Flot automatique d'évaluation pour l'exploration d'architectures à base de mémoires non volatiles

Thibaud Delobelle, Pierre-Yves Péneau, Sophiane Senni, Florent Bruguier, Abdoulaye Gamatié, Gilles Sassatelli, Lionel Torres

Compas'2016 : Parallélisme/ Architecture / Système
Lorient, France, du 5 au 8 juillet 2016

Flot automatique d'évaluation pour l'exploration d'architectures à base de mémoires non volatiles

Thibaud Delobelle, Pierre-Yves Péneau, Sophiane Senni, Florent Bruguier, Abdoulaye Gamatié, Gilles Sassatelli, Lionel Torres

LIRMM / CNRS - Université de Montpellier, 161 rue Ada
34095 Montpellier, Cedex 5
prenom.nom@lirmm.fr

Résumé
Dans cet article, nous présentons, à travers une combinaison judicieuse d'outils, la mise en œuvre d'un flot automatique d'évaluation pour l'exploration d'architectures permettant d'étudier les variations en surface, en puissance, en performance et en consommation énergétique de systèmes multicœurs intégrant des mémoires non volatiles. L'utilisation de ce flot est illustrée sur le benchmark NAS qui caractérise des algorithmes liés à la dynamique des fluides. L'impact de la variation du type de mémoire cache est évalué sur le temps d'exécution et l'énergie consommée.

1. Introduction

Les performances des ordinateurs ne cessent d'augmenter avec l'adoption massive des architectures multicœurs [4]. Dans le même temps, se pose le défi non trivial de l'efficacité énergétique, visant à trouver une adéquation entre la performance et la puisssance dissipée des systèmes multicœurs. Les leviers pour relever ce défi sont multiples car les opportunités d'une meilleure efficacité énergétique existent à différents niveaux, allant des couches logicielles à la technologie, en passant par l'architecture matérielle. Tout d'abord, des optimisations du logiciel peuvent diminuer l'empreinte énergétique de celui-ci [10]. Ensuite, des architectures dynamiques innovantes capables de s'adapter à ce contexte d'exécution pour une meilleure efficacité énergétique sont un autre levier [8]. Enfin l'intégration de composants technologiques aux propriétés physiques moins énergivores est envisageable [7]. Puisque les opportunités d'optimisation de l'efficacité énergétique existent à plusieurs niveaux, il est intéressant d'explorer la question de manière synergique pour un gain maximal. Pour cela, il indispensable de disposer d'outils facilitant une étude à l'échelle de l'ensemble des couches mentionnées précédemment.
Le travail présenté dans cet article vise à répondre à cette attente en proposant un flot automatisé d'évaluation pour l'exploration d'architectures multicœurs intégrant des technologies émergentes de mémoires non volatiles, telles que les mémoires magnétiques [6]. Ce flot permet à un concepteur d'évaluer facilement à l'échelle globale d'un système l'impact de différents choix architecturaux et technologiques. Plus concrètement, nous montrons à travers une combinaison judicieuse d'outils comment ce flot automatique d'évaluation pour l'exploration architecturale permet d'étudier les variations en surface, en performance, en puissance

et en consommation énergétique selon les paramètres architecturaux ou technologiques choisis pour une application donnée. Un accent particulier est mis sur l'exploration de différentes mémoires cache dans cet article. A notre connaissance, il n'existe aucune implantation d'un tel flot dans la littérature. Or celui-ci est un ingrédient indispensable pour adresser facilement à plusieurs niveaux un problème d'actualité tel que l'efficacité énergétique.
La suite de cet article est organisée comme suit : la section 2 présente un rapide état de l'art sur les mémoires non volatiles ; la section 3 décrit le flot d'évaluation d'exploration proposé ; la section 4 illustre une application de ce flot ; enfin la section 5 conclut et présente quelques perspectives.

2. Mémoires non volatiles

Les mémoires non volatiles sont caratérisées par leur capacité à conserver l'information stockée même lorsque l'alimentation électrique est coupée, contrairement aux mémoires volatiles comme les mémoires SRAM. Au delà de la non volatilité, cette propriété physique permet de réduire la consommation en puissance statique de la mémoire car il est inutile de les alimenter constamment en courant.
De nombreuses technologies de mémoires non volatiles sont actuellement en développement [6]. Chacune d'entre elles utilise une variation de résistance pour stocker l'information. Des phénomènes physiques différents sont mis en jeu pour faire varier la valeur de cette résistance. La technologie PCRAM (Phase Change RAM) utilise un phénomène de changement physique du matériau pour stocker l'information. Sous l'effet de la chaleur, une cellulle mémoire passe d'une forme cristaline à une forme amorphe et inversement. La technologie ReRAM (Resistive RAM) fait varier la tension appliquée au matériau isolant composant une cellule mémoire. Cela forme ou casse des filaments conducteurs dans ce matériau. Enfin, la technologie MRAM (Magnetoresistive RAM) utilise des propriétés de magnétisme. Il s'agit d'une technologie assez mature qui sera privilégiée dans cet article.
Le tableau 1 donne quelques éléments de comparaison tirés de la littérature concernant les technologies décrites précédemment par rapport à la technologie SRAM [6]. Le principal défaut des mémoires SRAM est leur consommation statique très élevé contrairement aux mémoires non volatiles. Cependant, certaines mémoires non volatiles ne sont pas performantes au regard de leurs caractéristiques : les temps d'accès pour la mémoire PCRAM, la consommation dynamique pour la mémoire ReRAM ou encore l'endurance (le temps de vie d'une cellule mémoire) pour les deux. Malgré la taille importante des cellules de technologie MRAM au regard des autres mémoires non volatiles, cette technologie reste le meilleur candidat pour remplacer la technologie SRAM.

	SRAM	PCRAM	ReRAM	MRAM
Maturité	Commercialisée	Avancée	Jeune	Avancée
Endurance	$>10^{15}$	$>10^{8}$	$>10^{5}$	$>10^{15}$
Temps d'accès (ns)	<10	10	40-150	10
Energie Statique	Elevée	Faible	Faible	Faible
Energie Dynamique (pJ par accès d'un bit)	>1	2	100	0.02
Taille de cellule (F^2)	>100	4-12	4-10	6-50

TABLE 1 – Comparaison de cellules mémoires non volatiles par rapport aux cellules mémoires SRAM

Les mémoires matures, comme par exemple les mémoires de type SRAM, stockent l'information en utilisant la charge électrique des électrons. Concernant la technologie MRAM, l'infor-

mation est stockée sous la forme d'une résistance électrique dépendant de l'état magnétique de la cellule mémoire. Le bit d'information est égal à 1 si la résistance électrique est maximale, à 0 si la résistance électrique est minimale.
L'état magnétique d'une cellule MRAM peut être modifié par différentes méthodes : Toggle, TAS (Thermally assisted switching), STT (Spin Transfer Torque) et SOT (Spin Orbit Torque). De récents travaux [9] ont conduit une exploration sur leur utilisation à des niveaux de caches L1, L2, mais aussi des bancs de registres du processeur. Ils fournissent également des informations concernant les propriétés de ces technologies (consommation, temps d'accès, etc.). En nous basant sur ces travaux, nous choisissons dans cet article les mémoires de type STTMRAM qui sont les plus matures et les plus utilisées.

3. Flot d'évaluation pour l'exploration d'architectures à base de mémoires non volatiles

Nous présentons le flot d'évaluation développé dans le cadre de notre étude afin de répondre aux besoins de conception d'architectures multicœurs intégrant des mémoires non volatiles.

3.1. Approche générale

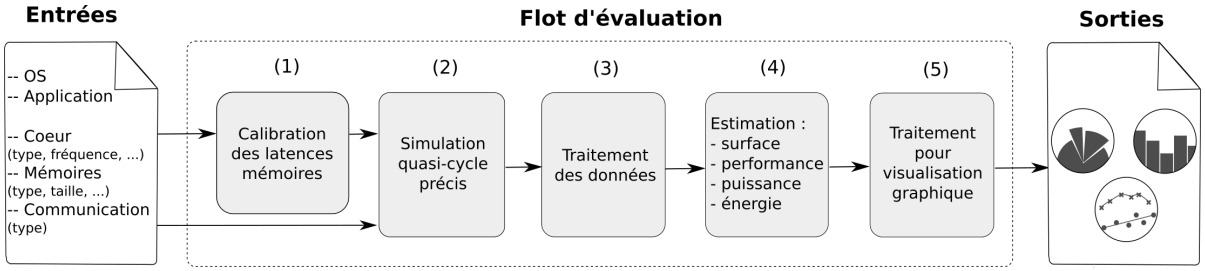


FIGURE 1 – Flot d'évaluation d'architectures multicœurs à mémoires non volatiles.

Le flot est composé de cinq étapes représentées dans la figure 1. Les paramètres d'entrées spécifient : le système d'exploitation et l'application à exécuter, les paramètres des cœurs, des mémoires et des mécanismes de communication. Dans la première étape, les latences des mémoires sont calibrées en fonction des paramètres choisis, i.e. taille, type, associativité et technologie de gravure. Ces latences et le reste des paramètres d'entrée sont utilisés dans la deuxième étape correspondant à la simulation quasi-cycle précis du système. Cela produit des résultats concernant les activités de chaque composant matériel. La troisième étape consiste à extraire ces informations et à les encoder dans un format adapté pour des outils d'estimation de métriques. La quatrième étape utilise ces outils et des modèles de calculs pour estimer la surface, la puissance, les performances et la consommation énergétique du système. Enfin, lors de la dernière étape, ces résultats sont mis en forme pour une visualisation graphique.

3.2. Mise en œuvre

Nous avons implanté ce flot en utilisant différents outils selon les étapes ci-dessus. La première étape correspondant à la calibration des latences de mémoires est réalisée avec NVSim [3]. Il s'agit d'un outil d'évaluation de latences d'accès et d'énergie dynamique pour des transactions mémoires en écriture/lecture, de puissance statique et de surface pour des mémoires non volatiles. Les entrées de NVSim comprennent les caractéristiques d'une mémoire (par exemple : taille, associativité et technologie de gravure) et les paramètres physiques de la technologie de

mémoire considérée. Les informations fournies en sortie sont stockées, et les latences d'accès sont extraites afin de configurer la simulation. Leur précision d'après les auteurs de NVSim est environ de 15% [3]. La deuxième étape est mise-en-oeuvre avec gem5 [2] qui offre une simulation dite "full System" (capable de simuler un système d' exploitation complet) d'architectures multicœurs à un niveau quasi-cycle précis. Dans gem5, différents paramètres du système peuvent être facilement modifiés : types et nombre de cœurs, caratéristiques des caches, système d'exploitation, etc. En résultat d'une simulation, gem5 produit de riches informations statistiques liées surtout aux performances telles que le temps d'exécution, les transactions mémoires effectuées en lecture et en écriture. Ces informations sont extraites dans la troisième étape par un script Python basé sur un travail antérieur [11] que nous avons amélioré en optimisant la lecture des fichiers de gem5 qui peuvent être très volumineux en taille afin de réduire le temps d'extraction des données. L'estimation de la quatrième étape est divisée en deux parties : une estimation des mémoires non volatiles et une estimation du reste de l'architecture. L'estimation des mémoires non volatiles se fait grâce à des calculs utilisant les informations de la simulation et les informations fournies par NVSim durant la première étape du flot. La consommation énergétique des caches lors de l'exécution d'une application est obtenue en sommant les deux composantes suivantes :

– Energie statique : $E_{stat} = P_{stat} * T$, où T est le temps d'exécution d'une application fourni par gem5 et P_{stat} est la puissance statique consommée par une mémoire cache obtenue à l'aide de NVSim.
– Energie dynamique : $E_{dyn} = (E_{dynWrite} * N_{write}) + (E_{dynRead} * N_{read}) + (E_{dynMiss} * N_{miss})$, où les tuples ($E_{dynWrite}$, $E_{dynRead}$, $E_{dynMiss}$) et (N_{write}, $N_{dynRead}$, $N_{dynMiss}$), obtenus respectivement à l'aide de NVSim et gem5, dénotent les énergies dynamiques et le nombre de transactions mémoires selon les trois types d'accès mémoire distingués.

L'estimation du reste de l'architecture est faite avec McPAT [5], un outil développé par HP Labs, qui permet de calculer la consommation énergétique d'une architecture matérielle n'intégrant pas de technologies émergentes telles que les mémoires non volatiles. Il fournit une estimation de la surface et de la consommation énergétique de chaque composant de l'architecture grâce aux statistiques extraites auparavant. La précision de McPAT est évaluée autour de 20% selon ses auteurs [5]. Enfin, pour permettre une bonne visualisation des résultats, différents scripts traitent ces estimations et génèrent des graphes.

4. Validation du flot d'évaluation

4.1. Cadre expérimental

Pour illustrer notre flot, nous considérons le benchmark "NAS parallel benchmark" [1] développé par la NASA. NAS simule le calcul et les mouvements de données caractéristiques des applications de la dynamique des fluides. Les différents kernels définis au sein de NAS sont brievement décrits dans le tableau 2.

L'architecture considérée comporte 8 cœurs Cortex-A15 fonctionnant à une fréquence de 2GHz. Chaque cœur comporte des mémoires caches d'instructions et de données, respectivement L1i et L1d, ayant chacune une taille de 32Ko et une associativité de 4. Le cache L2 est partagé par tous les cœurs. Sa taille et son associativité sont respectivement de 2Mo et 8. Comme système d'exploitation, nous utilisons la version 3.14.0 de Linux. Les configurations système explorées sont décrites comme suit :

– 2 hiérarchies de caches intégrant des MRAM (L1i_L1d_L2) : SRAM_SRAM_STTMRAM et STTMRAM_SRAM_STTMRAM
– 2 hiérarchies de caches avec taille variable de cache L1i (L1i_L1d_L2) : 32Ko_32Ko_2Mo

TABLE 2 – Description des kernels du benchmark NAS.

Kernel	Description
bt	Résolution d'un système non-linéaire par blocs
cg	Calcul du gradient conjugé
dc	Tri multi-dimensionnel de données
ft	Transformation de Fourrier
is	Tri de nombres entiers par paquets
lu	Résolution d'un système non-linéaire par surrelaxation successive
mg	Résolution d'une équation discrète de Poisson
ua	Résolution d'une équation de chaleur
sp	Résolution d'un système non-linéaire par scalaire pentadiagonal

et 64Ko_32Ko_2Mo

Les latences d'écritures des mémoires STTMRAM n'étant pas compétitives avec la SRAM, nous choisissons de ne pas modifier le cache L1 de données (L1d). Nous utilisons un cache L1 d'instructions en lecture-seule.

4.2. Résultats d'évaluation

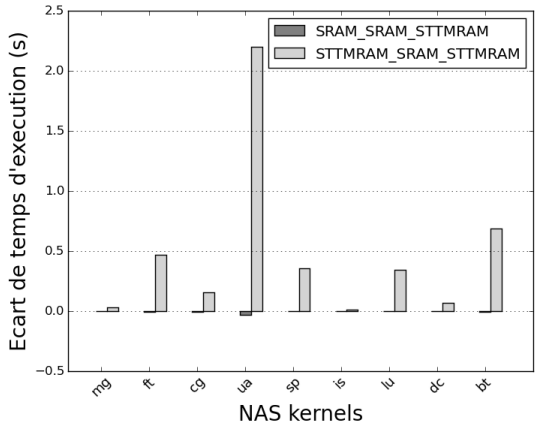
La première expérimentation consiste à explorer les variations de la consommation énergétique et du temps d'exécution en intégrant des mémoires de type STTMRAM dans une architecture de caches par rapport à une configuration dite "full SRAM" (où tous les caches sont de type SRAM) comme représenté sur la figure 2.

Sur le graphe de la figure 2a, une architecture avec uniquement un cache L2 de type STTMRAM a un temps d'exécution identique (voire légèrement inférieur) qu'une architecture "full SRAM". Dans une telle situation, on observe dans la figure 2b une diminution de la consommation énergétique grâce notamment au gain en puissance statique inhérent aux mémoires magnétiques.

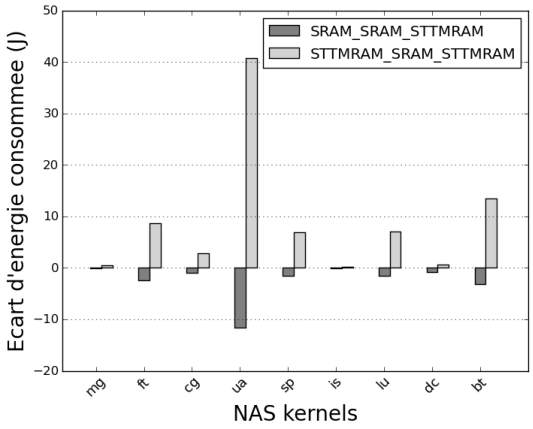
Cependant, lorsque les caches L1i et L2 sont tous les deux de type STTMRAM, le temps d'exécution est plus élevé qu'en "full SRAM" comme l'illustre la figure 2a. Cela masque les potentiels gains en puissance statique des STTMRAM et entraîne l'augmentation de la consommation énergétique de l'ensemble des composants de l'architecture comme illustré dans la figure 2b.

Par ailleurs, des travaux existants [7] montrent qu'à partir d'une taille de 64Ko, les latences en lecture des mémoires STTMRAM sont équivalentes à celles des mémoires SRAM. La tendance montre même qu'au delà de 64Ko, elles sont plus petites pour des mémoires STTMRAM. Sur la base de cette information, nous augmentons ainsi la taille du cache L1i de 32Ko à 64 Ko. Les effets résultant de cette modification sont illustrés dans les figures 2c et 2d. La différence en temps d'exécution entres les différentes architectures ayant toutes un cache L1i de 64Ko a diminué par rapport au scénario précédent de la figure 2a. On observe également cette diminution sur la consommation énergétique dans la figure 2d, avec même un gain pour le kernel dc.

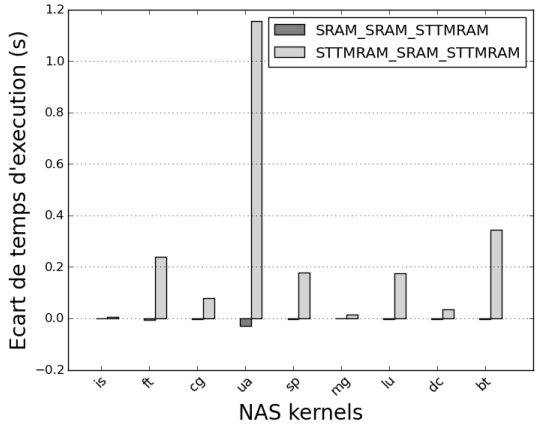
Au delà d'une analyse centrée sur les mémoires caches, notre flot fournit également la consommation énergétique d'autres composants de l'architecture tels que les cœurs ou les bus. La figure 3 présente deux répartitions de la consommation énergétique pour un certain nombre de composants d'une architecture. Nous observons clairement que la consommation énergétique des composants représentés est inchangée pour les deux configurations, excepté pour le cache L2 qui diminue significativement grâce à la faible puissance statique de la technologie STTMRAM.



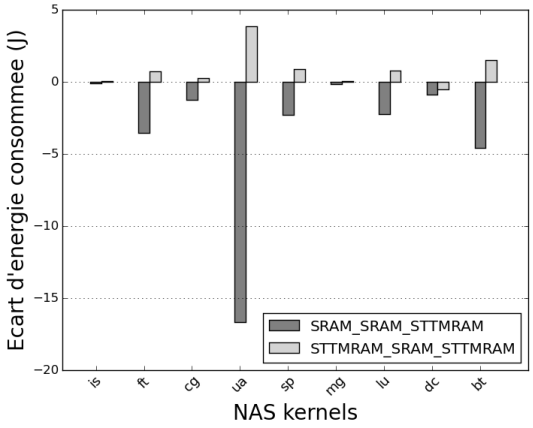
(a) Ecart en temps d'exécution (L1i 32Ko)



(b) Ecart en énergie (L1i 32Ko)



(c) Ecart en temps d'exécution (L1i 64Ko)



(d) Ecart en énergie (L1i 64Ko)

FIGURE 2 – Ecarts en temps d'exécution et en énergie consommée des configurations de caches (L1i_L1d_L2) SRAM_SRAM_STTMRAM et STTMRAM_SRAM_STTMRAM par rapport à une configuration "full SRAM".

De plus, le flot fournit des informations sur la surface des composants de l'architecture. Pour l'architecture "full SRAM" la surface totale de l'architecture est de 97.6 mm^2 alors que celle avec un cache L2 de type STTMRAM a une surface totale de 81.4 mm^2, soit un gain d'environ 17%. Cela s'explique par la taille des cellules qui sont plus petites pour les mémoires STTMRAM que les mémoires SRAM.
Le temps de simulation de ces expériences est en moyenne de 32 heures. Il est dû en majeure partie à gem5 (environ 99%) et dépend fortement de l'architecture et de l'application simulée.

5. Conclusion et perspectives

Dans cet article, nous avons présenté la mise en oeuvre d'un flot automatisé d'évaluation pour l'exploration d'architectures multicœurs intégrant des technologies émergentes de mémoires non volatiles, telles que les mémoires magnétiques. Ce flot vise à aider un concepteur à évaluer facilement l'impact de différents choix architecturaux et technologiques dans un système. En guise de validation, nous avons montré quelques expérimentations préliminaires quant à l'utilisation de notre flot en focalisant l'analyse sur des hiérachies hybrides de

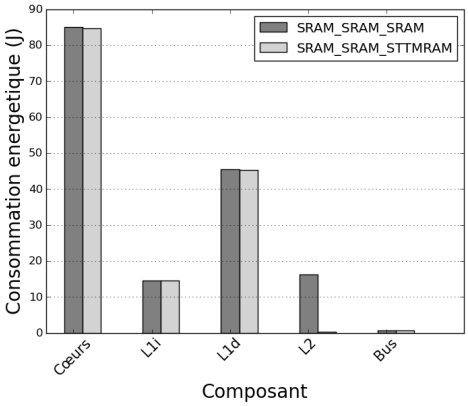


FIGURE 3 – Répartition de la consommation énergétique par cœur, niveaux de caches et bus : configuration "full SRAM" vs. configuration SRAM_SRAM_STTMRAM

mémoires caches incluant les technologies SRAM et STTMRAM. La perspective de ce travail est de définir une exploration multi-niveau ciblant à la fois les parties logicielles, architecturales et technologiques. Cela permettra d'étudier de façon synergique l'impact des choix de conception à ces différents niveaux sur l'efficacité énergétique.

6. Remerciements

Ce travail a été réalisé dans le cadre du projet ANR CONTINUUM (ANR-15-CE25-0007-01).
http ://www.lirmm.fr/continuum-project

Bibliographie

1. Bailey (D. H.), Barszcz (E.), Barton (J. T.), Browning (D. S.), Carter (R. L.), Dagum (L.), Fatoohi (R. A.), Frederickson (P. O.), Lasinski (T. A.), Schreiber (R. S.) et al. – The NAS parallel benchmarks. *International Journal of High Performance Computing Applications*, vol. 5, n3, 1991, pp. 63–73.
2. Binkert (N.), Beckmann (B.), Black (G.), Reinhardt (S. K.), Saidi (A.), Basu (A.), Hestness (J.), Hower (D. R.), Krishna (T.), Sardashti (S.) et al. – The gem5 simulator. *ACM SIGARCH Computer Architecture News*, vol. 39, n2, 2011, pp. 1–7.
3. Dong (X.), Xu (C.), Xie (Y.) et Jouppi (N. P.). – Nvsim : A circuit-level performance, energy, and area model for emerging nonvolatile memory. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 31, n7, 2012, pp. 994–1007.
4. Esmaeilzadeh (H.), Blem (E.), Amant (R. S.), Sankaralingam (K.) et Burger (D.). – Dark silicon and the end of multicore scaling. – In *International Symposium on Computer Architecture (ISCA)*, pp. 365–376. IEEE, 2011.
5. Li (S.), Ahn (J. H.), Strong (R. D.), Brockman (J. B.), Tullsen (D. M.) et Jouppi (N. P.). – Mcpat : an integrated power, area, and timing modeling framework for multicore and manycore architectures. – In *IEEE/ACM International Symposium on Microarchitecture*, pp. 469–480. IEEE, 2009.
6. Mittal (S.), Vetter (J. S.) et Li (D.-J.). – A survey of architectural approaches for managing embedded dram and non-volatile on-chip caches. 2014.
7. Oboril (F.), Bishnoi (R.), Ebrahimi (M.) et Tahoori (M. B.). – Evaluation of hybrid memory

technologies using sot-mram for on-chip cache hierarchy. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,*, vol. 34, n3, 2015, pp. 367–380.

8. Pathania (A.), Jiao (Q.), Prakash (A.) et Mitra (T.). – Integrated cpu-gpu power management for 3d mobile games. – In *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE*, pp. 1–6. IEEE, 2014.
9. Senni (S.). – *Exploration of non-volatile magnetic memory for processor architecture*. – PhD thesis, University of Montpellier 2, 2015.
10. Tiwari (V.), Malik (S.) et Wolfe (A.). – Compilation techniques for low energy : An overview. – In *IEEE Symposium on Low Power Electronics*, pp. 38–39. IEEE, 1994.
11. Vilanova (L.). – gem5 to mcpat converter, feb 2015. https ://projects.gso.ac.upc.edu/projects/gem5-mcpat.