

UPIS PODATAKA U KEŠ MEMORIJU

- Situacija pri upisu u keš nešto je komplikovanija nego pri čitanju. Razlog je taj što **upis u keš blok ne može početi pre no što se proverom etikete ustanovi pogodak**. Zbog ovoga upis u keš traje obično duže od čitanja.
- Dodatni problem je to što pri upisu procesor definiše dužinu podatka koji upisuje (obično 1-8 bajtova), i samo taj deo bloka podataka može biti izmenjen.
- To povlači pribavljanje sadržaja adresiranog keš bloka, modifikovanje odgovarajućeg dela bloka podataka, i vraćanje modifikovanog bloka podataka u keš blok iz koga je uzet.

Ponašanje keša pri upisu

- * Prema aktivnostima pri pogotku pri upisu, keš memorije mogu biti:
- sa **neposrednim upisom** (*eng. write through ili store through*) - podatak se upisuje u niži nivo memorije, kao i u keš memoriju ako je u njoj prisutan,
- sa **posrednim upisom** (*eng. write back, copy back ili store in*) - podatak se upisuje samo u keš memoriju; modifikovani blok podataka upućuje se u niži nivo memorije tek pri njegovoj zameni u kešu, ili po završetku programa.

Neophodni povratni upisi

* Da bi se pri posrednom upisu izbegla nepotrebna vraćanja nemodifikovanih blokova pri zameni, svakom keš bloku pridružuje se indikator modifikovan. Ovaj indikator svojim stanjem pokazuje da li je blok podataka modifikovan (izmenjen) za vreme prisustva u kešu.

Bafer upisa pridružen kešu

- Pri neposrednom upisu, upis podatka u niži nivo memorije blokira procesor do završetka upisa. Ovo se može izbeći umetanjem bafera upisa između keš memorije i nižeg nivoa memorije.
- Upisom podatka u bafer upisa procesor se deblokira, iako upis još nije prosleđen do odredišta.
- Ako je pri pokušaju upisa u bafer upisa on već pun, procesor ostaje blokiran sve dok se u baferu upisa ne oslobodi mesto za prihvatanje novog podatka.

Performanse keš memorija

- U najjednostavnijem slučaju čitanja podataka iz keš memorije, efektivno vreme pristupa keš memorije može se izraziti kao:

$$T_{ef} = T_{pog} + mT_{prom}$$

- U ovom izrazu T_{pog} je vreme potrebno za čitanje podatka iz keša pri pogotku, T_{prom} je vreme potrebno za pribavljanje promašenog podatka iz nižeg nivoa memorije, tzv. promašajna kazna, i m je faktor promašaja.

Vremenski parametri keša

- Razmotrimo aktivnosti pri ukupno n obraćanja memorijskom sistemu, obrazovanom od jedinstvene keš memorije KM (za instrukcije i podatke) i glavne memorije GM.
- Neka su n_c i n_u broj čitanja iz, i upisa u, memorijski sistem, pri čemu je $n = n_c + n_u$.
Dalje, neka su:
 - T_c - perioda takta keša,
 - T_{cb} - vreme pribavljanja bloka podataka iz GM u KM,
 - T_{ub} - vreme upisa žrtvovanog modifikovanog bloka podataka iz KM u GM,
 - T_{uGM} - vreme upisa reči podatka iz KM u GM

Provera pri upisu

- n_{prom} - broj promašaja u n obavljenih pristupa kešu,
- $n_{\check{\text{c}}\text{prom}}$ - broj promašaja pri čitanju u n obavljenih pristupa kešu,
- n_{zmd} - broj zamenjenih modifikovanih blokova podataka u n pristupa memoriji.
- Pošto se upisom blok podataka u kešu modifikuje, pre upisa moramo se uveriti da je blok podataka u koji upisujemo prisutan u kešu. Ta provera zahteva jedan taktni ciklus keša. Sam upis zahteva još jedan taktni ciklus, tako da su za upis potrebna dva taktna ciklusa keša.

Keš sa posrednim upisom

- Za keš sa posrednim upisom imamo:

$$\begin{aligned}n \cdot T_{\text{ef(PU)}} &= n_{\check{c}} \cdot T_c + n_u \cdot 2T_c + n_{\text{prom}} \cdot T_{\check{c}b} + n_{\text{zmd}} \cdot T_{ub} \\T_{\text{ef(PU)}} &= (n_{\check{c}}/n) \cdot T_c + (n_u/n) \cdot 2T_c + (n_{\text{prom}}/n) \cdot T_{\check{c}b} + (n_{\text{zmd}}/n) \cdot T_{ub} \\T_{\text{ef(PU)}} &= (p_{\check{c}} + 2p_u) \cdot T_c + m \cdot T_{\check{c}b} + m \cdot p_{\text{mod}} \cdot T_{ub} \\T_{\text{ef(PU)}} &= (1 + p_u) \cdot T_c + m \cdot (T_{\check{c}b} + p_{\text{mod}} \cdot T_{ub})\end{aligned}$$

- Pri postojanju bafera upisa blokova podataka dovoljno velikog kapaciteta između KM i GM, $T_{ub} < T_{\check{c}b}$ i preklapa se sa njim, pa izraz za $T_{\text{ef(PU)}}$ postaje $T_{\text{ef(PU)}} = (1 + p_u)T_c + mT_{\check{c}b}$

Keš sa neposrednim upisom

* Za keš sa neposrednim upisom bez pribavljanja promašenog bloka pri upisu imaćemo:

$$n \cdot T_{ef(NU)} = n_c \cdot T_c + n_u \cdot \max(2T_c, T_{uGM}) + n_{čpro} \cdot T_{čb}$$

$$T_{ef(NU)} = (n_c/n) \cdot T_c + (n_u/n) \cdot \max(2T_c, T_{uGM}) + (n_{čpro}/n) \cdot T_{čb}$$

$$T_{ef(NU)} = p_c \cdot T_c + p_u \cdot \max(2T_c, T_{uGM}) + m_c \cdot T_{čb}$$

Sa baferom upisa

- Pri postojanju bafera upisa reči dovoljno velikog kapaciteta između KM i GM, $T_{uGM} < 2T_c$ i preklapa se sa njim, pa izraz za $T_{ef(NU)}$ postaje

$$T_{ef(NU)} = p_c T_c + p_u * 2T_c + m_c T_{čb}$$

$$T_{ef(NU)} = (1+p_u)T_c + m_c T_{čb}$$

Faktor saobraćaja

* **Faktor saobraćaja je odnos obima prenesenih podataka između para (CP,KM) i GM kada memorijski sistem sadrži keš, i obima prenesenih podataka između CP i GM kada memorijski sistem ne sadrži keš. Označimo faktor saobraćaja sa FS. Onda je on za keš memorije sa posrednim i neposrednim upisom dat izrazima:**

$$FS_{PU} = (n_{\text{prom}} + n_{\text{zmd}})B / (n \cdot w) = (n_{\text{prom}} + n_{\text{prom}} \cdot p_{\text{mod}})B / (n \cdot w) = m(1 + p_{\text{mod}})B / w$$
$$FS_{NU} = (n_{\text{cpro}} \cdot B + n_u \cdot w_u) / (n \cdot w) = (m_c \cdot B + p_u \cdot w_u) / w$$

Faktor saobraćaja

* U ovim izrazima:

* **w je srednja dužina prenošenih podataka između CP i GM kada memorijski sistem ne sadrži keš, a**

*** w_u je srednja dužina podataka upisivanih iz KM u GM kod keša sa neposrednim upisom.**

Uticaj promašaja keša na vrijeme rada CP-a

* **Promašaji podataka u kešu dovode do zastoja u radu procesora. Uticaj promašaja keša na vreme rada CP-a može se iskazati na sledeći način**

$$Vreme_{rada_CP} = (Vreme_{rada_CP_bez_zastoja_u_taktnim_ciklusima} + Vreme_{čekanja_CP_na_memoriju_u_taktnim_ciklusima}) \times Perioda_{takta} \quad (6.15)$$

$$Vreme_{čekanja_CP_na_memoriju_u_taktnim_ciklusima} = Broj_{instrukcija} \times (Pomašaja / Instrukciji) \times Pomašajna_{kazna} \quad (6.16)$$

$$Pomašaja/Instrukciji = Broj_{obraćanja_memoriji_po_instrukciji} \times Faktor_{pomašaja} = m_{naredbi} + m_{podatka} \times Broj_{obraćanja_podacima_u_memoriji_po_instrukciji} \quad (6.17)$$

Klasifikacija promašaja

- **Neizbežni promašaji - pri prvom obraćanju nekom podatku taj podatak nije u kešu, tako da blok sa tim podatkom mora biti prenesen u keš. Ovi promašaji nazivaju se još i promašaji hladnog starta ili promašaji prvog obraćanja.**
- **Kapacitetni promašaji - ako keš ne može sadržati sve blokove podataka potrebne pri izvršenju programa, javljaju se promašaji zbog nedovoljnog kapaciteta, jer se blokovi koji su bili istisnuti iz keša drugim blokovima moraju vratiti u keš.**

Klasifikacija promašaja

- **Konfliktni promašaji** - kod keš memorija sa direktnim ili skupno-asocijativnim preslikavanjem mogu se javiti promašaji zato što se isuviše blokova podataka preslikava u isti skup keš blokova, tako da neki od njih moraju biti istisnuti iz keša a zatim kasnije vraćeni u keš.
- Ovi promašaji nazivaju se još i kolizioni promašaji ili interferentni promašaji.

Mere za smanjenje faktora promašaja

- Smanjenje faktora promašaja može se postići različitim merama. Neke od njih su:
 - povećanje dužine blokova,
 - povećanje asocijativnosti,
 - upotreba posebnog keša za blokove podataka istisnute iz keša, tzv. keš žrtvovanih blokova (eng. victim cache),
 - hardversko pribavljanje unapred instrukcija i podataka (npr korišćenjem stream bafera),
 - kompilatorom upravljano pribavljanje unapred.

Uticaj promene asocijativnosti

- **Analizom podataka ukazuje se na dva iskustvena pravila:**

- za keš memorije osmobločni skupno-asocijativni keš je efikasan koliko i potpuno-asocijativni keš iste veličine,
- keš sa direktnim preslikavanjem veličine C ima približno isti faktor promašaja kao dvobločni skupno-asocijativni keš veličine C/2. Pri konstantnoj promašajnoj kazni,

$$\Delta T_{ef} = \Delta T_{pog} + \Delta m * T_{prom}$$

Mere za sniženje promašajne kazne

- *** Kod savremenih keš memorija sreću se sledeća rešenja:**

- **davanje prioriteta promašajima pri čitanju u odnosu na upise,**
- **pribavljanje podbloka podataka pri promašaju,**
- **rani restart procesora i pribavljanje počev od zahtevane reči,**
- **neblokirajuće keš memorije,**
- **keš memorija u dva nivoa (sekundarni keš).**

Neblokirajuće keš memorije

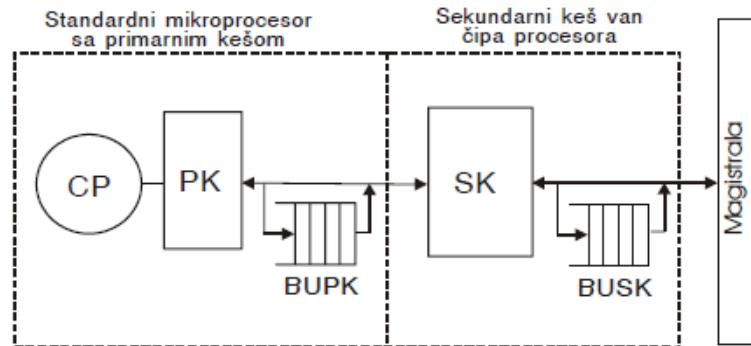
- **Zastoji u radu procesora mogu se značajno smanjiti ako se omogući da procesor nastavi sa pribavljanjem i izvršavanjem drugih instrukcija dok se u keš memoriji pribavlja promašeni blok podataka.**
- **To omogućuju neblokirajuće keš memorije, koje su danas implementirane u većini procesora sa "moćnijim" performansama.**
- **Ova koncepcija zahtjeva uvođenje u keš memoriju dodatnih memorijskih struktura, u kojima se za vreme opsluživanja promašaja čuvaju informacije o formatu i odredištu u procesoru promašenog podatka.**

Keš memorije u dva nivoa

- **Primarni keš**
- **Sekundarni keš**
- **U ovom izrazu m_{L2} je lokalni faktor promašaja, koji se definiše kao količnik broja promašaja u sekundarnom kešu i broja obraćanja sekundarnom kešu.**
- **Globalni faktor promašaja, koji se izražava količnikom broja promašaja keša i ukupnog broja obraćanja procesora memoriji, za sekundarni keš bio bi iskazan proizvodom $m_{L1}m_{L2}$.**
- **Za primarni keš, definicije lokalnog i globalnog faktora promašaja daju isti rezultat.**

$$T_{ef} = T_{pog}^{L1} + m_{L1} \cdot (T_{pog}^{L2} + m_{L2} \cdot T_{prom}^{L2})$$

Konfiguracija sa dva nivoa keša



Legenda:

PK - primarni keš, BUPK - bafer upisa primarnog keša,
SK - sekundarni keš, BUSK - bafer upisa sekundarnog keša,