

9

POMNILNIKI

BRANKO ŠTER

PO KNJIGI - DUŠAN KODEK: ARHITEKTURA IN
ORGANIZACIJA RAČUNALNIŠKIH SISTEMOV

Lastnosti pomnilnikov

1. Cena

- \$/GB
 - SRAM: 2000-5000 \$/GB
 - DRAM: 20-80 \$/GB
 - Bliskovni: nekaj \$/GB
 - Magnetni disk: 0,2-2 \$/GB
- poleg pomnilniških celic je treba v ceno vključiti še vso potrebno elektroniko in/ali mehaniko

2. Hitrost dostopa

- hitrost branja in pisanja
- **čas dostopa** (access time, t_a)
 - čas od pridobitve naslova do pojavitve podatkov
 - je definiran pri branju
 - pri pisanju je podoben
- pri nekaterih pomnilnikih (DRAM) mora po vsakem dostopu preteči nek čas, preden se lahko prične naslednji dostop
 - **čas cikla** $t_c = t_a + \text{čakanje}$
- **hitrost dostopa** (access rate) $b_a = 1/t_c$
- Gledano s strani naprave, ki bere ali piše v GP, imamo še čas t_p za prenos preko podatkovnih poti
 - prenos naslova in kontrolne informacije do GP ter prenos podatka nazaj (pri branju)
 - t_p je v rangi nekaj ns/m
- GP zaradi velikosti ne more biti na čipu CPE

- DRAM:
 - pri dostopu do poljubnega naslova: čas dostopa $t_a \sim 50$ ns, čas cikla $t_c \sim 60$ ns
 - pri dostopu do zaporednih naslovov hitreje
- SRAM:
 - čas dostopa t_a od 0,5 do 2,5 ns
- Hitrost dostopa pri magnetnem disku je približno 100.000 krat nižja kot pri polprevodniških pomnilnikih
 - v rangi več ms
 - nekje vmes so elektronski diski (EEPROM, Flash)
 - npr. USB ključki
- Razlog za uporabo pomnilniške hierarhije so velike razlike med pomnilniki v hitrosti in ceni
 - kljub zapletenosti, ki jo pomnilniška hierarhija vnaša, so prihranki v hitrosti tako veliki, da se jim ni mogoče odpovedati

3. Način dostopa

3.1 Naključni dostop (random access)

- čas dostopa t_a je konstanten in znan vnaprej ter neodvisen od prejšnjih naslovov
- RAM (random access memory)
 - DRAM (dinamični RAM) – GP
 - SRAM (statični RAM) - predpomnilnik
- načini dostopa do zaporednih bitov pri DRAM so hitrejši (vendar jih ne štejemo pod kategorijo zaporednega dostopa)
 - način strani (page mode, PM)
 - podamo NV, nato pa različne NS
 - potrebno je le, da so biti v isti vrstici (tudi, če niso zaporedni)
- rafalni ali eksplozijski način (burst mode)
 - zelo hiter, danes zelo pogosto uporabljan
 - dostop do zaporednih bitov s pomočjo majhnega internega števca, ki se prišteva k NS

3.2 Zaporedni dostop (serial access)

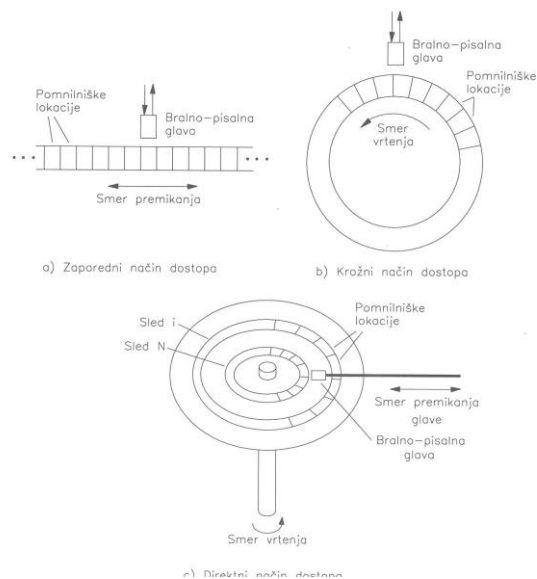
- čas dostopa je odvisen od prejšnjega naslova
 - če smo bili na naslovu A, je takoj dostopen le naslov A+1
- npr. magnetni trak

3.3 Krožni dostop (rotational access)

- posebna vrsta zaporednega dostopa
 - kot npr. magnetni trak, ki bi bil zlepljen v zanko
- npr. magnetni disk s fiksnimi glavami
- povprečen čas dostopa t_a je enak $\frac{1}{2}$ periode vrtenja

3.4 Kombinacija zaporednega in krožnega dostopa

- magnetni in optični diski s premičnimi glavami
- bralno-pisalna glava se najprej premakne na ustrezno sled (zaporedni dostop), nato pa imamo krožni dostop
- hitrejši od zaporednega ali krožnega dostopa



➤ Asociativni pomnilniki

- Pomnilniki z dostopom **preko (dela) vsebine** oz. *vsebinsko naslovljivi* (CAM, Content Addressable Memory) (ostali pomnilniki dostopajo **preko naslova**)
 - podamo del besede
 - primerja se z vsemi vpisanimi besedami (z ustreznimi biti)
 - primerjava je paralelna, zato zelo hitra
 - velika poraba logike (komparatorji), zato so AP majhni (< 1K)

4. Spremenljivost

- **Bralni pomnilniki** (ROM – Read Only Memory)
 - lahko ga beremo, vpis ni možen (vsaj za uporabnika ne)
 - luknjane kartice, tisk na papirju, CD-ROM, polprevodniški ROM
 - vsebina je obstojna (tj. tudi brez vira energije oz. napajanja)

- **Programirljivi bralni pomnilniki** (Programmable ROM - PROM)
 - lahko jih programiramo (vpišemo vsebino), sicer ne posebno hitro
 - PROM oz. OTP (One Time Programmable): na principu varovalk
 - EPROM (Erasable PROM): možen večkratni vpis in brisanje
 - programiranje z visoko napetostjo (rabimo programator), brisanje z UV-svetlobo (rabimo brisalnik) – čip ima na vrhu okence
 - EEPROM (Electrically Erasable PROM)
 - programiranje in brisanje z normalno napetostjo
 - Flash: podoben EEPROMu

- v računalnikih so bralni pomnilniki uporabljeni za shranjevanje **zagonskih programov**, ki se vključijo ob vklopu računalnika
 - majhen del GP je torej tipa ROM

- **Bralno-pisalni pomnilniki** (Random Access Memory)
 - z enako lahkoto beremo in pišemo
 - kratka je zavajajoča: to ni pomnilnik z naključnim dostopom!

5. Obstočnost

Obstaja več razlogov za izgubo informacije:

- **Destruktivno branje**
 - pri DRAM je informacija shranjena kot naboj na (zelo) majhnih kondenzatorjih
 - pri branju se kondenzatorji v vrstici praznijo, zato jih je treba ponovno nabiti
- **Dinamično shranjevanje**
 - tudi sicer se kondenzatorji s časom praznijo (dielektrik oz. izolator ni idealen) in jih je potrebno **osveževati** (refresh) večkrat na sekundo
 - odtod ime **dinamični RAM**
 - statični RAM ne potrebuje osveževanja
 - vrstica se prebere in zapiše nazaj
- **Izpad napajanja**
 - **Obstojni** pomnilniki (nonvolatile) ohranijo vsebino tudi, ko pride do izpada napajanja (ROM, magnetni disk, optični disk, ...)
 - RAM so neobstojni (volatile)

6. Zanesljivost

- Pomnilniki brez gibljivih delov (solid state), tj. polprevodniški, so bolj zanesljivi kot magnetni diski, pri katerih je potrebno mehanično gibanje
- Tudi pri polprevodniških pa so možne napake
 - kondenzator pomnilne celice pri DRAM je tako majhen, da mu lahko stanje spremenijo že kozmični žarki
 - to je **mehka napaka**, ker se celica ne poškoduje in dela naprej
 - zaradi mehkih napak se uporabljajo **kode za detekcijo in korekcijo napak** (dodatni biti)
 - **Trda napaka** (ki je redkejša) pa povzroči trajno okvaro celice

Zaščita glavnega pomnilnika

- Operacijski sistem (OS) je program (običajno več programov), ki teče na računalniku in upravlja s programskimi in strojnimi viri, npr.
 - omogoča (lažji) dostop do V/I naprav
 - upravljanje s pomnilnikom
 - večopravilni OS omogoča, da hkrati teče več procesov, itd.
- S pojavom prvih OS se pojavi potreba po mehanizmu, ki omogoča zaščititi en program pred posegi drugega programa
- Del OS mora biti stalno v GP
- Če programer zaradi napake v svojem programu spremeni vsebino pomnilniških lokacij, kjer je OS, lahko pride tudi do **razpada sistema** (crash)
 - v tem primeru je treba ponovno prenesti programe OS s pomožnega v glavni pomnilnik (s ponovnim zagonom računalnika)

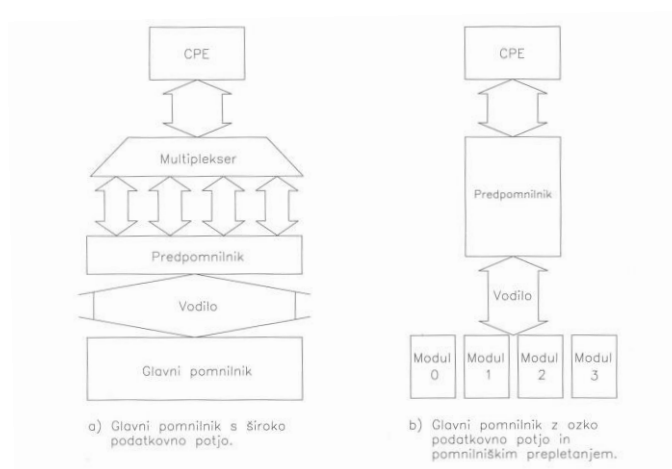
- Problem se je še povečal s pojavom večuporabniških (multiuser) in večopravilnih (multitasking) OS
 - istočasno se izvaja le en program (če imamo eno CPE), vendar si programi delijo isti pomnilniški prostor
 - treba je poskrbeti, da en program ne posega v prostor drugega (namenoma ali nehote, vseeno)
 - predvsem pisanje (spreminjanje), pa tudi branje, če gre za tajne informacije
- Nekateri programi OS so v bralnem pomnilniku in so s tem zaščiteni proti pisanju
 - ostali del OS se prenese z diska v GP
 - če bi bil ves OS v ROMu, bi bilo treba pri novejši verziji spremeniti čipe (oz. vsaj firmware)
 - nerodno, poleg tega to ne ščiti uporabnikov

- Najpreprostejši zaščitni mehanizem je par registrov, ki vsebuje spodnjo in zgornjo mejo naslova, ki pripada programu
 - vsak pomnilniški naslov A se pred dostopom do pomnilnika preveri
 - naslov je veljaven, če velja

$$\text{spodnja meja} \leq A \leq \text{zgornja meja}$$
 - slabosti:
 - programi morajo zasedati zvezen prostor v pomnilniku
 - vse besede so zaščitene na enak način
 - raje bi imeli "samo branje", "branje ali pisanje", ...
- Boljše rešitve uporabljajo **bloke** ali **strani** (pages) velikosti 1024, 2048 ali 4096 besed, ki so zaščiteni vsak zase
 - vsak program zaseda določeno število strani
 - vsaka stran ima svoj **zaščitni ključ** (protection key), ki je neko zaporedje bitov
 - shranjeno v tabeli strani za navidezni pomnilnik

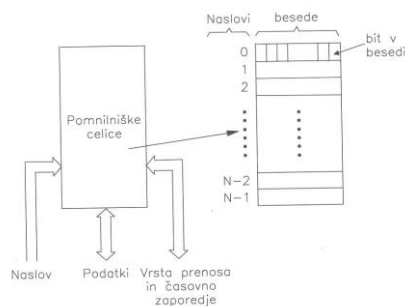
- Cilj zaščite je običajnim uporabnikom preprečiti dostop do *privilegiranega načina* delovanja (privileged mode)
 - v določenih primerih uporabnik potrebuje storitve, ki so dovoljene samo v privilegiranem načinu
 - mnogi OS imajo za ta namen *sistemske klice* (system calls)
 - preprosti sistemi (npr. vgrajeni - embedded) imajo običajno samo en način (privilegiran)
 - gonilnike naprav (device drivers) lahko programira običajen uporabnik
- Poleg strojne zaščite je možna tudi programska

- Kljub PP je potrebno eventualno še vedno dostopati do GP
 - Hitrost pomnilnikov DRAM se povečuje bistveno počasneje od hitrosti CPE
- Ena od možnosti pohitritve je povečanje števila naenkrat prenešenih bitov. 2 načina:
 1. **Širše podatkovne poti do GP.**
 - dostop do sestavljenih pomnilniških besed
 2. **Pomnilniško prepletanje (memory interleaving).**
 - GP je razdeljen na m samostojnih delov M_0, M_1, \dots, M_{m-1}
 - to so **moduli** oz. **banke**
 - **m -kratno prepletanje** (m -way interleaving)
 - širina podatkovnih poti se ne poveča (vsaj v osnovni izvedbi)
 - vsak modul je samostojen pomnilnik, ki deluje neodvisno od ostalih
 - z dekodiranjem določenih bitov naslova se izbere enega od modulov
 - možnih je m istočasnih dostopov
 - po začetni zakasnitvi je možen po en prenos na urino periodo



Organizacija glavnega pomnilnika

- Pove, kako so biti sestavljeni v pomnilniške besede in kakšen je dostop do njih
- GP je videti kot enodimenzionalno zaporedje pomnilniških besed; vsaka ima svoj enoličen naslov



POMNILNIKI

19

- Osnovna parametra pomnilnika sta:

1. Pomnilniška beseda

- to je najmanjše število bitov s svojim naslovom
 - **dolžina besede** (običajno 1B oz. 8 bitov)
- običajno je možen dostop do več besed

2. Pomnilniški naslov

- binarno število
- **dolžina naslova** določa velikost pomnilniškega prostora
 - pri m -bitnem naslovu $a_{m-1} \dots a_1 a_0$ je lahko največ 2^m besed

POMNILNIKI

20

- 3 vrste signalov
 - naslovni
 - podatkovni
 - kontrolni
- Dolžina registrov CPE je enaka mnogokratniku dolžine pomnilniške besede
- Pomnilniški prostor vsako leto naraste s faktorjem med 1,5 in 2 (torej eksponentno)
- Velikost naslova določa širino vsega, kar lahko vsebuje naslov:
 - ukazov
 - registrov
 - aritmetike za računanje naslova
- Zato je povečati dolžino naslova izjemno težko
 - premajhna dolžina naslova je največja možna napaka pri razvoju novega računalnika, ker jo je kasneje skoraj nemogoče popraviti

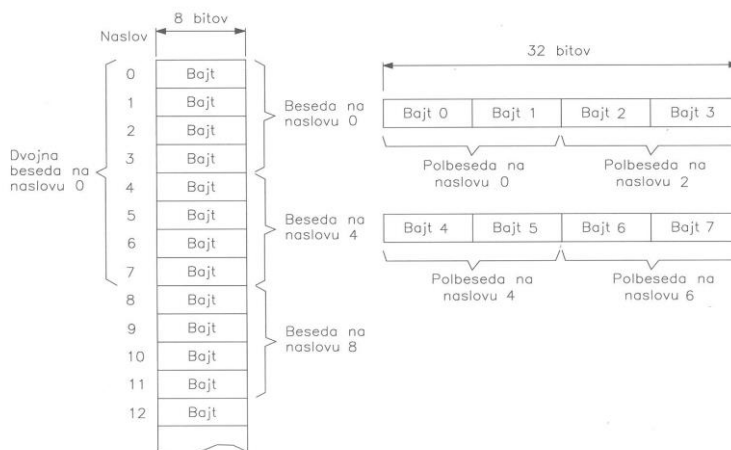
- GP, ki omogoča dostop do **sestavljenih pomnilniških besed**, je možno narediti na 2 načina:
 1. več paralelnih pomnilnikov
 - spodnji biti naslova določajo, za katerega gre
 - npr. pri dostopu do 8 besed naenkrat je 8 pomnilnikov, spodnji 3 biti določajo pomnilnik
 2. vedno se naredi dostop do vseh (npr. 8) besed
- Kjer je možen dostop do sestavljenih besed, je dobro, če je podatkovno vodilo temu ustrezno široko, sicer je potrebnih več prenosov
 - tudi, če je več prenosov, programer tega ne vidi

➤ Primer:

- dolžina pomnilniške besede 1B
- dva sosedna bajta tvorita polbesedo (halfword, 16 bitov)
- štirje sosedni bajti tvorijo besedo (word, 32 bitov)
- osem sosednih bajtov tvorijo dvojno besedo (doubleword, 64 bitov)
- npr. pravilo debelega konca
 - naslov vsake od sestavljenih besed je enak naslovu bajta z največjo težo
- pri večini računalnikov je potrebna **poravnanost**
 - sestavljene besede morajo biti na naslovih, ki so večkratniki 2, 4, oz. 8
 - sicer je potrebnih več dostopov!
 - npr. če je polbeseda na 24-bitnem naslovu 10FFFF
 - prvi bajt ima naslov 10FFFF, drugi pa $10FFFF+1 = 110000$
 - razlikujeta se v 17 bitih!

POMNILNIKI

23



POMNILNIKI

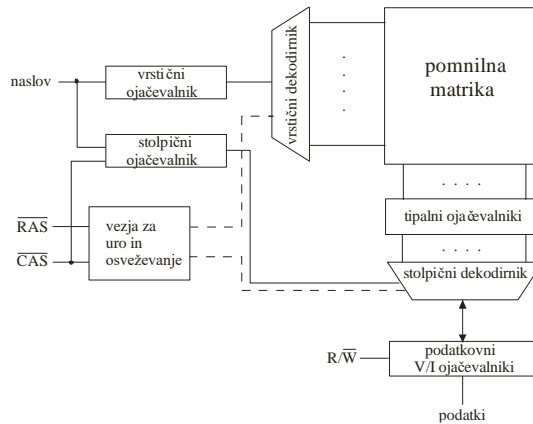
24

Tehnologija polprevodniških pomnilnikov

➤ DRAM (Dinamični RAM)

■ zgradba

- izhodi vrstičnega dekodirnika so *besedne linije*
- na stolpični dekodirnik so vezane *bitne linije*
- naslov je razdeljen na 2 dela:
 - vrstični
 - stolpični

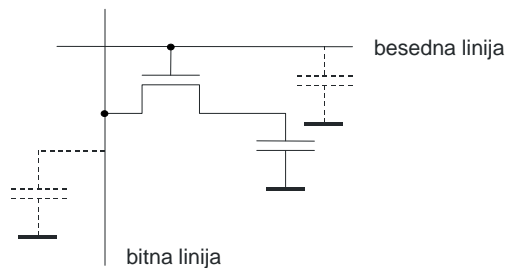


POMNILNIKI

25

■ Pomnilna celica DRAM

- kondenzator
 - nabit: eno logično stanje (npr. "1"); prazen: drugo logično stanje (npr. "0")
 - $C_s \sim 20\text{fF}$ (s ... storage)
- stikalni transistor (MOS)



POMNILNIKI

26

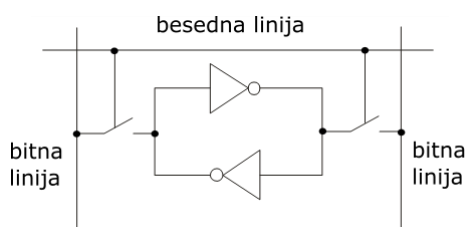
- DRAM vsebuje bitno ravnino oz. matriko ALI
 - v njej so besedne in bitne linije, na presečiščih pa so pomnilne celice
 - razlog za 2D organizacijo je velikost dekodirnika in število ter dolžina linij
 - npr. 1Mb pri 1D: dekodirnik 20/1M, 1M besednih linij, zelo dolga bitna linija (z 1M celicami! – ogromna kapacitivnost)
 - 2D: 2 dekodirnika 10/1024, 1024 besednih linij, 1024 bitnih linij, 1024 celic na bitni liniji
- Primer: DRAM 32Mb x 1
 - 25-bitni naslov: 15 (vrstični del) + 10 (stolpični del)
 - torej 2^{15} besednih linij, 2^{10} bitnih linij
 - običajno je besednih linij več kot bitnih
 - zato so lahko krajše (hitrejši dostop zaradi manjše kapacitivnosti)
- Primer: DRAM 32Mb x 8
 - podobno, vendar 8 bitnih ravnin

- Pomnilniški dostop:
 - bitne linije *prednabijemo* (precharge) na polovično napetost
 - se ne izpraznijo prav hitro zaradi relativno velike kapacitivnosti (C_b), ki je posledica parazitnih kapacitivnosti velikega števila celic na liniji
 - podamo naslov vrstice (NV)
 - aktiviramo signal RAS' (row address strobe), ki je aktivno nizek
 - vsebina vrstice (naboj na kondenzatorjih) gre preko bitnih linij na *tipalne ojačevalnike* (sense amplifier, SA)
 - v resnici ne čakamo, da se kondenzator popolnoma izprazni, ampak le delno (zaradi hitrosti)
 - ker je $C_b > C_s$, se napetost bitne linije le malo spremeni - običajno nekaj sto mV
 - SA zazna to razliko in vrne logično vrednost (0 ali 1)
 - vrednosti se shranijo v *register vrstice* (oz. *buffer*)
 - podamo naslov stolpca (NS)
 - aktiviramo signal CAS' (column address strobe), ki je tudi aktivno nizek
 - pri bralnem dostopu (WE' (write enable) = 1) dobimo na izhodu iskani bit
 - pri pisalnem dostopu ($WE' = 0$) se bit vpiše v register vrstice
 - register vrstice se vpiše nazaj v celice

- DRAMi uporabljajo *naslovno multipleksiranje*
 - naslov vrstice in naslov stolpca sta na istih pinih
 - s tem se zmanjša število priključkov (pinov) za bite naslova
 - naslovi so pri DRAMih seveda dolgi (npr. 30 bitov pri 1Gb)
 - priključki so glavni dejavnik pri ceni čipa
 - ne izgubimo kaj dosti na času, saj potrebujemo NV prej kot NS
- Današnji DRAMi so sinhronski (SDRAM)
 - sinhronizirani so s sistemsko uro
 - imajo 3-stopenjski cevovod
 - register na vходу
 - DRAM (asinhronski)
 - register na izhodu
 - najpogostejši so DDR (1,2,3)
 - double data rate

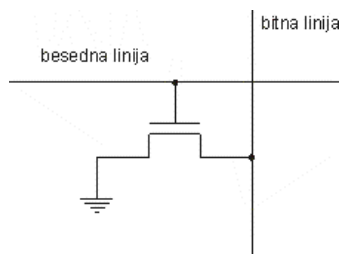
➤ SRAM (Statični RAM)

- zgradba je v osnovi podobna kot pri DRAM
- pomnilna celica je *zapah*
 - podoben RS-zapahu, le način vpisovanja je drugačen
 - informacija se ne izgublja (vkolikor ne izključimo napajanja)
 - zato se celica imenuje statična



➤ Pomnilna celica pri **ROM**:

- bitna linija je vnaprej nabita (prednabita)
- signal na besedni liniji povzroči, da transistor prične prevajati
- tok teče iz bitne linije proti masi, zato se zmanjša naboj na bitni liniji
- posledično upade napetost bitne linije, kar zazna posebna vezje (v izhodni stopnji), ki to tolmači kot "0"
- če transistorja ni, napetost ne upade ("1")



POMNILNIKI

31

➤ **Bliskovni pomnilnik (Flash memory)** je vrsta programirljivega pomnilnika ROM (programmable ROM), za katerega lahko uporabnik določi oz. vpiše vsebino, ta pa je potem obstojna (z izklopom napajanja se ne izgubi)

- V Flash celici je izpeljanka običajnega MOS tranzistorja, ki ima znotraj oksidne plasti dodatno (t.i. plavajočo) plast – kadar je ta nabita z elektroni, tranzistor efektivno ne prevaja (kakor da ga v celici ne bi bilo)

POMNILNIKI

32