

## Compte-rendu du TD4 :

### 1) Introduction

*Durant ce TD, nous allons étudier l'organisation, ainsi que l'efficacité, de diverses architectures de mémoire cache.*

### 2) Cache mémoire « Direct »

#### 2.1) Premier adressage de mémoire cache :

31	..	10	9	..	5	4	..	0
TAG			INDEX			OFFSET		

D'après la formule du cours, si la plage d'offset est constituée de  $b = 5$  bits, et que la plage d'index est constituée de  $k = 5$  bits, alors la mémoire cache est constituée de  $2^k = 2^5 = 32$  lignes, chacune composée de  $\frac{2^b}{4} = \frac{2^5}{4} = 8$  mots, soit 32 octets (dans le RISC-V5, un mot correspond à 4 octets).

De plus, une mémoire cache dispose d'autant d'entrées que de lignes, soit 32 entrées.

#### 2.2) Réorganisation de l'adressage :

31	..	12	11	..	6	5	..	0
TAG			INDEX			OFFSET		

De la même manière, cet adressage définit la plage d'index à  $k = 6$ , et la plage d'offset à  $b = 6$  bits. Cela signifie que la mémoire cache est cette fois-ci constituée de  $2^k = 2^6 = 64$  lignes, chacune composée de  $\frac{2^b}{4} = \frac{2^6}{4} = 16$  mots, soit 64 octets.

De plus, une mémoire cache dispose d'autant d'entrées que de lignes, soit 64 entrées.

### 3) Performance trois niveaux de cache

Soit un CPU fonctionnant à une période horloge de 1GHz et disposant d'une mémoire cache L1, L2, L3 avec les paramètres suivants :

L1 HIT time = 1 cycle

L2 HIT time = 4 cycles

L3 HIT time = 8 cycles Miss pénalité = 15 cycles

On constate les taux suivants :

L1 HIT rate = 95%

L2 HIT rate = 92%

L3 HIT rate = 90%

3.1) Pour calculer les taux de MISS de chacun des cache, on dispose de la formule suivante :

$$MissRate_i = 1 - HitRate_i$$

avec  $HitRate_i = \frac{nb\_accès\_memoire\_cache}{nb\_accès\_memoire\_totale}$ .

D'où :

Le taux de MISS de L1 vaut  $MISS_{Rate_1} = 1 - 95\% = 5\%$

Le taux de MISS de L2 vaut  $MISS_{Rate_2} = 1 - 92\% = 8\%$

Le taux de MISS de L3 vaut  $MISS_{Rate_3} = 1 - 90\% = 10\%$

Donc le taux de Miss Global est :

$$MISS_{Global} = MissRate_1 \times MissRate_2 \times MissRate_3 = 0.05 \times 0.08 \times 0.10 = 0.0004$$

$$MISS_{Global} = 0.04\%$$

3.2) La pénalité de MISS d'un cache Li se calcule de la manière suivante :

$$MissPénalité_i = HIT_{time_{i+1}} + MissRate_{i+1} \times MissPénalité_{i+1}$$

Avec cette formule, on en déduit la pénalité de MISS du cache L2 :

$$MISS_{Pénalité_2} = 8 \text{ cycles} + 15 \text{ cycles} \times (100\% - 90\%)$$

$$MISS_{Pénalité_2} = 9.5 \text{ cycles}$$

3.3) La pénalité de MISS du cache L1 est :

$$MISS_{Pénalité_1} = 4 \text{ cycles} + 9.5 \text{ cycles} \times 8\%$$

$$MISS_{Pénalité_1} = 4.76 \text{ cycles}$$

3.4) Le temps d'accès moyen du cache L1 est le suivant :

$$\text{Temps d'accès moyen} = \text{Hit}_{\text{Time}_1} + (\text{MISS}_{\text{Rate}_1} \times \text{Miss}_{\text{Pénalité}_1})$$

$$\text{Temps d'accès moyen} = 1 + 5\% \times 4.76$$

$$\text{Temps d'accès moyen} = 1.238 \text{ cycles}$$

#### 4) Performance deux niveaux de cache

On considère un système disposant d'une mémoire cache de niveau L1 et L2, en supposant que sur 1000 accès mémoire on ait :

40 MISS sur L1

20 MISS sur L2

D'après les formules utilisées précédemment, on a :

$$MissRate_1 = 1 - \frac{960}{1000} = 0.04$$

$$MissRate_1 = 4\%$$

De la même manière :

$$MissRate_2 = 1 - \frac{20}{40} = 0.5$$

$$MissRate_2 = 50\%$$

Ainsi, le temps d'accès moyen vaut :

$$Temps\ d'accès\ moyen = Hit_{Time_1} + (MissRate_1 \times MissPénalité_1)$$

En supposant  $Hit_{Time_1} = 1$ ,  $MissRate_1 = 4\%$  et

$$MissPénalité_1 = HIT_{Time_2} + MissRate_2 \times MissPénalité_2 = 10 + 0.5 \times 100 = 60,$$

on obtient :

$$Temps\ d'accès\ moyen = 1 + 0.04 \times 60$$

$$Temps\ d'accès\ moyen = 3.4\ cycles$$

### 5) Performance cache instructions et algorithme de remplacement

5.1) Le système considéré dispose d'une mémoire cache d'instructions constituées de 64 lignes contenant chacune 4 instructions de 32 bits. La taille de cette mémoire cache est donc de :

$$\text{Taille de mémoire cache} = 64 \times 4 \times \frac{32}{8} = 1024 = 1 \text{ Ko}$$

5.2) Chacune des lignes de la mémoire est constituée de 4 mots de 4 octets (4 instructions de 32 bits), donc si  $b$  est la largeur du champ offset, elle est telle que  $\frac{2^b}{4} = 4$ .

La largeur du champ offset est donc  $b = 4 \text{ bits}$ .

De plus, si  $k$  est la largeur du champ index, alors il est tel que  $2^k = 64$  lignes, donc  $k = 6 \text{ bits}$ .

On en déduit donc la largeur du champ de tag :  $l = 32 - 4 - 6 = 22 \text{ bits}$ .