HPU: A 168pJ/op Multi-Purpose and Multi-Dimension Binary Hyper-Dimensional Processing Unit for Energy-Efficient Machine Learning

Youbin Kim[1], Mohamed Ibrahim[1,2], and Jan M. Rabaey[1]
Email: youbin_kim@berkeley.edu, mibrahim81@gatech.edu, jan_rabaey@berkeley.edu
[1]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA, USA
[2]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA

***Abstract*—Recent interest in hyper-dimensional computing (HDC) has brought forth a wide variety of algorithms and applications well-suited for low-power edge AI. We present the Hyper-Dimensional Processing Unit (HPU), a digital processor optimized for energy-efficient acceleration of all-known binary HDC algorithms, achieving 168 pJ/op. The HPU features a novel architecture with a programmable datapath and integrated pseudo-random vector generators, enabling versatile computation over vectors of varying dimensions. This architectural flexibility supports a broad class of binary HDC algorithms without sacrificing energy efficiency. Fabricated and benchmarked in 28nm CMOS, the HPU demonstrates competitive performance across three representative applications (language recognition (82.3 nJ/query), gesture recognition (10.4 nJ/query), and hyper-dimensional factorization (14.7 µJ/query)) that equal or surpass prior application-specific work.**

I. Introduction

Hyper-dimensional computing (HDC) [1] is a machine learning framework that has made inroads in low-power edge-AI applications. With simple bitwise vector operations and a small memory footprint, HDC has demonstrated improved energy efficiency for biosensing classification tasks when compared to conventional machine learning methods such as support vector machines and deep neural networks [2,3]. Recently, more complex HDC algorithms have emerged such as the factorization of visual representations [4] and robotics control [5]. Previous hardware [6-9] for HDC achieves impressive energy efficiency for specific tasks but face several issues with the growing application space (Fig. 1). First, these processors are designed with custom datapaths for a single algorithm (i.e. classification or factorization) and therefore cannot support others. Second, prior designs operate on hyper-dimensional (HD) vectors of a fixed dimension. However, HD vector dimensionality presents a trade-off between application performance and energy efficiency, and the optimal dimension depends on the specific application and use-case.

We present the Hyper-Dimensional Processing Unit (HPU) as a digital energy-efficient processor for HDC with two key innovative features: 1) A flexible instruction set architecture that enables arbitrary sequences of vector operations to be programmed using any stored or computed vector. 2) On-chip pseudo-random vector generators and datapaths with scalable dimensionality that allow for computation over vectors of multiple dimensions. With these features, the HPU can accelerate and adapt to all known binary HDC algorithms while maintaining competitive energy efficiency. We fabricated and characterized two versions of the HPU architecture: HPUv1, a baseline proof of concept of the proposed design, and HPUv2, an implementation optimized for power and performance.

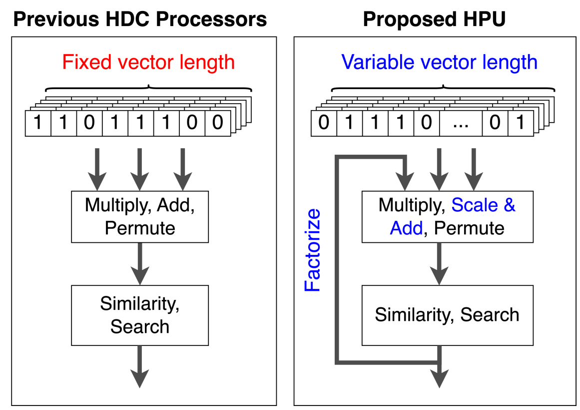


Fig. 1. Limitations of previous hardware for hyper-dimensional computing (HDC).

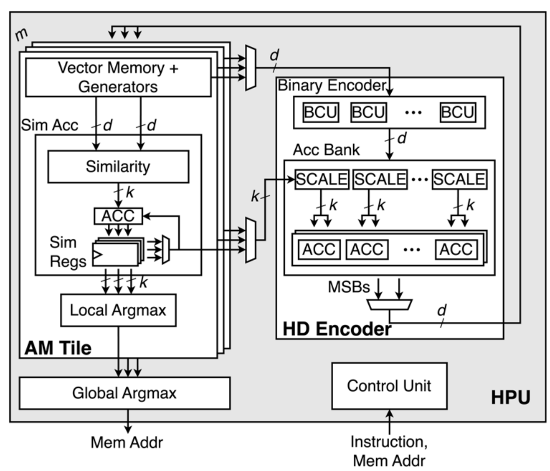


Fig. 2. Architecture diagram of proposed Hyper-Dimensional Processing Unit

II. The Proposed Processor

The HPU architecture (Fig. 2) contains a single HD encoder interconnected with multiple associative memory (AM) tiles.

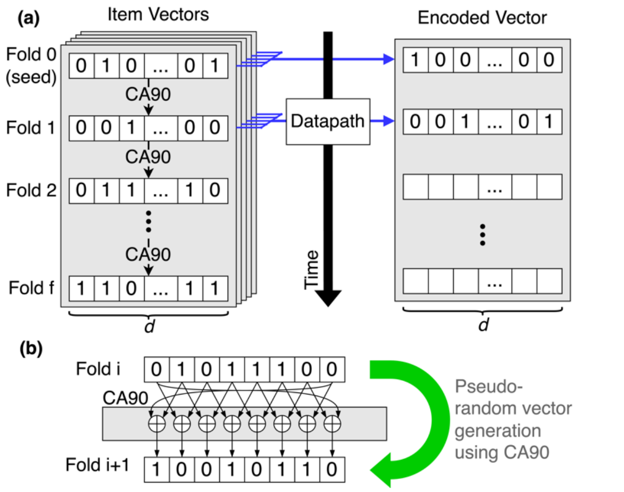


Fig. 3. HPU dimensionality scaling mechanism via (a) vector folding and (b) cellular automata 90 (CA90).

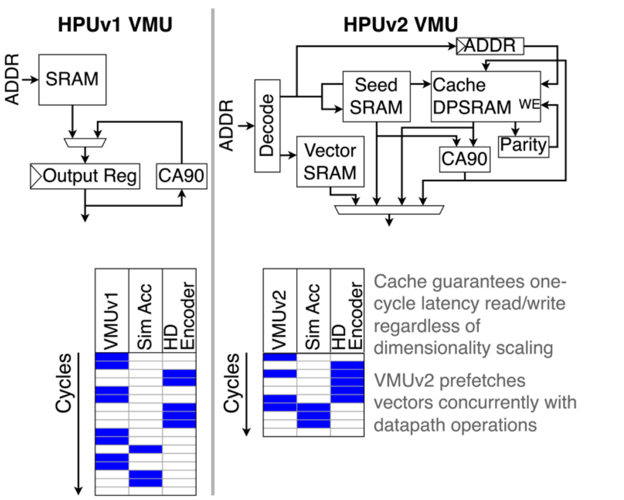


Fig. 4. Vector memory unit (VMU) with CA90 cache and resulting instruction efficiency improvement.

A. *HPU Programmability*

The HD encoder performs all vector-to-vector HDC operations, namely elementwise multiply (XOR), accumulate, and permute, using an array of one-bit datapaths. Vector-to-integer operations, such as vector similarity and associative search, are handled by the AM tiles. While vector encoding is inherently sequential, similarity computations can be parallelized across multiple AM tiles. To meet the high memory-bandwidth requirements, each tile includes a local SRAM-based binary vector memory unit (VMU) which stores both random item vectors and encoded vectors from the HD encoder. The distributed VMU architecture enables high-bandwidth, parallel vector reads for similarity computations, while also supporting serialized vector access for the HD encoder's read and write operations.

A novel addition that further enhances the HPU's programmability is the scaled accumulate feature. In addition to using stored vector similarities for associative search, they can also scale a vector in the HD encoder prior to accumulation as a weighted sum. This enables the HPU to perform matrix-vector multiplications required by more complex algorithms, such as

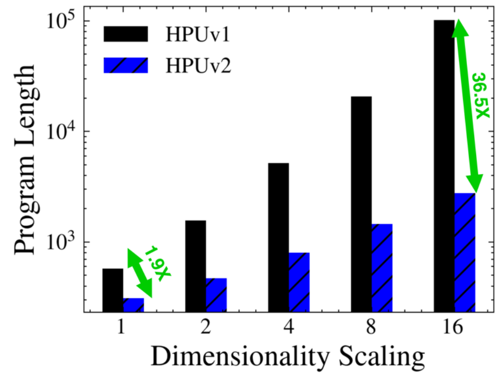


Fig. 5. Impact of instruction efficiency optimization from HPUv1 to HPUv2 on program length (i.e. number of instructions) for varying dimensionality scaling.

factorization. Combined with the ability to store and reuse encoded vectors, the HPU can be programmed to execute any ordering of all known binary HDC operations.

B. *Vector Dimensionality Scaling*

The main innovation of the HPU is its dimensionality scaling capability. While the vector memory and datapaths operate on a fixed dimension d, the HPU can scale the effective vector dimension to any integer multiple of d. Because all HD vector operations are elementwise, they can be time-multiplexed across smaller d-dimensional sub-vectors, or folds (Fig. 3a). This allows the HPU to process vectors with dimensions equal to any multiple of the native datapath width. The VMUs contain pseudo-random generators that only store a random seed of length d for each item vector. When the vector dimension is scaled, the seed generates pseudo-random bit sequences using cellular automata rule 90 (CA90), which is applied recursively to populate the folds of the extended vectors (Fig. 3b).

For HPUv1, the VMU includes a unified SRAM that stores both item and encoded vectors as well as a serial CA90 module. When using many vector folds however, the CA90 update bottlenecks the execution time of HD algorithms. Therefore, HPUv2 implements a novel CA90 cache which reduces VMU latency to one cycle (Fig. 4). The DPSRAM cache stores the last CA90 update and an additional parity bit which is used to automatically perform an update and write-back when needed. For the same benchmark, HPUv2 exhibits a 1.9x-36x program instruction reduction vs HPUv1 for 1-16 vector folds respectively (Fig. 5), which corresponds to vastly improved benchmark latency and energy.

The architecture parameters for HPUv2 were selected for energy efficiency based on benchmark simulations. Compared to the 4 AM tiles and 512 native datapath dimension of HPUv1, HPUv2 achieves optimal power and performance with 2 AM tiles and a datapath dimension of 1024.

III. Measurements and Comparisons

HPUv1 and HPUv2 were fabricated in TSMC28nm CMOS with die area of 2.7mm^2 and 4.5mm^2 respectively, each with 262KB of SRAMs for vector memory. Die micrographs and chip summaries are shown in Fig. 9. Both share a similar floorplan featuring a central HD encoder with the AM tiles placed above and below.

TABLE I. Comparison of HDC kernel power, latency, and energy measured on GPU (NVIDIA GTX 1080), HPUv1, and HPUv2

	Kernel	Multiply-Permute	Multiply-Accumulate	Associative Search	Average
GPU	Power	63.5W	106.1W	164.7W	
	Latency	2216ns	747ns	638ns	
	Energy/op	141uJ	79uJ	105uJ	108uJ
HPUv1	Power	0.58mW	0.60mW	1.07mW	
	Latency	1222ns	2111ns	387ns	
	Energy/op	561pJ	1274pJ	416pJ	750pJ
HPUv2	Power	0.76mW	0.83mW	0.80mW	
	Latency	175ns	246ns	208ns	
	Energy/op	133pJ	203pJ	167pJ	168pJ

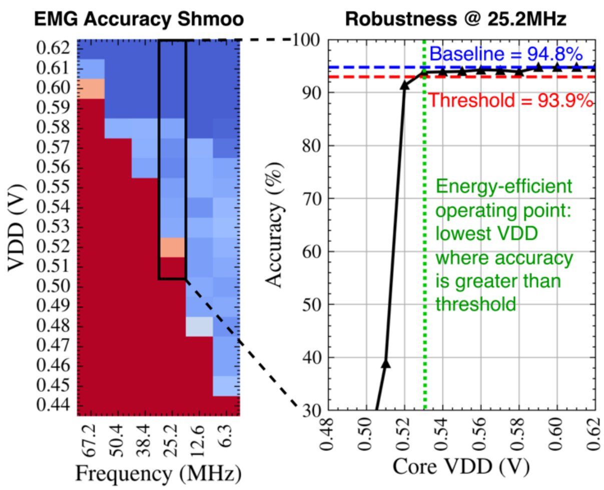


Fig. 6. Accuracy shmoo for the EMG gesture recognition benchmark on HPUv2 and corresponding low-VDD robustness of HDC.

HPUv2 achieves 168pJ/op at 0.45V-0.49V and 12.6MHz, a 4.5x improvement vs. HPUv1 (Tab. 1). The energy per operation is computed as an average over three common multi-cycle HPU kernel operations: multiply-permute, multiply-accumulate, and associative search. When compared vs. running the same kernel operations on an optimized GPU implementation using Torchhd [10] on a NVIDIA GTX1080, HPUv2 achieves six orders of magnitude better energy per operation.

Fig. 6 illustrates how the HPU can leverage the inherent robustness of HDC to operate at reduced core voltages while maintaining acceptable benchmark accuracy. For the EMG gesture recognition application, a voltage-frequency shmoo plot shows accuracy across test clock frequencies from 6.3 MHz to 67.2 MHz and core VDD from 0.44 V to 0.62 V. At 25 MHz, despite timing-induced bit errors at lower voltages, accuracy remains nearly unaffected down to 0.52 V, demonstrating the resilience of HDC to small amounts of injected error.

To characterize energy efficiency, we identify energy-optimal operating points, defined as the lowest VDD at which benchmark accuracy remains above 99% of the baseline, for each test frequency. At these points, we measure latency, power, and energy per query across three benchmarks (Fig. 7). Two of the benchmarks, EMG gesture recognition [11] and European language recognition [12], are classification algorithms while the third is a HD factorization algorithm [4]. Across all

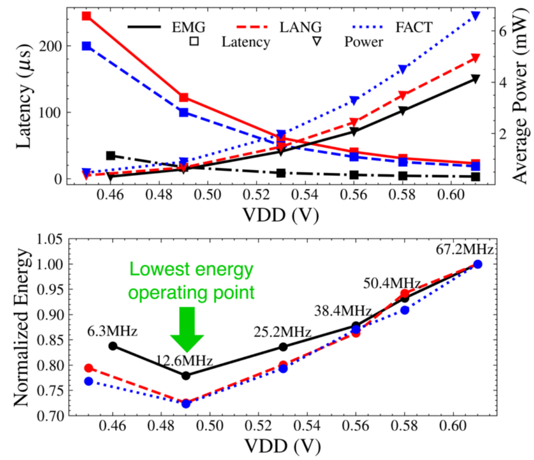


Fig. 7. Measured benchmark latency, power, and corresponding energy across energy-efficient operating points on HPUv2.

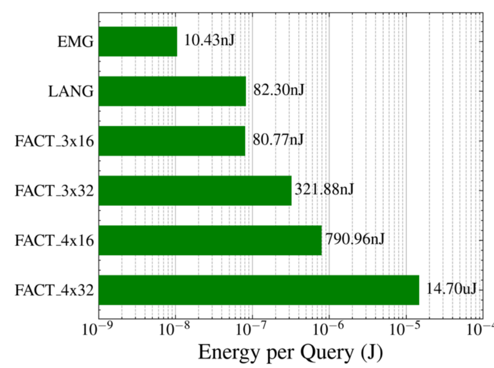


Fig. 8. Measured energy per query for EMG gesture recognition (EMG), European language recognition (LANG), and HD factorization (FACT) on HPUv2. FACT_AxB refers to A factors with B items per factor.

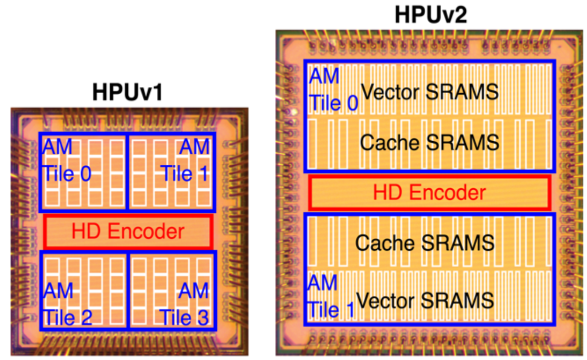
benchmarks, HPUv2 achieves minimum energy at 0.49 V and 12.6 MHz, with 10.4nJ/query for the EMG benchmark, 82.3nJ/query for the language benchmark, and between 80nJ-14uJ/query for the factorization benchmark depending on the search space (Fig. 8).

Tab. 2 compares HPUv2 against prior state-of-the-art HDC processors [6–9]. HPUv2 is the first processor to support and accelerate all known binary HD algorithms in hardware, as demonstrated by its ability to run both HD classification and factorization tasks. It is also the only processor capable of dynamically adjusting the vector dimension based on algorithmic complexity, using 1024 dimensions for the EMG benchmark, and scaling up to 16,384 for the factorization benchmark.

Compared to previous work, HPUv2 achieves best-in-class energy per query for language recognition and EMG gesture recognition with 4x and 2.5x less energy than the next best performer respectively. For the factorization benchmark, HPUv2 achieves 2.3x less energy than [9], although this comparison is not entirely equivalent due to [9] using a 16x larger search space. However, practical use cases of HD factorization, including those in [9], typically employ much smaller search spaces (e.g., 1200), which can be factorized on HPUv2 with 80.8 nJ/query.

TABLE II. Comparison with state-of-the-art processors for HDC

		Nature'20 [6]	TCAS-I'21 [7]	ESSCIRC'23 [8]	Nature'23 [9]	HPUv2
	Platform	65nm CMOS, 90nm RRAM	22nm CMOS	28nm CMOS	14nm PCM, 14nm CMOS	28nm CMOS
	Implementa-tion	IC (Not fully-integrated)	Post-layout	IC (Fully-integrated)	IC (Not fully-integrated)	IC (Fully-integrated)
	Applications	HD Classification	HD Classification	HD Classification	HD Factorization	**All known binary HD Algorithms**
LANG [12]	**HDC Dim**	10000	2048	–	–	**2048**
LANG [12]	**Accuracy**	92.8%	86%	–	–	93.1%
LANG [12]	**Latency**	–	14ms	–	–	122.6us
LANG [12]	**Energy/query**	430nJ*	332nJ*	–	–	**82.3nJ**
EMG [11]	**HDC Dim**	–	2048	2048	–	**1024**
EMG [11]	**Accuracy**	–	95%	92.9%	–	93.9%
EMG [11]	**Latency**	–	6.78ms	118us	–	17.3us
EMG [11]	**Energy/query**	–	191nJ*	25.6nJ	–	**10.4nJ**
FACT [4]	**HDC Dim**	–	–	–	256	**16384**
FACT [4]	**Search Space**	–	–	–	16,777,216	1,048,576
FACT [4]	**Accuracy**	–	–	–	99.7%	98.2%
FACT [4]	**Latency**	–	–	–	2.2ms	16.1ms
FACT [4]	**Energy/query**	–	–	–	33.1uJ*	**14.7uJ**



	HPUv1	HPUv2
Description	Baseline proof of concept	Power and performance optimized
Process	TSMC 28nm	TSMC 28nm
Size	1635um x 1635um	2110um x 2110um
Core VDD	0.43V - 0.60V	0.42V - 0.61V
Frequency	6.3MHz - 67.2MHz	6.3MHz - 67.2MHz
Datapath Dim	512	1024
AM Tiles	4	2
Total SRAM	262KB	262KB

Fig. 9. Chip micrographs and summary.

IV. Conclusion

In this paper, we presented the HPU, a fully programmable and energy-efficient digital processor for binary hyper-dimensional computing. Through architectural innovations including dimensionality scaling, on-chip pseudo-random vector generation, and scaled vector accumulation, HPUv2 supports a wide range of HDC algorithms, including both classification and factorization. Silicon measurements on 28nm CMOS demonstrate state-of-the-art energy efficiency across multiple benchmarks compared to previous application-specific hardware, which highlights HPUv2 as a general-purpose HDC processor for low-power edge AI applications.

Acknowledgment

Thank you to TSMC for funding this project and fabricating both ICs.

References

[1] P. Kanerva, "Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors," Cogn Comput, vol. 1, no. 2, pp. 139–159, Jun. 2009

[2] M. Imani, D. Kong, A. Rahimi, and T. Rosing, "VoiceHD: Hyperdimensional Computing for Efficient Speech Recognition," in 2017 IEEE International Conference on Rebooting Computing (ICRC), Nov. 2017, pp. 1–8.

[3] A. Burrello, L. Cavigelli, K. Schindler, L. Benini, and A. Rahimi, "Laelaps: An Energy-Efficient Seizure Detection Algorithm from Long-term Human iEEG Recordings without False Alarms," in 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), Mar. 2019, pp. 752–757

[4] E. P. Frady, S. J. Kent, B. A. Olshausen, and F. T. Sommer, "Resonator Networks, 1: An Efficient Solution for Factoring High-Dimensional, Distributed Representations of Data Structures," Neural Computation, vol. 32, no. 12, pp. 2311–2331, Dec. 2020

[5] P. Neubert, S. Schubert, and P. Protzel, "An Introduction to Hyperdimensional Computing for Robotics," Künstl Intell, vol. 33, no. 4, pp. 319–330, Dec. 2019

[6] G. Karunaratne, M. Le Gallo, G. Cherubini, L. Benini, A. Rahimi, and A. Sebastian, "In-memory hyperdimensional computing," Nat Electron, vol. 3, no. 6, pp. 327–337, Jun. 2020

[7] M. Eggimann, A. Rahimi, and L. Benini, "A 5 uW Standard Cell Memory-Based Configurable Hyperdimensional Computing Accelerator for Always-on Smart Sensing," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 10, pp. 4116–4128, Oct. 2021.

[8] S. Datta, B. Richards, H. Liew, Y. Kim, D. Sun, and J. M. Rabaey, "HDBinaryCore: A 28nm 2048-bit Hyper-Dimensional biosignal classifier achieving 25 nJ/prediction for EMG hand-gesture recognition," in ESSCIRC 2023- IEEE 49th European Solid State Circuits Conference (ESSCIRC), Sep. 2023, pp. 229–232.

[9] J. Langenegger, G. Karunaratne, M. Hersche, L. Benini, A. Sebastian, and A. Rahimi, "In-memory factorization of holographic perceptual representations," Nat. Nanotechnol., vol. 18, no. 5, pp. 479–485, May 2023

[10] M. Heddes et al., "Torchhd: An Open Source Python Library to Support Research on Hyperdimensional Computing and Vector Symbolic Architectures," Journal of Machine Learning Research, vol. 24, no. 255, pp. 1–10.

[11] A. Moin et al., "A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition," Nat Electron, vol. 4, no. 1, pp. 54–63, Jan. 2021

[12] A. Rahimi et al., "High-Dimensional Computing as a Nanoscalable Paradigm," IEEE Trans. Circuits Syst. I, vol. 64, no. 9, pp. 2508–2521, Sep. 2017