

Fakultät für Physik und Astronomie
Universität Heidelberg

Bachelorarbeit in Physik
eingereicht von

Maximilian Denne

geboren in Heidelberg (Deutschland)

Februar 2014

Testen der Software und Vermessen des Multi-Compartment Chips

Diese Bachelorarbeit wurde von

Maximilian Denne

am

Kirchoff-Institut für Physik an der Universität

geschrieben unter der Aufsicht von

Prof. Dr. Karlheinz Meier

Motivation

In dieser Arbeit geht es um das Testen und Vermessen des Multi-Compartment Chips (entwickelt und designed von Sebastian Millner und Andreas Hartel). Er wurde entwickelt und produziert, um die von S. Millner, während seiner Dissertation, entwickelte

Multi-Compartment Emulation zu überprüfen und zu bestätigen. Da an diesem Chip seit seiner Entwicklung noch keine quantitativen Messungen gemacht wurden und die Software zur Ansteuerung der Hardware neu angesetzt wurde, musste zuerst sichergestellt werden, dass die, von A. Hartel geschriebene und angepasste Software, zuverlässig funktioniert. Aufbauend darauf, sollten Experimente folgen, welche die neu implementierten Schaltungen charakterisieren und verdeutlichen sollen. Man muss ganz klar darauf hinweisen, dass es bei dieser Arbeit sehr viel um das Testen ging, was für die eventuelle spätere Verwendung, unter Umständen im Wafer Scale System, von Nöten ist. Denn dies sollte über kurz oder lang das Ziel sein, wofür diese Arbeit ein erster Grundstein darstellt.

Abstract

This thesis presents the results of testing the software and measuring of the Multi-Compartment chip (developed and designed by Sebastian Millner and Andreas Hartel).

It was developed and produced, to verify the Multi-Compartment emulation, which was developed by S. Millner in his PhD thesis. Since the completion of the chip, there are no quantitative measurements done. So first, it was necessary to ensure, that the software for activation, written and adapt by A. Hartel, works. Based on experiments should follow, that characterize and explain the new implemented circuits. It must be clear that testing was the main task in this thesis. The aim of the work of these and other should be, to use this circuits, possibly at the Wafer Scale System.

Inhaltsverzeichnis

1	Einleitung	1
2	Theoretische Grundlagen	2
2.1	Neuronenmodelle	2
2.1.1	LiF-Neuronen	2
2.1.2	AdEx-Modell	3
2.2	Hardware	4
2.2.1	MCC	4
2.3	Implementierung	5
2.3.1	Dendrite Routing	6
2.3.2	Grundlegende Schaltungen	8
3	Einzel-Neuron-Experimente	9
3.1	Spike-Test	9
3.2	Externer Stromstimulus	11
3.2.1	Experimente mit Stromstufen	11
3.2.2	Bestimmung der Membranzeitkonstanten	14
3.2.3	Feuerratenuntersuchung	19
4	Multi-Compartment-Experimente	22
4.1	Charakterisierung der Inter-Compartment Konduktanzen	22
5	Zusammenfassung	24
6	Anhang	25
7	Quellen	27

1 Einleitung

Trotz immer weiter fortschreitender Forschung ist das menschliche Gehirn immer noch ein großes Rätsel der Menschheit. Durch Entscheidungsfindung und Lernprozesse, bei sehr niedrigem Energieverbrauch, ist das Gehirn jedem Supercomputer der Welt weit überlegen. Im menschlichen Hirn befinden sich ca. 10^{10} Nervenzellen¹, welche für die Informationsverarbeitung zuständig sind. Durch elektrische Signale gelangt die Information über die Synapsen, von Neuron zu Neuron und somit im Gehirn von A nach B. Synapsen dienen als Bindungsglied zwischen den Neuronen. Um diese Sachverhalten besser zu verstehen und nachvollziehen zu können, soll das Brainscales Projekt Licht ins Dunkel bringen. Mithilfe neuromorpher Hardware sollen Nervennetzwerke emuliert werden. Vorteil der Implementierung verschiedener Neuronenmodelle durch analoge Schaltkreise ist die hohe Geschwindigkeit. Nun werden immer wieder neue Schaltungen und Schaltkreise entwickelt und implementiert und somit die Hardware weiterentwickelt. Die eigentliche Hardware, auf der diese Dinge alle realisiert werden sollen, ist das Wafer-Scale System. Mit seinen 384 HICANN²-Chips können hier bis zu 200 000 Neuronen, mit jeweils 224 synaptischen Verbindungen emuliert werden.³

Um nun in einem Punkt der Biologie näher zukommen, entwickelte Sebastian Millner eine Multi-Compartment Emulation, bei der die Punktneuronen auf dem Chip durch Dendriten, die als Kabelleitungen betrachtet werden können, emuliert werden und so miteinander verbunden werden können. Nachdem im Jahr 2012 Sebastian Millner seine Dissertation abschloss und die Arbeitsgruppe verließ, waren kaum Messungen oder Experimente an dem von S. Millner und A. Hartel entwickelt und designtem Multi-Compartment-Chip vorhanden. Um jedoch einen gewissen Überblick über die Funktionalität der einzelnen Bestandteile des Chips, insbesondere die neu implementierten Schaltkreise zu erhalten, für die dieser "Testchip" entwickelt wurde, musste eine gründliche Vermessung durchgeführt werden. Es war im Vorfeld klar, dass im Rahmen dieser Bachelorarbeit nicht alle Bereiche und Implementierungen bis ins kleinste Detail, beziehungsweise überhaupt, untersucht werden können, aber dennoch sollte hiermit ein Anfang gemacht werden, der die Grundlagen für spätere Arbeiten werden könnte. Da dies quasi nun als "Pioneer-Projekt" dienen sollte, mussten zu Beginn grundlegende Messungen gemacht werden, was auch für das Testen der von A. Hartel geschriebenen und auf den MCC angepassten Software und der FPGA⁴-Programmierung galt.

So soll das Kapitel 2 einen kurzen Überblick über sowohl die grundlegenden simulierten Neuronenmodelle als auch über die allgemein implementierten Schaltungen geben. Danach folgen die Ergebnisse einiger Einzel-Neuron-Messungen, die teilweise zum Testteil meiner Arbeit gehören. Darauf folgen Multi-Compartment-Experimente, die speziell auf den Multi-Compartment-Chip zugeschnitten sind. Meine Programme, die zum Messen verwendet wurden, sind von mir in C++ geschrieben worden und die Auswertung der gesammelten Daten erfolgte mit Python.

In der abschließenden Zusammenfassung und Diskussion werden die Ergebnisse noch einmal kritisch hinterfragt und gesammelt.

¹Herculano-Houzel S (2009). "The human brain in numbers: a linearly scaled-up primate brain" - [5]

²"High Input Count Analog Neural Network" ; KIP - Electronic Vision [6]

³s. Quelle [2]

⁴Field Programmable Gate Array

2 Theoretische Grundlagen

Dieses Kapitel soll Hintergrundwissen zu den folgenden Experimenten vermitteln. Sowohl Grundlagen der emulierten Neuronenmodelle, als auch Soft- und Hardware-Basics werden hier aufgeführt. Natürlich soll hier nicht ins kleinste Detail gegangen werden, aber dennoch alle nötigen Informationen gegeben werden, um die Vorgehensweisen und Messungen zu verdeutlichen und zu verstehen.

2.1 Neuronenmodelle

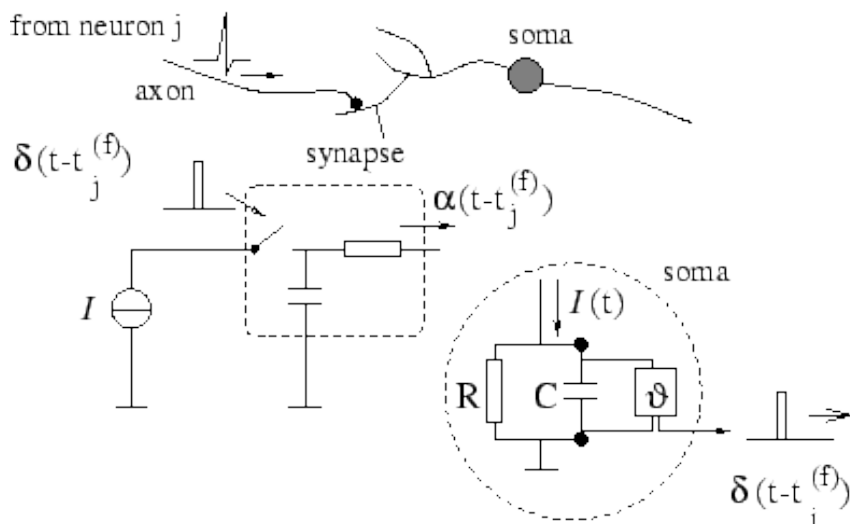
Zuerst muss natürlich erläutert werden, welche Art von Neuronenmodell auf dem MCC emuliert ist und wie deren Besonderheiten aussehen.

2.1.1 LiF-Neuronen

Eines der einfachsten und am weitverbreitetesten Modelle ist das Leaky Integrate-and-Fire Modell (LIF) [2]. Dies besteht aus einem Kondensator (Membrankapazität) und einem parallel geschalteten Widerstand. Dieses RC-Glied wird von einem äußeren Strom getrieben. Der in Abb. 2.1 schematisch gezeigte Schaltkreis kann mit folgender Differentialgleichung beschrieben werden:

$$C_m \frac{dV}{dt} = g_l(E_l - V) + I \quad (2.1)$$

Wobei hier C_m für die Membrankapazität, E_l für die Leakage-Spannung oder Leckspannung und g_l für die Konduktanz stehen. Wenn nun die Membranspannungen über einen bestimmten Wert V_t , auch Threshold-Spannung bezeichnet, steigt, wird sie auf die Reset-Spannung V_{reset} zurückgesetzt.



Quelle:[1]

Abbildung 2.1: Schematischer Schaltkreis eines LIF-Neurons (ϑ entspricht hier V_t)

Nun lässt sich das Neuron, wie gesehen, mittels externen Strominputs stimulieren und so zum Spiken bringen, obwohl die Threshold-Spannung über dem Leakage-Potential liegt. Die Frequenz mit der das Neuron spiked, lässt sich dann wie folgt ableiten:

$$f = \frac{1}{\tau_{ref} + T} \quad (2.2)$$

Wobei hier τ_{ref} die Refraktärzeit⁵ symbolisiert und T die Periode eines Spikes, welche wie folgt definiert ist:

$$T = \tau_m \cdot \ln\left(\frac{U_{reset} - U_{eq}}{U_t - U_{eq}}\right) \quad (2.3)$$

$$wobei : \quad \tau_m = \frac{C_m}{g_l} \quad U_{eq} = E_l + \frac{I}{g_l} \quad (2.4)$$

$$wenn : \quad I \rightarrow \infty \quad \Rightarrow \quad T \rightarrow 0 \quad \Rightarrow \quad f_{Grenz} \rightarrow \frac{1}{\tau_{ref}} \quad (2.5)$$

Genau das zuletzt erwähnte Verhalten soll in einem der experimentellen Teilen gezeigt werden.

2.1.2 AdEx-Modell

Eigentlich ist aber ein erweitertes Neuronenmodell auf dem Multi-Compartment-Chip emuliert. Und zwar das Adaptive Exponential Integrate-and-Fire Modell (AdEx) [3]. Hierbei wurde das LIF-Modell um eine weitere Variable, die Adaption ω , erweitert. Somit wird dieses Modell nun mit 2 Differentialgleichungen beschrieben:

$$C_m \frac{dV}{dt} = g_l(E_l - V) + g_l \Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) - \omega + I \quad (2.6)$$

$$\tau_\omega \frac{d\omega}{dt} = a(V - E_l) - \omega \quad (2.7)$$

$$Wenn \ V = V_t : \quad V \rightarrow V_{reset}, \ \omega \rightarrow \omega + b \quad (2.8)$$

Hier beschreibt Δ_t den Steigungsfaktor, τ_m die Adaptionszeitkonstante und a den Kopplungsfaktor der Adaption. Der Exponentialterm beeinflusst den Spike-Mechanismus, der, wenn das Membranpotential die Threshold-Spannung erreicht, auslöst und die Membranspannung auf das Resetpotential gesetzt wird. Bei diesem Modell kommt zu der eben erwähnte Grenzbedingung noch die einer Zweiten hinzu. Und zwar wird die Adaption nach dem Zurücksetzen auf V_{reset} um einen Faktor b erweitert, was zur Folge hat, dass ω , wenn das Neuron feuert, ansteigt. Dieses Modell ist auf der verwendeten Hardware emuliert. Bei der später verwendeten Hardware ignorierte ich die zusätzlichen Terme jedoch, sodass nur das LIF-Modell simuliert wird.

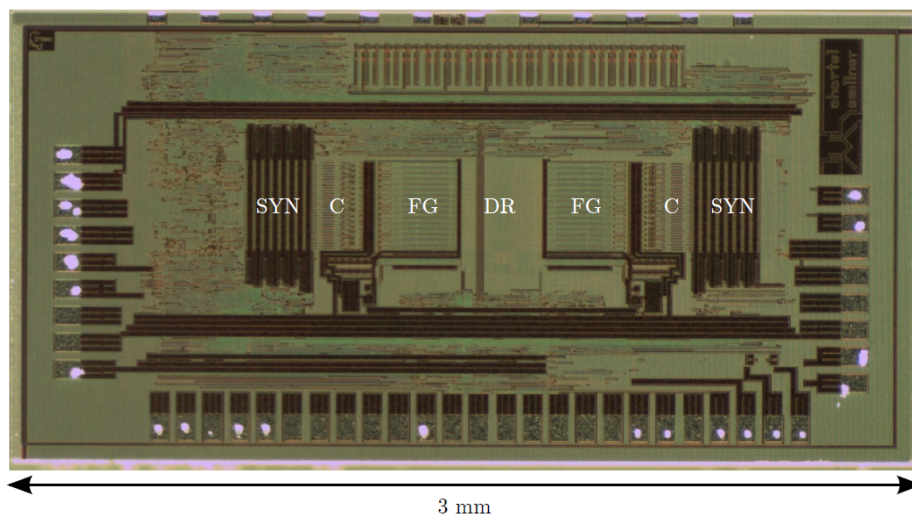
⁵ Als Refraktärzeit bezeichnet man die kurze Zeitspanne, in der an der Membran kein weiteres Aktionspotential erzeugt werden kann (s. [4])

2.2 Hardware

Eine kurze Übersicht über die verwendete Hardware und deren Aufbau soll dieses Kapitel geben. Der MCC ist ähnlich aufgebaut wie der HICANN-Chip, dennoch gibt es Unterschiede, zum Beispiel bei dem Aufbau und der Flächenbelegung. Der Chip ist mittels USB-Verbindung an den PC angeschlossen.

2.2.1 MCC

Auf dem Multi-Compartment-Chip [2], der in Abb. 2.2 dargestellt ist, sind 2 Reihen mit jeweils 32 Compartments implementiert. Jedes dieser Compartments kann synaptischen Input von 16 konfigurierbaren Synapsen erhalten. Der Chip wird von einer 100 MHz-clock betrieben. An den Neuronen sind zudem noch die Floating-Gates implementiert, welche die Neuronparameterspannungen halten. Der Floating-Gate-Block besteht aus 28 Zeilen, aufgrund von 28 verschiedenen Parametern und 33 Spalten. Die erste Spalte eines Blocks ist für die globalen Parameter vorgesehen, welche für alle Neuronen gelten. Die restlichen Spalten sind für jedes Neuron individuell. Im Unterschied zum HICANN Chip, auf dem viel Platz für Synapsen vorgesehen ist, ist der MCC hauptsächlich von den Compartments, Floating Gates und der Dendritenführung belegt.



Quelle:[2]

Abbildung 2.2: Foto eines MCC: Synapsen [SYN], Compartments [C], Floating Gates [FG] und das dendrite routing [DR]

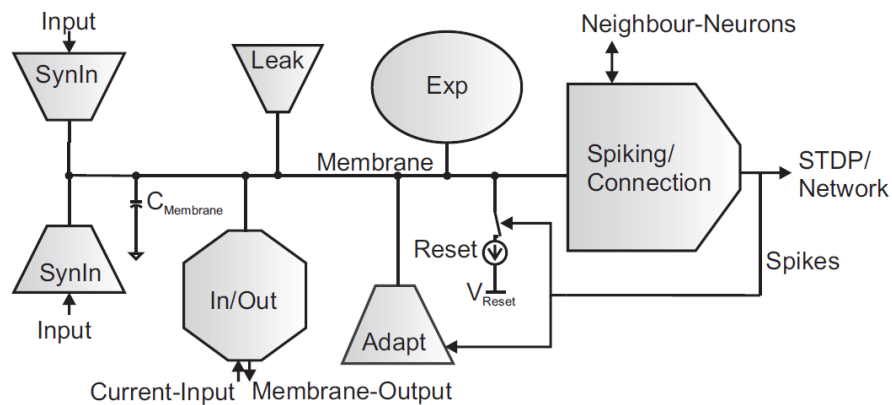
Der zweite Hauptunterschied zum HICANN, außer den Multi-Compartment Eigenschaften, besteht in der Variabilität der Membrankapazitäten. Auf dem HICANN kann nur zwischen 2 Größen unterschieden werden, wohingegen bei dem MCC 64 verschiedene Werte eingestellt werden können. Diese lassen sich über 6 Bits addierend zusammenschalten. Die Werte für die Kapazitäten sind in Tabelle 1 aufgelistet. So ist es möglich die Membrankapazität von 126.7 fF bis zu 4.3 pF einzustellen.

Bit	Wert in [fF]
0	126.7
1	$2 \cdot 126.7$
2	$4 \cdot 126.7$
3	$8 \cdot 126.7$
4	800
5	$2 \cdot 800$

Tabelle 1: Kapazitätsbits

2.3 Implementierung

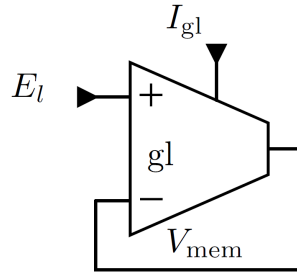
Wie die oben erwähnten Neuronmodelle implementiert sind und realisiert werden, soll in der Abb. 2.3 verdeutlicht und klar gemacht werden. Da ich bei meinen Experimenten nur auf das LIF-Modell baute, schaltete ich alle Terme, die für dieses Modell nicht notwendig sind, mittels Neuronparameter aus. Dies wird realisiert, indem ich die Parameter auf 0 setze, welche diese Terme einstellen.



Quelle:[2]

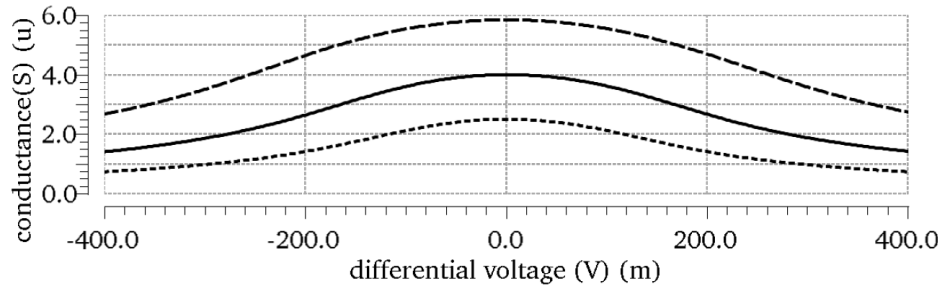
Abbildung 2.3: Vereinfachte schematische Darstellung der AdEx-Neuron Implementation

Somit wurden die Bereiche wie Adaption, exponentiales Verhalten und synaptischer Input nicht berücksichtigt. So sollte quasi nur das LIF-Modell emuliert sein. Inwieweit diese Terme allerdings wirklich keine Rolle spielten, bleibt fraglich, denn die nötigen und angesprochenen Floating Gate Parameter lassen sich nicht komplett auf 0 programmieren. Hierzu aber noch genauere Details in der Diskussion. Nun blieb quasi nur der Leakage-Term übrig, welcher durch einen OTA⁶ realisiert wird. Dieser Schaltkreis ist in Abb. 2.4 abgebildet. Eine Simulation der Konduktanz bei verschiedenen Spannungsdifferenzen zwischen der Membranspannung und V_{dd} ist ebenfalls in Abb. 2.5 abgebildet. Hierauf ist zu erkennen, dass sich die Leakage-Konduktanz, mit dieser emulierten Schaltung, sich im Wertebereich weniger μS befindet und spannungsabhängig ist.



Quelle:[2]

Abbildung 2.4: Leakage-Term Schaltkreis



Quelle:[2]

Abbildung 2.5: Ergebnis einer Simulation des OTA für verschiedene Biasströme (--- : $1 \mu\text{A}$; — : 500 nA ; - - : 250 nA)

2.3.1 Dendrite Routing

Hier jetzt eine kleine Beschreibung, wie die Kommunikation zwischen den Compartments funktioniert, sowohl innerhalb eines Neuronenblocks als auch zwischen den Blöcken. Wie schon kurz in der Einleitung erwähnt, sind die Dendriten als Kabelleitungen modelliert. So sind die einzelnen Compartments über Inter-Compartment Resistances, welche durch OTAs realisiert werden, miteinander verknüpfbar. Eine zweite Möglichkeit besteht darin sie direkt zu verbinden, ohne Zwischenwiderstand.

⁶Operational Transconductance Amplifiers (s. Quelle [2])

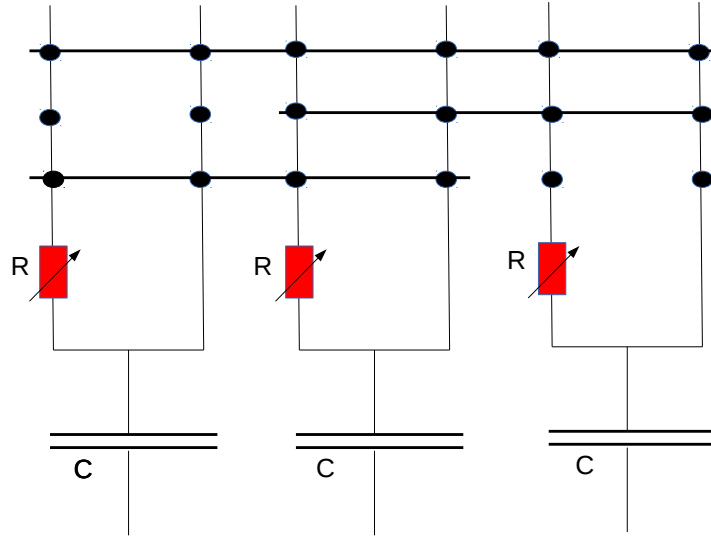


Abbildung 2.6: Compartment Interface mit Ausschnitt des Routing Netzwerks

So können alle Compartments miteinander verbunden werden. Einen Ausschnitt der Routing Matrix ist in Abb. 2.7 zu sehen. Die Switches werden über ein Bitfile gesteuert, wobei die ersten 8 Bits für Verbindungen innerhalb eines Blocks zur Verfügung stehen und das 9. als Verbindung zwischen den beiden Blöcken Top und Bottom. Je nach “Anschalten“ verschiedener Bits kann man nun mit mehreren Neuronen ein Membran-Netzwerk erstellen.

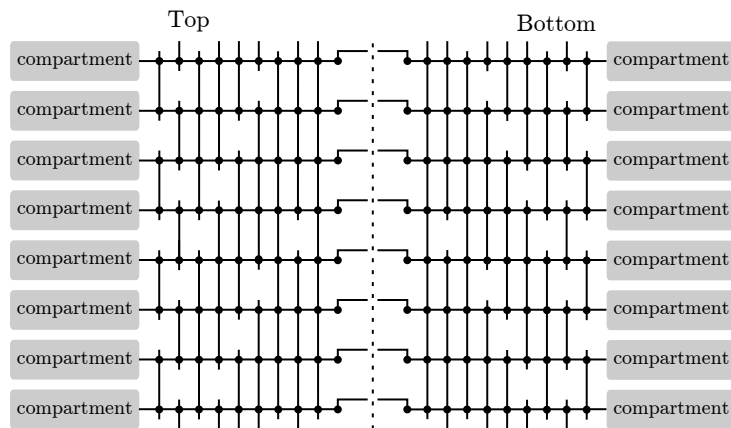


Abbildung 2.7: Routing Matrix des MCC

2.3.2 Grundlegende Schaltungen

Zum Abschluss der Theorie wird noch kurz eine verwendete Schaltung erläutert und zwar der Spannungsteiler. Von dem Konzept dieser Grundschaltung der Elektronik wird in Kapitel 4.1 Gebrauch gemacht.

Anhand der Grafik 2.8 wird ein Beispiel-Spannungsteiler gezeigt, bei dem die Spannungsabfälle an den Widerständen proportional zu den Widerstandswerten ist. Die Gesamtspannung wird in 2 Teilspannungen aufgeteilt. Setzt man nun eine Teilspannung mit der Gesamtspannung ins Verhältnis, dann verhält sich dies wie der Einzelwiderstand zur Summe der Widerstände. Vorausgesetzt die 2 Widerstände werden vom selben Strom durchflossen, gilt:

$$I = \frac{U_2}{R_2} = \frac{U_1}{R_1} = \frac{U}{R_1 + R_2} \quad (2.9)$$

$$\Rightarrow \frac{U_2}{U} = \frac{R_1}{R_1 + R_2} \quad (2.10)$$

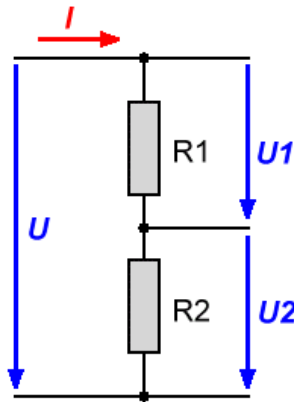


Abbildung 2.8: Spannungsteiler

Dieser Aufbau ist nun beliebig erweiterbar und ermöglicht es, die Widerstandswerte in Abhängigkeit der Spannungen auszudrücken, ohne den fließenden Strom zu kennen. Eben genau diese Gegebenheit wird bei den Multi-Compartment Messungen von Nutzen sein.

3 Einzel-Neuron-Experimente

In dem experimentelle Teil meiner Arbeit ging es zu Beginn darum, die von Andreas Hartel geschriebene Software auf ihre Funktionalität zu überprüfen. Einfache Tests um zum Beispiel das Verhalten der Neuronen bei bestimmten Parametereinstellungen oder Stromstimuli zu untersuchen, waren Voraussetzung für das weitere Arbeiten.

Nach der Fertigstellung des Multi-Compartment-Chips, welcher von Sebastian Millner und Andreas Hartel (2012) entwickelt wurde, musste in erster Linie sichergestellt werden, dass die vorhandene Software problemlos und zuverlässig funktioniert, um den Chip auch richtig einsetzen zu können. Denn nach dem S. Millner nach Beendigung seiner Doktorarbeit die Arbeitsgruppe verließ, wurden bis dahin keine ausführlichen Tests zur Vermessung des Chips durchgeführt. Somit begann ich mit "einfachen" Testläufen des MCC, um mich so systematisch komplexeren und unerforschten Aufgabenstellungen zu widmen.

3.1 Spike-Test

Der erste Test war eine Überprüfung des Spikeverhaltens der implementierten Neuronen. Da alle exponentiellen und adaptiven Terme ausgeschaltet wurden, sollten sich die Neuronen wie LIF-Neuronen verhalten, wie in Kapitel 2.2 beschrieben. Wie schon erwähnt, teilen sich die 64 Neuronen in eine untere Reihe (Neuron-Nr.: 0,2,4,...) und eine obere Reihe (Neuron-Nr.: 1,3,5,...) auf. Um nun das Spikeverhalten zu untersuchen, sweepte ich die Parameter V_t (Threshold-Spannung) und E_l (Leakage-Potential) in 100er Schritten von 0 - 1023 DAC⁷-Wert durch. Sobald die Threshold unter dem Leakage-Potential liegt, sollten die Neuronen anfangen zu spiken. Dies wurde exemplarisch an einem Neuron aus jeder der beiden Reihen getestet.

Bei den ersten Testläufen, stellte sich schnell heraus, dass die oberen Neuronen keinerlei Reaktion auf die Veränderung der Parameter zeigten. Deren Membranspannung hielt sich konstant bei allen Neuronen auf etwa 1600 mV. Dieses Verhalten war allerdings auf einen Fehler im Bitfile zurückzuführen, welcher von A. Hartel gefixt wurde und somit nicht mehr auftrat.

Um schnell und einfach feststellen zu können, wann das Neuron spikete und wann nicht, nahm ich mehrere Spikeplots, wie in Abb. 3.1 zu sehen, auf. Hierbei ließ ich mir im späteren Test, immer den Mittelwert und die Standardabweichung der Datenpunkte ausgeben, so konnte ich leicht festlegen, ob das Neuron

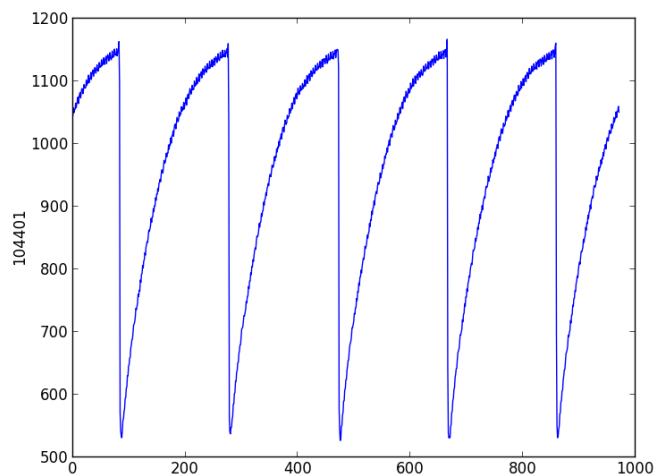


Abbildung 3.1: Spikes des 0. Neurons zur objektiven Charakterisierung spikender Neuronen

spikete und sogar qualitativ sagen, mit welcher Amplitude es dies tut. Nach ein paar Probeplots legte ich fest, dass man bei einer Standardabweichung von ca. 9-10 mV von Spikes reden konnte.

⁷Digital-to-Analog-Converter

Eine kleine Nebenentdeckung hierbei war, dass die Spikes der unteren Neuronenreihe die erwartete Form (Abb 3.1) hatten, wohingegen die oberen Neuronen doch recht verrauschte Spikemus-
ter ergaben, worauf ich mich hier aber nicht weiter konzentrierte. Mittels des oben genannten Sweepstest, erstellte ich nun folgende Tabellen.

$V_t : E_I -$	200	300	400	500	600	700	800	900	1000	1023
200	MW: 529 STD: 7	MW: 536 STD: 7	MW: 536 STD: 7	MW: 537 STD: 7	MW: 536 STD: 7	MW: 538 STD: 7	MW: 536 STD: 7	MW: 538 STD: 7	MW: 540 STD: 7	MW: 536 STD: 7
250	MW: 530 STD: 8	MW: 532 STD: 7	MW: 536 STD: 7	MW: 537 STD: 7	MW: 535 STD: 7	MW: 534 STD: 7	MW: 534 STD: 7	MW: 535 STD: 7	MW: 534 STD: 7	MW: 535 STD: 7
300	MW: 472 STD: 7	MW: 534 STD: 9	MW: 540 STD: 12	MW: 542 STD: 12	MW: 542 STD: 13	MW: 542 STD: 13	MW: 542 STD: 13	MW: 542 STD: 13	MW: 543 STD: 13	MW: 543 STD: 13
350	MW: 454 STD: 7	MW: 542 STD: 7	MW: 597 STD: 29	MW: 584 STD: 32	MW: 583 STD: 32	MW: 582 STD: 33	MW: 583 STD: 32	MW: 583 STD: 33	MW: 583 STD: 33	MW: 583 STD: 33
400	MW: 470 STD: 7	MW: 547 STD: 7	MW: 625 STD: 7	MW: 695 STD: 7	MW: 646 STD: 53	MW: 641 STD: 53	MW: 644 STD: 52	MW: 643 STD: 53	MW: 643 STD: 53	MW: 641 STD: 54
450	MW: 459 STD: 7	MW: 542 STD: 7	MW: 625 STD: 7	MW: 702 STD: 7	MW: 778 STD: 7	MW: 731 STD: 67	MW: 755 STD: 54	MW: 755 STD: 55	MW: 774 STD: 7	MW: 779 STD: 7
500	MW: 472 STD: 7	MW: 545 STD: 7	MW: 622 STD: 7	MW: 697 STD: 7	MW: 770 STD: 7	MW: 814 STD: 7	MW: 818 STD: 7	MW: 790 STD: 7	MW: 798 STD: 7	MW: 820 STD: 7

$V_t : E_I -$	200	300	400	500	600	700	800	900	1000	1023
200	MW: 526 STD: 4	MW: 528 STD: 4	MW: 537 STD: 6	MW: 530 STD: 7	MW: 530 STD: 7	MW: 531 STD: 8	MW: 536 STD: 7	MW: 539 STD: 8	MW: 532 STD: 7	MW: 539 STD: 8
250	MW: 526 STD: 4	MW: 535 STD: 5	MW: 536 STD: 6	MW: 539 STD: 7	MW: 535 STD: 8	MW: 538 STD: 8	MW: 536 STD: 8	MW: 537 STD: 8	MW: 534 STD: 8	MW: 537 STD: 8
300	MW: 475 STD: 3	MW: 536 STD: 9	MW: 541 STD: 13	MW: 544 STD: 14	MW: 542 STD: 16	MW: 545 STD: 14	MW: 545 STD: 16	MW: 544 STD: 16	MW: 544 STD: 16	MW: 549 STD: 14
350	MW: 476 STD: 4	MW: 563 STD: 4	MW: 586 STD: 36	MW: 582 STD: 38	MW: 580 STD: 39	MW: 585 STD: 38	MW: 584 STD: 38	MW: 580 STD: 40	MW: 583 STD: 39	MW: 584 STD: 39
400	MW: 473 STD: 3	MW: 563 STD: 4	MW: 670 STD: 4	MW: 633 STD: 60	MW: 632 STD: 61	MW: 629 STD: 62	MW: 627 STD: 64	MW: 630 STD: 64	MW: 626 STD: 64	MW: 630 STD: 65
450	MW: 474 STD: 3	MW: 562 STD: 4	MW: 670 STD: 4	MW: 703 STD: 82	MW: 679 STD: 86	MW: 677 STD: 86	MW: 678 STD: 87	MW: 677 STD: 86	MW: 677 STD: 87	MW: 679 STD: 87
500	MW: 480 STD: 4	MW: 563 STD: 4	MW: 674 STD: 4	MW: 822 STD: 4	MW: 742 STD: 106	MW: 734 STD: 107	MW: 728 STD: 110	MW: 731 STD: 107	MW: 734 STD: 107	MW: 729 STD: 110
550	MW: 475 STD: 4	MW: 558 STD: 4	MW: 677 STD: 4	MW: 829 STD: 4	MW: 843 STD: 129	MW: 789 STD: 132	MW: 790 STD: 132	MW: 789 STD: 133	MW: 789 STD: 134	MW: 783 STD: 133
600	MW: 475 STD: 3	MW: 566 STD: 4	MW: 671 STD: 3	MW: 828 STD: 4	MW: 974 STD: 4	MW: 870 STD: 157	MW: 867 STD: 155	MW: 854 STD: 156	MW: 855 STD: 150	MW: 847 STD: 154
650	MW: 472 STD: 3	MW: 561 STD: 4	MW: 676 STD: 4	MW: 827 STD: 4	MW: 983 STD: 4	MW: 1127 STD: 5	MW: 960 STD: 175	MW: 955 STD: 172	MW: 1090 STD: 110	MW: 986 STD: 170
700	MW: 482 STD: 4	MW: 559 STD: 3	MW: 670 STD: 3	MW: 819 STD: 4	MW: 975 STD: 4	MW: 1120 STD: 5	MW: 1156 STD: 5	MW: 1182 STD: 5	MW: 1145 STD: 5	MW: 1166 STD: 5

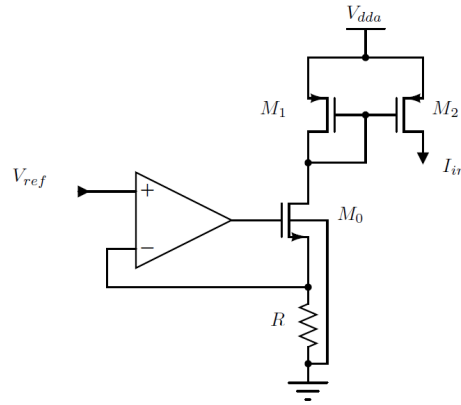
Abbildung 3.2: Obere Tabelle entspricht oberen Neuronen, untere Tabelle den Unteren. Mittelwert und Standardabweichung zur charakterisierung der Spikes, sowie farbliche Unterlegung: Grün = spiked, gelb = keine ausgeprägten Spikes, rot = spiked nicht.

Die Tabellen sind jeweils nur Ausschnitte aus den kompletten Datentabellen, um die wichtigen Bereiche besser darzustellen. Bei allen anderen Parameterpaarungen wurden keinerlei Spikes detektiert. Dazu kommt noch, dass ich die Sweepsschritte in den interessanten Bereichen von 100 auf 50 herabgesetzt habe. Wie man in Abb. 3.2 erkennen kann, gibt es bei den unteren Neuronen einen größeren Bereich, in dem sie spiken. Woran dies nun genau liegt ist bis dato noch nicht geklärt. Generell sollten Spikes zu sehen sein, sobald das Leakagepotential überhalb der Thresholdspannung liegt. Erst ab einem V_t von 350 werden die ersten richtigen Spikes detektiert. Warum allerdings in hohen Wertebereichen die Spikes ausbleiben, ist ebenso wie der Unterschied zwischen oberer und unterer Reihe, noch zu untersuchen. Auch gut zu erkennen, aufgrund der mit ausgegebenen Standardabweichung, dass bei anwachsender Differenz der beiden Parameter V_t und E_I , die Amplituden der Spikes ansteigen.

Mit Hilfe dieser Messung habe ich mir nun einen ersten Überblick verschafft in welchen Bereichen die Neuronen das tun, was man auch erwartet. Deshalb verwende ich im Folgenden für meine Messungen meist eines der unteren Neuronen. Zur Vollständigkeit habe ich im Anhang (Kap. 6) in Tabelle 1 die kompletten Neuronenparametereinstellungen in einer Tabelle dargestellt.

3.2 Externer Stromstimulus

Um ein Neuron durch einen externen Strom zu stimulieren, bedient man sich eines periodischen, rechteckigen Stromsignals, wobei ein von Sebastian Millner entwickelter Stromkreis (s. Abb.3.3), als Stromquelle dient. Der Stromstimulus kann, periodisch oder einmalig, 33 Samples mit 10-Bit Auflösung und einer mit 4-Bit einstellbaren Sample-Länge wiedergeben.



Quelle:[2]

Abbildung 3.3: Stromquelle für externen Stromstimulus des Neurons

Die Spannung V_{ref} , welche über ein Floating Gate erzeugt wird, so wie alle Neuronparameter, erzeugt einen Strom, der mit dem Spannungsabfall zu dem Widerstand R proportional ist. Über den Stromspiegel, bestehend aus M_1 und M_2 , wird der gespiegelte Strom auf das Neuron gegeben.

3.2.1 Experimente mit Stromstufen

Es sei hier noch einmal gesagt, dass es bei dieser Arbeit auch um das Testen der vorhandenen Software auf Funktionalität und Genauigkeit ging. Somit geht es in diesem Kapitel um das Testen der oben genannten Stromstufe. Insbesondere überprüfte ich die Periodizität und den Einfluss der Stärke der Stufe auf das Feuerverhalten der Neuronen.

In Abb 3.4 ist die Membranspannung bei aktivem und periodischem externen Stromstimulus zu sehen. Zur genaueren Untersuchung der Funktion und der Ansteuerung per Software des Strominputs, führte ich die im Folgenden erwähnten Tests und Experimente durch.

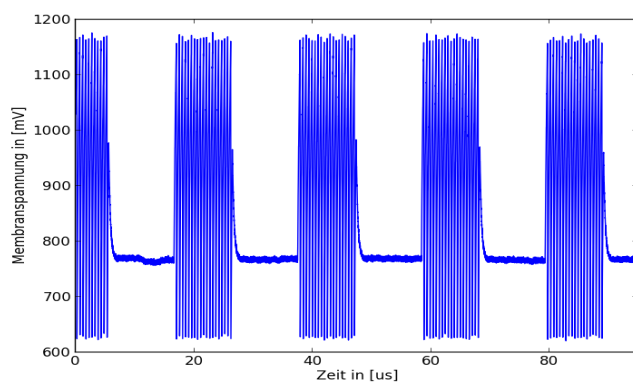


Abbildung 3.4: Periodisches Feuermuster eines stimulierten Neurons

Die ersten Tests waren ausschließlich qualitativer Natur, so stimulierte ich ein Neuron mit verschiedenen Stromstufenformen. Wie schon in Grundlagen erwähnt, sollte ein Ansteigen des Inputstroms eine Erhöhung der Frequenz hervorrufen.

Dies ist anhand der Formel (2.3) zu erschließen. Ebenso achtete ich darauf, ob die Periodizität und die Länge des Strompulses, die angelegt wurden, auf der Membran wiederzuerkennen waren. Zur Veranschaulichung meiner Messungen habe ich 2 Beispiel-Plots aufgenommen, welche in Abb 3.5 und Abb 3.6 zu sehen sind.

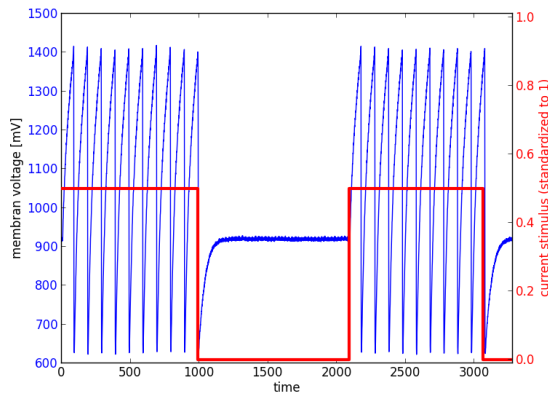


Abbildung 3.5: Membranspannung (blau) mit Stimulation einer einfachen periodischen Stromstufe (rot) (Stärke = 511 DAC)

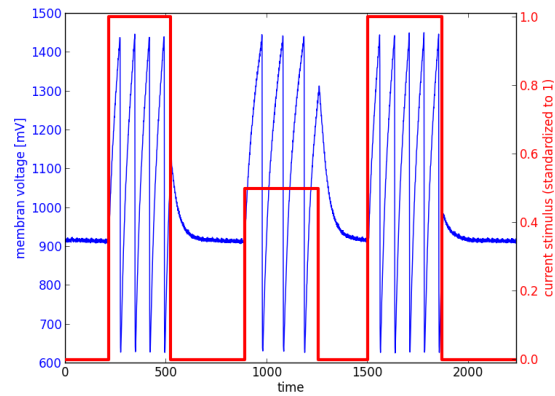


Abbildung 3.6: Membranspannung (blau) mit Stimulation auf einer periodischen Stromstufe unterschiedlicher Stärke (rot) (Stärke = 511 DAC & 1023 DAC)

Der rot gezeichneten Verlauf der Stromstufe beruht nicht auf gemessenen Daten, da es nicht möglich ist den Stromoutput der Schaltung zu messen. So sind die Längen der Rechteck-Funktionen nur nach der unten folgenden Formel (3.1) berechnet und die Startpunkte identisch zu den gemessenen Daten gewählt. Man erkennt, dass die Längen der Strompulse simultan auf die Membran übertragen werden. Auch das Erhöhen der Feuerfrequenz bei höherer Stromstufe ist leicht auszumachen. Wie schon erwähnt, sind dies nur 2 Beispiele aus meinen Messungen, die klar machen sollen, dass die Implementierung des Stromstimulus in der Software zuverlässig und präzise arbeitet. Somit kann der externe Stromstimulus für weitere Experimente genutzt werden. Ein weiteres Experiment ist in Kapitel 3.2.2 beschrieben.

Um nun die Stromstufe quantitativ zu vermessen, muss man hier zuerst erläutern, wie sich deren Periode theoretisch berechnen lässt. Wie schon erwähnt, hat der Strominput eine einstellbare Pulslänge und Anzahl der Strompulse. Da die maximal einstellbare Länge 32 beträgt, also 33 Möglichkeiten, und der Floating-Gate-Parameter von 0 - 15 DAC-Wert einstellbar ist, berechnet sich die Periode einer Stromstufe theoretisch wie folgt:

$$T_{theo} = 25MHz \cdot 33 \cdot (pulselength + 1) \quad (3.1)$$

Hierbei ergeben sich die 25 MHz aus der Taktfrequenz des Chips (100 MHz), die durch 4 geteilt wird. Das Addieren einer 1 in der Klammer, kommt durch den einfachen Grund zustande, dass der Ausdruck für T_{theo} nicht 0 werden darf, da sonst kein externer Strominput erfolgt.

Um diese theoretische Vermutung zu überprüfen, sweepte ich den Parameter “pulselength“ von 0 bis 15 durch und errechnete zu dem Wert die Periode der Stromstufe. Die Länge der Stromstufe wurde auf 1 gesetzt. Somit wurde das Neuron kurz stimuliert, die Membranspannung stieg an und so konnte man anhand dieser sich wiederholenden Spannungsspitzen die Periode ermitteln. Die Stärke der Strompulse beließ ich für dieses Experiment auf 300 DAC-Wert. Diese entstandene Grafik ist in Abb 3.7 zu sehen. Den Fehler dieser Messungen folgerte ich aus dem mehrmaligen Auslesen der Periode innerhalb einer aufgenommenen Messung. Was heißen soll, dass ich für jeden durchgesweepeten Wert eine Trace der Membranspannung aufnahm. Diese speicherte ich und las anschließend bei jeder Trace die Zeit von Spannungsspitze zur nächsten mehrmals aus, indem ich immer wieder andere Spannungsspitzen auswählte. So bildete ich jeweils die Mittelwerte und nahm als Fehler die Standardabweichung an.

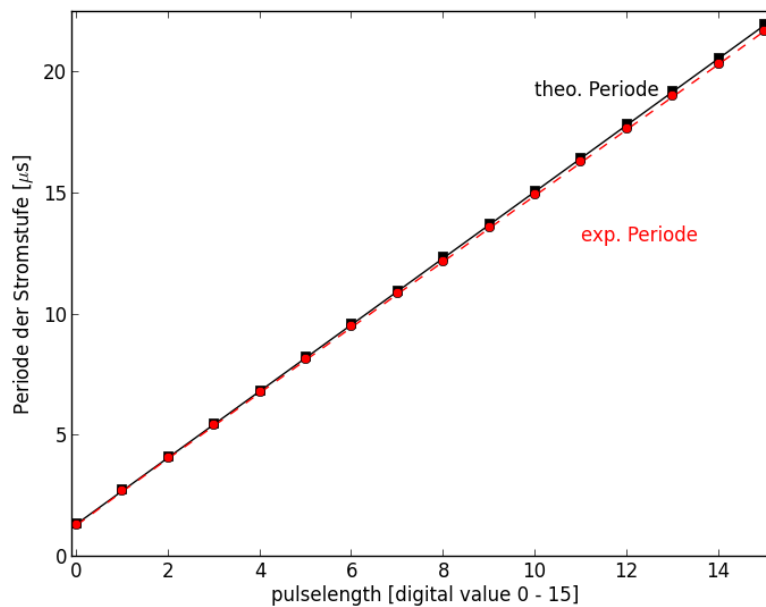


Abbildung 3.7: Quantitativer Vergleich der experimentellen und theoretisch errechneten Periode der implantierten Stromstufe (Fehlerbalken sind zu klein und deshalb nicht zu erkennen)

Auch hier komme ich zu einer guten Übereinstimmung der experimentellen mit den errechneten Werten. Warum die Mittelwerte mit größer werdender pulselength leicht abweichen, konnte ich leider nicht herausfinden. Aber trotzdem stimmen die Werte innerhalb der Fehlergrenzen überein. Eine kurze Variation der theoretischen Steigung beim Plotten, hat ergeben, dass bei einer Steigung von 24.33 MHz die experimentellen Werte genau mit den theoretischen übereinstimmen. Somit schließe ich nun den Teil, der sich ausschließlich mit der Funktion und der Genauigkeit der Stromstufe beschäftigte, ab. Da man sich nun sicher sein konnte, dass der externe Strominput zuverlässig benutzbar ist, verwendete ich diesen weiterhin für nachfolgende Experimente.

3.2.2 Bestimmung der Membranzeitkonstanten

Wie schon in Kapitel 2.2.1 dargelegt, hat man bei dem Multi-Compartment-Chip die Möglichkeit, die Membrankapazität auf 64 verschiedene Größen einzustellen. Da dies eine Neuerung bei diesem Chip ist, galt es natürlich auch diese verstellbare Größe zu charakterisieren. Zudem hat man aufgrund des formellen Zusammenhangs zwischen Membrankapazität und Membranzeitkonstanten die Möglichkeit, die Konduktanz für jedes einzelne Neuron zu errechnen. Dieser Sachverhalt wird im nächsten Kapitel noch einmal von Nutzen sein.

Wie schon in der Formel (2.4) verwendet, ergibt sich der Zusammenhang zwischen Membrankapazität und Membranzeitkonstanten wie folgt:

$$\tau_m = \frac{C_{mem}}{g_l} \quad (3.2)$$

Um nun die Membranzeitkonstante zu messen, stimulierte ich ein Neuron mit der oben erläuterten Stromstufe, so dass die Membranspannung ansteigt, das Neuron aber nicht spiked. Theoretisch erwartet man einen linearen Anstieg des Membranpotentials bei Strominput und einen exponentiellen Abfall nach Abschalten des Inputs. Nur wenn wir uns wie in Abb. 3.8 eine Beispiel-Kennlinie eines Transistors anschauen, ist zu erkennen, dass ein Transistor in Sättigung als Stromquelle fungiert und der Anstieg der Membrankapazität bei externem Stromstimulus sollte linear verlaufen.

Mit fortschreitender Zeit beim Aufladen des Kondensators kommt man in den linearen Bereich der Kennlinie und dann ist der Transistor als Widerstand zusehen. Somit ist der Anstieg des Membranpotentials zu Beginn linearer Natur und mit fortschreitender Stimulierung, nähert es sich einer Exponentialkurve, s. Formel (3.3), an. Dies liegt an der Spannungsdifferenz zwischen Membranspannung und der V_{dd} -Spannung, siehe auch Kap.

2.3. Der Abfall der Membrankapazität lässt sich ebenfalls aus Abb. 2.5

schließen und wird als exponentieller Abfall betrachtet, siehe Formel (3.4). Zur Bestimmung von τ_m stimulierte ich ein Neuron mittels Strominputs (Stärke = 300 DAC) und fittete dann einen exponentiellen Abfall wie in (3.4) an die Datenpunkte an (s. Abb 3.9). Ich habe hierbei sowohl Anstieg als auch Abfall gefittet, werde aber im Weiteren mich nur auf den Abfall konzentrieren. Trotz des Mixes aus linear und exponentiell fittete ich die oben genannten Ab- und Anstiege, auch wenn dadurch mitunter ein Fehler bei der Berechnung der Zeitkonstanten zustande kommt.

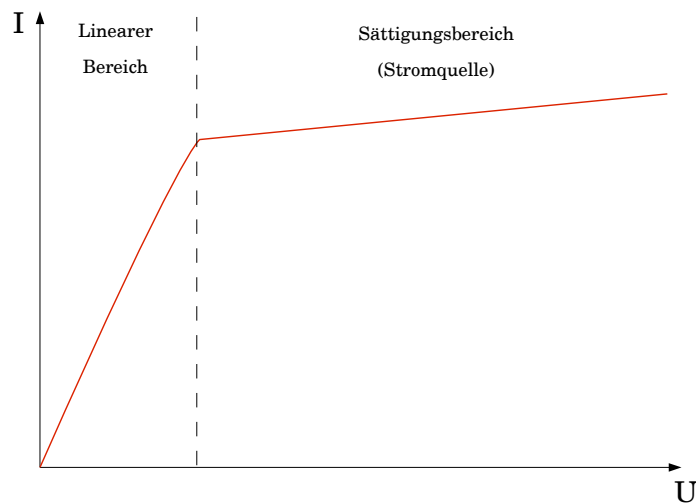


Abbildung 3.8: Beispiel-Kennlinie eines Transistors aufgeteilt in linearen Bereich und Sättigungsbereich

$$\overbrace{U_{mem}(t) = R \cdot I_0 \cdot (1 - e^{-\frac{t}{\tau_m}})}^{\text{Aufladen}} \quad (3.3)$$

$$\underbrace{U_{mem}(t) = R \cdot I_0 \cdot (e^{-\frac{t}{\tau_m}})}_{\text{Entladen}} \quad (3.4)$$

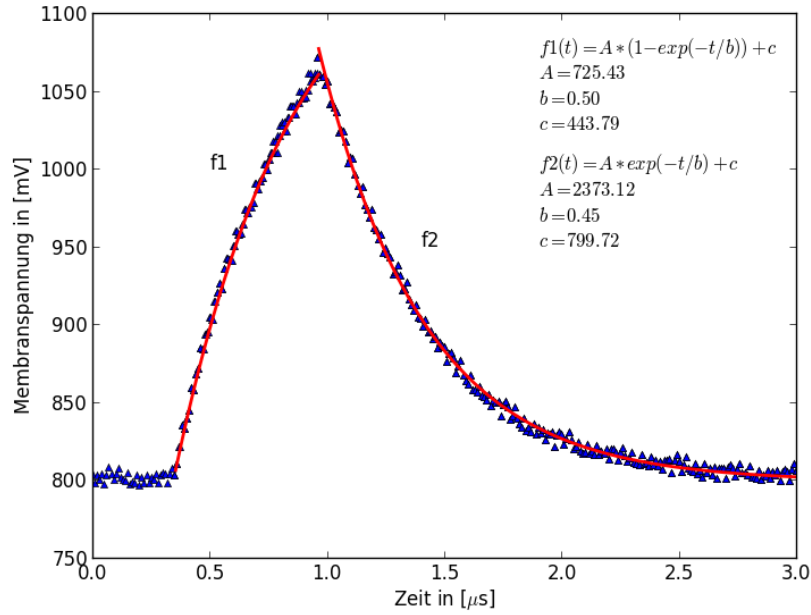


Abbildung 3.9: Angefitteter exponentieller Anstieg und Abfall des Membranpotential bei externen Stromstimulus eines Neurons

Wie schon oben erläutert, bestimmte ich die Membranzeitkonstante aus dem Fit des Abfalls des Potentials. Um im Folgenden einen statistischen Fehler für die Berechnung der Zeitkonstanten angeben zu können, habe ich den ADC⁸ so eingestellt, dass er immer eine Trace mit mehreren Spannungsan- und abstiegen aufnahm. Durch die Periodizität der Stromstufe war dies kein Problem. Jetzt konnte ich innerhalb eines Datensatzes τ_m mehrfach auslesen und mittels Standardabweichung einen Fehler abschätzen.

Nun komme ich zu einem Teil in dem eine Neuerung, die erstmals auf dem MCC implementiert wurde, untersucht wird. Wie schon in Kapitel 2.2.1 beschrieben, ist es möglich die Kapazitätswerte zu variieren. Und zwar gibt es 64 einstellbare Werte, die über 6 Bits eingestellt werden können.

Zu Demonstrationszwecken ist in Abb 3.10 der Verlauf der Membranspannung des 0. Neurons einmal mit einer eingestellten Kapazität von $C = 800$ fF und einmal von $C = 1600$ fF zu erkennen.

⁸Analog-to-Digital-Converter

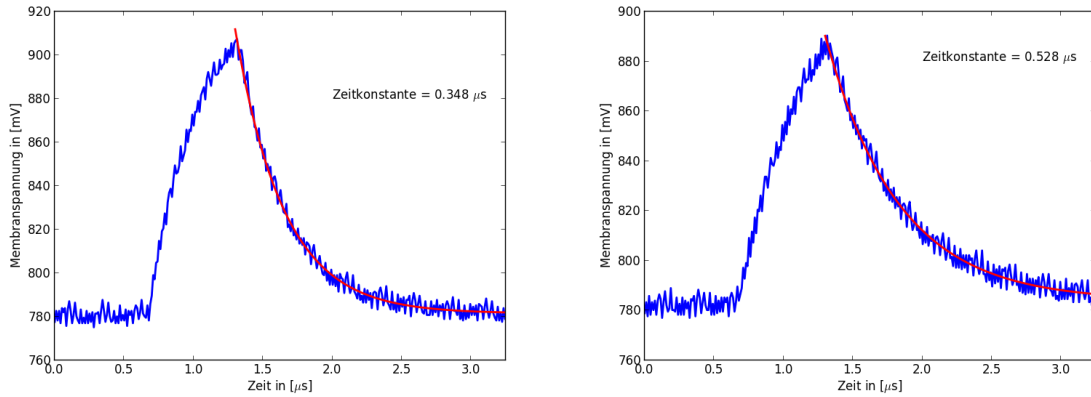


Abbildung 3.10: Stromstimuli eines Neurons bei 2 unterschiedlichen Membrankapazitäten (links: 800 fF ; rechts: 1600 fF)

Wie man von der Theorie her erwartet hat, weist der Abfall bei größerem Kondensator eine höhere Membranzeitkonstante τ_m auf. Da nun die Kapazitäten sich genau um einen Faktor 2 unterscheiden, ist es auf den ersten Blick verwunderlich, dass dieser Faktor nicht auch bei den Zeitkonstanten wieder zu finden ist. Wie man aber in Abb. 3.11 erkennt, ergibt das Auftragen der Kapazitäten gegen die Zeitkonstanten keine Ursprungsgerade und somit wird beim Verdoppeln des Arguments der Funktion, der Funktionswert nicht auch doppel so groß. Der Grund für diesen Offset liegt darin, dass der Strom nicht über einen echten Widerstand von der Membran fließt, sondern über einen OTA. Dessen maximaler Strom ist nämlich begrenzt und somit kann die Membranzeitkonstante nicht beliebig klein werden.

Mit Hilfe der folgenden Messung war es dann möglich den Widerstand des RC-Gliedes und damit die Konduktanz g_l zu bestimmen. So sweepte ich alle 64 Kapazitätswerte durch und ermittelte jeweils die Membranzeitkonstante τ_m . Aufgrund des Zusammenhangs aus Formel (2) sollte sich beim Auftragen der Zeitkonstanten gegen die Kapazitätswerte eine Gerade ergeben, deren Steigung das Inverse der Konduktanz und damit der Widerstand ist.

In Abb. 3.11 ist die aufgenommene Gerade zu sehen. Mit Hilfe eines linearen Fits der aufgenommenen Daten konnte ich die Steigung genau ermitteln. Da diese ja dem Widerstand entspricht, bildete ich das Inverse und erhielt die Konduktanz des RC-Gliedes. Der mit steigender Kapazität größer werdende Fehler ist damit zu erklären, dass bei höheren Kondensatorgrößen der Abfall des Membranpotentials zeitlich länger dauert. Somit ist mehr Spielraum beim Anfitzen der Exponentialkurve vorhanden, welcher zu größeren Abweichungen führt.

Um nun einen ersten Vergleichswert zu haben, habe ich den Neuron-Parameter I_{gl} so eingestellt, dass ein Biasstrom von ca. $1 \mu A$ anliegt. So war es mir möglich den errechneten Wert der Konduktanz mit einer Grafik aus der Dissertation von Sebastian Millner zu vergleichen, welche in Abb. 2.5 aus Kapitel 2.3 dargestellt ist. Hierauf ist der Verlauf der Konduktanz für verschiedene Ströme gezeigt. Die aufgetragene Spannungsdifferenz resultiert aus der Differenz zwischen Leakage-Potential und Membranpotential. Da sich dieser Spannungsunterschied jedoch während der Messung immer verändert, lässt sich keine genaue Aussage über den abgelesenen Leitwert machen, sondern nur eine Einordnung der Größenordnungen angeben. Wie auch schon in der oben erwähnten Grafik abzulesen ist, achtete ich darauf, dass die Spannungsdifferenz nie über 400 mV stieg. So kam ich zu dem Ergebnis, dass hier der experimentell gemessene und der abgelesene Wert in der selben Größenordnung liegen.

Um einen Überblick über alle Neuronen zu bekommen, habe ich die Konduktanz für jedes Neuron ausgerechnet und blockweise in ein Histogramm geplottet. Der verwendete Parametersatz

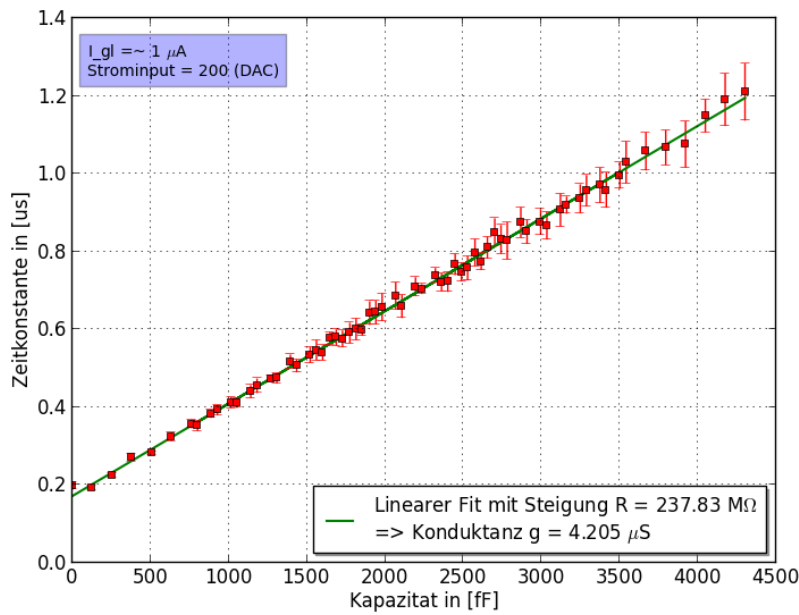


Abbildung 3.11: Grafische Berechnung der Konduktanz als Inverses der Steigung

ist im Anhang in Tabelle 2 nachzuschlagen. In Abb. 3.12 ist das Histogramm über alle Neuronen zu sehen. Es lässt sich eine Spanne von ca. $2.5 - 6.5 \mu\text{S}$ und ein Mittelwert von etwa $4.7 \mu\text{S}$ herauslesen. Was, wenn man es mit den Werten aus den der Grafik 2.5 aus Kapitel 2.3 vergleicht, doch recht gut zusammenpasst. Denn man muss bedenken, dass die Spannungsdifferenz beim Abfall des Membranpotential nach dem Stromstimulus sich stetig ändert und somit die Konduktanz auch. Hinzu kommt natürlich hier auch der Umstand, dass die Floating Gates eine gewisse Schwankung aufweisen und somit die Messung beeinflussen. Hauptgrund für die Schwankungen ist jedoch das Mismatch der Transistoren. Diese sind zwar identisch entworfen und sollten für alle Schaltungen verschiedener Neuronen die selben Dimensionen haben, sind jedoch real nicht gleich groß (produktionsbedingt) und verursachen somit von einander abweichende Werte für g_l . Selbstverständlich ist auch die Tatsache, dass sowohl der exponentielle Abfall der Membranspannung als auch die lineare Abhängigkeit der Membranzeitkonstanten von der Kapazität angefitet wurden und somit fehlerbehaftet sind.

Denn durch den Umstand, dass die Spannungsdifferenz nie größer als 400 mV betrug, lieferten die Datenpunkte des öfteren eine recht große Streuung, was einen genauen Fit nicht ganz einfach machten. In Abb. 3.13 und 3.14 sind nochmal 2 Histogramme über jeweils einen Neuronenblock aufgeführt.

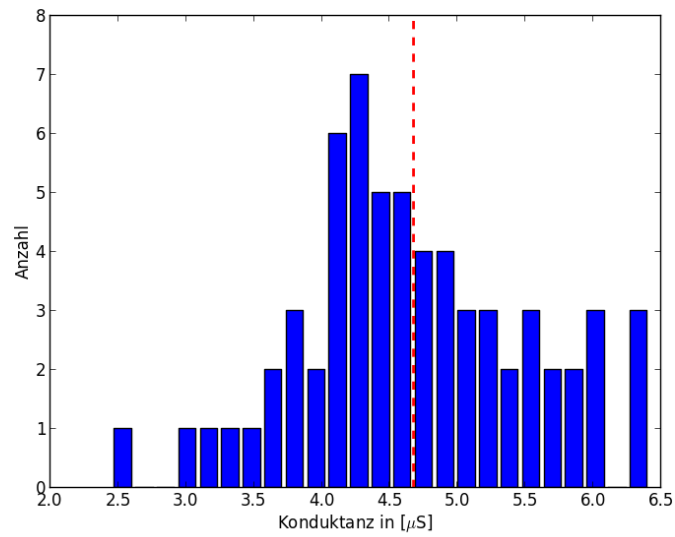


Abbildung 3.12: Histogramm der ermittelten Konduktanzwerte aller Neuronen mit veranschaulichtem Mittelwert (rote gestrichelte Linie)

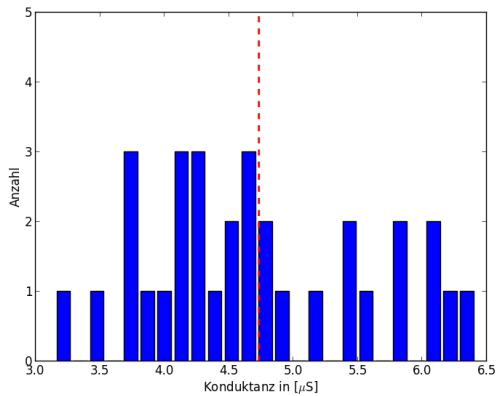


Abbildung 3.13: Histogramm der ermittelten Konduktanzwerte der oberen Neuronen mit veranschaulichtem Mittelwert (rote gestrichelte Linie)

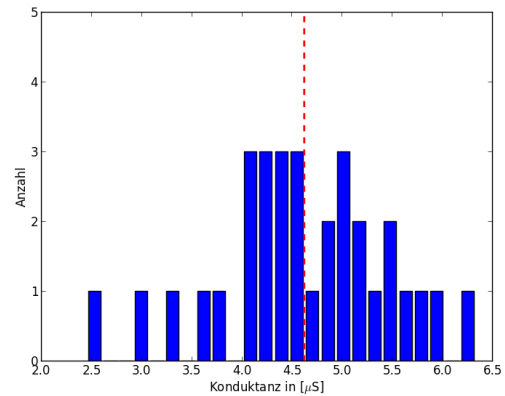


Abbildung 3.14: Histogramm der ermittelten Konduktanzwerte der unteren Neuronen mit veranschaulichtem Mittelwert (rote gestrichelte Linie)

Schön zu erkennen ist, dass die Mittelwerte sowohl bei den oberen als auch bei den unteren Neuronenblöcke ungefähr bei $4.7 \mu S$ liegen, was sich dann ja schon bei dem Histogramm aus Abb. 3.12 niederschlägt.

3.2.3 Feuerratenuntersuchung

Bei diesem Abschnitt widme ich mich noch einmal der Spikefrequenz, genauer gesagt, der Frequenz in Abhängigkeit des Inputstroms. Denn wie schon in Kap. 2.1.1 theoretisch beleuchtet, sollte sich bei steigendem externen Stromstimulus die Frequenz gemäß Formel (2.2) dem Inversen der Refraktärzeit annähern. Um dieses Verhalten zu untersuchen und dann mit den Theoriewerten zu vergleichen, sweepte ich in 50er Schritten die Stärke der Stromstufe durch. Hierbei musste natürlich gewährleistet sein, dass das Neuron nur bei Stimulierung durch den Strominput zum Spiken gebracht wird. Was heißen soll, dass die Threshold-Spannung über dem Leakage-Potential liegen muss. Mit Hilfe eines kleinen Python-Programms habe ich bei den aufgenommenen Spikes des 0. Neurons, zu jedem Stromwert wurde eine Trace aufgenommen, die Zeit zwischen 2 Spike-spitzen gemessen und somit die Frequenz errechnet. Um aber auch hier einen Fehler abschätzen zu können, habe ich die Messung mehrmals wiederholt. Durch Mittelung kam ich dann zu meinen Messwerten und durch Berechnen der Standardabweichung den dazugehörigen Fehler. Das Ergebnis dieser Messung ist in folgenden Abb. 3.15 zu sehen.

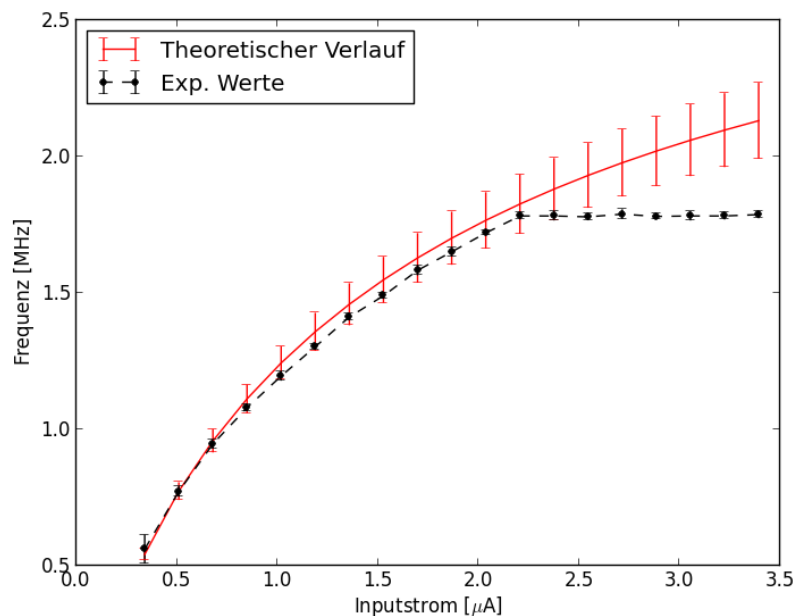


Abbildung 3.15: Grenzverhalten der Frequenz bei steigendem Strominput. Theoretisch erwarteter Verlauf (rote Linie); experimentell gemessene Werte (schwarze Linie)

Was auch die Formeln (2.2) und (2.3) zeigen, dass bei steigendem Strom im Grenzfall die Frequenz nur noch von τ_{ref} abhängt, lässt sich in dieser Grafik leider nur erahnen. Um dieses Verhalten zu hinterfragen, habe ich die Refraktärzeit bei denselben Parametereinstellungen gemessen, wie bei dem Experiment zuvor. Der Parameter um die Refraktärzeit einzustellen heißt I_{pl} . Um diesen sowohl zu charakterisieren als auch die gesuchte Größe zu erhalten, sweepte ich ihn im Bereich von 0 - 200 DAC in 20er-Schritten durch. Nur bis 200, weil bei größeren Werten sich der Wert von τ_{ref} nicht signifikant ändert. Die Daten wertete ich dann mit einem Python-Programm aus, welches die Zeitdifferenz zwischen 2 Spikes misst. Durch mehrmaliges Auslesen innerhalb eines Datensatzes ergab sich der Fehler durch die Standardabweichung. Allerdings musste ich beim Plotten den 0. Wert herausnehmen, da dieser zu große Schwankungen aufwies.

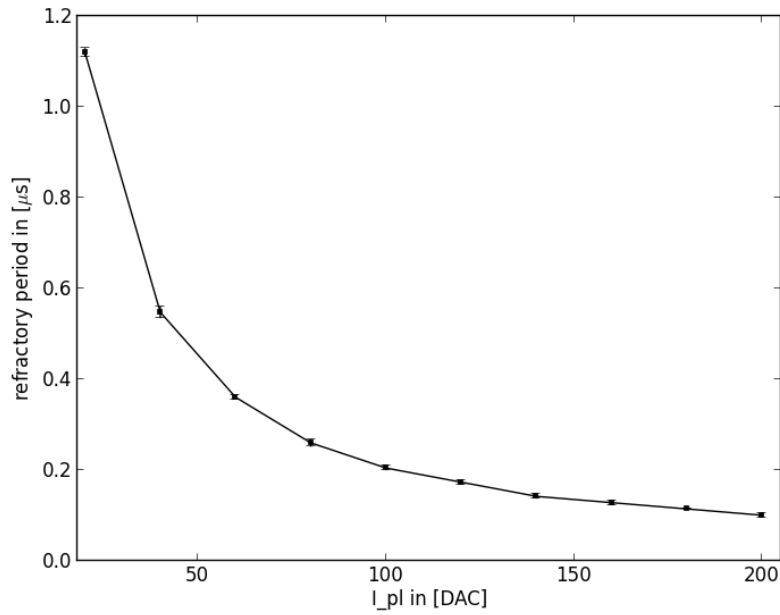


Abbildung 3.16: Charakterisierung des Parameters I_{pl} zur Ermittlung der Refraktärzeit τ_{ref}

Schaut man sich nun Abb. 3.16 an, liegt der Wert für τ_{ref} , bei einem eingestellten DAC-Wert von 80, wie bei dem oben erläuterten Feuerratenexperiment ca. bei $0.26 \mu s$. Die Grenzfrequenz berechnet sich dann zu:

$$f_{grenz} = \frac{1}{\tau_{ref}} = \frac{1}{0.26 \mu s} = 3.85 \text{ MHz} \quad (3.5)$$

Nun ist der Inputstrom nicht beliebig groß, soll heißen, er hat einen endlichen maximalen Wert. Somit wird man hiermit niemals in den Grenzbereich der Frequenz kommen, der aus Formel (3.5) anzunehmen ist. Hinzu kommt das abrupte Abbrechen des Anstiegs, was, wenn man sich nochmal die Schaltung aus Abb. 3.3 anschaut, folgendem Grund hat. Die Gatespannung des Transistors M_0 , die aus dem Operationsverstärker kommt, kann maximal 1.8 V betragen. Der Operationsverstärker versucht stets gleiche Eingangsspannungen zu erhalten und kann so maximal die eben erwähnten 1.8 V als Gatespannung liefern. Ab diesem Zeitpunkt steuert der Transistor nicht mehr aus, und somit erhalten wir einen Maximalwert für den Strominput. So ist bei ca. $2.25 \mu A$ die Maximalfrequenz erreicht.

Der theoretische Verlauf wurde mittels Formel (2.3) und den, nach dem Experiment, gemessenen Floating-Gate-Spannungen ausgerechnet. Um hierbei auch auf den Wert von I zu kommen, nahm ich die Tatsache zur Hilfe, dass die Stärke des Strominputs von 0 - 1023 eingestellt werden kann. So wird der Strom wie in Formel (3.6) berechnet. Hier resultieren die 1.8 V daher, dass dies Maximalspannung ist und der Widerstandswert aus der Dimensionierung der Schaltung. So kam ich auf die theoretische Verlaufslinie der Frequenz.

$$I = \frac{\frac{\text{Digitalwert der Stromstufe}}{1023} \cdot 1.8 \text{ V}}{518.6 \text{ k}\Omega} \quad (3.6)$$

Im Anhang 6 in Tabelle 3 sind die restlichen verwendeten Werte zu finden.

Die Differenz zwischen Theorie und Experiment ist vielleicht zum Einen auf die Unsicherheit der Floating-Gate-Spannungen zurückzuführen. Ebenso könnte ein möglicher Fehler in dem Programm, welches die Frequenz ausliest, stecken. Denn dieses Python-Programm ermittelt diese, indem es die Zeitdifferenz zwischen zwei Datenpunkten, welche durch Randbedingungen festgelegt werden, misst. Eine bessere Methode wäre natürlich die digitale Spikeauslese, welche aber zum Zeitpunkt des Experiments nicht einsatzfähig war. Auch der Parameter I_{pl} , welcher die Refraktärzeit einstellt, weißt doch gewisse Schwankungen auf, die den Verlauf der Soll-Kurve, auch bei kleinen Abweichungen, beeinflusst. So wie I_{pl} sind alle Floating Gate Spannungen fehlerbehaftet, gerade bei häufigem Programmieren. Diese Fehler habe ich durch mehrmaliges Auslesen der programmierten Werte und Bilden der Standardabweichung ermittelt. Somit spielt die Fehlerfortpflanzung hier eine entscheidende Rolle. Mittels eben dieser Fehlerfortpflanzung habe ich die Fehlerbalken der Theoriekurve abgeschätzt. Die Fehler wurden anhand Formel (3.7) errechnet. Der Inputstrom ist natürlich auch fehlerbehaftet. Es ist jedoch nicht möglich diesen zu messen. Deshalb wird hier für die Berechnungen kein Fehler angenommen.

$$\Delta f = \sqrt{\left(\Delta \tau_{ref} \frac{\delta f}{\delta \tau_{ref}}\right)^2 + \left(\Delta V_{reset} \frac{\delta f}{\delta V_{reset}}\right)^2 + \left(\Delta U_t \frac{\delta f}{\delta V_t}\right)^2 + \left(\Delta E_l \frac{\delta f}{\delta E_l}\right)^2 + \left(\Delta g_l \frac{\delta f}{\delta g_l}\right)^2} \quad (3.7)$$

Auch die hier verwendeten Zahlen sind im Anhang in Tabelle 3 festgehalten.

Aber dennoch decken sich die Werte, gerade im Anfangsbereich der Kurve. Die theoretischen und experimentellen Werte passen innerhalb der Fehlergrenzen gut zusammen. Somit ist auch hier wieder ein weitere Testaspekt des Multi-Compartment Chips abgedeckt. Hiermit schließe ich mit den Einzel-Neuron Experimenten ab und widme mich im nächsten Kapitel den Multi-Compartment Messungen.

4 Multi-Compartment-Experimente

In diesem Abschnitt sollen nun die Multi-Compartment Eigenschaften genauer untersucht und getestet werden. Auch hier betrat man, beispielsweise bei den Neuronenverbindungen via Inter-Compartment Resistances, Neuland und ich war erst einmal darauf bedacht, einfache Stromkreise zu realisieren, um somit den Chip wieder ein Stück vermessen zu können.

4.1 Charakterisierung der Inter-Compartment Konduktanzen

Um die Inter-Compartment Konduktanzen zu charakterisieren und zu vermessen, mache ich mir das Prinzip des Spannungsteilers (s. Kap 2.3.2) zu nutze. So schließe ich beliebig viele Neuronen wie folgt zusammen. Der Membran-Output wird über die Konduktanz mit der ersten Reihe des Dendrite Routing verbunden. Nun wird beim benachbarten Neuron der Membran-Input ohne Konduktanz auch mit der ersten Reihe der Matrix verbunden. Schaut man sich nochmal Grafik 2.6 und 2.7 an, sieht man, dass die 1. und die 2. Reihe immer 2 benachbarte Neuronen abwechselnd miteinander verbinden. So kann ich beliebig viele Neuronen als Spannungsteiler aufbauen. Die maximale Anzahl an Compartments, die eingebunden werden können, ist natürlich begrenzt durch die Anzahl an Neuronen pro Block.

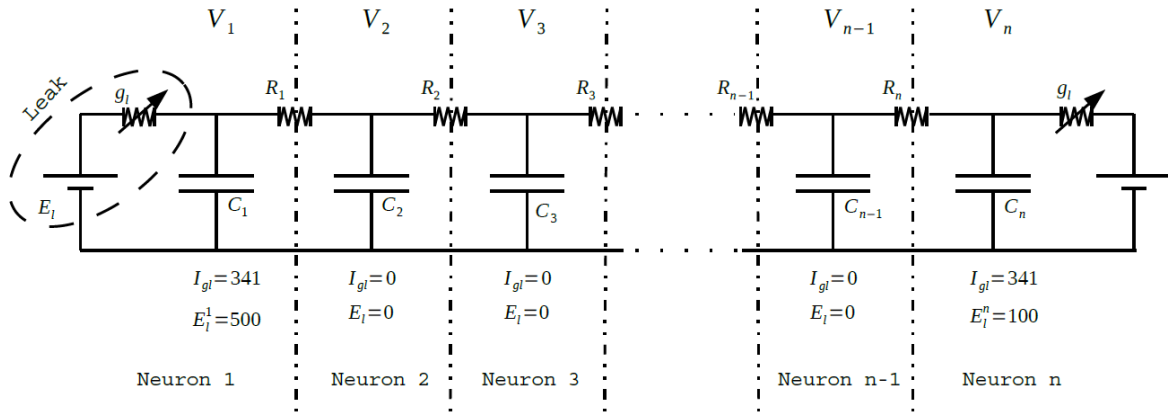


Abbildung 4.1: Schematische Schaltung zur charakterisierung der Inter-Compartment Konduktanzen realisiert durch einen Spannungsteiler (die Widerstände zwischen den Compartments sind ebenfalls einstellbar und somit kalibrierbar)

Die Grafik 4.1 ist in die einzelnen Compartments eingeteilt. Am 1. Neuron wird eine Leakage-Spannung von 500 DAC-Wert angelegt und am n-ten Neuron auf 100 DAC, im Gegensatz zu den Restlichen, wo diese auf 0 gesetzt wird. Lediglich beim 1. und dem n-ten Neuron wird der Parameter I_{gl} auf 341 DAC gesetzt, was einem Strom von ca. $1 \mu A$ entspricht. Diese Einstellung ist wichtig, um später V_n zu berechnen. Hierbei sollte sich ein statischer Zustand bilden.

Wie nun angedeutet, wird das Konzept des Spannungsteilers herangezogen, um die Inter-Compartment Resistance, beziehungsweise deren Widerstandswerte, nur über die entsprechenden Spannungsabfälle zu errechnen. Der fließende Strom muss hier nicht bekannt sein. Die Spannungsabfälle werden mittels Messung der einzelnen Membran-Spannungen ($V_1, V_2, \dots$) ermittelt. Somit können die Konduktanzwerte mit Hilfe folgenden Formeln berechnet werden:

$$V_n = \frac{V_{n-1} - E_l^n}{R_n + g_l} \cdot g_l \quad (4.1)$$

$\vdots$

$$\frac{(V_3 - V_4)}{(V_2 - V_4)} = \frac{R_3}{R_2 + R_3} \Rightarrow V_3 = \frac{V_2 R_3 + V_4 R_2}{R_2 + R_3} \quad (4.2)$$

$$\frac{(V_2 - V_3)}{(V_1 - V_3)} = \frac{R_2}{R_1 + R_2} \Rightarrow V_2 = \frac{V_1 R_2 + V_3 R_1}{R_1 + R_2} \quad (4.3)$$

$$V_1 = V_1 \quad (4.4)$$

Leider konnte ich diese Idee der Charakterisierung nicht in die Tat umsetzen, da die Zeit gegen Ende der Arbeit zu knapp bemessen war, und man noch nicht sicher sein konnte, dass die FPGA-Programmierung und nötigen Bitfiles zuverlässig funktionieren. Somit habe ich hier nur eine Möglichkeit dargelegt, wie dieses Experiment aufgebaut und ausgewertet werden könnte. Um dann auch wirklich eine richtige Charakterisierung angeben zu können, sollte man sich klarmachen, wie groß die Konduktanz bei gegebener Spannungsdifferenz ist. Soll heißen, die errechneten Werte müsste man gegen zugehörigen Spannungsabfall auftragen und erhält somit eine Art "Eichkurve". Erwartetes Ergebnis kann man anhand einer Simulation aus der Dissertation von S. Millner [2] abschätzen. Man sollte diese Messungen auch für verschiedene Ströme I_{gl} durchführen um ein breites Spektrum zu erhalten und optimalerweise eine Kurve findet, bei der die Konduktanzwerte ungefähr konstant sind.

5 Zusammenfassung

Im ersten Teil der Arbeit wurden Experimente und Messungen an einzelnen Neuronen durchgeführt. So wurden zu Beginn ein einfacher Spiketest über alle Neuronen der 2 Blöcke gemacht. Darauf folgte die Vermessung der Stromstufe, die es ermöglicht, einen externen Inputstrom auf das Neuron zu geben. Auch die neu implementierte Variabilität bei den Membrankapazitäten wurde zunutze gemacht, um die Leakage-Konduktanzen zu bestimmen, was durch die Berechnung der Membranzeitkonstante bei Stromstimulus des Neurons möglich war. Eine Vermessung aller Neuronen auf ihre Konduktanzwerte wurde mittels Histogramm veranschaulicht.

Der Zusammenhang zwischen Feuerratenfrequenz und Stromstimulus machte die Feuerratenuntersuchung deutlich. Hierbei deckten sich theoretische und experimentelle Werte innerhalb der Fehlergrenzen gut. An dieser Stelle sei noch ein Wort zu den Fehlern gesagt, die im Laufe der Arbeit immer wieder zur Diskussion standen. Im Anhang in Abbildung 6.1 wurde eine Floating Gate Spannung über einen Zeitraum von etwa 1.4 ms aufgenommen. Dies geschah mittels vorhandenem C++ Programm. Für ca. 0.5 ms wurde dann die Clock ausgeschaltet und man erkennt deutlich, dass in diesem Bereich das Rauschen wesentlich geringer ist. Natürlich ist klar, dass man diese Fehlergröße nicht eliminieren kann, aber dennoch kann man bei der Fehlerdiskussion dies im Hinterkopf behalten. Hier auch aufgreifen möchte ich die Tatsache, dass die Floating Gate Spannungen nie ganz auf 0 programmiert werden können. Dies war der Fall beim Reduzieren des impelmetierten AdEx-Modells auf das LIF-Modell. Minimaler Wert ist etwa 40 mV, was das Auslesen der Floating Gates mittels vorhandenem Programm ergab. Dies könnte durchaus einen Einfluss auf die Experimente haben.

Im zweiten Teil sollte es die Aufgaben sein, die Inter-Compartment Resistances zu charakterisieren und zu vermessen. Die Überlegung in Bezug auf den Aufbau und die Auswertung dieses Experiments habe ich oben beschrieben. Doch leider ließ die fehlende Zeit keine quantitativen Messungen zu. Ein paar kurze Testläufe schlugen fehl, was vielleicht noch auf einen Softwarefehler zurückzuführen ist.

Dennoch kann man mit den Ergebnissen zufrieden sein. Man muss dies immer unter dem Aspekt sehen, dass hier hauptsächlich das Testen der Software auf Ansteuerung der Hardware und Funktionalität im Fokus stand. Unter diesem Gesichtspunkt sind die dargelegten Ergebnisse zufriedenstellend und somit ist eine erster Grundstein zur weiteren Vermessung des Multi-Compartment Chips gelegt.

Nächste interessante Aufgabe ist definitiv das Durchführen des von mir schon erläuterten Experiments bezüglich der Inter-Compartment Konduktanzen. Blockübergreifende Verbindungen zwischen Neuronen oder weitere Dendritenverbindungen bieten ein breites Spektrum an Vermessungsmöglichkeiten. Auch das Untersuchen der Linearen Kabeltheorie ([7]) wäre eine weitere Aufgabe, die mittels des Multi-Compartment Chips zu bewältigen und zu verifizieren wäre.

Man sieht, es gibt noch reichlich zu tun, um eine vollständige Vermessung des MCC zu erhalten, sodass dieser vielleicht in naher Zukunft in einer Erweiterung des Wafer Scale Systems seinen Einsatz findet.

6 Anhang

Parameter	DAC-Wert
I_{pl}	511
$I_{spikeamp}$	511
$I_{spikelen}$	511
$I_{biascas}$	600
V_{br}	511
I_{OPbias}	511
V_{bb}	900
$V_{reset<0>}$	300
$V_{reset<1>}$	400
V_{resdll}	200
V_{clr}	511

Tabelle 1: Neuronparametersatz

Parameter	DAC-Wert
V_t	800
E_l	400
I_{gl}	341
Stärke Strominput	200
Länge des Strompulses	0 - 1

Tabelle 2: Neuronparametersatz

Parameter	DAC-Wert	Analogwerte *	Fehler
V_t	425	853 mV	1.43 mV
E_l	400	830 mV	3.47 mV
V_{reset}	300	648 mV	2.7 mV
I_{gl}	341	1 μ A	-
$I_{pl} (\hat{=} \tau_{ref})$	80	$\hat{=} 0.26 \mu$ s **	0.014 μ s
Länge des Strompulses	0 - 32	-	-
g_l	-	4.205 μ S ***	0.01 μ S
Membrankapazität C_m	30	1900.5 fF	-

Tabelle 3: Parametersatz (* FG ausgelesen mittels vorhandenem C++-Programm; ** entnommen aus Grafik 3.16; *** entnommen aus Grafik 3.11)

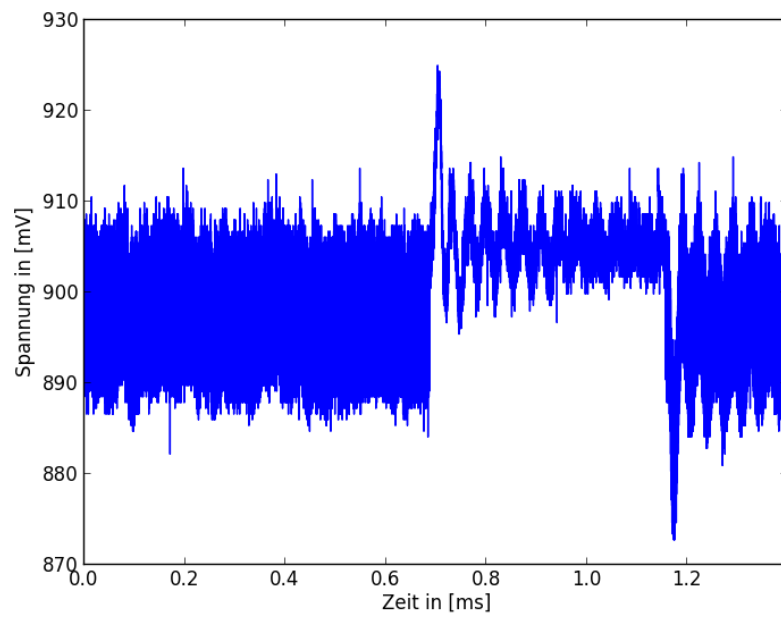


Abbildung 6.1: Trace einer FG-Spannung bei zeitweiser ausgeschaltener Clock

7 Quellen

Literatur

- [1] Gerstner and Kistler: “Spiking Neuron Models. Single Neurons, Populations, Plasticity“ - Cambridge University Press, 2002
- [2] Disseratation Sebastian Millner: “Development of a Multi-Compartment Neuron Model Emulation“ - 2012, Universität Heidelberg
- [3] Wulfram Gerstner and Romain Brette (2009), Scholarpedia:
http://www.scholarpedia.org/article/Adaptive_exponential_integrate-and-fire_model Stand: 08.01.2014
- [4] http://www.ipn.uni-kiel.de/aktuell/ibo/lernblaetter/lernblatt_aktionspotenzial.pdf Stand: 08.01.2014
- [5] http://www.frontiersin.org/human_neuroscience/10.3389/neuro.09.031.2009/abstract Stand: 13.01.2014
- [6] Johannes Schemmel, Daniel Brüderle, Andreas Grübl, Matthias Hock, Karlheinz Meier and Sebastian Millner: “A Wafer-Scale Neuromorphic Hardware System for Large-Scale Neural Modeling“ - <http://www.kip.uni-heidelberg.de/Veroeffentlichungen/details.php?id=2018> 2010
- [7] Christof Koch: “Biophysics of Computation - Information Processing in Single Neurons“ 1999
- [8] Rüdiger Brause: “Neuronale Netzwerke“ - B.G.Teubner Stuttgart 1991

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den.....