

Atmiņas elementi

Atmiņas pamatuzdevums -pēc iespējas ātri nodrošināt procesoru ar vajadzīgo informāciju - komandām un datiem, kas nepieciešami uzdotā darba veikšanai, kā arī uzglabāt informāciju, ko dators saņem no ievada -izvada ierīcēm. Datoru atmiņā informācija tiek glabāta un noteikta pēc tās atrašanās vietas jeb adreses, atskaitot zemāk apskatītos speciālos atmiņas veidus.

Datoru atmiņa pēc uzbūves dalās divās grupās:

- pastāvīgā atmiņa, kura pēc barošanas izslēgšanas nezaudē informāciju;
- operatīvā atmiņa, kura informāciju saglabā tikai pie ieslēgtas barošanas.

Operatīvā atmiņa

Operatīvā atmiņa tiek veidota uz pusvadītājiem gan kā statiskā, gan dinamiskā atmiņa. To izmanto bieži mainīgu programmu un datu glabāšanai.

Par *statisko atmiņas elementu* gan bipolāro, gan MOP tranzistoru gadījumā izmanto triggeru (9.1. b) zīm). Tas glabā informāciju, kamēr ir ieslēgta barošana.

Glabāšanas režīmā vārda kopne -atmiņas šūnas adrese vienam triggera plecam pievienoto slēdža tranzistoru T5 uztur aizvērtā stāvoklī, un vienā no triggera pleciem plūst strāva, kuras lielumu nosaka MOP tranzistora noplūdes strāva.

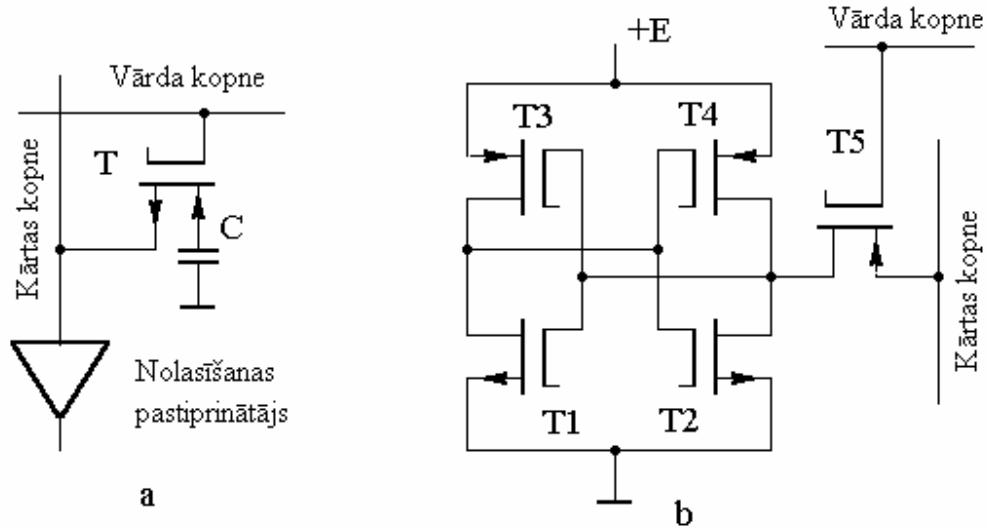
Nolasīšanas režīmā vārda kopnē padod nelielu spriegumu (ap 1-1,5V), kas nedaudz atver tranzistoru T5 un nodrošina informācijas nolasīšanu bez triggera stāvokļa maiņas. Kārtas kopnē padoto spriegumu saņem nolasīšanas pastiprinātājs un fiksē vieninieka kodu augsta sprieguma līmeņa gadījumā un nulles kodu pretējā gadījumā.

Ierakstīšanas režīmā kārtas kopnē vajadzīgo spriegumu iestāda ierakstīšanas pastiprinātājs. Vārda kopnē padotais spriegums pilnīgi atver slēdža tranzistoru T5, un tāds pats sprieguma līmenis iestādās kārtas kopnei pievienotajā triggera plecā neatkarīgi no iepriekšējā triggera stāvokļa.

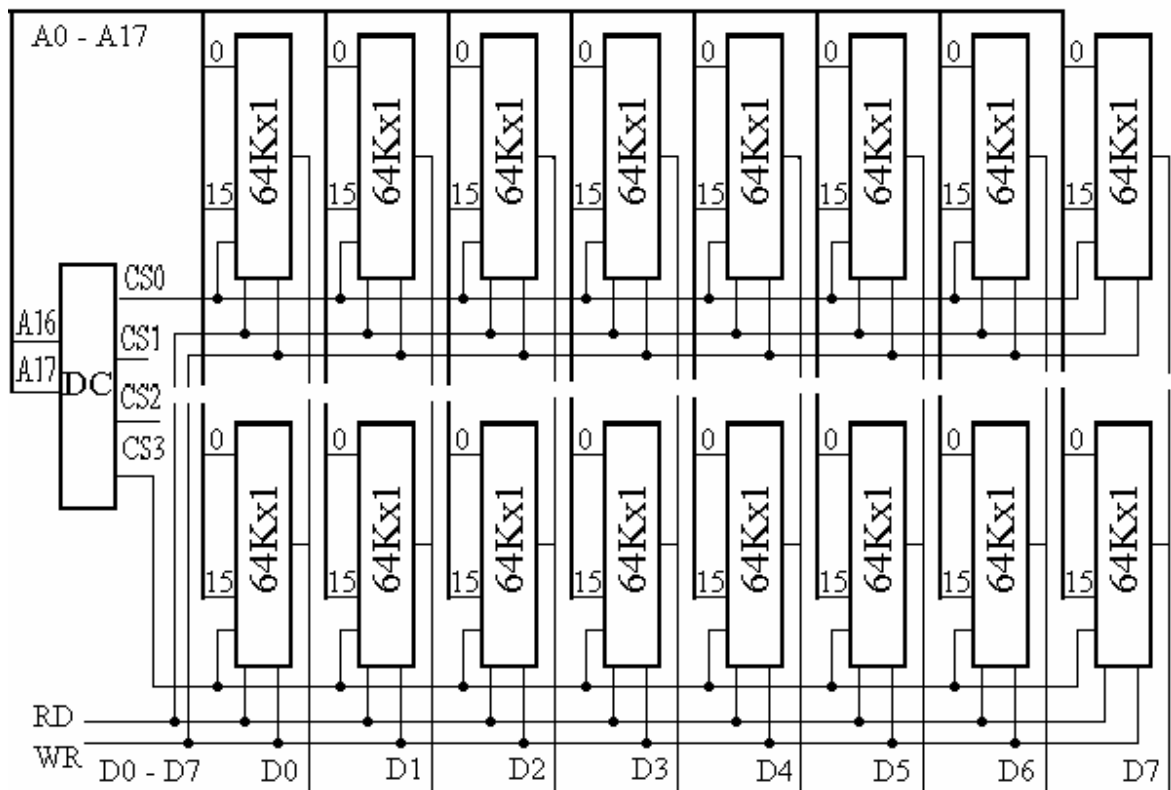
Atmiņas moduļa veidošana ar 64 Kbitu integrālajām shēmām notiek, visu shēmu korpusiem pievadot adresu kārtas A0 ÷ A15, lasīšanas un rakstīšanas atļaušanas signālus (9.2. zīm.). Korpusa izvēles signālu izstrādā ārējais dešifrators, un to pievada astoņām shēmām, kas veido baitu, atļaujot nolasīt tikai šo baitu. Savukārt katra vārda viena bita datu izvadi savienoti pa kolonām.

Čipa, Elektroniskie skaitļotāji, 139. lpp

Dinamiskajos atmiņas elementos informāciju glabā kā MOP tranzistoram pievienotas kapacitātes lādiņu (9.1. a) zīm.). Atmiņas elementam barošana



9.1. zīm. Operatīvās atmiņas elements: a -dinamiskais; b -statiskais

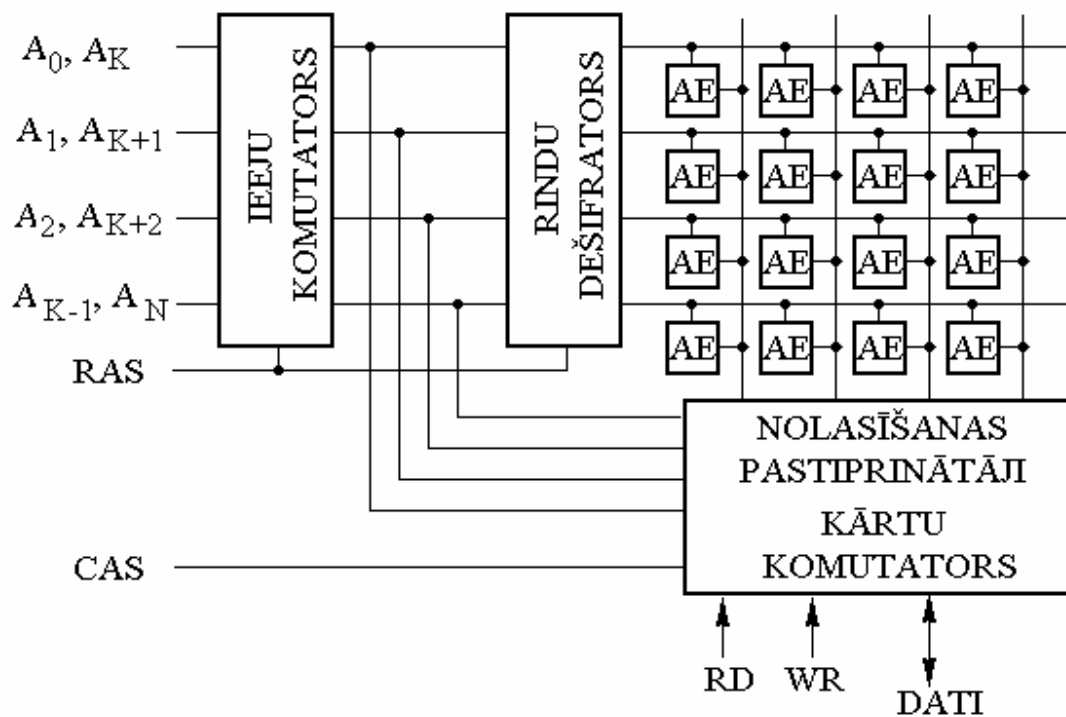


9.2. zīm. 256 Kbaiti statiskās atmiņas moduļa struktūra

principā nepieciešama tikai informācijas ierakstīšanas, reģenerācijas jeb atjaunošanas un nolasīšanas laikā, un kā periodisks barošanas avots tam noder kārtas kopnes montāžas kapacitāte $C_y > C$. Šodien atmiņas elementu veido ar vienu tranzistoru, tāpēc tas vairāk piemērots lielu atmiņas tilpumu ievietošanai vienā korpusā.

Ierakstīšanas režīmā kārtas kopnē vajadzīgo spriegumu iestāda ierakstīšanas pastiprinātājs, un vārda kopnē padotais spriegums atver tranzistorus T, uzlādējot atmiņas elementa kapacitāti C līdz kopnes sprieguma līmenim.

Nolasīšanas režīmā vārda kopnē padod spriegumu, kas atver visus kopnei pievienotos atmiņas elementu tranzistorus, un kārtas kopnēs iestādās spriegums, kuru nosaka informācijas atcerēšanās kapacitāte C. To saņem nolasīšanas pastiprinātāji un fiksē vieninieka kodu augsta sprieguma līmeņa gadījumā un nulles kodu pretējā gadījumā. Nolasot informācija zūd, tādēļ nolasīšanas pastiprinātāju izeju pievieno atpakaļ kārtas kopnēm, un seko informācijas ierakstīšana. Vajadzīgo atmiņas elementu nosaka multipleksors, kuram pievienotas visas kārtas kopnes.



9.3. zīm. Dinamiskās atmiņas shēmas korpusa struktūra

Glabāšanas režīmā vārda kopnē nav spriegums, un tranzistori T_i ir aizvērti. Bet uzlādēto kapacitāti pamazām izlādē noplūdes strāvas, tādēļ informācijas uzglabāšanai tas ik pēc 1-2 ms jāatjauno, ko realizē ar nolasīšanas ciklu un tam sekojošu ierakstīšanas ciklu visos vārda kopnei pievienotajos atmiņas elementos vienlaicīgi. Šajā gadījumā kārtas kopnes izejas multiplexoram nepievieno, un nekāda informācija mikroshēmas izejā neparādās. Tāpēc reģenerāciju var izpildīt visos atmiņas shēmu korpusos vienlaicīgi.

Dinamiskās atmiņas shēmas korpusi parasti paredzēti viena bita glabāšanai, un tā atmiņas elementi veido kvadrātveida matricu (9.3. zīm.). Uz tā ieeju padod pusi no visa adresu bitu skaita $A_0 \div A_{K-1} = A_{N/2}$, kas tiek strobēti ar

rindu izvēles signālu RAS, un pēc adresu rindas dešifrācijas tiek nolasīta informācija no visiem vienas rindas atmiņas elementiem. Pēc tam mikroshēmas ieejā padod otru pusi no visa adresu bitu skaita $A_K \div A_N$, kurus strobē kolonu izvēles signāls CAS, un pēc kolonu dešifrācijas mikroshēmas izejai pieslēdz tikai vienu izvēlēto kolonu. Šajā laikā notiek arī nolasītās informācijas ieraksts atpakaļ visā nolasītajā rindā, izpildot dotās rindas reģenerāciju.

Rakstīšanas režīmā rindas nolasīšana notiek tāpat, tikai ierakstāmajā bitā atpakaļ ieraksta to informāciju, kura no datu ieejas caur kārtu dešifratoru padota uz attiecīgo kolonas kopni.

Reģenerācija savukārt raksturojās ar to, ka rindas adrese un signāls RAS tiek padots visiem atmiņas shēmu korpusiem vienlaicīgi, bet netiek padots neviens signāls CAS, kādēļ visu mikroshēmu izejas ir atvienotas no datu kopnes. Tāpēc reģenerācija jāizpilda tikai tik reizes, cik korpusā rindu.

Šāda atmiņas shēmu organizācija būtiski samazina atmiņas shēmas korpusa izvadus skaitu, un jau 16 izvadus korpusā atļauj realizēt 256 Kbiti dinamisko atmiņu (1 dati + 4 vadības + 2 barošanas + 2x9 adreses signāli).

Čipa, Elektroniskie skaitļotāji, 141. Lpp

Pastāvīgā atmiņa

Pastāvīgā atmiņa principā var tikt apskatīta kā kodu pārveidotājs, kas adreses koda k kārtām piekārto informācijas koda n kārtas. Tāpēc pastāvīgo atmiņu parasti veido kā baitu struktūru un savstarpēji perpendikulāru kopņu sistēmu, kuru krustpunktos atrodas atmiņas elementi. Ja elements abas kopnes savieno, dotajā kārtā ierakstīts vieninieks, bet ja nesavieno, ierakstīta nulle. Vienkāršās

tehnoloģijas dēļ šodien par saites elementu starp kopnēm izmanto pāreju starp jebkuriem diviem kontaktiem MOP struktūrā, bet bipolāros atmiņas elementus uz diodēm un tranzistoriem izmanto tikai speciālos gadījumos.

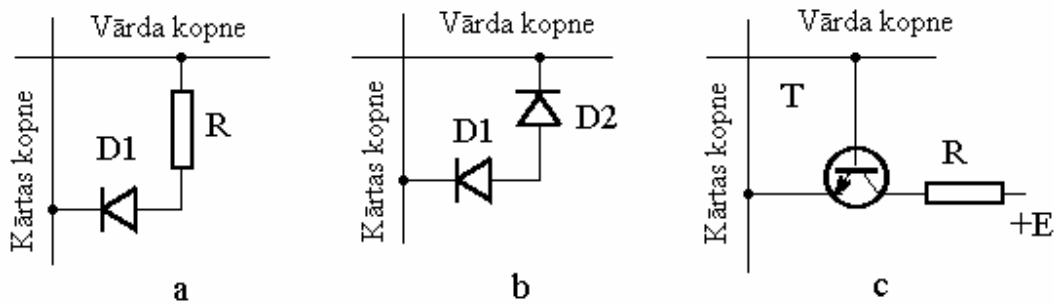
Konstruktīvi pastāvīgās atmiņas raksturo šādi parametri:

- atmiņas shēmu korpusi ir ar baitu organizāciju;
- minimāls integrālās shēmas korpusa izvadu skaits;
- iespēja izgatavot lielu atmiņas tilpumu vienā korpusā;
- konkrētās informācijas ierakstīšanai papildus nepieciešams tikai viens fotošablons.

Pastāvīgā atmiņa tiek lietota nemainīgu programmu glabāšanai. Tās ir mikroprogrammas procesoru vadības iekārtās, darba programmas specializētās iekārtās un dažādas palīgprogrammas universālos datoros. Personālajos datoros tās ir inicializācijas programmas, bet vadības kontroleros -darba programmas, kas tur ievietotas uz visu tā mūžu.

Pastāvīgo atmiņu uz diodēm veido kā baitu vai vārdu struktūru, kurā adrese nosaka baita vai vārda izvēli. Viena bita glabāšanai paredzēta elementa uzbūve parādīta 9.4.a un 9.4.b zīmējumos. Izvēloties vārdu, vārda kopnē padotais augstais sprieguma līmenis caur diodi D1 nonāk kārtas kopnē un kā vieninieka vērtība tiek nolasīts shēmas izejā. Pārējās neizvēlēto vārdu kopnēs sprieguma līmenis ir zems, bet kārtas kopne no tām ir atvienota ar pretēji slēgtu diodi. Vienreiz ieprogrammējamai atmiņai virknē ar diodi slēgts kustošs drošinātājs, kuru ar stingri dozētu strāvas impulsu izkausē, ja dotajā elementā jāieraksta nulle (5.3.a zīm.), t.i., augsts sprieguma līmenis kārtas kopnē nenonāk. Citā variantā atmiņas elementu veido divas pretēji slēgtas diodes (9.4.b zīm.), un programmēšanas gaitā diodi D2 caursit un veido vadošu ķēdi, ar to dotajā elementā ierakstot vieninieku.

Pastāvīgās atmiņas elementu ar bipolāro tranzistoru izmanto, lai samazinātu programmēšanai nepieciešamo strāvu vārda kopnē, virknē ar tranzistoru slēdzot drošinātāju (9.4.c zīm.). Pēdējo ar stingri dozētu strāvas impulsu



9.4. zīm. Pārprogrammējami bipolārie pastāvīgās atmiņas elementi:

a - ar kūstošu drošinātāju, b - ar pretēji slēgtām diodēm, c - ar tranzistoru.

izkausējot, dotajā elementā ieraksta nulli. Bet šo elementu trūkums ir lielais jaudas patēriņš, it īpaši programmējot, un līdz ar to zemā integrācijas pakāpe - līdz 8 Kbaiti vienā korpusā. Savulaik šo atmiņas shēmu izplatību noteica to samērā mazais atmiņas cikls -10 līdz 50 ns.

Pastāvīgās atmiņas ar MOP tranzistoriem adrešu dešifratora vienkāršošanas nolūkā MOP tranzistorus galvenokārt izmanto pastiprināšanas režīmā, kas atkarībā no tranzistoru izgatavošanas tehnoloģijas nodrošina informācijas izvēles laiku ap 20-200 ns. Tās rūpnieciski izgatavo trīs variantos.

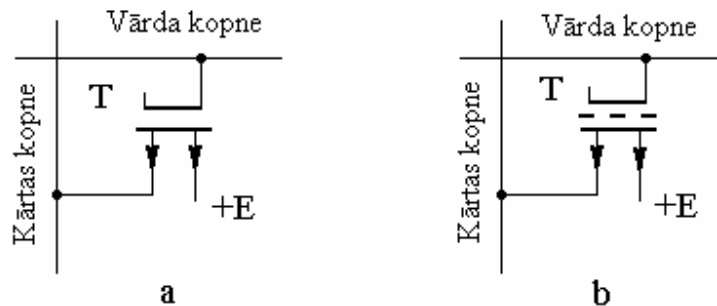
1) Rūpnīcā ieprogrammētās pastāvīgās atmiņas, kuru izgatavošanai izmantots speciāls fotošablons, tās izmanto aparatūras sērijas ražošanā. Vieninieka ierakstīšanu nodrošina normāls tranzistors, bet nulles ierakstam izolāciju zem tranzistora slēdža izgatavo tik biezu, lai adrešu kopnei pieliktais spriegums nevarētu tranzistoru atvērt, t.i., tranzistors faktiski šajā vietā nav (9.4. zīm.).

2) Pie lietotāja vienreiz ieprogrammējamās pastāvīgās atmiņas, kuras lieto sīksēriju ražošanā (9.5. zīm.). To tranzistora konstrukcijā ievie divus slēdžus:

-peldošo slēdzi, kas no visām pusēm izolēts ar biezu SiO₂ kārtu un noder injicētā lādiņa uzglabāšanai;

-izvēles slēdzi, kurš pievienots adrešu kopnei un izmanto pastiprināšanas efektu informācijas nolasīšanai, radot vadošu kanālu starp tranzistora avota un noplūdes elektrodiem. Vadības signālu uz izvēles slēdzi izvēlas tā, lai tas radītu vadošu kanālu starp avota un noplūdes elektrodiem tikai tad, ja uz peldošā slēdža nav injicētā lādiņa (vieninieka ieraksts), un neradītu vadošu kanālu pretējā gadījumā.

Nulles ierakstam MOP tranzistora avota un noplūdes elektrodiem pieliek pietiekami lielu spriegumu (12 -25 V), izolētajā slēdzī notiek elektronu lavīnveida injekcija un slēdzī uzkrājas negatīvs lādiņš. Slēdža uzlādēšanas gaitā pieliktā sprieguma ilgumu stingri dozē (ap 1 ms līdz 32 reizēm), lai novērstu materiāla pārkaršanu un tranzistora caursiti. Uzkrātā lādiņa rezultātā zem slēdža rodas inversais slānis un tranzistora kanāls kļūst vadošs, kas atbilst nulles ierakstam.



9.5. zīm. Atmiņas elements ar MOP tranzistoru:

a -pastāvīgās atmiņas elements; b -pārprogrammējamais ar peldošo slēdzi

3) Pārprogrammējamās pastāvīgās atmiņas, kuru saturu pie lietotāja iespējams daudzkārt nodzēst un atkal ieprogrammēt. To darbība analoga iepriekšējai atmiņas shēmai, tikai tās dzēšanai izmanto ultravioleto starojumu, kurš caur speciālu lodziņu nonāk uz pusvadītāja kristālu un izolācijas materiālā rada vadāmību, tā pakāpeniski izlādējot izolētā slēdža lādiņu.

Čipa, Elektroniskie skaitļotāji, 144, 147. lpp. Dešifrators 53 lpp.

Atmiņas organizācijas veidi

Atmiņas pamatzdevums -pēc iespējas ātri nodrošināt procesoru ar vajadzīgo informāciju - komandām un datiem, kas nepieciešami uzdotā darba veikšanai, kā arī uzglabāt informāciju, ko dators saņem no ievada -izvada ierīcēm. Šo īpašību visiem datoriem nodrošina divas atmiņas funkcionālas struktūras.

ASV matemātiķa Džona fon Neimana 1946.g. formulētais mūsdienu datoru darbības princips nosaka, ka apstrādājamos skaitļus un datora darba programmu, kas attēlota ar skaitļiem, ievieto kopīgā atmiņā, un bez konteksta nav iespējams atšķirt programmu no datiem. Arhitektūra pamatā universāliem datoriem, kuru darba programma ir liela vai kuriem tā bieži jāmaina.

Harvardas arhitektūrā datus un programmu glabā katru savā atmiņā. Par cik programma darba laikā netiek mainīta, tās glabāšanai izmanto lētāko pastāvīgo atmiņu, bet datu un starprezultātu glabāšanai -operatīvo atmiņu. To galvenokārt pielieto iebūvētajos vadības mikrokontroleros, kuros programma ir neliela un tā nav jāmaina, kā arī universālajos datoros inicializācijas programmu un bieži lietojamu īsu apkalpošanas programmu glabāšanai (BIOS).

Otrs šīs arhitektūras pielietojums ir jaudīgu procesoru keša atmiņā, jo striktais abu atmiņu nodalījums vienkāršo vairāku aritmētiski -loģisko mezglu paralēlā darba organizēšanu, kā arī nākamās adreses prognozēšanas mehānismu vadības nodošanas komandās.

Principā no atmiņas organizācijas pat vairāk nekā no procesora atkarīga sistēmas ātrdarbība.

Atmiņas hierarhiskā struktūra

Moderno datoru fiziskās atmiņas izgatavošanu izpilda ar dažādām tehnoloģijām: jo tā ātrāka, jo tā dārgāka, un tās apjoms mazāks; jo tā lielāka, jo lielāks laiks vajadzīgs informācijas atrašanai tajā. Bet procesors visu informāciju apstrādā noteiktā secībā, un tam pašlaik ir vajadzīga tikai neliela daļa no atmiņā ievietotās programmas un datiem. Tāpēc praksē lieto hierarhisku atmiņas organizāciju, speciālai vadības programmai -operacionālajai sistēmai pārsūtot pašreiz vajadzīgo informāciju no liela apjoma lēndarbīgās atmiņas uz nelielu ātrdarbīgu atmiņu.

Tāpēc atmiņa atkarībā no tās pielietošanas intensitātes un glabājamās informācijas apjoma tiek sadalīta vairākos līmeņos (9.6. zīm.). Procesorā un sistēmas mātes platē atmiņu veido uz pusvadītāju integrālo shēmu bāzes ar dažādu ātrdarbību, kamēr ārējā atmiņa tiek veidota uz magnētiskajiem un optiskajiem informācijas nesējiem. Ārējā atmiņa paredzēta ilgstošai informācijas uzglabāšanai, un to veido kā autonomu ārējo iekārtu. Dažādo atmiņas veidu ietilpība parādīta 9.1. tabulā.

9.1. tabula

procesora iekšējie reģistri	8 - 16 vārdi
komandu konveijers	6 - 16 baiti
kešs	2 - 256 Kbaiti
sistēmas atmiņa (Pentium)	1 - 4 Mbaiti
operatīvā atmiņa	70 - 100 ns
statiskā atmiņa	8 - 32 Mbaiti
dinamiskā atmiņa	1 - 64 Kbaiti/korpusā
	1 - 64 Mbiti/korpusā

pastāvīgā atmiņa		
ROM, PROM, EPROM		1 - 64 Kbaiti/korpusā
ārējā atmiņa: diski	10 - 20 ms	0,2 - 2 Gbaiti
disketes		1 - 3 Mbaiti
ZIP iekārtas		100 Mbaiti
CD-ROM		650 Mbaiti
Strīmeri		

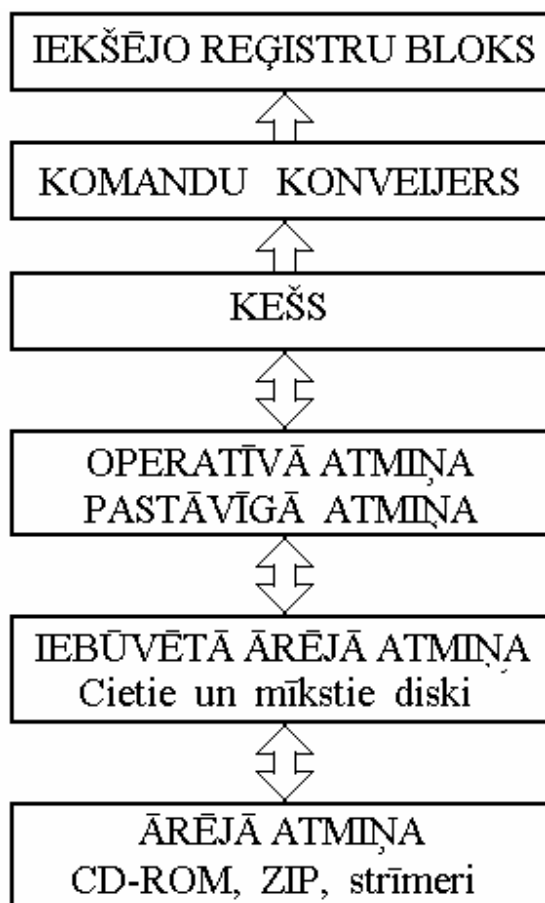
Procesora iekšējie reģistri veido superātru iekšējo atmiņu, kuru izmanto tieši pašreiz lietojamo datu un adresu glabāšanai.

Maģistrāles interfeiss ietilpst visu turpmāko procesoru sastāvā un apmierina visus aritmētiskā procesora maģistrāles pieprasījumus kā pēc komandu kodiem, tā arī datiem. Kamēr aritmētiskais procesors izpilda tekošo komandu, maģistrāles interfeiss, izmantojot komandu skaitītāja nākošo vērtību, patstāvīgi realizē fizisko adresu izskaitļošanu no divām loģiskām adresēm, nākošo komandu kodu izvēli no atmiņas un to ievietošanu speciālā sešu baitu garā konveijera reģistrā QUEUE. Nākošā komanda tiek izvēlēta tad, ja konveijerā ir brīvi kaut 2 baiti. Parasti konveijerā ir kaut viens neizmantots baits, tāpēc aritmētiskajam procesoram uz komandas izvēli no atmiņas nav jāgaida.

Sastopot vadības nodošanas komandu, konveijeru reģistrā iekrātie baiti ir lieki, un tā saturs netiek ievērots. Aritmētiskajam procesoram šajā gadījumā jāgaida, kamēr pirmā komanda tiek ielādēta konveijera reģistrā, un turpmāk notiek automātiska konveijera aizpildīšana. Ja šajā laikā parādās aritmētiskā procesora pieprasījums pēc lasīšanas vai rakstīšanas operācijas, tai vienmēr tiek piešķirta prioritāte.

Rezultātā komandu konveijera efektivitāte samazinās, ko var reducēt uz sekojošiem gadījumiem:

- 1) garā komanda visa neieiet komandu konveijerā, un nākas to papildināt pa daļām;
- 2) tieši operanda izmantošana ļauj izmantot komandu konveijeru kā papildu reģistru;



9.6. zīm. Atmiņas hierarhiskā struktūra

3) vadības nodošanas komandas prasa neievērot komandu konveijerā uzkrātos rezultātus, bet gaidīt uz nākamās komandas nolasīšanu no atmiņas.

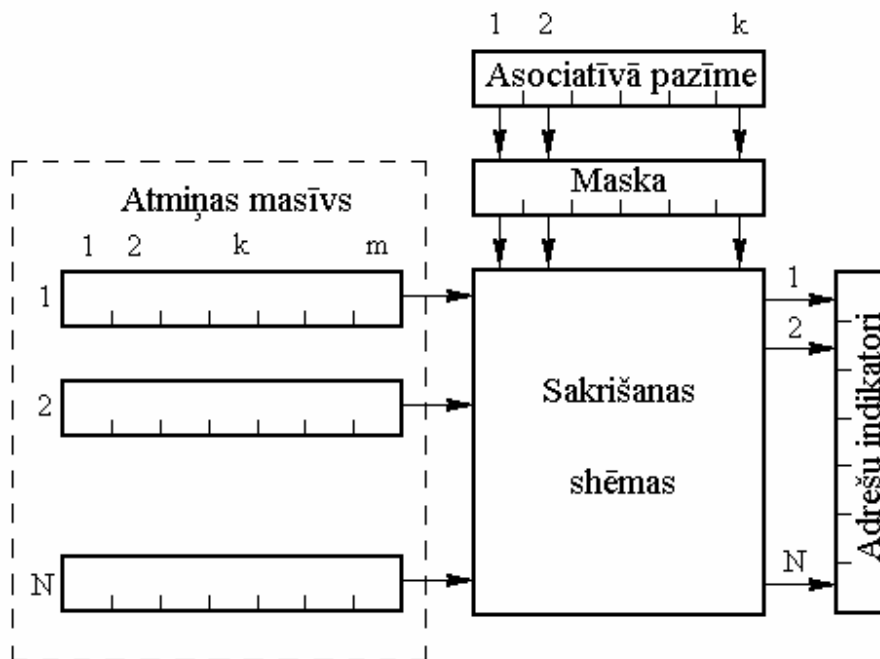
Kešs ir samērā neliela jaudīga mikroprocesora korpusā vai tā tiešā tuvumā novietota ātra atmiņa, kuru izmanto vairāku pašreiz izpildāmo komandu un datu glabāšanai, izmantojot asociatīvo meklēšanas principu.

Visbeidzot datu pārraides ātrumu nosaka arī maģistrāle. Jo ātrāka un platāka maģistrāle, jo lielāks informācijas pārraides ātrums. Datu maģistrāles platums 8, 16(kontroleri), 32 (486), 64 (Pentium) biti. Ātrdarbības palielināšanai lieto vairākas maģistrāles..

Asociatīvā atmiņa

Asociatīvā atmiņa paredzēta informācijas meklēšanai pēc kādas to raksturojošas pazīmes, kas ir daļa no meklējamās informācijas, un to plaši izmanto keša un virtuālās atmiņas organizēšanai. To (9.7. zīm.) veido atmiņas masīvs, kurā

katrs vārds satur vajadzīgo informāciju un asociatīvo pazīmi, pēc kuras šo informāciju meklē. Meklēšanas gaitā visu vārdu asociatīvās pazīmes tiek salīdzinātas ar meklēšanā uzdoto pazīmi sakrišanas shēmās, un adrešu indikatoros paliek vieninieks tikai tajā adresē, kura satur meklējamo informāciju. Šo vārdu tad no atmiņas masīva nolasa.



9.7. zīm. Asociatīvās atmiņas struktūra

Sakrišanas shēmas un nolasīšanas aparatūra aizņem daudz vietas, tādēļ asociatīvā atmiņa parasti ir neliela, un tai galvenokārt ir specifiska pielietošana, piemēram, to lieto keša un virtuālās atmiņas organizēšanai.

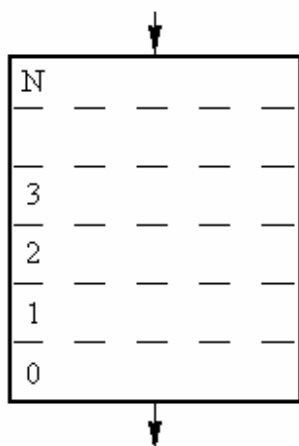
Čipa, Elektroniskie skaitļotāji, 152. lpp

Steka atmiņa

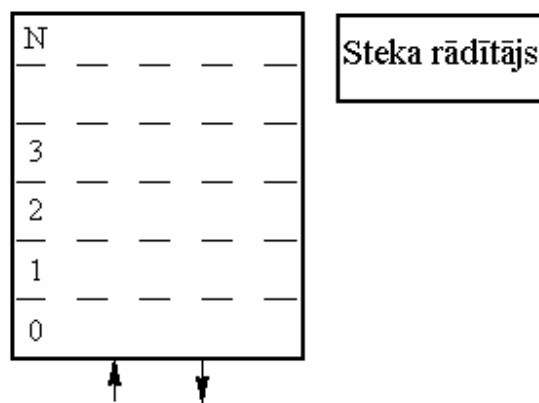
Steka atmiņa ar aparatūras līdzekļiem apkalpo informācijas sistēmas, kurās informācija jāsakārto rindā tās pienākšanas secībā. Sakārtotā informācija secīgi tiek nolasīta, izmantojot divus principus.

Struktūru pirmais ievietots -pirmais apstrādāts izmanto komandu konvejera organizēšanai modernajos mikroprocesoros (9.8. zīm.).

Čipa, Otrās paaudzes mikroprocesoru komplekts..., 4, 10 lpp.



9.8. zīm. Konvejera atmiņa



9.9. zīm. Steka atmiņa

Struktūru pēdējais ievietots -pirmais apstrādāts plaši lieto steka organizēšanai un izmanto:

- programmā paredzētajās apakšprogrammu izsaukšanas procedūrās;
- ar pārtraukumu sistēmu realizētajās apkalpošanas procedūrās;
- savstarpēji saistītu datu struktūru apstrādāšanai.

Neatņemama šīs organizācijas sastāvdaļa ir steka rādītājs (9.9. zīm.), kura garums universālās sistēmās vienāds ar komandu skaitītāju, un kurš atļauj steku ievietot jebkurā atmiņas vietā.

Čipa, Elektroniskie skaitļotāji, 153. lpp

Keša atmiņa

Keša atmiņā izmanto ļoti ātru RAM (10-20 ns), un to lieto kā buferi starp ātrām un lēnām iekārtām. Šodien izmanto divus keša veidus (9.10. zīm.):

- starp procesoru un operatīvo atmiņu;
- starp operatīvo atmiņu un cieto disku.

Keša organizācijā izmanto divus principus:

Tiešajā organizācijā vairākas atmiņas daļas tieši attēlojas ar vienu keša līniju. Atmiņa loģiski tiek dalīta blokos, kuri savstarpēji konkurē par vietu vienā līnijā. Ja uzdevums no dotā bloka prasa vēl informāciju, līnija ir jāatbrīvo.

Asociatīvā organizācijā katru loģisko bloku var glabāt vairākās keša vietās (divās, četrās vai jebkurā). Vajadzīgā informācija parasti vienmēr ir kešā, bet tās meklēšana ir lēna, jo jāmeklē uzreiz divās, četrās vai visās keša līnijās.

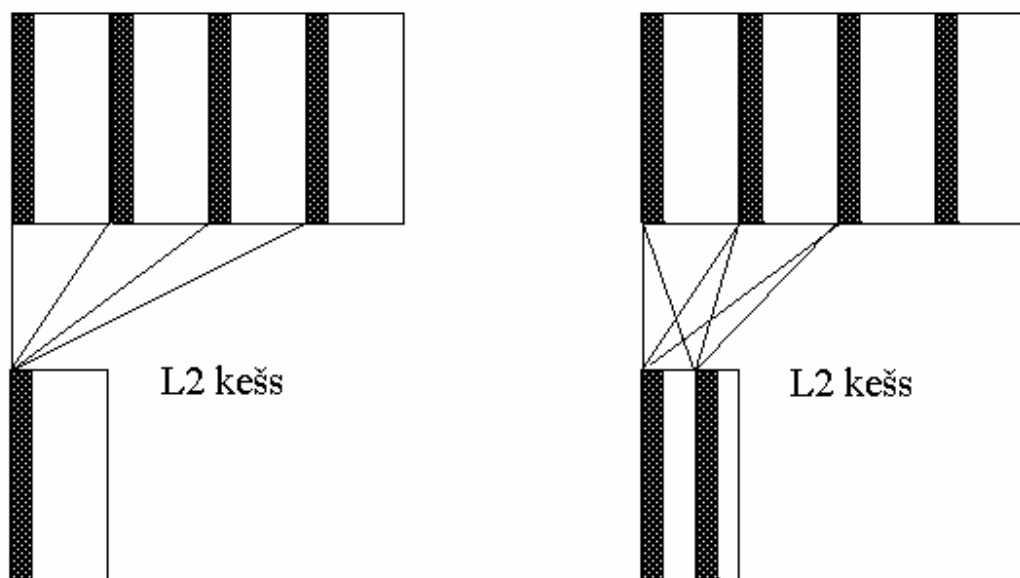


9.10. zīm. Keša atmiņas veidi.

Lietojot keša atmiņu par buferi starp procesoru un operatīvo atmiņu, tā bieži novietota vienā integrālās shēmas korpusā ar procesoru, un tās uzdevums ir saīsināt procesora griešanās ciklu pēc komandām vai datiem. Tā fiziski realizēta kā asociatīvā atmiņa, kurā katra līnija satur informāciju, tegu kā šīs informācijas asociatīvo pazīmi -adresi un bitus, kas norāda, vai dotā līnija kešā ir brīva. Datoros ar atmiņas segmentu organizāciju, piemēram, i80x86, kešs tiek dalīts blokos, katram segmenta tipam atvēlot savu bloku.

Atmiņas lasīšanas ciklā vispirms tiek pārbaudīts, vai dotajai adresei atbilstošā informācija nav kešā, un ja tā ir, procesors to nolasa. Ja informācija nav kešā, aktivizējas maģistrāles lasīšanas cikls, kā arī tiek noteikts, kura keša līnija visilgāk nav lietota un tātad tā ir brīva, un visa līnija tiek aizpildīta no operatīvās atmiņas. Pieprasītā daļa no šīs informācijas nonāk arī procesorā.

Datu rakstīšanas ciklā procesors izmanto divas dažādas tehnoloģijas: ierakstīšanu atpakaļ un MESI (Modified, Exclusive, Shared, Invalid) protokolu. Pirmā tehnoloģija paredz datu ierakstīšanu atpakaļ kešā bez to izvadīšanas uz sistēmas atmiņu, rakstīšanu sistēmas atmiņā realizējot tikai tad, kad dati kešā tiek dzēsti. Caurrakstīšanas tehnoloģija paredz datu rakstīšanu sistēmas atmiņā katru



9.11. zīm. Keša veidi: a) tiešā organizācija; b) divvirzienu asociatīvā organizācija.

reizi, kad tie tika rakstīti kešā. Rezultātā samazinās ārējās maģistrāles pieprasījums, kura ir sistēmas šaurā vieta, sevišķi vairākprocesoru sistēmās.

Pentium procesoros izmanto divus kešus:

- programmu kešs satur komandas, kuru kopija vienmēr atrodas operatīvajā atmiņā, un tā saturu nekad nevajag rakstīt atpakaļ atmiņā;
- datu kešs satur datus un citu mainīgu informāciju, un ja kaut reizi notikusi rakstīšana kešā, tā saturs tiek pārrakstīts operatīvajā atmiņā, keša saturu dzēšot.

Intel Pentium Processors and Related Products.

IBM PC Institute, September 8, 1998

Atmiņas aizsardzība

Atmiņas aizsardzībai lieto:

- pārības kontroli;
- deskriptori aizsargā vienu lietotāju no otra;
- segmenta pārsniegšana, nepareizs segments;
- aizsardzības gredzeni aizsargā sistēmu no lietotājiem;
- aizsardzības bits kontroleru ROM

Čipa Mikroprocesori i80286 un i80386

Visu atmiņu var sadalīt vairākos tipos:

- operatīvā atmiņa bez pārības kontroles nenodrošina kļūdu atpazīšanu un korekciju, praktiski to nelieto;

-operatīvā atmiņa ar pārības kontroli nodrošina kļūdu atpazīšanu, bet nenodrošina to korekciju, un tiek plaši lietota PC un darba stacijās;

- operatīvā atmiņa ar kļūdu korekcijas kodu (ECC -Error Corecting Code) nodrošina kļūdu atpazīšanu un korekciju, un tiek plaši lietota serveros.

Tā detektē un koriģē visas vienkāršās kļūdas (98% visu RAM kļūdu), kā arī uzrāda visas un koriģē dažas dubultās kļūdas, un uzrāda daudzas trīs- un četrkāršās kļūdas bez to korekcijas, bet prasa lietot speciālus atmiņas moduļus (ECC SIMM) un ECC algoritmus to kontroleros, kas ir 3-5% lēnāki nekā pārības kontroles algoritmi. Šo algoritmu galvenais trūkums ir liels korekcijas bitu skaits (16+6, 32+7, 64+8), kas ātri izlīdzinās pie liela vārda garuma. Rezultātā 32 MB SIMM ar pārības kontroli dod tikpat daudz kļūdu kā 1 GB ECC atmiņas sistēma.

Pieaugošā korpusu integrācija un darba ātrums palielina kļūdu rašanās biežumu. Tā 256 MB atmiņa ģenerē ap 1 programmas kļūdu mēnesī, no kurām līdz 5% ir daudzbitu kļūdas (0.6 kļūda gadā), kamēr aparatūra dod ap 0.01 kļūdu gadā (līdz 25% vai 0,0025 multibitu kļūdas). Rezultātā no 1000 serveriem ar 1 GB atmiņu katram gada laikā daudzbitu kļūdu dēļ atteiksies darboties 200, t.i., katrs piektais.

IBM 1998.g. sāka izlaist atmiņas moduļus DIMM, kuri detektēja un koriģēja daudzbitu kļūdas DRAM korpusos, kamēr ECC atmiņas kontroleris koriģēja viena bita kļūdas. Tas gan prasīja palielināt atmiņas korpusu izvadu skaitu (64 MB no 9 uz 11, 256 MB no 36 uz 44).

SIMM (Single In-line Memory Module) savieno uz īso abas spraudņa puses. Tā 72 kontaktu SIMM izmanto 144 kontaktu spraudni, palielinot tā drošību.

DIMM (Dual In-line Memory Module) abas spraudņa puses ir neatkarīgas un nodrošina 168 signālu izvadu, kas nepieciešams 64 bitu datu pārraidei Pentium procesoros.

Tālākā serveru attīstībā paplašina datu maģistrāli līdz 256 bitiem (64 +8-bitu ECC x4, kopā 288 kontakti) gadījumos, ja griešanās pie atmiņas ir secīga, kā tas ir nolasot programmas instrukcijas. Tāpat plaši pielieto vēl divas tehnoloģijas:

-EDO (Extended Data Out) izmanto sekojošās atmiņas lasīšanas komandās, pārslēdzot jauno adresi jau tad, kamēr notiek datu lasīšana iepriekšējai komandai, un novērš gaidīšanas stāvokli starp komandām;

-SDRAM (Synhronous DRAM) sinhronizē CPU un DRAM takts impulsus, atļaujot procesoram veikt citas operācijas bez gaidīšanas stāvokļa uz atmiņu, kamēr tai izdod adresi un veic datu lasīšanas vai rakstīšanas operāciju.

IBM PC Institute, September 8, 1998