

TD 4

TD cohérence

Ce sujet de TD repose sur la connaissance du protocole MSI (cf support de cours).

4.1 Comportement de MSI

Nous allons supposer l'architecture suivante :

- 2 coeurs d'exécution complets reliés par un bus à une mémoire centralisée
- Chaque coeur possède un cache de donnée. Les caractéristique de ces deux caches sont identiques
 - Le cache fonctionne en direct map.
 - Le cache contient 1024 entrées de 64 octets chacune.
 - L'adressage se fait octet par octet.
 - En supposant les bits d'adresse numéroté de gauche à droite de b32 à b1, l'étiquette sera donc constituée des bits b32-b17, et l'index du bloc sera déterminé par les bits b16-b7, et le décalage au sein du bloc b6-b1.
 - le cache a une politique de réécriture différée (write back). Lorsque le contenu d'un bloc est modifié on attend que le bloc soit évincé pour diffuser la mise à jour vers la mémoire.
- La cohérence entre cache et mémoire est assurée en appliquant le protocole MSI décrit en cours.
- La diffusion d'une mise à jour suite à une requête BusL4R ou BusL4W permet au cache de mettre à jour le bloc requis.

Concernant la description du protocole, nous reprenons les notations vues en cours concernant les événements internes aux coeurs et les événements transitants sur le bus.

Pr<XY> décrit l'un des événements suivants read hit (XY=RH), read miss (XY=RM), write hit (XY=WH), write miss (XY=WM).

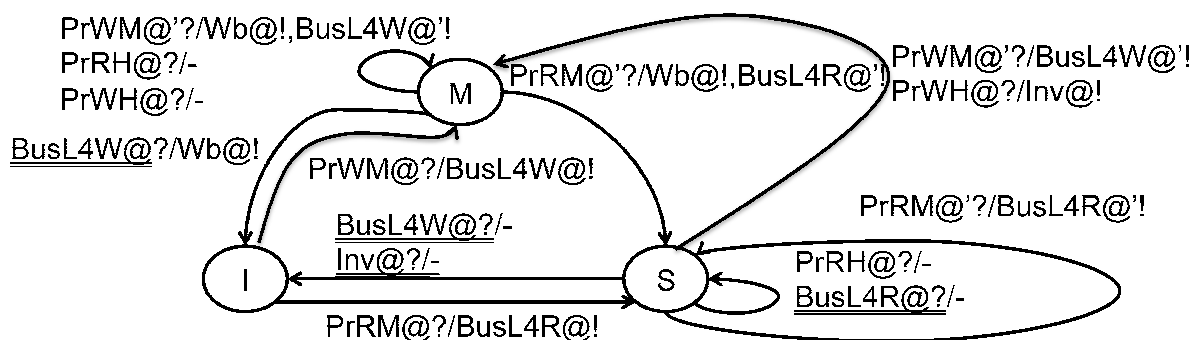


Figure 4.1: Protocole MSI : principe

BusL4Y décrit une requête de lecture de bloc émis depuis un cache à destination de la mémoire. Si $Y=R$, cette lecture permettra de réaliser un accès en lecture sur le cache. Si $Y=W$, alors cette lecture a pour finalité une écriture sur le bloc remonté.

Wb diffusion d'un bloc modifié à la mémoire et aux autres caches

Inv diffusion d'une demande d'invalidation d'un bloc.

Chacun de ces événements sera suivi de l'adresse concernée lorsque l'on voudra décrire l'événement complet.

La figure 4.1 donne la description du protocole fournie en cours.

Dans la suite nous noterons les coeurs respectivement P_0 et P_1 .

Exercice [No. 7] (Le protocole MSI)

Nous supposons qu'il y a une variable x sur 4 octets placée à l'adresse 4096. Le premier processeur exécute le code correspondant aux actions suivantes :

1. charger x dans $r1$
2. tester si $r1$ est nul, si oui sauter à 6.
3. décrémenter $r1$ de 1
4. mémoriser $r1$ dans x
5. revenir à 1.
6. fin

Le second processeur exécute le code correspondant aux actions suivantes :

1. charger x dans $r1$
2. tester si $r1$ est supérieur à 100, si oui sauter à 6.
3. incrémenter $r1$ de 1
4. mémoriser $r1$ dans x
5. revenir à 1
6. fin

On remarquera qu'aucun verrou n'est utilisé ici. Cet exemple a donc simplement pour but de vous faire manipuler le protocole MSI. Nous supposons que x vaut 50 au départ.

Question 1. Indiquer l'état de partage du bloc correspondant à x pour les deux processeurs après exécution de 1.2.3. sur P_0 et P_1

Question 2. Décrivez la séquence de messages circulant sur le bus (pour la cohérence du cache) si l'on exécute à partir de l'état précédent 4. sur P_0 puis 4. sur P_1 puis 5. et 1. sur P_0

4.2 Cohérence et faux partage

Nous allons aborder dans cette section un inconvénient connu des protocoles de cohérence de cache. Supposons que le second programme décrit auparavant modifie la variable y (au lieu de x). Le premier processeur exécute le code correspondant aux actions suivantes :

1. charger x dans $r1$
2. tester si $r1$ est nul, si oui sauter à 6.
3. décrémenter $r1$ de 1

4. mémoriser r1 dans x
5. revenir à 1.
6. fin

Le second processeur exécute le code correspond aux actions suivantes :

1. charger y dans r1
2. tester si r1 est supérieur à 100, si oui sauter à 6.
3. incrémenter r1 de 1
4. mémoriser r1 dans y
5. revenir à 1
6. fin

Les deux variables sont supposées être initialisées à 50. Nous supposons dans un premier temps que x et y sont d'adresses respectives (base 10) 4096 et 5096.

- Question 1. L'affirmation «il existe un entrelacement de ces programmes où l'activité de P_0 a un impact négatif sur le taux de cache miss sur P_1 » est elle vraie ?
- Question 2. Supposons maintenant que les adresses de x et y sont inconnues. Sous quelle condition l'affirmation précédente est vraie ?
- Question 3. Quel pourrait être l'impact de P_0 sur P_1 dans le pire cas ?
- Question 4. Identifiez un autre inconvénient de l'utilisation de bloc de grande taille.