

Multiprocesorski sistemi (SI4MPS, IR4MPS, MS1MPS)

Laboratorijska vežba 3 – Cache coherence

Cilj laboratorijske vežbe je da studenti korišćenjem simulatora poboljšaju poznavanje osluškujućih (engl. snoopy) protokola za održavanje koherencije keš memorije u multiprocesorskim sistemima.

Opis simulatora

U programu za pregledanje Interneta uneti adresu <http://mups.etf.rs/simulatori/vivio/>. Na datoj adresi su dostupni simulatori za nekoliko protokola za koherenciju keš memorije. Postoje dve verzije simulatora 4.0 i 5.1, u zavisnosti od verzije Vivio plugin-a koji se koristi. Za rad na laboratorijskim vežbama i izradu domaćih zadataka se može koristiti bilo koji

U slučaju da Vivio simulator nije dostupan na lokalnom računaru, ispratiti uputstva za podešavanje Vivio okruženja na lokalnom računaru i podesiti lokalni računar prema njima. Ukoliko se instalacija sama ne pokrene, kompletna instalacija i izvorni kod za Vivio plugin se može preuzeti sa adrese <http://www.scss.tcd.ie/~jones/vivio/source.htm>.

Implementirani osluškujuci protokoli

U simulatoru su implementirani sledeći poništavajući protokoli za koherenciju keš memorije:

1. Write Through-Invalidate (WTI), no-write-allocate
2. Write Through-Invalidate (WTI), write-allocate
3. Modified-Shared-Invalid (MSI)
4. Modified-Exclusive-Shared-Invalid (MESI)
5. Modified-Exclusive-Shared-Invalid (MESI), Jones-ova implementacija
6. Modified-Owned-Exclusive-Shared-Invalid (MOESI)

U simulatoru su implementirani sledeći ažurirajući protokoli za koherenciju keš memorije:

1. Dragon
2. Firefly

Za svaki protokol je dostupan dijagram simulacije u vidu odgovarajuće slike. Takođe, na stranici sa simulatorima je dostupan i dokument sa opisima svih korišćenih osluškujućih protokola za održavanje koherencije keš memorije. Dokument se može preuzeti sa adrese <http://mups.etf.rs/simulatori/vivio5.1/snoopy%20vivio%20dokumentacija.pdf>

Korišćenje simulatora

Simulacije se pokreću izborom željene simulacije na glavnoj stranici. Svaka od simulacija je vizuelno predstavljena sa 4 procesora od kojih svaki može da čita ili piše u jednu od 4 memorijske lokacije dostupne u okviru glavne memorije. Svakom procesoru odgovara tačno jedan keš kontroler koji u sebi sadrži dve linije za podatke u keš memoriji. Preslikavanje memorijskih adresa je direktno, postoje samo dve keš linije, jedna od njih odgovara parnim, a druga neparnim memorijskim adresama. Predstavljene su adresna magistrala i magistrala podataka, kao i deljena linija kontrolne magistrale. Pritiskom na odgovarajuće dugmiće na svakom od procesora mogu se izdavati komande paralelno, čime se daje utisak ponašanja realnog sistema prema protokolu te simulacije.

Scenario simulacije

Neka se razmatra se sledeći scenario:

1. P0,R,A0
2. P0,R,A1
3. P1,R,A0
4. P2,R,A0
5. P0,W,A0
6. P0,W,A2
7. P1,R,A0
8. P0,W,A0

Napisati stanja koherencije u svim procesorima posle svake promene.

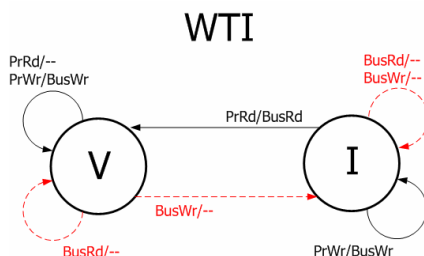
Koliko puta koji od procesora pristupa memoriji? Za svaki pristup navesti razlog.

Koliki je Hit Rate za svaki od procesora? Brojati i čitanja i upisr, prikazati zbirno.

Skicirati opisani sistem posle trenutaka 8.

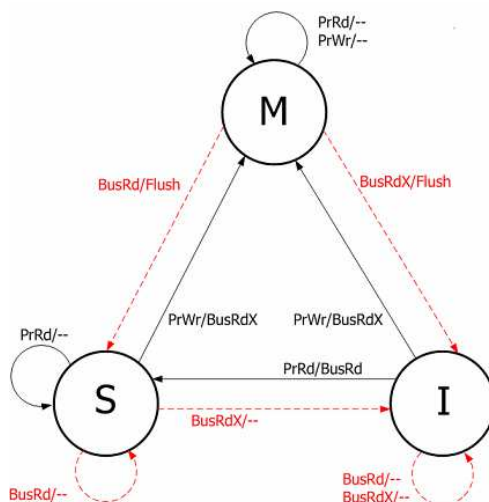
Zadatak 1 - WTI protokol

Prikazati rad WTI protokola na primeru date sekvence.



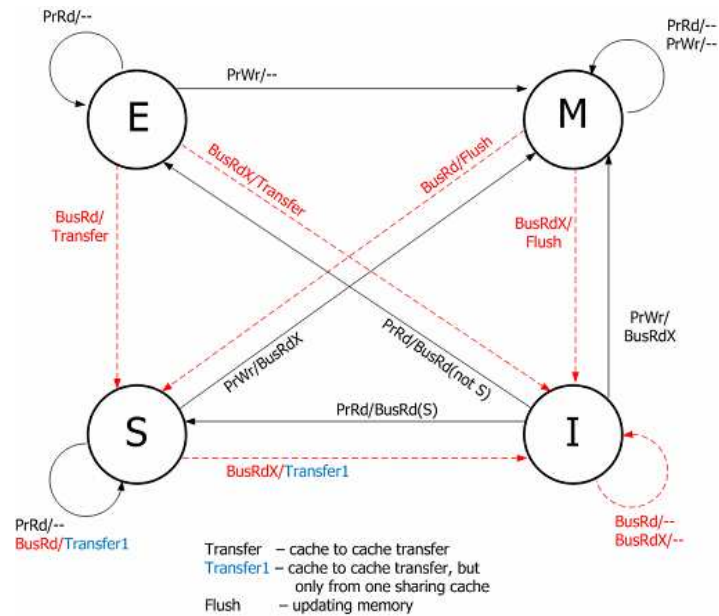
Zadatak 2 - MSI protokol

Prikazati rad MSI protokola na primeru date sekvence.



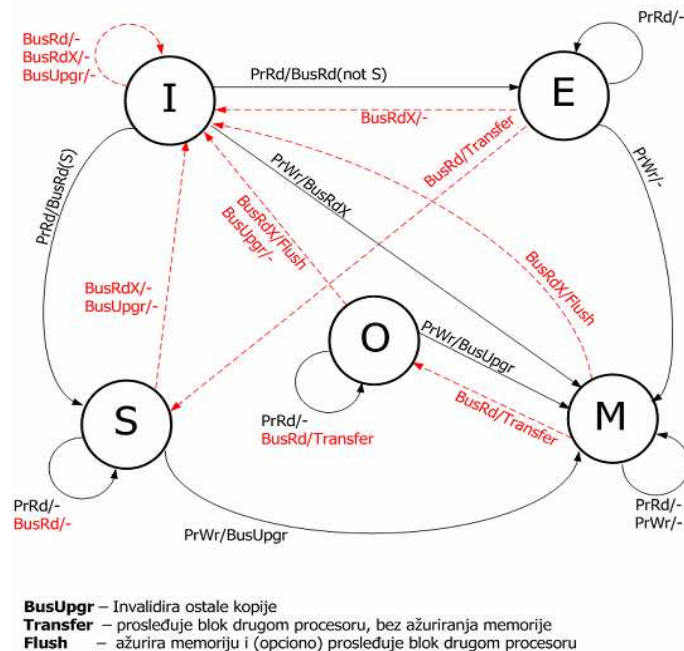
Zadatak 3 - MESI protokol

Prikazati rad MESI protokola na primeru date sekvence.



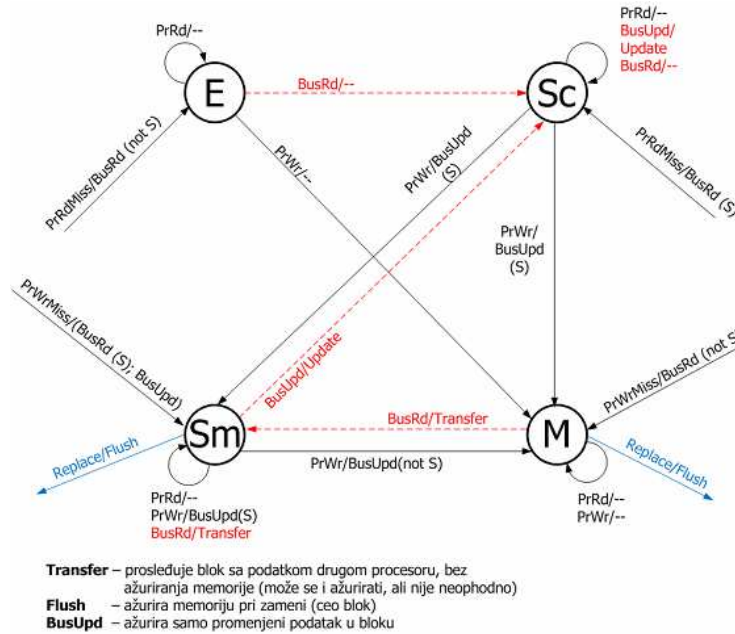
Zadatak 4 - MOESI protokol

Prikazati rad MOESI protokola na primeru date sekvence.



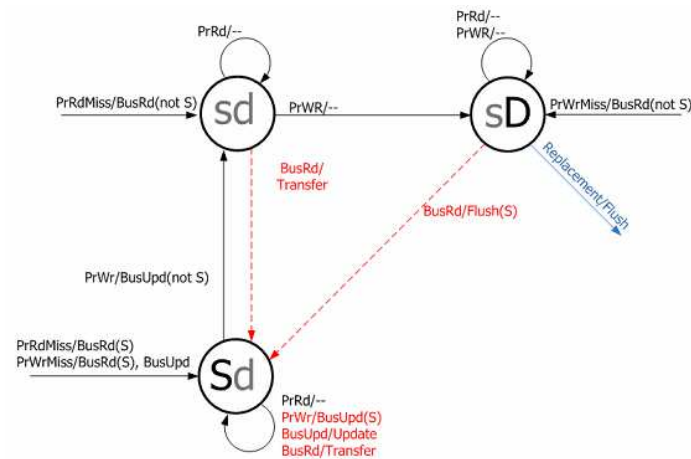
Zadatak 5 - Dragon protokol

Prikazati rad Dragon protokola na primeru date sekvence.



Zadatak 6 - Firefly protokol

Prikazati rad Dragon protokola na primeru date sekvence.



Flush – ažurira blok u memoriji i eventualno dostavlja procesoru koji je zahtevao podatak
BusUpd – ažurira samo promenjeni podatak u svim kopijama i u memoriji
Transfer – prenos celog bloka drugom procesoru iz keširane kopije

Napomena: magistrala mora biti realizovana tako da korektno radi ukoliko više procesora sa kopijom u stanju Sd uradi operaciju **Transfer**. Ukoliko to tehnološki nije moguće, mora se raditi čitanje iz glavne memorije, a ne iz drugog keša.