

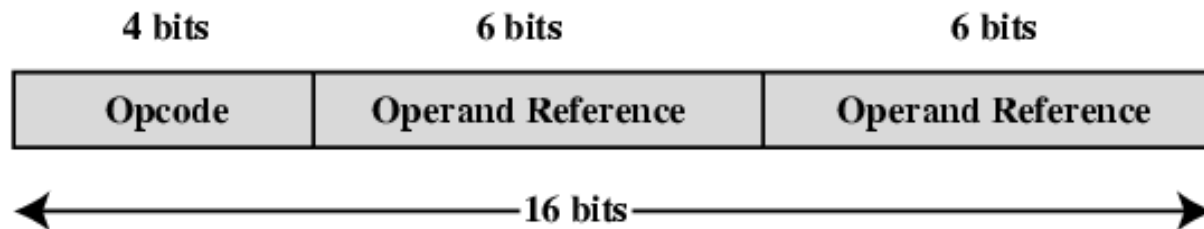
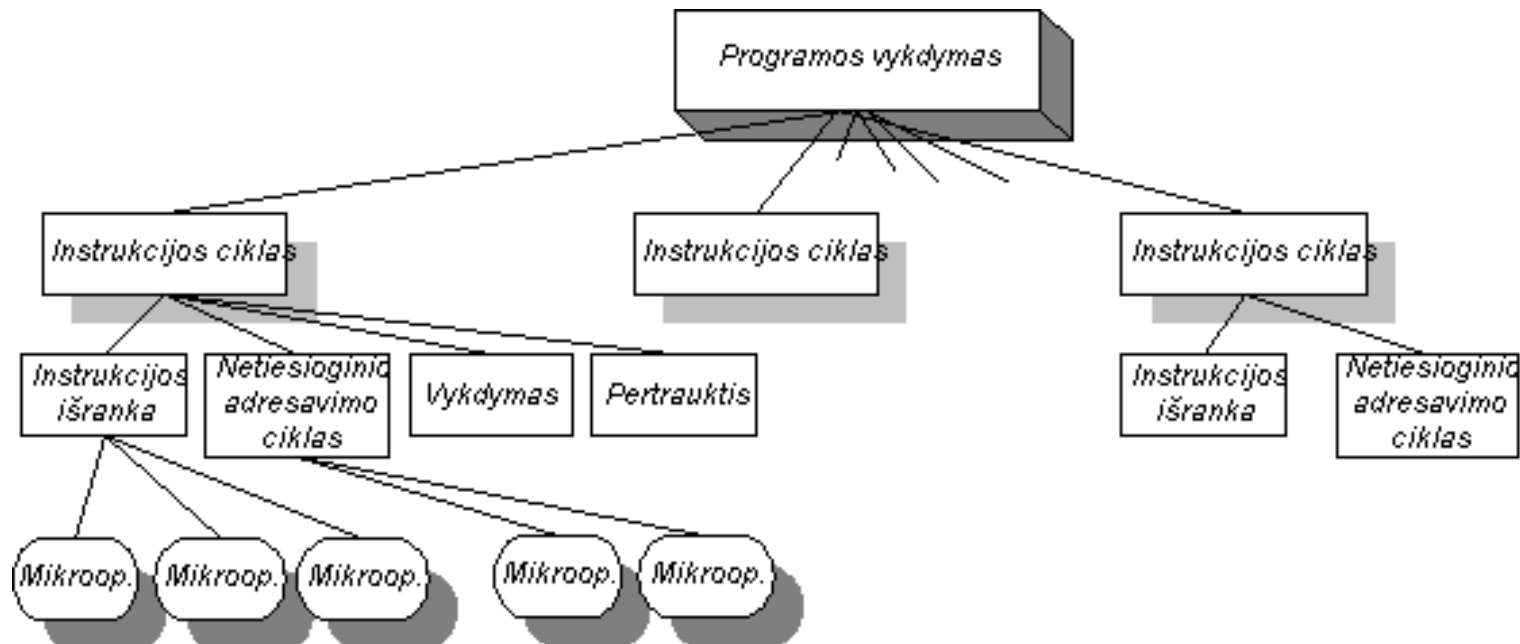
A decorative graphic consisting of a light gray circle on the left side. A thick gray horizontal bar extends from the left edge of the circle across the top of the slide. A large black left square bracket is positioned on the left side of the gray bar, and a large light gray right square bracket is on the right side. The main title text is centered within the gray bar.

Procesorių architektūriniai sprendimai

2 paskaita

Instrukcijų lygmens lygegriatumas

Programos vykdymo elementai



Procesorių architektūros vystymosi eiga

- 1975 - 1986 m. bitų lygmens lygiagretumas (4, 8, 16, 32, 64, 128 bitų žodžiai). *Žodis – informacijos vienetas.*
- 1980 - 1998 m. instrukcijų lygmens lygiagretumas (ILP) (*instrukcijos dekodavimas, sveiko ir trupmeninio skaičiaus aritmetika, adreso apskaičiavimas, operando išrinkimas, įrašymas atliekamas per vieną ciklą*):
 - konvejeris, superkonvejeris
 - superskaliarinis, superskaliarinis-superkonvejerinis procesorius
 - VLIW, RISC
- 1991m - ... gijų lygmens lygiagretumas
- 2004 m - ... branduolių lygmens lygiagretumas

[Žodžio ilgio įtaka našumui]

Informacijos kiekis (kvantas) esantis vienoje registro arba op.atminties ląstelėje, vadinamas **žodžiu**. Žodžiui naudojamas vienas adresas.

Bitų skaičius žodyje yra vadinamas **žodžio ilgiu**. Jis priklauso nuo kompiuterio procesoriaus tipo ir modelio.

Gali būti 8, 16, 32, 64 bitų ilgio žodžiai. Į žodį gali būti įpakuoti keli simboliai (buvo naudojami 8, 9, 12, 18, 24, 36, 39, 40, 48, 60 bitų ilgio žodžiai)

8 bitų kompiuteriai		16 bitų kompiuteriai		32 bitų kompiuteriai	
Pozicijos adresas	8 bitų pločio turinys	Pozicijos adresas	16 bitų pločio turinys	Pozicijos adresas	32 bitų pločio turinys
	↔		↔		↔
10	A	10	T A	10	I M T A
11	T	11	I M	11	S I T N
12	M	12	T N		
13	I	13	S I		
14	N				
15	T				
16	I				
17	S				

Instrukcijų lygmens lygiagretumas (ILP)

Instrukcijų lygmens lygiagretumas (ILP) – tai matas, kuris nusako, kiek operacijų kompiuterio programoje gali būti atliktos vienu metu.

Pavyzdys:

1. $e = a + b$
2. $f = c + d$
3. $g = e * f$

Pirma ir antra operacijos yra nepriklausomos, todėl gali būti atliktos vienu metu. Jei laikyti, kad bet kuriai operacijai atlikti naudojamas tas pats laikas, tuomet ILP lygus 3/2.

Kompiliatorių tikslas – sudėlioti programos instrukcijų srautą taip, kad išnaudoti ILP.

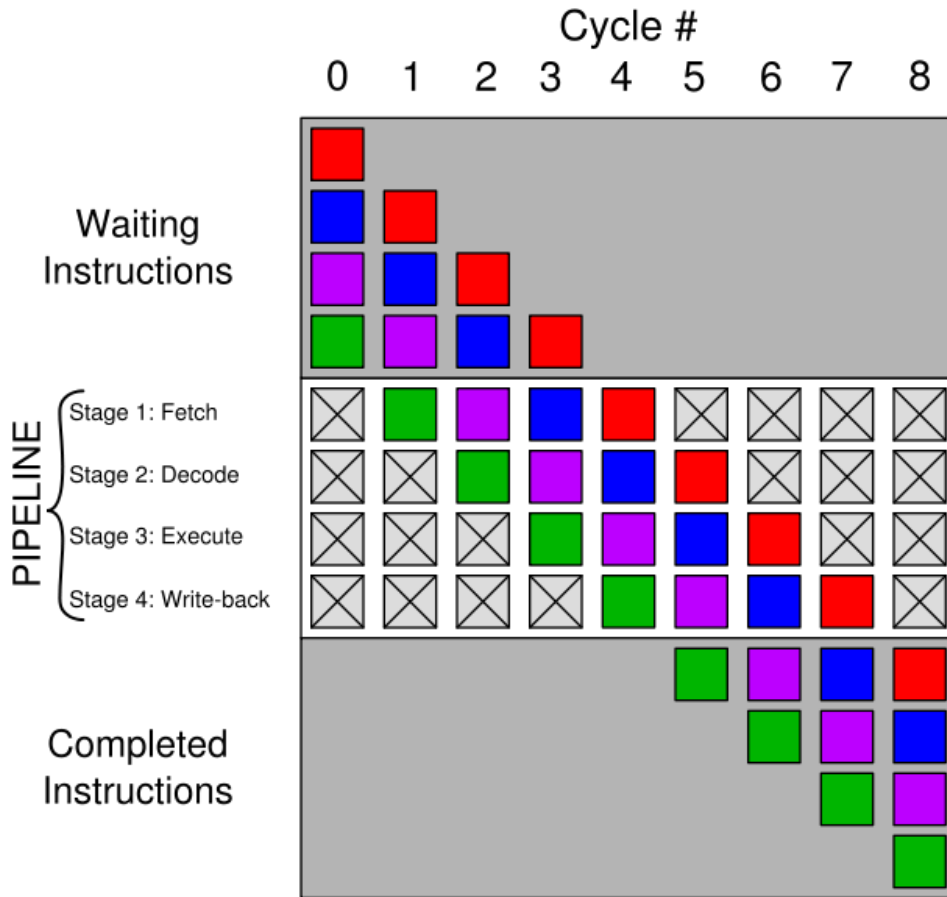
Instrukcijų lygmens lygiagretumas (ILP)

Procesorių mikro-architektūros ILP sprendimai:

- Konvejeris, superkonvejeris
- Registų pervardinimas
- Šakojimosi spėjimas
- Superskaliariškumas
- VLIW (very long instruction word)
- Vektorinis procesorius

$$\frac{\text{time}}{\text{program}} = \frac{\text{time}}{\text{cycle}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{instructions}}{\text{program}}$$

Instrukcijų konvejeris



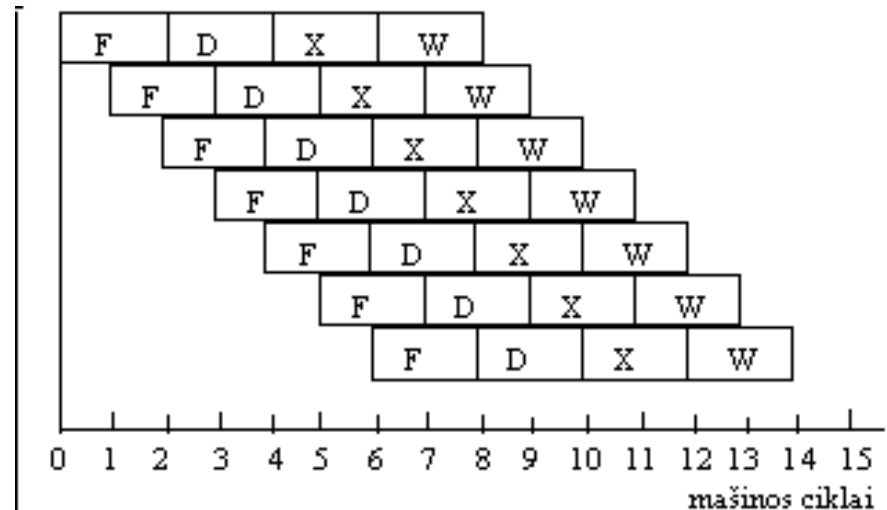
Instrukcijų superkonvejeris

Bazinio procesoriaus darbo greitis gali būti padidintas skaidant instrukcijos ciklą į m dalių.

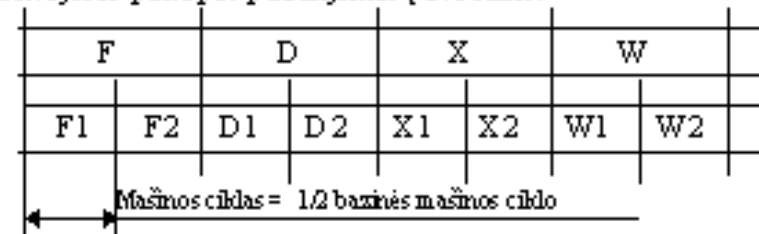
Tokiame procesoriuje vienu metu vykdymo fazėje gali būti m komandų, o pilnai užpildytame konvejeriulyje - $p \cdot m$ komandų, kur p – konvejerio pakopų skaičius.

Privalumai:

- instrukcijos rezultatas gaunamas kas $\frac{1}{2}$ takto
- Keletas mikrooperacijų gali kreiptis į atmintį vienu metu

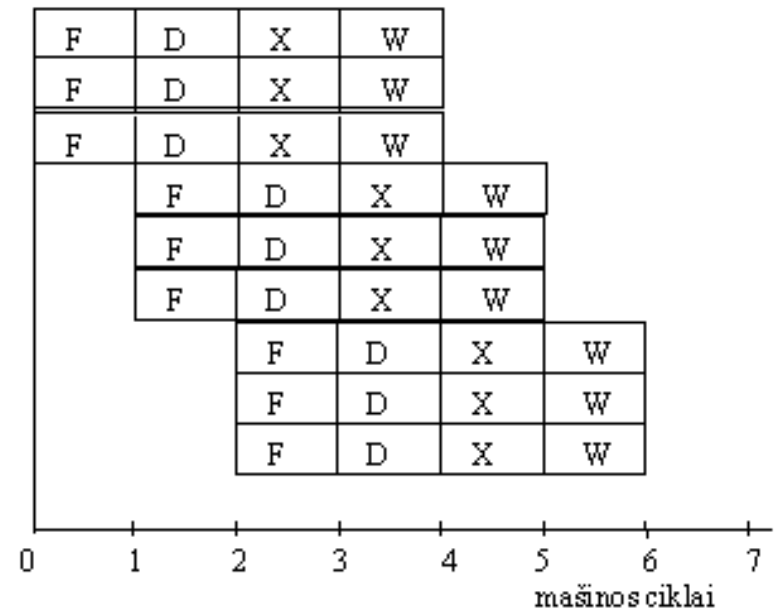


Konvejerio pakopos padalijimas į dvi fazes:



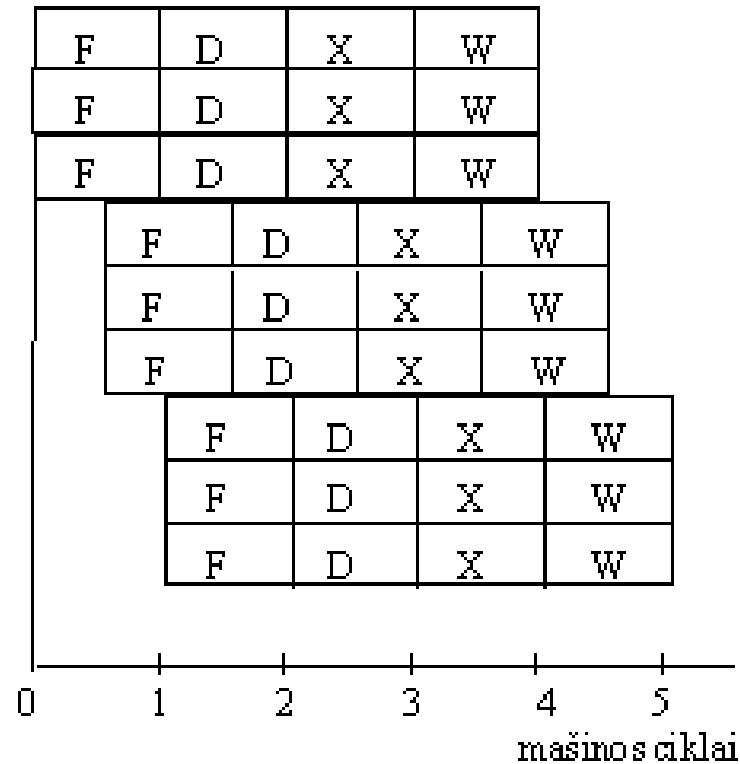
Superskaliarinis procesorius

Bazinio procesoriaus našumas gali būti padidintas, jei kiekvienu momentu galėtume inicijuoti ne vieną, o **kelias komandas**. Tokia galimybė realizuota **superskaliariniuose ir VLIW** (Very Long Instruction Word) procesoriuose. Toks procesorius turi **n** funkcinių įtaisų, kurie paprastai nėra identiški. Kad išnaudoti potencialias išlygiagretinimo galimybes, reikia, kad programos komandų sraute kiekvienu momentų būtų **n** nepriklausomų komandų. Jei kuriuo nors momentu nebus **n** nepriklausomų komandų, reikės įterpti atitinkamą skaičių tuščių (NOOP) operacijų.



Superskaliariškumas ypač aktualus x86 procesoriams, kai reikia didinti greitį ir išlaikyti suderinamumą.

Superskaliarinis, superkonvejerinis procesorius



Superskaliarinio ir superkonvejerinio procesorių savybės gali būti realizuotos viename procesoriuje, kuris vadinamas **superskaliariniu superkonvejeriniu**.

[VLIW procesoriai]

VLIW (very long instruction word) procesoriai pasižymi žymiai ilgesniu komandos žodžiu.

Komanda padalinta į laukus, kiekvienas iš kurių valdo atskirą funkcinį įtaisą.

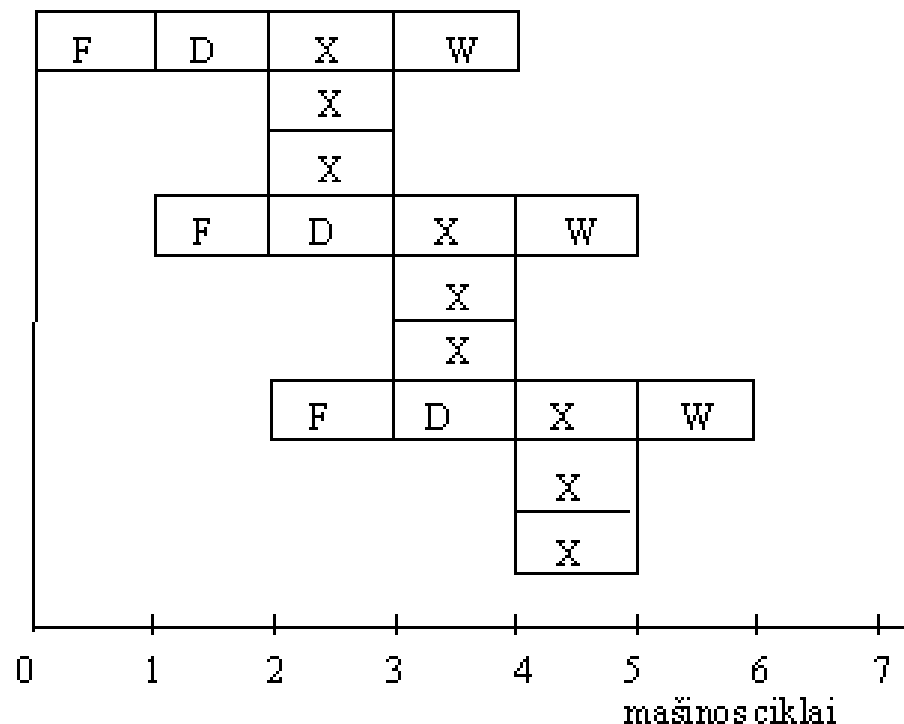
VLIW procesorius, kurio lygiagretumo laipsnis n , bet kuriuo laiko momentu taip pat gali vykdyti n komandų. Kokios komandos bus vykdomos kartu, sprendžiama kompiliavimo metu. Kompiliatorius analizuoja komandų srautą ir iš paprastų komandų formuoja ilgas, atsižvelgdamas į funkcinį įtaisų rinkinį ir jų užimtumą.

Jei kuriuo nors momentu nėra reikiamo skaičiaus tinkamų operacijų, įveda tuščias operacijas

VLIW procesoriai

VLIW instrukcijoje mikrooperacijų srautas išdėstytas taip, kad užtikrintų lygiagrečių darbą instrukcijos vykdymo etape X. Tai žymiai supaprastina procesoriaus architektūrą lyginant su RISC ir CISC.

VLIW pavyzdys - Intel Itanium2 procesorius



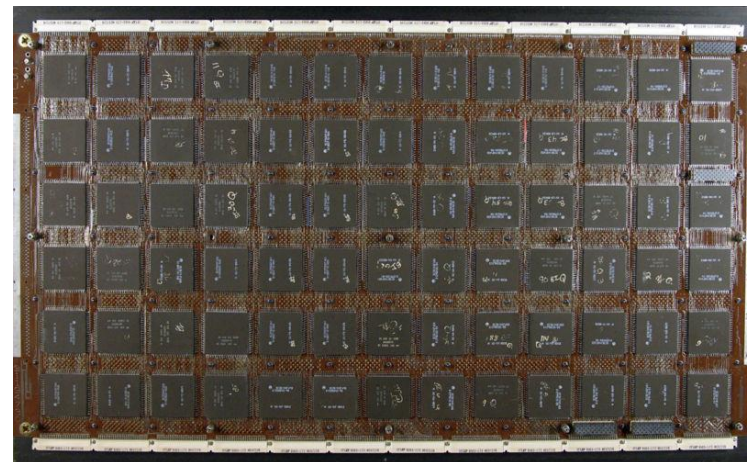
[Vektorinis procesorius]

Skaliarinis procesorius – tai procesorius atliekantis vieną operaciją su vienu duomenų elementu vieno ciklo metu.

Vektorinis procesorius – tai procesorių masyvas, kuris gali atlikti matematinės operacijas su daugeliu duomenų elementų vienu metu. (SIMD)

Dauguma procesorių skaliariniai. Vektoriniai procesoriai naudojami moksliniuose skaičiavimuose. Jie buvo populiarius 1980-1990. Gamintojai IBM, Cray, Toshiba.

Cray YMP vektorinių procesorių plokštė
1992-2000 m.



[Vektorinis procesorius]

Pavyzdys: reikia atlikti sudėties operacijas su 10 skaičių rinkiniu.

Skaliarinis procesorius

read the next instruction and decode it
get this number
get that number
add them
put the result here
read the next instruction and decode it
get this number
get that number
add them
put the result there

Vektorinis procesorius

read instruction and decode it
get these 10 numbers
get those 10 numbers add them
put the results here

[Kas yra “Instruction Set”?]

- Išbaigtas instrukcijų rinkinys, suprantamas procesoriui arba arba virtualiai mašinai.
- Mašininiai kodai, dažniausiai pateikiami, kaip assemblerio kodai.

Instrukciją sudaro

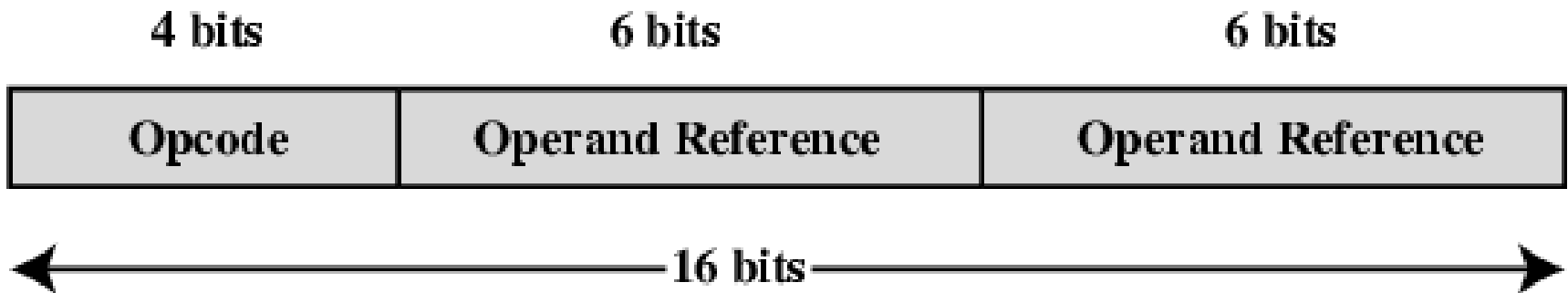
- Operacijos kodas (Opcode - mikrooperacija)
- Operando nuoroda (reference)
- Resultatų operando nuoroda (reference)
- Nuoroda į sekančią instrukciją

ISA (Instruction Set Architecture)

- “Žemo lygio” programuotojui matomas instrukcijų rinkinys.
- ISA – tarpininkas tarp aparatūrinės ir programinės įrangos

Instrukcijų rinkinių tipai

- Duomenų apdorojimas (data processing)
- Duomenų saugojimas atmintyje (data storage)
- Duomenų persiuntimas (I/O) (data movement)
- Programos vykdymo kontrolė (program flow control)



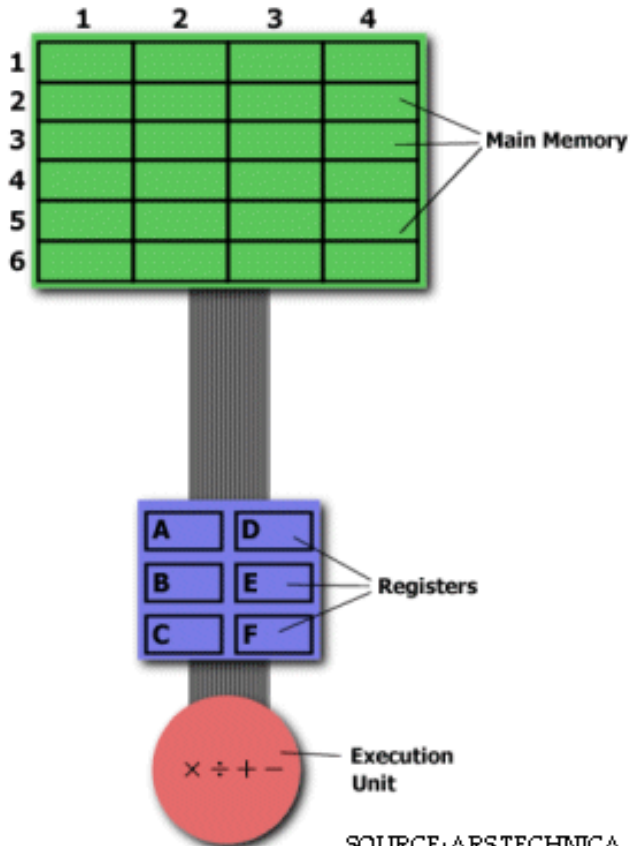
[Adreso įtaka instrukcijai]

- **Daugiau adresų**
 - Sudėtingesnė instrukcija
 - Reikalaujama daugiau registrų
 - Tarpregistrinės operacijos yra greitesnės
 - Programa įvykdoma naudojant mažiau instrukcijų
- **Mažiau adresų**
 - Nesudėtinga instrukcija
 - Programa įvykdoma naudojant daugiau instrukcijų
 - Greitesnės instrukcijos išrinkimo ir vykdymo operacijos

[CISC]

CISC - *Complex Instruction Set Computing* - sudėtinga komandų sistema

RISC - *Reduced Instruction Set Computing* - supaprastinta komandų sistema



CISC – idėja atlikti užduotį naudojant kuo mažiau kodo eilučių. Procesorius supranta sudėtingą kodo eilutę. Pvz.

MULT 2:3, 5:2

Pateiktame kode neaprašytos smulkios instrukcijos, kaip duomenų ir instrukcijų nuskaitymas iš atminties ir talpinimas registuose ir t.t. Tai daro pats procesorius, kuriame “įsiūtos” sudėtingos instrukcijos. Instrukcija atliekama per keletą procesoriaus taktų

CISC savybės:

turtingesnė sistema - trumpesnė programa

turtingesnė sistema - lengviau sukurti kompiliatorių

[CISC kriterijai]

- komandų skaičius – kiek galima daugiau
- adresacijos būdų skaičius – kiek galima daugiau
- komandų formatų skaičius – kiek galima daugiau
- komandų trukmė taktais – CPI (Clock Per Instruction) > 1
- į atmintį kreipiasi įvairių tipų komandos (*registry-memory ISA*)
- yra specialios paskirties registrai
- valdymas – mikroprograminis
- komandos būna aukštesnio lygio t.y. panašios į aukšto lygio programavimo kalbos.
- lengviau rašomi ir naudojami paprastesni kompiliatoriai

[CISC]

Ką parodė praktika?

- sunku pasiekti komandų visišką atitikimą aukšto lygio programavimo kalbos operatoriams
- CISC atveju sunkiau optimizuoti kodą, užtikrinti efektyvų konvejerio darbą
- kompiliatoriaus sugeneruotame kode panaudojama tik dalis komandų
- programos ilgis (komandų skaičius) svyruoja nedaug

[RISC]

RISC procesoriai naudoja paprastas instrukcijas, kurios atliekamos per vieną procesoriaus taktą. Analogiškas daugybos `MULT 2:3, 5:2` veiksmas čia atliekamas:

```
LOAD A, 2:3      // pakrovimas į registrą  
LOAD B, 5:2      // pakrovimas į registrą  
PROD A, B        // daugyba  
STORE 2:3, A     // pakrovimas į atmintį
```

Atlikdamas 4 instrukcijas RISC naudoja daugiau RAM, reikalingas sudėtingesnis kompiliatorius, perkoduoja aukšto lygio programavimo kalbos kodą į instrukcijas. Tačiau dėl paprastų instrukcijų, naudojamas paprastesnis (mažiau tranzistorių turintis) procesorius.

[RISC privalumai]

- trumpos komandos (dauguma įvykdomos per 1 CPU ciklą)
- registras-registras tipo komandos (*load-store* ISA)
- paprastesni adresavimo būdai
- paprastesni komandų formatai
- efektyvūs kompiliatoriai
- efektyvus konvejerizavimas

RISC ir CISC lyginimas

CISC

Svarbesnė aparatūrinė dalis

Naudojamos daugelio taktų
Sudėtingos instrukcijos

Registas-atmintis:
"LOAD" ir "STORE"
Įtrauktos į instrukciją

Trumpas kodas, didelis
procesoriaus dažnis

Tranzistoriai naudojami sudėtingų
instrukcijų saugojimui

RISC

Svarbesnė programinė dalis

Vieno takto, sumažintos instrukcijos

Registas - registras:
"LOAD" ir "STORE"
Instrukcijos nepriklausomos

Žemesnis procesoriaus dažnis,
Didelis kodas

Daugiau naudojama atmintis

RISC vystymasis

RISC pradžia:

- Stenfordo (MIPS) ir Berklio (RISC I, RISC II) bei IBM (801) kompiuteriai.
- Paprasti procesoriai (20 000 - 100 000 tranzistorių)
- Kompaktiškos komandų sistemos (30-50 komandų).

Kompiuteriai

- Stenfordo RISC: R3000; Berkley RISC: Am29000, SPARC, i960CA

RISC dabartis :

- superskaliariniai
- superkonvejerizuoti
- VLIW - labai ilgo komandos žodžio procesoriai

Kompiuteriai

Hewlett-Packard RISC: PA-RISC

Sun RISC: UltraSPARC III, UltraSPARC IV, UltraSPARC V

Hewlett-Packard + Intel : IA-64 (EPIC): Itanium, Madison, Montecito

IBM: POWER2, POWER3, POWER5, POWER6, POWER7

[Procesorių našumo didinimas taikant lygiagretinimą]

- Bitų lygmens lygiagretumas (4, 8, 16, 32, 64 bitų žodžiai).
- Instrukcijų lygmens lygiagretumas (*Instruction Level Parallelism*)
- Gijų lygmens lygiagretumas (*Multi-Threading*)
- Branduolių lygmens lygiagretumas (*Multi-core*)
- Procesorių lygmens lygiagretumas (*Multi-processing*)

[Gijos (Threads)]

Gija vadiname trumpinį **vykdymo gija** (*thread of execution*).

Gijos – tai būdas, padalinti programos vykdymą į du dar daugiau vienu metu vykdomus darbus.

Gijos gali būti vykdomos lygiagrečiai keliuose procesoriuose arba klasteriuose. Gija panaši į procesą, tačiau gijos egzistuoja vieno proceso rėmuose ir dalinasi resursais, kurie skirti procesui, todėl daugiaprosesorinės užduotys yra lėtesnės už daugelio gijų darbus.

Gijos valdomos per sisteminių kvietimų (*system calls*) sąsają.

(open, read, write, close, wait, exec, fork, exit, kill. Linux OS yra 319 skirtingų sisteminių kvietimų).

Skirtumai tarp gijų ir procesų

- Procesas
 - Atskira atminties adresų erdvė
 - Atskiras priėjimas prie I/O sistemos
 - Komunikacijos vykdomos tarp procesų naudojant tarprocesorinės komunikacijos (MPI, sinchronizacija, bendra atmintis, RPC)
- Gija
 - Naudoja proceso, kuriam ji priklauso atminties adresų erdvę
 - Naudojami proceso priėjimo prie I/O sistemos kanalai
 - Komunikacijos vykdomos per procesui išskirtą atminties erdvę
- **Gijų tipai:** branduolio gijos (threads), vartotojo gijos (fibers).
- **Gijų valdymo būdai:**
 - prioritetinis (preemptive multithreading),
 - kooperatyvinis (cooperative multithreading).

[Daugelio gijų skaičiavimai]

Daugelio gijų technologijos:

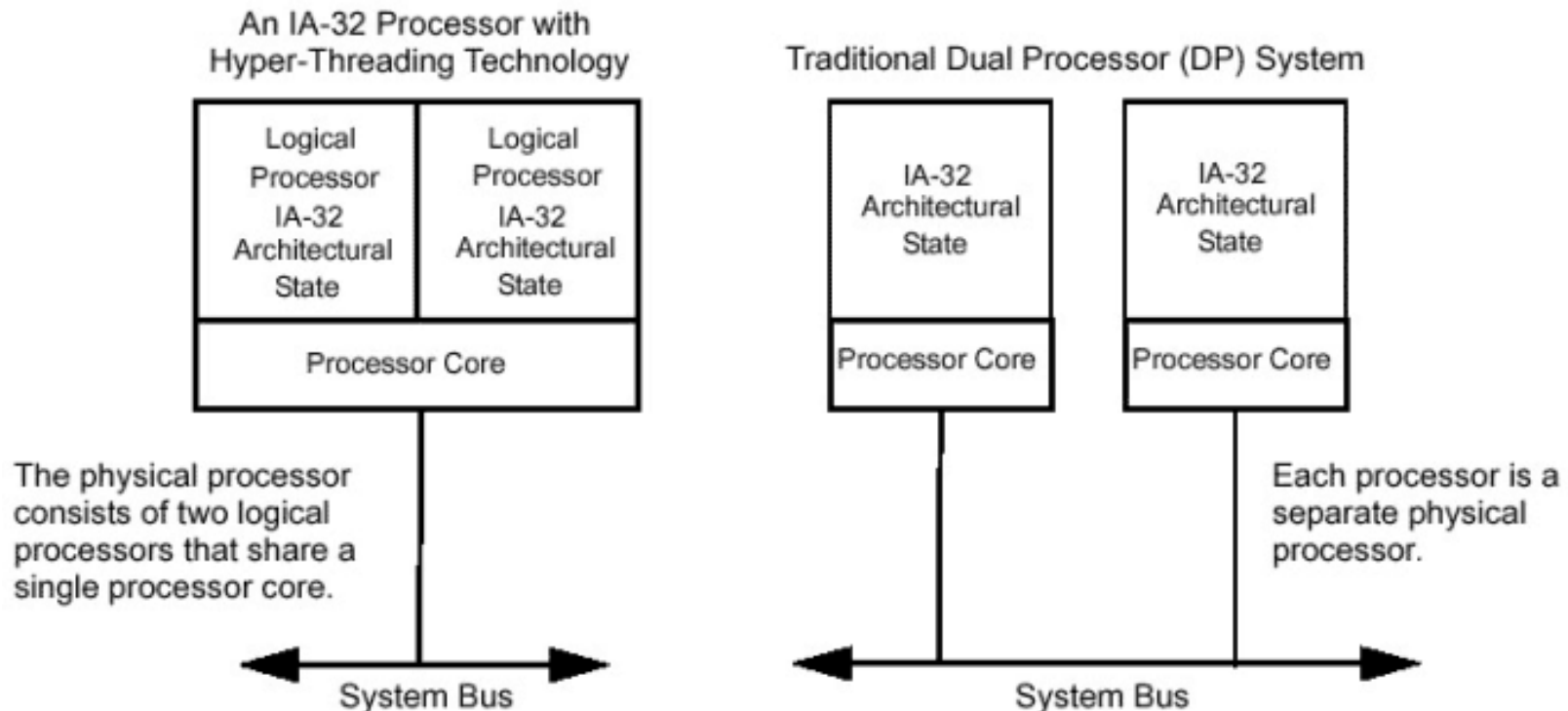
- **Multi-threaded**: vykdomos instrukcijos iš skirtingų gijų persijunginėjant nuo vienos gijos prie kitos.
- **Simultaneous multithreading** (SMT): vykdoma daug instrukcijų iš skirtingų gijų per vieną ciklą (superskaliariniai procesoriai).
- **Chip-level multiprocessing** (CMP arba Multi-core): integruojama du ar daugiau superskaliarinių procesorių/branduolių į vieną mikro-schemą. Kiekvienas branduolys vykdo atskirą giją nepriklausomai.
- **Kombinacija SMT/CMP**

Daugelio gijų skaičiavimai leidžia geriau išnaudoti esamus ALU. Paprastai procesorius modifikuojamas didinant registrų skaičių, kad galėtų saugoti kelių gijų duomenis. Vykdomas vienu metu gijų skaičius būna 2, 4, 8.

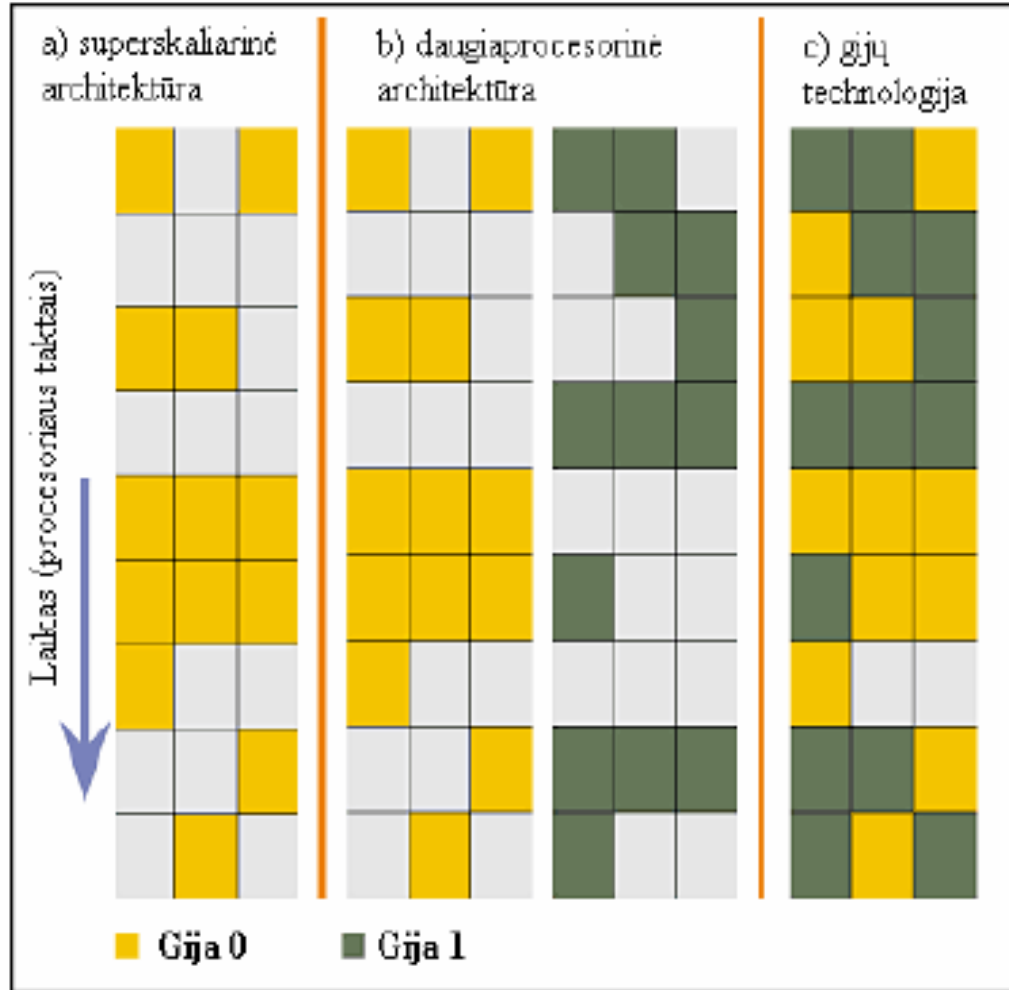
Problema – konfliktų sprendimai naudojant bendra atmintį gijoms.

Hyper-Threading technologija

Pirmą kartą **simultaneous multithreading** technologija komerciškai sėkmingai panaudota buvo panaudotas **Intel Xeon MP** (Foster MP). Ji pavadinta Hyper-Threading Technology. 2002 m. HTT pradėta naudoti ir Pentium 4 (nuo 3,0GHz). Kiti procesoriai su SMT: IBM POWER, MIPS MP, SUN SPARC



Hyper-Threading technologija

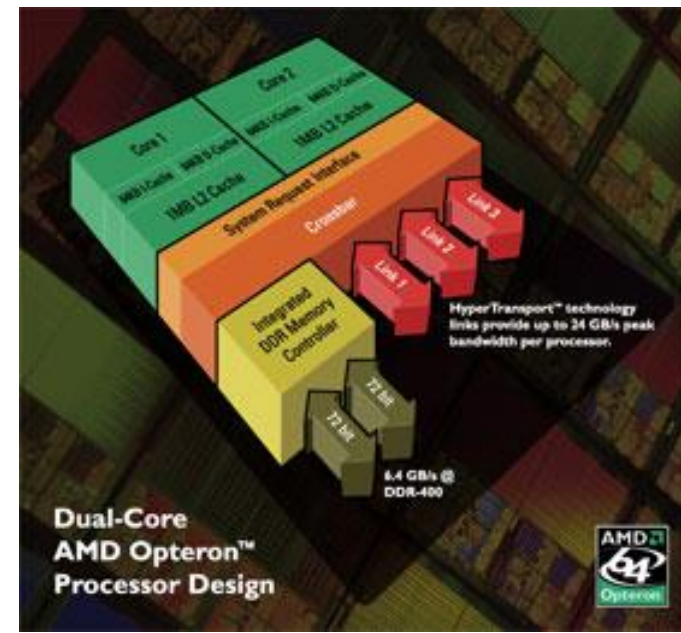
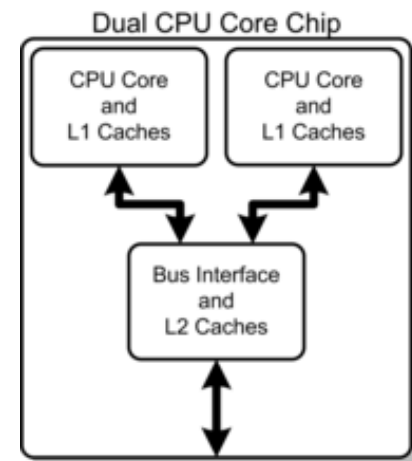


[Daugelio branduolių procesoriai]

Žiūrint į daugiabranduolinius procesorius iš TLP pusės jie priklauso multiprocesoriniai kristalų (*Chip MultiProcessing* – CMP, Multi-Core) grupei.

Implementacijos:

IBM POWER 5, POWER6, POWER 7
SUN SPARK IV, SPARK VII
HP PA-RISC(PA-8800)
AMD Opteron, Athlon 64 X2
INTEL Xeon, Pentium Core 2, i5, i7



[Daugelio branduolių procesoriai]

Pagal spartinančiosios atminties panaudojimą nustatomas ryšys tarp branduolių:

- Stiprus (naudojama bendra L2 atmintis)
- Silpnas (naudojama atskira L2 atmintis, siunčiami pranešimai)

Branduolių sujungimo topologijos:

- Linijinė magistralė
- Žiedas
- 2D tinklas
- Komutacinis tinklas

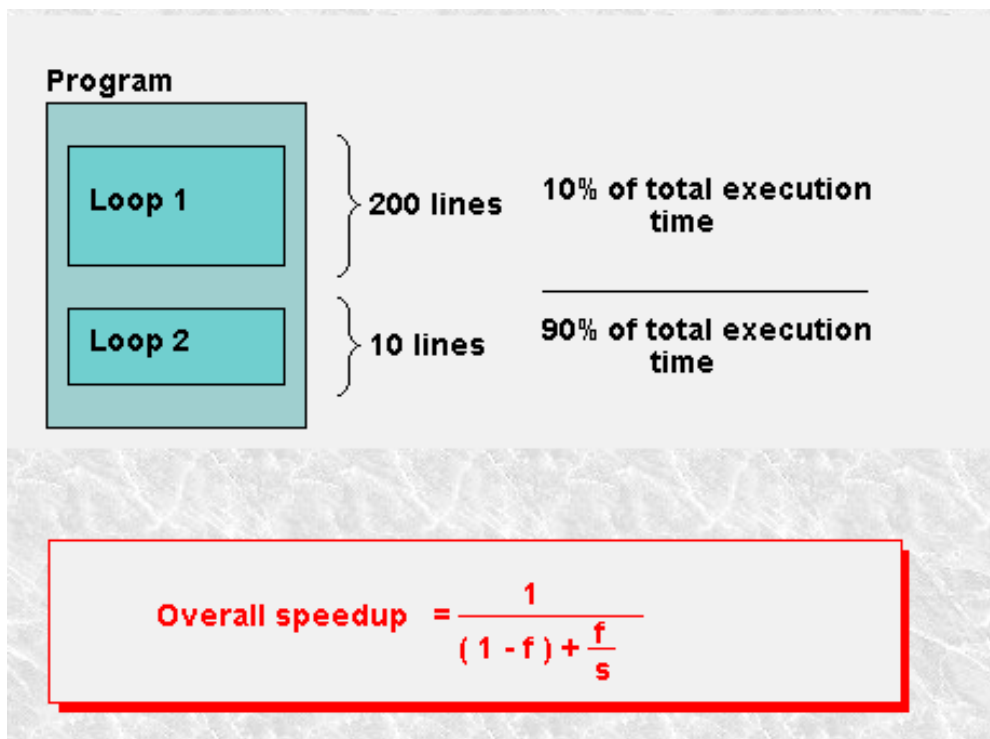
Pagal branduolių pajėgumus CMP sistemos skirstomos:

- Homogeninės (vienodi branduoliai)
- Heterogeninės (skirtingi branduoliai)

[Daugelio branduolių procesoriai]

Branduolių panaudojimo efektyvumas priklauso nuo programinės įrangos išlygiagretinimo lygio arba vykdomų užduočių nepriklausomumo.

Amdahl dėsnis:



f – programos kodo dalis, kuri gali būti vykdoma lygiagrečiai

S – procesorių skaičius

[Amdahl dėsnis]

Kokia dalis skaičiavimų turėtų būti lygiagreti, kad turint 100 procesorių būtų pasiektas pagreitėjimas lygus 80?

$$\frac{1}{(1-f) + f/100} = 80$$

$$f = 0.9975$$

[CMP privalumai ir trūkumai]

Privalumai:

- Greitesnės komunikacijos tarp procesorių, nes naudojama ta pati L2 spartinančioji atminti (ne visada).
- Fiziškai mažesnis procesoriaus plotas
- Mažesnės energijos sąnaudos

Trūkumai:

- Reikalinga OS palaikanti CMP architektūros subtilybes
- Sudėtingesnės temperatūrinės sąlygos (reikalingas geresnis aušinimas, mažinamas procesoriaus dažnis, įtampa)
- Sudėtingesnis gamybos procesas
- Efektyvumas didinant uždavinio apimtį, priklauso nuo uždavinio tipo.
- Bottleneck- I/O sistema. Efektyviai naudojamas tik sprendžiant lygiagrečias aplikacijas.

[SMP (Symmetric multiprocessing)]

Symmetric Multiprocessing, arba SMP – tai daugiaprosesorinė kompiuterių architektūra, kai du ar daugiau identiškai kompiuteriai naudojami ir dalinasi viena bendra atmintimi.

SMP sistemos leidžia bet kuriam procesoriui dirbti su bet kuria užduotimi, nepriklausomai nuo to, kurioje atminties vietoje yra duomenys. Operacinės sistemos lygmenyje yra realizuota galimybė darbams migruoti tarp procesorių, taip užtikrinant apkrovimo balansavimą.

Trūkumas:

Atmintis dirba žymiai lėčiau nei procesorius, todėl procesoriams dalinantis atmintimi, pailgėja atminties pasiekimo greitis.

Daugiaprocesorinių sistemų klasifikacija

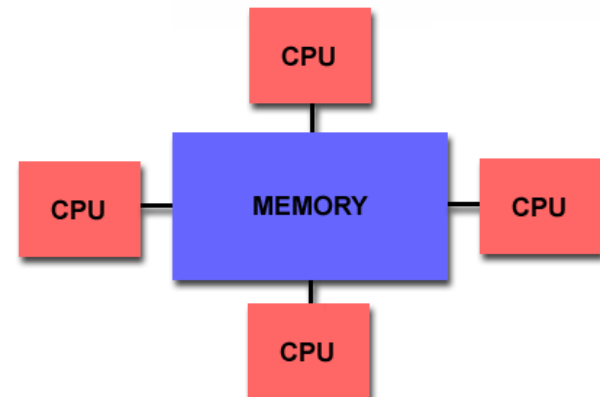
Daugelio procesorių sistemose būtina apibrėžti kaip vykdomos komunikacijos tarp procesorių, koks naudojamas principas pasiekiant duomenis atmintyje.

Pagrindinės koncepcijos:

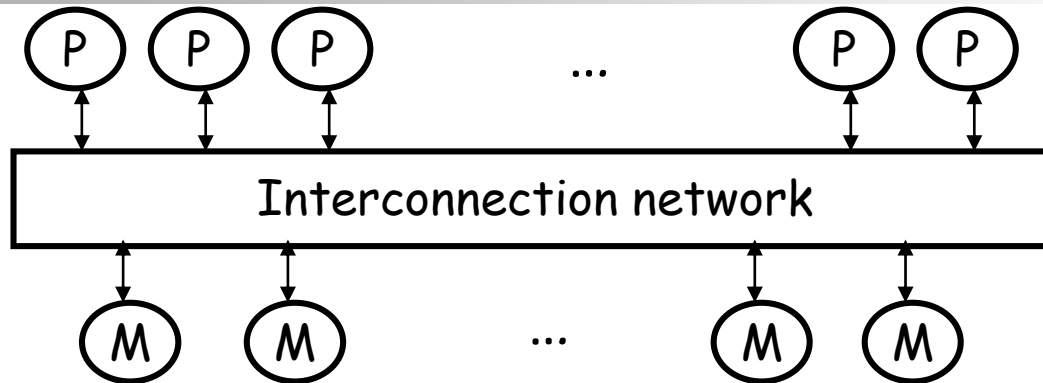
- Komunikacija per bendrą atmintį. **Bendrosios atminties** (shared memory) kompiuteriai.
 - UMA (Uniform Memory Access)
 - NUMA (Non-Uniform Memory Access)
- Komunikacija apsikeičiant pranešimais. **Paskirstytosios atminties** (distributed memory) kompiuteriai.
- Mišrios “Bendros-paskirstytos” atminties kompiuteriai

Bendrosios atminties kompiuteriai

- Procesoriai veikia nepriklausomai, bet dalinasi bendra atmintimi. Visi procesoriai gali pasiekti visą atmintį.
- Naudojama globali atminties adresavimo sistema
- Procesoriai apsikeičia duomenimis modifikuodami duomenis, esančius bendroje atmintyje.
- Bet kurio procesoriaus atlikti duomenų pakeitimai atmintyje matomi visiems procesoriams.
- Komunikacinio mechanizmo principas: **put()** and **get()**
- Pagal atminties pasiekimo laiką bendros atminties kompiuteriai skirstomi į : **UMA** ir **NUMA**.

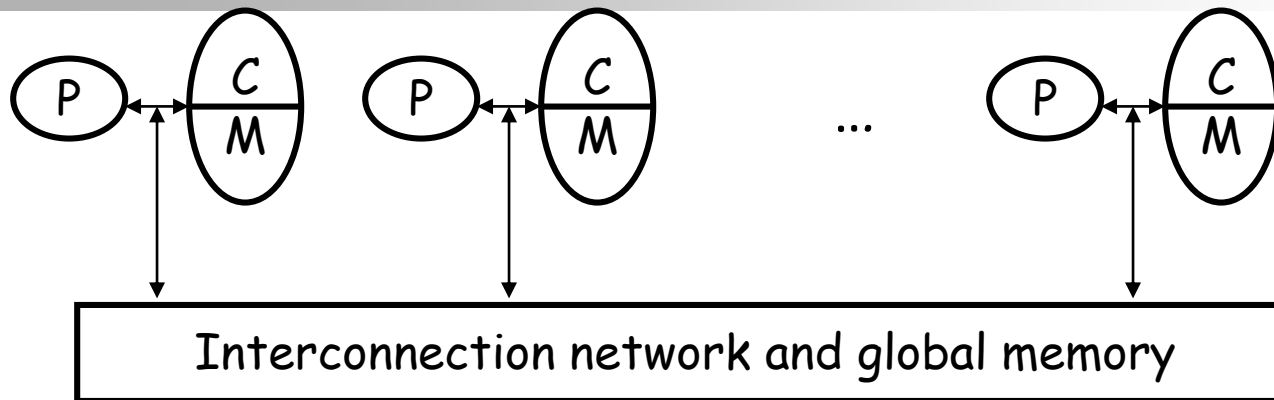


Uniform Memory Access (UMA)



- UMA: Vienodas pasiekimo galimybės ir laikas visiems procesoriams. Tai pasiekama naudojant dinaminis procesorių sujungimo tinklus.
- Vienodi procesoriai
- Spart. atmintis panaudojimas: užtikrina greitą atminties priėmimą ir leidžia sumažinti tinklo apkrovimą.
- **Cache coherent** reiškia, kad kai vienas procesorius keičia duomenis atmintyje, visi kiti sužino apie tai. Cache coherency užtikrinamas aparatinia lygmenyje. Toks mechanizmas dar vadinamas cc-UMA - Cache Coherent UMA.
- Realizacija **Symmetric Multiprocessor (SMP)** kompiuteriai.

Non-Uniform Memory Access (NUMA)



- Fizinė realizacija – fiziškai sujungti keli SMP kompiuteriai su atminties moduliu.
- Kiekvienas procesorius turi savo lokalią atmintį, bet atminties adresavimas yra globalus. Procesoriai gali pasiekti bet kurio kito procesoriaus atmintį.
- Lokalios atminties pasiekimas yra greitas.
- Nutolusi atmintis pasiekama per tinklą (lėtai)

- Cache-coherence problema sprendžiama aparatūriniame lygmenyje
- Jei naudojamas cache coherency, tokie kompiuteriai vadinami cc-NUMA Cache Coherent NUMA.

- Pavyzdys: SGI Altix UV (2048 Core, 64TB RAM).

[Atminčių darna (*coherence*)]

Galimi du būdai, kaip spręsti atminčių darnos problemą t.y naudoti:

- *Write invalidate* protokolą

Tai metodas, kurio pagalba užtikrinama, kad procesorius, prieš rašydamas į atmintį, turi ekskliuzyvinę prieigą prie duomenų.

- *Write update (write broadcast)* protokolą

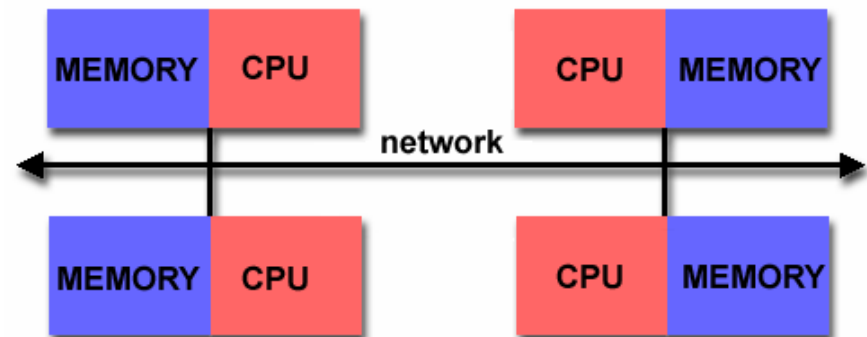
Tai metodas, kai atnaujinus spartinančiąją atmintį viename procesoriuje, tokie pat pakeitimai atliekami visų likusių procesorių spartinančiosiose atmintyse.

NUMA ir UMA savybės

- Nesunkiai programuojamos aplikacijos, nes atminties skaitymas-rašymas matomas visiems procesoriams. Greitas duomenų apsikeitimas.
- Tačiau bendrų duomenų skaitymas-rašymas turi būti koordinuojamas (sinchronizuojamas).
- NUMA and UMA platformoms programuojant naudojami skirtingi algoritmai. NUMA mašinose turi būti įvertintas atminties nevienodas pasiekiamumas.
- Cache (spart.atmintis) visuose procesoriuose turi būti koordinuojamas tam, kad būtų užtikrintas vienodas spart. atminties turinys. Cache coherence problema.
- **Pagrindinis trūkumas** – prastas sistemų plečiamumas. Padidinus CPU skaičių didėja komunikacinio tinklo apkrovimas tiek globalios atminties pasiekimo prasme, tiek ir spart.atminties darnos prasme.

Paskirstytosios atminties (distributed memory) kompiuteriai

- Procesoriai turi savo lokalias atmintis.
- Atminties adresavimas nėra globalus, kiekvienas procesorius dirba tik su savo lokalia atmintimi.
- Pakeitimai lokaliajame atmintyje nėra matomi globaliai, todėl nėra spart. atminties koherencijos problemos.
- Duomenų apsikeitimas tarp procesorių vyksta siunčiant pranešimus. Programoje turi būti labai aiškiai apibrėžta, kada ir kas ir kam siunčia pranešimą.
- Komunikacijų tinklui naudojamas kompiuterių tinklų technologijos (Ethernet iki spec. tinklų Myrinet, Infiniband....).



Paskirstytosios atminties (distributed memory) kompiuterių savybės

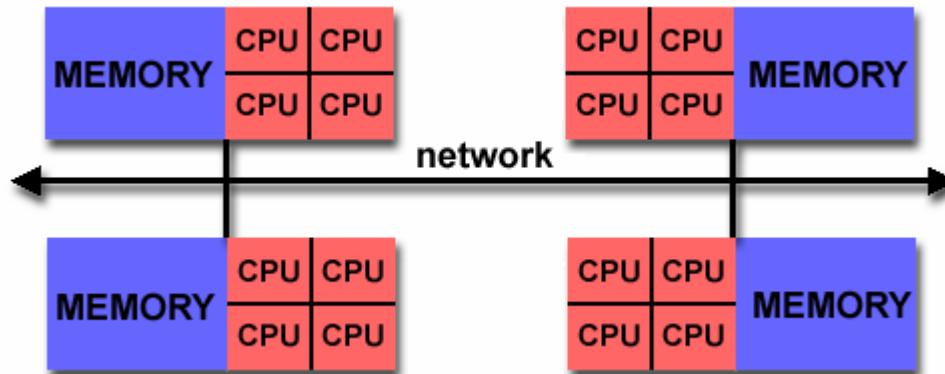
Privalumai:

- Galimas atminties ir CPU proporcingas didinimas. Jis nedidina tinklo apkrovimo.
- Kiekvienas procesorius dirba tik su savo atmintimi, todėl čia nėra atminties koherentiškumo problemos.
- Lengvai plečiama architektūra. Didžiausios pasaulyje paskirstytos atminties kompiuterinės sistemos turi iki 10^7 procesorių.
- Geras kainos ir našumo santykis. Galimi PC klasteriai, kaip pigiausias variantas.

Trūkumai:

- Programuotojas privalo rūpintis duomenų apsikeitimo tarp procesorių procedūromis
- Gali kilti problemų su duomenų paskirstymu tarp procesorių, atsižvelgiant į procesorių ir tinklo heterogeniškumą.

Mišriosios “Distributed-Shared” sistemos



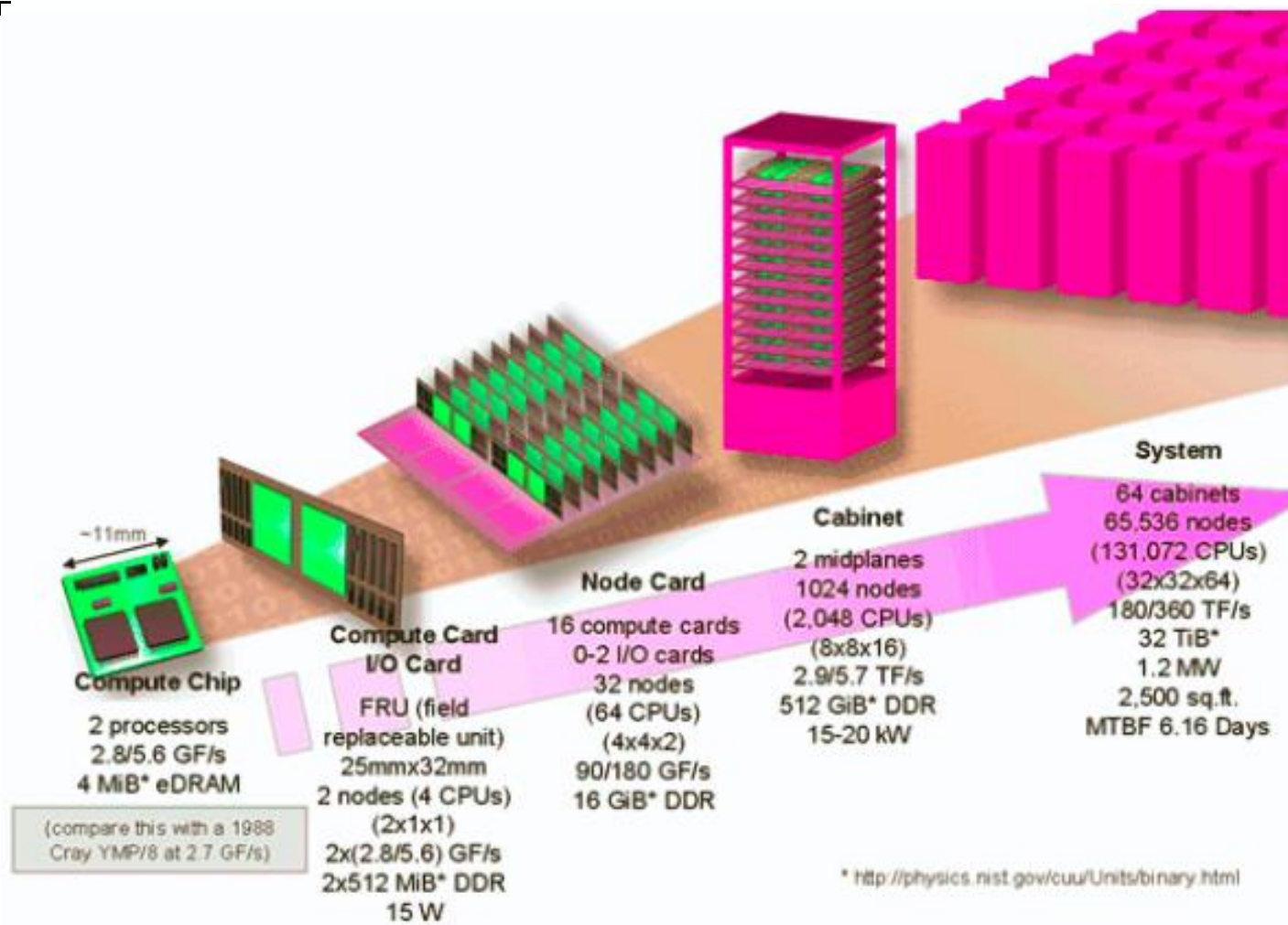
- Tai tokios sistemos, kuriose SMP kompiuteriai, turintys kelis procesorius ir bendrą atmintį sujungiami į tinklą.
- Atmintis matoma tik lokali, kituose kompiuteriuose esančios atmintis nėra pasiekama.
- Šiuo metu mišrios architektūros kompiuteriai naudojami HPC sistemose.

IBM Blue Gene/L (Nr. 1 top500.org 2004-2008)



- Greitis – 478,2 TFlops. Theoretical peak – 596,3 TFlops
- procesorių skaičius: 212.992

IBM Blue Gene/L



IBM Blue Gene/L

- Manufacturer : IBM
- Number processors : 212.992
- Processor type : PowerPC 440 core with FP enhancements,
700 MHz, 2.8 GFLOPS (peak)
- Number nodes : 65.536 (each with 2 processors)
- Main memory : 32.768 TB (0.5 GB per node)
- Disk space : 700 TB, 1,6 PB background memory
- Space requirement : 2.500 square feet (232.25 m²)
- Power consumption : 1.5 MW

Tianhe-2 (MilkyWay-2) 2013m

**Tianhe-2 (MilkyWay-2) - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C
2.200GHz, TH Express-2, Intel Xeon Phi 31S1P**

Site:	National University of Defense Technology
Manufacturer:	NUDT
Cores:	3,120,000
Linpack Performance (Rmax)	33,862.7 TFlop/s
Theoretical Peak (Rpeak)	54,902.4 TFlop/s
Power:	17,808.00 kW
Memory:	1,024,000 GB
Interconnect:	TH Express-2
Operating System:	Kylin Linux
Compiler:	icc
Math Library:	Intel MKL-11.0.0
MPI:	MPICH2 with a customized GLEX channel

Sunway TaihuLight - 2017 m.

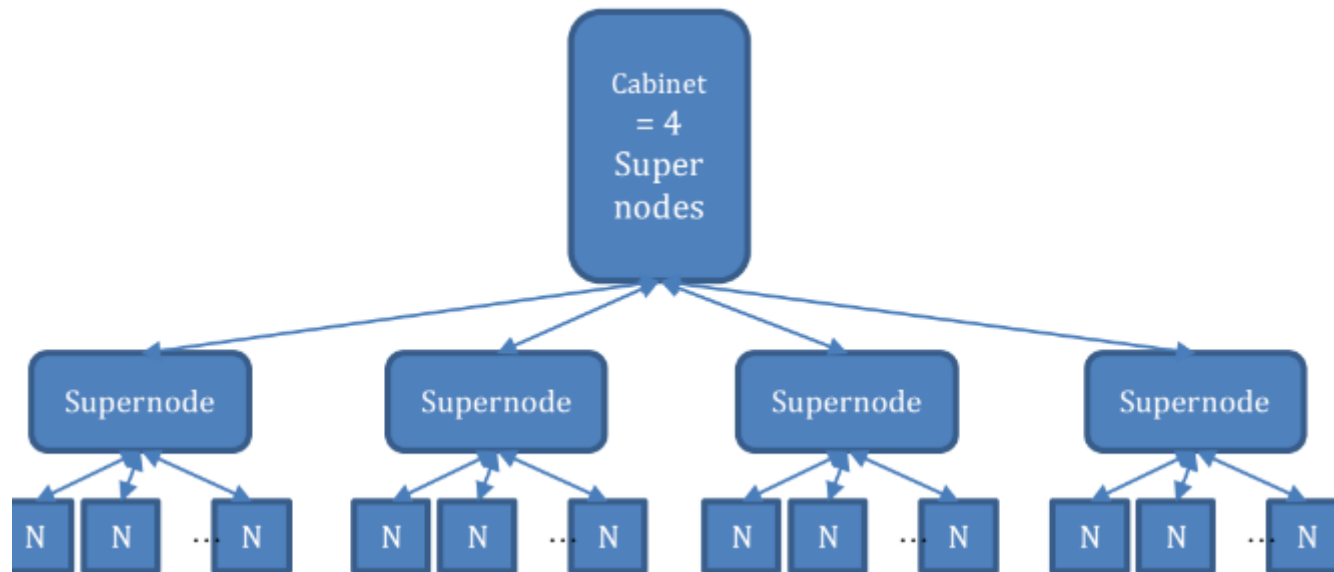
Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C
1.45GHz, Sunway

Site:	National Supercomputing Center in Wuxi
Manufacturer:	NRCPC
Cores:	10,649,600
Memory:	1,310,720 GB
Processor:	Sunway SW26010 260C 1.45GHz
Interconnect:	Sunway
Performance	
Linpack Performance (Rmax)	93,014.6 TFlop/s
Theoretical Peak (Rpeak)	125,436 TFlop/s
Nmax	12,288,000
HPCG [TFlop/s]	480.8
Power Consumption	
Power:	15,371.00 kW (Submitted)
Power Measurement Level:	2
Software	
Operating System:	Sunway RaiseOS 2.0.5

[Sunway TaihuLight - 2017 m.]



[Sunway TaihuLight - 2017 m]



Node = 2.06 Teraflops → 1 supernode (256 nodes) = 783.97 Teraflops →
1 cabinet = 4 supernodes/3.1359 petaflops

One 40 cabinet system (160 supernodes; 40,960 nodes = 10,649,600 cores).