



# Kompiuterinių architektūros koncepsija Procesoriai

## 1 paskaita

prof.dr. Dalius Mažeika

[Dalius.Mazeika@vgtu.lt](mailto:Dalius.Mazeika@vgtu.lt)

<http://dma.vgtu.lt>

VG TU L424

# [ Dalyko turinys ]

- Paskaitos (teorinė medžiaga)
  - Pratybos
- 

## **Vertinimas**

Galutinio pažymio struktūra:

- 30% pratybų pažymys
- 70% teorijos egzaminas

# [ Literatūra ]

- V. Urbanavičius. **Kompiuteriai ir jų architektūra.** VGTU. 2007. (*www.ebooks.vgtu.lt*)
- A. S. Tanenbaum, D. J. Wetherall. **Computer Networks.** Prentice Hall. 2010.
- J. F. Kurose, K. W. Ross. **Computer Networking. A Top Down Approach.** Addison Wesley. 2010.
- C. M. Kozierok. **The TCP/IP Guide.** 2005

# [ Teorijos turinys ]

## ■ **Kompiuterių architektūra**

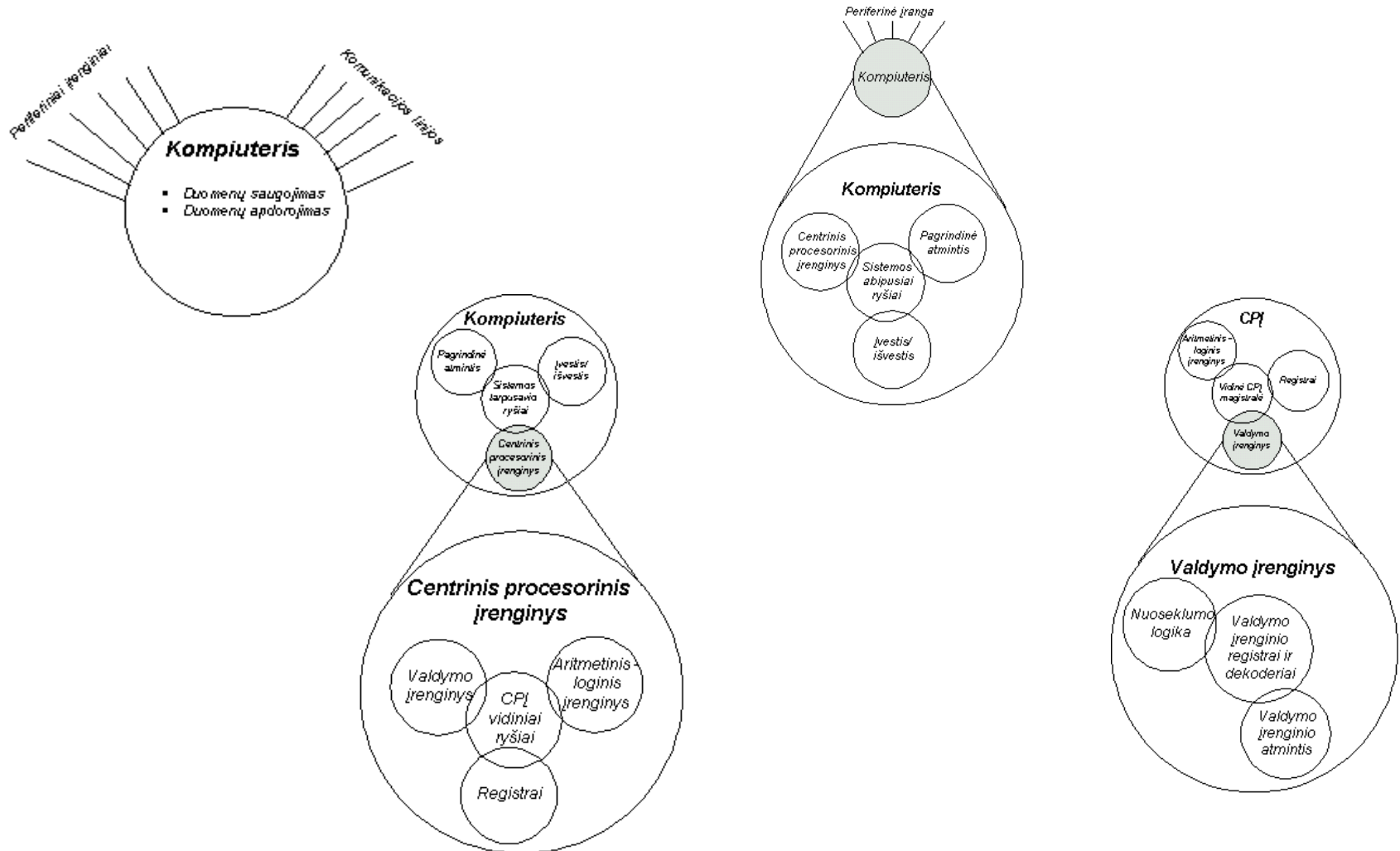
- Procesoriai (žodis, RISC, CISC, ILP, gijos, branduoliai)
- Operatyvioji atmintis (SRAM, DRAM, SDRAM, DDR, CAS, RAS, vėlinimas)
- Spartinančioji atmintis (L1, L2, L3, Direct mapped, Fully Associative, Set Associative)
- HDD, SDD, RAID, LUN, failų sistemos
- Failų saugyklos (DAS, NAS, SAN, protokolai)

# [ Teorijos turinys ]

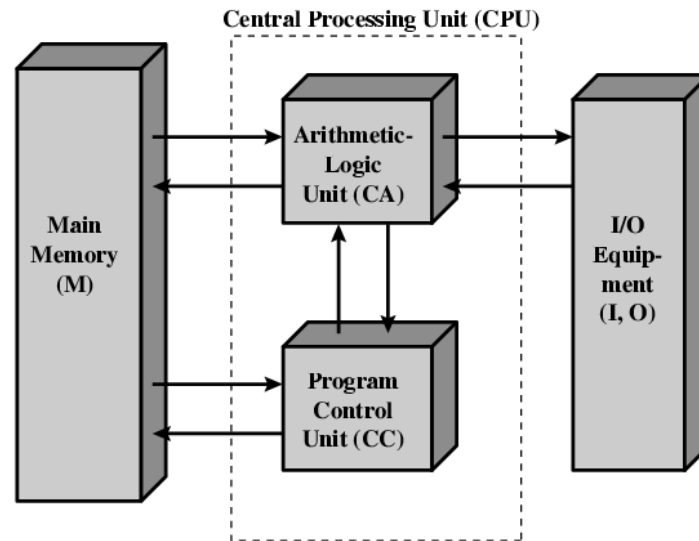
## ■ **Kompiuterių tinklai**

- Tinklo topologijos, fiziniai tinklų komponentai, jų pagrindinės savybės
- OSI ir TCP/IP modeliai
- Lokalių tinklų (LAN) technologijos, IEEE 802.x standartas
- Tinklo lygmuo, IP adresacija (IP v4, IPv6)
- Transporto lygmuo, TCP, UDP protokolai
- Globalių tinklų (WAN) technologijos
- Bevieliai tinklai (802.11x)
- Maršrutizavimo protokolai (BGP, RIP, OSPF)
- Taikomasis lygmuo (DNS, DHCP, FTP, ICMP, SMTP ir t.t.)

# Kompiuterio struktūra

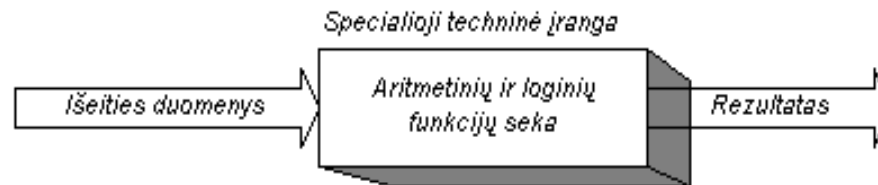


# von Neumann mašinos struktūra

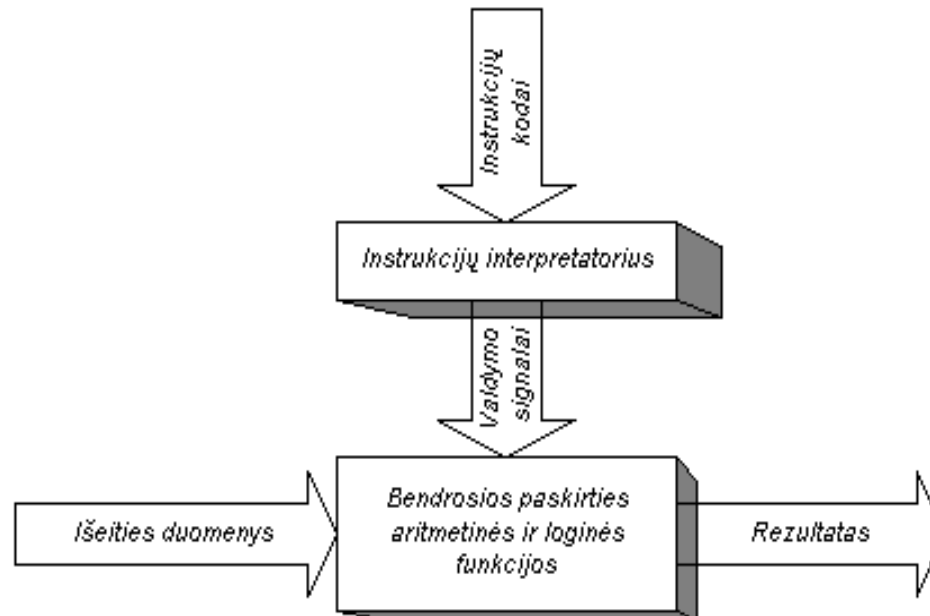


- Duomenys ir instrukcijos saugomos vienoje bendroje atmintyje, į kurią galima įrašinėti ir iš kurios galima skaityti.
- Šios atminties turinys yra adresuojamas pagal tam tikrą išsidėstymą, visiškai neatsižvelgiant į saugomų duomenų tipą.
- Programų vykdymas vyksta nuoseklyja tvarka pereinant nuo vienos instrukcijos prie kitos (kol ši tvarka bus sąmoningai modifikuota).

# Procesoriaus architektūros koncepsija



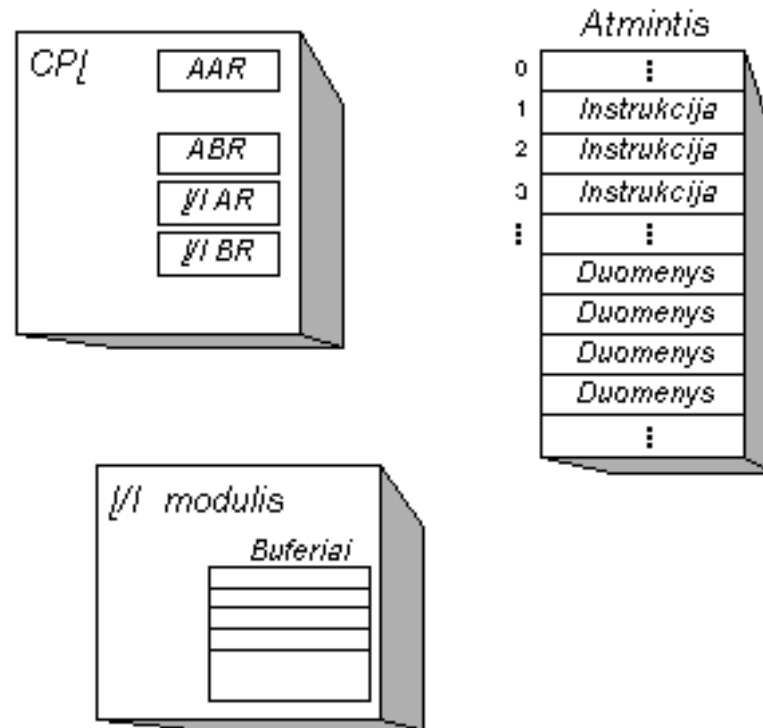
**a) Techninės įrangos "programavimas"**



**b) Programavimas specialia programine įranga**



# Kompiuterių architektūros koncepsija

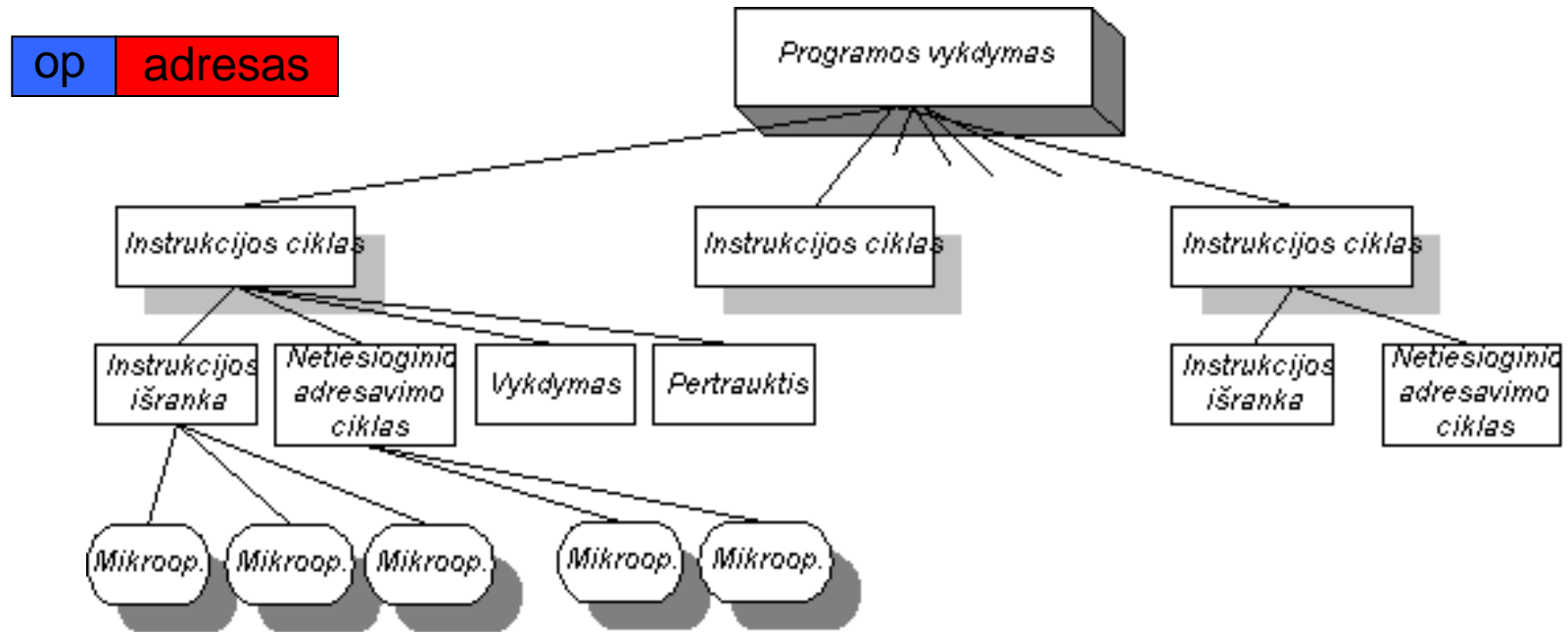


**2.2 pav.** Aukštesniojo hierarchijos lygmens kompiuterio komponentės; čia AAR – atminties adreso registras; ABR – atminties buferinis registras; ĮI AR – įvesties/išvesties adreso registras; ĮI BR – įvesties/išvesties buferinis registras

# [ Kompiuterio funkcionavimas ]

- **Programos vykdymas (duomenų apdorojimas) – pagrindinė kompiuterio funkcija.**
- **Vykdoma programa** – tai visuma op. atmintyje saugomų instrukcijų.
- Centrinis procesorinis įrenginys vykdo programoje nurodytas instrukcijas.
- **Programos vykdymo procesą** sudaro pasikartojantis instrukcijų išrankos bei instrukcijų vykdymo procesas.
- Patį instrukcijos vykdymo procesą gali sudaryti keli smulkesni žingsniai – operacijos (mikro-operacijos).

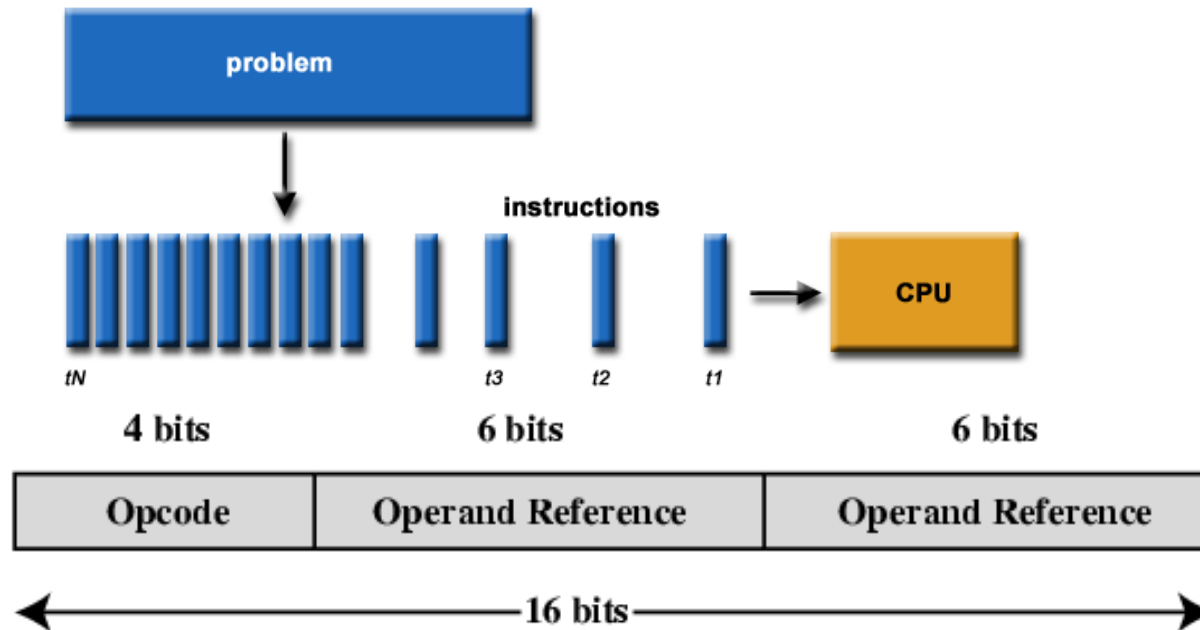
# Programos vykdymo elementai



**Pagrindinė kompiuterio funkcija** – programų vykdymas. Kompiuteris, vykdydamas programą, turi įvykdyti instrukcijų ciklą seką, vykdant kiekviename cikle po vieną mašininę instrukciją.

Akivaizdu, kad vykdomų instrukcijų seka gali neatkartoti programoje parašytą instrukcijų seką, nes joje gali būti šakojimosi instrukcijos.

# [Instrukcija]



**Instrukcija** - tai operacija, kurią vyko kompiuteris. Instrukcija priklauso nuo kompiuterio procesoriaus architektūros. Instrukcija gali būti skaidoma į smulkesnes operacijas.

`Add R1, R2, R3 ; R1 ← R2 + R3`

`Sub R4, R4, R1 ; R4 ← R4 - R1`

# [ Mikro-operacijos ]

Kiekvienas instrukcijos ciklas sudarytas iš smulkesnių etapų tokių, kaip: instrukcijos **išranka**, **netiesioginio adresavimo ciklas**, **vykdymas** ir **pertrauktis**.

Kiekvieną mažą instrukcijos ciklą galima suskaidyti į žingsnių, kuriuose visada naudojami CPJ registrai, seką. Šie žingsniai vadinami *mikrooperacijomis*.

**Mikro-operacijos** – tai pačios elementariausios CPJ vykdomos operacijos. Kiekviena instrukcijos ciklo fazė gali būti dekomponuota į mikrooperacijų seką.

# [ Instrukcijų rinkinio architektūra ]

**Instrukcijų rinkinys** – tai visų instrukcijų sąrašas, kurį gali įvykdyti procesorius.

## **Instrukcijos:**

Aritmetinės: **add, subtract**

Loginės: **and, or, not**

Duomenų: **move, input, output, load, store**

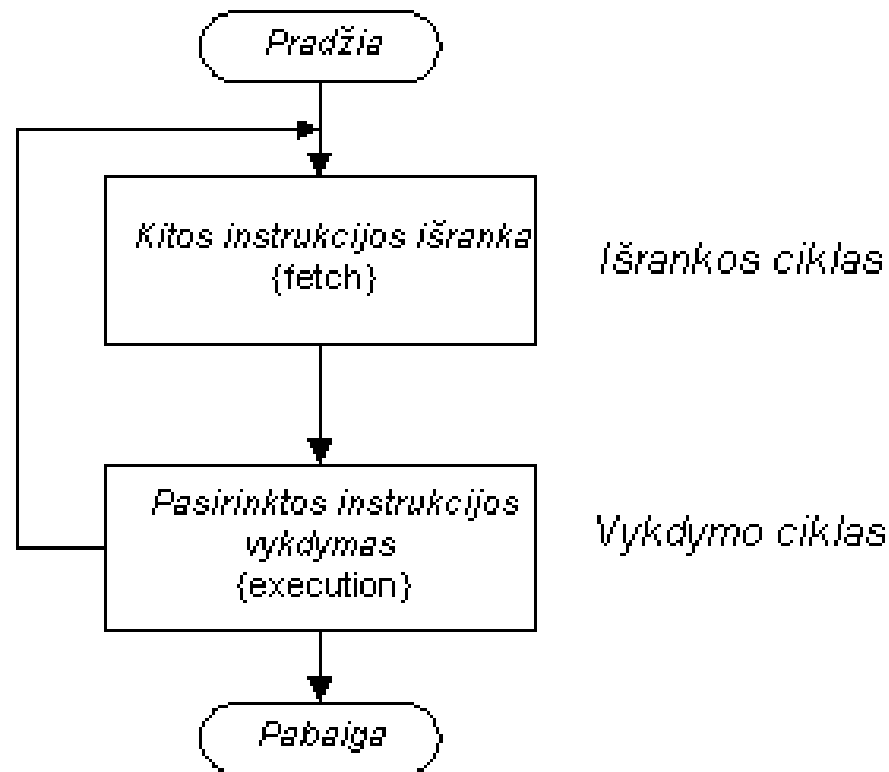
Valdymo: **goto, if ... goto, call, return**

Instrukcijų rinkinio architektūra (*ISA – Instruction Set Architecture*) apsprendžia procesoriaus architektūra.

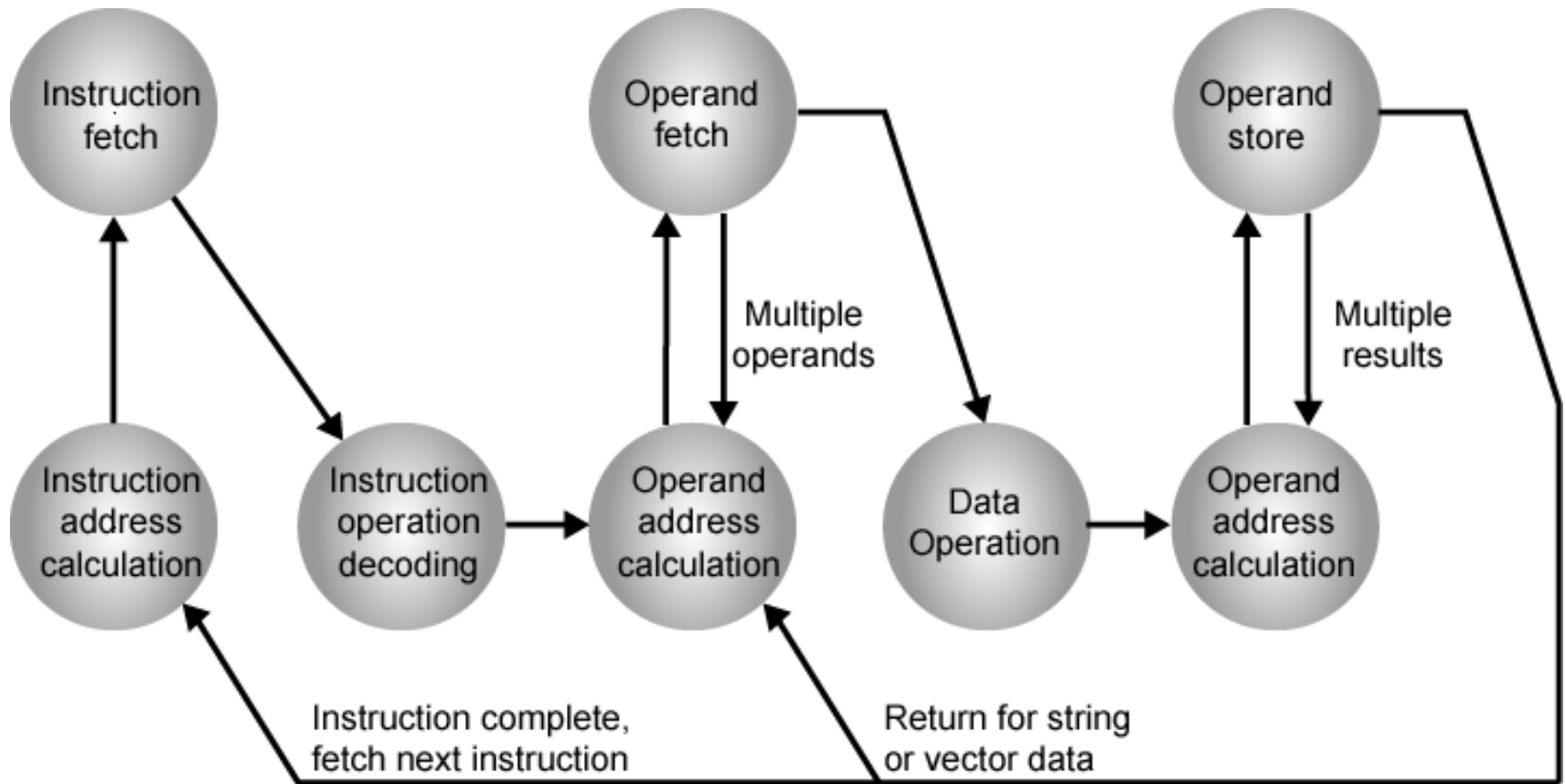
- Register-memory ISA (x86), kai atmintis pasiekama kaip dalis instrukcijos dalis
- Load-store ISA (MIPS, POWER, PA-RISC), kai atmintis pasiekama tik naudojant *load* ir *store* instrukcijas.

# Instrukcijos ciklas (trumpai)

Dviejų operacijų ( išrankos ir vykdymo) instrukcijos ciklas.

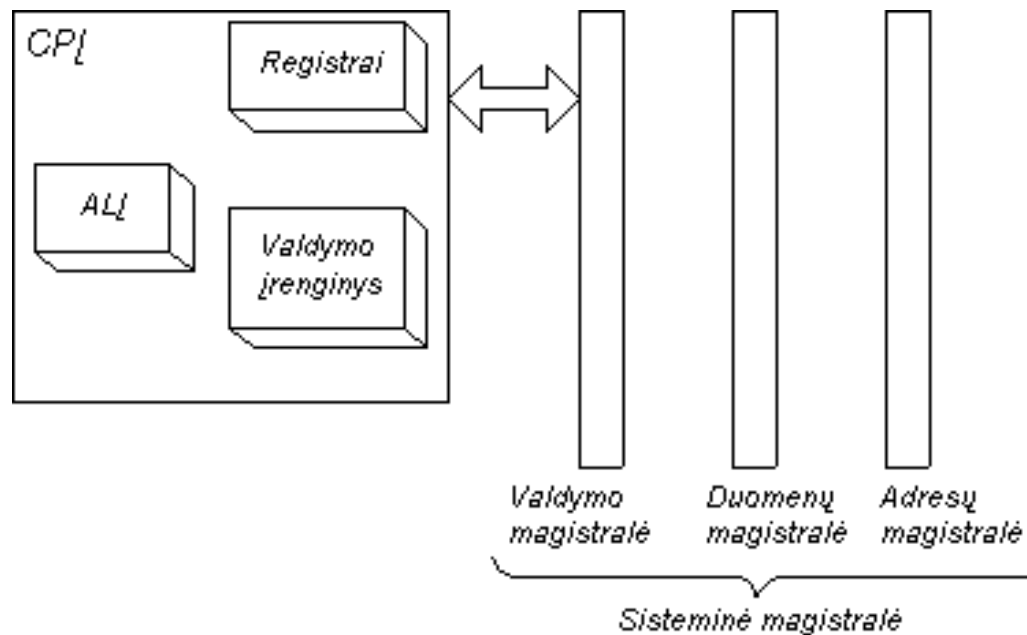


# Instrukcijos ciklinė diagrama

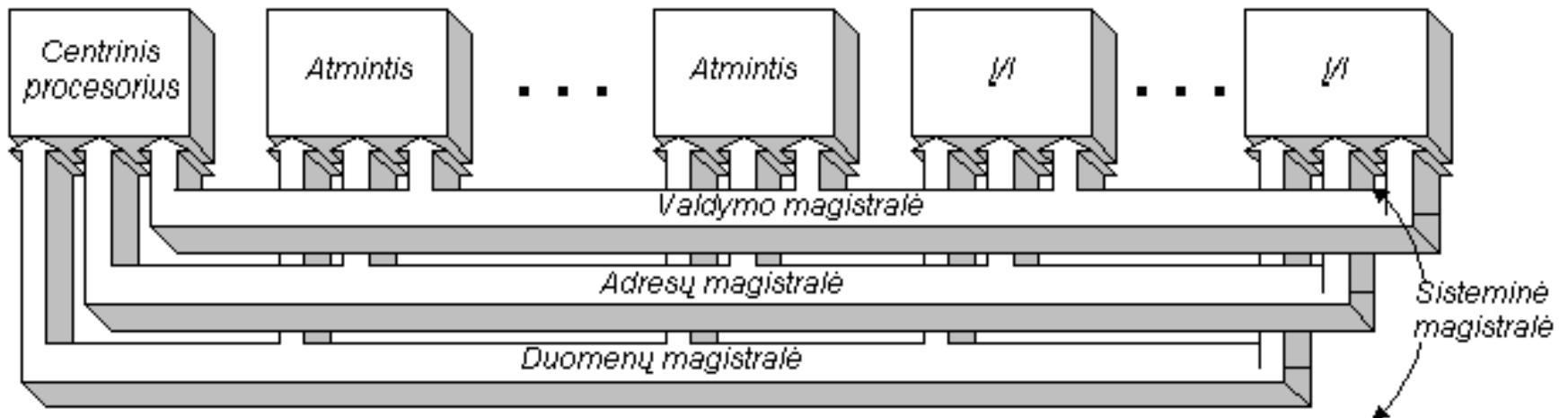




# Sisteminė magistralė



# Magistralės struktūra



- **Duomenų magistrale** vyksta keitimasis duomenimis tarp kompiuterio modulių.
- **Adresų magistrale** nurodo duomenų magistralėje esančios informacijos šaltinį ir paskirties įrenginį (destination).
- **Valdymo magistrale** kontroliuoja kreiptis (access) į duomenų ir adresų linijas ir šių linijų naudojimą.

# [ Registų sandara ]

Registų rinkinys funkcionuoja kaip atminties hierarchijos lygmuo, esantis aukščiau už pagrindinę ir spartinančiąją atmintis.

CPJ registrai skirti dviem funkcijoms vykdyti:

- **Vartotojo pasiekiami registrai.** Šie registrai leidžia programuotojui, taikant assemblerio programavimo kalbą arba tiesiog per mašininis kodus, sumažinti kreipčių į pagrindinę atmintį skaičių ir optimizuoti šių registų vartojimą.
- **Valdymo ir būklės registrai.** Jie naudojami CPJ valdymo įrenginio darbui kontroliuoti, o taip pat valdyti operacinių sistemų programų taikomųjų programų vykdymą.

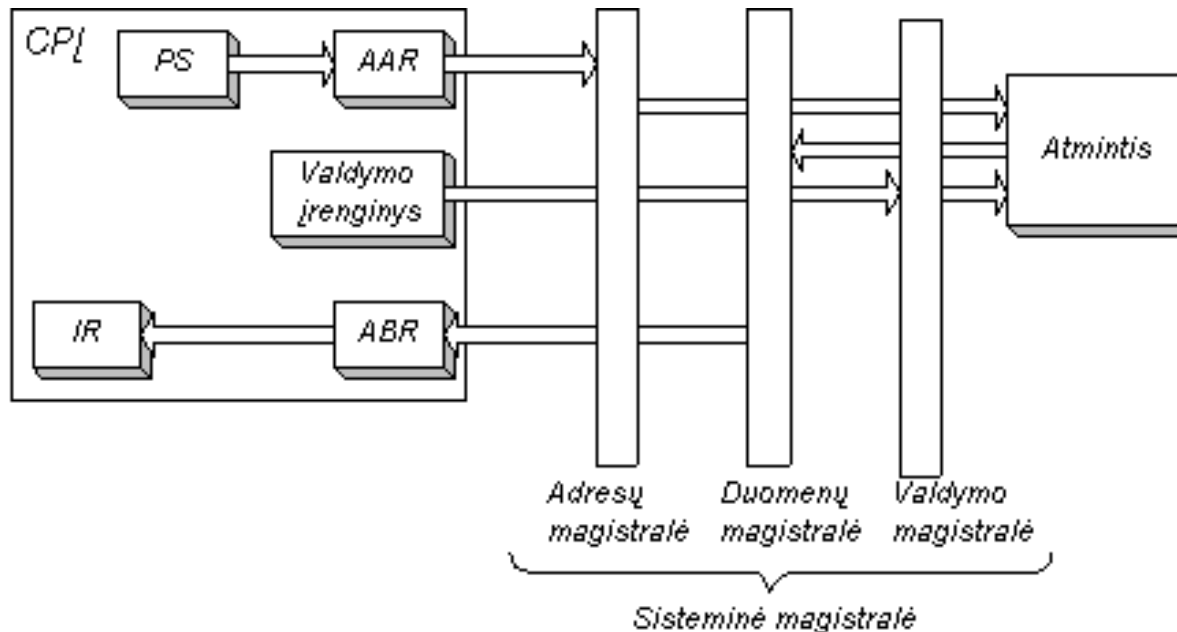
# Vartotojo pasiekiami registrai

- • Bendrosios paskirties;
  - Duomenų;
  - Adresų
  - Sąlygų kodų.
- **Bendrosios paskirties** registrai programuotojo gali būti pritaikyti įvairioms funkcijoms.
- **Duomenų registrai** gali būti taikomi tik duomenims saugoti ir niekada netaikomi apskaičiuojant operandų adresus.
- **Adresų registrai** patys gali būti realizuojami iš bendrosios paskirties registru, o gali būti skirti specialiems adresavimo metodams.
- **Sąlygų kodus** sudaro tam tikri bitai, kuriais gali operuoti CPJ techninė įranga, priklausomai nuo vykdomos operacijos rezultato.

# Valdymo būklės registrai

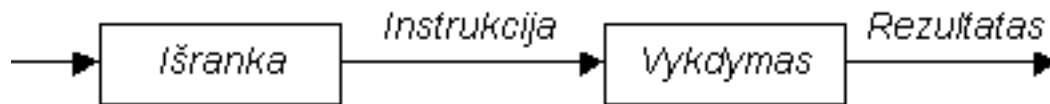
- **Programos skaitiklis (PS)**. Saugo sekančios išrenkamos instrukcijos adresą.
- **Instrukcijos registras (IR)**. Saugo naujausią išrinktą instrukciją.
- **Atminties adreso registras (AAR)**. Saugo atminties ląstelės, iš kurios ką tik buvo perskaityta informacija arba į kurią vyks duomenų rašymas, adresą.
- **Atminties buferio registras (ABR)**. Saugomas duomenų žodis, kuris bus įrašytas į atmintį arba iš atminties ką tik nuskaitytas.

# Duomenų srauto diagrama

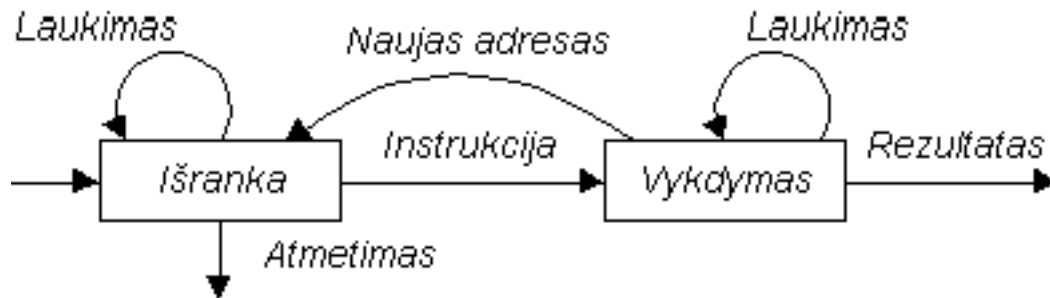


Instrukcija ciklo išrankos metu skaitoma iš atminties. PS registre saugomas sekančios išrenkamos instrukcijos adresas. Šis adresas siunčiamas į AAR ir patalpinamas adresų magistralėje. Valdymo įrenginys organizuoja skaitymą iš atminties. Nuskaitytas rezultatas patalpinamas duomenų magistralėje, kopijuojamas į ABR ir toliau siunčiamas į IR. Tuo tarpu PS turinys didinamas vienetu ir taip gaunamas sekančios instrukcijos adresas.

# Instrukcijų konvejeris



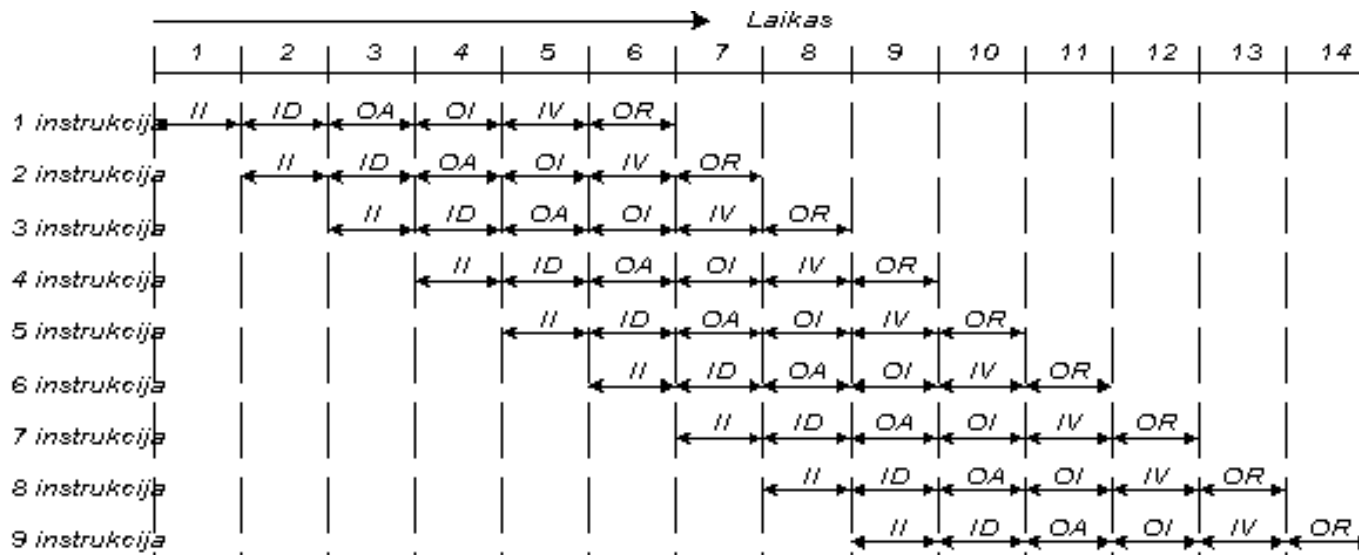
a) Supaprastinta schema



b) Išsamesnė schema

Instrukcijų konvejerizavimas panašus į surinkimo linijos taikymą produkcijos gamyboje. Surinkimo linijos privalumas yra tame, kad surinkimo procesas surinkimo linijoje išskaidomas į etapus ir tame, kad tam tikros gaminio surinkimo procedūros gali būti vykdomos vienu metu. Toks procesas dar vadinamas *konvejeriu* arba *vamzdynu* (pipeline), kadangi konvejeriye nauja operacija pradedama viename gale nesulaukus kol kita operacija kitame gale bus užbaigta.

# Instrukcijų konvejeris



**Instrukcijos išranka (II).** Į buferį nuskaitoma sekanti instrukcija po vykdomos.

**Instrukcijos dekodavimas (ID).** Nustatomi operacijos kodas ir operando specifikatoriai.

**Operando apskaičiavimas (OA).** Apskaičiuojamas operando efektyvusis (realusis arba fizinis) adresas. Tai gali būti perkėlimo, registrų netiesioginis adresavimo ir kitų formų adresų apskaičiavimai.

**Operandų išranka (OI).** Operandas arba operandai nuskaitomas iš atminties.

**Instrukcijos vykdymas (IV).** Vykdoma instrukcijoje nurodyta operacija ir išsaugomas jos rezultatas.

**Operando rašymas (OR).** Operacijos rezultatas įrašomas į atmintį.



# [ Konvejerio našumas problemos ]

Konvejerio našumą įtakojančius faktorius priimta skirstyti į tris klases:

- 1) **resursų konfliktus** (*structural hazards*);
  - 2) **duomenų priklausomumą** (*data hazards*);
  - 3) **valdymo nuoseklumo priklausomumą** (*control hazards*).
- 

**Resursų konfliktai** kyla tuomet, kai į tą patį vienintelį resursą kreipiasi ne mažiau kaip dvi komandos. Tokiu atveju viena komanda gali tuo resursu naudotis, o likusios komandos turi laukti, kol resursas bus atlaisvintas ir bus patenkinta kitos komandos paraiška. Jei to tipo resursas turi **k** egzempliorių (pavyzdžiui, superskaliariniame procesoriuje yra keli to paties tipo funkciniai įtaisai), aišku, kad vienu metu gali būti patenkinta **k** kreipinių.

# [ Konvejerio našumas problemos ]

**Duomenų priklausomumas** atsiranda dėl to, kad konvejerizuotame procesoriuje pasikeičia gretimų komandų atskirų žingsnių vykdymo nuoseklumas. Tai liečia operandų išrinkimą ir rezultato įrašymą. Pavyzdžiui, vykdoma tokia komandų pora:

Add R1, R2, R3 ; R1 ← R2 + R3

Sub R4, R4, R1 ; R4 ← R4 + R1

F	D	X	W	
	F	D	X	W

Komandos **Add** rezultatas bus įrašytas į registrą R1 fazėje W , o komanda **Sub** atėminį turi išrinkti iš registro R1 fazėje F , kuri vykdoma dviems ciklais anksčiau.

Šis pavyzdys atitinka vadinamąjį **tikrąjį duomenų priklausomumą**.

# Valdymo problemos konvejeryje

- II, OI ir OR etapuose vykdomi kreipiniai į atmintį. Diagramoje numatyta, kad visos šios kreiptys gali vykti vienu metu.
- Instrukcijos etapų trukmės skirtumai;
- Sąlygiškojo šakojimosi instrukcija;
- Tam tikrą neapibrėžtumą taip pat įneša pertrauktys.



**Klausimai?**