

FKEF.02.143

## Arvuti arhitektuur

Computer Architecture and Organization



dotsent  
Toomas Plank

©Toomas Plank, 2008

---

---

---

---

---

---

---

---

FKEF.02.143

## Arvuti tööpõhimõte



1. loeng,  
18. veebruar 2008

©Toomas Plank, 2008

---

---

---

---

---

---

---


---

### Sissejuhatus

Selles loengus tutvume mitmete arvuti riistvara ja tarkvara kontseptsioonidega, samuti terminoloogiaga

- Arvuti põhimõtteskeem
- Masinkäsud
- Süsteemitarkvara
- Arvuti jõudlus
- Ajalooline taust

Detailsema käsitluseni jõuame järgmistes loengutes

Arvuti arhitektuur FKEF.02.143

---

---

---

---

---

---

---

---

## Arvuti – mis see on?

- Arvuti – kiire elektrooniline arvutusmasin,
  - mis võtab vastu digitaalset infot,
  - töötleb seda vastavalt temas salvestatud käskude loeteluleja
  - väljastab tulemuse oma väljundisse.
- Käskude loetelu – s.o arvutiprogramm
- Sisemine salvestus – s.o arvuti mälu

Arvuti arhitektuur FKEE.02.143



---

---

---

---

---

---

---

---

## Arvutitüübid

- Personaalarvuti
  - lauaarvuti
  - sülearvuti
- Tööjaam
- Server
- Superarvuti

Arvuti arhitektuur FKEE.02.143



---

---

---

---

---

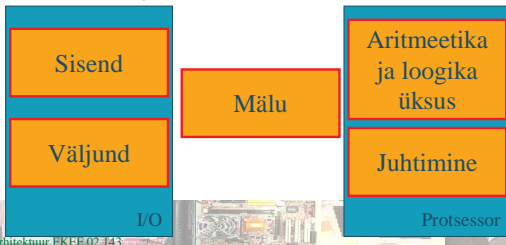
---

---

---

## Funktsionaalne jaotus

- Sisend ingl *Input*
- Mälu ingl *Memory*
- Aritmeetika ja loogika üksus
- Väljund ingl *Output*
- Juhtimine ingl *Control*



Arvuti arhitektuur FKEE.02.143

---

---

---

---

---

---

---

---

## Informatsioon (1)

- Käsud
  - juhivad info liikumist arvuti sees
  - juhivad info liikumist arvuti sisemuse ja sisend-väljund seadmete vahel
  - määravad täitmist ootavad aritmeetika- ja loogikatehted
- Andmed
  - numbrid
  - kodeeritud tähed
- Käskude loetelu arvuti mälus moodustab programmi.
- Protsessor kutsub käsud üksteise järel mälust välja ja teeb, mida käsud tal teha paluvad
- Arvuti tegevust kontrollib suurema osa ajast salvestatud programm(id), v.a siis, kui kasutaja või sisend-väljundseade soovib miskit muud vahepeal teha.

Arvuti arhitektuur FKEF.02.143



---

---

---

---

---

---

---

---

## Informatsioon (2)

- Käsud
  - juhivad info liikumist arvuti sees
  - juhivad info liikumist arvuti sisemuse ja sisend-väljund seadmete vahel
  - määravad täitmist ootavad aritmeetika- ja loogikatehted
- Andmed
  - numbrid
  - kodeeritud tähed
- Andmed on käskudele sisendparameetriteks
- Teatud tingimustel on kogu programm ise andmeteks
  - algkoodi kompileerimine tööprogrammiks

Arvuti arhitektuur FKEF.02.143



---

---

---

---

---

---

---

---

## Informatsioon (3)

- Info peab olema sobivalt kodeeritud
- Loogikaahelatel on kaks võimalikku olekut:
  - sees
  - väljas
- Kõik numbrid ja tähed on kodeeritud binaarsete numbrite – bittide – jadana.
- Igal bitil on kaks võimalikku väärtust
  - 0
  - 1
- Kümnendnumbri salvestamiseks vajame 4 bitti
- Tähtede salvestamiseks
  - 7-bitine ASCII (*American Standard Code for Information Interchange*)
  - 8-bitine EBCDIC (*Extended Binary-Coded Decimal Interchange Code*)

Arvuti arhitektuur FKEF.02.143



---

---

---

---

---

---

---

---

### Sisend

- Klaviatuur
- Hiir
- Joystick
- Trackball
- Mikrofon
- Veebikaamera
- Skanner
- Monitor (graafilise sisestusliidese osana)




Arvuti arhitektuur FKFE.02.143

---

---

---

---

---

---

---

---


### Mälu (1)

- Mälu ülesandeks on andmete ja programmide säilitamine

Primaarne ladustamine (*Primary storage*)

- Kiired mälad
- Pooljuhtelemendid, igaüks võimeline hoidma 1 biti infot
- Opereeritakse grupi kaupa –  $n$  bitti saab lugeda/kirjutada ühe käsuga
- Õige koha ülesleidmiseks kasutatakse adresseerimist

- *Random-access memory* (RAM)
  - Mälu pöördumisaeg (*memory access time*)
  - Vahemälu
  - Põhimälu



Arvuti arhitektuur FKFE.02.143

---

---

---

---

---

---

---

---

### Mälu (2)

- Sekundaarne ladustamine (*Secondary storage*)
  - Magnetkettad
  - Magnetlindid
  - Optilised kettad



Arvuti arhitektuur FKFE.02.143

---

---

---

---

---

---

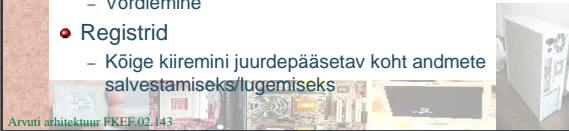
---

---

## Aritmeetika-loogika seade

- Ingl *Arithmetic and logic unit*
- Lühend ALU
- Koht, kus tehakse aritmeetika ja loogika tehteid
  - Liitmine
  - Lahutamine
  - Korrutamine
  - Jagamine
  - Võrdlemine
- Registrid
  - Kõige kiiremini juurdepääsetav koht andmete salvestamiseks/lugemiseks

Arvuti arhitektuur FKEE.02.143



---

---

---

---

---

---

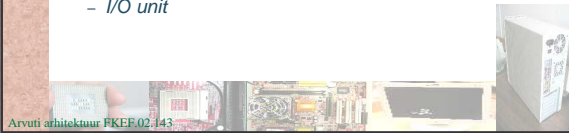
---

---

## Väljund

- Ülesandeks teha arvutis rehendatud tulemus kättesaadavaks kasutajale
- Printerid
- Monitorid
- Duaalse rolli tõttu võetakse sisendseadmed ja väljundseadmed sageli kokku ühtse nimetaja alla
  - sisend/väljund seade
  - I/O unit

Arvuti arhitektuur FKEE.02.143



---

---

---

---

---

---

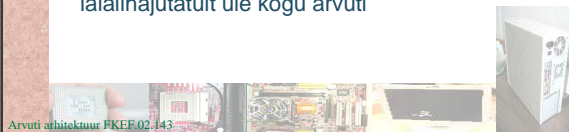
---

---

## Juhtimine

- Ingl *Control unit*
- Ülesandeks koordineerida mälu, ALU, sisendi ja väljundi tegevust
  - saadab teistele osapooltele juhtimissignaale ja
  - teab mis seisus teised osapooled parasjagu on
  - genereerib ajastussignaali (*timing signal*)
- Füüsiliselt paikneb see "seade" laialihajutatult üle kogu arvuti

Arvuti arhitektuur FKEE.02.143



---

---

---

---

---

---

---

---

## Arvuti töö kokkuvõte

- Arvuti võtab sisendseadmetelt vastu infot andmete ja programmide kujul...
- ... ja salvestab selle mälus
- Info tuuakse mälust aritmeetika-loogika seadmesse (vastavalt täidetavale programmile)...
- ... ja töödeldakse seal (ALUs)
- Töödeldud info saadetakse väljundseadmesse
- Kõiki arvutisiseid tegevusi koordineerib juhtimisüksus

Arvuti arhitektuur FKEF.02.143



---

---

---

---

---

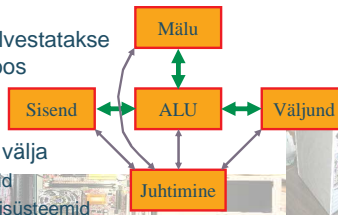
---

---

---

## Von Neumanni mudel

- John von Neumann (1903-1957)
- ENIAC'i meeskonna juhid J. Presper Eckert ja John William Mauchly
- Viis põhilist komponenti:
- Kõige olulisem aspekt: **salvestatud programmi käivitamine**
- Programm salvestatakse arvuti mälus koos andmetega



- Siit kasvavad välja
  - kompilaatorid
  - operatsioonisüsteemid

Arvuti arhitektuur FKEF.02.143



---

---

---

---

---

---

---

---

## Käsu näited

Näide 1

- Add LOCA, R1

Näide 2

- Load LOCA, R0
- Add R0, R1

Arvuti arhitektuur FKEF.02.143



---

---

---

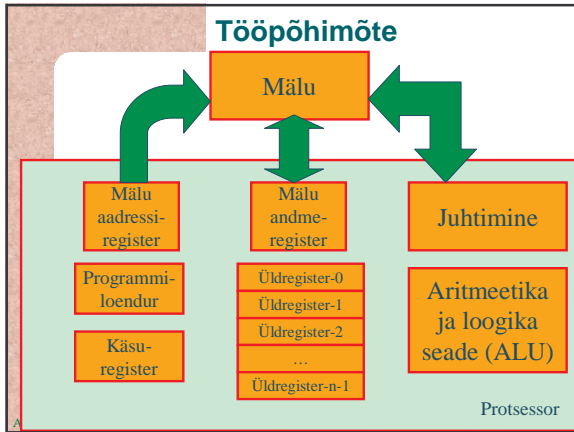
---

---

---

---

---




---

---

---

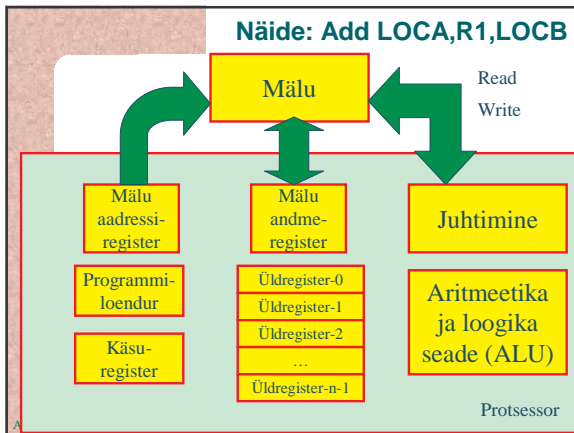
---

---

---

---

---




---

---

---

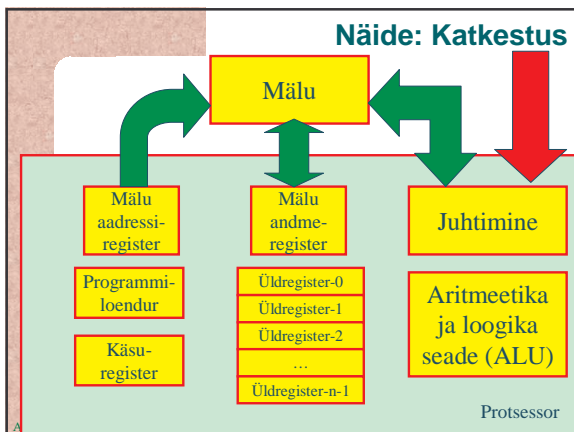
---

---

---

---

---




---

---

---

---

---

---

---

---

## Siini ehitus (1)

- Seni vaatlusime üksikuid süsteemielemente
- Need on vaja ühendada selliselt, et infot saaks liigutada ühelt seadmelt teisele
- Seda ülesannet täidab siin (*bus*)
- Lisaks andmetele peab siin üle kandma ka aadressiinfo ja juhtimiskorraldused

Arvuti arhitektuur FKEE.02.143



---

---

---

---

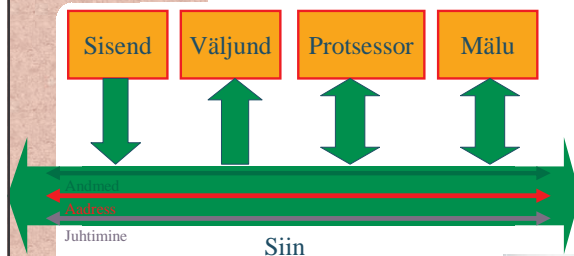
---

---

---

---

## Siini tööpõhimõte



Arvuti arhitektuur FKEE.02.143



---

---

---

---

---

---

---

---

## Siini ehitus (2)

- Ühe siiniga süsteem
  - Korraga saavad suhelda ainult kaks seadet
  - Tehniline teostatavus lihtne ja odav
- Mitme siiniga süsteem
  - Korraga saavad suhelda mitmed seadmete paarid
  - Jõudlus parem
  - Keerulisem süsteem ja kõrgem hind

Arvuti arhitektuur FKEE.02.143



---

---

---

---

---

---

---

---



### Siini ehitus (3)

- Siinil suhtlevad eri kiirusega seadmed
  - Aeglased (klaviatuur, hiir, printer)
  - Kiired (magnetilised ja optilised kettad)
  - Väga kiired (mälu, protsessor)
- Kiired seadmed ei tohi oodata aeglaste taga
- Lahendus: *buffer registers*

Arvuti arhitektuur FKFE02.143



---

---

---

---

---

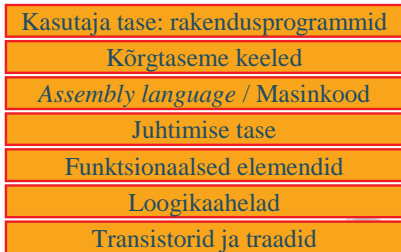
---

---

---

### Kihstruktuur

- Arvuti võib jagada seitsmeks üksteisest suhteliselt sõltumatuks tasemeks:



Arvuti arhitektuur FKFE02.143



---

---

---

---

---

---

---

---

### Süsteemitarkvara

- Kasutaja käskude vastuvõtmine ja interpreteerimine
- Rakendusprogrammide sisestamine, redigeerimine ja salvestamine
- Failide salvestamise ja lugemise haldamine
- Standardprogrammide käivitamine kasutaja poolsete andmetega
  - Tekstitöötlus
  - Tabelarvutus
  - Mängud
  - jne
- Sisend/väljundseadmete juhtimine. Info vastuvõtmine sisendist ja info väljastamine väljundisse
- Programmi lähteteksti tõlkimine objektikoodi (masinkoodi)
- Kasutajarakenduste sidumine standardalgoritmidega (*standard library*)

Arvuti arhitektuur FKFE02.143



---

---

---

---

---

---

---

---

## Süsteemitarkvara

- Süsteemitarkvara vastutab kõigi arvutiga tehtavate tegevuste koordineerimise eest
- Rakendused programmeeritakse kõrgtaseme programmeerimiskeeltes
- Kompilaator teisendab selle arvutile mõistetavaks masinkoodiks
- Tekstiredaktor
- Failisüsteem

Arvuti arhitektuur FKEF.02.143



---

---

---

---

---

---

---

---

## Operatsioonisüsteem

- Käskude kogum, mis haldab arvutikomponentide omavahelist suhtlust ja ressursside jagamist rakendusprogrammidega töötamisel
  - mälupiirkonna reserveerimine
  - salvestuspiirkonna reserveerimine kõva- või optilisel kettal
  - seda nii andmete kui ka programmi jaoks
  - andmete liigutamine mälu ja ketaste vahel
  - sisend/väljund operatsioonide juhtimine

Arvuti arhitektuur FKEF.02.143



---

---

---

---

---

---

---

---

## Näide

- Olgu meil lihtne arvuti (protsessor, mälu, kettaseade, võrguseade)
- Käivitame ühe rakendusprogrammi, mis
  - loeb andmed kettalt,
  - töötleb neidja siis
  - saadab tulemuse serverisse

Arvuti arhitektuur FKEF.02.143



---

---

---

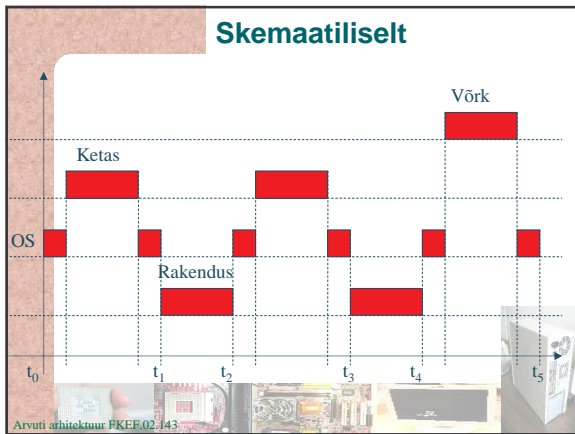
---

---

---

---

---




---

---

---

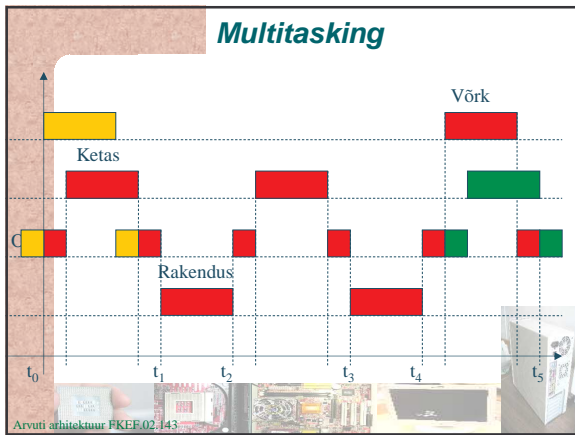
---

---

---

---

---




---

---

---

---

---

---

---

---

### Jõudlus

- Jõudlust mõjutavad
  - Riistvara
  - Masinkoodi ülesehitus
  - Kompilaatori efektiivsus
- Programmi tööks kulunud aeg (*elapsed time*)
  - $t_5 - t_0$  eelmisel joonisel
- Protsessori tööaeg
  - OS ja rakendusprogrammi tööaegade summa eelmisel joonisel

Arvuti arhitektuur FKFE.02.143

---

---

---

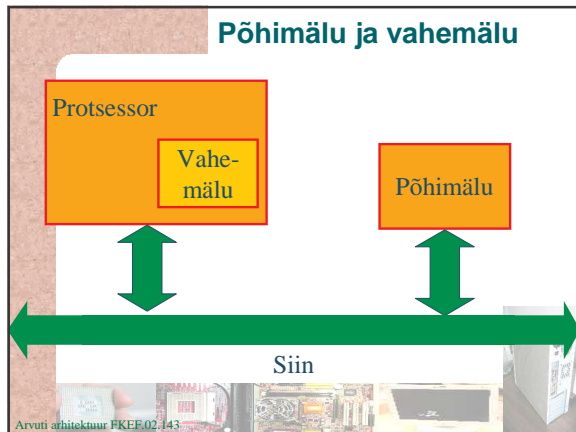
---

---

---

---

---




---

---

---

---

---

---

---

---

### Näide vahemälu kasulikkusest

- Olgu meil arvuti, mille vahemälust andmete lugemine on 20 korda kiirem andmete lugemisest põhimälust.
- Oletame, et programmi käivitusaeg on võrdeline käsu kättesaamise kiirusega ...mälust.
- Oletame veel, et konkreetse programmiga töötamisel leitakse käsk vahemälust tõenäosusega 97%.
- Arvuta programmi "ilma vahemäluta" käivitusaja suhe "vahemäluga" käivitusajaga.
- Seejuures eeldame veel, et ka põhimälust andmete lugemisel tuleb need andmed kirjutada esmalt vahemälusse (kui vahemälu olemas on).

$$t_1 = N \times (0,03 \times 20 \times t + t) = N \times 1,6 \times t$$

$$t_2 = N \times (20 \times t) = N \times 20 \times t$$

$$\frac{t_2}{t_1} = \frac{20}{1,6} = 12,5$$

Arvuti arhitektuur FKKEF.02.143

---

---

---

---

---

---

---

---

### Protsessori taktigeneraator

- Protsessori töö koordineerimiseks kasutatakse ajastamissignaali
- taktsignaali, ingl *clock*

- Taktsignaali defineerib regulaarse ajaintervalli kestusega  $T$
- Kõik käsud jagatakse väiksemateks sammudeks, mida saab teha aja  $T$  jooksul
- Protsessori taktsagedus  $F = 1 / T$

Arvuti arhitektuur FKKEF.02.143

---

---

---

---

---

---

---

---

## Taktsagedus

- Taktsagedust mõõdetakse ühikutes *takti sekundis*
- Füüsikas nimetatakse seda ühikut hertsiks, tähis Hz
- Tänapäeva arvutite taktsagedused on mõne GHz suurusjärgus (Giga = 1 000 000 000)
- Vanematel arvutitel mõnisada MHz (Mega = 1 000 000)
- 2 GHz taktsageduse korral on  $T = 0,5$  ns
- 400 MHz taktsageduse korral on  $T = 2,5$  ns

Arvuti arhitektuur FKFE02.143

---

---

---

---

---

---

---

---

## Jõudlusvõrrand

- Arvutame protsessori tööaja, mis kulub
  - $N$  masinkoodi käsu käivitamiseks
  - arvutis taktsagedusega  $F$ .
  - Ühe masinkoodi käsu täitmiseks läheb vaja keskmiselt  $S$  sammu (kõik sammud jõutakse täita ühe takti jooksul)
- **NB! Protsessor taktsagedusega 2 GHz ei pruugi alati jõudsam olla kui 1,83 GHz protsessor!**

$$t = \frac{N \times S}{F}$$

Arvuti arhitektuur FKFE02.143

---

---

---

---

---

---

---

---

## Näide

- Olgu meil kaks arvutit.
  - Esimese arvuti efektiivne  $S$  väärtus olgu 1,5 ja protsessori taktsagedus 2,0 GHz
  - Teise arvuti efektiivne  $S$  väärtus olgu 1,2 ja protsessori taktsagedus 1,83 GHz
- Masinkoodi käskude arvud neis arvutites olgu võrdsed
- Kumb arvuti saab testprogrammi täitmisega enne hakkama?

Vastus:  
Arvuti 2 on selle ülesande jaoks sobivam

$$\frac{N \times 1,5}{2 \text{ GHz}} > \frac{N \times 1,2}{1,83 \text{ GHz}}$$

Arvuti arhitektuur FKFE02.143

---

---

---

---

---

---

---

---

## Toru (Pipeline)

- Senises käsitluses arvestasime, et käsud täidetakse üksteise järel
- Oluliselt efektiivsem oleks, kui saaks osa käske täita üheaegselt
- Näide: Add R0,R1,R2
- Ideaaljuhul lõpetatakse iga takti jooksul ühe käsu täitmine, ehk siis efektiivselt  $S$  väärtuseks oleks 1
- NB! Iga üksiku käsu täitmiseks kulub endiselt suurem arv takte
- Reaalses elus  $S \geq 1$

Arvuti arhitektuur FKEE02.143



---

---

---

---

---

---

---

---

## Superscalar execution

- Mitu toru (*pipeline*)
- S.t kasutatakse mitut funktsionaalset elementi ja käske täidetakse paralleelselt
- Nüüd on võimalik, et  $S < 1$

Arvuti arhitektuur FKEE02.143



---

---

---

---

---

---

---

---

## Taktsageduse suurendamine

- Üks võimalus on teha kvaliteetsemad ja kiiremad loogikaahelad
  - See kiirendab kõiki protsessori sees tehtavaid operatsioone
  - Põhimõluga suhtlus sellest ei kiirene (kuna pudelikael on teises kohas)
- Teine võimalus on vähendada ühe sammu vältel tehtavaid rehkendusi
  - Kui vajalikud tegevused jäävad samaks, suureneb nüüd vajalike sammude arv
  - Kas suureneb ka arvuti jõudlus, on veidi keerulisem küsimus

Arvuti arhitektuur FKEE02.143



---

---

---

---

---

---

---

---

## CISC ja RISC

- Lihtsate käskude täitmiseks kulub vähe samme
  - Suur  $N$  aga väike  $S$  jõudlusvõrandis
- Keeruliste käskude täitmiseks kulub palju samme
  - Väike  $N$  aga suur  $S$  jõudlusvõrandis
- Võtmesõnaks siin on *pipelining*
  - Keeruliste käskude korral on sellest rohkem kasu
  - Lihtsate käskude korral on seda lihtsam rakendada
- RISC (*Reduced Instruction Set Computers*)
- CISC (*Complex Instruction Set Computers*)

Arvuti arhitektuur FKEE02.143

---

---

---

---

---

---

---

---

## Kompilaator

- Kompilaator tõlgib kõrg-keeke käsud masinale arusaadavaks masinkeeleks
- Selleks, et  $N$  oleks võimalikult väike, peab meil olema hea käsustik ja kompilaator peab oskama seda ka efektiivselt kasutada
- Osutub, et taktide arv, mis kulub programmi täitmiseks, ei sõltu mitte ainult **valitud käskudest** vaid ka **nende käskude järjekorrast**
- Parema tulemuse saamiseks tuleb kompilaator ja protsessor disainida üheaegselt, tihedas koostöös toote loojate vahel

Arvuti arhitektuur FKEE02.143

---

---

---

---

---

---

---

---

## Jõudluse hindamine

- Otsusatavaks on parameetri  $t$  väärtus vajalike programmide käivitamisel
- $t$  väljarehendamise ei ole lihtne
- Selle asemel kasutatakse testprogramme (*benchmark programs*)
- Mittetulundusühing *System Performance Evaluation Corporation* (SPEC) valib välja ja avaldab testimiseks valitud rakendusprogramme – vaata: [www.spec.org](http://www.spec.org)

$$SPEC\ rating = \frac{t_{\text{võrdlusarvuti}}}{t_{\text{testitav arvuti}}}$$

Arvuti arhitektuur FKEE02.143

---

---

---

---

---

---

---

---

## SPEC rating

- $n$  programmi korral tuleb arvutada geomeetriline keskmine neist hinnangutest:

$$SPEC\ rating = \sqrt[n]{\prod_{i=1}^n \frac{t(\text{võrdlusarvuti})_i}{t(\text{testarvuti})_i}}$$

Arvuti arhitektuur FKEE.02.143

---

---

---

---

---

---

---

---

## Näide

- Näide: kolme testprogrammi käivitamisel võrdlusarvutis saadi tulemusteks vastavalt 30, 20 ja 60 sekundit.
- Samade programmide käivitamisel testarvutis saadi tulemuseks 10, 5 ja 15 sekundit.

$$SPEC\ rating = \sqrt[3]{\frac{30}{10} \times \frac{20}{5} \times \frac{60}{15}} = \sqrt[3]{3 \times 4 \times 4} = 3,63$$

Arvuti arhitektuur FKEE.02.143

---

---

---

---

---

---

---

---

## Mitme protsessoriga arvutid

- Ühes arvutis võib sees olla ka mitu protsessorit
  - selline arvuti suudab üheaegselt käivitada mitut programmi
  - või teha ühe programmi alamoperatsioone paralleelselt
- Enamasti suudavad kõik protsessorid suhelda kogu süsteemi mälega
- Alternatiiv: ühendatud arvutid
  - siin kasutab iga arvuti ainult oma mälu
  - suheldakse sõnumite abil (kommunikatsioonivõrgus)

Arvuti arhitektuur FKEE.02.143

---

---

---

---

---

---

---

---



## Ajalooline taust

- Kursuses Arvutiriistvara I rääkisime mehaanilistest arvutitest ja sellest, kuidas II maailmasõja ajal jõuti esimeste elektronarvutiteni (vaata videoloengut)
- Tehnoloogia arengu alusel võib elektronarvutid jagada neljaks põlvkonnaks
  - 1. põlvkond 1945-1955
  - 2. põlvkond 1955-1965
  - 3. põlvkond 1965-1975
  - 4. põlvkond 1975-tänaeni

Arvuti arhitektuur FKFE.02.143

---

---

---

---

---

---

---

---

## Esimene põlvkond 1945-1955

- Salvestatud programmi kontseptsioon John von Neumann'ilt
- Programme ja andmeid hoiti samas mälus
- Assembleris (*Assembly language*) kirjutet programmid kompileeriti masinkoodiks
- Loogikaelemendid baseerusid vaakumlampidel
- Elavhõbe viivis-mälud (*Mercury delay-line memory*)
- Elementaarsete aritmeetikaoperatsioonide kiirus – mõned millisekundid

Arvuti arhitektuur FKFE.02.143

---

---

---

---

---

---

---

---

## Teine põlvkond 1955-1965

- Loogikaelemendid baseerusid transistoritel (loodi 1940-ndate lõpus AT&T Bell Laboratories)
- Programmeeriti juba kõrg-taseme keeltes (Fortran). See kompileeriti assemblerisse (*Assembly language*) ja sealt edasi masinkoodiks
- Kasutati eraldiseisvaid I/O protsessoreid
- Magnetilise tuumaga mälud ja magnetilised salvestusseadmed võeti laialdaselt kasutusele (esimesed juba 1. põlvkonnas)
- Elementaarsete aritmeetikaoperatsioonide kiirus – mõned mikrosekundid

Arvuti arhitektuur FKFE.02.143

---

---

---

---

---

---

---

---

### Kolmas põlvkond 1965-1975

- Õpiti paigutama suuremat hulka transistore ühele ränikiibile (Si) – integreeritud ahelate tehnoloogia
- Ka mälud valmistati sama tehnoloogiat kasutades
- Programmeerimises: mikroprogrammeerimine, parallelism, *pipelining*
- Operatsioonisüsteemi tarkvara võimaldas mitmel rakendusel jagada omavahel arvutusressurssi
- Vahemälu tegi mäluoperatsioonid kiiremaks
- Virtuaalmälu tegi maksimaalse mäluhulga suuremaks

Arvuti arhitektuur FKEE.02.143



---

---

---

---

---

---

---

---

### Neljas põlvkond 1975- ...

- Õpiti paigutama kogu protsessor koos suure osaga mälust ühele samale kiibile (*Very Large Scale Integration* - tehnoloogia)
- Sellist protsessorit nimetatakse mikroprotsessoriks
- Sülearvutid, lauaarvutid, tööjaamad ja kõik nad omavahel veel võrku (LAN, WAN, Internet) ühendatud
- Elementaarsete aritmeetikaoperatsioonide kiirus – mõned nanosekundid või alla selle

Arvuti arhitektuur FKEE.02.143



---

---

---

---

---

---

---

---

### Kasutatud kirjandus

- Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer organization 5<sup>th</sup> edition (2002) 805 p.
- Miles Murdocca, Vincent Heuring, Computer Architecture and Organization: An Integrated Approach (2007) 544 p.

Arvuti arhitektuur FKEE.02.143



---

---

---

---

---

---

---

---