

# Rapport de TP SoC/Cache

Binôme :

20 novembre 2016

## 1 Rappels et questions de cours

### 1.1 Question 1 :

Les mécanismes de mémoire cache se base sur le principe de localité qui dit que le code et les données des programmes ne sont pas utilisées de manière uniforme. On constate souvent que 10% du code d'un programme contribue à 90% des instructions exécutées. On distingue deux types de localité :

- Temporelle qui indique que des éléments auxquels on a eu accès récemment seront probablement utilisés dans un futur proche.
- Spatiale qui indique que des éléments proches ont tendances à être référencés à des instants proches.

### 1.2 Question 2 :

L'avantage des caches associatifs par rapport aux caches à correspondance direct est qu'il permet une grande souplesse et une efficacité pour gérer les lignes de manière optimale en terme de succès d'accès.

Et l'inconvénient de ce type de cache que l'on doit au pire parcourir toutes les lignes du cache pour savoir si la ligne cherchée s'y trouve ou pas.

### 1.3 Question 3 :

Les trois types de défaut de cache qui peuvent se produire dans un cache associatif sont :

- Défauts de première référence (compulsory misses) : À la première référence à un bloc, celui-ci doit être chargé dans le cache. Ces défaut sont en quelque sorte inévitables.
- Défauts de capacité (capacity misses) : Ces défauts sont dû au fait que le cache ne peut pas contenir tous les blocs référencés pendant

l'exécution du programme. Le nombre de ces défauts peut être réduit en augmentant la taille du cache.

- Défauts de conflit (conflict misses) : Ces défauts interviennent en plus des deux précédents types. Un bloc a pu être chargé puis enlevé du cache car d'autres blocs avec le même indice ont été chargés. Le nombre de ces défauts peut être réduit en augmentant l'associativité du cache.

#### 1.4 Question 4 :

Un processeur PPC 440 dispose d'un cache dont les caractéristiques sont les suivantes :

Taille	32 KOctet
Ligne	64 Octet
Associatif par groupe	64 voies

Le nombre de lignes d'un cache correspond simplement à sa capacité divisée par la longueur de ligne. Autrement dit il y a 512 lignes ( $32 \text{ Ko} = 32768$  et  $\frac{32768}{64} = 512$ ).

Le nombre de groupes associatifs dans le cache correspond au nombre de lignes divisé par le degré d'associativité du cache. Autrement dit il y a 8 groupes associatifs ( $\frac{512}{64} = 8$ ).

#### 1.5 Question 5 :

Les bits de poids fort forment une étiquette (tag) qui sont utilisés pour calculer le numéro du groupe.  $\text{numéro de l'ensemble} = \text{numéro de bloc} (= \text{adresse}) \bmod \text{le nombre d'ensembles dans le cache}$

## 2 Performance d'une hiérarchie mémoire

### 2.1 Contexte de l'étude

#### 2.1.1 Question 6 :

#### 2.1.2 Question 7 :

Le programme `rotation_original` effectue une rotation d'image 2D car dans la boucle de calcul de la fonction `main()` il y a :

$$x' = i \cos(\alpha) - j \sin(\alpha)$$

$$y' = i \sin(\alpha) + j \cos(\alpha)$$

Ceci sont les coordonnées  $(x', y')$  d'un point qui subit une rotation d'angle  $\alpha$  dans le plan  $(i, j)$ .

2.2 Question 8 :

### 3 Installation

3.1 Installation du simulateur de cache

3.1.1 Question 9 :

3.1.2 Question 10 :

3.1.3 Question 11 :

3.1.4 Question 12 :

### 4 Mesure de l'effet de la taille des tuiles

4.1 Effet de l'angle de rotation sur les performances

4.1.1 Question 13 :

4.1.2 Question 14 :

4.2 Influence des paramètres du cache

4.3 Question 15 :

4.4 Impact du schéma d'adressage

4.5 Question 16 :

4.6 Question 17 :

4.7 Question 18 :

### 5 Transformation de boucle

5.1 Question 19 :

5.2 Question 20 :

### 6 Hiérarchie mémoire

6.1 Question 21 :

6.2 Question 22 :

### 7 Mesures

FIGURE 1 – Alpha=0

FIGURE 2 –  $\text{Alpha}=0$

FIGURE 3 –  $\text{Alpha}=0$

FIGURE 4 –  $\text{Alpha}=0$