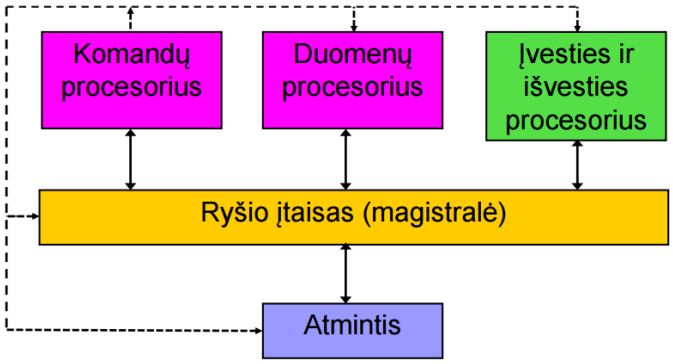
Konspektas kompiuterių architektūros egzaminui

I DALIS. PROCESORIAI

1. Fon Neimano ir Harvardo architektūros. Procesoriaus struktūra.

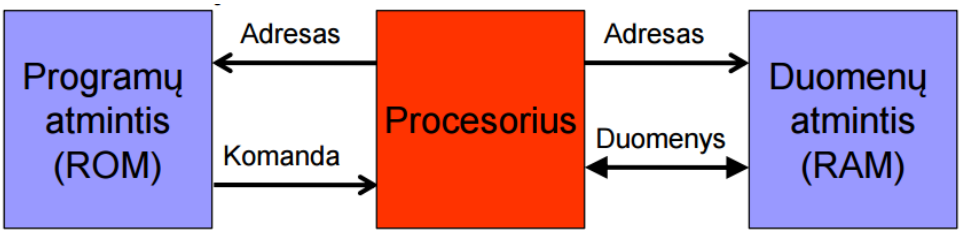
Fon Neimano: Čia priimta, kad visi informacijos mainai tarp kompiuterio įtaisų vykdomi per **vienintelę magistralę**. Komandų ir duomenų procesorius įprasta apjungti į vieną įtaisą - centrinį procesorių (CPU). Atmintis (ROM + RAM).



Neumano architektūrą sudaro tokie įtaisai:
1) **CPU** (komandų proc. + duomenų proc.), kuris interpretuoja ir vykdo programos komandas (išskyrus informacijos įvedimo ir išvedimo komandas);
2) **atmintis**, kurioje saugomi visi duomenys ir programos (**RAM + ROM**);
3) **įvesties ir išvesties įtaisas** - užtikrina kompiuterio ryšį su aplinka;
4) **magistralė**, kuri užtikrina informacijos mainus tarp visų kompiuterio įtaisų.

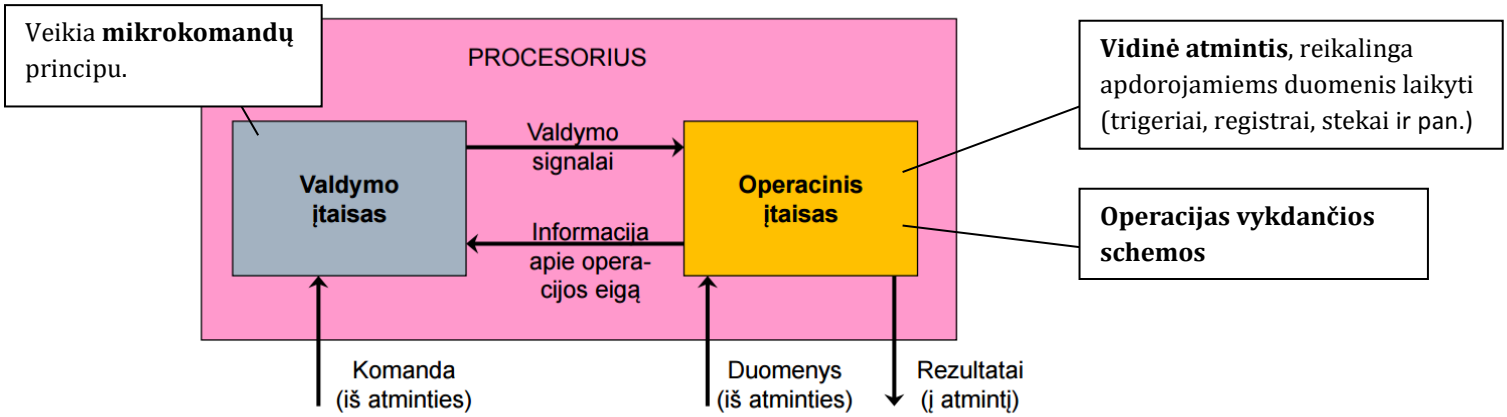
Fon Neimano principai:
- **Dvejetainio kodavimo principas**
- **Programinio valdymo principas**
- **Vienalytės atminties principas**
- **Informacijos adresavimo principas**

Harvardo: Čia duomenys ir programos saugomi atskiruose atminties įtaisuose: **ROM ir RAM**



Procesoriaus struktūra.

Dabar **komandų ir duomenų procesorius** įprasta apjungti į vieną įtaisą – **centrinį procesorių (CPU)**. Įprasta vadinti taip: komandų procesorius - valdymo įtaisas, duomenų procesorius - operacinis įtaisas.



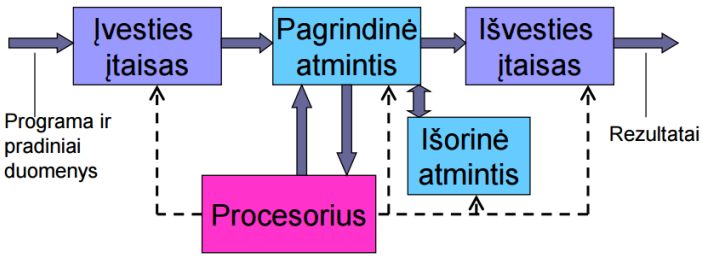
• **valdymo įtaisas** išrenka iš atminties *komandą*, ją analizuoja ir valdo *operacinio įtaiso darbą* (jame vykdomas operacijas, kreipinius į atmintį duomenims išrinkti ar rezultatui įrašyti);
• **operacinis įtaisas** vykdo *operaciją*, kurią nurodo *komanda*;

Šie du įtaisai dirba kartu: valdymo įtaisas pagal operacijos kodą formuoja signalus, valdančius operacinio įtaiso darbą; pastarasis perduoda į valdymo įtaisą signalus, informuojančius apie operacijos eigą, nuo kurių gali priklausyti paskesnių valdymo įtaiso signalų formavimas.

2. *Kompiuterių struktūrų raida*

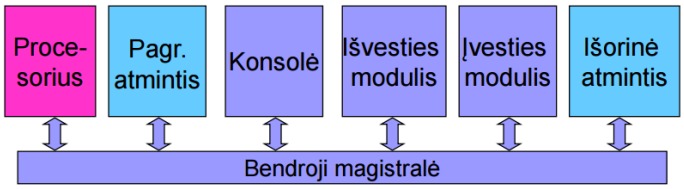
I karta

Procesorius – kartu ir centrinis valdymo įtaisas: jis ne tik interpretuoja programą ir vykdo komandas, bet ir *valdo pradinių duomenų įvedimą bei rezultatų išvedimą*. Procesorius daro viską, yra tiesiogiai sujungtas su viskuo.



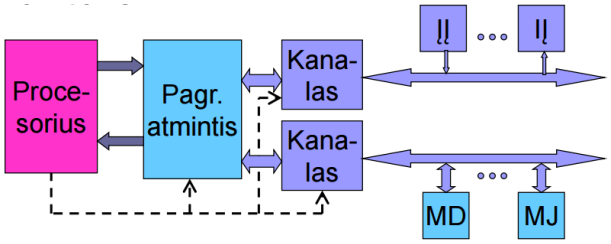
II karta

Čia **visi sistemos elementai sujungti tarpusavyje *bendra magistrale*.**
Lengva prijungti papildomus modulius, tačiau bendra magistralė – siaura sistemos vieta, ribojanti sistemos našumą.



III karta

Čia informacijos apdorojimo procesas *atskirtas* nuo jos įvedimo ir išvedimo. Duomenų įvedimą bei rezultatų išvedimą (taip pat ir mainus su išorine atmintimi) valdo specialūs įtaisai, vadinami kanalais arba įvesties ir išvesties procesoriais.
Procesorius interpretuoja programą ir vykdo komandas bei valdo kanalų darbą.



3. Komandų sistema. Pagrindiniai komandų sistemų tipai

Skaičiavimams reikalingą operacijų rinkinį kompiuteryje realizuoja *komandų sistema*, kurioje vieną operaciją gali atitikti ir kelios komandos, besiskiriančios operandų tipais, adresacijos būdais ir pan. Komandų sistemos apima tokias komandų grupes:

- **Aritmetinės ir loginės** (sveikųjų skaičių: +, –, *, /; loginės: IR, ARBA),
- **Duomenų persiuntimo** (registrų įkrovimas, įrašymas į atmintį),
- **Valdymo** (perėjimai, procedūrų iškvietimas ir grįžimas, ciklai)
- **Slankaus kablelio** (slankaus kablelio skaičių: +, –, *, /)
- **Dešimtainės** (dešimtainių skaičių: +, –, *, /)
- **Eilučių apdorojimo** (eilučių persiuntimas, palyginimas, paieška)
- **Sisteminės** (OS iškvietimas, virtualios atminties valdymas).

Pagrindiniai komandų sistemų tipai:

- Stekas;
- Akumuliatorius;
- Registrų grupė;

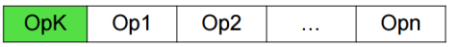
Stekas – specialiai suprojektuota atmintis, į kurią informacija įrašoma nuosekliai, iš jos informacija perskaitoma taip pat tik nuosekliai. Informacijos vietą steke rodo steko rodyklė (stack pointer).

Akumuliatorius: *vienas iš operandų* saugomas *akumuliatoriuje*, rezultatas – taip pat, operandų išrinkimas ir įrašymas – *load, store.*
Akumuliatorius: specialus registras (kaupiantysis registras), į kurį informacija įrašoma prieš operaciją; iš jo rezultatai grąžinami į atmintį.

Registrų grupė: *operandai* saugomi *registruose*, rezultatas – taip pat, operandų išrinkimas ir įrašymas – *load, store*.
Reg-Atm tipo architektūra. Registrų grupė procesoriuje, į kuriuos vienas iš operandų *įrašomas prieš operaciją*; kitas operandas imamas *tiesiai iš atminties*.
Load-store tipo architektūra. Čia taip naudojami registrai, į kuriuos prieš operaciją turi būti įrašyti *abu operandai*.

4. Komandų formatai. Operandų nuorodos

Bendruoju atveju procesorių komandų formatai gali būti pavaizduoti taip:



Čia: **OpK** – operacijos kodo laukas, **Op1, ..., Opn** – operandų nuorodos.

Tarkime turime kodo eilutes:
LOAD R1, A;
ADD R3, R1, B.
Tai LOAD ir ADD - operacijos pavadinimas (kodas), o R1, R3, B - operandų nuorodos.

Operando nuorodos pavidalas priklauso nuo to, koks yra operandas, kur jis saugomas.

Operandu gali būti:
- kurio nors registro turinys,
- atminties ląstelės turinys,
- konstanta,
- adresas,
- sąlyga.

Nuo to priklauso operando nuorodai skirto lauko (laukų) ilgis.

5. Operandų adresavimo būdai

M[A] žymi atminties ląstelę, kurios adresas A, Reg[n] žymi registrą, kurio numeris n.

1. Registrinė adresacija – operandas yra registre, kurio numeris nurodytas komandoje pirmiau, pavyzdžiui: ***Add R4, R1*** ***Reg[4] := Reg[4] + Reg[1]***

2. Netiesioginė registrinė – operando adresas yra registre, kurio numeris nurodytas komandoje, pavyzdžiui: ***Add R4, (R1)*** ***R4 := R4 + M[R1]***

3. Santykinė (bazinė arba su poslinkiu) – operando adresas paskaičiuojamas prie registro turinio pridedant nurodytą poslinkio reikšmę. Pavyzdžiui: ***Add R4, 9(R1)*** ***R4 := R4 + M[R1+9]***

4. Indeksinė – operando adresas paskaičiuojamas sumuojant bazės ir indekso registrų turinius; tinka masyvo elementams išrinkti (R1 – masyvo bazinis adresas, R2 – indekso reikšmė). Pavyzdžiui:

Add R4, (R1+R2) ***R4 := R4 + M[R1 +R2]***

4a. Indeksinė su koeficientu – nuo indeksinės skiriasi tik tuo, kad indekso reikšmė padauginama iš koeficiento (m), atitinkančio duomenų elemento ilgį. Pavyzdžiui:

Add R4, 8(R1) [R2] ***R4 := R4 + M[8+R1 +R2*m]***

5. Tiesioginė arba absoliutinė – adreso reikšmė (pilna ar jo dalis) nurodoma komandoje. Pavyzdžiui*:* ***Add R4, (704)*** ***R4 := R4 + M[704]***

6. Betarpiškoji – operandas yra komandos formato dalis. Tai gali būti interpretuojama, kaip adreso nuoroda yra komandų skaitiklyje. Pavyzdžiui*:* ***Add R4, #7*** ***R4 := R4 + 7***

7a. Autoinkrementinė – operando adresas yra registre, kurio turinys automatiškai didinamas, pavyzdžiui:

Add R4, (R1)+ ***R4 := R4 + M[R1]***

R1 := R1 + d

7b. Autodekrementinė – operando adresas yra registre, kurio turinys automatiškai didinamas, pavyzdžiui:

Add R4, –(R1) ***R4 := R4 + M[R1]***

R1 := R1 – d

Patogu apdorojant masyvus. Pradžios adresas įrašomas į lauke R1 nurodytą registrą.

6. Komandų sistemų išplėtimai procesoriuose. Multimedijos informacijos apdorojimo specifika

Papildymo priežastys ir prielaidos:

- procesoriai buvo orientuoti į *sveikųjų ir slankaus* taško skaičių apdorojimą,
- grafinės ir audio informacijos skaitmeninio apdorojimo plitimas,
- procesorių žodžio ilgio didinimas *nuo 32 bitų iki 64 bitų*,
- daugeliu atvejų skaitmeninei *grafinei ir audio informacijai koduoti pakanka 16 ar net 8 bitų,*
- technologijos vystymasis ir normos mažinimas nuo 0,35 μm iki 0,13 μm leido gerokai padidinti tranzistorių skaičių kristale,
- RISC (centrinių procesorių architektūra, pasižyminti paprastesne komandų sistema) architektūros branduolys kompaktiškas - realizuojamas palyginti nedideliu tranzistorių skaičiumi,
- SIMD (Single Instruction – Multiple Data) ir vektorinio principų panaudojimo galimybės.

1996 metais Intel įvedė *MMX (MultiMedia eXtention) technologiją* - savo **procesorių komandų sistemą *papildė*** *57 naujomis komandomis*, skirtomis *multimedijos programoms optimizuoti*. Šios komandos duomenis traktuoja taip, kaip tai priimta SIMD sistemose (Single Instruction – Multiple Data)

Panašūs *komandų sistemų papildymai* pasirodė ir kitų firmų procesoriuose.

7. Konvejerio ir superkonvejerio esmė

Konvejerizacija reiškia, kad sudėtingi veiksmai (šiuo atveju – komandos vykdymas) **suskaidomi į etapus**, kurių trukmė yra maža.

Kiekvienas etapas – *atskirose schemose (blokuose) atliekami veiksmai*. Jų trukmė atitinka CPU taktą.

Konvejeris:

Esant nuosekliam komandų vykdymui, i+1 komanda pradedama vykdyti tik *po to*, kai baigiama vykdyti i-toji komanda.

Superkonvejeris:

Pradžioje dauguma procesorių turėjo dviejų – penkių pakopų konvejerius.
Siekiant sumažinti takto trukmę, ***pakopų skaičius buvo didinamas*** *ir dabar viršija 10* (juo daugiau pakopų, tuo kiekviena iš jų paprastesnė, todėl gali būti įvykdyta per trumpesnį laiką). Tokie procesoriai vadinami **superkonvejerizuotais**.

8. Konvejerio kliūčių tipai. Kliūčių įveikimas

Konvejerio darbas iš tikro nėra toks idealus, kaip anksčiau buvo vaizduota. Tam trukdo kliūtys, kurias galima skirstyti į tris tipus:

- **struktūrinės kliūtys**, atsirandančios dėl to, kad resursai gali būti nepakankami įvairių pakopų ir komandų etapų reikmėms tenkinti;
- **duomenų kliūtys**, atsirandančios dėl to, kad vienos komandos vykdymo rezultatai naudojami kitoje komandoje kaip operandai;
- **valdymo kliūtys**, atsirandančios dėl būtinumo išrinkti komandas iš kitos vietos, nei tai atliekama jas vykdant nuosekliai.

Struktūrinių kliūčių problema sprendžiama:
- pridedant papildomų resursų;

Duomenų kliūčių problema sprendžiama:
- pakeičiant komandų vykdymo eilės tvarką;
- trumpam blokuojant konvejerio darbą;
- pridedant linijas, sutrumpinančias rezultatui kelią iki vykdymo įtaisų;
- įterpiant NOOP.

Valdymo kliūčių problema sprendžiama:
- pakeičiant komandų vykdymo eilės tvarką;
- naudojant perėjimo prognozę.

9. Našumas, jo įvertinimas. Technologijos išvystymo įtaka našumui

Kompiuterio našumą įtakojantys faktoriai:

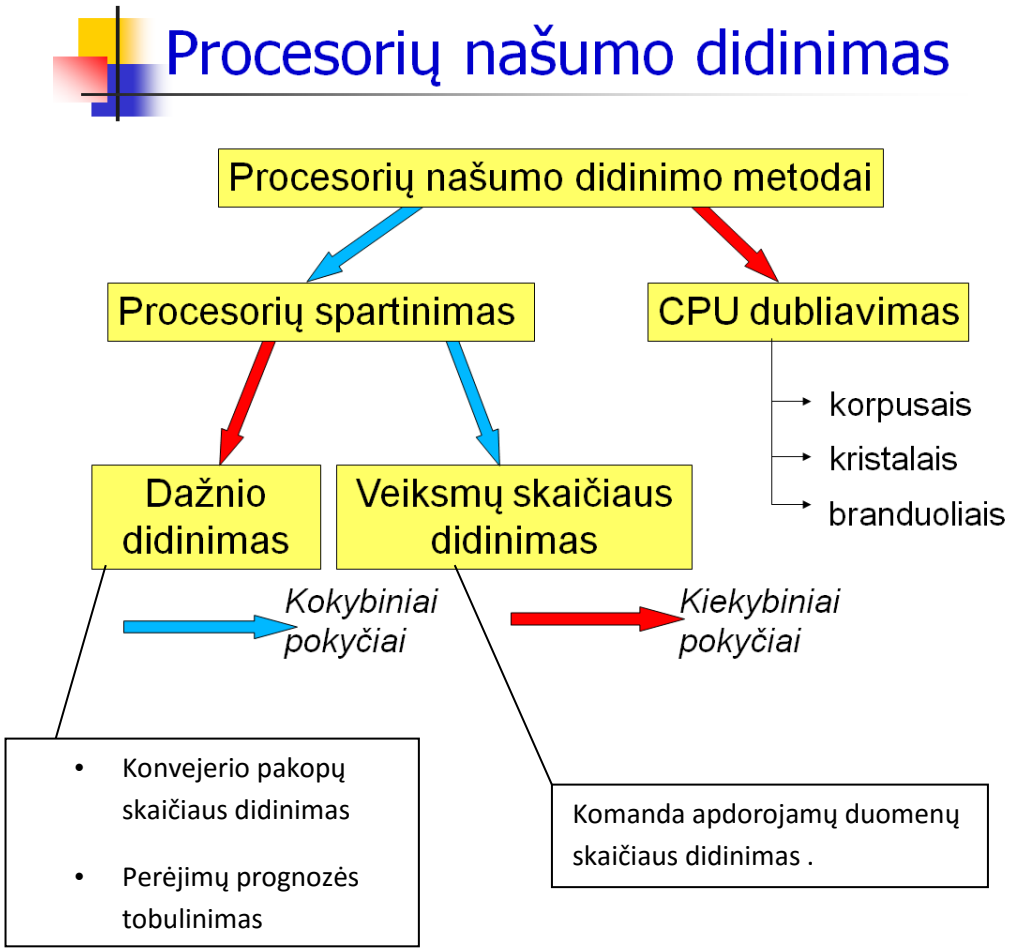
- **CPU sparta**: Kaip greitai dirba CPU apsprendžia jo laikrodžio taktinis dažnis. Kuo didesnis šis dažnis, tuo greičiau dirba visas kompiuteris.
- **RAM dydis**: Galioja taisyklė, kuri tvirtina, kad kuo kompiuteryje didesnė RAM tipo atmintinė, tuo greičiau jis dirba.
- **Standžiojo disko greitis ir talpa**
- **Laisva standžiojo disko erdvė**
- **Kelių uždavinių sprendimas vienu metu:** Windows aplinka yra pritaikyta spręsti daugeliui uždavinių vienu metu. Kuo daugiau uždavinių sistema spręs vienu metu, tuo lėčiau bus sprendžiamas kiekvienas iš tų uždavinių atskirai.

Našumą padidino procesoriai, turintys kelius branduolius, konvejerizacija.

Našumas yra atvirkščiai proporcingas uždavinio sprendimo laikui.

Našumą galima didinti dviem būdais:

- Didinant darbo dažnį,
- Tobulinant procesoriaus architektūrą.



10. Komandų vykdymas ne eilės tvarka. Registrų pakeitimas

Procesoriai, kurie turi keletą funkcinių įtaisų, vadinami **superskaliariniais**. Tokiuose procesoriuose *vienu metu* gali būti pradėtos vykdyti *kelios komandos*.

Kompiliatoriaus suformuota komandų seka programoje ne visuomet bus palanki kelių funkcinių įtaisų aprūpinimui darbu. Programos vykdymo metu šiuolaikiniuose procesoriuose esančios specialios schemos *analizuoja didelį vykdymo eilėje laukiančių komandų skaičių ir* ***dinamiškai*** *parenka vykdymui tas komandas, kurios konkrečiu momentu galėtų būti vykdomos, nors prieš jas eilėje yra kitos komandos.*

Registrų pakeitimas:
Tegul programoje turime tokias dvi komandas:
k1 : add ..., r2 , ...; [... ← (r2) + (…)]
k2 : mult r2 , ..., …; [r2 ← (...) + (…)]
Antroji komanda pakeičia registro r2 turinį nauju. Aišku, kad komandos k2 negalima vykdyti iki tol, kol k1 nepanaudojo buvusio jo turinio.
Jei procesorius turi "atsarginių" registrų, vieną iš jų (pavyzdžiui, r33) panaudokime komandoje k2 :
k1 : add ..., r2 , ...; [... ← (r2) + (…)]
k2 : mult r33, ..., …; [r33 ← (...) + (…)]
Taip konfliktas bus išspręstas. Aišku, kad ir paskesnės komandos turi kreiptis į r33 vietoj r2 (kol registro pakeitimas galioja).
Šiuolaikiniuose procesoriuose "atsarginių" registrų skaičius gerokai didesnis nei adresuojamų.

11. Perėjimų spėjimas. Sudėtingų sistemų realizacija viename kristale

Konvejerio delsai, vykdant perėjimo komandas, *sumažinti* procesoriuose įvestos **perėjimų prognozės schemos. Jos „spėja"** ***perėjimo kryptį***. Jei spėjimas pasitvirtina, konvejeris sėkmingai dirba (bent jau su minimaliais laiko nuostoliais). Jei spėjimas nepasitvirtina, tenka stabdyti konvejerį ir jį užpildyti komandomis iš tikrosios krypties. Juo tikslesnis perėjimų spėjimas, tuo mažiau laiko sugaištama konvejeriui perkrauti.

Perėjimų spėjimas gali būti statinis arba dinaminis:

♠ **Statinė prognozė** remiasi kai kuriais pastebėtais perėjimo komandų elgesio dėsningumais: analizuojamas operacijos kodas, poslinkio ženklas, ir pagal tai priimamas sprendimas apie tikėtiną perėjimą.

♠ **Dinaminė prognozė**: perėjimo krypties spėjimui panaudojama programos vykdymo metu sukaupta informacija apie anksčiau vykdytos komandos elgesį (pereita ar ne).

BHT **- Branch History Table** – paprasčiausia dinaminės prognozės priemonė. Prognozės efektyvumui padidinti kiekvienam perėjimui skiriami du bitai. Tuomet prognozuojama kryptis pakeičiama priešinga buvusiai, jei du kartus iš eilės buvo nepataikyta.

Branch Target Buffer. Kai kuriose prognozės sistemose naudojamas specialus buferis, kuriame laikomi perėjimo adresai (kartu su prognozės bitais).

Sistemos viename kristale

Kristalų gamybos technologijų pasiekimai sudaro galimybes viename kristale integruoti ir kitas kompiuterio sudėtines dalis .

Tai ne tik kompiuterio branduolį daro kompaktiškesniu, bet ir žymiai padidina jo darbo spartą, kadangi duomenų mainai viename kristale vykdomi žymiai sparčiau, nei tarp kristalų.

Tokioms sistemoms apibūdinti naudojamos sąvokos "sistema viename kristale" (Systems On the Chip - SOC).

12. Šiuolaikinių procesorių struktūra (branduoliai, perkodavimas, ...)

13. Koprocesorius ir jo ryšys su procesoriumi

Koprocesorius (angl. *Coprocessor*) – procesorius, *padedantis* kompiuterio centriniam procesoriui atlikti kai kuriuos veiksmus arba pats juos atliekantis. Tokiems veiksmams jis būna specialiai pritaikytas ir juos atlieka sparčiau negu centrinis procesorius.

Fiziškai koprocesorius *gali būti atskira mikroschema arba integruota į centrinį procesorių* (kaip tai daroma matematinius koprocesorius montuojant į procesorius).

Koprocesoriai skirstomi į:

- Bendrosios paskirties **matematinius koprocesorius**, kurie paprastai pagreitina veiksmus su **slankiojo kablelio skaičiais**;
- **Įvesties-išvesties koprocesorius** (pvz., Intel 8089), kurie perima įvesties-išvesties kontrolę iš centrinio procesoriaus arba praplečia procesoriaus standartinę adresų erdvę;
- **Koprocesorius**, skirtus atlikti specifinius algoritmus.

14. Intel procesoriai. Pagrindiniai jų bruožai

- **platus panaudojimo diapazonas – nuo serverių iki mobiliųjų įtaisų**
- **x86 komandų sistema**
- aukštas branduolio našumas realizuojant x86 komandas
- **x86 komandų perkodavimas į RISC (centrinių procesorių architektūra, pasižyminti paprastesne komandų sistema) tipo mikrooperacijas**
- **keli branduoliai**
- **trijų lygių spartinančioji atmintis**
- platus multimedijos komandų diapazonas – nuo MMX iki AVX
- **dviejų gijų palaikymas**

- naujausieji modeliai turi efektyvią integruotąją grafiką (GPU ploto dalis Sandy Bridge 17%, Ivy Brigde – 27%, Hasswell – 31%; plg. AMD Kaveri – 47%)
- Intel turbo boost - reikalui esant dažnio padininimas.

15. ARM procesoriai. Pagrindiniai jų bruožai

- **plačiai naudojami buitinėje elektronikoje** – mobiliuosiuose telefonuose, multimedia grotuvuose, skaičiuotuvuose ir asmeniniuose pagalbikliuose. **Kelių branduolių proc. naudojami** išmaniuosiuose mobiliuosiose telefonuose, planšetiniuose ir internetiniuose kompiuteriuose, namų multimedijos įtaisuose ir net nedideliuose mažai energijos naudojančiuose serveriuose. **Nuo mobiliųjų įtaisų iki nedidelių serverių.**
- Analizuodami programų vykdymą, ARM specialistai padarė išvadą, jog visada naudoti galingus branduolius nėra tikslinga. ***Taip gimė koncepcija big.LITTLE, pagal kurią vienoje procesoriaus mikroschemoje naudojami skirtingi branduoliai – galingi ir ekonomiški***. *Esant nedidelei apkrovai, dirba mažasis klasteris (ekonomiški branduoliai), tuo tarpu galingieji branduoliai yra išjungti. Padidėjus apkrovai, darbą perima didžiojo klasterio branduoliai.*

16. AMD procesoriai. Naujųjų AMD procesorių bruožai

- Amd turbo core - reikalui esant dažnio padininimas
- **AMD iškėlė naują idėją – kurti specialius įtaisus skaičiavimams spartinti, kuriuos jie pavadino APU (Accelerated Processing Units):**
  - ***kompiuteriuose bus naudojami bendrosios paskirties ir specializuoti procesoriai*;**
  - *uždaviniui spręsti bus* ***pasirenkamas*** *procesorius, kuris tai atliks optimaliai našumo, multimedijos informacijos apdorojimo, energetinio efektyvumo ir kainos požiūriu*;
  - programinė įranga išnaudos geriausiai tinkančios tai užduočiai aparatūros privalumus;

2 DALIS – SISTEMOS ARCHITEKTŪRA

1. *Lokališkumo principas. Spartinančioji atmintis. Jos lygiai, eilutės struktūra.*

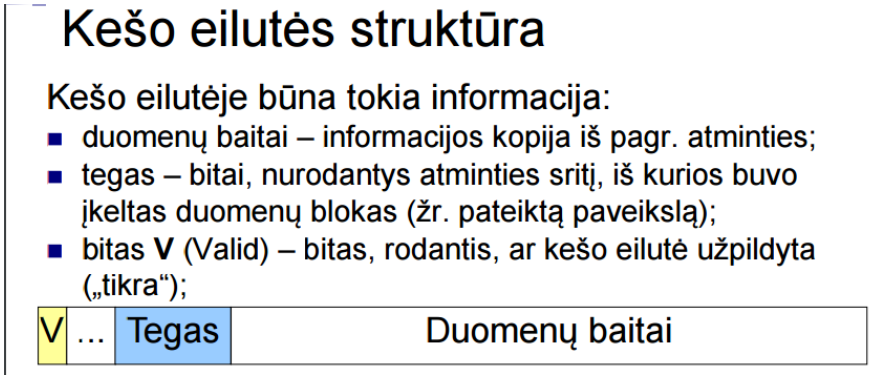
Programos linkę naudoti duomenis ir komandas, kurias jau yra naudoję. Sakoma, kad programa apie 90% vykdymo laiko skiria apie 10% kodo. **Daugelis programų naudoja tą pačią informaciją ar instrukcijas iš naujo ir iš naujo.**

Lokališkumas

- **laiko atžvilgiu:** jei dabar reikalingas kuris nors elementas, labai tikėtina, kad netrukus vėl jo prisireiks;
- **vietos atžvilgiu**: jei dabar reikalingas kuris nors elementas, labai tikėtina, kad netrukus bus reikalingas ir jam gretimas.

Spartinančioji atmintis (kešas) - tai *greita mažo ar vidutinio dydžio atmintis, kuri jungiama tarp procesoriaus ir pagrindinės atminties ir skirta atminties sistemos našumui padidinti*. Kešo koncepcija remiasi tuo, kad labiausiai tikėtini duomenys (prisiminkime lokališkumo principą) kopijuojami iš pagrindinės atminties į specialią greitą atmintį ir šioje saugomi kurį laiką. *Kai procesoriui prireikia duomenų ar komandų, jis pirmiausia jų ieško keše ir tik jų neradęs kreipiasi į pagrindinę atmintį.*
SRAM atmintis vadinama statine todėl, kad atminties elementas – trigeris – gali laikyti būseną kiek norima ilgai (kol jis gauna maitinimą). Vienam bitui saugoti statinės atminties ląstelėje reikia 6-8 tranzistorių (dinaminėje atmintyje pakanka vieno). Todėl SRAM ląstelė užima žymiai didesnį plotą, užtat dirba greičiau nei DRAM.
Dėl didelės darbo spartos SRAM naudojama keše.
Kešo lygiai:

- Sparčiausias (arčiausiai branduolio, mažesnės talpos) – L1 kešas; jo vėlinimas – apie 2-3 ciklus. L1 atmintis(kartais dar vadinama „Primary cache“ (pirminis kešas) – pirmine tarpine atmintimi) veikia tokia pat sparta, kaip ir procesorius. Joje saugomos dažniausiai naudojamos ir pasikartojančios instrukcijos procesoriui.
- Kiek mažesnės spartos (didesnės talpos) – L2 kešas; jo vėlinimas – 7-10 ciklų.
- Lėčiausias – L3 kešas; jo vėlinimas – 20 (lėtuose procesoriuose) – 40 ciklų (sparčiuose procesoriuose).



2. Spartinančiosios atminties tipai (tiesioginio atitikimo, pilnai asociatyvi, dalinai asociatyvi). Informacijos paieškos ir pakeitimo ypatumai.

- **Tiesioginio atitikimo kešas** – kiekvienas iš pagrindinės atminties paimtas eilutės dydžio blokas turi *vienintelę apibrėžtą vietą keše*.
- **Pilnai asociatyvus kešas** – kiekvienas iš pagrindinės atminties paimtas eilutės dydžio blokas *gali būti bet kurioje vietoje keše*.
- **Dalinai asociatyvus kešas** – kiekvienas iš pagrindinės atminties paimtas eilutės dydžio blokas *gali būti bet kurioje iš k vietų*; skaičius *k vadinamas asociatyvumo laipsniu* arba krypčių skaičiumi.

Išrinkimas iš kešo yra atliekamas tokia tvarka:

1. Pagal kreipinio adresą (fizinį!) *nustatomas indeksas*, rodantis kešo eilutę ar jų grupę (kelių krypčių keše).

2. Patikrinama, ar tarp jų yra užpildyta (“*tikra”)* eilutė, kurioje įrašytas tegas sutampa su kreipinio adreso aukščiausiais bitais.

3. Jei tokios eilutės *nėra*, fiksuojamas nepataikymas į kešą (miss). Jei tokia eilutė *rasta*, pagal vietos eilutėje bitus išrenkama reikalinga informacija.

Informacijos pakeitimas keše yra atliekamas tokia tvarka:

1. Pagal kreipinio adresą (fizinį !) *nustatomas indeksas*, rodantis kešo eilutę ar jų grupę (kelių krypčių keše).

2. Patikrinama, ar tarp jų yra laisva *(“netikra”)* eilutė. Jei tokia eilutė **yra**, ji užpildoma iš atminties perskaitytu informacijos (eilutės ilgio) bloku.

3. Jei tokios eilutės **nėra**, naudojant išrinkimo strategiją *parenkama keistina eilutė* ir ji užpildoma iš atminties perskaitytu informacijos (eilutės ilgio) bloku.

3. DDR, DDR2, DDR3 ir DDR4 atmintys. DRAM laiko parametrai.

RAM atmintis – kompiuterio darbinė atmintis, laisvai skaitoma iš bet kurios vietos.

DRAM – dinaminė atmintis, kurioje duomenys saugojami tik trumpą laiką ir nuolat turi būti atnaujinami.

Jei iki SDRAM sukurtieji tipai sąlyginai gali būti jau vadinami klasikiniais, tai pastaraisiais metais sukurti nauji DRAM tipai, kurie prasiskynė kelią į kompiuterius:

- **DDR** - Double Data Rate SDRAM **- SDRAM su dvigubu magistralės dažniu***: skaitymo ir rašymo operacijos vykdomos du kartus vieno takto metu* - pagal kylantį ir krintantį taktinio impulso frontus.
- **DDR2** SDRAM – dvigubai spartesnė nei DDR.

- **DDR3** SDRAM – keturgubai spartesnė nei DDR.
- **DDR4** - dar naujesnė DDR rūšis, kuri ne tik yra greitesnė, bet ir patikimesnė už buvusias.

Pagrindiniai DDR DRAM laiko parametrai yra:

- tRCD - RAS to CAS delay – *laikas*, reikalingas *eilutei išrinkti ir stulpelio adresui paduoti*
- CL - CAS delay (latency) – *laikas*, reikalingas *duomenims iš eilutės išrinkti*
- tRP - RAS precharge – *laikas*, reikalingas išrinktai eilutei *deaktyvuoti*
- tRAS - active to precharge delay – minimalus laikas, kurį eilutė turi būti *aktyvi iki ją deaktyvuojant*

4. DRAM regeneracija ir kontroleriai. DRAM moduliai.

DRAM Informacija saugoma krūvio pavidalu kondensatoriuje, kuris palaipsniui išsikrauna, **todėl ją periodiškai reikia atkurti (regeneruoti).** Skaitymo metu kondensatorius taip pat išsikrauna, todėl jo krūvis taip pat atkuriamas. Dėl to DRAM dirba maždaug 10 kartų lėčiau, nei SRAM.

DRAM regeneracija. Regeneracijos periodas - Tref . Pradžioje Tref buvo 2 ms, dabar siekia 64 ms. Per šį laiką turi būti "sujudinta" kiekviena DRAM eilutė.

DRAM kontroleris, *transformuodamas CPU kreipinio signalų rinkinį į atminčiai valdyti reikalingus signalus*, kartu ir formuoja laiko diagramą pagal konkrečiam DRAM tipui (DDR2, DDR3, ...) reikalingą jų tarpusavio išdėstymą.

5. Lyginumo kontrolė, klaidų kontrolė ir korekcija.

Klaidų kontrolei ir korekcijai naudojamas **Hemingo kodas**.

6. Magnetiniai diskai. SSD. Flash atmintis. Hibridiniai diskai.

Magnetiniai diskai (plokštelės) . Plokštelės yra tikrieji diskai įrenginio viduje, kurie saugo magnetinius duomenis. Į magnetinius diskus įrašoma informacija. Kietasis magnetinis diskas paprastai kalbant yra prietaisas, kuriame kompiuteris **saugo visą informaciją, tai yra operacinę sistemą, naudojamas programas ir kita.**

Magnetinį diską sudaro kietas pagrindas ant kurio užpurškiama magnetinė danga. Disko pagrindas gaminamas iš aliuminio lydinio arba stiklo ir keramikos.

SSD (Solid state drive) – įtaisas, kuris funkcionuoja kaip magnetinis diskas, tačiau neturi judančių dalių.

Privalumai:
- Sistema greičiau paleidžiama – maždaug per 20 sek.
- Nėra mechaninių besisukančių ar judančių dalių;
- Mažas energijos suvartojimas;
- Nėra triukšmo;

• Visiškai vienodas failų skaitymo laikas, nepriklausantis nuo jų vietos ar fragmentavimo;
• Nedideli gabaritai ir svoris.

Flash atmintis (angl. *flash memory*) – išliekamąją atmintį turintis duomenų saugojimo integrinis grandynas, kuriame esantys duomenys gali būti *elektriškai* ištrinti, įrašyti ir nuskaityti; atmintinės rūšis. *Flash* atminties integriniai grandynai dažniausiai naudojami atminties kortelėse, USB atmintukuose, MP3 grotuvuose ir SSD diskuose informacijai saugoti.

Pagrindinės savybės:

- Išliekamoji atmintis – nereikia energijos įrašytos informacijos palaikymui.
- Maža duomenų nuskaitymo trukmė.
- Didesnis atsparumas smūgiams palyginti su kietaisiais diskais.
- Supakavus į kortelę įgyjamas atsparumas spaudimui ir vandens poveikiui.

Hibridinis diskas - Hibridinio disko idėja - vienoje sistemoje ***apjungti tradicinį MD ir SSD***. Įtaisas maždaug du kartus brangesnis nei vidutinio našumo MD. Seagate duomenimis, hibridinio disko našumas apie 20% nusileidžia SSD našumui, bet 30% viršija įprasto MD našumą.

7. Magistralės samprata, jų tipai. Magistralės kompiuteryje. Magistralių arbitražas.

Magistralė – *komunikacijos linija, kuri sujungia keletą kompiuterio įrenginių*. Ja siunčiami duomenys bitų pavidalu (dvejetainiu kodu). Signalas, kurį siunčia vienas iš įrenginių, gali būti priimtas **visų** kitų prie magistralės prijungtų įtaisų. Dažniausiai magistralę sudaro daugybė komunikacijos linijų.

Tipai:

- **lygiagrečiosios magistralės**, kai duomenys perduodami naudojant kiekvienam bitui atskirą liniją.
- **nuosekliosios magistralės,** *kai duomenys perduodami bitas po bito per tą pačią liniją (nuosekliu kodu)*. Nuosekliosios turi didelį linijų skaičių, todėl dažnai skiriamos trys jų sudedamosios dalys: **adresų, duomenų ir valdymo**.

Kompiuterio magistralės:

- **vidinės magistralės**, jungiančios tarpusavyje vidinius kompiuterio komponentus pagrindinėje plokštėje;
- **išorinės magistralės**, jungiančios įvairius kompiuterio išorinius (periferinius) įtaisus prie pagrindinės plokštės.
- **Procesoriaus magistralė**. Ją naudoja valdymo schemų rinkinys (chipset) informacijos mainams su procesoriumi. Kai kurie šaltiniai ją vadina sistemine magistrale.
- **Kešo magistralė**. Procesoriuose ji buvo naudojama dideliam pralaidumui užtikrinti.
- **Atminties magistralė**. Taip vadinama magistralė, jungianti atminties posistemį su valdymo schemų rinkiniu (chipset) ir procesoriumi.

- **Lokalinė I/O magistralė**. Taip vadinama ***didelės spartos*** **įvesties ir išvesties magistralė,** jungianti sparčius I/O įtaisus su atminties posistemiu, su valdymo schemų rinkiniu (chipset) ir procesoriumi. Dabar populiariausia – PCI.
- **Standartinė I/O magistralė**. Tai ***nedidelės spartos*** **įvesties ir išvesties magistralė**, jungianti tokius I/O įtaisus, kaip pelė, klaviatūra. Gera seniems įtaisams prijungti. Populiariausia anksčiau – ISA, dabar - USB.
- **Greitoji grafikos magistralė** (AGP – Accelerated Graphics Port). Taip buvo vadinama didelės spartos magistralė, jungianti grafikos posistemį su valdymo schemų rinkiniu (chipset) ir procesoriumi.
- **Naujoji grafikos magistralė** (PCIe – PCI Express). Taip vadinama didelės spartos magistralė, pakeitusi AGP.

8. HyperTransport, QPI, PCI Express, USB. Valdymo schemų rinkiniai (chipsets)

HyperTransport - **didele sparta pasižyminti tiesioginio ryšio tarp schemų sąsaja** kompiuteriams, serveriams ir kt. įrangai.

OPI(QuickPath Interconnect) - intel procesoriuje Nehalem įvedė QuickPath Interconnect (QPI - vietoj anksčiau naudotos FSB). Ji tai padarė 5 metais vėliau nei AMD, įvedusi HyperTransport. QPI naudoja dvi 20 bitų magistrales (atskiras kiekvienai krypčiai).

PCI Express magistralė –didelės spartos nuoseklaus duomenų perdavimas. Svarbi jos funkcija – galimybė keisti perdavimo spartą keičiant linijų skaičių: x1, x2, x4, x8, x12, x16, x32.

USB - nuosekli magistralė, pasižyminti vidutine perdavimo sparta. Ji **skirta įvairiems periferiniams įtaisams** (klaviatūrai, spausdintuvams, skeneriams, ...) prijungti.

9. Išorinių įtaisų sąsajos (IDE (ATA), SATA ir SCSI)

IDE - Paprasta, primityvi sąsaja. Duomenys perduodami per ISA magistralę, todėl lėtai. Talpa nedidelė, naujuose PC nenaudojama. Apibrėžia duomenų perdavimą tarp procesoriaus ir standžiojo disko. IDE žinomas ir kitu vardu – ATA (AT Attachment).

SATA – nuoseklioji sąsaja. Atminties įrenginių (daugiausia kietųjų diskų) *prijungimo prie kompiuterio* standartas.

SCSI - nuoseklioji sąsaja. Našiausia, diskai jungiami per brangų kontrolerį. Paprastai MD yra aukščiausios kokybės, spartūs. SCSI kontroleris gali valdyti iki 7 MD, kurių talpa < 45GB.

10. Pertraukties esmė. Pertraukčių tipai ir lygiai. Pertraukties procedūra. Pertraukčių kontroleris.

Pertrauktys - tai mechanizmas, kuriuo naudodamiesi įvairūs kompiuterio įrenginiai ir programos atkreipia CPĮ dėmesį, reikalaudami juos aptarnauti.

Tipai:

- Programinės (sinchroniškos procesui):
- programinės (pertr. sistemai tikrinti)
- derinimo (po kiekvienos komandos - analizė)

Vykdymo variantai:

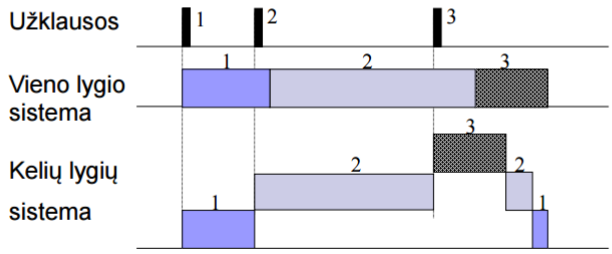
- po komandos (dažniausiai)
- komandoje (rečiau, kai negalima užbaigti):
- puslapio klaida
- ilgai trunkančioms komandoms

Apdorojama:

- mikroprograma (anksčiau)
- paprograme (dažniausiai)

Pertraukčių lygiai

Pertraukčių prioritetai
Aptarnavimo tvarka:



Čia 1 pertrauktis turi žemiausią prioritetą, 3 - aukščiausią

Apdorojimo procedūra:

- pertraukties signalo priėmimas
- pertraukties atpažinimas
- būsenos įsiminimas
- pertrauktį apdorojančios programos vykdymas
- būsenos atstatymas

11. Įvesties ir išvesties problemos bei jų sprendimas. Periferinio įtaiso kontroleris.

Įvesties ir išvesties problemos:

- Galimybė prijungti įvairius PĮ ;
- Lygiagretus įvesties ir išvesties sistemos darbas su procesoriumi;
- Maksimaliai supaprastintas įvesties ir išvesties procesų programavimas;
- Reakcija į įvairias kritines situacijas bei iškilusias problemas.

Problemų sprendimo keliai:

- PĮ moduliškumas (konstruktyvus išbaigtumas, paprastas prijungimas);
- Unifikuoti duomenų formatai;
- Unifikuota sąsaja (interfeisas);
- Unifikuoti komandų formatai ir tipai.

12. Programa valdomi ir tiesioginiai duomenų mainai. Kanalas ir TKA kontroleris

Duomenų mainai su PĮ

Atsižvelgiant į persiunčiamų duomenų kiekį bei PĮ darbo spartą, pasirenkamas vienas iš dviejų mainų tipų:
a) **programa valdomi duomenų mainai** – kai perduodama mažai duomenų kreipiantis į atmintį, ir PĮ darbo sparta nedidelė;
b) **tiesioginiai duomenų mainai** - kai perduodamų duomenų apimtis didelė ar PĮ darbo sparta didelė.

13. Grafikos procesorius ir jo panaudojimas

Grafikos procesorius GPU (Graphics Processing Unit) – pagrindinis elementas, skirtas į ekraną išvedamo vaizdo elementams skaičiuoti bei trimatės (3D) grafikos komandoms vykdyti.