

Institut für Signalverarbeitung und Prozessrechentechnik  
der Universität zu Lübeck  
Direktor: Prof. Dr.-Ing. Alfred Mertins

# **Data Acquisition and Processing System for Multisite Neural Recordings**

Inauguraldissertation  
zur Erlangung der Doktorwürde (Dr.-Ing.)  
der Universität zu Lübeck  
– Technisch-Naturwissenschaftliche Fakultät –

vorgelegt von  
**Diplom-Informatiker Andre Folkers**  
aus Westerstede

Lübeck, Oktober 2008

1. Berichterstatter: Professor Dr. Ulrich G. Hofmann
2. Berichterstatter: Professor Dr. Dr. Siegfried J. Pöpl

# Vorwort

„Lübeck ist 'ne schöne Stadt, weil sie sieben Türme hat!“, so schallt es jetzt oft von den hinteren Sitzen des Autos, wenn ich mit meiner Familie unterwegs bin. Als Stadt zum Studieren hatte ich damals Lübeck vor allem wegen des dort angebotenen Nebenfachs Medizinische Informatik gewählt. Wie sich herausstellte, war es eine gute Wahl. Aus der Verbindung von Informatik und Medizin ergaben sich schon während des Studiums viele interessante Aufgaben für Seminare und Praktika. So war ich froh, dass ich nach Abschluss meines Informatikstudiums am Institut für Signalverarbeitung und Prozessrechentechnik (ISIP) die Möglichkeit bekam, als wissenschaftlicher Mitarbeiter genau auf dieser Schnittstelle zu arbeiten.

Für die Unterstützung, die ich von meinen Kollegen, Freunden und von meiner Familie erhalten habe, möchte ich mich bedanken. Besonderer Dank gebührt dabei meinem Doktorvater Herrn Privatdozent Ulrich G. Hofmann, der meine Arbeit hervorragend betreut hat und dessen Ideen und Engagement es mir ermöglicht haben, diese Arbeit zu erstellen. Für die freundschaftliche Zusammenarbeit danke ich auch meinen Kollegen Alexandru Condurache, Kerstin Menne, Ingo Stuke und Daniel Toth.

Florian Mösch und Thomas Malina danke ich für ihren Beitrag, den sie im Rahmen ihrer Diplomarbeiten bei der Entwicklung des Datenaufnahmesystems geleistet haben. Für die Durchsicht der Dissertation bedanke ich mich bei Olaf Christ und bei Dr. Peter Vorländer.

Bei Herrn Professor Til Aach möchte ich mich für die freundliche Aufnahme am ISIP und für sein fachliches Interesse an meiner Arbeit bedanken und bei Herrn Professor Alfred Mertins für seine Unterstützung in der Endphase der Arbeit. Für die Übernahme des Koreferats danke ich Herrn Professor Siegfried J. Pöpl und für den Vorsitz der Prüfungskommission Herrn Professor Thomas Martinetz.

Bei den Projektpartnern aus dem VSAMUEL Konsortium möchte ich mich für die gute und produktive Zusammenarbeit bedanken. Für die Einblicke in ihre experimentelle Arbeit und die Evaluierung des Systems danke ich Marco de Curtis und Gerardo Biella vom Nationalen Institut für Neurophysiologie in Mailand, Ken Yoshida und Winnie Jensen der Universität Aalborg, Erik de Schutter, Patriq Fagerstedt und Antonia Volny-Luraghi von Universität Antwerpen. Thomas Recording danke ich dafür, dass ich die Entwicklung des Datenaufnahmesystems bei ihnen noch einige Monate fortführen konnte. Eine weitere Möglichkeit, das Datenaufnahmesystem im experimentellen Umfeld zu testen, verdanke ich Winrich Freiwald und Heiko Stemman von der Universität Bremen.

Mein besonderer Dank gilt meiner Familie, die mich in großartiger Weise unterstützt hat. Ohne ihre Unterstützung wäre diese Arbeit wohl kaum jemals ferti-

gestellt worden. Meinen Eltern Enno und Christiane Folkers danke ich dafür, dass sie mir mein Studium ermöglicht haben, das die Grundlage für diese Arbeit bildet. Bei meinen Schwiegereltern Arthur und Ursula Springfeld bedanke ich mich dafür, dass sie mir durch ihre Hilfe so manche Stunde Zeit verschafft haben. Meinen Kindern Antonia, Karolin, Frederik und Teresa danke für die Fröhlichkeit mit der sie mir Motivation gegeben haben. Den größten Dank möchte ich meiner Frau Verena aussprechen, die mich immer wieder dazu gebracht hat weiterzumachen.

*Für meine Familie*

Verl, im Oktober 2008



# Contents

<b>Vorwort</b>	<b>iii</b>
<b>Zusammenfassung</b>	<b>viii</b>
<b>Summary</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Neural activity</b>	<b>4</b>
2.1 Nervous system . . . . .	4
2.2 Neural cell . . . . .	5
2.2.1 Morphological classification . . . . .	7
2.2.2 Functional classification . . . . .	7
2.3 Glial cells . . . . .	9
2.4 Layered structure of the brain . . . . .	9
2.5 Action potentials . . . . .	9
2.5.1 Membrane potential . . . . .	10
2.5.2 Action potential generation . . . . .	12
2.6 Field potentials . . . . .	13
2.7 Tuning curves and information coding . . . . .	14
<b>3 Neural signal acquisition</b>	<b>17</b>
3.1 Sensing bioelectrical signals . . . . .	19
3.2 Extracellular potentials . . . . .	20
3.3 Probes for intracavitary or intratissue recordings . . . . .	20
3.4 Silicon-based multichannel microelectrodes . . . . .	22
3.5 Manufacturing VSAMUEL probes . . . . .	23
<b>4 Signal processing</b>	<b>27</b>
4.1 Digital filter . . . . .	27
4.1.1 FIR and IIR filter . . . . .	29
4.2 Wavelet Transformation . . . . .	30
4.2.1 Wavelet filter bank . . . . .	33
4.2.2 Translation invariance . . . . .	35
4.3 Fast lifting wavelet transform . . . . .	36
4.3.1 Example . . . . .	41
4.3.2 Mapping onto integer values . . . . .	43

4.3.3	Implementing lifting steps on a C6701 . . . . .	45
<b>5</b>	<b>Applying Wavelet Transform to Neural Signals</b>	<b>51</b>
5.1	Low pass, band pass, and high pass wavelet filter . . . . .	51
5.2	Compression and Denoising . . . . .	51
5.3	Spike detection . . . . .	54
5.4	Spike sorting . . . . .	55
5.4.1	Spike features . . . . .	57
5.4.2	Principal component analysis . . . . .	58
5.4.3	Online wavelet based spike sorting . . . . .	59
<b>6</b>	<b>Data Acquisition System</b>	<b>64</b>
6.1	Hardware . . . . .	64
6.1.1	Probes . . . . .	64
6.1.2	Amplifier . . . . .	65
6.1.3	AD conversion . . . . .	68
6.1.4	Digital signal processor board M67 . . . . .	71
6.1.5	Synchronization . . . . .	73
6.1.6	Host PC . . . . .	74
6.1.7	Thermal issues . . . . .	75
6.2	Software . . . . .	77
6.2.1	DSP application . . . . .	77
6.2.2	Host application . . . . .	82
<b>7</b>	<b>Biological validation</b>	<b>103</b>
7.1	Recording from the guinea pig brain . . . . .	103
7.1.1	Isolation of the guinea pig brain . . . . .	104
7.1.2	Recordings in the medial entorhinal cortex . . . . .	104
7.2	Recordings from rat cerebellum . . . . .	110
<b>8</b>	<b>Discussion</b>	<b>112</b>
<b>A</b>	<b>Microstructure techniques</b>	<b>115</b>
A.1	Metal evaporation . . . . .	115
A.2	Lift-off process . . . . .	115
A.3	PECVD . . . . .	116
A.4	Deep RIE . . . . .	116
<b>B</b>	<b>M67 and Ad16 extensions and modifications</b>	<b>118</b>
B.1	SyncLink and ClockLink . . . . .	118
B.2	Synchronizing the AD16 boards . . . . .	120
B.3	Trigger Input . . . . .	120
<b>C</b>	<b>Realtime operating system</b>	<b>122</b>
C.1	Overview . . . . .	122
C.2	Texas Instruments DSP/BIOS . . . . .	123
C.2.1	Threads . . . . .	123

C.2.2	Interthread communication and synchronization . . . . .	124
C.2.3	Timer . . . . .	125
C.2.4	Profiling . . . . .	125

# Zusammenfassung

Die hier vorgelegte Arbeit ist im Rahmen des EU-geförderten Projekts VSAMUEL (Development of a Versatile System for Advanced Neuronal Recordings with Multi-site Microelectrodes) entstanden. Ziel des Projektes war es, ein leicht verwendbares System zur Aufzeichnung von verschiedenen neurophysiologischen Signalen in lebendem Nervengewebe zu entwickeln. Die schwedische Firma Acreo hat dabei mit Mikrostrukturtechniken gefertigte Sonden auf Siliziumbasis entwickelt und hergestellt, die bis zu 64 Aufzeichnungspunkte besitzen. Von der deutschen Firma Thomas Recording aus Gießen wurden passende Vor- und Hauptverstärker entworfen und gebaut. Die Aufgabe des Instituts für Signalverarbeitung und Prozessrechentechnik der Universität zu Lübeck war die Entwicklung eines Datenaufnahmesystems, welches in dieser Arbeit im Besonderen dargestellt wird. Das komplette System wurde in Zusammenarbeit mit den Projektpartnern aus verschiedenen Bereichen der neurophysiologischen Grundlagenforschung praktisch evaluiert.

Die Informationsverarbeitung des zentralen und des peripheren Nervensystems basiert im Wesentlichen auf der Kommunikation von Nervenzellen, die sich sowohl anatomisch als auch durch ihre Funktion unterscheiden. Trotzdem sind sie sich in ihrem grundsätzlichen Aufbau ähnlich. Eine typische Nervenzelle hat einen Zellkörper aus dem ein oft stark verzweigter Dendritenbaum und ein weniger verzweigtes Axon hervorgeht. Das Axon hat an seinen Enden jeweils eine Synapse über die eine chemische, auf Neurotransmittern basierende Verbindung zu anderen Nervenzellen hergestellt wird. Vereinfacht gesagt sammelt eine Nervenzelle ihre Eingaben am Dendritenbaum und gibt ihre Ausgaben über das Axon weiter. Ausgaben werden in Form von Aktionspotentialen im Zellkörper am Ansatzpunkt des Axons gebildet, wenn die Zelle durch inhibitorische oder exzitatorische Eingaben an ihrem Dendritenbaum in Summe über einen bestimmten Schwellwert hinaus depolarisiert worden ist. Somit spiegeln einzelne Aktionspotentiale die Aktivität von einzelnen Nervenzellen wider. Die Potentiale an den synaptischen Verbindungen und insbesondere die synchronen Änderungen durch die Aktivität einer großen Gruppe von Nervenzellen addiert sich zu sogenannten Feldpotentialen. Bemerkenswert ist dabei, dass Feldpotentiale allem Anschein nach nicht direkt durch synchron erzeugte Aktionspotentiale entstehen. Die genauere Untersuchung des Zusammenhangs zwischen Feldpotentialen und Aktionspotentialen und dessen Bedeutung für die Informationsverarbeitung im Nervensystem erfordert die zeitgleiche Messung von beiden Potentialarten in einer möglichst großen Region.

Die im VSAMUEL Projekt entwickelten gabelförmigen Siliziumsonden ermöglichen Messungen von bis zu 64 Aufzeichnungspunkten pro Sonde. Die effektive Nutzung dieser Sonden erfordert neben entsprechenden Vor- und Hauptverstärker-

stufen ein leistungsfähiges Datenaufnahmesystem, das simultan von vielen Kanälen mit hoher Auflösung aufzeichnen kann und diese Daten in Echtzeit verarbeitet.

Ziel beim Entwurf der Signalverarbeitungsmethoden war die Entwicklung eines Rahmens zur Echtzeitverarbeitung, der verschiedene Verarbeitungs- und Analysemöglichkeiten bietet. Das Datenaufnahmesystem berechnet dabei mit Hilfe der Wavelet-Transformation eine Zeit-Frequenz-Analyse der aufgenommenen Daten. Basierend auf den resultierenden Wavelet Koeffizienten können die aufgenommenen Signale komprimiert oder entauscht werden. Weiter ist es möglich Hochpass-, Bandpass- oder Tiefpassfilter zu realisieren. Die echtzeitfähige Implementierung der Waveletzerlegung basiert auf dem sogenannten Lifting Schema. Diese Methode zerlegt eine Stufe einer Wavelet Filterbank in mehrere sogenannte Lifting Schritte, wobei redundante Berechnungen, die bei einer naiven Implementierung auftreten, vermieden werden, was wiederum die Anzahl der benötigten Instruktionen verringert. In Rahmen der Arbeit ist ein Tool entwickelt worden, welches aus gegebenen Lifting Schritten optimierten Assembler Code für den C6701 oder einen Pentium III erzeugt.

Bei der Aufzeichnung von Aktionspotentialen ist deren Detektion basierend auf Schwellwerten und Darstellung vorgesehen. Die Klassifikation von aufgezeichneten Aktionspotentialen auf Basis der in Echtzeit berechneten Waveletkoeffizienten war nicht möglich, da durch die Translationsvarianz der verwendeten dyadischen Waveletzerlegung mit Unterabtastung die Varianz in den Koeffizienten zu stark war. Einen vielversprechender Ansatz, dieses Problem zu überwinden, liegt in der Dual-Tree Complex Wavelet Transformation.

Das Datenaufnahmesystem basiert auf der DSP Karte M67 von Innovative Integration (Thousand Oaks, Kalifornien, USA), die mit dem C6701 Prozessor von Texas Instruments bestückt ist. Die M67 Karte wurde mit zwei A/D Wandler Modulen vom Typ AD16 desselben Herstellers erweitert. Jedes AD16 Modul kann simultan 16 Kanäle mit einer Abtastrate von 50 kHz und 16 bit Auflösung aufzeichnen. Durch Synchronisation der zwei AD16 Module pro M67 Karte und durch weitere Synchronisation von bis zu vier M67 Karten wurde ein Datenaufnahmesystem für 128 Kanäle gebaut. Auf jeder Karte wird eine DSP Applikation ausgeführt, die die aufgezeichneten Daten verarbeitet und über den PCI Bus in den Speicher des Hostrechners transferiert. Die Datenaufnahmeapplikation auf dem Hostrechner fügt die Daten von den einzelnen Karten zu einem Datenstrom zusammen und speichert sie dann in einer Datei. Zu Beginn einer Aufzeichnung wird die verwendete Sonde, die Abtastrate und die Art der Filterung festgelegt. Ferner man kann die Kanäle gruppieren und dabei pro Gruppe weitere Aufzeichnungsparameter definieren.

Die Datenaufnahmeapplikation verwaltet eine frei erweiterbare Liste von Sonden, wobei jeweils das Layout der Aufzeichnungspunkte und deren Zuordnung zu den Kanälen am Verstärker hinterlegt sind. über eine aus diesen Daten erzeugte schematische Darstellung der Sonde kann man die Verstärkungsfaktoren für die Aufzeichnungspunkte verändern und an den programmierbaren Hauptverstärker übertragen.

Die aufgezeichneten Daten können in einem virtuellen Oszilloskop oder in einem Blueplot dargestellt werden. Der Blueplot zeigt die Amplitude der Kanäle farblich kodiert an und ermöglicht die gleichzeitige Darstellung besonders vieler Kanäle. Die Datenaufnahmeapplikation kann für einen einzelnen frei wählbaren Kanal ein Spektrogramm anzeigen. Weiter ist es möglich für eine Auswahl an Kanälen, die vom

Anwender festgelegt werden kann, Wavelet basierte Filter und Entrauschungsalgorithmen zu konfigurieren und zu aktivieren.

Das Datenaufnahmesystem wurde zusammen mit Projektpartnern aus der neurophysiologischen Forschung im praktischen Einsatz erprobt. Die Forschungsgruppe am Institut für Theoretische Neurobiologie der Universität Antwerpen hat dabei mit der 64 Kanal Variante des Datenaufnahmesystems erfolgreich Aktionspotentiale von einzelnen und auch von mehreren Zellen im Kleinhirn von Ratten aufgezeichnet. Die am VSAMUEL Projekt beteiligte Forschungsgruppe vom Nationalen Institut für Neurologie in Mailand, Italien, hat mit einem Prototypen des Datenaufnahmesystems Feldpotentiale und auch Aktionspotentiale in einer in-vitro Präparation eines Meerschweinchen-Gehirns gemessen.

Das VSAMUEL Projekt konnte also mit der erfolgreichen Entwicklung eines kompletten, integrierten neurophysiologischen Datenaufnahmesystems basierend auf allgemein verfügbaren DSP Karten und neu entwickelten Siliziumsonden abgeschlossen werden.

# Summary

The work presented in this thesis was part of the EU-funded project VSAMUEL (Development of a Versatile System for Advanced Neuronal Recordings with Multi-site Microelectrodes), which aims to develop easy-to-use instrumentation for multi-channel recordings from functioning and living nervous tissue. It integrates advanced micro structuring techniques to batch fabricate multi-site microelectrode probes with novel PC based real-time data acquisition and signal processing.

The main objective of this PhD thesis is to develop a data acquisition system prototype which allows simultaneous recording from 128 channels at 50 kHz per channel with 16 bit resolution and the signal processing. The signal processing includes a wavelet based digital filter, the detection of action potentials and their classification. A core component of the data acquisition system is a DSP board which is used for data acquisition and for digital filtering. The host PC software provides different means of visualization and online processing, like e.g. virtual scope, blue plot, and spectrogram. A graphical user interface for data acquisition, data review, and data analysis has been developed with respect to a good design, usability, and speed. The whole system is evaluated together with experimental partners for different application area, i.e. peripheral nerves, cerebellum, and cortex.

# Chapter 1

## Introduction

Ob wir nun aber unsere Bemühung bloß für anatomisch erklären; so müßte sie doch, wenn sie fruchtbar, ja wenn sie in unserem Falle auch nur möglich sein sollte, stets in physiologischer Rücksicht unternommen werden. Man hat also nicht bloß auf das Nebeneinandersein der Teile zu sehen, sondern auf ihren lebendigen, wechselseitigen Einfluß, auf ihre Abhängigkeit und Wirkung.

Johann Wolfgang von Goethe  
(Entwürfe zu einem osteologischen Typus, 1796)

The nervous system is one of the most complex systems developed by nature. Over centuries scientists have tried to reveal the secrets of our brain [55]. Hippocrates (460-379 b.c.) identified the brain as being responsible for our sensation and intelligence. In the 16th century the macroscopic anatomy of the brain was examined by Leonardo da Vinci and Andreas Vesalius. In the end of the 18th century Luigi Galvani found that nerves and muscles can be stimulated by electricity. About 50 years later Jan Purkinje and Theodor Schwann described the microscopic structure of the brain and the peripheral nerves. Only in the last decades scientists had the technical instruments to measure activity of single neurons directly. A.L. Hodgkin and A.F. Huxley were the first, who derived and interpreted action potentials from inside neurons.

With this effort many details about the anatomic and physiologic structure have been found. However still, the brain is the least understood organ of our body [99]. The progress of the neurophysiological research was accompanied by a technical progress of measurement instruments. The behaviour of single neurons has been examined in-depth [62]. During the last decade the focus of research has moved on to the interaction details between multiple neurons [81]. For future research it is necessary to have probes which can intercept the activity of multiple single neurons working together. Such probes have to be designed with respect to many constraints, e.g. not damaging the tissue, physical stability, and signal quality. Furthermore, in order to truly find insights into the interactions among several neurons, it is mandatory to record their activity at many positions. This will lead to a huge amount of multichannel data which needs to be acquired and analysed.



The project VSAMUEL funded by the European Union aimed to develop those easy-to-use instrumentation for multi-channel recordings from functioning and living nervous tissue spanning a broad range of neurobiological questions. Figure 1.1 presents an overview of all project partners and their tasks. The group at Acreo designed silicon-based microelectrodes with up to 64 recording sites, Thomas RECORDING GmbH designed a 64 channel preamplifier and main amplifier, and the Institute for Signalprocessing at the University of Lübeck built the data acquisition system for 128 channels. The whole system was evaluated together with the experimental partners for different application area, i.e. peripheral nerves, cerebellum, and cortex. In this thesis we focus on the development and some exemplary uses of the data acquisition system.

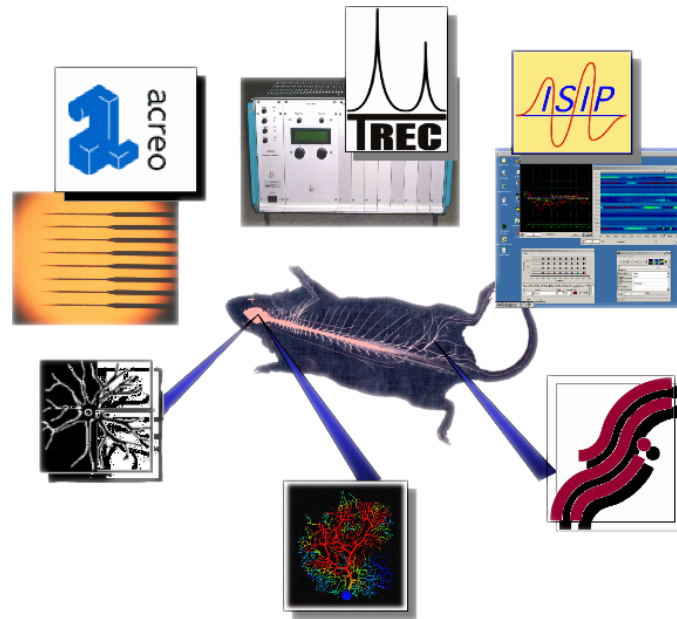


Figure 1.1: Overview of the project VSAMUEL. Clockwise from top left: Custom designed connectors and passive batch fabricated multi site microelectrode probes (Acreo AB, Kista, Sweden), discrete and modular multi-channel amplifier (Thomas RECORDING GmbH, Gießen, Germany), data acquisition system, hard- and software (ISIP, University of Lübeck, Germany), and experimental partners with their area of application, i.e. Peripheral nerves (SMI, Aalborg University, Denmark), Cerebellum (BBF, University of Antwerp, Belgium), and Cortex (INN Carlo Besta, Milano, Italy).

The chapter ‘Neural Activity’ gives an overview of the physiological background of neural activity. The next chapter ‘Neural Signal Acquisition’ is dedicated to the physics and mechanics of sensors for neural recordings. Afterwards the chapters ‘Signal Processing’ and ‘Applying Wavelet Transform to Neural Signals’ give a detailed description of the signal processing methods employed by the data acquisition system. Then the hardware and the software is presented in chapter ‘Data Acquisition System’. The chapter ‘Biological Validation’ describes the actual usage

of the system by the experimental partner of the project. Finally, the discussion reviews the positive and negative results of this work and explains the advantages and limitations of the proposed system.

# Chapter 2

## Neural activity

### 2.1 Nervous system

All kinds of higher animals living on earth have a nervous system. It occurs in a wide range of complexity and, depending on this complexity, it can provide its owner a wide range of services. Simple organisms use their nervous system just to accomplish basic and essential tasks like e.g. finding food or mates for reproduction. With an increasing complexity of the nervous system the tasks it can perform become more complex as well. Humans for instance have a very large nervous system with about  $10^{11}$  (100 billion) neural cells or neurons and additionally 10 to 50 times this number of glial cells [61]. This complex nervous system is the place where thinking, feeling, consciousness and subconsciousness of humans is located.

The two main types of cells occurring in the nervous systems are neural cells and glial cells. The neural cells – also called neurons – perform the information processing, while the glial cells are assumed to mainly have support functions. They sustain the brain structure, transport nutrients from blood to neurons and they form insulation layers, which speed up information transmissions between neurons [61].

The nervous system can be divided in two main parts, i.e. the central nervous system (CNS) and the peripheral nervous system (PNS) [61]. The CNS integrates information from PNS. High level processing tasks like conscious planning are performed based on this information. On the other hand the peripheral nervous system performs low level tasks like information acquisition for scenting, vision, and sensing. Furthermore some preprocessing of acquired sensory data and reflexes are located in this part of the nervous system [61].

An example illustrates the interactions between central and peripheral nervous system. Consider the very essential necessity of gathering food. Usually the feeling of hunger caused by a low concentration of glucose in the blood detected by sensory cells (PNS) gives the impulse to start searching for food. The search for food involves many sites of the nervous system. Complex nervous systems contain a memory (CNS) function which might provide information, where food can be found and how it looks, smells, and feels like. This information is used to get the body closer to a place, where food is available. It requires conscious planning (CNS). Information from peripheral nervous system, e.g. about odors, is compared to information in the memory of CNS until food is detected. The sensory information is used to decide if

the food is actually good or if it is bad. Finally, if any food was found and considered to be good, it is maybe eaten and the glucose concentration in blood will rise again. Even if the connections are simplified in this example, it should give an idea, how CNS and PNS work together.

## 2.2 Neural cell

Information processing, high as well as low level, is done by neural cells or neurons. These cells occur in a great variety with respect to shape and size. The differences depend on the location within the nervous system and on their actual task. Despite their different size and shape neural cells share a common structure, which is established by four morphologically defined regions: cell body, dendrites, axon, and synaptic terminals [61]. In Figure 2.1 a schematic drawing illustrates these regions.

The cell body or soma has a diameter of up to  $50\text{ }\mu\text{m}$ , which can sometimes even be exceeded. The shape of the soma is often ellipsoidal, triangular or pyramidal. In almost the same manner as in other cells the cell body contains a nucleus with DNA, endoplasmatic reticulum, and ribosomes, which produce proteins. In neurons the ribosomes only occur in the cell body. Thus proteins are produced only within the cell body and need to be transported to all other regions of the neuron. Mitochondrions, which produce the energy carrier ATP (Adenosin-Tri-Phosphate), are present in the cell body and also within other regions of the neuron.

Dendrites are processes which emerge from the cell body and form trees with many ramifications. A neuron has usually several dendrites which can differ a lot in size, shape, and grade of ramification (see also Figure 2.2). The neuron collects its input from other neurons through connections which are mainly established within the dendritic trees.

An axon is a process which emerges at a somewhat thickened area of the soma, called the axon hillock, where action potentials are generated (see also Section 2.5). Each neuron has only a single axon which serves as the output line. It has a diameter of  $0.2$  up to  $20\text{ }\mu\text{m}$  and an extend from a few millimeter up to one meter (in peripheral nervous system). Near its end the axon splits up into a couple of branches. At the tips of these branches the synaptic terminals reside establishing connections with other neural cells. Thus, the axon is capable of transmitting an action potential, which is generated under certain conditions at the axon hillock to other neural cells. Some particularly large axons, especially in the PNS are insulated by sheaths of myelin which accelerate the transmission speed significantly. These myelin sheaths are formed by certain glial cells. Along the axon the myelin sheath is regularly interrupted at the nodes of Ranvier.

A synaptic terminal establishes a chemical connection to another neuron, called post synaptic neuron, usually within its dendritic tree. Between the synaptic terminal and the post synaptic cell, there is a small space,  $30$  to  $40\text{ nm}$  wide, which is called the post synaptic cleft. Into this cleft neurotransmitters are released on arrival of an action potential at the synaptic terminal. Neurotransmitters are stored in synaptic vesicles. Energy is provided in form of ATP produced by many mitochondrions present in a synaptic terminal.

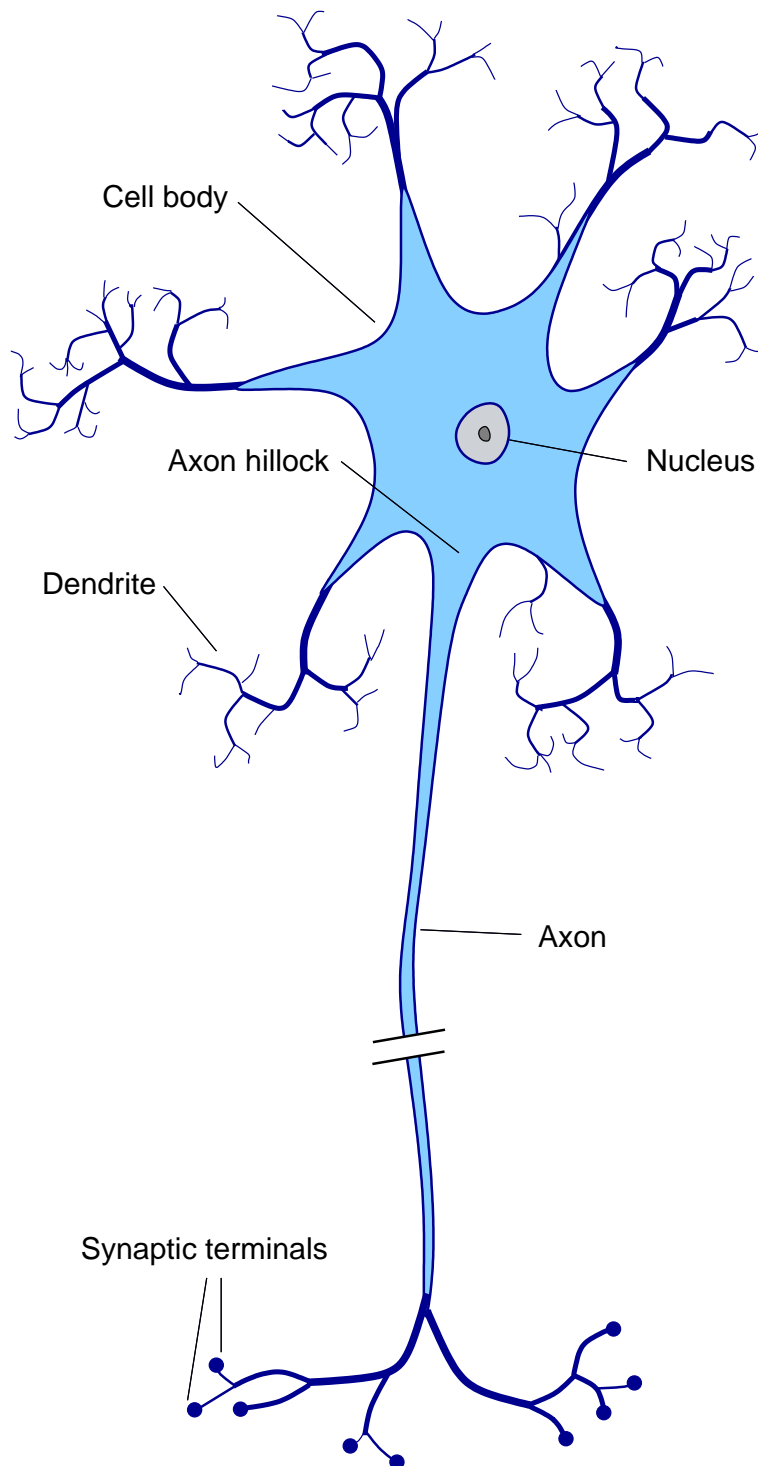


Figure 2.1: Schematic drawing of a basic neuron.

### 2.2.1 Morphological classification

Based on the number of processes arising from their cell body neurons are classified into three large groups: unipolar neurons, bipolar neurons, and multipolar neurons. Figure 2.2 shows instances of these types [61].

Unipolar neurons are characterized by a single process which emerges the cell body and ramifies into several branches as depicted in Figure 2.2(a). One branch serves as axon and the others are dendrites. This cell type is predominant in the nervous system of invertebrates, but they do also occur in certain ganglions of vertebrates' autonomic nervous system.

Bipolar neurons (Figure 2.2(c)) have an ovoid soma with two emerging processes: A dendrite, which conveys information from the periphery, and an axon which carries information toward the central nervous system. Bipolar cells often have sensory functionality, thus they occur e.g. in the retina or in olfactory epithelium. A special kind of bipolar cells can be found in spinal ganglia, where they carry information about pressure, touch, and pain. These cells have initially two processes, but during their development both processes converge and form a single process emerging the soma, which splits into two processes. Both parts work as axons, one runs toward the periphery (skin and muscle) and the other toward the spinal cord. These neurons shown in Figure 2.2(b) are called pseudo-unipolar.

Multipolar neurons are preponderant in the vertebrate nervous system. They have a single axon and one or more dendritic branches, emerging from all parts of the cell body. Great variations with respect to size and shape of cell bodies, and also to number and length of dendrites are found for this neuron type. Three examples which illustrate these variations are shown in Figure 2.2(d). The number of contacts varies with the size of the dendritic trees. A spinal motor cell receives input on about 10,000 contacts (2000 on cell body, 8000 on dendrites) while having a moderate number dendrites with a moderate extent itself. On the other hand the Purkinje cell of cerebellum with a large dendritic tree receives input of approximately 150,000 contacts.

### 2.2.2 Functional classification

Beside the morphological classification neurons can also be classified based on their functionality into three major groups, i.e. afferent, motor, and interneural neurons.

Afferent neurons carry information from the periphery into the nervous system both for conscious perception and for motor coordination. Motor neurons carry commands from nervous system into the periphery. These commands coordinate muscles and glands. Interneurons are all neurons which cannot be classified as afferent or motor neurons. They constitute the largest number of neurons and process information either locally or convey information from different sites within the nervous system.

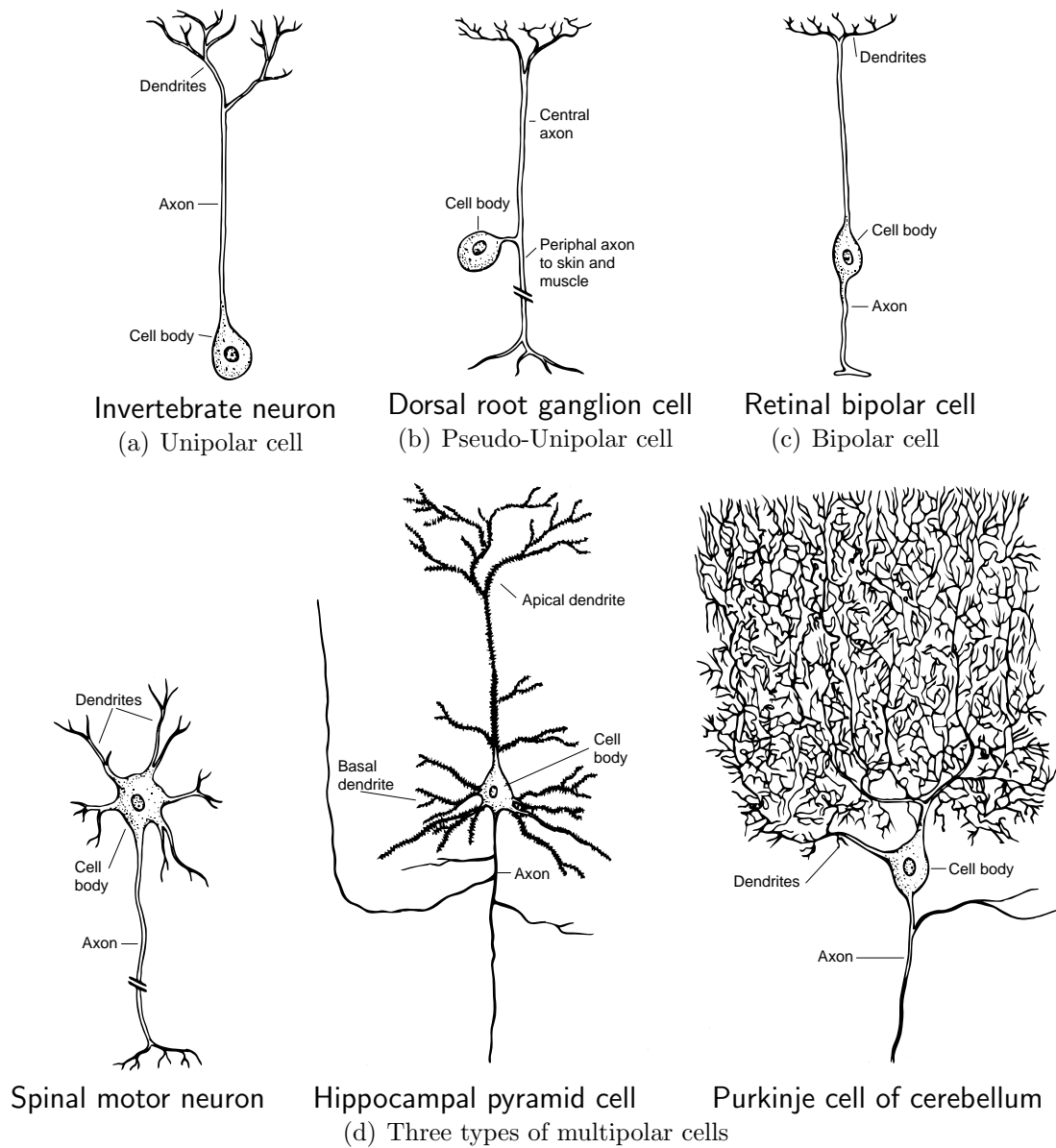


Figure 2.2: Neurons are categorized as unipolar, bipolar, and multipolar depending on the number of processes emerging the cell body, according to [62, 105].

## 2.3 Glial cells

Neurons are surrounded by glial cells, which seem not essential for information processing, but have several other tasks [61]. They support the brain structure and insulate groups of neurons from each other. Two types of glial cells form insulating myelin sheaths around large axons: Oligodendrocytes in CNS and Schwann cells in peripheral nervous system. These sheaths significantly speed up transmission of action potentials through long axons. Some glial cells act as scavengers, removing debris after an injury or neural death. Another task is the buffering of  $K^+$  ion concentration in extracellular space and to take up and remove chemical transmitters released by neurons during synaptic transmission. Glial cells guide migrating neurons during brain development and direct the outgrowth of axons. They form the blood-brain barrier and they might have nutritive functions.

## 2.4 Layered structure of the brain

The brain has a laminar structure. Number, size and also functional tasks of these layers depend on the location within the brain. The cerebral cortex for instance is organized into six distinct layers. In Figure 2.3 a schematic illustration shows these layers, starting at the outmost layer 1, which lies just below the pia mater and ending with layer 6 which lies just above the white matter. Layer 1 is made up mostly of glial cells and axons that run laterally through the layer and contains few cell bodies. Layer 2-6 contain different proportions of the two main classes of cortical neurons, pyramidal and nonpyramidal cells. Pyramidal cells send long axons down the spinal cord and are the major output neurons. They also have axonal branches that terminate in the local area. The axons of most nonpyramidal cells terminate locally (page 292, [62]).

The laminar structure of the cerebellum is different from that of the cerebral cortex as illustrated in the schematic shown in Figure 2.4. Basically, the cerebellum cortex has four layers that contain only five types of neurons: stellate, basket, Purkinje, Golgi, and granule cells [37]. Going from outermost to innermost, there are the molecular layer, the purkinje cell layer, granular layer, and medullary layer.

In order to gain more knowledge about interactions between these layers, simultaneous recordings from all layers are necessary. Therefore, the probes have different layouts to account for the different target regions within the brain. The layouts differ in shaft length, shaft count, and recording site spacing.

## 2.5 Action potentials

Neurons transmit information by means of action potentials which are generated in the soma at the axon hillock under certain conditions and then transported along axons to other neurons. An action potential is generated when the membrane potential has been depolarized above a threshold, which is specific for each cell. This section covers the mechanisms which are involved in the generation of action potentials.



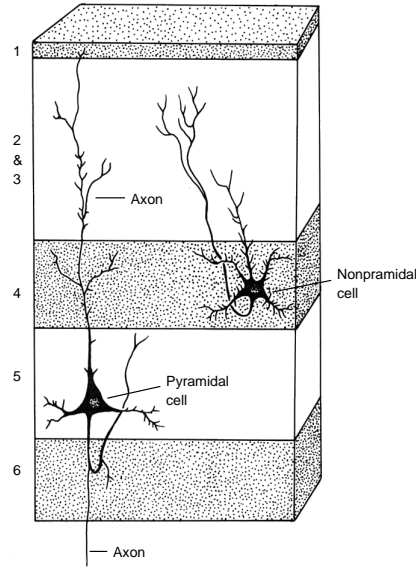


Figure 2.3: Layered structure of cerebral cortex, adapted from [62]

### 2.5.1 Membrane potential

Across the extremely thin membrane of cells a potential difference exists which is due to different concentrations of ions in extracellular and intracellular space [64]. Four different ions are mainly involved:  $K^+$ ,  $Na^+$ ,  $Cl^-$ , and  $A^-$  (organic anions). These ions except for the organic ions can pass the cell membrane through specific active or passive ion channels. Active ion channels consume energy (ATP) to transport ions across the membrane usually against an electrical or chemical gradient. On the other hand, ions cross the membrane through passive ion channels along an electrical or chemical gradient. These channels can be either gated or non gated. Gated channels change their flow rate for ions depending on the actual potential across the membrane, while non gated channels have a fixed flow rate. The permeability of the membrane for a certain ion depends on the number of channels for that ion, as well as on the flow rate of these channels. By means of active transportation and also passive flux of ions the membrane potential is established.

First, we consider the diffusion of  $K^+$  ions. The concentration of potassium ions is high inside the cell and low outside the cell and the cell membrane is assumed to be permeable for potassium through passive ion channels.

Therefore,  $K^+$  ions leave the cell along the concentration gradient, which imposes a chemical force on the  $K^+$  ions. The organic ions cannot pass through the cell membrane and are left behind. This increases the potential gradient across the cell membrane, which also affects the  $K^+$  ions pulling them back into the cell. If both, the electrical and the chemical force, compensate each other, the  $K^+$ -equilibrium has been reached, which is described by the Nernst equation:

$$E_K = \frac{RT}{ZF} \ln \frac{[K^+]_o}{[K^+]_i}$$

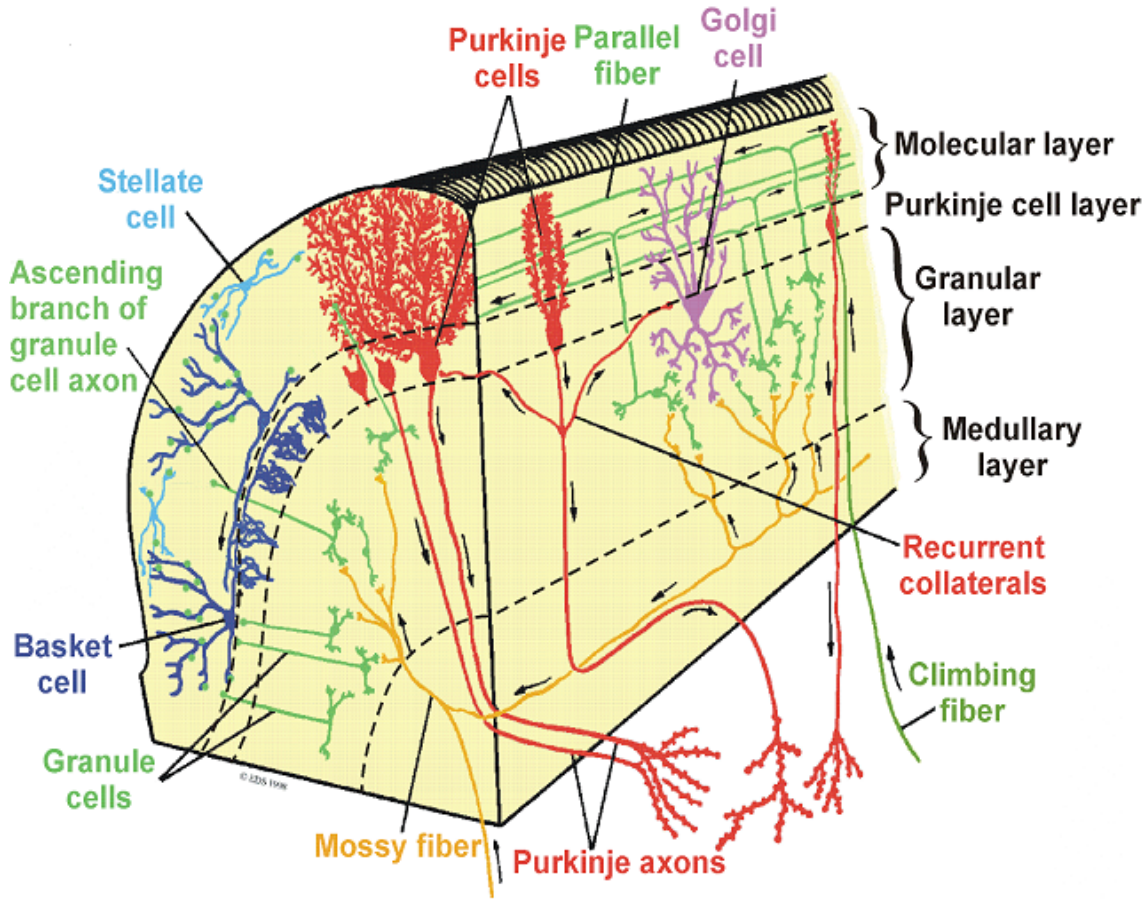


Figure 2.4: Layered structure of the cerebellum, in a schematic drawing of cerebellum [90].

where  $R$  is the Gas Constant,  $T$  is temperature in degrees Kelvin,  $Z$  is the valence of  $K^+$ ,  $F$  is Faraday constant, and  $[K^+]_o$ ,  $[K^+]_i$  are the concentrations of  $K^+$  outside and inside a cell. Assuming the concentrations of squid giant axon (see Table 2.1) the steady state membrane potential for  $K^+$  becomes

$$E_K = 26\text{mV} \times \ln \frac{20}{400} = -79\text{mV}$$

where temperature is set to  $25^\circ\text{C}$  and the valence of  $K^+$  is  $Z = +1$  resulting in  $RT/ZF = 26\text{mV}$ .

The membrane of neurons is also permeable for  $Na^+$  ions through passive ion channels. However, there are only a few  $Na^+$  channels compared to the number of  $K^+$  channels. Thus, permeability for  $Na^+$  ions is lower than for  $K^+$  ions. The concentration of  $Na^+$  ions outside the cell is higher than inside the cell. Thus  $Na^+$  ions are affected by two forces, a chemical along the concentration gradient, and an electrical along the potential gradient, both driving the  $Na^+$  ions into the cell. The equilibrium of  $Na^+$  is calculated with Nernst equation as

$$E_{Na} = \frac{RT}{ZF} \ln \frac{[Na^+]_o}{[Na^+]_i},$$

Ion	Cytoplasm (mM)	Extracellular fluid (mM)	Nernst potential (mV)
K <sup>+</sup>	400	20	-79
Na <sup>+</sup>	50	440	+55
Cl <sup>-</sup>	52	560	-60
A <sup>-</sup>	385	—	—

Table 2.1: Distribution of the major ions across the membrane of the squid giant axon [64].

i.e. for the squid giant axon about +55 mV given the concentrations from Table 2.1. Na<sup>+</sup> is about 130 mV away from its equilibrium at the K<sup>+</sup> Nernst potential of -79 mV. Thus, the electrochemical force which drives Na<sup>+</sup> into the cell is quite strong. The influx of Na<sup>+</sup> into the cell is compensated by an efflux of K<sup>+</sup> out of the cell. Note, that the permeability of the membrane to Na<sup>+</sup> is much lower than the permeability to K<sup>+</sup>. Therefore the balance point shifts only slightly away from the membrane potential established only by K<sup>+</sup> ions. But a constant influx of Na<sup>+</sup> and outflux of K<sup>+</sup> cannot persist forever. After a while the concentration gradients of K<sup>+</sup> and Na<sup>+</sup> would be flat, and the membrane potential would run low. This is prevented by an Na<sup>+</sup>-K<sup>+</sup> pump, i.e. an active channel, which extrudes Na<sup>+</sup> from the cell while taking in K<sup>+</sup>. Both types of ions are moved against their net electrochemical gradients and, thus, this process needs energy, which is provided by hydrolysis of ATP.

All nerve cells have non-gated Cl<sup>-</sup> ion channels. However, since Cl<sup>-</sup> is in equilibrium at a potential of -61 mV we can ignore the contribution of Cl<sup>-</sup> ions to the resting membrane potential of nerve cells, since they do not change much.

The actual resting membrane potential depends on the interaction of two or more types of ions and is described by the Goldman equation:

$$V_m = \frac{RT}{F} \ln \frac{P_K[K^+]_o + P_{Na}[Na^+]_o + P_{Cl}[Cl^-]_o}{P_K[K^+]_i + P_{Na}[Na^+]_i + P_{Cl}[Cl^-]_i}. \quad (2.1)$$

Both permeability of the membrane for the specific ion and concentration inside and outside the cell influence the resting membrane potential.

### 2.5.2 Action potential generation

Action potentials are generated by a change of permeability of the membrane (Hodgkin-Huxley-Model) [64]. The initial condition for an action potential is a depolarization of the membrane potential e.g. by an excitatory synaptic potential. This depolarization causes some voltage gated sodium channels to open and the influx of Na<sup>+</sup> into the cell driven by chemical and electrical forces increases. This in turn causes a further depolarization and again more voltage gated Na<sup>+</sup> channels open. The influx of Na<sup>+</sup> accelerates again. This regenerative, positive feedback cycle develops explosively, and it leads the membrane potential toward the Na<sup>+</sup> equilibrium potential of +55 mV. However, this potential is never reached, because of the continued K<sup>+</sup> efflux through the K<sup>+</sup> channels. Two mechanisms repolarize the membrane

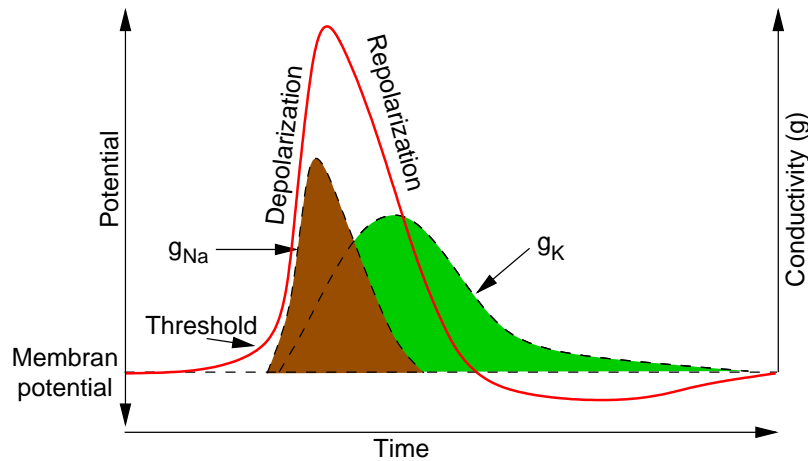


Figure 2.5: Scheme of an action potential. The conductivity of the membrane for  $\text{Na}^+$  and  $\text{K}^+$  is denoted by  $g_{\text{Na}}$  and  $g_{\text{K}}$ , respectively. The conductivities change over time triggered by an initial depolarization which must exceed a certain threshold.

and terminate the action potential. First, the continued depolarization slowly turns off or inactivates the voltage-gated  $\text{Na}^+$  channels. Answering to a depolarization the channels open fast, but a continued depolarization slowly closes the channels. Second, with a delay voltage-gated  $\text{K}^+$  channels open which increase permeability to  $\text{K}^+$  and therefore the efflux of  $\text{K}^+$ . Figure 2.5 shows changes of the membrane potential during an action potential.

Note, that the action potential generation follows an “All or Nothing” rule [36]. If the depolarization has exceeded a certain threshold, a full-fledged action potential is generated. The grade of exceedance does not change height, length, or shape of the action potential, at least if other parameters remain constant. After generation of an action potential the cell can generate the next action potential after the absolute refractory period has been expired, which is about 1 ms.

## 2.6 Field potentials

Field potentials or local field potentials (LFP) reflect almost exclusively synaptic potentials and synchronous subthreshold membrane deflections [65]. They do not seem to reflect synchronous firing of action potentials, as even similar shaped LFP show different temporal firing patterns of neurons, which is reported in [49] and illustrated in Figure 2.6. There is no clear pattern in the recorded single unit behaviour. A unit recording alone may give a wrong idea about the involvement of the cell that fires (or does not fire) in the population activity. In order to get a better understanding of the interaction between field potentials and action potentials it is necessary to study both together. This implies that the data acquisition system is able to record field and action potentials simultaneously, i.e. it must have a high bandwidth and a high analog digital conversion resolution, because the frequency of field potentials ranges from 0.5 Hz up to 80 Hz while their amplitude is a factor of

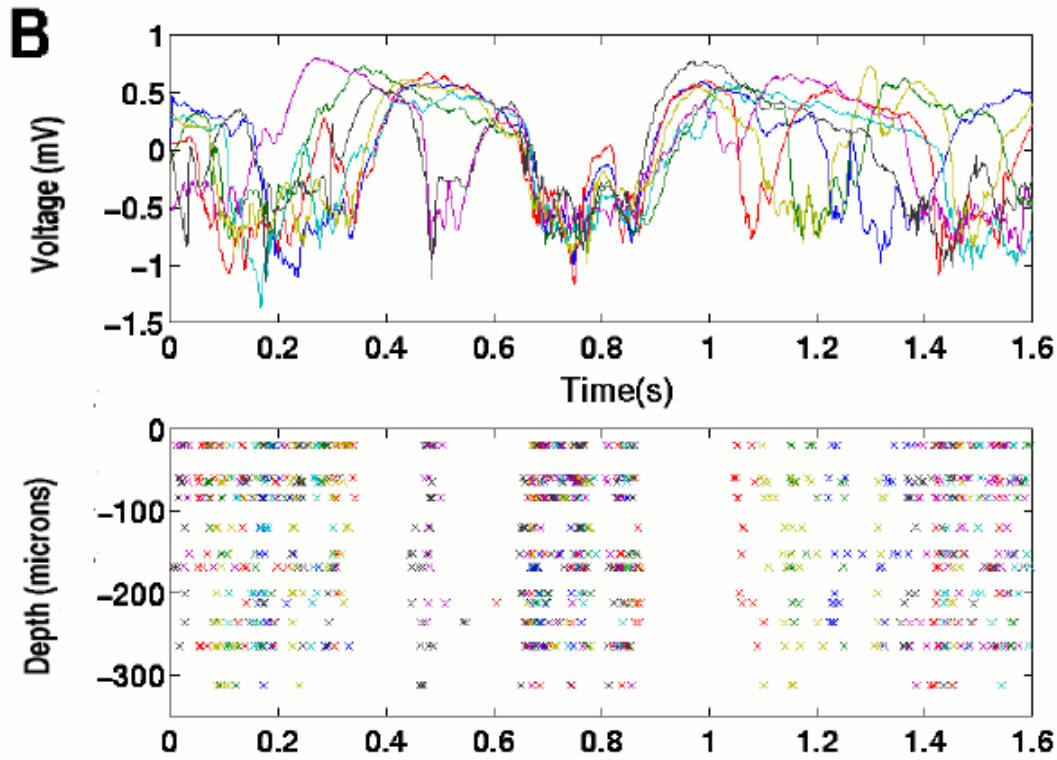


Figure 2.6: Connection between field and action potentials: A set of seven similar local field potential waveforms from a recording of spontaneous activity in the somatosensory cortex of a rat is shown at the top. At the bottom the corresponding single unit activity of 12 neurons is displayed. The color of each mark indicates in which waveform the action potential was detected. While the single units only fire during troughs of field potentials, there is a high variability with respect to which unit actually fires in a trough and at which time it fires [49].

10 to 20 times larger than the amplitude of action potentials.

## 2.7 Tuning curves and information coding

Information is represented in the nervous system by means of action potentials – also called spikes – presumably with different codes. Since it seems that shape and duration of action potentials do not encode information, it must be the time of firing which is important. Given the time of firing we can derive different features as there are for example [78]:

- Temporal patterns
- Number of spikes in the train
- Variance of interspike intervals

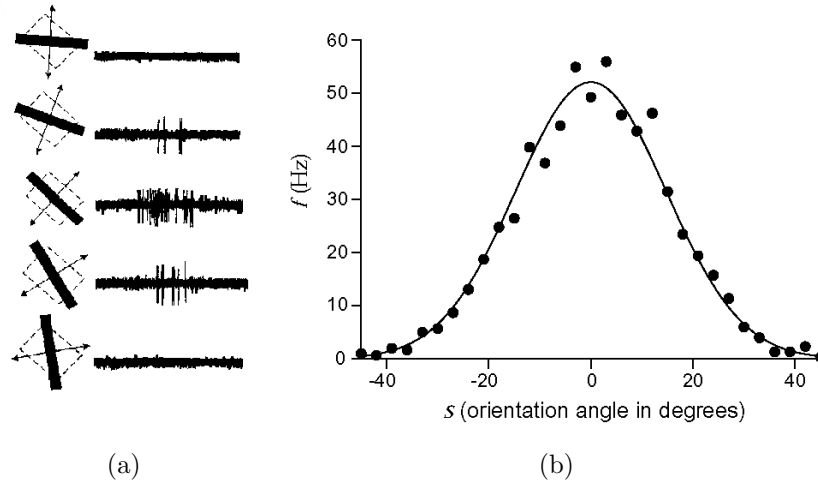


Figure 2.7: Tuning curve. (a) Extracellular recordings of a neuron in the primary visual cortex of a monkey which were recorded during presentation of moving light source with different orientations. The arrows indicate motion and the angle the orientation which corresponds to recording on the right. (b) Average firing rate of a cat V1 neuron plotted as a function of the orientation angle of a light bar stimulus. (Source: (a) from [51] and (b) data points from [42], both adapted from [15].)

- Spike delays or latencies
- Constellation code

Of particular importance is the constellation code, where we presume that information is encoded by simultaneous activation of certain groups of neurons.

The actual code, which encodes a particular parameter, is often unknown. In some cases it can be determined experimentally how a certain parameter is encoded in the nervous system. This is tested by imposing a stimulus, which is considered to be “normal” and which can be varied with respect to a certain parameter. The observed response is the code. Now, the input stimulus is changed with respect to one parameter. Hereby, a so called tuning curve [15] can be employed which shows the average firing rate of a neuron over several trials versus the parameter of the stimulus presented during the trials. By fitting an appropriate parametric model function to these data points we obtain a model of how the stimulus is represented by the examined neuron. However, only if measurements are performed in the same preparation and only the one parameter has changed, we can be rather sure that we found the actual code employed. Figure 2.7 shows an example for a tuning curve. Recordings are made here while a light bar moves across a receptive field at different orientations. An intercepted neuron produces different firing rates which depend on the orientation of the light bar and which can be modeled by a Gaussian tuning curve.

Finding experimental evidence for certain coding methods of the brain requires recording from as many active neural cells as possible. These multi neuron activities are then inspected for possible signs of interactions among neurons, conventionally

by use of cross-correlation techniques, usually applied to pairs or triplets of neurons [35]. General techniques for measuring neural activity and the particular design of the VSAMUEL probes are described in the next chapter.

# Chapter 3

## Neural signal acquisition

The neural activity within a living brain results in different interceptable signals. These can be measured by various means, which differ with respect to spatial resolution, time resolution, and whether they are invasive or non-invasive.

Electroencephalogram (EEG) is a non-invasive method which has a time resolution of up to 70 Hz and a rather low spatial resolution. Electrodes with diameters ranging from 0.4 cm to 1 cm are attached to the scalp with special pastes and caps or nets which hold them in place. In standard clinical practice an EEG is recorded with about 20 electrodes arranged using the International 10-20 System [58]. However, for research purposes also settings with about 100 or even more electrodes are used. One electrode records the electric field potentials generated in tissue that contains approximately 10 million to one billion neurons [85]. Therefore, this method gives an overview of neural activity in the complete brain. Despite the low spatial resolution (about 1 cm<sup>3</sup>) patterns which can be found in EEG recordings indicate a certain state of consciousness. For example the state of deep sleep is linked with large amplitude oscillations in the so-called *delta* band (about 1-4 Hz). Furthermore, by analyzing EEG data it is possible to identify distinct sleep stage, depth of anesthesia, or epileptic seizures. EEG is also used for the determination of cerebral death [50].

The major limitation for recording EEG signals is the distance from electrode on the scalp to the underlying neural tissue which ranges between 2 and 3 cm [91]. This barrier which consists of skin, bone, and liquor acts like a low-pass filter. An invasive variant of the EEG, which avoids this limitation, is the Electrocorticogram (ECoG) where recording electrodes are placed directly onto the cortex. This method achieves a much finer spatial resolution on the order of 1 mm and also allows the recording of higher-frequency content (10-200 Hz) [91]. Electrodes with about 4 mm diameter arranged in grids or strips on a foil with a spacing of 1 cm between the electrodes are implanted directly onto the cortex. An Electrocorticogram is for example recorded from epileptic patients in order to determine the cortical region that generates epileptic seizures [66].

Local field potentials (LFPs) are obtained as the low frequency (up to about 300 Hz without DC) component of extracellular voltage measurement with micro electrodes inserted into the cortex. There are various types of micro electrodes including e.g. micro wires, micro pipettes, or fork shaped silicon. Their recording



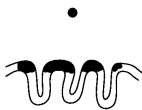
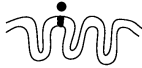



Electroencephalogram (EEG)	3 cm	
Electrocorticogram (ECoG)	0.5 cm	
Local Field Potential (LFP)	1-3 mm	
Multi Unit Action Potential (MUA)	0.4 mm	
Single Unit Action Potential (SUA)	0.2 mm	

Figure 3.1: Comparison of different modalities to record neural activity with respect to their spatial resolution [91, 65]. The electrodes are depicted by a black circle. Typically an EEG electrode (black circle) is located about 2-3 cm above the cortex where it averages gyral neural activity across a 3 cm spatial extent. This is indicated by the filled black cortical layers. ECoG electrodes are placed directly onto the cortex and average neural activity over a smaller 0.5 cm range. Local field potentials, multi-unit and single-unit action potentials are recorded from within the brain parenchyma and sample even smaller areas of neural tissue, yielding higher spatial resolutions.

site size is in the range of 10 by 10  $\mu\text{m}$ . More details are given in Section 3.3 and Section 3.4. It appears that LFPs reflect the combined activity of large numbers of neurons within a large region of the cortex [65]. At this, the main component of LFPs is associated with synchronized dendritic activity. Therefore, LFPs provide a measurement of the input signals of a neural population within a few millimeters from the recording site [70].

Multi unit and single unit action potential signals can be recorded with the same type of micro electrodes as LFPs. They are recorded with a high temporal (up to 50 kHz) and a high spatial ( $50 \mu\text{m}^3$ ) resolution. Only activity from a very small region of the brain is recorded. It depends to some degree on the impedance of a recording site but mainly on its relative position to single or multiple neurons and their axons if single or multi unit action potentials are recorded. Single unit activity can be derived from multi unit activity by spike sorting, which classifies action potential with respect to their shape. Action potentials can be considered as output of a cortical region. So the combined recording of LFPs and MUA/SUA is likely to help in understanding how the brain works, as it allows to apply a black box analysis to particular regions with given input and output.

Figure 3.1 shows a comparison of different modalities for recording neural activity. One goal of the work described in this thesis was to enlarge the region or the number of neural cells from which activity can be recorded using invasive methods.

Iron	-440 mV
Lead	-126 mV
Copper	+337 mV
Platinum	+1190 mV

Table 3.1: Electrode potentials for some metals

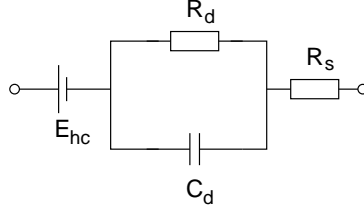


Figure 3.2: Equivalent circuit for a biopotential electrode.  $E_{hc}$  is the half-cell potential.  $R_d$  and  $C_d$  represent impedance and polarization at the electrode-electrolyte interface, respectively. And  $R_s$  is associated with interfacial effects and the resistances of the electrode materials.

The following paragraphs describe basic concepts and different methods for invasive recording of neural activity.

### 3.1 Sensing bioelectrical signals

Electric conductivity in the body is based on ions as charge carriers. Probes need to transform ionic currents into electric currents which are required by wired and electronic instrumentation. The transduction is usually implemented by electric conductors in contact with aqueous ionic solution of the body [8, 6].

At the contact region between metal and a solution of ions of these metal a local change of concentration of ions occurs. Therefore, the charge neutrality is not maintained in this region, i.e. the electrolyte surrounding the metal is at a different electrical potential as the rest of the solution, called the *half-cell potential*. Different metals establish different half-cell potentials as shown in Table 3.1 [9]. Note, that the potentials are very much larger than electrophysiological signals.

Electric characteristics of biopotential electrodes are generally nonlinear and a function of current density at their surface. Nevertheless, for low potentials and currents the circuit given in Figure 3.2 describes the behaviour of a biopotential electrode.  $R_d$  and  $C_d$  are components that represent the impedance associated with the electrode-electrolyte interface and polarization at this interface.  $R_s$  is the series resistance associated with interfacial effects and the resistance of the electrode materials themselves. The battery  $E_{hc}$  represents the half-cell potential. Note, that the impedance of the electrode will depend on the frequency. In particular, for low frequencies the impedance is dominated by series combination of  $R_s$  and  $R_d$  and for high frequencies  $C_d$  bypasses the effect of  $R_d$ , so that the impedance is close to  $R_s$ . Measuring the impedance at low and high frequencies enables the determination

of the parameters of the components of the circuit. However, electrode impedance is affected by a change in surface area due to roughness and radius of curvature. Furthermore the impedance is affected by contamination of the electrode surface.

## 3.2 Extracellular potentials

The generation of extracellular potentials can be completely derived from well known biophysical principles of neural excitability [84, 48]. Ionic currents which come along with action potentials and subthreshold activity flow in closed loops. They enter the neuron at current sink regions, propagate through the cytosolic medium, leave it at current source regions, and return to the sink points through the extracellular space. Spatio-temporal patterns of extracellular voltages appear as a result of this return current [12].

Figure 3.3 shows an extracellular recording setup with a single microelectrode. The electrode is moved toward the axon until the gap resistance increases to about 20 k $\Omega$ . Then the electrode in an in vitro setup is placed about 5  $\mu\text{m}$  away from the axon, as determined by visual inspection in [12]. With this gap resistance it is possible to obtain recordings from the soma. Note, that an appropriate resistance between reference electrode and microelectrode is crucial to obtain recordings. This corroborates the evidence for extracellular current pathways. However, if measurements are performed with multisite electrodes, it is impossible to place each single recording site as close as 5  $\mu\text{m}$  to the soma or axon of a neural cell. But the reduced extracellular space, which is due to the compact arrangement of neurons and glial cells, increases the effective resistance through which extracellular currents circulate. Thus, potential differences in the order of 100  $\mu\text{V}$  can be detected, even without placing the recordings sites very close to neurons [12]. Furthermore, the high density of neurons also maximizes the probability of an electrode being quite close to a sufficiently intense current source or sink.

## 3.3 Probes for intracavitary or intratissue recordings

Various probes have been developed based on different technologies and for different applications. Figure 3.4 shows probes with a single tip. These probes are used to record activity at a single point, e.g. the activity of a single neuron either intracellular or extracellular. The metal probe with a single fiber has been enhanced such that it contains multiple fibers. These probes are called tetrode or heptode depending on the number of fibers being either four or seven, respectively. Essentially, these probes record also from a single point, but with multiple recording sites which are a short distance – about 40  $\mu\text{m}$  – away from each other. Figure 3.4(d) shows the tip of a tetrode (see <http://www.thomasrecording.de>). The core fibers are made of platinum-tungsten alloy and the isolation material is quartz glass. Such probes have a diameter of about 100  $\mu\text{m}$ , and the core fibers width is between 7 and 10  $\mu\text{m}$ . Impedance of these electrodes ranges between 0.5 and 6 M $\Omega$ , depending

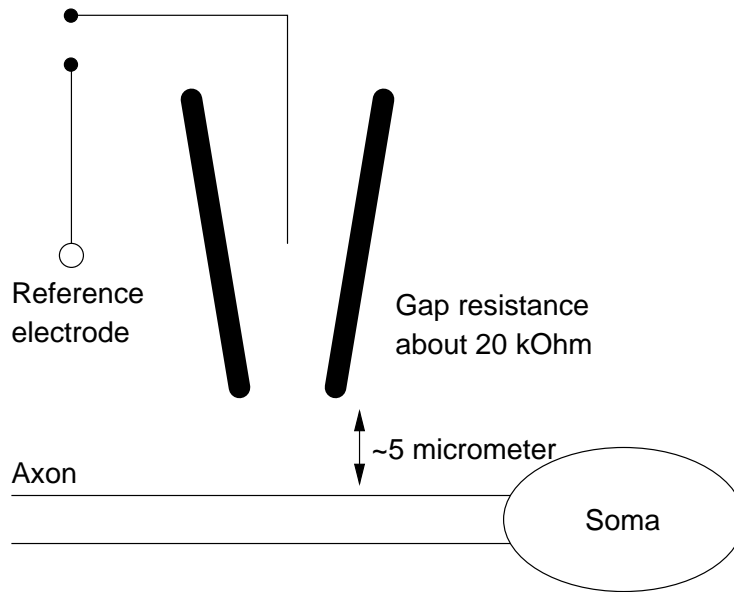


Figure 3.3: Extracellular recording and gap resistance

on the manufacturing process, in particular the grinding angle which ranges from  $15^\circ$  to  $17^\circ$ . The advantage of multiple recording sites close to each other is, that the performance of analysis tasks like e.g. spike sorting (see Section 5.4) can be increased [40].

Another class of probes has recording sites which are arranged as a two dimensional array. These probes differ in size, shape, and material and also in field of application. One example shown in Figure 3.5(a) is an array of microelectrodes developed at Moran Laboratories in the Department of Bioengineering at the University of Utah. 100 silicon spikes, each 1.5 mm long, are arranged in a regular grid on a 4 mm by 4 mm base. Each spike has one recording site and the array is used to record from cortex. Another example are the probes developed and produced by Center for Neural Communication Technology at University of Michigan as shown in Figure 3.6. They consist of fork shaped silicon with recording sites linearly arranged on the shafts. Number and length of shafts and also spacing and arrangement of recording sites can vary. All aforementioned probes are used to record from in vivo or in vitro preparations of nervous tissue. Another technique which requires a special type of probes is the recording from cultured neuron which grow in dish containing transparent conductors and recording sites (indium-tin oxide) on a glass substrate. This kind of probe is called multielectrode array (MEA) and Figure 3.5(b) shows an example MEA with 61 recording sites forming a hexagonal array (see <http://www.neuro.gatech.edu/potter/netinadish.html>).

After this overview the next paragraphs will describe the microelectrodes developed within the project VSAMUEL in detail.

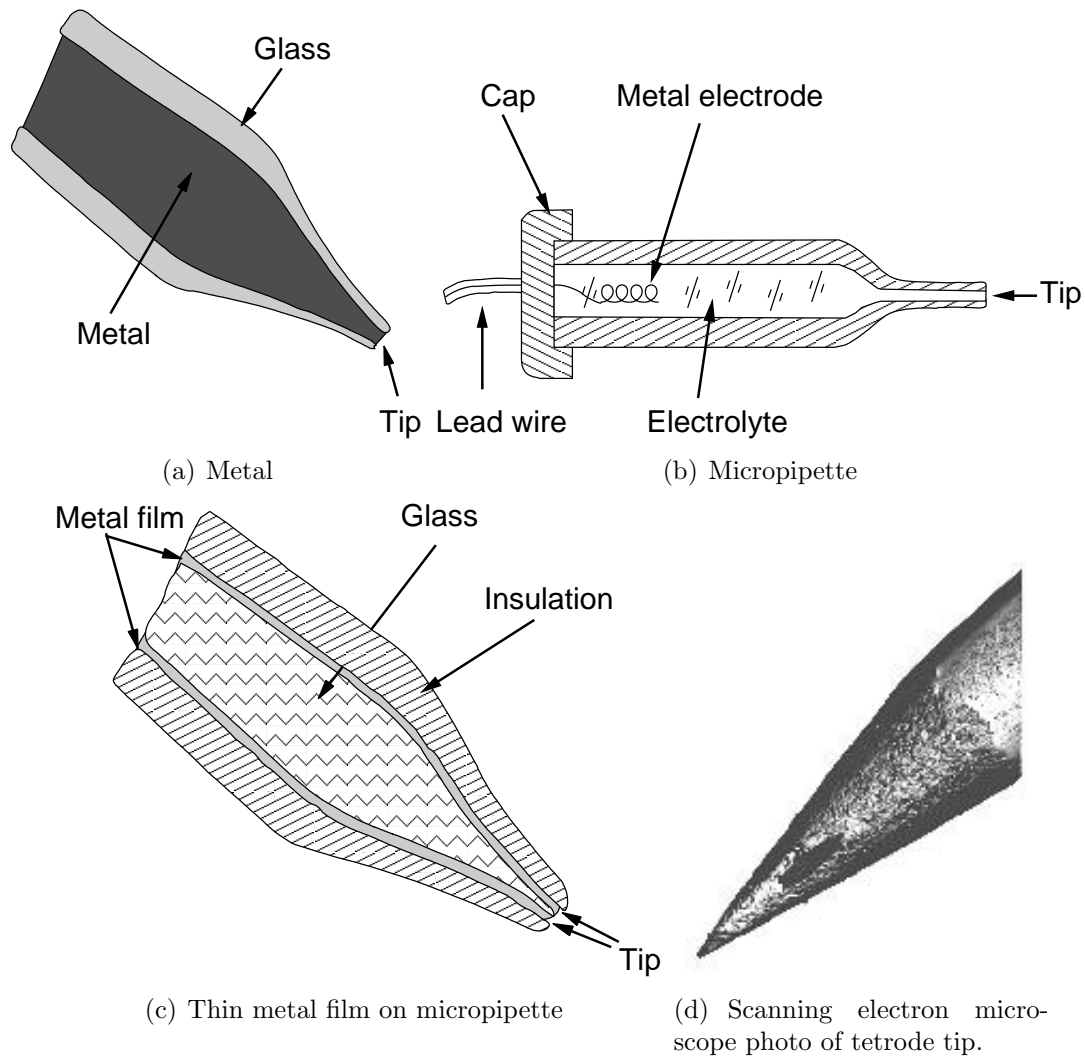


Figure 3.4: Microelectrodes with single tip for intracellular and extracellular recording.

### 3.4 Silicon-based multichannel microelectrodes

Multichannel microelectrodes are created to obtain simultaneous observations of the activity of a large number of neural cells. Therefore, it is desired to have a large number of recording sites inserted in the neural tissue, while the insertion should cause the smallest possible tissue damage. The design and construction of the microelectrodes was done by ACREO AB (Sweden) with respect to these constraints. The probes were developed in two major steps, i.e. first a set of probes with 32 recording sites, then as second step a set of probes with 64 recording sites. The designs vary with respect to the number and length of shafts and their spacing, as well as different distribution of the recording sites at the frontal end of the shafts.

In general the probes have a thin, narrow and pointed silicon shafts for penetration and insertion into the neural tissue. The shafts thickness varies between  $20\text{ }\mu\text{m}$  and  $30\text{ }\mu\text{m}$  depending on the wafer specification. Their platinum or iridium

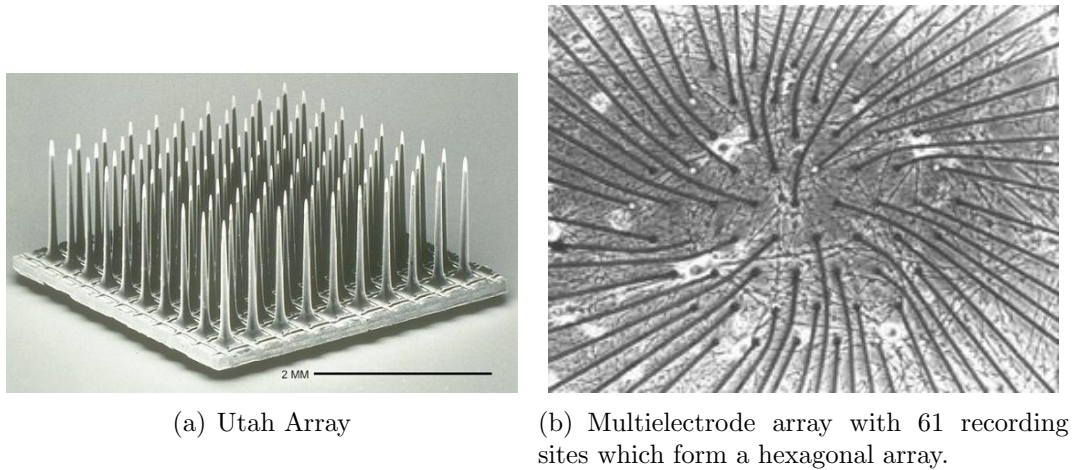


Figure 3.5: MEA probe and needle array.

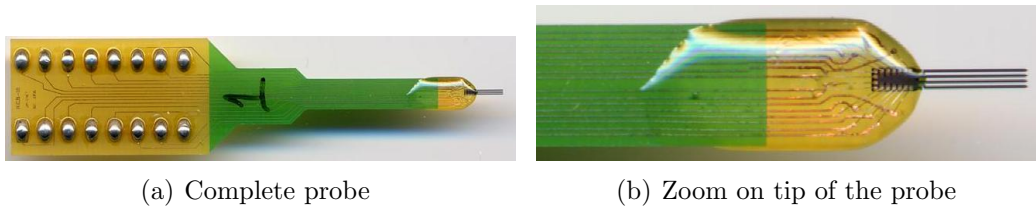


Figure 3.6: Michigan probe: (a) On the left is the connection to the headstage implemented by pins on the backside of the probe. Only the solder points are visible. On the right the silicon probe is mounted using transparent glue which insulates and protects the wire bonded connection between contact pads and the silicon tip. (b) The enlarged tip shows the four shafts of the silicon probe.

recording sites with a usual size of 10 by 10  $\mu\text{m}$  are distributed over the frontal end of the shafts. Fine and narrowly spaced golden conductor traces connect these recording sites with bond pads that reside on a relatively thick silicon base plate.

Our microelectrodes are of fork shaped silicon with shafts of 25  $\mu\text{m}$  width having a pointed end. Recording sites (Ir or Pt, 10  $\mu\text{m} \times 10 \mu\text{m}$ ) are arranged in a two dimensional array (Figure 3.8) at the front end of the shafts. They are connected with contact pads (Au) by fine and narrowly spaced metal (Au) conductor traces. The spacing is either 1.5  $\mu\text{m}$  with 1.5  $\mu\text{m}$  conductor traces, or 0.5  $\mu\text{m}$  with 0.5  $\mu\text{m}$  conductor traces. There are several layout designs, which differ in shaft length (4 mm to 15 mm) and shaft count (1 to 8), and also in recording site spacing (see also [83, 47, 106]).

### 3.5 Manufacturing VSAMUEL probes

Starting with a substrate sequences of metal or photoresist deposition, patterning, and etching are used, to create small structures on the substrate. Even the substrate

Probe	Site Count	Region	Application		Reference
			intra	extra	
Capillary electrodes	1	single cells	✓	✓	[8]
Tungsten wire	1	single cells	✓	✓	[89]
Stereotrode, Tetrode, Heptode	2, 4, 7	multiple cells	–	✓	[89]
Needle array	ca. 100	many cells $4 \times 4$ mm	–	✓	[10]
Flat array	ca. 100		–	✓	[38]
Fork shaped silicon	32, 64, 96	$2 \times 2$ mm	–	✓	[74]

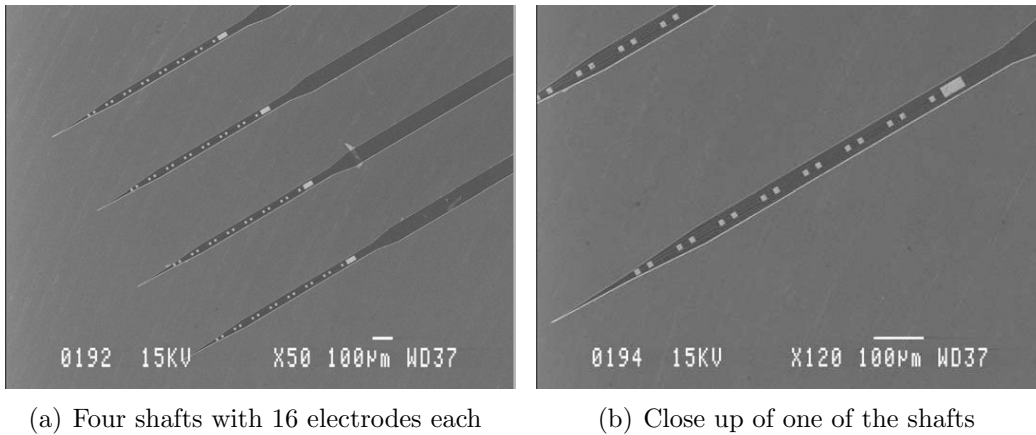


Figure 3.7: Scanning electron micrographs of a 64-site neural probe, model E4

itself can be shaped. The manufacturing of the VSAMUEL probes incorporates several microstructuring techniques. Some of the important techniques used are described in the Appendix. An overview of the manufacturing process is given by Figure 3.9. According to [83] the single steps are:

- (1) Probes are manufactured on silicon-on-insulator (SOI) substrate. Buried inside the substrate is an oxide layer which is supposed to stop the two deep etching processes, which are applied as the final process steps (Step 7 and 8).
- (2) A first isolation layer is deposited, either in form of a thermally grown oxide film or alternatively as a PECVD (Plasma Enhanced Chemical Vapor Deposition) silicon nitride film. Afterwards the conductor traces which consist of Ti or Au are created on top of this layer (e-beam evaporation, patterned with a photoresist lift-off process).
- (3) The conductor traces are covered by another silicon nitride layer, which in turn is opened down to the Au/Ti layer by reactive ion etching (RIE) through a resist mask. At these openings electrode sites and contact pads will arise later in the process.
- (4) Electrode sites are formed with Ti/Pt (e-beam evaporation, patterned with a photoresist lift-off process)

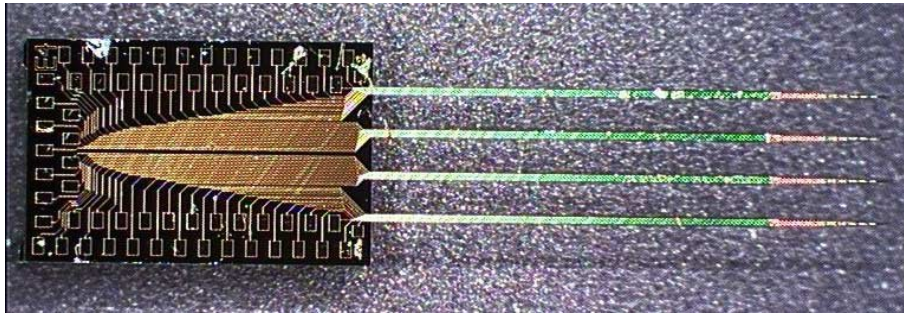


Figure 3.8: VSAMUEL Probes

- (5) The whole structure is protected by a final silicon nitride layer. At the contact pads and at the electrode sites windows are opened using RIE. The nitride covers the edges of the metal patterns for increased reliability in wet working environments.
- (6) The probe structures are protected by a resist mask while the remaining nitride layer is etched using RIE. Afterwards, the top silicon layer is etched 20 to 30  $\mu\text{m}$  down to the buried oxide layer in an inductively coupled plasma deep reactive ion etching process (ICP DRIE).
- (7) In order to protect the resulting structures during the back side etch a thick resist layer is spun on the wafer front side. A thick resist layer is also deposited on the wafer back side. This resist is patterned using double-sided alignment and then the silicon layer is etched full 525  $\mu\text{m}$  down to the buried oxide using an ICP DRIE.
- (8) Finally, shafts are released by etching the oxide in RIE or HF and by stripping away the thick resist layer.



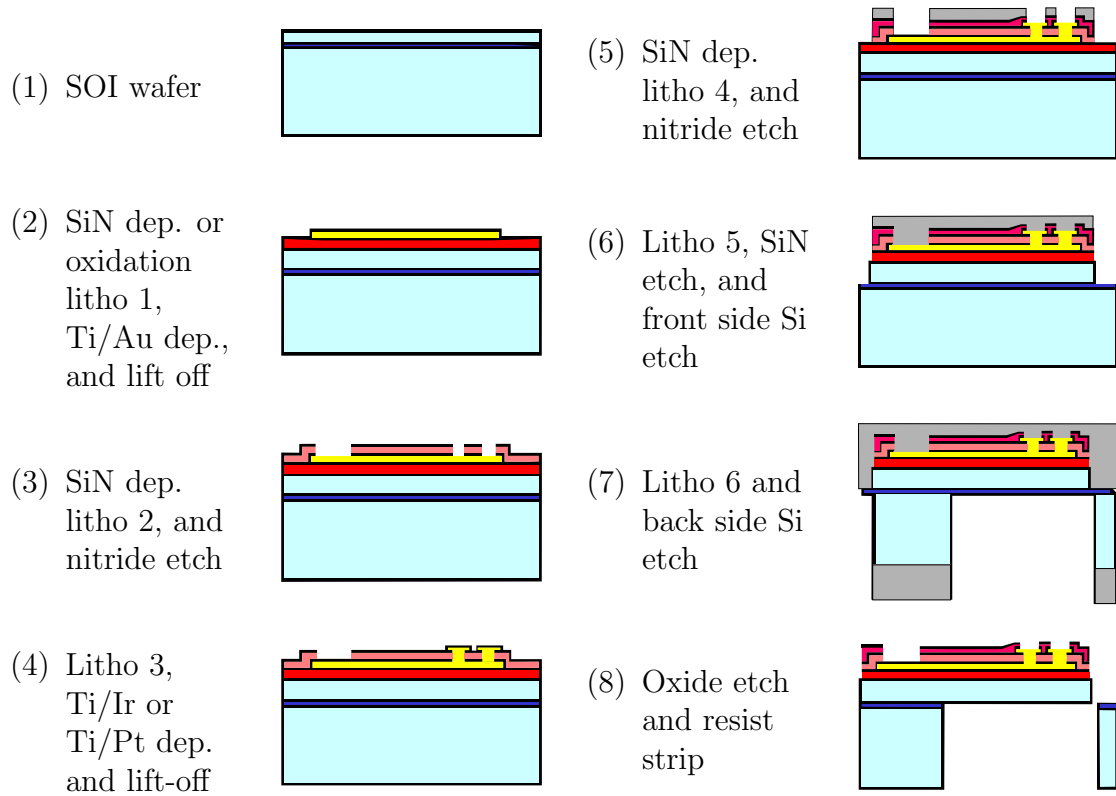


Figure 3.9: Manufacturing process of the VSAMUEL probes (not to scale). For details on the different microstructure techniques see Appendix A

# Chapter 4

## Signal processing

An essential part of the VSAMUEL data acquisition system is the signal processing module. Acquired data must be analyzed automatically by appropriate algorithms in order to extract the desired information. Digital signal processing is used by the data acquisition system to perform the following tasks:

- Separate action potentials and field potentials
- Remove noise
- Detect action potentials
- Classify action potentials

The data acquisition system is able to perform some of these tasks during recording in real time in order to provide direct feedback about signal quality. In particular the results of the first three tasks are relevant for a first ad hoc decision whether a recording setup yields useful signals. The classification of action potentials is a task that could also be performed online. However, not all methods for spike sorting can be applied online as for example a principal component analysis requires that the data is fully available.

The following paragraphs give an overview about the employed signal processing methods [33]. In particular implementation of the wavelet transformation using the lifting scheme is described.

### 4.1 Digital filter

Digital filters are usually instances of linear time invariant (LTI) systems which are used to select, attenuate, or amplify certain frequency components of an input signal  $s(n)$ . The result is an output signal  $g(n)$ . In general an LTI system can be described by a linear difference equation with constant coefficients

$$\sum_{k=0}^N a_k g(n-k) = \sum_{k=0}^M b_k s(n-k) \quad (4.1)$$

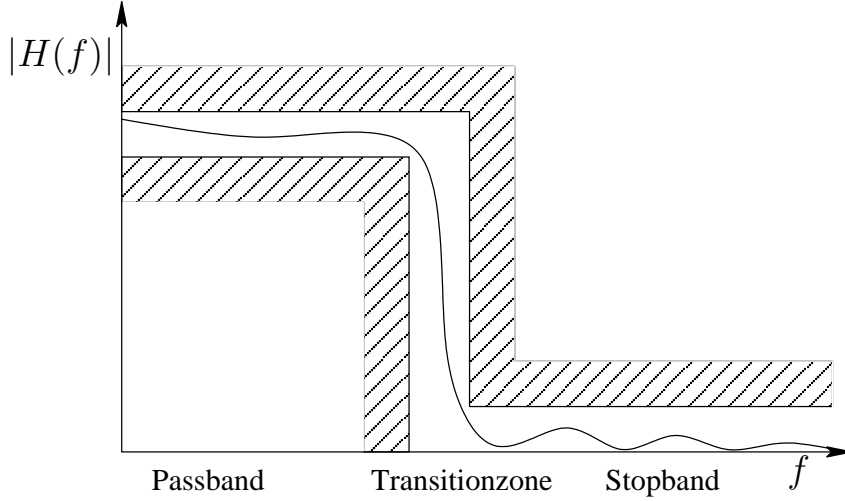


Figure 4.1: General attenuation schema of a filter having passband, transition zone, and stopband drawn in frequency domain.

where initial rest needs to be assumed, i.e.

$$s(n) = \epsilon(n - n_0)s(n) \quad \text{and} \quad g(n_0 - 1) = g(n_0 - 2) = \dots = g(n_0 - N) = 0.$$

For  $N > 0$  the difference equation is recursive and we obtain an infinite impulse response (IIR) filter, while if  $N = 0$  a finite impulse response (FIR) filter is represented.

The characteristics of digital filters can be illustrated using the Fourier transform of the impulse response, namely the frequency response  $H(f)$ , where

$$H(f) = \sum_{n=-\infty}^{\infty} h(n)e^{-j2\pi fn} \quad (4.2)$$

The frequency response is a complex valued function and can be decomposed into its magnitude and phase

$$H(f) = |H(f)|e^{j\angle H(f)}. \quad (4.3)$$

The magnitude  $|H(f)|$  represents amplification or attenuation characteristics of the filter. The phase spectrum represents how frequencies of input signals are delayed by the filter. One way to represent the phase are the principle angle values denoted by  $\angle H(f)$  which are between  $-\pi$  and  $\pi$ . The disadvantage of this representation is, that  $\angle H(f)$  is usually not continuous in  $f$ , because the principle angle values wrap at  $-\pi$  to  $\pi$  and vice versa. However, it is possible to unwrap the principal angle values obtaining a phase function continuous in  $f$  denoted by  $\phi(f) = \arg(H(f))$ . If  $\phi(f)$  is a linear function in  $f$ , then the filter imposes the same delay for all frequencies. The function  $\phi(f)$  can be used to derive an intuitive measure for phase modifications, the so called *group delay*. It is defined as

$$\tau(f) = -\frac{d}{df}\phi(f). \quad (4.4)$$

A system shifts a signal of frequency  $f$  by  $\tau(f)$  samples. The group delay is constant for a filter with linear phase. Deviations of the group delay from a constant indicate the degree of nonlinearity of the phase. Note, that a filter with constant group delay preserves the shape of the original signal better than a filter with non constant group delay. The following paragraph will illustrate this with an example.

#### 4.1.1 FIR and IIR filter

An FIR filter  $h(n)$  with  $M + 1$  taps is applied on a signal  $s(n)$  by computing the convolution as

$$g(n) = h(n) * s(n) = \sum_{k=0}^M h(k)s(n-k) \quad (4.5)$$

This FIR filter is causal, i.e.  $h(n) = 0, n < 0$ . The result  $g(n)$  does not depend on values  $s(k)$  where  $k > n$ . Thus, the computation involves only signal values from the past, and no values from the future. In an online processing system an FIR filter must be causal to be applied. As the signal  $s(n)$  is usually not completely available, it is processed in segments. After processing a segment the filter keeps an internal state needed to process the following segment correctly.

An FIR filter has linear phase and constant group delay if its impulse response is symmetric:

$$h(M-n) = \pm h(n), (M \text{ even}). \quad (4.6)$$

This condition is used in the Parks-McClellan algorithm [54, 87] for FIR filter design which computes linear phase FIR filter for given attenuation constraints. The algorithm was used to compute the coefficients of the FIR filter in the example in the paragraph below. As a rule of thumb the number of taps for the FIR filter increases with stronger constraints on the width of the transition zone and the maximal allowed error in passband and stopband.

IIR filter often have only a few coefficients  $b_k$  and  $a_k$  which are non zero compared to FIR filter which satisfy the same passband, transition, and stopband constraints. Thus the computational cost of FIR filters is usually higher than the computational cost of a corresponding IIR filter. This advantage of IIR filter goes with the disadvantage of being possibly unstable. In fact, instability can be caused just by quantization effects which occur when the IIR filter is implemented e.g. in fixed point 16 bit resolution.

The example in Figure 4.2 and Figure 4.3 illustrates the differences between an FIR filter  $h_{FIR}$  and an IIR filter  $h_{IIR}$  with similar constraints. The test signal, a superposition of sine waves with 80 Hz, 1 kHz, and 5 kHz, is defined as:

$$s(t) = \sin(80 \cdot 2\pi t) + \frac{1}{2} \sin(1000 \cdot 2\pi t) + \frac{1}{10} \sin(5000 \cdot 2\pi t). \quad (4.7)$$

This test signal is sampled at 50 kHz for one second yielding the time discrete function  $s(n)$ . Two digital high pass filters, i.e.  $h_{FIR}$  and  $h_{IIR}$  are designed such that they remove the 80 Hz sine wave from  $s(n)$ . The stopband ends at 500 Hz and the passband starts at 700 Hz. The frequencies 500 Hz and 700 Hz correspond to normalized frequency values 0.020 and 0.028, which are used for the frequency

axis in Figure 4.2. The FIR filter is designed using the Parks-McClellan algorithm which uses the Remez exchange algorithm and Chebyshev approximation theory to design a filter that fits optimally to the desired frequency response, in the sense that it minimizes the maximum error. The IIR filter  $h_{IIR}$  is an order 30 Butterworth filter [93], which are in general characterized by a magnitude response that is maximally flat in the passband and monotonic overall. Butterworth filter have a more linear phase response in the passband than the Chebyshev Type I/Type II and elliptic filters [103]. The magnitudes of the impulse responses for IIR and FIR filter are quite similar. The phase responses on the other hand are different. While the FIR filter has a linear phase with constant group delay of 343 samples, the IIR filter has a non-linear phase response. This translates into a group delay for the passband which starts around 80 samples at 700 Hz decaying to less than one sample at 4575 Hz.

Figure 4.3 shows a closeup, i.e. the first 40 ms of the results of filtering  $s(n)$  with IIR and FIR filter  $h_{IIR}$  and  $h_{FIR}$ . At the top of both plots the original sequence  $s(n)$  is drawn with the delay of the respective filter. Below the results  $g_{IIR}$  and  $g_{FIR}$  is a plot of the expected result:

$$g(t) = \frac{1}{2} \sin(1000 \ 2\pi t) + \frac{1}{10} \sin(5000 \ 2\pi t) = s(t) - \sin(80 \ 2\pi t). \quad (4.8)$$

This and the plot of the difference between actual and expected results at the bottom allows a visual comparison of both. As a consequence of the non-linear phase response, the shape of  $g_{IIR}$  shows more distortion than  $g_{FIR}$  which is quite close to the expected result.

The chosen example is similar to the task of separating local field potentials from action potentials. The problem with IIR filter is that it is likely to change the shape of the action potentials. This must be avoided, since the shape of an action potential is significant for its classification. The FIR filter nicely preserves the shape but it has a high computational cost due to its length. Therefore, both are not well suited to separate LFPs and APs in real time when recording at 50 kHz on 128 channels. A suitable alternative is a high pass filter implemented by means of a wavelet filter bank, which will be described in the following paragraphs.

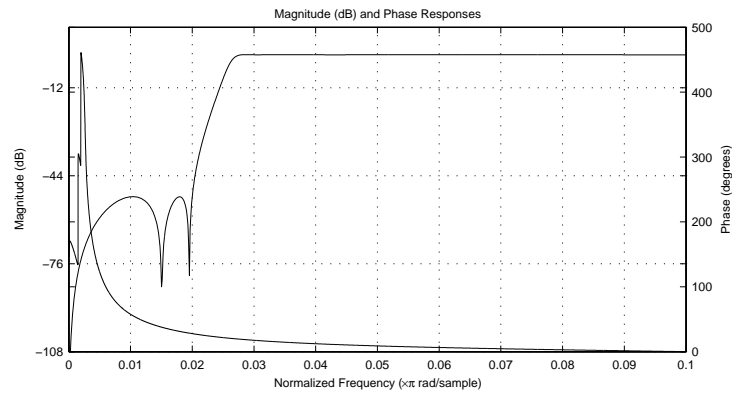
## 4.2 Wavelet Transformation

In this section a short introduction into wavelets and the wavelet transform is given. As W. Sweldens indicates in [95] it is hard to give the definition of a "wavelet", but they could be vaguely described with the following sentence:

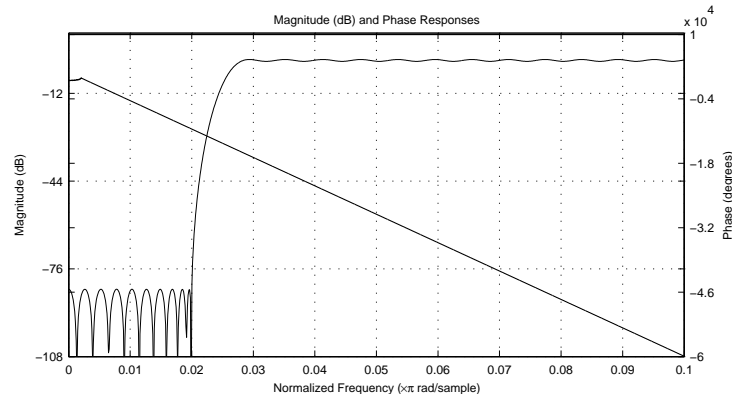
"Wavelets are **building blocks** that can **quickly decorrelate** data."

This sentence reflects three of the main features of wavelets. First, we can see them as *building blocks* for general data sets. More precisely, an arbitrary set of data can be represented by a linear combination of wavelets. And even more precisely, if we denote wavelets by  $\psi_j$  and the coefficients by  $d_j$ , a general function  $s$  can be written as:

$$s = \sum_j d_j \psi_j$$

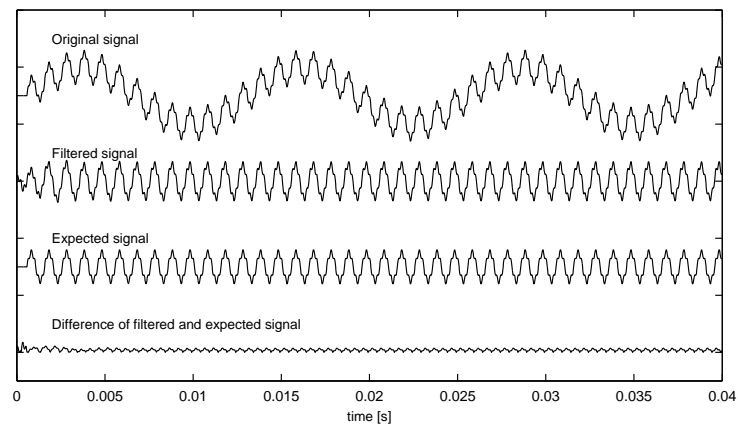


(a) IIR filter

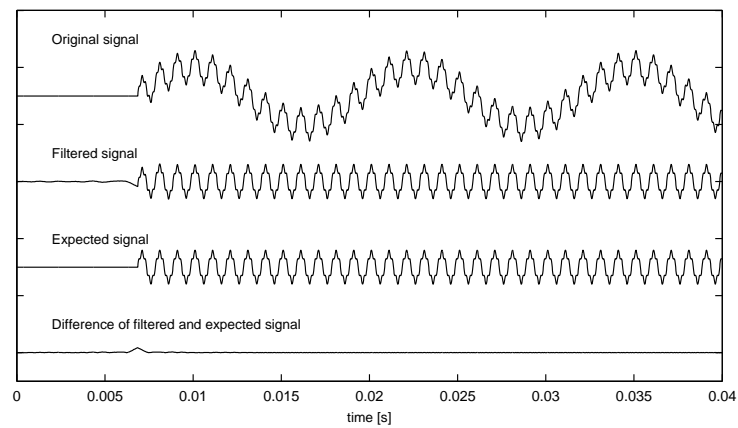


(b) FIR filter

Figure 4.2: Comparison of frequency responses of high pass IIR and FIR filter.



(a) IIR filter



(b) FIR filter

Figure 4.3: Result of applying IIR and FIR high pass filter to a superposition of sine waves.

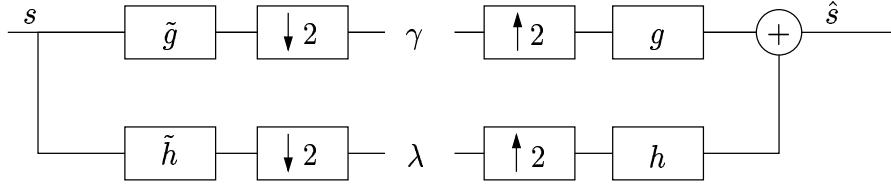


Figure 4.4: Single level decomposition and reconstruction filter bank

The second property is the capability of wavelets to *decorrelate* data. In other words the coefficients  $d_j$  can provide a more compact representation of the data. This feature is e.g. very useful for compression of the data. We only need to store the few coefficients that contain the main information, while we are still able to get a good approximation of the original data. In order to obtain a good decorrelation of the data, the respective wavelet should be constructed such that it resembles the data in some way.

Finally, as third property, we want to have quick algorithms for the computation of the coefficients from the data, and also vice versa, for the reconstruction of the original data from the coefficients. The time complexity of such algorithms should be linear in the order of the size of the data.

The discrete wavelet transform is often implemented with two FIR filters: a low pass filter  $\tilde{h}$  and a high pass filter  $\tilde{g}$ . The coefficients of both filters are derived from the particular wavelet that was chosen. The signal is filtered with both filters and subsequently both results are down sampled by a factor of two. By this *decomposition* step we obtain a high pass part of the signal, the detail coefficients  $d$  and a low pass part, also called the approximation coefficients  $a$ . The decomposition is inverted by the reconstruction step. First detail and approximation coefficients are up sampled by inserting zeros, and subsequently the up sampled coefficients are filtered by the reconstruction filters  $h$  and  $g$ . Adding the results yields the signal  $\hat{s}$ . In Figure 4.4 both steps, decomposition and reconstruction, are shown. If we assume that the chosen wavelet allows perfect reconstruction of the signal, then the only effect of the filter bank in Figure 4.4 is a delay, i.e.  $\hat{s}(n) = s(n - k)$  where  $k > 0$  is the delay, which depends on the order of the wavelet filters.

However, a straight forward implementation of this filter bank scheme leads to an algorithm which is not optimal. Section 4.3 presents a faster implementation for the discrete wavelet transform.

### 4.2.1 Wavelet filter bank

A wavelet filter bank decomposes the signal into a number of frequency bands. The bandwidth of the frequency bands is cut into halves in each step. In Figure 4.5 an example of a filter bank with depth three is given. On the left side are the decomposition steps and on the right side the reconstruction steps. Each filter block – down sampling and up sampling – can be implemented as described in the previous section. At each step in the decomposition we get coefficients representing the details or high frequencies and those representing the approximation or low frequencies of



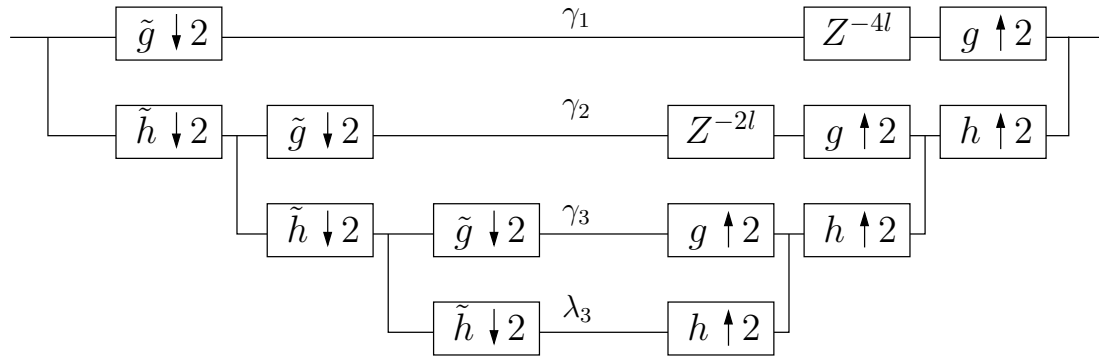


Figure 4.5: Wavelet filter bank of depth three

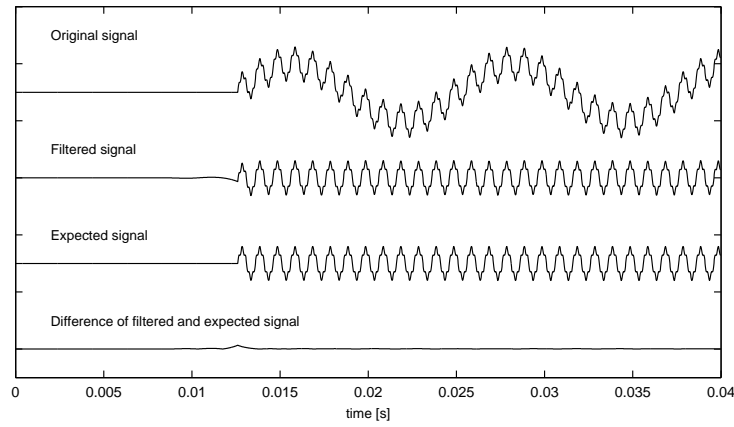


Figure 4.6: High pass filter using a wavelet filter bank

the input signal of the respective step. The approximation is used by the next step for further decomposition. Let  $L$  denote the number of steps, then we end up with the detail coefficients  $(d_j, j = 1, \dots, L)$  at each step and the approximation coefficients of the last step  $(a_L)$ .

For reconstruction the detail coefficients of the first steps must be delayed to account for the delay of the reconstructed approximation coefficients. The overall delay of the filter bank is  $(2^{L+1} - 2)b$ , where  $b$  is the delay of the used wavelet filter. Note, that the delay grows exponentially in the depth of the filter bank.

Figure 4.6 shows the result when a wavelet filter bank is used to implement a high pass filter with similar characteristics as used by the example of Section 4.1.1. The filtered signal has a longer delay compared to the FIR filtered signal in Figure 4.3. But the shape of the signal is preserved due to the nearly linear phase of the wavelet filter bank which is illustrated in Figure 4.7. So the result is quite similar to the expected result.

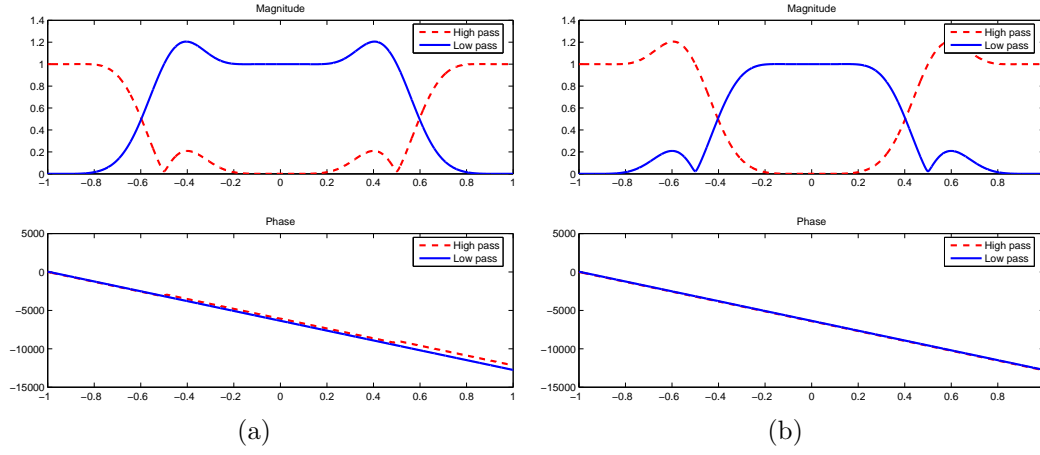


Figure 4.7: Differences between frequency responses of high and low pass filter implemented with a db6 wavelet filter bank, if the input is shifted by one.

### 4.2.2 Translation invariance

The wavelet decomposition is not translation invariant, that is, the coefficients obtained for a signal  $s(n)$  are generally not equal to the coefficients for  $s(n - n_0)$ . This leads to translation dependent frequency responses of filters implemented by means of a wavelet filter bank. Figure 4.7 illustrates this issue. In order to obtain these frequency responses a delta function which is defined as

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \quad (4.9)$$

has been processed within a wavelet filter bank using the Daubechies 2 wavelet. The filter bank is used to implement a high and a low pass filter by discarding either detail or approximation coefficients as described in Section 5.1. The frequency responses are calculated as the Fourier transform of the results. For Figure 4.7(b) the input delta has been shifted by 1 compared to the input delta in Figure 4.7(a). One can see clearly the differences around the cut-off frequencies.

The translation variance is a consequence of down sampling within the wavelet filter bank, because different values are discarded if the input is shifted with different values for  $n_0$ . More details on the translation variance are described in Section 5.4.3.

### 4.3 Fast lifting wavelet transform

In this section a fast implementation of the wavelet transformation will be presented. Within this section FIR filters and signals are equivalently represented by their  $z$ -transforms. The  $z$ -transform of an FIR filter  $h(n)$  with  $M + 1$  taps is defined as:

$$h(z) = \sum_{n=0}^M h(n)z^{-n}. \quad (4.10)$$

A convolution in time domain maps to a multiplication in  $z$ -domain [86], i.e. the convolution  $h(n) * x(n)$  can be written as  $h(z)x(z)$ . The  $z$ -transforms of the FIR filters are polynomials in  $z$ , also called Laurent polynomials. Having a polynomial representation will appear to be a good notation to deduce the theory for the fast implementation of a wavelet transformation.

In the classical implementation of the wavelet transformation two complementary FIR filters  $h(z)$  and  $g(z)$  are applied to a signal  $s(z)$  and both results are subsampled by a factor of 2 leading to a set of approximation coefficients  $a_j$  and a set of detail coefficients  $d_j$ . This implementation has the obvious drawback, that every second result value of the convolution with the FIR filter is discarded by the subsampling. Consider the convolution with an FIR filter  $y(z) = h(z)x(z)$  followed by subsampling:

$$\begin{aligned} & \vdots \\ y_0 &= h_0x_0 + h_1x_{-1}z^{-1} + h_2x_{-2}z^{-2} + \dots \\ y_1 &= h_0x_1 + h_1x_0z^{-1} + h_2x_{-1}z^{-2} + \dots \\ y_2 &= h_0x_2 + h_1x_1z^{-1} + h_2x_0z^{-2} + \dots \\ & \vdots \end{aligned}$$

Supposing that the odd rows are removed by the subsampling the remaining result values in  $y_e(z)$  are either obtained by

- even numbered filter coefficients  $h_e$  multiplied with even numbered samples  $x_e$ , or
- odd numbered filter coefficients  $h_o$  multiplied with odd numbered samples  $x_o$ .

In other terms the FIR filter with subsequent downsampling (Figure 4.8) can be rewritten as

$$y_e(z) = h_e(z)x_e(z) + z^{-1}h_o(z)x_o(z) \quad (4.11)$$

where  $h_e(z)$  and  $h_o(z)$  are defined as

$$h_e(z) = \sum_n h_{2n}z^{-n} \text{ and } h_o(z) = \sum_n h_{2n+1}z^{-n}. \quad (4.12)$$

The delay  $z^{-1}$  in front of the odd part in (4.11) comes from the delay between even and odd samples. Now we apply this remodeling of the FIR filter with subsequent

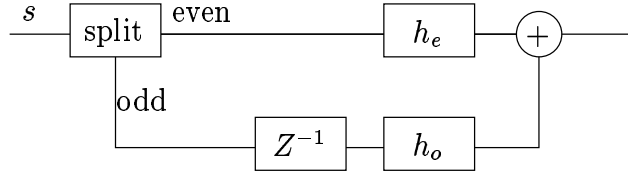


Figure 4.8: Efficient implementation of an FIR filter followed by a subsampler

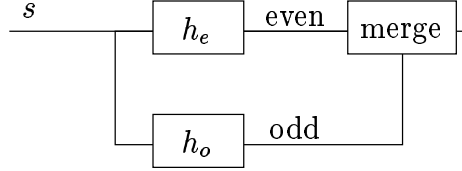


Figure 4.9: Efficient implementation of upsampling with subsequent FIR filter

downsampling to an analysis step of the wavelet transformation. Thus we rewrite both parts, i.e. filtering with  $\tilde{h}(z)$  and the filtering with  $\tilde{g}(z)$  and the subsampling like we did in (4.11). This leads to two equations and in vector notation these equations can be written as:

$$\begin{pmatrix} a(z) \\ d(z) \end{pmatrix} = \tilde{P}(z) \begin{pmatrix} x_e(z) \\ z^{-1}x_o(z) \end{pmatrix} \quad (4.13)$$

where  $\tilde{P}(z)$  is the so called *polyphase matrix* which is defined as

$$\tilde{P}(z) = \begin{bmatrix} \tilde{h}_e(z) & \tilde{h}_o(z) \\ \tilde{g}_e(z) & \tilde{g}_o(z) \end{bmatrix} \quad (4.14)$$

The polyphase matrix performs one decomposition step of the wavelet transformation.

The synthesis or reconstruction step can also be described by a polyphase matrix. For the reconstruction both the detail coefficients and the approximation coefficients are first upsampled and then an FIR filter is applied to the upsampled signals. Upsampling is basically insertion of zero values in between two successive values. Therefore, during the FIR filtering a lot of operations are multiplications with zero. For a fast implementation we want to omit these operations. We consider (4.11) as the result of an FIR filtering of an upsampled signal  $x$  with a filter  $h$  whereby the zeros are inserted at the odd positions of  $x$ . Since all values with odd indices in  $x$  are zero, i.e.  $x_o(z)$  is zero, we just have to deal with  $x_e(z)$ . The output values at even indices are obtained by convolution of  $h_e(z)$  with  $x_e(z)$  and the values at odd indices are obtained by convolution of  $h_o(z)$  with  $x_e(z)$  (Figure 4.9). This is summarized as

$$y_e = h_e(z)x_e(z) \quad (4.15)$$

$$zy_o = h_o(z)x_e(z). \quad (4.16)$$

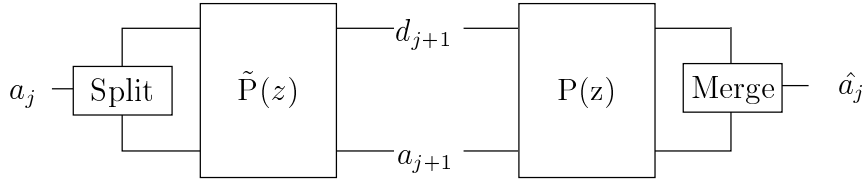


Figure 4.10: One stage wavelet filter bank in terms of polyphase matrices.

Similar to equation (4.13) the synthesis step of the wavelet transform can be written as

$$\begin{pmatrix} y_e(z) \\ zy_o(z) \end{pmatrix} = P(z) \begin{pmatrix} a_e(z) \\ d_e(z) \end{pmatrix} \quad (4.17)$$

where the synthesis polyphase matrix  $P(z)$  is defined as

$$P(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix}. \quad (4.18)$$

A one stage wavelet filter bank with an analysis and a synthesis step in terms of polyphase matrices is shown in Figure 4.10. The delays are incorporated into the split and merge operators.

Since the wavelet transform is completely represented by the polyphase matrices its properties can also be written in terms of polyphase matrices. For instance, if a wavelet transform allows perfect reconstruction then the analysis and synthesis are related as

$$P(z)\tilde{P}(z^{-1})^t = I \quad (4.19)$$

The analysis polyphase matrix is time reversed, because (4.19) holds only if the delays caused by the polyphase matrices cancel each other.

In order to obtain a lifting scheme we need to decompose the polyphase matrix into a product of matrices with an appropriate structure:

$$P(z) = \prod_{i=1}^m \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \quad (4.20)$$

Following [14] we give a factoring algorithm which proves by construction that it is always possible to find such a decomposition for a given complementary filter pair  $(h, g)$ . Note, that a filter pair  $(h, g)$  is called complementary, if its polyphase matrix has determinant 1. The algorithm is based on the Euclidean algorithm for Laurent polynomials. This Euclidean algorithm computes the greatest common divisor of two given Laurent polynomials. While the Euclidean algorithm for integer numbers has a unique solution, we do not have a unique solution in case of Laurent polynomials. This will turn out to be an advantage, since we can select a solution w.r.t. certain criterions. With operator  $|\cdot|$  we denote the degree of a Laurent polynomial, which is the difference between maximal and minimal order of the monomials with nonzero coefficients.

Euclidean Algorithm for Laurent polynomials as given in [14]

Take two Laurent polynomials  $a(z)$  and  $b(z) \neq 0$  with  $|a(z)| \geq |b(z)|$ . Let  $a_0(z) := a(z)$  and  $b_0(z) := b(z)$  and iterate the following steps starting from  $i = 0$  until  $b_i(z) = 0$ :

1. Choose  $b_{i+1}(z)$  and  $q_{i+1}(z)$  such that

$$\begin{aligned} |b_{i+1}(z)| &< |b_i(z)| \text{ and} \\ a_i(z) &= b_i(z)q_{i+1}(z) + b_{i+1}(z) \end{aligned}$$

2. Let  $a_{i+1}(z) := b_i(z)$

3. Increment  $i$ .

Then  $a_n(z) = \gcd(a(z), b(z))$  where  $n$  is the smallest number for which  $b_n(z) = 0$ .

At each step we must make sure that  $|b_{i+1}(z)| < |b_i(z)|$ . Then there is an  $m$  at which  $|b_m(z)| = 0$  and the algorithm finishes after  $n = m + 1$  steps. Therefore the number of steps is bound by  $n \leq |b(z)| + 1$ . However, this is the number of steps to find a single solution. At each step we can decide which monomials should be eliminated by the division. This means that there are different possibilities for  $b_{i+1}(z)$ . For example, if  $|a_i(z)| = 3$  and  $|b_i(z)| = 2$  there are three solutions for  $b_{i+1}(z)$ . We will use an example which is also given in [14] to clarify this. Let  $a(z) = z^{-1} + 6 + z$  and  $b(z) = 4 + 4z$ . Now find a polynomial  $q(z)$  of degree 1 so that the remainder  $r(z)$  given as

$$r(z) = a(z) - b(z)q(z) \tag{4.21}$$

has degree zero, i.e. degree of  $b(z)$  minus 1. There are three solutions:

$$\begin{aligned} r(z) &= (z^{-1} + 6 + z) - (4 + 4z) \left( \frac{1}{4}z^{-1} + \frac{5}{4} \right) = -4z \\ r(z) &= (z^{-1} + 6 + z) - (4 + 4z) \left( \frac{1}{4}z^{-1} + \frac{1}{4} \right) = 4 \\ r(z) &= (z^{-1} + 6 + z) - (4 + 4z) \left( \frac{5}{4}z^{-1} + \frac{1}{4} \right) = -4z^{-1} \end{aligned}$$

In the first solution we have eliminated the term in  $z^{-1}$  and the constant, in the second solution the terms in  $z^{-1}$  and in  $z$ , and finally in the third solution the constant and the term in  $z$ . Either of these  $r(z)$  is a possible value for  $b_{i+1}(z)$ , given  $b_i(z) = b(z)$  and  $a_i(z) = a(z)$ .

A particular filter pair  $(h, g)$  is factorized into lifting steps by applying the Euclidean algorithm with  $a_0(z) = h_e(z)$  and  $b_0(z) = h_o(z)$ . Note, that we can always find a constant greatest common divisor  $K$  as we start with a complementary filter pair  $(h, g)$  which polyphase matrix has determinant 1 by definition. This implies that the Laurent polynomials  $h_e(z)$  and  $h_o(z)$  are relative prime, because any common

factor would also divide the determinant of the polyphase matrix, which is

$$\det P(z) = \det \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} = h_e(z)g_o(z) - h_o(z)g_e(z) = 1. \quad (4.22)$$

Thus the gcd of  $h_e(z)$  and  $h_o(z)$  is a monomial. With the non-uniqueness of the division factors  $q_i(z)$  in the Euclidean algorithm we can always find a sequence of quotients  $q_i(z)$  which leads to a constant gcd.

The result of the Euclidean algorithm is the greatest common divisor (gcd)  $K$  and a sequence of quotients  $q_i(z)$  which fullfills

$$\begin{pmatrix} h_e(z) \\ h_o(z) \end{pmatrix} = \prod_{i=1}^n \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} K \\ 0 \end{pmatrix}. \quad (4.23)$$

If  $|h_o(z)| > |h_e(z)|$  then the first quotient  $q_1(z)$  is zero. Furthermore, we can always assume, that  $n$  is even, because if  $n$  is odd, we can multiply  $h(z)$  with  $z$  and  $g(z)$  with  $-z^{-1}$ . This preserves the determinant of the polyphase matrix being 1.

With (4.23) we have found a factorization for  $h$ . But the factorization for  $g$  is still missing. Note, that the determinant of a matrix as it is used in (4.23) is always  $-1$  regardless of  $q(z)$ :

$$\det \begin{bmatrix} q(z) & 1 \\ 1 & 0 \end{bmatrix} = q(z) \cdot 0 - 1 = -1. \quad (4.24)$$

Therefore the determinant of the product of an even number of such matrices is always 1. This in turn implies that the factorization

$$P^0(z) = \begin{bmatrix} h_e(z) & g_e^0(z) \\ h_o(z) & g_o^0(z) \end{bmatrix} = \prod_{i=1}^n \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \quad (4.25)$$

has also determinant 1, as the determinant of the matrix with constant entries has determinant 1. Hence, the filter  $g^0(z)$  composed from  $g_e^0(z)$  and  $g_o^0(z)$  is complementary to  $h$ . However, it is not equal to  $g$ . The factorization in (4.25) must be transformed into a factorization of form (4.20). Notice that

$$\begin{bmatrix} q(z) & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & q(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q(z) & 1 \end{bmatrix}. \quad (4.26)$$

If we replace each factor with even index  $i$  in (4.25) by the first equation and each factor with odd index  $i$  by the second one, then we obtain the following form of (4.25), because the permutation matrices cancel each other:

$$P^0(z) = \begin{bmatrix} h_e(z) & g_e^0(z) \\ h_o(z) & g_o^0(z) \end{bmatrix} = \prod_{i=1}^{n/2} \begin{bmatrix} 1 & q_{2i-1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_{2i} & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix}. \quad (4.27)$$

Since  $P^0(z)$  corresponds to the filter pair  $(h, g^0)$  a last lifting step is required which recovers the filter  $g$  from  $g^0$ . Given the lifting theorem from [14] this single step exists

and is sufficient. Therefore, the factorization of the polyphase matrix corresponding to filter pair  $(h, g)$  is

$$P(z) = P^0(z) \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix} \quad (4.28)$$

Combining all results from above it has been shown by construction that for any given complementary filter pair  $(h, g)$  a factorization into lifting steps can be found. The factorization has the form given in (4.20).

### 4.3.1 Example

In order to derive a sequence of lifting steps we first need to obtain the polyphase matrix which represents a single analysis step. In the following example we use the Daubechies wavelet with two vanishing moments, which filters are given in [13] as

$$h(z) = h_0z + h_1 + h_2z^{-1} + h_3z^{-2} \quad (4.29)$$

$$g(z) = h_3z - h_2 + h_1z^{-1} - h_0z^{-2} \quad (4.30)$$

with coefficients defined as

$$h_0 = \frac{1 + \sqrt{3}}{4\sqrt{2}}, \quad h_1 = \frac{3 + \sqrt{3}}{4\sqrt{2}}, \quad h_2 = \frac{3 - \sqrt{3}}{4\sqrt{2}}, \quad \text{and} \quad h_3 = \frac{1 - \sqrt{3}}{4\sqrt{2}}. \quad (4.31)$$

Since this filter is orthogonal, analysis and synthesis polyphase matrices are equal. They are constructed according to (4.12) as

$$P(z) = \tilde{P}(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} = \begin{bmatrix} h_1 + h_3z^{-1} & -h_2 - h_0z^{-1} \\ h_0z + h_2 & h_3z + h_1 \end{bmatrix} \quad (4.32)$$

Now we need to apply the Euclidean algorithm for Laurent polynomials  $h_e(z)$  and  $h_o(z)$  in order to obtain a factorization of the polyphase matrix.

$$\begin{aligned} a_0(z) &= h_1 + h_3z^{-1} \\ b_0(z) &= h_0z + h_2 \\ a_1(z) &= b_0(z) \\ b_1(z) &= a_0(z) - b_0(z)q_1(z) \\ &= (h_1 + h_3z^{-1}) - (h_0z + h_2)\left(\frac{h_3}{h_2}z^{-1}\right) \\ &= h_1 - \frac{h_0h_3}{h_2} =: r_0 \\ a_2(z) &= b_1(z) = r_0 \\ b_2(z) &= a_1(z) - b_1(z)q_2(z) \\ &= (h_0z + h_2) - r_0\left(\frac{h_0}{r_0}z + \frac{h_2}{r_0}\right) \\ &= 0 \end{aligned}$$



The quotient  $q_1(z)$  was chosen so that the term in  $z^{-1}$  of  $a_0(z)$  is eliminated. The constant remainder  $r_0$  is the constant  $K$ . Now we can construct  $g^0(z)$  as

$$P^0(z) = \begin{bmatrix} h_e(z) & g_e^0(z) \\ h_o(z) & g_o^0(z) \end{bmatrix} \quad (4.33)$$

$$= \begin{bmatrix} q_1(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} q_2(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \quad (4.34)$$

$$= \begin{bmatrix} q_1(z)q_2(z) + 1 & q_1(z) \\ q_2(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \quad (4.35)$$

$$= \begin{bmatrix} K(q_1(z)q_2(z) + 1) & q_1(z)/K \\ Kq_2(z) & 1/K \end{bmatrix}. \quad (4.36)$$

Therefore we obtain  $g_e^0(z) = q_1(z)/K$  and  $g_o^0(z) = 1/K$ . The missing lifting step can be obtained from (4.28) as there are two equations which need to hold:

$$g_e(z) = s(z)K(q_1(z)q_2(z) + 1) + q_1(z)/K \quad (4.37)$$

$$g_o(z) = s(z)Kq_2(z) + 1/K \quad (4.38)$$

We use the second one and resolve it for  $s(z)$ :

$$s(z) = \frac{g_o(z) - 1/K}{Kq_2(z)} = \frac{h_3z + h_1 - 1/K}{h_0z + h_2} = \frac{h_3}{h_0} \quad (4.39)$$

Now we have found a factorization of (4.32), i.e.

$$\begin{aligned} P(z) = \tilde{P}(z) &= \begin{bmatrix} 1 & q_1(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_2(z) & 1 \end{bmatrix} \begin{bmatrix} r_0 & 0 \\ 0 & 1/r_0 \end{bmatrix} \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & q_1(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_2(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & r_0^2 s(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r_0 & 0 \\ 0 & 1/r_0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & \frac{h_3}{h_2} z^{-1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{h_0}{r_0} z + \frac{h_2}{r_0} & 1 \end{bmatrix} \begin{bmatrix} 1 & r_0^2 \frac{h_3}{h_0} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r_0 & 0 \\ 0 & 1/r_0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & -\frac{1}{\sqrt{3}} z^{-1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{\sqrt{3}}{4} z + \frac{6-3\sqrt{3}}{4} & 1 \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{3} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{3+\sqrt{3}}{3\sqrt{2}} & 0 \\ 0 & \frac{3-\sqrt{3}}{\sqrt{2}} \end{bmatrix} \end{aligned}$$

If we consider this as the factorization of the synthesis polyphase matrix  $P(z)$  then the respective analysis implementation is obtained as  $\tilde{P}(z^{-1})^t$ :

$$\tilde{P}(z^{-1})^t = \begin{bmatrix} \frac{3+\sqrt{3}}{3\sqrt{2}} & 0 \\ 0 & \frac{3-\sqrt{3}}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{\sqrt{3}}{4} z^{-1} + \frac{6-3\sqrt{3}}{4} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{\sqrt{3}} z^1 & 1 \end{bmatrix}. \quad (4.40)$$

The steps of the implementation are derived from this factorization as:

$$d_l^{(0)} = x[2l + 1] \quad (4.41)$$

$$a_l^{(0)} = x[2l] \quad (4.42)$$

$$d_l^{(1)} = d_l^{(0)} - 1/\sqrt{3}a_{l+1}^{(0)} \quad (4.43)$$

$$a_l^{(1)} = a_l^{(0)} + (6 - 3\sqrt{3})/4d_l^{(1)} + \sqrt{3}/4d_{l-1}^{(1)} \quad (4.44)$$

$$d_l^{(2)} = d_l^{(1)} - 1/3a_l^{(1)} \quad (4.45)$$

$$a_l^{(2)} = (3 + \sqrt{3})/(3\sqrt{2})a_l^{(1)} \quad (4.46)$$

$$d_l^{(3)} = (3 - \sqrt{3})/(\sqrt{2})d_l^{(2)} \quad (4.47)$$

$$(4.48)$$

The inverse, i.e. the synthesis step can be easily derived by reversing the order and changing the signs of the previous steps.

### 4.3.2 Mapping onto integer values

The lifting scheme can be modified such that it maps integers onto integers, while computing the coefficients in floating point arithmetic. We introduce a rounding function each time we need to update the coefficients  $d_j$  or  $a_j$ . However, the iteration usually includes a scaling as last step which is not invertible without loss if the result is rounded. But we can replace the two scaling steps by four lifting steps, whereby the first lifting step of the scaling can always be merged with the last lifting step of the factorization [14]. Scaling has the following different factorizations:

$$\begin{aligned} P(z) &= \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \\ &= \begin{bmatrix} 1 & K - K^2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1/K & 1 \end{bmatrix} \begin{bmatrix} 1 & K - 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \end{aligned} \quad (4.49)$$

or

$$\begin{aligned} P(z) &= \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 - 1/K \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ K & 1 \end{bmatrix} \begin{bmatrix} 1 & 1/K^2 - 1/K \\ 0 & 1 \end{bmatrix}. \end{aligned} \quad (4.50)$$

We apply (4.49) to (4.41) and add the rounding. This leads to the following implementation, which maps integer values onto integer values, whereby  $K = (3 + \sqrt{3})/(3\sqrt{2})$  and  $[\cdot]$  denotes the rounding operation.

$$\begin{aligned}
d_l^{(0)} &= x[2l+1] \\
a_l^{(0)} &= x[2l] \\
d_l^{(1)} &= d_l^{(0)} + \left[-1/\sqrt{3}a_{l+1}^{(0)}\right] \\
a_l^{(1)} &= a_l^{(0)} + \left[(6-3\sqrt{3})/4d_l^{(1)} + \sqrt{3}/4d_{l-1}^{(1)}\right] \\
d_l^{(2)} &= d_l^{(1)} + \left[2/3a_l^{(1)}\right] \\
a_l^{(2)} &= a_l^{(1)} + \left[(K-1)d_l^{(2)}\right] \\
d_l^{(3)} &= d_l^{(2)} + \left[-1/Ka_l^{(2)}\right] \\
a_l^{(3)} &= a_l^{(2)} + \left[(K-K^2)d_l^{(3)}\right]
\end{aligned}$$

This iteration is also invertible by reversing the order of the steps and changing the signs from plus to minus. Therefore the perfect reconstruction property is not affected by introduction of the rounding operation. However, if we additionally consider that the number of bits per integer number is bounded for example by 16 then we cannot assure that coefficient values become not greater than numbers representable by 16 bit. After such an overflow it is not possible to reconstruct the original signal. The overflow can be bound by the triangular equation as

$$\left\| \sum_{k=-\infty}^{\infty} x(k)h(n-k) \right\| \leq \sum_{k=-\infty}^{\infty} |x(k)h(n-k)| \leq x_{\max} \sum_{k=-\infty}^{\infty} |h(n-k)| \quad (4.51)$$

where  $x_{\max}$  is the maximum magnitude of all elements of  $x(k)$ . We can conclude that there is no overflow if

$$\|h\|_1 = \sum_{k=-\infty}^{\infty} |h(n-k)| \leq 1. \quad (4.52)$$

In order to avoid an overflow we can scale the wavelet FIR filter such that the condition in (4.52) is fulfilled. But such a scaling will make perfect reconstruction impossible, because scaling of integer values by arbitrary floating point values is generally not invertible. As an alternative we modify  $K$  such that it scales the approximation coefficients by  $1/\|h\|_1$ , i.e.  $\tilde{K} = K/\|h\|_1$ . The detail coefficients are therefore scaled by the factor  $\|h\|_1$ . With this setting we can assure that no overflow will occur within the approximation coefficients. But we cannot guarantee that the detail coefficients will not overflow. In fact, we can easily construct a signal which will cause an overflow, i.e. a signal which values have magnitude  $s_{\max}$  with alternating signs. However, such signals will virtually not occur in our field of application. With this solution we retain the perfect reconstruction property for all relevant signals.

It turns out that a sine wave with a frequency within the recorded spectrum can cause an overflow within the detail coefficients. This is why we decided not only to

provide an integer based wavelet filter bank but also a floating point version. This version was implemented on the host side and utilizes the Pentium instruction set extensions for signal processing.

The main drawback of mapping to integer coefficients are quantization errors. The rounding operation modifies the contents of the different frequency bands. If the coefficients are not changed then this effect cancels out during reconstruction. So integer coefficients are e.g. suitable for lossless compression. However, if we implement a filter based on modifications of the coefficients, then the reconstructed signal suffers from quantization errors. The strategy described above which is supposed to avoid overflows increases the quantization errors even more. Also the effect of translation variance becomes larger.

As a consequence the mapping on integer coefficients is not optimal for our application..

### 4.3.3 Implementing lifting steps on a C6701

The actual implementation of the lifting scheme on a DSP includes serveral steps. In this section will give an overview on how to create an optimized assembly routine given a factorization of a polyphase matrix.

We will implement the Daubechies wavelet with three vanishing moments which can be factorized as follows:

$$\begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \begin{bmatrix} 1 & \lambda \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \epsilon z^{-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma & 1 \end{bmatrix} \begin{bmatrix} 1 & \beta z + \beta' \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix} \quad (4.53)$$

where

$$\begin{aligned} \alpha &= +4.122865950520000e - 01 \\ \beta &= +1.565136279631000e + 00 \\ \beta' &= +4.160137790400000e - 02 \\ \gamma &= -2.814352831900000e - 02 \\ \delta &= -3.984066649260000e - 01 \\ \epsilon &= -4.812981574140000e - 01 \\ \lambda &= +3.984066649266020e - 01 \\ K &= +5.155407358530000e - 01 \end{aligned}$$

We used a tool which computes all possible factorizations given a pair of complementary FIR filters. From this set of factorizations one was choosen w.r.t. a number of criterions:

- Number of lifting steps should be high
- Each lifting step should have at most order 1, i.e. only previous or next value from previous step is used

- Coefficients of the lifting steps should be in the same order of magnitude
- Leftmost factor must be a scaling by  $K$  and  $1/K$  of  $a_j$  and  $d_j$ , respectively

High number of lifting steps and a low order of each lifting step are related to each other, since the low order of each lifting step implies that the number of lifting steps is relatively high. We need lifting steps with maximal order 1, i.e. these lifting steps incorporate only the left or right neighbor from the previous lifting step. This factorization leads to the following iteration:

$$\begin{aligned}
d_l^{(0)} &= x[2l+1] \\
a_l^{(0)} &= x[2l] \\
d_l^{(1)} &= d_l^{(0)} + [\alpha a_l^{(0)}] \\
a_l^{(1)} &= a_l^{(0)} + [\beta d_{l-1}^{(1)} + \beta' d_l^{(1)}] \\
d_l^{(2)} &= d_l^{(1)} + [\gamma a_l^{(1)}] \\
a_l^{(2)} &= a_l^{(1)} + [\delta d_l^{(2)}] \\
d_l^{(3)} &= d_l^{(2)} + [\epsilon a_{l+1}^{(2)}] \\
a_l^{(3)} &= a_l^{(2)} + [(\lambda + (1/K^2 - 1/K))d_l^{(3)}] \\
d_l^{(4)} &= d_l^{(3)} + [K a_l^{(3)}] \\
a_l^{(4)} &= a_l^{(3)} + [(1 - 1/K)d_l^{(4)}]
\end{aligned}$$

As a first step towards a realtime implementation we need to make this iteration causal. In this example step 3 needs  $a_{l+1}^{(2)}$ , which is not available in an online environment. To overcome this issue, we shift this step and all successive steps and use the value we obtained in the previous iteration at this step to proceed. This leads to the following modified iteration:

$$\begin{aligned}
d_l^{(0)} &= x[2l+1] \\
a_l^{(0)} &= x[2l] \\
d_l^{(1)} &= d_l^{(0)} + [\alpha a_l^{(0)}] \\
a_l^{(1)} &= a_l^{(0)} + [\beta d_{l-1}^{(1)} + \beta' d_l^{(1)}] \\
d_l^{(2)} &= d_l^{(1)} + [\gamma a_l^{(1)}] \\
a_l^{(2)} &= a_l^{(1)} + [\delta d_l^{(2)}] \\
d_{l-1}^{(3)} &= d_{l-1}^{(2)} + [\epsilon a_l^{(2)}] \\
a_{l-1}^{(3)} &= a_{l-1}^{(2)} + [(\lambda + (1/K^2 - 1/K))d_{l-1}^{(3)}] \\
d_{l-1}^{(4)} &= d_{l-1}^{(3)} + [K a_{l-1}^{(3)}] \\
a_{l-1}^{(4)} &= a_{l-1}^{(3)} + [(1 - 1/K)d_{l-1}^{(4)}]
\end{aligned}$$

From this iteration we derived that we need to keep track on three states in each iteration, i.e.  $d_{l-1}^{(1)}$ ,  $d_{l-1}^{(2)}$ , and  $a_{l-1}^{(2)}$  which are all not available from the current iteration. In the implementation we deal with three main data structures: the signal  $x$ , the constants denoted by  $c$ , and the states denoted by  $s$ . Constants and states are stored in arrays of 32 bit elements with predetermined length – known e.g. at compile time. The constants are always float values and we will assume that the states are always 32 bit integer numbers. All equations basically have the same structure and can be implemented by similar sequences of assembler instructions. This is illustrated for a single equation:

$$a_l^{(1)} = a_l^{(0)} + [\beta d_{l-1}^{(1)} + \beta' d_l^{(1)}] \quad (4.54)$$

LDW	$c[\beta]$ , cst	load constant $\beta$
LDW	$s[d_{l-1}^{(1)}]$ , temp1	load $d_{l-1}^{(1)}$ from $s$
STW	d, $s[d_{l-1}^{(1)}]$	update $d_{l-1}^{(1)}$ in $s$ with $d_l^{(1)}$
INTSP	temp1, temp1	convert $d_{l-1}^{(1)}$ to single precision
MPYSP	cst, temp1, temp1	multiply constant $\beta$ and $d_{l-1}^{(1)}$
INTSP	d, temp0	convert $d_l^{(1)}$ to single precision
LDW	$c[\beta']$ , cst	load constant $\beta'$
MPYSP	cst, temp0, temp1, temp0	multiply constant $\beta'$ and $d_l^{(1)}$
ADDSP	temp0, temp1, temp0	add results of previous products
SPINT	temp0	convert it to integer
ADD	temp0, a, a	add this to $a_l^{(0)}$ and obtain $a_l^{(1)}$

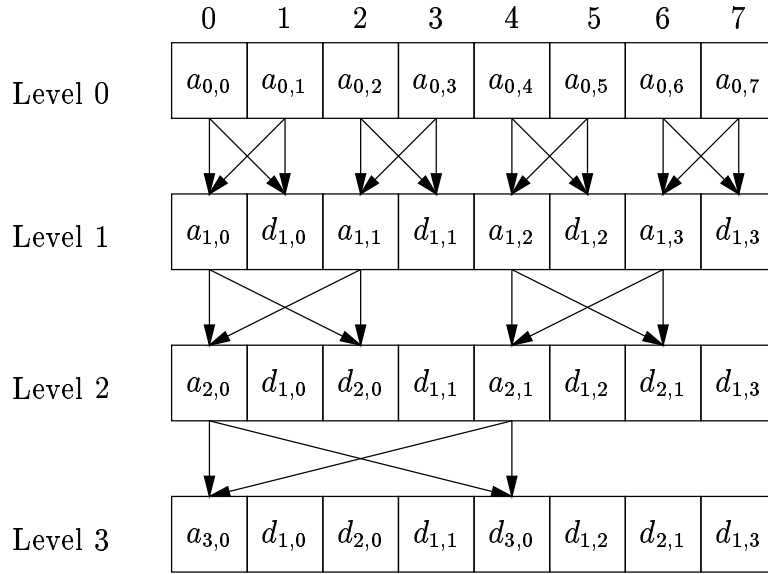


Figure 4.11: Organization of wavelet coefficients during the decomposition with the lifting algorithm. A block of 8 coefficient is completely decomposed, i.e. in three decomposition steps. The last row shows that the coefficients of the different levels are interleaved according to the so called lifting order.

This sequence of assembler commands is not well suited for optimization using the capabilities of the C6701. Mostly results from an assembly instruction are needed for the next instruction. This prevents us from using pipelining and parallel instructions for direct acceleration of the iteration. On the other hand, we can process multiple signals in parallel. In fact, the iteration uses only a small number of registers for a single signal. Moreover, some of these registers can be reused for all signals, as the constants pointer, the constant value (cst), and the states pointer. Furthermore we can assume that the sample values of the signals are particularly arranged in memory so that we can efficiently retrieve samples from multiple signals. Four registers, i.e. *a*, *d*, *temp0*, and *temp1* are needed individually for each signal.

We developed a semiautomatic tool which can generate linear assembler routines that implement wavelet decomposition and reconstruction algorithm based on the lifting scheme.

The generated code uses the capabilities of the C6701 at a great extend. The basic optimization strategy was to process multiple channels in parallel. Note, that 'in parallel' is meant literally. The C6701 has two register sets, each having 16 general purpose registers of 32 bit width. Each register set is associated with several operation units, which can perform operations in parallel. We managed to process four channels in parallel. This implementation uses the minimum number of registers per channel. However, all the 32 registers are directly or indirectly (counters, parameters) necessary for the optimized implementation.

In Figure 4.12 a fragment of the generated assembler code is shown. These

<pre> ; Step 1, predict    LDW .D2T2 **c[1],cst2 ; load constant LDDW .D1T1 **states1[0],temp1_i:temp1_ii ;    LDDW .D2T2 **states2[1],temp1_iii:temp1_iv ; STW .D1T1 a_i,**states1[0] ;    STW .D2T2 a_iii,**states2[2] ; STW .D1T1 a_ii,**states1[1] ;    STW .D2T2 a_iv,**states2[3] ; ADDAW .D1 states1,4,states1 ;    ADDAW .D2 states2,4,states2 ; MPYSP .M1 temp1_i,cst2,temp1_i ;    MPYSP .M2 temp1_iii,cst2,temp1_iii ; MPYSP .M1 temp1_ii,cst2,temp1_ii ;    MPYSP .M2 temp1_iv,cst2,temp1_iv ; LDW .D2T2 **c[2],cst2 ; load constant MV .L1 a_i,temp2_i ;    MV .D1 a_ii,temp2_ii ;    MV .L2 a_iii,temp2_iii ;    MV .D2 a_iv,temp2_iv ; MPYSP .M1 temp2_i,cst2,temp2_i ;    MPYSP .M2 temp2_iii,cst2,temp2_iii ; MPYSP .M1 temp2_ii,cst2,temp2_ii ;    MPYSP .M2 temp2_iv,cst2,temp2_iv ; ADDS .L1 temp1_i,temp2_i,temp1_i ;    ADDSP .L2 temp1_iii,temp2_iii,temp1_iii ; ADDS .L1 temp1_ii,temp2_ii,temp1_ii ;    ADDSP .L2 temp1_iv,temp2_iv,temp1_iv ; LDDW .D1T1 **states1[2],temp2_i:temp2_ii ;    LDDW .D2T2 **states2[3],temp2_iii:temp2_iv ; LDDW .D1T1 **states1[0],temp3_i:temp3_ii ;    LDDW .D2T2 **states2[1],temp3_iii:temp3_iv ; </pre>	<pre> STW .D1T1 temp2_i,**states1[0]; STW .D2T2 temp2_iii,**states2[2]; STW .D1T1 temp2_ii,**states1[1];    STW .D2T2 temp2_iv,**states2[3]; STW .D1T1 d_i,**states1[4] ;    STW .D2T2 d_iii,**states2[6] ; STW .D1T1 d_ii,**states1[5] ;    STW .D2T2 d_iv,**states2[7] ; ADDAW .D1 states1,8,states1 ;    ADDAW .D2 states2,8,states2 ; SPINT .L1 temp3_i,temp3_i ;    SPINT .L2 temp3_iii,temp3_iii ; SPINT .L1 temp3_ii,temp3_ii ;    SPINT .L2 temp3_iv,temp3_iv ; SPINT .L1 temp1_i,temp1_i ;    SPINT .L2 temp1_iii,temp1_iii ; SPINT .L1 temp1_ii,temp1_ii ;    SPINT .L2 temp1_iv,temp1_iv ; ADD .S1 temp3_i,temp1_i,d_i ;    ADD .D1 temp3_iii,temp1_iii,d_iii ;    ADD .S2 temp3_ii,temp1_ii,d_ii ;    ADD .D2 temp3_iv,temp1_iv,d_iv ; INTSP .L1 d_i,d_i ;    INTSP .L2 d_iii,d_iii ; INTSP .L1 d_ii,d_ii ;    INTSP .L2 d_iv,d_iv ; </pre>
--	---

Figure 4.12: Linear assembler code fragment which implements a prediction lifting step.

instructions implement a prediction lifting step like

$$d_l^{(k+1)} = d_l^{(k)} + \left[ c_0 a_{l-1}^{(k)} + c_1 a_{l-2}^{(k)} \right] \quad (4.55)$$

where the brackets denote the rounding operation. An assembly code line has four columns: optional parallel bars, instruction, operation unit, operands. The parallel bars || indicate parallel execution of the instructions in this line and the previous line. Execution of one instruction happens in several phases, where its number differs from instruction to instruction. The MV instruction only needs one phase, while the LDW needs five phases. Therefore the constants loaded from c[1] is available in cst2 at the fifth parallel bar. Note, that the linear assembler is translated into the actual assembler, and during translation NOPs are inserted where needed. One way to optimize the code is to rearrange the instructions such that as few NOPs as possible are inserted. Because we process four channels in parallel we can make an efficient use of as many operation units as possible.

The equivalent C code which processed a single channel is

```

/* Step 1, predict */
cst = c[1];
temp1 = states[1];
states[1] = a;
temp1 = temp1 * cst;
cst = c[2];
temp2 = a;
temp2 = temp2 * cst;
temp1 = temp1 + temp2;

```



```
temp3 = states[2];  
states[2] = states[3];  
states[3] = d;  
temp1 = (int)temp1;  
d = temp3 + temp1;
```

A comparison of the C implementation and the linear assembler implementation reveals that the linear assembler is about four times faster than the C implementation. At this we measured the number of instructions needed for 64 sample points from 32 channels using the decomposition with Daubechies wavelet with three vanishing moments (db6) into 6 levels. The C implementation needs 714928 instructions while the generated linear assembler needs only 187780 instructions. This corresponds to 4.5ms and to 1.2ms for C and Linear Assembler implementation, respectively. Both are translated using the maximum optimization level (-o3) of the compiler and assembler. However, some optimizations were disabled, like e.g. non-interruptable code, because the code needs to be interruptible in the DSP application.

The real time implementation must process the data as fast as it is acquired. For example a wavelet decomposition of depth six requires 64 sample points and if we record at 50 kHz we have 1.28 ms between every new block. Since the DSP application needs some time to transfer data from the ADC modules into the CPU and then further into the ASRAM for subsequent transmission to the host, which sums up to about 0.33 ms, there is even less time. The generated implementation for db2 works at 50 kHz, while the implementation for db6 works up to 44 kHz sampling rate.

# Chapter 5

## Applying Wavelet Transform to Neural Signals

### 5.1 Low pass, band pass, and high pass wavelet filter

A wavelet filter bank can be used to implement low pass, band pass, and high pass filter. This is simply done by setting the coefficients to zero, which represent the frequency band, that should be suppressed. For instance setting the coefficients  $a_{L-1}$  to zero yields a high pass filter, and elimination of the coefficients  $d_0$  a low pass filter. One drawback of this method is that we cannot choose the cut off frequencies for the high pass, band pass, or low pass filters arbitrarily. Instead we are restricted to positions that are presentable as sampling rate divided by  $2^j$  ( $j \in \mathbb{N}, j > 0$ ). However, this restriction does not pose a problem for applications in neural recordings. In Figure 5.1 an example of a high pass filtered signal is shown. The lower trace is the original neural recording acquired at 5 kHz, while the upper trace is the result of high pass filtering using a Daubechies 2 wavelet filter bank of depth 4. The approximation coefficients which correspond to the frequency band from DC up to 312.5 Hz ( $= 5.000 \text{ Hz}/2^4$ ) are set to zero before reconstruction. This implements a high pass filter with cut-off frequency at 312.5 Hz. Note, that this filter nicely preserves the shape of the action potentials, as the wavelet filter bank based high pass has a nearly linear phase (see also Section 4.2.1).

### 5.2 Compression and Denoising

Denoising and compression are related due to the fact that denoising techniques can be used to implement lossy compression [11], or in other words we can use a lossy compression scheme which denoises the signal. To grasp the idea behind that, consider that a typical signal like e.g. a neural recording usually contains some structural correlation which in turn is exploited by a good compression scheme. But a neural recording does also contain noise. As noise, in particular white noise, does not have any structural redundancies, it is difficult to compress it effectively. However, denoising the signal, yields a signal which can be compressed more effectively.

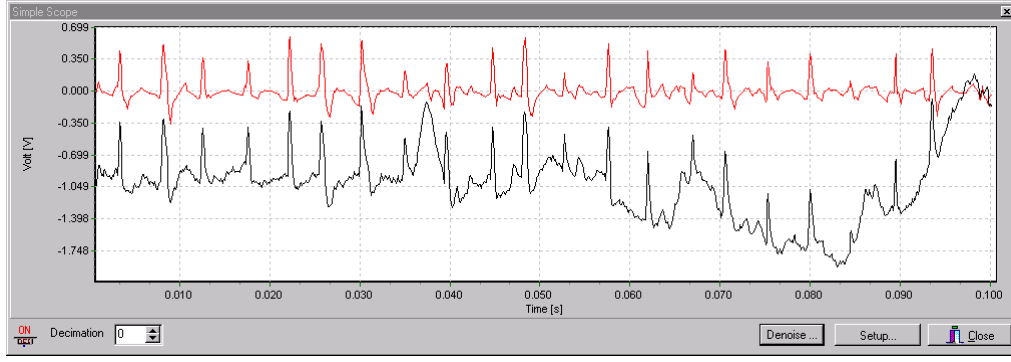


Figure 5.1: High pass filter implemented with a wavelet filter bank. The lower trace shows the original signal, and the upper trace the high pass filtered version. The frequency band 0-312.5 Hz has been discarded. Note, that the shape of the action potentials is nicely preserved.

The proposed compression scheme has three steps:

1. Decompose signal into wavelet coefficients
2. Denoise the signal, i.e. remove wavelet coefficients which represent the noise
3. Apply standard lossless entropy coder

The wavelet denoising works on the coefficients computed within a wavelet filter bank. The first step is the estimation of the standard deviation of the noise, which is supposed to have a Gaussian Distribution. We use the detail coefficients of the first step for this estimation, since these coefficients mainly represent the noise if noise is present. The standard deviation is estimated within a time window of given length  $N$  [22]:

$$\hat{\sigma} = \text{Median}(|d_{0,k}| : k \leq k < N) / 0.6745. \quad (5.1)$$

This median estimator is robust against outliers in the coefficients, that represent the signal instead of noise [52]. Note, that the constant 0.6745 makes the estimate unbiased for the normal distribution. All detail coefficients are divided by  $\hat{\sigma}$  and we obtain coefficients with normally distributed Gaussian noise. Using these coefficients we estimate the thresholds.

The VSAMUEL data acquisition system provides four different strategies to estimate the thresholds [23]. They are described in the following tabular. Thresholds are estimated within a time window and we denote the length of the window with  $N$ . The level is denoted by  $j \in 1, \dots, L$  and the threshold for a certain level with  $T_j$ .

MiniMaxi	$T_j = \begin{cases} 0 & N/2^j \leq 32 \\ 0.3936 + 0.1829 \log_2(N/2^j) & \text{otherwise} \end{cases}$
VisuShrink	$T_j = \sqrt{(2 \log(N))}$
Sure	Estimation of $T_j$ is based on Steins Unbiased Risk Estimate as described in [23]
SureShrink	Combination of Sure and VisuShrink where – based on a certain criteria – one of both strategies is selected. It is supposed to handle the case that the coefficients are sparse.

The threshold can be applied *hard* or *soft*. Hard thresholding means that all coefficients less than the threshold are set to zero and all others are kept.

$$\eta_h(x, T) = \begin{cases} x & |x| > T \\ 0 & \text{otherwise} \end{cases}$$

In case of soft thresholding [21] the magnitude of coefficients which are greater than the threshold is reduced by the threshold.

$$\eta_s(x, T) = \begin{cases} \text{sign}(x)(|x| - T) & |x| > T \\ 0 & \text{otherwise} \end{cases}$$

Note, that soft thresholding reduces the magnitude of all coefficients, thus in general we will find a reduced maximum peak to peak amplitude in the denoised signal.

The wavelet denoiser has two threshold estimation modes, i.e. either continuous or single. In continuous mode the thresholds are computed for each time window of length  $N$  samples, in single mode the thresholds are computed for the first  $N$  samples, i.e. the first time window, and then used for the subsequent windows without change. This is somewhat faster, but the result is not as good as in the continuous mode.

Figure 5.2 shows the result for different denoising methods applied to a neural recording with 16 bit at 50 kHz. We compare three different methods, i.e. MiniMaxi, Sure, and VisuShrink. SureShrink has been omitted because it is a mix of Sure and VisuShrink. The methods remove different amounts of noise from the original signal. Figure 5.2(d) shows a comparison between the methods in terms of entropy, which decreases with the amount of noise removed from the signal. VisuShrink and MiniMaxi remove about the same amount of noise, while Sure is less aggressive and more noise remains in the denoised signal. This is confirmed by the noise energy which is included in Figures 5.2(c), 5.2(a), and 5.2(b). Evaluation of the denoising quality by visual inspection shows that spikes are nicely preserved. Their shapes are hardly affected by the denoising.

The histograms of the removed noise in Figure 5.3 reveal a Gaussian distribution. The different widths of the histograms show also, that Sure removes less noise than VisuShrink and MiniMaxi.

The wavelet smoothness, which corresponds to the number of its vanishing moments, directly affects the smoothness of the denoising result. The smoother the wavelet the smoother is the denoised signal. Figure 5.4 shows an example. In Figure 5.4(a) a Daubechies wavelet with 6 vanishing moments has been used. The

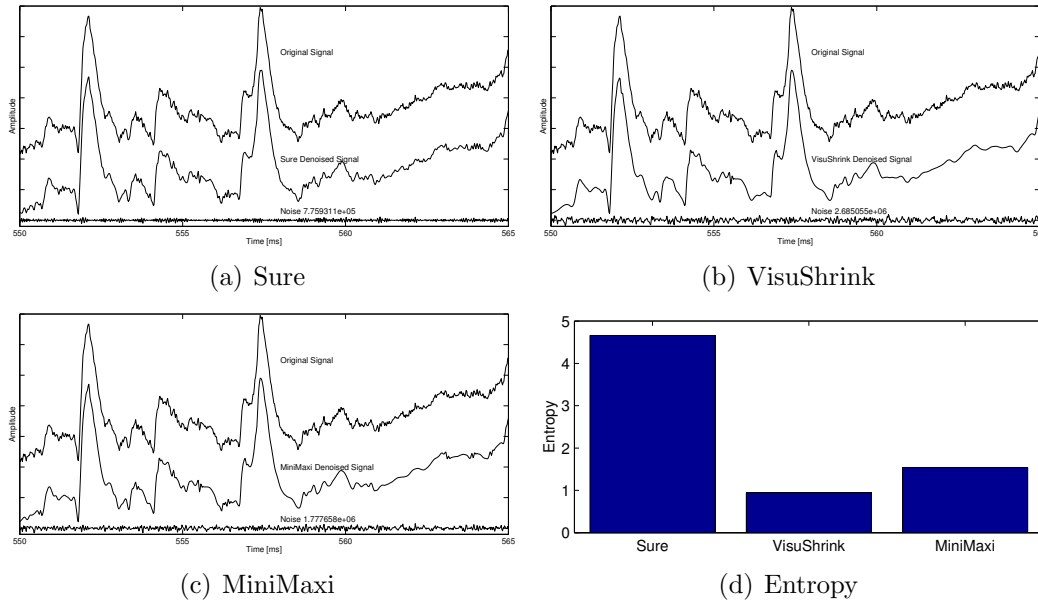


Figure 5.2: Denoising of a neural recording

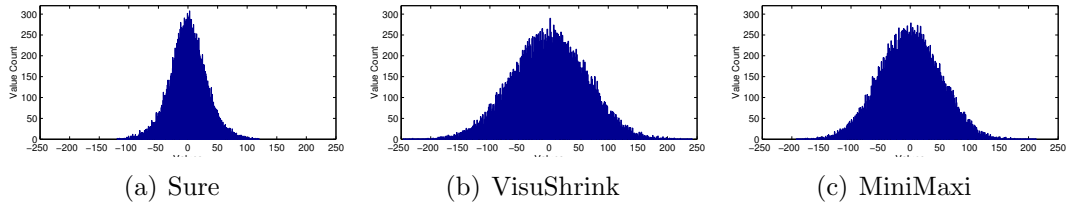


Figure 5.3: Histograms of extracted noise of a neural recording

result is significantly smoother, than the result in Figure 5.4(b) where a Daubechies 2 wavelet has been used.

Wavelet denoising depends on wavelet type, window length, threshold estimator, decomposition level. These parameters have different effects. The wavelet type does not significantly affect the denoising w.r.t to entropy, but it has an effect on the smoothness of the denoised signal. The three threshold estimation methods result in different entropies as they remove noise more or less aggressively.

### 5.3 Spike detection

Neurons exchange information by action potentials. Thus, it is of great interest to detect action potentials in neural recordings and to identify different neurons as their sources. If we put this information together, the processing done in the brain can be reconstructed on a rather low level.

There are different techniques to detect spikes, i.e. action potentials, in noisy neural recordings. Commonly used is the detection based on a threshold. However, we often face the problem of a low signal to noise ratio where the spike amplitudes

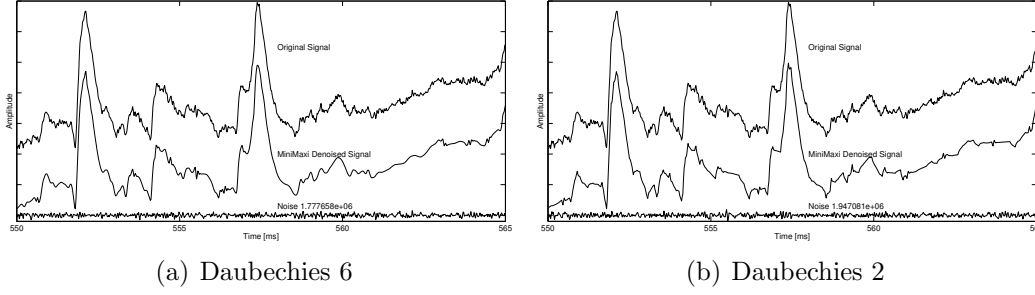


Figure 5.4: Comparison of denoising results using different wavelets

barely exceed the noise. This is particularly the case in multisite recordings, since the recording sites cannot all be placed close to a source of spikes, which would lead to a rather good signal to noise ratio.

The easiest approach suitable for recordings with good signal to noise ratio is a threshold which can be estimated using the root mean square and a scaling factor  $\alpha$ :

$$t_{\alpha, \text{rms}} = \alpha \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} (s(n))^2} \quad (5.2)$$

In order to make the thresholding more robust against single outliers, a minimum threshold pass count  $t_{\text{count}}$  is introduced. Thus, it is required that the signal must exceed the threshold at least for  $t_{\text{count}}$  successive sample points, before it is considered as a spike.

Spike detection can be further improved in order to deal with signals having a small signal to noise ratio. An example is the so called Non-linear Energy Operator (NEO) [80]. This operator is defined as

$$e_{\text{neo}}(n) = (s(n))^2 - s(n+1) * s(n-1) \quad (5.3)$$

The magnitude of the NEO value is large for fast changes in  $s(n)$ . Figure 5.3 illustrates the difference between spike detection with and without NEO. In both cases the neural input signal has been high pass filtered with a wavelet filter bank. Afterwards the root mean square has been computed and it is shown as a horizontal line in both plots. It appears that the NEO allows an easier detection compared to the plain high pass filtered signal.

## 5.4 Spike sorting

Spike sorting is a process where detected spikes are sorted into different classes. Each class is supposed to collect spikes issued by a certain single neuron. It is necessary if a neural recording contains spikes from different neurons that exceed the neural background noise. The classification is based on the recorded shape of each spike. Spike shapes are obtained from the recorded signal by spike detection. The signal might be preprocessed by a high pass filter to eliminate low frequencies as spikes are

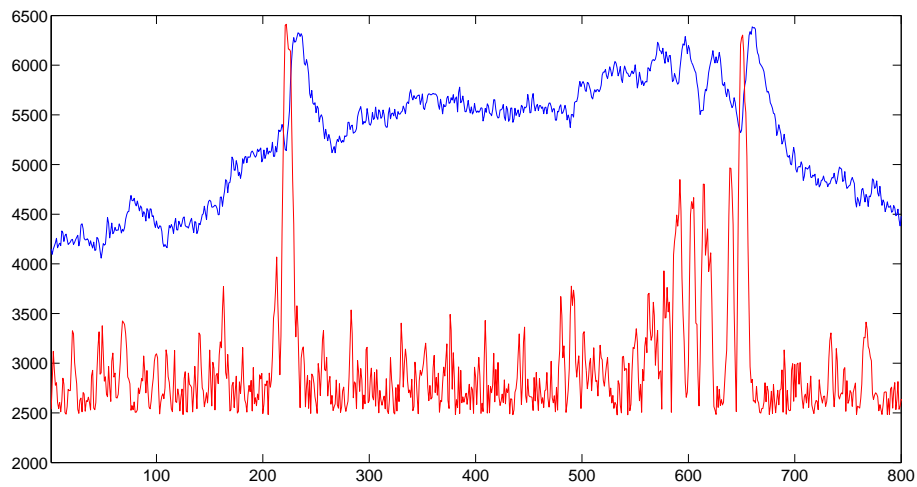


Figure 5.5: NEO operator applied to neural recording

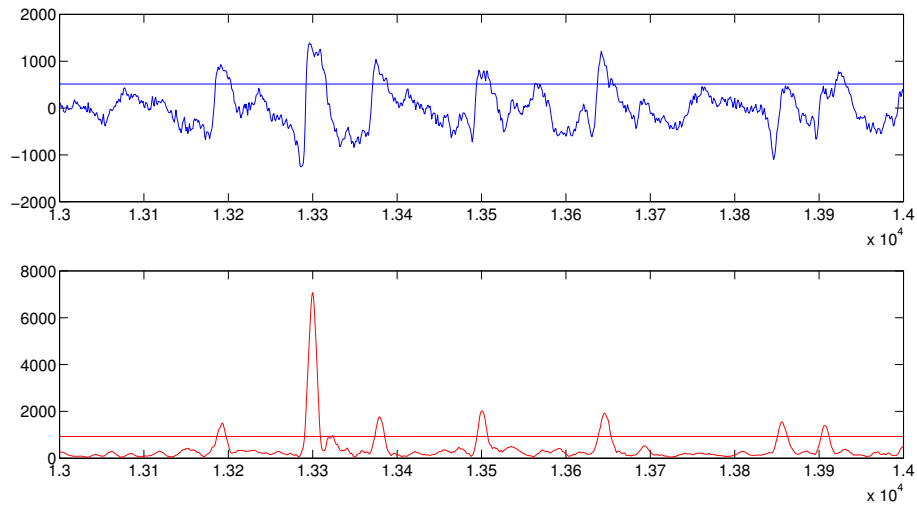


Figure 5.6: NEO operator applied to neural recording, which has been high pass filtered using a db6 wavelet filter bank with depth 6 (see Section 4.2.1).

in the frequency range from 500 Hz to 8 kHz. A second preprocessing step could be a denoising as described above in Section 5.2.

Spike sorting algorithms assume that the recorded versions of spikes issued by a single neuron have a characteristic shape. In principle, that is correct. However, several parameters influence the spike shape and the actual recorded version. There can be artefacts due to a moving electrode, or a changed position of the electrode or the tissue might move. Sometimes pulse artefacts disturb the recording. Furthermore, the spike shape depends also on the previous activity of the respective neuron. A neuron which is firing spikes rapidly in so called bursts produces spikes with changing shape, because the amplitude decreases and the width increases. The activity of two neighbouring neurons could overlap and cancel each other out. Also, the shape of spikes issued by two similar neurons might be also quite similar, so that they cannot be sorted into separate groups. With respect to these difficulties it is not possible to provide a spike sorting algorithm which works reliable in all cases. A whole lot of methods have been proposed and a review of some methods can be found in [69].

Basically, spike sorting is performed in two steps. The first step is the feature computation and the second step is the classification. Usually finding the discriminating features of the spike shapes is the main problem. Having found good discriminating features a simple clustering method like e.g. the k-means algorithm [41] suffices for the second step. The result of the sorting can be visually supervised easily, if the used features form distinct clusters in feature space.

When selecting a method for spike feature computation the experimental setting must be considered. If the probe has recording sites, which are very close to each other, these pick up similar but not equal signals. The differences are due to slightly different relative positions of the emitting neurons and the recording sites. With such probes multiple shapes for the same spike are obtained and the differences among these shapes can serve as features. In [40] the spike sorting performance using recordings from single electrodes, stereotrodes, and tetrodes are compared. They found that it improves the more recording sites are used, i.e. tetrodes are better than stereotrodes, which in turn are better than single electrodes.

Some of the fork shaped silicon probes described in Section 6.1.1 (e.g. U3) feature a layout where the recording sites are placed in pairs such that they effectively work as multiple stereotrodes. If these probes are used, stereotrode specific spike features can be utilized for the spike sorting. In other designs the recording sites are placed so far apart from each other, that they do not pick up spikes from the same neurons. In this case the signal from each recording site must be treated like a single electrode recording.

### 5.4.1 Spike features

The starting point for spike feature computation is usually a set of spike shapes extracted from a neural recording. Some commonly used features in time domain include for instance:

- Minimum amplitude,



- Maximal amplitude,
- Peak to peak amplitude,
- Temporal distance between minimum and maximal peak, and
- Temporal distance between zero crossings.

These simple descriptive features work quite well for neural recordings where the spikes have large amplitudes, with high signal to noise ratio. However, they reduce the full shape of spike to certain keypoints which might not be significant for discrimination. In general it is better to consider the complete shape of the spikes for discrimination. A widely used approach is to use the principal component analysis in order to find the discriminating features for a given set of spike shapes. This approach is outlined in Section 5.4.2. However, the principal component analysis fails to classify overlapping spikes correctly. In best case these fall into a separate class which collects all unspecific spike shapes.

Another approach which also accounts for the full shapes is based on Bayesian modeling of the spikes [68]. At this a probabilistic model for the neural recording is defined to compute the probability of both form and number of spike shapes. Furthermore, according to [68] the Bayesian approach is also able to analyse overlapping spike shapes.

Other approaches utilize wavelet based time frequency analysis in order to obtain discriminative features [67, 53]. In [67] a wavelet based approach is compared with principal component analysis (PCA, see Section 5.4.2) and reduced feature set (RFS, [88]), which are alternative methods to obtain features for spike shapes. In the described setting the wavelet based approach outperforms PCA and RFS. The spike shapes are decomposed by a dyadic wavelet filter bank. Then the sets of coefficients with largest mean value and largest standard deviation and – most preferable – having a bimodal or multimodal distribution over the population of spike shapes are selected manually by visual inspection. The classification is done manually through a software interface which allows to delimit the clusters interactively.

The approach in [53] utilizes a wavelet packet decomposition. This transform divides the frequency space into bands of equal width, the so-called wavelet packets, allowing a better localization in frequency than the wavelet decomposition. A set of wavelet packets is selected by using e.g. the best basis algorithm with Shannon’s information cost function and is used for detection and spike sorting. According to [53] this method is more efficient both in separating spikes from noise and in resolving overlapping spikes than methods like PCA and wavelet decomposition.

### 5.4.2 Principal component analysis

The spikes are extracted from the original signal by spike detection and then aligned in time for example by aligning their maximum amplitude. As an optimization it is possible to perform an upsampling of the extracted spike snippets in order to align them more precisely, then the alignment resolution can be higher than the actual sampling resolution. For principal component analysis (PCA) each spike

snippet is regarded as a feature vector. We define a matrix  $A$  which rows contain the feature vectors, i.e. the spike snippets. Each column in matrix  $A$  represents a variable in feature space. The PCA computes the eigenvectors and eigenvalues of the covariance matrix of  $A$ . They are sorted with respect to the magnitude of the eigenvalues which corresponds to the variance of variables in the original feature space, i.e. the first principal component vector represents the variables with the largest variance. The PCA results in a new basis for the feature space, which allows a compact representation of the feature vectors. The first few, i.e. three or four, coefficients of the new representation are the new feature values, thus the PCA has reduced the dimensionality to three or four dimensions. Using these feature values, standard clustering methods can be applied to separate different classes of spike snippets.

The PCA requires the complete set of feature vectors to be available. Therefore, it is not well suited for online feature computation.

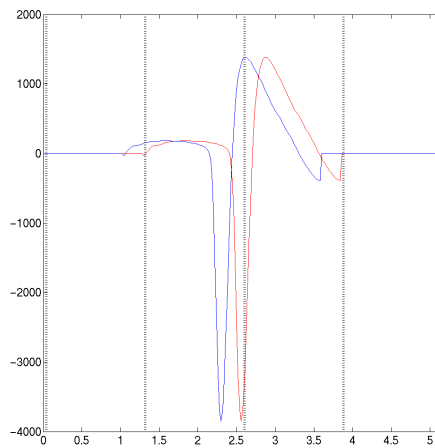
### 5.4.3 Online wavelet based spike sorting

The online wavelet based spike sorting is supposed to classify the spikes immediately after acquisition, in order to give the experimenter a direct feedback about the contents of the recorded signals. At the time of classification the complete signal is not available. This prevents us to utilize some classification methods, which require the complete signals containing the spikes, which shall be classified, as for instance the principal component method. Instead the signal is decomposed by a wavelet filter bank and the wavelet coefficients are used to compute feature vectors.

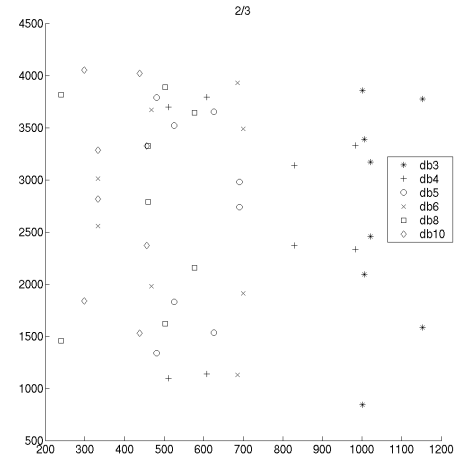
Given the wavelet based approach there are two main problems to deal with, i.e. translation variance and quantization errors, which are the subject of the following paragraphs.

#### Translation variance

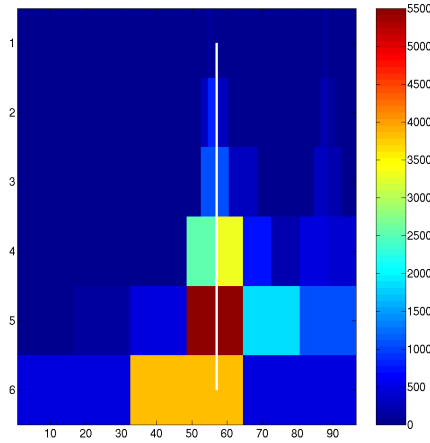
Figure 5.7 reveals the most prominent problem using wavelet based features for classification if the spike snippets are not aligned. Figure 5.7(a) shows the same exemplary spike snippet at two different positions relative to the wavelet decomposition raster which is indicated by the vertical dotted lines. The red (right hand) is delayed by 0.26 ms versus the blue (left hand) spike, which corresponds to 13 sample points at 50 kHz. The scalograms in Figure 5.7(c) and Figure 5.7(d) illustrate the translation variance in a direct comparison of the wavelet coefficients resulting from a decomposition with the Daubechies 3 wavelet. A scalogram indicates the magnitude of the wavelet coefficients in a two dimensional plot. The different scales are plotted versus the time on y- and x-axis, respectively. The coefficients magnitude is mapped onto a color according to the given colorbar. The white line in the scalogram plots indicates the reference position of the detected spike. A comparison of the values in level 3 for instance reveals a large difference although both spike only differ in their relative position to the decomposition raster. Consequently the translation variance leads to great differences in the final feature space. For Figure 5.7(b) the feature values have been computed as the absolute sum of the wavelet coefficients



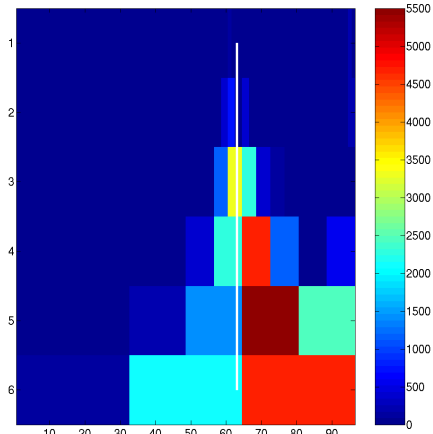
(a) Spike snippets



(b) Absolute sum of wavelet coefficients



(c) Scalogram of blue spike



(d) Scalogram of red spike

Figure 5.7: Spike snippets, wavelet based features, and scalograms of the same single spike snippet at different translations

within the different levels. Here the levels 2 and 3 are plotted versus each other. They have been computed based on different wavlets as indicated in the legend of Figure 5.7(b) for the same spike snippet at eight different translations. Ideally, the marks with the same symbol should be located at the same point in feature space, because they all represent the same spike. However, this is not the case. Instead they are spread over a large range of values. Thus, a precise classification based on these feature values is hardly possible. Only spikes that differ a lot in time domain, can be separated by these feature values.

In order to handle the translation variance problem, we need an exact reference position in time domain to determine the shift of the spike with respect to the decomposition raster. Therefore, the spikes are detected in time domain using the

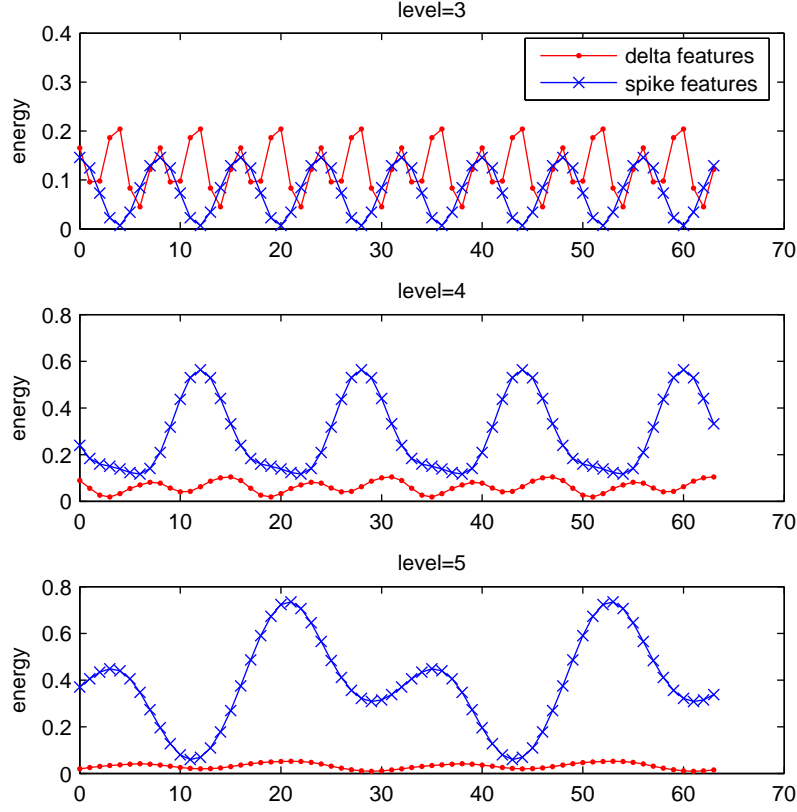


Figure 5.8: Translation variance of the energy feature

NEO operator (see also Section 5.3), as it turns out that a robust detection of the exact reference position in wavelet domain does not work well. This is because of the coarse time resolution in frequency bands where the spikes lead to large coefficients.

Generally, the translation variance of a dyadic filter bank is periodic with period  $M$ , i.e. it is translation invariant for translations  $n$  and  $n + M$ . The period  $M$  depends on the decomposition depth  $L$  as  $M = 2^L$ . Therefore, there are  $M$  different decomposition variants for a given signal  $s(n)$ . Two approaches were examined to tackle the translation variance problem:

#### 1. Normalization of translation variant wavelet coefficients

Compute all  $M$  decomposition variants of a fixed normalization signal  $s(n) = \delta(n)$ . So for each translation  $m = 0, \dots, M - 1$  we have a decomposition variant  $D_m$ . Given the reference position  $p$  of a spike we can determine the corresponding decomposition variant  $D_m, m = p \bmod M$ , which contains the normalization coefficients.

Figure 5.8 shows the distribution of energy over three different levels of the wavelet decomposition using the  $\delta$  function and a spike template as input signal. 64 instances of both  $\delta$  function and spike template have been decomposed by a Daubechies wavelet with 6 vanishing moments into 6 detail levels. The instances were all shifted differently against the decomposition raster. The

respective shift is indicated on the X-axis. The energy distribution varies periodically, where the period depends on the decomposition level  $l$  with  $2^l$ . The energy of the spike template has been normalized to one. However, the figure reveals that there is no systematic translation dependent variation of the energy distribution over different decomposition levels. Therefore, it appears impossible to utilize the energy distribution of the  $\delta$  function to normalize the translation variant decomposition of a given spike instance.

## 2. Translation dependent classification

The classification is done with respect to the translation, i.e. only spikes with the same translation are compared with each other. Thus, the classification is translation invariant, but this results in a separate feature space for each translation  $R_m, m = 0, \dots, M - 1$ . Each feature space contains cluster which must be mapped onto each other.

For a classification with a training dataset this mapping is easy, since the classes of the cluster in  $R_m$  are known a priori. For automatic classification without training data, we must find the corresponding clusters  $C_0$  in  $R_{m_0}$  and  $C_1$  in  $R_{m_1}$ . This could be achieved by calculation of the mean spike in cluster  $C_0$ , which is then transformed into the feature space  $R_{m_1}$  by translation and wavelet decomposition. The corresponding cluster  $C_1$  is the cluster which is next should be the desired cluster. However, it is possible that for certain translations two cluster overlap, making it impossible to find a suitable mapping and then the classification problem cannot be solved.

Approaches to overcome the translation variance problem are described in [7]. The simplest solution is to use an undecimated variant of the wavelet transform, which is called the *algorithme à trous* [73]. This algorithm is very similar to the fast wavelet transform. The only difference is, that instead of sub-sampling the data the wavelet filter are dilated by insertion of zeroes. This leads to shift invariant coefficients, but at the cost of an increased computational complexity regarding operations as well as memory usage which depends on the depth of the decomposition.

Another approach is the dual-tree complex wavelet transform (DTCWT), which was introduced with [63] and for which [92] gives an elaborated overview. The DTCWT is an enhancement of the traditional real wavelet transform. In particular, it is approximately shift invariant while it can be implemented based on the same decomposition and reconstruction algorithms as the traditional wavelet transform. As it uses two decomposition filter banks it needs about 2 times more operations and also 2 times more memory resources for the result. The wavelets in the two trees are Hilbert transforms of each other [92], i.e. they are  $90^\circ$  out of phase. Therefore, the wavelet coefficients from both decomposition trees interpolate each other. A single complex wavelet coefficient is composed from the two corresponding real value wavelet coefficients from both trees. Furthermore, the use of two trees reduces the effect of the subsampling, i.e. aliasing is greatly attenuated. The method relies on specially designed wavelets in order to achieve the shift invariance. It is not possible to just use e.g. the standard Daubechies wavelets.

**Quantization errors**

Massive quantization errors occur if a 16 bit input signal is decomposed into 16 bit integer wavelet coefficients. The method introduced in Section 4.3.2 adds a rounding to each lifting step, i.e. in each lifting step a quantization error occurs. These errors sum up to a large error. Changing the scale factor as the final step of the lifting in order to avoid 16 bit overflows also increases the quantization error. A solution is to double the precision for the wavelet coefficients and use 32 bit floating point values. The drawback is, that the wavelet coefficients need twice as much space.

# Chapter 6

## Data Acquisition System

### 6.1 Hardware

The VSAMUEL data acquisition system [32, 43] consists of several components. Figure 6.1 gives an overview in case of 128 channel recording setup. Signals are acquired with multichannel probes in the nervous tissue. The small amplitude of these signals is amplified in a preamplifier stage which has a gain of 16 (24 dB) per channel. It uses a battery as DC power supply. The preamplified signals are routed through the battery pack to the main amplifier which resides outside the recording area in a 19 inch rack together with the data acquisition host. The main amplifier has ten individual gain settings per channel. Both amplifier stages together provide gains ranging between 80 and 80,000 times. The data acquisition host has eight AD16 boards which are in turn mounted on four M67 boards. Each analog channel has an own analog to digital converter which resides on one of the AD16 boards. Analog digital conversion is possible with rates between 5 kHz and 50 kHz at 16 bit resolution. The CPU on the M67 board performs some signal processing tasks before the data is sent to the host CPU where it is visualized and stored to hard disk.

#### 6.1.1 Probes

The probes are manufactured by Acreo AB (Kista, Sweden) using micro structure technology as described in Section 3.5. Basically they are fork shaped silicon with linearly arranged recording sites on each shaft. With respect to different target areas, there are several different probe layouts. As a first step probes with 32 sites were developed. Figure 6.1 summarizes the technical data of the eight designs and Figure 6.2 shows exemplarily the design of the E3 probe. In a second step the number of recording sites was increased to 64. Some 64 channel probe designs are identical to the 32 channel version but with double recording sites. Table 6.2 give an overview of the different specifications. Figure 6.3(a) for instance shows a probe specially designed for recording from cerebellum. It has double recording sites on 4 mm shafts, where the recording site spacing of 200  $\mu\text{m}$  between the three double recording sites next to the shaft tips and a 100  $\mu\text{m}$  between the third and the fourth double recording site. It covers the Purkinje cell layer and the molecular layer with dense spacing and has a wider spacing to cover the granular and medullary layer

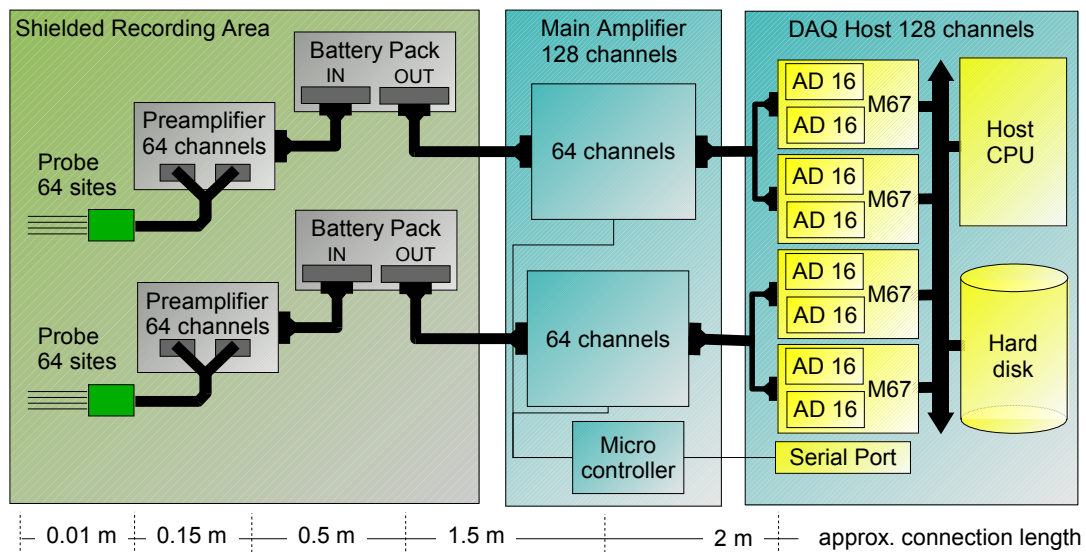


Figure 6.1: Data acquisition system hardware setup. Each of the two 64 channel probes is connected to a 64 channel preamplifier. The signals are routed through a battery pack, which provides power to the preamplifiers, toward the 128 channel main amplifier. The fully amplified signals are digitized by eight AD16 board mounted on four M67 boards which reside in the host computer. The host computer merges the digitized data and stores it on hard disk. All signal connections from preamplifier to data acquisition computer are established by 68 pin SCSI cables.

(Figure 2.4).

### 6.1.2 Amplifier

The amplifier stages, i.e. preamplifier and main amplifier, were designed and manufactured by Thomas Recording GmbH, Gießen. Key features of the preamplifier are its small size, low noise, high precision, and wide bandwidth. They are summarized in Table 6.3. The input connector is a 'zero insertion force' Molex 52559 ZIF connector ([www.molex.com](http://www.molex.com)) socket for flexible PC board cables as used by the VSAMUEL probes (see Figure 6.3(b)). The output connector is a 68 pin SCSI socket which is connected with the power supply, i.e. an accumulator device. The preamplifier uses a DC power supply in order to avoid line noise within the preamp circuits. Being DC coupled its lower cut-off frequency starts at 0.034 Hz. Beside the 64 channel version there is also a 32 channel version of the preamplifier which is only half the size and suitable for experiment setups with strong space constraints.

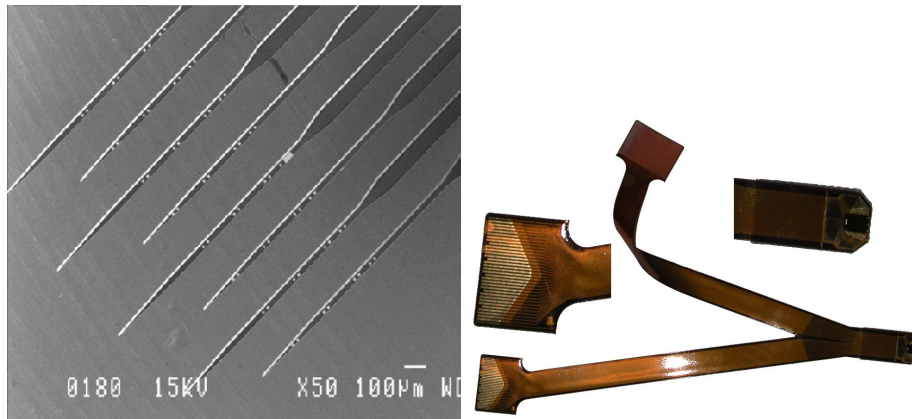
The main amplifier consists of two 64 channel amplifier segments (Figure 6.3(d)). Each segment resides in a 19 inch rack with eight rack units. The gain settings of the main amplifier are configured by a microcontroller which has a front end at the front panel of the main amplifier which enables an user to manually set the channel gains. Additionally, the microcontroller is connected to one of the serial ports of





Design name	Number of shafts and sites	Shaft length (L) $\mu\text{m}$	Site c/c distance (a) $\mu\text{m}$	Shaft c/c distance (b) $\mu\text{m}$	Shaft width at active region (w) $\mu\text{m}$
M3	2 shaft $\times$ 32 sites	7000	50 (ss)	1 000	Increasing from 0 to 100
M4	4 shafts $\times$ 16 sites	10000	50 (ss)	600	38
E4	4 shafts $\times$ 16 sites	5000	100 (ds)	400	38
E5	8 shafts $\times$ 8 sites	4000 and 3650	200, 200, 100 (ds)	200	25.5
E6	4 shafts $\times$ 16 sites	5000	100 (ss)	400	38
K2	4 shafts $\times$ 16 sites	8000	100 (ds)	500	38
U3	3 shafts $\times$ 16 sites + 1 shaft $\times$ 14 sites (= 62 sites)	10000	100 (ds)	250	38
U4	8 shafts $\times$ 8 sites	5000	50 (ss)	60	25.5

Table 6.2: Specification of VSAMUEL probes with 64 recording sites. Shaft length L, site center/center distance a, shaft center/center distance (b) and shaft width at active region w are drawn into Figure 6.2. (ds) indicates double recording sites and (ss) single recording sites.



(a) Scanning electron micrographs of a 64-site neural probe, model E5. Eight staggered shafts with eight electrodes 69 mm each. (b) 64-channel flexible printed circuit manufactured by HP Etch (138 mm x 69 mm).

Cabinet size	93 mm× 63 mm× 35 mm
Voltage noise density	7.5 nV/ $\sqrt{\text{Hz}}$ (at f=1 kHz)
Current noise density	0.5 fA/ $\sqrt{\text{Hz}}$
Lower cut-off frequency	0.034 Hz
Upper cut-off frequency	42 kHz
Gain per channel	×16 (24 dB)
Gain precision per channel	±0.5% (39.8 ... 40.2)
Input impedance	1 G $\Omega$ /12 pF
Power supply	±2.5V DC (Accu)

Table 6.3: Key features of the 64 channel preamplifier



(c) Preamplifier and power supply



(d) Main amplifier

Figure 6.3: Pre and main amplifier for 64 channels

the host. Via this connection the gain settings can also be configured by software running on the host. The main amplifier supports ten different gains which result from different combinations of pre-filter gain 1, pre-filter gain 2, and post filter gain (see Table 6.4). Considering both preamplifier and main amplifier stage gains from ×80 up to ×80,000 can be achieved.

### 6.1.3 AD conversion

An integral and essential part of the data acquisition hardware is the conversion of analog input (voltage) provided by the main amplifier into digital values. This step is called analog to digital conversion and is performed by analog to digital converters (ADC). A great variety of ADCs is available today. They differ in resolution, bandwidth, accuracy, architecture, packaging, power requirements, and temperature range. When choosing an ADC for a certain application these features should be taken into account.

Most data acquisition systems for neural data employ 12 bit or 14 bit ADCs. In order to record a broadband signal which contains action potentials and field potentials on one channel instead of using separate channels for both frequency bands as it is usually done, a higher ADC resolution is needed. With 16 bit resolution it is still possible to detect and classify action potentials superimposed onto field

Cabinet size	16 RU $\times$ 19 inch $\times$ d cm
Voltage noise density	21 nV/ $\sqrt{\text{Hz}}$ (@ Gain=5k, BW=16kHz)
Lower cut-off frequency	0.060 Hz
Upper cut-off frequency	16 kHz
Pre-filter gain 1	$\times 1$ , $\times 2$ , $\times 5$ , $\times 10$
Pre-filter gain 2	$\times 1$ , $\times 10$ , $\times 100$
Post filter gain	$\times 5$
Total gain	$\times 5$ , $\times 10$ , $\times 25$ , $\times 50$ , $\times 100$ , $\times 250$ , $\times 500$ , $\times 1000$ , $\times 2500$ , $\times 5000$
Gain precision	$\pm 1.5\%$
CMRR	114 dB(@f=1kHz)
Input impedance	2.2 M $\Omega$ /10 pF
Power supply	integrated

Table 6.4: Main amplifier features

potentials, while 12 bit or 14 bit are not sufficient. Thus, the VSDAQ system employs 16 bit ADCs.

For sake of simplicity we overestimate the maximum peak to peak amplitude of action and field potentials and suppose that their values are about 200mV and 4000mV, respectively. Therefore, with 12 bit AD conversion the resolution is about 1mV per bit, while 16 bit AD conversion results in 0.06mV per bit. In the 12 bit case the action potential are effectively sampled with 7.6 bit, while we have 11.7 bit in the 16 bit case.

The most popular ADC architectures include [5]:

- successive approximation (SAR) – a successive approximations shift register is the key defining element,
- flash – all decisions are made simultaneously,
- pipelined – multiple flash stages.
- sigma-delta ( $\Sigma\Delta$ ) – a charge-balancing type

These architectures mainly differ with respect to number of comparators and number of comparison cycles. The plain flash architecture requires one comparator for each digital output value, e.g. a 4 bit flash ADC requires 16 comparators. It can convert an analog input within a single cycle. This architecture is not suitable for high resolutions, as it requires e.g.  $2^{16} = 65,535$  comparators in case of a 16 bit ADC. The number of comparators can be reduced when using cascaded flash ADCs. This architecture is referred to as pipelined. Each stage can consist of a  $m$  bit flash ADC and  $p$  such stages establish a  $n = m \times p$  bit ADC, which only needs  $2^m \times p$  comparators. Sort of orthogonal to the flash architecture is the SAR architecture, which uses only a single comparator, but needs an additional cycle for each bit of resolution.

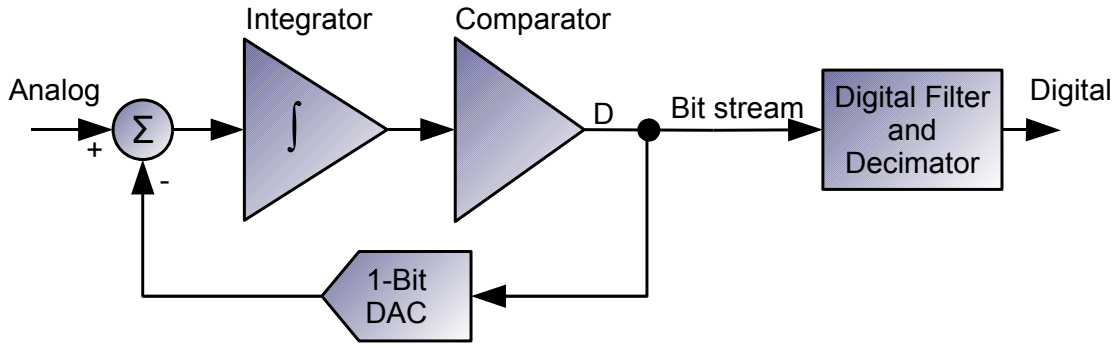


Figure 6.4: Basic concept of sigma-delta converter, [1]

In the VSDAQ system ADCs built according to the sigma-delta architecture, which takes another approach, are employed. Basically, it uses an integrator, a comparator, and a single bit DAC combined as shown in Figure 6.4.

Given a reference voltage  $V_{ref}$  the output of the digital analog converter (DAC) is either  $-V_{ref}$  or  $+V_{ref}$ , depending on the comparator output, which is either 0 or 1. The output of the DAC is subtracted from the input voltage  $V_{in}$  and the result is the new input to the comparator. The comparator output is 1 or 0 for input voltage greater or less than ground level, respectively. The output of the comparator is a sequence of zeros and ones. This sequence represents the input voltage  $V_{in}$  in the following manner: If  $V_{in}$  is closer to  $+V_{ref}$  the sequence will contain more ones than zeros and if  $V_{in}$  is closer to  $-V_{ref}$  the sequence will contain more zeros than ones. With  $V_{in}$  being close to midscale the sequence will have approximately the same number of zeros and ones. A digital filter can be employed to obtain a sample value from the sequence.

The ADC hardware used in the VSDAQ system is a set of so called AD16 boards manufactured by Innovative Integration. A single AD16 board features 16 sigma-delta ADCs (Analog Devices, AD7722), where each ADC is capable of sampling with 16 bit resolution at a rate of 195.3kHz. In the VSDAQ system they are, however, used up to a rate of 50kHz. The analog input is continuously sampled by a sigma-delta modulator which eliminates the need for external sample-and-hold circuitry. The modulator output is processed by two FIR digital filters in series. Due to this on-chip filtering external antialias filters only need to be of first order. The filter introduces a group delay of 215.5  $\mu$ s. The ADCs within a single AD16 board are synchronized such that data can be simultaneously acquired from all 16 channels.

The AD16 board has a 32 bit architecture, i.e. the smallest data unit which can be handled has 32 bits. As a consequence the 16 bit AD converters can only be enabled or disabled in pairs. Channel 0 and 1, 2 and 3, etc. form such a pair. However, these pairs can be controlled independently.

Conversion data is transferred from the ADC pairs into a First In First Out buffer (FIFO) on the AD16 board that can hold up to 512 samples or 256 samples pairs. The FIFO is connected to the DSP via a data bus on the M67 board (Omnibus). Additionally the FIFO uses one of the four external interrupt input lines of the DSP.

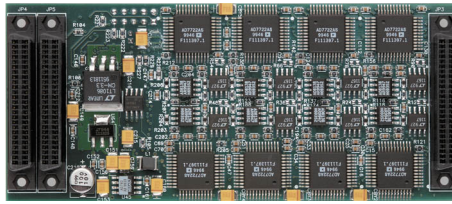


Figure 6.5: AD16 board

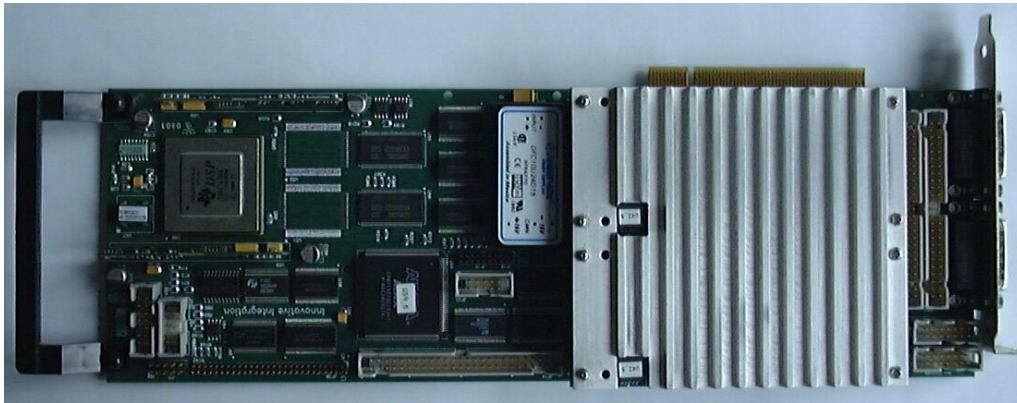


Figure 6.6: M67 board equipped with AD16 and its installed heat spreader

Via this interrupt line the FIFO notifies the DSP if its fill state exceeds a certain configurable threshold. The current 32 bit value of the FIFO is mapped into the DSP's memory space. A read access on this location by the DSP or by its DMA controller causes the FIFO to put the current value on the bus, and afterwards the FIFO's tail pointer is moved to the next value.

#### 6.1.4 Digital signal processor board M67

The digital signal processor board M67 is manufactured by Innovative Integration (Thousand Oaks, California, USA). Its core is the Texas Instruments C6701 digital signal processor. In addition to the two on-chip timers it has three 16 bit timers in FPGA logic and a 32 direct digital synthesizer (DDS) timer from Analog Devices (AD9850). The board provides a modular IO expansion system, i.e. two slots for expansion boards. In the VSDAQ system both slots are used by the AD16 board which perform the analog digital conversion. With the digital IO port 32 bit of digital input or output are available. The board has JTAG hardware emulation support, which makes it possible to perform realtime debugging using Texas Instruments IDE Code Composer Studio. The board interfaces with the host PC via the PCI bus, with a total bandwidth of 130 MB/s for fast digital data transfers. The M67 can perform busmaster transfers from or into a shared memory area on the host PC.

**Texas Instrument C6701**

The processor is clocked at 160 MHz by the M67 board. It has two sets of 16 general purpose 32 bit registers, called A0-A15 and B0-B15. They can be used for data, data address pointers, or condition registers. Additionally, two registers can be paired in order to perform operations on 40bit or 64bit data values. Associated with each register set is a set of four functional units. Each unit can perform different operations. However, their functionality overlaps, i.e. multiple units are able to do a 32 bit add operations. All functional units can virtually work in parallel. Thus, the C6701 is e.g. able to perform six 32 bit add operation at a time when using both register sets. The functional units can write their results only into registers of their associated register set. The input data can be read from each register of the associated register set, but also via one of the two crosspaths from the opposite register set. There are only two crosspaths, hence two of such accesses per cycle are possible.

Program instructions are executed on the C6701 in three pipeline phases:

- Fetch with four stages,
- Decode with two stages, and
- Execute with up to five stages.

During the fetch phase up to eight parallel instructions are transferred from the program memory. In the decode phase these instructions are dispatched onto the functional units and the instructions are decoded. The execution phase has up to five stages. Hereby different operations use different numbers of these execution phases. An add operation e.g. needs only one execution step, while a load data from memory operation needs all the five steps.

These features are useful for implementation of signal processing algorithms which usually apply the same sequence of operations to a large set of data values. The optimized implementation of the lifting scheme is an example on how to use these features. For details see Section 4.3.3

Block data transfers are one of the most frequently used operations. The C6701 has a *Direct Memory Access* (DMA) controller which can conduct such operations in parallel to the CPU. Therefore the CPU can be disburdened from performing block transfers by utilizing the DMA controller. Moreover, the DMA controller can operate with different configurations, e.g. reordering the data or extracting only certain elements of a block. Figure 6.7 a shows a scheme of the DMA controller. There are four independent channels, each having an own set of registers for source address, destination address, and configuration bits. The global registers are used for address reload, count reload, and different addressing modes. A DMA transfer is organized in a sequence of frame transfers. Each frame has a certain amount of elements. Both values are set in the count register of the respective DMA channel. The DMA controller uses different address increments or decrements after element and frame transfers. This mechanism is used to merge the data from both AD16 boards into a single buffer.



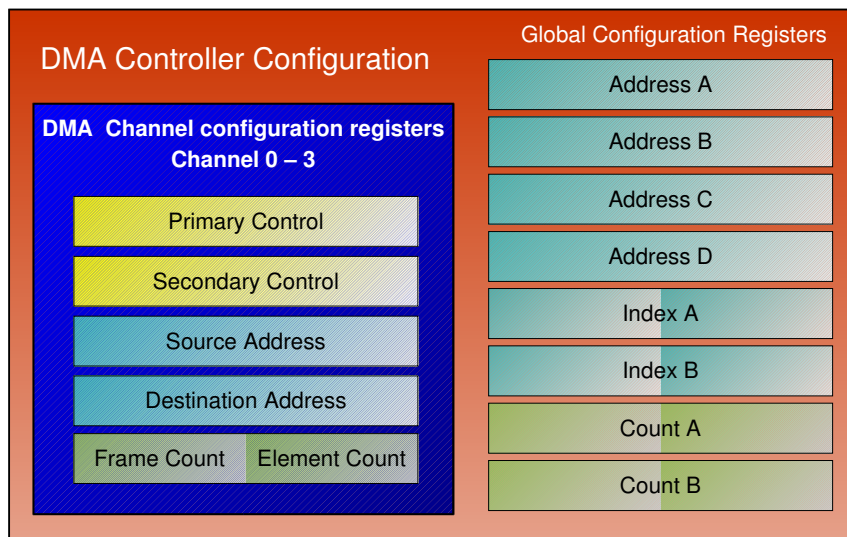


Figure 6.7: DMA controller configuration registers

The data acquisition board for 32 channels is shown in Figure 6.6 and an schematic overview in Figure 6.8. The M67 board is equipped with two AD16 boards. The AD16 boards are synchronized with the AdcSync cable (see also Appendix B.2). Data acquired by AD16 boards is stored in the respective FIFO and after a certain amount of samples are acquired each AD16 board triggers an interrupt which in turn initiates a DMA transfer which transports the data into the internal RAM of the DSP. If required the data can be processed and afterwards another DMA transfer copies the data into ASRAM. From ASRAM the data can be transported into a shared memory section on the host PC. This transfer is conducted by a separate controller which works independent from both host CPU and DSP board CPU. A finished transfer is signaled by interrupts. Note, that transfers from AD16 to DSP and from DSP to ASRAM both use the same bus, i.e. they cannot be performed simultaneously. This is taken into account, by triggering transfers into ASRAM, just after a transfer from AD16 boards has finished. This way there is no simultaneous usage of the bus and all transfers can be performed as fast as possible.

### 6.1.5 Synchronization

There are two means of synchronization one for the M67 boards and one for the AD16 boards. Multiple M67 boards are synchronized with a master-slave mechanism. The master board provides its clock signals to the other boards, i.e. the slave boards. Hereby the slave boards are connected to a clock bus, which is driven by the master board. For details refer to [56] or Appendix B.1. Since the ADCs on the AD16 board are driven by a clock with 128 times the sampling rate, the transmitted clock rate is about 6 MHz. The other synchronization is necessary among the ADCs of the AD16 boards. Each AD16 board provides a SyncIn pin and a SyncOut pin these. We connect SyncOut of AD16 board 0 with SyncIn of AD16 board 1, and continue



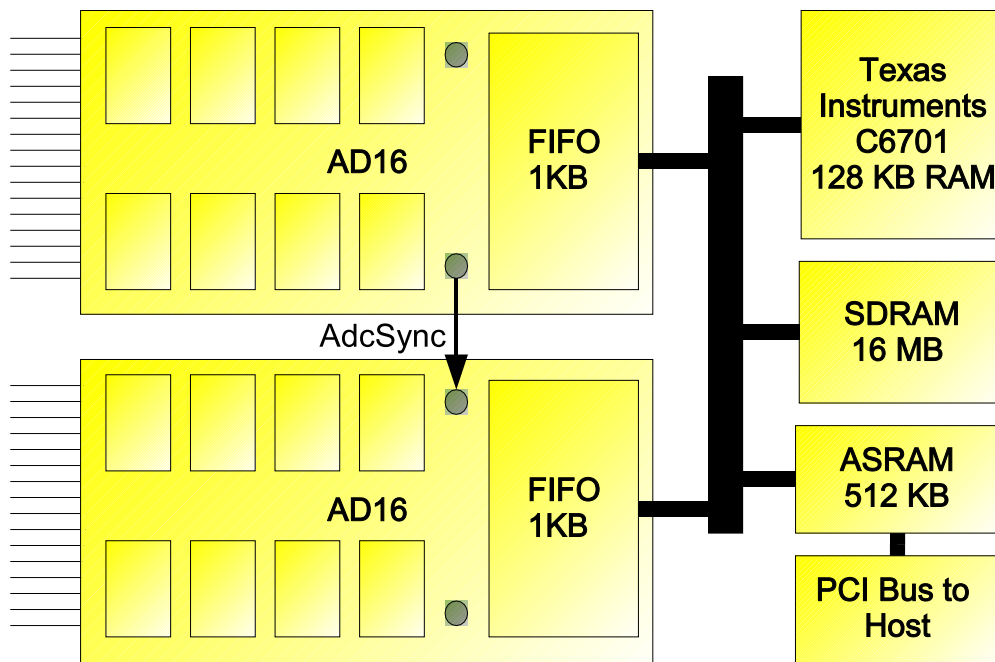


Figure 6.8: Datapath from AD16 to PCI bus

in this manner until all AD16 board are connected. The first AD16 board in this row should reside on the master DSP board in omnibus module slot 0 since the sync signal is triggered by software, and the software implicitly assumes the described topology. The clock transmitted from among the AD16 modules determines the point in time when the ADC move their current value into the FIFO.

### 6.1.6 Host PC

The data acquisition boards reside in an host PC, i.e. a common modern personal computer running Windows 2000. Currently we employ computer with 3 GHz Pentium IV CPU, 512 MB RAM and a hard disk having 250 GB storage capacity. The fully featured system with 128 Channels must have enough performance to store 12.6 MB per second on the hard disk. However, this is not an issue for currently available hard disks. The main constraint is the space available within the cabinet of the computer as four full length PCI boards together with the cabling must fit into it. Regarding this constraint we use an industrial 19" case, that contains a backplane with 20 PCI slots and in which most parts of the computer, i.e. CPU, memory, graphic adapter, Ethernet adapter, reside on a single board, called slot CPU. The remaining space is needed for the DSP boards. One of these boards needs one PCI slot and because of the AD16 modules which reside in a second level on the DSP board one neighboring PCI slot cannot be used.

The analog input of the AD16 boards must be connected to the 50 pin latch connectors of the M67 which are visible at the very right of the image in Figure 6.6. Because the amplifier delivers the analog input of 64 channels and the ground via

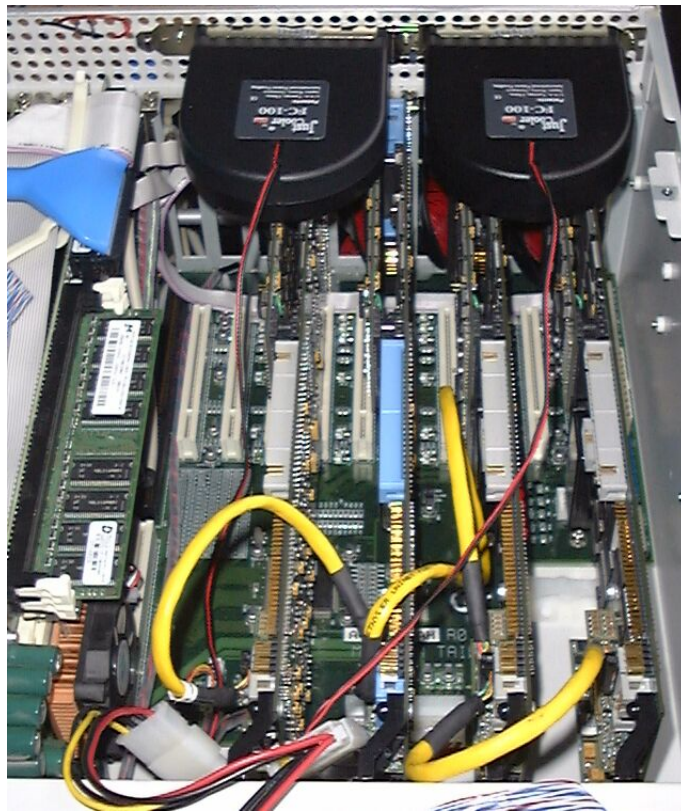


Figure 6.9: Host PC of data acquisition system with 128 channels. On the left you find the slot CPU, and the other four boards on the right are the data acquisition boards. At the bottom the SyncClockLink cable which synchronizes the boards and at the top the additional fans are visible.

68 pin SCSI cable an appropriate adapter board is needed. Figure 6.10 shows this board, which is called ScsiToLatch board. The 64 input channels are distributed to the four latch connectors, i.e. 16 channels on each latch connector, which must be connected with the respective connectors on the M67 boards using a 1 to 1 50 pin ribbon cable. Since the AD converter of the AD16 board need differential input there is a negative and a positive input pin for each AD converter. The ScsiToLatch board connects all negative input pins to ground and the positive input pins to the actual analog input channel. The SCSI connector has 4 lines with ground which are separately mapped to the four latch connectors. Thus, it is possible to provide different ground levels to for each block of 16 channels. However, with the jumpers on the ScsiToLatch board all ground lines can be connected in order to make sure that they are equal.

### 6.1.7 Thermal issues

In a series of test runs it appeared that the AD16 modules have a thermal problem. We tested different combinations of M67 boards equipped with two AD16 modules:

- Combination A with M67 board G274, AD16 modules B145-1 and B138

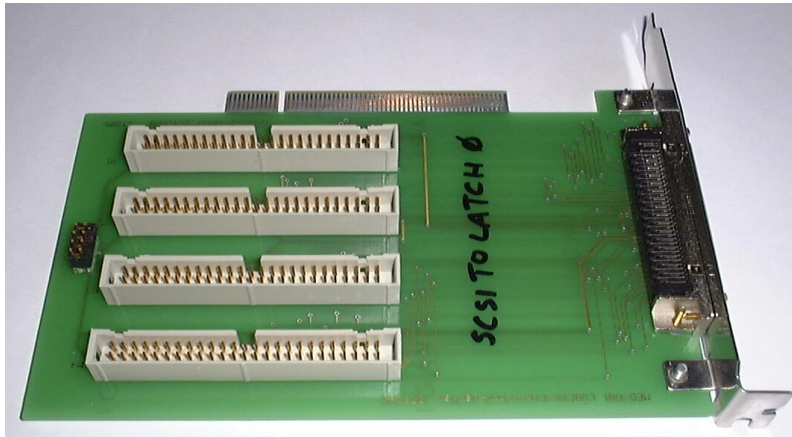


Figure 6.10: Adapter board for analog input SCSI to Latch.

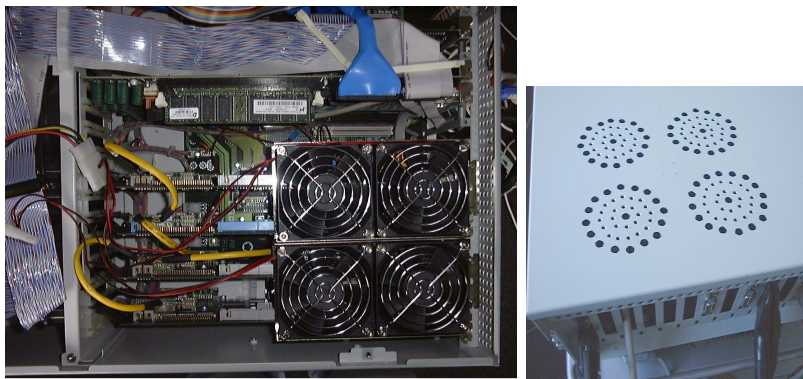


Figure 6.11: Additional fans and airholes in the cabinet.

- Combination B with M67 board G244, AD16 modules B145-2 and B144
- Combination C with M67 board G275, AD16 modules B149 and B140

Combination A and B both showed similar thermal problems. These thermal problems caused the data acquisition hardware to break down at a room temperature over  $25^{\circ}\text{C}$ . In order to resolve the thermal problems we made several different approaches. The approach with the fans visible in Figure 6.9 was not sufficient. Finally, a heat spreader designed by Innovative Integration for a pair of AD16 modules combined with additional fans and air holes in the cabinet resolved this problem for room temperatures up to  $30^{\circ}\text{C}$ . This was tested for combination A and B. However, the third combination C worked stable without additional cooling in our prototype system at a room temperature of  $23^{\circ}\text{C}$  for 16 hours recording data to disk from 32 channels at 50 kHz per channel.

The heat spreader assembled together with a pair of AD16 modules is shown in Figure 6.6. The additional fans and the air holes are shown in Figure 6.11.

We also tested an completely different cabinet design which is shown in Figure 6.12. This design features additional cooling through several fans. However, the



Figure 6.12: Small cabinet with improved air flow due to multiple fans with M67/AD16 board equipped with heat spreader

combination A worked not stable in this cabinet, i.e. the data acquisition stopped after 30 minutes at a room temperature about 24° Celcius. Instead, the combination B which does not have heatspreader worked stable over 18 hours at a room temperature at about 32° Celcius.

Our tests lead to the conclusion that the thermal stability of AD16 modules differs individually, which in turn means that the thermal behaviour of each AD16 module should be checked, before being used in a productive system. This seems to be custom for most PC-Systems and is called burn-in.

## 6.2 Software

The data acquisition software has two parts: a DSP application and a host application. The DSP application executes on the DSP boards, while the host application runs on the host PC. Both applications were implemented using an appropriate integrated development environments (IDE), i.e. Texas Instruments Code Composer Studio for the DSP application and Borland C++ Builder for the host application. Both IDEs run under Windows NT/2000 and provide project management, compilation, and execution/debugging interface. The DSP application is based on the DSP/Bios real time operating system, while the host application employs Borlands Visual Component Library (VCL) which encapsulates Windows API functions and Windows standard controls and dialogs.

### 6.2.1 DSP application

Being executed on the M67 board the DSP application is virtually invisible for the user. It is transparently configured and started by the host application and performs the following major tasks:

- Data acquisition
- Processing



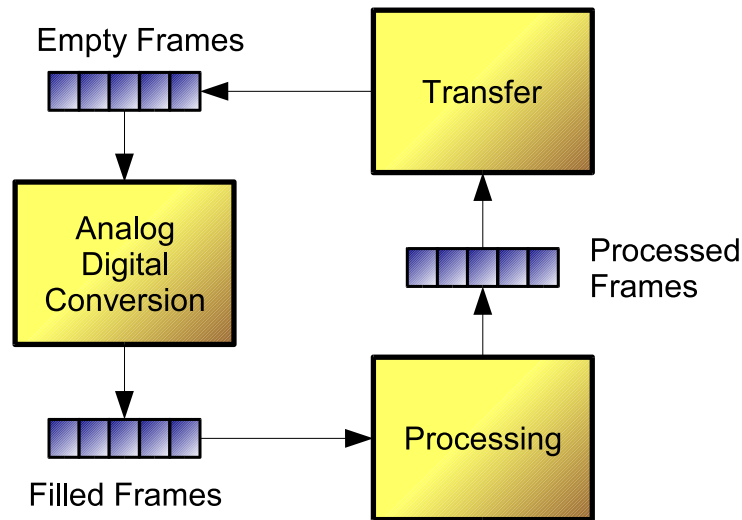


Figure 6.13: Packet management within DSP application

- Transfer
- Trigger detection

In other words the primary tasks of the data acquisition system are implemented within the DSP application. Since it interfaces with the real world and in some tasks hard deadlines must be met, the DSP application uses a realtime operating system. Appendix C.1 gives an overview of basic concepts and some important features of such operating systems and it describes some of the particular data structures provided by DSP/Bios. In the following paragraphs each module is discussed in detail.

### Data Acquisition Module

The data acquisition module consists of one part which provides the interface to a single AD16 board called VSAdc and a second part which integrates two such VSAdc modules, called VSAdcPair. The module VSAdcPair initializes and performs the simultaneous data acquisition from a pair of AD16 boards.

The data acquisition is driven by hardware interrupts. At this the FIFO of each AD16 is configured to notify the DSP with an interrupt signal each time its fill state exceeds a certain threshold. One FIFO is able to hold 512 sample values. This corresponds to 32 sample points if we acquire data from all 16 channels of the AD16 board. The threshold for interrupt generation is set to half the size of the FIFO. After both AD16 modules have triggered their interrupts, two DMA transfers are started which transfer the data from both FIFOs into the DSPs internal memory. Here the data is stored into so called packets. Each packet has a header and a data section. The header contains information about the data, like e.g. the size,

whether a trigger signal has been received, the type of the data, and a consecutive number of the packet. A data section of a packet consists of several sample point blocks, in which a sample point block contains the samples from all channels taken at the same point in time. The DMA transfers from the two FIFOs into a packet are configured so that they perform the merge operation which leads to this organization in memory. See Section 6.1.4 for a description of the DMA controller capabilities. After a packet has been filled it is passed to the processing module. The data is processed and then the packet is passed to the transfer module, and finally back to the data acquisition module. The exchange of packets is implemented by means of packet queues (data structure QUE, see Appendix C.2.2). There are three such queues which contain either empty packets, filled packets, or processed packets. Figure 6.13 shows the data flow within the DSP application.

During acquisition the DSP application can receive external trigger events. The trigger input line must be connected with a certain external interrupt line of the C6701 (i.e. HWI 6, see Appendix B.3). An interrupt on this line is handled by a routine which sets a flag and stores a time stamp and the current state of the DIO port. Each time the FIFO transfers have been finished, the trigger flag is checked and if a trigger has been detected the time stamp together with the values from DIO port is stored in the packet header. With the time stamp it is possible to compute the index of the respective sample point within the packet. This is done in the host application.

### Processing Module

The acquired data can be processed differently. Independent from the actual method the processing can either be in place, i.e. the original input is replaced by the processing results, or it might produce new data, which must be transferred in addition to the original data. For instance the results of a short time Fourier transform for a single channel would usually be transferred separately, while the result of an IIR filtering on all 32 channels would rather replace the original signal, at least if we cannot afford twice as much data being transferred to the host PC.

If processing is enabled, then the processing is triggered by software interrupt (SWI), which is generated by the data acquisition module each time a packet is available in the filled packet queue. The software interrupt is generated directly after the DMA transfers from the FIFOs have been finished. It has a lower priority than the software interrupts used for data acquisition. Otherwise, the processing could block the acquisition.

The communication protocol between host application and DSP application allows transfers of different kind of data as described in Section 6.2.1.

In the scope of this work a signal processing frame work based on the wavelet transform was developed. One of the key task here is the decomposition of the signal into its wavelet coefficients (see also Section 4.2). This task is performed by the DSP application within the processing module.

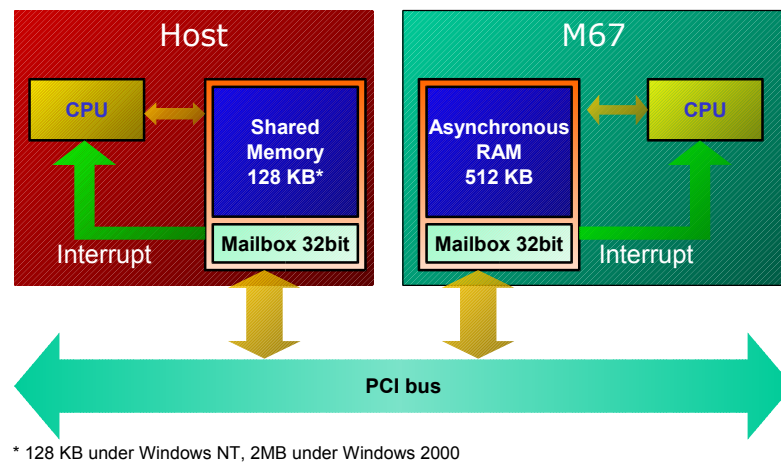


Figure 6.14: Communication channels between host and target

### Communication between host and target

Host and targets, i.e. PC and M67 boards, respectively, are connected via the PCI bus. Data can be transferred from host to target and vice versa. There are two mechanisms one for small messages and one for large amounts of data which are illustrated in Figure 6.14.

First we focus on the transfers of large amounts of data. The target can read from or write into a shared memory section within the host memory. Data transfers are always performed via the Asynchronous RAM (ASRAM) on the target, i.e. for transfers from target to host the data must be read from ASRAM, and for transfers from host to target the data is received in the ASRAM. The M67 can perform busmaster transfers, i.e. data can be read or written without burdening the target or the host CPU. Instead, transfers are executed by a special controller chip, the V360 PCI to local bus bridge (V3 Semiconductors). After the transfer has been completed the controller notifies the target by triggering an interrupt.

For small messages we use a mailbox system. A *mail* is a 32 bit value. The board supports two mailboxes. They can be accessed in a synchronous way: For example, if we want to send a message to the target, then we can write the value into the mailbox on the host and it is send to the target. Now the target must read the mailbox before a new value can be written by the host. If the target tries to read the mailbox when no message has been sent by the host, the target is blocked until a value is available. Using these mailbox functions, target and host can be synchronized. But synchronous communication is error prone, because messages need to be sent in the correct order which introduces a great dependency between target source code and host source code. This is why we do not use the synchronized mailboxes. Instead, we use an asynchronous approach which is also supported. A write to a mailbox can trigger an interrupt. Thus the host can interrupt the target and also the other way around, the target can interrupt the host. By using the interrupt signal we do not need to check for new messages actively, which saves valuable CPU cycles.

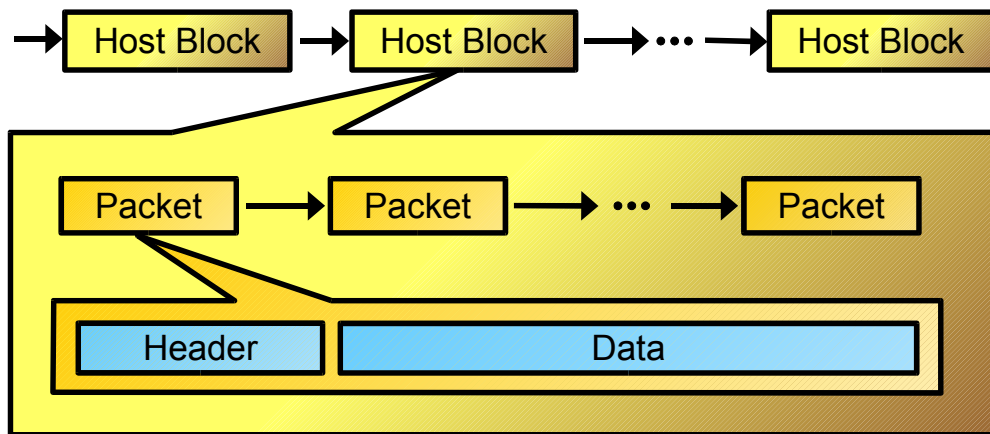


Figure 6.15: Data Transfer Protocol

### Data Transfer Module

After processing, or after acquisition, if only raw data is requested, the data must be transferred to the host. This task is implemented in the data transfer module. Since busmaster transfers, as described in the previous section, have to use the ASRAM as source for transfers into the shared memory on the host site, the first step is the transfer of the acquired data from internal DSP memory into the external ASRAM (see also Figure 6.8).

Several packets together form a host block. After a host block has been transferred from target to host, the target notifies the host using the mailbox communication so that it can fetch the new host block. Note, that we start a new busmaster transfer for each packet, but notify the host not until a complete host block has been transferred. This is done in order to keep the number of events which are generated by the notification as low as possible. Figure 6.15 illustrates the data organization of the busmaster transfers.

### Commands Module

In the DSP application a single task, the control task, receives messages sent by the host and also sends messages to the host. The 32 bit messages have three parts: a receiver tag (8 bit), a command tag (8 bit), and a data part (16 bit). The control task routes the messages according to the receiver tag. In order to set the sampling rate within the ADC module the host sends a message with receiver equal to `ADC_MODULE`, command is `SET_SAMPLING_RATE`, and the data contains the sampling rate. In case that the value exceeds the 16 bit of the data part, we use two or more messages.

### Extensions for data acquisition with multiple DSP boards

The acquisition with multiple DSP boards requires some modifications in the DSP application. As there are multiple DSP boards, there are also multiple instances of



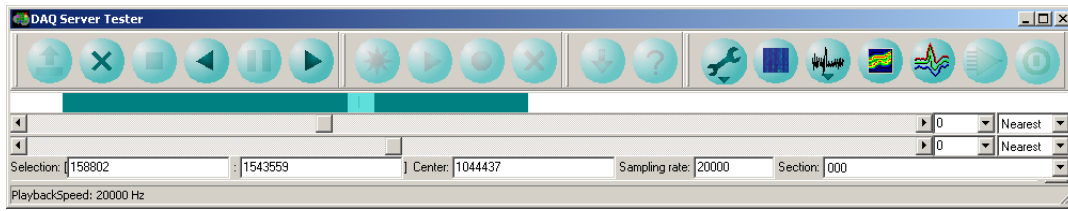


Figure 6.16: Main Panel

the DSP application. As described in Section 6.1.5 one board is defined to be the master board, while the others are slave boards. Similarly we also have a master DSP application and slave DSP applications, which run on master board and slave boards, respectively. On data acquisition start command the master DSP application initializes the DDS timer and synchronizes the ADCs on the AD16 modules, while the slave DSP applications do not perform these steps. Despite these details master and slave DSP applications behave identical. Note, that the start command must be sent to the slave DSP applications first, to ensure that they are ready for acquisition when the master DSP application enables the timer.

### 6.2.2 Host application

The host application provides a user interface through which new experiments can be created and old experiments can be reviewed. It has several means of visualization, e.g. a virtual scope, a blue plot, and a spectrogram. The host application receives the data from the target and stores it into a file on hard disk together with an experiment documentation file. The experiment documentation includes an experiment identifier, a global comment, an experiment log book, the sampling rate, the used probe and amplification settings. In review mode the user can browse through the data of an old experiment using controls similar to controls of a CD player. Recapitulating the host application performs four main tasks:

- Data acquisition including reception and storage,
- Experiment data organization,
- Data analysis
- Data visualization.

### Dialogs and visualization modules

In the following sections we describe the graphical user interface of the data acquisition software.

#### Main Panel

After the program has been started the main panel of the data acquisition system appears. The buttons in the top row trigger provide access to all main features of

	Open for play back		New experiment		Tools
	Close current		Play		Blue plot
	Stop playback		Record		Virtual scope
	Backward play-back		Cancel acquisition		Spectrogram
	Pause playback		Init DAQ HW		Spike plot
	Forward playback		Test DAQ HW		Amplifier setup
					Quit application

Table 6.5: Activation of visualization modules

the application. They are grouped by functionality (Table 6.5). The first group is active in replay mode, i.e. if the user reviews already acquired data from disk. With the first button (arrow up over rectangle) the user can open such a data file (\*.vse or \*.h5). The second button (green cross) closes the current file. The next four buttons are used to start playback in forward and backward mode and to stop or pause the playback. The playback speed can be modified in the edit field "Sampling rate". The area under the button row contains several widgets which form the so called browse bar. It is sort of an extended scroll bar and supports the user in browsing through datasets which are large in time. Furthermore, it can indicate a selected range within the dataset. The rectangular area at the top outlines the settings of the browse bar and is called browse bar overview. In its width it represents the full time line of the current data section. In Figure 6.16 the selected range is indicated by dark cyan colored rectangle. The white areas are not in the range. Within the range the light cyan colored rectangle indicates the part of the data set which is currently stored in the internal data buffer. However, the selected range is independent from the internal data buffer. Thus, if the selected range starts within or outside this buffer, then the unselected part of the data buffer is indicated by a gray rectangle. The selection range can be modified by clicking and dragging with right and left mouse button inside the browse bar overview or with the respective selection edit fields. Inside the rectangle which represents the data buffer, there is another rectangle with a smaller height. This rectangle, which might also be condensed into a vertical line, represents the view buffer, i.e. the section which is currently visualized in the scope or in the blue plot. The browse bar has two scroll bars. The upper scroll bar moves the data buffer and the lower scroll bar moves the view buffer within the data buffer. The combo boxes at the right of the scroll bars are used to change the time scale. The number defines a decimation, i.e. the number of values which will be discarded for each displayed value. Additionally, the decimation mode can be changed. At this the application supports nearest neighbor and maximum value.

The New Experiment button (red star) item brings up a dialog which is described in Section 6.2.2. When a new experiment has been created and the DSP application has been properly initialized, data acquisition may be started by pressing either the Play (red triangle) or the Rec (red circle) button. If the data acquisition is started

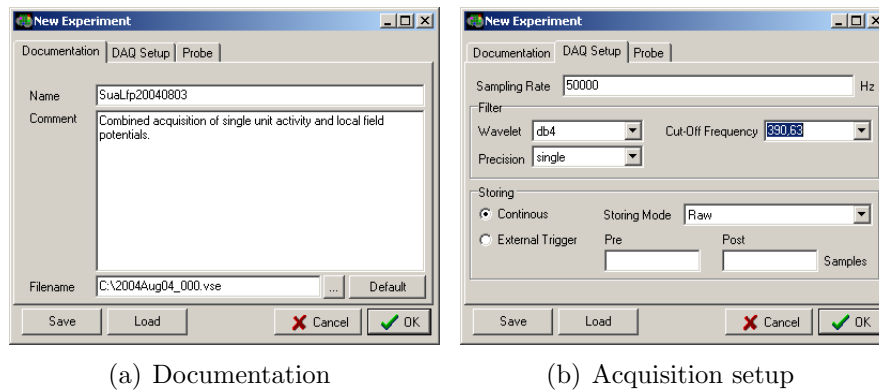


Figure 6.17: Configure a new experiment

with the Play button, data is acquired but not stored to disk. This mode may be used to view the current signal. Start the acquisition with Rec button in order to acquire data into a file on hard disk. By pressing Play button in record mode and Rec button in play mode one can switch between both modes, respectively. During acquisition different visualization modules can be used to display the data. The application provides multiple virtual scopes, a blue plot, and a spectrogram. The acquisition is stopped when the user presses the Stop button.

The visualization modules are opened and closed with their respective buttons corner of the main window.

## New Experiment

In the new experiment dialog (Figures 6.17(a), 6.17(b), and 6.18(a)) the user can change the experiment parameters as described in the next paragraphs. With the Save and Load buttons a setup can be stored to or retrieved from disk, respectively. Press button OK to accept the experiment settings. A new experiment is created and the DSP application is initialized appropriately. If this was successful, one can proceed by pressing the Play or Rec button to start acquisition, or by pressing the Stop button to cancel. Close the dialog New Experiment with button Cancel, when no new experiment should be created. The dialog has three tabbed pages. On the first page the general documentation for the experiment can be entered.

**Name** Experiment identifier

**Comment** A comment which should summarize the overall setting of the experiment

**Filename** File where the experiment will be stored. The application supports two different file formats, i.e. a proprietary raw data format (\*.vse, \*.rawdata) and HDF 5 (\*.h5). In case of the proprietary raw data format the experiment parameters are stored in the .vse and the raw data is stored in a set of \*.rawdata files (see also Section 6.2.2). The base file name is generated by default using the structure <year><month><day>\_<number>. One can also select a filename

in a file select dialog (push button with the three dots), or enter a name manually. Pressing the button Default generates a default name.

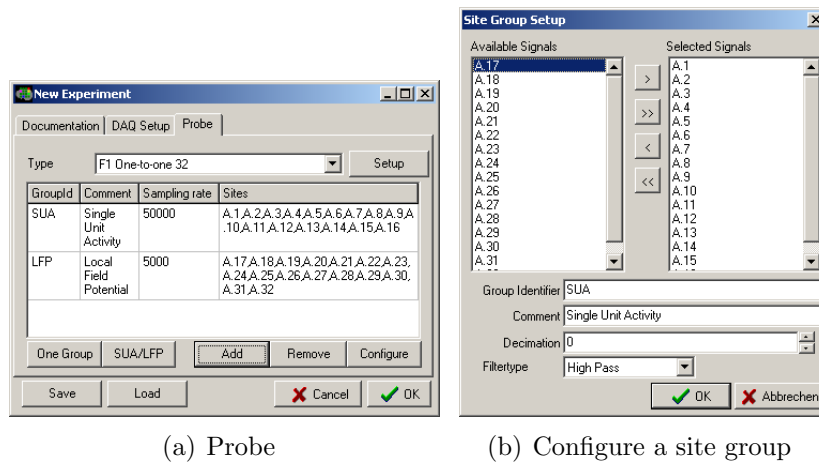
On the tab page DAQSetup the settings for the data acquisition are defined.

**Sampling Rate** Enter the sampling rate at which the data should be acquired. The rate can range between 5 kHz and 50 kHz.

**Filter** With these settings the application can be configured to process the acquired data in a wavelet filter bank. By default, if Wavelet is set to 'none', this filtering is disabled. Otherwise if a certain Wavelet is set, this wavelet is used in the filter bank. With the Precision setting one can select between 16 bit integer precision ('int16') and 32 bit floating ('single') for the wavelet coefficients. Finally, one can choose the desired cut-off frequency from the respective combo box. Due to the nature of a the filter implementation via an dyadic wavelet filter bank, only a certain set of cut-off frequencies is available (Section 5.1).

**Storing** The data can be acquired either continuously or based on an external trigger. In continuous mode all acquired sample points are stored. They can either be stored as raw data points, or also as wavelet coefficients. To acquire based on an external trigger, the number of sample points before and after the trigger event must be defined. The post trigger sample point count, includes the sample point which is associated with the trigger event. Thus, the length of each sweep is the sum of pre and post triggered samples.

With the Probe tab page a certain probe can be selected which is going to be used in the experiment. New probes can be added in the probe setup dialog which is opened by button Setup. Each recording site of the probe can be assigned to a certain site group. The groups can have different acquisition settings w.r.t. filter selection and decimation. The table shows all currently specified groups and some of their parameters. In particular a group identifier, a comment, a sampling for the group, and a list of recording site assigned to the site group. In the default case all recording sites are in one group. The button One Group creates this setting and each time a different probe type is selected, also a single group with all recording sites of this probe is created. Note, that at this the old setting is lost. With button SUA/LFP a setting with two groups one for single unit activity and another for local field potentials is prepared. After creation both groups are empty. The user can select a group with a mouse click into the respective row of the group site table and then modify the settings in the dialog which is opened by the Configure button (Figure 6.18(b)). In this dialog recordings sites from a list with available sites can be added to the list of selected recording sites and vice versa. Note, that a site can only be assigned to one site group. In the other edit fields group identifier, comment, decimation, and filter type can be defined. Group identifier and comment are arbitrary strings. The decimation is a positive integer or zero. It indicates the number of sample points which are discarded for each stored sample point. The filter type, which can be none, i.e. disabled, high, or low pass refers to the cut-off frequency of the filter specified on the DAQ Setup page (Figure 6.17(b)).



(a) Probe

(b) Configure a site group

Figure 6.18: Probe and site group settings

## Probe Setup

Within the probe setup dialog (Figure 6.19) new designs of probes can be entered or existing designs can be edited. The interface assumes that a probe has one or more shafts, where each shaft is occupied by a certain number of recording sites, which in turn are vertically oriented on the shaft.

Therefore, the dialog shows three lists: At top the list of probes, in the middle the list of shafts of the currently selected probe and at the bottom the list of recording sites of the currently selected shaft. In the table which has the input focus the currently selected item is marked by a dashed line around a cell in the row, in both other tables the selected items are marked by a blue background of one cell in the row. On the left of each table there are three buttons, which create a new item, delete an item, or duplicate an item in the respective table. For example to enter a new probe of type E1 one would select a present instance of E1 and duplicate it. It is appended at the end of the list of probes and can be selected. Now one can enter a new comment or number and modify the shaft and site configuration with respect to the actual probe. The position of the shaft is defined by the position of its tip. The position of each recording site is specified by the distance to the tip of the shaft. Given these informations the spatial relation of two recording sites can be derived easily. Additionally, the mapping of recording site onto channel number is defined in this dialog. Note, that channel numbers start at 1.

## Amplifier Setup

The amplifier setup dialog is activated by the button with the amplifier symbol (Figure 6.16).

The gain of each site of the current probe can be set in the amplifier setup. A schema of the probe is drawn in the upper half of the window as shown in Figure 6.20. The gain value can either be selected by using the gain combo box or by clicking onto the legend. The current setting of a gain value is indicated by different colors for each site. Each probe has a default gain value setting which can be set pushing

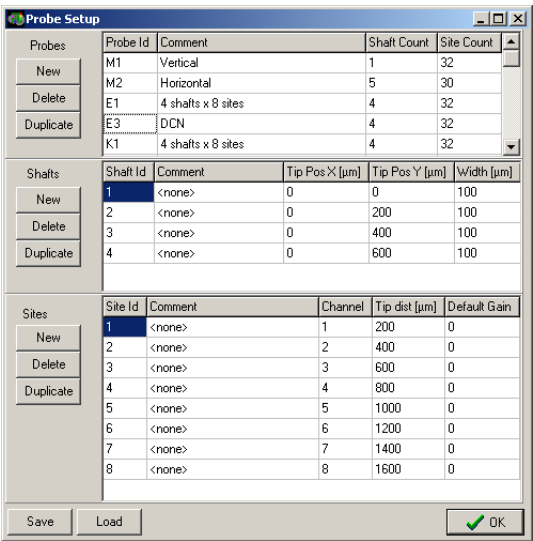


Figure 6.19: Probe setup dialog

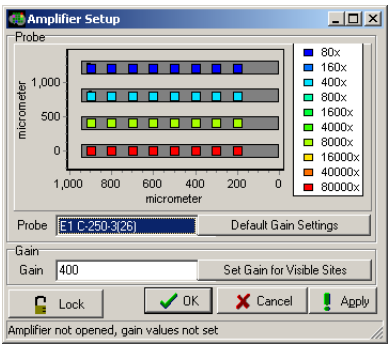


Figure 6.20: Amplifier setup dialog

the Default Gain Settings button. A set of sites can be selected by zooming into the schema of the probe such that only the desired sites are visible. Then the desired gain can be set for these sites using button Set Gain for Visible Sites.

The dialog is synchronized with the setting at the main amplifier, i.e. manual changes at the amplifier are visualized by the dialog and vice versa. Manual changing of the gain at the main amplifier can be locked using the button Lock. The button indicates the current state by showing the text "Locked" or "Unlocked". The gain settings are transferred to the amplifier when the Apply button is pressed.

### Blueplot

In a blue plot the amplitude of the displayed signals is color coded. The labels on the x-axis indicate the recording site names. The y-axis is the time line, which can indicate the time stamps of the sample points in terms of an sample point index or in terms of seconds at different scales, i.e. microseconds, milliseconds, or seconds. On the left a color map is displayed. The color mapping can be controlled by mouse

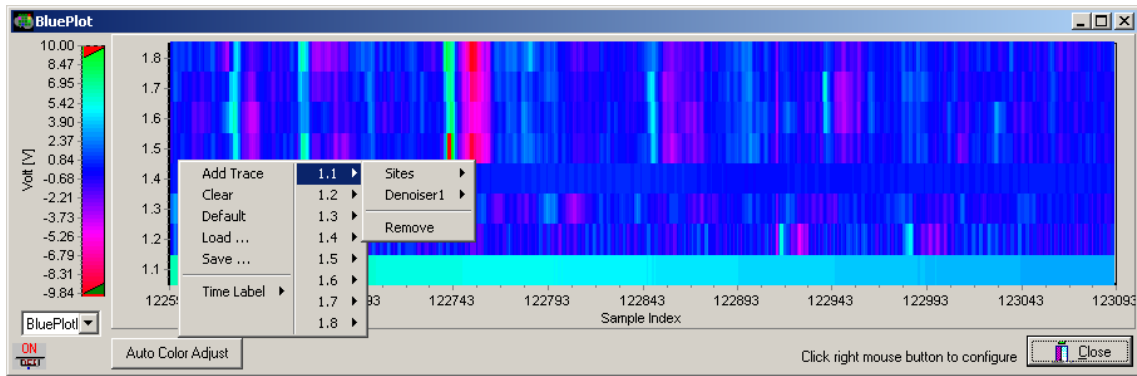


Figure 6.21: Blueplot

using the sliders in the color bar. They define the range of data values which is mapped onto the colors. If the range has length zero, the sliders define a threshold. It is also possible to change the color map using the combo box at the bottom of the color bar. With the button Auto Color Adjust the color mapping is adjusted to the data currently shown in the blue plot. After this adjustment the median of the current data is mapped on to the center of the color bar. Minimum and maximum are computed based on the mean deviation of the data values from the median. The mapping is symmetric around the median.

The blue plot supports an arbitrary number of traces, at which each trace can be associated with a recording site or an analyzer result. With the context menu of the blue plot the user can add a trace, remove a trace, and associate a trace with a signal. In Figure 6.21 the context menu is visible. It shows the default configuration for a probe with eight recording site, which is created if the entry Default is selected. The first trace is configured via its context menu. If the user selects the entry Remove, the trace is removed. The entries Sites and Denoiser1 refer to context menus which contain the currently available recording sites and denoising results from Denoiser1. The selection of one entry within these menus sets an association of the trace with the respective signal. The entry Clear removes all traces from the blue plot. With the entries Load and Save a setup can be retrieved from or stored to a file, respectively.

## Spectrogram

The spectrum of a single channel is visualized with the spectrogram. It is based on the short time Fourier transform which is computed within the host application. The short time Fourier transform is implemented by a fast discrete Fourier transform applied on a certain time window of the signal. Thus we compute the coefficients  $S_{w,d}(k)$  which depend on the window function  $w$  and the time window of signal  $s(n)$  selected by offset  $d$ :

$$S_{w,d}(k) = \sum_{n=0}^{N-1} s(n+d)w(n)e^{-j(2\pi/N)nk}$$

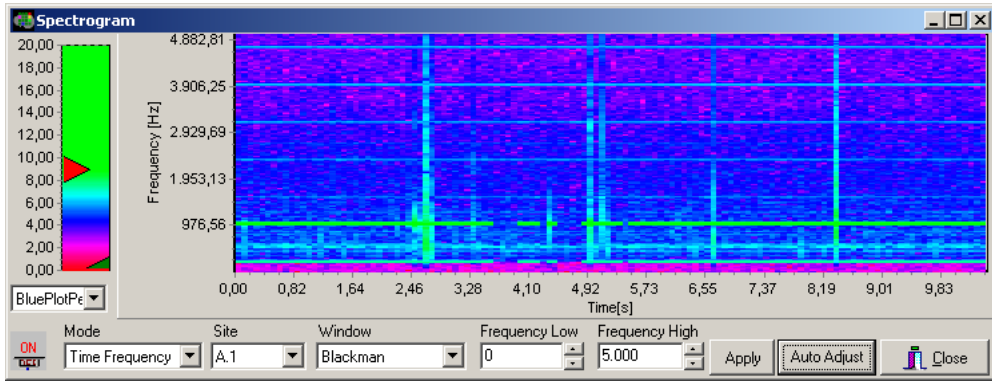


Figure 6.22: Spectrogram in time frequency mode

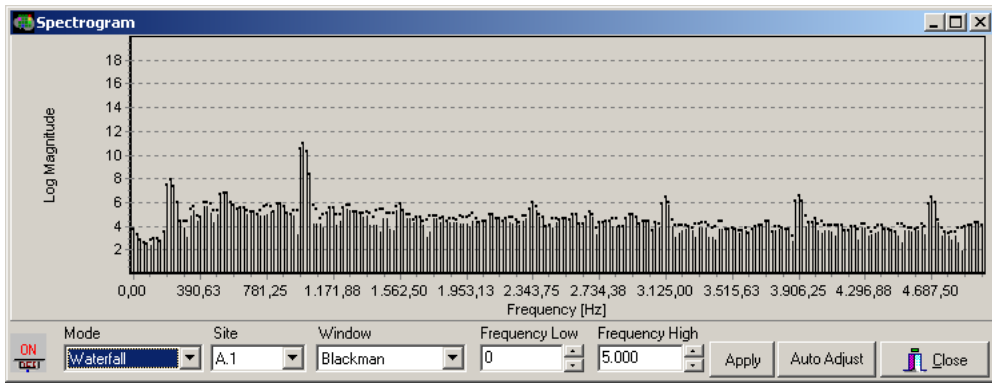


Figure 6.23: Spectrogram in waterfall mode

The length  $N$  of the window is chosen as a power of 2, and it determines the resolution in frequency domain, i.e. the width of the frequency band represented by a single Fourier coefficient. The spectrogram provides different windows taken from [86], which are described in the following, whereby  $M = N - 1$ :

#### Rectangular

$$w(n) = \begin{cases} 1, & 0 \leq n \leq M, \\ 0, & \text{otherwise} \end{cases}$$

#### Bartlett (triangular)

$$w(n) = \begin{cases} 2n/M, & 0 \leq n \leq M/2 \\ 2 - 2n/M, & M/2 < n \leq M \\ 0, & \text{otherwise} \end{cases}$$

#### Hamming

$$w(n) = \begin{cases} 0.54 - 0.46 \cos(2\pi n/M), & 0 \leq n \leq M, \\ 0, & \text{otherwise} \end{cases}$$



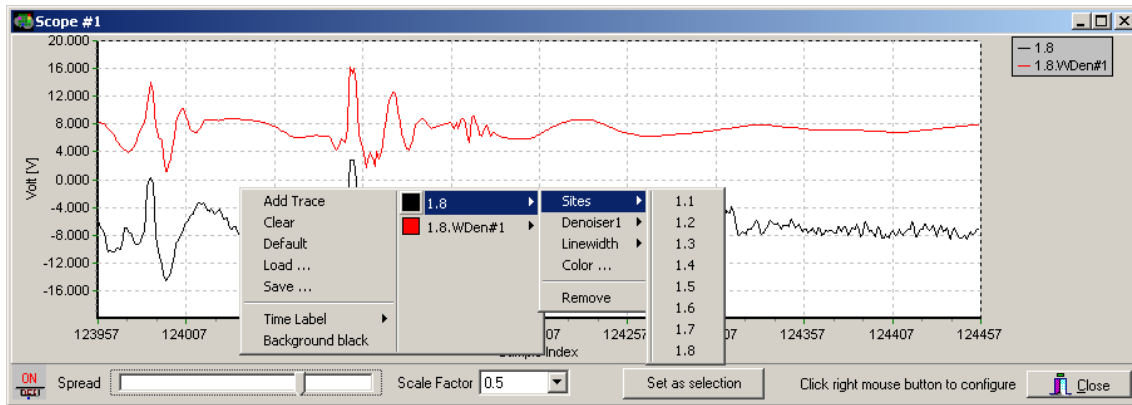


Figure 6.24: Scope with open setup context menu

### Hanning

$$w(n) = \begin{cases} 0.5 - 0.5 \cos(2\pi n/M), & 0 \leq n \leq M, \\ 0, & \text{otherwise} \end{cases}$$

### Blackman

$$w(n) = \begin{cases} 0.42 - 0.5 \cos(2\pi n/M) + 0.08 \cos(4\pi n/M), & 0 \leq n \leq M, \\ 0, & \text{otherwise} \end{cases}$$

The spectrogram shows the log magnitude of the short time Fourier coefficients for the selected channel with respect to the selected mode. The spectrogram has two modes: the time frequency mode and the waterfall mode which are shown in Figure 6.22 and 6.23, respectively. In time frequency mode the time scale is on the x-axis and the frequency on the y-axis. The log magnitude of the coefficients is coded into a color. This color mapping can be adjusted with the color bar on the left side. The time frequency mode visualizes the spectrum over a long period of time. In waterfall mode the frequency is indicated by the x-axis and the log magnitude on the y-axis. The magnitude of each coefficient is represented by a bar of respective height. Each bar has a little dot on top which indicates the maximum of this bar during the last few windows (currently three windows, at a window size of 512 samples for the Fourier transform). The waterfall mode shows the spectrum only over a short period, but visualization of peaks is better than in the time frequency mode. With the settings in the edit fields Frequency Low and Frequency High the visualized frequency range can be changed. The values are adopted if the Apply button is pressed. With the Auto Adjust button the color mapping is automatically adjusted as described in previous paragraph about the Blueplot.

The spectrogram is useful to locate noise sources which cause peaks at certain frequencies.

### Scope and Scope Setup

With the scope (Figure 6.24) one can visualize the signals of several selected recording sites. The scope has an arbitrary number of traces and each trace has independent settings for recording site, color, and line width. The scope configuration is

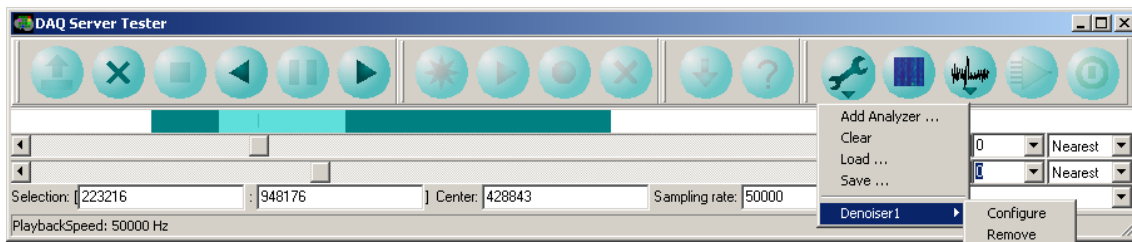


Figure 6.25: Menu of tool button

defined with the scope context menu (Figure 6.24) in the same manner as the blue plot (Section 6.2.2). The traces can be spread or overlaid by the track bar at the bottom of the scope. A recording site or result signal is assigned to a trace by selecting it from the context menu of the trace. A scope setup can be stored into a file and retrieved by using the Save and Load entries of the context menu, respectively. The scope is set back into its initial state by the Clear entry. The Default entry creates a configuration where all available recording sites are shown.

The scope has a zoom feature, i.e. the user can select a rectangular region within the scope axis, which is in turn enlarged. The region is specified by pressing the left mouse button at the upper left corner of the rectangle and holding it while the mouse is moved to the lower right corner, which is pinned down by releasing the left mouse button. Note that it is important to specify the upper left corner first, because specifying any other corner first invokes the unzoom function.

The time scale of the scope can be changed with the decimation control of the browse bar. A decimation value of  $k$  means that only every  $k$ -th sample is visualized. In terms of time scale we enlarge the time scale on the x-axis by a factor of  $k + 1$ .

## Analyzer

The acquired signals can be processed by different analyzers. In general an analyzer has a set of input signals, i.e. signals from recording sites. The analyzer processes this data and produces a set of output signals. The user creates a new analyzer by selecting the entry Add Analyzer from the menu of the Tool button (Figure 6.25). The dialog which is shown in Figure 6.26 opens and the user can select the signals which shall be processed. Then the method must be selected from the Method combo box. The application supports denoising and filtering. The user may specify a name for the analyzer and the analyzer is created if the user confirms the configuration with the OK button. With the Cancel button, the dialog is left without creating a new analyzer. If no signal was selected, an appropriate message is shown and no analyzer is created.

Afterwards, on successful creation of an analyzer, it can be configured or removed via the Tool menu, which contains an entry for each available analyzer (Figure 6.25). The configuration dialogs depend on the actual analyzer and are described in the following paragraphs.

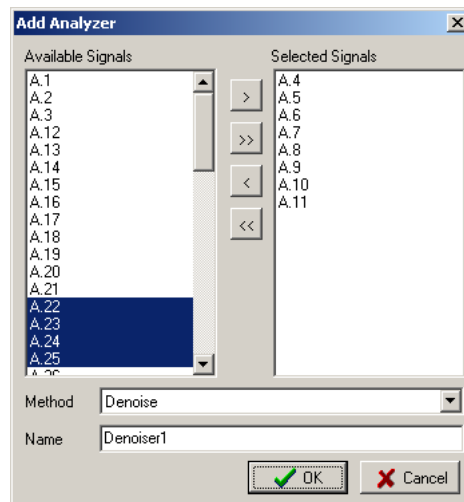


Figure 6.26: Add Analyzer dialog

### Wavelet filter bank setup

Both analyzers the filter and the denoiser utilize a wavelet filter bank for their processing. Thus, in their setup dialogs (Figure 6.27 and Figure 6.28) the parameters which configure the wavelet filter bank are identical:

**Wavelet Type** Select the actual wavelet. There are three different wavelet types available, i.e. Daubechies wavelets with 2, 4, and 6 vanishing moments, called db2, db4, and db6.

**Precision** The wavelet filter bank can use 16 bit integer or 32 bit floating point values as wavelet coefficients.

**Level Count** Select the number of levels of the wavelet decomposition. This number determines the number of frequency bands available. You can select levels from 1 to 6.

### Wavelet filter setup

The signal can be filtered using a wavelet decomposition to eliminate selected frequency bands (see Section 4.2.1). The filter is configured with the filter setup dialog shown in Figure 6.27. The parameters of the filter bank can be chosen according to the description in Section 6.2.2. The level count determines into which frequency bands the signal is decomposed by the wavelet filter bank. A frequency band is kept when the check box is checked or eliminated if the check box is unchecked. Thus, the filter can either be a high pass, a low pass, or a band pass.

### Wavelet denoising setup

The wavelet denoiser as described in Section 5.2 has several parameters which are set in the wavelet denoising setup dialog. The setup dialog is depicted in Figure 6.28

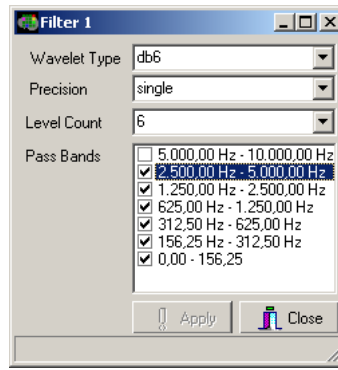


Figure 6.27: Wavelet filter setup dialog

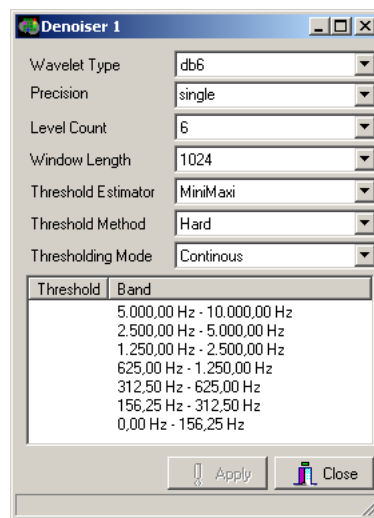


Figure 6.28: Wavelet denoising setup

and shows the specific setup which was used to obtain the result in Figure 6.29. In addition to the parameters of the wavelet filter bank described in Section 6.2.2 the following parameters can be set:

**Window Length** The window length is the number of samples which should be used to estimate the thresholds. Note, that some threshold estimators only use the window length and the noise estimation.

**Threshold Estimator** There are four different strategies available to estimate the thresholds, i.e. MiniMaxi, VisuShrink, Sure, SureShrink. For a detailed description refer to Section 5.2

**Threshold Method** Either "hard" or "soft" can be selected. Hard threshold means that all coefficients less than the threshold are set to zero and all others are kept as is. In case of soft thresholding the magnitude of coefficients which are greater than the threshold is reduced by the threshold.

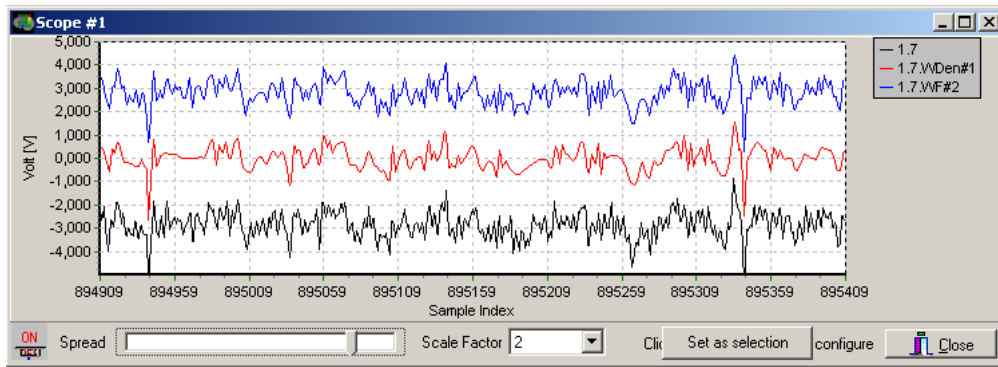


Figure 6.29: Wavelet denoising and Wavelet filtering

**Thresholding Mode** Choose between continuous or single. In continuous mode the threshold are computed for each window, in single mode the thresholds are computed for the first window and then used for the subsequent windows without change. One can force a recomputation by pressing button "Recalc".

**Threshold** The current thresholds with their respective frequency band are shown and can be changed by the user. The thresholds can range between 0 and 32768. However in continuous mode they keep on changing, and therefore manual change of the thresholds make only sense in single mode. In particular the threshold for the lowest frequency band can be set to 32768 in order to obtain a high pass filtered denoised signal.

## Storage

There are basically two parts that need to be stored, i.e. the experiment parameters and the acquired data. Experiment parameters include e.g. settings from the New Experiment dialog (Section 6.2.2). The acquired data consists of the values sampled from the input signals. During acquisition the user can switch between play and record mode. Every time the user switches to record mode, a new data section is created. Furthermore, the signals are organized in groups, where each group has different settings w.r.t to filtering and decimation. Therefore, the acquired data is indexed by two indices, the section index and the site group index.

The data acquisition system supports two file formats. The first file format stores all experiment parameters in a single text file and the acquired data is stored in a set of binary files. Each file corresponds to a certain site group and a certain data section. The storage format is 16 bit integer in little endian. The following Matlab program fragment could be used to load a part of such a file where the number of samples and the number of channels must be given.

```
fid = fopen('2004Aug05_002.000.001.rawdata','r','ieee-le');
s = fread(fid,sample_count*channel_count,'int16=>int16');
fclose(fid);
```

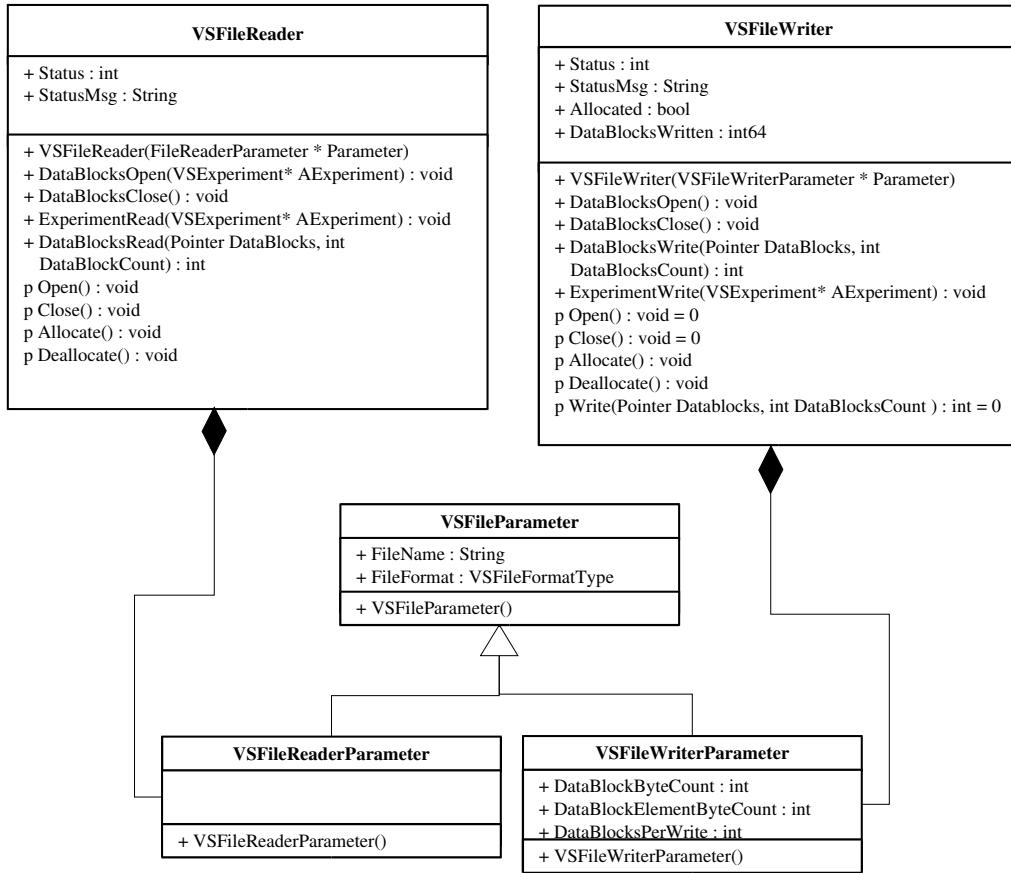


Figure 6.30: Class diagram for VSFileReader and VSFileWriter

Note, that the samples, which are acquired at the same point in time, i.e. a sample point from all channels, are stored as a block. Thus the binary file is a sequence of such blocks.

The other file format utilizes the HDF5 library (HDF stands for Hierarchical Data Format) provided by the NCSA. The library is widely used for exchange of scientific data sets. Using the standard data types of HDF5 and its capability to structure the data hierarchically in groups with associated attributes or subgroups it supports storage of meta data, i.e. experiment parameters. The acquired data is stored in two dimensional arrays of 16 bit signed integer values. Hereby, the HDF file contains one such array for each site group of each section. The experiment parameters are store in a dataset which corresponds to the class definition of VSExperiment.

Data storage and retrieval is handled by two classes: VSFileReader and VS-FileWriter. Both classes are abstract and define the interface for different file writer and reader classes which use their specific file format. Two different derived classes support the two different file formats which are supported, i.e. HDF 5 and the raw data format. Based on user input the respective file reader and writer are used. At this an instance of the factory class VSFileProvider is employed, which is responsible

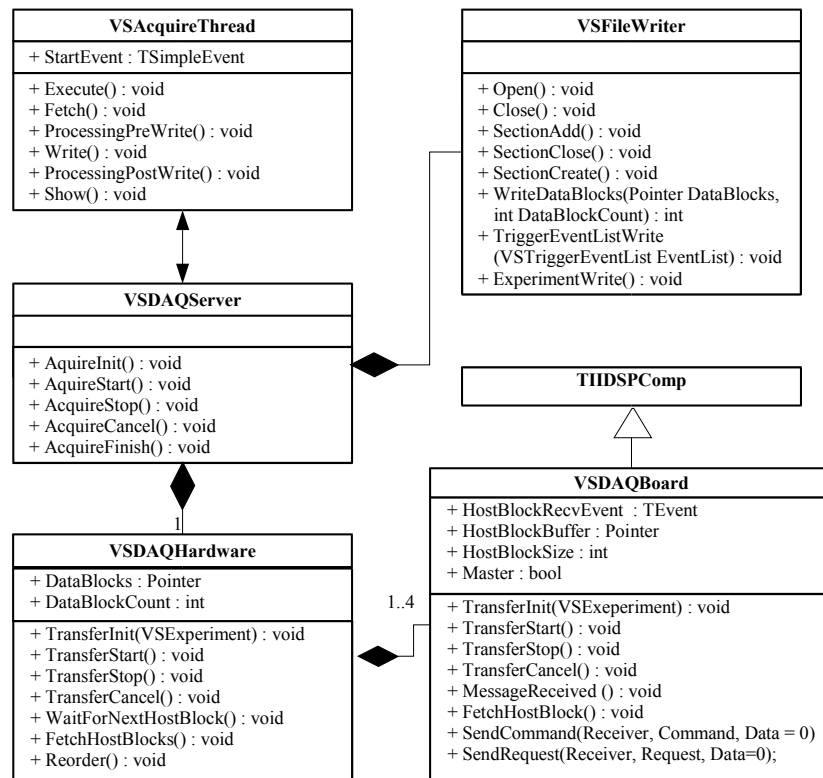


Figure 6.31: Classes for data acquisition from multiple DSP boards.

for creation and deletion of the correct file writer and reader.

### Data acquisition server

The data acquisition is controlled by an instance of class VSDAQServer. The DAQ server provides data to the visualization modules. It has two main working modes, i.e. acquire or playback mode. In acquire mode the data is received from the DSP applications and stored to hard disk while in playback mode it retrieves the data from a file on hard disk. The data acquisition server is composed from several classes. The classes used in acquire mode and their relationship is illustrated in Figure 6.31. The methods and classes which are used in playback mode are omitted. The coordination of the data acquisition hardware is implemented in class VSDAQHardware. This class has a list of all DSP boards available in the current system. Each board is represented by an instance of VSDSPBoard which is derived from TIIDSPComp a class supplied in a library from Innovate Integration. Most methods in VSDAQHardware are delegated to the respective method of each VSDSPBoard instance. The class VSDAQHardware distinguishes between present boards and boards which are actually used during the current experiment. Furthermore, it maintains a pointer to the master board and a list of slave boards, which together constitute the list of active DSP boards. Some methods need to delegate actions first to the slaves and then to the master or vice versa, therefore this differentiation is required. Messages





StoringMode	ProcessingPreWrite	ProcessingPostWrite
Raw	nothing	Wavelet filter bank
RawFiltered	Wavelet filter bank	nothing
Wavelet Coefficients	Wavelet decomposition and coefficient delay	Coefficient processing and wavelet reconstruction.

Table 6.6: Post- and pre-write processing

the method `WaitForNextHostBlock` in class `VSDAQHardware` in order to wait until each DSP application has sent the host block. Data arrival is indicated by a certain message from the DSP application. On message reception the `MessageReceive` method of the respective `VSDSPBoard` instance is called in the scope of a thread running within the DSP board driver with high priority. This methods sets the `HostBlockRecvEvent` which indicates that a new host block has arrived from the respective DSP application. If any board did not send new data then a time out occurs and the acquire thread terminates. However, if all boards deliver their host block the acquire thread fetches the data from all DSP boards and merges it into a single stream within method `FetchHostBlocks`. The merge operation is shown in Figure 6.33 and the reordering in Figure 6.34. The data blocks are then processed in the `ProcessingPreWrite` method, then written to disk, and again processed in the `ProcessingPostWrite` method. The actual processing performed in both methods depends on the storing mode and Table 6.6 shows the different cases. Data is written to disk with the `FileWriter` in method `WriteDataBlocks`. The `SiteGroupList` in `FileWriterParameter` contains the parameters which determine subsampling and filter settings for the acquired data.

### Data reception

This section addresses the merging and reordering of raw data from multiple DSP boards illustrated in Figure 6.33 and Figure 6.34. As described in Section 6.2.1 the data transfers is organized in host blocks and packets. The data arrives at the host PC in the shared memory area of each DSP board. A host block is divided into packets, and each packet has a header and a data part. In Figure 6.33 the header is shortly represented by its packet id, but contains some additional information. The packet id is used to detect packet losses. If the host was for some reason not able to fetch the data in time, packets might have been overwritten by the DSP application. In this case the sequence of packet ids will not be steadily increasing, and there will be a detectable gap. The data part of the packets contains the acquired values organized as shown by the schema at the bottom of Figure 6.33. For each point in the time the sample values from all 32 channels are stored sequentially in memory. The acquisition DSP board acquires the 32 channels at 16 bit. Therefore one sample point has 64 bytes. The data part of each packet contains  $N$  sample points. First we merge the packets into a data buffer so that the the  $N$  sample points from three DSPs form a block in memory. Then this block is reordered such that each sample

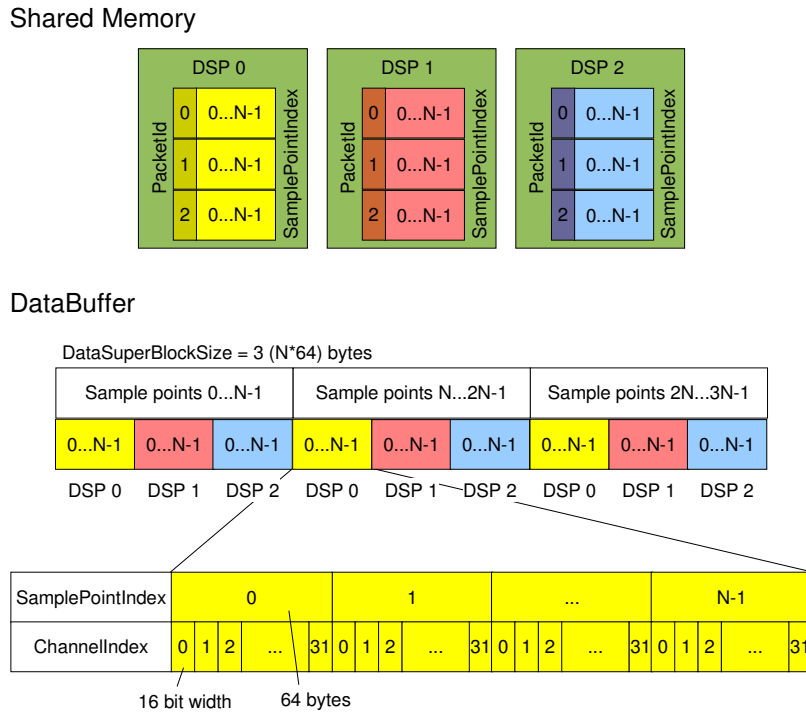


Figure 6.33: Merging of raw data from three DSP applications.

point forms single block in memory. In the actual implementation these operations, i.e. the merging and the reorder, have been combined into one assembler routine which performs both operations at the same time. This operation is one of the most time-critical ones and has been optimized using MMX extensions of the CPU. The optimized version is about 12 times faster, than a straight implementation with standard `memcpy`. The fast implementation is essential for the acquisition of data from multiple DSP boards and has been successfully used to acquire data from 128 channels with a Pentium III, 1GHz at 50 kHz per channel to disk.

The host application received the data acquired or computed by the DSP applications over the PCI bus. The reception of data is driven by events, which are generated within the vendor software library every time a DSP application sends a notification over the mailbox mechanism. If the message indicates that a new host block has been stored in the shared memory, then the acquire thread of the host application copies the packet data parts of all host blocks in shared memory into the main data buffer. Figure 6.33 and Figure 6.34 illustrate this operation which includes two steps, i.e. merging the data from multiple boards and reorder it such that it has the appropriate format for further usage by the host application.

### Data distribution for visualization

The acquired data and the results of the analyzers can be visualized by different means. However, the data must be provided to each of the visualization modules. The central data structures are `VSRingBuffer` and derived from that class the class

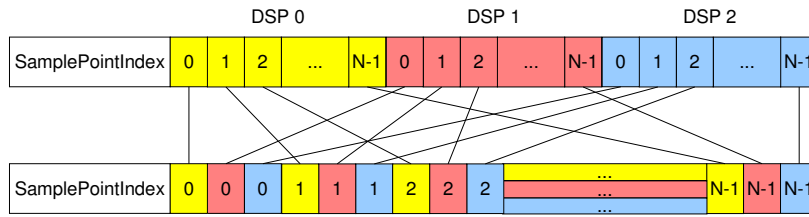


Figure 6.34: Reorder of acquired data after copy from shared memory, which is necessary if acquisition uses multiple DSP boards.

VSDataBuffer. The class VSRingBuffer implements a ring buffer. Data can be added or removed at head and at the tail of this particular ring buffer, which is used during forward and backward playback from disk. Counters and indices of VSRingBuffer are byte oriented. VSDataBuffer divides the actual buffer into data blocks with all of them having the same size at construction of an instance. Through the class each data block can be addressed by a global 64 bit index. The class provides a mechanism to keep track of already read data and newly added data. In Figure 6.35 a class diagram illustrates the features of VSDataBuffer and VSRingbuffer. The data is distributed to different visualisation modules by class VSDataBufferView which is associated with VSMModule. Details of both classes are shown in Figure 6.36. The class VSMModule holds views for all data buffers, and a reference to the current experiment parameters. With method Update of VSMModule the data buffer views are moved to the given position ViewIndexHead. This method is for example called after the view position has been changed in the browse bar. With the class VSDataBufferView one can extract a segment of data blocks from the associated VSDataBuffer instance. Hereby, it supports decimation of data in two modes, i.e. nearest neighbor and maximum over the decimation interval. VSMModule provides an interface for the main dialog to all visualization modules and visualization modules can execute methods of the main dialog through the VSMModule class. Thus, each visualisation dialog has an own instance of VSMModule. On the other hand the main dialog maintains a list of VSMModule instances which is used to triggered different actions on all modules. If for instance the visualization modules need to be updated the method the main dialog calls the Update method of each VSMModule instance in the list. The same mechanism is used for Allocate, Deallocate, and Reset.

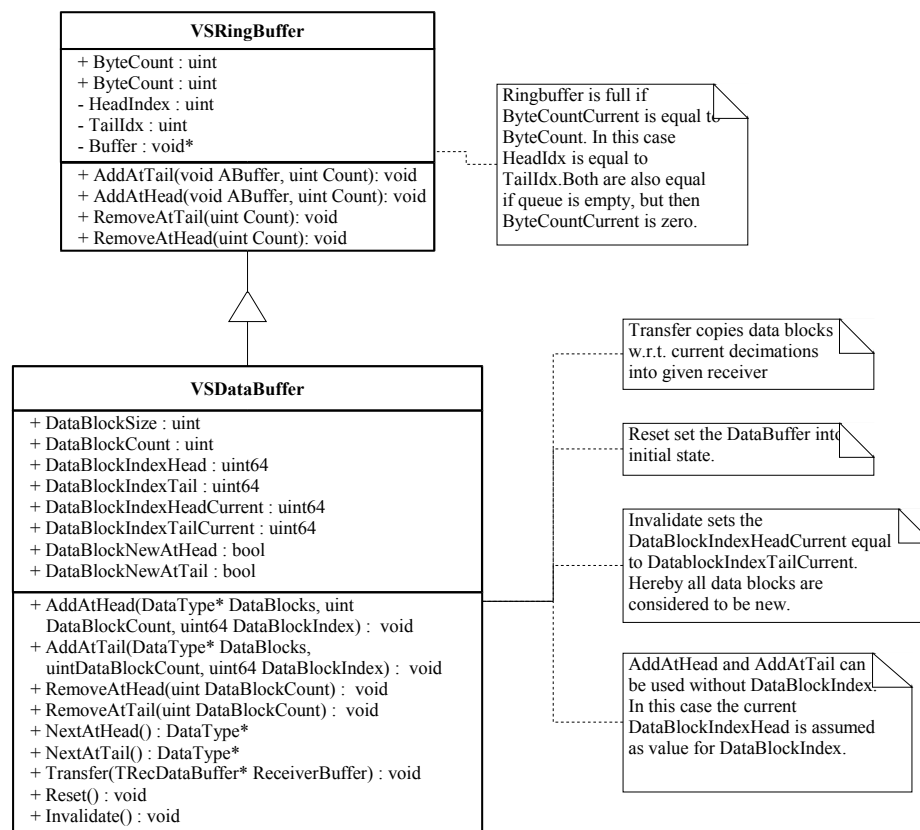


Figure 6.35: Class diagram for VSRingBuffer and VSDataBuffer

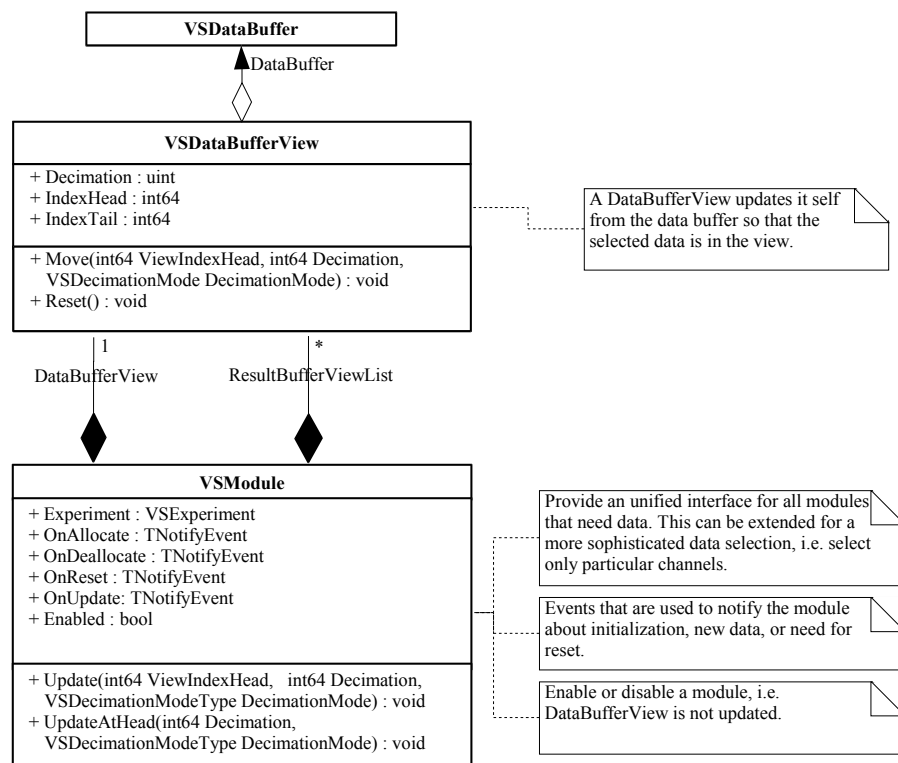


Figure 6.36: Classes for data distribution

# Chapter 7

## Biological validation

The following paragraphs illustrate the successful validation of the data acquisition system in different experimental environments. In particular the data acquisition system variants with 32 and 64 channels were utilized by the experimental partners of the VSAMUEL project. At the Istituto Nazionale Neurologico "Carlo Besta" (INNCB), Milano, Italy it was used by Gerardo Biella, who works in the group of Marco De Curtis, to record from the cortex of guinea pig brain maintained in vitro [4]. The group of Eric de Schutter, especially Antonia Volny-Luraghi and Patriq Fagerstedt, at the Laboratory of Theoretical Neurobiology, University of Antwerp (UIA), Belgium used it to record from the cerebellar cortex of anesthetized rats [26, 25]. These two experimental scenarios will be described in more detail. However, the system was also evaluated in some more environments: At the Center for Sensory-Motor Interaction (SMI) the system and particularly the probes have been tested in peripheral nervous system [60] and in the rat cerebellum [59] together with Winnie Jensen and Ken Yoshida. A special variant of the data acquisition system with different A/D conversion modules has been used to record electromyographic signals in a classical conditioning experiment [57]. Another early test of the system has been performed at the Brain Research Institute, University of Bremen, together with Heiko Stemman working in the group of Winrich Freiwald. The data acquisition prototype was used to record from the visual cortex of a rat's brain in an experiment setup similar to the one described in [34].

### 7.1 Recording from the guinea pig brain

The experiments at the INNCB were focused on the characterization of the distribution pattern of evoked activity and oscillatory activity at 25-50 Hz, the so-called *gamma* activity in the parahippocampal region. *Gamma* oscillations in this region are assumed to be involved in the cellular and network mechanisms that control the process of memory consolidation and retrieval performed by the medial entorhinal cortex [104]. Furthermore, they are also associated with cognitive tasks such as attention and active sensory processing [94]. Thus, it is a matter of particular interest to measure both population field potentials and single unit activity at multiple sites in this cortical region during *gamma* activity. The field potentials reflect the expression of the synchronous synaptic and massive activation of a homogeneous

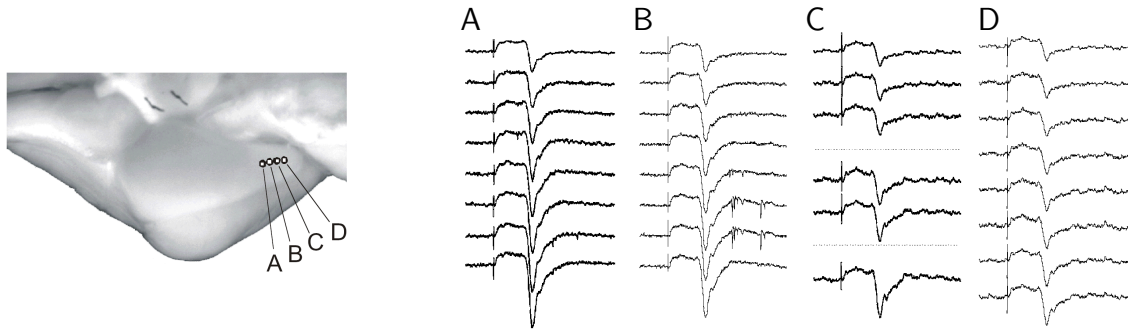


Figure 7.1: Field potential profiles obtained in the medial entorhinal cortex with a 32-channel 4-shafts E3 silicon probe, recorded with the VSAMUEL data acquisition system. The four laminar profiles (A, B, C, D) were simultaneously recorded in response to stimulation of the lateral olfactory tract. Traces were recorded at different depths separated by  $200\ \mu\text{m}$ , i.e. the c/c distance between recording sites within a single shaft of an E3 probe. The position of the 4 probe shafts (A to D) which are also separated by  $200\ \mu\text{m}$  is shown in the micro photograph of an isolated guinea pig brain illustrated on the left panel [17].

population of cells in the range of  $0 - 100\ \text{Hz}$ . On the other hand single unit activity expresses the discharge properties of single cells during the population event in a range of  $300\ \text{Hz} - 10\ \text{kHz}$ . Collecting both kind of signals simultaneously allows deeper insights into the functional organisation of the brain.

Using VSAMUEL probes, which have the respective impedance characteristic, and the VSAMUEL data acquisition system with its high bandwidth per channel combined with a high channel count and 16 bit A/D sampling resolution it is possible to record both population field potentials and single unit activity simultaneously at different sites.

### 7.1.1 Isolation of the guinea pig brain

At the INNCB the VSAMUEL data acquisition system was used to acquire neural activity from an isolated guinea pig brain, which is maintained in vitro by arterial perfusion, a technique which is described in detail in [79, 18, 16]. After anesthesia the animal was perfused with a cold, complex saline solution through the aorta to reduce brain temperature during the dissection. The brain was then carefully and quickly removed and transferred to a perfusion chamber under hypothermic conditions ( $15\ ^\circ\text{C}$ ). Here perfusion of the brain was ensured through the basilar artery at a rate of  $5.5\ \text{ml/min}$ . The experiments were carried out at  $32\ ^\circ\text{C}$ . At this, extracellular recordings were performed in different brain regions with VSAMUEL silicon probes.

### 7.1.2 Recordings in the medial entorhinal cortex

Recordings were performed with probe E3 (Figure 6.2) in a subfield of the parahippocampal region, the medial entorhinal cortex. Table 6.1 in the previous chapter gives an overview of the specific probe parameters. In the experimental setup the

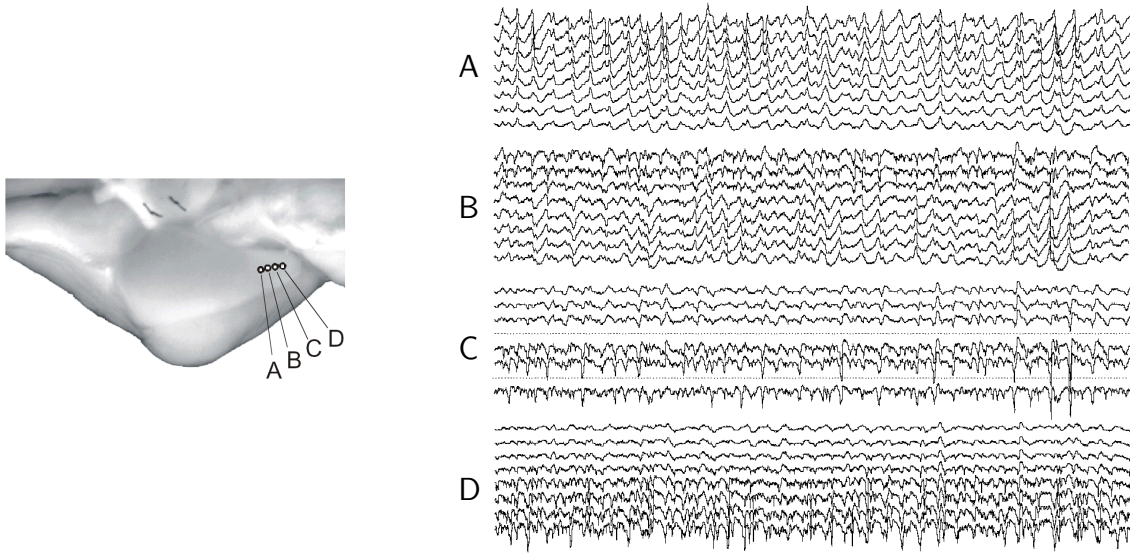


Figure 7.2: Simultaneous recording from 4 sites (A to D, separated by 0.5 mm) in the medial entorhinal cortex with an ACREO-E3 silicon probe after induction of fast *gamma* oscillations by arterial perfusion of the isolated guinea pig brain preparation with 50  $\mu\text{M}$  carbachol. Each shaft featured 8 recording sites separated by 200  $\mu\text{m}$ . Leads 4 and 7 on shaft C were not connected. [17]

signal was amplified by a factor of 1000 and filtered with an analog band pass filter with cut off frequencies at 0.1 Hz and 5 kHz and acquired at 20 kHz sampling rate for offline analysis.

Figure 7.1 shows typical responses in olfactory/limbic cortices evoked by bipolar stimulation of the lateral olfactory tract [2]. Each column illustrates the potentials recorded at 8 different depths. The order of the signals from top to bottom corresponds to an anatomical order from surface to depth. Recording sites within the shafts are separated by 0.2 mm from each other as well as the shafts. Note that the two dotted lines in the illustration indicate two recording leads on shaft C, which were not correctly connected on the used E3 probe.

Figure 7.2 illustrates a typical recording from 4 sites after induction of fast *gamma* oscillations obtained with E3 probe. The *gamma* activity at 25–40 Hz in the medial entorhinal cortex has been invoked by pharmacological stimulation with the muscarine receptor agonist carbachol (50  $\mu\text{M}$  applied by arterial perfusion) [100, 19, 20]. The simultaneous recording from 4 different shafts separated by 0.2 mm and at the same time also at 8 different depths allowed to verify the depth reversal of activity, which can be observed across the cortical depth for locally generated potentials [3]. Therefore the measured depth reversal of the potential allows to distinguish whether the activity is generated by the cortex or by a far field volume conducted from a distant source [17, 43].

Figure 7.3 shows an example where also single unit activity has been recorded in the same experimental setup illustrated with Figure 7.2 and Figure 7.1. The filter settings (high-pass filter at 0.1 Hz and low-pass filter at 5 kHz) allowed to record slow events as well as fast activities generated by single cells. The respective spikes



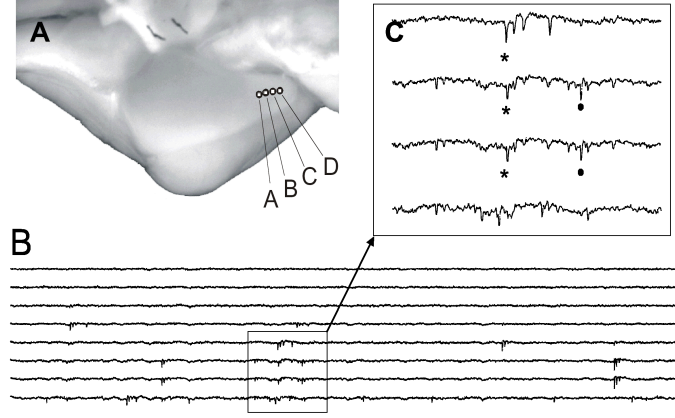


Figure 7.3: Single unit recordings performed with a M2 probe. The traces from 8 channels in a single probe shaft are illustrated in B. The unitary activities of two cells outlined by the box in B are illustrated at fast time resolution in the upper right panel C proving the wide band acquisition purposes of the VSAMUEL data acquisition system. [17]

from two different cells are displayed in the close up in panel C, marked with an asterisk and a filled circle, respectively [43]. Actually, this proves the wide band acquisition purposes of the VSAMUEL data acquisition system.

The experiment which is illustrated with Figure 7.4 is used to investigate the vertical organization of the cortex, i.e. interaction between layers within a restricted patch of cortex. For the vertical study it is crucial to explore at the same time the entire depth of the cortex (1.2-1.4 mm). To validate the use of the 32 channel VSAMUEL data acquisition system laminar profiles were performed in the entorhinal cortex and dentate gyrus of the hippocampus following electrical stimulation of the lateral olfactory tract (LOT), which is indicated by an arrow at the left of the field potential profile in Figure 7.4. The stimulation artifact is followed by the response measured in the dentate gyrus (DG) of the hippocampus. Panel A shows superimposed traces which represent the average of 7 responses simultaneously recorded at 20 adjacent recording sites, selected out of the 32 available channels. Below in panel B the traces are separated with respect to the recording depth. The photograph of the right shows the electrode track reconstructed on a histological sections cut after fixing the brain with a mixed aldehydes solution. Finally, panel C shows the contour plot of the sink/source distribution obtained by performing current source density analysis on the laminar potential profile. Sinks and sources are illustrated by continuous and dotted lines, respectively. The current source density is calculated as the discrete second-order differentiation of the field potential function  $\psi(z, t)$  along the depth  $z$ :

$$\text{CSD}(z, t) = \sigma \frac{2\psi(z, t) - \psi(z - N\Delta z, t) - \psi(z + N\Delta z, t)}{(N\Delta z)^2} \quad (7.1)$$

where  $\Delta z$  is the distance between the recording sites and  $t$  is the time variable. Conductivity  $\sigma$  is assumed to be constant over depth.

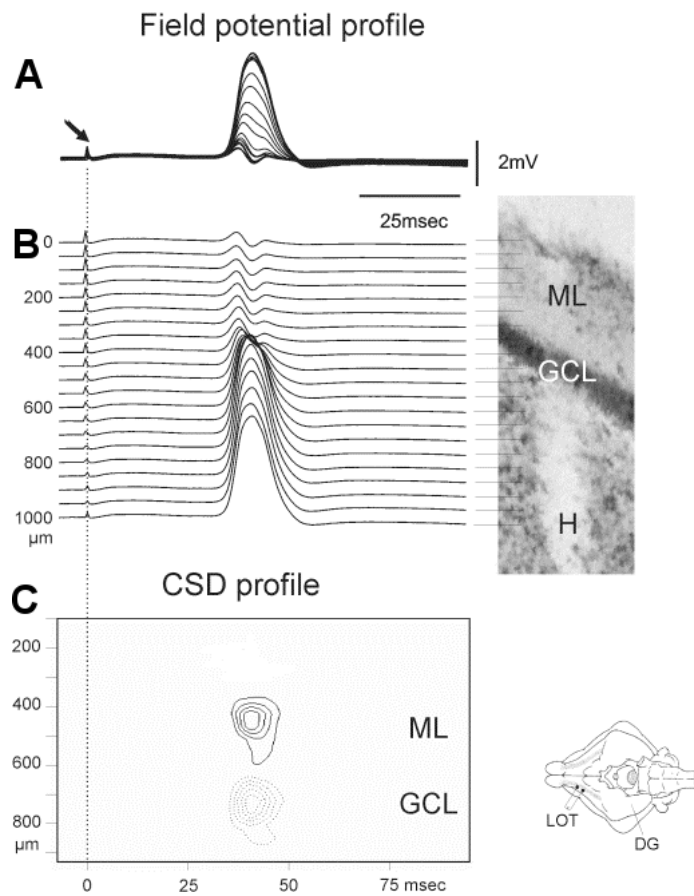


Figure 7.4: Field potential profile and obtained with a 32-channel ACREO silicon probe in the dentate gyrus (DG) of the isolated guinea pig brain evoked by stimulation of lateral olfactory (LOT) tract (position of DG and LOT are indicated in schema in lower right corner).

Current source density analysis identifies and localizes the spatio-temporal distribution of current sources and sinks, which are in turn responsible for the generation of field potentials. It provides an indication of the neuronal circuit activated in this area. In this particular case the CSD analysis confirmed the presence of a CSD sink in the molecular layer of the dentate gyrus (ML), where the olfactory input carried by the afferent fibers of entorhinal cortex neurons terminate. A source was observed in the granule cell layer (GCL).

Summing up the VSAMUEL data acquisition system has clearly proved its validity in the isolated Guinea pig system for slow field potentials as well as fast action potentials, thus exceeding the current industry standard provided by commercially available systems [39, 96, 24].

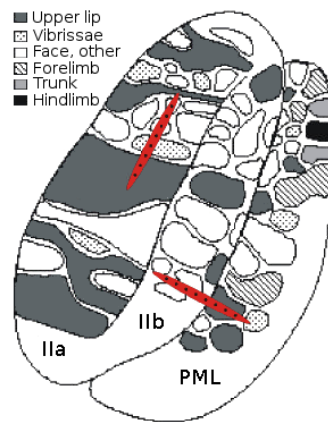


Figure 7.5: Receptive fields in the Crus II and PML (paramedian lobule) regions of the cerebellar cortex related to tactile stimuli. The insertion positions of the VSAMUEL probes are indicated by the elongated red areas.

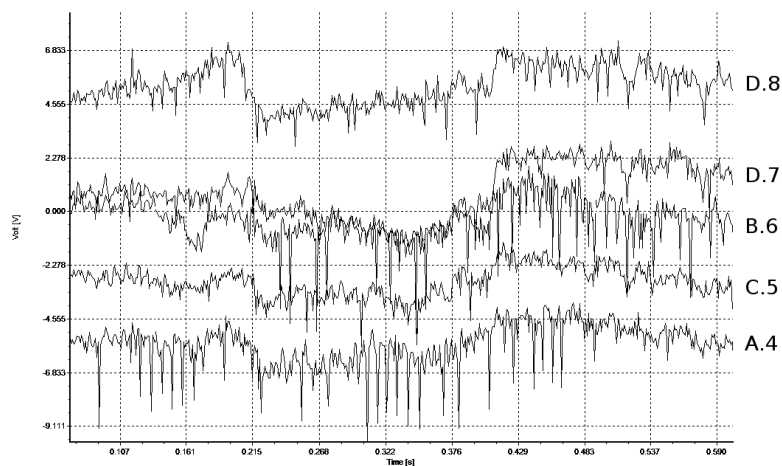


Figure 7.6: Five multi unit recordings representing spiking cerebellar neurons in response to tactile stimulus to the upper lip of the face recorded with an E1 probe. The names on the left side indicate the recording site by shaft name and site number (numbering start at the tip of the shaft)

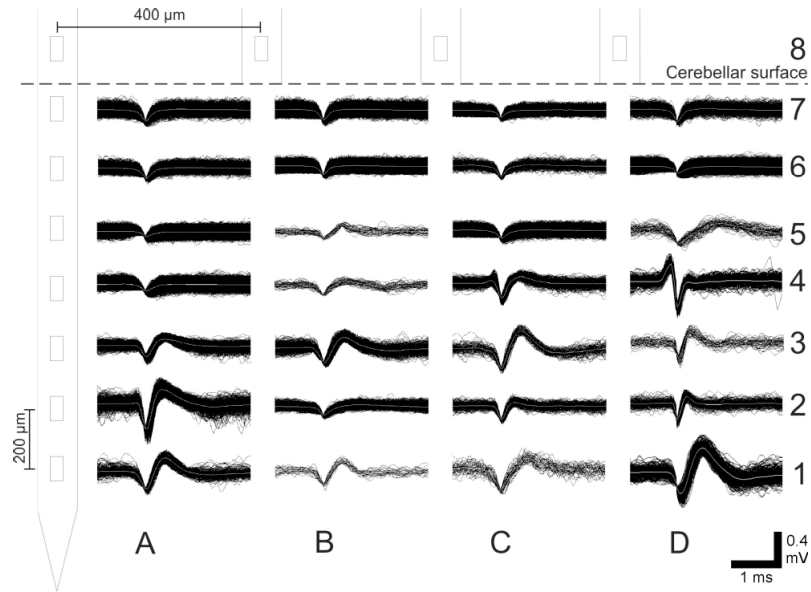


Figure 7.7: Extracellular action potential waveforms of multi- and single unit recordings of spontaneous activity from the cerebellar cortex Crus IIa of an anesthetized rat using the E2 probe with four shafts each having 8 recordings sites.

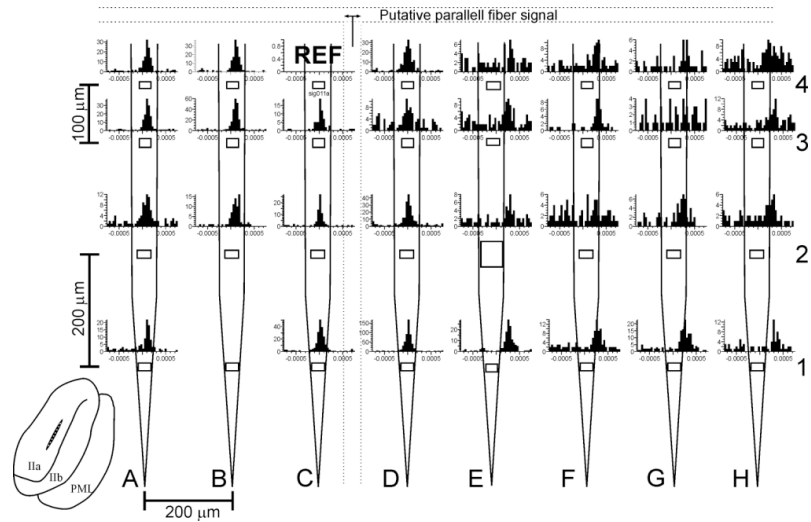


Figure 7.8: Spike interval histogram from multi- and single unit recordings of activity evoked in the cerebellum from tactile stimulation of the facial skin of an anesthetized rat using an E3 probe. Recording position in Crus IIa is indicated in lower left sketch.

## 7.2 Recordings from rat cerebellum

The experiments performed by the group at the Laboratory of Theoretical Neurobiology of the University of Antwerp are focused on the organisation of the oscillatory network of cerebellar Golgi cells and granule cells. The cerebellar granule cells form the largest neurone population within the brain. Their axons run as parallel fibers along the coronal axis (see also Figure 2.4). It is expected that this anatomical structure leads to a one-dimensional spread of excitation, which in turn results in a beam-like propagation of excitation, i.e. synchronized firing of Golgi cells along the parallel fiber [101]. The experiments aim to verify this expectation with recordings from the rat cerebellum during tactile stimulation.

In order to study the interaction between local circuits in different receptive fields it is required to simultaneously record the single unit activity and local field potentials of multiple nerve cells in different layers with high temporal and spatial resolution [26, 25]. Considering these constraints silicon multichannel probes seem to be the ideal tool to record from cerebellum. However, the cerebellum is a very damageable tissue and the lack of publications about recordings from cerebellum using silicon probes of older fabrication than the VSAMUEL probes appears to be quite a striking evidence for the difficulty of this task. The VSAMUEL probes E1-6, especially their tips, are particularly designed to reduce fatal dimpling of the cerebellar surface [59]. Furthermore, the spatial distribution of the recording sites is adapted to record in the layers of the cerebellar cortex. The probes were implanted along and perpendicular to the transverse axis, i.e. the parallel fiber axis as indicated in Figure 7.5.

Figure 7.6 shows five multi unit recordings acquired with an E1 probe in response to tactile stimulus to the upper lip of the face. The recording quality was excellent and single unit isolation should be possible with appropriate spike sorting algorithms [27]. It is remarkable that up till now the VSAMUEL probes are the only silicon array ever reported to measure in the rat cerebellum.

Of particular interest is the relation between simple spike activity of Purkinje cells and the synchronized firing of Golgi cells and granular cells. Figure 7.7 shows examples of captured multi- and single unit action potentials during spontaneous activity recorded from the cerebellar cortex area Crus IIa (see also Figure 7.5) using an E2 probe, which has four shaft with eight recording sites on each shaft with spacing as indicated in Figure 7.7. The recorded spikes were classified using standard spike sorting method, i.e. cluster analysis using the first two principle component coefficients (see also Section 5.4.2). Based on single unit firing patterns and action potential shapes it was possible to identify several supposed Purkinje and Golgi cells.

With the probe placed along the parallel fiber axis it should be possible to find experimental evidence for the expected beam-like propagation of excitation along the parallel fiber. Figure 7.8 shows interval spike histogramms which result from recording with an E3 probe along a parallel fiber in a receptive field of the rat's facial skin in Crus IIa. The E3 probe has eight shafts separated by 200  $\mu\text{m}$  and four recording sites per shaft with either 100  $\mu\text{m}$  or 200  $\mu\text{m}$  center to center distance. Using the action potentials recorded at site C4 as trigger (marked with

REF in Figure 7.8 a timeshift of peak activity can be observed in both directions of the probe, indicating a transmission of activity moving in both directions along the parallel fiber. The combination of different stimulus protocols and different recording sites within one or several receptive fields on the cerebellar surface yields useful data for detailed analysis of the dynamics of cerebellar networks during sensory processing [43].

The VSAMUEL data acquisition system based on DSP boards and silicon probes has successfully proved to be a complete integrated neurophysiological tool for recording in different regions of the nervous system.

# Chapter 8

## Discussion

This thesis has presented the VSAMUEL data acquisition system prototype for neuronal recordings. The first chapters of the thesis gave an overview of the field of application, i.e. the nervous system of mammals, and an overview of neural data acquisition including basic principles of bioelectrical sensing. This was followed by a review of commonly used probes and in particular by a presentation of the design of silicon multichannel probes developed within the VSAMUEL project. The next part described the signal processing methods, namely the lifting scheme to implement a real time wavelet decomposition on the used digital signal processor and how the wavelet decomposition can be used to implement various methods. This included the low, band, and high pass filtering and also a compression and denoising schema. Furthermore, the possibilities to use the wavelet coefficients for spike detection and spike sorting have been evaluated. The thesis is completed by a description of the data acquisition hardware and software and its adoption by the experimental partners of the VSAMUEL project.

The VSAMUEL data acquisition system allows simultaneous recording from 128 channels at 50 kHz per channel with 16-bit A/D conversion resolution. Having these features the system exceeds other systems which are commercially available for neural recordings [39, 96, 24]. The acquisition system provided by MultiChannelSystems (MCS) has 128 or 64 channels and can sample at 50 kHz per channel but only with 14 bit A/D resolution. Plexon provides different systems, one can acquire 16 channels at 50 kHz, but only with 12 bit resolution, another acquires data with 16 bit A/D resolution, but only supports 16 channels. Yet another system from Plexon supports 64 channels but also only with 12 bit A/D resolution 16 kHz sampling rate. Thus, summarizing the architecture of the VSAMUEL data acquisition system, it provides an outstanding performance in terms of sampling rate, A/D resolution and channel count.

Having 16 bit A/D resolution allows to record local field potentials and action potential simultaneously. Since the amplitudes of recorded local field potentials are

	VSAMUEL	MCS	Plexon	AlphaOmega
Channel Count	128	64/128	64/16/16	16
A/D resolution [bit]	16	14	12/12/16	12
Sampling rate [kHz]	50	50	16/50/16	50

by a factor of 10-20 times larger than extracellularly recorded action potentials, the amplification of the neural signals must be chosen such that the local field potentials fit into the input range of the A/D converters. As a consequence the amplitudes of the action potentials remain small, but with 16 bit A/D converters it is still possible to detect them and classify their shapes.

The core component of the VSAMUEL data acquisition system is the M67 DSP board (Innovative Integration) equipped with Texas Instruments DSP C6701 and A/D converter modules, which is used for data acquisition and also for data processing. Each of the four boards in the system processes a maximal number of 32 channels. During development of the system some drawbacks using DSP boards for the signal processing became apparent, which should be mentioned. The M67 boards are off-the-shelf products, still they are produced in small quantities and are rather expensive. The software development for the DSPs requires special experience and a special development environment. Especially, the implementation of highly optimized signal processing routines needs a lot of time, thus the development of new features and the software maintenance for the DSP board is expensive. However, software development for DSP becomes easier with more support from tools and libraries which have been developed recently, like e.g. the "Target Support Package" from "The MathWorks". Although the used DSP boards were equipped with DSPs from the highest performance category (C6701 @160 MHz), they still were not sufficient to implement the processing chain completely on the DSP. Even with a lot of optimization efforts it appears not to be possible to reach the expected 50 kHz sampling rate while performing a db6 wavelet decomposition on 32 channels with a depth of 6. Then again it worked for up to a sampling rate of 44 kHz (Section 4.3.3) and with shorter wavelets like e.g. the db2 wavelet it was actually possible to reach the 50 kHz. With a new DSP generation the desired performance will surely become available.

The system was designed such that the data processing is distributed onto multiple DSP boards. As an advantage of this solution it is possible to implement reliable real time processing for a huge amount of data, which is in this extend not possible on a standard Windows based host PC. The number of channels per DSP board depends on whether it is equipped with one or two A/D conversion modules. Therefore the acquisition system can easily be customized to variants with lower channel count, which could start at 16 channels and increase in steps of 16 up to 128 channels. With growing performance on the host PC side, it will also be possible to exceed even the 128 channels by just using more DSP boards. Also, the employed A/D conversion modules utilize one A/D converter per channel allowing for a perfectly synchronized data acquisition across all channels and all boards, as no multiplexing is necessary and the M67 boards can be synchronized with each other. The M67 hardware provides also a digital I/O port, which could be used to record parameters from the experimental setup with high precision time stamps like e.g. a stimulation trigger together with a digital stimulation event code, which might be available from the stimulation generation device.

In order to minimize the effort to obtain decomposition implementations for different wavelets, a semiautomatic tool has been developed which generates source code that implements wavelet decomposition and reconstruction filter banks based



on the lifting scheme. It can generate C code and also optimized assembler code for Intel Pentium III and compatible processors, as well as for the Texas Instruments C6701 signal processor. This tool is supposed to support software developer when adding new wavelet filters to the software. As the tool supports assembler of different targets, the filter implementation can be generated for the DSP or for the host PC software.

Beside its pure recording capabilities the different online visualisation and analysis options of an acquisition system are important for its adoption as a neural recording system. The graphical user interface for data acquisition, data review, and data analysis has been developed with respect to a good design, usability, and speed. The main functionality for recording is available via large buttons in order to support operation with a touch screen. The VSAMUEL data acquisition system presents the multichannel data not only by means of a virtual scope view, but also using the blue plot view which allows to present multichannel data at a glance, in particular spatially related recordings. The online spectrogram is for instance useful to track down noise sources. Our system is designed to allow digital filtering and detection of action potentials in real time on a large number of channels and to give a direct feedback about quality of the recorded signal with the current probe position to the experimenter. Another aspect is the support for data handling which includes the possibility to review and analyze large amounts of multichannel data and also the possibilities to extract and export raw data or analysis results into separate files. The VSAMUEL data acquisition system uses a standard data format in order to allow the import into other software tools.

The processing of neural signals using wavelet based compression, denoising, and classification methods has been evaluated. It became apparent that the translation variance of the decimated dyadic wavelet transform prevents robust classification of spikes based on wavelet coefficients, when it is not possible to align the spike shapes before computing the wavelet decomposition. In a further development one could overcome this problem by applying the dual-tree complex wavelet transform [92] as described in Section 5.4.3.

At the beginning of this thesis it is noted that progress in neurophysiological research was accompanied by technological progress of measurement instruments. The VSAMUEL data acquisition system as a complete integrated neurophysiological tool for recording from a large number of sites in different regions of the nervous system has its share in this progress.

# Appendix A

## Microstructure techniques

### A.1 Metal evaporation

The substrate is placed in a high vacuum chamber at room temperature with a crucible containing the material to be deposited. A heating source is used to heat the crucible causing the material to evaporate and condense on all exposed cool surfaces of the vacuum chamber and substrate. The process is typically performed on one side of the substrate at a time. Typical sources of heating are: E-beam, Resistive heating, RF-inductive heating. In some systems the substrate can be heated during deposition to alter the composition/stress of the deposited material. Figure A.1 shows the metal evaporation when using E-beam as heating source. In a high vacuum area a hot filament produces an electron beam which focuses and lead into the hearth by magnets. The electron beam causes the evaporant to be deposited onto the substrate.

### A.2 Lift-off process

Forming metal, e.g. Pt, Ti, or Au, conductor traces and electrode sites by directly etching away a respective metal layer is difficult due to the high chemical stability of these metals. The etching can attack the silicon and SiO<sub>2</sub> layers and can therefore destroy the already grown structures. In order to overcome this problem the lift-off method is used [98]. It has five steps illustrated in Figure A.2. First a photoresist material is deposited on the so far created structure and is patterned. Then an etching is performed. Hereby, all structures which should be left unchanged are protected by the resist. Afterwards the metal Ti/Pt is deposited onto the whole structure and rests on the photoresist or on the underlying structure where the photoresist has been opened. In the final step is to remove the photoresist in an ultrasonic acetone bath. The metal remains only at the locations where the photoresist was opened at the beginning.

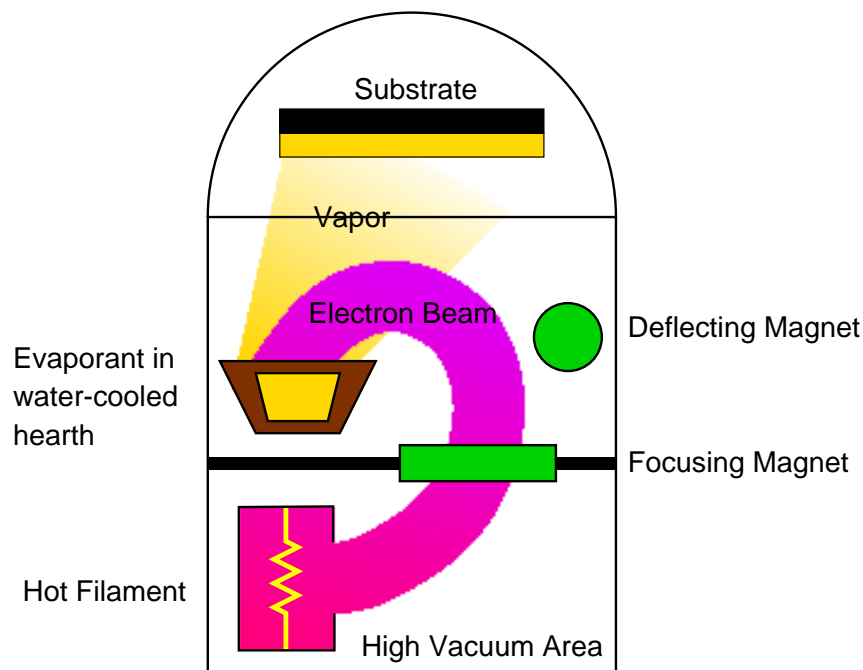


Figure A.1: Ebeam evaporation process

### A.3 PECVD

Plasma enhanced chemical vapor deposition (PECVD) is performed in a reactor at temperatures up to 400°C. The deposited film is a product of a chemical reaction between the source gases supplied to the reactor. A plasma is generated in the reactor to increase the energy available for the chemical reaction at a given temperature. The process is typically performed on one side of the substrate at a time.

### A.4 Deep RIE

Deep reactive ion etching is used to etch deep cavities in substrates with relatively high aspect ratio. Most systems utilize the so-called "Bosch process", in which a fluoropolymer is used to passivate the etching of the sidewalls. Typical aspect ratios of 10-20 can be achieved.

Isotropic etch, (e.g. HF etch), is a wet and dry etch processes in which the undercutting is equal to the etch depth.

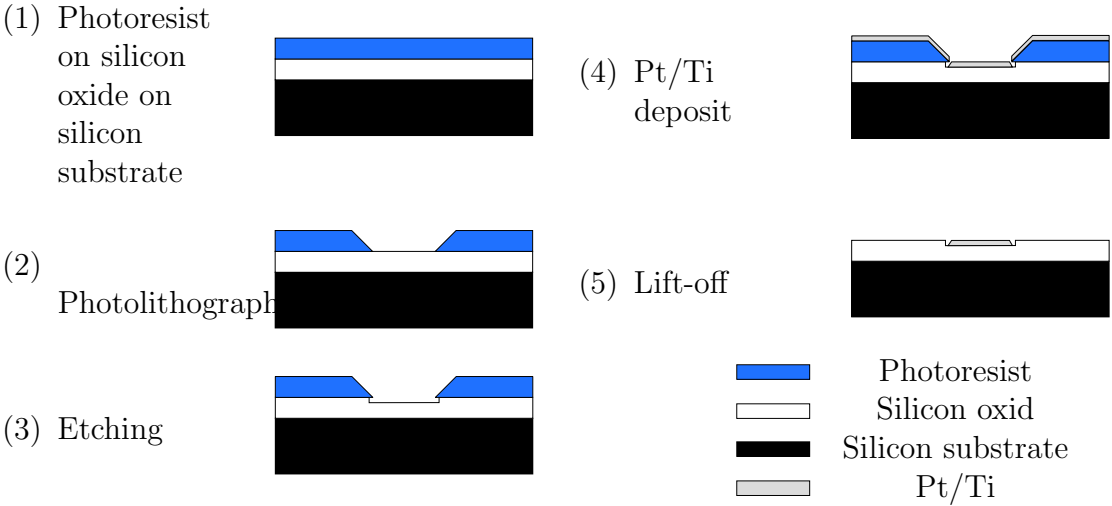


Figure A.2: Lift-off technique which is used to create the conductor traces and electrode sites

# Appendix B

## M67 and Ad16 extensions and modifications

### B.1 SyncLink and ClockLink

For synchronization of multiple M67 boards a special cable (connected on JP32) is used, which provides the clock signals of the master board to the slave boards. The cable has two parts, the SyncLink part and the ClockLink part. SyncLink is a bus with three pins and is supposed to work for slow clock rates below 1 MHz. Output of timer0 and timer1 can be connected to SyncLink bus pin 0 and pin1 by setting the SyncLink signal selection register at 0x014E8000. The ClockLink can be used for high clock rate up to 80 MHz and consists of four pins: Two pins for output and two pins for input. The output pins of the master board **must** be connected to the input pins of the slave boards, otherwise either board or both might be damaged. The ClockLink input pins are terminated, i.e. connected via an 100  $\Omega$  resistor (R94 on M67 Rev G). If more than two boards are synchronized then this resistor must be removed from the slave boards which reside between master and last slave board. It is recommended to use twisted pair cables for the ClockLink signals in order ensure signal quality over longer distances, which should not exceed 60 cm for 80 MHz signals.

We use the ClockLink part of the cable to transmit the output of the DDS timer to the slaves boards. The AD16 board requires  $128 \times$  sampling rate as clock input for its ADCs, i.e. for a sampling rate of 50 kHz a clock signal with about 6 MHz is transmitted over the ClockLink.

The slave/master setup with DDS clock being provided from master to slaves is implemented as follows:

```
%if (master) {  
% // configure ClockLink as "DDS Master"  
  
% // SyncLink Signal Selection Register  
% // bit 0 is set to 1 (Master)  
% Periph->SyncLink[0].Word = 1;
```

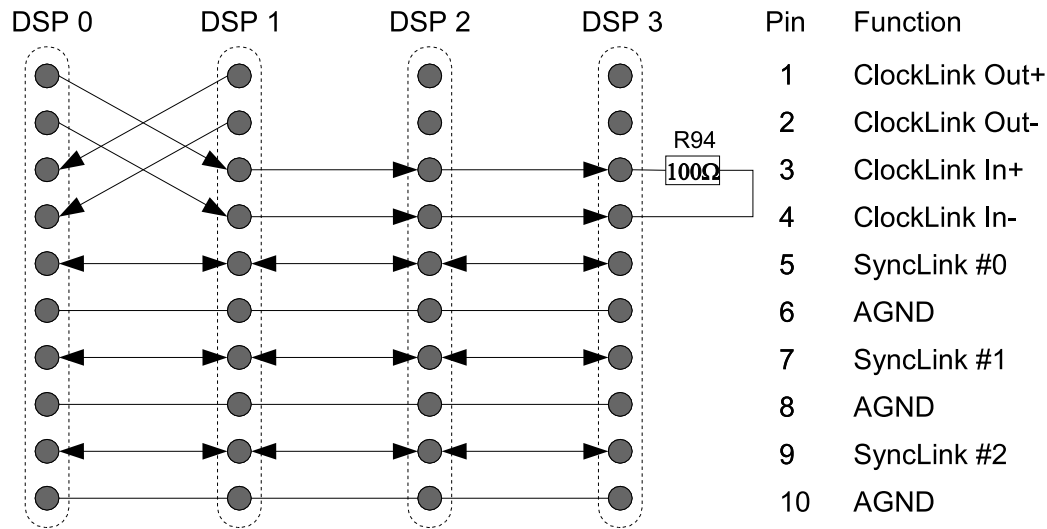


Figure B.1: SyncLink/ClockLink cable pin documentation. Note the  $100\ \Omega$  resistor which is integrated on the M67 board (Rev G) as resistor R94. This resistor terminates the ClockLink. But since more than one slave board is connected to the ClockLinkOut pins like in this example, resistor R94 needs to be removed on the slave boards DSP 1 and DSP 2.

```
// SyncLink Destination Selection Register
// activate ClockLink
Periph->SyncLink[1].Word = 0;

} else {
    // configure ClockLink as "DDS Slave",

    // SyncLink Signal Selection Register
    // bit 0 is set to 0 (Slave)
    // bit 3 is set to 1 (get DDS from ClockLink)
    Periph->SyncLink[0].Word = 8;

    // SyncLink Destination Selection Register
    // activate ClockLink
    Periph->SyncLink[1].Word = 0;
}
```

## B.2 Synchronizing the AD16 boards

The synchronization of the ADCs of multiple AD16 boards is established by the so called AdcSync cable. A single wire which connects the SyncOut pin of one AD16 board with the SynIn pin of the next AD16 board. All AD16 boards which should be synchronized are connected in a row by the AdcSync cable. In order to activate synchronization the Innovative Integration library function `AD16_sync` must be called for the first board in the row. The current implementation assumes that this AD16 board resides in omnibus slot 0 of the master board.

## B.3 Trigger Input

In many experiment settings a stimulus is presented to test subjects. The data acquisition system should be able to acquire the time stamps of stimulus presentation together with the measured data. This requires that a stimulus generator can output a TTL trigger signal, when it presents a stimulus. This trigger signal must be connected to the M67 data acquisition hardware and then it possible to acquire the stimulus time stamps. There are different scenarios what kind of data is available from a stimulus generator. It might have an digital port, where markers of different stimuli are generated. How to use such data is sketched by the following paragraphs.

The external interrupt source line 3 of the C6701 which can be accessed on a omnibus connector pin can be utilized as trigger input line. A physical access point must customized by connecting pin 50 on the digital IO (DIO) port and pin 29 on JP 24 on the M67 board.

In order to support multiple external trigger lines, a circuit is required which combines the different trigger input line such that a single interrupt source is generated for the DSP and for the DIO port clock input. In Figure B.2 this circuit is denoted with  $\geq$ . Additionally, the "Edge to 1" circuit generates a logical 1 for each active trigger input line and it is connected to the pins of the DIO port. A falling edge on the DIO port clock triggers that the DIO port latches all input lines into its internal register. The digital input values must be stable for 10ns for latching. The same falling edge should trigger an interrupt service routine in the C6701 which fetches the data from the internal DIO port register. This routine also stores a time stamp which is mapped onto a sample point index. As the time stamp is fetched inside the interrupt service routine, the time resolution depends mainly on the interrupt latency. According to the C6701 documentation it takes 13 CPU cycles until an interrupt service routine is executed. At 160MHz this corresponds to 87.5ns, which is the ideal case. Actually, the latency depends on some other factors:

- Maximum number of instructions for which interrupts might be disabled, which happens in compiler generated code that utilizes software pipelining (see also compiler option `-mi`).
- DMA transfers, which have higher priority than the CPU to access memory, might increase the latency.

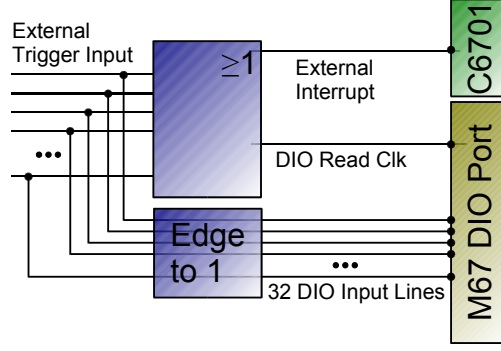


Figure B.2: External trigger

- Interrupt service routines, which do not allow other interrupts, can cause latencies, that are as long as the execution time of the respective routine (see also function `enable_nested_interrupts`)

It is possible to have an interrupt load of 1.5MHz (due to GBM/II) if the interrupt service routine is very fast. This suggests an upper latency bound of 660ns, if the application is optimized for minimal interrupt latency.

If a stimulus generator is utilized which supports different types of stimuli and which can drive a digital output port having respective markers for each stimulus, then these markers can be read using the DIO port, too. The data acquisition system is able to record the data and the stimuli markers synchronously.

Currently, the data acquisition application supports one trigger timestamp per 64 sample points. The maximal time resolution for trigger inputs therefore depends on the sampling rate. If data is acquired at 50kHz then the maximal resolution is 1.28ms which corresponds to about 780Hz. This restriction is due to the packet header format which allows only one trigger timestamp per packet. Extending the header to more trigger timestamps would directly increase the resolution.

In addition to a mapping of trigger timestamps to a sample point a higher resolution might be desired. This is supported by using the internal high resolution timestamp (40MHz) which is also stored in the packet header. The host application can resolve the sample point index and the high resolution offset into a global high resolution time stamp with up to 25ns resolution.

The interrupt service routine, which fetches the data from the FIFOs of the AD16 modules, updates a base time stamp each time when it is executed. If a trigger has been detected and the respective time stamp has been stored, then the routine can compute an high resolution index offset  $t_{\text{offset}}$  as the difference between the base time stamp and trigger time stamp. The offset in turn can be mapped onto a sampling point index  $t_{\text{index}}$ :

$$t_{\text{index}} = \left\lfloor t_{\text{offset}} \frac{f_{\text{sampling}}}{f_{\text{timer}}} \right\rfloor.$$



# Appendix C

## Realtime operating system

### C.1 Overview

Realtime Operating Systems (RTOS) are special kinds of operating systems. At large an RTOS provides a robust, high performance software architecture on which higher level software applications can be developed. In contrast to a general purpose operating system (GPOS) an RTOS provides deterministic behaviour of basic services, which usually include

- task scheduling,
- intertask communication,
- synchronization,
- dynamic memory allocation, and
- timer services.

Deterministic behaviour is particularly important for embedded systems interacting with the real world. Such systems need to react on events within a strict time deadline, which is usually short, regardless of whatever happens in the system, i.e. how many other tasks are running or messages are transmitted. In an RTOS the time and resources needed to execute an operating system service are known and they are limited, such that there are no random timing components. Random elements in service times would cause random delays in application software and would then make the application randomly miss real-time deadlines, which is unacceptable for a real-time system. GPOSs (such as Windows or Linux) are usually non-deterministic, e.g. moving a window influences the response or execution time of other applications. Furthermore, timing resolution is not as high as often necessary in embedded applications.

There are a number of commercial benefits when using an off the shelf RTOS for development of a real time application. Instead of implementing proprietary equivalents of the basic operating system services one can use proven and well documented functionality of the RTOS. Moreover, having such a foundation for an application

improves portability and maintainability of the application. Furthermore, coordination of several developers is easier, because an RTOS usually provides a message passing architecture which helps dividing a larger application into smaller building blocks that can be implemented by separate developers and communicate through the existing messaging routines. Some RTOSs provide features that allow extensive performance measurements within self developed code and therefore accelerate code analysis and bug tracking.

## C.2 Texas Instruments DSP/BIOS

The DSP application in the VSDAQ system is based on an RTOS called DSP/BIOS developed and supported by Texas Instruments [97]. DSP/BIOS provides many of the features which are typical for an RTOS. The following paragraphs will give an overview of features of DSP/BIOS which are used within the DSP application of VSDAQ.

### C.2.1 Threads

First we consider the thread management. Within the DSP application there are a number of functions performing different tasks. Some work in response to external events, like e.g. availability of new data, and other functions are started when certain processing tasks have been finished. Indeed, the whole DSP application is structured as a collection of threads, each of which carries out a modularized function. The threads can have different priorities, meaning that a high-priority thread can preempt a low-priority thread. In the context of DSP/BIOS the term *thread* denotes any thread of execution. The four major types of threads are <sup>1</sup>:

- Hardware interrupt functions (HWI) – HWI functions are executed in response to external asynchronous events, e.g. from ADC hardware or from PCI bus, in order to perform a task which is subject to a hard deadline. HWI functions have the highest priority within all DSP/BIOS threads. They can be run at frequencies approaching 200 kHz, which corresponds to a response time below 5  $\mu$ s. A HWI function runs always until completion and should not need more than 100  $\mu$ s execution time.
- Software interrupt functions (SWI) – SWI functions are figured similarly to HWI functions, but with the distinction that SWI functions are triggered by specific function calls from the program. This interface allows e.g. that a SWI can be triggered on occurrence of multiple independent events. A SWI function can have one of 14 different priority levels. Herby, a high-priority SWI function or any HWI preempts a lower-priority SWI function. SWI functions always run to completion and cannot block.
- Tasks (TSK) – Tasks differ from SWIs in that they can be suspended during execution until necessary resources are available. Their priority of a task is

---

<sup>1</sup>The three characters in parenthesis will be used to refer to the respective DSP/BIOS module.

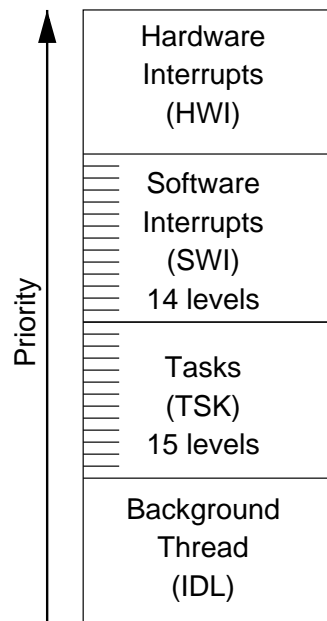


Figure C.1: Thread priorities

always lower than the priority of SWIs or HWIs. Additionally, there are 15 different priority levels available for prioritization within tasks. For communication and synchronization within tasks and with SWIs or HWIs a number of structures including queues, semaphores, and mailboxes can be employed.

- Background thread (IDL) – In the background thread the idle loop is executed, which continuously runs except when it is preempted by another higher-priority thread. Functions that are not subject to any hard deadline can be run within the idle loop. However, since each function is called after the previous has been finished, they should not block, because otherwise they would prevent other idle loop functions from being executed.

Figure C.1 illustrates priority levels of the different thread types.

### C.2.2 Interthread communication and synchronization

As already mentioned above DSP/BIOS has different structures to enable interthread communication and synchronization:

- Semaphores (SEM) – Semaphores are often used to coordinate access to a shared resource among a set of competing tasks. Basically, a semaphore is a counter which comes together with atomic operations for incrementing and decrementing. The value of the counter corresponds to the number of available resources. Thus the counter of the semaphore can only be decremented if it is greater than zero. Before a task accesses a resource it needs to acquire the respective semaphore (decrementing). The task is blocked if the semaphore counter is zero. In this case DSP/BIOS additionally supports different timeout

mechanisms, i.e. the acquisition of the semaphore fails immediately, after a certain period of time, or never, meaning that the task blocks until the semaphore becomes available. When the resource is not longer used, the task must release the semaphore (incrementing).

- Mailboxes (MBX) – A set of functions provides a mailbox functionality where messages are of fixed length and the number of messages is also fixed. Tasks can only post messages if there is a free slot in the mailbox. Otherwise they can wait for a certain timeout period or until a slot becomes available. With such a mailbox mechanism one can prevent an application being flooded by more messages, than it can handle. On the other hand a task can wait for messages to be posted into a mailbox, again for a certain timeout period or until a message is available.
- Queues (QUE) – Queues can be used to exchange data structures of arbitrary length and number between threads. Queue elements can be added and removed by atomic operations. Usually queues are used in a FIFO (First In First Out) sense, i.e. elements are inserted at the tail and removed at the head of the queue. However, insertion and removal of elements at arbitrary positions of the queue is also possible.

### C.2.3 Timer

DSP/BIOS has two different time services – a system clock and a high- and low-resolution timer. By default, the low-resolution time is equal to the system clock. However, any other event, such as availability of data, can also be the time base for the system clock. Moreover, low- and high-resolution timer can be disabled or enabled. Using one of the on-device timers as time base for low- and high-resolutions clocks a time resolution close to single instructions cycles is supported.

### C.2.4 Profiling

DSP/BIOS provides statistic objects (STS) which compute the sum, mean, and maximum for a series of data during execution of the program. The results are displayed within CCS. Based on the timer services one can measure execution times of algorithms, and thus determine if time deadlines are met by a specific part of the code. STS objects can also be employed in order to measure the needed amount of memory on the stack, or to check plausibility of acquired or processed data.

# List of Figures

1.1	Overview of the project VSAMUEL . . . . .	2
2.1	Basic neuron . . . . .	6
2.2	Morphological neural cell categorization . . . . .	8
2.3	Layered structure of cerebral cortex . . . . .	10
2.4	Layered structure of the cerebellum . . . . .	11
2.5	Scheme of an action potential . . . . .	13
2.6	Connection between field and action potentials. . . . .	14
2.7	Tuning curve . . . . .	15
3.1	Interceptable neural activity . . . . .	18
3.2	Equivalent circuit for a biopotential electrode . . . . .	19
3.3	Extracellular recording and gap resistance . . . . .	21
3.4	Microelectrodes with single tip. . . . .	22
3.5	MEA probe and needle array. . . . .	23
3.6	Michigan probe . . . . .	23
3.7	SEM of VSAMUEL probe E4 . . . . .	24
3.8	VSAMUEL Probes . . . . .	25
3.9	Manufacturing process of the VSAMUEL probes . . . . .	26
4.1	General attenuation scheme of a filter . . . . .	28
4.2	Comparison of frequency responses of high pass IIR and FIR filter. . . . .	31
4.3	IIR and FIR high pass filter . . . . .	32
4.4	Single level decomposition and reconstruction filter bank . . . . .	33
4.5	Wavelet filter bank of depth three . . . . .	34
4.6	High pass filter using a wavelet filter bank . . . . .	34
4.7	Variance of WFB filter in frequency domain due to translation . . . . .	35
4.8	Efficient implementation of an FIR filter followed by a subsampler . . . . .	37
4.9	Efficient implementation of upsampling with subsequent FIR filter . . . . .	37
4.10	One stage wavelet filter bank in term of polyphase matrices. . . . .	38
4.11	Wavelet coefficients in lifting order . . . . .	48
4.12	Lifting prediction step in linear assembler . . . . .	49
5.1	High pass filter implemented with a wavelet filter bank . . . . .	52
5.2	Denoising of a neural recording . . . . .	54
5.3	Histograms of extracted noise of a neural recording . . . . .	54
5.4	Comparison of denoising results using different wavelets . . . . .	55

5.5	NEO operator applied to neural recording . . . . .	56
5.6	NEO operator applied to neural recording . . . . .	56
5.7	Spike snippets, wavelet based features, and scalograms of the same single spike snippet at different translations . . . . .	60
5.8	Translation variance of the energy feature . . . . .	61
6.1	Data acquisition system hardware setup . . . . .	65
6.2	Design of the VSAMUEL probe E3 . . . . .	66
6.3	Pre and main amplifier for 64 channels . . . . .	68
6.4	Basic concept of sigma-delta converter, [1] . . . . .	70
6.5	AD16 board . . . . .	71
6.6	M67 board equipped with AD16 and its installed heat spreader . . . .	71
6.7	DMA controller configuration registers . . . . .	73
6.8	Datapath from AD16 to PCI bus . . . . .	74
6.9	Host PC of data acquisition system with 128 channels . . . . .	75
6.10	Adapter board for analog input SCSI to Latch. . . . .	76
6.11	Additional fans and airholes in the cabinet. . . . .	76
6.12	Small cabinet with improved air flow . . . . .	77
6.13	Packet management within DSP application . . . . .	78
6.14	Communication channels between host and target . . . . .	80
6.15	Data Transfer Protocol . . . . .	81
6.16	Main Panel . . . . .	82
6.17	Configure a new experiment . . . . .	84
6.18	Probe and site group settings . . . . .	86
6.19	Probe setup dialog . . . . .	87
6.20	Amplifier setup dialog . . . . .	87
6.21	Blueplot . . . . .	88
6.22	Spectrogram in time frequency mode . . . . .	89
6.23	Spectrogram in waterfall mode . . . . .	89
6.24	Scope with open setup context menu . . . . .	90
6.25	Menu of tool button . . . . .	91
6.26	Add Analyzer dialog . . . . .	92
6.27	Wavelet filter setup dialog . . . . .	93
6.28	Wavelet denoising setup . . . . .	93
6.29	Wavelet denoising and Wavelet filtering . . . . .	94
6.30	Class diagram for VSFileReader and VSFileWriter . . . . .	95
6.31	Classes for data acquisition from multiple DSP boards. . . . .	96
6.32	Interaction of classes during data acquisition. . . . .	97
6.33	Merging of raw data from three DSP applications. . . . .	99
6.34	Reorder of acquired data after copy from shared memory . . . . .	100
6.35	Class diagram for VSRingBuffer and VSDataBuffer . . . . .	101
6.36	Classes for data distribution . . . . .	102
7.1	Field potential profiles obtained in the medial entorhinal cortex . . .	104
7.2	Simultaneous recording from 4 sites after induction of fast <i>gamma</i> oscillations . . . . .	105

7.3	Single unit recordings performed with a M2 probe . . . . .	106
7.4	Field potential profile . . . . .	107
7.5	Receptive fields in Crus II and PML . . . . .	108
7.6	Multi unit recording in cerebellum . . . . .	108
7.7	Extracellular action potential recorded in rat cerebellum . . . . .	109
7.8	Spike interval histogram of evoked activity in rat cerebellum . . . . .	109
A.1	Ebeam evaporation process . . . . .	116
A.2	Lift-off technique . . . . .	117
B.1	SyncLink/ClockLink cable . . . . .	119
B.2	External trigger . . . . .	121
C.1	Thread priorities . . . . .	124

# Bibliography

- [1] Analog Devices. *Sigma-Delta ADCs and DACs*. search for AN-283 on <http://www.analog.com>.
- [2] Gerardo Biella and Marco de Curtis. Olfactory inputs activate the medial entorhinal cortex via the hippocampus. *Journal of Neurophysiology*, 83:1924–1931, 2000.
- [3] Gerardo Biella, Laura Uva, Ulrich G. Hofmann, and Marco de Curtis. Associative potentials in the entorhinal cortex of the guinea pig. *Journal of Neurophysiology*, 88:1159–65, 2002.
- [4] Gerardo Biella, Laura Uva, Ulrich G. Hofmann, and Marco de Curtis. Intrinsic and associative synaptic circuit in the temporal neocortex, perirhinal and entorhinal cortex: an intra- and extracellular electrophysiological study. *Forum of European Neurosciences - Abstracts*, 2002.
- [5] B. Black. Analog-to-digital converter architectures and choices for system design. *Analog Dialogue*, 33(8), September 1999.
- [6] J.O. Bockris and S.U.M. Khan. *Surface Electrochemistry: A Molecular Level Approach*. Plenum Press, New York, June 1993.
- [7] Andrew P. Bradley. Shift-invariance in the discrete wavelet transform. In *Proc. VIIth Digital Image Computing: Techniques and Applications*, pages 29–38, Sydney, December 2003.
- [8] Joseph D. Bronzino, editor. *The Biomedical Engineering Handbook*, volume 1. CRC Press and IEEE Press, 2 edition, 2000.
- [9] B.H. Brown, R.H. Smallwood, D.C. Barber, P.V. Lawford, and D.R. Hose. *Medical Physics and Biomedical Engineering*. Series in Medical Physics. Institute of Physics Publishing, Bristol and Philadelphia, 1999.
- [10] P.K. Campbell, R. A. Normann, K.W. Horch, and S. S. Stensaas. A chronic intracortical electrode array: Preliminary results. *J. Biomed. Mats. Res.: Applied Biomats*, 23:245 – 259, 1989.
- [11] S. Grace Chang, Bin Yu, and Martin Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on Image Processing*, 9(9), September 2000.



- [12] Enric Claverol-Tinture and Jerry Pine. Extracellular potentials in low-density dissociated neuronal cultures. *Journal of Neuroscience Methods*, 117(1):13–21, 2002.
- [13] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41:909–996, 1988.
- [14] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3):245–267, 1998.
- [15] Peter Dayan and L.F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, 1 edition, December 2001.
- [16] Marco de Curtis, Gerardo Biella, C. Buccellati, and G. Folco. Simultaneous investigation of the neuronal and vascular compartments in the guinea pig brain isolated in vitro. *Brain Research Protocols*, 3(2):221–228, 1998.
- [17] Marco de Curtis and Ulrich G. Hofmann. Development of a versatile system for advanced neuronal recordings with multi-site microelectrodes. Technical report, Istituto Nazionale Neurologico "Carlo Besta", Milano, Italy, February 2003.
- [18] Marco de Curtis, D Paré, and R Llinás. The electrophysiology of the olfactory hippocampal system in the isolated and perfused adult mammalian brain in vitro. *Hippocampus*, 1(4):341–354, 1991.
- [19] Clayton T. Dickson, Gerardo Biella, and Marco de Curtis. Evidence for spatial modules mediated by temporal synchronization of carbachol-induced gamma rhythm in medial entorhinal cortex. *Journal of Neuroscience*, 20:7846–7854, 2000.
- [20] Clayton T. Dickson, Gerardo Biella, and Marco de Curtis. Enhancement of spatial and temporal synchronization in the enthorinal cortex by phase reset. *Hippocampus*, 12(4):447–456, 2002.
- [21] David L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.
- [22] David L. Donoho and Iain M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.
- [23] David L. Donoho and Iain M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995.
- [24] Alpha Omega Engineering. Alpha map data acquisition system. Brochure on website, 2008.

- [25] P. Fagerstedt and E. DeSchutter. Multi-layer recordings from the cerebellar cortex of the anesthetized rat using multi-site silicon probes. *Forum of European Neurosciences - Abstracts*, page 15, 2004. A129.4.
- [26] P. Fagerstedt, A. Volny-Luraghi, M. Kindlundh, P. Norlin, U. G. Hofmann, and E. De Schutter. Novel silicon probes with multi-site microelectrodes for single unit and local field potential (lfp) recordings from cerebellar cortex of anesthetized rats. *Forum of European Neurosciences - Abstracts*, 2002. A126.8.
- [27] Patriq Fagerstedt, Antonia Volny-Luraghi, and Erik De Schutter. Results of in vivo acute recordings from cerebellum. Technical report, Laboratory of Theoretical Neurobiology, University of Antwerp, August 2003.
- [28] A. Folkers and U.G. Hofmann. Data acquisition system for 32 channels. VSAMUEL Deliverable Report 4.1, ISIP, University of Lübeck, dec 2002.
- [29] A. Folkers and U.G. Hofmann. Data acquisition system for 64 channels including compression schemes. VSAMUEL Deliverable Report 4.2, ISIP, University of Lübeck, May 2002.
- [30] A Folkers, S Jannasch, and U.G. Hofmann. Lifting scheme - an alternative to wavelet transforms for real time applications. *GlobalDSP Magazine*, 2(12):3–6, dec 2003.
- [31] A. Folkers, H Stemman, K.M.L. Menne, P. Norlin, M. Kindlundh, W. Freiwald, and U.G. Hofmann. Realtime wavelet-based clustering for a 64 channel multisite recording system. In *CNS'03*, page abstract and Poster, 2003.
- [32] Andre Folkers and Ulrich G. Hofmann. A multichannel data acquisition and analysis system based on off-the-shelf dsp boards. In Kálmán Fazekas, editor, *Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services*, pages 249–252, Budapest, September 2001. EURASIP, Scientific Association of Infocommunications - HTE. ISBN 963-8111-64-X.
- [33] Andre Folkers, Florian Mösch, Thomas Malina, and Ulrich G. Hofmann. Real-time bioelectrical data acquisition and processing from 128 channels utilizing the wavelet-transformation. *Neurocomputing*, 52–54:247–254, 2003.
- [34] W. Freiwald, H. Stemmann, A. Wannig, A. K. Kreiter, U. G. Hofmann, M. D. Hills, G. T. A. Kovacs, D. T. Kewley, J. M. Bower, C. W. Eurich, and S. D. Wilke. Stimulus representation in rat primary visual cortex: Multi-electrode recordings and estimation theory. *Neurocomputing*, 44–46:407–416, 2002.
- [35] Winrich Freiwald, Heiko Stemmann, Aurel Wanning, Andreas K. Kreiter, Ulrich G. Hofmann, Matthew D. Hills, Gregory T.A. Kovacs, David T. Kewley, James M. Bower, Axel Etzhold, Stefan D. Wilke, and Christian W. Eurich. Stimulus representation in rat primary visual cortex: multi-electrode recordings with micromachined silicon probes and estimation theory. *Neurocomputing*, 44–46:407–416, 2002.

- [36] William F. Ganong. *Review of Medical Physiology*. Appleton & Lange, 17 edition, 1995.
- [37] Claude Ghez. *Principles of Neural Science*, chapter The Cerebellum, pages 626–646. Appleton & Lange, 1991.
- [38] Ghassan Gholmieh, Walid Soussou, Martin Hana, Ashish Ahuja, Min-Chi Hsiao, Dong Songa, Armand R. Tanguay Jr., and Theodore W. Berger. Custom-designed high-density conformal planar multielectrode arrays for brain slice electrophysiology. *Journal of Neuroscience Methods*, 152:116–129, 2006.
- [39] Multi Channel Systems MCS GmbH. *ME System User Manual*, 2006.
- [40] Charles M. Gray, Pedro E. Maldonado, Mathew Wilson, and Bruce McNaughton. Tetropdes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cate striate cortex. *Journal of Neuroscience Methods*, 63:43–54, 1995.
- [41] J.A. Hartigan and M.A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.
- [42] G.H. Henry, B. Dreher, and P.O. Bishop. Orientation specificity of cells in cat striate cortex. *Journal of Neurophysiology*, 37:1394–1409, 1974.
- [43] U.G. Hofmann, A. Folkers, F. F. Mösch, T. Malina, K.M.L. Menne, G. Biella, P. Fagerstedt, E. De Schutter, W. Jensen, K. Yoshida, D. Hoehl, U. Thomas, M.G. Kindlundh, P. Norlin, and M. De Curtis. A novel high channel-count system for acute multisite neuronal recordings. *IEEE Transactions on Biomedical Engineering*, 53(8):1672 – 1677, 2006.
- [44] U.G. Hofmann, A Folkers, F Mösch, D Höhl, M Kindlundh, and P Norlin. A 64(128)-channel multisite neuronal recording system. *Biomedizinische Technik*, 47(E 1):194–197, 2002.
- [45] U.G. Hofmann, Dirk Höhl, Andreas Hartlep, Robert Schmidt, Peter Detemple, Marion Bär, and Andre Folkers. Geschichtete rundumableitung neuronaler aktivitäten mittels koaxial gelagerter linearer oder helikaler mikroelektrodenanordnungen, 2002.
- [46] U.G. Hofmann, T. Malina, F. Mösch, K. Labusch, and A. Folkers. A versatile daq system for multisite recordings. In *Society for Neuroscience Abstr.*, volume 27, page Program No. 821.72, San Diego, 2001.
- [47] Ulrich G. Hofmann, Erik De Schutter, Ken Yoshida, Marco De Curtis, Uwe Thomas, and Peter Norlin. On the design of multi-site microelectrodes for neural recordings. In P. M. Knoll, editor, *Proceedings of the MICRO.tec 2000*, volume 1, pages 283 – 288, Berlin, 2000. VDE Verlag.

- [48] G.R. Holt and C. Koch. Electrical interactions via the extracellular potential near cell bodies. *Journal of Computational Neuroscience*, 6(2):169–184, 1999.
- [49] S.D. Van Hooser, U.G. Hofmann, D.T. Kewley, and J.M. Bower. Relationship between field potentials and spike activity in rat s1: multi-site cortical recordings and simulations. *Neurocomputing*, 32-33:591–596, 2000.
- [50] M. Horikawa, H. Harada, and M. Yarita. Detection limit in low-amplitude eeg measurement. *Journal of Clinical Neurophysiology*, 20(1):45–53, February 2003.
- [51] D.H. Hubel and T.N. Wiesel. Receptive fields and functional architecture of the monkey striate cortex. *Journal of Physiology*, 195:215–243, 1968.
- [52] P. J. Huber. *Robust Statistics*. John Wiley and Sons, New York, 1981.
- [53] Eyal Hultata, Ronen Segev, and Eshel Ben-Jacob. A method for spike sorting and detection based on wavelet packets and shannon’s mutual information. *Journal of Neuroscience Methods*, 117:1–12, 2002.
- [54] IEEE. *Programs for Digital Signal Processing*. IEEE Press, New York, 1979.
- [55] R.B. Illing. Geschichte der Hirnforschung. *Lexikon der Neurowissenschaft*, 4:40–50, 2001.
- [56] Innovative Integration, Thousand Oaks, California, USA. *Multi-Card Timing Synchronization*, 2001. `m6x_syncblink.pdf` from <http://www.innovative-dsp.com>.
- [57] S. Jannasch, R. Klinger, N. Matter, and U. G. Hofmann. Recording electromyographic signals in a classical conditioning experiment. In *Short Article and Poster, Jahrestagung der DGBMT*, Ilmenau, 2004. Schiel & Schön.
- [58] Herbert H. Jasper. The ten-twenty electrode system of the international federation. *Electroencephalography and clinical neurophysiology*, 10:371–375, 1958.
- [59] W. Jensen, K. Yoshida, and U.G. Hofmann. In-vivo implant mechanics of flexible, silicon-based acro microelectrode arrays in rat cerebral cortex. *IEEE Transactions on Biomedical Engineering*, 53(5):934–940, May 2006.
- [60] W. Jensen, K. Yoshida, and U.G. Hofmann. In vivo implant mechanics of single-shaft microelectrodes in peripheral nervous tissue. In *Neural Engineering, CNE '07. 3rd International IEEE/EMBS Conference on*, pages 1–4, 2007.
- [61] Eric R. Kandel. *Principles of Neural Science*, chapter Nerve Cells and Behaviour, pages 18–32. Appleton & Lange, 1991.
- [62] Eric R. Kandel, James H. Schwartz, and Thomas M. Jessell, editors. *Principles of Neural Science*. Appleton & Lange, 25 Van Zant Street, East Norwalk, Connecticut 06855, 3 edition, 1991.

- [63] Nick G. Kingsbury. The dual-tree complex wavelet transform: A new technique for shift invariance and directional filters. In *Proc. 8th IEEE DSP Workshop*, Utah, August 1998. paper no. 86.
- [64] John Koester. *Principles of Neural Science*, chapter Membrane Potential, pages 81–94. Appleton & Lange, 1991.
- [65] G. Kreiman, C. Hung, A. Kraskov, R. Quiroga, T. Poggio, and J. DiCarlo. Object selectivity of local field potentials and spikes in the macaque inferior temporal cortex. *Neuron*, 49:433–445, 2006.
- [66] Abraham Kuruvilla and Roland Flink. Intraoperative electrocorticography in epilepsy surgery: useful or not? *Seizure*, 12(8):577–584, December 2003.
- [67] Juan Carlos Letelier and Pamela P. Weber. Spike sorting based on discrete wavelet transform coefficients. *JNM*, 101(2):93–106, September 2000.
- [68] Michael S. Lewicki. Bayesian modeling and classification of neural signals. *Neural Computation*, 6(5):1005–1030, 1994.
- [69] Michael S. Lewicki. A review of methods for spike sorting: the detection and classification of neural action potentials. *Network*, 9(4):53–78, November 1998.
- [70] Nikos Logothetis, Jon Pauls, Mark Augath, Torsten Trinath, and Axel Oeltermann. Neurophysiological investigation of the basis of the fmri signal. *Nature*, 412:150–157, 2006.
- [71] T. Malina, A. Folkers, and U.G. Hofmann. Real-time eeg processing based on wavelet transformation. In *IFMBE Proceedings of the 12th Nordic Baltic Conference on Biomedical Engineering and Medical Physics*, pages 166–167, 2002.
- [72] T. Malina, A. Folkers, F. Mösch, K. Labusch, M. Schürmann, and U.G. Hofmann. An integrated system to trigger feedback-coupled event related brain potentials (erps). *Biomedizinische Technik*, 46 E1:322–323, 2001.
- [73] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 2 edition, 1999.
- [74] McNaughton, O. Keefe, and Barnes. The stereotrode: A new technique for simultaneous isolation of several single units in the central nervous system from multiple unit records. *Journal of Neuroscience Methods*, 8:391–397, 1983.
- [75] K.M.L. Menne, A. Folkers, R. Maex, T. Malina, and U.G. Hofmann. Test of spike sorting algorithms on the basis of simulated data. *Neurocomputing*, 44–46:1119–1126, 2002. VSAMUEL.
- [76] K.M.L. Menne, T Malina, A Folkers, and U.G. Hofmann. Biologically realistic simulation of a part of hippocampal ca3: Generation of testdata for the evaluation of spike detection algorithms. In T. Martinetz D. Polani, J. Kim, editor,

*5th German Workshop on Artificial Life - Abstracting and Synthesizing the Principles of Living Systems*, pages 17–25, Lübeck, 2002. IOS Press Infinix.

- [77] K.M.L. Menne, H. Schuhart, A. Folkers, and U. G. Hofmann. Neurosurgery by multichannel electrophysiology, patient specific navigation, and customized database. In *Dreiländertagung der Gesellschaft für Biomedizinische Technik 2003*, volume 48, pages 500–501, Salzburg, Austria, 2003. Schiele & Schön.
- [78] Evangelia Micheli-Tzanakou. *The Biomedical Engineering Handbook*, volume 1, chapter Nervous System. CRC Press and IEEE Press, 2 edition, 2000.
- [79] M. Mühlethaler, Marco de Curtis, K. Walton, and R. Llinás. The isolated and perfused brain of the guinea pig in vitro. *European J. Neuroscience*, 5(7):915–926, 1993.
- [80] S. Mukhopadhyay and G.C. Ray. A new interpretation of nonlinear energy operator and its efficacy in spike detection. *IEEE Transactions on Biomedical Engineering*, 45(2):180–187, 1998.
- [81] Miguel A.L. Nicolelis, editor. *Methods for neural ensemble recordings*. CRC Press, 1999.
- [82] Peter Norlin and Maria Kindlundh. Custom designed 32 microelectrode probes (VSAMUEL Deliverable 2.1). Technical report, ACREO AB, Kista, Sweden, May 2001.
- [83] Peter Norlin, Maria Kindlundh, Aliette Mouroux, Ken Yoshida, and Ulrich G. Hofmann. A 32-site neural recording probe fabricated by drier of soi substrates. *Journal of Micromechanics and Microengineering*, 12:414–419, July 2002.
- [84] Paul L. Nuñez. *Electric Field of the Brain: The Neurophysics of EEG*. Oxford University Press, New York, 1981.
- [85] Paul L. Nuñez and Ramesh Srinivasan. Electroencephalogram. *Scholarpedia*, 2(2):1348, 2007.
- [86] A.V. Oppenheim and R.W. Schaffer. *Discrete-time signal processing*. signal processing series. Prentice Hall, 1989.
- [87] L.R. Rabiner, J.H. McClellan, and T.W. Parks. Fir digital filter design techniques using weighted chebyshev approximations. *Proc. IEEE*, 63, 1975.
- [88] M. Salganicoff, M. Sarna, L. Sax, and G.L. Gerstein. Unsupervised waveform classification for multi-neural recordings: a real-time, software based system. i. algorithms and implementation. *Journal of Neuroscience Methods*, 25:181–187, 1988.
- [89] Edward M. Schmidt. *Methods for neural ensemble recordings*, chapter Electrodes for many single neuron recordings. CRC Press, 1999.

- [90] Erik De Schutter. Models. <http://www.tnm.ua.ac.be/models/index.shtml>, 1998.
- [91] Andrew B. Schwartz, X. Tracy Cui, Douglas J. Weber, and Daniel W. Moran. Brain-controlled interfaces: Movement restoration with neural prosthetics. *Neuron*, 52:205–220, October 2006.
- [92] Ivan W. Selesnick, Richard G. Baraniuk, and Nick G. Kingsbury. The dual-tree complex wavelet transform. *IEEE Signal Processing Magazine*, 22(6):123–151, November 2005.
- [93] Ivan W. Selesnick and C.S. Burrus. Generalized digital butterworth filter design. *Proceedings of the IEEE Int. Conf. Acoust., Speech, Signal Processing*, 3, May 1996.
- [94] W. Singer and C.M. Gray. Visual feature integration and the temporal correlation hypothesis. *Ann Rev Neurosci*, 18:555–586, 1995.
- [95] W. Sweldens. The lifting scheme: A new philosophy in biorthogonal wavelet constructions. In A. F. Laine and M. Unser, editors, *Wavelet Applications in Signal and Image Processing III*, pages 68–79. Proc. SPIE 2569, 1995.
- [96] Plexon Neurotechnology Research Systems. *Neural Data Acquisition System – Recorder/16 and Recorder/64*, 2006.
- [97] Texas Instruments Incorporated, Post Office Box 655303, Dallas, Texas 75265. *TMS320 DSP/BIOS User’s Guide*, November 2001. <http://www-s.ti.com/sc/psheets/spru423a/spru423a.pdf>.
- [98] H.D. Tong, R.A.F. Zwijze, J.W. Berenschot, R.J. Wiegerink, G.J.M. Krijnen, and M.C. Elwenspoek. Characterization of platinum lift-off technique. In *Proceedings of the Semiconductor Sensor Workshop*, pages 697–702, Veldhoven, the Netherlands, December 2000. The Dutch Technology Foundation.
- [99] M. Trepel. *Neuroanatomie: Struktur und Funktion*. Urban und Schwarzenberg, München; Wien; Baltimore, 1995.
- [100] Solange van der Linden, Ferruccio Panzica, and Marco de Curtis. Carbachol induces fast oscillations in the medial but not in the lateral entorhinal cortex of the isolated guinea pig brain. *Journal of Neurophysiology*, 82:2441–2450, 1999.
- [101] A. Volny-Luraghi, R. Maex, B. Vos, and E. De Schutter. Peripheral stimuli excite coronal beams of golgi cells in rat cerebellar cortex. *Neuroscience*, 113(2):363–373, 2002.
- [102] B. Weber, T. Malina, K. Menne, A. Folkers, V. Metzler, and Ulrich G. Hofmann. Handling large files of multisite microelectrode recordings for the European VSAMUEL consortium. *Neurocomputing*, June 2001.

- [103] Wikipedia. Butterworth. *Wikipedia*, 2007. [http://en.wikipedia.org/wiki/Butterworth\\_filter](http://en.wikipedia.org/wiki/Butterworth_filter).
- [104] Menno Witter and Floris Wouterlood, editors. *The Parahippocampal Region: Organization and Role in Cognitive Functions*. Oxford University Press, New York, 2002.
- [105] S. Ramón y Cajal. *Histology*. Wood, 10 edition, 1933.
- [106] Ken Yoshida, W. Jensen, Peter Norlin, M. Kindlundh, and Ulrich G. Hofmann. Characterization of silicon microelectrodes from the EU VSAMUEL project. In J. Werner, editor, *Biomedizinische Technik*, Bochum, 2001. Schiele & Schön GmbH.



# LEBENS LAUF

## ANDRE FOLKERS

### Kontakt

Pausheide 9  
33415 Verl  
Andre.Folkers@web.de



## PERSÖNLICHE DATEN

### geboren am Familienstand

14. Januar 1974 in Westerstede  
verheiratet seit 26.11.1999  
mit Dr. med. Verena Folkers, geb. Springfield

### Kinder

Antonia, Karolin, Frederik und Teresa

## BERUFSTÄTIGKEIT

18. Mai 2009	Doktorprüfung zum Dr.-Ing. an der Universität zu Lübeck
seit 01/2007	Softwareentwickler bei Beckhoff Automation GmbH, Verl
10/2003 – 12/2006	Softwareentwickler bei Harman/Becker Automotive Systems, Hamburg
02/2003 – 09/2003	Software-Ingenieur bei Thomas Recording GmbH, Gießen, in Kooperation mit der Universität Lübeck
02/2000 – 01/2003	Wissenschaftlicher Mitarbeiter am Institut für Signalverarbeitung, Universität zu Lübeck

## STUDIUM UND SCHULE

10/1994 – 01/2000	Informatik mit Nebenfach Medizinische Informatik, Universität zu Lübeck
14. Januar 2000	Diplom in Informatik
04/1999 – 09/1999	Diplomarbeit „Pictorial Query Specification and Processing“ Universität von Maryland in College Park, USA
7. Oktober 1996	Vordiplom
02/1996 – 12/1998	Studentischer Mitarbeiter am Institut für Technische Informatik
04/1997 – 07/1997	Studentischer Übungsleiter am Institut für Informationssysteme
10/1996 – 02/1997	Studentischer Mitarbeiter am Institut für Mathematik
10/1993 – 09/1994	Sanitätssoldat, Sanitätsbataillon 11, Leer und Varel
3. Mai 1993	Abitur am Gymnasium Westerstede