A Bespoke Design Approach to Low-Power Printed Microprocessors for Machine Learning Applications

Panagiotis Chaidos, Giorgos Armeniakos, Sotirios Xydis and Dimitrios Soudris
National Technical University of Athens, Greece
{pchaidos, armeniakos, sxydis, dsoudris}@microlab.ntua.gr,

Abstract—Printed electronics have gained significant traction in recent years, presenting a viable path to integrating computing into everyday items, from disposable products to low-cost healthcare. However, the adoption of computing in these domains is hindered by strict area and power constraints, limiting the effectiveness of general-purpose microprocessors. This paper proposes a bespoke microprocessor design approach to address these challenges, by tailoring the design to specific applications and eliminating unnecessary logic. Targeting machine learning applications, we further optimize core operations by integrating a SIMD MAC unit supporting 4 precision configurations that boost the efficiency of microprocessors. Our evaluation across 6 ML models and the large-scale Zero-Riscy core, shows that our methodology can achieve improvements of 22.2%, 23.6%, and 33.79% in area, power, and speed, respectively, without compromising accuracy. Against state-of-the-art printed processors, our approach can still offer significant speedups, but along with some accuracy degradation. This work explores how such trade-offs can enable low-power printed microprocessors for diverse ML applications.

Index Terms—Printed Electronics, Machine Learning, Microprocessors

I. INTRODUCTION

In recent years, printed electronics have emerged as a key player in the Internet of Things (IoT) landscape, particularly for ultra-resource constrained devices [1]. These technologies are central to embedded machine learning (ML) inference, offering mechanical flexibility, low non-recurring engineering (NRE) costs, and affordability in production. As a crucial component of the "Fourth Industrial Revolution," printed electronics enable the development of smart, lightweight devices using inexpensive, widely available materials and simple printing processes, paving the way for innovative integration methods and functionality at reduced costs [2].

Printed electronics present an opportunity to bring computing and intelligence into sectors like disposable products (e.g., packaged foods and beverages), smart packaging, low-cost healthcare (e.g., smart bandages), in-situ monitoring, and the expansive fast-moving consumer goods (FMCG) market [3]. However, these domains have seen limited adoption due to strict area and power constraints, particularly in wearables and implantables. While high costs drive applications to use general-purpose microprocessors and microcontrollers, their inefficiency in meeting the demands of these applications highlights a growing need for custom (*bespoke*) microprocessors.

Bespoke [3], [4] microprocessors enhance area and power efficiency by removing unused logic specific to a given appli-

cation. By removing unnecessary components, these custom designs—tailored to one or several applications, significantly reduce both area and power consumption compared to general-purpose microprocessors. Removing unnecessary logic enables further optimization through the modification (or even addition [5]) of hardware units, tailoring them to the application's specific operations. As a result, performance can be significantly improved, as well.

In this work, we embrace the bespoke design paradigm and apply a set of bespoke logic reductions to boost area and power efficiency. As a proof-of-concept scenario we profile a set of 2 low power cores, i.e., Zero-Riscy of PULP platform and TP-ISA [1][1]. Continuously, by leveraging the freed-up area and focusing on core operations for ML applications, we modify existing ALU and develop an SIMD MAC unit with support for multiple precision levels. Overall, compared to our baseline zero-riscy core and across 3 MLPs and 3 SVMs models, our approach achieves 22.2%, 23.6% and 33.79% improvements in area, power and speedup, respectively, with zero accuracy loss. On the other hand, when compared to the state-of-the-art printed processor TP-ISA, our methodology delivers an 85.1% speedup, albeit with trade-offs of 1.98x area, 1.82x power, and a 0.5% top-1 accuracy loss.

Our main contributions in this work are the following:

1) We propose a generic methodology for bespoke microprocessors - an approach to reducing area and power by tailoring a processor to specific applications.
2) Leveraging our bespoke design paradigm, we further optimize the ALU unit of the examined processors by incorporating an SIMD MAC unit for several precision levels.
3) This is the first work to synthesize large-scale microprocessors for printed technologies. Despite the trade-offs explored, our findings represent a significant step toward enabling battery-powered printed microprocessors.

II. RELATED WORK & BACKGROUND

Printed electronics cannot match silicon-based electronics in terms of integration density, area, or performance. Typical operating frequencies for printed circuits range from a few Hz to a few kHz [6], and feature sizes are usually several microns [7]. Despite these limitations, printed electronics offer unique advantages such as flexibility, adaptability, and most

---

[1]We use TP-ISA and Zero-Riscy as our proof-of-concept microarchitecture. The proposed workflow can be straightforwardly adopted in other processors.

notably, drastically lower fabrication costs — even at low volumes — making them ideal for application domains that traditional silicon-based VLSI cannot easily access.

The field of printed electronics has seen a substantial increase in research across multiple application areas. One example is the development of RFID tags utilizing pseudo-CMOS logic to enhance thin-film circuit performance [8]. Another notable effort involved fabricating a 2-input neuron designed to perform MAC operations [9]. More recently, ARM achieved a breakthrough by creating a flexible 32-bit microprocessor comprising over $18,000$ gates [10] .

While several studies focus on optimization and approximation techniques to mitigate area and power limitations of printed circuits [11], [12], research on printed ML applications remains in its early stages, mainly due to the large feature sizes of printed circuits. For instance, [4] introduced an automated approach for creating bespoke classifiers, and in [13], a hard-wired machine learning processor was integrated into a system for odor recognition. Additionally, [14] explored Stochastic Computing (SC) to reduce the area and power of printed MLPs, but this often resulted in significant accuracy loss. This work goes along with state-of-the-art, investigates the potential of printed microprocessors using printed technology and evaluates how bespoke, optimized synthesis methods can facilitate the creation of efficient printed cores.

III. ENABLING BESPOKE MICROPROCESSORS

This section outlines the workflow for extracting hardware specifications, including ROM usage, processor area, timing, and power, as well as the process of eliminating underutilized hardware. We present a methodology for measuring hardware utilization according to the specific needs of the applications. The non-utilized hardware blocks and architectural components are removed based on information gathered from the workflow. Finally, we develop an SIMD Multiply-Accumulate(MAC) unit to accelerate the operations of ML models, utilizing several levels of precision to enable parallel computation and achieving significant speedup. The proposed methodology is demonstrated through a proof-of-concept implementation on two low-power processors.

A. Application Dependent Logic Reduction

This work investigates a suite of ML applications designed for printed computing, utilizing the EGFET standard cell library. Given the flexibility advantage of microprocessors compared to Application Specific Integrated Circuits(ASICs), the study profiles and applies the proposed methodology on two low-power microprocessors, Zero-Riscy, a 32-bit 2-stage pipeline RISC-V architecture, and TP-ISA [1], a minimal highly configurable core, as a proof-of-concept.

We synthesize Zero-Riscy and two configurations of TP-ISA. As shown in Figure 1, the total area and power consumption of both cores are presented, along with the percentage data for the large functional units of Zero-Riscy, since TP-ISA does not have clearly defined components. Zero-Riscy, being a larger-scale low power processor, appears to be prohibitively
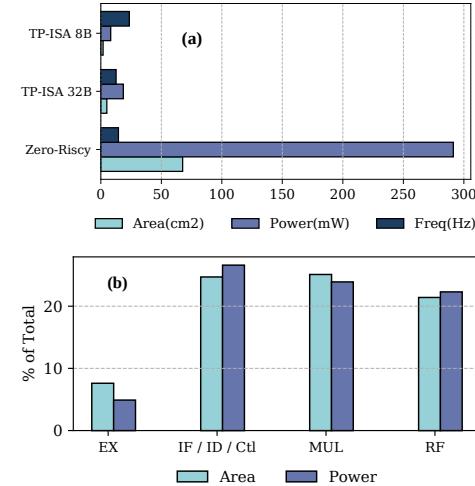

Fig. 1. **a**): Baseline Area, Power and System Clock for Zero-Riscy and TP-ISA for EGFET printed technology. **b**): Percentage of total area and power consumption for the main functional units of Zero-Riscy, **EX**(Execution Unit), **MUL**(Multiplier), **RF**(Register File) and **IF / ID / Ctl**(Instruction Fetch, Instruction Decode and Controller units) grouped together.

large in the EGFET technology, reaching a circuit area of 67.53 cm2 and power consumption of 291.21mW. It is worth noting that the multi-stage multiplier unit and the register file of Zero-Riscy account for almost half of the total area and power consumption, at 46.5% and 46.2% respectively. In contrast, both TP-ISA configurations fall well within the technology limitations, suggesting a focus on performance optimizations rather than area and power consumption. Furthermore, when considering printed memories, code length can also significantly reduce area and power consumption by utilizing fewer ROM cells. Each ROM cell takes up 0.84mm2 and 18.23uW, favoring designs with narrower bit-widths and smaller code sizes.

We assemble a collection of printed ML and associated applications [15], [1], including a 3-layer Multi-Layer Perceptron(MLP), a depth-2 Decision Tree(DT), simple Multiplication-Division and Insertion Sort on array of size 16. For Zero-Riscy, the Debug, Interrupt Controller, and Compressed Decoder Unit are not utilized and are completely removed. From the RISC-V instruction set, the SLT, most CSR, System Calls, and MULH instructions remain unused and can be effectively eliminated. Additionally, 12 registers are sufficient for executing all benchmarks, allowing for the removal of the rest. Since the code size and register usage for most applications are less than the baseline, this means that they can be addressed with narrower bitwidths, enabling a reduction in the Program Counter(PC) from 32 bits to 10 bits, and a reduction in the Base Address Registers(BARs) from 32 bits to 8 bits. TP-ISA is proven to be minimal and thus the focus here is primarily on improving performance and reducing code size, as will be discussed next.
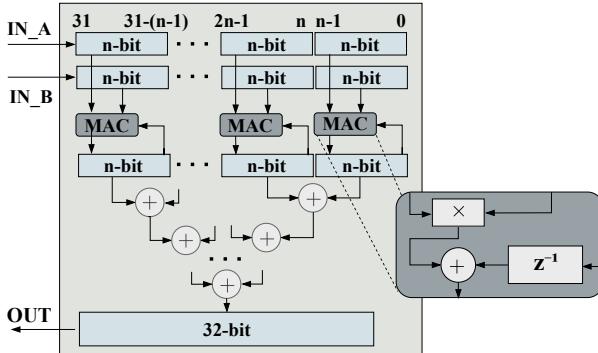
Fig. 2. Overview of proposed MAC unit. The unit has been implemented for precision options with n=32, 16, 8 and 4 bits. For each option, the unit can be split into 1, 2, 4 and 8 concurrent operations respectively.

B. ML acceleration unit

Logic reductions executed on the previous step create space for extensions that should lead to performance improvements. In order to optimize for ML workloads, we design a unit targeting the most frequent and computationally intensive ML operation - MAC. Most ML models heavily utilize MAC, which is mainly used for computing neuron activations. The proposed unit enables single-cycle multiplication and accumulation, significantly improving performance compared to the baseline, which require at least 3 cycles for Zero-Riscy and several more for TP-ISA where the whole operation is scheduled to the ALU. Additionally, the reduced instruction count for MAC-intensive code leads to program memory savings through smaller ROM requirements.

ML models exhibit inherent resilience to approximation errors, maintaining relatively high model accuracy up to certain fault thresholds [16]. We leverage the tolerance of these applications by integrating several precision options for the proposed MAC units. This design choice permits us to (a)utilize parallel execution and (b)replace large multipliers with small ones that have less depth. Figure 2 depicts the architecture of the unit for a selected precision n. By employing precision n, the unit effectively computes MAC of $32/n$ neurons in a single cycle, as shown in Equation 1.

$$
\begin{aligned}
acc_{total} &= \sum_{i=1}^{K} acc_i \\
acc_1 &= (r1[n-1:0] \times r2[n-1:0]) + acc_1 \\
acc_2 &= (r1[2n-1:n] \times r2[2n-1:n]) + acc_2 \\
&\vdots \\
acc_k &= (r1[31:31-(n-1)] \times r2[31:31-(n-1)]) + acc_k
\end{aligned}
\tag{1}
$$

C. Workflow

The complete overview of our proposed workflow for generating bespoke microprocessors is presented in Figure 3. Firstly, using Synopsys DC with the EGFET library we synthesize the cores and extract the area and power consumption information of the design(❶). With each benchmark's code, using the
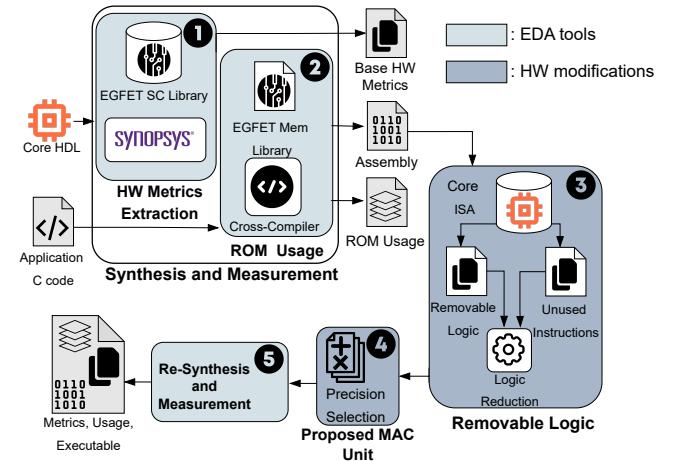

Fig. 3. Diagram of the proposed methodology for bespoke ML microprocessors.

respective compilers, the executables and assembly code are produced. Using the size of the executables along with the overhead of ROM cells in EGFET, we calculate the total area and power consumption of the program memory required to store each benchmark(❷).

Utilizing the compiled assembly code, we extract unused instructions for each ISA and remove all unused HW components and unused logic. We analyse the use of architectural components including flags and registers. By looking at the code sizes and utilized registers we determine the required bitwidth for addressing, thus trimming the PC and BAR(❸).

Having eliminated unused logic, we look towards performance optimizations and insert our proposed SIMD MAC unit as part of the ISA and test with several precision configurations(❹). The benchmarks are rewritten to be executed on the unit, reducing code length and optimizing program memory. Finally, the RTL is passed in the HW measurement process again to obtain the area and power information(❺).

IV. RESULTS & ANALYSIS

A. Experimental Setup

To evaluate our proposed methodology, we assess the performance of four models: MLP-C, MLP-R, SVM-C, and SVM-R, which are trained on the Cardiography, RedWine, and WhiteWine datasets from the UCI repository [17]. The models are trained using the scikit-learn library, employing randomized parameter optimization (RandomizedSearchCV) with a 5-fold cross-validation procedure. Input features are normalized to the range [0, 1], and the data is split into training and test sets with a 70%/30% ratio. For the MLP models, a single hidden layer with up to five neurons is used, and the ReLU activation function is applied. The SVM models use a linear kernel, with SVM-C models implementing a one-vs-one classification strategy. The architecture of each MLP is configured to use the minimal number of hidden nodes while ensuring that all MLPs achieve near-max accuracy.

TABLE I

PERCENTAGE AREA-POWER GAINS, AVERAGE SPEEDUP AND ERROR OF
BESPOKE ZERO-RISCY(ZR) CORE. **B** IS BESPOKE AND **P** IS PRECISION.
PRECISION IMPLEMENTATIONS UTILIZE PARALLELIZATION UP TO 32 BITS

Cores	Area	Power	Speedup	Accuracy Loss
ZR B	10.6%	11.4%	0%	0.0%
ZR B MAC 32	8.2%	14.4%	23.93%	0.0%
ZR B MAC P16	22.2%	23.6%	33.79%	0.0%
ZR B MAC P8	29.3%	28.7%	41.73%	0.5%
ZR B MAC P4	36.5%	34.1%	46.4%	15.66%

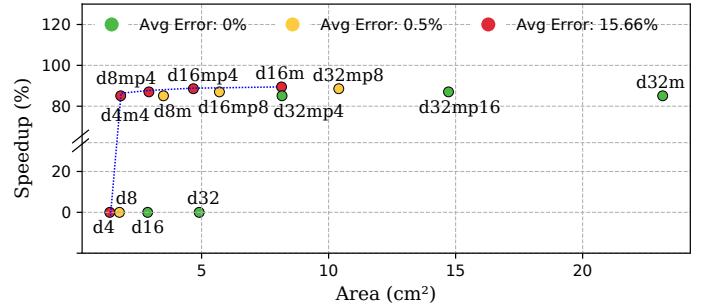Fig. 4. Average Accuracy Loss per Model introduced by each Precision Option

Fig. 5. Scatterplot of TP-ISA configurations, where **d** is the bits of the datapath, **m** signifies that the proposed MAC unit is implemented (with d bits) and **p** is the precision of the unit (lack of p means that the precision is the standard of the core and so there is no parallelization). The Pareto Front for Area and Speedup is highlighted in blue.

TABLE II
GAINS, AVERAGE SPEEDUP AND ERROR OF BESPOKE 8-BIT TP-ISA.
PARETO SOLUTION, MAINTAINING RELATIVELY LOW AVERAGE ERROR
INCREASE

Configuration	TP-ISA 8-BIT MAC
Area Overhead	x1.98
Power Overhead	x1.82
Avg Err (Base is 0.5%)	0.5%
Estimated Speedup	**up to 85.1%**

For RTL simulation of the two cores that underwent our proposed methodology, we use Modelsim with the compiled executable for each model. After applying the optimizations of our approach, we implement MAC units with various levels of precision for each core. The smallest 4-bit TP-ISA is realized with a 4-bit MAC unit and no parallelization, as the bitwidth is insufficient to support it. The 32-bit TP-ISA and Zero-Riscy were assessed with MACs of 4-bit, 8-bit, 16-bit, and 32-bit precision, with the maximum parallelization allowed for 16 bits or less, as shown in Figure 2.

B. Evaluation

Table I shows the exact gains and error trade-offs of our methodology, for several levels of precision, on the proof-of-concept Zero-Riscy compared to the baseline. Since all the models' parameters are 16-bits, we observe that we can parallelize the unit with 16 bit accuracy, gaining in all fronts and sacrificing no accuracy. In more detail, Figure 4 shows no error from 32 down to 16 bits, a small increase for 8 bits and a jump for most models at 4 bits, reaching a prohibitive 26% for RedWine. Balancing the trade-offs, 8-bit precision appears to be a suitable compromise, introducing just 0.5% average decrease in accuracy.

Figure 5 shows all base and generated TP-ISA [1] designs with the blue line highlighting the Area-Speedup Pareto curve. This curve remains similar even when considering power, as area and power exhibit a near-linear correlation in these examples. The lower-left group of points corresponds to the baseline cores, achieving no speedup, while the upper-side implementations are generated through the proposed methodology. Speedup increases rapidly when using a MAC unit and then slowly with SIMD. Table II shows a Pareto solution that achieves substantial speedup with minimal accuracy degradation of 0.5% and overhead factors of 1.98x and 1.82x for area and power, remaining still well within printed batteries'

capabilities while vastly speeding up execution time by 85.1% compared to [1].

When considering printed memories, we observe that (a)architectures with smaller bitwidths benefit more in terms of memory overheads due to direct reduction of cells per addressable space, (b)architectures that support multiplication save up to 11.1% on memory, as the multiplication instructions do not need to be scheduled for ALU being directly replaced with a single MUL command and (c)configurations that employ SIMD can introduce additional savings of up to 1-2% by calculating entire neurons in a single pass, without requiring additional control instructions for loops.

V. CONCLUSION

Printed electronics emerge as a promising technology, enabling low-cost, lightweight, battery powered applications, especially when paired with the flexibility of general purpose processors. However, the advantages of printed processors come at the cost of prohibitively large, power hungry and lower performance circuits. Our work explores the trade-offs of the printed computing design space and demonstrates that bespoke optimizations on low power processors, targeting ML applications, achieves consistent improvements in performance and hardware characteristics while offering a series of Pareto-optimal solutions considering trade-offs of area, power, speedup and accuracy loss for printed electronics use cases.

ACKNOWLEDGMENT

REFERENCES

[1] N. Bleier, M. Mubarik, F. Rasheed, J. Aghassi-Hagmann, M. B. Tahoori, and R. Kumar, "Printed microprocessors," in *Annu. Int. Symp. Computer Architecture (ISCA)*, jun 2020, pp. 213–226.

[2] H. A. Hobbie, J. L. Doherty, B. N. Smith, P. Maccarini, and A. D. Franklin, "Conformal printed electronics on flexible substrates and inflatable catheters using lathe-based aerosol jet printing," *npj Flexible Electronics*, vol. 8, no. 1, p. 54, 2024.

[3] H. Cherupalli, H. Duwe, W. Ye, R. Kumar, and J. Sartori, "Bespoke processors for applications with ultra-low area and power constraints," in *Annu. Int. Symp. Computer Architecture (ISCA)*, 2017, pp. 41–54.

[4] E. Ozer, J. Kufel, J. Biggs, G. Brown, J. Myers, A. Rana, A. Sou, and C. Ramsdale, "Bespoke machine learning processor development framework on flexible substrates," in *2019 IEEE International Conference on Flexible and Printable Sensors and Systems (FLEPS)*, 2019, pp. 1–3.

[5] G. Armeniakos, A. Maras, S. Xydis, and D. Soudris, "Mixed-precision neural networks on risc-v cores: Isa extensions for multi-pumped soft simd operations," *arXiv preprint arXiv:2407.14274*, 2024.

[6] G. Cadilha Marques, S. K. Garlapati, S. Dehm, S. Dasgupta, H. Hahn, M. Tahoori, and J. Aghassi-Hagmann, "Digital power and performance analysis of inkjet printed ring oscillators based on electrolyte-gated oxide electronics," *Applied Physics Letters*, vol. 111, no. 10, p. 102103, 2017.

[7] T. Lei, L.-L. Shao, Y.-Q. Zheng, G. Pitner, G. Fang, C. Zhu, S. Li, R. Beausoleil, H.-S. P. Wong, T.-C. Huang *et al.*, "Low-voltage high-performance flexible digital and analog circuits based on ultrahigh-purity semiconducting carbon nanotubes," *Nature communications*, vol. 10, no. 1, p. 2161, 2019.

[8] H. Çeliker, W. Dehaene, and K. Myny, "Dual-input pseudo-cmos logic for digital applications on flexible substrates," in *European Solid State Circuits Conference (ESSCIRC)*, 2021, pp. 255–258.

[9] D. D. Weller, M. Hefenbrock, M. B. Tahoori, J. Aghassi-Hagmann, and M. Beigl, "Programmable neuromorphic circuit based on printed electrolyte-gated transistors," in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2020, pp. 446–451.

[10] J. Biggs, J. Myers, J. Kufel, E. Özer, S. Craske, A. Sou, C. Ramsdale, K. Williamson, R. Price, and S. White, "A natively flexible 32-bit arm microprocessor," *Nature*, vol. 595, pp. 532–536, 07 2021.

[11] G. Armeniakos, G. Zervakis, D. Soudris, M. B. Tahoori, and J. Henkel, "Co-design of approximate multilayer perceptron for ultra-resource constrained printed circuits," *IEEE Trans. Comput.*, pp. 1–8, 2023.

[12] ——, "Model-to-circuit cross-approximation for printed machine learning classifiers," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 2023.

[13] E. Özer, J. Kufel, J. Myers, J. Biggs, G. Brown, A. Rana, A. Sou, C. Ramsdale, and S. White, "A hardwired machine learning processing engine fabricated with submicron metal-oxide thin-film transistors on a flexible substrate," *Nature Electronics*, vol. 3, pp. 1–7, 07 2020.

[14] D. D. Weller, N. Bleier, M. Hefenbrock, J. Aghassi-Hagmann, M. Beigl, R. Kumar, and M. B. Tahoori, "Printed stochastic computing neural networks," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2021, pp. 914–919.

[15] B. Zhai, S. Pant, L. Nazhandali, S. Hanson, J. Olson, A. Reeves, M. Minuth, R. Helfand, T. Austin, D. Sylvester *et al.*, "Energy-efficient subthreshold processor design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 8, pp. 1127–1137, 2009.

[16] C. Torres-Huitzil and B. Girau, "Fault and error tolerance in neural networks: A review," *IEEE Access*, vol. PP, pp. 1–1, 08 2017.

[17] K. N. Markelle Kelly, Rachel Longjohn, "The uci machine learning repository." [Online]. Available: https://archive.ics.uci.edu