001
10--	0001
0-10	0001
1-11	0001



abcd	klmn
0--0	1000
010-	1010
-100	1100
1-11	1001
0-1-	0100
-001	0100
-1-1	0010
01--	0010
1-00	0011
0-10	0011
-01-	0001



14 vs. 10 implikanti...



# Kahetasemeline minimeerimine ja mitmetasemeline realisatsioon

3 varianti

1: F, A, B

2: F, A, D

3: F, B, E

Lisaks

üksikult

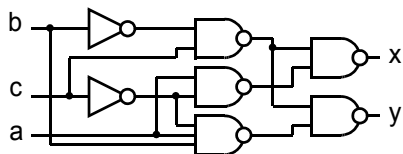
minimeeritud

4: A, E, D, F

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

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

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

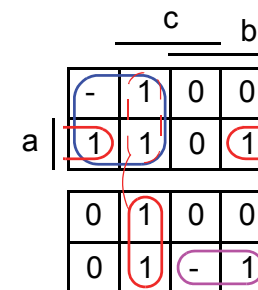
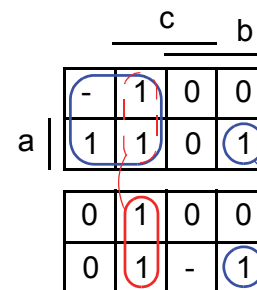
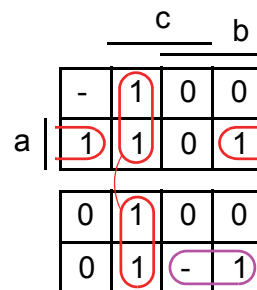
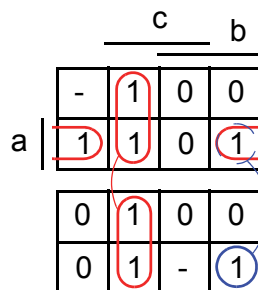


abc	xy
-01	11
1-0	10
110	<u>01</u>

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

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

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



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

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

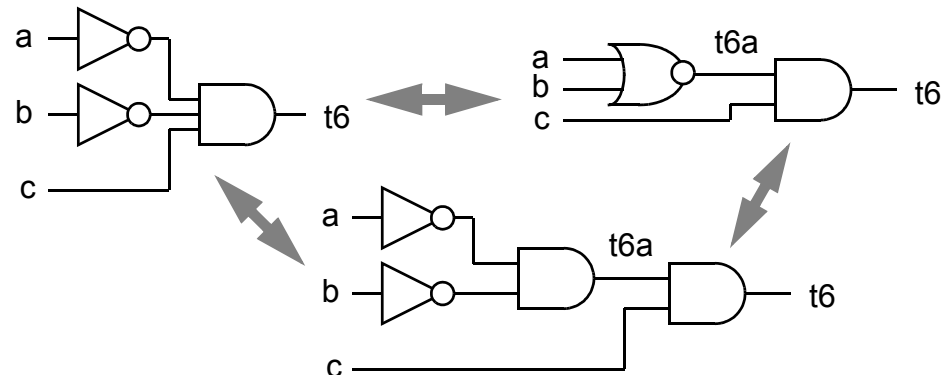
NOT - 2  
 2-NAND - 3  
 3-NAND - 1  
 22 transistori  
 [11 literaali]

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

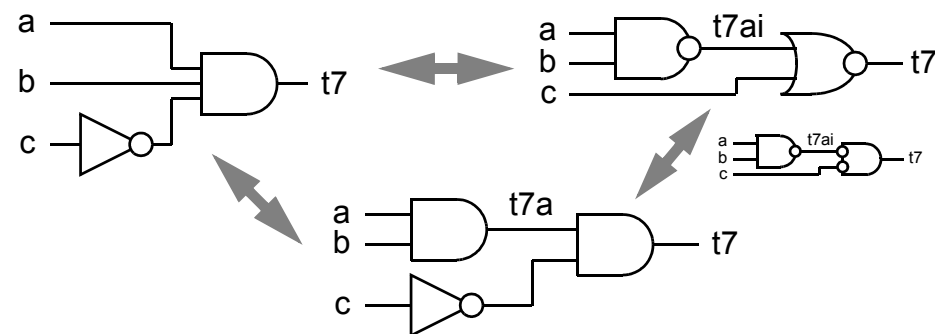
# Teisendused mitmetasemelisel realisatsioonil

## De Morgani & topelteiluse seadused

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

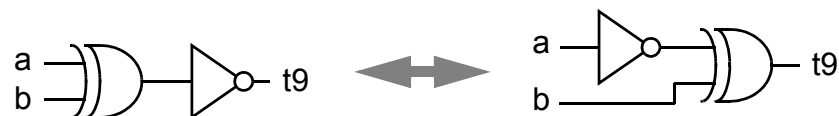


- $t7 = a b c'$  ;
- $t7a = a b$  ;  $t7 = t7a c'$  ;
- $[t7ai = (a b)'$  ;  $t7 = t7ai' c'$  ;]
- $t7ai = (a b)'$  ;  $t7 = (t7ai + c)'$  ;



a	b	$\bar{\oplus}$
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



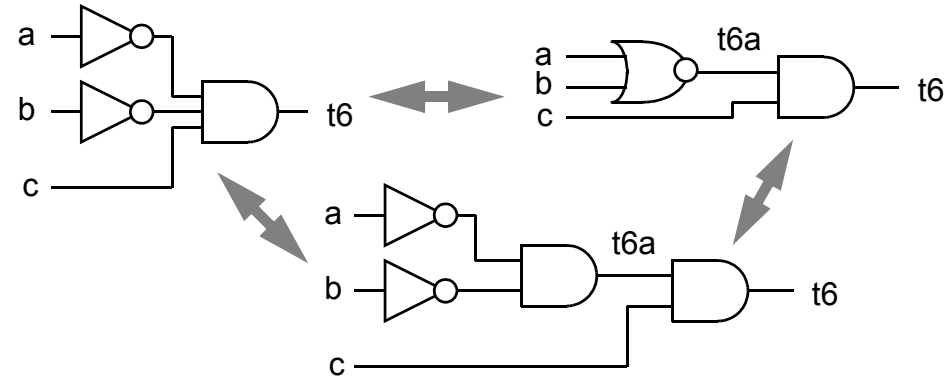
# Modelleerimine, mitmetasemeline realisatsioon, teisendused, signaali hilistumine...

```
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