

Arvutispetsialistidel on 10 probleemi:

- Targad kasutajad
- Rumalad kasutajad

Doktorikraadiga vanemlektor läheb koju ja leiab naise ühe oma tudengiga voodist.

“Ja mida teie, intelligentne inimene, mulle nüüd siis õige ütlete?” küsib ta tudengilt.

“Nagu intelligent intelligendile?” küsib tudeng vastu.

“Jah.”

“Koputama peab!”

EET3010 JUHTIMIS - JA ANDMESIDETEHNIKA ALUSED

Kevad 2025

Digitaaltehnik – arvutid

Martin Jaanus

NRG-308

martin.jaanus@ttu.ee

56 91 31 93

Õppetöö : <http://isc.ttu.ee>

Õppematerjalid : <http://isc.ttu.ee/martin>

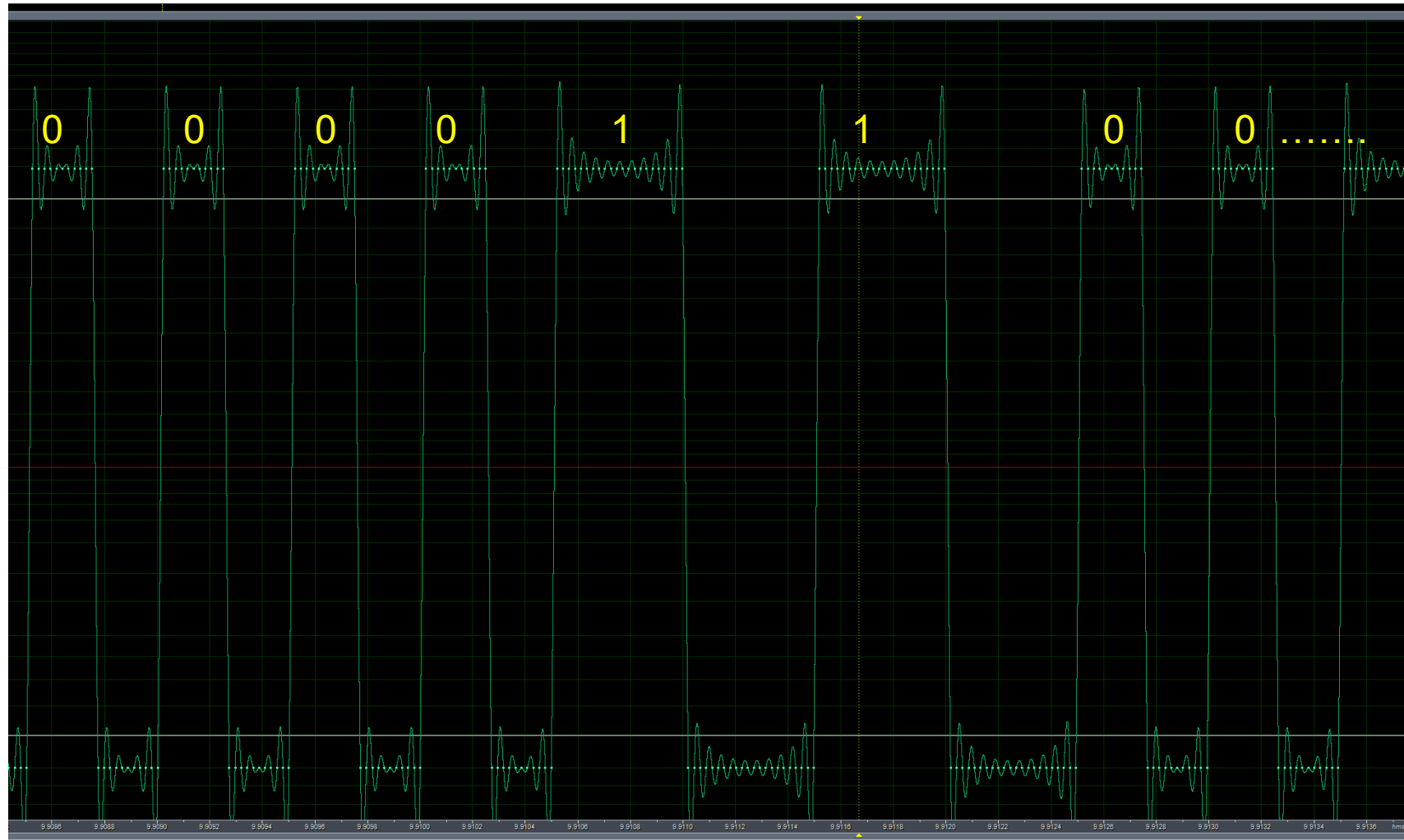
Teemad

Digitaalelektronika analoogmaailmas

- Arvuti üldine tööpõhimõte
- Andmete haldamise alused

DEMO - ZX spectrum andmete laadimine magnetlindilt

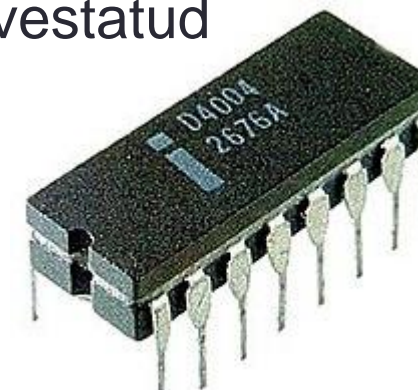
- **Sagedus** modulatsioon
- Andmete jadaedastus
- Loogiline 0 – 488 μ s
2046 baud
- Loogiline 1 – 978 μ s
1023 baud
- Keskmise 1364 baud
(kui 0 ja 1 võrdset)
- Pilt laetakse realajas
- demos videomällu



Sarnaselt edastati infot omal ajal üle telefoniliini (modulatsioon keerulisem, lisaks amplituud ja faas)

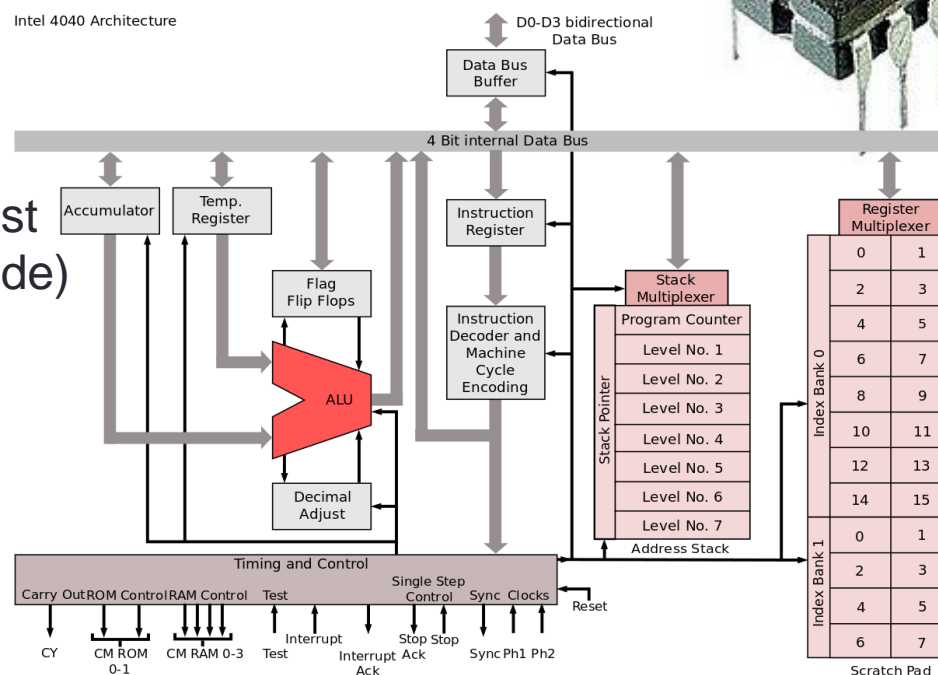
Mikroprotsessor

- Mikroprotsessor on mitmeotstarbeline programmeeritav seade, mis võtab sisendina vastu digitaalse info, töötleb seda vastavalt mälus salvestatud masinakäskudele ja annab tulemuse väljundina. (wikipedia)
- Tehnoloogia ajas muutub, idee jääb samaks



Saab luua lihtloogikaelementidest

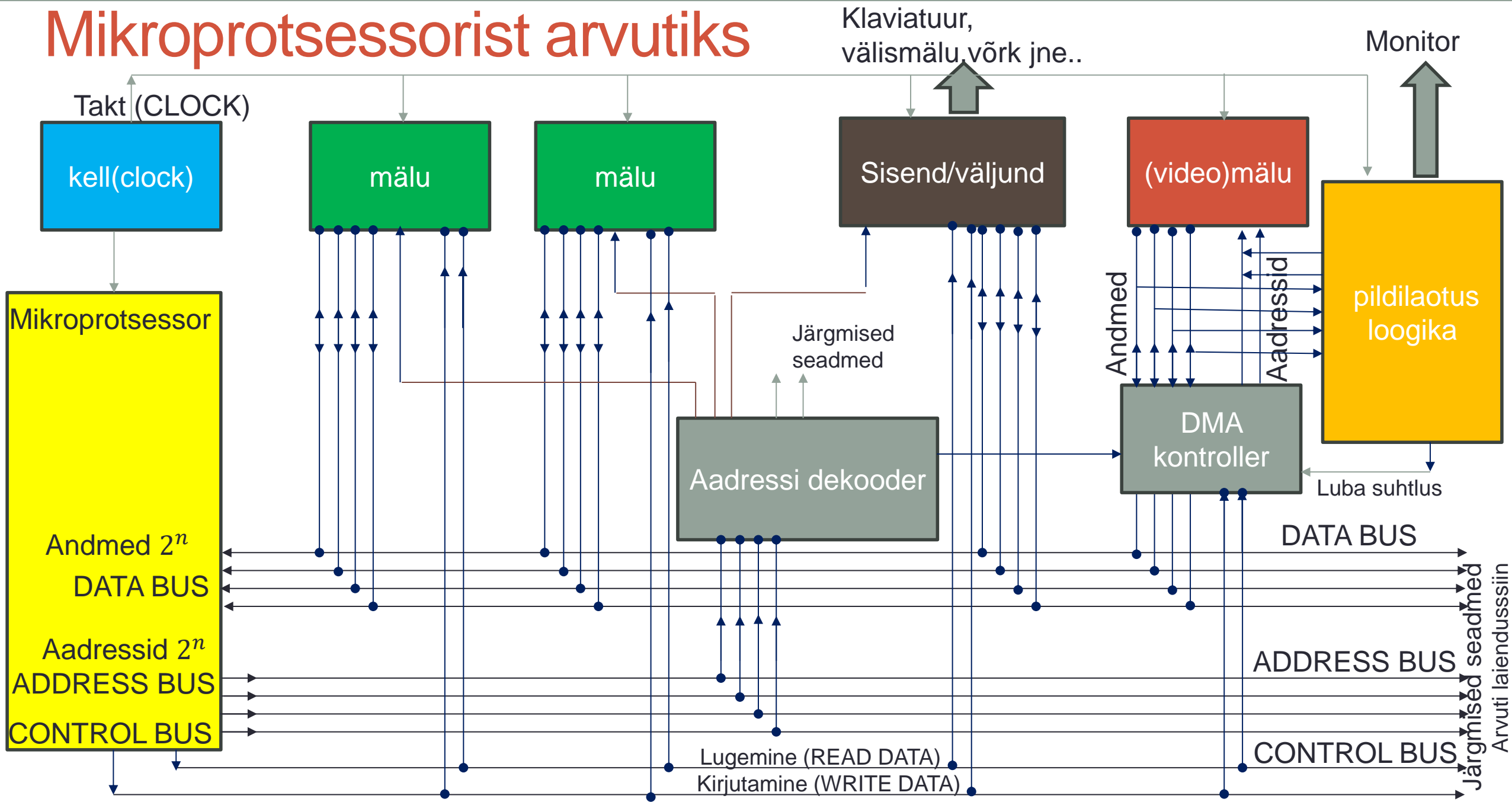
- ALU (aritmeetika-loogika seade)
- Aadressiregister
- Käsuregister
- Inforegister
- Andmesiin
- Aadressisiin



https://en.wikipedia.org/wiki/Intel_4004

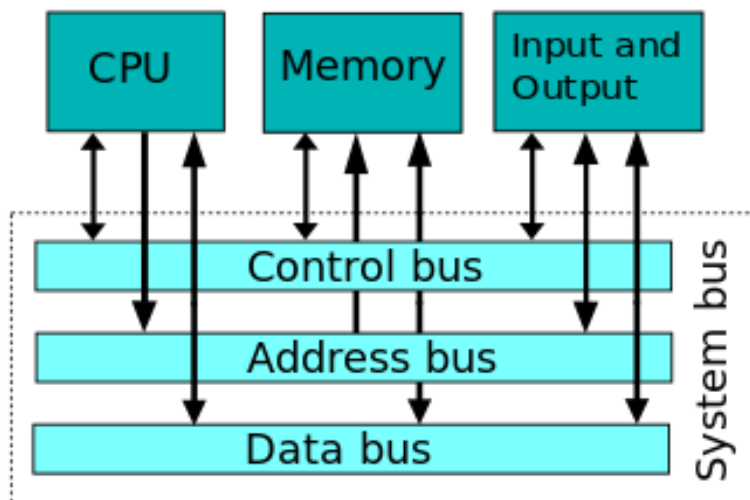
Intel 4004 (1971)

Mikroprotsessorist arvutiks

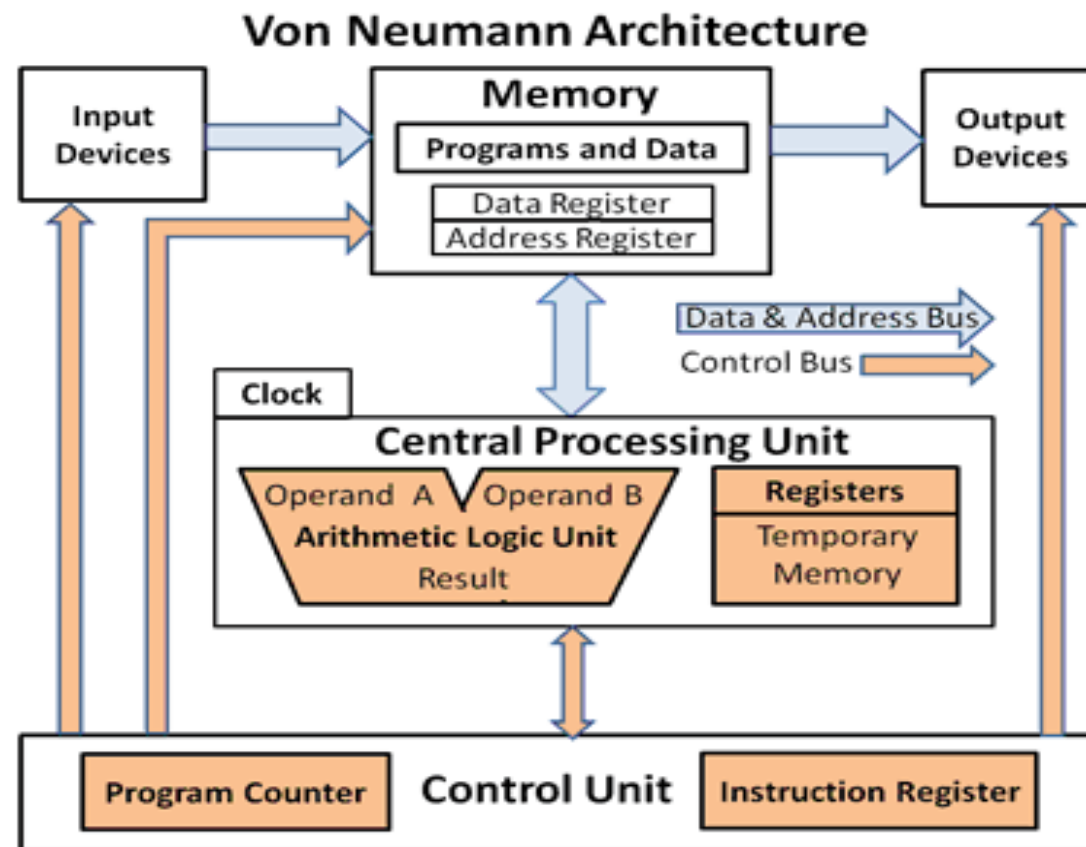


Von Neumanni arvutistruktuur

- Olemuselt digitaalautomaat ([John von Neumann](https://en.wikipedia.org/wiki/John_von_Neumann), 1945)
- Aritmeetika loogikaseade
- Juhtseade (käsuloendur)
- Mälu (käsukoodidele, andmetele)
- Välised seadmed



https://en.wikipedia.org/wiki/Von_Neumann_architecture



PC arvutid

X86 mikroprotsessorid

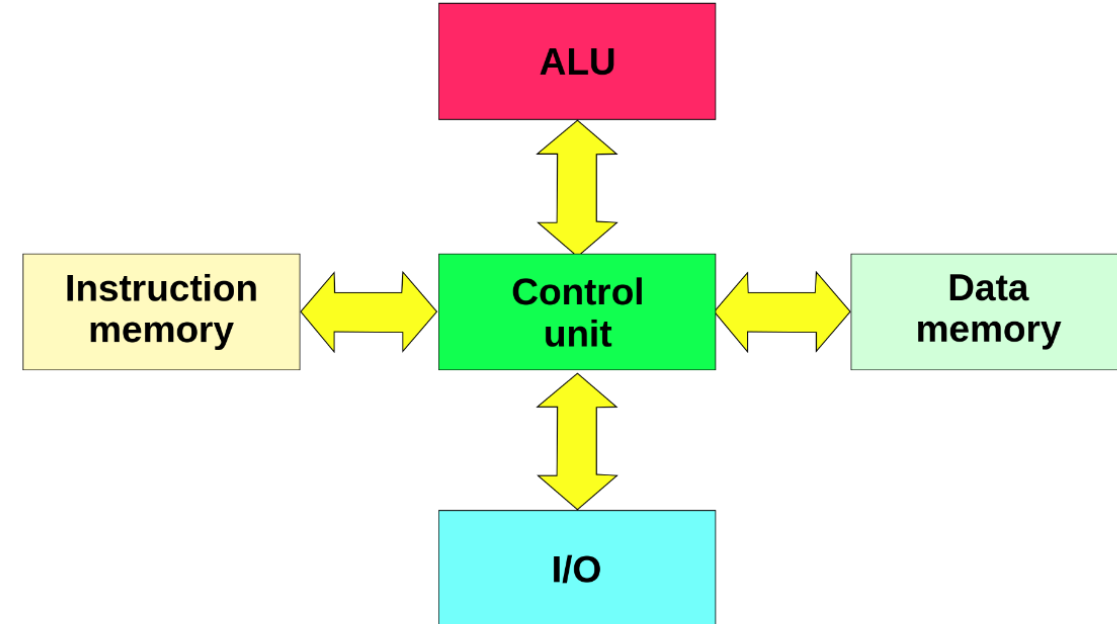
- 1978 Intel 8086 täissiiniga 16 bitine mikroprotsessor
- Nimetus X86 tuleb sellest, et hilisemate protsessorite nimed lõppesid 86ga (80186, 80286, 80386 ja 80486)
- Arvuti vananes 90ndatel mõne aastaga. Areng kiire.
- Erinevad tootjad, arhitektuur muutus kohmakaks, kuid suur eelis – **ühilduvus eelkäija protsessoriga !**
- **Tänu sellele : PC arvutite võidukäik 1981- tänaseni**
- Vanem tarkvara „jookseb“ uuema arvuti peal. (tänapäeval on see ka turvaprobleemiks)
- CISC protsessorid (complex instruction set) – palju käske toetatud riistvaras.



Esimene PC – IBM PC (1981)
Pilt- wikipedia

Harvardi arvutistruktuur

- Arvutis Harvard Mark I (1944)
- Eraldi andmed ja käsud (nii mälu kui ka töötlus)
- Eelis, saab samaaegselt tegeleda mõlema mäluga
- Tänapäeval enamkasutatav (nutiseadmed, kontrollid)



ARM mikroprotsessorid

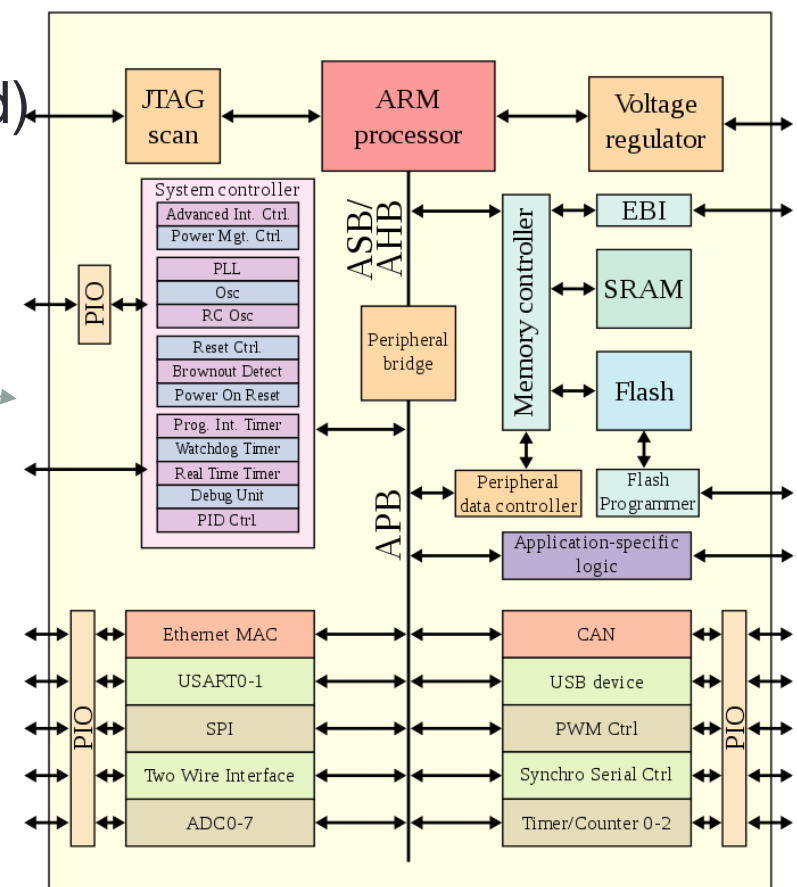
- ARM - Advanced RISC Machine, Acorn RISC Machine
- RISC - reduced instruction set computing (vähendatud käsustikuga protsessor)
- Lihtsam siseehitus, suurem kiirus
- Telefonid, tahvelarvutid (iseegi kohati võimekamad kui PC arvutid)
- Mikrokontrollerid
- Hea näide on **Arduino Uno** „ehituskiviks“ olev ATmega168 kontroileri sees RISC protsessor – vähe kāske.



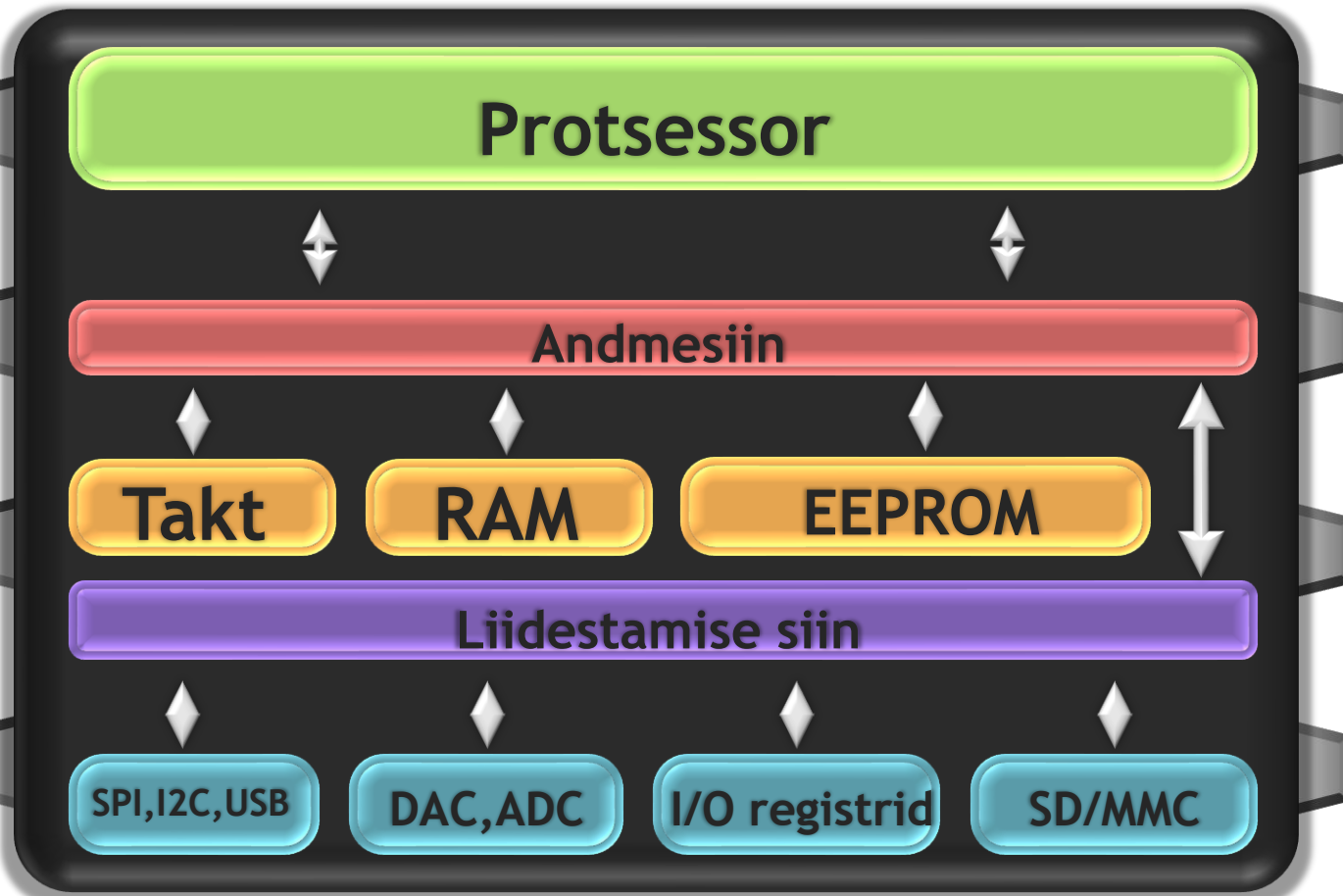
Raspberry Pi

https://en.wikipedia.org/wiki/ARM_architecture

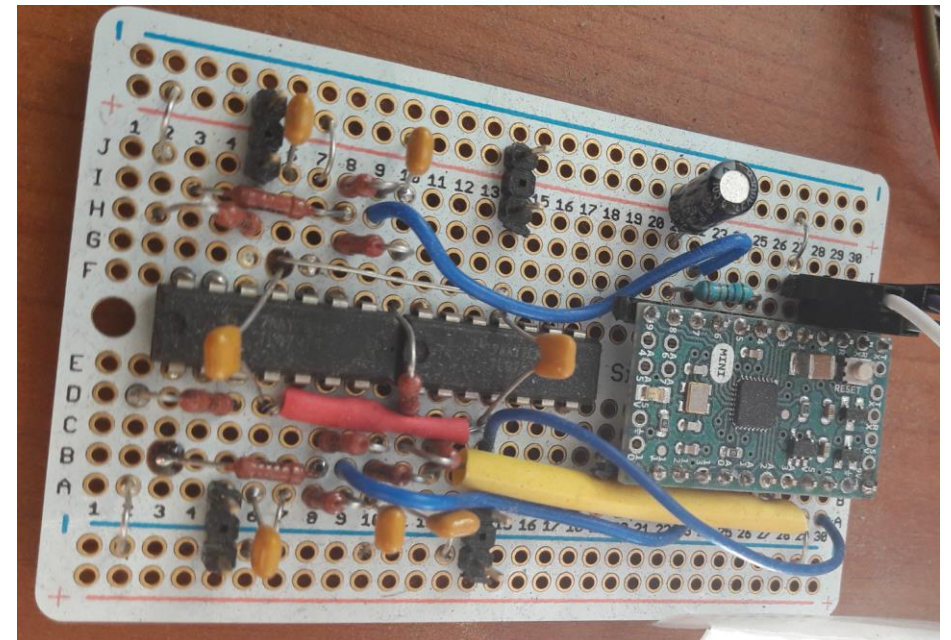
ARM7



Mikrokontrolleri eripärad

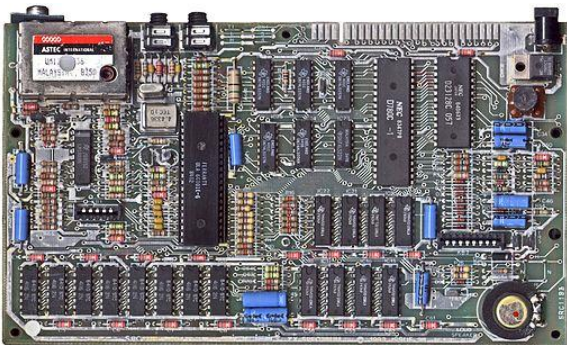


- Reeglina üsna aeglased või võimalus teha aeglaseks. (energiakulu !)
- Programmeeritakse vaid konkreetse ülesande jaoks.
- Programmeerija peab teadma üldist tööpõhimõtet.
- Sisend-väljund klemmid reeglina universaalsed ja multifunktsionaalsed.
- Kasutusel arendusplatvormides (nt. Arduino, Pi...)
- Odavad, programmeeritakse madala taseme keeles (C, ASM)



Arvuti siseehitus (näide – ZX spectrum)

- 8 bitine koduarvuti (1982-199x) . Suurbritannia (Sinclair)
- **8 bitine protsessor Z80A**, mälu ruum 64 kb (16 kB ROM, 48kB RAM) , välismälu – kassettmagnetofon, monitoriks televiisor – **sobib tänapäeval õppematerjaliks (näide inseneeria tipp-teosest)**
- Töökiiruselt sarnane Arduino Uno-ga
- Väga odav tolle aja kohta – £179
- Oli ülipopulaarne ka Eestis - 80ndate lõpp, 90ndate algus (kloonid - ZX Leningrad , iseehitamine)
- Tänapäeva mõistes (**ülimalt**) **lihtne** skeem võimaldab iseehitust.
- Mängude arv tuhandetes, neid tehakse ka tänapäeval
- Tänapäeval – emulaatorid, hobialektroonika, skeemilahendused kaasaegsetel komponentidel
- <https://www.specnext.com/> - kaasaegne riistvaras realiseeritud 8 bitine arvuti .



https://en.wikipedia.org/wiki/ZX_Spectrum

ZX spectrum 2 →

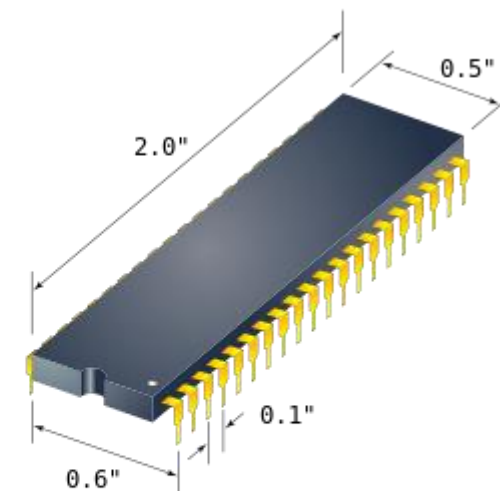
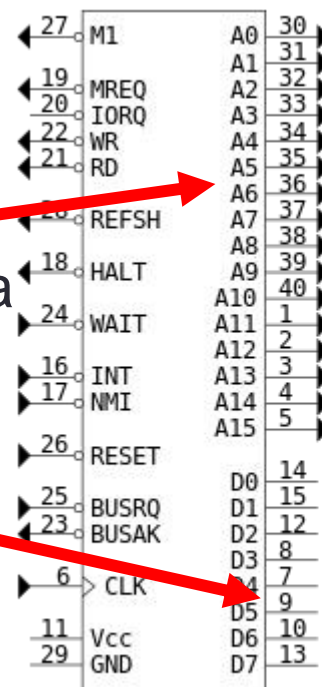


Z80

- 8 bitine , 16 bitise aadresssiiniga mikroprotsessor
- Võimaldab adresseerida $2^{16} = 65536$ erinevat mälupesasa (64kB)
- Koduarvutid, mängukonsoolid
- 1976.....2,5 MHz takt tänapäeval...kuni 1GHz (2019, Nasa NSG-7253)
- Tänapäeval realisatsioon programmeeritavas loogikamaatriksis

Miks 8 bitti ?

- Paljudes kohtades pole lihtsalt rohkem vaja .
- Väga lihtne (võrreldes tänapäeva 64 bitiste protsessoritega)
- Hea õppematerjal arvuti hingeeluga tutvumiseks.



Mäned signaalid mikroprotsessoritel

A – aadressid

D – andmed

WR – kirjutamine

RD – lugemine

WAIT – oota - väline võimalus panna ootele

CLK –kell

NB ! Osad signaalid on negatiivses loogikas.

Demo kah



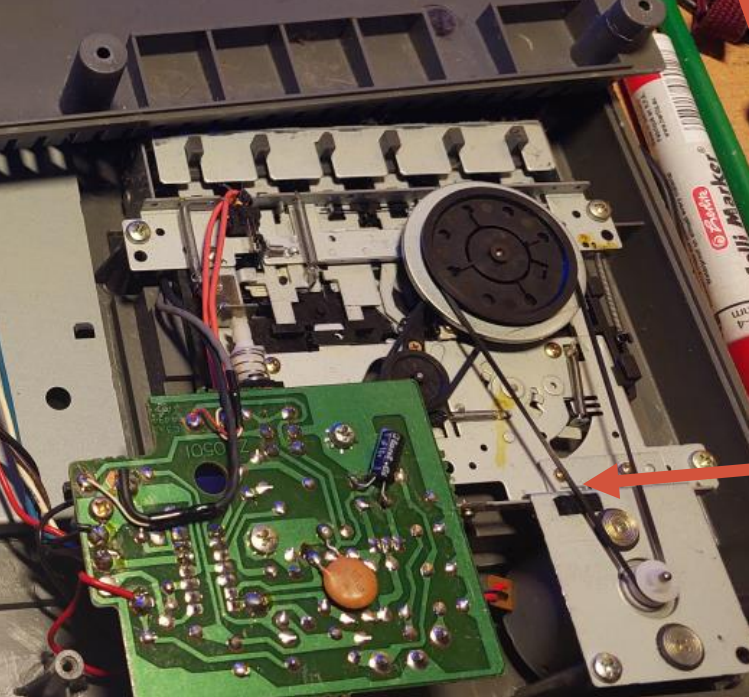
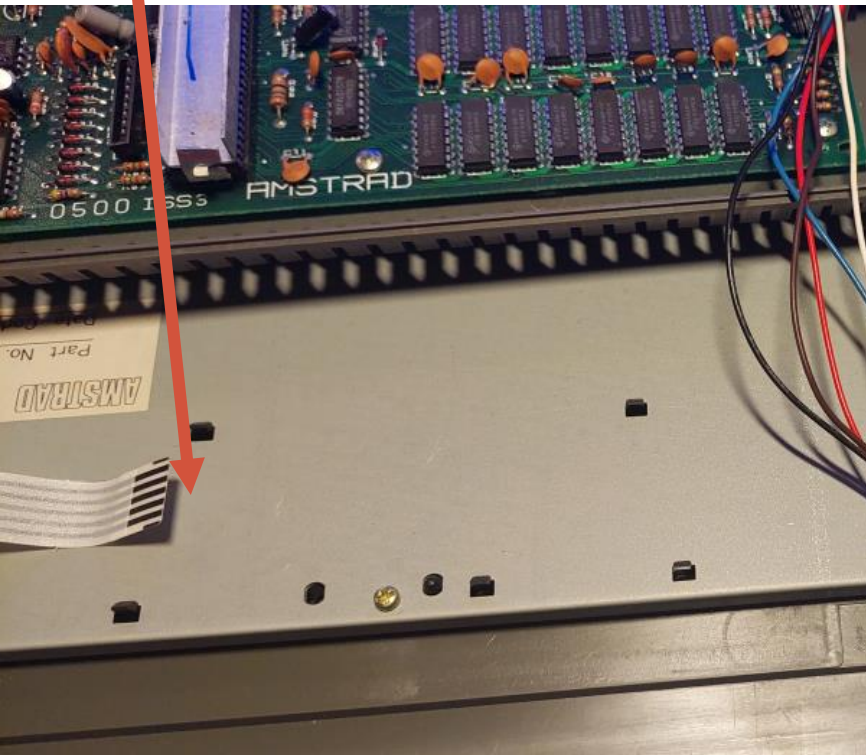
Lapsepõlveunistus
ZX Spectrum

Seestpoolt

Süsteemisiin
(arvuti taga)

Video
Siiniloogika
Takt

Klaviatuur

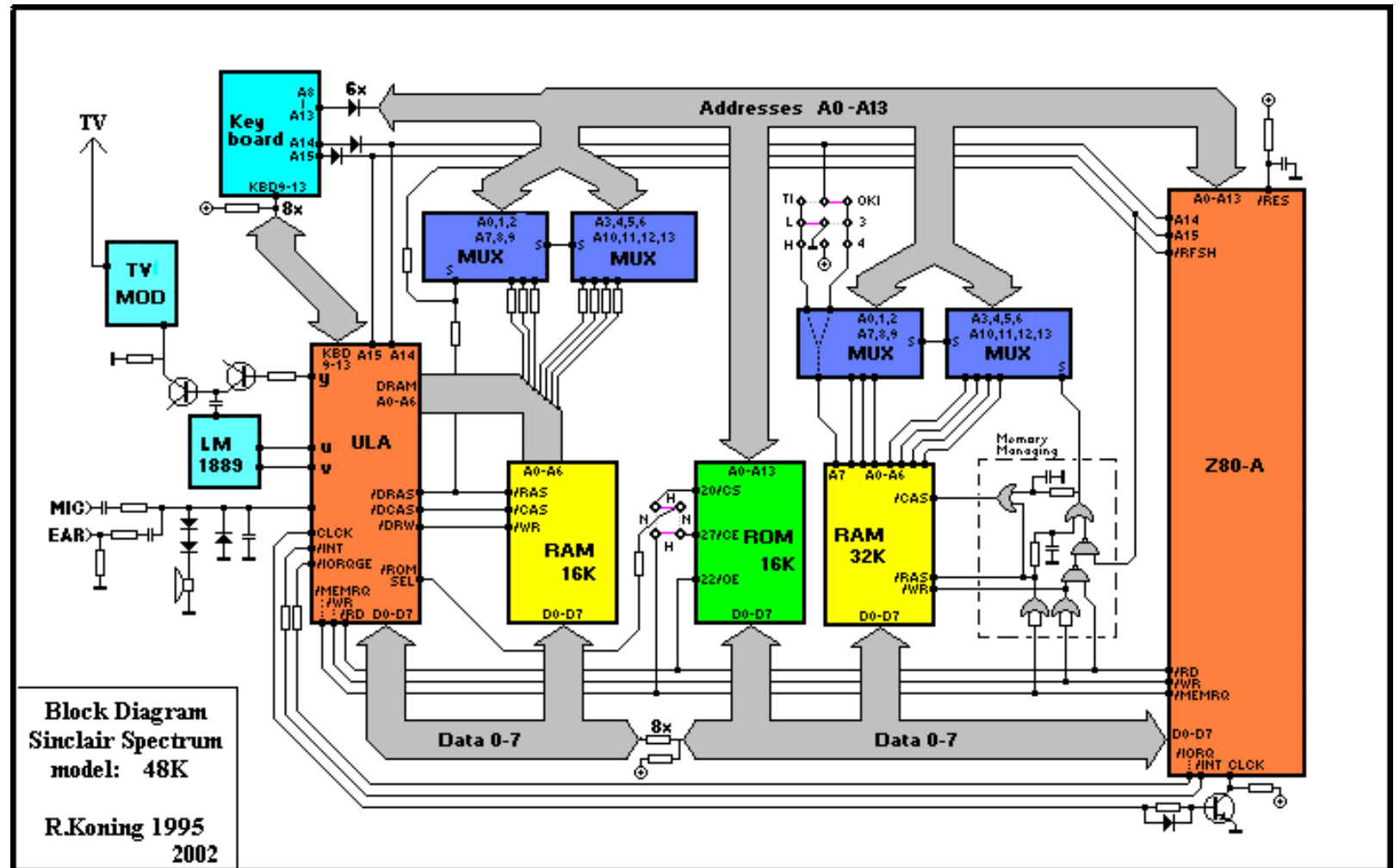


Mälu (RAM) , (ROM)
Protsessor
Heli
Aadressi dekooder

Kassettmagnetofon

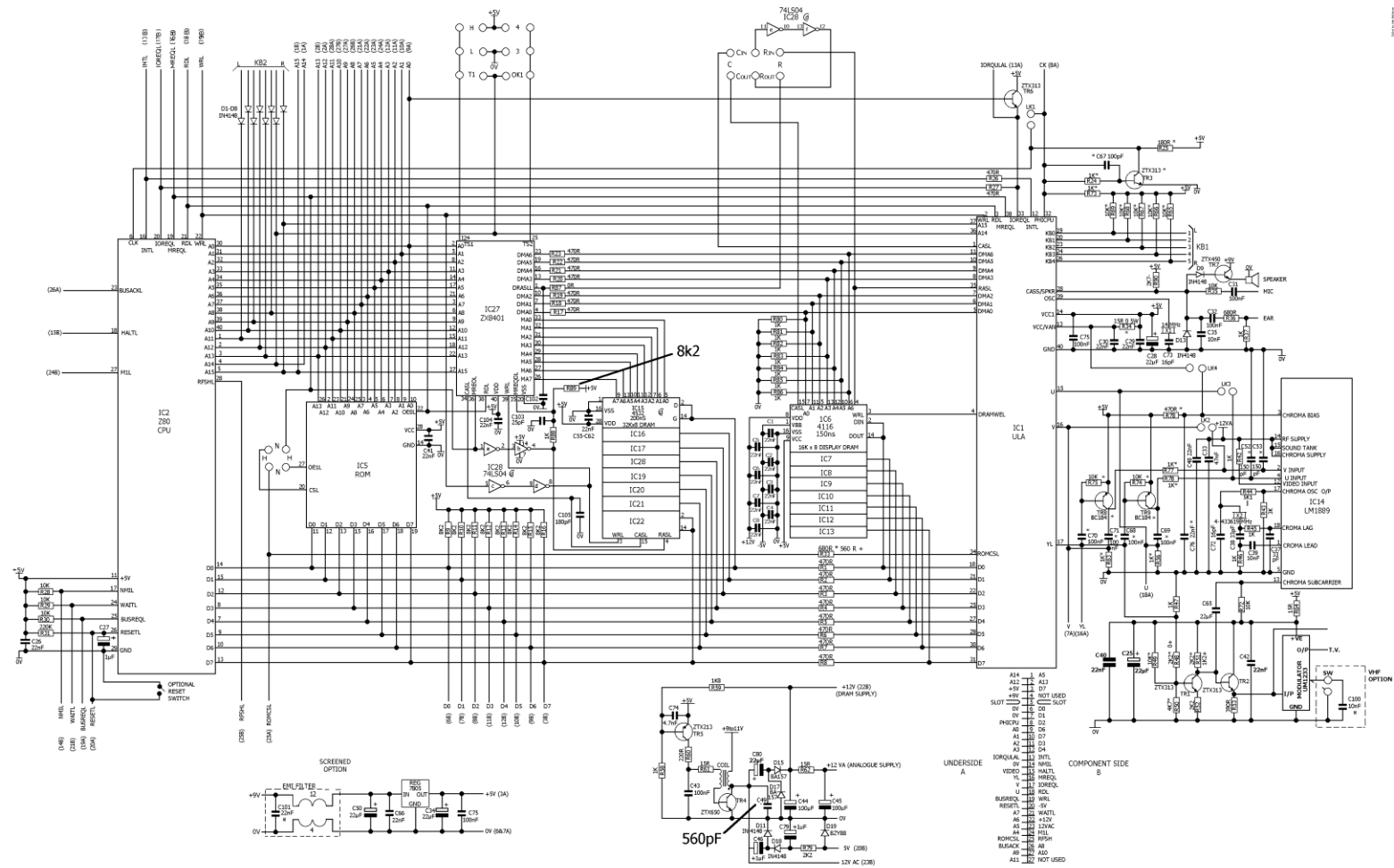
ZX spectrum

- <http://www.worldofspectrum.org/documentation.html>



ZX spectrum (elektriskeem)

- <https://spectrumforeveryone.com/wp-content/uploads/2017/08/ZXSpectrumIssue3-Schematics.gif>



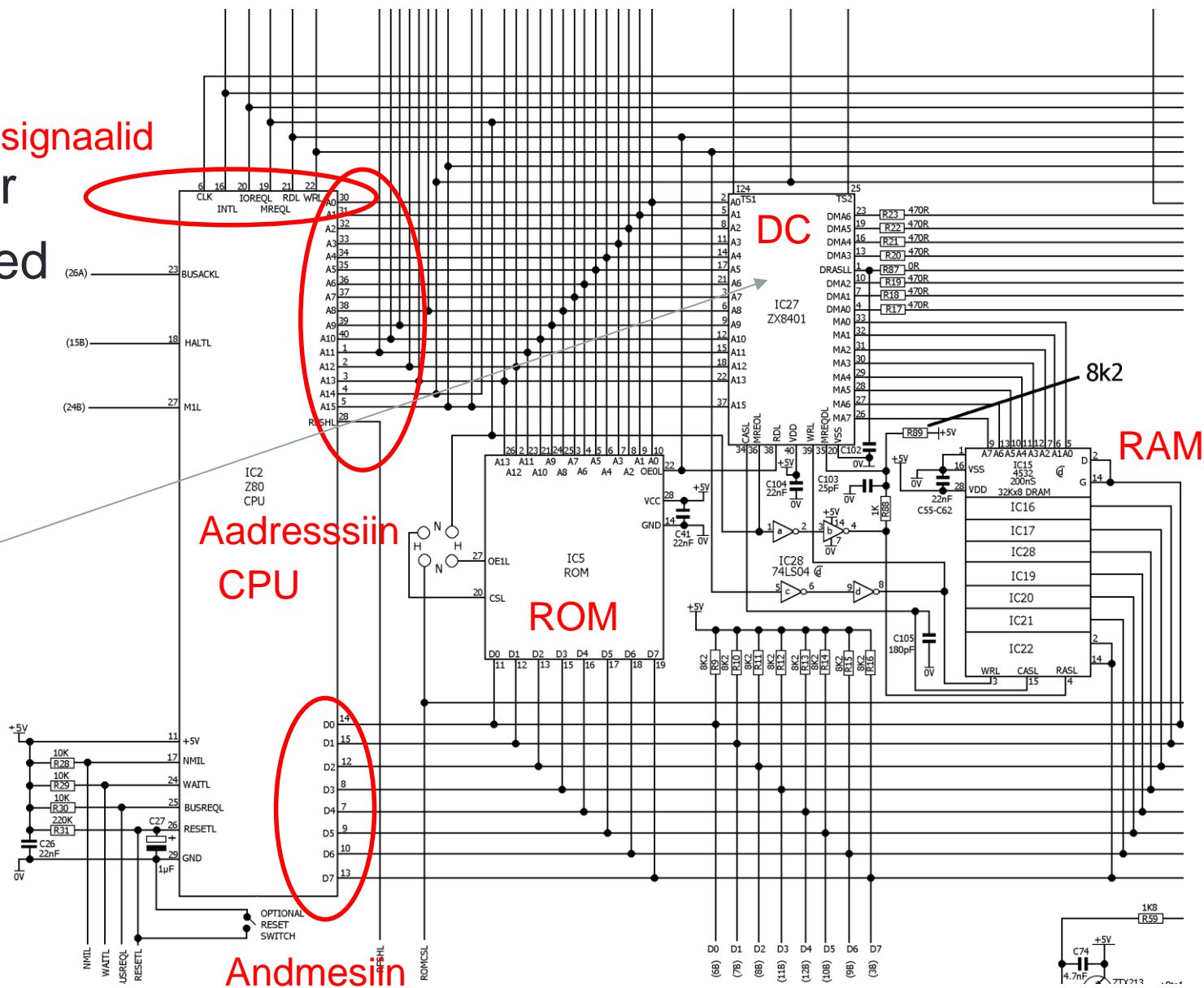
ZX SPECTRUM (ISSUE 6A) CIRCUIT DIAGRAM FIGURE 6

ZX spectrum

Siini juhtsignaalid

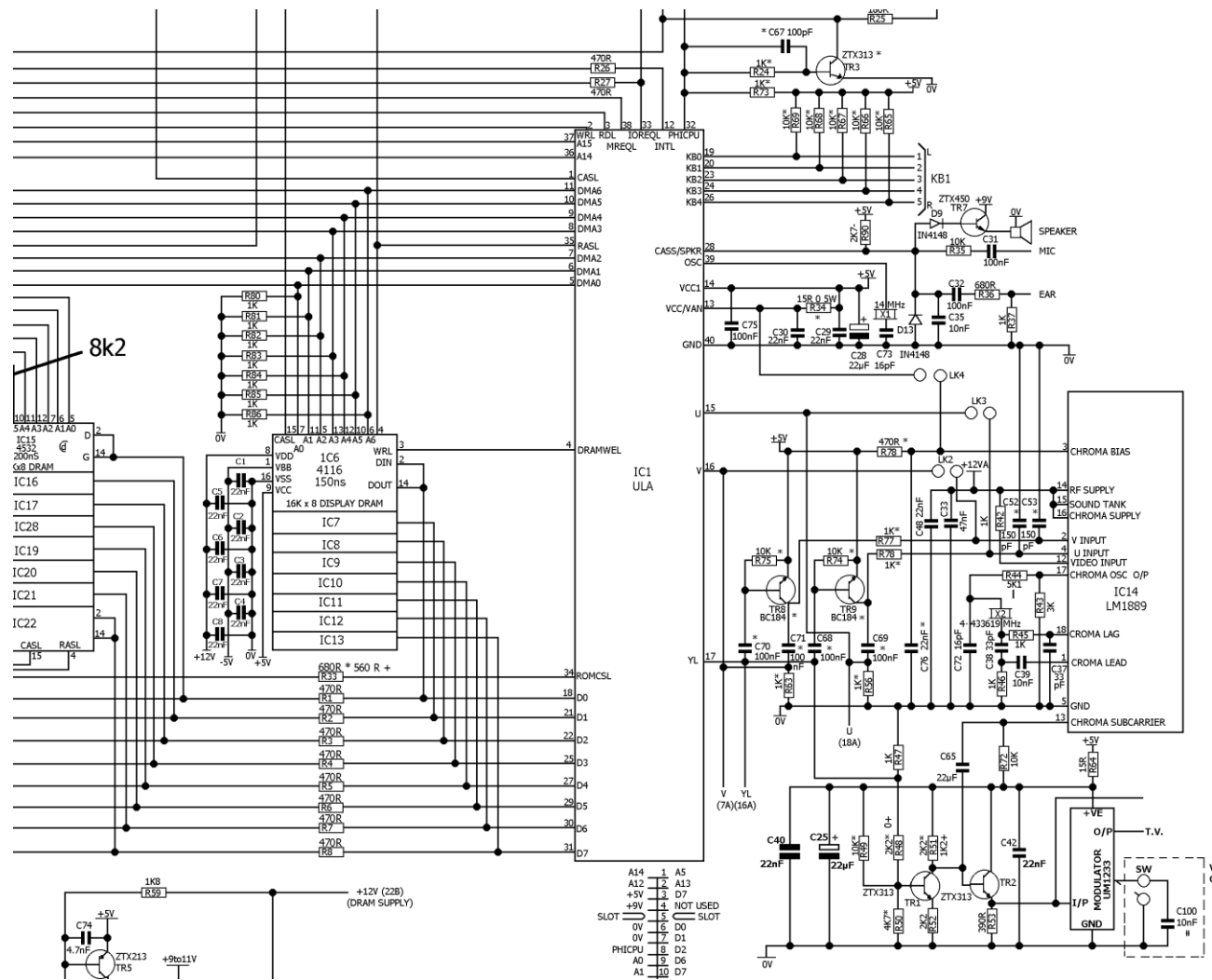
- Protsessor ja ühendused

Adressi dekooder



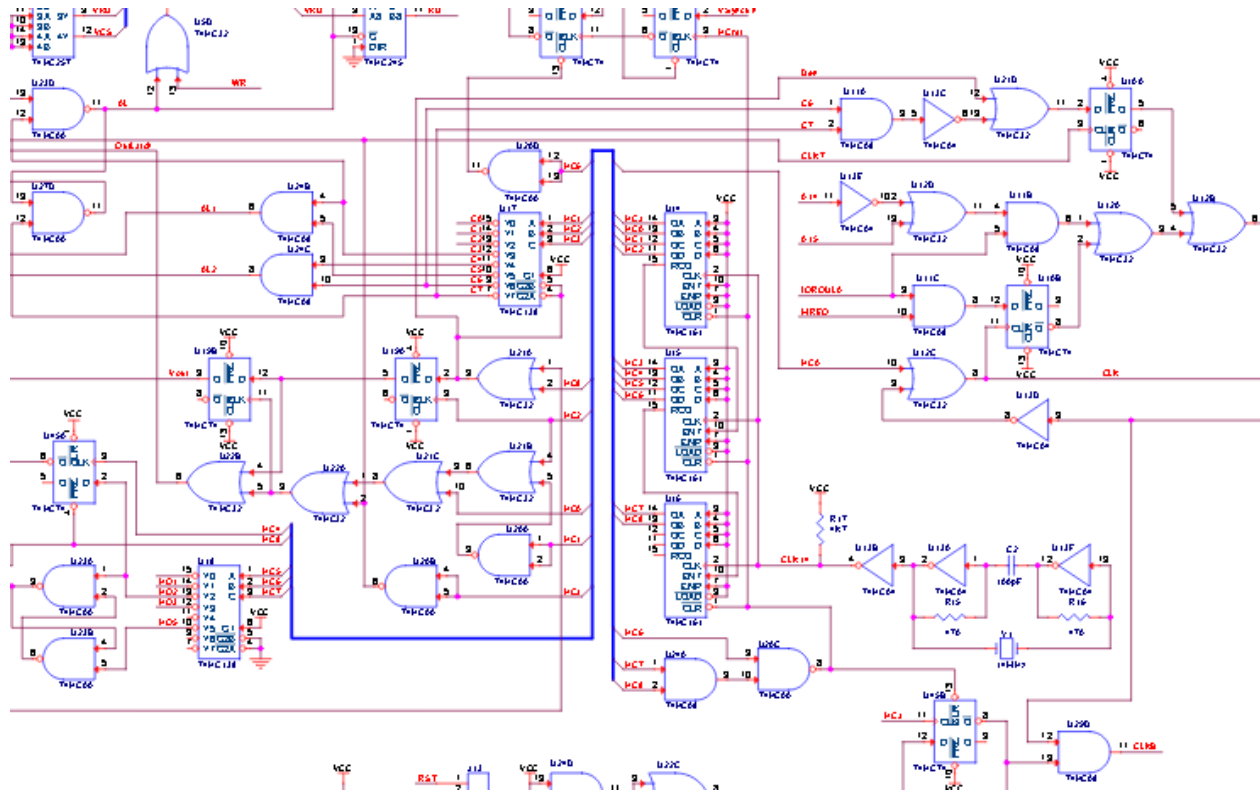
ZX spectrum

- Sisend – väljund
- Spetsiaalne mikroskeem



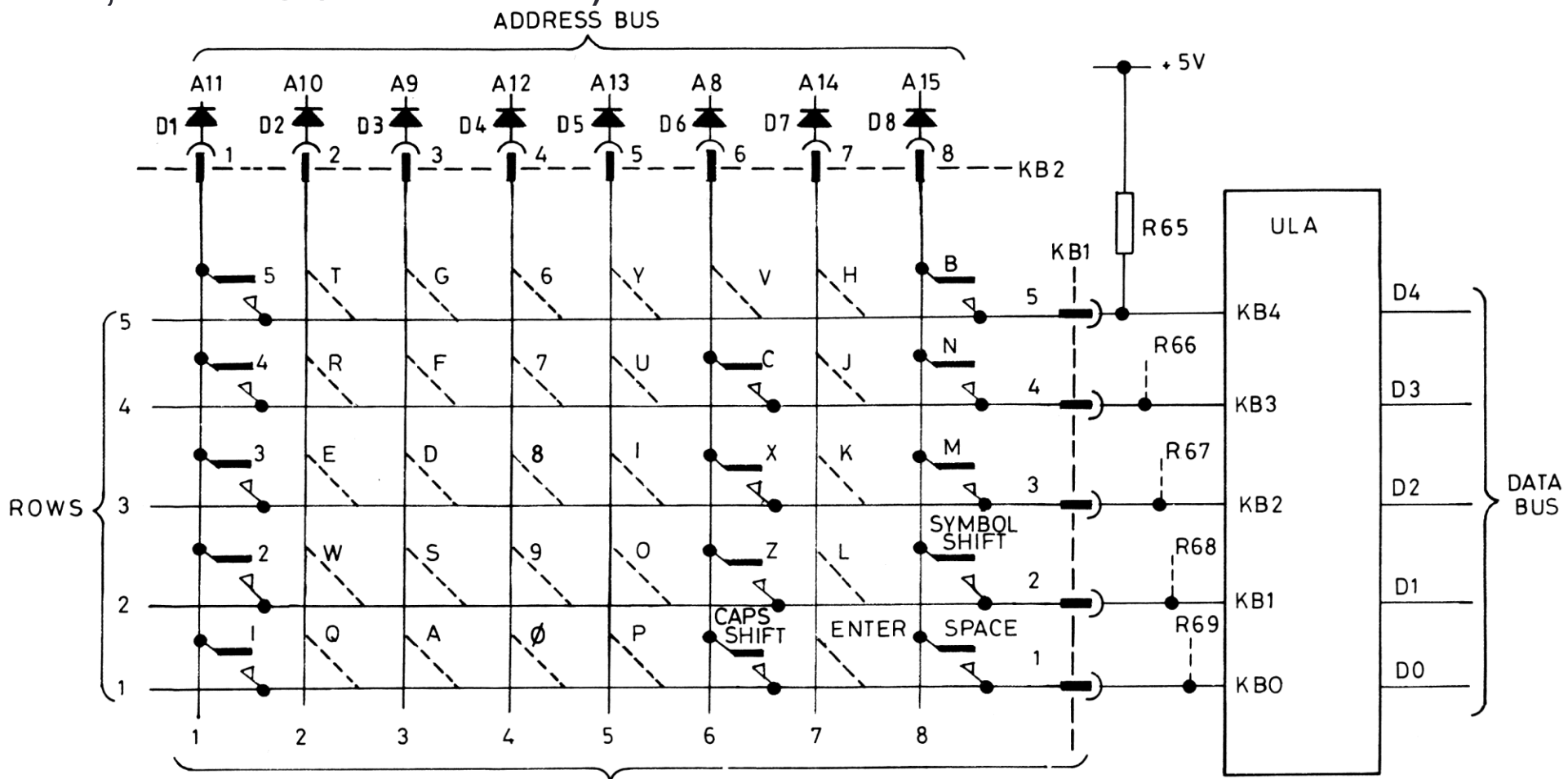
ZX spectrum

- Skeem ilma ULA-ta (fragment)
- http://trastero.speccy.org/cosas/JL/Harlequin/Documentacion/harlequin_rev_g.pdf



ZX spectrum

- Klaviatuur (maatriksühendus) – sama idee ka PC-I (PC arvutil on eraldi kontrollor, millel USB ühendus)



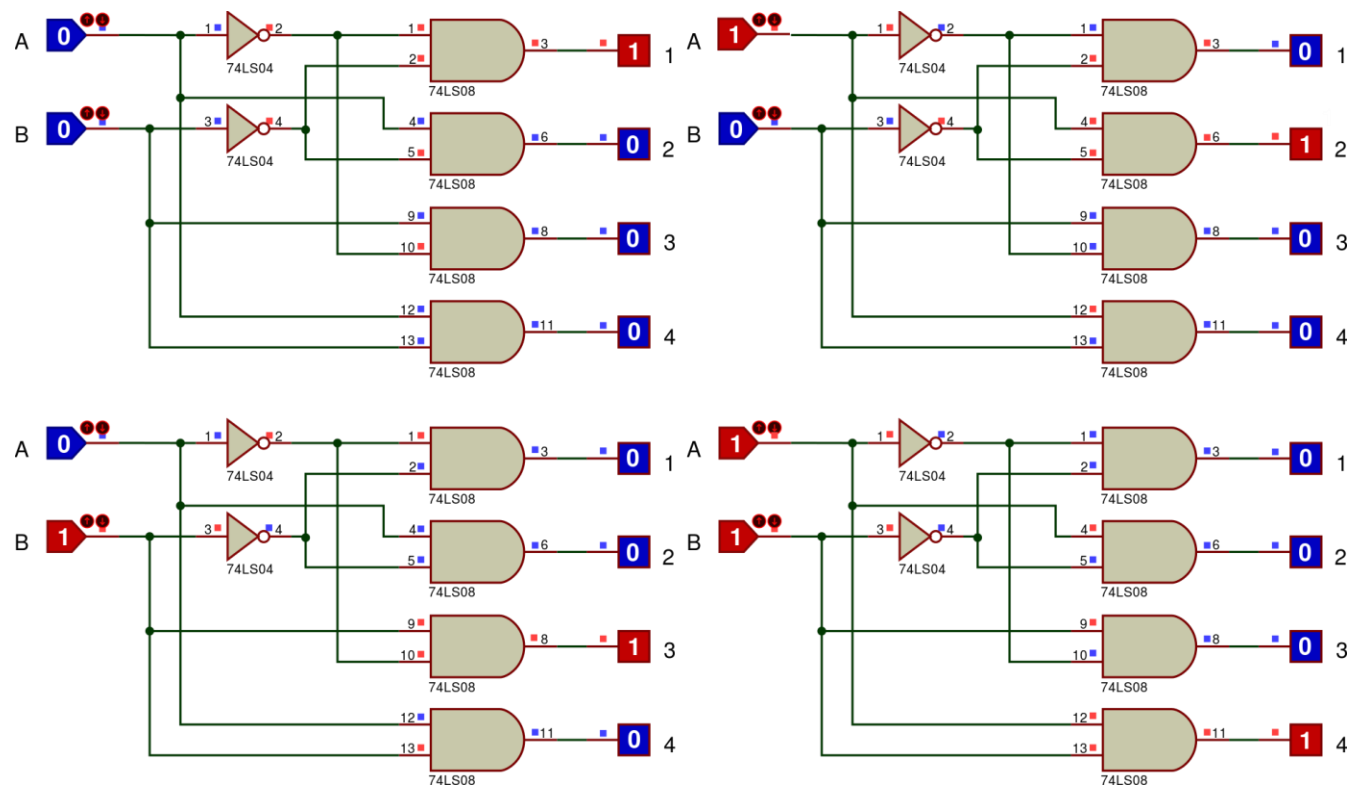
MÄLU



16 GB DDR4 mälmoodul sülearvutile
Pilt osta.ee

Aadressi dekooder

- (Adress multiplexer)
- Aadressisiin on 2^n bitti
- Nt Õige “mälupesa” leidmine
- Enamik on mikroskeemi sees
- Eraldi loogikaga valitakse aadressisegmente
- Antud skeem võimaldab
- 64 kB mälu jagada 16 kB plokkideks kui A ja B on “vanemad” aadressibitid



Kahebitine dekooder, pilt Wikipedia .

A	B	Y1	Y2	Y3	Y4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

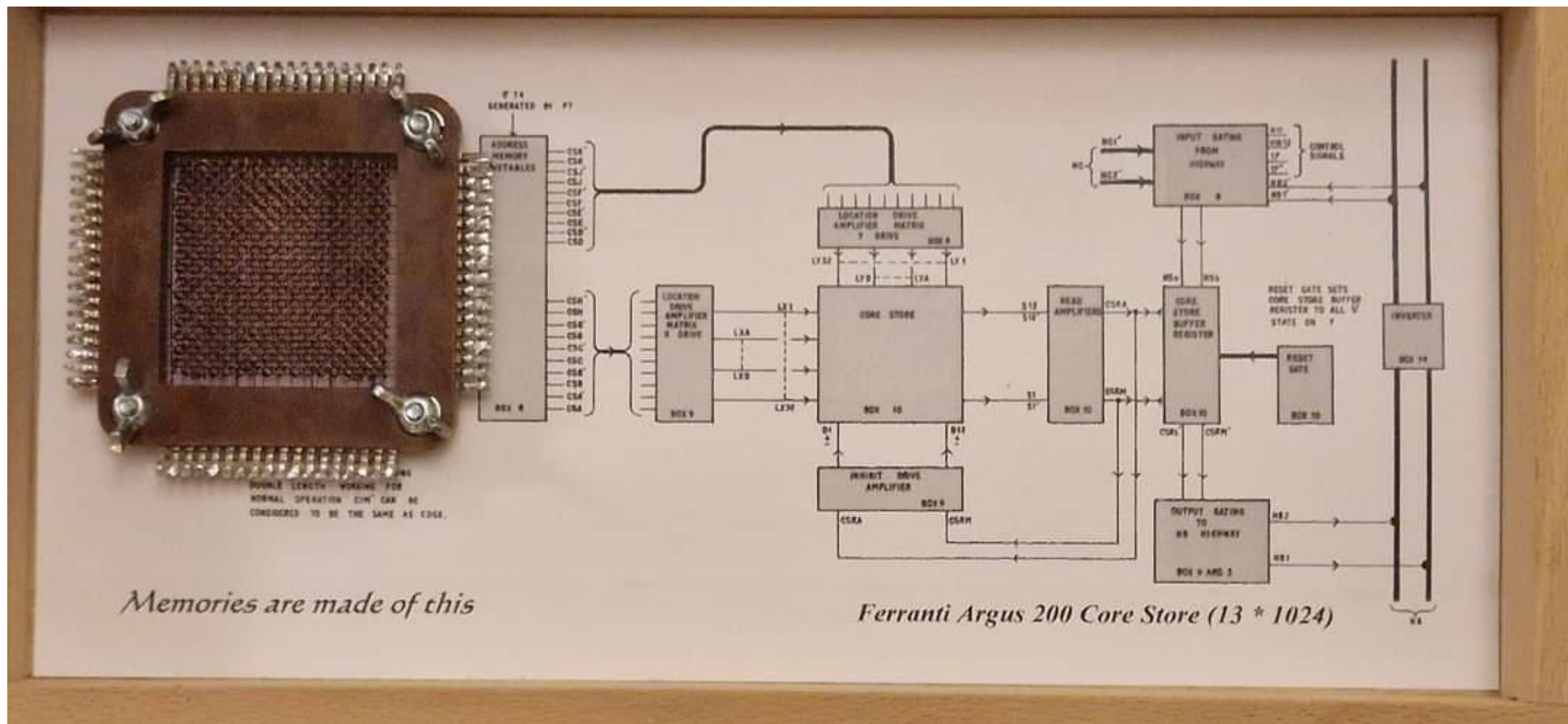
Kunagi tehti nii

- Mälu “mikroskeem-lamp” (ei oska paremat nime anda)
- Või üksikutel lampidel trigerid (arvutid suured ja aeglased tänapäevamõistes)
- Lampe vaja kütta



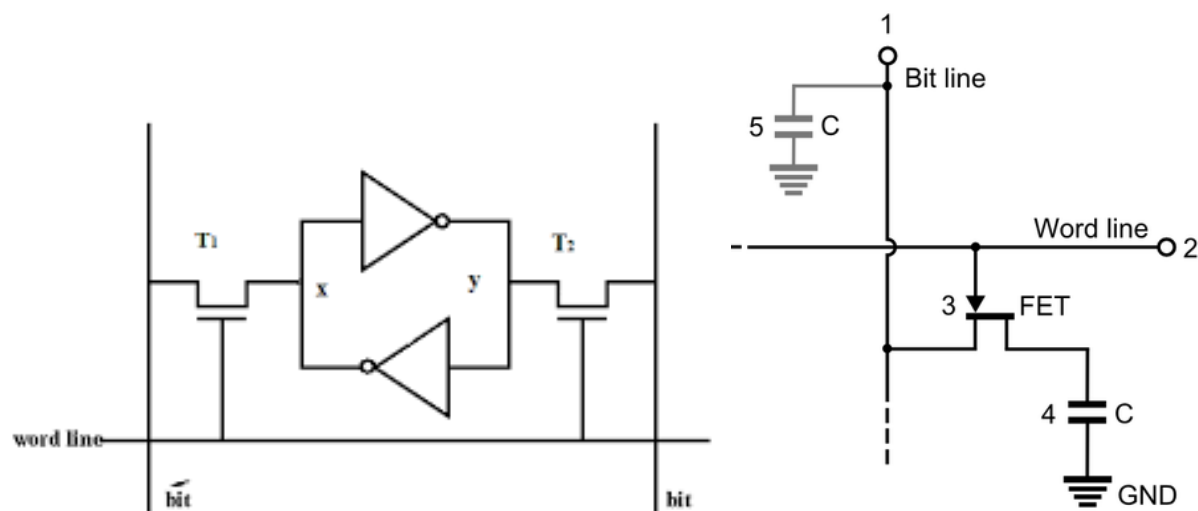
Tehti ka nii

- Ferriitmälu (info salvestati magnetvälja) – 13 kilobitti



Tänapäevane arvutimälu RAM

- Juhupöördusega mälu (Random Access Memory)
- Staatileine (basserub trigeril), kiirem, kallim
- Dünaamiline (mäluelement – kondensaator), võtab vähem ruumi, aga vajab värskendust (refresh) – laeng aja jooksul väheneb. Reeglina kaasajal mõned nanosekundid
- **Info hävineb toite katkestamisel !**



SRAM

DRAM

Levinum tüüp tänapäeval DDR (double data rate) – topeltestuskiirusega. Info edastatakse taktisignaali mõlemal frondil
Hetkel kõige uuem on DDR5 mälu

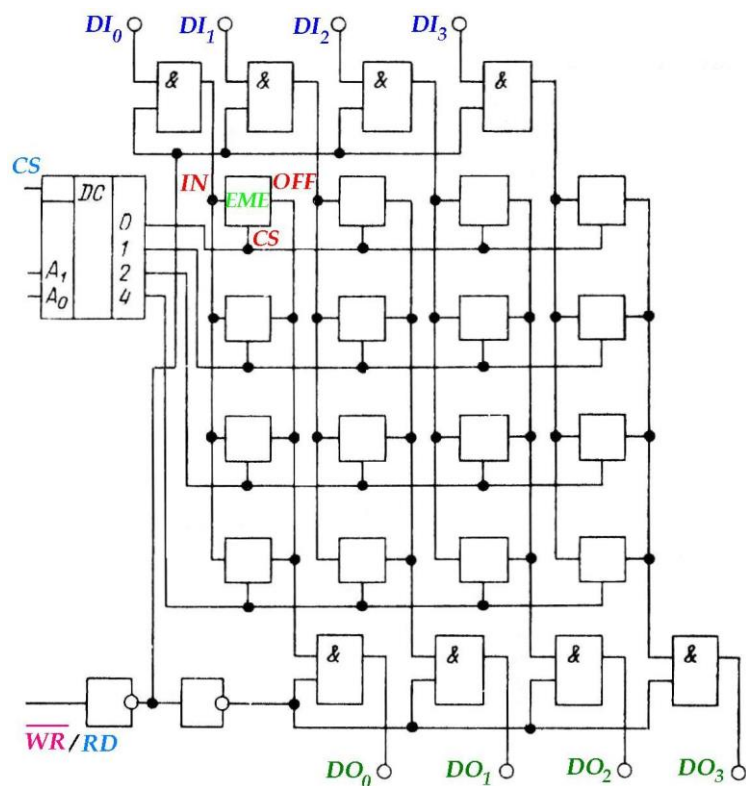


DDR5 sülearvutimälu 16 MB

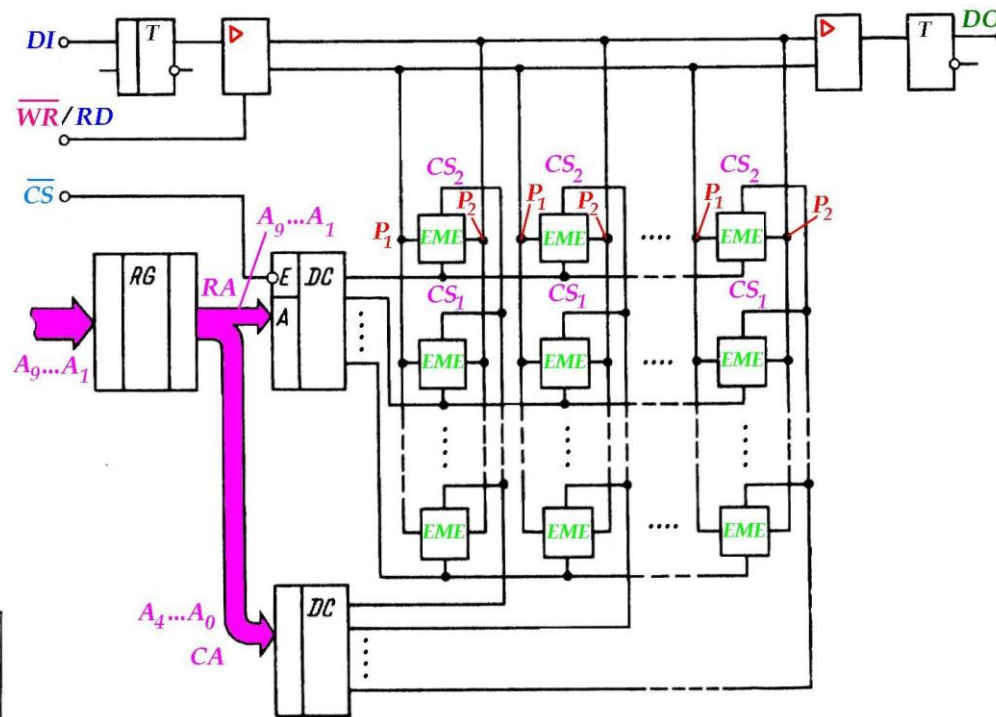
https://en.wikipedia.org/wiki/Random-access_memory

Suvapöördus - arvutimälu

- Iga biti jaoks eraldi mäluelement
- Vajab adresseerimist
- Sisend ja väljund on omavahel sageli koos



Ühemõõtmeline (4 bitti)



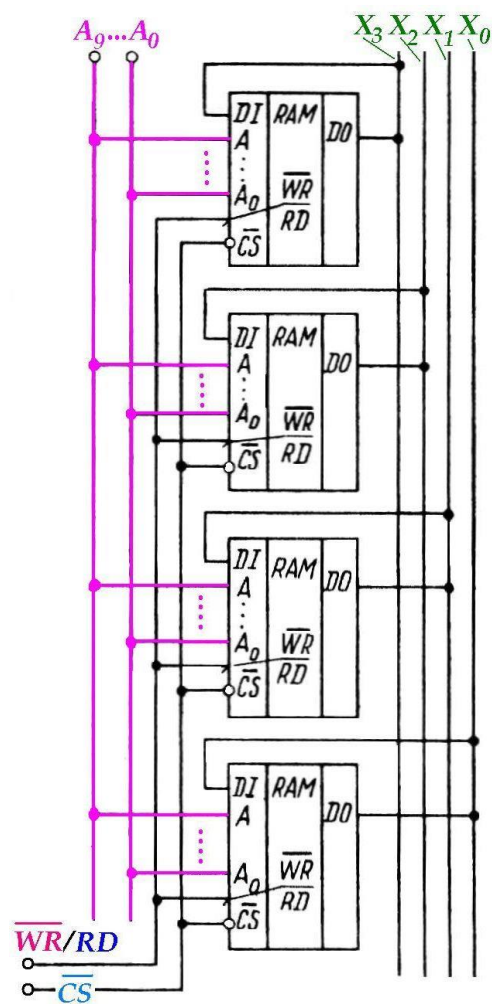
Kahemõõtmeline

Mälumahu suurendamine

- Mida võib ette võtta:
 - suurendada järkude arvu,
 - suurendada säilitatavate sõnade arvu.
- Järkude arvu suurendamine → aadressiini paralleelne lülitus

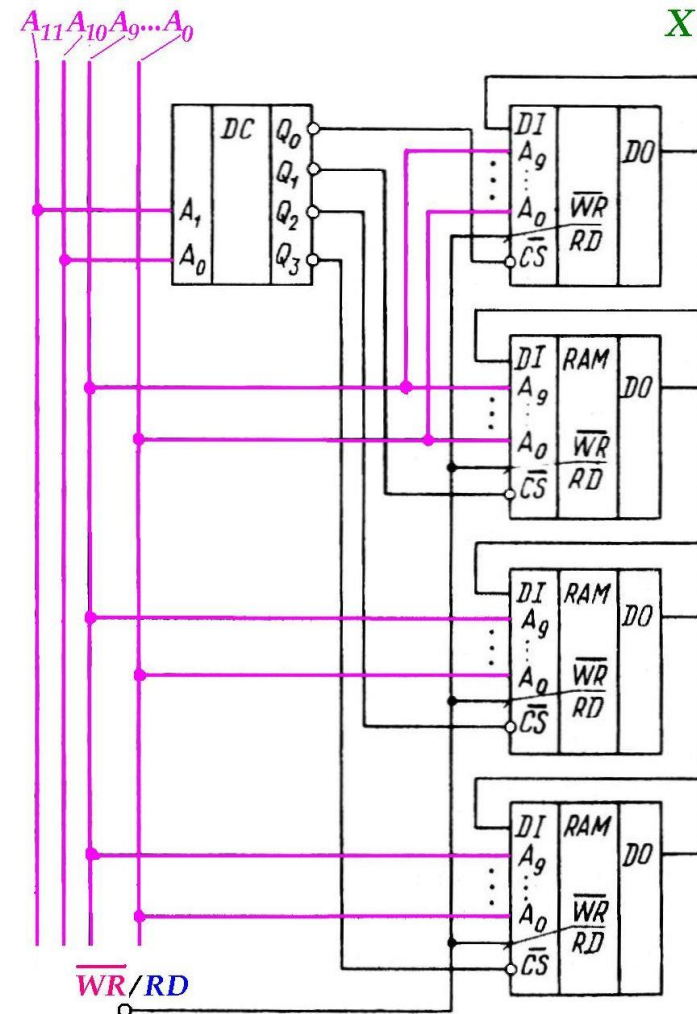
Aadressi sõna antakse kõigile MS korraga. D0 väljundil iga MS antaval aadressil loetakse maha 1 bit infot.

• Skeem 4 järgulise sõna jaoks →



Mälumahu suurendamine

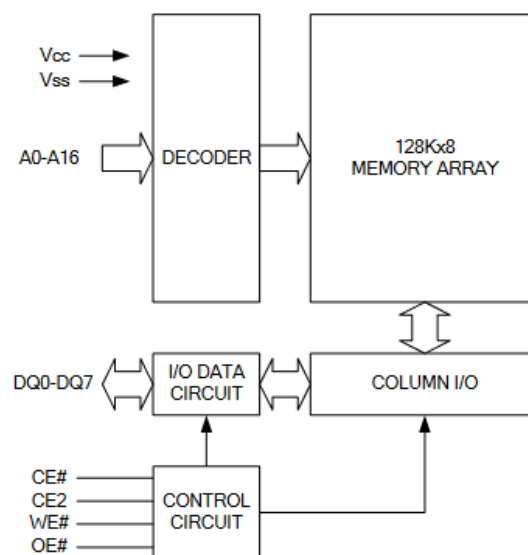
- Säilitatavate sõnade arvu suurendamine (aadressimahu suurendamine)
- Tuleb kasutada lisadekoorderid (A11, A10). Vastavalt sellele koodile dekooder lülitab vajaliku mälu. Iga mälu maht on $1K \cdot 1$. Kokku on siis antaval juhul $4K \cdot 1$ ja 12 – järguline aadressi sõna. Iga mälu juhitakse aadressi koodiga A9...A0.
- Kasutusel on mõlemad variandid (ka samaaegselt)



Tegelikud mälumikroskeemid (1)

- **AS6C1008** – Paralleelne 128kb SRAM (seeria toodetakse 8k-1024k)
- <https://www.tme.eu/Document/09fe96c7f3b090cdc0b23ae42768bc92/AS6C1008.pdf>
- Erinevate rakenduste jaoks erinevad korpused

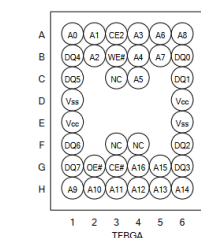
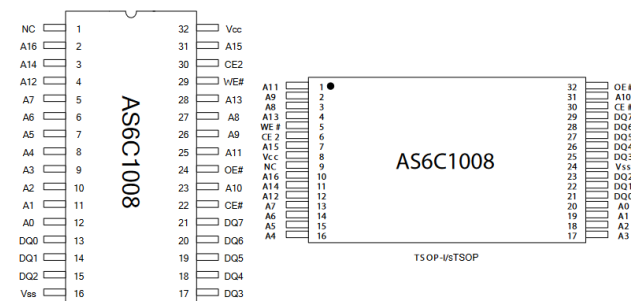
FUNCTIONAL BLOCK DIAGRAM



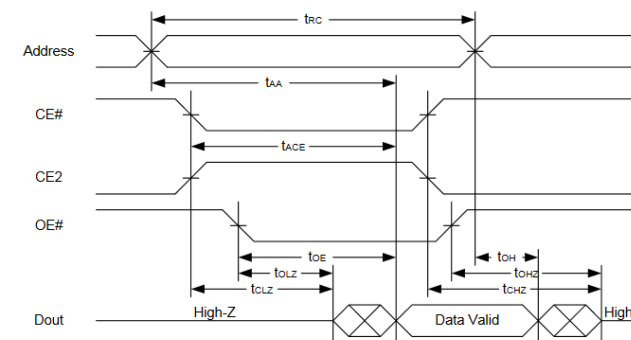
PIN DESCRIPTION

SYMBOL	DESCRIPTION
A0 - A16	Address Inputs
DQ0 - DQ7	Data Inputs/Outputs
CE#, CE2	Chip Enable Inputs
WE#	Write Enable Input
OE#	Output Enable Input
Vcc	Power Supply
Vss	Ground
NC	No Connection

PIN CONFIGURATION



READ CYCLE 2 (CE# and CE2 and OE# Controlled) (1,3,4,5)



Tegelikud mälumikroskeemid (2)

- Jadamälu 23A1024 1Mb mälu
- <https://www.tme.eu/Document/745fc52173836ec906dd19c8a3a18eb9/20005142C.pdf>
- SPI andmesiin (pöördumine läbi protokoll- keeruline ja ka aeglane).
- Sobilik mikrokontrollerite mälu mahu suurendamiseks
- Arduino teek: <https://forum.arduino.cc/t/23lc1024-sram-library/178534>



SOIC/TSSOP/PDIP

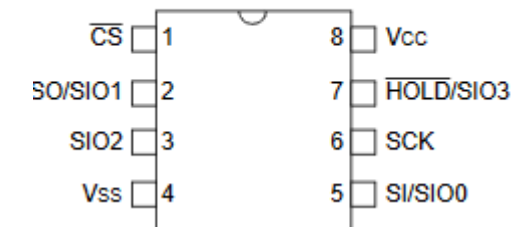
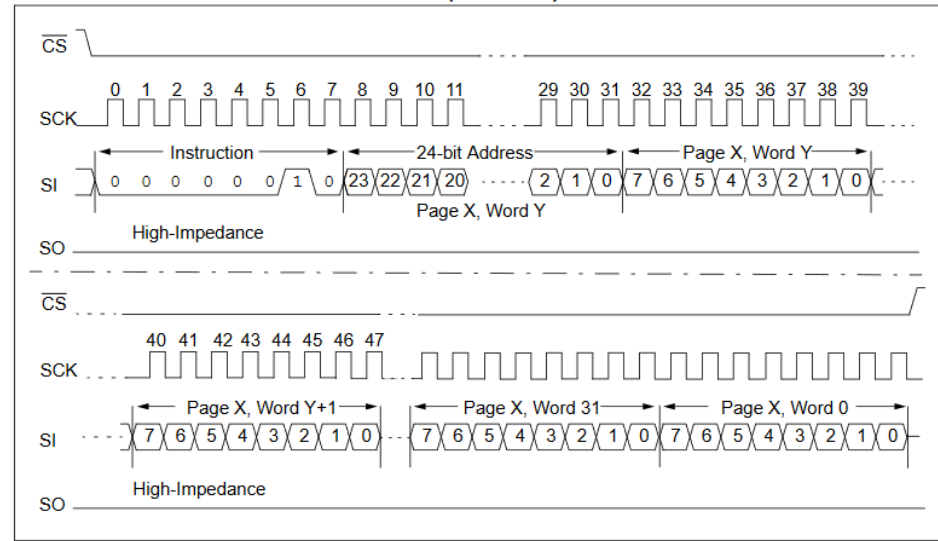


FIGURE 2-4: PAGE WRITE SEQUENCE (SPI MODE)



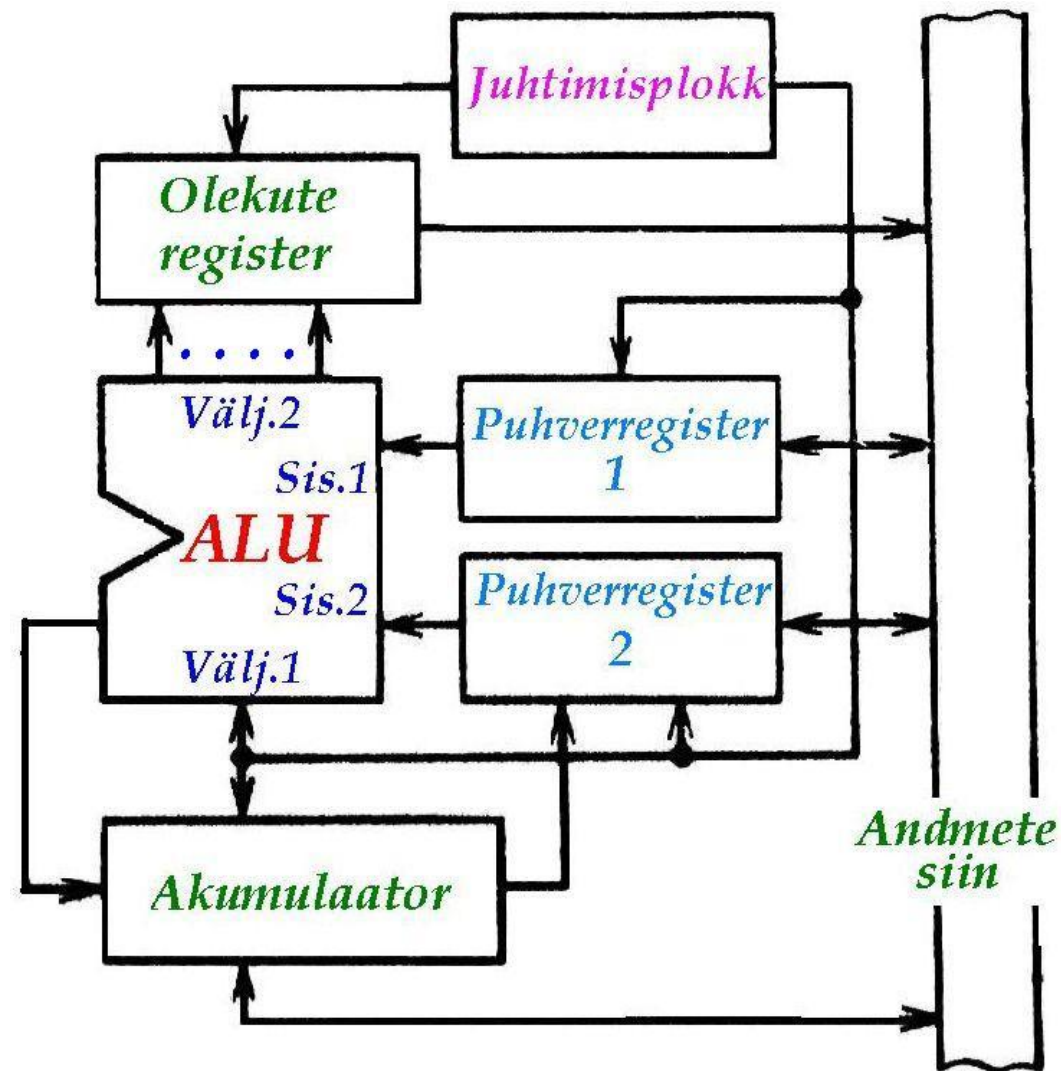
Pin Function Table

Name	Function
CS	Chip Select Input Pin
SO/SIO1	Serial Output/SDI/SQI Pin
SIO2	SQI Pin
Vss	Ground Pin
SI/SIO0	Serial Input/SDI/SQI Pin
SCK	Serial Clock Pin
HOLD/SIO3	Hold/SQI Pin
Vcc	Power Supply Pin

MIKROPROTSESSOR JA MATEMAATIKA

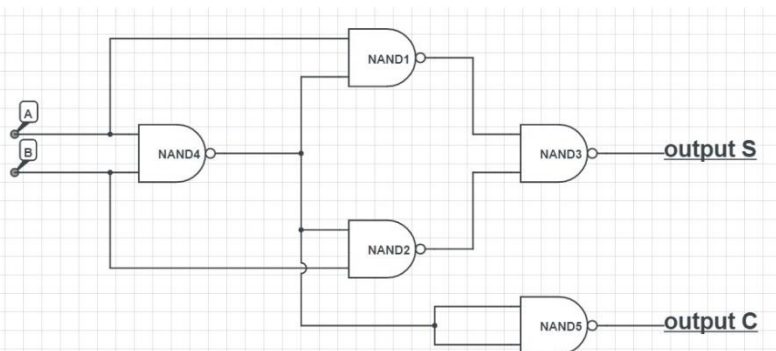
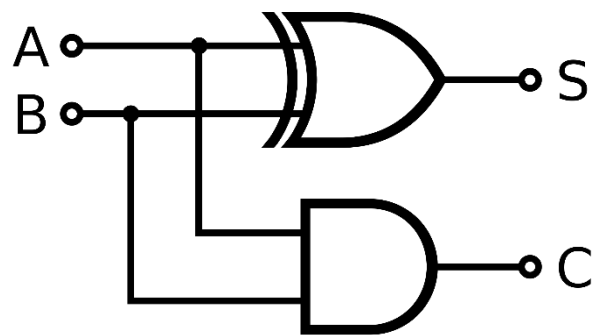
Aritmeetika-loogika seade

- Mikroprotsessori koostisosa
- ALU juurde kuuluvad ka registrid info hoidmiseks.
- ALU ise on kombinatsioonloogikaseade – mälu ei ole.
- Riistvaraliselt on toetatud üldjuhul vähemalt tehted:
 - → aritmeetiline liitmine;
 - → loogiline korrutamine;
 - → inversioon; nihe vasakule, nihe paremale, inkrement, dekrement.

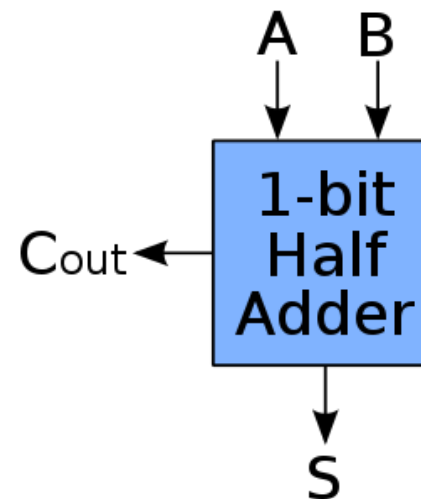


ALU -Liitmine

- Elementaaroperatsioon – välistav VÕI
- Vaja lisada NING element, et saada ülekannet, kuid ei arvestata eelmist ülekannet – Poolsummaator (half adder)



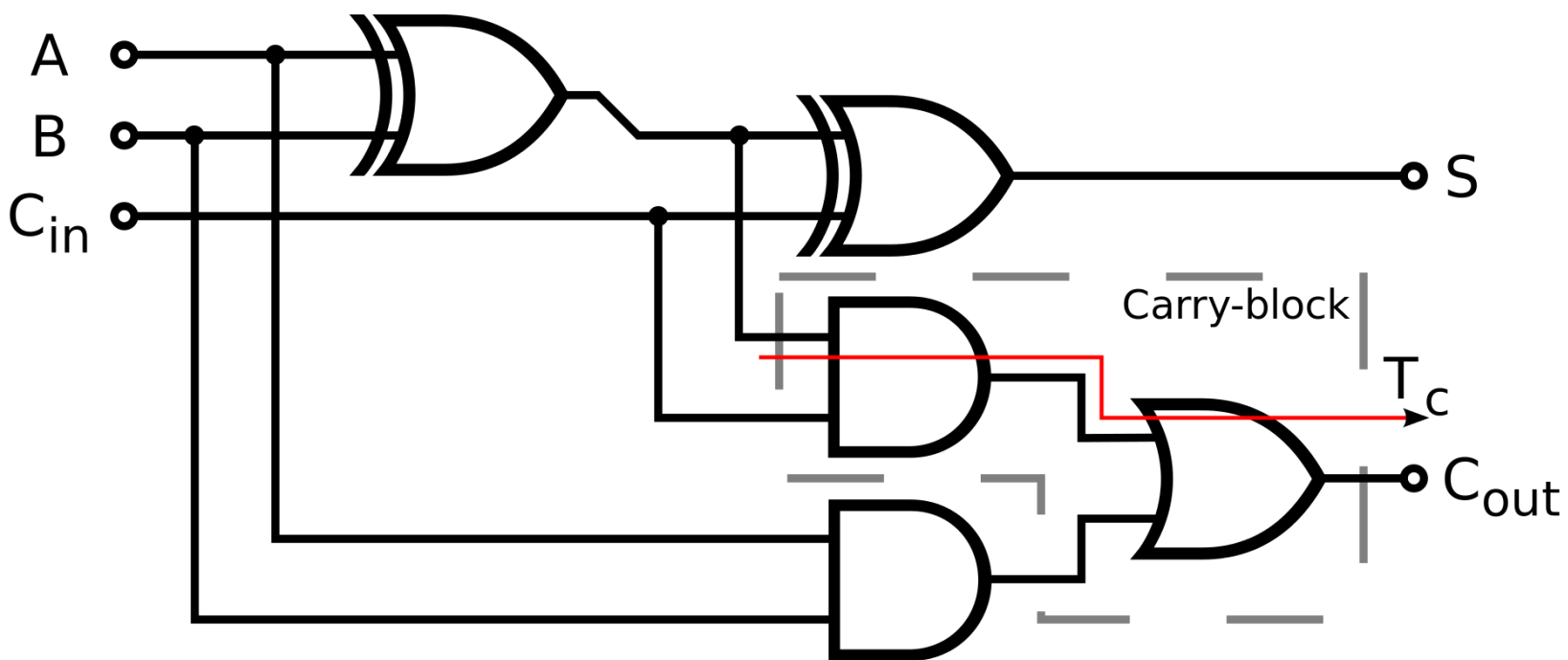
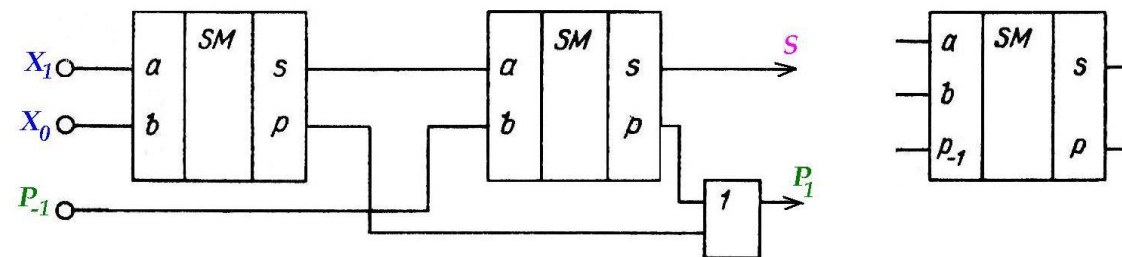
A	B	C _{out}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



Ülekanne järgmisesse järku

ALU - Täissummaator

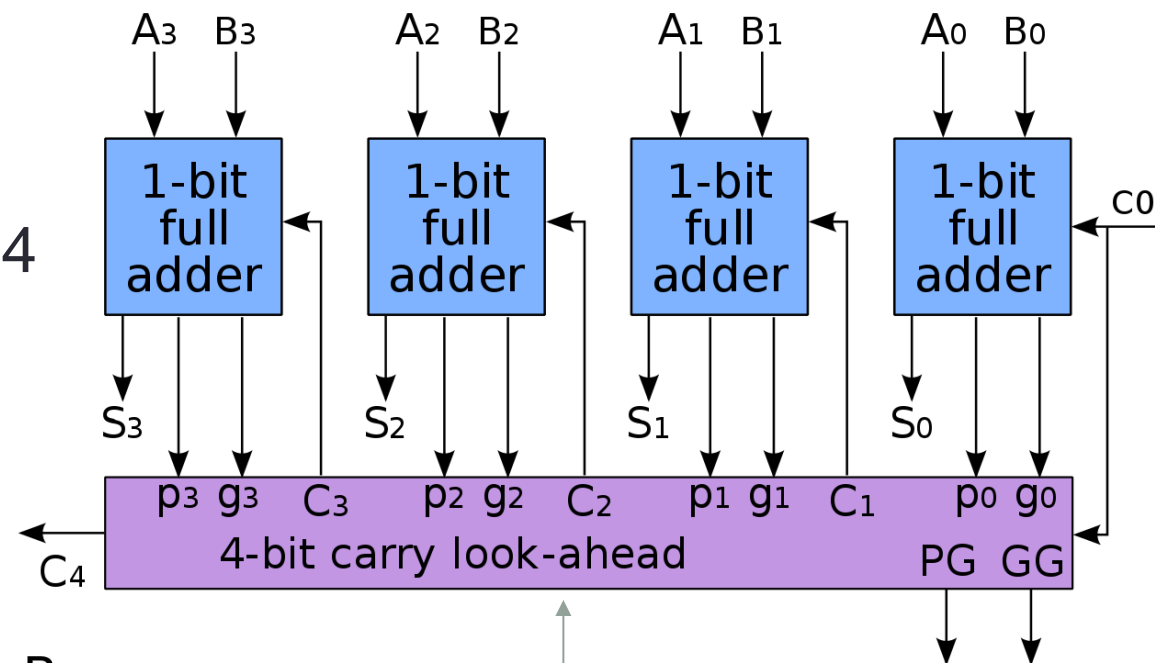
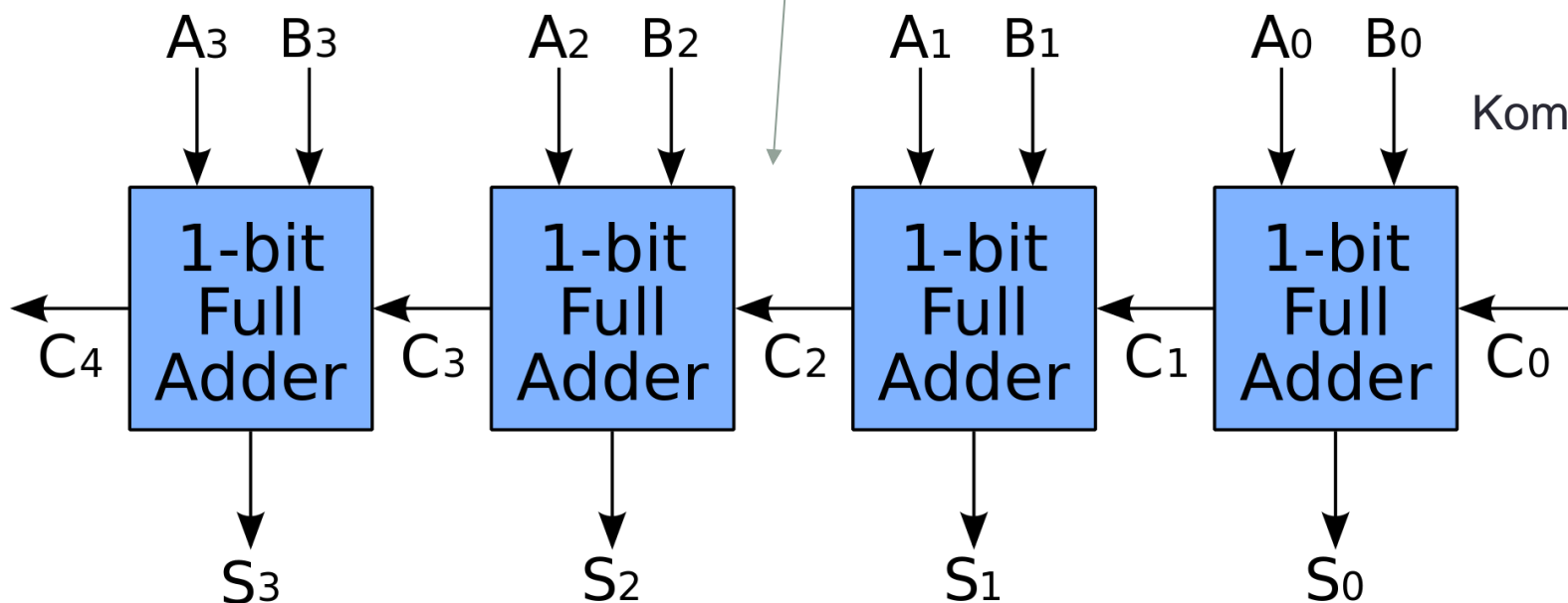
- Koostatakse kahest poolsummaatorist



A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

ALU - summaator

- Järjestikülekanne: Igale järgule eraldi (näide 4 bitiline –kerge on suurendada bittide arvu)
- Aega võtab ülekande liikumine.
- Saab teha ka paralleelülekandega



Kombinatsioonloogika ülekannete arvutamiseks

ALU - lahutamine

- Tegelikult tehakse **liitmine** aga täiendkoodiga !
- Täiendkood: Sama positiivne arv , vahetame nullid ühtedeks ning ühed nullideks ja liidame juurde 1 (sisuliselt teeme nullkoha nihutamise)
- Arv +14(dec)= 00001110b
- Arv -14(dec)= 11110001b+00000001b=11110010b

- Kontroll 00001110b
- + 11110010b
- -----
- (1)00000000b



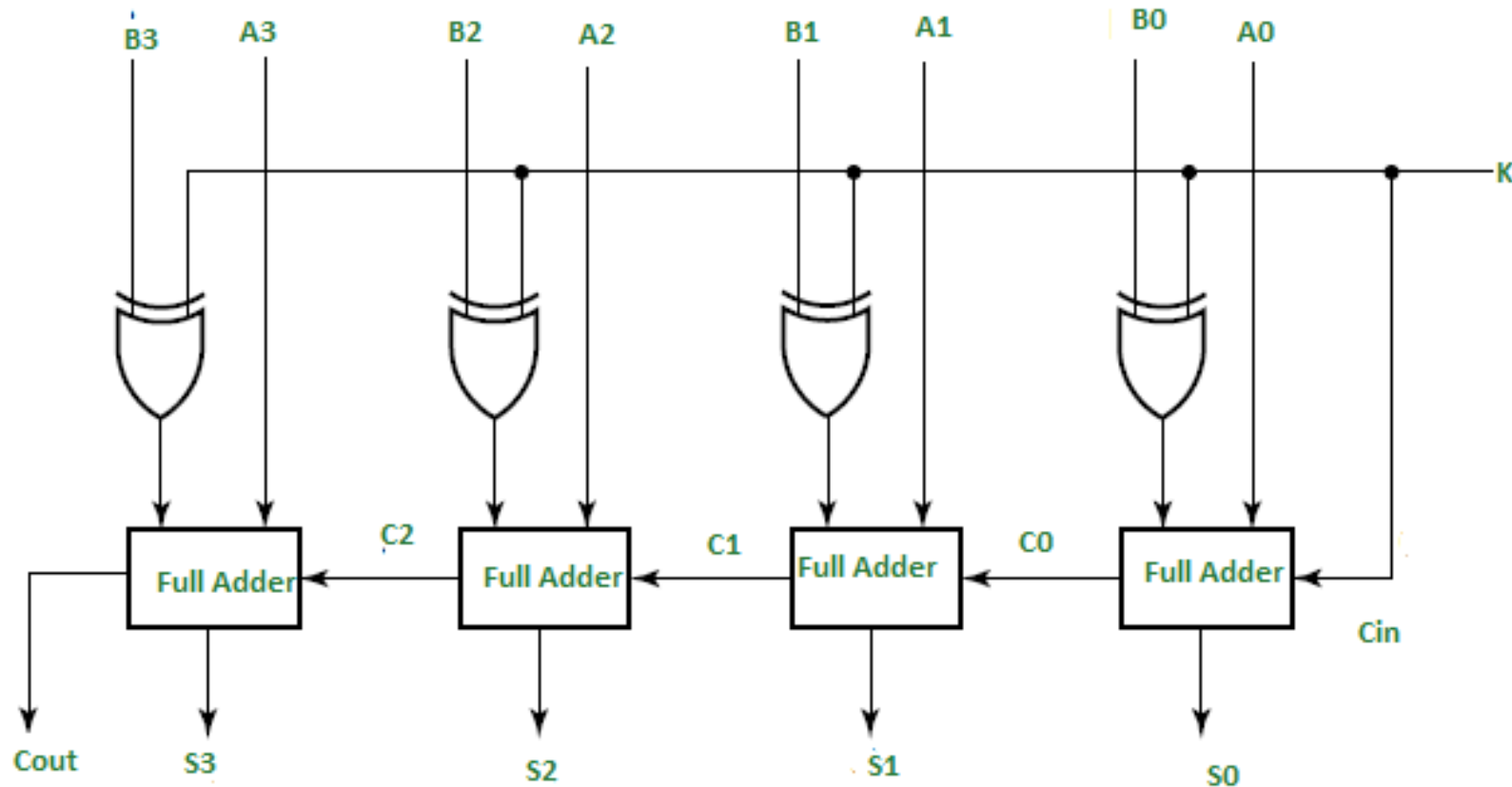
Ülekanne (Carry), kui ei võeta spetsiaalselt midagi ette, läheb see kaduma !

NB !

Arv -14 võib olla aga ka positiivne arv 242 !!! Arvuti ei tee sellel vahet !, vahet peab tegema programmeerija (ja vahest ka kompilaator).

ALU-lahutaja - näide

- Näide riistvaraliselt liitjast-lahutajast



K=0 – liitmine
K=1 – lahutamine

ALU -Korrutamine

- Ühe biti puhul lihtne – NING element
- Korrutamine – erikorrutiste liitmine (sarnaneb paberil „kirjaliku“ korrutamiselega)
- Nii saab hakkama kui bitte on vähe !
- Keerukus kasvab hüppeliselt bittide arvu kasvades.

×	0	1
0	0	0
1	0	1

$$\begin{array}{r}
 \\
 \\
 \\
 \\
 \hline
 1
 \end{array}$$

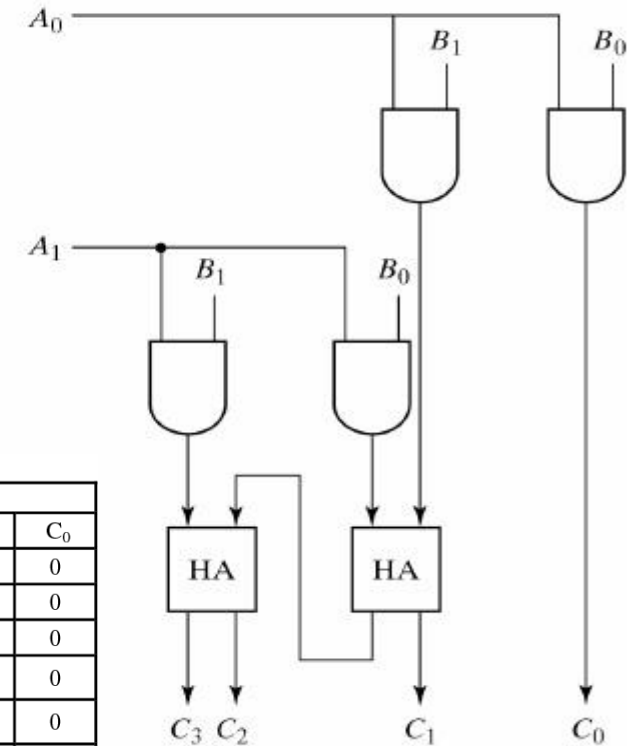
Korrutustabel

Ülejäänu on **liitmine** !

$$\begin{array}{r}
 B_1 B_0 \\
 \hline
 A_1 A_0 \\
 \hline
 A_0 B_1 \\
 A_1 B_0 \\
 \hline
 A_1 B_1 \\
 \hline
 C_3 C_1
 \end{array}$$

Table 1. Truth Table

INPUT A		INPUT B		OUTPUT			
A ₁	A ₀	B ₁	B ₀	C ₃	C ₂	C ₁	C ₀
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1



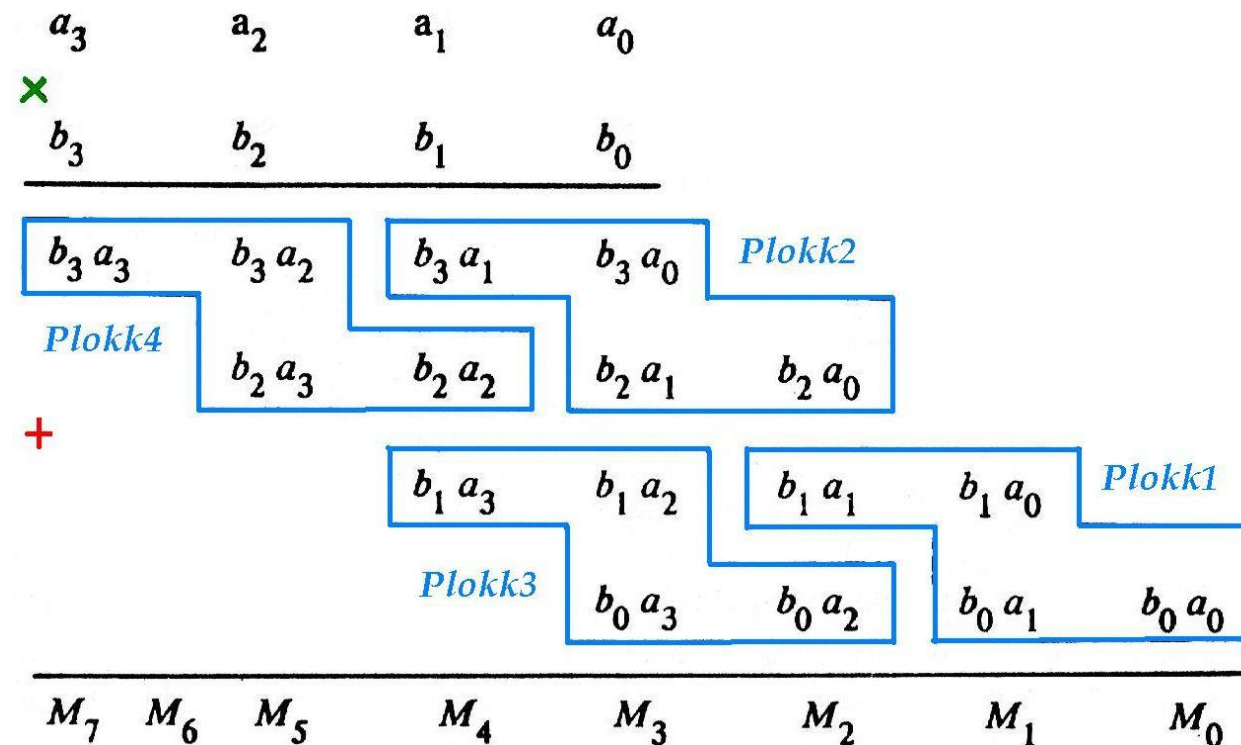
Kahebitine korrutaja

4 osakorrutist
2 liitmist

ALU -Korrutamine

- Reaalne lahendus - Osakorrutisteks jagamine ja summeerimine
- Iga plokk peab tegema korrutamise ja liitmise ja arvestama veel ka ülekannet.
- $M_i = A \cdot B + C + D$

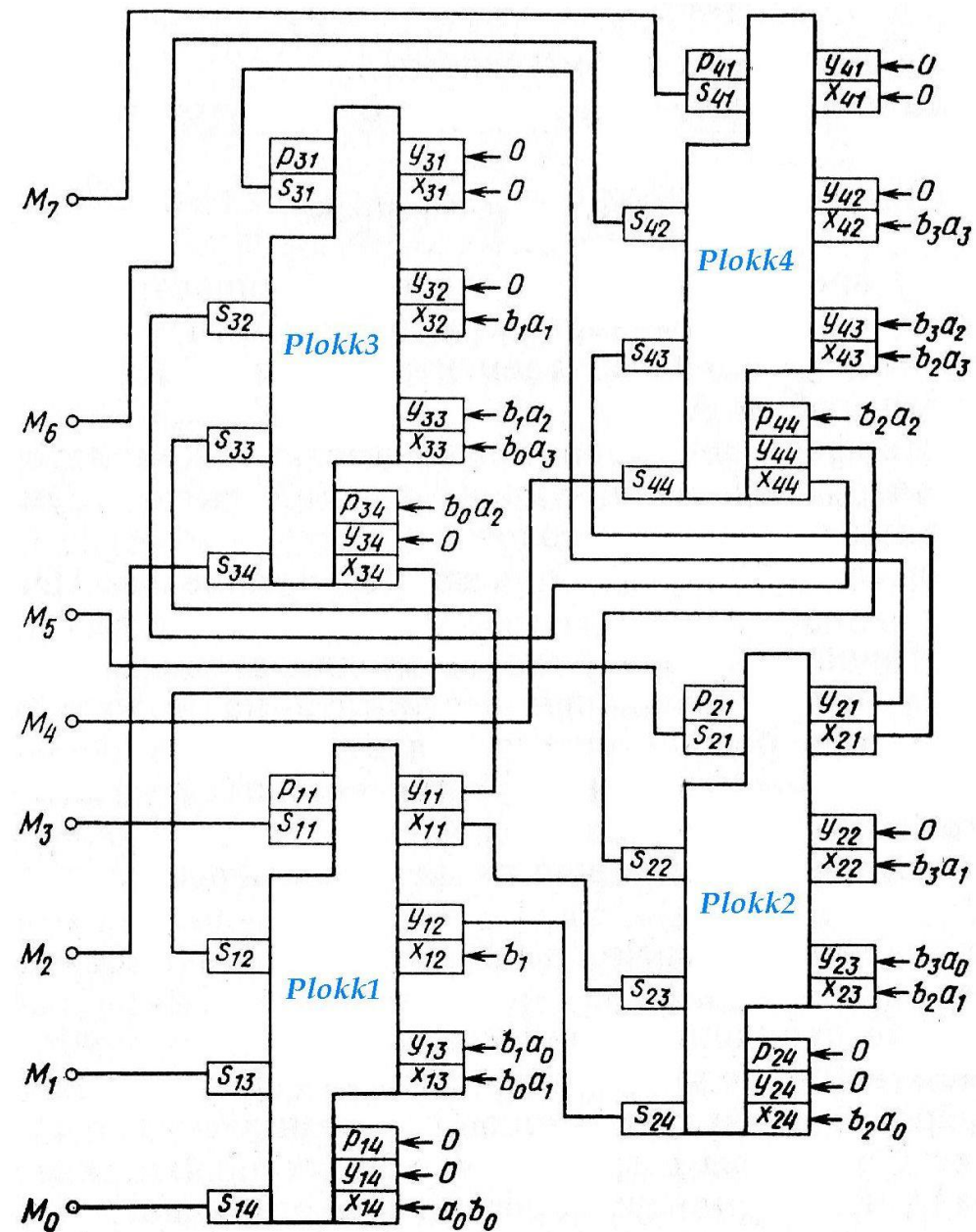
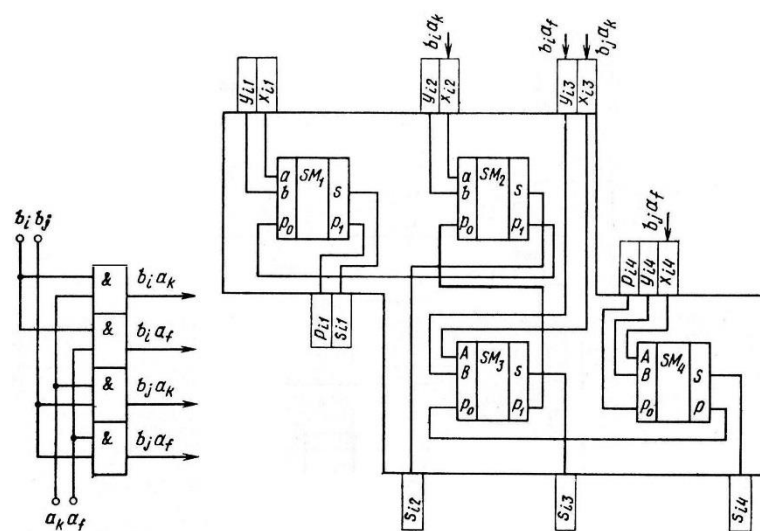
$$\begin{array}{r}
 \\
 \\
 \\
 \\
 \\
 \hline
 1
 \end{array}$$



Kahebitistest osakorrutajatest saadud neljabitine korruti

ALU- korrutamine

- Eelmise slaidi jätk
- Siia juurde kuuluvad veel NING elemendid osakorrutise saamiseks
- Tehakse ka olekutabelitega (nt 8 bitised osakorrutised)



ALU - Jagamine

- Väga lihtne ei ole .
- Sarnaneb kirjaliklu jagamisega
- Nihutame \rightarrow lahutame \rightarrow , kuni jääk on suurem kui jagaja
- Saab teha teisendustabelitega
- Nulliga jagamine on probleem.
- Head näidet slaidi jaoks ei leidnud.

ANDMETE HALDAMINE

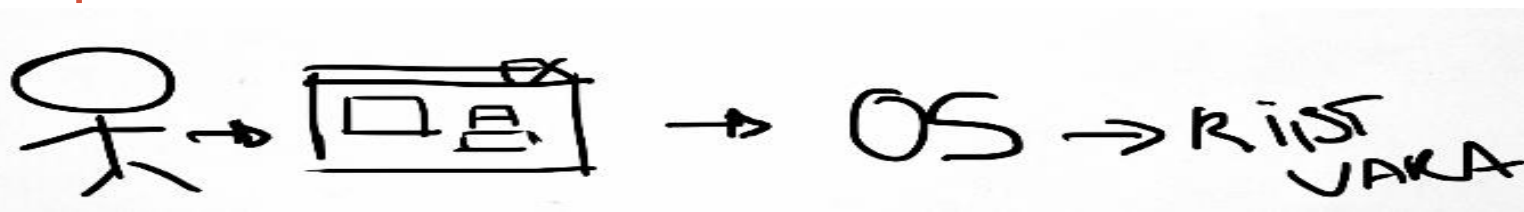
Protsessori, mälu efektiivne kasutamine

Infotehnoloogia igapäevased probleemid

- Andmete hulk on suur (ja kasvab...ja kasvab)
- Andmete optimaalne säilitamine (mälus, kõvakettal, pilves)
- Andmete transportimine ühest seadmest teise.
- Protsesside haldus
- Jadatöötlus vs paralleeltöötlus
- Jne...

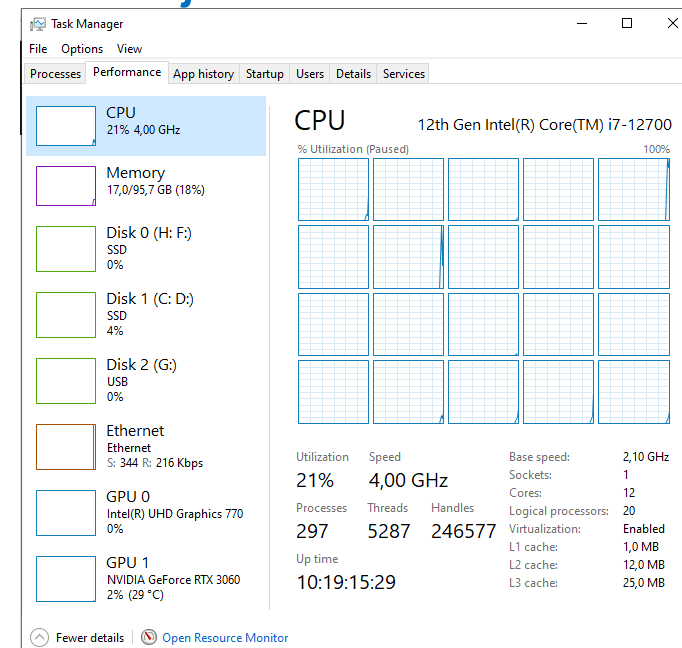
Sellega tegelevad tänapäeval tavakasutaja jaoks nähtamatud protsessi ja mäluhaldurid.

Operatsioonisüsteemi üks osa.



Põhilised komponendid:

- Riistvara
- Operatsioonisüsteem
- Kasutaja programmid
- Kasutaja



Mis on operatsioonisüsteem

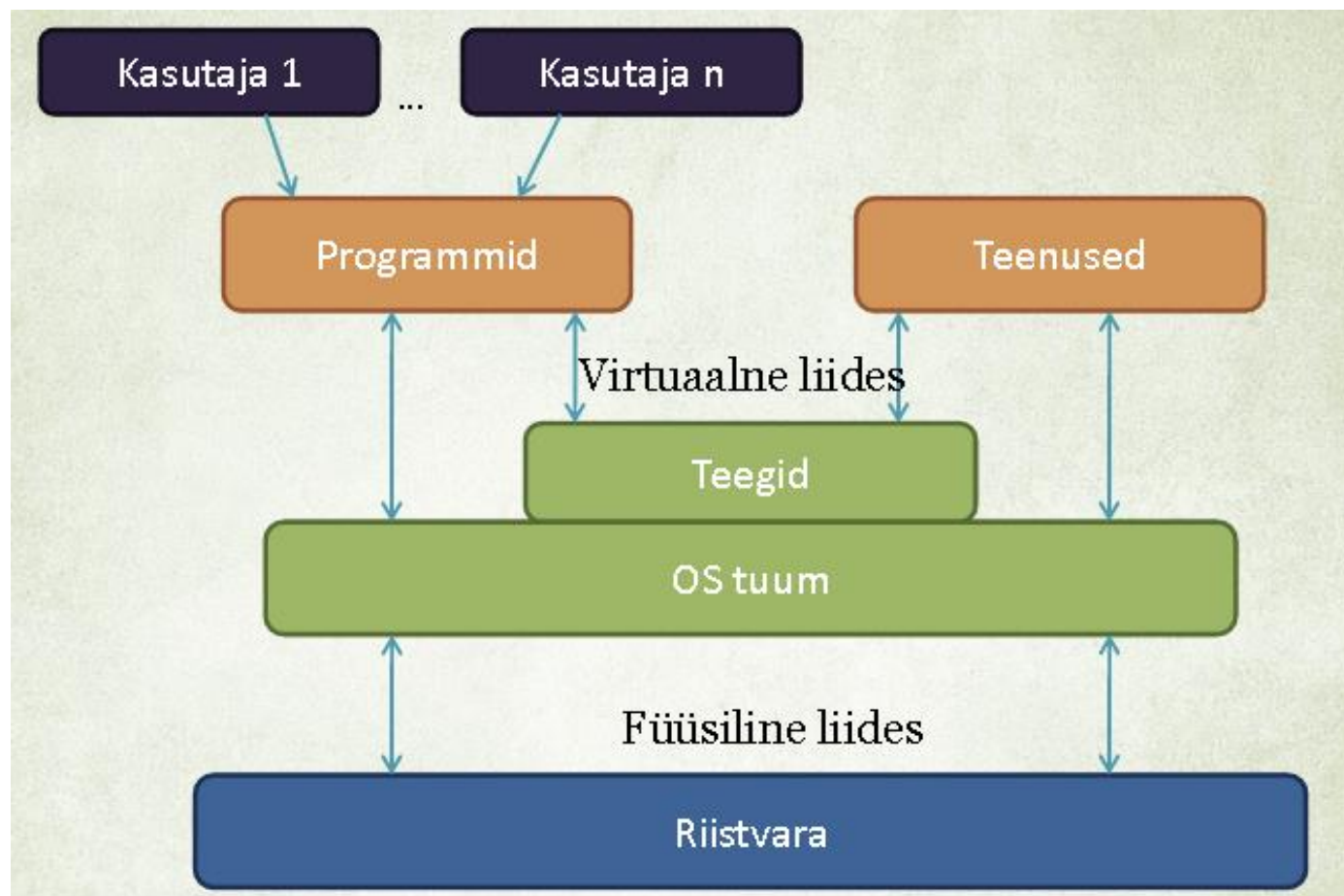
1960-80 ndatel – tarkvara, mis kontrollib riistvara.

Tänapäeval:

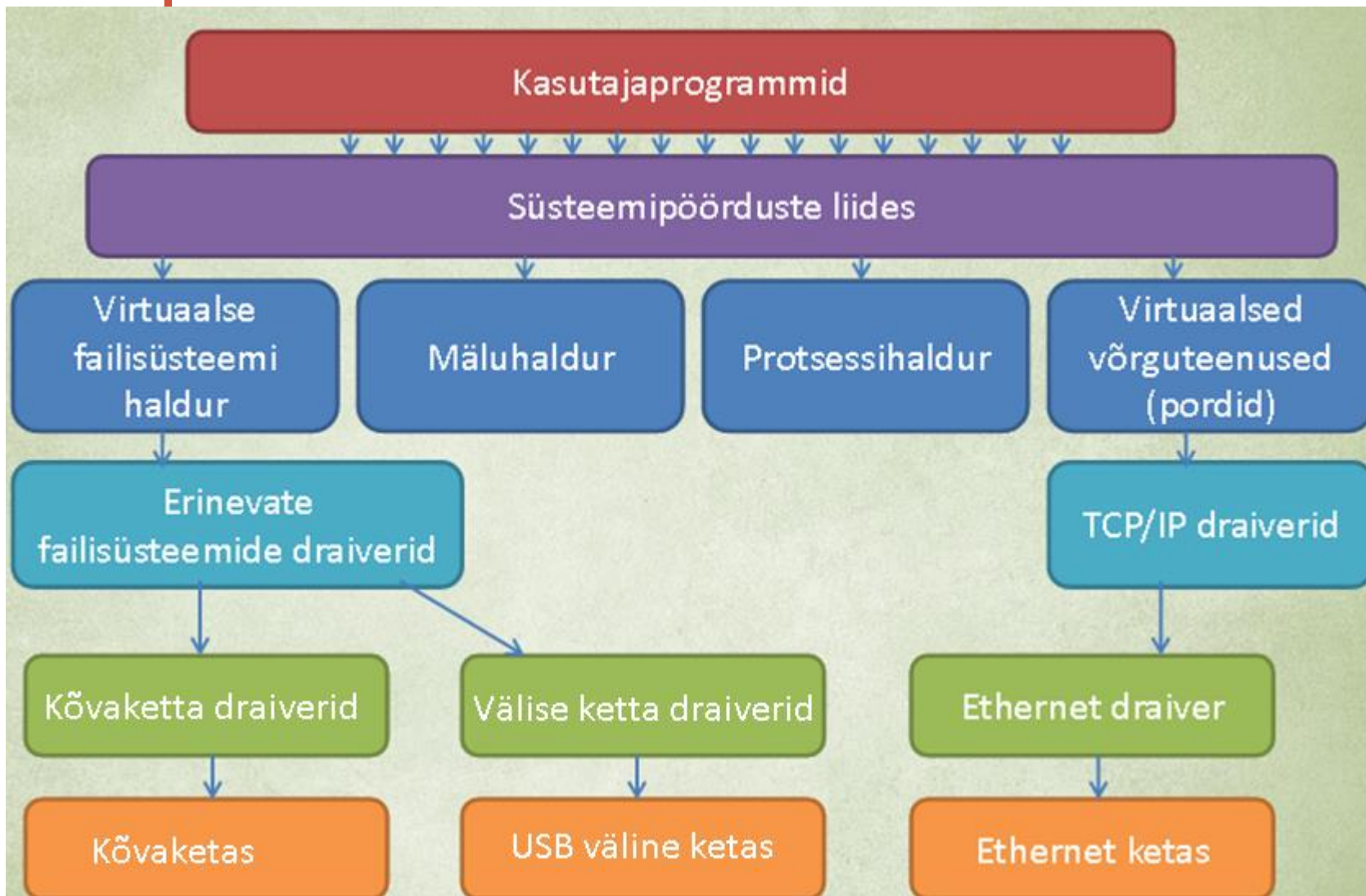
- programm, mis käitub kui vahendaja kasutaja ja riistvara vahel
- Kasutajaprogrammid saavad samal ajal töötada
 - Kui ei ole ettevaatlik, siis võivad mõjutada üksteise tööd ja tulemusi
- Operatsioonisüsteem on kui vahekiht, mis eraldab programme riistavarast ja pakub teenust, mille kaudu saavad programmid koos töötada ohutult ja efektiivselt.

Mis on operatsioonisüsteem

- Varustab kasutajat vahendite komplektiga, mille abil on võimalik programme käivitada
- Tarkvara põhiosa OS-is: kernel(tuum)
- Tuum -Sild rakendusprogrammide ja tegeliku andmete töötlemise vahel riistvara tasemel
- OS on resursside haldaja
- Protsessid, mälu, I/O



Operatsioonisüsteemi tuum



Operatsioonisüsteemi eesmärgid

- **Riistvara varjamine läbi abstraktsioonide**
- Abstraktsioon - tarkvara, mis varjab madalama taseme detaile, OS muudab reaalse, füüsiline (seaded, mälu, aeg) maailma virtuaalseks

Miks on abstraktsioone vaja?

- Kood välisseadmete kontrolliks ei ole standarteetritud -> **draiverid**
- OS annab uue taseme funktsionaalsusi läbi abstraktsioonide. (nt. programmid tegelevad failidega mitte kettaga)
- OS muudab riistvara mitmeks virtuaalseks masinaks.
- Iga protsess näeb riistvara läbi abstraktsiooni
- OS saab lisada turvalisust läbi abstraktsioonide
- Resursside määramine protsessidele
- Pakub sõbralikku kasutajaliidest (Kasutaja „näeb ja tunneb“ os läbi kasutajaliidese)

Opsüsteemi ajaloost

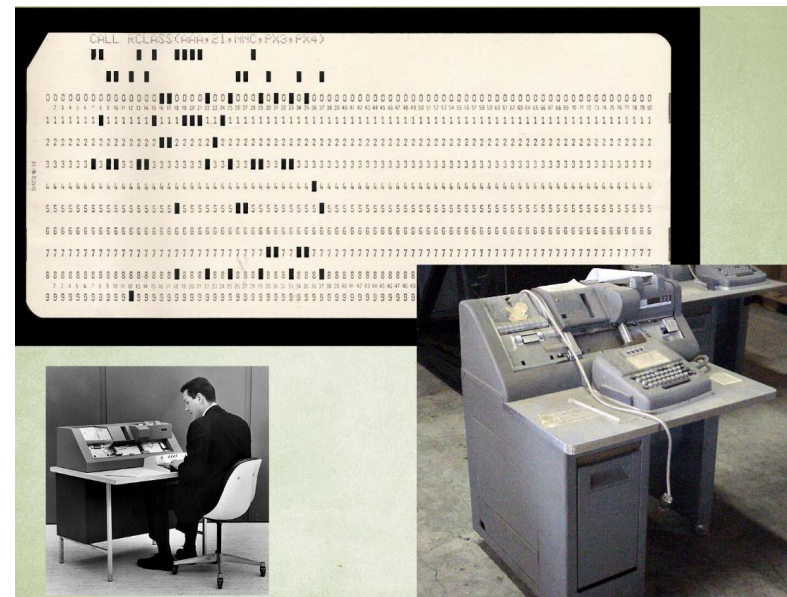
- 1940-1950 . Arvutid ei sisaldanud opsüsteeme , üks programm, masinkoodis (assembleris), primitiivsed
- Esimene OS :General Motors Research Laboratories1950-tel IBM 701 arvutile (1 töö korraga, töötas tunde, päevi ilma kasutaja sekkumiseta)
- Suured masinad
- Käivitati konsoolist
- Sisend :perfokaart või -lint
- Väljund : printer või perforaator
- Protsessori kasutus ebaefektiivne

IBM 701 , esimene kommertslik arvuti (1952)
wikipedia



Opsüsteemi ajaloost

- 1960 – juba efektiivsemad
- Mitu tööd korraga (üks protsess kasutab protsessorit, teine I/O)
- Seadmed – perfokaardid, magnetlindid, magnetkettad, **tänapäevase väljanägemisega kuvar ja klaviatuur.**
- **Optimeerimine** tööde vahel, draiverid



Mälupilt →



1970 - 1990

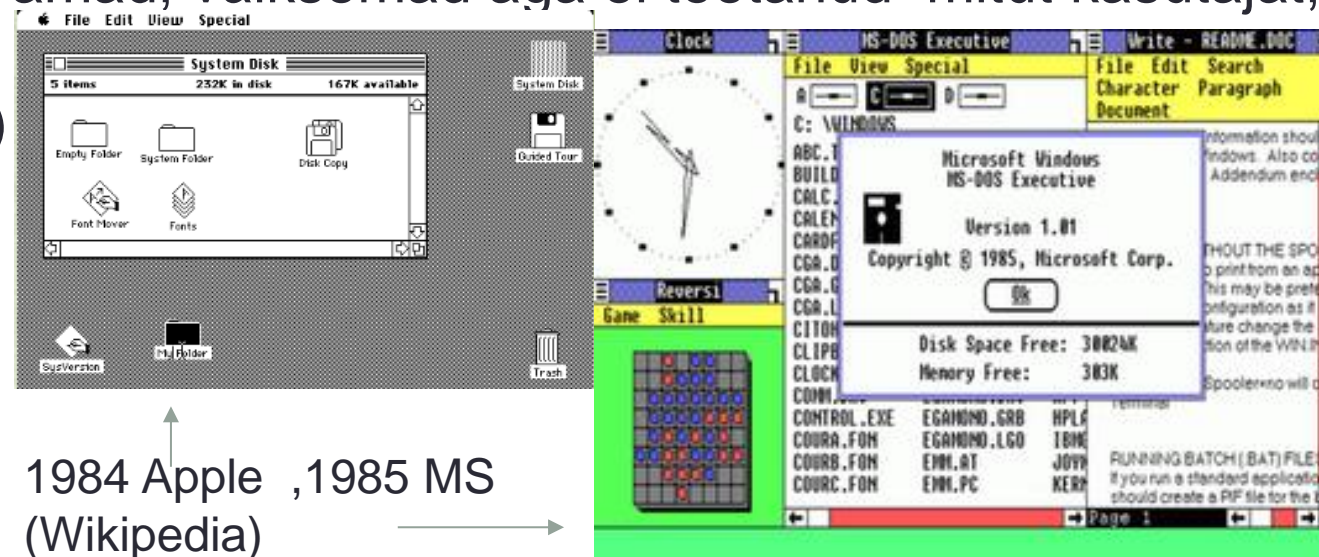
APPLE 2 (Wikipedia).



- Personaalarvutite kiire areng → tänaseni.
- Multiprogrammed süsteemid –“portsjonites” töötlemine
- Ajajaotus - reaalaajas töötavad rakendused
- Personaalarvutite tekkimine tänu mikroprotsessorite arengule– [Apple II\(1977\)](#)
- Arvutivõrkude kiire areng
- **Esimene PC – IBM PC (1981)** Universaalsed arvutid , tekkisid „valmis tarkvarapaketid“
- **8 bitised koduarvutid (nt ZX Spectrum)** Odavamad, väiksemad aqa ei toetanud mitut kasutajat, multiprogrammsust, failisüsteemi kaitset.
- Tekkis GUI (graafiline kasutajaliides,)1984)

Opsüsteemidel tekkis kaks arengusuunda

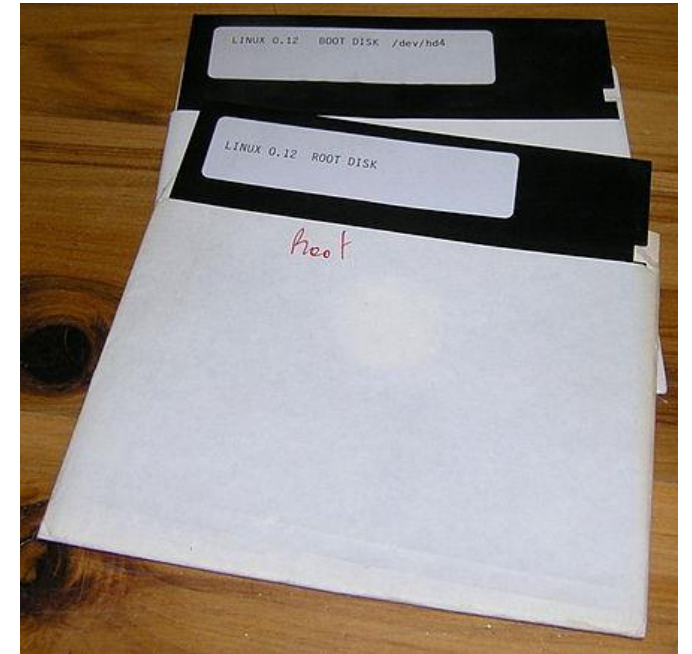
- Unix
- Microsoft (DOS, 80ndatel Windows)



1984 Apple ,1985 MS
(Wikipedia)

1990 – 2000.....

- Kasutajasõbralikus
- **“Plug-and-play”** – Võimalus lisada/eemaldada riistvara komponente ilma OS-i ümber konfigureerimata
- Kasutaja profiilid – Isikupärastamine – paroolid
- **Linux (1991)**, Open source , GPL
- **WWW (1993)**
- **Suureneb andmete turvalisuse probleem**
- Vahevara (middleware)
- Veebiteenused, Online kaubandus
- Järjestikune programmeerimine -> Paralleelne programmeerimine
- Äri ja kodu OS ühendamine (näiteks Windows XP ja edasi) – varem olid eraldi seeriad.
- Hajussüsteemid („arvuti“ moodustub paljudest arvutitest)



Linux algvariandi kettad (wikipedia)

2000 - tänapäev ja tulevik

- ManusOS (embedded) . [Nutiseadmed](#)
- Lihtsus, kasutajasõbralikus.
- Vähe(m) mälu, kood lihtsam
- [Kiirus - vs aku eluiga](#), puutekraan, väike ekraan.
- Kasutusel sageli vaid „terminalina“ Enamus arvutusi toimub serveris
- OS internetis – arvuti vaid terminal (Chrome OS, mängukonsoolid),
- Missioonikriitilised (nt lennujuhtimissüsteemid)
- Ärikriitilised (suured veebiteenused, serverid)
- [Vajalik väga kiire võrguühendus](#)
- Tehisintellekt (Open AI nt)
- Andmekorilus (suured andmed, big data, data mining)
- Pilvesüsteemid, [virtuaalmasinad \(tarkvara abstraktsioon – juhib riistvara, mida tegelikult ei ole , palju OS korraga samas masinas\)](#)
- Muide, [isc.ttu.ee](#) jookseb virtuaalmasinas.



STEAM mängukonsool
Pilt- [amazon.de](#)

OS-i ülesanded

- Protsesside haldamine(otsustab millal ja milline protsess kui palju CPU-d saab)
- Põhimälu haldamine(millal ja kuidas mälu antakse)
- Failide haldamine
- I/O haldamine
- Sekundaarmälu haldamine
- Võrgu haldamine
- Kaitsesüsteem
- Käsuinterpretaator

Protsesside haldamine

- Ühest programmist -> mitu protsessi
- Igal oma olek ja erinev käsuloendur
- Süsteem koosneb protsesside komplektist
- Igal ühel “oma” (virtuaalne või reaalne) CPU kasutada

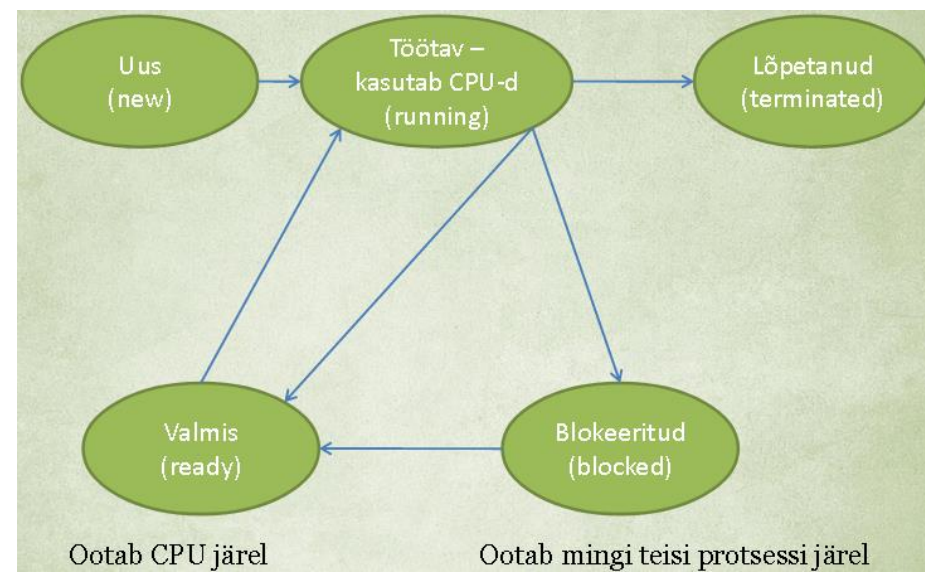
Protsessijuhtimine:

- Protsesside loomine ja kustutamine
- Protsesside peatamine ja taastamine
- Protsesside sünkroniseerimine
- Protsesside omavahelise kommunikatsiooni organiseerimine

Protsside haldamine

Protssori aja andmine

- realselt saab ühel (ka virtuaalsel) protssoril käia üks protss korraga
- Protsside olekud:
 - Jookseb (running) täidetakse protssoris
 - Valmis (ready) –saaks täita kui oleks protssor vaba
 - Blokeeritud (blocked) –ootab mingit kindlat sündmust



Protsesside haldamine

- Kasutaja jooksub programmi -> luuakse protsess -> protsess pannakse readylisti (new)
- Protsess jõuab listi algusesse -> antakse protsessori aega (running)
- Dispetšer (OS-i “komponent”, mis annab CPU readylisti alguses olevale protsessile)
- “Ärkvel olevad” protsessid (need, mille staatus on „ready“ või „running“)
- **Ükski protsess ei tohi protsessorit täielikult endale saada**
 - Katkestuskell -> katkestus
 - Protsessor võetakse ära (ready)

Kui protsess tahab I/O tegevust, siis loovutab vabatahtlikult protsessori (blocked)–
Peab I/O järel ootama

Protsesside haldamine

- Kui süsteem saab otsustada millist protsessi järgmisena täita -> siis peab tal olema mingi strateegia selleks.

Protsessi haldus teatud eesmärgiga nt:

- Max nr protsesse saab joosta teatud aja jooksul
- Min aeg, mida protsess ootama peab
- Ära hoida igavest ootamist
-

Protsesside haldamine

First Come, First Served (FCFS)

Kes ees, see mees.

- Lihtsaim
- Mitteennetav (ei vahetata välja)
- Protsess, mis esimesena nõuab CPU-d, teenindatakse esimesena
- Kõik saavad teenindatud, lühikesed peavad kaua ootama.
- Ei ole kasulik interaktiivsetele protsessidele
- Tänapäeva harva kasutuses

Protsesside haldamine

Shortest Job First (SJF)

- Lühim töö järgmisena
- Ooteaeg väheneb
- Mitteennetav (ei vahetata välja)
- Interaktiivsed protsessid on lühikesed □
- **Peab teadma kaua protsess jookseb -> tihti seda infot pole.**

Protsesside haldamine

Shortest remaining time First (SRTF)

- Lühim järgi jäänud aeg järgmisena
- Ooteaeg väheneb
- Ennetav (vahetatakse välja, kui tuleb lühem kokureeriv protsess)
- Enamasti efektiivsem ,kuid.. Protsess kohe lõpetamas, vahetatakse välja.. Kogu aeg vahetatakse–Ajakulu vahetamiseks on suurem kui olemasoleval lõpuni joosta
- **Peab teadma kaua protsess jookseb -> tihti seda infot pole.**

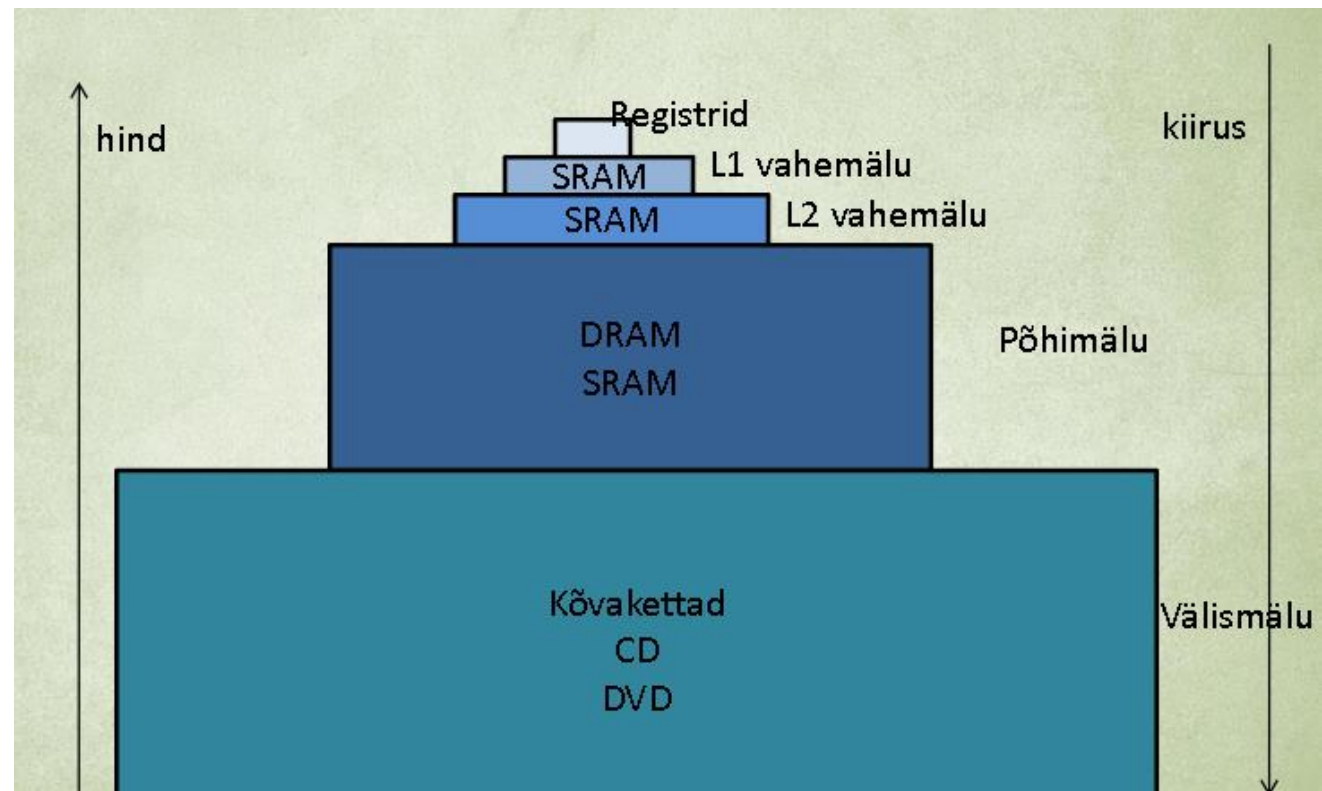
Protsesside haldamine

Round Robin

- Ajajaotussüsteemides
- Protsessori aega jagatakse kordamööda konstantsete taktidena
- Ei ole parim ooteaeg aga parim reaktsiooniaeg
- Tähtis takti valik
- – Kui väike protsesside kogukestvusega, siis on ümberlülituste lisakulu suur
- –Liiga suur takt taandab FCFS algoritmiks

Mälu haldamine

- Üks protsess mälus?
- Kui mitu, kas kõik saavad võrdse osa?
- Kas jagada mälu partitsioonidesse?
- Kas protsesside mälupiirkond peab tulema järjest
- OSi komponent, mis tegeleb süsteemi mälu organiseerimisega ja mälu haldamise strateegiatega
Kuidas olemasolevat mälu protsessidele anda (kui saadaval) suhtleb mälu haldamise riistvaraga.



Mälu haldamine

- Strateegiad –Sissetoomis (fetch) (otsustab millal tuua uus programm või andmed sekundaarsest mälust põhimällu)
- Nõude sissetoomine (demandfetch) - kui andmeid vaja, siis pannakse need alles põhimällu.
- Ennetav sissetoomine (anticipatoryfetch) (üritame ette aimata, mida vaja läheb ja toome need enne põhimällu kui nende poole pöördutakse. Võidakse tuua ka asju, mida ei lähe vaja – kasutu töö)
- Asetamise (placement) - kuhu mälus panna)
- First fit
- Best fit
- Worst fit
- Asendamise (replacement) (kui mälu täis, tuleb otsustada, mida asendada)

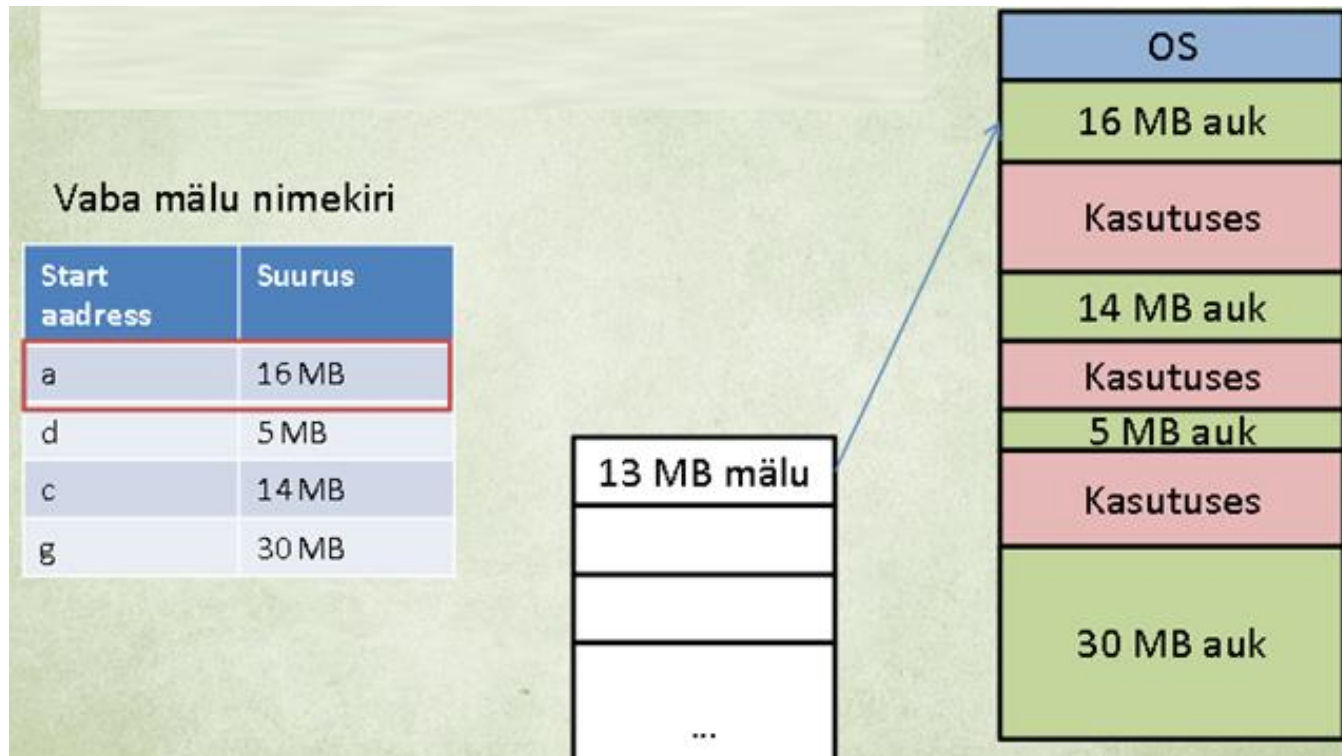
Mälu haldamine

- Mälu fragmenteerimine
- On piisavalt mälu aga seda ei ole võimalik kasutada
- Liigid–Välimine (mälu on, kuid see ei ole pidev)
- Sisemine (kinnistatud mälu on suurem kui tellitud. Vahel ei ole vajalik aga seda ei saa kasutada)
- Väline fragmenteerumine on omane dünaamilise mälujaotusele
- Erinevad strateegiad fragmenteerumise minimeerimiseks.

Mälu haldamine

First Fit (FF)

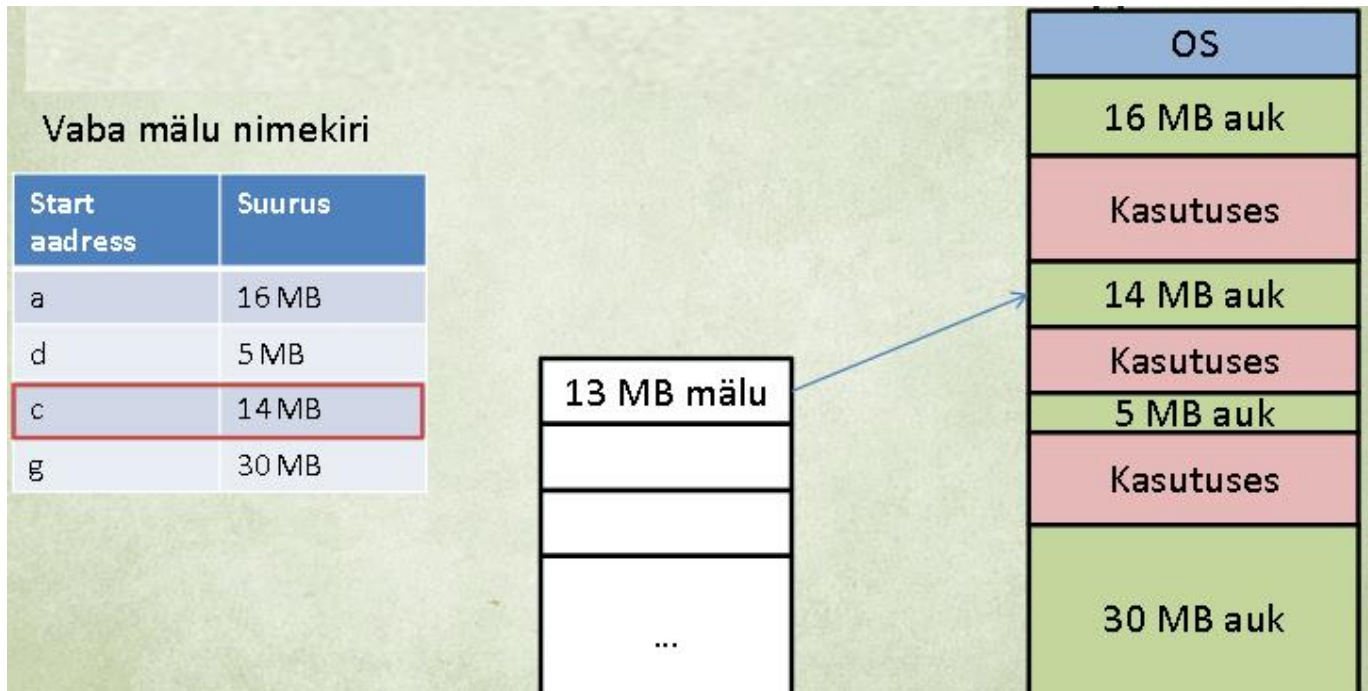
- Töö pannakse põhimõlul esimesse piisavalt suurde vabaolevasse auku. Otsuseid tehakse kiirelt.



Mälu haldamine

Best fit

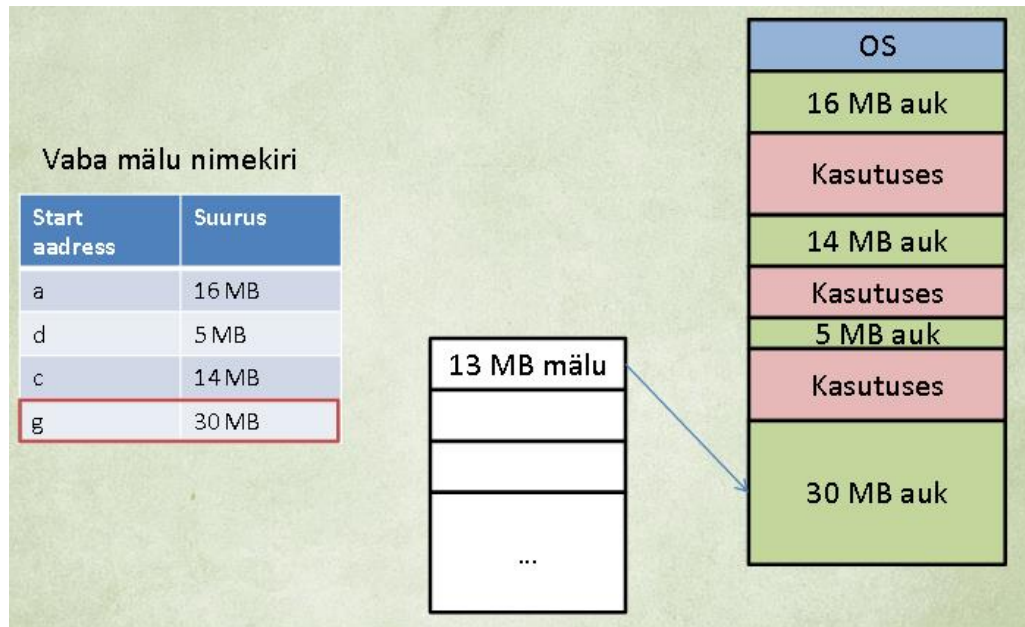
- Süsteem paneb sissetuleva töö peamälu auku, kuhu see kõige väiksemate kadudega mahub –ehk üritab saavutada parimat pakkimist. **Võtab aega -> parima augu leidmine**



Mälu haldamine

Worst Fit

- Paneb töö põhimälus kohta, kus kõige suuremad kaod (ehk kõige rohkem ruumi jääb üle)
- Kõlab imelikult, eks ju? Aga -> paneme töö suurde auku. See, mis alles jääb on tõenäoliselt ka suur ja saame sinna järgmise tellitava ploki panna.



- FF ja BF paremad kui WF.
- FF-võib põhjustada välimise fragmenteerumise

Viimane slaid

- 79 päeva pärast on LÕPP (sellel semestril)
- Kui on küsimusi, muresid, küsige ,pole vaja karta.
- **Ära loobu pärast esimest ebaõnnestumist !**
- Küsija suu pihta ma ei löö !
- Ära pinguta üle ! Lahendused võivad olla lihtsamad kui alguses paistab.

- Minuga võimalus kohtuda (üle)järgmine sügis : ATR0110 – Analoo ja Digitaaltehnik , Mehhatroonika suuna valikaine – neile, kes huvituvad praktilisest elektroonikast (aine pole veel uuel kujul lõpuni valmis).
- Aitäh !
- **Otsin sügiseks assistenti !**