

2.1. SISTEMINĖS MAGISTRALĖS

Aukščiausiojo hierarchijos struktūros lygmens kompiuterį sudaro CPL, atmintis ir I/O komponentės. Kad kompiuteris vykdytų pagrindines funkcijas, šios komponentės tam tikru būdu tarpusavyje sujungtos. Kompiuterio pagrindinių funkcijų vykdymas užtikrina programų vykdymą. Taigi hierarchijos struktūros aukščiausiojo lygmens kompiuterį galima aprašyti:

- pirma, nusakant jo visų komponentių išorinę „veiklą“, t. y. kokiais valdymo signalais ir duomenimis jos keičiasi su kitomis komponentėmis;
- antra, nusakant tarpusavio ryšių struktūrą ir kontrolės sistemą, kurią reikia užtikrinti norint šią struktūrą valdyti.

Toks aukščiausiojo hierarchijos lygmens kompiuterio struktūros ir funkcijų nagrinėjimas labai svarbus, nes atskleidžia kompiuterio esmę. Tai taip pat svarbu norint suprasti sudėtingas kompiuterio pajėgumo įvertinimo metodikas, suvokti silpnas kompiuterio sistemos vietas, alternatyvius keitimosi duomenimis būdus bei kokių mastu žlugs sistema, sugedus vienai iš komponentių, nuo ko priklauso sistemos pajėgumo didinimas. Daugeliu atvejų sistemai keliami pajėgumo ir atsparumo gedimams reikalavimai labiau verčia iš esmės keisti kompiuterizuotos sistemos projektavimą negu didinti tam tikrų komponentių spartą ir patikimumą.

Šiame skyriuje dėmesį sutelksime į pagrindines struktūras, taikomas kompiuterio komponentėms sujungti. Trumpai apžvelgsime šias pagrindines komponentes ir paanalizuosime būtinumą jas jungti.

Nagrinėsime kompiuterizuotos sistemos komponentių jungimą magistralėmis.

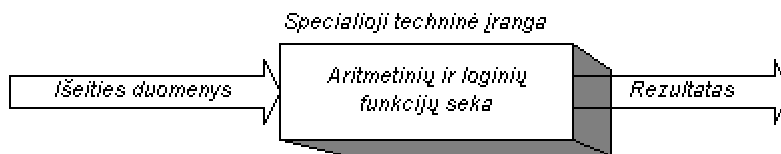
2.1. Kompiuterio komponentės

Visų šiuolaikinių kompiuterių projektavimas pagrįstas koncepcijomis, kurias Noimanas „Prinstono pažangiųjų studijų“ institute sukūrė dar 1945-aisiais metais. Tad šis projektavimas ir vadinamas Noimano architektūra. Ji remiasi trimis esminėmis koncepcijomis:

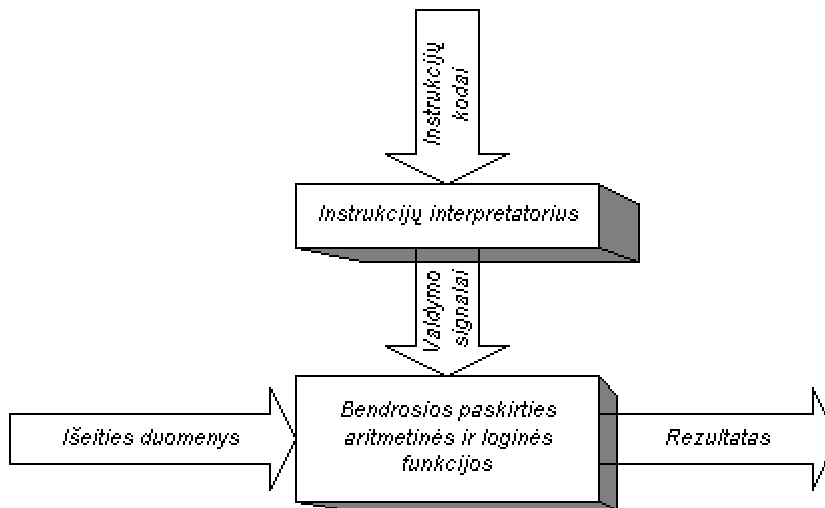
- Duomenys ir instrukcijos saugomos vienoje bendroje atmintyje, į kurią galima įrašinėti ir iš kurios galima skaityti.
- Šios atminties turinys yra adresuojamas pagal tam tikrą išsidėstymą, visiškai neatsižvelgiant į saugomų duomenų tipą.
- Programų vykdymas vyksta nuoseklia tvarka pereinant nuo vienos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

Kad dvejetainiai duomenys būtų įsimenami, saugomi, kad su jais būtų atliekami aritmetiniai ir loginiai veiksmai, paprastai reikia labai nedaug pagrindinių loginių komponentių, kurios pagal užduotį gali būti įvairiais būdais derinamos. Jeigu, pvz., bus vykdomi specifiniai skaičiavimai, loginių komponentių kombinacija projektuojama būtent taip, kad šie skaičiavimai galėtų vykti. Įvairių komponentių tarpusavio sujungimo procesą galime įsivaizduoti kaip tam tikrą programavimo formą. Galutinė „programa“ atrodys kaip specifiška techninė įranga {hardware}, todėl pavadinta *pastoviaja (techninė) programa* {hardwired program}.

Jeigu visos programos būtų realizuojamos panašiu būdu, iš tokios techninės įrangos būtų labai mažai naudos. Dabar panagrinėkime šio būdo alternatyvą. Įsivaizduokime, kad konstruojama bendrosios paskirties kompiuterio konfigūracija aritmetinėms ir loginėms funkcijoms. Tokia techninė įranga galės įvairiai apdoroti duomenis pagal gaunamus valdymo signalus. Kai techninė įranga originali, sukonstruota pagal specialų užsakymą, sistema gauna duomenis (juos apdoroja) ir teikia rezultatus (2.1 pav. a). Kai aparatūra yra bendrosios paskirties, sistema gauna duomenis, valdymo signalus (duomenis apdoroja) ir tik tada pateikia rezultatus. Taigi užuot techninę įrangą permontavus kiekvienai naujai programai programuotojui tereikia parengti naują valdymo signalų rinkinį.



a) Techninės įrangos "programavimas"



b) Programavimas specialia programine įranga

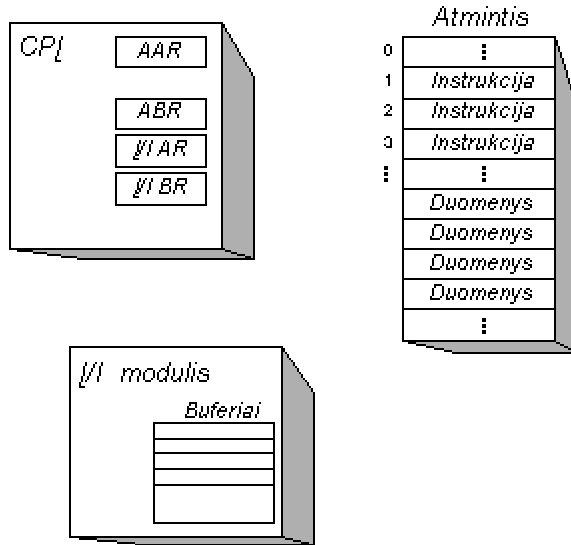
2.1 pav. Techniškai pagrįstas ir programinis požiūriai

Kaip valdymo signalai gali būti gaunami? Atsakymas labai paprastas. Visą programą sudaro atskirų žingsnių seka. Tam tikrame žingsnyje su tam tikrais duomenimis vykdomos aritmetinės arba loginės operacijos. Kiekvienam žingsniui reikalingas naujas valdymo signalų rinkinys. Įsivaizduokime, jog kiekvieną įmanomą valdymo signalų rinkinį atitinka unikalus kodas, ir, tarkime, kad prie bendrosios paskirties techninės įrangos pridėsime dar vieną bloką, kuris gali priimti kodą ir generuoti atitinkamus valdymo signalus (2.1 pav. b).

Dabar programuoti daug lengviau. Užuoat techninę įrangą permontavus kiekvienai naujai programai, tereikia gauti naują kodų seką. Kiekvienas kodas pagal veikimo rezultatą yra kaip instrukcija, o tam tikra techninės įrangos dalis kiekvieną instrukciją interpretuoja ir generuoja atitinkamus valdymo signalus. Šiam naujam programavimo metodui išskirti kodas arba instrukcijų seka pavadinta *programine įranga* {software}.

2.1 pav. b parodytos dvi svarbiausios kompiuterizuotos sistemos komponentės: instrukcijų interpretatorius ir bendrosios paskirties aritmetinių ir loginių funkcijų modulis. Šie du moduliai sudaro CPJ. Kompiuterio veikimui užtikrinti reikalingos dar kelios kitos komponentės. Duomenys ir instrukcijos turi būti talpinami kompiuterizuotoje sistemoje. Šiems tikslams būtinas tam tikro tipo įvesties modulis. Šiame modulyje turėtų būti komponentės, kurios priima duomenis bei instrukcijas ir juos konvertuoja į sistemai priimtina vidinių signalų formą. Natūralu, kad rezultatai turi įgyti tam tikrą išraišką, tai užtikrina išvesties modulis. Pastarųjų dviejų modulių visuma sudaro *I/O komponentes* {I/O components}.

Būtina dar viena komponentė. Įvesties įrenginys instrukcijas ir duomenis priiminėja nuosekliai. Tačiau programos ne visada vykdomos nuosekliai; jos gali daryti šuolius (pvz., pagal JUMP# instrukciją). Be to, operacijoms su duomenimis vienu metu gali prireikti daugiau negu vieno duomenų elemento iš anksto nustatytoje instrukcijų sekoje. Todėl turi būti numatyta vieta, kur laikinai bus saugomos ir instrukcijos, ir duomenys. Šis modulis pavadintas *atmintimi*, arba *pagrindine atmintimi*, siekiant jį atskirti nuo išorinių kaupiklių arba periferinių įrenginių. Noimanas pabrėžė, kad ta pati atmintis gali būti naudojama ir instrukcijoms, ir duomenims. Duomenys, kol vyksta skaičiavimai, visada traktuojami kaip duomenys. Tačiau instrukcijos taip pat gali būti traktuojamos kaip duomenys, iš kurių interpretuojami kodai valdymo signalams generuoti.



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

2.2 pav. parodytos kompiuterio hierarchijos aukštesniojo lygmens komponentės ir numatoma sąveika tarp jų. CPJ paprastai valdo visas funkcijas. Jis keičia duomenis atmintyje naudodamas du vidinius (CPJ atžvilgiu) registrus: atminties adresų registrą (AAR), kuris nusako adresą atmintyje, iš kur bus skaitomi arba rašomi kiti duomenys, ir atminties buferinį registrą (ABR), kuriame esti duomenys, rašomi į atmintį arba ką tik nuskaityti iš jos. Panašiu būdu I/I adresų registras (I/I AR) nusako tam tikrą I/I įrenginį. I/I buferinis registras (I/I BR) reikalingas tam, kad vyktų keitimasis duomenimis tarp I/I modulis ir CPJ.

Atminties modulį sudaro vadinamųjų ląstelių, kurios pažymimos iš eilės sunumeruotais adresais, visuma. Kiekvienoje ląstelėje yra dvejetainis skaičius, kuris gali būti arba instrukcija, arba duomenys. I/I modulis transportuoja duomenis iš išorinių įrenginių į CPJ ir į atmintį arba atvirkščiai. Jame yra vidiniai buferiai, kuriuose šie duomenys saugomi, kol juos bus galima išsiųsti.

2. 2. Kompiuterio funkcionavimas

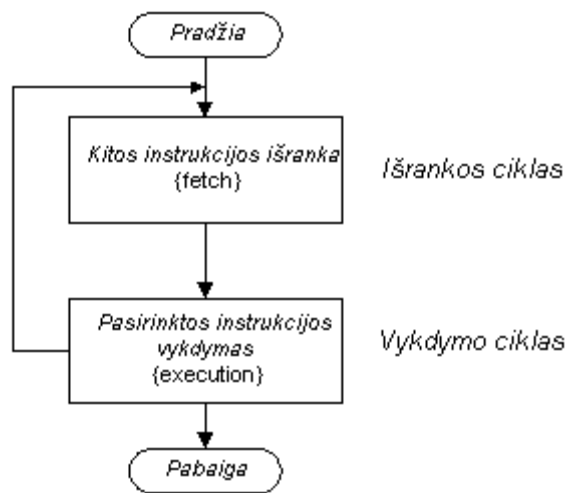
Programos vykdymas – pagrindinė kompiuterio funkcija. Vykdoma programa – tai visuma atmintyje saugomų instrukcijų. Centrinis procesorinis įrenginys dirba tam tikrą darbą vykdydamas programoje nurodytas instrukcijas.

Kad geriau suprastume šią pagrindinę funkciją, suvoktume, koku būdu vykdant programą sąveikauja pagrindinės kompiuterio komponentės, turime detaliau išnagrinėti patį programos vykdymo procesą. Pats paprasčiausias būdas – tarti, kad instrukcija apdorojama per du žingsnius: vienu laiko momentu CPJ skaito (*išrenka*) {fetches} instrukciją iš atminties, o kitu – ją vykdo.

Taigi programos vykdymo procesą sudaro pasikartojantis instrukcijų išrankos bei instrukcijų vykdymo procesas. Suprantama, patį instrukcijos vykdymo procesą tai pat gali sudaryti keli smulkesni žingsniai. Šiuo etapu apsiribosime tuo, kad instrukcijos apdorojimas dviejų stadijų – išrankos ir vykdymo:

- Išranka yra ta pati operacija visų instrukcijų ir vyksta skaitant jas atminties ląstelėse.
- Instrukcija, priklausomai nuo jos pobūdžio, gali būti vykdoma viena arba keliomis operacijomis.

Tam tikros instrukcijos apdorojimas vadinamas *instrukcijos ciklu*. 2.3 pav. parodytas ką tik išnagrinėtas supaprastintas dvižingsnis instrukcijos ciklas. Žingsniai atitinkamai pavadinti *išrankos* ir *vykdymo ciklais*. Programos vykdymas sustabdomas tik tuo atveju, jei skaičiavimo mašina išjunginama, atsiranda kažkokia neištaisoma klaida arba programoje aptinkama instrukcija, kad kompiuteris turi būti sustabdytas.



2.3 pav. Pagrindinis supaprastintas instrukcijos ciklas

2.2.1. Išrankos ir vykdymo ciklai

Kiekvieno instrukcijos ciklo pradžioje CPJ instrukciją išrenka iš atminties. Nustatant, kuri instrukcija bus išrenkama toliau, tipiškajame CPJ taikomas programos skaitiklio (PS) pramintas registras. Nekalbant apie specialius atvejus, CPJ po kiekvienos išrankos padidina PS vienetu ir taip gali išrinkti kitą sekos instrukciją (t. y. instrukciją, kurią rodo kitas, aukštesnis, atminties ląstelės adresas).

Išrinktoji instrukcija siunčiama į CPJ registrą, kuris žinomas kaip instrukcijų registras (IR). Instrukcijos pateikiamos dvejetainiu kodu, nurodančiu, kokį veiksmą turi daryti CPJ. CPJ interpretuoja instrukciją ir vykdo būtiną veiksmą. Bendroju atveju šie veiksmai suskirstomi į keturias kategorijas:

- *CPJ - Atmintis.* Duomenys gali būti siunčiami iš CPJ į atmintį arba iš atminties į CPJ.
- *CPJ - I/O.* Duomenys gali būti siunčiami į išorinę aplinką arba iš jos, juos transliuojant tarp CPJ ir I/O modulio.
- *Duomenų apdorojimas.* CPJ su duomenimis gali vykdyti tam tikras aritmetines ir logines operacijas.
- *Valdymas.* Tam tikra instrukcija gali nurodyti, kad vykdomoji seka pakeičiama (pvz., JUMP# instrukcija).

2.2.2. Įvesties/išvesties funkcija

Kompiuterio, kontroliuojamo CPJ, veikimą jau aptarėme, apžvelgėme ir CPJ bei atminties sąveiką. I/O komponentės reikšmė tik paminėta. Detaliau tai bus nagrinėjama vėliau, čia paaiškinsime tik trumpai.

I/O modulis su CPJ duomenimis gali keistis tiesiogiai. Lygiai taip pat kaip CPJ, specifikuojant tam tikrą adresą, gali inicijuoti skaitymą arba rašymą iš atminties arba į ją, CPJ taip pat gali duomenis skaityti arba rašyti iš – arba į I/O modulį. Pastaruoju atveju CPJ identifikuoja tam tikrą įrenginį, valdomą specialaus I/O modulio. Tam tikrais atvejais labai naudinga užtikrinti tiesioginę I/O ir atminties sąveiką. Tada CPJ „suteikia“ I/O moduliui „teisę“ skaityti iš – arba rašyti į atmintį taip, kad I/O ↔ atminties duomenų siunta vyktų nedalyvaujant pačiam CPJ. Vykstant tokioms siuntoms I/O modulis perduoda atminčiai skaitymo arba rašymo komandas ir palengvina CPJ darbą keičiantis duomenimis. Tokia operacija vadinama *tiesiogine kreiptimi į atmintį* {Direct Memory Access – DMA} ir detaliau bus nagrinėjama vėliau. Kol kas pakanka žinoti kompiuterio vidinių ryšių struktūrą vykstant tiesioginei I/O ir atminties sąveikai.

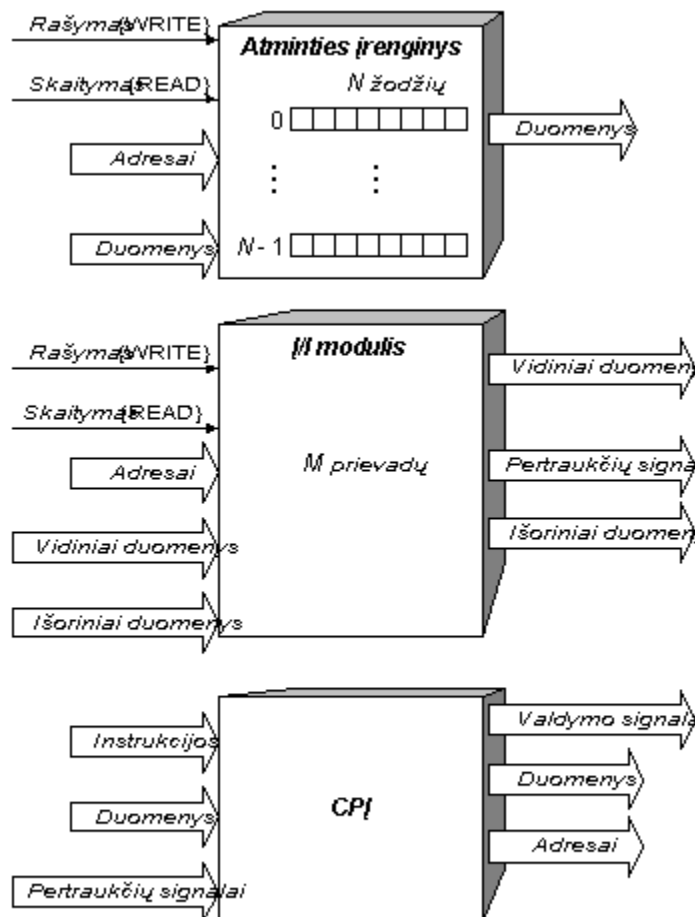
2. 3. Kompiuterio komponentių tarpusavio ryšių struktūra

Kompiuterį sudaro trys pagrindinės tarpusavyje sąveikaujančios komponentės arba moduliai (CPI, atmintis, I/I). Pagal veikimą kompiuterį galima traktuoti kaip tam tikrą pagrindinių modulių tinklą. Taigi turi būti „keliai“, kurie jungtų šiuos modulius.

Tokių duomenų kelių, jungiančių įvairius modulius, visuma vadinama *vidine sąryšine struktūra*. Šios struktūros projektavimas priklauso nuo sąveikos tarp modulių būtinumo.

2.4 pav. pateikti įvairūs keitimosi duomenimis būdai, apibrėžiantys kiekvieno modulio pagrindines įvesties ir išvesties formas:

- *Atmintis*. Paprastai atminties modulis susideda iš N vienodo ilgio žodžių. Kiekvienas žodis ženklina unikalų skaitmeninį adresą (0, 1, ..., $N - 1$). Duomenų žodis gali būti nuskaitytas iš atminties arba įrašytas į ją. Konkreti operacija nurodoma READ arba WRITE valdymo signalais. Operacijos vykdymo vieta specifikuojama adresu.
- *I/I modulis*. Stebint kompiuterio sistemą iš vidaus I/I funkcionavimas panašus į atminties funkcionavimą. Čia taip pat vykdomos dvi operacijos – rašymo ir skaitymo. Tačiau I/I modulis gali valdyti daugiau negu vieną išorinį įrenginį. Į kiekvieno išorinio įrenginio interfeisą galima kreiptis kaip į *prievadą* {port} su unikaliu numeriu (pvz., 0, 1, ..., $M - 1$). Ir dar – I/I modulis gali siųsti pertraukties signalus į CPI.
- *CPI*. CPI skaito instrukcijas ir duomenis iš išorės, jas apdorojęs rašo duomenis išorėn, be to, gaudamas valdymo signalus valdo visas kompiuterio sistemos operacijas. CPI taip pat „priima“ pertraukčių signalus.



2.4 pav. Kompiuterio moduliai ir duomenų srautai

Toliau pateiktas sąrašas nurodo kompiuterio modulių tarpusavio keitimąsi duomenimis. Vidinių ryšių struktūra turi užtikrinti tokius duomenų siuntimo tipus:

- Iš atminties į CPJ. CPJ iš atminties skaito instrukciją arba duomenų vienetą.
- Iš CPJ į atmintį. CPJ duomenų vienetą rašo į atmintį.
- Iš I/O į CPJ. CPJ duomenis skaito iš I/O įrenginio per I/O modulį.
- Iš CPJ į I/O. CPJ siunčia duomenis į I/O įrenginį.
- Iš atminties į I/O arba iš I/O į atmintį. Šiais dviem atvejais I/O modulis duomenimis gali keistis su atmintimi tiesiogiai, taikydamas tiesioginę kreiptį į atmintį {DMA} ir „apeidamas“ CPJ.

2. 4. Kompiuterio magistralės

Magistralė – komunikacijos kelias, jungiantis du (ar daugiau) kompiuterio įtaisus. Pagrindinis magistralės bruožas – bendroji duomenų perdavimo terpė. Magistralė jungia daugybę įtaisų, ir signalas, kurį siunčia vienas įtaisas, gali būti priimtas visų kitų prie magistralės prijungtų įtaisų. Jei du įtaisai vienu metu siųs signalus, jų signalai sutaps laike ir bus iškreipti. Taigi tam tikru laiko momentu signalus turi siųsti tik vienas įtaisas.

Daugeliu atveju magistralę sudaro daugybė komunikacijos kelių arba linijų. Kiekviena linija signalai siunčiami dvejetainės formos. Linijas sujungus į visumą, dvejetainiai skaičiai gali būti siunčiami vienu metu (t. y. lygiagrečiai).

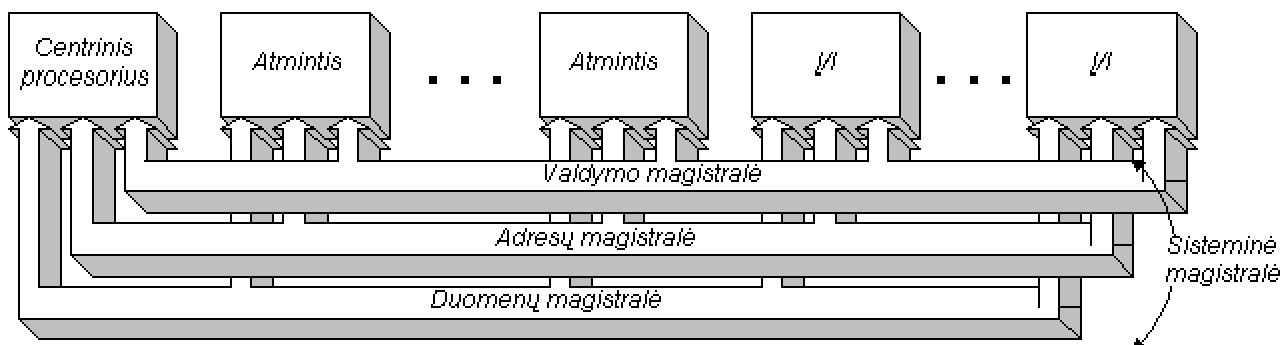
Kompiuteryje yra kelios skirtingos magistralės, užtikrinančios ryšį tarp įvairių skirtingų kompiuterio hierarchijos struktūros lygių komponentų. Magistralė, jungianti pagrindines kompiuterio komponentes (CPJ, atmintį, I/O), vadinama *sisteminė magistrale*. Apibendrintą kompiuterio jungimų tarpusavio struktūrą sudaro viena arba kelios sisteminės magistralės.

2.4.1. Magistralės struktūra

Paprastai sisteminę magistralę sudaro nuo 50 iki 100 laidininkų. Kiekvienas laidininkas atlieka skirtingą funkciją. Nepaisant to, kad yra daug magistralių tipų, kiekvienoje iš jų galima išskirti tris funkcinės laidininkų grupes (2.5 pav.):

- adresų,
- duomenų ir
- valdymo linijos.

Be to, čia gali būti maitinimo linijų, užtikrinančių prie magistralės prijungtų modulių maitinimą.



2.5 pav. Magistralių schema

Duomenų linijomis vyksta keitimasis duomenimis tarp kompiuterio modulių.

Adresų linijos nurodo duomenų magistralėje esančios informacijos šaltinį ir imtuvą (paskirties įrenginį {destination}).

Valdymo linijos kontroliuoja kreiptis {access} į duomenų ir adresų linijas ir šių linijų naudojimą.

Duomenų ir adresų linijos yra bendros {shared} visoms kompiuterio komponentėms, todėl turi būti numatytas būdas jas valdyti. Valdymo signalais siunčiama ir komandinė, ir sinchronizavimo {timing} informacija tarp kompiuterio modulių. Sinchronizavimo signalai rodo, kad duomenys ir adresai yra teisingi (nusistovėjo). Komandų signalai nurodo, kokios operacijos turi būti vykdomos.

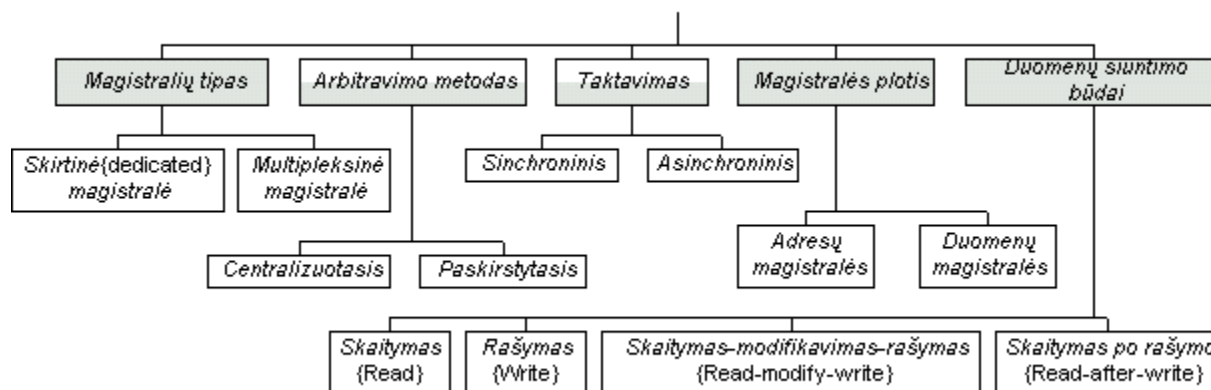
2.4.2. Magistralių hierarchija

Jungiant į magistralę daugiau įrenginių nukenčia jos pajėgumas. Tai lemia dvi priežastys:

1. Kuo daugiau įrenginių jungiama į magistralę, tuo didesnė signalų delsa. Delsą lemia laikas, per kurį tam tikras įrenginys koordinuoja naudojimąsi magistrale. Kai magistralės valdymas dažnai pereina nuo vieno įrenginio kitam, ši delsa gali labai paveikti našumą.
2. Magistralė gali tapti kompiuterio silpnąja vieta {bottleneck}, jeigu keitimosi duomenimis intensyvumas viršys magistralės galimybes. Šią problemą iš dalies galima įveikti didinant duomenų siuntimo intensyvumą ir taikant platesnes magistras (pvz., išplėtus duomenų magistralę iki 32 arba 64 bitų). Tačiau keitimosi duomenimis, kurį generuoja į magistralę įjungti įrenginiai (t. y. vaizdo, grafiniai valdikliai, tinklų įranga), tempai labai spartėja ir galiausiai nebebus užtikrinamas atitinkamas našumas.

Dauguma kompiuterizuotų sistemų naudoja kelias magistras. Yra tam tikra jų hierarchija. 2.6 pav. parodyta tipiška kompiuterio su magistralėmis struktūra.

Magistralių projektavimo elementai



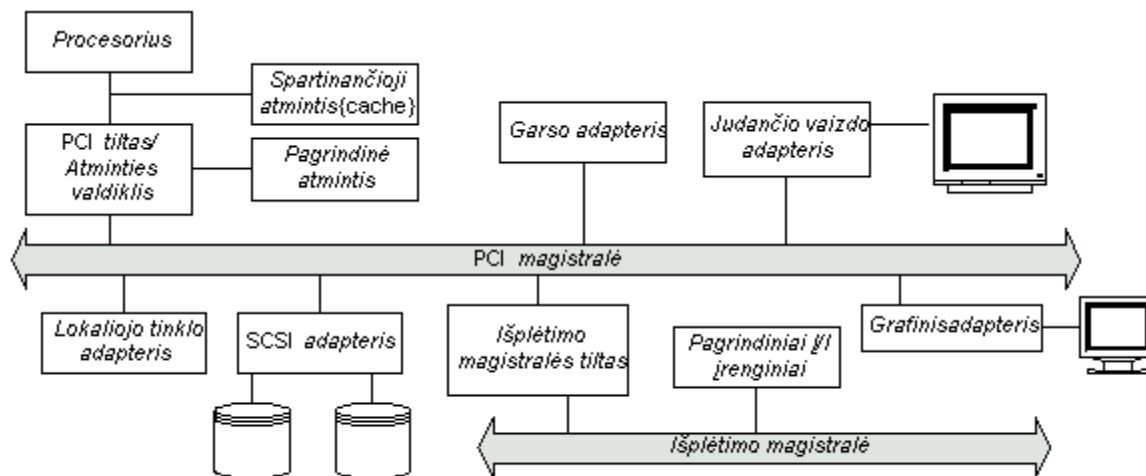
2.6 pav. Magistralių projektavimo elementai

2.4.3. Magistralės projektavimo elementai

Nepaisant to, kad magistralių fizinis realizavimas labai skirtingas, yra keli pagrindiniai parametrai, arba projektavimo elementai, pagal kuriuos magistras galima klasifikuoti. Šie elementai pateikti 2.18 pav.

2.4.4. PCI magistralė

Periferinių komponentių sujungimo {Peripheral Component Interconnect – PCI} magistralė yra didelio pralaidumo, nepriklausoma nuo procesoriaus, galinti veikti kaip mezoninė, arba periferinė (išplėtimo), magistralė. Palyginti su kitomis išplėtimo magistralėmis, PCI geriau pritaikyta spartiesiems įvesties/išvesties posistemiams (t. y. grafiniams ir tinklo adapteriams, diskiniams valdikliams ir t. t.). Pagal esamą standartą duomenis galima siųsti 64 duomenų linijomis, kurių taktinis dažnis 33 MHz, t. y. 264 MB/s arba $64 \times 33 = 2,112$ Gb/s (PCI 2.1 versija leidžia magistralės taktinį dažnį padidinti iki 66 MHz, jei tai užtikrina visi jos abonentai). Tačiau tai teorinis maksimalus greitis. Didelis pralaidumas nėra PCI magistralės vienintelis privalumas. PCI taip pat atitinka modernių įvesties/išvesties sistemų ekonomiško reikalavimus; pereiti iš PCI į kitas magistras galima taikant tik kelias mikroschemas.



2.7 pav. PCI magistralės konfigūravimo pavyzdys

„Intel“ kompanija PCI magistralę pradėjo kurti 1990 m. kompiuteriams su Pentium® procesoriais. Dabartinė PCI versija – PCI 2.0 sukurta 1993 m.

2.7 pav. parodytas tipiškas PCI magistralės pritaikymas vienprocesorėje kompiuterizuotoje sistemoje. Čia kombinuotas pagrindinės atminties valdiklis ir tiltas į PCI magistralę užtikrina glaudų ryšį su procesoriais ir galimybę siųsti duomenis didele sparta. Tiltas veikia kaip savotiškas duomenų buferis, todėl PCI magistralės sparta gali žymiai skirtis nuo procesoriaus I/O pajėgumo.

Papildoma literatūra apie sistemes magistrales

Kaip nebūtų keista, literatūros apie magistralinius abipusius ryšius nėra daug. [1] išsamiai analizuojamos magistralių struktūros ir duomenų siuntimas magistralėmis taip pat pateikti specifinių magistralių pavyzdžiai. PCI magistralė labai gerai aprašyta [3]. Gausu informacijos apie PCI magistralę [4] ir [5]. Labai populiarios SCSI ir USB kompiuterio magistralės aprašytos [6] bei [7].

1. **N.Alexandridis.** Design of microprocessor-based systems / Englewood Cliffs, New York: Prentice-Hall, ASIN: 0135885671 1993.
2. Computer System Bus (IEEE Standard, No 796-1983/Sh09001) / IEEE Computer Society; ISBN: 9997873637. (June 1983)
3. **T.Shanley, D.Anderson.** PCI system architecture 4th edition / Addison-Wesley Pub Co; ISBN: 0201309742 . 1999, 787 pages.
4. **E.Solari.** PCI Hardware and Software 5th Edition / The Coriolis Group; ISBN: 0929392639. 2000.
5. **T. Shanley, K.Gettman.** PCI-X System Architecture (With CD-ROM) / Addison-Wesley Pub Co; ISBN: 0201726823. 2001, 688 p.
6. **G.Field, P.M.Ridge.** The Book of SCSI: I/O for the New Millennium / No Starch Pr; ISBN: 1886411107. 2000, 428 p.
7. **D.Anderson, Inc.MindShare.** Usb System Architecture (PC System Architecture Series) / Addison-Wesley Pub Co; ISBN: 0201461374. 1997, 321 p.