Multiprocesorski sistemi

Osluškujući protokoli za koherenciju keš memorije

Andrija Bošnjaković

MS1MPS, RI5MPS, IR4MPS, SI4MPS

2008/2009.

Koherencija keš memorije

- Mogućnost da keš memorija procesora sadrži podatak koji nije ažuran se često sreće
- Uslov je da još neki uređaj ima pristup magistrali i da može da menja podatke u memoriji
  - DMA kontroler
  - Drugi procesor(i)
- Potrebno je da podaci uvek budu ažurni, i zato se koriste protokoli za koherenciju keš memorije

Protokoli za koherenciju keš memorije

- Obavezni u multiprocesorskim sistemima sa deljenom memorijom
- Tri osnovne vrste protokola
  - Svaki procesor prati aktivnosti na magistrali (engl. *snooping coherence*)
  - Centralizovani direktorijum vodi evidenciju o tome koji keš šta tačno sadrži (engl. *directory-based coherence*)
  - Prevodilac postavlja posebne instrukcije za koherenciju

    (engl. *compiler-supported coherence*)

Osluškujući protokoli (engl. *snooping*)

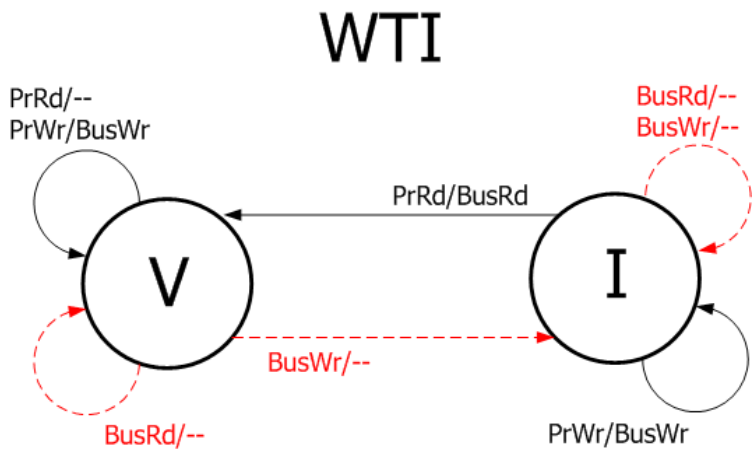
- Obavezno je da svi procesori imaju pristup istoj magistrali, koja mora biti dovoljno brza
- Povećavaju saobraćaj na magistrali, pa se zato sreću u sistemima sa deljenom memorijom sa manjim (jednocifrenim) brojem procesora
- Tri grupe
  - Poništavajući (engl. *invalidating*)
  - Ažurirajući (engl. *updating*)
  - Prilagodljivi (engl. *adaptive*)
    - Koriste jedno od dva iznad navedena ponašanja zavisno od dotadašnjeg ponašanja sistema
    - Neće biti obrađeni na vežbama

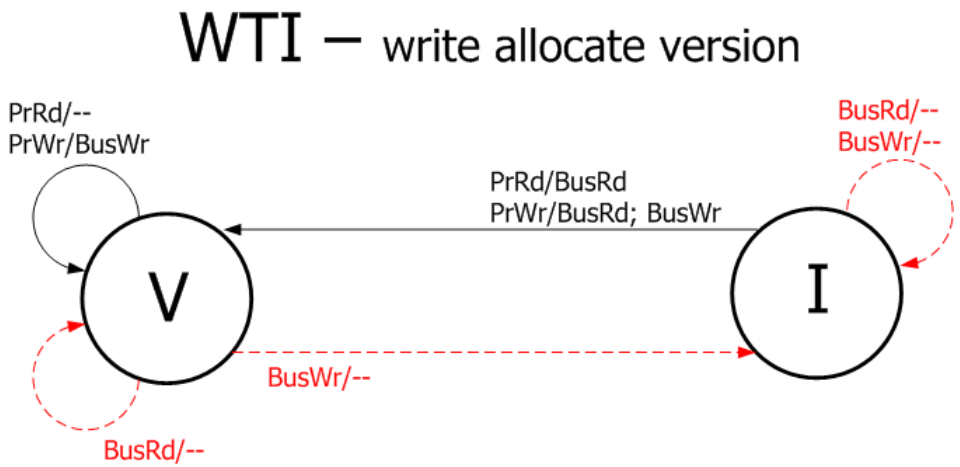
Poništavajući protokoli

- Svaki upis u neki podatak povlači **poništavanje** ostalih kopija tog podatka (ako postoje) u keš memorijama ostalih procesora
- Manji saobraćaj na magistrali nego kod ažurirajućih protokola (nema slanja podatka preko magistrale podataka)
- Pogodni za situacije gde više procesora retko pristupa istom podatku u bliskim trenucima
- U upotrebi u savremenim CMP (npr. Intel Xeon, AMD Opteron)

WTI (Write Through-Invalidate)

- Dve varijante, prema strategiji nakon *Write Miss*
  - *Write-Allocate* dovlači ceo blok iz operativne u keš memoriju, odgovarajuća linija keš memorije prelazi u stanje V (*Valid*)
  - *No-Write-Allocate* menja traženi podatak direktno u memoriji, odgovarajuća linija u keš memoriji ostaje u stanju I (*Invalid*)





WTI 1

- Dat je multiprocesorski sistem sa 4 procesora, koji koristi WTI za održavanje koherencije keš memorije. Data je sledeća sekvenca pristupa memoriji:
  - P0,R,A0
  - P1,R,A1
  - P0,R,A0
  - P2,R,A0
  - P3,R,A0
- Koliko puta koji od procesora pristupa memoriji?
- Koliki je Hit Rate za svaki od procesora?

WTI 2

- Dat je multiprocesorski sistem sa 4 procesora, koji koristi WTI za održavanje koherencije keš memorije. Data je sledeća sekvenca pristupa memoriji:
  - P0,R,A0
  - P1,R,A1
  - P0,W,A0
  - P2,R,A0
  - P3,R,A0
- Koliko puta koji od procesora pristupa memoriji?
- Koliki je Hit Rate za svaki od procesora?
- Da li je Hit Rate različit od prethodnog slučaja? Zašto?
- Nacrtati stanja linije keša koja čuva podatak sa adrese A0 nakon svakog od pristupa memoriji

WTI 3

- Dat je multiprocesorski sistem sa 4 procesora, koji koristi WTI za održavanje koherencije keš memorije. Data je sledeća sekvenca pristupa memoriji:
  - P0,R,A0
  - P1,W,A1
  - P0,W,A0
  - P2,W,A0
  - P3,W,A0
- Koliko puta koji od procesora pristupa memoriji?
- Koliki je Hit Rate za svaki od procesora?
- Da li je Hit Rate različit od prethodnog slučaja? Zašto?
- Nacrtati stanja linije keša koja čuva podatak sa adrese A0 nakon svakog od pristupa memoriji

WTI 4

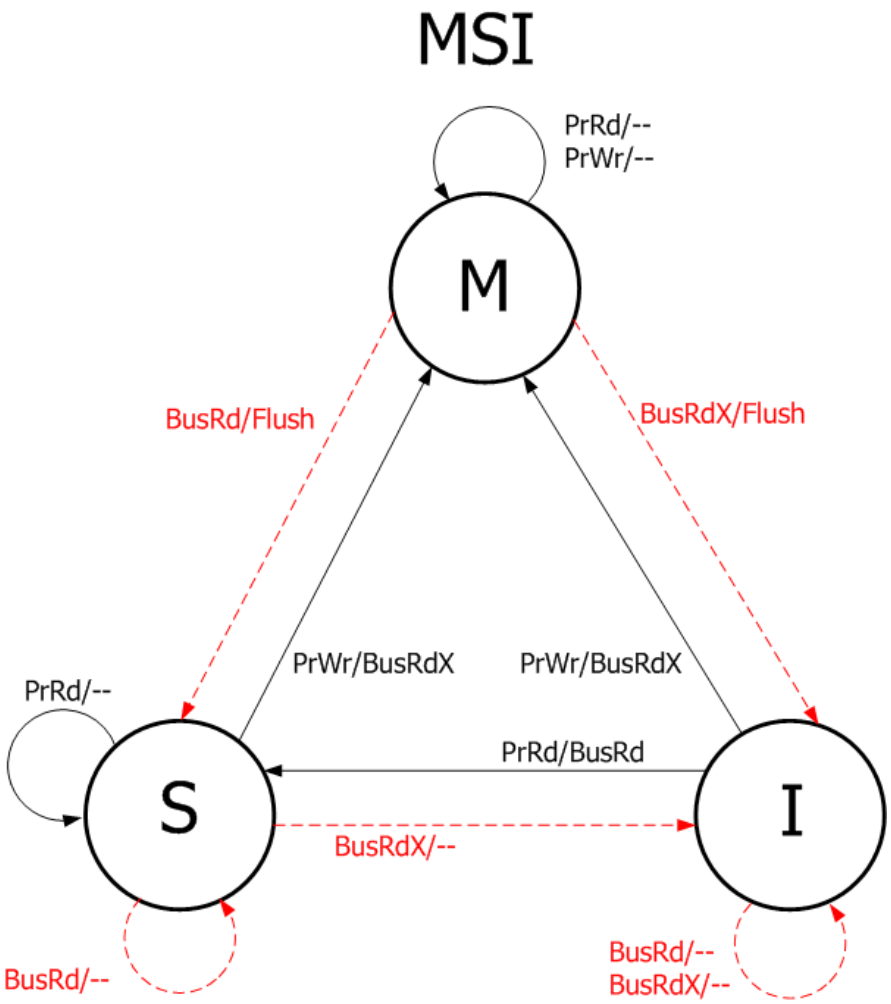
- Dat je multiprocesorski sistem sa 4 procesora, koji koristi WTI za održavanje koherencije keš memorije. Data je sledeća sekvenca pristupa memoriji:
  - P0,R,A0
  - P0,R,A0
  - P1,R,A0
  - P2,R,A0
  - P0,W,A0
  - P0,W,A0
  - P1,R,A0
  - P3,W,A0
- Koliko puta koji od procesora pristupa memoriji?
- Koliki je Hit Rate za svaki od procesora?
- Da li je Hit Rate različit od prethodnog slučaja? Zašto?
- Nacrtati stanja linije keša koja čuva podatak sa adrese A0 nakon svakog od pristupa memoriji

WTI 5

- Dat je multiprocesorski sistem sa 4 procesora, koji koristi WTI za održavanje koherencije keš memorije. Radi jednostavnosti, pretpostaviti da svaki keš ima po 2 ulaza i da su ti ulazi veličine jednog podatka. Preslikavanje je direktno – parne adrese idu u ulaz 0, neparne u ulaz 1. Data je sledeća sekvenca pristupa memoriji:
  - P0,R,A0
  - P0,R,A1
  - P1,R,A0
  - P2,R,A0
  - P0,W,A0
  - P0,W,A2
  - P1,R,A0
  - P0,W,A0
- Koliko puta koji od procesora pristupa memoriji?
- Koliki je Hit Rate za svaki od procesora?
- Da li je Hit Rate različit od prethodnog slučaja? Zašto?
- Za svaki od procesora nacrtati stanja linije keša koja čuva podatak sa adrese A0 nakon svakog od pristupa memoriji. Isto za A2.

MSI (Modified-Shared-Invalid)

- RH – bez promene, lokalno
- RM dobija blok u stanju S
  - ... čak i ako je jedina kopija !
  - M -> C ili C -> C, M
- WM ostavlja blok u M
  - BusRdX
  - Postaje vlasnik
- WH u S se tretira kao WM
  - Ignoriše poslate podatke
  - Može i BusUpgr (bez podataka)
  - Upis u M – lokalno!
- Zamena
  - Ako je u M, mora "write-back"



MSI 1

- Dat je multiprocesorski sistem sa 4 procesora, koji koristi MSI za održavanje koherencije keš memorije. Data je sledeća sekvenca pristupa memoriji:
  - P0,R,A0
  - P1,R,A1
  - P0,R,A0
  - P2,R,A0
  - P3,R,A0
- Nacrtati tabelu sa stanjima linije keša koja odgovara podatku na adresi A0 za svaki od procesora u svakom od navedenih trenutaka.

MSI 2

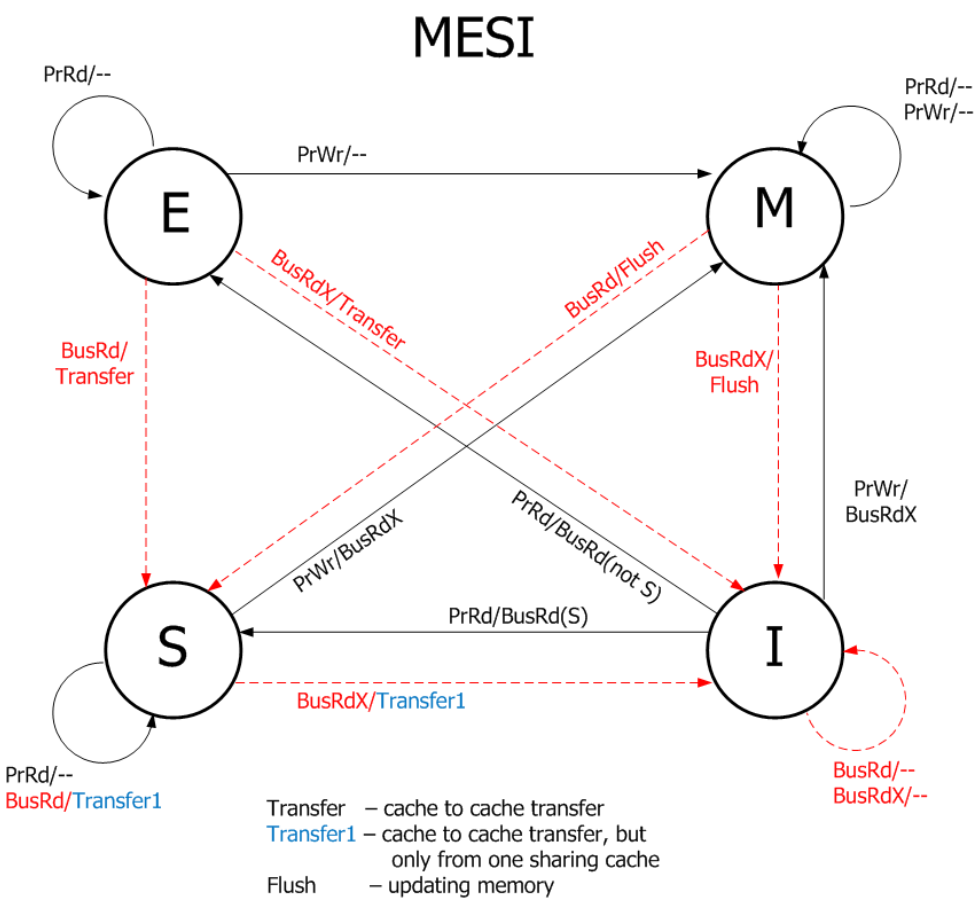
- Dat je multiprocesorski sistem sa 4 procesora, koji koristi MSI za održavanje koherencije keš memorije. Data je sledeća sekvenca pristupa memoriji:
  - P0,R,A0
  - P1,R,A1
  - P0,W,A0
  - P2,R,A0
  - P3,W,A0
- Nacrtati tabelu sa stanjima linije keša koja odgovara podatku na adresi A0 za svaki od procesora u svakom od navedenih trenutaka.

MSI 3

- Dat je multiprocesorski sistem sa 4 procesora, koji koristi MSI za održavanje koherencije keš memorije. Radi jednostavnosti, pretpostaviti da svaki keš ima po 2 ulaza i da su ti ulazi veličine jednog podatka. Preslikavanje je direktno – parne adrese idu u ulaz 0, neparne u ulaz 1. Data je sledeća sekvenca pristupa memoriji:
  - P0,R,A0
  - P0,R,A1
  - P1,R,A0
  - P2,R,A0
  - P0,W,A0
  - P0,W,A2
  - P1,R,A0
  - P0,W,A0
- Koliko puta koji od procesora pristupa memoriji?
- Koliki je Hit Rate za svaki od procesora?
- Da li se i zašto Hit Rate razlikuje od prethodnog slučaja?
- Za svaki od procesora nacrtati stanja linije keša koja čuva podatak sa adrese A0 nakon svakog od pristupa memoriji. Isto za A1.

MESI (MSI + Exclusive)

- Ima više verzija
- Pri promašaju
  - Direktan prenos C -> C
  - Ažuriranje memorije
- WH u stanju E efikasniji
- Obična zamena bloka može napraviti nelogično stanje
  - Može ostaviti u stanju S samo jednu kopiju



MESI 1

- Dat je multiprocesorski sistem sa 4 procesora, koji koristi MESI za održavanje koherencije keš memorije. Data je sledeća sekvenca pristupa memoriji:
  - P0,R,A0
  - P1,R,A1
  - P0,W,A0
  - P2,R,A0
  - P1,W,A0
- Nacrtati tabelu sa stanjima linije keša koja odgovara podatku na adresi A0 za svaki od procesora u svakom od navedenih trenutaka.

MESI 2

- Dat je multiprocesorski sistem sa 4 procesora, koji koristi MESI za održavanje koherencije keš memorije. Data je sledeća sekvenca pristupa memoriji:
  - P0,R,A0
  - P1,W,A1
  - P0,W,A0
  - P2,W,A0
  - P1,W,A0
- Nacrtati tabelu sa stanjima linije keša koja odgovara podatku na adresi A0 za svaki od procesora u svakom od navedenih trenutaka.

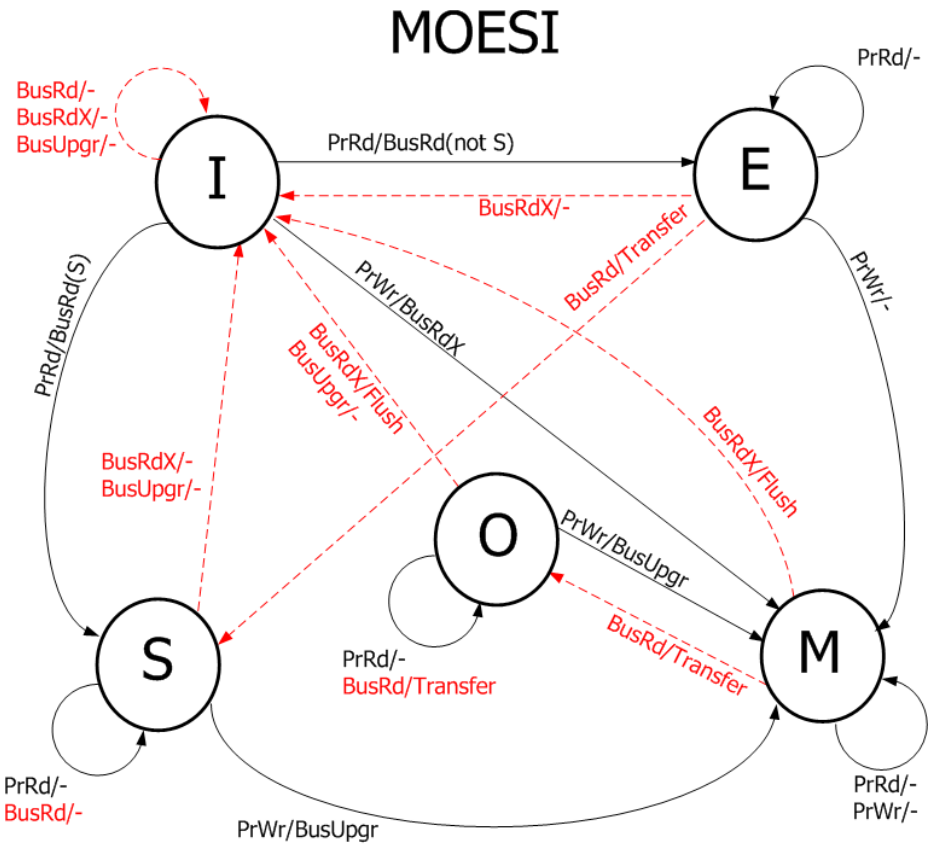
MESI 3

- Dat je multiprocesorski sistem sa 2 procesora, koji koristi MESI za održavanje koherencije keš memorije. Radi jednostavnosti, pretpostaviti da svaki keš ima po 2 ulaza i da su ti ulazi veličine jednog podatka. Preslikavanje je direktno – parne adrese idu u ulaz 0, neparne u ulaz 1. Data je sledeća sekvenca pristupa memoriji:
  - P0,R,A0
  - P0,R,A0
  - P0,W,A0
  - P0,W,A0
  - P1,R,A0
  - P1,W,A0
  - P1,W,A0
  - P0,W,A0
  - P0,W,A2
  - P0,W,A0
- Koliko puta koji od procesora pristupa memoriji?
- Koliki je Hit Rate za svaki od procesora?
- Za svaki od procesora nacrtati stanja linije keša koja čuva podatak sa adrese A0 nakon svakog od pristupa memoriji.

MESI 4

- Dat je multiprocesorski sistem sa 2 procesora, koji koristi MESI za održavanje koherencije keš memorije. Radi jednostavnosti, pretpostaviti da svaki keš ima po 2 ulaza i da su ti ulazi veličine jednog podatka. Preslikavanje je direktno – parne adrese idu u ulaz 0, neparne u ulaz 1. Data je sledeća sekvenca pristupa memoriji:
  - P0,R,A0
  - P0,R,A2
  - P0,W,A2
  - P0,W,A2
  - P1,R,A2
  - P1,W,A2
  - P1,W,A2
  - P1,R,A0
  - P0,W,A2
  - P0,W,A2
- Koliko puta koji od procesora pristupa memoriji?
- Koliki je Hit Rate za svaki od procesora?
- Za svaki od procesora nacrtati stanja linije keša koja čuva podatak sa adrese A0 nakon svakog od pristupa memoriji.

MOESI (MESI + Owned)

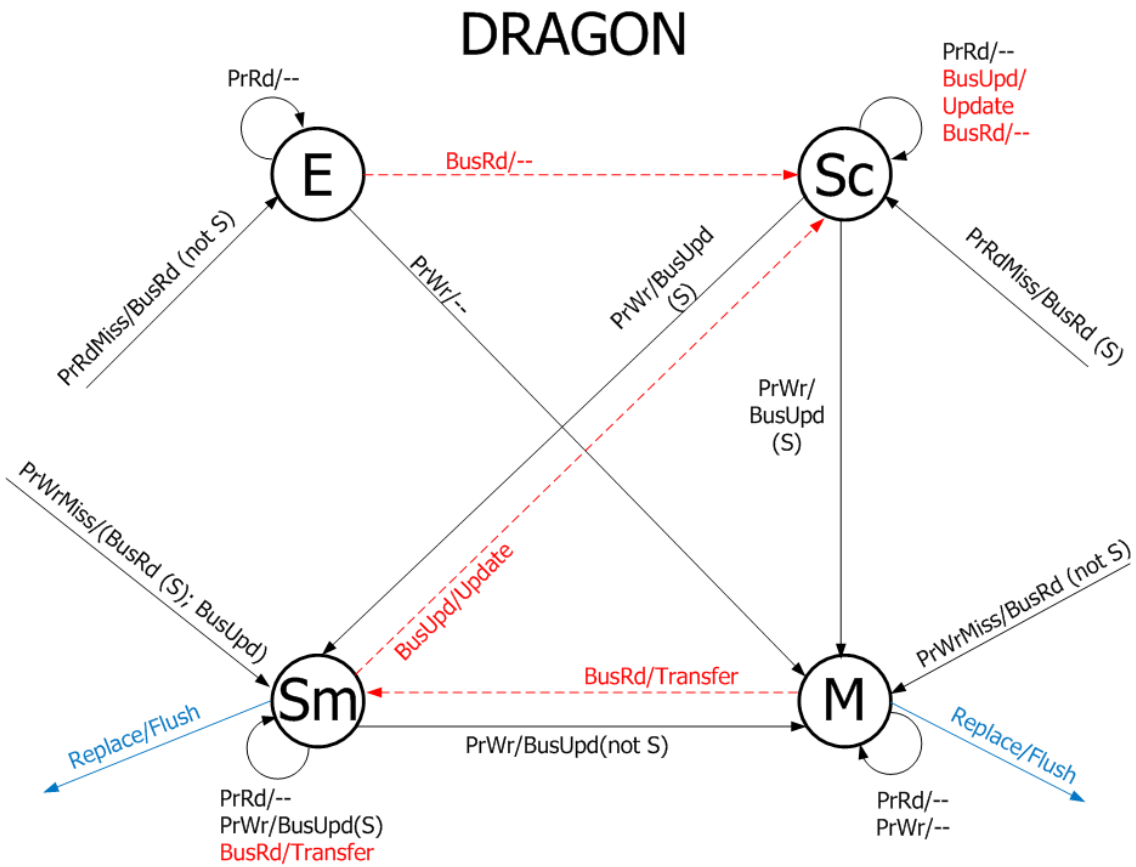


BusUpgr – Invalidira ostale kopije
Transfer – prosleđuje blok drugom procesoru, bez ažuriranja memorije
Flush – ažurira memoriju i (opciono) prosleđuje blok drugom procesoru

Ažurirajući protokoli

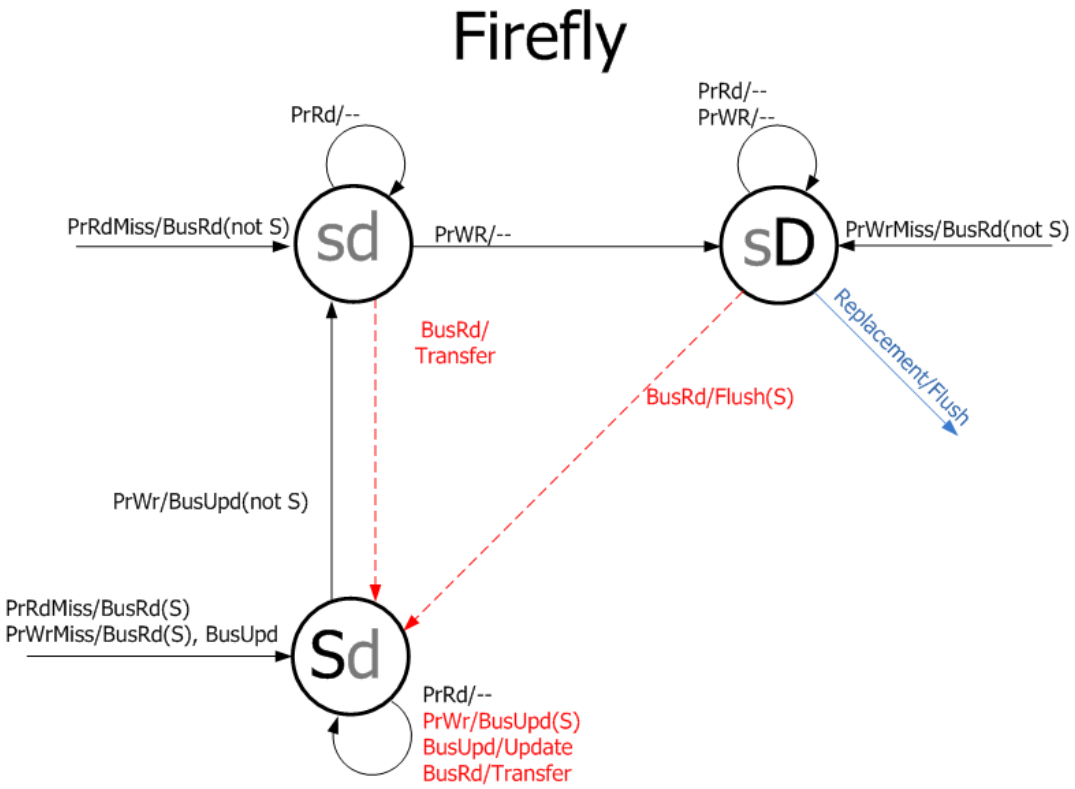
- Svaki upis u neki podatak povlači **ažuriranje** ostalih kopija tog podatka (ako postoje) u keš memorijama ostalih procesora
- Veći saobraćaj na magistrali nego kod poništavajućih protokola (zbog slanja podatka preko magistrale podataka)
- Pogodni za situacije gde više procesora često pristupa istom podatku u bliskim trenucima
- Pogodni za računare sa brzom magistralom (brzom dovoljno da isprati procesore)

Dragon



Transfer – prosleđuje blok sa podatkom drugom procesoru, bez ažuriranja memorije (može se i ažurirati, ali nije neophodno)
Flush – ažurira memoriju pri zameni (ceo blok)
BusUpd – ažurira samo promenjeni podatak u bloku

Firefly



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.

Reference

- Vivio simulatori na sajtu predmeta i njihova prateća dokumentacija na istoj stranici (http://mups.etf.rs/simulatori/vivio/)
- Slajdovi i beleške sa predavanja
- Knjiga Culler, Gupta, Singh
- Odgovarajući Wikipedia članci