Stochastic Implementation of the Activation Function for Artificial Neural Networks

Injune Yeo, Sang-gyun Gi, and Byung-geun Lee
School of Electrical Engineering and Computer Science,
Gwangju Institute of Science and Technology (GIST),
Gwangju, Korea
Email: {injuneyeo, y2ksk2, bglee}@gist.ac.kr

Myonglae Chu
Imager SoC Team,
Interuniversity Microelectronics center (IMEC),
Leuven, Belgium
Email: {myonglae.chu}@imec.be

***Abstract*— One of the key elements in an artificial neural networks (ANNs) is the activation function (AF), that converts the weighted sum of a neuron's input into a probability of firing rate. The hardware implementation of the AF requires complicated circuits and involves a considerable amount of power dissipation. This renders the integration of a number of neurons onto a single chip difficult. This paper presents circuit techniques for realizing four different types of AFs, such as the step, identity, rectified-linear unit (ReLU), and the sigmoid, based on stochastic computing. The proposed AF circuits are simpler and consume considerably lesser power than the existing ones. A handwritten digit recognition system employing the AF circuits has been simulated for verifying the effectiveness of the techniques.**

Keywords*— *Artificial neural network, nonlinear activation function, neuromorphic, stochastic neuron, ananlog computing element

I. INTRODUCTION

In an artificial neural network (ANN), the probability of the postsynaptic potential firing rate is modeled by an activation function (AF). Various AFs are available and one of the popular choices is the classical sigmoid function that is mathematically defined as:

$$f(x) = \frac{1}{1+e^{-G \cdot x}}, \quad G \in R^{+} \tag{1}$$

where G is the gain factor determining the steepness of the curve.

AFs can be realized by two different approaches: approximation [1-2] and stochastic computing [3-4]. In the approximation method, lookup tables [1] and piecewise linear approximations (PLAs) [2] are commonly used for the implementation. For an accurate approximation, a sufficient number of linear segments are required for the PLA. This trade-off between the hardware complexity and the accuracy of the AFs necessitates increasingly sophisticated hardware. Furthermore, the fixed shape of the AF can result in the reduction of the synaptic plasticity. On the other hand, stochastic computing (SC) can be implemented using simple digital logic [3-4]. The SC technique has an advantage over the approximation approaches in terms of the hardware complexity.

Previous SC techniques focus on working in a digital domain. Even though the SC is spatially more efficient than the approximation approaches, realization in a digital domain tends to consume more power and area compared to its relatively simple analog counterpart. Particularly, in multiplication, several of floating points are required for evaluating the activity of each presynaptic neuron. This issue worsens with the increase in the number of synapses, as the required floating points also exponentially increase. Another issue is related to the integration, when organizing the ANNs with analogue synaptic devices such as memristors [5] and floating-gates [6] that have a high memory capacity. In an analog/digital interface, additional analog to digital converters (ADCs) [7] are needed to read the analog storage states.

This paper presents a simple architecture for the VLSI implementation of an AF that has one floating point for sensing the analog voltage. The proposed design uses a central limit theorem (CLT) for generating Gaussian random numbers, and then performs stochastic computing to generate the AF functions such as the sigmoid, rectified-linear unit (ReLU), identity, and step using only a single regenerative clocked comparator. Using a multi-layered perceptron for handwritten digit pattern recognition, the validity and learning performance of the proposed configuration are analyzed quantitatively.

The rest of the paper is organized as follows. Section II and Section III introduce the architecture of the proposed stochastic activation function and present the details of the circuit implementation. The simulation results are presented in Section IV followed by the conclusions in Section V.

II. STOCHASTIC COMPUTING ARCHITECTURE

The proposed stochastic computing is the result of applying probabilistic laws to a regenerative clocked comparator, where one of the inputs is an analog random noise and the output is a digital bit-stream represented by a pulse rate (frequency) modulation. The averaged pulse rate signifies the probability of firing rate, in this case the AF. The proposed SC configuration consists of two different parts including an analog to pulse converter (A2P) and an analog random noise generator (RNG). Their roles are as follows.

A. Analog to Pulse Converter

Typically, comparator is widely used in the ANN to implement a hard threshold (step) for the activation function. Let us assume that the regenerative clocked comparator suffers from random noise at the input terminal, as shown in Fig. 1(a). If this noise, v_n, can be described by an equivalent Gaussian random variable with a zero mean and a standard deviation, σ_n, the probability of the comparator output spike becoming "HIGH" is:

$$P_G(D=1) = \frac{1}{2}\left[1 + erf\left(\frac{z_i - v_n}{\sqrt{2}\sigma_n}\right)\right] \quad (2)$$

with

$$z_i = \sum_j w_{ij} v_j \;,\; erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2} dt \quad (3)$$

let v_j denote the output activities of the j^{th} neuron, w_{ij} is the weight of the connection from the j^{th} neuron to the i^{th}, and z_i is the weighted sum of the i^{th} neuron. For an N-time trial on the same given z_i and an independent noise, v_n, the digital output spikes are governed by a binomial distribution, $B(N,P)$. The average rate of these spikes, S, have a mean, μ_s and a standard deviation, σ_s, defined as follows:

$$\mu_S = \frac{\#\,High\_Value}{\#\,Trials} = \frac{N \cdot P_G(D=1)}{N} = P_G(D=1) \quad (4)$$

$$\sigma_S = \sqrt{\frac{P_G(D=1)\left(1 - P_G(D=1)\right)}{N}} \quad (5)$$

Fig. 1(b) shows the average rate of the comparator output depending on z_i. The steepness of S monotonically decreases as σ_n increases. On the contrary, as σ_n is considerably small, this function has a step function shape.

Alternatively, if the random signal, v_n, is chosen to be an equivalent uniform random variable with a zero mean, the probability of the comparator output becoming "HIGH" is:

$$P_U(D=1) = \frac{z_i}{V_{REF}} \quad for \;\; z_i \in [0, V_{REF}] \quad (6)$$

Repeat the preceding multiple comparison for a given z_i. Then, the averaged rate of the comparator output, L, has a similar result with the mean, μ_L and the standard deviation, σ_L, being equal to (4) and (5), respectively, when the the P_G terms are substituted by P_U in (6). Finally, L corresponds to an identity activation function, as shown in Fig. 1(b).

Through the analysis, not only was the probability of the comparator output being "HIGH" confirmed but it was also determined that the shape of the AF changes depending on the distribution features of the random noise source. As illustrated by (5), the deviation of the actual AF depends on the number of trials (N). Increasing N to ensure an appropriate accuracy results in an increased energy consumption. However, as the ANNs provide high noise immunity [8], this issue can be alleviated to a certain extent.

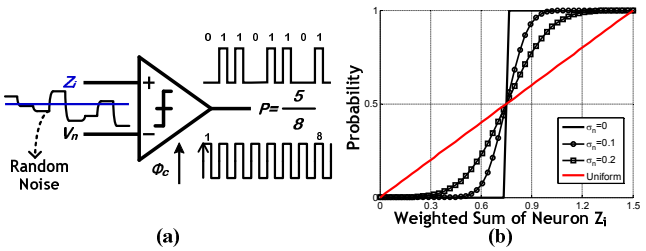

Fig. 1. *Proposed analog to pulse converter (a) stochastic computing, and (b) theoretical curve of the averaged output spikes depending on the distribution*

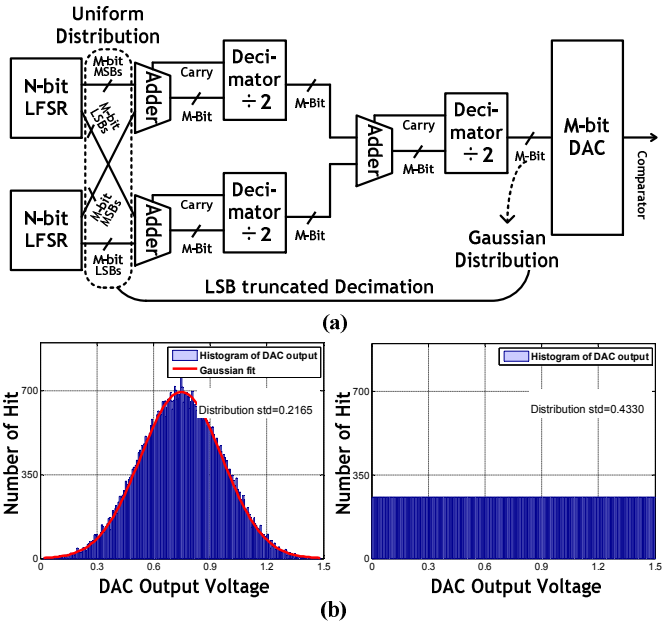

Fig. 2. *Proposed analog random noise generator (a) block diagram, and (b) DAC output histogram depending on the distribution*

B. Analog Random Noise Generator

An N-bit linear-feedback shift register (LFSR) produces up to 2^N-1 periods of uniform distribution pseudo random bit-streams. These sequences are transformed into Gaussian distribution by the central limit theorem. According to the CLT, as the sample size increases the generated samples approach a Gaussian distribution with a zero mean and a unit variance. In order to convert the pseudo random samples into analog voltages, a digital to analog converter (DAC) is employed. As the DAC has a linear one-to-one relationship between the digital input and the analog output, the Gaussian or uniform characteristics are intactly reflected.

Fig. 2 (a) shows the configuration of the proposed analog random signal generator. There are two N-bit lengths of the XOR-based LFSR. Each is divided into 2 groups, the MSBs and LSBs. Therefore, N=2M. The initial conditions of the LFSRs are different for generating better uncorrelated pseudo random signals. Likewise, to maintain uncorrelation in the decimation, the same group data are input to the adders. The addition is carried out by a simple full adder. After decimation, except for the LSB that is truncated from the outputs, (M+1) bits of the adder module, the remaining outputs are fed to an M-bit DAC. The MUX controls the type of, random bit-stream that is applied to the DAC. The DAC output distributions for

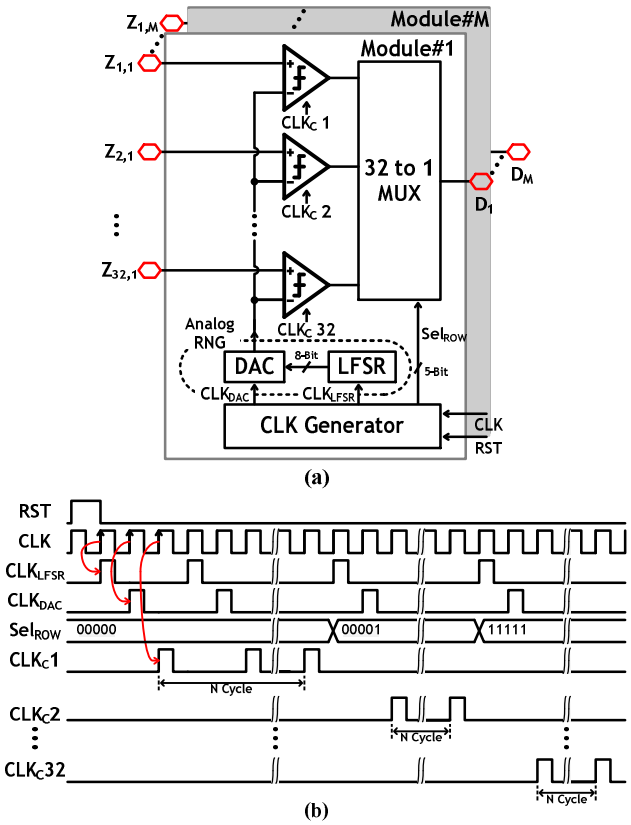

Fig.3. *(a) Configuration of the stochastic computing AF array module and (b) timing diagram*

each case are shown in Fig. 2 (b), when M=8 and with a 1.5 V reference voltage. 2^{16} samples are analyzed for each condition.

III. Hardware Implementation of the Stochastic Computing Array

In the previous section, the operating principles of a single AF structure were examined. In this section, the considerations for the hardware implementation are presented. The prototype stochastic computing AF array was fabricated in a 130 nm CMOS process. The proposed design has been devised for an 8-bit accuracy corresponding to a normal image processing and eventually applied to a handwritten digit pattern recognition. As shown in Fig. 3(a), each module consists of an analog random noise generator (RNG), 32 latch comparators, and a clock generator. The nodes at the negative input of the 32 latch comparators share the same analog RNG output. By operating a plurality of simultaneously, the parallelism can be secured..

A. Realization of four diffent types of Activation Functions

There are four situations that conversion of an analog voltage to an appropriate AF by stochastic computing. In the first case for producing the step function AF, the latch comparator does a single comparison with the common mode voltage (V_{CM}) from the analog RNG. In the second case, the latch comparator tries multiple comparisons with the Gaussian distribution noise for generating a sigmoid AF. The third case for an identity AF involves the substitution of the Gaussian by uniform distribution noise, followed by procedures similar to the second case. The last case for the ReLU AF is slightly

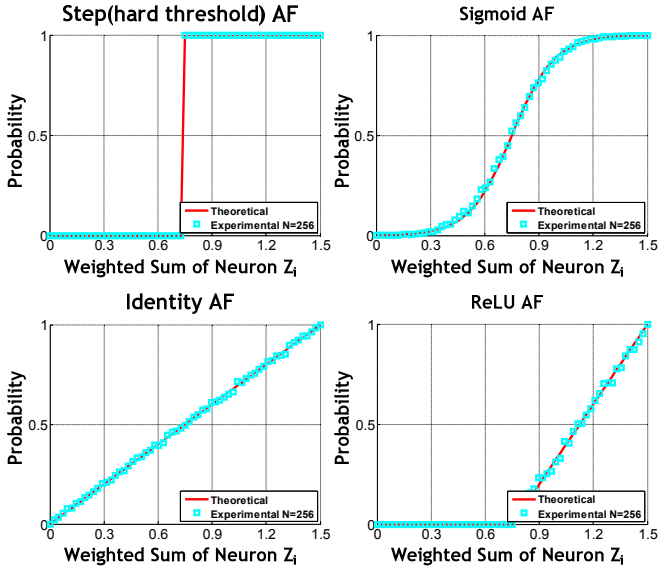

Fig.4. *Realization of four different types of AFs; step, sigmoid, identity, and ReLU based on stochastic computing*

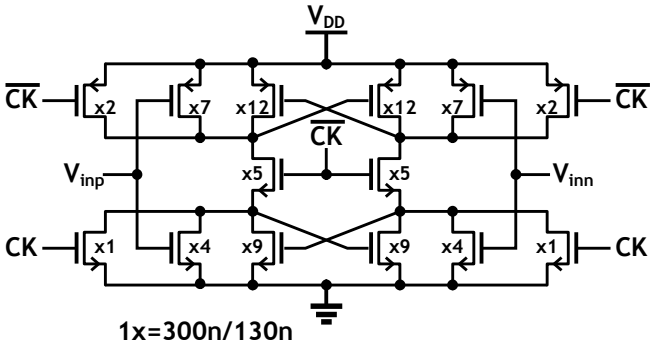

Fig.5. *Complementary input-pair latch comparator*

different from third case. The only change involved is the increase of the negative reference voltage of the DAC from 0 to V_{CM}. The post-layout simulation results of the four AF cases are shown in Fig. 4. Each circumstance is controlled by external digital options. The realization of the varied AF can support synaptic plasticity and can also enable an optimum performance to be obtained from any applied training data set.

B. Comparator Input Range and Noise

The latch comparator is the key building block for stochastic computing. Its input terminal is typically composed of an NMOS-pair or a PMOS-pair. For a single-ended circuit, there is an input voltage range limitation because of the transistor's threshold. In order to solve this issue, as depicted in Fig. 5, complementary transistors are used as the input terminals. As a result, regardless of whether the input signals are excessively low or high, either the PMOS or the NMOS works. The size of the transistor is the result of minimizing the intrinsic thermal noise of the comparator. As this noise follows a Gaussian distribution, it produces a variation in the linear AF (identity, ReLU) conversion. Increasing the width of the transistor enables the reduction of the input referred noise. The noise simulation [9] is run on a separate voltage region, where the respective transistors are more dominant. The measured

input referred noise for 0.1V (pFET), 0.7V (both), and 1.4V (nFET) are 2.86, 1.84, and 3.84, respectively, the unit is mV_{RMS}. The measurements are less than 1 LSB, when the reference voltage is equal to 1.5 V.

IV. Applying Stochastic Computing AFs to ANNs

To evaluate the performance of the proposed stochastic computing AF array, a feedforward multi-layer perceptron for handwritten digit recognition was tested. In this test, the an MNIST training data set [10] was used. Its samples consist of 8-bit gray-scales and, 28x28 pixel images of handwritten digits, 0–9. The stochastic computing network shown in Fig. 6(a), has 800 neurons (25 modules) in the visible layer, 320 (10 modules) neurons in the hidden layer, and 32 (1 module) neurons in the output layer. The MNIST images have 784 dimensions. However, the proposed AF array is composed of 32 units and the null value that does not affect the training is applied to the remaining 16 neurons in the visible layer. Additionally, only 10 neurons in the output layers are used. The weights are trained by conventional supervised learning based on back-propagation.

Figs 6.(b), (c) depicts the recognition rates depending on the number of comparisons. In each case, the stochastic computing sigmoid and ReLu AF are employed, respectively. The impacts of the DAC mismatch on realization of the sigmoid AF are shown in Fig. 7. The mismatch caused by process variation is generally considered as the main factor for the performance degradation of a VLSI system; however, it does not have a significant effect on the stochastic computing.

V. Conclusion

In this paper, the architecture and functional features of the proposed stochastic computing AFs are demonstrated. This architecture is particularly suited for large scale hardware neural networks, owing to its analog floating point and the nature of the stochastic pulse rate encoding. This configuration can generate four different types of AFs such as the step, identity, ReLU, and the sigmoid without modifications. The theoretical analysis of the proposed architecture has also been presented. The capabilities of the proposed AF array are verified by a multi-layer perceptron for the MNIST.

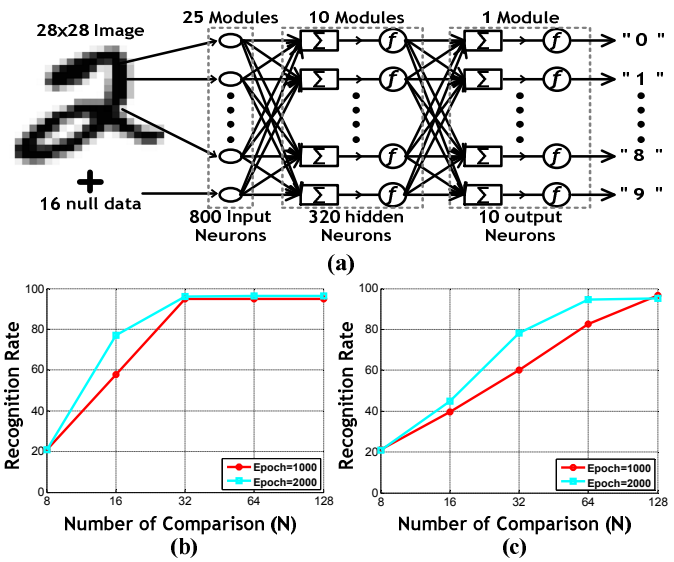

Fig.6. *Proposed AF array configuration and achieved recognition rates according to the number of trials (b) sigmoid AF (c) ReLU AF*

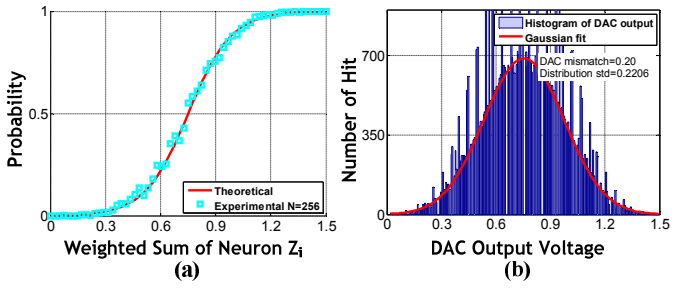

Fig.7. *(a) Realizing sigmoid AF with 20% DAC mismatch (b) its histogram*

Acknowledgment

This research was supported by the Pioneer Research Center Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT & Future Planning (2012-0009460), and Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.R7117-16-0166, Brain-Inspired Neuromorphic Perception and Learning Processor).

References

[1] K. Leboeuf, R. Muscedere, and M. Ahmadi, "Performance analysis of table-based approximations of the hyperbolic tangent activation function," in *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2011, pp. 1-4.

[2] A. H. Namin, K. Leboeuf, R. Muscedere, H. Wu, and M. Ahmadi, "Efficient hardware implementation of the hyperbolic tangent sigmoid function," in *2009 IEEE International Symposium on Circuits and Systems*, 2009, pp. 2117-2120.

[3] Y. Ji, F. Ran, C. Ma, and D. J. Lilja, "A hardware implementation of a radial basis function neural network using stochastic logic," in *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2015, pp. 880-883.

[4] M. Martincigh and A. Abramo, "A new architecture for digital stochastic pulse-mode neurons based on the voting circuit," *IEEE Transactions on Neural Networks,* vol. 16, pp. 1685-1693, 2005.

[5] H. Li, B. Liu, X. Liu, M. Mao, Y. Chen, Q. Wu, *et al.*, "The applications of memristor devices in next-generation cortical processor designs," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 17-20.

[6] F. Tenore, R. J. Vogelstein, R. Etienne-Cummings, G. Cauwenberghs, and P. Hasler, "A floating-gate programmable array of silicon neurons for central pattern generating networks," in *2006 IEEE International Symposium on Circuits and Systems*, 2006, pp. 4 pp.-3160.

[7] M. S. J. Tomlinson, D. J. Walker, and M. A. Sivilotti, "A digital neural network architecture for VLSI," in *Neural Networks, 1990 IJCNN International Joint Conference on*, 1990, pp. 545-550.

[8] S. Haykin, *Neural Networks and Learning Machines* (3rd Edition), Prentice Hall, 2009.

[9] P. Nuzzo, F. D. Bernardinis, P. Terreni, and G. V. d. Plas, "Noise Analysis of Regenerative Comparators for Reconfigurable ADC Architectures," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 55, pp. 1441-1454, 2008.

[10] Y. LeCun and C. Cortes, "The MNIST database of handwritten digits," ed, 1998.