Energy-Efficient Vision on the PULP Platform for Ultra-Low Power Parallel Computing

Francesco Conti*, Davide Rossi*, Antonio Pullini†, Igor Loi*, Luca Benini*†
*Department of Electrical, Electronic and Information Engineering, University of Bologna, Italy
†Integrated Systems Laboratory, ETH Zurich, Switzerland
{f.conti,davide.rossi,igor.loi}@unibo.it, {pullinia,lbenini}@iis.ee.ethz.ch

***Abstract*— Many-core architectures structured as fabrics of tightly-coupled clusters have shown promising results on embedded computer vision benchmarks, providing state-of-art performance with a reduced power budget. We propose PULP (Parallel processing Ultra-Low Power platform), an architecture built on clusters of tightly-coupled OpenRISC ISA cores, with advanced techniques for fast performance and energy scalability that exploit the capabilities of the STMicroelectronics UTB FD-SOI 28nm technology. As a use case for PULP, we show that a computationally demanding vision kernel based on Convolutional Neural Networks can be quickly and efficiently switched from a low power, low frame-rate operating point to a high frame-rate one when a detection is performed. Our results show that PULP performance can be scaled over a 1x-354x range, with a peak performance/power efficiency of 211 GOPS/W.**

I. Introduction

Embedded and mobile applications greatly benefit from a low-power, flexible computing fabric that is able to provide significant performance when needed and remain in a very low-consumption state when not. In particular, heavily energy-constrained applications such as wireless sensor nodes (WSNs) designed to work with input from low-power imagers and performing vision-related algorithms need an exceptional degree of performance and energy scalability to cope both with the limited energy budget and with the frame-rate requirements of vision applications. At the same time, a computing fabric answering to these needs should also provide very high flexibility and easy-to-use programming models to keep on track with the fast-moving CV field.

In this work we introduce *PULP* (Parallel processing Ultra-Low Power platform), a many-core platform answering to these demands. To achieve high performance when needed, PULP features clusters of simple, yet complete, OpenRISC [1] cores that can be used to exploit both coarse- and fine-grain data level parallelism or task level parallelism. At the same time, operating points (voltage, frequency, body biasing) can be controlled at a fine granularity and high speed to achieve high energy efficiency when the performance constraints are more relaxed or when the power budget is tighter. The proposed PULP platform exploits the capabilities of STMicroelectronics UTB FD-SOI technology [2] that, in contrast with deep submicron bulk technologies, allows to exploit an extended body bias range to modulate the performance/energy trade-off at different operating points.

We put our platform to test using Convolutional Neural Networks (*CNN*s or *ConvNet*s), a model that is state-of-art in many current CV benchmarks and has shown promising accuracy results in new classification, detection, and full-scene understanding tasks. CNN-based algorithms are typically computationally demanding and require a good level of performance to work at acceptable frame rates.

II. Related work

Architectural research on many-core architectures has focused on tiled platforms; each tile contains one or more cores and communicates with other tiles through a scalable medium. The dominating paradigm is that of general-purpose and embedded GPUs such as NVIDIA Fermi [3]. GPUs feature a restricted SPMD-based execution model that can be suboptimal for CV applications, which have often an irregular structure [4][5]. Many-core platforms with clusters of RISC cores have been proposed as a more flexible model: examples include STMicroelectronics P2012 [6], which is programmable in OpenCL [7] and OpenMP [8]; and Kalray MPPA [9], which supports a proprietary KPN-based programming model as well as OpenMP. To further improve efficiency in CV workloads, some platforms employ clusters of VLIW cores; for example, Movidius Myriad [10] features 8 SHAVE clusters, each including a VLIW core. Another example is the TI AccelerationPAC [11], which includes several EVE clusters, each composed of a RISC processor and a VLIW coprocessor.

For improved efficiency, many CV-focused platforms rely on fixed-function HW blocks. Most of these platforms are dataflow engines, often implemented on FPGAs or CGRAs. Examples of this approach include Vortex [12][13] for biologically-inspired vision acceleration, and NeuFlow [14] and nn-X [15], which focus on ConvNet acceleration. Also some commercial products follow this path: for example the Analog Devices Blackfin [16], which features a fixed-function Pipelined Vision Processor for CV acceleration. Another approach is to augment an existing many-core with accelerator cores, as is done in He-P2012 [17].

None of the platforms reported above currently targets ultra-low power operation, as their power budget ranges from hundreds of milliwatts to several watts. Conversely, state-of-art ULP microcontrollers can target power budgets lower than 10 mW: examples include the SiliconLabs EFM32 [18] and Texas Instruments MSP430 [19] families of MCUs. Significant efficiency can be reached by near-threshold microcontrollers such as the one shown in Ickes et al. [20], SleepWalker [21] and Bellevue [22], which also exploits SIMD parallelism to further improve performance. However, the performance level attainable by these low-power MCUs is still too low for most CV applications; for this reason,

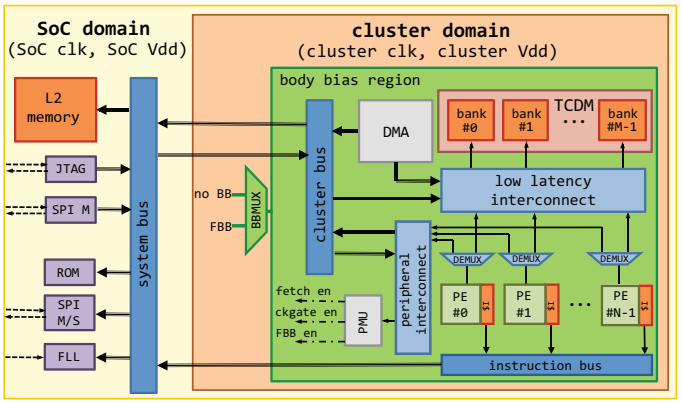


Figure 1: PULP architecture.

many CV-focused ULP accelerators employ fixed-function HW blocks [23][24][25].

Current parallel ULP processor designs are more directly comparable to the PULP platform. Centip3de [26] consists of a large scale 3D-integrated fabric of clusters of Cortex M3 cores. With 64 cores running at 10 MHz, it can reach a performance of 0.64 GOPS. DietSODA [27] features 128 SIMD lanes working at lower frequency (50 MHz) than the rest of the chip, reaching up to 6.4 GOPS. Dogan et al. [28] explore multicore design in subthreshold for biomedical usage, with a power budget as low as 10 μW.

To evaluate PULP we chose to use Convolutional Neural Networks (CNNs), which were originally proposed by Lecun et al. [29] to solve the MNIST digit classification problem. Interest in deep convolutional networks has been recently rekindled by the discovery of efficient ways to train them [30]; CNNs have been used to obtain state-of-art accuracy results on scene labeling, video classification and object detection by companies such as Google [31][32], Microsoft [33] and Facebook [34].

III. Architecture

A. PULP SoC overview

PULP (Parallel processing Ultra Low Power platform) is a scalable, clustered many-core computing platform able to operate on a large range of operating voltages, achieving in this way a high level of energy efficiency over a wide range of application workloads. Figure 1 shows the main building blocks of a single-cluster SoC. The PULP fabric is integrated in a SoC featuring a L2 memory (sized in the 32kB to 128kB range) shared among all clusters through a system bus, plus IO peripherals that provide flexibility to the whole platform.

The set of peripherals integrated in the PULP platform includes two SPI (Serial Peripheral Interface) interfaces (one master and one slave), GPIOs, a bootup ROM and a JTAG interface suitable for testing purposes. Both SPI interfaces can be configured in *single* mode or *quad* mode depending on the required bandwidth, and they are suitable for interfacing the SoC with a large set of off-chip components (non volatile memories, voltage regulators, cameras...). Moreover, the SPI slave can be configured as a master, and a set of enable signals placed on both SPI interfaces allow the SoC to interface to up to 4 slave peripherals.

Thanks to its peripheral architecture the SoC is able to operate in two different modes: *slave* mode or *stand-alone* mode. When configured in slave mode, PULP behaves as a many-core accelerator of a standard *host* processor (e.g. an ARM Cortex M low-power microcontroller). In this configuration the host microcontroller is responsible for loading the application and processing data on the PULP L2 through the SPI MASTER interface, and initiate and synchronize the computation through dedicated memory mapped signals (e.g. fetch enable) and GPIOs. When configured in stand-alone mode the SoC detects the presence of a flash memory on its SPI master interface, booting from the external flash if connected, from the L2 memory otherwise.

B. Cluster architecture

The cluster architecture features a parametric number of Processing Elements (*PE*s) consisting of a highly power optimized microarchitecture based on OpenRISC 32-bit ISA [1], each one with a private instruction cache (*I$*). The refill ports of all instruction caches converge on a common cluster instruction initiator port through a cluster instruction bus. The OpenRISC cores were optimized to achieve an IPC of almost 1 on a wide variety of benchmarks, including highly control-intensive code[35]. Energy efficiency is boosted by using a flat pipeline to reduce register and clocking overhead, while the datapath was area-optimized to reduce leakage. Further, extensive architectural clock gating was employed to reduce spurious dynamic power.

The PEs do not have private data caches, avoiding memory coherency overhead and increasing area efficiency, while they all share a L1 multi-banked tightly coupled data memory (*TCDM*) acting as a shared data scratchpad memory. The TCDM has a number of ports equal to the number of memory banks providing concurrent access to different memory locations. Intra-cluster communication is based on a high bandwidth *low-latency interconnect*, implementing a word-level interleaving scheme to reduce access contention [36].

A lightweight, ultra-low-programming-latency, multi-channel DMA enables fast and flexible communication with other clusters, the L2 memory and external peripherals [37]. The DMA uses minimal request buffering and features a direct connection to the TCDM, to eliminate the need for internal buffering, which is very expensive in terms of power. A peripheral interconnect provides access to all the cluster peripherals and to all the resources external to the cluster.

C. Power management

In order to provide the best energy efficiency across a wide range of workloads, each cluster can work at its own voltage and frequency. To enable fine grained tuning of the SoC frequency, a FLL (Frequency-Locked Loop [38]) is included as a peripheral at SoC level. Moreover, a set of clock dividers (one for the SoC + one for each cluster) allow to further divide the clock generated by the FLL. To reduce the dynamic power consumption in idle mode, each processor can be separately disabled and clock-gated through a set of registers mapped on the peripheral interconnect. In this way, depending on the required workload, each cluster is

able to work with an arbitrary number of processing elements, while the others consume zero dynamic power.

A body bias multiplexer (*BBMUX*) allows to dynamically select the back-bias voltage of the cluster, enabling ultra-fast transitions between the normal operating mode and the boost mode when temporary peaks of computation are required by the applications. To reduce the latency of the transitions between different operating modes, and making them transparent to the software, a power management unit (*PMU*) was added to generate the control signals of the processors fetch enables, clock gating units, and BBMUX.

IV. Benchmarking PULP

This section examines the implementation results of the PULP platform on a reference configuration targeting the ConvNet application, providing an estimation of the area of the platform, of the energy efficiency at the different operating points, and a comparison with other state of the art multi-core platforms for embedded computing.

A. Implementation results

In the context of this work we consider a single cluster PULP implementation operating in stand-alone mode. Thus, we assume the SoC connected to an external flash memory which contains the application code, a video surveillance camera periodically feeding the L2 of the SoC with a new frame, and a programmable DC/DC converter configured by the cores to switch between the idle, *search* and *follow* mode described in Section IV-C. The L2 memory was sized at 32kB to fit both the program code and the image frames. The cluster consists of 8 cores featuring 1kB of I$ each, while the TCDM is composed of 16 banks of 2kB each, leading to an overall TCDM size of 32kB. These architectural parameters were chosen to fit the constraints of the CNN described in Section IV-C, which should be sufficiently flexible for a variety of vision tasks. Both the TCDM banks and the processor's I$ are implemented using standard cell memory (SCM) cuts of 4kbits each. While SRAMs may achieve a higher density than SCMs (by a factor of ∼3x), SCMs are able to work at the same voltage ranges as the rest of the logic, with the key benefit of providing much smaller energy/access (∼4x)[39].

Our results refer to a post place & route implementation of the proposed SoC in STMicroelectronics 28nm UTB FD-SOI technology. Thus, they include the overheads (i.e. timing, area, power) caused by the clock tree implementation, accurate parasitic models extraction, cell sizing for setup fixing and delay buffers for hold fixing (neglecting these would cause significant underestimations in the clock tree dynamic power). The SoC was synthesized with Synopsys dc_shell, the place & route was performed using Cadence SoC Encounter, and the signoff was performed using Synopsys StarRC for parasitic extraction and Synopsys PrimeTime for timing and power analysis.

We tested our platform with power supplies ranging from 0.3V to 1.3V and forward body biasing ranging from 0 to 1V in the typical corner case at the temperature of 25°C. Table I shows the peak frequency that the PULP cluster can reach at each operating point. Being the cluster composed of 8 cores, the theoretical

V_{DD} [V]	f_{max}[MHz] $V_{FBB} = 0V$	f_{max}[MHz] $V_{BB} = 0.5V$	f_{max}[MHz] $V_{BB} = 1V$
0.3	**2.5**	4.45	6.31
0.4	**22**	35.9	**49.1**
0.6	**200**	**277**	**350**
0.8	400	**484**	**563**
1.0	588	**650**	**705**
1.3	**775**	**836**	885

Table I: Supply voltage and peak frequencies for the reference PULP cluster. Bold values indicate reference operating points.

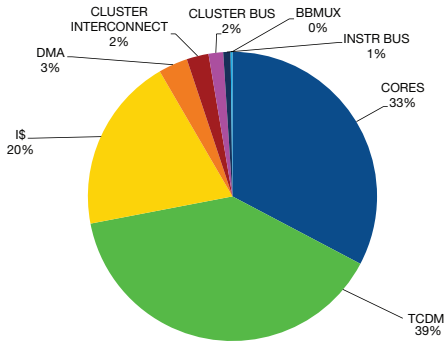


Figure 2: PULP cluster area breakdown.

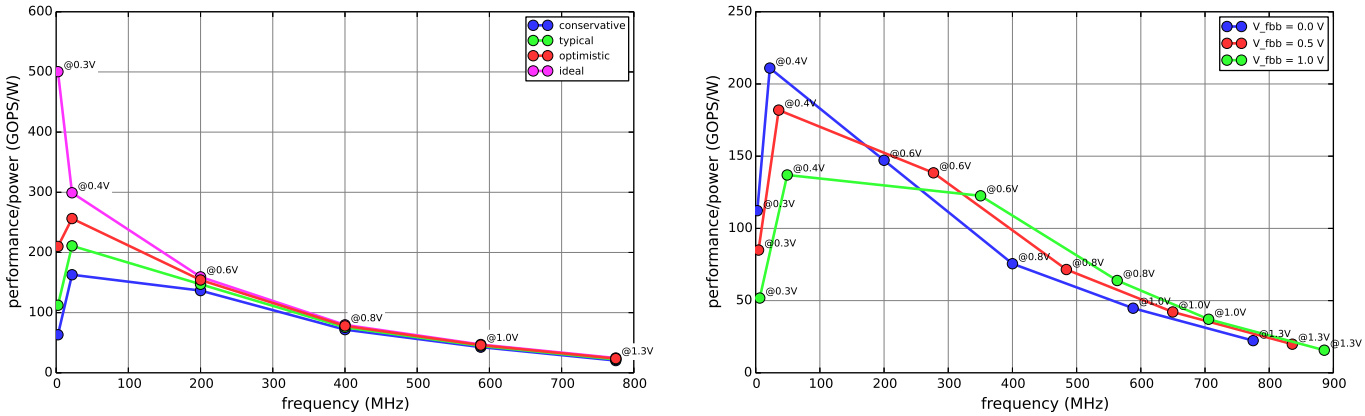
performance of the platform can easily scale between 20 MOPS @0.3V, no BB to 7 GOPS @ 1.3V, 1.0V FBB, demonstrating the dramatic performance scalability (354x) that can be exploited on PULP.

Figure 2 shows the area breakdown of the cluster, where the overall cluster area in the considered configuration is 1.2 mm^2. It is possible to note that the TCDM and the cores I$ occupy ∼59% of the overall cluster area, mainly due to the SCM based implementation. However, this is fully compensated by the improvement in terms of dynamic power consumption of the memories, which are responsible for the ∼15% of the overall cluster dynamic power, with an improvement of ∼4x with respect to a previous implementation of the same architecture [35].

B. Energy efficiency analysis

This section provides an evaluation of the energy efficiency of the proposed PULP implementation at the different operating points that can be exploited on the platform. To cope with the leakage power variation in the 28nm UTB FD-SOI, cell libraries are characterized very conservatively; early silicon measurements on PULP prototypes showed that there is more than a 2x guardband on power models. For this reason, we analyze the energy efficiency of the platform in four scenarios, accounting for various levels of pessimism for leakage: *conservative*, where the leakage power is directly extracted from the standard cell libraries; *typical*, with leakage scaled down by 2x; *optimistic*, where it is scaled down by 5x; and *ideal* with no leakage.

Figure 3a shows the results of this exploration; the platform is working at the maximum operating frequency achievable at each given supply voltage. The peak energy efficiency points in the four scenarios are 172 GOPS/W, 211 GOPS/W, 262 GOPS/W, and 500 GOPS/W respectively. The best energy efficiency point is around 0.4V in all the scenarios except for the ideal. In all but the ideal scenario, the impact of leakage power is huge in



(a) GOPS/W while scaling the leakage contribution to power.

(b) GOPS/W with 0V, 0.5V and 1.0V FBB.

Figure 3: PULP energy efficiency in GOPS/W.

the 0.3V to 0.4V operating range, when the supply voltage V_{DD} is close to V_{th} (0.28V for this technology), due to the relatively slow operating frequency (2.5MHz to 50 MHz) that causes the static contribution of leakage to be dominant. On the other hand, when working with V_{DD} larger than 0.6V, the combined effect of increased dynamic power density (which scales as V_{DD}^2), and higher operating frequency causes the impact of leakage to be smaller. In the rest of the paper we consider the typical scenario as the reference one for further power estimations and comparisons.

Figure 3b shows what happens when forward body biasing (FBB) is introduced. By applying FBB, it is possible to dynamically modulate the V_{th} of transistors to improve the frequency without changing the supply, with only a slight increase of dynamic power in the high-V_{DD} range. On the other hand, FBB introduces an overhead in leakage power, quantifiable as a 7x increase when V_{BB} is 1V [2]. For these reasons, FBB is an effective knob to increase the energy efficiency by up to 1.5x for workloads larger than 1.6 GOPS (200 MHz). For example, the target workload of 3.2 GOPS (400 MHz) can be achieved @0.8V with 0V FBB or @0.6V with 1V FBB, resulting in a 1.5x improvement in energy efficiency.

To further provide insight into the scaling capabilities of the PULP platform, in Figure 4 we investigate energy efficiency in terms of peak GOPS per Watt. We compare the reference PULP platform with several other commercial and academic platforms: the Processing System of the Xilinx Zynq platform (i.e. a dual core ARM Cortex A9), a Samsung Exynos 5 (i.e. a ARM big.LITTLE quad-core A7 + quad-core A15), and many of the ULP platforms referenced in Section II. PULP, providing up to 211 GOPS/W, is competitive with microcontrollers specialized for low-power (Bellevue, SleepWalker) and more performant parallel ULP platforms (Centip3de, DietSoda), and is much more efficient than mobile solutions such as the Exynos 5 due to the simpler, optimized architecture of the OpenRISC cores and to the fine-grain knobs for power management provided by the FDSOI technology. It must also be noted that both Centip3de and DietSoda do not support a programming model, whereas PULP has been designed for compatibility with standards such as OpenCL and OpenMP, to ease the exploitation of potential performance in applications.

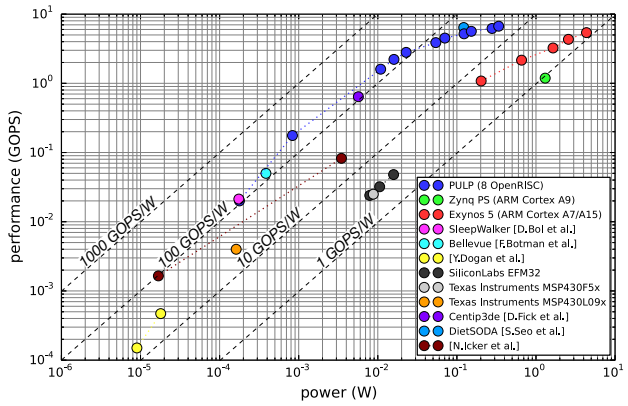


Figure 4: Energy efficiency comparison with several platforms.

C. *ConvNet benchmark*

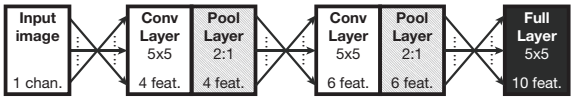


Figure 5: Reference CNN architecture.

A CNN is composed by a deep sequence of convolutional or fully-connected linear layers intermixed with pooling ones to perform a transformation on *feature maps* produced by the previous layer. Weights in convolutional and linear layers are trained by backpropagation but are used thereafter in a strictly feedforward fashion; due to their data parallel nature they are a natural candidate for acceleration in a parallel platform such as PULP. Convolutional layers in CNNs compute output feature maps of a layer as sums of convolutions over input feature maps; therefore, we chose to use a *convolution-accumulation* step as our basic kernel: $y(i,j) := y(i,j) + \Big(W * x\Big)(i,j)$.

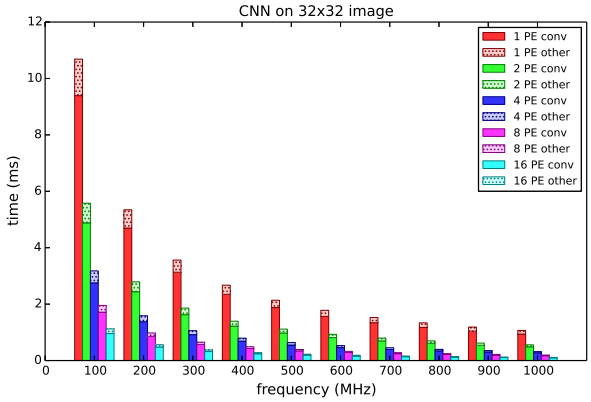


Figure 6: Performance of the reference CNN, scaling frequency from 100 to 1000 MHz and cores from 1 to 16.

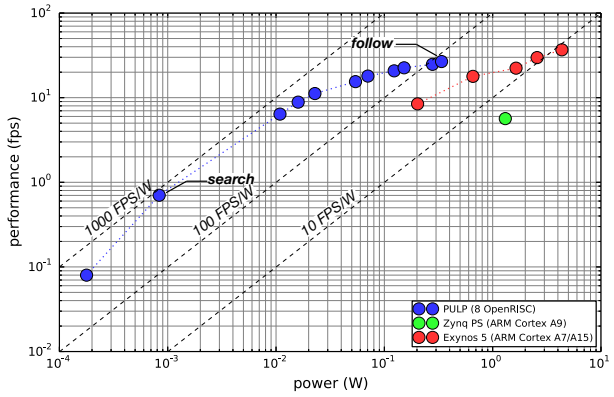


Figure 7: Energy efficiency for execution of the CNN on a QVGA frame.

Implementation	3x3	5x5	7x7	9x9	11x11
`naive`, single thread	0.26	0.32	0.34	0.35	0.36
`1-unrolled`, single thread	0.52	0.62	0.65	**0.69**	**0.88**
`2-unrolled`, single thread	**0.80**	**0.83**	**0.76**	0.26	0.18
`naive`, 8 threads	0.26	0.31	0.34	0.35	0.36
`1-unrolled`, 8 threads	0.49	0.60	0.65	**0.69**	**0.85**
`2-unrolled`, 8 threads	**0.71**	**0.77**	**0.74**	0.27	0.18

Table II: Convolution-Accumulation: average efficiency/core

We used 16-bit fixed point numbers for inputs, kernels and outputs. We implemented three versions of convolution-accumulation: `naive` directly implements it as four nested loops (two on the output pixels and two for the convolution kernel W); `1-unrolled` uses manual loop unrolling on the innermost loop; `2-unrolled` uses loop unrolling on the two innermost loops. We benchmarked these convolutions with a single thread or 8 parallel threads[1].

Table II shows the efficiency/core for the various convolution-accumulation implementations on a 32x32 input image, computed as the ratio between useful (i.e. computation) cycles and the total number of cycles spent in the outermost loop. For smaller convolution kernels, unrolling both inner loops provides a much better efficiency; however, for kernels bigger than 7x7, efficiency is reduced by I$ misses due to the size of the unrolled loop. As a consequence, the tighter `1-unrolled` convolution-accumulation step is more convenient for bigger kernels. Results are similar in the multi-threaded case, as data contention on the TCDM causes on average only a small amount of efficiency decrease.

On top of these optimized convolutions, we developed a network based on the one for MNIST classification from LeCun et al. [29], which is shown in Figure 5. This network has 2220 parameters and a footprint of 11408 bytes for data and 4400 bytes for weights on the L1 TCDM. The program code uses 16768 bytes on the L2 memory. As shown in Conti et al. [40], a network of this kind can be trained for complex object detection tasks by running it on a window sliding over the input frame.

We use this CNN for visual surveillance. The platform spends most of the time in a low-power *search* mode looking for suspicious objects (as this task requires only a relatively low frame rate), and it switches to a high-performance *follow* mode to keep track of a previously detected object. Input frames are brought inside the PULP cluster by DMA transfer from the L2. This transfer is superimposed to the computation of deeper layers and has no impact on the final throughput.

Figure 6 shows the performance of the reference CNN when run on a 32x32 image patch, scaling the clock frequency of the cluster from 100 MHz to 1 GHz and the number of OpenRISC cores in the PULP cluster between 1, 2, 4, 8 or 16. As expected from a highly data-parallel algorithm such as ConvNets, execution time scales almost linearly with the number of cores. In our visual surveillance application, the ConvNet is run on a 32x32 window spanning a QVGA (320x240) image with a stride of 32 pixels. Each frame is spanned two times: one with no offset, the other with an offset of 16 pixels in both directions so that the chance of missed detections on the border of a window are reduced. PULP can be set to work at a very low frame rate (∼0.7 fps at the 0.4V operating point) in the *search* mode, and then switched to a frame rate as high as 27 fps (at the 1.3V operating point with 1V FBB) in the *follow* mode.

Figure 7 shows the energy efficiency of the ConvNet execution on a frame in terms of FPS/W; we ran the same ConvNet on the Xilinx Zynq PS and on a Samsung Exynos 5 for comparison, as this benchmark is beyond the typical performance capabilities of most ULP microcontroller architectures. Benchmark results substantially confirm the theoretical values shown in Figure 4. The energy/execution time tradeoff when switching between *search* and *follow* mode is also clearly shown: in *search* mode, PULP consumes 1.18 mJ per frame, whereas in *follow* mode energy consumption jumps at 12.6 mJ per frame.

V. Conclusions

As our main contribution, we have introduced the PULP (*Parallel processing Ultra-Low Power*) platform that features clusters of tightly-coupled OpenRISC cores to achieve high energy efficiency through parallelism. We have analysed the platform, showing that its performance can be scaled by the dramatical factor of 354x and that it features a peak energy efficiency of 211 GOPS/W. As a use case for PULP, we implemented a ConvNet-

[1]We used the `or1k-elf-gcc` compiler (build `4.9.0 20140308`), with the following flags: `-O2 -nostdlib -mhard-mul -msoft-div`.

based algorithm for video surveillance, showing that it can be switched from a low-power state consuming just 1.18 mJ per frame with a rate of 0.7fps to a high-performance state running at 27fps and consuming 12.6mJ per frame. Our future work will focus on pushing the PULP architecture to the 1 GOPS/mW limit, making it competitive with special purpose mixed-signal accelerators such as the 1.57 TOPS/W in Kim et al. [41] in terms of energy efficiency, while also preserving general software programmability.

ACKNOWLEDGEMENTS

This work was funded by the project IcySoC, financed with a grant from the Swiss Nano-Tera.ch initiative and evaluated by the Swiss National Science Foundation, and by the EU FP7 project PHIDIAS (g.a. 318013).

REFERENCES

[1] "OpenRISC 1000 Architecture Manual." [Online]. Available: http://opencores.org/websvn,filedetails?repname=openrisc&path=%2Fopenrisc%2Ftrunk%2Fdocs%2Fopenrisc-arch-1.1-rev0.pdf

[2] D. Jacquet, F. Hasbani, P. Flatresse, R. Wilson, F. Arnaud, G. Cesana, T. Di Gilio, C. Lecocq, T. Roy, A. Chhabra, C. Grover, O. Minez, J. Uginet, G. Durieu, C. Adobati, D. Casalotto, F. Nyer, P. Menut, A. Cathelin, I. Vongsavady, and P. Magarshack, "A 3 GHz Dual Core Processor ARM Cortex TM -A9 in 28 nm UTBB FD-SOI CMOS With Ultra-Wide Voltage Range and Energy Efficiency Optimization," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 4, pp. 812–826, April 2014.

[3] D. Patterson, "The top 10 innovations in the new NVIDIA Fermi architecture, and the top 3 next challenges," *NVIDIA Whitepaper*, 2009.

[4] J. Clemons, H. Zhu, S. Savarese, and T. Austin, "MEVBench: A mobile computer vision benchmarking suite," in *2011 IEEE International Symposium on Workload Characterization (IISWC)*. IEEE, Nov. 2011, pp. 91–102.

[5] A. Mahesri, D. Johnson, N. Crago, and S. J. Patel, "Tradeoffs in designing accelerator architectures for visual computing," in *2008 41st IEEE/ACM International Symposium on Microarchitecture*. IEEE, Nov. 2008, pp. 164–175.

[6] L. Benini, E. Flamand, D. Fuin, and D. Melpignano, "P2012: Building an ecosystem for a scalable, modular and high-efficiency embedded computing accelerator," in *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, Mar. 2012, pp. 983–987.

[7] D. Melpignano, L. Benini, E. Flamand, B. Jego, T. Lepley, G. Haugou, F. Clermidy, and D. Dutoit, "Platform 2012, a many-core computing accelerator for embedded SoCs," in *Proceedings of the 49th Annual Design Automation Conference on - DAC '12*. New York, New York, USA: ACM Press, 2012, p. 1137.

[8] A. Marongiu, A. Capotondi, G. Tagliavini, and L. Benini, "Improving the programmability of STHORM-based heterogeneous systems with offload-enabled OpenMP," in *Proceedings of the First International Workshop on Many-core Embedded Systems - MES '13*. New York, New York, USA: ACM Press, 2013, pp. 1–8.

[9] B. D. de Dinechin, R. Ayrignac, P.-E. Beaucamps, P. Couvert, B. Ganne, P. G. de Massas, F. Jacquet, S. Jones, N. M. Chaisemartin, F. Riss, and T. Strudel, "A clustered manycore processor architecture for embedded and accelerated applications," in *2013 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, Sep. 2013, pp. 1–6.

[10] D. Moloney, "1TOPS/W software programmable media processor," in *HotChips HC23*, Stanford, 2011.

[11] Z. Lin, J. Sankaran, and T. Flanagan, "Empowering automotive vision with TIs Vision AccelerationPac," *TI White Paper*, 2013.

[12] S. Park, A. A. Maashri, K. M. Irick, A. Chandrashekhar, M. Cotter, N. Chandramoorthy, M. Debole, and V. Narayanan, "System-On-Chip for Biologically Inspired Vision Applications," *IPSJ Transactions on System LSI Design Methodology*, vol. 5, pp. 71–95, 2012.

[13] J. Sabarad, S. Kestur, D. Dantara, V. Narayanan, and D. Khosla, "A reconfigurable accelerator for neuromorphic object recognition," in *17th Asia and South Pacific Design Automation Conference*. IEEE, Jan. 2012, pp. 813–818.

[14] C. Farabet, B. Martini, B. Corda, P. Akselrod, E. Culurciello, and Y. LeCun, "NeuFlow: A runtime reconfigurable dataflow processor for vision," in *CVPR 2011 Workshops*. IEEE, Jun. 2011, pp. 109–116.

[15] V. Gokhale, J. Jin, A. Dundar, B. Martini, and E. Culurciello, "A 240 G-ops/s Mobile Coprocessor for Deep Neural Networks," in *CVPR 2014 Workshops*.

[16] "Analog Devices Blackfin Dual Core Processor." [Online]. Available: http://www.analog.com/en/processors-dsp/blackfin/adsp-bf608/products/product.html

[17] F. Conti, C. Pilkington, A. Marongiu, and L. Benini, "He-P2012 : Architectural Heterogeneity Exploration on a Scalable Many-Core Platform," in *Proceedings of 25th IEEE Conference on Application-Specific Architectures and Processors*, 2014.

[18] "SiliconLabs EFM32 Microcontroller." [Online]. Available: http://www.silabs.com/products/mcu/lowpower/pages/efm32g-gecko.aspx

[19] "Texas Instruments MSP430 Low-Power MCUs." [Online]. Available: http://www.ti.com/lsds/ti/microcontrollers_16-bit_32-bit/msp/overview.page

[20] N. Ickes, Y. Sinangil, F. Pappalardo, E. Guidetti, and A. P. Chandrakasan, "A 10 pJ/cycle ultra-low-voltage 32-bit microprocessor system-on-chip," in *2011 Proceedings of the ESSCIRC (ESSCIRC)*. IEEE, Sep. 2011, pp. 159–162.

[21] D. Bol, J. De Vos, C. Hocquet, F. Botman, F. Durvaux, S. Boyd, D. Flandre, and J.-D. Legat, "SleepWalker: A 25-MHz 0.4-V Sub-mm2 7-uW/MHz Microcontroller in 65-nm LP/GP CMOS for Low-Carbon Wireless Sensor Nodes," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 20–32, Jan. 2013.

[22] F. Botman, J. D. Vos, S. Bernard, F. Stas, J.-D. Legat, and D. Bol, "Bellevue : a 50MHz Variable-Width SIMD 32bit Microcontroller at 0 . 37V for Processing-Intensive Wireless Sensor Nodes," in *Proceedings of 2014 IEEE Symposium on Circuits and Systems*, 2014, pp. 1207–1210.

[23] J.-S. Yoon, J.-H. Kim, H.-E. Kim, W.-Y. Lee, S.-H. Kim, K. Chung, J.-S. Park, and L.-S. Kim, "A Unified Graphics and Vision Processor With a 0.89 uW/fps Pose Estimation Engine for Augmented Reality," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 2, pp. 206–216, Feb. 2013.

[24] D. Jeon, Y. Kim, I. Lee, Z. Zhang, D. Blaauw, and D. Sylvester, "A 470 mV 2.7mW Feature Extraction-Accelerator for Micro-Autonomous Vehicle Navigation in 28nm CMOS," pp. 166–168, 2013.

[25] J. Oh, S. Lee, and H.-J. Yoo, "1.2-mW Online Learning Mixed-Mode Intelligent Inference Engine for Low-Power Real-Time Object Recognition Processor," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 5, pp. 921–933, May 2013.

[26] D. Fick, R. G. Dreslinski, B. Giridhar, G. Kim, S. Seo, M. Fojtik, S. Satpathy, Y. Lee, D. Kim, N. Liu, M. Wieckowski, G. Chen, T. Mudge, D. Blaauw, and D. Sylvester, "Centip3De: A Cluster-Based NTC Architecture With 64 ARM Cortex-M3 Cores in 3D Stacked 130 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 104–117, Jan. 2013.

[27] S. Seo, R. G. Dreslinski, M. Woh, C. Chakrabarti, S. Mahlke, and T. Mudge, "Diet SODA: A Power-Efficient Processor for Digital Cameras," in *Proceedings of the 16th ACM/IEEE international symposium on Low power electronics and design - ISLPED '10*. New York, New York, USA: ACM Press, 2010, p. 79.

[28] A. Y. Dogan, D. Atienza, A. Burg, I. Loi, and L. Benini, "Power/performance exploration of single-core and multi-core processor approaches for biomedical signal processing," in *Integrated Circuit and System Design. Power and Timing Modeling, Optimization, and Simulation*. Springer, 2011, pp. 102–111.

[29] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[31] A. Karpathy and T. Leung, "Large-scale Video Classification with Convolutional Neural Networks," in *Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[32] A. Toshev and C. Szegedy, "DeepPose : Human Pose Estimation via Deep Neural Networks," in *Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[33] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong, and A. Acero, "Recent advances in deep learning for speech research at Microsoft," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8604–8608, May 2013.

[34] N. Zhang, M. Paluri, M. A. Ranzato, T. Darrell, L. Bourdev, and U. C. Berkeley, "PANDA : Pose Aligned Networks for Deep Attribute Modeling," in *Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2014.

[35] M. Gautschi, D. Rossi, and L. Benini, "Customizing an open source processor to fit in an ultra-low power cluster with a shared L1 memory," in *Proceedings of the 24th edition of the great lakes symposium on VLSI - GLSVLSI '14*. New York, New York, USA: ACM Press, 2014, pp. 87–88.

[36] A. Rahimi, I. Loi, M. R. Kakoee, and L. Benini, "A fully-synthesizable single-cycle interconnection network for Shared-L1 processor clusters," *2011 Design, Automation & Test in Europe*, pp. 1–6, Mar. 2011.

[37] D. Rossi, I. Loi, G. Haugou, and L. Benini, "Ultra-low-latency lightweight DMA for tightly coupled multi-core clusters," in *Proceedings of the 11th ACM Conference on Computing Frontiers - CF '14*. New York, New York, USA: ACM Press, 2014, pp. 1–10.

[38] I. Miro-Panades, E. Beigné, Y. Thonnart, L. Alacoque, P. Vivet, S. Lesecq, D. Puschini, A. Molnos, F. Thabet, B. Tain, K. B. Chehida, S. Engels, R. Wilson, and D. Fuin, "A Fine-Grain Variation-Aware Dynamic Vdd-Hopping AVFS Architecture on a 32 nm GALS MPSoC," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 7, pp. 1475–1486, Jul. 2014.

[39] P. Meinerzhagen, S. M. Y. Sherazi, A. Burg, and J. N. Rodrigues, "Benchmarking of Standard-Cell Based Memories in the Sub-Vt Domain in 65-nm CMOS Technology," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 2, pp. 173–182, Jun. 2011.

[40] F. Conti, A. Pullini, and L. Benini, "Brain-inspired Classroom Occupancy Monitoring on a Low-Power Mobile Platform," in *CVPR 2014 Workshops*, 2014.

[41] G. Kim, Y. Kim, K. Lee, and S. Park, "A 1.22TOPS and 1.52mW/MHz Augmented Reality Multi-Core Processor with Neural Network NoC for HMD Applications," in *Proceedings of 2014 IEEE International Solid-State Circuits Conference*, 2014, pp. 182–184.