

Quinta Lista de Exercícios – Multiprocessadores com Memória Compartilhada

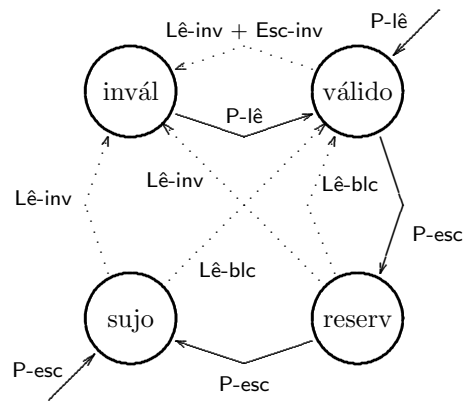
1) Considere um multiprocessador simétrico com 4 processadores interligados por um barramento. As caches são mantidas coerentes por um protocolo de invalidação, cuja máquina de estados está no diagrama abaixo. Os blocos das caches são de 8 palavras de 32 bits. Qual o estado final das caches, supondo o estado inicial mostrado abaixo, e a execução dos seguintes comandos pelos três processadores. Os comandos estão mostrados na ordem absoluta de tempo; os índices nos blocos das caches são irrelevantes; e as caches são infinitas.

```
P0: EXCL | v[0] .. v[7]
```

```
P1: SHAR | u[0] .. u[7]
```

```
P2: SHAR | u[0] .. u[7]
```

O protocolo usa escrita forçada na primeira escrita em um bloco; após a primeira escrita, o protocolo usa escrita preguiçosa. Quando um bloco **sujo** é substituído na cache, todo o bloco deve ser copiado para a linha em memória. As linhas contínuas denotam operações iniciadas pelo processador; linhas tracejadas denotam ações remotas iniciadas pelos controladores das outras caches e difundidas no barramento. O diagrama de estados mostra as transições que ocorrem em função dos comandos de coerência gerados pelo processador local (P-*) ou pelos controladores de cache remotos (Le-* ou Esc-*). Este protocolo usa os estados **reservado** e **sujo** para reduzir o tráfego causado por escritas.



```
P0
...
...
...
...
v[6] = 0;
...
a=0;
for(j=0; j<7; j++)
  a = a + t[j] + v[j+4];
```

```
P1
...
...
...
v[6] = 0;
...
...
...
...
...
...
...
...
...
...
...
...
```

```
P2
for(i=0; i<7; i++)
  t[i] = u[i] * v[i];
...
...
...
...
...
...
...
...
...
...
...
...
```

2) Considere um sistema com 2 processadores (P1,P2) com caches privadas *write-back* e memória compartilhada, todos interligados por um barramento. Cada cache possui 4 blocos, numerados 0..3. A memória compartilhada é dividida em 8 blocos, 0..7. Para manter coerência entre as caches, o sistema usa um protocolo de espionagem com 3 estados (RO,RW,inv) e política de invalidação nas escritas.

O mesmo relógio controla os processadores e o barramento. Em cada ciclo, qualquer processador pode tentar um acesso ao barramento. No caso de acessos simultâneos pelos dois processadores, a requisição de P1 é atendida e P2 deve esperar um ou mais ciclos para obter acesso ao barramento.

Em qualquer caso, o barramento suporta somente uma transação por ciclo. Uma vez que seja obtido o acesso acesso, a transação deve ser completada antes que a próxima requisição seja atendida. Quando não há contenção, acessos à memória dos processadores podem custar de um a dois ciclos, conforme listado abaixo. No caso de contenção, um ciclo adicional é necessário para arbitrar pelo barramento, em todos os casos, exceto num acerto na leitura.

- ▷ Acerto na leitura custa 1 ciclo, e não usa o barramento;
▷ falta na leitura custa 2 ciclos: 1 para buscar o bloco e 1 para a CPU ler da cache;
▷ acerto na escrita custa 1 ciclo para a CPU escrever e invalidar no barramento, simultaneamente;
▷ falta na escrita custa 2 ciclos: 1 para buscar o bloco e invalidar, mais 1 para a CPU escrever;
▷ substituição de um bloco sujo requer 1 ciclo para atualizar a memória através do barramento.

Considere as duas seqüências assíncronas de referências à memória, com números mostrados em negrito são escritas.

```
P1: 0,0,0,1,1,4,3,3,5,5,5
P2: 2,2,0,0,7,5,5,5,7,7,0
```

Compute um traçado da execução destas duas seqüências nos dois processadores, ao acessar blocos sucessivos. Ambas as caches estão inicialmente vazias. Indique o estado RO,RW dos blocos, marque as faltas, e calcule a utilização do barramento (ocupado/livre) a cada ciclo. Suponha que a primeira referência dos dois processadores inicia simultaneamente no ciclo 1. (a) Suponha caches com mapeamento direto; (b) suponha caches com associatividade binária e substituição LRU.