

TP4 : Architecture des microprocesseurs

Júlia Ellen Dias Leite
julia-ellen@ensta-paris.fr

16 février 2026

Exercice 3 : Analyse des Performances de la Hiérarchie Mémoire

Q1. Configurer le simulateur gem5 avec deux hiérarchies de caches différentes (C1 et C2) et extraire les taux de défauts (Miss Rates) pour l'Instruction Cache (L1I), le Data Cache (L1D) et le cache unifié de niveau 2 (L2).

TABLE 1 – Paramètres de cache pour chaque configuration

Configuration	IL1	DL1	UL2
C1	4kB, DM (assoc=1)	4kB, DM (assoc=1)	32kB, DM (assoc=1)
C2	4kB, DM (assoc=1)	4kB, 2-way (assoc=2)	32kB, 4-way (assoc=4)

Q2. Remplir les tableaux de mesures pour les différents programmes fournis (P1 à P4).

TABLE 2 – Instruction Cache (il1) Miss Rate

Programmes	Configurations de caches	
	C1	C2
P1 (normale)	0.000119	0.000119
P2 (pointeur)	0.000089	0.000089
P3 (tempo)	0.000123	0.000123
P4 (unrol)	0.000141	0.000141

TABLE 3 – Data Cache (dl1) Miss Rate

Programmes	Configurations de caches	
	C1	C2
P1 (normale)	0.298316	0.306917
P2 (pointeur)	0.299729	0.308452
P3 (tempo)	0.299724	0.308445
P4 (unrol)	0.300090	0.305467

Q3. Analysez les résultats obtenus. Pourquoi observe-t-on des variations de performance entre C1 et C2 ? Quel est l'impact de l'associativité sur les taux de défauts observés ?

D'après les résultats obtenus dans les Tableaux 9, 10 et 11, nous pouvons tirer les conclusions suivantes :

TABLE 4 – Unified Cache (ul2) Miss Rate

Programmes	Configurations de caches	
	C1	C2
P1 (normale)	0.437203	0.423355
P2 (pointeur)	0.437227	0.423262
P3 (tempo)	0.437231	0.423260
P4 (unrol)	0.434496	0.425146

- Analyse de l'IL1 :** Le taux de défauts reste identique entre C1 et C2. Cela s'explique par le fait que la configuration de l'IL1 n'a pas été modifiée (4kB, Direct Mapped dans les deux cas).
- Analyse de la DL1 :** On observe une légère augmentation du Miss Rate en C2 (2-way) par rapport à C1 (DM) pour certains programmes. Bien que l'associativité réduise normalement les défauts de conflit, dans des caches de très petite taille (4kB), l'algorithme de remplacement (LRU) peut parfois évincer des données utiles prématièrement par rapport à un mapping direct, ou la structure de l'accès aux données des boucles favorise un mapping fixe.
- Analyse de l'UL2 :** L'augmentation de l'associativité (de DM à 4-way) dans le cache L2 de 32kB montre une amélioration systématique (baisse du Miss Rate) pour C2. Cela démontre que pour un cache de second niveau recevant des flux de données et d'instructions, une associativité plus élevée est cruciale pour réduire les défauts de conflit entre les blocs provenant de la L1I et de la L1D.

Exercice 4 : Mémoires caches - Evaluation des performances de différentes configurations de mémoires caches (instructions et données)

L'objectif principal est la caractérisation détaillée de deux coeurs distincts : le Cortex-A7 (coeur à haut rendement et à exécution séquentielle) et le Cortex-A15 (coeur à hautes performances et à exécution déséquilibrée), à l'aide du simulateur d'architecture gem5 et de l'outil de modélisation physique CACTI.

Q1. Générez le pourcentage de chaque classe d'instructions de ces applications et remplissez les valeurs dans un tableau.

Pour générer le pourcentage de chaque classe d'instructions, nous avons utilisé les statistiques de performance extraites du simulateur gem5 pour les applications Blowfish et Dijkstra. Les classes d'instructions sont regroupées en catégories telles que IntAlu, IntMult, IntDiv, MemRead, MemWrite, FloatMemWrite, et No_OpClass.

TABLE 5 – Pourcentage par classe d'instructions (blowfish, dijkstra)

Classe	Blowfish	Dijkstra
IntAlu	65.48%	63.95%
IntMult	0.00%	3.30%
IntDiv	0.00%	0.00%
MemRead	22.51%	22.38%
MemWrite	12.01%	10.36%
FloatMemWrite	0.00%	0.00%
No_OpClass	0.00%	0.00%

Q2. Quelle catégorie d'instructions nécessiterait une amélioration de performances ? Expliquez en quelques lignes (max 5 lignes).

La catégorie la plus critique est MemRead/MemWrite (mémoire), car elle représente une part importante des instructions et subit les latences mémoire. Améliorer cette catégorie (meilleure hiérarchie de caches, prélecture, réduction des accès) aura l'impact le plus direct sur le temps d'exécution. Les IntAlu dominent en volume mais sont déjà rapides, donc moins sensibles.

Q3. Au regard des résultats obtenus lors du TP2, pouvez-vous justifier d'éventuelles similitudes/divergences comportementales entre dijkstra, BlowFish, SSCA2-BCS, SHA-1 et le produit de polynômes ?

Dijkstra et Blowfish ont des profils d'instructions similaires (dominance d'IntAlu et MemRead / Write), ce qui explique des comportements de cache comparables.

Q4 : Générez les figures de performances détaillées (performance générale, IPC, hiérarchie mémoire, prédiction de branchement, etc.) en fonction de la taille du cache L1 pour les configurations testées. Analysez les résultats. Quelle configuration de L1 donne les meilleures performances pour le Cortex A7 pour les applications sélectionnées ?

Pour analyser les performances du Cortex A7 avec les différentes tailles de cache L1, nous avons générés des figures détaillées pour l'IPC (Instructions Par Cycle) et le CPI (Cycles Par Instruction) pour les applications Blowfish et Dijkstra.

La métrique de l'IPC (Instructions Par Cycle) est un indicateur clé de la performance d'un processeur, mesurant le nombre d'instructions exécutées par cycle d'horloge.

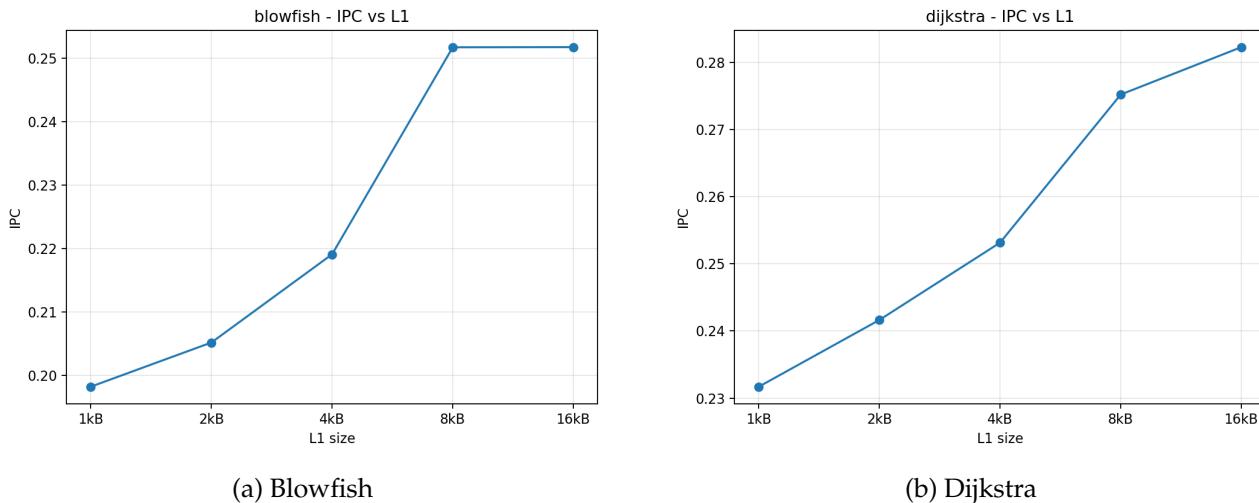
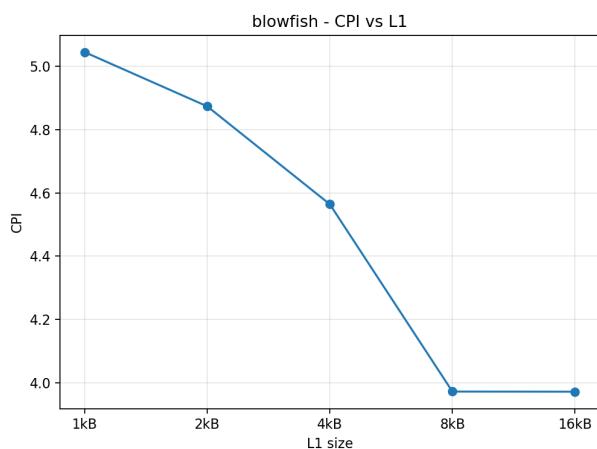
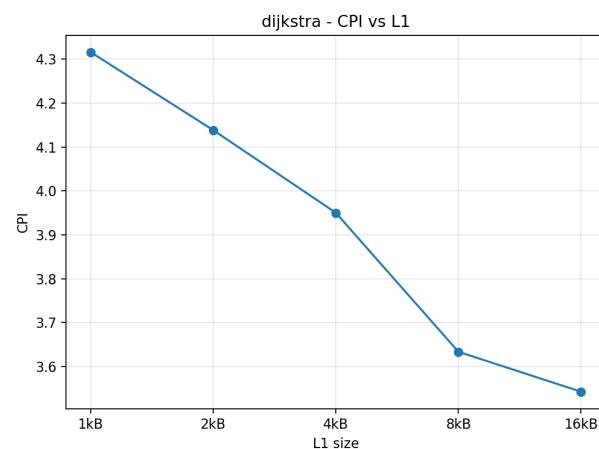


FIGURE 1 – IPC sur Cortex A7 avec différentes configurations de cache L1

La métrique du CPI (Cycles Par Instruction) est également cruciale, car elle indique le nombre de cycles d'horloge nécessaires pour exécuter une instruction. Un CPI plus bas signifie une meilleure performance.



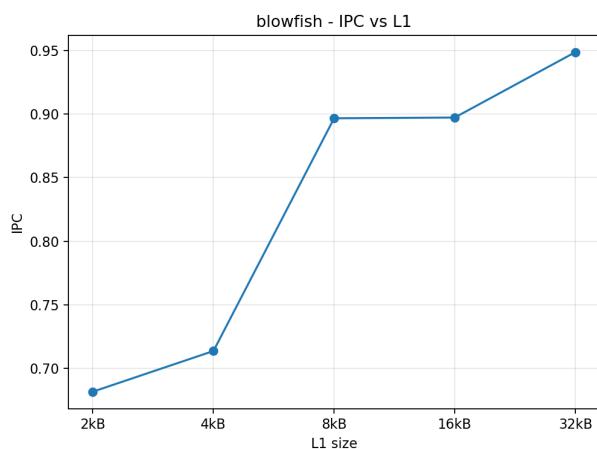
(a) Blowfish



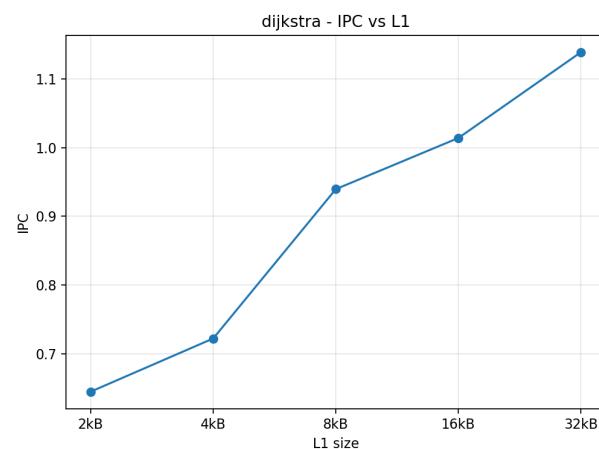
(b) Dijkstra

FIGURE 2 – cpi sur Cortex A7 avec différentes configurations de cache L1

Q5 : Générez les figures de performances détaillées (performance générale, IPC, hiérarchie mémoire, prédition de branchement, etc.) en fonction de la taille du cache L1 pour les configurations testées. Analysez les résultats. Quelle configuration de L1 donne les meilleures performances pour le Cortex A15 pour les applications sélectionnées ?

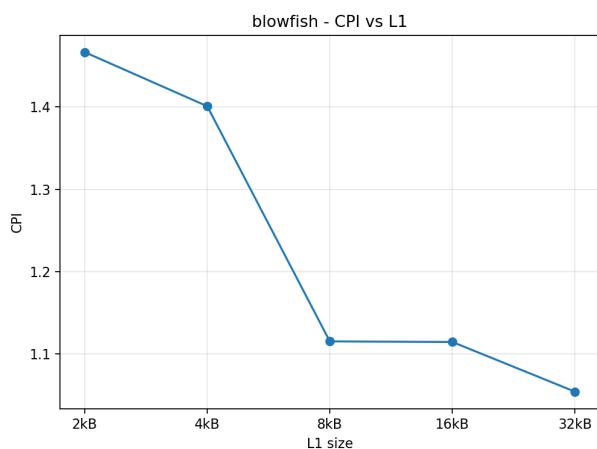


(a) Blowfish

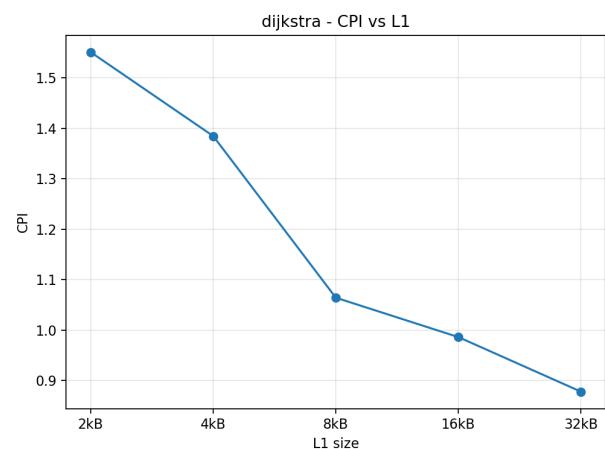


(b) Dijkstra

FIGURE 3 – IPC sur Cortex A15 avec différentes configurations de cache L1



(a) Blowfish



(b) Dijkstra

FIGURE 4 – cpi sur Cortex A15 avec différentes configurations de cache L1