

# Ievads Datoru Arhitektūrā

Arhitektūra un organizācija

# Kas tad ir DA

- Datoru Arhitektūras termins radies IBM kad Amdahl, Blaauw, and Brooks [1964] definēja to kā programmētājam redzamo komandu kopu. Tika uzskatīts ka datori ar vienādu arhitektūru varēs izpildīt programmas bez izmaiņām tajās.
- Pozitīvais:
  - Precīzi definētas arhitektūras var tikt realizētas dažādos veidos
  - Programmas kas rakstītas vienādās komandu kopās var tik izpildītas visās savietojamās realizācijās
- Mūsdienās datoru arhitektūra tiek definēta kā:
  - Komandu kopas arhitektūra (ISA) + Organizācija (konveijerizācija, atmiņas hierarhija, uzglabāšanas sistēmas ...)
  - ISA piemēri:
    - Digital Alpha (v1, v3) 1992
    - HP PA-RISC (v1.1, v2.0) 1986
    - Sun Sparc (v8, v9) 1987
    - MIPS (MIPS I, II, III, IV, V) 1986
    - Intel (8086,80286,80386,80486,Pentium, MMX, ...) 1978
    - Itanium 2001
    - Cell 2005

# Arhitektūra ↔ organizācija

- Viena arhitektūra var tikt realizēta izmantojot dažādas organizācijas
  - Visi x86 datori izmanto vienu pamata arhitektūru bet tiem ir dažādas kopnes, atmiņas veidi....
- Vienādas pamata arhitektūras dod iespēju izmantot koda saderību
- Arhitektūra tāpat ir programmētājam redzamo atribūtu kopa
  - Komandu kopas, attēlojuma bitu skaits, I/O iespējas un veidi, adresācijas veidi
  - Piemēram: vai ir reizināšanas komandas
- Organizācija tas ir kā funkcijas tiek realizētas
  - Funkcionālie mezgli un to savienojumi
  - Aparatūras parametri kas ir caurspīdīgi programmētājam
  - Vadības signāli, interfeisi, atmiņas tehnoloģijas
  - Piemēram: vai reizināšanas darbības ir realizētas kā Boota algoritma mezgls vai tās tiek veiktas izmantojot ciklisku saskaitīšanu

# Arhitektūra ⇔ Realizācija

- Vienai arhitektūrai var būt vairākas realizācijas
  - Veidojas datoru saimes
- Vairākas arhitektūras var tikt izveidotas izmantojot vienu realizāciju
  - Mikrokoda emulatori
- Vispārējas nozīmes arhitektūras – var veikt jebkuru uzdevumu (arī OS) par kurām tad arī turpmāk stāsts būs.....
- Aplikāciju specifiskas arhitektūras – konkrēta uzdevuma (biežāk gan klases) realizācija HW. Visdrīzāk nevar darbināt Windows. Pozitīvi tas ka HW ir daudz efektīvāks risinājums (veiktspējas, cenas, jaudas.. ziņās)

# Funkcijas ⇔ Struktūra

- Funkcijas ir individuālu komponentu (kas ir daļa no kopējās struktūras) darbības
- Datoru pamata funkcijas ir:
  - Datu apstrāde
  - Datu uzglabāšana
  - Datu pārvietošana
  - Vadība
- Struktūra ir veids kā saistītas komponentes

# Prasības DA

- Lietotnes
  - Vispārējas nozīmes (KD, darba stacijas, PC )  
sabalansēta veikspēja dažādiem uzdevumu apgabaliem (cena / veikspēja)
  - Zinātniskās (laika prognozēšana, simulācijas, CFD)  
augsta peldošā punkta aritmētikas veikspēja un adresējamās atmiņas apjoms
  - Iebūvētās (PDA, telefoni, drošības sistēmas)  
Zema cena un patērētā jauda (min. nepieciešmā jauda uzdevumam)
  - Komerciālās (DB, e-veikali, web-servisi)  
decimālās aritmētikas atbalsts, datubāzu/transakciju apstrāde (veikspēja, pieejamība, mērogojamība)
  - Sensoru (“gudrie putekļi”)  
neatjaunojami ar ultra zemu jaudu
- SW saderības veidi
  - Objektoda / binārā līmenī  
nav nepieciešama sw pārnese, lielāka hw izstrādes izmaksas
  - Programmēšanas valodas līmenī  
atrisina veco arhitektūru nastu

# Prasības DA

- Operētājsistēmu prasības
  - Adrešu lauks
  - Atmiņas pārvaldība / aizsardzība  
(drazu savākšana, reāllaika darbu plānošana)
  - Pārtraukumi / slazdi (traps)
- Standarti
  - Peldošā punkta (IEEE754)
  - I/O kopņu
  - OS
  - Tīklu
  - Programmēšanas valodu

# Datora komandas

- CPU var izpildīt tikai mašīnas kodu
- Katram datoram ir specifiska mašīnas komandu kopa ko tas spēj saprast un izpildīt
- Mašīnas komanda tiek attēlota kā bitu virkne
  - Šiem bitiem jādefinē:
    - Ko darīt (komandas kods)
    - Ar ko to darīt (izejas operandi)
    - Kur likt rezultātu (mērķa operands)
    - Kā turpināt darbu pēc tam kad komanda ir izpildīta



# Datora komandas

- Mašīnas kods tiek iedalīts laukos.
- Katrs lauks satur vienu no minētajām vienībām – operācijas kodu, operandus, ...).
- Lauki tiek organizēti atbilstoši *komandas formātam* ).



# Mašīnas komandas

- Mašīnas kodus iedala četros tipos:
  - Datu pārvietošanas komandas. (Pārvieto datus starp pamata atmiņu un CPU reģistriem)
  - Aritmētiskās un loģiskās komandas
  - Programmas vadības komandas (test , branch)
  - Ievadizvades komandas

# Mašīnas komandas

- Apskatot dažādas arhitektūras svarīgi ir noskaidrot sekojošas komandu atribūtus:
  - Adrešu skaitu komandā
  - Pieļaujamos operandu tipus
  - Izmantojamās adresācijas veidus
  - Komandu veidus
  - Reģistru piekļuves veidus
  - Komandas formātus
- Viss minētais kopumā tiek saukts par komandu kopu (instruction set)

# Komandas dzīves cikls

- Katru komandu CPU izpilda kā atbilstošajai komandai noteikto soļu kopu. Šo soļu kopu sauc par *komandas ciklu* (*instruction cycle*).

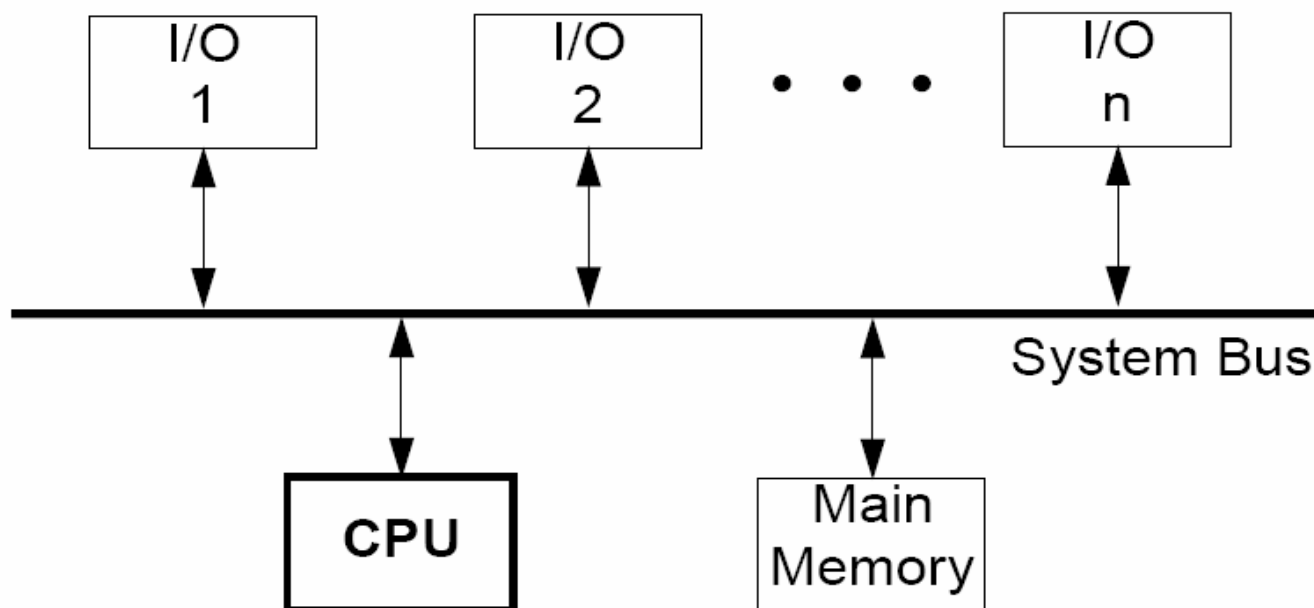
Vienkāršākajā gadījumā komandas dzīves ciklu var sadalīt divās daļās:

- Komandas nolasīšana
- Komandas izpilde

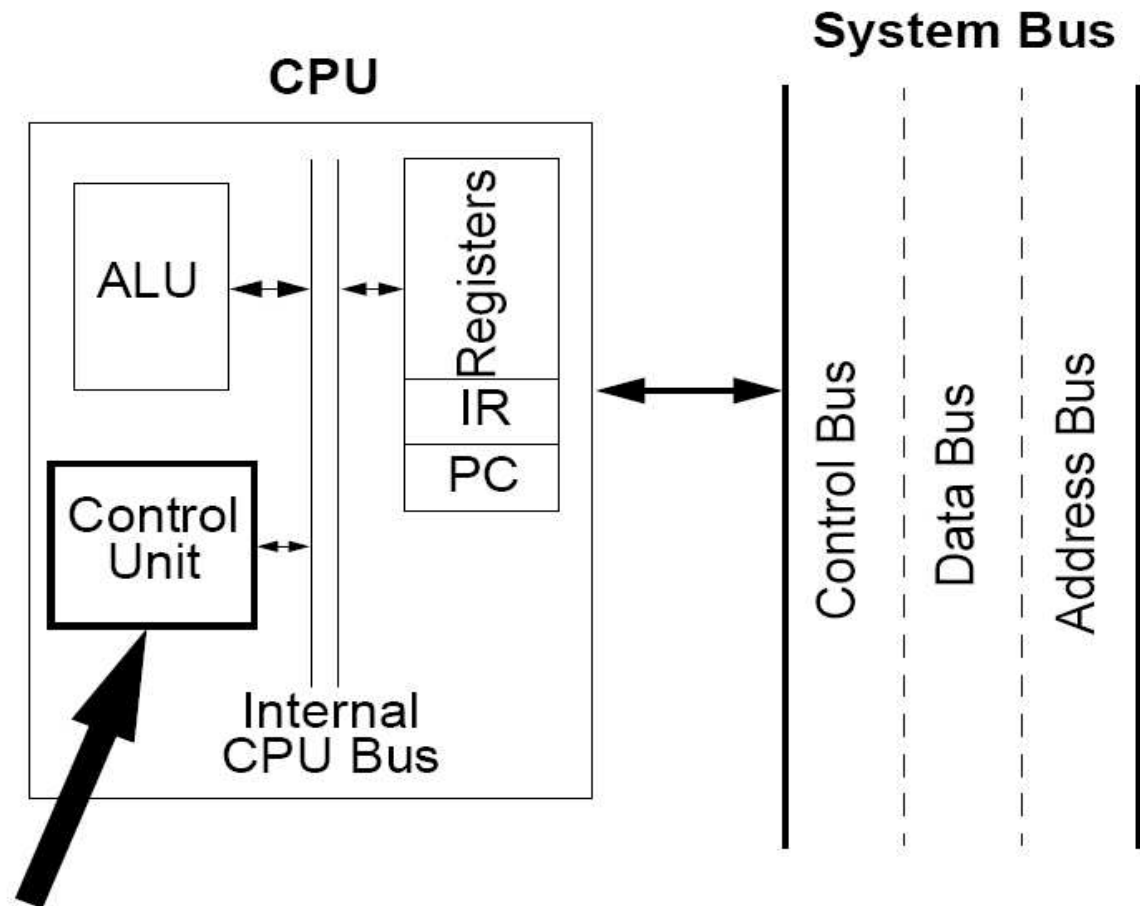
# Komandas dzīves cikls

- Smalkāk sadalot var iedomāties šādu dzīves ciklu:
  - Komandas nolasīšana
  - Komandas dekodēšana
  - Operandu nolasīšana
  - Komandas izpilde

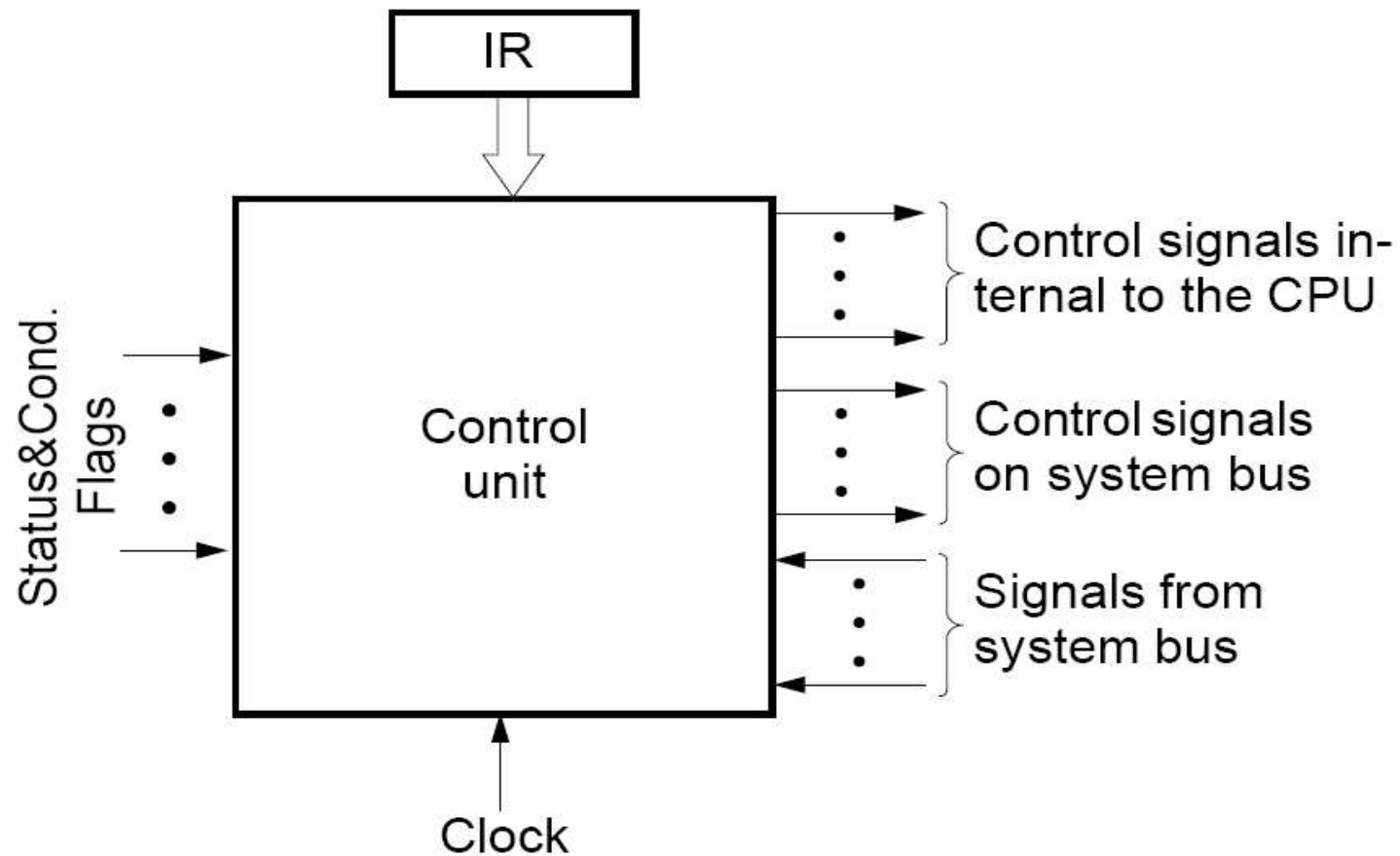
# Vadības mezgls



# Vadības mezgls



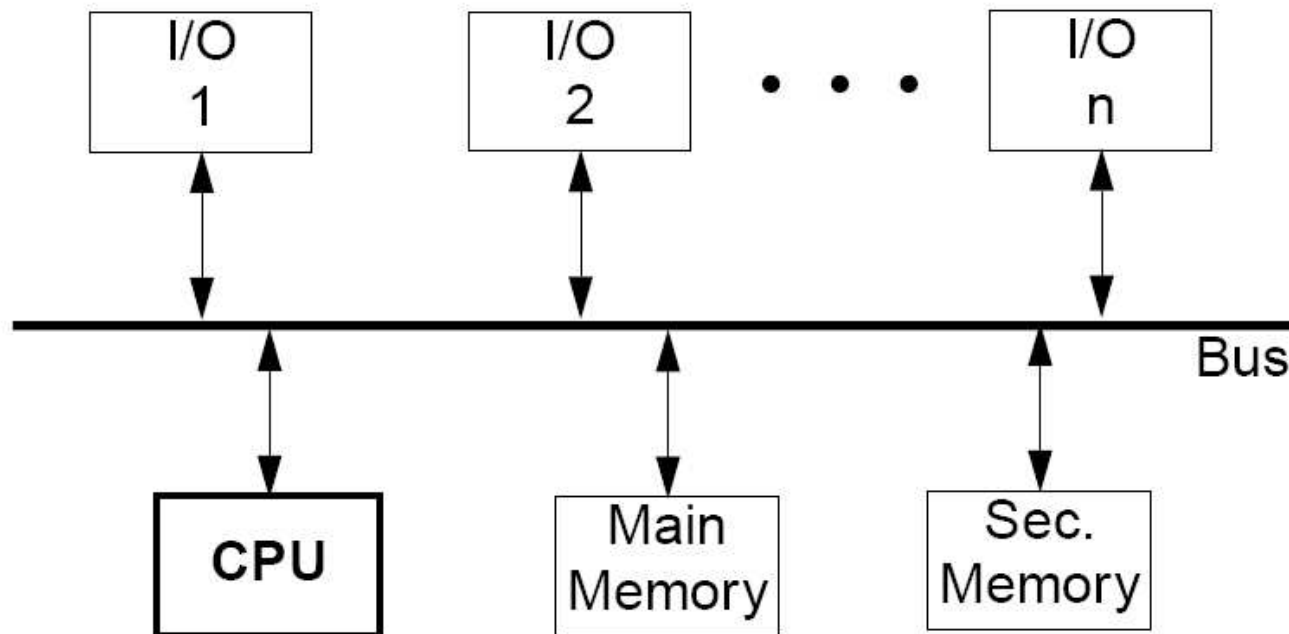
# Vadības mezgls





# Datorsistēma

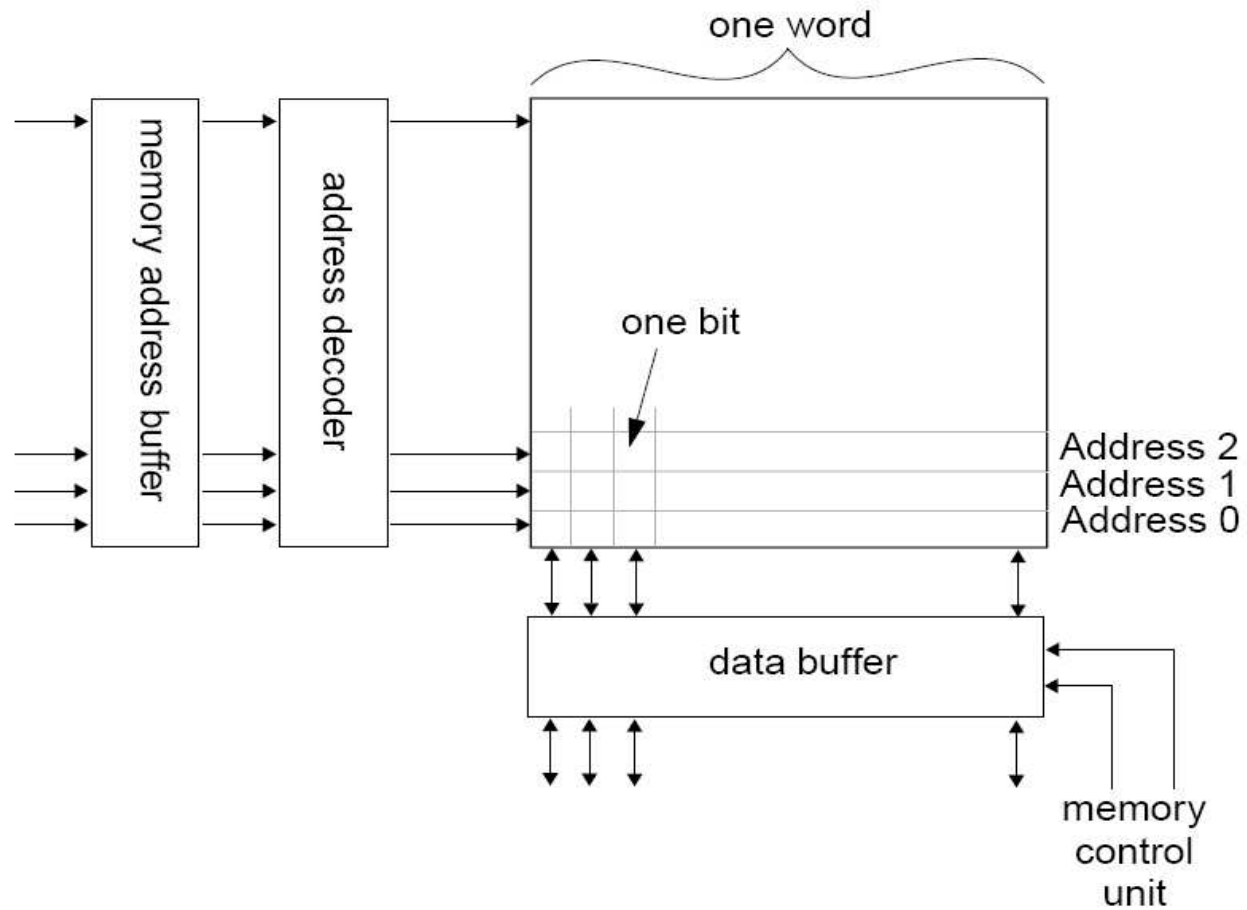
- *CPU + pamata atmiņa* sastāda datorsistēmas kodolu.
- *Sekundārā atmiņa + I/O iekārtas* sastāda perifēriju.
- Komunikāciju starp dažādām komponentēm pamatā šodien nodrošina viena vai vairākas *kopnes*.



# Atmiņa

- Pamata atmiņa tiek izmantota lai glabātu datus un programmu kurus dotajā momentā apstrādā CPU.
- Sekundārā atmiņa nodrošina ilglaicīgas glabāšanas vietu lieliem datu un programmu apjomiem.
- Pirms CPU var izmainīt vai apstrādāt datus tos vajag ielādēt pamata atmiņā.
- Atmiņas pamata parametri ir ātrdarbība, apjoms un izmaksas.
- Tipiski:
  - Pamata atmiņa ir ātra bet ierobežota savā apjomā.
  - Sekundārā atmiņa ir relatīvi lēna un ļoti apjomīga.

# Pamata atmiņa



# Pamata atmiņa

- Pamata atmiņu var uztvert kā uzglabāšanas vietu kopu kuras katra pozīcija var saglabāt vienu informācijas vārdu.
- Katrai kopas vietai tiek piekārtota viena unikāla adrese kas tiek numurēta secīgi 0,1,2,....
- Pamata atmiņā bez uzglabāšanas vietām vēl atrodas atmiņas *adrešu buferis* (darba adreses uzglabāšanai), *datu buferis* (darba datu uzglabāšanai), *adrešu dekoderis* un atmiņas *vadības bloks*.
- Pamat atmiņas realizācijai visplašāk šodien tiek pielietota pusvadītāju atmiņa.
- Visplašāk izplatītais pusvadītāju atmiņas tips ir operatīvā atmiņa - *random access memory* (RAM).
- RAM saturs pazudīs atslēdzot spriegumu.

# Mājās

- Izlasīt:
  - <http://dt.cs.rtu.lv/viewfile.php?path=18/file/4/431/1.clekija.pdf>
  - <http://dt.cs.rtu.lv/viewfile.php?path=18/file/75/432/2.lekcija.pdf>
  - <http://dt.cs.rtu.lv/viewfile.php?path=18/file/22/433/2.clekija.pdf>
- Nosaukt šodien plaši lietojamo sekundārās atmiņas iekārtu piemērus, minēt Jūsu datorā esošo kopņu nosaukumus, noskaidrot kā vēl (bez lekciju literatūrā minētā veida) var iedalīt datorsistēmu attīstību paaudzēs.