

# Compte-Rendu

## TP4 : Caractérisation et dimensionnement des caches

JIANG Hongbo  
3602103

### C) Système mémoire presque parfait

**Question C1 :** Lancez la simulation, avec des caches très grands : 256 sets, 16 mots par ligne, et 4 niveaux d'associativité soit une capacité de 64 Koctets pour chacun des deux caches. On choisit également un tampon d'écritures postées de profondeur 8 mots de 32 bits. Quel est le temps d'exécution de l'application?

-75725 cycles                      CPI=1.334  
READ%(cached)=28.40%  
READ%(uncached)=0.74%  
WRITE%=13.90%    IMISS%=0.058%    DMISS%=0.156%

**Question C2 :** Relancez la simulation, et déterminez les taux de MISS pour le cache instruction et le cache de données ainsi que la valeur du CPI. Les taux de MISS sont-ils constants au cours du temps ? Comment évoluent-ils au cours des 1000 premiers cycles? Interprétez ce comportement.

-\$./simul.x -STATS 1 -NCYCLES 1000 CPI=4.22  
IMISS%=8.02%                      DMISS%=12.82%

Les taux de MISS ne sont pas constants au cours du temps, ils diminuent au cours de cycles. Parce qu'il y a des MISS compulsifs au début et après il y a plus de HIT.

**Question C3 :** En analysant le contenu des fichiers **pibus\_mips32\_xcache.cpp** et **pibus\_mips32\_xcache.h**, expliquez comment sont calculés le CPI et les deux taux de MISS.

-Dans le fichier **pibus\_mips32\_xcache.cpp**, on a défini les variables **c\_imiss\_count** et **c\_dmiss\_count**, lors les requêtes sont validées, on fait

++ ces deux variables. Les deux taux de MISS :

$IMISS = (c\_imiss\_count / run\_cycles)$ .

$DMISS = (c\_dmiss\_count / (c\_dread\_count - c\_dunc\_count))$ . Pour calculer le CPI,  $CPI = c\_total\_cycles / run\_cycles$ .

## D) Influence de la capacité du cache instruction

#cycle	Taux de MISS	Cout du IMISS	CPI	ISSET	IWORD	IWAYS	DSET	DWORDS	DWAYS
75821	0.102%	16.5862	1.336	256	8	1	256	16	4
94253	2.249%	15.8855	1.697	64	8	1	256	16	4
150783	9.644%	15.5223	2.911	16	8	1	256	16	4
170355	12.533%	15.4109	3.376	4	8	1	256	16	4
238625	24.270%	15.0362	5.222	1	8	1	256	16	4

**Question D1 :** Relevez, pour chaque configuration, la durée d'exécution du programme, le taux de MISS sur le cache instruction, le coût du miss instruction et la valeur du CPI. Comment interprétez-vous ces résultats ?

-En diminuant la capacité de ICACHE, la durée d'exécution du programme augmente remarquablement, le taux de IMISS et CPI sont plus élevés, et le coût du IMISS devient plus petit, parce que on prend moins de temps de trouver l'adresse dans le petit ICACHE.

**Question D2 :** Pourquoi la valeur obtenue pour le coût du MISS est-elle différente de la valeur estimée dans le TP3 ? Comment expliquez-vous que le coût du miss puisse avoir une valeur non entière?

-En diminuant la capacité de ICACHE, la durée d'exécution du programme augmente remarquablement, le taux de IMISS et CPI sont plus élevés, et le coût du IMISS devient plus petit, parce que on prend moins de temps de trouver l'adresse dans le petit ICACHE.

## E) Influence de la largeur de la ligne de cache

#cycle	Taux de MISS	ISSET	IWORD	IWAYS	DSET	DWORDS	DWAYS
177661		256	1	1	256	16	4
141032		128	2	1	256	16	4
124500		64	4	1	256	16	4
118708		32	8	1	256	16	4
117885		16	16	1	256	16	4
140825		8	32	1	256	16	4

**Question E1 :** Représenter graphiquement la durée d'exécution du programme en fonction de la largeur de la ligne de cache. Quelle est la configuration la plus efficace ? Comment expliquez-vous ce résultat ?

-la configuration la plus efficace est 16×16. On peut voir que au début la durée d'exécution du programme diminue à mesure que la valeur de la ligne diminue, mais après la valeur de 16, il faut plus de temps pour les exécution car le temps d'accès augmente aussi quand la valeur de mots de chaque ligne augmente.

## F) Influence de la capacité du cache de données

**Question F1 :** Relevez, pour chaque valeur, la durée d'exécution du programme, le taux de MISS sur le cache de données, le cout du miss de données et la valeur du CPI.

#cycle	Taux de MISS	Cout du IMISS	CPI	ISSET	IWORD	IWAYS	DSET	DWORDS	DWAYS
75796	0.267%	16.2093	1.335	256	16	4	256	8	1
76332	0.452%	17.2740	1.345	256	16	4	64	8	1
101807	9.059%	17.1067	1.794	256	16	4	16	8	1
153010	23.217%	16.4671	2.696	256	16	4	4	8	1
244871	40.368%	15.9252	4.314	256	16	4	1	8	1

## G) Influence de la profondeur du tampon d'écritures postées

**Question G1 :** Comment est réalisé le tampon d'écriture postées?

-On le réalise par un FIFO.

quelle sont les informations qu'il faut stocker dans le tampon pour chaque requête d'écriture enregistrée ?

-Données et adresse

Que se passe-t-il lorsque le processeur effectue une requête d'écriture alors que le tampon d'écriture postées est plein?

-On provoque des cycles de gel dans ce cas et le programme est bloqué.

Que se passe-t-il lorsque le processeur fait une requête de lecture (instruction ou donnée) qui fait miss, alors que le tampon d'écriture est non-vidé ? Pourquoi ce comportement ?

-On exécute l'écriture prioritairement, parce qu'il existe la possibilité que l'on doit utiliser ces données ou ces instructions. Bref, c'est pour garantir que le programme soit exécuté séquentiel.

**Question G2 :** Mesurez le temps d'exécution de l'application, le CPI, le coût des écriture, et La fréquence des écritures pour des profondeurs de 1, 2, 4, et 8 mots. A quoi correspond le coût des écritures? Comment expliquez-vous le fait que le coût des écritures soit si faible?

#cycle	F de WRITE /inst	Cout de WRITE	CPI	W-BUF
78847	0.139	0.463	1.389	1
76619	0.139	0.119	1.350	2
75725	0.139	0	1.334	4
75725	0.139	0	1.334	8

Si le tampon d'écriture est plein et qu'il faut tout d'abord mettre le contenu dans la