Hindawi
International Journal of Reconfigurable Computing
Volume 2020, Article ID 2725809, 2 pages
https://doi.org/10.1155/2020/2725809

Editorial

FPGAs for Domain Experts

Wim Vanderbauwhede,[1] **Sven-Bodo Scholz**,[2] **and Martin Margala**[3]

[1]*University of Glasgow, Glasgow, UK*
[2]*Heriot-Watt University, Edinburgh, UK*
[3]*University of Massachusetts Lowell, Lowell, MA, USA*

Correspondence should be addressed to Wim Vanderbauwhede; wim.vanderbauwhede@glasgow.ac.uk

Received 4 May 2020; Accepted 16 September 2020; Published 27 October 2020

Field-Programmable Gate Arrays (FPGAs) have recently gained a lot of attention through demonstrated superior performance over off-the-shelf architectures, not only with respect to energy efficiency but also with respect to wall-clock runtimes. For a long time, FPGAs had been used primarily as prototyping devices or in embedded systems but are now increasingly accepted as first-order computing devices on desktops and servers. This change has been driven by a combination of increasingly larger and resourceful FPGAs and wider availability of mature and stable high-level FPGA programming tools.

The application areas range across many domains from high-finance to advanced machine learning. Despite the availability of many tools for high-level synthesis and increasing ease of access to FPGA-based computing nodes (e.g., via Amazon Web Services), domain experts still are far away from utilising FPGAs to gain processing performance unless preconfigured systems for their particular applications exist in readily available form. Manycore CPUs and GPUs are still generally considered the only viable options for domain experts looking to accelerate their applications.

Against this background, there has been considerable research in recent years on making FPGAs accessible for domain experts. With this special issue, we are bringing together work that aims to break this barrier for a wider applicability of FPGAs.

This special issue combines contributions from researchers and practitioners that share the vision of enabling domain experts to benefit from the performance opportunities of FPGAs.

We hope that you enjoy this special issue, and this paper collection as a whole can introduce readers to the varied and challenging area of FPGA computing, presenting several state-of-the-art solutions from diverse perspectives. All accepted papers provide relevant and interesting research techniques, models, and work directly applied to the area of scientific FPGA programming.

Finally, we would like to thank all the authors for their submissions to this special issue and the also the reviewers for dedicating their time to provide detailed comments and suggestions that helped to improve the quality of this special issue.

The first paper, "Automatic Pipelining and Vectorization of Scientific Code for FPGAs," focuses on FPGA compilation of legacy scientific code in Fortran. There is a very large body of legacy scientific code still in use today, and much new scientific code is still being written in Fortran-77. Many of these codes would benefit from acceleration on GPUs and FPGAs. Manual translation of such legacy code parallel code for GPUs or FPGAs requires a considerable manual effort. This is a major barrier to wider adoption of FPGAs. The authors of this paper have been developing an automated optimizing compiler to lower this barrier. Their aim is to compile legacy Fortran code automatically to FPGA, without any need for rewriting or insertion of pragma. The compiler applies suitable optimizations based on static code analysis. The paper focuses on two key optimizations, automatic pipelining and vectorization. The compiler identifies portions of the legacy code that can be pipelined and vectorized. The backend generates coarse-grained pipelines and

automatically vectorizes both the memory access and the data path based on a cost model, generating an OpenCL-HDL hybrid solution for FPGA targets on the Amazon cloud. The results show up a performance improvement of up to four times over baseline OpenCL code.

The second paper, "Dimension Reduction Using Quantum Wavelet Transform on a High-Performance Reconfigurable Computer," introduces a very interesting and exciting new field, the use of FPGAs for the acceleration of quantum computing simulations. Simulation is a crucial step in the development of quantum computers and algorithms, and FPGAs have huge potential to accelerate this type of simulations. The paper proposes to combine dimension reduction techniques with quantum information processing for application in domains that generate large volumes of data such as high-energy physics (HEP). It focuses on using quantum wavelet transform (QWT) [1] to reduce the dimensionality of high spatial resolution data. The quantum wavelet transform takes advantage of quantum superposition to reduce computing time for the processing of exponentially larger amounts of information. The authors present a new emulation architecture to perform QWT and its inverse on high-resolution data, and a prototype of an FPGA-based quantum emulator. Experiments using high-resolution image data on a state-of-the-art multinode high-performance reconfigurable computer show that the proposed concepts represent a feasible approach to reducing the dimensionality of high spatial resolution data generated by applications in HEP.

The third paper, "Translating Timing into an Architecture: The Synergy of COTSon and HLS (Domain Expertise—Designing a Computer Architecture via HLS)," provides an in-depth description of a high-level synthesis workflow around Vivado HLS tools. It comprises tools on both sides of HLS: tools for design space exploration prior to running HLS named COTSon and MYDSE as well as tools for targeting a custom build hardware, the AXIOM board. The article provides a good overview of the tools and the overall workflow through the HLS tool. The abstract description of the workflow is substantiated by an in-depth presentation of example applications including the design of a system for distributed computation across multiple FPGA boards. Finally, some empirical evidence for the predictive capabilities of the tool chain is being presented. Overall, this contribution not only demonstrates the challenges involved when designing complex systems with HLS at the core nicely but also features the presentation of custom-made tooling which can be used by the wider community.

The fourth paper, "An FPGA-Based Hardware Accelerator for CNNs Using On-Chip Memories Only: Design and Benchmarking with Intel Movidius Neural Compute Stick," presents a full on-chip FPGA hardware accelerator for a separable convolutional neural network designed for a keyword spotting application. This is a quantized neural network realized exclusively using on-chip memories. The design is based on the Intel Movidius Neural Compute Stick and compares against this device, which deploys a custom accelerator, the Intel Movidius Myriad X Vision Processing Unit (VPU) [2]. The results show that better inference time and energy per inference result can be obtained with comparable accuracy. This is a striking result as the VPU is a dedicated accelerator touting ultralow power and high performance and serves to showcase the potential of quantized CNNs on FPGAs.

The final paper "Implementing and Evaluating an Heterogeneous, Scalable, Tridiagonal Linear System Solver with OpenCL to Target FPGAs, GPUs, and CPUs," addresses the problem of solving linear systems, a very common problem in scientific computing and HPC. As indicated by the title, the paper focuses in particular on diagonally dominant tridiagonal linear systems using the truncated-SPIKE algorithm [3] and presents a numerically stable optimised FPGA implementation using the open standard OpenCL [4]. The paper compares implementations of the algorithm on CPU, GPU, and FPGA as well as provides comparison against an optimised implementation of the TDMA solver [5]. The FPGA implementation is shown to have better performance per Watt than the CPU and GPU, and the truncated-SPIKE algorithm outperforms the TDMA algorithm on FPGA and CPU. The paper also demonstrates the potential of utilising FPGAs, GPUs, and CPUs concurrently in a heterogeneous computing environment to solve linear systems.

Conflicts of Interest

The editors declare that they have no conflicts of interest.

Acknowledgments

The editors wish to acknowledge the collaborative funding support from the UK EPSRC under grant P/L00058X/1.

Wim Vanderbauwhede
Sven-Bodo Scholz
Martin Margala

References

[1] A. Fijany and C. P. Williams, "Quantum wavelet transforms: fast algorithms and complete circuits," in *Proceedings of the NASA International Conference on Quantum Computing and Quantum Communications*, pp. 10–33, Springer, Palm Springs, CA, USA, February 1998.

[2] S. Rivas-Gomez, A. J. Pena, D. Moloney, E. Laure, and S. Markidis, "Exploring the vision processing unit as co-processor for inference," in *Proceedings of the IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 589–598, IEEE, Vancouver, Canada, May 2018.

[3] C. C. K. Mikkelsen and M. Manguoglu, "Analysis of the truncated spike algorithm," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 4, pp. 1500–1519, 2009.

[4] A. Munshi, "The opencl specification," in *Proceedings of the IEEE Hot Chips 21 Symposium (HCS)*, pp. 1–314, IEEE, Stanford, CA, USA, August 2009.

[5] D. J. Warne, N. A. Kelson, and R. F. Hayward, "Comparison of high level fpga hardware design for solving tri-diagonal linear systems," *Procedia Computer Science*, vol. 29, pp. 95–101, 2014.