Dissertation

submitted to the
Combined Faculties for the Natural Sciences and for Mathematics
of the Ruperto-Carola University of Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

Put forward by

Dipl.-Phys. Paul Müller

born in Kujbyschew, Russia

Oral examination: 2017-11-02

Modeling and Verification for a Scalable Neuromorphic Substrate

Referees: Prof. Dr. Karlheinz Meier
Prof. Dr. Michael Hausmann

Modeling and Verification for a Scalable Neuromorphic Substrate

Mixed-signal accelerated neuromorphic hardware is a class of devices that implements physical models of neural networks in dedicated analog and digital circuits. These devices offer the advantages of high acceleration and energy efficiency for the emulation of spiking neural networks but pose constraints in form of device variability and of limited connectivity and bandwidth. We address these constraints using two complementary approaches: At the network level, the influence of multiple distortion mechanisms on two benchmark models is analyzed and compensation methods are developed that counteract the resulting effects. The compensation methods are validated using a simulation of the BrainScaleS neuromorphic hardware system. At the single neuron level, calibration procedures are presented that counteract device variability for a new analog implementation of an adaptive exponential integrate-and-fire neuron model in a 65 nm process. The functionality of the neuron circuit together with these calibration methods is verified in detailed transistor-level simulations before production. The versatility of the circuit design that includes novel multi-compartment and plateau-potential features is demonstrated in use cases inspired by biology and machine learning.

Modellierung und Verifikation für ein skalierbares neuromorphes Substrat

Beschleunigte, digital-analoge neuromorphe Hardware ist eine Klasse von Systemen, welche physikalische Modelle neuronaler Netzwerke in dedizierten, analogen und digitalen Schaltungen implementiert. Diese Systeme bieten die Vorteile von Energieeffizienz und hoher Beschleunigung bei der Emulation von aktionspotenzial-basierten neuronalen Netzen, aber besitzen Einschränkungen bezüglich Konnektivität, Bandbreite und der Variation einzelner Komponenten. Der Einfluss dieser Einschränkungen wird in zwei komplementären Ansätzen behandelt: Auf der Netzwerkebene wird der Einfluss mehrerer Störmechanismen auf zwei Benchmarkmodelle analysiert und durch modellspezifische Kompensationsmethoden minimiert. Diese Methoden werden auf einem Simulationsmodell des neuromorphen BrainScaleS-Systems validiert. Auf der Neuronebene werden Verfahren zur Kalibration entwickelt, die der Komponentenvariation in einer Neuentwicklung eines analogen AdEx-Neuronmodells in einem 65 nm-Prozess entgegenwirken. Die Funktionalität der Neuronschaltung und der Kalibration wird in detaillierten Simulationen auf Transistorebene vor der Produktion verifiziert. Die Vielseitigkeit der Schaltung wird in Anwendungfällen demonstriert, welche von biologischen und abstrakten Modellen inspiriert sind.

Contents

Chapter 1

Introduction

The advancement of the information age is characterized by the dependency of the global society on the reliable handling and processing of ever growing volumes of data. Driven by the exponential increase in computing power and storage capacity (Mack, 2011; Moore, 1965; Leventhal, 2008), mechanized data processing becomes ever more impressive, affecting not only our day-to-day life but also being deployed in contexts with high demands on reliability to guarantee human safety or economic efficiency. This is made apparent by the progress of, for example, simple automatic brakes on railway trains (De Nicola et al., 2005) to fully automatic vehicles navigating normal road traffic, a colossal computational difference between the correlation of a semaphore signal and the speed of a railroad cart and the real-time processing of high-bandwidth input signals under greatly varying environmental conditions (Bojarski et al., 2016). From automatically sorting mail using handwritten text to recognizing human faces and understanding voice commands on mobile devices, from beating novice players at checkers (Samuel, 1967) to winning against world-class players in the games of chess (Deep Blue vs. Kasparov, 1997) and Go (Silver et al., 2016), from simple dictionary-based translation to automatic subtitle generation for video footage, a significant part of the machine learning field is dominated by algorithms based on neural networks. Prominent examples of network models are LSTM, convolutional neural networks and Deep Boltzmann Machines (Hochreiter and Schmidhuber, 1997; Salakhutdinov and Hinton, 2009). These networks are the mathematical successors of networks loosely based on the structure that is observed in the human brain. Each neuron possesses a state which is determined by the state of the neurons connected to it and the properties of the connection. The ability to tune this connectivity – in many models represented as a connection matrix – to make an abstract network perform its desired task is one of the key components to making them so successful in a variety of tasks.

The human brain is an exceptional information processing and learning machine by itself. Humans can learn to perform a multitude of highly different tasks, and do so with a comparatively low power consumption, commonly given as 20 W (Raichle and Gusnard, 2002, 20 % of calorie intake). This kind of generality and efficiency makes understanding the computation in the human brain a highly valuable effort,

in addition to the possible positive effects on health and society. It is not clear yet to what extent the difference between the comparatively simple representation of neurons and synapses in abstract neural networks and the complex dynamics of biological neurons is rooted in the disparity of the underlying substrates – and in how far the biological realization is optimized by evolution for the information processing tasks the brain has to perform.

The field of computational neuroscience is dedicated to understanding the individual and collective behavior of biological neurons as well as their computational potential. A central tool for this endeavor is the simulation of spiking networks. The time evolution of a network of neurons – modeled by mathematical descriptions of neurons and synapses of a highly varying degree of simplification – is solved numerically to obtain insight into the emergent behavior of the system. This numeric solution is costly in terms of computational resources, time and power consumption. Kunkel et al. (2014), for example, demonstrate a large-scale simulation with 1.86×10^9 neurons which requires approximately 40 minutes for one second of simulated time on the K supercomputer.

One alternative is the construction of dedicated hardware which is designed to optimize the simulation of spiking networks. As it is not meaningful to try to compete with the generality of simulations running on conventional compute clusters, such devices are typically specialized for a set of tasks. One such approach is *neuromorphic computing* with mixed-signal devices. One core characteristic is that neurons are represented by a *physical model*, a dedicated analog circuit that mimics the dynamics of a biological neuron. The membrane capacitance of the neuron is represented by a physical capacitor and the currents that govern the neuron's behavior are emulated by analog circuits. The possible values of specific capacitance and utilized currents in standard VLSI processes make it easy to realize time constants far below those of typical biological neurons. Exploiting this fact, mixed-signal accelerated neuromorphic systems are constructed which implement analog neurons with a configurable connection matrix that allows to realize networks of vastly different topologies.

Each neuron and network model possesses an intrinsic level of deviation from the system it models: The simple leaky integrate-and-fire point neuron model, for example, does not capture the complex nonlinearities and spatial extent of the biological reference. Even a numerical model does not, in general, exactly capture its mathematical archetype, but rather is built to allow a control over the resulting error due to, e.g., discretization in time, space or limited resolution of values. The final appearance of any given model is driven not only by the faithful reproduction of the investigated system but, to a large extent, by what is easy to implement and control while still capturing the essential behavior of the system. For mathematical models it may be the existence of a mathematical tool set that facilitates the analysis of the given structure, such as the Rall model (Rall, 1959) for dendritic structures, the Ising model for the understanding of magnetism or the mean field approach to the description of neural networks (Brunel, 2000). For simulation, it is the availability of appropriate algorithms as well as software packages that allow the efficient implementation

of the model (Eppler et al., 2008; Hines and Carnevale, 2003). Valuable insights which result from the analysis of these models are characterized by their capacity to generalize to similar systems, the Ising model, for example, being a fundamental example of phase transitions in magnetic materials.

In the case of mixed-signal accelerated neuromorphic systems, the components that are promising to construct are energy- and area-efficient circuits with a high speedup compared to their biological counterparts. The advantages of the speedup and power consumption are gained at the cost of flexibility of implementation and parameterization, and at the cost of the homogeneity of the underlying substrate: First, the flexibility of implementation is limited to the amount that is provided by the circuit designer. In contrast to a software simulation, where, in principle, an arbitrary part of the model can be replaced quickly, the circuit implementation on a chip can not be modified once it is produced. To provide a flexible substrate for the user, the circuits are made configurable by including parameters which can be changed at runtime. Second, this parameterization does not achieve the gigantic range of double-precision floating-point numbers one is used to in software simulations, but is limited to few orders of magnitude at best. Finally, the size reduction is usually carried to a domain where device mismatch becomes significant (Pelgrom et al., 1989). Thus, the user has to cope with the fact that identically parameterized components behave differently due to variations in production.

We emphasize that the goals of neuromorphic computing are twofold, namely to help identify and understand effective paradigms of spike-based computation, and to uncover efficient hardware implementations of these computational concepts. Overcoming the constraints that are outlined above is an important step on the path to achieving these goals.

This thesis approaches the problems described above in two complementary ways. In the first part, we start from network models that are used in existing computational studies of spiking neural networks. Distortions that are expected and measured on neuromorphic hardware devices are identified and modeled in an idealized way. The effect of these distortions is investigated for each mechanism individually, providing compensation methods that counteract each distortion. In a final step, the developed compensation methods are validated with all modeled distortions using a simulated neuromorphic hardware. The result is a set of exemplary compensation methods for different network models.

In the second part of this thesis, the performance of individual neuron circuits is examined in more detail. The distortion mechanisms used in the initial study are idealized and do not cover the full range of possible effects. In the second part, a detailed transistor-level simulation of the neuron implementation in development is performed. Use cases derived from biological and abstract models are used to test the operation of the circuit in simulation, testing novel features that implement multi-compartment functionality and plateau-potentials as well as the re-implementation of features from a previous hardware generation. A detailed analysis of the expected parameter ranges, limitations and variation in the circuit is presented, along with a simulation-based pre-production verification which helped remedy mistakes in the

circuit design.

This document is organized as follows: The last part of this chapter provides an introduction to the mathematical and electronic modeling of spiking neurons. In chapter 2, the study of network-level anomalies introduced by their emulation on neuromorphic devices is presented. The simulation-based characterization and pre-production verification is discussed in chapter 3.

1.1 Biological neurons

The human brain contains approximately 86×10^9 neurons (Azevedo et al. (2009)) which form the computational substrate for the complex cognitive tasks that humans can perform. The cell body of the neuron is called the *soma*, from which long extensions, the *axon* and *dendrites* protrude (fig. 1.1). A voltage is maintained across the cell membrane by active ion pumps, leading to a decreased concentration of Na^+ and an increased concentration of K^+ ions inside the cell. The resting potential is given as negative by convention and amounts to approximately –70 mV. The soma can generate a sharp voltage peak, called *action potential*, which is transported along the axon. Synapses are formed between the axon and dendrites of other neurons. The pre- and postsynaptic cell membranes are in close proximity at the synaptic cleft. When an action potential propagates along the axon it triggers the release of neurotransmitters at the presynaptic site which bind to and activate receptor molecules within the membrane of the postsynaptic neuron. The receptors can open ligand-gated ion channels, leading to a flux of ions through the membrane and leading to a post-synaptic potential (PSP), which can be positive (excitatory) or negative (inhibitory) in voltage. The PSPs are summed electrically at the soma where an action potential is generated when – in the simplest possible model – the membrane potential exceeds a certain threshold. One of the most influential models for the computational investigation of neurons is the Hodgkin and Huxley model (Hodgkin and Huxley (1952)), which explains the mechanism behind the generation of the action potential in the giant axon of the squid. Their experimentally validated model contains sodium and potassium conductances which open and close according to an intrinsic, time- and voltage dependent dynamics.

For the investigation of large networks on an abstract level, the mechanism of spike generation is far less important than the interaction between neurons. Simple neuron models offer the advantages of requiring less computational resources for simulation and are easier to handle analytically. Two of such low-dimensional, threshold-based neurons models, leaky integrate-and-fire (LIF) and adaptive exponential integrate-and-fire (AdEx), are described in the following.

1.2 The leaky integrate-and-fire neuron model

The LIF neuron model is one of the simplest models that capture the basic properties of temporal integration and all-or-none spike emission. It describes a passive leaky

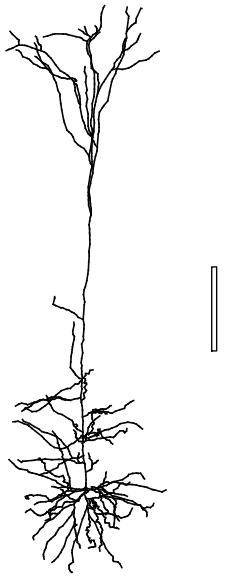


Figure 1.1: Morphology of a layer 5 neuron from rat neocortex (Chen et al. (2014)). Data provided by Ascoli et al. (2007) via *NeuroMorpho.org*. The rectangle on the right represents the area of a single hardware neuron on the HICANN DLS2 chip (Aamir et al. (2016)), without synapses, at the same scale with a length of 200 µm.

integrator, in which the time constant τ_m is given by $\frac{C_\mathrm{m}}{g_\mathrm{l}}$, the membrane capacitance and the leakage conductance of the membrane (Stein, 1967; Gerstner and Kistler, 2002):

$$C_\mathrm{m}\frac{\mathrm{d}V_\mathrm{m}}{\mathrm{d}t} = -g_\mathrm{l}(V_\mathrm{m} - V_\mathrm{leak}) + I \tag{1.1}$$

The current I comprises the synaptic input and, e.g., external stimulus current that is applied to the neuron. V_leak is leak potential of the neuron.

This linear differential equation is expanded to a spiking neuron model by introducing an explicit firing mechanism: The membrane potential is reset immediately after it reaches the firing threshold V_thresh. After each spike, the membrane potential is held constant at the reset potential V_reset (fig. 1.2).

$$\begin{aligned} & V_\mathrm{m}(t_\mathrm{spike}) = V_\mathrm{thresh} \\ \Rightarrow \quad & \forall t \in \left(t_\mathrm{spike}; t_\mathrm{spike} + \tau_\mathrm{ref}\right] : V_\mathrm{m}(t) = V_\mathrm{reset} \end{aligned} \tag{1.2}$$

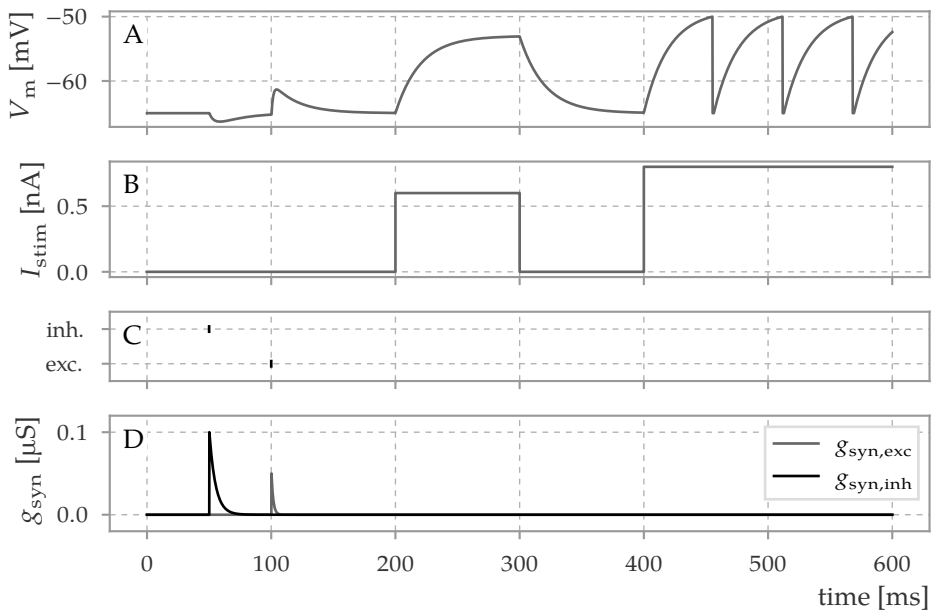


Figure 1.2: Leaky Integrate-and-Fire neuron model with exponential, conductance-based synapses. **A**: Membrane potential V_m of the simulated neuron. **B**: Stimulus current of 600 pA between 200 ms to 300 ms and 800 pA between 400 ms to 600 ms is added to the membrane capacitance, causing an exponential approach of the membrane voltage towards the new resting state. When the membrane voltage reaches the firing threshold of $V_\mathrm{thresh} = -50\,\mathrm{mV}$ it is held at the reset voltage $V_\mathrm{reset} = -65\,\mathrm{mV}$ for the duration of $\tau_\mathrm{ref} = 1\,\mathrm{ms}$. **C**: The neuron is stimulated by inhibitory and excitatory input at 50 ms and 100 ms. **D**: Each synaptic event causes an instantaneous increase of the synaptic conductance $g_\mathrm{syn,I}$ and $g_\mathrm{syn,E}$ by the corresponding synaptic weight, here $w_\mathrm{I} = 100\,\mathrm{nS}$, $w_\mathrm{E} = 50\,\mathrm{nS}$. When no input arrives, each synaptic conductance variable decays to zero with its synaptic time constant $\tau_\mathrm{syn,I}$ and $\tau_\mathrm{syn,E}$. Note that the PSP of the inhibitory input is smaller than that of the excitatory one, even though the inhibitory weight and time constant are greater. This is caused by the inhibitory reversal potential $E_\mathrm{rev,I} = -70\,\mathrm{mV}$ being closer to the resting potential $V_\mathrm{leak} = -65\,\mathrm{mV}$ than the excitatory reversal potential $E_\mathrm{rev,E} = 0\,\mathrm{mV}$.

1.3 The adaptive exponential integrate-and-fire neuron model

The one-dimensional LIF model is a simple and widely used model of spiking neurons that provides an easy approach to simulation and analytical investigations. The observation of widely varying, complex responses in cortical neurons (Markram et al., 2004) has driven the investigation of simple neuron models that can expand the diversity of the model dynamics while still being computationally affordable (Izhikevich, 2004). One prominent model is described in (Izhikevich, 2003). It extends the LIF model by adding a non-linear, quadratic term to the derivative of the membrane potential and introducing an adaptation variable, which gives the neuron model a memory beyond the time of its last reset.

The AdEx model introduced in (Brette and Gerstner, 2005) is a similar nonlinear, adaptive two-dimensional neuron model. However, it employs an exponential function as nonlinearity for the voltage feedback, which conforms to experimental data (Badel et al., 2008).

$$C_\mathrm{m}\frac{\mathrm{d}V_\mathrm{m}}{\mathrm{d}t} = -g_\mathrm{l}(V_\mathrm{m} - V_\mathrm{leak}) + g_\mathrm{l}\Delta_\mathrm{T}\exp\left(\frac{V_\mathrm{m} - V_\mathrm{T}}{\Delta_\mathrm{T}}\right) - w + I \tag{1.3}$$

$$\tau_\mathrm{w}\frac{\mathrm{d}w}{\mathrm{d}t} = a(V_\mathrm{m} - V_\mathrm{leak}) - w \tag{1.4}$$

The adaptation current w is coupled linearly to the membrane potential via the subthreshold adaptation a and has a time constant τ_w. The width of the exponential current is controlled via the slope factor Δ_T while the spike threshold V_T controls the location of its onset. The factor in front of the exponential term $g_\mathrm{l}\Delta_\mathrm{T}$ is chosen such that $\frac{\mathrm{d}}{\mathrm{d}V_\mathrm{m}}\frac{\mathrm{d}V_\mathrm{m}}{\mathrm{d}t} = 0$ at $V_\mathrm{m} = V_\mathrm{T}$. An explicit threshold voltage as in eq. (1.2) is used in practice, even though one could formally define the spike time as the time at which V_m grows towards infinity (Brette and Gerstner, 2005).

1.4 Models of synaptic interaction

The neuron models that are presented in section 1.2 and section 1.3 are missing an essential detail to use them for simulating network activity: the synaptic interaction between neurons. We enumerate the spike times of neuron j in a network

$$t_{\mathrm{sp},ji} \quad i \in \{1, \ldots, \text{number of spikes of neuron } j\}$$

A simple interaction model would be to directly increment the membrane voltage of the postsynaptic neuron proportionally to the synaptic weight of the coupling (e.g. Brunel, 2000). This is equivalent to a synaptic current that is a sum of delta functions:

$$I_{\mathrm{syn},k}(t) = \sum_j \sum_i w_{jk}\delta(t - t_{\mathrm{sp},ji} - \Delta) \tag{1.5}$$

Here, Δ stands for the transmission delay between pre- and postsynaptic neuron.

1.4.1 Current-based synapse model

In biological neurons, the time scale of synaptic interaction is not negligibly short. This is accommodated by replacing the delta function in eq. (1.5) by an interaction kernel with a finite length.

$$I_{\mathrm{syn},k}(t) = \sum_j \sum_i w_{jk} \kappa(t - t_{\mathrm{sp},ji} - \Delta) \tag{1.6}$$

A common interaction kernel is a step in the current, followed by an exponential decay.

$$\kappa(t) = \theta(t) \exp\left(\frac{-t}{\tau_{\mathrm{syn}}}\right) \tag{1.7}$$

with θ being the Heaviside step function.

Other interaction kernels are used to, e.g, include the rise time of the synaptic interaction, leading to a difference-of-exponentials or alpha-shaped kernels

$$\kappa_\alpha(t) = \theta(t) \frac{1}{\tau_s} \exp\left(\frac{-t}{\tau_s}\right) \tag{1.8}$$

$$\kappa_{\mathrm{de}}(t) = K * \theta(t) \left[\exp\left(\frac{-t}{\tau_1}\right) - \exp\left(\frac{-t}{\tau_2}\right)\right]. \tag{1.9}$$

The kernels shown above possess the helpful property that they are the Green's function of a low-dimensional differential operator. Assuming equal time constants for all synapses this allows one to equivalently write, e.g., eq. (1.6) with the exponential kernel eq. (1.7) as

$$\frac{\mathrm{d}I_{\mathrm{syn}}}{\mathrm{d}t} = \frac{-1}{\tau_{\mathrm{syn}}} I_{\mathrm{syn}} + \sum_j \sum_i w_{jk} \delta(t - t_{\mathrm{sp},ji} - \Delta) \tag{1.10}$$

This formulation allows to combine the effect of all synaptic interactions with the same synaptic time constant into one dynamic variable, which can reduce the memory requirement in the simulation of spiking neurons. (See, e.g., Brette (2006) for an in-depth description of an efficient simulation of neuron models.) The time constants of synaptic ion channels are different for different synaptic transmission mechanisms, such as relatively fast dynamics AMPA and $\mathrm{GABA_A}$ and slower dynamics in NMDA and $\mathrm{GABA_B}$ receptors (Destexhe et al., 1998). Because of this, separate time constants are typically provided for the simulation of spiking neurons. The software package *PyNN* (Davison et al., 2008), for example, provides separate parameters for excitatory and inhibitory synaptic time constants in its default neuron models.

1.4.2 Conductance-based synapse models

A more realistic model of synaptic interaction is that of *conductance-based* synapses (Meffin et al., 2004). Synaptic interaction causes the opening of ion channels for specific ion types, which possess distinct reversal potentials. One consequence of

this is the observation of the so-called *high-conductance state* in neocortical neurons (Destexhe et al., 2003), where an increased total conductance changes the effective input resistance and the temporal resolution of a neuron.

In conductance-based models, the synaptic conductance takes the place of the linear summation of individual post-synaptic kernels eq. (1.6):

$$\frac{\mathrm{d}g_{\mathrm{syn}}}{\mathrm{d}t} = \frac{-1}{\tau_{\mathrm{syn}}} g_{\mathrm{syn}} + \sum_j \sum_i w_{jk} \kappa(t - t_{\mathrm{sp},ji} - \Delta) \tag{1.11}$$

The synaptic current is given by

$$I_{\mathrm{syn}}(t) = g_{\mathrm{syn}}(t)(V_{\mathrm{m}} - E_{\mathrm{rev}}). \tag{1.12}$$

Here, the current is dependent on the membrane potential. The newly introduced parameter E_{rev} is the reversal potential of the ion type that is associated with the synaptic ion channels. As above, in simulations, per-channel-type parameters for the time constants are typically used; for one inhibitory and one excitatory synapse type the nomenclature is $E_{\mathrm{rev,E}}$, $E_{\mathrm{rev,I}}$, $\tau_{\mathrm{syn,E}}$, $\tau_{\mathrm{syn,I}}$.

1.5 Synaptic dynamics

The models in section 1.4 are formulated assuming static synaptic weights which are constant throughout the course of the simulation. In computational neuroscience, and even more so in machine learning, an interesting field is the process of learning – the dynamic reconfiguration of the network to fulfill its intended functionality. Two plasticity mechanisms are singled out due to their relevance to the hardware platforms that are presented in the following chapters: The Tsodyks-Markram mechanism of short-term plasticity and spike-timing-dependent plasticity.

1.5.1 Tsodyks-Markram mechanism

The PSP after a synaptic event is observed to depend on the history of spikes arriving at that synapse (Tsodyks and Markram, 1997). This is explained by the following kinetic model: Each synapse possesses resources which are separated in to the fractions of recovered (R), effective (E) and inactive (I) states. When a presynaptic event arrives at time t_{AP}, a fraction (U_{SE}, synaptic utilization) of remaining recovered resources moves from I to E. Using the notation from Tsodyks and Markram (1997):

$$\frac{\mathrm{d}R}{\mathrm{d}t} = \frac{I}{\tau_{\mathrm{rec}}} \tag{1.13}$$

$$\frac{\mathrm{d}E}{\mathrm{d}t} = \frac{-E}{\tau_{\mathrm{in}}} + U_{\mathrm{SE}} \cdot R \cdot \delta(t - t_{\mathrm{sp}}) \tag{1.14}$$

$$I = 1 - R - E \tag{1.15}$$

The net postsynaptic current is proportional to E.

This model is extended in Tsodyks et al. (cf. 1998, 2.1) to include synaptic facilitation by making U_{SE} a dynamic variable alongside I, R and E.

1.5.2 Spike-timing-dependent plasticity

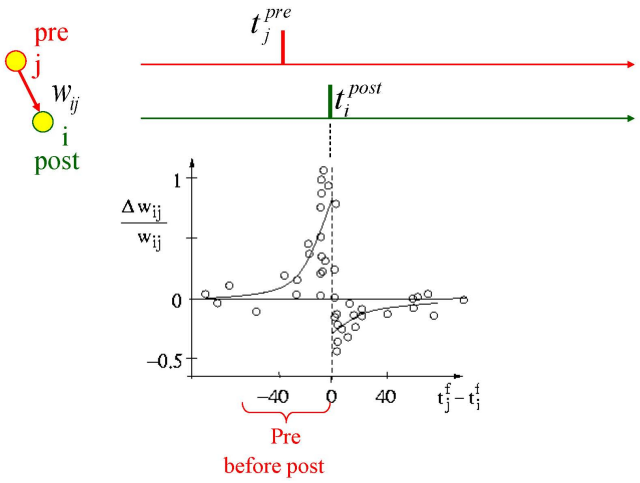


Figure 1.3: Used with permission from Sjöström and Gerstner (2010) (CC BY-NC-SA 3.0, *https://creativecommons.org/licenses/by-nc-sa/3.0/deed.en_US*), which is redrawn after Bi and Poo (1998). The unit on the time axis is ms.

One prominent model of long-term plasticity is based on the measurement of changes in synaptic efficacy in dependence of the relative timing of pre- and postsynaptic spikes (Markram et al., 1997; Bi and Poo, 1998). When neurons are stimulated by a series of presynaptic events followed or preceded by a series of induced postsynaptic events, a change of the synaptic efficacy is measured (fig. 1.3). The synaptic effect increases when the presynaptic event precedes the postsynaptic event in a given time window. Likewise, the effect decreases when the postsynaptic event is triggered earlier.

The effect is frequently modeled by an exponential dependency of the weight change on the timing of pre- and postsynaptic spikes, with variations on, e.g., whether the weight change is additive, multiplicative or whether the spike pairing is considered in an all-to-all or nearest-neighbor fashion (Morrison et al., 2007).

1.6 Compartmental models

The neuron models described in section 1.2 and section 1.3 above reduce the spatial structure to a single point for which the membrane potential is evaluated; they are thus called *point neuron models*. Biological neurons display a complex morphology with branching dendritic trees and dendrite diameters which are far smaller than the distance between the soma and the distal end of the dendrite. For a single, linear section of dendrite, the cable equation (cf. Gerstner et al., 2014, sec. 3.2) is

an adequate, analytically tractable model. For investigations of complex branching topologies or nonlinear terms in the transmembrane conductance, compartmental models are a useful approach (Gerstner et al., 2014, sec. 3.2.3, 3.4). The voltage along the branching neuron is discretized into a finite number of elements, e.g. segments of dendrites. The membrane voltage $V_{i\mu}$ at compartment μ of neuron i is governed by the differential equation

$$C_{i\mu}\frac{\mathrm{d}}{\mathrm{d}t}V_{i\mu} = -g_{T,i\mu}\cdot V_{i\mu} + \sum_{\nu} g_{L,i\mu\nu}\cdot\left(V_{i\nu} - V_{i\mu}\right) + I_{i\mu} \tag{1.16}$$

with the membrane (transversal) conductance $g_{T,i\mu}$, the core (longitudinal) conductance $g_{L,i\mu\nu}$ between compartments μ and ν and the compartment input current $I_{i\mu}$ which can contain synaptic input, external stimulus or nonlinear membrane currents (Gerstner and Kistler, 2002, sec. 4.4.1).

1.7 Accelerated physical models

An alternative approach to numerical modeling of neural circuits has been pioneered by Carver Mead (Mead, 1990). Rather than solving the differential equations of the time evolution of the membrane (eq. (1.1)), a physical, electrical model is constructed that solves the differential equation due to its intrinsic dynamics (Mahowald and Douglas, 1991; Farquhar and Hasler, 2005; Arthur and Boahen, 2007). The result of the time evolution of voltages in the physical model is called *emulation* to differentiate it from the numerical simulation. One possible approach is to create devices with time constants that match the time constants of biological neurons (Benjamin et al., 2014; Livi and Indiveri, 2009). Because small capacitances and resulting RC time constants are comparatively easy to realize in very-large-scale integration (VLSI) circuits, an alternative approach is promising that implements an accelerated dynamics; the equations eq. (1.1) are kept the same but the dynamics of the physical model is 10^3 to 10^5 faster than that of the biological system. This is the approach taken by Schemmel et al. (2008), the system that is described later in chapter 2.

Formally, the hardware dynamics V_{hw} is scaled by an *acceleration factor* α_t as compared to the original. In addition, the voltage level and scale is arbitrary, so it can be shifted by an offset ω_v and scaled by a factor α_V, so the relation between hardware voltage V_{hw} and biological voltage V is:

$$V_{\mathrm{hw}}(t) := V(\alpha_t \cdot t)\cdot\alpha_V + \omega_v \tag{1.17}$$

The hardware capacitance C_{hw} also differs from typical biological values C. From this we can derive the scaling rules for currents and conductances of the hardware

device:

$$\frac{\mathrm{d}V_{\mathrm{hw}}}{\mathrm{d}t}(t) = \alpha_V \alpha_t \frac{\mathrm{d}V}{\mathrm{d}t}(\alpha_t t) \tag{1.18}$$

$$I_{\mathrm{hw}} = C_{\mathrm{hw}} \frac{\mathrm{d}V_{\mathrm{hw}}}{\mathrm{d}t} = \frac{C_{\mathrm{hw}}}{C} \alpha_V \alpha_t \cdot I \tag{1.19}$$

$$g_{\mathrm{l,hw}} = \frac{C_{\mathrm{hw}}}{C} \alpha_t \cdot g_{\mathrm{l}} \tag{1.20}$$

where I and g_{l} are the original, biological quantities. The scaling factor for time constants is evidently α_t. It is noteworthy that conductances do not scale with α_V but only with α_t.

1.7.1 Hardware devices

Different hardware devices are used and analyzed within this thesis. The nomenclature that refers to these devices is outlined in the following:

The HICANN chip is a neuromorphic chip that was developed in the FACETS and BrainScaleS projects in a 180 nm complementary metal–oxide–semiconductor (CMOS) process (Millner et al., 2010). The chip has a targeted acceleration factor α_t of 10^4.

The *BrainScales system* is a large-scale system, consisting of 384 HICANN chips and the required supporting infrastructure (Schemmel et al., 2008). The properties of this system are investigated in chapter 2.

The successor of the BrainScaleS system, BrainScaleS 2, is based on the HICANN DLS chip, implemented in a 65 nm process (Friedmann et al., 2016; Hock et al., 2013; Schemmel et al., 2017). The targeted acceleration factor is 10^3. This chip is referred to as HICANN DLS, or DLS in short, and the neuron architecture for its third prototype is examined in chapter 3.

1.8 Sampling with leaky integrate-and-fire neurons

A method to transfer the machine learning concept of Boltzmann Machines to networks of spiking neural networks has been recently proposed by Petrovici et al. (2016). A Boltzmann Machine (Ackley et al., 1985) is a stochastic network of binary units. The probability distribution of the state vector of N units $\mathbf{z} = \{z_1, z_2, \ldots\} \in \{0, 1\}^N$ is given by

$$p(\mathbf{z}) = \frac{1}{Z} \exp\left[\frac{1}{2}\mathbf{z}^T \mathbf{W} \mathbf{z} + \mathbf{z}^T \mathbf{b}\right] \quad . \tag{1.21}$$

Here, Z is the partition function that normalizes the sum of the probability of all possible states to one. The probability distribution is parameterized by $\mathbf{W}$, the connection matrix and $\mathbf{b}$, the bias vector. This is exactly the probability distribution of a physical system with the states $\mathbf{z}$ in the canonical ensemble, the Boltzmann distribution

$$p(\mathbf{z}) = \frac{1}{Z} \exp\left[-\beta E(\mathbf{z})\right] \tag{1.22}$$

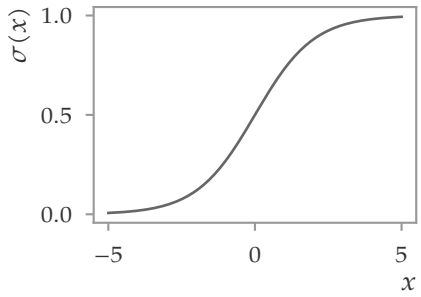


Figure 1.4: $\sigma(x) = \frac{1}{1+\exp(-x)}$

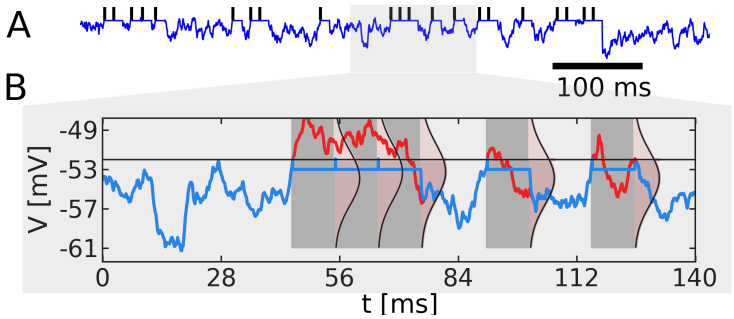


Figure 1.5: Dynamics of an LIF neuron model configured for LIF sampling (Petrovici, 2015). **A**: membrane potential and resulting spike train. **B**: Magnified excerpt of the voltage time course in A. The membrane voltage of the neuron (blue) follows the free membrane potential (red), except during spiking, where it is clamped to the reset potential. Taken from Petrovici (2015, Figure 6.31)

if one chooses β as $\frac{1}{kT} = \beta = 1$ and defines the energy as the negative argument of the exponential function in eq. (1.21):

$$E(\mathbf{z}) = -\frac{1}{2}\mathbf{z}^T\mathbf{W}\mathbf{z} - \mathbf{z}^T\mathbf{b} \quad . \tag{1.23}$$

This relation to statistical physics gives the model its name.

An intuitive relation to neural networks can be established by calculating the conditional probability of unit j being on given the remaining state of the network

$$p(z_j = 1 \mid \mathbf{z}_{\setminus j}) = \frac{p(\mathbf{z}_{\setminus j}, z_j = 1)}{p(\mathbf{z}_{\setminus j}, z_j = 0) + p(\mathbf{z}_{\setminus j}, z_j = 1)} = \frac{\exp(u_j)}{1 + \exp(u_j)} \tag{1.24}$$

$$= \sigma(u_j) \tag{1.25}$$

defining the sigmoid function (fig. 1.4)

$$\sigma(x) := \frac{1}{1 + \exp(-x)} \quad . \tag{1.26}$$

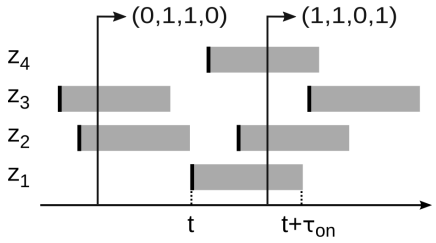


Figure 1.6: Interpretation of neuronal activity as sampling from a binary distribution. After a neuron fires it is considered active for a time τ_{on}, which corresponds to the refractory period in fig. 1.5. At any point in time, the state vector is the vector of active neurons (arrows). Taken from Petrovici (2015, Figure 6.25)

u_j is given by

$$u_j = b_j + \sum_{k=1}^{N} w_{jk} z_k \quad . \tag{1.27}$$

Thus, the probability of unit j to be active is a nonlinear function σ of the linearly weighted activity of the rest of the network.

The numerical solution of eq. (1.21) becomes impractical for even moderate N due to the requirement of representing the probabilities for each of the 2^N states. An alternative representation of the probability distribution is the sampling of states. A process is defined that produces samples from $\{0,1\}^N$ with a probability that corresponds to eq. (1.21). Markov chain Monte Carlo (MCMC) methods can be used for this purpose. Here, a Markov chain is constructed which has the desired probability distribution as its stationary distribution. A common choice is Gibbs sampling (Geman and Geman, 1984) which updates individual variables z_j using the marginal distribution (eq. (1.24)) to update the full vector of random variables. This sampling method in conjunction with eq. (1.27) reminds of a stochastic neuron model with membrane potential u_j and activation probability $\sigma(u_j)$. The sampling representation has advantageous properties: Only those states that have a comparatively high probability, and thus are relevant during the evolution of the system, are represented. Further, the computation of marginal distributions is elementary, as one can omit the unneeded variables during the collection of samples. Clamping parts of the system allows to calculate the conditional distribution of the remaining system.

An approach to map the system to more realistic stochastic neuron models incorporating long refractory times and relative refractoriness is described in Buesing et al. (2011) and is not discussed here in detail. It is one of the foundations for Petrovici et al. (2016) to implement Boltzmann Machines using deterministic LIF neurons. Here, input from background Poisson processes provides stochasticity to each neuron. Figure 1.5 shows the behavior of an LIF neuron in the LIF sampling regime. The neuron is stimulated by Poisson input such that the total conductance exceeds the leak conductance and thus lowers the effective time constant. This can

be understood by rewriting eq. (1.1) with conductance-based synapses eq. (1.11) as

$$\begin{aligned} C_\mathrm{m}\frac{\mathrm{d}}{\mathrm{d}t}V_\mathrm{m} &= -g_\mathrm{l}(V_\mathrm{m}-V_\mathrm{leak}) \\ &\quad - g_{\mathrm{syn,E}}(t)(V_\mathrm{m}-E_\mathrm{rev,E}) \\ &\quad - g_{\mathrm{syn,I}}(t)(V_\mathrm{m}-E_\mathrm{rev,I}) \qquad (1.28) \\ &= \underbrace{\left(g_\mathrm{l}\cdot V_\mathrm{leak} + g_{\mathrm{syn,E}}(t)\cdot E_\mathrm{rev,E} + g_{\mathrm{syn,I}}(t)\cdot E_\mathrm{rev,I}\right)}_{g_\mathrm{tot}(t)\cdot V_\mathrm{eff}(t)} \\ &\quad - \underbrace{\left(g_\mathrm{l} + g_{\mathrm{syn,E}}(t) + g_{\mathrm{syn,I}}(t)\right)}_{g_\mathrm{tot}(t)}\cdot V_\mathrm{m} \qquad (1.29) \\ &= g_\mathrm{tot}(t)\cdot V_\mathrm{eff}(t) - g_\mathrm{tot}(t)V_\mathrm{m} \qquad (1.30) \end{aligned}$$

defining

$$g_\mathrm{tot}(t) := g_\mathrm{l} + g_{\mathrm{syn,E}}(t) + g_{\mathrm{syn,I}}(t) \qquad (1.31)$$

$$V_\mathrm{eff}(t) := \frac{g_\mathrm{l}\cdot V_\mathrm{leak} + g_{\mathrm{syn,E}}(t)\cdot E_\mathrm{rev,E} + g_{\mathrm{syn,I}}(t)\cdot E_\mathrm{rev,I}}{g_\mathrm{tot}(t)} \qquad (1.32)$$

$$\tau_\mathrm{eff}(t) := \frac{C_\mathrm{m}}{g_\mathrm{tot}(t)} \qquad (1.33)$$

one obtains the time evolution of the membrane potential

$$\tau_\mathrm{eff}(t)\frac{\mathrm{d}}{\mathrm{d}t}V_\mathrm{m} = V_\mathrm{eff}(t) - V_\mathrm{m} \quad . \qquad (1.34)$$

In the case of current-based synapses, the effective membrane time constant does not change with the input, and the equilibrium potential is

$$V_{\mathrm{eff,curr}}(t) = V_\mathrm{leak} + \frac{I_\mathrm{syn}}{g_\mathrm{l}}(t) \qquad (1.35)$$

and the time evolution of the system is given by

$$\tau_\mathrm{m}\frac{\mathrm{d}}{\mathrm{d}t}V_\mathrm{m} = V_\mathrm{leak} + \frac{I_\mathrm{syn}}{g_\mathrm{l}}(t) - V_\mathrm{m} \quad . \qquad (1.36)$$

As the Poisson input rate increases, the equilibrium distribution of $V_{\mathrm{eff,curr}}$ (or g_syn in the conductance based case) converges to a normal distribution. This can be seen from the central limit theorem – for a proof see, e.g., (Petrovici et al., 2016, 4.3.2) – or by approximating the input current as Ornstein-Uhlenbeck process (Ricciardi and Sacerdote, 1979). The step that joins the sampling representation of eq. (1.21) and the activity of LIF networks is shown in fig. 1.6: Each neuron is defined to be active during its refractory period τ_ref and the set of the activity vectors from the time evolution of the network constitutes a sampling representation of the underlying probability distribution. The spike-based interaction between different neurons

raises (or lowers) the total input current shifting the membrane potential distribution in analogy to the effect of active variables in eq. (1.27) on any given variable j.

In fig. 1.5, further features of the setup are seen. The free membrane potential (red curve) is faithfully followed by V_{m} due to a low effective time constant τ_{eff}. The refractory period of the neuron is close to the synaptic time constant. This choice facilitates the matching of the internal on-state of a neuron, given by the refractory period τ_{ref}, and the on-state seen by other neurons in the network, given by the exponential decay of the post-synaptic current with the time constant τ_{syn}. For a more detailed investigation of, e.g., different PSP kernels or a variation of time constants we refer to (Petrovici et al., 2016, 6.5).

Chapter 2

Compensation of network-level effects on a neuromorphic platform

The BrainScaleS system is a large-scale, accelerated neuromorphic device that was created to investigate the dynamical and computational properties of networks of spiking neurons. At its core lies a full-custom VLSI implementation of abstract neuron models and a communication infrastructure that connects these neurons. The design of the system results from a trade-off between two goals: first, providing a versatile and configurable substrate for the investigation of spiking neural networks, and second, to establish an implementation which is fast and efficient in energy and cost. The implemented neuron models and the connectivity is developed to accommodate biological parameter ranges (Millner et al. (2010); Fieres et al. (2008)). On the other hand, the implementation must adhere to strict limits concerning die area of the involved components and their power consumption, to pose a viable alternative to conventional computing architectures with regard to the simulation of spiking neural networks. The outcome of the trade-off manifests in a large speed-up in comparison to biological real time of approximately 10^4 which comes at the cost of certain limitations: Neuron and synapse parameters have finite precision. For digital parameters this is caused by discretization of values. For analog parameters, fixed-pattern variability occurs due to device mismatch during chip manufacturing. The communication infrastructure that transports spike events also underlies constrains in terms of the number of possible realizable connections and in the bandwidth that is available for spike data.

A large effort is put into making the system accessible for non-hardware-experts: The top-level software interface for users is *PyNN* (Davison et al. (2008)), an application programming interface (API) that allows defining networks of spiking neurons and execute that definition on several simulation back ends, the BrainScaleS system being one of them. The back end includes algorithms to map the abstract network description into a configuration of the hardware device (Brüderle et al. (2011)). Additional software layers handle the communication with the system to actually perform the

configuration, transport input data to the device and return the emulation result to the user (see Brüderle (2009); Jeltsch (2014); Müller (2014)). The translation between biological and hardware time and voltage domains (section 1.7) is also handled at this point. Issues of variability are addressed by applying a *calibration*, a collection of methods to measure device properties on the level of individual sub-circuits and translate the desired neuron properties entered by the user to a corresponding hardware configuration. Typically, even after calibration, some variability remains which must be taken into account by the user of the system (see, e.g., Brüderle et al. (2011); Schwartz (2013); Koke (2017)). In the end, the user of the neuromorphic device has to deal with distortions due to limited communication resources, limited bandwidth and parameter precision in the design of their network.

During the production and commissioning phase of the BrainScaleS system, the impact of these limitations was analyzed in Petrovici et al. (2014) using a bottom-up approach. Three network models, which are derived from previous studies, are used to investigate the effect of idealized distortion mechanisms on their network dynamics. Compensation strategies that restore the original functionality of the network are proposed and tested in simulation. As a final step, all compensation mechanisms are verified using the Executable System Specification (ESS), a simulated hardware back end. Because the ESS faithfully represents the behavior of digital components of the hardware device and is interfaced using the same software as the hardware device, this serves as test of the combined stack of configuration software, mapping algorithms and hardware constraints.

Contribution

In the following, the analysis of two of the networks is presented which was carried out in large part by the author of this thesis. The analysis of the random network with self-sustained activity was performed in equal parts with Bernhard Vogginger and Lyle Muller; all ESS simulations were performed by Bernhard Vogginger.

2.1 Wafer-scale neuromorphic hardware

2.1.1 Hardware implementation

The BrainScaleS system is the reference neuromorphic system that is being used for the study in this chapter. It is an accelerated, continuous-time, mixed-signal neuromorphic system. At the core of the system lies the HICANN chip (fig. 2.1 A). It contains 512 neuron circuits which implement the AdEx neuron model (eq. (1.4)) as analog electronic circuits (Millner et al. (2010)). The intrinsic RC time constants of these circuits are approximately 10^4 smaller than those of biological neurons. Each neuron circuit receives 24 adjustable parameters as analog currents and voltages, which are used to control the neuron's properties. The parameters are provided by an array of floating gate cells (Srowig et al., 2007; Ehrlich et al., 2007; Millner, 2012).

A B

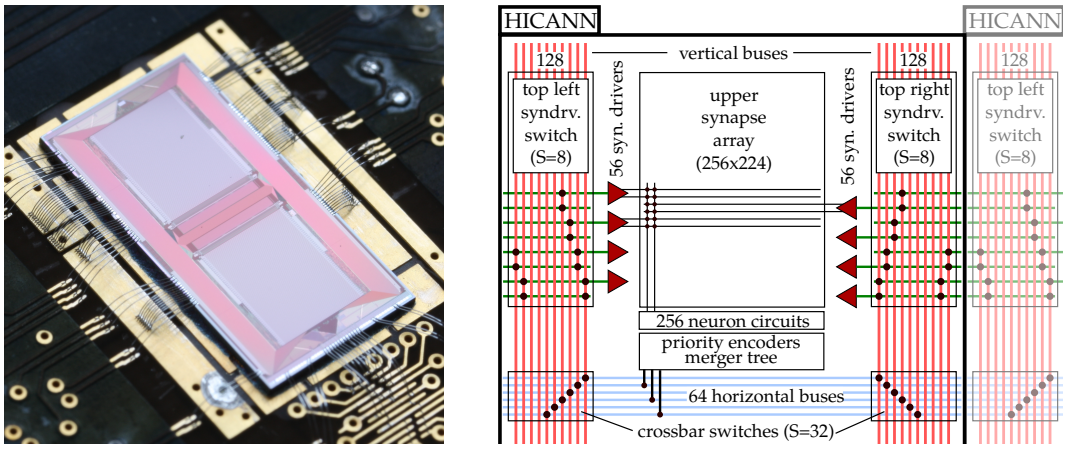


Figure 2.1: **A**: Photograph of a bonded HICANN chip die (taken by Matthias Hock). **B**: Schematic representation of components and connectivity of the HICANN building block, and its interconnection to other HICANNs within a wafer module. (The upper half of the chip is represented.) Most of the chip area is occupied by the synapse array. Each synapse column is connected to one of 256 neuron circuits, from which up to 64 can be interconnected to form larger neurons with up to 14336 input synapses. The horizontal and vertical buses transport spike events in the form of 6-bit addresses, which means they can be used to deliver spikes of up to 64 different sources. Sparse, statically configurable switches connect horizontal and vertical bus lines (called *crossbar switch*es) and vertical bus lines with the input of the synapse drivers. The synapse driver and the synapses filter incoming events according to their 6-bit address, so that each synapse forwards events from exactly one source address. The layer 1 (L1) buses of adjacent HICANNs are interconnected with repeaters, which enables the wafer-wide signal transport by L1. The figure on the right and corresponding caption are modified with permission from (Petrovici et al., 2014, Figure 3).

A large part of the chip area is occupied by communication infrastructure which transports spikes emitted by neuron circuits to other neurons within the system (fig. 2.1 B). Each neuron circuit is connected to a column of 224 synapses in the synapse array which provide the input from the presynaptic partners of the neuron and store the corresponding synaptic weight. Adjacent neuron circuits (neighboring on one half of the chip or opposing on the upper and lower half) can be interconnected to form a larger logical neuron to increase the number of synapses per neuron. When a neuron fires, it emits a signal with a 6-bit address which is merged with the output of other neurons onto a horizontal L1 bus. An L1 bus can be connected to a crossing bus using statically configurable sparse switches (denoted by black dots in fig. 2.1 B). A vertical L1 bus can be connected to synapse drivers, which pass the events into the synapse circuits. The drivers and synapses filter the incoming address packets so that

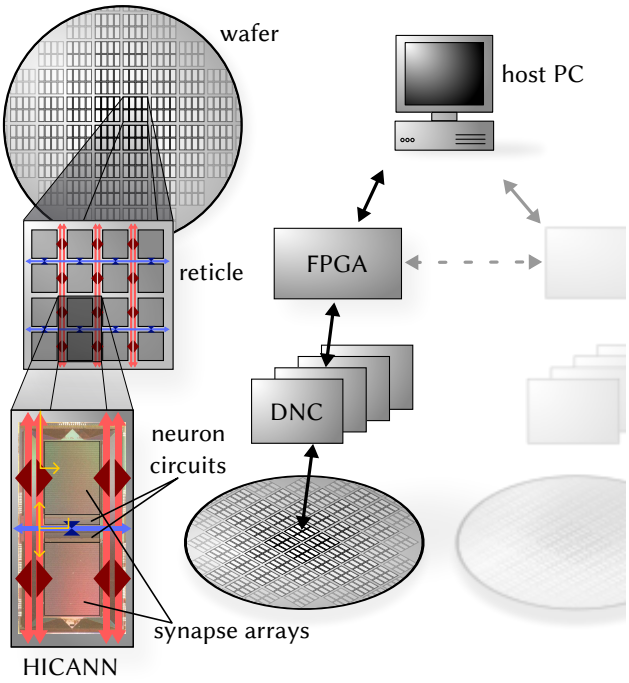


Figure 2.2: Wafer-scale integration and off-chip communication. Left: The L1 connections span over the whole wafer. For this, chips on different reticles, the largest unit of the photolithographic process, are electrically connected in a post-processing step. Right: Off-wafer connectivity by dedicated DNC chips (Ehrlich et al. (2007); Scholze et al. (2010)) and FPGAs to host computers controlling the experiment. Used with permission from (Petrovici et al., 2014, Figure 2)

in general, each synapse transports events from one presynaptic to one postsynaptic partner. The scalability of the architecture stems from wafer-scale integration: After production on a silicon wafer, the HICANN chips are not cut apart and the wafer is left intact (fig. 2.2). The HICANN chips are connected by additional wiring using a post-processing step, making it possible to transmit L1 packets between neighboring chips (Schemmel et al. (2008)). Thus, the static switches described above can be used to configure outgoing trees of connections that span the whole wafer. The realizable internal and external connectivity poses complex constraints due to the limited number of buses per chip, the number of synapse drivers and synapses, and the sparseness of switches which connect vertical bus lanes to horizontal buses and synapse drivers.

2.1.2 Supporting software

The software framework for the hardware system handles these constraints by automating the mapping of network architectures to the hardware substrate. It is interfaced using the network description API, *PyNN* (fig. 2.3), which also has several software-based simulator back ends. A general description of a spiking network

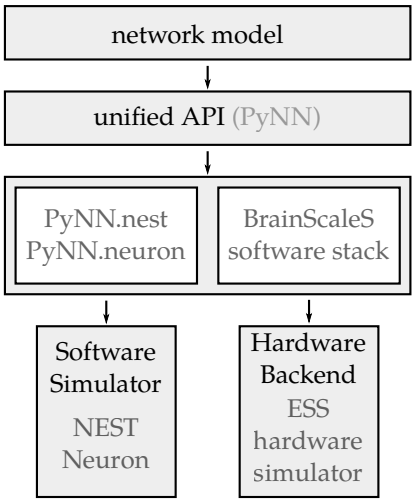


Figure 2.3: Software used for the analysis. The software simulators NEST (Gewaltig and Diesmann (2007)) and NEURON (Hines and Carnevale (2003)) and the BrainScaleS system simulator ESS are both interfaced using the *PyNN* API (Davison et al. (2008)).

is mapped to the hardware by first allocating neuron circuits for each neuron in the abstract network (*placement* step). Then, connections between the neurons are realized in the *routing* step. It can occur that some synapses can not be realized due to limited resources. These synapses effectively disappear from the network; this effect is thus called *synapse loss* from here on. For complex networks, the loss does not start abruptly with an increase in the number of neurons, but a small percentage of synapses can not be realized even for comparatively small networks, the proportion rising slowly with network size (see fig. 2.17 A), so it is not feasible to treat this effect as an error. Within the following study, synapse loss is sometimes enforced by using only a part of a wafer to test the presented compensation strategies with a network of a given size.

The ESS (Brüderle et al., 2011; Ehrlich et al., 2007; Vogginger, 2010) is an essential part of the investigation as the part of the software that simulates the hardware system. It provides a detailed, executable model of the spike transmission in individual hardware modules for on- and off-wafer communication. It further faithfully maps the configuration space of the hardware such as the interconnection topology, parameter discretization and sharing of parameters. The neuron circuits are represented numerical models of the ideal AdEx equation. In the simulations described below, parameter variations are artificially imposed for synaptic weights but not for neuron parameters.

2.2 Structure of the analysis

Figure 2.4 shows an outline of the analysis workflow. The investigated network models are selected and for each network, a set of functionality criteria is defined that

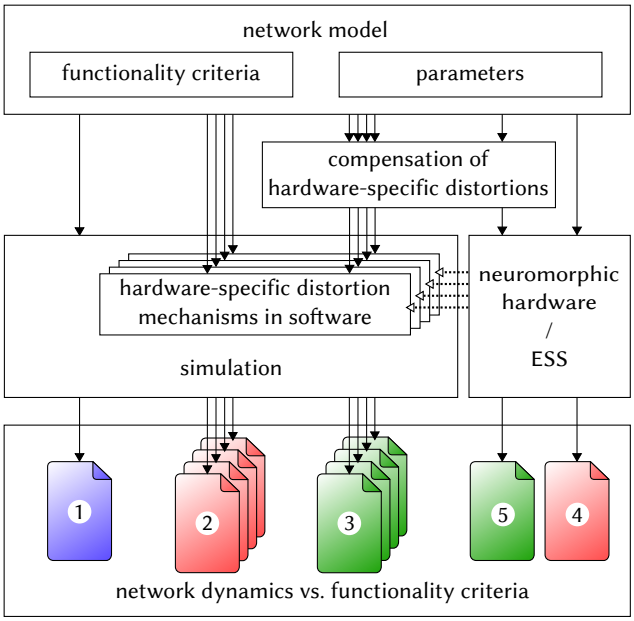


Figure 2.4: Outline of the distortion analysis workflow. A detailed description is given in section 2.2. Used with permission from (Petrovici et al., 2014)

capture essential characteristics of those models (1). The most important distortion mechanisms for emulating networks are identified and modeled in an abstract way. The effect of each distortion mechanism on the functionality criteria is investigated in simulation (fig. 2.4 2) using a software simulator back end (fig. 2.3). Compensation methods are developed and verified in simulation for the distortion mechanisms individually (3). In a last step, a simulation with all modeled distortions using the ESS is conducted, once without and once with all compensation mechanisms.

The distortion mechanisms that typically occur on mixed-signal neuromorphic devices are:

1. Fixed neuron and synapse models, i.e., only LIF and AdEx neurons with conductance-based synapses are available on the device

2. Limited parameter ranges

3. Limited number of components, such as neurons, synapses or spike routing resources.

4. Non-configurable transmission delays

5. Bandwidth constraints within the wafer and for external spike stimulus and readout.

The following three effects are isolated as being the most generalizable, i.e., which are expected to be present in most neuromorphic devices with analog components, and are expected to significantly affect most network models:

1. *Synapse loss* is modeled by randomly deleting a fraction of synapses in a network. The magnitude is denoted as percentage, which corresponds to the probability that a synapse is lost (p_{loss}).

2. Non-configurable delays, implemented by enforcing a constant delay of 1.5 ms in the biological time domain, the mean expected value of delay on the wafer system.

3. Synaptic *weight noise*, which models a fixed-pattern variation of the strength of synaptic weights due to transistor mismatch and weight discretization. The value is implemented as Gaussian random variable with the mean of the original synaptic weight and standard deviation proportional to the original weight. (Negative samples can occur and are clipped to zero.)

For ESS simulations, the synapse loss and delay constraints result implicitly from the realistic model of component behavior and routing constraints, while the weight noise is implemented as described in the list above.

2.3 Simulation results

In this section we describe two of the network models that were investigated in Petrovici et al. (2014). First we outline the functionality criteria which are derived from the dynamics of the undistorted network. Then, the effects of the distortion mechanisms and possible compensation methods are analyzed.

2.3.1 Synfire chain with feed-forward inhibition

Network description

The first model that is being investigated is a feed-forward chain based on the study presented in Kremkow et al. (2010). One of the core points of that study is the effect of feedforward inhibition on signal propagation through consecutive groups of neurons. Disynaptic inhibition, where excitatory connections trigger inhibitory neurons, is a connection scheme that has been observed in cortical structures (Silberberg and Markram (2007)), and is the mechanism of feedforward inhibition in the network model. On a functional level, it has been shown experimentally that stimulus in cat visual cortex evokes excitation shortly followed by inhibition, recorded hundreds of micrometers from the stimulus site Hirsch and Gilbert (1991).

Figure 2.5 A shows the structure of the network and the simulation setup. The network consists of consecutive groups, each group containing 100 regular spiking (RS) and 25 fast spiking inhibitory (FS) cells. All cells are modeled using LIF neurons with identical parameterization (table A.1). The disynaptic, feedforward inhibition scheme is implemented as follows: Neurons from the excitatory population within each group project onto both populations of the consecutive group, while the inhibitory population projects within the group only. The connectivity is dense, with 60 connections from the RS and 25 from the FS population to each RS population

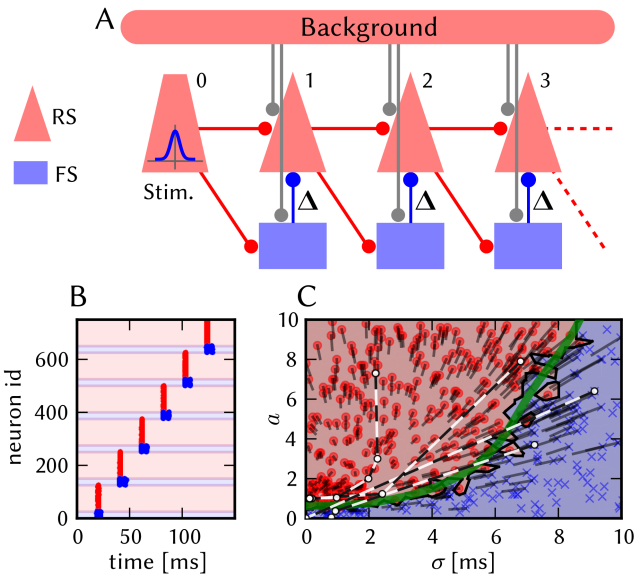


Figure 2.5: **A**: Schematic representation of excitatory (regular spiking (RS)) and inhibitory (fast spiking inhibitory (FS)) populations within the synfire chain network model. **B**: Example propagation of a pulse packed through individual groups of the synfire chain. **C**: Evolution of the strength and width of a pulse packet as it propagates through the chain, depending on the initial parameters. The marker is placed corresponding to the properties of the initial stimulus pulse, (σ_0, a_0). The evolution of the pulse packet properties is shown exemplarily for four stimulus parameters as black-and-white lines. Each marker is colored according to the activation of the last group in the simulation, red for $a_6 \geq 1$, blue for $a_6 = 0$ and linearly interpolated between the two colors for $0 < a_6 < 1$. To improve visibility, the background is colored using the same color as the nearest point. The green line represents a fit to the boundary between the region of stable and unstable propagation. Used with permission from (Petrovici et al., 2014, Figure 13)

(table A.2). The local delay Δ from the FS to the RS population is an important quantity that affects the functionality of the network (fig. 2.8); it is set to 4 ms in the default case. The inter-group delay is not relevant for the functionality, because there are no feedback connections. This delay only affects the time shift of the neuron's responses, and is set to 20 ms for visualization purposes, as in Kremkow et al. (2010). The background stimulus is implemented as Gaussian background noise in the original model in Kremkow et al. (2010), and is replaced by spiking background that is adjusted to keep the original mean and variance of the membrane potential distribution. This was achieved by an input firing rate of 2 kHz with a synaptic weight of 1 nS.

The input to the network is provided by a population of 100 spike sources, which are connected identically to an RS population within the chain. This input population emits a test pulse that is parameterized as follows: Each neuron within the

population emits a_0 spikes. Each spike time is drawn from a Gaussian distribution with a common mean and standard deviation σ_0. The functionality criteria for this network are based on the propagation properties of this pulse packet through the chain (fig. 2.5 B): The mean number of spikes in group i is denoted a_i and the standard deviation of the spike times σ_i. In the default configuration, the input converges to $a = 1$ and σ close to zero, i.e., each neuron in the chain spikes exactly once and the firing is nearly synchronous. Due to this behavior, the network is referred to as *synfire chain* from here on. The dynamics of the convergence to this state is shown in fig. 2.5 C. The initial stimulus parameters a_0, σ_0 are drawn from a range of $[0, 10]$ resp. $[0\,\text{ms}, 10\,\text{ms}]$. Depending on the starting position in the (σ, a) space, the activity parameters either converge to the stable point as the pulse is passed through the network, or the propagation stops. Exemplary trajectories of the activation properties in successive groups are shown as black-and-white lines. In fig. 2.5 C, points representing initial parameters that lead to an activation of the last group (a_6) are colored red, otherwise blue. The region in the stimulus space that leads to stable propagation is sharply separated. The approximate separation line between the two regions, called *separatrix* from here on, is shown in green in fig. 2.5 C. The presence of stable propagation and the location of the separatrix are defined as functional criteria for this network.

Synapse loss

The effect of synapse loss on the behavior of the network is shown in fig. 2.6. Homogeneous deletion of synapses is applied to all internal connections as well as the synchronized stimulus, but not to the background. The propagation for a synchronous pulse with $a = 1$ fails between 40 % and 50 % synapse loss (panels A, C, E). Compensating this by scaling all weights with the ideal factor $\frac{1}{1-p_{\text{loss}}}$ restores propagation (panels B, D), while increasing the pulse width for high values of synapse loss (panel F). The separatrix is affected in a minor way in the distorted case (panel G) until the region of stable propagation disappears altogether. The location of the separatrix at $\sigma = 0$ moves to higher a values, until it reaches the fixed point at $a = 1$ and the stable region disappears (not shown). In the compensated case, the separatrix location is not significantly affected by the distortion (panel H).

Weight noise

The effect of weight noise is shown in fig. 2.7. Panel A shows a representative stimulus state space for a high value of noise of 80 %. The primary cause of the distortion is the variation of the background stimulus weight. One solution to remedy the effect to keep the distribution of membrane potential values at a given point in time at its original, undistorted value. When the input spike volley arrives, the membrane voltages of the individual neurons are spread out, because they each receive different random background stimulus and because, due to weight noise, the magnitude of the background input varies. As the variation that is caused by synaptic weight heterogeneity increases, the variation caused by time-dependent

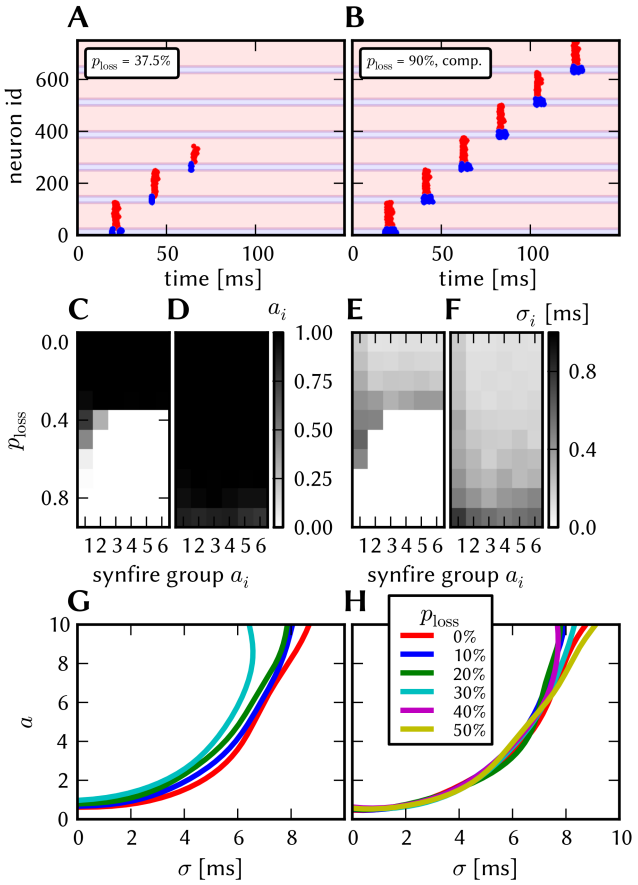


Figure 2.6: **A**: Pulse propagation in the synfire network with 37.5 % synapse loss, without compensation. **B**: Pulse propagation in the network with 90 % synapse loss and active compensation by increased synaptic weights. **C**: Activation a_i in each group i for different values of synapse loss. **D**: Activation a_i in each group i for different values of synapse loss and with active compensation. **E**: Pulse width σ_i in each group i for different values of synapse loss. **F**: Pulse width σ_i in each group i for different values of synapse loss and width active compensation. **G**: Approximate separatrix locations for different values of synapse loss. For 40 % and 50 % synapse loss no stable region exists, so no line is present. **H**: Approximate separatrix locations for different values of synapse loss with active compensation. Used with permission from (Petrovici et al., 2014, Figure 14)

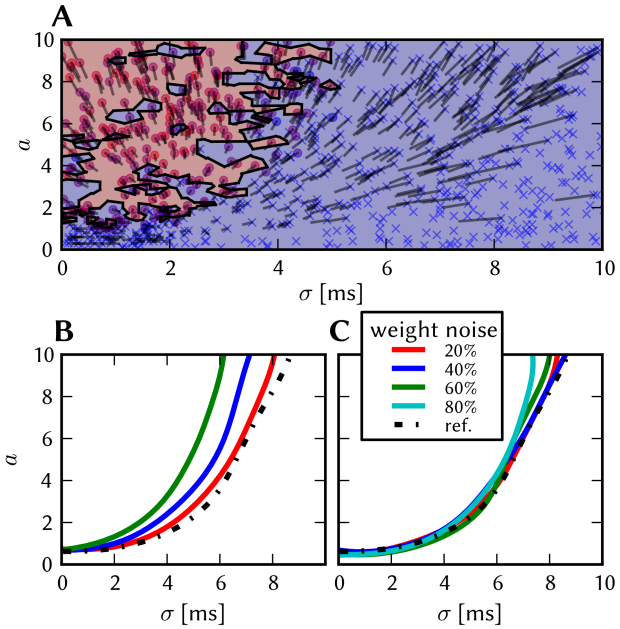


Figure 2.7: **A**: State space for 80 % weight noise. **B**: Approximate separatrix locations for smaller values of weight noise. **C**: Approximate separatrix locations for the compensated case. The separatrix from the reference simulation is shown as a dashed line. A spike filter with parameters $T = 10\,\text{ms}, N = 25$ is applied for these figures (cf., section A.1.2). A comparison with and without filter is shown in fig. A.1. Used with permission from (Petrovici et al., 2014, Figure 15)

Poissonian background stimulus is reduced. This can only work due to the short time during which the propagating pulse acts, so that the time-dependent variation of the background stimulus does play a significant role. The parameter V_{leak} is increased and the background synaptic weight is lowered to achieve this. The approximation of current-based neurons (linear addition of PSP kernels) is used to calculate the required factor. Then, the membrane potential of a neuron stimulated by background spikes can be approximated as

$$\begin{aligned} V_{\text{m}} &\approx w \cdot \sum_{j \in \text{backg. spikes}} \kappa(t - t_j) + V_{\text{leak}} \\ &= w \cdot K(t) + V_{\text{leak}} \quad , \end{aligned} \tag{2.1}$$

where κ is the post-synaptic kernel and $K(t)$ the sum of the kernels. The average over the time and weight distribution is then

$$\langle V_{\text{m}} \rangle \approx \langle w \rangle \cdot \langle K(t) \rangle + V_{\text{leak}}$$

The variance of w is given by $\text{Var}[w] = w_0^2 s^2$, where s denotes the relative weight noise and w_0 the original, undistorted weight. The variance of V_{m} then follows from

eq. (2.1)

$$\begin{aligned} \mathrm{Var}\left[V_\mathrm{m}\right] &\approx \langle w\rangle^2 \mathrm{Var}\left[K(t)\right] + \mathrm{Var}\left[w\right]\left(\langle K(t)\rangle^2 + \mathrm{Var}\left[K(t)\right]\right) && (2.2)\\ &= w_0^2\left\{\mathrm{Var}\left[K(t)\right] + s^2\left(\langle K(t)\rangle^2 + \mathrm{Var}\left[K(t)\right]\right)\right\} && (2.3) \end{aligned}$$

With increasing s^2, w_0 must decrease to keep $\mathrm{Var}\left[V_\mathrm{m}\right]$ constant. This changes $\langle V_\mathrm{m}\rangle$ which is compensated by modifying V_leak accordingly. Panels B and C of fig. 2.7 show the location of the separatrix in the distorted and compensated cases, demonstrating the viability of the method.

Non-configurable axonal delays

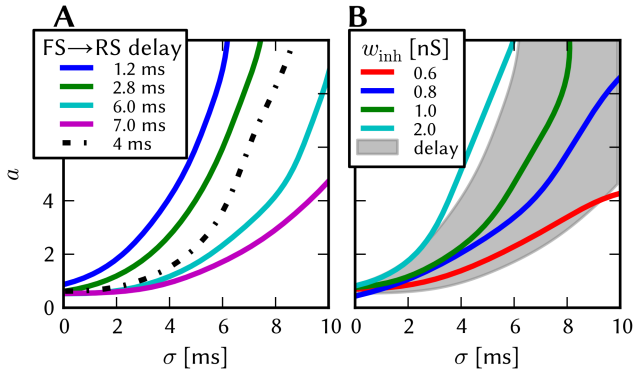


Figure 2.8: **A**: The location of the separatrix is controlled by the local axonal delay Δ. **B**: A comparable control is achieved by varying the inhibitory synaptic weight at a fixed delay of 1.5 µs. The inhibitory synaptic time constant is increased by a factor of three for this simulation. The gray region covers the extent of separatrix locations in A. Used with permission from (Petrovici et al., 2014, Figure 16)

Kremkow et al. (2010) show that the location of the separatrix can be changed by modifying the axonal delay of the local inhibitory projection from the FS to the RS population (shown in fig. 2.8 A). Diesmann (2002) also show that the location of the separatrix can be modified by other parameters, such as group size and noise level, in a synfire chain model without inhibition.

In the investigated model, stable propagation does not occur for short delays (0.1 ms), which can be countered by greater synaptic time constants and a lower synaptic weight for local inhibition. This prolongs the effect of local inhibition even though the onset at the target population occurs earlier. The mean delay value on the hardware at an acceleration factor of 10^4 is 1.5 ms. An alternative way of modifying the separatrix is provided, because the delay value on the hardware device can not be adjusted: The inhibitory synaptic time constant is increased by a factor of three and the inhibitory weight is used to modify the location of the separatrix (fig. 2.8 C).

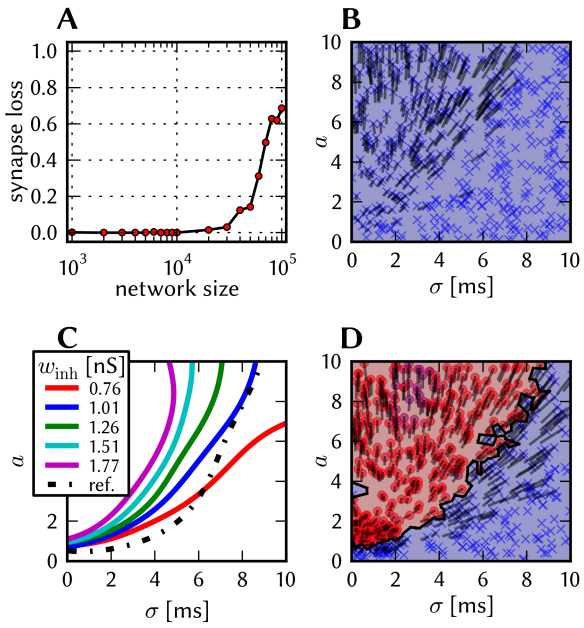


Figure 2.9: Distorted and compensated simulations of the feedforward synfire chain on the ESS: (**A**) Synapse loss after mapping the model with different numbers of neurons onto the BrainScaleS System. (**B**) (σ,a) state space on the ESS with default parameters, 20 % weight noise, and 27.4 % synapse loss. (**C**) After compensation for all distortion mechanisms, different separatrices are possible by setting different values of the inhibitory weight. (**D**) Compensated state space belonging to the blue separatrix in **C**. Figure and caption used with permission from (Petrovici et al., 2014, Figure 17)

Combined compensation on simulated hardware

As a final step, the network is simulated using the ESS including all distortion effects at once. The synapse loss that occurs when scaling the synfire network is shown in fig. 2.9 A. For this figure, the number of neurons per group as well as the number of groups in the chain is scaled while keeping the number of incoming synapses per neuron constant (see Petrovici et al., 2014, Table S3.3). Contrary to the case of a random network (fig. 2.17 A), loss occurs abruptly with increasing network size, starting at a network size of 30000 neurons. To better assess the compensation methods developed above, the number of available hardware resources is limited to 8 out of 48 reticles and make 50 percent of synapse drivers unavailable. This enforces mapping-induced loss at the original network size, with a total synapse loss of 27.4 % (table 2.1).

Several modifications are necessary to simulate the network on the ESS.

1. The speedup factor is reduced to 5000 to increase the effectively available bandwidth for the background stimulus.

2. The number of background stimulus sources was reduced from 750 to 192.

Table 2.1: Projection-wise synapse loss of the synfire chain model after the mapping process. Table and caption used with permission from (Petrovici et al., 2014, Table 2).

projection	**synapse loss [%]**
Pulse Packet → RS_0	21.3
Pulse Packet → FS_0	12.7
RS_n → RS_{n+1}	32.4
RS_n → FS_{n+1}	32.0
FS_n → RS_n	20.8
Poisson background → ALL	0
total	27.4

3. The input for each neuron was replaced by eight randomly chosen sources from the 192 background source pool.

The simulation is executed with a weight noise of $s = 20\,\%$ and resulting delays of 0.6 ms to 1.1 ms (due to the changed speedup factor). The network does not show a region of stable propagation in this setup, which is attributed to the low delays (fig. 2.9 B)

The synapse loss compensation has to be applied projection-wise due to the high heterogeneity of lost synapses (table 2.1). For the weight noise compensation, $\mathrm{Var}\,[w]$ must be reduced to $1/8\mathrm{Var}\,[w]$ in eq. (2.3) because background sources are distributed to multiple synapses, which have an independent amount of weight noise. The result of the successful compensation is shown in fig. 2.9 D. Panel C shows that the location of the separatrix can still be controlled by tuning the inhibitory synaptic weight.

Figure 2.9 D reveals two irregularities. The first is an area at high a within the region of stable propagation with a darker marker color, which indicates $a_6 < 1$. The second is one case where the propagation fails at $\sigma_0 = 0$, $a_0 \approx 4$, which never occurs in ideal simulations. The reason for both is the limitation of off-wafer bandwidth. The first effect of seemingly weak activation occurs because spikes are lost in the readout path, but all neurons within the last group fire on the simulated wafer. The second issue occurs because the input pulse was so synchronous that the input bandwidth did not suffice to transport it to the chip. This effect only occurs for spike pulses with $\sigma < 0.1$ ms.

2.3.2 Cortical network with self-sustained asynchronous activity in a random network

The second investigated model is a homogeneous network of two populations that displays self-sustained activity: after an initial stimulus, the firing within the network is sustained even in the absence of external input. The activity state is characterized as asynchronous and irregular (eqs. (2.4) and (2.5)). The network is thus referred to as *AI network* in short.

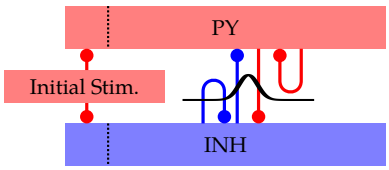


Figure 2.10: Architecture of the random cortical network. Used with permission from (Petrovici et al., 2014, Figure 18)

The asynchronous irregular activity regime is observed experimentally in cortex (Destexhe and Pare (1999); Destexhe et al. (2003)). The dependence of this activity state on the correlation dynamics of recurrent activity (Kumar et al. (2008); El Boustani and Destexhe (2009)) makes it an interesting test for the correlation structure that is introduced by synapse loss due to the mapping algorithm.

Figure 2.10 shows the architecture of the network which is based on Muller and Destexhe (2012); Destexhe (2009); Yger et al. (2011). Two populations of neurons are arranged on a lattice with a size of $1 \times 1\,\mathrm{mm}^2$ with periodic boundary conditions. The network contains 20 % fast spiking inhibitory (INH) and 80 % pyramidal (PY) cells which are modeled using the AdEx neuron model. The parameterization is shown in table A.3. The pyramidal (PY) neurons are modeled using spike-frequency adaptation (Connors and Gutnick, 1990) while for fast spiking inhibitory (INH) cells, only the sub-threshold adaptation parameter a is used to achieve weak adaptation. The connection probability is distance-dependent with a Gaussian profile ($\sigma = 0.2\,\mathrm{mm}$). The number of incoming connections is normalized to 200 excitatory and 50 inhibitory neurons. Two percent of the network are initially stimulated for a duration of 100 ms by individual Poisson sources with a mean firing rate of 100 Hz and a weight of 100 nS to initiate the activity. The synaptic delays in the reference model are distance-dependent, following $t_{\mathrm{delay}} = 0.3\,\mathrm{ms} + \frac{d}{v_{\mathrm{prop}}}$, with $v_{\mathrm{prop}} = 0.2\,\mathrm{mm\,ms}^{-1}$ and d being the distance between the two cells. The conduction velocity is that of unmyelinated horizontal fibers (Hirsch and Gilbert (1991); Murakoshi et al. (1993); Bringuier et al. (1999); González-Burgos et al. (2000); Telfeian and Connors (2003)). The resulting distribution of delays is shown in fig. 2.12. For the analysis of synapse loss (fig. 2.17), the following rule to scale the network is used: The number of neurons is increased while keeping the number of afferent synapses, the size of the cortical sheet and the distance-dependent connection probability constant.

The main functionality criterion for this network is the presence of a self-sustained activity in an asynchronous and irregular state. It is assessed by measuring the coefficient of variation $\mathrm{CV}_{\mathrm{isi}}$ and the correlation coefficient CC.

$$\mathrm{CV}_{\mathrm{isi}} := \frac{1}{N} \sum_{i=1}^{N} \frac{\sigma_i(\mathrm{ISI})}{\overline{\mathrm{ISI}_i}} \tag{2.4}$$

where $\sigma_i(\mathrm{ISI})$ is the standard deviation of the interspike intervals of the i-th spike train and $\overline{\mathrm{ISI}_i}$ is the mean interspike interval in the same spike train.

The correlation coefficient is defined as

$$\mathrm{CC} := \frac{1}{P} \sum_{j,k}^{P} \frac{\mathrm{Cov}(S_j, S_k)}{\sigma(S_j)\sigma(S_k)} \tag{2.5}$$

The sum runs over 5000 randomly chosen pairs of spike trains j and k of the excitatory population. S_i is the time-binned spike count in the i-th spike train with a bin width of 5 ms. $\sigma(S_i)$ denotes the standard deviation of S_i, and $\mathrm{Cov}(S_j, S_k)$ the covariance of S_j and S_k. $\mathrm{CV}_{\mathrm{isi}}$ is zero for a regular spike train and approaches 1 for a Poisson spike train; CC approaches zero for independent spike trains and is 1 for linearly dependent signals. For asynchronous irregular networks, $\mathrm{CV}_{\mathrm{isi}}$ should be greater or equal one (irregularity) and CC should be close to zero (asynchrony). As an additional functionality measure, the power spectrum of the excitatory activity is compared to the initial simulation. To compare the variability of firing rates within the network, the quantity $\mathrm{CV}_{\mathrm{rate}}$ is introduced. $\mathrm{CV}_{\mathrm{rate}} = \frac{\sigma(\nu)}{\overline{\nu}}$, where $\overline{\nu}$ and $\sigma(\nu)$ are the mean and standard deviation of the average firing rates ν of the inidividual neurons.

The behavior of the undistorted network is shown in fig. 2.11. The quantities are characterized in the $(g_{\mathrm{exc}}; g_{\mathrm{inh}})$ space of the recurrent synaptic weights. Panel A shows the region in the weight space for which the network displayed self-sustained activity. The minimum mean firing rate that is achieved in the self-sustaining regime is approximately 8 Hz (panels B and G). The activity is irregular ($\mathrm{CV}_{\mathrm{isi}} > 1$, panel C) and only weakly correlated ($\mathrm{CC} < 0.03$, panel D). The peak frequency of the power spectrum lies between 50 Hz to 100 Hz (in panel E in the $(g_{\mathrm{exc}}; g_{\mathrm{inh}})$ space and in panel J sorted by mean firing rate).

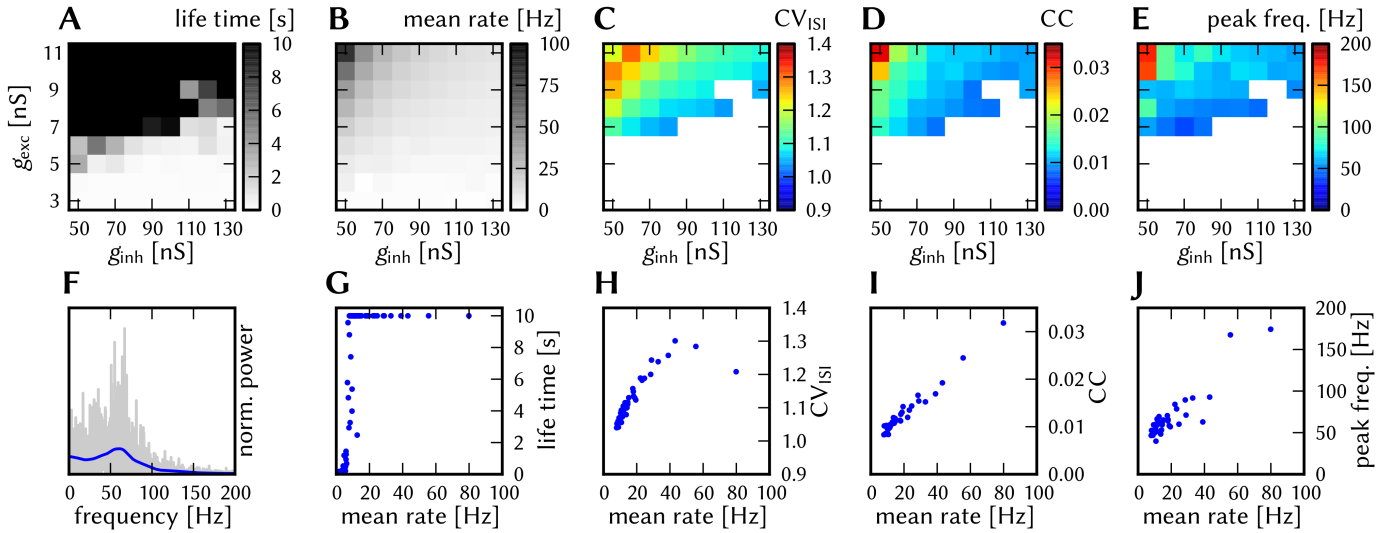


Figure 2.11: Behavior of the undistorted AI network. Top row: survival time (**A**), mean firing rate (**B**), coefficient of variance CV_{isi} (**C**), coefficient of correlation CC (**D**) and position of peak in power spectrum of global activity (**E**) in the parameter space for g_{exc} and g_{inh} for the default network with 3920 neurons without any distortions. (**F**) Power spectrum of the global pyramidal activity for the state ($g_{exc} = 9\,nS$, $g_{inh} = 90\,nS$). The population activity was binned with a time of 1 ms, the raw spectrum is shown in gray, the blue curve shows a Gauss-filtered ($\sigma = 5\,Hz$) version for better visualization. The position of the peak in the filtered version was used for (**E**). In (**G - J**) the dependence of single criteria on the mean firing rate is shown: survival time (**G**), CV_{isi} (**H**), CC (**I**), position of peak in power spectrum (**J**). In the last three plots only surviving states of the (g_{exc}, g_{inh}) space were considered. Figure and caption used with permission from (Petrovici et al., 2014, Figure 19). Simulations conducted by Bernhard Vogginger.

Non-configurable axonal delays

The average delay on the hardware system is 1.5 ms, which is very close to the mean delay in the undistorted reference simulation (1.55 ms). The mean value of the delay is deterministic for the dynamics of the model, while the distribution, to the extent that it is present in the given case, is not. In fig. 2.13, fixing the delay to 1.5 ms produces similar behavior as the default network (panels B, A). Similarly, a doubled delay leads to very similar results as a fixed delay of 3 ms (panels E, F). The ESS simulation produces comparable results to the cases in A and B (panel C).

It must be noted that, although the mean delay is accidentally the value that is present on the simulated hardware, the approach can be applied for different delay values as well: The acceleration factor can be changed so that the hardware delay matches the network mean delay. The limitation for this approach is only given by the limits of the time constants provided by the hardware; because the network is self-sustained there is no limitation due to external bandwidth.

Weight noise

The effect of the weight noise distortion mechanism is shown in fig. 2.14. The mean firing rate increases starting at a noise value of 20 %. CV_{rate}, the variance of firing rates, increases as well (panel B), which is expected because the homogeneity of the equal number of incoming synapses with equal weights is broken up. CV_{isi} is unchanged for low firing rates in all cases, but decreases for high firing rates with increasing weight noise. The global power spectrum is not significantly affected (not shown). Interestingly, the stability is increased: previously unstable points in the (g_{exc}, g_{inh}) space now lead to sustained activity (Panels C, G). The amount of synchrony decreases with increasing weight noise, weakly for low population firing rates and strongly for high ones (panel F).

Synapse loss

The effect of synapse loss is similar to that of weight noise. The mean firing rate increases, even stronger than in the case of weight noise (fig. 2.15 A). The region of stable activity increases as well (panels C, G) The heterogeneity of firing rates increases, which is indicated by the increased CV_{rate} (panel B). CC also decreases with increasing synapse loss. The firing mode stays asynchronous and irregular except for regions of the $(g_{exc}; g_{inh})$ state space with high firing rates above 40 Hz, where CV_{isi} drops below 1 (panel E).

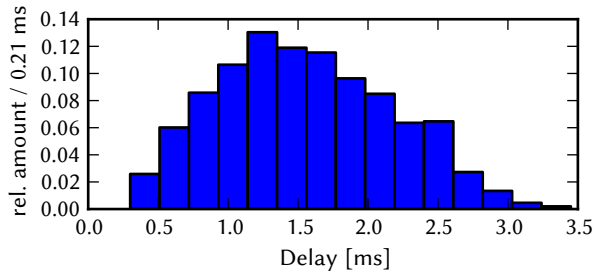


Figure 2.12: Histogram of delays in the AI network. The mean delay is 1.55 ms. Figure and caption used with permission from (Petrovici et al., 2014, Figure S4.1).

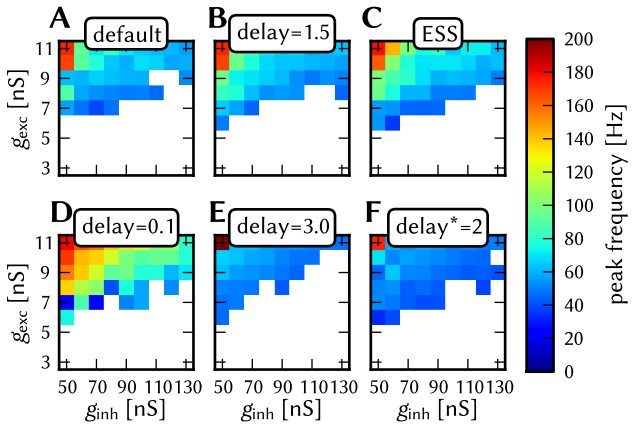


Figure 2.13: Effects of axonal delays on the AI network. (g_{exc}, g_{inh}) spaces with the peak frequency of the global pyramidal activity for different axonal delay setups: default with distance-dependent delays (**A**), constant delay of 1.5 ms (**B**), simulation on the ESS where delay is not configurable (**C**), constant delay of 0.1 ms (**D**), constant delay of 3.0 ms (**E**), distance-dependent delays scaled by factor of 2 with respect to default setup (**F**). Figure and caption used with permission from (Petrovici et al., 2014, Figure 4.1). Simulations by Bernhard Vogginger.

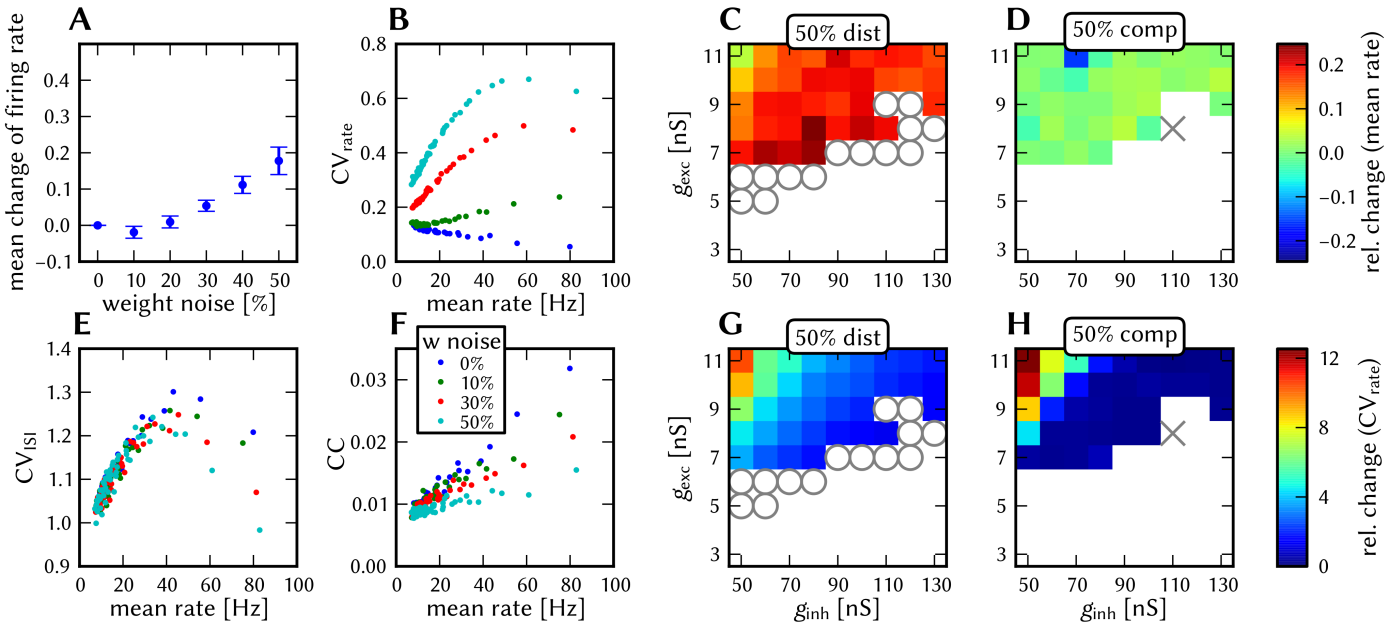


Figure 2.14: Effect and compensation of synapse weight noise in the AI network: (**A**) Relative change of the firing rate with respect to the undistorted network averaged over all sustained states for varying synapse weight noise. (**B**) CV_{rate} as a function of mean rate for every survived state for varying synapse weight noise. (**C** and **D**) Relative change of the firing rate with respect to the undistorted for each state for 50 % synapse weight noise(**C**) and compensated (**D**). (**E**) CV_{isi} as a function of mean rate for varying synapse weight noise. (**F**) CC as a function of mean rate for varying synapse weight noise. (**G** and **H**) Relative change of CV_{rate} with respect to the undistorted for each state for 50 % synapse weight noise(**G**) and compensated (**H**). In (**C** and **D**) and (**G** and **H**): A cross marks a state that was sustained in the undistorted but not sustained in the compared case. A circle marks a state that was not sustained in the original but sustained in the compared case. Figure and caption used with permission from (Petrovici et al., 2014, Figure 20). Simulations conducted by Bernhard Vogginger.

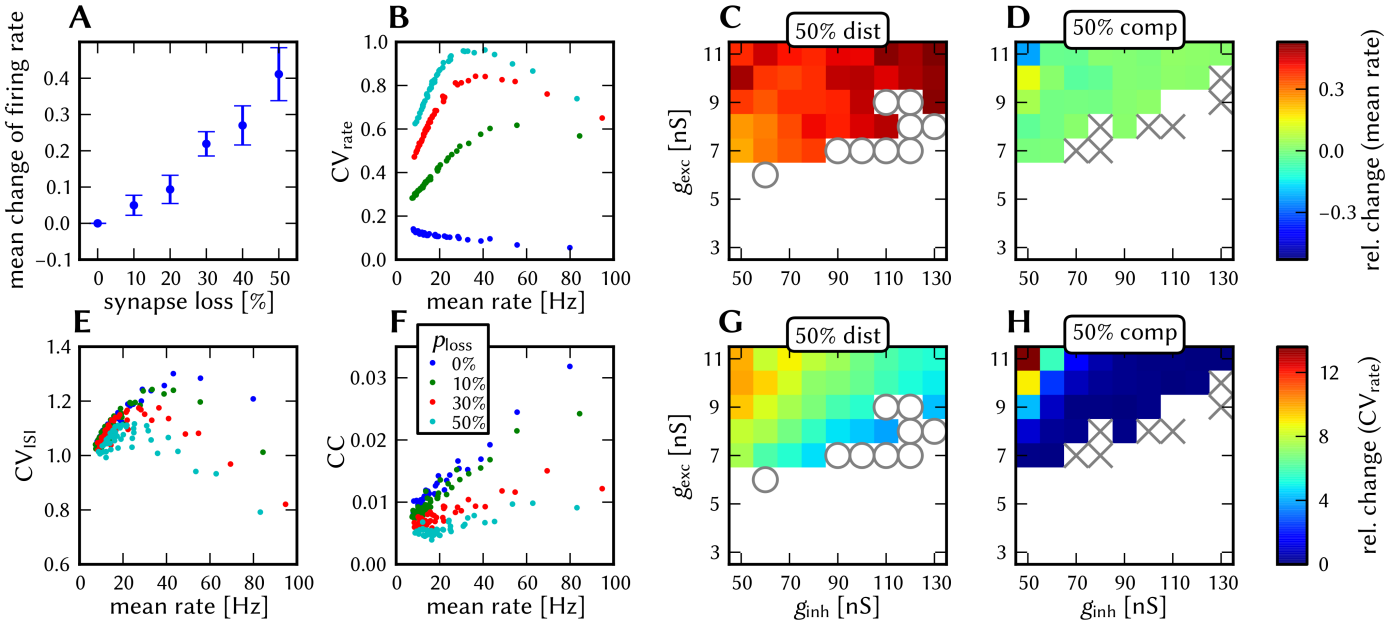


Figure 2.15: Effect and compensation of synapse loss in the AI network: (**A**) Relative change of the firing rate with respect to the undistorted network averaged over all sustained states for varying synapse loss. (**B**) CV_{rate} as a function of mean rate for every survived state for varying synapse loss. (**C**, **D**) Relative change of the firing rate with respect to the undistorted case for each state for 50 % synapse loss (**C**) and compensated (**D**). (**E**) CV_{isi} as a function of mean rate for varying synapse loss. (**F**) CC as a function of mean rate for varying synapse loss. (**G**, **H**) Relative change of CV_{rate} with respect to the undistorted case for each state for 50 % synapse loss (**G**) and compensated (**H**). In **C**, **D**, **G** and **H**: A cross marks a state that was sustained in the undistorted but not sustained in the compared case. A circle marks a state, that was not sustained in the original but sustained in the compared case. Figure and caption used with permission from (Petrovici et al., 2014, Figure 21). Simulations conducted by Bernhard Vogginger.

Compensation method based on mean-field approach

The compensation strategy for synapse loss is based on a mean-field approach. The idea is to scale all time constants within the network to restore the original firing rate. This can happen by simulating the distorted network once and recording the resulting network firing rate. In the case of the present network, however, the resulting network firing rate can be predicted from the behavior of a single neuron. Figure 2.16 A shows the output firing rate of an PY and an INH neuron with the same parameterization as used in the full network and stimulated by excitatory and inhibitory Poisson spike trains, simulating the activity seen by the neuron in the network context. In a mean-field approach the response of a neuron is assumed to be a function of the mean network firing rate:

$$\nu = f(\nu_{\text{in,exc}}, \nu_{\text{in,inh}})$$

Assuming the difference between the inhibitory and excitatory gain function is negligible, an assumption justified by fig. 2.16, the network firing rate is a self-consistent solution of

$$\hat{\nu} = f(N_{\text{exc}}\hat{\nu}, N_{\text{inh}}\hat{\nu})$$

where N_{exc} and N_{inh} are the number of pre-synaptic connections of a given neuron. In the presence of synapse loss, we approximate

$$\hat{\nu}(p_{\text{loss}}) = f\left(N_{\text{exc}}(1-p_{\text{loss}})\hat{\nu}, N_{\text{inh}}(1-p_{\text{loss}})\hat{\nu}\right) \tag{2.6}$$

To compensate for a given amount of synapse loss, we scale τ_{m}, $\tau_{\text{syn,E}}$, τ_{ref}, τ_{w} and the synaptic delays by a factor of

$$\alpha = \frac{\hat{\nu}(p_{\text{loss}})}{\hat{\nu}(0)}$$

The resulting mean network firing rate, with and without compensation, is shown in fig. 2.16 C, while the scaling factor α calculated from the neuron response is shown in fig. 2.16 B. The mean network firing rate is kept constant. At the same time, the variance of firing rates increases.

Iterative compensation

The mean-field-based compensation method restores the average firing rate within the network. An alternative method is required to reduce the variance that is introduced by synapse loss and weight noise. The proposed method relies on an iterative procedure to tune each neuron individually to restore the mean network firing rate and reduce the variability of firing rates at the same time. For each neuron, the spike initiation threshold V_{thresh} is modified proportionally to the difference between the actual firing rate ν_{act} and the target firing rate ν_{tgt}:

$$\Delta V_{\text{thresh}} = c_{\text{comp}}(\nu_{\text{tgt}} - \nu_{\text{act}})$$

Table 2.2: Statistics of the large-scale AI network. Reference (ref.) simulated with NEST, distorted (dist.) and compensated (comp.) with the ESS. Used with permission from (Petrovici et al., 2014, Table 3).

criteria	**ref.**	**dist.**	**comp.**
Rate [Hz]	13.4	15.5	13.6
CV_{rate}	0.107	0.726	0.212
CV_{isi}	1.12	1.11	1.09
CC	0.00103	0.0011	0.00166
Peak Frequency[Hz]	60.3	60.7	59.0

When the compensation factor is chosen appropriately (cf. (Petrovici et al., 2014, S4.3)), ten iterations are sufficient to restore the mean and reduce the variance of the firing rate in the network (fig. 2.17 C). Panels D and H in fig. 2.14 and fig. 2.15 show the result of this compensation method for the weight noise and synapse loss distortions, respectively. In the case of weight noise, the mean firing rate is recovered, while CV_{rate} only works well in the low $g_{exc}$and high g_{inh}region of the weight space. For synapse loss, the general effect is the same. In both cases, the stability of the network decreases again, after the increase caused by each distortion mechanism. For synapse loss, in particular, the distorted network is more stable and the compensated less stable than the reference.

Combined compensation on simulated hardware

In a final step, the compensation method is tested in an ESS simulation with combined distortions. The network is scaled up to 22445 neurons, leading to a total synapse loss of 28.1 % (fig. 2.17 A). CV_{rate} was significantly reduced from 0.726 to 0.212 but was twice as large as in the reference network (panel B), while the power spectrum (panel C) and the other criteria (table 2.2) match the reference simulation.

2.4 Summary

In this chapter we addressed the question of the effect of hardware-induced distortions on the emulation of networks on neuromorphic hardware. The primary effects that affect the model dynamics are identified and investigated in a systematic fashion. Network models were selected that differ largely in behavior: a cortical random network and a feedforward network that enables the propagation of synchronous spike volleys. For each network model, functional criteria were defined that characterize its performance. Three effects that occur during emulation on mixed signal neuromorphic hardware – variation of synaptic weights, synaptic loss due to mapping constraints and fixed delays – were investigated individually in simulation. Approaches to compensate the disruption due to these effects were developed and tested successfully in simulation. In a final step, a simulation of the

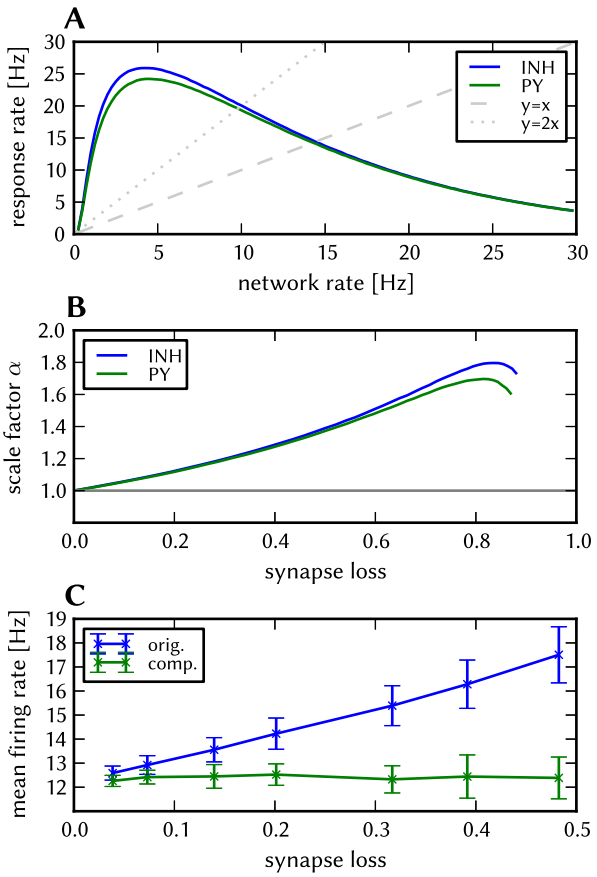


Figure 2.16: Mean-field-based compensation method for the AI network. **A**: Simulation of the mean firing rates of a single PY and INH neuron stimulated by Poisson background. The number and strength of the input synapses mimics the input that the neurons would receive when embedded in a network. **B**: Scale factor α, calculated from the data shown in A. **C**: Compensation applied to the self-sustained network (with parameters $g_{\text{inh}} = 90\,\text{nS}$, $g_{\text{exc}} = 9\,\text{nS}$). The error bars denote the standard deviation of mean firing rates across all neurons. "orig." marks the original network without compensation, in "comp." the neuron parameters were modified according to the compensation factor. Figure and caption adapted with permission from (Petrovici et al., 2014, Figure 22)

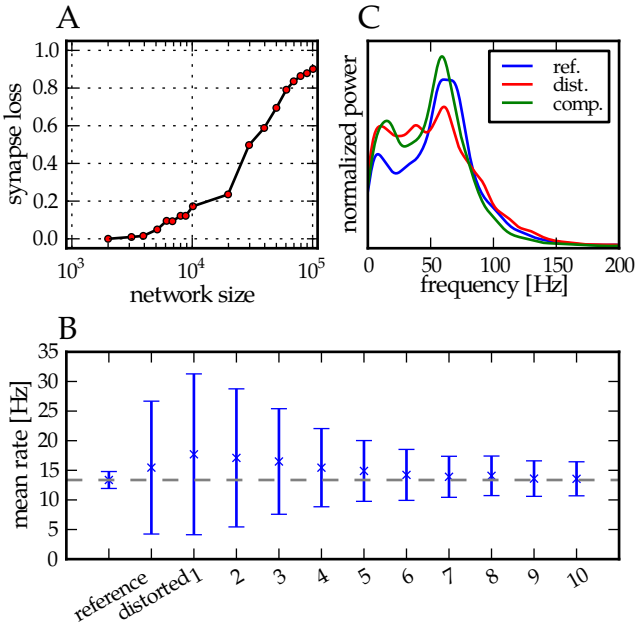


Figure 2.17: **A**: Synapse loss for the AI network. The results stem from mapping the network onto the BrainScaleS system. **B**: Results for the iterative compensation described in section 2.3.2. **C**: Power spectrum of the PY neurons within the network. The blue curve is calculated for the reference NEST simulation, the red represents the distorted ESS simulation. The green represents the distorted and compensated simulation using the ESS. The data were smoothed using a Gaussian filter. Used with permission from (Petrovici et al., 2014, Figure 23).

BrainScaleS hardware, the ESS, was used to test the compensation of all distortions at once.

It includes a realistic mechanism for synapse loss derived from the real constraints imposed by hardware implementation and adds a realistic simulation of the on- and off-wafer communication. The successful evaluation in this final step not only validates the presented compensation methods but also acts as a test of the operating software of the system (fig. 2.9, fig. 2.17, table 2.2). In the case of many heterogeneous connections, as in the example of the synfire chain network, a pre-mapping is required before applying the compensation, because the synapse loss for individual projections is highly different (table 2.1). For the AI network, a pre-mapping (section 2.3.2) or several emulations of the target network (section 2.3.2) are required to compensate for the distortion. As an advantage, the iterative compensation method (section 2.3.2) can counteract synapse loss and fixed-pattern inhomogeneities at once.

The presented compensation methods are specific to the network models that were selected for this study and can not be applied in general for arbitrary networks emulated on the device. However, the strategies presented above can be applied and adapted to by neuromorphic modelers according to their particular goals.

Chapter 3

Simulation-based characterization of neuron circuits

3.1 Introduction

This chapter covers the simulation-based characterization and verification of neuron circuits that are developed for the next generation of the BrainScaleS neuromorphic system. The goals of a detailed, simulation-based transistor-level characterization are twofold: First, to test the technical functionality of the neuron circuitry and provide an additional testing layer on top of the component-level verification performed by the individual circuit designers. The second goal is to assess the viability of the implemented neuron circuits for the investigation of biologically inspired spiking networks. This is accomplished by reproducing multiple single- and multi-compartment neural modeling use cases. An integrated simulation approach is particularly required in the case of the DLS3 chip where the full neuron functionality emerges from the correct interaction of several analog and digital components.

The DLS3 chip follows the concept of its predecessor, HICANN, in the regard that it incorporates a highly tunable neuron model. Alongside this tunability the analog neuron implementation is subject to variability, implying that identically configured circuits behave differently, for example by having different membrane time constants or firing thresholds. The number of tunable parameters is used to address the variability by implementing calibration algorithms: Individual circuits on the chip are characterized and their response function inverted so that a potential user can specify the desired property – such as a membrane time constant – rather than the technical parameter that tunes this property. This kind of calibration procedure is part of the typical workflow for multiple neuromorphic devices (Brüderle et al., 2011; Pfeil et al., 2013; Schmidt, 2014; Koke, 2017), and is also the intended operation mode for the DLS3 chip that is described in this chapter. Thus, for a pre-production validation of the neuron circuits, it is unavoidable to test whether calibration can be performed. Even if the mode of operation of the chip is changed to parameter tuning that happens using a feedback loop externally (Schmitt et al., 2017) or by using the built-in plasticity mechanisms (Friedmann et al., 2016), the simulations yield the

expected range within which the neurons will be tunable, alongside the variations in the parameters, the parameter sensitivity and precise dynamic properties of the individual circuit components.

This chapter is organized as follows: First the required parameterization is outlined by giving an overview over biological parameter ranges and by evaluating a number of computational studies that use the LIF or AdEx neuron model and are representative of the type of models that should be supported by the produced device (section 3.1.1). Second, the DLS3 prototype chip and the transistor-level neuron simulation setup are described (section 3.1.2), followed by a detailed description of individual neuron components (section 3.2). In section 3.3, the calibration methods for the individual terms as well as their evaluation in independent simulations is presented. The pre-production verification that was not an implicit part of the calibration development is described in section 3.4. Single- and multi-compartment test cases are analyzed in section 3.5 and section 3.6. Finally, initial chip measurements are shown in section 3.7.

Contribution

The DLS3 chip was developed by multiple designers. The contribution by different people to the content described in this chapter is as follows: The leak and reset circuit as well as the inter-compartment functionality and the synapse circuits were envisioned and designed by Johannes Schemmel. The remaining analog neuron circuits were implemented by Syed Ahmed Aamir. The digital neuron back end was implemented by Gerd Kiene. The Python-based interface to the simulation (section 3.1.3) was created by Sebastian Billaudelle during an internship that was co-supervised by the author of this thesis. Initial versions of the calibration methods for the synaptic time constant and the synaptic weight were implemented by Sebastian Billaudelle. Some simulations (denoted at the point of occurrence) were performed by Laura Kriener during the course of a Master's Thesis and student assistant work which were co-supervised by the author of this thesis.

3.1.1 Parameterization for biologically-inspired modeling

As stated in Millner et al. (2010) and Millner (2012), the goal for the implementation of the physical neuron model is a versatile substrate for the investigation of brain-inspired computing. The underlying mathematical description is the AdEx neuron model which was selected in Millner (2012) because it is flexible despite the comparative simplicity of a two-dimensional point neuron model. In particular, it can be used to model different neuron types by a change of parameterization only – an essential characteristic for a configurable analog neuron model which can only be modified by production of a new device. The main advantages of low-dimensional point neuron models are the simpler analytic approach to the description of their behavior and the possibility to more efficiently realize them on conventional computers as compared to detailed models, increasing the size of networks that can be simulated (Izhikevich, 2004).

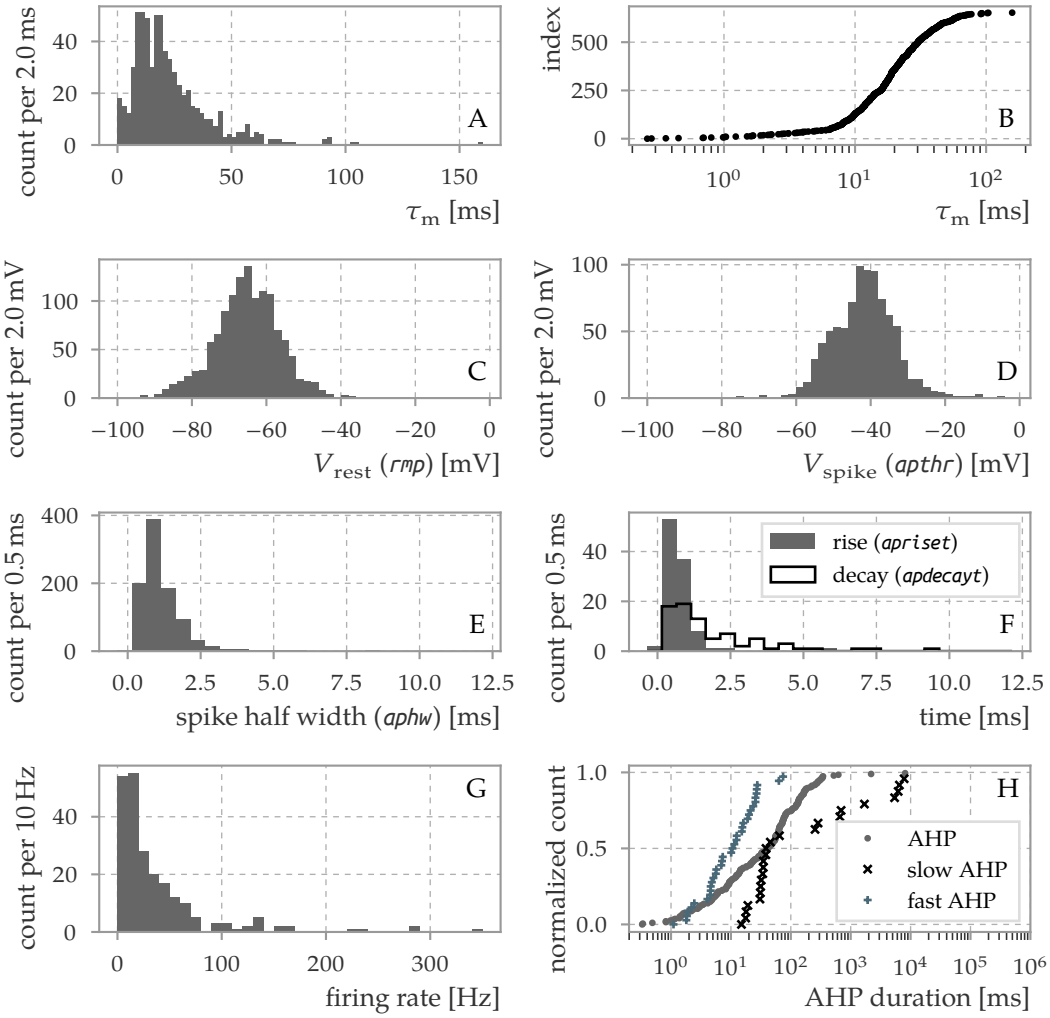


Figure 3.1: Distribution of membrane time constants in the NeuroElectro database (Tripathy et al., 2015; nel, 2017) **A**: Histogram of the *tau* metric in the database for all 654 non-empty entries. **B**: Cumulative distribution of the data shown in A. **C**: Resting potential **D**: Spike threshold. **E**: Spike half width – duration of the action potential at the voltage halfway between the firing threshold and the peak of the action potential. **F**: Rise and decay times of the action potential, "usually calculated as 10 % to 90 % decay time" (nel, 2017). **G**: Neuron firing rates (*apfreq*), including maximal, spontaneous firing rate. **H**: Duration of afterhyperpolarization, classified as slow, (*sahpdur*), fast (*fahpdur*) and not explicitly classified (*ahpdur*). The definition of the quantities may differ for different sources.

For analog implementations of neuron models an additional consideration is of relevance, that of parameterization. The configurable parameters in the AdEx neuron model are implemented using digital control bits and current or voltage signals from the analog parameter storage. The resolution and range of both configuration options is limited, for digital configuration by the discrete nature of the parameter space and for analog parameters by the precision of the analog memory. The implicit transformation from analog parameter to the physical property further affects the parameter precision. For example, the synaptic time constant results from the conductance of a tunable resistor which is controlled by a bias current in a nonlinear way (section 3.3.1). Thus, it is necessary to consider the required parameterization to conform to the anticipated use cases.

The desired parameter ranges should conform to ranges observed in the biological archetype of the neuromorphic system. Figure 3.1 A shows the distribution of membrane time constants collected by the *neuroelectro* project (nel, 2017) from articles published in several neuroscience-specific journals. The data in the figure stems from multiple species, primarily rats (309) and mice (274), and from various neuron types. Because the distribution is determined by the number of publications on certain species and neuron types as well as the ease of data extraction, it is not necessarily unbiased. However, it serves as a quantitative guideline for the required parameter range.

To supplement and extend the work in Millner (2012), sets of parameters were collected from modeling studies and publications on individual biological measurements. The studies are aimed at either explaining biological observations or at investigating brain-inspired computing paradigms with networks of spiking neurons. They include the studies that chapter 2 is based upon. Additionally, studies which compare different neuron models (Pospischil et al., 2011), and studies which utilize the AdEx neuron model for analytical and biological neuron modeling (Naud et al., 2008). Nessler et al. (2013) and Petrovici et al. (2013) are included as abstract modeling approaches which were previously considered for neuromorphic implementation (Breitwieser, 2015).

Tables 3.1 to 3.4 list the parameters used in the respective studies, including the requirements outlined in Millner (2012). The membrane time constants used in those studies are well within range of the data shown in Figure 3.1. The utilized synaptic time constants for AMPA and $GABA_A$ are on the order of 1 ms to 10 ms, while NMDA and $GABA_B$ require significantly longer time constants on the order of 100 ms. Typically used refractory periods range from 0 ms to 5 ms. The value of zero is used by Brette and Gerstner (2005) is presumably due to the fact that the integration time of the exponential rise during an action potential in the AdEx model takes the place of an explicit refractory period. Petrovici et al. (2013) use a comparatively long refractory period of 10 ms which is driven by the requirement of τ_{ref} to be approximately equal to τ_{syn}. Figure 3.1 E shows the distribution of spike duration defined from the spike half-width, the width of the action potential measured at the center voltage between spiking threshold and spike peak. Panel D shows the rise- and fall times of the action potential. The values of 0 ms to 2.5 ms

for the spike half width, 0 ms to 2 ms for the rise- and 0 ms to 10 ms for the fall-time provide a lower limit for the absolute refractory period of the abstract LIF neuron. Typical firing rates (fig. 3.1 G) impose a soft upper bound on the duration of the refractory period: for firing rates above 300 Hz, the τ_{ref} must be below 3.3 ms. The firing rate metric *apfreq* in (nel, 2017) is not necessarily the maximum firing rate but rather includes any firing rate that is referred to in the respective publication – so the metric is not indicative of the required refractory period for the LIF or AdEx models.

Values for V_{leak} in the selected studies cover a range of –100 mV to –56 mV; data from biological observations covers a wider range of approximately –100 mV to –40 mV (fig. 3.1 C). Similarly, the firing threshold V_{thresh} is set from –57 mV to –50 mV while fig. 3.1 D shows a distribution between –70 mV to –10 mV. The location of the leak and threshold potentials is, to a large extent, not critical for the hardware implementation, because of the scaling and shift of all voltages in the translation from biological to hardware domain (section 1.7), so biological models with different voltage parameters can all be mapped to the same optimal hardware range.

The reversal potentials for excitatory inputs (AMPA and NMDA receptors) is typically set to 0 mV while the inhibitory reversal potential lies within –90 mV to –70 mV (table 3.2). Note that the DLS3 chip implements current-based synaptic inputs. The presented studies utilize conductance-based synapses, to provide a basis for future conductance-based synaptic input circuits (as in HICANN and Spikey chips) The order of magnitude for equivalent current-based synaptic weights can be estimated from the distance of the membrane potential to the corresponding synaptic reversal potentials and the conductance-based weights (table 3.3).

The values used for the synaptic weights cover a wide range. One major reason is that modeling studies frequently use a reduced number of neurons to improve the required simulation time. This leads to higher synaptic weights to account for the reduced number of synapses per neuron. Often, the synaptic weights are used as free parameters to adjust the desired "ground state" of the network. For example, in Vogels and Abbott (2005); Destexhe (2009) where the $(w_{\text{exc}}; w_{\text{inh}})$ parameter space is scanned to adjust an irregular self-sustained activity. The collected parameters from the modeling studies are summarized in table 3.5.

Table 3.1: Exemplary neuron parameters from modeling studies and physiological measurements. (part 1)

variable unit (biological domain)	Comment	C_m nF	τ_m ms	τ_{syn} ms	τ_{ref} ms
HICANN v2 requirements					
Millner (2012, Sec. 3.12)		0.281	6.6 to 100	2.5 to 10.5	0.5 to 5 [1]
Vogels and Abbott (2005)		0.05	20	5 to 10	5
Deco and Jirsa (2012)		0.2 to 0.5	10 to 20	2 to 100 [2]	1 to 2
Masquelier and Deco (2013)	[3]	0.5	20	2 to 100	2
Brunel and Wang (2001)		0.2 to 0.5	10 to 20	2; 10; 100	1 to 2
Naud et al. (2008)		0.059 to 0.2	7 to 50	–	0
Pospischil et al. (2011)	[4]	0.33 to 0.412	13 to 35	–	–
Destexhe (2009)		0.2	20	5; 10	2.5
Brette and Gerstner (2005)		0.281	9.4	2.7; 10.5	0
Petrovici et al. (2014) Synfire		0.29	10	2 to 10	2
Petrovici et al. (2014) L23 Pyr		0.18	16.9	6 to 16	0.16
Petrovici et al. (2014) L23 RSNP, BAS		0.007	15 to 16	6 to 66	0.16
Petrovici et al. (2014) AI		0.25	15	5	5
Petrovici et al. (2013)		0.1	20	10	10
Nessler et al. (2013)	[5]	–	15	1	–
Nakanishi and Kukita (1998)	[6]	–	–	–	–
Destexhe et al. (1994)		–	–	5; 150; 5; 210 [7]	–

[1] Millner (2012, Table 3.7) for an acceleration factor of 10^4
[2] AMPA: 2 ms, GABA: 10 ms, NMDA decay: 100 ms
[3] Parameters from Brunel and Wang (2001)
[4] estimates from in-vitro experiments
[5] Synaptic and membrane time constant not identified (which of the two is smaller) in the PSP kernel. Stochastic neuron model.
[6] Neocortical neurons in culture from cerebral cortex of embryonic Wistar rats. EPSP measurements, measure total and synaptic delay for spikes and bursts
[7] Destexhe et al. (1994, Table 1, Table 3), fall times (r_2) in the two-state scheme for AMPA/kainate, NMDA, $GABA_A$, $GABA_B$

Table 3.2: Exemplary neuron parameters from modeling studies and physiological measurements. (part 2)

variable unit (biological domain)	V_{leak} mV	V_{thresh} mV	V_{reset} mV	$E_{\text{rev,E}}$ mV	$E_{\text{rev,I}}$ mV
HICANN v2 requirements					
Millner (2012, Sec. 3.12)	– [8]	–	–	–	–
Vogels and Abbott (2005)	−60	−50	−60	0	−80
Deco and Jirsa (2012)	−70	−50	−55	0	−70
Masquelier and Deco (2013)	−70	−50	−55	0	−70
Brunel and Wang (2001)	−70	−50	−55	0	−70
Naud et al. (2008)	−70 to −58	0	−58 to −46	–	–
Pospischil et al. (2011)	−100 to −80	–	–	–	–
Destexhe (2009)	−60	$V_T = -50$	−60	0	−80
Brette and Gerstner (2005)	−70.6	$V_T = -50.4$	−70.6	0	−75
Petrovici et al. (2014) Synfire	−70	−57	−70	0	−75
Petrovici et al. (2014) L23 Pyr	−61.7	−53	−60.7	0	−80
Petrovici et al. (2014) L23 RSNP, BAS	−57.5 to −56	−52.5 to −51	−72.5	0	–
Petrovici et al. (2014) AI	−70	−40; $V_T = -50$	−70	0	−80
Petrovici et al. (2013)	−65	−52	−53	0	−90
Nessler et al. (2013)	–	–	–	–	–
Nakanishi and Kukita (1998)	-61.4 ± 0.7 mV	–	–	–	–
Destexhe et al. (1994)	–	–	–	–	–

[8] Voltages can be configured in the full technical dynamic range (Millner, 2012, Table 3.7)

Table 3.3: Exemplary neuron parameters from modeling studies and physiological measurements. (part 3)

variable unit (biological domain)	w_{exc} nS	w_{inh} nS	delay ms	current input
HICANN v2 requirements				
Millner (2012, Sec. 3.12)	– [9]	–	–	–
Vogels and Abbott (2005)	0 to 10	0 to 100	0 to 20	0 pA to 500 pA
Deco and Jirsa (2012)	0.1 to 4.4	0.08 to 3.4	–	–
Masquelier and Deco (2013)	0.1 to 1.2	–	3	–
Brunel and Wang (2001)	0.08 to 2.1	1	0.5	–
Naud et al. (2008)	–	–	–	–
Pospischil et al. (2011)	1 to 36	–	–	400 pA to 900 pA
Destexhe (2009)	0 to 10	0 to 100	0	–
Brette and Gerstner (2005)	$g_{\mathrm{total}} : g_{\mathrm{leak}} \leq 5 : 1$ [10]	–	–	–1 nA to 2.5 nA [11]
Petrovici et al. (2014) Synfire	1 to 3.5	2	4 to 20	–
Petrovici et al. (2014) L23 Pyr	0.2 to 4	3 to 6	0.5 to 8	–
Petrovici et al. (2014) L23 RSNP, BAS	0.025 to 0.1	–	–	–
Petrovici et al. (2014) AI	3 to 11	50 to 130	0.3 to 3	–
Petrovici et al. (2013)	3.5 [12]	–	–	–
Nessler et al. (2013)	–	–	–	–
Nakanishi and Kukita (1998)	–	–	0 to 15 [13]	–
Destexhe et al. (1994)	–	–	–	–

[9] Not included in neuron parameter list.

[10] Brette and Gerstner (2005) use a maximal ratio of total conductance to leak conductance of 5:1, referring to Pare et al. (1998) who use cat neocortical pyramidal neurons in vivo.

[11] Brette and Gerstner (2005, Fig. 1 and 2)

[12] Petrovici et al. (2013, Appendix VII)

[13] 0 ms to 8 ms for response latency, cf. Figs. 4, 5, conduction velocity assumed 0.3 m s^{-1}, dendritic delay assumed 0.3 ms

Table 3.4: Exemplary neuron parameters from modeling studies and physiological measurements. (part 4)

variable	a	b	τ_w	Δ_T
unit (biological domain)	nS	pA	ms	mV
HICANN v2 requirements				
Millner (2012, Sec. 3.12)	4.7 to 56 [14]	25 to 250 [15]	100 to 600	2 [16]
Vogels and Abbott (2005)			–	–
Deco and Jirsa (2012)			–	–
Masquelier and Deco (2013)			–	–
Brunel and Wang (2001)			–	–
Naud et al. (2008)	−11 to 4	0 to 120	16 to 300	0.8 to 5.5
Pospischil et al. (2011)	–	–	–	–
Destexhe (2009)	1 to 40	0 to 80	600	2.5
Brette and Gerstner (2005)	4	80.5	144	1 to 3
Petrovici et al. (2014) Synfire	–	–	–	–
Petrovici et al. (2014) L23 Pyr	0	13.2	196	0
Petrovici et al. (2014) L23 RSNP, BAS	0 to 0.28	0 to 1	250	0
Petrovici et al. (2014) AI	1	0 to 5	600	2.5
Petrovici et al. (2013)	–	–	–	–
Nessler et al. (2013)	–	–	–	–
Nakanishi and Kukita (1998)	–	–	–	–
Destexhe et al. (1994)	–	–	–	–

[14] For $C_m = 0.281\,\text{nF}$ with a value for τ_a of 5 ms to 59 ms

[15] "one order of magnitude around the model value" of 80.5 pA, (Millner, 2012, 3.12.1)

[16] A margin is included leading to hardware values of 2 mV to 15 mV.

Table 3.5: Summary of Tables 3.1 to 3.4. The values include the maximal and minimal ranges for the parameters for the surveyed studies as well as the reported range for the precursor chip, HICANN. For Δ_T, values of 0 are also used, but this case is equivalent to a shifted V_{thresh}.

variable	min.	max.	unit
τ_{m}	7 [17]	50	ms
τ_{syn}	1	100	ms
τ_{ref}	0	10	ms
V_{leak}	−100	−56	mV
V_{thresh}	−57	−40	mV
V_{reset}	−72.5	−46	mV
$E_{\text{rev,E}}$	0	0	mV
$E_{\text{rev,I}}$	−90	−70	mV
a	−11	56	nS
b	0	250	nA
τ_{w}	16	600	ms
Δ_{T}	0.8	5.5	mV
delay	0	20	ms

3.1.2 Neuromorphic prototype chip

The HICANN DLS3 chip is a prototype chip for the next-generation accelerated neuromorphic hardware that is produced in a 65 nm process technology. The main changes as compared to the previous version are:

1. The LIF model in DLS2 (Aamir et al., 2016, 2017b) is extended to the adaptive exponential integrate-and fire model (Aamir et al., 2017a) (section 3.3).

2. Inter-compartment connectivity is added, allowing to form larger logical neurons (as in the HICANN system, Schemmel et al. (2008)).

3. The possibility to connect neurons by inter-compartment resistances is added (Schemmel et al., 2017) (section 3.6).

4. The reset mechanism is changed from an analog to a digital refractory period (Kiene, 2017) (section 3.2.1) and to reset using a configurable conductance (section 3.2.3).

5. A fast, on-chip analog-to-digital converter (ADC) for readout of neuron voltages is added (Hartel et al., 2017, sec. 12.3.5).

6. The correlation ADC is redesigned (Hartel et al., 2017, section 12.3.3).

[17] Since the DLS3 chip implements current-based synapses, lower membrane time constants are required to accommodate shorter effective time constants in the high conductance mode (Pare et al. (1998), Petrovici et al. (2015)).

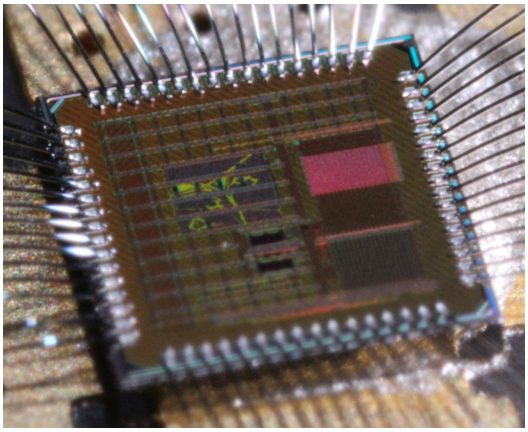


Figure 3.2: Photograph of the DLS3 prototype chip, taken by Ralf Achenbach. The full-custom structures are visible on the center and bottom right.

7. The possibility to use the correlation ADC to read out neuron voltages is added (fig. 3.51).

8. Synapse drivers with short-term plasticity are implemented (Billaudelle, 2017).

As in the previous prototype, the chip contains 32 neuron circuits with 32×32 synapses with a local timing correlation measurement and a plasticity processing unit (Friedmann (2013), Friedmann et al. (2016)).

3.1.3 Simulation setup

The aim of the simulations that are presented in this chapter is the verification of the usability of the neuron circuits for modeling applications. The simulation of the neuron design is more complex when compared to previous 180 nm and 65 nm chip versions. The full neuron behavior is reliant on the correct interaction of three separate design blocks, built by three designers: The leak and reset circuitry as well as the multi-compartment components are implemented by Johannes Schemmel. The remaining analog neuron circuits, including the added AdEx features, synaptic input, threshold comparator, are located in the main neuron block, implemented by Syed Ahmed Aamir (Aamir et al., 2017a). The refractory time and adaptation signals are generated in the digital back end, implemented by Gerd Kiene (Kiene, 2017). The synapses, which are implemented by Johannes Schemmel, generate the signals that are converted in the neuron circuit to post-synaptic currents. All analog circuits receive precisely controlled currents and voltages which are generated in the capacitive memory for analog parameters; this component was implemented by Matthias Hock (Hock, 2014).

Integrated scope of simulation

While each individual component is thoroughly simulated by the respective designer, the correct interaction between these components must be verified as well to prevent

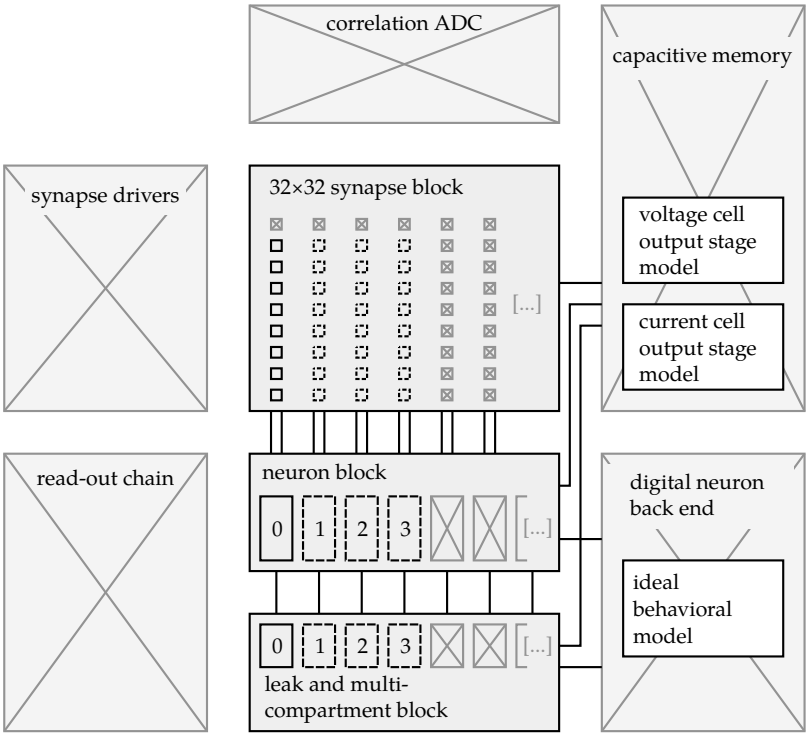


Figure 3.3: ANNCORE-based simulation setup. Components with a dashed outline can be included optionally. Crossed-out components are replaced by non-functional dummies. The component blocks for the capacitive memory and digital neuron back end have the same interface but contain behavioral models

incompatible behavior at the component boundaries. One approach is to create a simulation environment which contains the component under test and as much of the peripheral circuitry as needed to test certain functionality. This is the approach that was taken by the author in collaboration with Sebastian Billaudelle for the verification of the DLS2 neuron.

For DLS3, a different approach is chosen. The top-level ANNCORE schematic is kept unchanged, as it is used for production. Because simulating the full schematic is not feasible in reasonable time it is simplified: Configuration views[18] are used to replace individual blocks by custom components, as shown in fig. 3.3. This has the advantage that the top level of the ANNCORE design is identical to the one used for the layout versus schematic (LVS) verification, which reduces the chance of a mismatch between tested and produced connectivity.

For each neuron, only four inhibitory and four excitatory synapses are instantiated instead of the full row of 32. The neuron block contains one or four neuron circuits, with the rest replaced by ideal leakage of 1 µS to 600 mV. (This is only relevant if the rightmost non-ideal neuron is connected to the right using one of the inter-compartment connections.) The leak-, inter-compartment and synapse circuits

[18]Configuration views are a feature of the simulation back end that allow a dynamic replacement of a cell in the simulation.

are also simulated selectively, only using one or four columns, with the rest of the instances being replaced by dummies because they are not relevant to the behavior of the analog neuron.

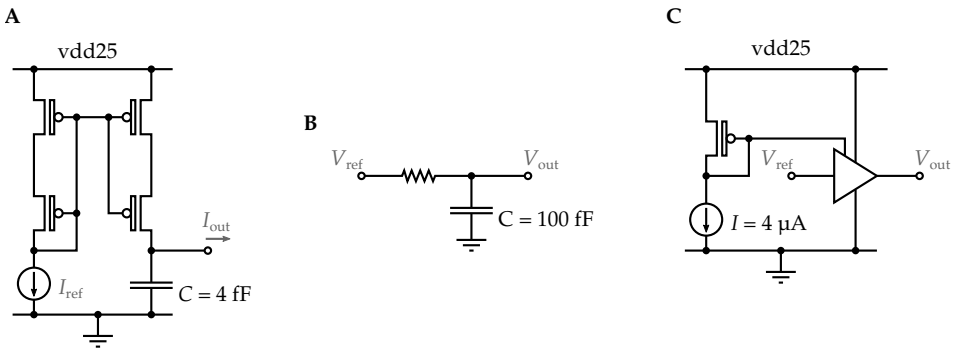


Figure 3.4: Output stage model for capacitive memory, as proposed by Matthias Hock. **A**: Current cell. **B**: Unbuffered voltage cell. **C**: Buffered voltage cell.

The digital neuron back end is replaced by an ideal behavioral model (section 3.2.1). The parameter storage cells of the capacitive memory that are used by different circuits are replaced by the output stage of the corresponding cell, provided by Matthias Hock, as shown in fig. 3.4: Current cells contain the output current mirror of the cell and a capacitor that models the parasitic capacitance of the connecting line. Unbuffered voltage cells consist only of an ideal resistance and a capacitance. The capacitance replaces the capacitor in the analog memory and the resistance serves to emulate the low-frequency periodic update that the capacitor receives during operation. The buffered voltage cells use a transistor-level model of the buffer inside the cell. (All voltage parameters that are adjustable per neuron are unbuffered. That includes all voltage parameters for the simulations shown in this chapter.)

In the actual implementation (Hock, 2014), the analog value can be adjusted using a digital control setting with a 10-bit resolution. In the simulation, the reference currents and voltages (I_{ref} and V_{ref} in fig. 3.4) are provided with floating point precision. An analysis of the dependence on the parameter precision can be carried out by varying the reference currents according to the expected resolution (section 3.6.3). All other components in the ANNCORE are replaced by empty dummies.

Software interface

The simulation is interfaced using a python library, *hdsal*, which encapsulates the configuration space of the design and simulation properties such as process corner or the Monte-Carlo sample. For each simulation, the necessary simulation parameters and current and voltage signals are calculated and passed to the simulation back end using the *teststand* library (fig. 3.6). The simulation itself is performed using the spectre simulator[19]. A short example of the usage of the library is shown in listing 1.

[19]Cadence Design Systems, Inc., San Jose, CA, USA, Version 12.1.1.096.isr12 64bit – 5 Aug 2013.

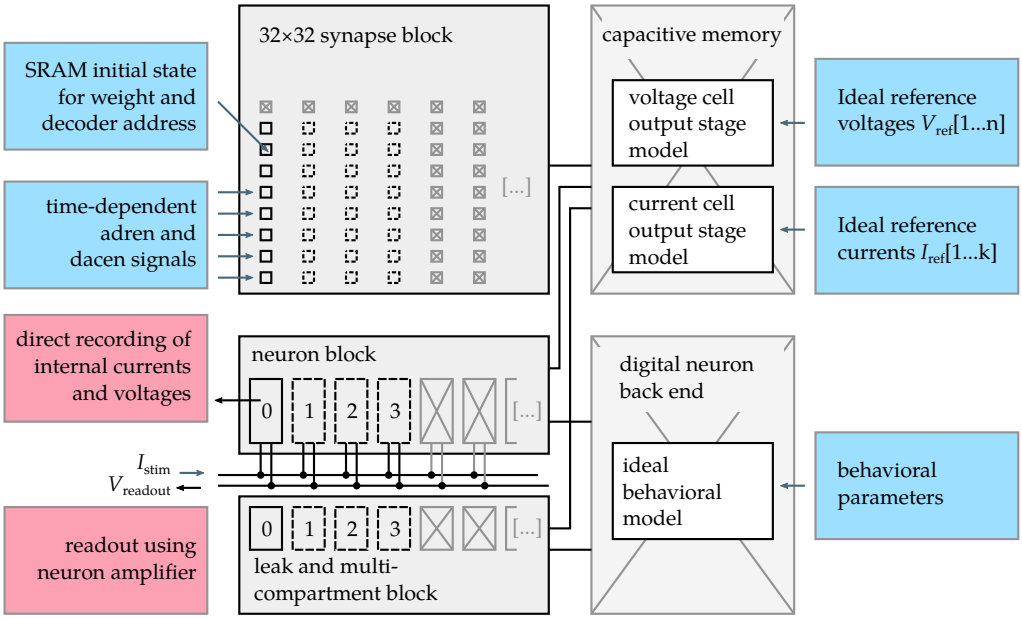


Figure 3.5: Input (blue) and output (red) signals for the ANNCORE-based simulation.

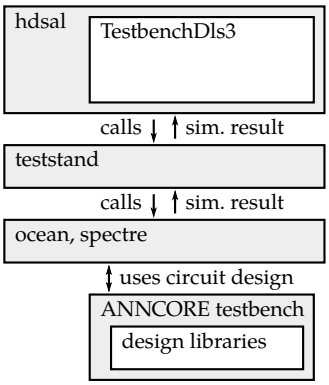


Figure 3.6: Software interface to ANNCORE-based simulation.

The *teststand* and *hdsal* libraries were developed by Sebastian Billaudelle during his internship and student assistant work; large parts of the ANNCORE interface were extended and re-written by the author.

Notation

The DLS3 chip is produced in a standard 65 nm CMOS process. The transistor symbols used in the schematic diagrams in this chapter are shown in fig. 3.7. The two transistor types that are used in the design are the standard, thin-oxide transistors which are operated at 1.2 V supply voltage, and thick-oxide transistors for 2.5 V operation. Following Hock (2014), individual terminals are not labeled. For n-channel MOSFET (NMOS) devices, the source contact is located at the lower and for p-channel MOSFET (PMOS) at the higher potential. Where necessary, the individual

```
from hdsal.testbench_dls3 import TestbenchDls3

# Simulate fs corner with one neuron circuit in the simulation
nrn = ('fs', 0, 1)

tb = TestbenchDls3(nrn)

# adjust parameters for neuron compartment 0
tb.parameters[0].v_leak = 0.5
tb.parameters[0].control_mask.en_exp = True
tb.parameters[0].control_mask.en_syn_i_exc = True
tb.parameters[0].synapse_config.weight_exc = [[10, 10, 10, 10]]

# synaptic events are sent via the first excitatory synapse
tb.parameters[0].synapse_config.spikes_exc = [[[10e-6, 20e-6], [], [], []]]

# simulate for 30 µs
result = tb.run(30e-6)

# process results
import matplotlib.pyplot as plt
plt.plot(result['time'], result['Ianc.mem\\<0\\>'])
```

Listing 1: Example of use for the *hdsal* library. A neuron is simulated in the *fs* corner with spike input using the first synapse and input spikes at times 10 µs and 20 µs. The tuple *('fs', 0, 1)* indicates the sample (corner or index in a Monte-Carlo simulation), the index of the primary neuron in the simulation (0) and the number of neuron circuits in simulation (1).

terminals are denoted explicitly. When the bulk contact is not shown it is connected to ground for NMOS devices and to the respective supply voltage for PMOS. Thick-oxide transmission gates are drawn using four triangles, otherwise two triangles are used. Parameter and signal names, such as *i_bias_leak* are printed in italic if they are derived from the user-adjustable parameters stored in the capacitive memory (section 3.1.3).

Signals in the schematics which are designated by the same name are implicitly connected. To improve readability, this kind of connection is usually denoted by a dashed line – see, e.g., “iSynExc” in fig. 3.8.

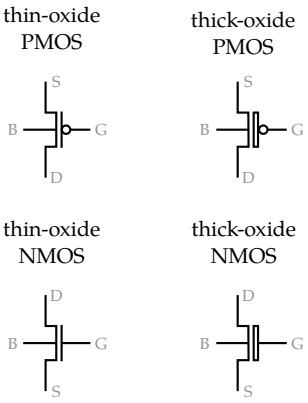


Figure 3.7: Symbols used for metal–oxide–semiconductor field-effect transistor (MOSFET) devices in this chapter.

Mismatch and process variations

The simulations within this chapter rely on models of transistor variation, which are provided by the chip manufacturer. The two main types of variability are transistor mismatch and process variation. Process variation is the effect of varying transistor behavior between chips produced on different wafers. This effect is modeled by using *process corner* simulations: All transistor parameters within the simulated circuit are modified identically to represent the extreme values that are expected during production. The typical case is denoted *tt* (for *typical-typical*). Models for slow and fast corners, representing variations in carrier mobilities and threshold voltages in NMOS and PMOS devices, are provided. They are denoted by a corresponding two-letter combination, *fs*, for example, referring to the case of fast NMOS and slow PMOS. Transistor mismatch denotes the variability of identically parameterized devices on a single chip due to production variations in device dimensions, number of dopant atoms under a gate etc. (Kinget, 2005). These effects typically increase in severity with reduced device size, which requires a trade-off during development between the level of variation and required area. Mismatch effects are the primary cause for the required circuit calibration on neuromorphic devices (chapter 3, section 3.3).

In the analysis within this chapter, Monte-Carlo simulations are used to estimate the severity of the effect that mismatch has on the neuron functionality. They rely on a statistical model of the expected variations within the individual devices that is provided by the chip manufacturer. The matching between the Monte Carlo models and test transistors on a previous prototype chip was compared in (Hock, 2014, section 5.3.1). The simulations overestimated the measured variation by at most 70 % for low reference currents and by 10 % for high reference currents.

3.2 The DLS3 neuron

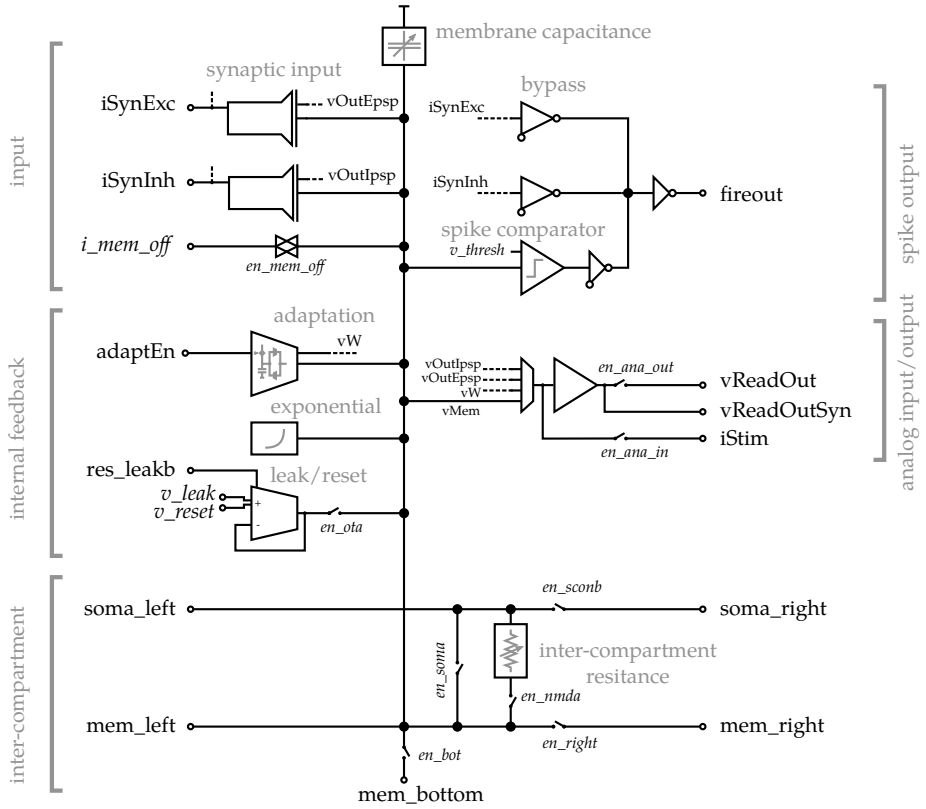


Figure 3.8: Top-level schematic of the DLS3 neuron. Only essential signals and configuration parameters are included. See the representation of the individual components for an exhaustive view of the configuration space. Most component symbols are adopted from Aamir et al. (2016), Aamir et al. (2017a) and the schematic design by Syed Ahmed Aamir for consistency reasons.

The neuron circuit in the DLS3 prototype chip is an analog, time- and voltage-scaled implementation of the AdEx neuron model. In this section, the implementation of the components of the neuron circuit is described in detail.

Figure 3.8 shows the top-level view of the neuron circuit. A digitally configurable capacitor serves as the membrane capacitance of the accelerated neuron. The main input source of the neuron comes from synapse array (fig. 3.9). Each synapse listens to incoming events. If the event address matches the address storage of the synapse, the synapse discharges one of the synaptic input lines (iSynExc and iSynInh). The amount of charge is determined by the output digital-to-analog converter (DAC) of the synapse which is controlled by a 6-bit synaptic weight (fig. 3.9). The second kind of stimulus for the neuron is the current cell *i_mem_off* which is connected directly to the membrane separated only by a thick-oxide transmission gate. External, i.e., off-chip current input can be achieved using the analog input/output module via the stimulus pin "iStim". The feedback terms of the AdEx equation are implemented

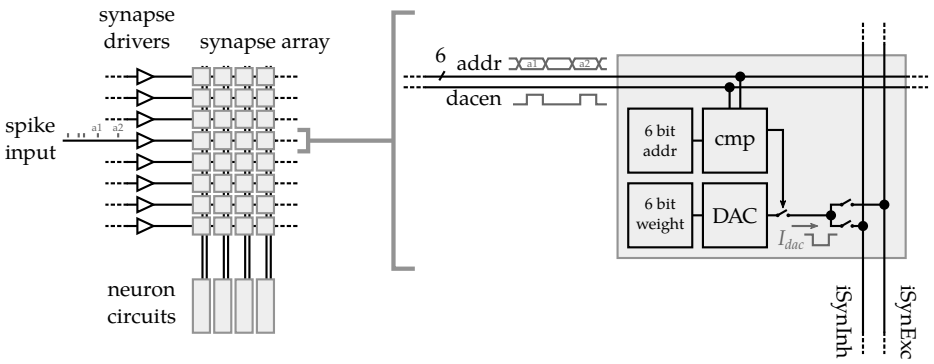


Figure 3.9: On-chip spike communication infrastructure.

as individual circuits for leak, adaptation and exponential current. In contrast to previous implementations, the membrane reset after an emitted spike is realized by the leak circuit as well (section 3.2.3).

The spike output is initiated by the spike comparator, which emits a digital signal for as long as the membrane is above threshold. The output signal "fireout" is then passed to the digital part of the reset mechanism. For debugging purposes, analog bypass modules are included that monitor the synaptic input lines for events and produce digital output spikes in place of the normal operation of the spike comparator. Their primary use case is to test whether digital events arrive at the neuron via the synapse array without relying on the analog behavior of the neuron circuit and without requiring to read out analog signals.

The input and output module consists of an amplifier and a multiplexer. The multiplexer selects one of four internal neuron signals – the membrane voltage, the voltage on the adaptation capacitor and one of the two buffered voltages of the synaptic input lines. The output side of the multiplexer can be connected to the "iStim" line, providing direct access to the neuron from outside of the chip, for example for a direct current measurement of individual components (fig. 3.13, fig. 3.14). The amplifier of the input/output module provides a buffered access to the internal signals. The output of the amplifier is switched to arbitrate which neuron drives one of the two shared "vReadOut" lines on the chip. Additionally, a non-switched amplifier output is routed to the correlation ADC to allow parallel access to all neurons using this read-out path. (Hartel et al., 2017, section 12.3.3).

The bottom of fig. 3.8 shows the inter-compartment connectivity of the neuron implementation. As in the 180 nm system, a connection to directly neighboring neuron compartments can be enabled using the *en_right* and *en_bot* switches. The main purpose of this direct connection is the creation of a larger, logical neuron with an increased number of attached synapses and more than two synaptic inputs. (The DLS3 prototype chip only has one row of neurons and the vertical connection is in place for the embedding of the circuit in a full chip, akin to fig. 2.1.) A second line can be connected to from each neuron compartment using a transmission gate or a configurable conductance. This line can be interrupted at given intervals – between each compartment in the DLS3 prototype chip – using the *en_sconb* configuration

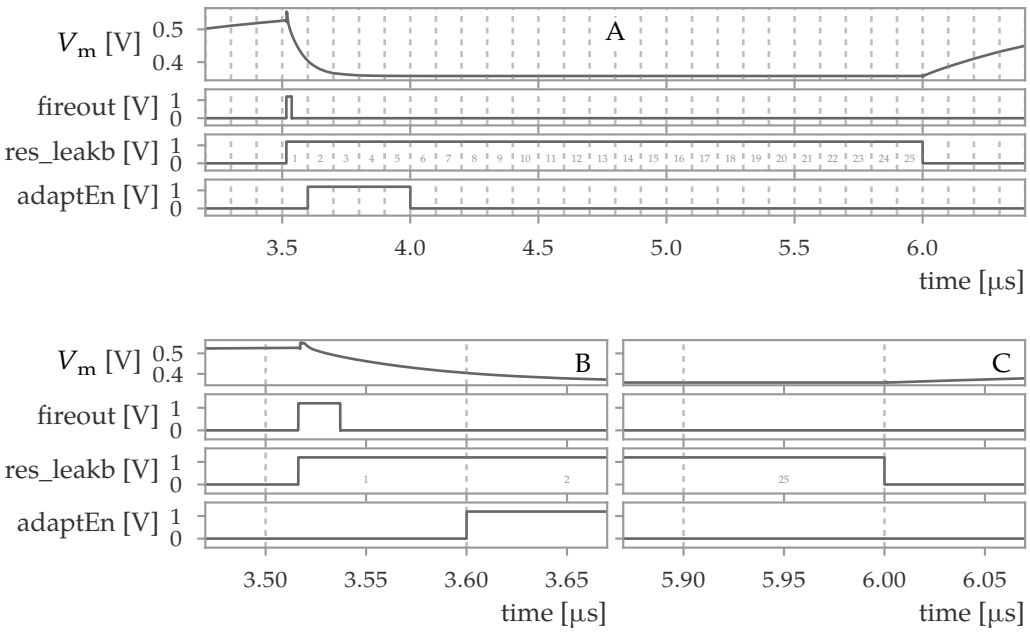


Figure 3.10: Timing of the reset and adaptation signals in the digital neuron back end. The figure shows the input and output of the digital back end of the neuron. **A**: Simulation result for a single reset. The membrane voltage (top) crosses the firing threshold of 0.53 V which is detected by the spike comparator, and a reset follows. The "fireout" signal, as emitted by the neuron, and the "res_leakb" and "adaptEn" signals, which are produced by the behavioral model of the digital back end. **B** and **C** show a magnification of the beginning and end of the reset period in A. The dashed vertical lines represent the clock edges at *refrac_clk_freq* = 10 MHz.

switch. This allows to connect neuron circuits to emulate multi-compartment models (section 3.6). This implementation is different from previous prototype chips (Millner et al., 2012) in that it does not highly configurable structures. Certain tree-like configuration are possible, as discussed in detail in section 3.6.

3.2.1 Interface to the digital back end

When a spike comparator in a neuron circuit detects that the membrane voltage is above the firing threshold, the neuron emits a signal on the "fireout" line (fig. 3.8), which is received by the digital neuron back end. The back end then generates reset and adaptation signals for the neuron. Their timing is shown in fig. 3.10.

The neuron membrane crosses the firing threshold, which is detected by the spike comparator: The "fireout" signal is active for as long as the membrane is above threshold. The digital neuron back end enables the reset signal immediately, i.e., not aligned to a clock (fig. 3.10B). This is necessary to prevent an overshoot of the membrane voltage which could affect the network activity in undesired ways, for example by stronger than intended influence of sub-threshold adaptation or

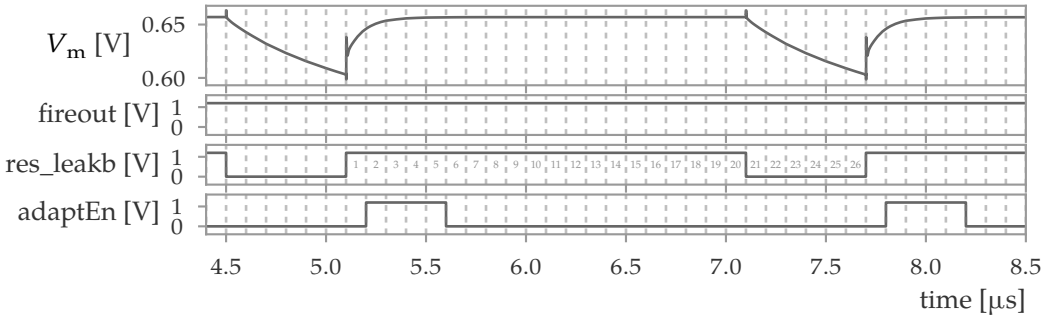


Figure 3.11: Hold-off functionality of digital reset. The same signals as in fig. 3.10 are shown. The reset potential is set above the firing threshold so the neuron continuously signals that it is spiking ("fireout" signal is constantly high). A *refrac_time* of 25 is configured with *holdoff_time* = 5 and *refrac_clk_freq* = 10 MHz.

by excitation of neighboring membrane compartments in a multi-compartment emulation setup. Even if those effects could be compensated for, the alternative to an asynchronous reset start – a reset start at the following clock edge – would introduce a jitter due to the relative alignment of threshold crossing and clock. This is particularly grave when a refractory clock frequency slower than 10 MHz is used. The end of the refractory period, on the other hand, is synchronous to the clock of the digital neuron back end (fig. 3.10C).

In contrast, the adaptation signal "adaptEn", which regulates the current source that implements spike-triggered adaptation (fig. 3.22), should have a well-defined duration. It determines the charge that flows onto the adaptation capacitance C_w for spike-triggered adaptation (section 3.2.4). Consequently, both start and end of the adaptation signal are synchronous with the clock (fig. 3.10B).

The durations of the refractory period and adaptation signal are configured by integer values *adaptation_time* and *refrac_time*. The time step is given by the clock of the digital neuron back end which can be selected per neuron from the values of 1 MHz and 10 MHz.

A third configuration value, the *holdoff_time*, is introduced to support the use of the reset mechanism to generate plateau potentials (section 3.6.2). Here, the membrane potential should be held at a high voltage level for a prolonged duration. For this it is necessary to set the reset potential above the firing threshold. If a new spike is admissible immediately after the end of a reset, that configuration would lead to a permanently firing neuron once it has crossed the threshold, because at the end of the reset the membrane is still at the reset voltage, and above threshold. To make the plateau potential feature controllable, the refractory period – the time in which the neuron can not emit a new spike – is split into a reset time, in which the membrane is pulled towards the reset potential, and a *holdoff* time, in which the leak conductance and potential are active. This can, for example, also be used to create a bistable neuron that is switched by synaptic input, as shown in fig. A.5.

3.2.2 Synaptic input

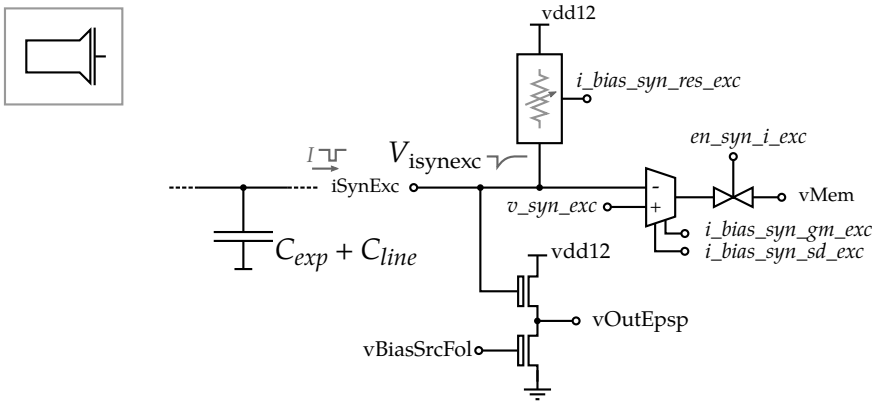


Figure 3.12: Schematic of DLS3 synaptic input. The excitatory input is shown; for the inhibitory synaptic input, the input terminals of the operational transconductance amplifier (OTA) are swapped. The symbol used in fig. 3.8 is shown in the top left.

The synaptic transmission in the DLS3 neuron model uses exponentially decaying post-synaptic currents. The implementation of the synaptic input, the circuit producing these currents from weighted synaptic events, is shown in fig. 3.12. It consists of an OTA, a tunable resistor and a source follower for read-out.

When synaptic events arrive, the synaptic input line ("iSynExc") is discharged by the synapse circuits (fig. 3.9). The difference between the voltage on the input line V_{isynexc} and the constant voltage *v_syn_exc* is converted to a current by the synaptic input OTA (fig. 3.9). The inhibitory synaptic input is constructed identically, but with swapped input terminals for the synaptic input OTA.

The time constant of the exponential decay is achieved by the tunable resistor circuit which discharges the input line voltage to 1.2 V. On the DLS3 prototype chip, the capacitance consists of the parasitic capacitance of the synaptic line in addition to two capacitors with nominal values of 481 fF and 440 fF. It is intended to replace as much as possible of this capacitance by the line capacitance itself in future versions of the chip – the number of synapse rows will increase from 32 to approximately 200, increasing the length of the connection.

The voltage on the synaptic input line can be read out externally by using the source follower circuit. Note that the bias for the source follower, "vBiasSrcFol", is generated in the read-out amplifier (fig. 3.13) using its bias current, which is derived from *i_ref_analog*. This is generally not a problem because the signal is read out in a buffered fashion which requires an enabled read-out amplifier in any case.

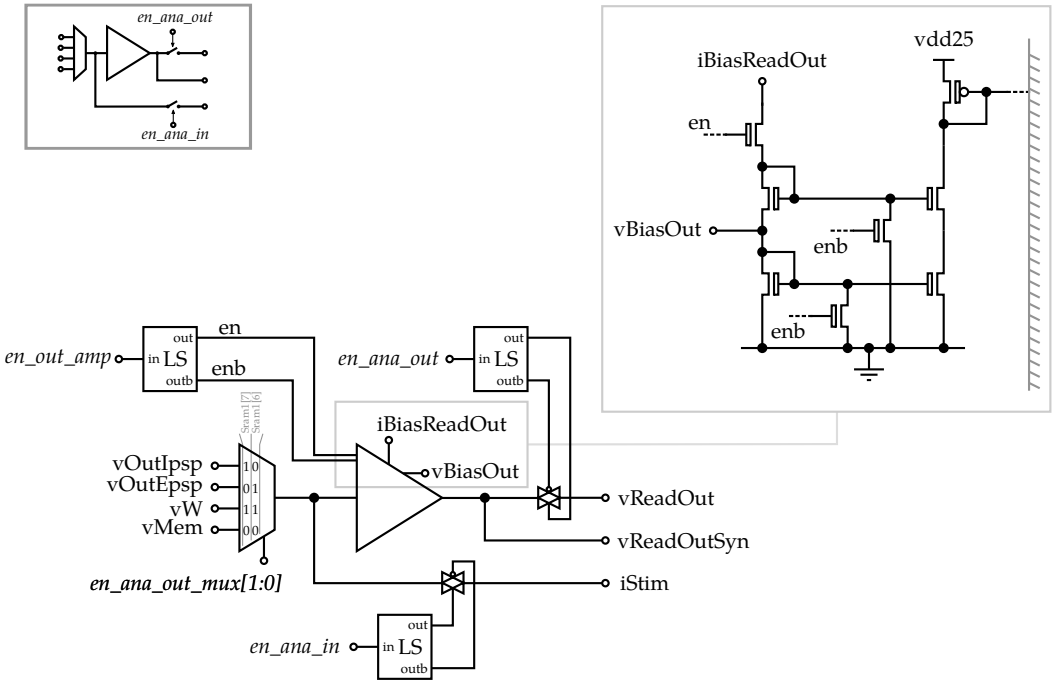


Figure 3.13: Schematic of the read-out amplifier configuration. Top left: symbol that is used in fig. 3.8 to represent the component. Top right: inset that shows the relevant circuit that generates the source follower bias ("vBiasOut") for the synaptic input (fig. 3.12).

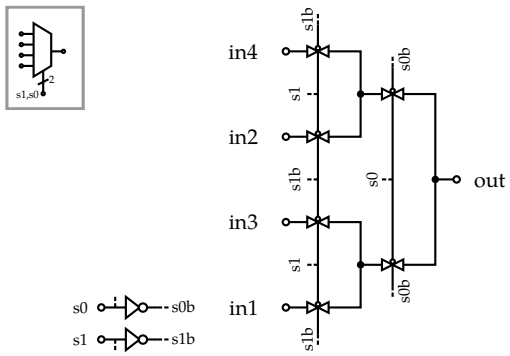


Figure 3.14: Schematic of the read-out multiplexer. The multiplexer is implemented as a binary tree from thin-oxide transmission gates.

3.2.3 Leakage and reset transconductance amplifier

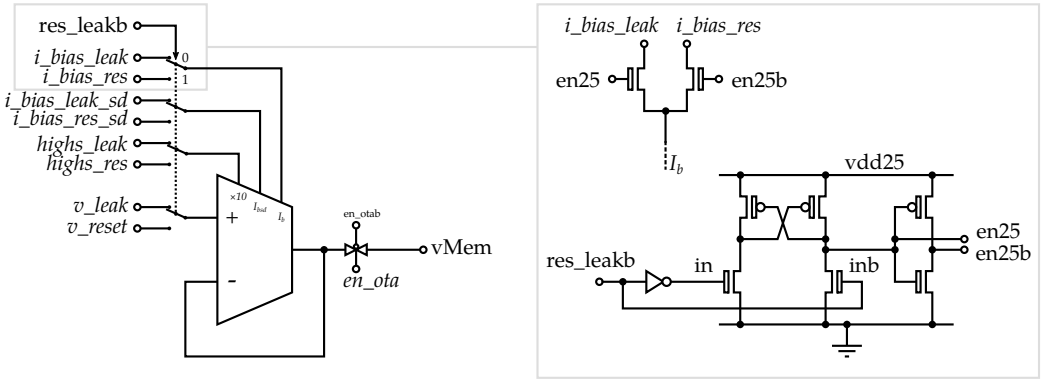


Figure 3.15: The main circuit of the leak/reset term is an OTA with a configurable bias, source-degeneration bias and switchable output current multiplier. All input signals are switched synchronously by the reset signal (left). The switching of current cells is exemplified on the right for the bias current: A level shifter converts the 1.2 V reset signal to 2.5 V signals which controls two NMOS devices that switch the output of current cells. The switching of voltage inputs is shown in fig. 3.16.

Figure 3.15 shows how an OTA is used to implement the leak and reset conductance. The reset signal ("res_leakb") switches the OTA to reset mode by switching all input signals from the leak- to the corresponding reset value. The selection of the positive input voltage from *v_leak* and *v_reset* is accomplished as shown in fig. 3.16: The voltage cells of the capacitive memory are decoupled through source follower circuits; the negative input of the OTA is processed in the same manner for symmetry. The output voltages V_+, V_- are then used as inputs for the OTA circuit. Note that I_b in fig. 3.16 is also switched between *i_bias_leak* and *i_bias_reset* during each reset cycle by the switching mechanism, so the bias current for the source followers may be different in the leak and reset phases. Because the paths for V_- and V_+ (current mirror M_{15} to M_{14} and then M_1 to M_6) are symmetric, this only introduces a common mode dependency for the output OTA (fig. 3.16 right) in first approximation. However, neither of the bias currents *i_bias_leak* and *i_bias_reset* may fall below a certain threshold to ensure the functioning of the input stage.

An operational transconductance amplifier generates a current which is proportional to the difference of the input voltages $I_{\text{out}} = g_m \cdot (V_+ - V_-)$. A simple CMOS implementation is shown in fig. 3.17. (cf., e.g., Wong and Salama (1986)). A bias current I_b is split into two paths over M_1 and M_2. In saturation, the drain current is approximated by a function of the gate-source voltage in the quadratic model: $I_D = K(V_{\text{GS}} - V_{\text{th}})^2$, where $K = \frac{\mu C_{\text{OX}}}{2}\frac{W}{L}$ is a constant for the transistor with given dimensions W, L, charge carrier mobility μ and specific gate oxide capacitance per area C_{OX}. The resulting current difference $I_+ - I_-$ is proportional to $V_+ - V_-$ up to third order and proportional to the $\sqrt{(}I_\text{b})$. The remaining current mirrors in fig. 3.17 (M_3 – M_8) are in place to produce the current difference at I_{out}.

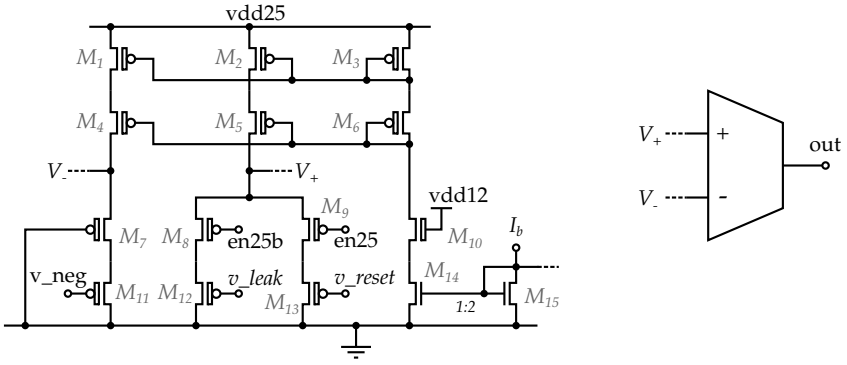


Figure 3.16: Switching of input voltages in the leak/reset OTA.

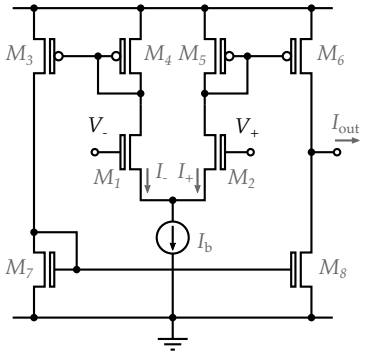


Figure 3.17: Concept of OTA implementation, reproduced from Fig. 1 in Wong and Salama (1986)

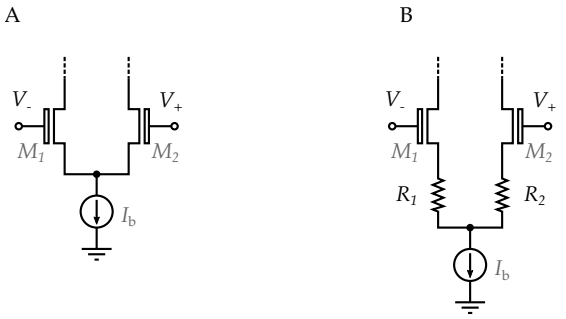


Figure 3.18: Concept of source degeneration for transconductance amplifiers. The differential pair of fig. 3.17 without (**A**) and with (**B**) source degeneration.

One method to extend the linear range of an OTA is to use source degeneration. Figure 3.18 shows the concept that is used in the leak and adaptation OTAs. A resistor is placed in both branches of the differential pair in the amplifier's input, which reduces the gain and increases the linearity of the amplifier. (For a detailed review of linearization techniques for OTA circuits see, Sanchez-Sinencio and Silva-Martinez (2000).) In the DLS3 neuron, two different implementations of source degeneration (fig. 3.18) are used, as shown in fig. 3.19 A and B. The transistors $M_{1A/B}$ and $M_{2A/B}$ are placed in the differential pair path and are biased in both cases using a reference current, I_{bsd}. In the first case, an NMOS, in the second a PMOS transistor is used to produce a gate voltage from I_{bsd}. In the PMOS case (A), the source current of M_A flows over M_{3A}, which acts as a current source of a differential pair. The implications of the two implementations are discussed in section 3.3.3.

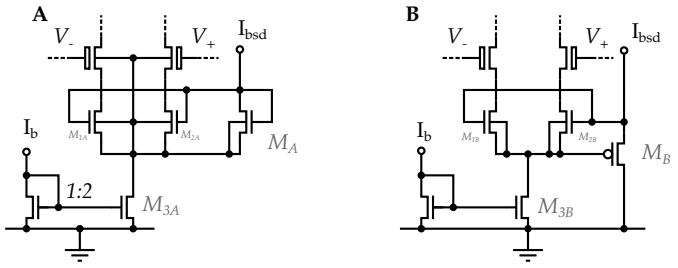


Figure 3.19: Alternative implementations of fig. 3.18B. **A** is used in the leak OTA while **B** is utilized in the adaptation and synaptic input OTAs. The difference is in transistors M_A and M_B which produce the bias voltage for the source degeneration transistors M_{1A}, M_{2A} and M_{1B}, M_{2B}, respectively. In A, the bias current I_{bsd} flows to ground over M_{3A} shared with the differential pair. In B, the bias voltage is generated using a PMOS device and the I_{bsd} is independent of the current mirror transistor M_{3B}.

3.2.4 Adaptation term

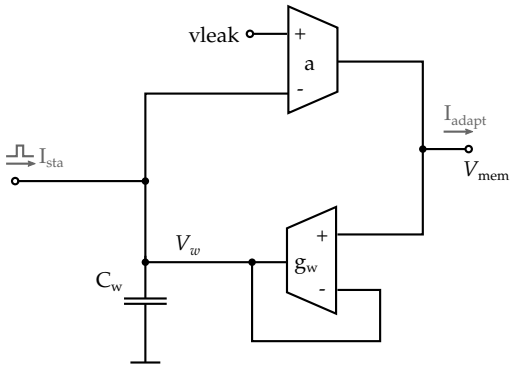


Figure 3.20: Concept of the adaptation circuit. (cf. Millner, 2012, Figure 3.12)

The concept of the adaptation term is shown in fig. 3.20. In addition to the membrane capacitor, a second capacitor C_w holds the voltage V_w which represents

as the dynamic variable equivalent of the adaptation current w in the AdEx model (eq. (1.4)). The implementation, first described in Millner (2012), relies on two transconductance circuits that couple membrane and adaptation voltage:

$$C_w \cdot \frac{\mathrm{d}V_w}{\mathrm{d}t} = I_{gw} \quad = g_w \cdot (V_\mathrm{m} - V_w) \tag{3.1}$$

$$w = -I_\mathrm{adapt} \quad = -g_a \cdot (\mathrm{V}_\mathrm{leak,a} - V_w) \tag{3.2}$$

Transforming these equations yields

$$\frac{\mathrm{d}w}{\mathrm{d}t} = g_a \cdot \frac{\mathrm{d}V_w}{\mathrm{d}t} = \frac{1}{C_w} g_a g_w (V_\mathrm{m} - V_w) \tag{3.3}$$

$$= \frac{1}{C_w} g_a g_w (V_\mathrm{m} - \frac{w}{g_a} - V_\mathrm{leak,a}) \tag{3.4}$$

$$= -\frac{g_w}{C_w} w + \frac{g_w g_a}{C_w} (V_\mathrm{m} - V_\mathrm{leak,a}) \quad . \tag{3.5}$$

By identifying $\tau_w := \frac{C_w}{g_w}$ and $a := g_a$ we recover the original AdEx equation

$$\frac{\mathrm{d}w}{\mathrm{d}t} = \frac{-w}{\tau_w} + \frac{a}{\tau_w} (V_\mathrm{m} - V_\mathrm{leak,a}) \quad . \tag{3.6}$$

One important technical difference is $V_\mathrm{leak,a}$ which takes the place of V_leak in eq. (1.4). To counteract the input offset of the leak and g_a-OTAs, separating the parameters for V_leak in the technical implementation is desired to improve the calibrability of the circuit.

The current I_sta in fig. 3.20 represents the technical implementation of the spike-triggered adaptation parameter b. After each spike that is emitted by the neuron, the charge on C_w is incremented by a configurable amount using a current pulse of defined duration T_b and magnitude I_b.

Using the translation above one calculates the change in adaptation current as

$$b := \Delta w = a \cdot \Delta V_w \tag{3.7}$$

$$= a \cdot \frac{T_b I_b}{C_w} \tag{3.8}$$

Here, a limitation of the described implementation becomes apparent: Because T_b and I_b are limited, the maximally available value of b scales proportionally with a. For example, in this implementation it is impossible to have spike-triggered adaptation without sub-threshold adaptation (i.e., $a = 0$ and $b \neq 0$).

An advantage which is mentioned in Millner (2012) is the fact that V_w is always pulled towards V_leak and thus the absolute difference between V_w and V_m is small in general. This implies that the "a"-amplifier stays within its operation regime even for a small linear range of the amplifier.

3.2.5 Circuit implementation of sub-threshold adaptation

Figure 3.21 shows a detailed schematic of the sub-threshold adaptation circuit which implements the concept shown in fig. 3.20. The g_w-OTA in fig. 3.20 is replaced by a

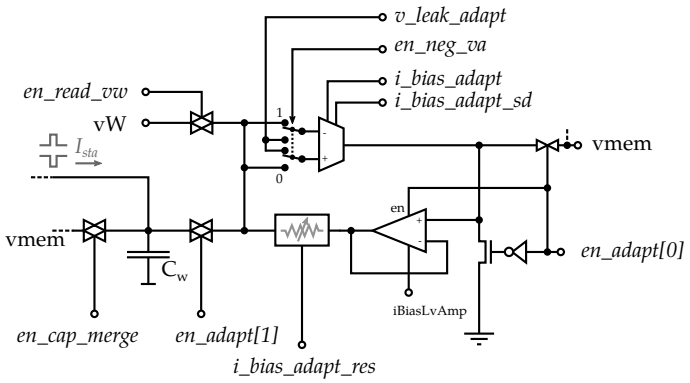


Figure 3.21: Circuit implementing sub-threshold adaptation.

functionally equivalent combination of a buffer amplifier and configurable resistor element. The output OTA has a switchable sign which is implemented by thin-oxide two-input multiplexers, controlled by the parameter *en_neg_va*. As described in section 3.2.4, the parameter *v_leak_adapt* is separate from *v_leak* to provide a possibility of compensating the input offset of the "a"-OTA. This OTA is closely based on the leak-ota in the previous chip implementation (Aamir et al., 2016).

To configure the adaptation circuit, the source-degeneration bias (*i_bias_adapt_sd*) is used as a per-neuron parameter. The parameter *i_bias_adapt* is shared between all neurons.

The voltage of the adaptation capacitance can be accessed during circuit measurements and usage of the chip. Because the spike-triggered adaptation circuit does not enforce an upper limit of V_w, an additional, thick-oxide transmission gate is introduced, which is switched by *en_read_vw*. In case it is enabled and the voltage on V_w rises above 1.2 V, current will flow from C_w onto the neuron membrane through the read-out multiplexer (fig. 3.13).

The *en_adapt* parameter is used to enable the adaptation circuit. This is used to disable the input amplifier and disconnect the adaptation term from the membrane (*en_adapt[0]*) and to disconnect the adaptation capacitor (*en_adapt[1]*). When the adaptation term is not used, the adaptation capacitance can be added to the membrane capacitance using the *en_cap_merge*, to increase the maximally possible membrane time constant. This improves the use of chip resources, because neurons with large time constants can be implemented using one neuron compartment rather than two.

3.2.6 Circuit implementation of spike-triggered adaptation

Figure 3.22 shows the implementation of the current pulse generation that increases the adaptation voltage V_w after each spike. The adaptation signal from the digital back end arrives at the "adaptEn" pin, which is switched to a high voltage for a configurable duration T_b (which is controlled by the parameter *adaptation_time* in the following simulations). The multiplexer s_1 switches between positive and negative sign of adaptation. In addition to switching s_1, the control setting *en_pos_vw* also

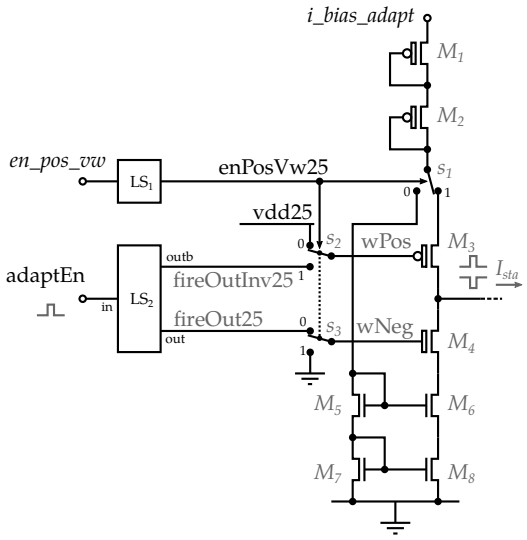


Figure 3.22: Circuit that generates the current pulse for spike-triggered adaptation.

selects the corresponding transistor M_3, M_4, so either the adaptation bias current is used directly or its mirrored version from $M_5 - M_8$. The level-shifters LS_1 and LS_2 convert the 1.2 V signals *en_pos_vw* and "adaptEn" to control the attached thick-oxide circuits.

3.2.7 Exponential term

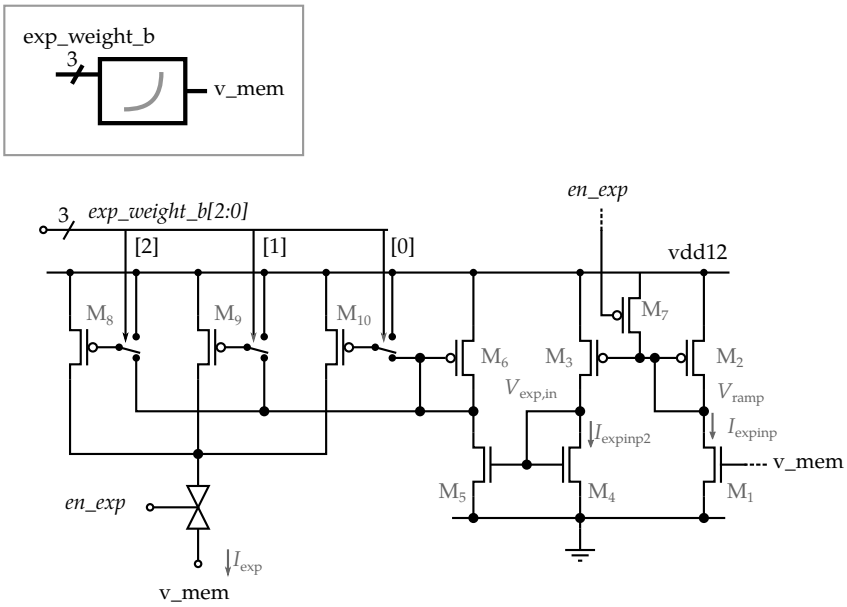


Figure 3.23: Schematic of exponential term The inset shown the symbol and outside connections used in fig. 3.8.

Figure 3.23 shows the implementation of the exponential current. The membrane voltage at the gate of M_1 is converted to an approximately inverted voltage V_{ramp}

at M_2. The transistor M_3 is operated in the sub-threshold regime and produces an exponential current as a function of V_{ramp}. This current is mirrored in $M_4 - M_5$, and then mirrored again in a three-bit current DAC, to produce the output current I_{exp}. A transmission gate is available to fully disable the current using the parameter *en_exp*, which additionally enables the pull-up for transistor M_7 to prevent current consumption in the disabled state.

The slope factor of the exponential current is thus fixed by the subthreshold characteristic of M_3 and the voltage scaling at the input stage $M_1 - M_2$. The exponential threshold is effectively varied by the three bit setting of the current multiplicator. This reduction in configurability is in contrast to the implementation in the 180 nm HICANN chip, where threshold voltage and slope factor were adjustable by analog parameters. The tunability of these parameters is expected to be included in future revisions of the device.

3.2.8 Inter-compartment connectivity

The inter-compartment connectivity of the neurons consists of switches that allow a configurable connectivity to neighboring neuron compartment circuits as well as one tunable resistance per neuron compartment.

Figure 3.24 shows the concept of tunable resistor circuits used in the neuron implementation (see, e.g., Tajalli et al. (2008)). For $V_{\text{SD}} \geq 0$, a single transistor shows a high resistance which can be adjusted by controlling V_{SG}. For $V_{\text{SD}} < 0$, the resistance drops significantly. In the symmetric implementation, two transistors are used to combine the resistance $R = R_{M1} + R_{M2}$, ensuring that $V_{\text{SD}} \geq 0$ is true for one of the two transistors, while their gate-source voltage is controlled by M_3.

Figure 3.25 shows the detailed structure of the inter-compartment resistor. Building upon fig. 3.24, the present implementation uses a total of four transistors, arranged symmetrically, to reduce the voltage drop occurring at each individual transistor. In addition to the main current bias, *i_bias_nmda*, the two digital control bits *ib_nmda_div4* and *ib_nmda_mul4* control the resistance value of the device. This is accomplished by switching the current mirror voltage bias (fig. 3.25 top). The bias current from the capacitive storage is then either mirrored directly (both control bits zero), split into four equal parts effectively lowering the bias current when *ib_nmda_div4* is enabled, and multiplied in the current mirror output if *ib_nmda_mul4*

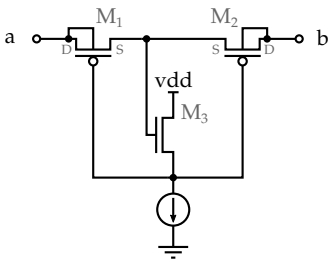


Figure 3.24: Concept of tunable resistor (Redrawn after Fig. 2 in Tajalli et al. (2008))

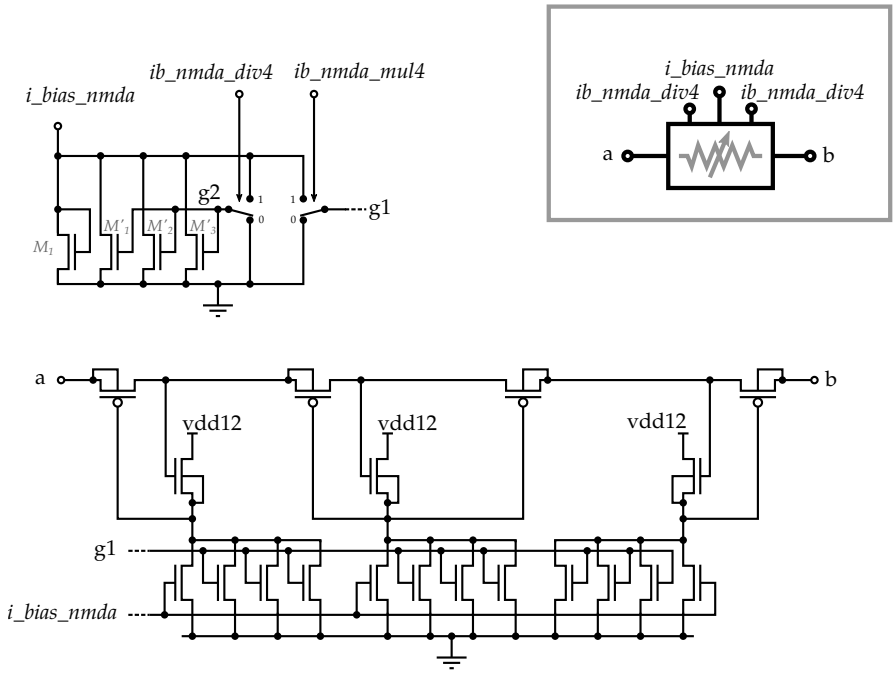


Figure 3.25: Implementation of tunable multi-compartment resistor.

is enabled. (Enabling both configuration bits is equivalent to using a wider current mirror with both bits disabled.)

3.3 Monte-Carlo calibration

3.3.1 Synaptic input

The output OTA of the synaptic input (fig. 3.12) has an input offset that is caused by mismatch. This mismatch is counteracted by making the parameters *v_syn_exc* and *v_syn_inh* individually tunable. In typical operation, the goal is to have an identical offset current from each synaptic input, such that other components can be calibrated individually without taking into consideration the behavior of the particular offset of two synaptic input circuits in the neuron. Because the offset current provided by *i_mem_off* is always positive, the excitatory synaptic input is calibrated with *i_mem_off* = 500 nA, so that the excitatory synaptic offset and the input current cancel each other. This makes it possible to effectively use negative compensation currents at later stages (section 3.3.3) by setting *i_mem_off* < 500 nA. This concept of calibration-based offset cancellation is not new and was implemented for version four of the HICANN chip (Koke, 2017), and was implemented by Sebastian Billaudelle during his internship for the simulated DLS neuron and by Yannik Stradmann for the DLS2 chip (Stradmann, 2016; Aamir et al., 2016).

Figure 3.26 shows the calibration procedure of the synaptic input circuits. A realistic procedure is used which means that the membrane potential is used as the observable. Figure 3.26 A shows a single experiment: A reference measurement of the membrane potential is taken with disabled synaptic input. Then, a bisection on the *v_syn* parameter is performed to restore the membrane potential measured before. An example of this bisection is shown in panel B for the membrane potential and in C for the resulting synaptic current. Panel D shows the distribution of *v_syn* values after calibration. These values correspond to the distribution of the input offset of the OTA, which is on the order of 50 mV. The mean of the distributions does not coincide because the excitatory input compensates a 500 nA offset, as described above.

Figure 3.27 shows the evaluation of the calibration routine. The resting potential with enabled and disabled synaptic inputs is compared. A large spread of the membrane resting potential is caused by the yet uncalibrated leak term. Enabling the synaptic input does not shift the resting potential. The histogram of the difference between the calibrated and uncalibrated case is shown in panel B. As control, the calibration data from the *tt* sample is applied, which showcases the expected deviation in an uncalibrated case of a more than sixty-fold increase in the sample standard deviation.

Panel D shows the effect of an incorrectly disabled synaptic input, which is included here for demonstration purposes. It is not sufficient to use the *en_syn_i_exc* setting to disable the synaptic input, but it is also necessary to guarantee that the voltage on the membrane-opposed side stays between 0 and 1.2 V. This is accomplished by ensuring that the positive input terminal of the OTA is below the negative one (fig. 3.12), e.g., *v_syn_exc* = 0 V, *v_syn_exc* = 1.8 V.

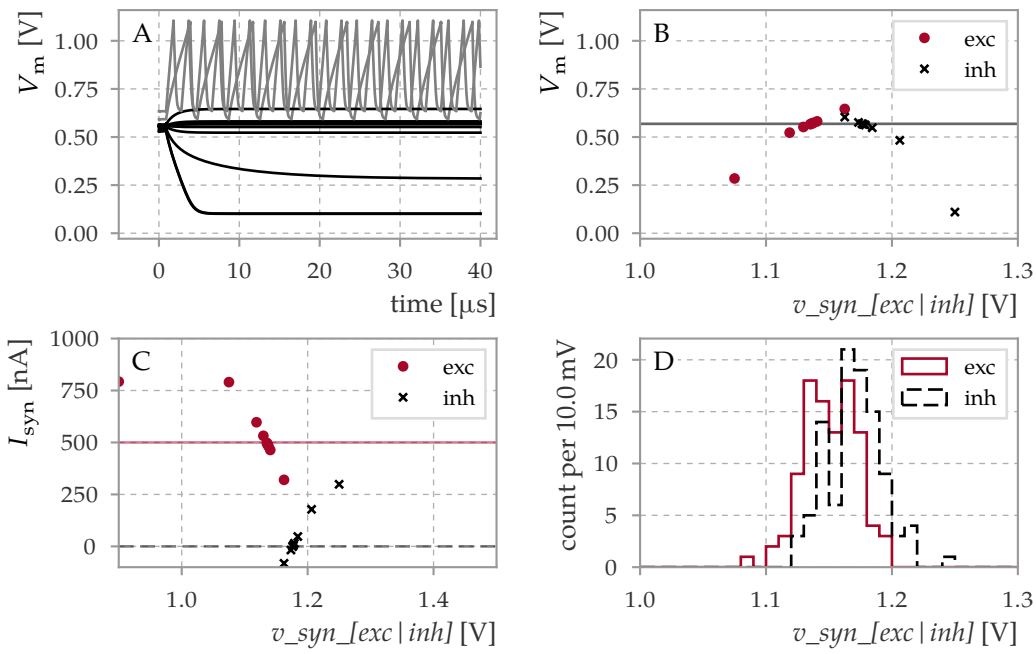


Figure 3.26: Calibration of synaptic input offset. **A:** Voltage traces of individual simulations during the bisection of *v_syn_exc*. **B:** Resting membrane voltage at the end of the simulation in dependence of the corresponding *v_syn_[exc | inh]* value. The horizontal line denotes the target value which is the resting membrane potential with disabled synaptic inputs. **C:** Resulting current of the calibrated synaptic input for each calibration step. During calibration of the excitatory synaptic input the offset current parameter *i_mem_off* is set to 500 nA to provide an equal current offset tuning range towards positive and negative currents. Thus, the resulting excitatory offset current converges towards 500 nA while the inhibitory synaptic offset converges to zero. The algorithm operates on the membrane voltage (B), and the current (C) is shown for cross-validation. **D:** Resulting calibration values for *v_syn_exc* and *v_syn_inh* parameters for 100 Monte-Carlo samples.

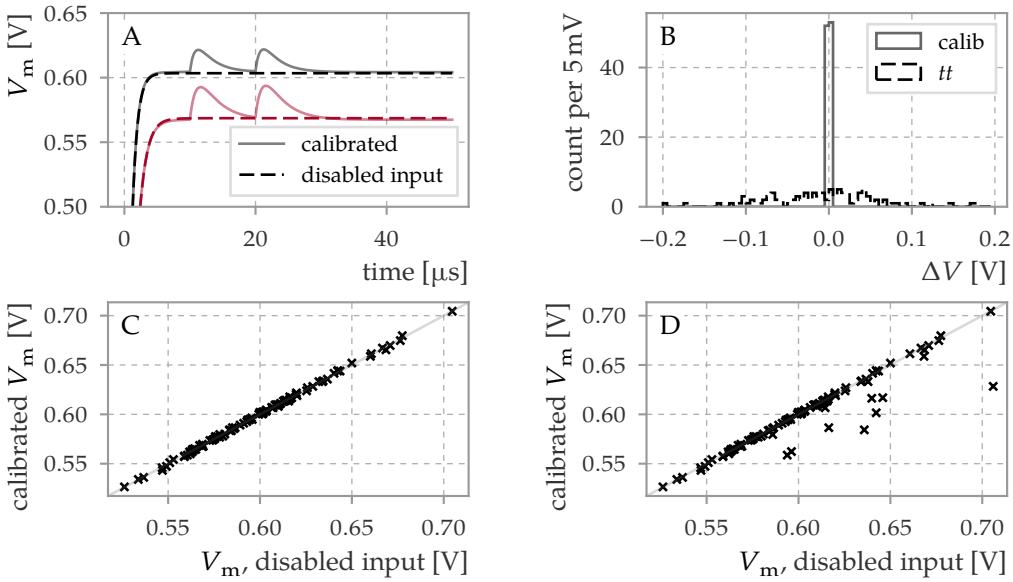


Figure 3.27: Evaluation of synaptic input calibration. **A:** Individual simulation with disabled (- - -) and enabled (——) synaptic inputs. Only the synaptic offset calibration is applied. The rise of the membrane potential in the beginning stems from the neurons being in the reset state with *v_reset* being set to 0.4 V. **B:** Histogram of $\Delta V := V_{\text{calibrated}} - V_{\text{disabled}}$ (—) for 100 Monte-Carlo samples of the simulation shown in A. The mean and standard deviation is (-0.0 ± 1.1) mV. The same simulation with the calibration data of the typical (*tt*) sample applied to all Monte-Carlo samples is shown as comparison (dashed lines, (11 ± 74) mV). **C:** All values before and after enabling the offset calibration, shown for the same data as in B. The calibration of the resting potential is not used, so the resting potential varies, but enabling the synaptic input with calibrated values does not change the resting value **D:** Example of incorrectly disabled synaptic input, that is included for documentation purposes. The control voltages *v_syn_inh* are kept at 1.2 V for the "disabled" case. This leads to leakage in the thin-oxide transmission gates which switch the synaptic input circuits and affect the resting potential in a significant number of samples. (cf. section 3.4.2)

3.3.2 Synaptic time constant

The synaptic time constant is generated as a combination of a tunable resistance and the capacitance of the synaptic input line, which is composed of the capacitance of the line itself and an additional capacitor (fig. 3.12). The resistor is a single-sided implementation of the resistor concept shown in fig. 3.24 (Aamir et al., 2017b). Because the resistance value has a complex dependency on the bias current, the tuning curve is characterized directly: An fit of an exponential function to the time course of the synaptic current is used to determine the time constant value for varied bias currents. The points are fitted using a rational transformation with degrees with exponents in the range of $\{-4, \ldots, 1\}$. The resulting translation from bias to

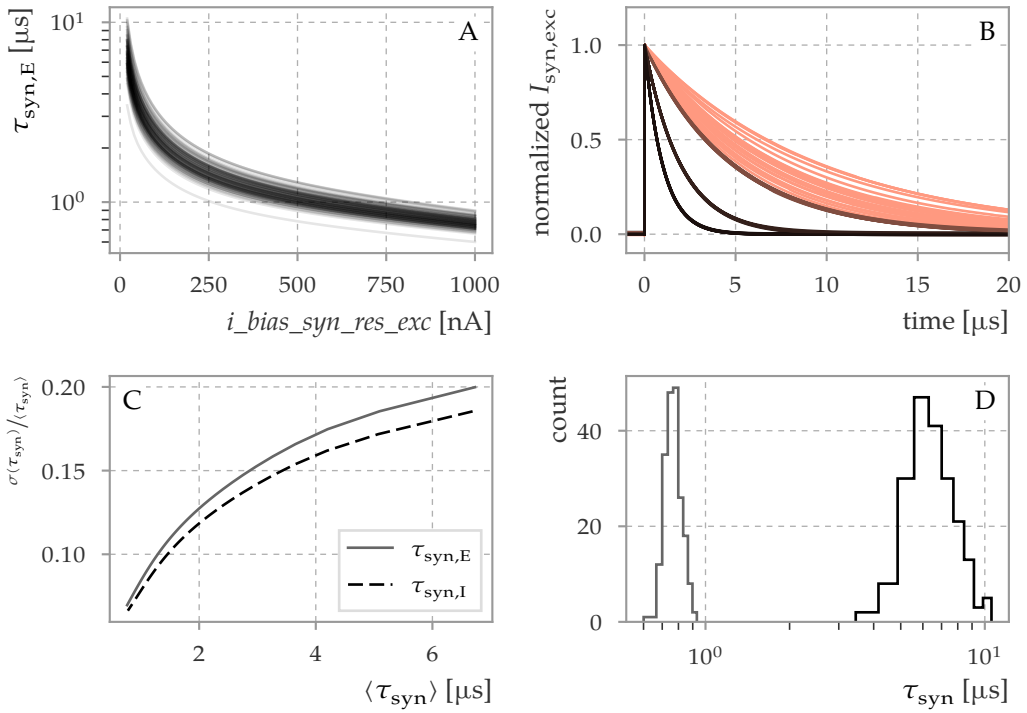


Figure 3.28: Calibration of the synaptic time constant. **A**: Synaptic time constants obtained from 100 Monte-Carlo samples. **B**: Verification using target synaptic time constants of 1 µs, 2 µs, 5 µs and 10 µs. The individual post-synaptic currents are normalized to the initial current step for each of the 20 Monte-Carlo samples, so that all curves start at a unit-less magnitude of 1. The time to decay to $1/e$ is (1.015 ± 0.040) µs, (2.01 ± 0.03) µs, (4.92 ± 0.06) µs, and (7.28 ± 1.40) µs, respectively. **C**: Relative mismatch of the synaptic time constant, calculated as sample standard deviation divided by the mean time constant at a constant bias current. **D**: The distributions of minimum and maximum time constants are shown. The inhibitory and excitatory time constants are included. (min = (0.77 ± 0.05) µs; max = (6.7 ± 1.3) µs). The distribution of the covered range is $^{max}/_{min} = 8.6 \pm 1.0$.

time constant is shown in fig. 3.28 A. To verify the quality of the fit, the calibration is applied to four target values (fig. 3.28 B). The width of the distribution is significantly reduced for moderate values of 2 µs and 5 µs to approximately one percent, from more than ten in the uncalibrated case (fig. 3.28 C). When the target is exceeded, the bias current is truncated and the width of the distribution increases.

The achieved limits for the time constant are shown in fig. 3.28 D. While the range that is guaranteed to be covered by every neuron has an extent of only a factor of approximately three (0.9 µs to 4.8 µs for the 95 % range on each side), each individual time constant can be varied by a factor of at least seven (95 % of all samples) because the minimal and maximal value are highly correlated. The spread may be beneficial for networks that intrinsically exploit heterogeneity because then, longer and shorter time constants can be assigned to suitable circuits. Mapping an existing network, however, may necessitate extensive blacklisting of circuits on the basis of the given network or a calibration-aware placement of neurons to accommodate the neuron parameter requirements.

3.3.3 Leakage and reset transconductance

Membrane time constant

The leak/reset OTA is a central component of the neuron circuit because it implements the fundamental functions of the LIF model. Figure 3.29 shows the basic characteristics of the OTA. The source degeneration bias *i_bias_leak_sd* is used to adjust the gain, while the bias current *i_bias_leak* is a technical bias that affects the base gain and is used as the bias currents for the input source followers (fig. 3.16). In general, lowering the source degeneration bias lowers the gain, and, at a certain point, the linear range is significantly increased – this is visible in Figure 3.29 A and B. Because of the bias sharing in the OTA between I_{bsd} and I_{b} (fig. 3.19A), the condition

$$i\_bias\_leak \cdot 2 \geq i\_bias\_leak\_sd$$

must be fulfilled. Otherwise, the OTA current-voltage characteristic is distorted and the leak is not functional (fig. 3.29 C for high values of *i_bias_leak_sd*). This effect is quantified in fig. 3.29 D: The transconductance increases with increasing *i_bias_leak_sd* in the valid operating range and decreases again for higher values of that current. This behavior is affected by mismatch and has to be taken into account by the calibration algorithm.

Figure 3.30 shows the response of the leak OTA for different values of the positive terminal, which is *v_leak* in this case. For high values of the source degeneration bias, the linear region is approximately 200 mV (fig. 3.30 A); it is increased for lower bias values (fig. 3.30 B). There is a strong input common mode dependency in both cases, which means that the output current does not only depend on the difference of the voltages at the two input terminals. The most important effect is the lowered saturation voltage at high V_{m}, which will be discussed in detail in section 3.3.4. The

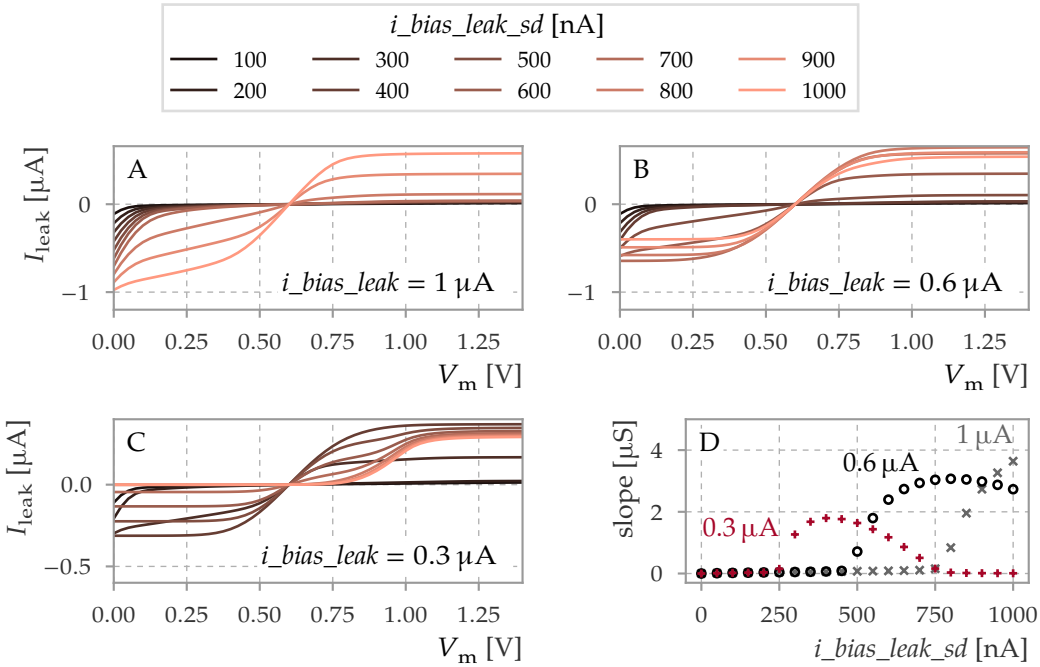


Figure 3.29: Simulated voltage-current relationship of the leak and reset OTA (*tt* corner). In general, the bias parameter *i_bias_leak* determines the saturation voltage while *i_bias_leak_sd* is used to adjust the transconductance. The value of *i_bias_leak* is set to 1 µA (**A**), 0.6 µA (**B**) and 0.3 µA (**C**). Only configurations with $i\_bias\_leak \cdot 2 \geq i\_bias\_leak\_sd$ are valid, which is particularly visible for the case of high *i_bias_leak_sd* (C), where only positive output current is produced but the output current stays zero for $V_m < 0.6\,V$. Panel **D** shows the slope of the OTA characteristic at $V_m = 0.6\,V$ for the values of *i_bias_leak* that are used in A–C. The current is scaled when using the *highs* switch (cf. fig. A.2).

second important feature is that the zero crossing of the V-I characteristic follows the positive terminal in the range between 0.4 V to 0.8 V.

The membrane time constant is not a simple function of the source degeneration bias, as seen in fig. 3.31 A. Further, the slope of the current-voltage characteristic changes significantly with V_m.

Considering this, a linear interpolation in 40 sections is used to create the inverse look-up for the time constant. The voltage-dependent time constant can not be easily counteracted by calibration because the operation regime of a neuron is dependent on network activity and can not be canceled by a general-purpose calibration procedure. To reduce simulation time, the time constant is not measured directly but the leak current difference at 100 mV is recorded and the time constant is calculated using the nominal capacitance of 2.36 pF.

It is important to note that V_m is kept constant for each sweep – the result accurately reflects the effective OTA transconductance at a given membrane voltage. This is only possible if a voltage clamp is possible for the neurons. If using a current

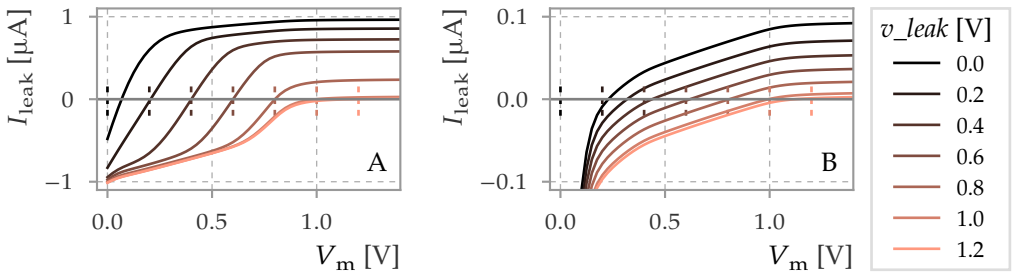


Figure 3.30: Effect of the *v_leak* parameter for the leak and reset OTA (simulation in *tt* corner). **A**: *i_bias_leak_sd* and *i_bias_leak* set to 1 µA. **B**: *i_bias_leak* = 1 µA *i_bias_leak_sd* = 0.6 µA. The dashed vertical lines denote the *v_leak* value that was set for the corresponding curve.

stimulus, the resting value of the membrane potential must be precisely controlled, or significant measurement bias is introduced due to the voltage-dependent membrane time constant.

The result of the simulation shows the expected performance of the leak OTA. Distinct low- and high conductance regions exist, with ranges for τ_m of 25 µs to 100 µs for the low-conductance and approximately one microsecond for the high-conductance regime. (This analog effect is not to be confused with the *highs* setting, which is a digital multiplier for the output current of the OTA.) The transition between the two regimes happens on a small scale in the parameter space on the order of 100 nA. The location of this transition is strongly influenced by mismatch (fig. 3.31 A), which causes the uncalibrated use of the circuit to be impractical for small time constants at full capacitance (fig. 3.31 B). The total covered range of the circuit from 2 µs to 90 µs is significantly larger than previous implementations if higher conductance regime is used even without using the *highs* option or changing the membrane capacitance (fig. 3.31 C and D). In case only the lower conductance regime is utilized, the available ratio of maximal to minimal time constant is approximately four and switching the capacitance is essential to utilize the full available range for the membrane time constant. A small number of Monte-Carlo samples does not reach the higher conductance regime within the available configuration space (fig. 3.31 C), which has to be taken into account when using this mode, for example by blacklisting the corresponding circuits or enabling *highs_leak*.

Resting potential

Within this section, the following nomenclature is used: the *resting potential* refers to the steady-state of the membrane potential, V_m. *v_leak*, on the other hand, refers to the technical voltage parameter that controls the circuit operation.

There are several possibilities to calibrate the resting potential of the neuron circuit. One possible method is to measure the input offset of the leak OTA and compensate for it, as it is done for the other calibrated quantities. There are several issues with this approach: First, the offset depends on the bias of the OTA, so the offset lookup must be adapted to the desired time constant when applying the

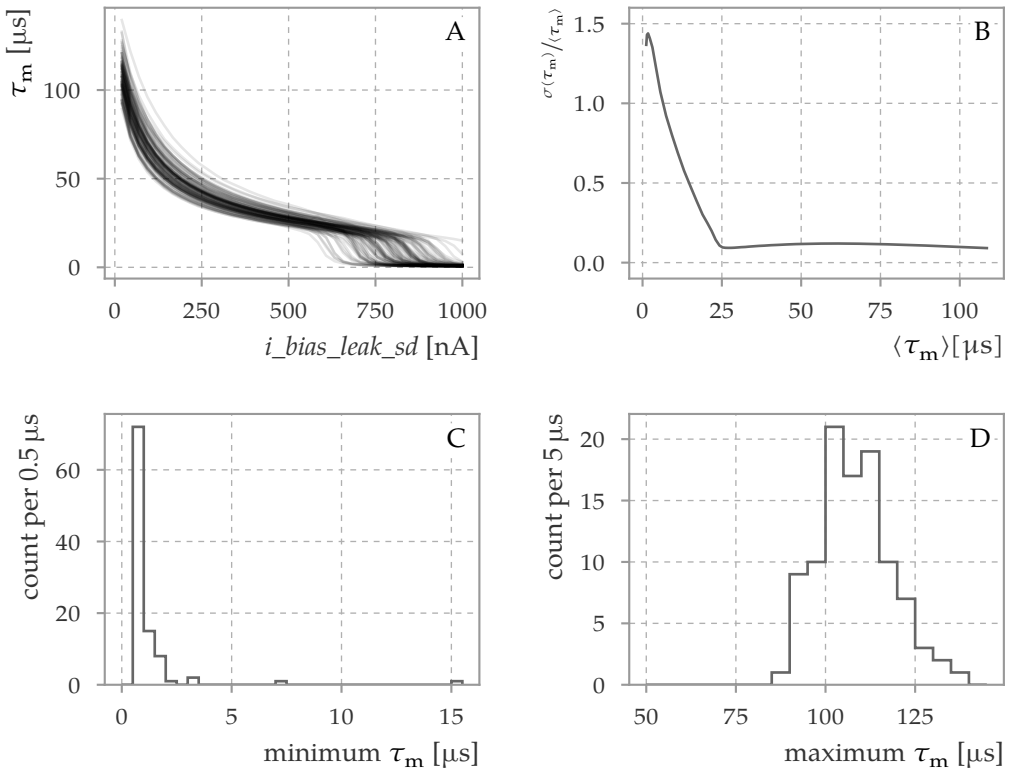


Figure 3.31: **A**: Dependency of the time constant on the source degeneration bias in the leak OTA. Data for 100 Monte-Carlo samples are shown. **B**: Ratio of variance and mean of the time constant **C**: shows the minimum and **D** the maximum of the achieved membrane time constant. The distribution is (1.2 ± 1.6) µs for the minimum and (108 ± 10) µs for the maximum. At least 95 % of the samples achieve a minimal time constant smaller than 2.1 µs and at least 95 % a maximal time constant larger than 93 µs. All simulations use the largest available membrane capacitance of approximately 2.36 pF. *i_bias_leak* is set to 1 µA.

calibration. Second, tuning the parameter *v_leak* may not suffice to reach all desired values of the resting potential, especially for high membrane time constants. In this case, the offset current *i_mem_off* must be used to change the resting potential. Third, the offset and the time constant depend on the resting potential, so a multi-dimensional lookup

$$(V_{\text{leak}}; \tau_m) \mapsto (\textit{i\_bias\_leak\_sd}; \textit{v\_leak}; \textit{i\_mem\_off})$$

must be performed. Because the relatively fast change of the OTA properties with the control parameters, a fine grid of measurements is required even for a single parameter (section 3.3.3). A two-dimensional scan was tested by the author but proved too time-consuming for a simulation in the required detail. [20]

[20] A single time constant estimate takes on the order of one minute to complete. The available

In response to the above considerations, an alternative approach to calibration was implemented that abandons the idea of individually calibratable sub-circuits.[21] An additional offset current parameter was added to the neuron circuit to allow canceling arbitrary offset currents. It is implemented as a direct connection from a current cell in the capacitive memory (fig. 3.8). This entails the limitation that only positive currents are available.

It was proposed that the resting potential calibration happens *in the loop*, which means that after a parameter change, consecutive measurements of the hardware response are used to tune parameters to obtain the desired response. This approach benefits from the high acceleration factor and on-chip processing that is provided by the plasticity processor (section 3.1.2) but has, a computational overhead over a single-step calibration.

One approach that the author proposes for the hardware implementation is a bisection of the measured resting membrane voltage over the offset current. The advantage is that the algorithm is simple enough to be implemented by the plasticity processor and the membrane potential can be read out in parallel using the path to the correlation ADC (Hartel et al., 2017, section 12.3.3).

For the simulation, the bisection would require approximately ten iterations to match the precision of the ten bit capacitive memory. To reduce simulation time, the offset current is determined by recording the required current that keeps the membrane at the target voltage. Despite this simulation shortcut it is demonstrated that the required current can be set in an evaluation procedure in the next section.

Verification of leak calibration

The Monte-Carlo calibration procedure is verified in transient simulations (see fig. 3.32). Figure 3.32 A and B show the response of the simulated neuron membrane with applied time constant and resting potential calibration, as described in section 3.3.3. The applied current step ends at 50 µs simulation time. Note that its magnitude is chosen inversely proportional to the time constant, so the different height, especially in panel A, is caused by the mismatch of the leak conductance after calibration.

Due to the switching between high and low resistance mode, the utilized bias varies strongly for low target τ_m (fig. 3.32 C). The offset current that is required to stabilize the membrane potential also varies with the τ_m setting (fig. 3.32 D).

The achieved precision and systematic deviation is shown in fig. 3.32 F. Especially for low time constants the fast change of the conductance leads to a systematic mismatch, primarily due to the linear interpolation of measured grid points. When the available range is exceeded, the bias current is truncated and the variance increases as well, which is seen for the case of 100 µs. Applied to the resting potential

computing resources available for parallelization are limited to approximately twenty parallel processes.

[21] The decision was made following a proposal of Johannes Schemmel with contributions by Sebastian Billaudelle; it was implemented by Syed Ahmed Aamir.

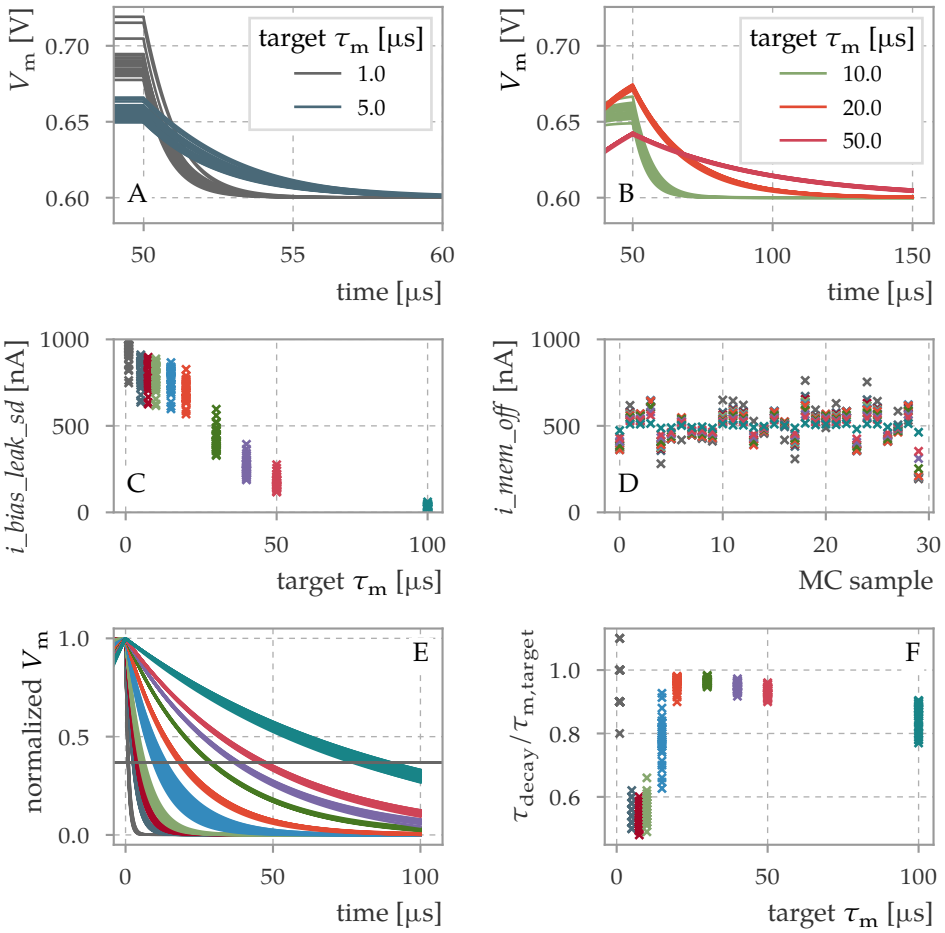


Figure 3.32: Verification of the simulated calibration for the membrane time constant. **A** and **B** show the result of the calibration for multiple target values of the time constant. **C**: The post-calibration value of the bias is shown. **D** shows the value of the offset current for each simulation. **E**: All evaluation simulations, normalized to the same range. **F**: Ratio of the decay time and the target time constant. The decay time is extracted from E as time to reach $1/e$ in the leak OTA of the initial deflection.

calibration, fig. 3.32 D shows the required offset currents to produce the results in panels A and B.

3.3.4 Reset current

One major change from DLS2 to DLS3 neuron is the introduction of a variable-strength reset which is implemented by combining the function of the old leak OTA and the reset circuit into the leak term. The adjustable reset conductance is required to allow more control over the reset for the emulation of multi-compartment neurons with active dendrites: Long-lasting plateau potentials are implemented using a long refractory time with a high reset voltage in compartments that are designated as active. In that case, a tunable reset conductance may be desired to control the

amount of effect of external input on the membrane, between no reset at all to the maximal possible value. Another example is the implementation of the LIF sampling paradigm (section 3.5.2), where setting the reset voltage and bias parameters to equivalent values of leak voltage and biases allows for a different interpretation of the hardware membrane potential as the free membrane potential.

The high conductance switch *highs_res* is used to scale the output current of the OTA by a factor of approximately 10 to provide a high reset conductance when a strong reset is required.

Figure 3.33 shows the reset current as a function of membrane voltage and bias current. The OTA current saturates at approximately 0.9 V and the value of the saturation current depends on the reset voltage. For $v\_reset = 0.96\,\mathrm{V}$, the saturation happens at below one microampere even with enabled *highs_res* (panel A). Note that the operation range of the membrane voltage is ideally in the range of 0 V to 1.2 V.

Figure 3.33 B shows the dependence of the saturated reset current on the reset bias, estimated from the current at 1.4 V (see panel A). The maximum of the saturation current shifts with the reset voltage as well. The reset calibration is thus implemented to maximize the robustness of neuron operation. The maximal reset current is recorded for eleven steps of *v_reset* and the bias that generates the maximal reset current for the closest reference *v_reset* is set. It is important to note that this is not equivalent to setting the maximal reset conductance: The slope of the current-voltage relationship at the reset voltage is not necessarily maximal for the same setting as the highest possible reset current.

It must be justified why *i_bias_res* and not the source degeneration bias *i_bias_res_sd* is used. Figure 3.34 shows that, for low *v_reset*, the maximal reset current decreases with decreasing *i_bias_res_sd* (panel A), while for high reset voltages (panel B and D) it does not significantly increase from its value at 1 µA. At constant *i_bias_res*, the maximum reset current has a minimum as a function of *i_bias_res_sd* (panel C), but its maximal value is reached at the maximal *i_bias_res_sd*.

Figure 3.35 shows the maximal reset current for a sample of mismatch simulation. The decrease of the maximal current as well as the decrease of the optimal *i_bias_res* with increasing *v_reset* is clearly visible. It is also shown that using the maximal value for *i_bias_res* is not a good option. The necessity of this calibration is clear: First, the best value that can be selected for the ensemble of neurons (e.g., *i_bias_res* = 750 nA in panel A) provides a significantly smaller reset current than an individual calibration by a factor of approximately two. Second, there is no meaningful common value for high reset voltages (panel D).

3.3.5 Synaptic efficacy

The circuit implementation offers several ways to influence the strength of synaptic events. The bias and source degeneration bias control the gain of the output OTA (fig. 3.12). The source degeneration bias is a shared parameter, but the normal bias could, in principle, be used to adjust the total synaptic strength for each synaptic input. However, a different approach is chosen here, that allows to calibrate the

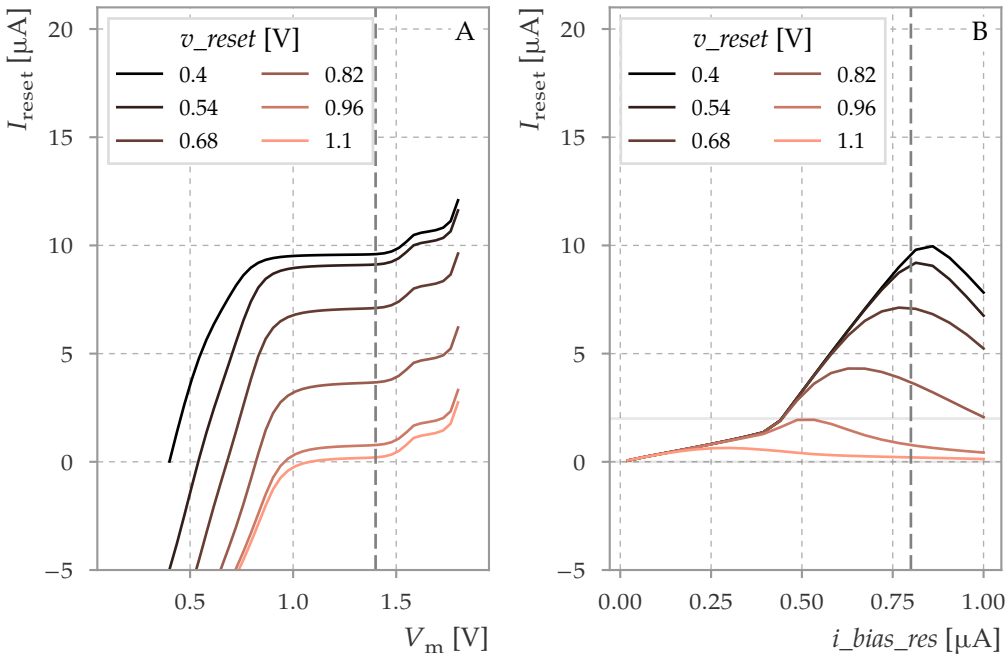


Figure 3.33: **A**: Reset current simulation in *tt* corner. The *high_s* switch (parameter *highs_res*) is set to active, which leads to high maximal currents on the order of 10 µA. The bias values are set to *i_bias_res* = 0.8 µA and *i_bias_res_sd* = 1 µA. A positive current I_{reset} pulls V_{m} to lower voltage values. The neuron parameter *v_reset* controls the location where the current intersects the zero-current line. Changing this parameter also changes the maximum current at which the OTA saturates (horizontal section between 1.0 V to 1.4 V). The saturation current is chosen as the reset current value at V_{m}=1.4 V (dashed line), even though V_{m} should stay below 1.2 V during normal operation. This selection of the rightmost part of the plateau is deliberate to prevent an estimate of the maximally possible reset current that is smaller than actually possible. **B**: Reset current at V_{m} = 1.4 V in dependence of the reset voltage *v_reset* and the OTA bias *i_bias_res*. The maximal strength of the reset depends on both of those parameters and the value of *i_bias_res* at which the current is maximal depends on *v_reset*. At a reset potential of *v_reset* = 0.96 V the current saturates at approximately 2 µA. Note that changing *i_bias_res* also changes the slope of the current at the zero-crossing, which is equivalent to the reset conductance (not shown). This data is also shown in Kriener (2017).

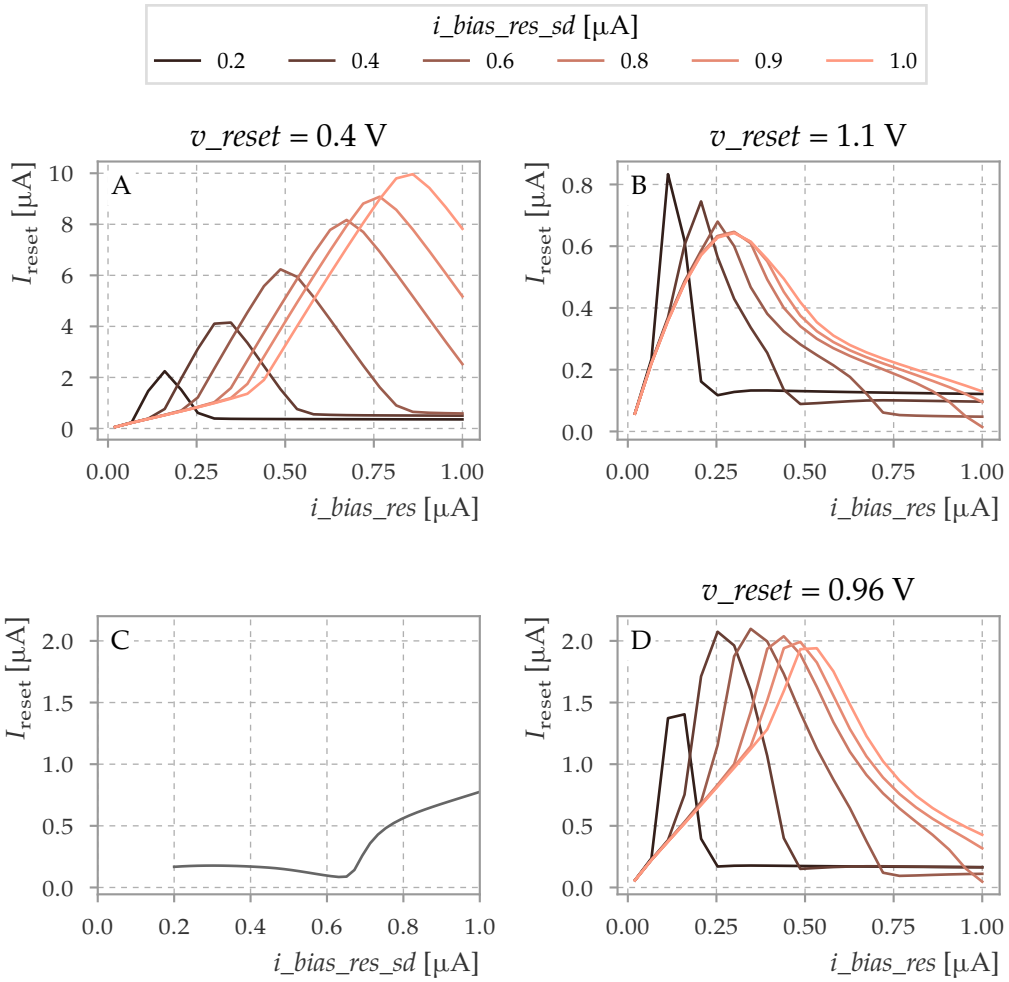


Figure 3.34: Maximum reset current in dependence of *i_bias_res_sd*. Lowering *i_bias_res_sd* greatly reduces the maximum reset current for small reset voltages (**A**) and increases the maximum reset current for higher reset voltages (**B**, **D**). **C**: The maximum reset current is not monotonic as a function of *i_bias_res_sd*, see also fig. A.3. An *i_bias_res* of 0.8 µA and a *v_reset* of 0.96 V is used. The rightmost curve in **A**, **B**, **D** corresponds to the simulation in fig. 3.33 B.

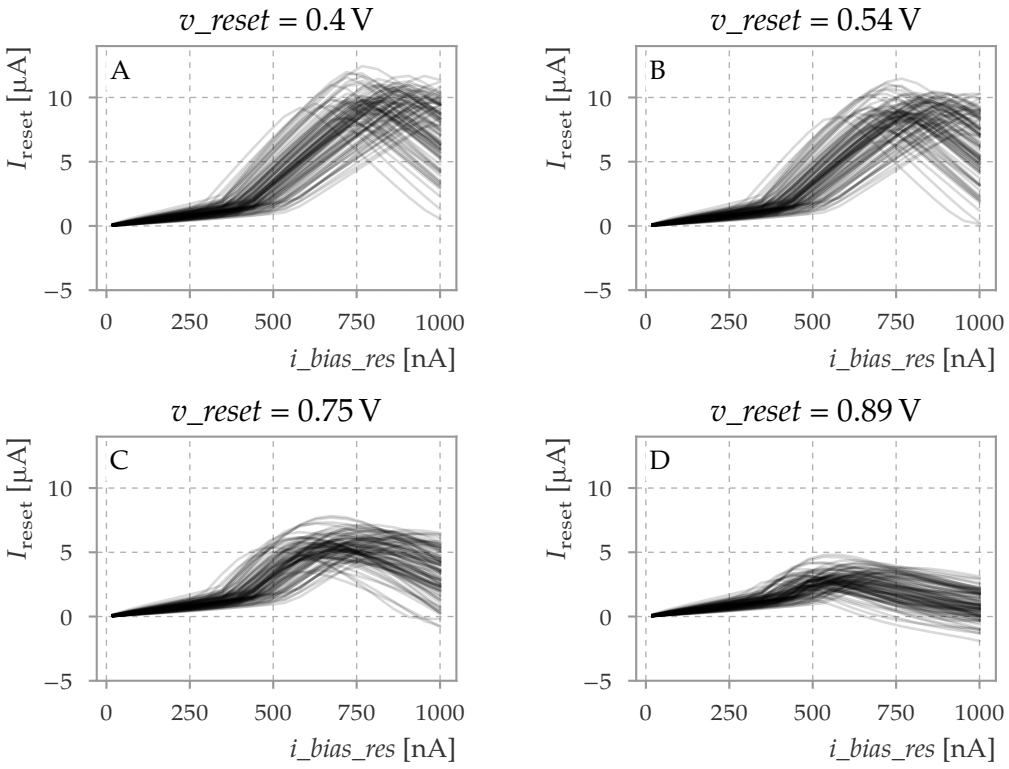


Figure 3.35: Maximum reset current in Monte-Carlo simulation. The maximal reset current is shown that is determined by the simulated calibration for 100 Monte-Carlo samples. The reset voltage is varied between 0.4 V (**A**) and 0.89 V (**D**).

synaptic efficacy while keeping the maximally possible synaptic effect reachable by changes of the digital weight only: The effect of synaptic transmission for the maximal digital weight setting is measured by calculating the integral of the post-synaptic membrane potential and post-synaptic current after spike stimulus:

$$I = \int V_\mathrm{m}(t)\mathrm{d}t \tag{3.9}$$

$$= \int \left(\frac{1}{C_\mathrm{m}}\Theta(t)\exp\left(-t/\tau_\mathrm{m}\right)\right) * \left(\Theta(t)\exp\left(-t/\tau_\mathrm{syn}\right)\cdot w\right)\mathrm{d}t \tag{3.10}$$

$$= \frac{w}{C_\mathrm{m}}\tau_\mathrm{syn}\tau_\mathrm{m} \tag{3.11}$$

The target values for the time constants are set to $\tau_\mathrm{m} = 10\,\mu\mathrm{s}$ and $\tau_\mathrm{syn} = 2\,\mu\mathrm{s}$ and the actually resulting time constants are extracted again from the simulation by exponential fits to the V_synint (after a spike stimulus) and V_m (after current input to the membrane). The additional evaluation of the time constant is added to increase the accuracy of the weight estimate. Otherwise, systematic errors from the synaptic and membrane time constant calibrations would add to the uncertainty of the calibration of synaptic efficacy.

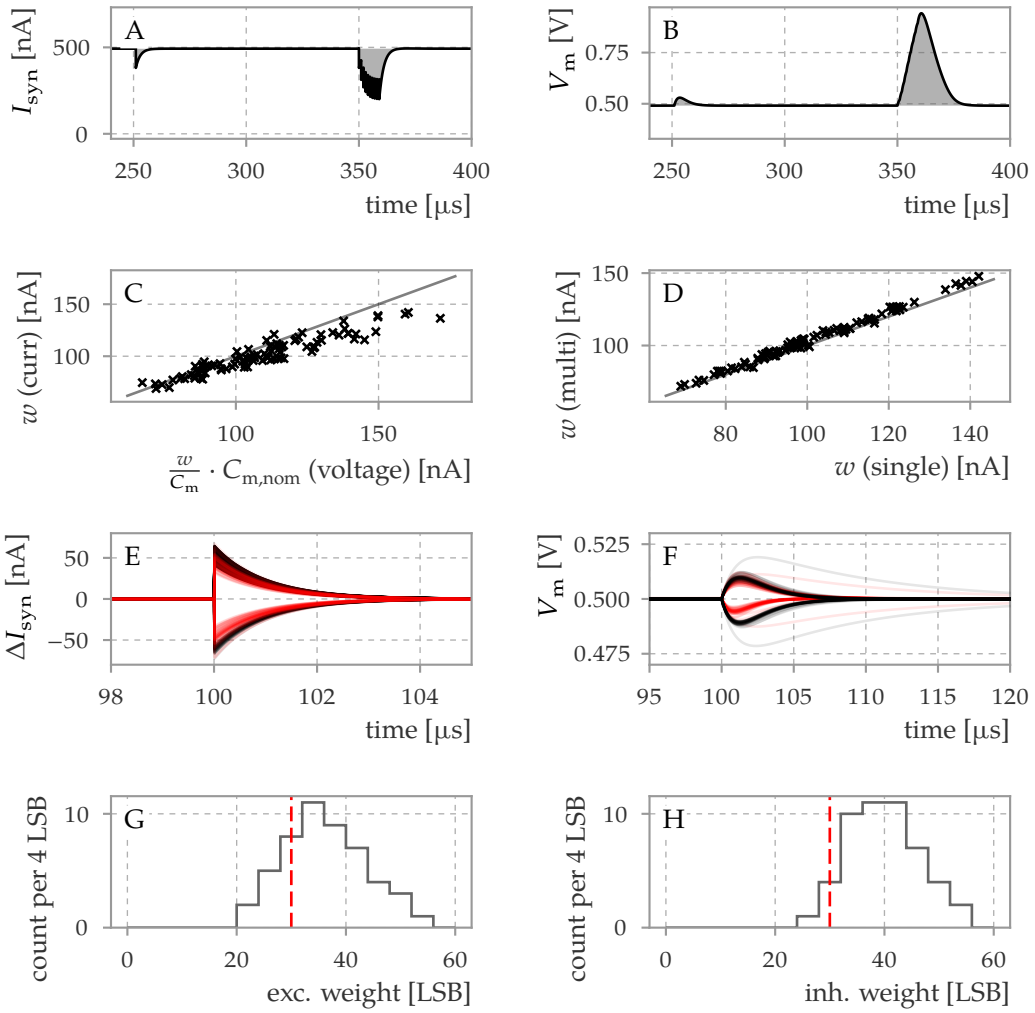


Figure 3.36: Calibration of synaptic weights. **A**: Example synaptic current trace. A single spike at 250 µs and a volley of ten spikes at 350 µs with one µs inter-spike distance is sent into the neuron. Panel **B** shows the membrane potential for the same simulation as A. The area of single and multiple stimulus is calculated for the post-synaptic current and membrane potential. **C** shows the comparison of the synaptic weight estimated by two different methods: Once using the membrane voltage trace (abscissa) and once using the synaptic current directly. The current-based method is used as the more precise reference to assess the precision of the voltage-based method. The data here and in the following figures is taken for 100 Monte-Carlo samples. **D** visualizes the comparison of the weight that is determined from the single and multiple stimulus post-synaptic current. **E** and **F** show an evaluation of the weight calibration, again for the post-synaptic current (E) and membrane potential (F). The post-synaptic current (E) is shown as a difference to the current without synaptic input $\Delta I_{syn} := I_{syn} - I_{syn,steady}$. The current of the excitatory synaptic input is shown on top. The absolute currents are shown additionally in fig. A.7. The red lines show the uncalibrated case with a constant digital weight for the synapse while the black lines use the calibration procedure described in the text. **G** and **H** show the distribution of digital weights that were used for the figures above; the dashed vertical line shows the value used for the uncalibrated simulation.

An interesting observation from eq. (3.11) is that not the synaptic weight but $^{w}/_{C_\mathrm{m}}$ is the basic quantity that can be calibrated by observing the membrane potential alone – the time constants can be determined from the shape of a PSP or from reference measurements.

To verify the result the weight is quantified by using the integral of the synaptic current directly

$$w_\mathrm{curr} = \int_{T_\mathrm{start}}^{T_\mathrm{end}} I_\mathrm{syn}(t) - I_\mathrm{syn,baseline}\mathrm{d}t \tag{3.12}$$

The resulting weight estimate is shown in fig. 3.36 C with a mean and standard deviation of $^{w}/_{w_\mathrm{curr}} = 1.08 \pm 0.07$. For comparison, the variation of the weight estimate using one or ten input spikes is negligible with $^{w_\mathrm{curr}}/_{w_\mathrm{multi}} = 1.030 \pm 0.017$ (fig. 3.36 D).

This calibration procedure reduces the variation of the post-synaptic current for a single PSP (fig. 3.36 E) by a factor of approximately two: from (41.8 ± 7.3) nA to (48.9 ± 4.2) nA for the excitatory and from (-42.8 ± 7.3) to (-55.8 ± 3.1) nA for the inhibitory case. The PSP that corresponds to the synaptic currents in fig. 3.36 E is shown in fig. 3.36 F. It is evident that the outliers stem from the leak term, the only other current term that is enabled in this simulation, because the synaptic current itself does not show such strong variation. The systematic mismatch between excitatory and inhibitory PSPs is compensated by the calibration procedure, which is also seen in the utilized synaptic weights (fig. 3.36 G and H) where the distribution is shifted to higher digital values for the inhibitory case.

The two last panels visualize how the dynamic range of the parameter is used for calibration: The sample standard deviation of the post-calibration synaptic weights is $\sigma = 7.8$ LSB and 6.6 LSB, for the excitatory and inhibitory weights, respectively. This variation reflects the measured variation in the PSP integral and requires half of the configuration range to compensate for mismatch.

The presented compensation could alternatively be achieved using the bias of the OTA in the synaptic input to homogenize the synaptic input independently of the weights. This alternative procedure would however limit the maximally available synaptic current to a lowest common value for all neurons. Using the synaptic weight as the tuning variable enables to keep the maximum synaptic current (which then varies from neuron to neuron) and allows further tuning – for example by dynamic plasticity – to utilize the maximum current that is provided by the substrate.

3.3.6 Spike threshold

Figure 3.37 shows the data generated for the calibration of the spike threshold. The membrane potential at which a spike is triggered is recorded for neurons which are set to fire continuously. Panel A and B show that the input offset of the spike comparator is only weakly dependent on *v_spike* above 0.6 V, while the total standard deviation of the recorded threshold at a common set value is 25 mV. The calibration procedure uses the data from simulation that is shown in panel A for a reverse lookup with a linear interpolation between the grid points. Because the dependency

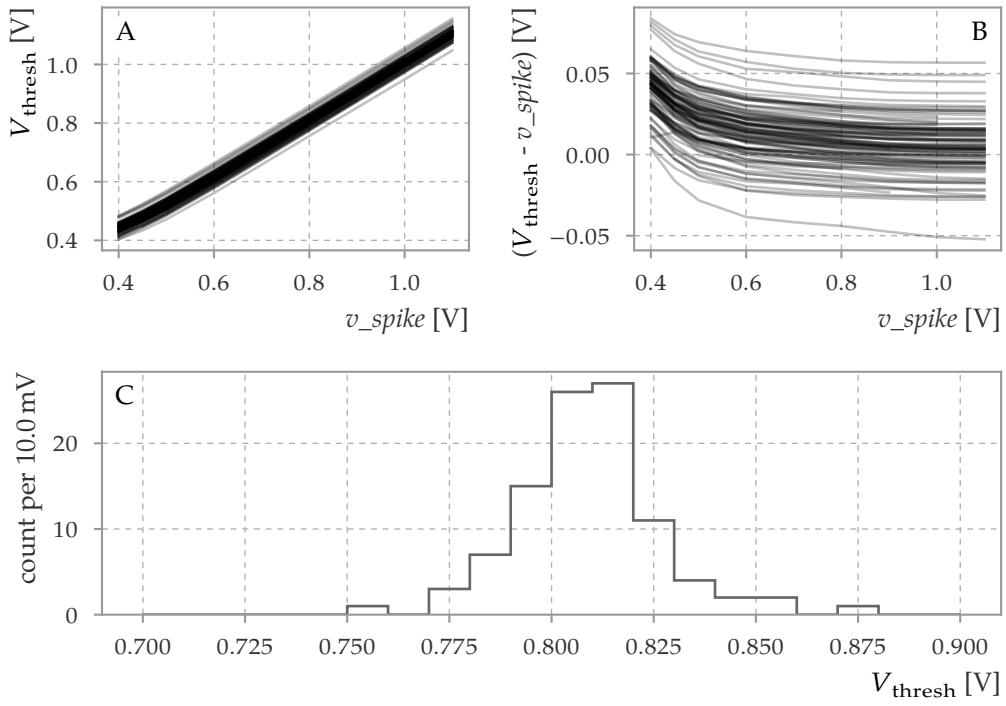


Figure 3.37: Spike threshold calibration **A**: Simulation results of the spike threshold in dependence of the spike threshold parameter for 100 Monte-Carlo samples. **B**: Difference of the recorded threshold voltage to the adjusted parameter *v_spike* (uncalibrated). Few curves stop before the maximal *v_spike* value, caused by the fact that the corresponding neurons did not spike in that setting. In this case the calibration extrapolates the recording at the highest valid *v_spike* (i.e., where the neuron did reach the threshold). **C**: Distribution of the recorded threshold voltage at *v_spike* $= 0.8\,\mathrm{V}$. The mean and standard deviation of the distribution are $(0.810 \pm 0.025)\,\mathrm{V}$.

of the offset on *v_spike* is only weak, the calibration procedure on the chip can be sped up by using only one measurement point, e.g. at 0.8 V, and use the difference of measured and desired threshold to compensate.

3.3.7 Adaptation

The adaptation term goes beyond the standard LIF model by adding a memory term to the neuron in form of an adaptation current. With this addition, the dynamics of the neuron can depend on previous activity that goes beyond the most recent spike of the membrane. The time constants of this adaptation current that are typically on the order of several hundred milliseconds and are longer than typical membrane time constants.

The hardware implementation (fig. 3.20) uses a capacitor, C_w, which slowly follows the membrane potential through an input transconductance. This mechanism

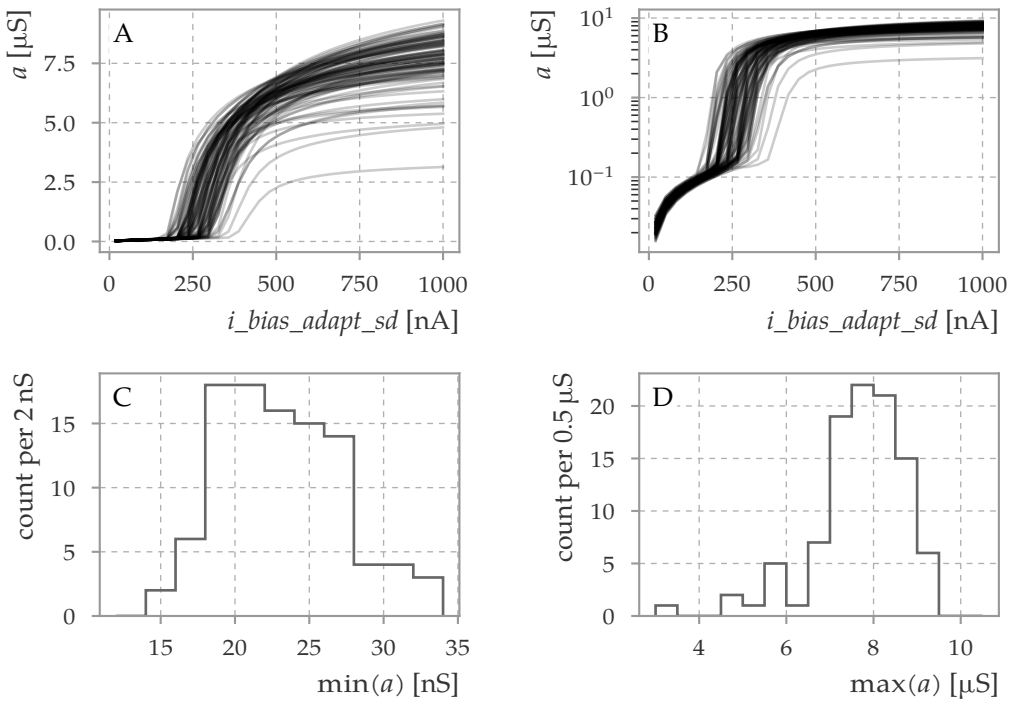


Figure 3.38: A, B: Dependency of adaptation parameter a on the OTA bias *i_bias_adapt_sd* for 100 Monte-Carlo samples. The OTA bias *i_bias_adapt* is a shared parameter. It is set to 1 µA in all cases. Linear and logarithmic scales are shown to visualize the low and high transconductance regions. The panels **C** and **D** show the distribution of minimal and maximal transconductances: min = (23 ± 4) nS, max = (7.7 ± 1.0) µS.

directly generates the time constant τ_w. The output current uses an output transconductance that is connected to C_w at one terminal and to a constant voltage at the other. The output transconductance is equal to the a parameter in the AdEx model (eq. (1.4)). The voltage on C_w is incremented at every spike time that is emitted by the neuron by a constant voltage step, implementing spike-triggered adaptation. The voltage step itself corresponds to b/a (see eq. (3.8)).

These three parameters – the strength of the two transconductances and the voltage increment – are the model-related parameters that must be calibrated. Additionally, the adaptation current has an offset which can be calibrated in multiple ways.

Adaptation coupling parameter *a*

The characteristic of the output OTA in the adaptation term is shown in fig. 3.38. The adaptation output circuit is a source-degenerated OTA, but uses a different implementation from the leak OTA (section 3.3.3). The source degeneration bias *i_bias_adapt_sd* is the parameter that is available per neuron and that is used to adjust the transconductance. The transition between high and low conductance regimes is more pronounced than in the leak OTA – the implementation details differ for those

two circuits. In particular, it uses the source degeneration implementation shown in panel B of fig. 3.19. The covered range of transconductance with $^{\max(a)}/_{\min(a)} > 300$ is large if both regimes are used. However, the transition is sharp, covering a change by a factor of more than 10 in a within approximately 50 nA of the bias parameter (see fig. 3.38 B). A lower precision is to be expected in this regime due to discretization of the analog parameter and error propagation in the case of noise. Applying the calibration uses the inverse of the data shown in fig. 3.38 A, using a linear interpolation with 33 nodes.

The calibration procedure on which fig. 3.38 is based uses a direct current recording from the adaptation term at two fixed values of the membrane voltage of 0.7 V to 0.8 V. This approach is technically possible on-chip when using the direct membrane connection ("iStim" in fig. 3.8), disabling all current terms of the neuron except for adaptation and using an external voltage source to clamp the membrane voltage. Additionally, it provides the best estimate for characterization purposes, i.e., it is free of effects that would be introduced by an indirect quantification. The method of measuring currents is used extensively in Stradmann (2016) to characterize and calibrate the DLS2 chip, which has comparable external connectivity. However, for a large-scale system, sequential current measurements are undesirable because external measurement hardware is required and measurement methods are preferred which use only one of the on-chip ADCs or an off-chip ADC that is part of the system, such as the *Flyspi* board (see (HBP SP9 partners, 2017, sec. I-7.1, "Flyspi FPGA PCB") and (Hartel, 2016, sec. 5.3)).

Voltage-based calibration method for subthreshold adaptation

It is possible to estimate the adaptation conductance a by measurements of the membrane voltage only. The method and its application to the DLS3 simulation is described in Kriener (2017):

A current pulse is applied with enabled and disabled adaptation term and the membrane voltage is recorded. The steady-state adaptation current w_{stat} is given by

$$\tau_{\text{w}} \frac{\mathrm{d}w}{\mathrm{d}t} = a \cdot (V_{\text{m}} - V_{\text{leak}}) - w \tag{3.13}$$

$$\Rightarrow w_{\text{stat}} = a \cdot (V_{\text{m}} - V_{\text{leak}}) \tag{3.14}$$

In equilibrium, the leak and adaptation currents cancel the stimulus current:

$$I_{\text{stim}} = -g_{\text{l}} \cdot \left(V_{\text{mem,a}} - V_{\text{leak}}\right) - w_{\text{stat}} \tag{3.15}$$

$$= (-g_{\text{l}} - a) \cdot \left(V_{\text{mem,a}} - V_{\text{leak}}\right) \tag{3.16}$$

$$= (-g_{\text{l}} - a) \cdot \Delta_a \quad , \tag{3.17}$$

where we call the membrane and adaptation voltage $V_{\text{mem,a}}$ and $V_{\text{w,a}}$ in the case where adaptation is switched on. Δ_a denotes the difference in the steady state of V_{m} with and without applied stimulus current. When adaptation is disabled, the offset

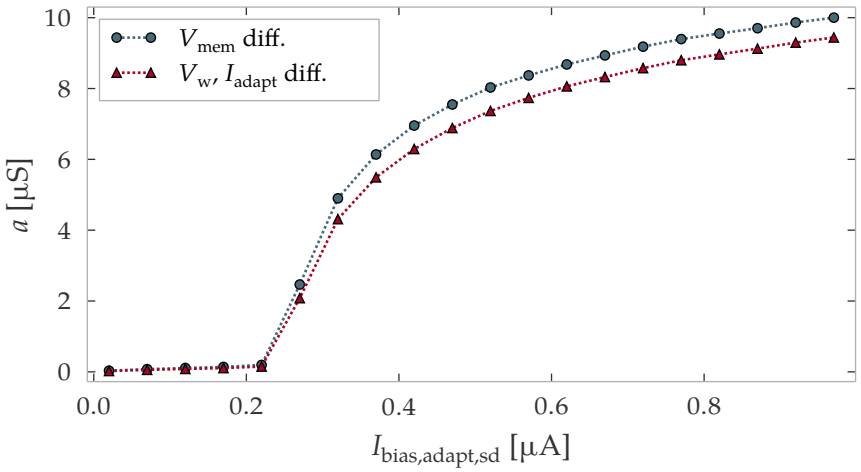


Figure 3.39: Comparison of the current and the voltage-based methods to determine a. *i_bias_adapt* is set to 1 µA. Adapted from Kriener (2017).

current is compensated by leakage only:

$$I_{\text{stim}} = -g_l \cdot (V_{\text{mem}} - V_{\text{leak}}) \tag{3.18}$$

$$= -g_l \cdot \Delta \quad , \tag{3.19}$$

where Δ is the difference in the steady state V_m with disabled adaptation. Assuming the stimulus current is equal in both experiments one can solve for a:

$$a = g_l \cdot \left(\frac{\Delta}{\Delta_a} - 1 \right) \quad . \tag{3.20}$$

The method works only if a and g_l are of comparable magnitude. Otherwise, Δ/Δ_a approaches one for $a \ll g_l$ or infinity for $a \gg g_l$ which incurs a large error on the estimate of a. On a fast hardware substrate it would also be conceivable to improve the precision by modifying g_l in a closed-loop fashion to tune for a constant ratio of, for example $\Delta/\Delta_a = 2$. Then $a = g_l$ and the leak conductance can be re-measured precisely using the membrane time constant. The method depends on two assumptions which must be ensured on the hardware device: First, the leak and adaptation OTAs are linear in the range that is used for the measurement. Second, the offset current must be equal in the two successive measurements.

Figure 3.39 shows the difference between the current- and the voltage-based calibration method for one exemplary sample. The systematic mismatch between the two methods is as large as 10 % which is attributed to the nonlinearity of the leak and adaptation terms. The two methods are matched sufficiently well to discern the transition between the high and low conductance regimes of the source degeneration feature, which is an important quantity for calibration as it varies strongly due to mismatch (fig. 3.38 B).

The adaptation OTA is derived from the leak OTA of the DLS2 prototype. One of the improvements in the leak OTA on DLS3 is the reduced sensitivity to process

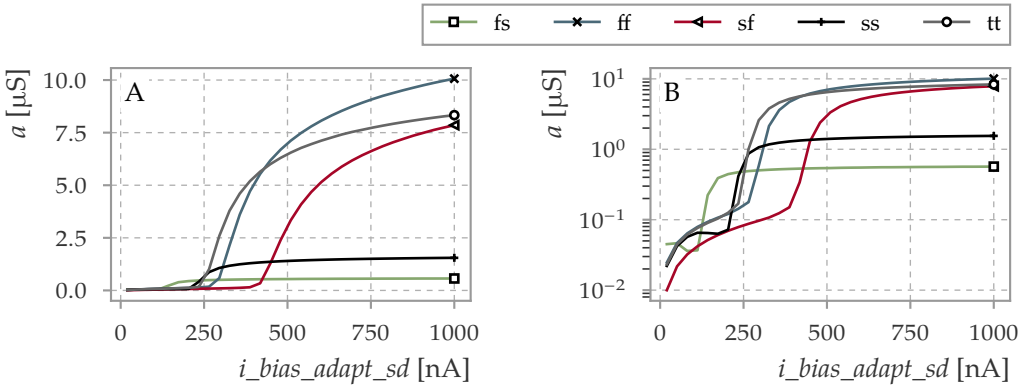


Figure 3.40: Dependency of the adaptation parameter *a* on the process corner. The transconductance of the adaptation output OTA is shown as a function of the source degeneration bias *i_bias_adapt_sd* in linear (**A**) and logarithmic (**B**) scale.

corners: Figure 3.40 shows the corner dependency of the adaptation output OTA. The maximum transconductance and the bias current at which the switch to the high conductance mode happens vary with the corner, the slow PMOS corners *fs* and *ss* being the least favorable. Additionally, these corners show non-monotonic dependence of *a* on the bias current. The difference in maximum transconductance is not harmful in this context because typically, low values of *a* are required. The early switch and the non-monotonicity, however, make the calibration procedure less robust and less precise. Figure 3.41 shows the same quantity for the leak/reset

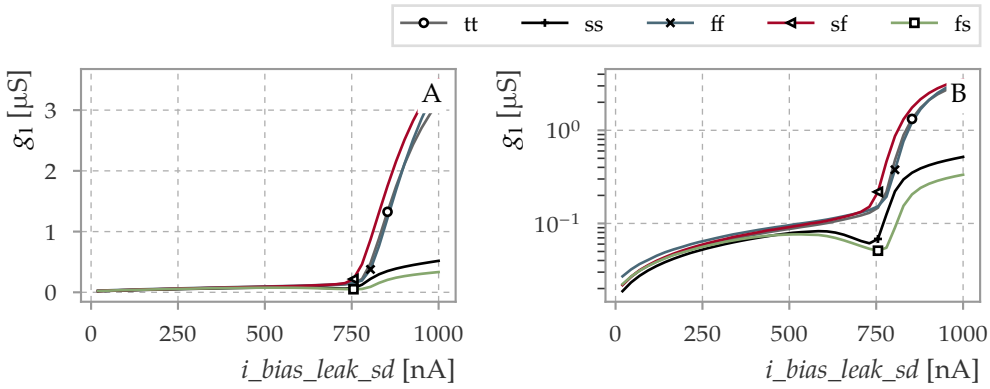


Figure 3.41: Corner dependency of the DLS3 leak and reset OTA. Data are calculated from the membrane time constant calibration (exactly the same simulation method that is used for fig. 3.31). The conductance is calculated using the nominal membrane capacitance, $g_1 = {}^{C_m}/_{\tau_m}$. **A** and **B** show the same data on linear and logarithmic scale.

OTA. Again, the behavior in the *ss* and *fs* corners shows non-monotonicity and lower maximal transconductance, but the rapid increase of transconductance happens

at approximately the same point of 750 nA, leaving significantly more room for parameter adjustment.

Adaptation time constant τ_w

Figure 3.42 shows the calibration results for the adaptation time constant. A voltage step from 0.8 V to 0.7 V is generated on the membrane and the decay of V_w is fit by an exponential function. The resistance of the adaptation resistor is not constant, which leads to some deviation from a perfect exponential function (see fig. 3.43). In simulation, V_w is used directly and on-chip it is accessible using the read-out switch in the output multiplexer (fig. 3.8, fig. 3.13).

The resulting time constants cover a range of 14.4 µs to 336 µs. The standard deviation due to mismatch that is to be expected when using an uncalibrated device is 20 % for the lowest achievable time constants and approaches 35 % to 40 % for τ_w above 100 µs (fig. 3.42 F), which is the range used in most reference models table 3.4.

It was discovered that the dependency on the process corner in the adaptation time constant is very strong for asymmetric corners (*sf*, *fs*): The resulting time constants deviate by a factor of approximately of ten from the typical case (fig. 3.44). The reason is that the voltage bias for the resistance transistors M_1 and M_2 (PMOS) in fig. 3.24 is generated by NMOS device M_3, so opposing changes in mobility cause the highest deviation from the typical case.

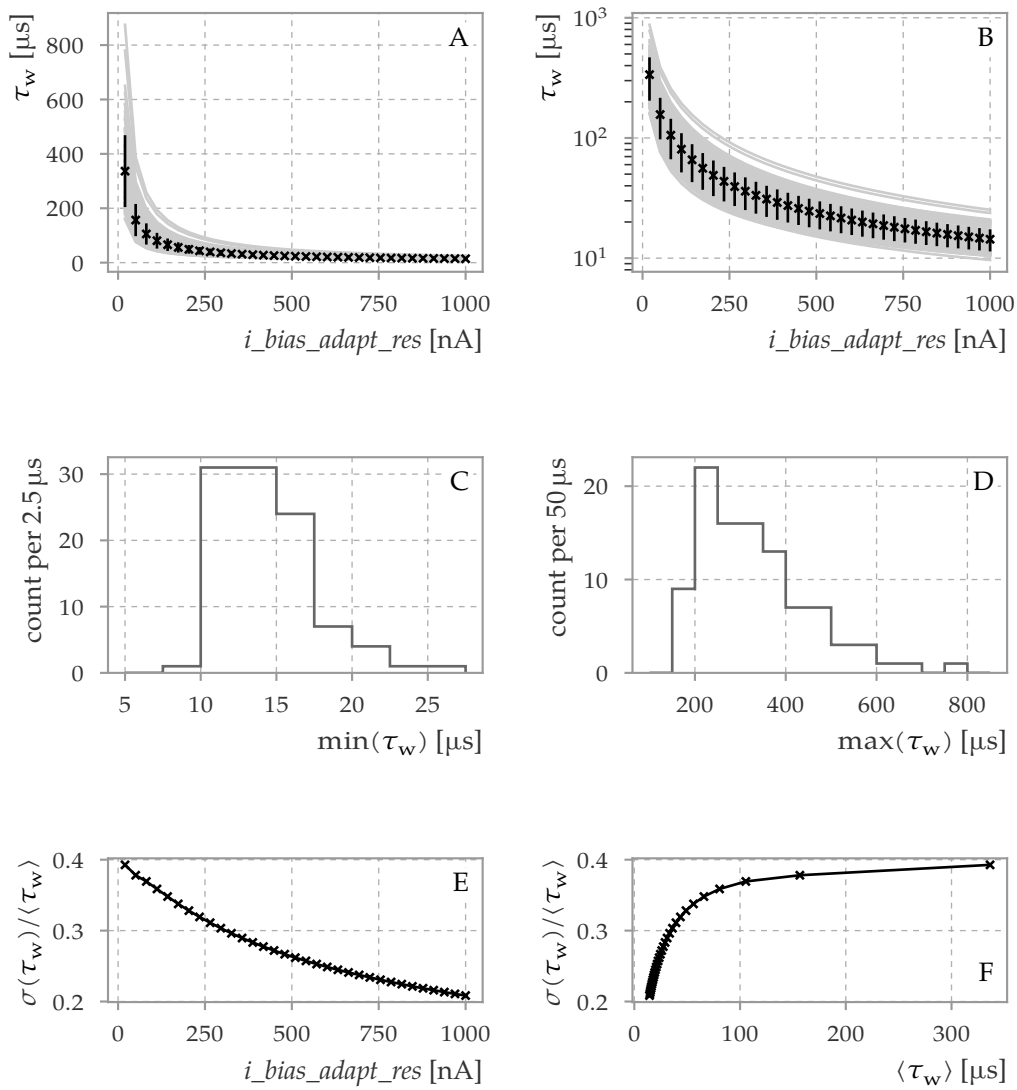


Figure 3.42: Calibration of the adaptation time constant. **A**, **B**: Dependency of the adaptation time constant on the bias parameter *i_bias_adapt_res*, shown on linear and logarithmic scale. The mean and standard deviation of the simulation result for 100 Monte-Carlo samples is displayed by black crosses and bars. The data for each individual sample is shown in gray. **C** shows the distribution of the minimal, **D** of the maximal τ_w for each sample, min $= (14.4 \pm 3.0)$ µs and max $= (336 \pm 130)$ µs. (Because of the strongly asymmetrical shapes of the distribution the median values are given as well: 14 µs and 315 µs.) At least 95 % of the samples have a minimum time constant smaller than 20.1 µs and at least 95 % have a maximal time constant greater than 187 µs. **E** and **F** show the coefficient of variation for the data as a function of the control parameter *i_bias_adapt_res* and of the mean time constant. This quantifies the variance that is expected when using an uncalibrated parameter.

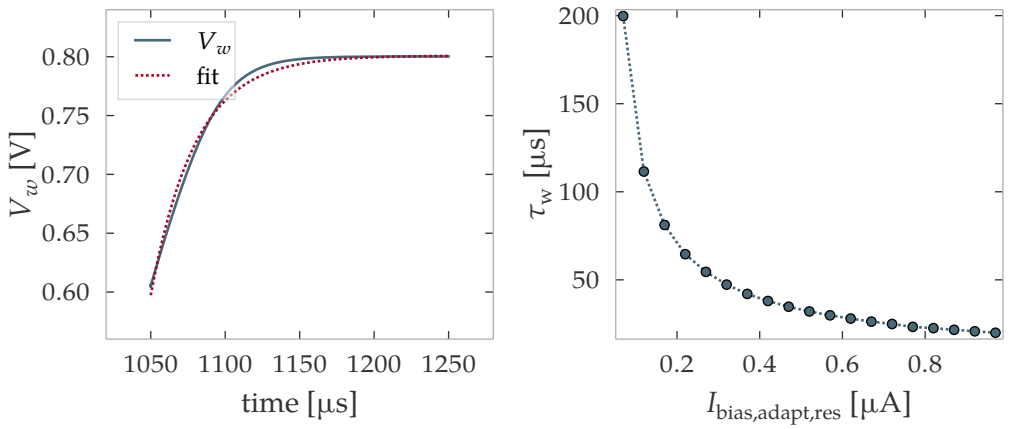


Figure 3.43: Left: exponential fit to V_w after a voltage step in V_m. On the right, a single result corresponding to fig. 3.42 B is shown. Reproduced from Kriener (2017).

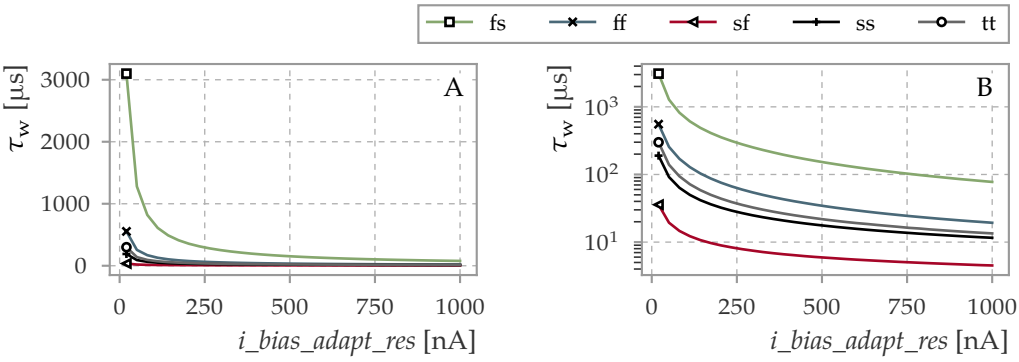


Figure 3.44: Dependency of the adaptation time constant τ_w on the process corner, shown in linear (**A**) and logarithmic (**B**) scale.

Spike-triggered adaptation b

To calibrate the parameter b an ideal method is used. Spike-triggered adaptation is generated by switching a constant current for a defined period of time onto the adaptation capacitance. The duration for which the adaptation current is switched onto C_w (fig. 3.22) is generated in the digital neuron back end and can be assumed significantly more precise than analog effects. The current is used directly or mirrored once if a negative adaptation is used.

Assuming the current magnitude to be close to the nominal current value, one has to select an appropriate adaptation pulse duration. The required change in voltage V_w is

$$\Delta_{V_w} = \frac{b}{a} = \frac{T_{\text{target}} \cdot i\_adapt\_w}{C_w} \tag{3.21}$$

Because the adaptation pulse duration must be an integer, it is selected using a maximal target adaptation current, which is empirically chosen to be $I_{\max} = 0.8\,\mu\text{A}$.

$$T_{\text{target}} = \frac{|b|\, C_w}{|a|\, I_{\max}} \tag{3.22}$$

$$adaptation\_time = \lceil T_{\text{target}} \nu_{\text{refrac}} \rceil \tag{3.23}$$

It is ensured that *adaptation_time* does not exceed the maximal available counter length. *i_adapt_w* is then adjusted to the rounded adaptation pulse duration.

Application of calibration

Figure 3.45 shows a comparison of uncalibrated (A), calibrated except for adaptation parameters (B) and fully calibrated Monte-Carlo samples (C). The neuron is stimulated by a regular burst of spikes for 50 µs. "Uncalibrated" means in this case that the calibration data from the typical process corner simulation is used. The calibration of adaptation significantly reduces the variance of the different neurons responses (panel D).

In fig. 3.46 the currents that contribute to the evolution of V_{m} are shown to clarify the source of the remaining variation. It can be seen that the input current caused by synaptic activity varies significantly: 80 nA minimum-to-maximum difference for a synaptic current of 150 nA (panel B). The change in adaptation and leak current vary up to 50 % of the mean maximal change (panels C and D) The simulation demonstrates that the adaptation calibration is not the sole source of the remaining variation and the relative amount of variance is comparable to that of the input.

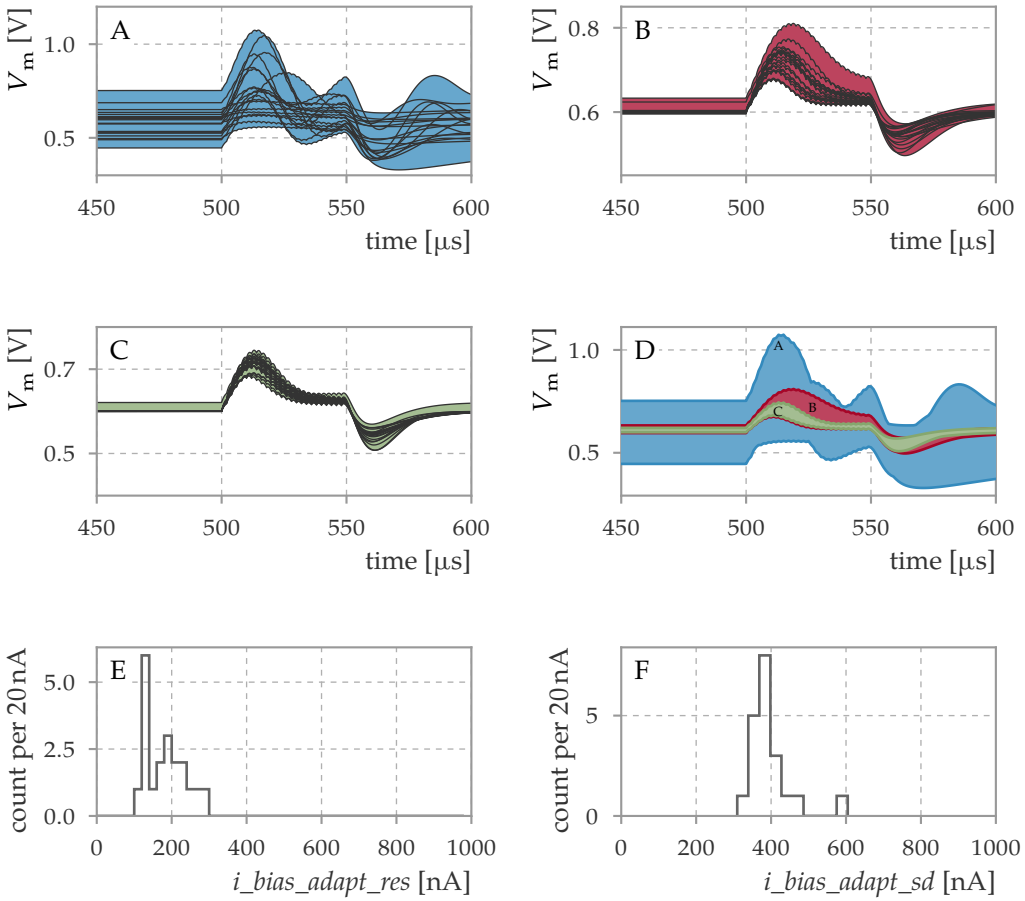


Figure 3.45: **A**: Membrane potential for 20 Monte-Carlo samples with identical current and voltage parameters, which are obtained from the *tt* calibration. The neurons are stimulated by synaptic excitatory stimulus in the range of 500 µs to 550 µs with an inter-spike interval of 2 µs. The range between the minimal and maximal membrane voltage is filled by a solid color for better comparison. **B**: Simulation as in A but using individual calibration except for the adaptation-related parameters *i_bias_adapt_res*, *i_bias_adapt*, *i_bias_adapt_sd* and *_adapt_w*. **C**: Simulation as in A but using individual calibration for all parameters, including adaptation. (Please note the different scales in A, B and C.) **D**: Comparison of the extent of the variation in panels A, B and C. **E**: Distribution of the parameter *i_bias_adapt_res* for the simulated calibration (C). **F**: Distribution of the parameter *i_bias_adapt_sd* for the simulated calibration (C).

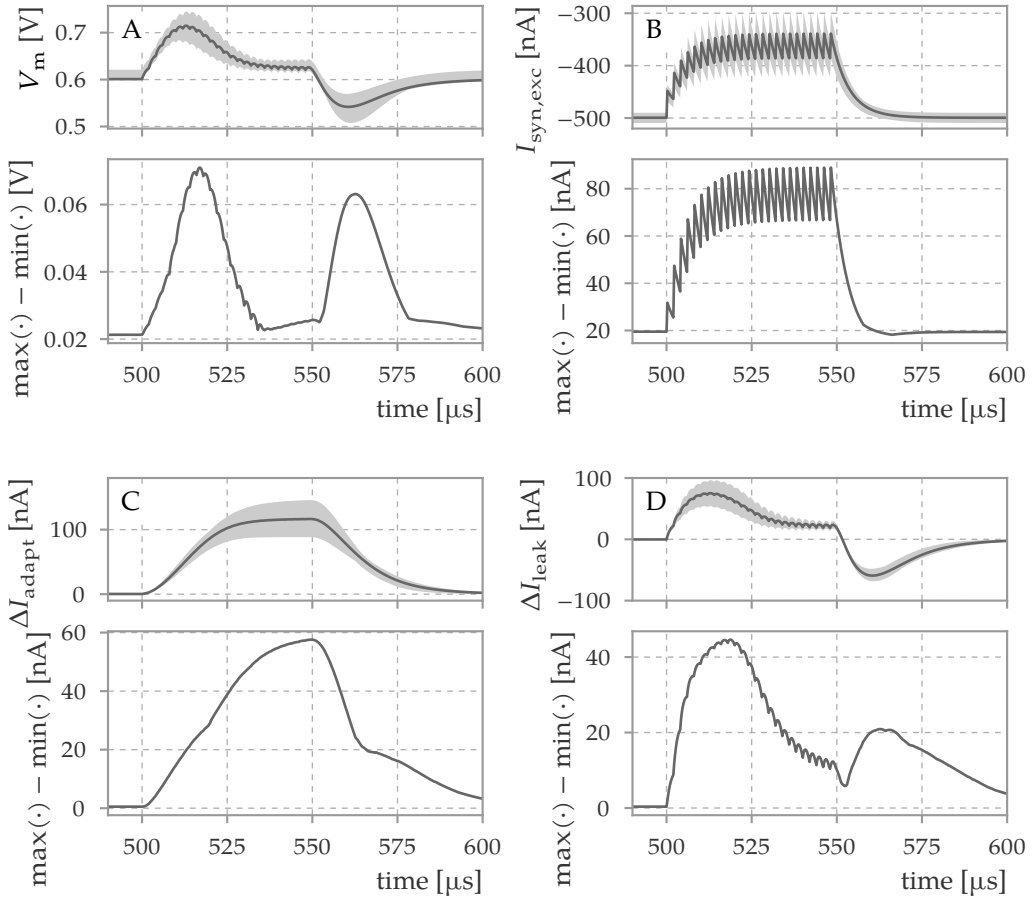


Figure 3.46: Detail of currents for the calibrated adaptation. In each panel, the upper graph shows the mean (dark line) and extent (minimum to maximum) of the corresponding curve for all Monte-Carlo samples. The lower graph shows the difference of the maximum and minimum of the same data to quantify the spread. **A**: Membrane voltage with enabled and calibrated adaptation. Simulation for 20 Monte-Carlo samples. The same data as in fig. 3.45C is shown. **B**: Excitatory synaptic input current. **C**: Deviation of the adaptation current I_{adapt} from its resting state. **D**: Deviation of the leak current I_{leak} from its resting state. The non-shifted value of I_{adapt} and I_{leak} is shown in fig. A.4.

3.3.8 Exponential term

The exponential term is a special case in on the test chip in so far as it does not have analog parameters adjusting the model parameters Δ_T and V_T. These parameters are planned for a future implementation, but a circuit was included to test the generation of the exponential current and already include a nonlinear membrane voltage feedback term which is required for several firing patterns that are characteristic for the AdEx neuron model (Naud et al. (2008)). The exponential current can instead be tuned by a three bit DAC that multiplies the exponential current, which is equivalent to setting discrete values of V_T.

Figure 3.47 shows the properties of the circuit. Panel A shows the current as a function of the membrane potential for different settings of the DAC parameter *exp_weight_b*. The circuits current output is zero at $V_\mathrm{m} = 1.2\,\mathrm{V}$ because the circuit is implemented with thin-oxide devices using 1.2 V supply only. This is the desired behavior, because the membrane potential should be limited to 1.2 V in any case due to other connected thin-oxide-based circuits.

Panel B shows the corner dependency of the exponential current, which varies by maximally a factor of two between the lowest and highest currents. The source of the variation is broken down in fig. 3.48: The input voltage is converted into an inverted voltage called V_ramp (fig. 3.48 C, fig. 3.23), at which point the voltage levels are clearly separated in the different corners. The transistor M_3 converts the voltage into an exponential current, re-grouping the different corners. This current is mirrored twice, in M_4 – M_5 and in the output DAC; the process corner does not affect the relative current ratios significantly in this step (fig. 3.48 D, F).

Figure 3.47 C and E show the effect of mismatch on the exponential current. A relative variation of approximately 35 % is expected in the uncalibrated case. Figure 3.47 D shows the effect of mismatch on the individual paths of the current multiplier. The distribution of ratios of the 2× and 4× bits has relative standard deviation of 18 % and 24 %. That shows that mismatch effects are also present in the output current mirrors – a significant proportion of the 4× current mirror deviates by more than one LSB from its expected value. The slope factor Δ_T – which characterizes the speed of the exponential rise with V_m, and is determined purely by the circuit characteristics – varies comparatively little with mismatch ($\sigma = 5\,\%$, fig. 3.47 F.)

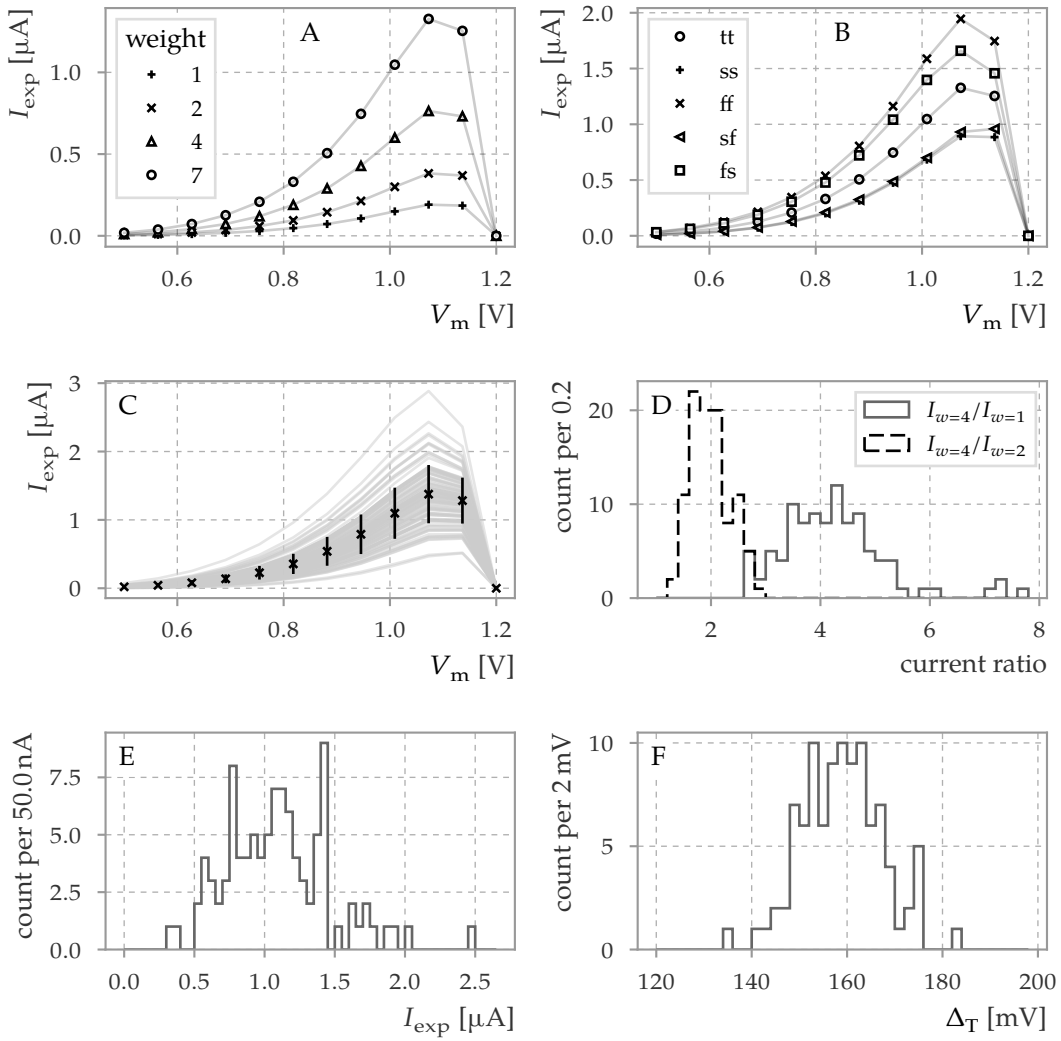


Figure 3.47: Properties of the exponential circuit. **A**: The exponential current as a function of V_m. The current with only one (1, 2, 4) and all three output bits enabled. The values are obtained from the exponential term at clamped membrane voltage in a simulation in the *tt* corner. **B**: Exponential current simulated for different process corners with a weight of 7. The exponential current is higher in the fast and lower in slow NMOS corners. **C**: Mismatch simulation for 100 Monte-Carlo samples. The error bars show one standard deviation. The individual simulations are shown in gray to provide a view of the outliers. **D**: Mismatch of the weight bits for 100 Monte-Carlo simulations. The ratio of the current at $V_m = 1.01\,V$ for the most significant bit and the two other bits is shown. The mean and one sigma deviation of the ratios are 4.2 ± 1.0 and 1.98 ± 0.35. **E**: Histogram of the current at $V_m = 1.01\,V$ for 100 Monte-Carlo samples (same data as in C). The mean and standard deviation is $(1.1 \pm 0.4)\,\mu A$. **F**: Histogram of Δ_T which is obtained from the data shown in C as the parameter of an exponential fit in the range of 0.5 V to 1.07 V. The mean and standard deviation is $\Delta_T = (159 \pm 8)\,mV$.

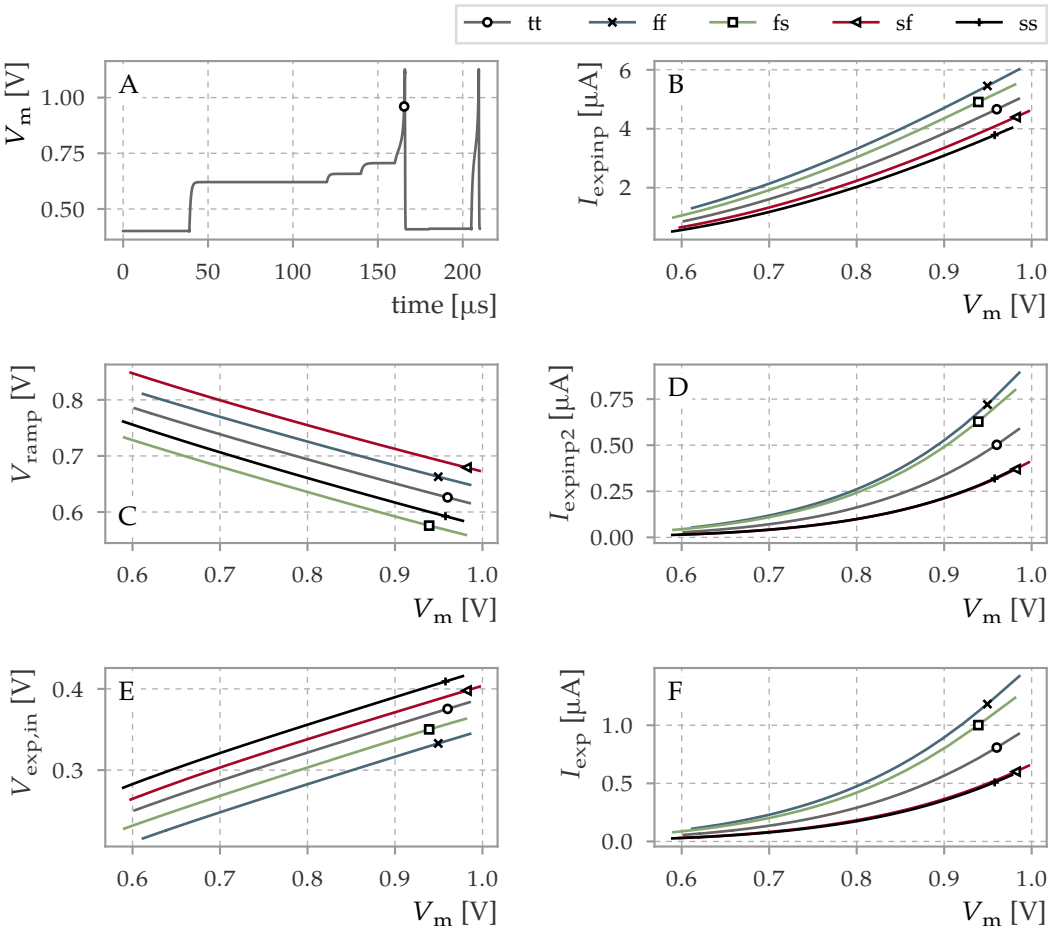


Figure 3.48: Distribution of sources of corner dependencies for the exponential term. See fig. 3.23 for the voltage and current identifiers. **A**: Exemplary membrane trace that is used to simulate the exponential term. A series of increasing current steps stimulate the membrane. A long refractory period is set to isolate a single exponential peak. **B**: Drain current of the NMOS device M_1 as a function of the membrane potential. The voltage V_{ramp} (**C**) splits due to the corner dependency of M_1 and M_2. **D**: Drain current of the PMOS device M_3, which operates in the subthreshold regime and converts its gate voltage V_{ramp} to the current $I_{expinp2}$. This current is then mirrored twice, using a three-bit digitally controlled second stage. The voltage at the gates of the first mirror is shown for completeness in **E**. **F** shows the output current with all three output transistors enabled.

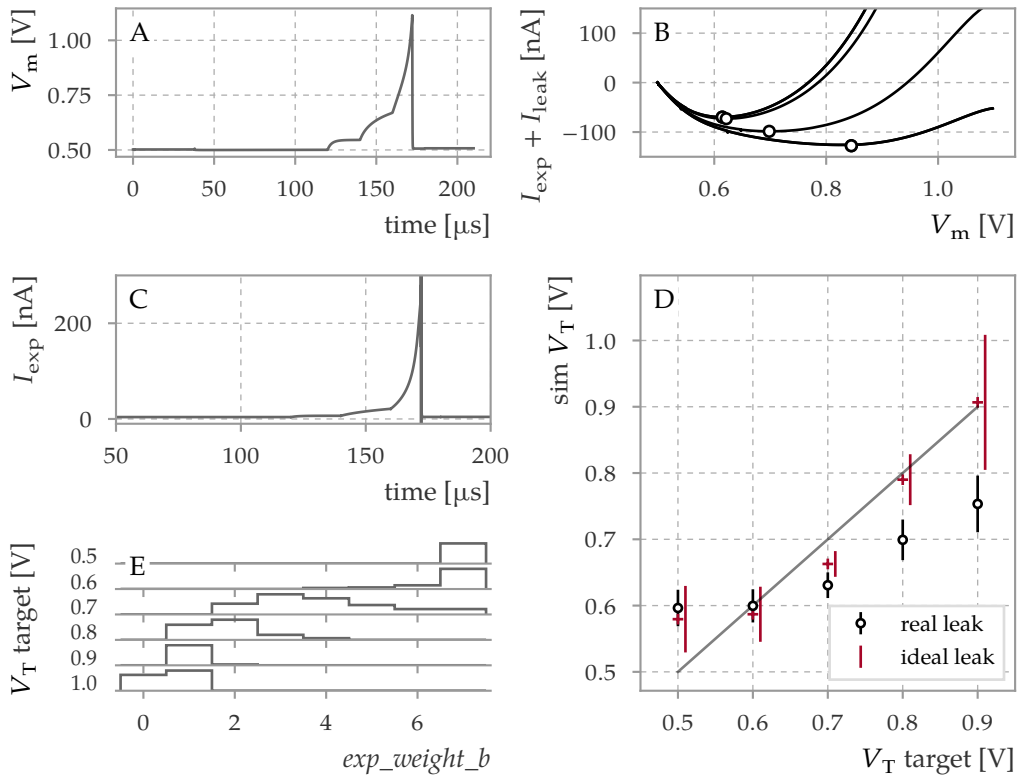


Figure 3.49: Calibration of the exponential term. **A**: Single simulation. The membrane is stimulated with increasing current steps until the neuron fires. The rapid rise of the membrane voltage due to the exponential current is visible after 160 µs. An example for a random Monte-Carlo sample is shown. **B**: The sum of exponential and leak currents is shown for four settings of the exponential current strength, *exp_weight_b*. The offset current of the resting membrane is subtracted in all samples. The minimum of each curve is marked by "∘" **C**: The exponential current that corresponds to the simulation in A is shown. **D**: Mean and standard deviation of min $\left(I_{exp} + I_{leak}\right)$ for 50 Monte-Carlo samples. The leak current for the data denoted by "real leak" is obtained from the transistor-level simulation. "ideal leak" uses the transistor-level data for I_{exp} but an ideal approximation $-g_l \cdot (V_m - V_{leak})$ for the leak current. For g_l, the target value is used. The identity line is shown in gray for convenience, and the error bars for the ideal case are shifted for better visual separation of the data sets. **E**: Histogram of the exponential weight after calibration for the given target values of the exponential threshold, V_T. A value of zero corresponds to a disabled exponential term.

In fig. 3.49, the calibration of the exponential term is shown. The calibration procedure is chosen as follows: The exponential current for each DAC bit is recorded individually for twelve steps between 0.5 V to 1.2 V. The current at the target threshold voltage V_T, $I_\mathrm{target} = g_\mathrm{l}\Delta_\mathrm{T}$ is calculated, and the DAC bit combination that minimizes the exponential current at V_T to the target current is selected:

$$W_\mathrm{exp} = \mathrm{argmin}_{b_1,\ldots,b_3}\left(\left|\sum_{i=1}^{3} b_i \cdot I_\mathrm{exp}(V_\mathrm{T}) - I_\mathrm{target}\right|\right) \tag{3.24}$$

Figure 3.49 C shows the verification for this procedure: The minimum of the voltage-dependent currents (leak plus exponential) corresponds to V_T in the ideal model. The evaluation of the calibration is shown in fig. 3.49 D. The minimum of the voltage-dependent currents (circles) is systematically lower than the target V_T (diagonal line). However, this is caused not primarily by an incorrect exponential calibration but by the voltage-dependent leak conductance, as evidenced by the calculated minimum with an ideal (i.e., linear) leak current (crosses). The parameter space for calibration is quite limited by only seven possible non-zero settings. In the example in fig. 3.49 E the available range is not saturated only between the V_T target of 0.7 V to 0.9 V.

3.4 Pre-production circuit verification

In addition to the implicit verification of the circuits that emerges as a byproduct of the implementation of the Monte-Carlo calibration (section 3.3), an explicit verification of basic and essential functionality was performed shortly before tape-out of the chip. Focus was put on parts of the circuit that are absolutely necessary for the neuron to function. Especially the functionality of digital configuration was simulated.

In this section the verification of the most important features is described first. Then, the discovered errors and, where applicable, the improved version is described in detail.

3.4.1 Test of digital configuration

The static random-access memory (SRAM) that stores the digital configuration of the neuron is implemented as a full-custom part of the neuron circuit. The following verification tests that the digital parameters do have their desired effect in simulation. It is intended to prevent mistakes like incorrectly wired signals, which may occur because digital control signals are occasionally level-shifted, passed from one module to another (the neuron module provides configuration space for the leak/reset and multi-compartment circuits) or using both of the complementary SRAM voltages to control complementary parts of some circuit. What is explicitly not tested is the writing of the SRAM – the stored state is assumed to be correct at the beginning of the simulation (see fig. 3.5). It is particularly important that possible current sources can be reliably disabled, because otherwise, sub-circuits of the neuron can not be easily characterized in isolation.

In the following, a list of the most important configuration parameters that were verified is shown:

1. *Enabling and disabling the exponential term:* It was verified that no significant exponential current flows when *en_exp* = `False` and that the expected exponential current flows when *en_exp* = `True`.

2. *Function of exponential weight:* It was verified that *exp_weight_b* scales the exponential current as expected.

3. *Adaptation term:* When *en_adapt* is set to `False`, no significant adaptation current flows onto the membrane from the adaptation term.

4. *Adaptation capacitor merging:* When *cap_merge* is set to `False`, no significant current flows onto the membrane from the adaptation term.

5. *Spike detection:* The spike comparator is disabled when *en_spk_cmp_b* is high and enabled when it is low.

6. *Switching of synaptic input en_syn_i_exc* and *en_syn_i_inh* can be used to enable each of the synaptic input circuits. Note that *v_syn_exc* and *v_syn_inh* and must

be set correctly for fully disabled synaptic input, see fig. 3.29. These parameters must be set to produce a negative output current to prevent leakage through the thin-oxide transmission gate.

7. *External current input:* The external current can be directed to the neuron circuit for which *en_ana_in* is enabled.

8. *Read-out multiplexer:* The possibility of recording for all available signals – membrane, adaptation and synaptic input voltages – was verified. Figure 3.50 shows the readout of the different signals. Panels A and B demonstrate the additional thick-oxide switch that was included for read access to the adaptation voltage. A single synaptic event on each of the synaptic input lines is read out using the source follower at a shifted voltage (panels C and D).

9. *Read-out towards membrane ADC:* The simulation shown in Figure 3.50 was conducted using a 500 fF load capacitance. The read-out multiplexer is switched through all available input signals. The dashed line shows the output voltage of the read-out amplifier. The synaptic input lines (C and D) are shifted due to the source follower. The adaptation voltage can only be read out when *en_read_vw* is enabled (A and B).

10. *Read-out towards correlation ADC:* The correlation ADC provides a means of parallel voltage readout from synaptic correlation measurements in each synapse column. For this purpose, the neuron read-out amplifier output is additionally connected to the causal and acausal read-out lines, separated by a switch in the synapse array (`a[5]` and `w[5]`, (Hartel et al., 2017, 12.2.12, synapse memory raw bit ordering)). The read-out for two of four neuron compartments is shown in fig. 3.51. The switch within the synapse array as well as the readout line is simulated using a 50 fF capacitance, a 3 µA biasing current produced by a current mirror and 32 disabled synapse correlation outputs to provide a realistic load for the output amplifier. The results are shown in fig. 3.51[22] The signal only arrives at the target synapse line when the amplifier is enabled.

11. *Membrane capacitance switching: en_mem_cap* can scale the membrane time constant (Figure 3.52). Adding the adaptation capacitance to the membrane capacitance (*en_cap_merge*) also has the desired effect of increasing the membrane time constant.

12. *Membrane capacitance merge:* The adaptation capacitance can be used as additional membrane capacitance if adaptation is not used (Figure 3.52).

13. *en_right:* Neighboring membranes can be connected and disconnected using *en_right*.

14. *en_scon:* The soma connection is established when *en_scon* is enabled; no current flows when *en_scon* is disabled.

[22]The simulation setup for fig. 3.51 fig. 3.50 was assisted by Korbinian Schreiber.

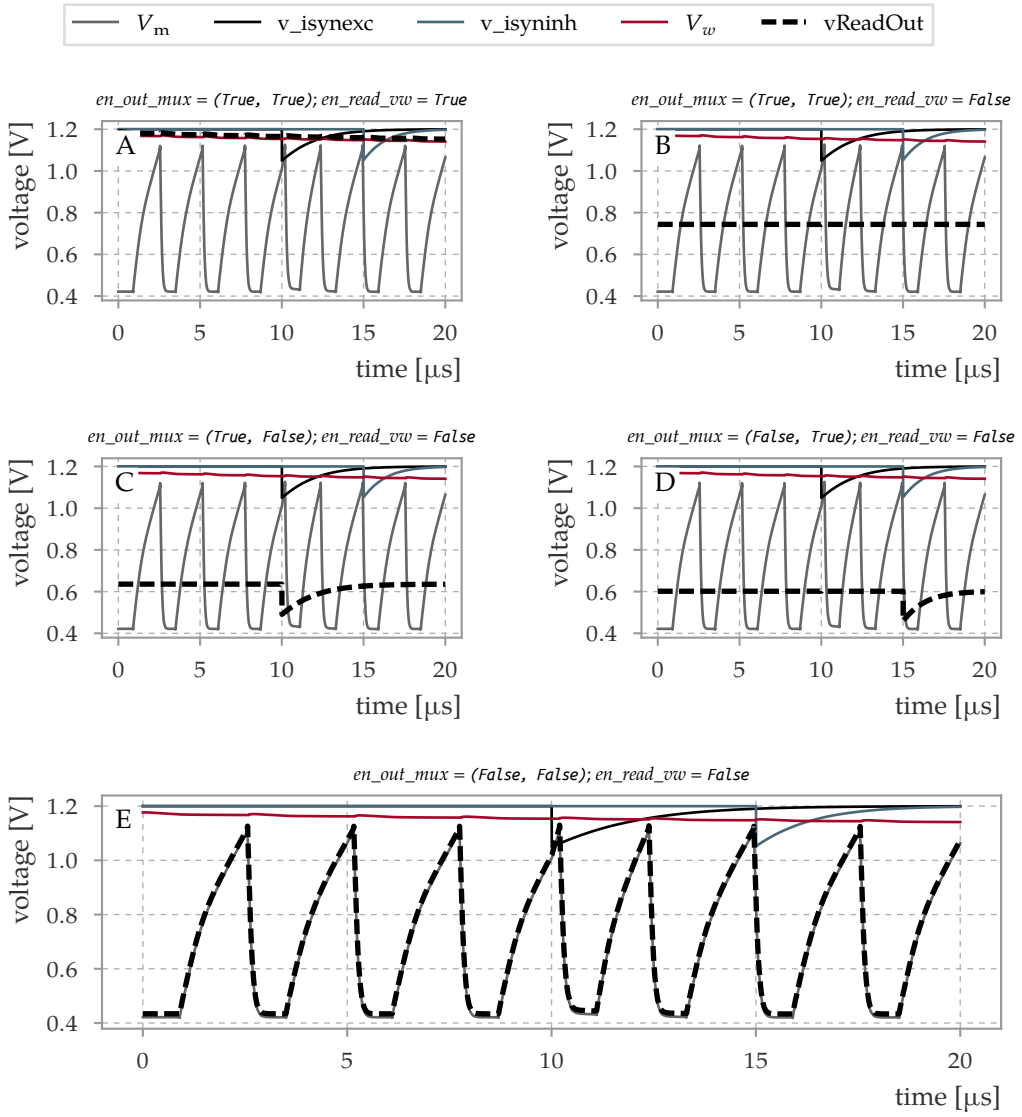


Figure 3.50: Usage of read-out multiplexer. **A** and **B**: When the multiplexer is set to read out V_w, *en_read_vw* must be set additionally to read out the signal. **C** and **D**: The excitatory and inhibitory input lines are read out using a source follower (lower voltage at read out). **E**: Membrane voltage read out.

15. *en_nmda:* Neurons can be connected using the inter-compartment switch (fig. 3.8). When the neuron circuits are disconnected, the input current to neuron compartment causes it to spike (fig. 3.53, solid lines). When the compartments are connected, the current flows into the leak terms of all four compartments (dashed lines).

16. *Scaling the inter-compartment conductance:* The parameters *ib_nmda_mul4* and *ib_nmda_div4* can be used to scale the inter-compartment conductance.

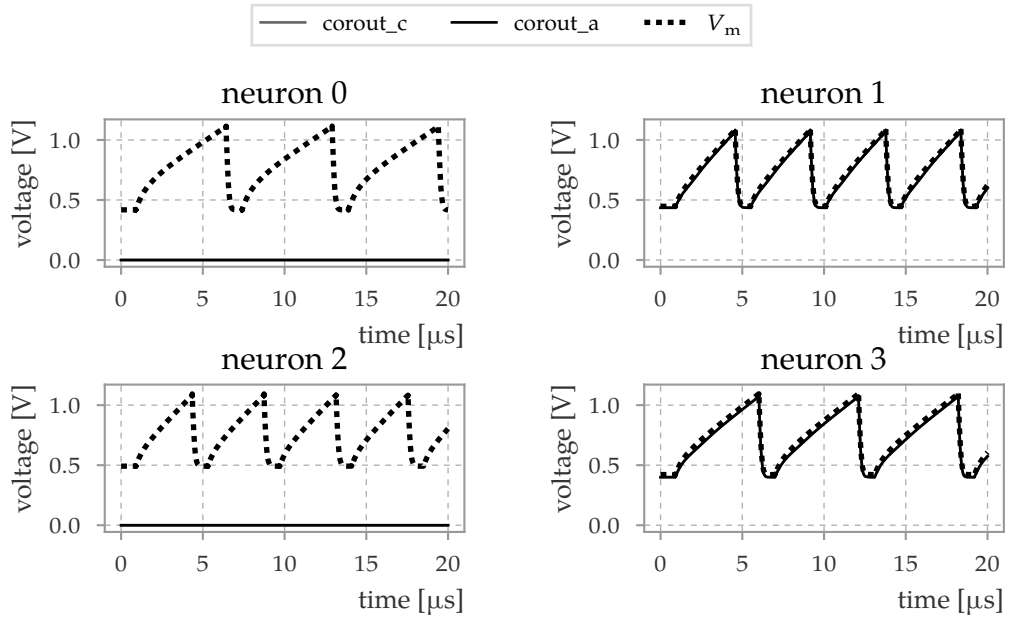


Figure 3.51: Read-out path towards correlation ADC. For neurons 1 and 3 the readout amplifier is enabled. The voltage signal arrives at the readout lines for the correlation ADC.

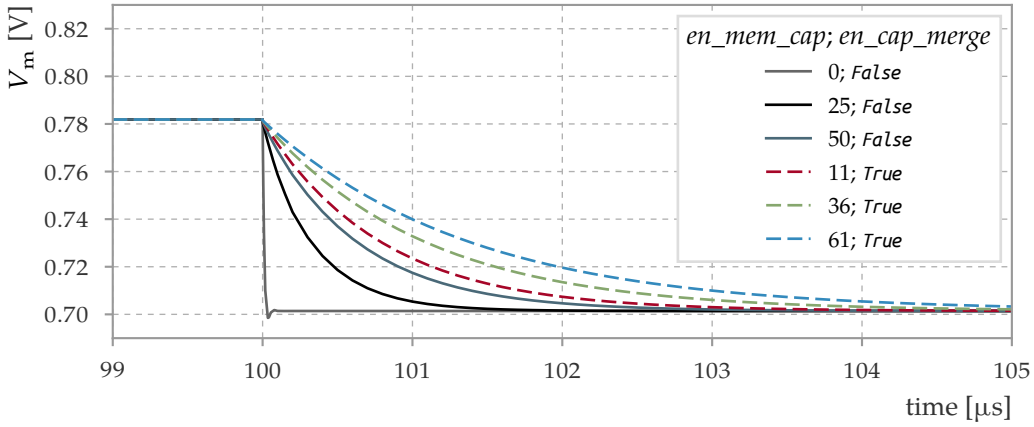


Figure 3.52: Capacitor scaling and merging with adaptation capacitor. The solid lines are recorded with different settings of the switchable membrane capacitance. The dashed lines additionally use the 2 pF capacitance in the adaptation term.

17. *Offset current switch:* The offset current can be switched by the *en_mem_off* switch.

18. *Adaptation sign:*. The sign of spike-triggered and sub-threshold adaptation can be changed by *en_pos_vw* and *en_neg_va* (fig. 3.54).

19. *High conductance mode:* The high conductance switches for leak and reset

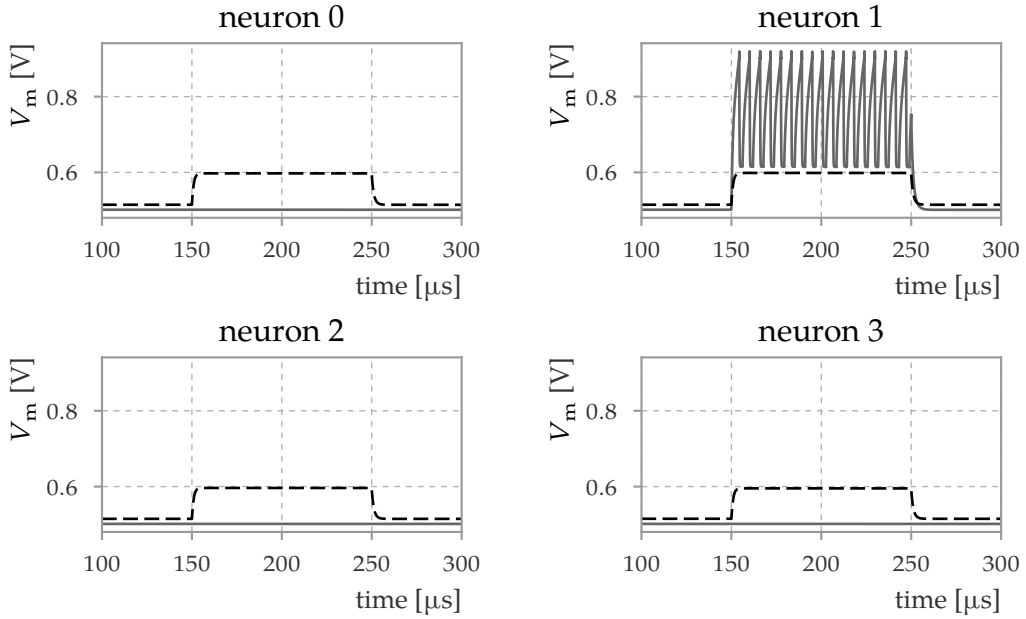


Figure 3.53: Verification of inter-compartment connectivity. Each panel shows the result of two simulations for each of four compartments. Solid lines: The connection to the *NMDA*-line is enabled, but the connection between the individual *NMDA* line segments is non-conducting (*en_nmda* = `True`, *en_scon* = `False`, see fig. 3.8). The current input that is injected to neuron 1 is not propagated to its neighbors. Dashed lines: The *en_right* switches are enabled and the membrane compartments are short-circuited. Simulation performed by Laura Kriener.

function independently as shown in fig. 3.55: The return to the leak potential is fast in panels C and D, and the reset towards the reset potential is fast in panels B and D.

20. *Leakage switch:*. The switch *en_ota* can be used to disable current from the OTA. Note that the switch is implemented as a thin-oxide transmission gate and it has to be ensured that the leak OTA does not emit a positive current by setting the bias currents and leak and reset potentials to low values. Otherwise the OTA can enforce a voltage above 1.2 V and the transmission gate will conduct even in the non-conducting setting.

21. *Bypass sensitivity:* For the DLS3 prototype chip, the pulse width of synaptic evens is reduced to the pulse width of the targeted large-scale devices, as compared to previous prototype versions where the pulse was generated by the attached field-programmable gate array (FPGA). The bypass mode was optimized to the pulse strength associated with the previous input scheme. It was discovered that with pulse lengths of 4 ns, the bypass does not function for a single input event. In the *tt* corner, four input spikes are required to trigger an output spike. This number varies between seven in the *ss* and two in the *ff*

corner. A volley of events can be used to trigger the bypass output (fig. 3.56). The chip also offers to use a larger strength of the synaptic input when using an external voltage bias for the synapse DACs instead of the internal *v_bdac*. It was not tested whether this can be used to increase the synaptic strength sufficiently. This issue *was not remedied* before tape-out due to limited time.

22. *Bypass polarity:*. It was discovered that the polarity of the bypass output was inverted, leading to an inoperable bypass. The mistake was remedied before tape-out (commits `0ed2c1c5` and `42704a9d` in the repository `hicann-dls-fc`).

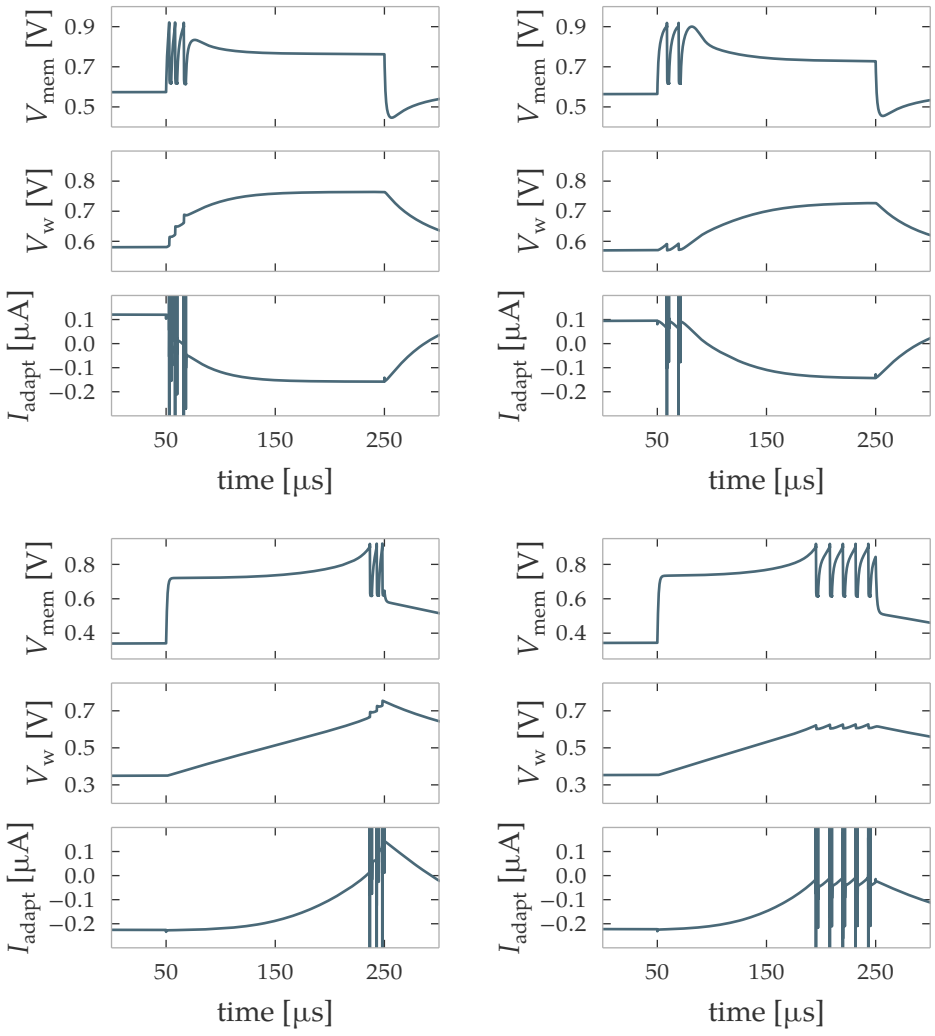


Figure 3.54: Switching the sign of a and b. Each group of panels shows the membrane voltage (top), the adaptation voltage (center) and the adaptation current (bottom). Top left: $a > 0$; $b > 0$ (*en_neg_va* = `False` and *en_pos_vw* = `True`). Top right: $a > 0$; $b < 0$ (*en_neg_va* = `False` and *en_pos_vw* = `False`). Bottom left: $a < 0$; $b > 0$ (*en_neg_va* = `True` and *en_pos_vw* = `True`). Bottom right: $a < 0$; $b < 0$ (*en_neg_va* = `True` and *en_pos_vw* = `False`). The vertical lines in the current recording are artifacts of the simulation due to the switching of reset and adaptation signals. (Kriener, 2017, Figure 29)

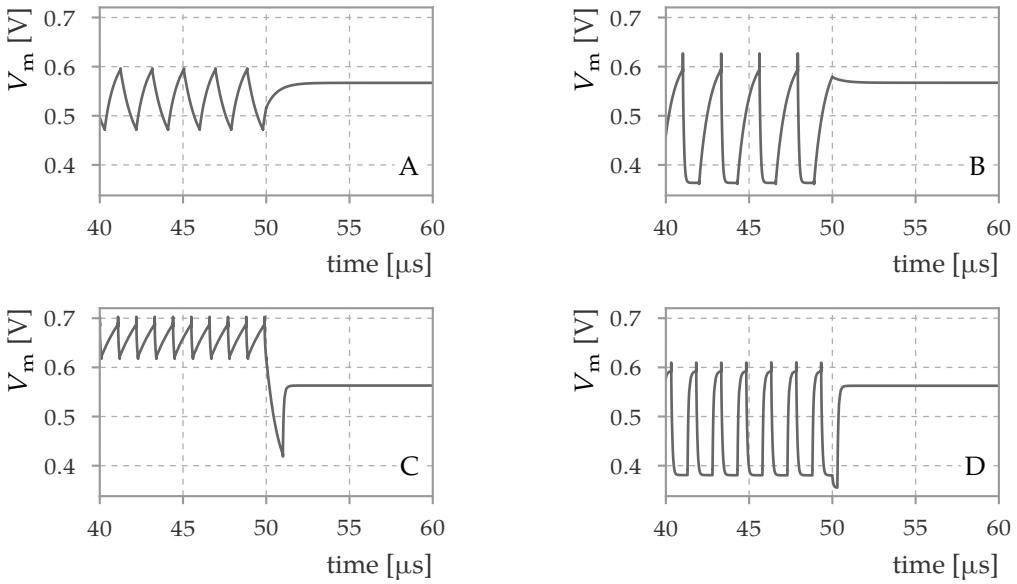


Figure 3.55: Switching the high conductance mode for leak and reset. The bias currents are 1 µA for *i_bias_leak*, *i_bias_leak_sd*, *i_bias_res*, *i_bias_res_sd*. The stimulus current is 250 nA for the small and 1000 nA for high leak conductance setting. **A**: *highs_leak* = `False`; *highs_res* = `False`. **B**: *highs_leak* = `False`; *highs_res* = `True`. **C**: *highs_leak* = `True`; *highs_res* = `False`. **D**: *highs_leak* = `True`; *highs_res* = `True`. The different voltages at which the membrane potential settles during reset in B and D are caused by a different magnitude of stimulus current.

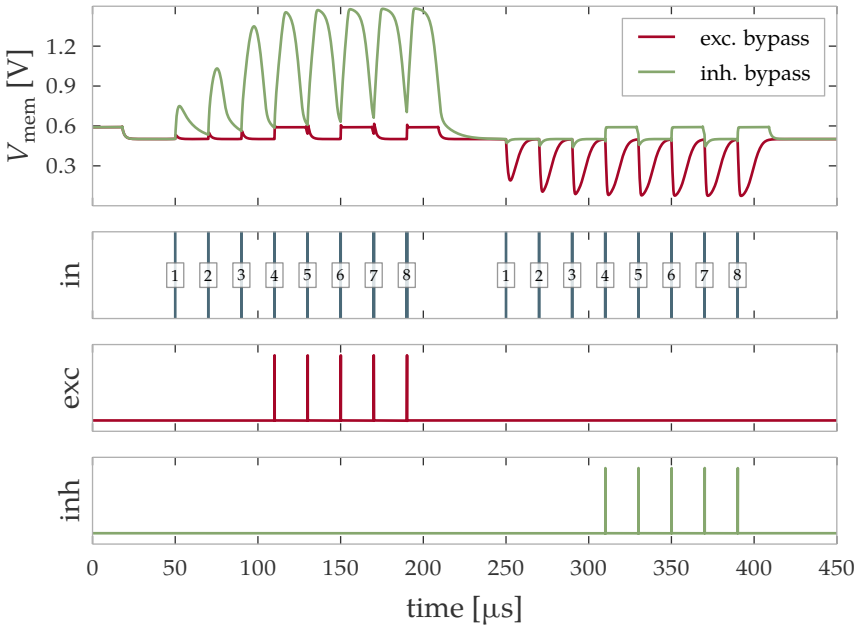


Figure 3.56: Enabling of the excitatory (red) and inhibitory (green) bypass. The global strength of the synaptic current is set using $v\_bdac = 1\,\mu A$. The input spikes are shown in blue. Each vertical line denotes a volley of input spikes with an inter-spike interval of 8 ns. The number of spikes in the volley is indicated in the overlaid box. The output fire signals are shown in red and in the two bottom panels. The difference in the post-synaptic potentials (top) with enabled and disabled bypass (top) is caused by the shortening of the synaptic time constant by the bypass circuit. This is an intended feature of the bypass circuit that increases its temporal resolution: The input line must be reset quickly to limit the length of the fire output pulse and to be able to receive new events. Taken from (Kriener, 2017, Figure 31)

3.4.2 Adaptation term

Leakage in the capacitance merge switch

Because capacitance is a valuable resource and leaky integrate-and-fire models without adaptation are used in a significant proportion of studies, merging of the membrane and adaptation capacitors was introduced. Due to the discovered possibility of V_w rising significantly above 1.2 V and consequently a leakage onto the membrane, the switch was implemented as a thick-oxide transmission gate. Figure 3.57 shows the state before the change, when a standard transmission gate was used for this switch.

Leakage through read-out multiplexer

A similar problem occurred in the readout multiplexer, which is implemented as thin-oxide transmission gate tree (fig. 3.14). The synaptic line inputs are buffered, but V_m and V_w are connected directly. If one of these voltages exceeds the operation voltage of the thin-oxide transmission gates, leakage onto the other occurs. To prevent this, a thick-oxide transmission gate between C_w and the readout multiplexer was introduced (*en_read_vw*, fig. 3.50).

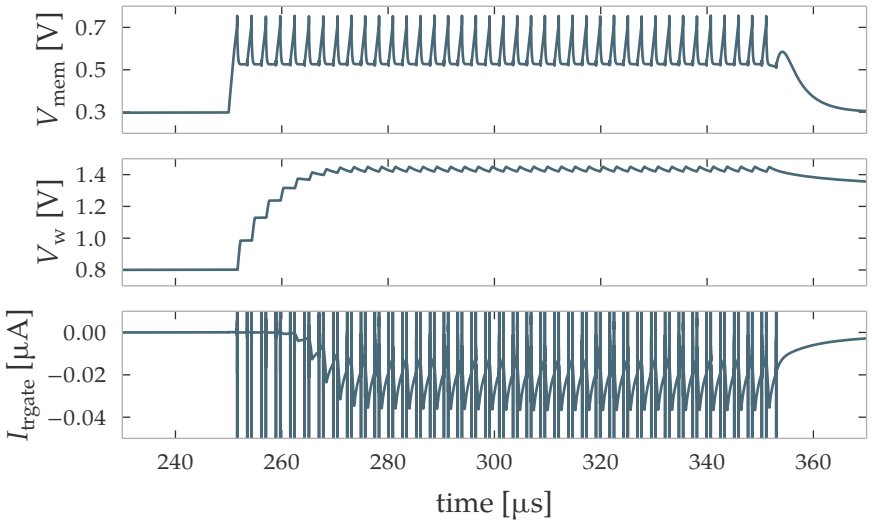


Figure 3.57: Leakage through the transmission gate that connects the membrane and adaptation capacitors. Top: membrane voltage of the neuron, which spikes continuously. Center: a large spike-triggered adaptation setting causes V_w to rise above 1.4 V. Bottom: current through the interconnecting transmission gate. The vertical lines in the bottom plot are caused by the transitions due to the switching between leak and reset mode in the leak/reset OTA. Figure taken from (Kriener, 2017, Figure 25).

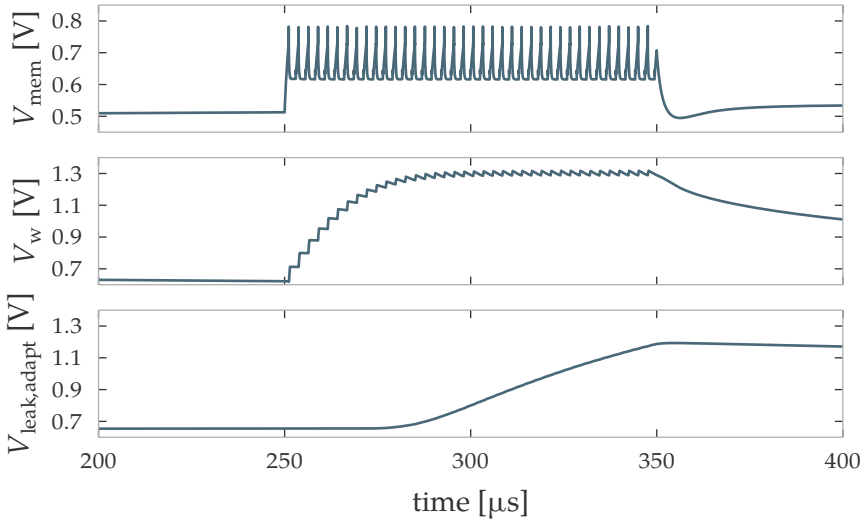


Figure 3.58: Leakage onto capacitive memory. Top: Membrane voltage. Center: adaptation voltage. Bottom: voltage on capacitor in the model of the analog parameter storage. Taken from (Kriener, 2017, Figure 26).

Leakage onto capacitive memory cell

The sign switching of the sub-threshold adaptation a was implemented using thin-oxide transmission gates at first. Due to the same problem as described in section 3.4.2, a leakage onto the analog memory capacitor occurred in cases where the voltage V_w exceeded the operation voltage of the multiplexer. As a consequence, the leak voltage for the adaptation, *v_leak_adapt*, was not stable, as shown in fig. 3.58.

Digital-analog adaptation pulse interface

The interface between the analog neuron circuit and the digital neuron back end foresaw the use of the *post pulse* – the signal that is passed on to the synapses for correlation measurement – to be used to enable the spike-triggered adaptation. Because a local timing generation or an unreasonably large current mirror would have been required within the neuron, the design was changed. In the produced chip version, the digital neuron back end generates a distinct timing pulse for the spike-triggered adaptation (see section 3.2.1). The design was changed in commit `bf5249ce`.

3.4.3 Spike-triggered adaptation signal

The current that is responsible for the increase in V_w for spike-triggered adaptation is conducted through two PMOS devices (M_1 and M_2 in fig. 3.22). This arrangement limits the maximal voltage seen at the switch s_1. In the original implementation, a total of three devices connected in series were used. This led to an early saturation of the current and consequently the available voltage step, as shown in fig. 3.59 (left).

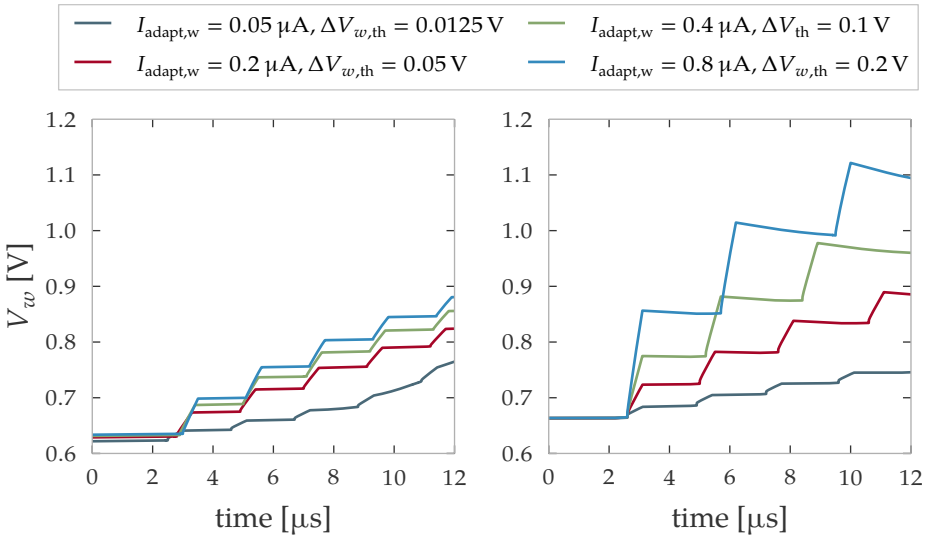


Figure 3.59: Steps in the voltage V_w on the adaptation capacitor induced by the spike-triggered adaptation mechanism using three (left) and two (right) transistors in the path. Figure taken from (Kriener, 2017, Figure 27)

The implementation was changed before tape-out to include only two transistors, as shown in fig. 3.22, leading to a reduced saturation (fig. 3.59, right). It must be noted that the method for reducing the maximal voltage is still suboptimal due to its dependence on the process corner. Directly switching the output of a current cell of the capacitive memory also has disadvantages because the voltage seen at the output of the capacitive memory cell changes quickly and can cause capacitive cross-talk onto the storage capacitor (cf. section 3.7.3).

3.4.4 Leak and reset term

Leakage between analog memory cells

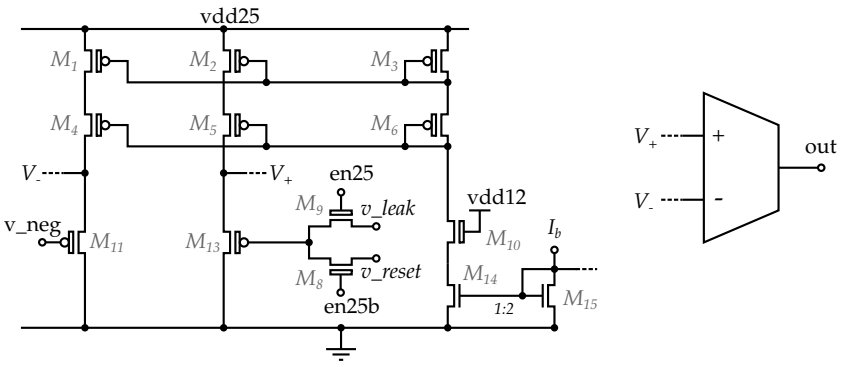


Figure 3.60: Initial implementation of switching between *v_leak* and *v_reset*. The finally produced version is shown in fig. 3.16.

The leak/reset OTA switches between *v_leak* and *v_reset* for the duration of the

neuron reset. The initial switching mechanism is shown in fig. 3.60. The complementary control signals "en25" and "en25b" flip during the beginning and end of the neuron reset. Due to the timing of the signals it can occur that both transistors are sufficiently conducting to transfer charge between the two voltage parameters. Leak and reset parameters initially set to 1.1 V and 0.3 V would change to approximately 0.95 V and 0.4 V after ten reset cycles. The issue was remedied before tape-out in commit `3bda16a7` in the repository `hicann-dls-fc`.

Process corner dependency of level shifter

The 2.5 V signals "en25" and "en25b" that switch the OTA between the reset and fig. 3.16 are derived from a 1.2 V signal using two identical level shifting circuits. The circuit switching time was delayed by more than 0.2 µs in the *ss* corner. The effect itself is minor, because typical refractory periods are on the order of microseconds (table 3.1). However, it can cause an overshoot of the membrane potential after a threshold crossing, due to a later onset of the refractory period. The problem was remedied in commits `c4963c66`, `018ff61a` and `df7139f0` in the repository `hicann-dls-fc`.

3.4.5 Summary

In summary, a large number of explicit verification simulations was performed, helping to uncover several problems in the system. Most of the problems were remedied by the respective designers before tape-out, with the exception of the bypass mode sensitivity, which was discovered too late in the design preparation process. While the bypass is only a minor feature in a small prototype chip because the presence and absence of synaptic input can be monitored using access to analog signals, it is an important tool for software development and hardware error diagnostic on large-scale systems. Especially for neuromorphic devices with neurons which require a software-based calibration procedure it is essential to have a reliable debugging feature that is robust against effects such as mismatch, process corner and variation of supply voltage and temperature. Otherwise, the localization of hardware configuration or implementation errors is impeded because the source of the error can not be easily attributed to one of the many, yet uncharacterized, components in a long signal transmission chain.

The implementation of the spike-triggered adaptation can be optimized to provide a more robust implementation of limiting the voltage at C_w to less than 1.2 V which is independent of the process corner. The issue extends to other circuits connected to C_w and C_m: The synaptic input OTAs, the leak/reset OTA, the adaptation term and the offset current can all produce currents at output voltages significantly higher than 1.2 V which is not inherently compatible with thin-oxide circuits attached to the membrane. Care must be taken during operation of the chip to prevent undesired behavior, such as configuring correct *v_syn_[exc | inh]*, *v_leak* and *v_reset* when disabling a circuit (section 3.4.1).

Considering the type of errors that were uncovered it becomes apparent that most issues arose at the boundary between individual components. This was to

be expected, because the simulation setup presented in this chapter encompasses the integration of several components, and further, the individual components themselves are thoroughly simulated by their respective designers. The multiple occurrence of issues arising with the interface to the capacitive parameter storage – in the dynamic switch of the leak OTA and the static multiplexer in the adaptation term – suggests that more thorough simulation should be considered, even at the level of individual component simulation. Implementing a more detailed behavioral model with automatic detection of failure conditions may benefit future development.

The integrative simulation approach described in section 3.1.3 proved useful in uncovering errors that would otherwise have been detected post-silicon. Errors such as leakage onto capacitive memory in particular are difficult to discover in measurements, because erroneous neuron behavior caused by this effect can not be immediately attributed to the analog storage, but can be traced comparatively easily in simulation – see, e.g., fig. 3.58. The verification of the full-custom analog implementation requires explicit handling of each parameter. Other tests can and should be automated: The connectivity between the SRAM cell to the input of the individual component can be tested programmatically. The load on the cells of the capacitive memory could also be verified for all cells during simulation by an automatic check.

3.5 Full neuron test cases

In this section, two test cases are presented that showcase the functionality of the neuron circuit. The first use case is the reproduction of various firing patterns in response to a current stimulus that are characteristic for the mathematical AdEx model. The second use case is the configuration of the circuit for LIF sampling (section 1.8). The necessity of prior calibration is examined for this case.

3.5.1 Firing patterns in the AdEx model

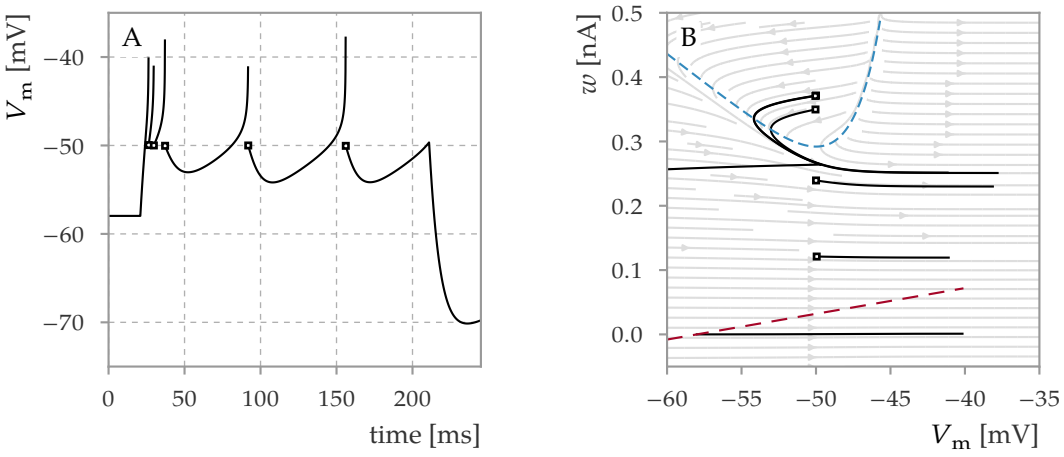


Figure 3.61: Simulation of the mathematical AdEx model. **A**: Membrane time course of the *initial bursting* firing pattern. A burst of three spikes with a short interspike intervals is followed by singular spikes separated by broad interspike intervals. **B**: Full state space of the same simulation as in A. The gray dashed line shows the V-nullcline, the red dashed line the w-nullcline. All quantities are shown in the biological value domain. Re-simulated and redrawn from (Naud et al., 2008, Figure 4c).

One advantage of the AdEx neuron model is the possibility to capture a wide range of neuron firing behaviors with a computationally inexpensive model by tuning a limited number of parameters (Markram et al., 2004). This property is even more important for hardware implementations of physical neuron models, which cannot be changed quickly once produced. A very illustrative way to showcase the capability of the neuron model is the production of various firing patterns in response to a simple step current stimulus.

Naud et al. (2008) demonstrate the versatility of the AdEx model by showing eight different firing patterns. Figure 3.61 A shows the *initial bursting* pattern as example: After a volley of spikes the inter-spike interval increases and the membrane potential drops after a spike before it rises again. The mechanism that generates this pattern can be explained by the evolution of the state space in the two-dimensional plane $(V_m; w)$: When the step current is enabled, the membrane follows the path given by the differential equation (denoted by the stream lines). After each spike,

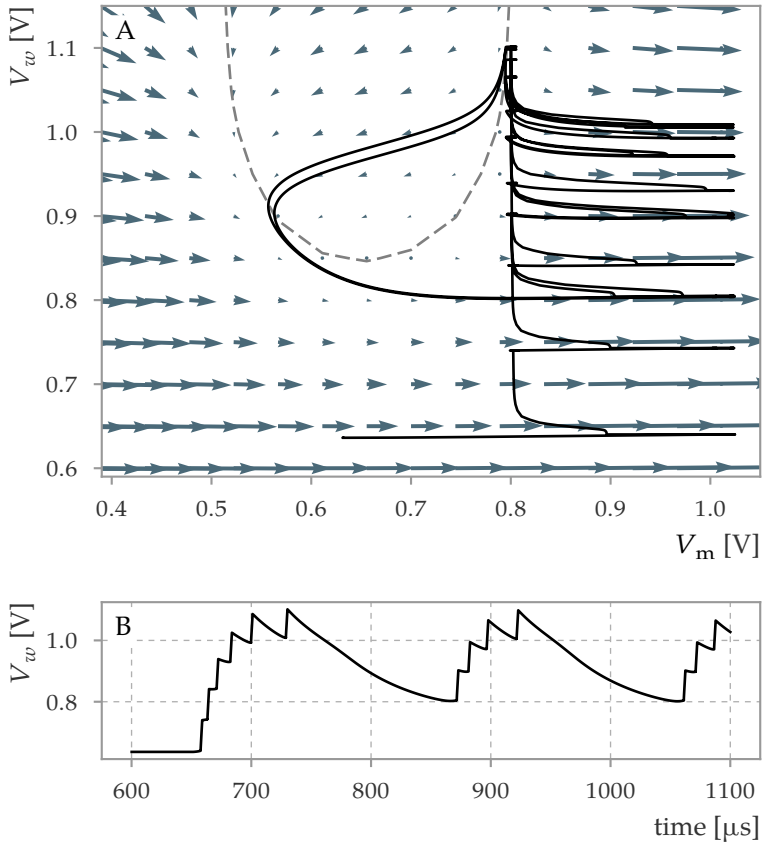


Figure 3.62: **A**: State space of the simulated neuron circuit for the voltage trace shown in fig. 3.63 F. The arrow length and direction is proportional to the total current onto the capacitors C_m and C_w. Note that, in contrast to the simulation of the ideal model in fig. 3.61 B, V_w is shown instead of w. **B**: Time course of the adaptation voltage. Adapted from Aamir et al. (2017a).

the w variable is incremented by b (in this case $b = 0.12\,\text{nA}$). After three spikes, the trajectory is reset with the w variable above the V_m-nullcline (gray dashed line). Now, V_m has to decrease first before a new spike can be triggered. This causes the broad resets seen in fig. 3.61.

In hardware implementations of the AdEx model, the reproduction of the firing patterns shown by Naud et al. (2008) has become a standard test (see, e.g., Millner et al. (2010), Millner (2012), Kiene (2014)). Figure 3.63 shows the circuit-level simulation of the DLS3 neuron that is set up to qualitatively reproduce different firing patterns. Varying levels of adaptation can switch the behavior from tonic spiking (A) to transient spiking (C). Panel D shows initial bursting, caused by the same mechanism as in the solution of the mathematical model in fig. 3.61. In panel E shows a delayed accelerating pattern. Here, the trajectory has to pass a slow region below the V_m-nullcline before the neuron can emit a spike, being pulled up by the exponential term. Additionally, the effect of negative adaptation ($a < 0$) causes decreasing interspike intervals. Panel F shows regular bursting; the equivalent state space of the circuit-

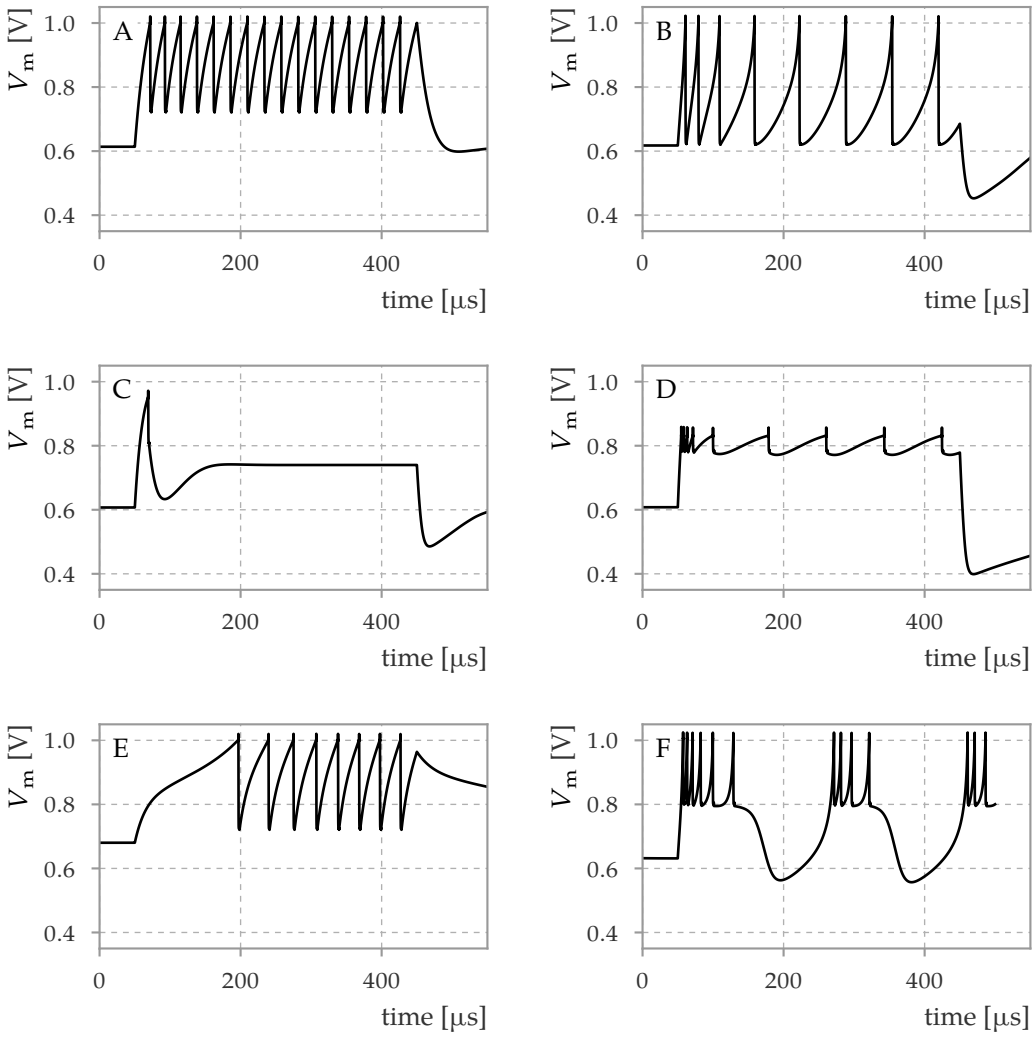


Figure 3.63: Multiple firing patterns in the circuit-level simulation. **A**: tonic spiking **B**: adaptation **C**: transient spiking **D**: initial bursting **E**: delayed accelerating **F**: regular bursting Adapted from Aamir et al. (2017a) and Kriener (2017).

level simulation is shown in fig. 3.62 A. A curved V_{m} nullcline is present, just as in the abstract model (dashed lines). When the neuron is reset above the nullcline the derivative of V_{m} is negative and the trajectory takes a detour over the low-V_{m} part of the state space. The shape of the nullcline differs clearly from the theoretical expectation shown in fig. 3.61, primarily due to the nonlinearity of the leak current at low membrane voltages. The Δ_{T} parameter in the hardware implementation is not adjustable in the current prototype version (section 3.3.8). Its value is larger than what would be the case for a linear transformation according to eq. (1.17), which also contributes to the difference in the shape of the nullcline. Two of the firing patterns are omitted in fig. 3.63. The first is the delayed regular bursting pattern, which could be reproduced but has an unstable resting state. It is described in detail in Kriener (2017). The irregular firing pattern could not be reproduced in simulation. The reason for this is that the region of the parameter space in which it occurs is sparse (Naud et al., 2008, fig. 6) and a parameter search was too time-consuming in simulation.

The simulations show that the properties of the AdEx dynamic system are incorporated in the circuit design to the extent that is necessary to qualitatively reproduce multiple firing patterns. The properties of the implemented circuit, in particular the non-adjustable Δ_{T} and the nonlinearity of the leak term, made a direct mapping of the parameters given in (Naud et al., 2008) unfeasible.

3.5.2 LIF-sampling calibration using novel reset functionality

In this section, a model-oriented calibration procedure is evaluated in simulation, which is complementary to the general characterization and calibration steps described in section 3.3. In many cases, modeling with neuromorphic devices is not based on the exact setting of effective properties of the physical neuron model, such as time constants or synaptic weights, but on more high-level properties of neurons, such as the f-I curve. Pfeil et al. (2013), for example, present the emulation of six different network models on the neuromorphic chip *Spikey*. For two models, the generic neuron calibration on the chip is complemented by model-specific parameter tuning: For the emulation of a balanced random network, synaptic weights are scaled to achieve a homogeneous firing rate in a population of neurons. Similarly, for the insect antenna lobe model, additional external and recurrent connections are created to approach the desired firing rate.

For the LIF sampling model (section 1.8), the fundamental property of the underlying spiking neuron is the presence of a sigmoid activation function. On hardware devices that are subject to fixed-pattern variation of parameters, this activation function can be recorded in an initial step. Then, the weights and biases of an abstract Boltzmann machine can be translated to hardware parameters for operation. An example of this approach for the HICANN system is documented in Kungl (2016) (a master thesis that was co-supervised by the author). In the following we investigate in simulation how the characteristics of the DLS3 neuron affect this kind of model-specific calibration.

The re-design of the reset circuit allows for a new interpretation of the LIF-sampling paradigm that is described in section 1.8. Because the reset conductance can be configured, it can also be set to the same value as the leak conductance. Additionally setting $V_{\text{leak}} = V_{\text{reset}}$ enables the decoupling of the membrane voltage and the spike output of the neuron. Figure 3.64 shows the resulting behavior of such a setup. The neuron is stimulated by excitatory and inhibitory Poisson background input (A) which results in a fluctuating membrane voltage (B). When the membrane crosses the firing threshold (C) the neuron starts emitting spikes with a firing frequency equal to the inverse refractory period (D), until the membrane potential drops below the threshold. This is equivalent to the spiking behavior of fig. 1.5 with the difference that the hardware membrane voltage is a physical representation of $V_{\text{eff,curr}}$ and not V_{m}. This removes the requirement for a short membrane time constant or a small distance between threshold and reset potential – both being features that are not easily available in analog neurons, the first due to limited parameter ranges and the second due to the need to precisely calibrate and configure the threshold and reset circuits.

A possible method to characterize the response of a neuron to an external input is to sweep the parameter *i_mem_off*, as shown in fig. 3.65. In the simulation for one neuron instance, the offset current shifts the membrane potential without significantly changing the voltage course, as shown in panel D. Figure 3.66 exemplifies the need for calibration: When the same parameter sweep as in fig. 3.65 is performed for

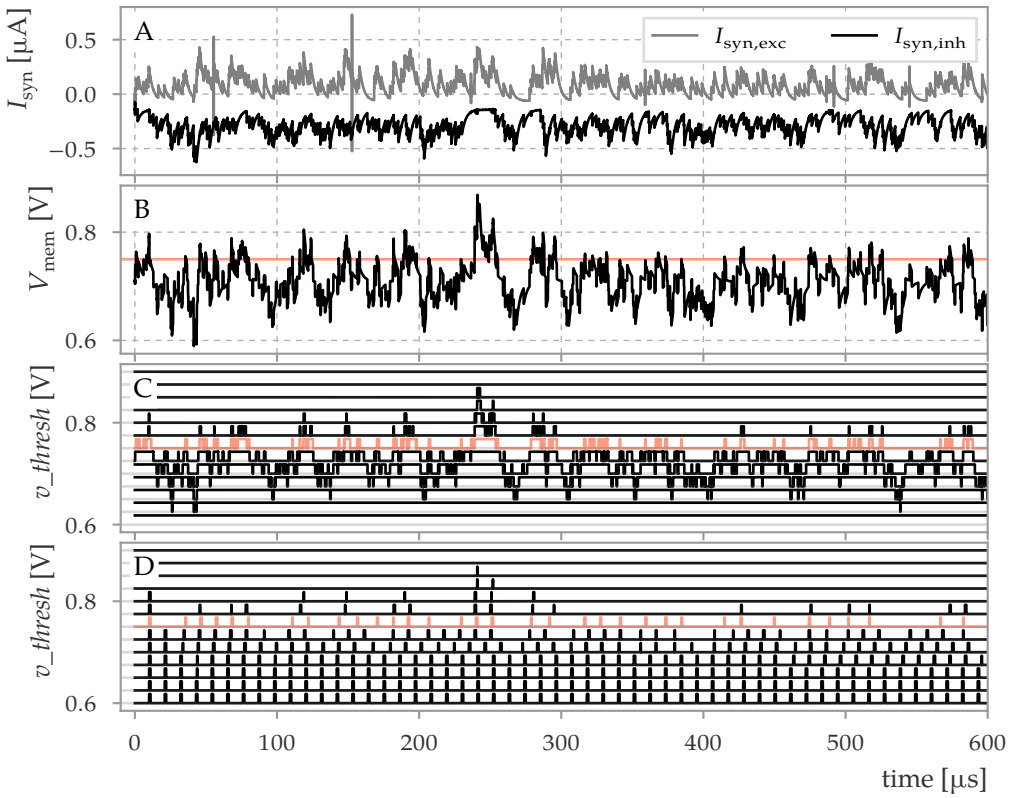


Figure 3.64: *LIF sampling* operation mode using non-resetting membrane. Simulation of a Monte-Carlo sample. **A**: Excitatory and inhibitory synaptic input currents for synapses stimulated with 500 kHz refractory Poisson input (with a small additional refractory period of 8 ns to prevent spikes from arriving too close to each other). No calibration is applied, and the offset of the synaptic input is visible in particular for the inhibitory synaptic current. **B**: Membrane potential resulting from input in A. The value of *v_thresh* = 0.75 V is denoted by a horizontal bar. No calibration is applied, i.e., it does not necessarily correspond to the effective firing threshold. **C**: The value of *v_thresh* is swept and the value of the digital *fireout* signal is shown. At low values of *v_thresh*, the neuron is constantly supra-threshold and *fireout* is high. The *fireout* signal for the *v_thresh* value marked in B is highlighted. **D**: *resetout* signal for the neuron. The neuron fires periodically when the membrane is above the firing threshold with an interspike interval given by the refractory period (here: 10 µs). The on-duration of the *resetout* is short because a long *holdoff*-value is used. A slow clock of 1 MHz is used for the reset.

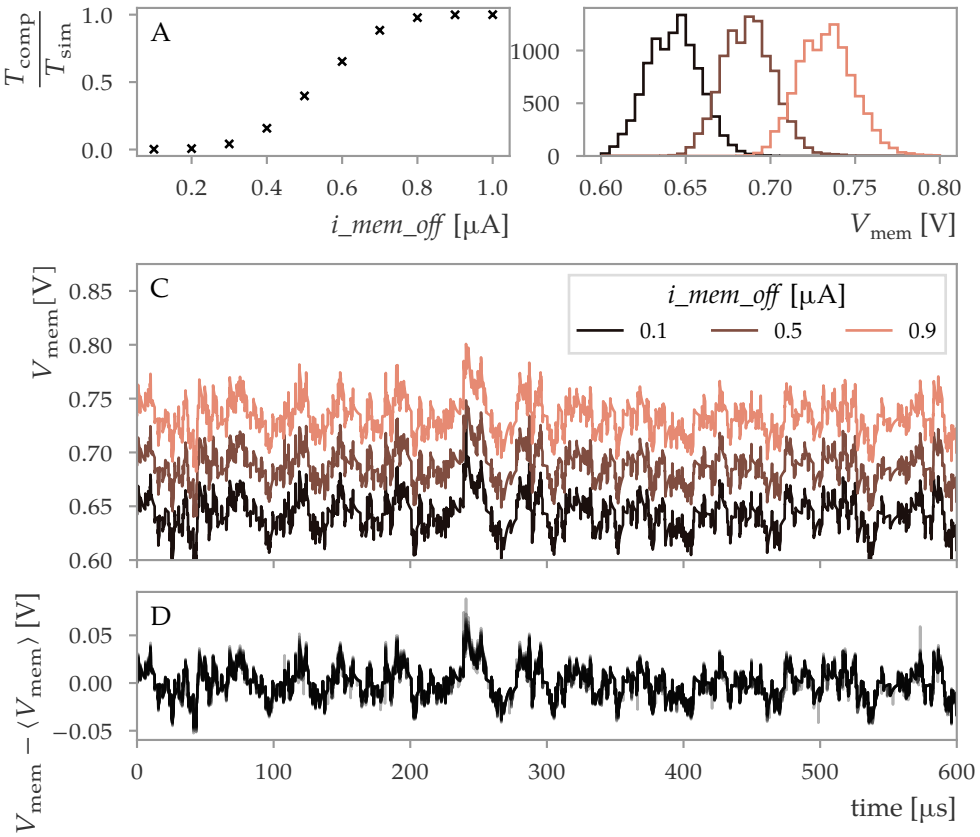


Figure 3.65: *LIF sampling* operation mode controlled by *i_mem_off*. **A**: fraction of time in which the neuron is above threshold, calculated as the fraction of time that the *fireout* signal is in the high state (T_{comp}) and the total simulation time (T_{sim}). **B**: histogram of the membrane voltage for a simulation duration of 1 ms. The colors encode the same value of *i_mem_off* as in C. **C**: Exemplary voltage trace for three values of *i_mem_off*. **D**: All measurements used for panel A superimposed to show that the parameter indeed effectively translates the membrane potential. The same parameter sweep for several Monte-Carlo samples is shown in fig. 3.66.

ten Monte-Carlo samples while using the same analog parameters for all simulated neurons, the location of the membrane voltage distribution varies greatly. For some samples, the voltage rises to high values above 1.2 V because the total (synaptic and offset) input current exceeds the saturation current of the leak OTA. The saturation current of the leak OTA is increased, together with its transconductance; the result is shown in the top right and both bottom panels. The membrane can be contained without applying any calibration for the highest possible setting (lower right panel). However, only the maximal bias prevents the membrane potential from entering the nonlinear regime. The resulting increase in transconductance leads to a correspondingly small width of the distribution of the membrane potential and a small impact of the offset current.

Figure 3.67 shows the result using a calibration for synaptic offset, synaptic time constants, firing threshold and resting potential. The synaptic input offset (parameters *v_syn_[exc|inh]*) is calibrated for zero offset current for the inhibitory input and for $-0.5\,\mu\text{A}$ for the excitatory one. This allows to use the offset current *i_mem_off* to compensate positive and negative offsets of the leak OTA. Using this setup it is possible to measure the activation function of the neuron as a function of the mean of its membrane potential distribution (fig. 3.67 C).

Note that the issue shown in fig. 3.66 is still present. A high input current could drive the neurons out of the linear and into the saturation range of the OTA. This can be counteracted by using a low source degeneration bias which, however, requires a precise calibration of the membrane time constant. A second issue is the common mode dependency of the leak OTA. In the ideal model (eq. (1.4)), a constant offset in all voltage parameters – V_{leak}, V_{thresh}, reversal potentials etc. – only shifts V_{m} but keeps the dynamics of the model otherwise identical. In the hardware implementation this is not the case, as can be seen, in fig. 3.33 A: In addition to the saturation effects, the shape of the curve $I_{\text{reset}}(V_{\text{m}})$ changes depending on *v_reset*. It is therefore necessary to validate that this effect does not detrimentally affect the dynamics of the membrane. One method is to compare the offset-free component of the voltage trace, as is done in fig. 3.65 D for the sweep based on the offset current parameter.

In summary, the usability of the DLS3 neuron implementation for the LIF sampling network model is investigated with two main results: First, the implementation of a conductance-based reset enables to operate the neuron circuit in a reset-free mode where the membrane potential can stay above the firing threshold while the neuron emits spikes. This feature allows a new interpretation of the LIF sampling model, where the hardware membrane voltage corresponds to the free membrane potential in the model (section 1.8). This approach is shown to be viable for the DLS3 neuron circuits in transistor-level simulations including mismatch.

Second, the model uses a dedicated calibration procedure, separate from the general-purpose calibration presented in section 3.3. For this procedure, the activation function of a neuron that is bombarded with Poisson stimulus is measured. It would be desirable to implement exclusively this calibration, not relying on the general-purpose calibration, to allow faster experimental results after commissioning

of the chip. However, not using the basic calibration is complicated by the variation and saturation behavior of the leak term (fig. 3.66). When using the basic calibration methods that were developed in section 3.3, the required sigmoidal activation function can be achieved in simulation (fig. 3.67).

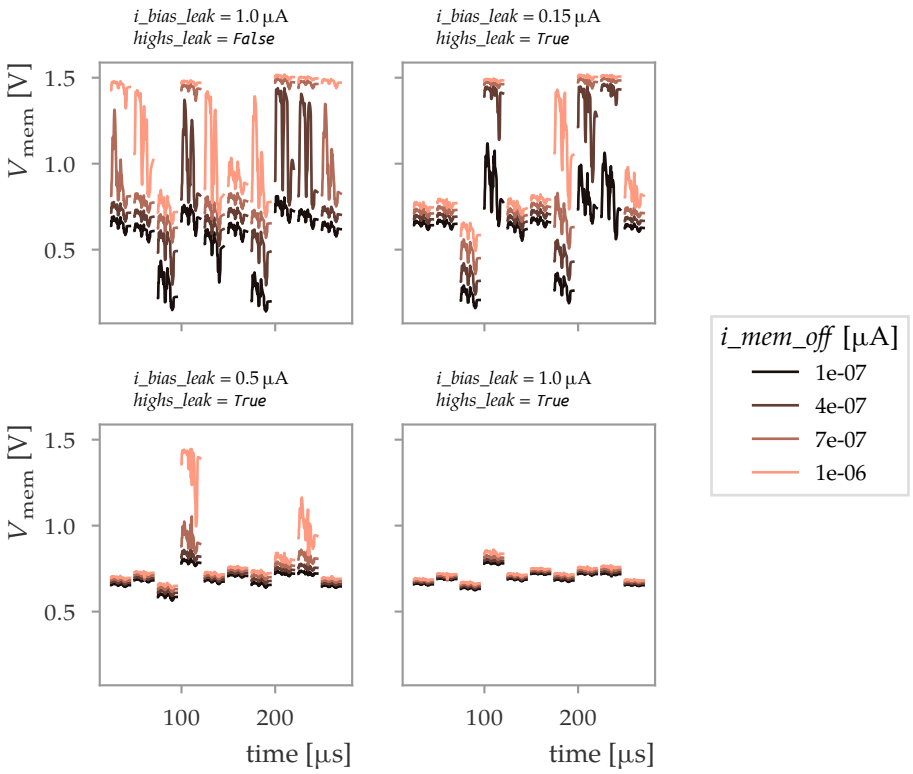


Figure 3.66: Activation function with uncalibrated parameters. The first 20 ms of the simulation in fig. 3.65 are repeated for 10 Monte-Carlo samples. The simulation results are arranged side by side for better visibility. fig. 3.65 corresponds to the first example on the upper right. *i_bias_leak_sd* is set to the same value as *i_bias_leak*, and the reset biases and *highs* switch are the same as the ones for the leak term. From top left to bottom right, the leak term conductance increases from the highest possible setting with *highs* = `False` to the highest setting with *highs* = `True`. Only in the highest setting the membrane voltage reliably stays below 1.2 V without further tuning. The width of the membrane potential distribution and the effect of the offset current is reduced by a large amount.

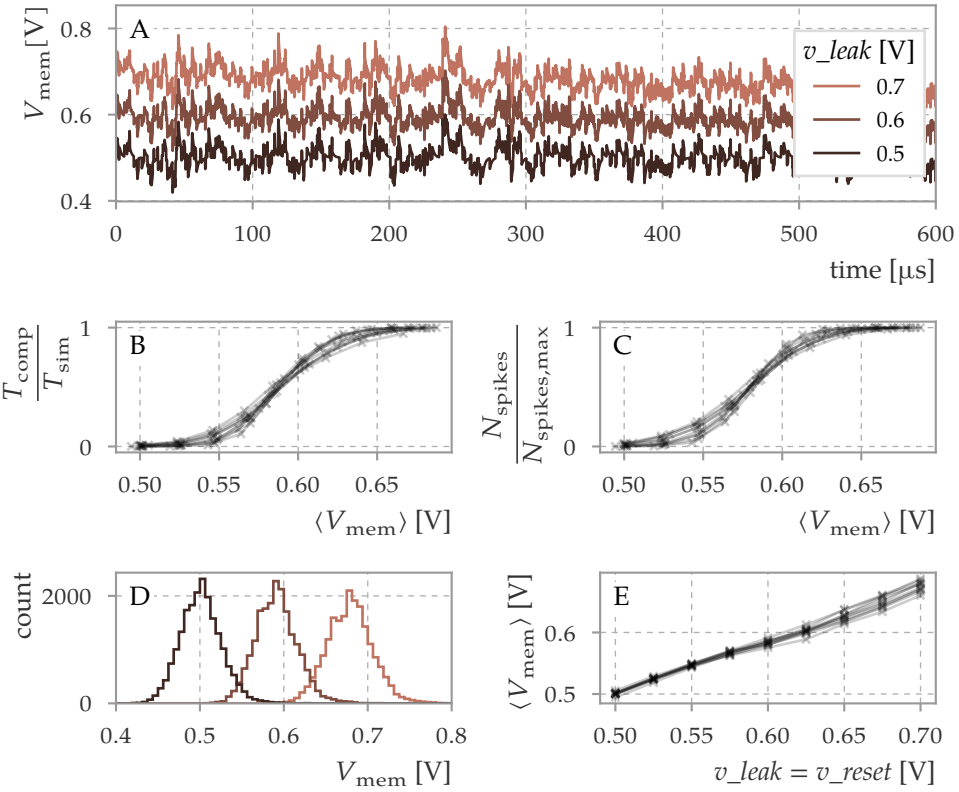


Figure 3.67: *LIF sampling* activation function with calibrated parameters. The synaptic offset, synaptic time constant and threshold calibrations are applied. Additionally, *i_mem_off* is set such that the resting potential is equal to *v_leak* when the neuron receives no synaptic input. This is done once for a resting voltage of 0.6 V and *i_mem_off* is not changed for the successive sweep, only the parameters *v_leak* and *v_reset*. **A**: Exemplary voltage traces. *v_leak* and *v_reset* are varied from 0.5 V to 0.7 V. The values for *i_mem_off* are adjusted only once, for *v_leak* = 0.6 V. Three exemplary traces for one Monte-Carlo sample are shown. The Poisson stimulus is the same as the one used in fig. 3.64 and equal for all Monte-Carlo samples and *v_leak* values. **B**: Fraction of time in which the neuron reports being above threshold, calculated as the fraction of time that the *fireout* signal is in the high state (T_{comp}) and the total simulation time. **C**: Activation function calculated from the number of emitted spikes divided by the maximal number of possible spikes. **D**: Distribution of membrane voltage that corresponds to the traces shown in A, calculated over 1 ms of simulation time. **E**: Correlation of *v_leak* parameter and mean membrane potential.

3.6 Multi-compartment simulations

In this section, transistor-level simulations for the multi-compartment use of the DLS3 neuron circuits are presented. The most simple use case is the direct connection of compartments which is implemented using a scheme similar to the HICANN system. Two neighboring membrane capacitors can be connected directly using the *en_right* switch (Figure 3.8). For future chip revisions with two rows of neuron circuits, the switch *en_bot* will connect adjacent membranes of the upper and lower half. An example of this functionality is shown in fig. 3.53. The primary purpose of this kind of direct connectivity is to increase the number of synaptic circuits that are connected to a single logical neuron, although it also allows to increase the maximal membrane capacitance, provide more than two synaptic time constants and more than one adaptation time constant per neuron.

The second mode of multi-compartment connectivity is implemented using a parallel interconnection line with switches *en_sconb*, to which individual compartments can be connected using a tunable resistance (fig. 3.25, section 3.2.8). In the following, application-oriented simulations of the conductance-based interconnection feature are presented after a description and a characterization of the inter-compartment conductance.

3.6.1 Inter-compartment conductance circuit

Figure 3.68 shows the behavior of the inter-compartment resistor (section 3.2.8). The current at fixed voltages is comparatively linear except when one of the voltages is low (panel A). This is also reflected in the conductance (panel B). The strength of the conductance can be adjusted over a wide range with the analog current bias (*i_bias_nmda*) and additionally by two scaling switches that modify the output transistors by a factor of four (fig. 3.68 C, fig. 3.25). The dependency on the process corner is limited, the mean maximal current varying between 3.2 µS and 5.4 µS for the *fs* and *sf* corners (fig. 3.68 D).

3.6.2 Active neuron compartments

The main new functionality for multi-compartment operation in DLS3 is the possibility to trigger plateau-potentials in each neuron compartment. The reset circuit that was present in previous generations is removed and the reset is implemented by switching the voltage and current parameters of the leak OTA for the duration of a reset (section 3.3.3). The use of a digital counter for the refractory period instead of the previous analog current-starved delay elements is beneficial because a larger range of refractory times can be realized as compared to the analog implementation (Aamir et al., 2017b). This is required to achieve the usual short refractory time used in LIF models to emulate action potentials as well as longer duration of plateau potentials and N-Methyl-D-aspartate (NMDA) spikes (fig. 3.70 B). For plateau potentials, the reset potential is set higher than threshold, so the compartment is pulled to a higher voltage after it crosses a certain voltage. A *holdoff* time is incorporated

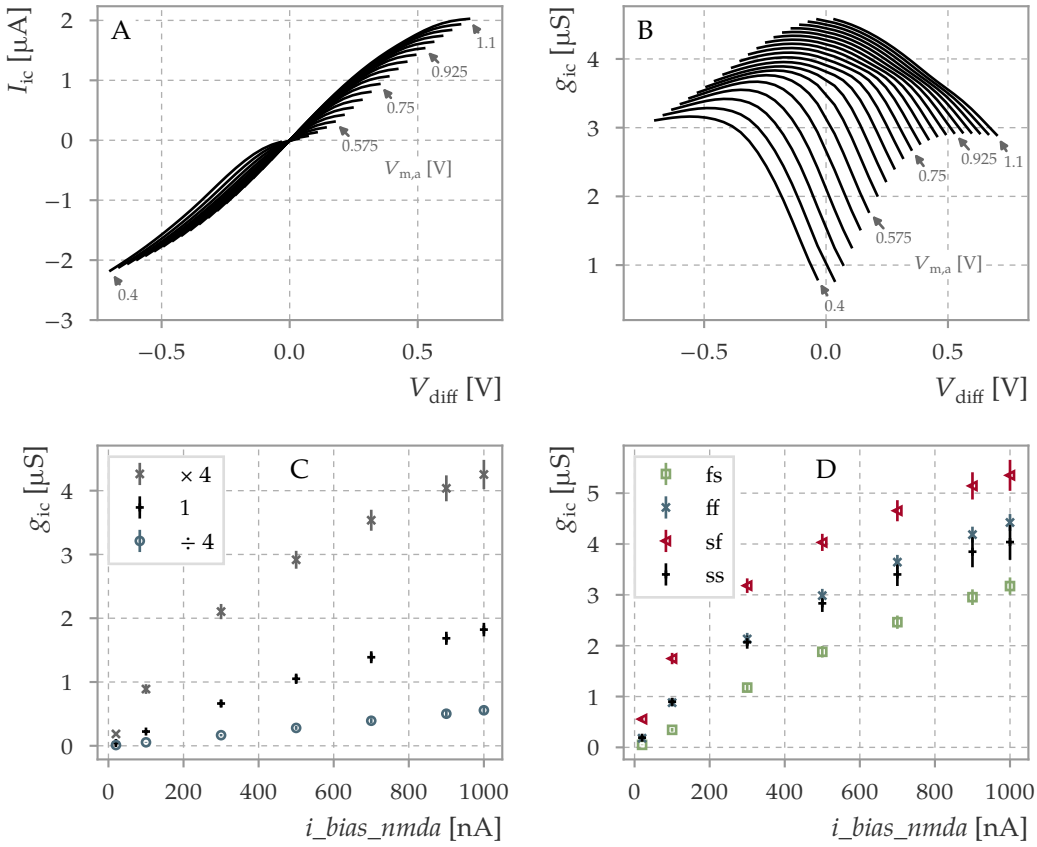


Figure 3.68: Simulation of the inter-compartment resistor circuit. **A**: The inter-compartment current as a function of the voltage difference $V_{\text{diff}} := V_{\text{m,a}} - V_{\text{m,b}}$. The voltage of one compartment, $V_{\text{m,a}}$ is indicated by arrows. *i_bias_nmda* = 0.5 µA and *ib_nmda_mul4* = `True`. **B**: The inter-compartment conductance, calculated from the data shown in A. **C**: Mean inter-compartment conductance as a function of the controlling bias for the three valid settings of *ib_nmda_mul4* and *ib_nmda_div4*. **D**: Corner simulation for the case of *ib_nmda_mul4* = `True`. In C and D the error bars show the standard deviation over the simulated voltage range. Simulations performed by Laura Kriener.

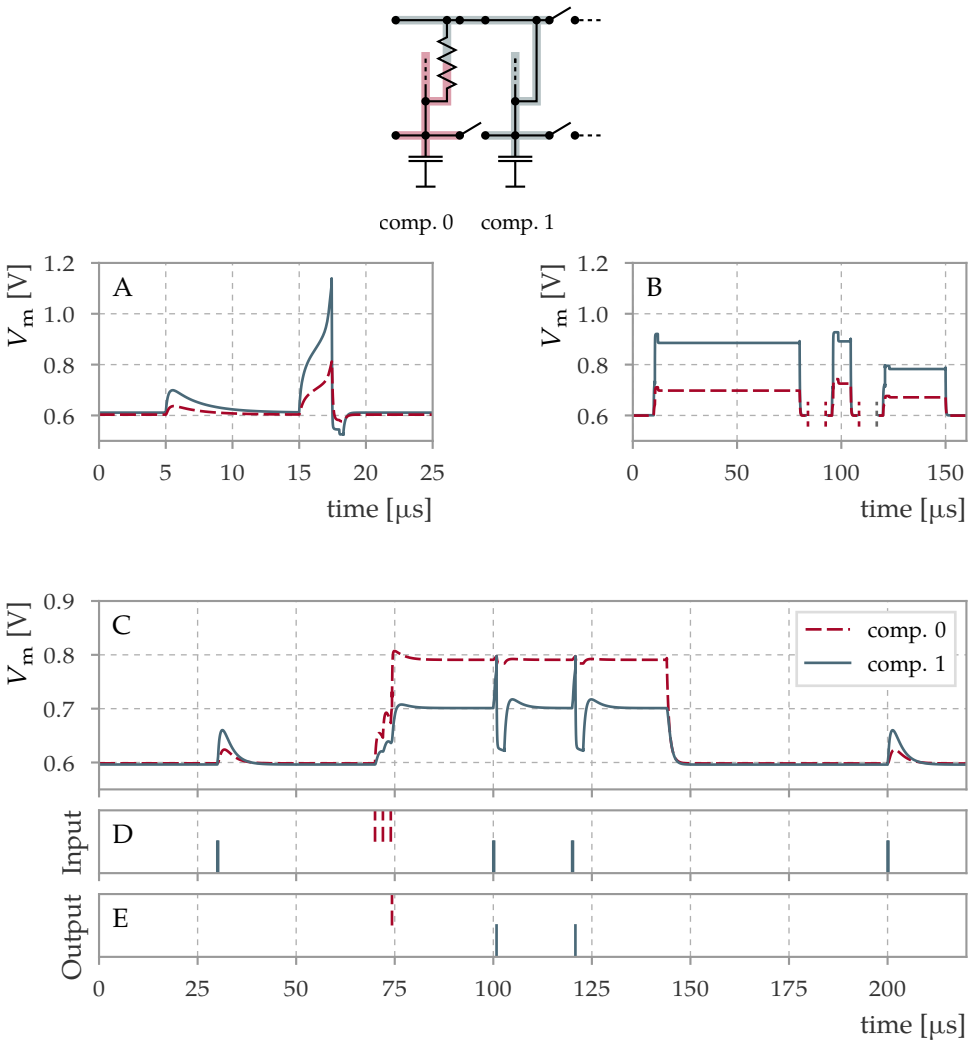


Figure 3.69: Top: sketch of the simulation setup. Two neighboring compartments are connected using the inter-compartment conductance. (Only the membrane capacitors and the inter-compartment connectivity is shown.) **A**: Compartment 1 (blue, solid line) is stimulated by synaptic and rectangular current input. Compartment 0 follows passively. **B**: Compartment 1 is stimulated by a short step current that triggers a plateau potential. Three separate simulations with varied parameters are shown: τ_{ref} is set to 70 µs, 9 µs and 30 µs. **C**: Input to compartment 1 (D) only leads to output spikes (E) if a plateau potential has been triggered before in compartment 0. Adapted from (Schemmel et al., 2017, Figure 8).

into the refractory period to allow the membrane potential of an active compartment to descend below the threshold after a plateau potential. (Without this feature, a plateau potential would be triggered at the end of the refractory period, and the resulting continuous up-state could only be stopped by sufficiently strong inhibition.)

Figure 3.69 shows a simulation of the basic functionality that can be achieved with

the described mechanism: Passive connection through the resistor (panel A) and the generation of plateau potentials (panel B) are combined to configure the circuit as a coincidence detector over long time scales (panel C). Only after a plateau potential is triggered in compartment 0 single events can cause a spike in compartment 1. In this case, the coincidence detection is clearly asymmetric, i.e., input 1-after-0 produces different results than 0-after-1. Note that the small peaks in compartment 1 in panel B and the overshoot in panel C in compartment 0 are not simulation artifacts but the effect of the input on the membrane which is kept at the reset potential by a high but finite conductance.

3.6.3 Backpropagation-activated calcium spike firing

Active and nonlinear currents in compartmental models

A multitude of active mechanisms exist in biological neurons that are assumed to provide the cells with their remarkable information processing capabilities. In fig. 3.70, a selection of these mechanisms is summarized by Antic et al. (2010). Figure 3.70 B3 shows the action potential that is generated in the axon initial segment and propagates along the axon, evoking post-synaptic potentials in neurons that are connected to it by synapses. This mechanism is modeled in abstract LIF models as a combination of threshold crossing and reset and an explicit generation of post-synaptic conductances or currents in synaptically linked neurons. Figure 3.70 B1 shows NMDA spikes, which were demonstrated to be generated in dendritic branches (Schiller et al. (2000)). The existence of these spikes triggered research interest because they could evoke long-term potentiation even without the emission of an action potential (Golding et al. (2002), Antic et al. (2010)). They are also shown to contribute significantly to neural excitability in simulation (Larkum et al. (2009)). Calcium spikes (fig. 3.70 B2) are generated in the apical trunk near the main bifurcation site (shown by Larkum et al. (1999)). They may provide a mechanism to allow events at distal synapses, which are strongly attenuated when propagated passively, to still have an effect on the generation of action potentials at the axon.

The term *backpropagation activated calcium spike* (BAC) (Larkum, 2013) describes the concept that the Ca^{2+} spikes are generated driven by the interaction between the calcium and action potential initiation zones over the main apical trunk and synaptic input at dendrites. Figure 3.73 B shows the supporting measurement: A layer 5 Wistar rat neocortical pyramidal neuron is identified in a slice and recorded at three sites in an in vitro experiment. The neuron responds nonlinearly and depending on the location of the stimulus within the cell. On the left of Figure 3.73 B, the reconstruction of the neuron with the pipette positions is shown. The traces on the right are colored corresponding to the pipette color. In the experiment, current is injected either in the dendrite in the shape of a post-synaptic current or at the soma as a step current. For dendritic stimulus, only the local membrane potential shows a significant response (Figure 3.73 B, top). The effect on the soma voltage is minimal. Soma stimulus that is sufficient to induce an action potential is shown in the center panel of Figure 3.73 B. The action potential travels back to the dendritic electrodes

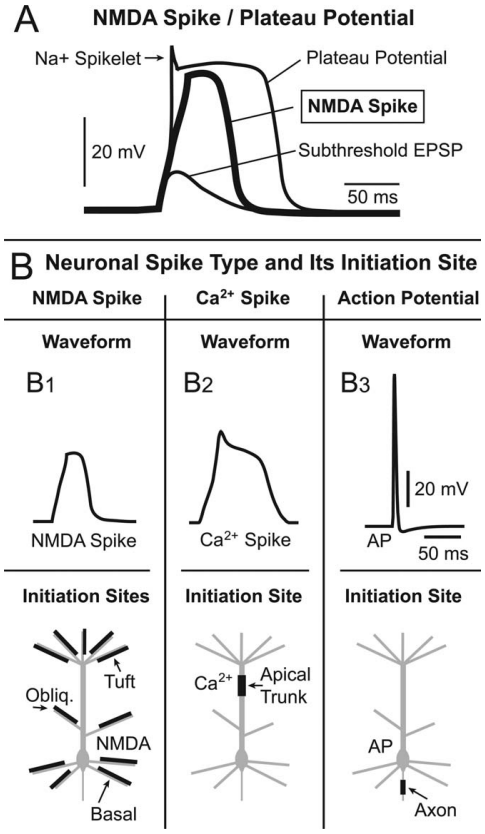


Figure 3.70: **A**: Plateau potentials are evoked by strong glutamatergic input and display a comparatively long depolarization on the order of 50 ms to 100 ms and an abrupt end. Multiple channels contribute to the generation of the plateau potential: The contribution of the NMDA channels is revealed by blocking sodium and calcium channels with tetrodotoxin (TTX) and Ca^{2+} (bold line). **B**: The regenerative potentials, also called spikes, are initiated in different compartments of the neuron. **B1**:NMDA spikes can be generated in apical tuft and oblique and in basal dendrites. Calcium spikes are observed in the *calcium spike initiation zone* in the apical trunk **B2**. The action potential is initiated in the axon initial segment (**B3**) and propagates along the axon as well as back into the dendritic tree (not shown) Waters et al. (2005). The duration of the action potential is smaller than that of the dendritic regenerative potentials shown in B1 and B2. The image is reproduced with permission from Antic et al. (2010).

with reduced amplitude and increased duration (black and red curves). The bottom panel shows the result of coincident application of the two stimuli. A Ca^{2+} spike occurs in the distal dendrite, and three action potentials are generated at the soma, the second two after the stimulus currents have stopped. This demonstrates the non-linear combination of somatic and dendritic stimulus in the neuron.

Circuit simulation

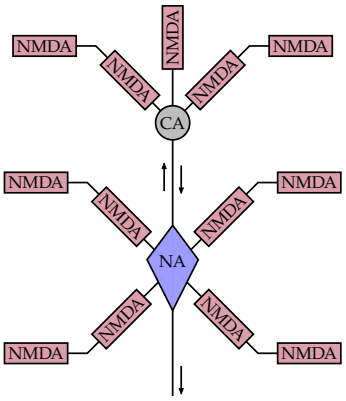


Figure 3.71: Schematic representation of a L5 pyramidal neuron. Local NMDA sites connect to the Ca^{2+} or Na spike initiation zones. The initiation zones interact through active signal propagation. Redrawn after (Larkum et al., 2009, Figure 4 H).

The aim of implementing the hardware features described above is to reproduce core aspects of information processing within individual neurons that is enabled by active mechanisms. Larkum et al. (2009) show that NMDA spikes occur in fine distal tuft dendrites. They summarize that integration of information relies on the initiation of NMDA spike at distal tuft branches, Ca^{2+} spike initiation near the main bifurcation of the neuron and sodium spike initiation at the axon hillock. Further, Larkum et al. (2009) hypothesize that dendrites form subunits which integrate information locally and forward their output to the spike initiation zones (fig. 3.71).

Figure 3.72 shows the envisaged configuration of this connectivity principle in a full-scale successor of DLS3 with two rows of neuron circuits. A sodium (NA) and a calcium (CA) compartment are connected to each other and to a number of NMDA compartments using the inter-compartment conductance circuits. Here integration of synaptic input will occur in multiple stages, first localized in the individual NMDA compartments then in CA and NA, all of which are coupled over the connecting resistors.

Figure 3.73 shows a simulation of a circuit configuration that mimics the coincidence detection by active mechanisms in a pyramidal neuron (fig. 3.73 B, Larkum (2013)): A dendritic stimulus (in compartment 0) alone does not evoke spiking and has only a small effect on the soma compartment (panels C, D). A step current stimulus at the soma (compartment 3) causes a single soma spike and weak response in the

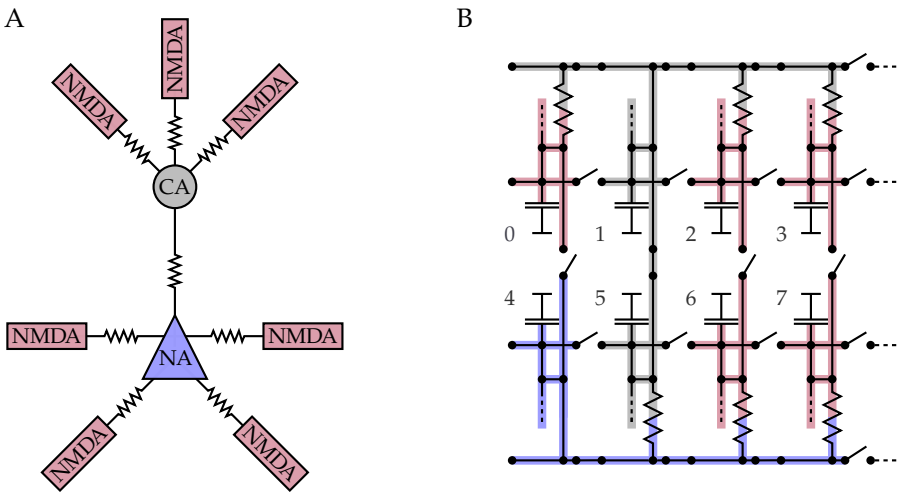


Figure 3.72: **A**: Circuit concept that approximates fig. 3.71. **B**: Sketch illustrating the configuration that realizes A. Compartment 4 acts as the NA compartment, compartments 1 and 5 emulate CA and NMDA compartments are attached to either via inter-compartment conductance. (Note that the number of NMDA compartments differs in A and B, but further compartments could be attached to compartment 4 by continuing the pattern.) Redrawn after (Schemmel et al., 2017, Figure 7).

other compartments (panels E and F). Both inputs together suffice to trigger plateau potentials in compartments 0 – 2 (panels G and H) which in turn cause a burst in the soma compartment. This high-level behavior is equivalent to the observation in panel B.

Some differences have to be highlighted as well: The center panel in fig. 3.73 B shows a time difference between the maxima of the black and red trace, indicating a propagation of the excited membrane state in time. Consequently, the synaptic and current stimulus in the bottom panel do not overlap but still trigger a burst in the soma. This is not the case for the hardware simulation. The signal propagation between compartment 0 and 3 is nearly instantaneous on the time scale of the experiment, which is why the input currents in panel H must overlap. A second difference is the precise duration of triggered plateau potentials which, in the biological reference, depend on the strength of the stimulus (cf. (Antic et al., 2010, Figure 2 G)). In the chip implementation, the duration of each plateau potential is fixed to a precise value.

Stability of the configuration

The circuit parameters for fig. 3.73 were selected purely on a functional level, to achieve the coincidence detection displayed by the biological reference. The setup is used to investigate how distortions of individual components affect a complex use case involving multiple interacting neuron compartments. Two distortion mechanisms are examined: The dependence of the simulated circuit on temperature variation and the stability with respect to small parameter deviations.

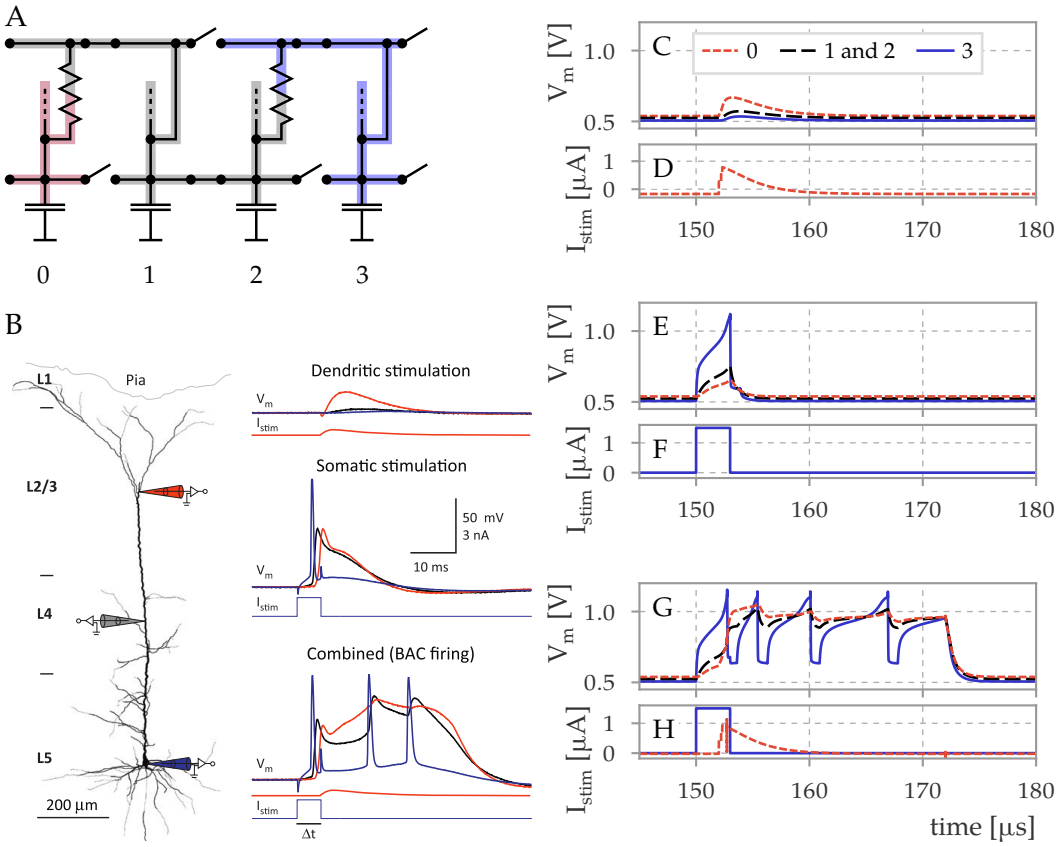


Figure 3.73: **A**: Schematic representation of the interconnection scheme used for the transistor-level simulations in C–H. The colors correspond to the recorded data and to the spatial locations in the neuron (B) that the circuit configuration emulates. Compartment 0 corresponds to the distal dendrite (red), compartment 3 (blue) to the soma and compartments 1 and 2 are in place as one interconnecting segment. Note that compartments 1 and 2 are connected directly, i.e., without using the tunable resistor. **B**: In vitro measurement of backpropagation activated calcium firing. Detailed description in text. The figure is used with permission from Larkum (2013), where it is adapted from Larkum et al. (1999). **C**: Circuit response to synaptic stimulus at compartment 0. Only a small deflection of the membrane voltage occurs at compartment 3. **D**: Input current to compartment 0, caused by synaptic events, that corresponds to the voltages in C. **E**: Response to a current input to compartment 3 only. Contrary to the measurement (B, center), the depolarization in the other compartments does not last significantly longer than in the stimulated compartment. **F**: Current input to compartment 3 only that is tuned to induce a single spike. **G**: Response to synchronous stimulus. Plateau potentials are triggered in compartments 0 and 2 which repeatedly pull the soma voltage above threshold, resulting in a burst. **H**: The stimuli shown in D and F are combined. The simulations were performed by Laura Kriener. Adapted from (Schemmel et al., 2017, Figure 8).

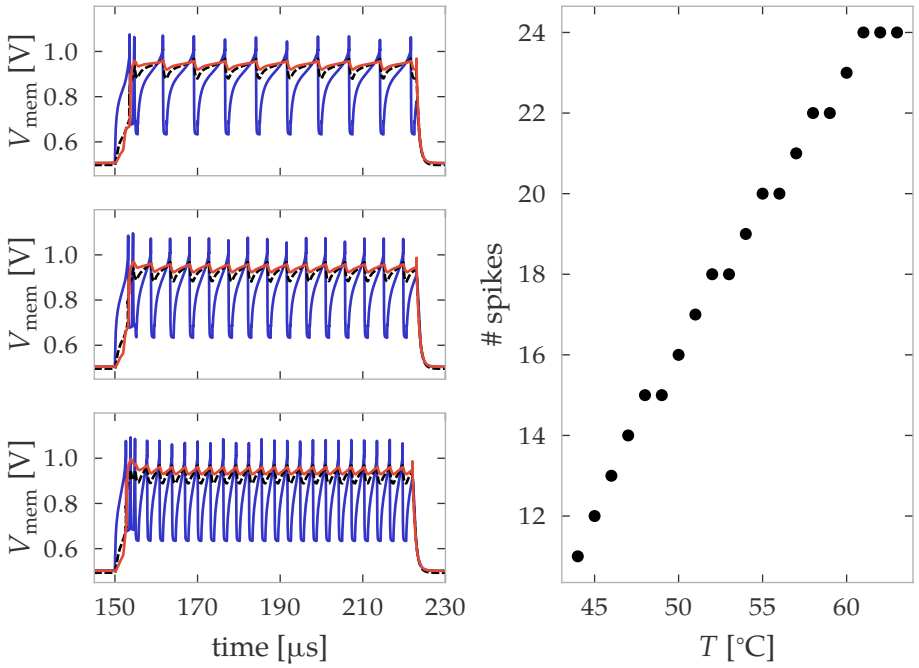


Figure 3.74: Left: A version of the simulation shown in fig. 3.73 with longer plateau potential durations is shown. The simulation temperature is varied with settings 44 °C, 50 °C and 63 °C, from top to bottom. The right panel shows the number of spikes in the "soma" compartment (blue). Beyond 44 °C and 63 °C the firing behavior is not correctly reproduced. Adapted from (Kriener, 2017, Figure 61)

Figure 3.74 shows the influence of the temperature parameter which is set in the simulation on the behavior of the circuit. A longer duration of the plateau potential is configured so that the number of spikes provides a more detailed quantification of the behavior. Over the range of 19 K the number of spikes within the burst doubles; outside of the temperature range, the circuit does not exhibit the desired behavior of generating a burst, or generated a burst for soma input alone. The most prominent difference in the individual traces is the rising speed of the "soma" compartment during a spike. Surprisingly, the variation of individual currents that contribute to the evolution of the membrane potential is small: the exponential term has the strongest variation with approximately ten percent difference between the 44 °C and 63 °C case (Kriener, 2017, Figure 62). But because the exponential and leak current have different signs and are nearly equal close to V_T, the relative variation of the total current is significantly higher, causing the large variation in firing frequency.

Figure 3.75 shows the effect of parameter variation on the number of spikes within a burst. Random values drawn from a uniform distribution ranging from −1 to 1 LSB were added to the analog parameter values of the original simulation setup. The amount of variation is chosen to represent an estimate of the imprecision

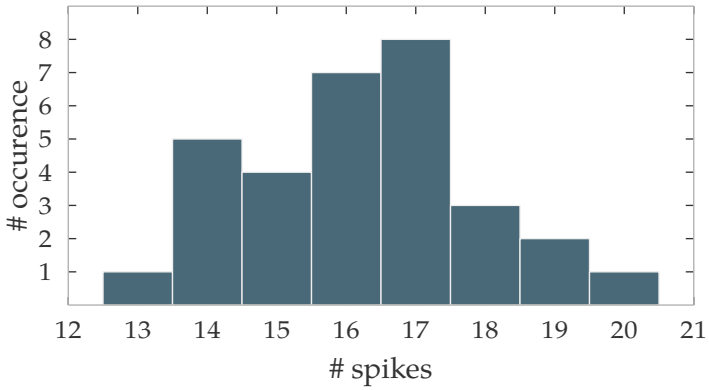


Figure 3.75: Variation of the number of spikes in the simulation shown in fig. 3.74 if the analog parameters are varied statistically. A uniform distribution of ± 1 least significant bit (LSB) is added to the original parameters for each of the 31 repetitions of the simulation. Taken from (Kriener, 2017, Figure 63)

due to discretization as well as dynamic effects within the parameter memory. In all trials the neuron shows the backpropagation activated calcium spike (BAC) firing behavior. The number of spikes within the burst varies by a factor of up to 1.5.

The initially selected parameters for the simulation in fig. 3.74 were not optimized for robustness. Neither is the BAC firing use case a static one: The response of the neuron varies continuously with the input strength ((Larkum, 2013, Figure 2 c)). The above simulations provide a quantitative estimate of the variation of high-level behavior under two important distortion mechanisms. Temperature, in particular, can vary in an unpredictable manner due to external effects or changing on-chip activity of certain components, for example of the plasticity processor. The amount of expected variation suggests that a dynamic control of parameters may be required if an experiment similar to the one presented above is intended to be precisely repeatable in strongly varying environment conditions.

3.6.4 Summary

In this section, the inter-compartment circuitry of the DLS3 neuron is characterized in simulation. A complex, biologically motivated use case is mapped to the hardware substrate and the resulting dynamics is analyzed with respect to variations of temperature and analog parameters.

The new inter-compartment and reset components can be used to implement an asymmetric, two-compartment coincidence detector by triggering a plateau potential in one of the compartments (fig. 3.69). A more complex case, inspired by the biological reference of BAC firing is demonstrated to reproduce the triggering of single spikes or bursts depending on the coincidence of somatic and dendritic stimulus. This is accomplished by a four-compartment setup in which one compart-

ment, designated as soma, resets to a low potential for a short duration and the other compartments are configured to produce plateau potentials. The stability of the circuit configuration is tested with respect to variations in temperature and analog parameters. The spiking behavior itself is stable within a range of 19 K and $\pm$1 LSB parameter variation.

3.7 Chip measurements

3.7.1 Adaptation and exponential term

In this section, initial measurements of the produced DLS3 prototype chip are presented. The measurements were performed in collaboration with Yannik Stradmann, Sebastian Billaudelle, Gerd Kiene and Syed Ahmed Aamir on DLS 3.0 chip number 4. All parameters were tuned directly without any prior calibration.

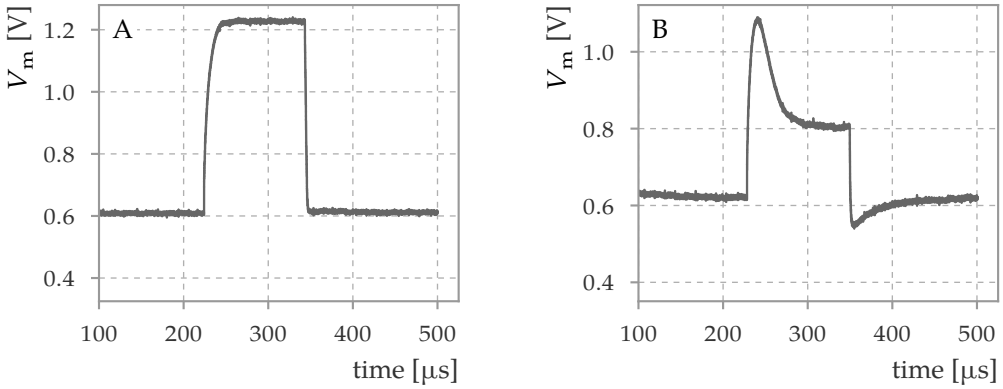


Figure 3.76: Chip measurement of adaptation term. **A**: Disabled adaptation. **B**: Enabled adaptation.

Figure 3.76 shows the function of the adaptation term. The *en_mem_off* setting is enabled and disabled again to implement a step current stimulus onto the membrane. The adaptation term is disabled in panel A and enabled in panel B. The adaptation current pulls the membrane down after the initial peak. After the end of the stimulus, hyperpolarization occurs while the adaptation current returns to its resting state. The strength of the adaptation during and after the current input seemingly varies; the cause for that is not known but is suspected to be in the non-linear behavior of the leak current and the adaptation output OTA. Figure 3.77 demonstrates the functionality of the exponential term. As in fig. 3.76, a current stimulus is applied. The change in the direction of the curvature of the membrane potential course is caused by the enabled exponential term. A reset after a triggered spike is seen after the peak; The kink at the beginning of the refractory period is caused by the end of the current stimulus.

3.7.2 Inter-compartment conductance

Figure 3.78 shows an initial demonstration of inter-compartment connectivity. In panel A, neuron compartments 0, 1 and 2 are connected directly using the *en_right* switch. In compartment 0, five neuron resets are triggered directly in the digital neuron back end. Compartment 2 reacts almost identically due to the near-sort-circuit connection. Panel B shows the same setup, but with one inter-compartment connection replaced by the inter-compartment conductance. The signal in compartment

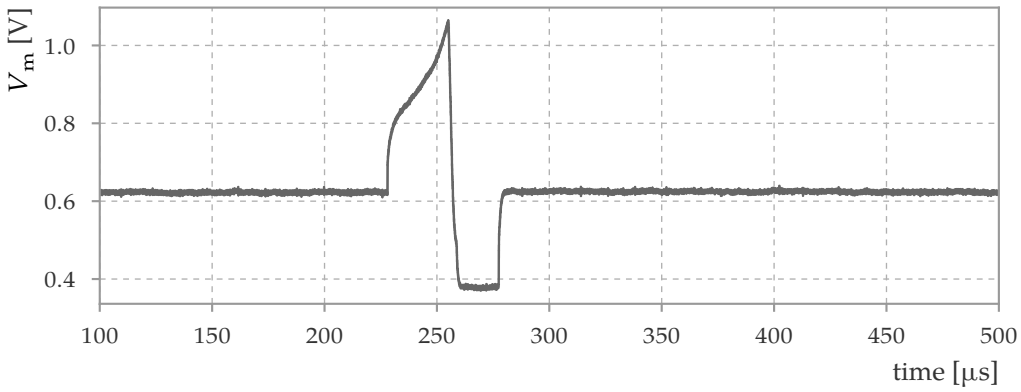


Figure 3.77: Chip measurement with enabled exponential term.

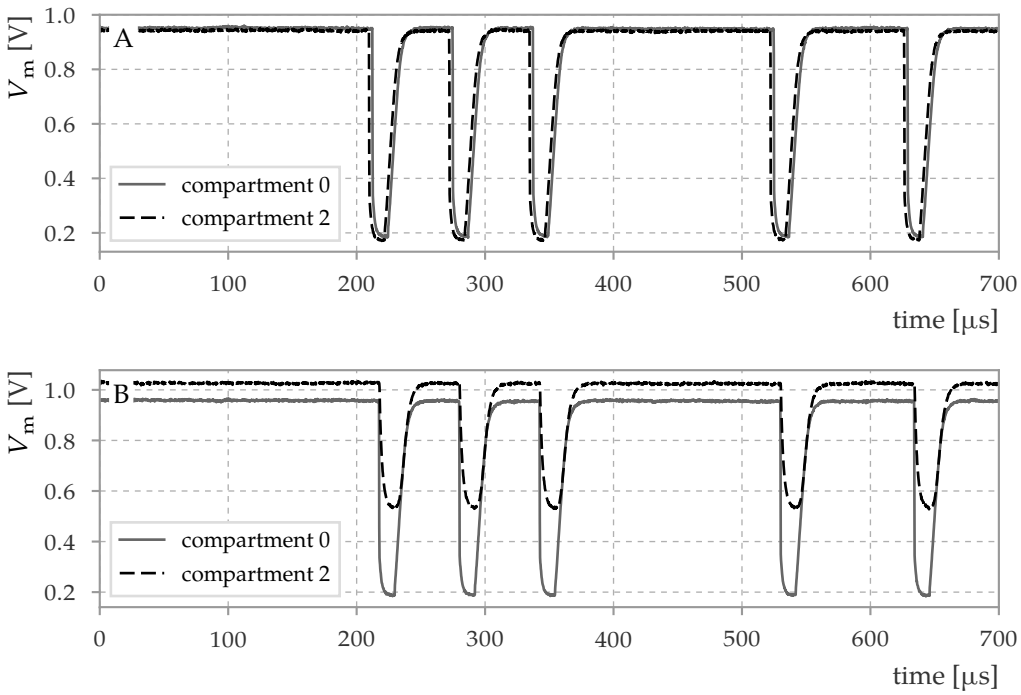


Figure 3.78: Demonstration of inter-compartment connectivity. **A**: Three directly connected compartments. **B**: Three compartments connected with one short-circuit switch and one inter-compartment conductance (*i_bias_nmda* = 100 DAC, *ib_nmda_div4* = `True`). The timing pattern of the five resets is chosen to be easily identifiable in the analog trace which also contains reset-like activity during the initialization of the chip. Two repetitions of the same experiment are overlaid with recording from compartment 0 and from compartment 2.

2 is attenuated due to the resistance. The resting value of both compartments is different in panel B because both compartments have different resting potentials due to mismatch. Contrary to A, where the compartments are connected with a large

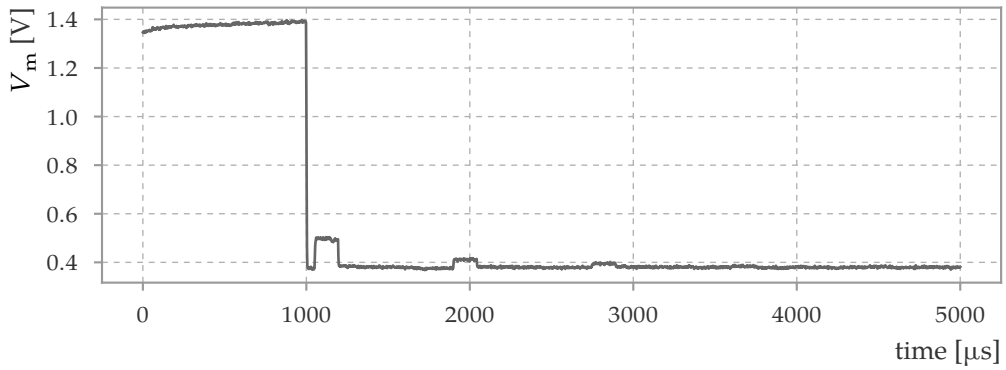


Figure 3.79: Undesired crosstalk from capacitive memory to leak/reset circuit.

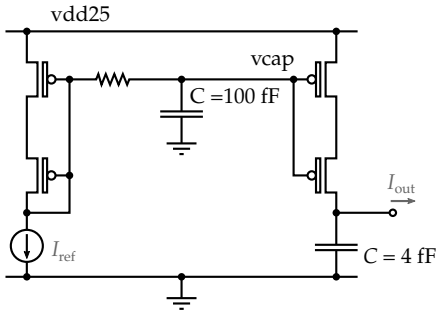


Figure 3.80: Modified model of capacitive memory output stage. (See fig. 3.4 for the original version.)

conductance, this effect is visible due to the smaller inter-compartment conductance.

3.7.3 Erroneous capacitive memory interface

Figure 3.79 shows unintended behavior of the circuit that was discovered during testing. A reset to a high potential is triggered, which ends at 1000 µs. Additionally, a constant offset current is applied (*en_mem_off* = `True`) to be able to see changes in the conductance of the OTA. No additional input is provided. The step responses at 1100 µs, 2000 µs are observed, and their start and end coincides with the voltage ramp within the capacitive memory (not shown). Additionally, the rising edge of the step pulses can be shifted in time by changing the digital code for *i_bias_leak_sd* (which is set to 800 DAC for the above measurement). The hypothesized source for the observed behavior is that after switching of the bias currents in the leak/reset term, the voltage on the input line rises quickly. Capacitive coupling then changes the voltage on the storage capacitor, which is updated to its target value during subsequent updates by the logic of the capacitive memory. When the leak/reset OTA switches back, the crosstalk now changes the voltage on the storage capacitor in the opposite direction, leading to a mismatch in the output current. This mismatch is again removed by following voltage updates within the capacitive memory.

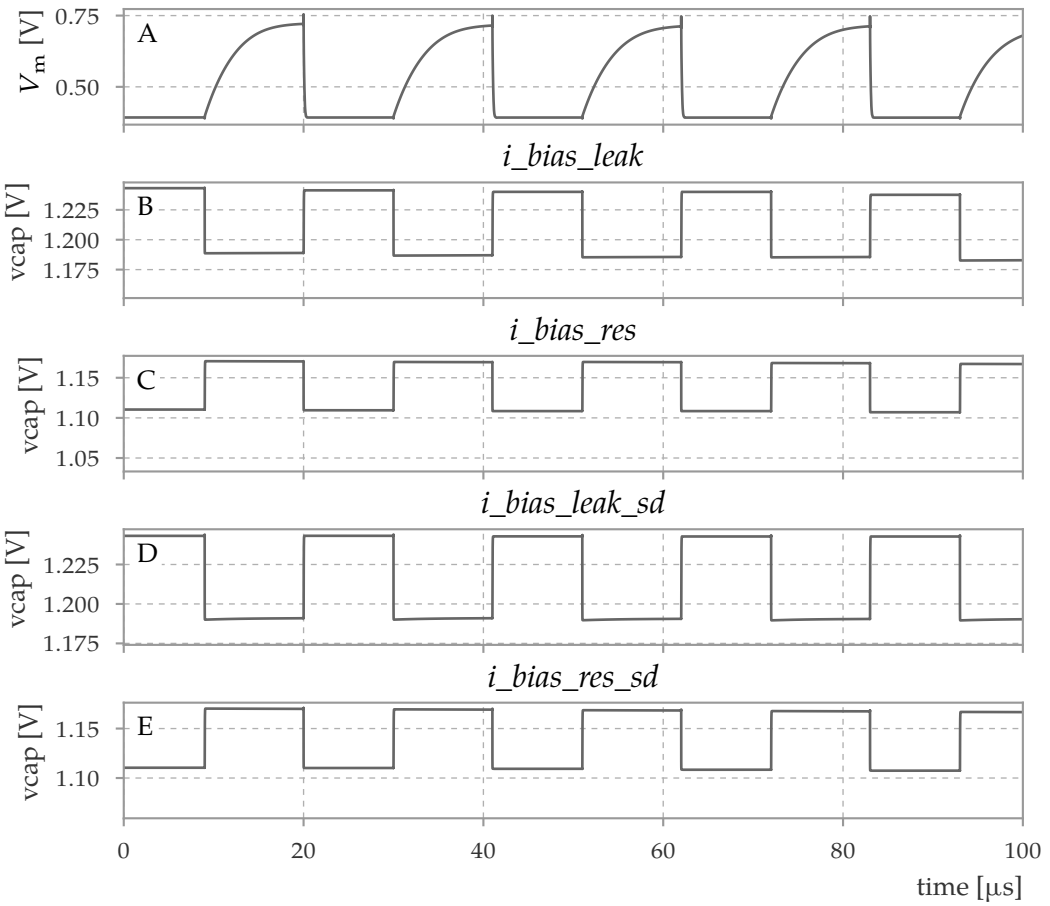


Figure 3.81: Simulation of continuously firing neuron with the modified current cell model (shown in fig. 3.80). **A**: Membrane potential. **B**–**E**: Voltage *vcap* (indicated in fig. 3.80) for each of the indicated bias currents. The values set for this simulation are: *i_bias_leak* = 0.7 µA *i_bias_leak_sd* = 0.7 µA *i_bias_res* = 1 µA *i_bias_res_sd* = 1 µA.

To quantify the effect, a simulation is performed with a modified model of a current cell of the capacitive memory, as shown in fig. 3.80. The storage capacitance is included directly, as is the resistor which emulates the slow charging of the capacitor by infrequent updates. The results (fig. 3.81) confirm that significant crosstalk on the order of 50 mV occurs. A similar problem also affects the *i_mem_off* current when it is switched during operation (not shown).

Table 3.6: Summary of the quantity ranges achieved in Monte-Carlo simulations. Minimum and maximum bias currents for these quantities are 20 nA and 1 µA. The errors denote the standard deviation of the sample. τ_m is calculated with a nominal capacitance of 2.36 pF, *highs_leak* = `False`. Low time constants should be achieved by using a smaller setting for C_m. ΔV_w: simulation by Syed Ahmed Aamir (Aamir et al., 2017a). For $I_\mathrm{max,exp}$, the current distribution with minimum and maximum weight setting at $V_\mathrm{m} = 1.01$ V is given. The simulation result for maximal synaptic currents is shown in fig. A.8. The reference values (table 3.5) are translated into the hardware domain using an acceleration factor $\alpha_t = 10^3$ (section 1.7). Typical values for α_V are between two and ten. The reference hardware values for a are calculated with a membrane hardware capacitance of 2.36 pF.

Quantity	unit	simulation		target (table 3.5)	
		min	max	min	max
τ_m	µs	1.2 ± 1.6	108 ± 10	7 (1.4)	50
g_l	nS	22 ± 2	2700 ± 800		
τ_syn	µs	0.77 ± 0.05	6.7 ± 1.3	1	100
$\mathrm{abs}(a)$	nS	23 ± 4	7700 ± 100	0	470
τ_w	µs	14.4 ± 3	336 ± 130	16	600
ΔV_w	mV	1.2 ± 0.05	518 ± 12		
Δ_T	mV	159 ± 8	159 ± 8	$0.8\,\alpha_V$	$5.5\,\alpha_V$
$I_\mathrm{max,exp}$	nA	150 ± 60	1100 ± 400		
$\max\left(I_\mathrm{syn,exc}\right)$	µA	–	1.15 ± 0.15		
$\max\left(I_\mathrm{syn,inh}\right)$	µA	–	1.3 ± 0.2		

3.8 Summary

A simulation-based calibration procedure for all essential parameters of the DLS3 neuron is presented that is based on a statistical transistor model of the chip manufacturing process (section 3.3). The simulations yield a detailed characterization of the circuit performance, including the expected parameter ranges (see table 3.6). The range for the membrane time constant exceeds the requirements even without using the capacitance switching feature. However, this large range entails the disadvantage of a strong parameter dependence of τ_m for certain values of the control bias currents (figs. 3.31 and 3.32). It is advised to use smaller capacitance values to achieve shorter membrane time constants. Further, the saturation behavior of the leak OTA suggests the use of lower leak and reset potentials (fig. 3.30). The synaptic time constant only covers the lower range of the surveyed computational models. However, the configuration range of nearly one order of magnitude still allows to implement different time scales, as used for example in section 2.3.1 and table A.1, by appropriately changing the acceleration factor. The adaptation output transconductance a covers a larger range than previous implementations (Millner (2012)) by including a switchable sign. The maximum possible hardware value is significantly

larger than the requirement. The larger required values reach into the steep region of the tuning curve in fig. 3.38 B, which can reduce the available resolution. The parameter that corresponds to the AdEx spike-triggered adaptation parameter b is the change of the voltage on the adaptation capacitance, ΔV_w. Particular to the implementation in DLS3 and HICANN, ΔV_w is proportional to b/a, so it diverges if a non-zero b is required for zero or very small a. Nevertheless, the hardware range of the parameter is large given by the combined adjustability of the length and strength of the adaptation pulse: A ΔV_w of more than 500 mV is sufficient to exploit the full input range of the adaptation OTA.

The exponential parameter Δ_T is not adjustable on the DLS3 prototype chip. The target values depend on the voltage scaling factor α_V, which is typically between two and ten. With these values, the intrinsic Δ_T of the circuit is larger than what is required by typical modeling studies (table 3.6). It is expected that the exponential term is extended in future revisions to allow for adjustable V_T and Δ_T. The current DAC that controls the exponential strength is characterized by the maximal exponential current at a fixed voltage in table 3.6. The maximal synaptic current is not part of table 3.5 but is given as important quantity nonetheless.

The calibration procedures presented in this chapter generally use internal signals of the simulated circuit to reduce the introduction of a measurement bias by indirect calculation methods of relevant quantities. For the most important calibration procedures, realistic methods were considered: The presented calibration of the synaptic input offset uses the membrane potential. For the calibration of synaptic weights, a current-based and a realistic, V_m-based method were compared. For the calibration of a, a method that uses the membrane voltage is compared to the more exact current-based method. The calibration of the resting potential is accomplished as a last step by canceling all remaining offset currents by the offset current parameter *i_mem_off*. On-chip this is anticipated to be implemented using bisection of the membrane potential, by off-chip measurements or by using the plasticity processing unit (PPU) directly; to save time in simulation, the bisection is not implemented but the required current is recorded directly in one preceding simulation. The level of realism for the individual calibration algorithms is summarized in table 3.7. The implementation of the integrated test environment and of the calibration methods described above led to an implicit verification of the interoperability of multiple components within the of the chip. The functionality of the full circuit was exemplified in test cases derived from existing modeling studies (section 3.5). In addition, explicit verification of selected components was performed that helped uncover multiple errors, most of which were remedied before the final submission (section 3.4). The initial functionality of the chip is shown in exemplary measurements (section 3.7), which also led to the discovery of an issue that was not discovered before production. This issue of capacitive crosstalk onto the capacitive memory storage was adapted into simulation by updating the model of capacitive memory (section 3.7.3). The methodical completion of the simulations described in this chapter, especially the Monte-Carlo calibration of the neuron circuits, provides a much more detailed, application-oriented pre-production characterization of the

Table 3.7: Reliance on internal circuit signals of the presented calibration methods.

Quantity	type	uses internal signals	comment
Synaptic input offset	V_m-based	no	bisection of V_m
w_syn	V_m-based	no	requires $V_\mathrm{m}(t)$, fitting and integration
w_syn	I_syn-based	yes	
τ_syn	current-based	yes	requires $I_\mathrm{syn}(t)$, fitting (current based, transferable to V_syn)
τ_m	current-based	yes	
Resting potential	current-based	yes	intended as bisection of V_m by PPU
Spike threshold	voltage-based	yes	
Reset current	current-based	yes	
a	current-based	yes	
a	V_m-based	no	
b	ideal	no	
Exponential current	current-based	yes	

device as compared to previous systems (e.g. Millner (2012)).

Chapter 4

Conclusion and outlook

One goal behind the development of large-scale accelerated neuromorphic devices is the creation of a fast and scalable substrate to investigate the computational properties of spiking neural networks. The advantages of speed and scalability are achieved in the analyzed devices by allowing limitations in the form of circuit variability and constraints on connectivity and bandwidth. The trade-off between power-efficiency, device size and precision is accepted, as the networks targeted for emulation by the device are precisely those that are robust against moderate levels of distortion. The result of this trade-off for applications is analyzed at the network and single-neuron level within the presented thesis.

In the first part (chapter 2), the sensitivity of existing spiking network models to distortions expected from the emulation on neuromorphic hardware is studied. It is investigated how a given network can be made robust against these types of distortions. As part of a larger study (Petrovici et al., 2014), two different network models that are derived from existing studies are used as case examples: The first network is a chain of neuron groups with feed-forward inhibition, which is used in the original publication (Kremkow et al., 2010) to study the propagation of a synchronous spike volley in dependence on the properties of local inhibition (section 2.3.1). The second network (Muller and Destexhe, 2012) is a random cortical network with distance-dependent connectivity that displays self-sustained, asynchronous and irregular activity (section 2.3.2). As a reference neuromorphic hardware implementation, the BrainScaleS system is used. Several categories of hardware-induced distortions are studied: The lack of certain features required by the model, such as adjustable axonal delays, the fixed-pattern variability of synaptic weights and an incomplete mapping of the network to the hardware, resulting in the loss of synapses. Additionally, dynamic effects introduced by the internal spike communication infrastructure are taken into account by using the ESS, a software simulation of the device.

For each network model, performance metrics were created that are used to quantify the functionality of the undistorted network. The level of distortion is assessed in simulation and compensation strategies are evaluated that counteract the effects. These compensation strategies are tested in a final step in simulation using the ESS and all distortion mechanisms at once. The resulting compensation mechanisms

reduce the effects that are introduced by the distortion mechanisms. They are specific to each network and range from iterative tuning of network parameters (fig. 2.17) to a re-distribution of background input to the network (section 2.3.1). Some of the methods show a level of generalization, such as the iterative compensation of heterogeneous firing, which can be applied to weight variation and synapse loss. The collection of methods is regarded as a toolbox for neuromorphic modelers who encounter problems similar to the investigated distortion mechanisms in related network models.

The analog components of the hardware system investigated in chapter 2 are idealized and the distortion models are set on an abstract level, such as a normal distribution for the variation of synaptic weights. In the second part of this thesis (chapter 3), the effect of transistor mismatch and of the dynamics of the implemented circuits is assessed for the DLS3 neuromorphic chip. This investigation complements the analysis in chapter 2 where the neuron was modeled as an ideal, accelerated version of the mathematical AdEx equation. Each component of the analog circuit is characterized and calibration methods are implemented and tested to evaluate the parameter range in which the circuits can be tuned. For this, transistor-level Monte-Carlo simulations of the full neuron circuit are employed that use manufacturer-provided data on the expected variability of the produced circuits (section 3.3). In addition to the implicit verification by the circuit simulations during the development of the calibration procedures, an explicit verification of essential components was performed before tape-out of the design(section 3.4) and led to improvements of the circuit. Multiple use cases derived from biological (section 3.6) and abstract (section 3.5.2) network modeling are investigated in simulation.

The neuron shows a high tunability (table 3.6). In particular, it exceeds the range covered by reference studies outlined in section 3.1.1 for the membrane time constant and sub-threshold adaptation parameter a. The range covered by the adaptation time constant does not fully cover the reference range. The synaptic time constants only cover fast synaptic current fall times. The exponential circuit that was included in the DLS3 prototype chip does not include an adjustable slope factor Δ_T. With the parameterization as-is it was possible to reproduce a biologically-inspired use case of backpropagation activated calcium spike in a multi-compartment simulation setup.

The full-neuron simulation setup proved a valuable addition to the pre-production verification of the neuron circuit, helping to uncover errors which were remedied before chip tape-out (section 3.4.5). The Python-based interface (section 3.1.3) proved to be a valuable tool for interfacing complex simulation setups including multiple neuron compartments as well as in the development of calibration algorithms which require the iterative set-up of individual simulations (e.g. section 3.3.1). The simulation time is a limiting factor of this approach, being on the order of minutes for microseconds of hardware time. Future work should focus on extending the library of use-cases that are used as full-neuron tests as well as automating the testing procedure.

The calibration methods should be implemented on the DLS3 prototype chip.

It is advised to implement essential calibration algorithms for the synaptic offset current (section 3.3.1), the resting potential (section 3.3.3) and the reset current (section 3.3.4) in a first step to allow initial experiments before a full calibration is available. It should be tested whether the resting potential calibration (section 3.3.3) can be performed on the PPU, which is expected to increase the experiment rate.

Some calibration procedures described in section 3.3 use internal signals to minimize the measurement bias for circuit characterization and increase simulation speed (table 3.7). The affected calibrations should be re-implemented to V_{m}-based alternatives. In the case of the synaptic time constants this means using the read-out of the synaptic input line instead of the synaptic input current. The calibration for maximal reset current should be adapted, either by using a reference current (*i_mem_off*) or an external current measurement. In the long term, a circuit-level solution that reduces the current saturation (section 3.3.4), would be preferred.

A number of possible improvements for the circuit implementation is collected from the simulations in chapter 3:

1. When disabling the synaptic input using the switches *en_syn_i_[exc|inh]*, the power to the output OTA can be disabled automatically to prevent unintended leakage (see, e.g., fig. 3.27).

2. The saturation current of the leak/reset OTA at high membrane potentials decreases with high *v_leak* and *v_reset* (fig. 3.30). Removing this effect would improve usability for uncalibrated or largely uncalibrated operation (section 3.5.2) and remove the requirement of one calibration step (section 3.3.4). This would presumably also improve the common-mode dependency of the membrane time constant (fig. 3.30).

3. The switching of input currents in the leak term should be improved to remove coupling to the analog parameter cell (section 3.7.3).

4. The problem present in the switching of current cells described in section 3.7.3 most certainly also affects the input pulse of spike-triggered adaptation, and should be addressed by modifying the circuit.

5. The spike-triggered adaptation mechanism does not always limit the maximum voltage on C_w to 1.2 V in all operating conditions (fig. 3.57), as is required due to the attached circuits that use thin-oxide transistors. The voltage-limiting mechanism (M_1 and M_2 in fig. 3.22) should be re-evaluated for the next chip revision.

6. The utilized thin-oxide transmission gates in the neuron design only function correctly for switched voltages below 1.2 V. Currently, attached circuits can produce higher voltages during operation, including the synaptic input, leak/reset term, adaptation term and offset current input. A consistent limitation of the respective voltages to the required level would improve the usability of the circuit by removing the reliance on the user to ensure proper operating range.

7. The parameter dependency of the sub-threshold adaptation parameter *a* is not smooth (fig. 3.38) and the utilized OTA implementation shows a strong corner dependency (fig. 3.40). This is the case because an OTA implementation from a previous chip is used; after the verification of the DLS3 leak OTA it may be considered to use its design in place, as the corner dependency is addressed there.

8. The switching of voltage cells in the leak/reset OTA should be verified on the prototype chip, quantifying the amount of remaining cross-talk between the *v_reset* and *v_leak* cells during neuron reset (section 3.4.4).

9. The exponential term should be completed by including an adjustable Δ_T.

10. The threshold parameter V_T is currently adjusted using a 3-bit current DAC. The level of mismatch makes it necessary to measure each of the output transistors individually (fig. 3.47 D) instead of relying on a binary switching scheme. If this tuning mechanism for V_T is kept in further chip revisions, the mismatch may be reduced to remove measurement steps that are required during calibration.

11. The bypass mode issue (fig. 3.56) should be addressed. Measurements should verify whether increasing the synaptic strength by providing *v_bdac* externally is sufficient to allow for a reliable function of the bypass. If not, the bypass circuit should be adapted or an alternative created that allows to detect incoming spikes without reliance on the analog configuration. This issue is not severe for small prototype chips where the number of neurons is limited and analog readout is readily available. For large-scale devices which include non-trivial routing, features that allow stepwise tracing of spike signal paths are very helpful to improve the development speed of control software.

12. The per-neuron adjustable offset current (*i_mem_off*) turned out to be a useful feature for calibration and experiment control. Unfortunately, the issue of switching current cells (section 3.7.3) also affects the switching of the offset current (*en_mem_off*) during an experiment. Removing this limitation at the circuit level would allow to implement the stimulus paradigm of step currents (e.g. Markram et al., 2004) directly on-chip. This would be very useful, e.g. for direct fitting of hardware parameters to reference biological data.

13. In the current prototype, only short synaptic time constants are implemented (table 3.6). One method to allow for long synaptic time constants while re-using existing circuitry is to use the adaptation term as a second stage of integration for synaptic input currents. This can be accomplished by optionally redirecting synaptic events onto C_w and removing the coupling of C_w to C_m. Either the synaptic events directly or the output of the synaptic input can be used to drive C_w in this scenario. If the synaptic output is used, rise times of the synaptic current can be controlled as well. The adaptation feature can not be used for the

neuron compartment if this proposal is implemented. Multiple compartments can be interconnected to use the propose feature and adaptation in the same logical neuron.

The produced DLS3 chip offers novel functionality that should be exploited in experiments. The configurable reset conductance makes it a candidate to implement LIF sampling, as described in section 3.5.2. The inter-compartment and plateau-potential functionality (section 3.6) should be used to extend the BAC firing example (section 3.6.3) to a functional network. One possible direction is the investigation of the influence of active components in multi-compartment models for structural plasticity, as described in Schemmel et al. (2017).

Neuromorphic hardware is a promising approach to investigate the computational capabilities of spiking neural networks, and to create a substrate on which this type of networks can be efficiently emulated. To continue the joint development of network architectures and hardware systems, it is necessary to do both: Implement proof-of-concept experiments on the small DLS3 prototype system that uses the novel functionality. Findings from these experiments should lead to goal-directed improvements of hardware implementation in future chip revisions. At the same time, the results from prototype chips should be assessed with respect to the application to large-scale spiking systems, identifying important characteristics from device measurements and extrapolating them to large networks, using the analysis workflow that was presented in the first part of this thesis.

List of coauthored publications

This section lists the coauthored publications, clarifying which are incorporated in this document, and to what extent.

1. Petrovici, M. A., Vogginger, B., Müller, P., Breitwieser, O., Lundqvist, M., Muller, L., Ehrlich, M., Destexhe, A., Lansner, A., Schüffny, R., et al. (2014). Characterization and compensation of network-level anomalies in mixed-signal neuromorphic modeling platforms. *PloS one*, 9(10):e108590

   Part of the research in the publication were conducted by the author and the author's contribution is summarized in more detail in this document in chapter 2.

2. Aamir, S. A., Müller, P., Hartel, A., Schemmel, J., and Meier, K. (2016). A highly tunable 65-nm CMOS LIF neuron for a large-scale neuromorphic system. In *Proceedings of IEEE European Solid-State Circuits Conference (ESSCIRC)*

   The described circuits were developed by Syed Ahmed Aamir and Johannes Schemmel in close collaboration with the author. The publication describes the DLS2 chip, which is not described in detail in this thesis.

3. Schemmel, J., Kriener, L., Müller, P., and Meier, K. (2017). An accelerated analog neuromorphic hardware system emulating nmda- and calcium-based non-linear dendrites. In *Proceedings of the 2017 IEEE International Joint Conference on Neural Networks*

   The described circuits were developed by Johannes Schemmel, Syed Ahmed Aamir and Gerd Kiene in close collaboration with the author. The author performed the simulations described therein in collaboration with Laura Kriener, who was co-supervised by the author during her Master's thesis. The methodology and results described in chapter 3 directly influenced the circuits that were developed for this publication.

4. Pfeil, T., Grübl, A., Jeltsch, S., Müller, E., Müller, P., Petrovici, M. A., Schmuker, M., Brüderle, D., Schemmel, J., and Meier, K. (2013). Six networks on a universal neuromorphic computing substrate. *Frontiers in Neuroscience*, 7

   The author performed the experiments and wrote the text for the part credited

to him (Section 3.1, Synfire Chain). This publication is not described in detail in this document.

5. Brüderle, D., Petrovici, M. A., Vogginger, B., Ehrlich, M., Pfeil, T., Millner, S., Grübl, A., Wendt, K., Müller, E., Schwartz, M.-O., de Oliveira, D., Jeltsch, S., Fieres, J., Schilling, M., Müller, P., Breitwieser, O., Petkov, V., Muller, L., Davison, A., Krishnamurthy, P., Kremkow, J., Lundqvist, M., Muller, E., Partzsch, J., Scholze, S., Zühl, L., Mayr, C., Destexhe, A., Diesmann, M., Potjans, T., Lansner, A., Schüffny, R., Schemmel, J., and Meier, K. (2011). A comprehensive workflow for general-purpose neural modeling with highly configurable neuromorphic hardware systems. *Biological Cybernetics*, 104:263–296

   The author performed the simulations related to the synfire chain and asynchronous irregular networks. The approach is a precursor to the one employed in Petrovici et al. (2014), but it differs in parameterization of the networks and in depth of analysis. This publication is not described in detail in this document.

6. Schmitt, S., Klähn, J., Bellec, G., Grübl, A., Güttler, M., Hartel, A., Hartmann, S., Husmann, D., Husmann, K., Jeltsch, S., Karasenko, V., Kleider, M., Koke, C., Kononov, A., Mauch, C., Müller, E., Müller, P., Partzsch, J., Petrovici, M. A., Schiefer, S., Scholze, S., Thanasoulis, V., Vogginger, B., Legenstein, R., Maass, W., Mayr, C., Schüffny, R., Schemmel, J., and Meier, K. (2017). Neuromorphic hardware in the loop: Training a deep spiking network on the brainscales wafer-scale system. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 2227–2234. IEEE

   In this publication, the wafer-scale BrainScaleS system is used to perform image classification. The author contributed to the calibration and automation for use of that system. This work is not described within this thesis.

7. Petrovici, M. A., Schmitt, S., Klähn, J., Stöckel, D., Schroeder, A., Bellec, G., Bill, J., Breitwieser, O., Bytschok, I., Grübl, A., Güttler, M., Hartel, A., Hartmann, S., Husmann, D., Husmann, K., Jeltsch, S., Karasenko, V., Kleider, M., Koke, C., Kononov, A., Mauch, C., Müller, E., Müller, P., Partzsch, J., Pfeil, T., Schiefer, S., Scholze, S., Subramoney, A., Thanasoulis, V., and et al., B. V. (2017). Pattern representation and recognition with accelerated analog neuromorphic systems. *to appear in Proceedings of the 2017 IEEE International Symposium on Circuits and Systems*

   This publication is a review on pattern representation in accelerated neuromorphic systems, including Schmitt et al. (2017). This work is not described within this thesis.

The author of this thesis contributed to the currently unpublished works Aamir et al. (2017b) and Aamir et al. (2017a), which contain results that are also shown in this thesis in chapter 3.

References

Aamir, S. A., Müller, P., Hartel, A., Schemmel, J., and Meier, K. (2016). A highly tunable 65-nm CMOS LIF neuron for a large-scale neuromorphic system. In *Proceedings of IEEE European Solid-State Circuits Conference (ESSCIRC)*. (Cited on pages 11, 58, 65, 75, and 79.)

Aamir, S. A., Müller, P., Kriener, L., Kiene, G., Schemmel, J., and Meier, K. (2017a). From lif to adex neuron models: Accelerated analog 65 nm cmos implementation. Article in preparation, git revision f37ebcce. (Cited on pages 58, 59, 65, 126, 127, 151, and 162.)

Aamir, S. A., Stradmann, Y., Müller, P., Pehle, C., Hartel, A., Grübl, A., Schemmel, J., and Meier, K. (2017b). An accelerated lif neuronal network array for a large scale mixed signal cmos architecture. Article in preparation, git revision 2b26bd59. (Cited on pages 58, 81, 136, and 162.)

Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, 9:147–169. (Cited on page 18.)

Antic, S. D., Zhou, W.-L., Moore, A. R., Short, S. M., and Ikonomu, K. D. (2010). The decade of the dendritic nmda spike. *Journal of neuroscience research*, 88(14):2991–3001. (Cited on pages 139, 140, and 142.)

Arthur, J. and Boahen, K. (2007). Silicon neurons that inhibit to synchronize. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pages 1186 –1186. (Cited on page 17.)

Ascoli, G. A., Donohue, D. E., and Halavi, M. (2007). Neuromorpho. org: a central resource for neuronal morphologies. *Journal of Neuroscience*, 27(35):9247–9251. (Cited on page 11.)

Azevedo, F. A., Carvalho, L. R., Grinberg, L. T., Farfel, J. M., Ferretti, R. E., Leite, R. E., Lent, R., Herculano-Houzel, S., et al. (2009). Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *Journal of Comparative Neurology*, 513(5):532–541. (Cited on page 10.)

Badel, L., Lefort, S., Berger, T. K., Petersen, C. C., Gerstner, W., and Richardson, M. J. (2008). Extracting non-linear integrate-and-fire models from experimental data using dynamic i–v curves. *Biological cybernetics*, 99(4-5):361. (Cited on page 13.)

Benjamin, B. V., Gao, P., McQuinn, E., Choudhary, S., Chandrasekaran, A. R., Bussat, J.-M., Alvarez-Icaza, R., Arthur, J. V., Merolla, P. A., and Boahen, K. (2014). Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE*, 102(5):699–716. (Cited on page 17.)

Bi, G.-q. and Poo, M.-m. (1998). Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, 18(24):10464–10472. (Cited on page 16.)

Billaudelle, S. (2017). Design and implementation of a short term plasticity circuit for a 65 nm neuromorphic hardware system. Master's thesis, Universität Heidelberg. (Cited on page 59.)

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*. (Cited on page 7.)

Breitwieser, O. (2015). Towards a neuromorphic implementation of spike-based expectation maximization. Masterarbeit, Universität Heidelberg. (Cited on page 52.)

Brette, R. (2006). Exact simulation of integrate-and-fire models with synaptic conductances. *Neural Computation*, 18(8):2004–2027. (Cited on page 14.)

Brette, R. and Gerstner, W. (2005). Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of neurophysiology*, 94(5):3637–3642. (Cited on pages 13, 52, 54, 55, 56, and 57.)

Bringuier, V., Chavane, F., Glaeser, L., and Frégnac, Y. (1999). Horizontal propagation of visual activity in the synaptic integration field of area 17 neurons. *Science*, 283(5402):695–699. (Cited on page 37.)

Brüderle, D. (2009). *Neuroscientific Modeling with a Mixed-Signal VLSI Hardware System*. PhD thesis, Ruprecht-Karls-Universität Heidelberg. (Cited on page 24.)

Brüderle, D., Petrovici, M. A., Vogginger, B., Ehrlich, M., Pfeil, T., Millner, S., Grübl, A., Wendt, K., Müller, E., Schwartz, M.-O., de Oliveira, D., Jeltsch, S., Fieres, J., Schilling, M., Müller, P., Breitwieser, O., Petkov, V., Muller, L., Davison, A., Krishnamurthy, P., Kremkow, J., Lundqvist, M., Muller, E., Partzsch, J., Scholze, S., Zühl, L., Mayr, C., Destexhe, A., Diesmann, M., Potjans, T., Lansner, A., Schüffny, R., Schemmel, J., and Meier, K. (2011). A comprehensive workflow for general-purpose neural modeling with highly configurable neuromorphic hardware systems. *Biological Cybernetics*, 104:263–296. (Cited on pages 23, 24, 27, and 49.)

Brunel, N. (2000). Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *Journal of computational neuroscience*, 8(3):183–208. (Cited on pages 8 and 13.)

Brunel, N. and Wang, X.-J. (2001). Effects of neuromodulation in a cortical network model of object working memory dominated by recurrent inhibition. *Journal of computational neuroscience*, 11(1):63–85. (Cited on pages 54, 55, 56, and 57.)

Buesing, L., Bill, J., Nessler, B., and Maass, W. (2011). Neural dynamics as sampling: A model for stochastic computation in recurrent networks of spiking neurons. *PLoS Computational Biology*, 7(11):e1002211. (Cited on page 20.)

Chen, J.-R., Wang, B.-N., Tseng, G.-F., Wang, Y.-J., Huang, Y.-S., and Wang, T.-J. (2014). Morphological changes of cortical pyramidal neurons in hepatic encephalopathy. *BMC neuroscience*, 15(1):15. (Cited on page 11.)

Connors, B. and Gutnick, M. (1990). Intrinsic firing patterns of diverse neocortical neurons. *Trends Neurosci.*, 13:99–104. (Cited on page 37.)

Davison, A. P., Brüderle, D., Eppler, J., Kremkow, J., Muller, E., Pecevski, D., Perrinet, L., and Yger, P. (2008). Pynn: A common interface for neuronal network simulators. *Frontiers in neuroinformatics*, 2. (Cited on pages 14, 23, and 27.)

De Nicola, G., di Tommaso, P., Rosaria, E., Francesco, F., Pietro, M., and Antonio, O. (2005). A grey-box approach to the functional testing of complex automatic train protection systems. *Dependable Computing-EDCC 5*, pages 305–317. (Cited on page 7.)

Deco, G. and Jirsa, V. K. (2012). Ongoing cortical activity at rest: criticality, multistability, and ghost attractors. *The Journal of Neuroscience*, 32(10):3366–3375. (Cited on pages 54, 55, 56, and 57.)

Destexhe, A. (2009). Self-sustained asynchronous irregular states and Up/Down states in thalamic, cortical and thalamocortical networks of nonlinear integrate-and-fire neurons. *Journal of Computational Neuroscience*, 3:493 – 506. (Cited on pages 37, 53, 54, 55, 56, and 57.)

Destexhe, A., Mainen, Z. F., and Sejnowski, T. J. (1994). Synthesis of models for excitable membranes, synaptic transmission and neuromodulation using a common kinetic formalism. *Journal of computational neuroscience*, 1(3):195–230. (Cited on pages 54, 55, 56, and 57.)

Destexhe, A., Mainen, Z. F., and Sejnowski, T. J. (1998). Kinetic models of synaptic transmission. *Methods in neuronal modeling*, 2:1–25. (Cited on page 14.)

Destexhe, A. and Pare, D. (1999). Impact of Network Activity on the Integrative Properties of Neocortical Pyramidal Neurons In Vivo. *J Neurophysiol*, 81(4):1531–1547. (Cited on page 37.)

Destexhe, A., Rudolph, M., and Paré, D. (2003). The high-conductance state of neocortical neurons in vivo. *Nature reviews neuroscience*, 4(9):739–751. (Cited on pages 15 and 37.)

Diesmann, M. (2002). *Conditions for Stable Propagation of Synchronous Spiking in Cortical Neural Networks: Single Neuron Dynamics and Network Properties*. PhD thesis, Ruhr-Universität Bochum. (Cited on page 34.)

Ehrlich, M., Mayr, C., Eisenreich, H., Henker, S., Srowig, A., Grübl, A., Schemmel, J., and Schüffny, R. (2007). Wafer-scale VLSI implementations of pulse coupled neural networks. In *Proceedings of the International Conference on Sensors, Circuits and Instrumentation Systems (SSD-07)*. (Cited on pages 24, 26, and 27.)

El Boustani, S. and Destexhe, A. (2009). A master equation formalism for macroscopic modeling of asynchronous irregular activity states. *Neural Computation*, 21(1):46–100. (Cited on page 37.)

Eppler, J. M., Helias, M., Muller, E., Diesmann, M., and Gewaltig, M.-O. (2008). PyNEST: a convenient interface to the NEST simulator. *Front. Neuroinform.*, 2(12). (Cited on page 9.)

Farquhar, E. and Hasler, P. (2005). A bio-physically inspired silicon neuron. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 52(3):477 – 488. (Cited on page 17.)

Fieres, J., Schemmel, J., and Meier, K. (2008). Realizing biological spiking network models in a configurable wafer-scale hardware system. In *Proceedings of the 2008 International Joint Conference on Neural Networks (IJCNN)*. (Cited on page 23.)

Friedmann, S. (2013). *A new approach to learning in neuromorphic hardware*. PhD thesis, Universität Heidelberg. (Cited on page 59.)

Friedmann, S., Schemmel, J., Grübl, A., Hartel, A., Hock, M., and Meier, K. (2016). Demonstrating hybrid learning in a flexible neuromorphic hardware system. *IEEE Transactions on Biomedical Circuits and Systems*, PP(99):1–15. (Cited on pages 18, 49, and 59.)

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741. (Cited on page 20.)

Gerstner, W. and Kistler, W. (2002). *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press. (Cited on pages 11 and 17.)

Gerstner, W., Kistler, W., Naud, R., and Paninski, L. (2014). *Neuronal Dynamics*. Cambridge University Press. (Cited on pages 16 and 17.)

Gewaltig, M.-O. and Diesmann, M. (2007). Nest (neural simulation tool). *Scholarpedia*, 2(4):1430. (Cited on page 27.)

Golding, N. L., Staff, N. P., and Spruston, N. (2002). Dendritic spikes as a mechanism for cooperative long-term potentiation. *Nature*, 418(6895):326. (Cited on page 139.)

González-Burgos, G., Barrionuevo, G., and Lewis, D. A. (2000). Horizontal synaptic connections in monkey prefrontal cortex: An in vitro electrophysiological study. *Cerebral Cortex*, 10(1):82–92. (Cited on page 37.)

Hartel, A. (2016). *Implementation and Characterization of Mixed-Signal Neuromorphic ASICs*. PhD thesis, Universität Heidelberg. (Cited on page 97.)

Hartel, A., Schemmel, J., et al. (2017). Specification of the hicann-dls asic. (Cited on pages 58, 66, 87, and 112.)

HBP SP9 partners (2017). *Neuromorphic Platform Specification, version 7786ee3*. Human Brain Project. (Cited on page 97.)

Hines, M. and Carnevale, N. (2003). *The NEURON simulation environment.*, pages 769–773. M.A. Arbib. (Cited on pages 9 and 27.)

Hirsch, J. and Gilbert, C. (1991). Synaptic physiology of horizontal connections in the cat's visual cortex. *The Journal of Neuroscience*, 11(6):1800–1809. (Cited on pages 29 and 37.)

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780. (Cited on page 7.)

Hock, M. (2014). *Modern semiconductor technologies for neuromorphic hardware*. PhD thesis, Universität Heidelberg. (Cited on pages 59, 61, 62, and 64.)

Hock, M., Hartel, A., Schemmel, J., and Meier, K. (2013). An analog dynamic memory array for neuromorphic hardware. In *Circuit Theory and Design (ECCTD), 2013 European Conference on*, pages 1–4. (Cited on page 18.)

Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544. (Cited on page 10.)

Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6):1569–1572. (Cited on page 13.)

Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? *IEEE transactions on neural networks*, 15(5):1063–1070. (Cited on pages 13 and 50.)

Jeltsch, S. (2014). *A Scalable Workflow for a Configurable Neuromorphic Platform*. PhD thesis, Universität Heidelberg. (Cited on page 24.)

Kiene, G. (2014). Evaluating the synaptic input of a neuromorphic circuit. Bachelor thesis, Universität Heidelberg. (Cited on page 126.)

Kiene, G. (2017). Mixed-signal neuron and readout circuits for a neuromorphic system. Master's thesis, Universität Heidelberg. (Cited on pages 58 and 59.)

Kinget, P. R. (2005). Device mismatch and tradeoffs in the design of analog circuits. *IEEE Journal of Solid-State Circuits*, 40(6):1212–1224. (Cited on page 64.)

Koke, C. (2017). *Device Variability in Synapses of Neuromorphic Circuits*. PhD thesis, Universität Heidelberg. (Cited on pages 24, 49, and 79.)

Kremkow, J., Perrinet, L., Masson, G., and Aertsen, A. (2010). Functional consequences of correlated excitatory and inhibitory conductances in cortical networks. *J Comput Neurosci*, 28:579–594. (Cited on pages 29, 30, 34, and 155.)

Kriener, L. (2017). Characterization of single-neuron dynamics in the development of neuromorphic hardware. Master's thesis, Heidelberg University. (Cited on pages 90, 97, 98, 102, 117, 119, 120, 121, 122, 127, 128, 144, and 145.)

Kumar, A., Schrader, S., Aertsen, A., and Rotter, S. (2008). The high-conductance state of cortical networks. *Neural Computation*, 20(1):1–43. (Cited on page 37.)

Kungl, A. (2016). Sampling with leaky integrate-and-fire neurons on the hicannv4 neuromorphic chip. Masterarbeit, Universität Heidelberg. (Cited on page 129.)

Kunkel, S., Schmidt, M., Eppler, J. M., Plesser, H. E., Masumoto, G., Igarashi, J., Ishii, S., Fukai, T., Morrison, A., Diesmann, M., et al. (2014). Spiking network simulation code for petascale computers. *Frontiers in neuroinformatics*, 8. (Cited on page 8.)

Larkum, M. (2013). A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex. *Trends in Neurosciences*, 36(3):141 – 151. (Cited on pages 139, 141, 143, and 145.)

Larkum, M. E., Nevian, T., Sandler, M., Polsky, A., and Schiller, J. (2009). Synaptic integration in tuft dendrites of layer 5 pyramidal neurons: a new unifying principle. *Science*, 325(5941):756–760. (Cited on pages 139 and 141.)

Larkum, M. E., Zhu, J. J., and Sakmann, B. (1999). A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature*, 398(6725):338–341. (Cited on pages 139 and 143.)

Leventhal, A. (2008). Flash storage memory. *Communications of the ACM*, 51(7):47–51. (Cited on page 7.)

Livi, P. and Indiveri, G. (2009). A current-mode conductance-based silicon neuron for address-event neuromorphic systems. In *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pages 2898 –2901. (Cited on page 17.)

Mack, C. A. (2011). Fifty years of moore's law. *IEEE Transactions on semiconductor manufacturing*, 24(2):202–207. (Cited on page 7.)

Mahowald, M. and Douglas, R. (1991). A silicon neuron. *Nature*, 354(6354):515–518. (Cited on page 17.)

Markram, H., Lübke, J., Frotscher, M., and Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps. *Science*, 275(5297):213–215. (Cited on page 16.)

Markram, H., Toledo-Rodriguez, M., Wang, Y., Gupta, A., Silberberg, G., and Wu, C. (2004). Interneurons of the neocortical inhibitory system. *Nature reviews. Neuroscience*, 5(10):793. (Cited on pages 13, 125, and 158.)

Masquelier, T. and Deco, G. (2013). Network bursting dynamics in excitatory cortical neuron cultures results from the combination of different adaptive mechanism. *PLOS ONE*, 8(10):e75824. (Cited on pages 54, 55, 56, and 57.)

Mead, C. A. (1990). Neuromorphic electronic systems. *Proceedings of the IEEE*, 78:1629–1636. (Cited on page 17.)

Meffin, H., Burkitt, A. N., and Grayden, D. B. (2004). An analytical model for the 'large, fluctuating synaptic conductance state'typical of neocortical neurons in vivo. *Journal of computational neuroscience*, 16(2):159–175. (Cited on page 14.)

Millner, S. (2012). *Development of a Multi-Compartment Neuron Model Emulation*. PhD thesis, Ruprecht-Karls University Heidelberg. (Cited on pages 24, 50, 52, 54, 55, 56, 57, 73, 74, 126, 151, and 153.)

Millner, S., Grübl, A., Meier, K., Schemmel, J., and Schwartz, M.-O. (2010). A VLSI implementation of the adaptive exponential integrate-and-fire neuron model. In Lafferty, J., Williams, C. K. I., Shawe-Taylor, J., Zemel, R., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 1642–1650. (Cited on pages 18, 23, 24, 50, and 126.)

Millner, S., Hartel, A., Schemmel, J., and Meier, K. (2012). Towards biologically realistic multi-compartment neuron model emulation in analog VLSI. In *Proceedings ESANN 2012*. (Cited on page 67.)

Moore, G. E. (1965). Cramming more components onto integrated circuits. *Electronics*, 38(8). (Cited on page 7.)

Morrison, A., Aertsen, A., and Diesmann, M. (2007). Spike-timing-dependent plasticity in balanced random networks. *Neural computation*, 19(6):1437–1467. (Cited on page 16.)

Müller, E. C. (2014). *Novel Operation Modes of Accelerated Neuromorphic Hardware*. PhD thesis, Ruprecht-Karls-Universität Heidelberg. HD-KIP 14-98. (Cited on page 24.)

Muller, L. and Destexhe, A. (2012). Propagating waves in thalamus, cortex and the thalamocortical system: Experiments and models. *Journal of Physiology-Paris*, 106(5–6):222 – 238. <ce:title>New trends in neurogeometrical approaches to the brain and mind problem</ce:title>. (Cited on pages 37 and 155.)

Murakoshi, T., Guo, J.-Z., and Ichinose, T. (1993). Electrophysiological identification of horizontal synaptic connections in rat visual cortex in vitro. *Neuroscience Letters*, 163(2):211 – 214. (Cited on page 37.)

Nakanishi, K. and Kukita, F. (1998). Functional synapses in synchronized bursting of neocortical neurons in culture. *Brain research*, 795(1):137–146. (Cited on pages 54, 55, 56, and 57.)

Naud, R., Marcille, N., Clopath, C., and Gerstner, W. (2008). Firing patterns in the adaptive exponential integrate-and-fire model. *Biological Cybernetics*, 99(4):335–347. (Cited on pages 52, 54, 55, 56, 57, 106, 125, 126, and 128.)

nel (2017). `article_ephys_metadata_curated.csv`. online. snapshot of neuroelectro database. (Cited on pages 51, 52, and 53.)

Nessler, B., Pfeiffer, M., Buesing, L., and Maass, W. (2013). Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLoS Computational Biology*, 9(4):e1003037. (Cited on pages 52, 54, 55, 56, and 57.)

Pare, D., Shink, E., Gaudreau, H., Destexhe, A., and Lang, E. J. (1998). Impact of spontaneous synaptic activity on the resting properties of cat neocortical pyramidal neurons in vivo. *J Neurophysiol*, 79(3):1450–60. (Cited on pages 56 and 58.)

Pelgrom, M. J., Duinmaijer, A. C., and Welbers, A. P. (1989). Matching properties of mos transistors. *IEEE Journal of solid-state circuits*, 24(5):1433–1439. (Cited on page 9.)

Petrovici, M. A. (2015). *Form vs. Function: Theory and Models for Neuronal Substrates*. PhD thesis, Universität Heidelberg. (Cited on pages 19 and 20.)

Petrovici, M. A., Bill, J., Bytschok, I., Schemmel, J., and Meier, K. (2013). Stochastic inference with deterministic spiking neurons. *arXiv preprint arXiv:1311.3211*. (Cited on pages 52, 54, 55, 56, and 57.)

Petrovici, M. A., Bill, J., Bytschok, I., Schemmel, J., and Meier, K. (2016). Stochastic inference with spiking neurons in the high-conductance state. *Physical Review E*, 94(4). (Cited on pages 18, 20, 21, and 22.)

Petrovici, M. A., Bytschok, I., Bill, J., Schemmel, J., and Meier, K. (2015). The high-conductance state enables neural sampling in networks of LIF neurons. *BMC Neuroscience*, 16(Suppl 1):O2. (Cited on page 58.)

Petrovici, M. A., Schmitt, S., Klähn, J., Stöckel, D., Schroeder, A., Bellec, G., Bill, J., Breitwieser, O., Bytschok, I., Grübl, A., Güttler, M., Hartel, A., Hartmann, S., Husmann, D., Husmann, K., Jeltsch, S., Karasenko, V., Kleider, M., Koke, C., Kononov, A., Mauch, C., Müller, E., Müller, P., Partzsch, J., Pfeil, T., Schiefer, S., Scholze, S., Subramoney, A., Thanasoulis, V., and et al., B. V. (2017). Pattern

representation and recognition with accelerated analog neuromorphic systems. *to appear in Proceedings of the 2017 IEEE International Symposium on Circuits and Systems*. (Not cited.)

Petrovici, M. A., Vogginger, B., Müller, P., Breitwieser, O., Lundqvist, M., Muller, L., Ehrlich, M., Destexhe, A., Lansner, A., Schüffny, R., et al. (2014). Characterization and compensation of network-level anomalies in mixed-signal neuromorphic modeling platforms. *PloS one*, 9(10):e108590. (Cited on pages 24, 25, 26, 28, 29, 30, 32, 33, 34, 35, 36, 37, 39, 41, 42, 43, 45, 46, 47, 54, 55, 56, 57, 155, 162, 183, and 184.)

Pfeil, T., Grübl, A., Jeltsch, S., Müller, E., Müller, P., Petrovici, M. A., Schmuker, M., Brüderle, D., Schemmel, J., and Meier, K. (2013). Six networks on a universal neuromorphic computing substrate. *Frontiers in Neuroscience*, 7. (Cited on pages 49 and 129.)

Pospischil, M., Piwkowska, Z., Bal, T., and Destexhe, A. (2011). Comparison of different neuron models to conductance-based post-stimulus time histograms obtained in cortical pyramidal cells using dynamic-clamp in vitro. *Biological cybernetics*, 105(2):167–180. (Cited on pages 52, 54, 55, 56, and 57.)

Raichle, M. E. and Gusnard, D. A. (2002). Appraising the brain's energy budget. *Proceedings of the National Academy of Sciences*, 99(16):10237–10239. (Cited on page 7.)

Rall, W. (1959). Branching dendritic trees and motoneuron membrane resistivity. *Experimental neurology*, 1(5):491–527. (Cited on page 8.)

Ricciardi, L. M. and Sacerdote, L. (1979). The ornstein-uhlenbeck process as a model for neuronal activity. *Biological cybernetics*, 35(1):1–9. (Cited on page 21.)

Salakhutdinov, R. and Hinton, G. (2009). Deep boltzmann machines. In *Artificial Intelligence and Statistics*, pages 448–455. (Cited on page 7.)

Samuel, A. L. (1967). Some studies in machine learning using the game of checkers. ii—recent progress. *IBM Journal of research and development*, 11(6):601–617. (Cited on page 7.)

Sanchez-Sinencio, E. and Silva-Martinez, J. (2000). Cmos transconductance amplifiers, architectures and active filters: a tutorial. *IEE proceedings-circuits, devices and systems*, 147(1):3–12. (Cited on page 73.)

Schemmel, J., Fieres, J., and Meier, K. (2008). Wafer-scale integration of analog neural networks. In *Proceedings of the 2008 International Joint Conference on Neural Networks (IJCNN)*. (Cited on pages 17, 18, 26, and 58.)

Schemmel, J., Kriener, L., Müller, P., and Meier, K. (2017). An accelerated analog neuromorphic hardware system emulating nmda- and calcium-based non-linear dendrites. In *Proceedings of the 2017 IEEE International Joint Conference on Neural Networks*. (Cited on pages 18, 58, 138, 142, 143, and 159.)

Schiller, J., Major, G., Koester, H. J., and Schiller, Y. (2000). Nmda spikes in basal dendrites of cortical pyramidal neurons. *Nature*, 404(6775):285–289. (Cited on page 139.)

Schmidt, D. (2014). Automated characterization of a wafer-scale neuromorphic hardware system. Master thesis, Ruprecht-Karls-Universität Heidelberg. (Cited on page 49.)

Schmitt, S., Klähn, J., Bellec, G., Grübl, A., Güttler, M., Hartel, A., Hartmann, S., Husmann, D., Husmann, K., Jeltsch, S., Karasenko, V., Kleider, M., Koke, C., Kononov, A., Mauch, C., Müller, E., Müller, P., Partzsch, J., Petrovici, M. A., Schiefer, S., Scholze, S., Thanasoulis, V., Vogginger, B., Legenstein, R., Maass, W., Mayr, C., Schüffny, R., Schemmel, J., and Meier, K. (2017). Neuromorphic hardware in the loop: Training a deep spiking network on the brainscales wafer-scale system. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 2227–2234. IEEE. (Cited on pages 49 and 162.)

Scholze, S., Henker, S., Partzsch, J., Mayr, C., and Schüffny, R. (2010). Optimized queue based communication in vlsi using a weakly ordered binary heap. In *Mixed Design of Integrated Circuits and Systems (MIXDES), 2010 Proceedings of the 17th International Conference*, pages 316–320. IEEE. (Cited on page 26.)

Schwartz, M.-O. (2013). *Reproducing Biologically Realistic Regimes on a Highly-Accelerated Neuromorphic Hardware System*. PhD thesis, Universität Heidelberg. (Cited on page 24.)

Silberberg, G. and Markram, H. (2007). Disynaptic inhibition between neocortical pyramidal cells mediated by martinotti cells. *Neuron*, 53(5):735–746. (Cited on page 29.)

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489. (Cited on page 7.)

Sjöström, J. and Gerstner, W. (2010). Spike-timing dependent plasticity. *Scholarpedia*, 5(2):1362. revision 151671. (Cited on page 16.)

Srowig, A., Loock, J.-P., Meier, K., Schemmel, J., Eisenreich, H., Ellguth, G., and Schüffny, R. (2007). Analog floating gate memory in a 0.18 μm single-poly CMOS process. *FACETS internal documentation*. (Cited on page 24.)

Stein, R. B. (1967). Some models of neuronal variability. *Biophysical journal*, 7(1):37–68. (Cited on page 11.)

Stradmann, Y. (2016). Characterization and calibration of a mixed-signal leaky integrate and fire neuron on hicann-dls. Bachelorarbeit, Universität Heidelberg. (Cited on pages 79 and 97.)

Tajalli, A., Leblebici, Y., and Brauer, E. J. (2008). Implementing ultra-high-value floating tunable cmos resistors. *Electronics Letters*, 44(5):349–350. (Cited on page 77.)

Telfeian, A. E. and Connors, B. W. (2003). Widely integrative properties of layer 5 pyramidal cells support a role for processing of extralaminar synaptic inputs in rat neocortex. *Neuroscience Letters*, 343(2):121 – 124. (Cited on page 37.)

Tripathy, S. J., Savitskaya, J., Burton, S. D., Urban, N. N., and Gerkin, R. C. (2015). Neuroelectro: a window to the world's neuron electrophysiology data. *Recent Advances and the Future Generation of Neuroinformatics Infrastructure*, page 146. (Cited on page 51.)

Tsodyks, M., Pawelzik, K., and Markram, H. (1998). Neural networks with dynamic synapses. *Neural computation*, 10(4):821–835. (Cited on page 15.)

Tsodyks, M. V. and Markram, H. (1997). The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. *Proceedings of the National Academy of Sciences*, 94(2):719–723. (Cited on page 15.)

Vogels, T. P. and Abbott, L. F. (2005). Signal propagation and logic gating in networks of integrate-and-fire neurons. *J Neurosci*, 25(46):10786–95. (Cited on pages 53, 54, 55, 56, and 57.)

Vogginger, B. (2010). Testing the operation workflow of a neuromorphic hardware system with a functionally accurate model. Diploma thesis, Ruprecht-Karls-Universität Heidelberg. HD-KIP-10-12. (Cited on page 27.)

Waters, J., Schaefer, A., and Sakmann, B. (2005). Backpropagating action potentials in neurones: measurement, mechanisms and potential functions. *Progress in biophysics and molecular biology*, 87(1):145–170. (Cited on page 140.)

Wong, S. L. and Salama, C. A. T. (1986). An efficient cmos buffer for driving large capacitive loads. *IEEE journal of solid-state circuits*, 21(3):464–469. (Cited on pages 71 and 72.)

Yger, P., El Boustani, S., Destexhe, A., and Frégnac, Y. (2011). Topologically invariant macroscopic statistics in balanced networks of conductance-based integrate-and-fire neurons. *Journal of computational neuroscience*, 31(2):229–245. (Cited on page 37.)

Glossary

ADC analog-to-digital converter. 58, 59, 97, 112, 114

AdEx adaptive exponential integrate-and-fire. 5, 10, 13, 24, 27, 28, 37, 50, 52, 53, 59, 65, 74, 96, 106, 125, 126, 128, 152, 156, 181

AMPA α-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid. 53

ANNCORE Analog Neural Network Core. 60, 61, 152

API application programming interface. 23, 26, 27

BAC backpropagation activated calcium spike. 139, 145, 156, 159

CMOS complementary metal–oxide–semiconductor. 18, 62, 71

DAC digital-to-analog converter. 65, 77, 106, 110, 116, 152, 158

DLS Digital Learning System. 18, 79

DLS2 Digital Learning System. 58, 60

DLS3 Digital Learning System. 4, 6, 49, 50, 53, 58, 60, 62, 65–67, 69, 71, 73, 75, 77, 98, 115, 126, 129, 132, 136, 141, 145, 147, 151, 152, 156, 158, 159, 195

DNC Digital Network Core. 26

ESS Executable System Specification. 24, 27–29, 35, 40, 45, 47, 155

FACETS fast analog computing with emergent transient states. 18

FPGA field-programmable gate array. 26, 115

FS fast spiking inhibitory. 29, 30, 34

HICANN High Input Count Analog Neural Network. 11, 18, 24–26, 49, 53, 58, 77, 79, 129, 136, 152, 179

INH fast spiking inhibitory. 37, 44, 46

L1 layer 1. 25, 26

LIF leaky integrate-and-fire. 10, 13, 19–21, 28, 29, 50, 53, 58, 83, 95, 129, 132, 136, 139, 159, 179

LSB least significant bit. 144–146

LVS layout versus schematic. 60

MCMC Markov chain Monte Carlo. 20

MOSFET metal–oxide–semiconductor field-effect transistor. 64, 179

NMDA N-Methyl-D-aspartate. 53, 136, 139–141

NMOS n-channel MOSFET. 62–64, 71, 73, 100, 107

OTA operational transconductance amplifier. 69, 71–75, 79, 83–86, 88–90, 94, 96, 98, 99, 115, 120, 122–124, 132, 136, 147, 149, 151, 152, 157, 158, 180, 194

PMOS p-channel MOSFET. 62–64, 73, 99, 100, 121

PPU plasticity processing unit. 59, 152, 153, 157

PSP post-synaptic potential. 10, 12, 15, 22, 33, 94, 193

PY pyramidal. 37, 44, 46, 47

RS regular spiking. 29, 30, 34

SRAM static random-access memory. 111, 124

STDP spike-timing-dependent plasticity. 15

TTX tetrodotoxin. 140

VLSI very-large-scale integration. 17, 23

Symbols

CV_{isi} Coefficient of variation of interspike intervals sort. 37–40, 42, 43, 45

CV_{rate} Coefficient of variation of firing rates sort. 38, 40, 42, 43, 45

g_{exc} Excitatory conductance-based synaptic weight sort. 38–41, 45, 46

g_{inh} Inhibitory conductance-based synaptic weight sort. 38–41, 45, 46

fireout Firing detection signal of the neuron circuit. 130, 131, 135

holdoff Holdoff time for refractory period. 130

resetout Reset enable signal for the leak/reset OTA. 130

α_t model-to-hardware time scaling factor. 17, 18, 151

α_V model-to-hardware voltage scaling factor. 17, 18, 151, 152

CC Correlation coefficient. 37, 38, 40

C_{hw} hardware capacitance. 17

C_m membrane capacitance. 11, 13, 21, 54, 57, 92–94, 99, 123, 126, 151, 158, 183, 185

Δ_T slope factor of AdEx exponential current. 13, 57, 58, 106, 107, 110, 128, 152, 156, 158, 185

E_{rev} reversal potential. 15

$E_{rev,E}$ excitatory reversal potential. 12, 15, 21, 55, 58, 183, 185

$E_{rev,I}$ inhibitory reversal potential. 12, 15, 21, 55, 58, 183, 185

g_l leakage conductance. 11, 13, 21, 97–99, 109, 110, 151

g_{syn} synaptic current. 15, 21

$g_{syn,E}$ excitatory synaptic conductance. 12, 21

$g_{syn,I}$ inhibitory synaptic conductance. 12, 21

g_{tot} total synaptic conductance. 21

I_{syn} synaptic current. 13–15, 21, 80, 93, 94, 153, 193, 194

ω_v model-to-hardware voltage offset. 17

τ_{eff} effective membrane time constant. 21, 22

τ_{in} inactivation time constant. 15

τ_{m} membrane time constant. 11, 21, 44, 54, 58, 85–88, 92, 99, 151, 153, 183, 185

τ_{rec} recovery time constant. 15

τ_{syn} synaptic time constant. 22, 52, 54, 58, 82, 92, 151, 153, 188

$\tau_{\mathrm{syn,E}}$ excitatory synaptic time constant. 12, 15, 44, 82, 183, 185

$\tau_{\mathrm{syn,I}}$ inhibitory synaptic time constant. 12, 15, 82, 183

τ_{w} adaptation time constant. 13, 44, 57, 58, 96, 97, 100–102, 180, 185

θ Heaviside step function. 14

t_{sp} spike time within a spike train. 13–15

U_{SE} fraction of synaptic utilization. 15

τ_{ref} refractory period. 11, 12, 22, 44, 52–54, 58, 138, 183, 185

V_{eff} effective equilibrium potential. 21

$V_{\mathrm{eff,curr}}$ effective equilibrium potential for current-based synapses. 21, 129

V_{hw} hardware voltage. 17, 18

V_{leak} leak potential. 11–13, 21, 33, 34, 53, 55, 58, 74, 86, 97, 98, 109, 129, 132, 183, 185, 191

V_{m} membrane voltage. 11–13, 15, 21, 22, 33, 34, 67, 68, 74, 80, 81, 83–85, 88, 90, 92, 93, 97, 98, 102–109, 113–115, 118, 120, 125–129, 132, 138, 147–153, 157, 188, 190, 193, 194

V_{reset} reset voltage. 11, 12, 55, 58, 129, 183, 185

V_{thresh} firing threshold. 11, 12, 44, 53, 55, 58, 95, 132, 183, 185

V_{T} threshold potential of AdEx exponential current. 13, 106, 109, 110, 144, 152, 158

List of Figures

List of Tables

Appendix A

Additional data and figures

A.1 Synfire chain with feed-forward inhibition

A.1.1 Model parameters

Table A.1: Neuron parameters used in the synfire chain benchmark model. Taken from (Petrovici et al., 2014, Table S3.1)

Parameter	Value	Unit
C_{m}	0.29	nF
τ_{ref}	2	ms
V_{thresh}	-57	mV
V_{reset}	-70	mV
V_{leak}	-70	mV
τ_{m}	10	ms
$E_{\mathrm{rev,E}}$	0	mV
$E_{\mathrm{rev,I}}$	-75	mV
$\tau_{\mathrm{syn,E}}$	1.5	ms
$\tau_{\mathrm{syn,I}}$	10	ms

Table A.2: Projection properties for the feed-forward synfire chain. Taken from (Petrovici et al., 2014, Table S3.1)

Projection	weight μS	incoming synapses	delay ms
$\mathrm{RS}_n \rightarrow \mathrm{RS}_{n+1}$	0.001	60	20
$\mathrm{RS}_n \rightarrow \mathrm{FS}_{n+1}$	0.0035	60	20
$\mathrm{FS}_n \rightarrow \mathrm{RS}_n$	0.002	25	4

A.1.2 Filtering of spontaneous activity

In the case of high weight noise, strong synapses from the background stimulus cause neurons to have increased background activity. To provide a clearer picture

of the filtering properties of the synfire chain, spikes are classified as being part of spontaneous activity, and discarded, if less than N spikes in the same excitatory group occur in a time window of $\pm T$. The values for N and T are given at the point of the application of the filter. These values are chosen such that synchronous volleys with $a \geq 0.5$ are not removed. Figure A.1 shows a comparison of a compensated case with and without applied filter.

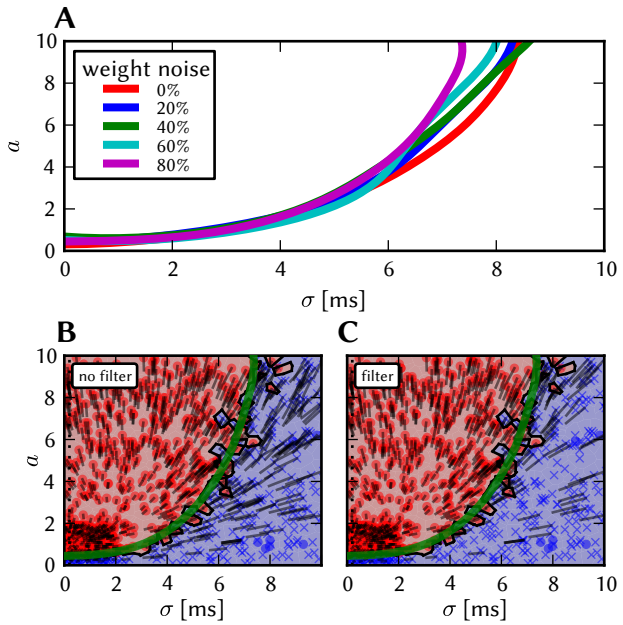


Figure A.1: Demonstration of spontaneous event filter in the weight noise compensation (Section A.1.2). **(A)** The same simulation setup as in fig. A.1 C (weight noise with active compensation) but without the filter for background spikes. The separatrix locations are comparable as the filter does not influence the result significantly in the compensated case. (**B**, **C**) Complete state space response for weight noise of 80 %, once with, once without filter. This demonstrates that the applied filter does not affect the result in the compensated case. Figure and caption used with permission from (Petrovici et al., 2014, Figure S3.3).

A.2 Cortical network with self-sustained asynchronous activity in a random network

A.2.1 Model parameters

Table A.3: AdEx Neuron parameters used in the AI network

Parameter	Pyramidal	Inhibitory	Unit
C_{m}	0.25	0.25	nF
τ_{ref}	5	5	ms
V_{thresh}	-40	-40	mV
V_{reset}	-70	-70	mV
V_{leak}	-70	-70	mV
τ_{m}	15	15	ms
a	1	1	nS
b	0.005	0	nA
Δ_{T}	2.5	2.5	mV
τ_{w}	600	600	ms
V_{thresh}	-50	-50	mV
$E_{\mathrm{rev,E}}$	0	0	mV
$E_{\mathrm{rev,I}}$	-80	-80	mV
$\tau_{\mathrm{syn,E}}$	5	5	ms
$\tau_{\mathrm{syn,E}}$	5	5	ms

A.3 Default parameters for DLS3 *testbench*

Table A.4: Default analog parameters used in the test bench.

parameter name	default value	unit	comment
i_adapt_w	20	nA	
i_bias_adapt_res	20	nA	
i_bias_adapt_sd	20	nA	
i_bias_adapt	1	µA	
i_bias_leak_sd	1	µA	
i_bias_leak	1	µA	
i_bias_nmda	20	nA	
i_bias_res_sd	1	µA	
i_bias_res	1	µA	
i_bias_syn_gm_exc	1	µA	
i_bias_syn_gm_inh	1	µA	
i_bias_syn_res_exc	100	nA	
i_bias_syn_res_inh	100	nA	
i_bias_syn_sd_exc	1	µA	
i_bias_syn_sd_inh	1	µA	
i_mem_off	20	nA	
i_ref_analog	250	nA	reference current which is mirrored to “iBiasReadOut” (1:4), “iBiasSpkCmp” (1:2) and “iBiasAdaptAmp” (1:4)
v_bdac	0.1	µA	is a current parameter
v_leak	0.6	V	
v_leak_adapt	0.6	V	
v_reset	0.4	V	
v_syn_exc	1.2	V	
v_syn_inh	1.2	V	
v_thresh	0.6	V	
v_thresh	1.1	V	

Table A.5: Digital parameters used in the DLS3 simulation set-up. Inverted parameters have a name ending in *_b* by convention.

parameter name	default value	comment
en_adapt	`(False, False)`	
en_ana_in	`False`	
en_ana_out_mux	`(False, False)`	To access the adaptation voltage *en_read_vw* must be enabled in addition to setting this parameter
en_ana_out	`False`	
en_bot	`False`	
en_cap_merge	`False`	
en_exp	`False`	
en_fire_out_b	`False`	Do not use this parameter to disable spike output, use *en_spk_cmp_b* instead. Exactly one of *en_fire_out_b*, *en_syn_byp_exc_b* and *en_syn_byp_inh_b* must be `False`, otherwise the input of the "fireout" inverter is floating (section A.11).
en_mem_cap	`(True, True, True, True, True, True)`	
en_mem_off	`False`	
en_neg_va	`False`	
en_nmda	`False`	
en_ota	`True`	
en_pos_vw	`True`	
en_read_vw	`False`	
en_right	`False`	
en_scon	`False`	
en_soma	`False`	
en_spk_cmp_b	`False`	
en_syn_byp_exc_b	`True`	see comment for *en_fire_out_b*
en_syn_byp_inh_b	`True`	see comment for *en_fire_out_b*
en_syn_i_exc	`False`	when `False`, *v_syn_exc* should be clearly *above* 1.2 V to prevent leakage through the output transmission gate
en_syn_i_inh	`False`	when `False`, *v_syn_inh* should be clearly *below* 1.2 V to prevent leakage through the output transmission gate
exp_weight_b	`(False, False, False)`	
highs_leak	`False`	
highs_res	`True`	
ib_nmda_div4	`False`	
ib_nmda_mul4	`False`	
refrac_clk_freq	10 MHz	possible values are 1 MHz and 10 MHz
refrac_time	10 (integer)	
holdoff_time	0 (integer)	
adaptation_time	1 (integer)	

A.4 LIF sampling

Table A.6: Parameters used for fig. 3.67

parameter name	value
i_bias_syn_res_exc	calibrated for $\tau_{\text{syn}} = 1.5\,\mu\text{s}$
i_bias_syn_res_inh	calibrated for $\tau_{\text{syn}} = 1.5\,\mu\text{s}$
i_mem_off	calibrated (see text)
v_reset	0.6
v_syn_exc	calibrated offset current
v_syn_inh	calibrated offset current
v_thresh	calibrated to 0.6
en_mem_cap	`(True, True, False, False, False, False)`
en_mem_off	`True`
en_syn_i_exc	`True`
en_syn_i_inh	`True`
highs_res	`False`
holdoff_time	`9`

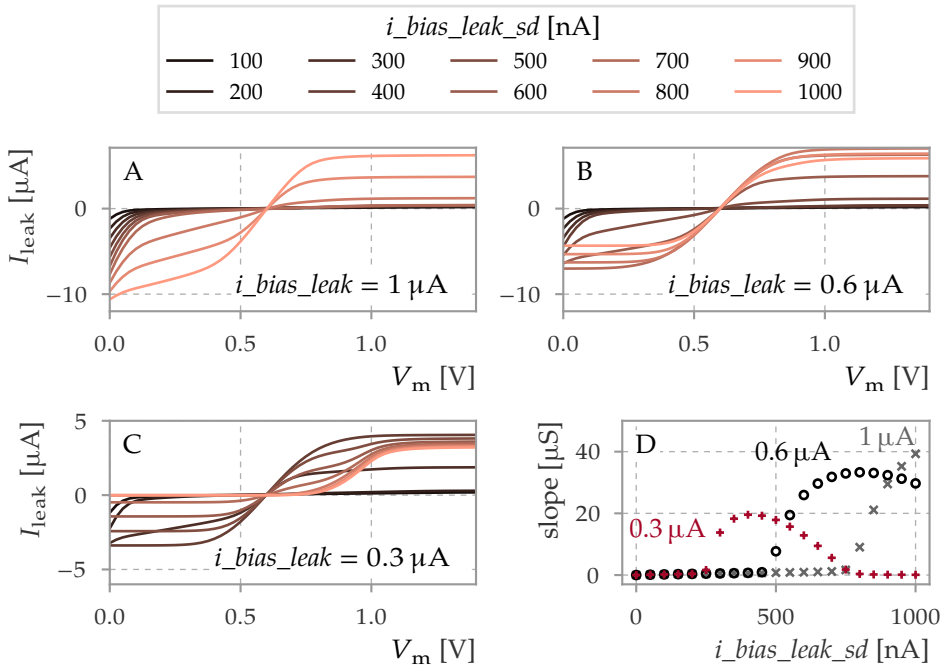


Figure A.2: Characteristics of the leak OTA with *highs* = `True`.

A.5 Reset current

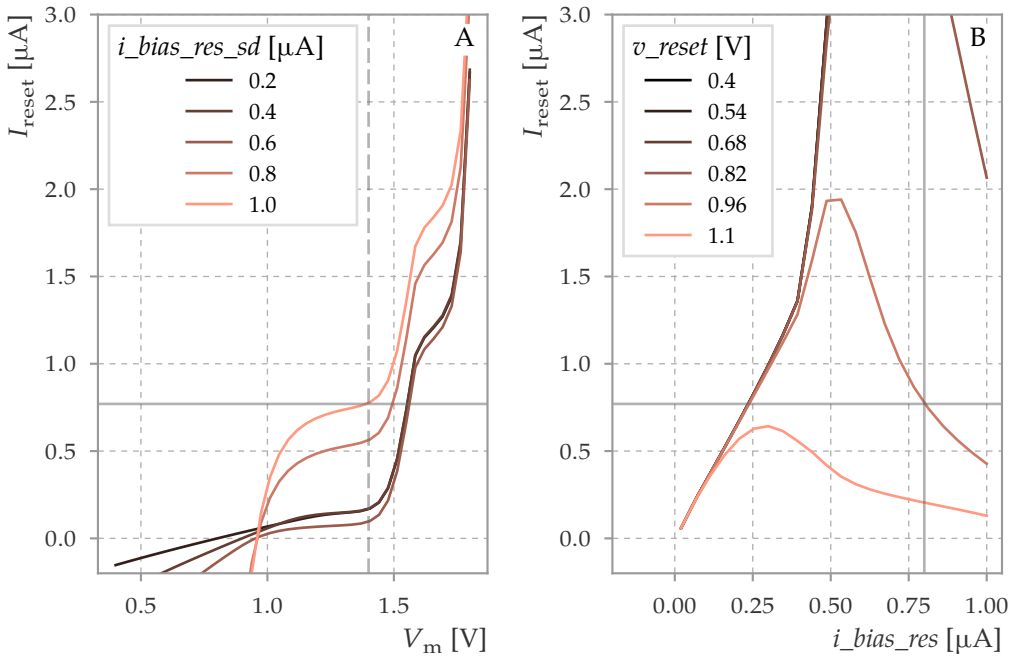


Figure A.3: Dependency of the reset current on *i_bias_res_sd*. **A**: The source degeneration bias changes the slope at the reset potential, which is the intended effect of this parameter. *v_reset* is set to 0.96 V and *i_bias_res* to 0.8 µA. The saturation voltage also changes with *i_bias_res_sd*. Note that the change is not monotonic. **B**: Zoom into fig. 3.33 A. The point marked by the horizontal and vertical line corresponds to the upper curve (*i_bias_res_sd* = 1 µA) in A.

A.6 Adaptation calibration

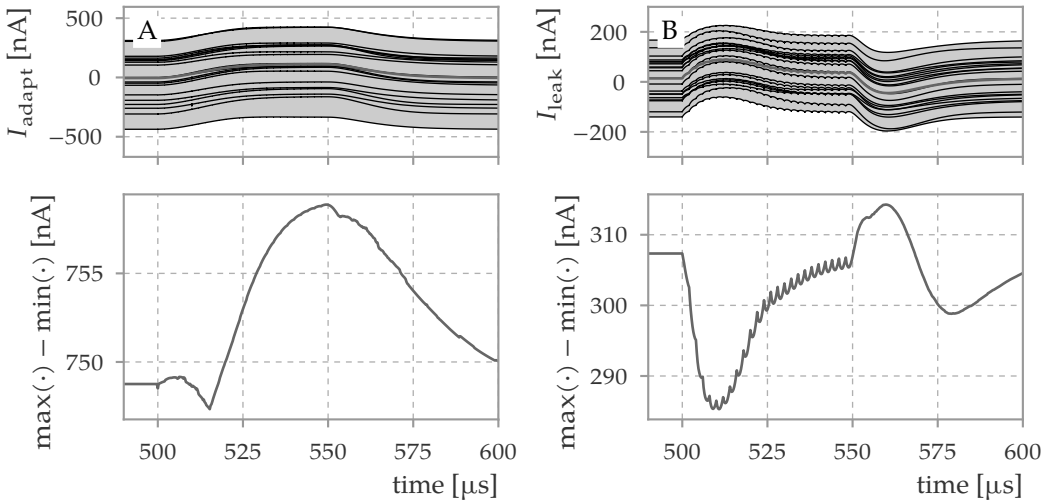


Figure A.4: Non-shifted values for the simulations shown in fig. 3.46 C and D. **A**: Adaptation current for 20 Monte-Carlo samples, using full calibration. **B**: Leak current for 20 Monte-Carlo samples, using full calibration.

A.7 Bistable firing

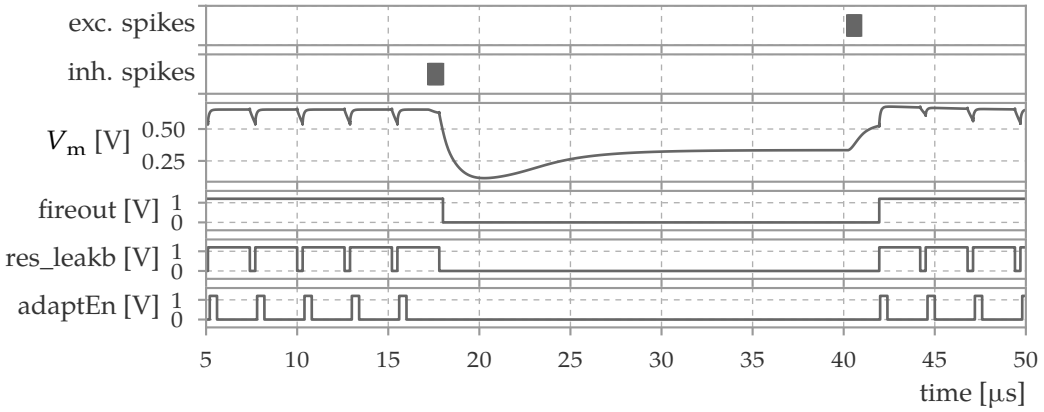


Figure A.5: When using a reset potential above the firing threshold, continuous firing can be enabled and disabled by excitatory and inhibitory spike input. This effectively creates a bistable neuron. Signal names as in fig. 3.10.

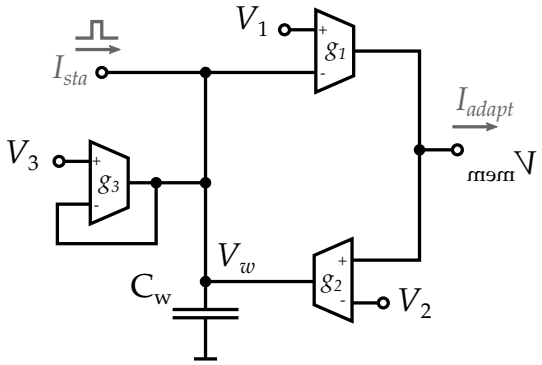


Figure A.6: Possible alternative design of the adaptation term.

A.8 Alternative implementation concept of adaptation circuit

An alternative design for the adaptation term is shown in fig. A.6. By introducing an additional conductance, the resting values of V_w and V_m are decoupled. The ideal behavior of the system is described by

$$w = -I_{\text{adapt}} = -g_1(V_1 - V_w) \tag{A.1}$$

$$C_w \frac{\mathrm{d}V_w}{\mathrm{d}t} = g_3(V_3 - V_w) + g_2(V_m - V_2) \tag{A.2}$$

$$V_w = \frac{w}{g_1} + V_1 \tag{A.3}$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = g_1 \frac{\mathrm{d}V_w}{\mathrm{d}t} \tag{A.4}$$

$$= \frac{g_1}{C_w}\left[g_3(V_3 - V_w) + g_2(V_m - V_2)\right] \tag{A.5}$$

$$= \frac{-g_3}{C_w} w + \frac{g_2}{C_w}\left[g_1(V_m - V_2) + \frac{g_1 g_3}{g_2}(V_3 - V_1)\right] \tag{A.6}$$

$$= \frac{-w}{\tau_w} + \frac{g_1 g_2}{C_w}\left[V_m - V_2 + \frac{g_3}{g_2}(V_3 - V_1)\right] \tag{A.7}$$

By setting $V_1 = V_3$, $V_2 = V_{\text{leak}}$, $\tau_w = \frac{C_w}{g_3}$, $a = \frac{g_1 g_2}{g_3}$, $b = g_1 \cdot \frac{T_b I_b}{C_w}$ we obtain the original form for the adaptation current

$$\tau_w \frac{\mathrm{d}w}{\mathrm{d}t} = -w + a\left[V_m - V_{\text{leak}}\right] \tag{A.8}$$

The voltages V_1, V_2, V_3 could be set to V_{leak} but should be separate parameters to compensate for the significant input offset, which is common to all amplifiers used in the neuron circuits of the DLS generation.

It is clear that the larger number of parameters offers a more flexible range of parameterization than eqs. (3.1) and (3.2). For example, by setting $g_2 = 0; g_1 \neq 0$, zero

voltage-coupled (sub-threshold) adaptation but non-zero spike-triggered adaptation can be achieved. The proposal was not considered for implementation, partly due to the larger number of required configuration parameters.

A.9 Calibration of synaptic weight

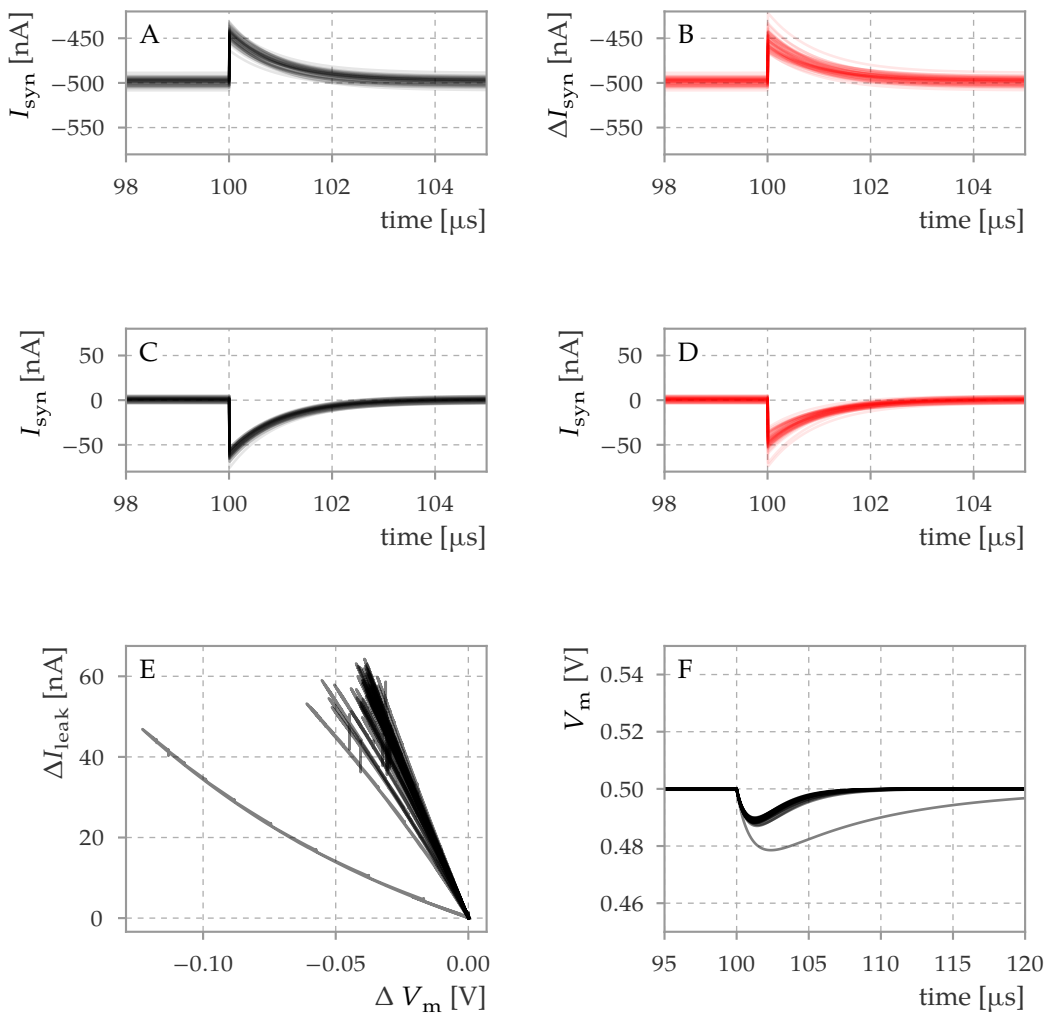


Figure A.7: Additional figures for synaptic weight calibration. **A – D**: Separated version of fig. 3.36 E with absolute values for the synaptic current. The excitatory input is calibrated to counterbalance an input current of 500 nA. **E**: The leak conductance is the slope of the change in leak current versus the change in membrane voltage. The outlier in the leak conductance is responsible for the outlier in the PSP (**F**). E and F show the data for the post-calibration membrane voltage from fig. 3.36.

A.10 Distribution of maximal synaptic input currents

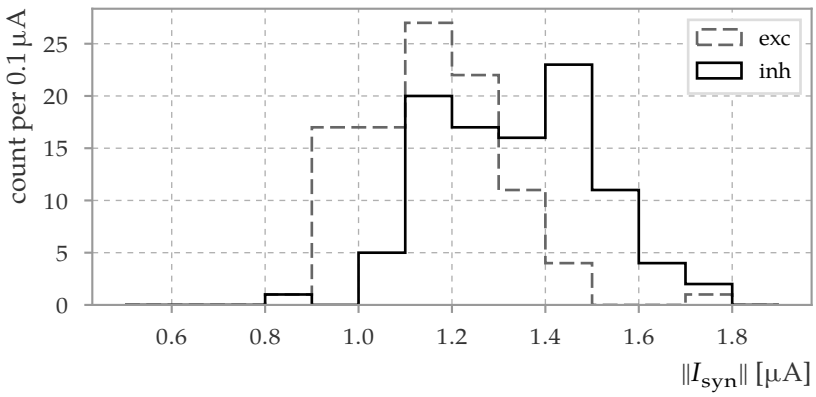
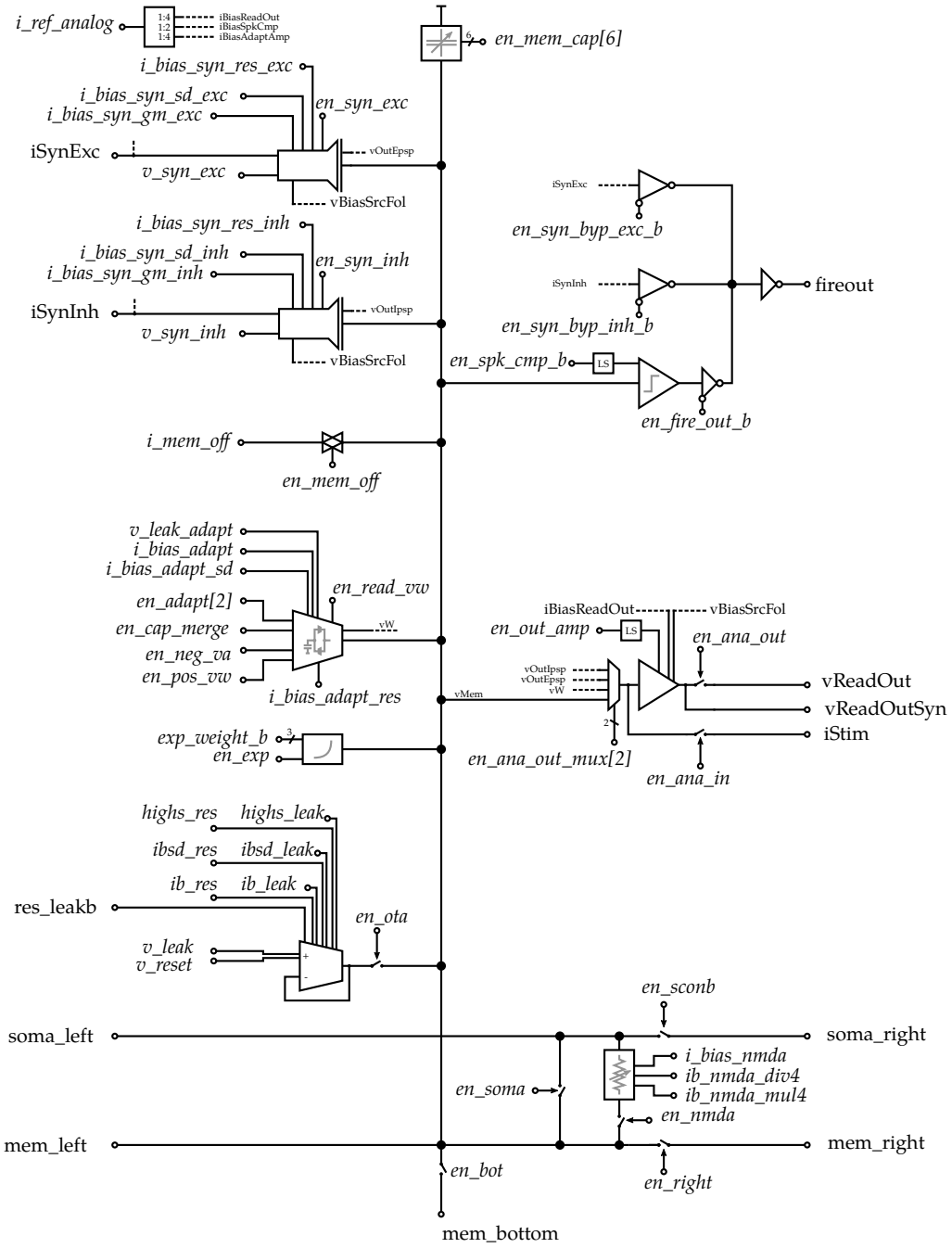


Figure A.8: Distribution of maximal synaptic currents. The output OTAs of the synaptic inputs are configured into saturation by setting *v_syn_[exc|inh]* = 1.6 V. $V_{\mathrm{m}} = 0.6\,\mathrm{V}$

A.11 DLS3 neuron schematic with configuration variables



Acknowledgements

I would like to thank

Prof. Karlheinz Meier for giving me the opportunity to work on neuromorphic hardware.

Prof. Michael Hausmann for agreeing to referee my thesis.

Manfred Stärk for making this work possible.

Björn Kindler for organizing.

My fellow visionaries, for creating our hardware: Aamir, Andreas, Andreas, Christian, Dan, Gerd, Johannes, Joscha, Korbi, Matthias, Maurice, Sebastian, Sebastian, Simon, Vitali.

My fellow visionaries, for building new theories: Akos, Agnes, Andreas, Anna, Carola, Christian, Dominik, Ilja, Johannes, Luziwei, Mihai, Oliver, Oskar, Venelin.

My fellow visionaries, for making it work: Arthur, Arno, Alexander, Bernhard, Benjamin, Christoph, Dan, Daniel, David, Eric, Jan, Sebastian, Sebastian, Johann, Johannes, Kai, Lukas, Lars, Christian, Mitja, Tom, Yannick.

Laura, for many things.

My family.

Statement of originality

I certify that this thesis, and the research to which it refers, are the product of my own work. Any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline.

Ich versichere, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, 2017-08-21 _____

Paul Müller