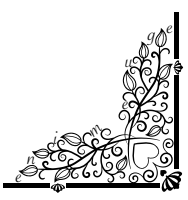
Department of Physics and Astronomy
Heidelberg University

Bachelor Thesis in Physics
submitted by

Carl Clemens von Randow

born in Hamburg



Neuron Calibration Using the ANANAS-System on BrainScaleS-1

This bachelor thesis has been carried out by

Carl Clemens von Randow

at the

Kirchhoff Institute for Physics

at

Heidelberg University

under the supervision of

Prof. Dr. Johannes Schemmel

Neuron Calibration Using the ANANAS-System on BrainScaleS-1

Carl Clemens von Randow

Abstract

Analog neuromorphic hardware allows the emulation of biological neurons by electronic circuits. These operate through analog voltages and currents which are made accessible by analog-to-digital converters. In order to enhance these analog readout capabilities for the used *BrainScaleS-1* system, the *ANANAS* boards have been developed and calibrated. In this thesis, they have been integrated into the BrainScaleS-1 system by adding the necessary software support. This allows to concurrently read out 84 analog channels, while simultaneously increasing the robustness of the readout and lowering the noise. This integration encompasses the measurement of the High Input Count Analog Neural Network chip (HICANN) source impedance, as well as the calibration of the 512 neurons of one HICANN. The calibrated neurons allow the hardware to be used for e. g. surrogate gradient experiments, which rely heavily on analog data. Furthermore, a clock drift between the HICANN and the analog recordings has been solved by using the same clock source.

Zusammenfassung

Analoge neuromorphe Hardware ermöglicht die Emulation von biologischen Neuronen durch elektronische Schaltungen. Diese funktionieren mittels analoger Spannungen und Strömen, welche durch analog-zu-digital Wandler ausgelesen werden. Um die analogen Auslesekapazitäten des benutzen *BrainScaleS-1* Systems zu verbessern, wurden die *ANANAS* Boards entwickelt und kalibriert. In dieser Arbeit wurden sie in das BrainScaleS-1 System integriert, indem die notwendige Software-Unterstützung geschrieben wurde. Dadurch können 84 analoge Kanäle parallel ausgelesen werden, ferner wird die Robustheit des Auslesens erhöht, sowie die Störgeräusche reduziert. Diese Integration umfasst sowohl die Messungen der High Input Count Analog Neural Network chip (HICANN) Ausgangsimpedanz, als auch die Kalibration der 512 Neuronen auf einem HICANN. Die kalibrierten Neuronen ermöglichen die Nutzung der Hardware beispielsweise für Experimente, die auf dem Surrogate-Gradient-Verfahren aufbauen, welches stark auf analoge Daten angewiesen ist. Zusätzlich wurde ein Auseinanderlaufen der Clock-Signale von HICANN und den analogen Aufnahmen gelöst, indem die gleiche Clock-Signal Quelle benutzt wird.

Contents

Chapter 1.

Introduction

The human brain is exceptionally powerful. For certain tasks (like pattern recognition), it outperforms any computer with comparable resources in several ways. These tasks are solved with an energy efficiency that is several magnitudes higher than those of conventional computers as well as at a higher speed. Furthermore, it is more resilient to noise and to changes to its underlying compute structures.

In today's world, where resources are scarce, trying to be as efficient as possible gains ever more focus. One approach to reach this efficiency is *neuromorphic computing*. It refers to building non von Neumann machines that emulate the biological brain by having dedicated electronic circuitry acting as artificial neurons [9].

One such system is the BSS-1 system, a wafer-scale neuromorphic hardware platform developed by the Electronic Vision(s) Group. This entails a large number of neurons being densely packed. In order to execute meaningful experiments on these neurons, calibrating the hardware is necessary. This is due to the fact that the emulation via a physical model leads inevitably to fixed pattern noise. This calibration is based on analog recordings of the membrane traces which were severely limited in the original analog readout system. Hence, a new readout system – the ANAlog Network Attached Sampling Board (ANANAS board) – was developed to address this restriction [4].

The objective of this thesis is to incorporate the ANANAS board into the wafer-scale system, to enhance the analog readout capabilities. The number of channels that can be read out in parallel as well as the signal quality improves by using the ANANAS system. This will allow for more complex experiments, especially ones that rely heavily on analog data.

These experiments in turn will lead to a better understanding of the learning processes taking place in the brain. Eventually, this knowledge can be used to build more efficient computing hardware inspired by biology, as well as leverage the attainment of new information, and possibly even treatments, for illnesses affecting the human brain.

Outline

This thesis is divided into five chapters.

Chapter 2 gives more information about the background of this thesis, its embedding into biology, and both the used hardware and software.

In Chapter 3, the methods of this thesis are outlined. It covers the calibration of the ANANAS boards, the measurement of the HICANN source impedances, the integration of the ANANAS boards into the full-wafer software stack, and its use in the calibration of

neurons.

The evaluation of the results is shown in Chapter 4. This ranges from synchronising the HICANN and ANANAS board clock signals, the quality of the obtained readout data, and the results of the neuron calibration.

Finally, Chapter 5 discusses the results and gives an outlook for future experiments making use of the developed results.

Chapter 2.

Background

This chapter begins by covering the biological background in Section 2.1. First, it is explained how neuromorphic computing aims to emulate the human brain. Then, the LIF neuron model is presented, as it is the model used in this thesis and can be realised on the hardware system BSS-1. The BSS-1 system is then introduced in more detail in Section 2.2.

2.1. Biological Background

Neuromorphic hardware is used to emulate spiking neural networks. These networks are composed of neurons, which are interconnected via synapses. In order to build neuromorphic hardware and choose suitable electronic circuits that replicate the behaviour of neurons and synapses, one needs to understand how biological neurons work. There exist multiple models which try to describe neuronal dynamics. Usually there is a trade-off between a model's accuracy and complexity. A notable model was described in 1952 by Alan Hodgkin and Andrew Huxley and is aptly known as the Hodgkin–Huxley model [3]. It consists of a set of non-linear ordinary differential equations, which means it approximates the biological dynamics relatively well, but is also quite complex. Therefore, there exist various other neuron models, one of them being the LIF neuron model, which is the one considered in this thesis and shown in detail in Section 2.1.1.

For the purpose of neuromorphic computing, a neuron is generally reduced to its essential parts, i. e. the parts that make up the firing mechanism. The central part of this is the membrane potential, which is typically around $-70\,\mathrm{mV}$ in its equilibrium state. It measures the difference in the potential between the inside and the outside of the cell. The neuron receives inputs from its dendrites, to which other neurons are connected. These incoming signals can affect the membrane potential and either increase it (so called *excitatory stimulation*) or decrease it (*inhibitory stimulation*). This is achieved by opening or closing ion channels in the membrane, which affect the concentration of ions inside and outside the cell membrane and thereby the membrane potential. If a neuron receives enough excitatory input that increases its membrane potential to a certain threshold (usually around $-55\,\mathrm{mV}$), the neuron *fires*. This means the neuron enters a depolarisation phase of about 1 ms, in which ion channels get opened and rapidly increase the membrane potential to around $40\,\mathrm{mV}$. Afterwards there is a repolarisation phase, which pulls the membrane potential down again and leads to a hyperpolarisation where the membrane potential is at around $-75\,\mathrm{mV}$. The period until the neuron reaches its equilibrium state again is called the *refractory time* of the neuron, and causes the neuron to have a lower probability to spike, due to the increased difference to the threshold. This sudden increase

and decrease of the membrane potential is called an *action potential* or *spike*, and travels along the axon. At the end of the axon are synapses, which connect the neuron to the dendrites of other neurons, where the spike acts as a new stimulus for all connected neurons. Information is primarily encoded in these spikes and their respective timings. But the membrane potential also holds some information, because it can be in a state of higher or lower excitation, compared to its equilibrium state.

2.1.1. The LIF Neuron Model

The LIF neuron model is a very simple approximation of the behaviour of a biological neuron , but this does not mean that there is no biological relevance [12]. On the contrary: Due to capturing the essentials without becoming overly complicated, the model is highly popular.

The equivalent circuit diagram for a LIF neuron can be seen in Figure 2.1. The figure shows CUBA synapses, which means that the inputs are implemented as input currents. The membrane potential of the neuron is modelled by the voltage across a capacitor with the capacitance C_m. Without any external input, the capacitor gets charged by the voltage source E_rest, which is connected to the capacitor via the leak conductance g_leak. The index *leak* refers to its biological counterpart of leakage currents through the membrane. At this rest potential, the system is in an equilibrium state, until some external input affects the neuron. These inputs can be either excitatory or inhibitory, modelled by their respective current sources I_exc and I_inh. The spiking mechanism is realised by a comparator, which compares the membrane potential to the threshold voltage V_thres. When the membrane gets stimulated enough, a digital spike signal is sent to all connected neurons. Additionally, for the duration of the refractory time period τ_refrac, the membrane potential is clamped to the reset potential V_reset. This is achieved by shorting the capacitor to the corresponding voltage source.

The differential equation that governs the LIF neuron dynamics is given by

$$C_\mathrm{m}\frac{\mathrm{d}U(t)}{\mathrm{d}t} = -g_\mathrm{leak}\left(U(t) - E_\mathrm{rest}\right) + I_\mathrm{exc}(t) + I_\mathrm{inh}(t), \tag{2.1}$$

and the spike condition can be expressed as

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

Aside from the CUBA synapses, there is also the possibility to approximate the inputs using COBA synapses. In that case the current sources I_exc and I_inh are replaced by the two time-dependent conductances $g_\mathrm{exc}(t)$ and $g_\mathrm{inh}(t)$, which connect the membrane potential to the excitatory and inhibitory reversal potentials, $E_\mathrm{rev}^\mathrm{exc}$ and $E_\mathrm{rev}^\mathrm{inh}$, respectively.

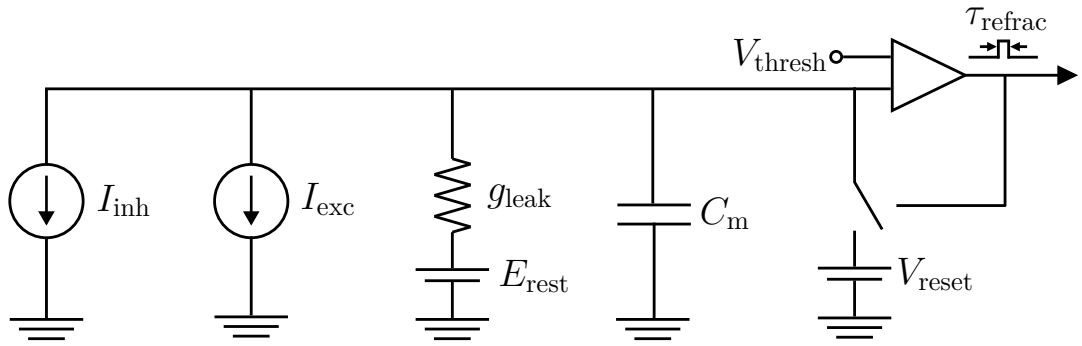


Figure 2.1.: Equivalent circuit diagram of a CUBA LIF neuron. The membrane voltage across the capacitor C_{m} is connected to the resting potential E_{rest} via the leak conductance g_{leak}. It receives synaptic input from the two current sources I_{exc} and I_{inh}. If the membrane potential reaches the threshold V_{thresh}, it elicits a spike and is set to the reset potential V_{reset} for the duration of the refractory period τ_{refrac}. In case of synapses, the two current sources are replaced by two voltage sources modelling the excitatory and inhibitory reversal potentials $E_{\text{rev}}^{\text{exc}}$ and $E_{\text{rev}}^{\text{inh}}$, which are then connected to the capacitor via two time-dependent conductances $g_{\text{exc}}(t)$ and $g_{\text{inh}}(t)$. Figure taken and adapted from [17].

2.2. The BrainScaleS-1 System

In this thesis, the neuromorphic hardware platform BrainScaleS-1 (BSS-1) is used. It implements the AdEx neuron model, which is more complex compared to the LIF model, and thereby allows to model additional neuron dynamics. But since the former contains the latter as a subset, for the purpose of this work, only the LIF model aspects of the BSS-1 system are used. Due to the fact that the electronic circuits evolve much faster compared to biological neurons, there is an acceleration factor of 10^4 between wall-clock time and the biological time domain. Thus, one biological year can be emulated on hardware in less than 53 minutes.

Since the objective of this thesis is to replace the old analog readout system used in BSS-1 with the new ANANAS boards, this section is split into four parts. Section 2.2.1 gives an overview of the whole hardware system, while Section 2.2.2 looks at the previous analog readout system initially implemented in BSS-1. In Section 2.2.3, the ANANAS board is presented in detail, followed by its calibration routine outlined in Section 2.2.4.

2.2.1. The Wafer Module

The BSS-1 system refers to a wafer-scale neuromorphic hardware system and is depicted in Figure 2.2.

The dimensions of the full module are 50 cm x 50 cm x 15 cm, whereas the silicon wafer

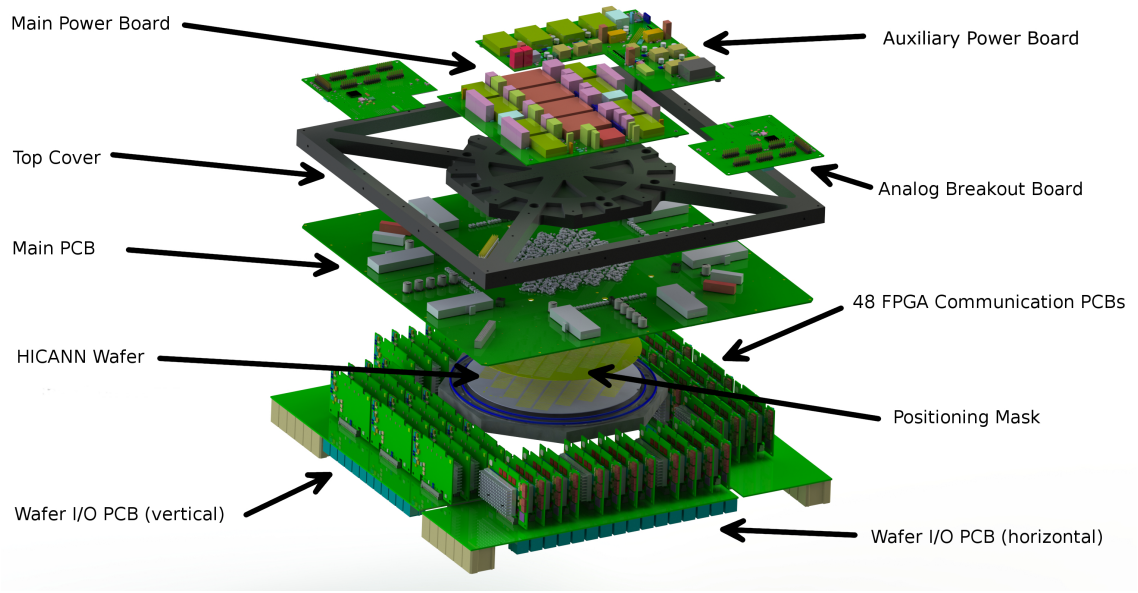


Figure 2.2.: Exploded view of a wafer module. The wafer containing the HICANN chips is located in the middle and connected to the main PCB via elastomeric connectors which are aligned by the positioning mask. The main PCB provides the connections to the power modules (main and auxiliary power boards) and analog breakout boards on the top, as well as the connections to the FCPs on the bottom. These FCPs each contain a Kintex-7 FPGA and are also connected to the wafer I/O PCBs. From there, an Ethernet connection to the host computers is established. The plot shows a module with the old digitiser system. For the new system, the analog breakout boards are replaced by the ANANAS board boards for improved analog readout capabilities. Courtesy of Dan Husmann.

located at its heart has a diameter of 20 cm. The wafer is made up of 384 HICANN chips [16], each containing 512 neurons and up to 114,688 synapses. This means the full wafer could technically emulate over 196 thousand neurons with over 44 million synapses. In practice, this number is difficult to obtain due to hardware defects or errors introduced in the manufacturing process. These components can be identified and blocked from further usage with the threshold for declaring a component good or bad adjustable to the desired quality in the behaviour of the hardware. The 384 HICANNs are organised into 48 so-called *reticles* meaning that in turn, every reticle comprises eight chips. Each reticle features two analog output lines which can be used to read out membrane traces from any neuron. This configurability is achieved by multiplexing the readout connections.

The positioning mask ensures the wafer is properly aligned with the main PCB which in turn is connected to the main power board (PowerIt) and the auxiliary power boards (AuxPwr), supplying the whole system with power. On the top side of the main PCB are also the analog breakout boards (AnaBs) located. These AnaBs are part of the old readout system. For the new system, they are replaced by the ANANAS boards. Except for comparisons between the two systems, the majority of this thesis considers and shows the new system utilising the ANANAS boards. On the bottom side of the main PCB are 48 FCPs which contain a Kintex-7 FPGA each. These FCPs allow the FPGA to communicate with the main PCB as well as with the wafer I/O PCBs (WIOs), which establish the connection to the host computers. The need for FPGAs arises from the fact that conventional CPUs do not allow to control the system with a precision of individual clock cycles. This fine-grained interaction with the neuromorphic hardware is necessary due to the fast evolving neuron circuits which allow emulating biology with an acceleration factor of 10^4 compared to biological time.

Additionally, there is a RaspberryPi which enables power control of individual parts of the system, as well as monitoring of power and temperature of the system. It also regulates the cooling fans to keep the wafer module at a constant operating temperature. If the systems are in danger of being damaged by overheating, the RaspberryPi can initiate an automatic emergency shutdown of the wafer module while simultaneously setting the fans to their maximum power.

2.2.2. Previous Analog Readout System

The original implementation of the BSS-1 system features two analog breakout boards: one master board, which is connected to the RaspberryPi, and one slave board. They route the analog signals off the wafer into a digitiser crate containing 12 FlySpis. On each FlySpi, there is one Spartan-6 FPGAs as well as a 12 bit ADC [5] with a sampling frequency of up to 125 MHz. This ADC converts the analog signals to digital data which is then made available to the host computer via a Small-Form-Factor PC (NUC). Each of the AnaB boards allow the routing of six analog signals simultaneously, thus 12 parallel analog signals per wafer module can be obtained.

However, there is another limitation; the wafer modules are mounted in a standard 19-inch rack, with four modules per rack, along with fan boards for cooling and the digitiser crate, which also contains the RaspberryPi and the NUC. This means that four wafer modules have to share the analog readout capabilities of one digitiser crate. One wafer module has 96 analog signals, given 48 reticles with two analog lines each, which can all be connected to the ADC crate thanks to one 8:1 analog input multiplexer per FlySpi, which allows it to cover eight channels. Nonetheless, only 12 signals can be recorded concurrently. In this configuration, only one out of four wafer modules can return analog data though. Furthermore, the spatial proximity of the recorded neurons is limited by the multiplexers. The wafer is divided into six areas consisting of eight reticles each. Two FlySpis are connected to each of these areas, meaning that at most two membrane traces can be retrieved from each area. Thus, experiments using all 12 signals need to span almost the whole wafer.

It is also possible to use additional 4:1 multiplexers on the AnaB boards which allow the number of inputs needed for each wafer to be reduced from 12 to three. That way, the whole rack can be connected to the ADC crate at the price of having only three analog signals in parallel from one module.

The physical distance from the wafer system to the ADC crate via the AnaBs makes the analog signals quite susceptible to noise. Despite using shielded ribbon cables, the noise is still a non-negligible problem affecting the quality of the measured traces.

2.2.3. The ANANAS board

In contrast to this possibility of recording three analog signals per wafer using the old system, a module equipped with ANANAS boards fares much better. Here, as many as 84 concurrent channels can be recorded, which is a 28-fold increase. The ANANAS board was designed for this very purpose, as well as improving the quality of the recorded traces – despite the same 12 bit resolution – by being more robust to noise, by not requiring long cables from the AnaB boards to the ADC crate. A photograph of the ANANAS board can be seen in Figure 2.3, and in-depth information about the board can be found in [4].

On each module with the new readout system, two ANANAS boards are mounted, replacing the two AnaB boards. In the same way one of the AnaB boards was connected to a RaspberryPi and thereby the master board, in the new system one of the ANANAS boards is connected to the RaspberryPi via a pin header on the upper end of the board. The main control unit of the ANANAS board is the FlySpi which was already the essential piece in the ADC crate. There are 48 input channels per board so that the 96 signals from the wafer can be fully connected to the two ANANAS boards.

The 48 inputs are connected to a high precision 20 bit 64-channel integrating Low-speed analog-to-digital converter (lsADC) [7] with a sampling rate of up to 3.125 kHz, in this thesis operated at 2.325 kHz. It can be used to measure DC voltages with a high accuracy

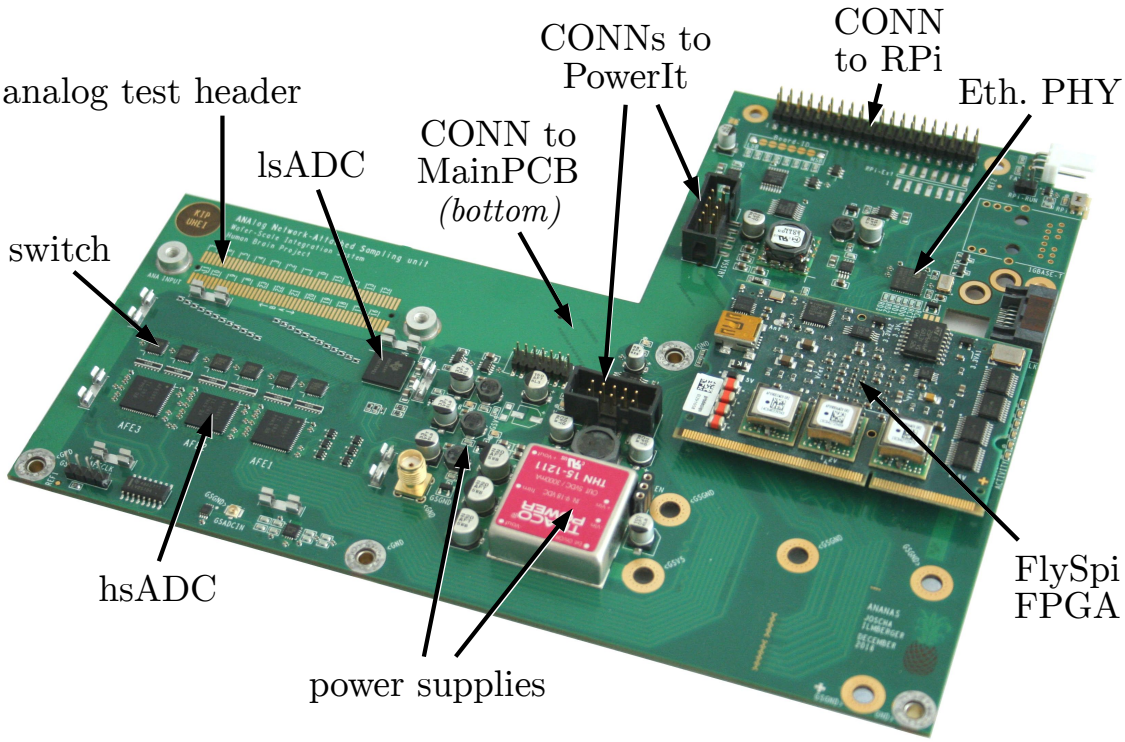


Figure 2.3.: Photograph of the ANANAS board. On the right hand side is the FlySpi FPGA, the main control unit of the ANANAS board. The ADCs are located on the left: One high precision lsADC for precise DC measurements and three hsADCs for recording neuron membrane traces. The hsADCs can be enabled by the analog switches together with the 50 Ω termination. The connection to the wafer via the main PCB is located on the bottom side of the board (i. e. not visible). The analog signal lines can (e. g. for debugging purposes) also be directly accessed from the top of the board via the analog test header, which is also used to calibrate the board. Two connections to the PowerIt, in conjunction with the on-board power supplies, deliver power to the ANANAS board itself as well as to the attached RaspberryPi. Since this picture is taken of a prototype board it lacks the Ethernet jack on the top right hand corner, near the pin header connecting the RaspberryPi. This Ethernet interface connects the ANANAS board directly to the compute cluster, as opposed to the indirect route via the NUC in the old ADC crate. Below this is a SATA connector allowing to supply an external clock signal. Image taken from [4].

due to its 4.87 MΩ input impedance. However, its sampling frequency is too low to resolve the neuron dynamics, which require around 4.3 MHz [4], so it cannot sensibly be applied to measure membrane traces. Its uses lie for example in calibration routines or debugging purposes.

By closing a corresponding analog switch, each channel can be terminated with 50 Ω as well as recorded by a 12 bit 16-channel High-speed analog-to-digital converter (hsADC) [6]. These hsADCs are differential flash ADCs, which measure the input differentially against a so-called Common mode voltage (CM) with a high sampling rate of up to 32.5 MHz, in this thesis operated at 31.25 MHz. In order to digitise all 48 input signals, each ANANAS board has three hsADCs assembled. The 16 channels of one hsADC are organised into two so-called *trigger groups* or *slices*. Each slice receives a different trigger signal from the wafer which starts the analog recording process.

Unfortunately, the input voltages affect the CM by coupling in via the internal biasing resistor, so that the CM is not constant but rather a function of time. Due to the relative nature of the measurements, any variations in the reference voltage are propagated directly onto the data. In order to correct for this, a channel needs to be dedicated to keeping track of how the CM varies over time. Each hsADC generates its own CM, so multiple channels per ananas board need to be reserved. Technically, one channel per hsADC would suffice, resulting in 90 useable channels for collecting traces. However, this would require substantial software changes. Depending on which implementation approach would be chosen for supporting this, resource allocation would become more complicated or parallel usage of a module would be hindered. In addition, using a tracking channel from the same hsADC but from a different slice would introduce a phase shift between the signals. It was found that this phase shift would be acceptable in the magnitude of the error [14], but it was decided to instead pursue the approach of using one CM tracking channel per slice. This merely sacrifices an additional three channels per ANANAS board for the benefit of cleaner and less complex code, thus preventing future problems as well as not introducing the inaccuracy of the phase shift. Taken together, this leads to the readout capabilities of 84 concurrent analog signals out of the 96 connected channels per wafer.

The analog voltages coming from the wafer are in the range of 0 V to 1.8 V, but the hsADC has an input range of only $V_{\text{range}} = 1\,\text{V}$ [6]. Consequently, the input signals need to be scaled to fit the input range. are terminated by the This is achieved by a voltage divider formed by the HICANN source impedance R_{S} and the terminating resistor R_{T} on the ANANAS board. Because both values are on the same order (50 Ω), the voltage gets roughly halved, leading to an membrane voltage range V_{m} between 0 mV and 900 mV. In addition to this scaling of the input voltages, it also needs to be shifted so that the input range is symmetrically distributed around the CM. The hsADC has an intrinsic value for the CM of $V_{\text{CM}} = 1.6\,\text{V}$. By operating the hsADCs on a shifted ground level of $V_{\text{GS}} = -1.15\,\text{V}$, the CM lies in the centre of the expected voltage range and therefore covers all voltages of interest. Thus, the final input range of the hsADC can be calculated

as (cf. [4] and [14]):

$$V_{\text{hsADC,in}} = V_{\text{CM}} + V_{\text{GS}} \pm \frac{V_{\text{range}}}{2} = (450 \pm 500)\,\text{mV}. \tag{2.3}$$

The ANANAS board is connected directly to the compute cluster via an Ethernet jack located on the top right of the board. This means it is no longer necessary to route the analog signals via the AnaB board to the ADC crate, digitise them using the FlySpi, and then transfer them via the NUC to the host computer. Instead, the signals are digitised directly on the board, buffered by the FlySpi's RAM and sent to the cluster.

Next to the Ethernet port is a SATA connector which allows the ANANAS board to use an external clock signal rather than using the on-board oscillator. This is of special interest, as the previous analog readout system and the HICANNs are running on separate clocks, which, due to e. g. manufacturing imperfections, were slowly drifting apart significantly affecting analog recordings of long-duration experiments. Using an extended calibration, the clock signals can be aligned in software at the cost of introducing computational overhead as well as commissioning efforts and this still does not allow for a perfect alignment.

2.2.4. ANANAS Calibration

The first step on the road of commissioning the ANANAS board is the calibration of the board itself. This is achieved by supplying it with known voltages and measuring the output of the corresponding ADCs in order to characterise their behaviour. Once this relationship has been found and saved, it can be fetched for the inverse operation of converting the output of one of the ADCs back to the voltage signal of interest.

The experimental setup of the calibration procedure can be seen in Figure 2.4. The user controls everything from the host PC through a connection to the ANANAS board via the compute cluster and the front end *Helvetica*. A sourcemeter [8] is used to generate precise known voltages which are transferred to the ANANAS board via the analog input test adapter and a series resistance of $R_{\text{S,ad}} = 49.85\,\Omega$, imitating the HICANN output impedance. The output of the different ADCs is recorded, which allows to calculate a conversion that maps the input voltages to the raw ADC outputs, and by inverting this function, the necessary transformation of raw ADC output values to an input voltage in Volts.

The calibration routine has been developed in [14] and an initial automated version has been implemented as part of [15]. It has been improved in [13], but is briefly summarised again here. For more detailed information, the reader is referred to the cited sources.

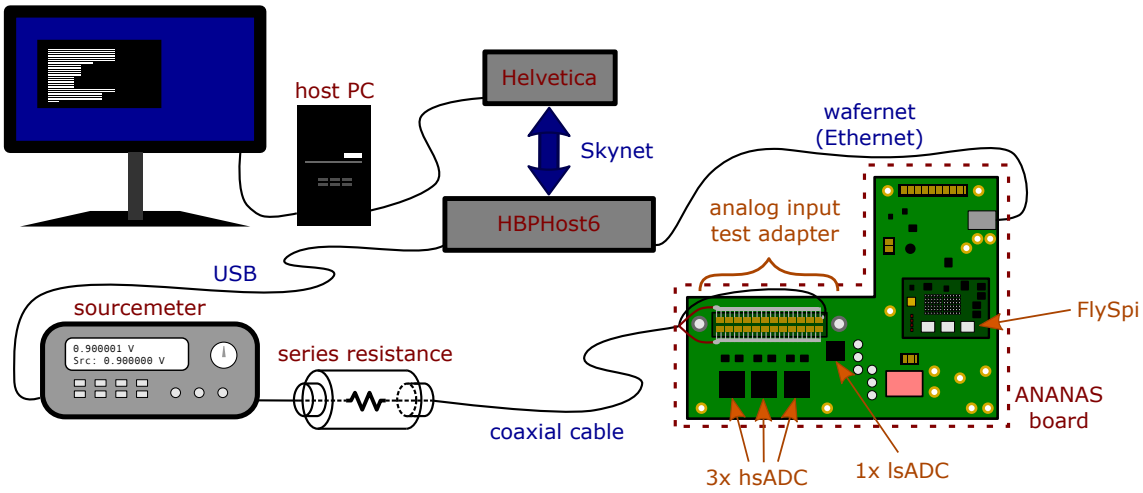


Figure 2.4.: The experimental setup of the ANANAS board calibration procedure. On the right hand side, the ANANAS board is shown with its three hsADCs and one lsADC, as well as its FlySpi. The input voltage is generated by the sourcemeter and connected to the ANANAS board via the $R_{\mathrm{S,ad}} = 49.85\,\Omega$ series resistance and the analog input test adapter. The user controls everything from the host PC, which is connected to the front end *Helvetica*, providing access to the compute cluster via the group's *Skynet* subnet. The cluster node (in this case *HBPHost6*) then controls the sourcemeter through a USB connection as well as configures and communicates with the ANANAS board through Ethernet, via the group's *wafernet* subnet. The recorded data is stored in the FlySpi's RAM during the measurement process and then read out at the end of the experiment via the Ethernet connection.
The sketch was created by Simon Rosenkranz, taken from his Bachelor's Thesis [14] and slightly modified.

Calibration of the lsADC

First, the lsADC is calibrated by an unterminated (i. e. open switches) voltage sweep of the whole input range from 0 V to 1.8 V. The output of the lsADC is recorded and a linear function is fitted to the data using a linear least-squares fit.

With this characterisation of the lsADC, a terminated voltage sweep of the input range is executed with the closing of the switches creating a voltage divider between $R_{\mathrm{S,ad}}$ and R_{T}. This allows an accurate determination of the terminating resistor values because the ratio of the sourced input voltage and the measured voltage of the lsADC is the same as the ratio of the two resistances.

Calibration of the hsADC

In contrast to the lsADC, the calibration of the hsADC is more complex due to the crosstalk between channels and the subsequent need for a dedicated tracking channel of the common mode variation, as well as the operation on the shifted ground level. All fits are done as linear least-squares fits.

The routine consists of the following five steps:

1. Baseline recording: All of the switches are kept open and all channels are measured without any input. The mean value of the recorded trace is saved as each channel's *baseline*.
2. Voltage sweep: For every channel a voltage sweep from -0.2 V to 1.8 V is executed and the data from all channels saved. Only the driven channel's switch is closed; all others are kept open as tracking channels.
3. common mode correction: For every driven channel the same procedure is performed: First, from the seven CM tracking channels their respective *baseline* is subtracted, yielding only the CM *variation*. This variation is fitted with a linear function, and the seven sets of fit parameters are averaged. Next, the driven channel's trace gets corrected for the CM influence by subtracting a linear function with the averaged parameters obtained from the tracking channels, whilst taking the factor of $\frac{5100}{5000}$ into account. Lastly, the sourced voltages returned by the sourcemeter need to be transformed to the actual input voltage that gets applied to the hsADC. On the one hand, the voltage divider of $R_{\mathrm{S,ad}}$ and R_{T} needs to be accounted for by multiplying the sourced voltages with a factor of $\frac{R_{\mathrm{T}}}{R_{\mathrm{S,ad}}+R_{\mathrm{T}}}$. On the other hand, in the same way as the $\frac{5100}{5000}$ factor needs to be considered for the CM influence, we also need to divide the source meter voltages by a factor of $\frac{5075}{5000}$ now.
4. Fitting: The fourth step is fitting the corrected data to a linear function after first cutting off the saturated areas in the beginning and in the end. This yields the gain parameter G of the hsADC transformation and also the offset V_0 relative to global ANANAS ground. Additionally, by calculating $V_{\mathrm{SM},0} = -\frac{V_0}{G}$, the voltage value can

be found for which the hsADC measures no difference to the CM value. By shifting the sourced voltages by this value, the hsADC internal coordinate system is entered. It is desired to obtain the hsADC internal offset which gets calculated in the next step.

5. HSADC INTERNAL OFFSET CALCULATION: In order to find the hsADC internal offset, the mean value of all 16 $V_{\mathrm{SM},0}$ values is calculated and then subtracted from each $V_{\mathrm{SM},0}$ value to obtain the remaining offset, which is the hsADC internal offset value.

Storing the calibration parameters

Continuing the work done in [13], a bug has been discovered and fixed, resolving the systematic differences in the quality of the hsADC calibration. Furthermore, the obtained calibration parameters are now being stored by the existing tool `calibtic` which is used throughout the software stack for storing, loading and applying calibrations.

Chapter 3.

Methods

In this chapter the methods employed in this thesis are outlined. The goal of this work is to commission the ANANAS board as part of the full wafer system. The analog nature of the HICANNs results in every chip being subject to slight manufacturing differences or defects, leading to a variation of behaviour and quality across the whole wafer. Thus, it is necessary to calibrate all operated neuron circuits to get good results when executing experiments. This requires the readout of the membrane traces, i. e. analog signals.

The first step towards this is the characterisation of the ANANAS board itself which has already been covered in Section 2.2.4.

The aforementioned voltage divider between the termination on the ANANAS board and the source impedance of the HICANN defines the relation of the measured voltage on the ANANAS board and the actual output signal of the wafer. Hence, the accurate measurement of the HICANN source impedances is the next step (Section 3.1).

In order to be able to run experiments including membrane traces, the ANANAS board needs to be incorporated into the software and work together with the full wafer stack. This integration into the existing framework throughout the different software layers is addressed in Section 3.2.

Finally, in Section 3.3 all the necessary requirements described beforehand are combined and the actual procedure of running a neuron calibration using a wafer module equipped with ANANAS boards is shown.

This thesis extends work done by numerous other people, especially [14], [4], and [15].

3.1. HICANN Source Impedance

In this section, the process of measuring the HICANN source impedance is outlined. It has also been described in [14] already including the derivations of the underlying equations. It is very similar to the measurement of the terminating resistors on the ANANAS board, since the same principle is leveraged. While in the case of the terminating resistors on the ANANAS board the resistor $R_{\mathrm{S,ad}}$ imitating the HICANN source impedance is known precisely and used to calculate R_{T} via the voltage drop measured with the lsADC, the whole procedure is executed again for the determination of the HICANN source impedances. The only difference is that the position of the known and unknown resistors are swapped because now the terminating resistors on the ANANAS board are known and used to deduce the source impedance values. Two measurements with the lsADC are done, one of them with the termination active ($V_{\mathrm{lsADC,on}}$), and the other one with the termination deactivated ($V_{\mathrm{lsADC,off}}$). Then the HICANN source impedance R_{S} can be calculated by

the following equation:

$$R_{\mathrm{S}} = R_{\mathrm{T}} \cdot \left(\frac{V_{\mathrm{lsADC,off}}}{V_{\mathrm{lsADC,on}}} - 1 \right). \tag{3.1}$$

For its derivation, the reader is referred to [14]. Additionally, a small correction for the effect of the CM coupling in on the terminated measurement has been included in the procedure, slightly changing the equation. It is a small effect on the order of 1% at most and can be neglected in principle but was added as a potential cause for differing results to the values presented in [14].

3.2. ANANAS Integration

In order to be able to use the ANANAS board to read out analog data from a wafer module, it needs to be integrated into the software stack. Due to the large system size and high complexity, the controlling software is equally complex to ensure that everything works properly.

The most relevant parts of the software for running an experiment, as well as for the neuron calibration, can be seen in Figure 3.1. When a user wants to run an experiment on the hardware, they specify the network model with the help of the *PyNN API* [2]. This description of the network model is represented by a *PyHMF Container*, which the mapping and routing tool (marocco) takes as input. Marocco translates the model into a valid hardware configuration by mapping it to the available resources, taking the block-listing of defective components into account and creating a Stateful hardware abstraction layer (StHAL) container, representing the hardware configuration. The back end directly accessing the hardware is Hardware abstraction layer back end (HALbe), which runs the experiment, collects the results and sends them back through the upper layers to the user. If desired, the user can check the realised network on hardware by looking into the mapping results stored by marocco.

The purpose of the cake repository is the calibration of neurons (cf. also Section 3.3). It directly interacts with StHAL, configuring the hardware in the different ways used to calibrate the system, which in turn calls HALbe again to execute the measurements. This calibration finds the relationships between the different DAC values that can be set to configure the hardware and the corresponding realised properties (e. g. voltages, currents) of the hardware circuits. The calibration results are stored using the calibtic framework, which can also be used for the retrieval of the calibrations and the application to the measured values.

In order to incorporate the ANANAS board into this software stack, multiple layers needed to be extended to support both analog readout systems. Some functionality could be reused from the old system, or implemented in a similar way, whereas other things needed to be implemented from scratch, e. g. due to the new procedure of having a CM tracking channel which is needed for all measurements. In earlier works (e. g. [1]), low-level

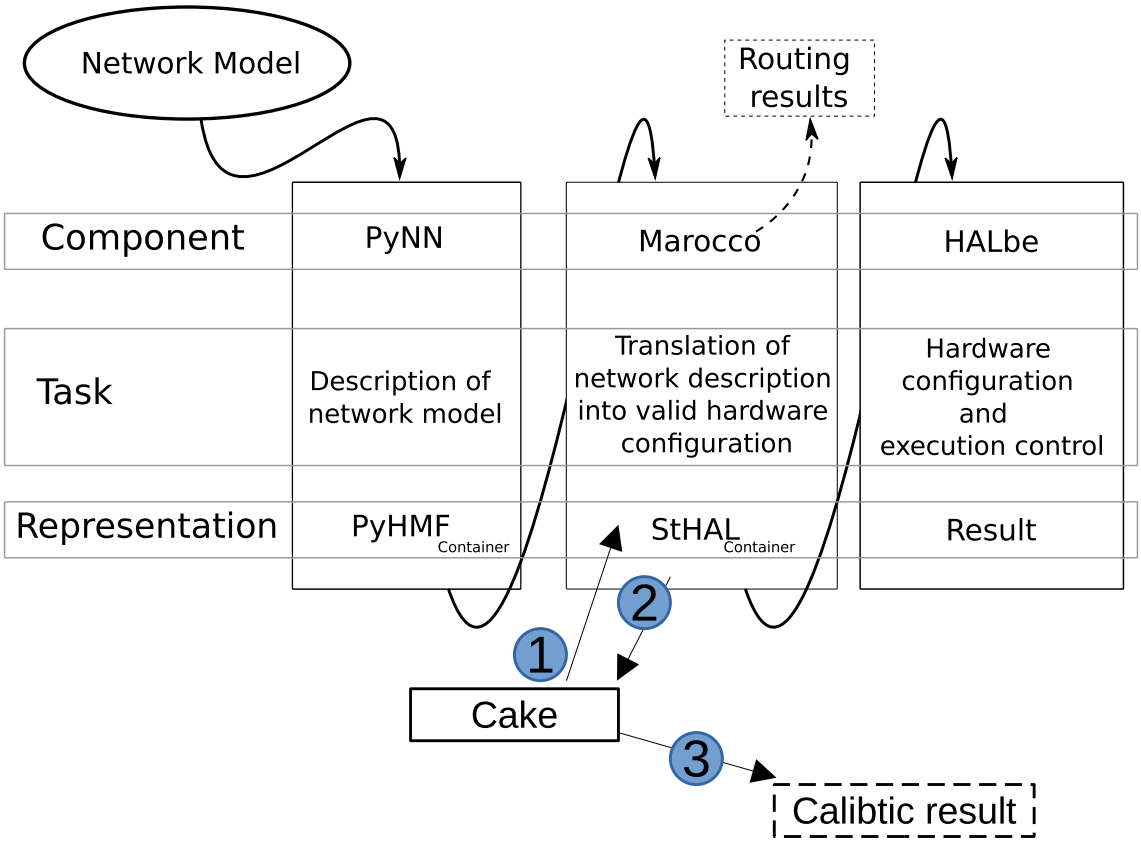


Figure 3.1.: The software stack enabling the use of the BSS-1 system. The user describes the network model using the *PyNN API* [2] and this description gets represented as a *PyHMF Container*. This is processed by *marocco*, creating a valid hardware configuration for the desired network model in form of a *StHAL Container*. The direct hardware access is executed by *HALbe* which then propagates the results back up the layers to the user. If the user wants to see how the network model was translated to the hardware configuration, the routing results stored by *marocco* can be checked. The neuron calibration is orchestrated by the repository *cake* which directly calls *StHAL*. After the execution on hardware via *HALbe*, *StHAL* sends the results back to *cake*, from where they are stored using the *calibtic* framework.
Courtesy of Hartmut Schmidt, who adapted it from [11].

support for the ANANAS board has been developed in form of the *anacoma* repository, and the necessary support for *anacoma* in HALbe (as well as exposing the functionality to StHAL) was partially implemented. In the course of this thesis, further changes to *anacoma*, HALbe and StHAL have been made, as well as supporting the ANANAS board in cake. This allows the new readout system to be used for neuron calibrations. Furthermore, support for the ANANAS board in marocco was implemented.

3.3. Neuron Calibration

Since every neuron circuit is a different piece of hardware, they display variations, which can stem, for example, from the manufacturing process, or some improper operation or handling. Since it is desirable to have the systems behave in a certain way when configuring them in a specific manner, their behaviour is observed and characterised during the calibration process. This enables reliable control of the system and allows to run the same experiment on different parts of the wafer or even a whole different module, without having to expect that the results are suddenly completely different.

The neuron calibration is orchestrated by the cake repository. It contains `python` code running the calibration for a single HICANN. Several HICANNs can be calibrated in parallel, the maximum number depending on the respective readout system limitations, i. e. 12 for the old system and 84 for the new system. However, the implementation of calibrating a full wafer module in parallel is yet to be done – this thesis focuses on a single HICANN calibration. While the calibrations of the particular neuron parameters build on each other, each parameter is calibrated in independent steps, sweeping the parameter range. The specific measurement points for the sweep can be configured but in any case the obtained transformation will be used for the whole parameter range. It is therefore advisable to keep this mind and try to use calibrations which used the same or similar parameter regions to the one of the experiment the calibration is applied to. During the parameter sweep, all other parameters are set to fixed, appropriate values. The calibration routine allows for interrupts by storing the results of a calibrated parameter as `hdf5` files before proceeding with the following parameter. That way in the worst case, only the current measurement will have to be started anew. Thus, intermediate results can be checked at will.

In Figure 3.2, the general structure of a calibration with cake can be seen. The user defines which calibrations to run and can adjust settings like the parameter ranges in a `config.py` file. This file is given the *CalibrationRunner* as input which then creates a *CalibrationUnit* for every parameter that will be calibrated. The CalibrationUnit spawns an *Experiment* instance which executes the measurements for the given parameter and extracts the desired features, like spike times or mean voltages of a trace. These measurement results are then fed into a *Calibrator* object, which interprets them, and creates the calibtic transformations. After every measurement, the CalibrationUnits are saved to

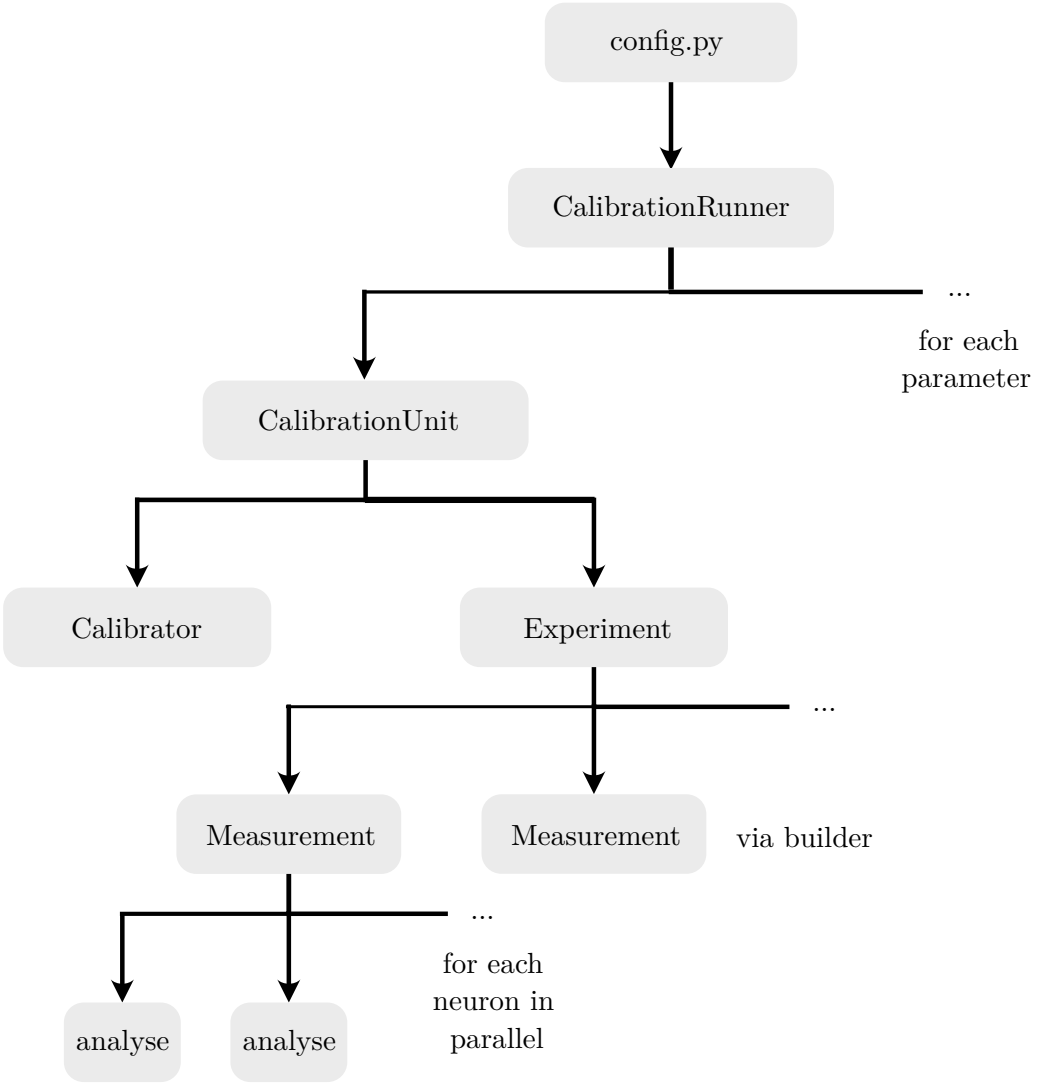


Figure 3.2.: Overview of the general structure of cake. The configuration (e. g. what parameters to run and which parameter ranges to use) is contained in a `config.py` file. The *CalibrationRunner* reads this file, iterates over all specified neuron parameters and creates a *CalibrationUnit* for each of them. Every CalibrationUnit first creates an *Experiment* instance which runs the required measurements and analyses them (e. g. extract spikes). These finished experiments are then handed to a *Calibrator* which interprets the data and generates the final calibration transformations.
Courtesy of Florian Fischer.

`hdf5` files and can be loaded and resumed if they are interrupted, having to redo only one (unfinished) measurement at most.

Chapter 4.

Evaluation

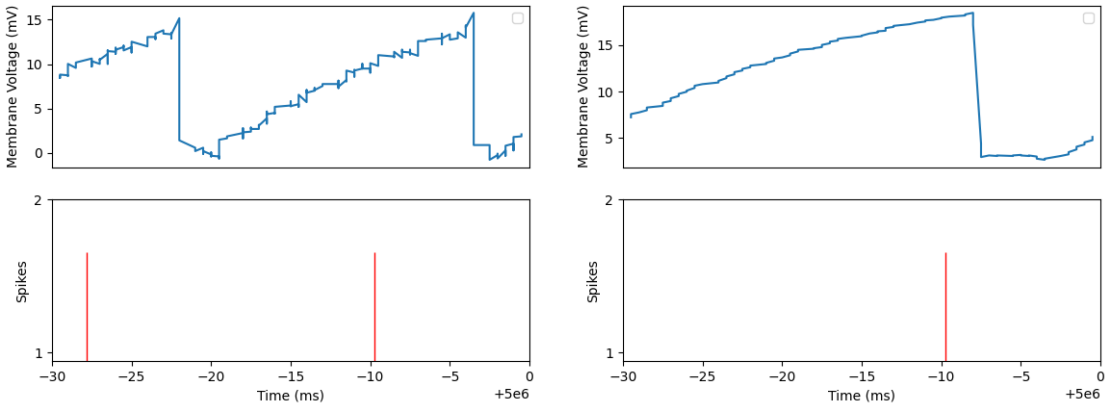
After covering the methods employed in the framework of this thesis in the previous chapter, this chapter shows the obtained results and evaluates them. First, in Section 4.1 the topic of the drift between the clock signals of the old analog readout system and the HICANN is addressed. Next, the noise of the recorded analog signals using the ANANAS boards is shown, as well as a comparison to the noise observed with the old system (Section 4.3). Concluding the chapter with Section 4.4, the results of the neuron calibration on an ANANAS module are presented.

4.1. Clock Distribution

The ADC on the FlySpi board converting the data in the old system is driven by a clock signal of 96 MHz received via USB. The HICANN, on the other hand, is supplied by a 125 MHz clock signal generated on one of the WIOs and distributed to the other WIOs, driving the Kintex-7 FPGAs on the FCPs, which in turn drive the HICANN. These are slowly drifting apart, e. g. due to manufacturing imperfections, which means the analog traces and the digital spikes are no longer perfectly aligned. This effect increases with the experiment duration as the drift is constant, so the difference becomes larger over time. With the help of an extended calibration, it is possible to increase the alignment of the two different clock signals which mitigates the problem to a degree that makes it useable. But there still remains some error, and it requires additional resources.

The ANANAS boards feature a SATA connector which allows to provide an external clock signal to be used. It also has an on-board oscillator capable of generating a clock signal which can be used for stand-alone operation, for example during the calibration routine. But to overcome the drifting clock signals, it is desirable to operate the ANANAS board on an external clock, specifically the HICANN clock. For this purpose, a small fan-out PCB was manufactured. The clock signal is generated on one of the WIOs, and because the four WIOs are connected in a ring, the clock can be distributed to the whole system. One of these small cables connecting two WIOs is replaced by the clock distribution PCB, which connects the two WIOs and additionally makes the clock signal available via four SATA connectors.

Such a solution allows for perfectly synchronized HICANN clock and ANANAS board sampling frequency, even if the absolute clock frequency drifts due to effects like temperature. To assess the correct operation of the clock distribution, we utilized a test in which a generator of events in the HICANN (called background generator) emits a pulse at regular intervals. By detecting these events in the analog traces, we can estimate the sampling frequency of the ADC as seen by the HICANN. In our tests, the estimation of the sam-



(a) Wafer 24 (old readout system) **(b)** Wafer 23 (new clock distribution board)

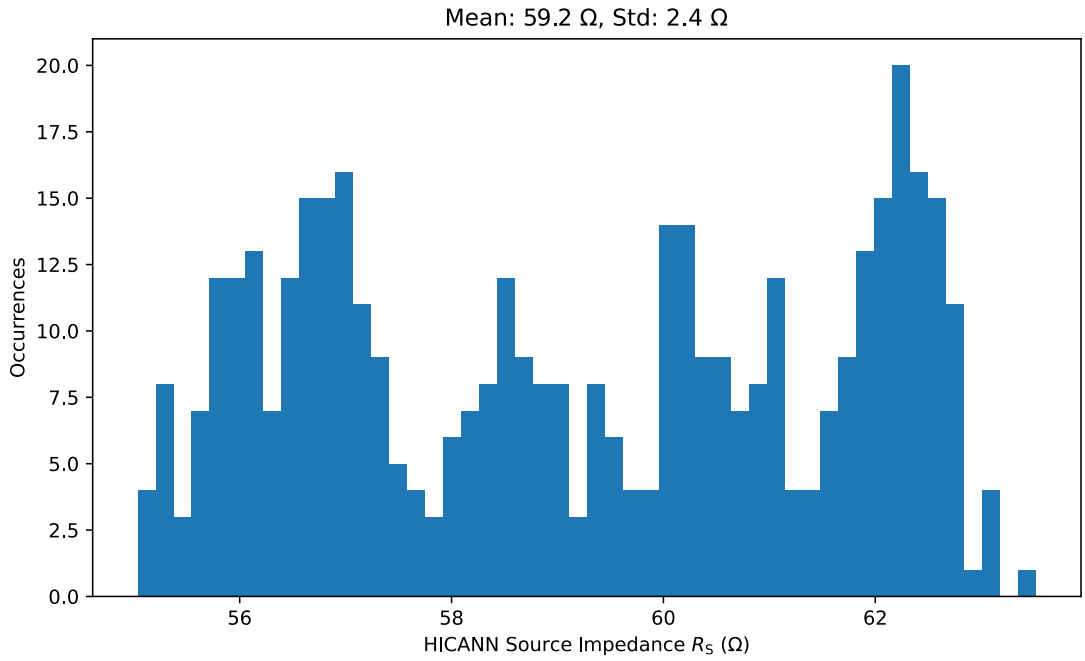
Figure 4.1.: The clock drift between analog membrane trace and digital spike data. Both traces show the last 30 ms of a 5000 s recording (biological time domain). On wafer 24 the old readout system displays a noticeable shift for the last spikes of the recording. Wafer 23 is using the new clock distribution board and shows a much better alignment between digital and analog data. Some remaining offset can be explained by the different data paths, which is therefore constant over time and can be corrected by a slight relative shift.

pling frequency was within 1 Hz of the nominal 31.25 MHz, by which we conclude that the clocks are aligned. Furthermore, the effects are shown in Figure 4.1 in a membrane trace for an experiment run for 5000 s in the biological parameter space, i. e. equivalent to 0.5 s in the hardware parameter space. The experiment consists in a neuron configured to spike continuously due to its resting potential being configured above the threshold potential at which the neuron spikes. The last 30 milliseconds in the biological parameter space are shown in the plots, for both the previous ADC system and using the ANANAS board readout; the effects of the frequency difference between HICANN and the analog sampling are evident. Such a synchronization enables running long experiments, which are ideally suited for the BSS-1 system, which has a large acceleration factor but also an experiment execution overhead largely dominated by the writing time of the floating gates.

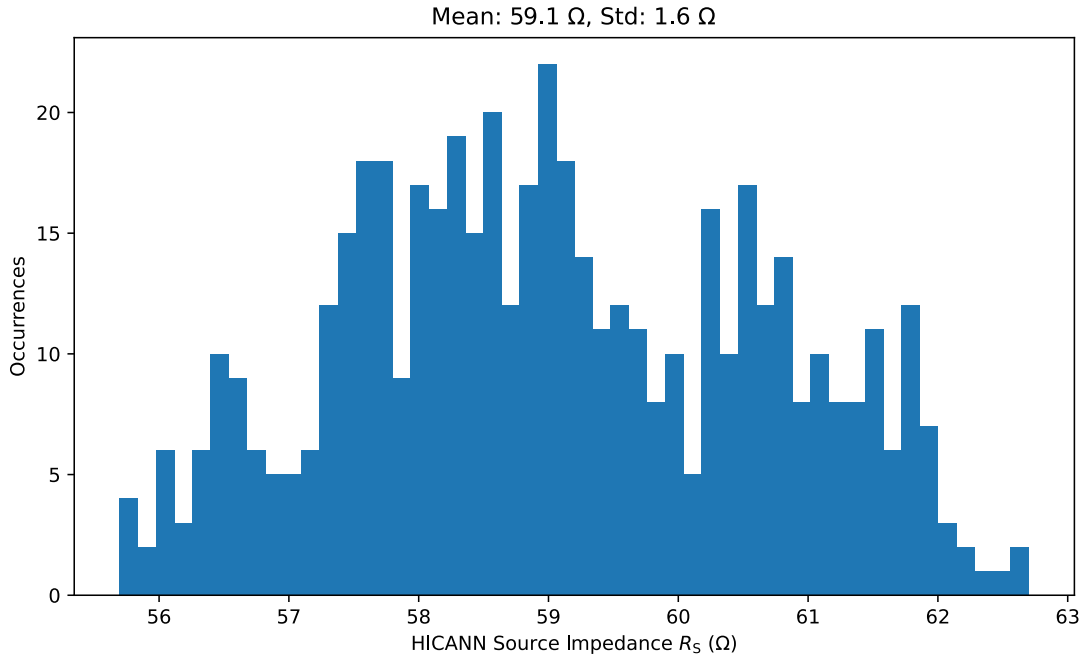
4.2. HICANN Source Impedance

The HICANN source impedances for all available HICANNs of the two used wafer modules have been measured in the framework of this thesis as a prerequisite for the calibration of neurons. The results are shown in Figure 4.2.

Not all of the 384 HICANNs were available, e. g. due to failing communication to the FPGAs or due to being blocklisted because of some hardware defects. All of the properly working HICANNs were measured with both analog output lines, resulting in 400-600 samples making up the histograms. Every measurement has been executed for a DAC



(a) Module 23 shows source impedances of $(59.2 \pm 2.4)\,\Omega$.



(b) Module 39 exhibits source impedances of $(59.1 \pm 1.6)\,\Omega$.

Figure 4.2.: Histogram of the measured HICANN Source Impedances. For each module, as many of the 384 HICANNs as possible have been measured with both analog output lines, the plots shows values obtained for 200-300 HICANNs. Each measurement has been executed for a DAC value of 500 being set for the floating gates.

value of 500 being set for the floating gates. The values for both modules show a mean of around 59 Ω. This is not perfectly in accordance with the reported values in [14], which are $(55.7 \pm 1.0)\,\Omega$, but still comparable. Those measurements were also done on a different wafer module with a previous generation of HICANNs which might explain some deviations. When calculating the correction factor of accounting for the imperfect voltage divider of source impedance and terminating, a value of $R_\mathrm{S} = 59\,\Omega$ yields the following:

$$k_\mathrm{correction} = \frac{R_\mathrm{S} + R_\mathrm{T,ideal}}{R_\mathrm{T,ideal}} = 2.18, \tag{4.1}$$

assuming an ideal termination of $R_\mathrm{T,ideal} = 50\,\Omega$. In [10, Table 4.1], correction factors of 2.197 ± 0.054 are listed. They also stem from measurements with earlier generations of HICANNs.

Furthermore, there might be a dependence of the measured source impedance on the used DAC value for the floating gates (cf. Figure A.2). This circumstance has not been thoroughly investigated in favour of first proceeding with the neuron calibrations, and revisiting the topic later on, if necessary.

4.3. Readout Data Quality

The main advantage of using the new analog readout system and digitising the data with the ANANAS boards is the increase in parallel recording capabilities. Another improvement is the reduction of the physical distance between the wafer and the ADCs digitising the data. The long cables necessary to route the signals from the analog breakout boards to the ADC crate in the old system made the signals susceptible to noise coupling in, despite the use of shielded ribbon cables. In the new system, the ADCs are located directly on the ananas board, close to the wafer, which should lead to a decreased readout noise and thus make the calibration of neurons more robust.

The resolution of the ADCs in both systems is 12 bit, so there is no expected change in the quality of the data in this regard. The old system is operated on a sampling frequency of 96 MHz, which is around three times faster than the sampling frequency of the hsADCs in the new system, which are operated at 31.25 MHz. Additionally, the hsADC is equipped with an internal 3$^\mathrm{rd}$ order anti-aliasing filter with programmable cut-off frequency, set to 14 MHz in this case. The frequency response curves can be found in [6].

In order to evaluate the readout noise, the `sthal_hwtest_wafer_AnaRM.py` test has been executed for different recording times. Its purpose is simply to test the analog readout system for proper functionality. To this end, floating gates are connected to the analog outputs, a measurement with zeroed floating gates is executed, and then another measurement with around half of the maximum settable voltage. In Figure 4.3, exemplary traces with a recording time of 1 ms wall clock time for the two different systems can be seen. The shown traces obtained with the new system (wafer 23 is equipped with

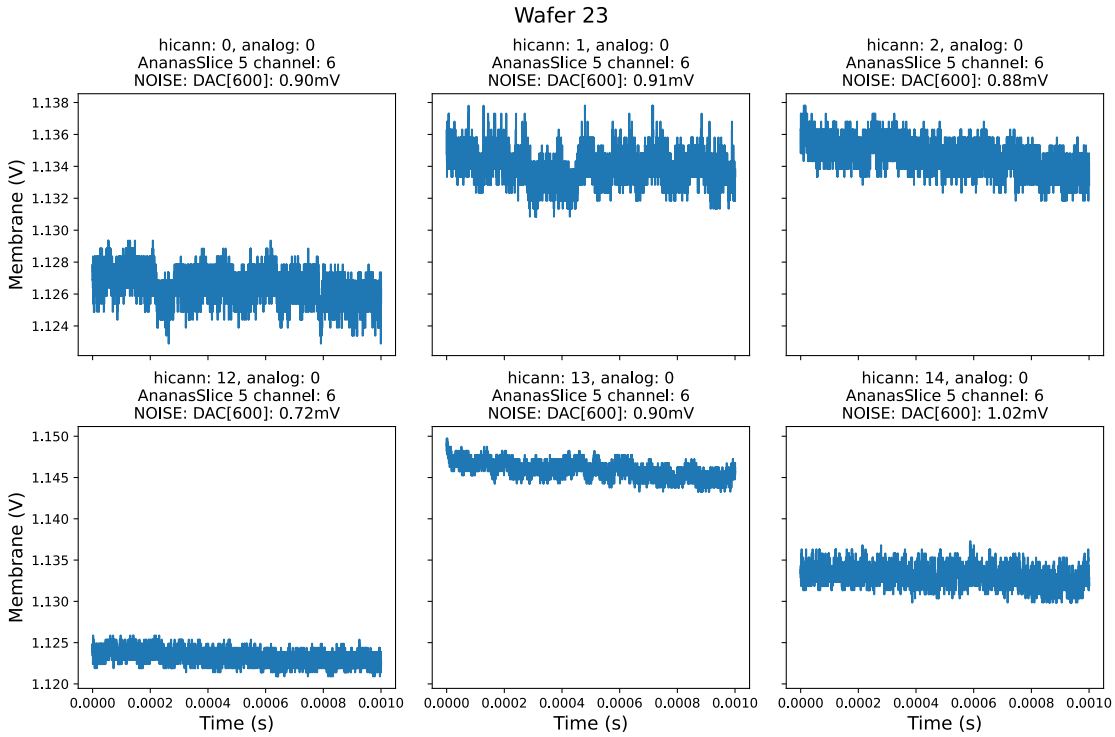
ANANAS boards) display a readout noise of around 1 mV. On the other hand, the data collected with the old system (wafer 24 still operates with the ADC crate) exhibits more visible noise which is on the order of 3 mV to 4 mV. This seems to indicate that the absence of the long ribbon cables indeed improves the readout quality. The test gets executed for each HICANN one after the other, which includes a reprogramming of the floating gates between runs, since the test is first executed for the minimum settable input voltage, before the floating gates are programmed to a testing value. Even when always programming the floating gates to a fixed value, the realisation of the voltage underlies relatively strong variations. This is most likely the cause for the different mean values of the displayed traces.

In order to investigate the readout noise and its correlation to the change of the digitising system systematically and be able to more reliable claim a better signal quality of the new system, the test was executed on all available HICANNs for two different modules of each readout system. The statistics of these results can be found in Figure 4.4. They confirm the supposition that the new readout system has a higher signal quality than the old system. Whilst the average noise for recordings with the ANANAS boards lies on the order of (1.0 ± 0.5) mV, the old system is subject to noise on the order of (3.0 ± 1.0) mV.[1] This comparison has to be treated with caution because the bandwidths of the two ADCs are not the same. On the one hand the old system has the higher sampling frequency, which naturally leads to the collection more noise from higher frequencies, which are too fast for the slower ADC to capture. On the other hand, the mentioned anti-aliasing filter further reduces the bandwidth of the hsADC, making it even more resilient against high frequency noise. The main reason for this is assumed to be the absence of the long ribbon cables routing the analog signals from the AnaB boards to the ADC crate. Another factor could be the anti-aliasing filter featured on the hsADC. Using a Fourier transform, it would be possible to investigate the different frequency components in the noise, which could give further insight into where the noise in the old system might have come from and what exactly led to the optimisation.

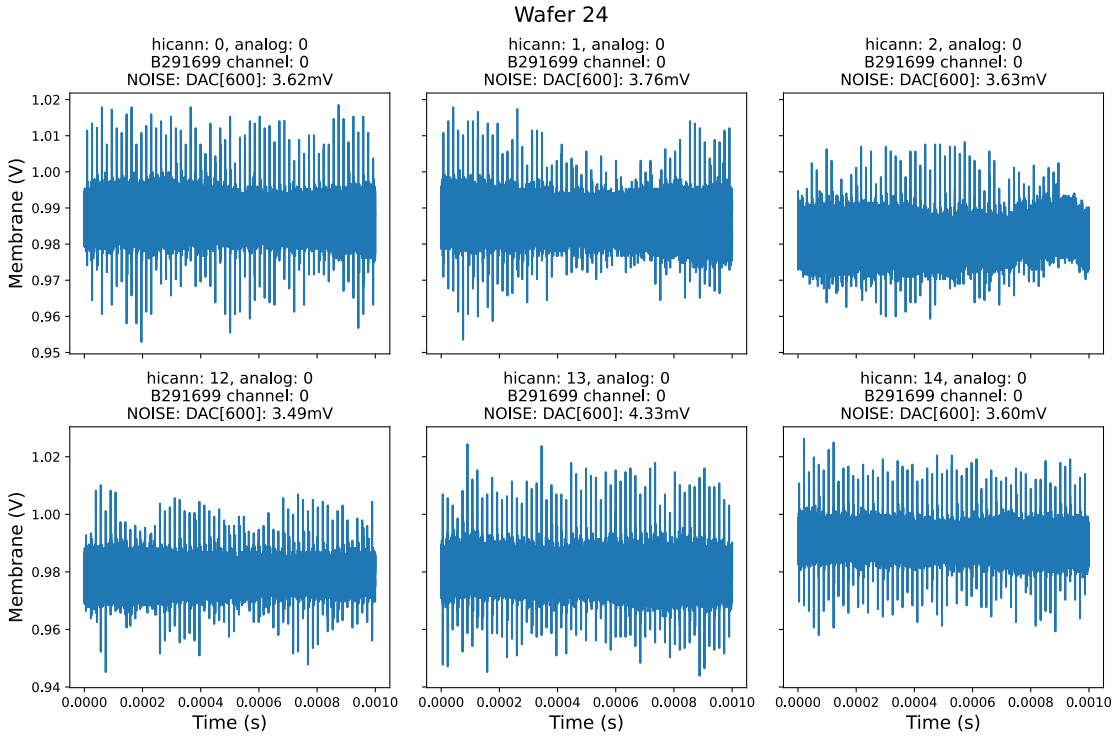
For both analog readout systems, the same dynamic signal range is expected since only the manner of recording has changed, but not the data itself. The exact range depends on the specific neuron parameters used – in [14, Figure 3.47] a trace with a signal range of about 0.5 V can be seen for example. The Signal-to-noise ratio (SNR) is commonly used to compare the levels of noise and signal. It can be calculated as follows:

$$\mathrm{SNR}_{\mathrm{dB}} = 20 \log_{10} \left(\frac{V_{\mathrm{signal}}}{V_{\mathrm{noise}}} \right). \tag{4.2}$$

[1] Averaging over the data in the plots yields (0.9 ± 0.4) mV for ANANAS and (3.3 ± 1.1) mV for the old system.



(a) Traces for wafer 23 display a readout noise of around 1 mV.



(b) Traces for wafer 24 exhibit a readout noise on the order of 3 mV to 4 mV.

Figure 4.3.: Exemplary membrane traces recorded with the old and new readout system. Wafer 23 is equipped with the new ANANAS boards, whereas wafer 24 still uses the old ADC crate. The visible differences in the mean value of the traces is attributed to floating gate variations.

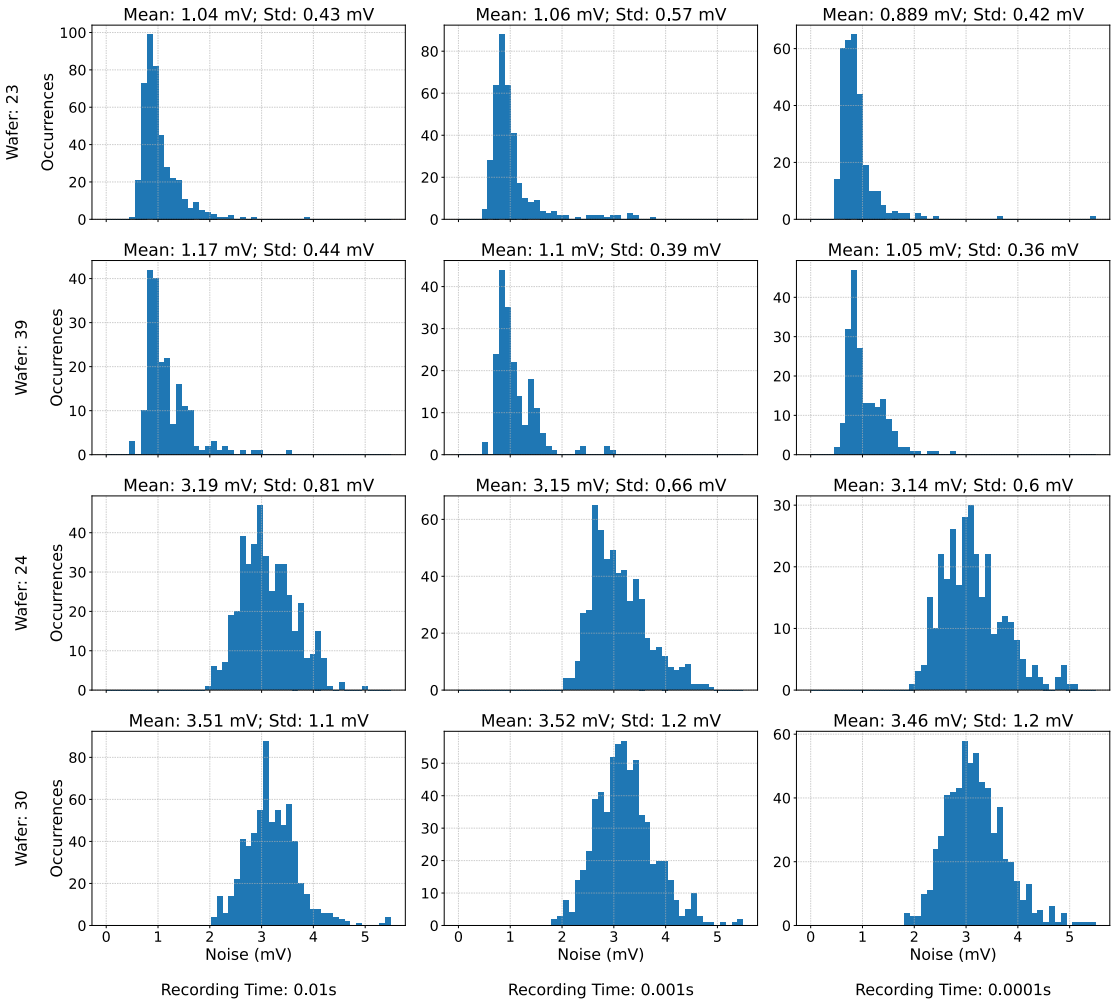


Figure 4.4.: The readout noise obtained with the two different systems. The upper two rows have been recorded on modules 23 and 39, both of which have ANANAS boards mounted. The lower two rows show the readout noise for wafers 24 and 30, each of which is digitising its analog data using the ADC crate. Three different recording times have been measured for each module, ranging from 10 ms to 0.1 ms wall clock time. The total number of data points varies due to hardware constraints of the number of available HICANNs on each module. It is apparent that the new readout system has a higher signal quality. The average noise of recordings lies on the order of (1.0 ± 0.5) mV, whereas the old system is subject to noise on the order of (3.0 ± 1.0) mV. Some outliers (for the old readout system) at higher noise levels have been cut off for visualisation purposes as they are assumed not to be relevant and have not been further investigated. For the sake of completeness, the full plot is included in the appendix (Figure A.1).

For a signal range of 0.5 V, the following SNRs can be estimated:

$$\text{for the ANANAS boards: } \mathrm{SNR_{dB}} = 53.98\,\mathrm{dB}, \tag{4.3}$$

$$\text{and for the old system: } \mathrm{SNR_{dB}} = 44.44\,\mathrm{dB}. \tag{4.4}$$

4.4. Neuron Calibration

The neuron calibration aims to provide the relationship between the values which can be set to control the hardware and the effective values that these configurations lead to. Using this calibration, it is possible to specify a given neuron parameter like the resting potential E_{rest} (cf. Section 2.1.1) in the biological or hardware domain in mV, and accurately know which value has to be set on the hardware to achieve the desired result.

The hardware is controlled by floating gates which receive a DAC value as input given as a 10 bit number (i. e. in the range of 0 to 1023). These DAC values are converted into analog voltages U or currents I. The theoretical equations governing these transformations are simply a linear scaling relative to the full range:

$$U = \frac{DAC}{1023} \cdot 1.8\,\mathrm{V}, \quad \text{and} \tag{4.5}$$

$$I = \frac{DAC}{1023} \cdot 2.5\,\mu\mathrm{A}. \tag{4.6}$$

This is the principle conversion, also referred to as *no calibration* or *without calibration*, since it is not based on any specific measurements. In order to gauge the quality of the calibration later on, these equations are the basis for the reference results. The floating gates are not perfectly deterministic, so that two different write cycles with the same input DAC value can lead to slightly different analog results. This is referred to as *trial-to-trial variation* and can be mitigated to a certain degree by simply repeating an experiment a certain number of times and averaging over the results.

The calibration of two specific neuron parameters is outlined in the following, exemplifying the process and showing the results.

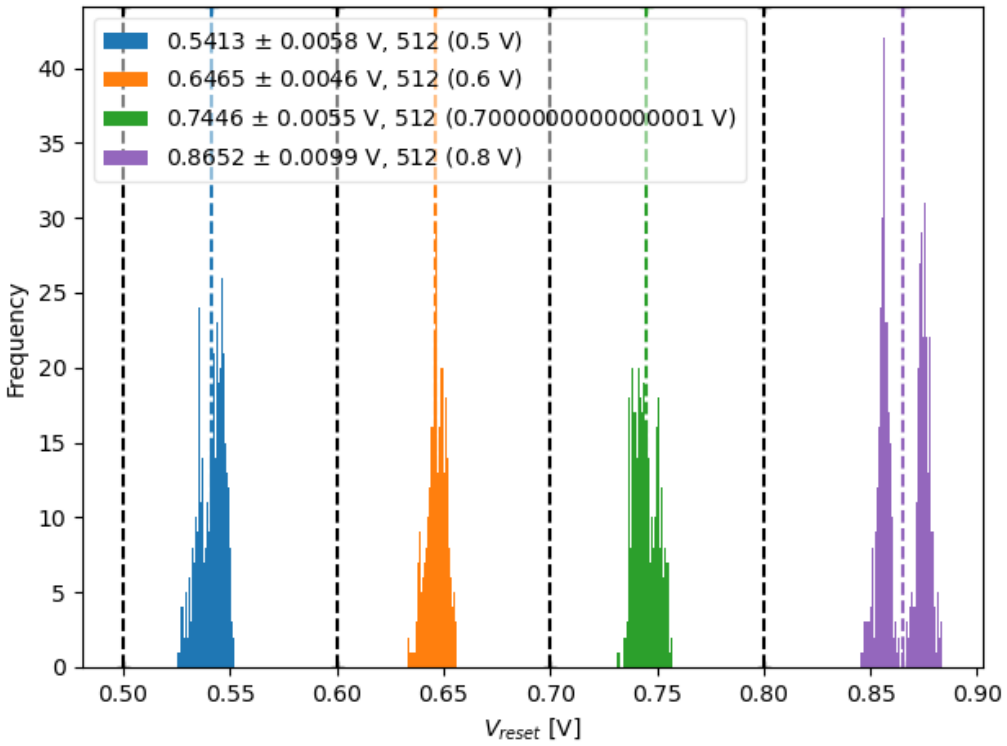
4.4.1. Reset Potential V_{reset}

After a neuron elicits a spike, the LIF model shorts the membrane potential to the reset potential V_{reset}. For the duration of the refractory period τ_{refrac} this clamping stays active before releasing the membrane potential again to evolve dynamically. In order to calibrate the reset potential, the resting potential E_{rest} is set to be above the threshold potential V_{thres}, causing the neuron to periodically spike. Additionally, the refractory period is set to a long time. The reset potential is then obtained by averaging over the membrane potential during the refractory period. The results of the calibration of the reset potential can be seen in Figure 4.5.

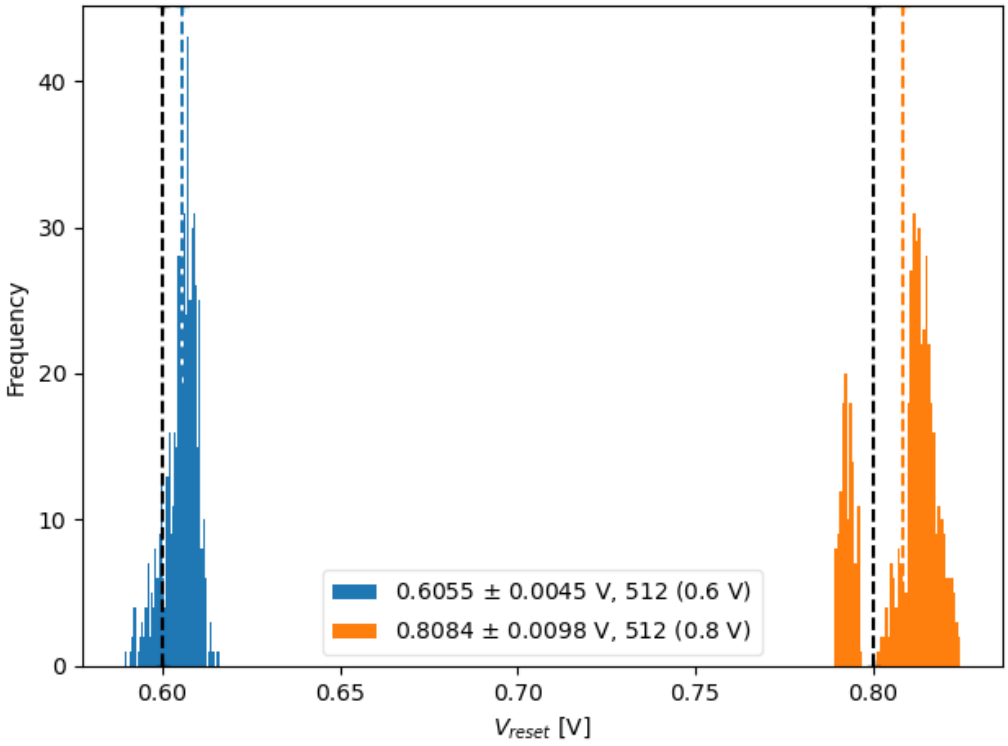
The reset potential parameter is special in the way that it is a shared parameter for a whole block of 128 neurons. The goal of the calibration of V_{reset} is therefore to minimise the block-to-block variation. The variation between the neurons in one block is addressed by a different calibration step, namely the *readout shift*.

4.4.2. Refractory Period τ_{refrac}

Since the refractory period τ_{refrac} is the duration of the clamping of the membrane potential to the reset potential V_{reset} after it elicits a spike, it is similar to the procedure in Section 4.4.1. The refractory period gets controlled by the hardware parameter I_{pl}, defining the pulse length for closing the switch shorting the membrane and reset potentials, and is inversely proportional to it, i.e. $\tau_{\text{refrac}} \propto \frac{1}{I_{\text{pl}}}$. The neuron is again configured to be in a leak-over-threshold state, i. e. periodically firing. By executing an initial measurement with the maximum settable I_{pl}, a near vanishing refractory period is achieved and the inter-spike interval (ISI) is measured. This is taken as a reference for how long it takes the leaky neuron to reach the threshold after the refractory period is over. Then, subsequent measurements with decreasing I_{pl} and therefore increasing τ_{refrac} are executed. The ISI is again calculated for each measurement, and after subtracting the reference value, the duration of the refractory time is obtained. The results are displayed in Figure 4.6.

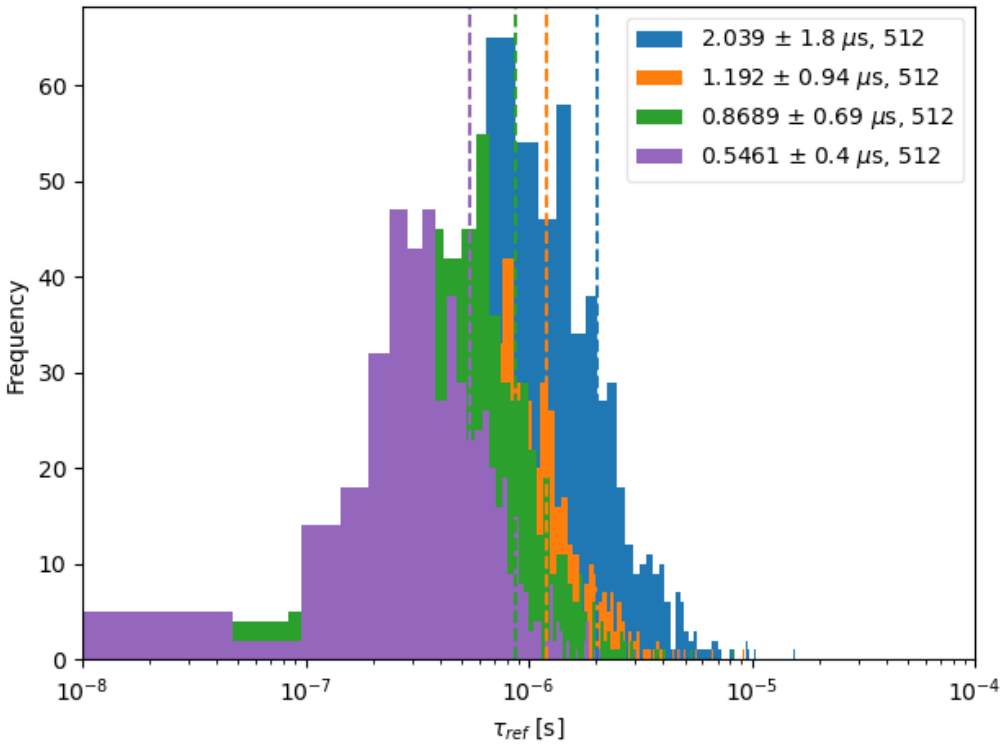


(a) uncalibrated

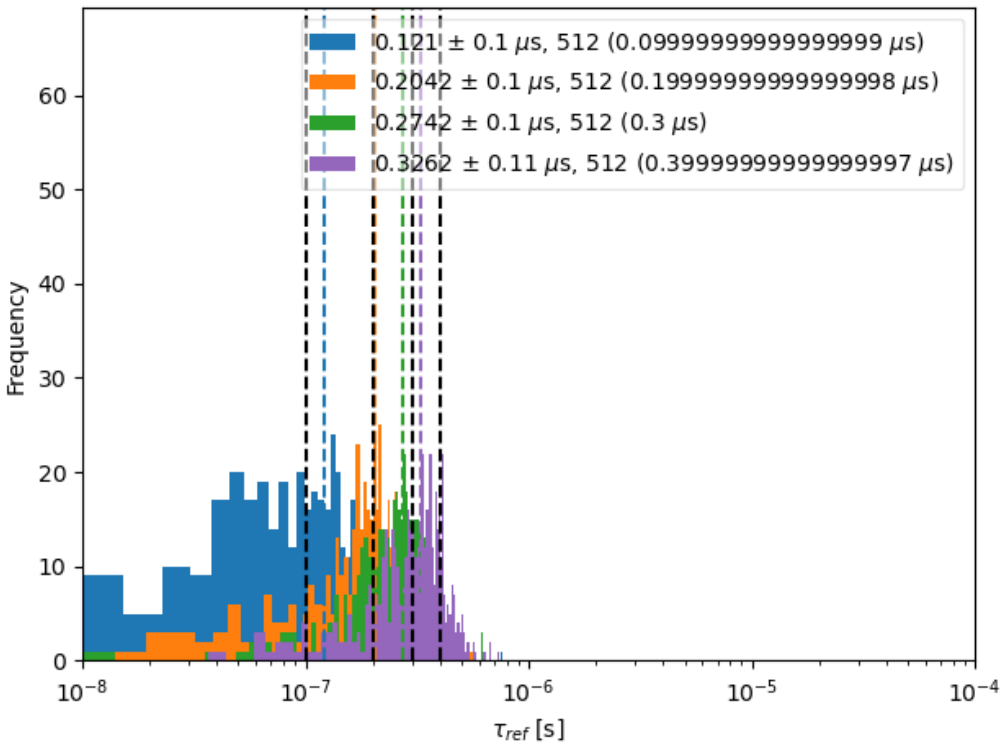


(b) calibrated

Figure 4.5.: Calibration results for the reset potential.



(a) uncalibrated



(b) calibrated

Figure 4.6.: Calibration results for the refractory time.

Chapter 5.

Discussion and Outlook

SO FAR...

...the ANANAS board has been integrated into the wafer software stack and made available for neuron calibrations on the BSS-1 system. Technically, the ANANAS boards could have already been used to record membrane traces without the neuron calibrations because the only requirements are the calibration of the ANANAS board itself, as well as the HICANN source impedance measurement (even this could have been approximated). But without the neuron calibrations this would have been of very little use as it is quite difficult to draw conclusions from a system which configuration and behaviour to a change of the configuration is unknown. That is why the neuron calibration is the last, but still an integral step of the road to running experiments on the wafer module. Now it is possible to calibrate neurons and thereby enable meaningful experiments requiring analog data.

The biggest advantage of the ANANAS system over the old digitiser crate is the number of parallel readout channels which is bigger by a factor of 28 (cf. Section 2.2.3). This can already be harnessed for full-wafer experiments – its demonstration was out of the scope of this thesis, though. Unfortunately, time did not permit to make these parallel capacities utilisable for the neuron calibrations yet.

The readout noise of the two analog systems was measured and compared. A decrease of a factor of ca. three was found, with the ANANAS board displaying a readout noise of around 1 mV, whereas the old analog readout system is exhibiting noise around 3 mV to 4 mV. Since each individual wafer displays certain variations, directly comparing two different modules only has a limited significance. Thus, two modules of each system were compared further indicating that the observed effect is of a systematic nature. Unfortunately, there was not enough time to start and measure additional modules. However, this comparison still needs to be treated with caution. The different ADCs have differing bandwidths and therefore different susceptibilities to noise. A fairer comparison would be to extract the frequency response curve of the anti-aliasing filter from the datasheet of the hsADC mounted on the ANANAS board and also use it on the signal the old adc measures. After performing a Fourier transform on the signal, the filter curve could be applied and the result transformed back to the time domain, and then the noise of the trace be calculated. On the other hand, a decrease of the noise is expected, since a large part to the noise in the old system was attributed to the long cables necessary to route the analog signals from the analog breakout boards to the adc crate. These cables are now no longer necessary because the ANANAS board and thereby the ADCs are now directly attached to the main PCB.

Another improvement is the usage of a small clock distribution PCB, making the clock signal driving the FCPs which in turn drive the HICANNs, also available to the ANANAS boards via the sata connector. This prevents the clock signals to slowly drift apart over long recording times.

Currently, the runtime for a full HICANN calibration with an ANANAS board lies on the order of roughly 10 hours. With the old analog readout system it took around 2 hours. Therefore, the current hypothesis for its cause would be a suboptimal implementation of the integration of the ANANAS board. This is not unexpected, since the primary goal was to successfully complete the integration. Unfortunately, time did not permit to proceed further and debug this long runtime, which could be caused for example by unnecessarily disconnecting from the hardware and reconnecting to it.

The observed apparent dependence of the measured source impedance on the used DAC value is not fully understood yet. There are quite a few different possibilities for what might play a role in this, which is why it was decided not to fully investigate this when the current accuracy is already sufficient. One of the factors playing a role might be the fact that the output drivers are not at their optimal operating point for small or high DAC values, leading to a potentially high gain error. Another reason might be the CM voltage of the hsADC coupling in and affecting the measurement. If there are any indications that this dependency of the source impedance might have a negative effect on the accuracy of the system, the topic can be revisited and re-evaluated, but so far it does not seem that way. It is also unlikely to change, because the membrane dynamics evolve almost exclusively in the range where the source impedance is at a stable value anyway.

STILL...

...there is room for improvement and further development. Most notably there is no support for concurrently useable channels of the ANANAS board for neuron calibrations yet. This is due to the granularity of the licenses allowing access to the hardware. Currently, the lowest accessible unit would be a whole ANANAS slice consisting of eight channels which are spread out across four reticles. However, for experiments spreading across multiple reticles, a global wafer synchronisation is necessary which requires the use of one specific master FPGA in its current implementation. Thus, the full wafer-scale parallel calibration mandates a non trivial restructuring of cake, which is beyond the scope of a bachelor's thesis. However, the usage of the new recording system has been showcased and validated to calibrate neurons, and already allows the parallel collection of data. Making full use of all channels would significantly speed up the calibration of a whole wafer. The trial-to-trial variations of the floating gate cells lead to the possibility of executing a higher number of repetitions and achieving a more accurate calibration at the cost of increased runtime. Thus, having more channels available could also enable higher precision calibra-

tions by moving the break-even point between needed runtime for a calibration and its quality.

For experiments, the ANANAS boards allow the use of 84 parallel analog readout channels per wafer already. This is due to the fact that the one master FPGA is suited to synchronise the whole wafer for one large experiment. Therefore, experiments relying heavily on analog traces, like *surrogate gradient* experiments, can already benefit significantly from the integration of the ANANAS board. For a specific experiment, it is also possible to fine-tune the neuron calibrations to improve the results. This could for example be done by running a calibration for the specific parameter ranges of the experiment and enhance the accuracy in those parts, while possibly sacrificing a broad validity of the calibration across the whole parameter range.

Also to be done more extensively are speed tests to compare single channel collections between the systems, in order to track down why exactly the current implementation using the ANANAS board is so much slower than using the old system.

Appendix A.

Appendix

Repository	Changeset	git commit hash
anacoma	23685	d8d27e558d6e5b6e4bf550f149ccc0818ed69cce
	23714	529a2ac169efec506af69a2240c45b907edfe326
ananas-calibration	22769	af4e9597e244b644397abe9145cd2f5999fb972f
	22771	07e7df0c5d10d7b09033e5e938052863f3a04aa4
	23723	761e5aa1e01418c61e1f4b381706d0a8c74f37ad
	23810	d108e9d8a9c1884cd876907808d17c0d9fc88231
	23974	68e538b7411683e2c29ef539fab0ac670b7f6a20
cake	14373	084ad4db89bde94d07a584174c5b8e6bce5a6b38
	23776	1f7de21c340dc6dc31f06194bc51ba698dfad850
	23971	dac9b0ea2f5a3323c86f4551386a6b13497e0f7a
	23972	926d800141f2428c508688bcfb6ea6489861627a
	24013	89b1a97edfff8183d8de2cdb6e4d926630ded91d
halbe	23916	ecce8c35901327ea1b8449d25b616fc99bf9a9c1
marocco	14982	048d54a4fd83518278fcf6664690de4175e28aaa
sthal	23620	9839a34a1022f5e2c1cd09faa8407a8ac9992522
	23623	15280ec5f39c5df5926f63be6cf5166afd59bf12
	23729	1cddd351a277ed048c9aa9e8c1e4a7a9414ece91
	23918	b47c86f4cc3814df712060e3b0617a056d06a23a
	23928	4c116e259b0c09c54a47d116890d3c0c92cbf48a
	23991	aa2c022f886d2d538307fdac335067b19f6e681a
	23994	7917f51e9b4eabb3d74743bcab5cbb363dab8930

Table A.1.: Software states used for this thesis.

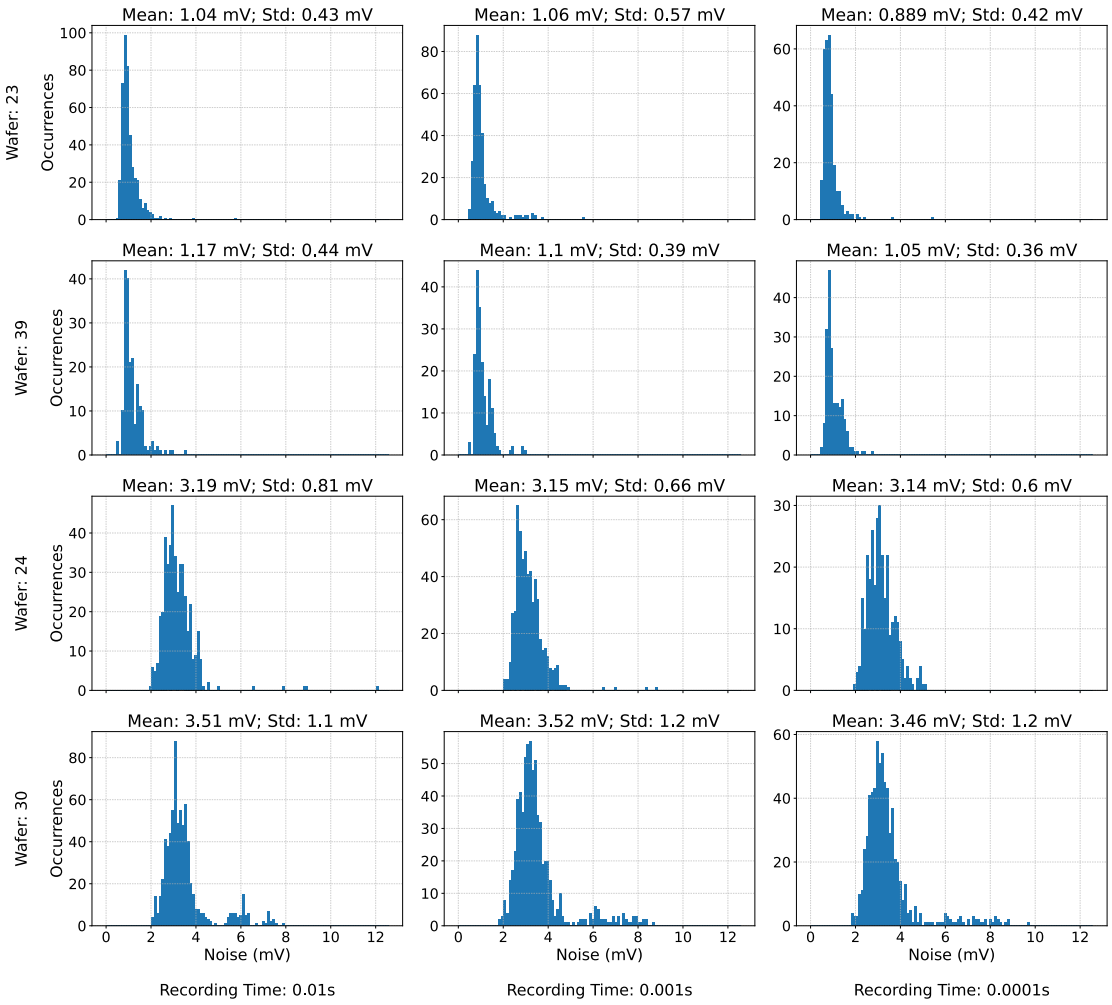


Figure A.1.: The full readout noise obtained for the two readout systems. This plot shows all data – including some outliers for the old readout system, which were omitted for visualisation purposes – from Figure 4.4. The upper two rows have been recorded on modules 23 and 39, both of which have ANANAS boards mounted. The lower two rows show the readout noise for wafers 24 and 30, each of which is digitising its analog data using the ADC crate. Three different recording times have been measured for each module, ranging from 10 ms to 0.1 ms wall clock time. The total number of data points varies due to hardware constraints of the number of available HICANNs on each module.

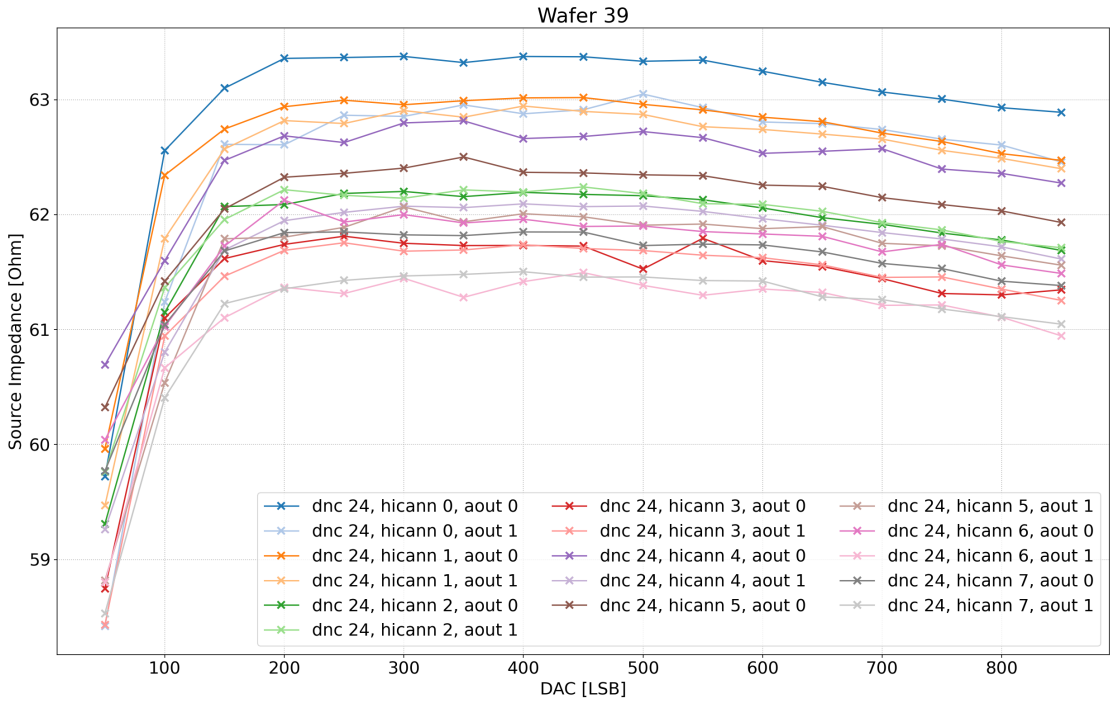


Figure A.2.: Source Impedance values versus DAC.

Glossary

ADC Analog-to-digital converter. 7–11, 21, 24–27, 32, 36

AdEx Adaptive Exponential Integrate-and-Fire. 5

ANANAS board ANAlog Network Attached Sampling Board. iv, 1, 2, 5–13, 15–18, 21, 22, 24–27, 32–34, 36, V

BSS-1 BrainScaleS-1. iv, 1, 3, 5–14, 17, 22, 32, V

cake Calibration framework for the BrainScaleS-1 system. 16, 18, 33

calibtic Framework for generating, storing, retrieving, and applying calibration files. 16, 18

CM Common mode voltage. 10, 13, 14, 16, 33

COBA Conductance-based. 4, 5

CUBA Current-based. 4, 5, V

DAC Digital-to-analog converter. 28, 33

FCP FPGA Communication PCB. 6, 7, 21, 33

FlySpi "Flying Spikey", Spartan-6 FPGA board. 7, 8, 12, 21

FPGA Field-programmable gate array. 6, 7, 9, 21, 22, 33, 34

HALbe Hardware abstraction layer back end. 16, 18

HICANN High Input Count Analog Neural Network chip. iii, iv, 1, 2, 6, 7, 10, 11, 15, 18, 21–25, 27, 32, 33, 36

hsADC High-speed analog-to-digital converter. 9, 10, 12–14, 24, 25, 32, 33

I/O Input/output. 6, 7

ISI inter-spike interval. 29

LIF Leaky Integrate-and-Fire. iv, 3–5, 28, V

lsADC Low-speed analog-to-digital converter. 8, 9, 12, 13, 15

marocco Mapping and routing software for the BrainScaleS-1 system. 16, 18

PCB Printed Circuit Board. 6, 7, 9, 21, 32, 33

SNR Signal-to-noise ratio. 25, 28

StHAL Stateful hardware abstraction layer. 16, 18

USB Universal Serial Bus. 21

WIO Wafer input/output board. 21

References

[1] Daniel Barley. "Development of a Communication Framework for the ANANAS System." Bachelor's thesis. Universität Heidelberg, 2019.

[2] Andrew P. Davison et al. "PyNN: a common interface for neuronal network simulators." In: *Front. Neuroinform.* 2.11 (2009).

[3] A. L. Hodgkin and A. F. Huxley. "A quantitative description of membrane current and its application to conduction and excitation in nerve." In: *The Journal of Physiology* 117.4 (1952), pp. 500–544. eprint: https://physoc.onlinelibrary.wiley.com/doi/pdf/10.1113/jphysiol.1952.sp004764.

[4] Joscha Ilmberger. "Development of a digitizer for the BrainScaleS neuromorphic hardware system." Master's thesis. Universität Heidelberg, 2017.

[5] Texas Instruments. *12-bits, 125/105/80/65 Msps ADC with DDR LVDS/CMOS Outputs*. ADS6125. Rev. A. Mar. 2008.

[6] Texas Instruments. *16 Channel Variable Gain Amplifier (VGA) with Octal High Speed ADC*. AFE5851. Rev. B. May 2010.

[7] Texas Instruments. *64-Channel, Current-Input Analog-to-Digital Converter*. DDC264. Rev. D. Dec. 2016.

[8] Keithley. *Series 2600B System SourceMeter Instrument*. 2600BS-901-01. Rev. A. Sept. 2012.

[9] C. A. Mead. "Neuromorphic Electronic Systems." In: *Proceedings of the IEEE* 78 (1990), pp. 1629–1636.

[10] Sebastian Millner. "Development of a Multi-Compartment Neuron Model Emulation." PhD thesis. Ruprecht-Karls-Universität Heidelberg, 2012.

[11] Eric Christian Müller. "Novel Operation Modes of Accelerated Neuromorphic Hardware." HD-KIP 14-98. PhD thesis. Ruprecht-Karls-Universität Heidelberg, 2014.

[12] Mihai Alexandru Petrovici. *Form Versus Function. Theory and Models for Neuronal Substrates*. Springer Theses. Springer Cham, 2016, pp. XXVI, 374.

[13] Carl Clemens von Randow. "Optimisation of the Automated Calibration Routine of the ANANAS Board." Internship report. Ruprecht-Karls-Universität Heidelberg, 2024.

[14] Simon Rosenkranz. "Calibration of the Analog HICANN Readout Process." Bachelor's thesis. Universität Heidelberg, 2018.

[15] Jakob Sawatzki. "Validation and Automation of the Calibration Routine of the ANANAS board." Bachelor's thesis. Universität Heidelberg, 2022.

[16] Johannes Schemmel, Johannes Fieres, and Karlheinz Meier. “Wafer-scale integration of analog neural networks.” In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. 2008, pp. 431–438.

[17] Hartmut Schmidt. “Large-Scale Experiments on Wafer-Scale Neuromorphic Hardware.” PhD thesis. Ruprecht-Karls-Universität Heidelberg, 2024.

List of Figures

Acknowledgements

First and foremost, I would like to thank Prof. Dr. Johannes Schemmel for the opportunity to deliver my bachelor's thesis in the Electronic Vision(s) Group.
No less I would like to thank José Montes for his excellent supervision, his prompt and thorough help with any questions I posed and problems I encountered, as well as his patience and continuing support throughout my whole time in the group.
Additionally, I would like to express my gratitude to Hartmut Schmidt for his co-supervision (including some tips for my `.bashrc`), and his daughter for her impeccable timing ensuring his aid until my very deadline.
Further thanks go to everyone else in the Electronic Visions Group, especially to Andreas Grübl, Eric Müller, Joscha Ilmberger, Philipp Spilger, Simon Rosenkranz, Jakob Kaiser, Yannik Stradmann, and Florian Fischer.

Living by the motto *better late than never*, I have now completed my Bachelor's degree and want to thank the *Nobelpreisträger 2.0* for setting out on this journey with me: Mathieu Kaltschmidt, Benjamin Bentner, Alexander Müller, Berkay Günes, and Oliver Isak, as well as Linus Balles and Quirinus Schwarzenböck.
Furthermore, I very much appreciate all my other friends, with all their help and assistance, most notably Tobias Jakoby.

Words cannot express my gratitude to my Jazzyboo, without whom I would not be where I am today.

Lastly, and most importantly, I would like to thank my parents and my siblings for their unwavering, unconditional, never-ending support in all regards throughout my whole time at university (and before).

The work carried out in this Bachelor's Thesis used systems, which received funding from the EC Horizon 2020 Framework Programme under grant agreements 785907 (HBP SGA2) and 945539 (HBP SGA3), the EC Horizon Europe Framework Programme under grant agreement 101147319 (EBRAINS 2.0), the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2181/1-390900948 (the Heidelberg STRUCTURES Excellence Cluster), and the Helmholtz Association Initiative and Networking Fund [Advanced Computing Architectures (ACA)] 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, 30 January, 2025