

EKSAMIKÜSIMUSED 2005

Sisukord

Sisukord.....	1
Arvuti riistvara matemaatilised alused.....	4
• Kahendsüsteem.....	4
• Boole funktsioonid ja nende esitus.....	4
• Diskreetne aeg.....	4
Lihtsamaid Boole` funktsioone realiseerivad loogikaelemendid.....	5
• AND.....	5
• OR.....	5
• NOT.....	5
• NAND.....	5
• NOR.....	6
Mikroskeemide valmistamise tehnoloogiad.....	6
Mikroskeemide valmistamise tehnoloogiad.....	6
• unipolaarsed tehnoloogiad (Metal Oxide Semiconductor Field Effect Transistor - MOSFET).....	6
○ n-channel MOS (Metal Oxide Semiconductor - MOS).....	6
○ p-channel MOS.....	6
○ Complementary MOS (CMOS).....	6
• bipolaarsed tehnoloogiad (Bipolar IC Technologies).....	6
○ diod loogika (Diod Logic - DL).....	6
○ diod transistor loogika (Diod Transistor Logic - DTL).....	6
○ transistor transistor loogika (Transistor Transistor Logic - TTL).....	6
○ emittersidestuses loogika (Emitter-Coupled Logic - ECL).....	6
○ integral injeksioon loogika (Interrated Injektion Logic - IIL).....	6
• kolme olekuga väljund.....	7
• avatud suudmega/kollektoriga loogikaelemendid.....	7
Enamkasutatavaid kombinatsioonskeeme.....	7
• välistav või (eXclusive-OR).....	7
• multiplexor (Multiplexers).....	7
• summaator (Adder).....	7
• ALU (Arithmetic-Logic Unit).....	8
• dekooder (Decoder).....	8
• koodimuundur (Code Converter).....	9
Enamkasutatavaid järjestikskeeme.....	9
• trigerid (Flip/flop, latch).....	9
• registrid (Registers) nihkega ja ilma.....	11
• loendurid (Counter).....	12
Protsessor.....	14
• Protsessori üldstruktuur.....	14
○ käsuloendur (PC - Program Counter, IP - Instruction Pointer).....	16

○ käsuregister (IR - Instruction Register)	16
○ käsudekooder (Instruction Decoder)	18
○ juhtautomaat (CU - Control Unit)	18
○ operatsioonautomaat (<i>Data Path</i>)	19
• Käsu täitmine protsessoris (Instruction Execution, fetch-decode-execute cycle).....	20
• RISC - CISC protsessor.....	21
• Konveier protsessoris (<i>Pipeline</i>)	22
• Siirete (hargnemiste) ennustamine.(<i>Branch Prediction</i>).....	23
• Peidikmälu, vahemälu (<i>Cache</i>)	24
Arvuti mälu	29
• Mälu hierarhia arvutis (<i>Memory hierarchy</i>).....	31
• Arvuti mälu klassifikatsioon (Computer memory classification)	32
• Muutmälu (RAM)	32
• Staatileine pooljuht suvapöördusmälu (<i>Static RAM</i>)	33
• Dünaamiline pooljuht suvapöördusmälu (<i>Dynamic RAM</i>)	35
• Püsimälu (ROM - Read Only Memory)	37
• Magnet mäluseadmed (<i>Magnetic memory</i>)	39
○ Mullmälu (Bubble).....	40
○ Pehme ketas (Floppy).....	40
○ Kõvaketas (Hard drive)	40
○ Magnet ketas	41
○ Lint (Tape).....	41
• Optilised mäluseadmed (<i>Optic memory</i>).....	42
○ CD-ROM.....	43
○ CD – R.....	43
○ CD – RW.....	43
○ DVD	43
○ Magnetoptiline (MO)	43
○ Holograafiline.....	45
• Erinevate pöördus viisidega mälud (pinumälu (<i>Stack, LIFO</i>), puhvermälu (<i>FIFO</i>))	45
○ Stack	45
○ LIFO	45
○ FIFO	45
Käsusüsteem ja adresseerimine	33
• Käsuformaadid ja käsusüsteem (<i>Instruction set</i>)	46
• Adresseerimise viisid (<i>Addressing modes</i>)	46
Mikroarvuti riistvara	46
• Mikroarvuti arhitektuur ja siinid.	46
• Erinevad siinid ja nende osa andmevahetuses.....	47
• Andmeedastus protokollid.....	48
• Andmevahetuse juhtimine (<i>Bus arbitration</i>).....	50
• Sisend-väljund seadmete ja protsessori andmevahetus	51
• Mikroprotsessori juurde kuuluvad komponendid (<i>Supporting System</i>).....	51
○ Mälu kontrollor (Memory controller).....	51

○ Peidikmälu, vahemälu kontrollid (<i>Cashe controller</i>).....	51
○ Siini kontrollid (<i>Bus controller</i>).....	51
○ Mälu otsepöördus režiimi kontrollid (<i>DMA controller</i>).....	51
○ Programmeeritav katkestuste kontrollid (<i>Programmable interrupt controller</i>)	51
○ Programmeeritav taimer (<i>Programmable interval timer controller</i>).....	51
Sisend-väljund seadmed	51
• Klaviatuur (<i>Keyboard</i>)	51
• Hiir ja juhtkang (<i>Mouse and joystick</i>).....	54
• Kuvar (<i>Display</i>).....	56
○ CRT (<i>Cathode Ray Tube</i>) kuvar	56
○ kujundi moodustamine	56
○ videomälu (<i>Video memory</i>)	57
○ vedelkristall kuvar LCD (<i>Liquid Crystal Display</i>).....	58
○ värviline kujund.....	60
• Printer	60
○ maatriksprinter (<i>Dot matrix printer</i>).....	60
○ laserprinter (<i>Laser Printer</i>)	61
○ jugaprinter (<i>Inkjet Printer</i>)	62
○ värviprinterid	62
• Plotter	62
• Skanner	63
• Modem (<i>Modem</i>)	64
• Analoo liides (<i>Analog Interface</i>).....	64
○ analoog-digitaal muundur (<i>Analog to Digital Conversion</i>)	64
○ digitaal-analoog muundur (<i>Digital to Analog Conversion</i>)	65
• UPS- Uninterruptible power supply	65
Spetsiaalse riistvara	67
• Spetsiaalse riistvara realiseerimise võimalused.	67
○ Programne realiseatsioon	67
○ Riistvaraline realiseatsioon	68
○ Programmeeritav loogika	68
• Erinevate spetsiaalse riistvara realiseerimise võimaluste kasutusvaldkonnad ja võrdlus.	71
Arvutite riistvara veakindlus.	71
• Rikked arvuti riistvaras.	71
• Testimine.	72
• Testitava riistvara projekteerimine.....	73
• Veakindlad koodid	73
• Töökindluse tõstmine.	73

Arvuti riistvara matemaatilised alused

• Kahendsüsteem

Kahendsüsteem on positsiooniline arvusüsteem, mille alus on 2.

Seega kasutatakse kahendsüsteemis kahte numbrimärki, milleks tavaliselt on 0 ja 1. Tihti öeldakse numbrimärgi 1 kohta tõene ja numbrimärgi 0 kohta väär -- seda seetõttu, et selliselt käsitletakse neid kahendloogikas.

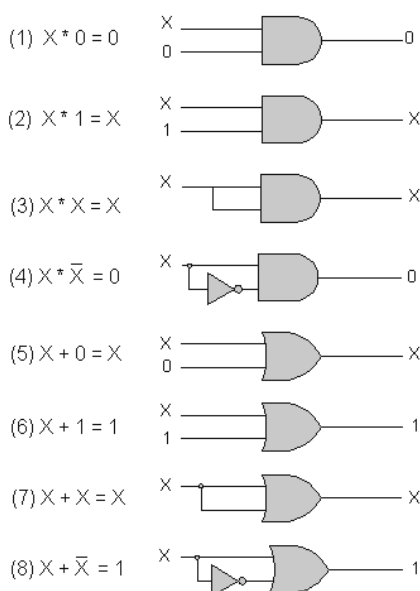
Kahendsüsteemis esitatakse arve samal põhimõttel nagu kümnendsüsteemis. Erinevus on ainult selles, et kümnendsüsteemi alus on 10 ja vastavalt ka numbrimärke on 10. Näiteks arv kaks esitatakse kahendsüsteemis kujul 10 (üks kaheline pluss 0 ühelist), arv 6 kujul 110 (üks neljaline pluss üks kaheline pluss 0 ühelist) ja arv 999 kujul 1111100111 (üks viiesajakaheteistkümmeline pluss üks kaheajaviiekümnekuueline pluss üks sajakahekümnekaheksaline pluss üks kuuekümmeneljaline pluss üks kolmekümnekaheline pluss null kuuteistkümmelist pluss null kaheksalist pluss üks neljaline pluss üks kaheline pluss üks üheline).

Kahendsüsteem on arvutikeele alus.

Kahendsüsteemi arvukoht tähistab vastavat kahe astet, nagu kümnendsüsteemi arvukoht tähistab vastavat kümne astet (10 on kümnendsüsteemis $10^1=10$, sest 1 on tagant teisel kohal, kahendsüsteemis $2^1=10$ samal põhjusel).

Esimesed arvud kahendsüsteemis: 0, 1, $10=2$, 11, $100=4$, 101, 110, 111, $1000=8$, 1001, 1010, 1011, 1100, 1101, 1110, 1111, $10000=16$. Nagu siit näha, korrutab iga nulli lisamine arvu kahega.

• Boole funktsioonid ja nende esitus



Korrutamine

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

Liitmine

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

Lahutamine

$$0 - 0 = 0$$

$$0 - 1 = -1$$

$$1 - 0 = 1$$

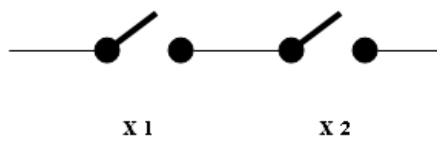
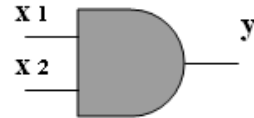
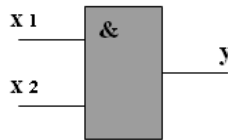
$$1 - 1 = 0$$

• Diskreetne aeg

Lihtsamaid Boole` funktsioone realiseerivad loogikaelemendid

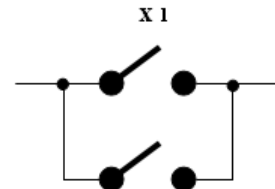
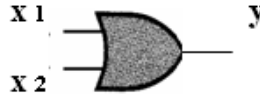
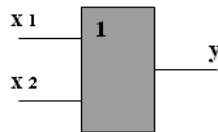
• AND

X ₁	X ₂	Y
0	0	0
0	1	0
1	0	0
1	1	1



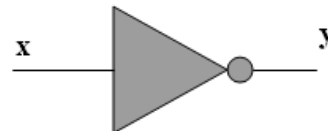
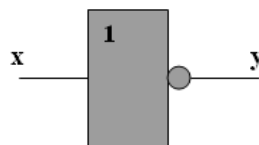
• OR

X ₁	X ₂	Y
0	0	0
0	1	1
1	0	1
1	1	1



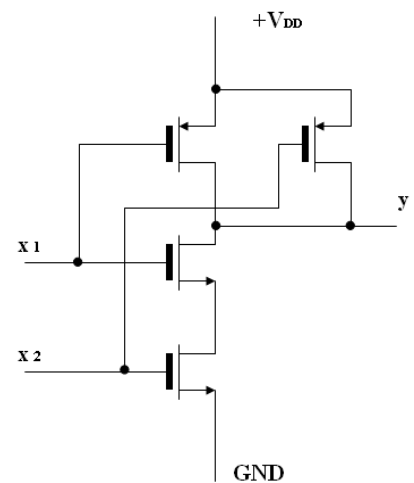
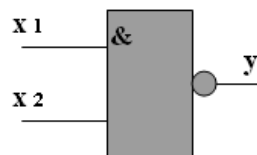
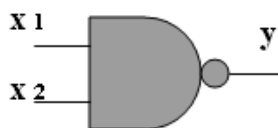
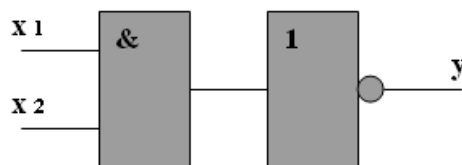
• NOT

X	Y
0	1
1	0

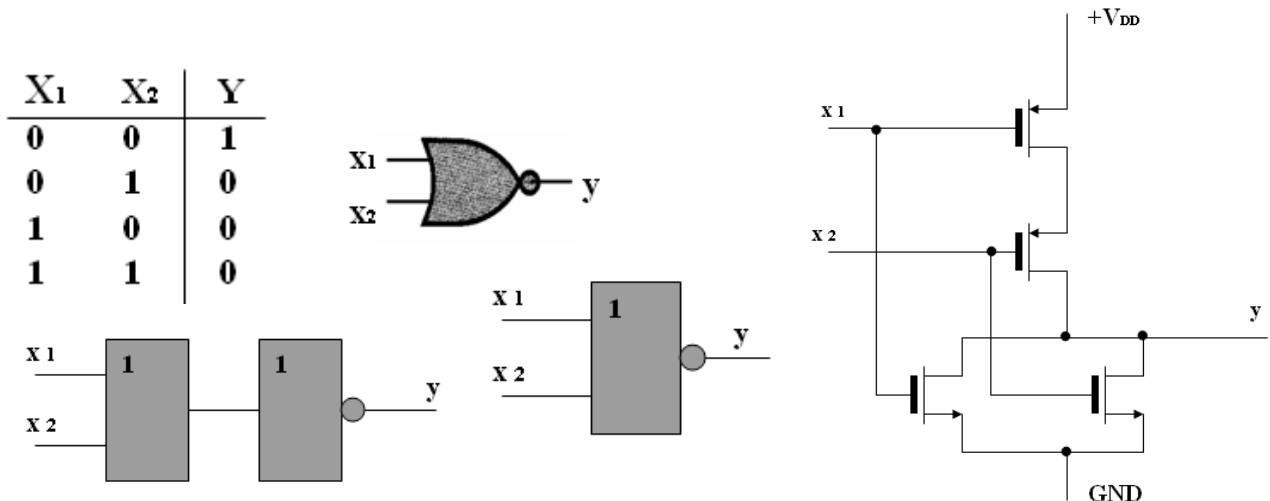


• NAND

X ₁	X ₂	Y
0	0	1
0	1	1
1	0	1
1	1	0



- **NOR**



Mikroskeemide valmistamise tehnoloogiad

- **unipolaarsed tehnoloogiad (Metal Oxide Semiconductor Field Effect Transistor - MOSFET)**
 - n-channel MOS (Metal Oxide Semiconductor - MOS)
 - p-channel MOS
 - Complementary MOS (CMOS) (Üksteist täiendav MOS)
- **bipolaarsed tehnoloogiad (Bipolar IC Technologies)**
 - diiod loogika (Diod Logic - DL)
 - diiod transistor loogika (Diod Transistor Logic - DTL)
 - transistor transistor loogika (Transistor Transistor Logic - TTL)
 - suure võimsusega (high power, H type)
 - madala võimsusega (low power, L type)
 - Schotky TTL
 - emittersidestuses loogika (Emitter-Coupled Logic - ECL)
 - integral injeksioon loogika (Interrated Injektion Logic - IIL)

- kolme olekuga väljund
- avatud suudmega/kollektoriga loogikaelemendid

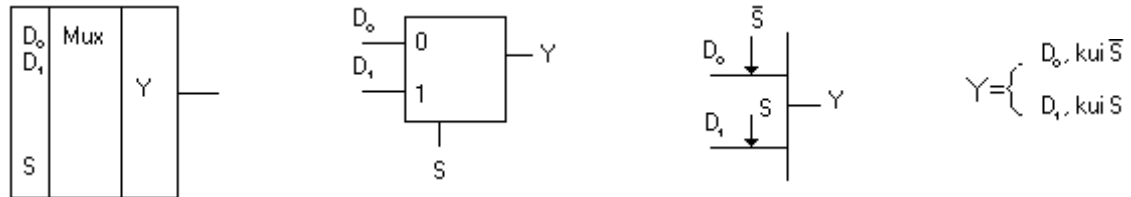
Enamkasutatavaid kombinatsioonskeeme

- välistav või (eXclusive-OR)
- multiplexor (Multiplexers)

Multiplexsor kujutab endast andmeselektorit. Multiplexsoril on mitu sisendit ja üks väljund. Sisendid jagunevad infosisenditeks ja juhtsisenditeks, kusjuures infosisendite arv määrab ära juhtsisendite arvu ning vastupidi. Vastavalt juhtsignaalile kommuteeritakse multiplexsori väljundisse signaal ühest infosisendist. Kommuteeritavate infosisendite arv võrdub 2^n , kus n on juhtsisendite arv. Järelikult saab kahe juhtsisendiga ehk kahebitise koodiga kommuteerida 4 sisendit, kolme juhtsisendiga 8 sisendit jne.

Piisava arvu sisenditega multiplexsori abil saab realiseerida suvalisi loogikafunktsioone.

Tähistused:



- summaator (**Adder**)

Summaatoriks nim. arvuti loogikalülitust, mis on ette nähtud arvkode aritmeetiliseks summeerimiseks. Mitmejärgulise kahendarvu summaator koosneb mitmest ühejärgulisest summaatorist. Arvu summeerimisel tuleb lisaks kahe summeeritava arvu vastavatele järkudele liita nendega ka nooremate järkude summeerimisel tekkinud ülekannet. Seega on ühejärgulisel summaatoril kolm sisendit ning 2 väljundit. (kaks sisendit ja kolm väljundit?)

Poolsummaator - ei arvesta liitmisel eelmisest järgust tulenevat ülekannet. Kasutades kahte poolsummaatorit, on alati saadav üks täissummaator.

Täissummaator - arvestab liitmisel eelmisest järgust tulenevat ülekannet

Jada ülekandega e. järjestikülekandega summaatoris moodustatakse väljundsignaal arvukohtade järjestikku summeerimisega, alates kõige nooremast (parempoolsest) kuni kõige vanema ehk vasakpoolsemani välja. Arvukoha summeerimiseks ja ülekande moodustamiseks

kulub teatud aeg, mida ülekande seisukohalt võib vaadelda hilistumisenä. Kuna ülekanne toimub järjestikku, siis aeglustab see summaatori tööd. Suure kohtade arvu korral on koguhilistumine võrdne hilistumise summaga üksikutes kohtades.

Rööpülekandega e. paralleelülekandega summaatorid töötavad palju kiiremini kui jadaülekandega summaatorid. Mitmekohalise kahendarvu summeerimisel moodustatakse ülekanne korraga kõigi kohtade jaoks. Seetõttu ei kulu ülekandeks lisaaega ning summaator töötab kiiremini kui jadaülekande korral.

Kiire ülekandega summaatorid - nende puhul on rakendatud rööpülekande põhimõtet kombineeritult koos jadaülekandega. Ülekanded on moodustatud kõigi kohtade jaoks korraga.

Lahutajad - $A-B=V$

1) otseteel (kõigi variantide analüüs)

2) matemaatiliselt

Vahe avaldis langeb kokku summa avaldisega. Ja kui joonistada skeem, siis teab, et see skeem on võimeline nii liitma kui ka lahutama.

$M=0$,toimub summeerimine “+”

$M=1$,toimub lahutamine “-”

• **ALU (Arithmetic-Logic Unit)**

Sõltumata arvuti ja protsessori ehitusest on arvutis alati üks skeemiosa, kus teostatakse otsesed arvutustehted ja muu infotöötlus – nimelt aritmeetika-loogikaseade ehk ALU (Arithmetical and Logical Unit). Eri protsessoritel on üldiselt erinev tehete hulk ja valik, kuid tavaliselt hõlmab see aritmeetilisi (minimaalselt liitmine ja lahutamine) ning loogilisi tehteid (JA, VÕI, EITUS) ja nihutusoperatsioone (kahendarvu bitid nihutatakse oma senise positsiooni suhtes kas vasakule või paremale).

• **dekooder (Decoder)**

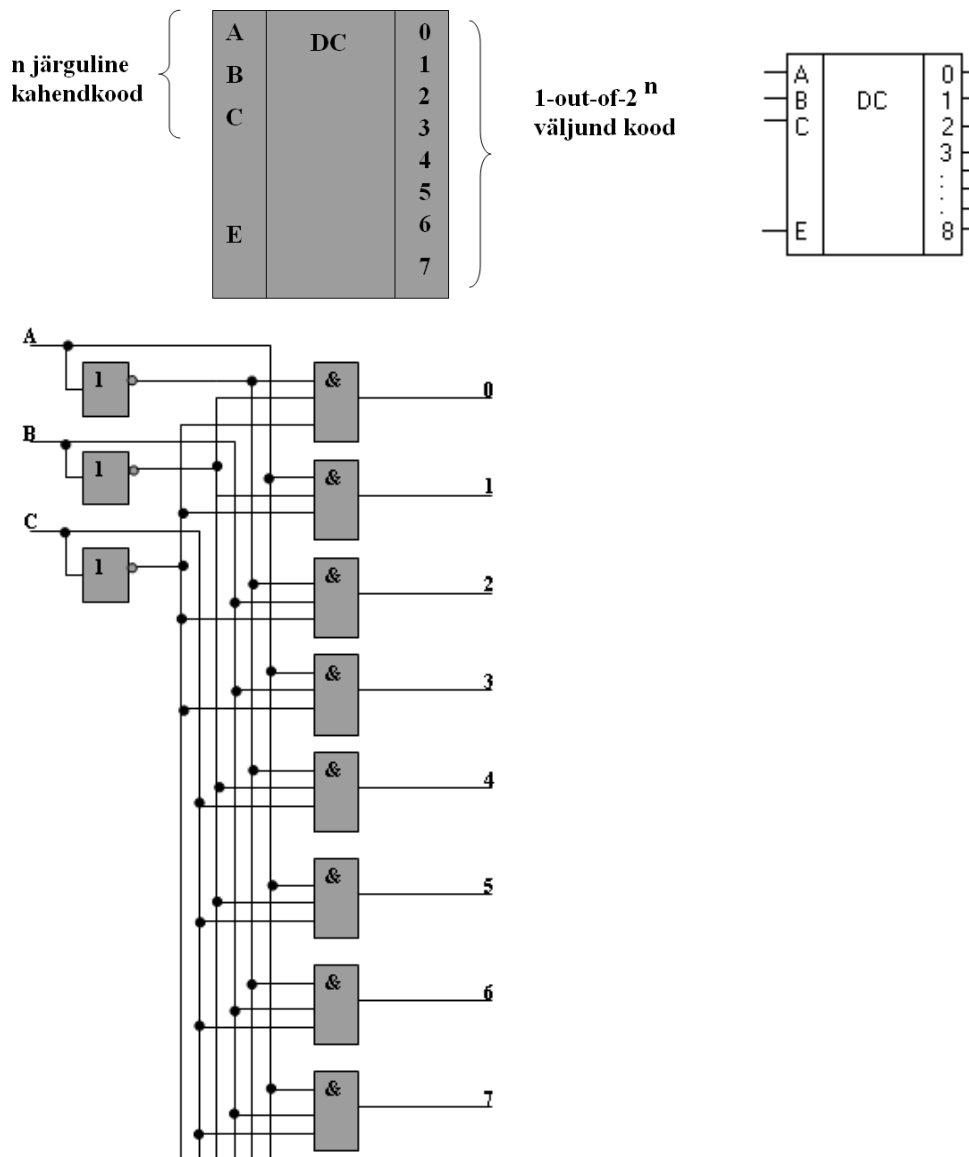
Dekooder on lülitus, mis on ette nähtud etteantud sisendkoodi muundamiseks soovitud väljundkoodiks. Ta tunneb ära sisestatava kahendarvu ja annab signali vastavasse väljundisse.

Dekoodri ülesandeks on muundada kahendkoodis arv niisuguseks koodiks, millega saab aktiveerida nõutava mälupea, juhtida number- või tähtindikaatorit, tunda ära mitmesuguseid kodeeritud signaale, muundada kahendkoodis antud arv kümnendsüsteemi arvuks jne.

Üldjuhul on dekodeeril nii mitu sisendit n , kui mitu kohta on sisendisse antaval kahendarvul. Maksimaalne väljundite arv võrdub kombinatsioonide arvuga 2^n . Dekodeerid koostatakse peamiselt NING- elementidest.

Suure sisendite arvu korral kasutatakse dekodeerimiseks nn. kaskaadlülitust, kus esimese astme dekodeer aktiveerib ühe teise astme dekodeer ja see omakorda ühe väljundi.

Kahend dekooder – n -to- 2^n decoder



- **koodimuundur (Code Converter)**

Muundab ühte tüüpi koodi teist tüüpi koodiks. Näiteks muundab kahendkoodi kümnendkoodiks.

Enamkasutatavaid järjestikiskeeme

- **trigerid (Flip/flop, latch)**

Triger on mäluelement, mis säilitab 1 biti informatsiooni. Triger on kahe stabiilse olekuga loogikalülitus (1 või 0). Trigeri olek vastab tema väljundsignaalile. Sõltuvalt sisendsignaalist säilitab triger endise oleku või muudab seda hüppeliselt. Trigeril tavaliselt 2 väljundit: otsene Q ja invertne $\bar{Q}$.

Tööpõhimõtte järgi jaotatakse trigerid seadesisenditega ehk SR- trigeriteks, loendussisenditega e. T-trigeriteks, andmesisenditega ehk D-trigeriteks ning universaalsisenditega e. JK-trigeriteks.

Kui trigeri oleku muutmine toimub kasvõi ühe sisendi kaudu täiendava sünkroniseerimis signaali abil, nim. trigerit sünkroonseks, vastupidisel juhul aga asünkroonseks. Sõltuvalt tööpõhimõttest ning ehitusest liigitatakse trigerid ühe- või kahetaktiliseks.

Triger on elementaarne salvestuselement, millel on 2 olekut. Kasutatakse mäluelementidena registrites, loendurites jne. Informatsiooni salvestusviisi järgi jagunevad 2-ks:

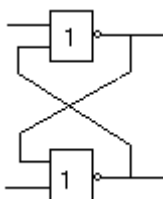
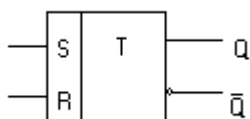
1) asünkroonsed - salvestatakse info vahetult sisenditesse antud signaalidega.

2) sünkroonsed - see on võimalik ainult sünkroimpulsi olemasolul.

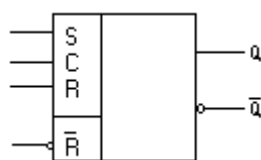
RS (reset-set) , ühe ja kahetaktiline, antud on asünkroonne, $R=S=1$ on keelatud. Töötab: RS; $Q(t)$, $00 \rightarrow Q(t-1)$, $01 = 1$, $10 = 0$, $11 = \text{---}$.

R	S	Q^t	
0	0	Q^{t-1}	ei muutu
0	1	1	Set
1	0	0	reset
1	1	-	keelatud

*a-sünkroonne



* sünkroonne



NB! Keelatud on anda mõlemasse

sisendisse signaal 1.

Sünkroonne ühetaktiline SR-triger erineb asünkroonses selle poolest, et trigeri olek muutub vaid kindlail sünkroimpulssidega määratud ajahetkeil. Lisaks infosisenditele S ja R on tal veel sünkroiseerimis sisend C (clock). Sünkroniseeritud infosisend toimib hetkel, mil saabub sünkroniseerimis- signaal.

Kahetaktiline sobib sinna (skeemidesse), kus on vaja saada tagasisidet. Näiteks mälu vaatamine jne.

T (toggle), 1infosisendiga, iga järgmine impulss muudab trigeri oleku vastupidiseks, nn. loendustriger. Töötab: $T; Q(t), 1 = \neg Q(t-1), 0 = Q(t-1)$.

T	Q^t
0	Q^{t-1}
1	Q^{t+1}

D (delay), data 1 infosisend, väljundis kordab sisendi signaali, aga sünkroimpulsi võrra hiljem, saab säilitada lühiajaliselt infot. Töötab: $CD; Q(t), 0 = \neg Q(t-1), 11 = 1, 10 = 0$.

C	D	Q^t
0	-	Q^{t-1}
1	1	1
1	0	0

JK (jump-key), samasugune nagu RS-triger, aint selle vahega et ei ole keelatud kombinatsiooni. Siin on lubatud $J=K=1$, mis muudab väljundi vastupidiseks. Töötab: $JK; Q(t), 00 = Q, 01 = 0, 10 = 1, 11 = \neg Q$.

J	K	Q^{t-1}	Q^t
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

MS (master-slave), kaksiktrigerid, siseviivitusega.

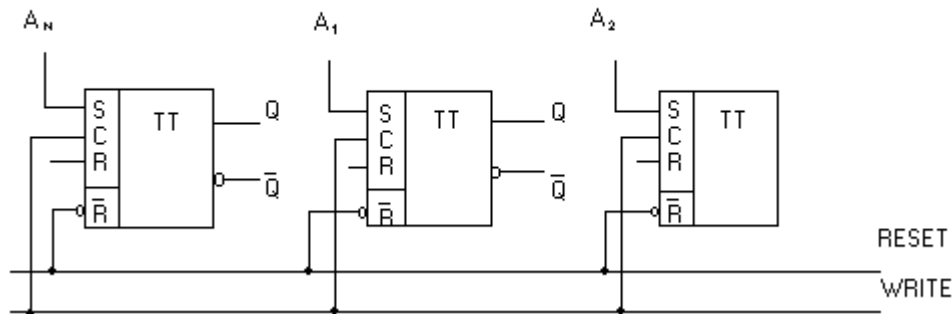
• registrid (*Registers*) nihkega ja ilma

Registriteks nim. trigeritest koosnevat seadet, mis võimaldab salvestada, säilitada ning taasesitada infot ühe sõna kaupa. Lisaks nihutatakse registri abil infosõna bitt vasakule või paremale. Sõna nihutamise muundatakse rööpkoodis esitatud info jadakoodiks ning vastupidi. Sõna pikkus sõltub registri trigerite arvust ning võib olla väga erinev. Enam on levinud 8-, 16-, 24-, ja 32-bitised registrid, mis vastavad sõnapikkusele 1, 2, 3 ja 4 baiti. Registrit juhitakse signaalidega: vastuvõtt (write) ja 0-seade (reset). Signaalidega write kirjut. sisendite $A^0 \dots A^n$ informatsioon registrisse, signaaliga reset aga kustutatakse sealt. Register on hulk ühtse juhtimisega trigereid. Ta on trigeritel põhinev lülitus kahendarvude registreerimiseks.

(Registriks nim. seadet, mis võimaldab salvestada, säilitada ning taasesitada infot ühe infosõna kaupa. Info säilib nii kaua kuni on toide sees. Bitt on võimalik sisestada ja väljastada rööbiti ja järjestikku. Rööbiti - mäluregister, järjestikku - nihkeregister. Registri põhiülesandeks on mitmejärgulise arvu säilitamine. Register koosneb trigeritest, kus iga triger säilitab ühte kahendarvu järku; n-järgulise arvu jaoks peab olema n trigerit. Registrit võib kasutada ka arvude nihutamiseks paremale või vasakule (arvu järgud liiguvad korraga üks järk paremale v. vasakule),

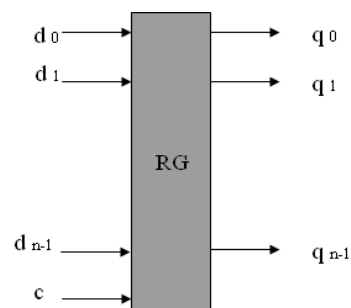
arvujada esituse viimiseks röökujule ja vastupidi. Sõltuvalt arvu esitusviisist jaotatakse registrid jada- ja rööpregistriteks. Rööpregistrisse antakse säilitavana arvu kõik järgud korraga. Jadaregistrisse antakse arvu järgud ühekaupa tavaliselt alates nooremast järgust.)

Registri põhimõtteskeem.



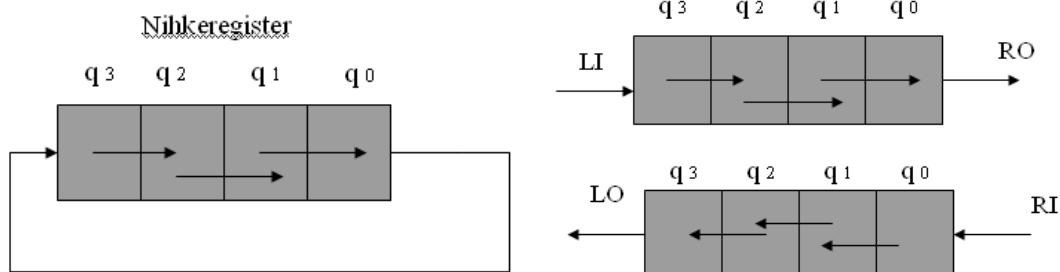
Ilma nihketa register

Hulk ühise juhtimisega trigereid.



Nihkeregister

Registrid, millesse info sisestamine ja väljastamine toimub järjestikku nim. nihkeregistriteks. Nihkeregistri koostamiseks kasut. nii RS-, D- kui ka JK- trigereid. Nihkeregistris ühendatakse otsene ja inverteeritud väljund järgmise trigeri seadesisenditega S ja R. Seega toimub iga taktiga infosõna nihutamine ühe biti võrra. Sõltuvalt sellest kuidas trigerid omavahel ühendatakse, nihkub infosõna kas paremale või vasakule. Iga takti keskel nihutab sünkrosignaali info trigerite esimesest astmest teise. Reversiivne register- selle puhul toimub kahe suunaline nihe.

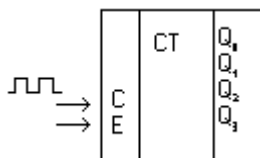


• Ioendurid (Counter)

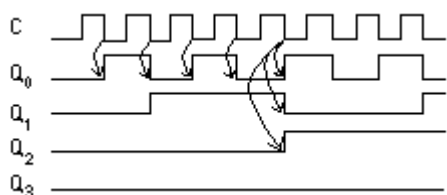
Loenduriteks nimetatakse impulsside loendamiseks ette nähtud loogikalülitust. Loendureid kasutatakse nii automaatikaseadmetes kui ka arvutustehnikas.

Sisse tulevad impulsid. Väljundiks 0,1 kombinatsioonid. Erinevate väljundkombinatsioonide arvu nim. mooduliks.

E- sisend, mis lubab loendamise



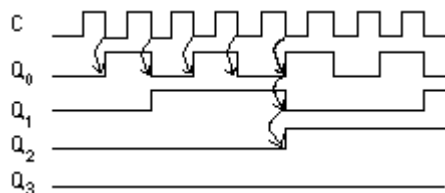
Kaks diagrammi- üks sünkroonse, teine asünkroonse jaoks.



Sünkroonne loendur - ümberlülitumine toimub samaaegselt v. paralleelselt.

Ümberlülitumisaeg on kogu aeg samasugune.

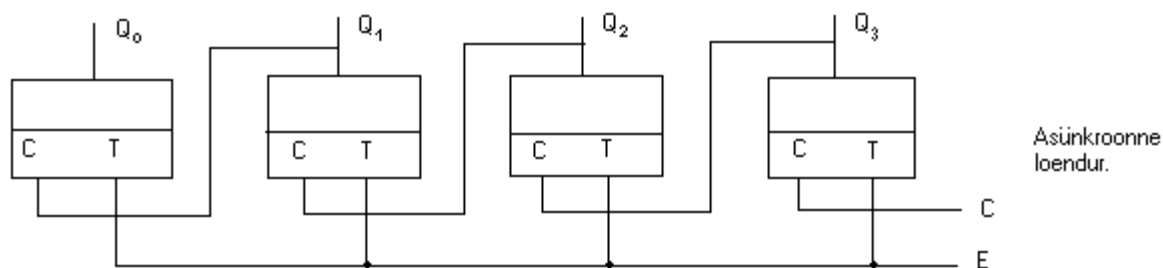
Kasut. arvutites andmetöötles.



Asünkroonne - ümberlülitusaeg pole samasugune.

Uue kombinatsiooni ilmumine sõltub sellest, missugusele üleminek toimub.

Kasut. indikatsiooniseadmetes ja sagedusjagajates.



Asünkroonne loendur.

Kahendloendur - on järjestikulised kahendkoodid.

Kümnenndloendur - järjestikuskoodid on 0-9 ja moodul on 10. See tähendab , et loenduril on 10 erinevat kombinatsiooni, millega ta vastab sissetulevale impulsijadale.

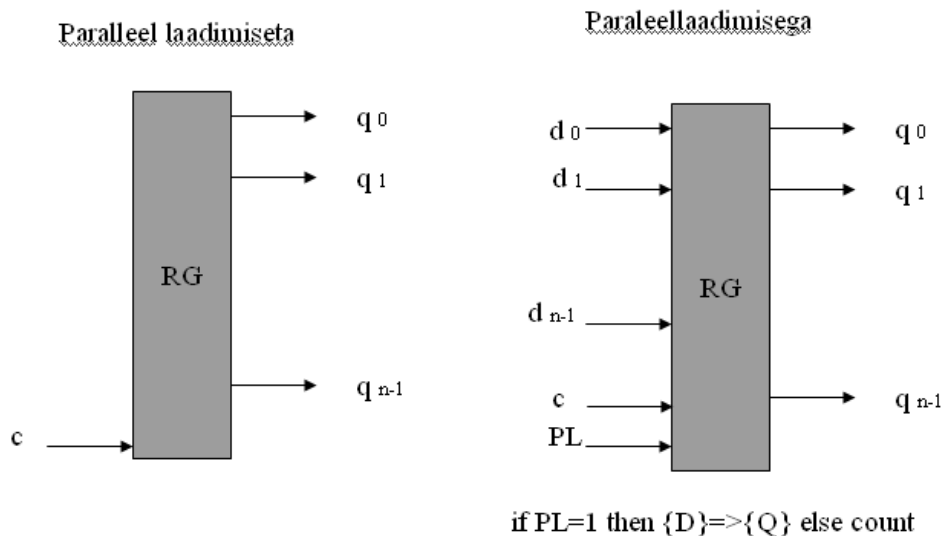
Suvalise mooduliga e. grey koodiga loendurid – kõik järgnevad koodid on naaberkoodid.

$$g = Q^i + Q^{i+1}$$

Suvalise mooduliga e. naaberkoodid on koodid, mis erinevad teineteisest ainult ühe kahendjärgu poolest. Gray koodi puhul lülitub korraka ümber ainult 1 triger.

Reversiivne loendur - Loendur, mis loendab nii pos kui ka neg suunas. Loendussuuna muutmine sõltub sellest, kas ülekandeks kasutatakse trigeri otsest või inverteeritud signaali.

Ringloendur - Loendur, mis on moodustatud nihkeregistrist, kui selle väljund ühendada sisendiga.

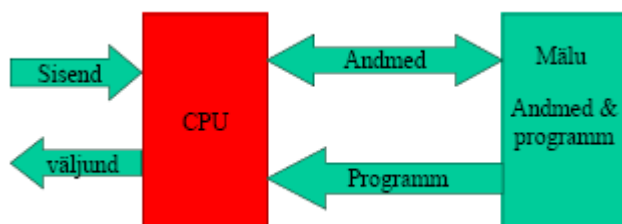


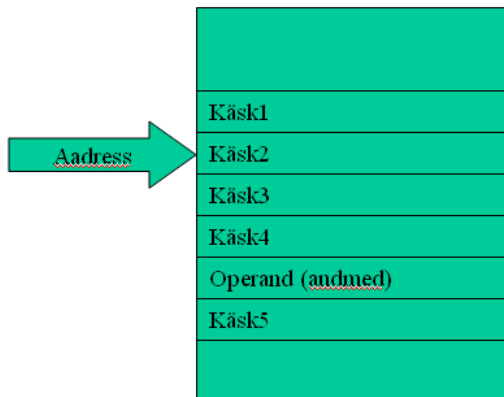
Protsessor

- Protsessori üldstruktuur**

Protsessori ja mälu osa andmetöötluses:

Arvutis säilitatakse programme (käskude jada) ja andmeid mälus kahendkujul (0-de ja 1-de jada). Põhiliselt on kasutusel von Neumanni tüüpi arvuti arhitektuur kus nii käsud kui ka andmed asuvad samas mälus. Eksisteerib ka Harvardi arhitektuur kus on eraldi mälu käskudele ja andmetele. Kogu programmi täitmine eeldab pidevat andmevahetust protsessori ja mälu vahel. Protsessorisse loetakse käske ja andmeid ning mällu kirjutatakse resultaate (andmeid, mitte käske). Sisend ja väljund ei pruugi toimuda üljuhul läbi protsessori vaid võib olla teostatud ka otse mälu ja sisend-väljund seadmete vahelise andmevahetusena.





Mälust saab lugeda ja sinna kirjutada käske-andmeid sõnade kaupa. Eri protsessoritel on erinev sõna järgulisus. Aadress on kahend kood (number) mis näitab millise sõna poole toimub pöördumine.

Mälus on taoline 0-de ja 1-de jada. Koodi enda järgi ei ole võimalik eristada kus on andmed ja kus käsud. Protsessorist peab tulema aadress mis näitab millisesõna poole pöördutakse. Lugemise korral peab juba protsessor teadma kas sõna interpreteerida käsuna või andmetena.



Kõrgtaseme keel – assembler – masinkood:

Kõrgtaseme keeles kirjutatud programmi (käskude) jada ei ole arvuti riistvara võimeline täitma. Riistvaras on olemas ainult pingnivoo mis vastab väärtusele 1 ja teine pingnivoo mis vastab väärtusele 0. Sellepärast teisendatakse programselt (transleeritakse) kõik programmid lõpuks masinkoodi. Masinkoodis vastab igale käsule oma kahendkood. Millised on transleerimise vahe etapid ja kuidas seda tehakse on tarkvaraprobleem, aga lõpuks peab ta olema masinkoodis, et protsessori riistvara saaks read täita.

Kõrgtaseme keel

If $n < 100$ then $a := b$

High-level language

else $v[i] := k[j-1]$

end;

Assembler keel

ADD a,b,c

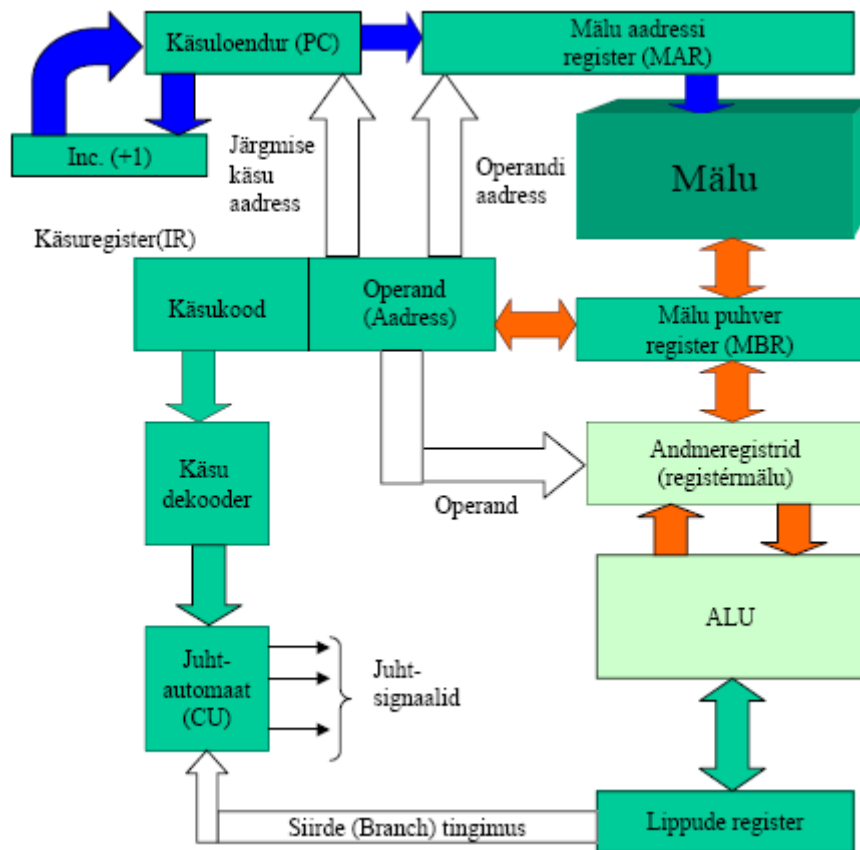
Assembly language

MOV d, M

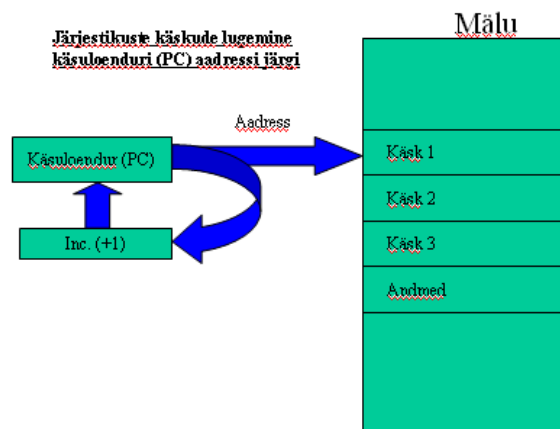
XOR c,d

Masinkood	00110111001110
Binary machine	11000110010100
language	11110100000101
	11110000011100

Järgnevalt on toodud protsessori üldstruktuur, mille erinevaid koostisosi hakkame allpool eraldi vaatama. Sellise struktuuriga protsessor võimaldab täita kõiki programme.

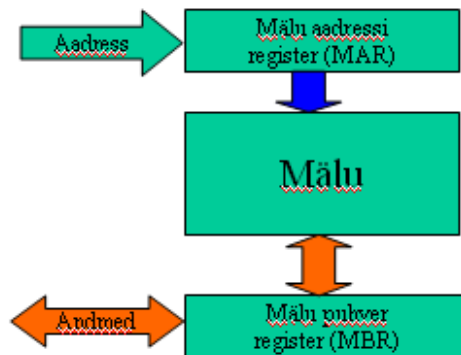


- käsuloendur (PC - Program Counter, IP - Instruction Pointer)



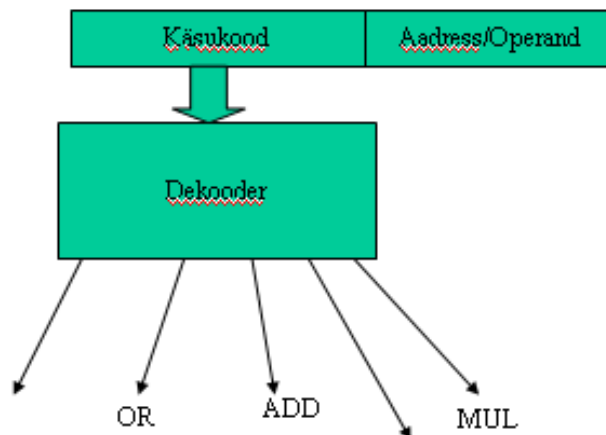
Eeldame, et meil on mälus programm (käskude jada) ja nende vahel ei ole andmeid. Üldiselt on nad segamini, kuid programmis võib ka olla selline lõik. Selle programmi lõigu täitmisel on meil vaja protsessoris “järjehoidjat”, et teada millise käsu täitmise juures ollakse. Selleks kasutatakse käsuloendurit (Intel on kasutanud ka käsuosuti (Instruction Pointer) mõistet). Loendur on siin loogikaelement, kus hoitakse järgmisena täitmisele tuleva käsu aadressi. Loendurit kasutatakse sellepärast, et temale on lihtne teha +1 (ühe võrra suurendamist) ja panna ta näitama järgmisele käsule. Käsuloenduri juures on kasutatud loenduri, kui loogikaelemendi mõistet, millel järjehoidja on realiseeritud ja käsuosuti puhul on terminis tema ülesanne. Mõlemal juhul on tegemist ühe ja sama asjaga. Käsuloendur sisaldab mingi käsu täitmisel alati järgmise käsu aadressi (mitte täitmisel oleva). Järgmise käsu aadress on näiteks vajalik katkestuste korral ja alamprogrammi poole pöördumisel, et fikseerida tagasipöörde aadress.

Mälu aadressi ja andmete registrid



Vaadeldes praegu käsu täitmise protsessi, on tehtud lihtsustusi protsessori ja mälu andmevahetuses. Mälul on aadressi register, kuhu saadetakse aadress ning puhver register, kuhu lugemisel tuleb sõna mälust (käsukood või andmed) ja kirjutamisel paneb protsessor sinna sõna, mis salvestatakse mälus vastavalt aadressile.

○ käsuregister (IR - Instruction Register)



Kui protsessor väljastab käsuloendurist (PC) aadressi ja saab mälust käsu koodi, siis salvestatakse see käsuregistrisse. Käsuregistri väljundisse on ühendatud dekodeer. Dekoodri väljunditest on iga sisendkoodi korral aktiivne ainult üks väljaund. Nüüd läheb käsudekoodril aktiivseks väljund, mis näitab millise käsu kood loeti protsessorisse. Kõik käsud sisaldavad alati käsukoodi, kuid sealjuures võib olla ka andmeid või aadress.

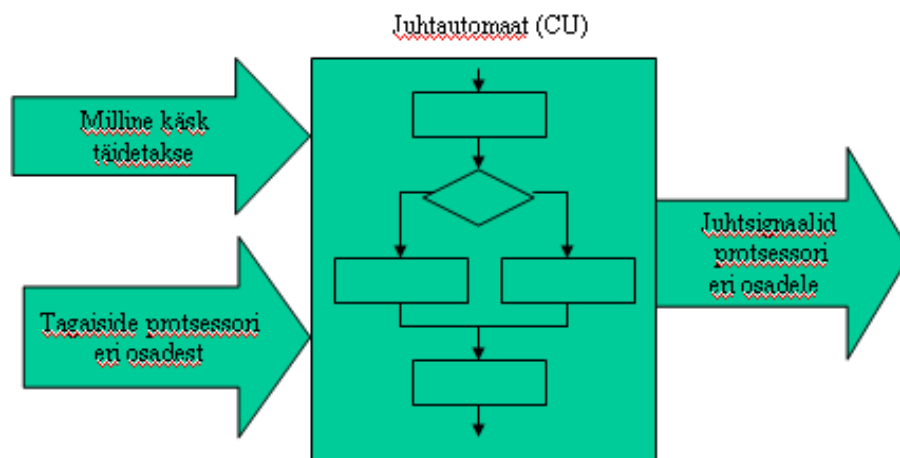
Aktiivne dekodeeri väljund näitab, millise käsu kood on käsuregistris.

- käsudekooder (Instruction Decoder)

Toodud eelmises punktis käsuregistriga koos.

- juhtautomaat (CU - Control Unit)

Jäiga loogikaga juhtautomaat *Hard Wired*



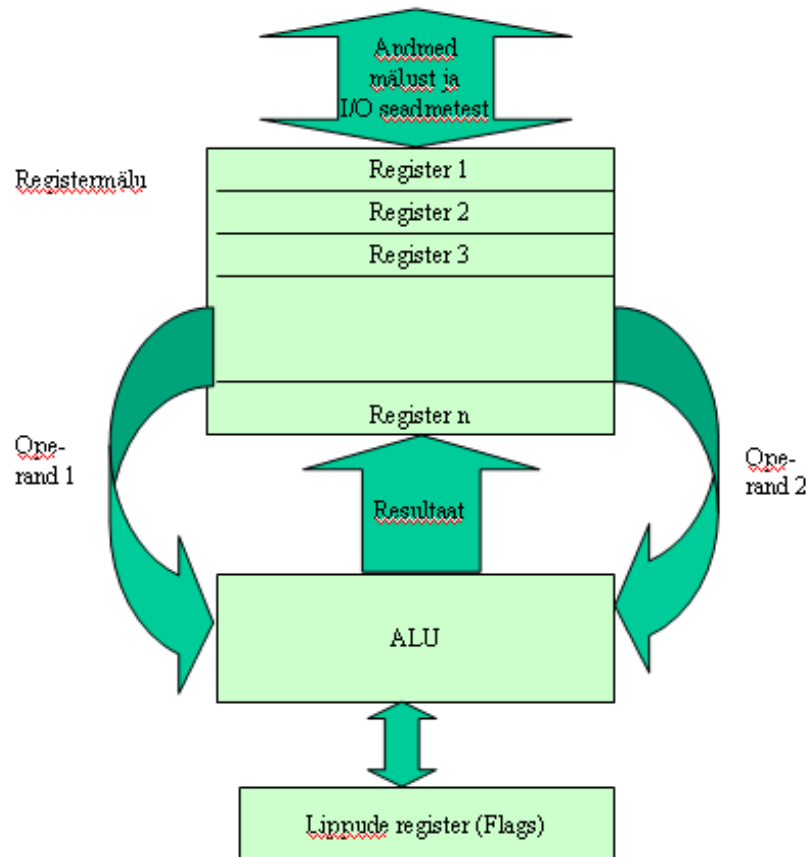
Juhtautomaat kujutab endast käsu täitmise algoritmi riistvaralist realiseerimist loogikaskeemina. Peale üldosa vastab igale käsule, mida protsessor on võimeline täitma (kuulub tema käsusüsteemi), algoritmis oma haru. Käsu dekodeerimise järgi toimub mikroprogrammis hargnemine. Selle hargnemise realiseerimiseks peab juhtautomaati tulema käsudekoodrist info selle kohta, milline on täitmisele tulev käsk. Mõnede käskude täitmisel on vaja realiseerida mikroprogrammis hargnemisi, mis sõltuvad protsessori mõne teise osa seisundist. Näiteks on vaja teada korrutamise realiseerimisel liitmise ja nihutamise abil eelneva ALU operatsiooni tulemuste võrdumist nulliga.

Põhimõtteliselt on juhtautomaadi realiseerimiseks kaks võimalust:

1) *jäiga loogikaga juhtautomaat*: Jäiga loogika korral realiseeritakse algoritm loogikaskeemina kristalli pinnal ja iga muutus käsusüsteemis tähendab uue loogikaskeemi realiseerimist.

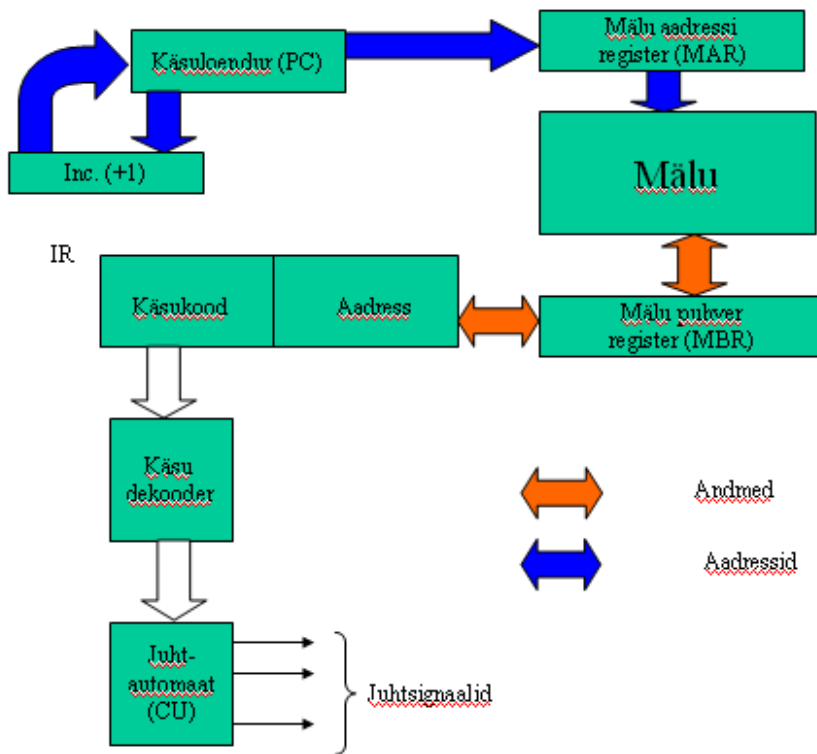
2) *püsimalus säilitatava mikroprogrammiga juhtautomaat*: Kui mikroprogrammi hoitakse püsimalus (näiteks Flash), siis saab käsusüsteemis teha muudatusi ilma uut loogikaskeemi koostamatta. Kogu mikroprogrammi täitmine taandub sõnade lugemisele mikroprogrammi sisaldavast püsimalust. Mingil määral toimib see analoogiliselt programmi täitmisega protsessoris.

- **operatsioonautomaat (*Data Path*)**



Operatsioonautomaat (OA) on vahetu andmete teisendaja. OA koosneb registermälust, ALU-st ja lippude registrist. Registermäl on väga kiire (SRAM tehnoloogia) ja väikesemahuline mälu vahetult teisendatavate operandide ja resultaate salvestamiseks. Kuidas leida operandid ja kuhu salvestada resultaate, määratakse käsukoodiga, mis käivitab juhtautomaadis vastava algoritmi. Mõne käsu täitmisel võivad operandid läbida korduvalt. Näiteks kui ALU-s puudub korrutamise riistvaraline realiseerimine, saab seda teha ka nihutamise ja liitmise abil. Lippude registries säilitatakse info alus teostatud operatsioonide tulemuste kohta (tulemus oli null, tekkis ületäitumine, tekkis ülekantne jne.).

- **Käsu täitmine protsessoris (Instruction Execution, fetch-decode-execute cycle)**

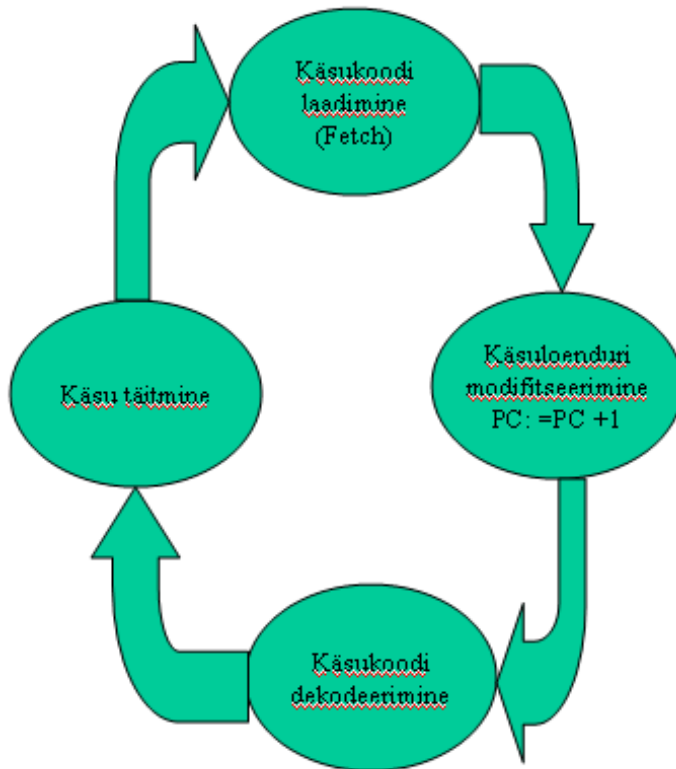


Järgnevalt on toodud käsu täitmise juhtimine protsessoris:

Pärast käskukoodi dekodeerimist asub käskukoodi edasist täitmist juhtima juhtautomaat. Iga käsu täitmine koosneb teatud hulga elementaar operatsioonide (mikrooperatsioonid) teostamisest. Mikrooperatsioone täidetakse teatud algoritmi (mikroprogramm) alusel. Juhtautomaat kujutab endast käsu täitmise algoritmi riistavaralist realiseerimist. Kõikide käskude täitmise algoritmidel on alguses ühisosa (käskukoodi lugemine, käsuolenduri modifitseerimine jne) ja pärast dekodeerimist täidetakse spetsiaalosa (operandide lugemine, ALU operatsioonid, resultaadi salvestamine jne.).

Käsu täitmise tsükkel (von Neumanni tsükkel):

Inglise keeles kasutatakse ka nimetust fetch-decode-execute cycle. Alumisel pildil on kogu käsu täitmine võetud kokku ühe tsükliks.



• RISC - CISC protsessor

Protsessorid võib oma ideoloogia järgi jagada kaheks : **RISC** -*Reduced Instruction Set Computer* ja **CISC** -*Complex Instruction Set Computer*. Nagu nimetusest näha on ühel ideoloogia protsessoril keerukas käsusüsteem ja teisel lihtsam. Lihtsamaid käsked on võimalik täita kiiremini kuid neid käsked kulub mingi programmi juures rohkem. Idee on selle, et kui CISC täidab käsu näiteks 10 ajaühikuga, siis RISC näiteks 1 ajaühikuga, kuid aga sama tegevuse teostamiseks kulub RISC protsessoris näiteks 5 korda rohkem käsked. Saame tulemuse, et RISC on ikkagi kaks korda kiirem (CISC-l 10 ajaühikut ja RISC-l $1 \times 5 = 5$ ajaühikut). Loomulikult eeldab suurema hulga käskude lugemine suuremat arvu mälu poole pöördumisi, mis kulutavad ka aega. RISC ideoloogia hakkas levima kaheksakümnendate aastatest. Praegu kasutatakse hübriidstruktuure mis ei ole puhas RISC ega CISC. Näites RISC tuuma ümber on ehitatud CISC kest. RISC protsessorites on palju tegeldud käskude täitmise efektiivsuse tõstmisega.

RISC protsessori projekteerimise põhimõtted:

- suhteliselt vähe käsked (eelistatavalt alla 100) ja vältida tuleb keerulisi käsked
- vähe adresseerimise viise (eelistatavalt üks või kaks)
- vähe erinevaid käsu formaate (eelistatavalt üks või kaks), et kiirendada dekodeerimist
- kõik käsud tuleb täita otse riistvaras ühe taktiga (täidetakse ALU-s)
- maksimaalne käskude täitmise kiirus, mis viib paralleelsuseni

- ainult LOAD ja STORE käsud pöörduvad mälu poole
- võimas registermälu (ulatudes 32 kuni 132-ni), et võimalikult palju oleks register-register tüüpi käske ja vähe pöördumisi mälu poole
- jäiga loogikaga (hardwired) juhtautomaat, mis võib ka tehnoloogia arenedes asendada mikro-programmeeritavaga
- efektiivne andmevahetus alamprogrammidega
- efektiivne käskude järjekorra juhtimine (siirded ja alamprogrammid)

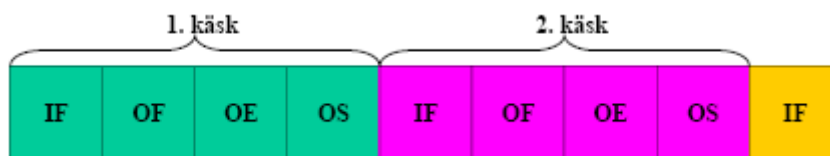
Oluline on käsu täitmisel, et seda teostatakse riistvaras. Näiteks kui ALU ei oska riistvaras korrutada. Nüüd võime teha keruka mikroprogrammi spetsiaalosa mis realiseerib selle liitmise ja nihke kaudu. RISC ideoloogia sellist keerukat (aeglast) mikroprogrammi ei luba. Korrutamine teostatakse riistvaras ALU-s või ta üldse puudub. Mikroprogrammis on olemas üldosa koos käsukoodi lugemise ja käsuloenduri modifitseerimisega ning operandide lugemine ja resultaadi salvestamine, kuid ta puudub täitmisel.

• Konveier protsessoris (*Pipeline*)

Käsu täitmise protsessoris võib jagada teatud sõltumatuteks etappideks. Näiteks on siin käsk jagatud neljaks etapiks:

- 1) IF Instruction Fetch (Käsu laadimine) + Instruction Decode
- 2) OF Operand Fetch (Operandi laadimine)
- 3) OE Operand Execute (Operatsiooni täitmine ALU-s)
- 4) OS Operand Store (Resutaadi salvestamine)

Programmi täitmine ilma konveierita:



Vaadates nüüd käsu täitmist selgub, et iga etapi täitmisel töötab ainult 20% riistvarast, samas kui 80% ei tee midagi. Siin on üks käskude täitmise efektiivsuse tõstmise võimalus, mille on arvutustehnikasse toonud RISC ideoloogia, nimelt **konveier**.

Teeme nüüd protsessori nii, et need neli käsu täitmise etappi oleksid kõik sõltumatud, ligilähedaselt sama kestvusega. Nüüd saame peale seda kui esimene käsk on läbinud esimese etapi ja jõudnud teise, alustada juba teise käsu juures esimese etapi täitmist. Kuivõrd etapid on sõltumatud saame siis, kui esimene käsk on kolmandas etapis ja teine käsk teises alustada juba kolmanda käsu juures esimese etapi täitmist jne. Seega ei ole siin suurenenud ühe käsu täitmise

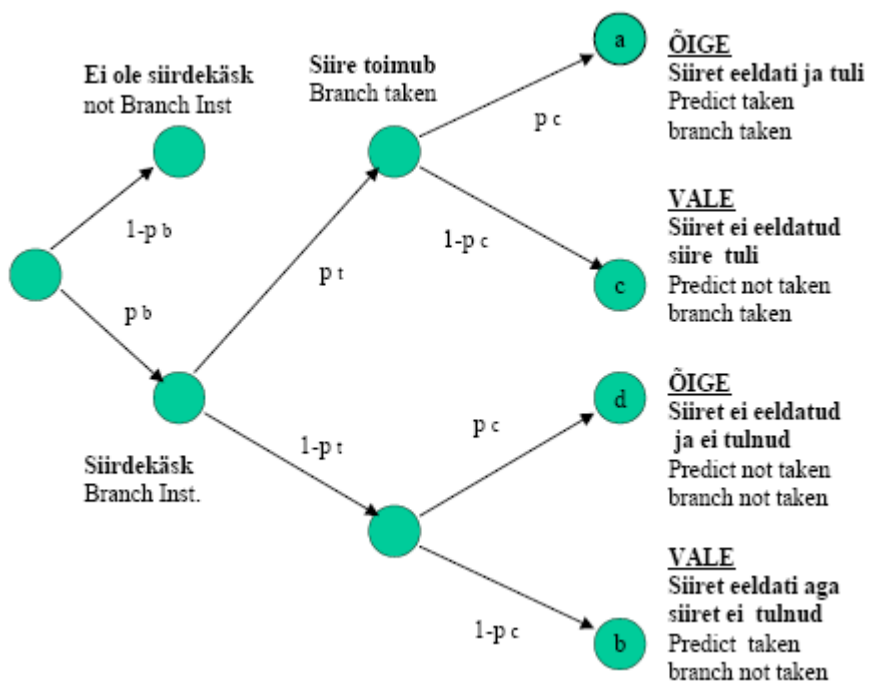
kiirus kuid tänu käskude täitmise paralleelsusele täidetakse neid keskmiselt ajaühikus rohkem. Samuti on siin kogu protsessor pidevalt koormatud. Analoogiline on konveieri töö tootmises.

Konveieriga programmi täitmine (Pipeline):

Käsk	Takt 1	Takt 2	Takt 3	Takt 4	Takt 5	Takt 6	Takt 7	Takt 8
1	IF	OF	OE	OS				
2		IF	OF	OE	OS			
3			IF	OF	OE	OS		
4				IF	OF	OE	OS	
5					IF	OF	OE	OS

- **Siirete (hargnemiste) ennustamine.(Branch Prediction)**

Protsessorites on tihti eraldi loogikaskeem mis tegeleb **hargnemiste ennustamisega**, et muuta konveier efektiivsemaks. Hargnemise ennustamine toimub teatud statistiliste kriteeriumite järgi ja ei saa anda alati õiget tulemust, kuid siiski suudab vähendada konveieri uuesti käivitamise vajaduse tõenäosust.

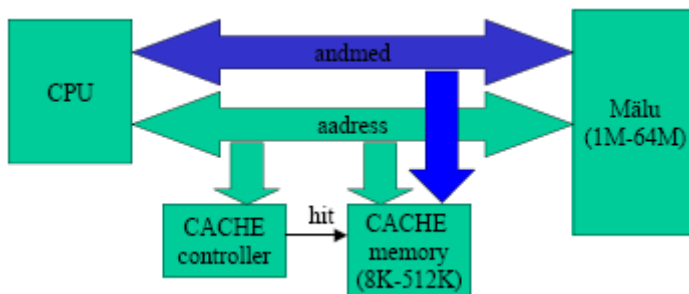


$$C_{ave} = a(p_r p_c) + b(1-p_r)(1-p_c) + c p_r(1-p_c) + d(1-p_r)p_c$$

• Peidikmälu, vahemälu (Cache)

Pöördumine mälu poole on protsessori töökiirusega võrreldes väga aeglane. Tehniliselt võiks teha mälu, mis töötaks protsessori taksagedusega, aga see oleks vajalike mälumahtude juures liialt kallis. Lahenduseks on **vahemälu** kus, hoitakse tihedamalt kasutatavat osa programmist ja andmeid. Tegemist on millegi telefoni märkmiku sarnasega, kus enamkasutatavad numbrid on märkmikus, ülejäänud aga telefoniraamatus. Alati on mõni number märkmikus vaja asendada teisega, kui teda enam ei kasutata ja kui mõnda numbrit on tihedamalt vaja võib ta ju alati telefoniraamatust kirjutada märkmikku.

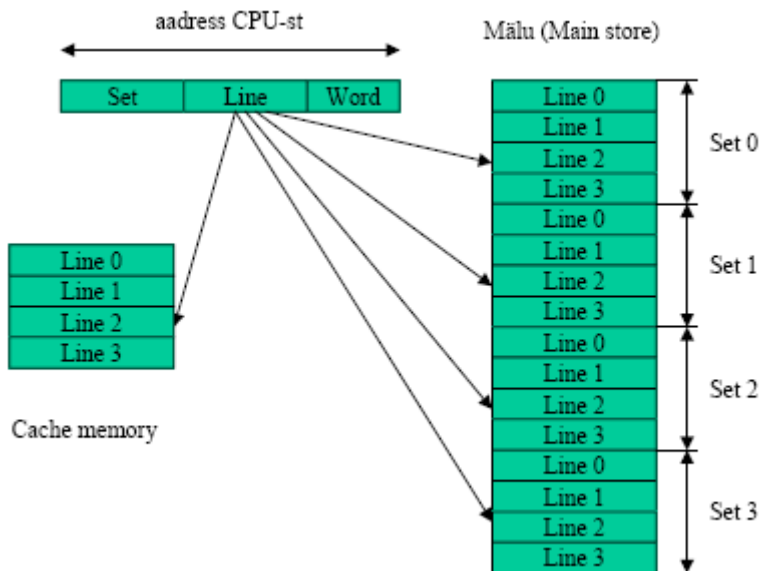
Analoogiliselt hoitakse kiires ja väikesemahulises vahemälus just tihedalt kasutatavat osa programmist ja andmetest. Alati on võimalik mõni osa vahemälust asendada infoga põhimälust. Loomulikult peab olema tagatud, et kui midagi vahemälus muudetakse (protsessor kirjutab resultaadi), siis muudetakse ka vastavat osa põhimälus.



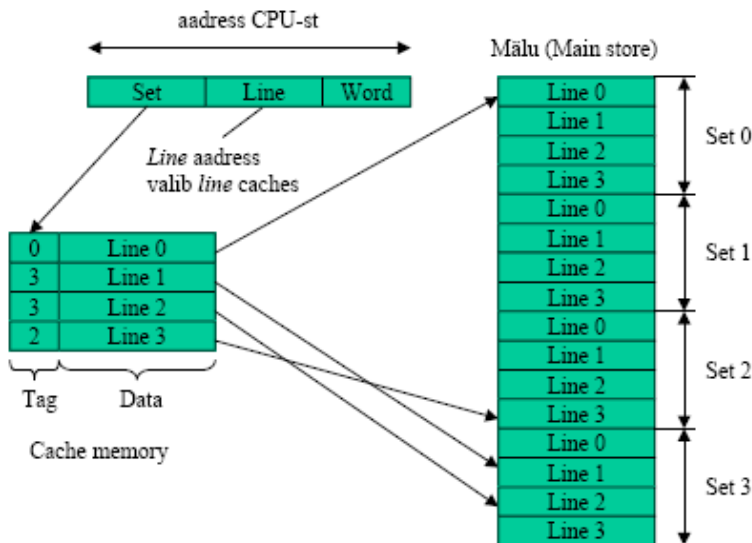
Otsevastavusega vahemälu (Direct-mapped cache):

Üks lihtsamaid vahemälu organiseerimise viise on **otsevastavusega vahemälu**. Info lugemine mälust toimub gruppidega (Line). Mälu on jagatud segmentideks (Set). Vahemälus on igale grupile oma koht. Antud näites on iga segment jagatud neljaks grupiks. Seega on vahemälu ruumi neljale grupile. Korraga saab vahemälus olla üks 0-s grupp, üks 1-ne grupp, üks 2-ne grupp ja üks 3-s grupp. Seejuures võib iga grupp olla ükskõik millisest segmentist. Korraga ei saa olla aga kahte 0-ndat gruppi, 1-st gruppi, 2-st gruppi ega 3-ndat gruppi erinevatest segmentidest.

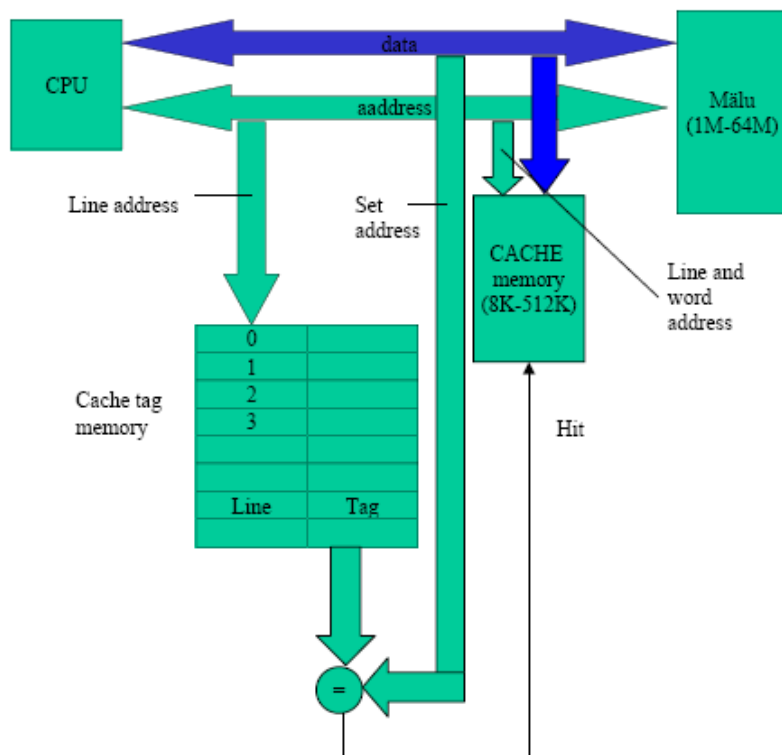
line – minimaalne vahemäluga vahetatav infohulk (4 – 16 baiti)



Selleks, et oleks võimalik kindlaks teha millisest segmendist mingi grupp on võetud, on vahemälus iga segmendi juures vastava segmendi number. Adresseerimine on lihtne ja ka suhteliselt odav, kuid kui vaja on näiteks 0-nda grupi infot erinevatest segmentidest, tuleb pidevalt teha ümberlaadimist.

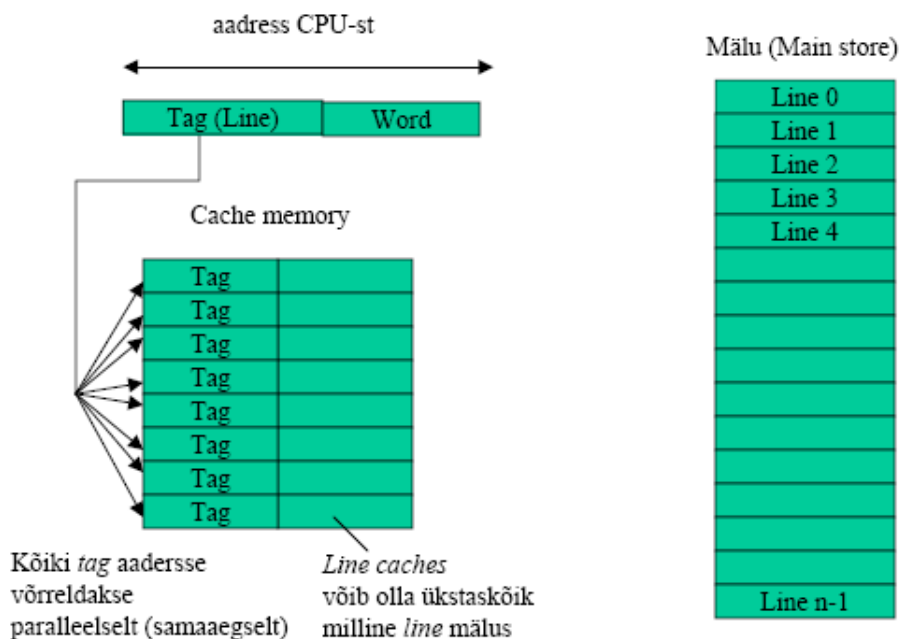


Vahemälu kontrollib mälu poole pöördumisel, kas vastava grupi juures on soovitud segmendi number. Kui on kokkulangemine (hit), siis pöördub protsessor vahemälu poole. Kui aga ei lange kokku, asendatakse vastav grupp teisega. Vajadusel (kui protsessor on antud gruppi midagi kirjutanud) kopeeritakse asendatav grupp ka põhimällu.



Assotsiatiivne vahemälu (Associative mapped cache):

Assotsiatiivne vahemälu ei ole jägatud segmentideks. Endiselt on olemas grupid. Nüüd võib vahemälus olla ükskaskõik milline grupp mälust. Otsimine vahemälust toimub grupi numbri järgi. Grupid ei ole järjestatud ja sellepärast kasutatakse assotsiatiivmälu. Siin on vähem vaja põhimälu poole pöördumisi, kuid mälu ise on kallim.



Otsevastavusega vahemälu korral oli alati teada, milline grupp tuleb asendada. Assotsiatiivse vahemälu puhul tuleb see otsustada kontrollerial. Kasutusel on terve rida vastandlikke strateegiaid. Assotsiatiivse vahemälu uuendamise strateegiad :

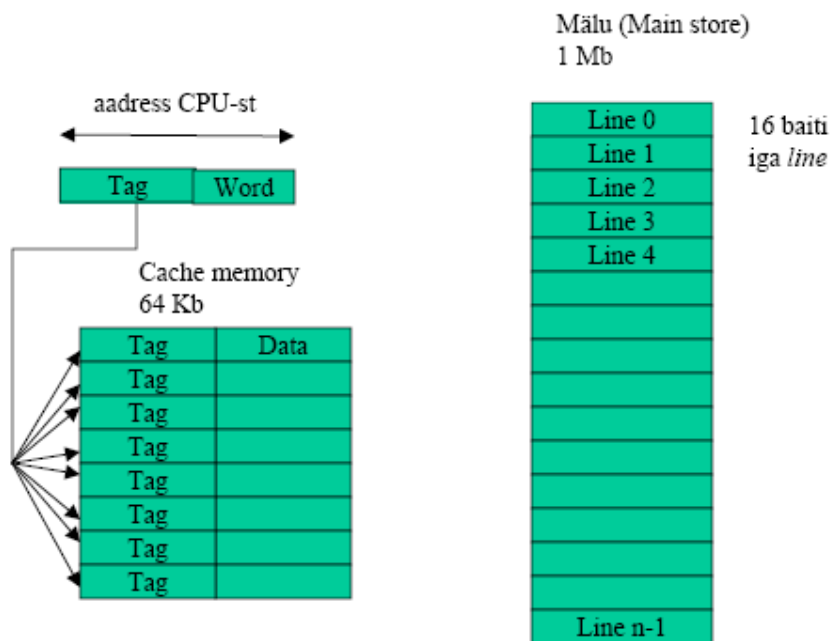
- LRU -Least Recently Used
- LFU –Least Frequently Used
- FIFO –First In First Out (kauem olnud andmed välja)
- Random –juhuslik

Alati peab olema tagatud, et kui protsessor on infot vahemälus muutnud, siis jõuaks vastav muutus ka põhimällu. Vahemälu peab olema põhimälu koopias. Kasutusel on kaks erinevat strateegiat. Andmete kirjutamine vahemälust põhimällu :

- Write-through, korraga muutused kirjutada vahemällu ja põhimälu (486)
- Write-back, kirjutatakse põhimällu vahemälu grupi asendamisel.

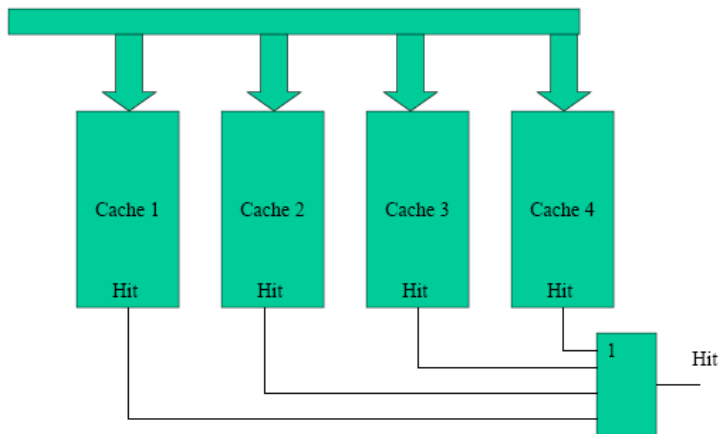
Näide: Line on 16 baiti s.o. neli 32 bitist sõna. Mälu on 1 Mb ja seega koosneb 64 K linest.

Cache koosneb 4096-st linest. Et laetud cachesse võib olla laetud ükskõik milline 64 K linest on tag 16 bitine.

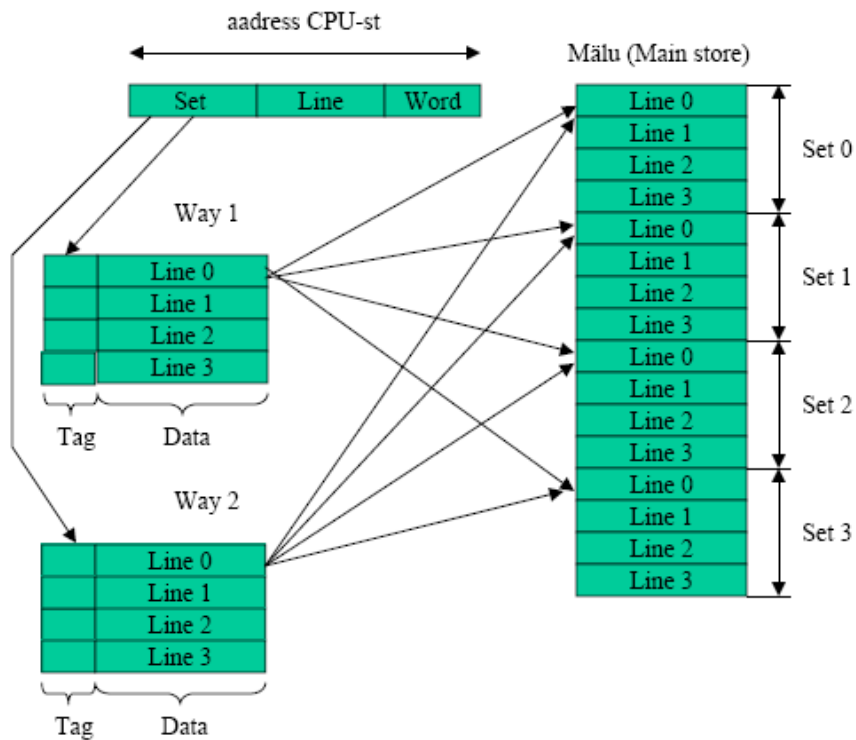


Kogumassotsiatiivne vahemälu (Set associative-mapped cache):

Set associative-mapped cache –kompromiss *direct-mapped* ja täielikult *associative cache* vahel.



Näide: 2-way set assotciative cache.

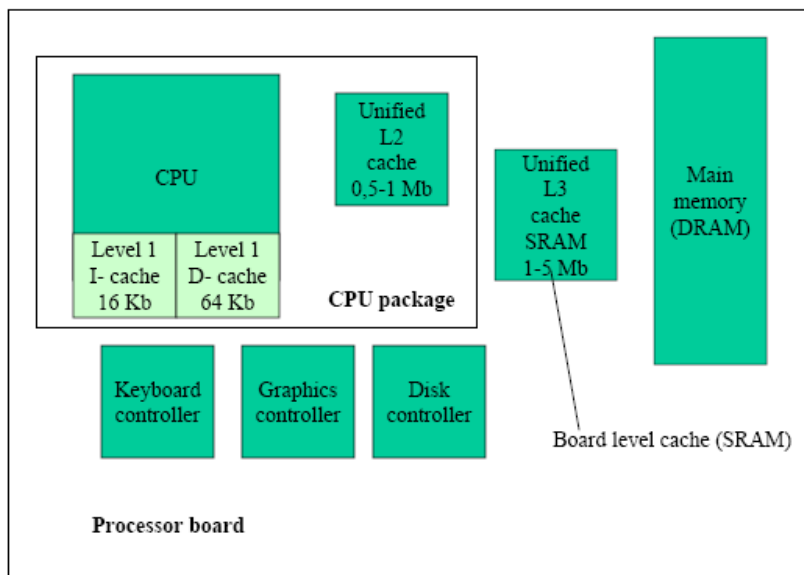


Vahemälu tasemed:

Level 1 cache - tavaliselt CPU korpuse sees. Käsu peidikmälu (Instruction Cache) umbes 16 Kb ja andmete peidikmälu (Data Cache) umbes 64 Kb.

Level 2 cache - väljaspool CPU korpust. Tavaliselt segamini andmed ja käsud. Maht umbes 0,5 kuni 1 Mb.

Kolmetasandilise peidikmälu arvuti:



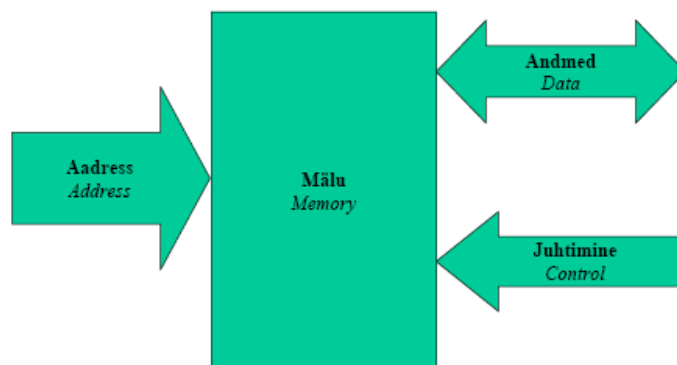
Arvuti mälu

Mälu üldstruktuur:

Mälu poole pöördumisel määrab aadress, millise mälusõna poole toimub pöördumine. Seejuures mälu sõna on info hulk, mille kaupa on võimalik mälusse kirjutada või sealt infot lugeda. Kui näiteks mälu sõna on 8 bitti (üks bait), siis saab lugeda või kirjutada infot baidi kaupa. Samal ajal ei saa lugeda/kirjutada ühte bitti või ühe pöördumisega lugeda/kirjutada 16 bitti infot. Mälusõna on kvant infot, millele viitab üks aadress (kahendkood) ja mille kaupa toimub igal mälu poole pöördumisel infovahetus.

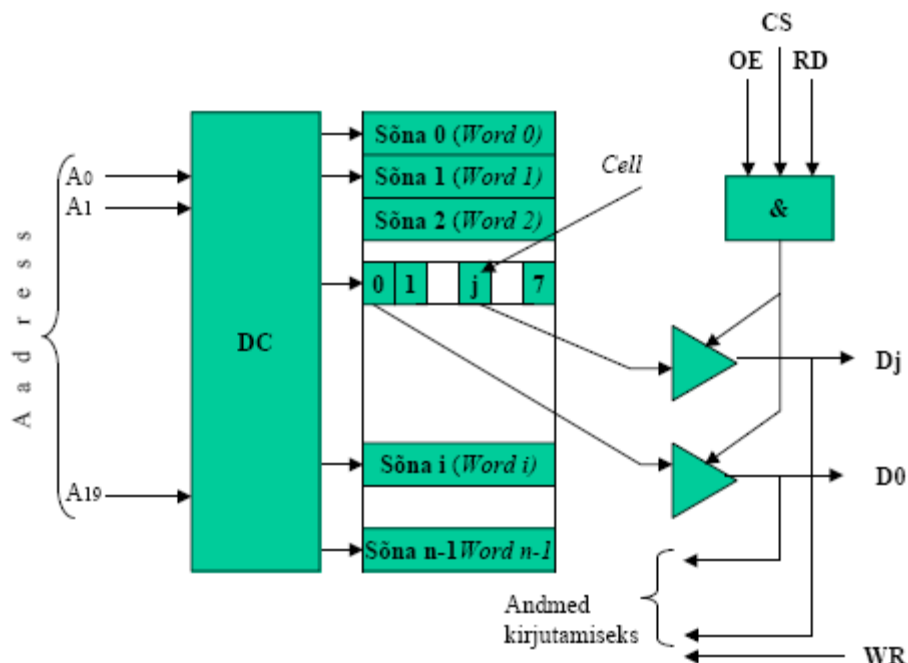
Infovahetuseks on mälu andmeliinid. Andmeliinide arv (andmesiini järgulusus) vastab tavaliselt mälusõna järgulususele. Andmevahetuseks

protsessori ja mälu vahel on veel juhtliinid (juhtsiin). Minimaalsed juhtsignaalid on mälu kirjutamine (MEMORY WRITE) ja mälu lugemine (MEMORY READ). Teatud mälu tüüpidel on muidugi veel täiendavaid juhtsignaale.



Mälu poole pöördumisel määrab aadressi dekodeer, millise mälupea poole toimub pöördumine. Dekodeeril oli teatavasti omadus, et iga sisendkombinatsiooni korral on aktiivne ainult üks väljund ja sellega on ka tagatud, et pöörduda saab korraga vaid ühe mälusõna poole. Mälu koosneb tavaliselt mitmest moodulist ja CS (Chip Select) signaal on mälu mooduli valikuks. Kui CS ei ole aktiivne, on kõikide mälu andmeliinide puhvrid kolmandas olekus ja kirjutamine antud moodulisse keelatud.

Sisendid RD (Read) ja WR (Write) on vastavalt lugemise ja kirjutamise lubamiseks. Tihti kasutatakse siin ka ühte liini mille puhul näiteks madal väärtus (0) näitab kirjutamist ja kõrge väärtus (1) lugemist. Paljudel mäludel on veel täiendav sisend OE (Output Enable) mis võimaldab viia andmeliinide puhvrid kolmandasse olekusse.



Mõisteid:

Cell (pesik) -elementaarne mälu komponent 1 biti info säilitamiseks.

Word (pesa, sõna) -mälu ühik millele viitab üks aadress.

Access time (pöördumis aeg) -aeg mälu poole pöördumise algusest kuni andmete saamiseni

Bandwidth (läbilaskevõime) -andmeedastus kiirus mälu ja CPU vahel (Mbaiti sekundis).

Latency (latensusaeg) -aeg mälu poole pöördumise algusest andmete saabumise alguseni

Random access memory –suvapöördusmälu(iga sõna poole pöördumine nõuab ühepalju aega sõltumatta tema sukohast mälus)

Serial access memory –jadapöördusmälu(erinevate mälu sõnade poole pöördumine nõuab erineva aja)

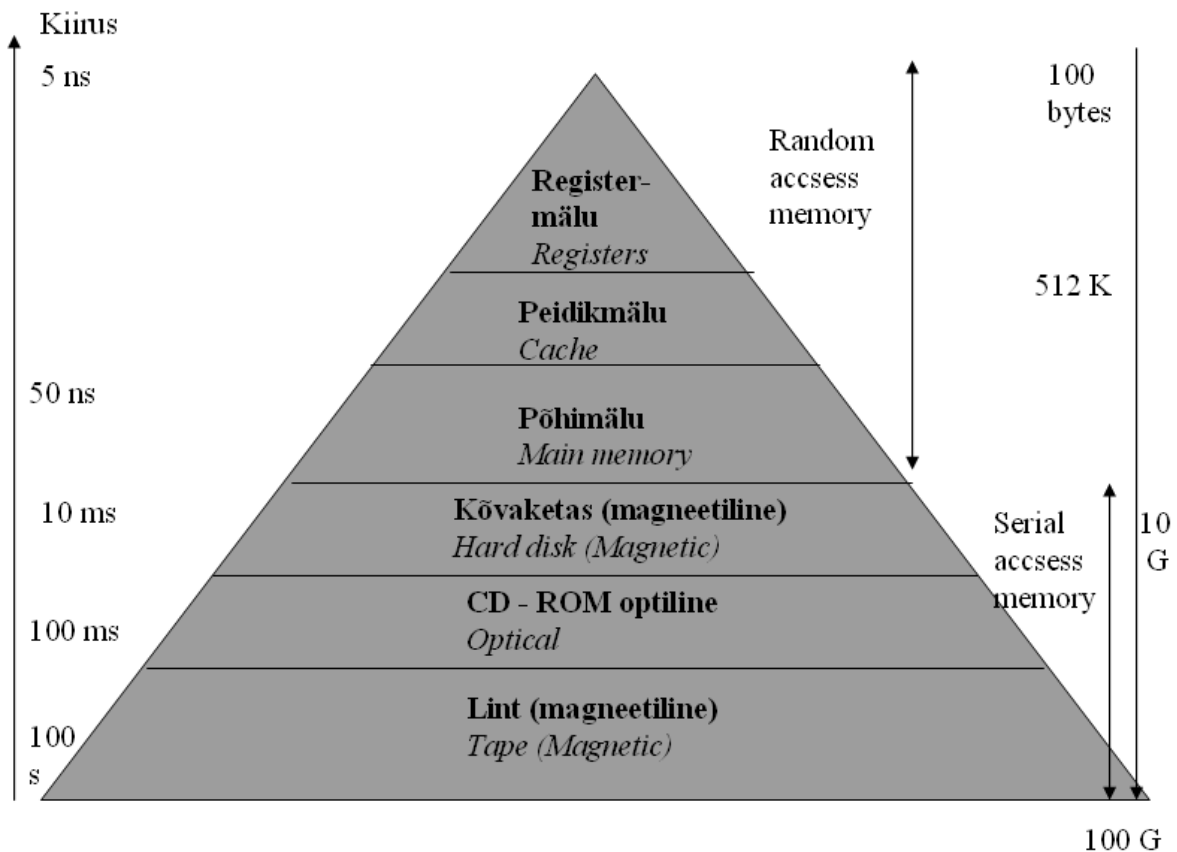
Volatile memory (hävimismälu) -andmed hävivad toite väljalülitamisel

Read-only memory (ROM) –püsimälu(suvapöördus mälu liik kus info säilib ka üärast toite välja lülitamist)

Static memory -staatiline mälu

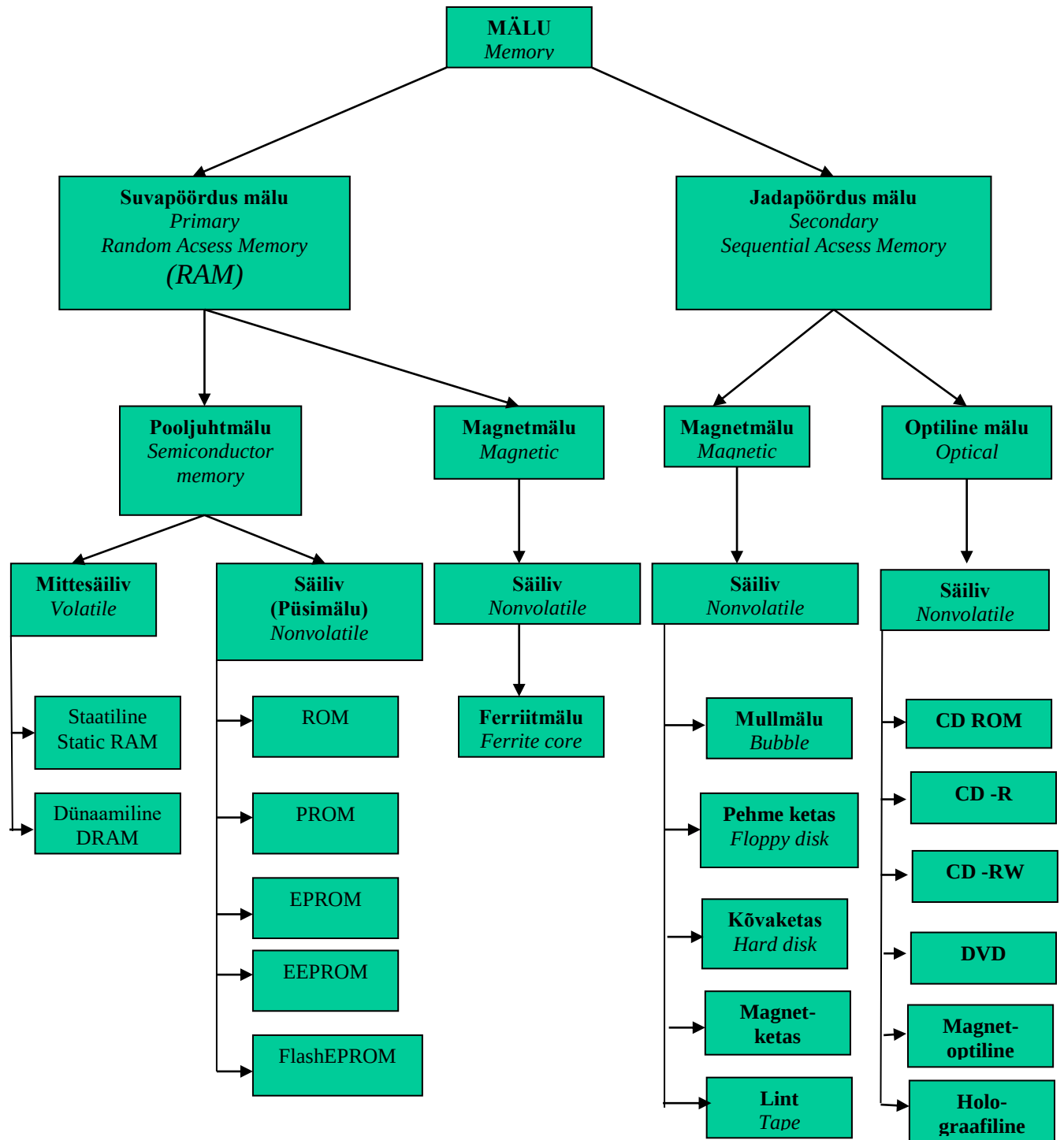
Dynamic memory -dünaamiline mälu

- **Mälu hierarhia arvutis (*Memory hierarchy*)**



Mälu hierarhias on tipus suhteliselt väikese mahuline, kuid kiire registermälu. Registermälu on suhteliselt kallis ja sellepärast tema maht on ka piiratud. Töötab ta protsessori kiirusega. Järgneb vahemälu (peidikmälu, Cache) mis on juba suurema mahuga, aga ka mõnevõrra aeglasem. Esimesed kaks on realiseeritud reeglina staatilise suvapöördus mäluna mis on kiirem dünaamilisest. Põhimälu on dünaamiline suvapöördus mälu mis tagab suurema pakkimistiheduse kristallil kui dünaamiline, kuid on ka aeglasem. Järgnevad juba järjesti pöördusega mälud mis on veelgi aeglasemad, kuid suurema mahulised

- **Arvuti mälu klassifikatsioon (Computer memory classification)**



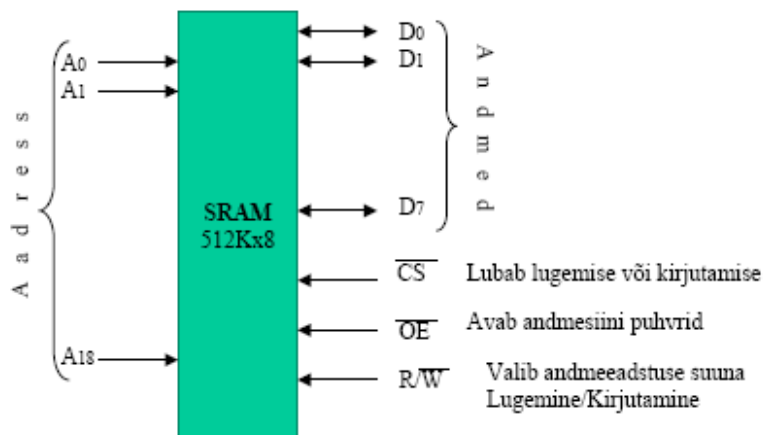
- **Muutmälu (RAM)**

Muutmälu on seade informatsiooni lühiajaliseks salvestamiseks, säilitamiseks, otsinguks ning lugemiseks. Muutmälud jagunevad staatilisteks ja dünaamilisteks. Muutmälude (RAM- Random

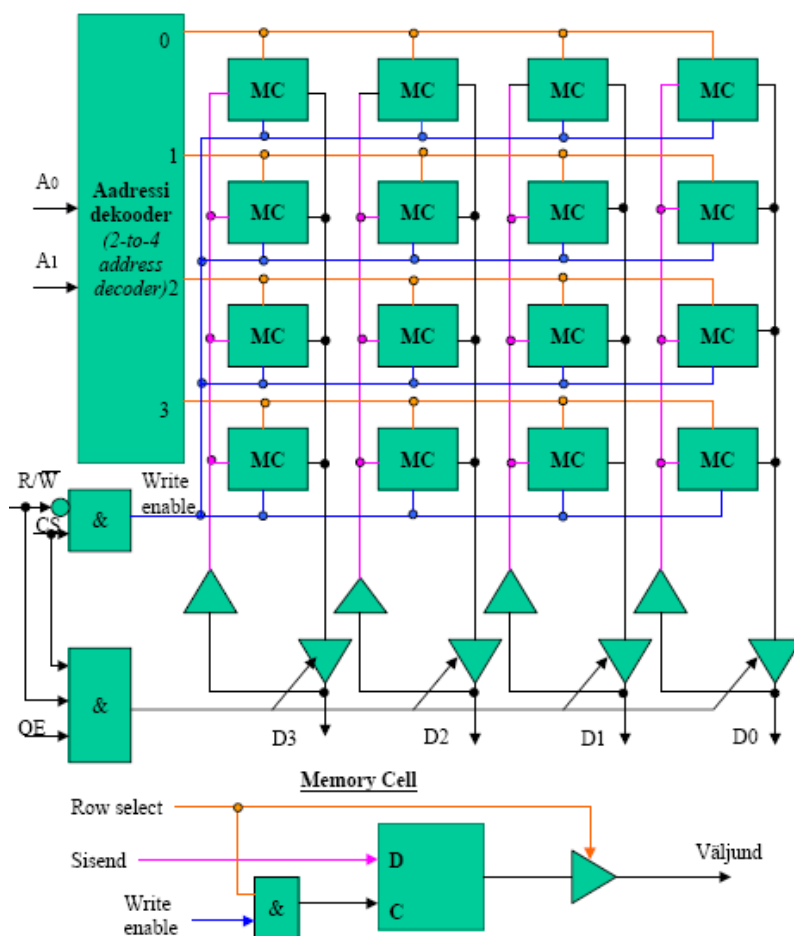
Access memory) põhiliigiks on pooljuht mälud, mis koosnevad trigeritest või muudest mäluelementidest. Muutmälud on toitepingest sõltuvad.

- **Staatiline pooljuht suvapöördusmälu (Static RAM)**

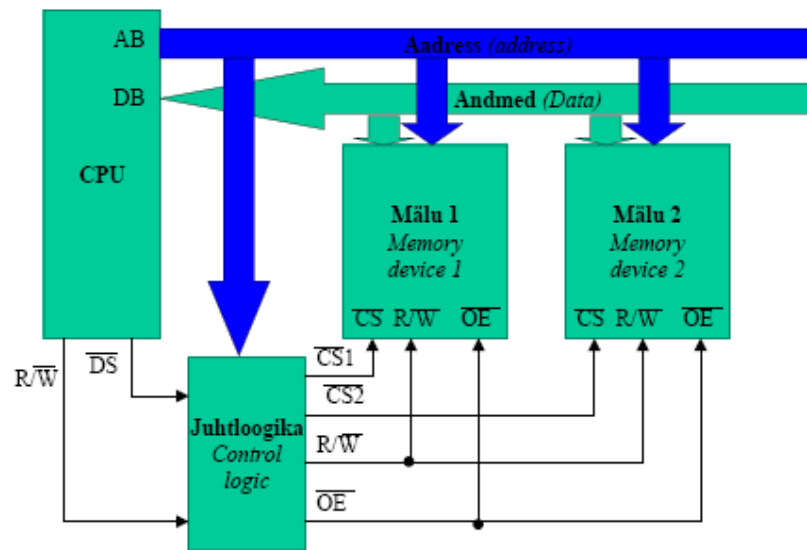
Staatilises pooljuht suvapöördusmälus (SRAM) on info salvestatud positiivse tagasiside kaudu trigerites. Tegemist on kiire mälu, mida kasutatakse näiteks registermälu ja vahemälu (Cache). Trigerite pakkimistihedus kristallil jääb väiksemaks kui dünaamilisel RAM-l. Joonisel on toodud ka tavaliselt SRAM-i juhtimiseks kasutatavad signaalid.



Järgmisel joonisel on toodud SRAM-i sisemine struktuur. Mälu pesikud on kujutatud D-trigeritena, kuid realiseeritakse nad kristalli pinnal muidugi transistoride abil.

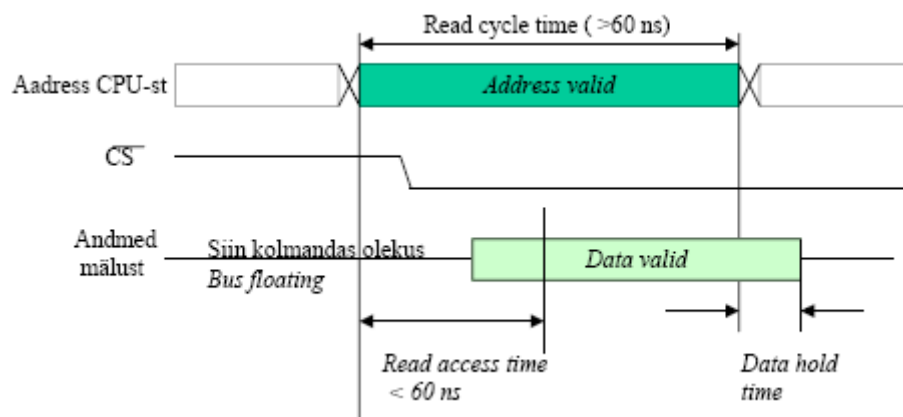


Mitu mälu moodulit:

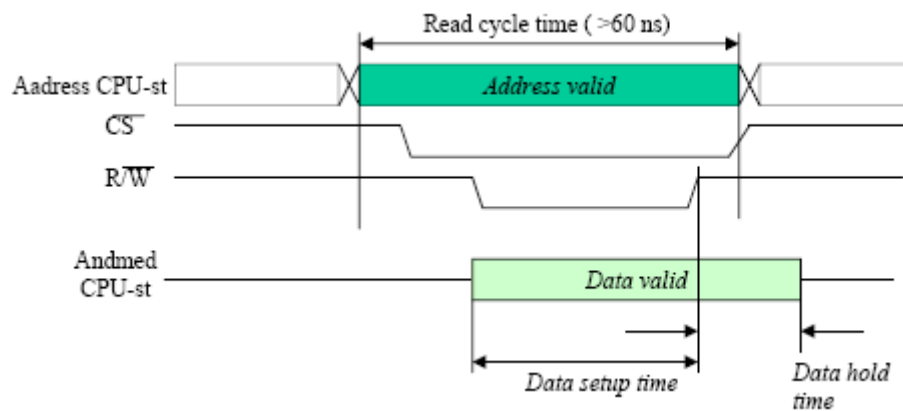


SRAM-i poole pöördumine:

Staatilisest mälust lugemise tsükkel (Read cycle of static RAM)



Staatilisse mälu kirjutamise tsükkel (Write cycle of static RAM)

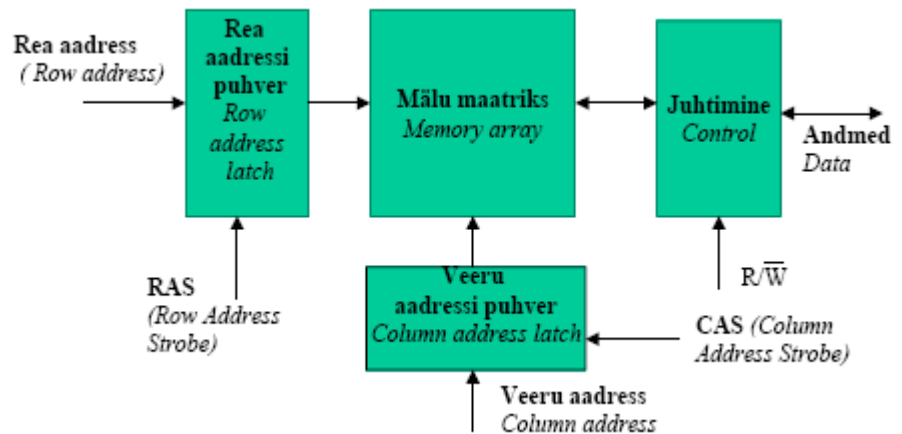


- **Dünaamiline pooljuht suvapöördusmälu (Dynamic RAM)**

DRAM-s on info kandajaks laeng. Kui SRAM-is kulub ühe biti info hoidmiseks kuni kuus transistori, siis siin on vaid üks transistor koos kondensaatoriga. Info pakkimise tihedus kristallile on parem kui SRAM-l. Kuivõrd ei ole olemas ideaalset isolaatorit, siis laeng teatud aja möödudes kaob ja info hävib. Selle vältimiseks toimub dünaamilises mälus pidev mälu värskendamine (refresh) mille käigus kirjutatakse pidevalt infot uuesti üle. SRAM-st odavama hinna tõttu kasutatakse DRAM-i just suurema mahulise põhimälu valmistamiseks. DRAM on aeglasem kui SRAM. Mälu moodulite mahud on siin suured (väike biti pindala), kuid mikroskeemile ei ole võimalik teha piisaval hulgal väljaviike. Sellepärast jagatakse tavaliselt DRAM-i aadress kaheks osaks – rea aadress ja

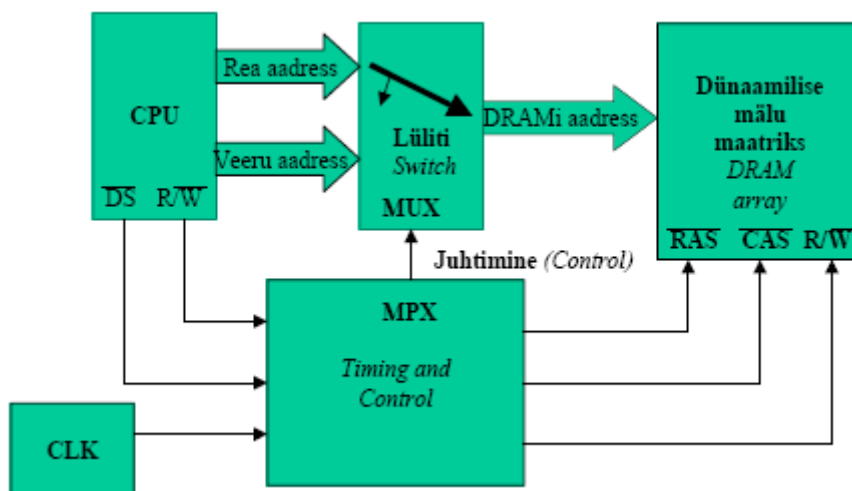
veeru aadress.

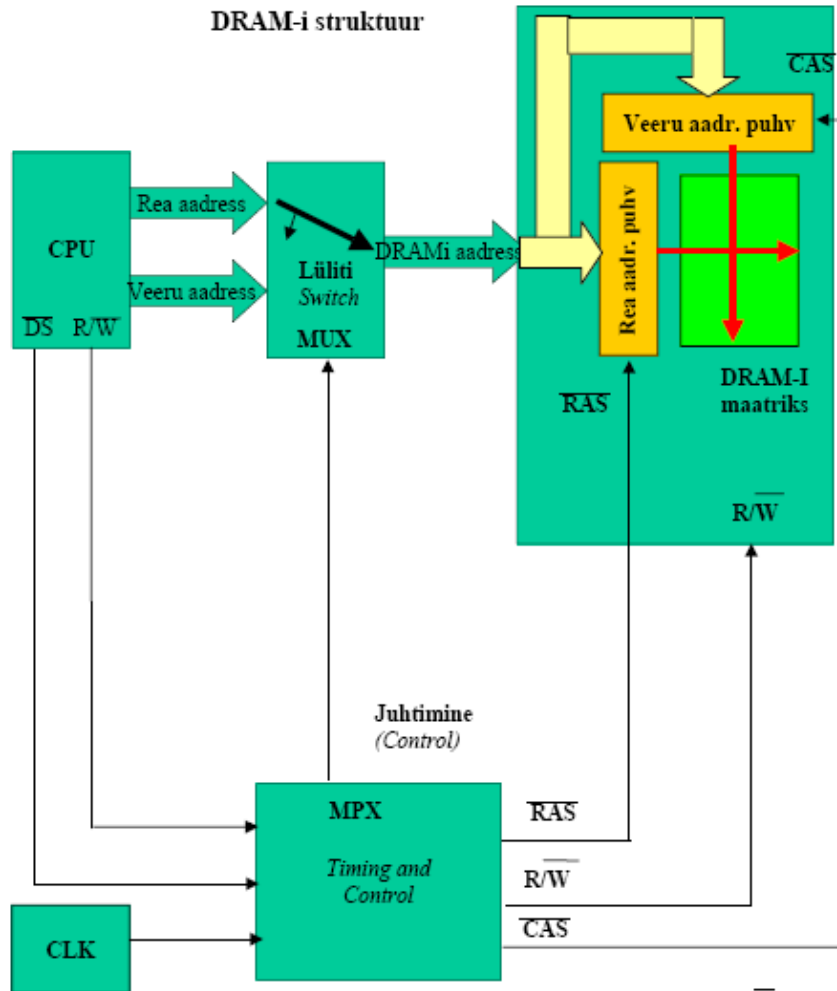
Järgnevatel pildidel on näidatud kuidas toimub nende aadresside sisselugemine samade väljaviikude kaudu. Kui on valitud rida ja veerg siis osutatakse ühele bitile maatriksis. Sõna järgulises saavutatakse sellega, et neid maatrikseid on üksteise peal mitu kihti.



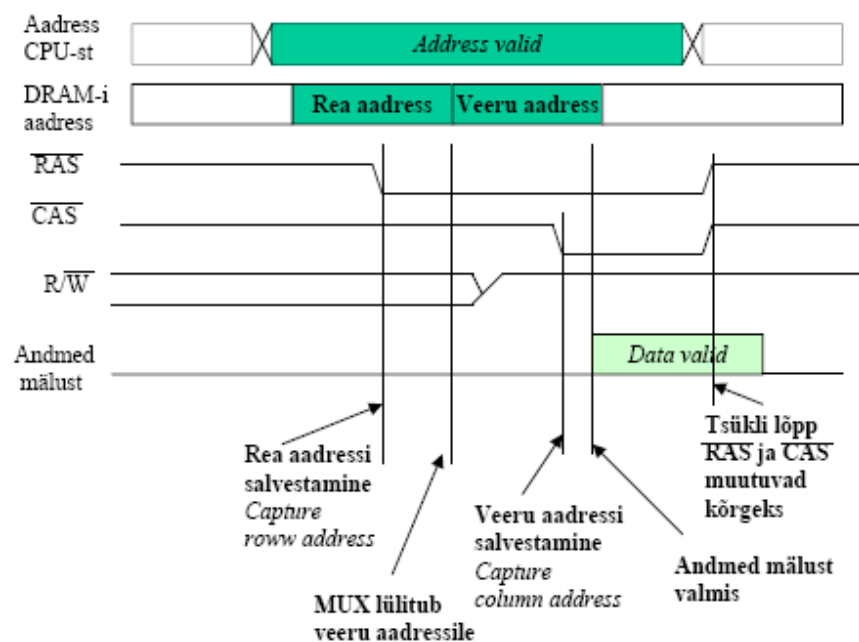
Dünaamilise mälu juhtimine:

Rea ja veeru aadressid loetakse sisse samade väljaviikude kaudu. Ajaliselt toimub lugemine järjestikuliselt, mis muudab mälu poole pöördumise aeglasemaks. DRAM-i mõistete juures on toodud ka mõned võtted, mis pöördumist võimaldavad kiirendada.

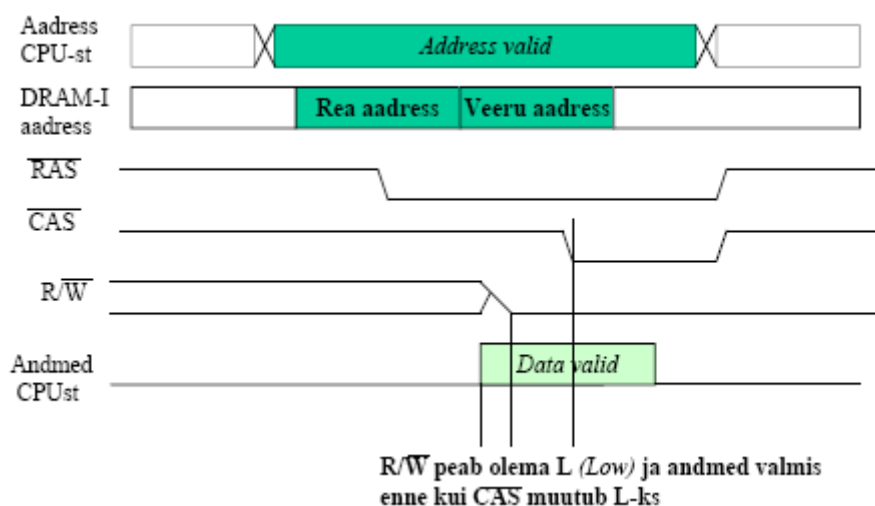




Dünaamilisest mälust lugemise tsükkel (Read cycle of Dynamic RAM):



Dünaamilisse mällu kirjutamise tsükkel (Write cycle of Dynamic RAM):



Mõisteid DRAM-i juurde:

Fast Page Mode (FPM) DRAM – Eeldatakse, et mälus järjestikku paiknevad andmed paiknevad mälumaatriksi aktiveeritud rea järjestikustes veergudes.

Extended Data Output (EDO) DRAM – väljundis puhver, mis lubab alustada uut pöördumist enne eelmise lõppu.

Synchronous DRAM (SDRAM) – ühendatud süsteemi kellaga s.o. sünkroonne, järjestikuste adresside poole pöördumisel genereerib järgnevad aadressid ise (burst) ja sisemiselt jagatud kaheks või neljaks pangaks. See võimaldab ühe panga poole pöördumisel teises infot uuendada (refresh).

Rambus DRAM (RDRAM) – Kiire sünkroonne mälu. Koosneb Multipank DRAM (MDRAM)-idest, millele on lisatud täiendav liidestuslülitus. Edastab infot nii esi- kui tagafrondist.

Content Adressable Memory (CAM) - assotsiatiivmälu.

Double Data Rate DRAM (DDR DRAM) – SDRAM-i edasiarendus. Edastab infot nii esi- kui tagafrondist.

SIMM – Single Inline Memory Module (72 klemmi)

DIMM – Dual Inline Memory Module (168 klemmi)

• Püsimälu (ROM - Read Only Memory)

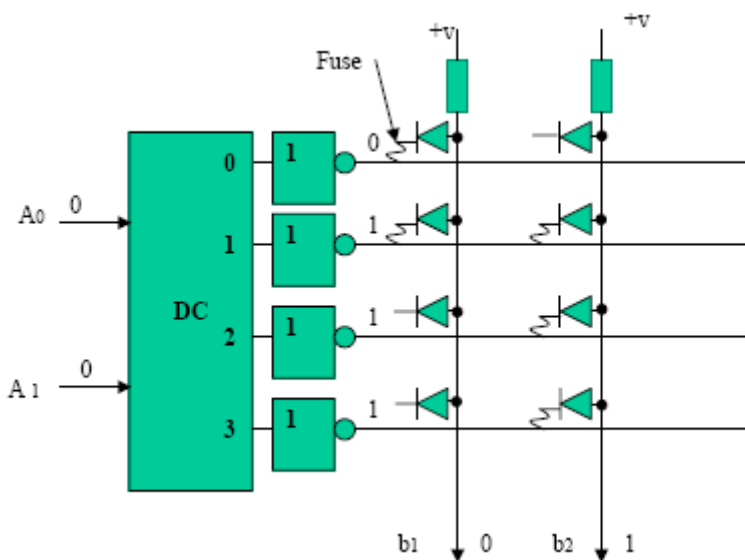
Püsimälu kasut. programmide ning andmete pikaajaliseks säilitamiseks ja lugemiseks. Püsimälud jagunevad ühekordselt programmeeritavateks ja ümberprogrammeeritavateks püsimäludeks. Ühekordselt programmeeritavaid mälusid liigitatakse sõltuvalt sellest, kas need programmeeritakse tehases mälu kiibi valmistaja poolt või programmeerib neid kiibi kasutaja.

Ümberprogrammeeritavaid püsimalusid saab kasutaja vajaduse korral kasutada ja uuesti programmeerida. Muut- ja püsimalude töökiirus peab olema võimalikult suur.

Püsimalude tähtsamad alaliigid:

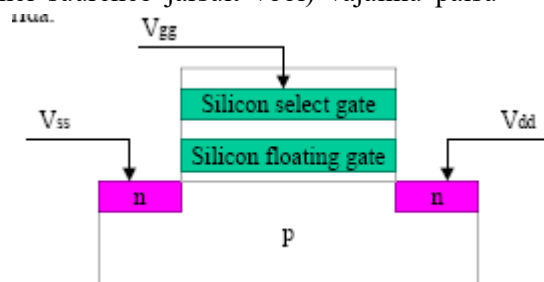
- programmeeritav püsimalu (PROM- programmable read only memory)

Programmeerib kasutaja spetsiaalse programmeerimisega, põletades mittevajalikud väiksema ristlõike pindalaga ühendused.

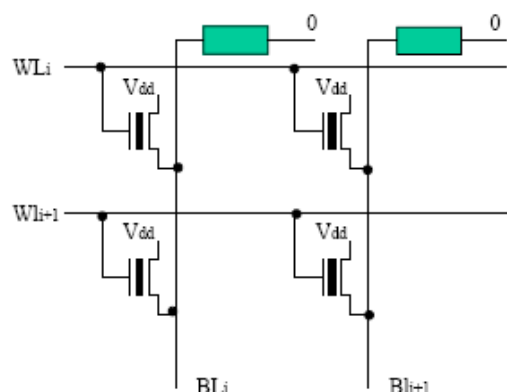


- ümberprog. püsimalu (EPROM- erasable programmable read only memory), elektriliselt kustutatav ümberprogrammeeritav püsimalu (EEPROM- electrically erasable programmable read only memory) ja Flash.

EPROM, EEPROM ja Flash tehnoloogiad põhinevad kõik ujuva paisuga väljatransistoridel. Transistorile on paisu juurde on lisatud teine pais kuhu on võimalik kanda laeng. Selle laenguga muudetakse transistori avamiseks (suudme ja lätte vahel suureneb järsult vool) vajaliku paisu pinge suurust. Põhineb sellel, et kui ujuval paisul on laeng, siis alguses tõstes paisu pinget, kulub ta selle laengu kompenseerimiseks ja seejärel alles avab transistori. Kustutamisel laeng ujuvalt paisult eemaldatakse (EPROM-il ultravioletse valgusega, EEPROM-il ja Flashil elektriväljaga). Seejärel saab teda spetsiaalse programmeerimisega uuesti programmeerida.

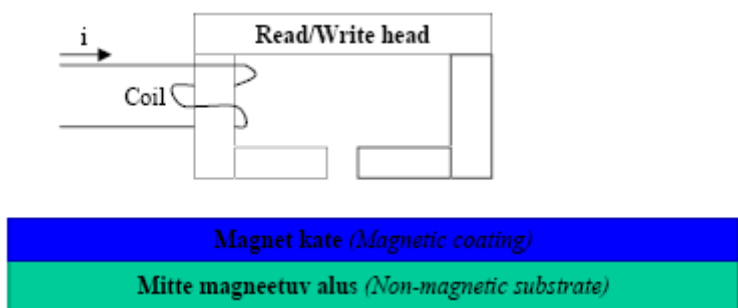


Valides mingi sõna, avanevad ainult need transistorid, mille ujuval paisul laeng puudub ja vastava bitiliini väärtus on üks. Transistorid, kus on

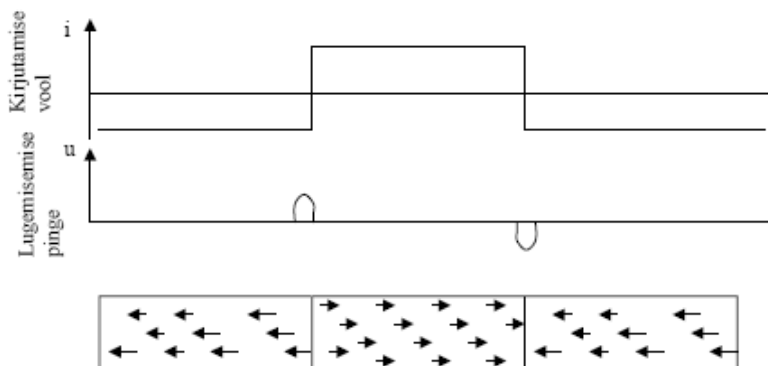


ujuval paisul laeng, ei avane ja vastava bitiliini väärtus on läbi takisti nulli nivool.

- **Magnet mäluseadmed (*Magnetic memory*)**

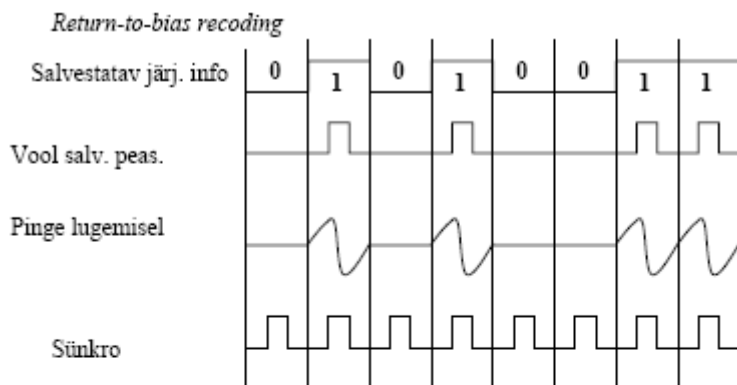


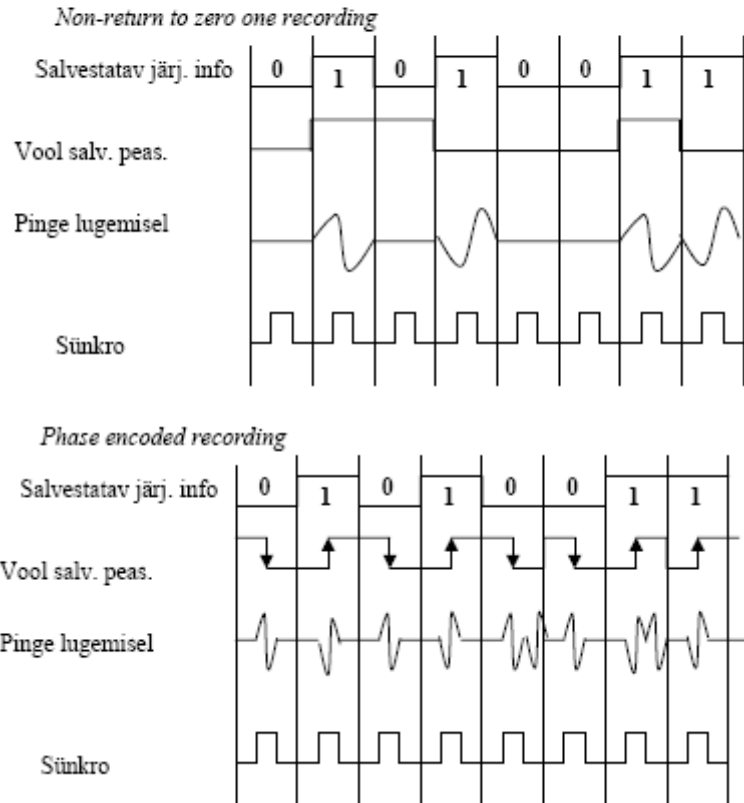
Magnetiline info salvestus põhineb magnet materjali magnetiseerimises ühes või teises suunas. Selleks kasutatakse lugemis/kirjutamis pead, mis on magnetmaterjalist ja mille peal on mähis. Juhtides mähisesse voolu ühes või teises suunas tekib ka vastava suunaline magnetväli. Magnet jõujooned kaarduvad materjalist välja sinna tehtud pilu kohal mis aga omakorda on salvestus materjali lähedal. Lugumisel aga indutseerib magnetvälja muutus mähises impulsse. Vool indutseeritakse selles kohas, kus toimub üleminek magneetamise ühelt suunalt teisele ja voolu suund sõltub sellest milises suunas on magnetvälja üleminek.



Info kodeerimine magnetmälus:

Suurema info tiheduse saavutamiseks info kodeeritakse. Siin on toodud mõned näited. Info salvestuse tiheduse tõstmiseks kasutatakse ka vertikaalset magneetimist, kus eri suunas magnetiseeritud piirkonnad ei ole mitte horisontaalselt pinna suhtes aga vertikaalselt. See võimaldab piirkondade mõõtmeid vähendada.





- Mullmälu (Bubble)
- Pehme ketas (Floppy)
- Kõvaketas (Hard drive)

Kõvakettaga täpsem tutvumine eeldab kirjanduse kasutamist. Siin on toodud vaid mõned mõisted.

Pöörlemiskiirus: Pöörlemiskiirus näitab kui kiiresti kõvaketta plaadid pöörlevad. Tavaliselt kiirused 3600, 4500, 5400, 7200, 10000 RPM.

Ülekande kiirus: *Sisemine ülekande kiirus* (Internal transfer rate) - kui kiiresti suudab lugemispea saata infot kontrolleriile. *Burst ülekandekiirus* (Burst transfer rate) näitab liidese ülekande kiirust. *Pidev ülekande kiirus* (Sustained transfer rate) näitab kui kiiresti liigub info arvuti ja draivide vahel teatud kindala aja jooksul keskmiselt.

Keskmine päringu kiirus (access time) = **Keskmine otsimisaeg** (seek time) + **varjatud otsimisaeg** (latency)

Otsimisaeg (seek time) näitab kaua võtab lugeja peal aega, et jõuda õigele rajale (tavaliselt 10 ja 15 millisekundit).

Varjatud otsimisaeg (latency) näitab, kaua võtab kõvakettal aega, et pöörata plaate nii, et pea jõuaks mööda rada liikudes õige punktini, kuhu info on salvestatud.

Kõvaketta liidesed:

IDE (Integrated Drive Electronics või Intelligent Drive Electronics). Personaalarvutite enimlevinud kõvakettaliides. Paralleelnimetus ATA (AT Attachment, eesti k. AT ühendus). Lubab maksimaalset andmete ülekandekiirust 8,3 MB/s. IDE puhul tekivad probleemid suuremate kui 528 MB ketastega.

EIDE (Enhanced IDE). IDE edasiarendus, mille maksimaalne andmete ülekandekiirus on 16,6 MB/s ning mis lubab CD-ROM-i lugejate ja üle 528 MB mahutavate ketaste kasutamist. Lubab maksimaalselt 4 kettaseadme ühendamist.

Töökindlus:

MTBF - keskmine tõrketa tööaeg (mean time between failures) on kõvaketaste puhul 200,000 ja 500,000 tunni vahel.

○ **Magnet ketas**

Kettad, mis on suurema mahuga kui ümbri kettad (Floppy Disc).

Iomega ZIP drive'i kettale mahub 70 korda enam andmeid, kui 3,5- tollisele disketile (100 Mb). Seejuures on ZIP seadme lugemiskiirus üle kahekümne korra tavalisest floppiseadmest kiirem.

LS-120- (Imation SuperDisk) see on Compaq/Imation- i 120 MB-ne standard. Ühendamiseks kasutatakse EIDE- liidest. Loeb nii vanu 1,44 MB, kui ka uusi 120 MB kettaid, kasutades selleks kahte lugemispead.

HiFD (High Floppy Disk) see on Sony disketiseade, mis suudab lugeda 200 MB- seid 3½" diskette.

Jaz (kõvaketta kantav version)- seadme maksimaalne pidev andmeedastuskiirus on 6,73 MB/s; keskmine otsiaeg 12 ms; pöörlemiskiirus 5400 pööret minutis; ketta vormindamise aeg 30 min; talub kukkumist 3 meetri kõrguselt; andmed säilivad 10 aastat; keskmine tõrketa töövältus (MTBF) 250 000 tundi.

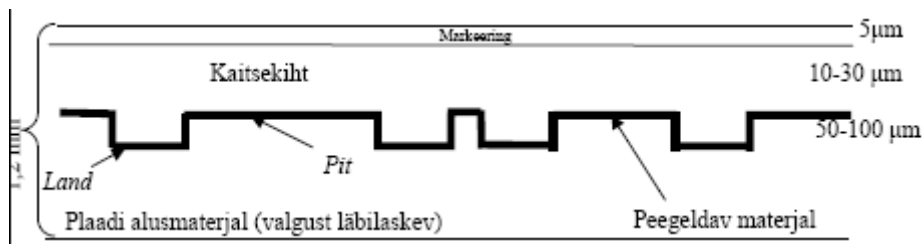
Caleb UHD144 - See uus seade lubab salvestada spetsiaalsele disketile 144 MB, olles samal ajal ühilduv ka vanade 1,44 MB ja 720 KB 3,5" diskettidega.

Samsung Pro-FD - See seade on ühilduv ka vanade 3.5-tolliste (1.44Mb and 720Kb) ketastega ning mahutab spetsiaalketastel 123 megabaiti.

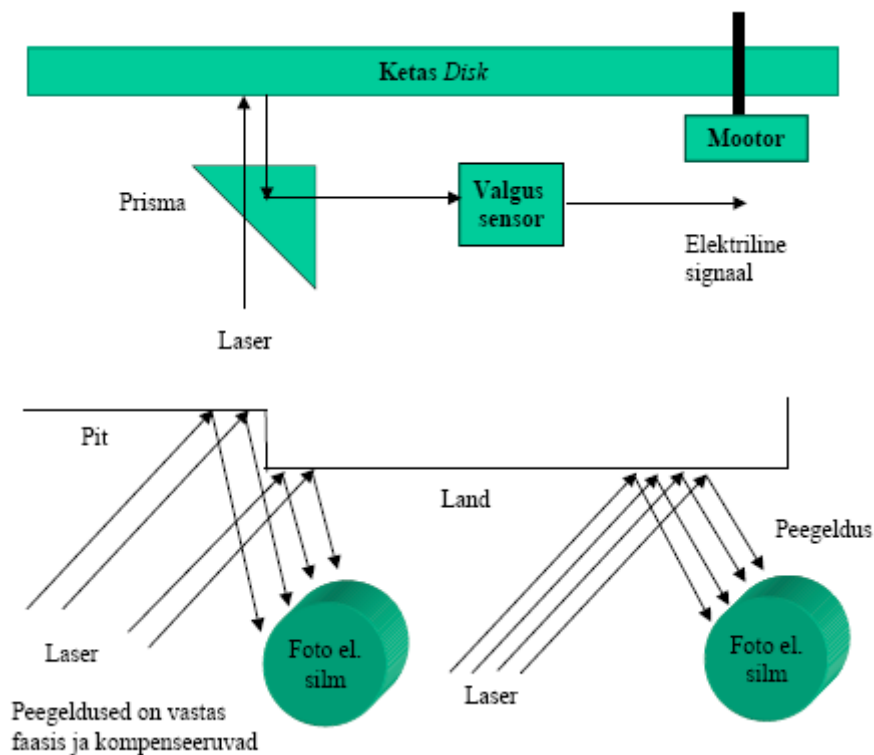
○ **Lint (Tape)**

- **Optilised mäluksadmed (*Optic memory*)**

Optilise info salvestamine:



Info salvestamisel kasutatakse peegelduvat materjali, milles on augud (süvendid). Rada on CD-ROM-l spiraali kujuline (mitte kontsentrilised ringid nagu kõvakettal). Lugeva laseri positsioneerimine on analoogiline kõvaketta peade positsioneerimisega. Peegeldunud laseri kiir teisendatakse elektriliseks signaaliks. Allpool on kirjeldatud info salvestamise füüsikat.



Süvendi sügavus on $\frac{1}{4}$ lainepikkusest. Kui laseri kiirest osa peegeldub ketta pinnalt ja osa süvendist, siis läbib süvendist peegeldunud kiir kaks korda $\frac{1}{4}$ lainepikkuse võrra pikema tee. Seega on need kaks osa kiirest nüüd vastas faasis ja kompenseeruvad teineteist. Seega tuntakse ära mitte süvendid vaid hoopis üleminekud. Salvestamisel kasutatakse spetsiaalset 14 bitist koodi kus ei ole kunagi kõrvuti kahte ühte. Kuivõrd üleminek vastab ühele ei ole neid võimalik ka kõrvuti teha. Koodis on kahe ühe vahel vähemalt kaks nulli. Selleks, et kahe kõrvuti oleva koodid ei oleks lubamatult lähedastikku on iga koodi vahel kolm bitti eraldajat. Laserit kasutatakse valgus allikana sellepärast, et laseri valgus on monokroomne ja kogu allikast lähtuv valgus pean olema samas faasis.

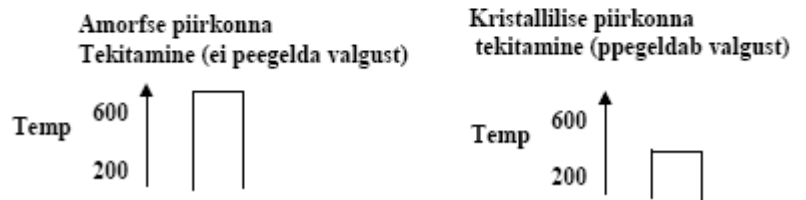
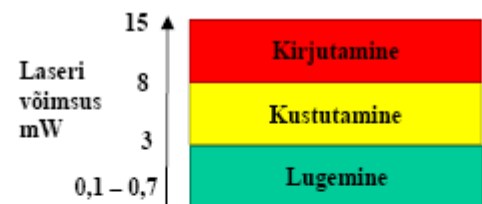
- CD-ROM

- CD – R

Sarnaneb ehituselt CD-ROM-ile, kuid põhimiku ja metallikihi vahel on valgustundlikust orgaanilisest materjalist (tsüaniin või seda sisaldavad segud) andmekiht. Põhimikku on pressitud pidev spiraalvagu, mille järgi kirjutav seade hiljem kirjutuslaserit positsioneerib (kirjutamiseks kasutatakse kõrge intensiivsusega laserkiirt). Kirjutamisel tekitatakse vagudevahelisele alale “lohke”. Need ei ole tegelikult lohud, vaid materjali kerge sulatamisega mittepeegeldavaks muudetud piirkonnad, mida CD- seadme laser peab lohkuks.

- CD – RW

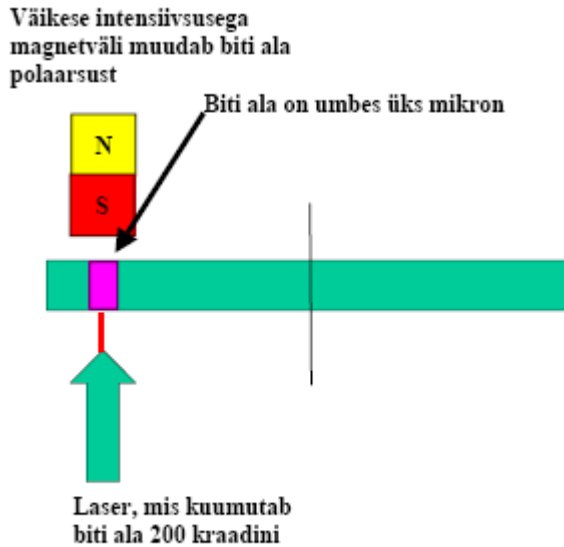
CD-RW andmekihi pind koosneb erilistest keemilistest komponentidest, mis võivad oma olekut korduvalt muuta ja säilitada, sõltuvalt temperatuurist. Materjali kuumutamisel ühe temperatuuriga ja seejärel jahutades, aine kristalliseerub ning teise temperatuuriga kuumutades, võtab aine mittekristalliseerunud oleku. Kui aine on kristalliseerunud, peegeldab ta rohkem valgust kui mittekristalliseerunud, seega saab kristalliseerunud pinda kasutada kui põhipinda "land" ja mittekristalliseerunud kohta lohuna "pit". Seega peab CD-RW seade kasutama korduvkirjutamisel kahte erinevat laserikiire võimsust.



- DVD

- Magnetoptiline (MO)

Magnet-optiline salvestus on vastupidavam põrutustele ja kõrgele temperatuurile. Ketta materjali kuumutamisel saab nõrga magnetväljaga muuta materjali omadusi nii, et ta hakkab sõltuvalt magnetiseerumise suunas peegeldama erineva polaarsusega valgust. Seega kirjutamine toimub magnet välja toimetel ja lugemine optiliste vahenditega.



Lugemisel peegelduva kiire polaarsus kannab infot.

Magnetoptiliste ketaste eelised:

-Andmete säilitamine MO-ketastel on mugav. Kettalt lugemine on praktiliselt sama kiire kui kõvaketta korral, kirjutamine umbes kolm-neli korda aeglasem.

-MO-kettal on hõlbust viia andmeid ühest kohast teise. Kui väiksemate andmehulkade viimiseks ühest arvutist teise kasutatakse tavaliselt disketti, siis suuremahuliste andmete jaoks jäävad disketid väikeseks

-Magnetoptilised kettad on oma olemuselt töökindlamad tavalistest kettaseadmetest. MO-kettad taluvad palju paremini magnetvälju kui tavalised magnetkandjad. Samuti on MO-ketaste lubatud temperatuurivahemik suurem 5-45°C. Andmete säilivusajaks pakub näiteks Fujitsu oma ketaste puhul 30 aastat.

Magnetoptilisi (MO) seadmeid on väga erineva mahutavusega. Toodetakse nii 3,5" kui ka 5 1/4" seadmeid. 3,5" kettaid on erinevate mahutavustega nt. 128 MB, 230 MB, 650 MB. Kõik senised realisatsioonid kirjutavad ketta ühele küljele. Magnetoptilisi kettaid tehakse ka 5,25- tollistena ning need seadmed võimaldavad suuremaid salvestusmahtusid ja reeglina ka suuremaid kiirusi. Sellised seadmed kasutavad juba ka kahepoolset kirjutamist

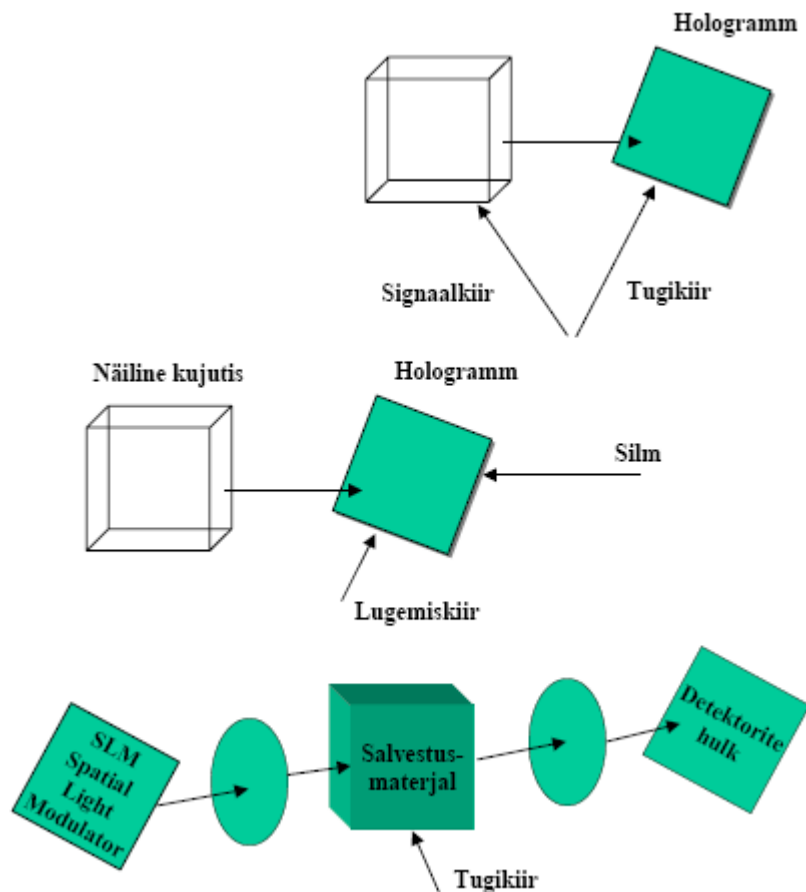
CAV Constant Angular Velocity, konstantse nurkkiirusega kettad. Pöörlemiskiirus on konstantne ja igal rajal on ühesugune arv sektoreid. Seega paiknevad bitid välimisel rajal suhteliselt väikese tihedusega ja palju ruumi läheb kaotsi. Nii töötab enamik magnetkettaid. Antud meetodit kasutavad tavaliselt ka alates 16-kordsetest CD-ROM-id.

CLV Constant Linear Velocity, konstantse joonkiirusega kettad. Siin hoitakse konstantsena parajasti loetava raja joonkiirus. Iga rajavahetuse järel tuleb reguleerida ketta pöörlemiskiirust ning see viib andmeedastuskiiruse alla. Salvestustihedus on kõigil radadel sama ja andmeedastuskiirus konstantne. Seda meetodit kasutatakse laserketaste puhul.

ZCAV Zoned Constant Angular Velocity, konstantse nurkkiirusega tsoonkettad. Need kettad paistavad silma muutuva andmeedastuskiirusega: välimiselt rajalt loevad nad pea kaks korda kiiremini kui sisemiselt. Ketas on jagatud tsoonideks ja üks rada sisaldab igas tsoonis erineva arvu sektoreid. Kuna ketta pöörlemiskiirust hoitakse konstantsena, siis liigub välimise raja salvestis lugemispeast lihtsalt kiiremini mööda, võimaldades kiiremat andmeedastust. See meetod on kasutusel paljude SCSI-magnetketaste ja uuemate magnetoptiliste ketaste juures.

- **Holograafiline**

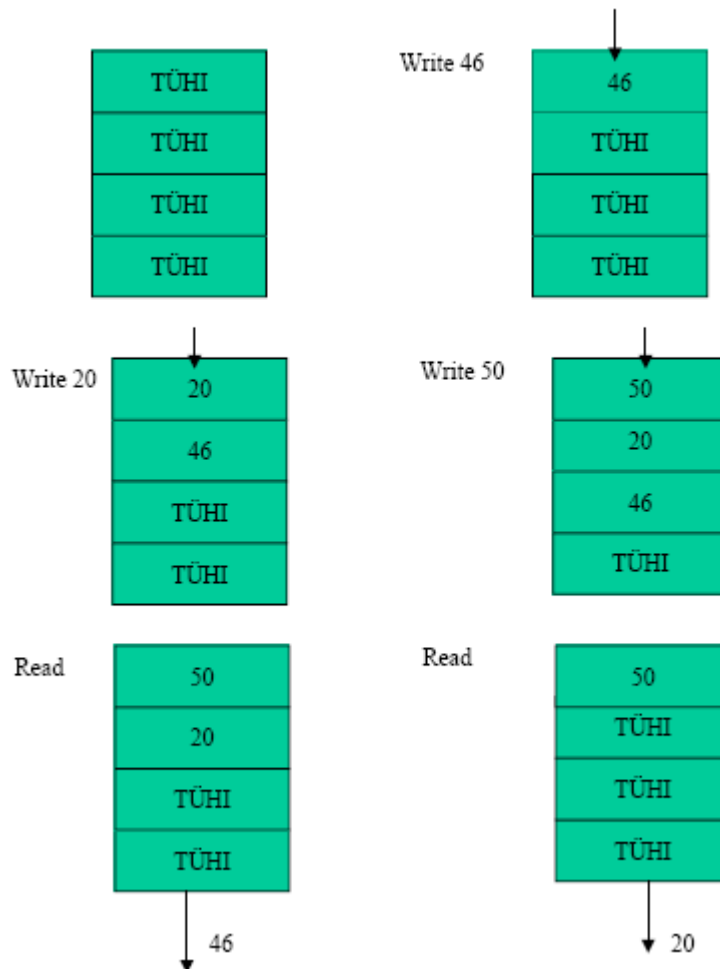
Holograafiline mälu võimaldab väga suurt info tihedust kuid paljude tehniliste probleemide tõttu ei ole veel laialdaselt kasutusel.



- **Erinevate pöördus viisidega mälud (pinumälu (*Stack*, *LIFO*), puhvermälu (*FIFO*)**

- Stack
- LIFO
- FIFO

Puhvermälu tööpõhimõte on First In First Out. Sõna mis kirjutati esimesena mälli lotakse esimesena välja. Kasutatakse näiteks erineva andmeedastus kiirusega seadmete vahel info puhverdamiseks.



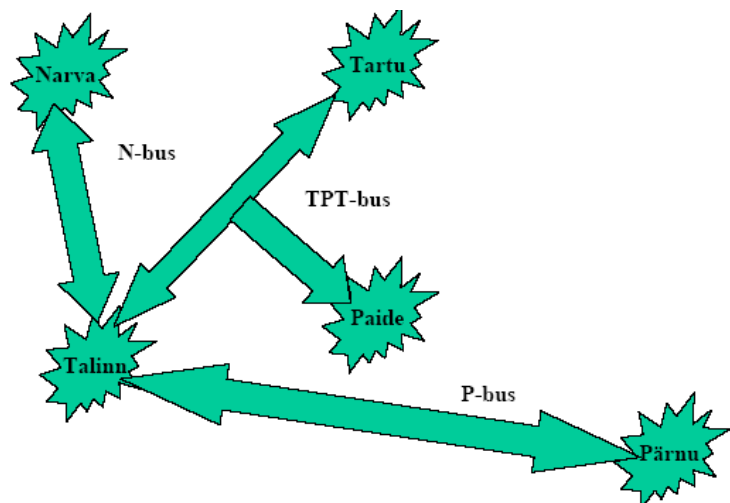
Käsusüsteem ja adresseerimine.

- Käsuformaadid ja käsusüsteem (*Instruction set*)
- Adresseerimise viisid (*Addressing modes*)

Mikroarvuti riistvara

- Mikroarvuti arhitektuur ja siinid.

Mikroprotsessorite ilmumine tõi kaasa uue arhitektuuri tekke. Selleks on siini arhitektuur kus ühedliinid e. siinid (füüsiliselt



komplekt juhtmeid) on kasutusel kõigi süsteemi komponentide vahel info vahetamoseks. Analoogiline kiirteede süsteem liikluses. Kõik süsteemi komponendid peavad täitma teatud formaalseid reegleid (siini protokol), et vältida konflikte. Liikluses on selleks liiklus eeskirjad. Probleem on selles, et tehnoloogiliselt on piiratud mikroskeemi väljaviikude arv ja nüüd peab suhteliselt võimas protsessor suhtlema kõigi teiste süsteemi komponentidega läbi selle piiratud kanali (siini).

- **Erinevad siinid ja nende osa andmevahetuses.**

1. **Andme-siin** Data Bus (DB)

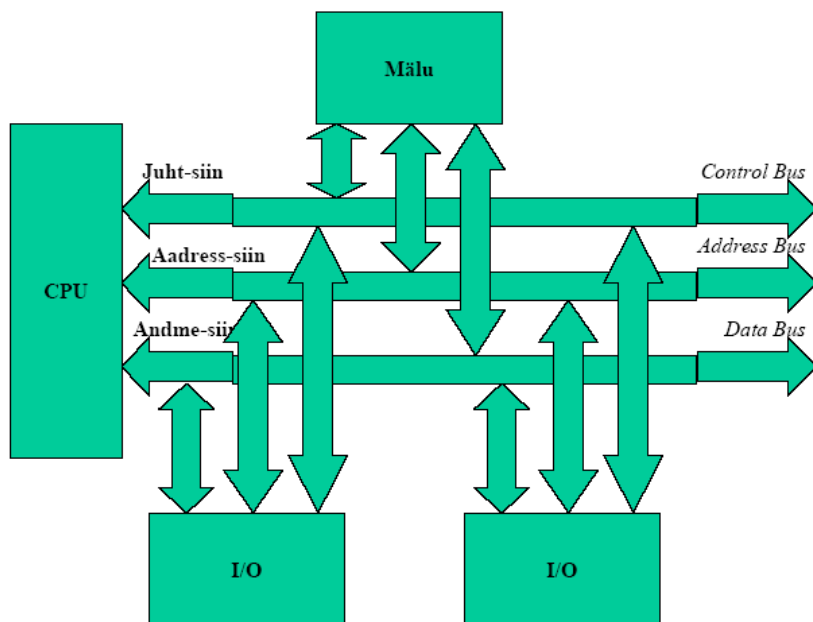
Andmete teisalduseks ette nähtud siin (ühendused kesksprotsessori, mälu ja välisseadmete vahel. Andmesiini laius ja taksagedus määravad ära andmeedastuskiiruse (edastatavate baitide arvu sekundis), mis on üks tähtsamaid arvuti võimsust mõjutavaid tegureid. Enamik kaasaegseid protsessoreid kasutab 32-bitist andmesiini, s.t. ühekorraga saab edastada 32 andmebitti. Mõnel protsessoril on olemas sisemine andmesiin, mis on välimisest siinist laiem, et teha välimised ühendused odavamaks ja samal ajal säilitada laiema siini eeliseid (suurem andmetöötlusvõimsus)

2. **Aadress-siin** Address Bus (AB)

Protsessori ja mälu vaheline siin aadresside edastuseks, kui protsessor tahab mälli kirjutada või sealt lugeda. Aadressisiini bittide arv määrab ära mälu maksimaalse suuruse, mille poole protsessor saab pöörduda.

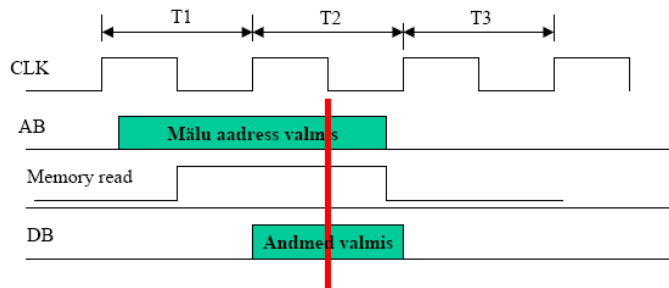
3. **Juht-siin** Control Bus (CB)

Juhtsiini kaudu antakse juhtimissignaale (-impulsse) üksikutele protsessoriüksustele.

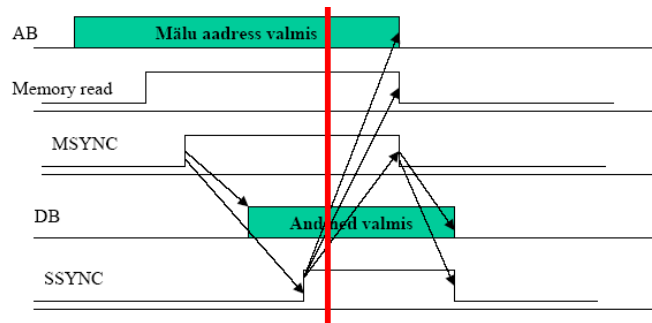


- Andmeedastus protokollid.

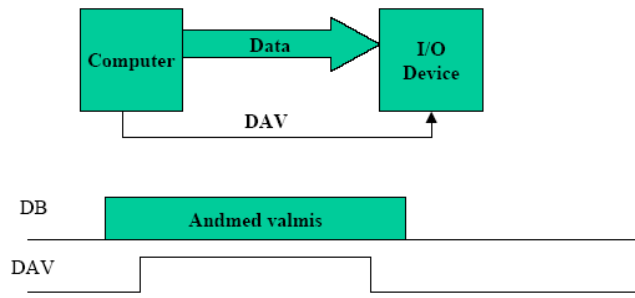
- Sünkroonne siin *Synchronous Bus*



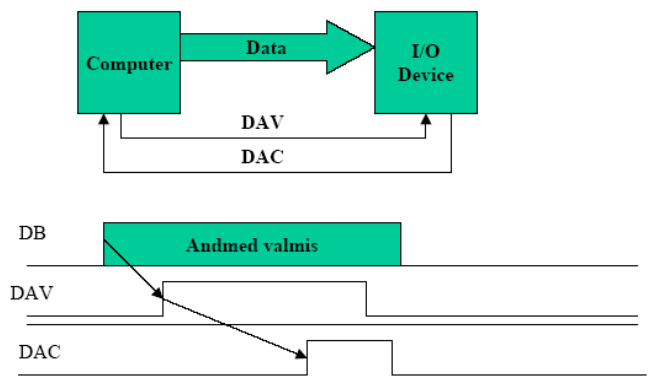
- Asünkroonne siin *Asynchronous Bus*



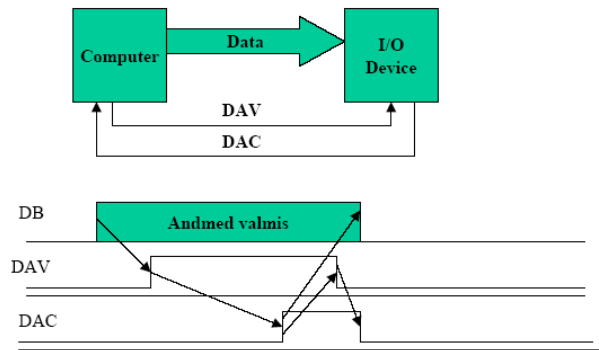
- Tagasisideta anmevahetus *Open-loop data transfer*



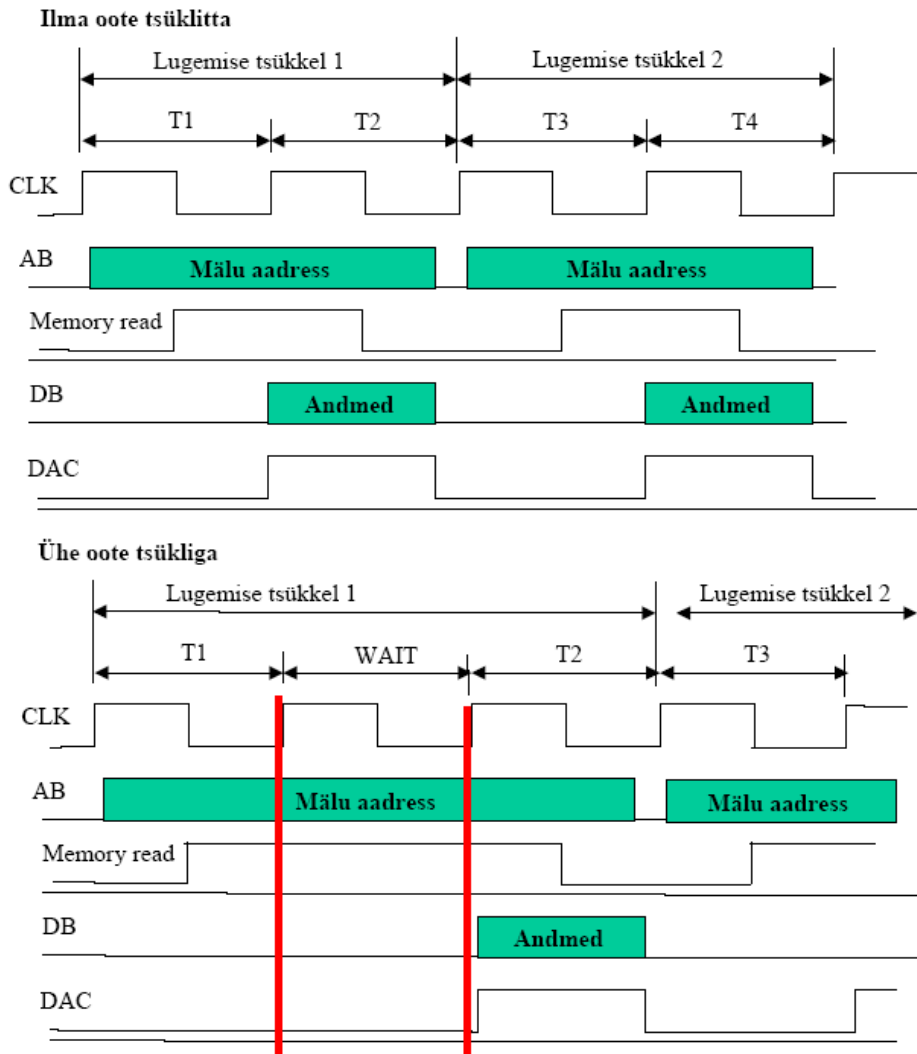
- Andmevahetus tagasisidega *Closed-loop data transfer*



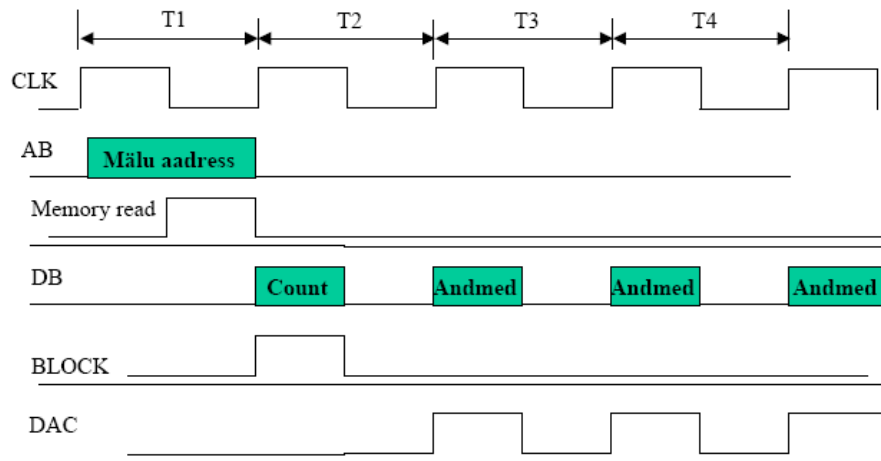
- Andmevahetus täieliku tagasisidega Fully inlocked handshaking



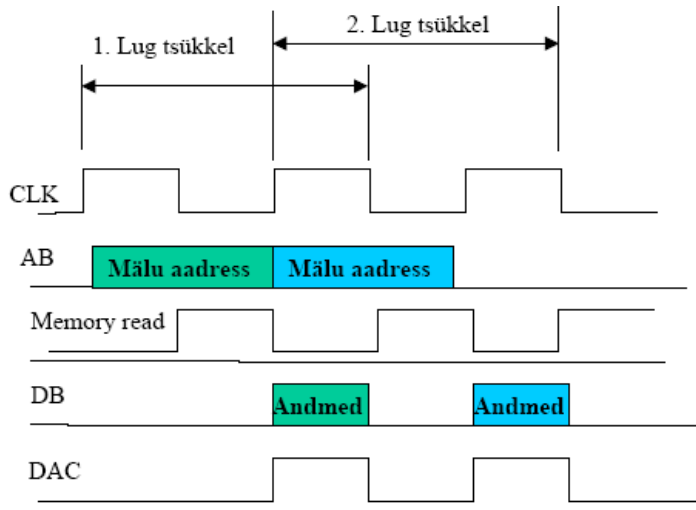
- Andmevahetus oote tsüklite lisamisega Data transfer adding wait states



- Grupiandmeedastus *Burst mode*



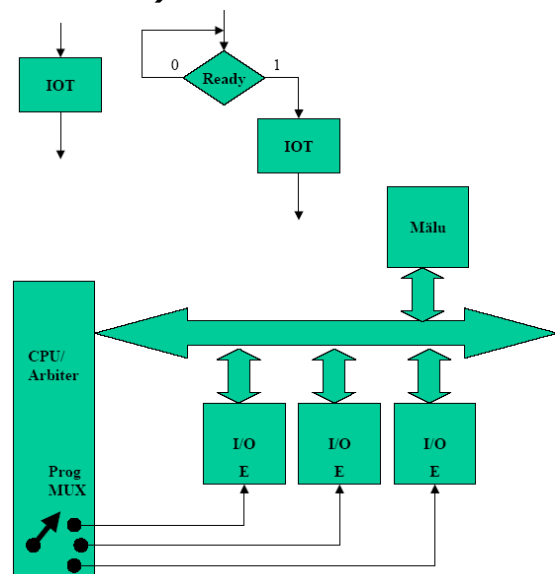
- Andme edastus konveierina *Pipelining*



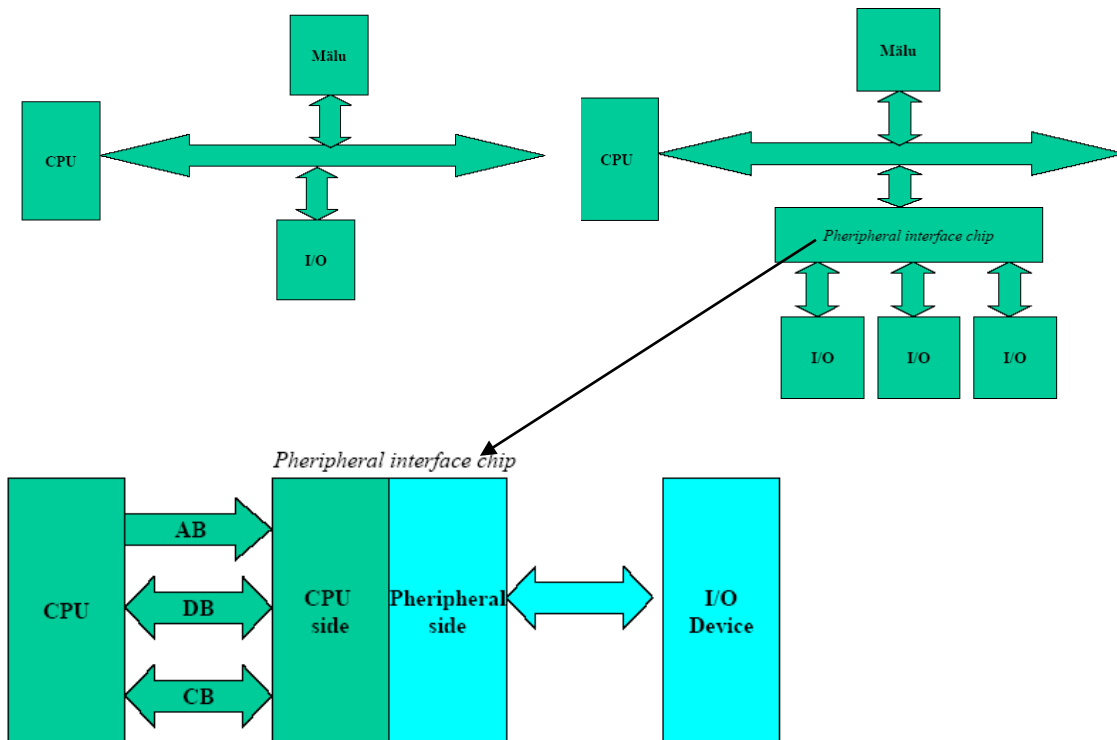
- Andmevahetuse juhtimine (*Bus arbitration*)

Andmevahetus passiivsete I/O seadmetega süsteemis.

-



- **Sisend-väljund seadmete ja protsessori andmevahetus**



- **Mikroprotsessori juurde kuuluvad komponendid (*Supporting System*)**

- Mälu kontrolleri (Memory controller)
- Peidikmälu, vahemälu kontrolleri (Cache controller)
- Siini kontrolleri (Bus controller)
- Mälu otsepöördus režiimi kontrolleri (DMA controller)
- Programmeeritav katkestuste kontrolleri (Programmable interrupt controller)
- Programmeeritav taimer (Programmable interval timer controller)

Sisend-väljund seadmed

- **Klaviatuur (Keyboard)**

Arvutiklaviatuur sarnaneb kirjutusmasina klaviatuurile, kuid sisaldab lisaklahve. Arvutiklaviatuuri klahvid jagunevad järgmistesse **kategooriatesse**:

- 1) **tärklahvid** (tähed, numbrid ja erimärgid)
- 2) **interpunktuatsiooniklahvid** (koma, punkt, semikoolon jne)

- 3) spetsiaalklahvid (funktsiooniklahvid, Control-klahvid, nooleklahvid, Caps Lock klahv jne)

Tärgide standardse paigutusega kirjutusmasinaklaviatuuri nimetatakse QWERTY klaviatuuriks ning see pärineb 19. sajandist. Pole olemas standardset arvutiklaviatuuri, kuigi paljud arvuteid tootvad firmad imiteerivad IBM PC klaviatuuri. Tegelikult **on olemas kolm erinevat IBM PC klaviatuuri**

1. esialgne 83 klahviga IBM PC/XT klaviatuur
2. 84 klahviga IBM PC/AT klaviatuur
3. 101 klahviga täiustatud klaviatuur

Neil kolmel klaviatuuril on mõnevõrra erinev Control-klahvi, Return-klahvi ja Shift-klahvi paigutus.

Lisaks neile klahvidele on IBM PC klaviatuuridel veel järgmised klahvid: PageUp, Page Down, Home, End, Insert, Pause, Num Lock, Scroll Lock, Break, Caps Lock, Print Screen.

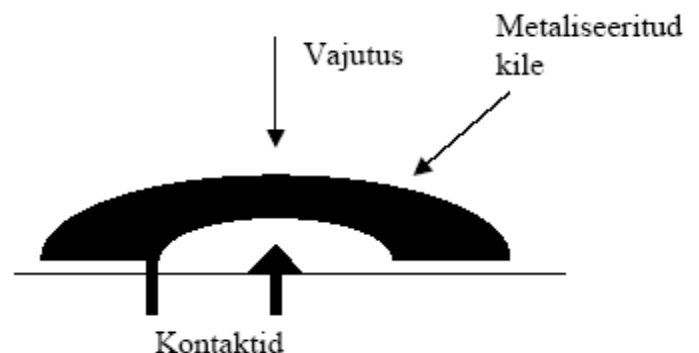
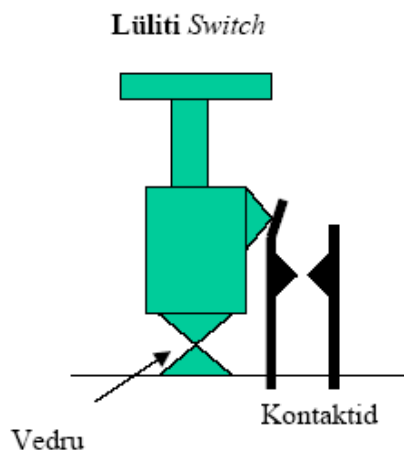
Laialt on levinud veel 104 klahviga nn. Windows'i klaviatuur, millel on täiustatud PC/AT klaviatuuri alumisse ritta lisatud veel kolm spetsiaalset, ainult Windows'i opsüsteemi puhul kasutatavat klahvi (nendele klahvidele ruumi tegemiseks on tühikuklahv tehtud lühemaks).

On olemas mitu erinevat klaviatuuri Apple Macintosh arvutite tarvis. Neid kõiki kutsutakse ADB klaviatuurideks, sest nad ühendatakse Apple Desktop siini (ADB) külge. Kaks enamlevinud Macintosh'i klaviatuuri on standardklaviatuur ja laiendatud (extended) klaviatuur, millel on 15 täiendavat erifunktsioonide klahvi.

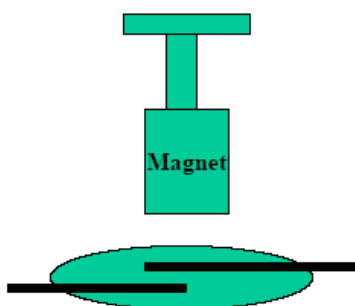
Klaviatuuri ühendamiseks arvutiga on kasutusel 3 erinevat tüüpi ühendusi:

1. 5 jalaga DIN konnektor ehk AT konnektor
1. 6 jalaga mini-DIN konnektor ehk PS/2 konnektor
2. 4 jalaga USB konnektor

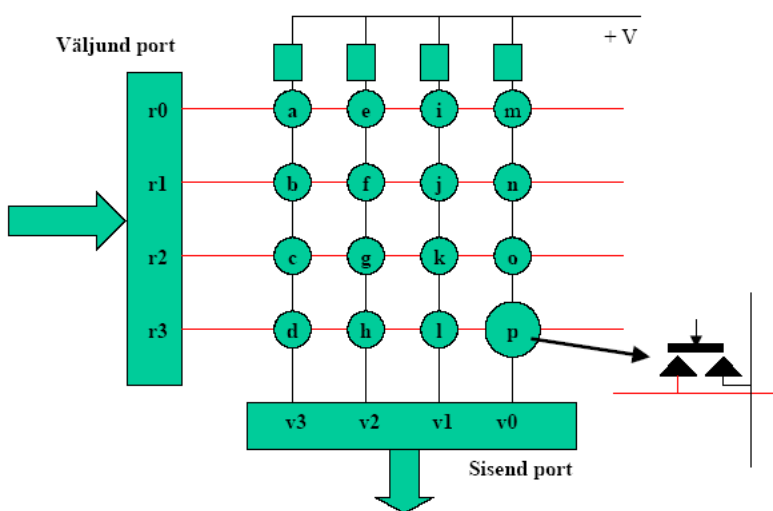
Klaviatuuri kõikide klahvide all on lülitid. Lülitid võivad olla kontaktidega või kontaktivabad. Kontaktide puhul on oluline vältida mustuse sattumist kontaktide vahele. Väga laialt on levinud klaviatuurid kus kasitatakse metaliseeritud plastmassist kilet mis täidab ühtlasi ka vedru ülesannet.



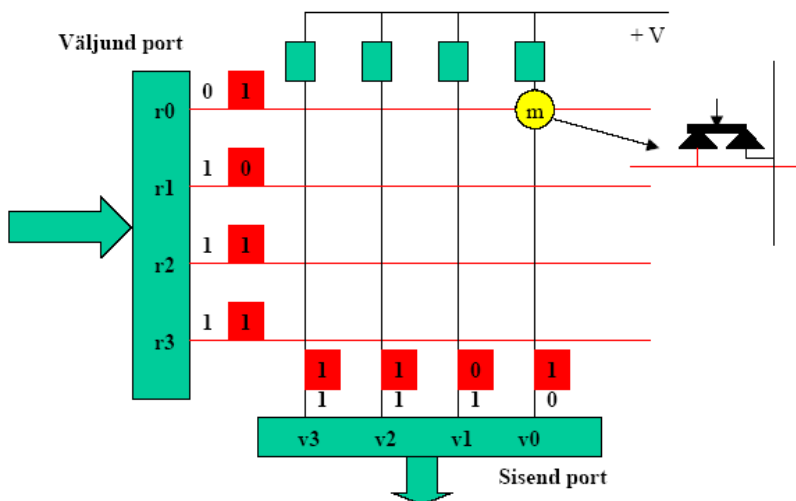
Kontaktid võivad olla ka hermeetilises klaasist baloonis. Kontaktide vahel tekib ühendus kui magneti lähenemisel kontaktid tõmbuvad kokku. Suhteliselt kallis kuid kontaktid püsivad puhtana.



Klaviatuur moodustab maatriksi kus read on ühendatud väljund pordi külge. See tähendab, et klaviatuuri kontroller saab sinna teatud skaneerimise koodi. Veerud on ühendatud läbi kaitsva takisti toite nivooaga (väärtus 1) ja samuti saab lugeda veergude väärtusi läbe sisend pordi klaviatuuri kontrollerisse. Kõigis horisontaalide ja vertikaalide ristumiskohtades on lülitid (klahvid). Iga lüliti küljes on klahv vastava numbri või tähega. Minu näites on 16 täte klaviatuuril.



Scan koodi saamiseks saadetakse vertikaal liinidel kood kus on väärtus 0 ainult esimesel ülemisel vertikaalil. Kõikidel teistel horisontaalidel on väärtuseks 1-d. Kui nüüd on mõni ülemise horisontaali klahv alla vajutatud, siis on ka vastaval vertikaalil väärtus 0. Kõigil neil vertikaalidel kus klahv ei ole alla vajutatud on väärtus üks. Kui alla on vajutatud klahv mõnel teisel horisontaalil, kus on



väljund pordi kaudu väärtus üks, siis ei mõjuta nad vertikaali väärtust. Järgmiseks saadetakse horisontaalidele kood kus on järgmisel liinil null ja kõik ülejäänud ühed. Sellega kontrollitakse kas järgmisel horisontaalil on mõni klahv vajutatud. Sellisel moel kontrollitakse järjest kõiki horisontaale. Kui on jõutud viimasele, siis algab protsess otsast peale. Teades horisontaalile väljastatud koodi ja vertikaalidelt loetud koodi saab kindlaks teha milline klahv on alla vajutatud. See ongi scan kood. Kui korraga on mõnel horisontaalil alla vajutatud rohkem kui üks klahv on ka vertikaalidel rohkem kui ühes järgus null. Valitakse välja neist üks ja vastav kood saadetakse protsessorisse. Kui see oli vale klahv siis selle parandamine on juba klav. kasutaja asi.

- **Hiir ja juhtkang (Mouse and joystick)**

- **Hiir**

Laual libistatav nuppudega osutusseadis, millega saab käivitada programme ja teha mitmesuguseid muid operatsioone, ilma et oleks vaja ära õppida selleks vajalikke arvutikäske

Hiirel võib olla 1 kuni 3 nuppu, mille funktsioonid sõltuvad sellest, mis programm parasjagu töötab. Uuemad hiired on varustatud ka kerimisrattaga.

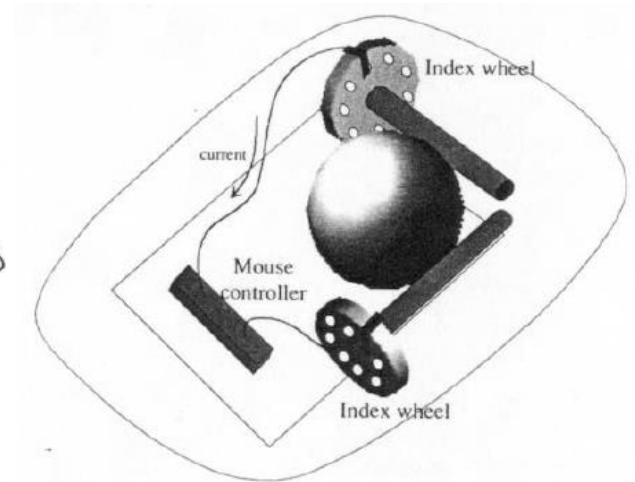
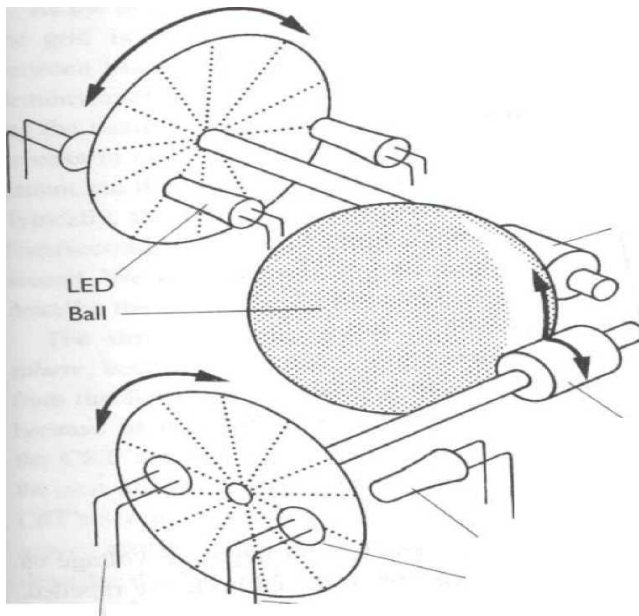
Hiiri on kolme tüüpi:

1. **mehaanilised**, mille põhja all on kummikihiga kaetud metallkuul. Hiire liigutamisel kuul pöörleb ja selle liikumine muudetakse elektrilisteks signaalideks kahe elektromehaanilise sensori abil
2. **optomehaanilised**, milles kuuli liikumist detekteerivad optilised sensorid. Optomehaaniline põhineb kahel (X ja Y koordinaadil) indeks kettal mis on völli kaudu ühendatud hiire all oleva kuuliga. Kui hiirt padjal liigutada pöörlevad indeks kettad. Ketastes on augud. Ühel pool ketast on valguse allikas ja teisel pool fototransistor. Hiire liikumist saab nüüd jälgida selle järgi mitu auku on valgusallika ja fototransistori vahelt läbi liikunud.

3. **optilised**

- a. hiired, milles kuuli ei ole ja hiire liikumist detekteeritakse laseri abil. Selliseid hiiri sai kasutada ainult koos spetsiaalse hiirepadjaga, millele oli kantud tihe joontevõrk. Iga kord, kui valguskiir liikus üle musta või sinise joone, saadeti arvutile elektrimpulss ja kursor nihkus ekraanil veidi edasi
- b. 1999.a. lõpus turule ilmunud hiired, mis sisaldavad tillukest digikaamerat. Need hiired ei vaja üldse hiirepatja, sest hiire põhja alla oleva valgusdioodi valguses teeb digikaamera igas sekundis 1500 18x18 piksli suurust pilti hiire all oleva laua pinnast. Digisignaali protsessor töötleb neid fotosid kiirusega 18 MIPS (miljonit käsku sekundis) ja teeb kindlaks, kui palju hiir on liikunud. See info saadetakse arvutisse ja teisendatakse kursori liikumiseks ekraanil. Sellise hiire eelisteks on:

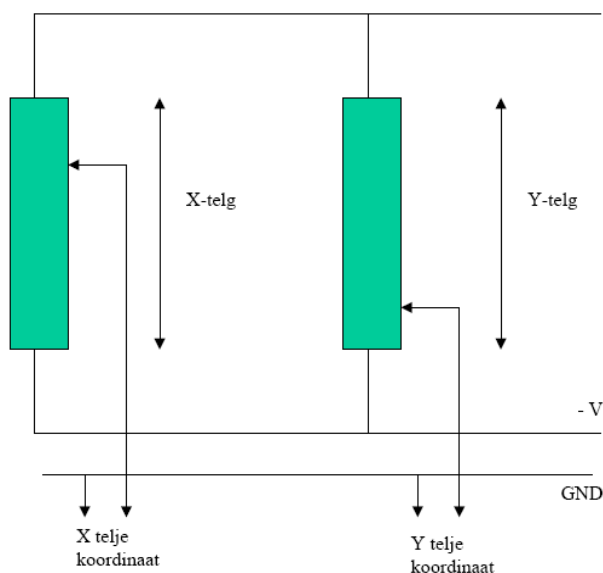
- kursori väga sujuv liikumine ekraanil
- liikuvate detailide puudumine ja sellest tulenev kulumiskindlus
- puudub vajadus hiirepadja järele



Olenevalt tüübist ühendatakse hiir arvutiga RS-232C või PS/2 jadapordi või USB pordi kaudu. Juhtmeta hiir suhtleb arvutiga infrapuna- või raadiosignaalide abil.

Indeks ketastes käsutatakse augu paare, et saaks kindlaks teha hiire liikumise suunda. Kui on kaks valgusallika-vastuvõtja paari mis ei ole täpselt teine teisepool ketta telge tekib ka vastuvõetud signaalide nihe mis määrab liikumise suuna.

○ Juhtkang



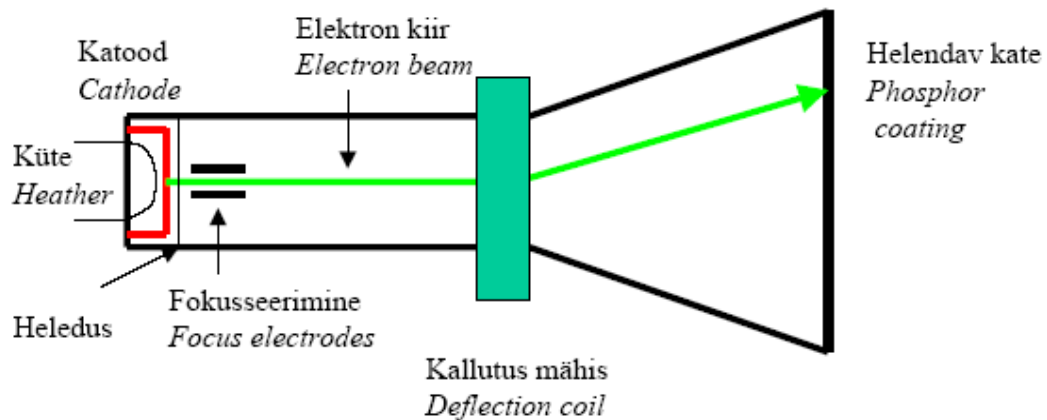
Põhimõte on samasugune kui vanematel mehaanilistel hiirtel. Juhtkang on ühendatud káhe takistiga (X ja Y telg) ja nüüd selle liikuva kontakti pinge sõltub tema asendist takistil väärtustega $-V$ st kuni $+V$ ni. Selle pinge järgi määratakse kontakti asukoht.

- **Kuvar (Display)**

- **CRT (Cathode Ray Tube) kuvar**

Kuvar, mille pilt tekitatakse kineskoobi ekraanile samuti nagu tavalises televiisoris. Kineskoop kujutab endast suurt klaasist vaakumlampi, mille ekraani siseküljele on kantud kolme värvi luminofoorist (punane, roheline ja sinine) koosnevad punktid. Kineskoobi kaelaosas asub elektronkahur, millest väljuv elektronkiir paneb luminofoori helendama. Kallutuspoolide abil pannakse elektronkiir ekraani pinda mööda ridahaaval ülalt alla liikuma ja kui üks kaader on ekraanile joonistatud (kiir on alla välja jõudnud), algab protsess otsast peale.

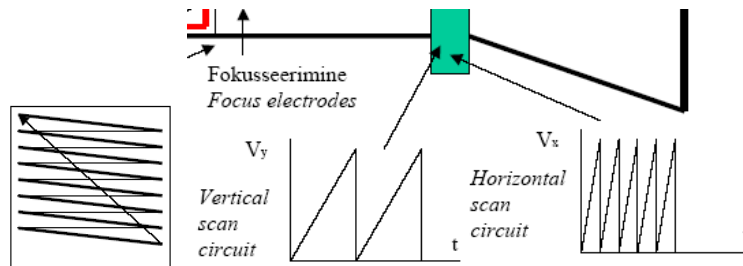
CRT kuvar põhineb elektronkiire torul. Idee on lähedane 1920-ndatest aastatest pärit raadiolampidele. Katoodi kuumutatakse ja sealt tekib elektronide emissioon. Ilma välise mõjuta tekiks varsti tasakaal niipalju kui elektrone lendub niipalju ka maandub uuetsi katoodil. Tekitatakse kõrgepinge (ca 20 000 volti) ekraani ja katoodi vahel mille toimele tekib katoodist väljuv intensiivne elektronide voog. Kõigepealt see voog fookuseeritakse plaatidega millele antakse vastav pinge. Edasi fookuseeritud elektronide kiirt juhitakse kallutus mähise abil vajalikku punkti ekraanil. Ekraan on käetud luminofooriga mis hakkab helenduma elektron kiire toimele. Mida intensiivsem on elektronide voog, seda heledam on luminofoor. Heleduse juhtimiseks on katoodi juures võrk. Andes võrgule negatiivse pinge surub ta osa elektrone katoodi tagasi ja positiivse pinge korral intensiivistab emissiooni.



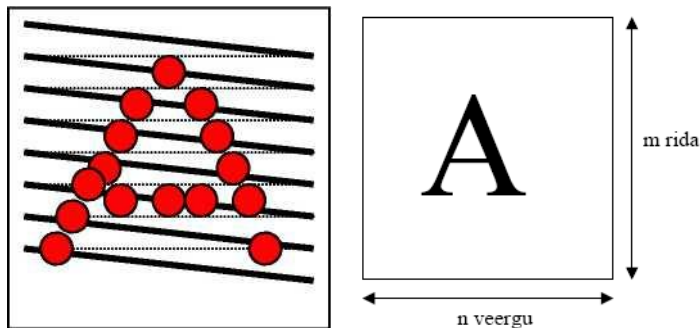
- **kujundi moodustamine**

Kujund koosneb üksikutest punktidest (pixel). Elektron kiir liigub mööda ekraani pannes neis punktides luminofoori elenduma või ei sõltuvalt kiire intensiivsusest. Kiir liigub vasakult paremale joonistades punkte ja seejärel väga kurest tagasi vasakusse serva kust alustab uue rea joonistamist. Kui kogu pilti joonistada üle piisavalt kiiresti tekib silmal tunne, et nagu oleks ekraanil püsiv kujutis. Kui kõik read on joonistatud siirdub kiir kiiresti tagasi vasakusse ülanurka ja kõik algab uuesti.

Kiire juhtimisel kasutatakse kahte hammaspinget. Üks juhib kiirt rea piires vasakult paremale lineaarselt ja kiiresti tagasi. Teine juhib kogu kaadrit ülevalt alla ja siis kiiresti tagasi vasakusse ülanurkka.



Kasutatavste punktide (pixel) arv mõjutab oluliselt pildi kvaliteeti.



Selleks, et kiiresti juhtida mpunktidest kujundi moodustamist kasutatakse videomälu kus igale pixelile vastab monokroomsesl kuvaril üks bit.

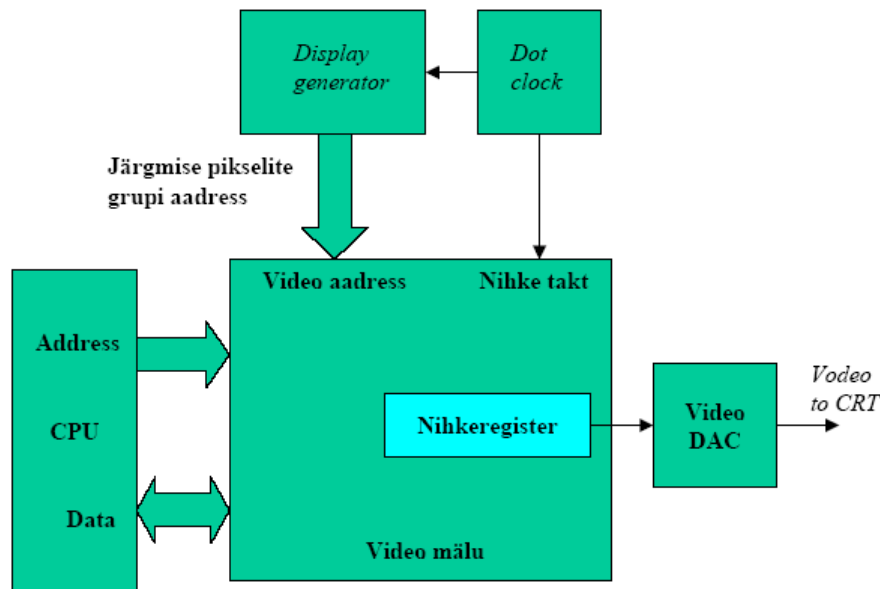
○ videomälu (Video memory)

VRAM - Igasugune muutmälu, mida kasutatakse kujutiseandmete hoidmiseks. Kõik videomälu tüübid on selleks otstarbeks kohandatud dünaamilised muutmälud (DRAM). Videomälu on tegelikult puhver arvuti protsessori ja kuvari vahel ning seda nimetatakse sageli ka kaardipuhvriks. Kui kujutis saadetakse kuvarile, siis loeb protsessor kõigepealt kujutiseandmed välja tavalisest muutmälust (põhimälust) ning kirjutab need siis videomällu. Videomälust (kaardipuhvrst) teisendatakse andmed kuvaadapteri digitaal-analoogmuunduri abil analoogsignaalsiks, mille abil juhitakse kuvarit (näit. elektronkiirt kuvari katoodkiiretorus).

Harilikult on videomälu suuruseks 1 või 2 MB ning see asub videoadapteri plaadil. Enamik videomälusid on kahepordilised, st sellal kui protsessor kirjutab uut kujutist videomällu, loeb kuvar sealt andmeid käesoleva kujutise värskendamiseks. Kahepordiline ehitus ongi peamine erinevus, mis eristab videomälu põhimälust.

Videomällu kirjutab info protsessor. Seda võib ta teha ka videokaardi kaudu. Arvestades küllaltki suurt kiirust millega tuleb teha kujundi ülekirjutamist peab ka videomälu olema kiire. Näiteks super VGA monitoril 1024x768 pixelt ümberkirjutamise sagedusega 70 Hz on

ühe pixeli lugemiseks 18 nS. See eeldab väga kiiret videomälu. Tihti kasutatakse videomälu ja kiire juhtimise vahel nihkeregistrit. See võimaldab lugeda mälust terve sõna korraga (ühe rea pixelite väärtused) ja saata laotusse juba suhteliselt kurest nihkeregistrist. Kiire juhtimine toimub alati analoog signaaliga (kallutus mähised ja kiire heleduse juhtimine). Sellepärast peab teisendama protsessori diskreetse info kuskil analoog kujule. Selleks on digitaal-analoog muundur mis võib asuda vodeokaardil, monitoris või kasutatakse ka video mälu millel on muundur sees (RAMDAC).



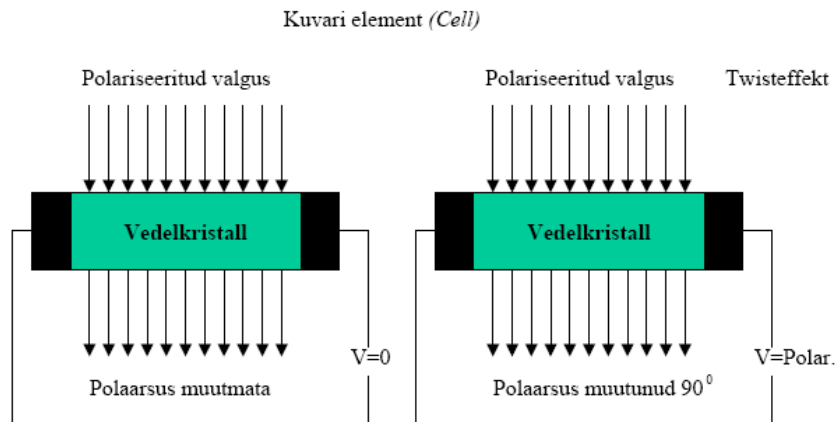
○ vedelkristall kuvar LCD (*Liquid Crystal Display*)

(Nad on kergemad ja vajavad palju vähem toiteenergiat kui tavalised katoodkiiretoruga kuvarid.

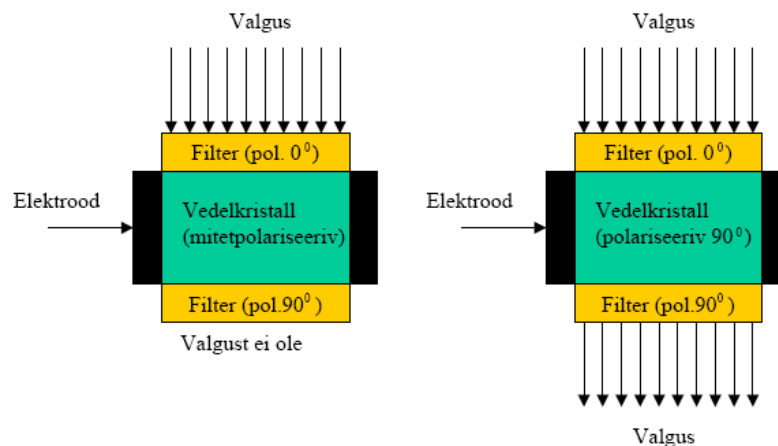
Vedelkristallid on pikad molekulid, mis keerduvad spiraali, kui neile rakendada elektrivälja. Vedelkristallirakkude kihti läbiva valguse polarisatsioonitasand pöörduv vastavalt molekulide spiraali orientatsioonile. Pärast vedelkristallikihi läbimist läbib valgus filtri, mis sõltuvalt polarisatsioonitasandi suunast kas laseb valgust läbi või ei lase. Nii on võimalik elektriväljaga juhtida iga pikseli heledust ja värvilise vedelkristallkuvari puhul ka värvi.)

LCD kuvarid on kahel põhimõttel: nemaatilised ja twisted efektil põhinevad. **Nemmatilistel LCD** kuvaritel muudetakse kristalli struktuuri vooluga mis muudab nende läbipaistvust. Kasutatakse nii tagant valgustamist kui ka tausatavalguse peegeldumist vedelkristalli taga olevalt peeglist. Peegelduse korral jääb valgus tihti nõrgaks ja kujundi kvaliteet ei ole piisav.

Twisted efekti korral muudab vedelkristall teda läbiva valguse polaarsust kui teda mõjutada pingega. Kui kristalli ei mõjutata polariseeriva valgusega läbib valgus muutumatul kujul vedelkristalli.



Kuvare vedelkristall paneeli taga on valgusallikas. Enne paneeli on filter mis laseb läbi valgust 0 kraadise polarisatsiooniga ja paneeli taga on filter mis laseb läbi ainult 90 kraadise polarisatsiooniga valgust. Kui vedelkristalli ei mõjutata polariseeriva pingega ei läbi valgus teist filtrit. Mõjutades vedelkristalli polariseeriva pingega muutub ka valguse polaarsus peale kristalli läbimist ja ta läbib ka teise filtri. Tihti on LCD kuvarite puuduseks aeglus, ebaselge kujund ja vajalik täpne vaatenurk. Tehnoloogia areng on muidugi neid puudusi oluliselt parandanud. Suurimaks energia tarbiaks on paneeli taga olev valgustus.



Passiivne maatriks (Passive matrix)

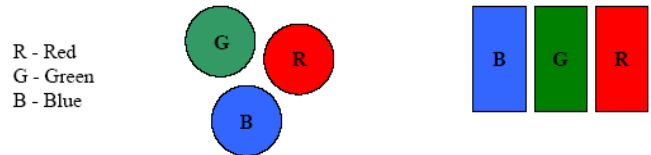
Passiivsel maatriksil toimub ridade ja veergude juhtimine ridade kaupa. Teatud aja möödudes on vaja kujund uuesti joonistada. Probleemiks on naaber pixelite läbikostmine s.t. naabrid mõjutavad üksteist.

Aktiivmaatrikskuvar LCD (active-matrix display)

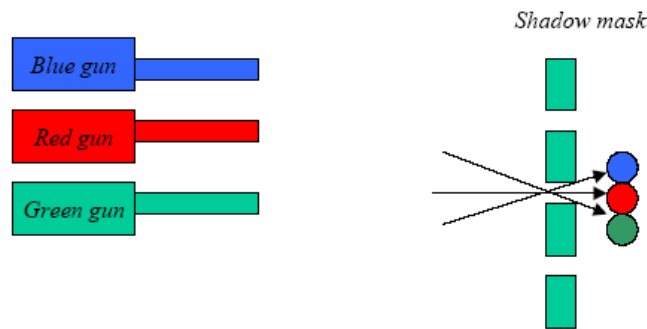
Parima tulemuse saab TFT (Thin Film Transistor) kuvaris (üks LCD alaliik) kus käsutatakse aktiivset maatriksit. Siin on analoogiliselt DRAM-le iga pixeli juures suure mahtuvusega transistor mis teatud ajaks säilitab pixeli oleku. Tegemist on transistoridega mis on realiseeritud LCD maatriksil. Probleem on selles, et neid kilel realiseeritavaid transistore on värvi kuvaril kolm korda pikselite arv. Tehnoloogiliselt tähendab teatud arvu defektsete transistoride olemasolu, et kogu paneel on kõlbmatu. See teeb aga TFT kuvarid suhteliselt kalliks. Pildi kvaliteet on neil väga hea.

- **värviline kujund**

Värviline kujund saadakse kolme värvi: punane, roheline ja sinine liitmisel. Valides summeeritavaid värve erineva intensiivsusega on meil võimalik saada ka erinevaid värve. Selline summeerimine kehtib monitoril kus on aktiivne valguse (värvide) allikas ja taust on must. Printeril on asi näiteks teisiti. Eri värvi punktid asuvad lähestikku ja silm tajub neid ühe punktina. Saadakse need kolme värvi punktid kolme elektron kurega mõjutades kolme eri lisanditega luminofoori punkti



Ekraani taga enne luminofoori on mask (Shadow mask) mis garanteerib, et elektronkiir langeb ainult õigele punktile aga mitte ka naabritele. Samas langeb osa elektron kurest ka maskile ja seega väheneb heledus

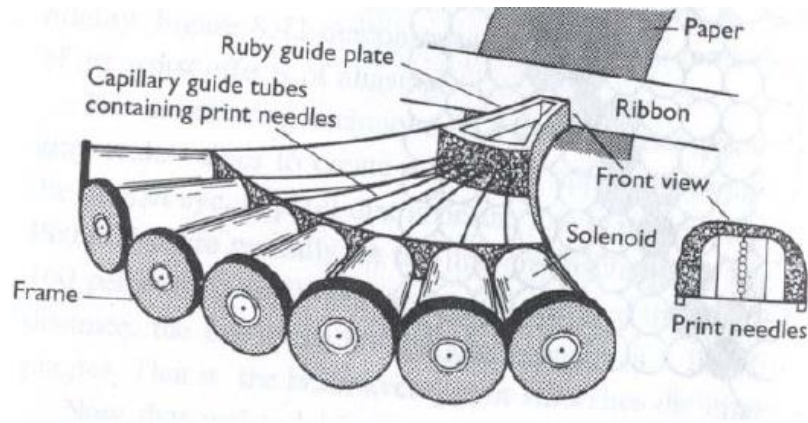


- **Printer (Printer)**

- **maatriksprinter (Dot matrix printer)**

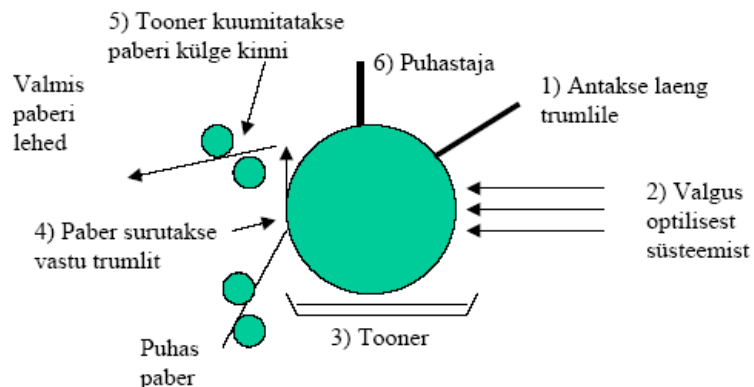
(Nõelprinter, mille printipea sisaldab üht või kaht rida nõelu, millest moodustatakse tähekujundeid ja siis surutakse läbi tindiga immutatud lindi vastu paberit. Printipea prindib rea algusest lõpuni, seejärel järgmise rea jne. Mida rohkem on nõelu, seda suurem on printeri eraldusvõime. Näit. 9 nõelaga saab mustandikvaliteedi ja 24 nõelaga kirjutusmasinakvaliteedi. Printimiskiirused ulatuvad 200 kuni 400 tähemärgini sekundis, mis vastab 90 kuni 180 reale minutis)

Kõigi löökprinterite juures tekitatakse kujund paberile löögiga läbi värvilindi (tindiga immutatud kangas). Erinevus on selles kuidas ja millega teostatakse löök. Kõige levinum om maatriksprinter kus vastu värvilinti mille taga on paber lüüakse trüki peas olevate nõeltega. Peas võib nõelu olla parematel printeritel kuni 24. Kõiki nõelu saab juhtida solenoididega. Kui printeri juhtseade tekitab solenoidis voolu impulsi lööb vastav nõel vastu värvilinti mis tekitab omakorda emda taga olevale paberilepunkti. Selline odav aga lärmakas printer võimaldab trükkida ka lihtsamat punktidest koosnevat graafikat. Varem enimlevinud prontereid kasutatakse veel vaid seal kus on vaja saada prinditust ka kopeerpaberi koopia (näiteks maksekorraldused pankades).



○ laserprinter (Laser Printer)

Laserprinter töö põhineb seleen trumlil. Seleen on pooljuht materjal mis valguse toimet muutub juhiks. Trummel laetakse kõrgepingega (1). Edasi mõjutatakse trumli pinda valgusega (2). Valguse allikaks on laserprinteris laser ja koopiamasinas originaali peegeldus. Need kohad mis saavad rohkemvalgust muutuvad rohkem juhiks ja neilt kaob ka laeng. Nüüd pöörleb trummel edasi ja läheneb toonrile (3). Need kohad mis on rohkem laetud tõmbavad rohkem toonerit külge ja need mis said rohkem valgust ja on vähem laetud vähem. Seega moodustub toonerist trumlile kujund. Koopiamasinal on ta vastavalt peegeldusele aga pronteris koosneb punktidest. Seejärel surutakse trummel vastu puhast paberit (4). Edasi kuumutatakse tooner paberile (5) ja trummel puhastatakse toonerist (6). Seega laser on printeris ainult valguse allikas.



○ Jugaprinter (Inkjet Printer)

tindiprits-printer. Idee meenutab natuke maariksprinerit ainult siin ei lööda trüki peas olevate nõeltega värvilinti vaid peas on pihustid millest pritsitakse paberile värvaine täppe. Pihusteid sisaldav trükipea liigub horisintaalselt paberi läheduses. Vertikaalne liikumine saadakse paberi kerimisega. Neist värvaine täppidest moodustatakse kujund. Suhteliselt lihtne on saada värvilist trükki.

Pihustamiseks on kaks võimalust:

- Piesokristalli pihusti mõjutatakse vooluga mille tulemusena ta muudab oma kuju ja paiskab tindi täpi pihustist välja.
- Trükipeasa on takisti mis voolu impulsi toimel kiiresti kuumenebja paiskab paisunud tinde tilga pihustist paberile.Viimasel meetodil on see hea omadus, et kuumenenud tint kuivab kiiremini.

○ Värviprinterid

Priterites ei ole kasutatav RGB süsteem mis monitoride puhul võimaldas värve liita. Põhjuseks on see, et paber ei ole aktiivne valgusallikas nagu kuvari elektronkiire toru ja taust on valge mitte must. Valge värv teatavast peegeldav kõiki värvusi. Kasutatakse kolme värvi:•CYAN mis peegeldab kõiki värvusi peale punase.•MAGENTA mis peegeldab kõiki värvusi peale rohelist.•YELLOW mis peegeldab kõiki värvusi peale sinist.Kõigi nende kolme värvi summa peaks andma musta, kuid must ei ole eriti kvaliteetne. Arvestades, et silm on musta kvaliteedi suhtes vägatundlik on lisatud eraldi ka must värv –black. Kokku saadaksegi värvisüsteem CMYK mida printerites kasutatakse.

CYAN - neelab punast

Magenta - neelab rohelist

Yellow - neelab sinist

Lisaks must



Laserprinterites saab värvilise trüki mitu korda ületrükkides serinevat värvi tooneritega. Jugaprinteris kas tehakse ületrükk vahetabes tinti või paremates printerites on iga värvi jaoks oma pea.

• **Plotter**

Joonestav kahekoordinaadilise juhtimisega väljundseade. Plotterid erinevad printeritest selle poolest, et nad kasutavad joonestamiseks sulge. Seetõttu suudavad nad tekitada paberile pidevaid jooni, sellal kui printerid moodustavad jooni täpikestest. Värviplotterid kasutavad automaatselt vahetatavaid sulgi värviliste jooniste ja kirjade jaoks.

Plotterid on märksa kallimad kui printerid ja neid kasutatakse enamasti insenertehniliste jooniste valmistamiseks.

Täpsemad ja kallimad kasutavad spetsiaalset sulge ja lihtsamad ning odavamad võivad kasutada ka tavalist pastapliiatsit. Sulg kinnitatakse kelgule ja võib seal liikuda Y suunal. Samal ajal kelk ise liigub alusel X suunal. Mitmevärvilise joonise samiseks kasutatakse eri värvi sulgi. Sulgede vahetamine toimub tarkvaraga juhitud. Paremate plotteritega võib olla joonise täosus 0,25 mm. Tegemist on spetsiifilise valdkonna jaoks projekteeritud seadmega mis ei ole sellepärast leidnud eriti laialdast kasutamist.

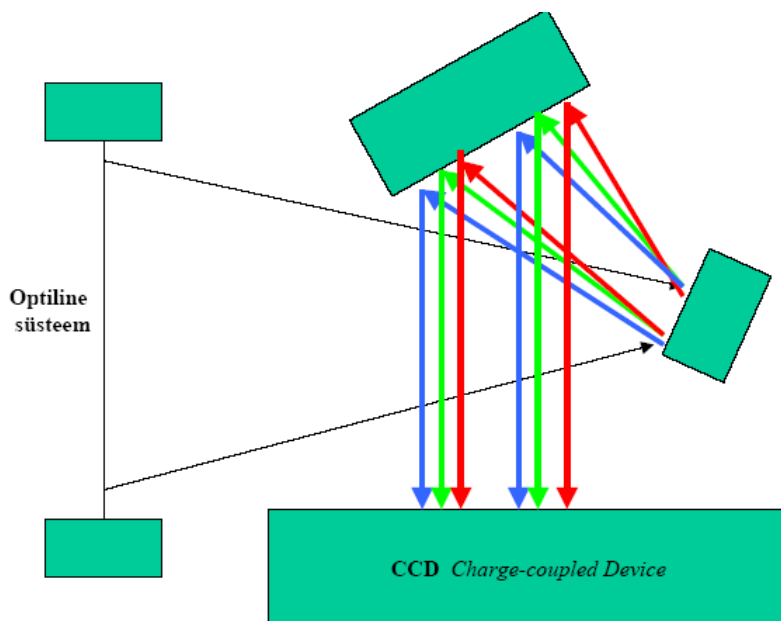
• Skanner

Optiline sisendseade, mis loeb paberilt teksti ja pilte ning teisendab kujutise ridahaaval digitaalsele kujule, nii et seda saab arvutiga töödelda, kuvada ja printida.

Kujutise pind jaotatakse tillukesteks ruudukesteks ja iga ruudukest esitatakse nulli või ühega sõltuvalt sellest, kas ruuduke on tühi või täidetud. Teisiti öeldes, ühe bitiga saab ära näidata, kas ruuduke on must või valge. Halltoonide esitamiseks on vaja iga ruudukese jaoks kasutada mitut bitti. Näiteks 3 bitiga saab esitada 8, 4 bitiga 16, 5 bitiga 32 jne. halltooni. Tavaliselt kasutatakse halltoonide esitamiseks kuni 24 bitti.

Värvidega toimitakse põhimõtteliselt samal viisil. Tabelit, kus igale ruudukesele vastab heledus- ja värvinformatsioon, nimetatakse bittrastriks.

Oluliseks osaks on laengusidestusseadis(*Charge-coupled Device, CCD*) milles tekib valgusetoimel laeng. Mida intensiivsem valgus seda tugevam on laeng. Kui nüüd võtta skaneeritava objekti üks rida ja juhtida tema peegeldus CCD peale saame selle rea kujundi laengutena. Nüüd edasi nidutame selle laengu välja CCD-st ja muudame AD muunduris digitaalseks koodiks.



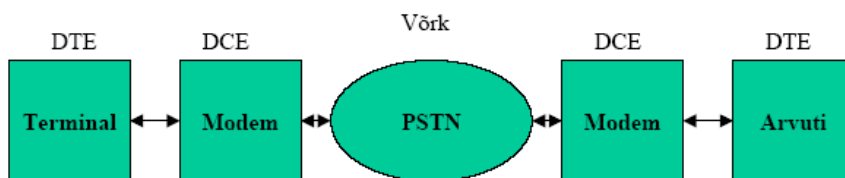
Probleem on selles et CCD ei ole värvitundlik. Küllaltki kallis osa skannerist on värvi lahutaja mis eraldab kolm värvi kkomponenti –RGB. Nüüd juhitakse nad optilise süsteemi kaudu eraldi kolmele CCD-le ja saadaksegi digitaalsel kujul värviline kujutis arvuti mälus.

- **Modem (Modem)**

Lühend sõnadest MODulaator-DEModulaator.

Modem on seade digitaalse info edastamiseks läbi üldkasutatava telefonivõrgu. Modem moduleerib arvutist või mõnest muust digitaalseadmest väljuva digitaalsignaali analoogsignaalsiks ja saadab selle telefonivõrku ning demoduleerib telefoniliinist vastu võetud analoogsignaali digitaalseks, nii et seda saab arvutiga töödelda.

Esimeste modemite kiirus oli 2,4 Kbps ja need võimaldasid edastada ainult e-posti. Vahepeal olid kasutusel 14,4 ja 28,8 Kbps modemid, alates 1988.a. varustati kõik personaalarvutid 56 Kbps modemitega. Võrdluseks olgu öeldud, et ISDN võimaldab samu liine kasutades andmekiirust 128 Kbps ja DSL'i kiirus ulatub megabittideni sekundis.



DTE - Data Terminal Equipment

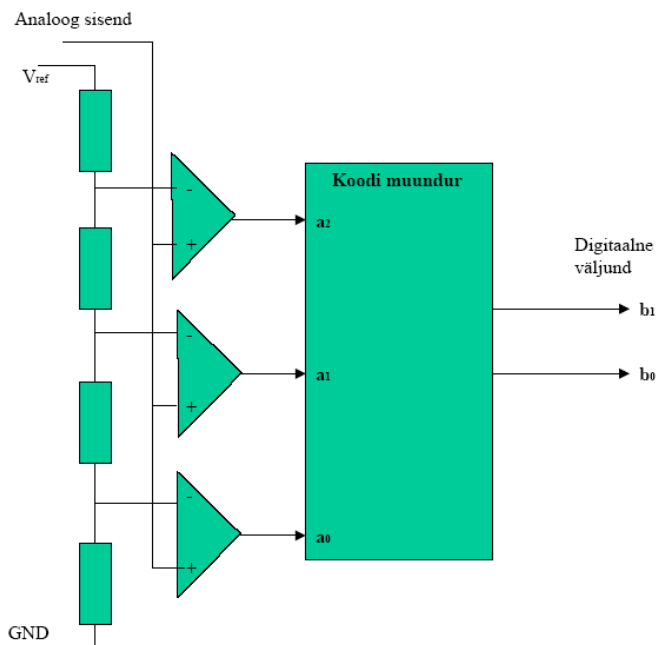
DCE - Data Communication Equipment

PSTN - Public Switched Telephone Network

- **Analoog liides (Analog Interface)**

- **analoog-digitaal muundur (Analog to Digital Conversion)**

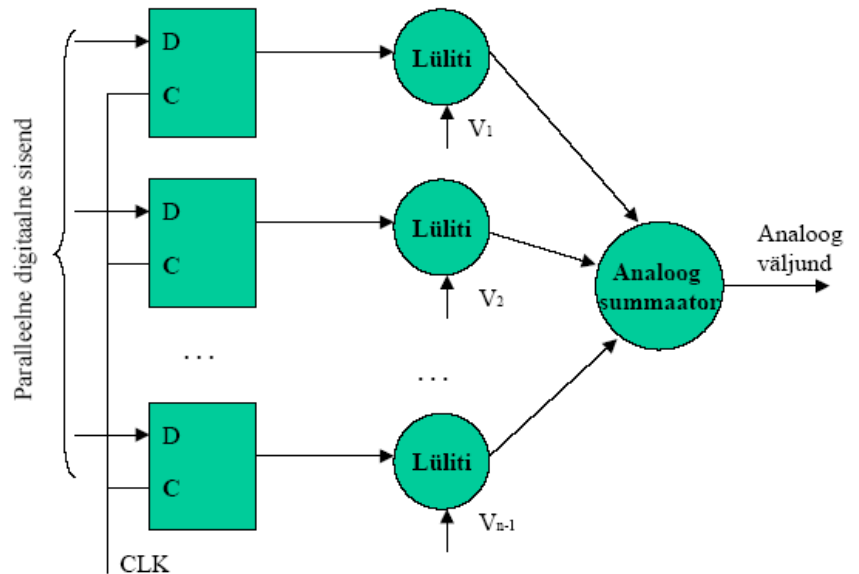
Analood-digitaal muunduril (ADC) on kaks siendit: muundatav analoog sisend ja konstantse fikseeritud pingega sisend (V_{ref}). Edasi tuleb analoog võrdlusskeem mille väljundisse ilmub loogilisele 1-le vastav pinge kohe kui + märgitud pinge on kõrgem kui – märgitud sisendi pinge. Koodimuundur peab teisendama muundamisel saadus koodi kahendkoodiks.



- **digital-analoog muundur (Digital to Analog Conversion)**

Digitaal analoog muundureid (DAC) võib realiseerida mitmel erineval viisil. Näitena olen välja valinud ühe meetodi mis põhineb pingete summeerimisel. Vasakul on register kus hoitakse muundatavat kahend koodi. Põhiliseks komponentideks on digitaalselt juhitud lülitid. Kui vastavas registri järgus on 1-ks, siis lüliti kaudu läheb vastav pinge analoog summaatori sisendisse.

Mida rohkem onkoodis ühtesid, seda suurem arv pingeid läheb analoog summaatori sisendisse. Omaette probleem on nende summeeritavate osapingete valik, et saada võimalikult täpne analoog pinge.



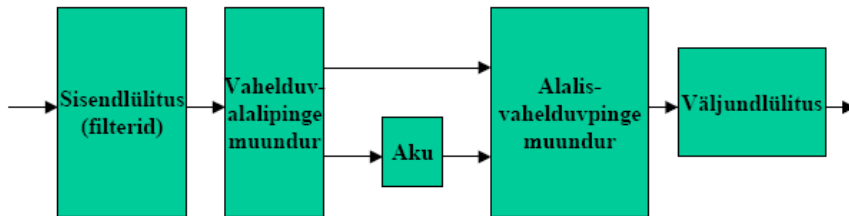
- **UPS- Uninterruptible power supply**

puhvertoiteallikas, ups. UPS on seade mis peab kaitsma arvuti riistvara ja tarkvara võimalike elektrivõrgu häirete mõju eest. Elektrivõrgu häired võivad rikkuda riistvara või katkestada programmi töö ebasobival hetkel ja rikkuda töödeldavat infot.

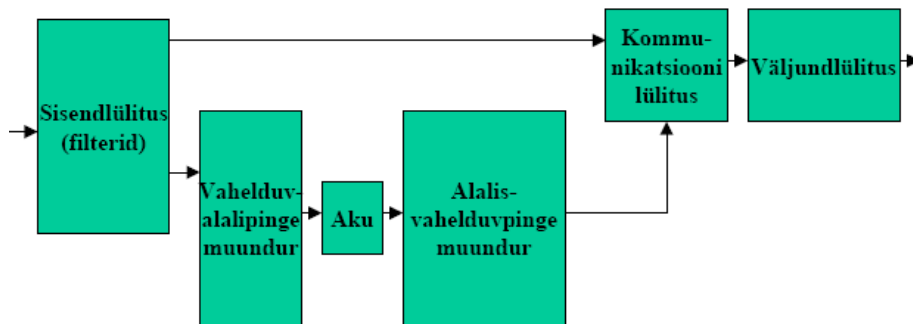
Häired:

1. **täielik elektrikatkestus** blackout
2. **ülepinge** surge, over-voltage
3. **pingelangus** under-voltage, sag, brownout
4. **impulsshäire** spike, transient, impulse
5. **mürad** noise

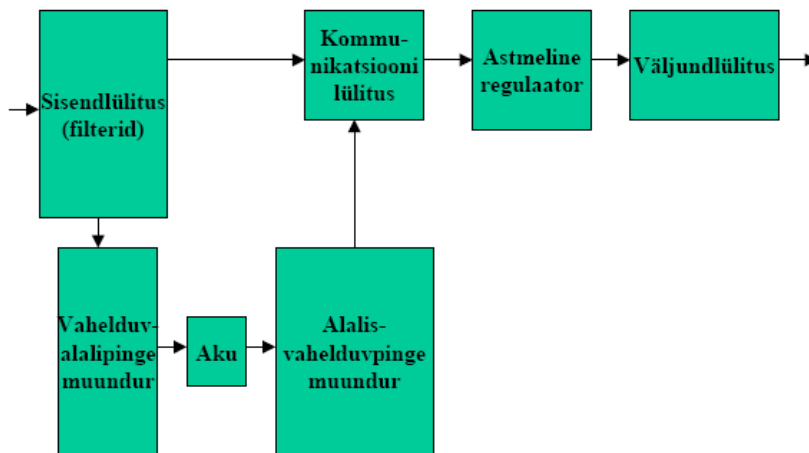
Sidus-UPS koosneb järgmistest osadest: •Sisendfiltrid •Vahelduv-alalispinge muundur (alaldi) mis toimib ka aku laadijana •Alalis-vahelduvpinge muundur mis saab normaalolukorras sisendpinge alaldist aga voolu katkestuse korral tuleb toide akust •Väljundlülitus mis muudab muundurist tulevat vahelduvpinget tavalise siinuselise pinge sarnaseks (vähemalt püüab muuta täisnurkseid impulsse siinuseliseks) Siin toimub kahekordne muundamine vahelduv voolust alalis vooluks ja tagasi. See tagab suhteliselt hästi müradest filtreeritud toite ja ka sisend pinge suurematel kõikumistel viiteta ülemineku aku toitele siludes ka kõikumisi.



Vallas-UPS Antud juhul kajastuvad võrgupinge kõikumised ka otseselt UPS-i väljundis. Puudub sisendpinge galvaaniline lahtisisdestus väljundist. Osa võrgu müradest ja häiretest filtreeritakse sisendlülitustes. Kui võrgu pinge langeb alla kriitilise väärtuse lähedusse UPS üle akutoitele (ümberlülitus on piisavalt kiire, et arvuti töö ei katkeks) ja arvuti töö jätkub kuni akud seda võimaldavad. Akud võivad võimaldada ainult kiiret programmeerimise lõpetamist kuid kallimad UPS-d lubavad ka teatud aja jooksul töötada edasi. Arvestades akude suurt hinda on viimased UPS-d muidugi oluliselt kallimad.



Aktiivne vallas-UPS Võimaldab astmelist pingeregulatsiooni. Kui sidus UPS võimaldab elektroonika abil pidevalt hoida pinget väärtuses paigas ja vallas UPS võid teatud kriitilisest väärtusest väiksema sisendpinge korral minna üle aku toitele, siis siin on võimalik stabiilne pingeregulatsioon.

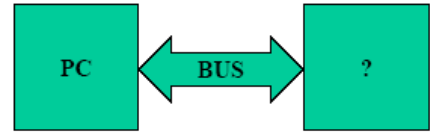


Spetsiaalse riistvara

- **Spetsiaalse riistvara realiseerimise võimalused.**

- **Programne realisatsioon**

1) See tähendab, et me ühendame näiteks paralleelpordi külge oma personaalarvutil juhitava seadme ning kirjutame programmi juhtalgoritmi täitmiseks. Realisatsioon on programme selles mõttes, et juhtalgoritm on realiseeritud arvuti mälus säilitatava programmina mida protsessoris käskhaaval täidetakse.



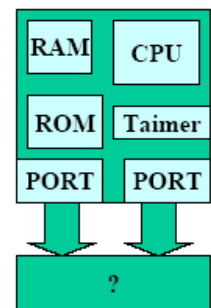
Head omadused:

- Saab kasutada harjumuspärast tarkvara (näiteks Windows keskkonda)
- Suhteliselt lihtne teha muudatusi
- Kui see on probleem, siis suhteliselt vähe on vaja tunda riistvara

Puudused:

- Aeglane võrreldes riistvaralise realisatsiooniga sest programmi täimisel toimub ju pidevalt käskude lugemine mälust protsessorisse ja nende täitmine seal (võib paljudes kohtades olla probleem)
- PC või mõni teine universaalne arvuti on paljudes kohtades mõttetult kallis. Juhtides lihtsa algoritmi järgi tööpinki ei kasuta me võimsa arvuti resurssidest väikestki osa.
- Füüsilised mõõtmised ei ole alati vastuvõetavad.

2) Eelmise versiooni mõned puudused lahendatakse mikrokontrolleri abil. Mikrokontroller kujutab endast ühel kristallil realiseeritud arvutit. Seal on olemas protsessor, taimer, liidesed, mälu, katkestuste süsteem jne. Tõsi mälu maht on piiratud ja ka muud parameetrid ei ole PC-ga võrreldavad, kuid lihtsamaid programme on ta võimeline täitma. Omadused võrreldes eelmise samuti programme realiseerimisega.



Head omadused:

- Kasutatda tuleb oma spetsiaalset tarkvara programmeerimisel
- Samuti suhteliselt lihtne teha muudatusi
- Eeldab suuremat riistvara tundmist, vähemalt riistvara lähedast programmeerist

Puudused:

- Aeglane võrreldes riistvaralise realisatsiooniga sest programmi täimisel toimub ju pidevalt käskude lugemine mälust protsessorisse ja nende täitmine seal (võib paljudes kohtades olla probleem)
- Võrreldes PC-ga suhteliselt odav, aga ka mälu ja muud resursid võivad osutuda paljudes kohtades ebapiisavateks
- Füüsilised mõõtmised on oluliselt väiksemad kui PC-l, kuid mõneski kohas kasutamiseks liiga suured (näiteks mobiiltelefon)

○ Riistvaraline realisatsioon

Alati võib algoritmi realiseerida riistvaras nagu juhtautomaadi protsessoris. See tähendab, et algoritm realiseeritakse loogikaskeemina. Edasi loogikaskeemi realiseerimine võib toimuda trükkplaadinakomponentidest(mikroskeemidest)koostatud loogikaskeemigavõi kristalli pinnal ühe mikroskeemina (*ASIC –Application Specific Integrated Circuit*). Erinevus on siin vaid tehnoloogilist laadi. ASIC-u valmistamine eeldab terve rea etappide läbimist enne kui meil on valmis oma loogikaskeemi prototüüp katsetusteks. Kogu disain nõuab suhteliselt kalli spetsiaalse tarkvara (*CAD-Computer Aided Design*) olemasolu. Kõigi realisatsioonide puhul ei ole sellise tarkvara hankimine võimalik. Selleks, et saada esimene prototüüp on vaja teha kõik maskid ja valmistada mikroskeem. Surte partiide korral on selliste maskide tegemine ja siis nendeabil paljude mikroskeemide valmistamine otstarbekas. Alguses katsetamise ajal on ka suhteliselt aeganõudev, kallis ja tülikas teha muudatusi mida siiski ilmselt vältida ei saa. Muudatused võivad olla tingitud nii disaini vigadest kuika tellija poolsetest nõudmistest muutustest mis kerkivad esile katsetamise faasis.

Eelised :

- Suurte seeriade puhul odavam toota;
- Turvalisus (Security);
- Väiksem komponentide arv;
- Suurem komponentide tihedus vähendab energia kulu ja suurendab töökiirust.

Puudused:

- Väikeste seeriade korral kõrged projekteerimise ja prototüübi valmistamise kulud;
- Pikk juurutamise ja prototüübi valmistamise aeg;
- Tülikas muudatuste tegemine.

Disain võib olla:

- Full Custom Design – toote jaoks tehakse algusest lõpuni oma mikroskeem (CAD – Computer Aided Design, Silicon Compiler);
- Semicustom Design – kasutatakse valmis toorikuid ja disainitakse ainult osa, mis realiseerib vajalikku toodet (gate arrays, standard cells).

○ Programmeeritav loogika

Siin programmeeritav loogika (Programmable logicinglise keeles) tähendab tegelikult mitte protsessoris täidetava programmi kirjutamist, aga riistvara tooriku konfigureerimist vastavalt oma rakendusele. Programmeerimise all tuleb siin mõista konfigureerimist. Konfigureerimiseks / programmeerimiseks kasutatakse põhiliselt kolm tehnoloogiat:

▪ riistvara programmeerimise tehnoloogiad

1) Staatilise suvapöördusmälu (SRAM) tehnoloogia

SRAM tehnoloogias moodustatakse toorikul (tavaliselt maatsriks)SRAM trigeritest suur nihkeregister. Kandes sinna registrisse bittide ja toimubkikonfigureerimine.

SRAM tehnoloogia omadusi:

- a) Funktsionaalseid blokke ja ühendusi juhitakse SRAM triggeritega; Ühenduselemendid on sama kristalli pinnal;
- b) Konfigureerimine ei ole destrukttiivne protsess;
- c) Programmeerimine toimub pärast toite sisselülitamist ja võimalik on töö ajal ümber konfigureerimine;
- d) Vajalik toite sisse lülitamisel konfigureerimiseks väline mäluga seade, kus hoitakse konfiguratsiooni faili;
- e) SRAM elemendid on suured (5 transistori), nõuavad toidet, infoliine, maandust ja valiku liine;
- f) Saab valmistagda koos muu loogikaga samas CMOS tehnoloogias;
- g) SRAM mäludelementide disaini on palju ja põhjalikult uuritud.

2) Anti-fuse ja Fuse tehnoloogiad

Antifuse tehnoloogia juures tekitatakse kahe metalli vahele voolu impulsiga ühendus. Algselt on metallide vahel amorfne räni millel on väga suur takistus (ühendus metall juhtide vahel praktiliselt puudub) mis voolu impusi toimel sulab ja moodustab ühenduse (väikese takistusega piirkond).

Fuse tehnoloogia korral on juhi teatud piirkondtehtud oluliselt väiksema ristlõike pindalaga ja nüüd saab selles piirkonnas ühenduse voolu impulsiga katkestada. Seega ühel juhul konfigureerimisel tekitatakse ühendusi ja teisel juhul neid ühendusi katkestatakse konfigureerimisel/programmeerimisel.

Anti-Fuse tehnoloogia omadusi:

- α) Modifitseeritud CMOS tehnoloogia ja vajalik eraldi valmistamise etapp, millega valmistatakse juhtide vahele väga õhuke isolatsiooni kiht;
- β) Programmeerimine on destrukttiivne – põletatud ühendus ei ole taastatav;
- γ) Programmeeritakse toitest oluliselt kõrgema pingega ja seega on paljudes kohtades vajalik täiendav isolatsiooni kiht;
- δ) Ei ole võimalik lugeda välja konfiguratsiooni faili;
- e) Sobivad multipleksorite valmistamiseks

3) EPROM, EEPROM ja Flash tehnoloogia

Antud tehnoloogiad on samasugused nagu on vastavate püsimälude programmeerimise tehnoloogiad ja neid on kirjeldatud püsimälude juures.

EPROM, EEPROM ja Flash tehnoloogia omadusi:

- α) Sobib kokku standard CMOS tehnoloogiaga;
- β) Protsess ei ole destrukttiivne;
- γ) Puuduseks on laengute hajumine;
- d) EEPROM ja Flash tehnoloogia korral saab programmeerida mikroskeemi eraldamatta.

- disjunktiiivset normaalkuju realiseerivad lihtsamad maatriks-struktuurid (*PAL Programmable Array Logic, PLA -Programmable Logic Array*)

A programmable logic array (PLA) is a programmable device used to implement combinational logic circuits. The PLA has a set of programmable AND planes, which link to a set of programmable OR planes, which can then be conditionally complemented to produce an output.

In a PAL the logic gates are arranged as a sum-of-products array. In Boolean terms, this means a number of AND gates whose outputs feed into a large OR gate that drives one output. By selecting which inputs drive each AND gate, and which AND gates drive the OR gate, any Boolean function can be created. It can be shown that any Boolean function can be reduced to a sum of products, and can therefore be created by a sufficiently large PAL.

A PAL is programmed by fitting it into a machine called a PAL programmer. PAL programmers are usually general-purpose machines that can program all types of PLD from all manufacturers. A PAL may be programmed only once.

The PAL programmer must be supplied with a description of the PAL's desired configuration. This is usually in the form of a computer text file with a standard format defined by the Joint Electron Device Engineering Council (JEDEC). JEDEC files can be hand-typed by the design engineer or, more commonly, produced by a computer program similar to the language compilers used by software engineers.

- kasutaja poolt programmeeritavad maatriks-struktuurid (*FPGA - Field Programmable Gate Array*)

A field-programmable gate array or FPGA is a gate array that can be reprogrammed after it is manufactured, rather than during the manufacturing — a programmable logic device.

FPGAs are generally slower than their ASIC counterparts, and draw more power. However, they have several advantages such as a shorter time-to-market, and lower development costs (for quantities < 10k). An ASIC can be made that is a so-called hard copy of an FPGA - that is, an integrated circuit with the same functionality as the FPGA, but faster and consuming less power.

To define the behaviour of the FPGA it is required to use a Hardware Description Language (HDL) or a schematic designed using an Electronic design automation tool. Either of these, when compiled, will generate a net list, that can be mapped to the actual fpga architecture. When done the binary file generated is used to (re)configure the FPGA device.

Näited:SRAM, Anti-fuse, EPROM, EEPROM, FLASH

- FPGA-de projekteerimine

Depending on the particular device, the program is either 'burned' in permanently or semi-permanently as part of a board assembly process, or is loaded from an external memory each time the device is powered up. This user programmability gives the user access to complex

integrated designs without the high engineering costs associated with application specific integrated circuits.

How are FPGA programs created? Individually defining the many switch connections and cell logic functions would be a daunting task. Fortunately, this task is handled by special software. The software translates a user's schematic diagrams or textual hardware description language code then places and routes the translated design. Most of the software packages have hooks to allow the user to influence implementation, placement and routing to obtain better performance and utilization of the device. Libraries of more complex function macros (eg. adders) further simplify the design process by providing common circuits that are already optimized for speed or area.

FPGA omadusi :

- Aeglasem,
- Väiksem tihedus;
- Odavam prototüüp;
- Suur seeria oluliselt kallim;
- Alati paindlikult võimalus teha muudatusi;
- Kiiremini tootmisse.

- **Erinevate spetsiaalse riistvara realiseerimise võimaluste kasutusvaldkonnad ja võrdlus**

SRAM - based on static memory technology. In-system programmable and re-programmable. Requires external boot devices. Usually CMOS.

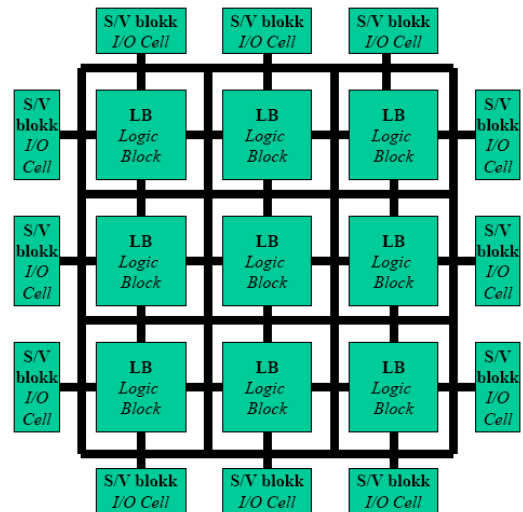
Anti-fuse - One-time programmable. CMOS.

EPROM - Electrically Programmable Read-Only Memory technology. Usually one-time programmable in production because of plastic packaging. Windowed devices can be erased with ultraviolet (UV) light. CMOS.

EEPROM - Electrically Erasable Programmable Read-Only Memory technology. Can be erased, even in plastic packages. Some, but not all, EEPROM devices can be in-system programmed. CMOS.

FLASH - Flash-erase EPROM technology. Can be erased, even in plastic packages. Some, but not all, FLASH devices can be in-system programmed. Usually, a FLASH cell is smaller than an equivalent EEPROM cell and is therefore less expensive to manufacture. CMOS.

Fuse - One-time programmable. Bipolar.



Arvutite riistvara veakindlus.

- **Rikked arvuti riistvaras**

Püsivad rikked (Permanent Faults):

- ühenduste rikked
- purunenud komponendid

- tootmisel tekkivad rikked
- kasutamisel tekkivad rikked
- projekteerimise vead

Mittepüsivad rikked (Nonpermanent Faults):

- keskkond (temperatuur, niiskus, rõhk, ...)
- vibratsioon ja müra
- toide
- magnetväli ja staatiline elekter
- halvad ühendused
- takistuse ja mahtuvuse muutused
- vananemine

• Testimine

Millal testitakse :

- Normaalses töörežiimis. (Online testing, Concurrent testing)
- Spetsiaalses testimise režiimis. (Off-line testing)

Kus on stiimulid :

- Süsteemi sees (Self-testing)
- Eraldi testri väliselt (External testing)

Milliseid rikkeid testitakse :

- Projekteerimise vigu. (Design verification)
- Tootmise vigu
- Tootmise praaki
- Rikkeid (Field testing, Maintenance testing)

Milline on testimise objekt :

- Mikroskeem IC (Component level testing)
- Plaat (Board level testing)
- Süsteem (System-level testing)

Kuidas saadakse testid/oodatavad reaktsioonid :

- Mälust. (Stored pattern testing)
- Genereeritakse testimise ajal. (Algorithmic testing)

Millises järjekorras antakse teste objektile :

- Fikseeritud järjekorras.
- Sõltuvalt eelmise testi tulemustest (Adaptive testing)

Milline on testimise kiirus :

- Normaalsest töökiirusest aeglasemalt (Static testing)
- Töökiirusel (At-speed testing)

Mida jälgitakse :

- Kõiki väljundkombinatsioone
- Funktsiooni väljundkombinatsioonidest (Compact testing)

Millistele objekti punktidele on ligipääs :

- Ainult sisenditele/väljunditele. (Edge-pin testing)
- Sisenditele/väljunditele lisaks ka sisemistele punktidele. (In-circuit testing, Bed-of-nails testing, ...)

Kes kontrollib tetsimise tulemusi :

- Süsteem ise (Self-testing, Self-checking)
- Väline seade-tester. (External testing)

- **Testitava riistvara projekteerimine**

- 1) kombinatsioonskeemid
- 2) järjestikiskeemid
- 3) Boundary-scan Standard IEEE 1149.1

- **Veakindlad koodid**

- 1) vigu avastavad koodid
- 2) vigu parandavad koodid

- **Töökindluse tõstmine**

- 1) püüda ennustada riknemist
- 2) projekteerida nii, et oleks töökindel