


# Memorija



# Pojmovi


- **Memorija** - specijalizovan hardver namenjen čuvanju informacija.
  - Sastoji se od **memorijskog medijuma i upravljačkog sistema**.
  - Dve osnovne operacije, čitanje i pisanje.
- 

# Pojmovi


- Podaci (u glavnoj memoriji) organizovani u **reči** (kao osnovne jedinice)\*.
- Van glavne memorije postoje različiti načini organizacije(npr. sektori kod hard diska)

\*Dužina reči memorije(najčešće) odgovara dužini instrukcije, odnosno dužini prikazivanja celog broja, odnosno dužini procesorske reči.


# Načini pristupa memoriji

- **Adresni pristup** - podaci se traže po njihovoj lokaciji
    1. Direktni
    2. Poludirektni
    3. Sekvencijalni
  - **Asocijativni pristup** - podaci se traže po sadržaju
- 


# Direktni pristup

- Karakteristično za RAM
  - Svaka memorijska lokacija ima jedinstveni identifikator
  - Vreme pristupa konstantno
  - Takođe se naziva i slučajni pristup
- 


# Poludirektni pristup

- Grupa podataka nalazi se na istoj adresi
  - Karakteristično kod spoljne memorije (recimo, hard diskova, gde su podaci organizovani po sektorima)
  - Vreme pristupa zavisi od pozicije mehanizma za čitanje
- 

# Sekvencijalni pristup

- Podaci organizovani u zapise koji slede jedan za drugim
  - Da bi se našao neki podatak, mora se preći kroz sve predhodne
  - Primer, magnetne trake (kod starijih računara)
  - Vreme pristupa se razlikuje za svaki podatak
- 

# Asocijativni pristup

- Reč se iz memorije izvlači na osnovu sadržaja, a ne na osnovu lokacije
  - Vreme pristupa konstantno
  - Čest slučaj kod keš memorija
- 



# Vreme pristupa

- Vreme koje protekne od dovođenja signala (za čitanje ili upis) do trenutka kada je podatak na magistrali podataka, ili do trenutka kada je podatak upisan, respektivno.
- Vreme između dva uzastopna pristupa memoriji jeste **memorijski ciklus\***.

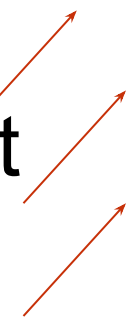
\*u opštem slučaju duži od vremena pristupa zbog tehničkih razloga




# Hijerarhija memorija

- Bitne karakteristike: brzina memorije, kapacitet, cena
- Idealni računar imao bi jeftinu, brzu memoriju velikog kapaciteta


brzina  
kapacitet  
kapacitet



cena  
cena  
brzina




# Hijerarhija memorija

- Za efikasan rad suštinski bitna brzina memorije.
  - Prilaskom kroz hijerarhiju:
    - smanjuje se cena po bitu
    - povećava se kapacitet
    - povećava se vreme pristupa
    - smanjuje se učestalost pristupa memoriji od strane procesora.
- 


# Hijerarhija memorija



# Registri

- Lokalna memorija procesora
  - Čuva podatke dok se obrađuju
  - Desetak registara po procesoru
  - Vreme pristupa  $\sim 1\text{ns}$
- 


# Keš

- Cache - skriven (fr.)
  - Brza memorija (brzine procesora ili uporediva)
  - Sastoji se od **keš kontrolera i memorijskih (SRAM) čipova.**
  - L1 unutar čipa (nekoliko desetina kilobajta)
  - L2 van čipa (nekoliko megabajta), vreme pristupa ~10ns
- 

# Operativna (glavna) memorija


- Vreme pristupa nekoliko desetina ns
- Procesor joj može pristupiti direktno

# Hard disk/SSD


- Vreme pristupa nekoliko desetina ms.
  - Pristup poludirektan (podaci organizovani u sektore).
  - Kod HDD-a podaci se čuvaju magnetskim putem
  - Kod SSD-a podaci se čuvaju elektronskim putem, na NAND čipovima (flash memorija)
- 



# Memorija sa izmeljivim diskovima

- CD
  - DVD
  - Floppy
  - Vreme pristupa nekoliko stotina ms (kod optičkih medijuma) do sekunde (kod magnetskih traka)
- 

# Virtuelizacija

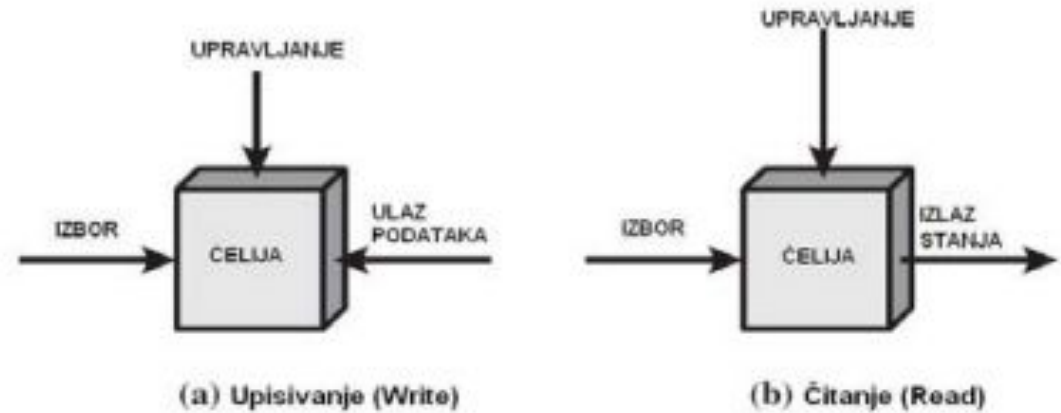
- Proces prividnog povećanja kapaciteta memorije
  - Adresni prostor (skup svih mogućih adresa) objedinjuje primarnu i sekundarnu memoriju (RAM i hard disk)
  - Svaka virtuelna adresa identifikuje podatak koji može biti u RAM-u ili na hard disku
  - Ukoliko je podatak na hard disku, on se mora prebaciti u RAM kako bi se mogao obraditi (**straničenje**)
- 

# Neki parametri memorija


Memorijski medijum	Tipično srednje vreme pristupa	Propusna moć	Kapacitet medijuma	Obim bloka prebačenog sa višeg na niži nivo	Ko upravlja prenosom podataka
Registri CPU-a	200ps-1ns	0.5-60GB/s	256B-1kB	Reč obima 2B ili 4B	Upravljačka jedinica CPU-a
L1 keš memorija	5-10ns	0.8-1GB/s	16-64kB	Linija 4-32B	Primarni keš kontroler
L2 keš memorija	15-40ns	0.1-0.3GB/s	128kB-1GB	Linije 4-128B	Sekundarni keš kontroler
Glavna memorija	50-100ns	20-80MB/s	256MB-1GB	Stranice 4kB	MMU (upravljačka jedinica memorije)
Slotovi proširenja glavne memorije	75-500ns	800kB-30MB/s	1-10GB	Stranice 4kB	MMU
Disk keš	60-500ns	900kB-30MB/s	1-10MB	Blokovi 4kB	Kontroler uređaja
Hard disk	5-50ms	1200-6000kB/s	100-500GB	Fajlovi obima MB	Kontroler uređaja
Flopi disk	95ms	100-200kB/s	1.44MB	Fajlovi obima MB	Kontroler uređaja
CD-ROM	100-500ms	500-4000kB/s	600MB-20GB	Fajlovi obima MB	Kontroler uređaja
Trake (cartridge)	0.5s pa naviše	2000kB/s	1-10TB	Fajlovi obima MB	Kontroler uređaja

# Poluprovodničke memorije


- Primarna memorija izrađena je u **poluprovodičkoj tehnologiji**. Osnovni element poluprovodničke memorije je **memorijska ćelija**.



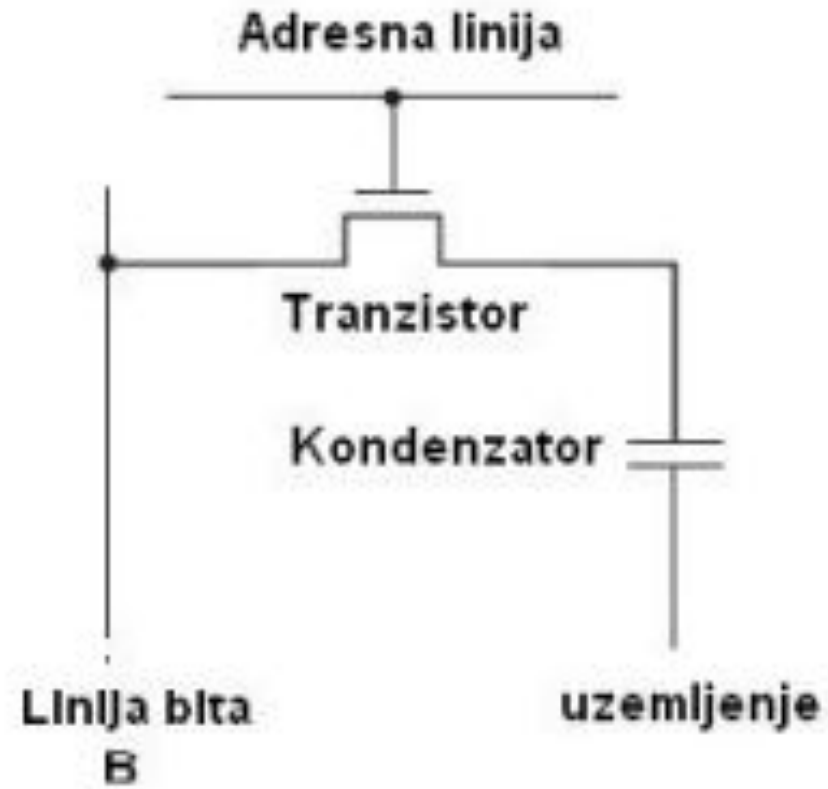
# Poluprovodničke memorije

- Čelija najčešće ima tri terminala koji mogu da nose električni signal.
    1. upravljački terminal
    2. terminal izbora
    3. ulazno/izlazni terminal
- 


# Poluprovodničke memorije

- **RAM** - random access memory, za čitanje i pisanje
  - **ROM** - read only memory, za trajno čuvanje podataka, pri čemu se može samo čitati
  - RAM mogu biti **statičke** i **dinamičke**
  - Memorijski elementi izrađeni su od **tranzistora**.
    1. bipolarni tranzistori (nnp, pnp)
    2. MOS tranzistori (NMOS, PMOS)
- 

# DRAM




# DRAM

- Jednostavna tehnologija
  - Može se gusto pakovati
  - Mora da se osvežava (potreban poseban hardver za osvežavanje)
  - Problem - potreba za memorijom dok se ona osvežava
- 



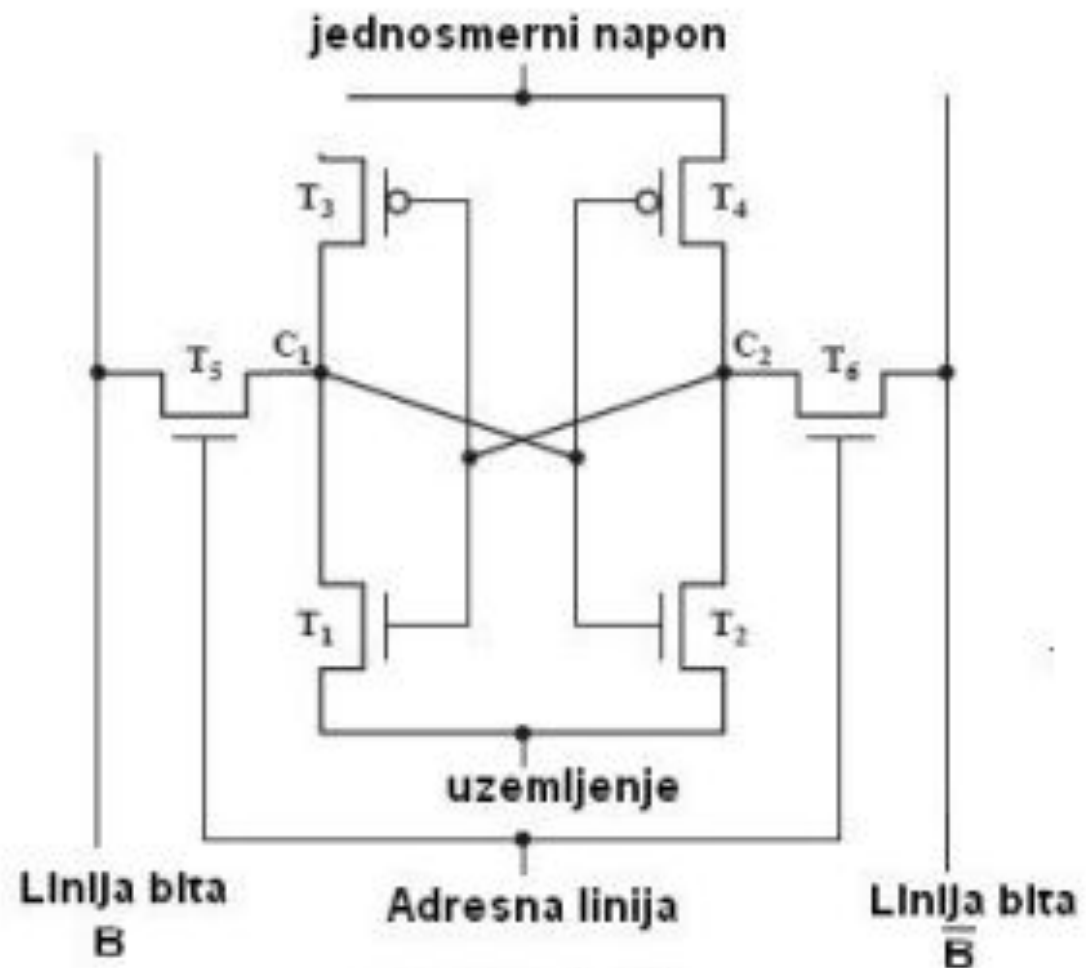
# DRAM

- Generalno, DRAM je asinhrona memorija, tj. radi van procesorskog takta.
  - **SDRAM** ili sinhroni DRAM radi na taktu procesora, što znači da je spreman za prenos podataka kada CPU to i očekuje.
  - Prenos kod SDRAM traje nekoliko taktova, tako da je CPU slobodan da radi neki drugi posao
- 


# DRAM

- **DDR SDRAM** je dvostruko brža od SDRAM memorije, zato što podatke može da šalje i na uzlaznom i na silaznom delu signala. Danas imamo verzije DDR3 i DDR4 ove memorije

# SRAM




# SRAM

- Čelije su bistabilne
  - Ne zahtevaju osvežavanje
  - Mnogo tranzistora => gustina pakovanja znatno manja od DRAM memorije
  - Skupe
- 


# Postojane (poluprovodničke) memorije

- Postojane memorije možemo podeliti u sledeće klase:
  - MASK ROM memorije kod kojih se sadržaj unosi još u procesu proizvodnje,
  - PROM memorije kod kojih se sadržaj unosi naknadno,
  - EPROM memorije kod kojih se naknadno uneti sadržaj može i izbrisati i
  - NVRAM memorije koje se veći deo vremena ponašaju kao ROM, ali čiji se podaci mogu promeniti na zahtev


# MASKROM

- Sadržaj memorije definiše se u samom procesu proizvodnje
  - U finalnom delu proizvodnje, tranzistori ove memorije se povezuju graviranjem metalne maske koja se naknadno nanosi
  - Ukoliko želimo logičku jedinicu, između adresne linije i linije podataka postavlja se dioda
- 

# PROM


- Programmable ROM - korisnik može sam trajno da zapiše podatak (jednom)
  - Redno sa diodom ugrađen topljivi osigurač
  - Programiranje podrazumeva da se na mestima logičke nule osigurač otopi
- 

# EPROM

- Erasable PROM
  - Programiranje se vrši preko posebnog kontrolera, brisanje u komorama sa ultraljubičastom svetlošću
  - EEPROM - electrically EPROM, brisanje se vrši električnim putem. Poseban tip EEPROM jeste flash memorija (dobila naziv po velikoj brzini upisa)
- 

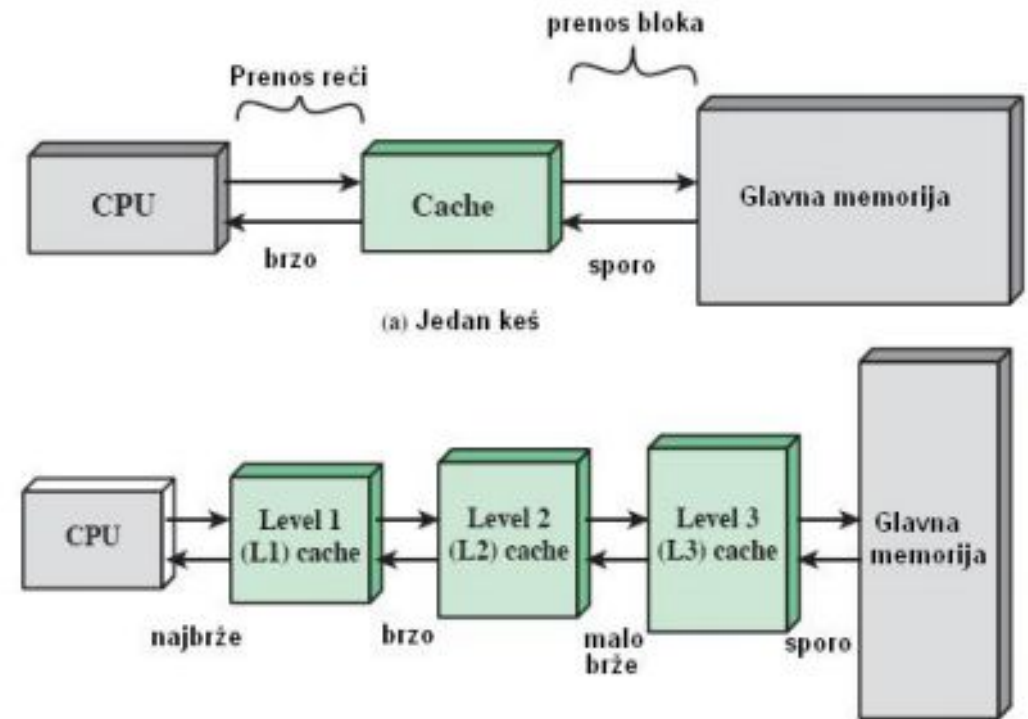


# NVRAM


- Non-volatile RAM, memorija sa nasumičnim pristupom i postojanim čuvanjem podataka
  - Izrađena najčešće u CMOS tehnologiji
- 

# Keš memorije


- Velikoj i relativno sporoj glavnoj memoriji pridružuje se manja, brža i skuplja keš memorija u koju se upisuje kopija delova informacija iz glavne memorije koji se najčešće koriste, pa se na taj način ubrzava pristup.



# Keš

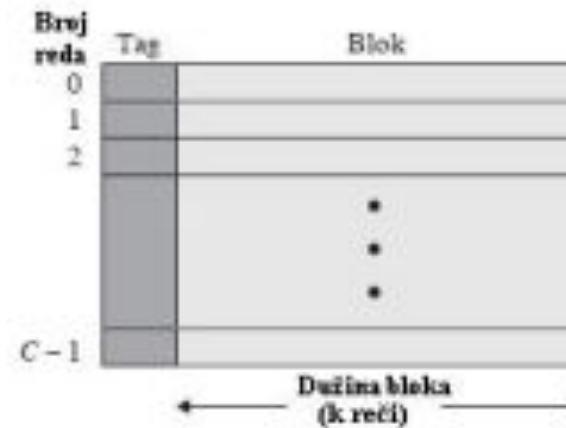
- Zasniva se na principu **lokalnosti** (prostorne i vremenske)
  - Vremenska lokalnost - instrukcije koje su skoro korišćene će verovatno biti ponovo korišćene
  - Prostorna lokalnost - instrukcije koje se nalaze blisko u memoriji će verovatno biti korišćene brzo jedna posle druge
- 

# Keš

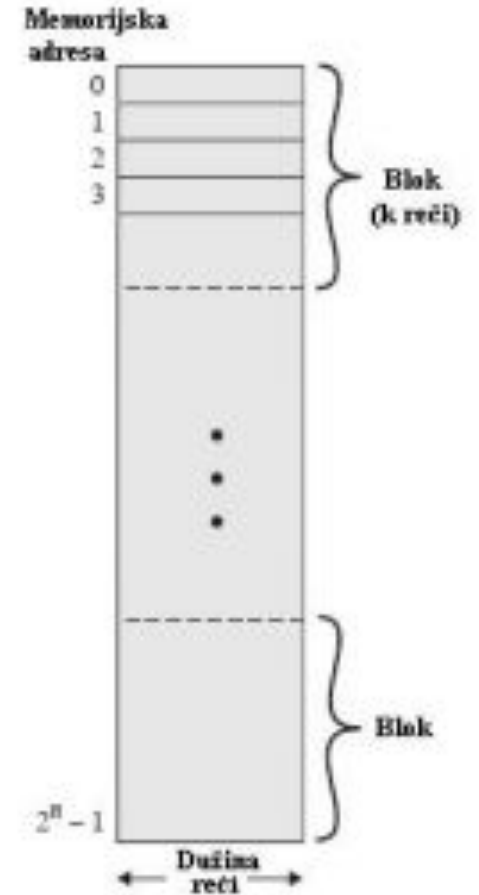
- Procesor šalje upit za neki podatak upravljačkom sistemu memorije. Keš kontroler će proveriti da li se taj podatak nalazi u okviru SRAM čipova (**cache hit**).
  - Možemo prvo tražiti u keš memoriji (look through cache), ili keš može nadgledati upit RAM memoriji i onda obezbediti podatak ako se nalazi kod njega (look aside cache)
  - Ukoliko podatak nije u kešu, to je **cache miss**.
- 

# Keš

- U slučaju promašaja, izmenjeni podaci iz keša vraćaju se u RAM, a u keš se ubacuje segmenat podataka iz RAM koji sadrži promašeni podatak (**keš linija/punjenje keš linije**)




(a) Keš memorija




(b) Glavna memorija

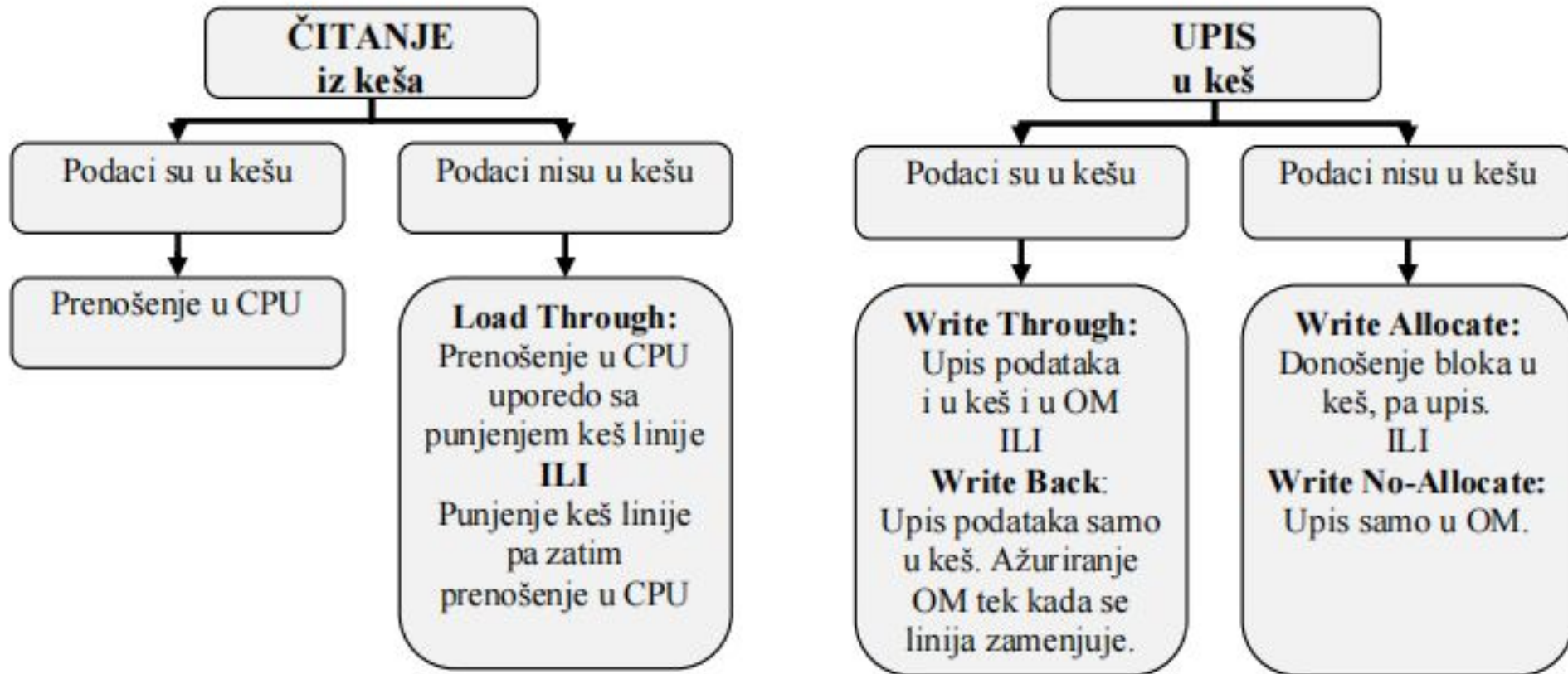
# Keš

- RAM se sastoji od reči, koje se organizuju u blokove od po  $k$  reči
  - Keš se sastoji od redova, u svakom redu može biti upisan jedan blok
  - Tag, identifikator koji blok u memorije se nalazi u nekom redu (nema dovoljno redova za sve blokove)
- 

# Keš


- Veza između redova u kešu i blokova u RAM data je **funkcijom mapiranja**
  - Ukoliko je keš pun, neku liniju treba vratiti nazad u RAM i doneti nove podatke; ovo je određeno **algoritmom zamene**
- 

# Keš





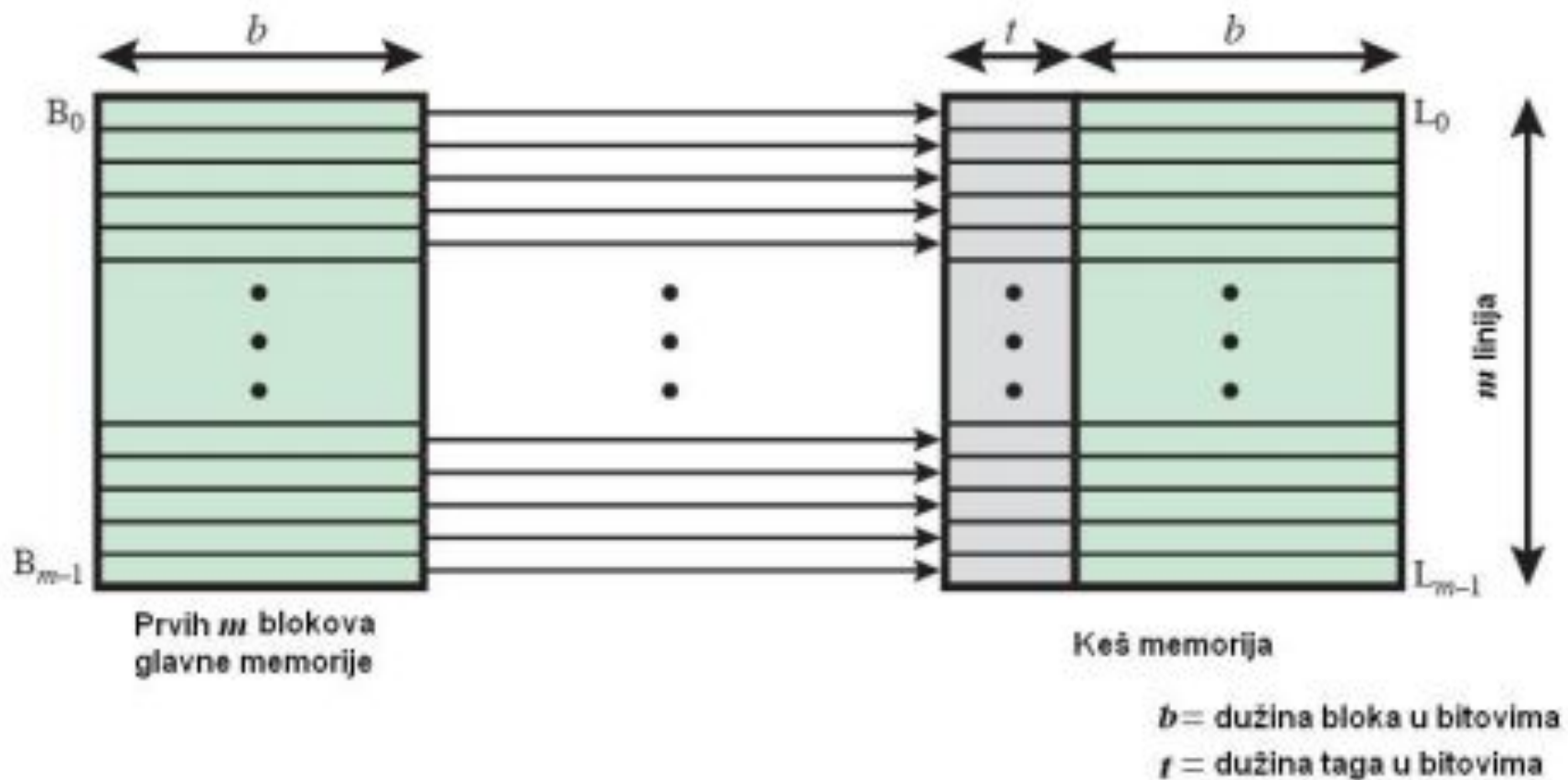
# Strategije za popunjavanje keša

- Odabir funkcije preslikavanja
  - Mogu da se koriste tri tehnike:
    1. direktno preslikavanje,
    2. asocijativno preslikavanje
    3. set asocijativno preslikavanje.
- 

# Strategije za popunjavanje keša

–Neka su pretpostavke sledeće: Dužina reči u memoriji je 1B. Memorija je veličine 128 bajtova, što znači da postoji  $2^7$  mogućih adresa. Samim tim, adresa je dužine 7 bitova. Neka je veličina svakog bloka u RAM memoriji 8B. To znači da postoji ukupno 16 blokova unutar RAM. Neka je ukupna veličina keša 32B. Svaki red(linija) u kešu odgovara jednom bloku, pa ukupno postoje 4 moguća reda.

# Direktno preslikavanje

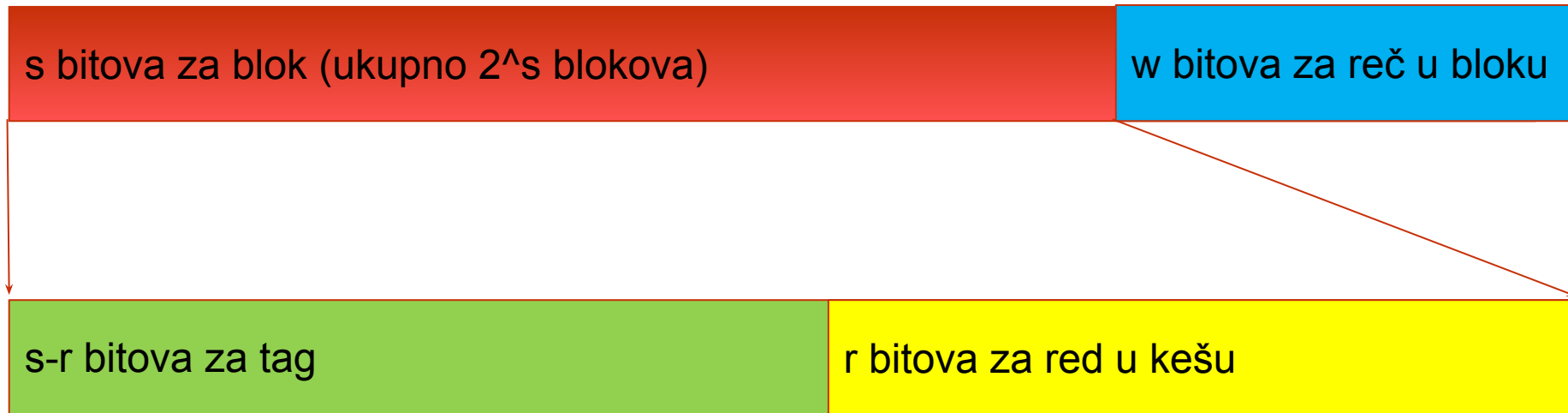


# Direktno preslikavanje


- Svaki blok u RAM se preslikava samo u jedan red keš memorije. Funkcija preslikavanja u ovom slučaju je
  - $i = j \bmod m$  gde je:
    - $i$  = broj reda keša
    - $j$  = broj bloka glavne mem.
    - $m$  = broj redova u kešu.

# Direktno preslikavanje

- Memorijska adresa se sastoji od tri dela




# Direktno preslikavanje

- Upotreba dela adrese kao broja reda obezbeđuje jedinstveno preslikavanje svakog bloka glavne memorije u keš. Kada se blok stvarno učitava u njemu dodeljen red, potrebno je da se obeleže podaci kako bi se razlikovali od drugih blokova koji bi mogli da se upišu u taj red. U tu svrhu služi najznačajnijih  $s - r$  bitova.
- 

# Direktno preslikavanje


Blokovi glavne memorije		Mogu da se upišu red keša
$0, m, 2m, \dots, 2^S - m$	$\rightarrow$	0
$1, m+1, \dots, 2^S - m+1$	$\rightarrow$	1
...		...
$m-1, 2m-1, 3m-1, \dots, 2^S - 1$	$\rightarrow$	$m-1$

# Asocijativno preslikavanje


- Prevazilazi nedostatak direktnog preslikavanja, svaki blok može da se učitati u bilo koji red keša
  - Koristi asocijativnu memoriju (lokacija podatka definisana njegovim sadržajem, ne adresom)
  - Ovakve memorije nazivaju se i **CAM** (content addressible memory)
- 




# Set Asocijativno preslikavanje

- Kompromis između direktnog i asocijativnog preslikavanja
  - Keš memorija deli se na skupove od po  $k$  elemenata
  - Preslikavanje na skupove direktno, preslikavanje u okviru samih skupova asocijativno
- 

# Politika upisivanja

- Upisivanje se uvek vrši u memoriju, bez obzira da li je došlo do pogotka keša (**write through**)
  - Upis u RAM loš aspekt, ali podaci su konzistentni između RAM i keš memorije (osim u slučaju multiprocesorskih sistema)
  - Alternativa je **write back** pristup, kada se podaci upisuju samo u keš. Upis u memoriju nije automatski. Podaci nekonzistentni, ali pristup brz.
- 

# Politika upisivanja

- Ako drugi procesori ili komponente sistema imaju pristup glavnoj memoriji (npr. DMA kontroler) u glavnoj memoriji može se promeniti podatak koji je u kešu neizmenjen. Stoga keš kontroler mora konstantno da nadgleda sve pristupe memoriji u cilju upisa, i markira odgovarajući sadržaj SRAM-a kao nekorektan (**cache invalidation**), ako se podatak u glavnoj memoriji menja.
- 

# Politika upisivanja

- Ako neki uređaj traži podatak iz RAM, a izmenjena vrednost se nalazi u kešu, onda se podatak mora prebaciti u RAM, pa tek onda do uređaja (**cache flush**).