A Mixed-Precision RISC-V Processor for Extreme-Edge DNN Inference

Gianmarco Ottavi†, Angelo Garofalo†, Giuseppe Tagliavini †, Francesco Conti†*, Luca Benini†* and Davide Rossi†
DEI, University of Bologna, Italy† *IIS lab, ETH Zurich, Switzerland**
{gianmarco.ottavi2, davide.rossi, angelo.garofalo, giuseppe.tagliavini}@unibo.it
{fconti, lbenini}@iis.ee.ethz.ch

***Abstract*—Low bit-width Quantized Neural Networks (QNNs) enable deployment of complex machine learning models on constrained devices such as microcontrollers (MCUs) by reducing their memory footprint. Fine-grained asymmetric quantization (i.e., different bit-widths assigned to weights and activations on a tensor-by-tensor basis) is a particularly interesting scheme to maximize accuracy under a tight memory constraint [1]. However, the lack of sub-byte instruction set architecture (ISA) support in SoA microprocessors makes it hard to fully exploit this extreme quantization paradigm in embedded MCUs. Support for sub-byte and asymmetric QNNs would require many precision formats and an exorbitant amount of opcode space. In this work, we attack this problem with status-based SIMD instructions: rather than encoding precision explicitly, each operand's precision is set dynamically in a core status register. We propose a novel RISC-V ISA core *MPIC* (Mixed Precision Inference Core) based on the open-source RI5CY core. Our approach enables full support for mixed-precision QNN inference with different combinations of operands at 16-, 8-, 4- and 2-bit precision, without adding any extra opcode or increasing the complexity of the decode stage. Our results show that MPIC improves both performance and energy efficiency by a factor of 1.1–4.9× when compared to software-based mixed-precision on RI5CY; with respect to commercially available Cortex-M4 and M7 microcontrollers, it delivers 3.6–11.7× better performance and 41–155× higher efficiency.**

***Index Terms*—PULP Platform, Embedded-Systems, Deep Neural Networks, Mixed-precision, Microcontroller**

I. Introduction

Running complex applications on embedded systems like microcontrollers (MCUs) requires optimization on both software and hardware due to severe constraints in terms of memory size, power consumption, and computing power. In an Internet-of-Things (IoT) environment, wireless communication to higher-level nodes often dominates the power budget. Algorithms such as Deep Neural Networks (DNNs), more specifically Convolutional Neural Networks (CNNs) which are state-of-the-art for computer vision and speech recognition, are used in computing at the edge of IoT to reduce the amount of data to transmit by communicating only classes or high-level features instead of the raw sensor data. The complexity of these algorithms typically requires millions of Multiply-Accumulate (MAC) operations and significant memory footprint, where memory is a valuable resource due to its cost in terms of area and power.

An effective way to reduce the memory footprint of DNNs is *quantization*, a technique that reduces inputs and weights to fixed-point formats such as 8- bits, and even sub-byte like 4- and 2- bits [1]–[3]. Banner *et al.* proposed a methodology to quantize both weights and activations to 4-bit with an accuracy drop of only a few percent, not modifying the training and not requiring a full dataset. Rusci *et al.* [1] show how, using mixed-precision quantization for each layer, it is possible to reduce by up to 7× the memory footprint of DNNs, incurring only in a 4% accuracy loss. However, although quantization provides a clear reduction of memory bandwidth visible also in general-purpose processors [4], much of the inference-time benefit is accessible only through customized hardware accelerators [5] or with an FPGA implementation of quantized arithmetic units [6]. To the best of the authors' knowledge, the only recent work taking advantage of quantized formats in software processors is the one presented by Anderson *et al.* [7]. It proposed a software technique exploiting arbitrary bit-precise signed and unsigned integer operations embedding a vector architecture with custom bit-width lanes in fixed-width scalar arithmetic [7]. However, this comes with significant effort in application porting.

From the hardware perspective, the only relevant research work in this field is the reconfigurable Parallel Balanced-Bit-Serial (PBBS) vector processing tile presented by Wu *et al.* [8], which is suitable for improving the efficiency of sub-byte single instruction multiple data (SIMD) computations of heavily leakage-dominated ULP designs. However, code serialization significantly degrades performance and efficiency in near- and super-threshold operating points. On the other hand, the totality of commercial MCUs operates at the finest granularity of 1-byte data [9], [10]. The new ARM [11] ISA specialized for machine learning, implemented by the *Cortex M55* processor, enhances the ARMv8 with extensions similar to the ones presented in [12], such as 8-bit SIMD instructions, loops, and conditional execution extensions. In addition, it provides pipelined execution of load and mac instructions [11] that allows maximizing utilization of MAC units during the execution of regular patterns (e.g., convolutions).

However, similarly to all other commercial cores, the ARM *Cortex M55* does not support natively smaller than 8-bit SIMD instructions. Hence, data have to be presented as a byte for computation, even if it is "packed" in a more compact representation. First, this means that there is no way to exploit the additional parallelism because the datapath is hardwired to 8 bits. Second, in the tight inner loops of the quantized DNN kernels, the cost of unpacking and packing data can be extremely high, leading to up to 2.5× worse performance than directly using 8-bit data, as shown in the results. In our experiments, sub-byte and mixed-precision quantization by itself improved only the implementation feasibility of a network in MCUs (in term of squeezing the network memory footprint),

This work was supported in part by OPRECOMP (Open transPREcisionCOMPuting) Grant Agreement No. 732631, and WiPLASH (Wireless Plas-ticity for Heterogeneous Massive Computer Architectures) Grant AgreementNo. 863337. Both projects are funded from the European Union's Horizon 2020 research and innovation program.

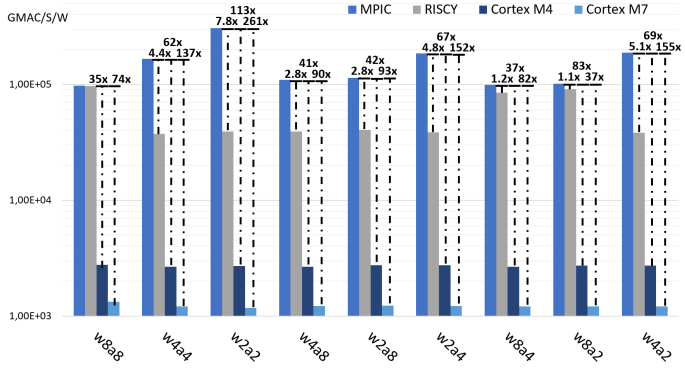

Fig. 7. Energy efficiency expressed in GMAC/s/W.

Significantly, mixed-precision QNN kernels also do not suffer any performance hit, thanks to unpacking done in hardware. The performance of 8x4 and 8x2 kernels are close to the 8-bit uniform kernel, likewise for the 4-bit one. This is because the selection of the dotp module (Fig. 5) is tied to the size of the greater operand (e.g., 8x4 uses 8-bit multipliers). However, we can see that the performance is slightly better than their equivalent uniform case, thanks to the higher operational intensity. If we perform a mixed-precision 8x4 operation, operand b needs to fetch fewer data from memory, since its register can hold twice as many operands as the register containing a. Another factor that impacts mixed-precision operation is the quantization process (*QntPack*). Focusing on the chart for activations of 4- and 2-bit, the performance is marginally worse than when we have 8-bit activations.

In contrast with performance in MAC/cycle, energy efficiency (expressed in GMAC/s/W) takes into account also physical design parameters such as the fabrication technology and the operating voltage and frequency. For the Cortex M7 and M4, we used an implementation from ST-Microelectronics consuming $\sim$234 mW at 480 MHz [19] and 10 mW at 80 MHz [20], respectively; while we used the power consumption figures reported in Table II for the RISC-V SoCs. In Figure 7, we can see that the lower performance of the Cortex M7 is emphasized even more by the technology factor, having a peak of 1.27 GMAC/s/W and being from 74x to 255x less efficient in these workloads compared to MPIC. The Cortex M4 is way more efficient than the Cortex M7 but still falls short when compared to RISC-V cores, being from 35x to 113x less efficient. For the RI5CY core, we have a slight disadvantage of 1% only in the 8-bit case, while in all other scenarios, the results are qualitatively similar to the performance ones.

V. Conclusion

In this work, we presented an alternate way to deal with a saturated encoding space. We extended the ISA to support sub-byte and mixed-precision formats aiming at improving the performance of QNN via removing the overhead caused by unpacking data before computation. The MPIC-based SoC implementation resulted in an area overhead of 11% when compared to the baseline core while having a negligible impact on frequency and power and so not compromising the general-purpose nature of the RI5CY core. The performance gain ranges from 1.1× to 7.7× when compared to the baseline during the execution of a QNN layer, and from 3.6× up to 19.3× in regard to the Cortex M7 and M4. The energy efficiency peaks at 303 GMAC/s/W for the 2-bit convolution and ranges from one to two orders of magnitude higher when compared with ARM counterpart, providing a solution that is considerably more efficient than commercially available MCUs solutions for QNN inference.

References

[1] M. Rusci, A. Capotondi, and L. Benini, "Memory-driven mixed low precision quantization for enabling deep network inference on micro-controllers," *arXiv preprint arXiv:1905.13082*, 2019.

[2] B. Moons, K. Goetschalckx, N. Van Berckelaer, and M. Verhelst, "Minimum energy quantized neural networks," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2017, pp. 1921–1925.

[3] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6869–6898, 2017.

[4] A. Stojanov, T. M. Smith, D. Alistarh, and M. Püschel, "Fast quantized arithmetic on x86: Trading compute for data movement," in *2018 IEEE International Workshop on Signal Processing Systems (SiPS)*, Oct 2018, pp. 349–354.

[5] B. Moons and M. Verhelst, "An energy-efficient precision-scalable convnet processor in 40-nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 4, pp. 903–914, April 2017.

[6] J. Qiu, J. Wang, S. Yao, K. Guo, B. Li, E. Zhou, J. Yu, T. Tang, N. Xu, S. Song, and et al., "Going deeper with embedded fpga platform for convolutional neural network," in *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 26–35. [Online]. Available: https://doi.org/10.1145/2847263.2847265

[7] A. Anderson, M. Doyle, and D. Gregg, "Scalar arithmetic multiple data: Customizable precision for deep neural networks," in *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)*, June 2019, pp. 61–68.

[8] B. Wu and I. Wey, "Parallel balanced-bit-serial design technique for ultra-low-voltage circuits with energy saving and area efficiency enhancement," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 1, pp. 141–153, Jan 2018.

[9] ARM, "Arm architecture reference manual armv8," 2013-2020, https://developer.arm.com/docs/ddi0487/latest/arm-architecture-reference-manual-armv8-for-armv8-a-architecture-profile.

[10] U. o. C. B. Andrew Waterman; Krste Asanovi; SiFive Inc., CS Division; EECS Department, "The risc-v instruction set manual, volume i: User-level isa," April 2019.

[11] D. E. Joseph Yiu, "Introduction to the arm cortex-m55 processor. available online: https://pages.arm.com/cortex-m55-introduction.html," February 2020.

[12] M. Gautschi, P. D. Schiavone, A. Traber, I. Loi, A. Pullini, D. Rossi, E. Flamand, F. K. Gürkaynak, and L. Benini, "Near-threshold risc-v core with dsp extensions for scalable iot endpoint devices," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 10, pp. 2700–2713, Oct 2017.

[13] A. Garofalo, M. Rusci, F. Conti, D. Rossi, and L. Benini, "Pulp-nn: accelerating quantized neural networks on parallel ultra-low-power risc-v processors," *Philosophical Transactions of the Royal Society A*, vol. 378, no. 2164, p. 20190155, 2020.

[14] O. Azizi, A. Mahesri, B. C. Lee, S. J. Patel, and M. Horowitz, "Energy-performance tradeoffs in processor architecture and circuit design: a marginal cost analysis," *ACM SIGARCH Computer Architecture News*, vol. 38, no. 3, pp. 26–36, 2010.

[15] P. D. Schiavone, D. Rossi, A. Pullini, A. Di Mauro, F. Conti, and L. Benini, "Quentin: an ultra-low-power pulpissimo soc in 22nm fdx," in *2018 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*. IEEE, 2018, pp. 1–3.

[16] D. Rossi, F. Conti, A. Marongiu, A. Pullini, I. Loi, M. Gautschi, G. Tagliavini, A. Capotondi, P. Flatresse, and L. Benini, "Pulp: A parallel ultra low power platform for next generation iot applications," in *2015 IEEE Hot Chips 27 Symposium (HCS)*, Aug 2015, pp. 1–39.

[17] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan, "Pact: Parameterized clipping activation for quantized neural networks," *arXiv preprint arXiv:1805.06085*, 2018.

[18] L. Lai, N. Suda, and V. Chandra, "Cmsis-nn: Efficient neural network kernels for arm cortex-m cpus," *arXiv preprint arXiv:1801.06601*, 2018.

[19] STMicroelctronics, "Stm32h743 datasheet," 2018, https://www.st.com/resource/en/datasheet/stm32h743bi.pdf.

[20] STMicroelctronics, "Stm32l476 datasheet," 2018, https://www.st.com/resource/en/datasheet/stm32l476je.pdf.