A log-domain implementation of the Mihalas-Niebur neuron model

André van Schaik, Craig Jin, Alistair McEwan
School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW 2006, Australia
andre.vanschaik@sydney.edu.au

Tara Julia Hamilton
School of Electrical Engineering and Telecommunications
University of New South Wales
Sydney, NSW 2052, Australia

Stefan Mihalas, Ernst Niebur
Zanvyl Krieger Mind/Brain Institute and Neuroscience Department,
Johns Hopkins University,
Baltimore, MD, USA

***Abstract*— We present an electronic neuron that uses first-order log-domain low-pass filters to implement the Mihalas-Niebur model. The neuron consists of a leaky-integrate-and-fire core and building blocks to implement an adaptive threshold and spike induced currents. Simulation results show that this modular neuron can emulate different spiking behaviours observed in biological neurons.**

I. Background

Mihalas and Niebur recently introduced a new spiking neuron model [1] that shows many of the spiking behaviours observed in real neurons. Unlike other simplifications of the Hodgkin-Huxley neuron model, such as the Fitzhugh-Nagumo, Morris-Lecar, and Izhikevich models, the Mihalas-Niebur (MN) model uses simple first-order differential equations to describe each of the state variables. All the complexity of the MN model derives from the reset rules that are applied when a spike is generated. A switched capacitor implementation of this model has been presented in [2, 3].

Of the previous neuron models, the Izhikevich (Iz) model [4] is the simplest and it has become quite popular for the simulation of spiking neural networks. The Iz model has recently been implemented very efficiently in silicon by Wijekoon and Dudek using only 14 MOS transistors [5, 6]. The MN model offers a number of advantages over the Iz model. Firstly, all the state variables in the MN model have a biophysical interpretation, which allows us to learn from the model what might be happening in biological neurons. Secondly, the MN model is modular with minimal interference between the state variables, and as such provides a systematic way of adding an arbitrary number of additional mechanisms and state variables. The equations of the MN model and of the Iz model, map quite naturally to a log-domain implementation. In this paper we present the log-domain implementation of the MN model, while in a companion paper, we present the log-domain implementation of the Iz model, using many of the same building blocks. We are not claiming in this paper that these neuron models are better than others, or that our implementation is necessarily better than others in the literature. Rather we are determining how well we can map the equations of the neuron model to log-domain circuits. Furthermore we want to compare the performance of the two neuron models as log-domain implementations, and to that end a test chip has been implemented containing 57 identical copies of each type. Here we first present the MN model in section II, followed by the circuit implementation in section III. In section IV we present the result of various simulations of the circuits and we conclude in section V.

II. The Mihalas-Niebur Neuron Model

The MN model is described by three equations:

$$\frac{dI_j}{dt} = -k_j I_j ; j = 1, \ldots, N \tag{1}$$

$$\frac{dV_{mem}}{dt} = \frac{1}{C_{mem}}\left(I_{ex} + \sum_j I_j - GV_{mem} \right) \tag{2}$$

$$\frac{d\theta}{dt} = (aV_{mem} - b\theta) \tag{3}$$

The first equation represents spiking related currents I_j of which any number can be added to the model to incorporate different behaviours and also to model different synaptic dynamics. The second equation represents the membrane potential V_{mem} (expressed relative to the membrane resting potential, V_{mem0}) and exhibits a typical leaky integration of the excitatory input current I_{ex} and the spiking related currents I_j. The third equation implements a slow adaptation of the spiking threshold θ (expressed relative to the resting threshold, θ_0) as a function of V_{mem}. Note that this threshold is not just updated when the neuron spikes, but rather continuously to model voltage-dependent currents. This is an important detail in the behaviour of the model – when the membrane potential remains just below threshold, i.e., when a neuron just doesn't spike, the highest threshold adaptation is obtained. When a neuron spikes, the average membrane voltage will be lower

978-1-4244-5309-2/10/$26.00 ©2010 IEEE

than in the previous case, due to the resetting after each spike, and the threshold will adapt less than in the previous case.

When $V_{mem} + V_{mem0}$ reaches $\theta + \theta_0$ the neuron spikes and the state variables are updated as follows:

$$I_j := R_j I_j + A_j \tag{4}$$

$$V_{mem} := V_r \tag{5}$$

$$\theta := \max(\theta, \theta_r) \tag{6}$$

where R_j, A_j, V_r, and θ_r are free parameters of the model. In [1], and in the cases studied here, R_j is chosen to be 0, leading to an update to a constant value A_j, or R_j is chosen to be 1, leading to an additive update. V_r is the reset potential of the membrane voltage and the threshold voltage θ is reset to θ_r only when θ is below θ_r when the neuron spikes.

Note that the spike related currents I_j are independent of all other state variables and only depend on the chosen parameters and the occurrence of a spike. The membrane potential V_{mem} only depends on its parameters and the input current I_{ex} plus the sum of the spike related currents. Finally, the threshold voltage θ depends only on its parameters and the membrane potential. This minimal interdependence simplifies adjusting the parameters to obtain the various spiking behaviours in the model.

III. CMOS Implementation

To facilitate implementing the MN model on chip, we first rewrite equations (1) – (3) as standard first-order low-pass filter equations in the Laplace domain as follows:

$$I_j = \frac{I_{DC}}{s\tau_j + 1}, \text{with } \tau_j = \frac{1}{k_j} \tag{7}$$

$$I_{mem} = \frac{I_{ex} + I_{mem\,0} + \sum_j I_j}{s\tau_{mem} + 1}, \text{with } \tau_{mem} = \frac{C_{mem}}{G} \tag{8}$$

$$I_\theta = \frac{A_\theta (I_{mem} - I_{mem\,0}) + I_{\theta 0}}{s\tau_\theta + 1}, \quad \text{with } \tau_\theta = \frac{1}{b} \text{and } A_\theta = \frac{a}{b} \tag{9}$$

Here we have expressed each state variable as a current to illustrate how the model can be implemented using log-domain filters. A log-domain implementation seems natural, since a proper resistive leak is needed in parallel with the membrane capacitance, so that I_{mem} can be charged to just below I_θ by an appropriately chosen DC excitatory current, yielding maximum adaptation of I_θ. In a voltage domain implementation, such a resistive leak is difficult to implement on-chip and often a constant current leak is used instead. However, for a DC excitatory current with a constant current leak, the membrane voltage would either remain at its resting potential or reach the threshold voltage after a time proportional to the difference in the excitatory current and the leakage current. Many of the different spiking behaviours of the MN neuron rely on the interplay between sub-threshold membrane charging and threshold adaptation that can only be obtained with a resistive leak.

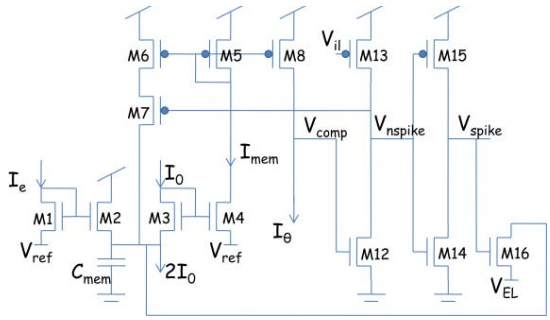

Figure 1. Log-domain Leaky-Integrate-and-Fire neuron.

Table 1. Typical values from [1].

τ_{mem} = 20 ms	τ_θ = 100 ms	A_θ = 0 or 1/2
τ_1 = 5 ms	R_1 = 0	A_1 = 0 or 10 [V/s]*C_{mem}
τ_2 = 50 ms	R_2 = 1	A_2 = 0 or -0.6 [V/s]*C_{mem}

Since log-domain filters do not operate well when the input current tends towards zero, an additional parameter I_{DC} has been introduced in (7) to set the resting level of each spike induced current. In our implementation $\sum_j I_{DC}$ cancels out.

When A_j and A_θ are set to zero, $\sum_j I_j = 0$ and $I_\theta = I_{\theta 0}$, the model reduces to a simple leaky-integrate-and-fire neuron, of which an implementation is shown in Figure 1. This implementation uses the tau-cell [7] as a first-order low-pass filter to model I_{mem}. The tau-cell core is implemented by NMOS transistors M1-M4, the capacitor C_{mem}, and the bias currents I_0 and $2I_0$. The transfer function of this block is:

$$I_{mem} = \frac{I_e}{s\tau_{mem} + 1}, \text{with } \tau_{mem} = \frac{U_T C_{mem}}{I_0} \tag{10}$$

where U_T is the thermal voltage. Thus, if we equate I_e with $I_{ex} + I_{mem\,0}$ we have implemented (8). A negative I_{ex} represents an inhibitory input current which would hyperpolarise the neuron. However, the circuit will only operate with $I_e > 0$, so that the maximum inhibitory current is limited by I_{mem0}.

In order to create a spike, I_{mem} is copied by PMOS transistors M5 and M8 and compared with the (constant) threshold current I_θ. Since I_{mem} can be arbitrary close to I_θ, a current limited inverter (M12, M13) is added to reduce power consumption while converting the result of the comparison into a digital value V_{nspike}. A positive voltage spike V_{spike} is generated with inverter M14, M15 with a slight delay with respect to V_{nspike}. PMOS transistors M5-M7 implement positive feedback based on V_{nspike} while NMOS transistor M16 resets I_{mem} to a value determined by V_{EL}, which implements the current domain version of equation (5). This reset causes the end of the positive feedback and the end of the spike and the membrane is ready to start the next integration cycle.

When A_θ is not zero, the threshold will adapt according to equation (9). This adaptation can be implemented by inserting a second tau-cell core (see Figure 2) whose transfer function of is given by:

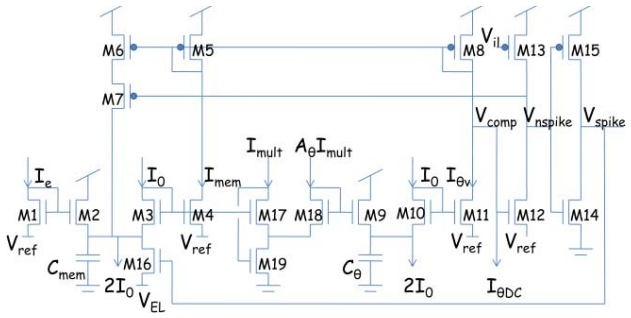

Figure 2. Leaky-Integrate-and-Fire neuron with threshold adaptation.

$$I_{\theta v} = \frac{A_\theta I_{mem}}{s\tau_\theta + 1}, \text{ with } \tau_\theta = \frac{U_T C_\theta}{I_0} \tag{11}$$

A DC current $I_{\theta DC} = I_{\theta 0} - A_\theta I_{mem\,0}$ is added to (11) to implement (9). The model needs $I_{\theta 0} > I_{mem\,0}$, i.e., the resting threshold needs to be larger than the resting potential of the membrane, so that $I_{\theta DC} > (1 - A_\theta) I_{mem\,0}$.

The gain $A_\theta = ½$ in (9) may be implemented by making M11 twice as long as M1-M4 and M9-M10 and connecting the gate of M3-M4 to that of M9 directly. This approach is simple and reliable, but doesn't allow A_θ to be changed post fabrication. Instead, for flexibility, we have implemented a variable gain using a five transistor translinear multiplier implemented with M3, M17-M19, and M9.

For simplicity (11) assumes that both tau-cells use the same I_0 and the ratio of the time constants is set by C_θ/C_{mem}. However, this need not be the case and, since τ_θ is typically five times larger than τ_{mem} (see Table 1), use of a smaller I_0 in (11) will lead to area savings by reducing the size of C_θ.

The circuits of Figure 1 and Figure 2 can create most of the non-bursting spiking behaviours shown in [1, 4]. However, we have not implemented equation (6), which resets the threshold to θ_r when a spike occurs and $\theta < \theta_r$. This case only happens when the threshold has first been lowered by a prolonged hyperpolarisation and we currently do not envisage needing this behaviour. If the need arises, a resetting mechanism can be added to the tau-cell implementing the threshold adaptation, similar to the reset of I_{mem}.

If bursting spiking behaviour is needed, then spike induced currents need to be added. A fast excitatory spike induced current together with a slower inhibitory spike induced current will cause bursting since at first the positive feedback is stronger, but eventually the inhibitory feedback will dominate and will inhibit the neuron for some time after the burst. The circuit to implement the excitatory spike induced currents is shown in Figure 3. In this circuit the capacitor is set to a fixed voltage after every spike, leading to a constant update. The circuit of Figure 4 implements the inhibitory spike induced current in which a constant current is added to the value of I_{k2} when the neuron spikes, implementing an additive update. Both circuits have a tau-cell at their core.

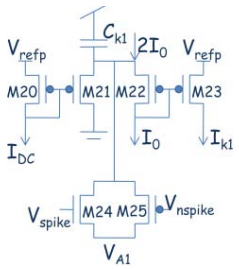

Figure 3. Excitatory spike induced current with constant update.

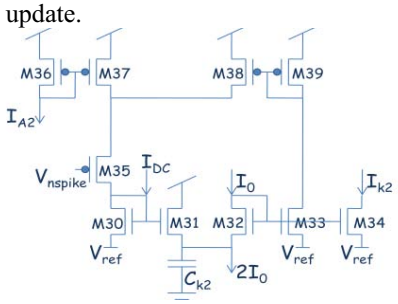

Figure 4. Inhibitory spike induced current with additive update.

IV. Simulation Results

We have simulated the proposed circuits using the parameters for the AMIS 0.5 µm process. All transistors are 2.8 µm wide and 4.2 µm long, except for the inverters M12-M15 which are 1.4 µm wide and 0.7 µm long and M16, which is 5.6 µm wide and 1.4 µm long. The capacitor values are: C_{mem} = 0.8 pF, C_θ = 2 pF, C_{k1} = 0.2 pF, and C_{k2} = 1 pF. I_0 = 12 pA for I_{mem} and I_{k1} and 3 pA for the slow variables I_θ and I_{k2}. The total size of the neuron is 0.04 mm^2.

In the simulations, the resting level of the membrane current was approximately 60 nA and the resting threshold current was 50 nA above that. V_{dd} was 3.3V, and the power consumption at rest was 4 µW. A single spike lasting 100 µs consumes approximately 70 µW. V_{EL} was 0.55 V, and the reference potential for the NMOS and PMOS tau-cells were 0.4 and 1.0 V from either rail, respectively. These reference potentials are needed so that the $2I_0$ currents can be provided by transistors operating in saturation. The spike induced currents were controlled with V_{A1} = 1.95 V and I_{A2} = 1 µA.

Figure 5 shows the simulation results. In Figure 5a the circuit was configured to be a simple leaky-integrate-and-fire neuron with no threshold adaptation (Figure 1) and the input current was only just enough to cause the neuron to spike. This shows clearly that the membrane current can be very close to the spiking threshold for extended periods, which would make the neuron very sensitive to additional spike inputs. Note that for clarity we have omitted the digital spike output from the plots and spikes are indicated by the abrupt reset of I_{mem}. Figure 5b-d show the results of simulating the neuron with threshold adaptation (Figure 2) but without spike induced currents. Figure 5b shows the spike frequency adaptation caused by the slow increase in threshold as the neuron spikes. Phasic spiking is obtained when the threshold adapts after initial spiking to a level where the excitatory input

current is no longer enough to cause the neuron to spike. The membrane potential will remain just below the threshold for as long as the constant input current remains. Figure 5d shows accommodation in the neuron. The input current at the start is the same as in Figure 5b, which causes the neuron to spike. However, when the input current is raised to this level in three equal steps, as shown in the 60 ms to 105 ms range in Figure 5d, the neuron's threshold accommodates to each increase in input current and the neuron won't spike, even when the input current reaches its original level. Figure 5e and Figure 5f show two types of bursting behaviour obtained when the circuits of Figure 4 are included, for different levels of excitatory current. A larger input current causes Tonic Bursting (Figure 5e) while a smaller input current causes Phasic Bursting (Figure 5f).

V. Conclusions

We have presented an implementation the Mihalas-Niebur neuron using a number of building blocks constructed around a first-order log-domain low-pass filter core. The neuron can be constructed from these building blocks to exhibit a simple leaky-integrate-and-fire behaviour, with or without threshold adaptation and spike induced currents can be added to model more complex neurons, such as those exhibiting bursting.

Acknowledgments

The collaboration on this design was started at the Telluride Neuromorphic Engineering Workshop 2008. The authors thank the organisers and the sponsors of the workshop for their support.

References

[1] S. Mihalas and N. Niebur, "A Generalized Linear Integrate-And-Fire Neural Model Produces Diverse Spiking Behaviors," *Neural Computation,* vol. 21, pp. 704-718, 2009.

[2] F. Folowosele, T. J. Hamilton, A. Harrison, S. Mihalas, E. Niebur, A. Cassidy, A. Andreou, and R. Etienne-Cummings, "A Switched Capacitor Implementation of the Generalized Linear Integrate-And-Fire Neuron," in *IEEE International Symposium on Circuits and Systems* Taiwan, 2009.

[3] F. Folowosele, R. Etienne-Cummings, and T. J. Hamilton, "A Switched Capacitor Implementation of the Mihalas-Niebur Neuron," in *IEEE Biomedical Circuits and Systems Conference* Beijing, China, 2009.

[4] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on Neural Networks,* vol. 14, pp. 1569-1572, Nov 2003.

[5] J. H. B. Wijekoon and P. Dudek, "Integrated circuit implementation of a cortical neuron," in *IEEE International Symposium on Circuits and Systems*, 2008, pp. 1784-1787.

[6] J. H. B. Wijekoon and P. Dudek, "Compact silicon neuron circuit with spiking and bursting behaviour," *Neural Networks,* vol. 21, pp. 524-534, Mar-Apr 2008.

[7] A. van Schaik and C. Jin, "The tau-cell: a new method for the implementation of arbitrary differential equations," in *IEEE International Symposium on Circuits and Systems - ISCAS 2003*, 2003, pp. 569 - 572.

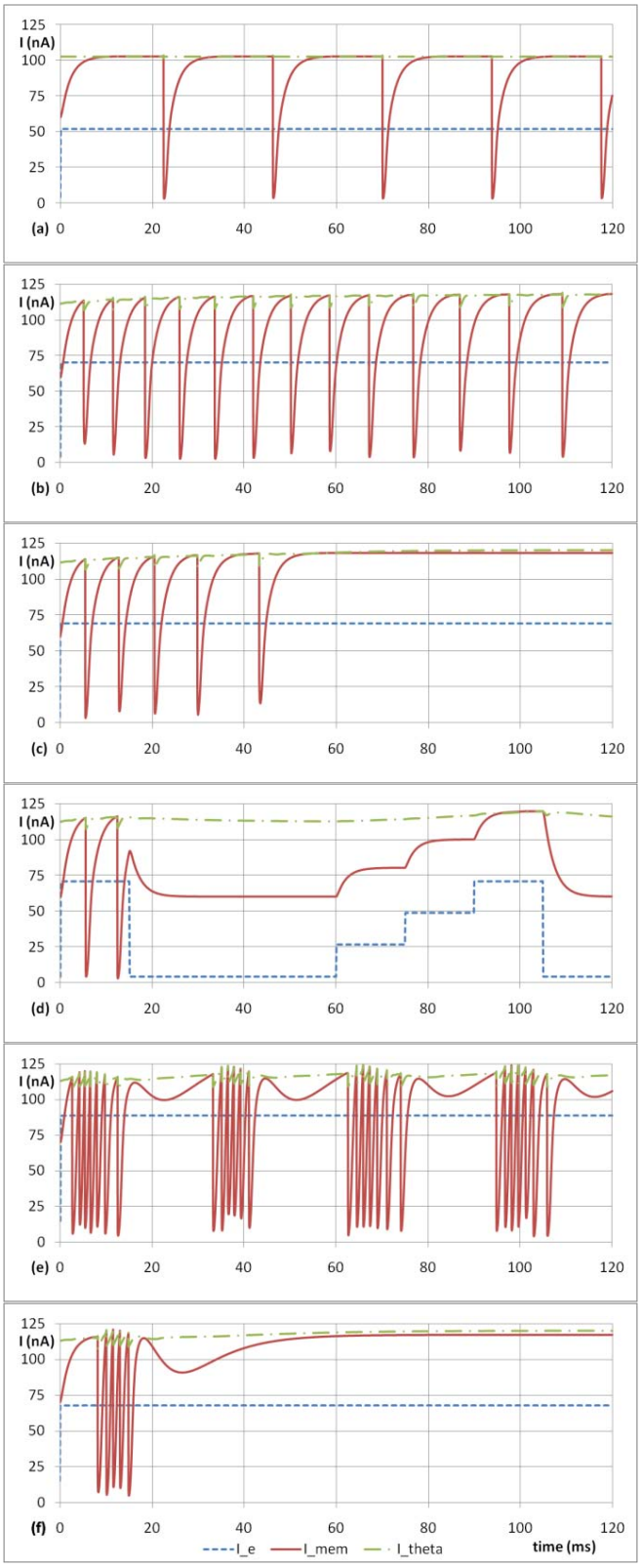

Figure 5. Examples of spike patterns. (a) Class 1; (b) Spike Frequency Adaptation; (c) Phasic Spiking; (d) Accomodation; (e) Tonic Bursting; (f) Phasic Bursting.