FIFO

- FIFO (*First In—First Out*) algoritmom zamene za zamenu se bira blok koji je najranije unet iz operativne memorije u keš memoriju.
-

LRU

- LRU (*Least Recently Used*) algoritmom zamene, za zamenu se bira blok kome se najduže vremena nije pristupalo.
- Mogući način realizacije je da se koristi 2^n brojač po modulu 2^n, gde 2^n predstavlja broj ulaza keš memorije sa asocijativnim preslikavanjem i broj ulaza po skupu keš memorije sa asocijativnim preslikavanjem skupa.

LRU

- Svakom od 2^n ulaza keš memorije dodeljuje se jedan od 2^n brojača po modulu 2^n.
- Brojači se tokom rada tako ažuriraju da je u njima uvek 2^n različitih vrednosti i da brojač ulaza kome se najduže vremena nije pristupalo ima sve jedinice i time vrednost 2^n-1.

LRU

- Svakom od 2^n ulaza keš memorije dodeljuje se jedan od 2^n brojača po modulu 2^n.
- Brojači se tokom rada tako ažuriraju da je u njima uvek 2^n različitih vrednosti i da brojač ulaza kome se najduže vremena nije pristupalo ima sve jedinice i time vrednost 2^n-1.

LRU

- U trenutku kada ima saglasnosti sadržaj brojača ulaza u kome je otkrivena saglasnost se upoređuje sa sadržajima brojača svih preostalih ulaza.
- Brojači koji imaju manju vrednost od brojača ulaza u kome je otkrivena saglasnost se inkrementiraju, brojači koji imaju veću vrednost od brojača ulaza u kome je otkrivena saglasnost se ne menjaju i brojač ulaza u kome je otkrivena saglasnost se postavlja na nulu.

LRU

- U trenutku kada nema saglasnosti za zamenu se bira ulaz čiji brojač ima sve jedinice i time vrednost 2^n-1, brojači svih preostalih ulaza imaju manju vrednost od brojača ulaza koji je odabran za zamenu pa se inkrementiraju i brojač ulaza koji je odabran za zamenu se postavlja na nulu.

LRU

- Na početku rada brojači 2^n ulaza treba tako da se inicijalizuju da u njima bude 2^n različitih vrednosti, pri čemu brojače ulaza treba inicijalizovati na vrednosti 2^n-1, $2^n- 2$, ..., *1* i *0* saglasno redosledu po kome se želi popunjavanje ulaza.
- Time se i za popunjavanje keš memorije i za zamenu blokova popunjene keš memorije koristi isti mehanizam.

PSEUDO LRU

- PSEUDO LRU algoritmom zamene se pokušava realizacija principa LRU algoritma zamene sa jednostavnijim hardverom.
- Mogući način realizacije je dat za keš memorije sa asocijativnim preslikavanjem i četiri ulaza i keš memorije sa asocijativnim preslikavanjem skupa i četiri ulaza po skupu.
- Ovaj algoritam zamene često se koristi kod asocijativnog preslikavanja skupa kod koga je broj ulaza po skupu najčešće dva, četiri ili osam.

PSEUDO LRU

- Ulazi 0 do 3 su upareni u dve grupe, pri čemu ulazi 0 i 1 jedan čine jednu grupu i ulazi 2 i 3 drugu grupu.
- Ovim grupama se dodeljuje indikator I_2 koji se postavlja na 0 kad god se pristupi ulazu 0 ili 1 i na 1 kod god se pristupi ulazu 2 ili 3.

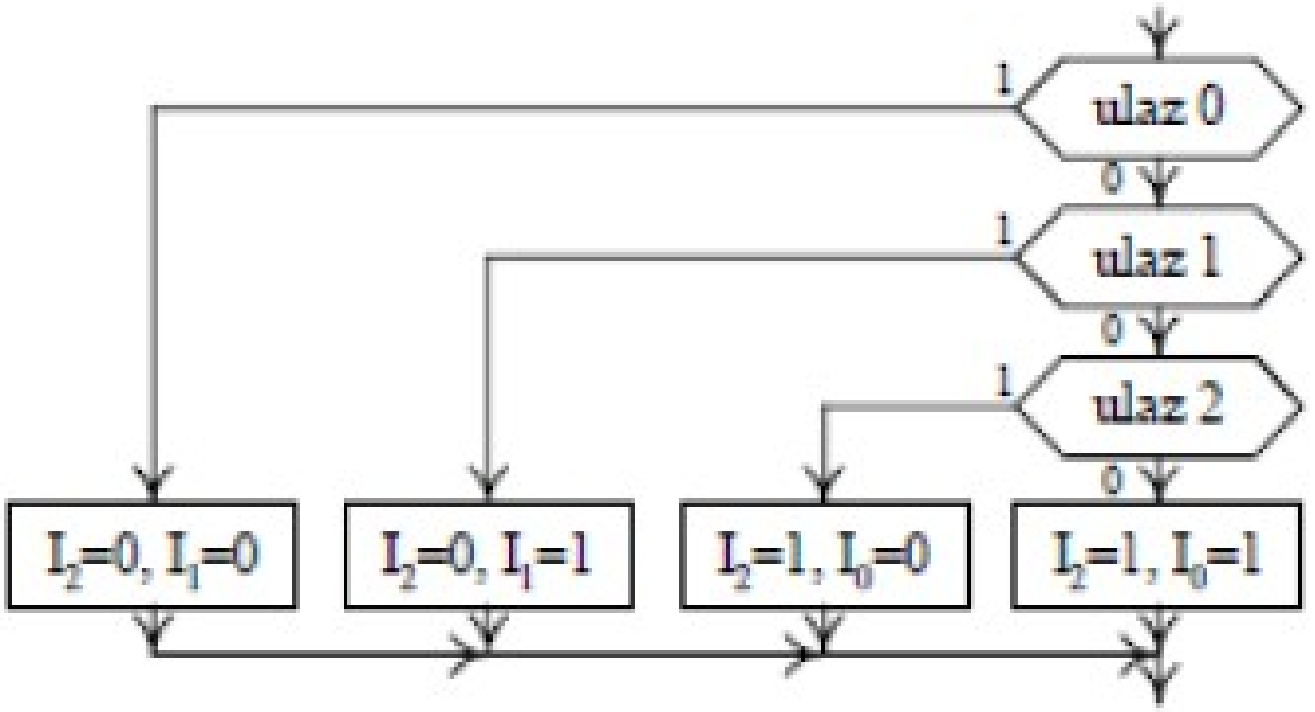
PSEUDO LRU

- Ulazima iz grupe koju čine ulazi 0 i 1 dodeljuje se indikator I_1 koji se postavlja na 0 kad god se pristupi ulazu 0 i na 1 kod god se pristupi ulazu 1.

PSEUDO LRU

- Ulazima iz grupe koju čine ulazi 2 i 3 dodeljuje se indikator I_0 koji se postavlja na 0 kad god se pristupi ulazu 2 i na 1 kod god se pristupi ulazu 3.
- Dijagram toka PSEUDO LRU algoritma zamene pri ažuriranju je dat na sledecem slajdu

PSEUDO LRU



Ažuriranje operativne memorije

- Ažuriranje operativne memorije određuje kako se kod operacije upisa menja sadržaj u operativnoj memoriji.
- Pri tome se, kod zahteva za upis, mogu javiti dve situacije.
- Prva je da u keš memoriji postoji saglasnost, a druga da nema saglasnosti.

Ažuriranje operativne memorije

- Za slučaj kada je u keš memoriji otkrivena saglasnost, postoje dva pristupa i to **upiši skroz** (*write through* ili *store through*) i **vrati nazad** (*write back* ili *copy back*).
- Kod pristupa upiši skroz, pri svakom zahtevu za upis istovremeno se vrši upis i u keš memoriju i u operativnu memoriju.

Ažuriranje operativne memorije

- Kod pristupa vrati nazad, pri svakom zahtevu za upis vrši se upis samo u keš memoriju, pa odgovarajući sadržaj u operativnoj memoriji nije ažuran.
- Zbog toga se za svaki blok u keš memoriji vodi evidencija o tome da li je modifikovan ili ne.

Ažuriranje operativne memorije

- Ukoliko je kasnije potrebno dovući novi blok iz operativne memorije na mesto nekog bloka u keš memoriji koji je nekim od prethodni upisa modifikovan, potrebno je, najpre, dati blok keš memorije vratiti u operativnu memoriju i time obezbediti da i sadržaj u operativnoj memoriji bude ažuran.

Ažuriranje operativne memorije

- Pored toga, kada se nekom procesu oduzima procesor, treba proveriti koji su blokovi u keš memoriji modifikovani, pa ih, radi ažuriranja sadržaja u operativnoj memoriji, vratiti iz keš memorije u operativnu memoriju.
- Stoga kod keš memorija koje koriste ovaj pristup ažuriranja sadržaja operativne memorije, pored zahteva za čitanje i upis, postoje i zahtevi za selektivno i kompletno vraćanje blokova iz keš memorije u operativnu memoriju (*flush*).

Ažuriranje operativne memorije

- Prednost pristupa upiši skroz je u tome da je operativna memorija uvek ažurna čime je obezbeđena konzistentnost sadržaja operativne i keš memorije.
- Nedostatak ovog pristupa je u obraćanju operativnoj memoriji pri svakom upisu u keš memoriju, čime se bespotrebno opterećuje magistrala upisivanjem međurezultata u operativnu memoriju.

Ažuriranje operativne memorije

- Prednost pristupa vrati nazad je u tome što se operativnoj memoriji i magistrali pristupa samo onda kada se blok vraća iz keš memorije u operativnu memoriju što rezultuje u manjem saobraćaju na magistrali.
- Nedostatak ovog pristupa je potreba da se blok koji se izbacuje iz keš memorije mora najpre vratiti u operativnu memoriju, pa tek onda dovući novi, što znatno usporava odziv keš memorije u slučaju promašaja.

Ažuriranje operativne memorije

- Ovde se vidi da su sve prednosti jednog pristupa ujedno i nedostaci drugog.
- Stoga se pristup vrati nazad koristi tamo gde je magistrala usko grlo sistema, a pristup upiši skroz gde magistrala to nije.

Ažuriranje operativne memorije

- Za slučaj kada je u keš memoriji nije otkrivena saglasnost, postoje dva pristupa i to **dovuci blok** (*write allocate*) i **ne dovlači blok** (*no write allocate*).
- Kod pristupa dovuci blok, blok se dovlači iz operativne u keš memoriju, čime se obezbeđuje da se sada u keš memoriji otkriva saglasnost.

Ažuriranje operativne memorije

- Dalji postupak odgovara prethodno opisanoj situaciji za operaciju upisa i otkrivenu saglasnost, u kojoj se upis vrši u keš memoriju, a za ažuriranje sadržaja operativne memorije koristi pristup upiši skroz ili vrati nazad.

Ažuriranje operativne memorije

- Kod pristupa ne dovlači blok, blok se ne dovlači iz operativne u keš memoriju, već se upis vrši samo u operativnu memoriju.
- Obično se uz pristup **vrati nazad** (*write back* ili *copy back*) koristi pristup **dovuci blok** (*write allocate*), dok se uz pristup **upiši skroz** (*write through* ili *store through*) koristi pristup **ne dovlači blok** (*no write allocate*).

Poboljšanja

- U osnovni mehanizam funkcionisanja keš memorije moguće je uvesti neka poboljšanja koja skraćuju vreme čitanja iz i upisa u keš memoriju.

Poboljšanja

- Moguće poboljšanje je u tome da keš memorija, ako se radi o operaciji upisa, odmah dozvoli procesoru da produži sa izvršavanjem tekuće instrukcije bez obzira na to da li je upis zaista izvršen ili nije.
- Time će paralelno keš memorija obavljati upis a procesor izvršavati instrukciju.
- Keš memorija neće moći da prihvati novi zahtev za upis ili čitanje ukoliko se prethodno započeti upis nije završio.

Poboljšanja

- Poboljšanje je moguće učiniti i u slučaju operacije čitanja kada traženi blok nije u keš memoriji, već ga treba dovući iz operativne memorije.
- Tada procesor ne mora da čeka da ceo blok bude prenesen iz operativne u keš memoriju i da tek tada dobije traženi sadržaj.

Poboljšanja

- Keš memorija može procesoru dostaviti traženi sadržaj čim on stigne iz operativne u keš memoriju.
- U tom slučaju procesor može ranije da nastavi izvršavanje tekuće instrukcije, a da se paralelno s time ostatak bloka prenese iz operativne u keš memoriju.

Poboljšanja

- Pri tome dovlačenje reči bloka treba započeti od reči čije je čitanje zahtevano.
- Kao u prethodnom slučaju, keš memorija opet ne može prihvatiti novi zahtev za čitanje ili upis sve dok se prenos prethodnog bloka ne obavi do kraja.
- Ova tehnika naziva se *by-pass*.

Poboljšanja

- Sledeće poboljšanje je moguće ostvariti u slučajevima kada je potrebno izvršiti vraćanje modifikovanog bloka u operativnu memoriju.
- Da bi se ubrzao taj postupak moguće je u poseban bafer privremeno smestiti ceo blok koji se vraća i odmah preći na dovlačenje novog bloka iz operativne memorije.

Poboljšanja

- Tek po završetku dovlačenja novog bloka prelazi se na vraćanje u operativnu memoriju bloka koji se nalazi u baferu.
- I ovde keš memorija ne može prihvatiti novi zahtev za čitanje ili upis sve dok se cela operacija ne završi do kraja.
- Ovo poboljšanje se naziva *baferisanje*.

Poboljšanja

- Sva navedena poboljšanja imaju za cilj da se procesor što manje zadržava prilikom obraćanja keš memoriji.
- Pri tome se pretpostavlja da se procesor vrlo verovatno neće uskoro ponovo obraćati keš memoriji, pa će do sledećeg obraćanja procesora keš memoriji, keš memorija moći da obavi prethodno započetu operaciju do kraja.

Poboljšanja

- U slučaju operacije upisa postoji više načina da se promene sadržaja operativne i keš memorije realizuju.
- Ako se koristi pristup vrati nazad onda se promena u operativnoj memoriji ostvaruje samo u slučaju vraćanja bloka u operativnu memoriju.

Poboljšanja

- Što se tiče keš memorije kod ovog pristupa se promena u keš memoriji ostvaruje uvek i to i u slučaju da ima saglasnosti i u slučaju da nema saglasnosti, pri čemu se u drugom slučaju to čini tek pošto se blok prenese iz operativne u keš memoriju.

Poboljšanja

- Ako se koristi pristup upiši skroz onda se promena u operativnoj memoriji ostvaruje uvek.
- Što se tiče keš memorije kod ovog pristupa se u slučaju saglasnosti ili upisuje novi sadržaj u keš memoriju ili se ulaz keš memorije proglašava nevažećim.
- U slučaju da nema saglasnosti u nekim situacijama ažurirani blok operativne memorije se dovlači u keš memoriju, dok se u drugim ažurirani blok operativne memorije ne dovlači u keš memoriju.

Poboljšanja

- U osnovni mehanizam funkcionisanja keš memorije je moguće uvesti neka poboljšanja, koja skraćuju vreme čitanja iz i upisa u keš memoriju.
- Moguće poboljšanje je u tome da keš memorija, ako se radi o operaciji upisa, upis izvrši u bafer podatka i odmah dozvoli procesoru da produži sa izvršavanjem tekuće instrukcije.
- Time će se paralelno obavljati upis iz bafera podatka u operativnu memoriju i izvršavati instrukcije procesora.
- Ove tehnike se nazivaju **baferovanje podatka** i **rani start procesora**. .

Poboljšanja

- Poboljšanje je moguće učiniti i u slučaju operacije čitanja kada traženi blok nije u keš memoriji, već ga treba dovući iz operativne memorije.
- Tada procesor ne mora da čeka da ceo blok bude dovučen iz operativne memorije u keš memoriju i da tek tada dobije traženi sadržaj.

Poboljšanja

- Keš memorija može procesoru dostaviti traženi sadržaj čim on stigne iz operativne u keš memoriju.
- U tom slučaju, procesor može ranije da nastavi izvršavanje tekuće instrukcije i da se paralelno ostatak bloka prenosi iz operativne u keš memoriju.
- Pri tome, dovlačenje reči bloka treba započeti od reči čije je čitanje zahtevano.
- Ove tehnike se nazivaju **prosleđivanje** i **rani start procesora**

Poboljšanja

- Sledeće poboljšanje je moguće ostvariti u slučajevima kada je potrebno izvršiti vraćanje modifikovanog bloka odabranog za zamenu iz keš memorije u operativnu memoriju.
- Da bi se ubrzao taj postupak, moguće je postaviti bafer bloka, koji će prihvatiti ceo blok koji se vraća, i odmah preći na dovlačenje bloka iz operativne memorije.
- Tek po završetku dovlačenja bloka iz operativne u keš memoriju, prelazi se na vraćanje bloka iz bafera bloka u operativnu memoriju.
- Ovo poboljšanje se naziva **baferovanje bloka podataka**.

LRU

- U trenutku kada nema saglasnosti za zamenu se bira ulaz čiji brojač ima sve jedinice i time vrednost 2^n-1, brojači svih preostalih ulaza imaju manju vrednost od brojača ulaza koji je odabran za zamenu pa se inkrementiraju i brojač ulaza koji je odabran za zamenu se postavlja na nulu.