A Low-Latency and High-Accuracy Dual-mode Neuron Design for Accelerating Neurological Diseases Simulation and Analysis

Jiatong Guo[1], Jinxiang Gao[2], Zixuan Shen[1], Jipeng Wang[1], Zhuo Cheng[2], Jingru Jiang[2], Wenjue Chen[2], Chao Wang[1]

[1] School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan, China

[2] School of Integrated Circuits, Huazhong University of Science and Technology, Wuhan, China

***Abstract*—Biological realism and computational efficiency are crucial for modeling neurological diseases with spiking neural networks (SNNs), as biological realism is necessary for observing neuron ion channel behaviors and computational efficiency is essential for simulating the action potentials of large-scale networks. However, existing spiking neuron models cannot achieve both high biological realism and computational efficiency, resulting in SNN constructed from a single type of neuron to make a compromise between these two attributes, thus reducing the SNN effectiveness in disease simulation and analysis. In this paper, we propose a dual-mode spiking neuron hardware design with an efficient reconfigurable architecture to achieve both biological realism and computational efficiency for diseases modeling. By exploiting the common arithmetic operators in Hodgkin-Huxley neuron and Adaptive Exponential (AdEx) neuron, our design can reuse the computational units including adder, multiplier, and CORDIC to efficiently realize these two neurons. An optimized pipeline design based on data flow dependency and Reconfigurable Fast-Convergence CORDIC is proposed to reduce overall computation latency, while a dynamic bit-width allocation strategy is employed to improve the implementation accuracy. FPGA implementation result shows that our design significantly improves computation latency and accuracy compared to previous neuron designs.**

***Keywords*—Reconfigurable Spiking Neuron, Hodgkin-Huxley Neuron, Adaptive Exponential Neuron, Fast-Convergence CORDIC, Neuromorphic Hardware.**

I. Introduction

Spiking neural network (SNN), which emulates real biological neural systems through spike-based information coding, holds potential for applications in neural phenomena studying and disease modeling [1]. In the specific field of using SNN to simulate and analyze neurological diseases, such as Fabry disease [2] and epilepsy, excellent biological realism is required for the dynamic observation of the abnormal changes in action potentials and ion channels, which helps clinicians deduce the pathogenesis of neurological diseases and guide targeted drug therapy. At the same time, high computational efficiency is also demanded for constructing large-scale neural network to simulate abnormal synchronous discharges in brain regions. Existing mainstream bio-inspired spiking neuron models fall into two categories: one is highly biologically-realistic neurons such as HH (Hodgkin-Huxley) neuron, and the other is simplified computationally-efficient IF (Integrated and Fire) neuron family. In this study, HH and AdEx (Adaptive Exponential) neuron models are chosen from these two categories by not only pursuing both biological realism and computational efficiency, but also considering the key operations including exponentiation and multiplication in computational process that can be shared by these two models in reconfigurable design.

Approximate computation and direct computation are two common approaches in hardware implementation of HH and AdEx neurons. Typical approximate computation methods include Piece Wise Linear (PWL) and base-2 function [3], which only use shifters and adders to fit the computation-intensive arithmetic operators including multiplication, division and exponentiation in the neuron equations. These strategies achieve low hardware overhead but sacrifice accuracy resulting from inevitable fitting errors and considerable rounding errors. In contrast, the direct calculation method such as CORDIC (Coordinate Rotation Digital Computer) algorithm that also only uses shifters and adders can avoid fitting errors and achieve higher accuracy through adjusting iteration cycles [4][5]. However, there exists redundant iterations in the conventional CORDIC, resulting in long computation cycles and poor accuracy. For the neuron design in [4], all the nonlinear terms are performed by CORDIC with considerable iteration cycles, of which data flow dependency introduces unavoidable long waiting time in computation units. To sum up, there are three critical issues in existing works:

1) All the existing works [3][4][5] are only designed to support single kind of spiking neuron. As the mainstream spiking neuron models cannot achieve both high biological realism and computational efficiency, the existing single-mode neuron designs have to make a compromise between these two attributes by choosing HH or AdEx neuron for hardware implementation.

2) The non-approximation pipelined HH design in [4] uses CORDIC to calculate massive multiplication operations with long iteration cycles, and therefore, the data dependency on intermediate calculation results introduces excessive waiting time during the computation. As a result, this kind of conventional CORDIC based HH design suffers both long latency and poor accuracy, jointly contributed by poor pipeline design and long redundant iterations.

3) Most existing neuron designs, e.g., the AdEx neuron design in [5], simply implements static bit-width allocation between integer and fraction parts, which ignores specific bit-width requirement for different parameters in the model equation, thereby leading to considerable rounding errors and lowering model accuracy. Moreover, the existing HH neuron designs either use the PWL or base-2 function to do approximate computation [3] or employ the conventional CORDIC with redundant iterations to compute all the nonlinear terms [4], further reducing model accuracy.

To solve the aforementioned issues, a low-latency and high-accuracy dual-mode spiking neuron hardware design with Reconfigurable-Fast-Convergence CORDIC (RFC-CORDIC) is proposed. There are three major contributions as follows. *1)* A dual-mode architecture with RFC-CORDIC,

TENCON 2024 - 2024 IEEE Region 10 Conference (TENCON) | 979-8-3503-5082-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/TENCON61640.2024.10902968

which reuses computational units including the RFC-CORDIC, multiplier and adder in a time-sharing manner to perform the common operators of HH and AdEx neurons, is proposed to support both neuron models and reduce overall hardware overhead when compared to implementing these two neurons separately. *2)* An optimized pipeline design efficiently exploiting data dependency in the data flow is implemented, which utilizes multiplier and Fast-Convergence CORDIC to obtain intermediate results as early as possible for effectively reducing excessive waiting cycles, thereby significantly reducing computation latency in both neuron modes. *3)* A dynamic bit-width allocation strategy is proposed to improve fixed-point model accuracy. Pre-shift is employed for specific parameters to reduce rounding errors and fast-convergence CORDIC is adopted to reduce iteration error accumulation, thereby achieving comparable accuracy in AdEx with less bit-width and also improving accuracy in HH.

II. Spiking Neuron Model

A. Hodgkin-Huxley (HH) Neuron Model

The HH neuron model is a pioneering conductance-based mathematical models describing the details of biological mechanisms happened in biological neurons. Through Euler discretization, the re-organized HH formula is given as:

$$V_{n+1} = V_n + \frac{\left[G_{Na}m^3h(E_{Na}-V_n)dt + G_K n^4(E_K-V_n)dt + G_l(E_l-V_n)dt + Idt\right]}{C_m} \quad (1)$$

$$x_{n+1} = x_n + [\alpha_x(1-x) - \beta_x x]dt \ , \ x = m,n,h \quad (2)$$

where

$$\alpha_m = \frac{0.1(V+35)}{1-e^{-(0.1V+3.5)}} \ , \ \alpha_n = \frac{0.01(V+50)}{1-e^{-(0.1V+5)}} \ , \ \beta_h = \frac{1}{1+e^{-(0.1V+3)}} \quad (3)$$

$$\beta_m = 4e^{\frac{-(V+60)}{18}} \ , \ \beta_n = 0.125e^{\frac{-(V+60)}{80}} \ , \ \alpha_h = 0.07e^{\frac{-(V+60)}{20}} \quad (4)$$

In the above equations, V is membrane potential, I is input current, while n, m, and h are probabilities associated with potassium channel activation, sodium channel activation and inactivation, respectively.

In (1), dt is multiplied by the term $(E_x - V_n)$ and I in advance to reduce integer bit-width. In order to turn two times of multiplication into one, changing the calculation sequence transforms (2) into (5) as described by

$$x_{n+1} = x_n + [\alpha_x - (\alpha_x + \beta_x)x]dt \quad , x = m,n,h \quad (5)$$

$$\alpha_m = \frac{0.1(V+35)}{1-e^{-(0.1V+3)}e^{-0.5}}, \ \alpha_n = \frac{0.01(V+50)}{1-e^{-(0.1V+3)}e^{-2}}, \ \beta_h = \frac{1}{1+e^{-(0.1V+3)}} \quad (6)$$

Extracting the common part of exponential term in (3), equation (6) is derived so as to simplify three times of exponent operations into one at the expense of two extra multiplications. The above transformations collaboratively contributes to a low-latency pipeline design which will be described in Section III.

B. Adaptive Exponential (AdEx) Neuron Model

The AdEx neuron model is an improved version of the integrate-and-fire (IF) neuron family with much lower computation complexity as compared to the HH neuron. The re-organized and discretized formula is given as:

$$\begin{cases} V_{n+1} = A_1V_n + exp\left(\frac{V_n - V_T}{\Delta T} + \ln A_2\right) + A_3 + [(A_4 \ll 1)\omega_n] \gg 1 \\ \omega_{n+1} = B_1\omega_n + B_2V_n + B_3 \end{cases} \quad (7)$$

$$if \ V_{n+1} > 0 \ or \ V_n - V_T > 20 \ then \begin{cases} V_{n+1} = V_r \\ \omega_{n+1} = \omega_n + b \end{cases} \quad (8)$$

$$\begin{cases} A_1 = 1 - g_L dt/C, A_2 = g_L \Delta T dt/C, A_3 = (I + g_L E_L)\, dt/C, A_4 = -\, dt/C \\ B_1 = 1 - dt/\tau_\omega, B_2 = \alpha dt/\tau_\omega \ , B_3 = -\alpha E_L\, dt/\tau_\omega \end{cases} \quad (9)$$

where V is the membrane potential and ω is the adaptation current.

Compared to the original formula, multiple coefficients for the same variable are extracted and summed up in (9) to simplify the hardware implementation. In (7), coefficient A_2 is taken as the logarithm and integrated into the input of exponential computation, which not only reduces one multiplication calculation, but also decreases the numerical value of exponential term and hence cuts down on the requirement for integer bit-width. Additionally, pre-shift and post-shift by 1 bit operations are employed for the term $A_4\omega_n$ in the fixed-point design to transform one vacant integer bit-width into fraction part, thus reducing rounding error and improving overall accuracy.

III. Proposed Dual-mode Spiking Neuron Design

A. RFC-CORDIC Design

Conventional CORDIC algorithm can efficiently compute complex functions using only shift-and-add operations in an iterative manner at low hardware overhead. The uniform iterative formulas of CORDIC in different rotation systems are described by

$$\begin{cases} x_{i+1} = x_i + \mu d_i\left(2^{-i}y_i\right) \\ y_{i+1} = y_i + d_i\left(2^{-i}x_i\right) \\ z_{i+1} = z_i - d_i\theta_i \end{cases} \quad (10)$$

where μ and θ_i depend on the specific calculation modes. The proposed RFC-CORDIC design sets μ=1 and θ_i=$\tan^{-1}2^{-i}$ for exponent calculation, while defining μ=0 and θ_i=2^{-i} for division calculation.

The Fast-Convergence CORDIC proposed by C. Wang *et al.* in [6] can reach the target value more quickly by choosing the optimal angle for each iteration, thus eliminating redundant iterations and reducing error accumulation, which can achieve up to 50% speed improvement with higher accuracy and energy efficiency.

Fig. 1 presents architecture and data flow chart of RFC-CORDIC used for exponent and division calculations in our proposed dual-mode neuron design. The core idea arises from the observation that these two arithmetic calculations can be conveniently executed through the same CORDIC iteration of y_i and z_i, thus reducing hardware overhead by sharing the iteration unit and optimal angle selection unit [7].

B. Proposed Dual-mode Neuron Design

Equations (1)(2)(7)(8) show that common computations for one Membrane Potential Update (MPU) in both HH and AdEx neurons includes additions, multiplications and exponent calculations, which can be realized by reconfigurable design reusing a set of commonly-shared computation units. Moreover, these formulas also manifest the data dependency for pipeline design. e.g., the calculation of α_m must wait until its denominator and its numerator are determined, while the computation of its denominator must wait for the exponent computation result.

Fig. 2 presents the overall pipeline architecture of dual-mode neuron design. The core circuit of pipeline is the computation unit consisting of one multiplier, one adder, and three RFC-CORDIC modules for EXP and DIV computations to balance the computation latency and resource overhead. Given the data flow dependency during MPU, one multiplier

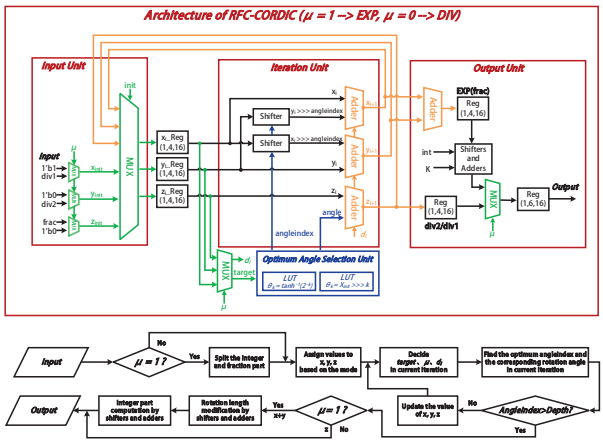

Fig. 1. Architecture diagram and data flow chart of RFC-CORDIC.

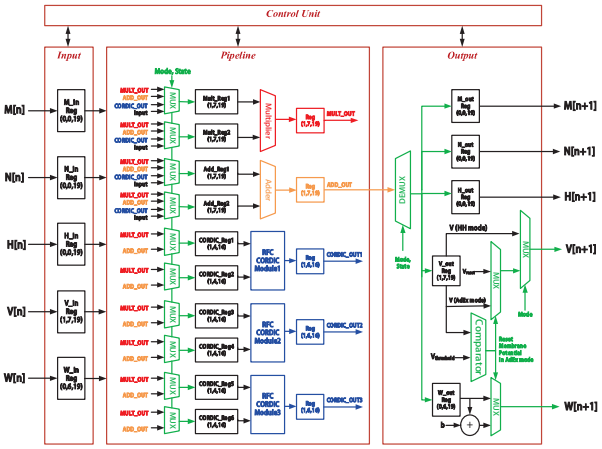

Fig. 2. Overall pipeline architecture of proposed dual-mode neuron.

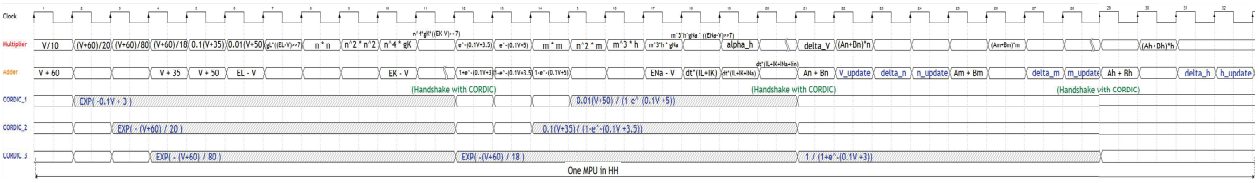

Fig. 3. Pipeline scheduling diagram of proposed neuron design in HH neuron mode.

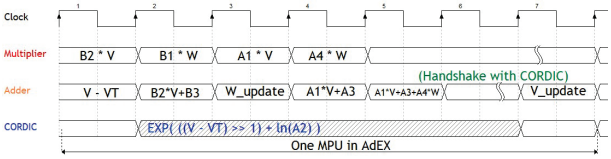

Fig. 4. Pipeline scheduling diagram of proposed neuron design in AdEx neuron mode.

is used for multiplications to eliminate massive waiting cycles resulted from the CORDIC-based multiplications in [4]. In addition, three RFC-CORDIC modules are adopted to calculate exponent terms in both neurons, as well as to be reconfigured to perform division operations in HH neuron. The input unit prepares necessary variables and parameters for pipelining, while the output unit stores the output signals and also resets membrane potential in AdEx mode. The control unit provides pipeline control and neuron mode selection signals.

Fig. 3 and Fig. 4 show pipeline scheduling diagram of HH and AdEx neurons, respectively. There are three design principles in the pipeline scheduling to achieve much shorter computational latency than the existing neuron designs [4][5]. Firstly, the three RFC-CORDIC modules should start respective calculations as early as possible, as the inherent iterative computation has the greatest impact on the pipeline's latency. Secondly, the multiplier and adder should compute the CORDIC module's input in highest priority. Thirdly, correct handshake timing with the CORDIC modules should be chosen to minimize the waiting cycles. The handshake moment in our design is based on the average CORDIC cycles for different computation which is obtained from Python-based fixed-point model simulation.

IV. FPGA Implementation Results and Discussion

The proposed dual-mode neuron design is implemented on Xilinx Zynq-7000 FPGA (xc7z100ffg900). A 27-bit signed bit-width (i.e., 1 sign-bit, 7 integer-bits, 19 fraction-bits) is employed, while the input and output bit-width for RFC-CORDIC module is 21-bit (1,4,16) and 23-bit (1,6,16), respectively.

TABLE I. Comparison Table of Different Methods for Multiplication, Division and Exponent Computation

Method	Multi-plier	Conventional CORDIC			FC -CORDIC		RFC-CORDIC
Operation	MUL	MUL	DIV	EXP	DIV	EXP	DIV&EXP
Bit-width	1+7+19 bit	Input: 1+4+16 bit, Output: 1+6+16 bit					
LUTs	38(833[d])	296	281	545	635	982	1336
Registers	76 (27)	117	95	123	111	123	146
DSP	4 (0)	0	0	0	0	0	0
Average Iterations	N/A	16	16	16	5.8	7.8	5.8 & 7.8 [e]
Max Frequency (MHz)	495.0 (202.4)	212.9	222.6	145.5	129.3	126.0	112.8
Latency [a] (μs)	0.013	0.180	0.180	0.204	0.095	0.112	0.084 & 0.111
Energy [a] (pJ)	130	1881	965	3435	1597	4501	2531 & 4607
MAE [b]	5e-5	0.067	1.7e-5	0.013	8.8e-6	0.002	8.8e-6 & 0.002
MRE [c]	0.000 %	0.64 %	0.009 %	0.17 %	0.004 %	0.06 %	0.004% & 0.06%

[a.] Latency or energy cost per operation, normalized to 100MHz. [b.] MAE: Mean Absolute Error. Input scanning range: MUL [-100, +100], DIV [-10, +10], EXP [-4, +4]. [c.] MRE: Mean Relative Error. [d.] Results from No-DSP implementation. [e.] Results of EXP operation.

Table I shows comparison of different methods for multiplication (MUL), division (DIV) and exponent (EXP) computation. As compared to the conventional CORDIC, multiplier actually has much better performance on latency, accuracy and energy efficiency at cost of more hardware overhead. In fact, the underlying logic of both CORDIC in linear system and multiplier are all shift-and-add operations. However, CORDIC inherently involves iterative processes that requires more clock cycles and also suffers error accumulation, while multiplier normally compute MUL in parallel manner that has much shorter latency and better accuracy. The FC-CORDIC can achieve up to 50% improvement in latency and higher accuracy than the conventional CORDIC due to the elimination of redundant iterations and error accumulation, at the expense of additional hardware overhead for optimal angle selection unit. The RFC-CORDIC has more resource overhead for supporting both

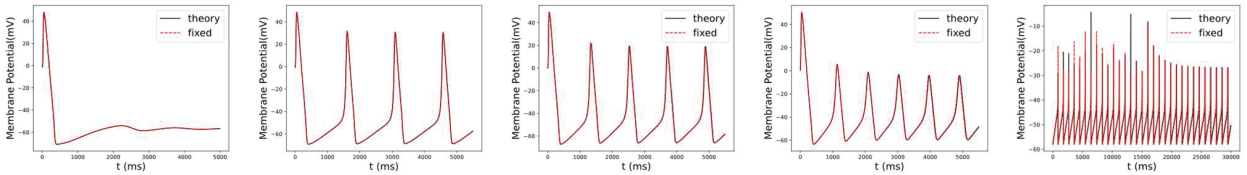

Fig. 5. Membrane potential comparisons between dual-mode neuron model (post-implementation simulation, in red) and original neuron model (Python simulation, in black). HH neuron: (a) I = 5μA, (b) I = 20μA, (c) I = 40μA, (d) I=80μA. AdEx neuron:(e) I = 500pA

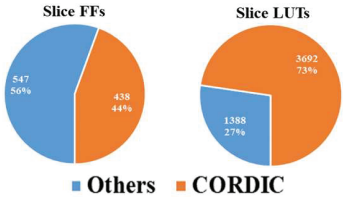

Fig. 6. The hardware overhead proportion of RFC-CORDIC modules in the proposed dual-mode neuron design.

EXP and DIV, but still save up to 19% LUTs and 38% FFs by resource-sharing as compared to using two FC-CORDIC to conduct EXP and DIV calculations, respectively. Fig. 5 shows the membrane potential comparisons between proposed dual-mode neuron and the original neuron models both in tonic spiking mode, which clearly shows that our design can achieve excellent accuracy. The Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Normalized RMSE (nRMSE) [4] are also used to quantitatively measure the accuracy of our design, as presented in Table II.

Table II shows comparisons between the proposed dual-mode neuron and state-of-the-art neuron designs. The approximation-method design in [3] sacrifices too much accuracy to achieve lower hardware cost in Slice FFs. To achieve dual-mode reconfigurability, low latency and high accuracy, our design has larger hardware overhead than the standalone CORDIC-based neuron designs, which is majorly contributed by the RFC-CORDIC modules as shown in Fig. 6. However, our design achieves 4.8× and 2.0× latency improvement in HH and AdEx neuron, respectively, thanks to the optimized pipeline design efficiently exploiting the data flow dependency and fast-convergence CORDIC. Benefiting from the low-error multiplier and RFC-CORDIC, our design achieves up to 3.03× accuracy improvement in HH neuron and comparable accuracy in AdEx neuron with less bit-width from the dynamic bit-width allocation strategy for parameters. In addition, our proposed design has achieved higher energy efficiency than the design in [5], i.e., 34% lower energy consumption per MPU, due to the adoption of high energy-efficiency multiplier and RFC-CORDIC.

V. Conclusion

This paper proposes a low-latency and high-accuracy dual-mode spiking neuron hardware design which can support both HH and AdEx neurons. The hardware implementation is featured with a low-latency pipeline design efficiently exploiting data flow dependency, an efficient dual-mode design based on Reconfigurable Fast-Convergence CORDIC, as well as a high-accuracy design enabled by a dynamic bit-width allocation strategy and the Fast-Convergence CORDIC. Implementation result shows the proposed design can achieve a significant improvement on latency and accuracy against the state-of-the-art neuron designs.

TABLE II. FPGA Implementation Comparisons Between This Work and Previous Works on HH Neuron and AdEx Neuron

Design	TCAS-I 2021 [3]	T-VLSI 2023 [4]	The Proposed Design		TCAS-I 2016[e] [5]
Neuron	HH	HH	HH	AdEx	AdEx
Method	Power-2	CORDIC	RFC-CORDIC		CORDIC
FPGA Platform [a]	Virtex-4 90nm 1.2V	Zynq-7 28nm 1.0V	Zynq-7 28nm 1.0V		Zynq-7 28nm 1.0V
Computation Efforts	N/A	3 DIV(8[d]) 6 EXP(10) 17 MUL(14)	3 DIV(5.8[d]) 4 EXP(6.4) 23 MUL	1 EXP 4 MUL	1 EXP 5 MUL
Slice LUTs	2344	2256	5080		2715
Slice FFs	480	1598	985		455
DSP	0	0	4		0
Bit-width	10+20	10+12	8+19		14+23
Max Freq.	200 MHz	144.2 MHz	111.2 MHz		102 MHz
Latency [b]	N/A	2.25 us	0.467 us	0.220 us	0.442 us
Energy Efficiency [b]	N/A	N/A	23.1 nJ	6.4 nJ	18.58 nJ
RMSE [c]	7.4	0.255	0.081	0.0156	N/A
nRMSE	9.34%	0.30%	0.10%	0.046%	0.040%
MAE	5	0.215	0.058	0.011	N/A

[a.] Process node and core voltage of FPGA. [b.] Latency or energy cost per membrane potential update, normalized to 100MHz. [c.] Both neurons are measured in tonic spiking mode. [d.] Average iterations number for CORDIC. [e.] Our re-implemented results in Tonic Spiking mode.

Acknowledgment

This work was supported by National Natural Science Foundation of China (61974053) and Fundamental Research Funds for the Central Universities, HUST (2024JYCXJJ063). Jiatong Guo and Jinxiang Gao equally contributed to this paper. Corresponding author: Chao Wang.

References

[1] H. Amin, "Spiking Neural Networks Learning, Applications, and Analysis," Ph.D. dissertation, Graduate Dept. Comput. Syst., Univ. Aizu, Aizuwakamatsu, Japan, Sep. 2006.

[2] B. Namer *et al.*, "Changes in Ionic Conductance Signature of Nociceptive Neurons Underlying Fabry Disease Phenotype," Frontiers Neurol., vol. 8, p. 335, Jul. 2017.

[3] S. Haghiri *et al.*, "High Speed and Low Digital Resources Implementation of Hodgkin–Huxley Neuronal Model Using Base-2 Functions," IEEE TCAS I, vol. 68, no. 1, pp. 275–287, Jan. 2021.

[4] A. J. Leigh *et al.*, "A Resource-Efficient and High-Accuracy CORDIC-Based Digital Implementation of the Hodgkin–Huxley Neuron," IEEE T-VLSI, vol. 31, no. 9, pp. 1377-1388, Sept. 2023.

[5] M. Heidarpour *et al.*, "A CORDIC Based Digital Hardware for Adaptive Exponential Integrate and Fire Neuron," IEEE TCAS I, vol. 63, no. 11, pp. 1986-1996, Nov. 2016.

[6] C. Wang *et al.*, "A 1-V to 0.29-V Sub-100-pJ/operation Ultra-Low Power Fast-Convergence CORDIC Processor in 0.18-μm CMOS", in MEJ, Elsevier, vol. 76, pp. 52-62, 2018.

[7] B. Wang *et al.*, "A Reconfigurable High-Precision and Energy-Efficient Circuit Design of Sigmoid, Tanh and Softmax Activation Functions," in IEEE ICTA, Hefei, China, 2023.