A Hardware/Software Co-Design Vision for Deep Learning at the Edge

Flavio Ponzina, Simone Machetti, Marco Rios, Benoît Walter Denkinger, Alexandre Levisse, Giovanni Ansaloni, Miguel Peón-Quirós, and David Atienza, *École Polytechnique Fédérale de Lausanne (EPFL), 1015, Lausanne, Switzerland*

The growing popularity of edgeAI requires novel solutions to support the deployment of compute-intense algorithms in embedded devices. In this article, we advocate for a holistic approach, where application-level transformations are jointly conceived with dedicated hardware platforms. We embody such a stance in a strategy that employs ensemble-based algorithmic transformations to increase robustness and accuracy in convolutional neural networks, enabling the aggressive quantization of weights and activations. Opportunities offered by algorithmic optimizations are then harnessed in domain-specific hardware solutions, such as the use of multiple ultra-low-power processing cores, the provision of shared acceleration resources, the presence of independently power-managed memory banks, and voltage scaling to ultra-low levels, greatly reducing (up to 60% in our experiments) energy requirements. Furthermore, we show that aggressive quantization schemes can be leveraged to perform efficient computations directly in memory banks, adopting in-memory computing solutions. We showcase that the combination of parallel in-memory execution and aggressive quantization leads to more than 70% energy and latency gains compared to baseline implementations.

The rise and ever-improving accuracy of artificial intelligence (AI) is fostering a revolution in a multitude of scenarios, ranging from healthcare to manufacturing. Still, this impressive rise in performance has been fueled by a concurrent increase in complexity.[1] For example, the state-of-the-art AI methods for object recognition and automated translation require a workload in the order of floating-point operations giga floating-point operations (10^9 floating-point operations) for each inference.

Such computational requirements strain the capabilities of digital architectures, especially when considering edge applications where processing is performed entirely or in part at the edge, where devices are typically constrained in terms of computing and memory capabilities. Indeed, a vast number of hardware and software solutions for improving the energy, runtime, and memory efficiency of AI algorithms have been recently proposed.[2–4] Nonetheless, hardware and software aspects are often considered in isolation. Instead, we advocate for combining hardware-friendly application optimization strategies and software-friendly architectural solutions to achieve disruptive efficiency gains.

The framework depicted in Figure 1 embodies such a stance. It receives as input a convolutional neural network (CNN) architecture designed (or selected from the state-of-the-art) to achieve the desired classification accuracy on the target dataset. As for software optimizations [see Figure 1(I)], we first consider resource-constrained ensembles, which increase accuracy and robustness against sources of internal noise (e.g., memory errors due to subnominal operating conditions or approximation due to operands' quantization). Then, this higher resiliency opens the path to aggressive quantization, which reduces memory requirements and improves efficiency [see Figure 1(II)]. Dedicated hardware resources exploit software optimizations. The parallelism exposed by ensembles allows their mapping and execution on

Digital Object Identifier 10.1109/MM.2022.3195617
Date of publication 1 August 2022; date of current version 28 October 2022.

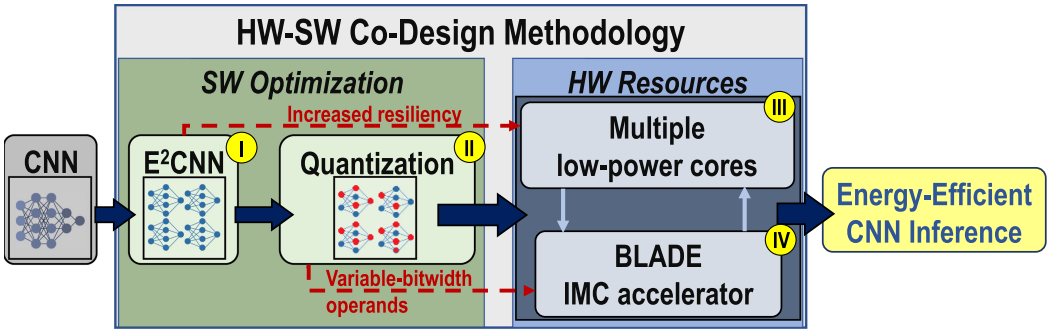

FIGURE 1. Overview of the hardware/software co-design framework. Left: application-level optimizations: ensembling (I) and quantization (II). These are leveraged to drive the design of domain-specific platforms featuring heterogeneous multicores (III) and IMC capabilities (IV).

platforms featuring multiple ultra-low-power cores [see Figure 1(III)]. Similarly, the presence of multiple, independently power-managed banks opens the opportunity for efficient in-memory computation [see Figure 1(IV)].

In the rest of this article, we detail our proposed strategy. We cover software-level optimizations in the "Resource-aware application optimization" section. Then, we describe how these can be effectively exploited in the design of domain-specific hardware for edgeAI in the "Domain-specific hardware" section and the "IMC: Bit-line accelerator for devices on the edge (BLADE)" section.

RESOURCE-AWARE APPLICATION OPTIMIZATION

Application-level optimization methodologies aim at modifying the structure of CNNs to build models with increased accuracy *and* efficiency.

Toward this goal, Ponzina et al.[2] introduced embedded ensembles of CNNs [E^{2}CNNs; see Figure 1(I)]. To build E^{2}CNNs, the filters of an untrained CNN architecture are first pruned to obtain a model with lower memory and computing requirements. The obtained structure is then replicated, hence deriving models composed of multiple, but lightweight, instances (see Figure 2). Afterward, each instance is independently trained starting from different initial weight values. E^{2}CNNs can also reduce storage requirements when the pruning factor exceeds replication. For example, pruning GoogLeNet by 8x to build an E^{2}CNNs implementation composed of just four instances halves the memory and computational requirements and reduces energy cost by 55%, without any degradation in accuracy when evaluated on the CIFAR100 dataset.

The accuracy and resiliency improvements of E^{2}CNNs support a synergic use of additional optimization approaches. First, the robustness of E^{2}CNNs is exploited by aggressive quantization schemes [see Figure 1(II)]. Indeed, in Ponzina et al.,[3] a strategy is described to aggressively reduces the width of activations and weights in convolutional and fully connected (FC) layers. This approach, summarized in Figure 3, is based on a greedy heuristic that, at each iteration, selects a layer in which the bitwidth should be reduced based on a measure of sensitivity and on its size (since quantizing larger layers achieves greater gains). The baseline model (a) is heterogeneously quantized, reducing the bitwidth of weights in convolutional layers and activations in FC layers while meeting a user-defined accuracy level (b). Then, convolutional filters composed of only 0-valued weights are pruned from the model (c), resulting in significant memory and energy savings with no impact on accuracy. Finally, to improve data-level parallelism, the bitwidth of FC weights and convolutional activations is selectively reduced (d). The resulting heterogeneous and fine-grained quantization schemes can be effectively implemented in in-memory computing (IMC) accelerators, resulting in notable energy gains and very limited accuracy degradations. The energy gains of our approach are discussed in the "IMC: Bit-Line

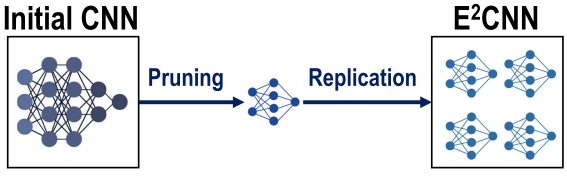

FIGURE 2. In E^{2}CNNs, a CNN model is first pruned. Then, it is replicated several times to build up an ensemble that meets the same memory and computational requirements of the original model.

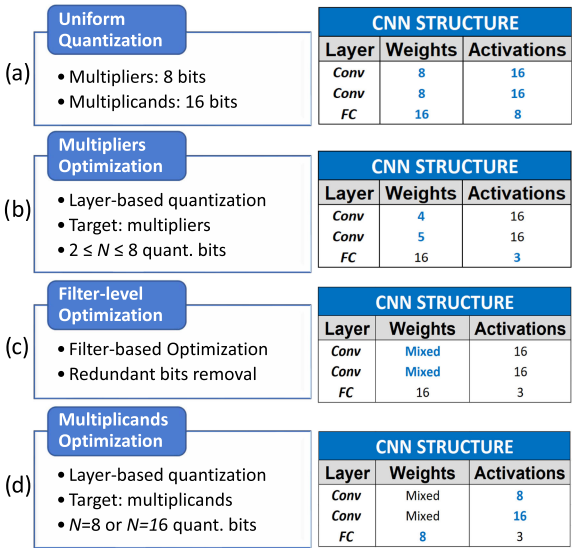

FIGURE 3. Workload-aware quantization and pruning methodology (left). Running example showing how the bitwidth of weights and activations are optimized in different steps (right). (a) Baseline quantization level. (b) Heterogeneous multipliers' quantization. (c) Filter-level optimization, where filters having all 0-valued weights are removed. (d) Multiplicands' quantization enabling in-memory SIMD.

Accelerator for Devices on the Edge (BLADE)" section, where an IMC accelerator supporting the described algorithmic optimization is presented.

Furthermore, ensembles of CNNs exhibit a high degree of robustness toward memory errors, because the instances composing the ensemble exhibit varying weight distributions due to their separate training. Hence, memory errors having a critical impact on the accuracy of one instance may have a significantly lower influence on the others, thus increasing the probability of returning the correct output. The increased resiliency of E^{2}CNNs enables scaling of the supply voltage while tolerating the ensuing error probability when accessing static random-access memory (SRAM) banks. In Ponzina et al.,[2] experiments on different benchmarks demonstrate that voltage scaling can increase energy efficiency up to 60% without appreciable impact accuracy.

DOMAIN-SPECIFIC HARDWARE

The parallelization of the computing and memory subsystems is a key to reducing the energy budget of edgeAI platforms. By using multiple processors, shallower in-order pipelines based on reduced and modular instruction sets (e.g., reduced instruction set computer-V) can be employed in conjunction with dedicated

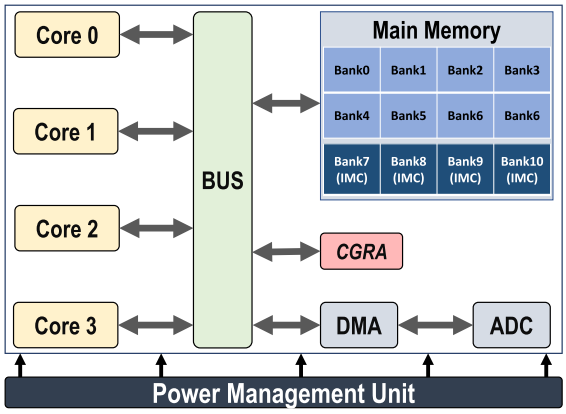

FIGURE 4. Architecture template for edgeAI platforms, including multiple cores, an independent input/output system, a multibanked memory supporting IMC, a reconfigurable CGRA accelerator, and a fine-grained power management unit.

components [e.g., direct memory access (DMA) and accelerators]. Such an approach effectively constrains energy without overly sacrificing performance, giving the flexibility to adapt to varying workloads at run-time. For example, when only signal acquisition is performed, solely the analog-to-digital converter (ADC) components and the DMA transferring data to memory banks are required, while processors and accelerators can be power gated. Moreover, clock gating can be employed to harvest energy-saving opportunities over short time intervals. As an example, cores and accelerators can be clock-gated during synchronization events.

Similarly, dividing the memory into small banks enables energy-saving opportunities. Banks can be individually powered off or put in retention mode when unused, hence increasing efficiency. Moreover, in-memory operations can be supported in multibanked memories with a high degree of run-time parallelism with limited area overhead, as detailed in the "IMC: Bit-Line Accelerator for Devices on the Edge (BLADE)" section.

A high-level block scheme of an architecture implementing the abovementioned features is depicted in Figure 4. It features multiple cores to cope with the high workloads of AI applications and several memory banks that can be independently powered off, possibly supporting IMC capabilities. The template architecture also includes flexible coarse-grained reconfigurable arrays (CGRAs), thus enabling the hardware acceleration of computational kernels, as showcased in Giovanni et al.,[12] where energy gains up to 32% are achieved compared to an equivalent single-core system.

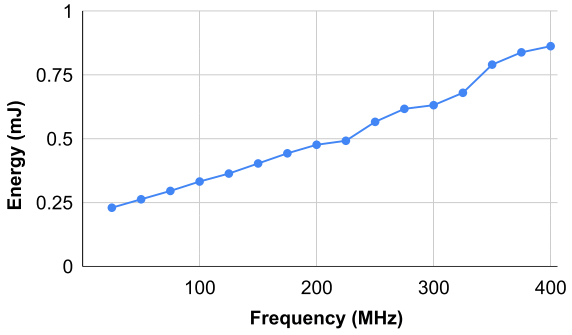

FIGURE 5. Change in the normalized energy consumption of a CNN inference for different frequency constraints imposed during synthesis.

Note that hardware-friendly software optimizations presented in the "Resource-aware application optimization" section can efficiently be included in this architecture. CNN instances composing the ensemble can be easily mapped on different cores, which selectively activate memory banks only when needed. The lower workload in each core can then be exploited to reduce the operating frequency (and therefore energy) while abiding to performance constraints, allowing the scaling of the voltage supply.

Although aggressive voltage reduction is possible as digital logic is error-resilient down to the technology voltage threshold, memories (e.g., SRAM cells) usually start failing at higher voltages, hence posing a limit to voltage scaling. The impact of memory errors due to voltage scaling on CNN accuracy has been studied in Denkinger et al.[11] and Ponzina et al.,[2] showing that ensembling improves the robustness of CNNs, allowing SRAM memories to operate at subnominal voltages while coping with the ensuing errors. These works show energy savings in memories of up to 90% due to voltage scaling while limiting CNN output quality degradation caused by memory errors to just 1%.

The implementation process also plays a role in energy efficiency. Hardware can be optimized at synthesis time by matching the system performance and power consumption to the demands of the target applications using multi-Vt libraries. Such libraries enable low-power and high-performance cells to be instantiated as required to meet timing constraints. Indeed, Figure 5 shows how the normalized energy consumption required to execute a CNN inference varies when different maximum operating frequencies are imposed.

IMC: BIT-LINE ACCELERATOR FOR DEVICES ON THE EDGE (BLADE)

Enabling computation inside SRAM memory banks is particularly appealing for edgeAI workloads, which are dominated by convolutions or other forms of matrix–matrix and matrix–vector multiplications. The high regularity of these operations in terms of access patterns enables ultraefficient IMC solutions.

IMC architectures can employ technologies ranging from emerging nonvolatile memories (eNVM) to traditional complementary metal–oxide–semiconductor (CMOS)-based memories. IMC based on eNVMs, such as resistive random-access memories, phase change memories, and magnetic random-access memories, can be arranged in cross-points with high integration density. However, these IMC methods rely on nonconventional fabrication processes, complex periphery circuitry including ADCs, and high write currents. On the other hand, IMC using SRAM memories 1) takes advantage of a well-known fabrication process and 2) can be operated as digital devices, with little additional logic at the periphery of memory cell arrays compared to the regular SRAM memories.

ENABLING COMPUTATION INSIDE SRAM MEMORY BANKS IS PARTICULARLY APPEALING FOR edgeAI WORKLOADS, WHICH ARE DOMINATED BY CONVOLUTIONS OR OTHER FORMS OF MATRIX–MATRIX AND MATRIX–VECTOR MULTIPLICATIONS.

Moreover, by relying on SRAMs and due to their very low circuit overhead, SRAM-based IMC architectures can be drop-down replacements for traditional memory banks. Hence, they can leverage the same system-level optimization: they can be power-gated when not used or put in retentive mode when no accesses are performed.

One notable SRAM-based IMC architectural solution is BLADE.[4] BLADE enables *in situ* arithmetic operations and nether rely on analog elements, nor on associated ADCs and digital-to-analog converters. Its circuit-level implementation is compatible with high-density 6T-SRAM bitcells, thanks to an organization of memory cells in Local Groups. Such characteristics make BLADE compatible with a large range of supply voltages and enable an aggressive voltage/frequency scaling, as shown in Figure 6(a).

In BLADE, operations are performed by simultaneously activating two word lines of different local groups. IMC operations are performed on the global

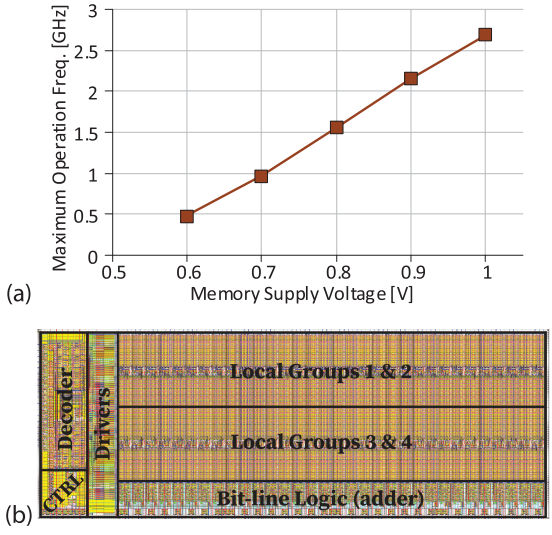

FIGURE 6. (a) Variability-aware performances (GHz) of IMC operations in BLADE simulated in a 28-nm CMOS technology. (b) BLADE 2KiB subarray physical layout in 65-nm technology as integrated in Darkside[5] with the composing blocks highlighted.

bit-lines and evaluated by conventional single-ended sense amplifiers. Operations such as additions, subtractions, logic shifts, and bitwise operations can be performed in the memory periphery. By chaining additions and shifts, multiply-and-accumulate (MAC) operations can also be implemented. As convolutional and FC layers of CNNs are composed of MAC operations, they can be executed with very high efficiency in a single-instruction multiple-data fashion on the subarrays composing each BLADE bank, as showcased in Ponzina et al.[3]

BLADE's performance is further increased when low-bitwidth quantization schemes are adopted. Indeed, in SRAM-based IMC architectures, the number of clock cycles required to execute a multiplication is proportional to the bitwidth of the multiplier. Therefore, the application-level strategy described in the "Resource-Aware Application Optimization" section can be effectively harnessed by executing the resulting heterogeneously quantized ensembles in BLADE. Results considering a single-instance implementation are summarized in Figure 7. They show energy (and latency) improvements of 72% with just 1% accuracy degradation compared to a homogeneously 8-bit single-instance CNN.

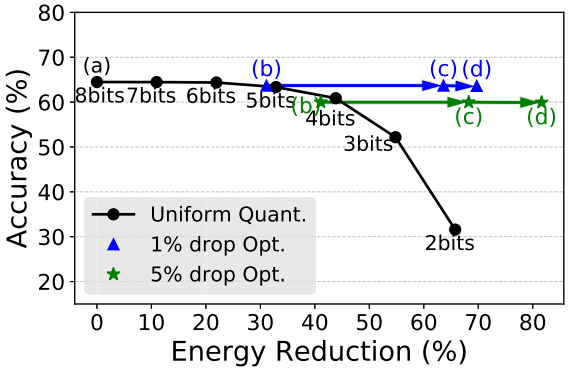

FIGURE 7. Accuracy of MobileNet-v2[7] on the CIFAR-100 dataset at different energy optimization levels in homogeneously quantized CNNs (black) and our optimized CNNs for a 1% (black) and a 5% (green) user-defined accuracy thresholds. Energy is measured for a BLADE implementation in 28-nm CMOS technology. (a)–(d) refer to the optimization steps in Figure 3.

CONCLUSION

In this article, we have discussed the importance of a comprehensive co-design approach for edgeAI, where algorithmic optimizations and hardware architectures are jointly designed. We have shown that very significant energy efficiency gains can be obtained when application-level optimizations are well supported by hardware resources. Embodying this paradigm, we have presented ensembling as a key optimization strategy that improves robustness against aggressive quantization schemes and memory errors. Such characteristics are harnessed by a domain-specific edgeAI system, which supports parallel execution on multiple ultra-low-power cores, and aggressive voltage scaling. In addition, we have shown that the heterogeneous quantization CNNs can be effectively leveraged by IMC architectures, and that these can seamlessly integrate into multicore and multibanked systems. The presented edgeAI co-design framework achieves up to 60% energy reduction in the memory subsystem thanks to voltage scaling. In addition, the IMC accelerator exploits application-level optimizations to improve inference performance and efficiency by 72%, without a significant output quality degradation.

ACKNOWLEDGMENTS

This work was supported by the ERC Consolidator Grant COMPUSAPIEN (GA No. 725657), the EC H2020 WiPLASH (GA No. 863337), the EC H2020 FVLLMONTI (GA No. 101016776), and the Swiss NSF ML-Edge (GA No. 200020_182009) projects.

REFERENCES

1. S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64270–64277, 2018.
2. F. Ponzina, M. Peón-Quirós, A. Burg, and D. Atienza, "E2CNNs: Ensembles of convolutional neural networks to improve robustness against memory errors in edge-computing devices," *IEEE Trans. Comput.*, vol. 70, no. 8, pp. 1199–1212, Aug. 2021.
3. F. Ponzina, M. Rios, G. Ansaloni, A. Levisse, and D. Atienza, "A flexible in-memory computing architecture for heterogeneously quantized CNNs," in *Proc. IEEE Comput. Soc. Annu. Symp.*, 2021, pp. 164–169.
4. W. Simon, Y. M. Qureshi, M. Rios, A. Levisse, M. Zapater, and D. Atienza, "BLADE: An in-cache computing architecture for edge devices," *IEEE Trans. Comput.*, vol. 69, no. 9, pp. 1349–1363, Sep. 2020.
5. Darkside, 2021. [Online]. Available: http://asic.ethz.ch/2021/Darkside.html
6. M. Gautschi et al., "Near-threshold RISC-V core with DSP extensions for scalable IoT endpoint devices," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 10, pp. 2700–2713, Oct. 2017.
7. A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
8. L. Duch, S. Basu, R. Braojos, G. Ansaloni, L. Pozzi, and D. Atienza, "HEAL-WEAR: An ultra-low power heterogeneous system for bio-signal analysis," *IEEE Trans. Circuits Syst. I, Reg. Papers.*, vol. 64, no. 9, pp. 2448–2461, Sep. 2017.
9. P. Davide Schiavone et al., "Slow and steady wins the race? A comparison of ultra-low-power RISC-V cores for Internet-of-Things applications," in *Proc. 27th Int. Symp. Power Timing Model., Optim., Simul.*, 2017, pp. 1–8.
10. A. Pullini, D. Rossi, I. Loi, G. Tagliavini, and L. Benini, "Mr. Wolf: An energy-precision scalable parallel ultra low power SoC for IoT edge processing," *IEEE J. Solid-State Circuits*, vol. 54, no. 7, pp. 1970–1981, Jul. 2019.
11. B. W. Denkinger et al., "Impact of memory voltage scaling on accuracy and resilience of deep learning based edge devices," *IEEE Des. Test*, vol. 37, no. 2, pp. 84–92, Apr. 2020.
12. D. Giovanni et al., "Modular design and optimization of biomedical applications for ultra-low power heterogeneous platforms," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 39, no. 11, pp. 3821–3832, Nov. 2020.

FLAVIO PONZINA is a Ph.D. student with the Embedded Systems Laboratory, École Polytechnique Fédérale de Lausanne, 1015, Lausanne, Switzerland. His research interests include low power architectures and AI-based systems optimization. Ponzina received an M.Sc. degree in computer engineering from Politecnico di Torino, Turin, Italy. Contact him at flavio.ponzina@epfl.ch.

SIMONE MACHETTI is working toward a Ph.D. degree in electrical engineering with the Embedded Systems Laboratory, École Polytechnique Fédérale de Lausanne, 1015, Lausanne, Switzerland. His research interests include hardware and software co-design for ultra-low-power embedded devices and artificial intelligence algorithms for the Internet of Things. Machetti received an M.Sc. degree in computer engineering from the Politecnico di Torino, Turin, Italy. Contact him at simone.machetti@epfl.ch.

MARCO RIOS is a Ph.D. student with the Embedded Systems Laboratory, École Polytechnique Fédérale de Lausanne, 1015, Lausanne, Switzerland. His research interests include design of integrated systems and circuits, in-SRAM computing and the system impact of emerging memories. Rios received an M.Sc. degree in computer science and electronics for embedded systems from Université Grenoble Alpes, Grenoble, France. Contact him at marco.rios@epfl.ch.

BENOÎT WALTER DENKINGER is working toward a Ph.D. degree with the Embedded Systems Laboratory, École Polytechnique Fédérale de Lausanne (EPFL), 1015, Lausanne, Switzerland. His research interests include low-power architectures for biomedical applications and artificial intelligence-enabled Internet-of-Things devices. Denkinger received an M.Sc. degree in robotics and autonomous systems from the Institute of Electrical and Micro Engineering, EPFL. Contact him at benoit.denkinger@epfl.ch.

ALEXANDRE LEVISSE was a postdoctoral researcher in the Embedded Systems Laboratory, Swiss Federal Institute of Technology Lausanne, École Polytechnique Fédérale de Lausanne (EPFL), 1015, Lausanne, Switzerland, from 2018 to 2021. His research interests include circuits and architectures for emerging memory and transistor technologies as well as in-memory computing and accelerators. Levisse received a Ph.D. degrees in electrical engineering from CEA-LETI, Grenoble, France, and from Aix-Marseille University, Marseille, France. Contact him at alexandre.levisse@epfl.ch.

GIOVANNI ANSALONI is a researcher with the Embedded Systems Laboratory, École Polytechnique Fédérale de

Lausanne (EPFL), 1015, Lausanne, Switzerland. His research focuses on domain-specific and ultra-low-power architectures and algorithms for edge computing systems, including hardware and software optimization techniques. Ansaloni received a Ph.D. degree in informatics from USI. Contact him at giovanni.ansaloni@epfl.ch.

MIGUEL PEÓN-QUIRÓS is a postdoctoral researcher with École Polytechnique Fédérale de Lausanne (EPFL), 1015, Lausanne, Switzerland. His research focuses on optimizations for embedded devices. Peón-Quirós received a Ph.D. degree in computer architecture from the Complutense University of Madrid, Madrid, Spain. Contact him at miguel.peon@epfl.ch.

DAVID ATIENZA is a professor of electrical engineering, and heads the Embedded Systems Laboratory, École Polytechnique Fédérale de Lausanne (EPFL), 1015, Lausanne, Switzerland. His research focuses on design methodologies for edge AI in the context of Internet of Things and thermal- and energy-aware design for server architectures and datacenters. He is an IEEE Fellow and an ACM distinguished member. Contact him at david.atienza@epfl.ch.