

TD n° 3 : CACHES

1. Etiquettes et index de cache.

Un processeur a 2 Go de mémoire principale.

Pour les différents caches ci-dessous, on demande

- a) Quelle est la décomposition d'une adresse mémoire (figure 1) ? Donner le nombre de bits pour les parties étiquettes, index et adresse dans la ligne.
- b) Donner les différentes parties d'une ligne (bloc) de cache (figure 2). Combien y a-t-il de bits pour le contrôle, l'étiquette et la partie donnée ? Quel est le nombre total de bits du cache ? Par rapport à la partie « données » du cache, quel est le surcoût lié aux bits de contrôle et d'étiquette ?

Etiquette	Index	Adresse dans bloc
-----------	-------	-------------------

Figure 1 : décomposition d'une adresse mémoire

Etiquette	ctl	Instructions ou données
-----------	-----	-------------------------

Figure 2 : ligne de cache

A) Cache de 2 Mo à correspondance directe et écriture simultanée avec des lignes de 16 octets

B) Cache de 4 Mo à correspondance directe, réécriture et lignes de 32 octets.

C) Cache de 4 Mo associatif 4 voies (4 lignes par ensemble), réécriture et lignes de 32 octets.

2. Caches données

On considère une architecture possédant un cache de données de 8K octets organisé en lignes de 32 octets. Les exercices suivants seront traités dans 2 cas : correspondance directe et associativité par ensembles de 2 lignes, avec pseudo-LRU. On considère des tableaux de 4096 flottants simple précision (32 bits), implantés aux adresses suivantes :

X	Y	Z	X1	Y1	X2	Y2
1 0000 _H	1 4000 _H	1 8000 _H	1 C000 _H	1 E000 _H	2 0000 _H	2 4000 _H

a) Quels sont les éléments des tableaux X et Y qui peuvent occuper le mot 0 de la ligne 0 du cache ?

b) Combien de défauts de cache de données par itération interviennent dans chacune des boucles suivantes, où on suppose que les variables scalaires sont toujours en registre :

b1 for (i=0; i<N; i++) S += X[i]*Y[i];	b2 for (i=0; i<N; i++){ S1 += X1[i]*Y1[i]; S2 += X2[i]*Y2[i]; }	b3 for (i=0; i<N; i++) S1. += X1[i]*Y1[i]; for (i=0; i<N; i++) S2 += X2[i]*Y2[i];	b4 for (i=0; i<N; i++) { S1 += X[i]*Y[i]; S2 += X[i]*Z[i]; }
---	--	--	---

3. Caches instructions

Un processeur a un jeu d'instructions RISC, avec des instructions de longueur fixe d'un mot. Il a un cache instructions de 2 Kmots, avec des lignes de 8 mots. Il utilise la correspondance directe.

Il exécute le programme Figure 3, constitué de deux boucles imbriquées. Les seuls branchements du programme sont les deux branchements de boucle, aux adresses 239 et 1200.

Le temps pour un succès cache est T et un défaut de cache coûte 8T.

a) En négligeant l'effet des défauts de caches pour les données, quel est le temps d'exécution du programme de la Figure 3.

b) Reprendre la question précédente en supposant un cache de 1 Kmots avec correspondance directe, puis l'associativité 2 voies (2 lignes par ensemble)

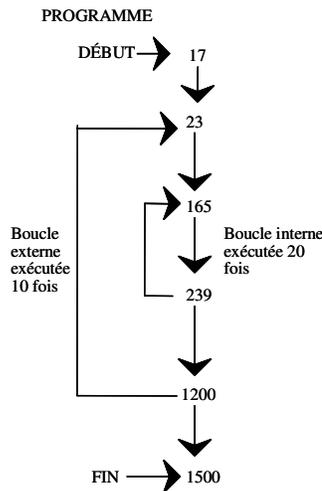


Figure 3 : programme considéré

4. Protocoles de cohérences des caches (MESI)

Le protocole MESI (Illinois) est un protocole avec réécriture et invalidation qui a quatre états : modifié (M), exclusif (E), partagé (S) et invalide (I). Son diagramme de transitions est donné en figure 4.

Exercice 1

Soit un multiprocesseur avec trois processeurs, avec la suite d'actions suivantes :

- P1 lit u
- P3 lit u
- P3 écrit dans u
- P1 lit u
- P2 lit u

	Action processeur	Etat P1	Etat P2	Etat P3	Action Bus	Donnée fournie par
1						
2						
2						
4						
5						

Exercice 2

Soit un multiprocesseur à 3 processeurs. Les caches sont initialement vides. Un succès cache lecture ou écriture prend 1 cycle. Les échecs nécessitant une transaction simple sur le bus (BusUpgr, BusUpd) prennent 60 cycles et les échecs nécessitant le transfert complet d'un bloc prennent 90 cycles. Tous les caches sont à allocation d'écriture.

Soient trois extraits différents de code qui accèdent à la même case mémoire en lecture (r) ou écriture (w).

Code 1 : r1 w1 r1 w1 r2 w2 r2 w2 r3 w3 r3 w3

Code 2 : r1 r2 r3 w1 w2 w3 r1 r2 r3 w3 r1

Code 3 : r1 r2 r3 r3 w1 w1 w1 w1 w2 w3

Donner les performances pour le protocole MESI.

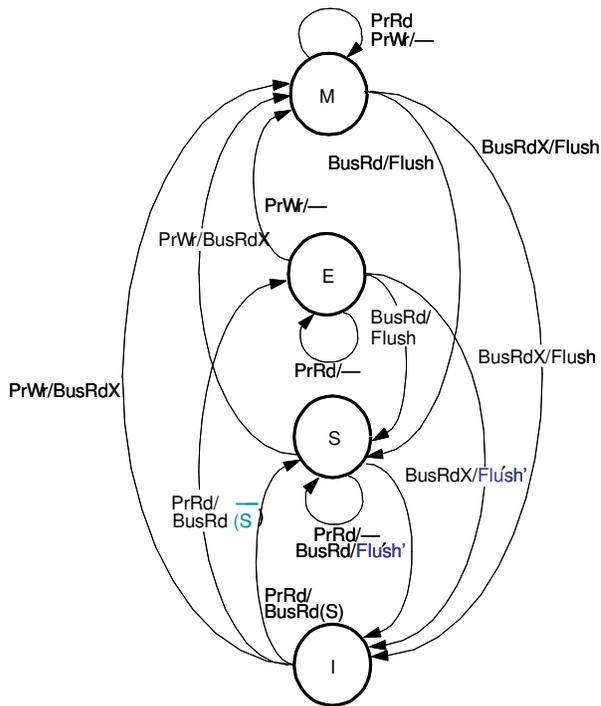


Figure 4 : protocole MESI