
Master SESI
Systèmes Multi-processeurs
Cours du professeur Alain Greiner
Examen de juin 2010

EXERCICE A : Caches de 1er niveau (10 points)

Les réponses aux questions de cours A1 à A5 n'ont pas besoin d'être longues, mais elles doivent être précises. Le but général de l'exercice qui suit est de mesurer le nombre de cycles nécessaires à l'exécution d'un programme C qui réalise le produit scalaire de deux vecteurs **A** et **B**, en tenant compte des effets de cache:

```
int A[32] = { 1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12,13,14,15,16,
            17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32};
int B[32] = {51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66
            67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82};

int main(){
    register int i, sum = 0;
    for (i=0; i<32; i++)
    {
        sum = sum + A[i] * B[i];
    }
    return 0;
}
```

On considère un processeur MIPS32 possédant un cache de données et un cache instruction ayant les mêmes caractéristiques: deux niveaux d'associativité, capacité totale de 256 octets. La ligne de cache a une largeur de 32 octets (8 mots de 32 bits).

Les données globales du programme sont stockées de façon contiguë dans le segment `seg_data` dont l'adresse de base est `0x10000000`.

Le mot clé "register" est une directive passée au compilateur pour qu'il place les variables `i` et `sum` dans des registres plutôt que sur la pile du programme. Les variables `i` et `sum` sont dans des registres durant toute la durée d'exécution du programme et ni leur lecture ni leur écriture ne provoquent d'accès au cache de données.

A1) Rappeler la différence entre les trois types de cache : cache à correspondance directe, cache associatif par ensembles, cache totalement associatif. Quels sont les avantages de l'associativité? Quels sont les inconvénients?

A2) On considère que les caches sont bloquants, ce qui signifie que le processeur se bloque en cas de MISS. Rappelez la définition du taux de MISS et du coût du MISS. Ces deux valeurs sont-elles des caractéristiques intrinsèques de l'architecture matérielle, ou dépendent-elles de l'application logicielle exécutée?

A3) Rappeler la définition de la stratégie d'écriture "Write-Through" et de la stratégie "Write-Back".

A4) Expliquer le principe du mécanisme d'espionnage du bus ("Snoop") pour garantir la cohérence entre les caches et la mémoire dans le cas d'une architecture multi-processeurs. Préciser ce que doit faire le mécanisme de "Snoop" dans le cas d'une stratégie "Write-Through" et dans le cas d'une stratégie "Write-Back".

A5) Pourquoi les caches de premier niveau fournissent-ils généralement une fonctionnalité permettant que certains segments de l'espace adressable soient non cachables?

A6) Donner le nombre de cases des caches définis ci-dessus (une case permet de stocker une ligne), ainsi que le nombre de bits des champs offset, index, et étiquette de l'adresse.

Après optimisation, la compilation du corps de la boucle du programme C proposé donne le code assembleur suivant pour le processeur MIPS32 (on ne considère pas la déclaration des variables ni leur initialisation):

```
loop:    lw    $4, 0($2)
        lw    $5, 128($2)
        mul   $6, $4, $5
        addi  $3, $3, -1
        addi  $2, $2, 4
        bne  $3, $0, loop
        nop
```

- Le registre \$2 contient l'adresse de base du segment seg_data.
- Le registre \$3 contient la valeur 32.
- Le label loop correspond à l'adresse 0x400010.

On considérera que les deux caches sont vides lorsqu'on entre dans la boucle.

A7) Représentez l'état du cache instruction après le premier tour de boucle. On construira un tableau contenant pour chaque case les informations suivantes:

- Index (numéro de l'ensemble associatif),
- Way (numéro de voie dans un ensemble associatif)
- Valid, (contenu de la case valide)
- Adresse, (adresse hexadécimale complète sur 32 bits) du mot 0 de la ligne de cache)
- Data[i] (où i est le numéro du mot dans la ligne de cache)

Donnez le nombre de MISS sur le cache instruction pour les 32 itérations de la boucle.

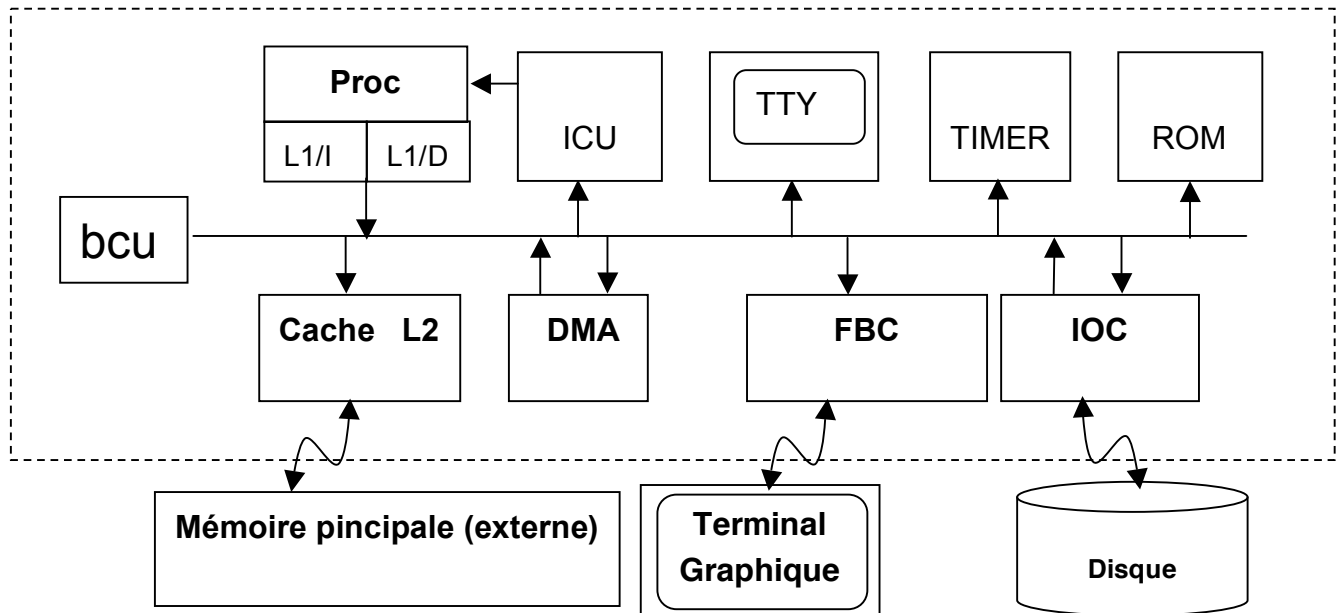
A8) représentez l'état du cache de données après le premier tour de boucle.

Donnez le nombre total de MISS sur le cache de données pour les 32 itérations.

A9) On suppose que le coût d'un MISS (instruction ou donnée) est de 15 cycles. En déduire le nombre de cycle total pour exécuter les 32 itérations de la boucle sur un processeur MIPS32 à architecture pipe-line. Quel est la valeur du CPI ?

EXERCICE B : Bus système (10 points)

On considère l'architecture matérielle ci-dessous, construite autour du PIBUS. Cette architecture contient un seul processeur (avec ses caches de 1^{er} niveau), un timer, un terminal écran/clavier de type TTY, un contrôleur graphique (« frame buffer ») permettant d'afficher des images, une ROM contenant le « code de boot », un cache de 2^e niveau permettant d'accéder à la mémoire externe, et deux périphériques possédant une capacité d'adressage de la mémoire : le contrôleur DMA permet de transférer des données d'un tampon mémoire vers un autre. Le contrôleur IOC permet de transférer des données entre le disque et un tampon mémoire (dans un sens ou dans l'autre).



On suppose que des images sont stockées sur le disque, et qu'on cherche à afficher une séquence d'image en respectant la cadence vidéo. La fréquence vidéo est de 25 images par seconde (une nouvelle image doit être affichée toutes les 40 ms). Comme vous l'avez vu en TP, l'affichage d'une image nécessite trois étapes :

- **phase Load** : chargement de l'image depuis le disque vers un premier tampon mémoire appelé `buf_in`. Ce transfert est réalisé par le contrôleur IOC.
- **phase Modif** : le processeur lit l'image stockée dans `buf_in`, la modifie, et recopie l'image modifiée dans un second tampon mémoire `buf_out`.
- **phase Display** : affichage de l'image par copie du tampon `buf_out` vers le Frame Buffer (composant FBC). Ces transferts (lecture puis écriture) sont réalisés par le contrôleur DMA.

Le but général de l'exercice est de déterminer la taille maximale d'une image, en faisant l'hypothèse que le facteur limitant est la bande passante du bus (mesurée en nombre d'octets par cycle). On suppose que l'image est codée en « niveaux de gris », et que chaque pixel est codé sur 8 bits. On notera N le nombre total de pixels d'une image : Par exemple, une image de 400 lignes de 600 pixels a une taille de $N = 240\,000$ pixels.

On suppose que cette architecture est celle d'un système embarqué (tel qu'un téléphone mobile), qui fonctionne à 25MHz. Cette fréquence est assez basse, pour limiter la consommation d'énergie, et augmenter la durée de vie de la batterie.

On rappelle que la largeur du PIBUS est de 32 bits, ce qui signifie qu'on peut transférer au plus un mot de 32 bits par cycle.

-
- B1)** Qu'est-ce qu'un composant maître ? Combien y a-t-il de composants maîtres sur ce bus, et quels sont-ils ? Combien y a-t-il de composants cibles, et quels sont-ils ? Rappelez la définition d'une transaction. Comment est désignée la cible d'une transaction ?
- B2)** Comment un programme utilisateur déclenche-t-il un transfert de données entre le disque et un tampon mémoire ? Quel est le composant matériel qui effectue ce transfert ? Quels sont les trois arguments de l'appel système permettant de déclencher un transfert de données dans le sens disque vers mémoire? Pourquoi le programme utilisateur doit-il utiliser un appel système ? Que fait le code de cet appel système avant de rendre la main au programme utilisateur?
- B3)** Pourquoi le contrôleur de disque doit-il signaler la fin d'un transfert qui lui a été demandé? Comment le contrôleur signale-t-il la fin du transfert ? Décrivez précisément les différentes étapes permettant au programme utilisateur qui a demandé le transfert d'être informé de la fin du transfert. Pourquoi cette signalisation de fin de transfert fait-elle intervenir un autre appel système ?
- B4)** En supposant que chaque transaction effectuée par le contrôleur IOC correspond au transfert d'un bloc de 512 octets, quelle est la durée totale d'occupation du bus pour transférer une image de N lignes de M pixels du disque vers la mémoire ? On rappelle que la durée d'occupation du bus est le nombre de cycles pendant lequel le bus ne peut pas être utilisé pour une autre transaction.
- B5)** Le contrôleur DMA est utilisé pour transférer l'image du tampon buf_out vers le frame buffer. Quels sont les registres adressables du contrôleur DMA? Quelles actions doit effectuer le logiciel pour lancer un transfert de données de K octets entre un tampon source et un tampon destination?
- B6)** En supposant que la mémoire privée du contrôleur DMA (qui définit la longueur de la rafale) a une capacité de 16 mots de 32 bits, combien de transactions sur le bus sont nécessaires pour déplacer une image de N * M pixels ? Quel est la durée totale d'occupation du bus pour déplacer une image complète?
- B7)** Pour effectuer le filtrage/modification de l'image, le processeur doit lire chaque pixel dans le tampon buf_in et doit re-écrire le pixel modifié dans le tampon buf-out. En faisant l'hypothèse d'une ligne de cache de 32 octets, combien de transactions sont nécessaires pour lire les M * N pixels d'une image ? Quelle est la durée d'occupation du bus liée à la lecture de l'image dans le tampon buf_in?
- B8)** En faisant l'hypothèse que chaque pixel écrit dans le tampon buf_out déclenche une transaction particulière sur le bus quel est la durée d'occupation du bus liée à l'écriture dans le tampon buf_out?
- B9)** De combien de cycles dispose-t-on entre deux affichages d'image si on veut respecter la cadence video?
- B10)** En rassemblant les résultats des questions précédentes, calculez une borne maximale pour la taille (M * N) de l'image.