

### 3. Kompiuterio atmintis

Iš pirmo žvilgsnio kompiuterio atminties koncepcija, kurią paskelbė Noimanas, gali atrodyti labai paprasta, tačiau bet kurioje kompiuterizuotoje sistemoje ji byloja apie atminties tipų, technologijų, sandaros ir realizacijos būdų įvairovę ir gausą. Nėra nė vienos technologijos, kuri optimaliai atitiktų visus kompiuterizuotos sistemos atminčiai keliamus reikalavimus. Paprastai tipinė kompiuterio sistema turi hierarchinius atminties posistemius – vidinius sistemos atžvilgiu (procesorius tiesiogiai siejasi su atmintimi) ir išorinius (procesorius sujungtas su atmintimi per įvesties/išvesties (I/I) modulį).

Šiame skyriuje nagrinėjami vidinės atminties elementai, o kitame dėmesys bus sutelktas į išorinę atmintį. Šio skyriaus pradžioje aptarsime pagrindines kompiuterio atminties charakteristikas. Apžvelgsime pagrindinius puslaidininkinės atminties posistemius, tarp jų ROM – Read Only Memory, DRAM – Dynamic Random Access Memory bei SRAM – Static RAM atmintis. Vėl sugrįžę prie DRAM atminties daugiausia dėmesio skirsime šiuo metu pažangiausiai DRAM architektūrai.

#### 3. 1. Kompiuterio atminties sistemų apžvalga

##### 3.1.1. Atminties sistemų charakteristikos

Kompleksinės kompiuterio atminties sąvoka bus labiau suprantama, jeigu atminties sistemas klasifikuosime pagal jų pagrindines charakteristikas. Svarbiausios atminties charakteristikos pateiktos 3.1 lentelėje.

<b>Išdėstymas</b>	Procesoriaus registrai. Vidinė (pagrindinė, darbinė). Išorinė (antrinė)
<b>Greitis</b>	Kreipimosi laikas. Ciklo laikas. Transliavimo (siuntimo) greitis
<b>Talpa</b>	Žodis. Tam tikras žodžių skaičius. Baitas
<b>Fizinis tipas</b>	Puslaidininkinė. Magnetinių ir optinių paviršių
<b>Siuntimo vienetai</b>	Žodis. Blokas
<b>Fizinės charakteristikos</b>	Priklausanti nuo elektros energijos (operatyvioji) / Nuo elektros energijos nepriklausanti (pastovioji). Trinioji/netrinioji
<b>Kreipimosi metodas</b>	Nuoseklusis kreipimasis. Tiesioginis kreipimasis. Laisvasis kreipimasis. Asociatyvusis kreipimasis

##### 3.1.2. Atminties hierarchija

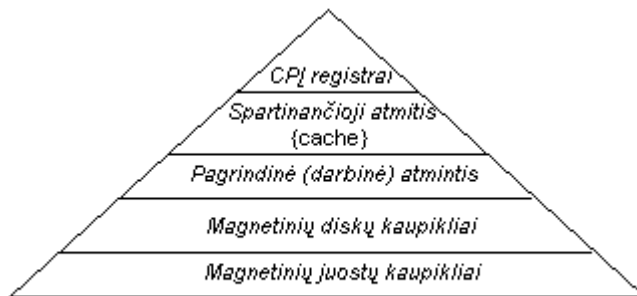
Bendruoju atveju, projektuojant kompiuterio atmintį, reikia iš anksto žinoti atsakymus į tris paprastus klausimus. Kiek daug jos reikia? Kaip sparčiai ji turi veikti? Kiek tai kainuos?

Klausimas „kiek daug?“ visada išliks. Gyvenimas rodo, kad jei jau yra tam tikros talpos atmintis, programos visada bus parašytos būtent taip, kad galėtų panaudoti visą šią talpą. Į klausimą „kaip sparčiai?“ iš esmės galima atsakyti paprastai. Kad kompiuterio sistemos sparta būtų didžiausia, atmintis turi turėti galimybę dirbti procesoriaus greičiu. Kai procesorius vykdo instrukcijas, jokios pauzės kitos instrukcijos arba operandų sulaukti yra nepageidautinos. Klausimas „kiek tai kainuos?“ irgi turi būti atidžiai išnagrinėtas projektuojant. Iš esmės atminties sistemos kaina turi būti racionaliai susijusi su kitų kompiuterio komponentų kainomis.

Tarp trijų pagrindinių atminties charakteristikų – kainos, talpos ir kreipties laiko turi būti dermė. Atminties sistemai realizuoti visais laikais buvo taikoma daug įvairių technologijų. Šių technologijų aibėje galioja tokie santykiai:

- Kuo mažesnė kreipties trukmė, tuo didesnė bito kaina.
- Kuo didesnė talpa, tuo mažesnė bito kaina.
- Kuo didesnė talpa, tuo ilgesnė kreipties trukmė.

Akivaizdu, kad projektuotojas turi spręsti dilemą. Gali taikyti tokias atminties technologijas, kurios užtikrina didelę atminties sistemų talpą. Mat rodos – reikalinga didelė talpa, ir kaina už vieną bitą bus maža. Tačiau sparčiai kompiuterio sistemai suprojektuoti projektuotojai turi taikyti palyginti mažesnės talpos atmintį su trumpa kreipties trukme.



**3.1 pav.** Kompiuterio atminties hierarchija

Tokiems prieštaringsiems reikalavimams įgyvendinti taikomas hierarchinis atminties organizavimas. Tipiška atminties hierarchija parodyta 3.1 pav.

Kuo žemesnė hierarchijos pakopa:

- Vieno bito kaina mažėja.
- Didėja talpa.
- Didėja kreipties trukmė.
- Mažėja procesoriaus kreipčių į atmintį dažnis.

Todėl mažesnės, žymiai brangesnės, sparčiosios atminties sistemos yra papildomos didesnės talpos pigesniais ir lėtesniais atminties posistemiais. Pagrindinis veiksnys, lemiantis šio organizavimo efektyvumą – kreipčių į atmintį dažnumo mažėjimas.

## 3. 2. Pagrindinė puslaidininkinė atmintis

Senesniuosiuose kompiuteriuose labiausiai paplitusi pagrindinės kompiuterio laisvosios kreipties atminties forma buvo feromagnetinių toroidų, vadintų šerdimis (branduoliu) {core}, ir laidininkų matricos. Būtent todėl pagrindinė atmintis buvo vadinama „korine“, t. y. pastoviai asocijuojama su šerdimis, šis terminas vartojamas iki šiol. Laikui bėgant puslaidininkinė atmintis tobulėjo ir nukonkuravo magnetinę „korinę“ atmintį. Dabar puslaidininkinių mikroschemų naudojimas pagrindinei atminčiai yra universalus. Toliau pateikta pagrindiniai puslaidininkinės atminties technologijos aspektai.

### 3.2.1. Laisvosios kreipties puslaidininkinės atminties tipai

Visi šiame skyriuje aprašyti atminties tipai yra laisvosios kreipties, t. y. bet kuris atminties žodis tiesiogiai pasiekiamas esant adresavimo mechanizmui.

3.2 lentelėje pateikti pagrindiniai puslaidininkinės atminties tipai. Labiausiai paplitęs – *laisvosios kreipties atmintis* {Random Access Memory, RAM}. Savaime aišku, kad tai įsigalėjusios terminijos klaida, nes visi lentelėje pateikti atminties tipai yra laisvosios kreipties.

**3.2 lentelė.** Puslaidininkinės atminties tipai

Atminties tipas	Kategorija	Išvalymas	Rašymo mechanizmas	Priklausomybė nuo elektros energijos
laisvosios kreipties atmintis {RAM}	skaitymo ir rašymo atmintis	elektra, baitų lygmeniu	elektrinis	priklausoma
pastovioji atmintis (tik skaitoma) {ROM}			šablonai {Masks}	nepriklausoma
programuojamoji pastovioji atmintis {PROM}	tik skaitoma atmintis	neįmanomas	elektrinis	
trinioji {Erasable} programuojamoji atmintis {EPROM}	daugiausia skaitoma atmintis	ultravioletine šviesa, mikroschemos lygmeniu		
„Fleš“ {Flash} atmintis		elektra, duomenų blokų lygmeniu		
elektra išvaloma programuojamoji atmintis {EEPROM}	{Read mostly memory}	elektra, baitų lygmeniu		

### 3.2.2. Puslaidininkinės atminties sandara

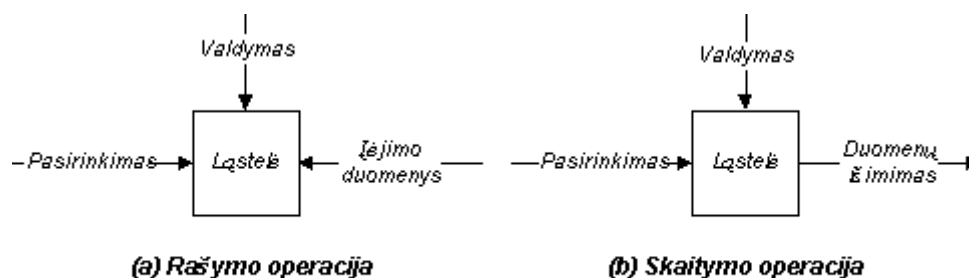
Pagrindinis puslaidininkinės atminties elementas yra atminties ląstelė. Nepaisant to, kad įvairių technologijų daug, visos puslaidininkinės atminties ląstelės turi keletą bendrų savybių. Jos gali būti:

vienos iš dviejų stabilių loginių būsenų, taikytinų binariniam 1 arba 0 pavaizduoti.

įrašytos (nors vieną kartą) – nustatytos į tam tikrą loginę būseną.

nuskaitytos siekiant sužinoti jų loginę būseną.

3.2 pav. pateikta dvi galimos operacijos su atminties ląstelėmis. Paprastai ląstelė turi tris funkcionalių kontaktus elektriniams signalams priimti arba išsiųsti (valdymo organus). Pasirinkimo kontaktas {select terminal}, kaip matyti iš pavadinimo, nurodo konkrečią atminties ląstelę įrašymo arba skaitymo operacijai. Valdymo kontaktas {control terminal} nurodo, kuri operacija bus vykdoma – skaitymo ar rašymo. Rašymui į trečiąjį kontaktą tiekama loginė 1 arba 0 atitinkanti įtampa. Skaitymo metu iš šio kontakto gaunamas atitinkamas ląstelėje įrašyto bito įtampos lygis. Vidinės atminties sutvarkymas, funkcionavimas, laiko diagramos priklauso nuo taikytos technologijos ir yra aprašyti šių technologijų specifikacijose. Nagrinėdami toliau tarkime, kad kiekviena atminties ląstelė gali būti pasirinkta skaitymui arba rašymui.



3.2 pav. Operacijos su atminties ląstele

## 4. Spartinančioji atmintis

Šiame skyriuje apžvelgsime spartinančiąją atmintį (angl. cache – spartinančioji atmintis). Iš pradžių nagrinėsime supaprastintą spartinančiosios atminties modelį. Toliau šis supaprastintas modelis bus plėtojamas: aprašomas spartinančiosios atminties veikimas, aptariami jos vykdomi ciklai, pateikiama apibendrinta jos architektūra, vardijamos svarbiausios komponentės ir bendros organizavimo schemos.

Kompiuteriuose spartinančiosios atminties realizacijos gali labai skirtis nuo aptariamų, nors jų koncepcija iš esmės ta pati. Todėl konkrečia kompiuterizuota sistema griežtai nebus remiamasi.

### 4.1. Spartinančiosios atminties sistemų apžvalga

Prieš aptariant pagrindines spartinančiosios atminties rūšis reikia išsiaiškinti, kas gi ši atmintis yra apskritai. Tai nedidelės talpos labai sparti atmintis (dažniausiai sudaroma iš statinės operatyviosios atminties {SRAM} mikroschemų), kurioje saugomi ypač dažnai naudojami pagrindinės atminties fragmentai.

Kam ši spartinančioji atmintis reikalinga ir kokia jos nauda? Šiuolaikiniuose kompiuteriuose trukmė, būtina instrukcijai (arba duomenims) įkelti į procesorių, labai ilga, palyginti su instrukcijos vykdymo trukme. Pvz., būdingiausias procesoriaus keitimosi duomenimis su dinamine operatyviaja atmintimi DRAM laikas – 60 ns (SDRAM – 15 ns). 1000 MHz taktinio dažnio Pentium® procesorius daugelį instrukcijų vykdo per vieną ciklą, t. y. per 1 ns. Todėl silpnoji vieta (angl. bottle-neck – butelio kaklelis) formuojasi procesoriaus įėjime. Spartinančioji atmintis gelbsti trumpinant keitimosi duomenimis tarp procesoriaus ir dinaminės atminties trukmę. Įprastinės kreipties į SRAM trukmė yra 3–10 ns. Todėl spartinančioji atmintis suteikia galimybę kreiptis į mažus pagrindinės atminties fragmentus 4–10 kartų greičiau nei DRAM mikroschemos (pagrindinė atmintis).

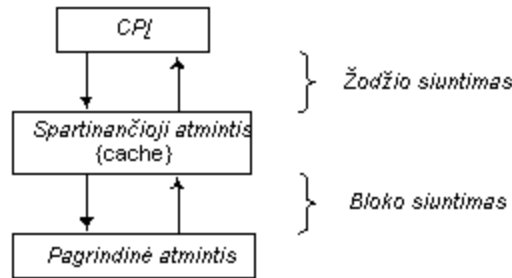
Kyla klausimas – kaip gali mažos talpos spartinančioji atmintis pagerinti viso kompiuterio galimybes? Teorija, aiškinanti šias galimybes, angl. vadinama Locality of Reference – „kreipčių lokalizavimas“. Jos pagrindinė koncepcija: **bet kuriuo laiko momentu tam tikra pagrindinės atminties dalis (manoma, kad 10–20%) gali būti reikalinga procesoriui (su 80–90% tikimybe)**. Spartinančioji atmintis įsikelia šią dalį ir tuomet procesorius greičiau operuoja visa kompiuterio atminties sistema.

Tyrimai rodo [8], kad vykdamą įprastas užduotis Pentium® procesorius savo 16 K vidinėje spartinančiojoje atmintyje saugo per 90% visų procesoriui būtinų adresų. Tai reiškia, kad daugiau kaip 90% kreipčių į atmintį bus atlikta per spartinčiąją atmintį.

Kyla klausimas – kodėl pagrindinėje atmintyje naudojamos DRAM, o ne SRAM mikroschemos? Pagrindinė priežastis – kaina. Statinės atminties mikroschemos keliskart brangesnės už dinaminės atminties mikroschemas. Be to, statinė atmintis naudoja žymiai daugiau energijos ir užima daugiau vietos. Taigi paminėję spartinančiosios atminties naudojimo priežastis, nagrinėsime apibendrintą spartinančiosios atminties modelį.

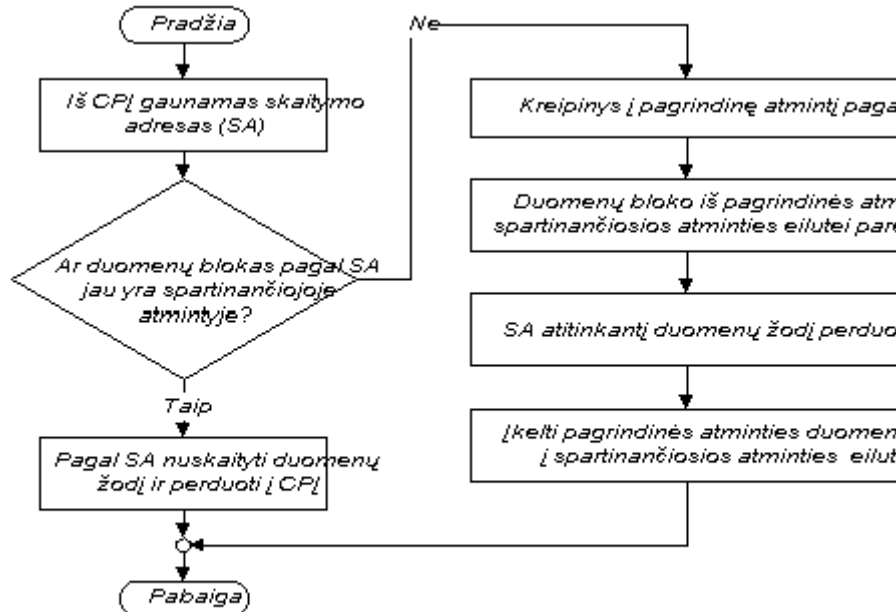
### Apibendrintas spartinančiosios atminties modelis

4.1 pav. pavaizduota kompiuterio su spartinčiąja atmintimi dalies supaprastinta schema. Tokiame kompiuteryje kiekvieną kartą, kai procesorius skaito iš atminties arba rašo į ją, spartinčioji atmintis gali pertraukti procesoriaus magistralės transakciją (duomenų kaitos operaciją), taip sumažėja kompiuterio atsako trukmė.



4.1 pav. Apibendrintas spartinančiosios atminties modelis

4.2 pav. parodytas skaitymo operacijos algoritmas kompiuterio sistemoje su spartinčiąja atmintimi. CPJ generuoja reikiamo duomenų žodžio skaitymo adresą (SA). Jei šis žodis jau yra spartinančiojoje atmintyje – jis perduodamas į CPJ. Priešingu atveju duomenų blokas su tuo žodžiu iš pradžių įkeliamas į spartinčiąją atmintį ir tik tada perduodamas į CPJ.



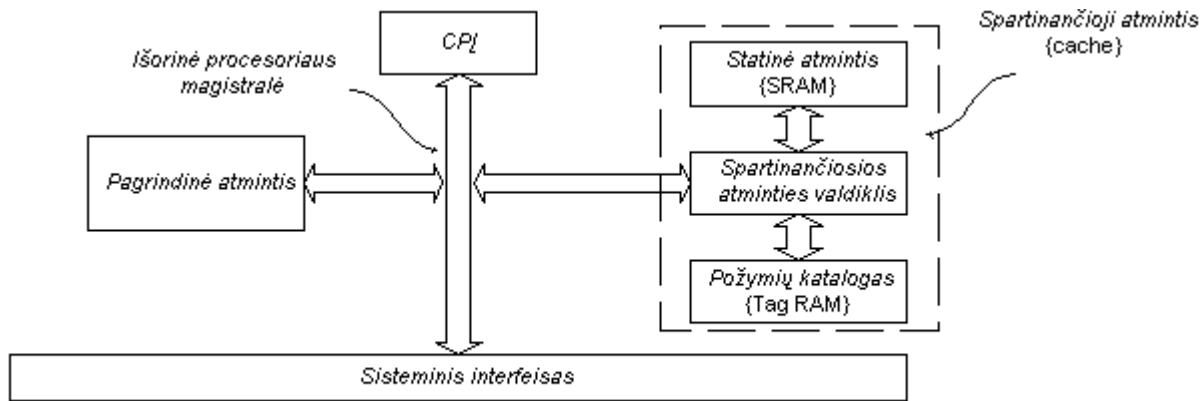
4.2 pav. Skaitymo iš spartinančiosios atminties operacija

## 4.2. Spartinančiosios atminties architektūra

Visos spartinančios atmintys apibūdinamos dviem charakteristikomis: *skaitymo architektūra* ir *įrašymo metodu*. Skaitymo architektūra gali būti arba „peržiūros iš šalies“ {Look Aside}, arba „ištinės peržiūros“ {Look Through} tipo. Įrašymo metodai taip pat gali būti dvejopi – arba „atgalinis rašymas“ {Write Back}, arba „ištinis rašymas“ {Write Through}. Kiekviena skaitymo architektūra, pagal kompiuterio organizavimą, gali būti pagrįsta bet kurio įrašymo metodo taikymu. Įrašymo metodai detalčiau bus aptariami kitame poskyryje. Dabar išnagrinėsime skaitymo architektūrą.

#### 4.2.1. „Peržiūros iš šalies“ architektūra

4.3 pav. parodyta spartinančiosios atminties „peržiūros iš šalies“ {Look Aside} architektūros schema. Šioje schemoje pagrindinė atmintis išdėstyta priešpriešais procesoriaus magistralei. Šios architektūros skiriamasis bruožas yra tai, kad spartinančioji atmintis veikia lygiagrečiai su pagrindine atmintimi. Būtina pabrėžti, kad ir pagrindinė atmintis, ir spartinančioji atmintis „pastebi“ magistralės kreipčių į atmintį ciklus tuo pačiu metu. Tai atitinka atminties pavadinimą – „peržiūra iš šalies“.



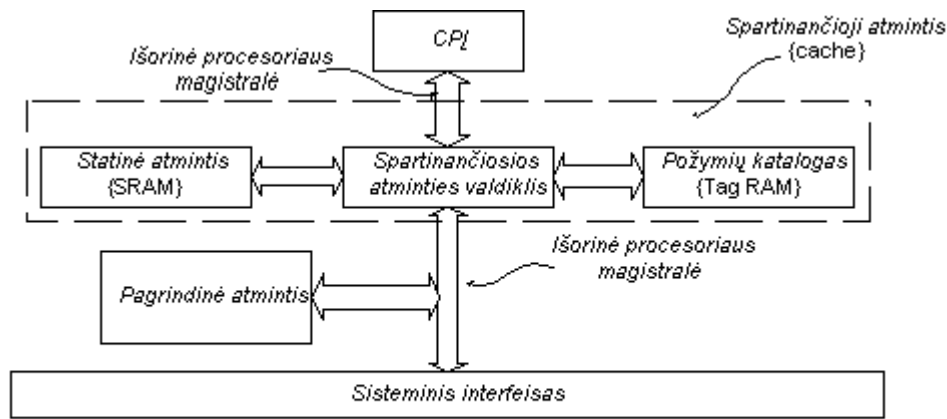
4.3 pav. Spartinančiosios atminties „peržiūra iš šalies“ {Look Aside} architektūra

„Peržiūros iš šalies“ spartinančiosios atminties architektūra nėra labai sudėtinga, tad ir nebrangi. Be to, ši architektūra užtikrina geresnį (t. y. greitesnį) sistemos atsaką į neigiamus spartinančiosios atminties rezultatus {miss}, kadangi ir dinaminė operatyvioji atmintis, ir spartinančioji atmintis vienu metu pastebi magistralės ciklą. Neigiamas bruožas yra tai, kad procesorius negali kreiptis į spartinančiąją atmintį tuo metu, kai kitas magistralės valdiklis kreipiasi į pagrindinę atmintį.

#### 4.2.2. „Ištisinės peržiūros“ architektūra

4.4 pav. parodyta supaprastinta „ištisinės peržiūros“ {Look Through} spartinančiosios atminties architektūros schema. Pagrindinė atmintis čia taip pat išdėstyta kitoje procesoriaus magistralės pusėje. Šios spartinančiosios atminties skiriamasis bruožas yra tai, kad ji išdėstyta tarp procesoriaus ir pagrindinės atminties. Svarbu pažymėti, kad spartinančioji atmintis „stebi“ procesoriaus magistralės ciklus prieš jiems patekus į sisteminę magistralę.

Esant šiai architektūrai procesorius gali dirbti su spartinančiąja atmintimi tuo metu, kai kitas kompiuterio magistralės valdiklis kreipiasi į pagrindinę atmintį. Taip procesorius esti „izoliuotas“ nuo kompiuterio sistemos laukimo (angl. rest – poilsio) stadijų. Bet ši spartinančiosios atminties architektūra daug sudėtingesnė, nes ji turėtų kontroliuoti užklausas į atmintį, kai kompiuterio sistema laukimo stadijoje. Visa kompiuterio sistema sudėtingesnė, tad ir brangesnė. Kitas šios architektūros trūkumas yra tai, kad kreipimasis į pagrindinę atmintį, kai spartinančiojoje atmintyje nėra būtinų duomenų, sulėtėja, nes pagrindinė atmintis nepasiekama, kol nėra patikrinta spartinančioji atmintis (tam reikia papildomų magistralės ciklų). Tai nėra didelė kliūtis, jei spartinančiosios atminties požymių katalogas {Tag RAM} pakankamai spartus ir jam skirtas specialus magistralės valdiklis.



4.4 pav. „Ištisinės peržiūros“ {Look Through} spartinančiosios atminties architektūra

### 4.3. Įrašymo metodai

Įrašymo metodai {Write Policy} lemia, kaip pasiekama spartinančioji atmintis įrašymo ciklo metu. Yra du pagrindiniai įrašymo metodai – *atgalinis įrašymas* {Write-Back} ir *ištisinis įrašymas* {Write-Through}.

Pagal atgalinio įrašymo {Write-Back} metodą spartinančioji atmintis veikia kaip buferis, t. y. procesorius pradeda įrašymo ciklą, spartinančioji atmintis priima duomenis ir nutraukia šį ciklą. Spartinančioji atmintis įrašo duomenis į pagrindinę atmintį tik tada, kai laisva procesoriaus magistralė. Šis metodas užtikrina kompiuterio sistemos spartą, nes procesorius gali tęsti darbą, o pagrindinė atmintis atnaujinama vėliau. Dėl tokios įrašymo į pagrindinę atmintį kontrolės spartinančioji atmintis sudėtingėja, tad ir jos kaina didėja.

Kitas metodas vadinamas *ištisiniu įrašymu* {Write-Through}. Kaip seka iš pavadinimo, procesorius duomenis į pagrindinę atmintį įrašinėja visada per spartinančiąją atmintį. Spartinančiosios atminties turinys gali atsinaujinti, tačiau įrašymo ciklas nesibaigia tol, kol šie visi duomenys įrašomi į pagrindinę atmintį. Šis metodas ne tiek sudėtingas ir todėl ne toks brangus. Kompiuterio sistemos, taikančios šį metodą, galimybės mažesnės, nes procesorius turi laukti, kol duomenys pasieks pagrindinę atmintį. (Teigiamas efektas gaunamas duomenis skaitant.)