PERGAMON

Neural Networks 14 (2001) 617–628

Neural Networks

www.elsevier.com/locate/neunet

2001 Special issue

Building blocks for electronic spiking neural networks

A. van Schaik*

Computer Engineering Laboratory, School of Electrical and Information Engineering, University of Sydney, NSW 2006, Australia

Received 31 January 2001; accepted 31 January 2001

Abstract

We present an electronic circuit modelling the spike generation process in the biological neuron. This simple circuit is capable of simulating the spiking behaviour of several different types of biological neurons. At the same time, the circuit is small so that many neurons can be implemented on a single silicon chip. This is important, as neural computation obtains its power not from a single neuron, but from the interaction between a large number of neurons. Circuits that model these interactions are also presented in this paper. They include the circuits for excitatory, inhibitory and shunting inhibitory synapses, a circuit which models the regeneration of spikes on the axon, and a circuit which models the reduction of input strength with the distance of the synapse to the cell body on the dendrite of the cell. Together these building blocks allow the implementation of electronic spiking neural networks.

Keywords: Analogue VLSI; Spiking neurons; Neuromorphic engineering; Bio-inspired systems; Electronic neuron; Electronic synapse; Electronic dendrite; Electronic axon

1. Introduction

Biological spiking neurons have been well studied over the course of the previous century and much has been discovered about the details of action and membrane potential generation. Although theoretically it might be possible to incorporate all known details into an electronic model, we shall have to make a trade-off between the detail incorporated in the model and the actual size of the circuit. One approach, started by Mahowald and Douglas (1991) is to implement neuron circuits which are analogue approximations of the Hodgkin and Huxley (1953) model. This can yield highly detailed single neuron models, but takes up a lot of chip area. For instance, Rasche and Douglas (2000) describe a silicon neuron that implements a conductance-based neuron model. They were able to put a single neuron featuring about 30 adjustable parameters on a 4 mm^2 chip. However, in most cases, building a single neuron model is not the ultimate goal, but only a necessary step in order to simulate the collective behaviour of a large group of neurons. It is thus important to minimise the amount of detail in order to reduce the circuit size, allowing more neurons to be put on the same chip.

In this paper we present a simple electronic spiking neuron model and some circuits to model the interaction between neurons, or between the neurons and the outside world. These circuits were developed as electronic models for the auditory pathway (van Schaik, 1998) and in consequence we will make reference to elements of the auditory pathway in this paper. However, the circuits presented here are in no way limited to modelling the auditory pathway, and are generally applicable as electronic models of spiking neural networks.

2. An electronic spiking neuron model

A very simple neuron model developed by van Schaik, Fragnière and Vittoz (1996) is shown in Fig. 1. In this section we will discuss the electronic implementation of the model, the spike generation process, and the difference between the electronic neuron model and the well-known Leaky-Integrate-and-Fire neuron model.

2.1. Implementation

The membrane of a biological neuron is modelled by a membrane capacitance, C_{mem}; the membrane leakage current is controlled by the current source, I_{leak}. In the absence of any input ($i = 0$), the membrane voltage will be drawn to its resting potential (controlled by V_-) by this leakage current. Excitatory inputs ($i > 0$) simply add charge to the membrane capacitance. Inhibitory inputs are modelled by a negative input current ($i < 0$). If an excitatory

* Tel.: +61-2-9351-4705; fax: +61-2-9351-7209.
E-mail address: andre@ee.usyd.edu.au (A. van Schaik).

0893-6080/01/$ - see front matter
PII: S0893-6080(01)00067-3

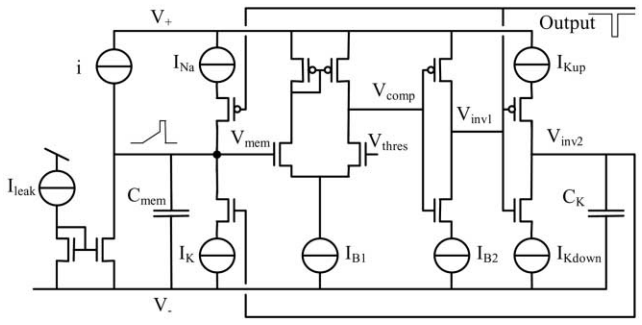

Fig. 1. A simple electronic neuron circuit.

current larger than the leakage current is injected, the membrane potential will increase from its resting potential. This membrane potential, V_{mem}, is compared with a controllable threshold voltage V_{thres}, using a basic transconductance amplifier driving a high impedance load. If V_{mem} exceeds V_{thres}, an action potential will be generated. We will discuss the generation of the action potential in detail in the next section.

The above circuit is very similar to the sodium–potassium neuron circuit described by Sarpeshkar, Watts and Mead (1992). Their implementation uses the current in the left branch of the comparator in Fig. 1 to create the upswing of the spike and to charge the capacitor C_K. This saves a few transistors, but also results in less freedom to control the different parameters of the neuron independently. The reduction in chip area is minor since most area is taken up by the two capacitors anyway. We have therefore preferred to keep the circuit more flexible by introducing a few extra parameters. Sarpeshkar's circuit is a good and straightforward way of simplifying the circuit presented here though when the extra options are not needed.

2.2. Spike generation

The generation of the action potential in the neuron circuit is patterned after the biological neuron, in which an increased sodium conductance creates the upswing of the spike and in which the delayed blocking of the sodium channels plus delayed increase of the potassium conductance creates the downswing. The circuit models this as follows: If V_{mem} rises above V_{thres}, the output voltage of the comparator V_{comp} will rise to the positive power supply. The output of the following inverter V_{inv1} will thus go low, thereby allowing the "sodium current" I_{Na} to pull up the membrane potential. At the same time, however, a second inverter will allow the capacitance C_K to be charged at a rate controlled by the current I_{Kup}. As soon as the voltage V_{inv2} on C_K is high enough to allow conduction of the NMOS whose gate it controls, the 'potassium current' I_K will be able to discharge the membrane capacitance. Although the operation of the circuit differs slightly from the biological neuron model in that the potassium current only creates the downswing of the spike and that the sodium current only switches off again when V_{mem} drops below V_{thres}, the spiking behaviour of the neuron does not seem to be affected by this.

Two different potassium channel time constants govern the opening and closing of the potassium channels. The current I_{Kup} which charges C_K controls the spike width, since the delay between the opening of the sodium channels and the opening of the potassium channels is inversely proportional to I_{Kup}. If V_{mem} now drops below V_{thres}, the output of the first inverter V_{inv1} will become high, cutting off the current I_{Na}. Furthermore, the second inverter will then allow C_K to be discharged by the current I_{Kdown}. If I_{Kdown} is small, the voltage on C_K will decrease only slowly, and as long as this voltage stays high enough to allow I_K to discharge the membrane, it will be impossible to stimulate the neuron if I_{ex} is smaller than I_K. Therefore, I_{Kdown} can be said to control the 'refractory period' of the neuron.

Finally, I_{B1} and I_{B2} are two bias currents needed to limit the power consumption of the circuit; they do not influence the spiking behaviour of the neuron in any major way.

All current sources in Fig. 1 are implemented with single transistors. These transistors of course will only behave like current sources when their drain-source voltage is larger than about 100 mV, since they operate in weak inversion. If we ignore this limitation and assume that these transistors behave like current sources until their drain-source voltage becomes zero, we can adopt a piecewise linear approach. Furthermore, if we assume that $V_- = 0$, we can describe the circuit model with the following equations:

$$C_{mem}\frac{dV_{mem}(t)}{dt} = i(t) - I_{leak}, \qquad V_{mem}(t) < V_{thres} \tag{1}$$

$$\text{If } V_{mem}(t) = V_{thres}: \text{ set } t_s = t;$$

$$\text{while } t_s < t < t_s + T_S: \text{ set } V_{mem}(t) = V_+;$$

$$\text{while } t_s + T_s < t < t_s + T_S + T_R: \text{ set } V_{mem}(t) = 0, \tag{2}$$

where $i(t)$ is the input current, C_{mem} the membrane capacitance, I_{leak} the leakage current, $V_{mem}(t)$ the membrane voltage, V_{thres} the threshold voltage, V_+ the power supply voltage, T_S the width of the spike, T_R the duration of the refractory period, and t_s is the time t when $V_{mem}(t)$ reaches V_{thres}. Eq. (1) describes the leaky-integration of the input current on the membrane capacitance. In this circuit the leak comes not from a conductance in parallel with the membrane capacitance, but from a leakage current. In the piecewise linear approach, I_{leak} will flow as soon as $V_{mem} > V_-$. The membrane potential will decrease until it reaches its resting value as long as $i(t) < I_{leak}$. Only when $i(t) > I_{leak}$ will the membrane potential increase.

Eq. (2) describes the reset operation. Whenever V_{mem} reaches the threshold V_{thres}, V_{mem} is drawn quickly to the positive power supply V_+, and after a delay T_S it is reset to its resting value, where it will stay for a duration T_R.

With a constant input current i ($i > I_{leak}$), the neuron will reach the spiking threshold in a constant time T_I after it

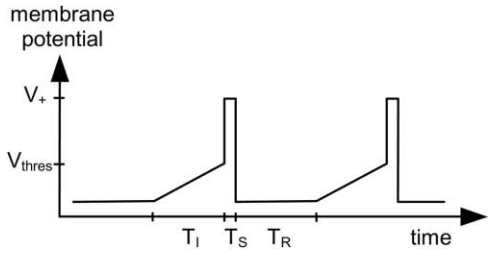

Fig. 2. Typical evolution of the membrane potential in the neuron circuit for a constant input current.

leaves its refractory period. Integrating (1) and solving for $V_{\text{mem}} = V_{\text{thres}}$ yields:

$$T_{\text{I}} = C_{\text{mem}} \frac{V_\theta}{i - I_{\text{leak}}} \tag{3}$$

The spike frequency is given by:

$$f_{\text{spike}} = \frac{1}{T_{\text{I}} + T_{\text{S}} + T_{\text{R}}} \tag{4}$$

An example of this spiking for a constant input current is given in Fig. 2.

The main difference between the (simplified) circuit model and the computational Leaky-Integrate-and-Fire (LIF) neuron (see for instance Tuckwell, 1988) is the fact that the circuit model has a leakage current and the LIF model a leakage conductance. In the neuron circuit, the membrane potential will stay at the resting potential when the stimulation current is smaller than the leakage current. When the stimulation current is larger than the leakage current, the membrane potential will always reach the spiking threshold after a time T_{I} given in Eq. (3). The leakage current thus not only functions as a leak which returns the membrane potential to its resting value, but also as a stimulation threshold. A stimulation current below this threshold will not cause spikes.

This is not dissimilar to the role of the leakage conductance g_{L} in the LIF neuron. In this model the threshold voltage V_θ will never be reached when $i/g_{\text{L}} < V_\theta$, so that we can define a stimulation threshold equal to $g_{\text{L}}V_\theta$ below which the input current will never generate spikes.

2.3. Results

We have realised 32 copies of the neuron shown in Fig. 1, together with some circuitry to facilitate communication of signals on- and off-chip (see Section 3.1) on a 1×2.5 mm die, using the ECPD10 (1 μm) technology of ES2. All transistors have a W/L ratio of 10/10 μm except for the switches and inverters, which are 2/10 μm, and C_{K} and C_{mem} are 10 pF.

Although the proposed neuron model is very simplistic, this neuron model already allows us to simulate the spiking behaviours characteristic of different neuron types by changing its biases. As shown in van Schaik et al. (1996), Post Stimulus Time Histograms (PSTHs) closely resembling those of ventral cochlear nucleus neurons can be obtained with this circuit. An advantage of the analogue VLSI implementation is that one may change the bias voltages and see the neuron model react in real time. This largely simplifies the task of determining the settings to use. Because of its simplicity, the neuron model is small; hundreds of neurons can be put on a reasonably sized chip. This makes it possible to implement and study the neural architectures of the auditory pathway and determine their utility as auditory signal processors, as for instance in van Schaik and Meddis (1999) and van Schaik (2000).

3. Neural interactions

Having an electronic model of a neuron is interesting in itself, but most of the neural processing in the brain is carried out by interactions between neurons. Most neurons in the brain communicate by sending spikes over axons which make synapses with the cell bodies and dendrites of other neurons. In this section, we will look in more detail at these different aspects of neural interaction.

3.1. Spikes

In van Schaik et al. (1996), we were able to obtain PSTHs from the electronic spiking neuron that are similar to those of different types of neurons in the ventral cochlear nucleus, even though the input to the neurons was a continuous analogue signal. In those experiments, the sound was filtered by an electronic cochlea and inner hair cell model. However, the output of these circuits was a current representing the spike probability on a group of auditory nerves innervating a small section of the cochlea. The fact that the PSTHs of the circuit can be made to look similar to the neural PSTHs, even when the input of the neuron circuit is a continuous signal instead of spikes, highlights a cardinal question in the research domain of neuroscience: ‘Why does the brain use spikes?’

There is an obvious, albeit not very satisfactory, answer to this question. It is clear that spikes are used to transmit information over relatively long distances in the brain and the entire nervous system. It would be impossible to communicate analogue signals over these distances because the internal resistance, membrane capacitance, and leakage of the axons will degenerate the signal. If the exact shape of the signal does not carry any information, then it is easy to regenerate the signal along an axon, but it is not possible to do this for an analogue waveform for which the shape does matter. It is generally accepted that the exact form of an action potential is not important, but that the spike signals an event, and that most information is coded by the timing of the spike, the average rate of spikes on the axon, and/or the position of the neuron which originates the spike.

Spikes are indeed regenerated along the axon at the nodes of Ranvier. A spike arriving at such a node will depolarise

the local membrane potential, and trigger the spike generation process. The spike generation process includes a refractory period, and this ensures that the spike only travels in one direction along the axon. Actually, the regenerated spike will travel in both directions along the axon to its neighbouring nodes of Ranvier, but since the node closer to the axon hillock has just spiked before, it will be in its refractory period. This means that it is impossible to trigger the spike regeneration process at this node and that the voltage-dependent potassium channels are still open, which shunts further diffusion of the spike in this direction. The node of Ranvier farther away from the axon hillock will not be in its refractory period, and here the regeneration process will be activated. Without a refractory period in the spiking mechanism each node of Ranvier would stimulate the others when it spikes, and spiking would never stop.

Both spiking itself and the refractory period are thus needed for communications between neurons. However, the refractory period is also an important variable that creates different spiking behaviours in different neurons. For instance, van Schaik and Meddis (1999) show that this can be used to tune the neuron and make it sensitive to particular amplitude modulation frequencies in its input signal. This example shows that although the refractory period is necessary to transmit spikes over an axon, it also serves an important function in the signal processing performed in the brain. This brings us to the real question about spikes: 'Are spikes just needed for communication in the brain, or do they also play a major role in neural signal processing?' And, more pragmatically: 'Does signal processing with spikes offer important advantages over other types of signal processing?'

The answer to the first question is certainly that spikes do play an important role in the actual signal processing, for the simple reason that the brain evolved as a system which uses spikes as its information carrier. However it is still largely unclear what the signal processing operations are that the brain performs in treating, for instance, the auditory sensory input with spikes. What is more, the way in which information is coded with spikes is still an issue of debate, and as long as we do not understand how the information is coded, then how can we hope to understand the operations that the brain performs on the information?

Information can be coded by an average rate of spikes, averaged either in time, or over a group of similar neurons, or both. In this case the precise timing of the individual spikes has no importance. At the other end of the scale, information can be coded by the arrival time of an individual spike; all the information is in the timing. The brain, however, does not have to choose between one code or the other and uses both, sometimes even at the same time. Furthermore, the brain will have to use group coding, be it a temporal or average rate code, because single neurons have too low maximum spike rates and are too prone to error, noise or cell death. For this reason all computation in the brain will have to be collective, and for the same reason collective computation is an interesting option for analogue VLSI, which also uses low-precision elements. Especially in the lower centres of the auditory pathway, timing seems to be important, but it costs effort to keep the timing between different events in the different stages in the auditory pathway. It is therefore likely that at the higher stages of processing the brain uses population codes which do not depend on relative spike timing anymore.

Relative timing of spikes hints at signal processing operations like coincidence detection and correlation of different signals, a multiplicative operation. On the other hand, average rates represent analogue variables, and hint of addition and subtraction of signals using excitatory or inhibitory synapses. It has been shown by Maass (1996, 1997) that it is theoretically possible to build a Turing machine from multilayer perceptrons using temporal coding with spikes, and Leaky-Integrate-and-Fire neurons. Most of the standard neural network theory to date however has been based on the assumption that the information is coded by spike rates. These mathematical constructions tell us that in principle all computational operations are possible with neurons using either temporal coding or average rate coding. They do not tell us, however, which operations can be implemented efficiently and which cannot.

Another approach is to look at operations we can perform particularly effectively with spikes. Three such operations are discussed below; all are believed to occur in the brain.

Perhaps the simplest such operation is coincidence detection: using spikes, only a digital AND operator is needed to implement it. The brain, however, does not use an AND operator, but a spiking neuron to implement coincidence detection. This results in fuzzy coincidence detection; because an incoming spike creates a change in a neuron's membrane potential lasting longer than the spike itself, a second spike can add to the influence of the first spike. Although membrane depolarisation is greatest when the spikes coincide, two not-quite-coincident spikes are still more likely to generate an action potential than a single, isolated spike. The temporal sensitivity of this process, controlled by the leakage conductance and membrane capacitance of the neuron, will be different for different cell types.

Applying fuzzy coincidence detection to spike trains results in synchronicity detection; a coincidence detecting neuron will fire most when two spike trains are synchronous; its response will decrease as the synchrony between the two spike trains decreases.

As third operator, consider cross-correlation of two signals: synchronicity must be detected between one signal and delayed versions of the other. A delay line is simpler to implement for spikes than for analogue signals, because no information is contained in the form of the spike; we can just regenerate each spike after a certain delay. In the case of an analogue waveform, we have to somehow reproduce the entire waveform after a certain delay.

In summary, one advantage of spike coding seems to be the fact that it is simple to implement synchronicity detection, which provides a powerful way of comparing two temporal signals. The ease with which coincidences can be detected also allows the brain to use synchronicity as a way of coding information. It has been shown for instance that many different types of neurons in the cochlear nucleus are synchronised by amplitude modulation at a particular modulation frequency; in absence of this modulation component these neurons automatically desynchronise (Frisina et al., 1990; Rhode & Greenberg, 1994). The synchronisation of a certain group of neurons is thus a way to code the presence of a particular feature, the amplitude modulation frequency in this case. In van Schaik and Meddis (1999), we show an example where the synchronisation of sustained chopper neurons from the ventral cochlear nucleus and coincidence detection performed by inferior colliculus neurons are used to extract this amplitude modulation frequency.

3.2. Inter-chip communication with spikes

Although unrelated to the spiking of neurons in the brain, it makes good engineering sense to adopt spike coding of signals for inter-chip communication. Whenever we have a large number of outputs in parallel, as is the case when we have a large group of neurons on chip, we cannot simply connect each output to another chip with a dedicated wire. The traditional solution in this case is to scan serially through the output. This means that at each clock cycle the output of a different element is read, so each output is only read once every N clock cycles, where N is the total number of outputs. When N grows large, the clock frequency will have to become too large in order to maintain enough temporal resolution on each output. In the case of a large array of neurons, we would be scanning all neurons, most of which would be inactive most of the time. It makes more sense just to transmit the fact that a certain neuron spiked at a certain time. This type of communication has been termed 'event driven communication' (Lazzaro, Wawrzynek, Mahowald, Sivilotti & Gillespie, 1993; Mortara, 1995).

Rather than providing a wire for each output, we can communicate spike events using a single data channel shared by all the neurons on a chip. Each neuron has a unique address; when a neuron spikes, its address is sent onto the channel. The address pulse can be very short if the address detector on the other end is fast enough; we can regenerate a spike with biological duration on the receiving end of the data channel. With spikes lasting only about 1 μs and a maximum spike rate limited to a few hundred Hertz (as it is in biological neurons), even the most active neuron will occupy the data channel less than 0.1% of the time.

In the most direct implementation of this idea (Mortara, 1995), every neuron may access the data channel at any time, which thus preserves perfectly the time structure of the output signals. The only problem with this communication is that it is possible for two or more neurons to access the data channel at the same time, resulting in an address collision. Using a special address coding scheme, for instance always the same numbers of ones and zeros in the address code of each neuron, we can detect these collisions and decide to ignore these addresses. This introduces noise in the communication process, but if collisions are rare, this noise level is very low.

One situation, in which the chance of having collisions is increased, is when information is coded by synchronous activity of a group of neurons. This situation is of particular interest to us and warrants special attention, since we have identified synchronicity as an important property in spike-based computation. In this case several neurons spike at the 'same' time to signal a salient feature. The 'sameness' in time of two spikes is however measured at a biological time scale, in the order of 1 ms. So two spikes about 100 μs apart are still considered synchronous on this scale, but on the time scale of the communication process they can be quite far apart, if for example each communication pulse only lasts 1 μs. After regenerating spikes of biological duration on the other end of the data channel, we can easily detect the synchronicity of the spikes on the biological time scale again. The problem is therefore not as bad as it looks, but still it increases the probability of collisions with respect to an activity of the neurons which is evenly distributed in time.

An alternative solution is to use the so called arbitered event driven communication protocol (see Sivilotti, 1991; Mahowald, 1992; Boahen, 2000). In this scheme an additional circuit is used to arbiter the access to the communication bus. The arbiter allows only one neuron at a time to access the bus and will make the other neurons wait until the bus becomes available again. Again, if short spikes are used and the bus is not overloaded, neurons will at most have to wait for a few microseconds and spike timing will be mostly preserved.

The unarbitered version of the address event coding as proposed by Mortara (1995) has been used on the neuron chip of Section 2.3. The circuit for the input decoding and output encoding is shown in Fig. 3. The address of each of the 32 neurons on chip is coded with three ones and four zeros, yielding a seven-bit communication bus. Note that the

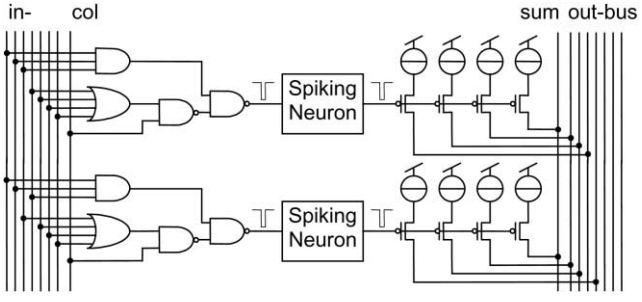

Fig. 3. Input and output coding of the neuron chip.

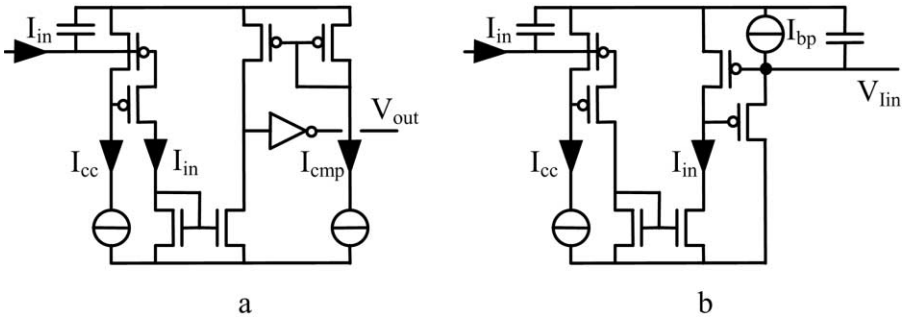

Fig. 4. Current conveyors: (a) with comparator, (b) with line driver.

minimum number of bits needed to code for 32 addresses is five, so the overhead created by this encoding (which allows collision detection) is only two wires. At the input side of the array, an address-decoding block stimulates with a spike the neuron at a position corresponding to the address that has been detected. The address-decoding block may be set either to stimulate none of the neurons when an address collision is detected ($col = 1$), or to stimulate all the neurons with addresses that could have created this collision code ($col = 0$).

When a neuron spikes, apart from sending its address on the bus, it also sends a current spike on a wire common to all neurons. This allows us to measure the number of spikes by measuring the total current on the wire and dividing it by the unit current used to send a single spike. This has been used in van Schaik et al. (1996) to create the PSTHs in that paper.

No circuits were included to reduce the spike width at the output side or to restore the biological spike width at the input side. With only 32 neurons on a chip, collisions will not happen very often even with the wider spikes, unless we are synchronising the neurons with the input signal. If that is the case, we will be interested specifically in the synchronicity of the neurons on the chip; we can use the single common output wire to detect this.

Note that the address is coded on the output bus by current pulses, but by voltage pulses on the input bus. We have used current pulses on the output bus because all neurons will drive the same bus wires, and currents injected on a single wire will just add up. Using current pulses on the output bus also allows combining of the output buses of two chips into a single bus, i.e. combining two seven-bit buses into a single seven-bit bus. In this way neurons with the same address on the two chips will both send their pulses to a neuron which detects this address on a third chip.

For the input bus, however, voltages are preferable so that we can contact the inputs of each address decoding block with the same bus wires. Each chip will receive input addresses coded by current pulses and will convert these to voltage pulses. Furthermore, the output pin of the sending chip and the input pin of the receiving chip, plus the external wire, will add a substantial capacitance to the communication link. Therefore, we would need a lot of current to change the voltage on the wire quickly. However, using a virtual ground at the input of the receiving chip to sense the current, we can keep the voltage on the wire constant. The circuit which does this for each input wire is shown in Fig. 4(a). The structure with the two PMOS transistors in the top left of the circuit, together with the current source I_{cc}, serves to keep the voltage on the input line constant while conveying the input current to an internal node. The input current is then copied by the current mirror, and this current is compared with the current I_{cmp} to determine if the line is at 1 or 0.

In some cases though, the input to the chip will not be in the form of pulsed addresses, but in the form of a continuous input current. Again, we can hold the voltage on the input line constant and use a current mirror with 32 outputs to create copies of the input current for each neuron. Creating 32 copies of the current means driving 33 gates which together have a considerable capacitance. To drive this capacitance, we can use the inverse of the current conveyor structure, as shown in the top right of Fig. 4(b). This avoids drawing the current needed to charge or discharge the capacitance from the input current. We can thus make more current available to drive the capacitance than the input current would allow, which will speed up the communication.

3.3. Synapses

The connections between biological neurons are called synapses. A spike arriving at a synapse can be excitatory, inhibitory, or shunting inhibitory, depending on the neurotransmitter released by the pre-synaptic cell, and on the type of receptors in the post-synaptic cell. An excitatory input will increase the membrane potential; an inhibitory input will decrease the membrane potential, and a shunting inhibitory input will draw the membrane potential towards the resting potential. The effect of these synapses can be modelled with the circuit of Fig. 5(a). The thin lines represent spikes arriving at the different types of synapses. Note that to control the excitatory synapse, the spike has to be inverted with respect to the biological spikes. The figure also shows the part of the neuron circuit that models the passive membrane properties, i.e. the membrane capacitance, and the leakage current towards the resting potential.

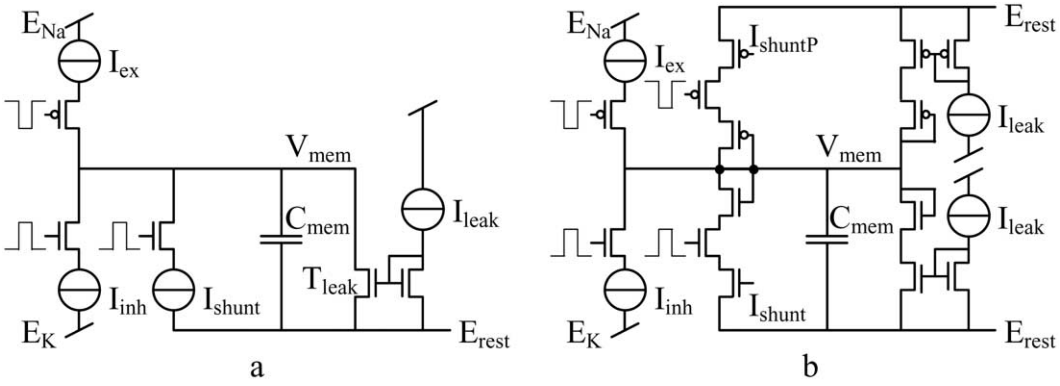

Fig. 5. Circuit for three different types of synapses. (a) basic circuit, (b) corrected for hyperpolarisation.

In this circuit, an excitatory input will increase the membrane potential by an amount:

$$\Delta V_{\text{mem}} = \frac{T_S(I_{\text{ex}} - I_{\text{leak}})}{C_{\text{mem}}}, \tag{5}$$

where T_S is the duration of the spike. Both the inhibitory and the shunting inhibitory inputs decrease the membrane potential by a similar amount. However, this is only true when V_{mem} is above E_{rest}, because the output current of transistor T_{leak} is only equal to I_{leak} when V_{mem} is at least about 100 mV larger than E_{rest}. When V_{mem} is equal to E_{rest}, the current through T_{leak} is obviously zero.

The implementation of Fig. 5(a) is limited to $V_{\text{mem}} \geq E_{\text{rest}}$; when V_{mem} is below E_{rest} the current through T_{leak} will be inverted, effectively drawing the membrane potential up towards the resting potential, but the amount of current with which it will do this will now increase exponentially with decreasing V_{mem}. Therefore, this current will get very large for a V_{mem} only a few hundred millivolts below E_{rest}. A simple way to attempt to correct this problem would be to use a diode-connected transistor in series with T_{leak} so that the current can only flow through this branch when V_{mem} is larger than E_{rest} and add a second branch using PMOS transistors to model the leakage current when V_{mem} is below E_{rest}. Furthermore, because the current source I_{shunt} will be implemented in a similar way as I_{leak}, we will have to do the same for the shunting inhibitory synapse. This then yields the circuit of Fig. 5(b).

The problem with this solution is that normally E_{rest} will be closer to E_K than to E_{Na}. The PMOS transistors that have their bulk at the E_{Na} and their source at E_{rest} will thus have their threshold voltage increased due to the body effect. This will mean that these branches can only conduct properly when V_{mem} is well below E_{rest}, which is not possible if E_{rest} is close to E_K. Furthermore, we cannot solve this by placing the PMOS transistors in a well that is tied to E_{rest}, because this would forward-bias the drain-well junction of the diode-connected transistors when V_{mem} rises above E_{rest}. We could design a bias circuit that correctly biases the well of the PMOS transistors as a function of V_{mem}, but it is clear that having hyperpolarisation and inhibitory synapses complicates the circuit considerably.

The neuron chip presented in Section 2.3 has not implemented hyperpolarisation. This also means that inhibition can only be in the form of shunting inhibition, because the inhibitory synapse can not draw the membrane potential to a value lower than its resting potential. The two possible synapses for this neuron are thus the excitatory and shunting inhibitory synapses shown in Fig. 5(a).

In the biological neuron, the change in membrane potential is mediated by changes in the ion conductances and thus a voltage-dependent current injection, whereas in the electronic circuit it is caused by direct (voltage-independent) current injection. This causes some differences in the shape of the Excitatory Post Synaptic Potential (EPSP), as shown in Fig. 6(a) and (b). However, in both cases the EPSP shows a steep rising flank and a shallow falling slope, the most important features of the EPSP shape. An excitatory input will have its maximum effect only at a certain time not too long after the arrival of the spike at the synapse and after this time the influence of the excitatory input on the membrane potential slowly decreases. This means that two spikes arriving at the same time at two excitatory synapses will yield a larger increase of the membrane potential than two spikes that arrive within a short time of each other, and two spikes arriving within a long time interval will have the same effect as just the second spike by itself.

Another difference between the biological synapse and the electronic synapse is that the chemical transmission at the synapse of the biological cell causes a synaptic delay, which is absent in the electronic circuit. Although this changes the absolute timing of the spikes, it does not alter the relative timing of the spike, at least not if the biological synaptic delay is considered constant. Of course, the relative timing between two different pathways is only retained if the number of neurons in each pathway is the same. Although an extra synapse in one pathway will not change the delay in the electronic circuit, in both the biological and the electronic case an extra neuron will add a considerable delay to this pathway. If we take care that the electronic

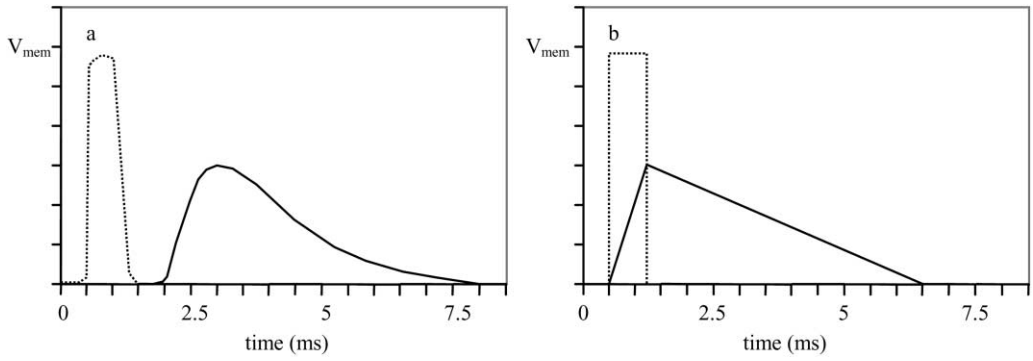

Fig. 6. Schematic representation of an EPSP in a biological neuron (a) and in the electronic neuron (b). The dotted trace represents the timing of the pre-synaptic membrane depolarisation, but is not necessarily on the same voltage scale. The post-synaptic depolarisation is often much smaller than the pre-synaptic depolarisation.

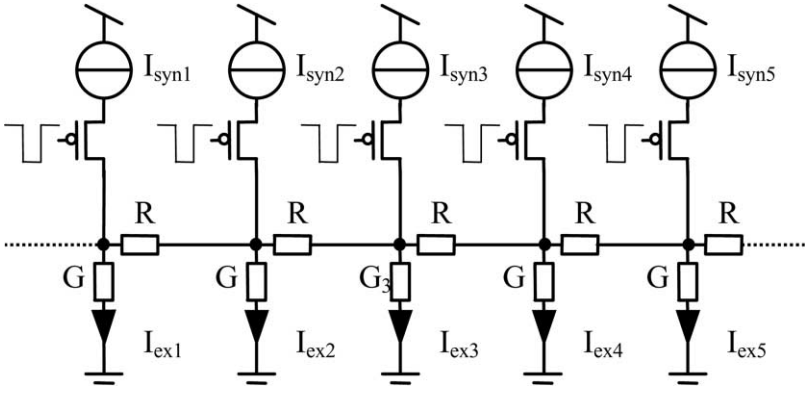

Fig. 7. An electronic dendrite with resistors.

neuron adds a delay in the pathway which is equal to the sum of the neural and synaptic delay in the biological case, then both systems are equivalent again, from at timing point of view.

3.4. Dendrites

Eq. (5) shows that the synaptic strength can be controlled using I_{ex} or the spike width T_S. This means however that we need a different current source for each synapse, or a way to adapt the spike width locally at the synapse, since a neuron sends a spike of the same width to all the synapses it makes with other neurons. Furthermore, the output of the neuron will have to be physically connected with a wire to each synapse it makes, and these connections would use up most of the silicon area when each neuron makes multiple synapses with other neurons. Moreover, the effect of an excitatory input at a synapse on the membrane potential also depends on where the synapse is situated on the dendritic tree. Because of the internal resistance of the dendrites combined with the membrane leakage current, a synapse situated more apically on a dendrite will change the membrane potential at the axon hillock less than a synapse closer to the cell body, even if they both create the same local membrane depolarisation. The circuit of Fig. 7 models this effect in a compact manner. (See also Elias, 1993 for electronic-dendrite models.)

At the top of Fig. 7, we recognise a number of excitatory synapses, modelled as before by current sources with switches controlled by the input spikes. When a spike arrives at synapse 3 for instance, the current I_{syn3} will flow through the switch for the duration of the spike. Part of this current will flow directly through the conductance G_3; the output current I_{ex3} will be equal to this current. The rest of the current flows into the network to either side of the node which connects the different synapses laterally. If we assume that the network extends infinitely to the left and to the right, equal amounts of current will flow left and right. At the neighbouring nodes these currents will split again, partially continuing to flow through the lateral transistors, and the rest creating the output currents I_{ex2} and I_{ex4}. In this way, a current injected at a certain node in the network creates a symmetrical distribution of output currents; the largest output current occurs at the same node as the injection, and the amplitudes of the output currents decrease as the node gets farther away.

Mead (1989) has shown that in such a resistive network the voltage on node j as a function of the voltage at the input

node i can be approximated by:

$$\frac{V_j}{V_i} \approx e^{\frac{-d_{ij}}{L}}, \qquad L = 1/\sqrt{RG}, \qquad L > 1 \tag{6}$$

where d_{ij} is the distance between nodes i and j measured as the number of intervening lateral resistors, R the value of the lateral resistance, and G the value of the vertical conductance. We can calculate the current distribution from this equation, since the current I_{exj} through the output conductance at node j is proportional to the voltage on node j and an input current I_{syni} at node i will create a certain node voltage V_i. We can therefore write:

$$\frac{I_{exj}}{I_{syni}} \approx \alpha e^{\frac{-d_{ij}}{L}}, \qquad L = 1/\sqrt{RG}, \qquad L > 1 \tag{7}$$

where α is a constant. We can calculate the value of α using the fact that the sum of all output currents has to be equal to the input current. If we assume that the resistive network is infinite, we can write:

$$\int_{d=-\infty}^{\infty} \alpha e^{\frac{-d}{L}} \tag{8}$$

where d is the distance between the output node and the input node measured in lateral elements. Because the distribution of output currents is symmetrical about $d = 0$ we can also write:

$$-\alpha + 2\int_{d=0}^{\infty} \alpha e^{\frac{-d}{L}} = 1 \tag{9}$$

where the '$-\alpha$' term corrects for the fact that we count the term at $d = 0$ twice. The summation term is a standard geometrical series, so that we can solve for α:

$$\alpha = \frac{1 - e^{-1/L}}{1 + e^{-1/L}} \tag{10}$$

Hence, we can determine α as a function of L.

It has been shown (Vittoz, 1994) that a network of resistors as in Fig. 7 can be implemented with MOS transistors in weak inversion, as long as we are only interested in the input and output currents and not in the voltages on the nodes. This can be explained by the following analysis.

The current through a MOS transistor operated in weak inversion as a function of the voltages on its terminals referred to the local substrate is given by:

$$I_{DS} - I_S e^{\frac{V_G - V_{T0}}{nU_T}} (e^{\frac{-V_S}{U_T}} - e^{\frac{-V_D}{U_T}}) \tag{11}$$

with V_G the voltage at the gate, V_S the voltage at the source and V_D the voltage at the drain terminal respectively, all referred to the local substrate. V_{T0} is the threshold voltage of the MOS transistor, U_T is the thermal voltage and I_S is the specific current of the MOS transistor.

The linear behaviour of the transistor in the current domain can be made visible by defining a pseudo-voltage V^* as:

$$V^* = -V_0 e^{\frac{-V}{U_T}} \tag{12}$$

where V_0 is and arbitrary scaling value. We can then rewrite Eq. (11):

$$I_{DS} = g^*(V_D^* - V_S^*) \tag{13}$$

with:

$$g^* = \frac{I_S}{V_0} e^{\frac{V_G - V_{T0}}{nU_T}} \tag{14}$$

This shows that with respect to the current, the MOS transistor in weak inversion behaves as a conductance of which the value can be controlled by its gate voltage. Furthermore, from Eq. (12), we can see that as soon as V_D becomes larger than a few U_T, V_D* becomes almost zero and the drain terminal can thus be considered as a pseudo-ground. We can combine these pseudo-conductances in a network as long as we use the same definition of pseudo-voltage for all transistors, and the same V_0.

It is not a real constraint in our case to operate the transistors in weak inversion. If we want small neurons, the membrane capacitance C_{mem} in the neuron circuit of Section 2 will have to be small (a few pico-Farad at most) since a capacitor consumes a large silicon area. Furthermore, the spike width T will be close to 1 ms to stay in the same range as the biological spikes. Eq. (5) shows that I_{ex} and thus I_{syn} will have to be small if we want to limit ΔV. With these small currents the transistors in the dendritic tree will naturally operate in weak inversion.

Replacing the conductances G and R in Fig. 7 with pseudo-conductances T_G and T_L, results in the circuit of Fig. 8.

Eq. (7), still valid for this circuit, is now a function of the pseudo-conductances:

$$\frac{I_{exj}}{I_{syni}} \approx \alpha e^{\frac{-d_{ij}}{L}}, \qquad L = \sqrt{(1/R)^*/G^*}, \qquad \alpha = \frac{1 - e^{-1/L}}{1 + e^{-1/L}} \tag{15}$$

We can use Eq. (15) to express the pseudo-conductances G* and $(1/R)$* as a function of the control voltages V_{CG} and V_{CR}. Assuming that T_R and T_G have the same geometry, we can write for L:

$$L = e^{\frac{V_{CG} - V_{CR}}{2nU_T}} \tag{16}$$

Together with Eq. (15) this shows us how the strength of a synapse i decreases with increasing distance, when seen from node j. Now imagine that each output current I_{exj} charges the membrane capacitance of a different neuron. Eqs. (15) and (16) then describe how a given synapse i influences a neuron at position j. Because the diffusion network is completely linear, we can apply the superposition principle. Summing up the influence of all synapses on each neuron, we see that although all synapses influence all

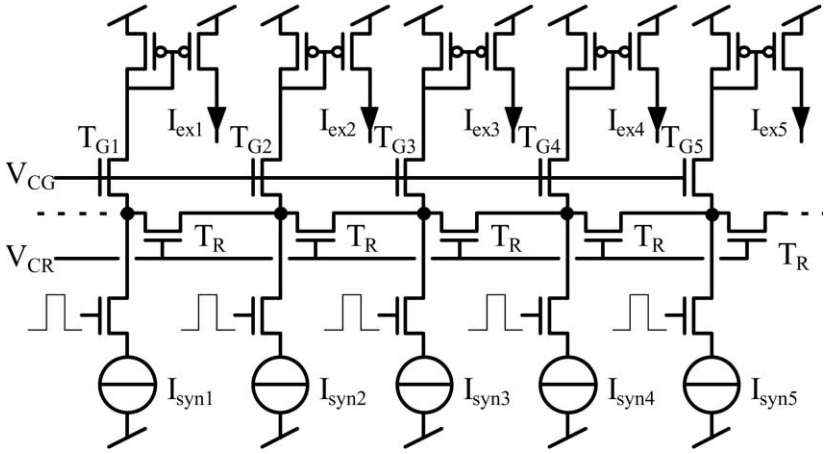

Fig. 8. Electronic dendrite with pseudo-conductances.

neurons, each input has maximum effect on the neuron it is closest to; its strength decreases with increasing distance. This structure can also be extended to a two-dimensional structure in which each input connects to a group of neurons in a 2-D map, and the strength of an input is maximal at the position of its synapse and falls off with distance.

The dendritic structure gives us a nice way to connect the output of a neuron to a group of other neurons which are close to each other in an array, without the need to draw a connection wire from this neuron to all the neurons it influences. Two such structures have been included at the inputs of the 32 neurons on the neuron chip of Section 2.3, one for the excitatory synapses, and one for the inhibitory synapses. This allows the excitatory and inhibitory inputs to have different diffusion lengths. Each address received from another chip will have one bit indicating if it is excitatory or inhibitory in addition to the seven bits which code for position. Fig. 9 shows the final structure.

The circuit of Fig. 8 does not model the effect that a synapse more distal on the dendrite will generate a post synaptic potential that starts later and lasts longer in addition to being weaker, because the dendritic membrane capacitance has not been modelled in this circuit. We could model these temporal effects by adding a capacitor at each node in the circuit of Fig. 7. However, in this case we are not only interested in the network currents anymore, but also in the node voltages at the nodes to which the capacitors are connected. If we want to do this for the circuit with the pseudo-conductances in Fig. 8 we have to add a (pseudo) capacitor, which is compatible with pseudo-voltages in this circuit (seeFragnière, van Schaik & Vittoz, 2000).

3.5. Axons

In principle it is not necessary to model the regeneration of a spike in an axon at the nodes of Ranvier (Fig. 10), because the resistance of the metal wires on chip is low enough to ensure proper transmission of a spike. However, a biological axon will also add a delay which increases with increasing length of the synapse. This delay may be used in certain neural computations; it is a lot easier to model than the dendritic delay because the form of the spike remains the same. In principle, we may use the same circuit as the neuron circuit in Fig. 1 to model the nodes of Ranvier, since both have the same membrane properties. However, in the neuron circuit C_{mem} models the capacitance of the cell body, whereas for the node of Ranvier circuit we only have to model the capacitance of the node plus a section of the axon. We can thus use a smaller capacitance in this circuit. Moreover, since in the electrical circuit there is no coupling from the output of the circuit to its input, unidirectional transmission of the spike is guaranteed and we do not need to model the refractory period of the neuron in the node of Ranvier circuit. Modelling the refractory period in the neuron circuit of Fig. 1 will be sufficient to ensure that there will be no spikes on the axon for a certain period after each spike. This means that the capacitor C_K can be reduced in size too, since it is only used for controlling the spike width, which is in general much shorter than the refractory period of the neuron. For the input of the node of Ranvier circuit we can use the same circuit as for the excitatory synapse, which will allow us to control the delay created by the circuit. Since this circuit only charges the capacitance during the input spike, the delay created by the node of Ranvier circuit will not be larger than one spike width, because each input spike has to create an output spike. For this same reason, we do not need to model the leakage properties of the membrane, which simplifies the node of Ranvier circuit even more.

4. Conclusions

An electronic circuit modelling the spike generation process in the biological neuron has been presented. This neural circuit integrates charge on a capacitor which models the membrane capacitance of a nerve cell. When the voltage on the capacitor reaches a certain threshold value, a positive

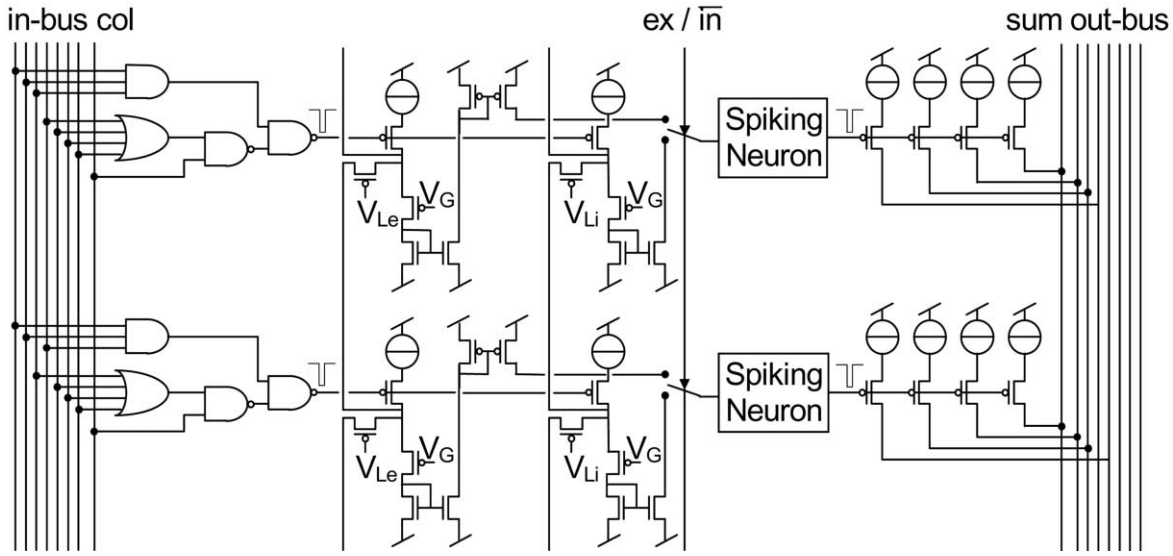

Fig. 9. Input and output structure of the neuron chip, including two dendritic structures.

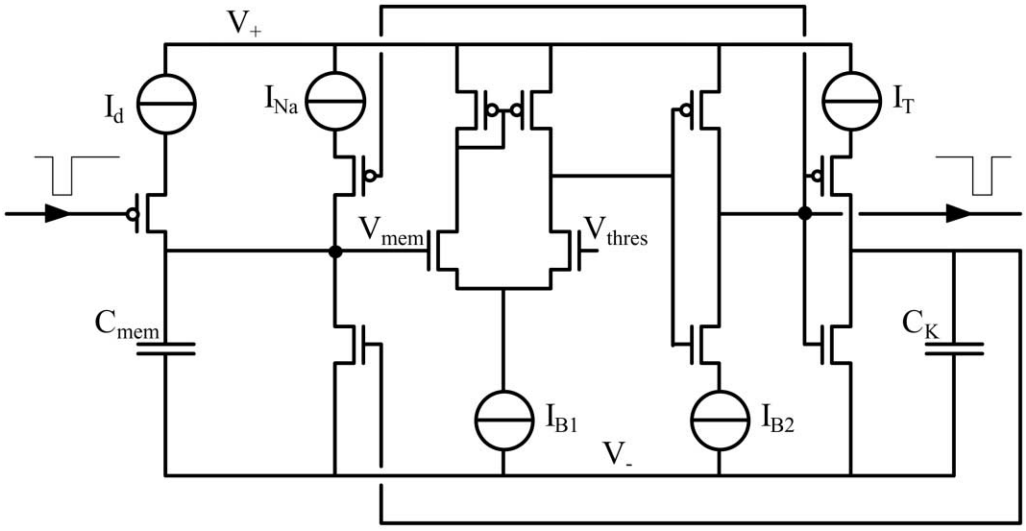

Fig. 10. The node of Ranvier circuit.

feedback cycle is activated, which quickly increases the capacitor voltage. After a short delay, a second feedback restores the resting voltage on the capacitor. After the generation of a spike, during the refractory period, it is harder or impossible to create a new spike, due to the dynamics of this second feedback.

Neural computation obtains its power not from a single neuron, but from the interaction between a large number of neurons. Neurons typically communicate by sending spikes over axons which make synaptic contacts with the dendrites and cell bodies of other neurons. Circuits that model these interactions have also been presented in this paper. They include the circuits for excitatory, inhibitory and shunting inhibitory synapses, a circuit which models the regeneration of spikes on the axon at the nodes of Ranvier, which can be used as a delay line, and a circuit which models the reduction of input strength with the distance of the synapse to the cell body on the dendrite of the cell.

References

Boahen, K. A. (2000). Point-to-point connectivity between neuromorphic chips using address events. *IEEE Transactions on Circuits and Systems II*, *47* (5), 416–434.

Elias, J. (1993). Artificial dendritic trees. *Neural Computation*, *5*, 648–663.

Fragnière, E., van Schaik, A., & Vittoz, E. A. (2000). A log-domain CMOS transcapacitor: design, analysis and application. *Analog Integrated Circuit and Signal Processing*, *22* (2/3), 195–208.

Frisina, R. D., Smith, R. L., & Chamberlain, S. C. (1990). Encoding of amplitude modulation in the gerbil cochlear nucleus: I—A hierarchy of enhancement. *Hearing Research*, *44* (2–3), 99–122.

Hodgkin, A. L., & Huxley, A. F. (1953). A quantative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, *117*, 500–544.

Lazzaro, J., Wawrzynek, J., Mahowald, M., Sivilotti, M., & Gillespie, D. (1993). Silicon auditory processors as computer peripherals. *IEEE Journal of Solid State Circuits*, *26*, 523–528.

Maass, W. (1996). Lower bounds for the computational power of networks of spiking neurons. *Neural Computation*, *8*, 1–40.

Maass, W. (1997). Fast sigmoidal networks via spiking neurons. *Neural Computation.*, *9*, 279–304.

Mahowald, M. (1992). VLSI analogs of neuronal visual processing: a synthesis of form and function. PhD Thesis, California Institute of Technology, Pasadena, CA.

Mahowald, M., & Douglas, R. (1991). A silicon neuron. *Nature*, *354*, 515–518.

Mead, C. A. (1989). *Analog VLSI and neural systems*, Reading MA: Addison-Wesley.

Mortara, A. (1995). Communication techniques for analog VLSI perceptive systems. PhD Thesis, Ecole Polytechnique Fédérale, Lausanne.

Rasche, C., & Douglas, R. J. (2000). An improved silicon neuron. *Analog Integrated Circuits and Signal Processing*, *23*, 227–236.

Rhode, W. S., & Greenberg, S. (1994). Encoding of amplitude modulation in the cochlear nucleus of the cat. *Journal of Neurophysiology*, *71* (5), 1797–1825.

Sarpeshkar, R., Watts, L., & Mead, C. (1992). Refractory neuron circuits. CNS Technical Report, No. CNS-TR-92-08, California Institute of Technology, Pasadena, CA.

Sivilotti, M. (1991). Wiring considerations in analog VLSI systems, with application to field-programmable networks. PhD Thesis, California Institute of Technology, Pasadena, CA.

Tuckwell, H. C. (1988). *Introduction to theoretical neurobiology*, Cambridge, UK: Cambridge University Press.

van Schaik, A. (1998). Analogue VLSI building blocks for an electronic auditory pathway. PhD Thesis, no. 1764, Swiss Federal Institute of Technology, Lausanne, Switzerland.

van Schaik, A. (2000). An analog VLSI model of periodicity extraction. In S. Solla, *Advances in neural information processing systems 12* (pp. 738–744). Cambridge, MA: MIT Press.

van Schaik, A., Fragnière, E., & Vittoz, E. (1996). An analogue electronic model of ventral cochlear nucleus neurons. *Proceedings of the Fifth International Conference on Microelectronics for Neural Networks and Fuzzy Systems; MicroNeuro'96* (pp. 52–59). Los Alamitos, CA: IEEE Computer Society Press.

van Schaik, A., & Meddis, R. (1999). Analog very large-scale integrated (VLSI) implementation of a model of amplitude-modulation sensitivity in the auditory brainstem. *Journal of the Acoustical Society of America*, *105* (2), 811–821.

Vittoz, E. A. (1994). Analog VLSI signal processing: why, where, and how. *Journal of VLSI Signal Processing*, *8*, 27–44.