End-to-end Hyperdimensional Computing with 24.65 µJ per Training Sample in 22 nm Technology

Wei-Ji Chao*†, Paul R. Genssler*, Sandy A. Wasif*, Albi Mema* and Hussam Amrouch*
*Chair of AI Processor Design; Munich Institute of Robotics and Machine Intelligence
Technical University of Munich; TUM School of Computation, Information and Technology, Munich, Germany
†Department of Electrical Engineering, National Sun Yat-sen University, Taiwan
amrouch@tum.de

***Abstract*—Hyperdimensional computing (HDC) is an emerging computing paradigm promising to replace classical machine learning algorithms for edge training. This work proposes FixedHD as an end-to-end efficient and generic fixed-point HDC for edge AI. To efficiently deploy FixedHD, a chip is fabricated with an extended RISC-V. The extensions include vector computation and customized instructions. The chip, fabricated in 22 nm, can perform a training iteration for the UCI-HAR dataset with 24.65 µJ energy consumption in 93 ms per sample at 20 MHz. Our chip is capable of operating up to 120 MHz, which decreases the training time per sample to 15.6 ms. Our custom instructions provide a 4x speedup compared to the baseline processor.**

***Index Terms*—Hyperdimensional Computing, RISC-V, Edge AI, Energy Efficient, Real Time**

I. Introduction

Hyperdimensional computing (HDC) is an emerging computational paradigm that holds the promise of bringing AI training to the edge with success in multiple applications [1]. Fine-tuning AI models at the edge is essential for subject-specific medical applications involving wearables [2]. HDC processes data as patterns rather than absolute values, replacing complex computations with lightweight operations more suitable for constrained devices. The representation of the HDC model depends on the application specifications. Binary HDC architectures are implemented using simple hardware, therefore consume less power [3]. However, accuracy is restricted, and the dimensionality increases to compensate for lost representation. On the other hand, floating-point HDC offers accuracy comparable to neural networks at the expense of significant design complexity. In this work, fixed-point representation is employed to balance the contradicting demands of inference accuracy and design complexity.

In the literature, different implementation platforms for HDC were explored, such as GPUs, FPGAs, ASICs, and custom processors [1]. Customized processors as a platform for HDC benefit from design flexibility, scalability, generality, compactness, and efficiency [4]; however, these explorations were only conducted for binary HDC. RISC-V customization for acceleration includes vector extensions to benefit from the fact that HDC operation relies on multiple computations over many independent data items.

In this work, a 1 mm^2 chip featuring a RISC-V processor with vector extensions is fabricated in 22 nm to deploy a generic HDC model for complete end-to-end operation, including training, retraining, and inference. A training sample takes 93 ms with an energy consumption of 24.65 µJ for the UCI-HAR dataset at a 20 MHz clock frequency. At a higher frequency of 120 MHz, the energy consumption increases by 2.5x to 60.71 µJ but the time is reduced by 5.9x to 15.6 ms.

II. Our Proposed HDC FixedHD Model

The computation of the proposed FixedHD model is divided into three stages: training, retraining, and inference, as shown in Fig. 1. One common module in all stages is encoding, which converts data with n-features into hypervectors with d-dimensions. This non-linear mapping process involves matrix multiplication between the data feature vector (F) and a randomly generated matrix (B) with standard normal distribution. Encoding consumes most of the computation time as it scales with features and dimensionality. The encoded sample is then normalized using trigonometric functions. The data is represented in fixed-point format to balance inference accuracy with computational complexity. Fixed-point format can maintain same inference accuracy as floating-point representation while performing simple computations on integer data. For the proposed 16-bit fixed-point model, a detailed bit distribution analysis is conducted to fit various applications such as MNIST and Isolet into the HDC model. The trigonometric functions for normalization are realized with codebooks to replace the computation with memory access. To benefit from HDC resiliency against errors, the codebooks are quantized to reduce memory requirements with minimal effect on accuracy. The first stage is training, which is the process of model creation, conducted through two phases: initial model creation through hypervector bundling, followed by adaptive training through cosine similarity to fine-tune the model. Adaptive training improves the model's accuracy and achieves faster convergence [5]. The second stage is retraining to enhance the model through a few extra iterations of adaptive model tuning across the complete training dataset. In FixedHD, convergence is achieved for multiple datasets with only 10 retraining iterations. The final stage is the inference that tests the accuracy of the model against unseen test data items. The same encoding and cosine similarity modules are used again in inference.

The model is implemented in C code and is compiled for a 32-bit RISC-V architecture that supports integer operations, including multiplications. Software optimizations are applied

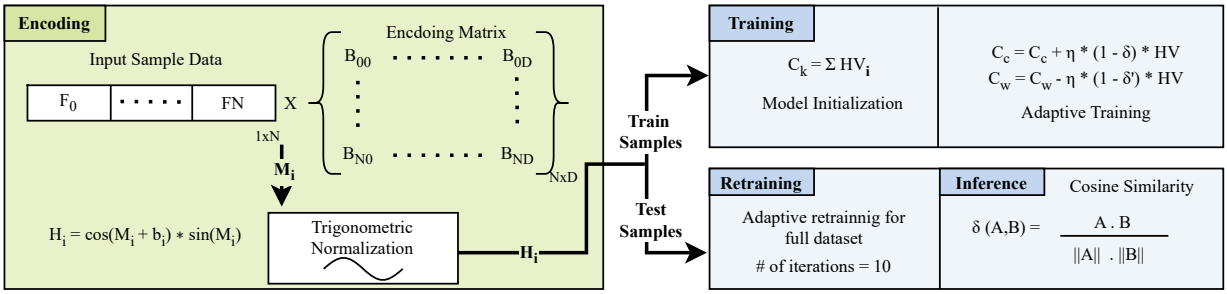


Figure 1: Overview of the FixedHD Model highlighting the main computational stages: training, retraining, and inference. Encoding is a common step for all stages, it contains the normalization through trigonometric functions to produce hypervector H. Training initializes the model through addition of hypervector belonging to the same class (C). The adaptive training applies for wrongly predicted only, where the wrongly predicted hypervector is added to the correct class and subtracted from the wrong class scaled with the cosine similarity δ and the learning rate η. Cosine similarity is the metric utilized in the inference stage.

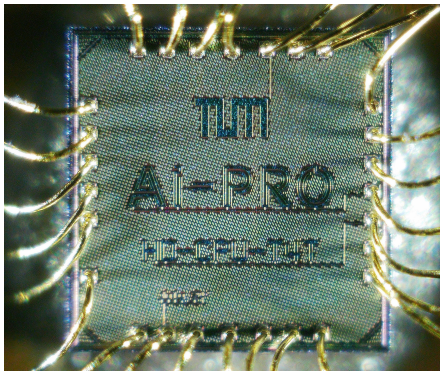


Figure 2: Fabricated 1 mm^2 chip for customized RISC-V processor with the 32 KB instruction memory and 128 KB data memory

to achieve a compact model suitable for edge applications. These optimizations include approximate codebooks to replace trigonometric functions, and compressed encoding matrix through rotational accessing. The model can fit various datasets, such as MNIST, FMNIST, UCI-HAR, and Isolet, with comparable accuracy; the details of the implementation and results of FixedHD are presented in our previous work [6].

III. Chip Architecture

The proposed chip contains the vector-extended RISC-V processor with separate instruction and data memories. Additionally, there is an AXI-lite inspired bus for communication that is controlled through a UART master module. Multiple registers are utilized to configure the chip and monitor performance. The fabricated chip is presented in Fig. 2.

A. Core processor

The 32-bit RISC-V processor architecture is shown in Fig. 3. It is provided by Synopsys ASIP designer based on the RVIM32 instruction set [7]. It consists of five standard pipeline stages, an ALU, and a register file with 32 registers. To accelerate the encoding step in both the training and inference stages, the processor is extended to support SIMD (Single Instruction Multiple Data) architecture. This is accomplished by introducing a vector register file, vector pipeline stages, and customized vector instructions. The vector register file consists of eight registers, each holding 128-bit data. The customized vector instruction is a VMAC-shift, combining three processes: multiplication, accumulation, and shifting on four data items in one clock cycle. To implement this instruction, the hardware is extended with four 16-bit MAC units that have direct access to the vector register file through vector pipeline stages. The data access for vector instructions is 128-bits per clock cycle.

B. Memories

The chip contains two memories for instructions and data with sizes of 3072x64 (32 kB) and 8192x128 (128 kB), respectively. They support standard one read port and one write port. The memories are loaded through UART with data and instructions before running any application on the core processor.

C. Control Registers

There are 16 control registers that can be read or written through the bus and three registers that are read-only by the bus. The registers are used to change the chip configurations, such as the baud rate for UART communication and the reset signal for the processor. They are also used to monitor outputs from the processor, for example, the number of clock cycles to execute a test program.

D. Communication protocol

The communication between the peripherals, memories, and registers is performed through an AXI-lite inspired bus. This bus has a width of 32 bits and can only be accessed by one entity at a time, the bus master is controlled by the UART.

E. Printed Circuit Board (PCB)

Two PCBs are designed to connect the chip with the necessary supplies and equipments as shown in Fig. 4. The first PCB is connected to an external clock generator that is capable of supplying an input clock up to 100 MHz. In this version of the PCB, the chip is also connected to an external voltage supply to power on the chip and peripheral devices. Additionally an external USB to UART interface is connected to complete the setup. The second PCB, has an on-board clock generator that reaches up to 120 MHz and works at a constant voltage supply of 0.8 V. Both PCBs contain extra

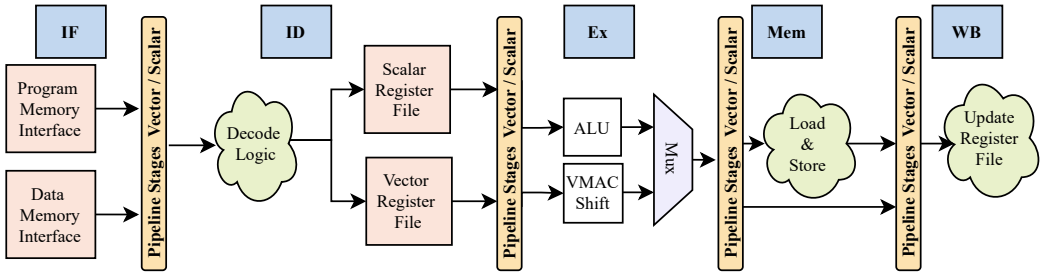
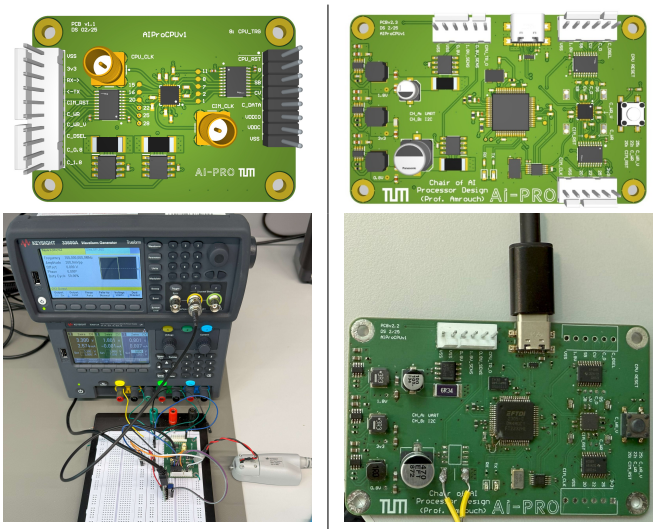


Figure 3: Processor Architecture with vector extensions added to the data-path.



(a) (b)

Figure 4: (a) PCB version 1 with external clock generator and power supply. (b) PCB version 2 with on-board clock generator and power supply.

ICs that are added to facilitate power measurements as well as communication.

IV. Chip Evaluation

A. Experimental Setup

The evaluation is conducted to verify the functionality under different operating conditions and for various test applications. The test applications are prepared by generating the data memory and program memory binary files to be loaded onto the chip memories. This step is performed using Synopsys ASIP designer [7]. Chip specifications are listed in Tab. I.

Table I: Chip specifications.

Fabrication Technology	GF 22 nm FDSOI
Die Size	1 mm^2
Voltage level	0.55 – 1.0 V
Frequency	10 – 120 MHz
Gate Count	1.97 M

B. Energy Measurements

Energy analysis is conducted for HDC in different configurations, all the measurements are for an average of 10 samples from the UCI-HAR dataset [12]. The dimensions used are 1408 for FixedHD and 4096 for Binary HDC. The reported inference

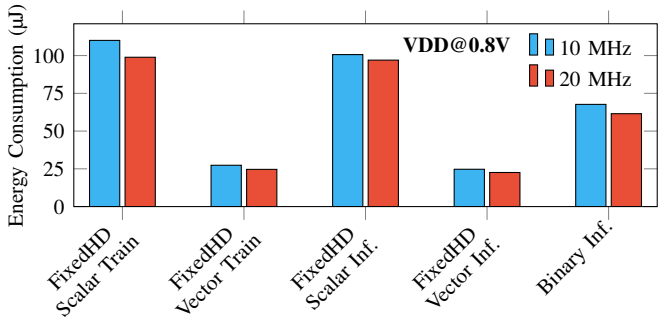


Figure 5: Energy Consumption for various frequencies at constant voltage 0.55 V. Vector operations reduces energy consumption significantly even compared to binary HDC.

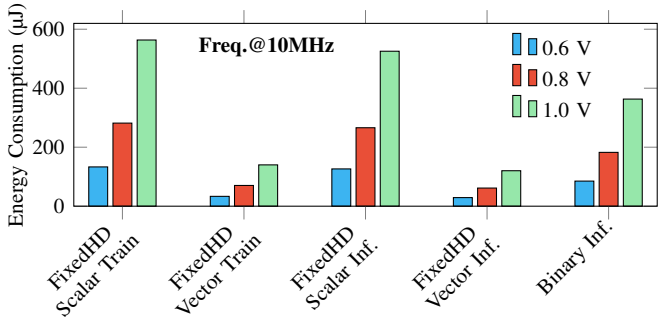


Figure 6: Energy Consumption for various voltage levels at constant frequency 10 MHz. The lowest energy consumption is observed at 0.6 V. however, larger voltages can be utilized for higher frequencies.

accuracy with 10 retraining epochs is 94.03 %, 91.72 %, and 89.31 % for scalar FixedHD, vector FixedHD, and binary HDC, respectively. In Fig. 5, energy is reported for different frequencies at the voltage of 0.55 V. The energy consumption for FixedHD with VMAC-shift utilization is reduced by 4x when compared to scalar FixedHD, due to the speedup of 4 VMAC-shift units operating in parallel. The FixedHD with VMAC-shift also consumes less energy than binary HDC by almost 2.5x. At 20 MHz configuration, the vector training sample consumes 24.65 µJ with a computation time of 93 ms. Another analysis of energy consumption with different operating voltages is shown in Fig. 6. The larger voltage is required for high clock frequency like 120 MHz, but for lower frequencies, 0.55 V is very suitable with the least power consumption. For example, a single training iteration takes only 15.6 ms at 120 MHz but the energy consumption is around 60.71 µJ as the supply voltage is 0.8 V.

Table II: Comparison with state of the art HDC

	Risc-HD [4]	Edge-side [8]	BRIC [9]	Human Centric IoT [3]	E^3HDC [10]	TinyHD [11]	Proposed FixedHD Scalar	Proposed FixedHD Vector
Platform	FPGA	FPGA	FPGA	ASIC	FPGA	ASIC	ASIC	ASIC
Technology	Vritex-7	Zynq-7000	Kintex-7	28 nm	PYNQ-Z2	22 nm	22 nm	22 nm
Representation	Binary	Binary	Binary	Binary	Binary	8 Bit Int	16 bit Int	16 bit Int
Application	Isolet	MNIST	Isolet	Lang	Isolet	UCIHAR, k*=12	UCIHAR, k*=6	UCIHAR, k*=6
Energy (µJ)	50	N/A	0.1	0.25	1.9	0.2	96.99	22.58
Power (mW)	753	630	N/A	267	308	10	0.264	0.296
Time (µs)	72.28	1,103	0.3	N/A	6.17	22.85	367 * 10^3	76.3 * 10^3
Accuracy (%)	86	97.18	93	90	85.18	95.5	94.03	91.72
Frequency (MHz)	83	N/A	N/A	417	100	400	20	20
Voltage (V)	N/A	N/A	N/A	0.8	N/A	1.1	0.55	0.55
Area (mm^2)	N/A	N/A	N/A	1.27	N/A	0.496	1.0	1.0
Support Training	×	✓	✓	✓	✓	×	✓	✓

* *k stands for number of classes*

C. Comparison with state-of-the-art HDC

Comparison with related work from the literature is presented in Tab. II for two different configurations: scalar and vector inference. This work has the lowest power consumption as compared to all other work listed in Tab. II. However, the energy is relatively high due to longer sample processing time. It is also worth noting that other works validated across FPGAs which have much lower energy consumption [9, 10]. However, FPGAs are unsuitable for edge applications as their area is too large and with considerable power consumption.

V. Conclusion

In this work, a modified RISC-V processor is designed, fabricated, and measured. for HDC acceleration with vector extensions and fixed-point operation. The optimized chip is generic, and various applications can benefit from the optimizations. Results show that a training sample with fixed-point HDC representation takes 93 ms and 24.65 µJ for 20 MHz and 0.55 V.

Acknowledgment

Authors would like to thank Daniel Spitze from TU Munich for his contribution in developing the measurement PCBs.

References

[1] C.-Y. Chang, Y.-C. Chuang, et al., "Recent progress and development of hyperdimensional computing (hdc) for edge intelligence," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 13, no. 1, pp. 119–136, 2023.

[2] M. R. Islam, D. Massicotte, et al., "Surface emg-based intersession/intersubject gesture recognition by leveraging lightweight all-convnet and transfer learning," IEEE Transactions on Instrumentation and Measurement, vol. 73, pp. 1–16, 2024.

[3] S. Datta, R. A. Antonio, et al., "A programmable hyper-dimensional processor architecture for human-centric iot," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 9, no. 3, pp. 439–452, 2019.

[4] F. Taheri, S. Bayat-Sarmadi, et al., "Risc-hd: Lightweight risc-v processor for efficient hyperdimensional computing inference," IEEE Internet of Things Journal, vol. 9, no. 23, pp. 24 030–24 037, 2022.

[5] M. Imani, J. Morris, et al., "Adapthd: Adaptive efficient training for brain-inspired hyperdimensional computing," in *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, IEEE, 2019, pp. 1–4.

[6] S. A. Wasif, M. Wael, et al., "Domain-specific hyperdimensional risc-v processor for edge-ai training," IEEE Transactions on Circuits and Systems I: Regular Papers, pp. 1–14, 2025.

[7] Synopsys. "Asip designer." (2023), [Online]. Available: https://www.synopsys.com/dw/ipdir.php?ds=asip-designer.

[8] T. Yu, B. Wu, et al., "Fully learnable hyperdimensional computing framework with ultratiny accelerator for edge-side applications," IEEE Transactions on Computers, vol. 73, no. 2, pp. 574–585, 2023.

[9] M. Imani, J. Morris, et al., "Bric: Locality-based encoding for energy-efficient brain-inspired hyperdimensional computing," in *Proceedings of the 56th Annual Design Automation Conference 2019*, 2019, pp. 1–6.

[10] M. S. Roodsari, J. Krautter, et al., "E 3 hdc: Energy efficient encoding for hyper-dimensional computing on edge devices," in *2024 34th International Conference on Field-Programmable Logic and Applications (FPL)*, IEEE, 2024, pp. 274–280.

[11] B. Khaleghi, H. Xu, et al., "Tiny-hd: Ultra-efficient hyperdimensional computing engine for iot applications," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 2021, pp. 408–413.

[12] D. Anguita, A. Ghio, et al., "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Ambient Assisted Living and Home Care: 4th International Workshop, IWAAL 2012, Vitoria-Gasteiz, Spain, December 3-5, 2012. Proceedings 4*, Springer, 2012, pp. 216–223.