Dissertation

submitted to the Combined Faculty of
Mathematics, Engineering and Natural Sciences
of Heidelberg University, Germany
for the degree of

Doctor of Natural Sciences

Put forward by

Hartmut Schmidt

born in: Heilbronn, Germany

Oral examination: 06.02.2024

Large-Scale Experiments on Wafer-Scale Neuromorphic Hardware

Referees:
Dr. habil. Johannes Schemmel (Heidelberg University)
Prof. Dr. Wolfram Pernice (Heidelberg University)

Abstract

Neuromorphic hardware addresses the limitations of traditional computers, particularly in terms of power consumption and simulation speed when handling neural networks. The first-generation BrainScaleS system achieves this by physically implementing neurons and synapses with analog circuits, complemented by the utilization of wafer-scale integration to realize high circuit counts. However, both techniques come with the compromise of limited control over the system, constraining previous emulations to small network sizes.

This thesis introduces an optimized approach to hardware utilization that enables the execution of large-scale experiments. Techniques are developed that address hardware defects, reduce parameter variations through extended calibrations, increase neuron and synapse utilization by enhancing the routing capabilities, and bypass undesired circuit behaviors. Building upon these improvements, a precise model of hardware behavior is generated. This model serves as a foundation for aligning two large-scale biological networks with the inherent constraints of the hardware. To achieve this alignment, methods are developed that facilitate the necessary modifications while preserving biological behavior. By emulating these adapted network descriptions, the thesis demonstrates the system's capabilities for large-scale experiments and enables performance comparisons with other simulators.

Zusammenfassung

Neuromorphe Hardware begegnet den Einschränkungen traditioneller Computer, insbesondere hinsichtlich Energieverbrauch und Simulationsgeschwindigkeit bei der Verarbeitung neuronaler Netzwerke. Das BrainScaleS System der ersten Generation erreicht dies, indem es Neuronen und Synapsen physisch mithilfe analoger Schaltungen implementiert, ergänzt durch den Einsatz von Wafer-Scale Integration zur Realisierung hoher Schaltungszahlen. Beide Techniken gehen jedoch mit dem Kompromiss einer eingeschränkten Kontrolle über das System einher, was bisherige Emulationen auf kleine Netzwerkgrößen beschränkte.

Diese Arbeit stellt einen optimierten Ansatz zur Hardwarenutzung vor, der die Durchführung von großskaligen Experimenten ermöglicht. Es werden Techniken entwickelt, die Hardwaredefekte angehen, Parametervariationen durch erweiterte Kalibrierungen reduzieren, die Nutzung von Neuronen und Synapsen steigern, indem die Fähigkeit zur Routenfindung erhöht wird und unerwünschtes Schaltungsverhalten umgehen. Aufbauend auf diesen Verbesserungen wird ein präzises Modell des Hardwareverhaltens generiert. Dieses Modell dient als Grundlage für die Anpassung von zwei großskaligen biologischen Netzwerken an die inhärenten Einschränkungen der Hardware. Um diese Anpassung zu erreichen, werden Methoden entwickelt, die die notwendigen Modifikationen ermöglichen, während das biologische Verhalten erhalten bleibt. Durch die Emulation dieser angepassten Netzwerkbeschreibungen demonstriert die Arbeit die Fähigkeiten des Systems bezüglich großskaliger Experimente und ermöglicht Leistungsvergleiche mit anderen Simulatoren.

Contents

1. Introduction

The evolutionary success of the human species can be attributed to the remarkable features of the human brain. Serving as an immensely powerful computational system, it demonstrates a unique combination of efficiency and robustness that has no equal. These capabilities stem from its massive network structure, characterized by approximately 10^{11} parallel operational neurons interconnected by 10^{14} synapses [Azevedo et al. 2009; Pakkenberg et al. 2003].

Exploring its operational principles and underlying structure holds great promise. It not only has the potential to unveil diagnostic and treatment possibilities for diseases affecting the nervous system but also provides an opportunity to learn from its design, enabling the achievement of hardware implementations that surpass the capabilities of existing technologies.

To accomplish this, diverse models are developed to depict the intricacies of the brain at varying scales and levels of abstraction. While simple models still permit analytical evaluations, the utilization of simulations becomes inevitable with increasing model complexity [Gerstner et al. 2012]. However, as the model size increases, simulations conducted on conventional computers based on the von Neumann architecture encounter limitations. The necessity to distribute neural events introduces a substantial communication overhead, limiting the potential for parallelization [Zenke et al. 2014]. Consequently, reduced simulation speed and power efficiency are observed when simulating the numerous parallel operational components present in the human brain.

To overcome these limitations, the development of dedicated hardware architectures has been initiated [Mead 1989; Mead 1990]. With a focus on simulating spiking neural networks, these architectures promise reduced power consumption at accelerated speeds. Summarized under the term neuromorphic computing, various systems have been developed, ranging from specialized FPGA-based implementations to full-custom ASICs [Furber et al. 2014; Davies et al. 2018; Merolla et al. 2014; Moradi et al. 2018; Furber 2016; Schuman et al. 2017; Indiveri et al. 2011].

The first-generation wafer-scale neuromorphic hardware platform BrainScaleS-1 is such a system [Schemmel et al. 2010; Schemmel et al. 2008]. By physically implementing models of neurons and synapses through analog circuits, their dynamics are emulated, eliminating the need for numerical calculations. As a result of this, network dynamics are obtained at variable speedup factors between 10^3 and 10^5 compared to biological real-time. Additionally, high circuit counts are achieved through wafer-scale integration. Interconnecting 384 individual ASICs on a single wafer, each system comprises approximately 2×10^5 neuron circuits and 43×10^6 synapses.

However, both approaches come with trade-offs in terms of reliability and flexibility. Physical modeling imposes restrictions on the configurability of neuron parameters and

leads to parameter variations. Furthermore, the utilization of wafer-scale integration introduces constraints in addressing malfunctioning components. As a consequence of this, it is crucial to address and mitigate the inherent constraints of the system during hardware operation. Moreover, benchmarks are required to demonstrate the correct operation of the system and to facilitate comparisons with other simulators [Davies 2019].

In previous works, successful operation of the system could be demonstrated on networks comprising a small subset of neurons [Schmitt et al. 2017; Kungl et al. 2019; Göltz et al. 2021]. This thesis builds upon these prior efforts and concludes long-standing endeavors by demonstrating the system's full potential through the emulation of two large-scale biological models, the balanced random network [Brunel 2000] and the cortical microcircuit [Potjans et al. 2012].

Based on idealized assumptions, the balanced random network is analytically traceable and explores various states of spiking neural network behavior. Therefore, it forms the foundation for several biologically plausible network descriptions and is utilized in this thesis to investigate the system's behavior and limits. Building upon the obtained results, the cortical microcircuit is implemented. Representing the sub-surface structure of approximately $1\,\mathrm{mm}^2$ of the cerebral cortex of mammal brains, it serves as a typical benchmark that has been recently implemented on various simulators [Albada et al. 2018; Rhodes et al. 2020; Knight et al. 2021; Golosio et al. 2021]. Consequently, it allows for a comparison of the system.

Given the limitations of the hardware, a direct implementation of the network structures, as outlined in their respective publications, proves unfeasible. To this end, this thesis adopts a twofold strategy to address these constraints.

Firstly, techniques are developed to alleviate the inherent limitations of wafer-scale neuromorphic hardware, thereby facilitating the execution of large-scale experiments. A key aspect of this development involves the implementation of defect management aimed at enhancing system reliability. Furthermore, significant progress is made in improving control and predictability of neuron parameters by extending the system's calibration routines. Additionally, the utilization of the system for neural networks is enhanced through the refinement of its routing capabilities. Simultaneously, concepts are developed and applied to mitigate undesired effects observed during hardware operation. Conclusively, leveraging the insights derived from the results obtained, a precise model characterizing the hardware's behavior is constructed.

In the second part, this model is employed to modify the investigated network descriptions, aligning them with the constraints of the system. These modifications include reducing neuron and synapse counts to fit within a single wafer system, transitioning the synapse model, and adjusting neuron parameters to comply with hardware limitations. Given that the network behavior inevitably changes under these modifications [Albada et al. 2014], software simulations are conducted using the NEST simulator [Gewaltig et al. 2007]. Based on these simulations, techniques are developed that facilitate the transition of the networks while preserving biologically plausible characteristics, as defined in Potjans et al. 2012.

Emulating the adapted models on the hardware while preserving these characteristics demonstrates the functionality of the system. Furthermore, obtaining similar behavior

between the simulation and emulation serves as a validation for the correctness of the hardware model.

Utilizing these results, comparisons can be drawn with other simulators, showcasing the distinct advantages of the techniques employed in the BrainScaleS-1 system. Simultaneously, any shortcomings identified offer valuable insights for advancing the development of future large-scale neuromorphic systems.

Outline

The thesis is organized into seven chapters. Chapter 2 introduces the fundamental concepts of spiking neural networks and presents the neuron model adopted in this work. Additionally, it provides an overview of the two networks under investigation.

Chapter 3 explains the structure and operation of the BrainScaleS-1 system, thereby providing essential details and concepts for subsequent chapters.

Building upon this, chapter 4 delves into the enhancements made to the system for large-scale experiments. This encompasses the availability management to address hardware defects, dedicated calibration routines, improvements to network routing, and strategies for bypassing undesired hardware behavior.

Moving forward, the implementations of the two network models on the hardware are presented, separated into the implementation of the balanced random network in chapter 5 and the cortical microcircuit in chapter 6. In the initial part of each chapter, the focus lies on aligning the NEST simulation of the model with the constraints imposed by the hardware. Following this, the second part discusses the emulation of the adapted models on the hardware.

Finally, in chapter 7, the performance of the system is analyzed and compared to other simulators. Moreover, conclusions are drawn, offering insights for future large-scale emulations.

2. Biological Models

The human brain represents an immensely efficient, robust, and powerful computation system [Mead 1990]. Gaining a profound understanding of its structure and fundamental operational concepts not only holds the potential to unveil new possibilities for diagnosing and treating nervous system-related diseases but also enables learning from its design to construct hardware that may surpass the capabilities of existing systems.

One approach to gain more insights into the human brain is to construct systems that imitate its operational principles, thereby facilitating simulations of brain dynamics. However, because of the immense complexity of its fundamental building block, the neuron, modeling it in all its details is computationally costly. Therefore, simplified neuron models, which focus on mirroring key features of their archetype, are utilized. By doing so, simulations of large network models investigating structures found in the human brain are achievable, which is this thesis' focus.

This chapter provides the necessary background for the biological models investigated. It starts with a summary of the fundamental operational principles of the biological neuron in section 2.1. Building upon it, section 2.2 introduces the LIF neuron, which constitutes the simplified neuron model implemented in the network structures of this thesis. Moving forward, section 2.3 shows the network characteristics necessary to evaluate and compare the behavior of the investigated networks. Finally, the two biological network models investigated in this thesis are presented: the balanced random network model in section 2.4 and the cortical microcircuit model in section 2.5.

2.1. The Biological Neuron

As per Azevedo et al. 2009, the human brain consists of approximately 86×10^9 neurons, which, according to Pakkenberg et al. 2003, are interconnected by approximately 15×10^{13} synapses. Besides its immense network structure, the properties of its basic building blocks, the neurons, constitute to its efficiency, robustness, and learning capabilities. To facilitate the implementation of brain-like networks and to justify the adoption of simpler neuron models like the LIF neuron, introduced in section 2.2, this section provides an overview of the fundamental characteristics of the biological neuron. A more detailed summary of the biological principles is given in Alberts et al. 1994.

The biological neuron is a cell that is specialized to propagate electric pulses. Like any other cell in the human body, it contains organelles like the nucleus or mitochondria. However, as they are common in all cells and have no particular role in the transmission of signals, they are not further introduced in this consideration.

Characteristic for the neuron is its spatial structure, which is basically split into three sections: dendritic tree, soma, and axon. A sketch depicting the fundamental

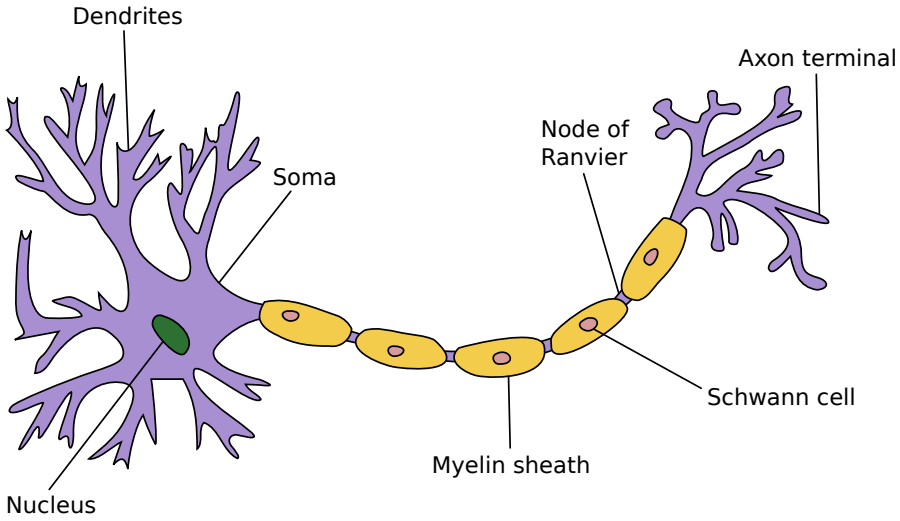


Figure 2.1.: Schematic of a neuron cell. At the dendrites, incoming signals are collected and transmitted to the soma, the main cell body that hosts the nucleus. There, the signals are accumulated, and, if strong enough, a spike is generated and sent along the axon. The transmission speed is increased by myelin sheaths that isolate the neuron, therefore forming a saltatory conduction where the spike is only regenerated at the nodes of Ranvier. At the axon terminals at the end of the axon, the neuron connects via synapses to the dendrites of other neurons. Adapted from Jarosz 2009.

components essential for signal transmission within the brain is shown in fig. 2.1. Signals are represented by differences in the electric potential between the inside and the outside of the neuron. This potential, separated by the neuron's membrane, is referred to as the membrane potential. It is defined by the concentration of different ions, mainly sodium Na^+, potassium K^+, calcium Ca^{2+} and chloride Cl^-. These ions can be exchanged between the inside and outside of the cell through either passive or actively driven ion channels hosted in the membrane. In the absence of any stimulation, the electrical and chemical potential, along with the permeability of the ion channels, establish an equilibrium value for the membrane potential, the so-called resting potential. For most neurons, this is approximately $-70\,\mathrm{mV}$ but it differs depending on the purpose of the neuron.

If the potential difference caused by stimulations accumulated at the soma exceeds a threshold, which is normally $-55\,\mathrm{mV}$, a so-called action potential or spike is triggered. At this point, a runaway process is started and the permeability of the Na^+ ion channels increases such that the membrane potential is drastically rising and the neuron is depolarized. After less than 1 ms, the Na^+ channels close and the slower K^+ channels open and the neuron repolarizes by releasing K^+ ions. As the K^+ channels do not close immediately when reaching the resting potential, the neuron enters the hyperpolarization, where its membrane potential is below the resting potential and the neuron is less

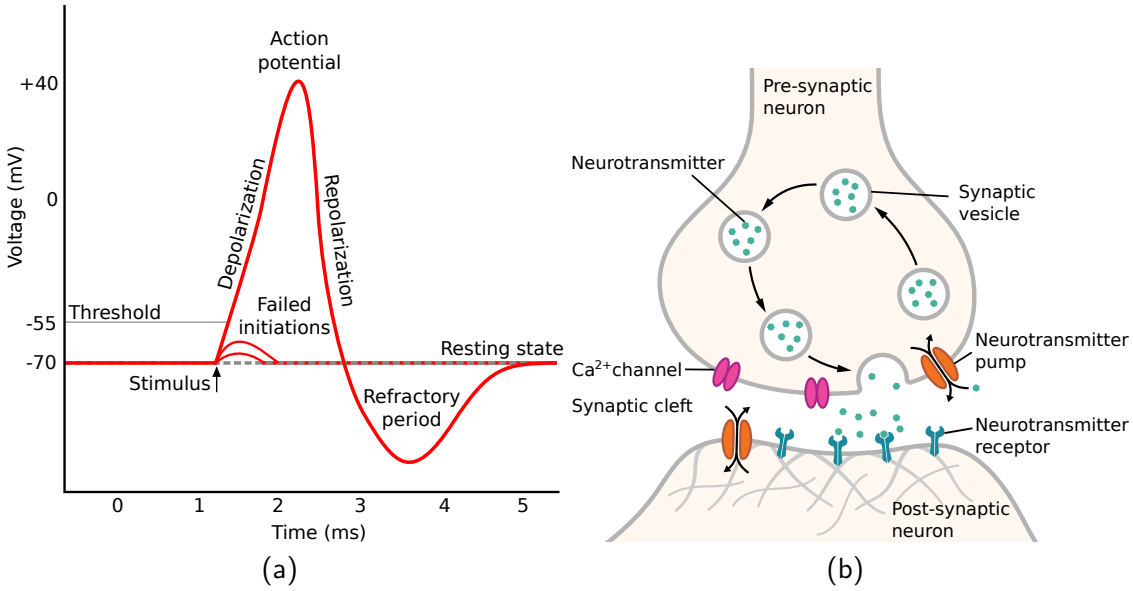


Figure 2.2.: Spike-based signal transmission in biological neurons. (a) Shape of an action potential, also called spike, generated at the soma of the neuron. Starting from the resting potential, stimuli are accumulated until the threshold voltage is reached and a spike is triggered. This starts a process where the membrane potential is first depolarized and after approximately 1 ms repolarizes again, followed by the hyperpolarization in the refractory period, during which the neuron is less excitable. Taken from Chris 73 et al. 2007. (b) Schematic of a chemical synapse. Synapses transmit signals from the pre-synaptic neuron to the postsynaptic neuron. If a spike arrives at the pre-synaptic part, neurotransmitters stored in vesicles are released into the synaptic cleft that separates the two neurons. These neurotransmitters bind to receptors in the postsynaptic neuron. As a result of this, ion-channels are opened, triggering a voltage change in the postsynaptic neuron. Taken from Splettstoesser 2015

excitable. This is called the refractory period and typically lasts for a few milliseconds. The voltage trace of a spike is sketched in fig. 2.2a.

This spike then propagates along the axon of the now so-called spiking neuron. Myelin sheaths covering and insulating the axon speedup the transmission as the spike is only refreshed in myelin-sheath gaps, the nodes of Ranvier, which are highly enriched in ion channels. During its propagation, the spike is distributed on different branches of the axon, thereby reaching on average approximately 7000 end points, so-called axon terminals [Drachman 2005]. This count significantly varies based on the specific function of the neuron.

At the axon terminal, connections to either the dendritic tree of other neurons or to other cells like muscles are established. These connections are called synapses, which, due to their ability to modify the transmitted signal, significantly contribute to the learning

capabilities of the brain. To achieve this, complex underlying structures and processes within synapses are required, which are still under current research. Here, only a short overview of the basic principles of synapses is given sufficient for the studies of this thesis. For a more comprehensive overview, the reader is referred to Cowan et al. 2003.

Synapses form a directed transmission of information from the pre-synaptic neuron, where the spike is generated, to the postsynaptic neuron. Although some synapses implement an electrical transmission of signals, in most synapses a chemical process takes place. Figure 2.2b sketches the process of signal transmission in such a chemical synapse. If a spike arrives at the pre-synaptic part of the synapse, the changed membrane potential leads to an influx of Ca^{2+} ions via the voltage gated ion channels. These Ca^{2+} ions bind to the synaptic vesicles present in the pre-synaptic neuron. These vesicles carry neurotransmitters that act as chemical messengers. As a result of the binding, the vesicles fuse with the pre-synaptic membrane, releasing their neurotransmitters into the synaptic cleft, the gap between the two neurons. These neurotransmitters then bind to receptors at the postsynaptic neuron, thereby opening ion gates there. The resulting voltage change in the postsynaptic neuron is called postsynaptic potential (PSP). Depending on the type of neurotransmitters involved and the ion channels that are activated in the synapse, the membrane voltage either increases or decreases. As an increase in the PSP facilitates the emergence of a spike in the postsynaptic neuron such a synapse is called excitatory. In contrast, a synapse that decreases the PSP inhibits the postsynaptic neuron and is therefore called inhibitory. With a few exceptions, in biology, a neuron releases the same set of neurotransmitters in all synapses where it contributes as the pre-synaptic partner [Eccles et al. 1954]. Consequently, as a rule of thumb, a neuron appears either excitatory or inhibitory to all its connected neurons. This generalization is referred to as Dale's law. Moreover, by changing its structure over time and thereby the strength of the stimulation at the postsynaptic neuron, synapses are able to adapt to the tasks they are faced with.

All in all, the interactions of neurons and synapses in the human brain are complex and this introduction only forms a strongly simplified summary of the real processes focused on the basic network functions necessary to understand this thesis' investigations. Using the presented principles, in the next section, a simplified neuron model is introduced, which reduces the computational overhead and thereby allows for implementing large-scale networks.

2.2. The Leaky Integrate-and-Fire Neuron

The replication of biological neuron behavior is computationally expensive due to its tremendous complexity. However, many properties found in biology are not necessary to imitate the basic neuron behavior found in the brain. Depending on the functions that are investigated, simplifications can be made, resulting in different abstractions of the neuron model. Therefore, a variety of different descriptions exist, ranging from the close-to-biology Hodgkin-Huxley neuron [Hodgkin et al. 1952], which still models individual ion channels, to the McCulloch-Pitts cell [McCulloch et al. 1943], an idealized

model often used in artificial neural networks.

In this thesis, biological networks using the LIF neuron are investigated. This model boils down the neuron dynamic to its very basic principles and is therefore extensively used in computational neuroscience [Brunel et al. 2007]. An introduction to the model is provided in this section. For an in-depth analysis, reference is made to Petrovici 2016.

The LIF neuron is a point neuron model. In contrast to multi compartment approaches [London et al. 2005], which replicate the spatial structure of biological neurons, any signal transmissions inside the neuron are neglected and the neuron is treated without extension. In addition, since it is observed that all action potentials in biological neurons have approximately the same shape, it is assumed that the only relevant information is their timing. Therefore, the internal dynamics of the neuron are drastically simplified.

A circuit diagram of the model is depicted in fig. 2.3. There, the membrane potential is abstracted by the voltage across a capacitor C_m. The resting potential of the neuron, which the membrane potential approaches in the absence of any stimulus, is implemented by a voltage source E_rest that is connected in parallel to the capacitor via a resistor. As this resistor resembles leak currents across the cell membrane, its conductance is called leak conductance g_leak. Stimulations the neuron receives via the dendritic tree are modeled by a time-dependent synaptic input current $I_\mathrm{syn}(t)$. There are two approaches to generate this current as a response to an incoming spike: current- or conductance-based. Figure 2.3 shows the LIF neuron with conductance-based synapses. A corresponding circuit diagram for current-based synapses is obtained by replacing I_syn with a time-dependent current source. Both possibilities are discussed in more detail in the second part of this section.

In summary, the time evolution of the membrane voltage U of the LIF neuron below the threshold voltage can be described by the differential equation

$$C_\mathrm{m}\frac{dU}{dt} = -g_\mathrm{leak}\left(U - E_\mathrm{rest}\right) + I_\mathrm{syn}(t). \tag{2.1}$$

Furthermore, the decision whether the neuron fires, i.e., elicits a spike, is made by a comparator, which compares the membrane potential to the threshold voltage V_thres. In case the threshold value is reached, a spike is generated and distributed to all connected neurons. In addition, a voltage source is short-circuited to the capacitor, forcing the membrane to the reset potential V_reset. This connection is established for a time period of τ_refrac, reproducing the refractory time during the hyperpolarization of the neuron after emitting a spike. In short, the fire mechanism is expressed as

$$U(t) = V_\mathrm{reset} \quad \text{for} \quad t \in (t_s, t_s + \tau_\mathrm{refrac}) \quad \text{if} \quad U(t_s) = V_\mathrm{thres}. \tag{2.2}$$

2.2.1. Current-Based Synapses

Current-based synapses form one of two possibilities to model the synaptic input current of the LIF neuron. There, a time-dependent current source is used to model the inputs the neuron receives. At first glance, this is in contrast to biology, where a PSP is generated by

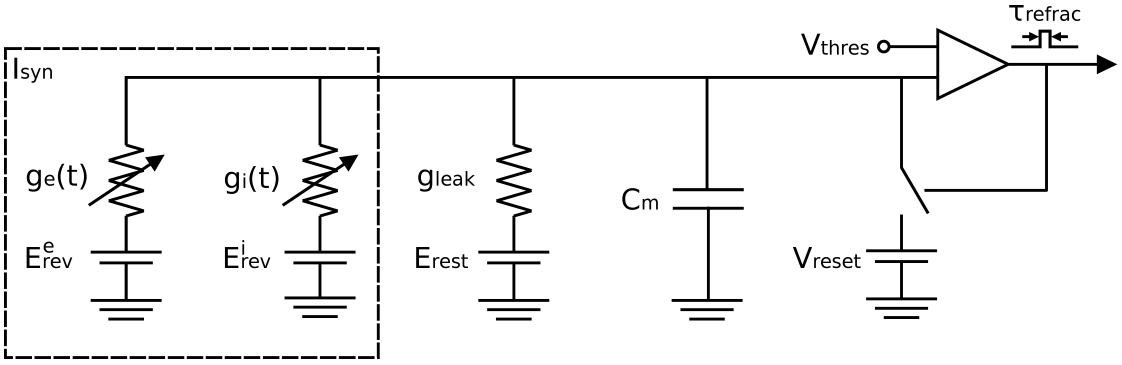


Figure 2.3.: Circuit diagram of the LIF neuron with conductance-based synapses. The membrane of the neuron is modeled by a capacitor C_m that is connected via the leak conductance g_leak to the resting potential E_rest. A comparator compares the voltage of the capacitor to the threshold voltage V_thres. If it is exceeded, a spike is sent and the reset potential V_reset is short-circuited to the capacitor for the time period τ_refrac. The synaptic input current I_syn is implemented as two time-dependent conductances $g_\mathrm{e}(t)$ and $g_\mathrm{i}(t)$ that either connect the membrane to the excitatory reversal potential $E^\mathrm{e}_\mathrm{rev}$ or the inhibitory reversal potential $E^\mathrm{i}_\mathrm{rev}$, respectively.

a change in the conductance of the neuron's membrane. However, in the LIF model, the spatial structure of the neuron and therefore the signal transmission from the dendrites to the soma is neglected. Considering the neuron from the point of view of the soma, the conductance-based behavior at the dendrites can be ignored for distant synapses, and the incoming PSPs resemble an input current. Furthermore, compared to conductance-based synapses, input currents are easier to implement and simplify the analytical solution of the membrane behavior.

In this case, the synaptic input current $I_\mathrm{syn}(t)$ in eq. (2.1) is given by

$$I_\mathrm{syn} = \sum_k \sum_s w_k \epsilon(t - t_s), \tag{2.3}$$

where the first sum iterates over all synapses k stimulating the neuron and the second sum iterates over all spikes s transmitted via each synapse. The weight w_k determines the strength of the stimulation caused by synapse k that receives a spike at time t_s. It is positive for excitatory connections and negative for inhibitory ones. Finally, ϵ is the synaptic kernel, which determines the temporal course of the synaptic input current. Different functions can be used to imitate different synaptic behavior.

In this thesis, the delta peak kernel and an exponentially decaying kernel are used. For the delta peak, the whole charge is immediately transmitted to the postsynaptic neuron, therefore modeled by a delta peak

$$\epsilon(t) = \delta(t). \tag{2.4}$$

While serving as a practical simplification for analytical considerations, it lacks biological plausibility and can only be approximated in physical models. In contrast, the exponentially decaying kernel more accurately resembles currents in the neuron's soma. It is described by

$$\epsilon(t) = \Theta(t) \exp\left(-\frac{t}{\tau_{\text{syn}}}\right), \tag{2.5}$$

where τ_{syn} is the synaptic time constant of the exponentially decaying current. In addition, the Heaviside step function Θ ensures that current is only flowing after the pre-synaptic neuron spiked.

Inserting the synaptic input current of the current-based model with an exponentially decaying kernel into eq. (2.1), according to Petrovici 2016, the analytic solution

$$U(t) = E_{\text{leak}} + \sum_k \sum_s \frac{\tau_{\text{syn}}^k w_k}{g_{\text{leak}}\left(\tau_{\text{syn}}^k - \tau_{\text{m}}\right)} \Theta(t - t_s) \left(\exp\left(-\frac{t - t_s}{\tau_{\text{syn}}^k}\right) - \exp\left(-\frac{t - t_s}{\tau_{\text{m}}}\right)\right) \tag{2.6}$$

of the neuron's membrane voltage time course is found, with the membrane time constant $\tau_{\text{m}} = \frac{C_{\text{m}}}{g_{\text{leak}}}$. The shape of a PSP stimulated by a single spike is visualized in fig. 2.4a.

2.2.2. Conductance-Based Synapses

The other possibility to model the synaptic input current of the LIF neuron are conductance-based synapses. By adapting the conductance towards a so-called reversal potential, this approach imitates the behavior of the membrane in the synapses.

The circuit diagram of the model is shown in fig. 2.3. Two additional voltage sources imitate the potential difference over the neuron's membrane seen by the different ions. The excitatory reversal potential $E_{\text{rev}}^{\text{e}}$ resembles primarily the sodium channels and the inhibitory reversal potential $E_{\text{rev}}^{\text{i}}$ the potassium channels. Moreover, the permeability of the membrane is modeled by time-dependent conductances, which connect the membrane to the two reversal potentials. For excitatory inputs, the excitatory conductance $g_{\text{e}}(t)$ is increased and for inhibitory inputs the inhibitory conductance $g_{\text{i}}(t)$. Consequently, the synaptic input current is expressed as

$$I_{\text{syn}} = g_{\text{e}}(t)\left(E_{\text{rev}}^{\text{e}} - U(t)\right) + g_{\text{i}}(t)\left(E_{\text{rev}}^{\text{i}} - U(t)\right), \tag{2.7}$$

with the temporal behavior of the conductances

$$g_x = \sum_{k_x} \sum_s w_{k_x} \epsilon(t - t_s) \quad \text{for} \quad x \in \{\text{e}, \text{i}\}. \tag{2.8}$$

Again, k and s represent the synapses and their spikes. However, this time they are divided into excitatory and inhibitory synapses, each exclusively contributing to the respective conductance. Additionally, it is worth mentioning that, in contrast to the current-based model, the weight w_k is given in Siemens instead of Ampere.

Similar to the current-based case, different temporal behavior of the conductances can be modeled by different kernels. In this thesis, exclusively exponentially decaying conductances are used, represented by the kernel introduced in eq. (2.5).

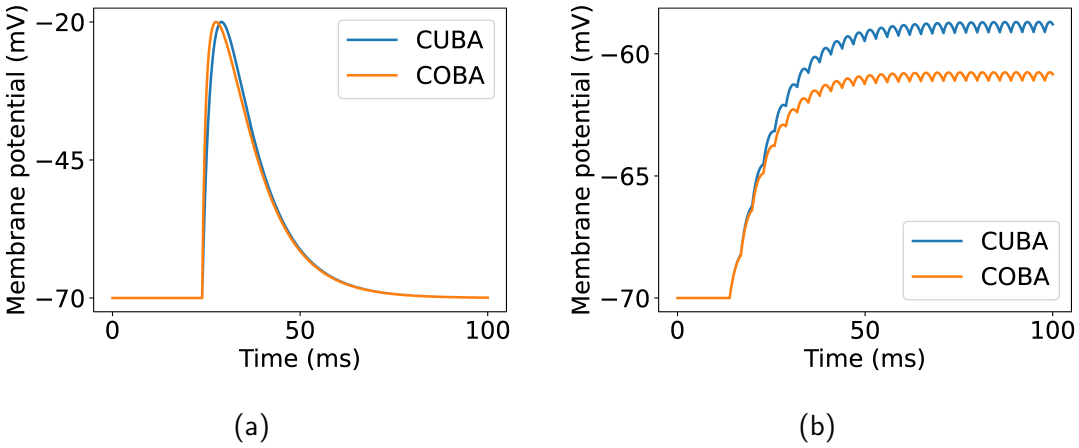


Figure 2.4.: Comparison of current-based and conductance-based synapses. Both figures are generated with the NEST simulator. (a) Voltage trace of a LIF neuron with either current-based (CUBA) or conductance-based (COBA) synapses that is stimulated by a single input spike. The weight of the stimulating synapse is chosen such that both PSPs have the same height. To visualize the adaptation of the effective membrane time constant, a biologically implausible high weight and a relatively low excitatory reversal potential is used. Consequently, high-conductances are reached in the conductance-based synapse and the membrane potential changes slightly faster compared to the current-based synapse. Used parameters are listed in table A.1. (b) Stacking of PSPs. The measured neuron receives spikes with a fixed interval and the resulting PSPs add up to the membrane potential. For larger distances between membrane potential and resting potential, the leak current rises and reduces the voltage increase until an equilibrium is reached. While the first spike achieves a similar PSP height in both synapse models, the stimulation strength decreases for the conductance-based synapse as the membrane potential approaches the excitatory reversal potential. Consequently, the membrane saturates at a lower voltage level. Used parameters are listed in table A.2.

Modeling the synaptic input current with time-dependent conductances to the reversal potentials results in two effects. On the one hand, incoming spikes change the total conductance

$$g_{\text{tot}} = g_{\text{leak}} + \sum_{\text{exc syn } m} g_{\text{e}}^{m} + \sum_{\text{inh syn } n} g_{\text{i}}^{n} \tag{2.9}$$

of the membrane, which results in an increased effective membrane time constant of

$$\tau_{\text{eff}} = \frac{C_{\text{m}}}{g_{\text{tot}}}. \tag{2.10}$$

For strong stimulation, this leads to a faster reacting membrane potential compared to the current-based neuron, visualized in fig. 2.4a. On the other hand, the synaptic current explicitly depends on the distance between the current membrane potential and the reversal potentials. This effect is demonstrated in fig. 2.4b. There, starting with similar PSP heights in both synapse models, the strength of received stimulations reduces in the conductance-based case the more the membrane potential approaches the reversal potential. Combining both effects, reduced deviations from the resting potential and a faster reacting membrane are observed, which leads to a smaller dynamic range with fewer fluctuations for conductance-based synapses.

Furthermore, due to this increased complexity, no analytical solution to the membrane dynamic can be found. However, considering a strongly stimulated neuron, the dynamic of the model simplifies as the synaptic conductances are much larger compared to the leak conductance and the total conductance is found to be approximately constant [Petrovici 2016]. This state of the neuron is called high-conductance state and is either achieved by strong input weights or a high input rate. Assuming a constant total conductance $\langle g_{\text{tot}} \rangle$, the membrane time constant in eq. (2.6) can be replaced by a constant effective membrane time constant

$$\langle \tau_{\text{eff}} \rangle = \frac{C_{\text{m}}}{\langle g_{\text{tot}} \rangle}. \tag{2.11}$$

This substitution leads to the closed-form solution

$$\text{PSP}(t) = \frac{\tau_{\text{syn}} \langle \tau_{\text{eff}} \rangle w \left(E_{\text{rev}} - \langle U \rangle \right)}{C_{\text{m}} \left(\tau_{\text{syn}} - \langle \tau_{\text{eff}} \rangle \right)} \Theta(t - t_{\text{s}}) \left(\exp\left(-\frac{t - t_{\text{s}}}{\tau_{\text{syn}}} \right) - \exp\left(-\frac{t - t_{\text{s}}}{\langle \tau_{\text{eff}} \rangle} \right) \right) \tag{2.12}$$

for a PSP of a conductance-based neuron in the high-conductance state starting from its average membrane potential $\langle U \rangle$ that is stimulated by a single additional spike [Petrovici 2016]. Dependent of the considered synapse type, E_{rev} represents the excitatory or inhibitory reversal potential.

Consequently, in the high-conductance state, the behavior of the conductance-based model differs from the current-based model. Reaching this state is hindered by utilizing small weights and less stimulating synapses. Considering a neuron without previous stimulation that is stimulated by a single spike with low weight, the effective membrane time constant $\langle \tau_{\text{eff}} \rangle$ can be replaced with the unmodified membrane time constant τ_{m} in eq. (2.12) as the change in conductance can be neglected. The obtained membrane behavior resembles a current-based neuron with weight $w / \left(E_{\text{rev}} - \langle U \rangle \right)$.

2.3. Network Characteristics

Building upon models of individual neurons, networks can be constructed, consisting of multiple neurons, providing valuable insights into the functioning of the brain. However, to thoroughly investigate and compare these networks, the utilization of appropriate measurement variables is essential. Although the neuron's membrane potential can be measured, for example using the patch clamp technique [Sakmann et al. 1984], the most prominent observable in the brain is the spike output of the neurons. Therefore, the networks investigated in this thesis are focused on the analysis of the spikes emitted by their neurons. Besides the timing of the spikes, two characteristics of the spiking behavior of the neurons are considered, the irregularity and the synchrony. Both are discussed in this section, subsequent to a short introduction into the most important terms used during the evaluation.

All considerations presented in this section are based on the spike times of investigated neurons. The collection of all outgoing spikes s of a single neuron is called spike train and is defined by

$$\rho(t) = \sum_s \delta(t - t_s). \tag{2.13}$$

An often used quantity to describe such a spike train is the inter-spike interval, i.e., the time between consecutive spikes. Its nth element is calculated by

$$\mathrm{ISI}_n = t_{n+1} - t_n \tag{2.14}$$

and its mean value by

$$\overline{\mathrm{ISI}} = \frac{1}{N-1} \sum_{n=1}^{N-1} \mathrm{ISI}_n, \tag{2.15}$$

with N being the total number of spikes in the spike train.

A related quantity is the mean firing rate

$$\overline{\nu} = \frac{N}{T}, \tag{2.16}$$

where T is the total measurement time. For sufficiently long spike trains or high firing rates, it is equal to the inverse of the inter-spike interval

$$\overline{\nu} = \frac{1}{\overline{\mathrm{ISI}}} \tag{2.17}$$

as the time before the first spike and after the last spike becomes negligible.

2.3.1. Irregularity

The irregularity is a measure of the variation observed in the spiking behavior of a single neuron. It is described by the coefficient of variation (CV) of its inter-spike interval, which is calculated by

$$\mathrm{CV}(\mathrm{ISI}) = \frac{\sigma(\mathrm{ISI})}{\overline{\mathrm{ISI}}}, \tag{2.18}$$

where $\sigma(\mathrm{ISI})$ denotes the standard deviation of the inter-spike interval. For an irregularity of 0, the neuron spikes perfectly regular with all spikes being equidistant. For larger irregularities, the variation of the spike times increases. E.g., a spike train that is generated by a Poisson process has an irregularity of 1.

Moreover, the irregularity is also used to characterize the behavior of a collection of neurons, a so-called population. There, the mean value

$$\overline{\mathrm{CV}} = \frac{1}{N} \sum_{n=1}^{N} \mathrm{CV}_n(\mathrm{ISI}), \tag{2.19}$$

of the irregularity of all neurons N is used to describe the firing behavior of the whole population.

2.3.2. Synchrony

The synchrony classifies the correlation between neurons and is therefore a measurement parameter concerning an entire population of neurons. One method to determine the synchrony, introduced in Brunel 2000, is based on the evaluation of global firing behavior and holds under the assumption of randomly and sparsely connected neurons. This means that the number of connections between a pair of neurons C is much smaller than the total number of neurons N in the network. In the limit of $C/N \to 0$, the correlations between neurons due to shared inputs can be neglected. Consequently, any correlations are caused by the shared network behavior, which, for a random point process, is described by the instantaneous firing rate

$$\nu(t) = \lim_{\Delta t \to 0} \frac{P(t, t + \Delta t)}{\Delta t} \tag{2.20}$$

of its neurons, where $P(t, t+\Delta t)$ is the probability for a spike to occur in the time interval between t and $t + \Delta t$. Since neurons that share a common instantaneous firing rate are correlated, the temporal behavior of the instantaneous firing rate of all neurons is a measure of the synchrony of the network. If it is constant, the global network behavior is not changing over time and the neurons spike independently of each other. This network state is referred to as asynchronous. In contrast, if the global instantaneous firing rate varies, the neurons adopt a common firing behavior and are therefore correlated. This state is termed synchronous.

The spike data evaluated in this thesis is based on implementations of network models with a finite number of elements. Therefore, idealized network behavior, which is necessary for analytical solutions, does not necessarily hold and an experimental approach is pursued to evaluate the global firing behavior of the neurons. For this, the spike times of all investigated neurons are gathered in a histogram and the ratio between the variance and mean of the resulting bin heights is employed as an indicator of the global firing pattern. High values indicate strong fluctuations of the global spike count compared to its average value and thereby represent variations of the instantaneous firing rate, which is accompanied by synchronous network behavior. In contrast, small values are found for a constant global network behavior, which is obtained for asynchronously spiking neurons.

A disadvantage of this approach is that the absolute values of the synchrony depend on the chosen bin width and the number of investigated neurons. Therefore, an appropriate parametrization has to be found, which is discussed in more detail in section 5.1.1 and section 6.1.1. Moreover, for comparison reasons, equal values have to be chosen in all implementations of the model.

2.4. The Balanced Random Network Model

The enormous capabilities of the human brain strongly rely on the properties of its basic building blocks, the neurons and synapses. However, arguably even more important is its immense network structure. Therefore, to understand and benefit from the brain's working principles, the investigation of its connections and network behavior is essential. For this, in recent years, network models have been created based on connectivity principles found in animal brains [Bragin et al. 1995; Bassett et al. 2018]. One important model, especially for the investigation of the network behavior of large brain areas, is the balanced random network model. Focused on the connectivity properties and firing behavior of sparsely connected LIF neurons, it forms the foundation of more complex models. Being one of the models that are investigated in this thesis, its network structure and firing characteristics are introduced in this section. More details about the model and the utilized analytical methods can be found in Brunel 2000.

2.4.1. Network Structure

The model is based on analytical considerations but also comprises a network description that resembles its theoretical requirements. It is built on the basis of sparsely and randomly connected LIF neurons, which are organized in two populations. An excitatory one with N_E neurons, which, in accordance with Dale's law, exclusively implement excitatory connections to their postsynaptic partners and an inhibitory one with N_I neurons, which only implement inhibitory connections. Each of these neurons is connected to a fixed number C of randomly chosen other neurons, from which $C_\mathrm{E} = \epsilon N_\mathrm{E}$ are excitatory and $C_\mathrm{I} = \epsilon N_\mathrm{I}$ are inhibitory. Sparsity is achieved by a large pool of available neurons compared to the number of implemented connections per neuron, i.e., $\epsilon \ll 1$. In addition, each neuron receives $C_\mathrm{ext} = C_\mathrm{E}$ connections from excitatory external neurons. These external neurons are not modeled themselves but are represented by spike trains, which are generated by independent Poisson processes with a fixed firing rate ν_ext. This firing rate is given by $\nu_\mathrm{ext} = \eta\nu_\mathrm{thr}$, in relation to

$$\nu_\mathrm{thr} = \frac{V_\mathrm{thres} - E_\mathrm{rest}}{w_\mathrm{ext} C_\mathrm{E} \tau_\mathrm{m}}, \tag{2.21}$$

which represents the frequency required for the external input to induce a single spike in the neuron over an infinite time period while sending equidistant spikes with a synaptic weight of w_ext in the absence of any internal stimulation.

Table 2.1.: Parameters of the balanced random network model. The external input rate parameter η, the transmission delay D, and the relation between excitatory and inhibitory weight g are not listed since they are modified during the investigation. The parameter notation introduced in section 2.2 is used.

Network parameter	Value	Neuron parameter	Value
N_E	10 000	τ_m	20 ms
N_ext	10 000	τ_refrac	2 ms
N_I	2500	V_thres	20 mV
C_E	1000	V_reset	10 mV
C_ext	1000	E_rest	0 mV
C_I	250	w_E	0.1mV
ν_ext	$\eta\nu_\mathrm{thr}$	w_I	$-gw_\mathrm{E}$

Based on anatomical studies, 80 % of the neurons are are chosen to be excitatory, implying that $N_\mathrm{E} = 4N_\mathrm{I}$. As a consequence of this, each neuron receives four times more excitatory internal connections than inhibitory ones.

The synaptic input is modeled by current-based synapses with a delta peak kernel. As this stimulation lacks temporal dependency, the weight w of each connection is characterized by the resulting PSP height. It has to be much smaller than the voltage necessary to reach the threshold V_thres of the neuron, i.e., $w \ll V_\mathrm{thres}$, such that the neuron has to accumulate many stimulations to elicit a spike. Moreover, the same weights are used for all excitatory and all inhibitory synapses, respectively. Both weights are connected via the parameter g, which is given by

$$g = \frac{-w_\mathrm{I}}{w_\mathrm{E}}, \tag{2.22}$$

where $w_\mathrm{E} > 0$ is the excitatory and $w_\mathrm{I} < 0$ the inhibitory weight. In addition, the external input matches the weight of the excitatory connections, i.e., $w_\mathrm{ext} = w_\mathrm{E}$. For the spike transmission, a delay D is added to the spike time. Therefore, spikes arrive at time $t_\mathrm{s} + D$ at the postsynaptic neuron, imitating the time it takes to travel from one neuron to another. Finally, the spiking behavior of the network is studied for different values of the parameters g, ν_ext, and D. A summary of all model parameters is listed in table 2.1.

2.4.2. Network Behavior

The network behavior of the balanced random network model can be obtained either analytically, using idealized network assumptions, or by simulations. With the aim of replicating the model on wafer-scale neuromorphic hardware, this section focuses on introducing network characteristics that allow for a comparison between software simulations and results obtained on the hardware. To this end, the firing behavior of the network is investigated for different external input frequencies given by the parameter η, transmission delays D, and relative strength of the inhibitory synapses g. Dependent on

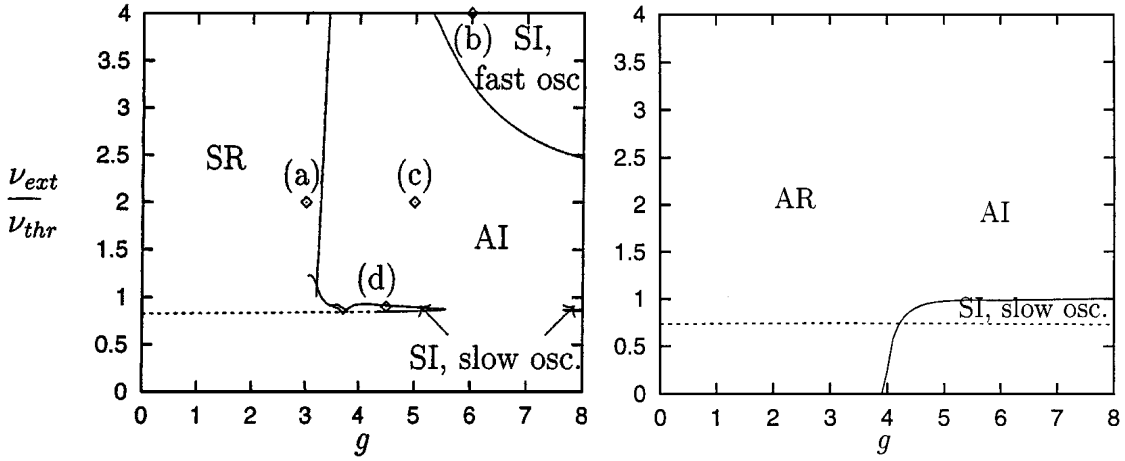


Figure 2.5.: Firing regimes of the balanced random network model. For different values of the external input rate ν_{ext} and the relative strength of the inhibitory weight g the network exhibits different firing regimes. Transitions between regimes are identified by Hopf bifurcation curves, shown as solid lines. SR indicates synchronous regular, AR asynchronous regular, SI synchronous irregular, and AI asynchronous irregular firing behavior. In the left panel, a fixed transmission delay of $D = 1.5\,\text{ms}$ is assumed for all neurons. Dependent on the external input frequency, the synchronous irregular regime occurs with fast and slow oscillations. Diamonds mark the parameters of the simulations visualized in fig. 2.6. In the right panel, the neurons' delays are uniformly distributed between 0 ms and 3 ms. As a consequence of this, the synchronous regular and synchronous irregular regime with fast oscillations are lost and the neurons spike asynchronously. Adapted from Brunel 2000. Reproduced with permission from Springer Nature.

these values, four different firing regimes, based on the network characteristics introduced in section 2.3, are observed, depicted in fig. 2.5. In addition, the firing patterns of chosen parameter sets are visualized in fig. 2.6.

For values of $g < 4$ the excitatory stimulation exceeds the inhibitory stimulation, since, in addition to the already exclusively excitatory external input, each neuron receives four times more excitatory internal connections. Due to the strong internal stimulation, the neurons reach nearly immediately their threshold and therefore spike approximately independently of the external input frequency close to their maximum frequency

$$\nu_{\max} = \frac{1}{\tau_{\text{refrac}}}, \tag{2.23}$$

only limited by the neurons' refractory period τ_{refrac}. This leads to an approximately equidistant inter-spike interval and the neuron behavior is regular.

Moreover, the global behavior of the neurons is found to be dependent on their

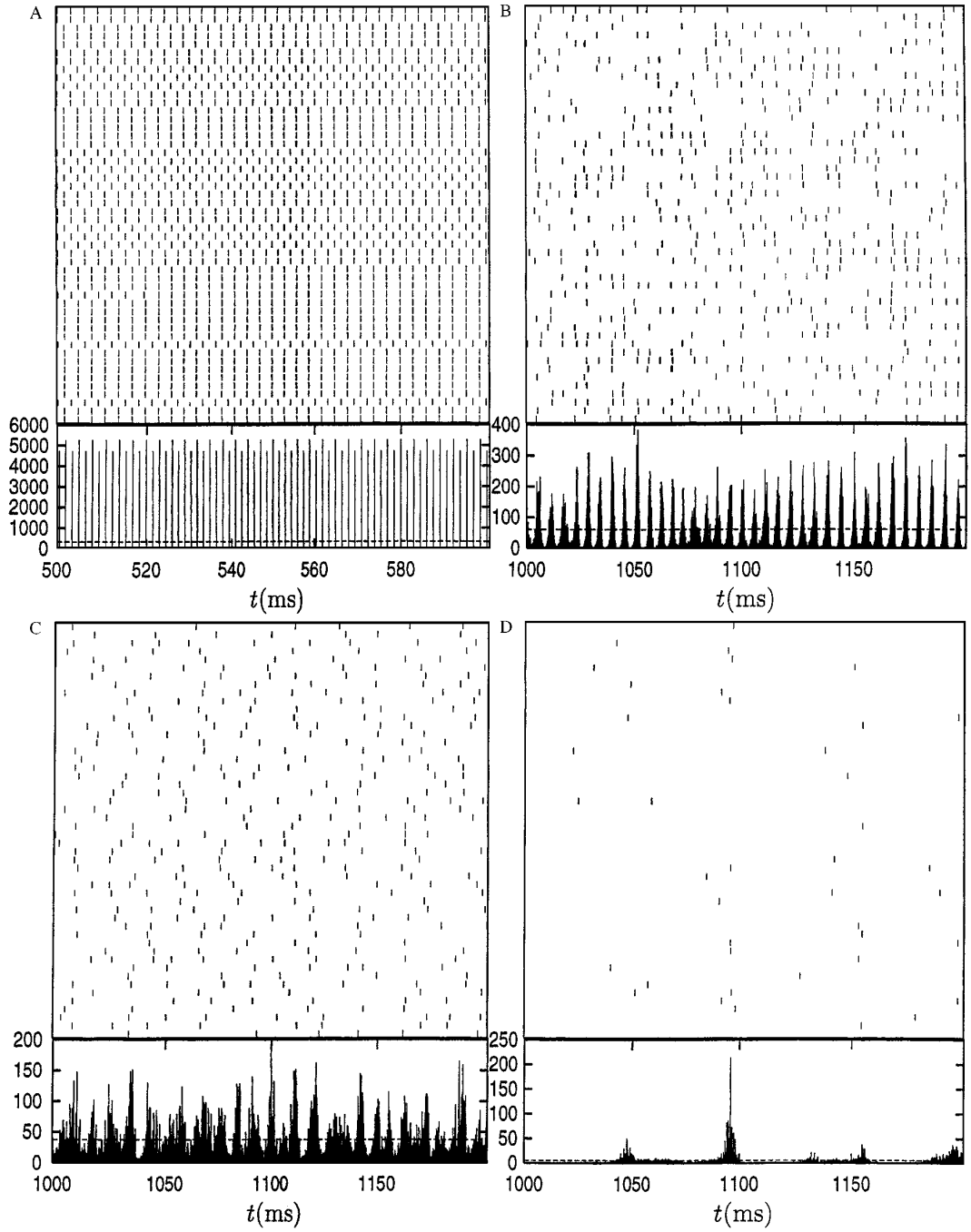


Figure 2.6.: Simulation results of the balanced random network in its different regimes. For each regime, the top part shows the spike times of 50 randomly chosen neurons and the lower part the histogram of all spikes with a bin width of 0.1 ms. A dashed line indicates the average firing activity. **A** synchronous regular (SR) at $g = 3$, $\eta = 2$. **B** synchronous irregular (SI) with fast oscillations at $g = 6$, $\eta = 4$. **C** asynchronous irregular (AI) at $g = 5$, $\eta = 2$. **D** synchronous irregular (SI) with slow oscillations at $g = 4.5$, $\eta = 0.9$. Taken from Brunel 2000. Reproduced with permission from Springer Nature.

transmission delay. If all neurons share the same delay value, they form groups with similar firing times, as observed in panel A of fig. 2.6. This is possible since all neurons share identical parameters and therefore behave identically. If many neurons spike close to each other, their accumulated inputs will cause other neurons to spike again with similar spike times. Consequently, for a delay value of $D = 1.5\,\mathrm{ms}$, independent of the external input, the neurons show synchronous regular firing, shown in fig. 2.5. The only exception occurs if the delay value gets close to the refractory period, i.e., $D \approx \tau_{\mathrm{refrac}}$, and spikes arrive at the same time the neurons are just able to be excited again. Therefore, some neurons are still refractory and the accumulation of neurons with identical spike times is disturbed and the stimulation is dominated by the Poisson generated external input. As a result of this, asynchronous behavior is found and synchronous behavior is only preserved for weak external inputs, e.g., for $g = 1$ and $\eta < 1.5$.

Additionally, since the synchronization strongly depends on the identical parameter settings of the neurons, it is lost if the fixed delay value of the neurons is replaced by a uniform distribution. As a consequence of this, asynchronous spiking is observed, depicted in fig. 2.5.

The behavior of the model changes significantly for $g > 4$, where the inhibitory stimulation exceeds the excitatory one. In general, the behavior is dominated by the strong inhibition and spikes are mainly driven by the external input. The expected firing patterns are depicted in panel C in fig. 2.6. Due to the permanent inhibition, the neurons spike irregularly. Moreover, small variations in the global firing rate are observed compared to the average firing activity of the network. Therefore, the neurons are expected to spike independent, hence asynchronous. In this regime, the firing rate can be approximated by

$$\nu_0 = \frac{\nu_{\mathrm{ext}} - \nu_{\mathrm{thr}}}{g\frac{C_{\mathrm{I}}}{C_{\mathrm{E}}} - 1}, \tag{2.24}$$

derived in Brunel 2000. However, for specific parameter settings, two additional firing regimes are observed. On the one hand, for strong inhibitory weights with $g > 6$ and strong external inputs with $\eta > 2.5$, a synchronous irregular firing with fast oscillating global behavior is found. There, the strong inhibitory weight leads to a reduced network activity when the internal spikes are received D time units after the network has been active. This, however, results in a lack of inhibition and due to the strong external input, groups of neurons will start to spike in close proximity in time, which will then again lead to reduced network activity after a time period of D. Consequently, the global network behavior oscillates with a frequency smaller than $1/(2D)$. Since this behavior is based on a fixed delay value for all neurons, it is not observed in networks with uniformly distributed delays.

On the other hand, for $g > 4$ and weak external inputs with $\eta \approx 1$, a synchronous irregular firing with slow oscillating global behavior is obtained. There, the strong inhibition leads to a mostly non-spiking network behavior. However, due to the missing inhibitory spikes in this state, the external Poisson input will excite some neurons to elicit a spike, which will then again inhibit all neurons in the network. As a result of this, the frequency of the oscillation depends on the strength of the external input.

All in all, depending on the chosen parametrization, the model exhibits distinguishable firing patterns. Although the synchronous states are mainly based on the idealized assumption of identical neurons, the measurement of its firing rate and the neurons' irregularity allows for the validation of replications of the model on wafer-scale neuromorphic hardware. Moreover, introducing basic concepts of connected inhibitory and excitatory LIF neurons, the model lays the foundation for models of more complex network structures found in the human brain.

2.5. The Cortical Microcircuit Model

The structure of the human brain, with approximately 86×10^9 neurons and 15×10^{13} synapses, forms an inherently complex system with hard to obtain network dynamics. Subdividing it into smaller modular building blocks with specific functions promises a simplification of the problem. Such modular organization is found in the cerebral cortex of mammal brains. There, neurons are organized in layers with cell-type specific connectivity, which form a repetitive cylindrical structure, the so-called cortical column [Mountcastle 1997]. Although extensively studied since its first discovery more than 50 years ago [Mountcastle 1957], its function remains poorly understood. However, over the years, a large amount of experimental data has been obtained describing its connectivity and activity. This facilitates the construction of connectivity maps, which model the cortical column's structure and behavior.

The cortical microcircuit model, introduced in Potjans et al. 2012 and subject to this thesis' investigation, is such a connectivity map. Its structure and network characteristics are introduced in this section.

2.5.1. Network Structure

The connectivity map of the cortical microcircuit model combines the connectivity data of various studies, e.g, based on anatomy [Binzegger et al. 2004], electrophysiology [Thomson et al. 2002], photostimulation [Dantzker et al. 2000] or electron microscopy [McGuire et al. 1984]. Thereby, it focuses on data obtained from the primary visual and somatosensory areas of rat brains and area 17 of cat brains. Resembling the structure under the surface of $1\,\mathrm{mm}^2$ of the cerebral cortex, it is subdivided into iour different layers, displayed in fig. 2.7.

In total, the network comprises approximately 8×10^4 neurons and 3×10^8 synapses. Since the incorporated studies provide only neuron and synapse counts and do not include individual pre- and postsynaptic neuron partners, i.e., the structure the network adopted to take over specific tasks, connectivity is treated statistically and is described by connection probabilities between populations. Inspired by the balanced random network model, introduced in section 2.4, each layer is subdivided into an excitatory and inhibitory population, which implement randomly drawn connections to all populations according to this predetermined connection probabilities. The number of neurons in each population is listed in table A.3 and the connection probability between populations in table A.4.

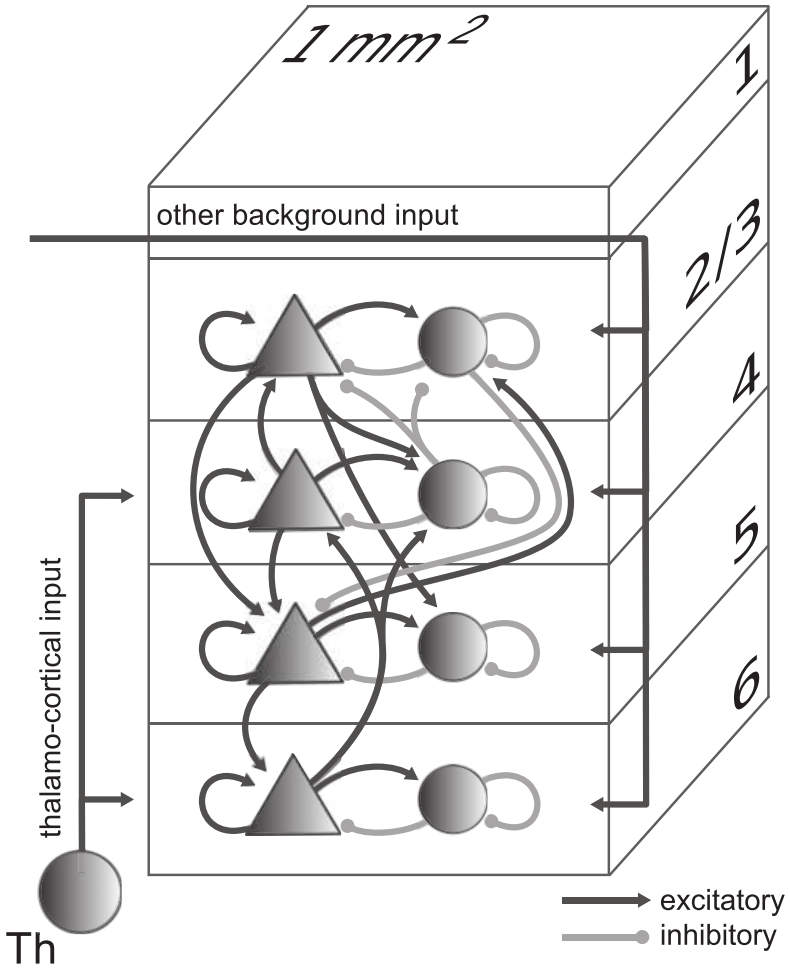


Figure 2.7.: Connectivity of the cortical microcircuit. The model replicates the structure under the surface of 1 mm^2 of the cortex of the brain. It is organized in four layers, L2/3, L4, L5 and L6, which host two populations, each. One of them, depicted by a triangle, acts excitatory to all its postsynaptic neurons, which is indicated by arrows with a triangular head. The other one, illustrated by a circle, exclusively forms inhibitory connections, shown by arrows with a circular head. Each population receives Poisson generated excitatory external stimulations. Additional inputs to L4 and L6 represent thalamo-cortical stimulations. Connections are drawn randomly with a predetermined probability between a pair of populations. Only connections with a connection probability > 0.04 are depicted. Taken from Potjans et al. 2012 by permission of Oxford University Press.

In addition to the internal connections, stimulation received from its surrounding are modeled by Poisson generated excitatory external inputs, which spike with a fixed rate of $\nu_{\text{bg}} = 8\,\text{Hz}$. The number of external pre-synaptic partners depends on the target neuron's population and is displayed in table A.3. Moreover, an optional time-dependent external input can be used to imitate thalamo-cortical stimulations. It is represented by 902 external neurons connected randomly with predefined probabilities to neurons in layer L4 and L6. During the first 10 ms of the experiment, these neurons send Poisson generated excitatory spikes with a firing rate of $\nu_{\text{th}} = 15\,\text{Hz}$.

Similar to the balanced random network, the cortical microcircuit is build of LIF neurons. However, a different parametrization (table A.5) is used to resemble biological neuron behavior. In particular, the synapses are current-based and use an exponential kernel characterised by the synaptic time constant τ_{syn}. In addition, to include heterogeneity, its weight and delay values are Gaussian distributed, with the mean inhibitory weight being four times larger than the excitatory one. Although in general layer-independent, the weight values of connections from the excitatory population of layer L4 to the excitatory population of L2/3 are doubled. Consequently, the cortical microcircuit model extends the balanced random network model into a multi-layered network with biologically inspired connectivity obtained from measurements on mammal brains.

2.5.2. Network Behavior

Implementing a biologically realistic network structure and neuron parameters, the cortical microcircuit model tries to imitate firing patterns found in the cortex. According to Amit et al. 1997, this corresponds to asynchronous irregular neuron behavior with low firing rates, which is specified by a mean firing rate of $\overline{\nu} < 30\,\text{Hz}$, an irregularity of $0.7 < \overline{\text{CV}} < 1.2$ and a synchrony smaller 8 [Potjans et al. 2012]. The balanced random network model demonstrates this behavior for a relative strength of the inhibitory synapses of $g > 4$ in combination with a sufficiently strong external input. Using a similar parametrization, the cortical microcircuit achieves asynchronous irregular firing in all populations.

A detailed overview of the model's firing behavior is depicted in fig. 2.8. In accordance with experimentally obtained data from awake animals, the firing activity of neurons is layer dependent. With focus on excitatory populations, layers L2/3 and L6 show the lowest firing rates with $\overline{\nu} < 1\,\text{Hz}$. While for L2/3 this is caused by the integration of primarily inhibitory inputs from all layers, L6 is dominated by inhibitory inputs from within its layer. Similar to L6, L4 is mainly characterized by connections within its layer but demonstrates elevated firing rates. The smallest layer, L5, receives inputs from all other layers containing the highest proportion of excitatory inputs. This results in the highest firing activity and also leads to the largest distribution of firing rates. In contrast, inhibitory populations generally receive a greater amount of excitatory input compared to the excitatory population within the same layer and therefore show higher firing rates.

Moreover, the observed variation of spike frequencies is caused by the random selection of pre- and postsynaptic connection partners in the model. For this reason, the number

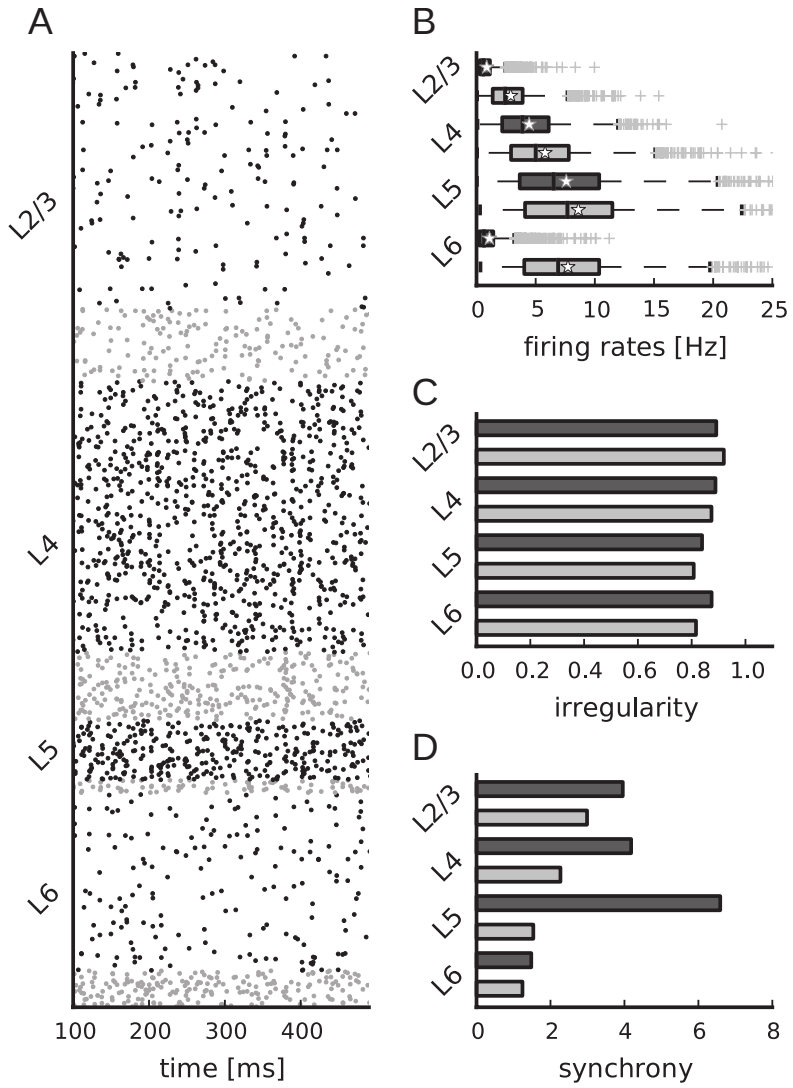


Figure 2.8.: Firing characteristic of the cortical microcircuit model. **A** shows the spike times of randomly chosen neurons for each layer. The number of presented neurons is chosen relative to the size of their population. Inhibitory neurons are depicted in gray and excitatory neurons in black. The remaining box plots display the firing rate (**B**), the irregularity (**C**) and the synchrony (**D**) for all populations. For this, 1000 spike trains per population are recorded for 60 s in **B** and **C** and 5 s in **D**. In **B**, stars mark the mean firing rates. A bin width of 3 ms is used for the synchrony calculation. Taken from Potjans et al. 2012 by permission of Oxford University Press.

of stimulations a neuron receives and its pre-synaptic partners differ within a population.

Representing a dynamic system, the stability of the model can be tested for modifications of the external input, and weight and delay values. Similar to measurements obtained from animals executing different tasks, small variations of the firing rates are observed. Nevertheless, the overall layer-dependent firing behavior of the model remains stable [Potjans et al. 2012]. This suggests that the model's behavior is encoded in its connectivity. Consequently, reproducing the cortical microcircuit's structure and comparing the obtained network dynamics serves as a benchmark for software simulators as well as for implementations on neuromorphic hardware.

3. The BrainScaleS-1 Neuromorphic Hardware System

Simulations are a widely recognized technique for gaining insights into systems that are otherwise difficult or even impossible to comprehend. This also applies to the human brain, with its inherently complex network structure. In recent years, with increasing compute power, the dimensions of simulated brain models have expanded, with the potential to reach the size of the entire human brain in the future [Gerstner et al. 2012]. Therefore, in general, traditional computers based on the von Neumann architecture are utilized. However, despite their flexible and easy to scale up technology, their different operational principle introduces a large computational overhead when simulating biological systems with numerous parallel operational components. Consequently, large compute clusters with high energy consumption are necessary to achieve reasonable simulation durations for large models.

In order to address these limitations, the development of specialized computational systems has been started. Optimized to model the structures of the human brain, these so-called neuromorphic hardware systems promise reduced power consumption at accelerated simulation speed [Mead 1989; Mead 1990].

The BrainScaleS-1 system, the first generation wafer-scale mixed-signal accelerated neuromorphic hardware platform developed in Heidelberg, is such a neuromorphic system [Schemmel et al. 2010; Schemmel et al. 2008]. Realized in 180 nm CMOS technology, it follows a physical modeling approach where neurons are built from analog circuits that communicate via digital spike transmission. Instead of solving differential equations that describe the dynamics of the neuron model, analog circuits implement the neurons' behavior. Consequently, in contrast to software simulations, the hardware is emulating the investigated network dynamic.

The system achieves a high neuron count by utilizing wafer-scale integration. There, the entire silicon wafer is used and not diced after fabrication. Therefore, a single wafer comprises 384 individual ASICs, so-called High Input Count Analog Neural Network (HICANN) chips with up to 196 608 analog neuron circuits and a total of 43 253 760 synapses.

Moreover, the high configurability of analog parameters allows for a variable acceleration speed between 1000 and 100 000 compared to biological real time. Assuming the typically used acceleration factor of 10 000, this means the emulation of 1 s of biological behavior takes 0.1 ms on hardware.

The system and the means to operate it are introduced in this chapter. It starts with the introduction of its basic building block, the HICANN chip, in section 3.1. Subsequently, in section 3.2, the wafer-scale integration and module assembly as well as the resulting

implications for experiments are discussed. Finally, since such a complex system cannot be operated without appropriate software, key features of the BrainScaleS-1 operating system necessary to execute hardware emulations are presented. Throughout the chapter, selected concepts that are used in the later parts of this thesis are presented with the necessary increased level of detail.

3.1. The HICANN Chip

The central building block of the BrainScaleS-1 system is the HICANN chip. A picture of it with sketched components is illustrated in fig. 3.1. Depicting several different aspects of the chip, it is used throughout the following sections to introduce major functional parts of the chip following the signal path of a spike.

3.1.1. Neuron Circuit

In the chip's center, 512 analog membrane circuits are located, each implementing the behavior of the adaptive exponential leaky integrate-and-fire neuron, an extension of the LIF neuron model, which allows for generating sophisticated neuron firing patterns [Brette et al. 2005; Millner et al. 2010]. Since the LIF neuron is a subset of this model, it is obtained by deactivating the adaptation mechanism, which represents the mode of operation employed throughout this thesis.

The circuits are based on the principles introduced in section 2.2 and their membranes are modeled by one of two available capacitors. Depending on the user's selection, either a small capacitor with approximately 0.16 pF or a big capacitor with approximately 2.16 pF is used. Moreover, the membrane circuits are organized in blocks of 64, which are split into a top and bottom row. Each of them is connected via a synaptic input circuit, introduced in section 3.1.5, to a synapse column consisting of 220 synapses that is, in accordance with its membrane circuit, either located in the top or the bottom synapse array. Since the synapse count of a single membrane circuit is in general insufficient for large-scale networks, membrane circuits of a single block can be interconnected to build larger neurons. Therefore, at the expense of available neurons, the number of stimulating synapses per neuron can be increased to a maximum of 14 080. A side effect of this is an increased membrane capacitance of the composite neuron as its individual capacitors are connected in parallel. However, since each membrane circuit implements its own leak conductance, the membrane time constant of the composite neuron is not affected by this.

Each membrane circuit is configured by a set of parameters, with the most relevant ones depicted in fig. 3.2. In addition to routing-specific configurations, these parameters are stored in single-poly floating gate cells, which provide voltages between 0 V and 1.8 V or currents between 0 µA and 2.5 µA according to their gate's accumulated charge [Millner 2012; Lande et al. 1996]. Subdivided into 4 blocks, in this thesis called FG blocks, the target value of each floating gate is set according to a configurable 10-bit value stored in one of two available SRAM cells per block. During programming, the cell's gate charge is modified incrementally in feedback loops until a satisfactory alignment with

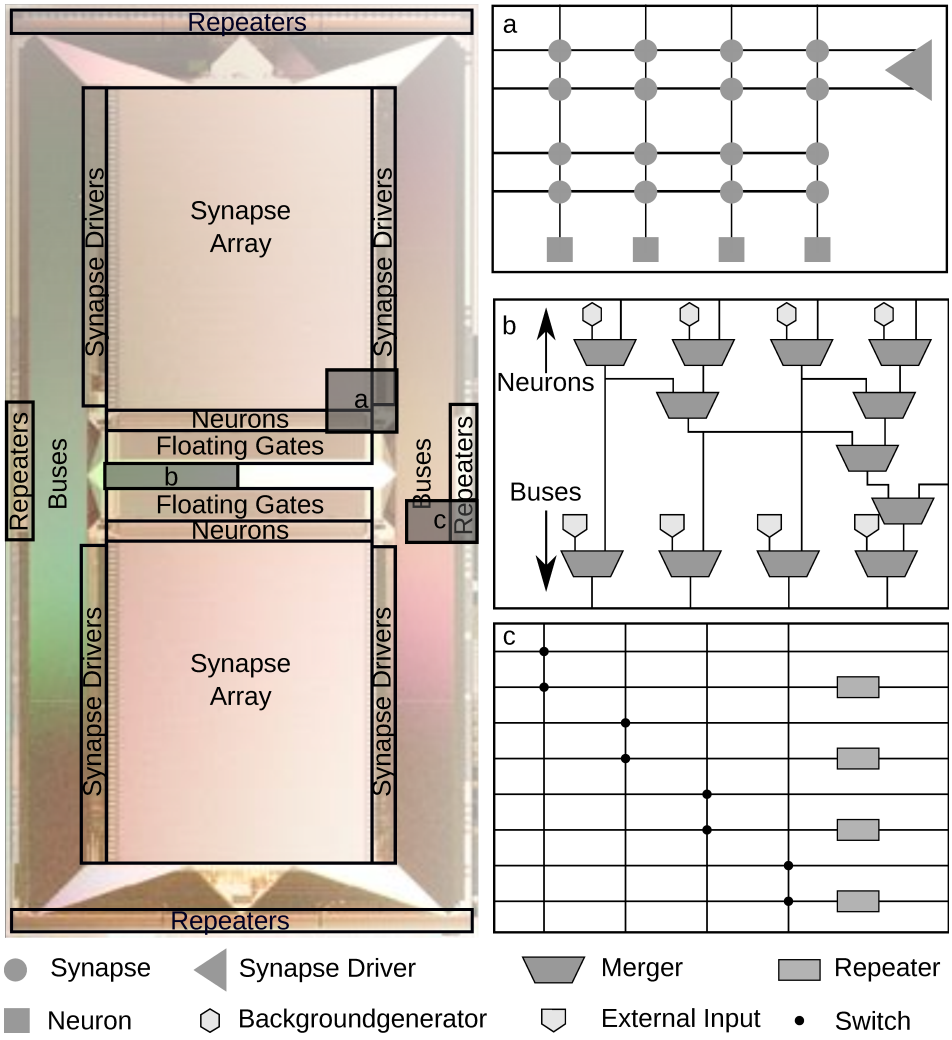


Figure 3.1.: Overview of the HICANN chip. On the left, a photograph of the chip is shown with labeled components and marked cutouts displayed on the right. **a** sketch of the lower right part of the synapse array, comprising a synapse driver connected to two synapse rows. Two additional synapse rows are connected to a synapse driver on the left half of the syanpse array. Neurons are represented by compoundable membrane circuits, each linked to a synapse column with 220 synapses. **b** left half of the merger tree. Spikes received at the top are optionally merged with spikes generated by background generators. The following layers allow for combining adjacent signals. The last stage enables injecting external spikes and reading out the received spikes to the host computer. Finally, all spikes are transmitted to the bus system of the chip. The right half of the tree is structured accordingly. **c** bus system of the chip. Horizontal and vertical buses, connected by configurable switches, distribute spikes across the wafer. On chip boundaries, repeater circuits regenerate the signals. Hosted alternating on two chips, at each edge they connect to a bus on the same and on the adjacent chip. Adapted from Schmidt et al. 2023.

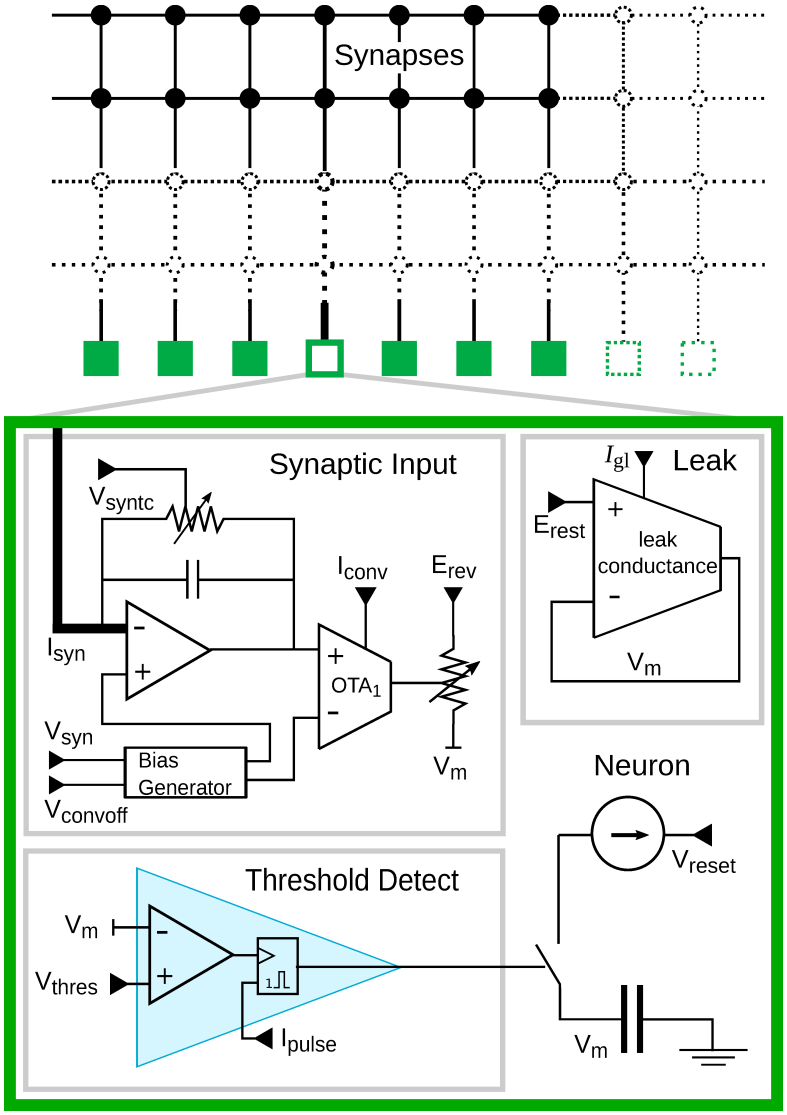


Figure 3.2.: Simplified schematic of the HICANN's neuron circuit and its synaptic input. If a digital spike is received, the synapse drivers and synapses of the connected synapse column generate an input current, denoted as I_{syn}, and transmit it to one of the two synaptic input circuits, depending on whether the input is excitatory or inhibitory. There, it is converted into an exponentially decaying signal with an adjustable synaptic time constant, configured by the parameter V_{syntc}. This signal controls the conductance towards the reversal potential E_{rev}. Furthermore, the signal's strength is scaled by the parameter I_{conv} and a bias generator. Configured by the parameters V_{syn} and V_{convoff}, it allows for mismatch corrections between the involved components. The neuron's membrane V_{m} is constantly connected via the leak conductance, set by the parameter I_{gl}, to the resting potential E_{rest}. If the membrane potential reaches the threshold value V_{thres} a spike is elicited and the membrane is connected for the refractory time, configured by I_{pulse}, to the reset potential V_{reset}. Adapted from Schmidt et al. 2023.

the target value is found. Although representing a low-power and low-space solution to store the 12 384 configuration parameters per HICANN, the incremental programming is time-consuming in comparison to digital parameter storage implemented in other parts of the chip and introduces write-cycle to write-cycle variability to the neurons' configuration [Kononov 2011].

3.1.2. Merger Tree

If the membrane potential of a neuron reaches its threshold, a digital spike is emitted. This spike signal exclusively consists of a 6-bit address, which is used to identify the target synapses of the neuron. It is injected into the merger tree, a multilayered structure of so-called mergers. These circuits are either configured to merge two distinct sources or to forward one of two input signals to a shared output line. Additionally, a FIFO buffer implemented for each neuron circuit permits the storage of a single spike if the merger tree is still blocked by a prior signal. In the event that the neuron spikes again before the buffer is emptied, the second spike will be discarded.

As depicted in fig. 3.1 b, all neurons in a single neuron block enter the merger tree at the same merger, located in the first layer of the merger tree. Consequently, on each HICANN there are 8 mergers, which allow for injecting additional spikes from so-called background generators. These circuits can be programmed to either generate regular or Poisson-distributed spike trains with a predefined frequency and spike address.

In the following layers of the merger tree, neighboring signals can either be merged until all are combined in the central merger or individual connections can be directly forwarded into the last merger stage. A duplication of spike signals is prevented, as in the routing logic, only one merger is allowed to use the output of a previous merger.

The last merger stage contains again 8 mergers to allow for a configuration where all layer-1 mergers directly forward their spike signals. On the one hand, these mergers are used to inject external spike signals with spike times programmed into the memory of the connected FPGA, as introduced in section 3.2. On the other hand, spikes received from the neurons are sent to the FPGA and, from there, read out by the host computer. For this, each external input merger can store up to 2 spike signals in a FIFO buffer to compensate for the reduced transmission speed of theoretically 25 MEvents/s from each HICANN to its FPGA [Klähn 2017].

Finally, each merger of the last stage is connected to one of eight specialized repeater circuits, the so-called sending repeaters, which inject the received signals into the layer-1 network, introduced in more detail in section 3.1.3.

In general, the digital circuits involved in signal transmission are operated with a configurable clock speed between 100 MHz and 250 MHz, which is provided by an internal PLL. However, as the sending repeaters require two clock cycles to process a signal, the speed of the merger tree is reduced to operate at every second clock cycle. Consequently, with a clock frequency of 125 MHz, which is used throughout this thesis, the smallest interval between consecutive spikes corresponds to 16 ns.

3.1.3. Layer-1 Routing

The layer-1 network routes on-wafer spike signals from the neurons to their target synapses. Possible injection points are 8 sending repeaters per HICANN receiving signals from the last stage of the merger tree. Depending on its configuration, each sending repeater connects to one horizontal bus on the same as well as on the neighboring HICANN to the left. These buses distribute the signal as they connect to buses on both neighboring HICANNs. In total, 64 buses are available per HICANN to distribute the signal horizontally across the wafer. Moreover, switches allow for connecting vertically aligned buses. Due to space constraints, these switches are implemented sparsely such that from each horizontal bus only 8 of the 256 available vertical buses can be reached. A cutout of the resulting switch matrix is visualized in fig. 3.1 c.

Once routed to its destination, a second set of horizontal buses can be connected to vertical buses to transmit the signal to the synapse drivers of the synapse array, introduced in section 3.1.5. Again, a sparse switch representation is implemented, enabling the connection of 24 out of 224 horizontal buses, organized in an alternating pattern. Depending on whether they are located on the left or right side of the chip, these buses also establish a direct connection to a synapse driver on the adjacent HICANN to the left or right, respectively. Furthermore, to reduce bus utilization, each synapse driver is able to inject its input into neighboring drivers, thereby implementing a chain of drivers that receive the same spike addresses.

As the signal strength decreases with the number of connected buses, it is regenerated between chip boundaries by repeater circuits. To this end, each HICANN hosts 64 repeaters for horizontal and 256 repeaters for vertical connections. Since each boundary requires only one repeater, their location alternates between the chip's left and right or top and bottom edges. Consequently, each repeater connects to one bus on the same and one on the neighboring HICANN. The previously introduced sending repeater is a specialized version of this circuit that, in addition to normal operation, allows for injecting signals from the merger tree. An in-depth discussion of the repeater circuit is provided in section 3.1.4.

In order to ensure signal integrity, during routing, the maximum number of connected buses is constrained. Therefore, a maximum of two closed switches are allowed before the signal has to be regenerated by a repeater. Additionally, a chain of synapse drivers is limited to 3 drivers. These restrictions are handled by the BrainScaleS-1 operating system, introduced in section 3.3. Providing experiment-specific routing results, at the beginning of an emulation, repeaters and switches are configured such that the layer-1 network statically connects utilized resources.

3.1.4. Layer-1 Repeater

The fundamental principle of a repeater circuit on the BrainScaleS-1 system is to deserialize a received spike signal from one bus and resend it serialized again on the other bus. Each signal consists of 8 bits. A 6-bit address, which identifies the target synapses, enclosed by a start and stop bit. Since, due to space constraints, no clock signal is

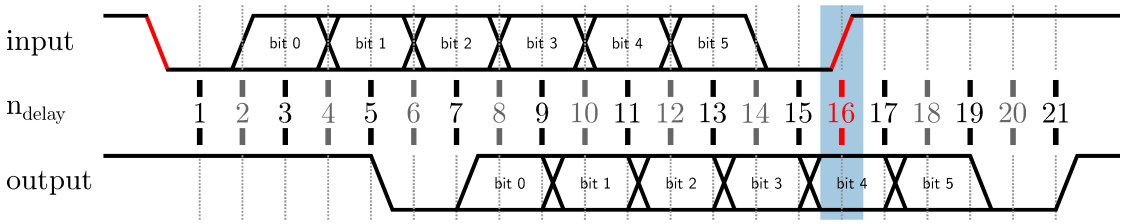


Figure 3.3.: Timing diagram of the repeater circuit. A spike signal arriving at the input of the repeater consists of a 6-bit address enclosed by a start and stop bit. There, the start bit is detected, and a DLL subdivides the time period between the falling edge of the start bit and the rising edge of the stop bit, marked in red, into 16 equidistant time steps. At every second time step, the address bit at the input is read and cached. After 3 address bits are received, the repeater starts to regenerate the cached data on the output. In addition, the DLL's timing is updated by detecting the rising edge of the stop bit. This edge detection is limited to the expected time frame of the stop bit, indicated in blue. Taken from Kaiser 2020.

provided to the repeaters, they have to extract the correct timing from the transmitted signals. To this end, at the beginning of each experiment, the background generators send spikes with address 0, which results in a low state of the bus for the duration of the signal transmission. By detecting the falling edge of the start bit and the rising edge of the stop bit, the duration of the signal is estimated. A DLL in the repeater circuit uses this timing information and divides it into 16 equally sized time bins. Once set to the correct timing, only the start and stop bit are exposed to the edge detection circuit, while the address bits are masked. Therefore, the DLL's timing can be adjusted according to all received signals to compensate for small changes caused by temperature or voltage fluctuations. This is further facilitated by adding one background generator to each connection, which sends regular spikes with address 0 at a frequency of 1 MHz to allow the DLLs to regularly refresh their timing during experiments.

During normal hardware operation, the repeaters detect the rising edge of the start bit and de-serialize and serialize the received signals according to the obtained time bins, visualized in fig. 3.3. Since every second time step occurs in the middle of a transmitted bit, the repeater captures and caches the values there. After it received 3 address bits, it regenerates the cached values on the second bus.

Repeaters are bidirectional and can be configured to either transmit in one direction or block the signal. Organized in 6 blocks, each group of repeaters is programmed by a custom on-chip SRAM controller. As a consequence of this, all repeaters in the same group share common settings, and their DLLs can only be resetted collectively. Moreover, each repeater, except the sending repeaters, implements a test data output, which allows for reading out its de-serialized addresses. A more comprehensive overview of the repeater circuits can be found in Hock 2009 and Schemmel et al. 2008.

3.1.5. Synapse Array and Synaptic Input Circuit

Each HICANN implements two synapse arrays, each hosting 110 synapse drivers connected to two synapse rows, respectively, as visualized in fig. 3.1. Implementing one synapse per connected membrane circuit, each synapse row contains 256 synapses. Consequently, each membrane circuit connects to 220 synapses, which are, according to the position of the membrane circuit, either located in the top or bottom synapse array.

Spike signals from the layer-1 network arrive at the synapse drivers, whose positions alternate between the left and right side of the array. There, the address of the received spike is de-serialized as each driver implements a reduced repeater circuit without test data output. This address determines which synapses in the two connected rows are enabled to stimulate their membrane circuit. To this end, each synapse implements a configurable digital SRAM cell, the so-called synapse decoder, which stores the address at which the synapse is enabled. Due to space constraints, this memory only represents the 4 least significant bits of the spike address. The two remaining bits get separated in the synapse driver and determine which of 4 possible strobe lines gets activated. Subdividing both synapse rows into blocks of 4 synapses, each of these lines implements a fixed connection to either the top left, top right, bottom left or bottom right synapses in these blocks. Consequently, a synapse is enabled if its strobe line is switched on and the lower four bits of the spike address match the value stored in its synapse decoder. In this case, the synapse is connected to either the excitatory, inhibitory, or both synaptic input circuits of its membrane circuit. Since this connection is selected row-wise in the synapse driver, all synapses in the same synapse row share the same synapse type.

The length of the input current pulse that gets added to the synaptic input line for each enabled synapse is controlled by the clock frequency of the HICANN provided by its PLL. For 125 MHz it is $t_{\text{syn}} = 8\,\text{ns}$. Its strength is provided by the synapse driver. As visualized in fig. 3.4, it is configurable to allow for different weight settings. The HICANN also incorporates two additional options for altering the synaptic weights, specifically for modeling weight modulations caused by the neurons' spiking behavior: short-term plasticity (STP) [Tsodyks et al. 1997; Abbott et al. 2004; Billaudelle 2014] and spike timing-dependent plasticity (STDP) [Song et al. 2000; Schemmel et al. 2006]. However, both features are not utilized in this thesis and are therefore deactivated and not taken into account in this consideration.

In the absence of the two mechanisms, the input current I_{syn} of a single synapse is designed to be

$$I_{\text{syn}} = V_{\text{gmax}} \cdot g_{\text{scale}} \cdot \frac{w}{g_{\text{div}}}. \tag{3.1}$$

There, the reference current V_{gmax} is provided by 16 floating gate cells distributed on 4 FG blocks. Every synapse driver selects one of 4 available values, as its placement on the HICANN, which is either top left, top right, bottom left, or bottom right, dictates its connection to a specific FG block. Subsequently, its reference current is scaled by two current mirrors. The first one has a fixed scaling factor of $g_{\text{scale}} = 0.4$. The factor of

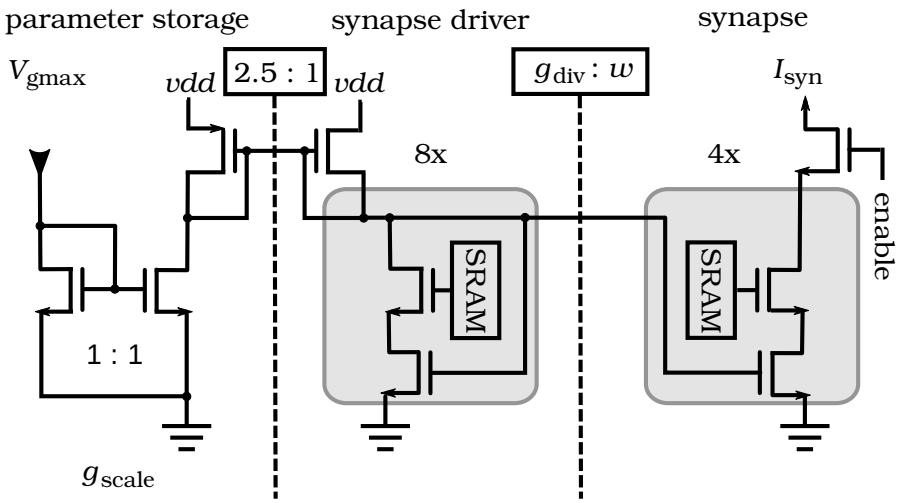


Figure 3.4.: Generation of the synaptic input current I_{syn}. The reference current V_{gmax}, stored in a floating gate cell, gets rescaled by two current mirrors. The first one realizes a fixed scaling factor of $g_{\mathrm{scale}} = 0.4$. The second factor depends on the ratio between the two parameters w/g_{div}. Both values are determined by transistors of different sizes, which are controlled by programmable SRAM cells. The configuration of g_{div} is stored per synapse row in the synapse driver and selects from 8 available transistors implementing scaling factors between 2 and 30. In contrast, w is stored per synapse and controls 4 transistors, realizing scaling factors ranging from 0 to 15. If an arriving spike activates the synapse, the enable switch is closed for 1 cycle of the HICANN clock. Adapted from Koke 2017.

the second one depends on the ratio between the parameters w and g_{div}, which are both realized by combining transistors of different sizes. The digital weight w is configured by a 4-bit value stored in an individual SRAM cell for each synapse. Due to space limitations, it is configured, along with the synapse decoder value, using a custom on-chip SRAM controller, which is implemented per synapse array [Friedmann 2013]. Each bit of the digital weight controls a different transistor representing the factors 1, 2, 4, and 8, therefore, in combination, realizing values ranging from 0 to 15. In contrast, the parameter g_{div} is stored in the synapse driver and configured per synapse row. Its 8-bit value controls transistors implementing the same scaling factors as used in the digital weight. However, each transistor is implemented twice. Consequently, scaling factors between 2 and 30 are realized.

Unfortunately, parasitic capacities residing in the synaptic circuit lead to a deviating behavior of the input current, as investigated in Koke 2017. Between consecutive pulses the synaptic circuit and thus also the parasitic capacities discharge. Hence, they have to be charged again during the next pulse. This results in a high peak current $I_{\mathrm{syn}}^{\mathrm{peak}}$ at the beginning of the pulse, used to charge the parasitic capacities, followed by a plateau of height $I_{\mathrm{syn}}^{\mathrm{mid}}$, which is approximately given by I_{syn}. The height of $I_{\mathrm{syn}}^{\mathrm{peak}}$ depends on the

total parasitic capacitance of the synaptic circuit and thus on the different transistors used to realize the synaptic weight. Consequently, it can be approximated by

$$I_{\text{syn}}^{\text{peak}} = i_0 + i_1 w_1 + i_2 w_2 + i_4 w_4 + i_8 w_8, \tag{3.2}$$

where $w_n \in \{0, 1\}$ is the state of the nth significant bit of the 4-bit weight value stored in the synapse, and i_n is the current required to charge the parasitic capacity of the corresponding transistor over the whole pulse length t_{syn}.

As a result of this charging effect, the current arriving at the denmem circuit is no longer rectangular but follows a more complex shape and is prolonged. However, since the pulse length of the synaptic input is 2-3 orders of magnitude smaller than the synaptic time constant, the actual shape of the current has no effect on the PSP trace of the neuron and thus allows for approximating it by a constant current

$$I_{\text{syn}} = I_{\text{syn}}^{\text{peak}} + I_{\text{syn}}^{\text{mid}} = V_{\text{gmax}} \cdot g_{\text{scale}} \cdot \frac{w}{g_{\text{div}}} + i_0 + i_1 w_1 + i_2 w_2 + i_4 w_4 + i_8 w_8 \tag{3.3}$$

of length t_{syn}, which preserves the total charge. Since the charging is independent of the duration of the pulse, the effect is more prominent for smaller timescales and thus for higher PLL values.

Fitting the model to simulation results reveals the need to extend the model. Finally, according to Koke 2017, the synaptic current can be best fitted to the simulation using

$$I_{\text{syn}} = V_{\text{gmax}} \cdot g_{\text{scale}} \cdot \frac{w}{g_{\text{div}}^{\gamma}} + \frac{\beta_1 w + \beta_2 w^2}{g_{\text{div}}} + i_0 + i_1 w_1 + i_2 w_2 + i_4 w_4 + i_8 w_8, \tag{3.4}$$

with the additional fit parameters γ, β_1 and β_2.

This input current is accumulated for all synapses in one synapse column and enters the synaptic input circuit, sketched in fig. 3.2. There, it is modified into a time-dependent signal, imitating an exponentially decaying kernel, c.f., section 2.2.2. Implementing conductance-based synapses, the generated signal modifies the conductance towards the reversal potential to stimulate the target neuron's membrane potential, thereby concluding the circle of a transmitted spike.

3.2. Wafer-Scale Integration and Module Assembly

In traditional semiconductor manufacturing, multiple copies of integrated circuits are fabricated on a single silicon wafer, which is then cut to obtain individual chips. This is exploited in the context of wafer-scale integration, where the entire wafer is employed as a single, uncut, large circuit. While sacrificing the flexibility to swap individual defective components, dense packaging is achieved. Moreover, the short on-chip interconnections lead to improved performance and power efficiency in contrast to multiple single-chip systems.

With all these features being desirable for large neuronal network emulators, wafer-scale integration is utilized in the BrainScaleS-1 neuromorphic hardware system to implement high neuron counts. This way, a 20 cm wafer accommodating 384 single HICANN chips

is obtained. Manufacturing-related, the wafer is subdivided into individual chips or groups of chips that implement no connections, as each of them is fabricated separately by applying the same structure with different alignment. On the BrainScaleS-1 wafer, such a group of chips comprises 8 HICANNs oriented in a rectangular shape, which is in this thesis referred to as HICANN-Group. Typically, connecting them is not required since the wafer is usually diced between chip boundaries anyway. However, for the BrainScaleS-1 system, utilizing wafer-scale integration, this does not hold. Therefore, the wafer is covered with a multilayered routing structure that interconnects all 384 HICANNs. Applied post wafer production, it is called post-processing layer.

In order to utilize a wafer, it is embedded into a module, which, together, form a BrainScaleS-1 system as depicted in fig. 3.5. This happens during the assembly process, which is described in detail in Schmidt et al. 2023. In the following, a short introduction to the most relevant parts of the system and their functions is given.

The system is supplied by a 48 V source, which connects to the main power supply. This, together with two auxiliary power supply boards, generates all intermediate voltages required by the system's components.

Central to the system is the main PCB, which combines all components. Correctly aligned and put under pressure, 384 elastomeric connectors, which are held in place by a positioning mask, establish a connection between the post-processing layer of the wafer and the main PCB. These connections are used to supply power and exchange data with the HICANNs. Thereby, individual parts of the wafer can be switched on and off, as the main PCB implements a separate power control for each HICANN-Group.

Moreover, two methods are implemented to communicate with the wafer. On the one hand, individual links to each HICANN, called high-speed communication, enable high bandwidth communication of 1 Gbit/s in both directions with each HICANN. On the other hand, one JTAG connection per HICANN-Group is daisy-chained through all of its 8 HICANNs. Although much slower, in the event of a failing high-speed connection, it provides an independent and reliable option to configure each HICANN into a desirable state for experiments.

Furthermore, 48 FPGAs, one per HICANN-Group, are connected via custom designed communication PCBs to the main PCB. Positioned next to the wafer, they are used for time-critical operations that cannot be performed using the larger overhead of the host connection. Therefore, each FPGA implements logic to configure, monitor and communicate with the HICANNs of their associated group.

Together with the large number of repetitive components of the HICANN, this modular design makes up the fault tolerance of the system, as undesired components can be deactivated and replaced by other parts. This capability is essential for operating a wafer-scale system.

Fully assembled, the system is placed into one rack of the BrainScaleS-1 machine room, shown in fig. 3.5. There, the analog breakout board of the system is connected to the analog readout module located in a drawer in the center of each rack. Equipped with twelve 12-bit ADCs, the module allows for the digitization and reading of analog membrane traces provided by the analog breakout boards. In general, a wafer provides 96 analog outputs, 2 per HICANN-Group, which are multiplexed between all its membrane

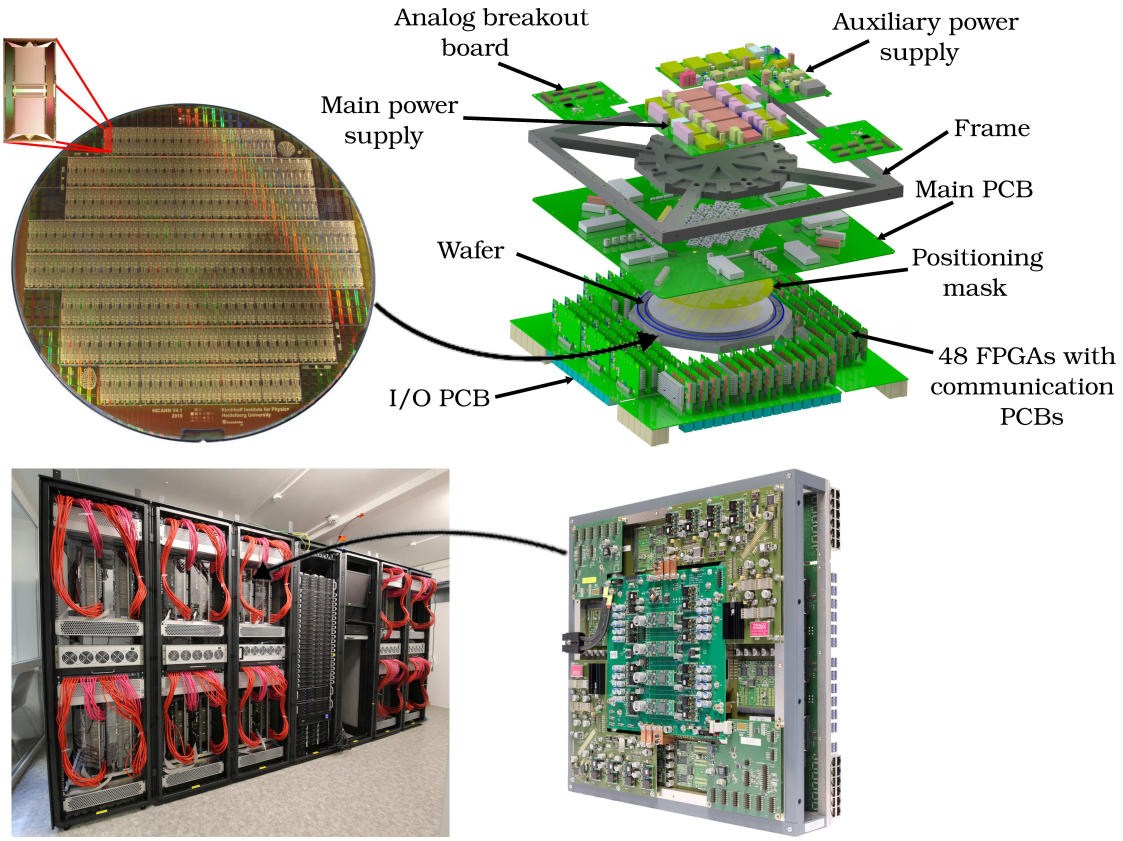


Figure 3.5.: Structure of the BrainScaleS-1 neuromorphic hardware system. In the top-left, a photograph of a wafer is shown that accommodates 384 HICANNs, interconnected by the post-processing layer, visible as a golden covering. Due to design constraints, additional circuits observable at the edge of the wafer are omitted in the final system. During system assembly, the wafer is built into a module, where it connects via elastomeric connectors to the main PCB. Different boards connected to the main PCB supply power and I/O possibilities to the wafer. Moreover, 48 FPGAs are used to orchestrate experiments. All components of a wafer module are visualized in the top right. Fully assembled, the module, shown in the lower right corner, is placed in the machine room depicted in the lower left corner. There, each rack accommodates up to 4 wafer modules, and in its center the analog readout module, which is connected to the analog breakout boards and allows for digitizing analog membrane traces. All modules are connected via Ethernet cables to the host computers positioned in the central rack.

circuits. However, the capabilities of the readout module used in this thesis are limited. Only 12 membrane traces of neurons, which have to be located in different wafer regions, can be recorded in parallel. Moreover, being distant from the system, the module adds noise to the signals during readout. Additional information about the structure and capabilities of the analog readout system can be found in HBP SP9 partners 2014.

The assembly is finalized by connecting a Raspberry Pi [Upton et al. 2017]. Communicating with individual components of the system via the I^2C [NXP Semiconductors 2012] protocol, it is used to start up the system, adapt voltage values, and readout monitoring data. Furthermore, it regulates the fans located in the rack, which keep the system at a constant temperature of approximately 50 °C, allowing for variations of ±5 °C.

In total, the machine room provides capacity for 20 BrainScaleS-1 systems. Designed to be interconnected with each other, it becomes possible to combine multiple systems and achieve even higher neuron and synapse counts. However, as the development of inter-module communication is still ongoing, this thesis focuses on conducting experiments on a single wafer module.

3.3. Software Implementation

Operating large-scale neuromorphic hardware is complex, as its many custom components add different constraints to the system. Therefore, to ensure proper experiment execution, the correct configuration of a substantial number of parameters is required. Moreover, starting from biological models, appropriate hardware representations have to be found. This also includes the translation of model parameters into hardware configurations, which often requires detailed knowledge of the hardware structure.

To accomplish this, the BrainScaleS-1 system deploys an extensive software framework that facilitates the emulation of experiments. This section introduces its most relevant features that are used throughout this thesis. Further details about the entire operating system are available in Müller et al. 2022.

The BrainScaleS-1 software is mainly written in the C++ programming language [ISO 2017]. However, with Python [Van Rossum et al. 2009] being very popular in the neuroscience community [Muller et al. 2015], automatically generated Python bindings are used to provide a Python interface. Allowing for direct access to a majority of C++ functions, network description and experiment control can entirely be done from within the Python programming language.

An experiment on the BrainScaleS-1 hardware typically unfolds as outlined below. The user describes the network under investigation and the software translates it into a valid hardware configuration. This configuration is written to the hardware and network execution is started. During the experiment, external spikes are injected into the network at predefined times stored in the FPGAs. In addition, spikes generated on the hardware are transmitted to the FPGAs and stored there. After a predefined execution time, spike results and analog traces are read back from the FPGAs and the analog readout module and made available for the user.

To accomplish this, the software is structured in different abstraction layers, as demon-

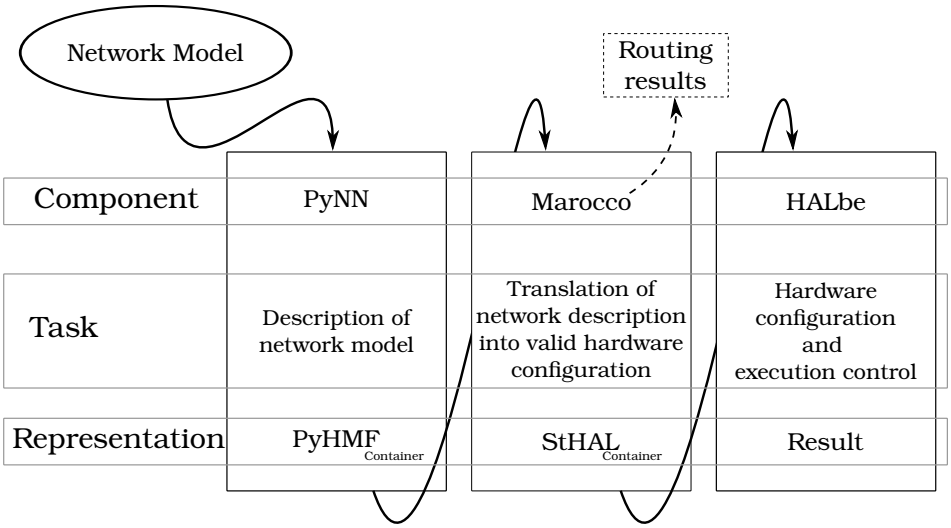


Figure 3.6.: Data-flow diagram of the BrainScaleS-1 software stack. The user describes the network model using a BrainScaleS-1-specific implementation of the PyNN API called PyHMF. Subsequently, the data structure, encapsulated in the PyHMF container, is translated by the map and route tool marocco into a valid hardware configuration. Stored in the StHAL container, it gets used by the next software layer, HALbe, to configure the hardware and execute the emulation. Finally, recorded membrane and spike results are transmitted back through all layers, allowing them to be accessed within the PyNN interface. Moreover, intermediate representations and the routing results generated by marocco can be stored on disk. Therefore, network translation and hardware execution can be separated and emulations can be rerun with identical or manually modified configurations. Adapted from Müller 2014.

strated in fig. 3.6. Its first layer forms the BrainScaleS-1-specific PyNN backend PyHMF[1]. Fully configurable in Python, it provides the user the PyNN API [Davison et al. 2009] to generate a description of the investigated biological network model. Since PyNN is simulator-independent and also provides other neural network simulators as backends [Rhodes et al. 2018; Eppler et al. 2008; Goodman et al. 2008], this enables the reuse of network descriptions.

Results are stored in a binary representation, which is handed over to the next software layer, the map and route routine marocco[2]. There, a network graph is generated, which matches the structure and all restrictions of the hardware. This procedure is mainly split into two steps.

It starts with the mapping, during which each neuron of the biological model gets assigned to physically available membrane circuits on the wafer. Since neurons on the HICANN are composed of multiple membrane circuits, cf. section 3.1.1, a configuration

[1] Available at `https://github.com/electronicvisions/pyhmf`
[2] Available at `https://github.com/electronicvisions/marocco`

parameter enables the user to select the desired neuron size. Moreover, manual placement requests can be applied to define target HICANNs or membrane circuits for groups of biological neurons. Networks in which neurons lack corresponding available membrane circuits are prohibited and result in the termination of the program.

In the second step, according to the network description, available connections on the bus system of the wafer are routed between the previously placed neurons. To this end, a hardware graph is generated, which encompasses all hardware-specific constraints and allows for handling undesired components by excluding corresponding vertices and edges. Subsequently, either the backbone algorithm [Fieres et al. 2008] or the Dijkstra algorithm [Dijkstra 1959] is utilized to find routes on the hardware between the pre-synaptic and postsynaptic neuron circuits. Due to the limited amount of available resources on the hardware, it is possible that not all connections required by the biological model can be established on the hardware. The percentage of not implemented routes is called synapse loss and represents a performance measure for the map and route results. Due to the separation of mapping and routing as well as the limited capabilities of the routing algorithms, the results found do not necessarily represent the optimal solution. Often, manual placement requests and the selection of different neuron sizes can be applied to assist the algorithms in further reducing the synapse loss. An in depth description of the map and route software is presented in Jeltsch 2014.

Once the network is mapped to the hardware, the map and route layer is also responsible for translating biological neuron parameters into corresponding circuit configurations. This is necessary since the voltages and currents used on the hardware differ from those found in biology, cf. section 3.1.1 and section 2.1. Moreover, utilizing analog components, the BrainScaleS-1 hardware is affected by manufacturing induced circuit mismatches. Therefore, during parameter translation, the algorithm utilizes a circuit-specific calibration to minimize resulting variations.

This calibration is generated using the standalone Python framework cake[3]. In line with the map and route software, it utilizes the lower software layers to execute measurements with specialized configurations. Evaluating the response of the neuron's membrane potential to different settings of individual configuration parameters, depicted in fig. 3.2, a translation of the corresponding neuron property is obtained for the whole parameter range. During the measurements, all other configuration parameters are set to appropriate but fixed values. Therefore, possible effects of configuration parameters on unrelated neuron properties are not considered. In addition, the precision of the calibration is limited by the write-cycle to write-cycle variability of the floating gates. Individual steps of the calibration routines and their performance are discussed in detail in Koke 2017; Kleider 2017; Schmidt 2014; Schmidt et al. 2023.

The results of the calibration are represented in a custom database[4]. Different states of it can be stored on disk in either XML or binary format and later loaded during routine parameter translation, therefore facilitating the utilization of experiment-specific calibration data.

[3] Available at `https://github.com/electronicvisions/cake`
[4] Available at `https://github.com/electronicvisions/calibtic`

Finally, the map and route layer generates a binary representation of the desired hardware state for the experiment, consisting of target values for all floating gates, switch and repeater configurations, as well as synapse driver and synapse settings. Moreover, the graph representation of the network on the hardware is available as a routing result. Storing both containers on disk enables the evaluation of the resulting network topology. Furthermore, since configuration generation and hardware execution are separable, previously obtained results can be loaded and executed on the hardware. This allows for faster execution of experiments with the same or slightly adapted settings. However, still under development, files are stored in human-readable XML format, which is not optimized for performance.

The final layer of the BrainScaleS-1 software stack forms HALbe[5]. It implements the communication with the FPGAs and is used to configure the wafer according to the map and route results.

Upon completion of the experiment, the recorded spikes and analog traces are transmitted back through all the software layers. In this process, they are translated from the hardware domain into the biological domain and finally made available in PyNN.

In addition, independent of the experiment workflow, continuous system monitoring ensures the correct operation of the system. This is accomplished through the attached Raspberry Pi, which continuously reads more than 1800 metrics per system, including data such as voltage, current or temperature [Schmidt et al. 2023]. Visualized in Grafana dashboards [Labs 2018] this allows for tracking system changes over time or during hardware operation.

[5] Available at `https://github.com/electronicvisions/halbe`

4. Commissioning of the BrainScaleS-1 System Towards Large-Scale Experiments

This thesis aims to replicate large-scale biological networks on BrainScaleS-1, a wafer-scale neuromorphic hardware system. In this context, the system's physical modeling approach and the utilization of wafer-scale integration introduce intricate challenges in realizing emulations. These challenges arise from the diminished flexibility and reliability of components. Furthermore, in contrast to the successful implementation of smaller networks [Göltz et al. 2021; Kungl et al. 2019; Schmitt et al. 2017], conducting large-scale experiments demands an even higher level of control over the utilized system. Manual adjustments and fine-tuning of individual parameters are no longer feasible once several thousand components have to be adjusted. Therefore, the establishment of automatic tests and a controlled workflow is imperative.

At the initial state of this thesis, several BrainScaleS-1 systems were fully assembled or were close to completion. Although extensively tested during assembly to achieve a maximum number of working components, cf. Schmidt et al. 2023, the state of individual components was unknown, and accordingly, large-scale emulations were not possible due to the utilization of malfunctioning components. A calibration framework was in place to translate neuron parameters from the biological regime to hardware parameters while minimizing the fixed pattern noise of the analog components [Koke 2017; Kleider 2017; Schmidt 2014; Schmidt et al. 2023]. To find a physical representation of the biological network described in the PyNN framework, the map and route routine was implemented [Jeltsch 2014; Müller et al. 2022].

During this thesis, the commissioning of the BrainScaleS-1 system was extended to allow for large-scale experiments. An extensive test framework was established, which, in combination with the availability management and fault tolerance of the system, allows the user to handle the system as an idealized substrate without malfunctioning components. This framework is presented in section 4.1. In addition, missing calibration routines necessary to parametrize biologically plausible large-scale networks such as the cortical microcircuit or the balanced random network were implemented, shown in section 4.2. Furthermore, the map and route routine was improved, as discussed in section 4.3, to cope with the requirements of larger experiments by improving the utilization of the available hardware resources. Finally, section 4.4 introduces undesired characteristics identified during hardware operation, along with the developed solutions designed to minimize their influence on experiments.

If not mentioned otherwise, in this chapter, all parameters are given in the biological regime.

4.1. Availability Management

Utilizing wafer-scale integration to interconnect individual chips, the BrainScaleS-1 system achieves high energy efficiency and high-bandwidth inter-chip communication [Zoschke et al. 2017]. The downside of this approach is the reduced flexibility in handling malfunctioning components, since individual problematic chips cannot be replaced. Therefore, malfunctioning components are inevitable on a wafer hosting 384 individual HICANNs. While these components can be manually avoided in small-scale experiments this is no longer feasible using large parts of the system, which is the focus of this thesis. Not handled correctly, these components either disturb experiments or prohibit the execution of them in the first place. For this reason, a resource management was developed that is subject to this section, which is used to find, store and handle malfunctioning components. A custom availability database stores components that should not be used. Several steps, comprising communication tests, digital memory tests, an exclusion of dependent components, and a calibration-based exclusion are filling this database. There, the execution sequence is important as each step uses the resulting database state of the previous step. Additionally, storing different states of the database on disk allows for distinguishing the origin of entries, as well as the utilization of dedicated components for each operation.

4.1.1. Availability Database

The basic principle of the availability database of the BrainScaleS-1 system is to store components that should not be used during hardware operation. While the fundamental implementation of the database was already present [Jeltsch 2014] it was largely extended during this thesis and full support for it was added to the map and route algorithm. The database is written in C++ and only stores excluded components without further information. Different states of the database can be stored on disk in either XML-based or binary format using boost serialization. This makes it possible to generate experiment-specific availability data. In addition, as the hardware is still under development and system components might be replaced or modified, it allows for flexible adjustments of the stored data and for tracking the state of the system over time.

The database is stored in separate files for different involved hardware components, reducing the amount of data that has to be updated if components are changed and facilitating parallel executions. There, a wafer file contains the communication details of the corresponding wafer. One file per FPGA holds the communication possibilities of its corresponding FPGA, and one file per HICANN the information of each chips' malfunctioning components. Using the coordinate system of the BrainScaleS-1 operating system, introduced in Müller et al. 2022, this HICANN file is subdivided into different abstraction layers of the chip. For example, in case of problems with individual synapse configuration registers, the corresponding components can be excluded individually, whereas for a malfunctioning control flow, the affected synapse array can be removed as a whole, cf. section 4.1.3. This constitutes a lightweight solution to store the data as a single entry may encompass all unavailable components. Loaded during every hardware

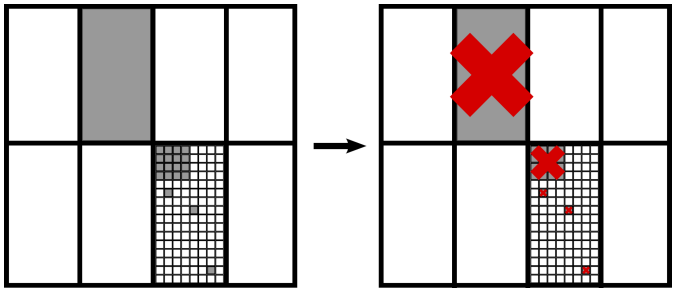


Figure 4.1.: Operation of the availability database. The chips of one HICANN-Group are depicted by eight large rectangles, left before and right after the exclusion of malfunctioning components. Using the hierarchical structure of the system, in the database, each HICANN is subdivided from an individual-component level up to larger functional units, visualized by different-sized rectangles within one of the HICANNs. For demonstration purposes, some components, highlighted in gray, are chosen to be malfunctioning. Detected in the availability tests, the appropriate unit dependent on these components is excluded from the database, visualized by red crosses. Figure taken from Müller et al. 2022.

execution and fully integrated into the map and route algorithm, unwanted components are removed from the hardware graph and thus are not utilized. Therefore, the wafer can be treated as an idealized substrate without malfunctioning components from the user's perspective. The operation of the database is demonstrated in fig. 4.1.

Since excluded components are ignored in experiments, another application of the database is to manually adjust the automatic routing process. A command-line tool was developed to facilitate the generation of experiment-specific databases. It uses the syntax

```
$ redman_cli.py <PATH> <FILE> <OPERATION> <NAME> <NUMERATIONS>
```

where the database file that should be adapted is stored in the directory specified by `PATH`. `FILE` represents the short format of the target file, where the wafer id prefixed with "W" and if required, the HICANN or FPGA id, prefixed with an "H" or "F", respectively, can be handed over to the program. The user chooses from three `OPERATION`s: "enable", "disable" and "has". While the first two options remove or add components from the database, respectively, the last one checks if a specific component is available. Finally, the `NAME` and the `NUMERATIONS` of the target coordinates are chosen.

```
$ redman_cli.py . W30H100 disable synapses 0 1
```

for example removes the first and second synapse of HICANN 100 on wafer 30 from the availability database and the results are stored in the current directory ".".

Table 4.1.: Communication test results at the time of writing for five fully assembled wafer modules, named by their position in the racks. The number of HICANNs failing the initialization via JTAG communication are shown in the first row and the number of HICANNs that fail the initialization via high-speed communication are shown in the second row. HICANNs that have no high-speed connection by design or that have already failed the JTAG communication test are excluded from the high-speed test results.

Resource	Module 23	Module 24	Module 30	Module 33	Module 37
JTAG	51	11	2	39	73
High-speed	1	1	4	0	0

4.1.2. Communication Test

After assembly, a series of tests, developed in the course of this thesis, have to be executed on the hardware to generate the availability data that is essential for large-scale experiments. These tests build upon each other and start with the communication test. It is separated from the digital memory test since during normal hardware operation all HICANNs controlled by the same FPGA have to be initialized correctly. Therefore, the communication test is executed in advance to find the appropriate communication method for each HICANN. During the communication test, only the HICANN under test is initialized. Failing initialization of one HICANN leads to an unexpected termination of the program and might result in an unresponsive state of the corresponding FPGA. To circumvent this, before each communication test, the FPGA is reprogrammed and thus a reliable state is ensured.

There are two possibilities for communicating with the HICANNs. The first one is given by the JTAG ports of the chips that are daisy-chained within one HICANN-Group consisting of eight HICANNs. By design, all control registers can be reset using this JTAG communication. However, chaining through all HICANNs of one group, the connection offers not enough bandwidth during experiments. Therefore, a faster high-speed serial link connects each HICANN individually to the communication board. Since the link initialization requires an existing JTAG connection, HICANNs without JTAG communication also have no high-speed communication. During the test, the chip is initialized using both communication methods. Failures indicate malfunctioning behavior and the used communication method of the corresponding chip is marked in the availability database.

In table 4.1 the number of HICANNs failing the communication test are shown for five fully assembled wafer modules. Using the test results and the fault tolerance of the system, all wafer modules can be used for experiments. However, to maximize the number of usable components, the large-scale experiments discussed in this thesis are mainly emulated on wafer module 30, which performs best in the test. Therefore, the methods in this section are demonstrated by means of this wafer module but are acquired for all other wafer modules accordingly.

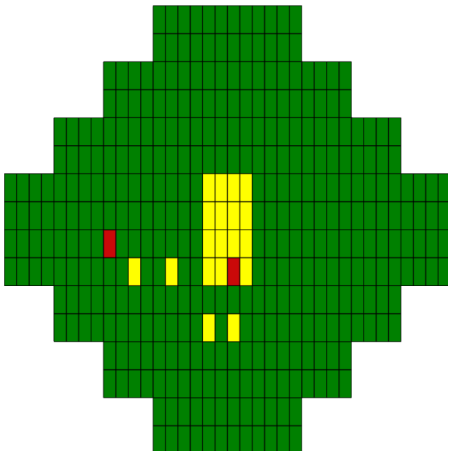


Figure 4.2.: Overview of the commissioning test results for wafer module 30. Each rectangle represents the position of one HICANN on the wafer and its color indicates the test result. Red HICANNs cannot be initialized, neither via JTAG nor high-speed communication. Yellow HICANNs only fail initialization via high-speed communication. Green HICANNs can be initialized using both methods. Sixteen HICANNs in the center of the wafer are without high-speed communication by design.

The location of the initialization problems found on wafer module 30 can be seen in fig. 4.2. Two HICANNs cannot be initialized via JTAG communication and therefore also have no high-speed communication. Since the JTAG connection is implemented as a chain through all HICANNs of one group, single-chip failures most likely indicate that the problem is not caused by the JTAG connection itself. One reason for the failing initialization could be a underpowered supply of these chips caused by an insufficient connection to the main PCB. As the correct behavior of circuits with undervoltage is not guaranteed, this could also explain an unstable circuit behavior observed in the long-term stability measurements shown in fig. 4.3. There, dependent on the current state of the system, HICANN at position 200 occasionally fails the JTAG communication and HICANN 304 the high-speed communication test.

This unstable behavior presents a problem as all hardware executions depend on a stable connection to the HICANNs. However, successful communication with problematic connections is only achieved during the limited interaction with the chip required for the communication test. Consequently, the memory test, outlined in the subsequent section, enables the detection of unstable HICANNs, allowing for their manual exclusion from the final test results.

Executed on the systems once they are fully assembled and integrated into the rack, these tests aim to guarantee their correct operation despite having malfunctioning components. Therefore, at this point, distinctions are not made regarding whether problems are caused by the communication method, malfunctioning auxiliary boards, insufficient power supply, or issues on the chips themselves With these test results

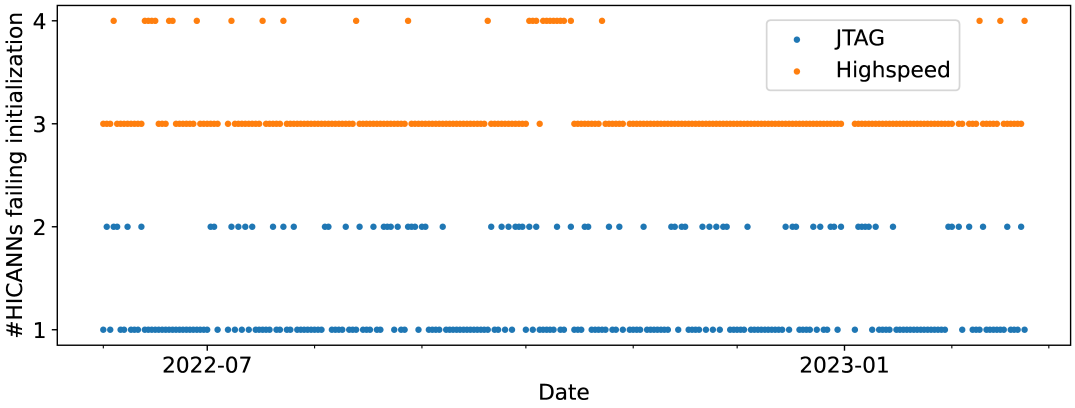


Figure 4.3.: Long-term stability of the nightly executed communication tests of wafer 30. The number of HICANNs that cannot be initialized via JTAG or high-speed communication is shown. HICANNs that have no high-speed connection by design or that already failed the JTAG communication test are excluded from the high-speed test results. One HICANN for each of the two communication types fails the test occasionally. All other HICANNs show stable results in all tests.

available, chips failing initialization are excluded in all following steps and thus do not affect hardware executions.

4.1.3. Digital Memory Test

Manufacture-induced malfunctioning components are inevitable on wafer-scale microelectronics. Without careful management, especially in large-scale experiments with complex and hard-to-track network dynamics, the unknown behavior of components could compromise the reliability of results. Therefore, a digital memory test was developed to identify malfunctioning digital configuration registers on each HICANN. On the one hand, using the test results, malfunctioning components can be excluded and therefore do not disturb experiments. On the other hand, the results are used to characterize the wafer modules and allow for the evaluation of their digital building blocks.

The digital memory test utilizes the communication test results and the whole HICANN-Group of the HICANN under test is initialized using the available communication methods, respectively. HICANNs that cannot be initialized are removed from the test and therefore do not further disturb the execution. If the HICANN under test is not available in the used availability database, the test is skipped. As explained in the previous section, HICANNs with communication-induced initialization problems may fail the digital memory test if these problems go undetected beforehand. In this case they are identified by their missing test results and the respective communication method for this HICANN has

to be manually excluded from the database. Subsequently, the memory test has to be repeated for all the other HICANNs of the related HICANN-Group using the adapted availability database.

During the test, all digital configuration registers on the HICANN are repeatedly write/read-tested with random but valid configuration values. Instead of testing the whole parameter space, random values are chosen to cover different parameter combinations without exceeding reasonable runtimes. Thereby, 10 repetitions were found sufficient to characterize a chip. If a write/read mismatch is detected, functional units that cannot be used without the register under test have to be marked in the database. Here, it is enough to exclusively exclude the highest-level affected functional unit in the system's hierarchy. This ensures a minimal but sufficient exclusion of components. Moreover, since only a single entry has to be stored in the availability database, it makes up its sparsity.

For example, if a register in the decoder of an individual synapse fails the test, the whole synapse driver is marked in the database since this is the only possibility to ensure that the malfunctioning decoder is not activating its related synapse when receiving a spike. As a result, all 512 synapses connected to this synapse driver are not used in experiments.

In total, the test checks more than 42 MiB of configuration registers on a wafer. For wafer 30 the size of the tested registers per resource and the number of excluded components are depicted in table 4.2. Additionally, the distribution of malfunctioning components on individual HICANNs is shown in table 4.3.

It can be seen that only on 19 out of 384 HICANNs components failing the test are observed. There, most malfunctioning behavior in synapse related registers can be traced back to problems in the configuration registers of individual synapses, which dominate the test with 110 KiB per HICANN. For example, only on HICANN 373 synapse driver related registers show malfunctioning behavior, while all other excluded drivers can be attributed to issues in individual synapse decoders.

Moreover, synapse-related problems have been observed to occur concentrated on single synapse arrays. However, manufacturing-induced errors in digital registers are expected to occur rarely and to be isolated, as observed on HICANN 360. Therefore, the accumulation of malfunctioning behavior most likely indicates a problem in the digital control chain of individual HICANNs. Consequently, the reliability of write/read operations during the test is not ensured on these HICANNs. To address this issue, tailored handling procedures are established depending on the type of components.

For non-synapse-related components, the utilized approach is described in section 4.1.4. However, synapse-related registers are treated differently. On the one hand, they face a higher likelihood of manufacturing-induced errors due to the large number of registers. On the other hand, they are programmed per synapse array by a custom on-chip SRAM controller, described in Friedmann 2013, introducing an additional source of errors. Therefore, special attention was given to the evaluation of synapse behavior, and an additional memory test was implemented as detailed in the following.

On HICANNs 204 and 373 malfunctioning SRAM controller registers are found. As a consequence of this, the correct programming of the synapses on the corresponding

Table 4.2.: Details of the digital memory test and the synapse stability test for wafer module 30 at the time of writing. The size in bits of the tested configuration registers per HICANN is given. Since no register is exclusively assigned to synapse rows, they are marked with a hyphen ("-"). However, they are still listed as they form functional units that are excluded if registers they depend on fail the test. In addition, the number of components that are tested on the whole wafer is shown. These numbers are smaller than the total number of components on a wafer, since components on HICANNs without communication possibilities are excluded as they cannot be tested. In the final column, the ratio of excluded components to the total number of tested components is presented. There, synapse-related components can be excluded multiple times. For example, an excluded synapse could be part of an excluded synapse row or array and could be connected to an excluded synapse driver.

Resource	Register size (Bit)	#Components	Excluded
Synapse arrays	560	726	0.96%
Synapse drivers	10 560	79 860	1.24%
Synapse rows	-	162 624	0.07%
Synapses	901 120	40 888 320	0.3 %
FG blocks	184	1528	0.2 %
Analog outputs	22	726	0.0 %
Background-generators	192	2904	0.0 %
Mergers	77	8349	0.0 %
Switches	7680	2 933 760	0.14%
Repeaters	2560	122 240	0.04%

Table 4.3.: Distribution of excluded components on wafer module 30 at the time of writing. All HICANNs that contain components that failed the digital memory test or synapse stability test are shown. In the first column, the positions of the relevant HICANNs are listed. HICANNs are enumerated per wafer, row-wise, starting at the top left. The remaining columns display the number of components per resource failing the tests on the respective HICANN. For each resource, the total number of available components on a single HICANN is shown in parentheses. Due to observed problems in the digital control chain on a small number of HICANNs, the number of excluded synapses and repeaters show varying test results.

HICANN	Synapse				FG blocks	Switches	Repeaters
	Arrays (2)	Drivers (220)	Rows (440)	Individual (112640)	(4)	(7680)	(320)
23	1	110	0	948	0	0	0
36	0	110	0	32 885	0	0	0
86	0	0	0	0	0	170	0
109	0	110	0	220	0	0	0
121	1	1	0	2	0	0	0
132	0	110	0	56 320	0	0	0
152	1	1	0	8	0	0	0
154	0	0	0	0	0	0	2
181	0	110	0	475	0	0	0
189	0	0	0	0	0	0	16
190	0	0	0	0	0	0	35
204	1	110	0	7035	0	1530	0
272	1	110	0	660	0	0	0
275	1	110	0	8363	0	0	0
287	0	0	0	0	1	0	0
304	0	0	0	0	0	225	0
336	0	0	0	0	2	284	0
360	0	0	0	1	0	0	0
373	1	110	110	14 077	0	1762	0

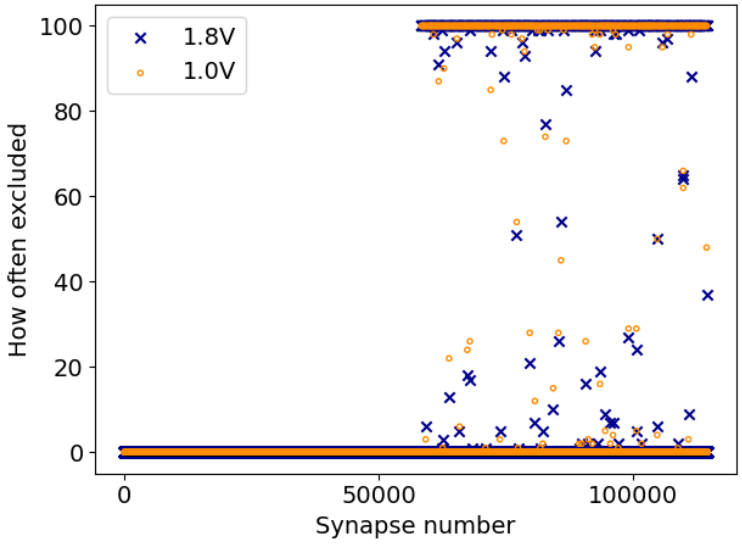


Figure 4.4.: Stability test results for different values of the supply voltage VDDBUS for HICANN 272 that shows unstable behavior and hosts most malfunctioning synapses. For each synapse of the HICANN it is shown how often it is excluded in 100 write/read repetitions of the memory test. A value of 0 or 100 demonstrates that the synapse is stable and fails either none or all of the tests, respectively. Values in between indicate unstable behavior in the test. Synapses with a synapse number above 56 320 are located on the second synapse array. The test results differ depending on the array the synapses are located on.

synapse array is not possible and the whole array is excluded. On the remaining HICANNs, five synapse arrays show varying test results. This means, in consecutive executions of the memory test different synapse registers fail the test, as shown in fig. 4.4. There, no difference is observed if the registers are written once and read repeatedly or write/read tested in each iteration (cf. fig. A.2).

To rule out potential issues with the power supply, all measurements are conducted at two distinct voltage levels of the controller's supply voltage (VDDBUS). These levels are 1 V and 1.8 V. Although no improvement is observed, the default supply voltage used for experiments is set to the maximum value of 1.8 V to minimize potential further implications. Nevertheless, the power supply of individual HICANNs could still suffer from an insufficient connection to the main PCB. Dependent on the current state of the system, which is subject to vibrations and temperature fluctuations in the order of 5 °C, this could lead to malfunctioning behavior in individual components farthest away from their operation point, possibly explaining the unstable programming observed in some controllers.

Unstable behavior during the test is a problem, as malfunctioning synapses may remain

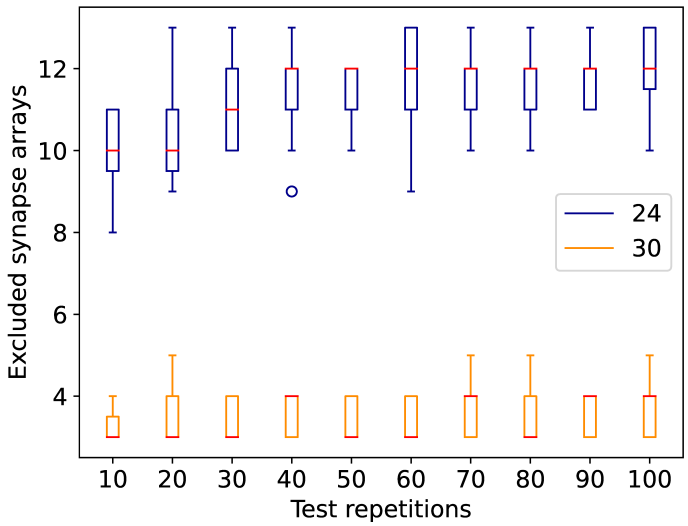


Figure 4.5.: Synapse array stability results in dependence on the number of test repetitions for wafer 30 in comparison to wafer 24. Wafer 24 shows up to 13 unstable synapse arrays, wafer 30 up to 5. On wafer 24, more than 10 repetitions of the test turn out to be beneficial to reliably detect more unstable synapses. In contrast, on wafer 30, 3 arrays host only a few unstable synapses and therefore fail the test rarely. There, taking the increased execution time into account, more than ten test repetitions are not practical as they show no reasonable improvement. Furthermore, undetected synapse arrays can be excluded through long-term measurements, as discussed in section 4.1.3.

undetected. To address this, a stability test was developed. During this test, all synapses of each synapse array are write/read tested repeatedly with a fixed value. If at least one synapse shows different results in one of the tests, the whole array is assumed to be unstable and is excluded. The proportion of affected synapse arrays on wafer 30 is shown in table 4.2 and the number of excluded arrays per HICANN in table 4.3.

Results of the stability test on wafer 24 and wafer 30 for different repetition counts are visualized in fig. 4.5. In contrast to wafer 24, on wafer 30 instability is only observed on a small subset of synapses on affected arrays. Therefore, unstable behavior rarely occurs and might remain undetected even after many repetitions of the test. To account for this, further exclusion can be made, as discussed in the following paragraph.

During long-term measurements on wafer 30 spanning from June 2022 to March 2023, a consistent observation was made (cf. fig. A.4). Across 216 measurements, each consisting of ten repetitions of the stability test, the presence of the same five HICANNs with one unstable synapse array was identified. There, HICANNs 272 and 23 are always detected. HICANN 275 fails 74 % of the tests, HICANN 152 58 %, and HICANN 121 4 %. Consequenly, HICANNs with fewer individual synapse problems show less unstable

behavior and are therefore harder to detect. This shows that the test only fails to reliably detect problems on arrays having a small number of unstable synapses. However, since only a small percentage of synapse arrays show unstable behavior, depending on the requirements of the experiment, all arrays that failed the test once or show more than isolated failing synapses can be excluded from the final database. Thereby, at the expense of losing stable programmable synapses, it is ensured that no unstable behavior is observed in experiments.

4.1.4. Effective Exclusion of Components

In the last section, memory tests were introduced, developed to detect and exclude malfunctioning digital registers to ensure the correct configuration of all components utilized in experiments. However, due to dependencies between components of the hardware and detected instability in the digital control chain of some HICANNs, additional steps are necessary to circumvent all problems arising from wrongly configured digital registers. In addition, the flexibility of the availability database and its utilization in different software layers can be exploited to take into account chip versioning and optimizations of the map and route algorithm. Throughout all steps of this process, the previously acquired communication and memory test results are evaluated, and no additional measurements on the hardware are made.

Executed after the memory test, this process allows to exclude all the components that should not be used during experiments, not only due to their individual problems, but also because of their dependencies on other components. Consequently, without further knowledge about low-level dependencies, the hardware can be treated as a perfect substrate for experiments by the user. Exclusion steps covering all expected hardware dependencies were developed in the context of this thesis and are discussed in more detail in the following. The numbers of thereby excluded components on wafer 30 are listed in table 4.4.

Handling of unstable repeater registers: The memory test results reveal that most registers on a wafer can be reliably programmed. Only a small number of synapse and repeater related registers show varying results, demonstrated in appendix A.2. Both components are organized in larger building blocks, on which individual components are programmed by custom on-chip SRAM-controllers. Due to the increased complexity in the digital control chain and the fact that errors are isolated to individual blocks, excluding entire blocks exhibiting components with unstable behavior is expected to effectively address the issue. To this end, for the synapses a stability test was developed to distinguish individual malfunctioning registers from problems arising due to unstable building blocks, as introduced in the previous section. In contrast, the digital registers of the repeaters only represent a small amount of the tested memory. Consequently, individual failing registers are very unlikely. As a result of this, repeater blocks that host more than one repeater failing the memory test are considered unstable, and thus all repeaters controlled by this block are excluded from the availability database.

Buses connected to malfunctioning repeaters: Repeaters are used to regenerate the signals

Table 4.4.: Details of excluded components after the effective exclusion for wafer module 30. The values extend the memory test results shown in table 4.2. Only affected resources are listed. Additionally, for each resource, the number of available components on the whole wafer is shown. Exclusions caused by hardware versioning are not shown, as these would include components that are not present in wafer 30. The results represent the number of components unavailable for experiments, but these components do not necessarily show malfunctioning behavior.

Resource	#Components	Excluded
JTAG communication	384	2.08%
High-speed communication	384	6.77%
Mergers	8501	1.79%
Repeaters	122 240	0.16%
Buses	122 240	0.79%

transmitted via the buses between two HICANNs. Therefore, each repeater is connected to buses on its own and the neighboring HICANN. If a repeater is excluded from the availability database, it is not ensured that it does not send wrong signals to its connected buses. To prevent this, all connected buses are also excluded.

Additionally, due to a bug in the control chain, the reset bit of the repeaters is not automatically released after powering up the chip, leading to incorrect signals on buses connected to powered but non-initialized HICANNs. Therefore, all buses connected to HICANNs without JTAG communication are also removed from the availability database, as these chips cannot be initialized. Extending beyond chip boundaries, this dependency rules out a simplistic approach of addressing malfunctioning components by merely removing entire chips, an approach which on top has the drawback of losing many usable components.

Malfunctioning switch registers: The observed quantity of malfunctioning switch registers exceeds the anticipated number attributed to manufacturing-induced issues. For these components, where no additional on-chip SRAM controller is involved in the programming process, problems in the digital control chain of the entire HICANN cannot be ruled out. However, digital memory tests are inconclusive in the presence of a malfunctioning control chain. Consequently, HICANNs with failing switch registers are entirely removed from the database. This is achieved by excluding the respective JTAG communication. As a consequence of this, the HICANN is no longer initialized during experiments. This is desired because, due to the potentially unreliable control chain, correct programming is not guaranteed. However, as explained earlier, this necessitates the exclusion of buses on the neighboring chips.

Malfunctioning floating gate controllers: On the HICANN, neuron parameters as well as routing-specific configurations are stored in floating gates. If a malfunctioning controller

programming the floating gates is found, the correct operation of neurons and repeaters is not ensured. To prevent the utilization of such chips, affected HICANNs are excluded by removing their JTAG communication from the availability database. Consequently, the HICANN is neither used for the placement of neurons nor for routing of signals.

HICANNs without high-speed connection: Injection and recording of spikes, as well as the configuration of neurons, is expensive in terms of bandwidth requirements. Although chips can be programmed via their JTAG connection, they cannot be employed for neuron placement or external input injection based solely on this communication method. Therefore, HICANNs without the possibility to communicate via high-speed are exclusively used for routing signals between chips. This is achieved by excluding all their neurons and external input mergers from the availability database.

HICANNs without routing possibilities: A spike sent by a neuron circuit must traverse the merger tree until it is injected via a sending repeater into the layer-1 interface. There, due to malfunctioning components, it might occur that no routing possibilities are left for functioning neurons. For this reason, any neuron and external input merger lacking at least one essential component for signal transmission is excluded from the availability database. Since different components are required for different merger tree routing algorithms, introduced in section 4.3.1, the script allows the user to select the desired strategy to generate experiment-specific availability data. As a consequence of this, resource utilization is enhanced in the routing process.

Hardware versioning: The availability database facilitates the management of various chip versions by adjusting the excluded components. At the time of writing, it serves two purposes. On the one hand, in the latest chip version, the number of synapse rows was reduced from 448 to 440 to make space for an extended neuron circuit [Koke 2017]. To prevent algorithms to take removed components into consideration, the synapse drivers of the 8 rows are excluded from the database for wafers built with the latest chip version. On the other hand, in an earlier version of the post-processing layer, connections were established from HICANNs at the corner of the wafer to unused chips beyond the area of the 48 utilized HICANN groups. To prevent leakage currents arising from these connections, all affected buses are excluded from the database. Consequently, no further distinctions have to be made in software loading the dedicated availability database for each wafer, regardless of its chip or post-processing versions.

Depending on the requirements of the user, different steps can be skipped or extended to find a suitable availability database for each experiment. Results of the effective exclusion are stored separately to disk so that malfunctioning and dependent components can be distinguished afterward. In addition, the memory test results are used to initialize the chips before experiments, while the results of the effective exclusion are used during experiments.

4.1.5. Calibration Based Exclusion

The final step to acquire the availability data for experiments is the calibration based exclusion. Since all components have to work correctly for the neuron calibration to succeed, this constitutes a test for all analog components of the chip. All neurons that do not successfully complete all calibration steps are excluded from the availability database. Failing calibrations are either caused by malfunctioning circuits or by outliers that do not satisfy defined thresholds. In the calibration based exclusion on wafer 30, at the time of writing, 11.29 % of the 195 584 neuron circuits are excluded. This number strongly depends on the requested neuron settings and set thresholds. For the biologically inspired large-scale experiments described in this thesis, stable neuron behavior is essential. Therefore, the calibration is optimized for stability rather than to maximize the number of usable neurons.

4.2. Extended Calibration for Large-Scale Experiments

The investigation of biologically inspired large-scale neural networks requires the precise tuning of neuron parameters. However, due to the analog nature of the BrainScaleS-1 chips, manufacturing-induced device variability is unavoidable. In addition, for each neuron, many hardware-specific parameters have to be set to obtain the desired neuron behavior. While for very small neuron numbers, manual adjusting of hardware parameters might still be possible, for large-scale experiments, this is no longer feasible. Therefore, calibration routines were developed in Koke 2017; Kleider 2017; Schmidt 2014 to allow for configuring the hardware in biological parameters and to minimize the variability of the circuits. Focused on in-the-loop training of comparatively small networks, these calibration routines were extended in the course of this thesis to cope with the requirements of large-scale experiments, which is this section's topic.

In order to become independent of predetermined settings of the investigated biological networks, the static parameter translation from the biological domain to the hardware domain was changed to an automated translation that always utilizes the whole dynamic range of the hardware circuits, which is introduced in section 4.2.1. Furthermore, saturation effects were found in the circuits of the reversal potentials that were not handled by the existing calibration. To allow for a precise setting of the reversal potential, a new calibration method was developed, shown in section 4.2.2. Moreover, in previous experiments the synaptic weights were either set manually or adjusted by in the loop training of the network, thus requiring no calibration. However, the investigation of biological networks requires the setting of predefined weight values in the biological domain. To this end, a weight calibration and automatic translation was developed, presented in sections 4.2.3 and 4.2.4. Finally, in order to enable simulations of the hardware behavior, the transmission delays between chips were determined, provided in section 4.2.5

4.2.1. Parameter Translation

The typical resting potential of neurons in the human brain is −70 mV, with the amplitude of an action potential being approximately 100 mV [Petrovici 2016]. In contrast, the BrainScaleS-1 system operates with voltages between 0 V and 1.8 V. Additionally, the time constants arising from its electronic components make up a typical acceleration factor of 10 000 compared to biological real time. Therefore, parameters have to be translated between the two regimes. This transition is possible since the network behavior of LIF neurons is independent of the absolute voltage values if all parameters are adapted accordingly. To this end, in previous experiments, the static parameter translation

$$U_{\text{hardware}} = 10 \cdot U_{\text{bio}} + 1.2\,\text{V} \tag{4.1}$$

for voltage parameters and

$$t_{\text{hardware}} = \frac{t_{\text{bio}}}{t_{\text{scaling}}} \tag{4.2}$$

for time constants was used, where t_{scaling} depends on the used hardware settings and is typically set to 10 000. Handled by the BrainScaleS-1 operating system, experiments are exclusively configured in the biological regime.

However, there are two major problems using a static translation for emulating biologically inspired networks. On the one hand, different biological network descriptions utilize different parameter ranges that require different translation parameters. For example, the two models investigated in this thesis use a resting potential of 0 mV for the balanced random network [Brunel 2000] and −65 mV for the cortical microcircuit [Potjans et al. 2012]. On the other hand, the static parametrization does not restrict the utilized parameter range to the hardware boundaries, which are further constrained by saturation effects present in the circuits of the reversal potentials, as discussed in section 4.2.2.

Both problems are solved by a dynamic parameter translation that always utilizes the whole available parameter range of the hardware. This is achieved by exploiting the characteristic of the LIF neuron that the reversal potentials are not exceeded by other voltages. Therefore, the reversal potentials are mapped to the maximum voltage U_{max} and minimum voltage U_{min} of the hardware, which are introduced in section 4.2.2. This results in the translation

$$U_{\text{hardware}} = \frac{U_{\text{max}} - U_{\text{min}}}{E^{\text{e}}_{\text{rev}} - E^{\text{i}}_{\text{rev}}} \cdot U_{\text{bio}} + \frac{E^{\text{e}}_{\text{rev}} \cdot U_{\text{min}} - E^{\text{i}}_{\text{rev}} \cdot U_{\text{max}}}{E^{\text{e}}_{\text{rev}} - E^{\text{i}}_{\text{rev}}}. \tag{4.3}$$

Here, $E^{\text{e}}_{\text{rev}}$ is the excitatory and $E^{\text{i}}_{\text{rev}}$ the inhibitory reversal potential of the biological model. In this way, used voltages never exceed hardware boundaries and the maximum voltage resolution is achieved by utilizing the whole dynamic range of the circuits.

4.2.2. Extended Reversal Potential Calibration

The excitatory and inhibitory reversal potentials of the LIF neuron constitute the boundaries for all voltages of the model. Their precise configuration on the hardware is

fundamental for the parameter translation from the biological to the hardware domain and therefore a prerequisite for the weight calibration.

In the previously existing calibration routine, the membrane voltage of the neuron was clamped to one of the two reversal potentials using a strong input current. Subsequently it was read out using the analog readout system. In this manner, a corresponding reversal potential was determined for various configurations of the respective floating gate, which are specified in units of LSB and therefore represent the value of the 10 bit register used for its programming. This method will be called direct measurement in the following.

Although providing accurate results for the inhibitory reversal potential, deviating values are obtained when measuring the excitatory reversal potential using this approach [Wehrheim 2019]. The reason for this is that, similar to its biological counterpart, the membrane potential on the hardware is not designed to reach the excitatory reversal potential since it is typically above the threshold value of the neuron. Therefore, the circuits are not optimized to operate close to this operation point and show non-linear effects when approaching the excitatory reversal potential. This unintended behavior can be made visible by measuring the PSP height of a stimulated neuron at different resting potentials. In the conductance-based LIF neuron model, the height of a single PSP starting from the resting state can be described by

$$h = \frac{w\tau_{\mathrm{syn}}(E_{\mathrm{rev}} - E_{\mathrm{rest}})}{g_{\mathrm{leak}}(\tau_{\mathrm{m}} - \tau_{\mathrm{syn}})} \left(\frac{\tau_{\mathrm{syn}}}{\tau_{\mathrm{m}}}^{\frac{\tau_{\mathrm{syn}}}{\tau_{\mathrm{m}} - \tau_{\mathrm{syn}}}} - \frac{\tau_{\mathrm{syn}}}{\tau_{\mathrm{m}}}^{\frac{\tau_{\mathrm{m}}}{\tau_{\mathrm{m}} - \tau_{\mathrm{syn}}}} \right) \tag{4.4}$$

with the same notations used in section 2.2, taken from Koke 2017. Considering only modifications of the resting potential, all other neuron parameters can be substituted into the constant C, resulting in the linear dependency

$$h = C(E_{\mathrm{e,i}}^{\mathrm{rev}} - E_{\mathrm{rest}}). \tag{4.5}$$

Measurement results of the PSP height of a single neuron circuit for different settings of the excitatory reversal potential are shown in fig. 4.6a. The expected linear behavior is is indeed observed, but only when the membrane potential is distant from the reversal potential. Approaching it, the PSP height drops. Consequently, using the direct measurement method leads to the determination of an incorrect reversal potential.

To circumvent this, the indirect measurement method was developed. Here, the reversal potential is obtained by extrapolating the linear regime of the PSP height measurement to the resting potential value where the height reaches 0 V. At this point, according to eq. (4.5), the used resting potential is equal to the reversal potential the neuron is actually affected by in the linear regime, where it is exclusively operated during experiments.

Calibration results of the direct and indirect method measured on a single neuron circuit are compared in fig. 4.6b. On the one hand, due to the non-linear behavior of the circuits close to the excitatory reversal potential, the direct method underestimates the correct reversal potential. On the other hand, the error of the indirect calibration is larger, given its additional dependency on variations in all other neuron parameters, stemming from the necessity to determine the PSP height.

Extrapolating from 4 measured PSP heights for 4 different values of the reversal potential, respectively, all neuron circuits of the wafer are calibrated. Results before

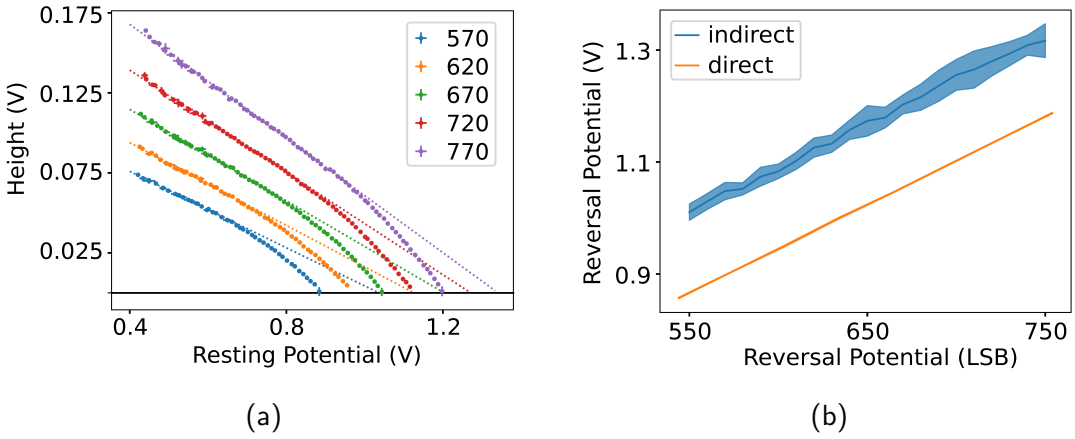


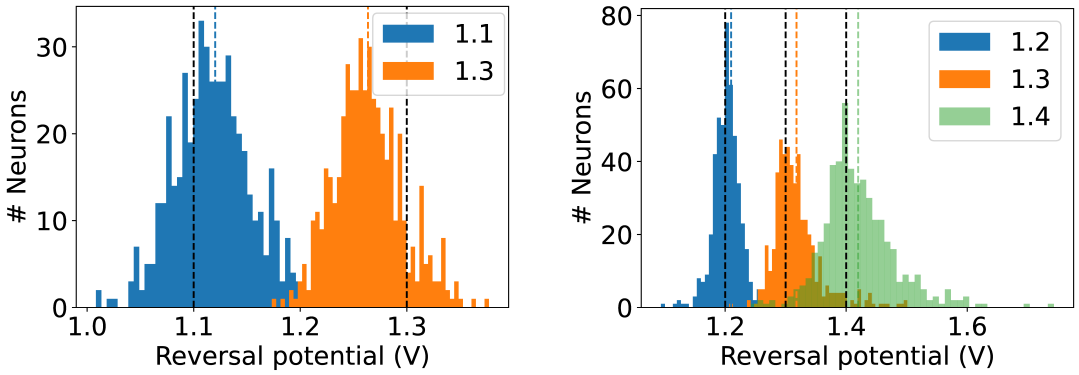
Figure 4.6.: Excitatory reversal potential calibration. (a) Indirect measurement of the excitatory reversal potential. The PSP height of a stimulated neuron is extracted for different resting potentials. Different colors indicate distinct hardware settings of the reversal potential in LSB. Non-linear behavior is observed for small distances between membrane potential and reversal potential. A linear extrapolation of the linear region (dotted line) is used to extract the correct reversal potential. (b) Comparison of direct and indirect calibration of the excitatory reversal potential. The direct measurement underestimates the reversal potential, while the indirect measurement exhibits a greater degree of uncertainty. Taken from Schmidt et al. 2023.

and after the reversal potential calibration of all neurons of one HICANN on Wafer 30 are shown in fig. 4.7. Due to the calibration, the distribution of obtained voltages is narrowed and shifted towards the expected mean value. Even if the uncertainty during the measurement leads to a larger deviation compared to results of the direct method, the obtained mean values of the indirect method correspond to the values seen by the neurons in their operated range.

The intended voltage range in which the circuits of the HICANN should be operated is between approximately 0.4 V and 1.4 V and the membrane potential must be kept below approximately 1.2 V, which is the maximum voltage the membrane can reach. As a result of this, the deviation of the calibration at 1.4 V increases as some neuron circuits are already reaching this boundary. For even higher hardware voltage settings, the measured mean reversal potential always corresponds to the maximum of 1.4 V.

Similar behavior is observed for the inhibitory reversal potential. Although no non-linear behavior of the circuits is found when operated in the intended regime above 0.4 V, as illustrated in the measurement of a single neuron circuit in fig. 4.8a, for smaller settings of the reversal potentials, the direct and indirect measurement results deviate due to non-linear circuit behavior, as shown in fig. 4.8b.

From measurements on all neuron circuits, a lower voltage boundary of 0.45 V is



(a) (b)

Figure 4.7.: Performance demonstration of the excitatory reversal potential calibration. Both figures present reversal potentials obtained from all neurons within a single HICANN using the indirect measurement method. Different colors indicate distinct target values to which the reversal potential is configured. Neurons are configured using (a) a default parameter translation or (b) the extended excitatory reversal potential calibration. Black dotted lines signify the configured mean values, while colored dotted lines represent the measured mean values of the corresponding histograms.

identified for the inhibitory reversal potential. Above this threshold, the already existing direct-measurement calibration method can be applied, and the circuits exhibit the intended behavior.

Constraining the operating range of the circuits to the obtained boundaries is achieved through the parameter translation, introduced in section 4.2.1. By default, this translation fixes the excitatory and inhibitory reversal potentials to 1.3 V and 0.45 V, respectively.

4.2.3. Synaptic Weight Calibration

In the BrainScaleS-1 system, the configuration of the synaptic weight is special compared to all other parameters due to its increased complexity. Additional to the 512 individual neuron circuits, there are 110 synapse drivers that generate the synaptic input signal for each neuron. The strength of this signal is configured per driver by two hardware parameters, V_{gmax} and g_{div}, and is further modified by the digital parameter w stored per synapse. On top of that, the neuron's time constants, the membrane capacitance that changes with the number of interconnected neuron circuits (cf. section 3.1.1), and the reversal potentials also affect the strength of the final stimulation. Consequently, with the current analog readout possibilities of the system, a precise per-circuit weight calibration would exceed reasonable runtimes [Schmidt et al. 2023].

In previous experiments, weights were manually adjusted or learned during hardware in-

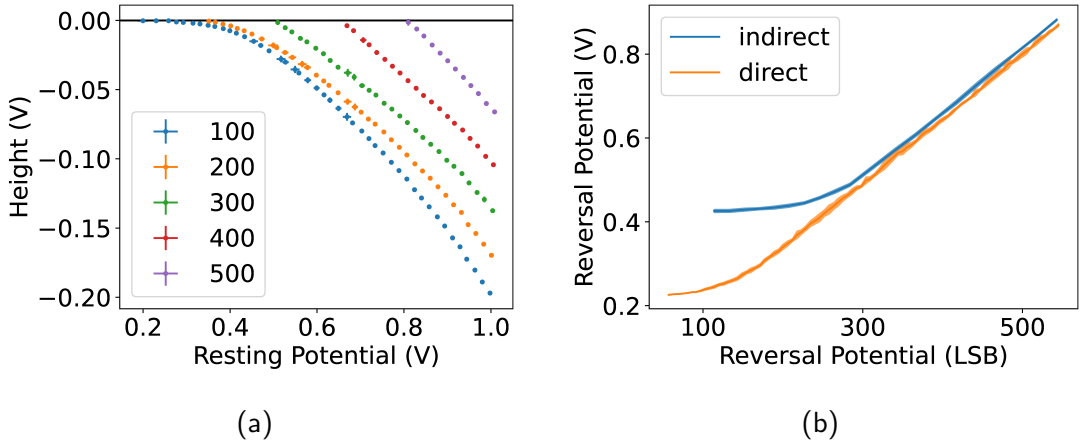


Figure 4.8.: Inhibitory reversal potential calibration. (a) Verification of the linear behavior of the inhibitory reversal potential. The PSP height of a stimulated neuron is extracted for different resting potentials. Different colors indicate distinct hardware settings of the reversal potential in LSB. Non-linear behavior is only observed for reversal potential settings below 300 LSB. (b) Comparison of direct and indirect calibration of the inhibitory reversal potential. Results deviate for values below 0.45 V.

the-loop experiments [Göltz et al. 2021; Kungl et al. 2019; Schmitt et al. 2017]. However, the manual adjustment of each weight is no longer feasible for large-scale experiments. In addition, the goal of the experiments carried out in this thesis is to investigate the spiking behavior of biologically inspired networks with a given parametrization, which requires defining weights in the biological regime.

Therefore, a per-wafer calibration that configures the weights to match the mean weight of all circuits of one wafer is developed in this thesis. This is possible, since only a small fraction of the available synaptic input circuits has to be investigated there. This per-wafer calibration and its dependency on all other neuron parameters are discussed in this section.

The only two observables on the HICANN that can be used for calibration are the membrane potential and spike times of the neurons. In order to determine the strength of a synapse for a specific hardware setting, the voltage trace of a stimulated neuron's membrane is recorded. To reduce the noise in the measurements, which partially originates from the analog readout itself, the neuron is stimulated by 100 consecutive spikes. Subsequently, the recorded membrane trace is segmented, the segments are overlapped, and their mean value is extracted. A comparison between the raw data and the resulting mean trace is shown in fig. 4.9.

Following this, the differential equation of the conductance-based LIF neuron model, given in eq. (2.1), is fitted to the data using *scipy curve_fit* [Jones et al. 2001] to obtain the parameters of the emulated neuron, in particular the desired weight parameter.

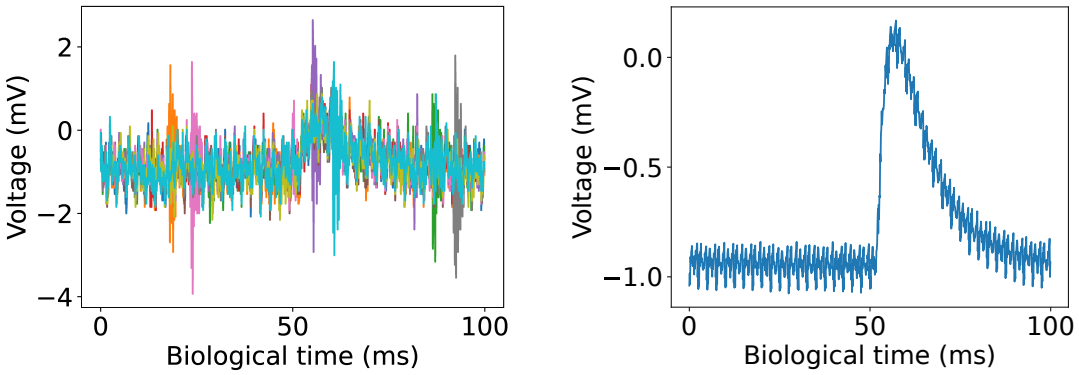


Figure 4.9.: Signal quality of membrane voltage recordings during weight calibration. (a) shows the membrane voltage of a continuously stimulated neuron, where the traces of 10 out of 1000 recorded PSPs are cut and overlapped. The intended stimulation of the membrane causes the voltage increase at approximately 60 ms that is common in all traces. Further deviations of the membrane voltage are caused by noise partially added by the readout itself and therefore not present on the chip. In (b) the mean trace of all 1000 overlapped PSPs is calculated to improve the signal quality.

To minimize variations caused by a different parametrization during calibration and experiment, all the other parameters are pre-calibrated and set to the values of the experiment under investigation. On top of that, in the following, 3 neuron parameters are investigated in particular due to their major impact on the result of the fit.

First of all, observing the PSP trace caused by a single spike of a conductance-based LIF neuron, a change of the reversal potential cannot be distinguished from a change in the synaptic weight, cf. eq. (2.12). Consequently, it is not possible to correctly fit the synapse weight and reversal potential at the same time. Therefore, the value of the reversal potential is fixed during the fit. As a result of this, the weight calibration is only valid if the setting of the reversal potential during calibration is also used during experiments. This was taken into account and addressed while developing the extended reversal potential calibration, introduced in section 4.2.2, and the parameter translation, discussed in section 4.2.1. Another consequence of fixing the value of the reversal potential during the weight calibration is that variations of the reversal potential add up to the variations observed during the weight calibration.

Secondly, on the HICANN it is not possible to directly measure the capacitance C_{HW} of the neuron's membrane, which is required to extract the exact weight value from the fit. However, similar to the membrane time constant, given by $\tau_{\mathrm{m}} = C_{\mathrm{HW}}/g_{\mathrm{leak}}$, it is possible to characterize the stimulation strength through the ratio between the weight and the neuron's membrane capacitance $w_{\mathrm{bio}}/C_{\mathrm{HW}}$. As introduced in section 3.1.1, there are

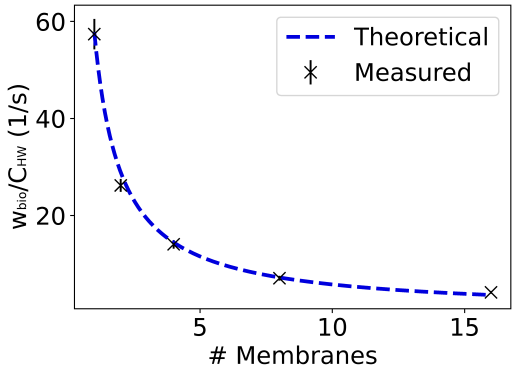


Figure 4.10.: Weight dependency on the number of connected membrane circuits. A single neuron is stimulated by a single synapse with a fixed weight configuration ($w = 15$, $g_{\mathrm{div}} = 8$, $V_{\mathrm{gmax}} = 1000\,\mathrm{mV}$). The ratio between weight and membrane capacitance $w_{\mathrm{bio}}/C_{\mathrm{HW}}$ is determined for varying numbers of connected membrane circuits that the neuron is constructed from. The dashed line represents the expected behavior of the weight relation, assuming a linear dependency between membrane capacitance and number of connected membrane circuits, normalized to the mean weight value measured for a single membrane circuit.

only two distinct settings for the membrane capacitance on the HICANN. Therefore, a separate calibration is done for each of them. In addition, the capacitance of the neuron's membrane depends on the number of interconnected neuron circuits as their capacitors are connected in parallel. Consequently, a linear increase in capacitance is anticipated with these interconnections. This linear dependency is confirmed by measurements with fixed neuron parameters for various sizes of the recorded neuron, as illustrated in fig. 4.10. As a result of this, during experiments, the calibrated weight-to-capacitance ratio can be rescaled to the used neuron size. Furthermore, in this thesis, the calibration is executed with a neuron size of eight to match the configuration used during experiments. It is subsequently normalized to a neuron size of one to ensure universal applicability, as discussed in section 4.2.4.

Finally, the shape of the PSP is expected to be symmetrical for the synaptic time constant τ_{syn} and the membrane time constant τ_{m} (cf. eq. (2.12)). Due to the observed deterioration in fit performance when swapping their values, these parameters are constrained to $\pm 20\,\%$ of their anticipated values during the fitting process. However, as demonstrated in Schmidt et al. 2023, the time constants exhibit significant deviations during calibration. Therefore, on some HICANNs, the actual values might surpass the constraints of the fit, leading to incorrect parameter assumptions. To address this and identify and reject erroneous fit results, quality measurements are introduced, as detailed in the following.

To guarantee a correct weight estimation, each fit has to fulfill different quality

measurements. Therefore, the reduced χ^2 value

$$\chi^2_{\text{red}} = \frac{1}{\nu} \sum_{i=0}^{N} \frac{(v_i^{\text{fit}} - v_i)^2}{\sigma^2_{\text{err}}} \tag{4.6}$$

of the fit is calculated, where $\nu = N - p$ is the number of degrees of freedom, N is the number of measurement points, p is the number of free parameters of the fit, v_i^{fit} are the fit values, v_i are the measured values, and σ^2_{err} is the estimated error of the measurement given by the standard deviation of a recording of the membrane voltage of the same neuron without stimulation. This allows for the rejection of traces that show large deviations between model and measurement that are most likely caused by variations of the time constants or saturation effects, mostly found for strong excitations of the membrane, where the circuits leave their linear range. There, an upper limit of $\chi^2_{\text{red}} = 2$ was found to be a good match between fit quality and number of rejected traces.

In addition, an estimation of the signal-to-noise ratio is done. For small weights, the probability of failing fits increases since PSPs can no longer be separated from the noise. For this reason, the ratio between the standard deviation of the neuron's membrane trace with and without stimulation is calculated and traces with values below 1.7 are rejected.

Finally, the recorded membrane voltage is checked for negative peaks that could be caused by wrong configuration of the neurons or measurement artifacts. There, traces are rejected if the distance between baseline and minimum value is larger than the distance between baseline and maximum value. This most likely happens for small weights due to the expected lower peak height.

In general, traces can be rejected by several quality measurements. The typical percentage of rejected traces during a weight calibration on wafer 30 is shown in fig. 4.11. As expected for large weight values, more fits are rejected by the reduced χ^2 value, since the circuits leave their linear range. In contrast, for smaller weights, the reduced signal-to-noise ratio leads to more rejections. This is especially a problem for very low weight configurations, which are affected to a large extend by the parasitic capacities of the circuits. There, only strong PSPs can be identified, which leads to a wrong weight estimation in this region.

In total, approximately half of the traces get rejected, whereby a large deviation is found between neuron circuits. This adds a bias to the available data used for the calibration. Nevertheless, apart from very low weight settings, the remaining measurements are found to be sufficient to find a satisfactory average weight translation.

In the first part of this section, the method to extract the biological weight expected for different weight configurations of the hardware is presented. This data is used to fit the expected hardware behavior given by

$$w_{\text{bio}} = A\left(\frac{w \cdot V_{\text{gmax}} \cdot g_{\text{scale}}}{g_{\text{div}}} + i_0 + i_1 \cdot w_1 + i_2 \cdot w_2 + i_4 \cdot w_4 + i_8 \cdot w_8\right) \tag{4.7}$$

that allows for translating biological weights into an appropriate set of the three hardware parameters V_{gmax}, g_{div} and the digital weight w. Equation (4.7) is adapted from eq. (3.4) without additional correction terms due to their negligible impact on the fit accuracy

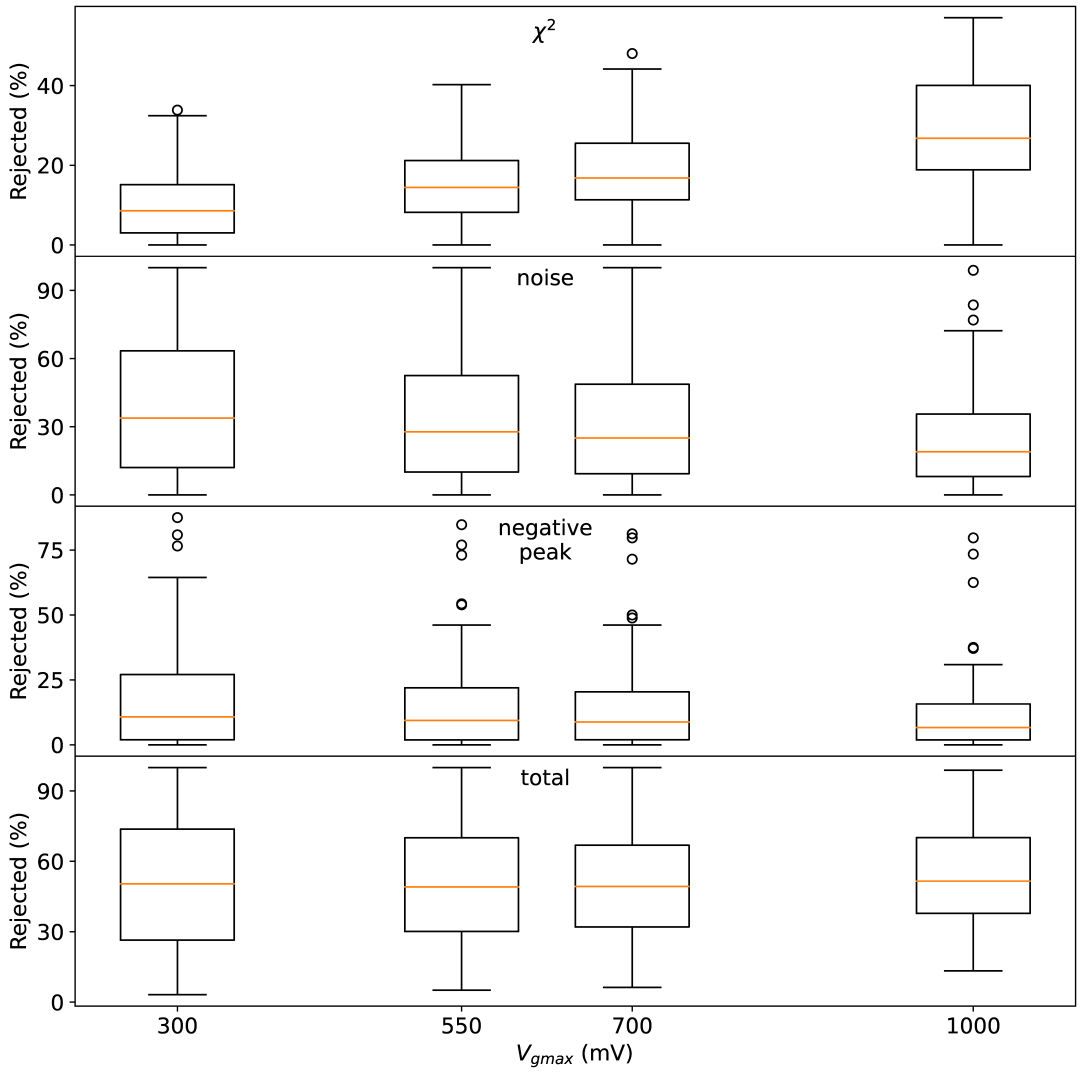


Figure 4.11.: Rejected membrane recordings during weight calibration. Shown is the percentage of rejected recordings of measurements taken for all combinations of the 16 possible digital weight values w and the g_{div} values $2, 8, 15$, and 25 with 4 repetitions per setting. From top to bottom, the rejections due to the χ^2 criteria, the signal-to-noise ratio, and the negative peak detection are displayed (cf. section 4.2.3), followed by the total number of rejected traces. Each box displays the deviation between measurements done on 10 neurons on 12 HICANNs, respectively. To investigate the dependency on the synaptic strength, different boxes represent the results for different V_{gmax} settings. Circles represent outliers.

taking the variations caused by the rewrites of the floating gate and the per-wafer calibration into account.

Figure 4.12 shows measured data and the found weight calibration for a single neuron circuit stimulated by a single synapse driver using a single V_{gmax} value without rewriting the floating gates during the measurement. The fit accuracy suffices to describe the influence of the parasitic capacities localized in the synaptic input line, shown in detail in fig. 4.12b. Therefore, the non-linear weight increase that is caused by input currents added for each enabled bit of the digital weight is taken into account by the calibration for each combination of V_{gmax} and g_{div}.

However, since the parametrization of the neuron changes slightly with each rewriting of the floating gates, additional variations are added to the measurements, demonstrated in fig. 4.13a. On top of that, the necessity to record neuron traces for 110 synapse drivers per neuron in combination with the time required to reconfigure the floating gate value V_{gmax} exceeds reasonable runtimes using the possibilities of the current analog readout module attached to the BrainScaleS-1 system. As a consequence, the per-wafer calibration is chosen, where only a subset of neuron circuits is calibrated to find the average translation between hardware configuration and expected biological weight, which is later applied for all circuits.

Since the per-wafer calibration is not circuit specific, the found values are less accurate, as demonstrated in fig. 4.13b. A higher precision is achieved by a per-neuron calibration, where variations of individual circuits can be corrected, and malfunctioning components can be detected and removed from the availability database. With the current readout system, this is only possible for measurements with a constant floating gate value V_{gmax}, changing only the faster programmable digital weight parameters. However, this would result in a reduced parameter range for the weights, which is not sufficient for the experiments conducted in this thesis.

Nevertheless, the presented per-wafer method preserves the mean weight value, which is most important for the biologically inspired networks investigated in this thesis [Dasbach et al. 2021]. Moreover, the found weight variations can be used to emulate the Gaussian weight distributions of the investigated networks. For this reason, the correlation across synapses is examined in the following.

Although different neuron circuits are expected to be independent of each other, they are stimulated by a common set of synapse drivers and each neuron is connected via the same input circuit to different synapses (cf. section 3.1.5). Consequently, synapses on the same HICANN are not independent. The existence of row-wise or column-wise correlations would be undesired for modeling weight deviations. Instead, similar variations across both rows and columns would be more suitable. This is investigated in fig. 4.14a, which shows the weight variations for different neuron circuits with a fixed stimulus in comparison to a fixed neuron circuit stimulated by different synapse drivers. On the one hand, the non-negligible variations between synapse drivers demonstrate the necessity to investigate all possible combinations of drivers and neurons during a potential per-circuit calibration. On the other hand, the average variation between synapse drivers is smaller than between neuron circuits. Consequently, stronger correlations between weights connected to the same neuron, i.e. column-wise, are expected. Moreover, additional

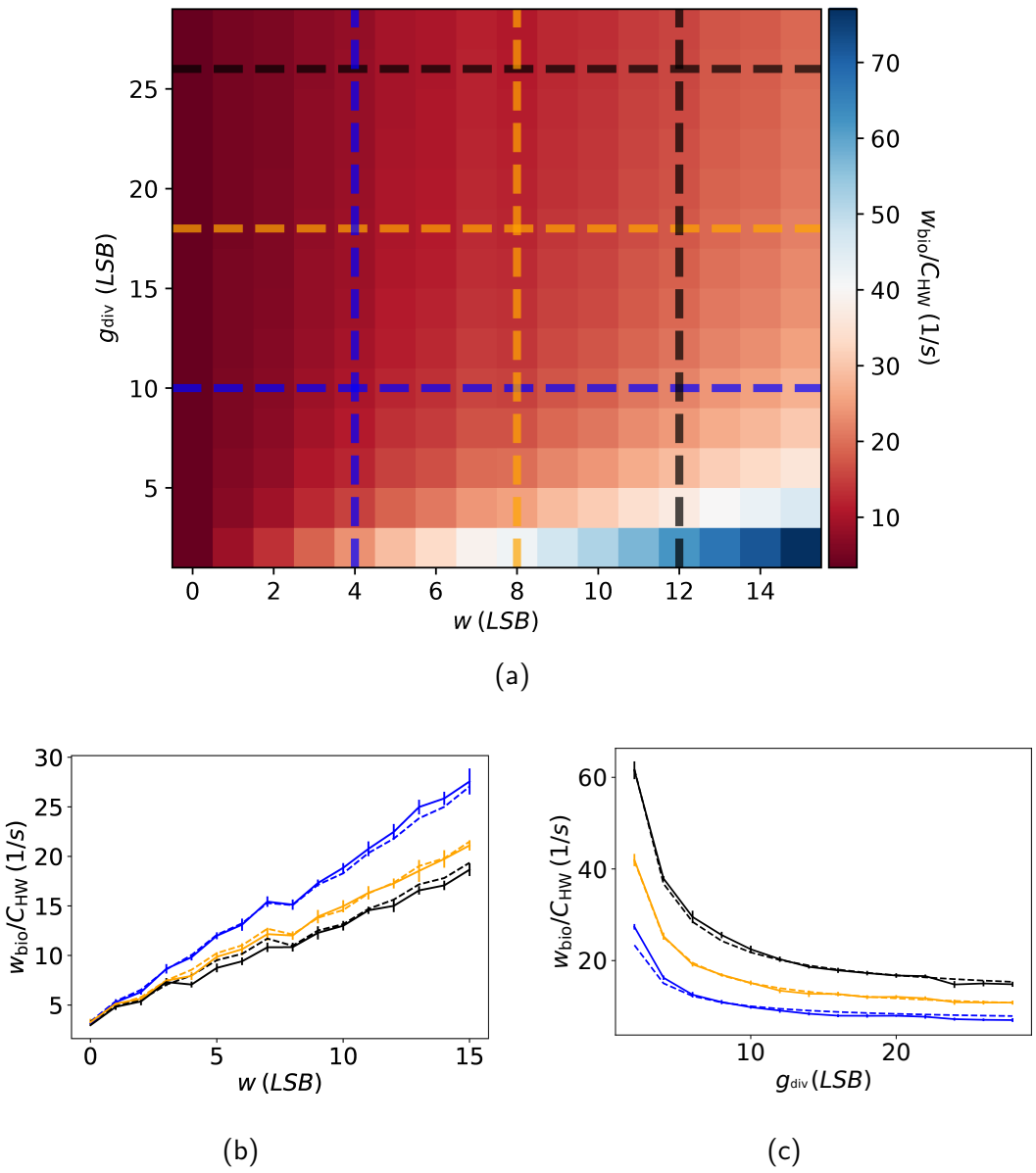


Figure 4.12.: Results of the synapse weight calibration for a single neuron circuit. (a) Weight measurement for different settings of the digital weight parameter w and hardware parameter g_{div} with $V_{\mathrm{gmax}} = 700\,\mathrm{LSB}$. Horizontal dashed lines indicate cuts with fixed values of the hardware parameter g_{div}, shown in (b); vertical dashed lines indicate cuts with fixed digital weight values w, shown in (c). In (b) and (c), solid lines represent measured values, and dashed lines illustrate fit results. The fit is obtained by applying eq. (4.7) to the whole measured parameter space. Taken from Schmidt et al. 2023.

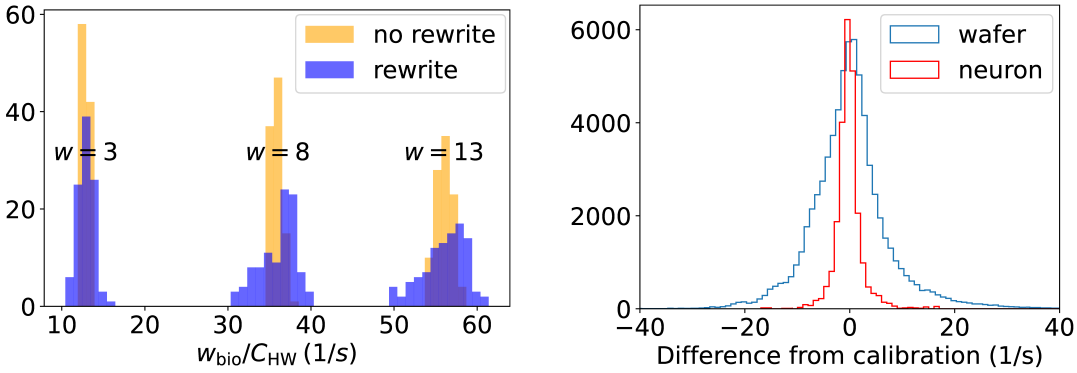


Figure 4.13.: (a) Variations of weight measurements with and without rewriting the floating gates. Values are extracted for 3 digital weight parameters w from a single neuron with fixed hardware parameters ($V_{\mathrm{gmax}} = 700\,\mathrm{LSB}$, $g_{\mathrm{div}} = 2\,\mathrm{LSB}$). (b) Comparison of a per-wafer and a per-neuron weight calibration. Measurements across the entire parameter spectrum of the synaptic input are carried out on a subset neurons. Subsequently, the calibration fit is applied to the entire subset or executed individually for each neuron. The histogram depicts the difference between measured and expected values derived from the respective fit. Taken from Schmidt et al. 2023.

variations are found for the four floating gate cells that are common for 55 synapse drivers to store the four V_{gmax} values each driver can be configured with, demonstrated in fig. 4.14b. Although the obtained variations are smaller compared to changing the driver or neuron circuits, they add a correlation to all weights that are connected to the same floating gate cell. Nonetheless, given that the per-wafer calibration conducted in this thesis does not yield per-circuit information, coupled with the limited configurability of the hardware weight as discussed in section 4.2.4, there is no possibility to adjust the weight distribution during experiments. For this reason, the variations are left unmodified during emulations.

All previous measurements are executed using the excitatory inputs of the neurons. However, the inhibitory input circuits can be calibrated accordingly using the same routines on the inverted membrane potentials. A comparison of calibration results of the excitatory and inhibitory input is shown in fig. 4.15. There, for a fixed set of parameters a slightly smaller biological weight is found for the inhibitory circuits. Most likely, this is caused by a wrong assumption of the reversal potential during the calibration. As described above, during the calibration the reversal potentials have to be fixed to the value finally used during the experiment. Assuming a wrong value of the reversal potential, the weight is wrongly scaled by an additional factor. This is demonstrated in fig. 4.15 with an

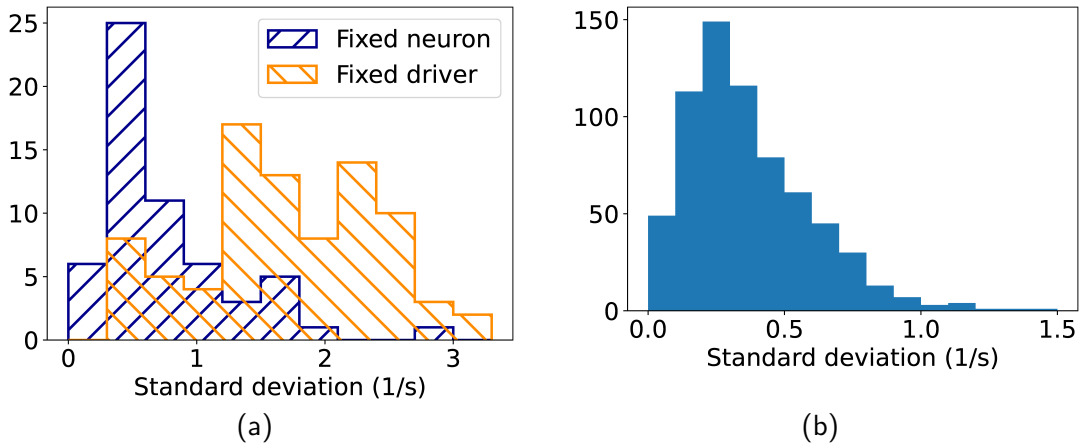


Figure 4.14.: Weight variations expected from different components involved in the synaptic input line. For all measurements, the weight of a stimulated neuron is extracted for a fixed setting of $w = 15$, $g_{\text{div}} = 8$, and $V_{\text{gmax}} = 1000\,\text{mV}$. To rule out variations caused by the floating gates, each weight value is determined by the mean value of 100 repetitions, with rewritten floating gates in between. Reliability of the extracted data is ensured by rejecting all weight values where less than 50 repetitions pass the quality measurements. (a) Comparison of deviations caused by neuron circuits or synapse drivers. Shown are the standard deviations of the weights measured for either 7 different synapse drivers stimulating a fixed neuron or a fixed driver stimulating 7 different neurons. A fixed neuron demonstrates the variations expected from different synapse drivers, and a fixed synapse driver the variations expected from different neuron circuits. In total, all combinations of 7 synapse drivers and 7 neurons on 12 HICANNs are measured. There, only if a weight value for at least 3 different neuron or driver circuits is found their standard deviation is considered. (b) Deviation between different V_{gmax} input lines. The histogram shows the standard deviation of the weights measured for the 4 possible V_{gmax} input lines of a single synapse driver stimulating a fixed neuron. A floating-gate value of $V_{\text{gmax}} = 1000\,\text{mV}$ is configured for each of the four input lines. The measurement is done for 775 neurons on 78 randomly chosen HICANNs. However, neurons are only considered if a weight value is found for all 4 floating gate connections.

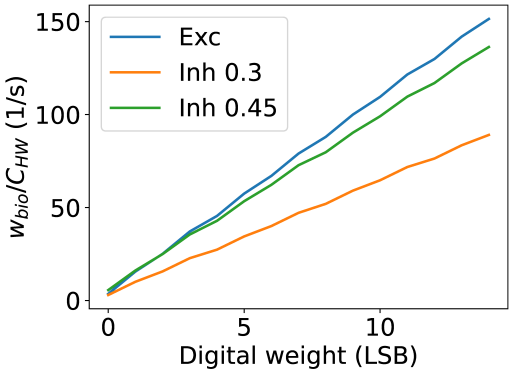


Figure 4.15.: Comparison of the weight calibration results for excitatory and inhibitory input circuits. All lines show the expected weight ratio $w_{\text{bio}}/C_{\text{HW}}$ expected from the weight calibration for different digital weight values w with the fixed settings $V_{\text{gmax}} = 1000\,\text{mV}$ and $g_{\text{div}} = 2$. "Exc" shows the calibration results of the excitatory synaptic input and "Inh 0.45" the results of the inhibitory synaptic input with a reversal potential set to 0.45 V. In addition, the result of a weight calibration done for the inhibitory synaptic input with a wrongly configured reversal potential of 0.3 V that cannot be reached on the hardware is demonstrated with "Inh 0.3".

additional inhibitory calibration utilizing a reversal potential value of 0.3 V that cannot be reached by the circuits, as described in section 4.2.2. Since the distance between resting and reversal potential is overestimated during the calibration, the derived weight is too low. Although the reversal potentials used for the other calibrations are within the designed range of the circuits, small offsets of the mean value of the less accurately obtained excitatory reversal potential could explain the found difference between the excitatory and inhibitory weight calibrations.

4.2.4. Implementation of the Weight Translation

To allow for parametrizing experiments using biological weight values, once the weight calibration is found, the hardware has to be configured accordingly. For this reason, the targeted biological weight has to be translated into the hardware domain and for each synapse an appropriate set of hardware parameters has to be found (cf. section 3.1.5). Parametrization is made more complex by the fact that not all weight related hardware parameters can be set independently. Furthermore, the weight translation, given by eq. (4.7), allows for different parametrizations that lead to similar biological weights. Therefore, a routine was developed that addresses these circumstances and by utilizing the weight calibration results automatically determines hardware settings for all synapses that provide the closest representation of their targeted value. This routine is presented in this section.

Initially, as demonstrated in the following, a hardware-setting independent target weight w_{target} is calculated for the biological weight of each synapse, aligning with the parameters of the weight calibration. As explained in the previous section, the weight calibration is conducted for the ratio between weight and membrane capacitance $w_{\text{rel}} = \frac{w_{\text{bio}}}{C_{\text{HW}}}$, without determining C_{HW} explicitly. This is possible since the only other quantity of the LIF model that depends on the membrane capacitance, the membrane time constant $\tau_{\text{m}} = \frac{C_{\text{HW}}}{g_{\text{leak}}}$, is calibrated using the identical neuron configuration. Consequently, the differential equation of the conductance-based LIF neuron can be parametrised as

$$\frac{dU}{dt} = \frac{1}{\tau_{\text{m}}}(E_{\text{rest}} - U) + \sum_i w_{\text{rel}}^i (E_{\text{rev}} - U), \tag{4.8}$$

which is independent of the absolute value of the membrane capacitance. Therefore, arbitrary membrane constants of the biological model C_{model} can be emulated as long as the hardware weight is set to resemble $w_{\text{rel}} = \frac{w_{\text{model}}}{C_{\text{model}}}$. Without measuring C_{HW} this is only possible if the hardware neurons are configured to the same membrane capacitance during calibration and experiment. On the one hand, this is achieved by recording separate calibration datasets for the two membrane capacitors available on the hardware. On the other hand, the linear increase of the membrane capacitance with the number of interconnected membrane circuits is compensated by rescaling the target weight with the size of the used neurons n_{neuron}. As a result of this, the target weight of each synapse is calculated by

$$w_{\text{target}} = \frac{n_{\text{neuron}}}{C_{\text{model}}} w_{\text{model}}. \tag{4.9}$$

Since the results of the weight calibration, which are treated like an inverse time parameter, are already stored in the biological domain, no further parameter translation has to be done to compensate for the speedup factor of the hardware at this stage.

Subsequently, the set of hardware parameters w, g_{div} and V_{gmax} has to be determined for each target weight. Due to the complexity of eq. (4.7), it is necessary to reduce the dimensionality of the problem, as multiple parametrizations exist that represent similar target weights. This is achieved by successively setting the parameters, starting with the least flexible, the four V_{gmax} values of each HICANN quadrant.

At this early stage of the configuration, it is not desirable to add constraints to the other parameters. Consequently, only V_{gmax} values that lie outside a suitable range to configure the target weights of the respective quadrant are excluded. Boundary values are calculated using the maximum and minimum target weight of the synapses of each quadrant and the maximum and minimum g_{div} setting, respectively. If the target weight cannot be reached, the maximum or minimum V_{gmax} setting is used. Results are rounded to the nearest integer value. The four floating gate cells of the respective HICANN quadrant are then configured with the found maximum and minimum values and two evenly spaced values in between. During all calculations, the digital weight parameter w is set to a user defined value w_{start} that is only increased if the current target weight cannot be reached within the limits of the other two parameters. Consequently, parametrizations using w_{start} for the maximum target weight are preferred by the algorithm. As a result

of this, it becomes feasible to define initial values for fast per-synapse weight updates during in-the-loop training or parameter sweeps during experiments.

Subsequently, for each synapse row, the appropriate floating gate cell, selected from the four available, and the best-matching g_{div} value are determined. Both values are optimized for the maximum target weight of the row to ensure all values can be reached. Therefore, the pair of available V_{gmax} and g_{div} values is obtained that minimizes the distance to the maximum target weight. Smaller weights on the same row are then addressed in the next step via the most flexible of the three parameters, the digital weight w.

As a final step, the digital weight w is set per synapse to get as close as possible to the target weight. With all other parameters already defined, the expected weights for all sixteen values of the digital weight parameter are calculated per synapse row and the nearest values which are higher and smaller than the target weight of each synapse is selected. All possible configurations are considered here to take the non-linear weight dependency caused by the parasitic capacities into account. The final digital weight setting is then selected through statistic rounding, proportional to the distance between the calculated weights and the target weight. This is done to prevent systematic strengthening or weakening.

Figure 4.16a demonstrates the weight deviations after the weight translation caused by shared configurations and the limited resolution of the hardware parameters. For the measurement, the worst-case scenario of uniformly distributed weights covering the whole weight range of the hardware and placed on a single HICANN is used. Neglecting the effect of the parasitic capacities, the maximum difference between expected and configured weight Δ_{max} can be estimated. It is obtained when the shared parameters are set to their maximum values and thereby enlarge the distances between the sixteen quantized weights. Moreover, a synapse that would ideally be represented by a specific digital weight value is configured to a different value due to stochastic rounding. In this case, the maximum difference can be approximated by

$$\frac{\Delta_{\mathrm{max}}}{\widetilde{w}_{\mathrm{exp}}} = \frac{A}{\widetilde{w}_{\mathrm{exp}}} \cdot \frac{V_{\mathrm{gmax}} \cdot g_{\mathrm{scale}}}{g_{\mathrm{div}}} \cdot \frac{1}{n_{\mathrm{neuron}}} \approx \frac{0.195\,\mathrm{s}^{-1}}{45\,\mathrm{s}^{-1}} \cdot \frac{1023 \cdot 0.4}{2} \cdot \frac{1}{8} \approx 0.11 \tag{4.10}$$

derived from eq. (4.7) with rescaled target weights according to eq. (4.9) considering a neuron size of 8 and normalized to the mean expected weight value $\widetilde{w}_{\mathrm{exp}}$ of the uniform distribution. In the measurement, slightly larger deviations are observed due to different parasitic terms that are added to the equation by different enabled bits of the digital weight value. The average deviation, however, is smaller since it is scaled with the shared parameter settings per row, which are kept as low as possible by the algorithm. Moreover, due to the stochastic rounding, the probability of selecting a specific weight configuration decreases linearly with the difference to the target weight, which explains the approximately linear decrease of synapses found for larger weight deviations.

Although the final distribution also depends on the routing algorithm and the number of synapses that get placed on each synapse row, due to the stochastic rounding the mean weight value is preserved. Therefore, the found deviations add up to the deviations expected from the circuit mismatch without changing the mean weight.

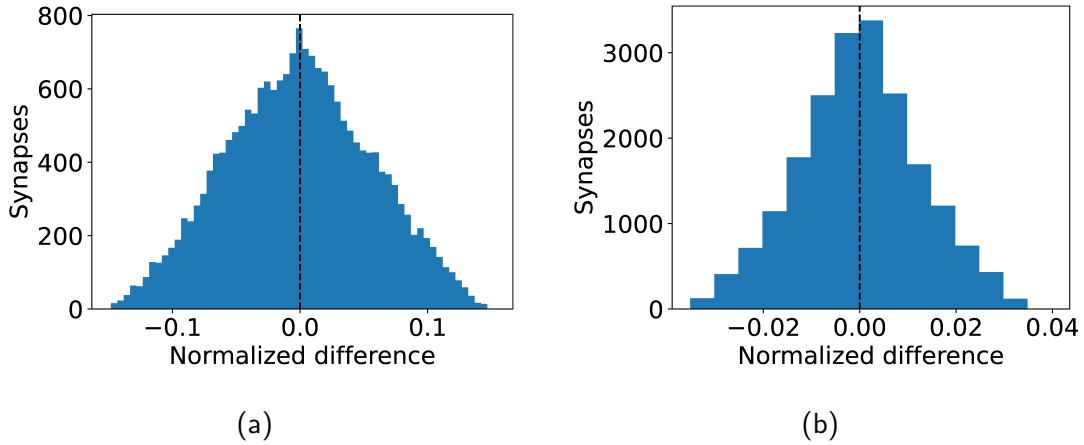


Figure 4.16.: Weight deviations expected from the weight configuration algorithm. A network of 50 neurons is placed on a single HICANN and stimulated by 20000 randomly connected synapses from an external population consisting of 50 neurons. The weights $\frac{w_{\text{model}}}{C_{\text{model}}}$ of the synapses are drawn from a uniform distribution between $15\,\text{s}^{-1}$ and $75\,\text{s}^{-1}$. For each synapse, the hardware parameters V_{gmax}, g_{div} and w are found by the weight translation algorithm considering shared configurations. These values are then translated back into corresponding biological weights, and the difference from the original weight is calculated and normalized to the mean original weight of all synapses. The histogram of the normalized differences is shown for a resolution of the digital weight parameter w of (a) 4 bit and (b) 6 bit. For the 4 bit resolution, the existing weight calibration of the small capacitor is used for the weight translation. Since no calibration is available for a hypothetical 6 bit resolution, the 4 bit calibration was rescaled so that both comprise the same parameter range. Additionally, the terms of the parasitic capacities (cf. eq. (4.7)) were removed in the 6 bit case. The black dashed lines mark the mean values of the distributions.

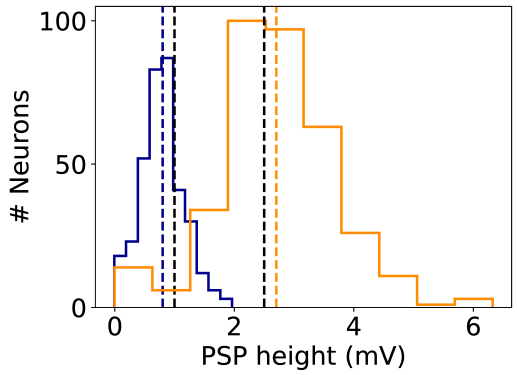


Figure 4.17.: Demonstration of the automated weight configuration using the weight calibration. Histograms of PSP heights, measured on a single neuron on each available HICANN within wafer 30, are displayed. Each neuron is stimulated by a single synapse configured with a biological weight value corresponding to a PSP height of either 1 mV (blue) or 2.5 mV (orange), as indicated by the nearest black dashed line. Colored dashed lines represent the mean values of the respective histograms.

In general, a more precise weight configuration is achieved by increasing the parameter space of the digital weight value. In fig. 4.16b, the same measurement is shown for a hypothetical resolution of the digital weight parameter of 6 bit, which reflects the weight resolution implemented in the next chip generation, the BrainScaleS-2 chip. There, the maximum weight deviation is reduced by a factor of 4. It is also possible to increase the weight resolution on the BrainScaleS-1 system by combining synapses to resemble a single synapse with higher resolution. However, due to the higher demand for synapses, which are the limited resource for large networks, it was not included in the routing algorithm and therefore not used.

Aside from that, the effect of the limited configurability is much smaller for the weight configurations of the networks investigated in this thesis. In both networks, excitatory and inhibitory synapses are configured to the same or similar weight values, respectively. Since excitatory and inhibitory synapses are never placed on the same synapse row, a precise weight configuration is achieved since only one target value is chosen, discussed in more detail in appendix A.3.

The final performance of the weight calibration in combination with the weight translation during an emulation is demonstrated in fig. 4.17. It resembles the weight requirements of the balanced random network model, introduced in section 2.4. As a result of the weight calibration, the same mean PSP height is found during emulation and simulation.

4.2.5. Delay Calibration

The delay of a connection between two neurons is the time interval between when an action potential is elicited at the pre-synaptic neuron and its arrival at the postsynaptic neuron's membrane. Consequently, it is predetermined by the network structure. However, the network descriptions investigated in this thesis are based on statistical evaluations of biologically inspired connectivity models. Therefore, connections between neurons are described by their average delay value. In contrast, similar to its biological counterpart, the delay on the BrainScaleS-1 hardware is defined by the physical distance between the neuron circuits. Neglecting RC-delays of the buses, the delay of a connection is given by the number of involved repeaters used to regenerate the signal between HICANNs. In addition, a constant offset is expected for the time the signal spends in the merger tree and synaptic input of the neurons, cf. sections 3.1.2 and 3.1.5. Consequently, it is not possible to adjust delays according to the investigated network descriptions.

To still be able to compare the hardware behavior with software simulations, the network model has to be adapted to match the hardware restrictions. To this end, an exact model of the network structure on the hardware is transferred to software. Although for each connection the number of repeaters is available in the routing results, it is necessary to estimate their contribution to the delay. Therefore, a delay calibration is performed.

For this reason, one neuron is configured to spike continuously by setting its resting potential above its threshold value. It stimulates a second neuron whose membrane potential is recorded. The delay between the two neurons is then represented by the time interval between spike time and start of the PSP on the recorded membrane.

Figure 4.18a demonstrates the principle of the delay measurement displaying a cutout of the membrane recording for two spikes. In total, the neuron is recorded for 1000 ms of biological time, which corresponds to approximately 25 spikes. Finally, the average delay of all recorded spikes determines the delay value of the connection. This measurement is repeated for connections comprising all repeater counts used during experiments. The resulting delay calibration is shown in fig. 4.18b. There, an offset of approximately 0.6 ms is observed that corresponds to the time the signal spends in the merger tree and the synaptic input. In addition, a linear increase of the delay with the number of repeaters is found. Finally, the result of a linear fit is used to translate the routing results into expected connection delays during software simulations.

The introduced delay measurements are based on the comparison of spike times recorded on the HICANN chip and membrane recordings done by the analog readout system. Both systems work with individual clocks that show variations from their nominal values. To prevent divergence between spike data and membrane recordings the results of each combination of HICANN and readout system is corrected for these variations during the delay calibration. Therefore, one background generator on the HICANN is used to produce a regular source of spikes. These spikes are routed to one of two synapse drivers from which the preout-debug signal can be recorded using the analog readout system. This debug signal is used to monitor the digital pulse generated by the synapse driver as a result of a spike. Consequently, it allows for measuring the interval of the

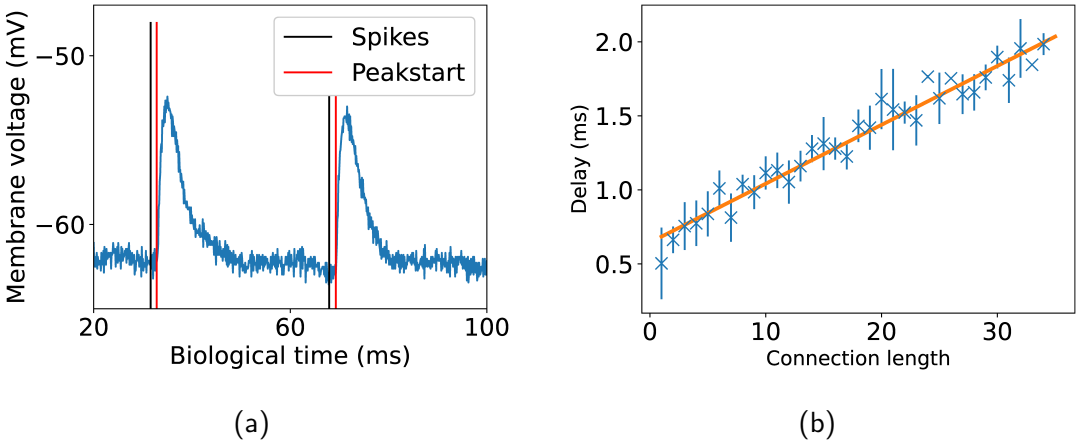


Figure 4.18.: Measurement of synapse transmissions delays on the wafer. (a) Membrane trace of a neuron that is continuously stimulated by a pre-synaptic neuron. The black line indicates the point in time the spike is sent from the pre-synaptic neuron, while the red line shows the estimated start of the PSP, i.e., the time the spike arrives at the postsynaptic membrane. The delay of the connection is given by the time gap. (b) Measured delay and linear fit for different connection lengths, characterized by the number of repeaters connecting two neurons. For each connection length, the delay between 7 randomly selected HICANNs is measured.

arriving spikes. By comparison with the frequency of the original spike source and the nominal values of the HICANN clock and the analog readout system, a correction factor for the clock frequency of the analog readout system can be calculated. This method is introduced in more detail in Koke 2017.

In all possible combinations of HICANNs and analog readout systems, a maximum deviation of 1000 Hz from the nominal value of 96 MHz is found. Without correction, this leads to a final shift of 0.01 ms between spike data and membrane recording for a measurement time of 1000 ms, which is already much smaller than the error of the delay measurement. Consequently, using the correction, divergences of the clock frequencies are negligible for the delay calibration.

4.3. Improvements of the Hardware Connectivity

The BrainScaleS-1 system uses a physical modeling approach to emulate neural networks. Therefore, in advance of each experiment, a physical representation has to be found on the hardware that resembles the investigated network, as introduced in section 3.3. Especially large-scale experiments, investigated in this thesis, form a challenging task due to the limited number of resources available on the hardware. To this end, the map

and route algorithms were improved towards large-scale experiments in the course of the master thesis by Felix Passenberg [Passenberg 2019]. In addition, the algorithms were extended for compatibility with the availability database. A short summary of the most important adaptions is given in the first part of this section. Moreover, to ensure implemented routes transmit spike signals as expected, the locking of the repeater circuits was investigated and improved during the master thesis of Jakob Kaiser [Kaiser 2020]. The resulting procedure and performance improvements are demonstrated in the second part of this section.

4.3.1. Improvements of the Map and Route Algorithm

In this section, the most important improvements of the map and route algorithm towards large-scale experiments are summarized, which were mainly implemented during the master thesis of Felix Passenberg [Passenberg 2019]. In the course of this, full support for the availability database was also added to the algorithms so that malfunctioning components are not utilized in experiments.

First, the mapping between model and hardware neurons, the placement algorithm, was extended to allow the user to choose from different strategies. Possible choices are:

- Model neurons are placed on the wafer in descending order of their population's ID on neuron blocks, starting on the block with the least available space. Space limitations are caused by excluded neuron circuits in the availability database. In case of equal sized blocks, a spiral ordering starting at the center of the wafer is applied. This algorithm represents the already existing algorithm before the adjustments, which has two major disadvantages. On the one hand, populations are more likely split on different HICANNs due to the preference of small neuron blocks. Since neurons of the same population often share destinations and only neurons placed on the same HICANN can share buses, the number of required buses increases. On the other hand, using this algorithm, the distribution of neurons primarily depends on the availability data. Therefore, target neurons get split over the whole wafer, which leads to resource intensive long bus connections.

- Model neurons are placed on the wafer in either descending or ascending order of their population's ID on neuron blocks, starting on the top left of the wafer. Consequently, populations and target neurons are no longer split over the whole wafer. Additionally, due to its simplicity, it is more performant compared to the original algorithm and was chosen as the new default. An ascending order is preferred over a descending order due to the more intuitive placement results.

- Populations are clustered by their connectivity. Populations are prioritized according to the number of connections to an already placed population and placed close to it using a spiral order. Although computationally expensive, a lower bus utilization is expected due to the preference of short connection lengths.

- Neurons are clustered by their connectivity. Similar to the previous algorithm but the priority is calculated per neuron instead of per population.

Since it is enforced that all neurons are placed on the hardware, the performance of the algorithms is evaluated by the number of connections that can be realized. The performance of the individual algorithm strongly depends on the respective network topology. More details and performance comparisons for commonly used network structures are shown in Passenberg 2019.

Furthermore, for the networks investigated in this thesis, manual placement requests that restrict individual neurons to target components were used to further improve the performance. The final place and route results of the networks investigated in this thesis are discussed for each network in sections 5.2.2 and 6.2.1.

Once circuits are found for each model neuron, connections are implemented. Due to the limited number of buses available, it is beneficial to merge as many signals on a single bus as possible. This is achieved in the merger tree, introduced in section 3.1.2. However, the number of signals that can be fed into the synapse array is limited by the number of synapses that can be reached from the target synapse driver. Although the signal can be mirrored to neighboring drivers, in the current chip version this is limited to adjacent drivers only. Therefore, the number of target synapses that can be reached by a single bus depends on the number of available drivers and is restricted to at most 3 drivers. In the original implementation of the merger routing, this restriction was ignored and as many signals as possible were merged on a single bus, resulting in increased synapse loss for large networks. To this end, a synapse driver-aware merger tree strategy was implemented that calculates the expected number of synapse drivers needed for a specific merger tree configuration, starting with maximum merging. If it exceeds the number of available drivers for this route, the signals are split using an additional bus. This is repeated until the driver requirement is satisfied. Consequently, for all synapse connections that could not be routed previously, a different bus connection is tested. Synapses excluded from the availability database are not considered during the calculations. However, this should have almost no effect on the performance of the algorithm since excluded synapses normally go along with the exclusion of the related synapse driver and exclusively excluded synapses are rare, cf. table 4.3.

Although, merging as many signals as possible is beneficial in regard to synapse loss, it is also limited by the bandwidth of a single bus system. This is not considered by the algorithm, since the expected spike rate of the model is normally not available at the time of routing. Therefore, detailed bandwidth considerations were done with focus on the networks investigated in this thesis in section 5.2.1.

Additionally, during the overhaul of the algorithms, several problems were detected and corrected that led to wrong hardware configurations. A description of all corrections can be found in Passenberg 2019. Undetected, such errors could lead to wrong network behavior that is nearly impossible to distinguish from, for example, a wrong parametrization. Due to the complexity of the algorithms, it is difficult to identify incorrect hardware assumptions during the routing process. Therefore, tools were developed in the course of this thesis to validate the final routing results. There, the results of the map and route algorithm as well as the final hardware configuration are loaded and checked for all hardware restrictions described in section 3.1. Results that do not fulfill all restrictions are rejected. Consequently, correct routing results are ensured during experiments, in

particular when testing different algorithms.

Finally, full support of the availability database was added to the map and route algorithm. Since the algorithm selects available components from a graph representation of the hardware this is achieved by removing all unavailable components from this graph.

The influence of excluded components on the performance of the map and route algorithm for a feed-forward network is shown in fig. 4.19.

There, a chain of populations is established, where the neurons of each population are exclusively connected to the next population. On the one hand, the simplicity of the network structure allows the algorithm to implement short connections utilizing a minimum number of resources. On the other hand, the dense network structure leaves little to no room for reroutes. Therefore, an approximately linear dependency between removed buses and synapse loss is found.

All in all, besides improving the overall performance, the introduced modifications allow the user to adjust the used algorithms to the investigated network structure to minimize synapse loss. However, with increasing complexity and size of the network structure, synapse loss becomes inevitable using a limited number of hardware resources. To address this in the investigations, not-implemented synapses are extracted from the routing results and are incorporated into the software simulations.

4.3.2. Repeater Re-Locking

During emulations, network connectivity does not only depend on the quality of the routing results, but also on the reliability of the connections themselves. As explained in section 3.1.3, spike signals are regenerated in repeater circuits between HICANNs. Introduced in section 3.1.4, these circuits recover the timing reference necessary to decode and encode the spike addresses from the received signal itself at the beginning of each experiment. This process is called repeater locking, and repeaters that found the correct timing reference are referred to as locked repeaters.

All repeaters except 8 per HICANN that are directly connected to the merger tree have a test data output. It allows for reading out the addresses decoded by the repeater, and by comparison with the sent addresses, it is used to check the locking state of the repeater. Using this, in large-scale experiments, repeaters are found that do not recover the correct timing during the initial locking phase and therefore stay unlocked. These repeaters are not able to correctly decode the signals they receive and therefore forward arbitrary spikes. To this end, a re-locking scheme was developed in the master thesis of Jakob Kaiser to reduce the number of unlocked repeaters, which is summarized in the following. It is discussed in more detail in Kaiser 2020.

The idea of the repeater re-locking is to repeat the locking for unlocked repeaters. Therefore, the test data outputs are used to check the locking state of all measurable repeaters after locking. Subsequently, the locking process is rerun for unlocked repeaters. Since the repeaters are organized in groups, the DLL reset that is necessary to restart the locking process can only be done for a whole repeater block. Consequently, the locking process is rerun for all repeaters of repeater blocks that host at least one unlocked repeater. This process is repeated a predefined number of times or until all repeaters are

Figure 4.19.: Map and route performance with excluded components. Shown is the synapse loss of a feed-forward network placed and routed on a wafer for different states of the availability data. The network consists of 25 populations with 200 neurons each, where all neurons of one population are connected to all neurons of the next population. Neurons are placed according to their population's connectivity. Moreover, synapse driver-aware merger tree routing and neurons consisting of 8 membrane circuits are utilized. Due to the hierarchical structure of the availability data, the exclusion of some components is equivalent to the exclusion of multiple other components. Therefore, for comparison reasons, the test availability data is exclusively generated from randomly excluded bus circuits. The shaded area represents the standard deviations of measurements with 100 randomly generated datasets. The dotted line indicates the synapse loss that is observed using the availability data extracted from a real wafer given in table 4.2 and table 4.4.

locked. The effect of repeater re-locking is demonstrated in fig. 4.20.

Using one re-locking attempt, a significant number of additional repeaters can be locked. Further repetitions seem to have no effect. In addition, different waiting times during the re-locking are tested. There, t_{reset} determines the time between pulling and releasing the reset of the delay locked loop before a new locking attempt and t_{locking} the time between locking and testing of the repeaters. A constant number of locked repeaters is found for all timings.

Due to the necessity to record the decoded addresses, the test is only possible for repeaters that have a test data output. This is not the case for the 8 repeaters per HICANN that are used to inject the signal from the merger tree and the repeater circuits that are implemented in the synapse drivers. Therefore, these circuits are not considered in the test.

Although the reliability of connections increases with re-locking some repeaters remain unlocked. Since they only make up a small part of the whole network, a limited influence on the final network behavior is expected. Therefore, no further efforts were undertaken in the course of this thesis to further reduce their number.

Nevertheless, for future experiments, it would be possible to use the availability database to exclude undesired repeaters. To this end, a test was developed that generates random connections on the wafer and measures the locking state of all repeaters. Since the locking probability of each repeater depends on the quality of the signal it receives from previous repeaters, correlations are expected during the measurement. Therefore, a statistical evaluation of different routes would be necessary to detect repeaters with the highest probability of remaining unlocked. The final result would then be a compromise between available circuits for routing and the probability of finding unlocked repeaters.

Additionally, repeaters with no test data output could be tested by detecting correctly received spikes using membrane potential measurements. However, due to the larger measurement overhead and the limited analog readout possibilities, such measurements are expected to be slow and are therefore not considered during re-locking.

4.4. Hardware Characteristics and Solutions

One of the challenges of operating analog hardware is unintended circuit behavior. The BrainScaleS-1 system, the first implementation of a wafer-scale analog neuromorphic system for emulating large-scale networks of spiking neurons, is not an exception. During the hardware operation done for this thesis, unintended hardware effects were observed that lead to modifications of the network behavior. Since currently no revision of the BrainScaleS-1 system is planned, these effects have to be compensated by suitable hardware operation. In this section, all hardware characteristics and methods used to compensate for them are introduced.

4.4.1. Iterative Calibration

The calibration of the BrainScaleS-1 system is generated to cover the whole parameter range of the circuits. Therefore, each neuron parameter is calibrated by sweeps of its

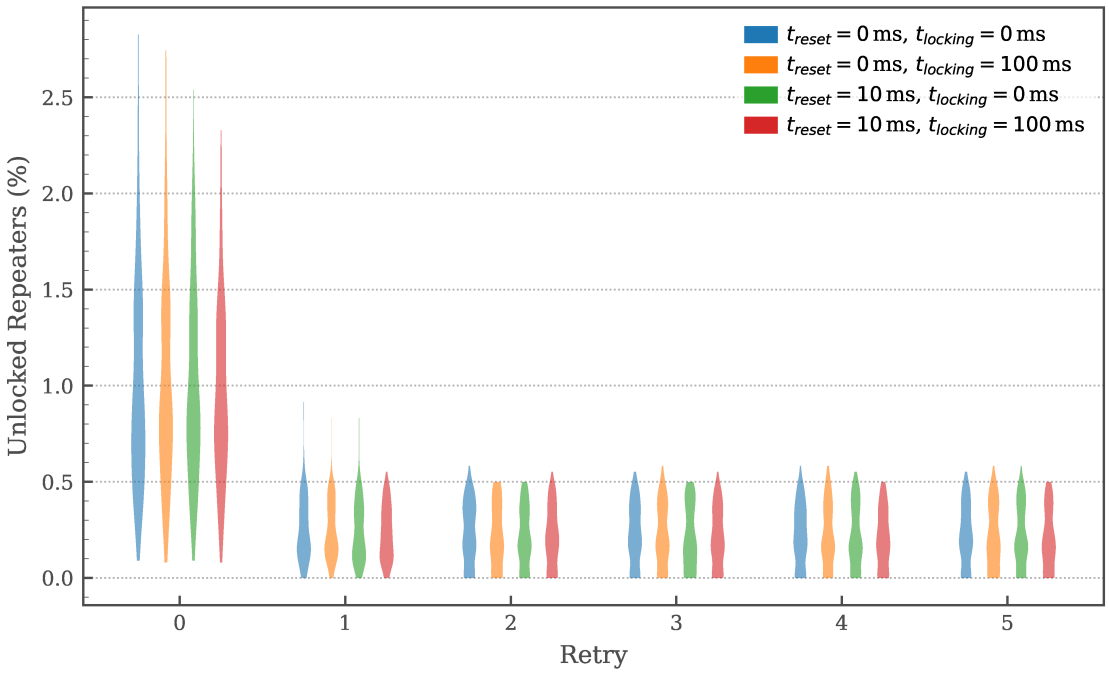


Figure 4.20.: Effect of repeater re-locking on the number of unlocked repeaters. The percentage of unlocked repeaters after various numbers of re-locking retries is shown. Investigated networks are generated by placing a single population on each of 39 randomly selected HICANNs on wafer 33. Populations consist of 120 neurons and connect to all neurons of the next population. In total, the measurements are repeated 50 times for 5 differently placed networks, respectively. Due to the implementation of different routings, the number of utilized repeaters ranges from 1089 to 1348. The percentage of unlocked repeaters drops after the first retry and stays basically the same afterward. Different colors indicate distinct waiting times during the locking that have nearly no effect on the success rate. Taken from Kaiser 2020.

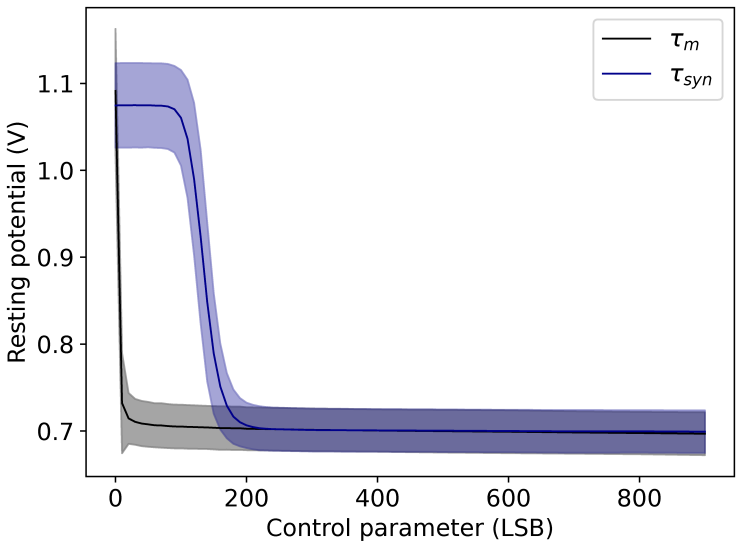


Figure 4.21.: Dependency of the resting potential on the neuron time constants. The actual resting potential of a calibrated neuron configured to $E_{\text{rest}} = 0.7\,\text{V}$ is measured for different hardware settings of the membrane time constant τ_{m} or synaptic time constant τ_{syn}. Only one time constant is modified at a time, while the other one is set to a value above 200 LSB where no effect on the resting potential is expected. The solid line represents the mean value, and the pale area the standard deviation of measurements on all 512 neurons of 12 HICANNs. Below a specific value of their control parameters, the time constants affect the resting potential and therefore invalidate the calibration.

control parameter, while all other parameters are kept fixed at suitable values. Although this approach allows for great flexibility when configuring the hardware, it does not account for dependencies between different neuron parameters. However, such a dependency is observed between the resting potential and the time constants of the neuron. This is demonstrated in fig. 4.21. For small control parameters of the time constants, leakage currents increase the resting potential. This effect is strong enough to change the neuron behavior considerably.

To circumvent this, the range of the control parameters of the time constants was reduced. Too small values are clipped to a minimum value where no leakage currents are expected. Excitatory synaptic time constants that can be realized with the reduced parameter range for each neuron are exemplarily shown for HICANN 0 of wafer 30 in fig. 4.22. Furthermore, the investigated network models were adapted to time constants that correspond to large control parameters. Preferably, small values between 2 ms and 3 ms, which can be realized by most neuron circuits, are chosen.

To prevent further parameter dependencies that are not covered by the calibration, an

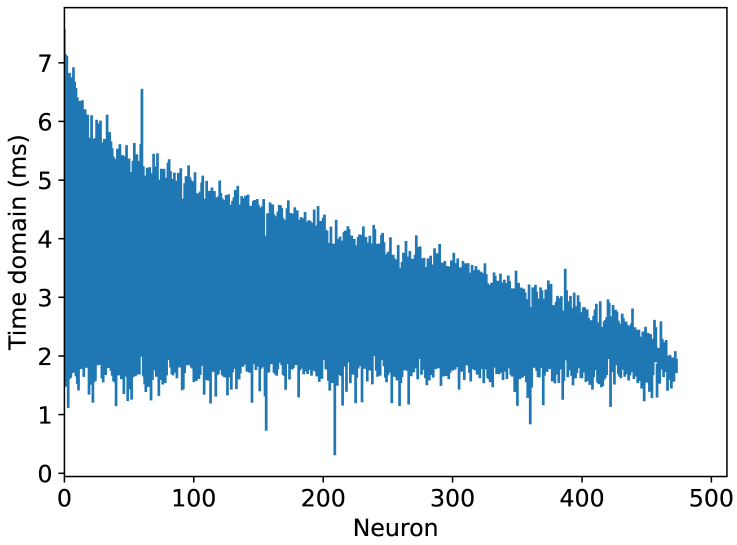


Figure 4.22.: Excitatory synaptic time constants available on HICANN 0 on wafer 30. For each neuron, a vertical line represents the available parameter range obtained for the synaptic time constant during calibration with restricted control parameters. The neurons are arranged in descending order based on the parameter range. 38 neurons that fail calibration are excluded.

experiment-specific calibration was done that uses the same neuron configuration that is finally used during each experiment. This is achieved through an iterative approach where the calibration is executed twice. The first iteration begins with an ideal parameter transformation used to configure the neurons. Subsequently, during the second run, the results of the first iteration are utilized. As a result of this, more accurate model parameters are anticipated during calibration. Additional repetitions are found to have no improving effect. Furthermore, neurons that cannot achieve the desired configuration fail the calibration and are excluded from the availability database.

4.4.2. Finite Resistance Between Membrane Circuits

On the BrainScaleS-1 system, neurons are build of an adjustable number of membrane circuits. Thereby, the number of possible incoming synapses per neuron can be configured to meet the requirements of the experiment, cf. section 3.1.1. The membrane circuits are expected to behave like a single membrane and are therefore short circuited. However, measurements indicate a finite resistance between circuits that is no longer negligible for large neurons. Consequently, stimulations of the first membrane circuit get attenuated until they reach the last circuit, which is demonstrated in fig. 4.23. Moreover, fig. 4.24 shows that stimulations at the edge of the neuron lead to stronger excitations of the stimulated membrane circuit compared to stimulations at the center. Due to the non-zero

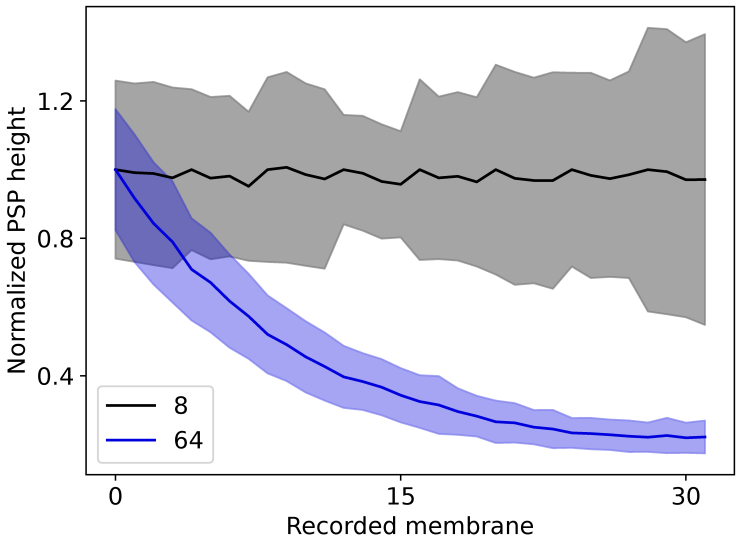


Figure 4.23.: Finite resistance between membrane circuits. Neurons, comprising either 8 or 64 connected membrane circuits, are stimulated at their top-left membrane circuit. Furthermore, in various measurements, the neuron's membrane potential is recorded at different membrane circuits. For each recorded position, the PSP height, resulting from the stimulation, normalized to the mean height measured at the stimulated membrane circuit, is depicted. During the measurement, only circuits located in the top row of the neuron block are recorded. As the same number of circuits is located in the bottom row, the maximum distance to the stimulating neuron is equal to half of the neuron size. Consequently, for neurons of size 8, every fifth membrane circuit is part of a new neuron that is investigated. In contrast, for a neuron size of 64, all recorded membranes belong to a single neuron. The pale area shows the standard deviation of 20 repetitions of each measurement on 12 different HICANNs. Due to their smaller membrane capacity, neurons of size 8 are more sensitive to variations.

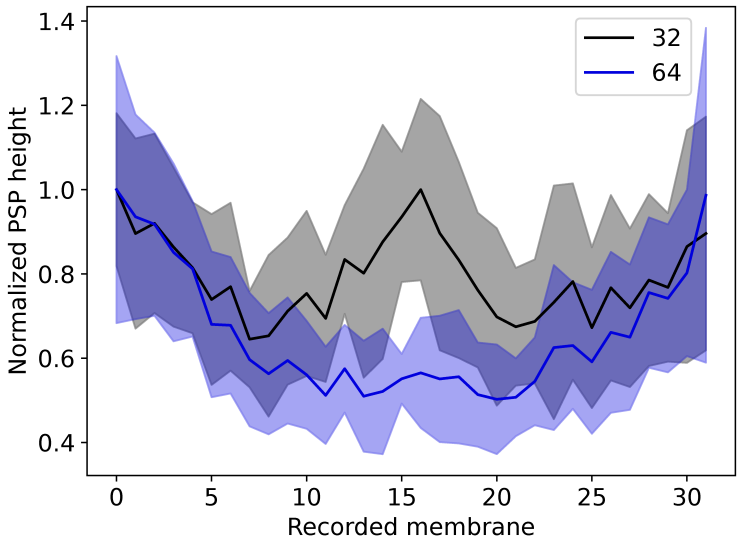


Figure 4.24.: Effects of connected membrane circuits. Neurons, comprising either 32 or 64 connected membrane circuits, are stimulated and recorded on the same membrane circuit. In this manner, the PSP height, resulting from the stimulation, is measured for all membrane circuits located in the top row of each investigated neuron. Subsequently, the obtained heights are normalized to the mean height measured at the top left membrane circuit of each neuron. Since only the top row is considered, the maximum distance between first and last circuit of each neuron is equal to half of the neuron size. Consequently, for neurons of size 32, membrane 16 is part of a new neuron that is investigated. Higher PSP values are observed when the neuron is stimulated at membrane circuits on the edge of the neuron. The pale area shows the standard deviation of 20 repetitions of each measurement on 12 different HICANNs.

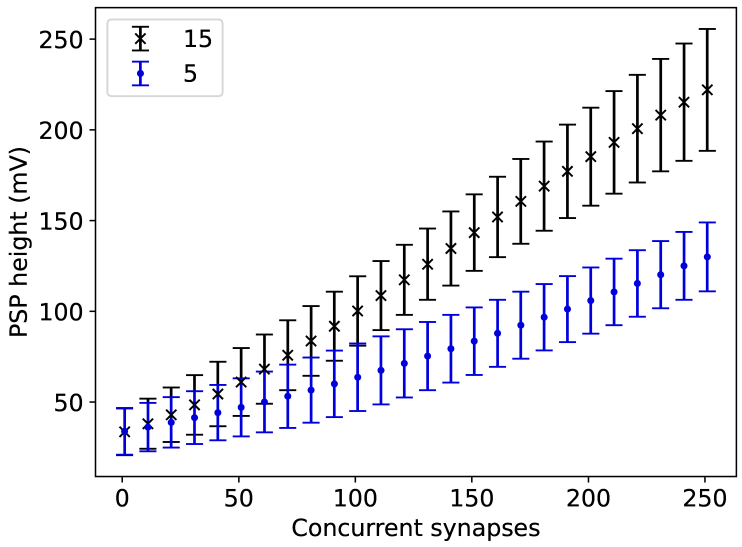


Figure 4.25.: Effect of concurrently spiking synapses. The PSP height of a fixed neuron is measured for varying numbers of concurrently spiking synapses placed in one synapse row. The recorded neuron comprises a single membrane circuit and is consistently stimulated exclusively by a single synapse of the same row. Different colors signify distinct digital weight parameters w for the remaining synapses, although the neuron is consistently stimulated with a fixed weight. Each measurement is repeated 20 times per neuron on 12 HICANNs.

resistance, the charge is not instantly distributed, and in contrast to the center, it can only dissipate in one direction at the edges.

This is a problem since the models investigated in this thesis are based on point neurons. There, an equal distribution of the charge is necessary, since both the stimulation strength and the threshold comparison depend on the momentary membrane potential. To this end, only neuron sizes smaller or equal to 8 are used in this thesis, for which no effects of the membrane potential are observed.

4.4.3. Synaptic Weight Enhancement by Concurrent Spiking

Investigations of biologically inspired neural networks depend on a precise configuration of the synaptic weights. Therefore, the weight calibration introduced in section 4.2.3 is implemented. However, on the BrainScaleS-1 hardware, weights are found to be dependent on the number of concurrently spiking synapses on a single synapse row, demonstrated in fig. 4.25. There, concurrent stimulation is achieved by programming all synapses to respond to the same address. Although the membrane is always stimulated by a single synapse with a fixed weight configuration, the PSP height of the stimulated membrane

increases with more concurrently spiking synapses. The effect is more pronounced if the additional synapses are configured to higher digital weight values.

As introduced in section 3.1.5, each synapse circuit implements a current mirror that copies the current provided by the synapse driver. In case of a spike, using a HICANN clock frequency of 125 MHz, the output current is connected for 8 ns to the synaptic input line of the neuron. Therefore, the observed effects could be explained by a modified gate potential of the current mirror caused by charge injection during the switching process. Since all synapses of one row share the same gate potential, this effect is enhanced if more synapses are activated at the same time. Moreover, the digital weight parameter defines the conductance to the synaptic input line. Consequently, stronger capacitive coupling between gate and synaptic input line is expected for higher weight parameters.

To assess the impact of the effect during experiments, the duration for which a previously spiking synapse affects other synapses on the same row is estimated. At maximum every second clock cycle a signal is transmitted on the bus system, cf. section 3.1.2. Since it is not possible to provide the exact clock cycle a signal is sent, all synapses of one row except one are continuously stimulated by the on-chip background generator with rates close to the maximum frequency, which corresponds to 16 ns between consecutive signals using a HICANN clock frequency of 125 MHz. The maximum frequency cannot be reached since occasional signals are required to keep the repeaters locked, and one additional signal that exclusively stimulates the remaining synapse which excites the measured neuron. It is assumed that each stimulation of the separate synapse happens less than 3 time steps or 36 ns after all other synapses have been stimulated simultaneously. In these measurements, no weight enhancement caused by the additionally spiking synapses is observed.

However, the measurement does not guarantee that the investigated neuron is stimulated in the next possible clock cycle after all other synapses have been stimulated. Therefore, the probability of two spikes sent in 16 ns is simulated using the routing results of the investigated networks. For the cortical microcircuit model, on average, 26 synapses are placed on a single synapse row. Targeting for asynchronous irregular spiking behavior, Poisson distributed spikes with an average rate of 10 Hz are expected. With this, on average, only 6 % of all sent spikes occur in the next possible clock cycle after a previous spike on the same row.

Since the weight enhancement is only significant for many preceding spikes and no weight enhancement is observed after 3 time steps, the effect is negligible for consecutive spikes.

Nevertheless, synapses on the same row that are stimulated by the same address will always spike simultaneously. To minimize resulting deviations, the weight calibration is performed with the expected average number of concurrently spiking synapses during experiments. Thereby, all involved synapses are configured identically since similar weights are expected for all synapses sharing the same row. For the cortical microcircuit model, a maximum of 23 and on average 2.4 synapses on the same row share addresses. Therefore, the anticipated effect is small compared to the deviations of the weight calibration for most synapses. Nevertheless, it depends on the synapse placement and has to be considered for adapted routing.

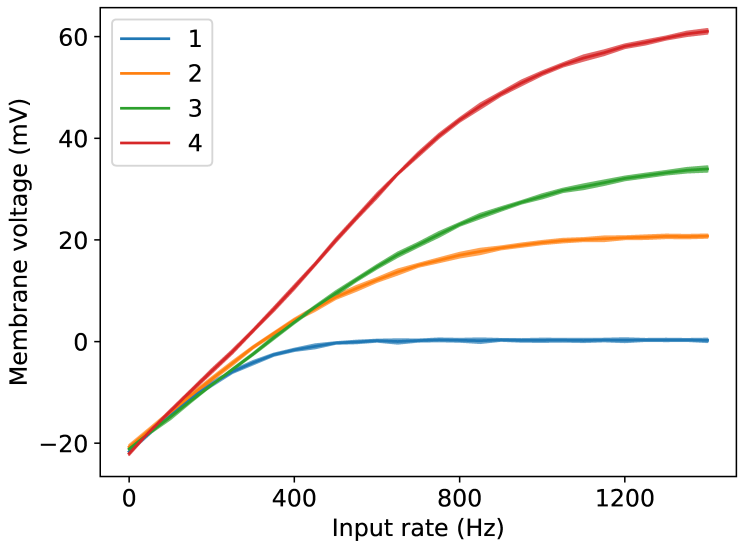


Figure 4.26.: Saturation of the synaptic input. The mean membrane voltage of a single neuron is measured for different rates of a regularly spiking input with maximum weight settings. The neuron consists of 8 membrane circuits and is configured to a resting potential of −20 mV and its threshold value is set high enough that it is never reached. Different colors represent the number of synapse columns that are used to stimulate the neuron. In all measurements, the same spike times are used, which are distributed round-robin on the available synapse columns. The shaded area represents the standard deviation of 20 repetitions.

4.4.4. Synapse Input Saturation

In software simulations, network parameters are only limited by the precision of the data types used. Therefore, no restrictions are expected for high weight values. In contrast, emulations on the BrainScaleS-1 system are restricted to the physical limits of the involved circuits. Consequently, for strong stimulation the input current saturates. Since neurons are built from individual membrane circuits that all implement their own synaptic input circuit, currents from different circuits add up, and a higher maximum stimulation is achieved for larger neuron sizes. This is demonstrated in fig. 4.26. There, the mean membrane potential of a non-spiking but stimulated neuron is used to visualize the stimulation strength. While for higher input rates a linear increase of the recorded membrane voltage would be expected from the LIF model, the saturation of the synaptic input is observed. Using more synapse columns, individual inputs add up and an approximately linear increase of the maximum stimulation is found.

During experiments, there are two possibilities to mitigate this saturation. On the one hand, if possible, a large distance between the reversal potentials is desirable. According to the parameter translation eq. (4.3) this results in a smaller distance between resting

potential and threshold potential. Therefore, less stimulation is required for the neuron to spike. On the other hand, large neurons should be used. Thereby, the number of available synapse columns increases, although the finally used number depends on the routing, specifically on the number of synapses connected to the same target neuron that are stimulated from a single bus. However, due to the effects described in section 4.4.2, the maximum neuron size is limited to eight interconnected membrane circuits.

In biologically plausible networks, where typical spike rates remain below 10 Hz and with the expected connection counts on the system, saturation effects are anticipated to be small. Nevertheless, for higher spike rates, it becomes imperative to consider effects arising from the saturation.

5. The Balanced Random Network Model on BrainScaleS-1

Inspired by the human brain, neuromorphic hardware seeks to address the limitations of conventional computers. Particularly when emulating models of its archetype, i.e., spiking neural networks, its numerous parallel computational units offer the potential to reduce power consumption at accelerated speed. This becomes significant as the complexity of the simulation grows, as seen in large brain models. However, it is precisely these models that are of interest, as it is assumed that the computational power and stability of the human brain reside in its immense network structure with many redundant components. Therefore, the emulation of large-scale biological network models is the ultimate benchmark for existing neuromorphic hardware systems. Furthermore, by physically implementing neuron circuits, their reduced flexibility compared to software simulators provides an opportunity to verify the plausibility and stability of the investigated models.

The focus of this thesis is on the BrainScaleS-1 neuromorphic hardware system, introduced in chapter 3. Already successfully demonstrated its capabilities on small neural networks in Göltz et al. 2021; Kungl et al. 2019; Schmitt et al. 2017, this thesis concludes long-standing efforts with the emulation of large-scale experiments that utilize a significant portion of a single wafer-scale system. To this end, two large-scale biological models are investigated. The first one is the balanced random network, introduced in section 2.4. Subject to this chapter, its network behavior with respect to the limitations of the BrainScaleS-1 hardware is examined, which forms the foundation for the emulation of the cortical microcircuit discussed in chapter 6.

Representing numerous sparsely connected LIF neurons, the idealized balanced random network exceeds the capabilities of the hardware system. Therefore, modifications have to be applied to the model to match the restrictions found on the hardware. Since these changes, in turn, affect the network behavior, a software simulation of the model using the simulator backend NEST [Gewaltig et al. 2007] is performed in parallel to the hardware implementation efforts. On the one hand, this allows for investigating the network's behavior with respect to applied changes. On the other hand, the final emulation can be validated with the obtained simulation results.

This parallel development of model adaptations and hardware implementation is also reflected in the structure of this chapter. Its first section introduces the model modifications conducted on the NEST simulator. Subdivided into the presentation and evaluation of individual changes, the model approaches the capabilities of the hardware. Subsequently, in section 5.2, the implementation of the adapted model on the hardware is presented. Based on the improvements for large-scale experiments, discussed in chapter 4, the network is mapped to the hardware structure and emulated. Finally, the obtained

network behavior is compared to the simulation results.

5.1. Adapting the Model to the Neuromorphic Hardware

When comparing the balanced random network model, introduced in section 2.4, to the BrainScaleS-1 hardware, introduced in chapter 3, several significant differences become apparent. The balanced random network, based on theoretical considerations, makes assumptions that cannot be met by the hardware. Demanding a high number of neurons and synapses, it exceeds the routing capacities of a single BrainScaleS-1 system. Additionally, it implements current-based synapses with a delta peak kernel, whereas the hardware realizes conductance-based synapses with an exponentially decaying kernel. Furthermore, unlike an identical parametrization of all neurons in the model, the configurability of neuron parameters on the BrainScaleS-1 system is limited and subject to variations due to the constraints of physically available analog circuits. This limitation extends to the neuron delay, which does not follow a random distribution but instead depends on the distance between circuits on the wafer. Moreover, despite all efforts, the routing capabilities of the BrainScaleS-1 system are limited and for sufficiently large networks it is not possible to incorporate all synapses into the hardware representation.

Consequently, in parallel to preparing the hardware for large-scale experiments, which is discussed in chapter 4, the model is adapted to match the restrictions of the hardware. This is accomplished through a NEST simulation, which permits incorporating modifications while studying the inevitable changes in the model's behavior. Furthermore, since the operating system of the hardware and the software simulator provide the PyNN API, both implementations are based on the same experiment description and evaluation routines, which allows for comparing and validating the obtained network results.

The simulation and modifications applied to the model were implemented during the bachelor thesis of Quirinus Schwarzenböck [Schwarzenböck 2019] and later extended to provide a more precise representation of the hardware restrictions. The results of these investigations are introduced in this section.

It starts in section 5.1.1 with a simulation, aiming for the closest possible representation of the original network description, which forms the basis for all subsequent considerations. The following section 5.1.2 addresses the reduction of the neuron and synapse numbers to accommodate placement on a single BrainScaleS-1 wafer. Based on this, in section 5.1.3, the transition from current-based to conductance-based synapses is discussed. This leads to section 5.1.4, where the parametrization of the model is adjusted to the hardware values and parameter variations are introduced. Finally, in section 5.1.5, an exact routing model of the hardware is incorporated into the software simulation, including not implemented synapses and delay values obtained from the hardware.

5.1.1. Simulation of the Original Model

In this section, the NEST simulation of the original model, described in Brunel 2000, is investigated to provide a reference for all following network modifications. Based on theoretical considerations, the original network structure is defined by idealized

assumptions of sparsely connected neurons. However, the publication also suggests network parameters that approach these assumptions. They are introduced in section 2.4, with relevant parameters listed in table 2.1.

This network comprises 12 500 LIF neurons with 15 625 000 internal and 12 500 000 external current-based synapses. In order to reduce the computational overhead during the simulation, a simplification is applied, which is possible as the external inputs are not simulated but modeled by Poisson-distributed spiketrains. Therefore, instead of providing C_E external connections to each neuron, a single external source with a firing rate of

$$\nu_\mathrm{ext} = \eta \cdot C_\mathrm{E} \cdot \nu_\mathrm{thr} \tag{5.1}$$

is used per neuron, reducing the external connection count to 12 500. Due to the additive characteristic of the Poisson distribution, this results in the same stimulation. To account for this, in the following studies, the external reference frequency ν_thr is redefined to

$$\nu_\mathrm{thr} = C_\mathrm{E} \cdot \nu'_\mathrm{thr}, \tag{5.2}$$

where ν'_thr represents its original definition, introduced in eq. (2.21).

Moreover, to facilitate the transition to hardware, the synapses are simulated with an exponentially decaying kernel from the very beginning. Implementing a very short synaptic time constant of $\tau_\mathrm{syn} = 0.01\,\mathrm{ms}$, it forms a good approximation of the originally used delta peak kernel. As a consequence of this, the weights of the model, given by the PSP height of a single spike, must be translated into a corresponding current value. Starting from eq. (2.6), the time of the maximum PSP height t_max as a result of a single spike arriving at a membrane in its resting state at time $t_\mathrm{s} = 0\,\mathrm{s}$ is calculated by setting $\mathrm{d}V/\mathrm{d}t = 0$. This yields

$$t_\mathrm{max} = \frac{\tau_\mathrm{syn}\tau_\mathrm{m} \log \frac{\tau_\mathrm{syn}}{\tau_\mathrm{m}}}{\tau_\mathrm{m} - \tau_\mathrm{syn}}. \tag{5.3}$$

Inserting it into eq. (2.6), the maximum height

$$\Delta U = U(t_\mathrm{max}) - E_\mathrm{leak} = \frac{w\tau_\mathrm{syn}}{g_\mathrm{leak}(\tau_\mathrm{m} - \tau_\mathrm{syn})} \left(\frac{\tau_\mathrm{syn}}{\tau_\mathrm{m}}^{\frac{\tau_\mathrm{syn}}{\tau_\mathrm{m}-\tau_\mathrm{syn}}} - \frac{\tau_\mathrm{syn}}{\tau_\mathrm{m}}^{\frac{\tau_\mathrm{m}}{\tau_\mathrm{m}-\tau_\mathrm{syn}}} \right) \tag{5.4}$$

of the PSP is found. Assuming $\tau_\mathrm{syn} \ll \tau_\mathrm{m}$, setting the synapse weight according to

$$w = \frac{\Delta U \cdot C_\mathrm{m}}{\tau_\mathrm{syn}}, \tag{5.5}$$

approximately preserves the PSP height ΔU, which is verified in simulation. For the simulation, the membrane capacitance is set to $C_\mathrm{m} = 0.001\,\mathrm{nF}$. However, since it only affects the time constants and synaptic efficacy, which remain fixed during the simulation, it can be freely adjusted.

Finally, as the software simulation calculates the model behavior for discrete points in time, an appropriate time step has to be found. On the one hand, employing smaller time intervals leads to a more accurate representation of the network's dynamic, with

the specific requirement that each time interval should be significantly shorter than the delay of the spike transmission. On the other hand, increasing the number of time steps results in higher computational costs. Therefore, in experimental trials, a time step of $dt = 0.1\,\text{ms}$ was determined as an appropriate compromise [Schwarzenböck 2019]. This value is implemented in all simulations within this thesis.

Using the presented parametrization and a fixed delay value of 1.5 ms, the original model is simulated for 2 s of biological time. Restricting the evaluation to observables obtained in the last 1000 ms allows the network to settle in the beginning of the experiment. Figure 5.1 shows an overview of the resulting network states in relation to the relative inhibitory weight g and the external input frequency ν_{ext}. Following the methodology outlined in the original publication, the network is evaluated with respect to the neurons' average firing rate, irregularity, and synchrony, which are introduced in section 2.3. Given the neurons' inter-spike intervals approximate a Gaussian distribution, which is demonstrated in Schwarzenböck 2019, their mean coefficient of variation represents a good approximation of the network's irregularity. In addition, the measure of synchrony is determined by the variance divided by the mean of the bin heights of the spike time histogram. As mentioned in section 2.3, its value depends on the chosen bin size and number of considered neurons. To align the results with all following evaluations, 2083 neurons are regarded, which corresponds to the maximum number of neurons in the downscaled model presented in section 5.1.2. In this context, the origin of the chosen neurons is unimportant, as all neurons, regardless of their population, demonstrate uniform behavior due to their shared parametrization and input properties. Moreover, in all analyses, a bin width of 0.2 ms is set, which is short enough to detect individual spike clusters and represents a multiple of the simulation time step to avoid artifacts caused by the discrete nature of the simulation.

The thus observed network behavior is in good agreement with the regimes of the original publication, presented in fig. 2.5. As long as the excitatory stimulation exceeds the inhibitory one in the regime $g < 4$, the neurons' firing behavior is regular and synchronous with a high firing rate. For high external frequencies of $\nu_{\text{ext}} \approx 4 \cdot \nu_{\text{thr}}$ and weak inhibition of $g \approx 0$ the strong Poisson generated external input disturbs the synchronous firing behavior, which results in a reduced synchrony. This also leads to an increased firing rate of approximately 400 Hz. A similar behavior is observed for low external stimulation of $\nu_{\text{ext}} \approx \nu_{\text{thr}}$ with negligible internal inhibition of $g \approx 0$, which is not described in the original publication. It is assumed that due to the weak external stimulation, the neurons start to spike at different points in time and therefore do not synchronize. This leads to more equally distributed stimulation compared to the otherwise accumulated input spikes, which results in higher firing rates. It is found that this state is not stable but varies between differently seeded simulations. Once the neurons synchronize, the obtained behavior is in agreement with observations of higher external firing rates. However, given that the synchronous firing behavior is not preserved once variations of the delay value are taken into account anyway, this unstable state is not further investigated.

For strong inhibitory weights in the regime $g > 4$, the neurons mainly exhibit asynchronous irregular firing behavior, as indicated by the reduced synchrony and increased irregularity. Only for high ($\nu_{\text{ext}} \approx 4 \cdot \nu_{\text{thr}}$) and low ($\nu_{\text{ext}} \approx \nu_{\text{thr}}$) external input rates

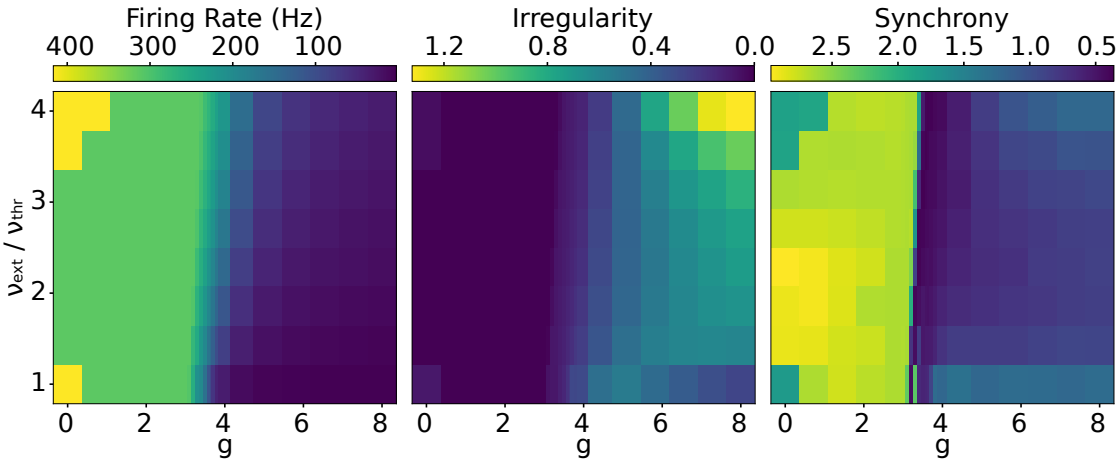


Figure 5.1.: Heat map of the simulated balanced random network behavior with parameters as outlined in Brunel 2000. From left to right, the neurons' mean firing rate, irregularity and synchrony are shown for different relative inhibitory strengths g and external input rates ν_{ext}. The synchrony is determined from the spike time histogram of 2083 neurons using a bin width of 0.2 ms. For $g < 4$, excitation exceeds inhibition and the network exhibits synchronous regular firing with rates of approximately 350 Hz. Slightly higher rates of approximately 400 Hz are obtained for strong external excitation and weak inhibition as the network adopts a less synchronous firing behavior. Furthermore, unstable network behavior is found for weak external excitation without internal inhibition. Depending on the simulator's seed, the network either aligns its firing pattern with its surrounding parametrization or, as depicted, shows less synchronous firing with elevated rates. In contrast, when g exceeds 4, the neurons change into an asynchronous irregular state with low firing rates. The sole exception to this pattern occurs in the presence of either strong external stimuli, resulting in rapid global oscillations, or weak external stimuli, leading to slow global oscillations. Both instances are indicated by heightened synchrony.

synchronous behavior with fast or slow oscillations is found. Therefore, all firing regimes of the original publication are visible, as demonstrated by the spike times and spike histograms shown in fig. 5.2.

Representing the behavior of the original model, the parametrization and results of the simulation form the basis for all subsequent adaptations.

5.1.2. Downscaling the Model

When comparing the size of the software implementation of the balanced random network, introduced in the last section, with the component count of a single BrainScaleS-1 system, it is not immediately apparent that it does not match. In general, the approximately 200×10^3 membrane circuits and 43×10^6 synapses would suffice to host the whole model. This still holds considering that each neuron receives 1250 inputs from sources within the network, necessitating composite neurons. To achieve such high input counts, e.g., 8 membrane circuits can be interconnected. This reduces the number of available neurons by a factor of 8, which would still suffice for the model. However, the network implements an all-to-all connection scheme where pre- and postsynaptic neurons are chosen randomly. Therefore, the number of buses required to transmit the signals increases quadratically with network size, which exceeds the available routing capabilities and leads to a substantial loss of synapses when mapping the original network. This is further intensified by the unavailability of a subset of components that exhibit undesired behavior, as introduced in section 4.1. Consequently, since it is not possible to increase the number of physically available components on the hardware, as can be achieved in software simulations through time multiplexing, the model must be scaled down to accommodate the hardware limitations.

In general, the model does not enforce a specific number of neurons or synapses but is based on the assumption of sparse connections, which might no longer hold for smaller networks. There are two options for downscaling a network: either by decreasing the number of neurons or by reducing the number of synapses per neuron. Both produce distinct effects on the network. On the one hand, by solely reducing the number of neurons, the stimulation and therefore the behavior of individual neurons remains unaffected. The same applies to the global network behavior as long as the network remains sparse, i.e., the connections of each neuron are sampled from a much larger pool of neurons. Once this is no longer the case, neurons share common stimuli and begin to synchronize. On the other hand, reducing the synapse count also decreases the stimuli received by each neuron, which, if left unattended, results in modified neuron and network behavior.

Since the routing profits from a reduction of both quantities, they are decreased simultaneously by the same scaling factor k. As a consequence of this, the ratio of neurons and connections per neuron remains unchanged. Moreover, missing stimulations are compensated by an increased synaptic weight.

In this thesis, the focus is on sustaining the asynchronous irregular firing regime, which, due to its biological plausibility, also represents the neuron behavior of the cortical microcircuit model. As described in Albada et al. 2014, in this regime, the neuron behavior is defined by a below threshold mean membrane potential, where spikes are

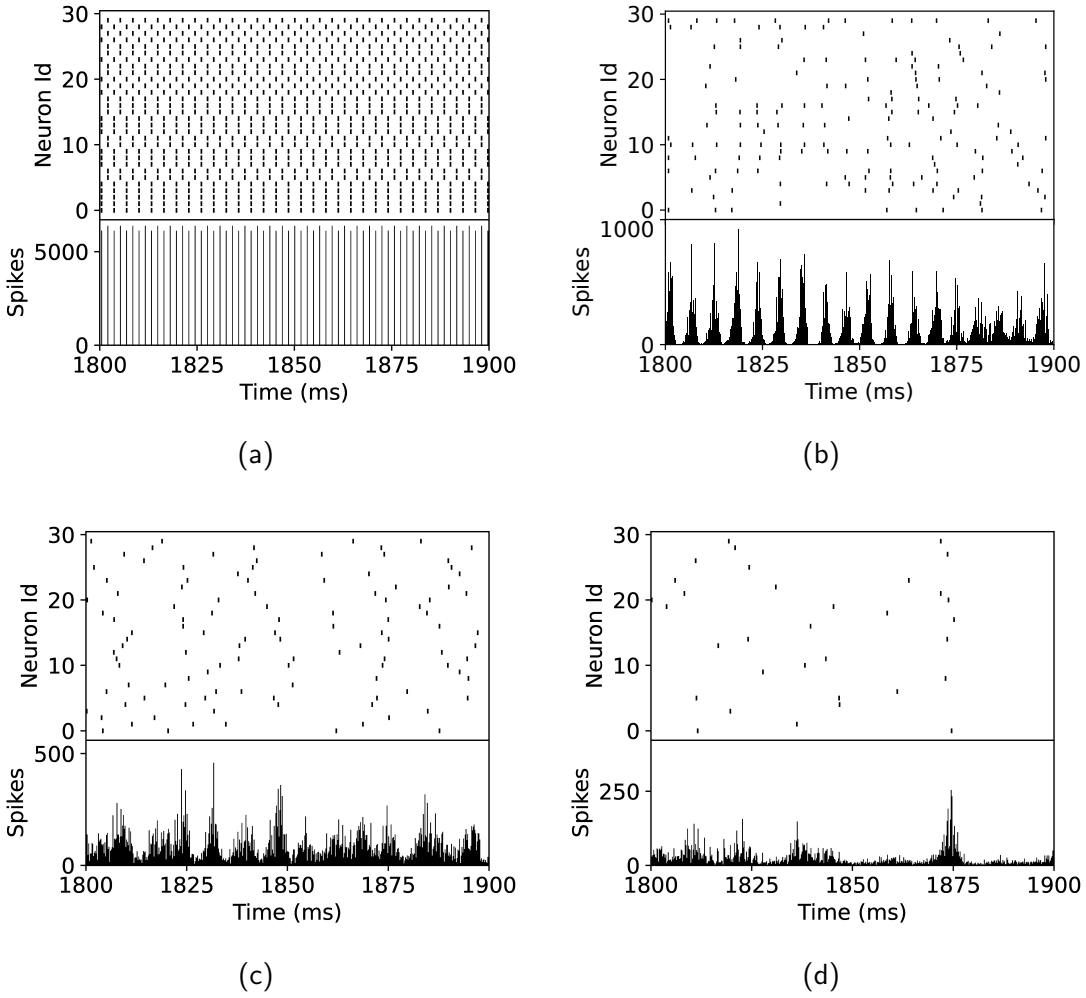


Figure 5.2.: Firing patterns of the simulated balanced random network model with parameters as outlined in Brunel 2000. In the upper part of each figure, the spike times of 30 neurons are shown, indicated by vertical lines. In the lower part, the spike time histogram of all neurons is visualized using a bin size of 0.2 ms. For better visualization, only 100 ms of biological time close to the end of the simulation are illustrated. All figures represent parametrizations similar to those used in fig. 2.6 to portray the various regimes of the network. (a) demonstrates the synchronous regular regime for $g = 3$ and $\nu_{\text{ext}} = 2 \cdot \nu_{\text{thr}}$. The synchronous irregular regime is shown with fast oscillations in (b), found for $g = 6$ and $\nu_{\text{ext}} = 4 \cdot \nu_{\text{thr}}$, and with slow oscillations in (d), obtained for $g = 4.5$ and $\nu_{\text{ext}} = \nu_{\text{thr}}$. The asynchronous irregular regime is demonstrated for $g = 5$ and $\nu_{\text{ext}} = 2 \cdot \nu_{\text{thr}}$ in (c).

Table 5.1.: Parametrization of the downscaled balanced random network model for a scaling factor k. Values are given in comparison to the full-scale model introduced in table 2.1, where $N = N_\mathrm{E} + N_\mathrm{I}$ is the sum of excitatory and inhibitory neurons. The change of the external input frequency is also reflected in the external reference frequency ν_thr due to its inverse proportionality to the synaptic weight (cf. eq. (2.21)).

	Full-scale	Downscaled
Neuron number	N	N/k
Internal inputs per neuron	C	C/k
Total internal connections	NC	NC/k^2
External frequency	ν_ext	ν_ext/k
Synaptic weight	w	kw

driven by fluctuations arising from dominating inhibitory stimulation. This suggests two possibilities to perform weight adjustments to compensate for a reduced synapse number. On the one hand, the weight can be upscaled with the square root of the scaling factor used to downscale the synapses. Since the variations of independent inputs are quadratically dependent on their synaptic weight, this method targets to preserve the network fluctuations. At the same time, an external constant current can be applied to replace the missing mean stimulation and elevate the membrane potential to its original state. On the other hand, the synaptic weight can be increased linearly with the scaling factor. Maintaining the mean input of the neuron, the fluctuations increase. To compensate for this, the external Poisson generated input can be substituted by a constant current. However, this technique is limited to the total amount of variation added by the original external input of the model.

For the scaling factors of $k \geq 6$, utilized in this thesis, both techniques fail to preserve the original network behavior, as demonstrated in Albada et al. 2014. However, since replacing the extensive external input is desired for the cortical microcircuit model and as it preserves the mean firing rates of the neurons, linear upscaling of the synaptic weight is performed in this thesis. Nevertheless, in the case of the balanced random network model, the substitution of external inputs is deliberately avoided to demonstrate the capabilities of the hardware to incorporate external spikes, which is possible due to the reduced routing complexity compared to the cortical microcircuit model.

The thus found parametrization of the downscaled balanced random network model is presented in table 5.1. Moreover, in repeated attempts to map the network to the hardware, a scaling factor of $k = 6$ has been determined as a favorable compromise between lost synapses and network size. This scaling factor was ultimately applied to the model. The resulting network behavior of the downscaled model is demonstrated in fig. 5.3.

In agreement with the original model, the network is split in a synchronous regular regime for relative inhibitory weights of $g < 4$ and an irregular asynchronous regime

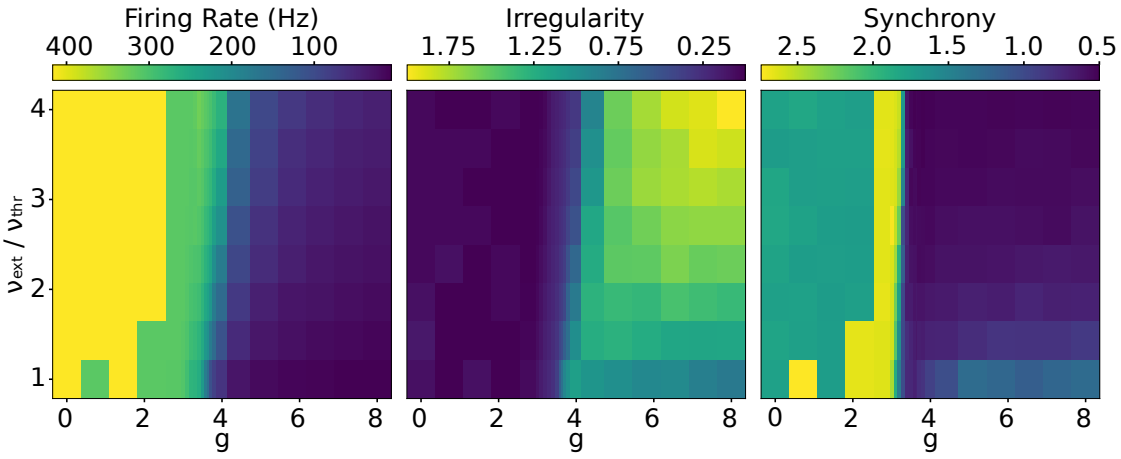


Figure 5.3.: Heat map of the downscaled balanced random network behavior with parameters as outlined in Brunel 2000. The visualization and synchrony parametrization correspond to the full-scale network (fig. 5.1). The network is still split in a synchronous regular regime for $g < 4$ and an asynchronous irregular regime for $g > 4$. For low internal inhibition ($g \leq 2$), additional neuron clusters are formed, leading to an elevated firing rate with reduced synchrony. Compared to the original model, the irregularity is increased and no fast oscillating regime is observed.

for $g > 4$. Moreover, observed firing rates are approximately preserved. However, the necessity to increase the synaptic weight leads to elevated neuron variations, indicated by higher irregularity values. Another consequence of this is the enlargement of the high firing rate region, with rates of 400 Hz, in the synchronous regular regime. Caused by the reduced number of neurons necessary to elicit spikes, the formation of additional clusters of concurrently spiking neurons is facilitated, as indicated in fig. A.6. This limits the observed synchrony and results in higher firing rates than observed in the original model for small values of g and ν_{ext}. Since more concurrent spikes are necessary for inhibited neurons, the formation of additional clusters is hindered for stronger inhibitory synapses in the range of $2.5 \ll g \ll 4$. Furthermore, the highest variations are obtained in the asynchronous irregular regime. There, the increased synaptic weight further intensifies the fluctuations caused by the already strong inhibitory connections. Therefore, compared to the original model, no fast oscillating global neuron behavior is observed for high external input rates.

5.1.3. Transition From Current-Based to Conductance-Based Synapses

The next step in aligning the balanced random network with the hardware is to adjust the synapse model. This means a transition from current-based to conductance-based synapses. Both models and their distinct features are introduced section 2.2.

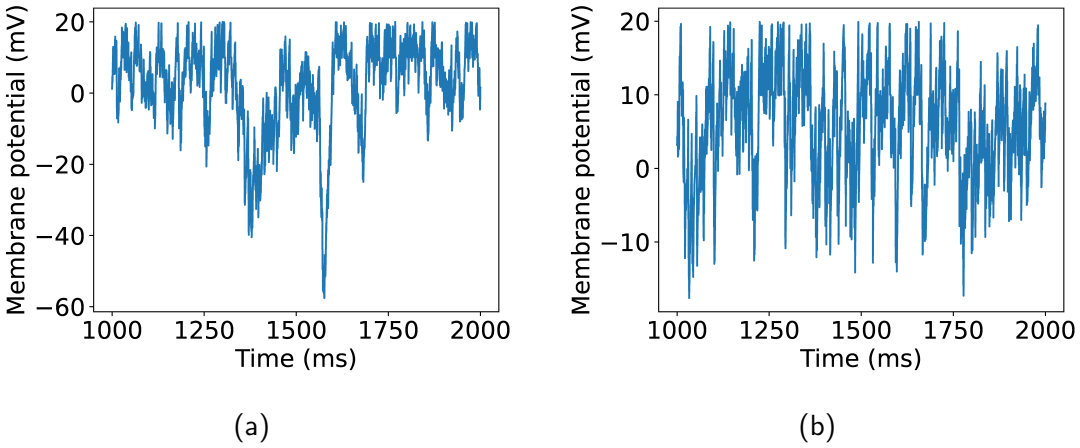


Figure 5.4.: Comparison of membrane fluctuations. The membrane recording of the same neuron within the simulated downscaled balanced random network model is displayed using current-based synapses in (a) and conductance based synapses in (b). Both figures show 1 s of biological time, obtained in the asynchronous irregular regime ($g = 6$, $\nu_{\text{ext}} = 2 \cdot \nu_{\text{thr}}$).

The differences between the two models become apparent when the conductance-based synapses enter the high-conductance state. There, due to the modification of the membrane time constant, the neurons' membranes react much faster to input currents. Moreover, stimulations no longer add up linearly but are attenuated as they approach the reversal potentials. Both effects lead to a reduction of the amplitude of membrane fluctuations, which is demonstrated by recording a single neuron's membrane in the asynchronous irregular regime, depicted in fig. 5.4.

As a result of this, the intrinsic neuron behavior differs. In current-based synapses, there is a balance between excitatory and inhibitory inputs, resulting in a mean membrane potential below threshold. Spiking occurs due to significant fluctuations in the membrane potential. In contrast, with conductance-based synapses, these fluctuations are reduced. Therefore, to preserve the model's connectivity, membrane potentials must remain close to the threshold to trigger spikes. This is discussed in detail in Sanzeni et al. 2022. However, selecting an appropriate translation, it is still possible to achieve comparable network statistics in terms of mean firing rates, irregularity, and synchrony, as demonstrated in the following.

In this thesis, the synaptic weights of the conductance-based synapses are chosen such that the same PSP height is obtained for a single spike in the absence of the high-conductance state, starting from the mean membrane potential $\langle U \rangle$ of the neurons. This is achieved by translating the weight according to

$$w_{\text{cond}} = \frac{w_{\text{curr}}}{E_{\text{rev}} - \langle U \rangle}, \tag{5.6}$$

where w_{curr} is the synaptic weight in the current-based model and E_{rev} is the excitatory or inhibitory reversal potential, dependent of the considered synapse type. Since the model only holds true when the mean membrane potential falls between the two reversal potentials, this, as desired, results in a positive conductance for both synapse types. For the balanced random network, the resting potential is observed to be a suitable approximation for the mean membrane potential of the neurons.

In general, this weight transition already suffices to preserve the characteristics of the network behavior, as the model is expected to be robust with respect to small changes of the absolute weight value. This is already evident in the original model, which does not enforce specific weight settings but is built on the assumption that many spikes are required for the membrane potential to reach the threshold. There, the impact of changing to conductance-based synapses is much smaller compared to the already implemented downscaling. Furthermore, the possibility of creating an imbalance between inhibitory and excitatory weights does not render beneficial, given that all neurons share identical parameters and the relationship between these two weights is already under investigation by altering the parameter g.

However, in contrast to the weight configuration, the selection of appropriate reversal potentials, which are newly introduced into the model, is crucial. This is because the distance between them dictates the strength of the high-conductance state's effects. This becomes obvious when considering an infinite separation. In such a scenario, synaptic conductance diminishes, and as per eq. (2.10), the membrane time constant remains unaltered. Additionally, a reduction of the distance between the momentary membrane potential and the reversal potentials is negligible under these conditions. Consequently, the neurons behave identically to the current-based version, which is confirmed through simulations. In contrast, when selecting reversal potentials close to the membrane potential, the effects of the high-conductance state start to dominate. In the extreme case, no spikes are elicited since the network's stimulation is insufficient to reach the threshold value. This happens because even minor changes from the resting potential, for which the weight is calculated, strongly reduce the efficacy of the synapses.

As a result of this, distant reversal potentials are desirable. However, they are biologically implausible and challenging to implement on hardware. As introduced in section 4.2.1, neuron parameters are translated into the hardware domain. To this end, the maximum and minimum achievable voltages represent the reversal potentials and the remaining parameters are set to preserve relative distances. Therefore, selecting large reversal potentials, the membrane's dynamic is limited to a few millivolts and approaches the noise level of the circuits. Moreover, according to eq. (5.6), the minimum representable weight on the hardware further limits possible choices.

In agreement with the restrictions imposed by the hardware, reversal potentials of $E_{\text{rev}} = \pm 140\,\text{mV}$ are chosen. Symmetrical with respect to the resting potential, no bias is added to the efficacy of excitatory or inhibitory synapses. This also applies to the asymmetry, which is in general added by the threshold mechanic. Exclusively constraining the membrane dynamic towards the excitatory reversal potential, inhibitory stimulations are diminished for lower membrane potentials. Based on the observed voltage fluctuation, depicted in fig. 5.4b, it becomes evident that this effect is negligible for the chosen

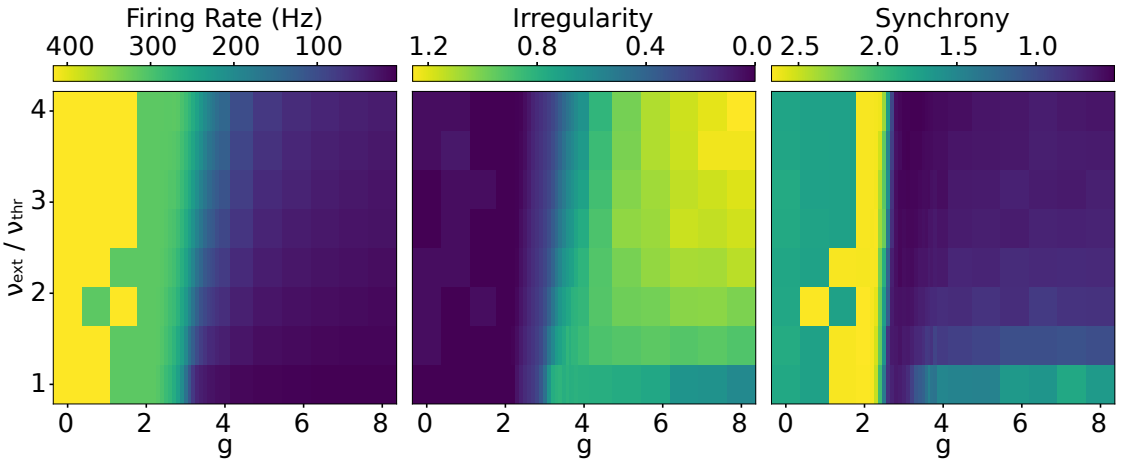


Figure 5.5.: Heat map of the downscaled balanced random network behavior with conductance-based synapses. The visualization and synchrony parametrization correspond to the full-scale network (fig. 5.1). The same regimes are observed as for the model with current-based synapses, depicted in fig. 5.3. However, the transition from the synchronous regular to the asynchronous irregular regime is shifted to $g \approx 3$ and the irregularity is reduced.

reversal potentials. Another option to preserve the original behavior is to alter the neurons' time constants. This is natural because the high-conductance state decreases the membrane time constant. Nevertheless, there are some constraints to consider. On the one hand, the synaptic time constant in the original model is already significantly smaller than the membrane time constant. Consequently, there is only a limited scope for improvement. On the other hand, as discussed in section 4.4.1, the achievable time constants on the hardware are strongly restricted, preventing any further modifications. Thus, the adjustments are limited to the presented weight translation.

By simulating this parametrization, the network characteristics, visualized in fig. 5.5, are obtained for the downscaled conductance-based balanced random network. Compared to the model with current-based synapses, the same network regimes with similar firing statistics are obtained, with two noticeable differences. The overall network behavior is less irregular and the transition from the synchronous regular to the asynchronous irregular regime is shifted to less dominant inhibitory weights. Both effects are caused by the changed underlying neuron behavior with reduced membrane fluctuations.

Therefore, the original neuron behavior is not preserved. However, focused on first- and second-order statistics, similar results are obtained, which still represent a biologically plausible benchmark for neuromorphic hardware.

5.1.4. Introducing Hardware Parameters With Variations

In contrast to software simulators, the configurability of the BrainScaleS-1 system, which is based on physically implemented circuits, is limited. Moreover, being subject to manufacturing induced device mismatch, parameter variations are introduced that are not considered in the original model. Therefore, despite extensive optimization efforts towards biologically plausible large-scale networks, which are discussed in chapter 4, it is not possible to preserve the idealized neuron parametrization of the original model, listed in table 2.1. To this end, in the following, the parametrization of the model is adapted to match possible configurations on the hardware, and the resulting network behavior is studied in simulation.

Variations

The model behavior with respect to parameter variations is investigated in Schwarzenböck 2019. There, the original parametrization is used and Gaussian distributed parameters are considered. While the final adapted model implements distributions with standard deviations obtained from the hardware calibration, for this consideration, standard deviations of 10 % from the respective mean value of all voltage parameters and 20 % of all time constants are assumed. This reflects the less precise calibration of the time constants, explained in Schmidt et al. 2023. Moreover, a Gaussian distributed delay ($\sigma_{\mathrm{D}} = 20\,\%\overline{D}$) and membrane capacitance ($\sigma_{\mathrm{C_m}} = 5\,\%\overline{C_{\mathrm{m}}}$) are considered. Since the membrane capacitance does not directly affect the neuron's dynamics, its variation represents modifications of the synaptic weights. In the final model, this variation is absorbed into the parameter distribution of the weights and the capacitance is considered fixed at $C_{\mathrm{m}} = 1\,\mathrm{nF}$.

The effects on the network behavior of individually modified parameters are depicted in fig. 5.6a. The results are split into synchrony and irregularity values obtained in the four network regimes. As observed, distributed parameters have a minor impact on the irregularity. Only in the synchronous irregular (SI) state the irregularity is reduced, mainly caused by adaptations of the delay and the threshold voltage. The most significant modifications are detected when all variations are applied simultaneously. In this case, the reduction of the irregularity surpasses all individual effects, even though the measurement is conducted with fixed delays.

Similarly, only in the synchronous irregular state is the synchrony of the network affected by parameter variations. However, the delay constitutes an exception. As expected from the original studies, introduced in section 2.4.2, distributed delay values prevent clustering of neurons in the synchronous regular (SR) regime. Therefore, asynchronous firing behavior is obtained in this regime. This is visualized in detail in fig. 5.6b, which depicts the synchrony of the network for different standard deviations of the delay distribution. For values larger than 10 % of the mean delay neuron clustering is disturbed. This also results in unstable network behavior, indicated by elevated error bars, as cluster formation depends on the network's current state.

All in all, despite distributed neuron parameters, the network statistic is still split

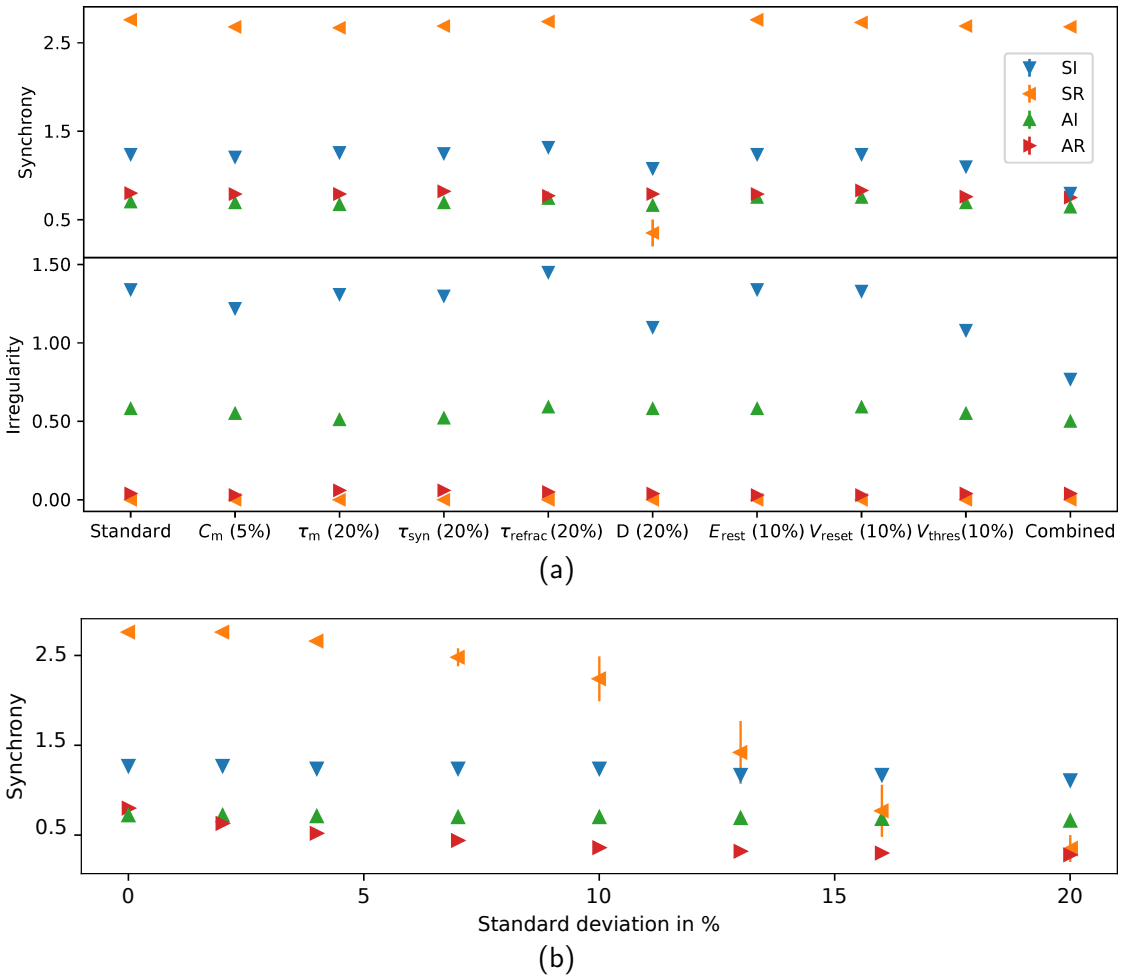


Figure 5.6.: Effect of parameter variations for the full-scale balanced random network model with current-based synapses. Each color represents a distinct regime in the original model (cf. fig. 2.5) from which data is collected using the following parametrization: SI ($D = 1.5\,\text{ms}$, $g = 8$, $\nu_{\text{ext}} = 4\nu_{\text{thr}}$), SR ($D = 1.5\,\text{ms}$, $g = 2.2$, $\nu_{\text{ext}} = 2.2\nu_{\text{thr}}$), AI ($D = 1.5\,\text{ms}$, $g = 6.5$, $\nu_{\text{ext}} = 1.9\nu_{\text{thr}}$), and AR ($D = 2\,\text{ms}$, $g = 1$, $\nu_{\text{ext}} = 3.5\nu_{\text{thr}}$). Error bars indicate the standard deviation of differently seeded simulations. Variations are modeled by Gaussian distributed neuron parameters with mean values matching those of the original model. In (a), the firing statistics of the network are investigated when variations are applied exclusively to individual neuron parameters with standard deviations in relation to their mean value given in parentheses. For comparison reasons, the column labeled "Standard" displays the network statistics without variations. The column labeled "Combined" presents results obtained with all variations, except the delay value D, simultaneously applied to the network. The influence of varying standard deviations of the delay value with all other parameters fixed is separately illustrated in fig. 5.6b. Adapted from Schwarzenböck 2019.

in different regimes with distinct characteristics. However, as described in the original publication, the synchronous regular regime is not preserved and is replaced by an asynchronous regular regime.

Hardware Parametrization

Transitioning the model to the BrainScaleS-1 hardware involves not only the introduction of distributed parameters but also shifts of their mean values to align with hardware constraints. Excluded from these adaptations are all voltage related neuron parameters. Stored in the floating gate cells of the chips, a linear voltage range between approximately 0.45 V and 1.1 V (1.3 V for the excitatory reversal potential) is provided, as demonstrated in section 4.2.2. Although representing a very different parameter range, this discrepancy is handled by automated parameter translations, introduced in section 4.2.1. In general, this enables the realization of arbitrary network parametrizations with biologically plausible settings, meaning that all voltages fall within the excitatory and inhibitory reversal potentials. This is only limited by the measurement precision of the sub-threshold membrane dynamic and the synaptic input currents that can be implemented on the hardware. These aspects are discussed during the selection of the reversal potentials in section 5.1.3. Mapping the introduced reversal potentials to the maximum and minimum hardware voltages, the noise of the membrane is negligible compared to the resulting distance between resting and threshold potentials. Moreover, the synaptic efficacy suffices to represent all network states. Therefore, it is possible to preserve the model's original voltage values on the hardware.

In contrast to this, the time constants of the model must be adapted. On the one hand, the delta peak kernels of the originally used synapses are resembled by biologically implausible very short synaptic time constants, which represent a challenging parametrization. On the other hand, the hardware provides only very limited parameters for the time constants, demonstrated in section 4.4.1. Therefore, the excitatory and inhibitory synaptic time constants are prolonged from $\tau_{\text{syn}} = 0.01\,\text{ms}$ to $\tau_{\text{syn}} = 3\,\text{ms}$. Furthermore, the membrane time constant is reduced from $\tau_{\text{m}} = 20\,\text{ms}$ to $\tau_{\text{m}} = 10\,\text{ms}$. Only the refractory period is preserved at $\tau_{\text{refrac}} = 2\,\text{ms}$. It is worth mentioning that, due to the acceleration factor on the hardware, all membrane dynamics run 10 000 times faster than the presented values but are translated into the biological regime.

Considering that the weight calculation, introduced in eq. (5.5), is only valid for $\tau_{\text{syn}} \ll \tau_{\text{m}}$, a correction factor of $f_{\text{corr}} = 1.7$ is introduced, which has been determined in NEST simulations. As illustrated in fig. 5.7, although the time course of a PSP is altered, its amplitude remains preserved. The effect of modified synaptic time constants with preserved PSP heights is investigated in fig. 5.8. Caused by the different temporal behavior of the neurons' membranes as a response to an input spike, synchronous network behavior is disturbed. This is mainly observed in the synchronous regular regime with clustered neurons. There, the formation of clusters depends on simultaneous stimulation from various sources. However, with prolonged synaptic time constants, the charge is no longer immediately applied to the membranes but accumulates over time, diminishing the impact of concurrent spikes. Furthermore, the irregularity in the synchronous irregular

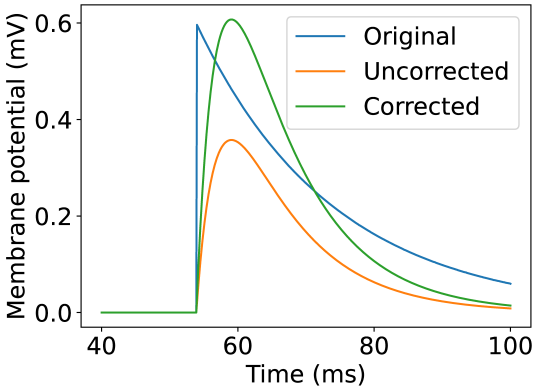


Figure 5.7.: Comparison of PSP height for modified time constants, with and without correction. According to the increased weight of the downscaled balanced random network model, the targeted PSP height corresponds to 0.6 mV. The curve labeled "Original" shows the PSP shape of the originally used delta peak kernel, which is approximated by a synaptic time constant of $\tau_{\text{syn}} = 0.01$ ms. The two remaining curves depict the PSP shape of a neuron with a synaptic time constant of $\tau_{\text{syn}} = 3$ ms and a reduced membrane time constant of $\tau_{\text{m}} = 10$ ms that is either corrected for a weight factor of 1.7 or uncorrected. All simulations are performed using conductance-based synapses.

regime is reduced due to the distribution of stimulations over time. Overall, similar network effects are observed for prolonged synaptic time constants and distributed parameters.

Resulting Model Behavior

Finally, a model is established that comprises adapted and distributed parameters. To closely resemble the behavior expected on the hardware, these parameters are chosen according to the mean values and standard deviations obtained in a network specific calibration. Utilized parameters and variations are listed in table 5.2. As evident, the variations of the parameters differ. In general, more precise calibration results are achieved for voltage parameters because they can be directly read out via the membrane potential using suitable neuron configurations. The only exception to this is the reset potential and the excitatory reversal potential. The former is limited by its configurability, which only allows for setting a common value for all 64 neuron circuits of each neuron block. Therefore, no precise per-neuron calibration is possible. The latter, as introduced in section 4.2.2, requires an indirect measurement method due to the observed non-linear behavior of the membrane potential when approaching the reversal potential. As a result of this, both parameters are modeled by broader distributions.

In contrast to the voltages, the time constants show larger deviations. On the one

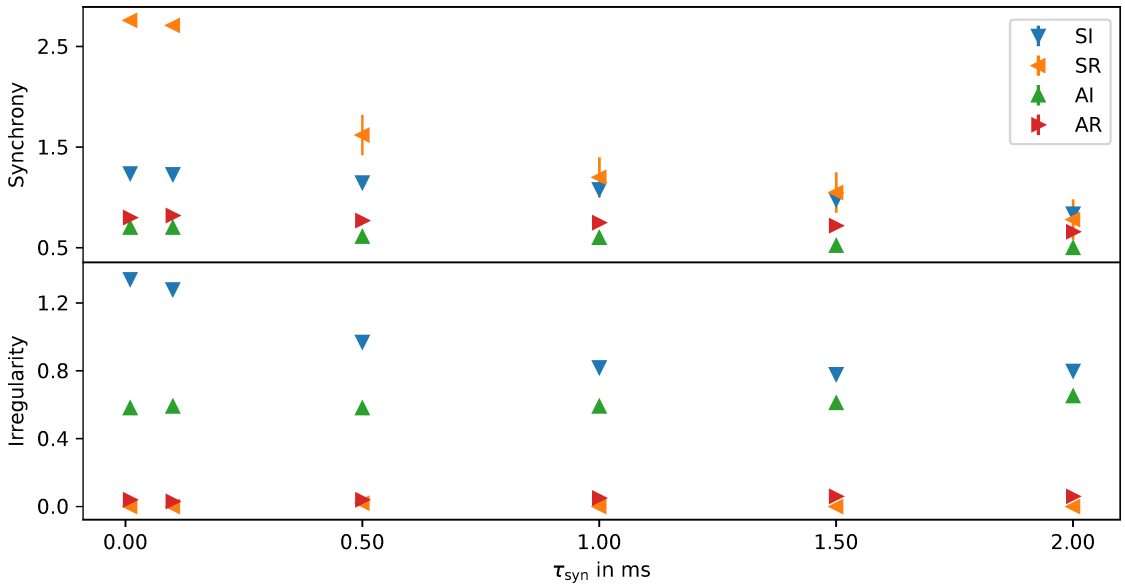


Figure 5.8.: Effect of prolonged synaptic time constants on the firing statistic of the full-scale balanced random network model with current-based synapses. The synaptic weight is modified such that the PSP height of an isolated spike is identical in all measurements. The same parametrizations used in fig. 5.6 are depicted. Error bars indicate the standard deviation of differently seeded simulations. Adapted from Schwarzenböck 2019.

Table 5.2.: Neuron parametrization of the hardware representation of the downscaled balanced random network with conductance-based synapses. Defined by Gaussian distributions, the mean and standard deviation are presented. The distributions' ranges are provided in the last column. The mean and standard deviation of the inhibitory weight are scaled according to gw_{E}.

Parameter	Mean	±	Std		Range
τ_{m}	10	±	2.0	ms	$[0\,,\infty)$
τ_{refrac}	2	±	1.0	ms	$[0\,,\infty)$
$\tau^{\mathrm{e}}_{\mathrm{syn}}$	3	±	0.4	ms	$[0\,,\infty)$
$\tau^{\mathrm{i}}_{\mathrm{syn}}$	3	±	0.2	ms	$[0\,,\infty)$
V_{thres}	20	±	1.12	mV	$(-\infty\,,\infty)$
V_{reset}	10	±	4.2	mV	$(-\infty\,,0.9V_{\mathrm{thres}}]$
E_{rest}	0	±	1.96	mV	$(-\infty\,,\infty)$
$E^{\mathrm{e}}_{\mathrm{rev}}$	140	±	14.0	mV	$(-\infty\,,\infty)$
$E^{\mathrm{i}}_{\mathrm{rev}}$	−140	±	1.96	mV	$(-\infty\,,\infty)$
w_{E}	2.4	±	1.2	nS	$[0\,,\infty)$
C_{m}	1	±	0	nF	$[0\,,\infty)$

hand, they are calibrated via appropriate fits of the neuron's membrane trace in response to stimulation. Analog to the calibration of the excitatory reversal potential, this leads to an increased complexity with dependencies on several neuron parameters. On the other hand, parameters stored in the floating gates are either provided as voltages or currents. Consequently, to control time dependent behavior, these parameters have to be translated via appropriate circuitry. These translations follow non-linear functions, which further deteriorate the calibration precision. For more details on the utilized calibration methodologies it is referred to Schmidt et al. 2023.

The synaptic weights and the membrane capacitance are treated differently. As introduced in section 4.2.3 the absolute value of the membrane capacitance has no effect on the neuron's behavior and its uncertainties are already considered in the distributions of the membrane time constant and the calibrated weight to membrane capacitance ratio $\frac{w}{C_\mathrm{m}}$. Therefore, no additional deviations are assumed, and for simplicity, it is set to $C_\mathrm{m} = 1\,\mathrm{nF}$. In case of the synaptic weight, the large parameter space in combination with the limited analog readout capabilities of the BrainScaleS-1 system does not allow for a per-circuit calibration. Furthermore, the limited configurability of its control parameters adds additional uncertainty to the configured values. Consequently, no exact parameter distribution is obtained and based on the results from section 4.2.3 and section 4.2.4, a standard deviation of 50 % of the respective mean weight value is approximated.

Using this parametrization, the network behavior is simulated for the downscaled model with conductance-based synapses. This simulation does not yet incorporate network properties extracted from the hardware's map and route results but includes all connections with a generic delay distribution, which follows a Gaussian distribution with a mean of 1.5 mV and a standard deviation of 0.2 mV, with a minimum value of 0.3 mV. The results are visualized in fig. 5.9. As a consequence of the distributed parameters and the prolonged synaptic time constant, synchronous states are replaced by asynchronous firing. Nevertheless, the network still exhibits distinct firing regimes with behavior similar to the original model. For small inhibitory weights at $g < 3$, elevated firing rates with regular neuron behavior are observed. Due to the distributed refractory times, with values approaching 0 ms, the mean firing rate is no longer limited to a maximum of 500 Hz. Furthermore, the loss of synchrony results in a distributed stimulation of the neurons, leading to higher firing activity for weak inhibition. The disappearance of clustered neuron firing also replaces the sharp transition between the regimes at $g \approx 3$ with a gradual one since increasing inhibition reduces the mean neuron stimulation. As expected from the prolonged synaptic time constants, the overall irregularity is diminished. However, the asynchronous irregular regime for $g > 3$ is preserved, and the synchronization caused by global oscillations for small external inputs and strong inhibitory connections is still present, although less pronounced.

5.1.5. Incorporating Map and Route Results

As a final modification, the model is adapted with exact routing data obtained from the hardware. To this end, the results of the map and route step, discussed in section 5.2.2, are evaluated and all established connections are extracted and loaded into the simulation.

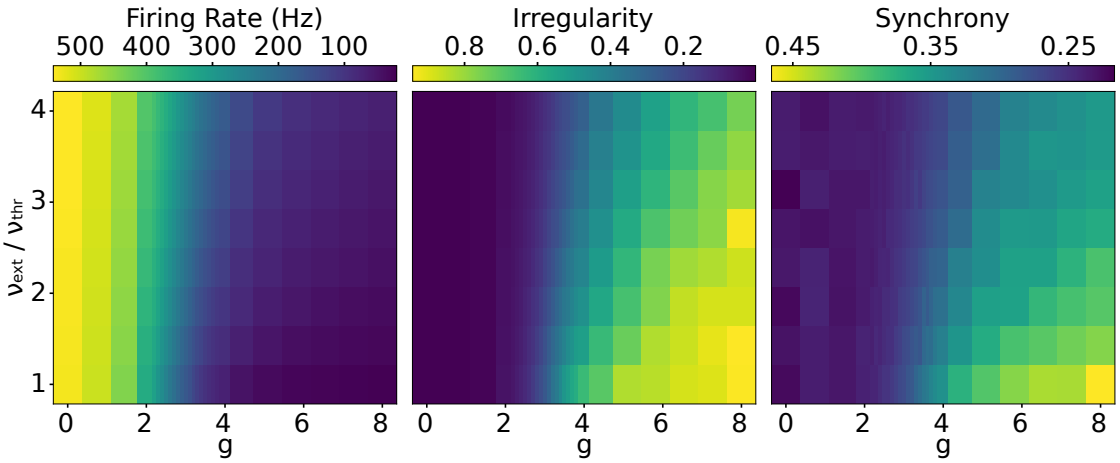


Figure 5.9.: Heat map of the downscaled balanced random network behavior with conductance-based synapses and adjusted parameter distributions. The visualization and synchrony parametrization correspond to fig. 5.1. The transition between regimes around $g \approx 3$ becomes less distinct, and the synchronous network behavior is lost.

Consequently, only routes that are also available on the hardware are incorporated into the model. This is necessary since, due to the limited flexibility and component count on the BrainScaleS-1 system, it is not possible to route each synapse's connection. As a result of this, synapse loss is observed, which is listed in table 5.3. There, different loss values are obtained for different external input rates. This is caused by an additional modification that is applied to the model to mirror the network topology on the hardware. External inputs are no longer modeled by a single external source per internal neuron but by a pool of external sources from which connections are sampled. As a consequence of this, the map and route results are dependent on the external input rate since the connections between populations compete for the available resources and higher rates require additional external sources that get distributed over the whole wafer. The reason for this modification and its implementation details are introduced in section 5.2.2.

In addition to the precise network topology, the number of repeaters involved in each route is extracted. Combined with the results of the delay calibration presented in section 4.2.5, a time delay is associated with each connection and incorporated into the model. Solely based on the repeater count, a discrete distribution is obtained, as demonstrated in fig. 5.10a. However, utilizing analog hardware, circuit variations have to be considered. Therefore, taking into account the uncertainties observed during the delay calibration, every delay value is sampled from a Gaussian distribution with its corresponding mean value and a standard deviation of 0.1 ms. Figure 5.10b illustrates the resulting delay distribution. It is shown for mappings with minimum and maximum external input rate. As expected, for higher rates, additional external sources are necessary, which get spread over the whole wafer. Hence, the routes cover greater

Table 5.3.: Synapse loss of the map and route results of the downscaled balanced random network model. Connections are classified according to participating populations, where neurons from the population to the left of the arrow transmit spikes to neurons of the population to the right of the arrow. There, "exc" symbolizes the excitatory and "inh" the inhibitory population. In the last two columns, the proportion of lost external connections and the total synapse loss are represented. As the map and route results vary with the utilized external input rate, a comprehensive list of all simulated rates is given.

$\nu_{\text{ext}}/\nu_{\text{thres}}$	exc → exc	exc → inh	inh → exc	inh → inh	External	Total
1.0	30.50 %	31.81 %	30.86 %	30.51 %	2.78 %	17.02 %
1.43	28.46 %	29.47 %	26.36 %	26.02 %	4.57 %	16.58 %
1.86	30.26 %	31.15 %	28.33 %	28.34 %	1.20 %	15.86 %
2.29	30.66 %	31.87 %	26.38 %	26.47 %	1.90 %	16.20 %
2.71	29.36 %	30.25 %	22.63 %	22.31 %	3.47 %	16.02 %
3.14	25.87 %	27.43 %	21.43 %	22.40 %	6.01 %	15.81 %
3.57	19.45 %	20.42 %	16.54 %	16.41 %	13.66 %	16.39 %
4.0	17.48 %	18.72 %	13.69 %	13.58 %	19.57 %	18.22 %

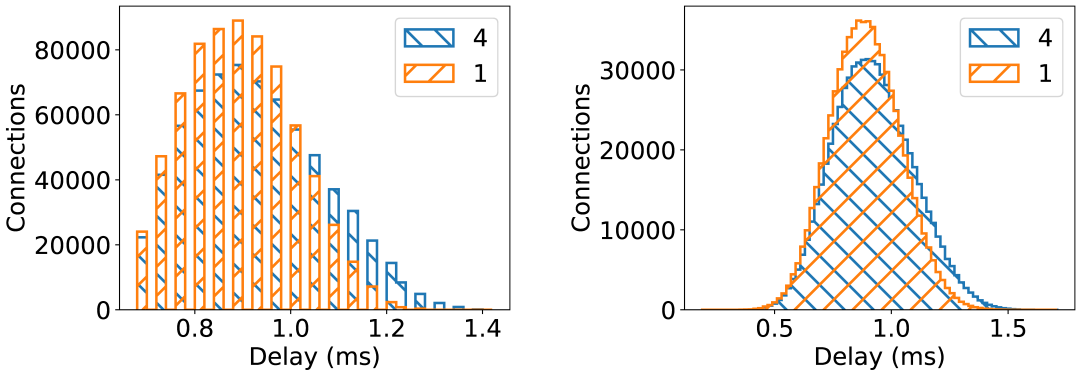


Figure 5.10.: Delays obtained from the map and route results of the downscaled balanced random network. According to the calibration results discussed in section 4.2.5, extracted connection lengths are translated into corresponding delay values. Since the delay depends solely on the number of repeaters, discrete results are observed in (a). There, each bin represents a specific repeater count. To model circuit variations measured during the calibration, in (b), individual connection delays are modeled by a Gaussian distribution with a standard deviation of 0.1 ms. Results are shown for the minimum and maximum external rates $\nu_{\text{ext}}/\nu_{\text{thres}}$, indicated by different colors.

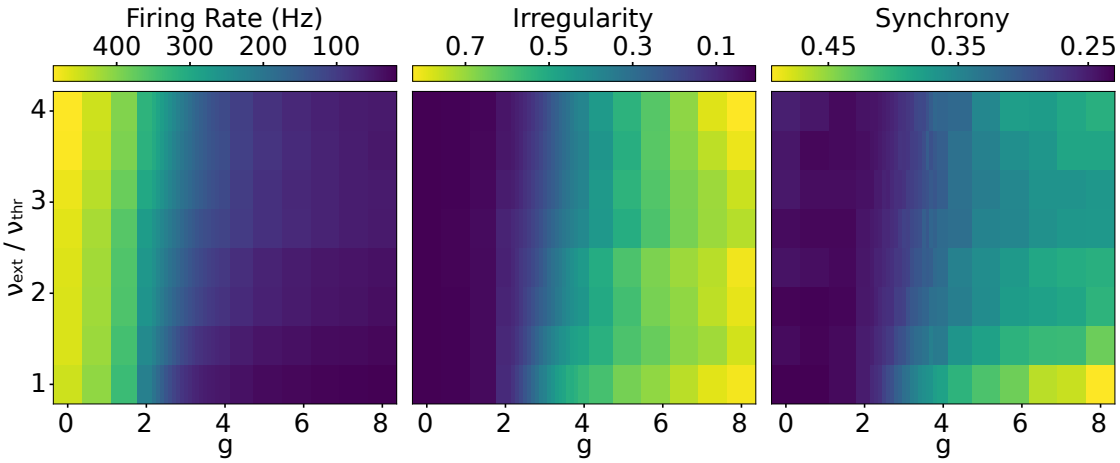


Figure 5.11.: Heat map of the downscaled balanced random network behavior with conductance-based synapses, adjusted parameter distributions, reduced synapse weight and incorporated map and route results. For each external input rate, only connections of the corresponding map and route result of the hardware are incorporated into the simulation. Delays are modeled using a Gaussian distribution with a standard deviation of 0.1 ms and a mean value that is determined in relation to the connection's length. External inputs are represented by a pool of sources, from which each internal neuron samples 200 connections (cf. section 5.2.2). The visualization and synchrony parametrization correspond to fig. 5.1. Due to the incorporated map and route results, slightly elevated irregularity values are observed in the asynchronous irregular regime ($g > 4$).

distances, resulting in a shift of the corresponding delay distribution towards larger values.

Finally, all the results are integrated into the model to simulate the downscaled balanced random network, which has been adapted to the hardware constraints. This includes incorporating conductance-based synapses, hardware parametrizations, as well as the routing and delay values extracted from the hardware for each external input rate. Furthermore, the weights of both excitatory and inhibitory synapses are adjusted to align with the available hardware configuration. This modification involves altering the correction factor of the prolonged synaptic time constant f_{corr} from 1.7 to 1.2. Simulation results confirm that this reduction does not alter the network's regimes. However, it does result in a slight decrease in irregularity and mean firing rates.

Resulting network statistics of the final network model are displayed in fig. 5.11. Despite the lost connections and the thus introduced inequality in synapses per neuron, the characteristic behavior of the different network regimes is preserved. However, an increased irregularity is obtained in the asynchronous irregular state for high external input rates. This is not observed in simulations with fixed map and route results

extracted from an external rate of $\nu_{\text{ext}} = \nu_{\text{thr}}$, which are used for the whole parameter range. Therefore, modified irregularities are traced back to the distinct routing results, which get more complex for elevated firing rates. Nevertheless, as the differences are small, the simulated firing statistics still represent a valid benchmark for the BrainScaleS-1 system obtained from a model that now matches the hardware's restrictions.

5.2. Implementation on BrainScaleS-1

Emulating large-scale biological models on wafer-scale neuromorphic hardware is a complex task that demands a high degree of control over both the hardware and the model itself. Therefore, as introduced in chapter 4, the handling of the BrainScaleS-1 system is improved. On the one hand, this enables large-scale emulations in the first place. On the other hand, it defines and minimizes the limitations of the hardware. Building upon these results, the balanced random network is adapted to align with the hardware's restrictions, as detailed in section 5.1. By combining both efforts, the modified model is successfully emulated on a single BrainScaleS-1 wafer system, which is demonstrated in this section.

To achieve this emulation, neurons within the network are mapped onto the hardware and routes are implemented between them with the aim of minimizing the number of lost synapses. In contrast to the cortical microcircuit, the structure of the balanced random network is simpler, comprising only two distinct populations. Consequently, it allows for routing the model with external inputs. Furthermore, it comprises a biologically implausible high firing rate regime with up to 500 Hz. Both aspects make it an excellent benchmark for assessing the I/O capabilities of the hardware.

This focus of the investigation is also addressed in the structure of this section. It begins by introducing the bandwidth limitations of the BrainScaleS-1 system and outlines techniques to overcome them. Following that, section 5.2.2 demonstrates the considerations made to translate the network topology into a hardware representation. Finally, section 5.2.3 discusses the results of the emulation.

5.2.1. Bandwidth Consideration

With its high firing regime for small inhibitory weights, the balanced random network represents a challenging benchmark in terms of spike communication. When taking into account the used speedup factor of 10 000 achieved by the hardware and the substantial number of concurrently spiking neurons, it becomes evident that bandwidth is a critical factor. Therefore, this section discusses the limits of spike processing on the hardware and presents solutions to overcome bottlenecks, enabling the emulation of the high firing regime. Unless otherwise specified, in this section, all values are provided in wall-clock time, i.e., in the hardware domain.

This consideration is divided into three spike transportation pathways: injecting external spikes from the FPGAs to the HICANNs, reading out spike results from the HICANNs to the FPGAs, and inter-neuron communication between HICANNs. The specific transportation layers are introduced in sections 3.1.2, 3.1.3 and 3.2.

External Input

Starting with the injection of external inputs, in agreement with Klähn 2017, a maximum transmission rate of 17.8 MEvents/s is measured between an FPGA and each of its 8 connected HICANNs. Here, each event corresponds to a single spike transmission. In comparison, the neurons of the downscaled balanced random network require Poisson-distributed spikes with firing rates of $\eta \cdot 3333\,\mathrm{Hz}$ in the biological regime, as outlined in eq. (5.2). For the maximum rate of $\eta = 4$, this translates to a minimum bandwidth requirement of 133 MEvents/s per neuron, assuming equidistant spike transmission.

To assess this requirement against the available bandwidth, it is multiplied by the number of neurons per HICANN, which is extracted from the final mapping results. This calculation remains independent of the finally utilized routing as only the chosen mapping strategy and the availability data of the membrane circuits influence the placement. As illustrated in fig. 5.12a, a maximum number of 60 neurons per HICANN is observed. Consequently, in the worst-case scenario, a rate of 8 GHz of Poisson-distributed spikes is necessary for such a HICANN, which already exceeds the combined capacity of all 368 links on a single wafer.

Therefore, in order to reduce the total number of external spikes, a pool of sources is generated from which the connections to the internal neurons are sampled. Its implementation details are introduced in section 5.2.2. In the final network topology, each neuron is connected to 200 external sources distributed across the entire wafer. As a result of this, a single external source transmits spikes at a maximum rate of 667 kHz and multiple sources can be accommodated by a single HICANN link. Nevertheless, the links are only filled up to 80 % capacity to account for periods of higher firing rates caused by the Poisson distribution.

Furthermore, the FPGA implementation was modified to count spikes that could not be transmitted. During the network emulations, it is determined that with the introduced configuration a maximum of 47 spikes are discarded on a single link, which is considered negligible. Consequently, no limitations are anticipated in the transmission from the FPGAs to the HICANNs.

Recording of Spikes

In the other direction, i.e., from the HICANNs to the FPGAs, a slightly higher bandwidth of 25 MEvents/s is measured. At the same time, the network requirements become more challenging. Assuming a firing rate of 400 Hz in the biological regime and on average 50 neurons per HICANN spikes occur at a rate of 200 MHz. However, in this direction, spike transmission is no longer time critical and spikes are buffered per neuron and per external input merger, as introduced in section 3.1.2. Therefore, a straightforward bandwidth comparison is no longer sufficient, and the count of dropped spikes is estimated using a Monte Carlo simulation of the whole merger tree and external readout. Results are shown in fig. 5.12b.

In addition to the implemented buffer size of 2 at the external input merger a larger buffer of size 16 is demonstrated, which was planned for a future chip revision. While

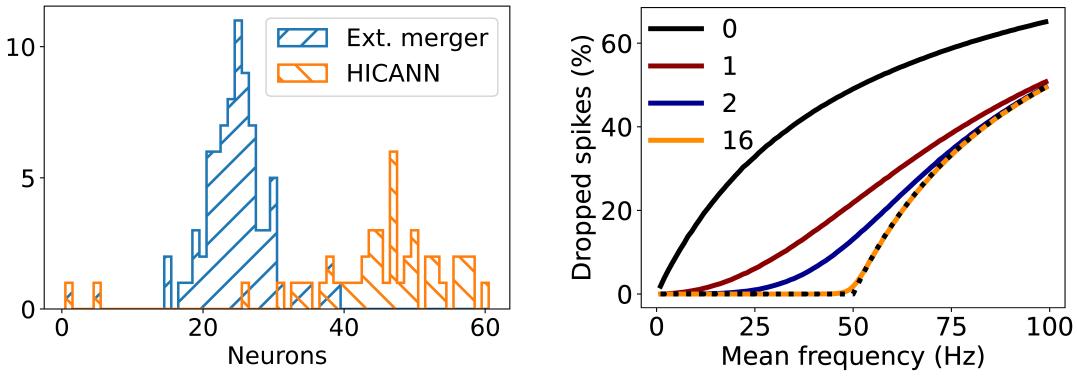


Figure 5.12.: Evaluation of spike drops during readout. (a) number of neurons placed per HICANN or per external input merger extracted from the placement results of the balanced random network, scaled down by a factor of 6. (b) Monte Carlo simulation of dropped spikes in the transmission from the HICANN to the FPGA. Simulating 10 s of biological time, 50 neurons are randomly placed on two external mergers on a single HICANN. Each neuron sends a random spiketrain generated by a homogeneous Poisson process with a fixed mean frequency shown in biological time. Identical to the hardware implementation, the minimum distance between two spikes is adjusted to be larger than 0.16 ms biological time, defined by the maximum clock frequency of the merger tree. Due to the faster operation and since spikes injected into the merger tree are buffered per neuron, possible drops are neglected at this stage. In addition, for each merger a FIFO buffer is simulated that stores spikes until they can be sent to the shared bus of the HICANN that transmits up to 25 MEvents/s. Different colors indicate different sizes of this buffer. Spikes arriving at a full buffer are dropped. The standard deviation of 50 repetitions of the simulation is shown as an error band, which is too small to be visible. The dotted line visualizes the theoretical minimum spike loss given by the bandwidth limitations of the bus, assuming an infinite buffer size where only spikes that have not been transmitted at the end of the experiment are dropped.

the larger buffer is close to the optimal solution, the current implementation already significantly reduces the number of dropped spikes.

On the one hand, firing rates below 30 Hz in the biological regime are considered acceptable, and the impact of dropped spikes during the readout is disregarded in this case. On the other hand, it is evident that the maximum readout rate is significantly limited for higher rates, with an expected maximum of 50 Hz per neuron, which aligns with emulation results obtained from a naive network emulation conducted using the presented neuron placement. Furthermore, the presented simulation aims for the irregular firing behavior expected for low rates. In contrast, in the high firing regime, regular spikes are anticipated. It is worth noting that worse performance is expected when neurons accommodated by the same external input merger spike simultaneously.

Consequently, a distinct solution needs to be devised for the high firing regime, particularly since it is not possible to detect dropped spikes on-chip. It is important to keep in mind that each HICANN provides its own link, enabling the parallelization of readout across all available chips. By manually adjusting the network mapping, it is possible to enforce a maximum of 6 neurons per HICANN, precisely matching the readout restrictions assuming a firing rate of 400 Hz of regularly spiking neurons. However, utilizing the entire wafer goes hand in hand with longer routes, resulting in significantly increased bus usage and high synapse losses. Therefore, an alternative approach is pursued, but the presented mapping is still used to verify internal bandwidth limitations.

In the final implementation, a subset of 30 neurons is separated and individually placed on 30 HICANNs. Thus, each of these neurons has full bandwidth available, equivalent to a firing rate of 2500 Hz in the biological regime. By restricting the evaluation to only these 30 neurons, the bandwidth limitation during readout is circumvented.

On-Chip Communication

Finally, the on-chip communication is examined. Neglecting the 1 MHz signals from the background generators used by the repeater circuits to adjust their timing, as introduced in section 3.1.4, the same network requirements as for the external readout apply. However, each internal bus provides a bandwidth of 62.5 MEvents/s. Consequently, the bandwidth is no longer shared by all neurons of a HICANN but is allocated per external input merger. Figure 5.12a illustrates the utilization of mergers in the final mapping, where two mergers are used per HICANN, each serving an average of 25 neurons.

Considering that each neuron buffers up to one spike, as introduced in section 3.1.2, no dropped spikes are expected, provided that all neurons spike regularly with a rate below 250 Hz in the biological regime, as all buffers are emptied before the next spike iteration occurs. For the network behavior as originally simulated, this would not suffice, and mapping adjustments would be required. However, due to additional limitations discussed in section 5.2.3, the hardware's neurons spike at exactly this maximum rate of 250 Hz. Consequently, no further adaptations are necessary. This has been confirmed through emulations with a maximum of 6 neurons per HICANN. Even when in this scenario the neurons are less constrained in terms of internal bandwidth, their results show the same maximum firing rates. Moreover, in the final mapping of the high firing

regime, 30 separated neurons on individual HICANNs also do not exhibit elevated rates.

Nevertheless, as spikes are buffered, spike times get shifted. To assess its extent, the merger tree behavior of 50 neurons divided onto two external input mergers is evaluated in a Monte Carlo simulation. Regarding neurons that spike regularly at a rate of 200 Hz for 10 s of biological time, only 78 of all spikes get shifted. For lower spike rates and irregular firing behavior even fewer shifts are observed. Therefore, the effect of shifted spikes is considered negligible and no internal limitations are expected for the observed rates.

5.2.2. Mapping the Model to the Hardware System

In this section, the map and route process for the adjusted balanced random network is discussed. Based on the bandwidth considerations introduced in the previous section, the excessive external input is implemented using a pool of external sources from which connections are sampled. As a consequence, the necessary external bandwidth is notably diminished. However, due to neurons sharing connections, this reduction is limited to the point where external stimulations are no longer considered independent. To address this limitation, a sufficiently large external pool has to be employed. Its size is aligned with the internal network's structure, which is also designed under the assumption of independent stimulations from a shared pool of neurons. Consequently, the number of external sources is matched to the count of internal neurons. Moreover, the number of utilized connections is determined by the best routing results. To this end, different network structures are realized and evaluated, visualized in fig. 5.13. Additional parameters of the map and route algorithms were chosen to minimize synapse loss and can be found in appendix A.5. When dealing with low external connection numbers, each external source is required to maintain high firing rates to preserve the total stimulus. Consequently, due to bandwidth limitations, fewer sources can be accommodated by a single HICANN link. Therefore, the sources are distributed across the entire wafer, necessitating longer connections, which, in turn, results in increased synapse loss. Simultaneously, with the reduced connection count, fewer buses are occupied near the internally placed neurons, leaving more available for internal connections, thus resulting in reduced synapse loss there.

Conversely, when more external connections are implemented, it becomes feasible to position many sources in close proximity to the target HICANNs. This results in fewer lost external routes. However, at the same time, bus utilization increases, leading to a higher number of internal routes being lost.

To maintain the balanced state of the network, an even distribution of losses on all connections is pursued. This is approximately the case for 200 external connections per neuron, which are consequently used in the final hardware implementation. Resulting statistics regarding synapse loss are listed in table 5.3, and the network behavior of the final model is showcased in section 5.1.5. In addition, simulations of the model without synapse loss indicate that the investigated network statistics remain unaffected by the sampling of external connections using the specified parametrization.

As introduced in section 5.2.1, specialized neuron placement is required to record the immense number of spikes obtained in the high firing regime. There, manual adjustments

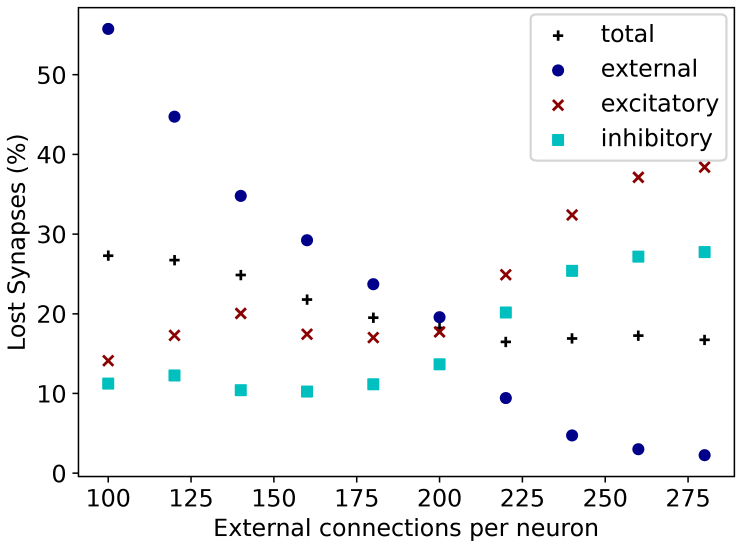
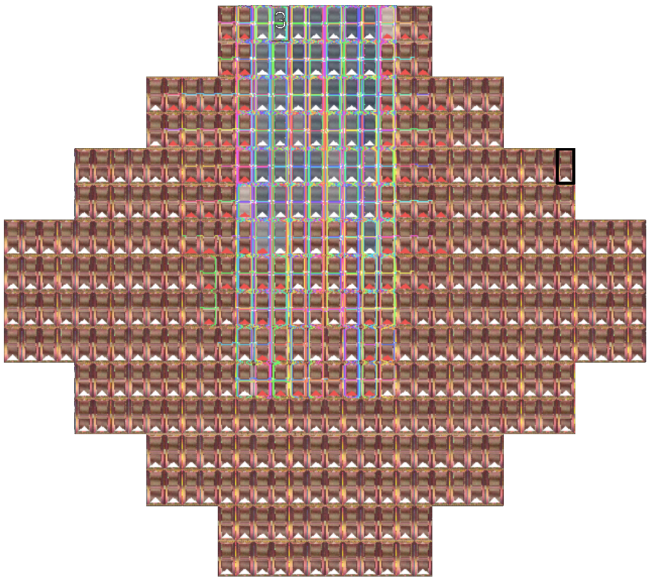


Figure 5.13.: Impact of the external connection count per neuron on the map and route performance. Depicted is the synapse loss derived from the map and route results of the downscaled balanced random network for varying numbers of connections to the external pool of sources. Different colors represent the total loss of synapses, the loss of external connections, or the loss of internal excitatory or inhibitory connections. Similar losses are observed across all connection types when a total of 200 connections are employed. The data is obtained for the maximum external firing rate of $\nu_{\text{ext}} = 4\nu_{\text{thr}}$ and the parametrization displayed in appendix A.5.

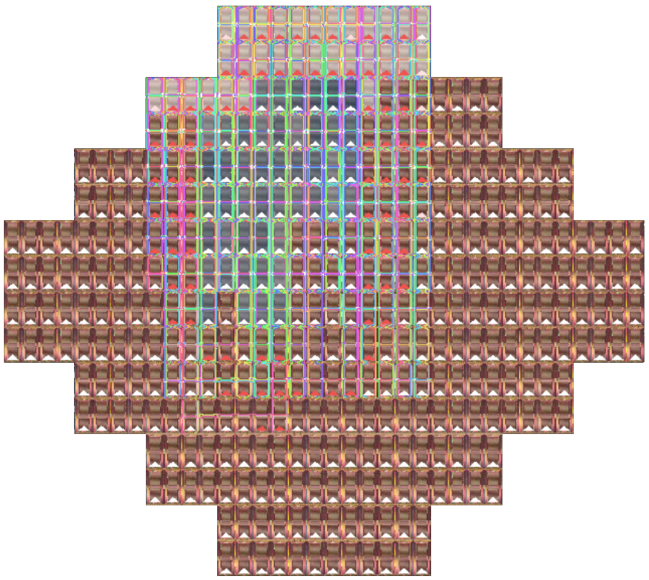
are made to the automatic mapping to separate 30 neurons onto individual HICANNs. All other settings remain unchanged. Due to the larger distribution of neurons, a small increase of the total synapse loss to an average value of 24 % is observed. A visualization of the resulting distribution of placed neurons and routes on the wafer for both models is depicted in fig. 5.14. As a result of the random connection topology of the model, which allows for potential connections between all neurons, simplifications cannot be employed, and routing complexity grows quadratically with size. Consequently, even though only a fraction of the wafer is occupied by neurons, the model's size is constrained by bus utilization.

5.2.3. Emulation on BrainScaleS-1

Finally, with the model being adapted and mapped to the hardware, it is emulated on the BrainScaleS-1 system. To this end, the neurons are configured according to the mean values obtained from the calibration, introduced in table 5.2. There, no parameter distributions are applied, as they are used to model the circuit variations in the first



(a)



(b)

Figure 5.14.: Visualization of the map and route result of the balanced random network on BrainScaleS-1. (a) displays the unmodified placement and (b) the placement with 30 neurons separated onto distinct HICANNs. Each HICANN is depicted as a rectangle with a triangle at the bottom. To illustrate this, in (a), an unused HICANN at the right edge is outlined in black. A red triangle indicates the injection of external spikes. Neuron placement is represented by the use of blue coloration, with darker shades indicating higher neuron counts. Therefore, the light blue HICANNs at the top of (b) signify the separately placed neurons. Connections are visualized as colored lines routed along the edges of the chips.

place.

The only exception to this is the synaptic weight. As the inhibitory weight is examined in relation to the excitatory weight by the parameter g, its values span almost one order of magnitude. In the adapted downscaled model, values between $2.4\,\mathrm{s}^{-1}$ and $19.2\,\mathrm{s}^{-1}$ are required for $w_\mathrm{I}/C_\mathrm{m}$. In comparison, the inhibitory weights that are achieved on the hardware for the chosen neuron parametrization range from $0.70\,\mathrm{s}^{-1}$ to $18.72\,\mathrm{s}^{-1}$, with only two possibilities to modify them: on the one hand, as discussed in section 5.1.3, choosing more distant reversal potentials results in smaller weight requirements. However, the reversal potentials are already optimized and further increasing them leads to an undesired low signal-to-noise ratio of the membrane. On the other hand, the small capacitors of the neuron circuits could be used. In this case, the weight calibration reveals a minimum synaptic weight of $5.82\,\mathrm{s}^{-1}$. Since this exceeds the minimum required weight by more than a factor of two it is even less suitable. Therefore, the emulations are limited to the presented weight range.

In general, this suffices to explore the various network behaviors. However, the primary focus of this thesis lies in obtaining the asynchronous irregular regime with low firing rates. On the one hand, this regime mirrors a biologically plausible firing pattern, which is also observed in the cortical microcircuit. On the other hand, with the BrainScaleS-1 system being specifically designed for biological networks, it comfortably operates within the hardware's bandwidth constraints. Therefore, an adequate hardware representation is found without the need to reduce the number of evaluated neurons. As a consequence, the model's synaptic weights are adjusted to achieve even higher values of g on the hardware. To this end, the excitatory weight is set to $w_\mathrm{E}/C_\mathrm{m} = 1.69\,\mathrm{s}^{-1}$. For comparison reasons, this weight adjustment is also incorporated in the final simulation results, depicted in fig. 5.11.

With these settings, the network is emulated on the BrainScaleS-1 hardware. In contrast to the software simulations, the network is emulated for 60 s of biological time. The resulting extension of runtime is negligible when considering the system's speedup factor. However, prolonging the emulation provides the network with additional time to settle in the beginning, which is desirable as it is not possible to initialize the neuron circuits on the hardware.

Figure 5.15 presents the obtained network statistics. For values of g and ν_ext that result in mean firing rates above 30 Hz, the alternative neuron placement with 30 separated neurons is used. In this case, only the separated neurons are evaluated to overcome the bandwidth limitations. Due to the synchrony's dependence on the number of evaluated neurons and since 30 neurons are insufficient to produce reliable results, no synchrony values are determined for the alternative placement.

To simplify the automatic weight configuration process, the hardware parameters for both synapse types are set according to the excitatory weight calibration. However, to address the discrepancies in calibration results observed for the inhibitory synapses, as demonstrated in fig. 4.15, the evaluation process takes into account the correct biological representation for each set of hardware parameters. As a consequence of this approach, irregular values of the parameter g are obtained.

Although network regimes with firing statistics similar to those depicted in fig. 5.11

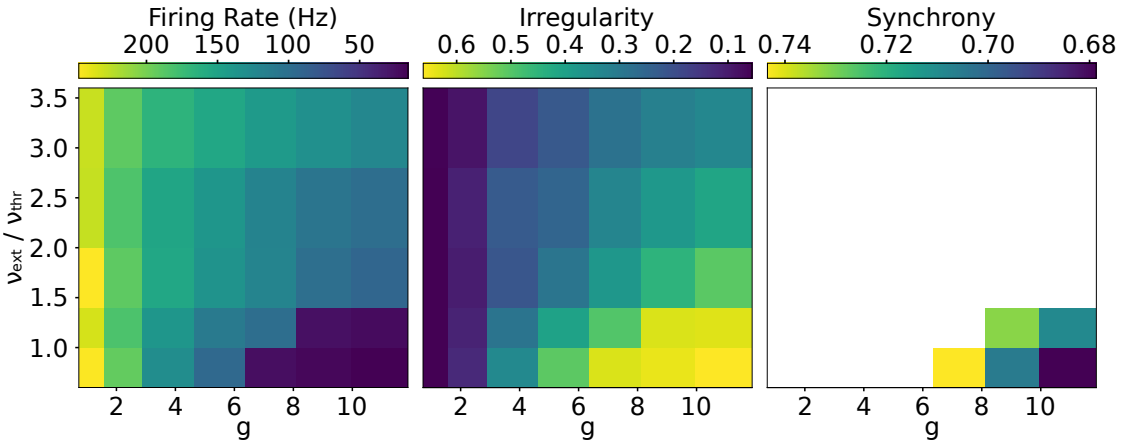


Figure 5.15.: Heat map of the emulated downscaled balanced random network behavior on BrainScaleS-1. The visualization and synchrony parametrization correspond to fig. 5.1. Each configuration is emulated 10 times, and the resulting mean values are depicted. For configurations with mean firing rates below 30 Hz, the unmodified placement is used. For all other configurations, the alternative placement is utilized, where only 30 separated neurons are evaluated. Different placements can be distinguished, as a synchrony value is determined only when all neurons are evaluated.

are observed, there are distinct differences noticeable.

Firstly, the maximum firing rate of the network is reduced to less than 250 Hz. This rate is also evident for a weight factor of $g = 0$, which, due to the parasitic capacities in the synaptic input lines, is represented by removing all inhibitory connections. Measured on separately placed neurons and also confirmed in emulations with a maximum of 6 neurons per HICANN, this is not a result of bandwidth limitations on the chips. However, as demonstrated in section 4.4.4, there is a limitation on the maximum conductance that can be generated by a single synaptic input circuit. Consequently, in contrast to the software simulation, the maximum neuron stimulation and thus the maximum firing rate are reduced. This observation is further validated by testing a single neuron that receives strong stimulations from eight synaptic inputs, resulting in similar maximum firing rates.

Furthermore, this limitation becomes apparent when comparing the firing rates obtained for external input rates of $\nu_{\text{ext}} = 0.8\nu_{\text{thr}}$ with $\nu_{\text{ext}} = 3.2\nu_{\text{thr}}$ at $g = 1.16$. For higher external stimulations, lower firing rates are observed. This is caused by the different number of utilized synaptic inputs, which is depicted in fig. 5.16. On the one hand, for lower external rates, it is possible to accommodate more external sources on a single HICANN, as explained in section 5.2.2. Due to the random sampling of connections, sources end up sharing the same target neurons. Consequently, more connections have to be routed from a single HICANN link to individual neurons. On the other hand, the topology of the wafer dictates that only connections from the same external input

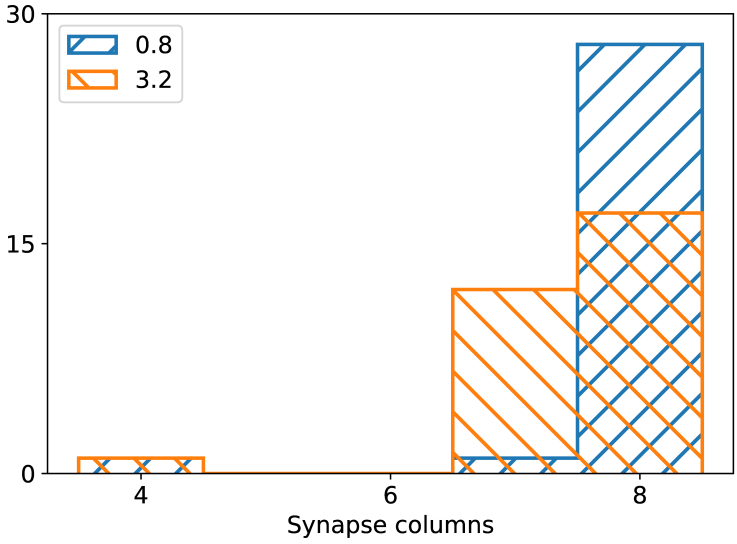


Figure 5.16.: Synaptic input utilization. The histogram shows the number of synaptic inputs utilized in the 30 separated neurons used to investigate the network behavior in the high firing regime. Different colors show the placement results for different external input rates ν_{ext} given in relation to ν_{thr}. For each neuron, there are a maximum of 8 inputs available, one for each of its membrane circuits.

merger can share buses. Moreover, only when the signal is received from the same bus it is distributed across different synapse columns instead of different synapse rows. This results in more utilized synaptic inputs and therefore also higher firing rates for lower external rates.

This effect could be circumvented by standardizing the placement of external sources. However, in this context, it is intentionally refrained from doing so to demonstrate the hardware's behavior.

Figure 5.16 also demonstrates that the implemented placement closely approaches maximum synaptic input utilization. The only possibility to increase this number is to build larger composite neurons comprising more membrane circuits. However, as demonstrated in section 4.4.2, the finite resistances between the circuits result in undesired changes of the membrane behavior. Therefore, no further adjustments are made in this thesis.

Additional modifications of the network behavior are demonstrated in fig. 5.17. It displays a comparison of the firing rates in the emulation with those of a NEST simulation of the final model. Both models share the same parametrization and network topology. The results are presented for both the minimum and maximum external rates.

Except for small values of g, the emulations exhibit higher firing rates. This is also attributed to synaptic input saturation. With higher weights on inhibitory connections,

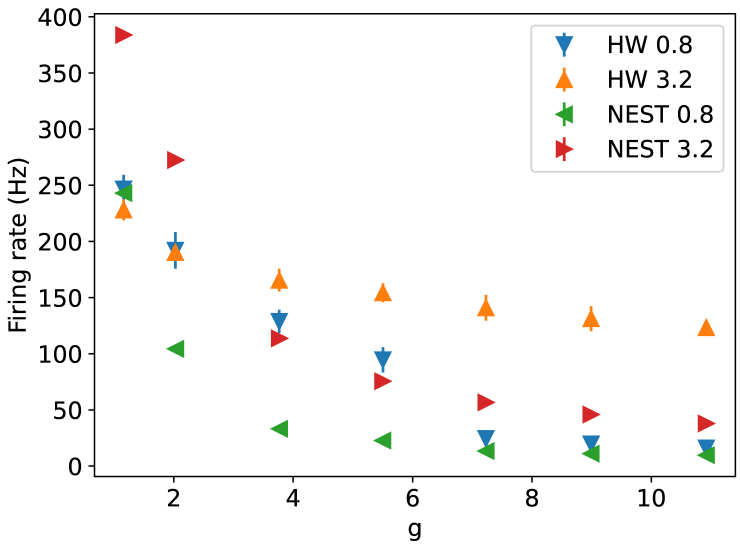


Figure 5.17.: Comparison of firing rates between emulation and simulation. The graph illustrates the firing rates of the hardware emulation (HW) and NEST simulation (NEST) at external input rates of 0.8 and 3.2 in relation to ν_{thr}. In both implementations, the same network topology is employed for each external rate. Error bars represent the standard deviation across 10 experiment repetitions. To achieve this, different seeds are used in the simulation. In all simulations and for low firing rates in the emulations, deviations are too small to be discernible.

their saturation effect is more pronounced compared to excitatory ones. Consequently, inhibition is diminished, leading to elevated firing rates. However, in the asynchronous irregular regime with firing rates below approximately 50 Hz, this effect becomes negligible, and similar results are observed. This observation is also evident in terms of irregularity. For small values of g, where both inputs are equally saturated, regular firing in accordance with the simulation results is obtained. However, with the attenuation of inhibition, less irregularity is observed for higher values of g and high firing rates.

To account for this and to extend the observable range in the desired regime, the investigated external input rates are shifted to slightly smaller values compared to the original simulations. In agreement with the original model, for even lower external rates than those depicted, the neurons cease to spike due to insufficient stimulation.

Furthermore, in the regime characterized by firing rates below 30 Hz, the level of network synchrony is determined, revealing increased values. A possible explanation for this is the presence of additional neuron correlations on the hardware, such as correlated weights caused by shared synapse drivers, as explained in section 4.2.3, or increased stimulation due to concurrently spiking synapses, as discussed in section 4.4.3. These factors are not accounted for in simulation. However, since the regime is still characterized

by asynchronous irregular behavior, no further investigations are conducted in this regard.

All in all, similar firing patterns are observed in simulation and on the hardware. For high inhibition and small external input rates, asynchronous irregular network behavior is obtained with rates similar to those observed in simulation. Moreover, the emulation results are stable, and consistent statistics are measured when repeating the experiment. By reducing the inhibitory weight factor, a transition to regular firing with elevated rates is found. Although the hardware is limited with respect to the biologically implausible high stimulation, it is still valuable in demonstrating the capabilities of the bus system to handle this enormous number of spikes, especially considering that the hardware operates with a speedup factor of 10 000 compared to biological real-time.

6. The Cortical Microcircuit Model on BrainScaleS-1

Built upon the principles of the human brain, neuromorphic hardware is designed to accelerate and enhance the power efficiency of spiking neural network simulations. To accomplish this, it is necessary to have appropriate benchmarks that facilitate the demonstration of system functionality and enable performance comparisons [Davies 2019]. Large-scale biological networks are particularly well-suited for this purpose, as they present a significant challenge to conventional computers due to their intricate network topology with numerous interacting components. The cortical microcircuit, introduced in section 2.5, is such a network. By replicating the structure and behavior of the brain's cortex under the surface of $1\,\mathrm{mm}^2$, it has become a standard benchmark for neuromorphic computing [Ostrau et al. 2022]. Recently, it has been used to showcase the capabilities of various simulators [Albada et al. 2018; Rhodes et al. 2020; Knight et al. 2021; Golosio et al. 2021].

Therefore, in this thesis, the network is emulated on a BrainScaleS-1 system to evaluate the hardware's capabilities. However, due to the limited flexibility of the system, emulating the original model is not feasible. Instead, adaptations are necessary to align the model with hardware constraints. For this purpose, similar to the balanced random network implementation discussed in chapter 5, a parallel simulation of the model using the NEST simulator is conducted alongside the hardware-based efforts. This approach facilitates the exploration of network adaptations and allows for a comparison of the final emulation results.

This chapter encompasses both the simulation and emulation endeavors. Section 6.1 introduces the NEST simulation, its necessary modifications, and their impact on the network's behavior. Subsequently, section 6.2 delves into the hardware implementation.

6.1. Adapting the Model to the Neuromorphic Hardware

Following a physical modelling approach, the neuron and network properties on the BrainScaleS-1 hardware are restricted by the capabilities of its integrated circuits. As a result, networks must either conform to these limitations or be adjusted to align with them. For this purpose, this thesis utilizes software simulations that facilitate the exploration of model modifications.

Chapter 5 illustrates existing hardware constraints and presents potential adaptations for large-scale biological networks using the balanced random network model as a foundation. Since the structure of the cortical microcircuit is derived from the balanced random network, the same constraints and strategies to address them can be applied

to it. However, due to its more complex multi-layered structure, preserving the original network behavior is not feasible. Consequently, the adaptations primarily aim to preserve a biologically plausible neuron behavior characterized by an asynchronous irregular firing pattern with firing rates below 30 Hz, which is also observed in the balanced random network.

This section introduces the impacts of these adaptations on the cortical microcircuit. It closely follows the structure of section 5.1, commencing with a demonstration of the original network behavior in section 6.1.1. Following that, section 6.1.2 discusses the reduction of neuron and synapse counts to fit the network within a single BrainScaleS-1 wafer. Subsequently, section 6.1.3 addresses the transition from current-based to conductance-based synapses. Based on these findings, section 6.1.4 illustrates the effects of adjusted and distributed neuron parameters, focusing on different delay distributions.

6.1.1. Simulation of the Original Model

This section presents the NEST simulation of the original cortical microcircuit model, as detailed in Potjans et al. 2012. Implemented during the bachelor thesis of Jonas Weidner [Weidner 2019], it serves as a reference for all subsequent model adaptations. Additionally, it is used to determine appropriate parametrizations for the synchrony evaluation.

The simulation follows the neuron and network descriptions provided in section 2.5.1. For its implementation, the connection probability $K_{\text{pre/post}}$ for each pair of pre- and postsynaptic populations is converted into a corresponding connection count $C_{\text{pre/post}}$. By reversing the probability calculation, it is determined to

$$C_{\text{pre/post}} = \frac{\log\left(1 - K_{\text{pre/post}}\right)}{\log\left(1 - \frac{1}{N_{\text{post}} N_{\text{pre}}}\right)}. \tag{6.1}$$

Here, N_{pre} and N_{post} represent the neuron counts of the populations involved. Each of these connections is established by randomly selecting a pre- and post synaptic neuron from the respective populations. Consequently, unlike in the balanced random network model, neurons do not have identical synapse counts.

The resulting network is simulated for 10 s of biological time with a simulation time-step of 0.1 ms. Spikes obtained in the first second of the simulation are excluded from the evaluation to allow for the network behavior to stabilize. Furthermore, since the thalamo-cortical external input of the model is limited to the first 10 ms of the simulation, it is expected to have negligible impact on the obtained network behavior and is therefore omitted.

To enable a comparison, the resulting network characteristics are initially prepared as described in Albada et al. 2018. There, histograms are generated using a bin width according to the Freedman-Diaconis [Freedman et al. 1981] rule:

$$\text{Binwidth} = 2\frac{\text{IQR(x)}}{\sqrt[3]{n}}. \tag{6.2}$$

Here, IQR(x) represents the interquartile range of the data and n is the number of data points. Moreover, the results are smoothed using the *scipy* Gaussian kernel density function [Jones et al. 2001] with a bandwidth of $0.3\,\mathrm{s}^{-1}$. The firing rate distributions obtained in this manner for individual populations closely align with those from the original publication and the implementation on the SpiNNaker system [Albada et al. 2018], as illustrated in fig. A.12.

However, given the substantial modifications in the model behavior due to necessary adaptations, the given visualization method is unsuitable for meaningful comparisons. With alterations in network characteristics, bin sizes change, leading to substantial modifications in the visualized data. Unfortunately, this effect is not discernible due to the application of Gaussian smoothing. For this reason, in the following, the firing rates are represented in normalized histograms with a fixed bin width of 1 ms. The outcomes of this method for the original model are visualized in fig. 6.1. For the purpose of better comparison, the figure also includes the downscaled model with both current-based and conductance-based synapses, which will be discussed in later sections.

In addition, the irregularity and synchrony of the network are assessed using the methods described in section 2.3. Given that modifications to the network behavior due to model adaptations are inevitable, the focus during the alignment of the model is on preserving the asynchronous irregular firing behavior of the populations. In the original publication, this behavior is defined by mean firing rates below 30 Hz, mean irregularity values falling between 0.7 and 1.2, and a synchrony value below 8. There, particular attention must be paid to the synchrony due to its dependence on the chosen bin width of the spike count histogram and the number of evaluated neurons. For this reason, in accordance with the original publication, a bin width of 3 ms is utilized. However, due to the necessary downscaling of the model, it is not possible to preserve a neuron count of 1000 evaluated neurons per population. For this purpose, the evaluation is limited to 100 neurons, and the synchrony boundary is rescaled accordingly.

The relationship between synchrony and neuron count is determined by randomly sampling various numbers of neurons from the full-scale model. As visualized in fig. 6.2, a linear correlation is observed. Furthermore, for lower neuron counts, the reduction of the mean bin height and a maximum of one spike per bin, results in similar mean and variation values for the bin heights, leading to synchrony values of 1. Therefore, the original boundary value of 8 can be linearly extrapolated to a new threshold of 1.7 for 100 investigated neurons, which is applied in the following considerations.

Table 6.1 lists the resulting mean network characteristics of the simulated model. There, the mean irregularity of all populations is smaller than expected from the original publication and the obtained values are situated at the lower boundary of the irregular firing regime. For a more comprehensive understanding of the results, the irregularity distribution within each population is depicted in fig. A.13. Theses values closely align with those observed in the SpiNNaker implementation in Albada et al. 2018. Moreover, the population behavior is consistent with the original model. For instance, the excitatory population of layer 5 demonstrates the highest level of synchrony while the inhibitory population of layer 6 exhibits the lowest synchrony value. This suggests that the observed discrepancies likely arise from differences in the evaluation method and are considered

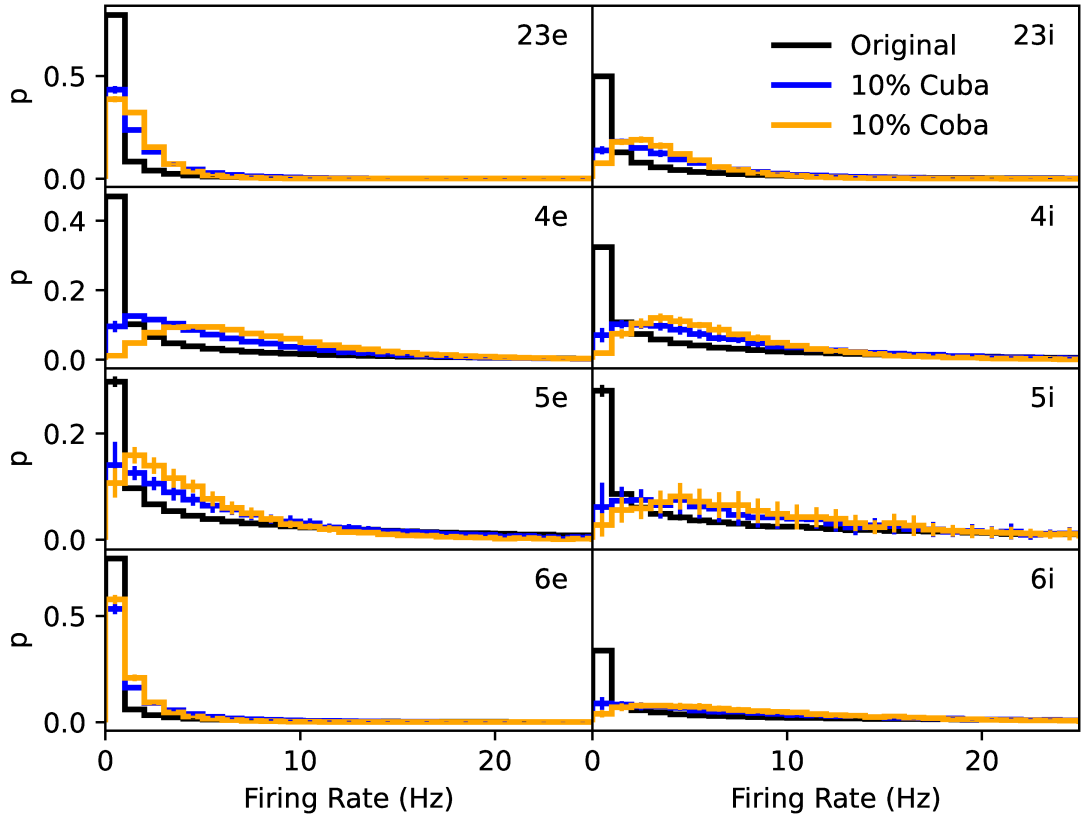


Figure 6.1.: Firing rate distributions of the NEST simulation of the cortical microcircuit model for different stages of adaptation. "Original" shows the behavior of the model with parameters extracted from Potjans et al. 2012. "10 % Cuba" demonstrates the downscaled current-based version, and "10 % Coba" the downscaled conductance-based implementation of the model. The mean firing rates of the neurons are depicted as a histogram, with a fixed bin width of 1 ms. The area beneath the histograms is normalized to one. Each row displays the results of a different layer of the network, with the excitatory population on the left and the inhibitory population on the right. Displayed are the mean values obtained from 30 simulations, each featuring different randomly generated connections. The error bars represent the standard deviation across these simulations.

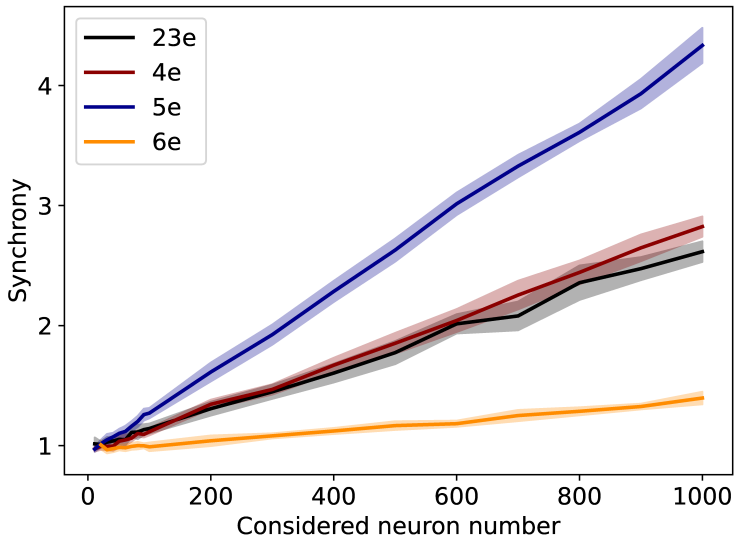


Figure 6.2.: Relationship between synchrony and the number of considered neurons. The synchrony calculation is limited to spikes obtained from a subset of neurons which are randomly drawn. The results are depicted for different sizes of these subsets. Exemplarily, only the results for the excitatory populations are shown; inhibitory populations behave analogous. The shaded area represents the standard deviation across 10 subsets of neurons with same size.

Table 6.1.: Network characteristics obtained form the NEST simulation of the cortical microcircuit model with parameters as outlined in Potjans et al. 2012. Different columns list the mean rates, mean irregularities and synchrony values of all eight populations. Displayed are the mean value and standard deviation obtained from 30 simulations, each featuring different randomly generated connections.

Population	Rate (Hz)	Irregularity	Synchrony
23e	0.85 ± 0.02	0.646 ± 0.004	1.11 ± 0.05
23i	2.99 ± 0.02	0.725 ± 0.005	1.03 ± 0.04
4e	4.52 ± 0.03	0.719 ± 0.002	1.10 ± 0.05
4i	6.30 ± 0.01	0.738 ± 0.003	1.01 ± 0.04
5e	7.7 ± 0.2	0.721 ± 0.004	1.24 ± 0.05
5i	8.97 ± 0.02	0.695 ± 0.008	0.95 ± 0.03
6e	1.65 ± 0.03	0.683 ± 0.004	1.00 ± 0.02
6i	8.43 ± 0.02	0.683 ± 0.005	0.94 ± 0.03

negligible compared to the expected model modifications resulting from the alignment to the hardware. Additionally, the network behavior is found stable with respect to different implementations of its random network connections.

When disregarding the peak of small irregularity values obtained from a small percentage of neurons exhibiting regular firing at higher rates, the irregularity distribution is accurately represented by its mean value. Therefore, in the subsequent analysis, the primarily focus is on comparing the mean values, with detailed distributions provided in the appendix for reference.

6.1.2. Downscaling the Model and Replacing the External Input

As already demonstrated through the balanced random network, the routing possibilities of the BrainScaleS-1 system are limited. Therefore, in order to mitigate large number of not implemented connections, it is necessary to downscale the cortical microcircuit to 10 % of its original size to fit on a single wafer, as detailed in section 6.2.1.

To achieve this, the same methodology as employed in the balanced random network is used. Neuron and synapse counts are both downscaled by a factor of $k = 10$. At the same time, to compensate for the lost stimulation, the synaptic weights are linearly increased by the same factor. This results in an increased internal variance, which can be compensated by substituting portions of the external Poisson input with a constant input current. However, as demonstrated in Albada et al. 2014, there exists a limitation to this scaling method, which is reached when the external input is completely replaced. As a result, while a network with 90 % of the original size can still be simulated with comparable network statistics, the same does not apply for a network scaled down to 10 % of the original model. Consequently, modifications of the network behavior become inevitable and the primary focus of this thesis is on preserving a biologically plausible network behavior characterized by asynchronous irregular firing patterns with mean firing rates similar to the original model.

To achieve this, the entire external input is replaced by an external current, calculated for each population according to:

$$I_{\text{ext}}^{\text{pop}} = w_{\text{e}} \cdot C_{\text{ext}}^{\text{pop}} \cdot \nu_{\text{bg}} \cdot \tau_{\text{syn}}^{\text{e}}. \tag{6.3}$$

Here, $C_{\text{ext}}^{\text{pop}}$ is the number of external connections received by the population, which is listed in table A.3, w_{e} and $\tau_{\text{syn}}^{\text{e}}$ are the weight and synaptic time constant of the excitatory external input, and ν_{bg} the external input rate. Due to the missing possibility on the hardware to inject currents, this external input is generated by an increased leak potential of

$$E_{\text{leak,new}}^{\text{pop}} = E_{\text{leak}} + \frac{I_{\text{ext}}^{\text{pop}}}{g_{\text{leak}}} = E_{\text{leak}} + I_{\text{ext}}^{\text{pop}} \frac{\tau_{\text{m}}}{C_{\text{m}}}. \tag{6.4}$$

According to eq. (2.1) this represents the same external stimulation.

Additionally, two factors are introduced, which independently modify the weights of the excitatory and inhibitory synapses. By this, similar to the balanced random network model, the imbalance between excitatory and inhibitory weights can be adjusted to correct

Table 6.2.: Network characteristics obtained form the downscaled NEST simulation of the cortical microcircuit model. Different columns list the mean rates, mean irregularities and synchrony values of all eight populations. Displayed are the mean values and standard deviation obtained from 30 simulations, each featuring different randomly generated connections.

Population	Rate (Hz)	Irregularity	Synchrony
23e	1.91 ± 0.05	0.915 ± 0.009	1.11 ± 0.04
23i	4.28 ± 0.06	1.02 ± 0.01	1.12 ± 0.04
4e	6.4 ± 0.1	1.059 ± 0.009	1.15 ± 0.04
4i	7.51 ± 0.04	1.12 ± 0.01	1.07 ± 0.04
5e	6.7 ± 0.5	1.06 ± 0.02	1.48 ± 0.07
5i	10.4 ± 0.2	1.08 ± 0.03	1.12 ± 0.05
6e	2.1 ± 0.1	0.88 ± 0.02	1.10 ± 0.04
6i	10.01 ± 0.06	1.02 ± 0.02	1.05 ± 0.04

the observed network activity. Due to the random connectivity of the 8 populations with either excitatory or inhibitory synapses the network behavior is complex and higher firing rates of a specific population are not necessarily compensated by a reduction of the excitatory weight. Therefore, various parameter combinations and network models have been tested regarding the best-matching mean firing rates with the original model. The closest match was found for a linearly downscaled network with completely replaced external input and an additional excitatory weight factor of 0.7 and an inhibitory weight factor of 1.4, which corresponds to a relative inhibitory weight factor of $g = 8$.

The resulting firing rate distribution of this network is depicted in fig. 6.1, and the mean network characteristics are detailed in table 6.2. Notably, the network behavior changes. As introduced for the balanced random network, neuron firing is driven by membrane fluctuations within the asynchronous irregular regime. Therefore, despite partial compensation through an increase in the relative inhibitory weight, the heightened variations caused by the stronger weights lead to higher firing rates in the majority of neurons. Consequently, the distributions are not longer dominated by neurons with firing rates below 1 Hz but are shifted towards higher rates, predominantly below 20 Hz. As a consequence of this shift, the mean firing rates of all populations increase.

With the heightened internal variation of the neurons, consistent with the findings from the downscaled balanced random network model, also higher irregularity values are observed. Regarding synchrony, no specific trend is identified. While certain populations exhibit elevated values, they still remain within the asynchronous regime.

In summary, as anticipated, the network characteristics are not preserved in the downscaled model. Nevertheless, all observables remain within the asynchronous irregular regime, with mean firing rates comparable to the original model.

6.1.3. Transition From Current-Based to Conductance-Based Synapses

Analogous to section 5.1.3, the synapse model of the downscaled cortical microcircuit has to be changed from current-based synapses to conductance-based synapses. This transition involves the modification of mean synaptic weights, as specified in eq. (5.6). This modification ensures the preservation of the PSP height of the membrane potential in response to a single spike for a neuron that is far from the high-conductance state and starts from its mean membrane potential $\langle U \rangle$. In this context, the mean membrane potential of a specific postsynaptic neuron is calculated as

$$\langle U \rangle_{\text{post}} = E_{\text{rest}} + \frac{\tau_{\text{m}}}{C_{\text{m}}} \cdot \tau_{\text{syn}} \left(\sum_{\text{pre}} w_{\text{pre}} \cdot C_{\text{pre/post}} \cdot \langle \nu_{\text{pre}} \rangle + w_{\text{exc}} \cdot C_{\text{ext}}^{\text{post}} \cdot \nu_{\text{bg}} \right), \tag{6.5}$$

taking into account the inputs received from all eight pre-synaptic populations, each firing with an average rate of $\langle \nu_{\text{pre}} \rangle$, along with the contribution from external sources. Here, all parameters are adopted from the original full-scale network description. Additionally, the standard deviation of the weight distributions is again set to 10 % of the newly calculated mean value for each population.

While considering additional adjustments, such as modifying the time constants of the neurons, they are found to be less advantageous since they would require parameters that cannot be implemented on the hardware. However, similar to the balanced random network, the selection of appropriate reversal potentials plays a crucial role. Optimal results are achieved when employing reversal potentials of $E_{\text{rev}}^{\text{e}} = 50\,\text{mV}$ and $E_{\text{rev}}^{\text{i}} = -150\,\text{mV}$. In this case, the threshold value, which remains consistent across all populations, is positioned in their center. Furthermore, distant values are chosen to limit the influence of the high-conductance state. However, a narrower gap is selected in comparison to the parametrization of the balanced random network. This choice is influenced by the reduced separation between the threshold and resting potential of the neurons within the cortical microcircuit. Given the smaller sub-threshold regime, the hardware achieves higher resolution by mapping its maximum and minimum voltages to biologically closer values, as elaborated in section 4.2.1.

Utilizing this parametrization, the simulation reveals the firing rate distribution illustrated in fig. 6.1, and the mean network characteristics listed in table 6.3. As introduced in section 5.1.3, a distinct neuron behavior is observed in the case of conductance-based synapses. As a consequence of this, the network characteristic changes and slightly smaller mean firing rates and irregularity values are measured. Furthermore, the difference in synchrony between the populations is reduced. However, with the presented parametrization, similar first- and second-order statistics are obtained without the need for further adjustments, and the network maintains the desired asynchronous irregular behavior.

6.1.4. Introducing Hardware Parameters With Variations

Due to the limited configurability of the hardware and the manufacturing-induced variations between circuits, the neuron parameters of the model must be adjusted

Table 6.3.: Network characteristics obtained form the downscaled NEST simulation of the cortical microcircuit model with conductance-based synapses. Different columns list the mean rates, mean irregularities and synchrony values of all eight populations. Displayed are the mean values and standard deviation obtained from 30 simulations, each featuring different randomly generated connections.

Population	Rate (Hz)	Irregularity	Synchrony
23e	1.62 ± 0.04	0.853 ± 0.008	1.13 ± 0.06
23i	3.93 ± 0.03	0.97 ± 0.01	1.12 ± 0.04
4e	8.09 ± 0.09	1.010 ± 0.003	1.25 ± 0.05
4i	6.78 ± 0.02	1.026 ± 0.008	1.12 ± 0.05
5e	5.0 ± 0.3	0.98 ± 0.01	1.48 ± 0.07
5i	10.1 ± 0.1	1.00 ± 0.02	1.17 ± 0.04
6e	1.35 ± 0.06	0.78 ± 0.01	1.07 ± 0.04
6i	9.38 ± 0.05	0.97 ± 0.01	1.05 ± 0.03

to match the hardware conditions. These modifications were incorporated into the simulation during the bachelor thesis of Moritz Hornung [Hornung 2020]. The results of these simulations, expanded by an exact model of the parameter distributions obtained from the calibration routines, are presented in this section.

The adaptations are separated into three distinct parts. First, the neuron parameters are aligned with the configurations available on the hardware. Subsequently, the influence of distributed parameters on the network behavior is studied. Based on this, the different delay configurations of the connections are tested resulting in the final model which is implemented on the hardware.

Hardware Parametrization

Modeling a biologically plausible network structure, the cortical microcircuit aligns with the parameter ranges for which the BrainScaleS-1 hardware is designed. Therefore, only the synaptic time constants have to be prolonged from 0.5 ms to 2.2 ms to match their limited configurability measured in fig. 4.22. This change, in turn, influences the strength of synaptic inputs, as stimulations act on the membrane for a longer duration. To compensate this, the synaptic weights are adjusted according to

$$w^{\text{new}} = w^{\text{orig}} \frac{\tau_{\text{syn}}^{\text{orig}}}{\tau_{\text{syn}}^{\text{new}}}, \tag{6.6}$$

where “new” represents adapted parameters and “orig” the original parameters. As a result of this, even though it is distributed over a longer time span, the overall stimulation strength is approximately preserved. The resulting mean weight of each population in the final model is listed in table A.6.

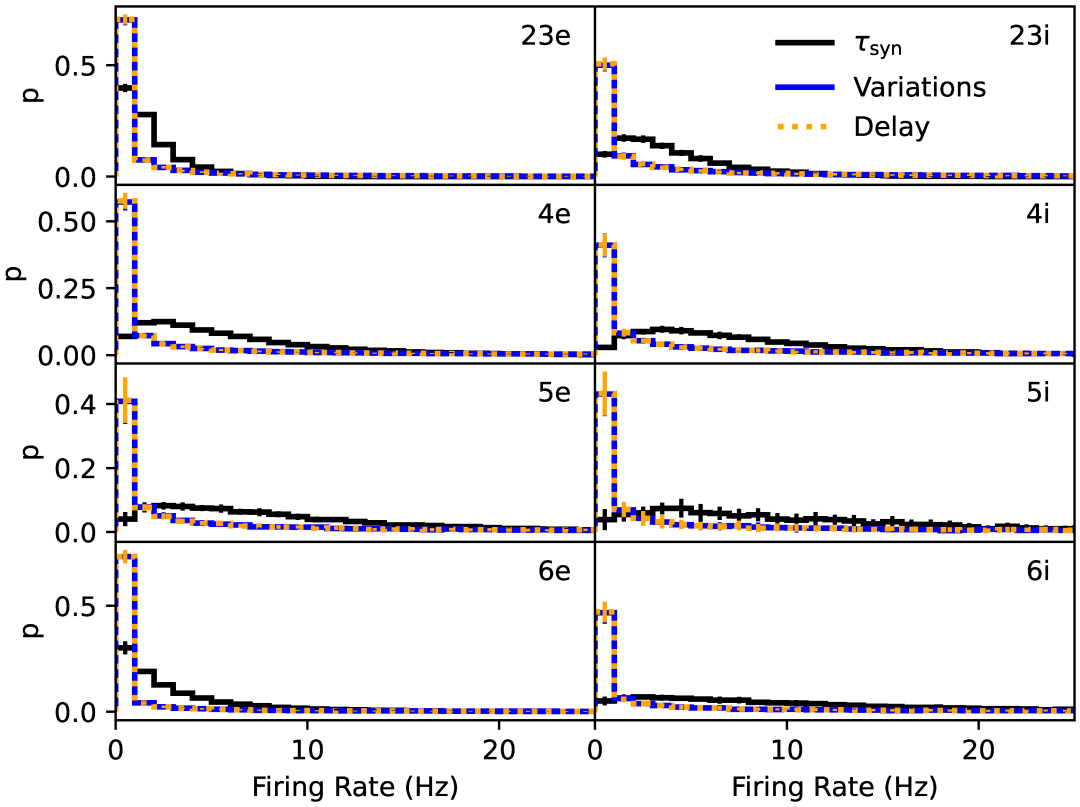


Figure 6.3.: Firing rate distributions of the NEST simulation of the cortical microcircuit model for different stages of adaptation. Each modification is added on top of the previous adaptation. Therefore, "τ_{syn}" shows the behavior of the downscaled model with conductance-based synapses under the influence of prolonged synaptic time constants. Based on this, "Variations" illustrates the effects of additionally distributed neuron parameters. Finally, "Delay" displays the final network model with identical delays for excitatory and inhibitory connections, which are described by a Gaussian distribution with a mean value of 1 ms and a standard deviation of 0.25 ms. The mean firing rates of the neurons are depicted as a histogram, with a fixed bin width of 1 ms. Additionally, the area beneath the histograms is normalized to one. Each row displays the results of a different layer of the network, with the excitatory population on the left and the inhibitory population on the right. Displayed are the mean values obtained from 30 simulations, each featuring different randomly generated connections. In simulations involving distributed parameters, in each repetition, both the connections and the neuron parameters are regenerated. The error bars represent the standard deviation across these simulations.

Figure 6.3 shows the resulting firing rate distributions of the adapted model. Comparing the results to the previous adaptation, which is illustrated in fig. 6.1, the most significant deviation is observed in the excitatory populations of layer 5 and layer 6. In both cases, the distributions are broader, accompanied by higher mean firing rates. Nonetheless, it is worth noting that the mean firing rates of all eight populations remain below 11 Hz. This trend is similarly evident in the irregularity distributions depicted in fig. A.14. There, slightly broader distributions are obtained for the same populations. At the same time, the mean values of all populations are approximately preserved. Consequently, the network behavior is still irregular with similar firing rates.

In terms of synchrony, prolonging the synaptic time constants results in either a slight reduction or no change in most populations. Notably, the only exception is the excitatory population of layer 5, which demonstrates an elevated synchrony value of 1.68 ± 0.08. Consequently, this population is situated close to the boundary value of synchronous behavior. However, as the model is further modified in the following, no additional adjustments are made to correct for this deviating behavior.

Variations

Due to circuit variations of the analog components of the BrainScaleS-1 hardware, the network behavior is examined in the presence of distributed parameters. To achieve this, the downscaled cortical microcircuit with conductance-based synapses is simulated, employing neuron parameters selected from Gaussian distributions, which roughly resemble the hardware variations. Initially, the network's behavior is studied with individually distributed parameters, applying the same standard deviations for voltages and time constants as discussed in section 5.1.4. The results of this investigation are presented in fig. 6.4. There, the impact on network behavior is measured by analyzing the variation in the mean firing rate of each population.

The observations indicate that introducing distributed values has a limited impact on the network behavior for the majority of neuron parameters. However, noticeable differences become apparent in the case of distributed threshold, reset and resting potentials. On the one hand, these potentials are associated with individual neurons. Consequently, their influence on neuron behavior is not averaged across all incoming synapses, unlike the case with distributed weights. On the other hand, along with the weights, the three potentials have the most significant influence on the firing behavior.

An increased resting potential is equivalent to a constant positive input current, while a reduction of the threshold potential lowers the stimulation required for a neuron to spike. The same principle applies to elevated reset potentials, where less stimulation is needed immediately after the refractory period. Consequently, neurons with parameters from the edges of the Gaussian distribution exhibit either higher or lower firing rates. Capped at 0 Hz, the distribution is shifted towards higher mean firing rates. However, due to the complex network structure with inhibitory connections, the effects manifest differently for individual populations, with inhibitory populations generally showing larger deviations due to their higher baseline firing rates.

In this study, the weights are not directly investigated, as they are already Gaussian

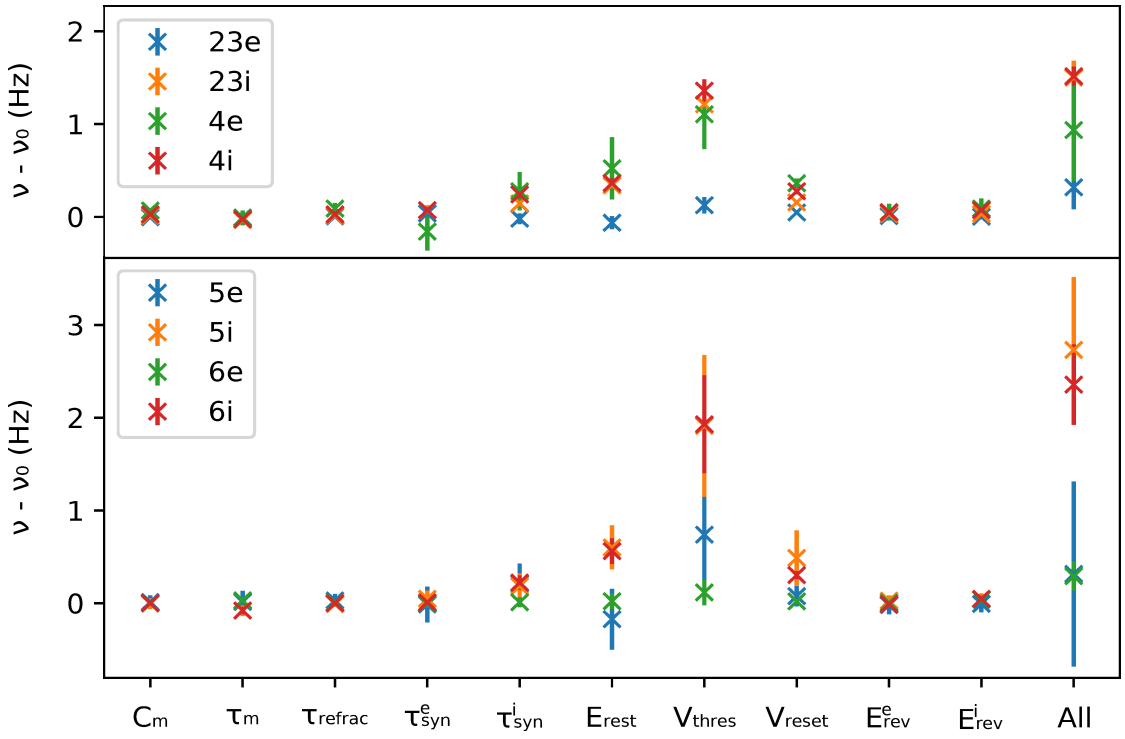


Figure 6.4.: Influence of distributed neuron parameters on mean firing rates. Presented are simulation results from the downscaled cortical microcircuit with conductance-based synapses. In these simulations, individual neuron parameters are replaced by Gaussian distributed parameters, while all other remain fixed. The mean values of these distributions are set in accordance with the values of the original model, with $E^{\mathrm{e}}_{\mathrm{rev}} = 0\,\mathrm{mV}$ and $E^{\mathrm{i}}_{\mathrm{rev}} = -100\,\mathrm{mV}$ for the reversal potentials. Furthermore, a standard deviation of 5 % for the membrane capacitance, 10 % for voltages, and 20 % for time constants is considered, all in relation to their respective mean values. Only for the reversal potentials, a standard deviation of 10 % of the difference from the resting potential is taken into account. Parameters are redrawn if the reset potential exceeds the threshold potential. For each parameter, the figure illustrates the difference of the mean firing rate relative to the mean without variations. The data points labeled "All" depict results obtained with all parameters distributed. Results are shown for all eight populations. Error bars represent the standard deviation of the measurement, obtained from 10 simulations with randomly drawn variations. Figure adapted from Hornung 2020.

Table 6.4.: Neuron parametrization of the hardware representation of the downscaled cortical microcircuit network with conductance-based synapses. Defined by Gaussian distributions, the mean and standard deviation are presented. The distributions' ranges are provided in the last column. For the resting potential and the weights, only the standard deviations are provided. Their mean values are dependent on the populations and are documented in tables A.6 and A.7. The specified percentage for the weight is relative to the respective mean value of the population.

Parameter	Mean	±	Std		Range
τ_{m}	10	±	8.0	ms	$[3\,,\infty)$
τ_{refrac}	2	±	1.5	ms	$[0\,,\infty)$
$\tau^{\mathrm{e}}_{\mathrm{syn}}$	2.2	±	0.6	ms	$[1.8\,,4]$
$\tau^{\mathrm{i}}_{\mathrm{syn}}$	2.2	±	0.4	ms	$[1.9\,,6]$
V_{thres}	−50	±	1.1	mV	$(-\infty\,,\infty)$
V_{reset}	−65	±	1.6	mV	$(-\infty\,,0.9V_{\mathrm{thres}}]$
E_{rest}	-	±	2.0	mV	$(-\infty\,,\infty)$
$E^{\mathrm{e}}_{\mathrm{rev}}$	50	±	11.1	mV	$(-\infty\,,\infty)$
$E^{\mathrm{i}}_{\mathrm{rev}}$	−150	±	1.6	mV	$(-\infty\,,\infty)$
w	-	±	50	%	$[0\,,\infty)$
C_{m}	250	±	0	pF	$[0\,,\infty)$

distributed in the original model. However, due to the limited precision of the weight calibration, as explained in section 4.2.3, it is necessary to increase the standard deviation of this distribution from 10 % to 50 % of the respective mean value. Simulation results indicate that the model remains robust in response to this adjustment, as the network behavior remains largely unchanged. This resilience is also evident in the negligible variation observed for distributed membrane capacitances. Since the time constants are held constant for each neuron, distributed capacitances result solely in a modification of neuron excitability.

In the final hardware representation, the parameter distributions are aligned with the calibration results of the hardware, which are listed in table 6.4. Similar to the parametrization of the balanced random network presented in section 5.1.4, the accuracy of the calibration varies depending on the neuron parameter. However, different parameters are obtained. On the one hand, this discrepancy arises from dedicated calibrations conducted for both models, which are necessary since the neurons in the cortical microcircuit operate with different weights. Therefore, different capacitors are utilized on the hardware to model the membrane capacitances. On the other hand, as explained in section 4.2.1, variations measured on the hardware are translated differently into the biological domain due to the distinct choice of reversal potentials.

Figure 6.3 illustrates the firing rate distribution of the model, using the presented parameterizations. As a result of the parameter variations, the distributions adopt a

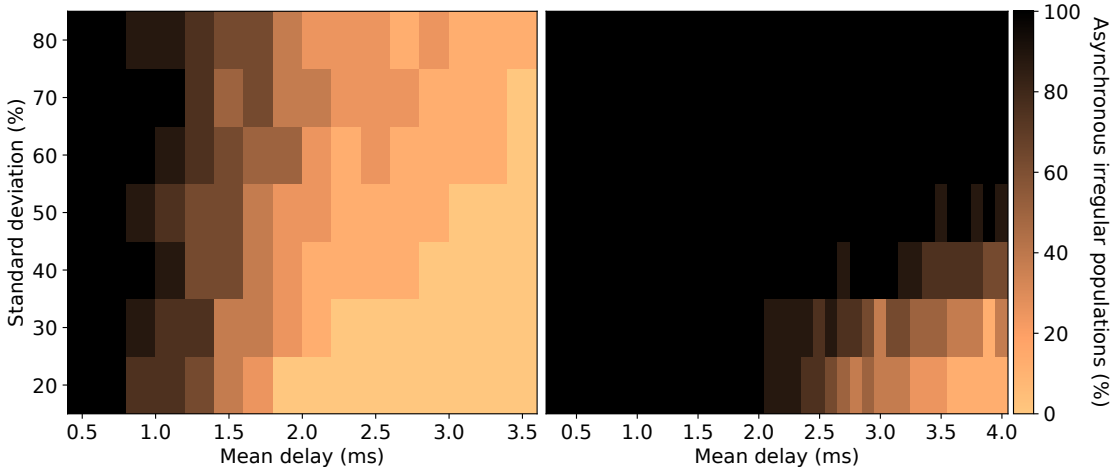


Figure 6.5.: Dependence of asynchronous irregular behavior on the distribution of delays. Different colors represent the percentage of populations within the downscaled cortical microcircuit with conductance-based synapses that show asynchronous irregular behavior, according to the definition introduced in section 6.1.1. Results are shown for various Gaussian delay distributions characterized by their mean value and standard deviation, given in percent of the respective mean value. In contrast to the original model, identical distributions are used for excitatory and inhibitory populations. On the left, results are obtained from the network without distributed neuron parameters, and on the right, with distributed parameters.

behavior similar to the original model. Reflecting increased differences among neurons within the same population, larger variations in firing rates are observed. While the distributions are once again primarily characterized by neurons with firing rates below 1 Hz, a small percentage of neurons displays higher firing rates distributed throughout the entire spectrum, constrained by their refractory period. Nonetheless, each population continues to exhibit a distinct firing pattern with characteristics of the asynchronous irregular regime, similar to the original model. This consistency persists across repetitions with randomly selected connections and neuron parameters.

Adjusting the Delay Distribution

Finally, the delays of the model are adjusted to match the restrictions of the hardware. To accomplish this, both the excitatory and inhibitory delays are configured to the same mean values, as no hardware-related distinctions are expected, given that delays are determined by the spatial separation between pre- and postsynaptic neurons. With this adjustment, the network's behavior is analysed for various delay distributions in fig. 6.5. Without distributed parameters, the percentage of populations exhibiting asynchronous

Table 6.5.: Network characteristics obtained form the downscaled NEST simulation of the cortical microcircuit model with conductance-based synapses and distributed parameters according to table 6.4. Moreover, the delays of excitatory and inhibitory connections are drawn from a Gaussian distribution with a mean value of 1 ms and a standard deviation of 0.25 ms. Different columns list the mean rates, mean irregularities and synchrony values of all eight populations. Displayed are the mean values and standard deviation obtained from 30 simulations, each featuring different randomly generated connections and parameter distributions.

Population	Rate (Hz)	Irregularity	Synchrony
23e	2.8 ± 0.3	0.81 ± 0.02	1.05 ± 0.05
23i	6.5 ± 0.3	0.86 ± 0.02	1.01 ± 0.06
4e	7.8 ± 0.6	0.85 ± 0.02	1.01 ± 0.05
4i	12.5 ± 0.3	0.89 ± 0.03	1.05 ± 0.06
5e	14 ± 3	0.86 ± 0.02	1.5 ± 0.2
5i	15.1 ± 0.7	0.82 ± 0.04	1.08 ± 0.10
6e	7.0 ± 0.8	0.81 ± 0.02	1.02 ± 0.09
6i	16.3 ± 0.5	0.82 ± 0.04	0.94 ± 0.07

irregular behavior decreases for mean delay values exceeding 1 ms. This reduction is solely attributed to changes in synchrony, which can reach values of up to 5 with higher mean delays. Since the average stimulation remains unaffected by these adaptations, the mean firing rates and irregularity values remain unaltered. Consequently, related to the findings of the balanced random network model, the choice of delay distribution plays a significant role for the correlations in the network. In the original model, asynchronous behavior is achieved through the use of different distributions for inhibitory and excitatory populations.

The simulations indicate two possibilities for preventing neuron synchronization. On the one hand, synchronous behavior diminishes as the standard deviation of the delay increases. On the other hand, the introduction of distributed parameters hinders neurons from exhibiting correlated behavior, a pattern also observed for the balanced random network model. The only exception to this behavior occurs when the mean delay approaches and surpasses the refractory period of the neurons. In this case, with and without distributed parameters, the network behavior changes, resulting in higher firing rates, increased synchrony, and reduced irregularity. However, with an expected mean delay of 1 ms and a standard deviation of 0.25 ms, the hardware implementation of the model is expected within the asynchronous irregular regime.

The network characteristics of the final model are presented in fig. 6.3 and table 6.5. It is evident that the firing rate distribution remains unaffected by the chosen delay modifications. In addition, as a result of the parameter variations, higher mean firing rates with reduced irregularity and synchrony values are observed. Furthermore, the

disparities between populations are reduced.

In conclusion, the original model's behavior is not preserved after all the necessary modification. Nonetheless, a model has been developed that exhibits biologically plausible firing characteristics similar to the original network behavior and adheres to the constraints of the BrainScaleS-1 hardware.

6.2. Implementation on BrainScaleS-1

Inspired by the structure of the human brain's cortex, the cortical microcircuit resembles as an ideal benchmark for neuromorphic hardware. To this end, it is utilized in this thesis to demonstrate the capabilities of the BrainScaleS-1 wafer-scale system. However, due to the physical modeling approach of the system, which comes with reduced flexibility in model parameters, implementing the original model is not feasible. Consequently, the model is adapted to the hardware restrictions in section 6.1, while preserving biologically plausible firing characteristics. Finally, in this section, the adapted model is emulated on the hardware and its resulting network characteristics are compared with those obtained in a NEST simulation.

This section is divided in three parts: Section 6.2.1 describes the generation of a hardware representation of the model with a focus on minimizing the number of unrealized synapses. Based on this, section 6.2.2 discusses the emulation of the resulting network description on the hardware. Following the successful emulation of the model, section 6.2.3 presents the time spent during different steps of the execution.

6.2.1. Mapping the Model to the Hardware System

Although downscaled to 10 % of its original size and with replaced external input connections, mapping the adapted cortical microcircuit to the BrainScaleS-1 hardware is a challenging task. This involves identifying neuron circuits for all of its approximately 8000 neurons and establishing up to 3 million connections among them. Given the hardware's limited resources, this high-connectivity between all neurons of the model inevitably results in connections that have no physical representation.

To address this issue, various map and route options, as detailed in section 4.3, are evaluated. There, the primary objective is not just minimizing the overall number of lost connections but also to preserve the original network structure by distributing the loss across all possible connections.

Without manual adjustments, the BrainScaleS-1 operating system automatically places neurons on the wafer following different strategies. Figure 6.6 illustrates the performance of three of these strategies across various network sizes of the the cortical microcircuit. Additionally, the results take into account different numbers of neuron circuits within each composite neuron, as introduced in section 3.1.1.

For network sizes smaller than 5 % of the original model, consistent performance is observed across all neuron sizes. However, as the network size increases, smaller neurons exhibit decreasing performance due to an insufficient number of synapses per neuron to handle the growing influx of connections.

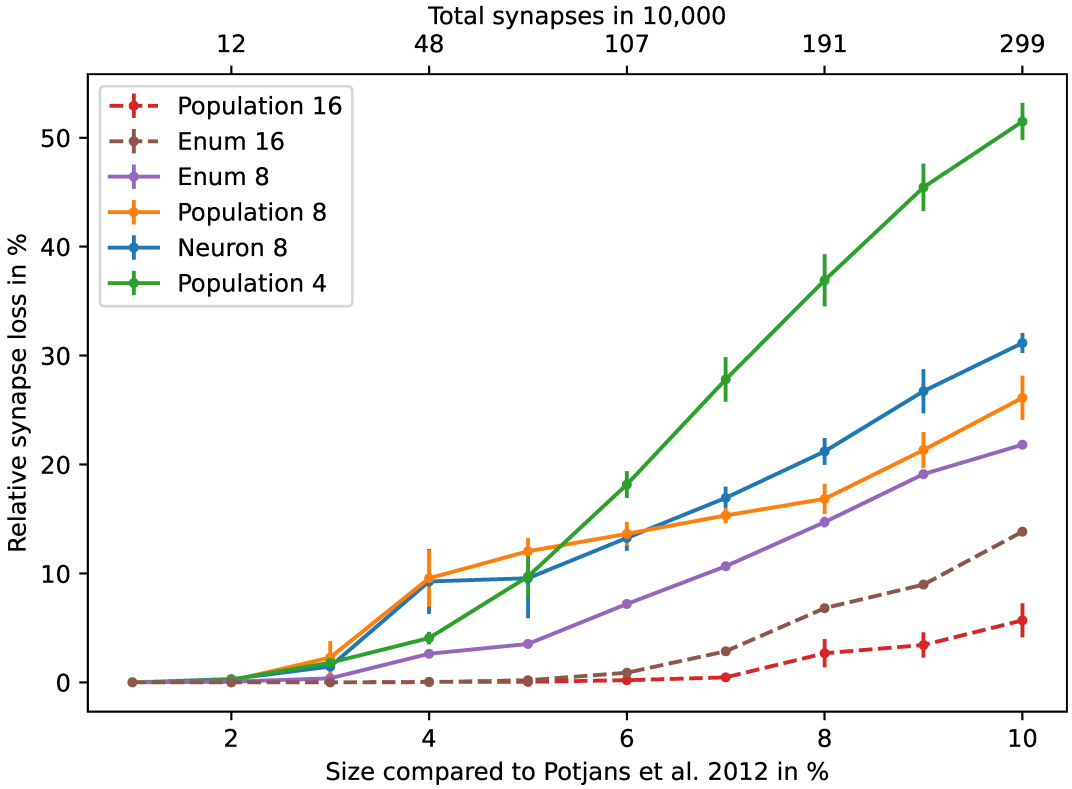


Figure 6.6.: Synapse loss of the cortical microcircuit at varying degrees of downscaling. Different colors represent distinct placement strategies, where neurons are organized according to their population's connectivity (Population), individual neuron connectivity (Neuron), or placed in ascending order (Enum), as detailed in section 4.3.1. Additionally, the numeric value associated with each line specifies the used neuron size, i.e., the number of neuron circuits within each composite neuron. The labels are arranged in ascending order of synapse loss at 10 % of the original size. For neuron sizes of 4 and 8, actual availability data from wafer 30 is taken into account. In contrast, for neuron sizes of 16, performance on a perfect wafer without missing components is demonstrated. Error bars indicate the standard deviations of 10 repetitions, each featuring models with randomly generated connections. For visibility reasons, only the results with the lowest synapse loss are presented for specific combinations of neuron size and placement strategy.

Theoretically, a neuron of size 4 comprises 880 synapses while the 10 % model of the cortical microcircuit requires a maximum of 700 synapses per neuron. However, due to routing restrictions, only a subset of these synapses can be utilized to implement the synaptic input of each neuron. This limitation arises for two main reasons. Firstly, due to the sparse connectivity matrix of the bus system and the limited number of synapse drivers that can share their inputs, as explained in section 3.1.3, only a subset of synapses are accessible from each bus. Secondly, connections compete for the available routing resources. In particular, each synapse driver is exclusively linked to a single route, and routes can only be shared by pre-synaptic neurons originating from the same HICANN. Consequently, in combination with the random connectivity of the cortical microcircuit, where pre-synaptic partners are distributed across the entire wafer, routing limitations are already reached with smaller synapse counts.

For a neuron size of 8, this performance deterioration reaches a critical point at a network size of 10 %. This becomes evident when assessing the synapse loss of different neuron sizes on a ideal wafer without excluded components (cf. section 4.1), where the loss decreases even further for larger neurons.

Nevertheless, when considering a real wafer, it is not feasible to implement even larger neurons. This limitation arises from two factors. On the one hand, due to unavailability of certain circuits, there are not enough neighboring neuron circuits to represent all neurons within the model. For example for a neuron size of 16, this limit is reached at a network size of 5 %. On the other hand, as elaborated in section 4.4.2, finite resistances are observed between connected membranes, leading to altered neuron behavior for large neuron sizes. Consequently, a neuron size of 8 and a scaling factor of 10 % is chosen for the final hardware implementation.

Furthermore, fig. 6.6 serves as a basis for evaluating the available placement strategies. These strategies are responsible for distributing neurons on the wafer and result in different utilizations of the available buses.

Given the population-specific connectivity within the model, it seems natural to cluster neurons based on their populations. However, for the targeted neuron size of 8, lower losses are observed when neurons are placed in numerical ordering. In this configuration, neurons are placed independently of the underlying network structure, closely adjacent to one another, which leads to fewer variations in losses when altering the network structure, in comparison to other placement strategies. Nevertheless, as demonstrated for neurons of size 16 on an ideal wafer, improved performance is seen with population-specific placement. This suggests that the loss reduction of the population-independent placement is a consequence of densely packing neurons in a local minimum of unavailable components on the wafer.

To circumvent such correlations with the availability data and to optimize the utilization of wafer components, the placement process is manually adjusted. This approach also affords control over routing performance and permits the distribution of lost synapses across all available connections. This is crucial because treating all neurons equally carries a high risk of losing all connections between populations with limited synapse counts.

To this end, each population is assigned to a specific group of adjacent HICANNs. These

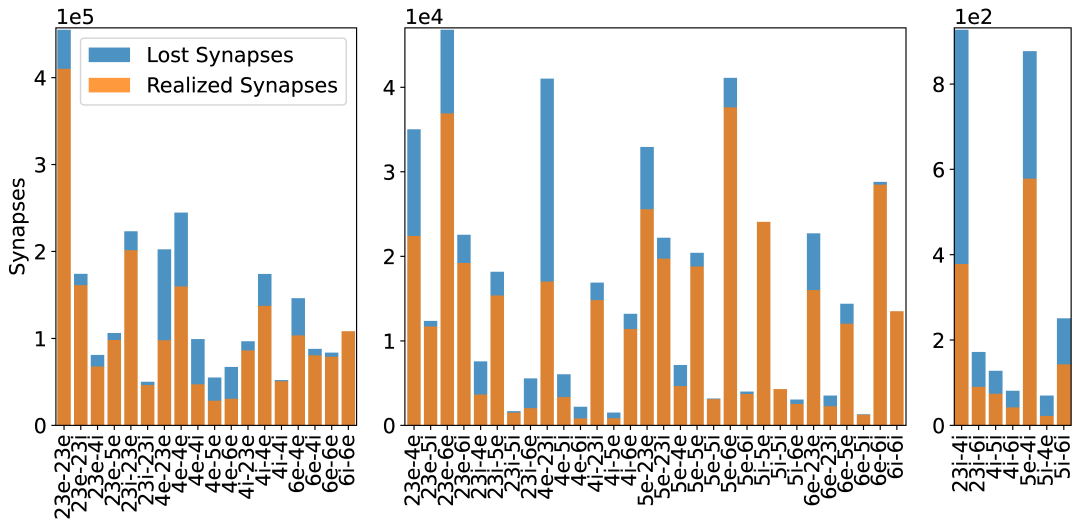


Figure 6.7.: Synapse loss per connection for the 10 % cortical microcircuit mapped to the availability data of wafer 30. Only connections containing synapses are visualized. For each connection, the total number of synapses and the number of successfully realized synapses is illustrated. To account for the significant disparities in synapse counts, the connections are categorized into three distinct ranges.

groups are randomly generated according to their population's size and are distributed over the entire wafer, prioritizing HICANNs with high routing possibilities. Subsequently, various realizations are evaluated for minimal overall and per-connection synapse loss.

Following this approach, a network structure with a synapse loss of 20.53 % is obtained. Moreover, as illustrated in fig. 6.7, no connections are entirely lost, and at most, a 68.57 % synapse loss is observed in the connections between the inhibitory population of layer L5 and the excitatory population of layer L4. Nonetheless, it is inevitable that the number of lost synapses varies depending on the specific connection.

Figure 6.8 visualizes the map and route results on the wafer, thereby illustrating the distribution of the neurons. As evident, neurons are predominantly placed in the central region of the wafer, since HICANNs at the edges have fewer routing partners. The only exception to this pattern are the HICANNs that are unavailable for placement. Excluded from the availability database, these HICANNs do not host any neurons and if necessary are not even used to route connections.

In total, the neurons are accommodated by 266 HICANNs, while 351 HICANNs are employed to route their connections. Therefore, the majority of the wafer area is utilized for the network.

While manual distribution of neurons across the wafer yields benefits such as reduced synapse loss and enhanced network control, it has the drawback of modifying the

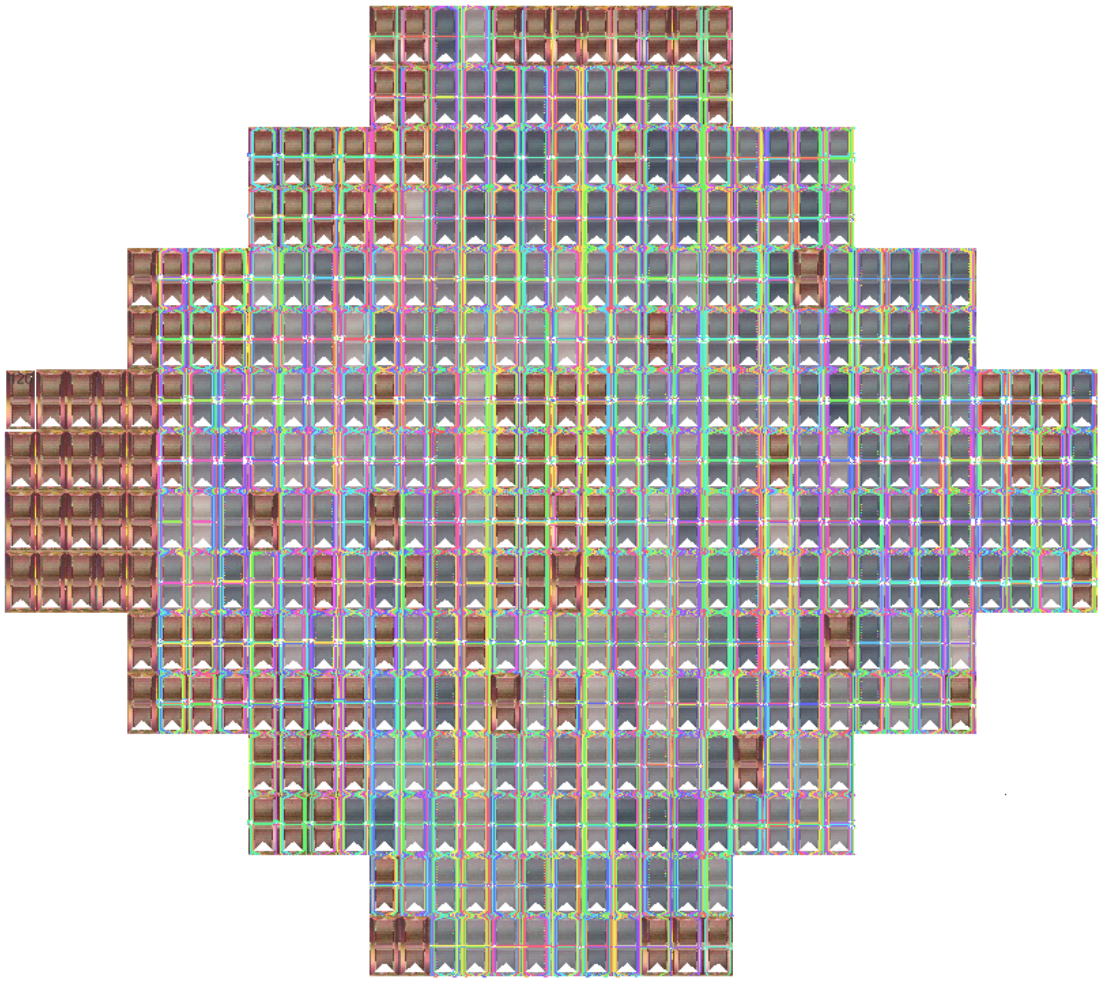


Figure 6.8.: Visualization of the map and route result of the cortical microcircuit on BrainScaleS-1. Each HICANN is depicted as a rectangle with a white triangle at the bottom. Neuron placement is represented by the use of blue coloration, with darker shades indicating higher neuron counts. Connections are visualized as colored lines routed along the edges of the chips.

network's delay characteristics. Clustering neurons in distinct regions of the wafer leads to an increase in large delays. As a result, instead of the approximately Gaussian-distributed delay values found in the automatically placed balanced random network, a skewed distribution towards larger values is observed, which is visualized in fig. A.11a. Additionally, as neurons are grouped according to their associated populations, smaller delay values are expected between neurons within the same population.

To account for these delay adjustments and the altered network structure arising from the loss of synapses, the software simulation is extended to incorporate the precise routing results obtained from the hardware, as discussed in section 6.2.2. Hence, despite the changed network structure, this approach facilitates a thorough comparison between simulation and emulation results.

6.2.2. Emulation on BrainScaleS-1

This section addresses the emulation of the downscaled cortical microcircuit on the BrainScaleS-1 hardware. A single wafer system is configured according to the network description obtained in the preceding section and the network is emulated for 60 s of biological time. Subsequently, the spike times of the neurons are retrieved from the system and a comparison is drawn between the network characteristics obtained from the hardware emulation and the results of the NEST simulation of the adapted model, as discussed in section 6.1. However, as highlighted in the previous section, an altered network structure is expected in the hardware representation of the model, attributed to the loss of synapses and adjusted delay values. Since these changes are not accounted for in the adapted model, a final adjustment is introduced, incorporating the projected network structure of the hardware. Using this model, various weight parameterizations are simulated to identify suitable adjustments to compensate the modified network structure. Furthermore, executing the same experiments on the hardware enables a comparison of resulting network characteristics for different parametrizations.

To integrate the hardware representation into the NEST model, the delay values for each realized connection in the map and route results are assessed based on the delay calibration discussed in section 4.2.5. Since the calibration does not encompass circuit variations, the obtained values are further smoothed using a Gaussian distribution with a standard deviation of 0.1 mV. The resulting delay distribution is depicted in fig. A.11b. Following this, the extracted connection results are loaded into the NEST model and the network is simulated with the same structure and delay values as expected on the hardware.

Given the altered structure of the model, the network characteristics undergo changes. To account for this, the network is simulated with varying excitatory and inhibitory weight factors introduced in the downscaled network model. Subsequently, the resulting network characteristics are evaluated for asynchronous irregular behavior, according to its definition in section 6.1.1. The outcomes of these measurements are illustrated in fig. 6.9.

In the model without synapse loss, asynchronous irregular firing is observed with an excitatory factor of 0.7 and an inhibitory factor of 1.4. Consequently, the results

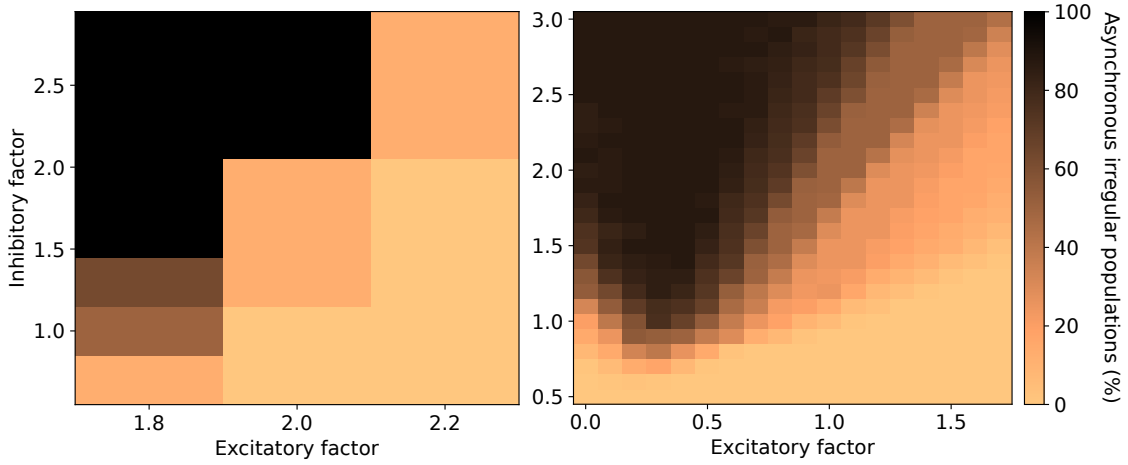


Figure 6.9.: Dependence of asynchronous irregular behavior on the excitatory and inhibitory weight factors. Different colors represent the percentage of populations within the cortical microcircuit that exhibit asynchronous irregular behavior, according to the definition introduced in section 6.1.1. Results are presented for different factors applied to all excitatory and inhibitory weights. On the left, findings are displayed from the network emulation on the BrainScaleS-1 hardware, while on the right, results are shown from the adapted NEST simulation with incorporated map and route results. The illustrated values represent the mean values of 10 repetitions with fixed network topologies but randomly generated parameter variations.

indicate that in the presence of lost synapses, achieving similar firing patterns requires stronger inhibitory and weaker excitatory weights. This effect is expected, since in the adapted model the excitatory external input is replaced by an increased resting potential that is not affected by synapse loss. Therefore, excitation predominates in the network with synapse loss, which is compensated by elevated inhibitory weights. As a result, asynchronous irregular behavior is found in most populations for excitatory factors below 0.5 and inhibitory factors above 1.5.

Considering the already four times larger inhibitory weights of the original model, the resulting parametrization poses a challenge due to the limited configurability of the hardware. For instance, an excitatory factor of 0.2 and an inhibitory factor of 2.8 correspond to a weight range from $w/C_\mathrm{m} = 1.34\,\mathrm{s}^{-1}$ to $w/C_\mathrm{m} = 110.39\,\mathrm{s}^{-1}$. In comparison, the minimum and maximum weights that, according to the weight calibration, can be implemented on the hardware range from $w/C_\mathrm{m} = 5.82\,\mathrm{s}^{-1}$ to $w/C_\mathrm{m} = 96.73\,\mathrm{s}^{-1}$. However, as discussed in the section 4.2.3, the lower bound of the hardware weights is not addressed by the weight calibration. In this regime, the synaptic current mainly arises from charge stored in parasitic capacities residing in the synaptic input line, which also prevent the setting of the weight to zero. Due to the weak effect of the synapses there, the noise of the membrane, added by the analog readout, surpasses the measured PSP heights. Therefore, for specific configurations, smaller weights are anticipated although they cannot be directly evaluated.

To address this limitation, similar to the software simulation, the network behavior of the emulation is evaluated for different weight factors, as depicted in fig. 6.9. Here, an excitatory factor of 1.7 corresponds to the minimum weight configuration possible on the hardware. By employing this minimal weight configuration, asynchronous irregular behavior is found in all populations for inhibitory weight factors exceeding 1.5. Moreover, the observed network behavior aligns notably with the characteristics of the software model, under the assumption of reduced excitatory weights on the hardware.

Given the close relationship between hardware and software implementation, the obtained results serve as a basis for estimating the excitatory weight factor of the hardware. To this end, the rate distribution of the hardware is measured under a fixed configuration featuring minimal excitatory weights and an inhibitory weight factor of 2.8. Following this, the bin heights of this distribution are compared with simulation results obtained across various weight factors. The outcomes of this analysis are presented in fig. 6.10.

Best matching rate distributions are obtained for an excitatory weight factor of 0.2 and an inhibitory factor of approximately 2.8. Further reducing the excitatory weights leads again to distinct network behavior. This suggests, that the minimal weight that can be set on the hardware corresponds to approximately $w/C_\mathrm{m} = 1.34\,\mathrm{s}^{-1}$. Moreover, since the inhibitory factors align in both implementations, it demonstrates the capabilities of the weight calibration for weights that are not too small.

Figure 6.11 illustrates the resulting distributions of neuron firing rates of both the emulation and simulation, using best matching parametrizations. Compared to the NEST simulation without incorporated synapse losses, modified network behavior is observed. Strongest deviations are apparent in the excitatory population of layer 4. As

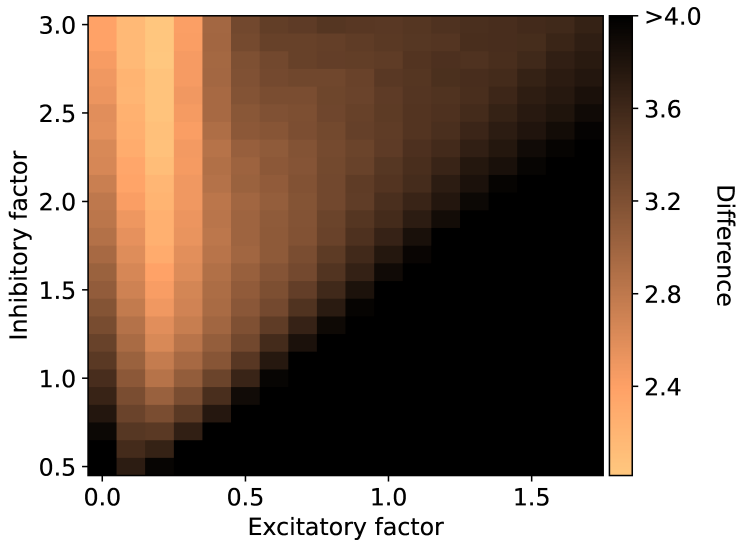


Figure 6.10.: Firing rate deviation between emulation and simulation results for different weight factors. Different colors represent the disparity in the firing rate distribution between the hardware emulation with fixed weights (excitatory = 1.8, inhibitory = 2.8) and the adapted NEST simulation with incorporated map and route results for various excitatory and inhibitory weight factors. The deviation is calculated as the sum of the absolute differences of the bin heights of the normalized firing rate histograms, using a fixed bin width of 1 ms. To enhance the resolution of small variations, all values with differences larger than 4 are represented by the same color. While the hardware results are derived from the mean values of 30 emulations, for each weight factor, the mean values of 10 simulations with fixed network topologies but randomly generated parameter variations are considered.

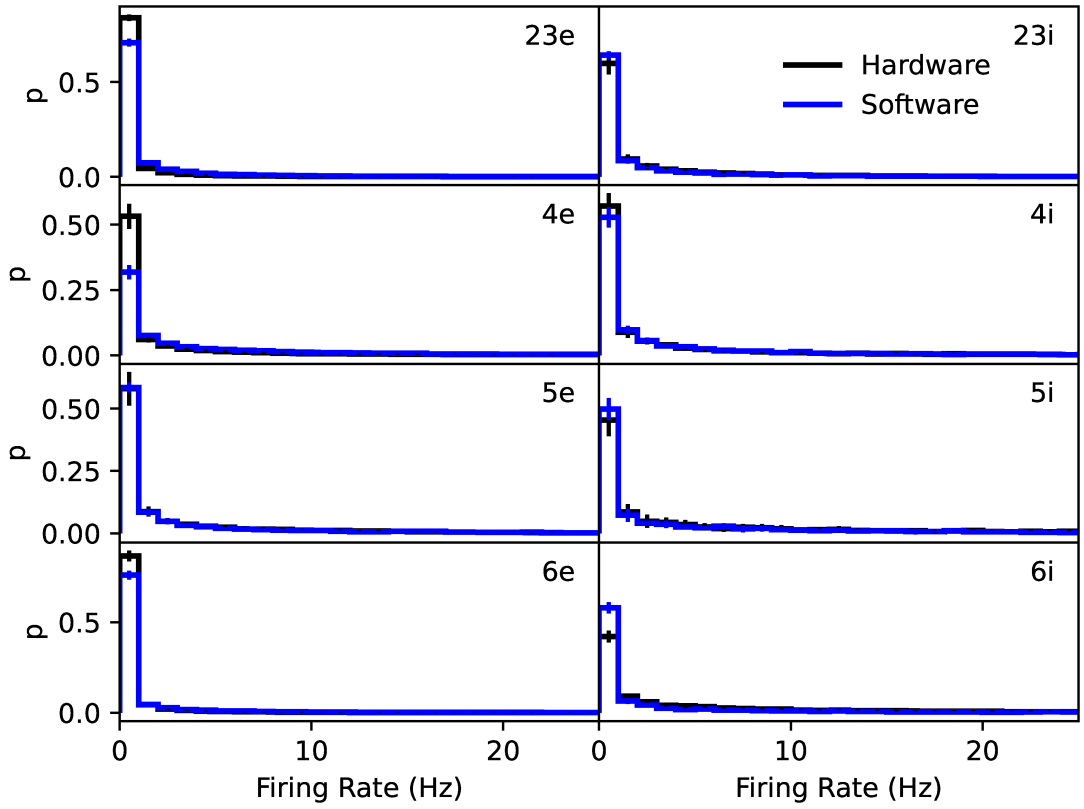


Figure 6.11.: Firing rate distributions of the hardware emulation and software simulation of the adapted cortical microcircuit with identical network topologies. While in both implementations an inhibitory weight factor of 2.8 is used, the excitatory weight factor is set to 0.2 in the simulation and 1.8 in the emulation. The mean firing rates of the neurons are depicted as a histogram, with a fixed bin width of 1 ms. The area beneath the histograms is normalized to one. Each row displays the results of a different layer of the network, with the excitatory population on the left and the inhibitory population on the right. Displayed are the mean values obtained from 30 repetitions. In the case of the emulation, the floating gates are reconfigured in between consecutive executions. For the software implementation, each simulation features different randomly generated parameter variations. The error bars represent the standard deviation across these repetitions.

demonstrated in fig. 6.7, large parts of this population's incoming internal connections are lost in the hardware representation. Furthermore, due to its strong external stimulation, it is characterized by one of the highest resting potentials. Therefore, the global adjustments of the weights cannot mitigate the modified network behavior, and higher mean firing rates, as depicted in table 6.6, are observed. As a result of this, the population is no longer classified asynchronous irregular. In contrast, although exhibiting modified behavior, all other populations remain within the asynchronous irregular regime.

In comparison to the hardware emulation, with its network characteristics shown in table 6.7, higher mean firing rates are observed in the final NEST simulation. This is primarily attributed to a small percentage of neurons with firing rates above 25 Hz in both models, demonstrating even higher values in the simulation. According to the bandwidth considerations outlined in section 5.2.1, the anticipated loss of spikes during readout is expected to be negligible for the network topology and firing rates of the cortical microcircuit. Therefore, the increased rates are traced back to the saturation of the synaptic input lines, as detailed in section 4.4.4. As already demonstrated in the analysis of the balanced random network model in section 5.2.3, this effect restricts the maximum firing rate achievable by neurons on the BrainScaleS-1 hardware.

This behavior is also evident in the irregularity distribution of the excitatory population of layer 4, as illustrated in fig. A.15. A subset of neurons within this population exhibits elevated firing rates. Given the strong stimulation of these neurons, their firing behavior is relatively independent of inhibitory inputs, resulting in a firing activity characterized by small irregularity values. In contrast, on the hardware, the maximum achievable stimulation is limited due to the saturation effect. As a consequence, these neurons are still influenced by inhibitory inputs, leading to a firing behavior characterized by intermediate irregularity values.

Furthermore, additional deviations in network behavior between the two implementations can be attributed to hardware effects that have not been included into the NEST model. For example, the reduced weight resolution of the excitatory weights on the hardware is not represented in simulation. As a result of this, weight deviations between individual populations, especially the duplication of the weight between the excitatory populations of layer L4 and L23 cannot be represented on the hardware. Moreover, correlations between synaptic inputs, arising from shared circuits on the hardware are not considered. These correlations might be the reason for elevated synchrony values that are measured on hardware. Finally, despite the efforts to enhance repeater stability, as discussed in section 4.3.2, an average of 0.5 % of the repeaters remain unlocked during emulation. Therefore, connections routed via these repeaters are considered unreliable, potentially transmitting inaccurate spike signals.

Nevertheless, when comparing the observed deviations with the model's unstable behavior during the adaptations, the observed differences in network behavior are minimal in most populations. Both implementations exhibit similar firing rate distributions in the regime below 25 Hz across the majority of populations. Additionally, a comparison of the behavior among individual populations reveals analogous characteristics in both models. These similarities are also evident when examining individual spike times, as shown in fig. 6.12. Neurons within the same population display various firing patterns

Table 6.6.: Network characteristics obtained from the NEST simulation of the adapted cortical microcircuit with incorporated map and route results. Different columns list the mean rates, mean irregularities and synchrony values of all eight populations. Displayed are the mean values and standard deviation obtained from 30 simulations, each featuring different randomly generated parameter variations.

Population	Rate (Hz)	Irregularity	Synchrony
23e	4.5 ± 0.4	0.81 ± 0.01	0.77 ± 0.06
23i	3.6 ± 0.2	0.84 ± 0.02	0.96 ± 0.05
4e	38.0 ± 0.7	0.72 ± 0.02	0.59 ± 0.04
4i	5.9 ± 0.1	0.92 ± 0.02	0.94 ± 0.06
5e	5.4 ± 0.6	0.88 ± 0.02	1.17 ± 0.06
5i	8.5 ± 0.3	0.89 ± 0.04	0.89 ± 0.02
6e	4.2 ± 0.6	0.87 ± 0.04	0.90 ± 0.09
6i	8.9 ± 0.4	0.90 ± 0.04	0.83 ± 0.07

Table 6.7.: Network characteristics obtained form the hardware emulation of the down-scaled cortical microcircuit. Different columns list the mean rates, mean irregularities and synchrony values of all eight populations. Displayed are the mean values and standard deviation obtained from 30 emulations with reconfigured floating gates in between consecutive executions.

Population	Rate (Hz)	Irregularity	Synchrony
23e	2.2 ± 0.2	0.83 ± 0.02	0.91 ± 0.04
23i	3.2 ± 0.2	0.95 ± 0.02	1.03 ± 0.04
4e	15 ± 2	0.79 ± 0.03	0.82 ± 0.04
4i	4.7 ± 0.3	0.94 ± 0.03	1.07 ± 0.05
5e	5 ± 1	0.92 ± 0.03	1.21 ± 0.08
5i	6.7 ± 0.8	0.94 ± 0.04	1.19 ± 0.05
6e	1.0 ± 0.7	0.83 ± 0.02	1.03 ± 0.03
6i	7.04 ± 0.09	0.96 ± 0.02	1.03 ± 0.04

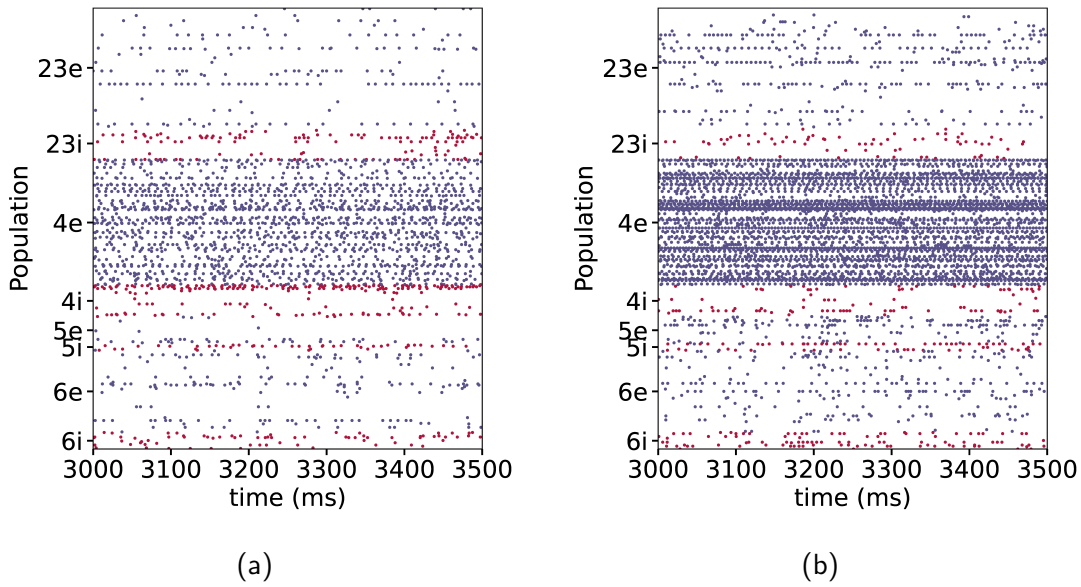


Figure 6.12.: Spike times in (a) the hardware emulation and (b) the NEST simulation of the adapted cortical microcircuit with identical network topologies. Displayed are the spike times of the initial neurons in each layer. The number of presented neurons is chosen relative to the size of their population. Inhibitory neurons are depicted in red and excitatory neurons in blue.

primarily characterized by irregular and asynchronous activity.

Moreover, obtained results are stable with respect to variations of the neuron parameters. This stability is exemplified on the hardware through repeated emulations, involving reprogramming of the floating gates in between. Despite the resulting write-cycle to write-cycle variability in the stored neuron parameters, all emulations consistently exhibit similar firing characteristics.

Therefore, despite subtle differences, the extracted model can be considered as an accurate representation of the hardware. Moreover, by successfully preserving the asynchronous irregular behavior across all populations, the results demonstrate a successful emulation of the downscaled cortical microcircuit on the BrainScaleS-1 system. Additionally, the stability and adaptability of the results for different parameterizations highlight the robustness and accuracy of the emulation.

6.2.3. Temporal Analysis of the Hardware Execution

Having successfully emulated the downscaled cortical microcircuit with biologically plausible firing patterns on the BrainScaleS-1 system, the subsequent analysis focuses on the time allocated for various stages of the hardware execution. A analysis of the performance of the emulation in comparison to other simulators is presented in chapter 7.

Table 6.8.: Time spend in different steps during the hardware execution of the downscaled cortical microcircuit emulated for 60 s of biological time. Presented are the mean values and standard deviations based on 10 experiment repetitions for both loaded and newly generated map and route results. Results are obtained for the manually adjusted placement, where neurons are positioned in ascending order. Additionally, a maximum of 5 cycles is utilized for repeater re-locking. The host computer employed to generate the map and route data and to communicate with the FPGAs is equipped with an Intel(R) Core(TM) i7-4771 CPU featuring 8 threads. This CPU operates at a base clock frequency of 3.5 GHz and can achieve 3.9 GHz in its boost mode.

Step	Time			
Map and route	35.9	±	0.6	min
Loading results	14	±	0.6	min
Configuration	78	±	2	s
Re-Locking	240	±	30	s
Execution	6.66	±	0.09	ms
Retrieving data	150	±	30	ms

Table 6.8 illustrates the time spent in various steps during the experiment execution.

For a newly generated network model, the predominant portion of the execution time is dedicated to generating the network description. In this phase, half of the time is allocated for placing neurons, while the remaining half is devoted to route connections. Since the process is not time-critical and prioritizes the generation of reliable results, it is neither multithreaded nor optimized for execution speed.

Once generated, existing network descriptions can also be loaded from disk. However, still under development, it is not yet optimized for performance and results are currently stored in human-readable XML format. Consequently, while this loading process more than cuts the execution time in half, it still demands a significant amount of time.

As the executions transition from the host computer to the hardware system, there is a decrease in execution times. Nevertheless, as explained in section 3.1.1, programming the floating gates of the system is a time-consuming process, contributing to the measured duration of the configuration. Moreover, improving routing reliability through repeater re-locking, as introduced in section 4.3.2, comes at the cost of execution time. Full wafer tests must be repetitively executed and evaluated up to five times. Generally, this step can be skipped; however, for the purpose of achieving the highest network reliability, it is opted for in this case.

Once configured, the hardware continuously emulates the desired network behavior. The speedup of the system is crucial in this context, as 60 s of biological time corresponds to a fixed duration of 6 ms during emulation. The additional time displayed is spent at the beginning of the experiment to communicate with the system, synchronize the FPGAs, and initiate the recording process.

Due to the time constraints during emulations, recorded spikes are stored in the memory of the FPGAs. Consequently, as a final step, after the predefined execution time, the results must be retrieved from the hardware and made available on the host computer.

Given the independence of emulation speed from the network size, hardware executions demonstrate an advantage for large-scale experiments. Furthermore, since only the execution time scales with the experiment duration, the system benefits from either long or repetitive experiments that do not require reconfigurations of the floating gates. Therefore, the current hardware implementation serves as a platform enabling researchers to explore long-term neural developments, as years of biological network dynamics can be emulated in hours of wall-clock time.

7. Discussion and Outlook

The field of neuromorphic computing addresses the limitations of conventional computers in terms of simulation speed and power consumption by drawing inspiration from operational principles observed in the human brain. To validate these assumptions and prove that the systems are functioning, benchmarks are required [Davies 2019; Yik et al. 2023]. In the context of brain inspired implementations, the balanced random network [Brunel 2000] emerges as a natural choice for this task. Through the exploration of various firing characteristics in spiking neural networks, this model provides the basis for several biologically plausible network structures. This also applies to the cortical microcircuit, as presented in Potjans et al. 2012, which has become a quasi-standard benchmark for neuromorphic computing in recent years [Ostrau et al. 2022].

With a focus on the first-generation wafer-scale neuromorphic hardware platform BrainScaleS-1, introduced in chapter 3, this thesis concludes long-standing efforts by demonstrating the system's capabilities through the emulation of both models. Accomplishing this demonstration is inherently challenging, given the constraints imposed by the physical modeling approach of the hardware. To overcome these limitations, it is crucial not only to ensure reliable hardware performance but also to develop a comprehensive understanding of its behavior. This was achieved through the optimization of the hardware operation for large-scale experiments, as discussed in chapter 4.

As a basis for all hardware operations the systems had to be transitioned into a reliable substrate for experiments. This is necessary due to the employed wafer-scale integration, which limits flexibility in addressing malfunctioning components as individual problematic chips cannot be replaced. To this end, the availability management was developed, as introduced in section 4.1. In comparison to state-of-the-art denylisting, where in general groups of components are tested and removed as a whole, the tests are executed for each circuit individually. In combination with the modular structure of the systems, this allows for a minimalistic exclusion of malfunctioning components. This is necessary as each circuit is valuable facing the complexity of large-scale biological networks. Moreover, all circuit interdependencies and unstable behaviors observed in the hardware are taken into consideration. Therefore, the results enable the user to treat each system as a idealized substrate for experiments. Furthermore, at the expense of functioning components, circuits that display undesired analog behavior can be excluded, thereby facilitating predictable network results.

Hardware reliability was further enhanced by extending the existing calibration framework tailored to the requirements of biologically inspired network models, as detailed in section 4.2. In pursuit of this objective, utilizing the scale invariance of the LIF neuron, an automated parameter translation was implemented, enabling the configuration of model parameters within the biological regime. By maximizing the dynamic range of the

neurons in the subthreshold regime, it minimizes the influence of noise observed on the membrane.

As a consequence of this translation, voltages are configured based on the reversal potentials. For this reason, their calibration routine was revisited and a new indirect measurement method was introduced specifically for the excitatory reversal potential, leading to more accurate results. Moreover, the operational limits of the circuits were identified and integrated into the automated parameter translation, mitigating the otherwise observed saturation effects.

Given the necessity to precisely define the synaptic efficacy in the investigated networks, a weight calibration was implemented. During this process, the insufficient capabilities of the used analog readout system became evident. While a full-wafer characterization of neuron parameters takes approximately 48 h and is thus already performed with reduced precision, attempting a per-circuit calibration for the large parameter space of the synaptic input generation would exceed practical runtimes. Moreover, restricting the weight calibration exclusively to digitally configurable parameters, which can be programmed more rapidly, proves unfeasible due to the networks' requirement for an extensive weight range that demands the full dynamic range of the hardware.

Therefore, to overcome these limitations, a per-wafer calibration was introduced. While this method does not offer the possibility to correct circuit variations, it does allow for the assessment of the average corresponding biological weight value for each configuration. In addition, an algorithm was developed to configure the synapses based on the biological weights of the models, considering parameter-specific interdependencies. In section 6.1.4, it was demonstrated that the investigated networks are robust with respect to weight deviations, as long as the mean is preserved. Since it could be shown in section 4.2.4 that the calibration and configuration, in good approximation, preserve the mean values of the weights, the weight calibration is considered suitable for the networks. However, it should be noted that the calibration exhibits limitations for small weight values, which cannot be accurately assessed due to the relatively larger voltage fluctuations introduced by the analog readout system in comparison.

To tackle the immense connection complexity of the biological models, the map and route algorithms of the systems were improved, as introduced in section 4.3.1. In section 6.2.1, this has proven crucial for preserving a desired network structure despite the limited routing capabilities and the subsequent loss of connections. Furthermore, as demonstrated in section 4.3.2, it was shown that iterative re-locking of the on-chip repeater circuits improves routing reliability.

Additionally, as presented in section 4.4, operational limitations of the hardware were identified, and it was demonstrated that, albeit at the cost of configurability, their impact can be minimized through appropriate hardware operation.

Building upon the introduced enhancements in hardware management, the reliability of the systems was significantly improved. However, the networks under investigation still require adjustments to accommodate inherent variations of the hardware that cannot be avoided. This involves the reduction of neuron and synapse counts, modification of the synapse model, and the incorporation of distributed neuron parameters and transmission delays aligned with those derived from the acquired hardware model.

To achieve this, software simulations were conducted for both models using the NEST simulator. As discussed in Albada et al. 2014 and demonstrated in section 5.1 and section 6.1, the behavior of the networks is found to be unstable with respect to changes. From this, two insights arise. On the one hand, the idealized network assumptions of the two models are not well-suited to serve as benchmarks for analog neuromorphic hardware. Because of the reduced flexibility associated with the physical modeling approach, the models' behavior is not stable enough with respect to network variations. Furthermore, evaluating the internal dynamics of the hardware becomes challenging when dealing with a large-scale network right from the outset. Consequently, scalable models are desirable. On the other hand, according to Albada et al. 2014, the cortical microcircuit represents the smallest network description with biologically plausible connection probabilities and synapse counts. This demonstrates the necessity of developing even larger hardware systems in the future.

To overcome the limited flexibility of the models, the evaluation was focused on preserving biological firing characteristics, characterized by asynchronous irregular behavior as defined in Potjans et al. 2012. Based on the NEST simulations, techniques were developed that allow for obtaining networks in all states of adaptation, all of which preserve the desired characteristics. These results enable benchmarks for less flexible systems and were utilized to demonstrate the capabilities of a single BrainScaleS-1 wafer.

As presented in section 5.2, the adapted balanced random network model, comprising 2083 neurons and 690 157 synapses, was emulated on the hardware. While in the asynchronous irregular regime comparable network behavior to the software simulation is obtained, the regime with high firing rates reveals the limitations of the neuron circuits concerning biologically implausible spike patterns. Nevertheless, by observing typical firing characteristics in this regime, the system's immense capabilities in spike transmission with up to 1×10^{12} synaptic events/s are demonstrated.

Moreover, section 6.2 illustrates the successful emulation of the adapted cortical microcircuit on BrainScaleS-1. Comprising 7712 neurons and 2 373 933 synapses, this implementation represents approximately 10 % of the model's original size. Based on the obtained results, performance comparisons are drawn between the hardware and best-performing implementations of the full-scale model across various simulators. Depicted in table 7.1, size-independent measures are employed to account for the different investigated network sizes. Additionally, limited by the processing speed in smaller networks and the overhead introduced in distributing spike signals among all-to-all connected neurons in larger networks, no significant performance improvements are expected in the presented implementations for adapted network sizes with comparable network structure [Kauth et al. 2023]. Therefore, the presented results demonstrate, to the best of one's knowledge, the fastest operation in terms of synaptic events per second in a network with the complexity of the cortical microcircuit.

It should be noted that the performance stems from the large speedup factor of 10 000 during emulation, which remains independent of the implemented network size. Operational overhead that is introduced by the configuration of the system and the transmission of recorded spikes is not incorporated into the presented results. As a consequence of this, the speedup of the system can only be exploited during long or repetitive emulations

Table 7.1.: Performance and energy comparison on the basis of the network structure of the cortical microcircuit for different simulators. The energy estimation of the BrainScaleS-1 system is based on the maximum possible power consumption of 2 kW of the entire system, with the actual power consumption expected to be considerably lower due to the application of typical safety margins.

Simulator	Performance (synaptic event/s)	Energy (µJ/synaptic event)
BrainScaleS-1[1]	162×10^9	<0.012
NeuroAIx-Framework[2]	$19 \times 10^{9*}$	0.048
CsNN[3]	$3.8 \times 10^{9*}$	0.783
NEST[4]	$1.8 \times 10^{9*}$	0.48
SpiNNaker[5]	0.9×10^9	0.6

References: [1]Emulation results of this work, [2][Kauth et al. 2023], [3][Heittmann et al. 2022], [4][Kurth et al. 2022], [5][Rhodes et al. 2020].
[*] Values are estimated from the reported speedup factor and the network behavior of the full-scale model with external Poisson inputs.

of large-scale networks. Nevertheless, in combination with the comparable low energy consumption during emulation, this thesis succeeds in demonstrating the advantages of the physical modeling approach of the system.

Outlook

In this thesis, the capabilities of the BrainScaleS-1 system have been showcased. The successful emulation of large-scale networks highlights the feasibility of wafer-scale integration in the field of neuromorphic computing. The numerous parallel operational components support a modular and, therefore, robust system design. Furthermore, the study reveals that with precise hardware operation, the drawbacks associated with the physical modeling approach can be effectively mitigated. However, the system's inherently limited flexibility still necessitates modifications to the biologically inspired network descriptions. Therefore, future implementations should prioritize addressing and enhancing this aspect.

An initial and essential step in this direction involves rectifying the identified operational limitations, as discussed in detail in section 4.4. Moreover, concerning system reliability, eliminating inter-chip dependencies enables a more minimalistic exclusion of hardware defects. In the context of calibration, the limitations in observations mainly based on the membrane potential and spike output of neurons often lead to compromises in precision. Therefore, additional observables, such as integrating test data outputs into all repeater circuits, would be desirable.

One immediate strategy for enhancing the existing BrainScaleS-1 system is by utilizing the improved analog readout developed in Ilmberger 2017. By extending the parallel readout capabilities from 12 to 96 channels at reduced noise, the implementations would

benefit from more precise calibration results.

Furthermore, transitioning implementations to the second-generation BrainScaleS-2 chip [Schemmel et al. 2021] promises several enhancements. Firstly, it introduces a per-neuron parallel membrane readout, enabling circuit-specific weight calibration based on the presented algorithms. Additionally, reduced parameter variations are anticipated due to the shift from a floating gate based storage to a digital solution [Billaudelle et al. 2022]. Finally, the chip allows for the implementation of current-based synapses, thereby minimizing the need for extensive model adaptations.

Although currently utilized as a single-chip solution, the possibility to interconnect individual chips via the EXTOLL network was recently demonstrated in Thommes 2023. With comparable neuron and synapse counts per chip to BrainScaleS-1, it becomes feasible to implement the presented 10 % representation of the cortical microcircuit on such an interconnected system.

To facilitate the integration of even larger networks in the future, such as the full-scale microcircuit model, more extensive systems will be necessary. This could be achieved by interconnecting multiple wafer-scale systems. Nevertheless, the findings of this thesis emphasize that, with the current implementations, larger circuit counts alone are not sufficient. Given the limitations in synapses per neuron, as demonstrated in sections 4.4.2 and 6.2.1, such a system would not cope with the increasing demand for connections. Moreover, even if the implementation of larger synapse counts per neuron were feasible, the current network sizes already reveal insufficient routing capabilities due to limited flexibility and the restricted number of injection points into the synapse array. Therefore, apart from scaling up the systems, it becomes imperative to enhance on-chip routing capabilities for future large-scale implementations.

Furthermore, future implementations on the BrainScaleS-1 system would benefit from employing different models. Evaluating network statistics for predefined neuron configurations poses a challenge due to the system's inherent parameter variations. Similar to its biological archetype, the human brain, the system benefits from synapse-specific weight updates to learn specific tasks. This approach enables the reduction of parameter variations through training. One suitable choice for this could be evolutionary algorithms, which, given their large timescale, would take advantage of the speedup factor of the system.

Finally, the constant emulation speed provided by the physical modeling approach presents a clear advantage when handling more complex neuron models. In line with this, the BrainScaleS-2 system facilitates the emulation of multi-compartment neurons [Kaiser et al. 2022]. Expanding on this capability, the demonstration of future large-scale interconnected single-chip or wafer-scale systems holds great potential for further highlighting the advantages of neuromorphic hardware over conventional computers.

A. Appendix

A.1. Model Parameters

This section highlights additional model parameters utilized in this thesis. All software simulations are conducted with the NEST simulator, and network descriptions are formulated through the PyNN interface. The *pyNN.IF_curr_exp* model is employed for current-based synapses, while the *pyNN.IF_cond_exp* model is utilized for conductance-based synapses.

Neuron Stimulated by a Single Synapse

Table A.1.: Parameters used to illustrate the distinct shapes of the PSPs for current-based and conductance-based synapses. In the context of the current-based model, the weight is represented by the value w, while in the conductance-based model, the weight is denoted by g. Both values are chosen to result in identical PSP heights. Moreover, strong weights and a low reversal potential are utilized to achieve the high-conductance state in the conductance-based neuron. Furthermore, the threshold is set to a high value to prevent the neuron from spiking. The value of the reversal potential is exclusively utilized in the simulation with the conductance-based synapse. The terminology introduced in section 2.2 is employed.

Parameter	Value	
C_m	1	nF
$E^\mathrm{e}_\mathrm{rev}$	0	mV
E_leak	−70	mV
V_thres	20	mV
τ_m	10	ms
$\tau^\mathrm{e}_\mathrm{syn}$	3	ms
w	27.92	nA
g	0.87	µS

Table A.2.: Parameters employed to illustrate the stacking of PSPs. The same considerations as those presented in table A.1 apply here. However, reduced weights are utilized as the high-conductance state is not required for the demonstration. Moreover, the reversal potential is further reduced to enhance the PSP height reduction in the conductance-based model.

Parameter	Value	
C_m	1	nF
$E^\mathrm{e}_\mathrm{rev}$	−20	mV
E_leak	−70	mV
V_thres	20	mV
τ_m	10	ms
$\tau^\mathrm{e}_\mathrm{syn}$	3	ms
w	1.12	nA
g	0.022	µS

Network and Neuron Parameters of the Cortical Microcircuit

Table A.3.: Population sizes and input counts of the cortical microcircuit model. Populations are labeled by the layer they are located in and an "e" for the excitatory and an"i" for the inhibitory population in this layer. In the last column, the number of external inputs per neuron are given for each population. Extracted from Potjans et al. 2012.

Population	Neuron number	External inputs
L2/3e	20 683	1600
L2/3i	5834	1500
L4e	21 915	2100
L4i	5479	1900
L5e	4850	2000
L5i	1065	1900
L6e	14 395	2900
L6i	2948	2100

Table A.4.: Connection probability of the cortical microcircuit model. Populations are identified by the layer they are located in and an "e" for the excitatory and an"i" for the inhibitory population in this layer. In the table, the column specifies the pre-synaptic neuron's population and the row the postsynaptic neuron's population. The last column shows the connection probability to the thalamo-cortical inputs. Extracted from Potjans et al. 2012.

	L2/3e	L2/3i	L4e	L4i	L5e	L5i	L6e	L6i	Th
L2/3e	0.101	0.169	0.044	0.082	0.032	0.0	0.008	0.0	0.0
L2/3i	0.135	0.137	0.032	0.052	0.075	0.0	0.004	0.0	0.0
L4e	0.008	0.006	0.05	0.135	0.007	0.0003	0.045	0.0	0.0983
L4i	0.069	0.003	0.079	0.160	0.003	0.0	0.106	0.0	0.0619
L5e	0.100	0.062	0.051	0.006	0.083	0.373	0.02	0.0	0.0
L5i	0.055	0.027	0.026	0.002	0.06	0.316	0.009	0.0	0.0
L6e	0.016	0.007	0.021	0.017	0.057	0.02	0.04	0.225	0.0512
L6i	0.036	0.001	0.003	0.001	0.028	0.008	0.066	0.144	0.0196

Table A.5.: Neuron parameters of the cortical microcircuit model. The terminology introduced in section 2.2 is employed. Moreover, D_e represented the transmission delay of excitatory connections and D_i the delay of inhibitory connections. If parameters are drawn from a Gaussian distribution, a standard deviation σ_x for the model parameter x is given. In addition, in such cases, the value listed for the parameter x represents the mean value of the distribution. Extracted from Potjans et al. 2012.

Parameter	Value
τ_m	10 ms
τ_refrac	2 ms
τ_syn	0.5ms
C_m	250 pF
E_rest	−65 mV
V_reset	−65 mV
V_thres	−50 mV
w_e	87.8pA
$\sigma_{\mathrm{w_e}}$	$0.1w_\mathrm{e}$
w_i	$-4w_\mathrm{e}$
$\sigma_{\mathrm{w_i}}$	$0.1w_\mathrm{i}$
D_e	1.5ms
$\sigma_{\mathrm{D_e}}$	$0.5D_\mathrm{e}$
D_i	0.8ms
$\sigma_{\mathrm{D_i}}$	$0.5D_\mathrm{i}$

Additional Neuron Parameters of the Adapted Cortical Microcircuit

Table A.6.: Mean weight values in the downscaled cortical microcircuit model with conductance-based synapses and prolonged synaptic time constants, as described in section 6.1. Weights exhibit variability based on the postsynaptic neuron to which they are applied, as well as their classification as excitatory (exc) or inhibitory (inh). In the table, the column specifies the postsynaptic neuron's population while the row shows the synapse type of the connection. Populations are denoted by their layer and an "e" for the excitatory and an"i" for the inhibitory population in this layer. All values are presented in pS. In adherence with the model description in Potjans et al. 2012, the weight of the connections from the excitatory population of layer L4 to the excitatory population of L2/3 is doubled, equivalent to a value of 2.7148 pS.

	L2/3e	L2/3i	L4e	L4i	L5e	L5i	L6e	L6i
exc	1.3574	1.3791	1.4689	1.4189	1.4436	1.4564	1.5720	1.4958
inh	11.5090	11.3204	10.6520	11.0030	10.8238	10.7352	10.0541	10.4810

Table A.7.: Resting potential for each population in the downscaled cortical microcircuit model with replaced external inputs. Populations are denoted by their layer and an "e" for the excitatory and an"i" for the inhibitory population in this layer. All values are given in mV.

L2/3e	L2/3i	L4e	L4i	L5e	L5i	L6e	L6i
−42.52	−42.93	−35.50	−38.31	−36.90	−38.31	−24.26	−35.50

A.2. Synapse Stability

As detailed in section 4.1.3, the digital memory tests reveal unstable behavior in the synapse circuits on a small subset of HICANNs. Consequently, extensive testing has been undertaken to scrutinize their performance under diverse conditions. This section provides additional insights into the behavior of the synapses during these tests.

Figure A.1 illustrates, for each synapse within individual HICANNs, the frequency of failures in the repeatedly executed memory test under two different values of the supply voltage (VDDBUS) of the on-chip SRAM controller. Synapses that exhibit partial test failures are categorized unstable. Results are provided for four distinct HICANNs containing synapses marked as unstable. The figure demonstrates that varying the supply voltage has no discernible impact on the observed unstable behavior.

In fig. A.2, an exploration of unstable synapse behavior is undertaken with different writing patterns. At the beginning of each test, the synapse registers undergo either a single write or repeated writes with a fixed value. Subsequently, the stored value is read out 100 times. Additionally, results are presented where the memory is rewritten directly before each read command. Each bin in the histogram represent a count of synapses that failed an individual read test, signifying that a different value was read compared to the value initially written. Thus, the height of each bin corresponds to the number of read cycles demonstrating the same quantity of synapses that failed the test. A single bin with height 100 would demonstrate stable synapse behavior.

Despite the repeated writing of values at the beginning, no improved results were observed. Moreover, comparable variations are obtained in all methods. Consequently, issuing several write commands did not demonstrate improved stability. As a result of this, it is assumed that either values cannot be reliably read out, or each write process poses a risk of setting wrong values.

Figure A.3 depicts long-term stability measurements for all resources exhibiting malfunctioning behavior on wafer 30. As evident, unstable behavior is exclusively observed in the registers of the synapse and repeater circuits. The observed jumps in the synapse measurements result from the behavior of HICANN 275, which alternates between showing no failing synapses or a significant number, O(8000), of failing synapses.

Figure A.4 illustrates the results of long-term measurements involving the memory test in combination with the synapses stability test. If unstable behavior is detected in a single synapse, the entire synapse array containing this synapse is excluded. Notably, the reliability of individual synapse arrays is observed to vary over time. Therefore, all 7 synapse arrays that exhibited unstable behavior at least once are excluded from the final database.

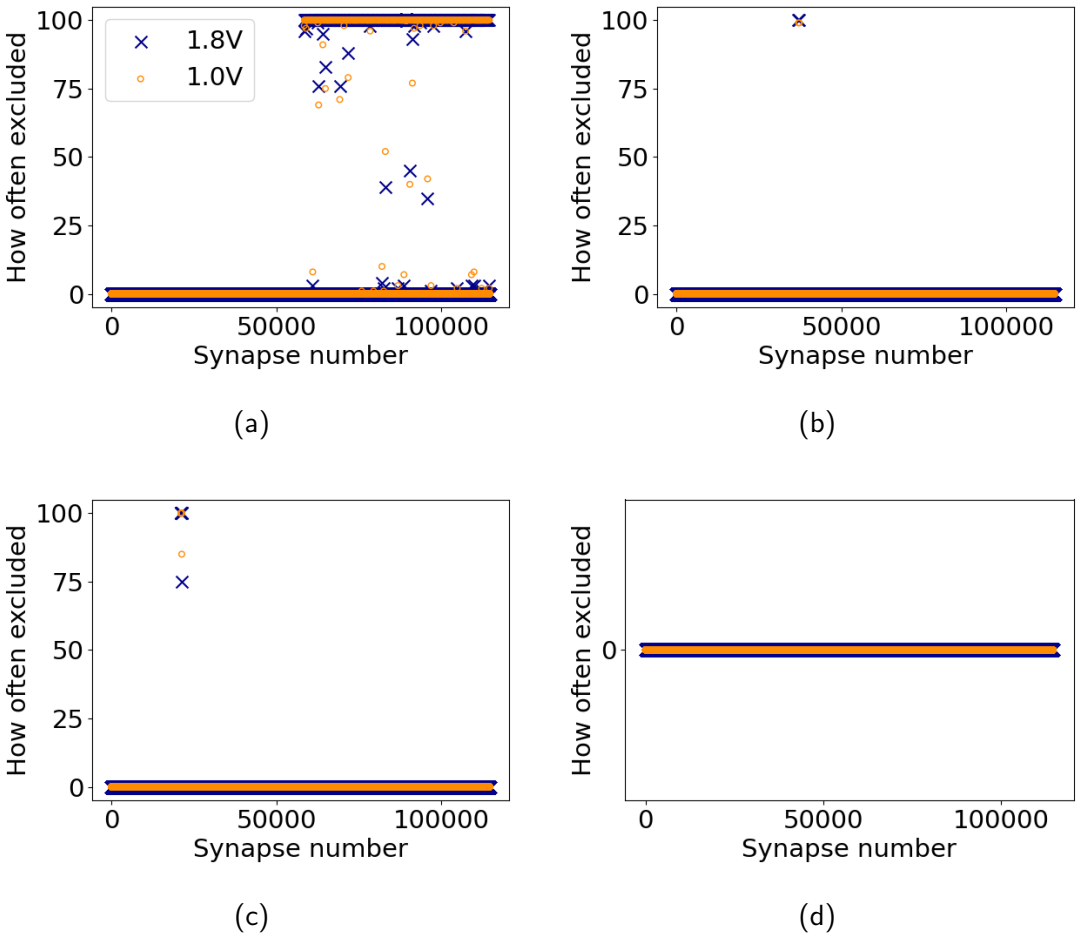


Figure A.1.: Stability test results for different values of the supply voltage VDDBUS for (a) HICANN 23, (b) HICANN 121, (c) HICANN 151 and (d) HICANN 275. For each synapse of a HICANN it is shown how often it is excluded in 100 write/read repetitions of the memory test. A value of 0 or 100 demonstrates that the synapse is stable and fails no or all tests respectively. Values in between indicate unstable behavior in the test. Synapses with numbers above 56 320 are located on the second synapse array. The test results change dependent on the array the synapses are located in.

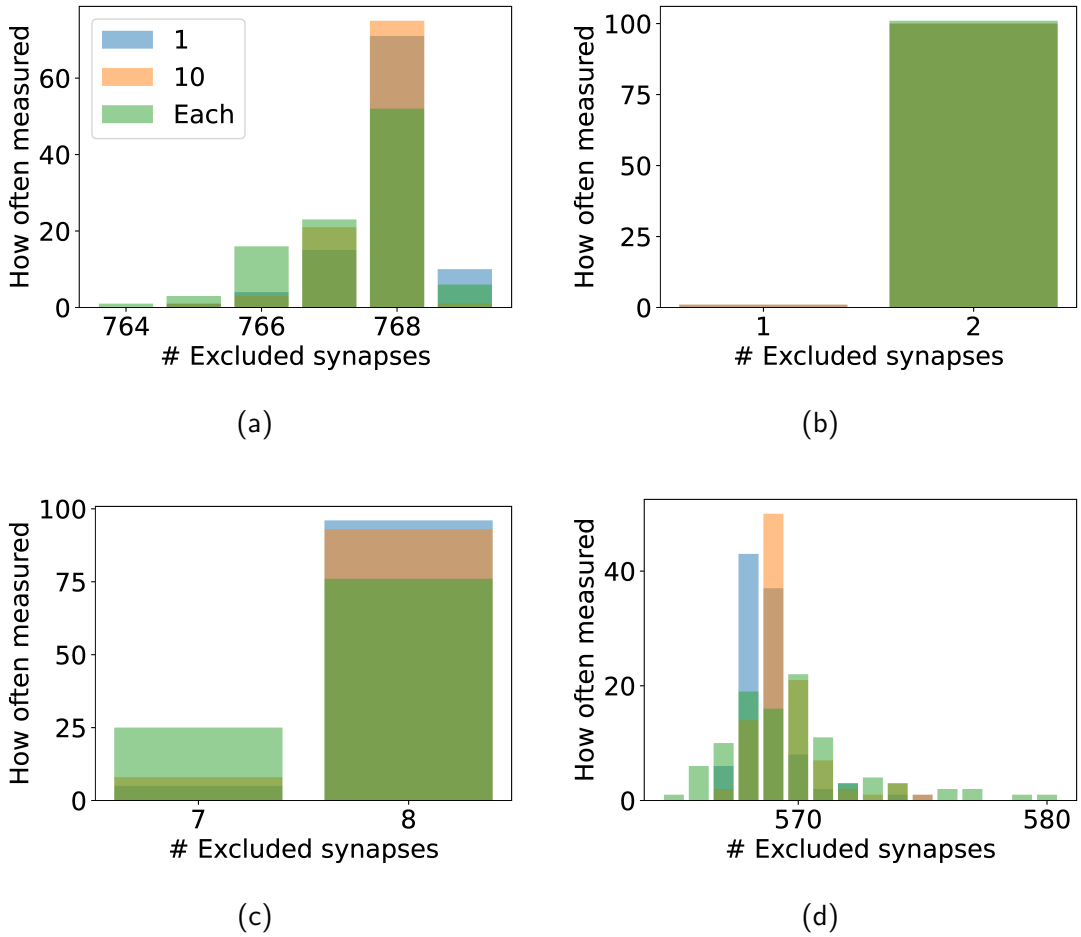


Figure A.2.: Stability test results for different write/read patterns for (a) HICANN 23, (b) HICANN 121, (c) HICANN 151 and (d) HICANN 272. HICANN 275 is not shown since no unstable synapse was detected in the test. Each synapse register is written with a fixed but random value and is read 100 times. On the x-axis the number of synapses a wrong value is read from is shown. The height of each bar represents how often this number of problematic synapses is found in the 100 reads. Here, the color indicates how often the synapse register is written with the same value before the reads are executed. "Each" denotes that all registers are rewritten once before each read command.

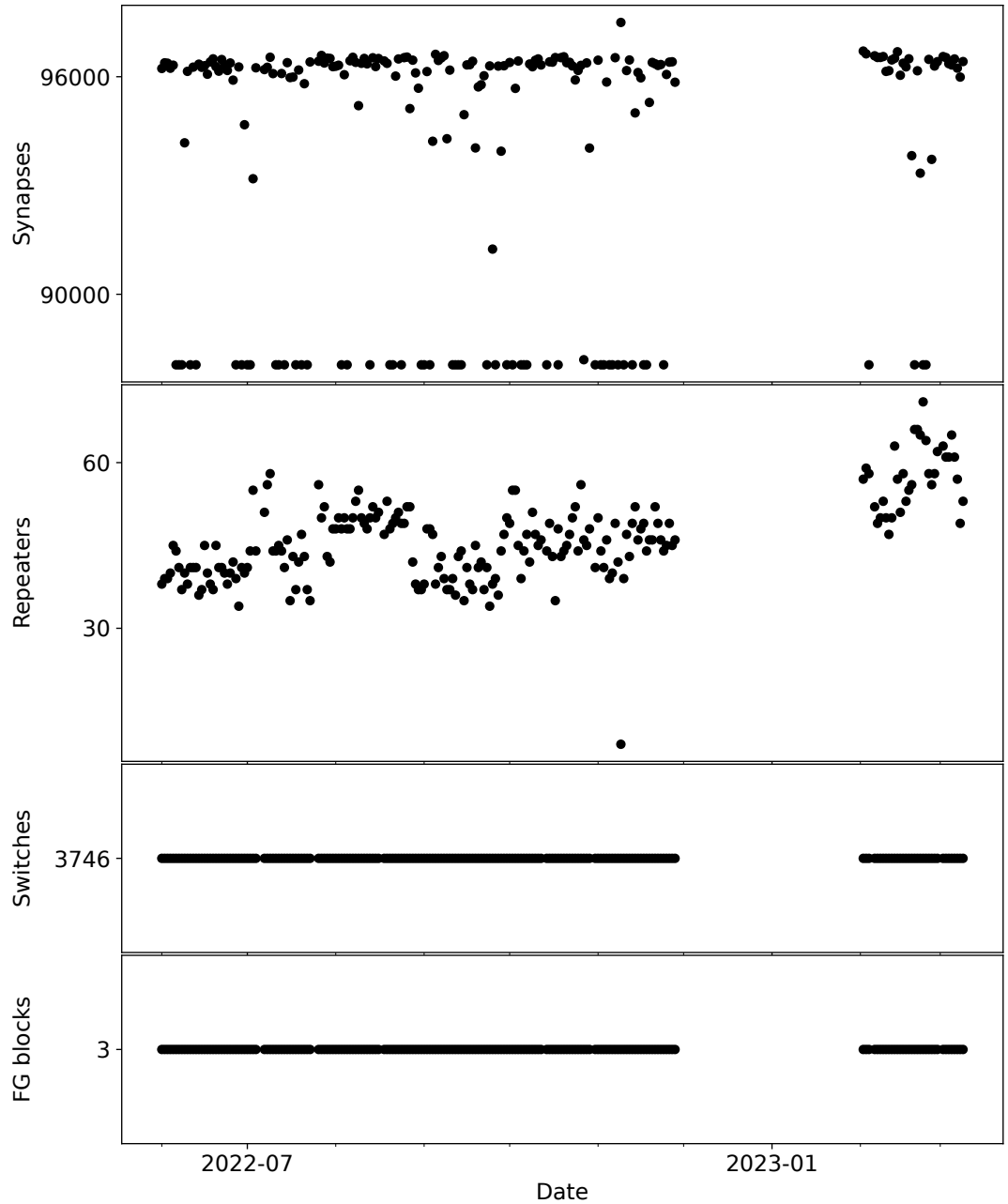


Figure A.3.: Long-term stability of wafer 30. The number of excluded components in the nightly executed memory test are shown. Only resources with found malfunctioning registers are visualized. For synapses and repeaters unstable behavior is observed. Missing data points indicate that the test was not executed either due to software issues in the continuous execution or since the hardware was turned off. The detection of the correct communication possibilities of HICANN 304 is only ensured in the memory test. Therefore, obtaining correct results from it in the memory test requires manual interaction. For this reason, in the nightly executed tests the results for this HICANN, in particular 225 excluded switches, are missing.

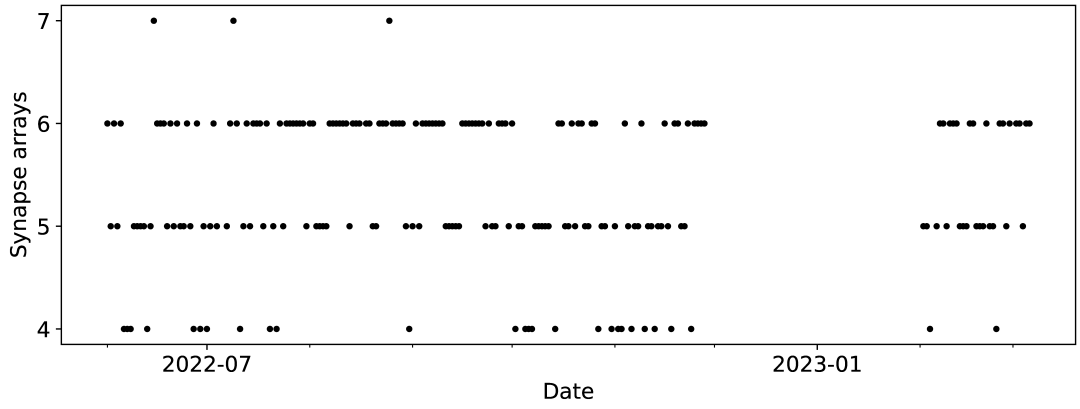


Figure A.4.: Long term synapse array stability of wafer 30. The number of excluded synapse arrays of the nightly executed memory test in combination with the synapse stability test are shown. Missing data points indicate that the test was not executed either due to software issues in the continuous execution or since the hardware was turned off. Two synapse arrays are always excluded due to malfunctioning registers in their controller observed in the memory test. Additionally, five synapse arrays on different HICANNs are excluded due to unstable behavior in the stability test. There, two arrays are found in all tests. The remaining three arrays only fail occasionally and therefore sometimes remain undetected in the test.

A.3. Weight Configuration With Fixed Weight

As discussed in section 4.2.4 the efficacy of synapse circuits on the BrainScaleS-1 system is configured by three parameters. One of it, the digital weight w, is individually set per synapse with a precision of 4 bit. Due to this limited configurability, weights cannot be precisely set.

Defined within the biological regime, weights are translated into a set of hardware parameters based on the results of the weight calibration. The precision of the found parametrization can be assessed by translating the determined hardware values back into the biological regime and comparing them to the targeted values. It is important to note that in this consideration, circuit variations are not taken into account.

For a fixed weight setting across all synapses, the precision of the weight configuration is illustrated in fig. A.5. Due to the stochastic rounding, utilized to prevent modifications of the mean weight, a precise and a less precise weight configuration is obtained, representing two distinct settings of w. As evident by the large difference in synapse counts, the results represent a scenario where one weight configuration closely aligns with the target weight. Consequently, a considerable deviation is observed for the other configuration. Nevertheless, in contrast to the anticipated variations resulting from circuit differences, as depicted in fig. 4.13b, the imprecision arising from limited configurability and stochastic rounding is considered negligible.

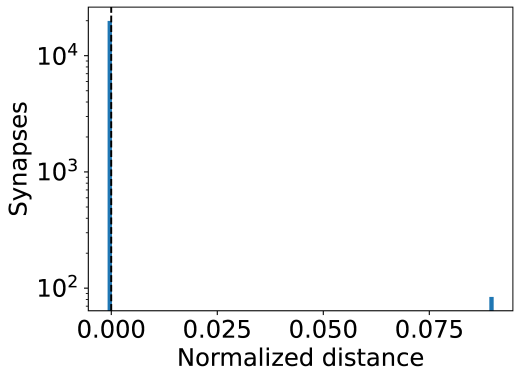


Figure A.5.: Weight deviations expected from the weight configuration algorithm for a fixed target weight. A network of 50 neurons is placed on a single HICANN and stimulated by 20 000 randomly connected synapses from an external population consisting of 50 neurons. The target weight $\frac{w_{\text{model}}}{C_{\text{model}}}$ of all synapses is set to $60\,\text{s}^{-1}$. For each synapse, the hardware parameters V_{gmax}, g_{div} and w are found by the weight translation algorithm considering shared configurations. These values are subsequently converted back into corresponding biological weights, and the disparity from the target weight is computed, normalized in relation to the target weight. The black dashed line marks the mean value of the distribution.

A.4. Firing Patterns of the Balanced Random Network for Different Adaptions

In this section additional spike time plots and spike time histograms of the balanced random network are depicted for different states of adaptation. The various regimes of the model are comprehensively characterized by the neurons' mean firing rate, irregularity, and synchrony. However, the illustration of both individual and collective neuron dynamics contributes valuable insights into the network's behavior.

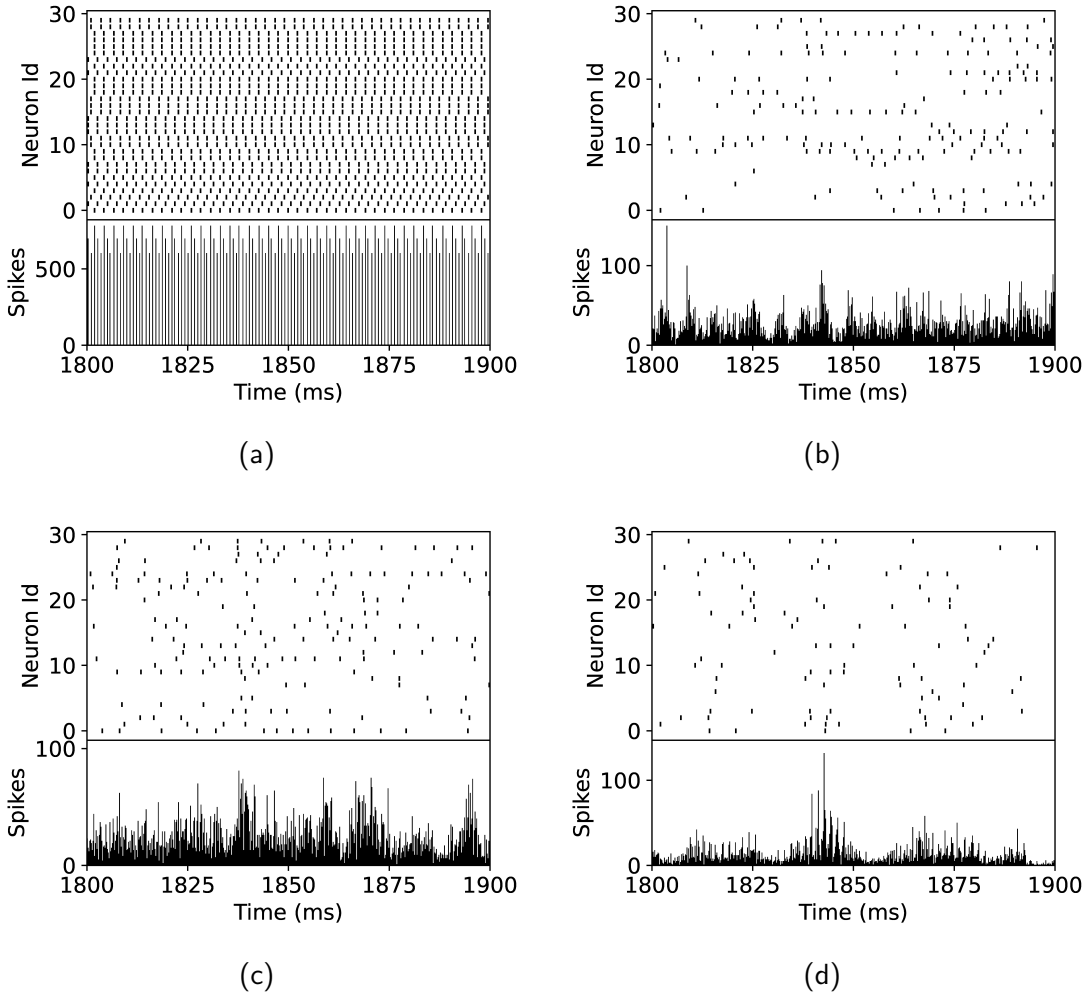


Figure A.6.: Firing patterns of the downscaled balanced random network model. This model aligns with the adaptation state outlined in section 5.1.2. In the upper part of each figure, the spike times of 30 neurons are shown, indicated by vertical lines. In the lower part, the spike time histogram of all 2083 neurons is visualized using a bin size of 0.2 ms. For better visualization, only 100 ms of biological time close to the end of the simulation are illustrated. The figures (b), (c), and (d) represent the same parametrization as the corresponding figures in fig. 5.2. Only (a) is taken from a different regime of $g = 1$ and $\nu_{\text{ext}} = 2 \cdot \nu_{\text{thr}}$ to demonstrate the formation of additional neuron clusters. In contrast to the full-scale model, no fast global oscillations are observed in (b).

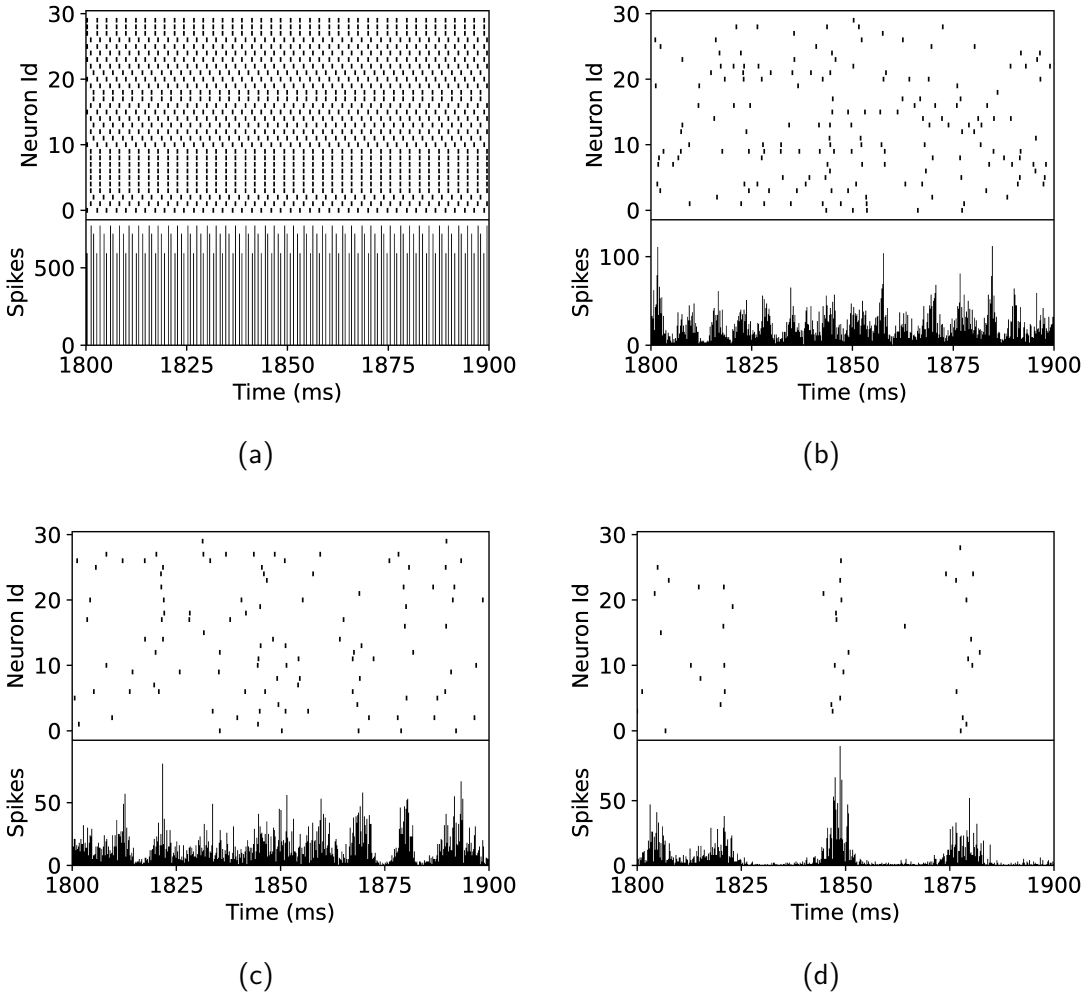


Figure A.7.: Firing patterns of the downscaled balanced random network model with conductance-based synapses. This model aligns with the adaptation state outlined in section 5.1.3. In the upper part of each figure, the spike times of 30 neurons are shown, indicated by vertical lines. In the lower part, the spike time histogram of all 2083 neurons is visualized using a bin size of 0.2 ms. For better visualization, only 100 ms of biological time close to the end of the simulation are illustrated. The figures show the same regimes depicted in fig. A.6.

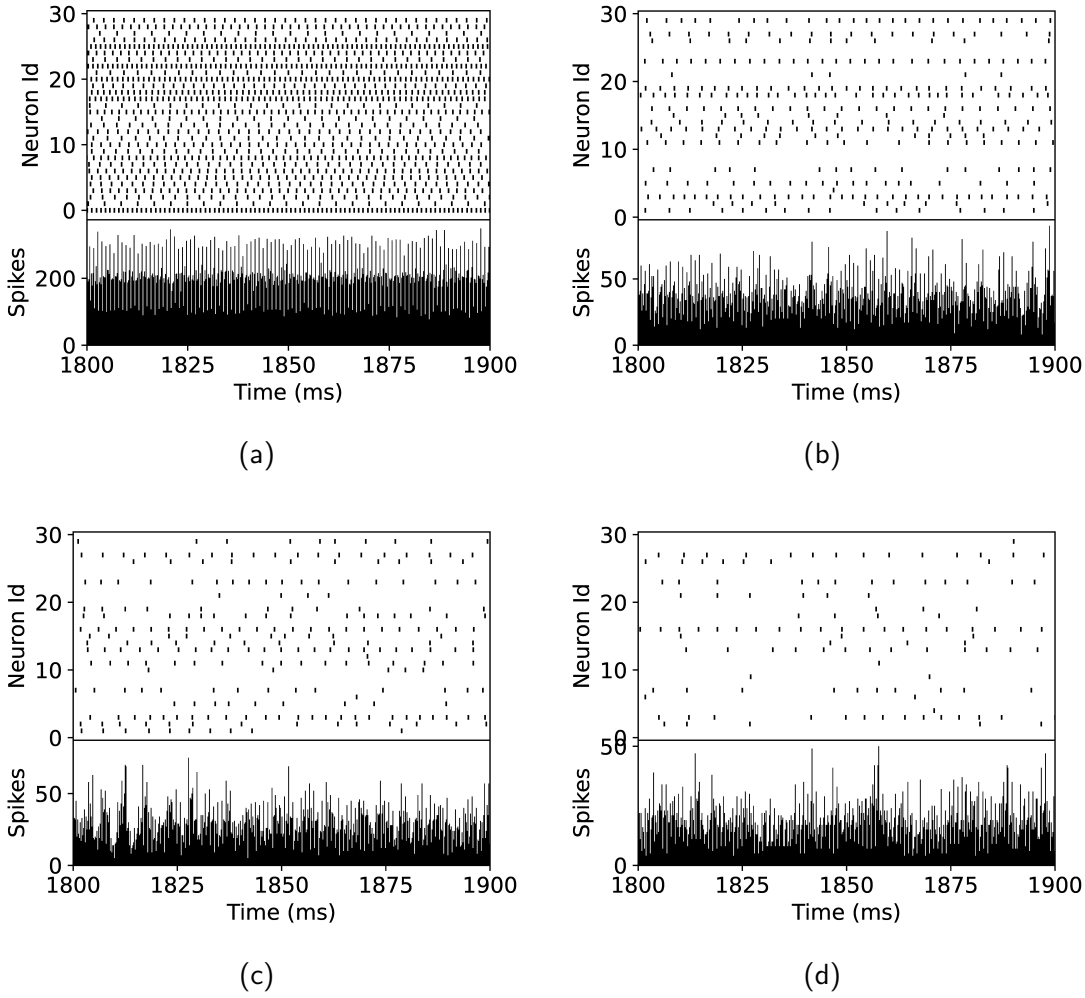


Figure A.8.: Firing patterns of the downscaled balanced random network model with conductance-based synapses and distributed parameters This model aligns with the adaptation state outlined in section 5.1.4. In the upper part of each figure, the spike times of 30 neurons are shown, indicated by vertical lines. In the lower part, the spike time histogram of all 2083 neurons is visualized using a bin size of 0.2 ms. For better visualization, only 100 ms of biological time close to the end of the simulation are illustrated. The figures show the same regimes depicted in fig. A.6.

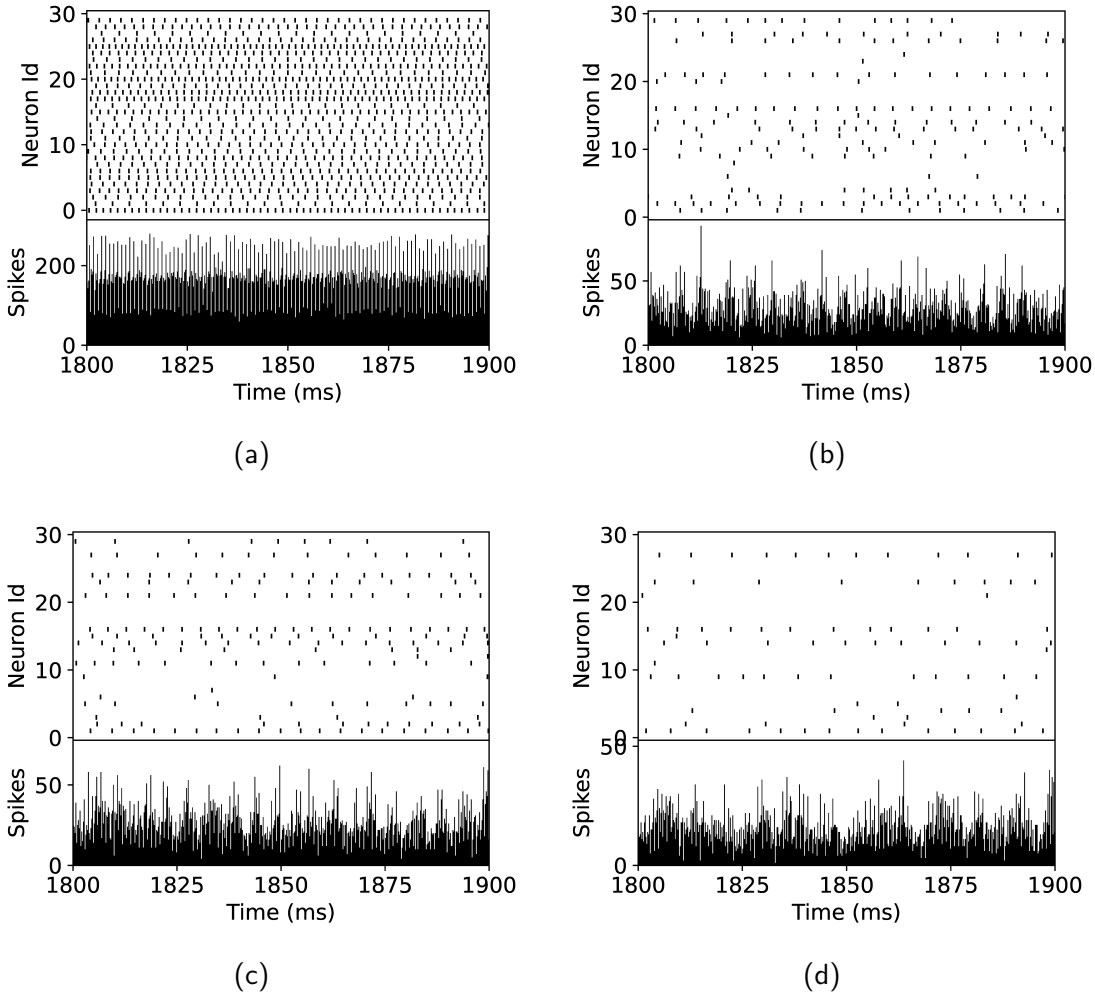


Figure A.9.: Firing patterns of the downscaled balanced random network model with conductance-based synapses, reduced weights, and distributed parameters with extracted map and route results. This model aligns with the adaptation state outlined in section 5.1.5. In the upper part of each figure, the spike times of 30 neurons are shown, indicated by vertical lines. In the lower part, the spike time histogram of all 2083 neurons is visualized using a bin size of 0.2 ms. For better visualization, only 100 ms of biological time close to the end of the simulation are illustrated. The figures show the same regimes depicted in fig. A.6.

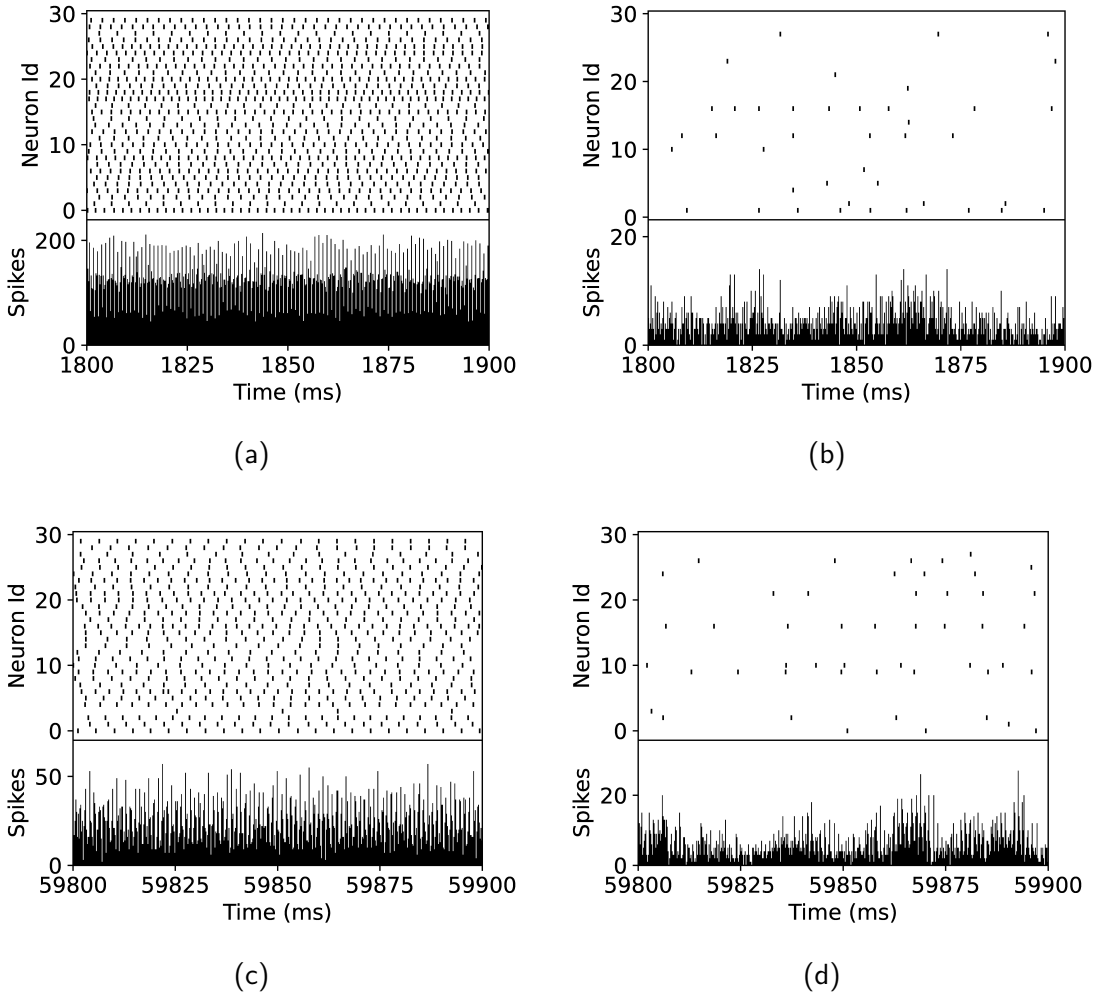


Figure A.10.: Comparison of firing patterns obtained with the NEST simulator and the BrainScaleS-1 hardware. In the upper part of each figure, the spike times of 30 neurons are shown, indicated by vertical lines. In the lower part, the spike time histogram of all 2083 neurons is visualized using a bin size of 0.2 ms. Only in (c) the evaluation is limited to 30 separately placed neurons. For better visualization, only 100 ms of biological time close to the end of the simulation are illustrated. While figures (a) and (b) show the results of the NEST simulation, figures (c) and (d) depict the hardware results. Moreover, figures to the left are taken from the high firing regime with $g = 1.16$ and $\nu_{\text{ext}} = 1.2 \cdot \nu_{\text{thr}}$ and figures to the right from the asynchronous irregular firing regime with low rates at $g = 10.92$ and $\nu_{\text{ext}} = 0.8 \cdot \nu_{\text{thr}}$.

A.5. Map and Route Parameters of the Balanced Random Network

As introduced in section 4.3.1 different map and route algorithms have been implemented in order to find results with a minimum of synapse loss. This section highlights the parametrization of the map and route algorithms, which shows the best results for the balanced random network model and therefore is finally emulated. The utilized algorithms are introduced in section 4.3.1.

To mitigate the impact of finite resistances between membranes, as discussed in section 4.4.2, each neuron is constructed using a set of 8 membrane circuits.

When it comes to the placement algorithm, clustering neurons by their individual connectivity yields best results. This observation is intuitive, as shorter connections with less bus utilization are expected for clustered neurons. Moreover, the balanced random network does not exhibit a population dependent connectivity. Thus, it makes sense to cluster neurons on an individual basis. In addition, the merger tree is configured to merge as many neurons onto a single bus as possible, while still taking into account the maximum number of target synapses.

For handling connections, the Backbone router is employed. The use of the Dijkstra router in a secondary step to route unplaced connections has been determined to provide no significant benefits and, consequently, is not applied.

Spike addresses, used to identify the target synapses, are assigned starting with lower addresses. Therefore, because the most significant bit is transmitted last, there is a reduced likelihood of repeaters misinterpreting the last bit with the rising edge of the stop bit while updating their timing.

Furthermore, the large neuron capacitor is used, given that the hardware implementation is constrained by small weight values. Additionally, the reversal potentials are set to hardware values of $E_{\mathrm{rev}}^{\mathrm{e}}1.3\,\mathrm{V}$ and $E_{\mathrm{rev}}^{\mathrm{i}}0.45\,\mathrm{V}$ utilizing the entire available range, cf. section 4.2.2,

A.6. Additional Network Characteristics of the Cortical Microcircuit

In this section, additional network characteristics of the cortical microcircuit are depicted.

Starting with fig. A.11, the distribution of transmission delays anticipated in the hardware implementation of the cortical microcircuit is illustrated. Furthermore, to enable comparisons with simulation results obtained on the SpiNNaker system, fig. A.12 illustrates the firing rate results derived from the NEST simulation of the cortical microcircuit, processed similarly as presented in Albada et al. 2018.

In addition, to enhance the comprehensiveness of the mean irregularity values utilized for model comparisons in the main text, the distribution of irregularity values within each population of the model is shown for all states of adaptation as well as the final hardware implementation. Each figure illustrates the respective irregularity values corresponding to a firing rate plot presented in the main text. Thus, fig. A.13 corresponds to fig. 6.1, fig. A.14 to fig. 6.3, and fig. A.15 to fig. 6.11.

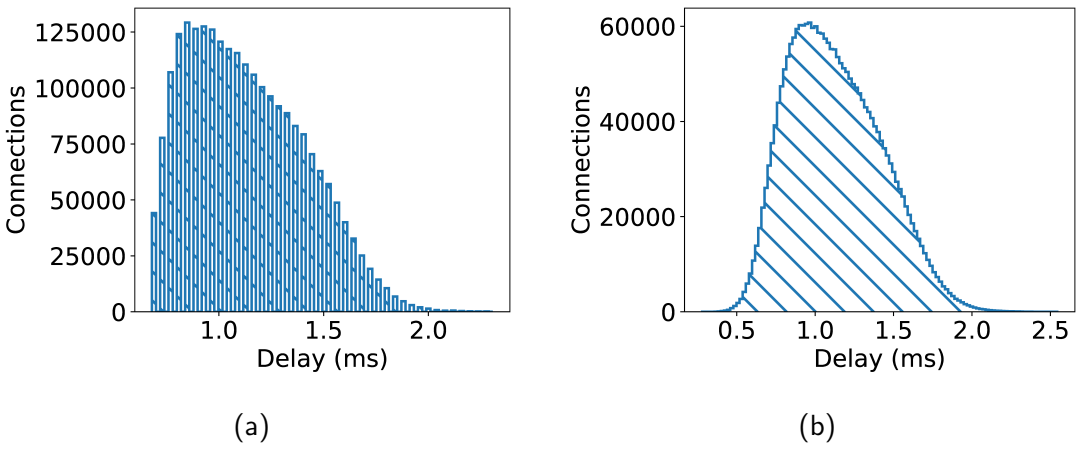


Figure A.11.: Delays obtained from the map and route results of the downscaled cortical microcircuit. According to the calibration results discussed in section 4.2.5, extracted connection lengths are translated into corresponding delay values. Since the delay depends solely on the number of repeaters, discrete results are observed in (a), where each bin represents a specific repeater count. To model circuit variations measured during the calibration, in (b), obtained results are smoothed by a Gaussian distribution with a standard deviation of 0.1 ms.

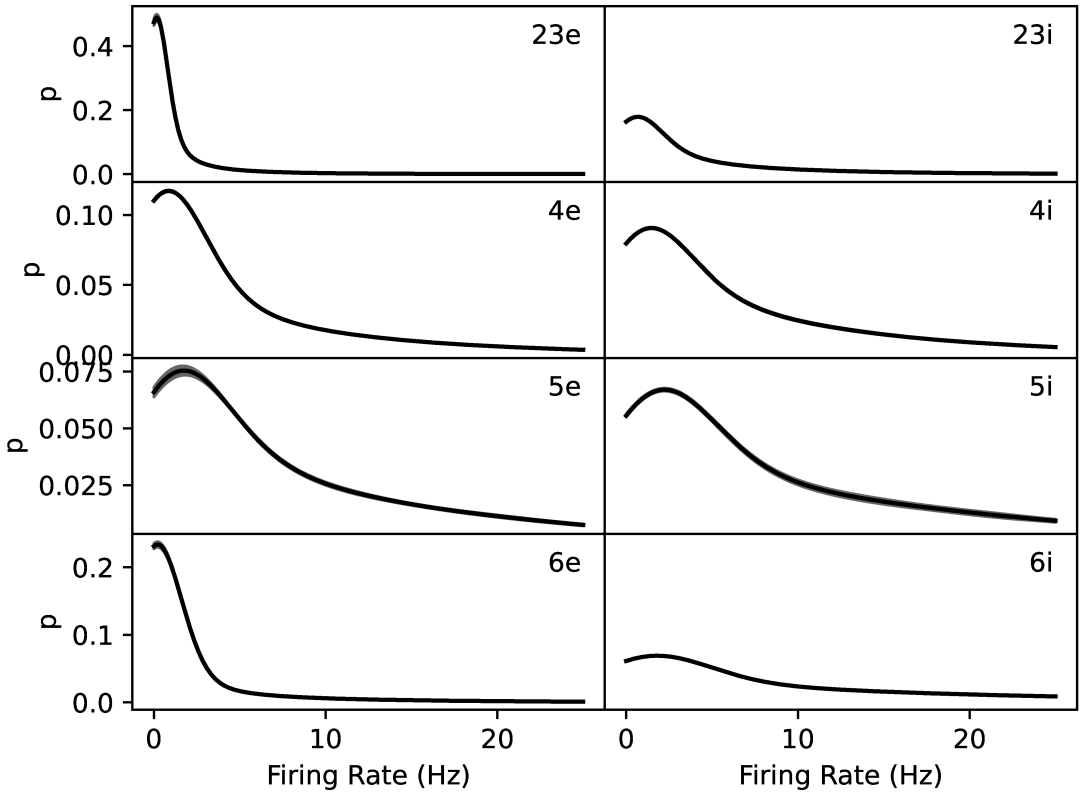


Figure A.12.: Firing rate distributions of the NEST simulation of the cortical microcircuit model as outlined in Potjans et al. 2012. The probability for each firing rate is calculated using a histogram, with bin sizes determined according to equation eq. (6.2). Additionally, the values are smoothed using a Gaussian kernel density function with a bandwidth of $0.3\,\mathrm{s}^{-1}$. Each row displays the results of a different layer of the network, with the excitatory population on the left and the inhibitory population on the right. Displayed are the mean values obtained from 30 simulations, each featuring different randomly generated connections. The shaded area represents the standard deviation across these simulations.

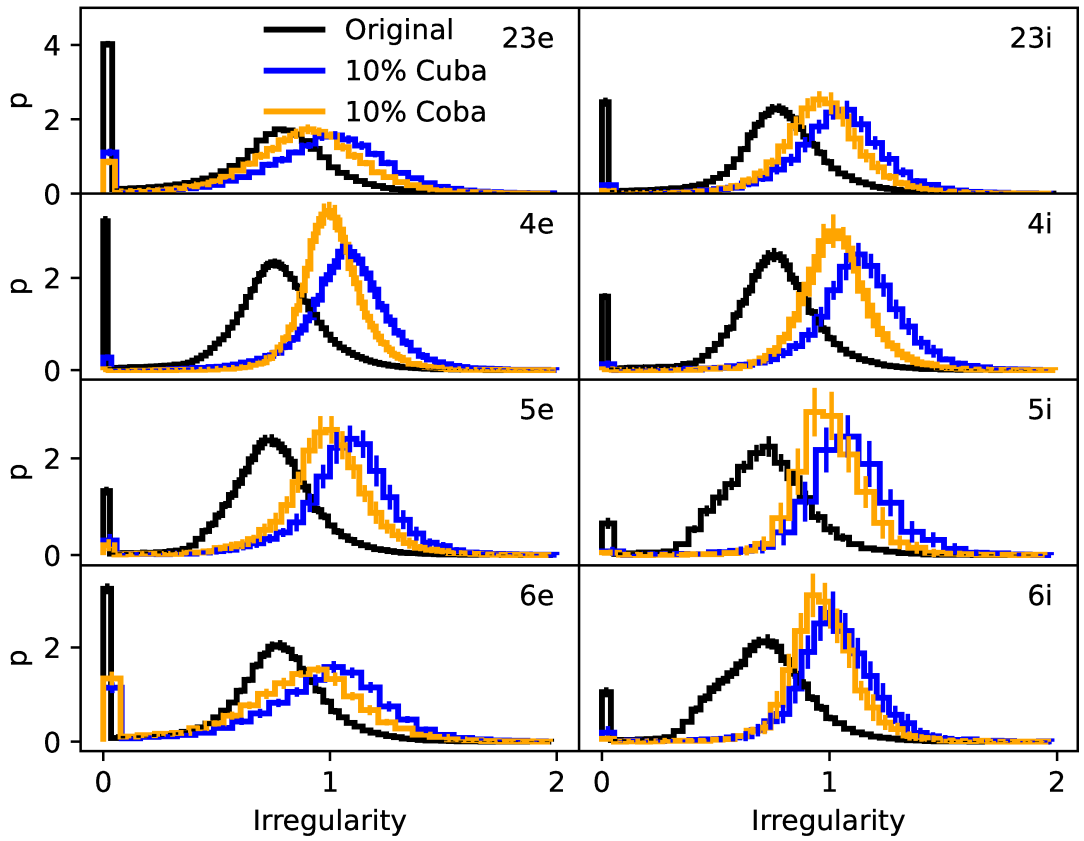


Figure A.13.: Irregularity distributions of the NEST simulation of the cortical microcircuit model for different stages of adaptation. "Original" shows the behavior of the full-scale model, "10 % Cuba" the downscaled current-based version, and "10 % Coba" the downscaled conductance-based implementation. The probability for each value is calculated using a histogram, with bin sizes determined according to equation eq. (6.2). Additionally, the area beneath the histograms is normalized to one. Each row displays the results of a different layer of the network, with the excitatory population on the left and the inhibitory population on the right. Displayed are the mean values obtained from 30 simulations, each featuring different randomly generated connections. Error bars illustrate the standard deviation across these simulations. The represented models correspond to those depicted in fig. 6.1.

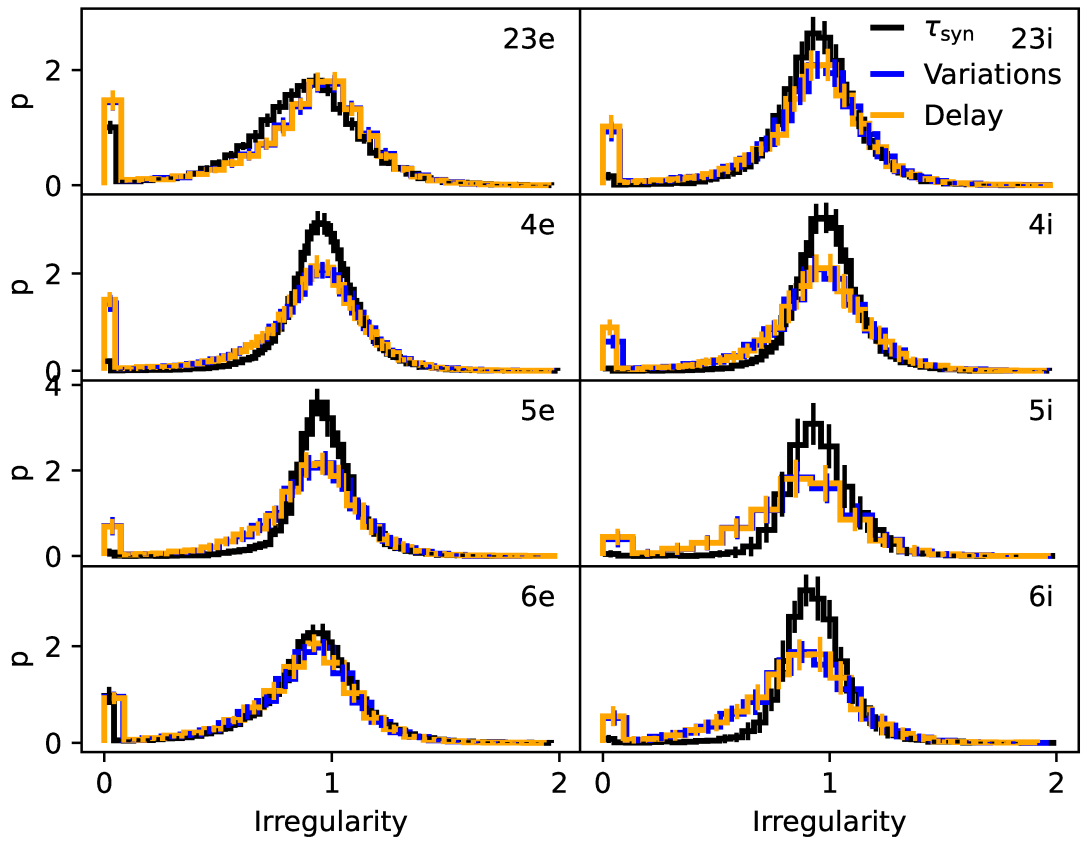


Figure A.14.: Irregularity distributions of the NEST simulation of the cortical microcircuit model for different stages of adaptation. Each modification is added on top of the previous adaptation. Therefore, "τ_{syn}" shows the behavior of the downscaled model with conductance-based synapses under the influence of prolonged synaptic time constants. Based on this, "Variations" illustrates the effects of additionally distributed parameters. Finally, "Delay" displays the final network model with identical delays for excitatory and inhibitory connections, which are described by a Gaussian distribution with a mean value of 1 ms and a standard deviation of 0.25 ms. The values are obtained according to fig. A.13. In simulations involving distributed parameters, it is worth noting that in each repetition, both the connections and the neuron parameters are regenerated. The represented models correspond to those depicted in fig. 6.3.

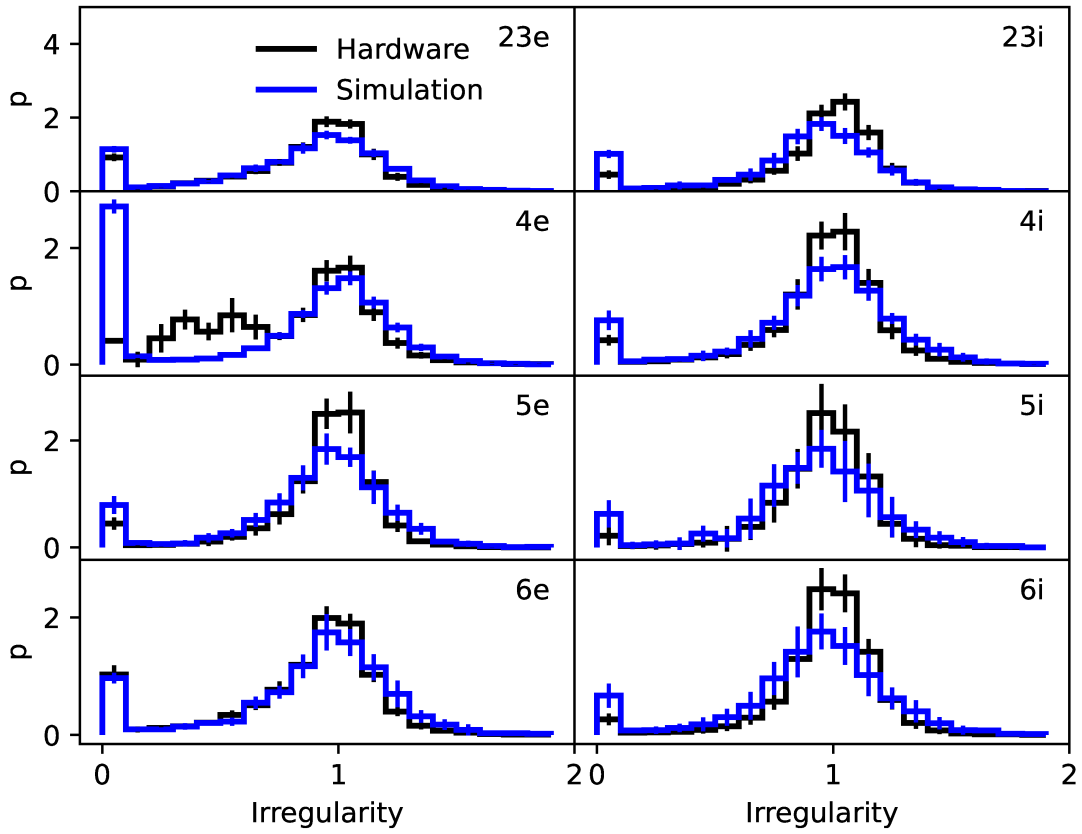


Figure A.15.: Irregularity distributions of the hardware emulation and the NEST simulation of the adapted cortical microcircuit with identical network topologies. The values are obtained according to fig. A.13. Displayed are the mean values obtained from 30 repetitions. In the case of the emulation, the floating gates are reconfigured in between consecutive executions. For the software implementation, each simulation features different randomly generated parameter variations. The error bars represent the standard deviation across these repetitions. The represented models correspond to those depicted in fig. 6.11.

Glossary

BrainScaleS-1 The first generation BrainScaleS system; a wafer-scale mixed-signal accelerated neuromorphic system. 1, 3, 25, 26, 30, 34–42, 55, 56, 59, 66, 71, 73, 75, 81, 83, 86, 87, 90, 91, 95, 101, 104, 107, 108, 110, 111, 116–119, 122, 123, 127, 130, 132, 136, 137, 141–143, 148, 150, 152, 154–156, 166, 172

BrainScaleS-2 The second generation BrainScaleS system; an analog neuromorphic system. 71, 156

ADC Analog-to-digital converter. 35

API Application programming interface. 38, 91

ASIC Application-specific integrated circuit. 1, 25

C++ C++ programming language. 37, 42

cake Calibration framework for the BrainScaleS-1 system. 39

CMOS Complementary metal-oxide-semiconductor. 25

DLL Delay-Locked Loop. 31, 79

EXTOLL Extended atomic low latency. 156

FG block One of 4 floating gate blocks per HICANN. 26, 32, 48, 49

FIFO First in, first out. 29, 113

FPGA Field-Programmable-Gate-Array. 1, 29, 35–37, 40, 42–44, 111, 112, 150, 151

HALbe Hardware abstraction layer back end. 38, 40

HICANN High Input Count Analog Neural Network. 25–30, 32–36, 38, 39, 42–54, 57–62, 64, 67, 68, 70–76, 79–87, 111, 112, 114, 116–119, 139–141, 161–166

HICANN-Group Group of 8 HICANNs connected to the same FPGA. 35, 43, 44, 46, 47

I^2C Inter-integrated circuit protocol. 37

I/O input/output. 36, 111

inter-spike interval Time between consecutive spikes in a spike train. 13, 14, 19, 93

JTAG Joint Test Action Group. 35, 44–46, 52–54

LIF Leaky Integrate-and-Fire. 4, 8–11, 15, 20, 22, 26, 56, 57, 60, 69, 70, 88, 90, 92, 152

LSB Least Significant Bit. 57, 58, 60, 65, 66, 82

marocco Mapping and routing software for the BrainScaleS-1 system. 38

NEST NEural Simulation Tool; simulator for spiking neural network models. 2, 3, 11, 90, 91, 104, 120–123, 125, 126, 128, 130, 131, 136, 137, 142–145, 147–149, 154, 155, 157, 172, 174–178

PCB Printed circuit board. 35, 36, 45, 50

PLL Phase-Locked Loop. 29, 32, 34

PSP Postsynaptic Potential. 7, 9–12, 16, 34, 57, 58, 60–63, 73, 74, 84–86, 92, 99, 104–106, 129, 144, 157, 158

PyHMF PyNN for the Hybrid Multiscale Facility; a PyNN implementation for the BrainScaleS-1 system. 38

PyNN A Python package for simulator-independent specification of neuronal network models. 38, 40, 41, 91, 157

Python Python programming language. 37–39

SpiNNaker Spiking neural network architecture. 124, 155, 174

SRAM Static random-access memory. 26, 31–33, 47, 53, 161

StHAL Stateful hardware abstraction layer. 38

XML Extensible markup language. 39, 40, 42, 150

Contributions

Peer-Reviewed Publications

Hartmut Schmidt, José Montes, Andreas Grübl, Maurice Güttler, Dan Husmann, Joscha Ilmberger, Jakob Kaiser, Christian Mauch, Eric Müller, Lars Sterzenbach, Johannes Schemmel, and Sebastian Schmitt (2023). „From Clean Room to Machine Room: Commissioning of the First-Generation BrainScaleS Wafer-Scale Neuromorphic System“. In: *Neuromorphic Computing and Engineering*. DOI: 10.1088/2634-4386/acf7e4
Content of this publication is presented in chapter 4.

Eric Müller, Sebastian Schmitt, Christian Mauch, Sebastian Billaudelle, Andreas Grübl, Maurice Güttler, Dan Husmann, Joscha Ilmberger, Sebastian Jeltsch, Jakob Kaiser, Johann Klähn, Mitja Kleider, Christoph Koke, José Montes, Paul Müller, Johannes Partzsch, Felix Passenberg, Hartmut Schmidt, Bernhard Vogginger, Jonas Weidner, Christian Mayr, and Johannes Schemmel (2022). „The Operating System of the Neuromorphic BrainScaleS-1 System“. In: *Neurocomputing* 501, pp. 790–810. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2022.05.081. arXiv: 2003.13749 [cs.NE]
Content of this publication is presented in section 4.1 and section 4.3.1.

Supervision

Quirinus Schwarzenböck (2019). „Towards Balanced Random Networks on the BrainScaleS I System“. Bachelor. Universität Heidelberg
Content of this thesis is presented in section 5.1.

Jonas Weidner (2019). „Experiment Visualization and Simulations towards a Cortical Microcircuit on the BrainScaleS Neuromorphic Hardware“. Bachelor thesis. Universität Heidelberg
Content of this thesis is presented in section 6.1.1.

Jakob Kaiser (2020). „Implementation of Large Scale Neural Networks on the Neuromorphic BrainScaleS-1 System“. Master thesis. Ruprecht-Karls-Universität Heidelberg
Content of this thesis is presented in section 4.3.2.

Moritz Hornung (2020). „Adapting the Cortical Microcircuit Model for the BrainScaleS-1 hardware“. Bachelor thesis. Universität Heidelberg
Content of this thesis is presented in section 6.1.4.

Bibliography

Abbott, L and Wade Regehr (Nov. 2004). „Synaptic computation“. In: *Nature* 431, pp. 796–803. DOI: 10.1038/nature03010.

Albada, Sacha van, Moritz Helias, and Markus Diesmann (Nov. 2014). „Scalability of Asynchronous Networks Is Limited by One-to-One Mapping between Effective Connectivity and Correlations“. In: *PLOS Computational Biology* 11. DOI: 10.1371/journal.pcbi.1004490.

Albada, Sacha J. van, Andrew G. Rowley, Johanna Senk, Michael Hopkins, Maximilian Schmidt, Alan B. Stokes, David R. Lester, Markus Diesmann, and Steve B. Furber (2018). „Performance Comparison of the Digital Neuromorphic Hardware SpiNNaker and the Neural Network Simulation Software NEST for a Full-Scale Cortical Microcircuit Model“. In: *Frontiers in Neuroscience* 12. ISSN: 1662-453X. DOI: 10.3389/fnins.2018.00291. URL: https://www.frontiersin.org/article/10.3389/fnins.2018.00291.

Alberts, Bruce, Dennis Bray, Julian Lewis, Martin Raff, Keith Roberts, and Jamed D. Watsonx (1994). *Molecular Biology of the Cell, third edition*. Garland Publishing, Inc. ISBN: 0815316208.

Amit, D J and N Brunel (Jan. 1997). „Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex“. In: *Cereb Cortex* 7.3, pp. 237–52.

Azevedo, Frederico A.C., Ludmila R.B. Carvalho, Lea T. Grinberg, José Marcelo Farfel, Renata E.L. Ferretti, Renata E.P. Leite, Wilson Jacob Filho, Roberto Lent, and Suzana Herculano-Houzel (2009). „Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain“. In: *Journal of Comparative Neurology* 513.5, pp. 532–541. DOI: https://doi.org/10.1002/cne.21974. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cne.21974. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cne.21974.

Bassett, Danielle S., Perry Zurn, and Joshua I. Gold (2018). „On the nature and use of models in network neuroscience“. In: *Nature Reviews Neuroscience* 19, pp. 566–578. DOI: https://doi.org/10.1038/s41583-018-0038-8.

Billaudelle, Sebastian (2014). „Characterisation and Calibration of Short Term Plasticity on a Neuromorphic Hardware Chip“. Bachelor thesis. Universität Heidelberg.

Billaudelle, Sebastian, Johannes Weis, Philipp Dauer, and Johannes Schemmel (2022). „An accurate and flexible analog emulation of AdEx neuron dynamics in silicon“.

In: *2022 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pp. 1–4. DOI: 10.1109/ICECS202256217.2022.9971058.

Binzegger, Tom, Rodney J Douglas, and Kevan A C Martin (Sept. 2004). „A quantitative map of the circuit of cat primary visual cortex“. In: *J Neurosci* 24.39, pp. 8441–53.

Bragin, A, G Jando, Z Nadasdy, J Hetke, K Wise, and G Buzsaki (1995). „Gamma (40-100 Hz) oscillation in the hippocampus of the behaving rat“. In: *Journal of Neuroscience* 15.1, pp. 47–60. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.15-01-00047.1995. eprint: https://www.jneurosci.org/content/15/1/47.full.pdf. URL: https://www.jneurosci.org/content/15/1/47.

Brette, R. and W. Gerstner (2005). „Adaptive Exponential Integrate-and-Fire Model as an Effective Description of Neuronal Activity“. In: *J. Neurophysiol.* 94, pp. 3637–3642. DOI: 10.1152/jn.00686.2005.

Brunel, N. (2000). „Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons“. In: *Journal of Computational Neuroscience* 8.3, pp. 183–208.

Brunel, Nicolas and Mark CW Van Rossum (2007). „Lapicque's 1907 paper: from frogs to integrate-and-fire“. In: *Biological cybernetics* 97.5-6, pp. 337–339.

Chris 73, Diberri, and tiZom (2007). *Schematic of an action potential*. Accessed: 2023-08-21, Licence: CC BY-SA at https://creativecommons.org/licenses/by-sa/3.0/, Original by en:User:Chris 73, updated by en:User:Diberri, converted to SVG by tiZom. URL: https://commons.wikimedia.org/wiki/File:Action_potential.svg.

Cowan, W Maxwell, Thomas C Südhof, and Charles F Stevens (2003). *Synapses*. Johns Hopkins University Press.

Dantzker, J.L. and E.M. Callaway (Aug. 2000). „Laminar sources of synaptic input to cortical inhibitory interneurons and pyramidal neurons“. In: *Nature neuroscience* 3, pp. 701–707. DOI: 10.1038/76656.

Dasbach, Stefan, Tom Tetzlaff, Markus Diesmann, and Johanna Senk (2021). „Dynamical Characteristics of Recurrent Neuronal Networks Are Robust Against Low Synaptic Weight Resolution“. In: *Frontiers in Neuroscience* 15. ISSN: 1662-453X. DOI: 10.3389/fnins.2021.757790. URL: https://www.frontiersin.org/articles/10.3389/fnins.2021.757790.

Davies, Mike (Sept. 2019). „Benchmarks for progress in neuromorphic computing“. In: *Nature Machine Intelligence* 1, pp. 386–388. DOI: 10.1038/s42256-019-0097-1.

Davies, Mike, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. (2018). „Loihi: A neuromorphic manycore processor with on-chip learning“. In: *IEEE Micro* 38.1, pp. 82–99. DOI: 10.1109/MM.2018.112130359.

Davison, Andrew P., Daniel Brüderle, Jochen Eppler, Jens Kremkow, Eilif Muller, Dejan Pecevski, Laurent Perrinet, and Pierre Yger (2009). „PyNN: a common interface for

neuronal network simulators". In: *Front. Neuroinform.* 2.11. DOI: 10.3389/neuro.11.011.2008.

Dijkstra, E.W. (1959). „A Note on Two Problems in Connexion with Graphs." In: *Numerische Mathematik* 1, pp. 269–271. URL: http://eudml.org/doc/131436.

Drachman, David A. (2005). „Do we have brain to spare?" In: *Neurology* 64.12, pp. 2004–2005. ISSN: 0028-3878. DOI: 10.1212/01.WNL.0000166914.38327.BB. eprint: https://n.neurology.org/content/64/12/2004.pdf. URL: https://n.neurology.org/content/64/12/2004.

Eccles, J. C., P. Fatt, and K. Koketsu (1954). „Cholinergic and Inhibitory Synapses in a Pathway from Motor-Axon Collaterals to Motoneurones". In: *J. Physiol.* 126, pp. 524–562.

Eppler, Jochen M., Moritz Helias, Eilif Muller, Markus Diesmann, and Marc-Oliver Gewaltig (2008). „PyNEST: a convenient interface to the NEST simulator". In: *Front. Neuroinform.* 2.12. DOI: 10.3389/neuro.11.012.2008.

Fieres, J., J. Schemmel, and K. Meier (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)*.

Freedman, David A. and Persi Diaconis (1981). „On the histogram as a density estimator:L2 theory". In: *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 57, pp. 453–476. URL: https://api.semanticscholar.org/CorpusID:14437088.

Friedmann, Simon (2013). „A New Approach to Learning in Neuromorphic Hardware". PhD thesis. Ruprecht-Karls-Universität Heidelberg. DOI: 10.11588/heidok.00015359. URL: http://archiv.ub.uni-heidelberg.de/volltextserver/15359/.

Furber, Steve (Aug. 2016). „Large-scale neuromorphic computing systems". In: *Journal of Neural Engineering* 13.5, p. 051001. DOI: 10.1088/1741-2560/13/5/051001.

Furber, Steve B., Francesco Galluppi, Steve Temple, and Luis A. Plana (2014). „The SpiNNaker Project". In: *Proceedings of the IEEE*. Vol. 102. 5, pp. 652–665. DOI: 10.1109/JPROC.2014.2304638.

Gerstner, Wulfram, Henning Sprekeler, and Gustavo Deco (2012). „Theory and Simulation in Neuroscience". In: *Science* 338.6103, pp. 60–65. DOI: 10.1126/science.1227356. eprint: https://www.science.org/doi/pdf/10.1126/science.1227356. URL: https://www.science.org/doi/abs/10.1126/science.1227356.

Gewaltig, Marc-Oliver and Markus Diesmann (2007). „NEST (NEural Simulation Tool)". In: *Scholarpedia* 2.4, p. 1430. DOI: 10.4249/scholarpedia.1430.

Golosio, Bruno, Gianmarco Tiddia, Chiara De Luca, Elena Pastorelli, Francesco Simula, and Pier Stanislao Paolucci (2021). „Fast simulations of highly-connected spiking

cortical models using GPUs“. In: *Frontiers in Computational Neuroscience* 15, p. 627620.

Göltz, Julian, Laura Kriener, Andreas Baumbach, Sebastian Billaudelle, Oliver Breitwieser, Benjamin Cramer, Dominik Dold, Ákos Ferenc Kungl, Walter Senn, Johannes Schemmel, Karlheinz Meier, and Mihai A. Petrovici (2021). „Fast and energy-efficient neuromorphic deep learning with first-spike times“. In: *Nature Machine Intelligence* 3.9, pp. 823–835. DOI: 10.1038/s42256-021-00388-x.

Goodman, Dan and Romain Brette (2008). „Brian: a simulator for spiking neural networks in Python“. In: *Front. Neuroinform.* 2.5.

HBP SP9 partners (Mar. 2014). *Neuromorphic Platform Specification*. Human Brain Project.

Heittmann, Arne, Georgia Psychou, Guido Trensch, Charles E. Cox, Winfried W. Wilcke, Markus Diesmann, and Tobias G. Noll (2022). „Simulating the Cortical Microcircuit Significantly Faster Than Real Time on the IBM INC-3000 Neural Supercomputer“. In: *Frontiers in Neuroscience* 15. ISSN: 1662-453X. DOI: 10.3389/fnins.2021.728460. URL: https://www.frontiersin.org/articles/10.3389/fnins.2021.728460.

Hock, Matthias (2009). *Test of Components for a Wafer-Scale Neuromorphic Hardware System*. Diploma thesis, University of Heidelberg, HD-KIP-09-37, http://www.kip.uni-heidelberg.de/Veroeffentlichungen/details.php?id=1935.

Hodgkin, Alan Lloyd and Andrew F. Huxley (Aug. 1952). „A quantitative description of membrane current and its application to conduction and excitation in nerve.“ In: *J Physiol* 117.4, pp. 500–544. ISSN: 0022-3751. URL: http://view.ncbi.nlm.nih.gov/pubmed/12991237.

Hornung, Moritz (2020). „Adapting the Cortical Microcircuit Model for the BrainScaleS-1 hardware“. Bachelor thesis. Universität Heidelberg.

NXP Semiconductors (2012). *I2C-bus specification and user manual*.

Ilmberger, Joscha (2017). „Development of a digitizer for the BrainScaleS neuromorphic hardware system“. Master thesis. Ruprecht-Karls-Universität Heidelberg.

Indiveri, Giacomo, Bernabe Linares-Barranco, Tara Julia Hamilton, André van Schaik, Ralph Etienne-Cummings, Tobi Delbruck, Shih-Chii Liu, Piotr Dudek, Philipp Häfliger, Sylvie Renaud, Johannes Schemmel, Gert Cauwenberghs, John Arthur, Kai Hynna, Fopefolu Folowosele, Sylvain Saighi, Teresa Serrano-Gotarredona, Jayawan Wijekoon, Yingxue Wang, and Kwabena Boahen (2011). „Neuromorphic silicon neuron circuits“. In: *Frontiers in Neuroscience* 5.0. ISSN: 1662-453X. DOI: 10.3389/fnins.2011.00073. URL: http://www.frontiersin.org/Journal/Abstract.aspx?s=755&name=neuromorphic%20engineering&ART_DOI=10.3389/fnins.2011.00073.

ISO (2017). *Programming languages — C++*. 5th ed. Geneva, Swiss: International Organization for Standardization, p. 1605. URL: https://www.iso.org/standard/68564.html.

Jarosz, Quasar (2009). *Sketch of a neuron*. Accessed: 2023-08-21, Licence: CC BY-SA at https://creativecommons.org/licenses/by-sa/3.0/, Original by en:User:Quasar Jarosz. URL: https://commons.wikimedia.org/wiki/File:Neuron_Hand-tuned.svg.

Jeltsch, Sebastian (2014). „A Scalable Workflow for a Configurable Neuromorphic Platform“. PhD thesis. Universität Heidelberg.

Jones, Eric, Travis Oliphant, and Pearu Peterson (2001). *SciPy: Open source scientific tools for Python*. URL: http://www.scipy.org/.

Kaiser, Jakob (2020). „Implementation of Large Scale Neural Networks on the Neuromorphic BrainScaleS-1 System“. Master thesis. Ruprecht-Karls-Universität Heidelberg.

Kaiser, Jakob, Sebastian Billaudelle, Eric Müller, Christian Tetzlaff, Johannes Schemmel, and Sebastian Schmitt (2022). „Emulating dendritic computing paradigms on analog neuromorphic hardware“. In: *Neuroscience* 489, pp. 290–300. ISSN: 0306-4522. DOI: 10.1016/j.neuroscience.2021.08.013. URL: https://www.sciencedirect.com/science/article/pii/S0306452221004218.

Kauth, Kevin, Tim Stadtmann, Vida Sobhani, and Tobias Gemmeke (2023). „neuroAIx-Framework: design of future neuroscience simulation systems exhibiting execution of the cortical microcircuit model 20x faster than biological real-time“. In: *Frontiers in Computational Neuroscience* 17. ISSN: 1662-5188. DOI: 10.3389/fncom.2023.1144143. URL: https://www.frontiersin.org/articles/10.3389/fncom.2023.1144143.

Klähn, Johann (2017). „Training Functional Networks on Large-Scale Neuromorphic Hardware“. Master. Universität Heidelberg.

Kleider, Mitja (2017). „Neuron Circuit Characterization in a Neuromorphic System“. HD-KIP 17-135. PhD thesis. Universität Heidelberg. URL: http://www.kip.uni-heidelberg.de/Veroeffentlichungen/details.php?id=3657.

Knight, James C and Thomas Nowotny (2021). „Larger GPU-accelerated brain simulations with procedural connectivity“. In: *Nature Computational Science* 1.2, pp. 136–142.

Koke, Christoph (Feb. 2017). „Device Variability in Synapses of Neuromorphic Circuits“. PhD thesis. Ruprecht-Karls-University Heidelberg. URL: http://www.ub.uni-heidelberg.de/archiv/22742.

Kononov, Alex (2011). „Testing of an Analog Neuromorphic Network Chip“. HD-KIP-11-83. Diploma thesis. Ruprecht-Karls-Universität Heidelberg.

Kungl, Akos F., Sebastian Schmitt, Johann Klähn, Paul Müller, Andreas Baumbach, Dominik Dold, Alexander Kugele, Eric Müller, Christoph Koke, Mitja Kleider, Christian Mauch, Oliver Breitwieser, Luziwei Leng, Nico Gürtler, Maurice Güttler,

Dan Husmann, Kai Husmann, Andreas Hartel, Vitali Karasenko, Andreas Grübl, Johannes Schemmel, Karlheinz Meier, and Mihai A. Petrovici (2019). „Accelerated Physical Emulation of Bayesian Inference in Spiking Neural Networks“. In: *Frontiers in Neuroscience* 13, p. 1201. ISSN: 1662-453X. DOI: 10.3389/fnins.2019.01201. URL: https://www.frontiersin.org/article/10.3389/fnins.2019.01201.

Kurth, Anno C, Johanna Senk, Dennis Terhorst, Justin Finnerty, and Markus Diesmann (Mar. 2022). „Sub-realtime simulation of a neuronal network of natural density“. In: *Neuromorphic Computing and Engineering* 2.2, p. 021001. DOI: 10.1088/2634-4386/ac55fc. URL: https://dx.doi.org/10.1088/2634-4386/ac55fc.

Labs, Grafana (2018). *Grafana: The open observability platform*. URL: https://grafana.com.

Lande, T.S., H. Ranjbar, M. Ismail, and Y. Berg (Feb. 1996). „An analog floating-gate memory in a standard digital technology“. In: *Microelectronics for Neural Networks, 1996., Proceedings of Fifth International Conference on*, pp. 271–276. DOI: 10.1109/MNNFS.1996.493802.

London, M. and M. Häusser (2005). „Dendritic computation“. In: *Annu. Rev. Neurosci.* 28, pp. 503–532. DOI: 10.1146/annurev.neuro.28.061604.135703.

McCulloch, Warren S. and Walter Pitts (1943). „A logical calculus of the ideas immanent in nervous activity“. In: *Bulletin of Mathematical Biophysics*, pp. 127–147.

McGuire, Barbara A, Jean-Pierre Hornung, Charles D Gilbert, and Torsten N Wiesel (1984). „Patterns of synaptic input to layer 4 of cat striate cortex“. In: *Journal of Neuroscience* 4.12, pp. 3021–3033.

Mead, C. A. (1989). *Analog VLSI and Neural Systems*. Reading, MA: Addison Wesley.

Mead, C. A. (1990). „Neuromorphic Electronic Systems“. In: *Proceedings of the IEEE* 78, pp. 1629–1636.

Merolla, Paul A., John V. Arthur, Rodrigo Alvarez-Icaza, Andrew S. Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. (2014). „A million spiking-neuron integrated circuit with a scalable communication network and interface“. In: *Science* 345.6197, pp. 668–673. DOI: 10.1126/science.1254642.

Millner, Sebastian (Nov. 2012). „Development of a Multi-Compartment Neuron Model Emulation“. PhD thesis. Ruprecht-Karls-University Heidelberg. URL: http://www.ub.uni-heidelberg.de/archiv/13979.

Millner, Sebastian, Andreas Grübl, Karlheinz Meier, Johannes Schemmel, and Marc-Olivier Schwartz (2010). „A VLSI Implementation of the Adaptive Exponential Integrate-and-Fire Neuron Model“. In: *Advances in Neural Information Processing Systems 23*. Ed. by J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, pp. 1642–1650.

Moradi, Saber, Ning Qiao, Fabio Stefanini, and Giacomo Indiveri (2018). „A scalable multicore architecture with heterogeneous memory structures for Dynamic Neuromorphic Asynchronous Processors (DYNAPs)“. In: *IEEE Trans. Biomed. Circuits Syst.* 12.1, pp. 106–122.

Mountcastle, V B (1997). „The columnar organization of the neocortex“. In: *Brain* 120.4, pp. 701–722.

Mountcastle, Vernon B. (1957). „MODALITY AND TOPOGRAPHIC PROPERTIES OF SINGLE NEURONS OF CAT'S SOMATIC SENSORY CORTEX“. In: *Journal of Neurophysiology* 20.4. PMID: 13439410, pp. 408–434. DOI: 10.1152/jn.1957.20.4.408. eprint: https://doi.org/10.1152/jn.1957.20.4.408. URL: https://doi.org/10.1152/jn.1957.20.4.408.

Muller, Eilif, James A. Bednar, Markus Diesmann, Marc-Oliver Gewaltig, Michael Hines, and Andrew P. Davison (2015). „Python in neuroscience“. In: *Frontiers in Neuroinformatics* 9. ISSN: 1662-5196. DOI: 10.3389/fninf.2015.00011. URL: https://www.frontiersin.org/articles/10.3389/fninf.2015.00011.

Müller, Eric, Sebastian Schmitt, Christian Mauch, Sebastian Billaudelle, Andreas Grübl, Maurice Güttler, Dan Husmann, Joscha Ilmberger, Sebastian Jeltsch, Jakob Kaiser, Johann Klähn, Mitja Kleider, Christoph Koke, José Montes, Paul Müller, Johannes Partzsch, Felix Passenberg, Hartmut Schmidt, Bernhard Vogginger, Jonas Weidner, Christian Mayr, and Johannes Schemmel (2022). „The Operating System of the Neuromorphic BrainScaleS-1 System“. In: *Neurocomputing* 501, pp. 790–810. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2022.05.081. arXiv: 2003.13749 [cs.NE].

Müller, Eric Christian (2014). „Novel Operation Modes of Accelerated Neuromorphic Hardware“. HD-KIP 14-98. PhD thesis. Ruprecht-Karls-Universität Heidelberg. URL: http://www.kip.uni-heidelberg.de/Veroeffentlichungen/details.php?id=3112.

Ostrau, Christoph, Christian Klarhorst, Michael Thies, and Ulrich Rückert (2022). „Benchmarking Neuromorphic Hardware and Its Energy Expenditure“. In: *Frontiers in Neuroscience* 16. ISSN: 1662-453X. DOI: 10.3389/fnins.2022.873935. URL: https://www.frontiersin.org/articles/10.3389/fnins.2022.873935.

Pakkenberg, Bente, Dorte Pelvig, Lisbeth Marner, Mads J Bundgaard, Hans Jørgen G Gundersen, Jens R Nyengaard, and Lisbeth Regeur (2003). „Aging and the human neocortex“. In: *Experimental gerontology* 38.1, pp. 95–99.

Passenberg, Felix Constantin (2019). „Improving the BrainScaleS-1 place and route software towards real world waferscale experiments“. Master thesis. Ruprecht-Karls-Universität Heidelberg.

Petrovici, Mihai Alexandru (July 2016). *Form Versus Function. Theory and Models for Neuronal Substrates*. Springer Theses. Springer Cham, pp. XXVI, 374. ISBN: 978-3-319-39552-4. DOI: 10.1007/978-3-319-39552-4.

Potjans, Tobias C. and Markus Diesmann (2012). „The Cell-Type Specific Cortical Microcircuit: Relating Structure and Activity in a Full-Scale Spiking Network Modela“. In: *Cereb. Cortex* 24 (3), pp. 785–806. DOI: 10.1093/cercor/bhs358.

Rhodes, Oliver, Petruţ A. Bogdan, Christian Brenninkmeijer, Simon Davidson, Donal Fellows, Andrew Gait, David R. Lester, Mantas Mikaitis, Luis A. Plana, Andrew G. D. Rowley, Alan B. Stokes, and Steve B. Furber (2018). „sPyNNaker: A Software Package for Running PyNN Simulations on SpiNNaker“. In: *Frontiers in Neuroscience* 12, p. 816. ISSN: 1662-453X. DOI: 10.3389/fnins.2018.00816.

Rhodes, Oliver, Luca Peres, Andrew GD Rowley, Andrew Gait, Luis A Plana, Christian Brenninkmeijer, and Steve B Furber (2020). „Real-time cortical simulation on neuromorphic hardware“. In: *Philosophical Transactions of the Royal Society A* 378.2164, p. 20190160.

Sakmann, B and E Neher (1984). „Patch Clamp Techniques for Studying Ionic Channels in Excitable Membranes“. In: *Annual Review of Physiology* 46.1. PMID: 6143532, pp. 455–472. DOI: 10.1146/annurev.ph.46.030184.002323. eprint: https://doi.org/10.1146/annurev.ph.46.030184.002323. URL: https://doi.org/10.1146/annurev.ph.46.030184.002323.

Sanzeni, A., M. H. Histed, and N. Brunel (Mar. 2022). „Emergence of Irregular Activity in Networks of Strongly Coupled Conductance-Based Neurons“. In: *Phys. Rev. X* 12 (1), p. 011044. DOI: 10.1103/PhysRevX.12.011044. URL: https://link.aps.org/doi/10.1103/PhysRevX.12.011044.

Schemmel, Johannes, Sebastian Billaudelle, Philipp Dauer, and Johannes Weis (Dec. 2021). „Accelerated Analog Neuromorphic Computing“. In: pp. 83–102. ISBN: 978-3-030-91740-1. DOI: 10.1007/978-3-030-91741-8_6.

Schemmel, Johannes, Daniel Brüderle, Andreas Grübl, Matthias Hock, Karlheinz Meier, and Sebastian Millner (2010). „A Wafer-Scale Neuromorphic Hardware System for Large-Scale Neural Modeling“. In: *Proceedings of the 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1947–1950. DOI: 10.1109/ISCAS.2010.5536970.

Schemmel, Johannes, Johannes Fieres, and Karlheinz Meier (2008). „Wafer-Scale Integration of Analog Neural Networks“. In: *Proceedings of the 2008 International Joint Conference on Neural Networks (IJCNN)*.

Schemmel, Johannes, Andreas Grübl, Karlheinz Meier, and Eilif Muller (2006). „Implementing Synaptic Plasticity in a VLSI Spiking Neural Network Model“. In: *Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN)*. IEEE Press. DOI: 10.1109/IJCNN.2006.246651.

Schmidt, Dominik (2014). „Automated Characterization of a Wafer-Scale Neuromorphic Hardware System“. Master thesis. Ruprecht-Karls-Universität Heidelberg.

Schmidt, Hartmut, José Montes, Andreas Grübl, Maurice Güttler, Dan Husmann, Joscha Ilmberger, Jakob Kaiser, Christian Mauch, Eric Müller, Lars Sterzenbach, Johannes Schemmel, and Sebastian Schmitt (2023). „From Clean Room to Machine Room: Commissioning of the First-Generation BrainScaleS Wafer-Scale Neuromorphic System". In: *Neuromorphic Computing and Engineering*. DOI: 10.1088/2634-4386/acf7e4.

Schmitt, Sebastian, Johann Klähn, Guillaume Bellec, Andreas Grübl, Maurice Güttler, Andreas Hartel, Stephan Hartmann, Dan Husmann, Kai Husmann, Sebastian Jeltsch, Mitja Kleider, Christoph Koke, Alexander Kononov, Christian Mauch, Eric Müller, Paul Müller, Johannes Partzsch, Mihai A. Petrovici, Bernhard Vogginger, Stefan Schiefer, Stefan Scholze, Vasilis Thanasoulis, Johannes Schemmel, Robert Legenstein, Wolfgang Maass, Christian Mayr, and Karlheinz Meier (2017). „Neuromorphic Hardware In The Loop: Training a Deep Spiking Network on the BrainScaleS Wafer-Scale System". In: *Proceedings of the 2017 IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 2227–2234. DOI: 10.1109/IJCNN.2017.7966125. URL: http://ieeexplore.ieee.org/document/7966125/.

Schuman, Catherine D., Thomas E. Potok, Robert M. Patton, J. Douglas Birdwell, Mark E. Dean, Garrett S. Rose, and James S. Plank (2017). *A Survey of Neuromorphic Computing and Neural Networks in Hardware*. eprint: arXiv:1705.06963.

Schwarzenböck, Quirinus (2019). „Towards Balanced Random Networks on the BrainScaleS I System". Bachelor. Universität Heidelberg.

Song, Sen, Kenneth D Miller, and Larry F Abbott (2000). „Competitive Hebbian learning through spike-timing-dependent synaptic plasticity". In: *Nature Neuroscience* 3.9, pp. 919–926.

Splettstoesser, Thomas (2015). *Synapse schematic*. Accessed: 2023-08-22, Licence: CC BY-SA at https://creativecommons.org/licenses/by-sa/4.0/, Original by en:User:Thomas Splettstoesser. URL: https://commons.wikimedia.org/wiki/File:SynapseSchematic_unlabeled.svg.

Thommes, Tobias (2023). „Interconnect technologies for very large spiking neural networks". PhD thesis. Ruprecht-Karls-Universität Heidelberg.

Thomson, A. M., D. C. West, Y. Wang, and A. P. Bannister (2002). „Synaptic connections and small circuits involving excitatory and inhibitory neurons in layers 2-5 of adult rat and cat neocortex: triple intracellular recordings and biocytin labelling in vitro". In: *Cerebral Cortex* 12, pp. 936–953.

Tsodyks, M. and H. Markram (Jan. 1997). „The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability". In: *Proceedings of the national academy of science USA* 94, pp. 719–723.

Upton, Eben and Gareth Halfacree (Mar. 2017). „Meet the Raspberry Pi". In: pp. 11–22. ISBN: 9781119264361. DOI: 10.1002/9781119415572.ch1.

Van Rossum, Guido and Fred L. Drake (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace. ISBN: 1441412697.

Wehrheim, Malte (2019). „Reconstruction of Synaptic Weight on the Neuromorphic BrainscaleS-1 System“. Bachelor thesis. Universität Heidelberg.

Weidner, Jonas (2019). „Experiment Visualization and Simulations towards a Cortical Microcircuit on the BrainScaleS Neuromorphic Hardware“. Bachelor thesis. Universität Heidelberg.

Yik, Jason, Soikat Hasan Ahmed, Zergham Ahmed, Brian Anderson, Andreas G. Andreou, Chiara Bartolozzi, Arindam Basu, Douwe den Blanken, Petrut Bogdan, Sander Bohte, Younes Bouhadjar, Sonia Buckley, Gert Cauwenberghs, Federico Corradi, Guido de Croon, Andreea Danielescu, Anurag Daram, Mike Davies, Yigit Demirag, Jason Eshraghian, Jeremy Forest, Steve Furber, Michael Furlong, Aditya Gilra, Giacomo Indiveri, Siddharth Joshi, Vedant Karia, Lyes Khacef, James C. Knight, Laura Kriener, Rajkumar Kubendran, Dhireesha Kudithipudi, Gregor Lenz, Rajit Manohar, Christian Mayr, Konstantinos Michmizos, Dylan Muir, Emre Neftci, Thomas Nowotny, Fabrizio Ottati, Ayca Ozcelikkale, Noah Pacik-Nelson, Priyadarshini Panda, Sun Pao-Sheng, Melika Payvand, Christian Pehle, Mihai A. Petrovici, Christoph Posch, Alpha Renner, Yulia Sandamirskaya, Clemens JS Schaefer, André van Schaik, Johannes Schemmel, Catherine Schuman, Jae-sun Seo, Sadique Sheik, Sumit Bam Shrestha, Manolis Sifalakis, Amos Sironi, Kenneth Stewart, Terrence C. Stewart, Philipp Stratmann, Guangzhi Tang, Jonathan Timcheck, Marian Verhelst, Craig M. Vineyard, Bernhard Vogginger, Amirreza Yousefzadeh, Biyan Zhou, Fatima Tuz Zohora, Charlotte Frenkel, and Vijay Janapa Reddi (2023). *NeuroBench: Advancing Neuromorphic Computing through Collaborative, Fair and Representative Benchmarking*. arXiv: 2304.04640 [cs.AI].

Zenke, Friedemann and Wulfram Gerstner (2014). „Limits to high-speed simulations of spiking neural networks using general-purpose computers“. In: *Frontiers in Neuroinformatics* 8.76. ISSN: 1662-5196. DOI: 10.3389/fninf.2014.00076. URL: http://www.frontiersin.org/neuroinformatics/10.3389/fninf.2014.00076/abstract.

Zoschke, Kai, Maurice Guettler, Lars Boettcher, Andreas Gruebl, Dan Husmann, Johannes Schemmel, Karlheinz Meier, and Oswin Ehrmann (2017). „Full Wafer Redistribution and Wafer Embedding as Key Technologies for a Multi-Scale Neuromorphic Hardware Cluster“. In: *EPTC 2017*.

Acknowledgments

I would like to express my gratitude to the following people:

Dr. habil. Johannes Schemmel for taking over the supervision of my thesis and for providing all the necessary resources that made this work possible.

Prof. Dr. Wolfram Pernice for agreeing to undertake the role of the second reviewer for my thesis.

Prof. Dr. Ulrich Schwarz and Prof. Dr. Selim Jochim for participating in my examination committee.

Sebastian Schmitt for guiding me through the past years and teaching me everything there is to know about the BrainScaleS-1 system.

José Montes, Lars Sterzenbach, Andreas Grübl, Joscha Ilmberger, Christian Mauch, Maurice Güttler and Dan Husmann for helping me keeping the systems alive.

Eric Müller for teaching me how to write sustainable software.

The students I had the pleasure to work with and who assisted me in achieving experiments on the systems. By name: Felix Passenberg, Jakob Kaiser, Jonas Weidner, Moritz Hornung and Quirinus Schwarzenböck.

Christian Mauch, Eric Müller, Jakob Kaiser, Joscha Ilmberger, José Montes, Julian Göltz, Philip Spilger, Sebastian Schmitt and Yannik Stradmann for proofreading this manuscript.

All the other members of the Electronic Visions Group for the pleasant and collegial atmosphere.

Stefanie for being by my side the whole time.

My parents who have always supported me unconditionally.

Funding Statement

This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nos. 720270, 785907 and 945539 (Human Brain Project, HBP), from the European Union Seventh Framework Programme (FP7) under grant agreement no 269921 (BrainScaleS), and the Helmholtz Association Initiative and Networking Fund (ACA, Advanced Computing Architectures) under Project SO-092.

Statement of Originality (Erklärung):

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, November 27, 2023

.....
(signature)