Master Thesis proposal

Title: Hybrid Floating- and Fixed-Point Customized Arithmetic for Energy-Efficiency

Keywords: arithmetic operators, floating-point, fixed-point, high-level synthesis, floating-point to fixed-point conversion, design space exploration, optimization, accuracy analysis, power consumption, high-performance embedded systems, FPGA, ASIC

Supervisor: Olivier Sentieys <olivier.sentieys@inria.fr>, Silviu Filip <silviu.filip@inria.fr>
Laboratory: IRISA/INRIA – Cairn team
Place: Rennes (Campus de Beaulieu) or Lannion (ENSSAT)

Energy consumption is one of the major issues in computing today shared by all domains in computer science, from high-performance computing to embedded systems. The two main factors that influence energy consumption are the execution time and data volume. Execution time directly impacts energy, as energy is the product of time and (average) power. Another large source of energy consumption is data transfer and storage - the amount of energy consumed is directly proportional to the volume of data.

In the recent years, **approximate computing** has become a major field of research to improve both speed and energy consumption in embedded systems [1-2]. Many applications in embedded systems do not require high accuracy, and hardware designers often seek for a good balance between accuracy, speed, energy, and area cost. In addition to well-established applications (e.g. signal, image, vision, wireless communications), recent emerging applications (e.g. machine learning, data mining, web search) also exhibit the property to be inherently resilient to errors. Approximate or inexact calculation provides energy gains by exploiting the **trade-off between energy efficiency and application quality**. As an example, the gain in energy between a low-precision 8-bit operation suitable for vision and a 64-bit double-precision floating-point operation necessary for high-precision scientific computations, can reach up to 50x by considering storage, transport and computing of the data. The gain in energy efficiency (the amount of computations per Joule) is even larger. By relaxing the need for fully precise operations, it is therefore possible to improve energy efficiency substantially. Various techniques for approximate computing augment the design space by providing another set of design knobs for performance-accuracy trade-off. Controlling approximations requires methods not only at compile time but also at runtime. There is strong research activity on these topics in the Cairn team.

At compile-time, the aim is to minimize a cost function constrained by a given computation accuracy constraint. This requires methods (1) to evaluate the accuracy and (2) to estimate the cost, as a function of number representations (floating-point, fixed-point) and word-length (i.e. bit-width) of operations and data. Cost function can include energy consumption and execution time, as well as resources used for hardware implementation or code size for software implementation. In this work, we will mainly consider hardware implementation (FPGA, ASIC) with C++-based design using high-level synthesis (HLS).

Choosing the right computation precision at runtime, while preserving the application functionality in reasonable bounds, is another promising approach to improve energy

efficiency significantly. Nevertheless, analysing and managing the quantity errors during execution is a complex problem.

The objective of this Master Thesis is to define a method to explore, for a given application, specified using C code, different number representations in order to find near-optimal solutions minimizing cost while preserving the application functionality in reasonable bounds.

The challenge is hence to find the good trade-off between the numerical accuracy, dynamic range and the implementation cost. We will consider fixed-point and floating-point arithmetic [8]. Floating-point data types can be customized by adjusting the word-length of the exponent and the mantissa part. Another objective is to study systems mixing fixed-point and floating-point in the same application.

- A library of operators with different word-lengths is already available for both representations [6] [8]. This library provides area, energy and delay of arithmetic operators based on fixed-point for various word-lengths, and on floating-point for various exponent and mantissa lengths. There are already commercial platforms (Nvidia, Google, Intel) that mix 32-bit high precision floating-point computing with low precision 16-bit formats (or even recently 8-bit) for increased performance [11-13].
- We will rely on a set of signal and image processing benchmarks and machine learning applications, that will be used as vehicle to provide results on (1) quality degradations and (2) energy efficiency advantage, with reduced precision. Metrics like average error power, error bounds or distribution can be considered. As we already studied a lot signal and image [9], the main focus will be on machine learning applications such as deep neural networks.
- Quality degradation could be obtained through simulations by automatically instrumenting the code to modify data types, verify correctness and launch simulations to explore solutions. However, exploration with simulations can quickly become time prohibitive. So, the first objective is to study a method to evaluate the accuracy of an application based on static analysis. Based on the distributions of values of the different data processed in an application, we believe that it is possible to define rules to (1) choose the best representation for operations and data (float, fix) and (2) determine the word-lengths of computation (integer and fractional parts for fixed-point, exponent and mantissa for floating-point). It is mainly an optimization problem. This Master thesis will therefore be dedicated to some preliminary work on this topic, towards word-length optimization [9] methods for interbred design mixing floating- and fixed-point arithmetic.
- The second objective is to obtain cost (energy, delay, area) for application with and without word-length optimization. For this part, results will be achieved using high-level synthesis from C++ and synthesis scripts with different timing constraints, to provide us with a set of accuracy-cost, near-optimal solutions, for FPGA and ASIC targets.

Some tools [6,10] previously developed in the Cairn team at Inria will also be used for automatic floating-point to fixed-point conversion, operator characterization, custom floating-point, and HLS design. A follow-up of this work as a PhD thesis will be possible.

Note on the bibliography: For the bibliography part of this Master thesis, first floating-point arithmetic number representation and operators will be studied from the literature as well as some research papers on customized floating-point arithmetic. The project will consist in the analysis of the accuracy on some customized floating-operators (e.g. 6-bit exponent and 4-bit mantissa) when running an application (e.g. image denoising) as well their area, power, clock frequency when compared to IEEE 754 single-precision 32-bit floating-point.

References

[1] Oh, that's near enough computing: Letting microchips make a few mistakes here and there could make them much faster and more energy-efficient, The Economist, June 2012, http://www.economist.com/node/21556087
[2] S.-L. Lu, "Speeding up processing with approximation circuits," Computer, vol. 37, no. 3, pp. 67-73, 2004.
[3] Florent de Dinechin and Bogdan Pasca. Designing custom arithmetic data paths with FloPoCo. IEEE Design & Test of Computers, 28(4):18--27, July 2011.
[4] J. Kim and S. Tiwari, "Inexact computing for ultra low-power nanometer digital circuit design," in IEEE/ACM Intl. Symp. on Nanoscale Architectures, pp. 24-31, 2011.
[5] D. Menard, R. Rocher, and O. Sentieys. Analytical Fixed-Point Accuracy Evaluation in Linear Time-Invariant Systems. IEEE Transactions on Circuits and Systems I: Regular Papers, 55(10):3197-3208, 2008.
[6] Benjamin Barrois, Olivier Sentieys, and Daniel Menard. The Hidden Cost of Functional Approximation Against Careful Data Sizing – A Case Study. In Design, Automation & Test in Europe Conference & Exhibition (DATE), Lausanne, France, 2017.
[7] Rengarajan Ragavan, Benjamin Barrois, Cedric Killian, and Olivier Sentieys. Pushing the Limits of Voltage Over- Scaling for Error-Resilient Applications. In Design, Automation & Test in Europe Conference & Exhibition (DATE), Lausanne, Switzerland, March 2017.
[8] Benjamin Barrois and Olivier Sentieys. Customizing Fixed-Point and Floating-Point Arithmetic - A Case Study in K-Means Clustering. In IEEE International Workshop on Signal Processing Systems (SiPS), page 6, October 2017.
[9] Van-Phu Ha, Tomofumi Yuki, and Olivier Sentieys. Towards Generic and Scalable Word-Length Optimization. IEEE/ACM Design Automation and Test in Europe (DATE), Grenoble, France, March 2020.
[10] https://gitlab.inria.fr/gecos/gecos-float2fix
[11] NVIDIA. (2019) Automatic Mixed Precision for Deep Learning. Available: https://developer.nvidia.com/automatic-mixed-precision
[12] Google. (2019) Using bfloat16 with TensorFlow models. Available: https://cloud.google.com/tpu/docs/bfloat16
[13] Intel. (2018) BFLOAT16: hardware numerics definition. Available: https://software.intel.com/sites/default/files/managed/40/8b/bf16-hardware-numerics-definition-white-paper.pdf