A LOW COMPLEXITY AND CLOCK GATED HYBRID STOCHASTIC NUMBER FOR MACHINE LEARNING ACCELERATION

Mr.Maria Antony S,Assitant Professor (SI.G)
Department of ECE,KIT-Kalaignarkarunanidhi
Institute of Technology,Coimbatore, antony4316@gmail.com.
Ms.*Nishmitha K R,*
*Department of ECE,*KIT-Kalaignarkarunanidhi
Institute of Technology,Coimbatore,nishmithakarunanithi2001@gmail.com.

Abstract— ***A type of number-crunching computer based on stochastic numbers (bit-stream), stochastic computing (SC) is intriguing in step of twofold numbers (BNs).CMOS entryway circuits use stochastic numbers (SN) to represent and transport data in the form of pseudo-analogue probabilities .Super moo is the main factor influencing the stochastic number framework's rejuvenated win manage use and a strong, consistent quality for edge computing. Actually, the stochastic number is a non-positional representation of a number that is inherently successive and is used for some essential number-crunching operations (including increment and addition/subtraction), and compares to a super moo zone circuit. Cross breed stochastic number (HSN), a revolutionary crossover number framework comprising BNs and stochastic number representation, is proposed in this paper. This study demonstrates the characteristics of crossover stochastic computing (HSC) and provides the basic theoretical viewpoints of the HSN.Equipment employed in profound neural organisation using HSC is made using a traditional 40-nm moo control CMOS handle. It features 4544 increase amassing operations (MACs), a clock of 400 MHz, a control of 102.3 mW, and a central range of 0.53 mm2.***

Index Terms — ***Deep neural networks, hybrid stochastic computing (HSC), hybrid stochastic number (HSN), application-specific integrated circuits (ASIC), and stochastic numbers.***

I. INTRODUCTION

To speed up number juggling activities, expansion using carry-free engendering chains is feasible thanks to the well-known excess number representations. Currently, the program's strict requirements, such as moo control, tall execution, and delightfully moo taken toll, force equipment computing. In the 1960s, stochastic computing (SC) was put out as an inexpensive, low-power alternative to traditional twofold number (BN) computing. Simple entrance circuits may be used to efficiently generate stochastics with exceptional qualities, replacing intricate circuits of BNs. SC is a computational problem based on the likelihood and implications of the bit stream.

For complex calculations, BNs' speed advantage is crucial. In any case, its number juggling structure is defined by the need for a wide communication width between measured processors to actualise high-performance computing systems.The calculation is carried out on a bitstream instep of a BN from the standpoint of SC. Depending on the duration of the collection's stream, the probability of ending with "1" speaks to the stochastic bitstream. All of the computational control required by the SC was dispersed throughout the inexpensive door circuit. Stated differently, traditional SC with high blame resistance saves a lot of equipment and control assets at the expense of efficiency.Despite this advantage and its inalienable properties, SC cannot supplant BN because of their usual lack of accuracy and high computational idleness.In routine SC, the tall idleness is primarily driven by the 1-bit stochastic stream's low information-carrying capacity.When the stochastic rationale's bitstream length is break even with to L, it can as it were speak to L diverse information points. It is obvious that a bitstream's data carrying capability is inferior to a BN's. A stochastic bitstream instep of a BN is used for the computerised computation of SC. To handle negative numbers, several SC frameworks employ the bipolar organisation, where the SC extends viably to [−1, 1].Gaines demonstrated dual-rail unipolar and bipolar number designs and the fundamental circuits for each configuration.A few articles have addressed forms, which are quite helpful when using equipment to prepare complicated capacities.proposed an approach to compute univariate mathematical workloads using guessed crossover binary-unary computation.A few analysts have attempted to overcome latency with various developments in order to increase the effectiveness of SC encoding. Some analysts have progressed the computational exactness of SC with distinctive advances, such as deterministic and so on . Xia et al. proposed the neural synaptic plasticity-inspired computing and reconfigurable neural synaptic versatility for executing tall computing proficiency in convolutional neural organize (CNN) quickening agent. SC has generally aimed to focus on a wide range of specialised applications, including photo preparation and neural systems polynomial computations.But according to Poppelbaum, "short arrangements are untrustworthy," and a significant drawback of SC is its moo transmission speed, which in turn affects its moo computational speed.Some crucial potential disclosures are included related to the crucial concept of stochastic number (SN) that have pulled in small consideration, in spite of the fact that they have positive suggestions for SC. Common methods for analysing stochastic combinational rationale circuits in half breed, unipolar, and bipolar encoding schemes have been demonstrated by Parhi. A simpler combinational logic structure for the union of any polynomial may be realised by strategies that make use of bipolar and cross breed groups. A contemporary encoding technique that uses a fragmented frame and extends the run of bitstream representations over the whole real-number axis was proposed by Canals et al.We suggest the HSN and crossover stochastic computing (HSC) technique and discuss the characteristics of the HSN representation in order to understand the fundamental problem and main plan problems of conventional SC. The following are this ponder's main obligations.

1) First, the number framework of BN or SC proposes the half breed stochastic number (HSN) representation approach from the BN and stochastic number.
2) Join the representations of BN, SN, and HSN. There is discussion of the changing connection and numerical representation of the BN, SN, and HSN.
3) HSN is more productive and inactive than conventional SN since there is no converter between the BN and SN.

II. NUMBER SYSTEM

A. The traditional binary number

As science has advanced, so too have methods for representing numbers. The most common fixed-radix representation technique is the positional number framework, in which the unit corresponding to each position is a constant deviation of the unit for its right neighbouring position. Additionally, the BN is a positional number framework that is nonredundant, with a digit set of [0, 1]. It is widely used in central processing units (CPU), graphics processing units (GPU), and neural processing units (NPU) and is easy to implement in CMOS logic circuits. Avizienis[1] described a class

of marked digit (excess) number frameworks in the early 1960s. These frameworks have symmetric digit sets [−α, α] with radix r > 2, where α is a subjective integer. Therefore, all of the repetitive number representations used in hone were thought to be bound together by the repetitive signed-digit number frameworks with common, sometimes deviating, digit sets of the frame [−α, β].

Parhami [2], Gaines [8], and Alaghi et al. [9] described expanded signed-digit number representation frameworks. New HSN frameworks are extended as shown in Fig. 1.

B. The Standard Stochastic Number

A traditional SN represents data with a single bitstream, and the stochastic bitstream's expectation is either −1 to 1 (bipolar) or 0 to 1 (unipolar). Here, unipolar SN is our primary focus. One bit of the encoding output is produced as illustrated in Fig. 2 after the random number generator (RNG) in the sequence is compared with the BN. The comparator produces a high-level "1" output if BN is greater than a random integer and a low-level "0" output otherwise. The likelihood that the BN is greater than a random number is equivalent to the output bitstream's expectation. The likelihood of an SN of bitstream length when it is in a unipolar form,

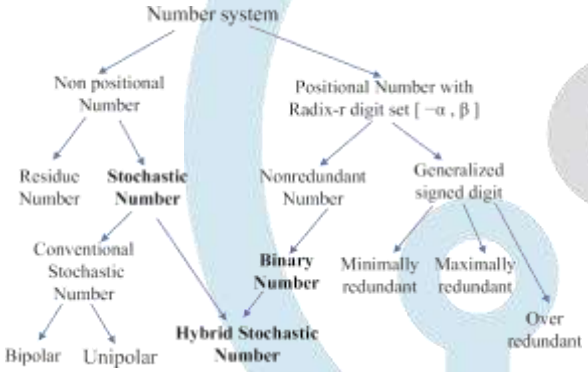


Fig. 1. HSN, SN, and the traditional BN scheme.

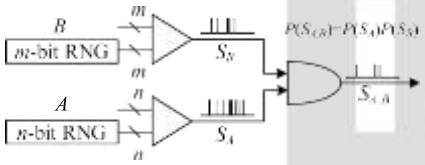


Fig. 2. SC with unipolar stochastic bitstreams.

The relative frequency of encoding 1 is denoted by L,

$$\hat{P} = \hat{P}(SN = 1) = \frac{\sum_{j=0}^{L-1} SN(j)}{L} \in [0,1] \qquad (1)$$

Where $\sum_{j=0}^{L-1} SN(j)$ is the number of 1's in the sequence compared, and $\hat{P}$ is the estimated value of the probability. The SN representation of a particular stream is not defined by the individual The SC is a non-positional number representation since it represents the value of a bit in a particular location. A redundant number is used by SN. system, where each number has a redundant representation, $\hat{P}(SN = 1)$.One algebraic variable in duplicated representation Value can be expressed in a variety of ways. For instance, 1/3 can be represented in three different ways with bitstream length L = 3: (0, 0, 1), (0, 1, 0), and (1, 0, 0).

Fig. 3. A variety of (a) BN, (b) SN, and (c) HSN representations.

Basic rationale entryway circuits can be used to do SC number juggling, which is caused by the stochastic number associated with mapping from space to time encoding. SC advances the strength of circuits vulnerable to arbitrary defects or component inconsistencies as another non-positional representation.

C. The Combination of Binary and Stochastic Numbers

As shown in Fig. 3, the representations of SN and BN are two extremes of HSN and two rare types of HSN. A parallel multi-bit stream communicates with the HSN, which then receives the position number from the BNs. In essence, advances have been made in the numerical representation precision or the sum of data conveyed by a multi-bit stream HSN. Additionally, the problem of tall inactivity in the traditional SN representation is well addressed by the fact that the bit stream length (L) of HSN is completely shortened for the same precision number representation.HSN frameworks are repeating number representations as opposed to regular BN frameworks. Compared to an SN, encoding the same number with an HSN may result in distinct representations. For instance, since the harsh values of the three HSNs all amount to 5/4, (2, 0, 1, 2)HSN, (2, 1, 1, 1)HSN, and (1, 3, 1, 0)HSN are all valued equally. HSN frameworks are hence agents of frameworks that include repeating numbers. The HSN yield has a coefficient of 2−kmax and speaks to x effectively. To elucidate the intricate aspects of the encoding circuit, Fig. 3.4(a) provides an example of the period of a 3-bit HSN.

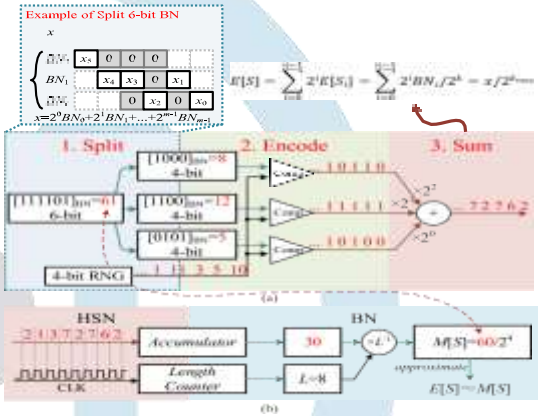


Fig. 4. Encoding and decoding processes of the HSN. (a) 6-bit BN is encoded as 3-bit HSN. (b) HSN is decoded as BN.

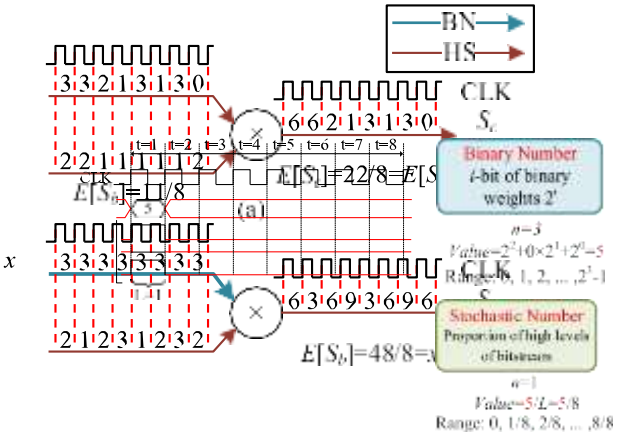
As previously said, the HSN's want determines the respect it receives, and this desire is determined by adding up the cruel of an HSN with infinite length, which is unfathomable in real circuits. Then, as shown in Fig. 4(b), the cruel of an HSN with a restricted length stream is numbered to obtain the imprecise desire.

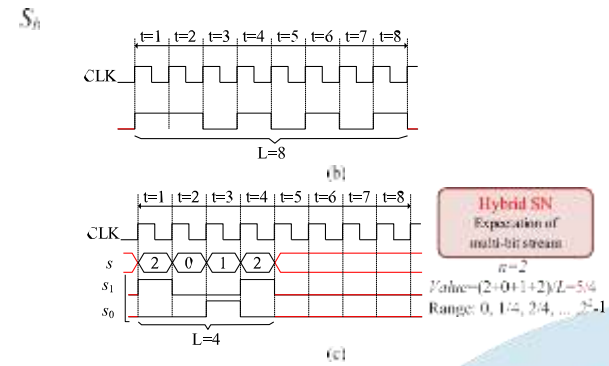
D. HSC arithmetic operations

Duplication: As seen in Sa and Sb talk to the two HSNs and the bit stream lengths were both L, the most frequent situation is the duplication of two HSNs. In the HSC, the Sc desire gives birth to the Sa and Sb desires. The link between the duplicating operation's yields and input demands is completed. When Sa and Sb are independent, a duplication operation occurs.

$$E[S_c] = E[S_a \cdot S_b] = E[S_a] \cdot E[S_b] \text{ iff } S_a, S_b \text{ indepedent.}$$

There are two main reasons why this kind of growth might result in errors. First of all, achieving a completely independent relationship between the inputs is challenging. At this point, the HSN's short-lived cruelty is used to express its desire, which might lead to errors. Figure 5 shows an example of this technique.





CLK

Fig. 5. Techniques for multiplying HSC using various input types.Both (a) HSN × HSN and (b) BN × SN have adjustable bit widths.

III. NEURAL NETWORK IMPLEMENTATION OF HSC

One popular actuation work for neural networks is Relu(). It is carried out by applying the progressive guess technique. The positive-and-negative number judgement circuits in Fig. 6 can replace the LUT shown in, reducing the amount of equipment used.

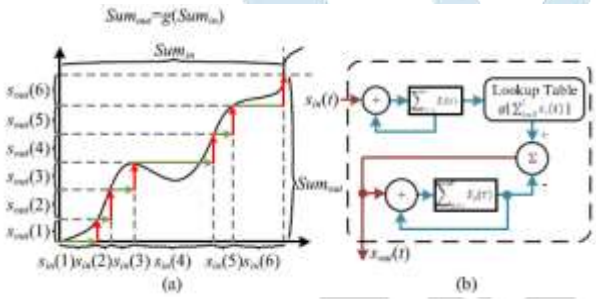


Fig. 6. Utilising the successive approximation approach, the function was calculated. The sequential approximation method is illustrated in (a). (b) Repeated approximation circuit

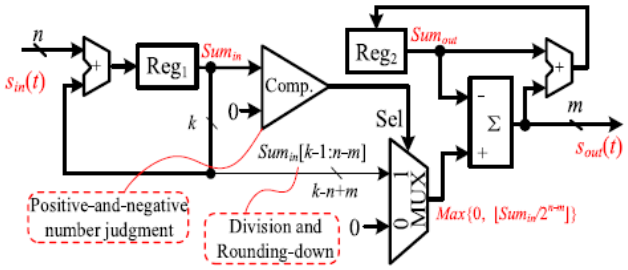


Fig. 7. The successive approximation approach is used to realise the circuit for the Relu() function.

Without LUT, Defective Relu() can also be implemented. The difference is that an is a consistent coefficient for Defective Relu() to untangle the circuit, and an is mandated to be 2−k, k ∈ Z. An increase may be achieved by using a move operation. The input of port0 of MUX is a Pt τ=1 Sin (τ) instep of 0.To reduce the bit width in expansion, the HSN yield might be shortened. In Fig. 8, the input bit width of HSC is n-bit, using Relu() as an example. The following criteria replace the enactment capacity of (20) and (21) in order to obtain the m-bit yield HSN. where the rounding-down procedure is indicated by ⌊ ⌋. Typically, the truncation should occur after the Relu() action. In any case, it occurs before the Relu() procedure in Figure 7. This change reduces the MUX's input bit width without affecting the computation results, which is a rare strategy for Relu() or Cracked Relu().The HSN-based yield may also be truncated by other complicated capabilities when they are calculated. This conspiracy may change the yield HSN's bit width in a flexible way, which helps to reduce the equipment overhead of the other cascade circuit.

A. Circuit of Neurones

To avoid the impact of freedom in a traditional SC, information must be switched between the SN and BN many times. As a result, the circuit has difficulty implementing a pipeline engineering, and a significant amount of handling time is needed for information changes, resulting in high levels of inactivity. Therefore, based on computing productivity, which is operations per joule (or per range) per moment, the vitality (or region) productivity of SC is often worse than that of traditional parallel computing because of the enormous disadvantage in data effectiveness and encoding costs. However, SC's low-cost focus areas haven't been entirely misused. The suggested HSC sheds light on this problem by eliminating the requirement to switch information between different groups, which saves time and equipment costs associated with transformation. The HSC approach can demonstrate its benefits in highly parallel and intricate computing scenarios, like neural networks. As shown in Fig. 7, the neurone circuit was delineated using HSC. The idea uses a double snake as an HSC viper. Double viper can achieve greater processing accuracy and reduced idleness, while having a larger range than MUX viper. S1(t), S2(t),..., Si (t) are n-bit HSs that communicate with the neuron's inputs; w1, w2,..., wj, and inclination are the neuronal parameters that are stored as BN. As shown in Fig. 7, the neurone uses Relu() as an actuation work. To ensure that the input and yield HSNs have the same bit width, Relu() also truncates. The cascade association of several neurones was realised using the identical bit lengths of the input and produce HSs. This makes it possible to customise a neural organization's pipeline engineering to achieve low-latency execution. In expansion, the information transformation issue of ordinary SC, which is the major change between SN and BN, was overly tended to[11],[10],[9].

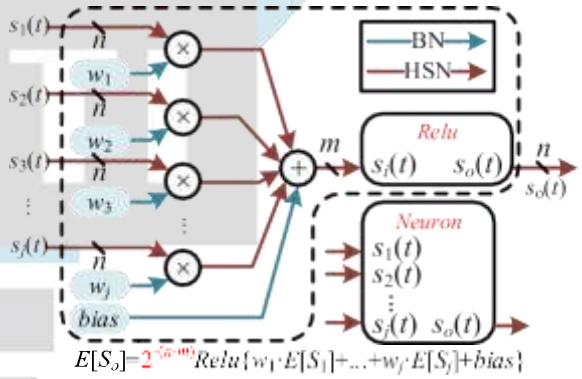


Fig. 9. HSC-based neuronal structure without any weight and bias data converters. HSN is used for both input and output, and the output's bit width is modifiable.

TABLE I

NEURONE CIRCUIT HARDWARE RESULT WITH VARIOUS NUMBER OF SYNAPSES

Number o Synapses	Method		LUT	FF	Power (W)	Power pe Synapse (mW)	L.
32	HSC	1-bit	388	344	0.02	0.63	256
		2-bit	631	453	0.031	0.97	128
	B. (8-bit)		3142	1065	0.105	3.28	1
64	HSC	1-bit	736	667	0.036	0.56	256
		2-bit	951	786	0.046	0.72	128
	B. (8-bit)		6249	2151	0.21	3.28	1

All information were gotten from the companion plan suite of FGPA,KCU116 .Clock recurrence is 3000 MHz. Bit width of input HSC is 1-Bit or 2-bit.The weight and predisposition are 8-bit BN. The circuits of Twofold are consequently optimized from the plan suite .B. speaks to Parallel, L. speaks to idleness.

Table I shows how the neurone circuit is built for a variety of neural connections based on HSC and conventional parallel, with the HSC circuit being implemented in accordance with Fig. 8. Wj, inclination, and Si (t) are all set to 8-bit designated BN and 1-bit HSN, respectively. The Relu() circuit's fetch finds the median value of each neural connection as the number of neural connections increases. [18] Thus, when 64 inputs are used, the typical control utilisation is a little lower than when 32 inputs are used.Neurones realised by HSC (1-bit HSN) required less than 20% of the equipment compared to neurones realised in parallel. By increasing the bit width, the benefit of the equipment used for HSC may be further enhanced. Without repeatedly reading data from the external memory, the set parameters are simply stored at the end of the computer. Therefore, it avoids the problem that the speed of planning information is required to drive the speed of the artificial intelligence (AI) quickening agent in simultaneously.

TABLE II

DEEP NEURAL NETWORK IMPLEMENTATION USING FPGA-BASED HARDWARE

Bit Width of Quantization	LUT	FF	Power (W)
8-bit	53110	47503	2.389

Number of synapses	Clock Frequency (MHz)	computation Performance (TOPS)	Energy Efficiency (TOPS/W)
4544	300	2.726/L	1.141/L

All data where obtained from the companion design suite of FGPA

B. Neural Network Use Case

A five-layer fully connected neural organisation is created to categorise two straight-line inseparability moons based on the neurone circuit shown in Figure 8. The profound organise had the following structure: [2, 64, 32, 32, 32, 2]. The results of the categorisation are shown in Fig. 9. MATLAB and PyTorch were used to undertake extensive arrangement preparation. Eight-bit quantization was applied to the weight and the predilection of the organisation, including the inputs x and y.[15]The crimson dashed line in Figure 9 represents the border of the twofold categorisation. Strong lines of various colours represent the HSC-based categorisation borders with characteristic stream lengths. The difference between the HSC and the double boundary is still very small when L $\geqslant$ 26. In fact, the HSC's boundary between the two sets of foci might be easily seen if L dropped to 25.

A circuit of equipment was used to validate this neuronal organisation using HSC. The main feature of the HSC organisation is that, as differences develop, the input information is not independent, which goes against the freedom requirement of traditional SC. There were 4544 neural connections in the test, which included 4544 duplicates-aggregation (MAC) connections. Table II shows the results of the usage. The multiplier and viper typically have equipment measurements of 11.7 LUTs and 10.5 FFs, respectively. With a control of 2.389 W, it can execute 2.726 × 1012 HSC operations per minute (OPS) at a frequency of 300 MHz, which is equivalent to 2.726/L × 1012 parallel OPS. The bitstream length, L, is determined by the level of accuracy required. The 40-nm CMOS invention was used to implement and produce the full interface (FC) neural arrange over based on HSC. For MNIST classification, the deep neural arrange's structure was adjusted to [64, 32, 32, 32, 10]. Arrange preparation was done with PyTorch. The misclassification rate increased to 3.89% using the coasting point number in the computer programme because the photos were downsized from 28 × 28 to 8 × 8 in order to reduce the number of MACs. With a misclassification rate of 3.96%, MATLAB quantized the weights and inclinations of the arrange's input data to 8-bit INT using an extend straight symmetric quantization technique.

Method		SC				
		Conventional			Pipeline	Analog
		[30]	[31]	[32]	[33]	[34]
Process		ASIC 45nm	ASIC 45nm	ASIC 45nm	ASIC 45nm	ASIC 65nm
Number of MAC		25	256	512	550	-
power		1.28	1490	2619.8	651	33.17
clock		1136	561	1563	200	-
Area	mm² for ASIC)	0.00085	0.0012	1.09	2.01	1.321
	LUT for FPGA	-	-	-	-	-
Length of bit		2048	1024	1024	512	8
latency		1.8	1.83	0.66	2.56	-
Network Model		Lenet5 Conv×3 FC×3				
Data Set		MNIST		Cifar10		MNIST
Top-1 accuracy (%)		99.12	99.07	88	91	99.06
Operation per second		0.028	0.28	1.563	220	-
Energy efficiency		22	0.18	0.596	340	1039
Length x Area per MAC		0.07	0.0048	2.18	1.87	-
Length x Power per MAC		0.092	10.62	3.35	3.03	-

	BN				Proposed HSC (8-BIT)		
	16-bit		8-bit				
	[35]	[36]	[37]	[38]			
	FPGA VC709	FPGA GX1150	FPGA VX690T	ASIC 16nm	FPGA KCU116		ASIC 40nm
	2144	1320	1027	1024	10332	6804	4544
	30200	37448	9180	4160	5852	3985	102.3
	156	370	200	2001	300		400
	-	-	-	3.1	-	-	0.53
	273805	437000	231761	-	115143	78401	-
	-	-	-	-	32		
	-	-	-	-	0.11		0.08
	AlexNet Conv×5 FC×3	-	VGG16 Conv×13 FC×3	-	Conv×8	Conv×4	FC×4
	ImageNet	-	ImageNet	-	Cifar10	MNIST	MNIST
	62.5	-	63.74	-	86.99	99.1	96.01
	565.94	1790	760.83	4098	193.7	127.56	113.6
	22.15	47.8	82.88	985.1	33.10	32.01	1109.8
	-	-	-	-	-	-	0.0037
	-	-	-	-	0.0604	0.0625	0.0018

IV. CONCLUSION

We presented the HSN and HSC approach, a contemporary number representation, in this paper. In order to bring the number representation together and demonstrate the fundamental characteristics of the HSC, this thought first proposed the number representation from BN and SC to HSN. Depending on the weight of the position number framework, the multibit stream stochastic number speaks to the HSN. Two extremes and two types of unusual situations of HSN are represented by 1-bit stream SN and BNs. The suggested multibit stream desire approach may complete the intricate HSC calculation. HSC achieved high vitality proficiency, reduced idleness, and completely accurate computation in comparison to the standard SC. A few basic number juggling circuits for the HSC framework are also described, along with an illustration of the validity of the computation rules for multibit streams.This

article described the deep brain arrangement, neuro, and actuation work circuits with HSC. For a handwritten number recognition study of a 40-nm ASIC, a five-layer neural architecture with an HSC was implemented. The HSCCompared to the execution of analogue or parallel plans, the organisation achieved more than 50× the vitality proficiency of standard SC plans. According to these applications, the HSC computing approach outperforms the standard SC in terms of execution, avoiding the latter's shortcomings in terms of change and idleness. [20], [29], [30], and 31. In this sense, it demonstrates the core ideas of HSN and HSC theory in relation to edge computing frameworks and shows promising application opportunities in highly parallel computing.

REFERENCES

[1] A. Avizienis, "Signed-digit numbe representations for fast parallel arith- metic," *IEEE Trans. Electron. Comput.*, vol. EC-10, no. 3, pp. 389–400, Sep. 1961.

[2] B. Parhami, "Generalized signed-digit number systems: A unifying framework for redundant number representations," *IEEE Trans. Comput.*, vol. 39, no. 1, pp. 89–98, Jan. 1990.

[3] B. R. Gaines, "Stochastic computing," in *Proc. Spring Joint Com- put. Conf. AFIPS (Spring)*, 1967, pp. 149–156, doi: 10.1145/1465482. 1465505.

[4] B. D. Brown and H. C. Card, "Stochastic neural computation. I. Com- putational elements," *IEEE Trans. Comput.*, vol. 50, no. 9, pp. 891–905, Sep. 2001.

[5] A. Alaghi and J. P. Hayes, "Survey of stochastic computing," *ACM Trans. Embedded Comput. Syst.*, vol. 12, no. 2s, pp. 1–19, May 2013.

[6] J. P. Hayes, "Introduction to stochastic computing and its challenges," in *Proc. DAC*, no. 59, 2015, pp. 1–3.

[7] B. R. Gaines, "Stochastic computing systems," in *Advances in Infor- mation Systems Science*. Boston, MA, USA: Springer-Verlag, 1969, pp. 37–172.

[8] A. Alaghi, W. Qian, and J. P. Hayes, "The promise and challenge of stochastic computing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 8, pp. 1515–1531, Aug. 2018.

[9] S. R. Faraji and K. Bazargan, "Hybrid binary-unary hardware accelera- tor," *IEEE Trans. Comput.*, vol. 69, no. 9, pp. 1308–1319, Sep. 2020.

[10] S. Liu and J. Han, "Energy efficient stochastic computing with sobol sequences," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2017, pp. 650–653.

[11] S. Aygun, L. Kouhalvandi, M. H. Najafi, S. Ozoguz, and E. O. Gunes, "Hardware-software co-optimization of long-latency stochastic comput- ing," *IEEE Embedded Syst. Lett.*, early access, Sep. 25, 2023, doi: 10.1109/LES.2023.3298734.

[12] X. Tang et al., "Delta sigma modulator-based dividers for accurate and low latency stochastic computing systems," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 13, no. 1, pp. 270–284, Mar. 2023.

[13] M. H. Najafi, D. Jenson, D. J. Lilja, and M. D. Riedel, "Performing stochastic computation deterministically," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 12, pp. 2925–2938, Dec. 2019.

[14] J. Wang, H. Chen, D. Wang, K. Mei, S. Zhang, and X. Fan, "A noise-driven heterogeneous stochastic computing multiplier for heuristic precision improvement in energy-efficient DNNs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 2, pp. 630–643, Feb. 2023.

[15] Z. Xia, J. Chen, Q. Huang, J. Luo, and J. Hu, "Neural synaptic plasticity- inspired computing: A high computing efficient deep convolutional neural network accelerator," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 2, pp. 728–740, Feb. 2021.

[16] H. Chen and J. Han, "Stochastic computational models for accurate reliability evaluation of logic circuits," in *Proc. 20th Symp. Great lakes Symp. (VLSI)*, May 2010, pp. 61–66.

[17] H. Aliee and H. R. Zarandi, "Fault tree analysis using stochastic logic: A reliable and high speed computing," in *Proc. Annu. Rel. Maintainability Symp.*, Jan. 2011, pp. 1–6.

[18] H. Sim, D. Nguyen, J. Lee, and K. Choi, "Scalable stochastic-computing accelerator for convolutional neural networks," in *Proc. 22nd Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2017, pp. 696–701.

[19] N. Temenos and P. P. Sotiriadis, "A stochastic computing sigma-delta adder architecture for efficient neural network design," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 13, no. 1, pp. 285–294, Mar. 2023.

[20] S. Khoram, K. Daruwalla, and M. Lipasti, "Energy-efficient Bayesian inference using bitstream computing," *IEEE Comput. Archit. Lett.*, vol. 22, no. 1, pp. 37–40, Jan. 2023.

[21] Hongge Li (Member, IEEE) received the Ph.D.degree in engineering from the Graduate School of Information Sciences, Tohoku University, Sendai,Japan, in 2005.

[22] Y. Chen and H. Li, "Stochastic computing using amplitude and frequency encoding," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 30, no. 5, pp. 656–660, May 2022.

[23] H. Li and Y. Chen, "Hybrid logic computing of binary and stochastic,"IEEE Embedded Syst. Lett., vol. 14, no. 4, pp. 171–174, Dec. 2022.

[24] S. R. Faraji, M. H. Najafi, B. Li, D. J. Lilja, and K. Bazargan, "Energyefficient convolutional neural networks with deterministic bit-stream processing," in Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE),Mar. 2019, pp. 1757–1762.

[25] Z. Li et al., "HEIF: Highly efficient stochastic computing-based inference framework for deep neural networks," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 38, no. 8, pp. 1543–1556, Aug. 2019.

[26] A. Zhakatayev, S. Lee, H. Sim, and J. Lee, "Sign-magnitude SC: Getting10X accuracy for free in stochastic computing for deep neural networks,"in Proc. 55th ACM/ESDA/IEEE Design Autom. Conf. (DAC), Jun. 2018, pp. 1–6, doi: 10.1109/DAC.2018.8465807.

[27] C. F. Frasser et al., "Fully parallel stochastic computing hardware implementation of convolutional neural networks for edge computing applications," IEEE Trans. Neural Network Learn. Syst., early access,Apr. 22, 2022, doi: 10.1109/TNNLS.2022.3166799.

[28] V. T. Lee, A. Alaghi, J. P. Hayes, V. Sathe, and L. Ceze, "Energy-efficient hybrid stochastic-binary neural networks for near-sensor computing,"in Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE), Mar. 2017 pp. 13–18, doi: 10.23919/DATE.2017.7926951.

[29] H. X. Fan, L. Jiao, W. Cao, X. Zhou, and L. Wang, "A high performance FPGA-based accelerator for large-scale convolutional neural networks,"in Proc. 26th Int. Conf. Field Program. Logic Appl. (FPL), Sep. 2016,pp. 1–9.

[30] J. Zhang and J. Li, "Improving the performance of OpenCL-based FPGA accelerator for convolutional neural network," in Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays, Feb. 2017, pp. 25–34.

[31] X. Lian, Z. Liu, Z. Song, J. Dai, W. Zhou, and X. Ji, "High-performance FPGA-based CNN accelerator with block-floating-point arithmetic,"IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 27, no. 8,pp. 1874–1885, Aug. 2019.

[32] B. Zimmer et al., "A 0.32–128 TOPS, scalable multi-chip-modulebased deep neural network inference accelerator with ground-referenced signaling in 16 nm," IEEE J. Solid-State Circuits, vol. 55, no. 4,pp. 920–932, Apr. 2020.