Project AdaptING - PhD proposal - starting September 2023

Reconfigurable Computing in Memory

Context

Memory is a vital part of any digital system, but memory technologies have never really managed to keep up with the blistering pace imposed by continual advances in processors. Computer performance hits a limit: the memory wall. Through the many hardware solutions implemented from generation to generation, more and more transistors in a circuit are dedicated to the sole purpose of improving memory accesses. These accesses are nowadays more expensive than an arithmetic operation. As a result, the total energy spent for moving data has reached excessive proportions. Therefore, in order to ever improve performances and energy efficiency of either embedded systems or power-supply devices, new memory architectures and organisations are needed. One trend is to explore domain specific architectures (DSA). DSAs make it possible to better use the different forms of parallelism in an application domain, they can make better use of the memory hierarchy, and can benefit from arithmetic precision that better meets the application needs. Coarse-Grained Reconfigurable Architectures (CGRA) are the ideal candidates to build DSAs and they offer the right trade-off between flexibility (programmability) and performance [2]. CGRAs feature a set of simple processing elements tightly interconnected, and are a key member of the reconfigurable computing family. Directly connected to the same memory as the processor, these devices are used to offload the processor from data- and compute-intensive workloads. Coming with their associated compilation chain, CGRAs are the ideal candidates to operate as programmable hardware accelerators.

The other trend is to compute where the data is: in the memory. Computing in memory (CIM) or Near Memory Computing (NMC) are two incarnations of this trend. Numerous studies have explored various solutions as three dimensions (computation location, memory technology, and computation parallelism) offer a wide exploration space [3]. In this thesis, the right location for the CGRA between CIM-Periphery (CIM-P), and Computation-Outside-Memory Near (COM-N) will be particularly studied. How to reconfigure the computing part is studied for decades now, but how to reconfigure the memory part is still at its early stage nowadays.

Goals

The goal of the thesis is to explore together two promising trends to relief memory pressure: DSA and computing in memory (CIM), a prospective suggested in [1]. Besides, although CGRAs are inherently suited for parallel programming models (like OpenMP, SYCL, etc.), only few of them support these models [1].

The objectives of this thesis are: (i) to explore a CGRA-based CIM-P or COM-N architecture, (ii) to set up accordingly the compilation flow, (iii) to run experiments on applications from AI domain. The work will focus on the design of the reconfigurable memory and the associated compilation tools.

Keywords

'CGRA', 'Memory', 'Compilation', 'AI'

Skills

- hardware architectures, VHDL, Verilog, SystemVerilog
- LLVM
- C/C++
- AI applications

Location

- Research team: ARCAD, Lab-STICC
- Address: Lab-STICC rue Saint-Maudé 56100 Lorient France

Supervisors

- Philippe Coussy (Lab-STICC, Equipe ARCAD, Lorient) - philippe.coussy@univ-ubs.fr
- Kevin Martin (Lab-STICC, Equipe ARCAD, Lorient) - kevin.martin@univ-ubs.fr

Application

Send resume and application letter to kevin.martin@univ-ubs.fr

References

[1] Liu, L., Zhu, J., Li, Z., Lu, Y., Deng, Y., Han, J., Yin, S., Wei, S.: A survey of coarse-grained reconfigurable architecture and design: Taxonomy, challenges, and applications. ACM Comput. Surv. 52(6) (Oct 2019). https://doi.org/10.1145/3357375 [2] Podobas, A., Sano, K., Matsuoka, S.: A survey on coarse-grained reconfigurable architectures from a performance perspective. IEEE Access 8, 146719–146743 (2020). https://doi.org/10.1109/ACCESS.2020.3012084 [3] Hoang Anh Du Nguyen, Jintao Yu, Muath Abu Lebdeh, Mottaqiallah Taouil, Said Hamdioui, and Francky Catthoor. 2020. A Classification of Memory-Centric Computing. J. Emerg. Technol. Comput. Syst. 16, 2, Article 13 (April 2020), 26 pages. https://doi.org/10.1145/3365837