

Offre de stage 2024 : Overclocking sur FPGA

Encadrant : Thibaut Marty (thibaut.marty@insa-rennes.fr)

Lieu de stage : Laboratoire IETR / INSA Rennes, bâtiment 10

Profil recherché

Vous êtes en Bac+4 ou Bac+5 (en école d'ingénieur ou Master) en informatique ou électronique ; vous êtes intéressé par la recherche scientifique. Vous avez des notions d'architecture des calculateurs et de développement d'accélérateurs pour FPGA (Verilog, VHDL, HLS ou autres).

Résumé du stage

Contexte

La *spéculation temporelle*, aussi appelée *overclocking*, est une technique permettant d'améliorer les performance (calculs par seconde) et l'efficacité énergétique (calculs par joule) de noyaux de calculs [1], [2], notamment sur des plateformes de calcul intensif comme les FPGA.

En général, un noyau de calcul spéculatif est accompagné d'un mécanisme de détection d'erreur qui permet de déterminer si les résultats de calcul sont corrects [1], [3], [4]. Dans le cas où une erreur est détectée, le calcul doit être exécuté à nouveau, dans des conditions ayant moins de chance de provoquer des erreurs (plus faible fréquence), afin d'obtenir le résultat correct.

Ainsi, les gains espérés grâce à la spéculation temporelle sont la somme des impacts positifs et négatifs du système complet :

- Sur l'efficacité énergétique :
 - le gain brut de la spéculation temporelle en fonction du taux de spéculation ;
 - le coût énergétique du mécanisme de détection d'erreur (il nécessite de l'énergie pour détecter les erreurs) ;
 - le coût énergétique du deuxième calcul lorsqu'une erreur est détectée ;
 - le coût énergétique des mémoires supplémentaires nécessaires.
- Sur les performances :
 - le gain brut de la spéculation temporelle en fonction du taux de spéculation ;
 - le coût en performance du mécanisme de détection d'erreur (il nécessite des ressources de calcul qui ne sont pas allouées au noyau de calcul) ;
 - la pénalité lorsqu'une erreur est détectée, due à l'exécution une seconde fois le calcul, en fonction du taux d'erreur.

La décision de rendre un noyau de calcul spéculatif dépend donc de la différence entre les gains espérés et les coûts. Elle dépend également de certaines conditions portant sur l'exécution du noyau dans un système complet : le noyau se trouve-t-il sur le *chemin critique* de l'application ? Les contraintes temps réel sont-elles respectées en cas d'erreur ? etc.

Ce type de décision peut être pris automatiquement dans le cadre d'outils permettant de modéliser des applications et de générer des plateformes tels que Preesm [5].

Objectifs

L’objectif de ce stage est de concevoir une plateforme sur FPGA pour permettre l’exécution de plusieurs noyaux spéculatifs pour une ou quelques applications (par exemple : une chaîne de traitement vidéo ou un réseau de neurones).

La plateforme pourra permettre de mesurer les coûts et gains de la spéculation temporelle pour plusieurs noyaux d’exécution et de créer des modèles de consommation d’énergie et de performance.

Dans un second temps et à partir de ces mesures, l’objectif est de développer une méthodologie permettant de déterminer quels noyaux d’une application devraient être spéculatifs afin de maximiser les performances et l’efficacité énergétique d’une chaîne de traitement.

Références

- [1] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and others, “Razor: A low-power pipeline based on circuit-level timing speculation,” in *Proceedings. 36th annual IEEE/ACM international symposium on microarchitecture, 2003. MICRO-36.*, 2003, pp. 7–18.
- [2] J. Nunez-Yanez, “Energy proportional neural network inference with adaptive voltage and frequency scaling,” *IEEE Transactions on Computers*, vol. 68, no. 5, pp. 676–687, 2018.
- [3] J. Nunez-Yanez, “Energy proportional computing in commercial FPGAs with adaptive voltage scaling,” in *Proceedings of the 10th FPGAworld conference*, 2013, pp. 1–5.
- [4] T. Marty, T. Yuki, and S. Derrien, “Safe overclocking for CNN accelerators through algorithm-level error detection,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 12, pp. 4777–4790, 2020.
- [5] M. Pelcat, K. Desnos, J. Heulot, C. Guy, J.-F. Nezan, and S. Aridhi, “Preesm: A dataflow-based rapid prototyping framework for simplifying multicore DSP programming,” in *Education and research conference (EDERC), 2014 6th european embedded design in*, 2014, pp. 36–40.