**Projet**

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

# Analyse énergétique d'un processeur vectoriel pour des calculs de DNN
# Proposition

ELE6307 - Machines neuronales : architectures et applications

Hiver 2022
Département de génie électrique
École Polytechnique de Montréal

Dernière mise à jour: 23 mars 2022


Yoan **Fournier** 1958736

Victor **Gaudreau-Blouin** 1958297

## Présentation du sujet

Dans les dernières années, l'utilisation de réseaux de neurones profonds (DNN) pour résoudre différentes tâches a beaucoup augmenté. Ces DNNs nécessitent des puissances de calculs considérables, mais à la fois nécessitent une bonne efficacité énergétique étant donné leur déploiement dans des appareils mobiles. Bien que les processeurs généralistes que l'on retrouve aujourd'hui ont augmenté rapidement en performance, les limitations du *Dennard Scaling* se font ressentir. Il est donc souhaitable de trouver une autre approche pour accélérer de façon efficace les calculs dans les DNNs. Une manière intéressante pour l'accélération des calculs pour des DNNs est l'utilisation de processeurs vectoriels plutôt que des processeurs scalaires standards. Ces processeurs utilisent un jeu d'instruction SIMD (*Single instruction, Multiple Data*) plutôt que SISD (*Single instruction, Single data*). Ainsi une instruction SIMD peut performer la même opération sur plusieurs données plutôt qu'une seule, ce qui permet de plus facilement augmenter le parallélisme des calculs. Le projet proposé vise donc à étudier l'efficacité énergétique pour différentes configurations de processeurs vectoriels en comparant cette efficacité avec une architecture de processeur scalaire standard.

## Méthodologie

L'architecture sur laquelle se basera notre modélisation est sur celle du coprocesseur vectoriel ARA [2]. Ce coprocesseur est une extension vectorielle du processeur scalaire RISCV CVA6. Le fonctionnement d'ARA se base sur le concept de *Lanes*. La figure 1 présente l'architecture d'ARA [1]. Le coprocesseur ARA possède un nombre d'allées, ou *Lanes*, qui sont capables de traiter des éléments vectoriels indépendament. Le routage des données vers ces *Lanes* grandit évidemment en complexité avec le nombre de *Lanes*. Le *Sequencer* s'occupe de lire les instructions à exécuter et transmettre les informations correctes au *Vector Load and Store Unit* (VLSU), *Slide Unit* (SLDU) ou aux *Lanes*. Le VLSU est responsable de générer les bonnes addresses, amenant les données séquentiellement vers et depuis les *Lanes*. Le SLDU exécute des opérations qui n'ont pas d'indépendance entre les *Lanes*. Finalement, chaque *Lane* a son propre *sequencer* pour les instructions à exécuter sur son élément vectoriel. Le reste de la *Lane* peut être considéré comme un PE, avec sa mémoire locale, un étage de conversion et un étage d'exécution.

La première étape du projet visera à modéliser l'architecture d'une *Lane* à l'aide de Timeloop/Accelergy [3]. Timeloop/Accelergy est un simulateur d'architecture d'accélérateurs de DNN permettant nous permettant de définir une architecture et un flot de données. Timeloop se charge de calculer les accès à la mémoire et aux PE, tandis qu'Accelergy estime l'énergie nécessaire à chaque opération. En définissant correctement les coûts d'accès et l'architecture correctement, nous pouvons avoir une bonne approximation de l'efficacité énergétique de l'architecture en question pour un certain flot de données. L'architecture présentée à la figure 1b est plutôt complexe, mais elle peut, à sa plus simple expression, être modélisée comme un PE avec une mémoire locale de 8 données. Ainsi, par la suite, l'architecture globale sera modélisée en combinant un nombre paramétrable de

*Lanes* à une mémoire globale. Le processeur scalaire, lui peut être modélisé en utilisant une seule *Lane*.

La deuxième étape utilisera *timeloop mapper* afin de venir analyser la consommation énergétique de l'architecture pour un problème standard de convolution comme, par exemple, une des couches convolutives du DNN VGG16. La consommation sera évaluée pour 2, 4, 8 et 16 *Lanes*. Ensuite, d'autres paramètres comme la taille en bits des poids pourront être modifiés afin d'évaluer leur impact sur la performance énergétique.

La mesure de performance qui sera utilisée est une mesure d'efficacité en GFLOP/J. La commande *timeloop mapper* donne l'énergie totale et le nombre de cycles pour le mapping entre un problème et une architecture. On peut donc calculer les GFLOP/J comme ceci :

$$\text{GFLOP/J} = \frac{\#cycles \times \#lanes}{E_{tot}}$$

Avec une fréquence d'horloge connue, nous pouvons évaluer la puissance nécessaire pour le circuit comme ceci :

$$P = f \times \#lanes \times \frac{E}{op}$$

## Objectifs et résultats

L'objectif du projet et de comparer l'efficacité et l'évolutivité d'un processeur vectoriel vis-à-vis un processeur scalaire plus standard pour les types de calculs principalement utilisés dans des DNNs. Le projet vise à démontrer qu'il est avantageux d'opter pour une architecture vectorielle.

Les résultats présentés seront les mesures d'efficacité énergétique en GFLOP/J pour un processeur scalaire (1 *Lanes*) et un processeur vectoriel à (2, 4, 8, 16 *Lanes*).

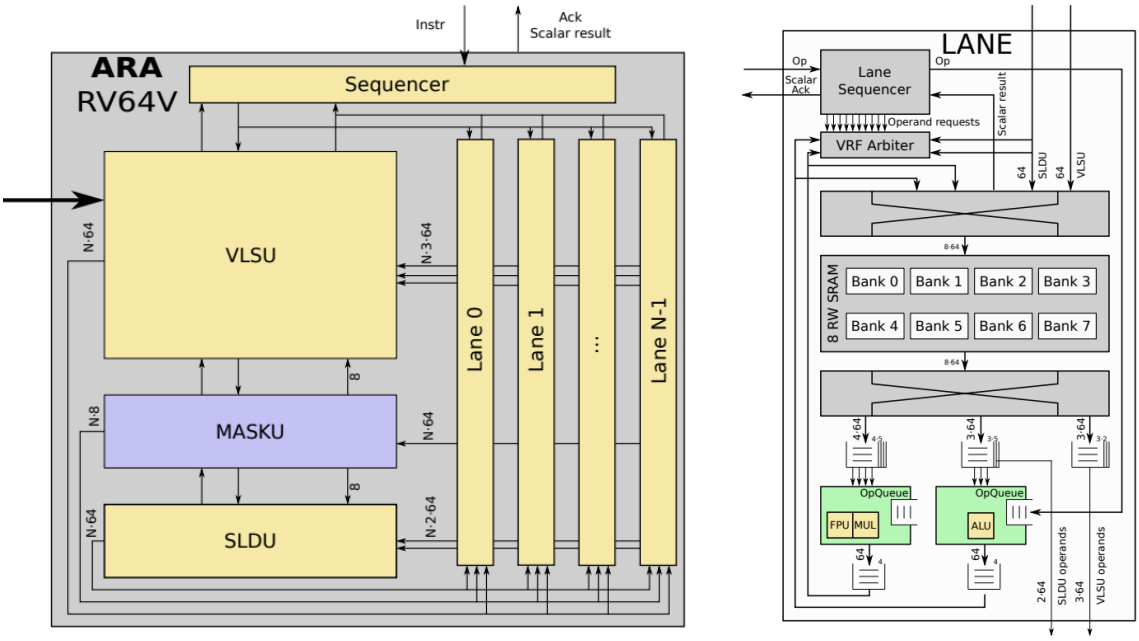
## Références

[1] B. Bougenot. Ara : Update pulp's vector processor. Master's thesis, Swiss Federal Institute of Technology in Zürich (ETH), 2020.

[2] M. Cavalcante, F. Schuiki, F. Zaruba, M. Schaffner, and L. Benini. Ara : A 1 ghz+ scalable and energy-efficient risc-v vector processor with multi-precision floating point support in 22 nm fd-soi, 2019.

[3] A. Parashar, P. Raina, Y. S. Shao, Y.-H. Chen, V. A. Ying, A. Mukkara, R. Venkatesan, B. Khailany, S. W. Keckler, and J. Emer. Timeloop : A systematic approach to dnn accelerator evaluation. In *2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pages 304–315, 2019.

# Annexes




(a) Arch. globale

(b) Arch. *Lane*

FIGURE 1 – Architecture d'ARA