Circuits and Systems
Mekelweg 4,
2628 CD Delft
The Netherlands
http://cas.et.tudelft.nl/

CAS-2024-

M.Sc. Thesis

Circuits and Systems for a Spiking Neuromorphic Network in 28 nm CMOS

Bart Leonard Hettema, B.Sc.

Abstract

Neuromorphic computing can be used to efficiently implement spiking neural networks. Such spiking neural networks can be used in edge AI applications, where low power consumption is paramount. The use of analog components allows for extremely low power implementations. This thesis contributes the designs of an analog spike generator, synaptic elements and an accumulating neuron in 28 nm CMOS technology. The elements are assembled in a neural network and laid out in an SoC. Energy consumption numbers of less than 1 pJ/synaptic operation are achieved in the analog neuromorphic components.

TU Delft

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Circuits and Systems for a Spiking Neuromorphic Network in 28 nm CMOS

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Bart Leonard Hettema, B.Sc.
born in Voorburg, The Netherlands

This work was performed in:

Circuits and Systems Group
Department of Microelectronics
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Delft University of Technology
Department of
Microelectronics

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Circuits and Systems for a Spiking Neuromorphic Network in 28 nm CMOS"** by **Bart Leonard Hettema, B.Sc.** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 28 January 2024

Chairman:

dr. Rene van Leuken

Advisor:

dr. Rene van Leuken

Committee Members:

dr. Rajendra Bishnoi

dr. Amir Zjajo

Abstract

Neuromorphic computing can be used to efficiently implement spiking neural networks. Such spiking neural networks can be used in edge AI applications, where low power consumption is paramount. The use of analog components allows for extremely low power implementations. This thesis contributes the designs of an analog spike generator, synaptic elements and an accumulating neuron in 28 nm CMOS technology. The elements are assembled in a neural network and laid out in an SoC. Energy consumption numbers of less than 1 pJ/synaptic operation are achieved in the analog neuromorphic components.

Acknowledgments

This project would not have been possible without the people I had around me. Shashanka and Davide, it was great have you in our room on the 17th floor. This was an amazing start of our neuromorphic adventure. My thanks also to Jan, who taught me about analog layout, and to Johan, who taught me that it isn't any easier for digital.

I would like to thank Amir, Rene and Sumeet for their assistance and patience during the writing of this thesis. It wasn't always a smooth ride for me, even when the finish was nearly in sight, but your combination of relentless optimism, support and some tough love when needed helped me get there.

My parents and sister, who are always there for me.

And last but not least my thanks to my fantastic colleagues at Innatera, who were always supportive and made it possible for me spend time on this project when it was necessary.

Bart Leonard Hettema, B.Sc.
Delft, The Netherlands
28 January 2024

Contents

List of Figures

List of Tables

1 Introduction

Neuromorphic computing systems aim to mimic the biological behaviour of neuron cells. Early work on this was conducted in the late 1980s by Mead [1]. Since then, the work on neuromorphic systems has specialised from the general imitation of the nerve system to implementations of neural networks for learning capabilities [2]. Among applications of these networks are general pattern recognition, image classification and sensor data processing, such as radar data. A network of spiking neurons is able to react very fast to changing input data and from the beginning has shown the potential for large power savings [1]. This makes the technology well-suited for time-series data processing.

A neuromorphic network is a spiking neural network (SNN). This means that input data is encoded as pulse trains. These pulses are propagated through the neural network, the network is an event-based system.

1.1 Problem statement

In previous work by Stienstra [3] and You [4], circuits for spiking neurons and synapses in have been developed in 65 nm CMOS technology. These circuits can be made smaller and more power efficient by implementing them in the 28 nm technology node. A challenge in this project is the porting of designs that are designed for 65 nm CMOS to the 28 nm technology node. Such scaling can make the circuit more power efficient, but can also make it more susceptible for variability in the manufacturing process. The behavioural dependence on temperature changes with this scaling as well. The circuit can be optimised for power, area, stability and speed.

Another challenge is the large amount of input and output connections in the network. The components must be made such that they connect together easily and uniformly in an array. These connections include not only the data inputs to the network and outputs from the neurons, but also the connections that are necessary to control and read the biasing and weight voltages of the network elements.

1.2 Approach

The first step is to investigate the existing work. It is investigated which behavioural properties of neurons and synapses are implemented in models and circuits. The average power consumption and energy per spike of the current designs are key parameters to review, as well as the spike frequency range that the designs handle.

Next the neuromorphic circuits are implemented in 28 nm CMOS technology. Starting from minimum sized transistors, the dimensions and biases are optimised for the desired sub-threshold transistor behaviour. Furthermore, once the schematic design is completed in 28 nm technology, a complete hardware layout of the network will be made.

1.3 Goals

The goal of this thesis project is to build a test chip of a neuromorphic network by combining and improving the neuron and synapse designs of Stienstra [3] and You [4] and porting them to a more efficient technology, comparing them the state of the art and previous implementations. Since it is a prototype chip, as many parameters, weights and biasing voltages of the network as possible will be externally settable and readable. The design will be laid out in the TSMC 28 nm CMOS technology. The effect of manufacturing mismatch and process variability for the implemented neuromorphic circuits are characterised.

1.4 Contributions

The main contributions presented in this thesis are:

- Designs of a synapse and a neuron in 28 nm CMOS technology.
- A neuromorphic network architecture using a distributed synapse structure.
- An implementation of this network as a self-contained system on chip (SoC) in silicon.
- Simulated and measured results of the neuromorphic components,.

1.5 Thesis outline

Chapter 2 provides a background on neurons and synapses and models of their spiking behaviour. Prior work on neuron circuits is reviewed and different neuron implementations are compared. In Chapter 3 the design, implementation and optimisation of the synapse and neuron circuits is discussed, as well as the integration of the components in a network on an SOC. Chapter 4 shows the simulation and characterisation of the neuromorphic components. The taped-out chip has been used to verify the spiking functionality and to measure the power consumption of the neuromorphic components. These measurements are presented in Chapter 5. Finally, the results of this project are discussed and the conclusions drawn. Proposals for future work are presented.

2 Background

2.1 Neuromorphic computing

Electronic circuits can mimic the behaviour of biological nervous systems. Such electronic systems are called neuromorphic systems or networks. What is remarkable about the way the biological nervous system processes data, is the incredible energy efficiency with which it does so. The most efficient supercomputer at the moment of writing is the Henri at the Lenovo Flatiron Institute, USA, capable of performing 65,4 milliard operations per Watt [5], or 15 pW/operation. For normal CPUs, this efficiency lowers to about 10 milliard operations per Watt [6], about 100 pW/operation. In contrast, our brain uses on the order of 10^{-16} W/operation [1], about a hundred thousand to a million times less.

The main mechanism that enables the use of electronic components to construct nervous systems, is the behaviour of CMOS transistors operating in their sub-threshold region, described in Section 2.5.1. In this region, the current through the transistor is exponentially dependant on its input voltage. This is analogous to the exponential dependence of active ionic channels to the membrane voltage of a neuron cell [1], [7]. Section 2.2.3 expands further on a biological model of the neuron cell, its membrane, and its interfaces.

The nervous system consists of a network of nerve cells, referred to as neurons. Figure 2.1 shows an illustration of neurons. A neuron consists of the cell body, dendrites and axons—each indicated in the illustration. The dendrites reach out around the neuron in a tree and receive signals from their surroundings. Axons, on the other hand send signals to other neurons. The interface between an axon and the receiving neuron is called the synapse. In the synapse, neurotransmitter molecules go from the axon to receptors on the receiving neuron.

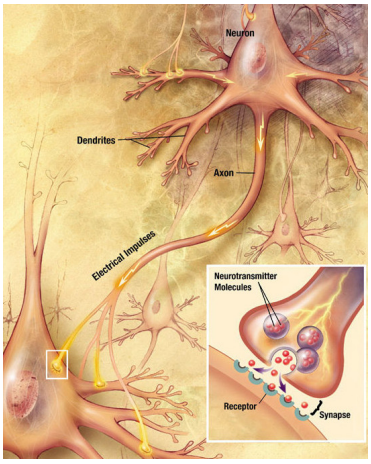


Figure 2.1: Neuron and synapse

A neuromorphic system implements two mechanisms of this nerve system. The neuron cell is implemented as a whole, without regard for the individual parts that make up the cell, since in neuromorphic systems we only aim to mimic the behaviour of the cell, not its physical layout. Additionally, the synapses are implemented as separate circuits. The synapses not only connect the neurons together, but also implement memory storage and enable learning behaviour [8]. The circuits are divided in this way because multiple synapses are connected to a single neuron.

2.1.1 Spiking neural networks

Similarly to the biological nervous system, neuromorphic systems are event-based. These events are encoded as pulse trains, or spikes in time, leading to the term spiking neural network (SNN). Input spikes are propagated through the network, amplified or diminished by synapses and converted to different spike shapes and firing frequencies at neurons. The usage of spikes, as opposed to more continous signals, means that the system elements are only active for short periods, when they are receiving, processing and outputting spikes. This enables the system to use little power.

One distinguishing feature of neuromorphic networks is that they, in contrast to classical Von Neumann architectures, mix computation and memory storage [8]. The same is true for biological networks. In biological systems memory is in place as the long- and short-term plasticity mady by the synapse connections. In mixed-signal CMOS implementations the memory is often implemented as multiple local digital storage bits connected to a local DAC, which can control an analog synapse.

2.1.1.1 Weights

Neural networks performs inference by summing input signals on a node with a different weight assigned to each input. This is done by multiplying weights on signals in the synapses, which are connecting an input neuron and output neuron. In an analog network, a spike generates a current in a synapse, with an ouput amplitude corresponding to the weight of that synapse. These currents can be added by connecting the outputs of synapses together in a single node.

In its simplest form, in a network of N input neurons and M output neurons, the current I_j into the output neuron No_j—with $0 < j \leq M$—can be summarised as

$$I_j = \sum_{i=1}^{N} A_i(t) \cdot w_{ij} \tag{2.1}$$

where A_i is the input spike into the synapses $S_{i1} \ldots S_{iM}$ and w_{ij} is the weight of synapse S_{ij}, connecting input neuron Ni_i and output neuron No_j.

2.1.1.2 Encoding

In an SNN simulations, these values represent frequencies. For an individual output:

$$f_{\text{out},j}(t) = f\left(\sum_{i=1}^{N} \left(w_{ij} \cdot f_{\text{in},i}(t)\right) + f_{\text{bias},j}\right) \tag{2.2}$$

where $f_{\text{in},i}(t)$ is the input frequency into the synapses $S_{i1} \ldots S_{iM}$ over time, w_{ij} is the weight of synapse S_{ij}, connecting input neuron Ni_i and output neuron No_j, $f_{\text{bias},j}$ is the constant bias input into output

neuron No_j, $f(f)$ is the activation function and $f_{\text{out},j}(t)$ is the output frequency of output neuron No_j over time.

2.2 Models

There are multiple models that describe the behaviour of the neuron cell and the shape of the AP. These models work mostly by evaluating the voltage over the cell membrane as a function of the injected current. First the biological mechanism of neural spikes is explained. Then four different models will be discussed, ranging from very biologically realistic to abstract but less complicated.

2.2.1 Action potential

The spikes or impulses that are sent between neurons in the nerve system are called action potentials (APs). Figure 2.2 shows a schematic of an action potential. The cell membrane of a nerve cell contains ion channels, which allow specific kinds of ions to cross the membrane. The conductance of these channels can vary, influencing the effect of incoming current on the membrane potential. If a large enough current is injected into the neuron and the membrane voltage crosses a certain threshold, the membrane voltage will spike, producing an AP.

Four stages can be distinguished during this process. The AP begins with a stimulus, in Figure 2.2 slightly after 1 ms. The depolarisation stage is entered if the stimulus is large enough to make the membrane voltage exceed the threshold. The rapid rise at the beginning of this stage opens the sodium channels in the membrane, allowing sodium ions (Na^+) to flow in, resulting in a spike. The third stage is the repolarisation phase. During this phase, the sodium channels are closed, while potassium channels are opened, resulting in an efflux of potassium ions (K^+), lowering the voltage.

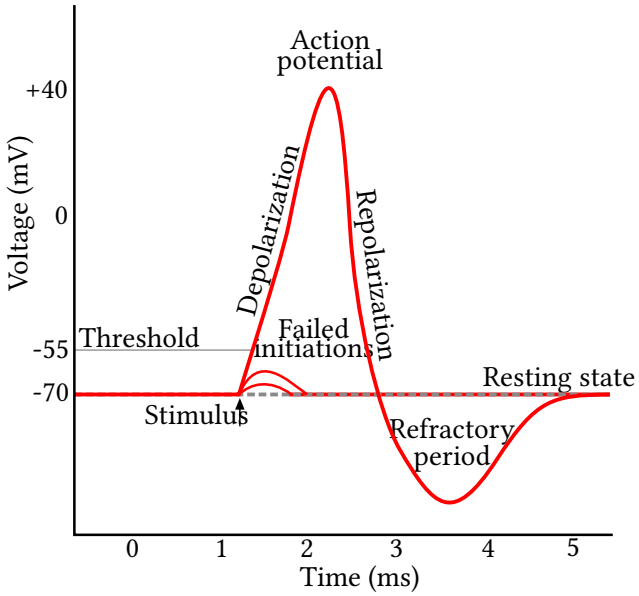


Figure 2.2: Schematic of an action potential, showing the stimulus, the depolarisation and polarisation phases and the refractory period.

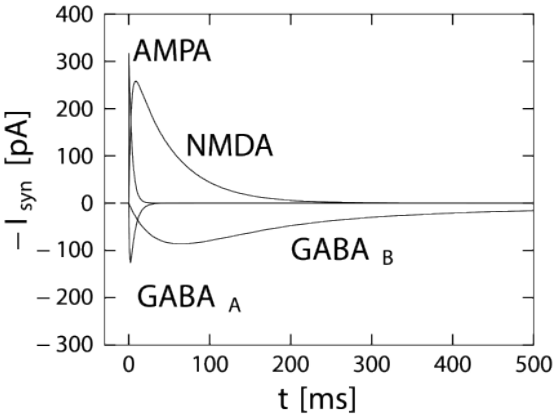


Figure from [10, Fig. 3.2]

Figure 2.3: Post-synaptic current dynamics of receptor types

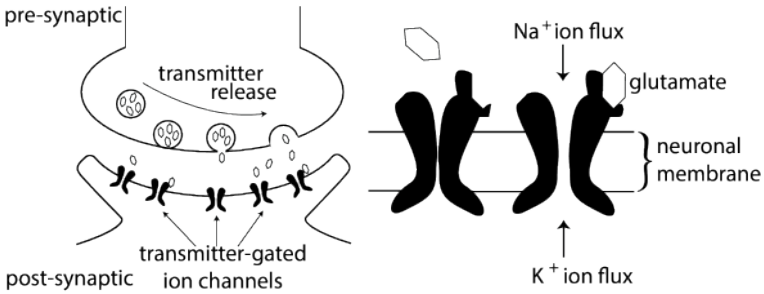


Figure from [10, Fig. 3.1]

Figure 2.4: Schema of synaptic tranmission (left) and postsynaptic AMPA receptor (right)

This efflux of K^+ ions causes an undershoot in the voltage, the refractory period or hyperpolarisation phase. During this time, the potassium channels close and the voltage returns to its resting state. In the refractory period, no new impulses can be generated. The strength of the hyperpolarisation largely determines the maximum frequency at which the spikes can fire [9, Ch. 1].

2.2.2 Synapse dynamics

As shown in Section 2.2.3, current can be passed through ion-activated channels. A similar process can be observed in the synapse, the interface between two neurons (see Figure 2.1). These channels are present at the receptors of the receiving—or post-synaptic—neuron. When a spike arrives at the pre-synaptic neuron, neurotransmitter molecules are activated. These activated neurotransmitters cross the synaptic cleft and are received by a receptor. Specific transmitters fit specific receptors. Once a receptor has received a fitting transmitter, the ion channel can be opened and ions can carry a current into or out of the receiving neuron [10, Ch. 3.1].

The current through the multiple channels of the synapse can be modelled as

$$I_{\text{syn}}(t) = g_{\text{syn}}(t)\big(V_{\text{post}}(t) - E_{\text{syn}}\big) \tag{2.3}$$

with a decaying conductance

$$g_{\text{syn}}(t) = \sum_f \bar{g}_{\text{syn}} \exp\left(\frac{-\left(t - t_{\text{pre}}\right)}{\tau}\right) \Theta\left(t - t_{\text{pre}}\right) \tag{2.4}$$

where V_{post} is the postsynaptic voltage, E_{syn} is the synapse's reverse potential, g_{syn} is the conductance of the synapse, $\bar{g}_{\text{syn}}$ is the maximum conductance, t_{t} is the arrival time of the presynaptic spike and Θ is the Heaviside step function. E_{syn} is often set to 0 V in excitatory synapses and −75 mV in inhibitory synapses [10, Eq. 3.1–3.2], [11].

There are multiple types of receptors. The most important receptor types for neuromorphic computing are α-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid (AMPA), *N*-methyl-D-aspartate (NMDA) and γ-aminobutyric acid (GABA). The AMPA and NMDA receptors are so-called excitatory receptors. When they are active, the current into the neuron increases. Contrarily, the GABA receptors are inhibitory, when they are activated, the current decreases or can go negative. A summary of the properties of the three receptor types is presented in Table 2.1.

Table 2.1: Properties of receptor types

Receptor	Polarity	Rise time [ms]	Fall time [ms]	Conductance dependency
AMPA	+	0,4 to 0,8	5	Neurotransmitters
NMDA	+	20	100	Neurotransmitters, postsynaptic voltage
GABAa	-	3,9	20	Neurotransmitters

2.2.2.1 AMPA receptor

The AMPA receptor is one of the most common receptors. When neurotransmitters bind to AMPA receptors, NA^+ ion channels open. The resulting influx of current polarises the cell and can result in the firing of an AP. This means that the AMPA conductance is directly dependant on the amount of neurotransmitters received.

AMPA are the fastest receptors. The rise time of the synaptic currents due to AMPA receptors is 0,4 ms to 0,8 ms, their fall time is 5 ms. These short rise and fall times can be attributed to the rapid clearing of neurotransmitters and closure of the channels [11]. The different temporal dynamics of current due to the receptor types is shown in Figure 2.3 and summarised in Table 2.1.

2.2.2.2 GABA receptor

The GABA receptor is much the same as the AMPA receptor, but with a reverse effect. However, instead of the positively charged Na^+ ions, the channel of a GABA receptor lets negatively charged Cl^- ions through. This results in a decreasing or even negative current to the neuron, preventing or slowing the firing of an AP.

There are two types of GABA receptors, GABAa and GABAb. GABAb receptors, compared to GABAa, NMDA and AMPA receptors, need a much higher presynaptic signal in order to activate. It is hard to achieve such high stimulation, and for this reason only GABAa receptors will be taken into account. GABAb receptors also have different temporal dynamics than GABAa receptors. While GABAa has a typical rise time of 3,9 ms and a fall time of 20 ms, GABAb is about 20 times slower [11].

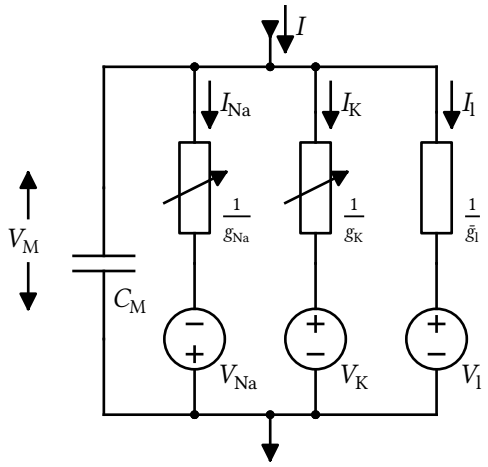


Figure 2.5: Hodgkin & Huxley model equivalent circuit

2.2.2.3 NMDA receptor

Like the AMPA receptor, the NMDA receptor is an excitatory receptor. However, NMDA receptors have a different activation mechanism. NMDA receptors can be blocked by a concentration of Mg^{2+}. This blockage is removed when a postsynaptic current is applied. This means that NMDA receptors can only conduct if both a presynaptic voltage (to release neurotransmitters) and a postsynaptic voltage (to remove the Mg^{2+} blockage) are present. This double effect also means that the synapse model as presented in Equation (2.3) is too simple for NMDA receptors. To account for the Mg^{2+} block, an extra term $B(V)$ is introduced:

$$I_{syn}(t) = g_{syn}(t)B(V_{post}(t))\Big(V_{post}(t) - E_{syn}\Big) \tag{2.5}$$

and

$$B(V) = \frac{1}{1 + \exp(-0{,}062V)\frac{[Mg^{2+}]_0}{3{,}57}} \tag{2.6}$$

where $[Mg^{2+}]_0$ is the external Mg^{2+} concentration [11].

NMDA receptors are also the slowest of the three receptor types, with a typical rise time of 20 ms and a fall time of 100 ms [12].

2.2.3 Hodgkin & Huxley model

For modelling, the most important characteristics of a neuron is the current through the cell membrane and the way that this excites an AP. The AP can be seen as a changing voltage over the cell membrane. An extensive model of the membrane behaviour has been developed by Hodgkin and Huxley [13]. The fundamentals of this model will be explained. An equivalent circuit of this model can be found in Figure 2.5. The top of the circuit is outside of the cell, while the bottom is inside the cell. The circuit thus represents the cell membrane between. The membrane is normally at a constant resting potential. A current I enters the cell, and causes the voltage difference V_M over the membrane.

The current density through the membrane and the change of voltage over the membrane is de-

scribed in [13] as:

$$I = C_{\mathrm{M}} \frac{\mathrm{d}V}{\mathrm{d}t} + I_{\mathrm{Na}} + I_{\mathrm{K}} + I_{\mathrm{l}} \tag{2.7}$$

where

$$I_{\mathrm{Na}} = g_{\mathrm{Na}} (V - V_{\mathrm{Na}}) \text{ is the current density carried by sodium ions,} \tag{2.8a}$$
$$I_{\mathrm{K}} = g_{\mathrm{K}} (V - V_{\mathrm{K}}) \quad \text{is the current density carried by potassium ions, and} \tag{2.8b}$$
$$I_{\mathrm{l}} = \bar{g}_{\mathrm{l}} (V - V_{\mathrm{l}}) \quad \text{is a leakage current density due to chloride and other ions} \tag{2.8c}$$

where V is in mV, I in μA/cm², C_{M} in μF/cm² and the conductances g in mS/cm².

V is the membrane voltage V_{M} minus its resting potential. V_{Na}, V_{K} and V_{l} are constant and represent the saturation points of the ion channels. [9, Ch. 2]

The potassium and sodium conductances are dynamic, and are described by:

$$g_{\mathrm{K}} = \bar{g}_{\mathrm{K}} n^4 \tag{2.9a}$$
$$g_{\mathrm{Na}} = \bar{g}_{\mathrm{Na}} m^3 h \tag{2.9b}$$

n, m and h are dimensionless and are defined by the differential equations

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \alpha_x (1 - x) - \beta_x x \tag{2.10}$$

for $x = n, m, h$; where α_x and β_x parameters are in ms⁻¹. They are fitted values depending on the value of V, but do not change over time.

α and β are different for different types of neurons. In [13] Hodgkin and Huxley fit α and β for their experiments to be

$$\alpha_n = 0{,}01 \frac{V + 10}{\exp\left(\frac{V+10}{10} - 1\right)}, \quad \beta_n = 0{,}125 \exp\left(\frac{V}{80}\right) \tag{2.11a}$$
$$\alpha_m = 0{,}1 \frac{V + 25}{\exp\left(\frac{V+25}{10} - 1\right)}, \quad \beta_m = 4 \exp\left(\frac{V}{18}\right) \tag{2.11b}$$
$$\alpha_h = 0{,}07 \exp\left(\frac{V}{20}\right), \quad \beta_h = \frac{1}{\exp\left(\frac{V+30}{10} + 1\right)} \tag{2.11c}$$

The Hodgkin & Huxley model describes the behaviour of a neuron as changing conductances. The equations above show the model with three conductance types, as defined in the original model. However, the model is not inherently limited to these three. Additional conductances are often added, for example to model potassium ion channels with different time constants. These conductances can be added by fitting additional functions for α and β. Detailed conductance-based models exist with tens of ion channels. The addition of more ion channels in the adds more parameters, studies with up to a hundred parameters exist [14].

The model as described in Equations (2.7) to (2.11) has a total of 13 parameters that can be tuned to describe a neuron:

- C_M: membrane capacitance
- V_{Na}, V_K, V_l: saturation points
- $\bar{g_K}$, $\bar{g_{Na}}$, $\bar{g_l}$: ion channel conductances
- α_n, α_m, α_h, β_n, β_m, β_h: function parameters

Computing $\frac{dV}{dt}$ requires 20 additions and subtractions, 32 multiplications and divisions and 6 exponentiations.

The Hodgkin & Huxley model is most useful when we look at the behaviour of a neuron as changing conductances. This changing conductance can be well modelled in electrical circuits. Some neuron implementations using conductance-based models will be discussed in Section 2.3. The modelling of sodium and potassium ions and leakage can be seen in the presence of the Na^+, K^+ and *Leak* blocks in the neuron circuit in Figure 3.5.

2.2.4 Izhikevich model

The Hodgkin & Huxley model describes the neuron in much detail. However, this comes with the price of a large computational complexity.

An alternative model that greatly reduces computational complexity has been introduced by Izhikevich [15]. This model is able to produce twenty different spiking patterns [16], using only four main tuning parameters. In contrast to the Hodgkin & Huxley model, Izhikevich does not model the system in a biologically meaningful way. Still, the model attempts to imitate the biological behaviour. Some details, such as the ability to change the spike's precise shape, are lost though [17].

This model uses the membrane potential of the neuron V, a membrane recovery variable u and the current input to the cell I—together with the four parameters a, b, c and d—to construct a two-dimensional system of differential equations:

$$\frac{dV}{dt} = 0{,}04V^2 + 5V + 140 - u + I \tag{2.12}$$

$$\frac{du}{dt} = a(bV - u) \tag{2.13}$$

with the additional condition:

$$\text{if } V \geq 30\,\text{mV, then } \begin{cases} V \leftarrow c \\ u \leftarrow u + d \end{cases} \tag{2.14}$$

where V and the potential reset value c are in mV, t in ms, I and the parameter d in pA, the parameter b describes the relation between u and v and is in nS and the recovery times constant a is in ms^{-1}.

Typical values are $a = 0{,}02$, $b = 0{,}2$, $c = -65\,\text{mV}$, $d = 2$. The expression $0{,}04v^2 + 5v + 140$ and the reset condition of v are fitted for a neuron with resting potential between -70 and $-60\,\text{mV}$, threshold between -55 and $-40\,\text{mV}$ and a spike peak value of 30 mV. Other values are possible for other neurons [15]. This model uses 4 parameters (a, b, c, d) to model the neuron behaviour and one parameter (spike peak value) and a fitted expression to describe the voltage levels, such as the resting potential and threshold range. 13 operations are needed to compute $\frac{dV}{dt}$.

2.2.5 Integrate and fire model

A simpler model than the Izhikevich model is the integrate-and-fire (IF) model. In this model, the current input to the neuron cell is integrated onto the cell membrane. Once a certain treshold voltage is reached, the neuron fires a spike. This model can be extended using a leaky integrator. This causes the membrane potential to decay over time to a resting potential. The basic IF model can be very simple to implement as a circuit [2]. This has the disadvantage, however, that although the behaviour can be made suitable for a spiking neural network, it does not realistically model the biological system. The IF model can be made more realistic by adding non-linear or quadratic terms. Of course, this increases the complexity as well. The most complex implementation of the I&F model is the adaptive exponential I&F model, which is comparable to the Izhikevich model in complexity and is able to model many of the same spiking patterns [17].

In general, IF models consist of a differential equation describing the integration of input current I on a membrane capacitance and the resulting change in membrane voltage u during the integration phase, and a reset condition which resets the neuron when a spike has occurred. The base form of the IF is then:

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = I \tag{2.15}$$

$$\text{if } V \geq V_{\mathrm{T}}, \text{ then } V \leftarrow V_{\mathrm{reset}} \tag{2.16}$$

This base model requires 3 parameters: the capacitance C, the threshold voltage for a spike V_{T} and the potential to which the membrane voltage is reset after a spike, V_{reset}.

2.2.5.1 Leaky integrate and fire

Leaky integrate and fire (LIF) is the simplest model that is in use. The model can be represented as a circuit consisting of a capacitor with capacitance C in parallel with a leakage resistor connected to a resting potential u_{rest}, with leak conductivity g_{l} . This makes it so that when there is no input current, the membrane voltage will go to V_{rest} [18]:

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = -g_{\mathrm{l}}\left(V - V_{\mathrm{rest}}\right) + I \tag{2.17}$$

$$\text{if } V \geq V_{\mathrm{T}}, \text{ then } V \leftarrow V_{\mathrm{reset}} \tag{2.18}$$

The rate at which this occurs is expressed with the timeconstant $\tau = RC$.

The LIF model adds two parameters to the base model, g_{l} and V_{rest}, totalling 5. 6 operations are needed to compute a timestep. [16] reports 5 operations, but does not use C as a parameter in the model, therefore needing one less multiplication or division.

2.2.5.2 Adaptive exponential integrate and fire

Exponential In the LIF model, an output spike is generated when the integrator voltage reaches a specific threshold voltage. This spike has no defined amplitude and has no width, it is reset immediately. This does not accurately model some neuron behaviours. A distinction can be made between a threshold voltage, and the peak voltage of the spike. If the external input is stopped while V is

below the threshold voltage V_T, it will decay to the resting potential. However, if V is above V_T, V will increase up to the spike peak voltage V_{peak} before it is reset [18].

This behaviour can be implemented by adding an exponential term depending on the difference between V and V_T:

$$C\frac{dV}{dt} = -g_l(V - V_{rest}) + g_l\Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) + I \tag{2.19}$$

$$\text{if } V \geq V_{peak}, \text{ then } V \leftarrow V_{reset} \tag{2.20}$$

This will cause V to increase exponentially towards V_{peak} when larger than V_T. The rate at which it increases is controlled by the new parameter Δ_T, the slope factor. The lower Δ_T is, the closer the exponential model resemples the standard LIF model [14]. When V is slightly below V_T there will be a balance between the linear leakage and the exponential term. This replaces the strict threshold with a more realistic threshold zone, in which spikes can be initiated but are not certain to do so. This allows the model to describe additional spike patterns like subthreshold oscillations. When V is significantly lower than V_T the exponential term is negligible and the model is a linear integrator.

Frequency adaptation Neuron behaviour can be affected by the spike history of the neuron. A period of high activity can raise the effective threshold, reducing spike frequency, and can change subthreshold behaviour. Adding adaptation to the model allows the model to describe frequency relaxation and bursting behaviour. An adaptation current w is added to the model:

$$C\frac{dV}{dt} = -g_l(V - V_{rest}) + g_l\Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) - w + I \tag{2.21}$$

$$\tau_w\frac{dw}{dt} = a(V - V_{rest}) - w \tag{2.22}$$

$$\text{if } V \geq V_{peak}, \text{ then } \begin{cases} V \leftarrow V_{reset} \\ w \leftarrow w + b \end{cases} \tag{2.23}$$

The parameters τ_w, a and b are added. τ_w is the adaptation time constant. a is the factor for subthreshold adaptation. This increases the adaptation current when V is high but not spiking, causing subthreshold oscillation. w is increased by b whenever a spike occurs, the spike-triggered frequency adaptation.

Compared to an accurate reference neuron, the adaptive exponential integrate and fire (AdExp IF) model correctly described 96 % of spikes. Removing the exponential current term for a sharp threshold reduces this to 88 %. Disabling the subthreshold adaptation reduces the spike accuracy by 2 %, while setting b to 0, removing the spike-triggered adaptation, has a larger effect, reducing the accuracy to 67 %. The combination of the adaptation current and the exponential spike mechanism significantly improves on the standard LIF model, or an IF model with just one of the two [14].

The combined AdExp IF model needs 7 additions and subtractions, 7 multiplications and divisions and 1 exponentiation to compute $\frac{dV}{dt}$.

2.2.6 Model comparison

The models are compared on the amount of parameters that can be adjusted to change the neuron behaviour. More parameters allows for more adjustability of the spike shape and pattern, but also

make it more difficult to fit the model and find the desired behaviour. An increase in parameters will also generally be more complicated to implement in a neuron circuit design. However, having fewer parameters can mean that not all neuron behaviours can be modelled.

The amount of operations to compute one timestep is determined in the expression $V(t+1) = V(t) + \frac{dV}{dt}$. Each addition, subtraction, multiplication, division, exponentiation and comparison is counted as one operation. The total amount of operations is then the operations to calculate $\frac{dV}{dt}$ plus 1 addition to add it to the previous value plus 1 comparison. In [16] exponentiation is approximated as 10 operations in order to compare the compute complexity on a digital system, the amount of operations using this approximation is reported in parentheses for the models that compute exponentiations. The amount of operations for the Hodgkin & Huxley model is an approximation, since it depends on which fitted functions are chosen for the α and β parameters. The number of parameters and operations will increase if additional conductance functions are added to the model to describe more ion channels.

Table 2.2: Comparison of neuron models

	H&H	Izhikevich	Leaky IF	AdExp IF
Parameters	13[1]	5	5	9
Operations	60 (112)[1]	13	6	17 (26)
Spike shape	dynamic	quadratic	none	exponential
Spike patterns	all	most	some	most
Non-linear near-threshold	yes	yes	no	yes
Frequency adaptation	yes	yes	no	yes
Conductance-based	yes	no	yes	yes

[1] Minimum for the model described here, more if additional ion channels are described

2.3 Implementations

2.3.1 Leaky integrate-and-fire implementations

An ultra-low energy analog leaky integrate-and-fire (LIF) neuron is implemented in [19]. This circuit is based around a hysteresis comparator. The comparator consists of a non-linear transconductance amplifier with a feedback connection from its output to the positive input, and a resistor connected to a bias to set the output DC level.

The neuron uses a reverse polarity architecture, meaning that integration of excitatory current decreases the membrane voltage instead of increasing it. This is done because an inverting comparator was more efficient to implement. The membrane voltage decreases until the lower threshold voltage of the hysteresis comparator is reached. The output then switches high and the output spike begins. This switches on a current source that increases the membrane voltage until the higher threshold voltage of the hysteresis is reached, this is the reset phase. The output then switches low again and the spike has ended. The width of the output spike can therefore be controlled by the hysteresis range in the comparator design and the amount of feedback current during the reset phase.

The very low power consumption is achieved by running the entire neuron circuit in weak inversion. A supply voltage of 0,6 V is used with super high VT transistors with a threshold voltage of

0,66 V. This neuron design implements leaky behaviour in its corresponding synapse instead of in the neuron itself. However, leakyness could also be added to the neuron with an additional resistor or current source between the membrane node and the supply voltage.

A more complex LIF neuron is presented in [20]. This design adds a synaptic integrator between the neuron core and the synaptic current input. The synaptic integrator functions as a first-order low-pass filter core and results in a synaptic voltage V_{syn}, which will be increased or decreased by excitatory and inhibitory input currents.

The combination of the synaptic voltage and the membrane voltage of the soma is

$$C_{\text{mem}} \frac{\mathrm{d}}{\mathrm{d}t} V_{\text{mem}} = I_{\text{p}} - I_{\text{shift}} \tag{2.24}$$

$$C_{\text{syn}} \frac{\mathrm{d}}{\mathrm{d}t} V_{\text{syn}} = I_{\text{in}} - I_{\text{n}} + I_{\text{shift}} \tag{2.25}$$

where I_{p} depends on the membrane potential V_{mem} and control voltage V_{p} and I_{n} depends on the synapse voltage V_{syn} and control voltage V_{n}. I_{in} is the difference between the excitatory and inhibitory input current to the synapse.

A diode-connected NMOS transistor connects V_{mem} and V_{syn} and functions as a level shifter. The current through this transistor, connecting the soma and synapse parts of the circuit, is I_{shift}.

Multiple sources of post-synaptic potentials can be connected to the single synaptic integrator. The amplitude of incoming spike current can be regulated by adjusting two control voltages. A transistor to ground next to the synapse capacitor implements leakage and is controlled by V_{n}.

In steady state the leak current I_{n}, charge current I_{p} and connecting current I_{shift} are all equal. When V_{syn} charges and the difference between V_{syn} and V_{mem} decreases, the level shifter will turn off and $I_{\text{shift}} = 0$. I_{p} then charges C_{mem} and V_{mem} increases until the level shifter turns on again. Conversely, when V_{syn} decreases and the difference to V_{mem} increases, I_{shift} will increase as well and will be larger than I_{p}, discharging C_{mem} and bringing V_{mem} down.

IF based neurons have also been used for digital implementations of a neuromorphic network. Digital neuromorphic networks are often simpler to implement, but have a higher area and power usage. Joubert, Belhadj, Temam *et al.* [21] show that an analog implementation of a leaky I&F neuron uses 20 times less energy and 5 times less area than a comparable digital implementation.

2.3.2 Adaptive exponential I&F implementations

The Institute of Neuroinformatics in Zurich has developed a series of implementations of the adaptive exponential I&F model. Characteristics of these circuits is the presence of a differential pair integrator (DPI) at the neuron input and in an afterhyperpolarisation circuit. The first version is presented in [22]. In [23] the same circuit is shown with an AER interface added to the neuron output.

The neuron behaviour of this circuit is described by a subthreshold current I_{mem} controlled by the membrane potential and an adaptation current I_{ahp} [24]:

$$\left(1 + \frac{I_{\text{th}}}{I_{\text{mem}}}\right) \tau \frac{\mathrm{d}}{\mathrm{d}t} I_{\text{mem}} + I_{\text{mem}} \left(1 + \frac{I_{\text{ahp}}}{I_{\tau}}\right) = I_{\text{mem}_{\infty}} + f\left(I_{\text{mem}}\right) \tag{2.26}$$

$$\tau_{\text{ahp}} \frac{\mathrm{d}}{\mathrm{d}t} I_{\text{ahp}} + I_{\text{ahp}} = I_{\text{ahp}_{\infty}} \tag{2.27}$$

Ignoring the adaptation current, [24] shows that this simplifies to

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} I_{\mathrm{mem}} = -I_{\mathrm{mem}} + f\left(I_{\mathrm{mem}}\right) + \frac{I_{\mathrm{th}}}{I_{\tau}} I_{\mathrm{in}} \tag{2.28}$$

A derivative of this I&F circuit is used in [25], where it is integrated in the ROLLS neuromorphic processor chip. This is an implementation in 180 nm.

[26] and [27] are a further development of this neuron. These implementations strive to minimise leakage currents and firing rate variability. Lower leakage currents allow for longer time constants without increasing capacitor sizes and therefore area, or increasing bias currents and therefore power usage. These longer time constant allow the neuron to have a sub-kHz firing rate, which more closely resembles the biological neuron behaviour the processor aims to mimic. This low leakage is achieved in [27] by using an FD-SOI process and by improving the spike generation circuit of the neuron. The neuron in [26] has an active area of 20 µm². Approximately 0,9 pF of capacitance is needed in addition. [27] uses the same active area but requires a higher capacitance of approximately 1,5 pF.

An alternative is implemented in [28]. This implementation uses the circuit of [27] as its base, but then focuses on the minimisation of the energy per spike. An energy per spike of 990 fJ is achieved. This implementation requires a large capacitor area than other implementations and uses alternate polarity MOM capacitors in contrast to the MIM capacitors used previously. There is a larger firing rate variability reported than in [26]. The cause of this is that the output frequency distribution is bimodal, with the first mode around the expected frequency and a second mode around three times the expected value. A sensitivity analysis shows that the bias current mirrors used to set the leakage, threshold and refractory period are sensitive to mismatch, as well as the DPI input. It is proposed that additional series and parallel transistors must be added to these sensitive parts.

2.3.3 Prior work in group

Previous work done in the Circuits and Systems group [3] has implemented existing neuron circuits in UMC 65 nm. Implementing the different neuron circuits in the same CMOS technology allows for a more direct comparison. The energy per spike, area and necessary capacitance of these implementation were determined.

Among the neurons that were implemented were a LIF neuron based on [20] and an adaptive exponential IF neuron based on the neuron found in [25].

It was found that the LIF neuron was able to be implemented using the least amount of area. However, this design uses more energy per spike event than other neurons. The LIF neuron is not capable of producing biologically realistic spikes.

The adaptive exponential based neuron offered the most realistic spike shape and sits between the other solutions with regards to energy usage. A downside is the large area that is needed for its capacitors. The large advantage of the adaptive exponential based neuron is that both its timing characteristics, such as the spike rate and refractory period, and characteristics of its spike shape, such as the with and reset potential can be controlled.

2.3.4 Implementation comparison

Table 2.3 shows a comparison of the neuron implementations that have discussed.

Key parameters to compare are the energy per spike and the area that is used by the neuron. Energy per spike is used instead of average power because the neuromorphic cores asynchronous and the amount of spikes that is processed varies through time. When no spikes are processed the neurons only consume a small amount of static power.

Since the neurons are often arranged in arrays of hundreds to thousands of neurons, it is desirable to minimise their area in order to maximise the amount of neurons that can be fitted in a neuromorphic core. All neuron implementations that are compared use capacitors in their circuits. The area used by the capacitors is a significant part of the total area, often even the majority.

Capacitance density varies greatly with process technology and implementation choice. The implementations differ in using MIM, MOM or MOS capacitors. For example, the total capacitance of [28] is 4,8 times as large as that of [26] but the capacitor area is 36 times larger. MIM capacitors are used in the latter while alternate polarity MOM capacitors are used in the former. For this reason the implementations are compared on their total capacitance and the capacitor area used in the implementation is reported separately from the active transistor area. Where no separate capacitor area is specified, the active area is the total area including capacitors.

Frequency variability is defined as the standard deviation over the mean of the output frequency of the neuron in a statistical analysis. This shows the effect of process variation and mismatch on the neuron firing rate. When implementing the neuron circuits in smaller CMOS technologies it is more difficult to have only a small amount of variability.

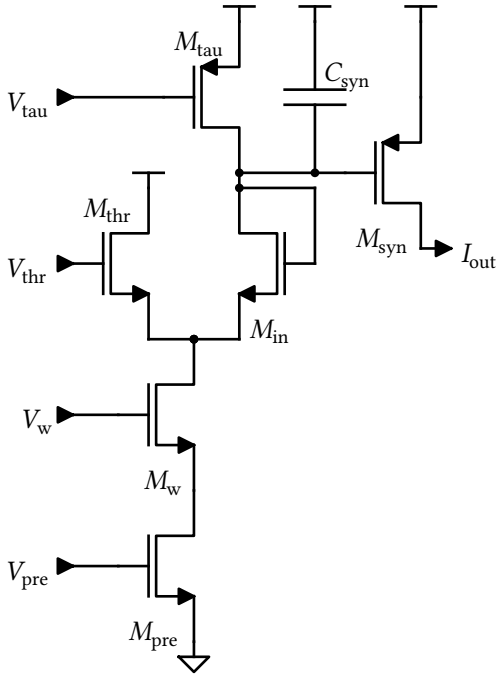
Table 2.3: Comparison of neuron circuit implementations

	[19]	[20]	[26]	[28]	[3]	[3]
Model	LIF	LIF	AdExp IF	AdExp IF	LIF	AdExp IF
Technology	90 nm	130 nm	28 nm FD-SOI	22 nm FD-SOI	65 nm	65 nm
Supply voltage	0,6 V	1,2 V	1,0 V	0,8 V	1,0 V	1,0 V
Energy per spike	0,4 pJ		50 pJ	990 fJ	260 pJ	730 fJ
Active area	442 µm²	22,80 µm²	20 µm²		0,65 µm²	0,18 µm²
Capacitance		40 fF	900 fF	4,3 pF	106 fF	345 fF
Capacitor area		3,696 µm²	50 µm²	1799 µm²	2,6 µm²	25,8 µm²
Typical spike frequency	100 Hz	1 kHz	100 Hz	100 Hz	2 kHz	1 kHz
Frequency variability			5,86 %	56,55 %		
Control signals	3	8	8	10	12	15

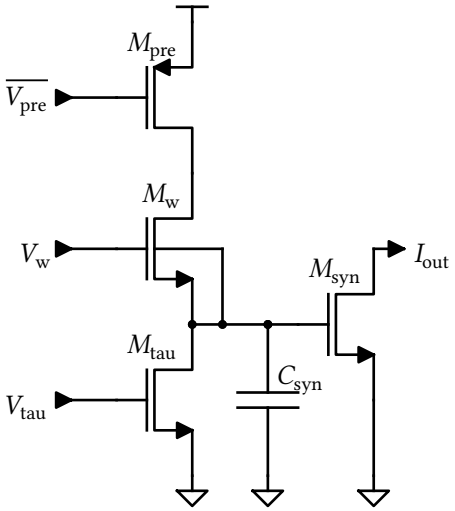
2.4 Synapse

Additionally, in the Circuits and Systems group the work a combination of synapse circuits has been developed for a multi-compartment receptor [29]. The three receptor types discussed in Section 2.2.2 are implemented [4]. These circuits are based on the log-domain integrator and differential-pair integrator synapses of [30] and [23] and are shown in Figure 2.6.

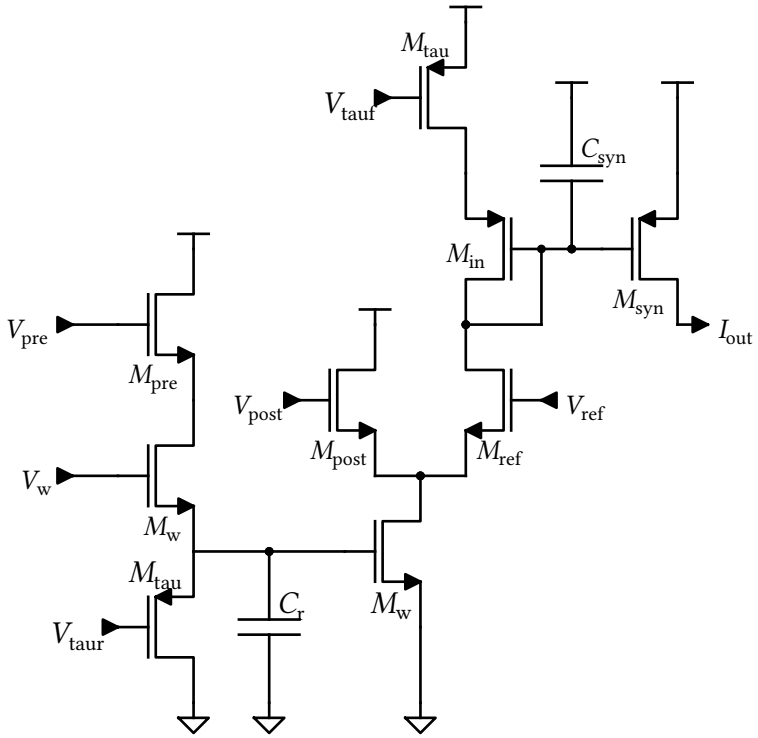
The linear charge-and-discharge synapse[30] forms the base of these implementations. An analog voltage V_w represents the weight of the synapse. A current, controlled by V_w, is integrated on the synapse capacitor C_{syn} during the time that an input pulse is active. The voltage over C_{syn} control the synaptic output current through M_{syn}. M_{syn} operates in weak inversion. Therefore a linear



(a) Circuit of an AMPA receptor



(b) Circuit of an GABAa receptor



(c) Circuit of a NMDA receptor

Figure 2.6: Receptor circuits

decrease in voltage over C_{syn} results in an exponential decay of the output current. This decrease is implemented with a constant leak current that adjusted with the bias signal V_{tau}, controlling the synaptic decay time constant τ.

The AMPA and NMDA receptor circuits use this structure in such a polarity that it results in a current sourcing output, while the GABA receptor circuit uses a complementary structure to have a sinking output current.

All three receptor types can use this same structure as their base. Different input filters are used in the receptors to control how the weight-controlled current charges the synapse capacitor.

The AMPA receptor uses a differential-pair integrator to implement an exponential filter. An adjustable threshold voltage V_{thr} is used to control to which voltage C_{syn} can be charged. This affects the limit to which C_{syn} is charged when multiple input spikes arrive after each other during the decay time of a single output spike, summing them up to V_{thr}.

The NMDA receptor is dependant on the post-synaptic membrane voltage. This implementation adds a filter that uses the membrane voltage as a threshold voltage, conducting when it rises above an adjustable reference voltage. A secondary capacitor is added to the input stage, implementing an integrating low-pass filter representing the much slower time scales of the NMDA receptor compared to the other receptors (see Table 2.1).

2.5 Implementation techniques

2.5.1 Sub-threshold operation

Low-power deep sub-micron CMOS processes target relatively high threshold voltages with respect to the nominal supply voltage. This is done to minimise leakage in the low-power processes. Threshold voltages of about half the supply voltage are typical [31, Ch. 3]. Additionally, circuits can be operated below the nominal supply voltage to further decrease power, for example in [19]. This means that in low-power neuron implementations in deep sub-micron processes many, or even all, of the transistors operate in or near the sub-threshold region, in moderate to weak inversion.

The threshold voltage V_T changes when the drain-source voltage V_{DS} creates an electric field, this is drain-induced barrier lowering (DIBL). V_T decreases by V_{DS} times the DIBL coefficient η. η is device dependant and is typically between 0,01 and 0,1. This effect results in a higher sub-threshold drain current when V_{DS} is high, also increasing unwanted leakage, and it is stronger in short-channel devices [32, Ch. 2].

In weak inversion the drain current through a MOSFET is described by

$$I_D = I_0 \exp\left(\frac{V_{GS} - V_T + \eta V_{DS}}{n V_{th}}\right)\left(1 - \exp\left(\frac{-V_{DS}}{V_{th}}\right)\right) \text{ for } V_{GS} - V_T < 0 \tag{2.29}$$

with

$$I_0 = \mu_0 C_{ox} \frac{W}{L}(n-1)V_{th}^2 \tag{2.30a}$$

$$V_{th} = \frac{kT}{q} \approx 26\,\text{mV at } T = 300\,\text{K} \tag{2.30b}$$

I_0 is the device-specific current at threshold, V_T is the MOSFET threshold voltage, n is the process-dependant sub-threshold slope factor and V_{th} is the thermal voltage.

When V_{DS} is more then approximately four times V_{th} the last term of Equation (2.29) becomes nearly 1 and the transistor is operating in saturation.

In saturation I_D can be approximated to

$$I_D \simeq I_0 \exp\left(\frac{V_{GS} - V_T}{nV_{th}}\right) \tag{2.31}$$

The transconductance calculated as $\frac{dI_D}{dV_{GS}}$ is then

$$g_m = \frac{I_D}{nV_{th}} \tag{2.32}$$

It is notable that the $\frac{g_m}{I_D}$ ratio is therefore approximately constant in weak inversion.

In strong inversion there is a linear to quadratic relation between the gate-source voltage and the drain current, depending on the saturation operating region. In weak inversion this relation is exponential however. This makes sub-threshold operation well-suited for low-power neuron implementations. The exponential relation between voltage and current fits the exponential decay seen in synapse dynamics and conductance based neuron models, as well as the exponential current term based on the membrane voltage in exponential LIF models. Sub-threshold operation allows a lower supply voltage, enabling lower power and static leakage in neuron implementations.

2.5.2 Sizing & leakage

One of the design objectives is to minimise the used area. Ideally minimum-sized transistors would be used. In advanced processes these will have significantly larger leakage currents though. This is detrimental for two reasons. Firstly, higher leakage will increase the static power consumption. Secondly, higher parasitic leakage requires larger signal currents to achieve the same time constants, especially in LIF neuron implementations, since some of the unwanted leakage will add to the intended leakage current. This leads to a higher dynamic power consumption and a larger current also requires a larger membrane capacitor value for the same integration time. This increases the area needed.

Neuron implementations use low currents, ranging from a few pA to to nA, to operate at time constants that are compatible with natural signals [26]. The drain-source leakage of minimum-sized transistors can be well within that range, [26] showing I_D is approximately 100 and 10 pA for NMOS and PMOS transistors respectively with a width of 200 nm and length of 30 nm at $V_{GS} = 0$ and $V_{DS} = 0{,}5\,\text{V}$. The leakage can be decreased significantly by increasing the transistor length. Tripling the length for these examples reduces the drain leakage by approximately 2 orders of magnitude.

2.6 Conclusions

The behaviour of a neuron cell can be described as the voltage over a capacitive membrane. This voltage changes as the result of current that is flows into or out of the neuron through ion channels of different types. The Hodgkin & Huxley(HH) model describes the current through ion channels as current through changing conductances. Different functions to model the change in conductance

over time are used for each ion channel type. Membrane voltage-dependant parameters can be fitted for the functions. Additional conductance functions can be added to model additional ion channel with different behaviour. The model is closely based on the observed biological mechanisms of a neuron cell. It is possible to describe neuron behaviour accurately, and the model can be made more detailed by fitting additional conductance functions. However, this results in the highest amount of operations to compute one timestep of the reviewed models. The parameter functions of the model are more difficult to implement as a circuit than in the other models, since they consist of more operations.

The Izhikevich model aims to be much more computationally efficient. The model can describe the spike patterns observed in biological neurons, but is not directly based on biological mechanisms like the HH model is. Although the model can describe spike patterns, it is not accurate for the shape of individual spikes, approximating them with a quadratic function.

The leaky integrate-and-fire (LIF) model is the simplest model that is in use. Like the HH model it is a conductance-based model that uses a membrane capacitor and the current through a leakage resistor, but there is only one conductance and it has a static value. Spikes are modelled as ideal delta spike that are generated when the membrane reaches a fixed threshold. It uses the least amount of operations to compute a timestep and has few parameters. It is easy to translate to an electronic circuit. However, the model does not describe the shape of spikes and can only model a few spike patterns.

The LIF model can be made more realistic by adding an exponential function that describes the shape of the spike, by modelling the change in the membrane voltage when it is around and above the threshold voltage. An exponential function is a good fit for the spike shape observed in biological neurons. The addition of this term replaces the strict voltage threshold with a wider zone in which spikes can be initiated, enabling the modelling of more spike patterns.

The effective threshold of can change due to a period with high or low spike activity. Adding a function that models the neuron adaptation due to frequency significantly increases the amount of different spike patterns that are modelled. This adaptive exponential (AdExp) IF model can then describe most observed neuron behaviour, using only a few more operations than the Izhikevich model. Because the model is conductance-based its elements can be compared to the original biological mechanisms when implemented as electronic circuits.

Neuron implementations of the LIF and AdExp IF models have been compared. A simple LIF implementation [19] consumes the lowest amount of energy per spike of all implementations that are reviewed. It only needs three control signals, making it easy to bias. However, it uses a larger area than other neuron implementations.

A more complicated LIF implementation [20] adds filtering to the synapse input, making it more accurate. This circuit uses less area. An implementation of the same circuit in 65 nm [3] shows that, although very low area can be achieved, the energy per spike is higher than in other implementations.

Multiple circuits based on the AdExp IF model [26], [28] use a differential pair integrator to implement the exponential term of the model. An additional transistors and a capacitor compared to a simple LIF circuit are added to implement the frequency adaptation. This increases the total area of some implementations, most of the area is used by the capacitors. However, this enables these circuits to implement most spike patterns. These implementations are able to achieve a low energy

consumption. The implementation of an AdExp-based neuron in the same 65 nm process as above [3] shows that it can achieve a good balance between energy consumption and area.

MOSFETs operating in weak inversion have an exponential relation between the gate voltage and drain-source current. This can be used to implement the exponential terms of an AdExp IF model. Operating the neuron circuit in weak inversion allows for a lower supply voltage to be used, reducing energy consumption. However, care must be taken to reduce sub-threshold leakage, and the circuit can be more sensitive to mismatch and temperature effects.

3 Implementation

In this chapter the implementation of the analog computational elements is explained. The main elements are the presynapse, synapse and neuron. First the general network structure is discussed, then the circuit level description of the individual elements, and finally the integration of the components in an SOC.

3.1 Structure

Figure 3.1 shows the structure of the analog network [33]. The network receives input spikes in the form of digital pulses. These are converted into analog exponentially decaying spikes, in order to take have a spike that is closer to the biologically inspired models. This is done in the presynapses. The output of a presynapse is a current. This current is replicated at the input of every synapse in the same row as the presynapse. The synapse contains a current-steering DAC and the storage of a weight to set the DAC. The current is put into the DAC, and based on the weight the DAC outputs between 0 and 100 % of the input current. The output currents of all the synapses in a row are summed and sent to the input of a neuron. Based on this input current the neuron fires a spike when a threshold has been reached. This neuron output spike is buffered so it becomes a clean digital pulse, and can again be used as an input to the network.

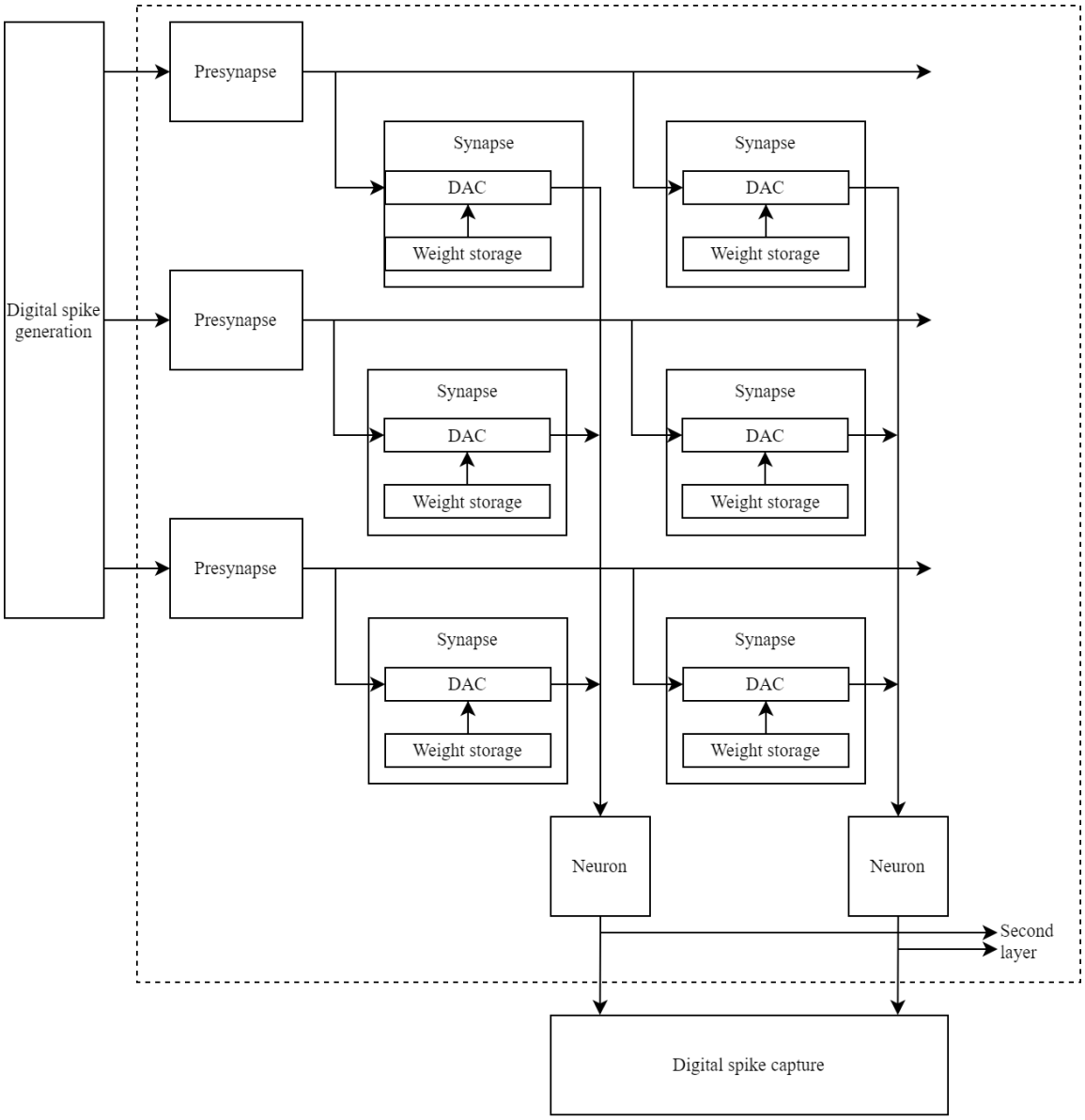


Figure 3.1: Network

3.2 Presynapse

3.2.1 Possible improvements to state-of-the-art

The synapses as described in Section 2.4 use an analog voltage to set the weight that is applied. The integration current is set directly controlled by the weight. This structure has a disadvantage. Because the weight is applied on the input side of the synapse circuit, the complete synapse needs to be replicated for each crossbar connection.

Within the synapse circuit, most of the area is used by the integration capacitor. Putting this capacitor at each crossbar connection uses a large amount of area.

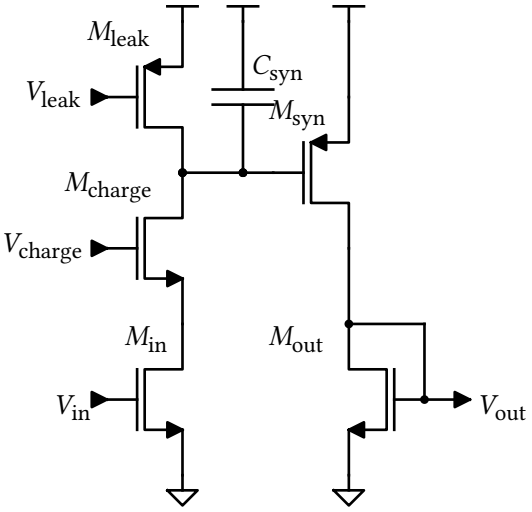


Figure 3.2: Circuit of simple AMPA presynapse

Additionally, these capacitors needs to be charged and discharged for each spike that goes through that synapse. The charge current of the capacitor is a large part of the power consumption of the component.

These problems can be solved by applying the weight to the output of the synapse, instead of to its charge current. The synapse is split into a presynapse and a synapse. The presynapse converts a short binary input pulse from an input neuron or digital spike generator into an exponentially decaying analog spike. It uses the structure from Figure 2.6a as its base for this, this operation corresponds to the behaviour of an AMPA receptor in biological neural systems.

The synapse stores a weight and applies this to the analog spike coming from the presynapse. This allows to use a single presynapse for an entire row of synapses, instead of replicating the spike generation circuit in each synapse.

3.2.2 AMPA-based

Two variations of the presynapse have been designed and taped out. The first is a linear-charge-discharge based synapse, shown in Figure 3.2. This synapse implements the spike conductance as seen through a simple AMPA receptor model, with an slowed-down, exponential output current

A charge current, controlled by V_{charge}, is integrated on the capacitor C_{syn} while the input pulse in active. Due to the constant leakage current controlled by V_{leak}, C_{syn} discharges linearly. M_{syn} always operates in sub-threshold. Therefore, the linear decrease in the voltage over C_{syn} will result in an exponential decrease in the output current. This produces the exponentially decaying analog current spike.

In contrast to the synapses in Section 2.4, no additional input filter is implemented before the active-pulse input integration. The adaptive-exponential neuron circuit includes this mechanism already in the form of the DPI input leakage. Secondary low-pass filtering in the synapse is useful for synapses that include long-term plasticity and STDP circuits for self-learning behaviour. However, these circuits are not included in the scope of this implementation.

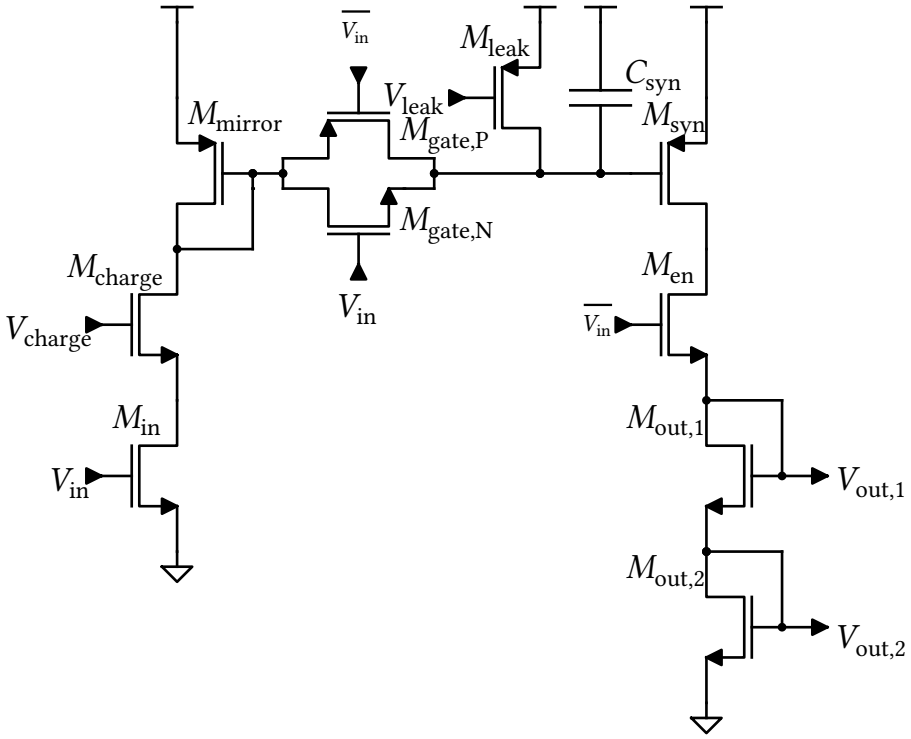


Figure 3.3: Circuit of improved presynapse

This circuit is very sensitive to variation. Due to the exponential nature of the sub-threshold operation of M_{syn}, a small variation in the voltage over C_{syn}, due to variations in its capacitance value or mismatch in the charge current mirror, will produce a much larger variation in the output current. Additionally, variation in the capacitance and charge current can cause the voltage over C_{syn} to get too large, which causes M_{syn} to operate out of sub-threshold.

3.2.3 Improvements

The second version is a more sophisticated iteration, shown in Figure 3.3. The charge current is still controlled by V_{charge}. M_{mirror} and M_{syn} form a current mirror together. When the input pulse at V_{in} is active, the transmission gate formed by $M_{gate,N}$ and $M_{gate,P}$ is switched on and the voltage across C_{syn} will stabilise to the current mirror voltage. Due to M_{en}, the output of the presynapse is not yet enabled at this moment. When V_{in} goes low, the output of the presynapse is enabled. The initial output of the presynapse will be equal to the current set by the current mirror. As M_{leak} discharges C_{syn} over time, the output current will decrease accordingly.

3.3 Synapse

The synapse consists of four parts: input current generation, weight storage, the DAC and sink-source selection. The input current for the DAC is generated by M_{in}. This transistor is sized identically to M_{out} of the presynapse, forming the other half of the current mirror. This current is sent into a current steering DAC consisting of current splitters[34], and based on [35].

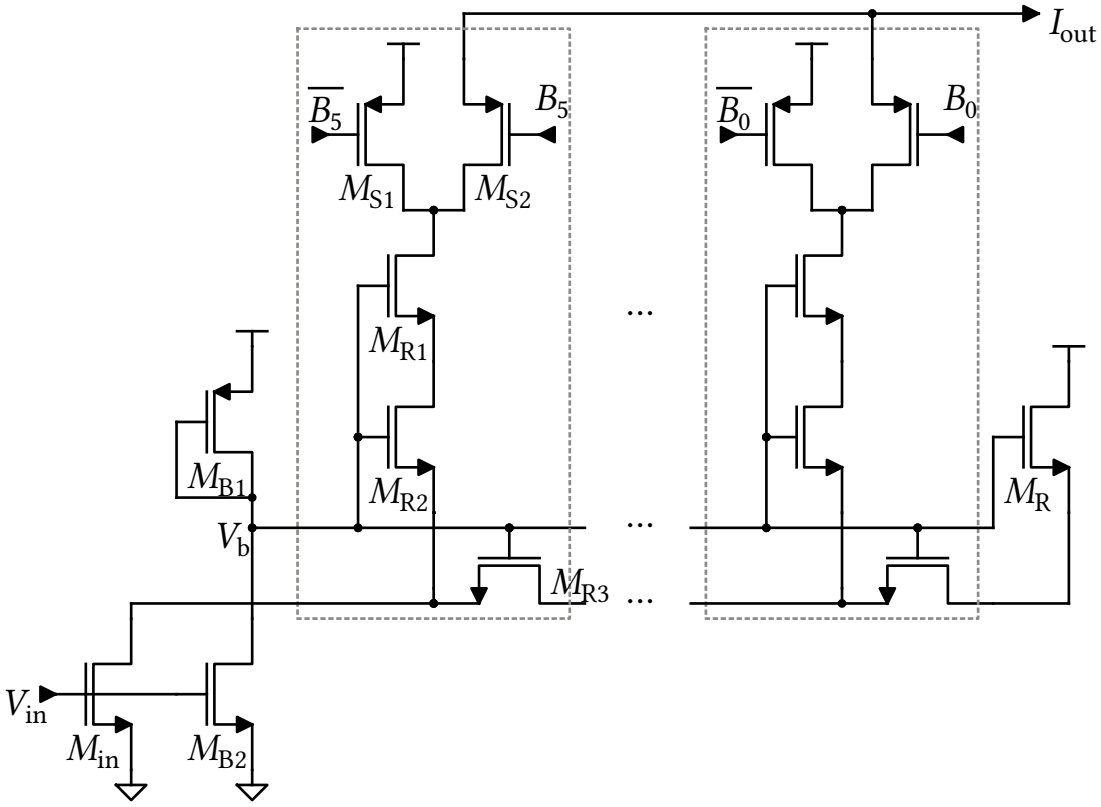


Figure 3.4: Circuit of the input current generation and the DAC

The DAC is an R-2R style DAC. The circuit is shown in Figure 3.4. The current-steering DAC applies a factor ranging from 0 to 1 on its input current, attenuating it and sending the selected portion of the input current to its output. Because this is a current-steering DAC, the outputs of multiple synapse can easily be summed.

The advantage of this topology is that its area scales linearly with the number of bits in the DAC. This in contrast to other DAC topologies, where the area often scales exponentially with the number of bits. The resolution of the DAC is dependant on the leakage and mismatch in the least significant bit. An additional current splitter can be added for an additional bit, however it might be necessary to increase the area of all bits to improve leakage and mismatch performance. It was determined that for this neuromorphic network implementation a 6-bit DAC provides enough resolution, with additional bits not significant increasing the recognition rate.

For constant biasing of the resistive transistors M_{R1}, M_{R2} and M_{R3} across all the current splitters it is important that V_b is significantly above the source voltage of each M_{R2} and M_{R3}. This is achieved by copying the input current through a diode-connected transistor. This ensures that the biasing voltage V_b scales with the voltage drop across the current splitters.

3.4 Neuron

The neuron circuit is shown in Figure 3.5. There are five main parts to the neuron, each of which corresponds to a biological function in the conductance model. These are the central membrane

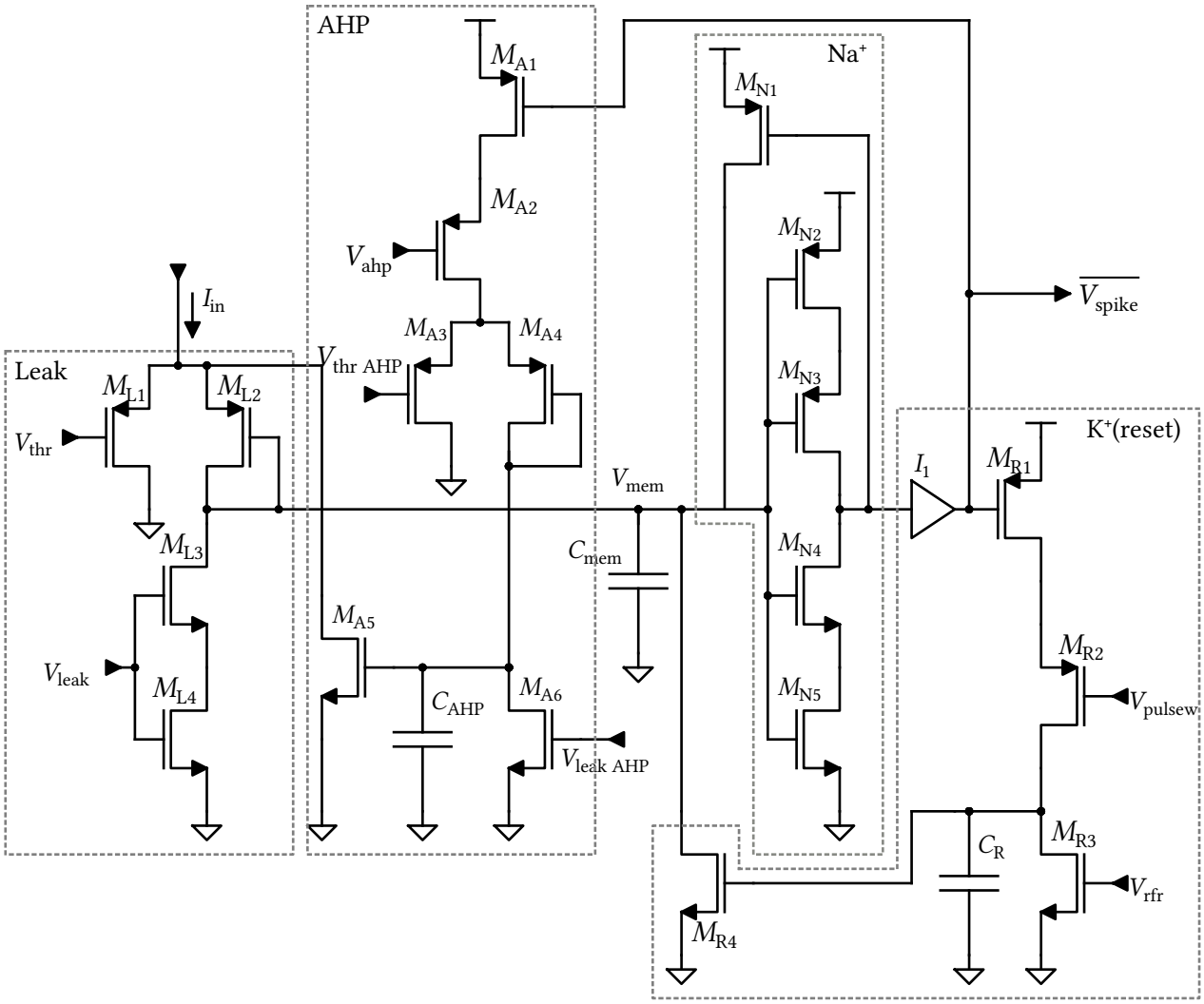


Figure 3.5: Circuit of conductance based neuron

capacitance, leakage, afterhyperpolarisation (AHP), sodium activation and potassium reset.

3.4.1 Membrane capacitance

This capacitor is the heart of the neuron circuit. Spikes from the neuron input are integrated on the capacitor. Because the spikes are current-controlled, this results in a changing voltage across the capacitor. This voltage corresponds to the membrane voltage of the neuron model.

The size of the membrane capacitor has a large impact on the rest of the design, since it influences both the integration rate and the leakage time constant. A size of 100 fF was chosen. This makes currents in the nanoampere-range possible with spike frequencies in the tens to hundreds of kilohertz.

In principle a capacitance that is as low as possible is desired. As the capacitance becomes lower, the current that needs to be integrated on it in order to achieve the same effect also becomes lower. If the currents become too low, they will become too susceptible to noise and interference. Leakage in

the transistors will also start to become a significant part of the signal currents. These leakages can vary with multiple orders of magnitude across process corners and temperature. In leaky corners, the leakage of a 120n/1u transistor can be around 100 pA. To limit the influence of transistor leakage on the normal circuit operation the signals are kept at least an order of magnitude larger than this, putting a lower limit on how small the membrane capacitor can be. Another factor is that in a smaller capacitor, manufacturing variance can have a larger effect on the capacitance value. Variation in the capacitance value would directly influence the frequency variability of the output spikes.

3.4.2 Activation

The activation corresponds to the changing conductance due to sodium ions in the conductance model. This section is responsible for the spike generation. It consists of an inverting amplifier and a positive feedback element.

$M_{\mathrm{N2-5}}$ are operating as a low-leakage inverter. Once the voltage on the membrane node reaches the threshold voltage of this inverter, its output voltage will go down, but due to the long length of the inverter this does not happen instantaneously. However, once inverter output voltage is going down, feedback transistor M_{N1} becomes active. This results in a positive feedback loop, where the current through M_{N1} cause V_{mem} to go up, which results in the output of the inverter going down further, which increases the amount of current going through the feedback transistor even more. This leads to a sharp spike at not only the output of the inverter, but also at V_{mem} itself. Without this feedback mechanism, V_{mem} would not change while the output of the circuit spikes.

3.4.3 Reset

Once the output signal is pulled high, the circuit needs to reset to form this into a spike, instead of just a step function. This is the function of the reset section of the circuit, which implements the function that corresponds to the changing conductance due to potassium ions in the conductance model.

The reset mechanism consists of a capacitor, current sources to charge and discharge this capacitor, a switch to enable the charging of it, and a pull-down transistor connected to V_{mem}. The main function of the reset is to pull V_{mem} down to ground after a spike event has happened. This is done by increasing the voltage across C_{R} above the threshold of M_{R4}.

When a spike occurs and V_{mem} is set high, M_{R1} is switched on. This allows the current source M_{R2} to charge C_{R} with a constant current. Once the voltage over C_{R} reaches the threshold of M_{R4}, V_{mem} is pulled low and the spike is ended. The amount of charge current thus determines the length of the spike.

Once the spike has ended in this way M_{R1} is switched off. Since C_{R} is no longer being charged, it is discharged with a constant current by M_{R3}. This discharge current is always on, but is much smaller than the charge current. When the voltage over C_{R} goes below the threshold of M_{R4} V_{mem} is no longer pulled to ground. It can then start integrating spikes again. The period that V_{mem} is pulled down is the refractory period. During this period no spikes can be generated. The length of the refractory period is controlled by the amount of discharge current.

3.4.4 Leakage

(Intentional) leakage is implemented in two places. First there is a differential-pair integrator (DPI) filter on the input. This differential pair implements an exponential filter based on the difference between the external parameter V_{thr} and V_{mem}. The smaller the difference, the more input current is filtered out. This implements an exponential behaviour where a large spike is needed to put the membrane over its threshold. This dynamic behaviour is observed from biological synapse connections.

Secondly, a leakage current is directly subtracted from the membrane node. This is a constant current that can be set as one of the parameters to control the neuron behaviour.

3.4.5 Afterhyperpolarisation

Afterhyperpolarisation (AHP) is a phase after the spike has occurred. In the context of this work AHP refers to the slow afterhyperpolarisation. This is a mechanism that changes the input conductance based on previous spike activity. If a period of high spike activity has occurred, the AHP will make it more difficult for a new spike to be generated. It does this by filtering out part of the input current.

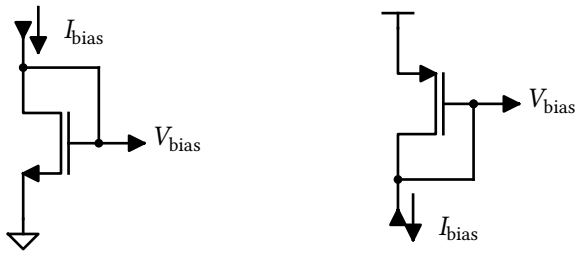
The AHP works in a similar fashion to the reset circuit. When an output spike is generated, M_{A1} is switched on and capacitor C_{AHP} is charged a bit by a constant current from M_{A2}. If no further activity occurs C_{R} is discharged through its leakage current from M_{A6}.

The DPI filter in the charge current implements a logarithmic dynamic in the activation of the AHP, making it so that exponentially more spike activity is needed to increase input current filtering.

This mechanism is important for the self-synchronisation of multiple neurons in a network.

3.5 Biasing

The presynapse and neuron use analog control currents to adjust their circuit parameters. These are supplied by current sources operated as a distributed simple current mirror. A reference control current is supplied through a diode-connected mosfet, shown in Figure 3.6. The resulting gate/drain voltage is distributed as a biasing voltage to identically sized mosfets in the local circuits. This is implemented for both sourcing and sinking control currents.



(a) NMOS (b) PMOS

Figure 3.6: Biasing circuits

The biasing diodes and corresponding connected current sources are sized to have a long channel but to have minimum widths. A gate that is as large as possible is desirable to negate the effects of manufacturing variability. However, a wide gate would result in a very low overdrive voltage. This due to the low currents used for most control currents, in the range of 0,5 nA to 50 nA.

The biasing could be improved by using current-distribution with a many-branched mirror instead of distributing the diode voltage. This was not done due to routing constraints. Designing an architecture for biasing generation and distribution is outside the core scope of this work.

3.6 Layout

3.6.1 Floorplan

The presynapse, synapse and neuron are designed to all exactly abut. The synapses are placed in a rectangular grid. A presynapse is placed adjacent to each row and a neuron at the end of each column. Half of the rows, in an alternating pattern, is mirrored in the vertical dimension. This allows the NMOS and PMOS sections of the presynapses and synapses to abut cleanly. This layout is shown in Figure 3.7.

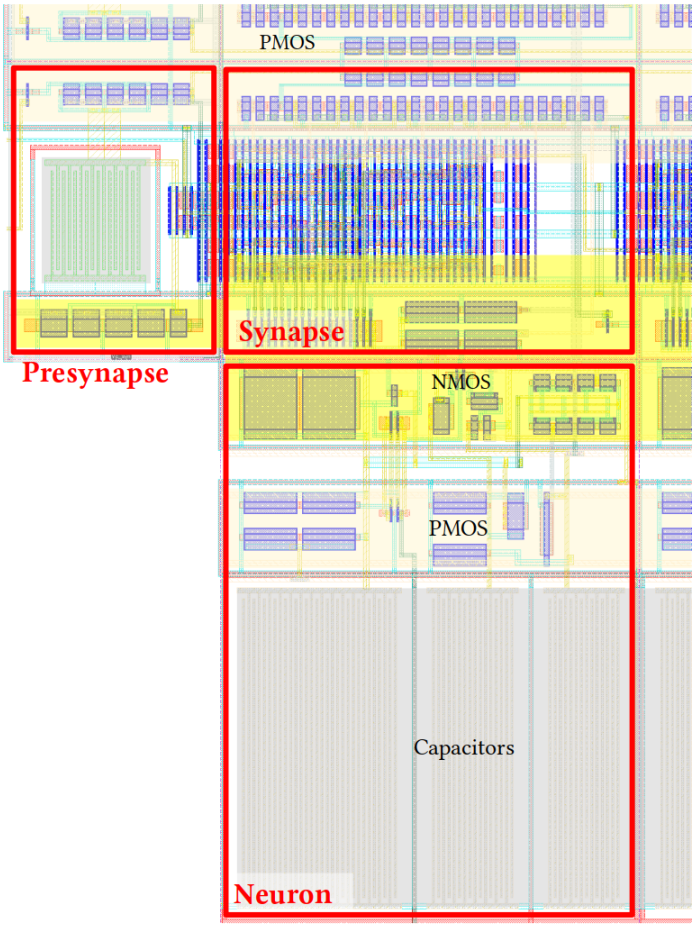


Figure 3.7: Layout of a presynapse, synapse and neuron in an array

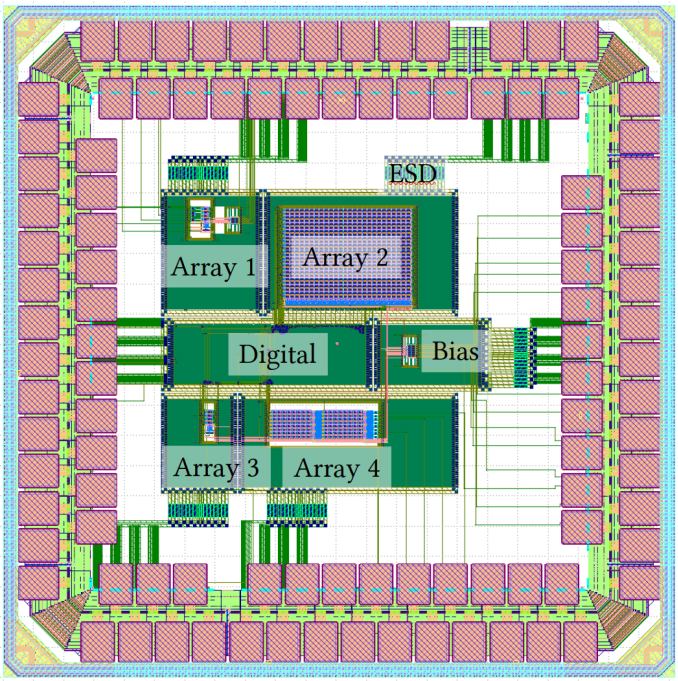


Figure 3.8: SOC layout

Each core element is surrounded by a guardring. The layout of the elements is such that the PMOS area of one abuts with the PMOS area of the next. This results in a continuous NWell between them, which removes the need for additional space in between elements. The PMOS and NMOS areas are separated by empty space to avoid parasitic effects due to the edge of the NWell.

Spikes are injected in the network at the left side from a digital buffer into the presynapse. The spike from the presynapse output is distributed along the entire row of synapses next to it. The spike is distributed on the M6 layer, which is horizontal and above the routing layers of the synapse. The output spikes from all synapses in a column are connected to a vertical M5 trace running the entire height of the array. It connects to the neuron on the bottom of the array, where the spikes are integrated. Output spikes from the neuron are routed from the bottom of the neuron to the digital domain where they are captured.

3.6.2 SOC integration

The neuromorphic circuits are assembled in four test arrays on a chip, see Figure 3.8. In the center of the chip is a digital controller. A central bias block distributes bias currents to the neuromorphic blocks. Each block has its own power domain.

Array 1 is an isolated block which consists of 4 presynapses, 4 synapses and 1 neuron. The input and output signals are directly connected to the chip's pads, isolated from the rest of the chip. This allows for analog characterisation of the components without going through the digital controller. The power consumption and timing of the spiking output can be measured.

The input and output signals of the other arrays are controlled through the digital controller. This controller block generates spikes signals for the circuits from an external spike interface and cap-

tures the output spikes that are generated by the neurons. The controller also sets the weights of the synapses in the arrays through a serial interface. It functions as an interface to an external microcontroller or FPGA.

Array 3 has the same structure as array 1, but is not isolated and interfaces with the digital controller instead. This array can be used to verify the basic spike generation, spike capture and weight setting functionality of the controller, as well as other interactions between the analog neuromorphic circuits and the digital circuits.

Arrays 2 and 4 are larger neuromorphic arrays. Array 2 consists of a neural network matrix with 26 presynapses, 26 neurons and 676 synapses. This array can be used to characterise the variability of synapses and neurons in an array. It can also be used to test small-scale, single-layer inference.

The last array consists of two networks with 8 presynapses, 8 neurons and 64 synapses connected after each other. This can be used to test a two-layer network. If both blocks are biased identically, a feedback network into the original layer can be simulated. Otherwise, the effects of a hidden layer can be observed in this network, an essential element in improving the accuracy of neural networks.

4 Simulation & characterisation

The neuron, presynapse and synapse are simulated in an analog simulator. Their functionality is verified in different process corners. Key parameters are characterised with Mont-Carlo simulations.

4.1 Presynapse

4.1.1 Functionality

4.1.1.1 Basic presynapse

This section shows the simulated results of the implementation of the circuit of Figure 3.2. Figure 4.1 shows the input voltage pulse signal and output current of the presynapse. The voltage at V_{syn} is shown in Figure 4.2.

$V_{GS,syn}$ of the output transistor M_{syn} is the voltage over the capacitor C_{syn} and is $0{,}8\,V - V_{syn}$. V_{syn} is kept above 0,4 V, ensuring that $V_{GS,syn}$ is below the (PMOS) threshold voltage of (also) 0,4 V. Therefore M_{syns} is always in weak inversion and the exponential relation between V_{syn} and the output current holds. It can be seen that the discharge of V_{syn} is linear until when $V_{GS,syn}$ becomes less than approximately 70 mV. This is because the discharge transistor M_{leak} is no longer in saturation at that point and the '$\left(1 - \exp\left(\frac{-V_{DS}}{V_{th}}\right)\right)$' portion of Equation (2.29), in which $V_{GS,syn}$ is V_{DS}, then

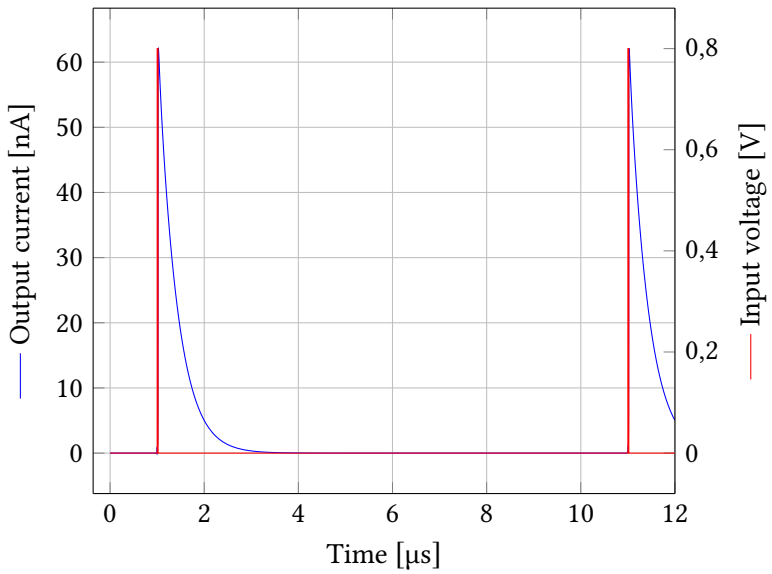


Figure 4.1: Output current spike and input pulse signal of the basic presynapse

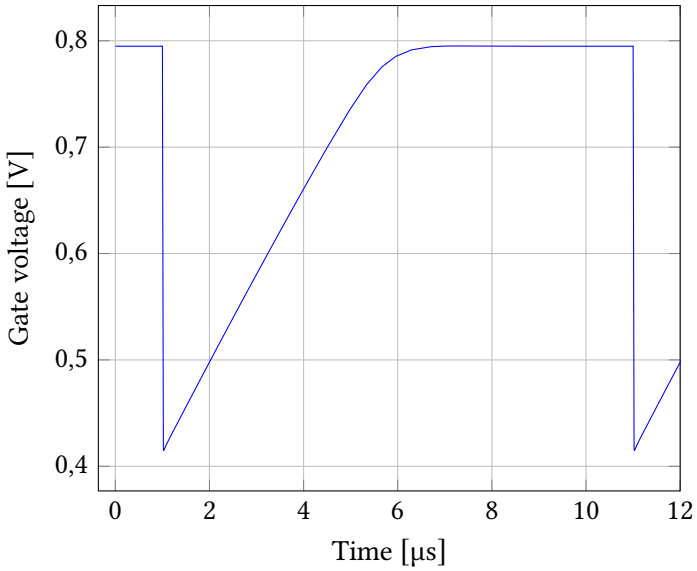


Figure 4.2: Gate voltage of the output of the basic presynapse

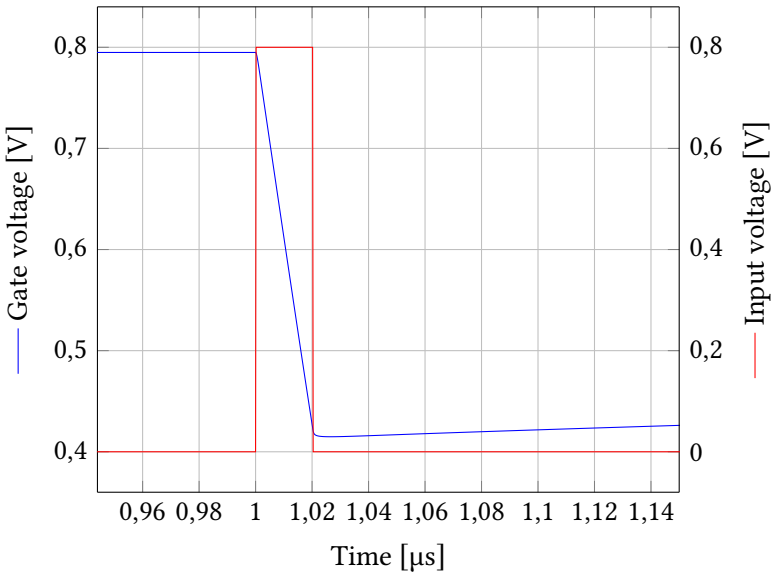


Figure 4.3: Closeup of input pulse signal and charging of output gate voltage of the basic presynapse

significantly lessens the expected current. However, because the output current is already so low at this point due to the exponential relationship this soft bend of V_{syn} is not noticeable in practice in the output current.

Figure 4.3 shows a closeup of Figure 4.2 during an input pulse. The voltage over C_{syn} increases while the input pulse is high. The amount of change in V_{syn} is dependent on the pulse width of the input pulse.

The average power consumption was determined while the presynapse has no input activity (static power) and for a spike rate of 100 kHz. The difference between the active and static power is used

Table 4.1: Basic presynape power specifications

corner	average power [nW]	static power [nW]	energy/spike [fJ]	**charge/spike [fC]**
nom	6,6948	4,2423	24,53	**45,38**
ss	1,6687	0,9169	7,518	**2,841**
ff	58,4520	21,6621	367,90	**895,65**

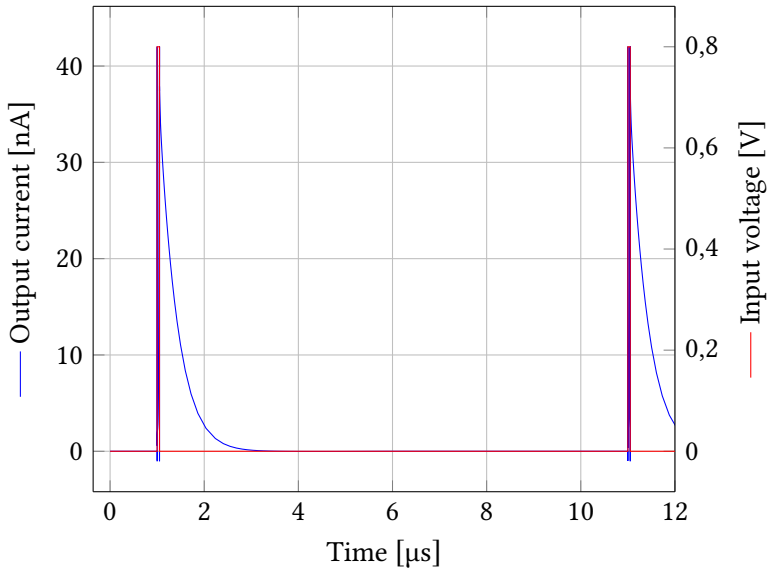


Figure 4.4: Output current spike and input pulse signal of the improved presynapse

to compute the energy per spike. These are reported in Table 4.1 for the typical process corner as well as the extreme fast and slow corners.

The other important metric for the presynapse is the charge per spike that is received by the synapse. The amount of charge represents the weight of a spike. It is influenced by the peak amplitude of the spike, the decay time, and the exact shape of the exponential decay. There is a large amount of variation in the charge per spike for different process corners. The main objective of the improved presynapse design is to reduce this variation in the amount of charge per spike.

4.1.1.2 Improved presynapse

Figures 4.4 and 4.5 show the same signals as in Figures 4.1 and 4.2 for the improved presynapse. The discharge starts when the input pulse is zero and happens in the same way as in the basic presynapse described before.

In contrast to the basic presynapse, the output current is not active during the charging phase, when the input pulse is high. Instead, the output spike starts on the falling edge of the input pulse. This makes the amount of charge independant of the input pulse width.

Figure 4.6 shows a closeup of the input pulse signal and the start of the output current spike. Similarly to the presynapse charging in Figure 4.3, V_{syn} starts by decreasing linearly due to charging of C_{syn}

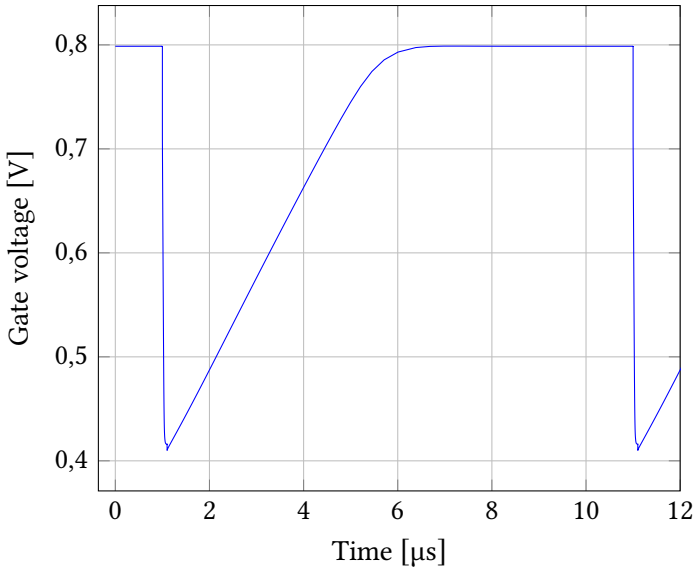


Figure 4.5: Gate voltage of the output of the improved presynapse

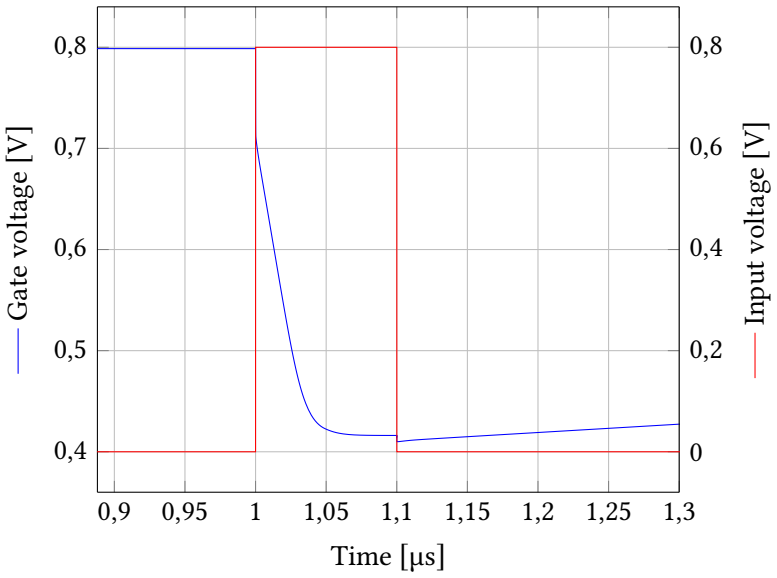


Figure 4.6: Closeup of input pulse signal and start of output current spike of the improved presynapse

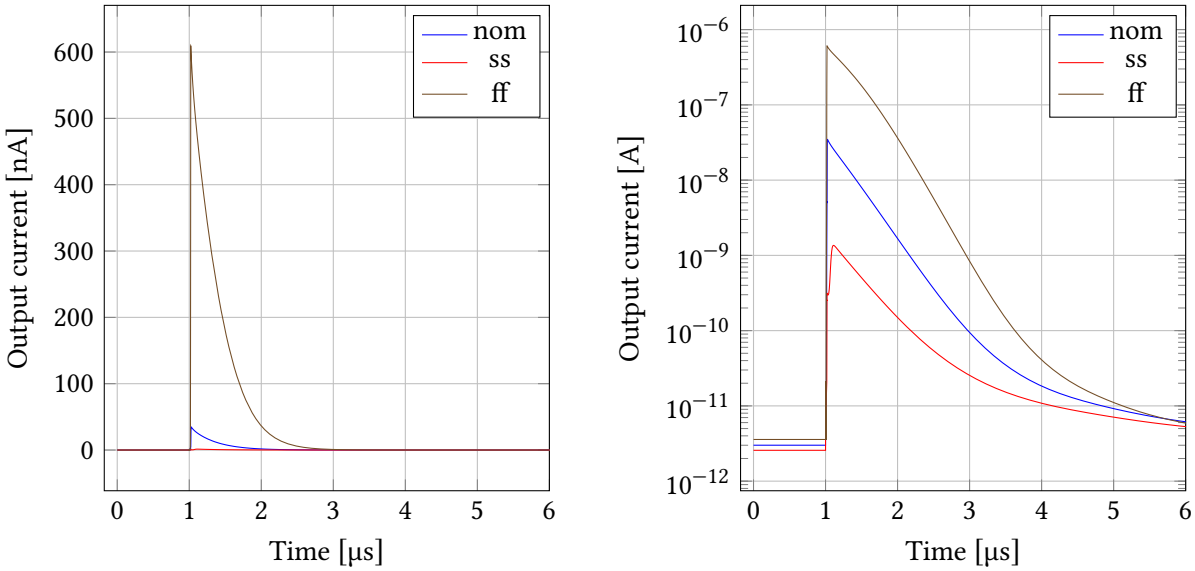
with the charge current I_{charge}. Due to the decrease of the gate voltage, M_{mir} starts to conducting and equilibrium is reached at the diode voltage of M_{mir} corresponding to the charge current.

This can be seen in Figure 4.6, where V_{syn} stabilises after about 80 ns. The voltage to which C_{syn} is charged, determining the peak amplitude of the output spike, is therefore no longer dependant on the duration of the input pulse, as long as it is longer than a minimum required duration.

Table 4.2 lists the power consumption for a spike rate of 100 kHz. Both the static and average power are higher than those of the basic presynapse. The energy consumed per spike is higher as well, except for the fast process corner. However, the amount of charge in each output spike is much

Table 4.2: Improved presynape power specifications

corner	average power [nW]	static power [nW]	energy/spike [fJ]	**charge/spike [fC]**
nom	10,7336	6,3692	43,64	**136,25**
ss	5,7534	1,2430	45,10	**164,94**
ff	42,2512	38,1833	40,68	**108,17**



(a) Linear scale

(b) Logarithmic scale

Figure 4.7: Basic presynapse output current

more consistent.

4.1.2 Process variation

The output current for the typical, slow-slow and fast-fast corners of the basic presynapse is shown in Figure 4.7. Both the amplitude and the time constant of the exponential decay vary with corner. There is a very large difference in spike amplitude visible between the process corners. This results in the wide variation in charge per spike in Table 4.1. The output spike is so small in slow corner that it is barely visible in Figure 4.7a. Figure 4.7b shows the spike current in logarithmic scale instead.

The same is shown in Figure 4.8 for the improved presynapse. It is visually obvious that the variability in output current is much smaller than in Figure 4.7. There is some variation in peak current, but the total amount of charge per spike—the area under the curve—is largely the same.

A Monte-Carlo simulation was run to analyse process variation using a 500 point Low-Discrepancy Sequence. The statistics are in Table 4.3. The exponential variability in the basic presynapse circuit results in a high variability. The improvements in the circuit bring the process variability to less than 5 %. Only the relative difference in mean value between the two circuits is important. The absolute amount of current can be scaled by adjusting the I_{charge} parameter or by changing the mirror ratio between the presynapse output and synapse input.

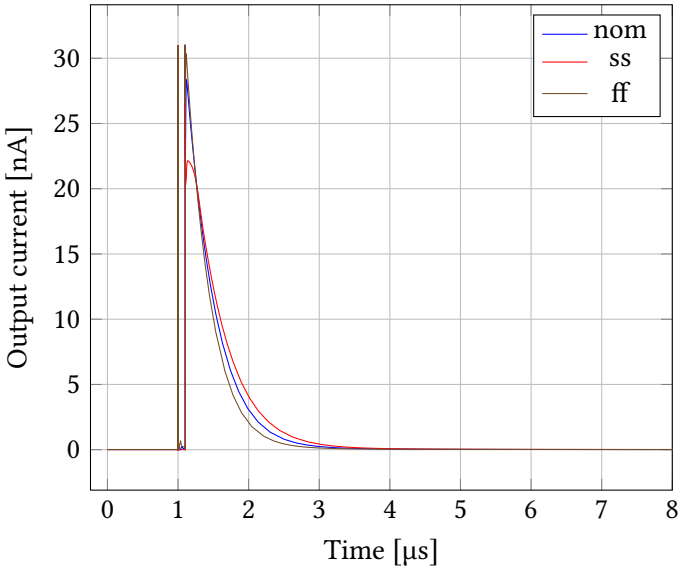


Figure 4.8: Improved presynapse output current

Table 4.3: Presynape spike charge

Presynapse	mean [fC]	min [fC]	max [fC]	std. dev. [fC]	variability [%]
Basic	60,64	8,75	590,28	53,54	88,29
Improved	136,24	114,53	157,37	6,433	4,72

4.1.3 Parameters

The presynapse is controlled by two bias currents. These are I_{charge}, the amount of current with which the presynapse is charged, determining the amplitude of the output spike, and I_{leak}, the leakage current at the output gate, determining the decay time of the output spike.

Increasing I_{leak} shortens the duration of the current spike. The effect of increasing I_{charge}, increasing the current spike amplitude, is shown at the neuron in Figure 4.13.

4.2 Synapse

4.2.1 Linearity

The linearity of the synapse DAC is characterised by sinking a constant current of 160 nA in its input and measuring the output current of the synapse for each DAC code. See Figure 4.9.

INL and DNL are shown in Figures 4.10 and 4.11. DNL is plotted as steps away from code 0 in both directions. INL is with reference to a line that goes through 0 and is fitted to the values at weights -31 and 31. For the nominal corner this comes to a step size of 1,82 nA/bit.

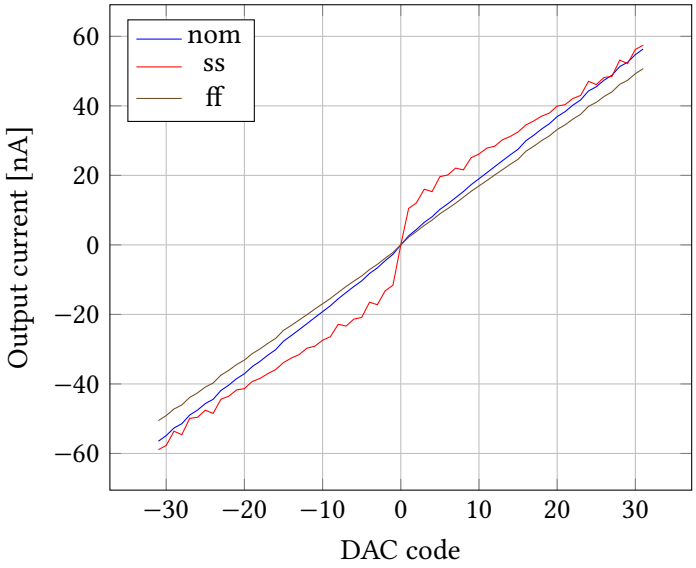


Figure 4.9: Synapse output current for an input current of 80 nA

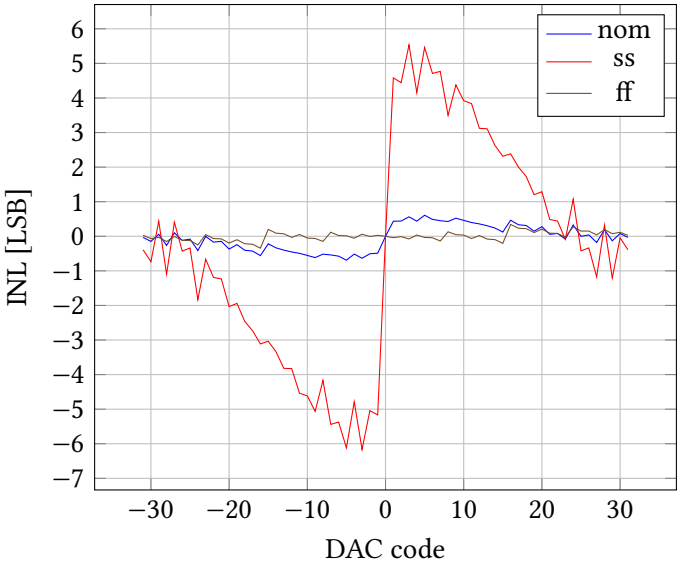


Figure 4.10: Synapse INL

4.2.2 Process variation

In the nominal and slow corners the INL and DNL are within 1 LSB. However, the fast corner shows a larger deviation at lower weights. This is due to the increased leakage in the PMOS R-2R branches. When only the branches corresponding to lower weights are connected to the output, the (increased) leakage current of those branches becomes significant compared to the intended output current.

Table 4.4 shows a statistical analysis of the process variation for various weights. 1 LSB corresponds to a mean current of 1,82 nA. The standard deviation of the expected output current of almost all

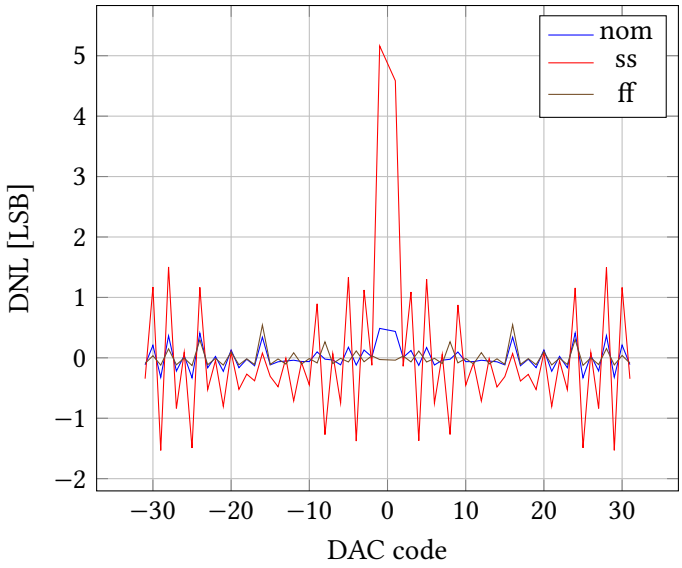


Figure 4.11: Synapse DNL

Table 4.4: Synapse output current

weight	mean [nA]	min [nA]	max [nA]	std. dev. [nA]	variability [%]
3	6,535	5,165	8,647	0,594	9,09
15	27,51	24,35	30,73	1,077	3,91
31	58,28	50,56	61,56	1,847	3,17

weights in Table 4.4 is below 1 LSB, only at the highest weight of 31 it is slightly above it.

4.3 Neuron

4.3.1 Functionality

Figure 4.12 shows the accumulation of spike charge on the neuron's membrane capacitor. When the threshold at 0,4 V is reached a spike is generated, causing the membrane potential to go to the supply voltage and then reset to 0. After a refractory period during which the voltage is kept at or near 0, the accumulation of incoming spikes starts again.

It can be seen that a spike causes a larger increase in membrane voltage when it is closer to 0, and a smaller increase when it is close to the threshold. This is the effect of the exponential filter of the input DPI circuit.

The output frequency is characterised for multiple synapse input spike rates in Table 4.5. Four synapses are connected to the neuron. A periodic input spike current is sent to the synapses. The weights are set such that the nominal output rate of the neuron is approximately equal to the input rate of the individual synapses, so four input spikes result in one output spike. A Monte Carlo simulation shows the effect of process variation on the output frequency. An average variability of 19,5 % is achieved. This falls in between the variability that was achieved by comparable implementations

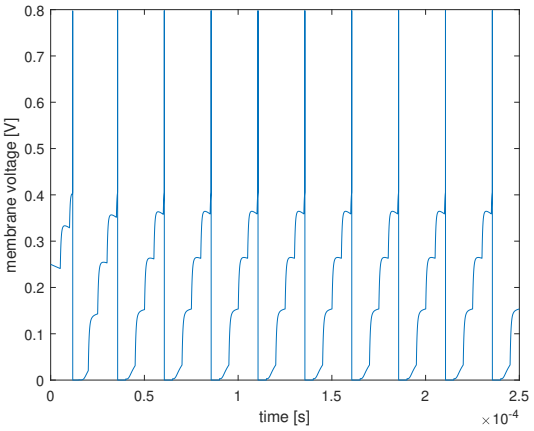


Figure 4.12: Accumulating spike charge on the neuron membrane node

Table 4.5: Neuron output frequency

input freq. [kHz]	mean [kHz]	std. dev. [kHz]	variability [%]
20	19,18	2,53	13,2
50	47,01	6,72	14,3
100	83,59	21,03	25,2
150	132,00	27,61	20,9
200	162,30	39,12	24,1

as discussed in Table 2.3.

4.3.2 Parameters

The amount of accumulation depends on the amount of charge in a spike, representing it's weight. Figure 4.13 shows the integration of spikes with increasing amplitudes on the neuron membrane. A higher amount of charge results in a smaller increase and a longer integration period in the neuron.

The pulse width of the output spike of the neuron can be increased by decreasing the biasing current I_{pulsew}. The effect of changing I_{pulsew} is graphed in Figure 4.14. This inverse relations limits the useful range of the pulse width adjustment: when I_{pulsew} is too small the sensitivity of the pulse width becomes too high, making it difficult to adjust it accurately. If a very small I_{pulsew} is desired, the large I_{pulsew} that necessary increases the amount of power that is consumed.

The refractory period inhibits the rise of the membrane voltage. Figure 4.15 shows the effect of I_{rfr} on the spike integration. Pike integration is stopped for multiple microseconds before the membrane voltage is able to rise again. A higher I_{rfr} decreases the refractory period.

Increasing I_{ahp} makes the effect of an output spike on the afterhyperpolarisation current larger, shown in Figure 4.16. When I_{ahp} is small, a period of activity does not result in an inhibition of activity. Increasing I_{ahp} reduces the time until the inhibition due to the AHP current happens.

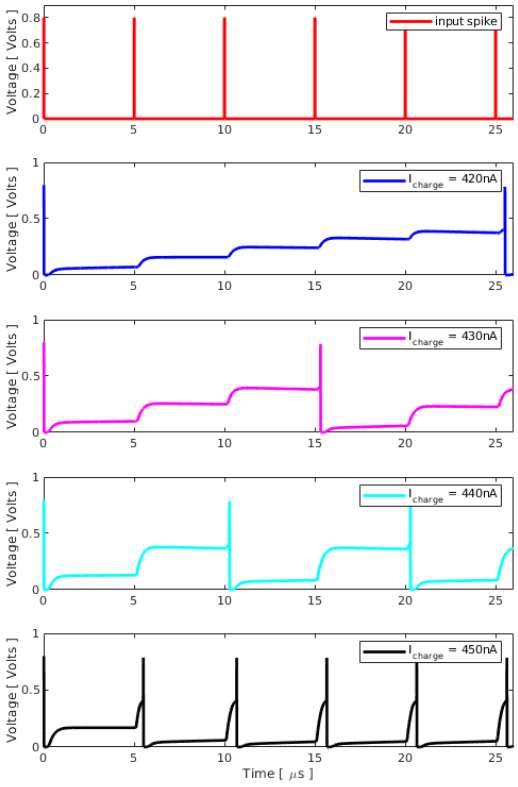


Figure 4.13: Effect of $I_{\text{syn\_charge}}$ on accumulation period

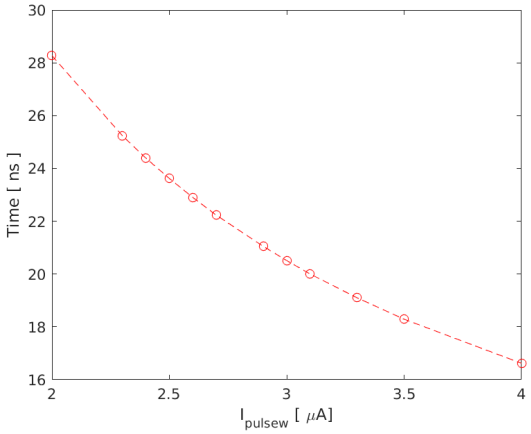


Figure 4.14: Effect of I_{pulsew} on output pulse width

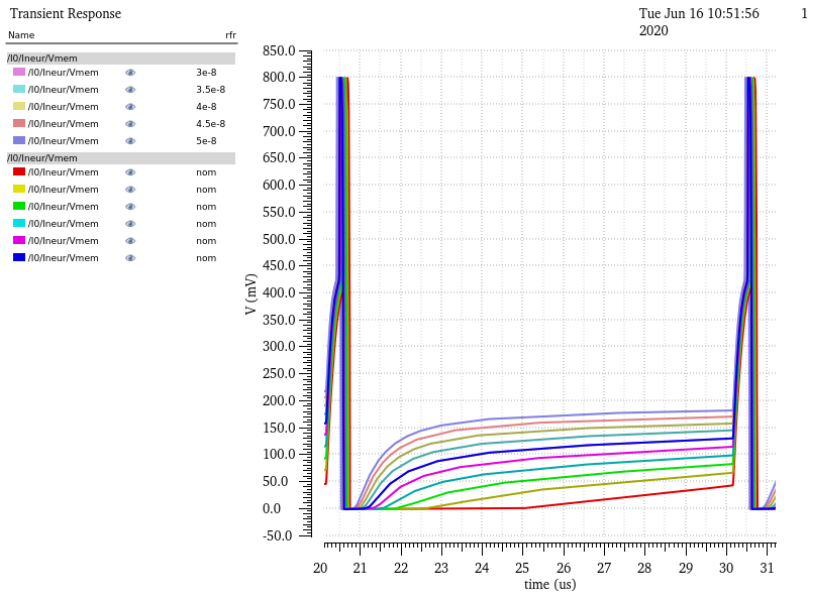


Figure 4.15: Effect of I_{rfr} on refraction period

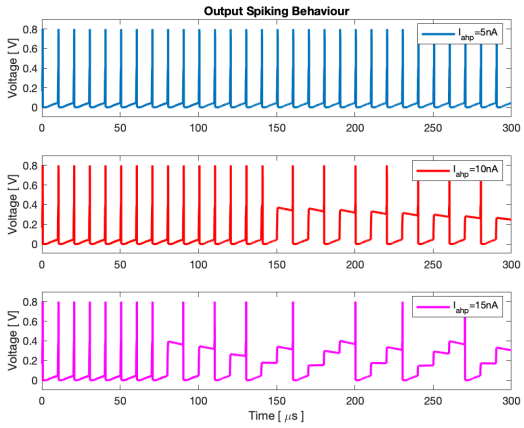


Figure 4.16: Effect of I_{ahp} on afterhyperpolarisation period

4.3.3 Power

The power usage of the neurons has a static component, due to biasing and leakage currents, and a dynamic component, dependent on the spike rate. The dynamic power usage is characterised as the amount of energy per input spike for the synapses, and energy per output spike for the neurons. The energy per spike can be calculated as

$$E_{\text{spike}} = \frac{\int_{t_0}^{t_1} V_{\text{DD}} \cdot I(t) - P_{\text{static}}\,\mathrm{d}t}{f_{\text{spike}} \cdot (t_1 - t_0)} \tag{4.1}$$

where V_{DD} is the supply voltage and $I(t)$ is the current through the supply voltage source.

The energy per spike was determined at different spike rates, shown in Table 4.6. The setup for

Table 4.6: Energy per output spike

input freq. [kHz]	mean [fJ]	std. dev. [fJ]	variability [%]
20	82,39	49,03	59,5
50	266,7	61,50	23,1
100	354,1	95,27	26,9
150	341,8	70,80	20,7
200	344,2	45,87	13,3

this measurement is equal to that as described in Section 4.3.1. The average energy per spike is nearly constant at higher input spike rates. At lower input rates, the energy per spike is lower and variability is higher.

4.4 Discussion

The presynapse is able to produce an exponentially decaying current spike from an input pulse. The basic presynapse design is not accurate across process variation.

The improved presynapse design shows that it performs the same function, while greatly reducing the variability due to process variation. This is at the expense of some increase in energy consumption though.

The synapse scales its input current from the presynapse proportional to the set weight. The linearity of the DAC in the synapse is within 1 LSB for the typical and slow process corners. For the fast corner the linearity is worse due to increased leakage, which becomes significant compared to the current corresponding to 1 LSB. This could be improved by increasing the transistor lengths in the R-2R DAC structure.

The neuron achieves an energy per spike consumption of less than 0,4 pJ. This is better than the comparable implementations in Table 2.3.

Bias currents are able to adjust the time constants and the amount of frequency adaptation of the neuron. The refractory period and the output spike width are inversely proportional to their control currents. This limits the usable range of these parameters since relatively high currents are needed to obtain short times, and and precision is lost at long time settings due to the very small control currents and the large sensitivity to small changes in the current. A possible improvement to the neuron would be to extend the usable adjustable range of these parameters or to make them linearly proportional to the control currents.

5 Measurements

The taped-out SoC as described in Section 3.6.2 was measured in a lab setting. The spiking functionality was verified and the power consumption of the neuromorphic components was measured.

5.1 Setup & equipment

The measurement setup consists of an FPGA board and a custom measurement board, see the schematic block diagram in Figure 5.1. The FPGA board is a Zybo Zynq-7010. This FPGA is used to interface with the digital signals from the neuromorphic SoC, such as serial communication and spike generation.

The measurement board is connected to the Pmod connectors of the Zybo board for the digital communication. The measurement board contains level shifters for the digital signals to the FPGA, voltage regulators for the supply power of the SoC and current sources to generate the control currents of the analog circuits in the SoC. The measurement board contains the device under test (DUT) in a central test socket.

A photo of the boards of the measurement setup is shown in Figure 5.2.

The measurement board is powered by a Tektronix PS503A power supply. Measurements were taken with Agilent 34410A and Keysight 34461A 6,5 digit multimeters and an Agilent DSO6104A oscilloscope. External input spikes are generated using a Tektronix PG502 pulse generator. A Stanford Research Systems CG635 clock generator is used to generate a system clock signal for the digital controller.

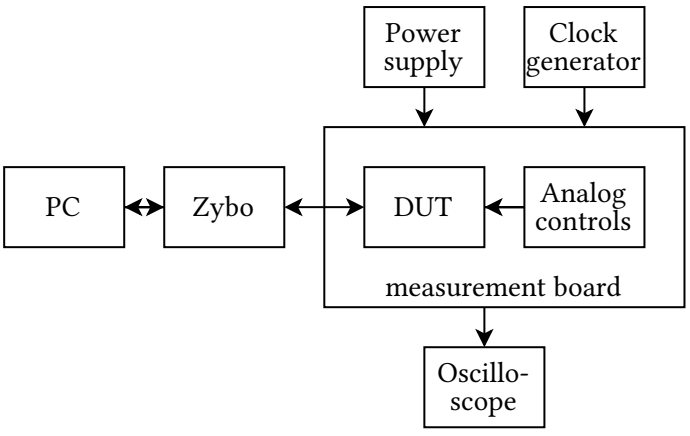


Figure 5.1: Schematic of measurement setup

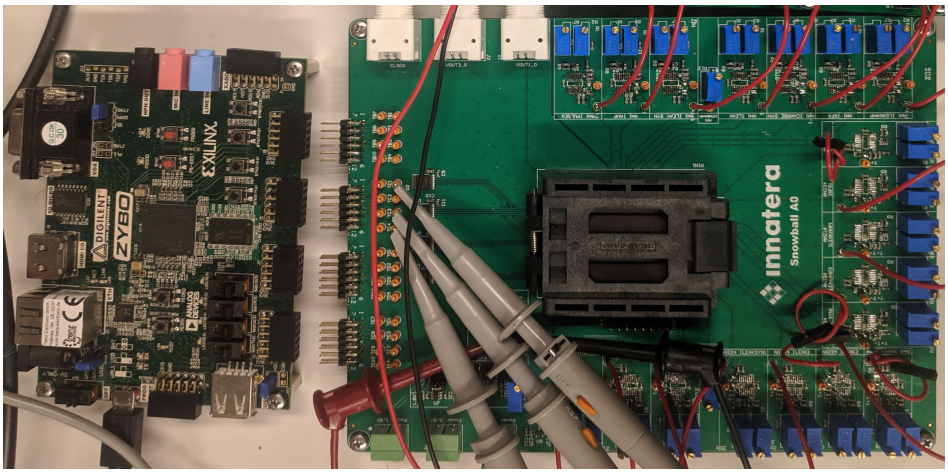


Figure 5.2: Test setup with measurement board (right), FPGA board (left) and DUT IC in socket

5.2 Functional verification

The generated output spikes from the neuron are observed in the SOC. Figure 5.3 shows the output spikes (channel 1, yellow) of a neuron in reaction to a periodic pulse input (channel 2, green) of 100 kHz. The weights in the network are configured in such a way that output spikes are generated at that same rate.

The integrating behaviour of the neuron is shown in Figure 5.4. A 100 khz periodic input results in an output spike signal of approximately 15 kHz during this test. When there is no input activity no output spikes are produced.

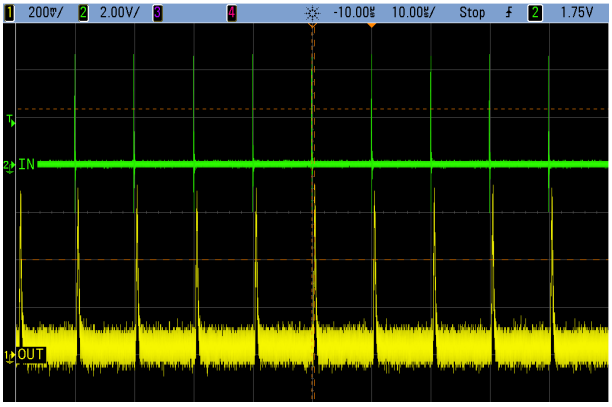


Figure 5.3: Periodic spike signals of 100 kHz

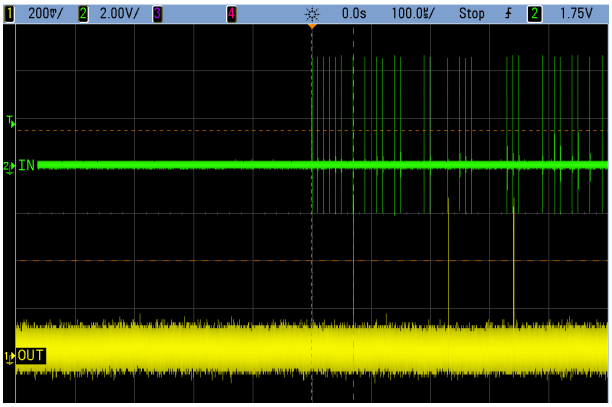


Figure 5.4: Output spikes resulting from the integration of a 100 kHz input spike train

5.3 Power

The power consumption of the neuromorphic network is measured. Power measurements of different network configurations can then used to calculate the energy consumption of the individual elements of the network.

The power measurements are performed on a network of 26 input presynapses, 676 synapses and 26 output neurons. The measurements are taken with a periodic input of 80 kHz to all 26 inputs. The dynamic power is measured, which is the average additional power that is consumed during spike activity minus the static power of the network at rest.

Table 5.1: Power measurements

Active elements	Weight setting	Leakage setting	Supply current [uA]	Dynamic power [uW]
Presynapses	0	N/A	40,97	32,78
Presynapses, synapses, neurons	on	low	49,66	39,73
Presynapses, synapses	on	high	47,42	37,94

Three different measurements are necessary, shown in Table 5.1. For the first measurement all weights are set to 0. The spike generation in the presynapses is active, but the synapses produce no output spikes and there is no activity in the output neurons. The power that is measured in this configuration includes the distribution of the spikes to the synapses and the supply current consumed to generate a spike at the synapse input. Therefore the power and energy per element are reported per synapse.

For the second measurement the weights of the synapses are turned on. In this configuration the power consumption of all elements in the network is measured.

To determine the power consumption of the neurons a measurement is taken in which the neurons are disabled, which can then be subtracted from the previous measurement. The neurons are disabled by setting the leakage current of the neuron to a current that is higher than the (average) input current to the neuron, such that no integration and resulting spike generation occurs.

From these power measurements the power consumption of the presynapses, synapses and neurons

is calculated, see Table 5.2. For the power consumption of the presynapses the power measured in the first configuration is taken. The power consumption of the synapses is calculated by subtracting the first power measurement from the third. By subtracting the third power measurement from the second the power consumption of the neurons is obtained.

Table 5.2: Calculated energy consumption

Element	Dynamic power [uW]	Power per element [nW]	Energy per spike [pJ]
Presynapses[1]	32,78	48,49	0,61
Synapses	5,16	7,63	0,10
Neurons	1,79	68,81	0,86

[1] Includes spike current distribution at synapse inputs. Presynapse energy is reported per connected synapse

The energy consumption per synapse is the sum of the energy required to generate an analog spike at the synapse input and the energy consumption of the synapse itself. This comes to 0,70 pJ per synaptic event.

5.4 Conclusion

The network is able to apply weights to incoming spikes, sum them to a neuron input and integrate the spikes in order to generate output spikes in the neuron.

The measured energy consumption of the presynapse and synapses is higher than is reported in Table 4.1, but this is due to the inclusion of the spike distribution to the synapse inputs in the measurement. The energy consumption of the spike generation at a synapse can be approximated as the charge per spike times the supply voltage. The energy per spike for the nominal process corner in Table 4.1 that can be directly compared to the measured value is then:

$$E_{\text{spike}} = \frac{E_{\text{presynapse}}}{26} + Q_{\text{spike}} V_{\text{DD}} \tag{5.1}$$

For the nominal process corner in Table 4.1 this comes to $E_{\text{spike}} = \frac{24{,}53\,\text{fJ}}{26} + 45{,}38\,\text{fC} \cdot 0{,}8\,\text{V} = 37{,}3\,\text{fJ}$ per spike per synapse, for the fast process corner $E_{\text{spike}} = \frac{367{,}90\,\text{fJ}}{26} + 895{,}65\,\text{fC} \cdot 0{,}8\,\text{V} = 0{,}73\,\text{pJ}$. The measured energy per synaptic event (0,61 pJ) falls within these simulated corners.

The energy consumed to generate an output spike in the neuron is 2,5 times larger than predicted in simulation (Table 4.6). This might be due to unmodelled parasitics in the simulation, especially parasitic leakage and capacitive loading of critical nodes. Based on the power consumption of the synapses it is possible that the measured die is produced at a faster than nominal process corner. This would also increase the power consumption in the neuron and would increase parasitic leakage.

6 Conclusion

6.1 Conclusion

In this thesis neuron and synapse circuits have been implemented in 28 nm CMOS technology. A novel distributed synapse structure was developed. The components were integrated in neuromorphic arrays in an SOC, which was successfully taped out.

Different neuron models in literature were compared. The adaptive exponential integrate-and-fire model is a good compromise between modelling accuracy, biological plausibility and ease of circuit implementation. The exponential terms of the model map well to the exponential relation of subthreshold MOSFET transistors.

Table 6.1 shows a comparison of a selection of the implementations in Table 2.3 to this work. The neuron that was developed shows a lower energy consumption per spike than previous work. Accelerating the spike frequency, although at the cost of direct compatibility to biological systems, enables this implementation to have much smaller capacitance and area than [28], an implementation that has a comparable supply voltage and an energy consumption on the same order of magnitude.

Table 6.1: Comparison of adaptive exponential integrate-and-fire neuron circuit implementations

	[26]	[28]	[3]	This work
Technology	28 nm FD-SOI	22 nm FD-SOI	65 nm	28 nm
Supply voltage	1,0 V	0,8 V	1,0 V	0,8 V
Energy per spike	50 pJ	990 fJ	730 fJ	354 fJ
Active area	20 µm²		0,18 µm²	26,6 µm²
Capacitance	900 fF	4,3 pF	345 fF	200 fF
Capacitor area	50 µm²	1799 µm²	25,8 µm²	43,4 µm²
Typical spike frequency	100 Hz	100 Hz	1 kHz	100 kHz
Frequency variability	5,86 %	56,55 %		25,2 %

A synapse design was developed that splits a previous synapse circuit into a distributed presynapse-synapse structure. Improvements were made to the presynapse circuit to significantly decrease process variability. The distributed synaptic structure and the design of the presynapse have been used in the patent application 'Distributed multi-component synaptic computational structure' [33].

6.2 Future work

The neuron design presented in this thesis can be further improved. Although it is competitive in terms of area and energy usage per spike, other state of the art has less spike frequency variab-

ility [26]. It might possible to improve performance in this regard by increasing the size of transistors in the input filter and feedback circuits, as well as those controlling the leakage and AHP currents.

The DAC used in the synapse has demonstrated a high non-linearity at low weights in process corners with high leakage. Increasing the transistor length of critical components in the R-2R structure would likely improve linearity, but will impact the area that is used for the synapse.

The presynapse–synapse structure distributes the output spike from the presynapse to the synapses using a voltage signal. This method leads to differences between synapses, depending on their distance to the presynapse, because the capacitive load on the distribution track increases along the row. The voltage-based distribution is also sensitive to crosstalk from neighbouring signals. The structure could be improved to use current-mode distribution. A challenge in this approach is that an individual routing track is needed for every synapse in a row, instead of the shared track in the current design of the neuromorphic network. This limits the amount of synapses in a row to the usable routing area in a row.

Bibliography

[1] C. Mead, 'Neuromorphic electronic systems,' *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990. DOI: 10.1109/5.58356.

[2] L. S. Smith, 'Neuromorphic systems: Past, present and future,' in *Brain Inspired Cognitive Systems 2008*, ser. Advances in Experimental Medicine and Biology, A. Hussain, I. Aleksander, L. S. Smith, A. K. Barros, R. Chrisley and V. Cutsuridis, Eds., vol. 657, Springer New York, Nov. 2010, pp. 167–182. DOI: 10.1007/978-0-387-79100-5_9.

[3] E. Stienstra, 'A 32 x 32 spiking neural network system on chip,' M.S. thesis, Delft University of Technology, Delft, The Netherlands, 2017. eprint: http://resolver.tudelft.nl/uuid:94d5f6b7-3d06-4b7a-9f39-0b346083386e.

[4] X. You, 'Full-custom multi-compartment synaptic circuits in neuromorphic structures,' M.S. thesis, Delft University of Technology, Delft, The Netherlands, 2017. eprint: http://resolver.tudelft.nl/uuid:0f761b83-6087-4b3a-a39e-955a508d0f3c.

[5] TOP500.org. 'Green500 June 2023.' (Jun. 2023), [Online]. Available: https://www.top500.org/lists/green500/2023/06/.

[6] K. Rupp. 'CPU, GPU and MIC hardware characteristics over time.' (Jun. 2013), [Online]. Available: https://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/ (visited on 26-03-2019).

[7] S.-C. Liu and T. Delbruck, 'Neuromorphic sensory systems,' *Current opinion in neurobiology*, vol. 20, no. 3, pp. 288–295, 2010, Sensory systems. DOI: 10.1016/j.conb.2010.03.007.

[8] G. Indiveri and S.-C. Liu, 'Memory and information processing in neuromorphic systems,' *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1379–1397, Aug. 2015. DOI: 10.1109/JPROC.2015.2444094.

[9] J. Wu, *Introduction to Neural Dynamics and Signal Transmission Delay* (de Gruyter Series in Nonlinear Analysis and Applications 6). De Gruyter, 15 May 2001, ISBN: 9783110169881.

[10] W. Gerstner, W. M. Kistler, R. Naud and L. Paninski, *Neuronal Dynamics, From single neurons to networks and models of cognition*. Cambridge University Press, Jul. 2014. [Online]. Available: https://neuronaldynamics.epfl.ch/online (visited on 02-2019).

[11] A. Destexhe, Z. F. Mainen and T. J. Sejnowski, 'Kinetic models of synaptic transmission,' in *Methods in Neuronal Modeling* (Computational Neuroscience Series), C. Koch and I. Segev, Eds., 2nd ed., Computational Neuroscience Series. Cambridge: MIT Press, 1998, ch. 1, pp. 1–25, ISBN: 9780262112314.

[12] S. H. Wu, C. L. Ma and J. B. Kelly, 'Contribution of AMPA, NMDA, and $GABA_A$ receptors to temporal pattern of postsynaptic responses in the inferior colliculus of the rat,' *Journal of Neuroscience*, vol. 24, no. 19, pp. 4625–4634, 12 May 2004. DOI: 10.1523/JNEUROSCI.0318-04.2004.

[13] A. L. Hodgkin and A. F. Huxley, 'A quantitative description of membrane current and its application to conduction and excitation in nerve,' *The Journal of physiology*, vol. 117, no. 4, pp. 500–544, 28 Aug. 1952. DOI: 10.1113/jphysiol.1952.sp004764.

[14] R. Brette and W. Gerstner, 'Adaptive exponential integrate-and-fire model as an effective description of neuronal activity,' *Journal of Neurophysiology*, vol. 94, no. 5, pp. 3637–3642, Nov. 2005. DOI: 10.1152/jn.00686.2005.

[15] E. M. Izhikevich, 'Simple model of spiking neurons,' *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003. DOI: 10.1109/TNN.2003.820440.

[16] E. M. Izhikevich, 'Which model to use for cortical spiking neurons?' *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1063–1070, Sep. 2004. DOI: 10.1109/TNN.2004.832719.

[17] C. D. Schuman, T. E. Potok, R. M. Patton *et al.*, 'A survey of neuromorphic computing and neural networks in hardware,' *CoRR*, 2017. arXiv: 1705.06963.

[18] R. Jolivet, T. J. Lewis and W. Gerstner, 'Generalized integrate-and-fire models of neuronal activity approximate spike trains of a detailed model to a high degree of accuracy,' *Journal of Neurophysiology*, vol. 92, no. 2, pp. 959–976, Aug. 2004. DOI: 10.1152/jn.00190.2004.

[19] J. M. Cruz-Albrecht, M. W. Yung and N. Srinivasa, 'Energy-efficient neuron, synapse and stdp integrated circuits,' 3, vol. 6, IEEE, 2012, pp. 246–256. DOI: 10.1109/tbcas.2011.2174152.

[20] R. Wang, C. S. Thakur, T. J. Hamilton, J. Tapson and A. van Schaik, 'A compact aVLSI conductance-based silicon neuron,' in *2015 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, IEEE, Oct. 2015. DOI: 10.1109/BioCAS.2015.7348396.

[21] A. Joubert, B. Belhadj, O. Temam and R. Héliot, 'Hardware spiking neurons design: Analog or digital?' In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, Jun. 2012, pp. 1–5. DOI: 10.1109/IJCNN.2012.6252600.

[22] P. Livi and G. Indiveri, 'A current-mode conductance-based silicon neuron for Address-Event neuromorphic systems,' *2009 IEEE International Symposium on Circuits and Systems*, 2009. DOI: 10.1109/iscas.2009.5118408.

[23] G. Indiveri, B. Linares-Barranco, T. J. Hamilton *et al.*, 'Neuromorphic silicon neuron circuits,' *Frontiers in Neuroscience*, vol. 5, no. 73, B. Shi, Ed., 31 May 2011. DOI: 10.3389/fnins.2011.00073.

[24] E. Chicca, F. Stefanini, C. Bartolozzi and G. Indiveri, 'Neuromorphic electronic circuits for building autonomous cognitive systems,' *Proceedings of the IEEE*, vol. 102, no. 9, pp. 1367–1388, Sep. 2014. DOI: 10.1109/JPROC.2014.2313954.

[25] N. Qiao, H. Mostafa, F. Corradi *et al.*, 'A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses,' *Frontiers in Neuroscience*, vol. 9, 29 Apr. 2015. DOI: 10.3389/fnins.2015.00141.

[26] N. Qiao and G. Indiveri, 'Scaling mixed-signal neuromorphic processors to 28 nm FD-SOI technologies,' in *2016 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, IEEE, Oct. 2016. DOI: 10.1109/BioCAS.2016.7833854.

[27] N. Qiao and G. Indiveri, 'Analog circuits for mixed-signal neuromorphic computing architectures in 28 nm FD-SOI technology,' in *2017 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*, IEEE, Oct. 2017. DOI: 10.1109/S3S.2017.8309203.

[28] A. Rubino, M. Payvand and G. Indiveri, 'Ultra-low power silicon neuron circuit for extreme-edge neuromorphic intelligence,' IEEE, 2019. DOI: 10.1109/icecs46596.2019.8964713.

[29] X. You, A. Zjajo, S. S. Kumar and R. van Leuken, 'Energy-efficient neuromorphic receptors for wide-range temporal patterns of post-synaptic responses,' in *2017 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC)*, IEEE, Oct. 2017. DOI: 10.1109/NORCHIP.2017.8124951.

[30] C. Bartolozzi and G. Indiveri, 'Synaptic dynamics in analog VLSI,' *Neural Computation*, vol. 19, no. 10, pp. 2581–2603, Oct. 2007. DOI: 10.1162/neco.2007.19.10.2581.

[31] A. Wang, A. P. Chandrakasan and B. H. Calhoun, *Sub-threshold Design for Ultra Low-Power Systems*. Boston, MA: Springer, 11 Dec. 2006. DOI: 10.1007/978-0-387-34501-7.

[32] N. H. E. Weste and D. M. Harris, *CMOS VLSI design, A circuits and systems perspective*, 4th ed. Boston, Mass.: Addison-Wesley, 2011, 838 pp., ISBN: 9780321547743.

[33] B. Hettema and A. Zjajo, 'Distributed multi-component synaptic computational structure,' U.S. Patent WO/2022/090542, 5 May 2022.

[34] K. Bult and G. Geelen, 'An inherently linear and compact MOST-only current-division technique,' *IEEE Journal of Solid-State Circuits*, vol. 27, no. 12, pp. 1730–1735, Dec. 1992. DOI: 10.1109/ISSCC.1992.200480.

[35] T. Delbrück and A. van Schaik, 'Bias current generators with wide dynamic range,' *Analog Integrated Circuits and Signal Processing*, vol. 43, no. 3, pp. 247–268, Jun. 2005. DOI: 10.1007/s10470-005-1606-1.