

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

Description du problème

Un panorama détaillé de l'offre actuelle des composants microprocesseurs (8 bits, 16 bits, 32 bits et 64 bits) peut être consulté sur le site EDN (EDN 2014 Microprocessor Directory [1]). Il existe plus de 70 fabricants de microprocesseurs et de microcontrôleurs, chacun se positionnant soit dans le domaine généraliste soit dans un domaine spécifique. Par exemple, le marché des processeurs généralistes pour serveurs et desktops est dominé par Intel [2] et AMD [3], tandis que le marché des processeurs embarqués est dominé par ARM [4] et MIPS [5]. Au sein des processeurs embarqués, on trouve généralement une classification supplémentaire selon le domaine applicatif (automobile, image/vidéo, militaire/espace, télécommunications, etc.).

Cependant, chez les fabricants de ces deux grandes familles de processeurs (généralistes et embarqués), la tendance commune est à l'élargissement de la gamme de processeurs disponibles : processeurs très performants mais énergivores d'un côté, processeurs légèrement moins performants mais plus efficaces de l'autre. Cela rend les frontières entre ces deux mondes de plus en plus floues, mais cela dégage un objectif commun à tous les fabricants : la recherche de l'efficacité énergétique (ratio performances pures à consommation d'énergie donnée), en témoigne la nouvelle architecture **big.LITTLE** de ARM [6].

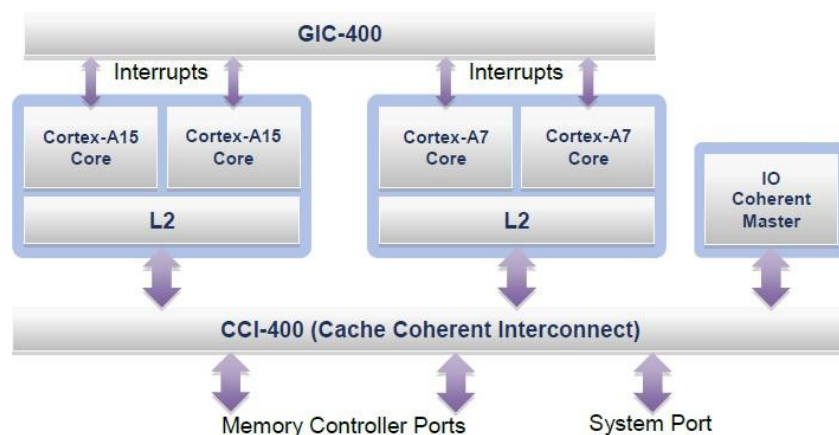


Figure 12 Une configuration typique de big.LITTLE, constituée de 2 clusters, avec 2 CPUs par cluster (Source : ARM)

Cette architecture intègre au sein du même processeur deux cœurs ARM embarqués : un **Cortex A15** (hautes performances, [4]) et un **Cortex A7** (haute efficacité énergétique, [4]). Ces deux cœurs étant entièrement compatibles au niveau du jeu d'instructions, l'architecture peut décider de reloger l'exécution sur l'un ou l'autre des cœurs en fonction des besoins en calcul de l'application. Ces deux cœurs fonctionnent donc de manière alternative (un cœur ON - un cœur OFF), afin de maximiser l'efficacité de l'architecture globale.

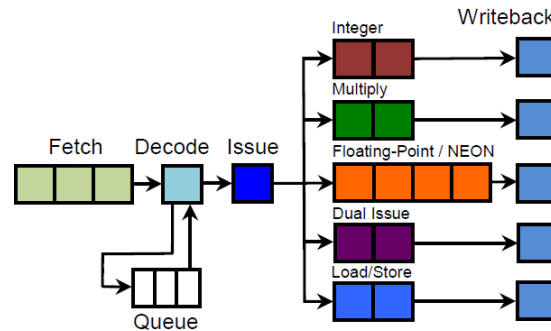


Figure 13 Pipeline du Cortex A7

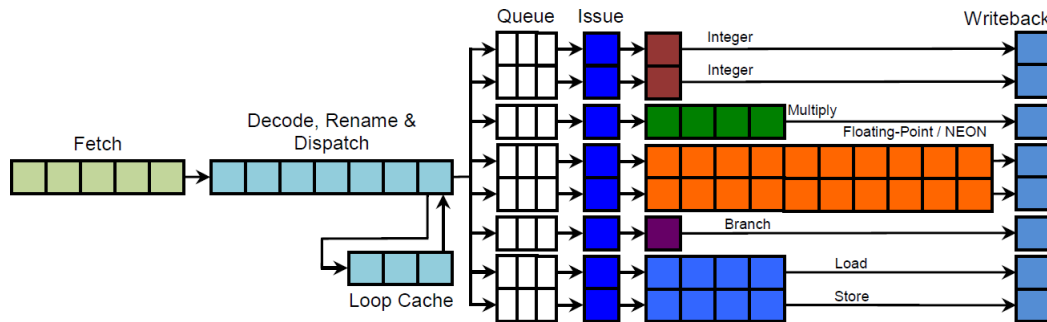


Figure 14 Pipeline du Cortex A15

Comme on a pu le voir, les critères exigés par chaque domaine applicatif (performances, énergie, surface) influencent les choix architecturaux. La microarchitecture d'un processeur spécifie : (1) le mode de traitement des instructions (pipeline, parallélisme d'instructions), (2) l'exécution dans l'ordre ou dans le désordre et enfin (3) le nombre et le type d'unités fonctionnelles (additionneurs, multiplieurs, diviseurs, unités spécialisées, etc.). Il est possible d'identifier pour un programme donné la classe (le type) des instructions sollicitées et le pourcentage d'instructions exécutées issues de cette classe. Cette phase d'identification s'appelle le **profiling**, et a déjà été abordée dans les TD/TP précédents. Il est clair que si le profiling d'une application montre qu'il existe un pourcentage élevé d'utilisation d'une classe particulière, il est alors souhaitable, pour accroître les performances, d'augmenter le nombre d'unités fonctionnelles correspondant à ce type. A contrario, les classes d'instructions ayant un faible taux d'utilisation pourraient voir leur nombre d'unités associées réduites, voire éliminées.

Ce processus consiste à faire une **exploration manuelle/automatique** de la microarchitecture en modifiant les paramètres de configuration de l'architecture et à évaluer l'impact sur les performances et la surface. Néanmoins, ce processus s'applique plutôt à des processeurs en cours de conception. Dans le cas d'une offre existante, il s'agit plutôt d'estimer les performances potentielles en modélisant des processeurs existants.

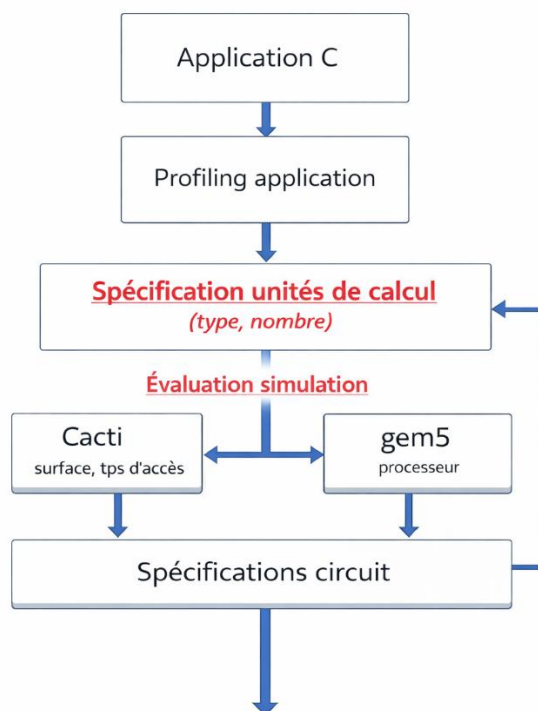


Figure 15 Flot de modification de la microarchitecture d'un microprocesseur

Le but de ce TD/TP est d'analyser les performances des 2 cœurs ARM décrits en introduction : **Cortex A7** et **Cortex A15**, sur des applications type traitement de graphes. Ces paramètres d'architectures ne sont pas supportés par le mode SE de Gem5, nous allons pour cela les simuler sur l'ISA RISCv. Les configurations de ces cœurs ainsi qu'une configuration classique de leurs architectures de caches sont décrites dans le tableau suivant :

Tableau 12 Paramètres de configuration des processeurs

Cœur	I-L1\$ (cache/ bloc/ assoc.)	D-L1\$ (cache/ bloc/ assoc.)	L2\$ (cache/ bloc/ assoc.)	Prédicteur de branchement		Fetch queue	Decode/ Issue/ Commit	RUU/ LSQ	# ALU entiers/ flottants	# multip. entiers/ flottants
				Type	Latence mis-préd. (cycles)					
RISCV A15	32KB/ 64/ 2	32KB/ 64/ 2	512KB/ 64/ 16	2 level BTB=256	15	8	4/8/4	16/16	5/1	1/1
RISCV A7	32KB/ 32/ 2	32KB/ 32/ 2	512KB/ 32/ 8	bimodal BTB=256	8	4	2/4/2	2/8	1/1	1/1

Pour les applications, la suite de benchmarks pour l'embarqué **MiBench** sera utilisée et notamment l'applicatif Dijkstra qui recherche les plus courts chemins d'un graphe. Cette suite est téléchargeable sur <https://github.com/embecosm/mibench>

Une description du benchmark est disponible ici :

<https://vhosts.eecs.umich.edu/mibench/Publications/MiBench.pdf>