Improved Data Encoding for Emerging Computing Models: From Stochastic Computing to Hyperdimensional Computing

Author's Name: *Mehran Shoushtari Moghadam*
Affiliation: *Electrical, Computer, and Systems Engineering Department, Case Western Reserve University*
Postal Address: *321 Glennan Building, 2123 Martin Luther King Jr. Dr. Cleveland, OH 44106*
E-mail Address: *mxs2254@case.edu*
Research Advisor's Name: *Assistant Professor M. Hassan Najafi*
ACM student member number: *9133783*
Category: *Graduate*

I. Problem and Motivation

Proper encoding of data has always been a challenging task as the rudimentary step in emerging computing models. Stochastic Computing (SC) and Hyperdimensional Computing (HDC) have emerged as two promising computing paradigms for efficient hardware design of machine learning systems. Both SC and HDC deal with long streams of '0's and '1's instead of conventional binary values (with positional encoding or bit-significance) as the basic computational elements. Bit-streams ($\mathcal{BS}$s) and hypervectors ($\mathcal{HV}$s) (without bit-significance) are the primitive components acting as atomic data elements in SC and HDC, respectively. These atomic data elements are generated by using a proper source of randomness. The state-of-the-art (SOTA) approaches utilize pseudo-randomn number generators as the source of randomness to generate the $\mathcal{BS}$s [1], [2] and $\mathcal{HV}$s [3]–[6].

Providing high-quality randomness through employing a proper random number generator (RNG) is crucial to achieve the desired throughput and efficient hardware design for both SC and HDC systems. In contrast to the pseudo-random sequences (e.g. Linear Feedback Shift Register; *LFSR*), quasi-random sequences (e.g. Sobol, Halton) provide homogeneous distribution. These sequences enjoy the *low-discrepancy* (LD) property. Discrepancy is defined as the amount of deviation of the sequence points from the uniformity. A lower discrepancy of the sequence points results in improved uniformity. Figs. 1(a)-(d) compare the sparsity and distribution of points in non-deterministic and deterministic sequences. It can be seen that the sequence points are equally distributed in deterministic sequences compared to the non-deterministic ones. Another important factor signifying proper randomness is the amount of correlation between each $\mathcal{BS}$ or $\mathcal{HV}$ pair (Figs. 1(e),(f)). The level of correlation plays a critical role in SC and HDC operations to produce high quality results. For instance, SC *multiplication* requires independent (or uncorrelated) $\mathcal{BS}$s while SC *min* demands highly correlated input $\mathcal{BS}$s when implemented using AND gates. Nevertheless, pseudo-random sequences exhibit suboptimal performance in cascaded circuit architectures where mid-level correlation among $\mathcal{BS}$s is crucial [7]. Besides that, in HDC, every distinct data symbol requires orthogonal (uncorrelated) $\mathcal{HV}$s to be fed to the encoding stage of the model [8].

Adopting pseudo-randomness for encoding data into $\mathcal{BS}$s or $\mathcal{HV}$s leads to performance degradation of SC and HDC models while it also affects the overall hardware costs. To obtain the desired level of accuracy, the model needs to be run iteratively which in turn leads to increasing the computational overhead, excessive system runtime, wasting the hardware resources, energy inefficiency, and degraded performance. To mitigate these challenges, my research suggests some novel encoding methods based on quasi-randomness to improve the overall performance of SC and HDC systems.

II. Background and Related Work

In SC, any data value is represented by a sequence of random bits ('0's and '1's) [9], [10]. The probability of having '1's in the $\mathcal{BS}$ represents the data value. For instance, data value x with n-bit precision is represented by a $\mathcal{BS}$ denoted as X, with the probability of having '1's over 2^n ($Pr(X = 1)/2^n$). A common method to generate $\mathcal{BS}$s is to compare the given data value with another value coming from an RNG source. While the majority of the SOTA works employ LFSRs (Fig. 1(g)) as the RNG, the authors in [11], [12] proposed the utilization of Sobol sequences as a deterministic approach for SC which significantly improves the model accuracy. Unary computing (UC) [13], [14] is a special class of SC, known for its thermometer encoding. Any scalar value is represented by a sequence of consecutive '1's equal to the scalar value followed by '0's. UC is free from random fluctuations in the '0' and '1' bits, an important source of error in SC.

Similarly, in HDC any atomic data unit ($\mathcal{HV}$) is comprising of '-1's (or '0's) and '1's, represented in high dimensionality. In HDC, any symbolic data including letters, numbers, sensor data, and temporal and spatial information can be represented by distinct and orthogonal $\mathcal{HV}$s. This type of structured information encoding is also a Holographic representation. Orthogonality is provided by randomness to generate independent $\mathcal{HV}$s. The ideal $\mathcal{HV}$ configuration consists of an equal number of '1's and '0's, with each constituting 50% of the vector [15]–[17].

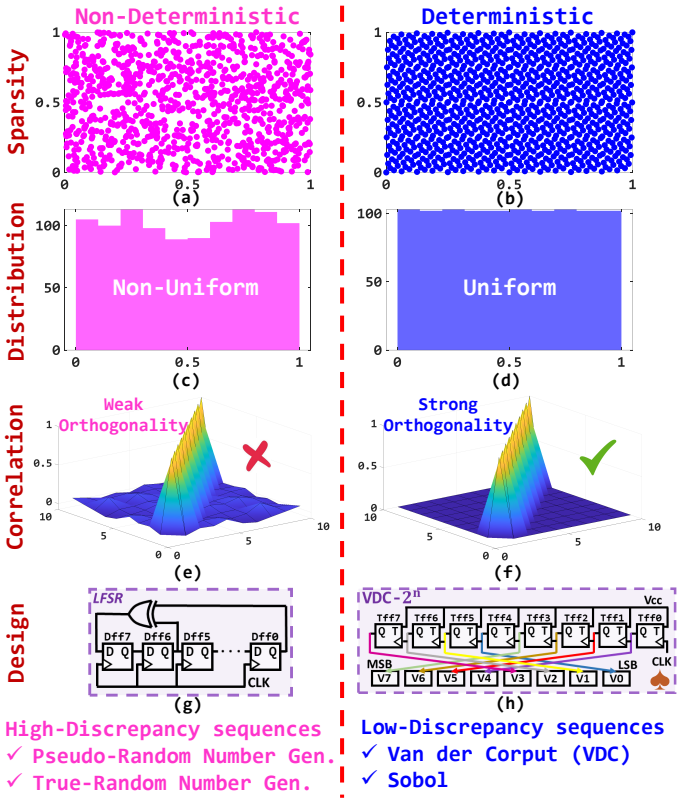

Fig. 1: Determinism vs. Non-Determinism in random sequences. (a),(b) *Sparsity* plots, (c),(d) *Distribution* plots, (e),(f) *Correlation/Orthogonality* plots, and (g), (h) *Design Implementation*.

III. Approach and Uniqueness

Considering the inefficiency of employing pseudo-randomness in SC and HDC, we proposed an efficient, lightweight and highly accurate deterministic bit-stream generator by utilizing Van der Corput (VDC) sequences [7], [18]. In general, any VDC-$\mathcal{B}$ (in base $\mathcal{B}$) sequence number can be generated by reversing the digits in the base $\mathcal{B}$ according to the radix point. The resulting value falls within the [0,1] interval. Our proposal considers *Powers-of-2* bases for the VDC sequences (**VDC−2**^{n}). The advantage of employing **VDC−2**^{n} sequences lies in its simplicity of design, high accuracy, and efficiency. To implement any **VDC−2**^{n} sequence, a single n-bit counter (including T flip flops) is utilized and simple hardwiring scheme is employed to obtain the target sequence value (Fig. 1(h)). A distinguishing feature of our proposed design is its capability to produce multiple distinct sequences simultaneously through various hardwiring schemes. Another significant attribute that sets it apart from SOTA methods is that traditional approaches require multiple executions to achieve the desired accuracy, whereas our design avoids iterative model execution by leveraging deterministic sequences. This feature enhances the overall efficiency and throughput of the system, particularly in scenarios involving resource-constrained devices. The first use case scenario is in the SC.

Fig. 2(a) demonstrates a SC implementation of a non-linear function, i.e., **sin(x)**, considering its conventional design (❶) [1] and our modified design with deterministic RNG source (**VDC−2**^{n}) (❷) [19], [20]. We also show the implementation of the basic SC division operation (❸) [21] and its enhanced design structure (❹) [22], [23] within the SC domain by applying the proposed deterministic method. Employing **VDC−2**^{n} in the design of SC operations [18] and non-linear functions [24] significantly improve the accuracy while at the same time mitigating the overall hardware costs.

Similarly, we showed when the HDC models are equipped with the deterministic sequences, the overall performance improves while the hardware costs are reduced. Fig. 2(b) exhibits the process of applying our deterministic method to the encoding stage of HDC. Although the

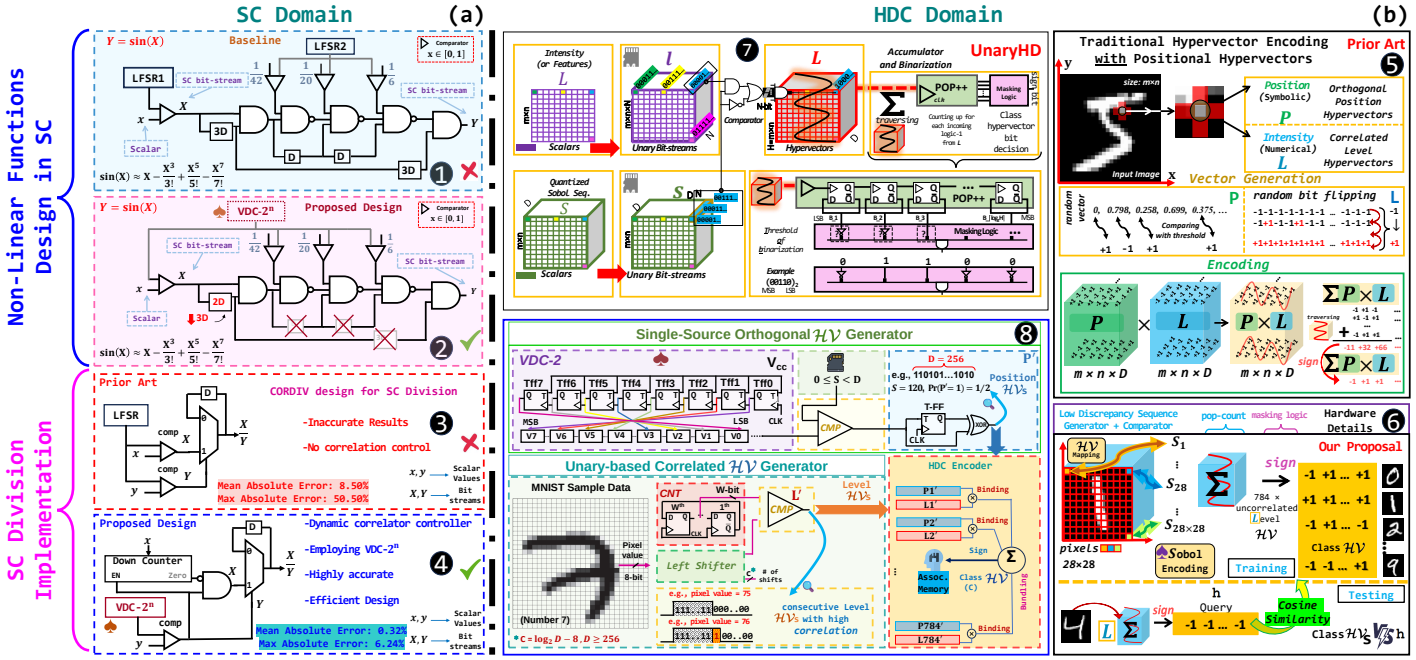

Fig. 2: Applying *Determinism* to the encoding part of SC and HDC models. (a) Implementation of non-linear functions and division operation using the proposed **VDC−2**^{n} sequence generator in the SC domain and (b) Improving HDC performance by exploiting deterministic sequences to generate high quality $\mathcal{HV}$s.

TABLE I: Accuracy Evaluation of implementing **sin(x)** in **SC** (Fig. 2(a)❶,❷).

Function	Design Approach	Polyn. order	N†	Sequence type		Total # of Delay Elements				MSE ($\times 10^{-4}$)
				BSG* Input	BSG Coefficients	@ Input	@ 1st stage	@ 2nd stage	@ 3rd stage	
sin(x)	Proposed ❷	7	1024		VDC128,256,512					**0.523**
			512	VDC4	VDC128,256,512	**2**	**0**	**0**	**0**	0.582
			256		VDC128					0.576
	Baseline ❶	7	1024	LFSR1	LFSR2	3	1	1	3	2.256

*: $\mathcal{BS}$ Generator ‖ †: $\mathcal{BS}$ length

TABLE II: Hardware Cost Comparison of implementing **sin(x)** in **SC** (Fig. 2(a)❶,❷).

Design Approach	N	sin(x)					
		Area (μm^2)	CPL (ns)	Power (μW)	Energy (pJ)	ADP	EDP
Proposed ❷	1024	**554**	0.44	**812.2**	**365.9**	**243.7**	**160.9**
Baseline ❶	1024	801	0.42	2178.2	936.8	336.4	393.4

TABLE III: Accuracy(%) comparison of Design in Fig. 2(b)❻ and Baseline(❺) in **HDC**.

Method		Minimum	Average	Maximum
D=1K	Baseline	70.65	79.09	84.89
	Design of ❻ with *Sobol*+VDC−2^n		**85.10**	
D=2K	Baseline	71.81	81.29	86.96
	Design of ❻ with *Sobol*+VDC−2^n		**87.12**	
D=8K	Baseline	86.19	87.27	87.51
	Design of ❻ with *Sobol*+VDC−2^n		**88.68**	

SOTA HDCs accuracy (MNIST): ① ➞ [29] 75.40% (w/o retraining) D=2K ‖ ② ➞ [30] 86.00% (w/o retraining) D=10K ‖ ③ ➞ [31] 88.00% (w/ retraining) D=10K ‖ ④ ➞ [32], [33] 87.38% (w/ retraining) D=10K.

TABLE IV: Accuracy(%) comparison of Design in Fig. 2(b)❼ and Baseline(❺) in **HDC**.

Datasets	D=1K		D=2K		D=8K	
	UnaryHD	*Baseline*	UnaryHD	*Baseline*	UnaryHD	*Baseline*
CIFAR-10	**39.29**	38.21	**40.28**	40.26	**41.97**	41.71
Blood MNIST	**53.05**	48.52	**55.86**	51.20	**57.88**	51.82
Breast MNIST	**68.59**	68.47	**69.23**	69.11	**71.15**	70.93
Fashion MNIST	**68.60**	54.19	**70.06**	69.97	**71.37**	70.87
SVHN	**60.29**	60.06	**61.73**	61.24	**62.87**	62.82

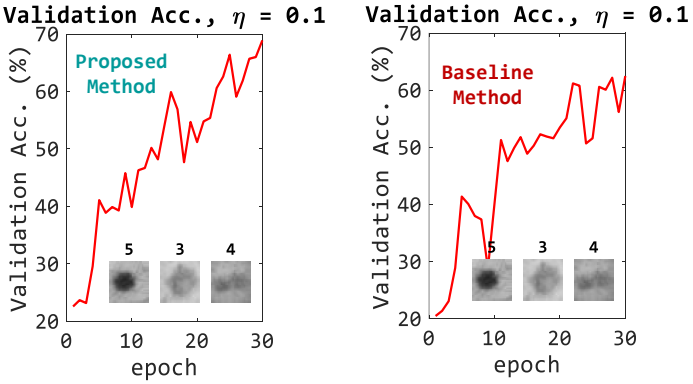

Fig. 3: Performance evaluation of the end-to-end Unary structure for **HDC** on the DermaMNIST dataset using single-source $\mathcal{HV}$ generator design (Fig. 2(b)❽), D=1024 [28].

baseline methods (❺) incorporates both *Position* and *Level* $\mathcal{HV}$s for the encoding parts of the images and binding them together (i.e., element-wise multiplication), applying deterministic sequences to generate $\mathcal{HV}$s eliminates the need for *Position* $\mathcal{HV}$s and further for multiplication operations (❻) [25], [26]. As an extension of this approach, we introduced *UnaryHD* architecture (❼), in which Unary encoding is applied to HDC models through employing quantized LD sequences for $\mathcal{HV}$ generation [27]. Simplifying hardware implementation, providing significant cost savings, and contributing to more efficient encoding of data in HDC systems are among the major benefits of utilizing *UnaryHD*. To further improve the performance of HDC systems we propose an end-to-end Unary structure. This streamlined design is equipped with a lightweight and single-source dynamic $\mathcal{HV}$ generator. The primary objective of this $\mathcal{HV}$ generator design is to achieve optimal randomness in a single iteration, while being entangled with the recurrent nature of the random sequence. Distinguishing itself from baseline HDC (with LFSR), our proposed design does not employ multiple random sequences to generate m different D-sized vectors. Instead, we generate only a single D-sized sequence and employ it to generate different $\mathcal{HV}$s [28]. Another key contribution of this design is a lightweight logic hardware to represent *Level* $\mathcal{HV}$s in the HDC system. For the first time in the literature, *Level* $\mathcal{HV}$s are represented not randomly but deterministically by using our unary generator. We suggest there is no need for randomness in *Level* $\mathcal{HV}$s. Our proposed design for generating unary style *Level* $\mathcal{HV}$s is built with a left shifter module, an up-counter, and a comparator (❽).

IV. Results and Contributions

In order to assess the effectiveness of our proposed deterministic sequences, especially the **VDC−2**^{n}, we first applied them to SC designs. Tables I and II illustrate the evaluation of accuracy and hardware costs for implementing the **sin**(x) function. This can highlight the potential of our approach for designing trigonometric and non-linear functions in SC, which are fundamental components in artificial intelligence and computer vision models [19], [24]. As can be seen, the proposed design significantly improves the accuracy and reduces energy consumption up to 77% and 92%, respectively.

Employing deterministic sequences for generating high quality $\mathcal{HV}$s leads to improving the overall performance of the HDC models. Tables III and IV exhibit the accuracy comparison of employing deterministic sequences for the image classification tasks in the designs of Fig. 2(b) (❻) and (❼), respectively. As it can be seen from the tables, the HDC model encoded with deterministic $\mathcal{HV}$s surpasses the baseline model. Fig. 3 showcases the performance of the end-to-end unary structure for the HDC model of Fig. 2(b) (❽) on the DermaMNIST dataset [34]. Additionally, we incorporated epoch-based training options, given the increased complexity of these datasets compared to conventional handwritten digit classification tasks. The results showcase promising achievement when deterministic sequences are used in HDC encoding. The proposed $\mathcal{HV}$ generator reduces the power consumption by 98% and improves the energy efficiency by 15% compared to the baseline method, which makes it a potential design for dynamic vector generation suitable for resource-constrained devices.

This research undertakes four key endeavors: ① Utilizing deterministic sequences within the SC and HDC systems to generate high-quality bit-streams for the first time in the literature. ② Enhancing the model throughput, efficiency, and accuracy together with reducing hardware costs compared to SOTA. ③ Introducing a novel, streamlined, and efficient RNG for both SC and HDC designs as a promising solution for resource-constrained devices. ④ Integrating UC and HDC for the first time for lightweight and energy-efficient HDC system design.

REFERENCES

[1] K. K. Parhi and Y. Liu, "Computing arithmetic functions using stochastic logic by series expansion," *IEEE TETC*, vol. 7, no. 1, pp. 44–59, 2019.

[2] S.-I. Chu, C.-L. Wu, T. N. Nguyen, and B.-H. Liu, "Polynomial computation using unipolar stochastic logic and correlation technique," *IEEE TC*, vol. 71, no. 6, pp. 1358–1373, 2022.

[3] S. Duan, Y. Liu, S. Ren, and X. Xu, "Lehdc: Learning-based hyperdimensional computing classifier," ser. DAC'22, 2022 , p. 1111–1116.

[4] A. Hernández-Cano, N. Matsumoto, E. Ping, and M. Imani, "Onlinehd: Robust, efficient, and single-pass online learning using hyperdimensional system," in *DATE*, 2021, pp. 56–61.

[5] Y. Kim, J. Kim, and M. Imani, "Cascadehd: Efficient many-class learning framework using hyperdimensional computing," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE Press, 2021, p. 775–780.

[6] A. G. Ayar, S. Aygun, M. H. Najafi, and M. Margala, "Word2hypervec: From word embeddings to hypervectors for hyperdimensional computing," in *The 34th Great Lakes Symposium on VLSI (GLSVLSI'24)*, 2024.

[7] **Mehran Shoushtari Moghadam**, S. Aygun, M. Riahi Alam, J. I. Schmidt, M. H. Najafi, and N. Taherinejad, "Accurate and energy-efficient stochastic computing with Van Der Corput sequences," in *Proceedings of the 18th ACM International Symposium on Nanoscale Architectures*, ser. NANOARCH '23. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3611315.3633265

[8] S. Aygun, **Mehran Shoushtari Moghadam**, M. H. Najafi, and M. Imani, "Learning from hypervectors: A survey on hypervector encoding," 2023, under review.

[9] S. Liu, J. L. Rosselló, S. Liu, X. Tang, J. Font-Rosselló, C. F. Frasser, W. Qian, J. Han, P. Reviriego, and F. Lombardi, "From multipliers to integrators: A survey of stochastic computing primitives," *IEEE Transactions on Nanotechnology*, vol. 23, pp. 238–249, 2024.

[10] S. Aygun, M. H. Najafi, M. Imani, and E. O. Gunes, "Agile simulation of stochastic computing image processing with contingency tables," *IEEE TCAD*, pp. 1–1, 2023.

[11] S. Liu and J. Han, "Toward energy-efficient stochastic circuits using parallel sobol sequences," *IEEE TVLSI*, vol. 26, no. 7, 2018.

[12] M. H. Najafi, D. Jenson, D. J. Lilja, and M. D. Riedel, "Performing stochastic computation deterministically," *IEEE TVLSI*, vol. 27, no. 12, pp. 2925–2938, 2019.

[13] M. H. Najafi, D. J. Lilja, M. D. Riedel, and K. Bazargan, "Low-cost sorting network circuits using unary processing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 8, pp. 1471–1480, 2018.

[14] D. Wu, J. Li, R. Yin, H. Hsiao, Y. Kim, and J. S. Miguel, "Ugemm: Unary computing architecture for gemm applications," in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, 2020, pp. 377–390.

[15] D. Kleyko, D. A. Rachkovskij, E. Osipov, and A. Rahimi, "A survey on hyperdimensional computing aka vector symbolic architectures, part i: Models and data transformations," *ACM Comput. Surv.*, vol. 55, no. 6, dec 2022. [Online]. Available: https://doi.org/10.1145/3538531

[16] L. Ge and K. K. Parhi, "Classification using hyperdimensional computing: A review," *IEEE Circ. & Sys. Mag.*, vol. 20, no. 2, pp. 30–47, 2020.

[17] S. Aygun, M. H. Najafi, and M. Imani, "A linear-time, optimization-free, and edge device-compatible hypervector encoding," in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2023, pp. 1–2.

[18] **Mehran Shoushtari Moghadam**, S. Aygun, M. R. Alam, and M. H. Najafi, "P2LSG: Powers-of-2 low-discrepancy sequence generator for stochastic computing," in *2024 29th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2024, pp. 38–45.

[19] S. Aygun, **Mehran Shoushtari Moghadam**, and M. H. Najafi, "TriSC: Low-cost design of trigonometric functions with quasi stochastic computing," in *The 61th Design Automation Conference (DAC'24)*, 2024.

[20] S. Aygun, **Mehran, Shoushtari Moghadam**, and M. H. Najafi, "Ultra low-cost trigonometric functions design in stochastic computing," in *The 2nd Workshop on Unary Computing (WUC'24)*, 2024.

[21] T.-H. Chen and J. P. Hayes, "Design of division circuits for stochastic computing," in *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2016, pp. 116–121.

[22] **Mehran Shoushtari Moghadam**, S. Aygun, S. Asadi, and M. H. Najafi, "Low-cost and highly-efficient bit-stream generator for stochastic computing division," *IEEE Transactions on Nanotechnology*, vol. 23, pp. 195–202, 2024.

[23] M. H. Najafi, **Mehran, Shoushtari Moghadam**, , S. Aygun, and S. Asadi, "EFFICIENT BIT-STREAM GENERATOR FOR ACCURATE STOCHASTIC COMPUTING DIVISION," 2024, **US Patent App. 63/659,553**.

[24] **Mehran Shoushtari Moghadam**, S. Aygun, and M. H. Najafi, "QR-SC: Lightweight yet accurate implementations of non-linear functions with quasi-random stochastic computing," 2024, under review.

[25] **Mehran, Shoushtari Moghadam**, S. Aygun, and M. H. Najafi, "No-multiplication deterministic hyperdimensional encoding for resource-constrained devices," *IEEE Embedded Systems Letters*, vol. 15, no. 4, pp. 210–213, 2023.

[26] S. Aygun, **Mehran, Shoushtari Moghadam**, and M. H. Najafi, "METHOD FOR IMPROVED HYPERVECTOR ENCODING," 2023, **US Patent App. 63/530,034**.

[27] S. Aygun, **Mehran Shoushtari Moghadam**, and M. H. Najafi, "uHD: Unary processing for lightweight and dynamic hyperdimensional computing," *Design, Automation and Test in Europe (DATE'24)*, 2024.

[28] **Mehran Shoushtari Moghadam**, S. Aygun, F. Banitaba, and M. H. Najafi, "All you need is unary: End-to-end unary bit-stream processing in hyperdimensional computing," *International Symposium on Low Power Electronics and Design (ISLPED'24)*, 2024.

[29] S. Datta, R. A. G. Antonio, A. R. S. Ison, and J. M. Rabaey, "A programmable hyper-dimensional processor architecture for human-centric iot," *IEEE JETCAS*, vol. 9, no. 3, pp. 439–452, 2019.

[30] E. Hassan, Y. Halawani, B. Mohammad, and H. Saleh, "Hyper-dimensional computing challenges and opportunities for ai applications," *IEEE Access*, vol. 10, pp. 97 651–97 664, 2022.

[31] C.-Y. Hsieh, Y.-C. Chuang, and A.-Y. A. Wu, "Fl-hdc: Hyperdimensional computing design for the application of federated learning," in *AICAS*, 2021, pp. 1–5.

[32] M. Imani, S. Bosch, S. Datta, S. Ramakrishna, S. Salamat, J. M. Rabaey, and T. Rosing, "Quanthd: A quantization framework for hyperdimensional computing," *IEEE TCAD*, vol. 39, no. 10, pp. 2268–2278, 2020 .

[33] S. Duan, X. Xu, and S. Ren, "A brain-inspired low-dimensional computing classifier for inference on tiny devices," *arXiv preprint arXiv:2203.04894*, 2022.

[34] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni, "Medmnist v2 - a large-scale lightweight benchmark for 2d and 3d biomedical image classification," *Scientific Data*, vol. 10, no. 1, p. 41, Jan 2023. [Online]. Available: https://doi.org/10.1038/s41597-022-01721-8