



Digitaalsüsteemid / Digital Systems

IAS0150 – 6 EAP 4 2-2-0 E S

IAY0150 – 5 EAP 4 2-1-1 E S

<http://mini.pld.ttu.ee/~lrw/IAS0150/>

<https://ained.ttu.ee/course/view.php?id=189>

Peeter Ellervee

ICT-526

620 2258

511 3631

LRV@ati.ttu.ee

<http://www.ati.ttu.ee/~lrw/>

<http://mini.pld.ttu.ee/~lrw/>



Kursuse sisu

- **IAS0150**
 - 15 loengut ja praktikumi (pluss varuaeg)
 - kolm kodutööd
 - kirjalik eksam, eeldusteks esitatud kodutööd ja osalemine praktikumides (testid!)
 - koondhinne (max 100) – kodutööd 25+15+15, praktikumid 15, eksam 30
 - hinded: 1: 50-59, 2: 60-69, 3: 70-79, 4: 80-89, 5: 90-... (max 100)
- **IAY0150**
 - 15 loengut ja praktikumi (pluss varuaeg)
 - kaks kodutööd
 - kirjalik eksam, eeldusteks esitatud kodutööd ja osalemine praktikumides (testid!)
 - koondhinne (max 100) – kodutööd 25+15, praktikumid 15, eksam 30
 - hinded: 1: 42-50, 2: 51-59, 3: 60-67, 4: 68-76, 5: 77-... (max 85)

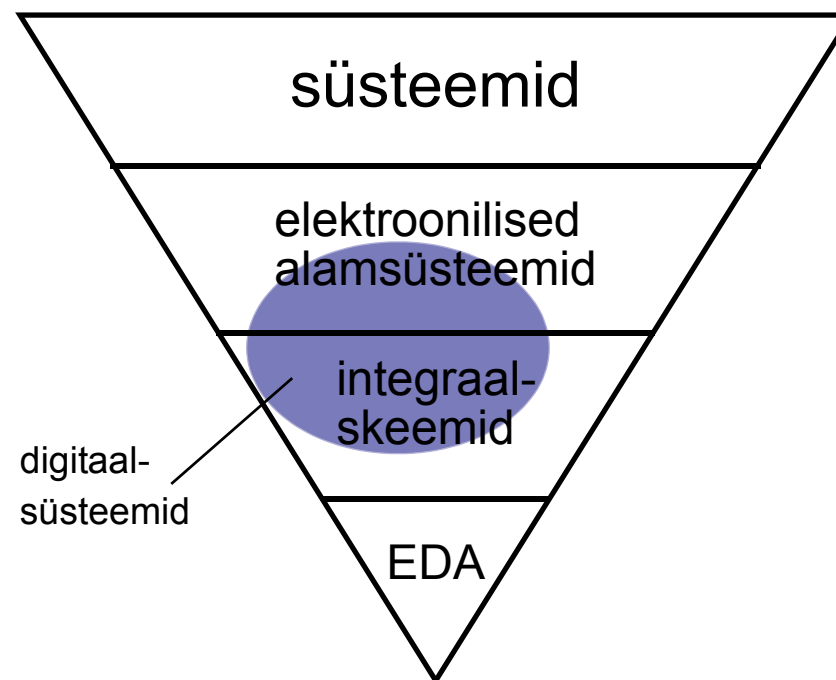


Digitaalsüsteemid

- **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, ...
 - Küberfüüsikalised süsteemid – süsteem + keskkond, kasutajad, ...

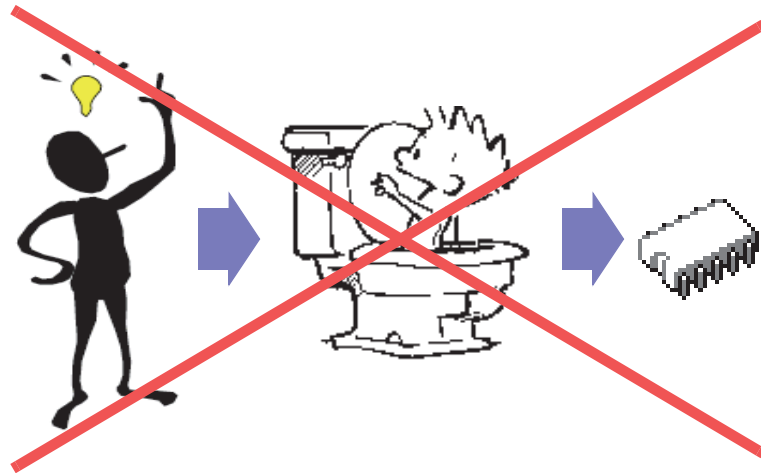
Digitaalsüsteemid

- **Digitaalsüsteem**
 - digitaalne andmetöötlus
 - andmeosa, juhtosa, sisend/väljund
- **Primaarturud**
 - infosüsteemid
 - telekommunikatsioon
 - laiatarbe-elektronika
- **Sekundaarturud**
 - süsteemid (nt. transport)
 - tootmine (nt. robotid)
- **VLSI tehnoloogiate rakendused**
VLSI – Very Large Scale Integration



Digitaalsüsteemide projekteerimine e. disain

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



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





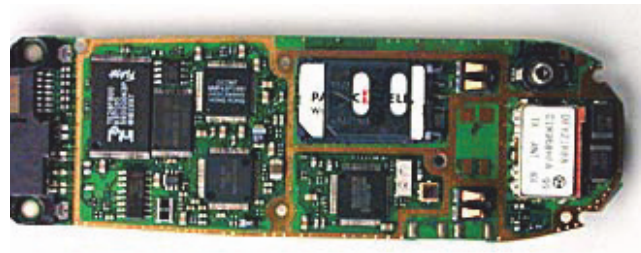
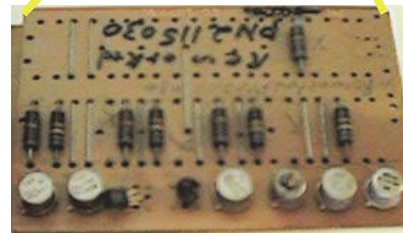
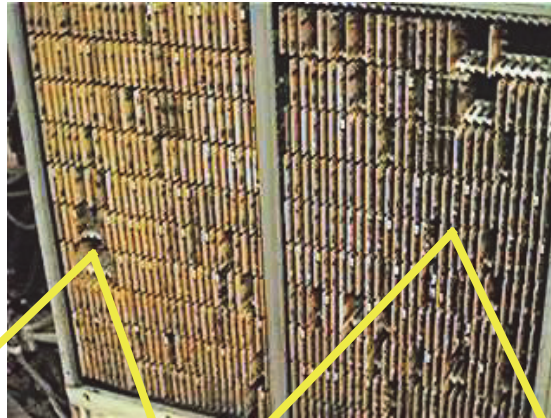
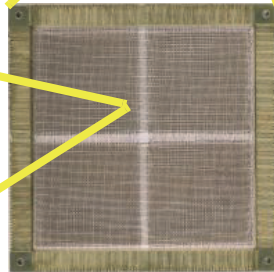
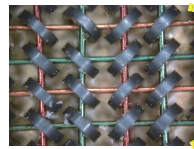
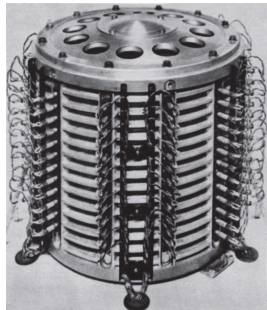
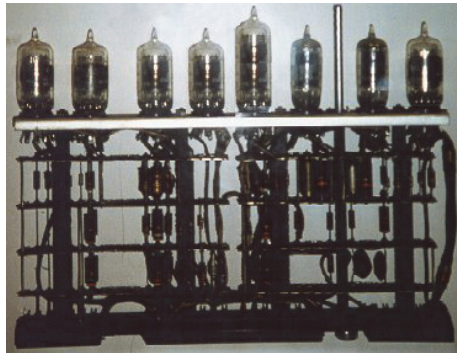
Disaini põhietapid

- **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

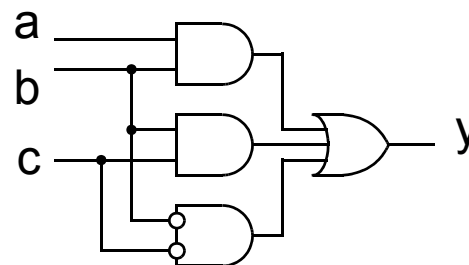
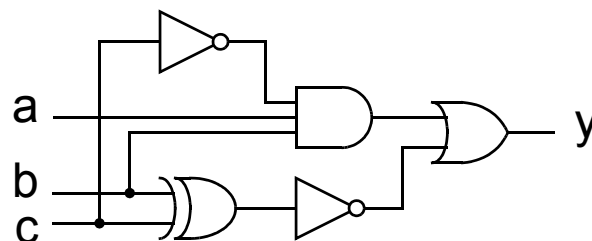
Tase	Abstraktsioon	Töövahendid
süsteem	käitumine ruumis ja ajas kui suuniste, ajastuse ja s/v spetsifikatsioonid	plokk-skeemid, diagrammid, kõrgtaseme (programmeerimis) keeled
arhitektuur	funktsionaalsete olemite üldine süsteem (organisatsioon)	riistvara kirjelduskeeled, moodulite paigalduse planeerimine 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)

Digitaalsüsteemide realiseerimine – ajalugu ja tänapäev



Algoritmist skeemini

- **Peatuspunkt / sünkroniseerimine**
 - olek (automaadi olek)
 - mäluelemendid
 - takteerimine
- **Funktsioon**
 - $y = \overline{(b \oplus c)} + a b \bar{c}$
- **Normaalkuju**
 - $y = a b + b c + \bar{b} \bar{c}$
- **Tõeväärtustabel**



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

Algoritmist skeemini (2)

- Lähteandmed**

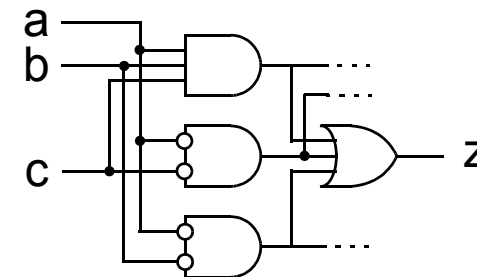
- $x(a,b,c) = \bar{a} \bar{b} \bar{c} + \bar{a} b \bar{c} + \bar{a} b c + a b c$
- $y(a,b,c) = \bar{a} \bar{b} \bar{c} + \bar{a} \bar{b} c + \bar{a} b \bar{c} + a \bar{b} c$
- $z(a,b,c) = \bar{a} \bar{b} \bar{c} + \bar{a} \bar{b} c + \bar{a} b \bar{c} + a b c$

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

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

- Lõpptulemus**

- $x(a,b,c) = \bar{a} b c + a b c + \bar{a} \bar{c}$
- $y(a,b,c) = \bar{b} c + \bar{a} b c + \bar{a} \bar{b}$
- $z(a,b,c) = a b c + \bar{a} \bar{c} + \bar{a} \bar{b}$



Algoritmist skeemini (3)

- Lõplik funktsioon**

$$j = a' + b + c'$$

$$k = c + d$$

$$q = a + b$$

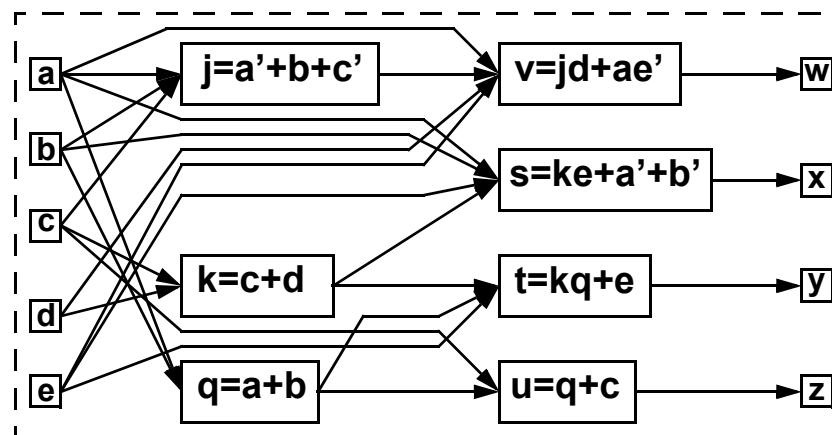
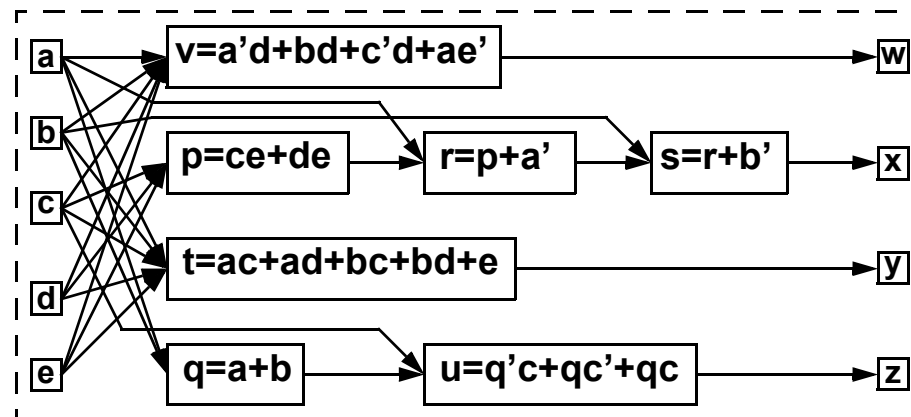
$$s = ke + a' + b'$$

$$t = q + c$$

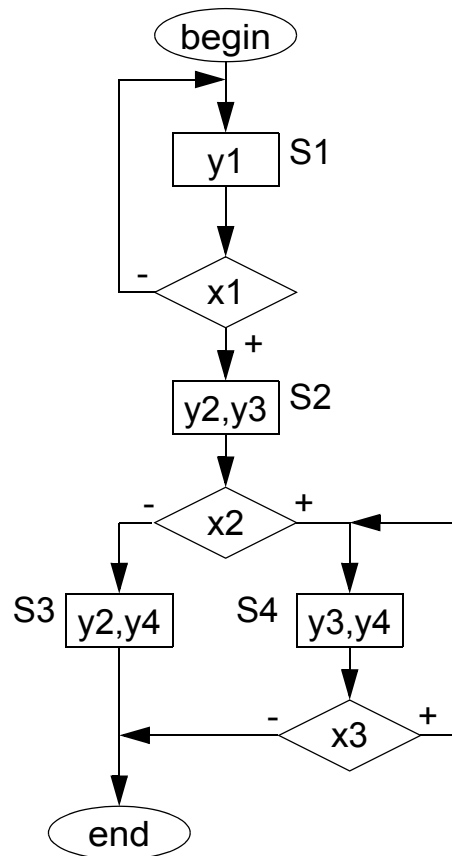
$$u = q + c$$

$$v = jd + ae'$$

- Funktsioone/loogikalülisid – 7 ja 7**
- Literaale – 33 ja 20**
- Viide – 3 ja 2 (sõlmi)**
- Viide – 9 ja 7 (sõlmi+literaale)**



Algoritmist skeemini (4)



i^t	s^t	$q1\ q0$	s^{t+1}	$q1\ q0$	o^t
$x1'$	S1	0 0	S1	0 0	y1
$x1$			S2		
$x2'$	S2	0 1	S3	1 1	y2,y3
$x2$			S4		
-	S3	1 1	S1	0 0	y2,y4
$x3'$	S4	1 0	S1	0 0	y3,y4
$x3$			S4		

Skeem

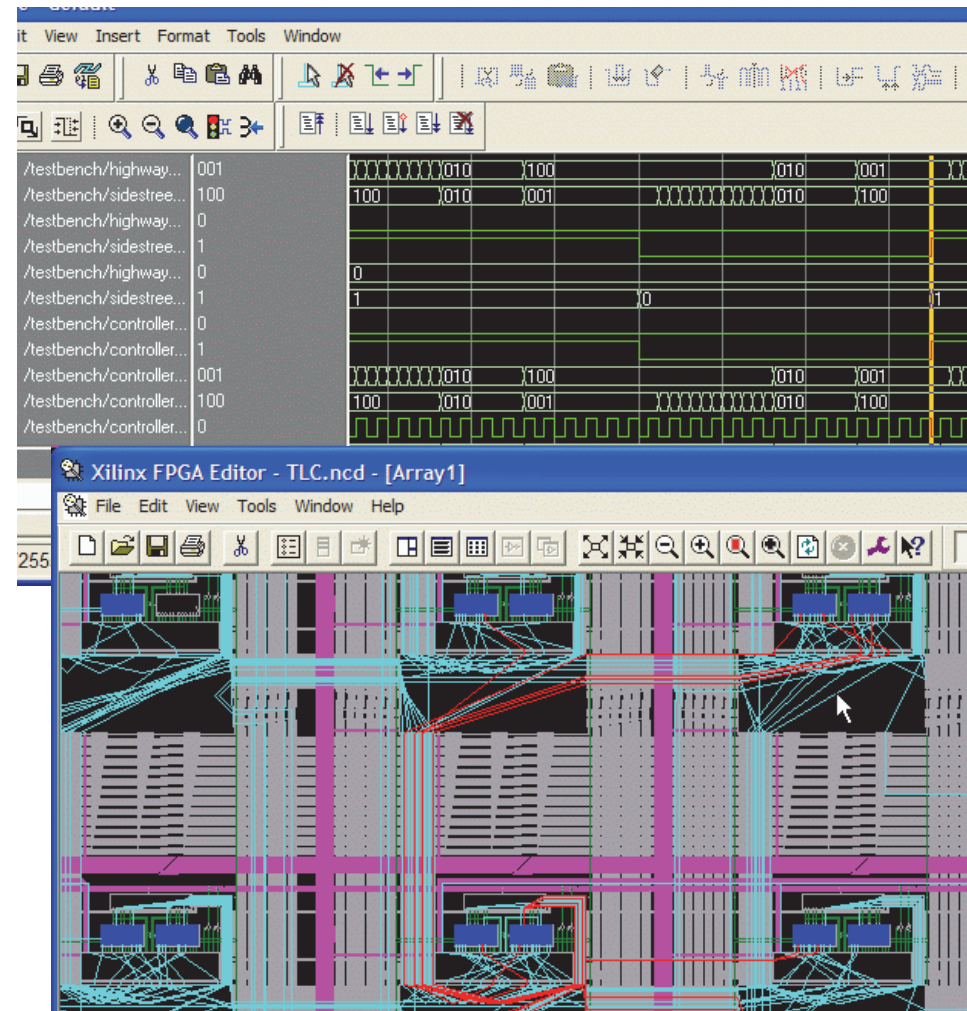
- $n0 = \overline{q1}q0$
- $d1 = n0 + x3n1$
- $y1 = \overline{q1}\ \overline{q0}$
- $y3 = n0 + n1$

- $n1 = q1\overline{q0}$
- $d0 = x1y1 + \overline{x2}n0$
- $y2 = q0$
- $y4 = q1$

Projekteerimine tänapäeval

- Riistvara kirjelduskeel**

```
--
-- Highway is green, sidestreet is red.
--
if  sidestreet_car = NoCar  then
    wait until  sidestreet_car = Car;
end if;
-- Waiting for no more than 25 seconds ...
if  highway_car = Car  then
    wait until  highway_car = NoCar  for 25 sec;
end if;
-- ... and changing lights
highway_light <= GreenBlink;
wait for 3 sec;
highway_light <= Yellow;
sidestreet_light <= Yellow;
wait for 2 sec;
highway_light <= Red;
sidestreet_light <= Green;
```





Turg e. \$\$\$

- **Projekteerimise maksumus**
 - *projekteerimisaeg & kristallide tootmise hind*
 - *suured kapitalimahutused*
 - *pea-aegu võimatu parandada*
- **Muudatuste kõrge hind**
 - *suured tootmismahud rentaablimad*
 - *null-defekti on äärmiselt oluline*
 - *turusuundumuste järgimine oluline*
- **Hind pöördvõrdeline tootmismahuga**
 - *üldotstarbelised protsessorid - odav kuid pole alati kasutatav*
 - **ASIC (Application-Specific Integrated Circuits)** – häälestamine vastavalt vajadusele (nt. telekommunikatsioon)
 - *prototüübid – väljatöötluses on paindlikkus äärmiselt oluline*
 - *spetsrakendused (nt. sateliidid)*
- **Rekonfigureeritavus**
 - *paindlikud tooted, võimalus modifitseerida töötavat skeemi*



Automatiseeritud projekteerimine

- **Gordon Moore seadus (1965)**
- **Edusammud tehnoloogias**
 - väiksemad skeemid
 - suurem jõudlus
 - rohkem transistore kristallil
- **Suurem integratsiooniaste**
 - kompleksemad süsteemid
 - arvutusvõimsuse odavnemine
 - suurem töökindlus
- **Automatiseerimine võimaldab:**
 - uusimate tehnoloogiate kasutamist
 - vähendada projekteerimiskulusi
 - kiirendada projekteerimist



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.*



Miks kahendloogika?

- **Digitaali eelised**
 - **Tulemuste korratavus**
 - samad sisendväärtused annavad alati sama tulemuse
 - analoog – temperatuur, toitepinge, vananemine, ...
 - **Projekteerimise lihtsus**
 - loogikafunktsioonid, optimeerimisalgoritmid
 - **Paindlikkus ja funktsionaalsus**
 - erinevad algoritmid, sama funktsionaalsus (võimsustarve, kiiruse, suuruse jne. erinevused)
 - **Programmeeritavus**
 - programmeerimiskeeled / riistvara kirjelduskeeled
 - **Töökiirus**
 - **Turu ja tehnoloogia areng**
 - ränikiipide/-tehnoloogia skaleeritavus/korratavus



Miks kahendloogika?

- **Analoogi eelised**
 - Differentsiaalvõrrandite realiseerimine
 - Energeetiline efektiivsus
 - Kõrge töösagedus
- **Nivoo-loogika**
 - Kindel pingevahemik (vooluvahemik) – 0: $<0,8V$, 1: $>3,8V$
- **Mitmevalentne loogika**
 - Rohkem kui kaks diskreetset väärtust
 - Suurem infotihedus – nt. 4- ja 16-valentsed mälud
 - Boole'i algebra edasiarendus
 - funktsioonide süsteemi minimeerimine



Hulgad – Sets

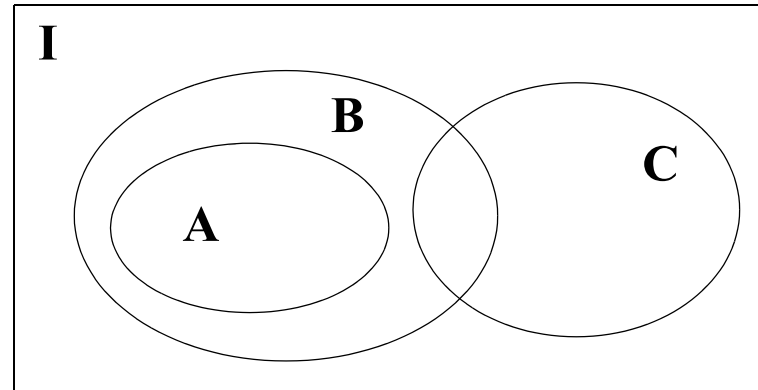
- *hulk on elementide* (objektide) kogum [Georg Cantor, 1845-1918]
 - hulk – set, element – *element, member*
- $x \in A$ – element x kuulub hulka A
- $|A|$, $N(A)$ – hulga võimsus (kardinaalsus / cardinality)
 - lõplikud (finite) ja lõpmatud (infinite) hulgad
 - loenduvad (countable) hulgad
- $P \subseteq Q$ – alamhulk (subset) – hulk P on hulga Q alamhulk kui iga hulga P element on ka hulga Q element
- ühisosata (disjunksed) hulgad (disjoint sets)
- tühihulk (empty set) – $\emptyset$, universaalhulk (universal set) – I
- 2^A , $P(A)$ – astmehulk (potentshulk / power set) – hulga A kõigi alamhulkade hulk

Tehted hulkadega

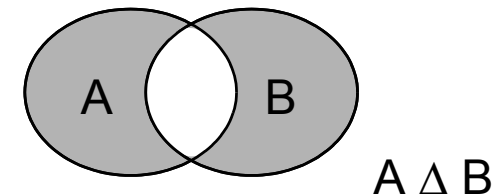
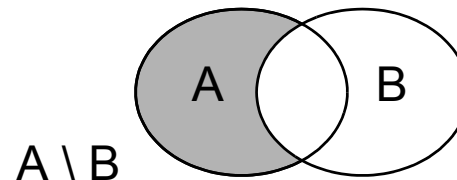
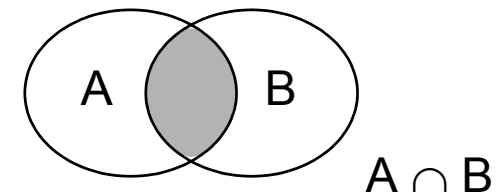
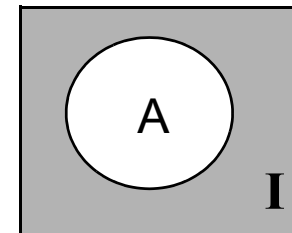
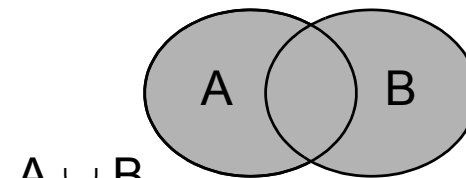
- $A \times B = \{ (a,b) \mid a \in A \ \& \ b \in B \}$ – otsekorrutis (Descartes'i korrutis, Cartesian product)
 - $(a_1, a_2), \langle a_1, a_2 \rangle$ - järjestatud paar (ordered pair, ordered 2-tuple)
- $A \cup B = \{ x \mid x \in A \vee x \in B \}$ – hulkade ühend (union)
- $A \cap B = \{ x \mid x \in A \ \& \ x \in B \}$ – hulkade ühisosa (lõige, intersection)
- $\overline{A} = \{ x \mid x \in I \ \& \ x \notin A \}$ – hulga täiend (complement, A^C)
- $A \setminus B = \{ x \mid x \in A \ \& \ x \notin B \}$ – hulkade vahe (difference, relative complement), $(\overline{A} = I \setminus A)$
- $A \triangle B = \{ x \mid (x \in A \ \& \ x \notin B) \vee (x \notin A \ \& \ x \in B) \}$ – hulkade sümmeetriline vahe

Venn'i diagramm

John Venn, 1834-1923



- $A \subseteq B$
- $A \cap C = \emptyset$
- $B \cap C \neq \emptyset$
- $A \cup B$
- $A \cap B$
- $\bar{A}$
- $A \setminus B$
- $A \Delta B$



Operatsioonide omadused

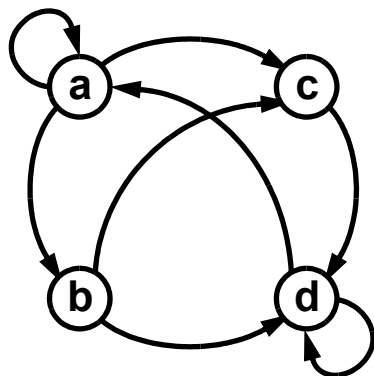
- **Kommutatiivsus (vahetuvus, commutativity)** $A \cup B = B \cup A$ $A \cap B = B \cap A$
- **Assotsiatiivsus (ühenduvus, associativity)**
- $A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$
- **Distributiivsus (jaotuvus, distributivity)**
- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
- **De Morgani seadused** $\overline{A \cup B} = \overline{A} \cap \overline{B}$ $\overline{A \cap B} = \overline{A} \cup \overline{B}$
- **Idempotentsusseadus (idempotency)** $A \cap A = A \cup A = A$
- **Välistatud kolmanda seadused** $A \cup \overline{A} = I$ $A \cap \overline{A} = \emptyset$
- **Topelttäiendi seadus** $\overline{\overline{A}} = A$
- $A \cap \emptyset = \emptyset$ $A \cap I = A$ $A \cup \emptyset = A$ $A \cup I = I$
- **Neeldumisseadused**
- $A \cup (A \cap B) = A$ $A \cup (\overline{A} \cap B) = A \cup B$ $A \cap (A \cup B) = A$ $A \cap (\overline{A} \cup B) = A \cap B$
- **Kleepimisseadused** $(A \cap B) \cup (A \cap \overline{B}) = A$ $(A \cup B) \cap (A \cup \overline{B}) = A$
- $A \setminus B = A \cap \overline{B}$
- $A \Delta B = (A \setminus B) \cup (B \setminus A) = (A \cup B) \setminus (A \cap B)$

Vastavused – Functions (Relations)

- Antud 2 hulka A ja B ning reegel, kuidas hulka A elemendid on vastavuses φ hulga B elementidega – $\varphi \subseteq A \times B$ $\varphi : A \rightarrow B$

Binaarsuhted – Binary Relations

- Vastavuse φ erijuhtu, kus lähte- ja sihthulk langevad kokku – $D(\varphi) = R(\varphi) = A$
- Tähistus – $R \subseteq A \times A$
- Mugav interpreteerida suunatud graafina
 - hulga A elemendid vastavad tippudele
 - seosed elementide vahel vastavad kaarte
- Binaarmatriks (naabrusmaatriks)



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



Binaarsuhete R omadused

Näite-
graaf

- | | |
|---|-----|
| • Refleksiivsus (α_1) – ($\forall a \in A$ [$\langle a, a \rangle \in R$]) | ei |
| • Antirefleksiivsus (α_2) – ($\forall a \in A$ [$\langle a, a \rangle \notin R$]) | ei |
| • Suhe, mis ei täida nõudeid α_1 ega α_2 , on mitterefleksiivne | jah |
| • Sümmeetria (α_3) – ($\forall a, b \in A$ [$\langle a, b \rangle \in R \rightarrow \langle b, a \rangle \in R$]), kus $a \neq b$ | ei |
| • Antisümmeetria (α_4) – ($\forall a, b \in A$ [$\langle a, b \rangle \in R \rightarrow \langle b, a \rangle \notin R$]), kus $a \neq b$ | jah |
| • Suhe, mis ei täida nõudeid α_3 ega α_4 , on mittesümmeetriline | - |
| • Transitiiivsus (α_5) – ($\forall a, b, c \in A$ [($\langle a, b \rangle \in R$ & $\langle b, c \rangle \in R$) $\rightarrow$ $\langle a, c \rangle \in R$]), kus $a \neq b$, $b \neq c$, $a \neq c$ | ei |
| • Antitransitiiivsus (α_6) – ($\forall a, b, c \in A$ [($\langle a, b \rangle \in R$ & $\langle b, c \rangle \in R$) $\rightarrow$ $\langle a, c \rangle \notin R$]), kus $a \neq b$, $b \neq c$, $a \neq c$ | ei |
| • Suhe, mis ei täida nõudeid α_5 ega α_6 , on mittetransitiivne | jah |



Boole'i algebra

- signatuur koosneb 2 binaarsest ja ühest unaarsest operatsioonist $(+, \cdot, \bar{})$
- $+$ ja $\cdot$ on kommutatiivsed, assotsiatiivsed, idempotentsed ning teineteise suhtes distributiivsed
- eksisteerivad elemendid 0 ja 1 nii, et $x \cdot \bar{x} = 0$ ning $x + \bar{x} = 1$
- **Näited**
 - $\{2^A, \cap, \cup, \bar{}\}$ – Cantori algebra
 - $\{(0,1)^n, \&, \vee, \bar{}\}$ – loogikaalgebra
- Kaks algebrat on isomorfseid ($A_1 = \langle M_1, S_1 \rangle \equiv A_2 = \langle M_2, S_2 \rangle$), kui eksisteerib üksühene vastavus φ nii, et $\varphi: (M_1 \cup S_1) \Leftrightarrow (M_2 \cup S_2)$, kus $f_i(m_{j1}, \dots, m_{jk-1}) = m_{jk} \Leftrightarrow j(f_i)(j(m_{j1}), \dots, j(m_{jk-1})) = j(m_{jk})$, $m_{jl} \in M_1, j(m_{jl}) \in M_2, f_i \in S_1, j(f_i) \in S_2$

Cantori algebra ja loogikaalgebra on isomorfseid



Loogikafunktsioonid

- **$f(x_1, x_2, \dots, x_n)$** , kus
 - nii argumendid kui funktsiooni väärtus kuuluvad hulka $\{0,1\}$
 - iga loogikafunktsiooni võib esitada tõeväärtustabelina
- Erinevate loogikafunktsioonide **$f(x_1, x_2, \dots, x_n)$** arv K on 2^{2^n}
 - $n=1 \rightarrow K=4$
 - $n=2 \rightarrow K=16$
 - $n=3 \rightarrow K=256$
 - $n=4 \rightarrow K=65536$
 - $n=5 \rightarrow K=4,3 \cdot 10^9$

Kõikvõimalikud kahe muutuja funktsioonid $f(x_1, x_2)$

a	b	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

f_0 – konstant 0

f_2 – implikatsiooni eitus $\overline{a \rightarrow b}$

f_4 – pöördimplikatsiooni eitus $\overline{b \rightarrow a}$

f_6 – summa mooduliga 2, $a \oplus b$

f_8 – Pierce'i funktsioon, $\overline{a \vee b}$ ehk $a|b$

f_{10} – argumendi inversioon $\overline{b}$

f_{12} – argumendi inversioon $\overline{a}$

f_{14} – Shefferi funktsioon, $\overline{a \& b}$ ehk $a \downarrow b$

f_1 – konjunktsioon, $a \& b$ ehk $a \cdot b$ ehk ab

f_3 – argumendi a väärtus

f_5 – argumendi b väärtus

f_7 – disjunktsioon, $a \vee b$ ehk $a + b$

f_9 – samaväärsusfunktsioon, $a \leftrightarrow b$

f_{11} – pöördimplikatsioon $b \rightarrow a$

f_{13} – implikatsioon $a \rightarrow b$

f_{15} – konstant 1

- Üldlevinud prioriteedid – $\overline{}$, $\&$, $\vee$, $\rightarrow$, $\leftrightarrow$



Loogika põhiseadused

- **Idempotentsus** – $a \& a = a$ $a + a = a$
- **Kommutatiivsus** – $a \& b = b \& a$ $a + b = b + a$
- **Assotsiatiivsus** – $(a \& b) \& c = a \& (b \& c)$ $(a + b) + c = a + (b + c)$
- **Distributiivsus** – $a \& (b + c) = (a \& b) + (a \& c)$ $a + (b \& c) = (a + b) \& (a + c)$
- **Topelteitus** – $\overline{\overline{a}} = a$
- **De Morgan** – $\overline{a \& b} = \overline{a} + \overline{b}$ $\overline{a + b} = \overline{a} \& \overline{b}$
- **Kleepimine** – $(a \& b) + (a \& \overline{b}) = a$ $(a + b) \& (a + \overline{b}) = a$
- **Neeldumine** – $a + (a \& b) = a$ $a \& (a + b) = a$ $a + (\overline{a} \& b) = a + b$ $a \& (\overline{a} + b) = a \& b$
- **Konstandid** – $a + \overline{a} = 1$ $a \& \overline{a} = 0$ $a \& 0 = 0$ $a + 0 = a$ $a \& 1 = a$ $a + 1 = 1$
- **Lisateisendusi** – $a \rightarrow b = \overline{a} + b$ $a \oplus b = (a \& \overline{b}) + (\overline{a} \& b)$ $a \leftrightarrow b = (a \& b) + (\overline{a} \& \overline{b})$

Normaalkujud

- **Kanoonilised standardised esitusvalemid – normaalkujud**
 - **Disjunktiiivne normaalkuju (DNK, DNF) – elemantaarkonjunktsioonide disjunktsioon**
 - Elemantaarkonjunktsioon koosneb argumentide ja/või nende inversioonide konjunktsioonist
 - **Konjunktiiivne normaalkuju (KNK, CNF) – elemantaardisjunktsioonide konjunktsioon**
 - Elemantaardisjunktsioon koosneb argumentide ja/või nende inversioonide disjunktsioonist
 - Iga funktsioon on esitatav DNK ja KNK kujul, kuid mitte üheselt
- **Täielik DNK (TDNK, CDNF) – iga elemantaarkonjunktsiooni pikkus on n**
(st. iga elemantaarkonjunktsioon sisaldab funktsiooni kõiki argumente)
- **Täielik KNK (TKNK, CCNF) – iga elemantaardisjunktsiooni pikkus on n**
(st. iga elemantaardisjunktsioon sisaldab funktsiooni kõiki argumente)
- **Igal funktsioonil on täpselt üks TDNK ja üks TKNK**



Esitusviisid

- Funktsioon – $f = (a \leftrightarrow c) + b = (\bar{a} \bar{c}) + (a \& c) + b$

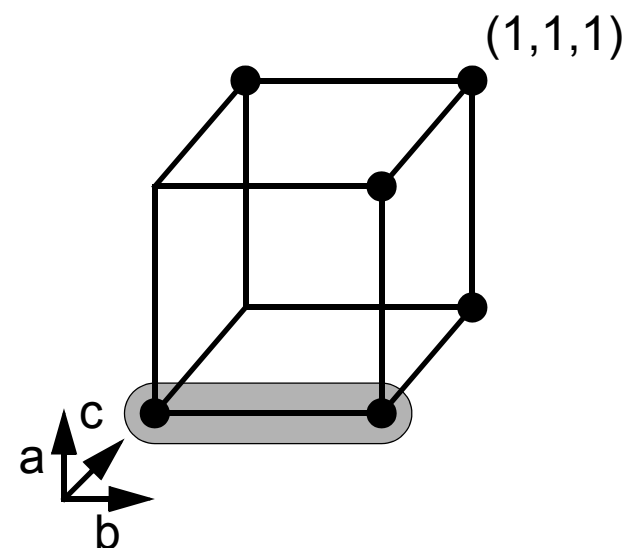
tõeväärtustabel

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

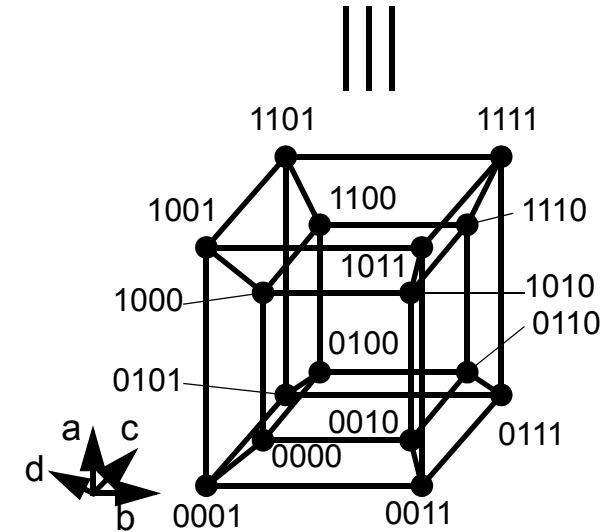
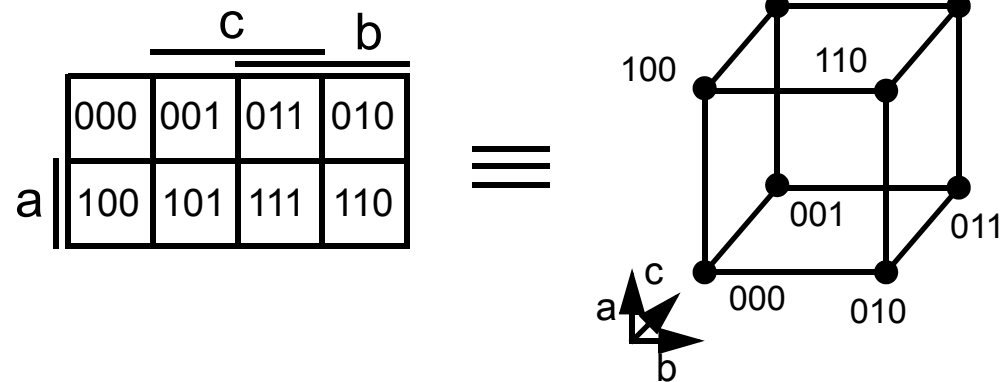
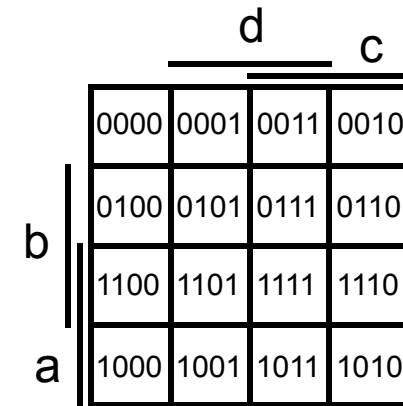
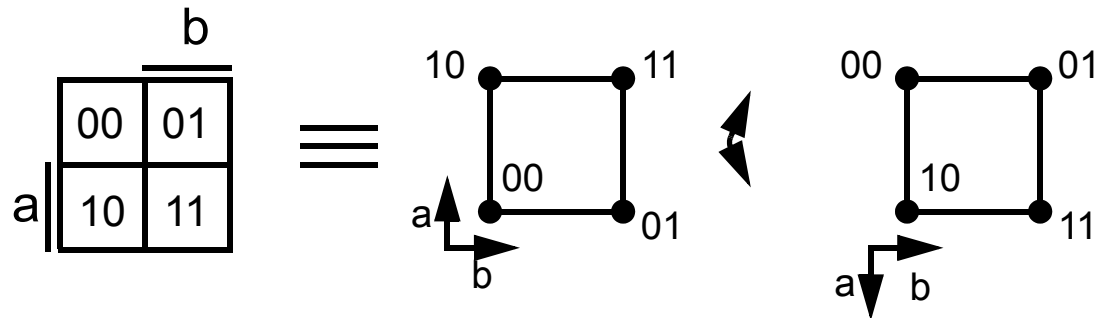
Karnaugh' kaart

	b		c	
	1	1	1	0
a	0	1	1	1

hüper-kuup



Esitusviisid





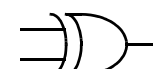
Loogikaelemendid

- 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 (-invertor)


 AND

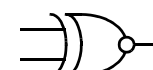
$$o = a \cdot b$$


 XOR

$$o = a \oplus b$$


 NAND

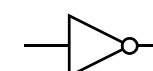
$$o = \overline{a \cdot b}$$


 XNOR

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


 OR

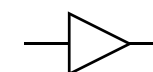
$$o = a + b$$


 NOT

$$o = \bar{a}$$


 NOR

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

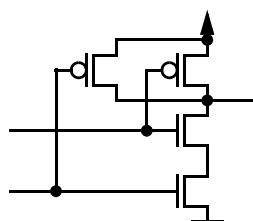

 BUFF

$$o = a$$

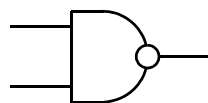
- Pluss hulk komplekseid elemente**

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

Kahendloogika pööratavus



2-NAND



AND

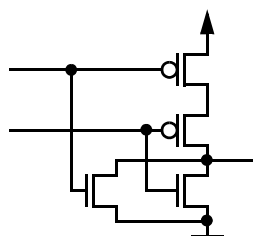
a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

 $0 \rightarrow 1 / 1 \rightarrow 0$

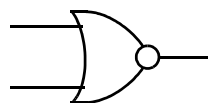
a	b	y
1	1	1
1	0	1
0	1	1
0	0	0

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

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

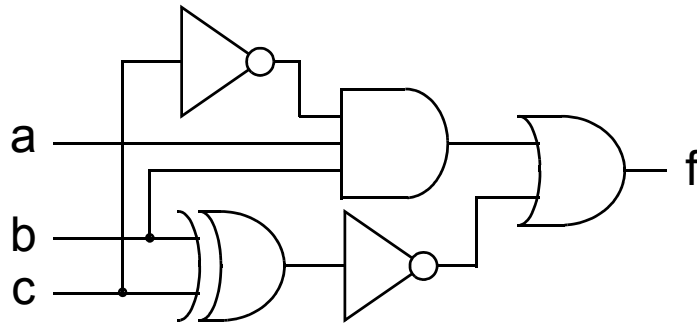
abc	f
000	1
001	0
010	0
011	1
100	1
101	0
110	1
111	1

- Loogika-avaldis ja normaalkujud**

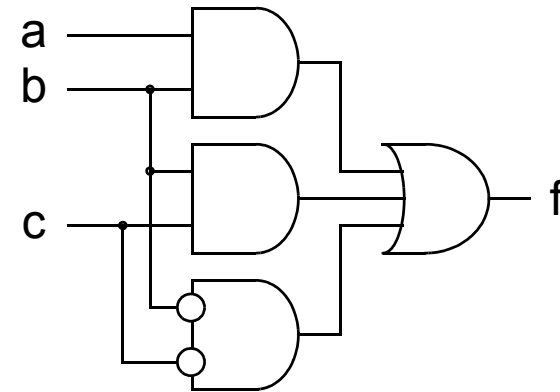
- $f = \overline{(b \oplus c)} + a b \bar{c}; \quad f = b(a + c) + \bar{b} \bar{c}; \quad f = b c + \bar{c}(a + \bar{b});$
- DNK: $f = a b + b c + \bar{b} \bar{c}; \quad f = b c + a \bar{c} + \bar{b} \bar{c}; \quad f = a b + b c + a \bar{c} + \bar{b} \bar{c};$
- KNK: $f = (b + \bar{c})(a + \bar{b} + c);$

Avaldis ja skeem

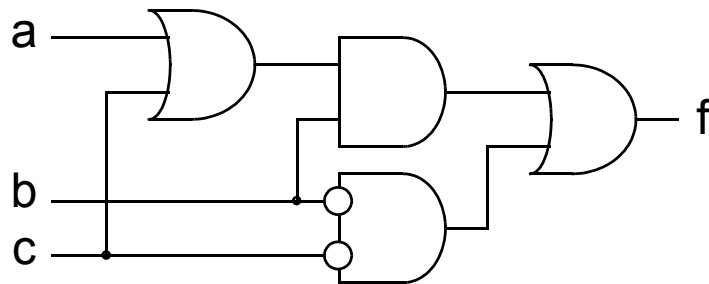
$$f = \overline{(b \oplus c)} + a b \bar{c}$$



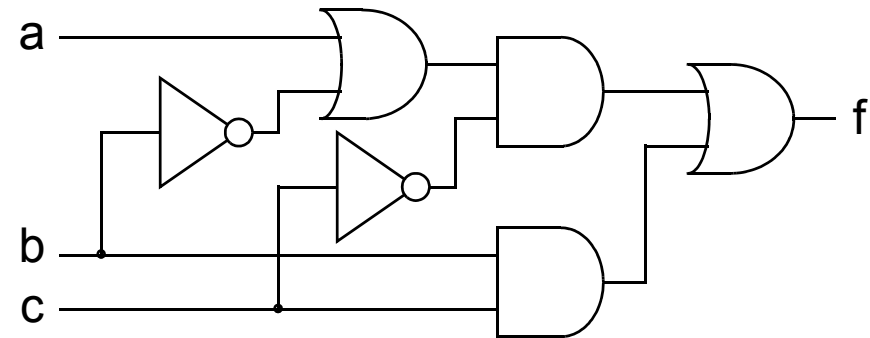
$$f = a b + b c + \bar{b} \bar{c}$$



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

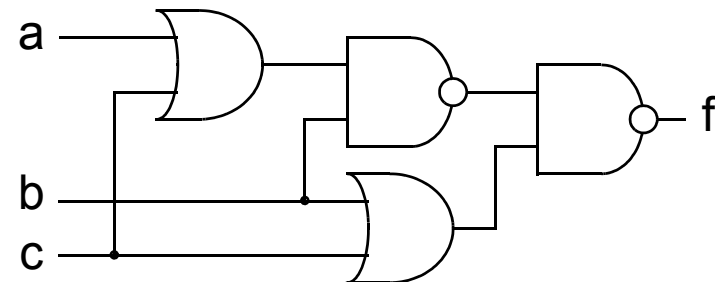
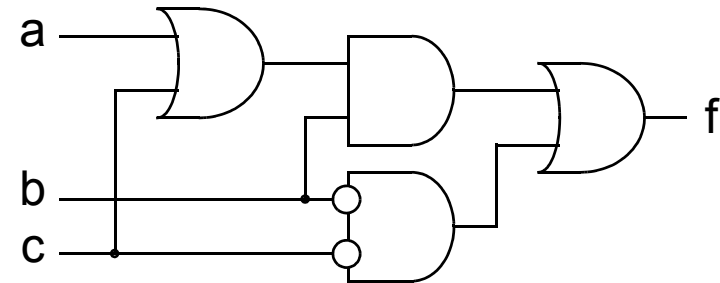
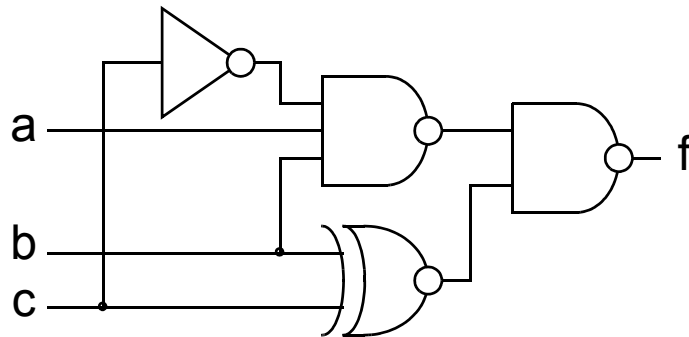
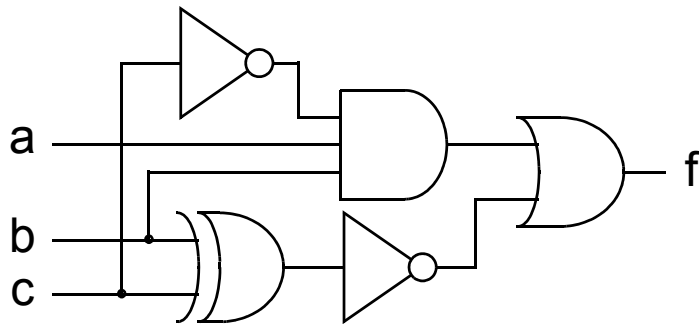


$$f = b c + \bar{c}(a + \bar{b})$$



Skeemi optimaalsus?

- Kuidas alustada ja mida kasutada?



Ülesanne – funktsioonide süsteemi minimeerimine

NB!

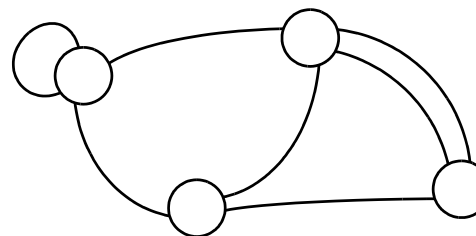
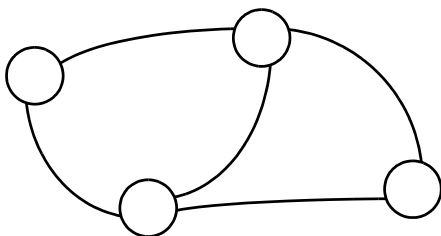
abcd	klmn
0--0	1000
010-	1000
-1-0	1000
1-1-	1000
0-1-	0100
-001	0100
-100	0100
-1-1	0010
01--	0010
1-00	0010
-01-	0001
10--	0001
0-10	0001
1-11	0001

NB!

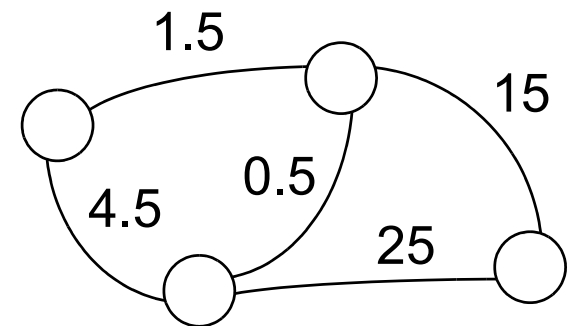
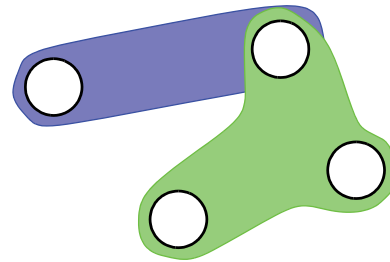
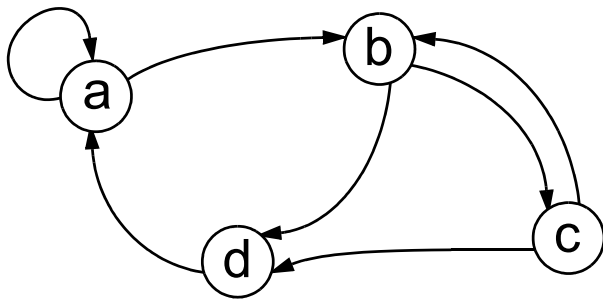
14 vs. 10 implikanti...

Graafid – Graphs

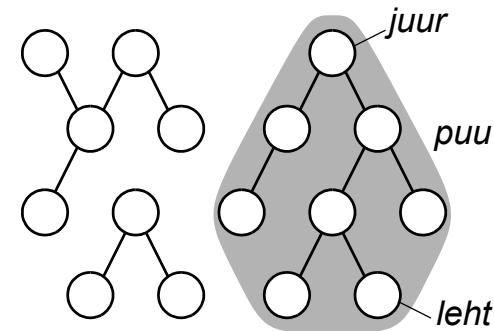
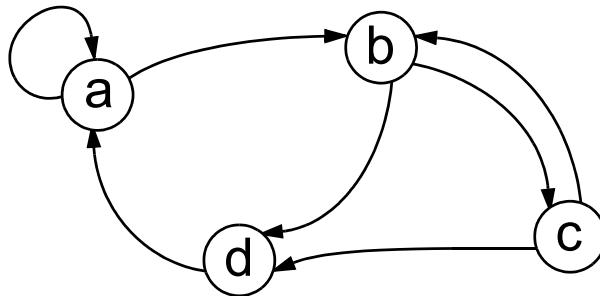
- Leonhard Euler (1707-1783), praktiliselt alles 1930. alates
- **Graaf** (graph) – $G=(V,E)$
 - $v_i \in V$ – sõlmede hulk [node, vertex (pl. vertices)]
 - $e_n = \langle v_i, v_j \rangle \in E$ – servade/kaarte hulk [edge]
- **Silmus** (loop) – serv sõlmest iseendale
- **Multigraaf** (multigraph) – rohkem kui üks serv kahe sõlme vahel (mitmikserv / multiple edge)
- **Lihtne graaf** (simple graph) – puuduvad nii silmused kui ka mitmikservad



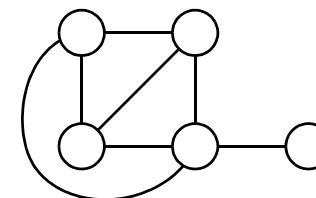
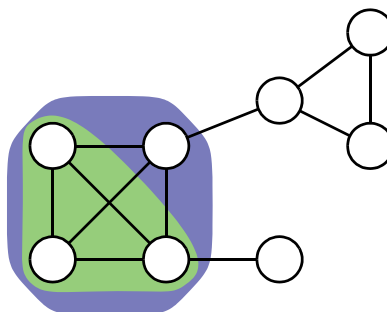
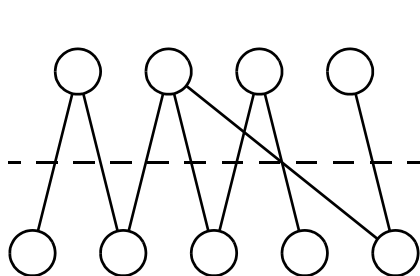
- **Orienteerimata graaf** (undirected graph) – sümmeetriline
- **Orienteeritud graaf** (suunatud graaf, directed graph) – antisümmeetriline (servad)
- **Sõlme aste** (degree) – sõlmega seotud kaarte arv
- **Hüpergraaf** (hypergraph) – iga servaga võib olla seotud rohkem kui kaks sõlme
- **Kaalutud graaf** (weighted graph, network) – reaalarv seostatud servaga (sõlmega)



- **Käik (walk)** - sõlmede ja servade jada – (a, <a,a>, a, <a,b>, b, <b,c>, c, <c,b>, b, <b,c>, c)
- **Rada (trail)** - kõik erinevate servadega – (a, <a,a>, a, <a,b>, b, <b,c>, c)
- **Tee (path)** - rada erinevate sõlmedega – (a, <a,b>, b, <b,c>, c, <c,d>, d)
- **Tsükel (cycle)** - suletud kõik erinevate sõlmedega – (a, <a,b>, b, <b,d>, d, <d,a>)
- **Sidus graaf (connected graph)** - $\forall v_i, v_j \in N \exists \text{path}(v_i, v_j)$,
- **Mets (forest)** / atsükliline graaf (acyclic graph) – graaf ilma tsükliteta
- **Puu (tree)** – sidus atsükliline graaf – juur (root) & leht (leaf/leaves)

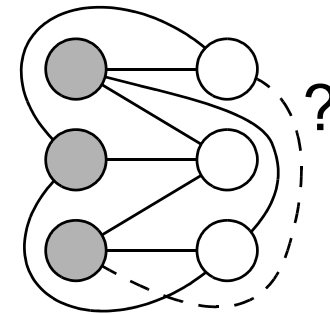
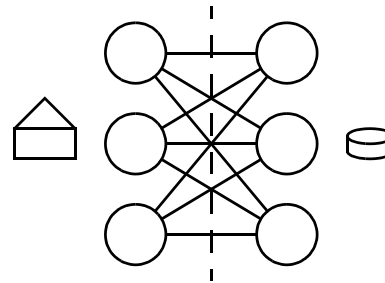
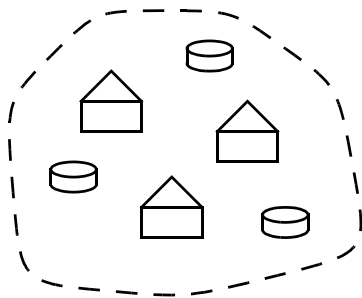


- **Täielik graaf** (complete graph) – iga kahe sõlme vahel on kaar
- **Bikromaatiline graaf** (bipartite graph) – sõlmed on võimalik jagada kahte rühma selliselt, et iga serva üks ottest on ühes ja teine ottest on teises sõlmede rühmas
- **Graafi täiend** (complement) – $G^{-1}=(V,E^{-1})$, e. $E^{-1}=\{ \langle v_i,v_j \rangle \mid \langle v_i,v_j \rangle \notin E \}$
- **Alamgraaf** (subgraph) – $G_S=(V_S,E_S)$, $V_S \subseteq V$ & $E_S \subseteq E$
- **Kliik** (clique) – täielik alamgraaf
- **Maksimaalne kliik** – kliik, mis ei sisaldu üheski teises klikis (mõned autorid nimetavad ainult maksimaalseid alamgraafe klikkideks)
- **Planaarne graaf** (tasandgraaf, planar graph) – kujutades tasandil ükski serv ei lõiku
- **Isomorfsed graafid** – üksühene vastavus sõlmede ja servade hulkade vahel

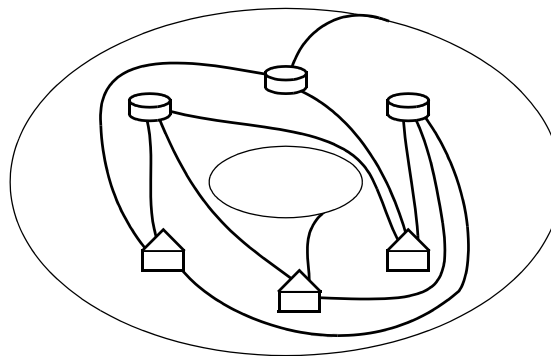


- Näiteülesanne #1 – 3 kaevu & 3 maja

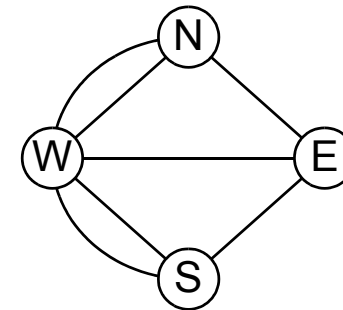
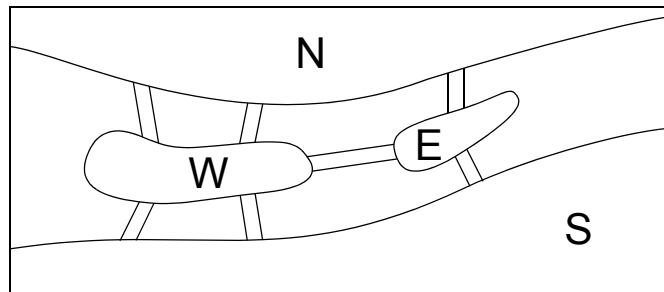
- bikromaatiline graaf
- kas see graaf on planaarne?



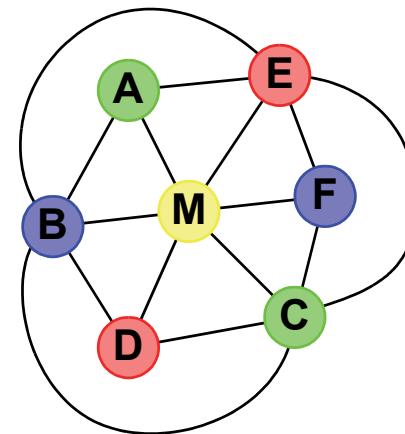
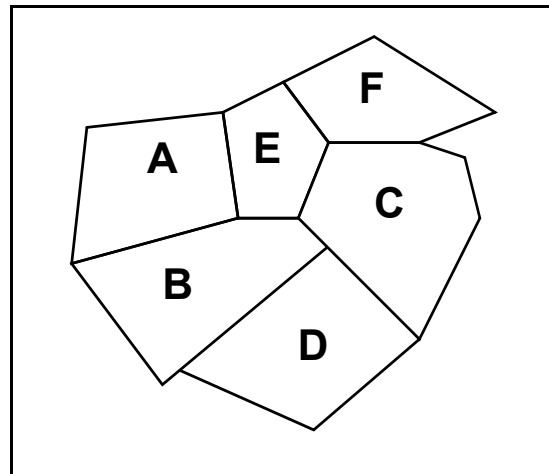
Vaja on kõrgemat järku pinda! :-)



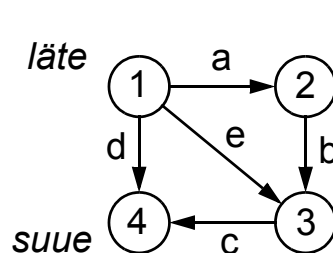
- **Näiteülesanne #2 – Königsbergi sillad (L. Euler 1736)**
 - 2 saart & 7 silda – ületada iga sild täpselt üks kord
 - kas see on võimalik?



- **Näiteülesanne #3 – poliitilise kaardi värvimine**
 - planaarse graafi värvimine
 - mitu värvi on vaja?



- **Suunatud graafid**
 - sisendaste (indegree) & väljundaste (outdegree) – sõlme sisenevate & sõlmest väljuvate kaarte arv
 - alusgraaf (underlying graph) – sõlmede ja kaarte hulk sama, kuid puudub orienteeritus
- **Suunatud atsüklilised graafid (directed acyclic graphs – DAGs)**
 - osaliselt järjestatud hulk
 - järglane (successor, descendant) – v_j on järglane v_i -le kui $\exists \text{ path}(v_i, v_j)$
 - eellane (predecessor, ancestor), otsene järglane, otsene eellane
 - polaarne s.a.g. (polar dag) – $\exists$ läte (source) & suue (sink) ($\equiv$ võre)
- **Intsidentsusmaatriks (incidence matrix) – $|V|$ rida & $|E|$ veergu**
- **Naabrusmaatriks (adjacency matrix) – $|V|$ rida & veergu**



	a	b	c	d	e		1	2	3	4
1	-1	0	0	-1	-1	1	0	1	1	1
2	1	-1	0	0	0	2	0	0	1	0
3	0	1	-1	0	1	3	0	0	0	1
4	0	0	1	1	0	4	0	0	0	0

• Perfektsed graafid (perfect graphs)

- klikiarv (clique number) – $\omega(G)$ – suurima kliki suurus
- tükeldus klikkideks – G on *tükeldatud* täielikeks mittekattuvateks alamgraafideks
- klikikate (clique cover) – G on *kaetud* täielike alamgraafidega
- klikikattearv (clique cover number) – $\kappa(G)$ – minimaalse klikikatte (-tükelduse) võimsus
- stabiilne hulk (stable set) – sõlmed hulgas ei ole kaarega ühendatud
- stabiilsusarv (stability number) – $\alpha(G)$ – suurima stabiilse hulga suurus
- graafi värvimine (coloring) – graafi tükeldamine stabiilseteks hulkadeks
- kromaatileine arv (chromatic number) – $\chi(G)$ – minimaalse stabiilseteks hulkadeks tükelduse võimsus

• Graafi perfektsus

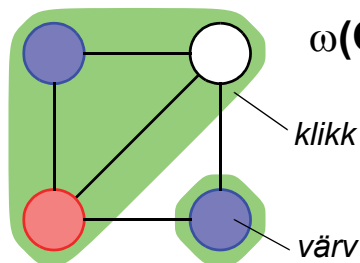
- tavaliselt – $\omega(G) \leq \chi(G)$ ehk $\alpha(G) \leq \kappa(G)$ perfektsed graafid – $\omega(G) = \chi(G)$ ehk $\alpha(G) = \kappa(G)$

$$\omega(G) = 3$$

$$\kappa(G) = 2$$

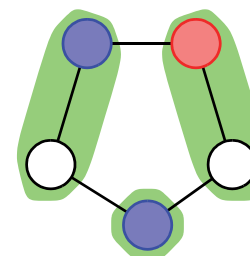
$$\alpha(G) = 2$$

$$\chi(G) = 3$$



$$\omega(G) = \chi(G) \text{ \& } \alpha(G) = \kappa(G)$$

$$\omega(G) < \chi(G) \text{ \& } \alpha(G) < \kappa(G)$$



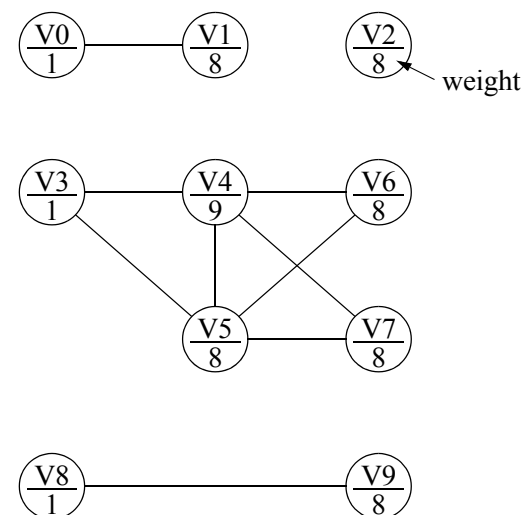
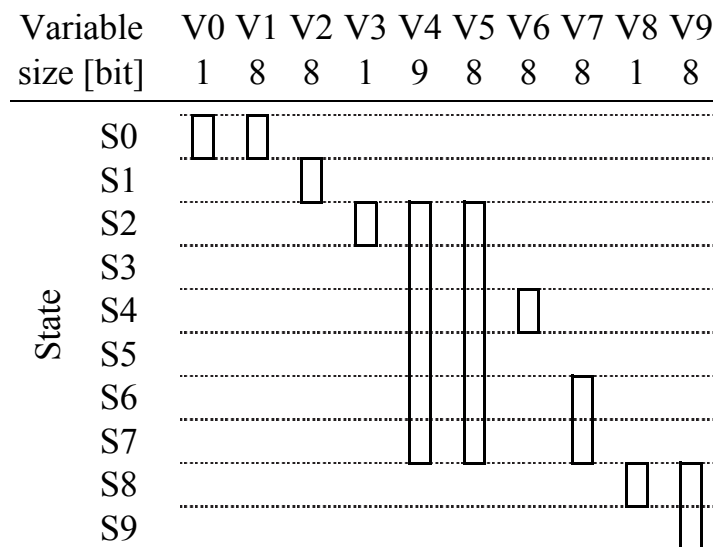
$$\omega(G) = 2$$

$$\kappa(G) = 3$$

$$\alpha(G) = 2$$

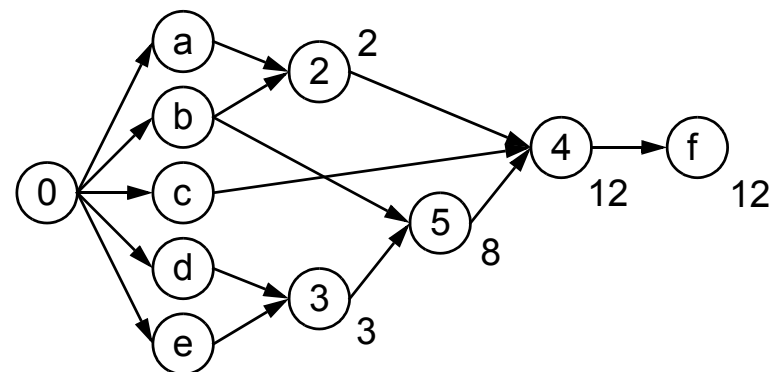
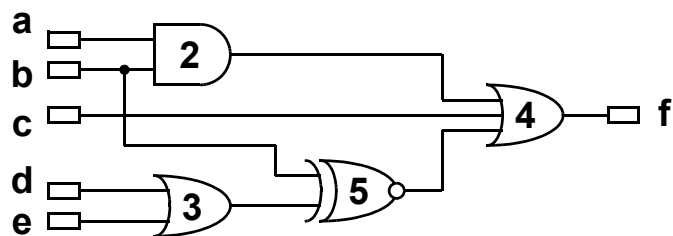
$$\chi(G) = 3$$

- **Kõõlgraafid (kolmnurk graafid, chordal graphs) –**
iga tsükel rohkem kui kolmest sõlmest omab kõõlu (chord)
- **Intervallgraafid (interval graphs) –**
sõlmed on võimalik seada vastavusse vahemikega selliselt, et kahe sõlme vahel on kaar, kui kaks intervalli kattuvad



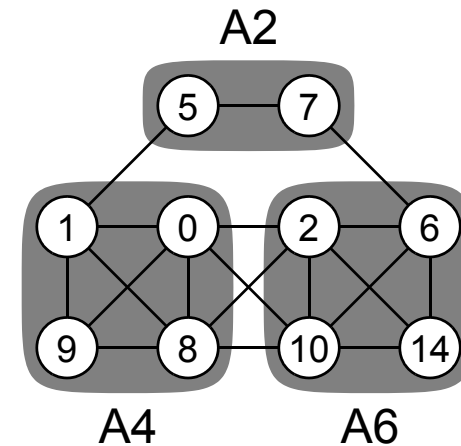
Graafid digitaalsüsteemide disainis

- Disaini ülesanne #1 – viite leidmine ahelas
 - *pikima tee* ülesanne (ainult atsüklilised graafid)
 - sõlm – loogikaelement, sõlme kaal – viide
 - kaar – ühendused loogikaelementide vahel



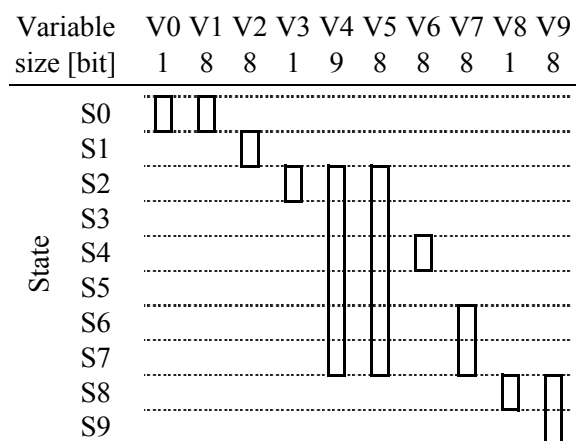
- **Disaini ülesanne #2 – lihtimplikantide hulga minimeerimine**
 - *minimaalse klikikatte* leidmine graafil
 - sõlm – oluline ('1') sisendvektorid (minterm)
 - klikk – lihtimplikant (võimalus asendada hüperservadega)

Impl.	0	1	2	5	6	7	8	9	10	14
A1		x		x						
A2				x		x				
A3					x	x				
A4	x	x					x	x		
A5	x		x				x		x	
A6			x		x				x	x



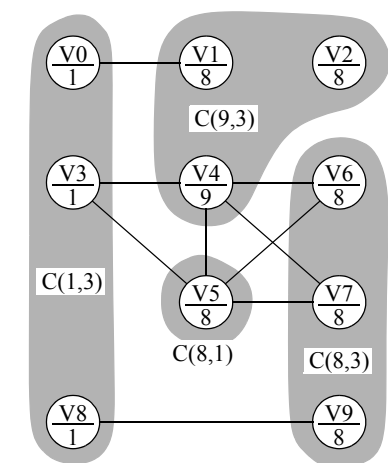
Disaini ülesanne #3 – registrite sidumine

- kaalutud (intervall)graafi värvimine
- sõlm – register, kaar – registrid on korraga kasutuses (intervallid kattuvad)

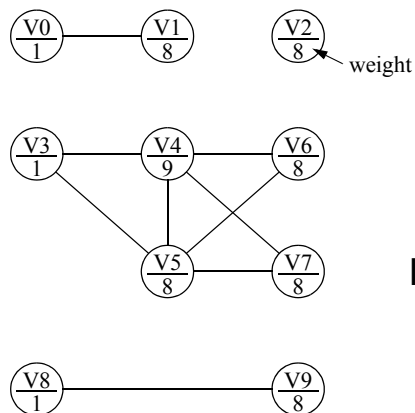


Lähteülesanne:
intervallgraaf

Värvitud graaf

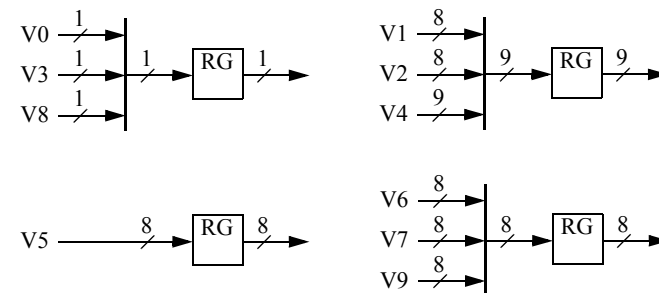


V4, V1, V2, V5, V6, V7, V9, V0, V3, V8



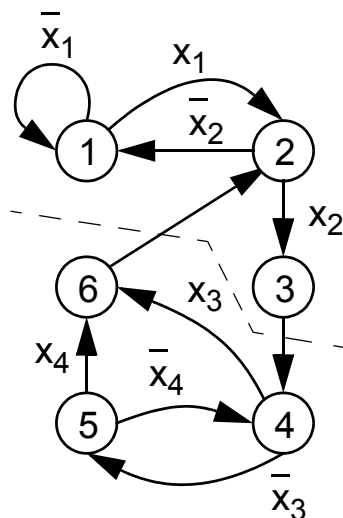
Intervallgraaf

Lõpptulemus:
registrid &
multiplekserid

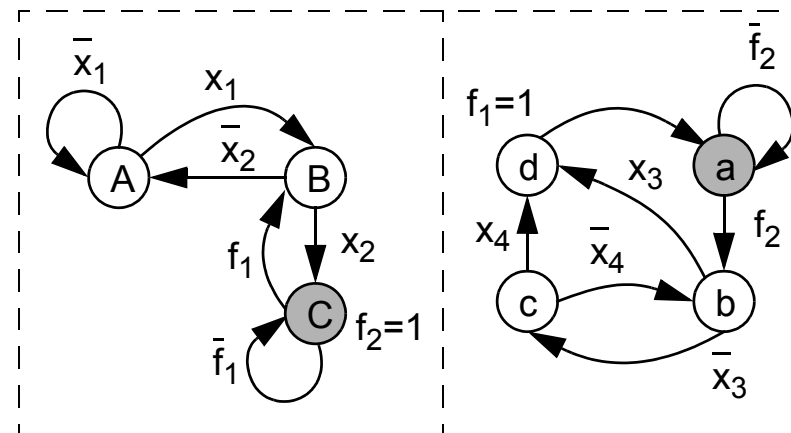


- Disaini ülesanne #4 – algoritmi (automaadi) tükeldamine
 - kaalutud graafi *tükeldamine*
 - sõlm – olek, kaar – siire + tingimused + sagedused/tõenäosused

lähteautomaat



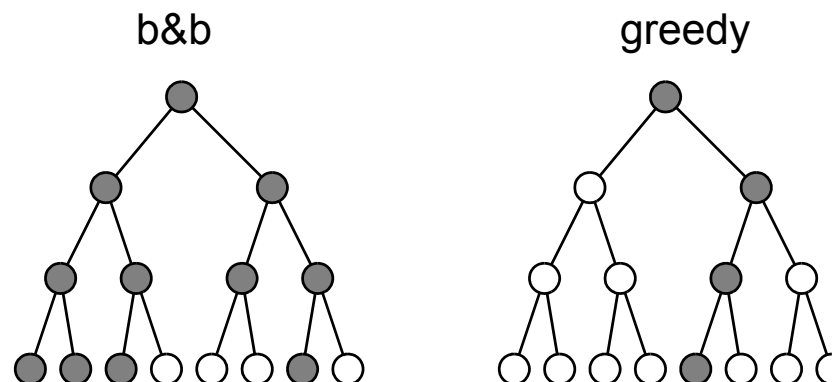
vaheldumisi töötavad komponentautomaadid



Algoritmid ja keerukus

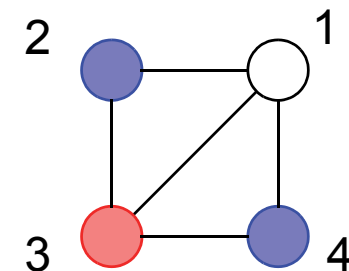
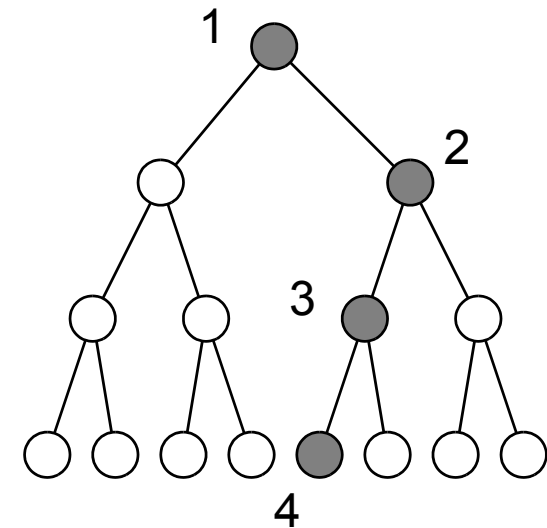
- Algoritmi keerukus – kui kaua kulub aega ja kui palju on vaja resursse
- Keerukus – $O(n)$, $O(n^2)$, $O(2^n)$, jne.
 - polünomiaalne keerukus – P
 - mitte-polünomiaalne keerukus – NP
lahenduv polünomiaalse ajaga, kui on võimalik õige tulemus ära arvata
 - kas $P \subseteq NP$ või $P=NP$ on siiani lahendamata!

- Harude ja tõkete meetod (branch and bound method)
 - parim tulemus
- Ahne meetod (greedy method)
 - piisavalt hea tulemus?



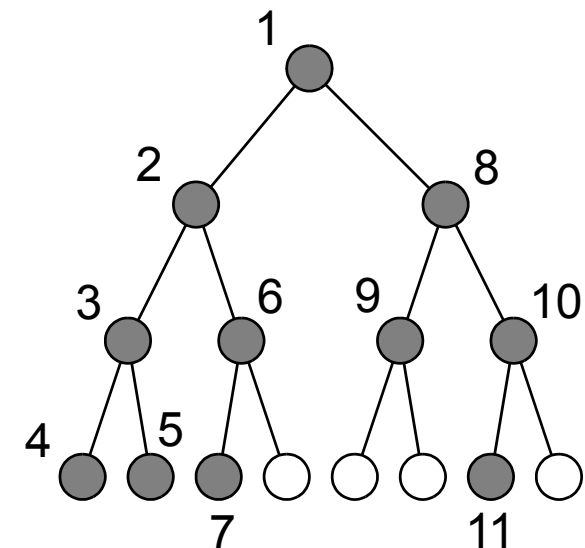
Ahne meetod (Greedy Method)

- **Vali jooksev objekt**
 - suurim, väikseim, juhuslik jne.
 - vali värvimata sõlm
- **Teosta operatsioon**
 - sidumine, eraldamine, värvimine jne.
 - vali legaalne värv
- **Korda kuni operatsioonid on sooritatud kõigi objektidega**
 - korda kuni kõik sõlmed on värvitud



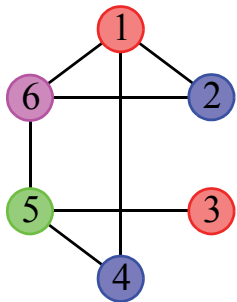
Harude ja tõkete meetod (Branch and Bound Method)

- **Vali jooksev objekt**
 - antud kombinatsiooni korral kasutamata
 - vali värvimata sõlm
- **Teosta operatsioon**
 - sidumine, eraldamine, värvimine jne.
 - vali legaalne värv
- **Kui vahetulemus on halvem jooksvast parimast, siis alusta uut kombinatsiooni**
- **Korda kuni operatsioonid on sooritatud kõigi objektidega**
 - korda kuni kõik sõlmed on värvitud
- **Jäta jooksev parim tulemus meelde**
- **Alusta uut kombinatsiooni**

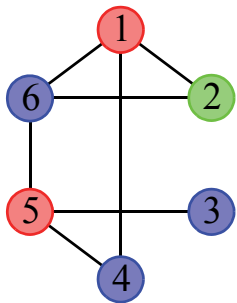


Algorithm – ahne või harud-tõkked?

Greedy

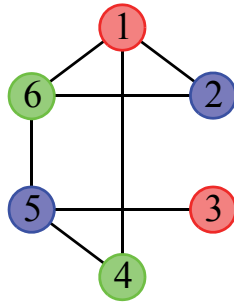


1. 1-[1]-1
2. 2-[2]-2
3. 3-[1]-2
4. 4-[2]-2
5. 5-[3]-3
6. 6-[4]-4

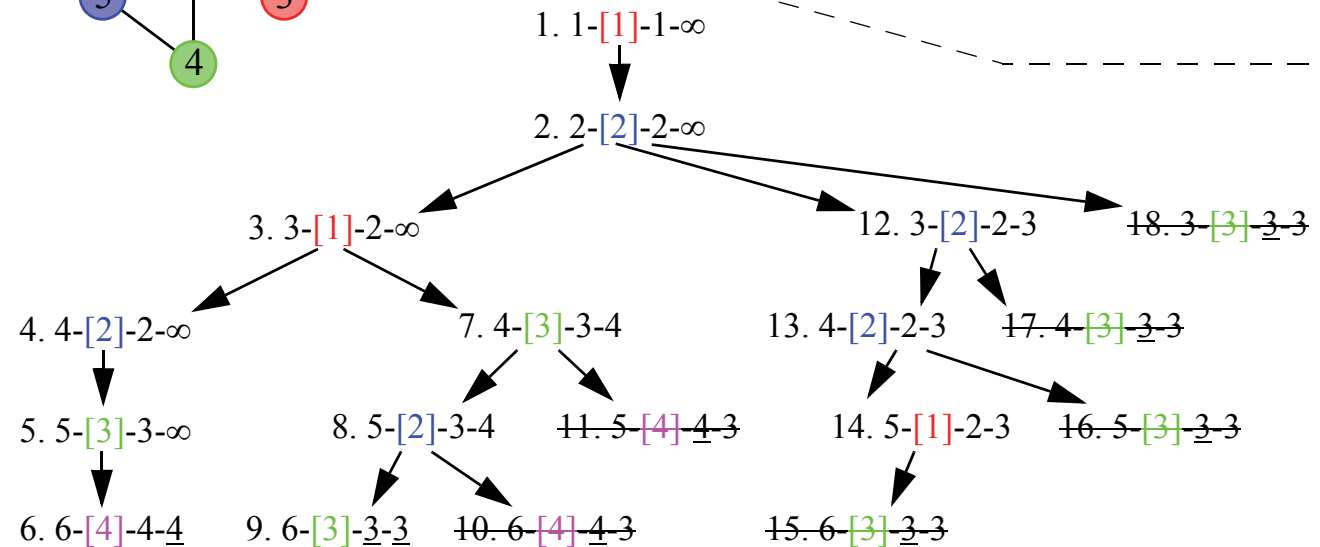
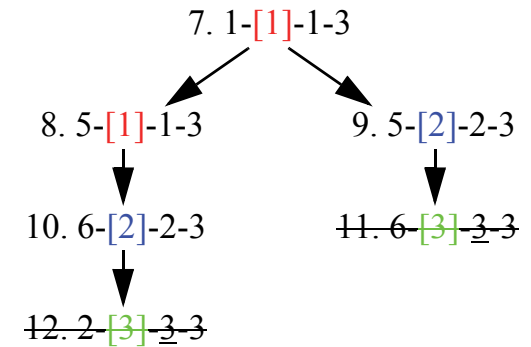
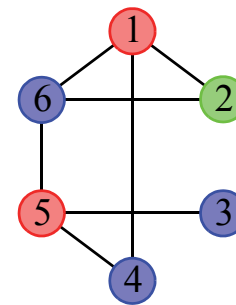


1. 1-[1]-1
2. 5-[1]-1
3. 6-[2]-2
4. 2-[3]-3
5. 4-[2]-3
6. 3-[2]-3

Full search



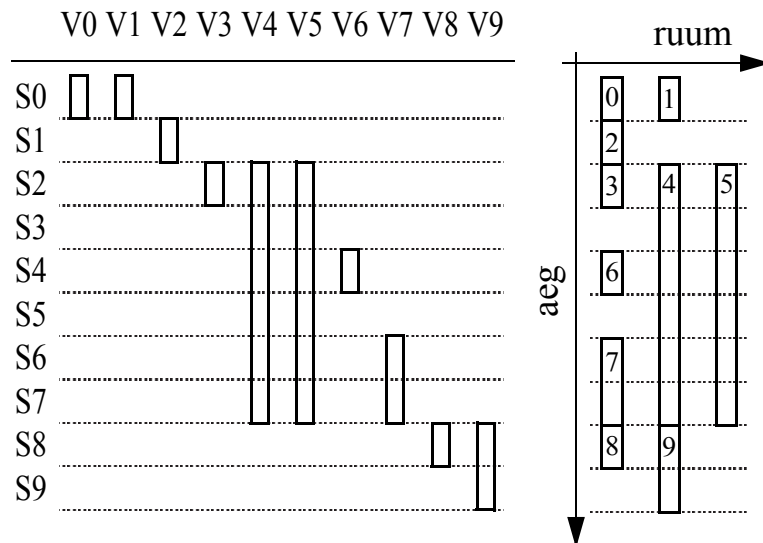
B&B



Pakkimine

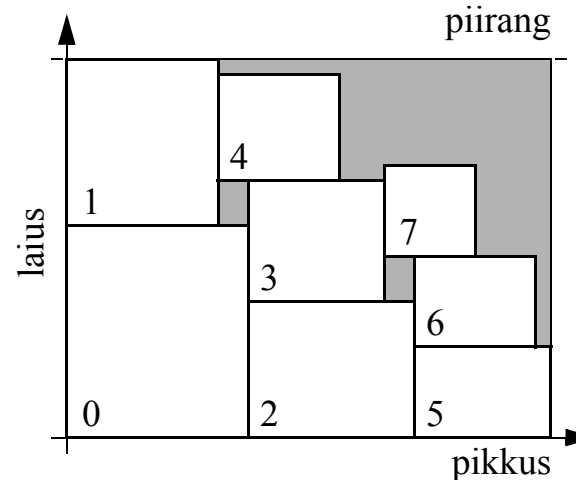
- Võimalikult efektiivne ressursside kasutamine

1D pakkimine
(intervall-graafi värvimine)
[left-edge algorithm]



Täpne lahendus lihtsa algoritmiga...

2D pakkimine
(kastid laos)
[bottom-left algorithm]



Täpne lahendus tömahuka algoritmiga!

