IEEE Xplore® Digital Library

ISCAS 2020

2020 IEEE International Symposium on Circuits and Systems
Virtual, October 10-21, 2020

CAS

CORDIC-SNN: On-FPGA STDP Learning With Izhikevich Neurons

M. Heidarpur, A. Ahmadi, M. Ahamdi, M. Azghadi

University of Windsor, James Cook University

2020 IEEE International Symposium on Circuits and Systems
Virtual, October 10-21, 2020

University of Windsor

JAMES COOK UNIVERSITY AUSTRALIA

Outline

- **Background**
- **Problem statement**
- **Proposed solution**
- **Results and discussion**
- **Conclusion**
- **Future Work**

Background

Neuromorphic Systems

- **Definition**
  - A large number of neurons, synapses and their interconnecting structure on hardware
- **Properties**
  - Highly parallel, fast, fault tolerant and intelligent
- **Applications**
  - Information processing
  - Medical diagnosis

Fig. 1 : Neuromorphic systems are inspired by brain

Background

Spiking Neural Network

- **Definition**
  - Third generation of neural networks in which components communicate through sequences of spikes
- **Properties**
  - Biologically inspired
  - Sparse spikes
  - Incorporate the concept of time
- **Advantages**
  - Biological meaningful, energy efficient and fast.

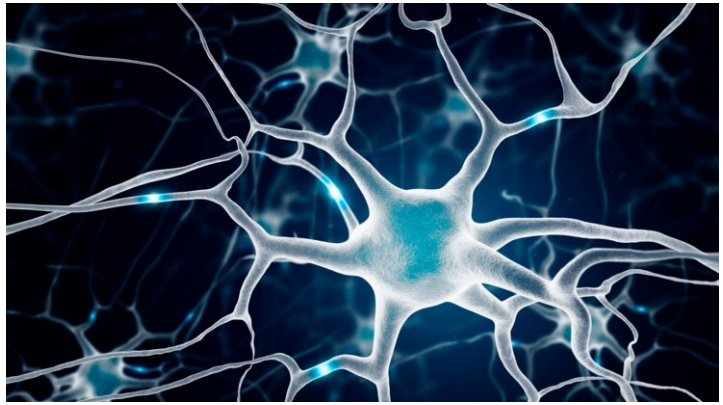


Fig. 2 : Spiking neurons imitate biology

Background

Spiking Neural Network

- **Models**
  - Plethora of models to mimic real neurons, astrocytes and synapses with different levels of abstraction
- **Properties**
  - Biologically-plausible
    - Hodgkin Huxley, Li-Rinzel
  - Biologically-inspired
    - Izhikevich, Postnov
  - High-level models
    - IF, LIF
- **Selecting the models**
  - Application
  - Available resources

Problem Statement

Implementation Platforms

- **The challenge**
  - High complexity of biological networks, due to number of cells and numerous communication pathways, makes such implementation a difficult task.
- **Platfroms**
  - Software
  - Hardware
    - ASICS
    - Re-configurable platforms

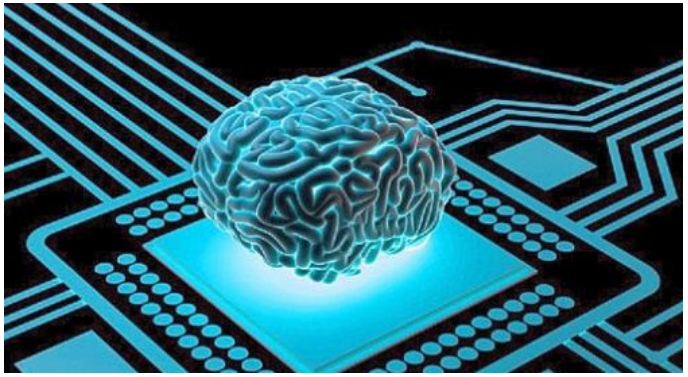


Fig. 3 : A computer as efficient, intelligent and fault tolerant as brain

Problem Statement

ASIC Versus FPGA

- **ASIC**
  - Fast, parallel, power efficient
  - Expensive, long design process, not possible to update design
- **FPGAs**
  - Cheap, easy to work with, fast, parallel
  - Limited resources

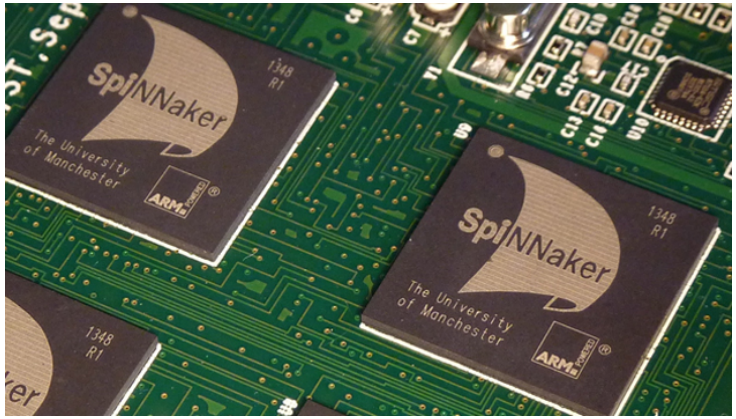


Fig. 4 : SpiNNaker: A neuromorphic chip

Problem Statement

FPGA Implementation Challenges

- **Nonlinear terms**
  - Neurons, astrocytes and synapses are typically described using Ordinary Differential Equations (ODEs).
- **Large scale architecture**
  - Large number of cells and numerous communication pathways

Proposed Solutions

CORDIC

- **COordinate Rotation DIgital Computer**
  - Why CORDIC?
    - High precision
    - Well suited for hardware implementation
  - An iterative algorithm to calculate nonlinear functions
    - First iteration, this vector is rotated 45°
    - The direction of rotation is determined
    - The new angle determined by step size
    - New value is calculated

- B>45
- Acc=Acc+f(sin(45))
- next iteration: 45+45/2
- B<45+45/2
- then Acc=Acc-f(sin(45/2))
- next iteration: 45+45/2-45/4

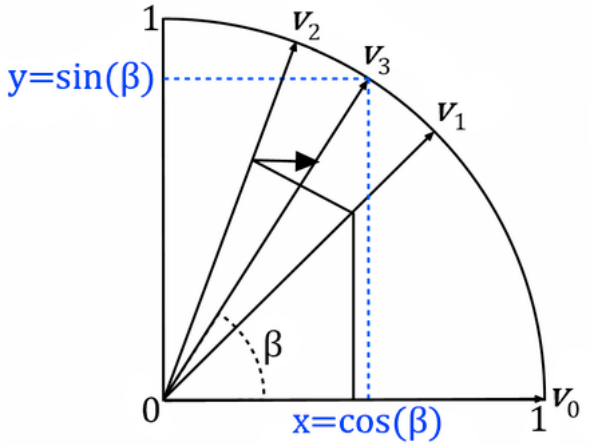


Fig. 5 An illustration of the CORDIC algorithm in progress

Proposed Solution

Izhikevich Neuron

- **Izhikevich Neuron ODEs [1]**

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I$$

$$\frac{du}{dt} = a(bv - u)$$

$$if \;\; v > 30mv \;\; then \;\; \begin{cases} v \rightarrow v_r \\ u \rightarrow w_r = u + d. \end{cases}$$

Where

v: Membrane potential
I : Applied current
b: Sensitivity of the u to v
d: After-spike reset value of the u

u : Recovery variable
a: Time scale of the recovery variable
c: After-spike reset value of the v

Proposed Solution

STDP

- **Spike Time Dependent Plasticity (STDP) [2]**

  The synaptic weight changes when a pre-synaptic neuron fires in a short time before or after the post-synaptic neuron

  Weight change is determined as [2]:

$$\begin{cases} w_i(\Delta t) = +A_+ \, e^{-\Delta t/\tau_+} & if \quad \Delta t > 0 \\ w_i(\Delta t) = -A_- \, e^{+\Delta t/\tau_-} & if \quad \Delta t < 0 \end{cases}$$

  Where:

  $\Delta t = t_{post} - t_{pre}$, τ_+, τ_-: Learning windows, A_+, A_-: Gain parameters

- **Challenges**
  - Exponential term in STDP
  - Storing the spike times

Proposed Solution

CORDIC Neuron and STDP

- **CORDIC is an iterative algorithm**
  - Different neuron models developed:
    - IZHCOR6
    - IZHCOR8
    - IZHCOR10
    - IZHCOR12
- **Network Topology**
  - Feed-forward Network
  - Input layer: 20 neuron, Output Layer: 1 neuron
  - CORDIC and 2^x model

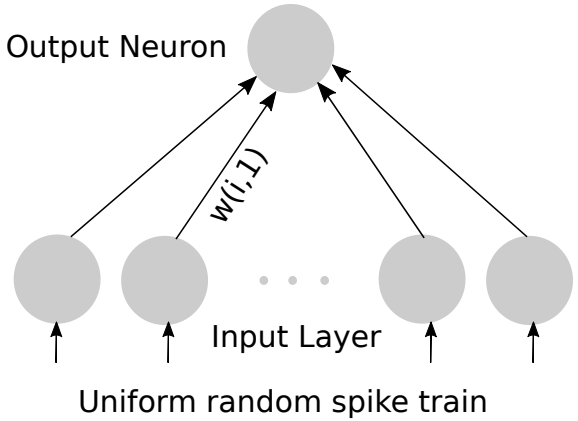
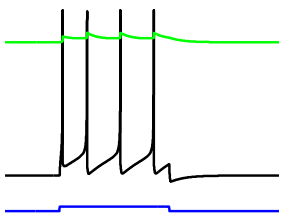


Fig. 6 : Spiking neural network and its connections

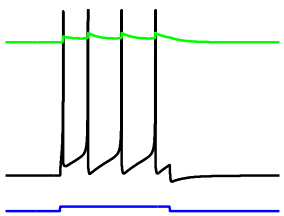
Proposed Solution

Simulation

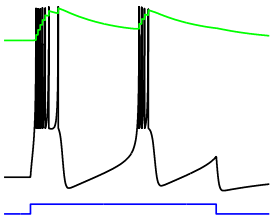
- **CORDIC Izhikevich simulation**



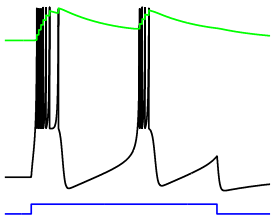
Tonic Spiking-Original Model



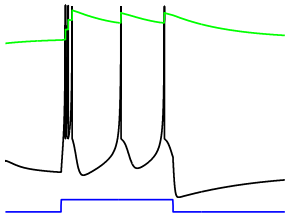
Tonic Spiking-Proposed Model



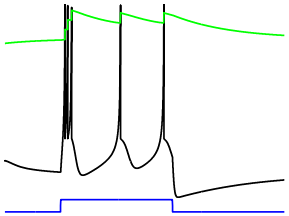
Tonic Bursting-Original Model



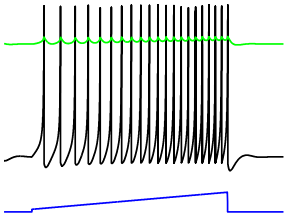
Tonic Bursting-Proposed Model



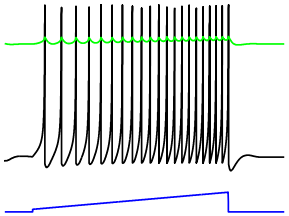
Mixed Mode-Original Model



Mixed-Proposed Model



Class II Excitability-Original Model



Class II Excitability-Proposed Model

Fig. 7: Computer simulation of the original and cordic model (IZHCOR10) for different neuronal behaviours.

Proposed Solution

Simulation

- **Bifurcation analysis**

$$\begin{cases} \dfrac{dv}{dt} = 0 \\ \dfrac{du}{dt} = 0 \end{cases} \Longrightarrow \begin{cases} 0.04v^2 + 5v + 140 - u + I = 0 \\ bv - u = 0 \end{cases}$$

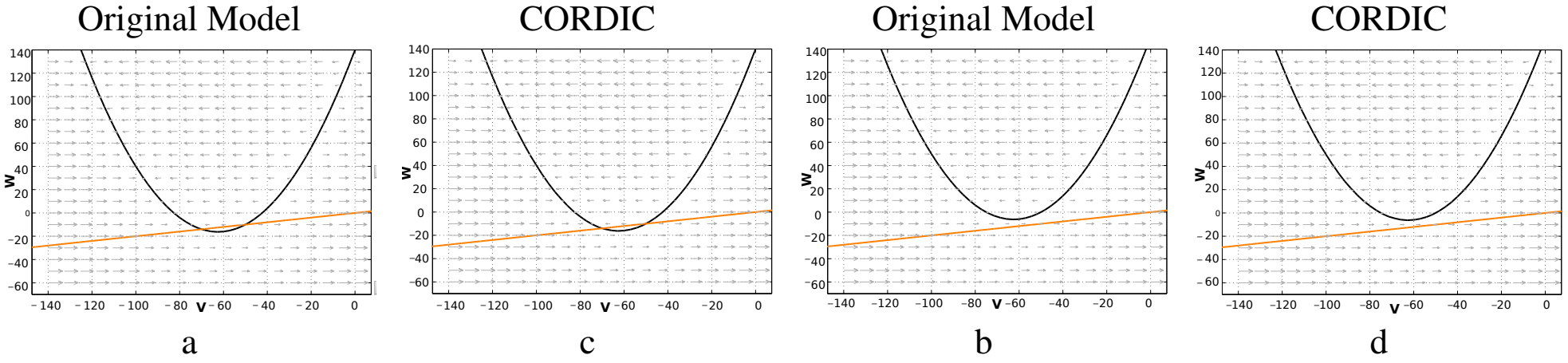


Fig. 8: Nullclines of original and CORDIC model.

Proposed Solution

Simulation

- **Bimodal weight distribution in STDP**

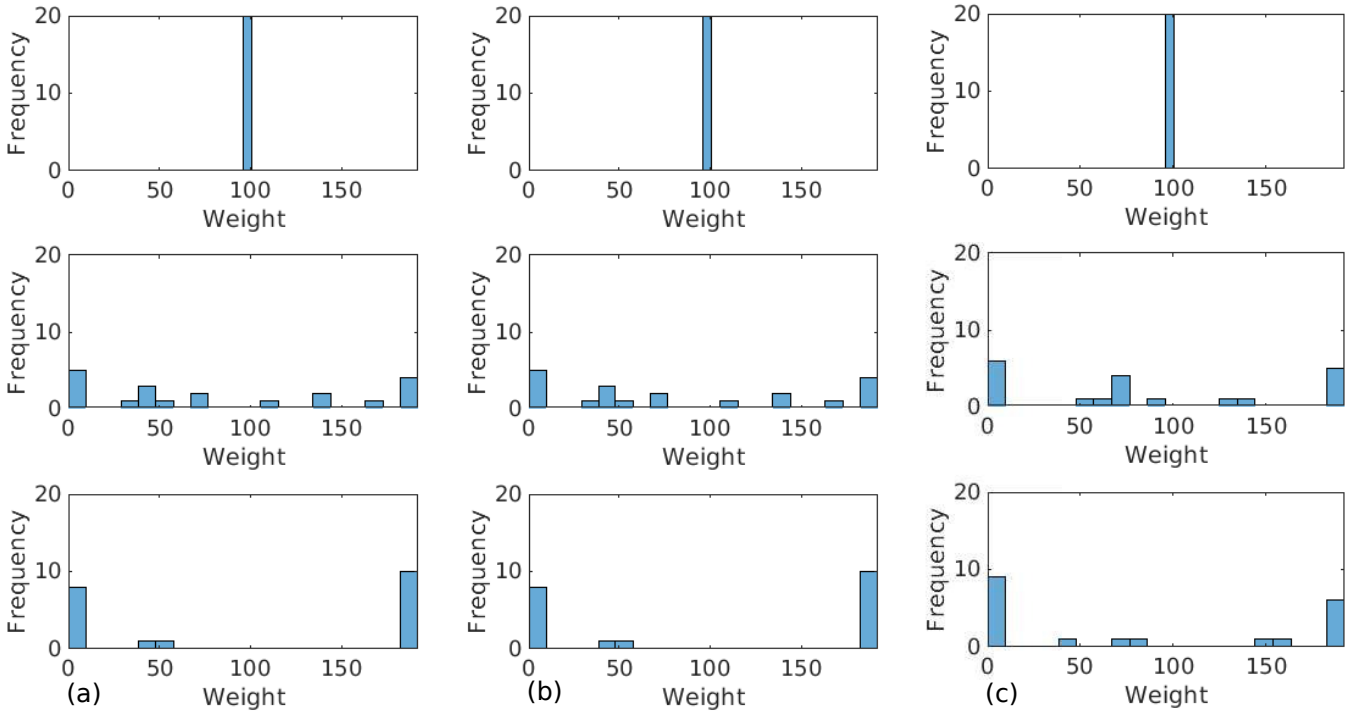


Fig. 9: Weight distribution after applying STDP rule in the network of (a) original, (b) CORDIC and (c) and 2^x based model

Proposed Solution

Error Analysis

- **Error analysis**
  - Normalized Root Mean Square Deviation (NRMSD)

$$RMSD = \sqrt{\frac{\sum_{k=1}^{n} (COR(x,y) - OR(x,y))^2}{n}}$$

$$NRMSD = \frac{RMSD}{(Max\ (OR(x,y)) - Min\ (OR(x,y))}$$

  - Maximum Deviation (MD)

$$MD = Max(|COR(x,y) - OR(x,y)|)$$

  - Error in timing (Errt)

$$Errt = \left| \frac{\Delta t_c - \Delta t_o}{\Delta t_o} \right| \times 100$$

Proposed Solution

Error Analysis

- **Error calculation for different CORDIC models**

	Model	Error Type	Ton. Spiking	Reg. Bursting
Computer simulation	IZHCOR6	Errt	%0.2549	%0.0000
		NRMSD	%0.0034	%0.0705
	IZHCOR8	Errt	%0.2049	%0.0000
		NRMSD	%0.0006	%0.0136
	IZHCOR10	Errt	%0.1025	%0.0000
		NRMSD	%0.0001	%0.0082
	IZHCOR12	Errt	%0.0000	%0.0000
		NRMSD	%0.0000	%0.0063
FPGA implementation	IZHCOR6	Errt	%0.0191	%0.0000
		NRMSD	%0.3951	%2.0631

Table 1 : ERRt and NRMSD for tonic spiking and regular bursting

Proposed Solution

Hardware Implementation

- **Izhikevich neuron hardware**

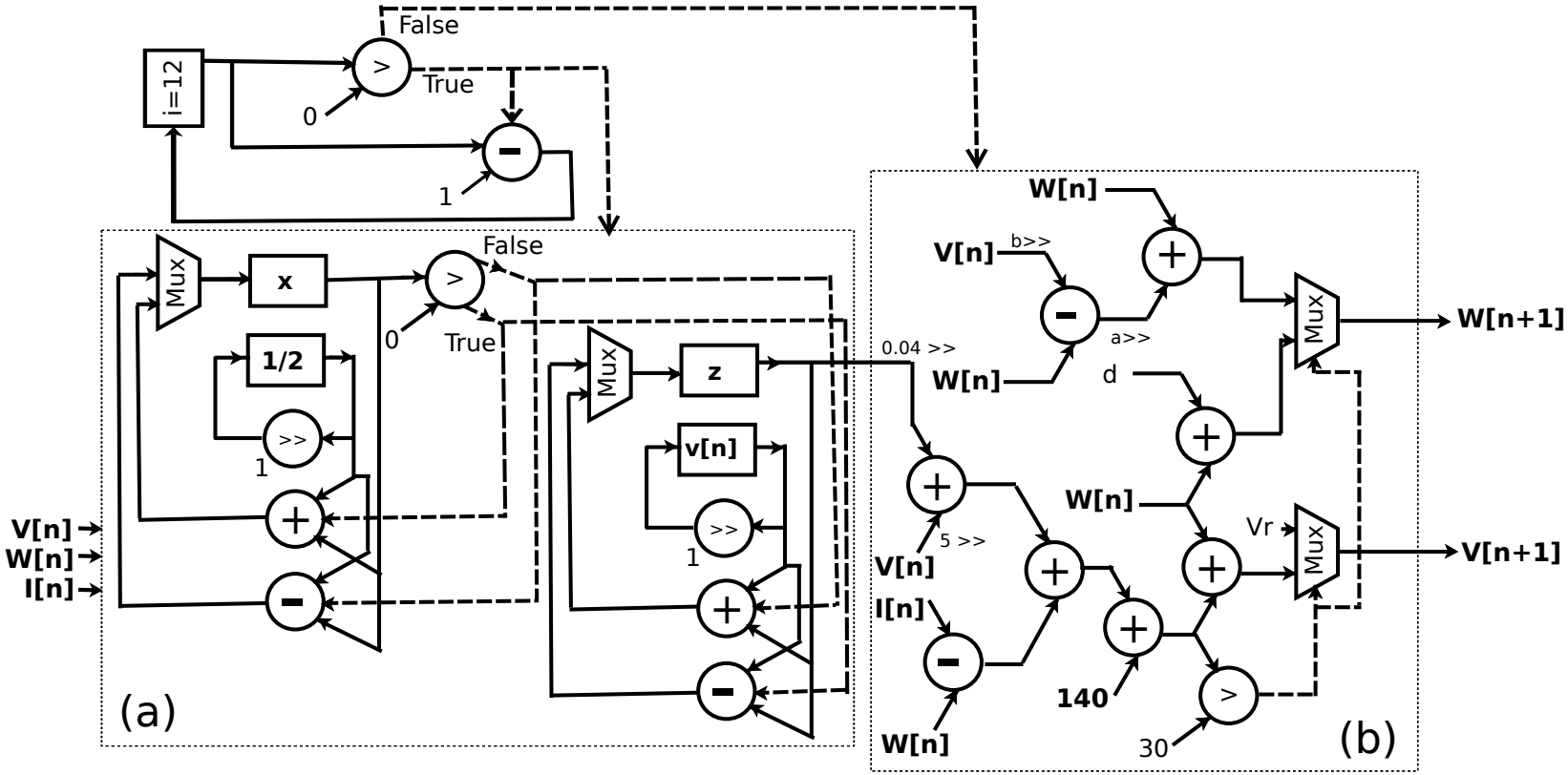


Fig. 10: Control data flow graph for FPGA implementation of CORDIC Izhikevich neuron.

Proposed Solution

Hardware Implementation

- **CORDIC exponential hardware**

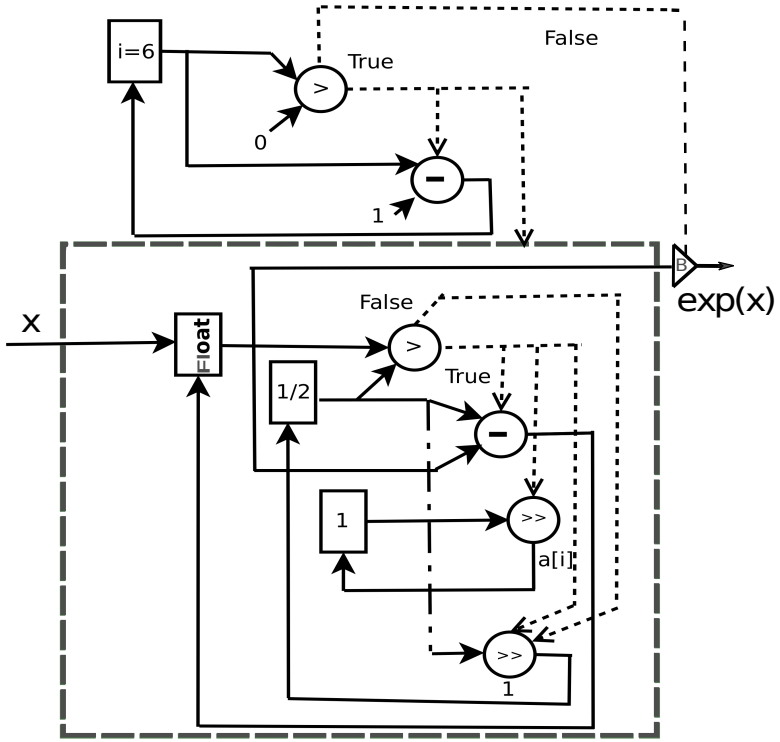


Fig. 11: Control data flow graph for FPGA implementation of CORDIC exponential calculator

Proposed Solution

Hardware Implementation

- **Neural network and STDP hardware**

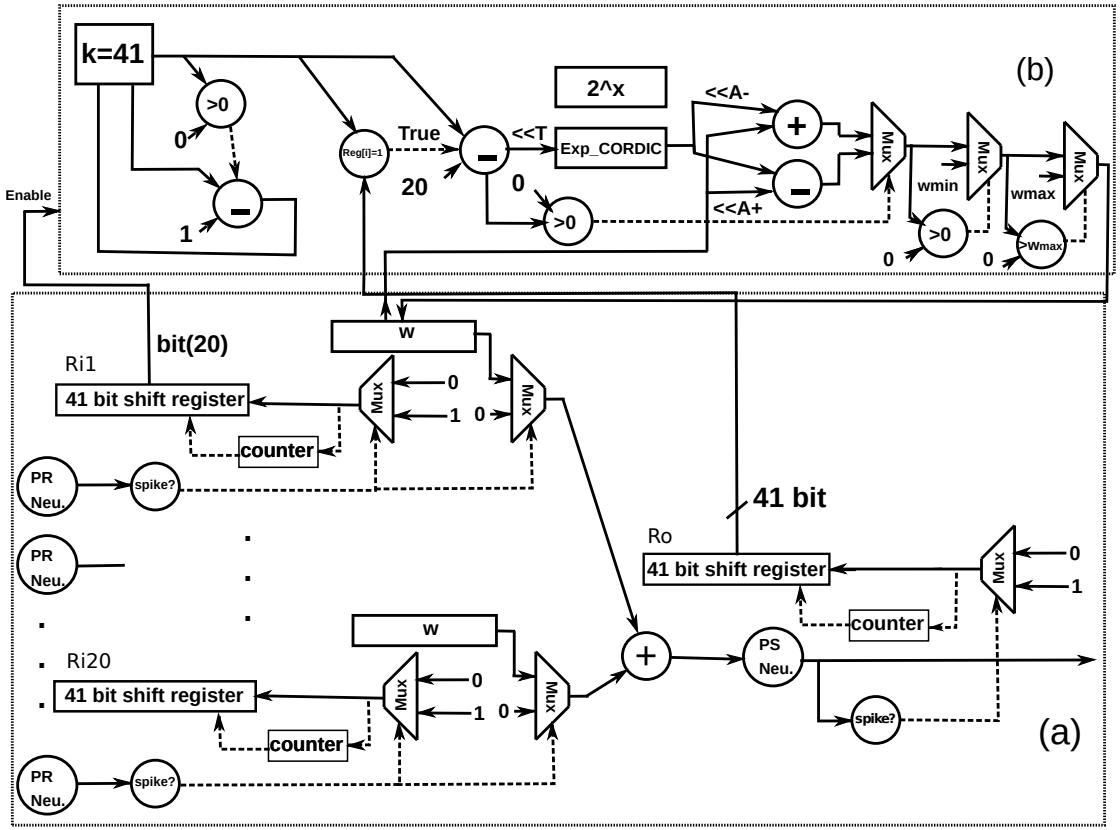
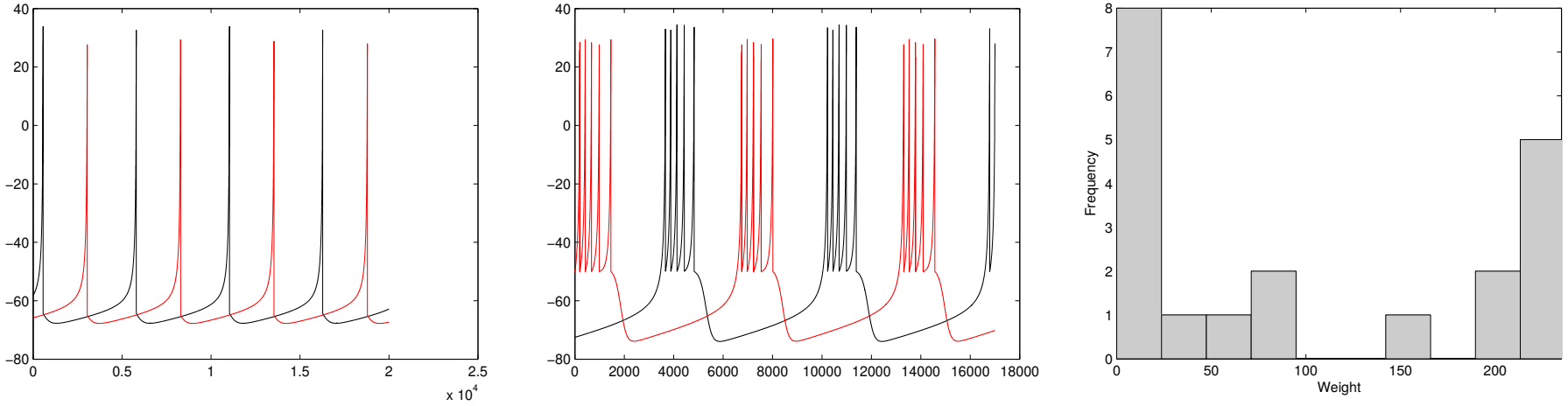


Fig. 12: Control data flow graph for digital implementation of STDP algorithm.

Results and Discussion

Implementation Results

- **On-FPGA neuron and STDP**



(A) (B) (C)

Fig. 13: (A), (B): FPGA Implementation of CORDIC modified Izhikevich (red line) and computer simulation of Izhikevich model (black line). (C) Weight distribution after execution of the online on-FPGA STDP on the network of Izhikevich neuron.

Results and Discussion

Implementation Results

- **Izhikevich neuron area and frequency**

Device	CORDIC		DSP Multiplier	
	Number	Speed	Number	Speed
Spartan-6 XC6LX75	105	183 MHz	33	44 MHz
Virtex-5 XC5VTX240T	240	220 MHz	145	102 MHz
Virtex-6 XC6VLX550T	750	332 MHz	120	111 MHz
Virtex-7 XC7VX980T	1280	370 MHz	540	130 MHz

Table 2: Number of the CORDIC and original Izhikevich neuron (using a DSP 36 bit multiplier) that can implement on some of FPGAs

- **Neural network and STDP area and frequency**

	Slice Registers	Utilization Perc.	Slice LUT's	Utilization Perc.	Max Speed (MHz)
CORDIC STDP	7,088	7%	10,376	22%	84.1
2^x STDP	7,047	7%	10,234	21%	84.5

Table 3: Total FPGA utilization for implementation of CORDIC and 2^x online STDP on a network of CORDIC Izhikevich neurons

Results and Discussion

Comparisson

- **Resource utilization and speed comparison**

Refrence	Slice Registers	Slice LUT's	Max Speed (MHz)	DSPs%	NRMSD%	Errt%	Device
Soleimani et al [3].	493	617	241.9	0	-	1.54	Virtex-II Pro XC2VP30
Gomar et al. [4]	388	1279	190	0	4.02	-	Virtex-II Pro XC2VP30
Hayati et al. [5]	476	856	135	0	3.7	-	Virtex-II Pro XC2VP30
Grassia et al. [6]	646	1048	105	22	-	-	Virtex-5 XC5VLX50
Heidarpur et al. [7]	829	1221	134.3	0	0.04	0.39	Spartan-6 XC6SLX9
Shimada et al. [8]	357	1776	Asynchronous	-	-	-	Zync-7000 XC7Z020
IZHCOR6 -Area optimization	229	410	183.4	0	0.003	0.26	Spartan-6 XC6SLX75
IZHCOR6 -Speed optimization	280	469	212.8	0	0.003	0.26	Spartan-6 XC6SLX75

Table 4: Comparison between proposed method and previously published works.

Conclusion

- **Conclusion**
  - Hardware were presented based on the CORDIC method to implement Izhikevich neuron and on-FPGA STDP with Izhikevich neuron
  - Design accuracy verified through simulation and various error analysis. Results confirmed the model has considerably lower error compared with previous works.
  - Hardware were designed and implemented on FPGA
  - Comparing with previous works, the proposed neuromorphic system offers better performance including lower power cosumtion and higher speed. Furthermore, this design requires less FPGA resources including slice registers and LUTs.

Publication

Publications

[1] M. Heidarpur, A. Ahmadi, M. Ahmadi, and M. Rahimi Azghadi, "Cordic-snn: On-fpga stdp learning with izhikevich neurons," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 7, pp. 2651–2661, July 2019.

References

References

[1] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on neural networks*, vol. 14, no. 6, pp. 1569–1572, 2003.

[2] A. Morrison, M. Diesmann, and W. Gerstner, "Phenomenological models of synaptic plasticity based on spike timing," *Biological cybernetics*, vol. 98, no. 6, pp. 459–478, 2008.

[3] H. Soleimani and E. M. Drakakis, "An efficient and reconfigurable synchronous neuron model," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. PP, no. 99, pp. 1–1, 2017.

[4] S. Gomar and A. Ahmadi, "Digital multiplierless implementation of biological adaptive-exponential neuron model," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 61, no. 4, pp. 1206–1219, April 2014.

[5] M. Hayati, M. Nouri, S. Haghiri, and D. Abbott, "Digital multiplierless realization of two coupled biological morris-lecar neuron model," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 7, pp. 1805–1814, July 2015.

[6] F. Grassia, T. Levi, T. Kohno, and S. Saïghi, "Silicon neuron: digital hardware implementation of the quartic model," *Artif Life Robotics*, vol. 19, no. 3, pp. 215–219, 2014.

[7] M. Heidarpour, A. Ahmadi, and R. Rashidzadeh, "A cordic based digital hardware for adaptive exponential integrate and fire neuron," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 11, pp. 1986–1996, Nov 2016.

[8] N. Shimada and H. Torikai, "A novel asynchronous cellular automaton multicompartment neuron model," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 8, pp. 776–780, Aug 2015.

Thank You