

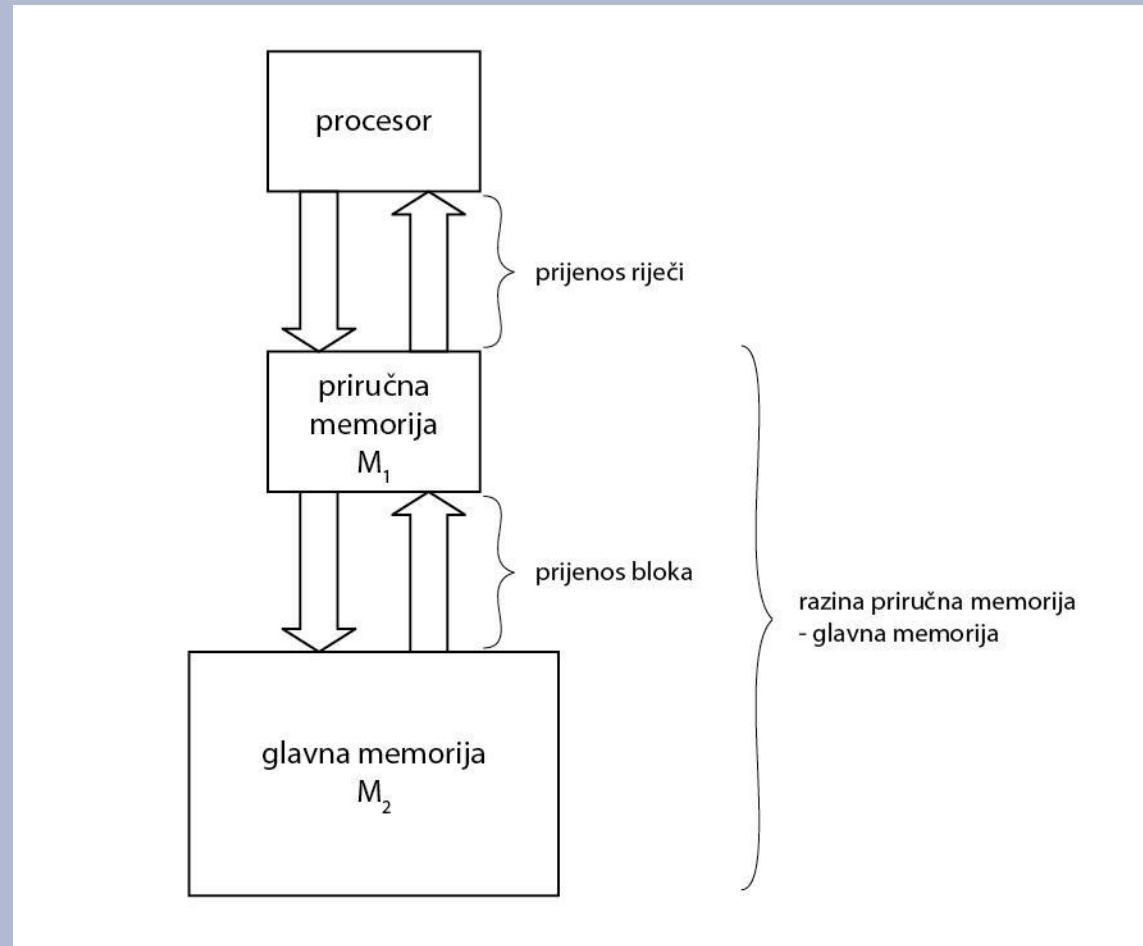
Obraditi temu: Ulazno-izlazni sustav, Građa računala,  
Arhitektura i organizacija računarskih sustava, poglavljje 13.,  
str. 393 – 428.

# Priručna (engl. cache) memorija

(Građa računala, Arhitektura i organizacija računarskih sustava,  
str. 311 – 336)

- između procesora i  $M_1$  prenose se podaci u obliku riječi
- između  $M_2$  i  $M_1$  prenosi se blok podataka (priručni blok, linija)

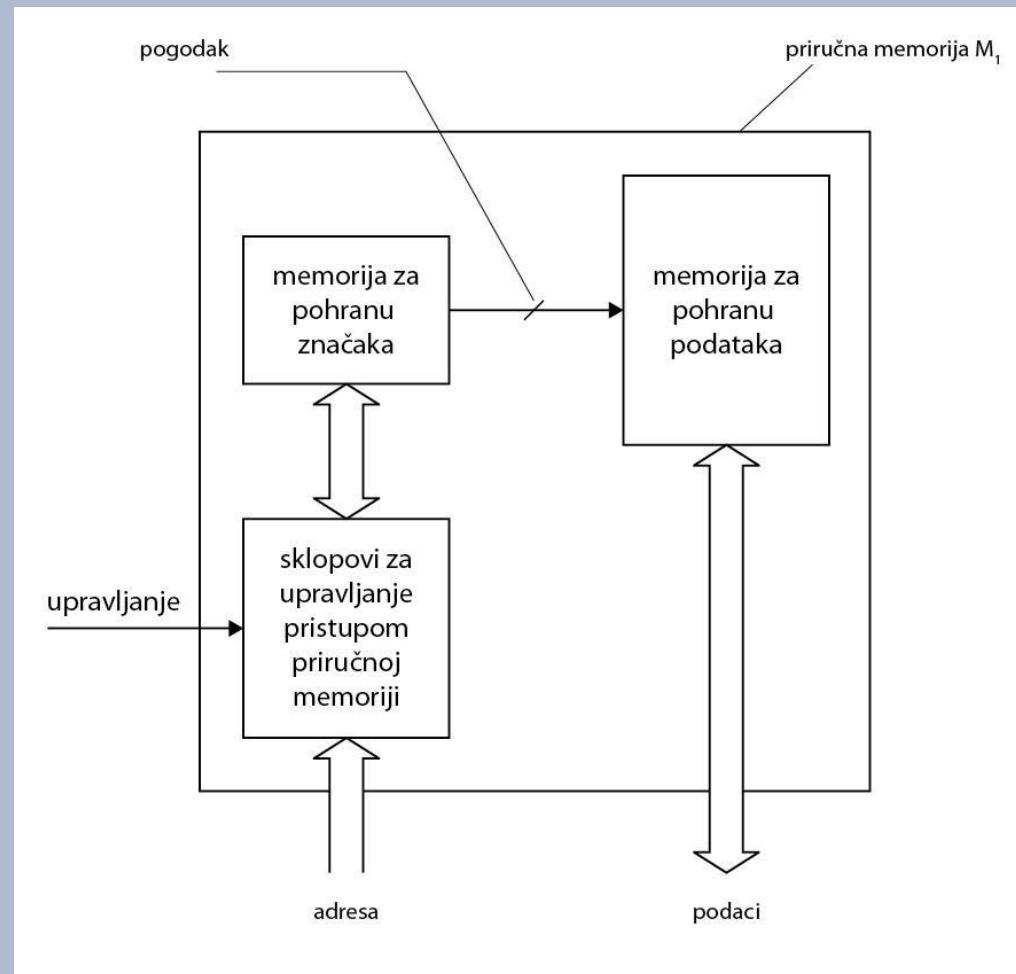
Priručna memorija  
sadržava kopiju sadržaja  
dijela glavne memorije –  
tekuće aktivne segmente  
programa i podataka



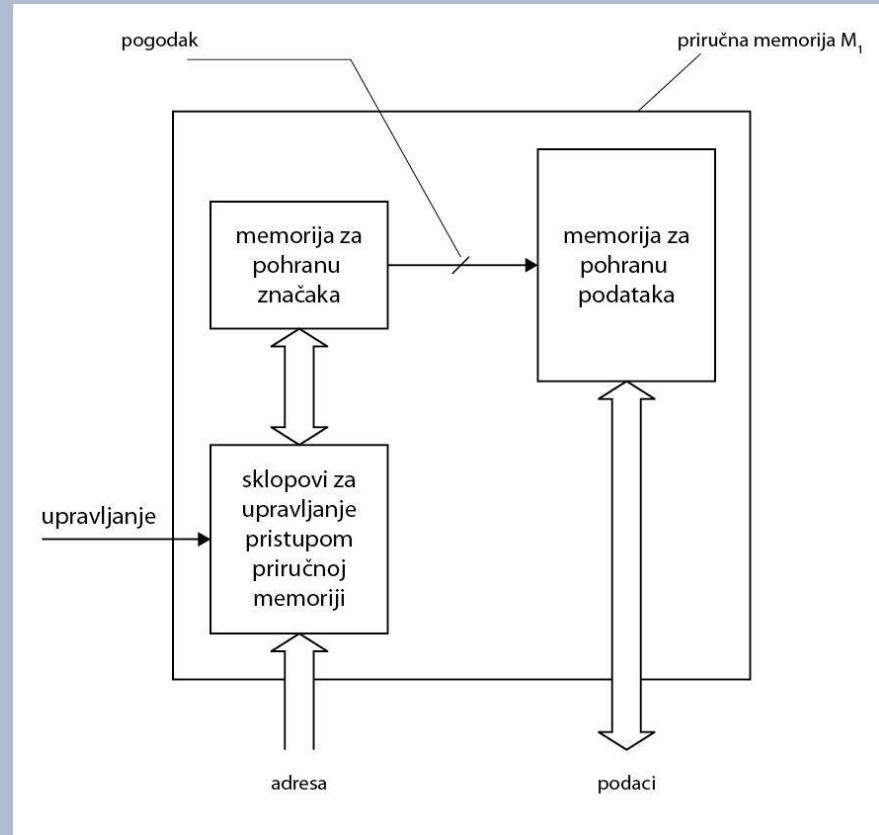
# Interna organizacija priručne memorije

Opaska: ovdje se podaci podrazumijevaju u širem smislu – instrukcije i podaci

Kopija sadržaja dijela glavne memorije pohranjuje se u **memoriji za pohranu podataka** koja je organizirana u male blokove – **priručni blokovi ili linije**



- Svaki je priručni blok označen svojom bločnom adresom koja se naziva **značka** tako da priručna memorija “zna” kojem dijelu glavne memorije odgovara dotični blok
- Značke su pohranjene u memoriji za pohranu značaka
- Glavna memorija je **također** podijeljena na blokove koji su jednake veličine kao i priručni blokovi



Sljedeća organizacija:

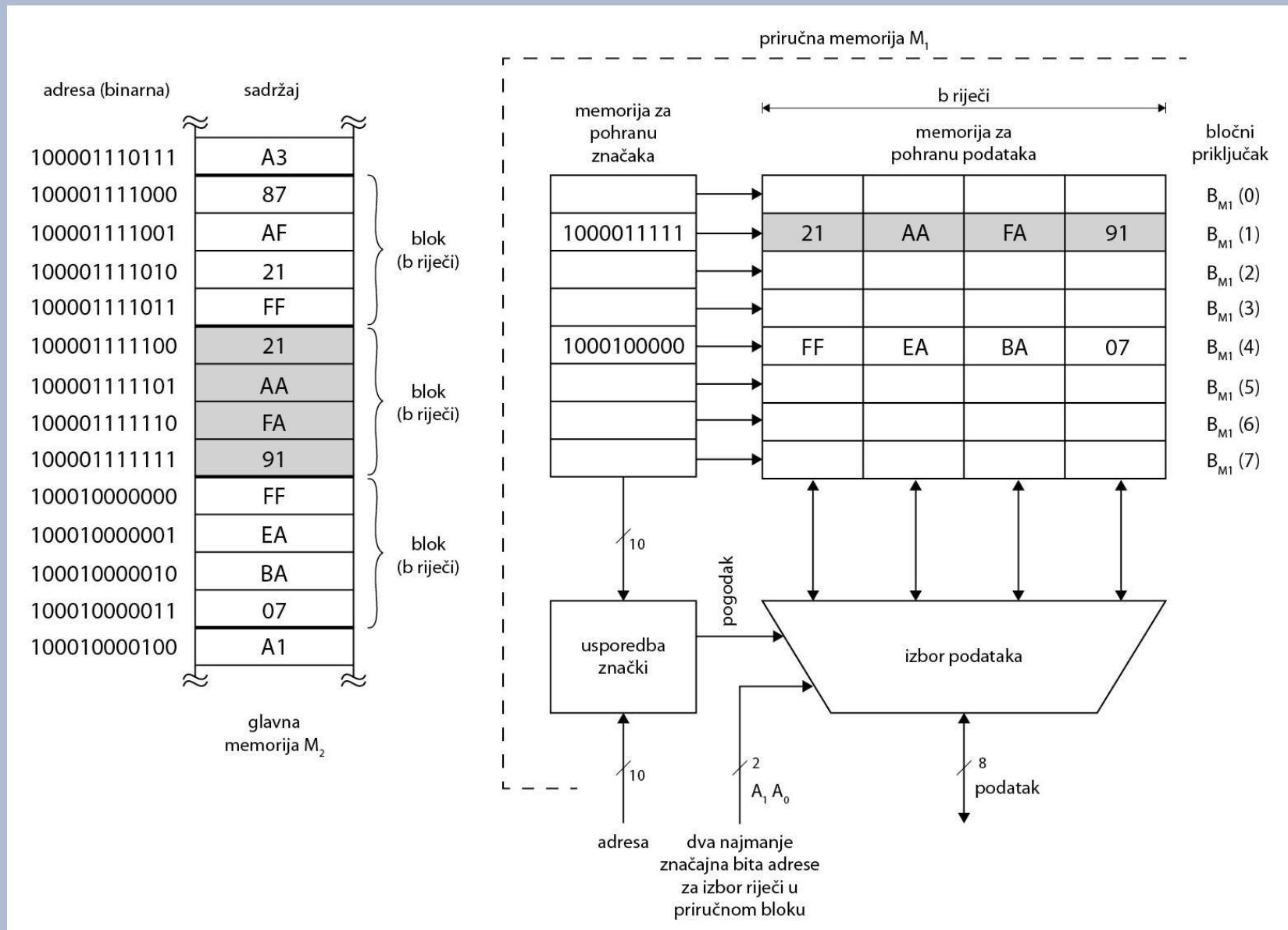
- Glavna memorija koja se sastoji od  $2^M$  slijednih riječi, razdijeljena je na slijedne blokove koji se sastoje od  $b$  riječi
  - glavna memorija ima  $B_{M2} = 2^M / b$  blokova;  $b$  je također potencija broja 2;  $b = 2^m$ , pri čem je  $m \ll M$

Memorija za pohranu podataka (u priručnoj memoriji!) sastoji se od  $B_{M1}$  priručnih blokova ili linija koji su također veličine  $b = 2^m$

- svakom je priručnom bloku pridružena značka (pohranjena u memoriji za pohranu značaka) koja sadržava informaciju o adresi bloka pokazujući na početnu lokaciju odgovarajućeg bloka u glavnoj memoriji

Vrijedi  $B_{M2} \gg B_{M1}$

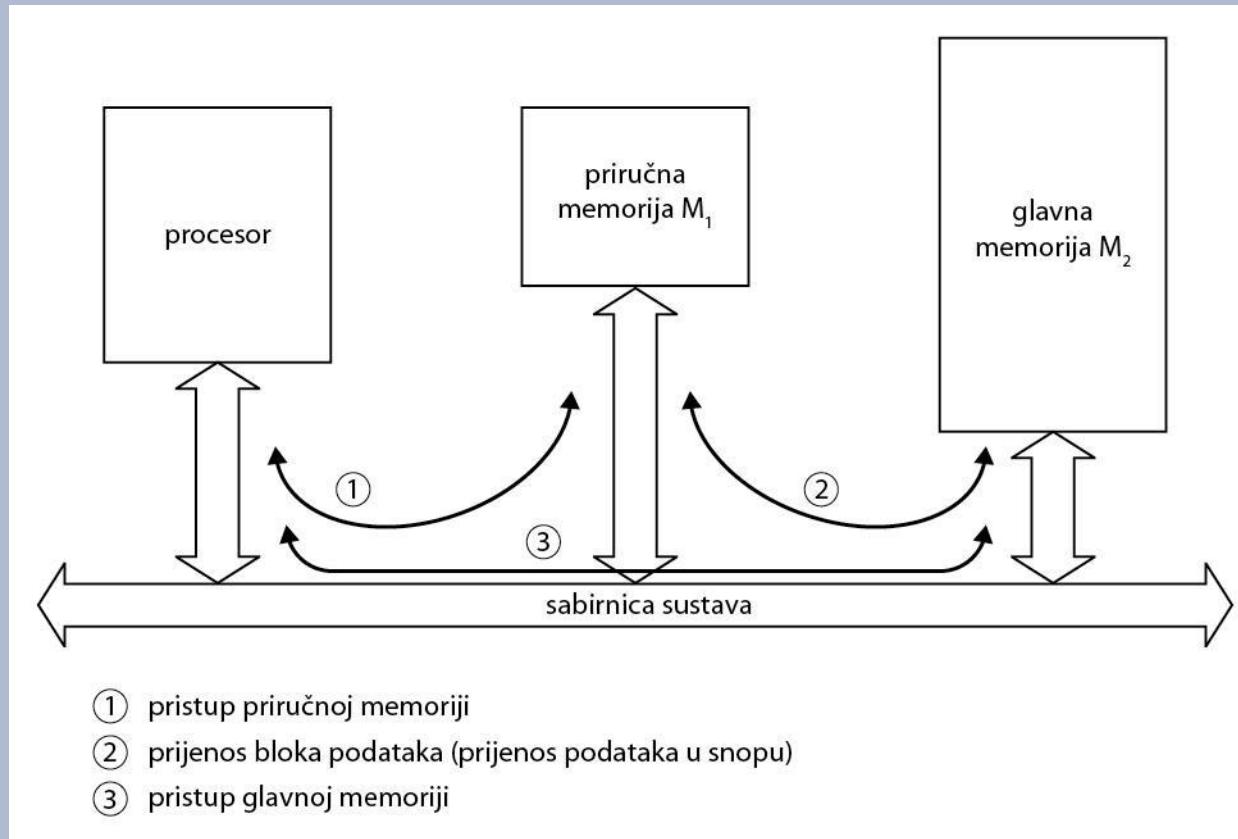
Priručni blok i značka smještaju se u priručnoj memoriji u tzv. bločni priključak



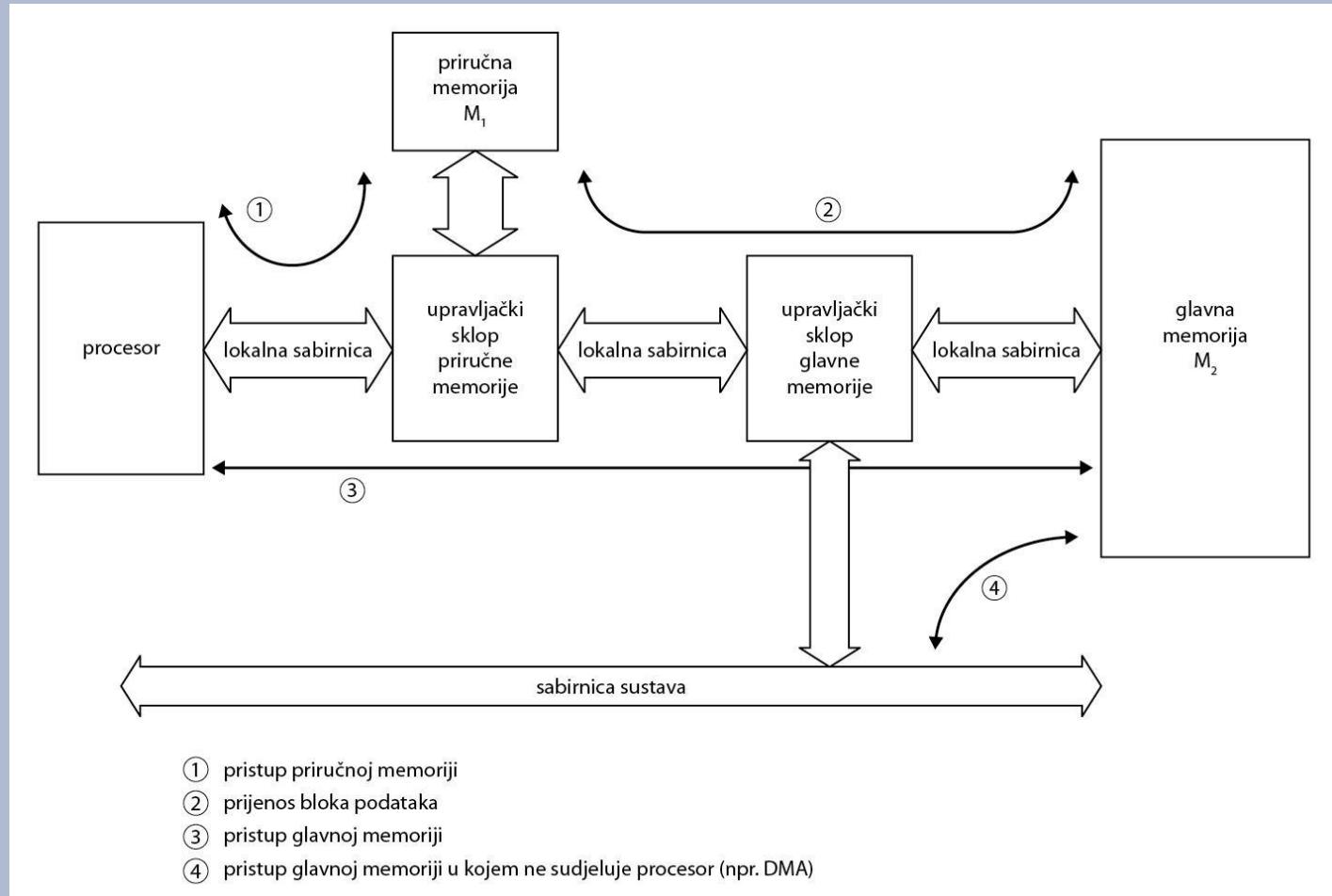
Prema načinu uključivanja priručne memorije razlikujemo dvije osnovne organizacije:

- a) “pogled sa strane” (engl. look-aside)
- b) “pogled kroz” (engl. look-through)

a)

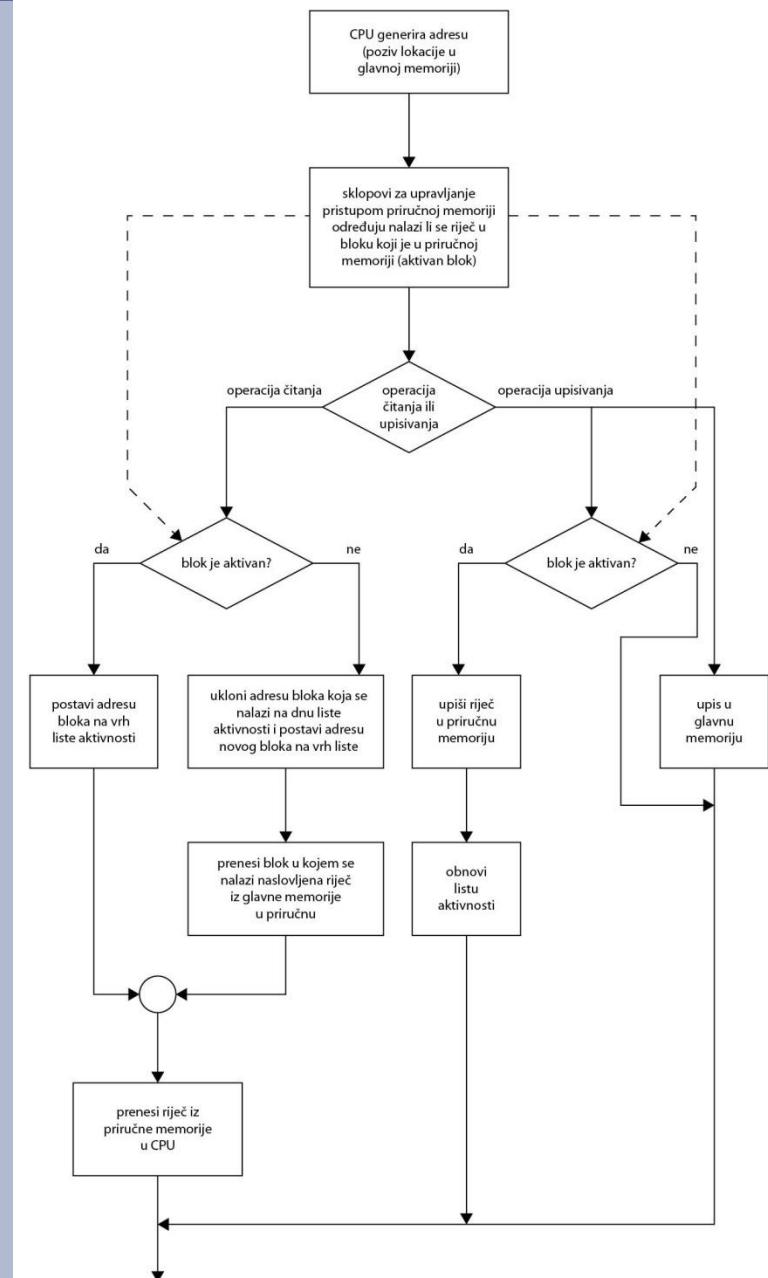


b)



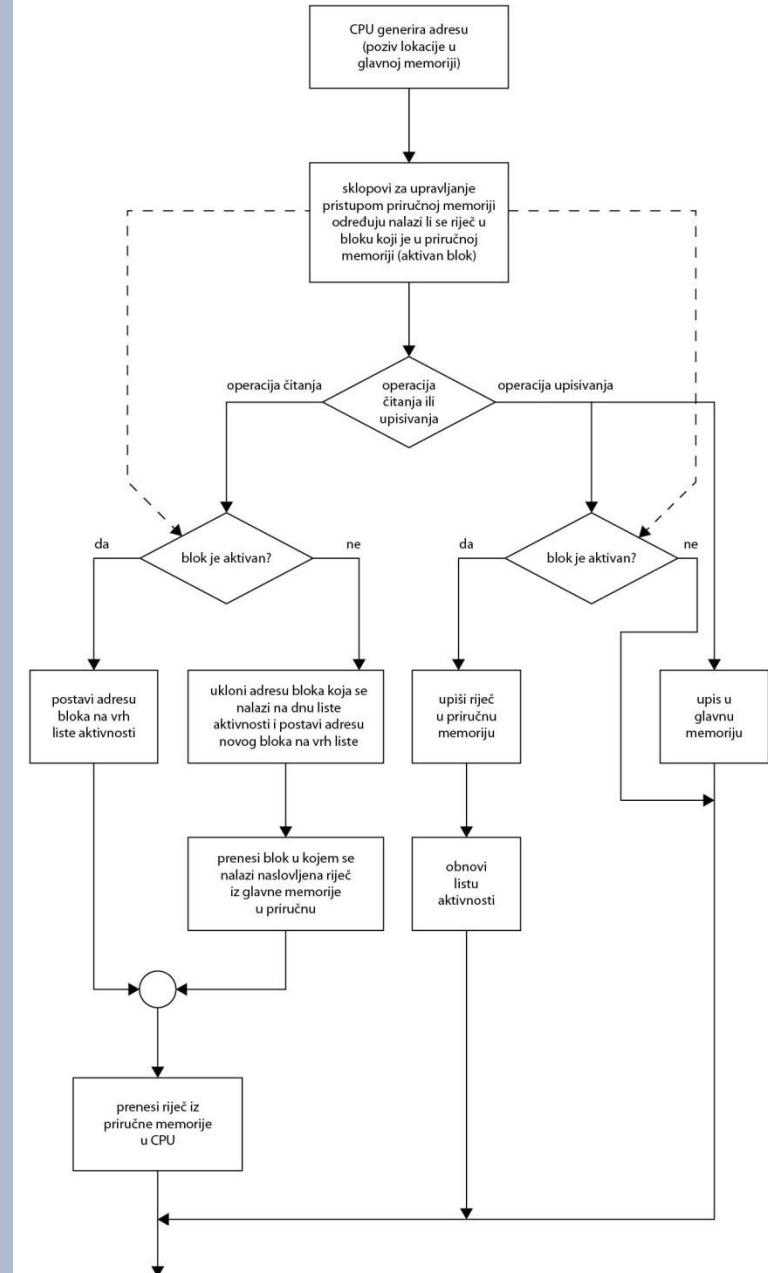
## Djelovanje priručne memorije

- utvrđuje se nalazi li se riječ koju referencira procesor u priručnoj memoriji, tj. utvrđuje se nalazili se ona u nekom bloku u bločnom priključku priručne memorije (uspoređuju se značke sa značajnijim bitovima adrese – ako se tijekom uspoređivanja dogodi podudaranje – **pogodak! /blok koji sadržava referenciranu riječ nalazi se u priručnoj memoriji – aktivan blok/**)



## Dogodio se pogodak!

- Operacija čitanja – referencirana se riječ **dohvaća iz priručne memorije i pribavlja procesoru /glavna memorija ne sudjeluje u toj operaciji/**
- Operacija pisanja – istodobno se podatak upisuje u odgovarajuću lokaciju priručne memorije ali i u lokaciju glavne memorije. Takav se postupak obnavljanja glavne memorije naziva “**pohranjivanje-kroz**” (engl. store-through) /inačica upisivanje u međuspremnik (engl. write buffer)/

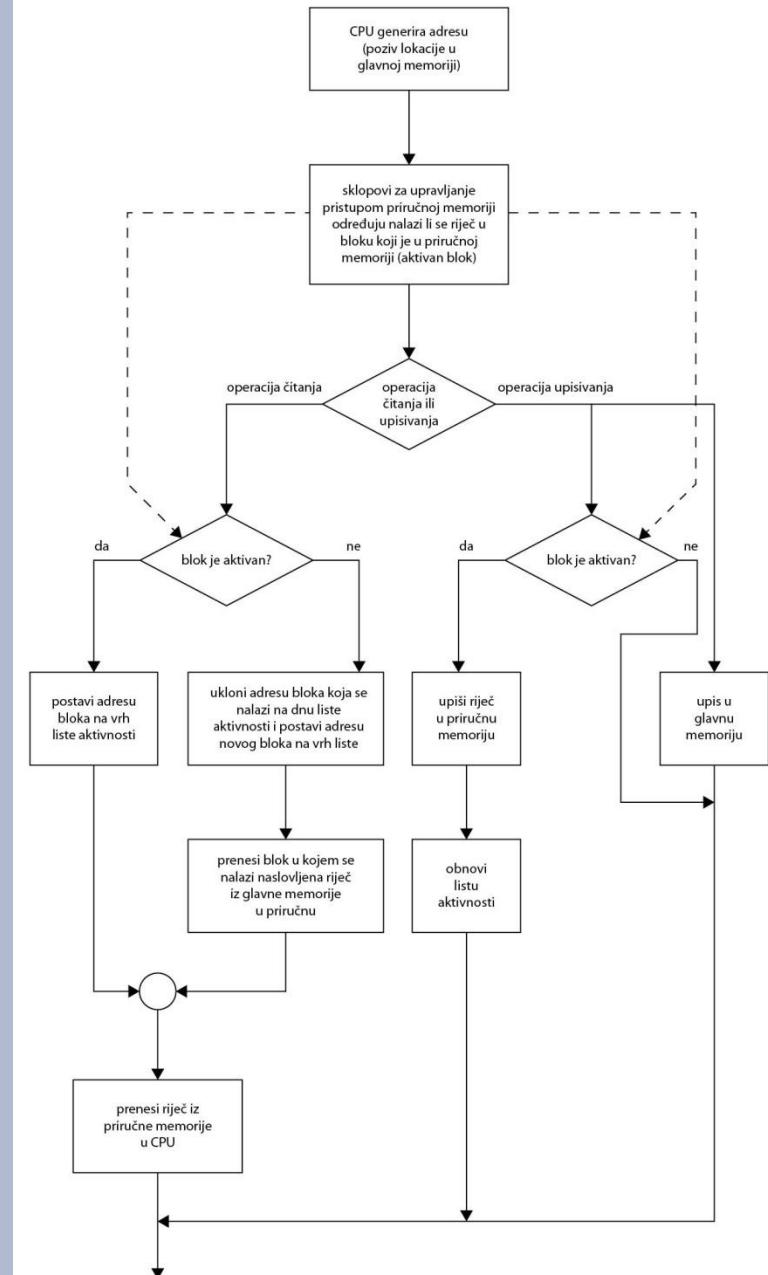


Druga metoda obnavljanja sadržaja glavne memorije naziva se “kopiranje nazad” (engl. write back, copy back) i sastoji se od toga da se podatak upisuje samo u lokaciju u priručnoj memoriji, a da se upis označi posebnom zastavicom pridruženom bloku. Ta se zastavica naziva “prljavi bit” (engl. dirty bit) i njome se označava blok koji će se morati upisati natrag u glavnu memoriju tijekom zamjene blokova.

## Dogodio se promašaj!

Uspoređivanje adrese sa značkama bilo je **neuspješno!**

- Operacija čitanja – adresa se prosljeđuje glavnoj memoriji. Modul za rukovanje u slučaju promašaja na temelju informacije o zamjeni blokova, određuje koji se blok zamjenjuje u priručnoj memoriji blokom koji treba dohvatiti iz glavne memorije. Nakon ili tijekom prijenosa bloka, referencirana se riječ šalje procesoru.



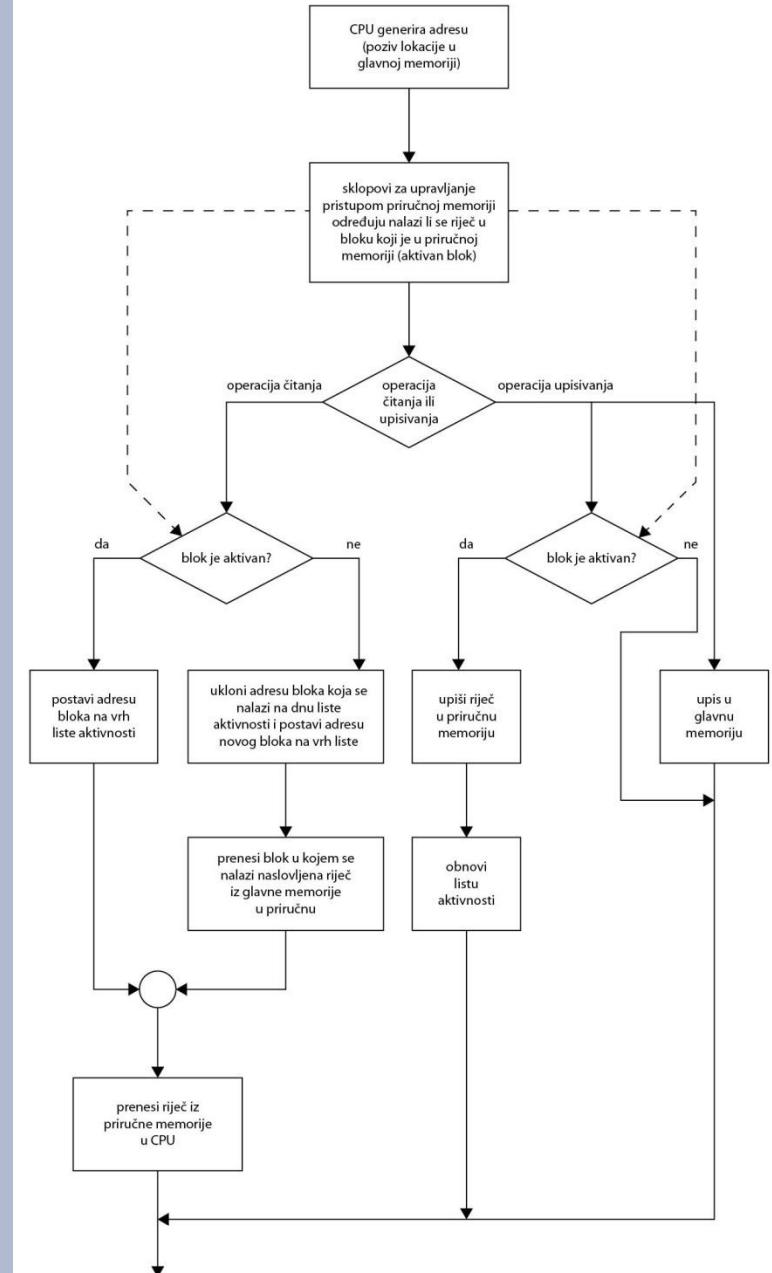
## Dogodio se promašaj!

- Operacija pisanja – dva pristupa:

1) podatak se izravno upisuje u glavnu memoriju – sadržaj priručne memorije se ne mijenja;

/tehnika *write-no allocation*/

2) odgađa se upis u memoriju da bi se prenio blok u priručnu memoriju i tek onda obavi upis u priručnu memoriju  
/tehnika *write allocation*/



# Performansa priručne memorije

Omjer pogotka H (engl. hit ratio)

$$H = (\text{broj referenciranja u kojima je postignut pogodak}) / (\text{ukupan broj referenciranja})$$

Omjer promašaja MR = 1 - H (engl. miss ratio)

U nekom vremenskom intervalu možemo ocijeniti srednje vrijeme pristupa memorijskom sustavu  $t_A$ :

$$t_A = t_{M1} \times H + (1-H) \times t_{M2},$$

gdje je  $t_{M1}$  vrijeme pristupa priručnoj memoriji,

$t_{M2}$  vrijeme pristupa glavnoj memoriji

Primjer:

$$t_{M2} = 250 \text{ ns}$$

$$t_{M1} = 25 \text{ ns}$$

Omjer pogotka H neka je 0.98, što je vrijednost koja se nalazi u okviru tipičnih vrijednosti za priručne memorije

$$t_A = t_{M1} \times H + (1-H) \times t_{M2} = 25 \times 0.98 + (1-0.98) \times 250$$

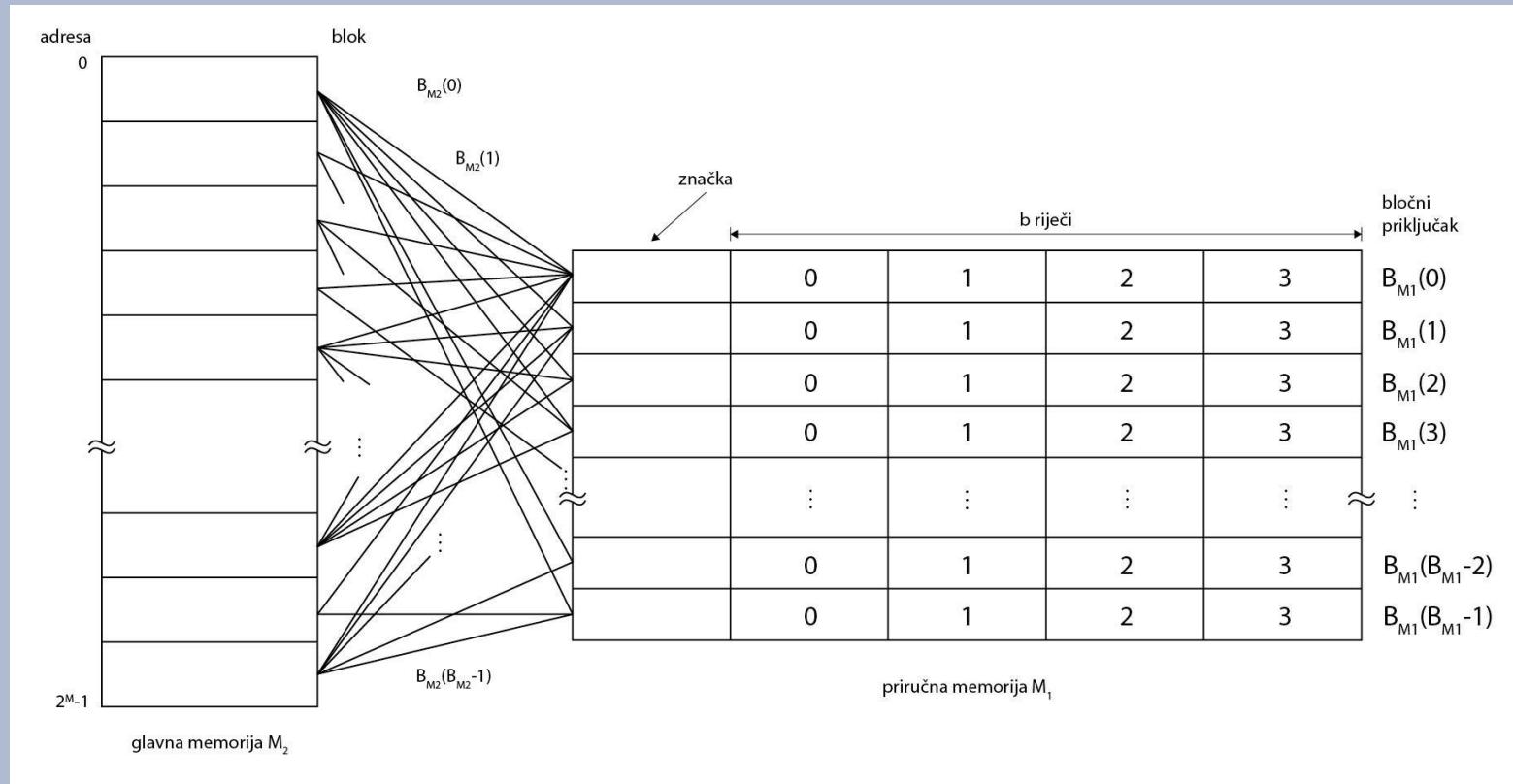
$$t_A = 24.5 + 5.0 = 29.5 \text{ ns}$$

uvodenjem priručne memorije srednje vrijeme pristupa memoriji popravilo se skoro za jedan razred veličine

# Organizacija priručne memorije

Način smještanja blokova u bločne priključke

- **priručna memorija s potpuno asocijativnim preslikavanjem**  
(engl. fully associative mapping)



- Priručna memorija s izravnim preslikavanjem

Pravilo: Blok  $B_{M2}(i)$  iz glavne memorije smješta se u bločni priključak  $B_{M1}(j)$  priručne memorije pri čemu vrijedi

$$j = i \text{ (modulo } B_{M1})$$

Primjer:

- kapacitet priručne memorije 64 KB, veličina priručnog bloka 4 bajta;

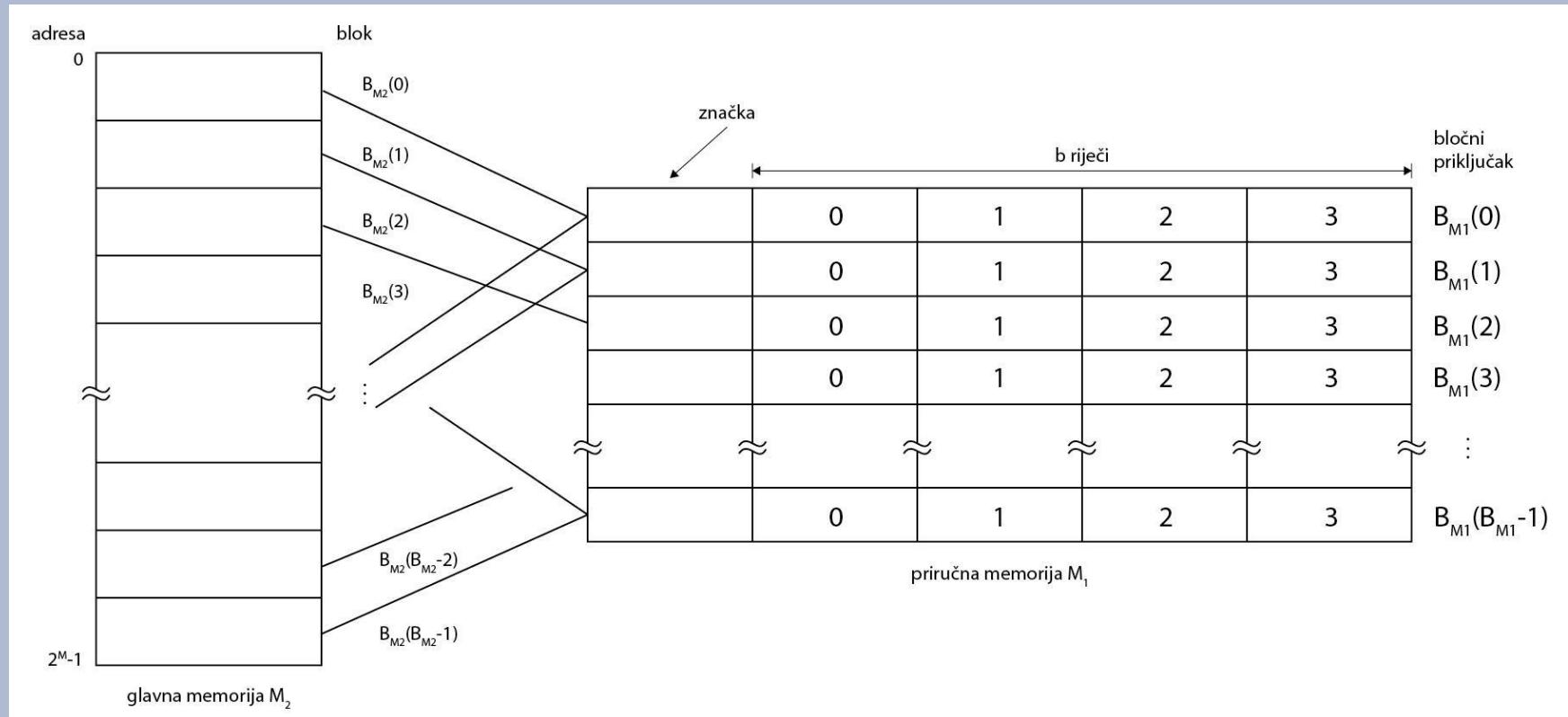
$$B_{M1} = 64 \text{ K} / 4 = 16 \text{ K} = 16384$$

- kapacitet glavne memorije 16 MB

$$B_{M2} = 16 \text{ M} / 4 = 4 \text{ M} = 4194304$$

Blok  $B_{M2}(32780)$  preslikat će se u  $B_{M1}(j)$  pri čemu je  $j$ :

$$j = 32780 \text{ (modulo } 16384) = 12$$



Priručna memorija s izravnim preslikavanjem

$$B_{M1} = 4$$

$$B_{M2} = 8$$

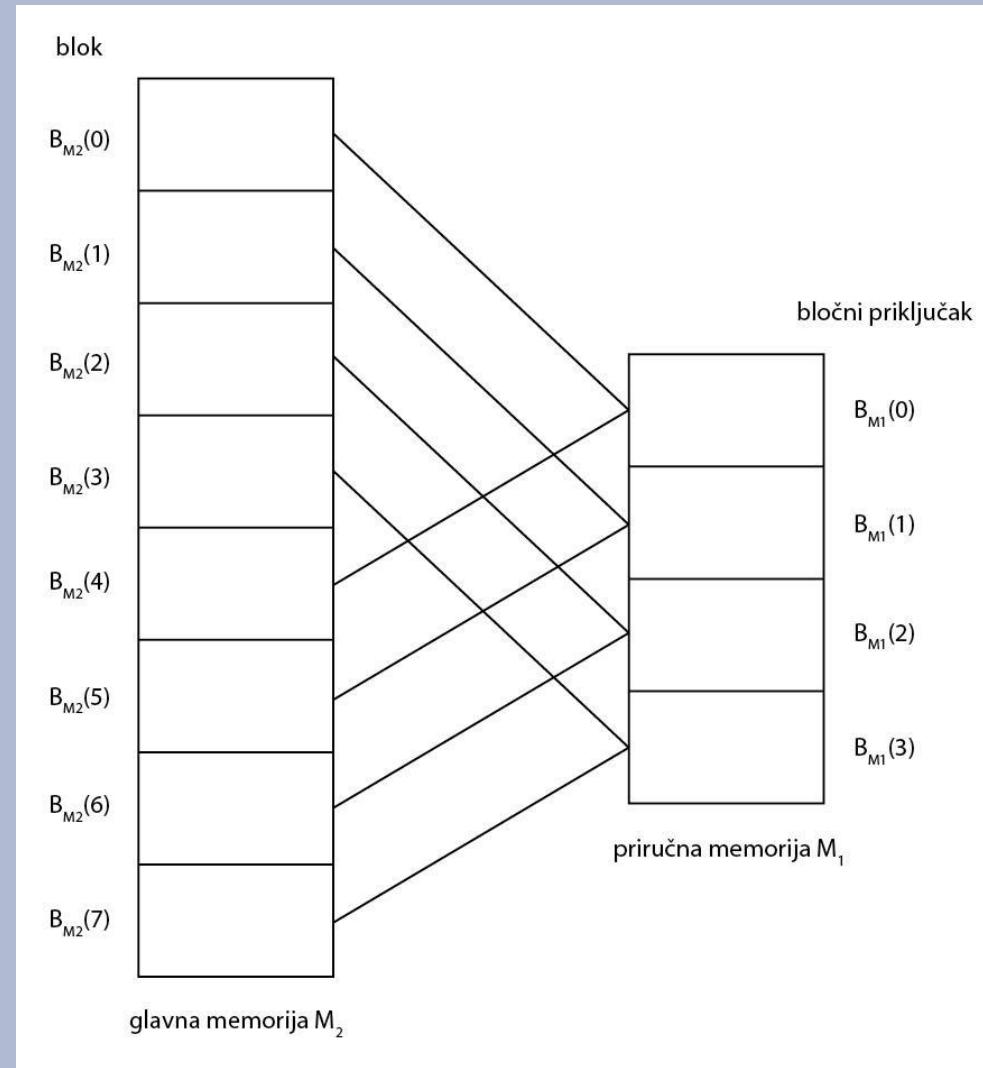
$$B_{M1}(j); j = 0, 1, 2, 3$$

$$B_{M2}(i); i = 0, 1, \dots, 7$$

Npr.  $B_{M2}(6)$  se preslikava u  
 $j = i \text{ (modulo } B_{M1})$   
 $j = 6 \text{ (modulo } 4) = 2$

---

$$B_{M2}(7) \rightarrow B_{M1}(3)$$



## Priručna memorija sa **skupnim asocijativnim preslikavanjem** (engl. set-associative mapping)

- kombinacija prethodno opisanih dvaju načina smještanja blokova

*bločni priključci priručne memorije organizirani su u skupine tako da mehanizam preslikavanja dopušta da se blok iz glavne memorije priključuje u bilo koji bločni priključak koji pripada odgovarajućoj skupini*

- priručna memorija ima  $B_S$  skupina bločnih priključaka

- svaka skupina ima  $k$  bločnih priključaka (ukupno je  $B_S \times k$  bločnih priključaka u priručnoj memoriji)

**Blok iz glavne memorije  $B_{M2}(i)$  može se priključiti u bilo koji od  $k$  slobodnih bločnih priključaka skupine  $j$ , pri čemu je**

$$j = i \text{ (modulo } B_S\text{)}$$

Pretpostavimo da imamo priručnu memoriju koja ima  $B_{M1} = 128$  bločnih priključaka organiziranih u 32 skupine  $B_s = 32$ ;  $B_s(0) - B_s(31)$ . Svaki bločni priključak ima  $b = 16$  riječi.

U svakoj skupini imamo  $B_{M1} / B_s = 4$  bločna priključka ("četveroputna asocijativna memorija")

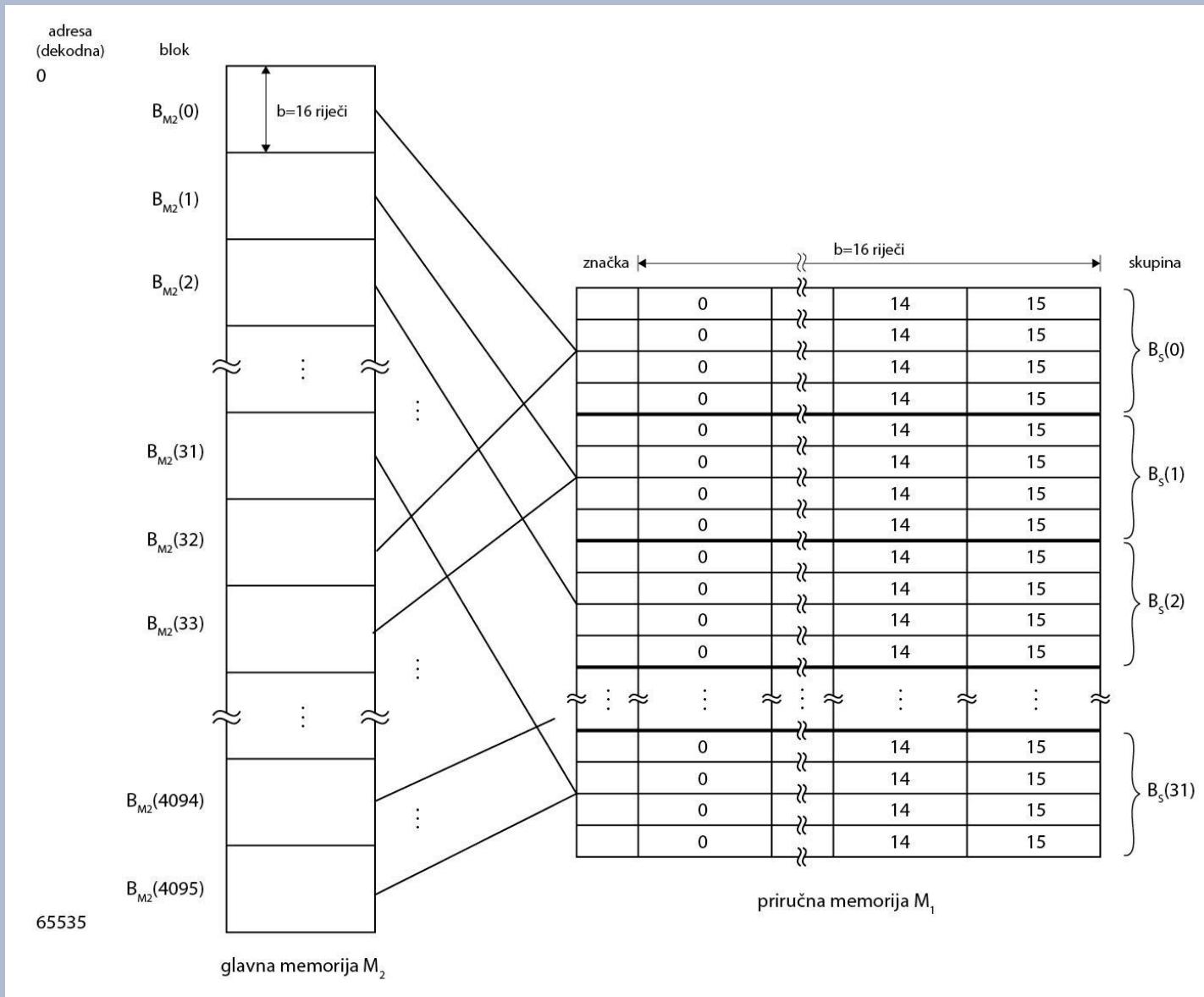
Glavna memorija neka je kapaciteta 64 K riječi i organizirana je u  $B_{M2} = 64 K / b = 64 K / 16 = 4096$  blokova. Glavna memorija ima blokove  $B_{M2}(0) - B_{M2}(4095)$ .

Primjer preslikavanja:

Neka se referencirana riječ nalazi u bloku  $B_{M2}(255)$ ;  $i = 255$ .

Blok  $B_{M2}(255)$  može se priključiti na jedan od četiri slobodna priključka skupine  $j$ :

$$j = i \text{ (modulo } B_s) = 255 \text{ (modulo } 32) = 31$$



## Način zamjene blokova

- kad se novi blok iz glavne memorije treba prenijeti u priručnu memoriju i kad su svi bločni priključci zauzeti, **potrebno je donijeti odluku o tome koji se od blokova u priručnoj memoriji mora zamijeniti**
- u slučaju priručne memorije s izravnim preslikavanjem odluka je već donesena –  $j = i \text{ (modulo } B_{M1})$
- za priručne memorije s potpuno asocijativnim preslikavanjem i priručne memorije sa skupnim asocijativnim preslikavanjem odluka se donosi na temelju **algoritma zamjene**

## Algoritmi zamjene:

- FIFO
- LRU (Least Recently Used) izabire se blok za zamjenu koji je najdulje u priručnoj memoriji najduže bez da bude referenciran
- NRU (Not Recently Used) – algoritmom se zamjenjuje blok koji nije u posljednje vrijeme bio referenciran  
/”posljednje vrijeme” – ovisno o implementaciji/
- LFU (Least Frequently Used)
- Random

## Načini obnavljanja sadržaja glavne memorije

- pohranjivanje-kroz
- upisivanje u međuspremnik
- kopiranje nazad

## Načini pribavljanja bloka iz glavne memorije

- pribavljanje na zahtjev
- pretpribavljanje

Itanium-2	L1	L2	L3
Kapacitet	16 KB I priručna memorija 16 KB D priručna memorija	256 KB I&D priručna memorija	3 MB I&D priručna memorija
Način smještanja blokova u bločne priključke	Skupna asocijativna (4-putna)	Skupna asocijativna (8-putna)	Skupna asocijativna (24-putna)
Veličina bloka (linije)	64 bajta	128 bajta	128 bajta
Algoritam zamjene blokova	LRU	NRU	NRU
Način obnavljanja sadržaja glavne memorije	Pohranjivanje-skroz	Kopiranje nazad	Kopiranje nazad
Detekcija/ispravljanje pogreške	ECC	ECC (SECDED)	ECC (SECDED)
Mjesto implementacije	Procesorski čip	Procesorski čip	Procesorski čip

Tablica 10.2. Osnovne značajke priručne memorije za Intel Itanium-2 (EPIC ISA)

Proučiti:

Organizacija priručne memorije

- priručna memorija s potpuno asocijativnim preslikavanjem
- priručna memorija s izravnim preslikavanjem
- priručna memorija sa skupnim asocijativnim preslikavanjem
- načini zamjene blokova
- način obnavljanja sadržaja glavne memorije
- način pribavljanja bloka iz glavne memorije

(Građa računala, str. 320 – 334)