34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Compact digital implementation of a quadratic integrate-and-fire neuron

Eric J. Basham, *Member, IEEE*, David. W. Parent, *Member, IEEE*

***Abstract*—A compact fixed-point digital implementation of a quadratic integrate-and-fire (QIF) neural model was developed. Equations were derived to determine the minimum number of bits the digital QIF model requires to represent all four states of the QIF model and control the switching threshold of the output voltage. In addition, the equations were used to minimize the size of the multiplier used for the non-linear squaring function, V^2. These design equations were used to develop test vectors that could unambiguously show all four states of a digital QIF model. The FPGA implementation of the QIF model was shown to be computationally efficient, requiring only two fixed-point adders and one fixed-point multiplier.**

I. Introduction

THEQuadratic Integrate-and-Fire (QIF) neural model is one of the most computationally efficient neural models that can exhibit true spiking behavior [1]. There are eight typical neurocomputational properties of biological spiking neurons, including tonic spiking, Class 1 and 2 excitability, spiking latency, input integration, threshold variability, and bistability that the QIF can exhibit [1]. The QIF model can be implemented as an analog circuit, where:

$$RC\frac{dV}{dt} = B + V^2 \quad (1)$$

Subject to:

$$V \text{ is set to } V_{reset} \text{ if } V > V_{peak} \quad (2)$$

The analog version of the QIF has advantages in direct biological interfacing and continuous input and output operation. With current technology, analog integrated circuit implementations of neuromorphic hardware may also have advantages in density and lower power operation [2]. However, when compared with digital implementations, analog circuits have an inherently higher cost of manufacture and are more challenging to implement in a full custom integrated circuit [3].

A digital QIF can leverage digital manufacturing processes and scale easily with technology node changes. This may be particularly important if an array of QIF neurons is implemented with the goal of high-speed computation [4–10] or even complex control systems [11–13]. A fixed point digital QIF consists of digital adders and multipliers or lookup tables, which are more area efficient than a floating point digital implementation [14-16], although care has to be taken to ensure accuracy. A fixed point digital QIF neural model implementation allows dynamic adjustment of the clock frequency to balance computational demands, and thus power vs. performance can be traded off in real time. Fixed point digital implementations also allow the implementation of pipeline and serial operation so that one QIF neuron may be time division multiplexed into many "virtual" neurons, or one neuron can be made to have a very small area by performing serial computation [17, 18].

Field Programmable Gate Arrays (FPGAs) have been used to investigate neural processors [14] for learning, threshold neural models (such as McCulloch-Pitts) [19], conductance-based neural models(such as Hindmarsh-Rose or Hodgkin-Huxley) [16, 20, 21], the leaky integrate-and-fire (LIF) neural model [17, 22–25], the QIF neural model [25–28], the Fitzhugh-Nagumo neural model [18], and the Izhikevich neural model [29, 30]. FPGAs not only allow rapid prototyping of neural models, but can also be used to investigate the relationship between fixed-point word length and synthesized area.

Although there have been many reports concerning the appropriate word length of the fixed-point operation of the neural models [16, 17, 19, 21–24, 26, 29, 31–34], word length design equations have not yet been presented. The reported bit widths range from 10 bits for an LIF model meant for applications in visual perception [24] to 32 bits for a Hodgkin-Huxley model intended for biointerfacing [16]. The smallest bit width reported for QIF is 16 bits [26]. The smallest reported bit width for an Izhikevich model (an extension of the QIF) is 24 bits [29].

Besides controlling the word length of a neuron to keep complexity low, look-up tables [18] or piecewise linear approximations [20] have been used to reduce the computational complexity of the required nonlinear functions (such as V^2 or V^3) that ensure a bistable neural response. Researchers have successfully used a 2-piecewise approximation to approximate V^2 in a QIF model [26-28].

In this work, a QIF neural model using nine bit vectors in 2's complement is demonstrated. The approach to reducing the complexity of the neuron model (selecting the word length and minimizing the multiplier width used for the V^2 function) was to develop and use design equations for the system.

This work supported in part by NSF grant award #ECCS-0964983 EAGER.
Eric J. Basham and David W. Parent are with the Electrical Engineering department, San Jose State University, One Washington Square, San Jose CA, 95192 USA (corresponding author e-mail: basham.eric@gmail.com).

The organization of the paper is as follows. First the architecture of the QIF model is presented. Then, system design equations for V_{peak} and V_{th} are derived followed by an overview of the four states of a QIF neural model and the design equations used to develop test vectors that are then verified in simulation. Results showing that a nine-bit QIF model can demonstrate all states are then shown in simulation (Excel, Verilog), along with the results from a QIF synthesized with an FPGA.

Fig. 1 shows a fixed point QIF implementation. The signals and blocks are defined in Table 1. The discrete system equation describing the QIF implementation of Fig. 1 is:

$$V_n = V_{(n-1)} + A\left(V^2{}_{(n-1)} + B_{(n-1)}\right) \quad (3)$$

The multiplier is denoted by X, addition by +, and delay (flip-flops) by Z^{-1}.

TABLEI
Definition of Terms.

Term	Definition
A	Gain of the integrator, unitless, input
B	Input to the circuit, which has two parts, the steady state value, and a dynamic value. (Volts)
V_{peak}	This is the value, which if V is greater than will cause V to be set to V_{reset}. (Volts)
V_{reset}	The value V is set to after a reset event. (Volts)
V_{th}	When V>V_{th}, the circuit will spike.
V	Output voltage of the QIF. (Volts)
rst	The signal that sets V to V_{reset}, and the other delay elements to 0, when V >V_{th}
clk	System clock signal

Not shown in Fig. 1 is the comparator used to detect if V>V_{peak}. For example, a nine bit (2's complement) wide system (eight data bits with the most significant bit (MSB) used as the sign bit), all registers (delay) would be nine bits wide. The adders would be nine bits wide with no carry in or carry out. (Overflow is prevented by proper selection of V_{peak}.) Only the least significant bits of V are fed into the multiplier. For a nine bit (2's complement) system, this would be the lower 4 bits. A reset event would occur if any of the upper significant bits were equal to one, and the MSB were equal to zero. All of the data bits are integers, given that the data can be shifted before and after entering the system to account for data to the right of the decimal point. The model could include fractional data by appropriately shifting the data output by the multiplier (V^2).

II. System Design for a QIF Neural Model

Design issues that must be dealt with when designing a fixed point QIF neuron are minimizing the size of vectors and preventing overflow of the multiplier.

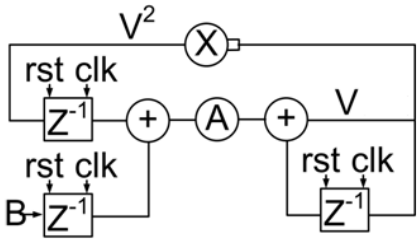

Fig. 1. Signal flow graph of QIF.

For example, a multiplier with 8 bit wide data inputs, outputs a vector that is 16 bits wide (plus the sign bit). Since the multiplier is fed back into the system, this means that V^2 could cause the system to overflow. This problem is solved by choosing V_{peak} (the reset condition) such that V^2 will not cause an overflow. To determine the maximum V_{peak} for the number of bits in a system (m) use the flowing equation, neglecting the sign bit:

$$V_{max}^2 = 2^m - 1 \quad (4)$$

that results in:

$$V_{peak} = V_{max} = \sqrt{2^m - 1} \quad (5)$$

For an eight bit width (ignoring the extra sign bit) system, V_{peak} would be set to 15. This can be done by monitoring the 4 most significant bits of V, and setting registers that receive V^2 and B to zero and the register that receives V to V_{reset}, when at least one of these bits is set to 1 and the MSB is equal to 0.

Another important design parameter for the QIF circuit is V_{th}. If V_n is greater than or equal to V_{th} the output voltage V_n will spike, even if the input is brought back to less than 0V. An expression for a digital QIF model's V_{th} can be found. For the system to integrate the following condition must be met:

$$\left({V_{n-1}}^2 + B_{n-1}\right) \times A \geq 2^{-P} \quad (6)$$

Where P is the precision of the fixed point system and is the number of digits to the right of the "binary" point. For example, if P was set equal 1, the precision would be 0.1.

If the system is at the switching threshold then:

$$V_{n-1} = V_{th} \quad (7)$$

Substituting equation7 into equation 6:

$$\left({V_{th}}^2 + B_{n-1}\right) \times A = 2^{-P} \quad (8)$$

Assuming the input B_n is set to a number less than or equal to zero:

$$V_{th} = \sqrt{\frac{2^{-P} - A \times B_{n-1}}{A}} \quad (9)$$

For this work the precision and B_n were set to zero and thus equation 9 reduces to:

$$V_{th} = \sqrt{\frac{1}{A}} \quad (10)$$

The input gain A, can be implemented with simple right shifting (with respect to the decimal point) of the sum of $V^2_n$and B_n, which results in dividing by powers of two. V_{th} as a function of Gain can be seen in table 2.

Verification of the designed V_{th} is easily accomplished by graphing equation 3 (subject to the reset condition) for values of B_n that generate a single spike. Specifically, set B_n constant until a spike is observed and then set the following B_n's to zero, and set V_{reset} to below V_{th}. A model that uses the dec2bin, bin2dec, and some hand coded 2's complement data functions to simulate a fixed point system was developed in Excel. An example is plotted in Fig. 2. For values of n from 0 to 8, integration occurs, and then the input is set to V_{reset}, at n=9. In this case, V_{reset} is three, and since $V_{reset}<V_{th}$ no further spiking occurs.

For further proof that V_{th} has been designed properly one may use the same test vector shown in Fig. 2, but instead change $V_{reset}=V_{th}$. The result is plotted in Fig. 3. For values of n from 0 to 8, integration occurs, and then the input is set to zero at n=9. In this case, V_{reset} is four and since $V_{reset}=V_{th}$ spiking occurs regardless of B_n. Figs. 2 and 3, show that the circuit will not spike without input if V_n is less than V_{th}, and it will spike without input if V_n is equal to V_{th} (instability). In this fixed point system, there are no fractional numbers and as a result all V_{th}'s are rounded to the next whole number as shown in Table 2.An engineer can easily design test vectors using the equation for V_{th} that demonstrate monostable or bistable behaviors by simply setting V_{reset} to below or above V_{th}.

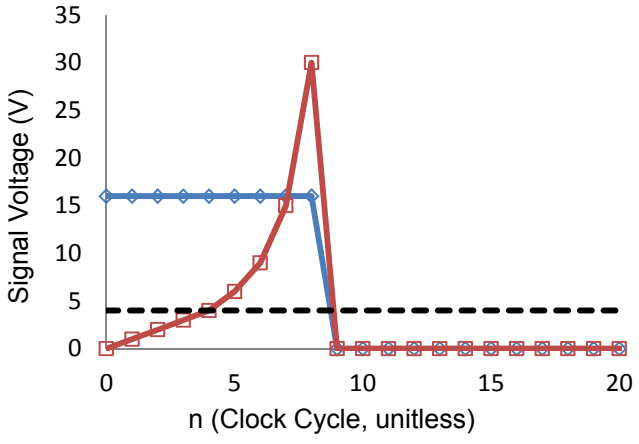

Fig. . Digital QIF Excel model results showing integration. A=0.0625, V_{peak}=15V, V_{reset}=3V, V_{th}=4V. The signal V_n is denoted by (□), and the signal B_n is denoted by (◊) .V_{th} is shown as a black dashed line (-).

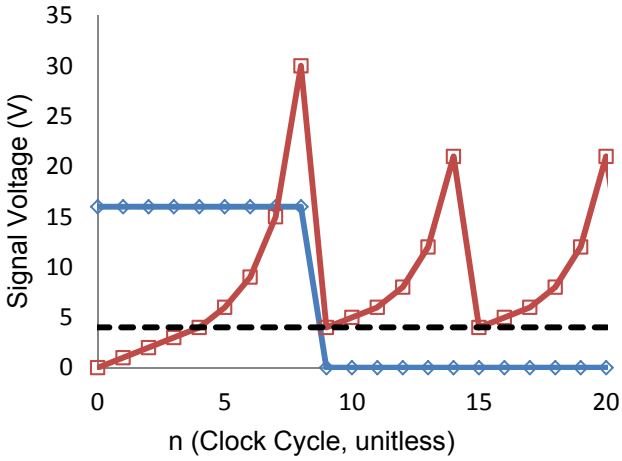

Fig. 2. Digital QIF Excel model results showing bistability. A=0.0625, V_{peak}=15V, $V_{reset}=V_{th}$, V_{th}=4V. The signal V_n is denoted by (□), and the signal B_n is denoted by (◊). V_{th} is shown as a black dashed line (-).

TABLE II
V_{th} as a function of Gain (B_n reset to 0V).

Right shift	A	Calculated V_{th}	Observed V_{th}
0	1	1	1
1	0.5	1.41	2
2	0.25	2	2
3	0.125	2.82	3
4	0.0625	4	4

III. Verification of the Two States of a Digital QIF Circuit

Although there are eight neural behaviors that a QIF can exhibit, for testing/verification purposes, a more thorough approach (based on knowing V_{th}) is to evaluate the fundamental neurocomputational properties embodied by a dynamical system. There are three characteristic behaviors that a QIF model can exhibit [35]:

1. Integrator
2. Bistable
3. Monostable

The integrator behavior of a QIF is characterized by the output V, increasing when the input B, is greater than zero. The bistable behavior of a QIF occurs when the V_{reset} is greater than V_{th} and the monostable state occurs when V_{reset} is less than V_{th}.The bistable and monostable behaviors are mutually exclusive. This leads to two possible states:

1. Bistable, integrator (steady state of B_n is greater than zero Volts, $V_{reset}>V_{th}$)
2. Monostable integrator (steady state of B_n is greater than zero Volts, $V_{reset}<V_{th}$)

These two states can be easily observed based on the above conditions and the fact that V_{th} can be calculated from the gain parameter A (eq. 10) and are shown in Figs. 4-7.

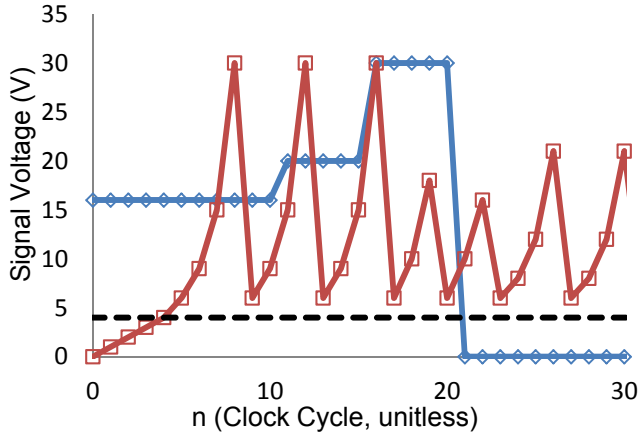

Fig. 3. Digital QIF excel model results showing the bistable, integrator state. A=0.0625, V_{peak}=15V, V_{reset}=6V, V_{th}=4V. The signal V_n is denoted by (□), and the signal B_n is denoted by (◊). V_{th} is shown as a black dashed line (-).

The QIF operating in the bistable-integrator state can be seen in Fig. 4. The circuit oscillates at relatively the same frequency for all positive values of B, once the circuit starts to spike. For B values of 1, 16, and 20, the spiking rate is one spike per fours clock cycles. For a B value of 30, the spike rate is one spike per three clock cycles. The bistable behavior can be seen because from clock cycle zero to three there is no spiking when B is set to zero Volts, but the circuit still spikes from clock cycles 30 to 40 even though the input is reset to zero volts because V is always above V_{th}.

The monostable-integrator state is demonstrated in Fig. 5. From clock cycle 0 to 30, the spiking frequency increases with increasing B values. The spiking rates for constant B values of 16, 20, 30, and 40 is 1 spike per 9 clock cycles, 1 spike per 9 clock cycles,1 spike per 7 clock cycles,1 spike per 6 clock cycles, respectively. The circuit stops spiking at clock cycle 31 when the input is set to 0V, which demonstrates monostable behavior.

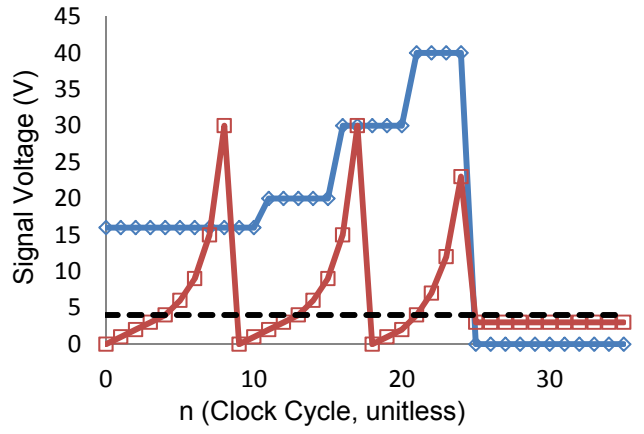

Fig. 4. Digital QIF excel model results showing the monostable, integrator state. A=0.0625, V_{peak}=15V, V_{reset}=0V, V_{th}=4V. The signal V_n is denoted by (□), and the signal B_n is denoted by (◊) .V_{th} is shown as a black dashed line (-).

IV. FPGA Verification of Design Equations

The QIF model was synthesized with ISE Design Suite 11 (32bit) into a Xilinx 100K Spartan 3E FPGA. The power consumption of the QIF model and the logic used to interface with the QIF was 34mW at a clock frequency of 12MHz. The efficiency of the digital QIF model can be seen Table 2.

The chip was tested and controlled with a USB interface with a Demand Peripherals Baseboard-4 development kit (http://www.demandperipherals.com/baseboard4.html).The Baseboard-4 development kit was selected because it is an inexpensive prototyping environment that can directly control motors used in robotic applications. A 1672G Standalone Logic Analyzer was used to generate the test vectors, and a Logicport 34 channel logic analyzer was used to capture the output vector.

TABLE III
Device Utilization Summary of Digital QIF Implemented in a FPGA.

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	41	1,920	2%
Number of 4 input LUTs	85	1,920	4%
Number of occupied Slices	59	960	6%
Number of Slices containing only related logic	59	59	100%
Number of Slices containing unrelated logic	0	59	0%
Total Number of 4 input LUTs	86	1,920	4%
Number used as logic	85		
Number used as a route-thru	1		
Number of bonded IOBs	52	66	78%
Number of BUFGMUXs	1	24	4%
Number of MULT18X18SIOs	1	4	25%
Average Fanout of Non-Clock Nets	2.33		

Fig. 6 shows spiking being turned off as the input is set to -30 at n=104. The input vectors where chosen to verify the FPGA implementation of the digital QIF model, because these vectors required special consideration of 2's complement in the Excel worksheet. Fig. 7 shows that the QIF model implemented in Verilog gives the same results as the excel model used to verify the V_{th}, design equation. Fig. 7 was generated from the raw measured data from the FPGA shown in Fig. 8. The timing data was replaced with clock cycle N, to more easily compare the Excel QIF model and the FPGA QIF implementation.

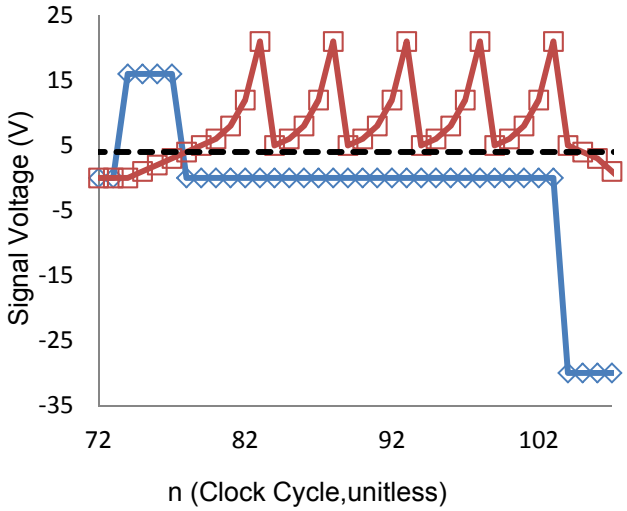

Fig.6. Digital QIF Excel model results showing the spiking of the bistable, integrator state being started and stopped with input vector B. A=0.0625, V_{peak}=15V, V_{reset}=5V, V_{th}=4V. The signal V_n is denoted by (□), and the signal B_n is denoted by (◊). V_{th} is shown as a black dashed line (-).

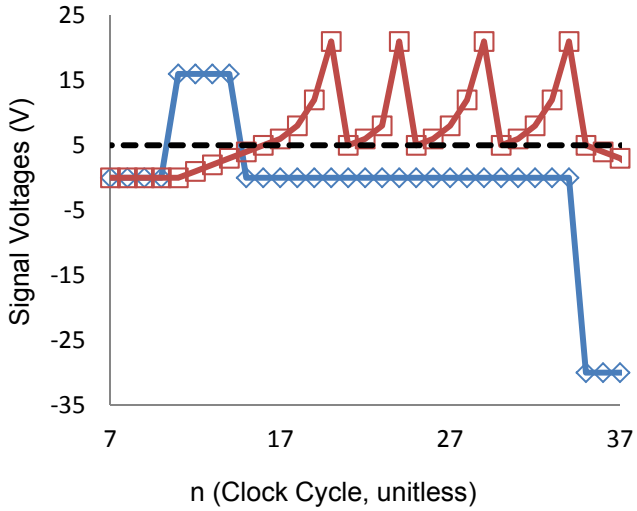

Fig.7. Digital QIF Verilog model results showing the spiking of the bistable, integrator state being started and stopped with input vector B. A=0.0625, V_{peak}=15V, V_{reset}=5V, V_{th}=4V. The signal V_n is denoted by (□), and the signal B_n is denoted by (◊). V_{th} is shown as a black dashed line (-).

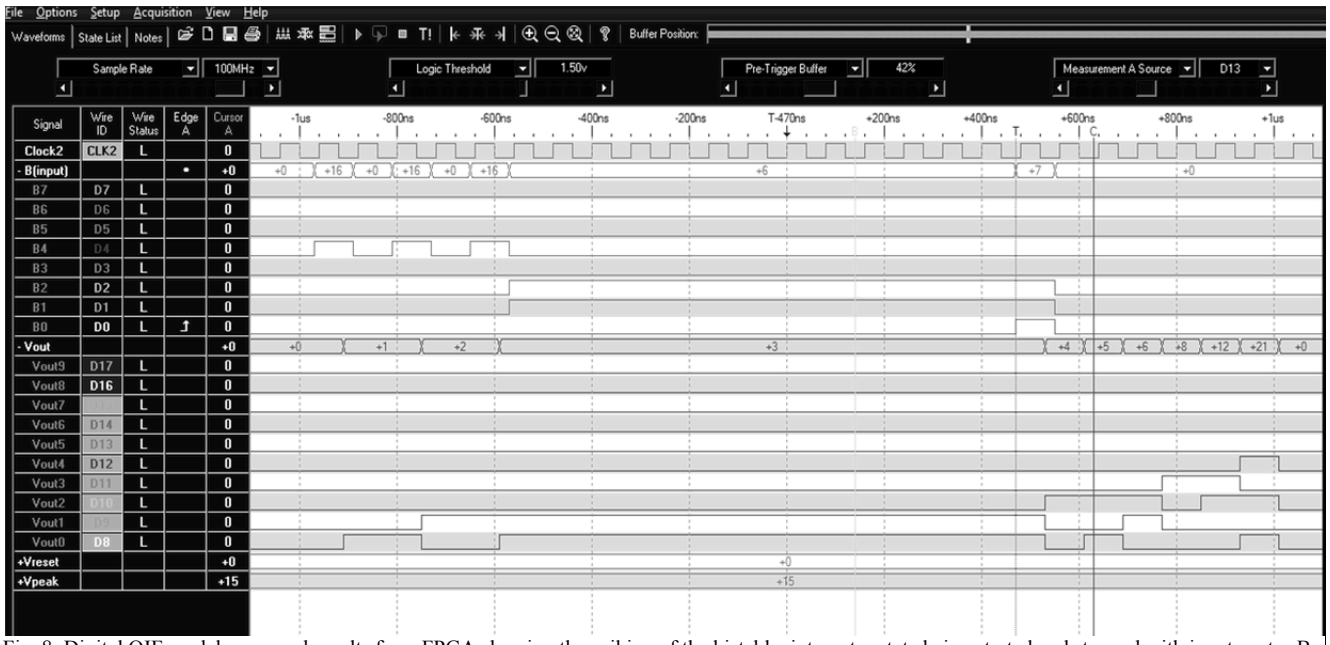

Fig. 8: Digital QIF model measured results from FPGA showing the spiking of the bistable, integrator state being started and stopped with input vector B. A=0.0625, V_{peak}=15V, V_{reset}=5V, V_{th}=4V.

Both states were also implemented for gains of 1, 0.25, and 0.125 (not shown). Even though the system did not have a large dynamic range for A=1, only nine bits were required to demonstrate all four states. Since the gain A, is just another input vector, it can be used to mimic the slow variable of the Izhikevich model or act as an adaptive controller that can self-learn a control signal routine.

V. Conclusions

Design equations for a fixed point digital implementation of a quadratic integrate-and-fire neural model were developed and verified in Excel and Verilog. dynamical computational states were demonstrated in a FPGA. To show all four states of a QIF model, only a 9 bit word length was required. This minimized the multiplier to an input width of four bits. Future studies include using this system as a reset controller in a robotic system, an ASIC implementation, and using the model to study networks of QIF neural models.

Acknowledgments

This work was supported in part by NSF grant award #ECCS-0964983 EAGER. We would like to thank Dr. David Tauck for useful discussions on neural behavior.

REFERENCES

[1] E. Izhikevich, "Which model to use for cortical spiking neurons?," *IEEE Transactions on Neural Networks,* vol. 15, pp. 1063-1070, 2004.

[2] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. Van Schaik, R. Etienne-Cummings, T. Delbruck, S. C. Liu, P. Dudek, P. Häfliger, and S. Renaud, "Neuromorphic silicon neuron circuits," Frontiers in neuroscience, vol. 5, 2011.

[3] D. Hammerstrom, "Digital VLSI for neural networks," in *The handbook of brain theory and neural networks*: The MIT Press, 1995, p. 304.

[4] D. Mishra, A. Yadav, and P. Kalra, "Learning with Single Quadratic Integrate-and-Fire Neuron," *Advances in Neural Networks-ISNN 2006,* pp. 424-429, 2006 2006.

[5] M. Pearson, A. Pipe, B. Mitchinson, K. Gurney, C. Melhuish, I. Gilhespy, and M. Nibouche, "Implementing spiking neural networks for real-time signal-processing and control applications: a model-validated FPGA approach," *IEEE Transactions on Neural Networks,* vol. 18, pp. 1472-1487, 2007.

[6] D. Roggen, S. Hofmann, Y. Thoma, and D. Floreano, "Hardware spiking neural network with run-time reconfigurable connectivity in an autonomous robot," in *NASA/DoD Conf. on Evolvable Hardware*, 2003, pp. 189-198.

[7] J. Šíma and P. Orponen, "General-purpose computation with neural networks: A survey of complexity theoretic results," *Neural Computation,* vol. 15, pp. 2727-2778, 2003.

[8] L. Smith, "Implementing neural models in silicon," *Handbook of nature-inspired and innovative computing,* pp. 433-475, 2006.

[9] S. Wermter, M. Elshaw, J. Austin, and D. Willshaw, "Towards novel neuroscience-inspired computing," in *Emergent neural computational architectures based on neuroscience*. vol. 2036/2001, 2001, pp. 1-19.

[10] D. Xu, J. Principe, and J. Harris, "Logic computation using coupled neural oscillators," in *IEEE Int. Symp. Circ. Syst.* , 2004, pp. 788-791.

[11] O. Beker, C. Hollot, Y. Chait, and H. Han, "Fundamental properties of reset control systems," *Automatica,* vol. 40, pp. 905-915, 2004.

[12] E. Rietman, M. Tilden, and M. Askenazi, "Analog computation with rings of quasiperiodic oscillators: the microdynamics of cognition in living machines," *Robotics and Autonomous Systems,* vol. 45, pp. 249-263, 2003.

[13] F. Tenore, R. Etienne-Cummings, and M. Lewis, "A programmable array of silicon neurons for the control of legged locomotion," *ISCAS*, 2004.

[14] D. Anguita, A. Boni, and S. Ridella, "A digital architecture for support vector machines: theory, algorithm, and FPGA implementation," *IEEE Transactions on Neural Networks,* vol. 14, p. 993, 2003.

[15] A. W. Savich, M. Moussa, and S. Areibi, "The Impact of Arithmetic Representation on Implementing MLP-BP on FPGAs: A Study," *IEEE Transactions on Neural Networks,* vol. 18, pp. 240-252, 2007.

[16] Y. Zhang, J. Nunez-Yanez, J. McGeehan, E. Regan, and S. Kelly, "A biophysically accurate floating point somatic neuroprocessor,", *Field Programmable Logic and Applications*, pp. 26-31, 2009.

[17] B. Schrauwen and J. Van Campenhout, "Parallel hardware implementation of a broad class of spiking neurons using serial arithmetic," *Proceedings of ESANN*, pp. 623-628, 2006.

[18] R. Weinstein and R. Lee, "Architectures for high-performance FPGA implementations of neural models," *Journal of Neural Engineering,* vol. 3, p. 21, 2006.

[19] M. Bañuelos-Saucedo, J. Castillo-Hernández, S. Quintana-Thierry, R. Damián-Zamacona, J. Valeriano-Assem, R. Cervantes, R. Fuentes-González, G. Calva-Olmos, and J. Pérez-Silva, "Implementation of a neuron model using FPGAs," *Journal of Applied Research and Technology,* vol. 1, 2009.

[20] M. Storace, D. Linaro, and E. de Lange, "The Hindmarsh–Rose neuron model: Bifurcation analysis and piecewise-linear approximations," *Chaos: An Interdisciplinary Journal of Nonlinear Science,* vol. 18, p. 033128, 2008.

[21] T. Poggi, A. Sciutto, and M. Storace, "Piecewise linear implementation of nonlinear dynamical systems: from theory to practice," *Electronics letters,* vol. 45, p. 966, 2009.

[22] A. Upegui, C. Peña-Reyes, and E. Sanchez, "An FPGA platform for on-line topology exploration of spiking neural networks," *Microprocessors and microsystems,* vol. 29, pp. 211-223, 2005.

[23] A. Cassidy, S. Denham, P. Kanold, and A. Andreou, "FPGA based silicon spiking neural array," *Proceedings of the Biomedical Circuits and Systems,* pp. 75–78, 2007.

[24] B. Girau and C. Torres-Huitzil, "Massively distributed digital implementation of an integrate-and-fire LEGION network for visual scene segmentation," *Neurocomputing,* vol. 70, pp. 1186-1197, 2007.

[25] R. Vilela and B. Lindner, "Comparative study of different integrate-and-fire neurons: Spontaneous activity, dynamical response, and stimulus-induced correlation," *Physical Review E,* vol. 80, p. 31909, 2009.

[26] H. Shayani, P. Bentley, and A. Tyrrell, "An FPGA-based Model Suitable for Evolution and Development of Spiking Neural Networks,"*Proceedings of ESANN,*pp.197-202, 2008.

[27] H. Shayani, P. Bentley, and A. Tyrrell, "A Cellular Structure for Online Routing of Digital Spiking Neuron Axons and Dendrites on FPGAs," 2008, p. 284.

[28] H. Shayani, P. J. Bentley, and A. M. Tyrrell, "Hardware Implementation of a Bio-plausible Neuron Model for Evolution and Growth of Spiking Neural Networks on FPGA," in *Adaptive Hardware and Systems, 2008. AHS '08. NASA/ESA Conference on*, 2008, pp. 236-243.

[29] M. La Rosa, E. Caruso, L. Fortuna, M. Frasca, L. Occhipinti, and F. Rivoli, "Neuronal dynamics on FPGA: Izhikevich's model," 2005, p. 87.

[30] A. Cassidy and A. Andreou, "Dynamical digital silicon neurons,", *BioCAS*, pp. 289-292, 2008.

[31] J. Zhu and P. Sutton, "FPGA implementations of neural networks-a survey of a decade of progress,"*Field Programmable Logic and Application*, pp. 1062-1066, 2003.

[32] L. Maguire, T. McGinnity, B. Glackin, A. Ghani, A. Belatreche, and J. Harkin, "Challenges for large-scale implementations of spiking neural networks on FPGAs," *Neurocomputing,* vol. 71, pp. 13-29, 2007.

[33] R. Naud, N. Marcille, C. Clopath, and W. Gerstner, "Firing patterns in the adaptive exponential integrate-and-fire model," *Biological Cybernetics,* vol. 99, pp. 335-347, 2008.

[34] K. Rice, M. Bhuiyan, T. Taha, C. Vutsinas, and M. Smith, "FPGA Implementation of Izhikevich Spiking Neural Networks for Character Recognition," *International Conference on Reconfigurable Computing and FPGA*, pp. 451-456, 2009.

[35] E. Izhikevich, *Dynamical systems in neuroscience: The geometry of excitability and bursting*: The MIT press, 2007.