UNIVERSITY OF OSLO
Department of Informatics

Neuromorphic Electronics: Lecture Notes

Compendium for INF5470

Philipp Häfliger, Juan A. Leñero Bardallo

Fall 2012

Abstract

This book is an introductory course in neuromorphic electronic circuits. These are circuits inspired by the nervous system that either help verifying neuro-physiological models, or that are useful components in artificial perception/action systems. Research also aims at using them in implants. These circuits are computational devices and intelligent sensors that are very differently organized than digital processors. Their storage and processing capacity is distributed. They are asynchronous and use no clock signal. They are often purely analog and operate time continuous. They are adaptive or can even learn on a basic level instead of being programmed. A short introduction into the area of brain research is also included in the course.

The students will learn to exploit mechanisms employed by the nervous system for compact energy efficient analog integrated circuits. They will get insight into a multidisciplinary research area. The students will learn to analyze analog CMOS circuits and acquire basic knowledge in brain research methods.

The document is suitable for everyone that wants to have an general overview about neuromorphic ingeneering. However, the script is intended as supportive material for the course and may be unsatisfying to read without attending the lectures. The students are strongly encouraged to add their own notes to these pages.

Contents

Abstract 1

Abstract 1
Contents i
List of Figures v
List of Tables ix

1 Introduction 1
1.1 Neuromorphic Circuits at Present 1

2 Neurophysiology in a Nutshell 3
2.1 Methods 3
2.1.1 Psychophysical Experiments 3
2.1.2 EEG 4
2.1.3 fMRI and PET 4
2.1.4 Extracellular Electrodes 7
2.1.5 Intracellular Electrodes 7
2.1.6 Fluorescent Tracers and Imaging 9
2.1.7 Briefly Mentioned: Methods in Neuroanatomy 9
2.2 Knowledge 10
2.2.1 Brain Anatomy 10
2.2.2 Cortical Regions 10
2.2.3 Organization within Cortical Regions 13
2.2.4 Microcolumns and Cortical Layers 14
2.2.5 Neurons and Synapses 15

3 Basic Analog CMOS 19
3.1 Field Effect Transistors 19
3.1.1 Basic Formulae 19
3.1.2 Early effect 22
3.1.3 Gate leakage 23
3.2 Capacitors 23
3.3 Current Mirror 24
3.4 Differential Pair 25
3.5 Transconductance Amplifier 26
3.6 Follower 28
3.7 Resistor 28
3.8 Resistive Nets 29
3.9 The Winner Take All Circuit 30

4 Real and Silicon Neurons 35
4.1 Real Neurons 35
4.2 aVLSI Models of Neurons 35
4.2.1 Simple Electrical Nodes as Neurons 38
4.2.2 Perceptrons (Mc Culloch Pitts neurons) 38

4.2.3 Integrate and Fire Neurons 42
4.2.4 Compartemental Neuronal Models (Silicon Neurons) 43

5 Coding in the Nervous System 49
5.1 The action potential 49
5.2 Hints in Experiments 50
5.2.1 Classical experiments based on observing spike rates 50
5.2.2 Classical Experiments observing temporal spike codes 52
5.3 Candidate Codes 55

6 Neuromorphic Communication: the AER Protocol 59
6.1 The Basic Idea of Address Event Representation (AER) 59
6.2 Collision Handling 60
6.2.1 Full Arbitration 61
6.2.2 Collision Discarding 64
6.2.3 Aging versus Loss Trade Off 65

7 Photo Receptors in CMOS Technology 67
7.1 Introduction 67
7.2 Fundamentals of Photo Detectors Operation 67
7.3 Overlaying Material 68
7.4 Light Absorption in Silicon 69
7.5 Recombination Lifetime 69
7.6 Diffusion Length 70
7.7 Photo Charge and Photo Current 70
7.8 Dark Current 71
7.9 CMOS Photo Detector Structures 72
7.10 Logarithmic Receptors 74

8 Retinomorphic Circuits 77
8.1 The Retina 77
8.2 CMOS photo sensors 80
8.2.1 Photo diodes 81
8.2.2 Photo transistors 81
8.2.3 Photo gates 82
8.3 Photo Current Amplification 82
8.3.1 Linear by Early effect 82
8.3.2 Logarithmic by gate to source voltage 83
8.3.3 Common source amplification 83
8.3.4 Source follower 84
8.4 Read Out Strategies 85
8.4.1 Addressing and scanning 85
8.4.2 Charge coupled devices (CCD) 87
8.4.3 Address event representation 87
8.5 Silicon retinae 89
8.5.1 Adaptive photo cell 89
8.5.2 Spatial contrast retina 89
8.5.3 Temporal contrast retina 89
8.6 Further Image Processing 93
8.6.1 Motion 94
8.6.2 Feature maps 96

9 Cochleomorphic Circuits **101**
9.1 The Cochlea 101
9.2 Silicon Cochlea 101

10 Neuromorphic Learning **109**
10.1 Neural Learning Algorithms 109
10.1.1 An overview of classes of learning algorithms 110
10.1.2 Supervised Learning 110
10.1.3 Reinforcement learning 111
10.1.4 Unsupervised learning 111
10.2 Analogue Storage 121
10.2.1 Dynamic Analogue Storage 121
10.2.2 Static Analogue Storage 121
10.2.3 Non-Volatile Analogue Storage 126
10.3 Neuromorphic Learning Circuits 130
10.3.1 Hebbian learning circuits 130
10.3.2 A spike based learning circuit 132

A Questions Catalogue **135**
A.1 Introduction 135
A.2 Neurophysiology 135
A.3 Basic Analogue CMOS 135
A.4 Real and Silicon Neurons 136
A.5 Coding in the Nervous System 136
A.6 Neuromorphic Communication: the AER Protocol 136
A.7 Retinomorphic Circuits 137
A.8 Cochleomorphic Circuits 137
A.9 Neuromorphic Learning 137

Bibliography **139**

List of Figures

2.1 EEG array based reconstruction of brain activity 5
2.2 EEG of sleep stages 5
2.3 fMRI in bilingual task 6
2.4 The Utah electrode array 7
2.5 Patch clamp electrodes 8
2.6 Fluorescent tracers in Purkinje cell 9
2.7 Brain Cross section illustration 10
2.8 Motor-cortex homunculus 11
2.9 Brain research by Garry Larson 12
2.10 Cortical regions in a cat 12
2.11 Cortical regions hierarchy 13
2.12 ocular dominance patterns on V1 14
2.13 ocular dominance patterns close-up 14
2.14 Orientation selectivity patterns 15
2.15 Cortical layers staining illustration 1 16
2.16 Cortical layers staining illustration 2 17
2.17 Cortical layers staining techniques 17
2.18 Schematic synapse 18

3.1 FETs 20
3.2 I_{DS} vs. V_DS 20
3.3 I_{DS} vs. V_G 21
3.4 I_{DS} vs. V_G 22
3.5 Capacitances in CMOS 23
3.6 Current mirror 25
3.7 Differential pair 26
3.8 Transconductance amplifier 27
3.9 Follower 28
3.10 Resistors in CMOS 29
3.11 Resistive net 29
3.12 Diffuser net 30
3.13 WTA principle 31
3.14 CMOS WTA 31
3.15 WTA with spatial smoothing 32
3.16 local WTA 33
3.17 WTA with hysteresis 33

4.1 Anatomical Parts of a Neuron 36
4.2 Light Microscope Neuron 36
4.3 3D Reconstruction of Pyramidal Cell 37
4.4 Perceptron Concept 38
4.5 Perceptron Schematics 39
4.6 Gilbert Multiplier 40
4.7 Concept Integrate-and-Fire Neuron 41

4.8 Carver Mead Integrate-and-Fire Neuron 41
4.9 Adaptive Integrate-and-Fire Neuron 42
4.10 Compartemental Neuron Model 44
4.11 The Hudgkin Huxley Model 45
4.12 A CMOS Implementation of a HH-soma 45
4.13 Cable Model 46

5.1 Galvani experiment with twitching frog legs 50
5.2 Orientation selective neuron responses 51
5.3 Exact responses to random dot patterns I 52
5.4 Synfire chains 53
5.5 Phase relation of place cells 54
5.6 Illustration of coding schemes 56
5.7 Latency coding 57

6.1 Address Event Representation 60
6.2 4 Phase Handshake 60
6.3 two-input 'greedy' arbiter 61
6.4 Glitch free two-input 'greedy' arbiter 62
6.5 Binary arbiter tree 62
6.6 AER with collision discarding 64
6.7 Aging versus Loss trade off AER 65

8.1 Eyball cross section 78
8.2 Detailed retinal cells 79
8.3 Schematic retinal cells 80
8.4 Photo diode 81
8.5 Photo diode layout 81
8.6 PNP photo transistor 82
8.7 Photo gate 82
8.8 Amplification by drain resistance 83
8.9 Logarithmic amplification 84
8.10 Two transistor inverting amplifier 85
8.11 Negative feedback 86
8.12 Active pixel 87
8.13 CCD 88
8.14 AER photo pixel 88
8.15 Adaptive photo cell 90
8.16 Non linear element 91
8.17 Mahowald silicon retina 91
8.18 Boahen silicon retina 92
8.19 Temporal contrast retina diagram 92
8.20 Temporal contrast retina transistor level circuit 93
8.21 Reichardt detector 94
8.22 Intensity based motion estimation 95
8.23 Original natural scene 96
8.24 Surface plot of 2D 'difference of Gaussians' 97
8.25 Colour code plot of 2D 'difference of Gaussians' 98
8.26 'Difference of Gaussians' convolved image 98
8.27 Surface plot of a 45 degree edge extraction kernel 99
8.28 Colour code plot of a 45 degree edge extraction kernel 99

8.29 Image after 45% edge extraction 100

9.1 Ear cross section 102
9.2 Cochlea cross section 103
9.3 EM of hair cells 103
9.4 A second order filter stage 104
9.5 Parallel second order filter spectra 105
9.6 Cascaded second order filter spectra 106

10.1 Character recognition by associative memory 113
10.2 Classification with LVQ 114
10.3 Dynamics in Learning Vector Quantization 115
10.4 Dynamics in competitive Hebbian learning 115
10.5 competitive learning vs. associative memory 116
10.6 Spike based learning in a silicon neuron 119
10.7 Spike based Learning simulation variables 119
10.8 Capacitive dynamic analog storage 122
10.9 AD/DA multi-level static storage 123
10.10 A 'fusing' amplifier 124
10.11 Weak multi-level static memory 125
10.12 Floating gate, non-volatile analog storage 126
10.13 Band diagram for tunneling through the gate oxide 127
10.14 High voltage NFET 127
10.15 On chip high voltage switch 128
10.16 Diorio learning array 130
10.17 Fusi bistable learning circuit 131
10.18 Blockdiagram of a spike based learning circuit 132
10.19 Positive term circuit 133
10.20 Negative term circuit 133
10.21 Floating gate analog storage cell 134

List of Tables

4.1 aVLSI Models of Neurons 38

Chapter 1

Introduction

The term 'neuromorphic' was introduced by Carver Mead around 1990. He defined neuromorphic systems as artificial systems that share organization principles with biological nervous system. So what are those organization principles?

A brain is fundamentally differently organized than a computer and science is still a long way from understanding how the whole thing works. A computer is really easy to understand by comparison. Features (or organization principles) that clearly distinguish a brain from a computer are massive parallelism, distributed storage, asynchronous processing, self organization. Neuromorphic circuits try to incorporate those principles. Whereas a computer by contrast is a synchronous serial machine, with centralized storage and it is programmed. Table 1.1 compares these different organizing principles.

Research in neuromorphic aVLSI can lead to computing machines that are fundamentally differently organized than computers. Machines that best computers at tasks that humans are better at than computers. This can be achieved by more closely copying the biological pendant. And although computers are general machines that can simulate any other digital machine and arbitrarily approximate analog processes, neuromorphic machines can outrank them in terms of speed, energy efficacy, and size.

1.1 Neuromorphic Circuits at Present

Our understanding of the nervous system can be considered good in the peripheral parts only: Many sensors and their low level processing and muscles and their enervation were exhaustively explored. So it comes as no surprise that neuromorphic electronics that mimics those parts is the most successful. Especially visual and auditory 'intelligent' sensors came out of that line of research.

Computer	Brain
Serial	Parallel
One powerful central CPU, memory	10^{11} simple distributed computational and memory units
Busses shared by several components	Dedicated local point to point connections
Not very power efficient (needs cooling)	Very power efficient (hair to keep it warm ;-))
Digital, time-discrete	Analog, continuous time
Programmed	Learning
Sensitive to errors	Robust to errors (using redundancy)

Table 1.1: Organizing principles of computers and the nervous system

Chapter 2

Neurophysiology in a Nutshell

Understanding the brain is one of the biggest remaining challenges to science. Scientists are very busy scratching together bits and pieces of knowledge about the nervous system and in fact the data acquired is enormous. But how all the parts work together to solve the fuzzily defined tasks in our daily lives is still a mystery.

This chapter tries to give an impression of today's brain research. To the neuromorphic engineer the most interesting subfield is neurophysiology, where the nervous system is observed in action. So at first an overview on the methods used for these observations is given, followed by some examples of the knowledge that has been gained by these methods.

2.1 Selected Methods in Neurophysiological Research

The different observation methods that are applied in neurophysiology do all have strengths and weaknesses. There is always a trade off between resolution and the total observed area/volume. Table 2.1 gives a short summary of the methods that are discussed some more in the following.

2.1.1 Psychophysical Experiments

Psychophysical experiments constitute the most holistic approach to brain research. Human subjects are tested for reactions to all kinds of stimuli. Combined with some of the following observation methods one can establish corellation of activity patterns with the tasks that a testsubject is asked to perform. But also completely non-invasive observations, for example reaction times and error rates in visual recognition tasks, sometimes allow to draw conclusions about neurophysiological hypotheses.

A very illuminating example is the experiment reported in [1, 2], where the reaction times in quite complex recognition tasks led to a strong conclusion. People and monkeys were asked/trained to push one of two buttons dependent on the presence or absence of an animal in a picture that was flashed at them. The speed of their reaction and the estimated delay in the signal path along the axons from visual input all the way to

method	test subjects	observation area	order of temp. res.	order of spat. res.
Psycho Physical Exp.	alert humans			
EEG	alert humans	patches of brain surface	ms	cm^2
fMRI and PET	alert humans	brain cross-sections	40ms	$5mm^3$
Extra Cellular Electrodes	alert test animals	a neighbourhood of neurons	μs	$100^3 \mu m^3$
Intra Cellular Electrodes	anesthetized test animals, slice preparations	one neuron	μs	$10^3 \mu m^3$
Fluorescent Tracers	anesthetized test animals, slice preparations	a dendritic tree	?	$< \mu$m

Table 2.1: Summary on some methods in neurophysiological research

motor output would allow for only one to two action potentials per neuron before the task was completed. The conclusion was that the brain does not merely use average action potential rate signals as the fundamental carrier of information, since an average cannot at all be computed with only 1 action potential, and not reliably with only 2.

2.1.2 EEG

EEG applies surface electrodes to the human test subject's skin on the skull. Thus, minuscule changes in potential can be measured that are related to correlated neuron activity in the underlying cortical area. A certain spatial resolution can be achieved by placing multiple electrodes on the skull. In the example in figure 2.1 is a reconstruction of activity measured by a net of 256 electrodes.

Output of individual electrodes is very detailed with a good temporal resolution. Example traces are depicted in figure 2.2. These traces are the result of the averaged activity of a huge number of neurons in the observed area though, and only correlated activity can be observed. Anything more subtle going on, e.g. if half of the neurons stop firing and the other half doubles their firing rate, will not be visible.

EEG has been very prominent in the study of sleep/alertness studies, since very distinct correlated cyclic activity of certain frequencies can be linked to alertness, drowsiness and sleep phases.

2.1.3 fMRI and PET

PET and fMRI are methods that allow to observe processes in the brain as a whole. They are also quite unobtrusive and are thus used on human

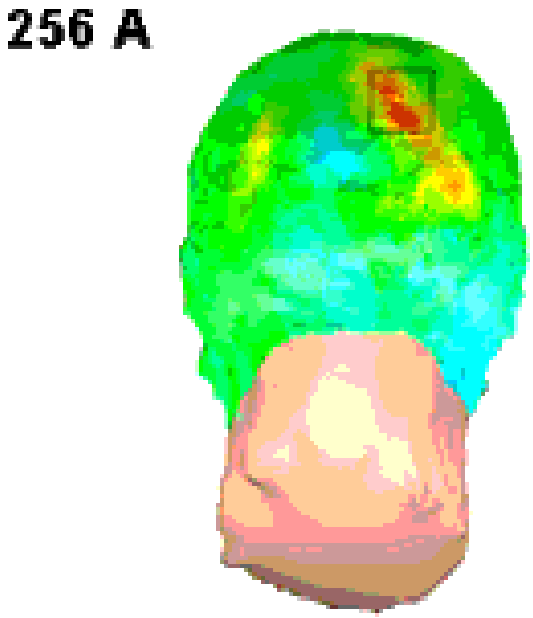


Figure 2.1: Reconstruction of correlated brain activity on the surface of the cortex with data from a 256 surface electrode array (EEG) [3]

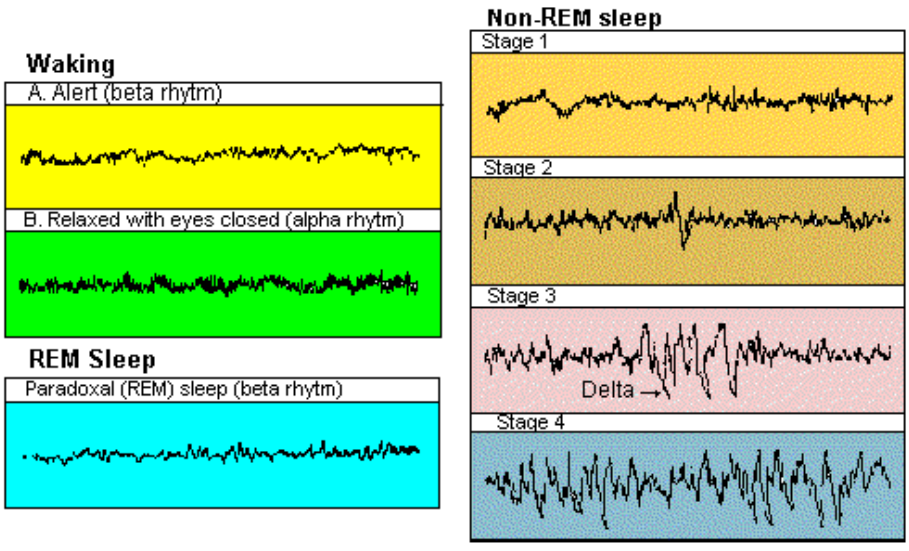


Figure 2.2: Brain EEG activity in different sleep stages [4]

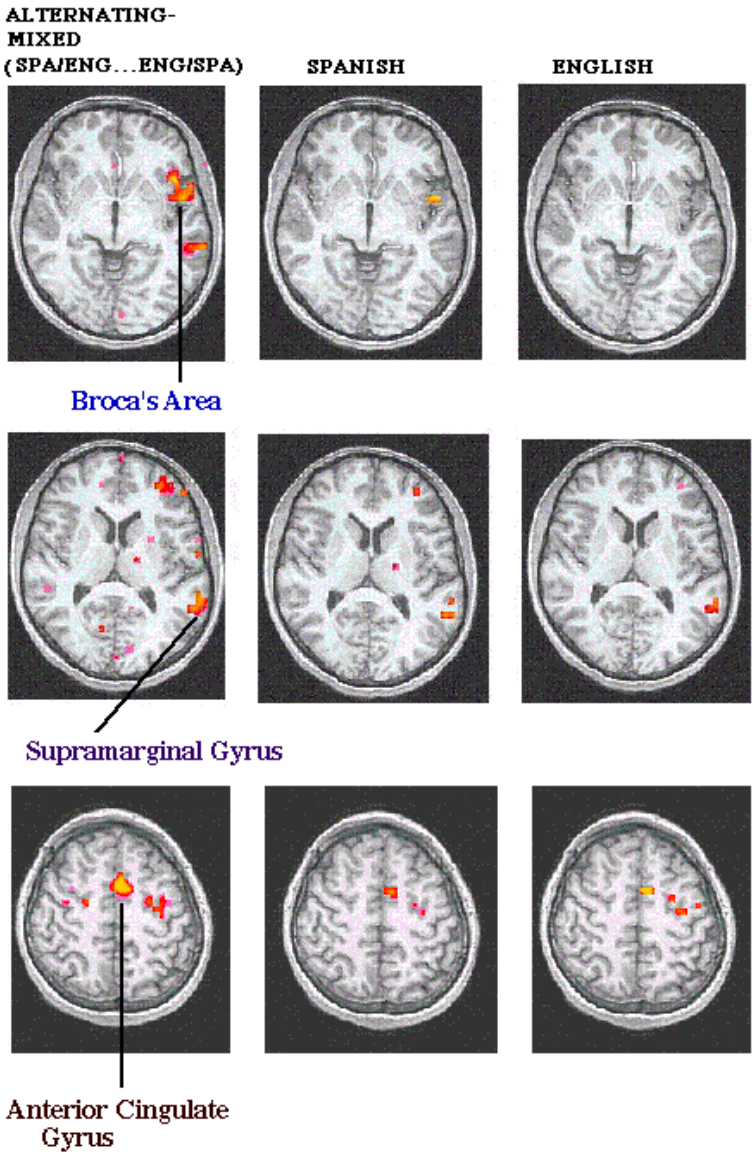


Figure 2.3: fMRI in bilingual task: picture naming [5]

test subjects. Both measure 'brain activity' indirectly, i.e. they measure observables that can be related to 'brain activity', such as oxygen or sugar levels. PET uses radioactive tracers, e.g. labeled oxygen or glucose. fMRI detects for example oxygen level in blood and/or blood-flow. An example of the kind of data gathered by fMRI is shown in figure 2.3.

fMRI and PET have become very important methods these days to map out activity in brain regions in relation to executing certain tasks in psychophysical experiments. the example shown in figure 2.3 is typical in that respect: the study identifies brain areas that are active when bilingual people are rapidly switching between languages. they are asked to name what they see in a series of pictures, either only in English or Spanish or

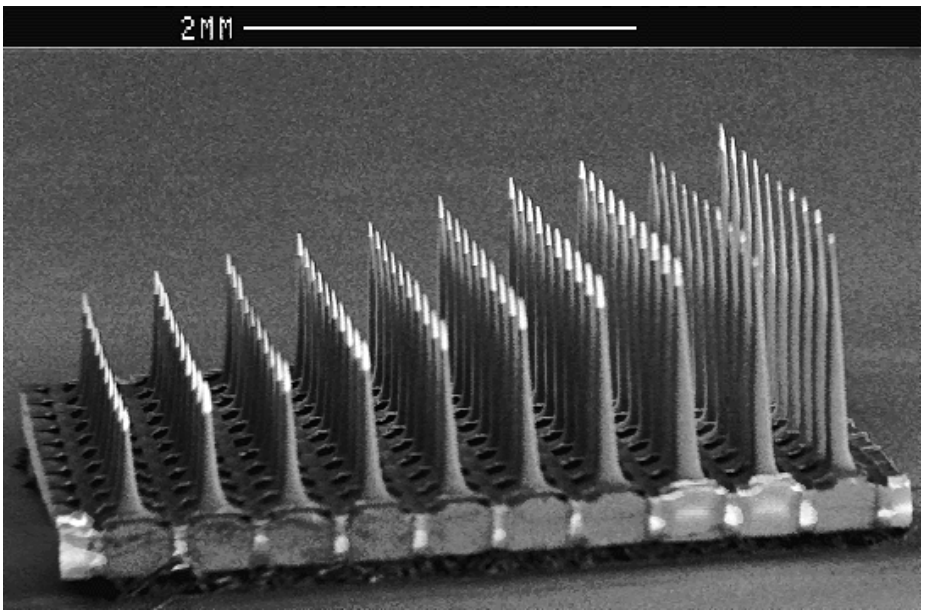


Figure 2.4: The slanted version of the Utah electrode array

alternating between the languages.

2.1.4 Extracellular Electrodes

Much more detailed data, but limited to a much smaller observable volume, can be gained by measuring intrusively in the brain tissue. Electrodes that lie inbetween the neurons pick up changes of potential that are caused by the production of action potentials. With appropriate post-processing of the data, action potentials of single neurons can be isolated. That works especially well with placement of multiple electrodes. Quite famous in that regard has become the so called 'Utah array'[6], a 2D array of electrodes on a single substrate that has in 2010 in one instance even been implanted underneath the skull and into the cortex of a first human subject, a patient that suffers from complete paralysis from the neck downwards. With this electrode array experiments are being conducted that allow him to exert some control on, for example, a cursor on a computer screen through thid direct brain-machine interface (BMI). But in general these kind of measurements require serious surgery and are otherwise so far only performed on lab animals (in vivo) or in brain tissue preparations (in vitro).

2.1.5 Intracellular Electrodes

Even more local details can be recorded by actually penetrating the cell membrane of neurons. This can be achieved by sharp electrodes, even in immobile anesthetized lab animals. An instrument that is less damaging to the cell is the patch clamp electrode, that can be patched upon the cell membrane and that cuts out a hole causing minimal leakage of extracellular liquid into the cell. This electrode, however, requires even more precise placing and, thus, simultaneous observation with, for example, a phase contrast microscope, and has thus only been used in slice preparations so far. A prticullarly challanging task is to place more than

Figure 2.5: Patch clamp electrodes onto two neurons in in vitro preparation under a phase contrast microscope[7]

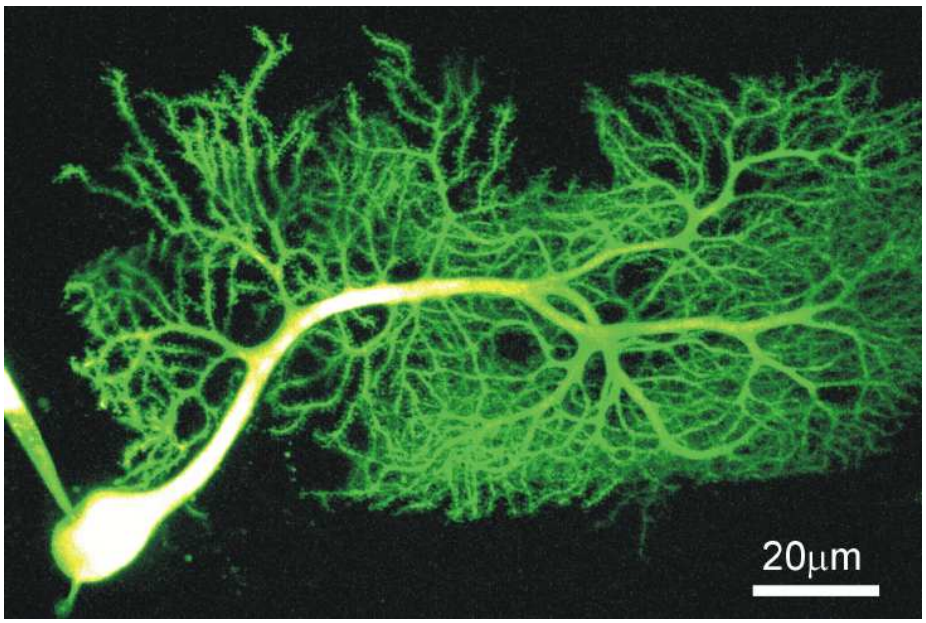


Figure 2.6: Purkinje cell with fluorescent tracer in two photon microscope [8]

one of those electrodes upon the same cell or interconnected neighbouring cells simultaneously, to observe intrcellular local dynamics or correlated dynamics of connected cells.

Figure 2.5 is an example of a phase contrast microscopy picture of two patch clamp electrodes that are placed on two cell bodies of interconnected cells. A trick that requires days or weeks of patience from the experimenter.

2.1.6 Fluorescent Tracers and Imaging

Spatially continuous intracellular dynamics in neurons can be observed with help of fluorescent tracers. Two photon microscopy has received some press in that context, e.g. in observing calcium dynamics inside a dendrite. Research also explores possibilities of multi-photon microscopy. Two- and multi-photon microscopy offers excellent spatial resolution. the temporal resolution is good too but somewhat limited by the fact that only one point at a time can be observed and the observed area is thus scanned rapidly to produce a complete picture. The method can be used in slice preparations and even on anesthetized animals. With the help of fluorescent tracers e.g. observations of calcium dynamics can be made, with excellent spatial resolution, very good temporal resolution, for an area of observation of up to a complete dendritic tree. The figure 2.6 shows a Hyppocampal Purkinje cell that is suffused with calcium that is tagged with a fluorescent tracer [8]

2.1.7 Briefly Mentioned: Methods in Neuroanatomy

We did not discuss methods that reveal purely anatomical information. There would still be a lot to be said about methods in microscopy in combination with various tracers and stainers that colour particular structures. Like antibodies that attach to particular ion channels, or

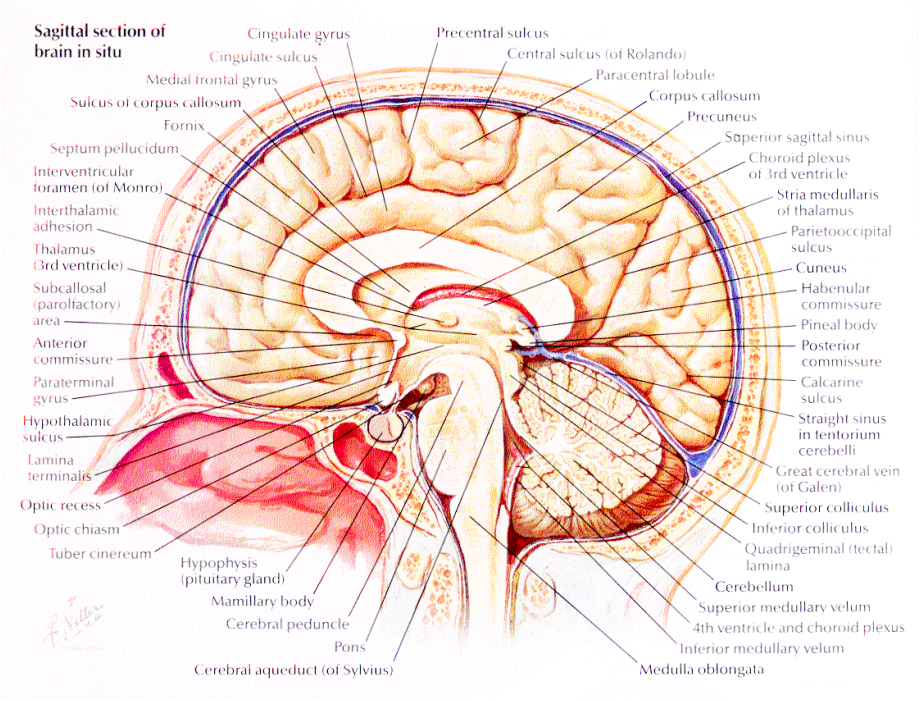


Figure 2.7: A drawing of a crossection of the brain according to [9]

chemicals that spread through a whole neuron cell body, including the complete dendritic and axonal tree.

2.2 Selected Neurophysiological and Neuroanatomical Knowledge

2.2.1 Brain Anatomy

A good impression of the brains structure and its parts with their Latin names is shown in figure 2.7. Of particular interest to neurophysiology in the recent years is the cortex, the most 'modern' part of the brain, believed to host the 'higher' brain functions, such as visual processing in the occipital lobe (on the backside of cortex) and conscious thought in the frontal lobes (towards the front). It is thoroughly explored and divided into functional subregions as shall be further discussed in the following.

2.2.2 Cortical Regions

Activity in many cortical areas is known to be correlated with particular body functions or perceptions. Thus, one can identify for example a visual cortical region, a motor cortex (mapped to the muscles in the body), or somatosensory cortex (mapped to the haptic-, pain-, temperatur- and other sensors on the body). It is also known in many instances that there is a topological mapping between sensors or actuators and subsequent brain areas, i.e. neighbourhood relations in the real world are preserved on the cortical surface that real world data is mapped onto. The motor-cortex, for example, has been mapped early by stimulating regions of patients in brain surgery and observing the motor reactions. A famous illustration of

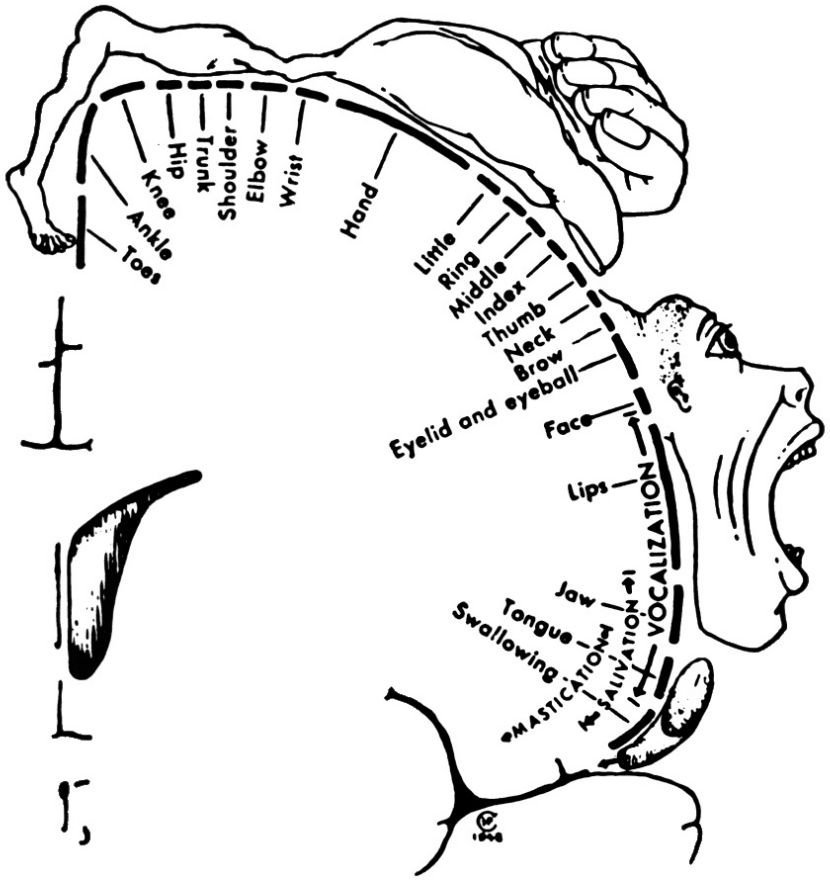


Figure 2.8: The famous Motor-Homunculus representing the body parts on motor cortex [10]

these findings is the so called motor-homunculus (figure 2.8). An equivalent picture exists for somatosensory cortex. An illustration that is not too far from the truth of the method applied to get these results can be found in figure 2.9.

The most explored cortical area is the visual cortex. In primates processing of visual data seems to involve a substantial part of the cortical area. Sub areas have been mapped out by presenting anesthetized lab animals with particular optical stimuli and measuring resulting activity with electrodes. Some of them are involved in perception of motion, of colour, of low level optical features, or of high level objects. Again topological mapping (this time of the visual field) is maintained from one region to the next. In a famous publication D. C. van Essen, C. H. Anderson and D. J. Felleman [11] summarized all known visual cortical regions for macaque monkeys (figure 2.10) and ordered them in a hierarchy (figure 2.11). The hierarchy is based on a huge number of anatomical studies that explored which regions have direct connections to which other regions. This can for example be done by injecting colored tracer chemicals that spread through the axons into one region. Then it will be revealed where those axons go.

Figure 2.9: An illustration that is not far from the truth by Gerry Larson

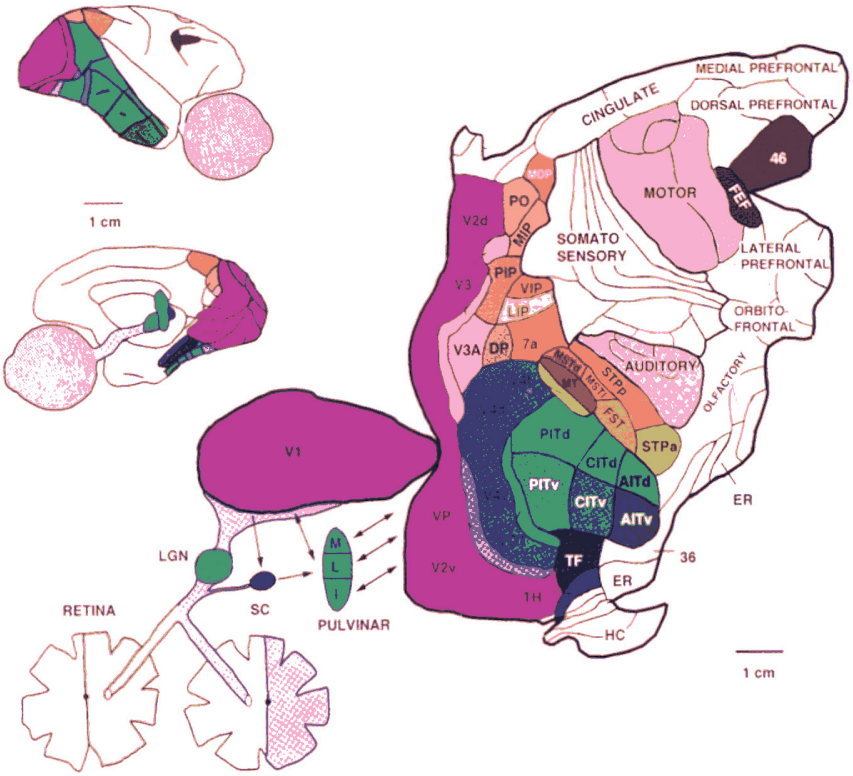


Figure 2.10: The most famous classification of cortical regions according to [11]

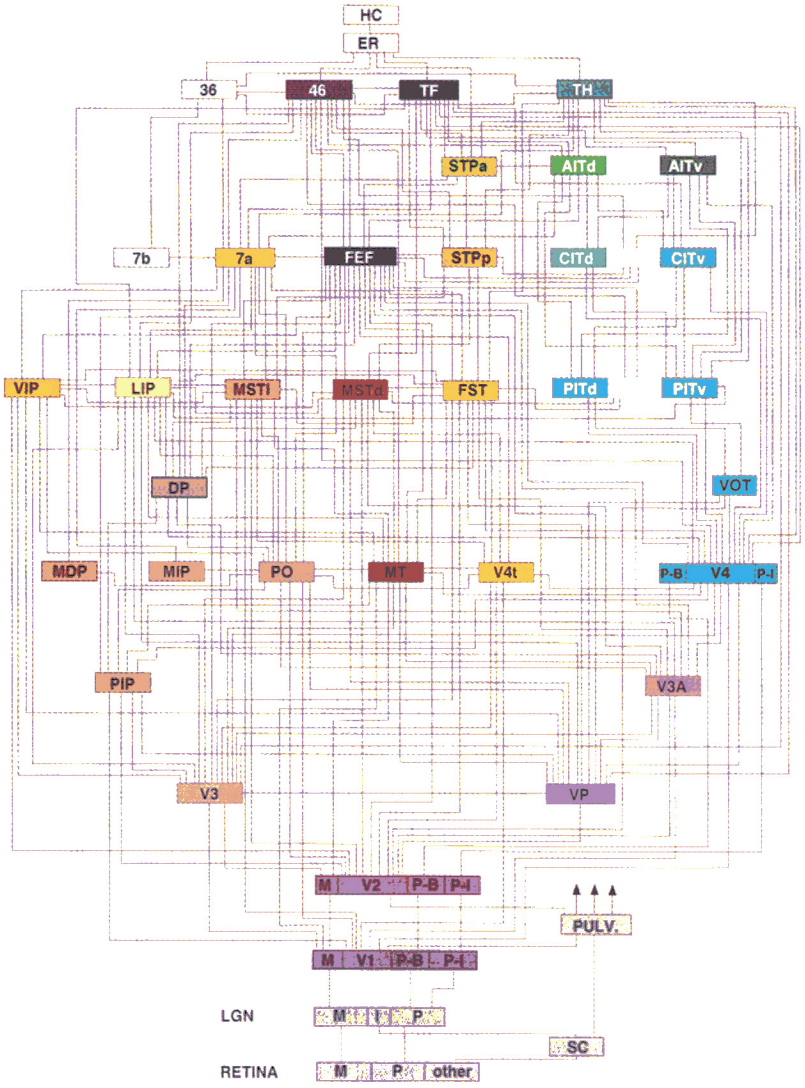


Figure 2.11: The most famous connection hierarchy of cortical regions according to [11]

2.2.3 Organization within Cortical Regions

The use of staining tracers also gave insight into some of the intra regional organization of the cortex. For example in region V1 it has been shown that there are subregions that receive input dominantly from one eye. These connection patterns are known as ocular dominance patterns (figure 2.12 by [12] and 2.13 by [13]).

It seems to be a general rule for cortical cells that neighbouring cells within a cortical area are always involved in similar tasks, not just in terms of the spatial topological mapping mentioned earlier, but also regarding the features that a cell is optimally stimulated with. So neighbouring neurons do not only tend to observe neighbouring areas in the real world but they also tend to react to similar features. Thus a stimulus that activates one cell is very likely to elicit a reaction also in the surrounding cells. Consequently, the optimal stimulus for cells shifts gradually as one moves along the cortical surface. Such stimulus properties in V1, for instance, are position

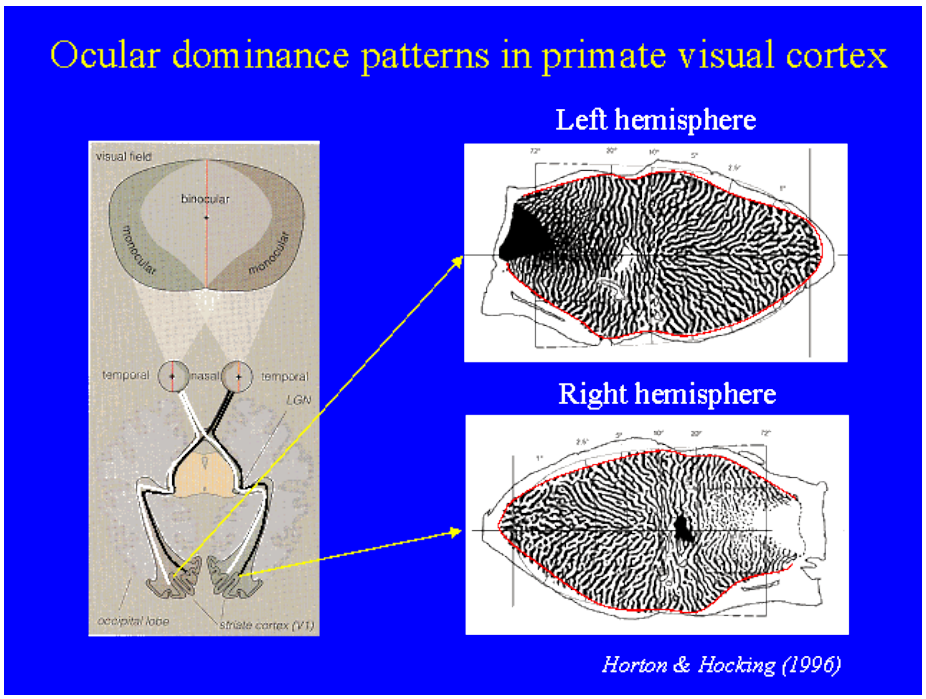


Figure 2.12: Ocular dominance patterns on V1 according to [12]

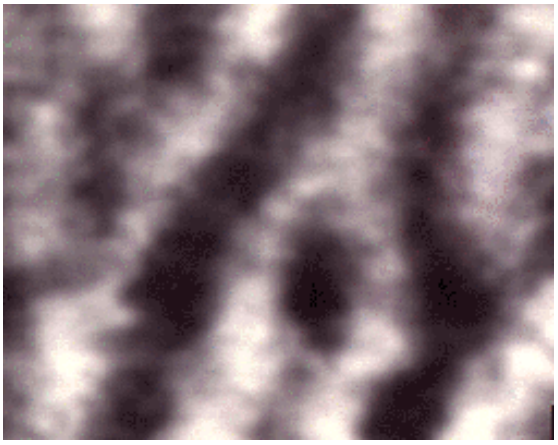


Figure 2.13: A close-up of ocular dominance patterns by [13]

in the visual field, ocular dominance, and orientation selectivity. That last property of V1 cells is that they are not optimally stimulated by a point in the visual field but rather by a bar of a particular orientation. Figure 2.14 shows regions on the V1 surface where the cells respond optimally to bars of a particular orientation. The orientation preference is indicated by colour and little bars.

2.2.4 Microcolumns and Cortical Layers

If the anatomy of cortex is examined it appears to be quite uniform across its surface. Researchers thus have still reason to hope that they can model cortex as an assembly of identical computational units and that only the connectivity pattern between those units defines the

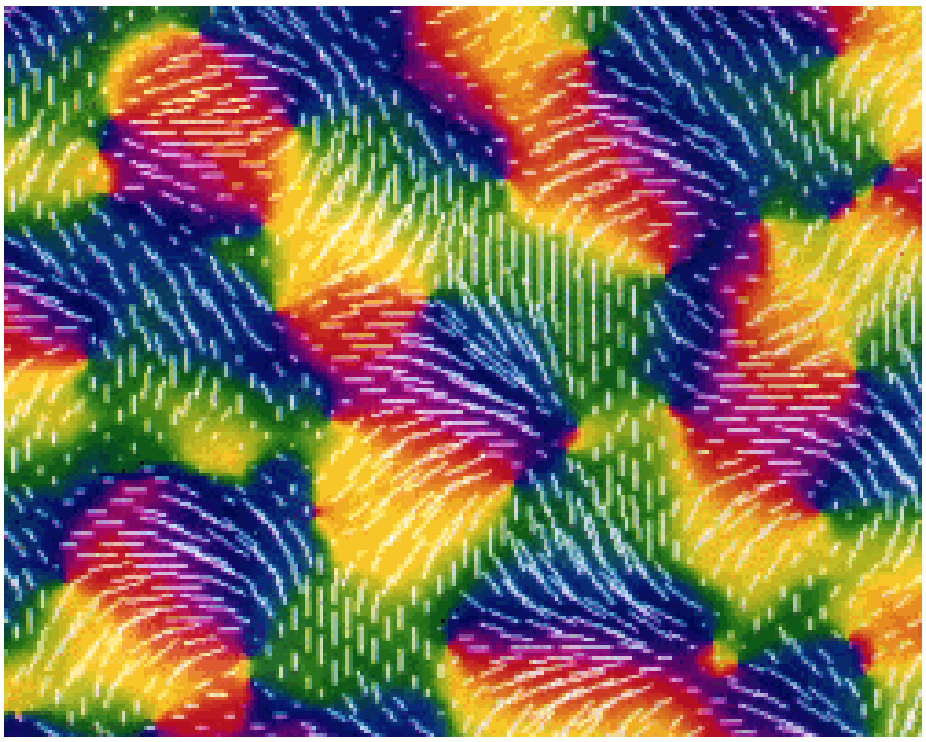


Figure 2.14: Orientation selectivity patterns [14]. The classic article of this author is [15].

system behaviour. A name that has been given to those units is 'cortical microcircuit' or 'microcolumn'. Since neurophysiological properties, e.g. occular dominance or direction selectivity, seem to be very similar accross a distance of about 1mm, these microcolumns are modelled as spanning $1mm^2$ of the cortical surface.

As one examines the structure through the cortex, i.e. vertical to its surface, a layered organization is observed. Anatomists distinguish 6 principal layers and some sublayers. The layers can be identified by their mixture of cell types and by their connectivity pattern: Where their input comes from and where they project to. It is for example known that layer 4 and to a lesser degree layer 6 in V1 receive the bulk of the direct sensory input from the Thalamus.

The layers can be made visible with different kinds of stainings. Figure 2.16 shows a light microscope picture of a vertical cut of V1 and figure 2.17 is a drawn interpretation of such a microscope picture that emphasises some relevant structures.

2.2.5 Neurons and Synapses

A big variety of neuron types can be found in the brain. Many anatomical studies are occupied with them and their connection patterns. One major distinction criteria is that of excitatory and inhibitory neurons. But also their form and area of occurrence identify them. Some basic insight on the operation principal of neurons have been gained by experiments conducted on the so called 'Giant Squid Axon'. In particular the spike generating

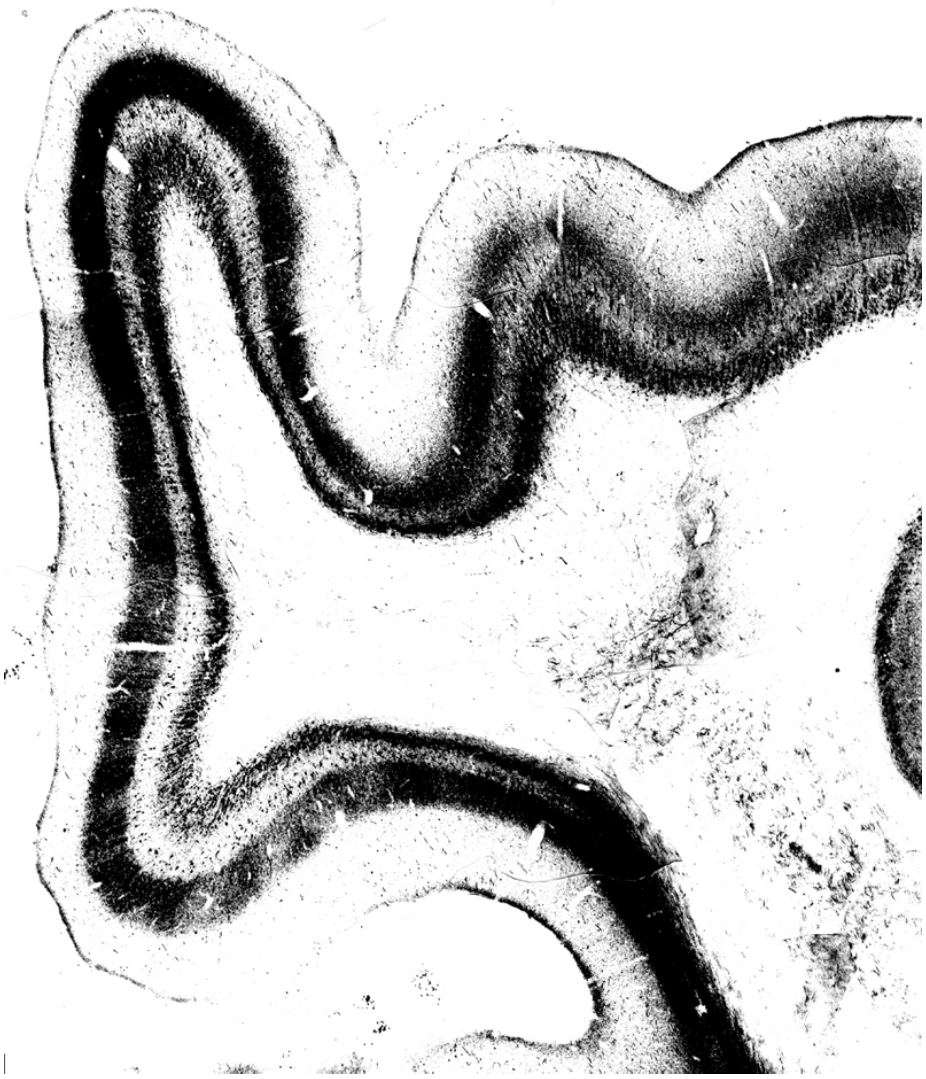


Figure 2.15: Staining of cortical layers in cat cortex [16] illustrtaion
I

mechanism has been studied on this special neuron. The role of sodium (Na+) and potassium (K+) ions have been studied by Hodgkin and Huxley [19]. More detail on their model will be discussed later in this script in a separate chapter (chapter 4). But in short: neurons receive charge packages as inputs triggered by voltage spikes (action potentials) that they receive from other neurons. Once the accumulated charge exceeds a threshold, they in turn produce an action potential that they send to other neurons through a narrow 'cable' called the axon.

The connection sites between neurons are called synapses (compare figure 2.18). In most cortical synapses the presynaptic terminal (part of the axon of the 'sending' cell) and the postsynaptic terminal (part of a dendrite or cell body of the 'receiving' cell) are actually physically separated by a narrow gap, the synaptic cleft. The transmission of the signal between the two terminals is conducted chemically. Vesicles with those chemicals (transmitters) are released from the presynaptic terminal when triggered

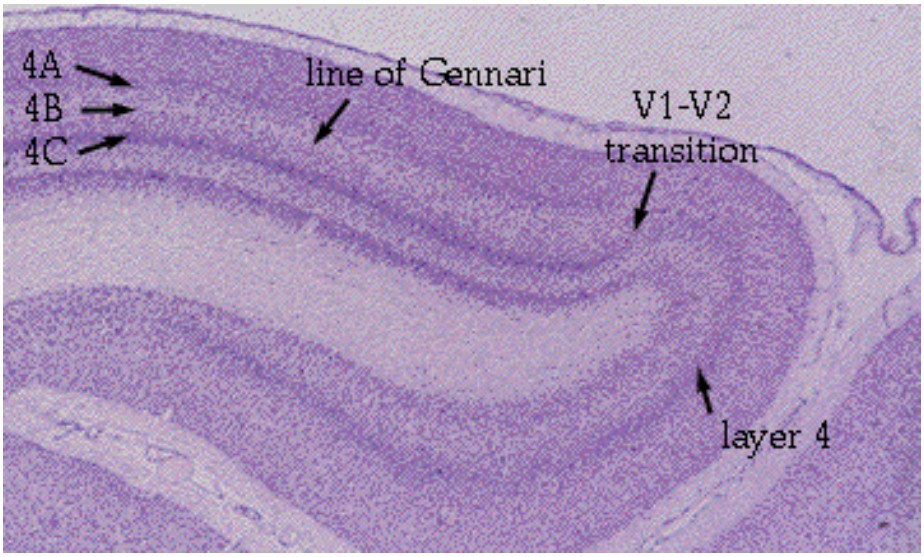


Figure 2.16: Staining of cortical layers in cat cortex [16] illustrtaion 2

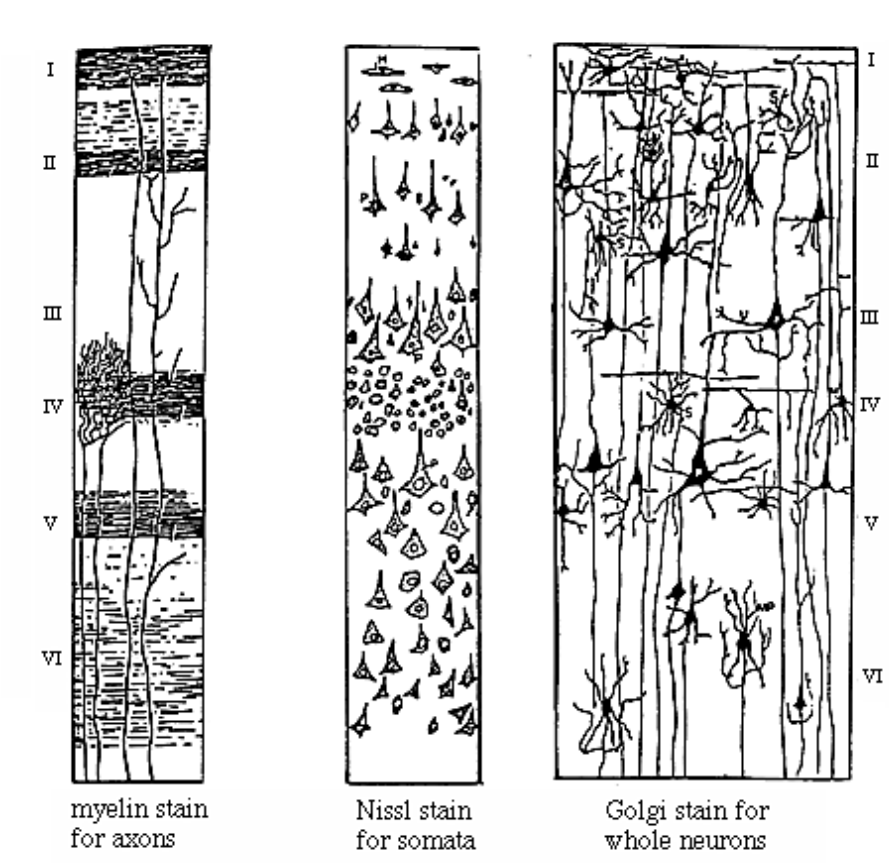


Figure 2.17: Different staining techniques revealing different features of cortical layers [17]

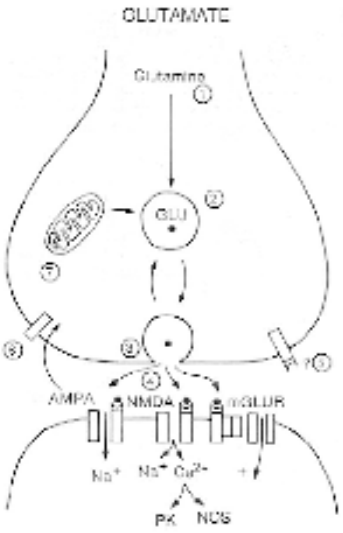


Figure 2.18: Exitatory glutamate synapse and some of its processes [18]

by an action potential. They cross the synaptic cleft and attach to ion channels on the post synaptic terminal which are thus opened. They then admit positively or negatively charged ions into the cell or release them from the cell, in effect adding or removing charge. Note that a sending neuron is exclusively excitatory or inhibitory, i.e. releasing the same type of transmitter on all its terminals.

The synapses play a very central role as they are believed to be the primary means of storing (learnt/experienced) information in the brain. Their connection strength, i.e. the amount of charge transmitted to the post synaptic cell per input spike, can change permanently and, thus, change the behaviour of the network. Neurophysiologist refer to the long term changes in synaptic strength as long term potentiation/depression (LTP/LTD) [20]. More recently the term spike timing dependent plasticity (STDP) has been coined to describe changes in synaptic strength triggered by temporal spiking patterns of a pre- and postsynaptic neuron [21, 22]. More on learning behaviour in artificial neural models in the learning chapter 10.

Chapter 3

Basic Analog CMOS

3.1 Field Effect Transistors

THE basic electronic element used on digital microelectronic chips is the transistor. It comes in different variants. We will only discuss FETs (field effect transistors) here (in particular CMOS-FETs (complementary metal oxide silicon field effect transistors)), which is the dominant transistor type nowadays. It is a four terminal device but often the 'Bulk' terminal is not explicitly represented and just connected to a supply voltage (Gnd for NFETs and Vdd for PFETs). Digital electronics uses them as a voltage controlled switch. The Gate voltage (terminal G) turns on or of the Drain (terminal D) current. However, in neuromorphic circuit the analog properties of this device are often used explicitly, according to the formulae given below. The subthreshold variant is most often used, because of its exponential behaviour that appears so often in nature too and because of the low current consumption in this mode of operation. The drawback is increased sensitivity to noise.

3.1.1 Basic Formulae

The CMOS-FET is a symmetric electronic device. The current from drain to source can be described as the difference between a forward current I_F and a reverse current I_R, both of which are dependent on the gate voltage and the voltage on either source or drain as given by the same equation (according to [23]):

$$I_{F(R)} = I_S ln^2 \left[1 + e^{\frac{V_G - V_{T0} - nV_{S(D)}}{2nU_T}} \right] \tag{3.1}$$

If the reverse current is so small as to be negligible, the transistor is said to be in *saturation*. This property mostly depends on the drain to source voltage V_{DS}. Check figure 3.2: On the right hand side, as I_D is almost independent on V_{DS} that's where the transistor is in saturation. On the left hand side, where I_D is increasing linearly and steeply, that's where the transistor is not saturated, also refered to as the triode region, where the transistor behaves like a resistor.

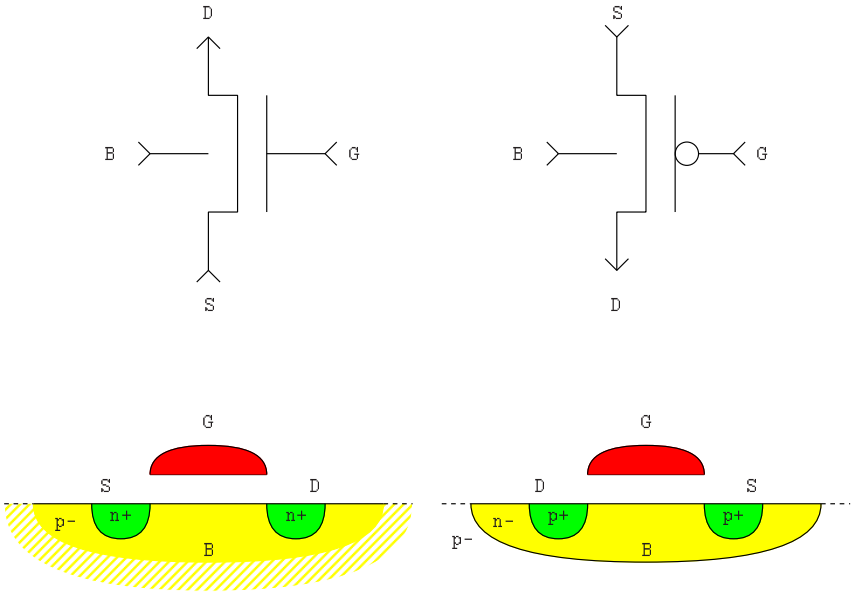


Figure 3.1: nFET and pFET in an nwell CMOS technology, symbol and cross section

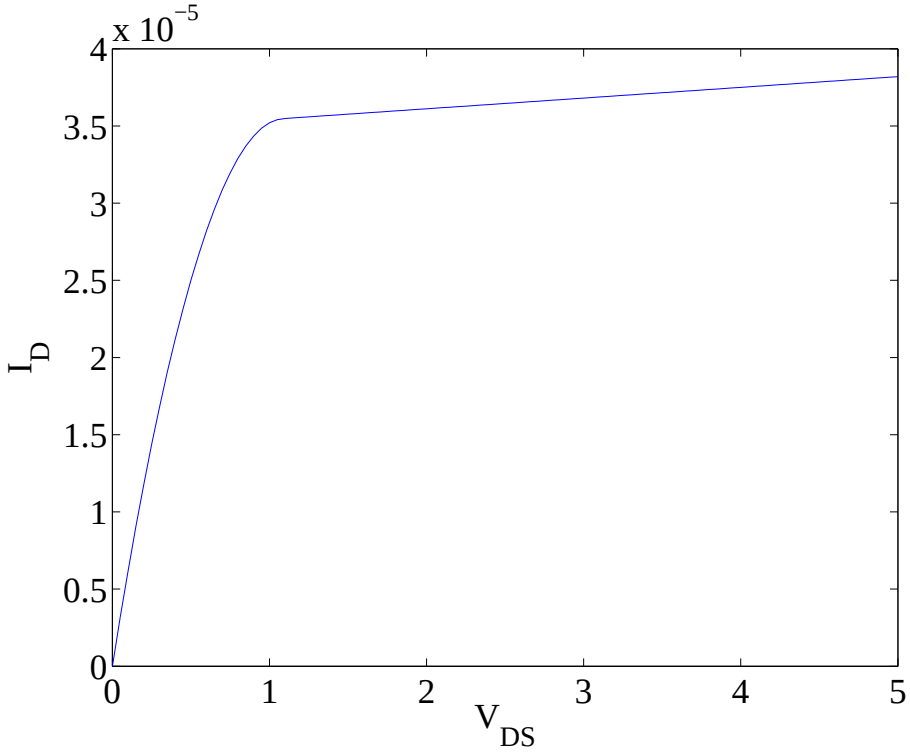


Figure 3.2: Transistor current I_{DS} in dependency of drain to source voltage V_{DS} including Early effect.

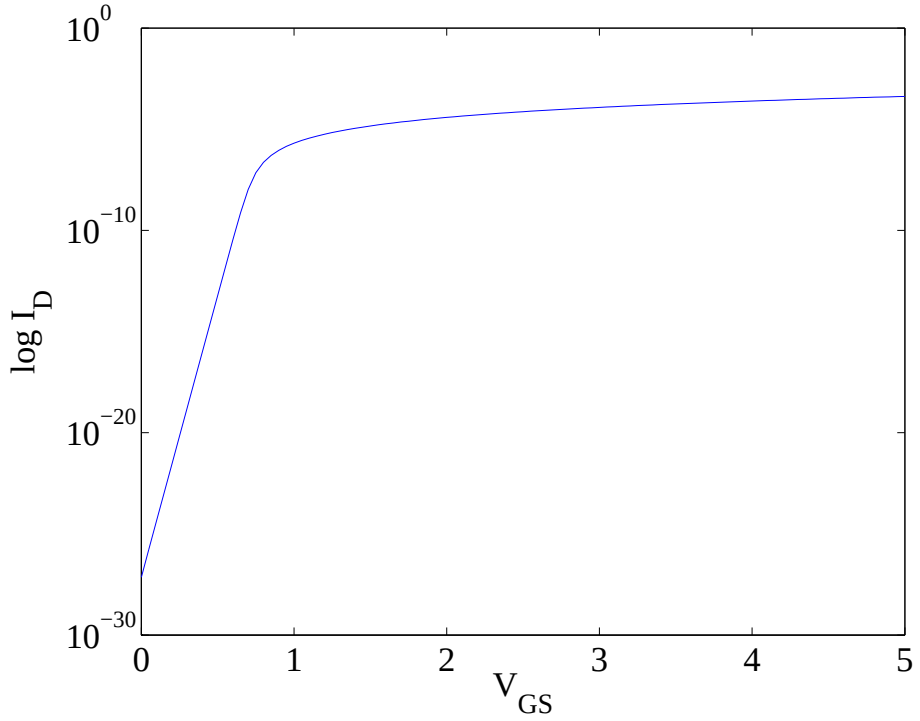


Figure 3.3: Transistor current I_{DS} in dependency of gate voltage V_G on a logarithmic scale. The logarithmic scale reveales that the current below threshold/in weak inversion is not zero but only very small and actually exponantially increasing. So, transistors can also be operated in the weak inversion regime with much smaller currents and still function somewhat similarly than in strong inversion. Even in the digital domain this has been recognized and ultra low power circuits are today operated in subthreshold.

If $I_F << I_S$ i.e. $V_G < (V_{T0} + nV_S)$ the transistor is said to be in *weak inversion*(Europe) or in *subthreshold*(USA) and (3.1) can be simplified to [1]:

$$I_F = I_S e^{\frac{V_G - V_{T0} - nV_S}{nU_T}} \tag{3.2}$$

This is best illustrated in figure 3.3 where one sees a nice linear straight dependency on a log scale on the lefthand side of the graph, i.e. where $V_G - nV_S$ is below the threshold voltage V_{T0}, which is at 0.7V in this example (typical for older CMOS processes that run with a supply voltage of 5V)

If $I_F >> I_S$ i.e. $V_G > (V_{T0} + nV_S)$ the transistor is said to be in *strong inversion* (Europe) or *above threshold* (USA) and (3.1) can be simplified to[2]:

$$I_{F(R)} = \frac{I_S}{4}\left(\frac{V_G - V_{T0} - nV_{S(D)}}{nU_T}\right)^2 \tag{3.3}$$

[1] Since $\ln(1 + f(x)) \approx f(x)$ for $f(x) \approx 0$, $\ln^2(1 + e^{\frac{x}{2}})$ becomes $\left(e^{\frac{x}{2}}\right)^2$ (for $x \ll 0$) and thus simply e^x

[2] Since $1 + e^x \approx e^x$ for $x \gg 0$, $\ln^2(1 + e^{\frac{x}{2}})$ becomes $\frac{x^2}{4}$

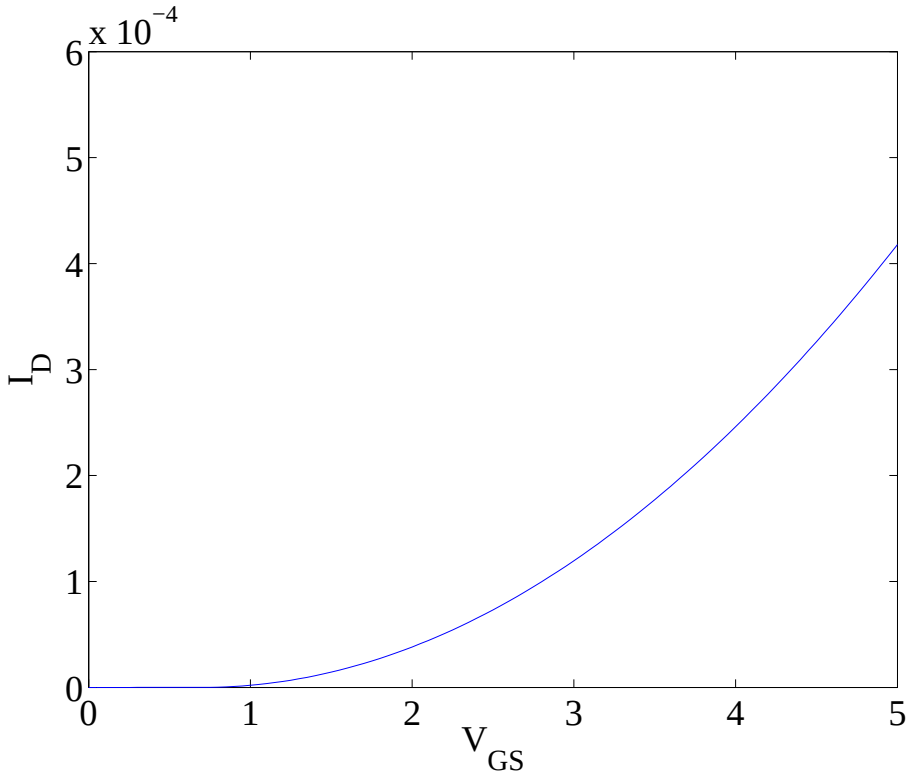


Figure 3.4: Transistor current I_{DS} in dependency of gate voltage V_G on a linear scale. The linear scale effectively hides the weak inversion regime. Only the strong inversion regime as V_G gets bigger than V_{T0} (0.7V in this example) is recognizably bigger than 0 and increases quadratically with the gate voltage. That's the 'classic' view.

This quadratic dependency can be seen above threshold in figure 3.4, which is the same graph as figure 3.3 but on a linear scale.

3.1.2 Early effect

A secondary effect that is neglected in these formulae, but never the less often important is the Early effect. It expresses a slight linear dependency of the saturated transistor current I_{DS} on the drain voltage V_D (see figure 3.2).

$$I_F^{\text{real}} = \frac{V_D + V_{Early}}{V_{Early}} I_F \tag{3.4}$$

Despite its being termed a 'secondary' effect, it is still quite important. It does, for instance, limit the voltage gain of operational amplifiers and is a major reason why simple current mirror implementations do not behave ideally (see section 3.3)

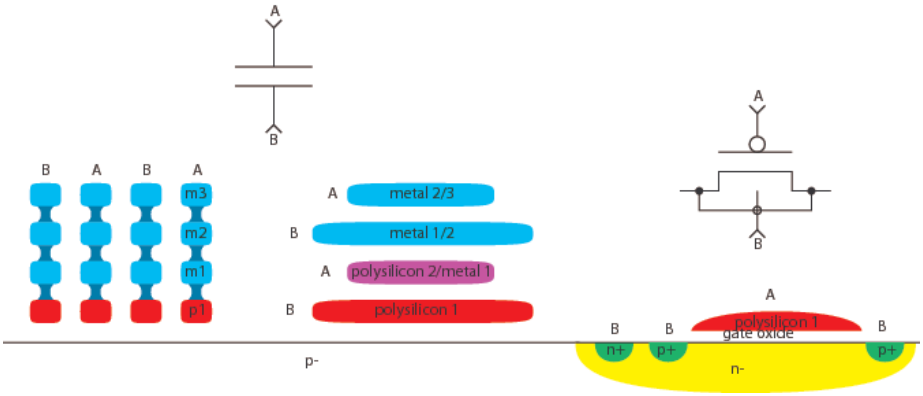


Figure 3.5: Capacitances in CMOS, symbol and cross section

3.1.3 Gate leakage

Gate leakage (or direct tunneling) is an effect that becomes significant in the most advanced CMOS technologies where the gate oxide is thinner than about 2-3nm. In effect one can observe a small current *through the gate oxide* which is a perfect insulator. So in Newtonian physics this current should not exist. In quantum physics this is possible because an electron is not simply located at a particular point in space. It is more a 'probability-cloud' of positions that remains undecided as long as no-one goes looking. That cloud is quite 'dense' accross a diameter of a few nano-meters, and thus there are electrons which 'extend' beyond the gate oxide. They may 'tunnel' from the gate to the channel and the probability of this happening increases if the gate oxide gets thinner.

In [24, 25] the current density J_g due to gate leakage/direct tunneling is modelled according to:

$$J_g = \begin{cases} A\frac{V_{ox}^2}{t_{ox}^2} e^{-\frac{B\left(1-\left(1-\frac{V_{ox}q_e}{\phi_{ox}}\right)^{\frac{2}{3}}\right)}{\frac{V_{ox}}{t_{ox}}}} & \text{if } V_{ox} < \frac{\phi_{ox}}{q_e} \\ A\frac{V_{ox}^2}{t_{ox}^2} e^{-\frac{B}{\frac{V_{ox}}{t_{ox}}}} & \text{if } V_{ox} > \frac{\phi_{ox}}{q_e} \end{cases} \tag{3.5}$$

where ϕ_{ox} is the energy barrier and q_e is the electron charge, thus $\frac{\phi_{ox}}{q_e}$ is the energy barrier in electron-volts. t_{ox} is the oxide thikness. A and B are parameters deduced from a number of physical constants, some of them process dependent. See [26] for example values.

3.2 Capacitors

The capacitor is an omnipresent device, intentionally or not. On microchips there are many unintentional 'parasitic' capacitances that influence the dynamic behaviour. Also intentionally placed capacitances influence the circuit dynamics. The behaviour can be described with a simple equation:

$$V = \frac{1}{C}Q \tag{3.6}$$

or as a differential equation if we differentiate (3.6) and substitute $\frac{\delta Q}{\delta t}$ with I.

$$\frac{\delta V}{\delta t} = \frac{1}{C} I \tag{3.7}$$

In neuromorphic and other analog designs they are often defining circuit time-constants or they are used to control floating nodes. In the later case one can actually move the voltage V_f on a floating node with a controlling capacitance C and a controlling voltage V_c applied to the other terminal of the capacitance according to:

$$\frac{\delta V_f}{\delta t} = \frac{\delta V_c}{\delta t} \frac{C}{C_{tot}} \tag{3.8}$$

where C_{tot} is the total capacitance of the node, parasitic or intended, assuming constant voltages on all other capacitor terminals.

In CMOS intentional capacitors can for example be constructed between polysilicon or metal layers as simple plate capacitors (figure 3.5, centre). Alternatively one can construct so called fringe capacitors, where 'fingers' extend into eachother and each finger actually extend over several matal and/or polysilicon layers coupled by vias. Thus, one constructs vertical capacitor plates (figure 3.5, left). Another means to construct a capacitance, is as MOS-capacitors. MOS-capacitors use the capacitance across the gate oxide as shown to the right in figure 3.5. This results in a really high capacitance, since the gate oxide is the thinnest insulating layer in a CMOS process and a plate capacitor's capacitance increases the closer the plates are to each other. That's necessary for it's usual use in field effect transistors, where the gate needs a strong capacitive coupling with the transistor channel. Unfortunately, MOS-capacitors have a drawback: their capacitance is not constant, as the capacitance changes dependent if the 'channel' is 'depleted' or even 'inverted', i.e. equivalent to the operation regimes of the FET. Thus, the capacitor based on a pFET depicted in figure 3.5 will have constant capacitance when it is in accumulation ($V_A > V_B$) and in strong inversion ($V_A \ll V_B$), but in depleation and somewhat in weak inversion ($V_A \approx V_B - VT0$) the capacitance drops down to a minimum of about $1/3$ of its normal value.

3.3 Current Mirror

The basic current mirror (figure 3.6) creates a 'copy' of a current. That's because the gate voltage V_{GS} is the same for the input and output transistors, and it is forced to the appropriate level by the feedback connection from drain to gate for the input transistor.

In order to work properly one has to take care that both transistors are in saturation. And even if they are, the early effect will introduce a slight error into the output current, if the drain-to-source voltage is different for the two transistors.

Keeping these restrictions in mind, the output current is the same as the input current, provided that the two transistors have the same width to

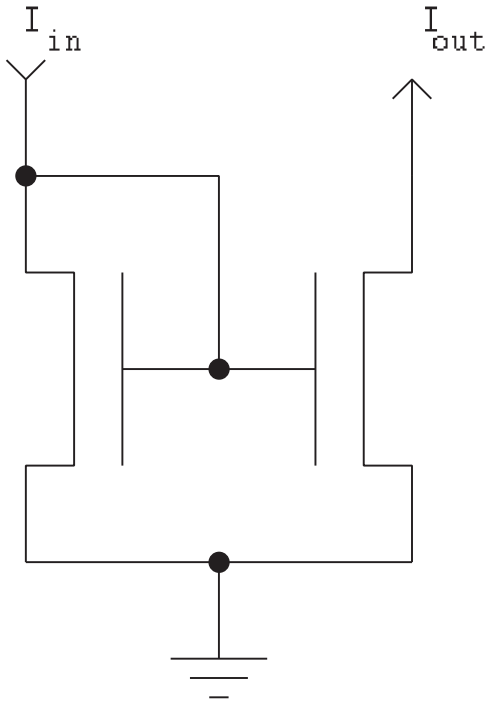


Figure 3.6: Schematics of a basic CMOS current mirror.

length ratio (W/L). If W/L is different for the two, the output will be a weighted copy of the input:

$$\frac{I_{out}}{I_{in}} = \frac{W_{out}L_{in}}{W_{in}L_{out}} \tag{3.9}$$

The current mirror is especially useful if the way a current is used could potentially influence the current source. In that case it is better to work on a copy without interfering with the original current.

3.4 Differential Pair

A classic circuit (figure 3.7) that is the basis of many more complex circuits. It comes into play whenever voltages or currents are compared. In a first approximation, the bias transistor can be seen as a current source supplying a bias current I_b. According to Kirchhoff's law, that bias current needs to be supplied by the two branches of the differential pair. And according to the transistor characteristics in subthreshold and in saturation those two currents are given by the gate voltages V_1 and V_2 and the source voltage V_C that is common to those two transistors. These facts lead to the formula:

$$I_b = I_1 + I_2 = I_S e^{\frac{-V_{T0}-V_C}{nU_T}} \left(e^{\frac{V_1}{nU_T}} + e^{\frac{V_2}{nU_T}} \right) \tag{3.10}$$

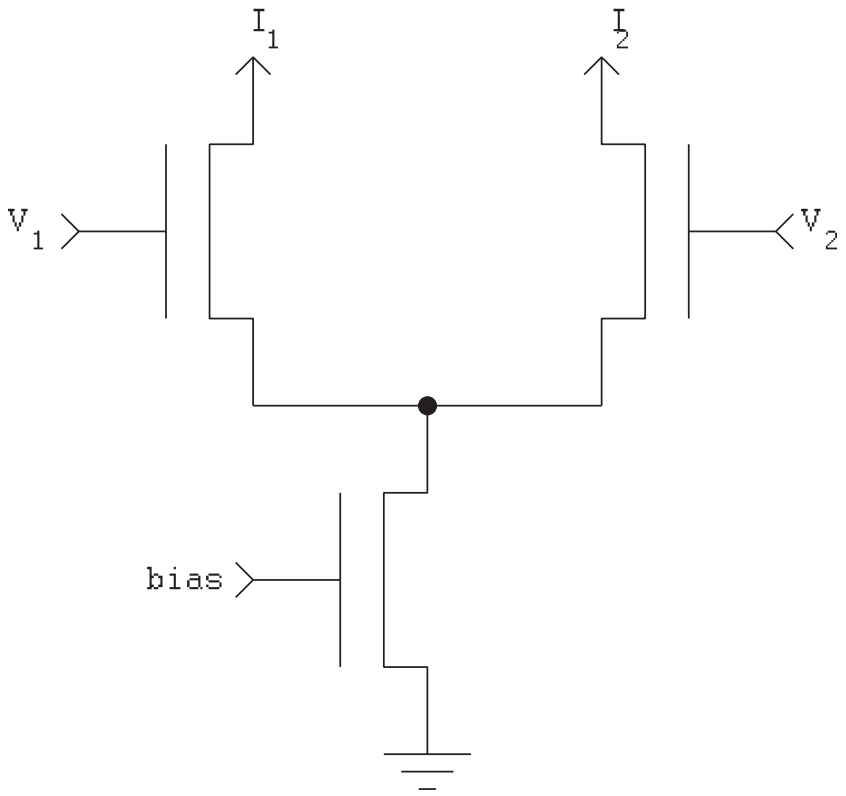


Figure 3.7: A differential pair

One thing one can see from that formula is that the ratio of the two currents will be exponentially dependent on the difference of the input voltages:

$$\frac{I_1}{I_2} = \frac{I_1}{I_b - I_1} = e^{\frac{V_1 - V_2}{nU_T}} \tag{3.11}$$

So for just a small difference in voltage, the bigger of the two currents will rapidly approach I_b. In other words: the circuit can almost be seen as a element with a binary output, where all the bias current is supplied only by the branch with the bigger gate voltage as input.

3.5 Transconductance Amplifier

The transconductance amplifier in its basic form is the father of all CMOS amplifiers (figure 3.8). Based on a differential pair it amplifies the difference of two input voltages in a range of up to a few 100mV. Beyond that range it works as a comparator, identifying the bigger input, since the output saturates.

It is a combination of a differential pair and a current mirror. The current mirror projects one of the currents in the differential into the other branch

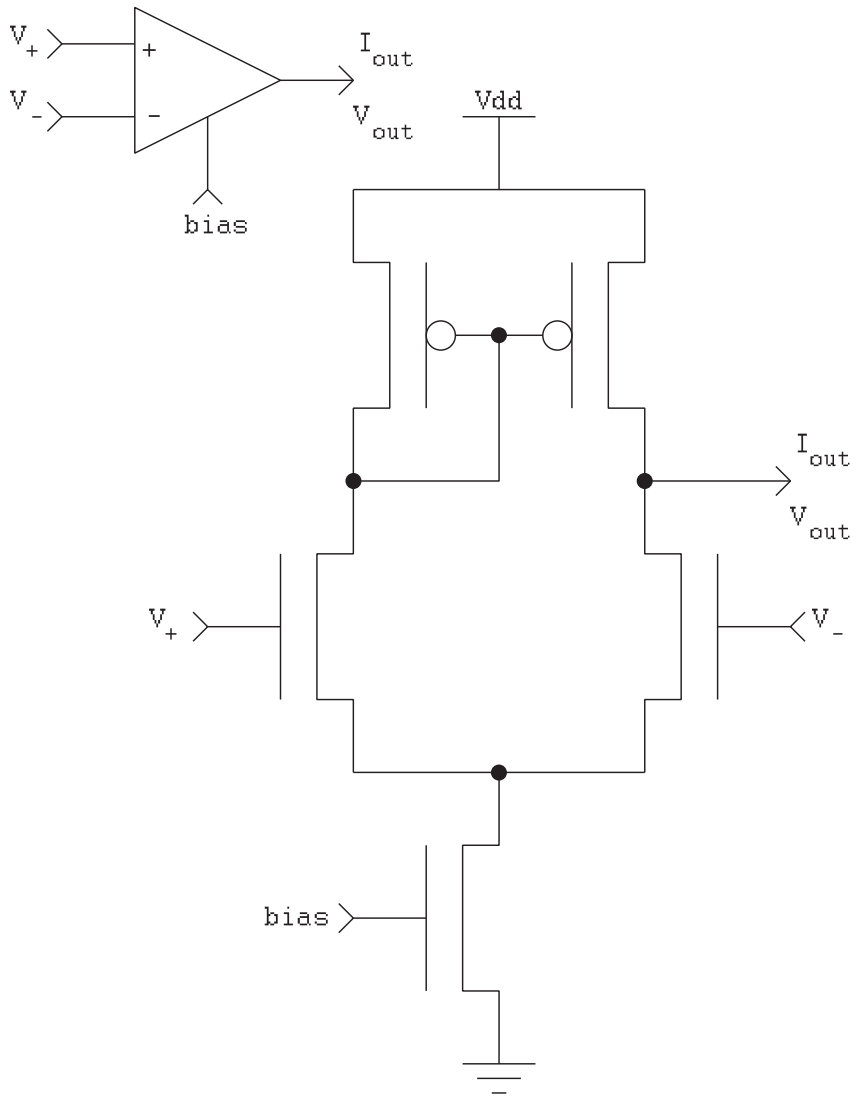


Figure 3.8: The basic transconductance amplifier

of the diff-pair. In other words: the output current from that node is the difference of the currents in the two branches in the diff-pair. Its characteristics in the subthreshold can be deduced to follow this equation:

$$\begin{aligned} I_{out} &= I_b \frac{e^{\frac{V_+}{nU_T}} - e^{\frac{V_-}{nU_T}}}{e^{\frac{V_+}{nU_T}} + e^{\frac{V_-}{nU_T}}} = I_b \tanh \frac{V_+ - V_-}{2nU_T} \\ I_{out} &= g(V_+ - V_-) \text{ in the linear region where } g = \frac{I_b}{2nU_T} \end{aligned} \tag{3.12}$$

That's if you hook up a load of zero resistance to the output, i.e. the output *voltage* does not budge but only the output *current* is influenced by the input voltage. Vice versa, if you hook up an infinite output resistive load (i.e. no load), then no *current* will flow from the output and your output signal will be a *voltage*. The equation giving that voltage gain

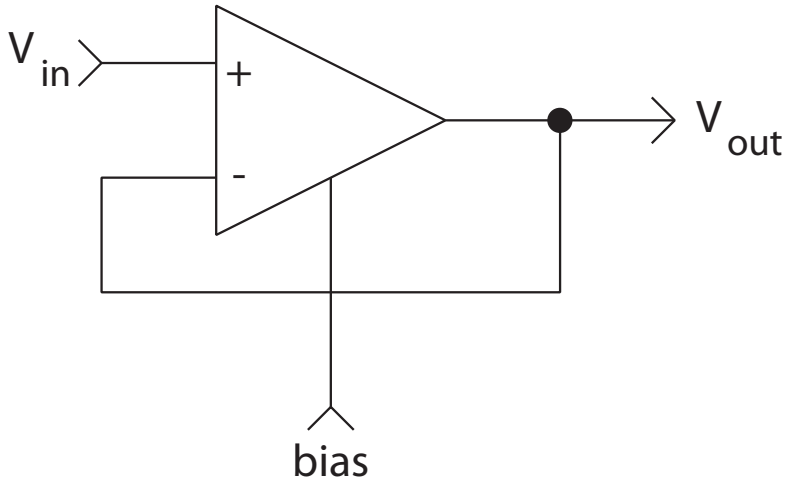


Figure 3.9: A follower, or analog voltage buffer

is less straight forward and mainly dependent on the Early effect/output resistance of the output transistors, but in quality it will also be a sigmoid characteristic, quite like a tanh. The voltage gain A in the linear region will be the transconductane g multiplied with the output impedance which is roughly half the Early voltage.

3.6 Follower

A very nice circuit that allows to observe an analog voltage with minimal interference. It is just an amplifier with its output fed back onto the minus input (figure 3.9). Thus, the amplifier will always simply try to compensate any difference between the input and the output. If the amplifier has infinite transconductance and infinite output impedance, the output voltage will faithfully follow the input. In reality, there is a small offset between the two, composed of a constant offset and a relative offset.

3.7 Resistor

A resistor is a very useful element that is unfortunately hard to come by in any usefully big strength on a CMOS chip, at least not without using up a lot of layout space. It is often just a cable of polysilicon with a resistance in the order of a few Ohms per square (figure 3.10). Many CMOS processes do now offer high-resist polysilicon as well, approaching kilo-Ohms per square. But still, if many resistors are required, the space consumption is often too big. In many models intended for CMOS implementation resistors are, thus, replaced by followers which has similar but not quite the same properties as long as it remains in its linear range of operation.

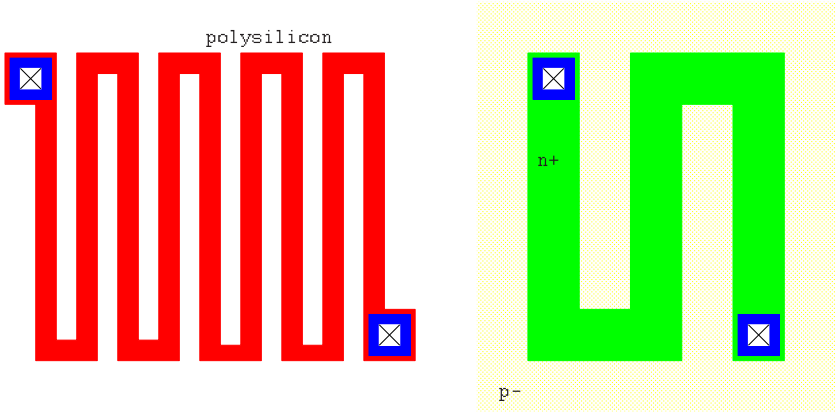


Figure 3.10: Possible resistor implementations in CMOS

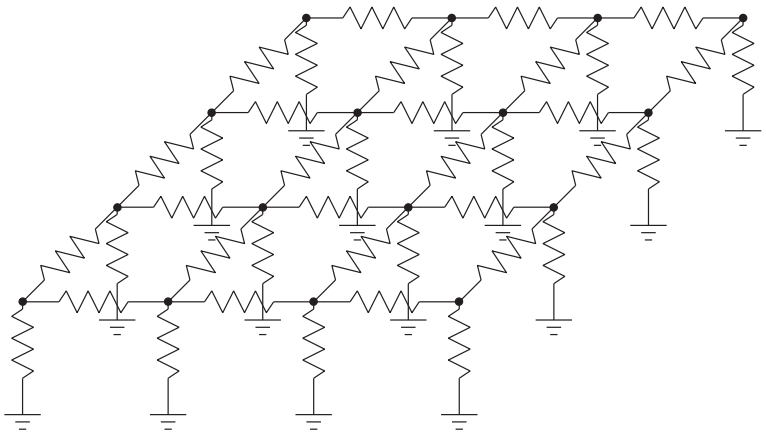


Figure 3.11: A resistive net, often used for spatial smoothing of an array of analog inputs.

3.8 Resistive Nets

resistive nets can compute local averages (figure 3.11) in a 1D or 2D array: When injecting a current (e.g. a sensor input) into a node of the net it will rise the voltage of that node and to an exponentially declining degree the voltage of the neighbouring nodes. Thus it is for example used in the 'silicon retina'. The equation governing its behaviour is the following:

$$\frac{V}{R_V} = \frac{\delta^2}{\delta x^2 \delta y^2} \frac{V}{R_H} \tag{3.13}$$

As mentioned before, it is unfortunately quite layout-space consuming to implement resistors in CMOS. Thus, diffuser nets instead of resistive nets

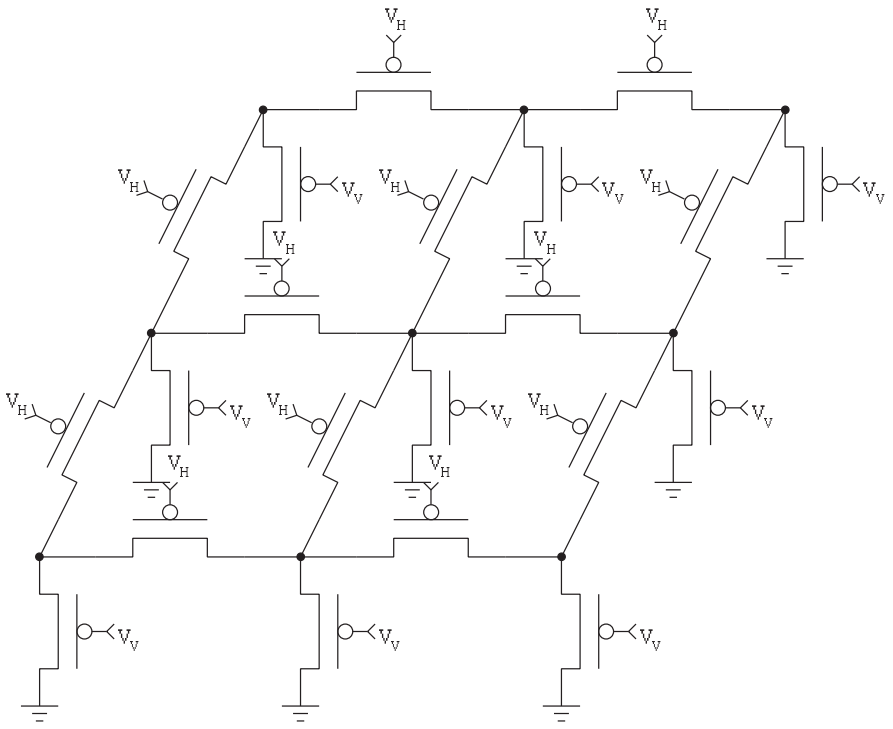


Figure 3.12: A diffuser net, linear in current mode.

are often used (figure 3.12). They replace the resistors with transistors with fixed gate voltage. This network behaves the same as a resistive network if one only observes the currents. This is nicely deduced in [23] for subthreshold by defining a pseudo voltage V^* that is exponentially dependent on the voltage, and a pseudo resistance R_H^* and R_V^* respectively.

$$\begin{aligned} I &= \frac{V^*}{R^*} \\ \frac{V^*}{R_V^*} &= \frac{\delta^2}{\delta x^2 \delta y^2} \frac{V^*}{R_H^*} \\ V^* &= -e^{\frac{-V}{U_T}} \\ \frac{1}{R^*} &= g^* = I_S e^{\frac{V_G - V_{T0}}{nU_T}} \end{aligned} \tag{3.14}$$

3.9 The Winner Take All Circuit

The winner take all (WTA) circuit (figure 3.14) is certainly not a standard circuits in analog CMOS electronics books. But it has become one for the neuromorphic engineers. A WTA circuit/network filters out the strongest of a number of inputs, namely the 'winner'. In mathematical terms it is a function $\overrightarrow{\text{wta}}(\vec{x}) : R^n \to R^n$ with

$$\overrightarrow{\text{wta}}(\vec{x}) = \sum_{i \in S} \vec{e}_i, \; S = \left\{ s : s \in \mathbb{N} \wedge s \leq n, x_s = \max_{j \in \mathbb{N} \wedge j \leq n} x_j \right\} \tag{3.15}$$

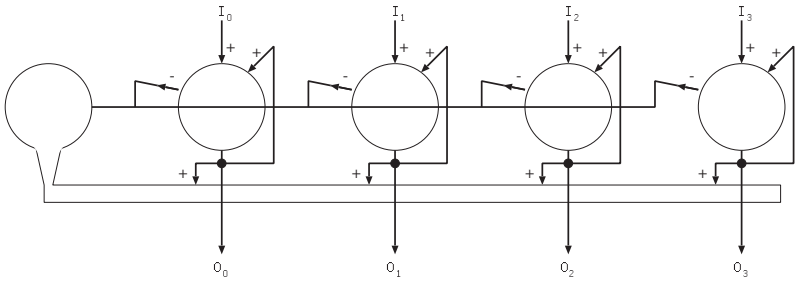


Figure 3.13: Neuronal winner take all principle.

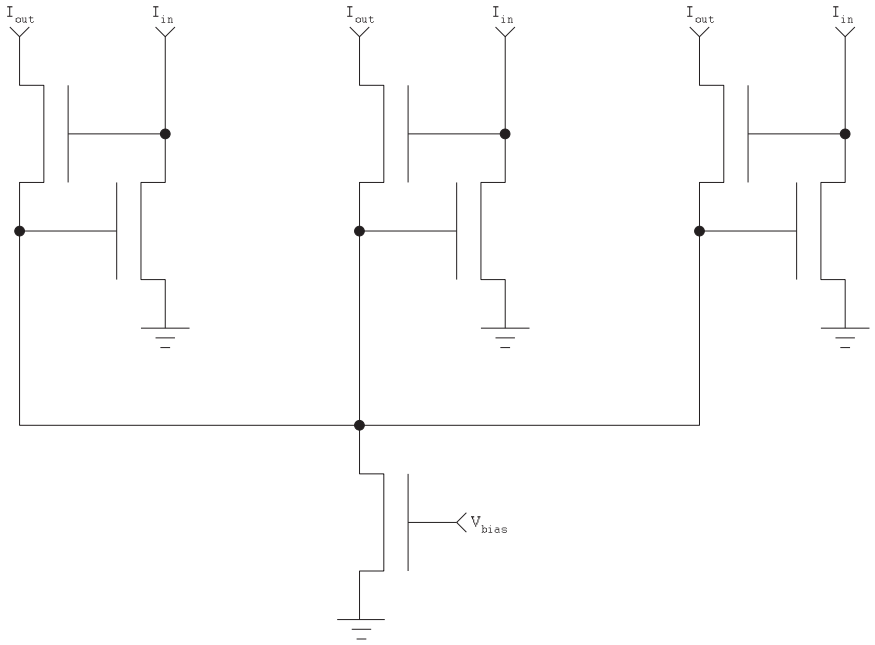


Figure 3.14: Basic analog CMOS WTA

Where $\vec{e}_i$ is the unity vector (a vector with one element that is 1 and all others equal to zero) in direction of dimension i. In words this formula means: $\vec{wta}(\vec{x})$ is a vector with all elements equal to zero except for where $\vec{x}$ is maximal. those are equal to 1. Since several entries in $\vec{x}$ can be maximal the expression (3.15) is somewhat cumbersome with its sum of unity vectors, just to be entirely correct.

But actually also variants of that function count as WTA networks. In particular if the output of the winner(s) is not just 1 but actually the original input (the corresponding maximal value of the maximal element in $\vec{x}$) to that node.

A neural network implementation of that function is sketched in figure 3.13. Global inhibition exerts inhibition on all neurons that is equal to the sum of the activity of all neurons. All neurons receive one excitatory external input and feedback from their own output. When running this

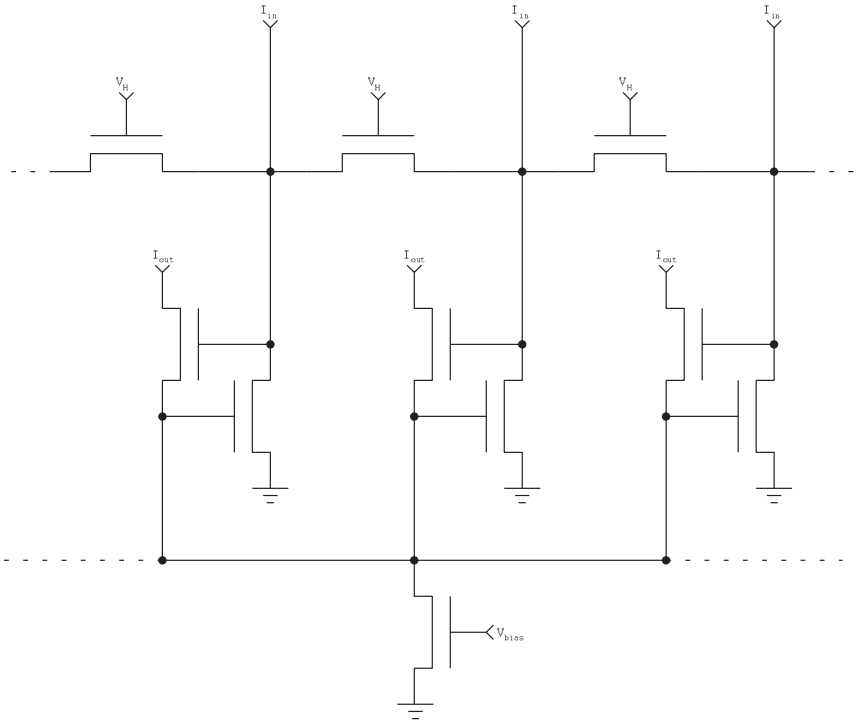


Figure 3.15: WTA with spatial smoothing/ cross excitation.

setup recursively, the only stable state of that network is that in which only the neuron with the strongest input remains active.

The WTA circuit in figure 3.14 is actually based on a differential pair, only that we don't just talk of a pair here but possibly of several more than two branches. In figure 3.14 there are three branches shown. The basic differential pair really is a WTA circuit already: with quite a sharp transition, all the bias current flows through that branch with the biggest gate voltage. In the WTA circuit the differential pair structure can be seen when considering only the output current branches. The gate voltage of the transistor in this differential pair structure is set by a feedback loop however. The inputs are currents. The feedback loop drives the voltages even further apart, for a small difference in input currents.

There exists extensions of the WTA circuit, e.g. the one in figure 3.15. Basically it connects the WTA input in a diffuser network. The effect of this is that strong neighbours can collaborate to make one node the winner, even if there are other isolated nodes that by themselves would be stronger but do not receive support from their neighbours. This supportive collaboration is often refered to as cross-excitation.

Another extension is known as cross-inhibition. This time each output branch has it's own bias current supply and they are also connected in a diffuser network. The outputs of the basic WTA network can be seen as competing for the bias current as their exclusive output current, 'inhibiting' the other output nodes by taking their share of output current away from them. In this extension here in figure 3.16 the strength of the cross-

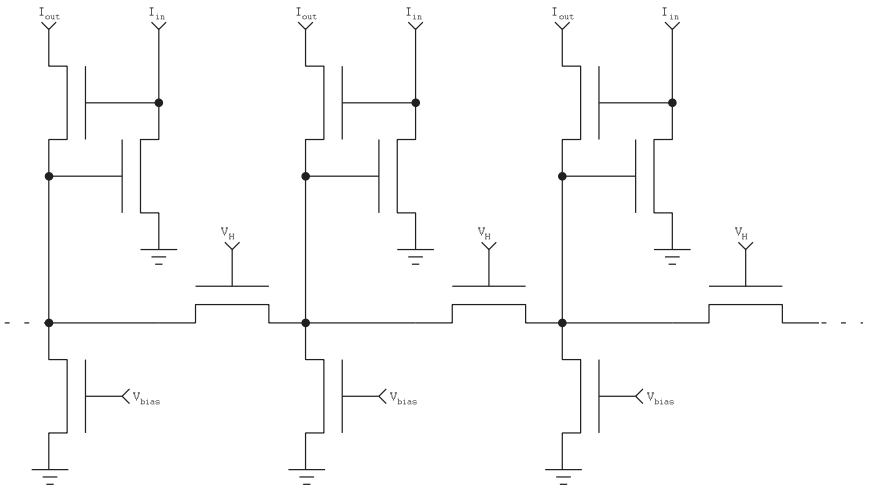


Figure 3.16: WTA with local winners/local cross inhibition

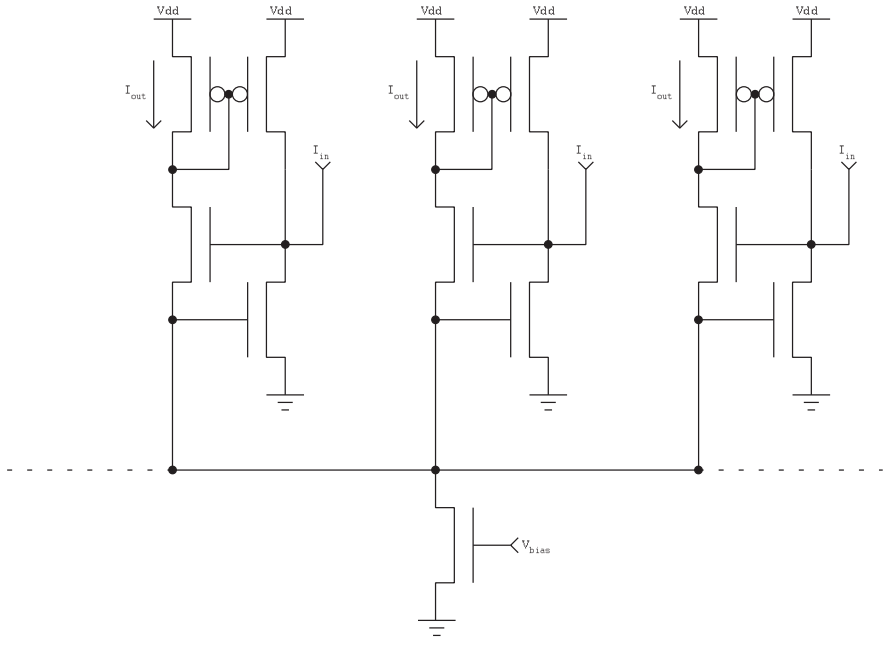


Figure 3.17: WTA with hysteresis

inhibition can be regulated by the diffuser network, e.g. in the extereme case of the diffuser network being turned off completely, there is no competition/cross-inhibition and each output branch receives its own bias current, and in the other extreme case of no resistance in the diffuser network we are back to the basic WTA operation where a single exclusive winner receives all the bias currents. If the diffuser network is tuned to something inbetween, the competition is localized, i.e. there can be severall winners in the entire network, but only one in a local region. the size of this region is determined by the horizontal resistance in the diffuser network.

A last modification is the hysteretic WTA (figure 3.17). Hysteresis is the

tendency of a system to maintain its present state.

The best known example is the Schmitt-trigger employed in digital åphysical interfaces to supress ringing when input signals are slow or noisy. It's an inverter with a switching threshold dependent on its presnt state, i.e. a higher switching threshold if the output is high and a lower swiching threshold if the output is low. Thus, if a slowly rising input signal makes the Schmitt-trigger's output go low, input noise just at the switching threshold will not cause the output to oscillate/ring.

In the WTA variant here (figure 3.17), the output current is fed back to the input. This gives the present winner an advantage and an increasing input on another node cannot easily dethrone it: in order to do so, it is not sufficient to receive a bigger input than the presnt winner, but the input must be bigger by a good margin, by a margin as big as the bias current, to be precise.

Chapter 4

Real and Silicon Neurons

4.1 Real Neurons

The neuron can be seen as the basic information processing unit of the brain of which it has around 10^{11}, each with about 10^4 connections to others of its kind. 'Neuron' is the name for a biological brain cell that is an infinitly complicated organism and the arena for many chemical and electric processes. It is the way of science to try to understand such complicated systems by formulating simplified models thereof. A crude anatomical model of a neuron is shown in the figure 4.1. Of course, there are much more detailed models. Be aware that a lot of research has gone into describing different classes of neurons in great detail.

Synapses are the connection sites between neurons. They usually connect axons and dendrites. The axon carries the neuron's output voltage pulse, the so called action potential (AP). Dendrites are the input sites receiving current input that flows to the soma and is integrated there. The axon hillock is the site where an action potential is initiated as the integrated input current (often referred to as the membrane voltage V_m) amounts to a voltage higher than some threshold.

Figure 4.2 gives an impression of how a real neuron looks like. It is a light microscope photograph kindly provided by John Anderson, Institute for Neuroinformatics, Zürich Switzerland. He also provided the reconstruction of the dendritic tree of a neuron (a cortical pyramidal cell) shown in figure 4.3, the result of many days of work. The bar at the bottom represents 100 micro meters.

For an idea of the electrophysiology of a real neuron, please consult section 4.2.4 about the most detailed models used to simulate neural behaviour.

4.2 aVLSI Models of Neurons

Neuromorphic engineers are more interested in the physiolocical rather than the anatomical model of a neuron though, which is concerned with the funcionality rather than only classifying its parts. And their preference lies with models that can be realized in aVLSI circuits. Luckily many of

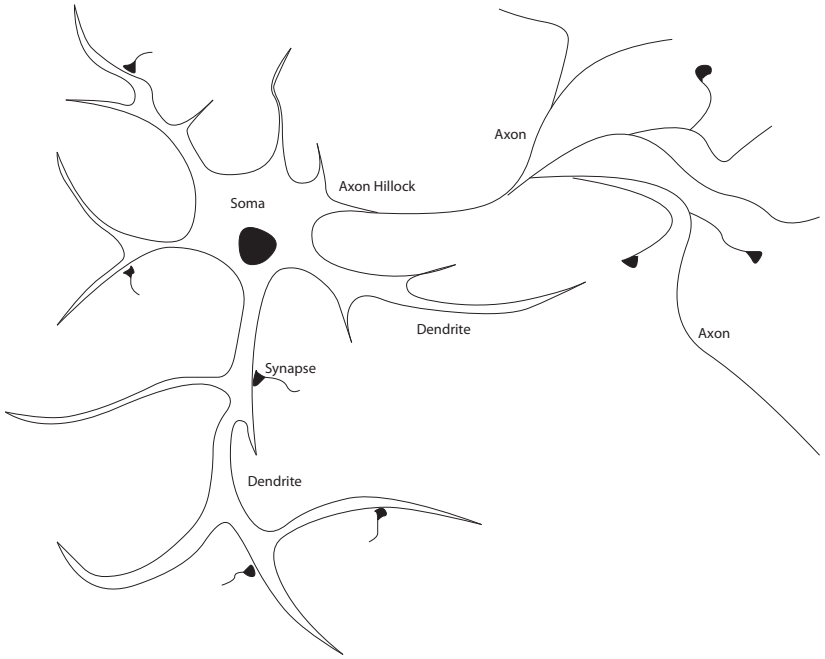


Figure 4.1: Basic anatomical parts of a Neuron

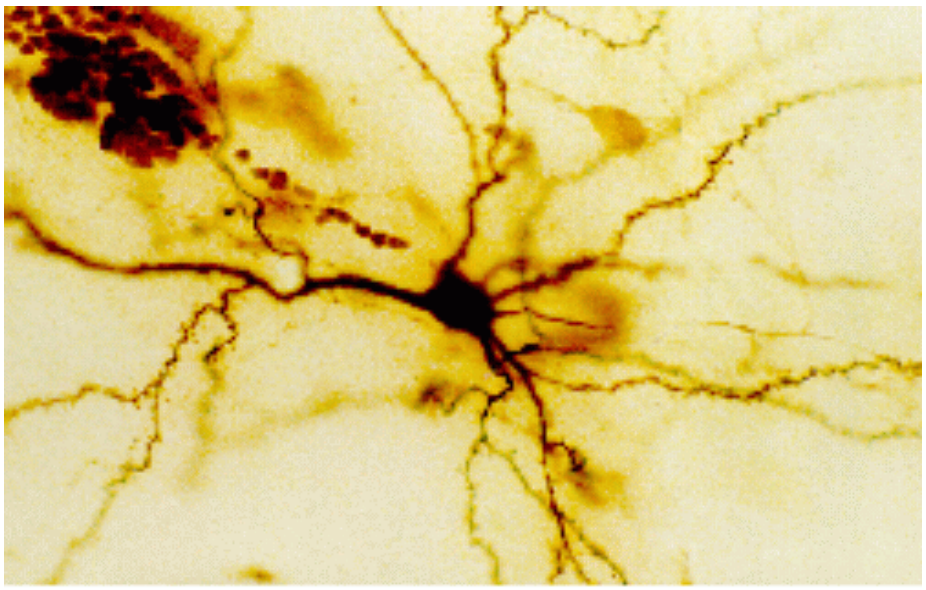


Figure 4.2: Light microscope photograph of a stained neuron

Figure 4.3: The 3D reconstruction of a cortical layer 5 pyramidal cell.

the models of neurons have always been formulated as electronic circuits since many of the varyiing observables in biological neurons are voltages and currents. So it was relatively straight forward to implement them in VLSI electronic circuits.

There exist now many aVLSI models of neurons which can be classified by their level of detail that is represented in them. A summary can be found in table 4.1. The most detailed ones are known as 'silicon neurons'. A bit cruder on the level of detail are 'integrate and fire neurons' and even more simplifying are 'Perceptrons' also known as 'Mc Culloch Pitts neurons'. The simplest way however of representing a neuron in electronics is to represent neurons as electrical nodes.

electrical nodes	most simple, big networks implementable
perceptrons	mathematically simple, but complicated in aVLSI
integrate and fire neurons	mathematically complex, but simple in a VLSI
compartemental models	complex, simulation of big networks are very slow

Table 4.1: aVLSI models of neurons

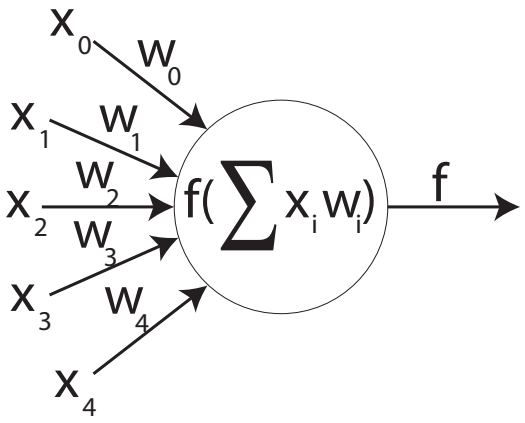


Figure 4.4: The mathematical model of a Perceptron

4.2.1 Simple Electrical Nodes as Neurons

The most simple of all neuronal models is to just represent a neuron's activity by a voltage or a current in an electrical circuit, and input and output are identical, with no transfer function inbetween. If a voltage node represents a neuron, excitatory bidirectional connections can be realized simply by resistive elements between the neurons. If you want to add the possibility for inhibitory and monodirectional connections, followers can be used instead of resistors. Or if a current represents neuronal activity then a simple current mirror can implement a synapse. Many useful processing networks can be implemented in this manner or in similar ways. For example a resistive network can compute local averages of current inputs.

4.2.2 Perceptrons (Mc Culloch Pitts neurons)

A perceptron is a simple mathematical model of a neuron. Like real neurons it is an entity that is connected to others of it's kind by one output and several inputs. Simple signals pass through these connections. In the case of the perceptron these signals are not action potentials but time discrete (i.e. static) real numbers. To draw the analogy to real neurons these numbers may represent average frequencies of action potentials at a given moment in time. The output of a perceptron is a monotonic function (refered to as activation function) of the weighted sum of its inputs (see figure 4.4).

Perceptrons are not so much implemented in analog hardware. They have originaly been formulated as a mathematical rather than an electronic model and traditional computers and digital hardware are good at those, whereas it is not so straight forward to implement simple maths into

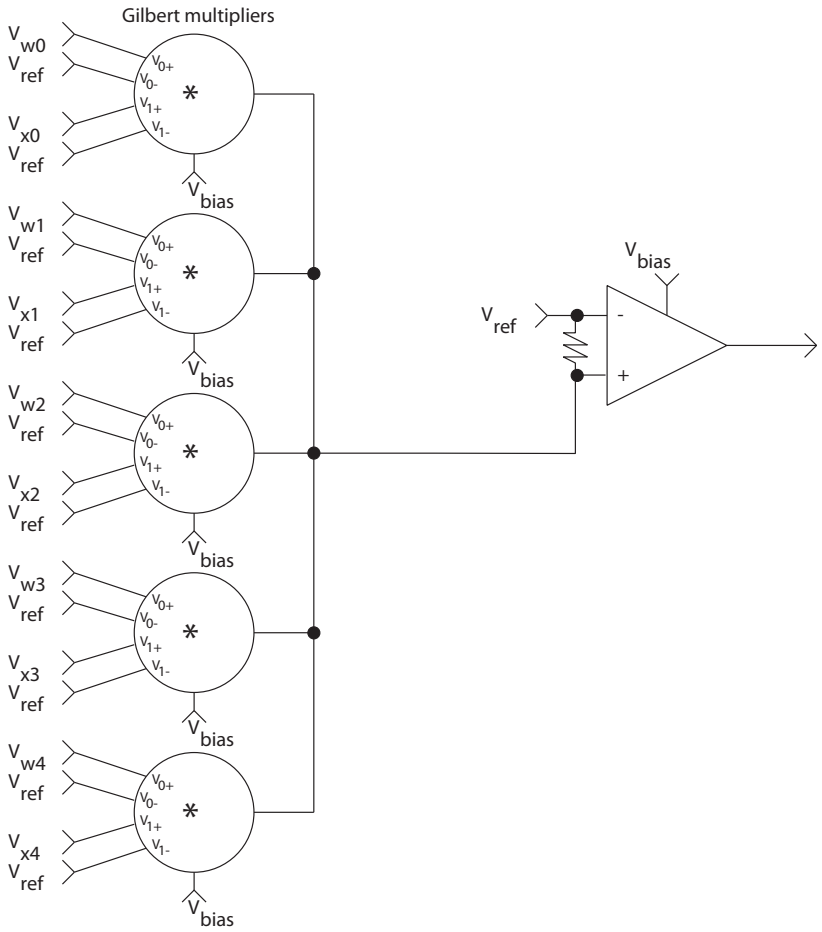


Figure 4.5: A possible implementation of a Perceptron

aVLSI. Still, there do exist aVLSI implementations of perceptrons since they still promise the advantage of a real fully parallel, energy and space conservative implementation.

A simple aVLSI implementation of a perceptron is given in the schematics in figure 4.5. This particular implementation works well enough in theory, in praxis however it is on one hand not flexible enough (particularly the activation function), on the other already difficult to tune by its bias voltages and prone to noise on the a chip. Circuits that have really been used are based on this one but were more extensive to deal with the problems.

The basic Gilbert multiplier schematics that is used in the proposed perceptron is shown in figure 4.6. It's composed of three differential pairs, two of which obtain their bias current from the branches of the third. the analysis is simple if one restricts oneslef to the liner region of operation of the diff-pairs. We have established in section 3.5 about the transconductance amplifier, that the difference of the output currents of a differential pair in its linear region can be expressed as a gain g multiplied with the differnce of the input voltages 3.12, where g is proportional to the

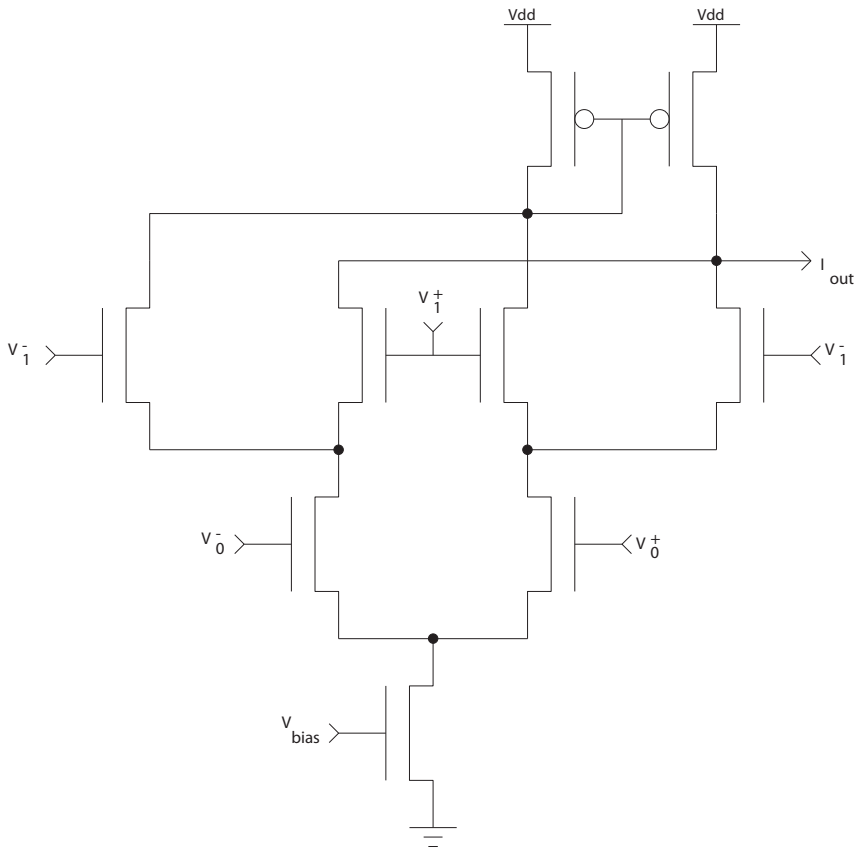


Figure 4.6: The Gilbert multiplier, multiplying differential input voltages, resulting in an output current

bias current I_b. Let' simply assume a proportionality constant C and write that $\mathcal{g} = CI_b$:

$$(I^+ - I^-) = CI_b(V^+ - V^-) \tag{4.1}$$

If we denote the currents in the branches of the bottom diff-pair Gilbert multiplier as I_0^+ and I_0^-, in the top left diff-pair as I_L^+ and I_L^-, and in the top right as I_R^+ and I_R^-, and we note that the bias currents for the top diff-pairs are I_0^+ and I_0^-, then we can put up a set of equations:

$$\begin{aligned}
(I_0^+ - I_0^-) &= CI_b(V_0^+ - V_0^-) \\
(I_L^+ - I_L^-) &= CI_0^-(V_1^+ - V_1^-) \\
(I_R^+ - I_R^-) &= CI_0^+(V_1^+ - V_1^-) \\
I_{\text{out}} &= (I_R^+ - I_R^-) - (I_L^+ - I_L^-) \\
&= C(I_0^+ - I_0^-)(V_1^+ - V_1^-) \\
&= C^2 I_b(V_0^+ - V_0^-)(V_1^+ - V_1^-)
\end{aligned} \tag{4.2}$$

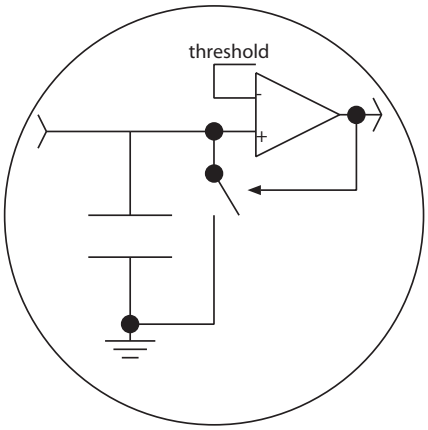


Figure 4.7: Concept of the Integrate-and-Fire neuronal model

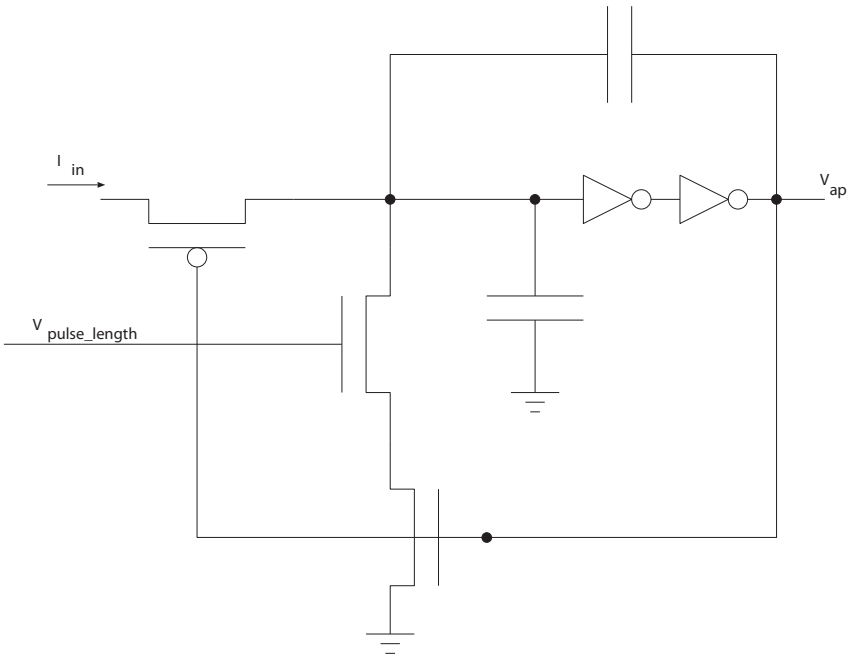


Figure 4.8: A implementation of an Integrate-and-Fire neuron according to C. Mead [27].

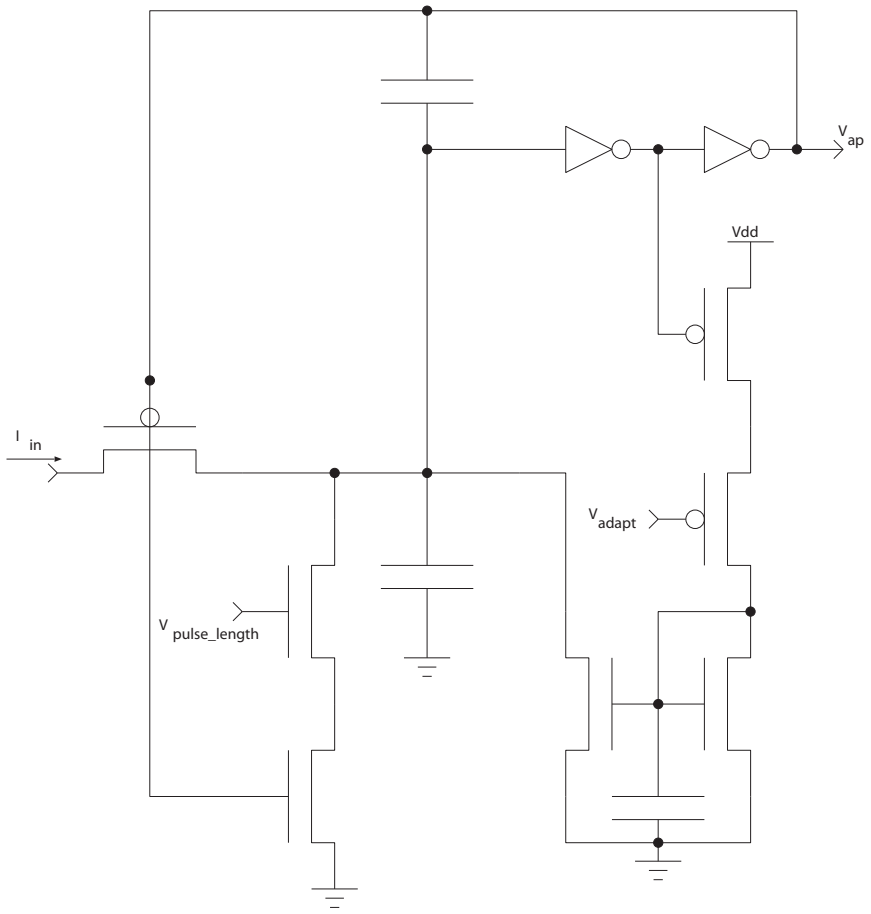


Figure 4.9: A possible implementation of an adaptive Integrate-and-Fire neuron

4.2.3 Integrate and Fire Neurons

This model of a neuron sticks closer to the original in terms of its signals. Its output and its inputs are pulse signals. In terms of frequencies it actually can be modelled by a perceptron and vice versa. It is however much better suited to be implemented in aVLSI. And the spike communication also has distinct advantages in noise robustness. That is also thought to be a reason, why the nervous system uses that kind of communication.

An integrate and fire neuron integrates weighted charge inputs triggered by presynaptic action potentials. If the integrated voltage reaches a threshold, the neuron fires a short output pulse and the integrator is reset. These basic properties are depicted in figure 4.7.

The most popular and very simple and elegant aVLSI implementation of an integrate and fire neuron of figure 4.8 has been proposed by Carver Mead [27]. It consists of a integrating capacitor (representing the membrane capacitance of a neuron), a simple high gain amplifier (double inverter) that switches as the integrated input current exceeds its threshold voltage and a feedback capacitor that adds extra charge to the membrane capacitor

to stabilize the firing state and to avoid oscillations around the switching point. The reset is achieved through a leakage that is turned on while the output is high. That leakage current determines the pulse length of the output pulse. As the switching threshold is reached anew during the reset, the feedback capacitance kicks in again and stabilizes the idle state of the neuron.

Such a neuron can be looked at as an analog-to-analog converter that transforms an analog current into an analog time interval. The pulse frequency will be proportional to the analog input current. One advantage of this representation in the *time domain* is that one can convey analog data with a digital signal, for example for transmission via a digital wireless link.

A neuronal property that is highly valued in some applications is that it attenuates its output when the input is constant. The neuron is said to adapt and is therefore more sensitive to transients (derivative) than to sustained activity. One possible extension of the above integrate and fire neuron to make it adaptive could be the one in figure 4.9.

Again this can be looked at as a analog-to-analog transformation into the time domain. In this instance the pulse frequency will be an approximation of the derivative of the input current.

4.2.4 Compartemental Neuronal Models (Silicon Neurons)

So called 'silicon neurons' [28, 29] take the analogy to real neurons even further. They take into account that a neuron is not uniform and has varying properties and states along its dendrites and the axon. Signals internal to the neuron do not spread instantly. These facts are modelled in 'silicon neurons' by assembling the cell out of connected compartments. Within one compartement different ionic currents that flow through ion-specific channels in real neurons are represented. These channels open or close dependent on voltage or concentrations (represented as voltages) of other ions or chemicals. Action potentials are not mere digital pulses but resemble more closely the real ones with sharp voltage increas as the membrane reaches threshold and undershooting as the action potential is turned of.

Such 'compartemental models' are also popular in software, but only aVLSI implementations make real time simulations possible.

On that level of detail the behaviour of neurons has been described as electric circuits from the start. Maybe the most popular and certainly the earliest example of this is the model of the spike generating circuit in the axon by Hudgkin and Huxley [19, 30] (figure 4.11) based on experiments with the 'giant squid axon'. Basic elements are slow working ion pumps that are represented as voltage sources, i.e. potential differences in the model. They maintain an imbalance of ion-concentration inside and outside of the cell accross the cell membrane. The voltage of the voltage source labelled with 'resting potential' is the neteffect from all ion pumps combined whereas the 'reversal potentials' would be the resulting voltages if the concentration difference of the particular Ion type that labels the voltage source would be levelled to zero. In other words it would be the voltage resulting when turning off that particular ion pump. So the actual force that

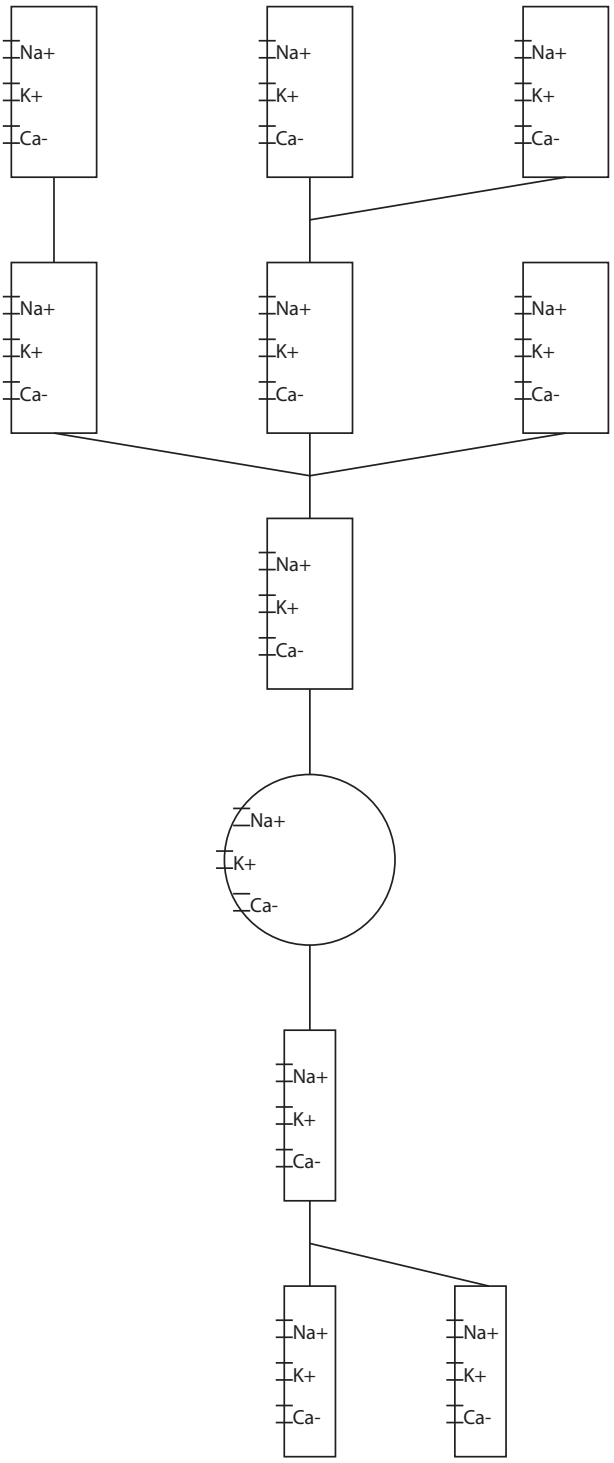


Figure 4.10: Concept of a compartemental model of a neuron, its dendrites and its axon.

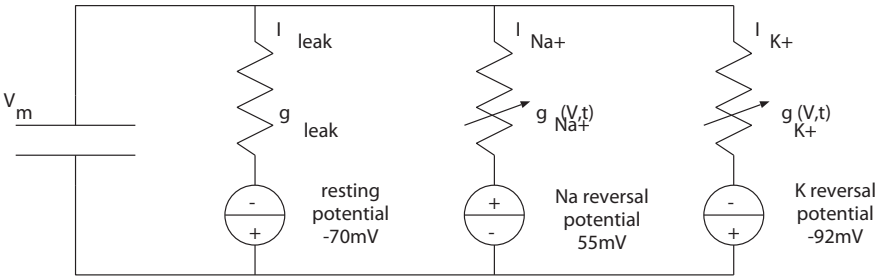


Figure 4.11: The Hudgkin Huxley model of action potential production, i.e. the active channel properties in the axon

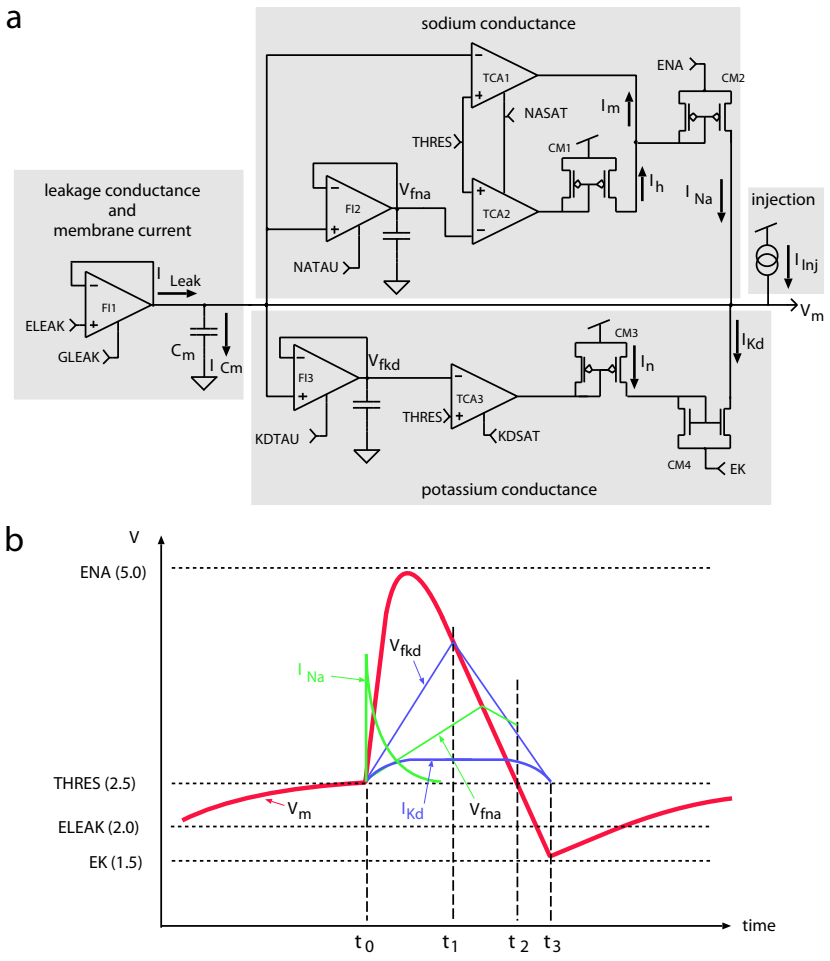


Figure 4.12: A CMOS Implementation of a HH-soma, a) schematics, b) typical signal time course

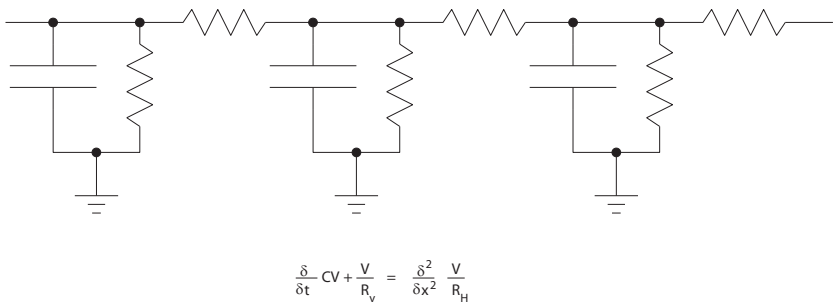


Figure 4.13: Model of a cable

fuels these voltage sources is diffusion accross the cell membrane mediated by the permitivity of the cell membrane for the ions. Ion channels that change that permitivity are thus represented as resistors/conductances, one constant g_{leak} representing the idle permitivity of the membrane and the others that change in dpendance of voltage as variable resistors.

More advanced models may contain a goodly number of other conductances, but in this basic first issue there are but the most prominent conductances responsible for action potential generation. g_{Na+} and g_{K+} are both opening up as the membrane voltage crosses a certain threshold (e.g. due to external synaptic input) but with different delay: g_{Na+} opens up more quickly and thus initiates the action potential. g_{K+} opens up more slowly, finally exceeding g_{Na+} and thus terminates the action potential after some delay. g_{Na+} diminishes again immediately as the membrane voltage falls below the threshold and again g_{K+} reacts more slowly, causing an 'overshoot' of the membrane voltage below the resting potential, which is only slowly reestablished again by the ion-pumps.

This same model has been approximated in CMOS aVLSI (a variant of a 'silicon neuron') like depicted in figure 4.12 [29]. The leakage conductance is modelled with a follower that expresses quite similar properties to a resistor as long as it remains in its linear range, i.e. as long as the voøtage difference accross it is not very big.

The potassium conductance is activated after the membrane voltage V_m crosses the threshold $THRESH$. It's acticvation delay is achieved by a RC-type delay filter/integrator circuit (amplifier FI3 operating in its linear region and the capacitive load at its output) acting as a delay element. Thus, V_{fkd} is a delayed version of V_m and it activates the potassium current at the output of the transconductance amplifier TCA3. Before it is applied to the membrane capacitance C_m it is rectified and sign corrected by two current mirrors. (Without rectification there would be a positive potassium current while the membrane is below threshold.)

The sodium current is implemented using the same building blocks. It is turned on without delay by TCA1, the output of which is also rectified before it is applied to C_m, and then explicitly turned off again (contrary to the original Hudgkin and Huxley model) after some delay (determined by FI2 and its output capacitor) by TCA2. 'Turning off' is achieved by

subtracting the rectified output current of TCA2 (I_h) from the output current of TCA1 (I_m).

The connection between compartments is merly resistive. The passive behaviour of a chain of such compartments can be modelled by a cable equation (figure 4.13):

$$\frac{d}{dt}CV + \frac{V}{R_V} = \frac{d^2}{dx^2}\frac{V}{R_H} \tag{4.3}$$

If one node voltage in an infinite cable is caused to change, the neighbours will follow the change with a delay and with attenuated amplitude and speed. The effects of more complicated stimulations and within branching cables (dendritic or axonal trees) is more difficult to analyze and is therfore usually observed in numerical simulations or in a hardware model.

It gets more complicated if the compartments are active elements, e.g. the axon is modelled as a chain of Hudgkin and Huxley type compartements where the action potential is actively amplified in each. This reflects biology where active sodium (Na+) and potassium (K+) channels can be found all along the axons, comparable to relay stations amplifying the action potential along the way. Be aware that the switching of the relay stations is a chemical process and subject to a certain delay, which makes the action potential traveling down the axon relatively slow as compared to a purely electrical signal. Thus, the action potential in a cortical axon travels only at about 1m/s. Peripheral nerves consist of myelinated axons that increase the speed up to in the order of 100m/s. Myelinated axons are surrounded by a myelin sheet with short unsheeted segments (Ranvier nodes) every now and then. The effect of the myelin sheet is a heavily reduced membrane capacitance and leakage conductance, and a blockage of ion channels. the AP travels thus purely electrically and quite quickly along the myelinated stretches of the axon before it is amplified again at the Ranvier nodes.

Chapter 5

Coding in the Nervous System

5.1 The action potential

The action potential (AP, a voltage pulse, spike) is the main form of information transmission in the nervous system. But also analog electrical signals are sometimes used and a lot of chemical processes are involved, for example at the synapse, in conveying the AP to the postsynaptic cell, and in initiating changes in the synapse that result in learning. Especially the exact workings of the later are still unknown. APs are considered to be digital in amplitude (ca. -70mV, 30mV in the mammal brain), and fixed in duration (ca. 1ms). They are issued by most neurons with a minimal inter-spike interval of ca. 5ms.

How information is encoded with those spikes in the huge network that is the nervous system is not yet completely known. In the periphery of the nervous system we do have some knowledge however: We know a lot about how to stimulate optical, auditory and touch sensor-cells and about their responses. We also know some about how motor neurons stimulate muscles. But the waters get more murky in the 'deeper' parts of the nervous system.

The interesting questions in this context are 'Do we know how to reconstruct a stimulus from observing the activity in the nervous system?' and 'Do we know how to predict an action from observing activity in the nervous system?'. As mentioned above, the answer to those questions is yes, if we observe the peripheries of the nervous system, and if the stimuli and actions are simple (like a single flashing light point at a particular location in the visual field or the contraction of a single muscle). But for more complex stimuli (seeing the face of someone we know or experiencing a sunny day in autumn in the forest, with all the rich information that such an experience comprises) and actions (pushing the green button or going down to Giusepe's to buy a pizza) the answer to those questions is rapidly approaching 'absolutely not'.

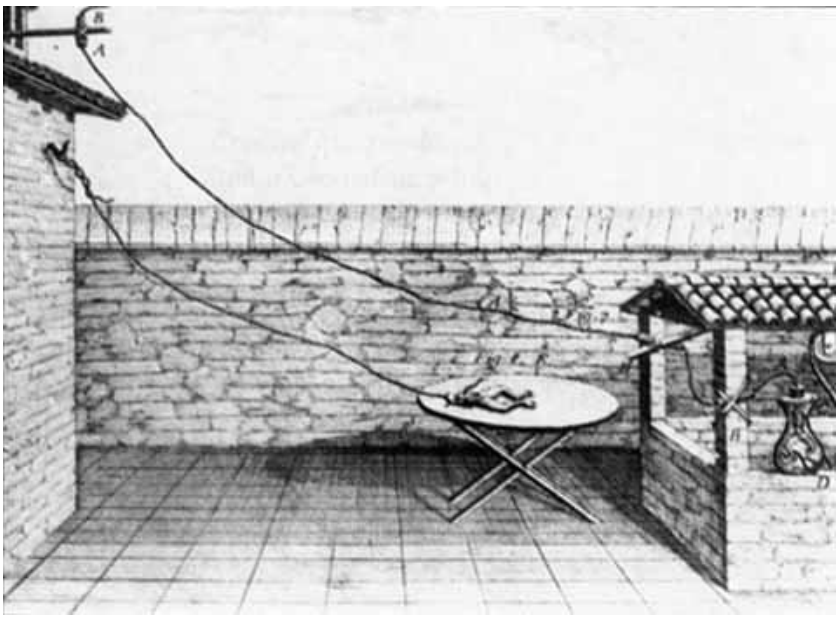


Figure 5.1: Muscle stimulation in frog legs by static charge as reported by Luigi Galvani in 1780

5.2 Hints in Experiments

5.2.1 Classical experiments based on observing spike rates

Already for a long time researchers have tried to gather knowledge about the connection of brain activity and sensing and acting. They have been trying to find correlations between any kind of activity in the nervous system and stimuli or motor responses.

Initially, the main observable has been average neuronal firing rates/indexfiring rates. It has been observed that this mean firing rate varies substantially over time and correlations between these rates and specific stimuly or muscle activity have quickly been found. here come a few examples of experiments that have established such correlations.

Motorneurons/Muscle stimulation The first documented artificial stimulation of muscles has been conducted by Luigi Galvani as documented in 1780. By touching the muscles in a frog leg with a metal, it could be made to twitch. Thus was electricity discovered and for obvious reasons believed to be somehow linked with lifeforce, which is probably the reason why Frankenstein's monster in the famous novel is brought to life by lightning some 38 years later.

Much later, the source of the electricity in the intact body has been located in motor cortex. Muscle fibers react with a contracting force to an action-potential from a motorneuron. Steady force can be achieved by sending spikes with a steady frequency to the fiber.

Simple and Complex Cells in Visual Cortex A real classic finding. Hubel and Wiesel were the first to document correlations between firing rates of cortical neurons in visual cortex and bar-stimuli that are characterized by an orientation (simple cells) and sometimes a direction of motion (complex cells). An experiment that measured

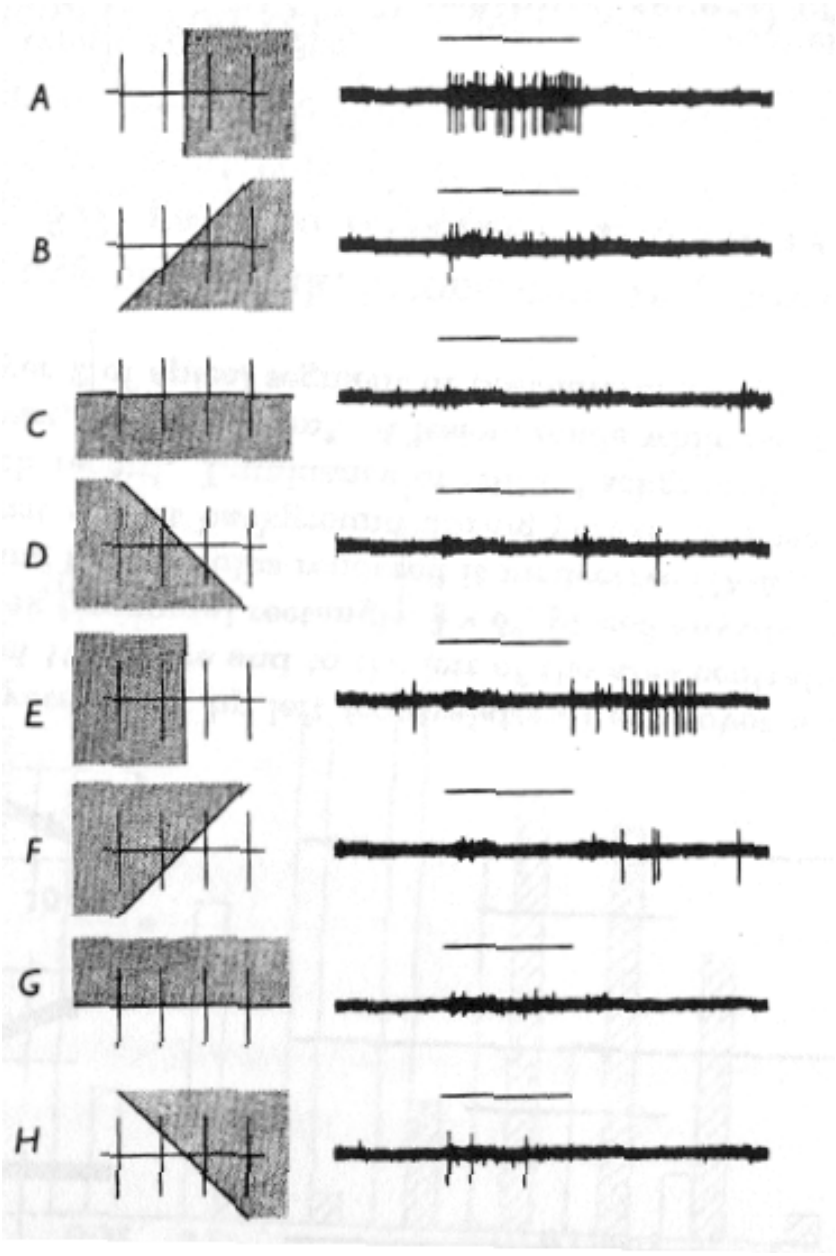


Figure 5.2: Orientation selective cells in cat visual cortex, stimuli and responses of one cell [31]

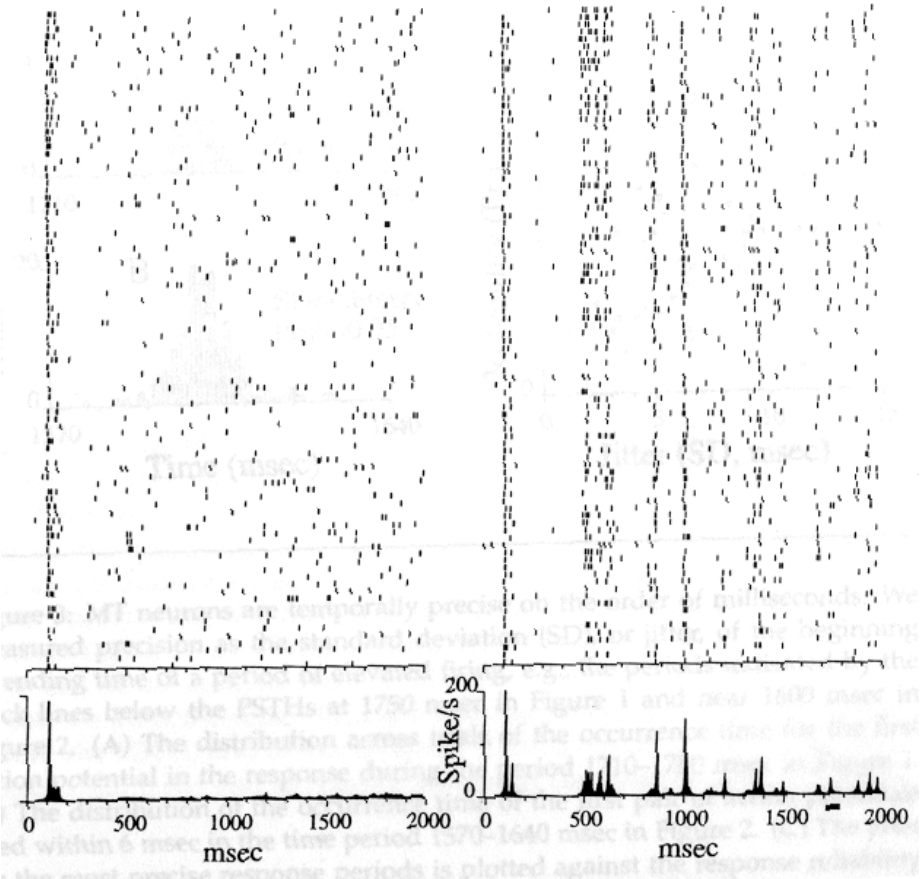


Figure 5.3: Neuron response to a moving random dot pattern, when repeatedly presented the very same random dot pattern (right) and different random dot patterns (left) [33]

orientation preference in a cortical cell is shown in figure 5.2). Since then many people have documented 'receptive fields' and 'optimal stimuli' for cells in visual cortex, by looking for maximal response in terms of firing rate.

Segmentation by synchronization Regular oscillations in population activity in neurons is a widely spread phenomenon in the brain. Some researchers suggest (e.g. by Milner, von der Malsburg, and then Singer[32]) that observed features that are represented by oscillatory activity of particular populations will synchronize the oscillations for features that belong to the same object. Dividing a scene into regions that belong to different objects is known as 'scene segmentation'.

5.2.2 Classical Experiments observing temporal spike codes

There are however experiments that suggest that there are more details about some stimuli encoded in the precise firing patterns of individual neurons. A provocating thought along this line is that it is not necessary to change the average activity of a neuron to transmit information. So maybe

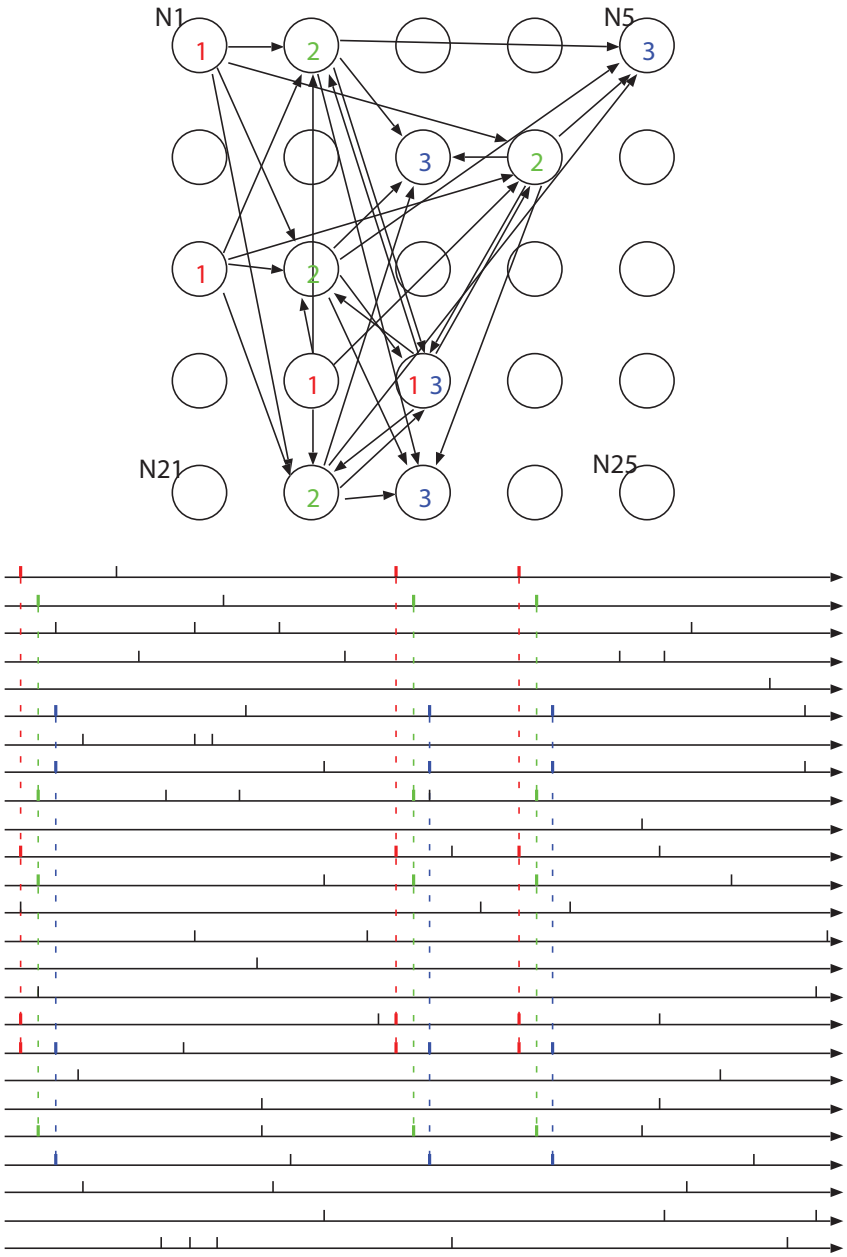


Figure 5.4: The concept of Synfire Chains originally proposed by M. Abeles [34]

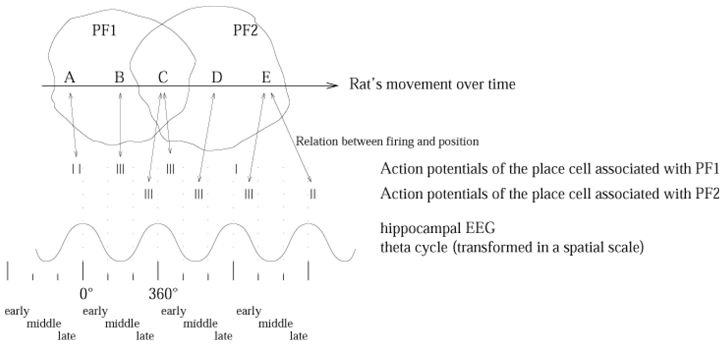


Figure 5.5: Phase relation of place cells' activity to synchronous background activity indicating more detailed position within the receptive field of the cell [35]

some of the information about stimuli is encoded in places where no-one did bother to look yet.

Highly consistent firing patterns for high entropy stimuli One particular very interesting discovery was made by accident by Bair and Koch[33], so the rumors say. During a recording from a neuron (in the MT cortical area) a lab-animal was repeatedly presented with a moving random dot pattern. Surprisingly the cell responded very consistently with almost the same spike train. The scatter plot the right of figure 5.3 shows these repeated trials along the y-axis and the histogram below shows the summed spikes accross all those trials. It turned out that due to a programming bug, the random pattern was always created with the same random-generator-seed and thus it was the exact same pattern in every trial. (When the bug had been corrected, the result looked as originally expected as shown on the left hand side of the figure.) So it seems that if the stimulus has high entropy (in the information theoretical sense) a lot of detailed information is contained in a single cell's spike pattern. And it also seems that a neuron is quite an exact device. By contrast, if the stimuli are simpler, e.g. moving bars, then the detailed pattern looks random and different from trial to trial. Maybe such 'simple' stimuli need not be represented by the full capacity of the neuron.

Synfire Chains The research group of Abeles noticed a strange thing when recording from multiple units in the cortex[34]. A specific while after a stimulus had been presented, a particular group of neurons consistently tended to fire an AP simultaneously. Other than that the firing patterns looked more or less random. The researchers explained that phenomenon by a 'synchrony code' model known as synfire chain (figure 5.4). They propose that groups of neurons fire a spike in synchrony (triggered by some kind of meaningful information event) and that this synchronous activity propagates from group to group, establishing a temporal relation to the triggering event. One neuron may participate in several of these groups. Thus, observing only one neuron, this will very much look like Poisson distributed random activity.

Place Cells in Hippocampus Place cells are cells in Hippocampus, often observed in Rats[35]. They become active if the rat is in a particular place within an environment. This place can be reconstructed by looking at the rate of the neuron. But it has turned out that by looking at the temporal firing pattern of the cell in relation to the entire population theta rhythm activity (observed by EEG) one can even reconstruct the rat's position within that place more precisely (figure 5.5). That more detailed position information is encoded in the phase relationship between the cells burst activity and the population theta rhythm. (theta rhythm: 5-8Hz)

Spike Based Learning Experiments by: Larry Abbot etal. or Henry Markram etal. [22]

Larry Abbot etal. and Henry Markram etal. [22] conducted experiments that do not directly allow to make a connection between a stimulus and a temporal firing pattern, but that show the interest of the nervous system in particular temporal patterns. Their experiments revealed changes in synaptic strength caused by particular firing sequences. This has later become popularly known as spike timing dependent plasticity (STDP) and is explained in more detail in chapter 10 about learning. The bottomline is that synaptic strength changes in dependency, not only of average firing rates of the presynaptic and postsynaptic neuron as the earlier LTP/LTD experiments have shown, but more specifically in dependency of the accurate relative timing of individual spikes of the pre- and postsynaptic neuron. This clearly indicates that this timing carries relevant information that substantially changes the neuron's behaviour.

Reaction time in object recognition The group of Simon Thorpe conducted experiments that are an intriguingly simple, but still a convincing argument for the relevance of temporal encoding in the nervous system[1, 2]. Monkeys and human test subjects were presented pictures and had to decide if there was food (or an animal in another series of experiments) in the picture or not. they had to push one of two buttons accordingly. The reaction time for this task was about 200ms-500ms. The experimenters argue that the neurons along the processing path through the nervous system, from the optical input to the motor reaction, do only have time to fire one to two spikes during that time. So there is no time to compute a temporal average. The information is conveyed with only a few spikes per neuron. Candidate encoding schemes that could provide such fast reaction times are population- or purely temporal codes.

5.3 Candidate Codes

Based on all of these experiments theoreticians propose many different ways of extracting information from neuronal activity or to encode information in artificial neural systems. These encoding schemes can be divided into several, not mutually exclusive classes. All of them look at patterns of neuronal activity over time and space, some of them look at average activity (over time (and space): rate coding; over space population coding) some of them rather at spike patterns (over time (and space):

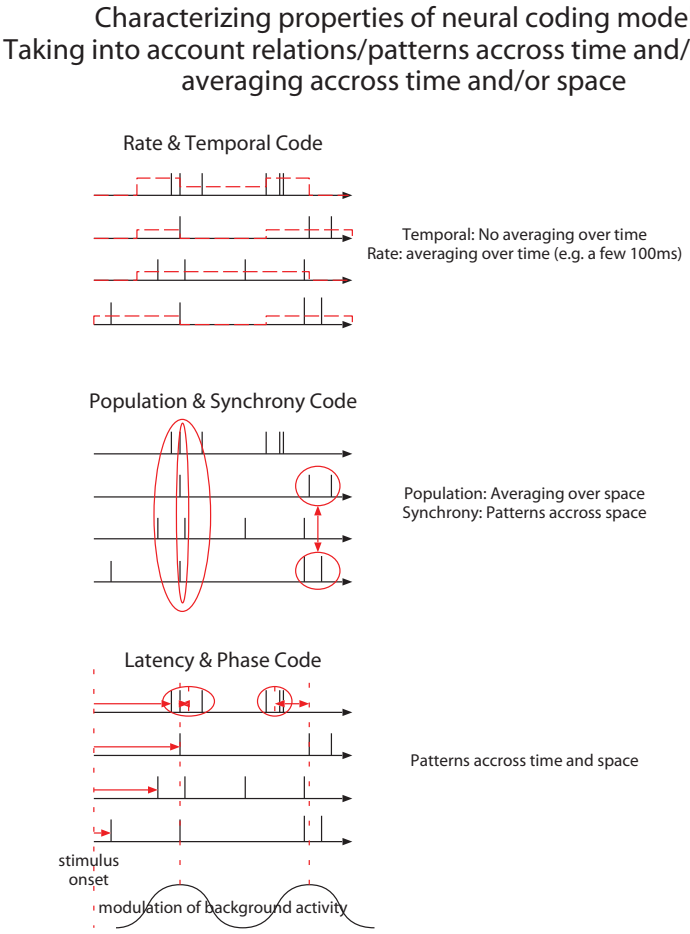


Figure 5.6: AN illustration of the coding schemes mentioned in the text

temporal coding; over space: synchrony coding). Figure 5.6 illustrates some of these coding concepts.

Some properties of some of these coding schemes are listed in the following:

Rate Codes Information (about stimulus or action or thoughts) contained in: Spike activity averaged over time. Properties: Robust against noise, good spatial resolution, but slow

Population Codes Information contained in: Spike activity averaged over space Properties: Robust against noise, fast, but worse spatial resolution

Temporal Codes Information contained in: Spatio-temporal spike patterns Properties: Fast with a good spatial resolution but sensitive to noise

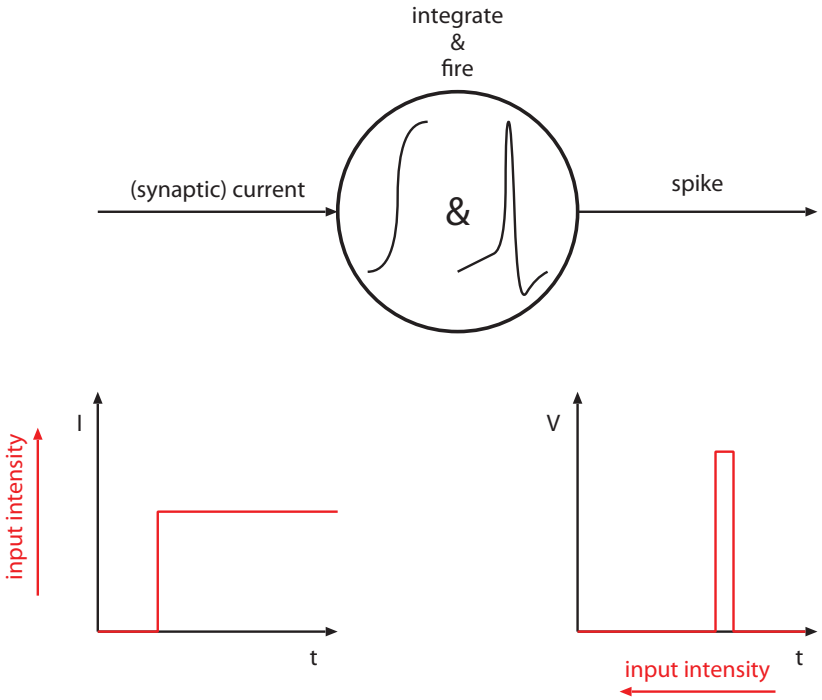


Figure 5.7: One popular temporal coding scheme that looks at spatio temporal patterns in a group of neurons is latency encoding, or rank-order encoding. It looks at the first spike of each neuron after stimulus onset. An I&F neuron naturally encodes the inverse of stimulus strength in the latency of this first spike.

Synchrony Codes Information contained in: Spatial activity patterns Properties: Robust against noise, with a good spatial resolution but slow

Typical concrete examples of temporal codes are the latency code or time to first spike code (see figure 5.7)

The phase code scheme proposed for the place cells in rats is somewhat trickier to fit into the four mentioned categories: it relates precise firing patterns (temporal codes) to population/global activity oscillations. I would thus tend to call it a temporal code.

Chapter 6

Neuromorphic Communication: the AER Protocol

The way of communicating information within the nervous system is rather different from the methods used in most artificial electronic systems. Most neurons communicate by way of nerve pulses (action potentials) via dedicated point-to-point connections (axons). On the other hand the communication channels between computers or inside computers (such as busses) transmit more complex signals at higher rates than an axon. The physical channels are mostly shared and time multiplexed by several components and their density can therefore be kept lower.

As neuromorphic engineers try to implement systems that are organized more like the nervous system, the brain's way of communicating internal signals becomes a major obstacle to electronics. The sheer numbers are staggering: The human brain contains about 10^{11} neurons each of which has 1000 to 10000 synapses. All those synapses receive input via a dedicated output line of the sending neuron. The so called white matter underneath the folded surface of the cortex (the most recently developed part of the brain in evolution, believed to hold conscious functions) is densely packed with such connections between cortical areas and other brain parts. It holds much more volume than the gray matter, the surface layer of the cortex which holds the neurons. The most progressed digital computer chips cannot by far provide the same density of wires. That is mainly because computer chips are confined to two dimensions in their layout whereas the nervous system makes full use of all the volume it occupies.

6.1 The Basic Idea of Address Event Representation (AER)

The most prominent idea of how to come closer to the capacity of point-to-point in the nervous system is the one of address event representation (AER) [36, 37, 38, 39]. It uses the advantage of speed in digital systems to make up for the disadvantage in density and emulates point-to-point connections rather than physically implement them.

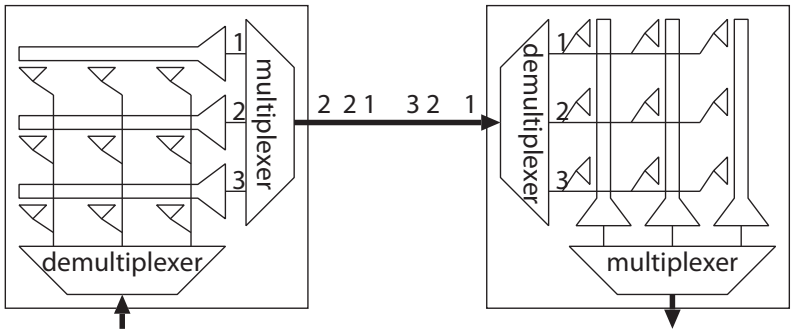


Figure 6.1: The basic idea of Address Event Representation (AER)

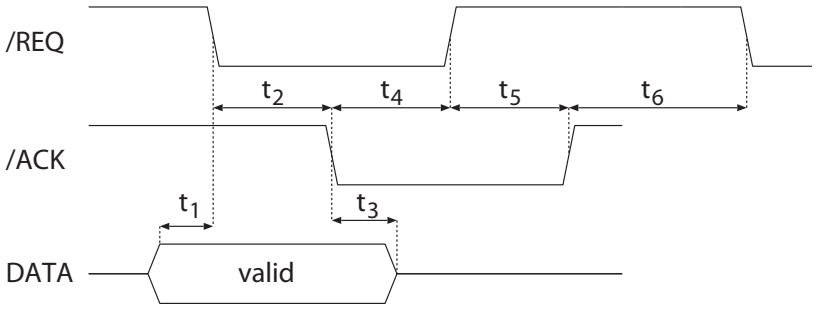


Figure 6.2: Asynchronous control signals for a four phase handshake communication of one data package

The principle is intriguingly simple: Each neuron in a network is assigned an address. When a neuron wishes to send a nerve pulse (action potential) it places its address on a digital bus via an encoder. Synapses that are supposed to receive that action potential are connected to the same bus via a decoder and get stimulated when their address occurs. The bus can unfortunately only transmit APs serially but it can do it much faster than an axon. It can transmit APs so tightly spaced in time that for a network of neurons that operate on a natural time-scale these pulses are virtually simultaneous.

6.2 Collision Handling

Since multiple point-to-point connections share the same bus there is need for a bus control mechanism or a collision handling. Different research groups have developed different approaches to that problem. The main principles are:

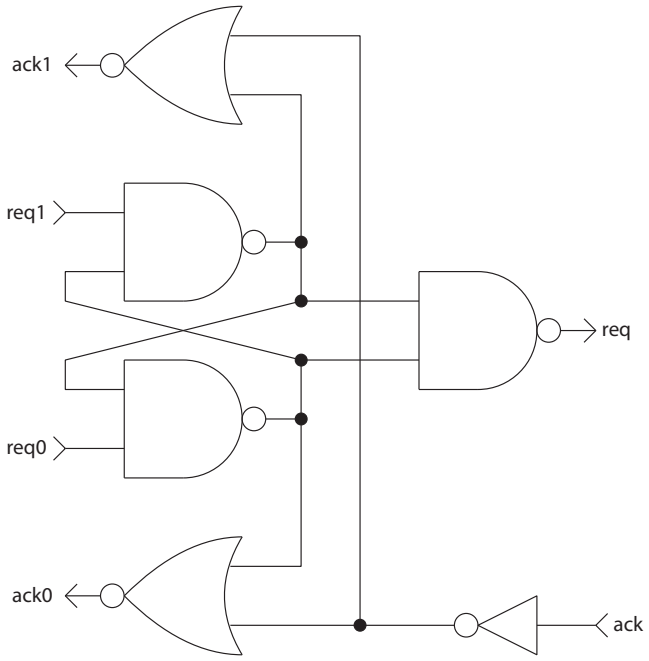


Figure 6.3: A two-input 'greedy' arbiter

- full arbitration (The Caltech solution [36, 39] and WTA arbitration [40])
- discarding (The Swiss solution [37])
- aging versus loss trade off (The Oslo solution [41])

6.2.1 Full Arbitration

In full arbitration [36, 39] the workload lies with the sender. Collisions are resolved before sending an APs address via the bus. Neurons request bus access from an arbiter if they want to transmit a spike. The arbiter can for example be implemented as a binary tree of elements that handle two input requests and issue a request of their own to the next layer in the tree-structure. Such a binary arbiter cell can be implemented like shown in figure 6.3:

The two coupled NAND gates can be looked at as a RS flipflop with active low inputs. Normally if there is no request active this RS flipflop is in its 'illegal' state when R and S are set. Both outputs will be forced to 1. Which state the flipflop will fall into is determined by which of the two inputs, S or R, is withdrawn first. Or in other words which active high request ('req0' or 'req1') comes in first. If req0 is the winner, then the output of the lower NAND gate will become 0 and the signal 'req' will be propagated to the next layer of the tree. If that 'req' is granted by an incoming 'ack', 'ack0' is set. Following a four phase handshake protocol the requesting instance will thereupon withdraw its 'req0'. If in the meantime 'req1' has been set, that request gets also granted without releasing 'req'. That principle of serving

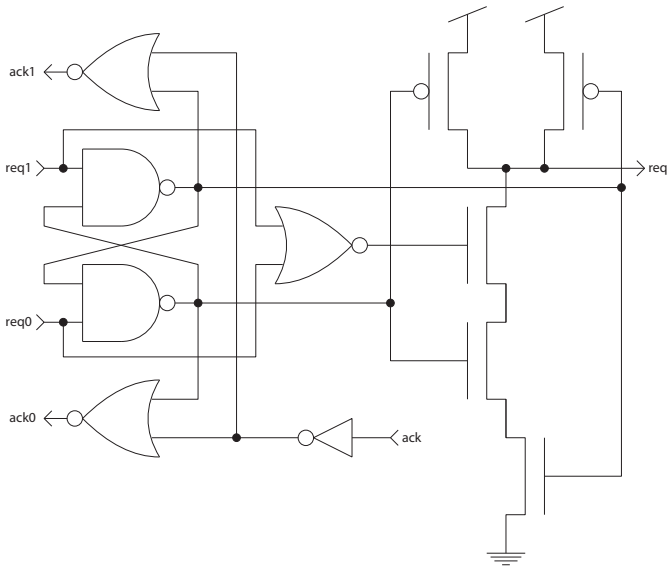


Figure 6.4: A glitch free two-input 'greedy' arbiter

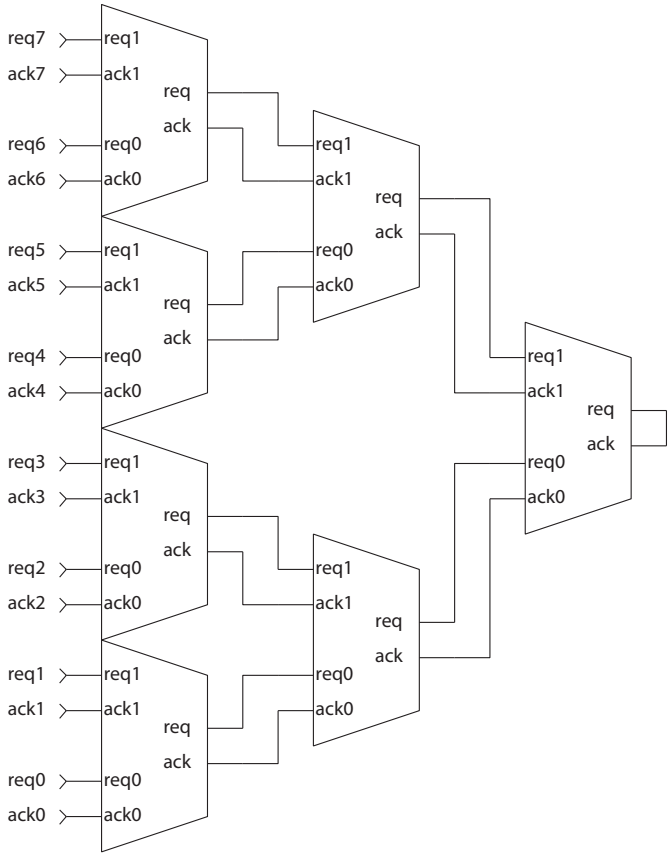


Figure 6.5: A binary tree of 'greedy' arbiters

all local requests first before giving control back to the higher level in the hierarchy is called greedy arbitration. Only if there are no more request from the left ('req0' and 'req1') is 'req' withdrawn and other arbiter cells on the same level get a chance of being acknowledged.

The solution in figure 6.3 is not safe tough: there is a danger of a glitch (a very short unintentional pulse) at the output request if there are two input requests waiting ('req0' and 'req1' are high) and the first of those requests (let's say 'req0') is withdrawn after 'ack' goes high. In this case there is the possibility of a glitch of 'req': As 'req0' is withdrawn, the RS-flipflop formed by the two NAND gates flips from (0,1) to (1,0). But the 1 needs some time to travel from the output of the lower NAND to the input of the upper NAND. Therefore, if we look at this very closely with high temporal resolution, what actually happens is (1,0)→(1,1)→(0,1), and while the state is (1,1), 'req' may very briefly switch to 0.

This is corrected with the circuit in figure 6.4. Here, the NAND with 'req' at its output is replaced with something very similar to a NAND. There is an extra input, however. It controlls a NFET that prevents 'req' from being pulled towards 0 in this glitch-situation. It only allows 'req' to be pulled low, in the normal situation where both input requests are low. In the transitory state where one input is still high and both outputs from the flipflop are high, 'req' is left momentaryly floating/tri-stated. That means it will maintain its state for a short while, which is long enough to bridge that transitory state. No glitch will occure.

These arbiter cells can be assembled into a binary tree (figure 6.5). The top most 'ack' and 'req' need to be shorted together. Thus, the arbiter can be extended from two inputs to any number of inputs.

6.2.1.1 pro

- fast

  Since this method of arbitration is among the fastest methods it can reach a much faster throughput than other methods before the probability of collisions becomes significant.

- No loss of APs

  In case of a small number of collisions, no spikes will be lost. Only in the case of heavy overload on the bus it might happen that neurons are delayed for so long that they would like to spike again before the first AP could be transmitted. In that case the second spike is lost.

6.2.1.2 contra

If the load of request begins to exceed the throughput of the bus, some non idealities start to show their influence.

- unfair arbitration

  Because of the greedy arbitration neighbours in the binary tree structure of the transmitting neuron get priority over more distant ones. That can lead to a total monopolizing of sub regions of neurons of the communication resources in case of continuous collisions. But

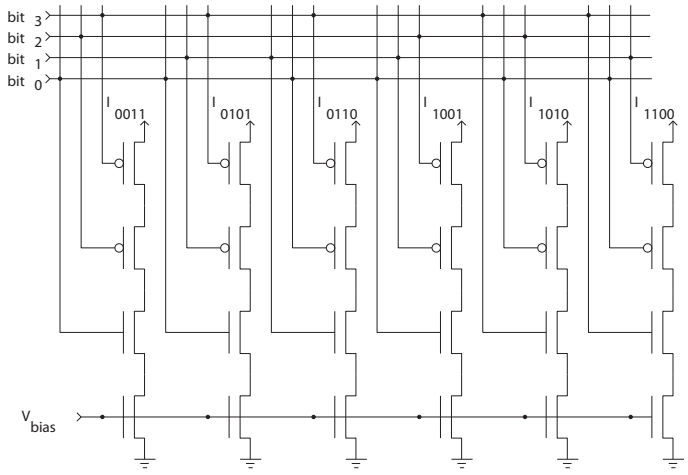


Figure 6.6: An AER receiver that discards addresses that are rendered invalid by a collision

one might say that in that case the heavy overload on the bus renders the transmitted data meaningless anyway.

- uncontrolled delays

  In applications where a high temporal precision is needed already the jitter in the spike timing caused by a few collisions might be a problem.

Another variant of full arbitration is WTA arbitration [40]. This variant is somewhat slower, though, than the greedy arbitration. And dependent on the actual implementation, it might even be more unfair than the greedy/binary-tree arbitration (e.g. if there is a default winner due to circuit mismatch). The 'aging versus loss' concept described later in this chapter also uses a WTA arbitration and does actually achieve fairer arbitration.

6.2.2 Collision Discarding

This strategy [37] resolves collisions on the receiver side. Any sender can anytime pull the high bits of its address up. There is some redundancy in the address codes such that collisions can be detected. One coding that has been proposed was to only use codes with a fixed number of 1s. It has been claimed that with a code length of n bits and allowing only codes of n/2 1s one needs only two bits more than a normal binary number encoding, for $n < 20$. In other words one needs n bits to encode 2^{n-2} addresses. Events on the receiver side can be decoded as depicted in figure 6.6

The addresses are decoded by their zeros. Only one of the ones needs to be monitored to detect an event. It is assumed that the output currents get integrated and that a pulse is much longer than any glitches that might occur due to switching. In that way only an intentional pulse contributes significantly to the integrated current.

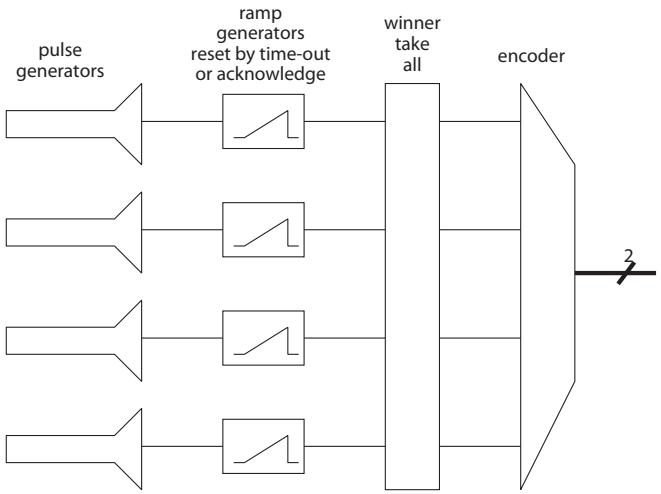


Figure 6.7: The principle of 'Aging versus Loss trade Off' AER communication

6.2.2.1 pro

- easy circuitry (e.g. no handshake)
- no distortion in the timing of transmitted pulses

6.2.2.2 contra

- loss of pulses
- pulses need certain width
- specific requirements of the receiver

6.2.3 Aging versus Loss Trade Off

This concept [41] is a compromise between the other two. The idea is that full arbitration is a fine thing usually, but if one starts to delay pulses too much, one is better off discarding them. In addition the suggested implementation offers fair arbitration.

The principle is that of a fifo queue with a time out. The implementation is depicted in the blockdiagram in figure 6.7: A spike triggers the linear rise of a voltage. A WTA circuit determines the highest voltage among those ramped up voltages of all neurons. The neuron with the highest voltage is granted bus access and the voltage is reset to its idle state. Thus, something similar to a FIFO is implemented. If the ramped voltage reaches a threshold without getting access to the bus, it is reset anyway. Like this, pulse transmissions that are delayed for too long, get discarded.

6.2.3.1 pro

- possibility of trading loss versus maximal delays

6.2.3.2 contra

- slow
- extensive circuitry

Chapter 7

Photo Receptors in CMOS Technology

7.1 Introduction

One of the most successful fields of the neuromorphic engineering is neuromorphic vision. This field studies how to emulate biological retinaes using electronic devices. To understand their principles of operation, it is necessary to know the photo transduction process in silicon, i.e. how photons can be detected and transduced into electrical signals. In this chapter, we will review the fundamental of light detection in silicon. We will also present basic photo receptors used in CMOS technology. In the next one, we will explain how to use this receptors to build retinomorphic circuits that behave as the human retina.

7.2 Fundamentals of Photo Detectors Operation

The transduction of light into electrical signals in a semiconductor is a complex process that depends on may factors: quantum efficiency, wavelength of the incident light, doping concentration of the semiconductor, etc. The electrical signals are electrons that 'jump' from the valence band to the conductance band excited by photons absorbed by the detector. A photon is an elementary particle that represents the basic unit of light or electromagnetic radiation. Not do all photons can excite electrons from the valence band to the conductance band. Only those whose energy E_{ph} exceed a certain threshold (semiconductor bandgap E_g) can create them. This imposes conditions for the wavelength detection depending on the semiconductor material. The energy of one photon can be described by:

$$E_{ph} = \frac{h \cdot c}{\lambda} \tag{7.1}$$

where h is the Planck constant, c is the speed of the light and λ is the light wavelength. Thus,

$$\lambda_c = \frac{h \cdot c}{E_g} = \frac{1.24}{E_g(eV)}[\mu m] \tag{7.2}$$

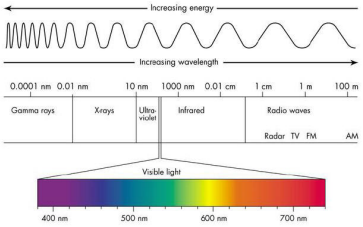


Figure 7.1: Electromagnetic Spectrum

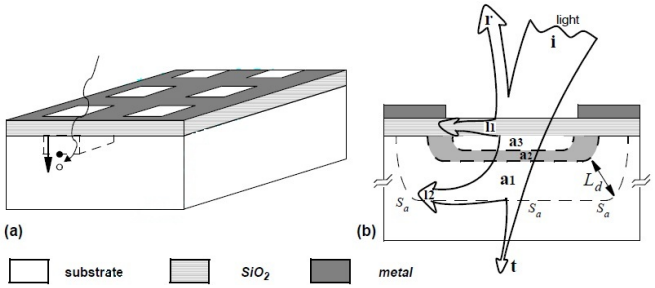


Figure 7.2: Layout of a 2-D pixel array and cross section illustrating the light absorption process.

If $\lambda > \lambda_c$, the energy of the incident photon will not be enough to create one electron and light will pass through the semiconductor. In the case of silicon, $E_g = 1.12eV$ and $\lambda_c = 1.11\mu m$. So typically, silicon conductors can detect light within the visual spectra (See Fig. 7.1) and even are capable of detecting early infrared light.

Unfortunately, the number of electrons that incide in the silicon is not the same than the number of excited electrons created by then. This number is always lower and depends on several parameters. For this reason, we define a figure of merit called *quantum efficiency* which is the number of electrons per incident photon at a given wavelength. Quantum efficiency is determined by several parameters. Just to mention a few: the absorption coefficient of the semiconductor, the semiconductor's overlaying material, the recombination life time and the diffusion length. So we can state that the quantum efficiency has a strong dependence with technology. For this reason, there are dedicated fabrication processes optimized for light detection. The depletion region is a spatial region between two silicon regions with different doping concentrations. In this region there are electrics fields that attract the electrons created by the incident photons. Quantum efficiency will be determined by the efficiency of charge collection in this region. In the next sections, we will discuss briefly the influence of some of these factors on the light transduction into photo current.

7.3 Overlaying Material

Fig. 7.2 shows how light can penetrate through a the photo sensing/processing part of a chip. Chips are usually covered by an opaque mask to prevent it of their formation of parasitic photo diodes that could inject noise (photo current) and alter the chip performance. The parts that have to sense light are uncovered as is shown in Fig. 7.2. However, not can

all the incident light be sensed. Part of it, is reflected in the oxide/air interface (r in Fig. 7.2). Typically, some anti reflective layers are used on the top of the detector's area to reduce the amount of light that is reflected back. Another part is absorbed before arriving to the semiconductor due to the imperfect transparency of the silicon oxide and/or other cover layers (*I1* in Fig. 7.2). And always some amount of light will pass through the silicon without generating electron-hole pairs (t in Fig. 7.2). Furthermore, some CMOS processes provide silice layers on the drain/source regions and gates to increase its resistance. They also use to create resistances. However, this silicide material is highly opaque to visible light. Some precesses offer the possibility of removing these layers over certain regions of the chip to improve sensitivity to light. Companies usually have dedicated foundries and tailored processes to fabricate photonics devices. Nowadays, new trends in CMOS technology are *backthining*: the chip is illuminated from the bottom and not from the top, or 3D processes: several dies are stacked in 3 dimensions. One specific layer can be dedicated just to sense light. Thus, high spatial resolution and sensitivity can be achieved.

7.4 Light Absorption in Silicon

To measure how fast light (packets of photons) is absorbed in the semiconductor, we define the absorption coefficient. It describes how far a foton can travel through the semiconductor without being absorbed, creating an excited electron. If ρ_0 is the incident power per unit area of monochromatic light incident on the semiconductor surface, then the available power at a depth x from the surface is given by

$$\rho(x) = \rho_0 \cdot e^{-\alpha x} \tag{7.3}$$

Parameter α depends on photon wavelength. Shorter wavelengths are absorbed close to the surface and longer wavelengths are absorbed deeper. Based on this property, color detection is possible stacking photo diodes at different depths [42]. For deep blue light ($400nm$), α in silicon is $50,000cm^{-1}$ and is absorbed at average depth of $3,3\mu m$ from the surface. For red light ($650nm$), $\alpha = 3000cm^{-1}$ and is absorbed at an averaged depth of $3,3\mu m$. If the absorption coefficient is too high or too low, the excited electron could be created in one region far away of the depletion region and could not be collected. This effect can be seen on the right of Fig. 7.2 represented as *I2*. The light that has been absorbed at different depths has been represented with the letters *a1, a2*, and *a3*.

7.5 Recombination Lifetime

Photon-induced electrons have a finite lifetime in which they are mobile within the silicon substrate before returning to the valence band [43]. This time constant is called the recombination lifetime τ_r and depends on the quality of the silicon and its dopant density. Longer lifetime increases the probability of collection and therefore the quantum efficiency. Recombination can occur in several ways. Direct transitions between the valence and the conductance band yielding a photon are possible. However, this is unlike to happen. Indirect recombination has higher probability.

This is mainly due to impurity dopants, phonons and lattice defects in the silicon. Typically, detectors are fabricated in regions that are free of defects. This region has a depth of a few microns and long recombination times. The bulk has a large number of defects and recombination lifetimes are shorter. Long wavelengths typically reach the bulk producing excited electrons with a very low lifetime, reducing quantum efficiency.

7.6 Diffusion Length

Excited electrons in CMOS detectors are collected by an electric field created in the depletion region (shadowed area in Fig.7.2). We can separate the collected carriers into two groups: Those absorbed within the depletion region (they are represented as *a2* in Fig. 7.2) and those who are created outside the depletion region and then they diffuse towards the depletion region where they are also collected (they are represented as *a1* and *a3* in Fig. 7.2). During the diffusion process, some carriers can recombine and being lost.

The diffusion length L_d represents the average distance that an excited electron can travels without recombining. L_d value depends on doping and temperature and can reach hundreds of micrometers in modern CMOS technologies. It decreases when we increase doping and temperature. Obviously, some carries created far away of the depletion region will not be collected. They have been represented as *I2* in Fig. 7.2. The depletion region is small in comparison to the semiconductor's volume. For this reason, in modern technologies, the majority of the carriers absorbed in the depletion region corresponds to photons absorbed outside the depletion region. They have been represented as *a1* and *a3* in Fig. 7.2.

7.7 Photo Charge and Photo Current

If ρ_0 is the incident power of monochromatic light per unit of area penetrating the semiconductor's bulk, the corresponding number of photons is given by

$$\Delta ph = \frac{A \cdot \rho_0 \cdot \Delta T_{int}}{(h \cdot \lambda)/\lambda} \tag{7.4}$$

where A is the detector area, and ΔT_{int} is the time the semiconductor is exposed to light. Previously, we defined the *quantum efficiency*, $\eta(\lambda)$, as the ration between the number of incident photons and the number of collected charges. Hence, we can express it mathematically as

$$\Delta n = \eta(\lambda) \cdot \Delta ph \tag{7.5}$$

As we stated before, the quantum efficiency is a parameter that shows a strong dependence with technology and the incident wavelength λ. It is difficult to model it mathematically and for this reason, it is usually determined experimentally before characterizing sensors. If we combine the two prior equations, the number of detected charges is:

$$\Delta n = \frac{A \cdot \rho_0 \cdot \Delta T_{int}}{h \cdot c} \cdot \lambda \cdot \eta(\lambda) = A \cdot \rho_0 \cdot H \cdot \xi(\lambda) \cdot \Delta T_{int} \tag{7.6}$$

where H is a physical constant defined as $H = (h \cdot c)^{-1}$ and $\xi(\lambda) = \lambda \cdot \eta(\lambda)$ is a function of wavelength.

We define the *photo charge*, Q_{ph}, as the amount of charge detected. Thus,

$$Q_{ph} = q \cdot \Delta n = [A \cdot \xi(\lambda) \Delta T_{int}] \cdot \rho_0 \cdot q \cdot H \tag{7.7}$$

where q is the electron charge. It can also be described in terms of charge per unit time, *photo current*,

$$I_{ph} = \frac{Q_{ph}}{\Delta T_{int}} = [A \cdot \xi(\lambda)] \cdot \rho_0 \cdot q \cdot H \tag{7.8}$$

There is a linear dependence of photo current or photo charge with light power. Such dependence, also called *optical dynamic range* is kept during several decades. The upper limit is usually establish by the optical experimental setup. It is complicated to obtain scenes with an illumination above $50Klux$. The lower limit is determined by the *dark current*. We will explain it in the next section. The terms in brackets of the previous equation can be controlled directly or indirectly by the designer. Engineers usually choose tailored processes of fabrication with high quantum efficiency for the design of photo sensors. The area of the photo receptor is chosen by the designer. It has to be enough to generate a photo current that the front-end circuitry can detect/sense. Making it too high would increase the pixel size and hence, reduce the resolution of our sensor if we have an array of pixels. Therefore, there is an important trade-off between dynamic range and area consumption.

7.8 Dark Current

Dark current limits the minimum detectable photo current in CMOS sensors. It can be defined as the reverse current measured without illumination. The reverse current through a photo diode does not depend on illumination. It depends on the doping of the diode and temperature. According to [44], it can be expressed as

$$I_{dark} = \frac{A_j \cdot q \cdot n_i \cdot W}{2\tau_0} \tag{7.9}$$

where τ_0 is the effective lifetime of minority carriers, W is the width of the depletion region, and A_j is the effective area of influence of the incident light. The parameter τ_0 has a strong dependence with temperature. For this reason, temperature value when the dark current is measured is usually specified. We have to remark that dark current is a limiting factor in photo sensors design. If the photo receptor's area is reduced, the photo current will be reduced in the same way and their ration will remain constant.

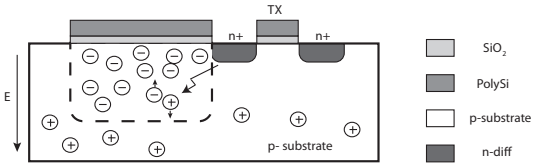


Figure 7.3: Photo gate and discharging transistor TX.

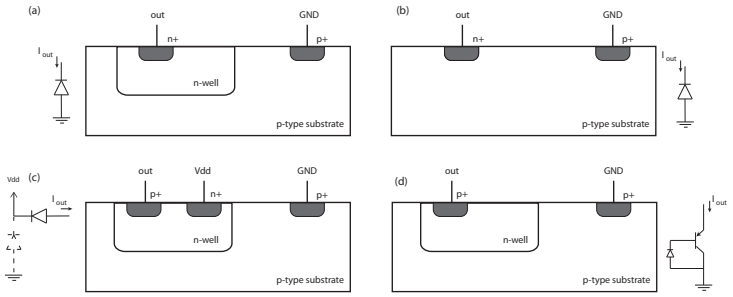


Figure 7.4: Examples of possible junction photo detectors available in CMOS technology

It exists tailored fabrication processes to reduce dark current. Some of them offer the possibility of incorporating to the designs *pinned photo diodes*. They used special or additional layers to minimize dark current. They also optimize the depth of the depletion region to improve the absorption of short wavelengths.

7.9 CMOS Photo Detector Structures

In CMOS technology there are two basic types of photo detectors. The first type is based on diodes reverse biased (photo diodes or photo transistors). The second one is known as photo gates. In both cases the depletion region is created in the junction of two different regions connected to different potentials to increase the size of the depletion region. The difference is how the excited electrons are collected. In the photo diodes, one of the terminals of one PN junction is directly connected to photo sensing front-end, so photo current can be directly sensed by the processing circuitry. Photo gates are however isolated and they just accumulate photo charge. Then, this charge is transferred to the photo sensing stage using discharging transistor switches. Fig. 7.4 shows some examples of photo diodes used to detect light. In theory, all the possible PN junctions available in CMOS technology could be used as photo diodes to detect light [45], but their quantum efficiency is different. Phototransistors are similar to photo diodes, but they use three PN junctions. They have more sensitivity to light than photo diodes because they have internal gain. However, they are more noisy and they are not available in all the technologies. Nowadays, they are rarely used. In Fig. 7.4(a),(b) and (c) 3 different photo diodes are shown. The examples (a) and (b) have higher quantum efficiency than (c). Fig. 7.4(d) displays a vertical photo transistor ($p^+/n^-/p^-$) available in CMOS technology. Fig. 7.3 shows a photo gate. Basically is made up by the gate of a MOS transistor placed over the substrate. The gate is set to a voltage value higher than the substrate. By this way, we can create a

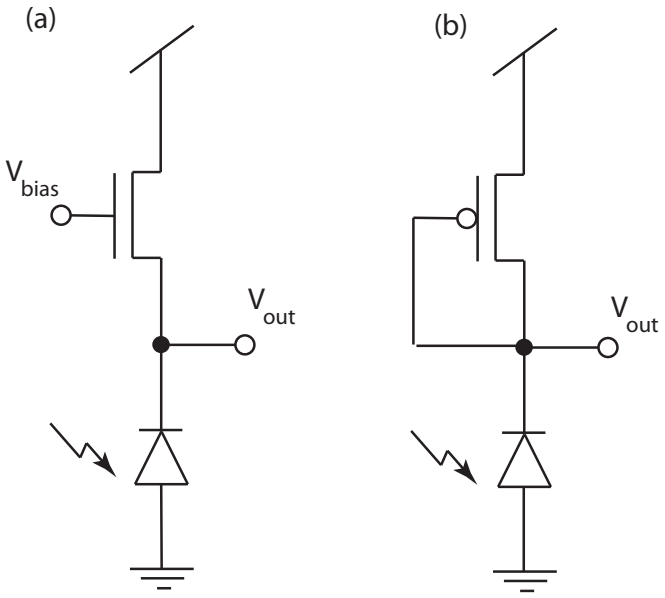


Figure 7.5: Two logarithmic amplifiers. The first one has an adjustable bias to control the output DC level. For the second one, the output voltage only depends on photo current value.

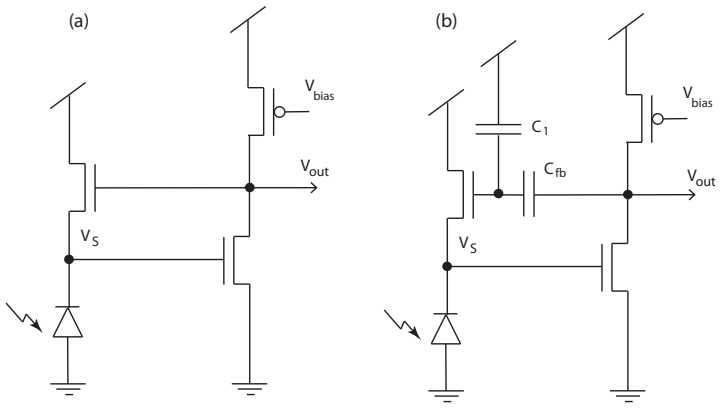


Figure 7.6: (a) Negative feedback amplifier. (b) Negative feedback amplifier with capacitive voltage divider to control the influence of the feedback loop.

depletion region underneath the oxide that can store photo charge. Then, this charge can be transferred using a transistor switch to transfer the charge to the next photo gate or the read-out circuitry. Photo gates are usually connected forming arrays. Their charge is transferred sequencially to the circuitry placed on the chip periphery.

7.10 Logarithmic Receptors

Human eye has a dynamic range of 10 decades. We are capable of distinguish objects and shapes under moonlight (< 1 lux) and we can see with very bright conditions (above 50Klux). Neuromorphic circuits have to cope with these hard constraints. Logarithm amplification is the way forward to compress the huge dynamic range of the input signal into a voltage that can range between GND and Vdd. Fig. 7.5 shows two examples of logarithmic I-V amplifiers. The first one has an adjustable bias. Its output signal is given by the expression:

$$V_{out} = \kappa \cdot V_{bias} - \kappa \cdot V_{T0} - U_T \cdot ln\left(\frac{I_{ph}}{I_S}\right) \tag{7.10}$$

The second one is simpler and avoid the use of one external pin to control the output DC level that is given by

$$V_{out} = n \cdot V_{DD} - V_{T0} - U_T \cdot n \cdot ln\left(\frac{I_{ph}}{I_S}\right) \tag{7.11}$$

We are usually more interested in measured the relative variations or increments of the photo current with respect an static value. The logarithmic amplifier with negative feedback shown in Fig. 7.6(a) offers this possibility and provides stability and robustness against noise. It has an operation range of several decades and its its gain only depends on the relative variations of the photo current. We can assume that the negative feedback loop can compensate all the variations of the input voltage and its small signal gain remains constant during all the operation range. Thus, the average (great signal) voltage level at the output of the amplifier is given by:

$$V_{out} = n^2 \cdot V_{dd} - n \cdot V_{bias} + U_T \cdot n \cdot ln\left(\frac{I_{ph}}{I_S}\right) + V_{T0} \tag{7.12}$$

Negative feedback also makes the sensor more strong against noise and external perturbations. It is also possible to limit or control the effect of the negative feedback placing a capacitive divider [46] in the feedback loop (see 7.6(b)). In that case, the voltage at the output of the amplifier can be expressed as:

$$V_{out} = n^2 \cdot V_{dd} - n \cdot V_{bias} + \frac{C_{tot}}{C_{fb}} \cdot U_T \cdot n \cdot ln\left(\frac{I_{ph}}{I_S}\right) + V_{T0} \tag{7.13}$$

Where $C_{tot} = C_1 + C_{fb}$.

Removing the DC component of this circuit, a sensor capable of detecting the temporal variations of intensity could be created. This is equivalent to detect movement and it is the principle of operation of the transient sensor described in [47].

Chapter 8

Retinomorphic Circuits

8.1 The Retina

The retina is the photo sensitive layer at the back of our eyeball (figure 8.1).

It comes as quite some surprise to most people, how much image processing in the human visual system is done before we consciously perceive an image. Taking one glance at a scene, we seem to see the whole picture as a camera would see it. But that is not really the case: The information that really reaches the higher areas of our brain is highly condensed. Just the relevant features of the picture are present here and a lot of details have been discarded.

Already the eye is more specialized than a camera. The retina which is the photo-sensitive part of the eye, what would be the film in a photo camera, is not uniform. We only perceive a clear colour camera-like picture in the very center of our visual field. With distance from the center (the fovea or macula) the resolution and the colour perception decreases and the light and motion sensitivity increases (Example: when moving a hand forward from out of sight to the side of ones head, we are able to spot it sooner if the fingers are wiggled). The image is further processed before it is sent on to the Thalamus (the major relay station of most sensory information before it reaches cortex) and the brain. There is some horizontal interactions going on between the nerve cells on top of the photo receptors that help adapt cells to the average light level (Example: looking out a window we can see features outside and inside at the same time, although the illumination levels are some orders of magnitude different. A camera can only adjust its shutter to either the inside light level (with the window appearing one brilliantly white spot) or to the outside light level (with the inside as a uniform dark frame).) and to enhance edges (Example: if you look at walls in a white room that meet in a corner, one wall will be more bright and the other more dark because of the angle of the main source of illumination. And very close to the corner, you will perceive the brighter wall becoming brighter still, and the darker more dark. Thus, the contrast between them is enhanced in your eye, but physically the brightness is not really changing towards that corner.).

A conceptual model that shows some of the nerve cells that lie on top of the actual photo receptors could look something like figure 8.2, or even more schematized and simplified in figure 8.3. The photo receptors (rods (peripheral vision, not colour- but very illumination-sensitive: better for

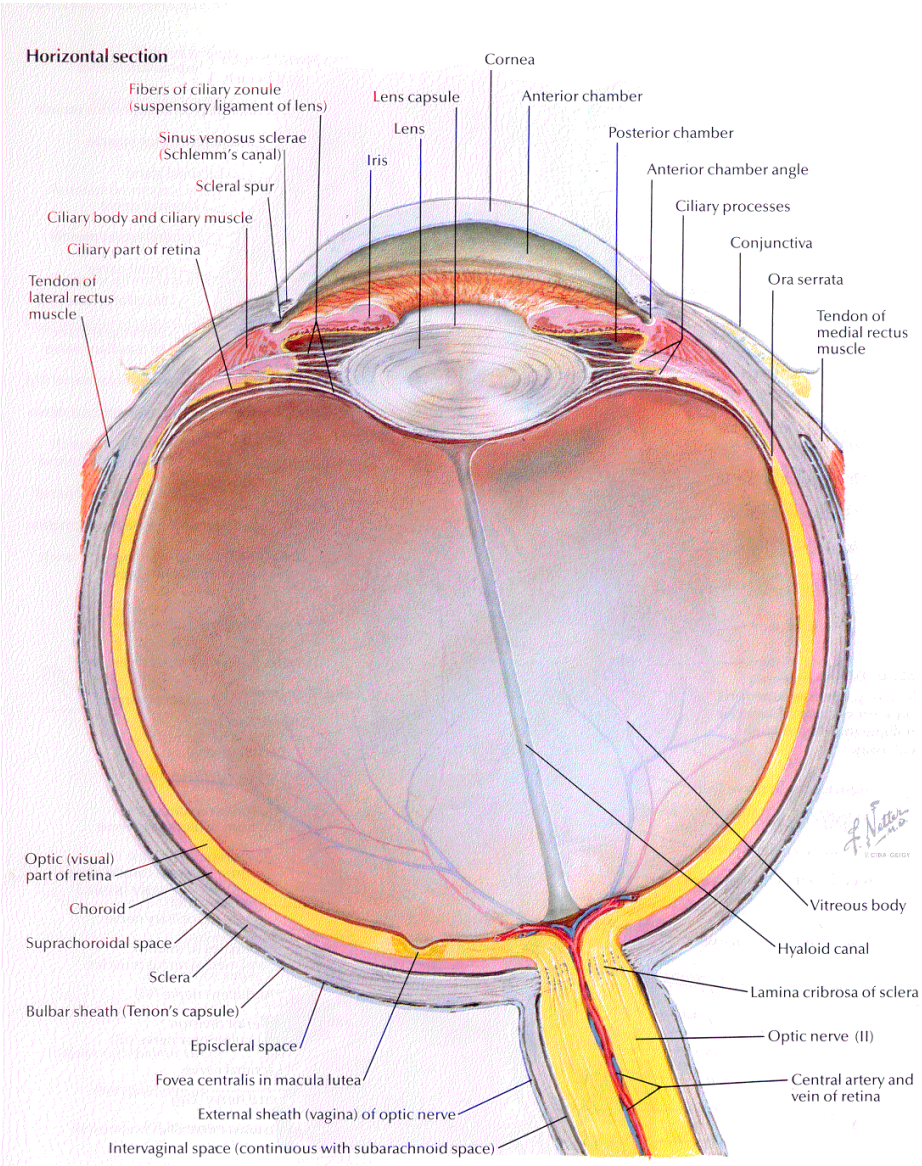


Figure 8.1: Cross section of the eyball according to [9]

night vision) and cones (colour sensitive, concentrated in the center of the visual field)) are excited by light that is focused on them through the eye lens. They adapt or tire when stimulated and the signal they are sending out is thus attenuated when presented with a constant stimulus. The photo receptors in turn excite so called horizontal cells that actually collect input from a group of photo receptors and from their colleagues. Thus, their own activity reflects the collective average light level of a certain neighbourhood. The difference between that collective light level and the local light level is computed in the bipolar cells. Some of them get excited (arrow synapses in figure 8.3) by the direct input from the photo receptor and subtract (bubble synapses in figure 8.3) the collective average from it. They react to bright spots. Other Bipolar cells get inhibited by the local photo receptor and excited by the neighbourhood average. Those get excited by dark spots.

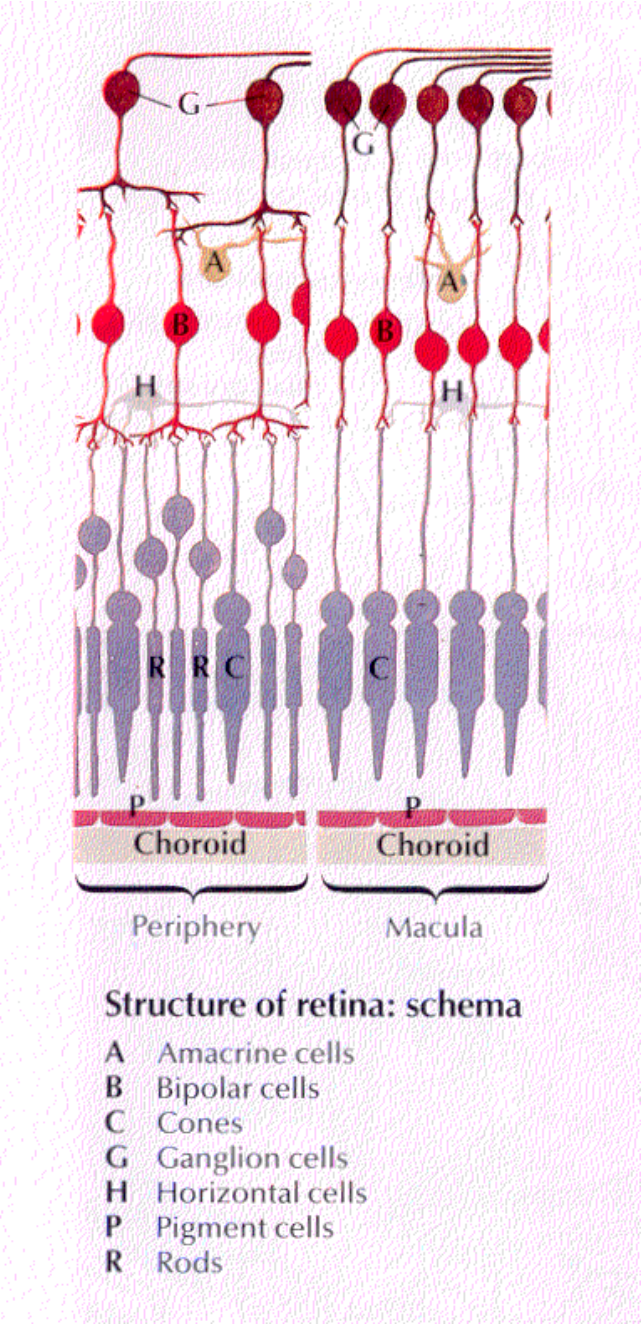


Figure 8.2: Retinal Cells according to [9]

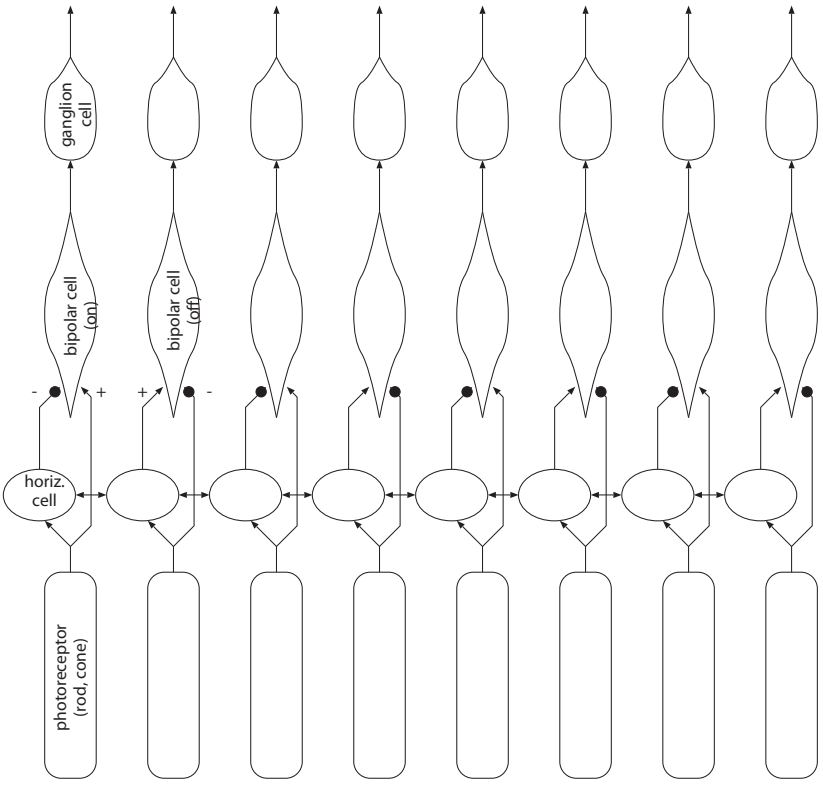


Figure 8.3: A schematic impression of retinal cells

8.2 CMOS photo sensors

Neuromorphic imagers/vision sensors or simply retinomorphic circuits make use of the fact that CMOS photo sensors can be co-located on the same substrate as processing circuitry. Thus, they are perfectly suited to mimic those local computations of the retina.

Photo active structures in semiconductors make use of the fact that photons lead to impact ionizations in the material. Thus, a electron is set free from an atom in the conductor and a electron-hole pair is created: a negative and a positive carrier charge. Normally those two charges recombine immediately again, unless there is an electric field within the material separating them for good, in which case there will be a current of these charges flowing along that electric field.

Some definitions:

Intensity radiation energy per time unit per area

Contrast ratio of highest and lowest intensity in a picture

Some criterions applicable to photo cells are:

- gain
- speed

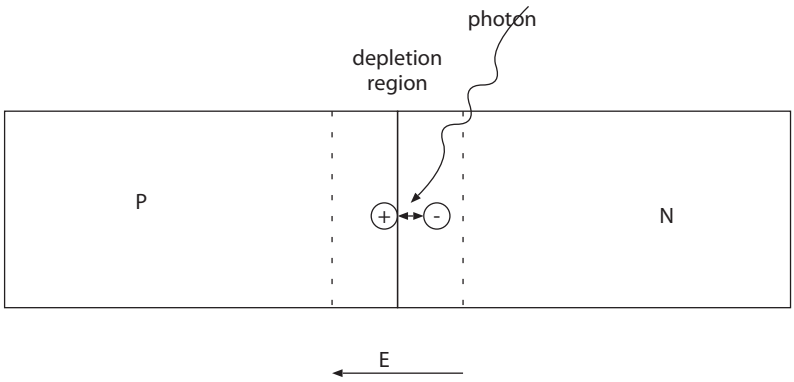


Figure 8.4: Separating light impact electron-hole pairs in the depletion region of a PN junction (photo diode)

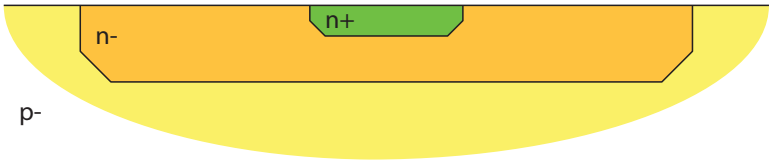


Figure 8.5: A possible layout for a photo diode

- signal noise (temporal)
- mismatch noise (spatial)
- linear, logarithmic etc.
- output resistance
- fill factor

8.2.1 Photo diodes

There is an electric field in the depletion region of PN junction that can be used to separate spontaneous electron hole pairs caused by photon impact (figure 8.4, layout in figure 8.5). A steady current starts to flow across the junction. The structure can be approximated as a current source that is linear with light intensity. Photo diodes are fast, low gain and low noise photo active structures.

8.2.2 Photo transistors

In the layout in figure figure 8.6 a PNP photo diode is depicted. The photo current, an inverse current through the lower base-emitter junction builds

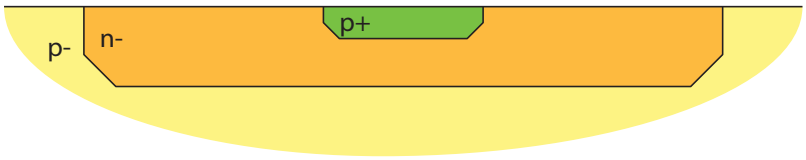


Figure 8.6: PNP photo transistor layout

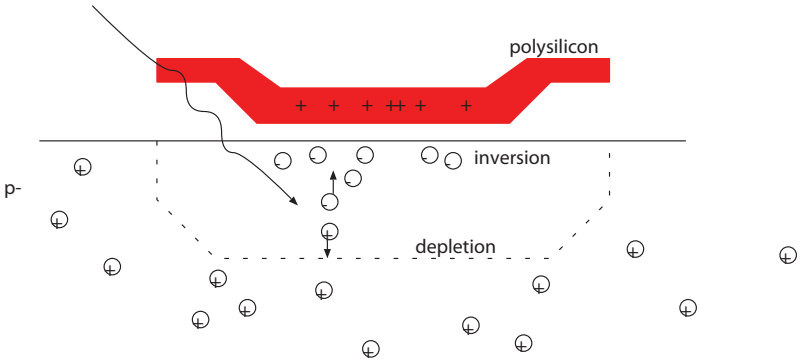


Figure 8.7: Charge separation in a photo gate

up charge that then flows back as a forward control current from emitter to base and thus induces a much amplified current from the emitter to the collector in the manner of a bipolar PNPtransistor. Photo transistors are slower than photo diodes but have bigger gain. On the other hand the signal is more noisy too.

8.2.3 Photo gates

Transistor gates in CMOS technology can also be used to make photo active devices (figure 8.7). Above a P (or N) substrate and at a potential higher (lower) than the substrate's, they push away the majority carriers and create a depletion region (electric field in the substrate) that also separates light-generated electron-hole pairs. Thus, minority carriers will accumulate on the substrate surface in what would be the 'channel' in a transistor. They are for instance used in charge coupled device (CCD) cameras (see section 8.4.2).

8.3 Photo Current Amplification

8.3.1 Linear by Early effect

Usually photo currents are very small. Thus, it is often desirable to amplify them already in the pixel, to allow more reliable read out or (as in neuromorphic circuits) to use them for further computations.

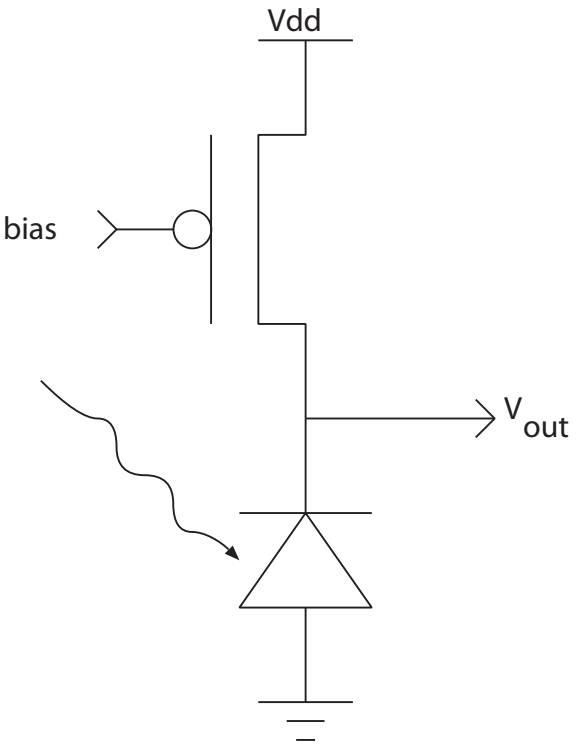


Figure 8.8: Linear amplification due to drain resistance/Early effect

One way of achieving high linear voltage gain is shown in figure 8.8. The Early effect supplies the gain and therefore it is not easily controllable. Also, the output resistance is too high to drive a read out line, so an extra driver stage (e.g. follower) needs to be added to the pixel. The high gain allows the cell to be very sensitive within a very small range of intensity. But outside that range the cell simply saturates. The range can be set by the bias voltage.

8.3.2 Logarithmic by gate to source voltage

Logarithmic amplification (as provided by the two circuits in figure 8.9) is highly desirable because it allows the pixel to operate on a big range of intensities with constant gain proportional to contrast: a 20% change of illumination, for example, will result in a constant output difference, independent of the absolute lighting conditions. The output voltage can be found analytically by solving the normal subthreshold transistor equation for the source voltage (for the NFET) or the gate voltage (for the PFET).

Still these pixels have too high output resistance to drive a read out line.

8.3.3 Common source amplification

All of the above amplifiers cannot drive a big load as it's only the photo current that drives the output voltage. To read out a photo cell by

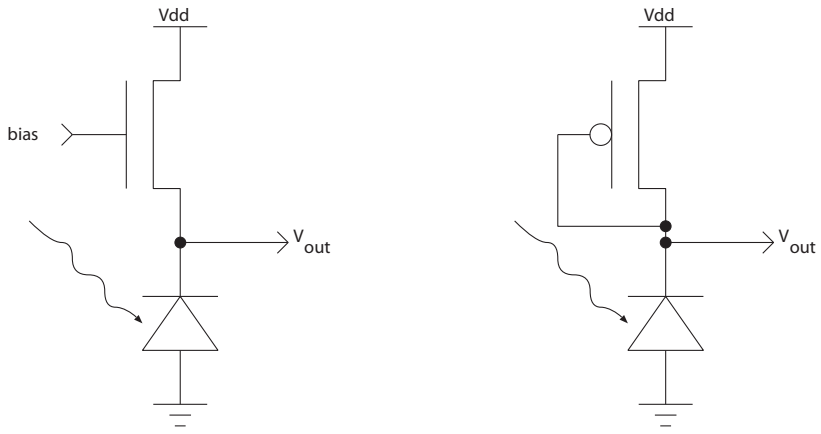


Figure 8.9: Two ways of achieving logarithmic amplification by the 'exponential source conductance' of an NFET (left) and by the 'exponential conductance' of a diode connected PFET (right).

addressing or scanning, however, it must be able to drive a common read out line, i.e. a low output resistance is needed. Thus, an additional amplification stage is needed. A most simple two transistor inverting amplification stage can achieve that: a common source amplifier (figure 8.10). This solution is more space preserving than for example a full blown follower (which is also possible to use).

To keep the amplification (of noise) limited, negative feedback can be applied e.g. in the manner depicted on the top of figure 8.11. Capacitive division can be used to implement different strength of negative feedback (bottom of figure 8.11). If one assumes infinite gain in the inverter, then the system gain is given by the ratio of the capacitances and the parameters of the transistor only: One can assume that whatever the change in photo current, the feedback transistor will compensate it 100%. Thus, one can assume that the current through the transistor, as given by the gate to source voltage, is equal to the photo current. Hence, the following formula can be derived:

$$V_{out} = \frac{C_{tot}}{C_{fb}} n U_T \log I_{photo} - n U_T \log I_S + V_{T0} + n V_S \tag{8.1}$$

Also, see figure 8.15 later in this text for a most clever extension of that circuit.

8.3.4 Source follower

The 'active pixel sensor' (APS, so named by the inventing company Photobit (later bought by Micron)), employs a source follower for read out amplification (figure 8.12 shows the 3 transistor APS variant). The charge accumulates under the photo gate. The control signal TX connects the photo gate to the gate of the transistor driving the read out line. A current is drawn by an external/global current source on the read out line and as

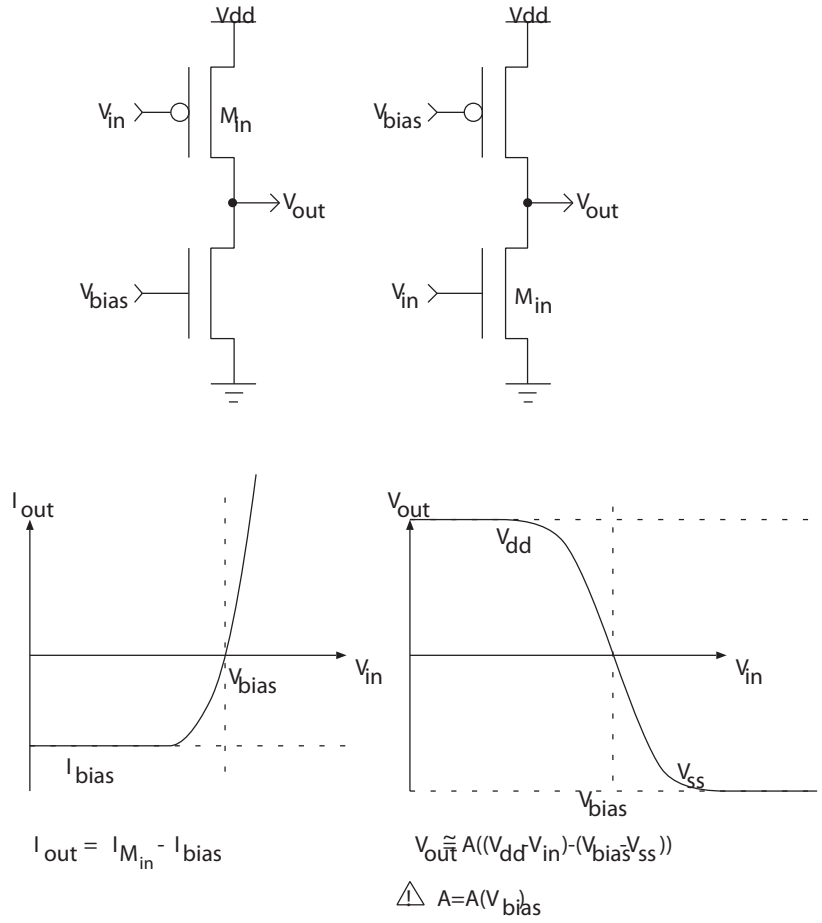


Figure 8.10: A variant of an inverter with a bias voltage that allows to set the gain and the driving current

the read out transistor's source is connected to it, the voltage on that line will be driven to the transistors gate voltage minus the threshold voltage. The driving current is given by the external current source and can be as big as required to drive the load of the entire line at the desired speed.

8.4 Read Out Strategies

8.4.1 Addressing and scanning

Addressing (random access or sequential scanning) is the most classic read out strategy used for many two dimensional structures, like for example memories. Individual cells are addressed from two sides of the two dimensional array by demultiplexers or scanners. The selected cell is granted access to a common read out line. The cells need to be able to drive that line. In order to make this feasible for reasonably large arrays, there is not really one single common line connecting all cells, but only one line connecting a single row which either connects all those rows to a single column line through a buffer, or through an ADC to a parallel shift register

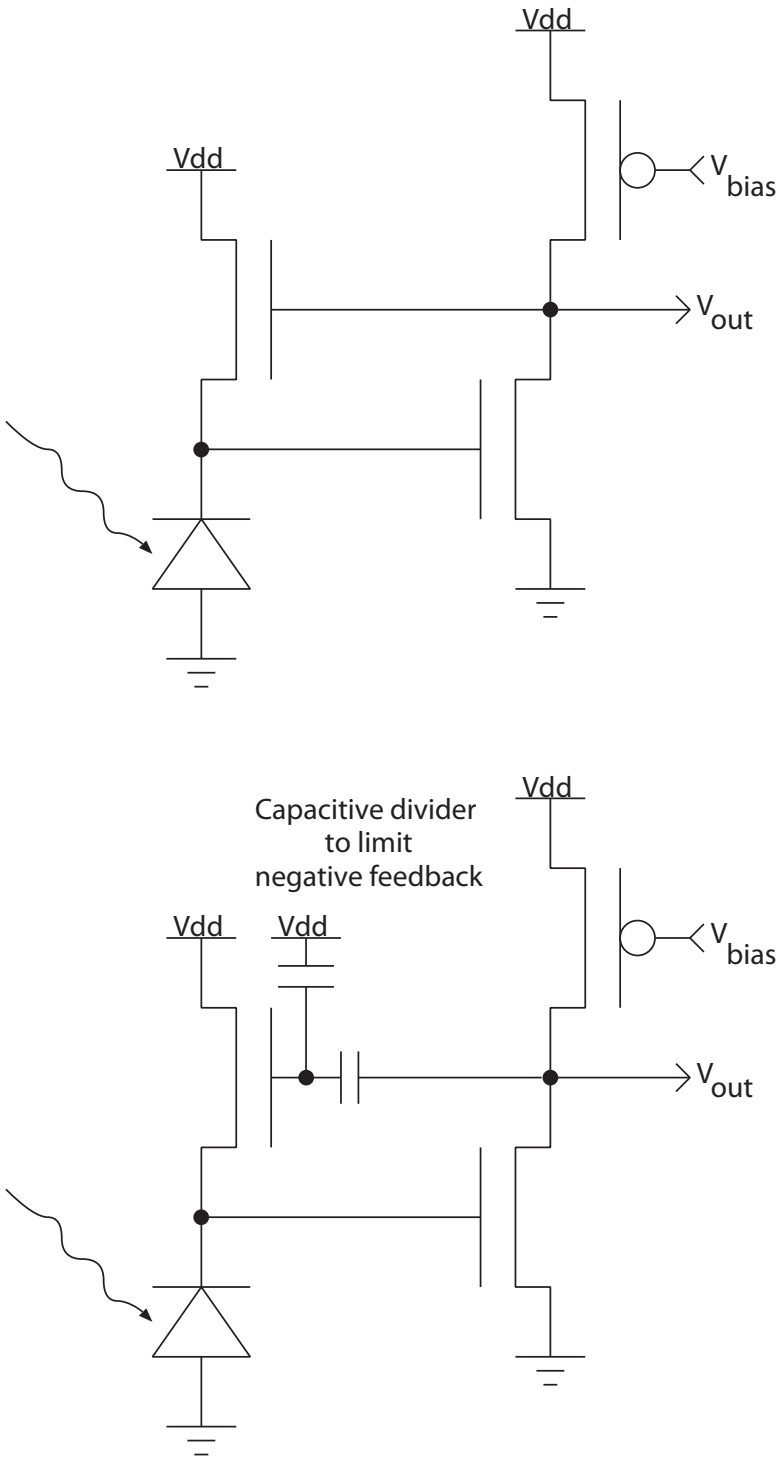


Figure 8.11: Amplified negative feedback almost nullifies the gain (top), actually if one assumes near infinite gain in the inverter, one can use capacitive voltage division to control the system gain (bottom). A problem, though, is the initialisation of the floating feedback node.

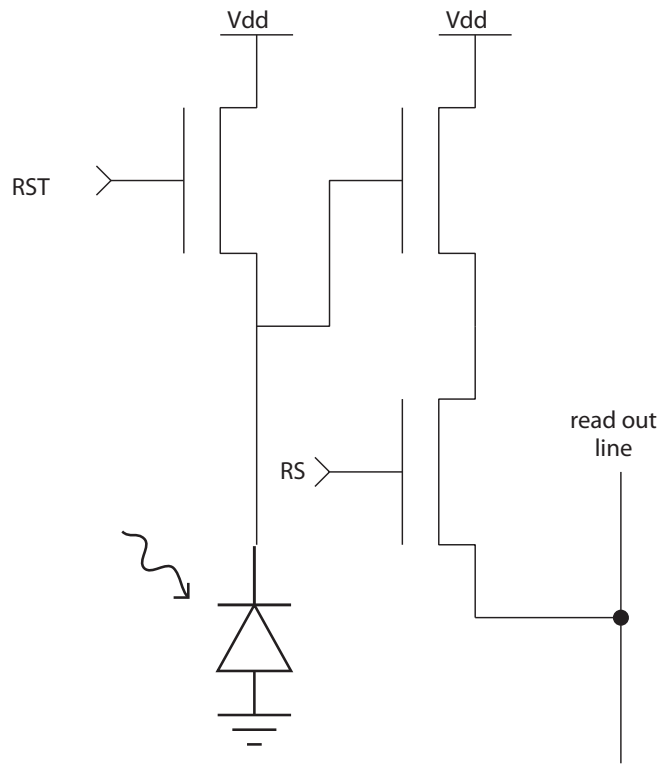


Figure 8.12: The most popular CMOS pixel present in almost every mobile phone today is the 'active pixel sensor' (APS) and is 'active' in so far as an amplification is performed in the pixel.

for reading the along the columns digitally. As an added bonus, this also makes a 'and' gate in the pixel unnecessary and a simple read strobe (RS) like in the APS in figure 8.12 is sufficient.

8.4.2 Charge coupled devices (CCD)

Before the APS, CCD used to be the dominating technique in the imager market allowing for the highest fill factor (the fraction of chip layout covered by photoactive elements). The key lies in the read out technique of the pixels (see figure 8.13). All pixels are connected via silicon of the same doping. (Gate-)electrodes on top of that substrate can create electrically isolated regions that can be charged individually by the photo gate principle (section 8.2.3). By letting a 'wave' travel through the electrodes, these charge pockets can be shifted from one electrode to the next to the border of the array where it can be read out. Very dense structures can be achieved like this.

8.4.3 Address event representation

An asynchronous integrating read out strategy is AER (also see chapter 6). How it could be used with a photo cell is depicted in figure 8.14. Much like

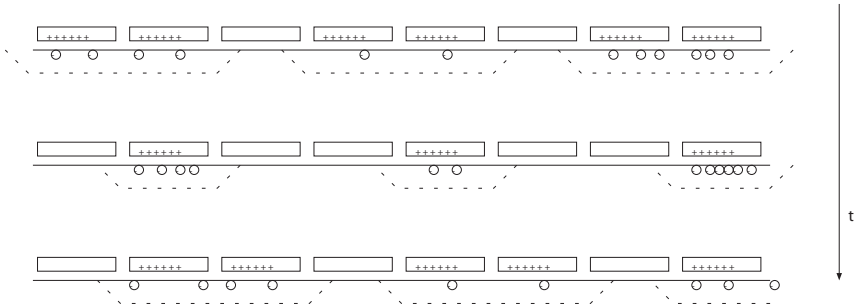


Figure 8.13: Read out in a charge coupled device: charges are accumulated under photo gates and then shifted in the array by appropriate switching among overlapping groups of gates

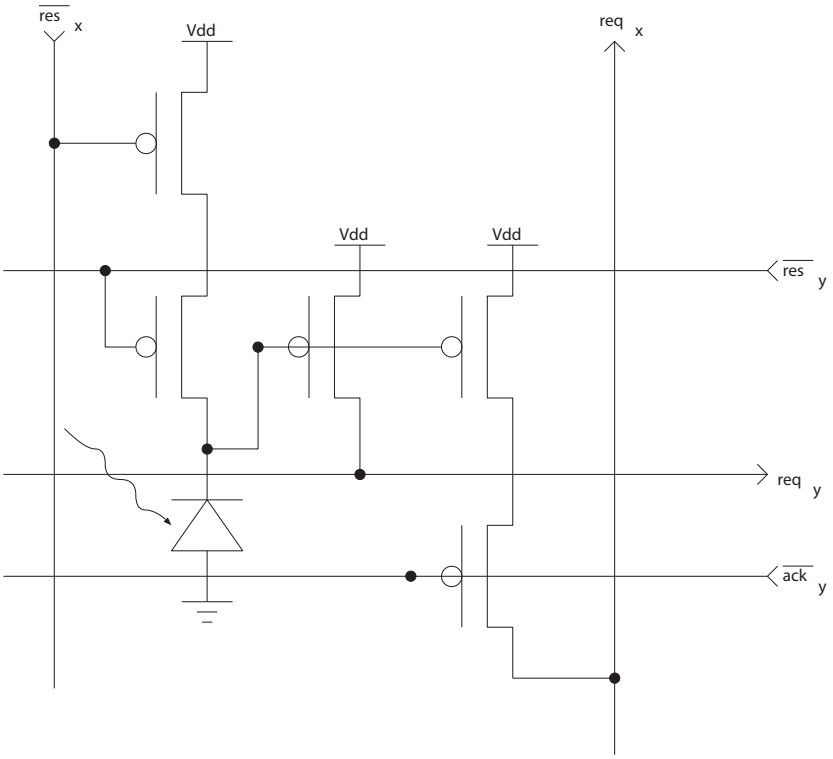


Figure 8.14: A (too) simple photo pixel for AER read out

in an integrate-and-fire neuron, a node in the photo cell gets charged by a photo current until it reaches a threshold. Then a signal is sent out along both x and y axis to the periphery where an address event is generated. The light intensity is so coded in the frequency of AEs. The example in the figure shows the principle of an arbitered AER, where the pixel has to apply for an AE to be generated, first with the y-arbiter on the right of the two dimensional array, and then (if acknowledged) with the x-arbiter on the top. A real implementation will need to address some non ideal properties of that circuit, though, and will look somewhat more complex.

8.5 Silicon retinae

8.5.1 Adaptive photo cell

One rather famous adaptive photo cell, as introduced in its original form by Tobi Delbrück [48] (figure 8.15), uses capacitive negative feedback in a most clever manner, achieving little negative feedback and therefore big gain for small signals, and much negative feedback and therefore little gain for large signals. In addition, this solution has high pass properties, such that it decreases DC offsets. It thus, shows some of the adaptive properties that are present in real retinal photo receptors, as described earlier in this text. The non-linear element in that schematics can for example be implemented compactly like in figure 8.16

8.5.2 Spatial contrast retina

An analog circuit (by Misha Mahowald [36] and similarly and much earlier with descrete components, i.e. not integrated on an ASIC by Fukushima [50]) that also implements the horizontal cells' functionality of contrast enhancement is shown in figure 8.17. The direct photo cell input is compared to a local average through a resistive net. The resistive network is layed out to connect a hexagonal grid. In this implementation there went a bit of an effort in implementing the resistances with transistors as quasi linear circuits. The photo cells were the adaptive photo cells.

An alternative solution [49] in 'current mode' that has no need of liner resistances but replaces them with a current mode diffuser network, and that is therefore better suited for direct CMOS implementation is in figure 8.18. on closer inspection an observant reader might spot its close relationship with the extended WTA circuits in section 3.9. This implementation does not use the adaptive pixel before contrast enahncement but the current output (EPSC) of the above photo and diffuser cell goes to an adaptive integrate and fire neuron (i.e. after contrast enhancement), which performs a similar function.

8.5.3 Temporal contrast retina

Another kind of processing in our retina is motion detection or change detection. Especially the rods in the retina react strongly to change. They are more dense than the colour sensitive cones in our peripheral vision. Therefore, motion out of the corner of our eyes is very likely to attract our attention.

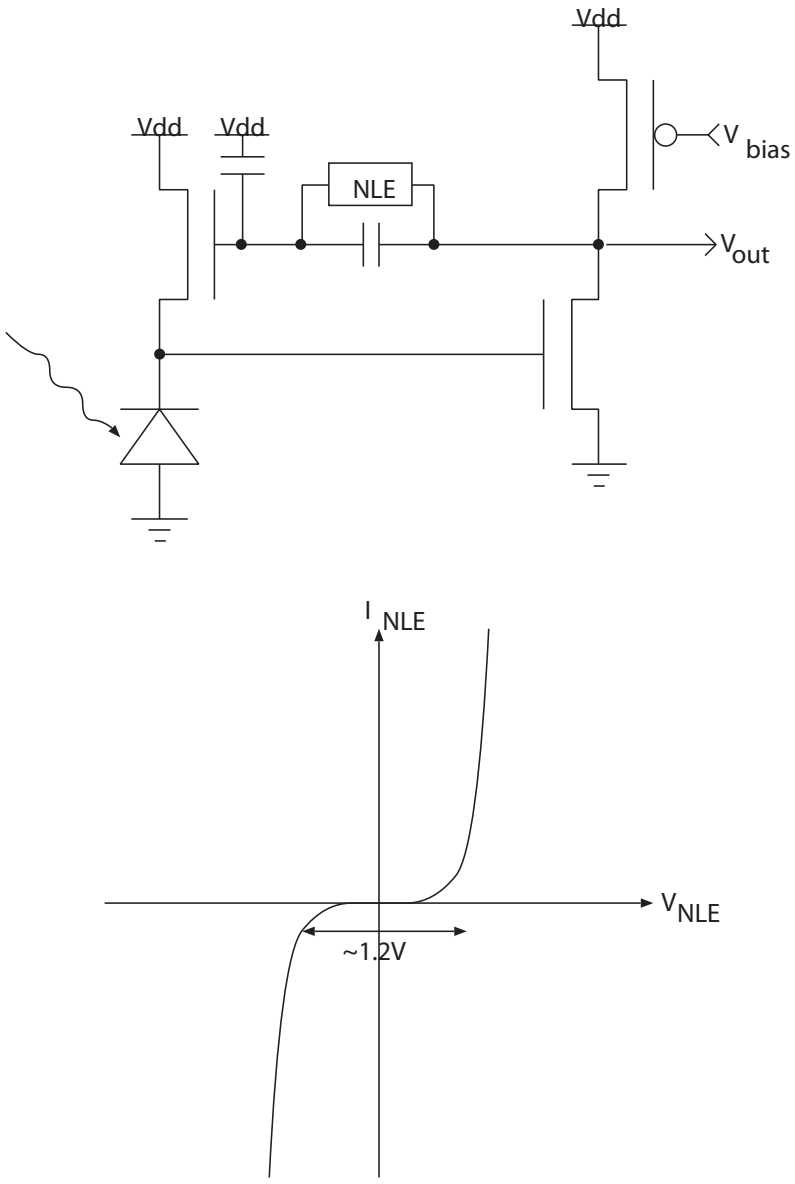


Figure 8.15: Adaptive photo cell as proposed in [48]

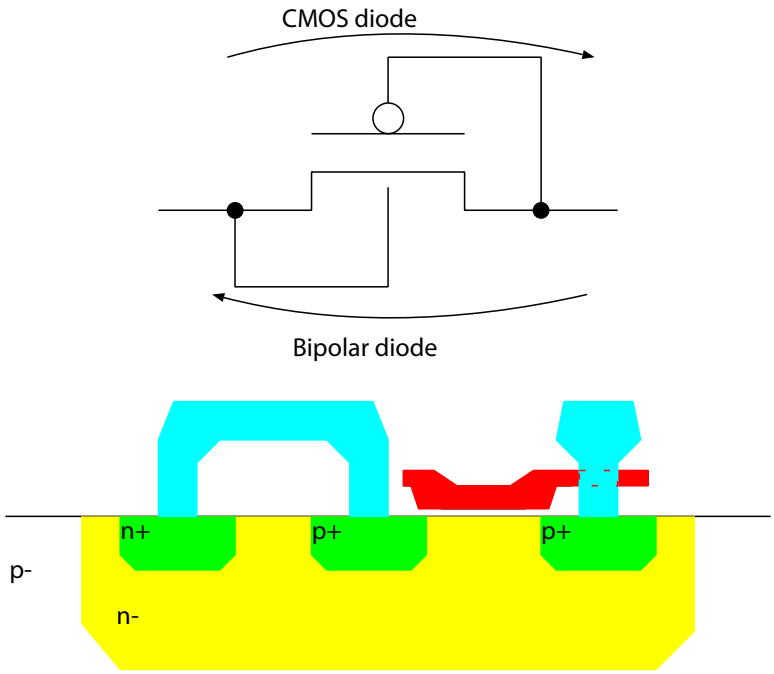


Figure 8.16: A non linear element (Tobi element), in effect consisting of two diodes in parallel, a CMOS diode connected transistor and a PN bipolar diode

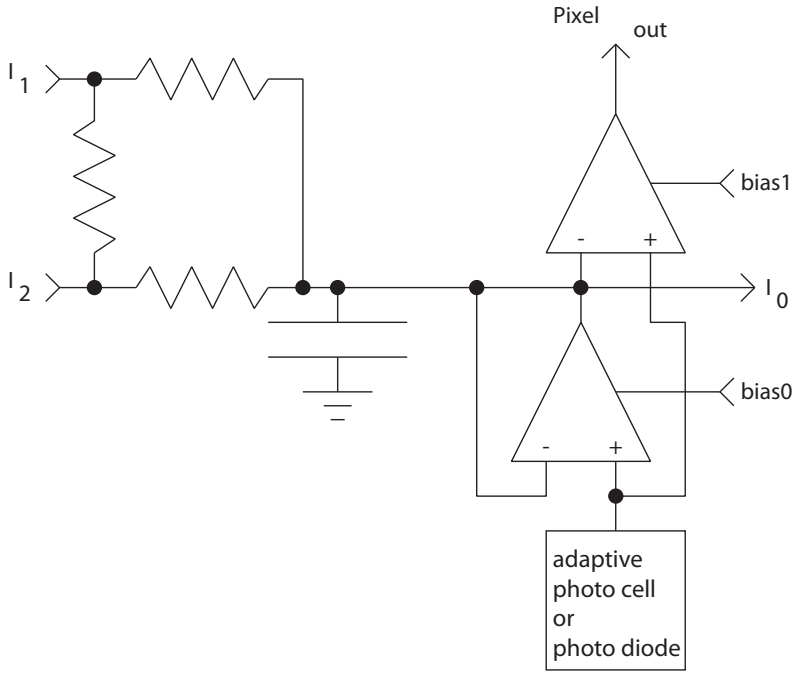


Figure 8.17: Silicon retina pixel according to [36]

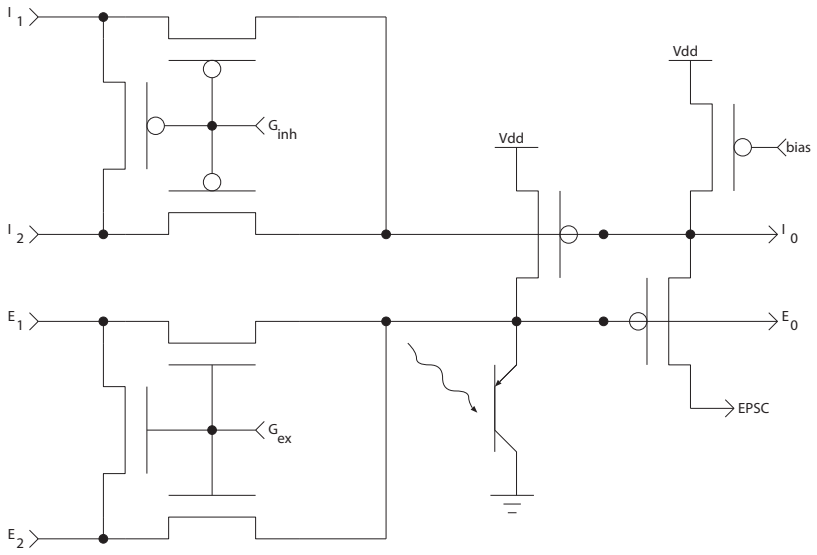


Figure 8.18: Silicon retina pixel according to [49]

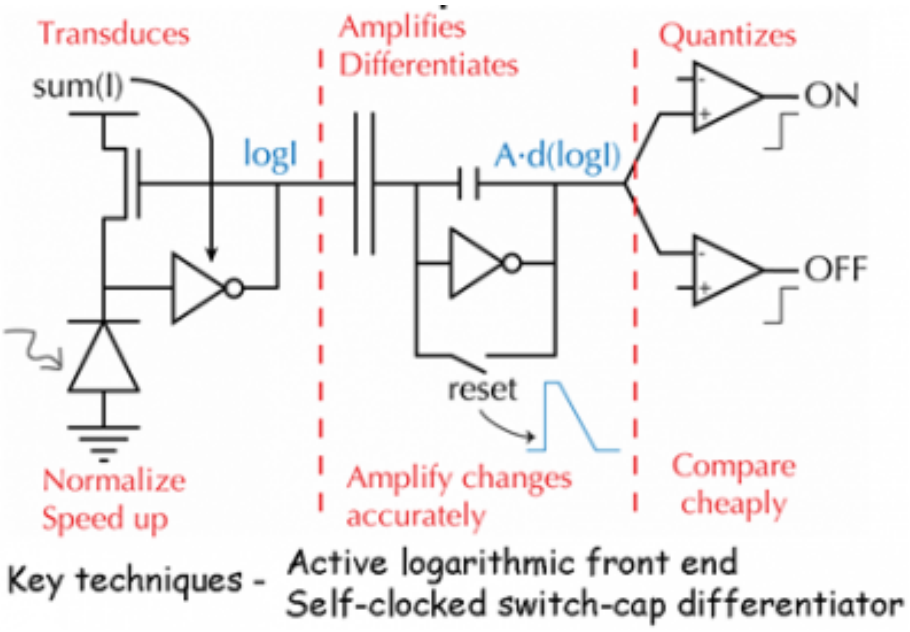


Figure 8.19: Temporal contrast vision sensor (later refered to as 'dynamic vision sensor' (DVS)) principle according to [47]

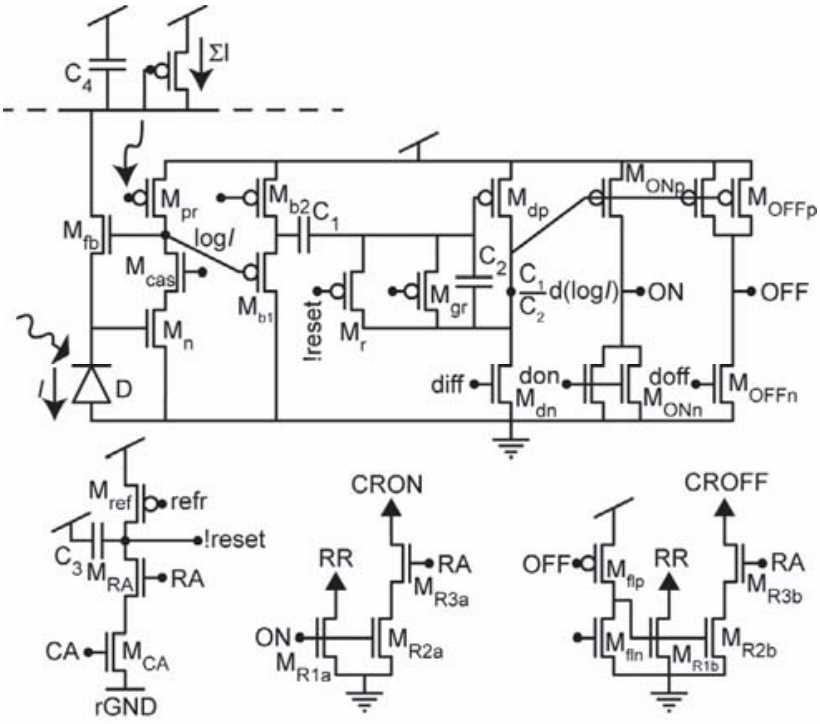


Figure 8.20: Temporal contrast vision sensor transistor level circuit according to [47]

This kind of sensor behaviour can be achieved by computing 'temporal contrast', i.e. compare a pixel value now with its value a short while ago and react according to the difference. This has been implemented in the 'dynamic vision sensor' (DVS) described in [47]. It is an AER-pixel that produces events at arate proportional to the temporal derivative of the logarithm of the pixel illumination.

Its principal is shown in figure 8.19. A photo current gets amplified logarithmically, further amplified and compared to its value at the last event. If the value exceeds a threshold, a new event is produced and the reference level is set to the current level. Since the change can be positive or negative, there are two thresholds.

More detail of the actual implementation on transistor-level is shown in figure 8.20.

8.6 Further Image Processing

Neuroscientists have long ago discovered neurons beyond the retina (e.g in the lateral geniculate nucleus (LGN, part of the Thalamus) and visual cortex) that are responsive to quite distinct properties of visual stimuli. Those properties become more and more complex, from simple retinal on/off cells to center surround in the LGN, 'simple' and 'complex' cells in

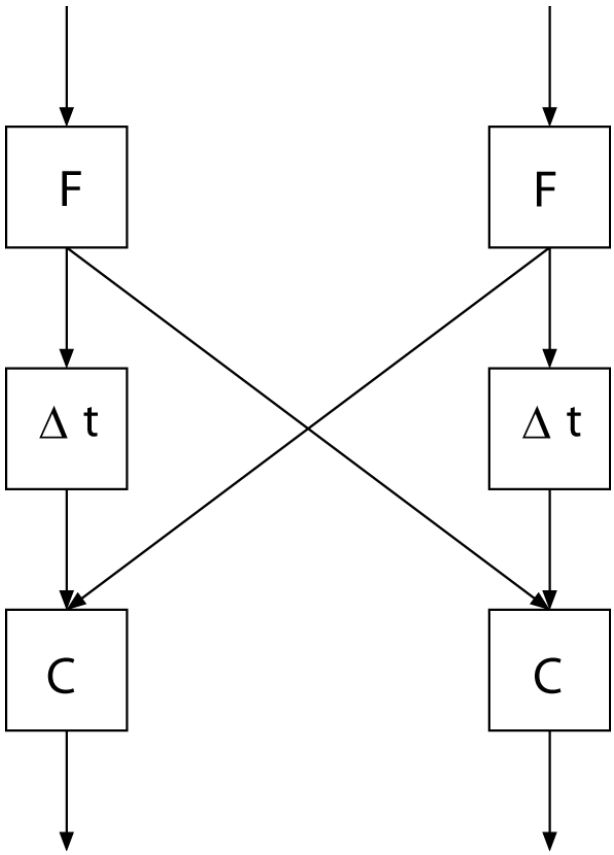


Figure 8.21: The Reichardt Detector. (F: feature (=token) detector / Δt: delay / C: coincidence detector)

V1 that are responsive to bars of a particular orientation and sometimes to direction of motion, to all kind of motion sensitive cells in the visual area MT (middle temporal, also known as V5), and finally cells that are responsive for example to high level objects such as faces. But in contrast to the retina, the architecture giving rise to these particular responses of cells is not clearly known and cause for debates. Most models are quite content in reproducing the properties without a strong claim of being faithful to the underlying neuroanatomy.

8.6.1 Motion

8.6.1.1 Token Based

These are algorithms that rely on a previous image processing stage that is able to identify and localize a particular token (feature/object) in the visual field. The classic example is the Reichardt detector (figure 8.21). A later example is found in [51]. It correlates the occurence of a token in two different locations at two different times, i.e. if a token (detected by block F) moves from left to right in the time that a signal is delayed in block Δt, then the lefthand side correlation/coincidence block C will receive two simultaneous inputs and thus produce an output indicating this left to right motion. As may be intuitively clear from the way the Reichardt detector is depicted, these algorithms can be implemented relatively easily with plausible neural building blocks. Delays can be achieved in dendrites

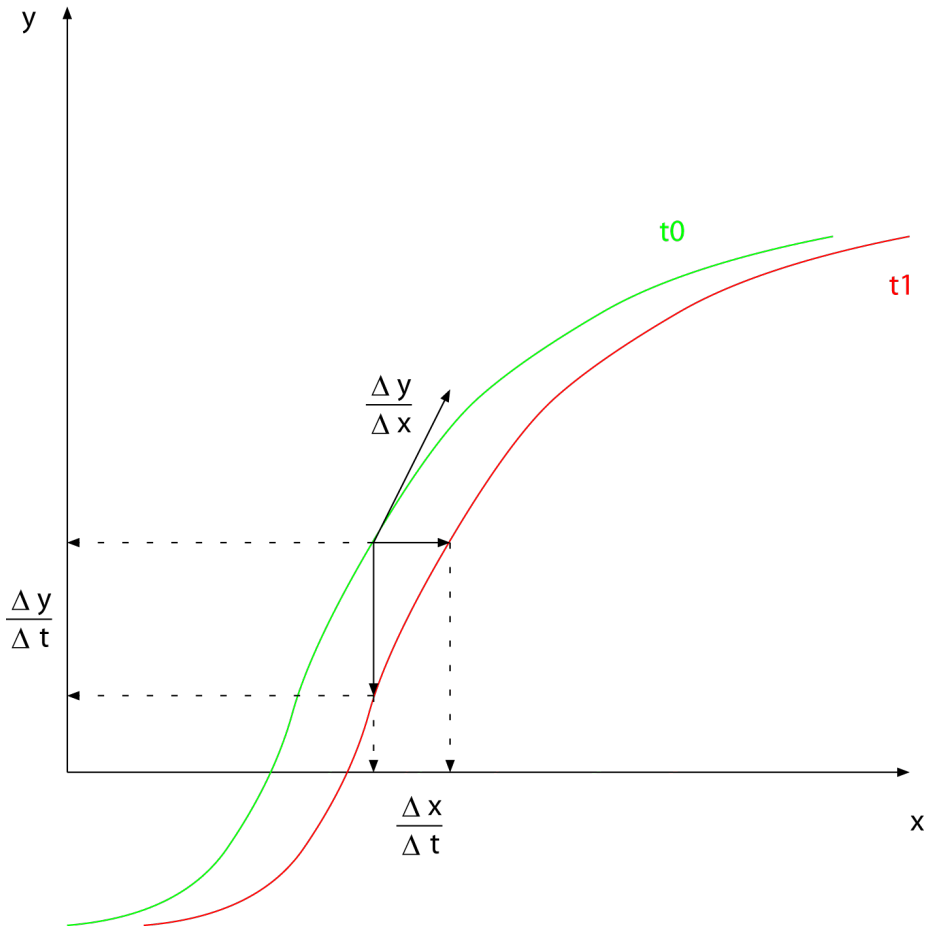


Figure 8.22: An illustration of intensity based motion estimation. The intensity profile of a surface at t_0 (green) and t_1 (red is shown). Assuming that this profile is constant on the object surface, only motion can be responsible for the change in this interval. Dividing the change in time, by the change in space, one obtains the speed of the surface, i.e. in the 1-dimensional case. The 2-dimensional case has unfortunately multiple solutions, when looking at just one pixel.

and axons. Coincidence detection can be performed by integrate and fire neurons with a rather high leakage. So it is well possible, although not proven, that the nervous system performs a kind of token based motion detection.

8.6.1.2 Intensity Based

In contrast to token based motion detection algorithms, these work directly on the intensity profile of a moving surface. (A recent example: [52]) Assuming that the surface does not change its emmission intensity the following equations describe the speed of the observed surface for a 1D surface (where y is the surface brightness):

$$v_x \frac{dy}{dx} = -\frac{dy}{dt} \tag{8.2}$$

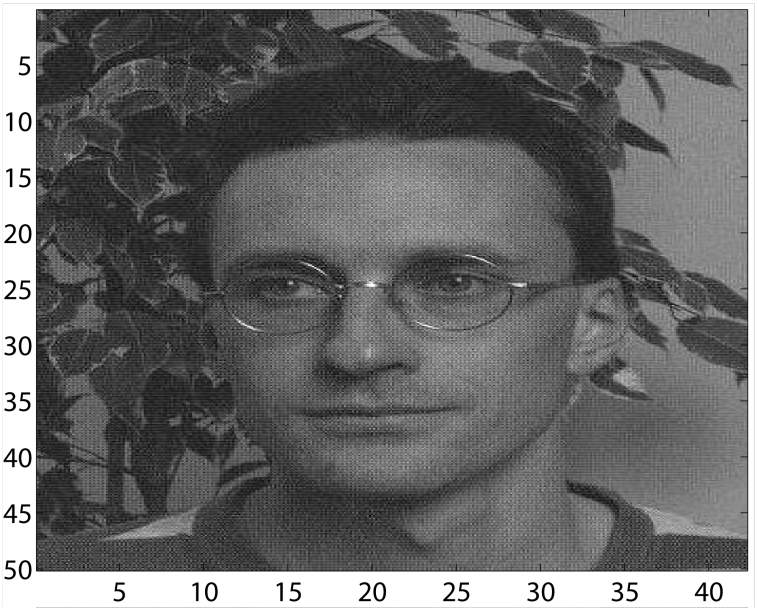


Figure 8.23: A photograph of an every day natural scene, that will be used to illustrate some image processing

for a 2D surface (where z is the surface brightness):

$$\nu_x \frac{dz}{dx} + \nu_y \frac{dz}{dy} = -\frac{dz}{dt} \tag{8.3}$$

In 2D in general, this equation does not have just one solution for $\vec{\nu} = (\nu_x, \nu_y)$ in one point, when the spatial derivatives ($\frac{dz}{dx}$ and $\frac{dz}{dy}$) and temporal derivative ($dzdt$) are given. The situation is better, if one can assume that one observes several points of a rigid object, i.e. points that move with the same speed and direction. But even so there are non trivial cases where a single solution cannot be found. An example thereof is the so called 'aperture problem'. It appears if within a limited field of view the observed moving surface intensity has spatial derivative equal to zero along one direction. Imagine a 45 degree (lower left to upper right) infinitely long edge moving from right to left across your visual field. You would observe the same, if the edge would move from bottom to the top at the same speed.

8.6.2 Feature maps

Feature maps are formed by convolving an image with a feature kernel. Convolution is a mathematical operator on two functions (f and g) that combines them to one function h. 'h is the convolution of f with g' means:

$$h(x) = \int_{-\infty}^{\infty} f(x-s)g(s)ds \tag{8.4}$$

Such a convolution can be performed by a particular neural network: The output neurons represent h(x). They are all connected with the same weight

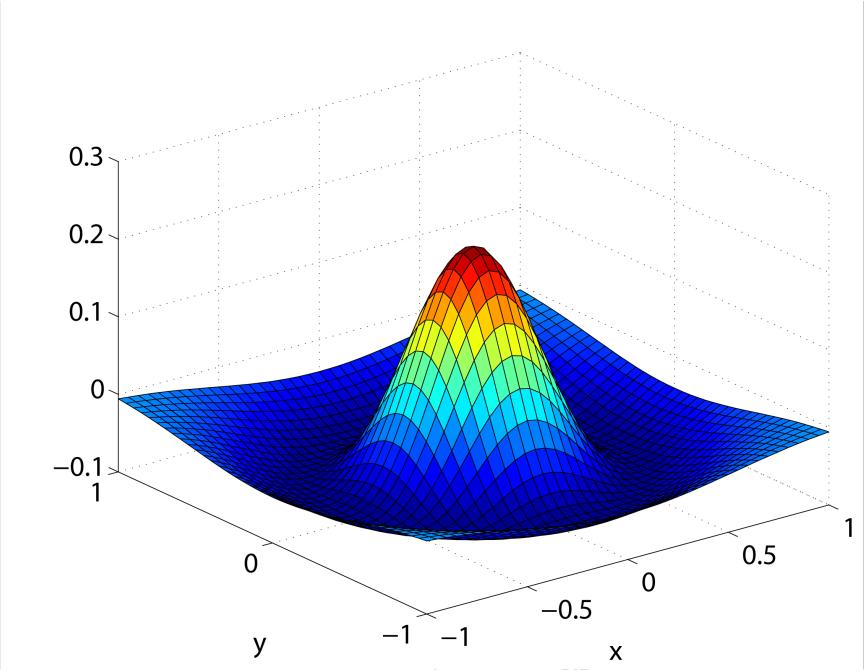


Figure 8.24: 2 dimensional surface plot of a 'difference of Gaussians'- or 'Mexican hat' function.

pattern g(s) to the input neurons f(x-s), relative to their position x. In other words if we denote the output neurons more traditionally with indices h_j and the input neurons with f_i, and a weight from f_i to h_j with $g_{i,j}$, then $\forall (i,j) \in \mathrm{N}^2, \nu \in \mathrm{Z} : g_{i,j} = g_{i+\nu,j+\nu}$

In image processing convolution is often used to bring out particular features in pictures. Two examples are illustrated in figures 8.23 to figure 8.29. Figure 8.23 is the original picture. Figure 8.26 shows the same picture with contrast enhancement, i.e. where there has been a local contrast, a transition from dark to bright or vice versa, there is now a bright and a dark bar marking that transition. Areas that where of uniform illumination are now all of a uniform gray. That is a useful preprocessing of a picture for edge extraction. This effect is achieved by the convolution with the function/kernel in figures 8.24(surface plot) and 8.25 (colour encoded contour plot), often referred to as 'difference of Gaussians' or 'Mexican hat'.

Another feature is extracted in figure 8.29. There only edges of a particular orientation remain clear in the picture, namely edges with an orientation parallel to the diagonal from lower left to upper right of the picture. Edges that mark a transition from dark to bright from the upper left to the lower right are marked with a bright bar, and transitions from bright to dark, with a dark bar. The picture gets a relief like quality through this. The convolution kernel achieving this is shown in figures 8.27 and 8.28.

Along the visual processing pathway of the nervous system such feature maps can indeed be found. For example in the retina there are the bipolar cells that respond to the same feature (strongly resembling a Mexican hat function) in different positions and that is also true for cells in the LGN (lateral geniculate nucleus, the part of the Thalamus relaying visual signals).

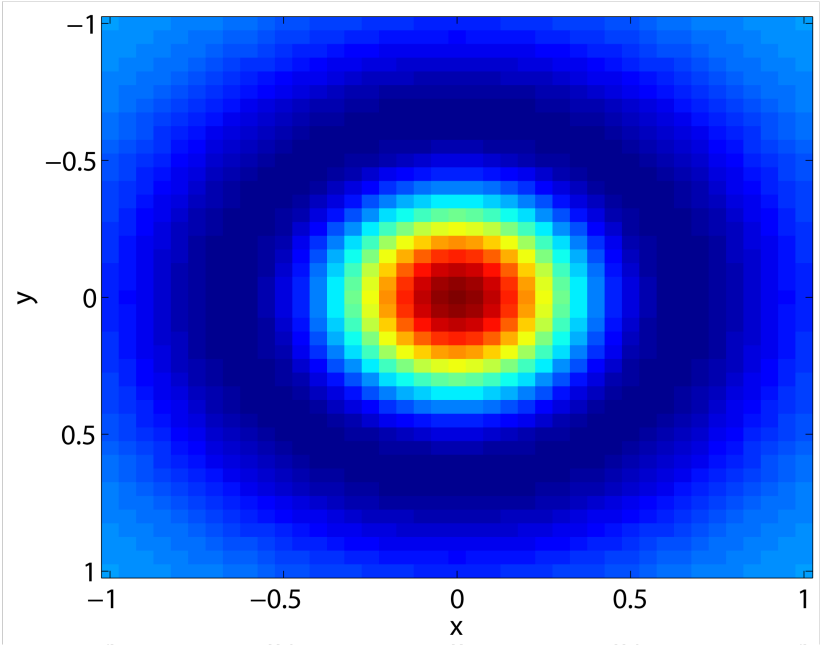


Figure 8.25: 2 dimensional colour code plot of a 'difference of Gaussians'- or 'Mexican hat' function.

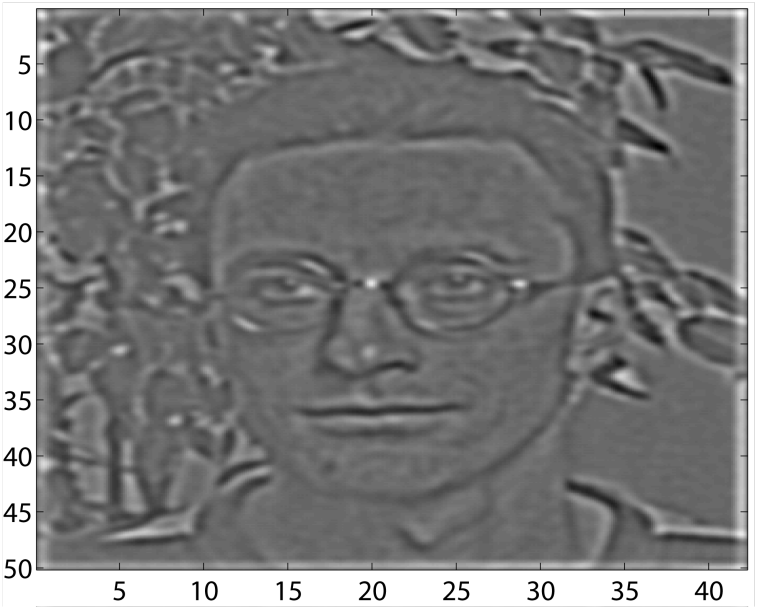


Figure 8.26: The image of figure 8.23 that has been convolved with a 'difference of Gaussians' convolution kernel: Local contrast, mostly edges, are enhanced. Uniform surfaces appear gray.

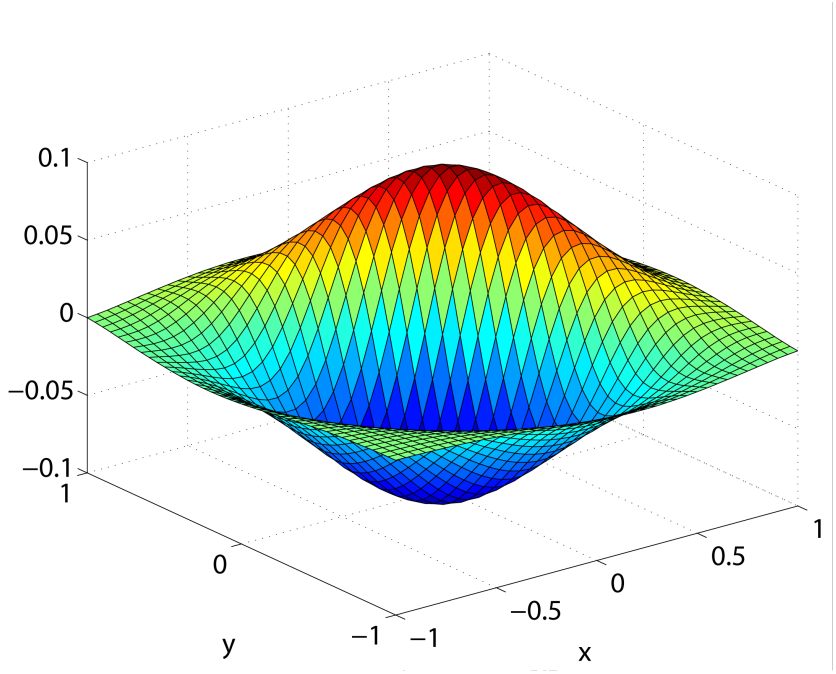


Figure 8.27: A function suited as a convolution kernel for extracting edges at an angle of 45 degrees in surface plot

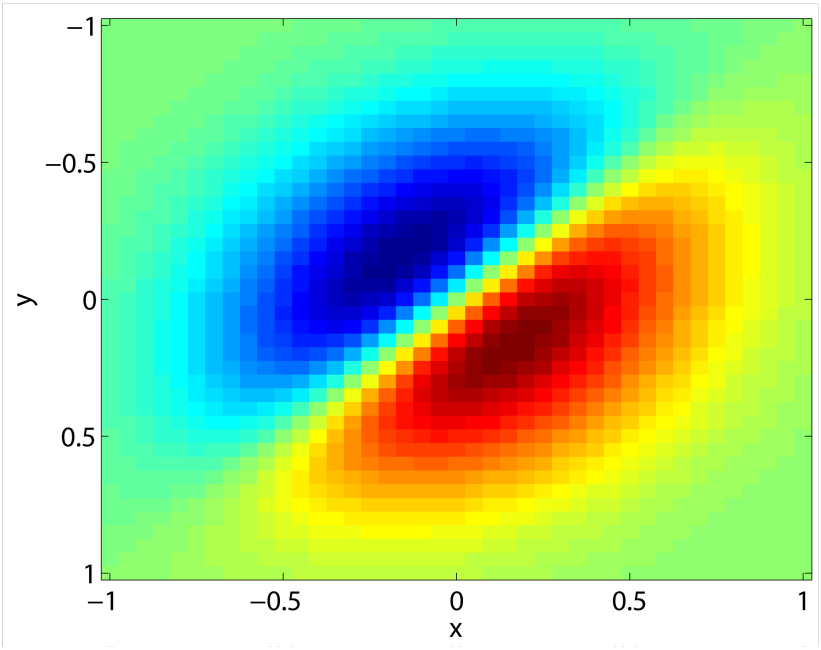


Figure 8.28: A function suited as a convolution kernel for extracting edges at an angle of 45 degrees in colour code plot

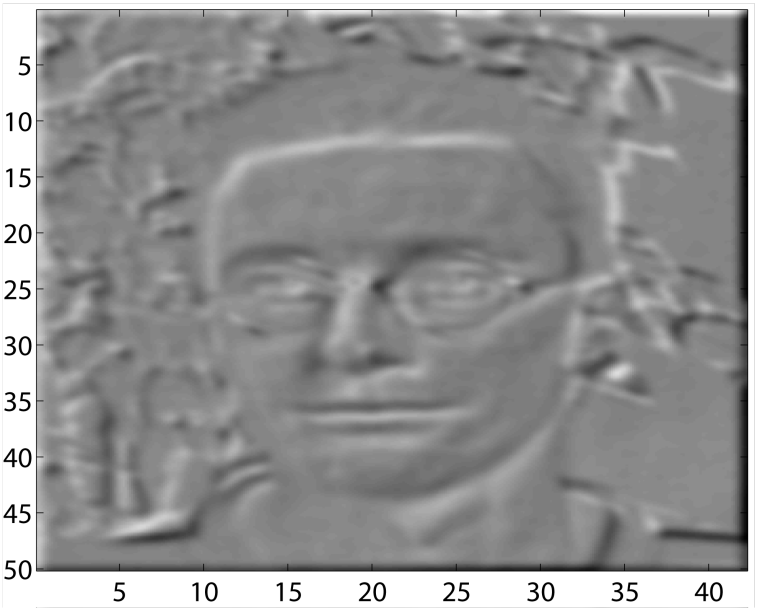


Figure 8.29: The original image convolved with the 45 degree edge extraction kernel. Edges at 45 degrees are emphasized, whereas edges at 135 degrees are suppressed. The picture looks like a flat relief.

Also in visual cortex one finds topologically organized cells that respond to the same feature for all positions of the visual field. But visual cortex extracts many different features and thus one could say that it consists of several colocalized feature maps and just from anatomical clues it is not possible to see which cells belong to which feature maps.

An example for an image processing system that is based on convolutions alternated with some other biologically plausible processing steps has been proposed by Stephen Grossberg etal. [53]: The boundary contour system (BCS). An aVLSI implementation was proposed by Serrano etal. [54]

Chapter 9

Cochleomorphic Circuits

9.1 The Cochlea

The cochlea is a mechanical filter for hydrodynamic waves. Figure 9.1 shows a cross section of the outer ear, middle ear and inner ear (i.e. the cochlea). Sound is channeled by the outer ear. The filter property of the outer ear is direction sensitive, so sound is subtly changed dependent on the direction it comes from. This allows us to judge if a sound comes, for example, from above or behind us. In the middle ear the oscillations of the eardrum get conveyed by the ossicals, i.e. the body's smallest bones, to the oval window. It is believed that not much filtering is going on in this step, just some clipping of loud sounds. Past the oval window the sound waves enter the liquid that fills the cochlea. A traveling wave can be observed along the basilar membrane (cross section in figure 9.2). The stiffness of the basilar membrane is decreasing deeper into the cochlea, therefore it acts as a low pass filter with decreasing cutoff frequency with distance from the oval window. The outer hair cells provide active feedback that causes resonance close to the cutoff. Thus, something like a spectral analysis is performed mechanically in the cochlea, where different frequency components of the input signal are extracted in different positions. The inner hair cells then translate mechanical motion into electrical signals to enervate the ganglion cells that then send action potentials via the auditory nerve to the brain. There is also feedback from the brain to the outer hair cells that is believed to actively modify the resonance properties.

9.2 Silicon Cochlea

In [55] it has been suggested that a 'silicon retina' can be obtained for the use in artificial systems or implants by modeling mechanical and electrical properties of the biological cochlea in aVLSI CMOS electronics. The local mechanical filter properties of a section of the basilar membrane closely match the electrical filter properties of a suggested second order filters (figure 9.4). Cascading these filters with exponentially decreasing time constants one obtains a model of the entire basilar membrane.

The individual second order stage can be described by the differential equation (9.1):

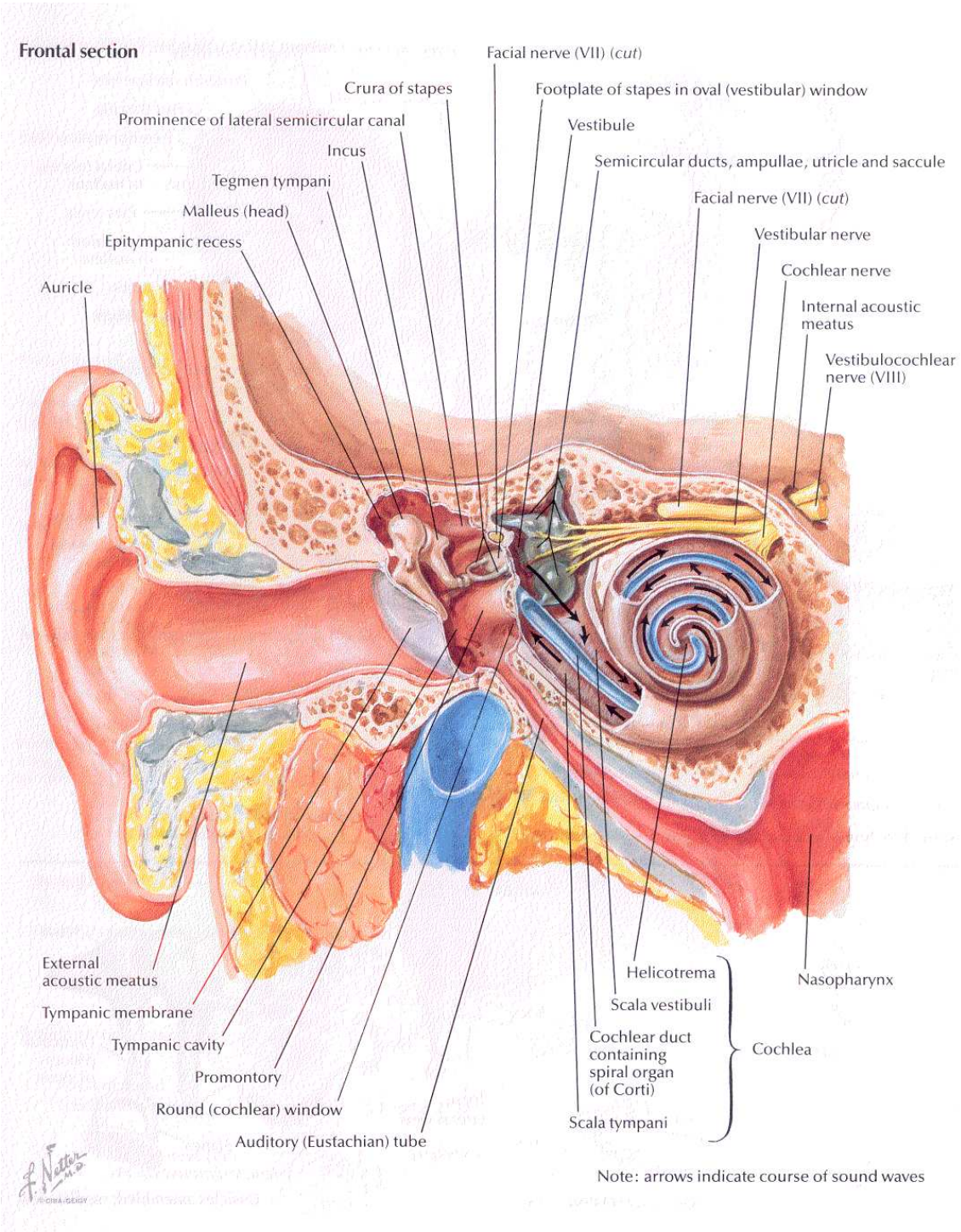


Figure 9.1: Cross section of the ear according to [9]

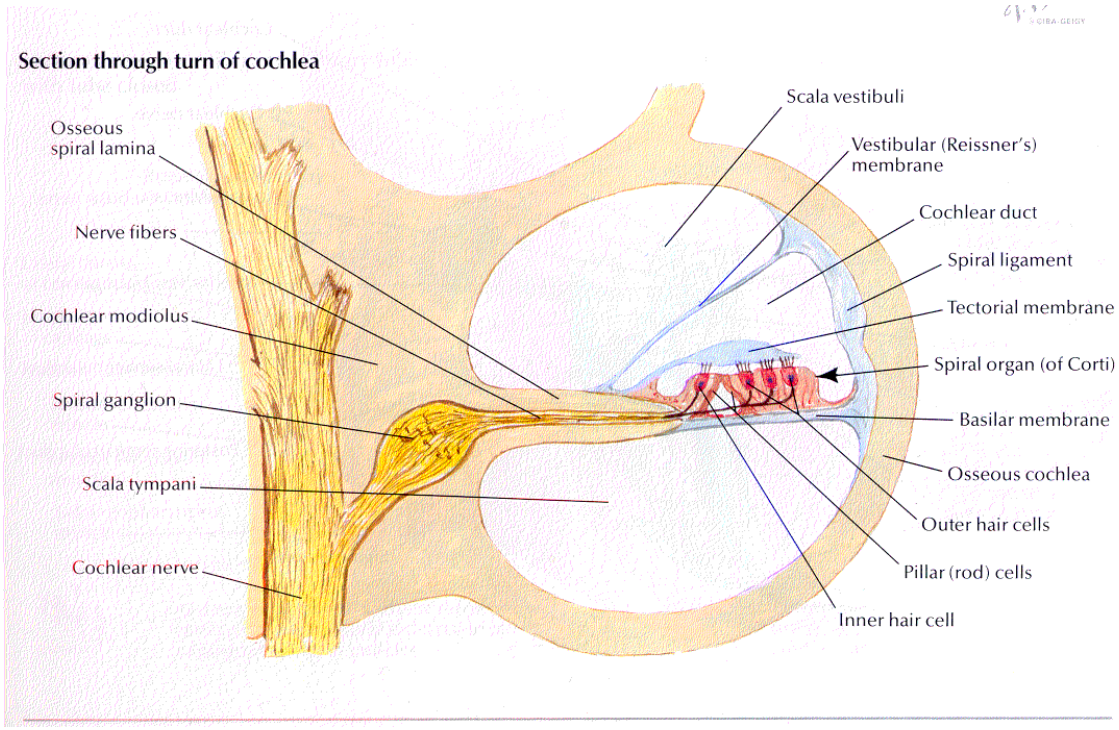


Figure 9.2: Cross section of the cochlea according to [9]

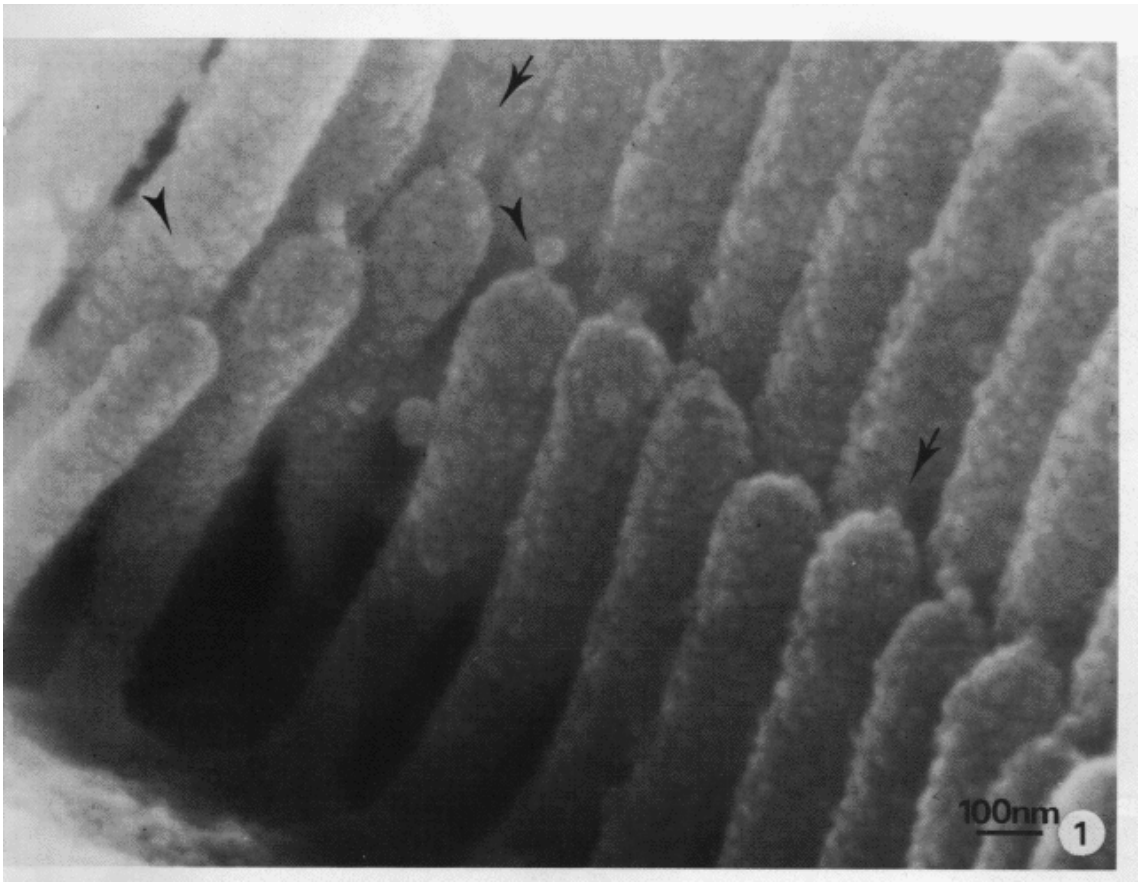


Figure 9.3: An electron microscope image of the hairs of hair cells

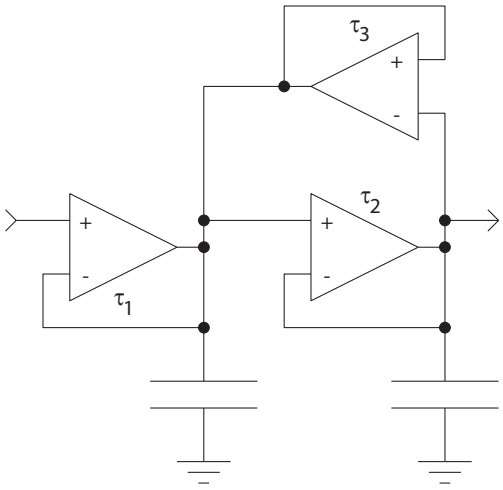


Figure 9.4: A second order filter stage modeling the basilar membrane

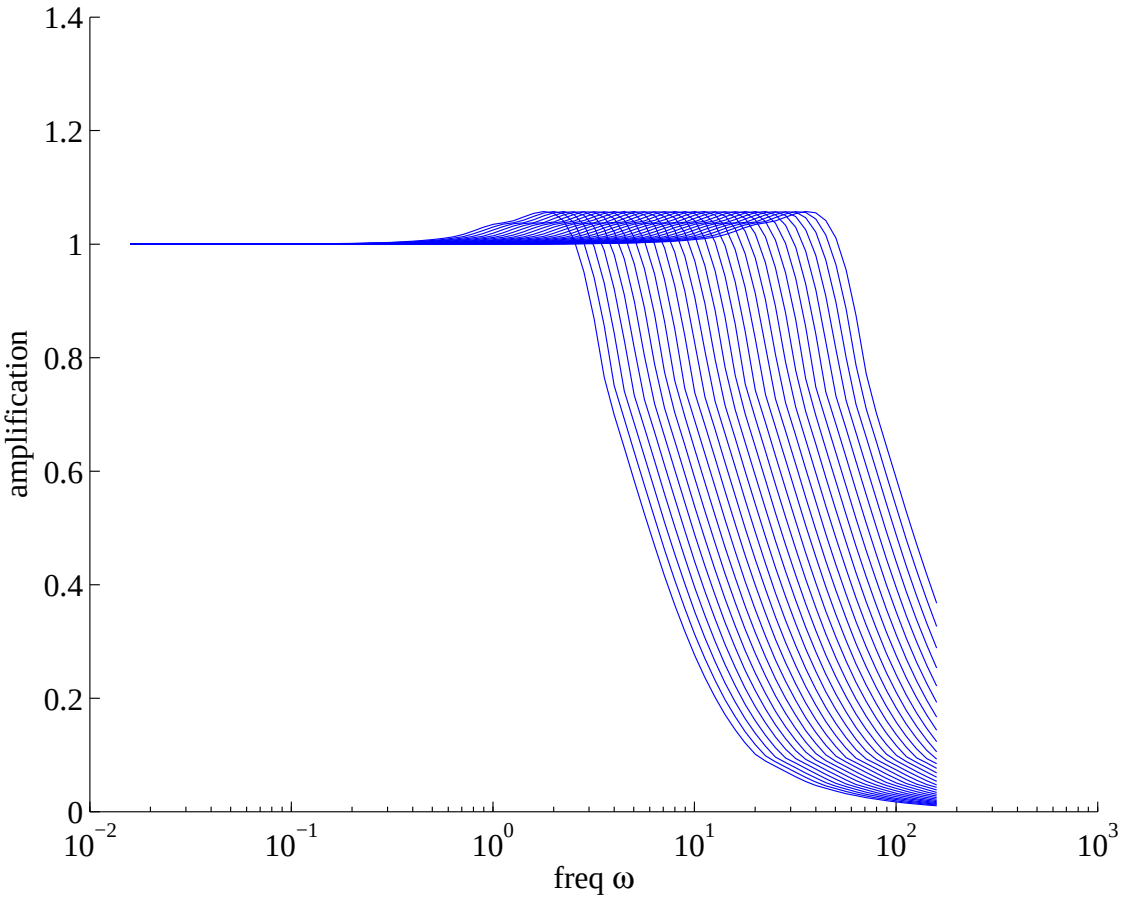


Figure 9.5: Spectrum of resonant second order filter stages tuned with exponentially increasing time constants

$$\ddot{V}_{out} = \frac{1}{\tau_1 \tau_2}(V_{in} - V_{out}) - \left(\frac{1}{\tau_1} + \frac{1}{\tau_2} - \frac{1}{\tau_3}\right)\dot{V}_{out} \tag{9.1}$$

If one takes the Laplace transform to convert it into S-space and a transfer function:

$$H(s) = \frac{V_{out}}{V_{in}} = \frac{\frac{1}{\tau_1 \tau_2}}{s^2 + s\left(\frac{1}{\tau_1} + \frac{1}{\tau_2} - \frac{1}{\tau_3}\right) + \frac{1}{\tau_1 \tau_2}} \tag{9.2}$$

To make analysis a bit simpler, τ_1 and τ_2 can be chosen the same. The spectrum of such individual filters can be seen in figure 9.5. They resonate (amplify) close to their cut off frequency and have a moderate roll off (steepness of the cut off). The resonance can be controlled by τ_3 realtive to τ_2: the longer this time constant is the less resonance.

Here is an attempt of an intuitive explanation of the resonance: the positive feedback through the τ_3 stage is dependent on the difference of the output of the τ_1 and the τ_2 stage. At very low frequencies far away from the cut-off (i.e. virtually in a DC situation), there will be no differnce between those two and ergo no positive feedback. But as one gets nearer the cut-off

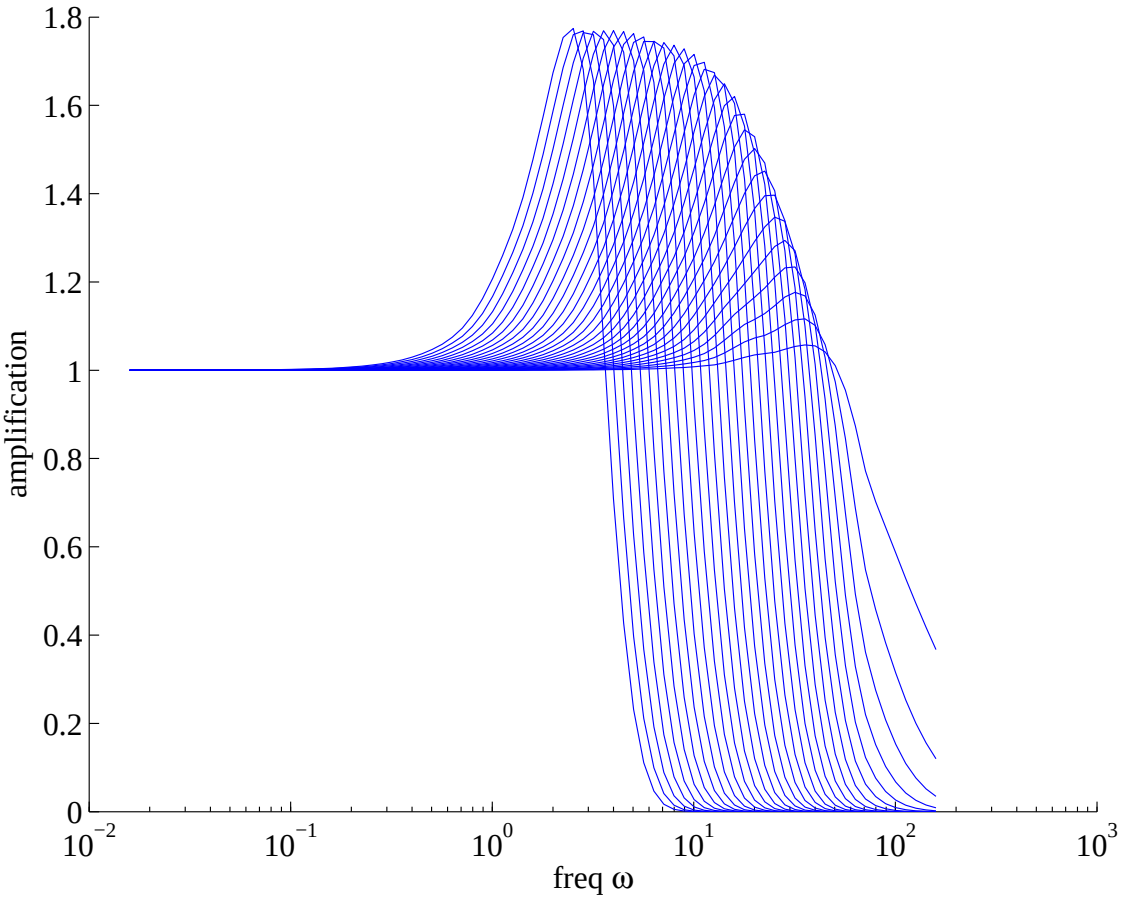


Figure 9.6: Spectrum of resonant second order filter stages tuned with exponentially increasing time constants when put in a cascade

frequency, the phase shift will get closer to $90\,\mathrm{deg}$ and the positive feedback will increase the gain beyond 1.

For the figures 9.5 and 9.6 τ_3 was set to 120% of $\tau_1 = \tau_2$ (i.e. the τ_3 transamp bias current to 83% of the other transamps). The basilar membrane is said to decrease its time constant exponentially with distance from the oval window. So the spacing of the time constant of the displayed filters is also exponential. This is rather easily achieved, as these time constants are inversely proportional with the bias current of the transconductance amplifiers. And those bias currents are themselves exponentially dependent on the bias voltage. Thus, a linear spacing of the bias voltages along the cascade yields the desired result. This can be provided by a series of resistors, for example.

Now by cascading those filters, the filter properties change to the liking of engineers and nature alike: the amplification of the resonance frequency gets bigger (since the amplified frequencies overlap from stage to stage), the specifity to it gets sharper, and the roll off steeper. This is shown in figure 9.6. Accumulating resonance this way is also less threatening to circuit stability.

The authors of [55] also model the inner hair cells (as half wave rectifiers) and the ganglion cells (as integrate and fire neurons). Others have also taken the approach further (as for example in [56]) and different versions of the basic model presented here exist. Since the system closely matches the properties of the Cochlea, it is a good analog front end for all kinds of natural speech processing.

Chapter 10

Neuromorphic Learning

10.1 Neural Learning Algorithms

What is learning? The answer is not so easy. Many equally valid definitions exist. The view of people who work with machine learning might be formulated as this:

Definition: Learning is the search of a parameter space in order to optimize performance.

In this view there is a vocabulary of terms that we are going to use as learning is discussed further:

$\vec{w}$ The vector of parameters that defines the multi-dimensional search space of the learning algorithm. In neural networks these parameters are the synaptic *weights* (i.e. the connection strength between neurons) and they are sometimes also organized in a matrix W, instead of in a vector. An NxN matrix is more convenient to describe all weights between all N neurons in a network. A vector is more convenient if one considers a single vector and all its weights to all inputs, i.e. with which to weigh the 'input vector'.

$\vec{x}$ The (sensory) input to the system (a network or just a single neuron). Also organized as a matrix sometimes, e.g. when the input is an image.

$\vec{y}(\vec{x}, \vec{w})$ The (behavioural) output of the system that changes with learning, i.e. the performance of whom the learning tries to optimize.

$\vec{d}(\vec{x})$ The 'desired' output of the system, a teacher or expert opinion. It is normally not defined for the whole input space spanned by $\vec{x}$. (Thus, the system needs to *generalize* what we learn from a teacher and apply it to unknown situations/inputs $\vec{x}$)

$P(\vec{y})$ A performance evaluation function to judge the quality of the learning state of the system. In general this function can be stochastic. It can also have multiple extrema.

$E(\vec{y}, \vec{d})$ An error function, a special case of a performance evaluation function, that evaluates system performance as compared to the expert/teacher.

μ The learning rate. A parameter that is used by many learning algorithms influencing the learning convergence speed and quality.

10.1.1 An overview of classes of learning algorithms

There are several attributes that have been used to classify learning rules. Maybe at the top level of a classification they can be divided into supervised and unsupervised learning rules. The following list tries to give an example of how that classification could be further refined, listing specific learning rule's names without giving any details. Interested readers may use the names to find relevant references, but for this text we will only eleaborate on a scarce few representatives.

- supervised
  - supervised by expert (learning a target function)
    - continuous target functions
      - gradient descent, Error Backpropagation (for space continuous target functions, interpolation)
      - temporal difference learning (TDλ, for time continuous target functions)
      - statistical methods, interpolation
      - weight perturbation (can also be used in reinforcement)
    - supervised classification
      - supervised LVQ
      - support vector machines
  - reinforcement, supervised by critic (optimizing performance)
    - associative reward-penalty
    - evolutionary algorithms
    - deduction, induction (predicate logic learning)
- unsupervised (optimizing statistics, data reduction/compression)
  - Correlation, Association, Hebbian Learning, Spike based learning
  - Competitive Learning, LVQ, Competitive Hebbian Learning
  - Optimizing data reduction, PCA, ICA

10.1.2 Supervised Learning

The performance measure in supervised and reinforcement learning is dependent on feedback external to the system, i.e. feedback from an expert or critic. These algorithms are mainly based in psychology and mathematics. How supervised learning operates on the level of the neural system is a mystery not yet solved.

10.1.2.1 Example: General Gradient Descent Optimization and the Error Backpropagation Algorithm for ANNs

The possibly most famous supervised learning algorithm, the Error Backpropagation algorithm [57, 58], is applied to multi layer Perceptrons

(MLP, see also chapter 4) which are neural like structures. It is an implementation of **gradient descent** optimization that minimizes an error function. That could for example be the square of the difference of a desired output to the actual net output over the input range where $\vec{d}$ is defined or over a predefined subset of inputs.

$$E = \|\vec{d} - \vec{y}\| \tag{10.1}$$

The weights are initialized randomly, the error is computed, and the derivative of the error with respect to every weight. The weights are then corrected proportional to their derivative and the learning rate μ in the inverse direction of the derivative.

$$\frac{d\vec{w}}{dt} = -\mu \frac{dE}{d\vec{w}} \tag{10.2}$$

If μ is small enough, this will necessarily result in a reduction of the error. This is repeated until the error is acceptable. The name 'Error Backpropagation' is a result of a particularly elegant algorithm to compute these derivatives in a MLP, by retrogradely traversing the neural network (check the literature [57, 58] for details!). The general concept though, is simply that of gradient descent in an optimization problem, that can also be applied to all kinds of other parameterized functions, of which MLPs and ANNs are but one example.

10.1.3 Reinforcement learning

Reinforcement learning is similar to learning with an expert, but there is no desired output $\vec{d}$ defined for any input examples. Thus the performance measure P is not an error function but some judgment on the quality of the output. One quite famous example of this category of learning algorithms is TD(λ) [59, 60]. Genetic algorithms could also fall under this definition.

TD(λ) has very successfully been applied to the game of Backgammon, where a computer program learned to choose moves in any position during the game by the temporally delayed qualitative performance measure of the game being won or lost in the end. [60]. So no expert was available to teach every move but every move was qualitatively assessed after the game was finished as 'probably a good/bad choice', i.e. a classicle example of reinforcement learning.

10.1.4 Unsupervised learning

Unsupervised learning also tries to optimize performance. But this time the performance measure does not depend on direct external feedback. Classically, it tries to optimize encoding/compression of a huge input space to a limited capacity information channel. It tries to get as much relevant information as possible over that channel and adapts according to the input statistics.

10.1.4.1 Hebbian learning

Hebbian learning is THE form of learning that has actually been observed in real neurons. Hebb formulated it as follows [61]:

> When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.

This postulate inspired a number of so called Hebbian learning rules. They contain a positive term of the input to the synapse multiplied with output of the neuron in their weight update rule. Thus, they 'reward' correlations of input and output by a weight increase.

$$\frac{dw_i}{dt} = ...x_i y_i... \tag{10.3}$$

This kind of Hebbian learning changes the weight vector's direction $\frac{\vec{w}}{\|\vec{w}\|}$ towards the direction of the input vector $\frac{\vec{x}}{\|\vec{x}\|}$, i.e. the angle α between the weight vector and the input vector is reduced. If the neuronal model is that of a perceptron, then the output is defined as:

$$y = f(\vec{w}^T\vec{x}) = f(\cos\alpha\|\vec{w}\|\|\vec{x}\|) \tag{10.4}$$

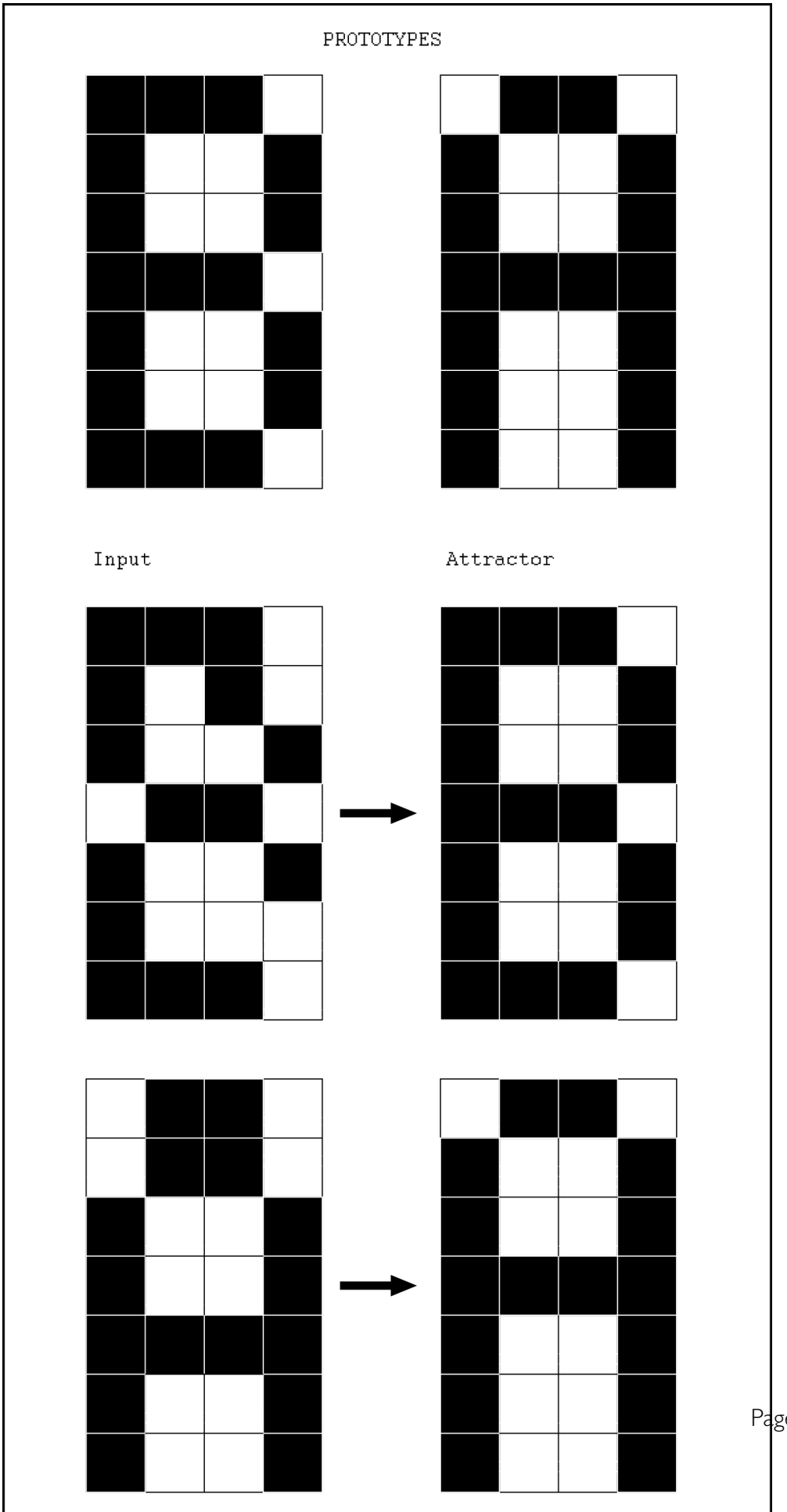
Where f is normally a monotonically increasing function. Which means that the response to that particular input is increased by the learning.

10.1.4.2 Associative memory

Hebbian learning has for example been used to 'train' (it's actually rather programming) **associative memory** or **attractor neural networks** or **Hoppfield Tank networks**. These are neural networks that are used for cateforization of patterns (e.g. letters of an alphabet). They are, thus, normally arranged in a two dimensional topology: neurons can conveniently be thought of as pixels in a two-dimensional image. Every neuron receives one individual sensory input. The input to the entire net can therefore again be thought of as an image. All neurons are 'horizontally', bidirectionally connected to all other neurons, or a random selection of other neurons, or a neighbourhood pattern (The later is called Cellular Neural Network (CNN).)

In a training phase prototype images that are exemplar for one pattern category (e.g. a standard letter 'A', a standard letter 'B', etc.) are presented to the neuron's inputs (figure 10.1, top). Hebbian learning strengthens the connections between neurons that are active in the same image and weakens or makes inhibitory the connections between neuron pairs where one is active and the other is not. In this manner, each prototype pattern 'imprints' itself into the network structure:

$$\frac{dw_{i,j}}{dt} = \frac{1}{t}(-w_{i,j} + x_j y_i) \tag{10.5}$$


PROTOTYPES
Input
Attractor

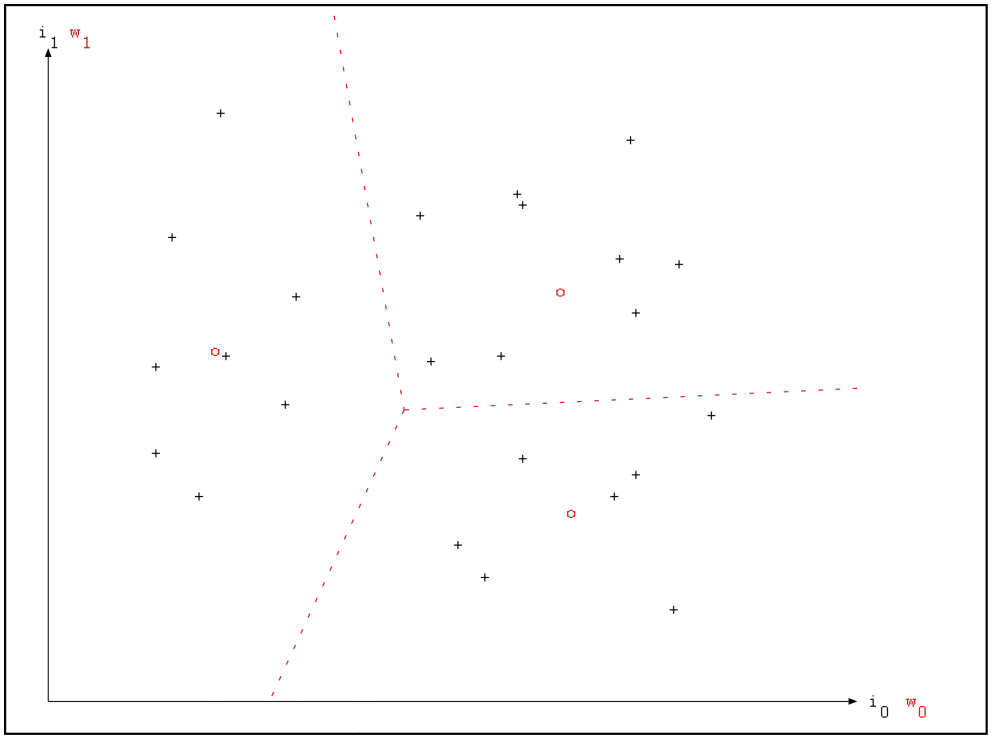


Figure 10.2: Classification with LVQ

where t is the number of input patterns that have been presented so far. (This learning rule works with binary (0/1) model neurons with threshold 0.5).

When the learning is completed and all input images 'stored' one can briefly present other input patterns to the net (i.e. set them as initial state of the neurons) and let it run freely afterwards, driven only by the recursive connections between the neurons. It turns out that the net will approach one of the stored input patterns which have become stable states of the recursive activity in the network (figure 10.1, middle and bottom). Like this input patterns that are close to a particular stored pattern, are 'associated' with that pattern. Letter recognition is a typical task applied to that kind of network.

The presence of such associative memory in the brain is debatable. A main argument against it are that it requires time to converge and usually we are really fast in recognizing objects. It is also not quite clear how to use the associated pattern on a next level of computation. It would need to be read or interpreted somehow.

10.1.4.3 Competitive learning

Competitive learning is an important concept that can also be used for fuzzy pattern recognition. In contrast to associative memory the recovery of a pattern is not expressed by the output of the whole neuron assembly but strong activity of individual neurons will represent the presence of a specific input patterns. That means in other words that each neuron reacts to a specific chunk of the input space. The basic idea is that neurons adapt

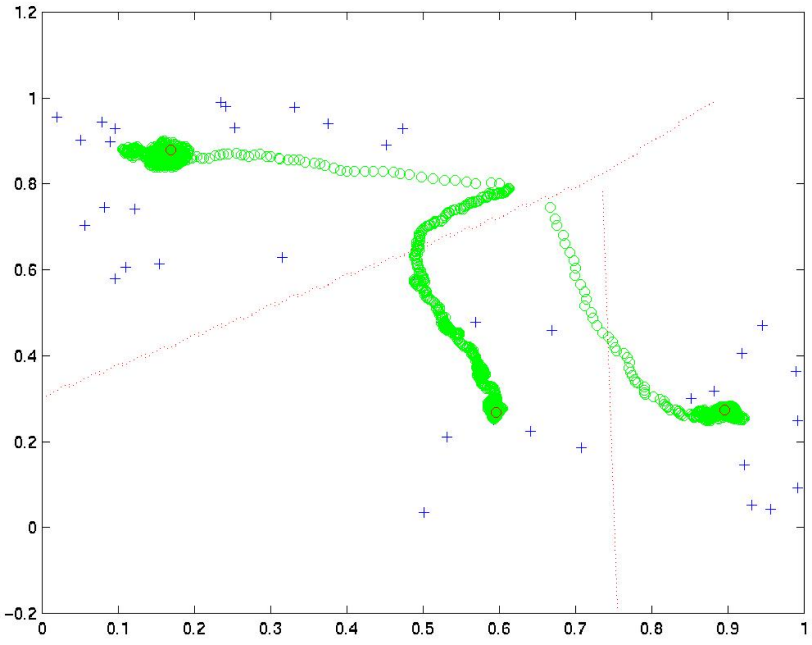


Figure 10.3: Dynamics in Learning Vector Quantization

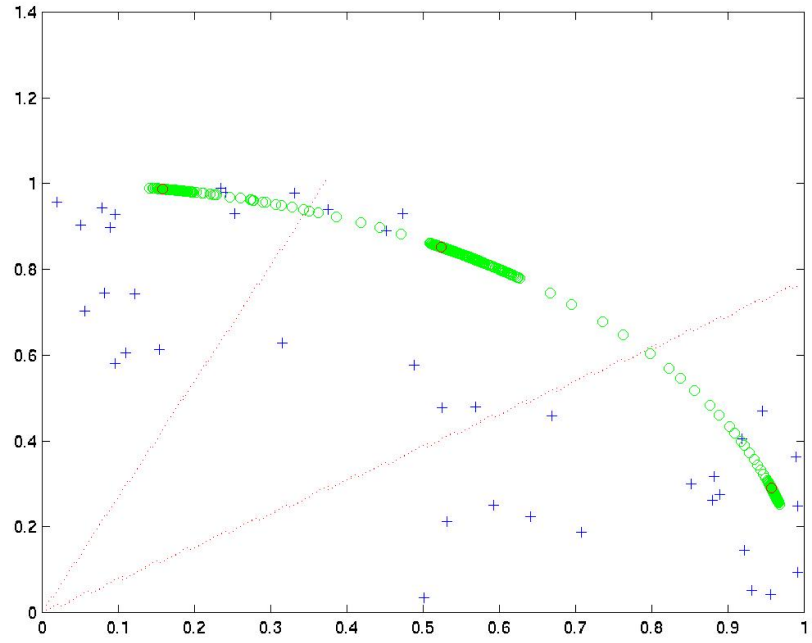


Figure 10.4: Dynamics in competitive Hebbian learning with Oja's learning rule

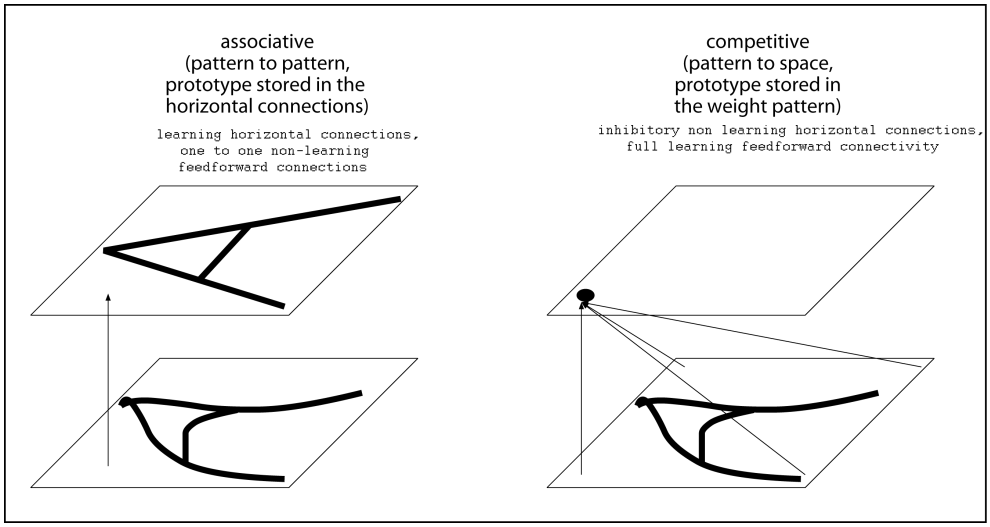


Figure 10.5: An illustration on the differences of competitive learning an associative memory.

to the present input in a way that the output gets increased for that present input. They do adapt to that input with different speeds such that that neuron that has the strongest output already adapts the most. Actually in many variants of competitive learning ONLY the strongest neuron is allowed to further adapt. This can be achieved by making the degree of adaption proportional to the neurons output. (This is for example the case for classical Hebbian learning.) And if the neurons are connected to form a WTA network, then indeed only the winner adapts to the present input.

One classic example that will be used to illustrate competitive learning in the following is 'Learning Vector Quantisation' (LVQ).

Learning Vector Quantisation (LVQ)

We want a system to observe a certain input space. (That could for example be output of a number of sensors, e.g. a camera.) Certain inputs should trigger certain responses. An easy way of achieving this would be to store an action for every possible input. But that most often would exceed the storage capacity of a system, most certainly so, if the input space is infinite. The solution is to 'quantize' the input space by **classification**: One finds rules that project the inputs into a number of classes. And the number of classes is chosen small enough that the appropriate action for each class of inputs can be stored.

One way to classify vectors of real numbers are competitive neurons or winner take all (WTA) neurons. Winner take all neurons compete among each other in a network and eventually only one neuron remains active. The basis for a WTA network can be almost any neuronal model. All the neurons in the network receive the same feed-forward input, i.e. they share the same input space. Now in addition to that basis model, every neuron inhibits every other neuron in the network. If the inhibition that a neuron exerts on its fellow neurons is equal to its activity, the only stable state of the network is when only the strongest neuron remains active. Every neuron represents one class of inputs.

A particular kind of neurons are abstract neurons whose activity is inversely proportional to the distance of the input vector to the weight vector.

$$y = \|\vec{w} - \vec{x}\|^{-1} \tag{10.6}$$

Other distance measures than the Euklidian one can also be used and the choice depends on the particular application. If you use such neurons as WTA neurons, then the one whose weight vector is closest to the input vector wins. In the figure 10.2 it is illustrated how such neurons quantize an input space.

We humans too approach our daily live with subdividing the perceived world into digestible chunks. We deal with 'similar' situations in the same way and what we do think of as similar is actually determined by our own kind of 'distance function'.

Given a particular distance function that determines the activities of WTA neurons a quantization can actually be learnt by the neurons by unsupervised learning. This makes sense as a first data reduction stage in a more extensive learning system. Unsupervised learning does not know about any action that could be an appropriate response to a certain class of inputs. It classifies the inputs purely by statistical criterions, e.g. it tries that the classes give a good/optimal representation of all the input examples that it experiences during learning.

The basic idea is quite simple. The weight vectors are randomly initialized at first. Then the networks begins to watch the input space. The winner who is active at any one time corrects its weight-vector in the direction of the input vector. For example

$$\frac{d\vec{w}}{dt} = \mu y(\vec{x} - \vec{w}) \tag{10.7}$$

An example of how the weight vectors move is shown in figure 10.3. It is an example of three neurons with two inputs that they all share. It is convenient to draw the inputs and the weight vector into the same coordinate system. All the inputs are the crosses. They appear repeatedly in random order. The green circles are the weight vectors of the three neurons, drawn repeatedly during learning. They start close to each other and then start to move towards clusters in the input distribution. Actually there are two intentionally distinct clusters in the lower left and upper right. One neuron becomes responsible for the upper left cluster. The other two neurons define their own subclusters within the lower right cluster. The dashed lines are the separation lines of the three clusters after learning is completed.

Competitive Hebbian Learning

Hebbian learning in a competitive network with (more plausible) linear perceptrons can be used to the same ends as LVQ. The difference is that the input space is now divided along rays from the coordinate system origin. The relative length of the weight vectors and the difference in their angle

determines how wide that chunk is. Figure 10.4 illustrates this division and the dynamics of the learning process with one particular Hebbian learning rule:

$$\frac{d\vec{w}}{dt} = y(\mu_0\vec{x} - \mu_1 y\vec{w}) \tag{10.8}$$

This rule is known as Oja's learning rule [62]. It is a Hebbian learning rule that keeps the length of the weight vectors constant (on condition that the neuron is a perceptron with the identity as the activation function). That's why the weight vectors, marked as green circles, only move along a circle around the origin. The inputs (the crosses) are the same as in the LVQ learning example in figure 10.3. The separation of the space by the neurons happens now according to a different rule, along separation lines through the coordinate space origin. That is because the neurons' outputs are defined according to equation 10.4, where for a given input vector $\vec{x}$ and a constant weight vector length $\|\vec{w}\|$ the output only depends on the angle α between input and weight vector. Thus, the neuron with its weight vector at the smallest angle to the input vector wins.

The input to a ANN trained by competitive learning, like for associative memory, can for example be an image. But unlike in associative memory, every neuron receives input from the entire picture, not just from one pixel. And the stored prototypes would not be represented in the inter neuron weight pattern, but one individual neuron would store a prototype image in its weight pattern and be responsible for that particular image. Figure 10.5 is an attempt at illustrating those distinguishing properties of the two approaches.

The representation of input patterns by individual neurons has also been observed in recordings in visual cortex and seems, thus, a more plausible neurophysiological model.

10.1.4.4 Spike based learning

Spike based learning is also known as spike timing dependent plasticity (STDP) (although some researchers associate STDP more exclusively with rules that claim a somehow higher degree of faithfulness with the neurophysiological observations whereas they regard spike based learning as a more general term). All the learning rules we have been talking about until now can work on Perceptron-like neurons, either with a continuous or binary input and output. But lately there has been a lot of talk about so called spike based learning rules. Those operate on neurons that receive and send spike signals in a manner that does not allow for them to be simplified to average frequency signals. To handle such signals in differential equations they are often expressed as sums of Dirac delta functions:

$$\begin{aligned} x_i &= \sum_k \delta(t - t_{i,k}) \\ y &= \sum_s \delta(t - t_s) \end{aligned} \tag{10.9}$$

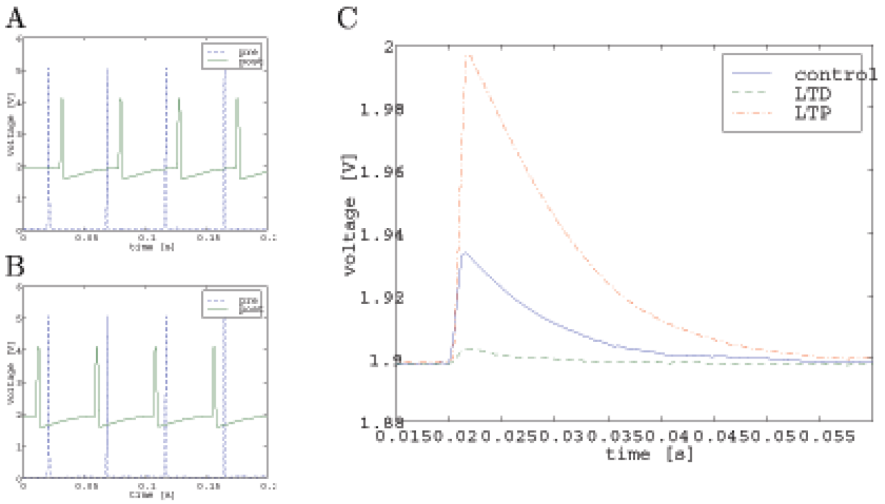


Figure 10.6: Spike based learning behaviour implemented into a silicon neuron

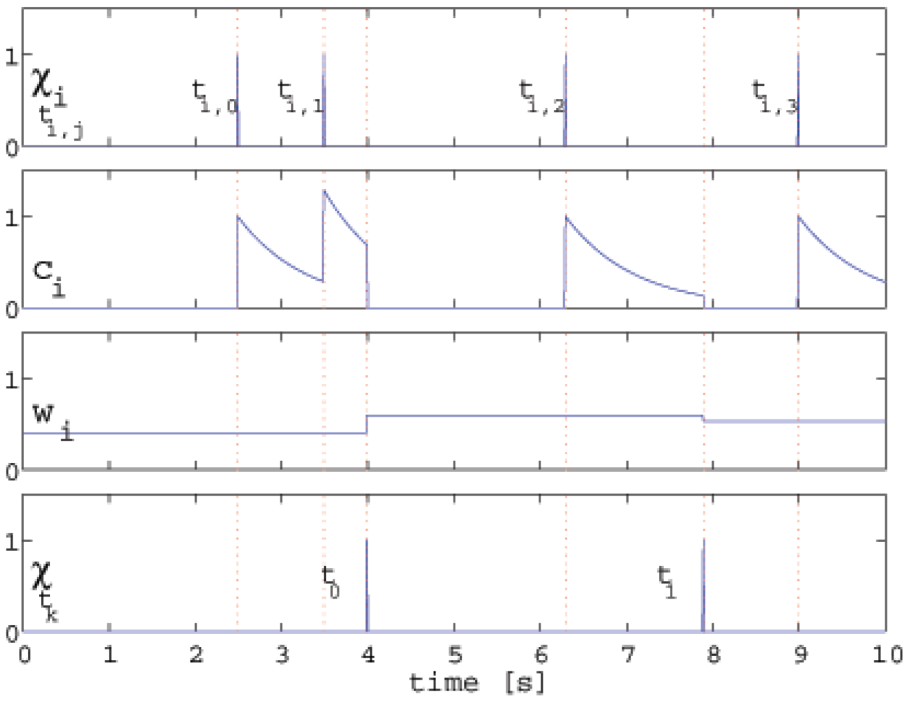


Figure 10.7: Simulation variables involved in a particular spike based learning algorithm at one synapse.

Such spike based learning behaviour has been observed in physiological experiments [22] and it has been suggested that it can be used in the manner of Hebbian learning to correlate spikes (instead of frequency signals), although concrete practical applications are still sparse. Normally spike based learning rules increase weights of synapses whose spike input precedes an output-action potential.

Figure 10.6 shows spike based learning in a silicon neuron. On the left hand (A and B) the input and the output are depicted. The output looks more like a biological action potential with hyperpolarisation (overshoot) after the pulse. The inputs are the narrow pulses. On the right (C) the EPSPs (electrical postsynaptic potential) that are caused on the cell membrane voltage by a single spike input to the cell are shown. The middle trace is an EPSP before learning. The top trace is an EPSP after the stimulation pattern in figure A has been applied for 50 pulses. And finally the lower trace is the result of the stimulation pattern in figure B.

It is not yet very clear how nature uses such a spike based learning behaviour. It could for example be used like Hebbian learning with a very short time constant. Such that single spikes within a certain short time window are considered correlated. Similar properties like in 'normal' Hebbian learning can be achieved by this, only that the variables are average frequencies no longer but spike occurrences. The time delay between the incoming and outgoing spike however makes it hard to have symmetric bidirectional connections. It seems rather that in case of bidirectional connections between neurons one of the directions will be suppressed.

It has thus been suggested, that such learning rules do not simply correlate events (expressed by the activity of a neuron) that are constantly coincident in time and thus very likely have a common cause, but that they rather correlate events that follow each other, i.e. events that are connected in a causal chain. Thus, this can teach a neuron to respond sooner in this causal chain of events, i.e. to use early observations to predict that some particular event *will* happen [63].

One learning rule that leads to such behaviour is the one I described in my thesis [64]. There is a variable that we called 'correlation signal' present in every synapse ($\vec{c}$). It gets increased by presynaptic pulses and decays over time. It's magnitude is thus a measure of resent presynaptic activity (see figure 10.7):

$$\frac{d\vec{c}}{dt} = \vec{x} - \frac{1}{\tau}c \tag{10.10}$$

Note that in addition $\vec{c}$ is reset with every output spike.Furthermore, the elements of $\vec{x}$ are Dirac delta-functions which causes the increments by one of the elements of $\vec{c}$.

The weight is then updated with every postsynaptic pulse, the change dependent on the maggnitude of the correlation signal (Note that also y is a sum of Dirac delta functions causing discontinuous increments/decrements of $\vec{c}$):

$$\frac{d\vec{w}}{dt} = (\mu_0 \vec{c} - \mu_1 \vec{w}) y \tag{10.11}$$

In figure 10.7, the weight changing behaviour of a single synapse is shown. Presynaptic spikes are depicted on the top axis, postsynaptic ones on the bottom. c evolves as described in the text and is reset by postsynaptic APs and w according to (10.11).

10.2 Analogue Storage

We defined learning as the search of a parameter space. If a learning algorithm should be implemented in aVLSI, this parameter space and the state of the search needs to be represented somehow. In most learning algorithms that parameter space is continuous and could be represented by a vector of real numbers. How can such analog parameters be represented and stored in aVLSI? Several methods have been explored:

- volatile short term storage
  - capacitive storage/dynamic storage
- volatile long term storage
  - digital storage and DAC
  - capacitive storage with refresh
  - multi level flipflops
- non-volatile storage
  - floating gates, analog flash ROM

10.2.1 Dynamic Analogue Storage

The problem of analog storage on a aVLSI chip is simple if one has no big requirements on the duration of the retention. Simple capacitances that are connected to transistors for addition and removal of charge, can hold analog voltages with a leakage rate in the order of millivolts per second.

Pro
- Real analog
- Easy to write
- As gate readable uninvasively by current through a transistor
- Compact

Contra
- Only short term, leaking some mV per second

10.2.2 Static Analogue Storage

Static storage as well as dynamic storage is volatile storage. Volatile storage needs a power supply to maintain its state. In static storage that state is maintained through some sort of internal feedback. The classical digital example is the flip-flop. Positive feedback drives it to one of two rails. But

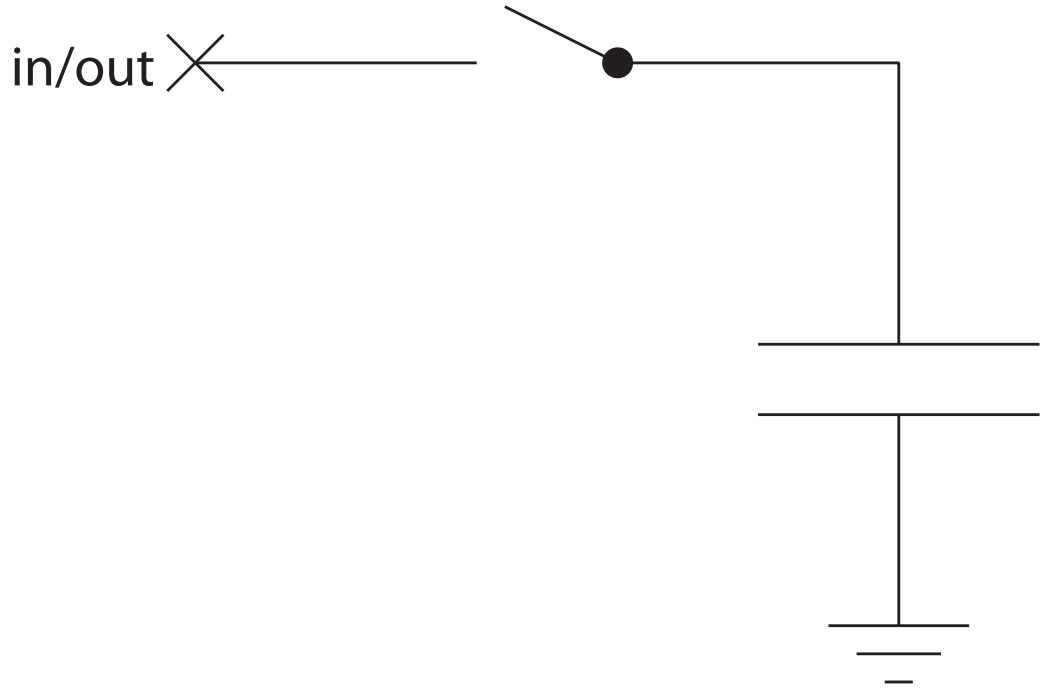


Figure 10.8: Capacitive dynamic analog storage

there is up to now no fully analog static memory available. As alternative quantized (multi-level) storage is used, with a limited resolution. This can for instance be achieved by the combination of digital memory and analog to digital / digital to analog (AD/DA) conversion.

A AD/DA conversion memory cell has been used in a learning context in [65]. The storage cell that they used is described in figure 10.9. While 'restore' is high, the stored current is mirrored to the synaptic current I_{syn}. When the signal 'restore' goes low and the signals C_i are applied in the sequence at the bottom of the graph, the current I_w is stored into the memory cell by means of an successive approximation register (SAR) ADC variant. The bit blocks have a bias current that is doubled from left to right. Thus the first cell to receive a low control pulse C_2 has the biggest bias current, i.e. $4I_b$. The current I_w (that should not exceed $8I_b$) is compared in it to that bias current. The result of that comparison sets the latch (the feedback double inverters on top) in that 1 bit memory block. As C_2 goes high again, this 1 bit block now supplies the current $4I_b$ if $I_w > 4I_b$, otherwise it supplies no current. Thus, as the cell C_1 is now coupled to the common line, the current drawn from it is I_w modulo $4I_b$. The process is repeated until the last cell is reached and all 1 bit blocks together now supply I_w with a precision of I_b.

Pro
- Easy to write
- Uninvasive direct reading

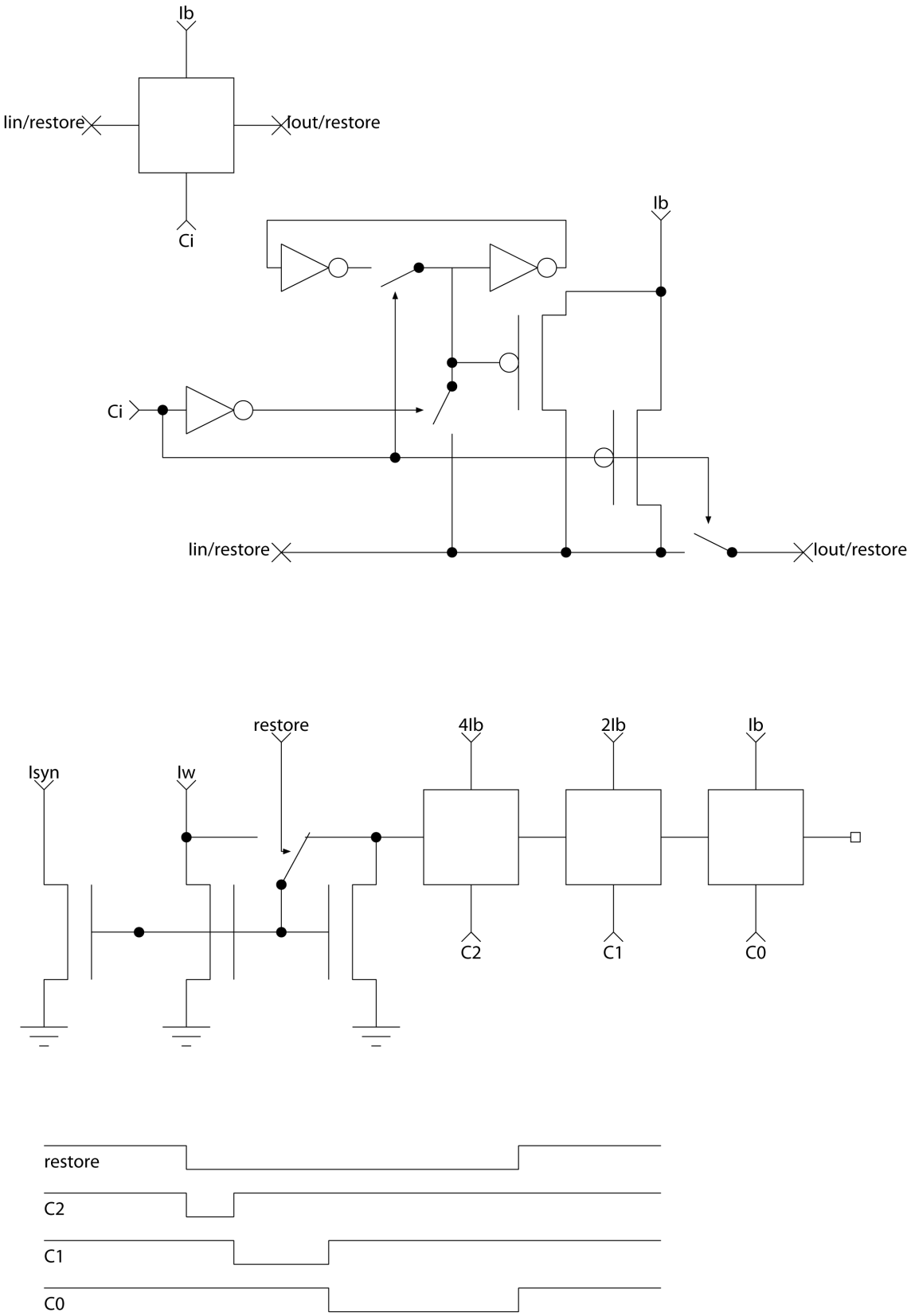


Figure 10.9: An implementation of AD/DA multi-level static storage

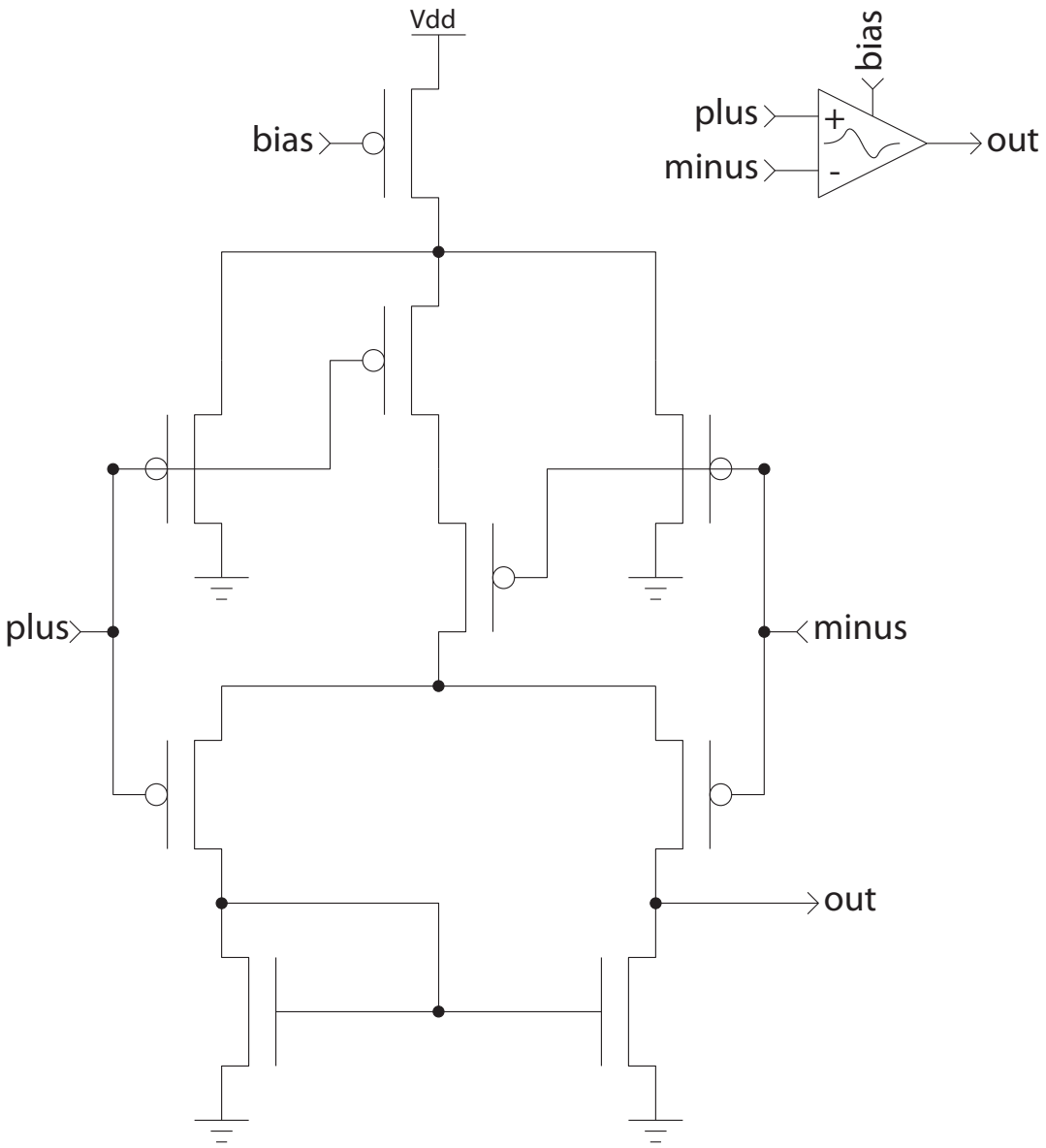


Figure 10.10: A 'fusing' amplifier, that turns off if the inputs are too far apart

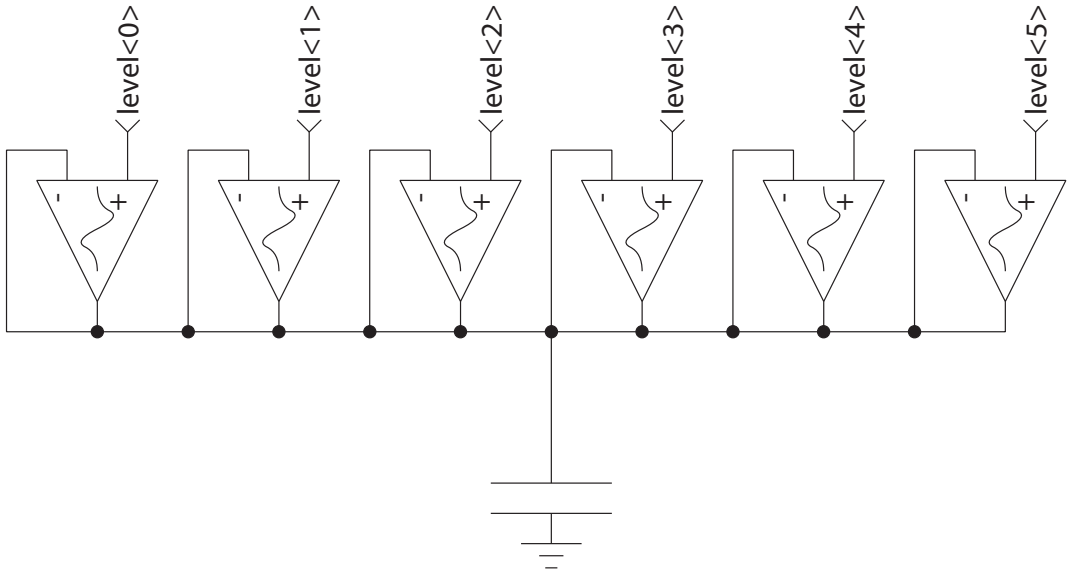


Figure 10.11: An example of real multi-level static storage: weak multi-level static memory

Contra
- Only multi-level
- Digital control circuitry and signals

But also real multi-level static memory is possible, i.e. with no internal binary representation. This has been introduced as weak multi-level static memory in [66, 67].

An important building block is the so called 'fusing' amplifier (figure 10.10). It combines a transconductance amplifier with a so called 'bumb circuit'. The bumb circuit supplies a bias current to the transamp only if the two inputs are close together. The fusing amplifier, thus, operates like a transconduction amplifier only when the two input voltages are close enough together. If they are too far apart, the output current will turn off. If this circuit is connected as a follower, then it will drive the output towards the input voltage, if they are close already, otherwise it will not influence the output voltage.

If the outputs of an array of those fusing followers with different input voltages are coupled together (figure 10.11), this output will always be driven towards the closest of those input voltages. This results in a multi-level memory cell, very similar in principle to the binary double inverter flip-flop. If the bias currents of the fusing followers are weak and a capacitance is coupled to the output, this capacitance can be used as a fully analog capacitive dynamic storage cell for short term storage. Only on a long time scale will its content be discretized by the weak driving force of the fusing followers.

Pro
- Easy to write
- As gate readable uninvasively by current through a transistor

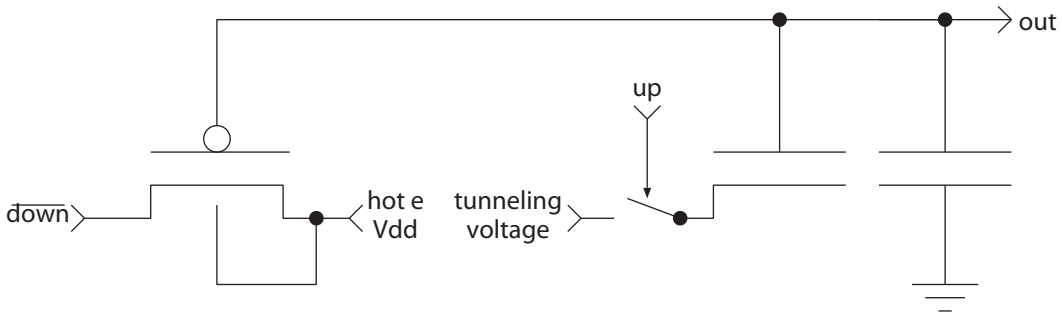


Figure 10.12: Real analog and non-volatile: Storage on a floating gate (Flash ROM, EEPROM)

- 'Real analog' on a short time scale
- No digital control signals necessary

Contra

- Only multi-level on a long time scale
- Space consuming (linear increase with number of levels)

10.2.3 Non-Volatile Analogue Storage

In many aspects idealized non volatile analog storage is the most desirable way of analog storage for many applications. In reality the techniques involved are tricky and sometimes unpractical. For non-volatile storage in a electronics context, floating gate and magnetic storage have been used in the digital world. Floating gates (FG, as used digitally in Flash ROMs or EEPROMs) are most promising to the neuromorphic engineer, since they are devices that can be implemented on a CMOS chip, right beside the processing circuitry. They have been used for analogue storage also [68, 69, 70]. FGs are conductors with no physical connection to any other conductors . Charge that is deposited on a FG will remain there indefinitely in an ideal world and for years in real circuits. Still charge can be added or removed across the energy barrier of about 3.2eV posed by the insulation oxide around a CMOS polysilicon conductor by several techniques (figure 10.12).

Fowler Nordheim tunneling [71] Tunneling is caused by the electron's quantum mechanic property, sometimes to be were it is not supposed to be. By increasing the voltage across the insulation, the distance from where the electrons are supposed to be to the point by which they have enough energy to be in the conductive band is reduced (figure 10.13). This increases the probability of the tunneling. It has been modeled by the following equation:

$$I_g = I_{T0} e^{-\frac{V_f}{V_{ox}}} \tag{10.12}$$

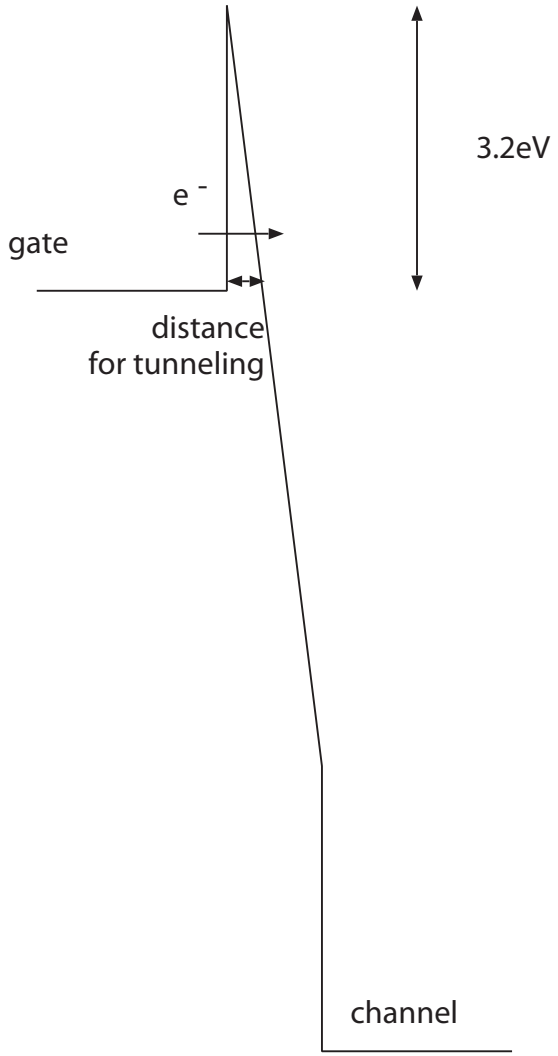


Figure 10.13: Band diagram for tunneling through the gate oxide

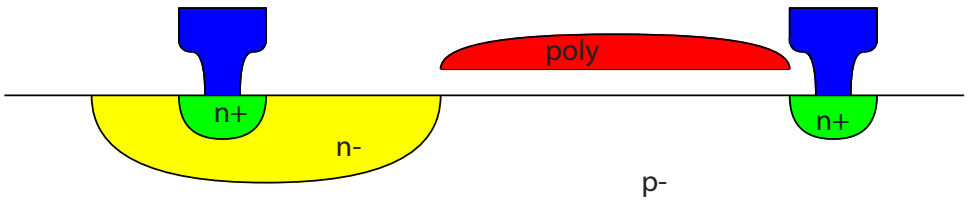


Figure 10.14: High voltage NFET

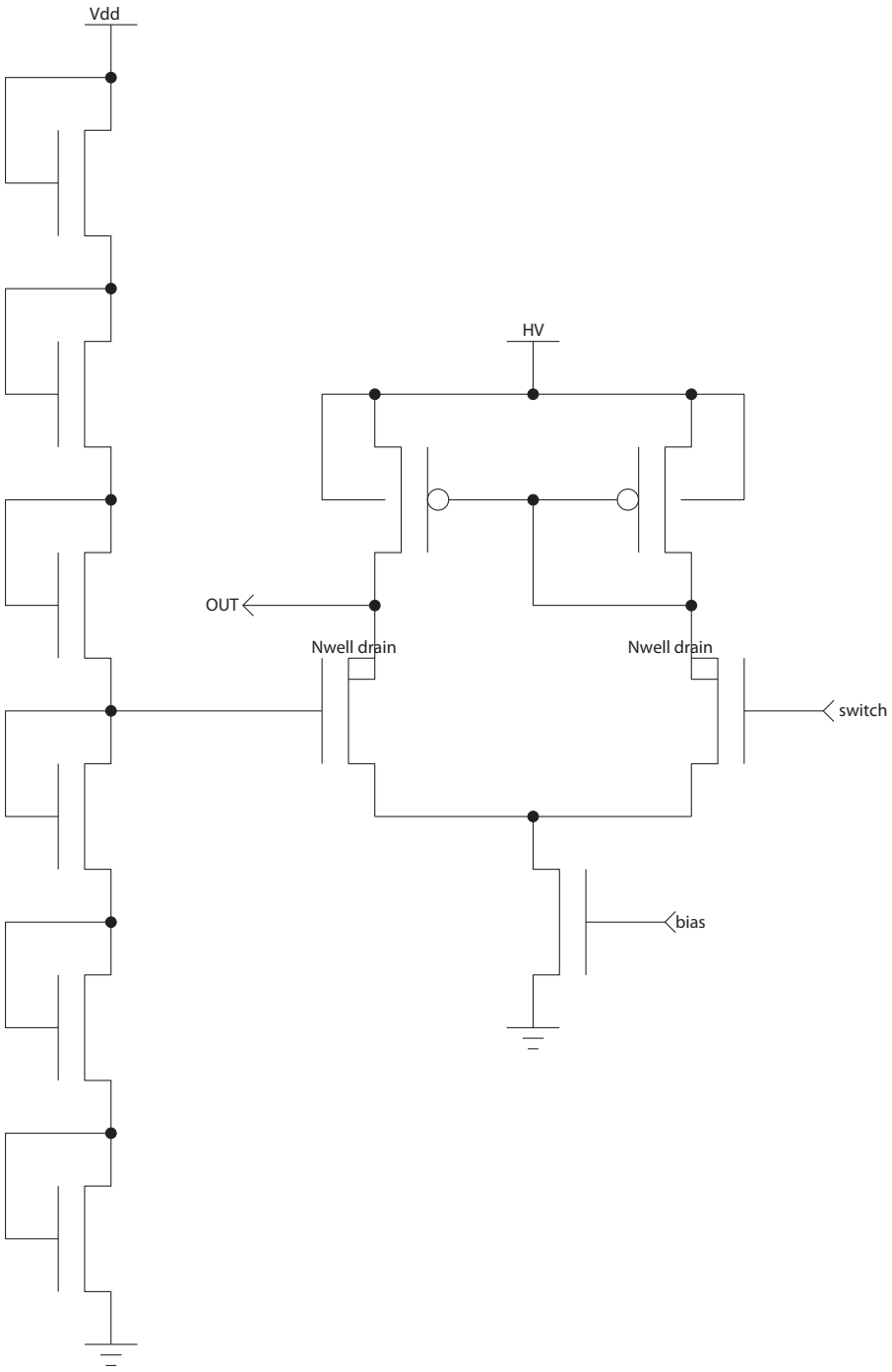


Figure 10.15: On chip high voltage switch

In a 2.0μm Orbit CMOS process, for example, the parameters were I_{T0}=0.0818A and V_f=984V.

Hot electron injection For hot electron injection, electrons must have a lot of kinetic energy (>3.2eV), i.e. they are hot. This can be achieved by the intense electric field between the channel and the drain of a CMOS transistor. The model for the hot electron injection current is:

$$I_g = I_S \beta e^{\frac{V_{DC}}{V_{inj}}} \tag{10.13}$$

Again in 2.0μm Orbit CMOS, β=1.17e-21 and V_{inj}=0.10V .

Tunneling is used in digital Flash ROMs or EEPROMs. Specialized processes with thin gate oxide make tunneling possible with voltages close to the supply voltage. In the 0.6 AMS process that we used for this in the past, about 15 V are required. Hot electron injection is usually an undesired effect, often present and sometimes bothersome in submicron processes. It can be put to work however in this context. Another method to change the charge on a floating gate besides tunneling and hot electron injection is UV-light.

A difficulty in controlling tunneling in standard CMOS processes that require voltages beyond the supply voltage is the switching of these high voltages. One can employ NFETs with lower doping in the drain. These transistors are capable to stand a high source-drain voltage. A cross section of such a transistor is shown in figure 10.14. The break through voltage of this HV-NFET is very high. In the 0.6 AMS process somewhere above 30V. A switch that switches its output between a high voltage and a voltage that is determined by the break through voltage of the normal PFETs is shown in figure 10.15. Note that the left PFET of the amplifier here is exposed to a high voltage when the output is switched low and it is actually breaking through. But since the current is limited in this situation, they do not get damaged by this. As a consequence, the output will not be switched all the way to ground but only by the break through voltage of the PFET. Fortunately, since the tunneling currnt is quasi exponentially dependent on the tunneling voltage, this switching voltage is quite sufficient and serves its purpose well.

Pro
- Real analog
- As gate readable uninvasively by current through a transistor
- Rather compact
- Stores value even without power supply

Contra
- Difficult to write
- Tunneling and hot carrier injection efficacy changes with use. Limited lifetime (write cycles in flash ROMs)
- Leaky in more advanced processes (thinner gate oxide < 3nm)

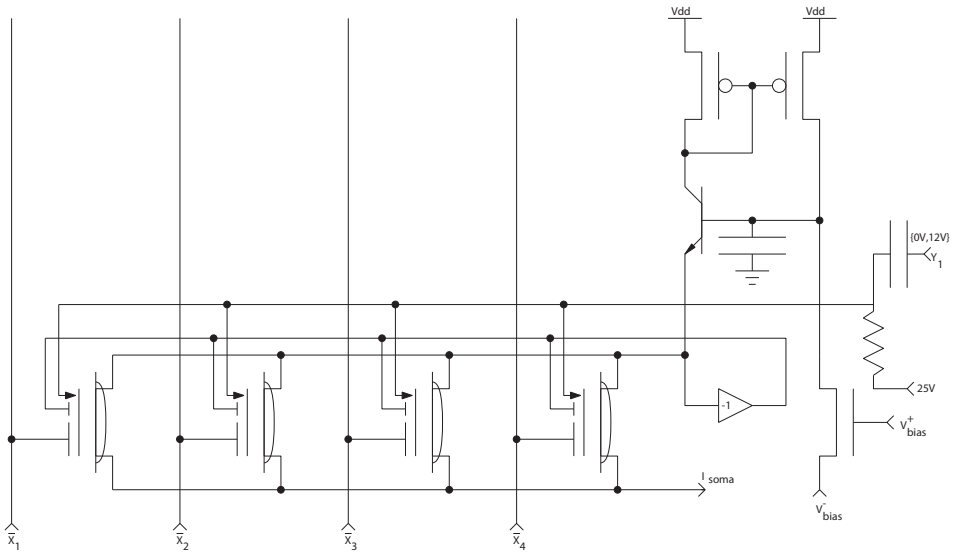


Figure 10.16: An explicitly weight normalizing circuit with off chip controlled learning behaviour [72, 73]

10.3 Neuromorphic Learning Circuits

The most prominent learning algorithm in the context of simulated artificial neural networks, is the Error Backpropagation algorithm (gradient descent) used with multi layer Perceptrons (MLP). However, since it is a mathematical model, it does not lend itself to direct implementation into aVLSI. It is prone to all the weaknesses of realizing simple mathematics in analog hardware: Non-linearities, offsets, bounded range, etc. The result: simple maths, complicated circuits.

In contrast, neuromorphic engineers try to use the building blocks of aVLSI (instead of maths) to define their model. The resulting circuits are thus comparatively simple, whereas a precise mathematical description of them would be complex. Still, qualitatively they are mostly Hebbian learning circuits, as this is a learning behaviour that is actually based in neurophysiological observations.

10.3.1 Hebbian learning circuits

An elegant weight normalizing learning circuit in aVLSI has been introduced by Chris Diorio [72, 73]. His group uses floating gate transistors to implement very compact learning systems with non-volatile analog storage (figure 10.16).

This implementation is strange in some ways. The learning rule itself is not part of the circuit: the signals controlling the learning are independent input signals that are superimposed on the input signals. The synapse input (X) is active high and can be analog continuous. One half of the learning input is active low pulses superimposed on this input signal. If an active low X learning input (lowering the gate voltage) meets with a Y active high pulse (increasing the tunnel voltage), tunneling to that particular floating gate is induced (because the difference between floating gate and

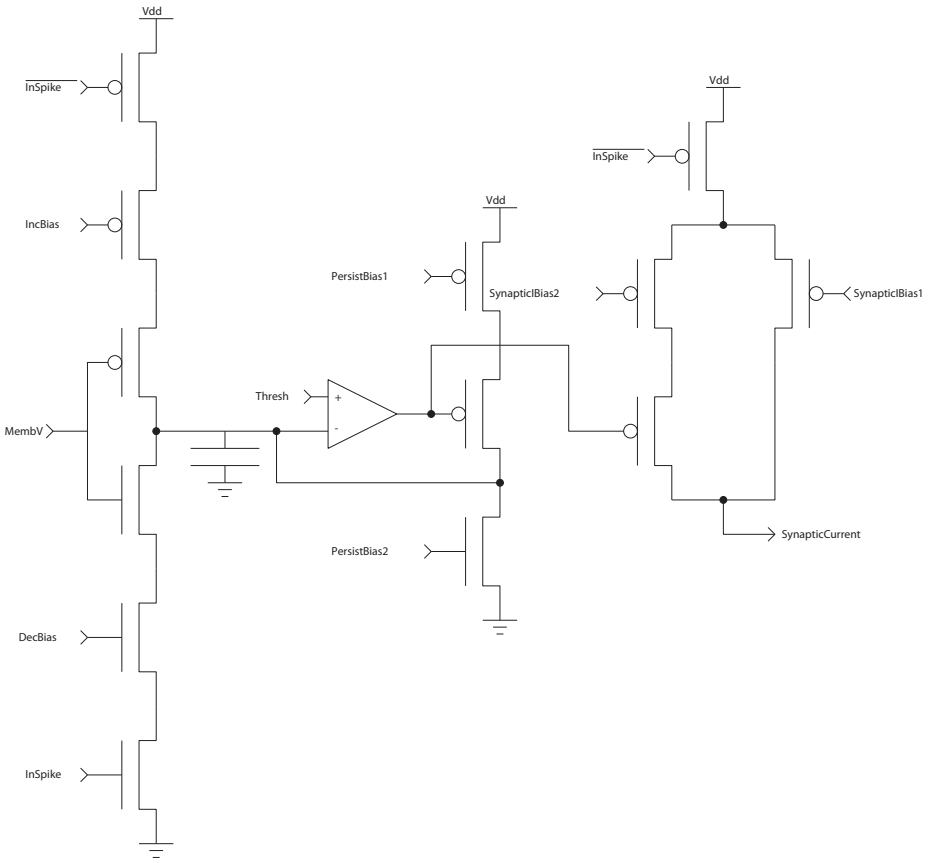


Figure 10.17: A bistable learning circuit [74]

tunneling node is big enough only in that synapse now) and the synaptic connection gets strengthened. To get a sort of Hebbian behaviour the Y inputs could be the same as or be closely correlated to the neuron's output and the active low learning inputs on the X inputs should be correlated with the input activity.

The normalization circuit on the right basically compares the total current of the synapses with a bias current. If it exceeds that bias current, the drain voltage to the synapse transistors is increased such that hot electron injection starts to lower all floating gates until the total current is equal to the bias current again. (Inverted capacitive feedback to the gates turns down the synapse at the same time to stabilize the current output during that readaption phase). This normalization is also dependent on the input because if any input causes a higher total output, it will also be activated. It limits the maximum of the total output current.

Another synaptic model has been introduced by Fusi etal.[74]. It is depicted in figure 10.17. It uses a kind of digital storage and the synapse has two states only. The transition between the two states depends on correlations in input activity and the membrane voltage.

The synapse is attached to an integrate-and-fire neuron. The signal 'MembV' is a digital signal resulting from a comparison of the membrane

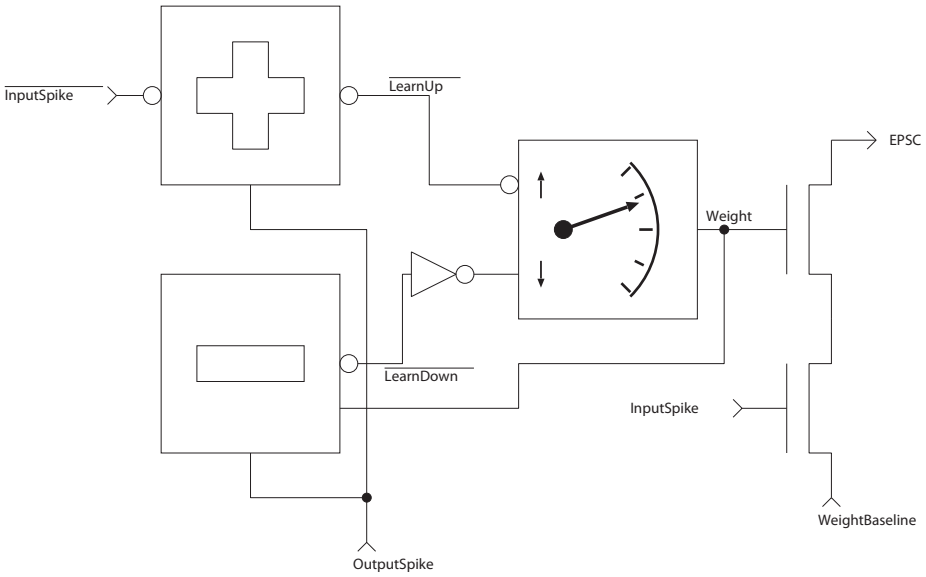


Figure 10.18: Blockdiagram of an example of a spike based learning circuit [64]

voltage with a threshold. If the membrane is above this threshold 'MembV' is low and vice versa. Thus whenever an 'InSpike' meets with a high membrane voltage, the voltage on the capacitor is increased and vice versa. As long as this voltage is above 'Thresh' it is slowly drifting towards Vdd. And the synapse is in the strong state: both branches supplying synaptic current, are open. If however, repeated input activity that meets with a low membrane potential drives the voltage on the capacitor below 'Thresh', then the voltage is slowly drifting towards Gnd and the synapse is in its weak state: the left branch supplying synaptic current is switched off. The overall behaviour is qualitatively Hebbian: correlation of inputs with high membrane voltages increase synaptic efficacy, whereas input activity that is not correlated with an excited neural state, decreases that efficacy.

10.3.2 A spike based learning circuit

The following circuits (block diagram in figure 10.18) presented in this subsection are based on the example learning rule mentioned in 10.1.4.4. There is a separate floating gate memory cells that stores an analog voltage and is controlled by digital input signals that move the content up or down (figure 10.21).

A 'learn up' block (figure 10.19) computes the length of a pulse according to the positive term in the weight update rule for every output spike (according to formula (10.11)). The voltage on the capacitance is incremented with every input spike and decays with a constant leakage current. With a output spike, the current comparator to the right and an additional leakage current from the capacitor are turned on. The current comparator issues a low voltage until the capacitor voltage falls below a

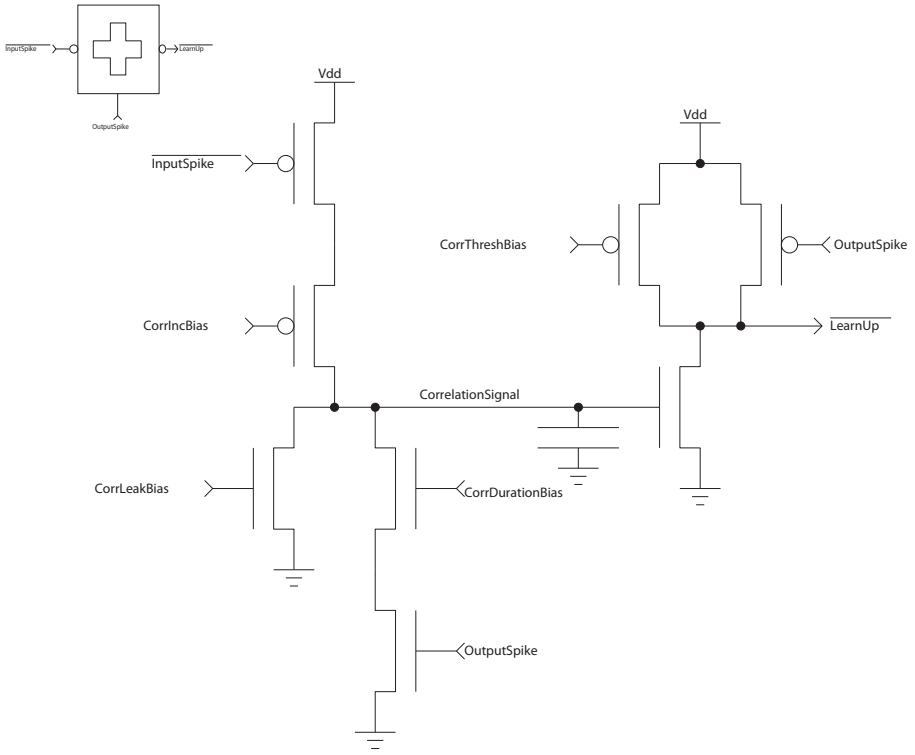


Figure 10.19: Circuit computing the positive term of the weight update

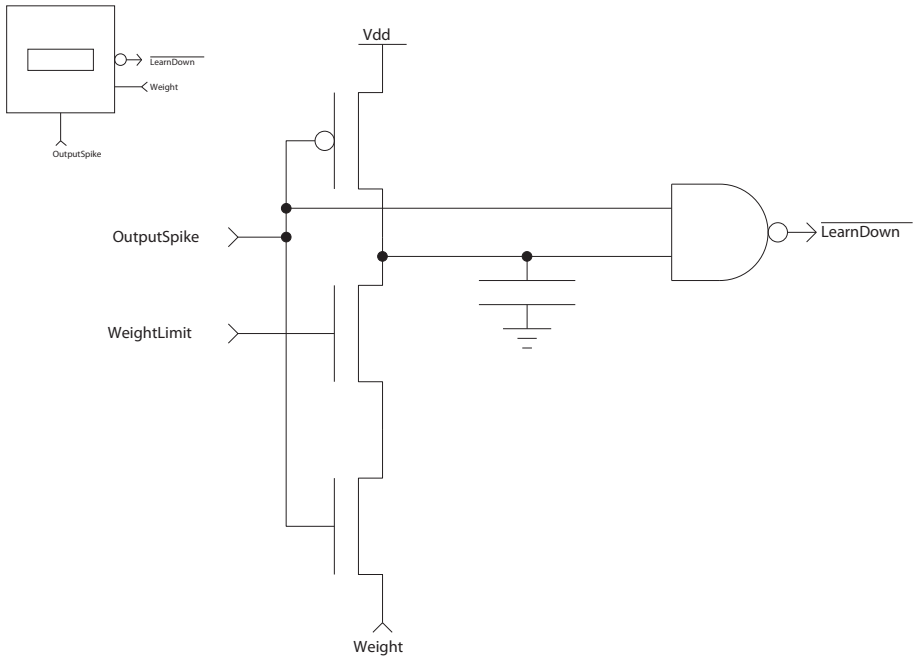


Figure 10.20: Circuit computing the negative term of the weight update rule.

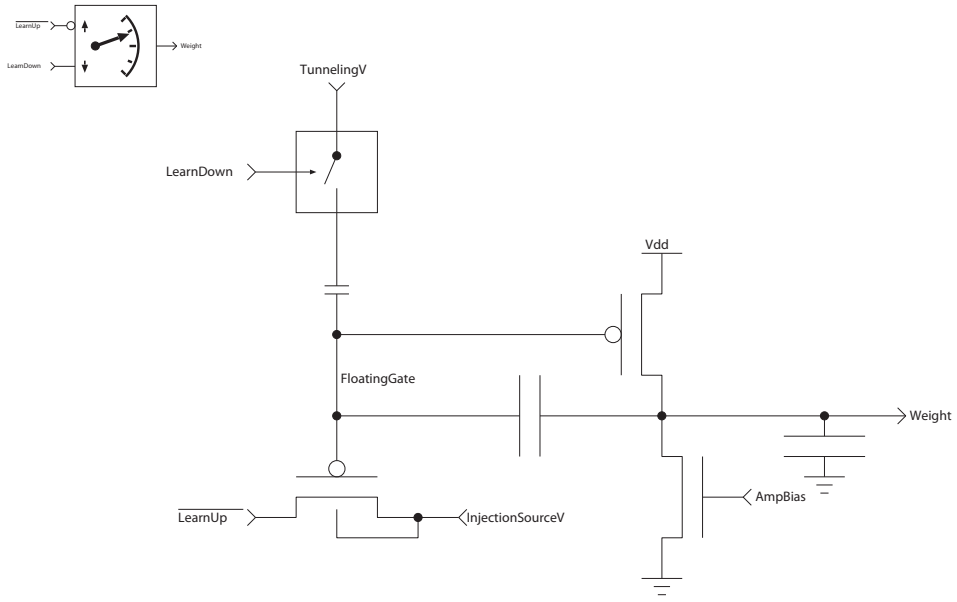


Figure 10.21: Floating gate analog storage cell

given level (chosen as low as possible), thus the resulting active low pulse is roughly proportional to the voltage on the capacitor.

The 'learn down' circuit (figure 10.20) computes the pulse length for the negative term in the learning rule. The pulse length, and thus the magnitude of this negative term, like in formula (10.11) is dependent on the momentary weight. The basic circuit is a NAND that receives the output spike as a direct input and the inverse of the output spike as a delayed input. The delay of the falling flank of the inverted spike input determines the pulse duration. That delay is proportional to the inverse of the current that is used to discharge the capacitor. That current is exponentially dependent on the weight voltage that sits at the source of the current limiting transistor.

Appendix A

Questions Catalogue

This is a catalogue of questions as they might appear at the oral exam. Each question is a starting point for a discussion of a particular topic in which knowledge and understanding will be tested.

A.1 Introduction

1) Can you name some differences between a computer and a brain as they were discussed in the lecture?

A.2 Neurophysiology

1) What do you know about tools/methods employed in neurophysiology?
2) What do you know about cortical regions?
3) Can you explain 'topological mapping' in brain areas?
4) What do you know of ocular dominance patterns in V1?
5) What do you know of orientation selectivity patterns in V1?
6) What do you know about the concept of cortical microcolumns?
7) What do you know of cortical layers?

A.3 Basic Analogue CMOS

1) can you discuss basic electronic building blocks available on a CMOS chip?
2) can you explain the characteristics of a field effect transistor (FET)?
3) Can you describe the Early effect?
4) Can you explain a current mirror?
5) Can you explain a differential pair?

6) Can you explain a transconductance amplifier?
7) Can you explain a follower?
8) Can you describe a resistive net?
9) Can you describe a diffuser network implemented with transistors?
10) Can you explain the winner take all circuit presented in the course?
11) Can you explain some extensions of the WTA circuit?

A.4 Real and Silicon Neurons

1) What do you know about the anatomy/physiology of a neuron?
2) Can you explain a Perceptron or Mc Culloch Pitts neuron?
3) Can you describe a Gilbert multiplyer?
4) Can you explain the integrate-and-fire circuit presented in the course?
5) Can you describe the adaptive integrate-and-fire circuit presented in the course?
6) Can you explain the firing mechanism of a neuron (compartemental neuron model) according to Hodgkin and Huxley?
7) Can you say something about how to implement a compartemental neuron model into CMOS?
8) Can you describe a compartemental model of a passive cable?

A.5 Coding in the Nervous System

1) Can you describe some physiological experiments that reveal clues on neuronal coding mechanisms? (At least one on rate and one on temporal encoding!)
2) What do you know of neural coding principles?
3) What do you know about the distinction of temporal and rate coding?
4) What distinguishes population and synchrony codes?
5) Can you explain rank order and latency encoding?

A.6 Neuromorphic Communication: the AER Protocol

1) What is the basic principle of AER
2) What do you know about different collision handling concepts employed in AER?
3) Can you explain the arbitration circuit presented in the course?

4) Can you explain the collision detecting/discarding AER receiver presented in the course?
5) Can you describe the principle of the 'aging versus loss' arbitration?

A.7 Retinomorphic Circuits

1) What do you know about the anatomy/physiology of the eye?
2) What photo active CMOS elements do you know?
3) How can you achieve logarithmic amplification of a photo current?
4) What is a common source amplifier?
5) Can you explain a source follower?
6) Explain the 'active pixel'!
7) Can you describe read out methods for photo arrays?
8) Can you explain one of the two silicon retina circuits presented in the course?
9) Can you explain the non-linear element according to Delbrück?
10) Can you explain the adaptive photo cell?
11) Can you explain token based motion detection?
12) Can you explain intensity based motion detection?
13) Can you explain convolution and feature maps?

A.8 Cochleomorphic Circuits

1) What do you know about the anatomy and physiology of the ear?
2) Can you explain the second order filter used for the silicon cochlea?
3) Can you describe a silicon cochlea?

A.9 Neuromorphic Learning

1) Can you define 'learning'?
2) What do you know about the main categories of learning algorithms?
3) Can you explain Hebbian learning?
4) Can you explain gradient decent learning?
5) Can you tell something about competitive learning?
6) Can you tell something about spike based learning?
7) What do you know about methods for analog or quasi-analog storage on a CMOS device?

8) Can you explain the DA/AD storage cell presented in the course?

9) Can you explain the high-voltage switch that was presented in the course?

10) What do you know about Fowler-Nordheim tunneling and hot electron injection?

11) Can you explain the bump circuit presented in the course?

12) Can you explain the fusing amplifier?

13) Can you explain 'weak' multi-level memory?

14) Can you describe one of the learning circuits presented in the course (Diorio/Fusi/Häfliger)?

Bibliography

[1] S. Thorpe, F. Fize, and C. Marlot, "Speed of processing in the human visual system," *Nature*, vol. 381, pp. 520–522, June 1996.

[2] M. Fabre-Thorpe, A. Delorme, C. Marlot, and S. Thorpe, "A limit to the speed of processing in ultra-rapid visual categorization of novel natural scenes," *Journal of Cognitive Neuroscience*, vol. 13, pp. 171–180, 2001.

[3] S. Sands and M. Pflieger, 1999, http://www.neuro.com.

[4] S. H. Cardoso, 1997, http://www.epub.org.br/cm/n02/mente/neurobiologia_i.htm.

[5] A. E. Hernandez, A. Martinez, E. C. Wong, L. R. Frank, and R. B. Buxton, "Bilingual research projects, fMRI," 1999, http://crl.ucsd.edu/bilingual/fMRI3.html.

[6] A. Branner, R. Stein, and R. Normann, "Selective stimulation and recording using a slanted multielectrode array," in *Proceedings of the First Joint BMES/EMBS Conference*, vol. 1, 1999, p. 377.

[7] C. Stricker and A. Cowan, "Private communication," 2002, .

[8] W. Denk, 2002, http://wbmo.mpimf-heidelberg.mpg.de/Biomedizinische.Optik.html.

[9] F. H. Netter, *Atlas of Human Anatomy*. Arsley, USA: Ciba-Geigy Corp., 1989.

[10] C. G. Phillips and R. Porter, *Cortico-Spinal Neurones: Their Role in Movement*. Academic Press, London, 1977.

[11] D. C. van Essen, C. H. Anderson, and D. J. Felleman, "Information processing in the primate visual system: An integrated systems perspective," *Science*, vol. 255, pp. 419–422, 1992.

[12] J. Horton and D. Hocking, "Intrinsic variability of ocular dominance column periodicity in normal macaque monkeys," *Journal of Neuroscience*, vol. 16, pp. 7228–7239, 1996.

[13] G. G. Blasdel, "Differential imaging of ocular dominance columns and orientation selectivity in monkey striate cortex," *Journal of Neuroscience*, vol. 12, pp. 3115–3138, 1992.

[14] ——, "Orientation selectivity, preference, and continuity in monkey striate cortex," *Journal of Neuroscience*, vol. 12, pp. 3139–3161, 1992.

[15] G. G. Blasdel and G. Salama, "Voltage-sensitive dyes reveal a modular organization in monkey striate cortex," *Nature*, vol. 321, pp. 579–585, 1986.

[16] M. E. McCourt, "Central visual pathways," 1997, http://www.psychology.psych.ndsu.nodak.edu/mccourt/website/htdocs/

HomePage/Psy460/Central%20visual%20pathways/Central%20Visual%20Pathways.html.

[17] J. Nolte, *The Human Brain*. Mosby, 1988.

[18] G. M. Shepherd, *Neurobiology*, 3rd ed. Oxford University Press, 1994.

[19] A. L. Hodgkin and A. F. Huxley, "Current carried by sodium and potassium ions through the membrane of the giant axon of loglio," *Journal of Physiology*, vol. 116, p. 449, 1952.

[20] T. V. P. Bliss and T. Lømø, "Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path," *Journal of Physiology*, vol. 232, no. 2, pp. 331–356, July 1973.

[21] M. V. Tsodyks and H. Markram, "Redistribution of synaptic efficacy between neocortical pyramidal neurons," *Nature*, vol. 382, pp. 807–810, August 1996.

[22] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, "Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs," *Science*, vol. 275, pp. 213–215, 1997.

[23] E. Vittoz, "Analog VLSI signal processing: Why, where and how?" *Analog Integrated Circuits and Signal Processing*, pp. 27–44, July 1994.

[24] K. Schuegraf and C. Hu, "Hole injection SiO_2 breakdown model for very low voltage lifetime extrapolation," *IEEE Tran. on Electron Devices*, vol. 41, no. 5, pp. 761–767, 1994.

[25] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proc. of the IEEE*, vol. 91, no. 2, pp. 305 – 327, 2003.

[26] P. Häfliger and H. K. O. Berge, "Exploiting gate leakage in deep-submicrometer CMOS for input offset adaptation," *IEEE Transactions on Circuits and Systems II*, vol. 54, no. 2, pp. 127–130, 2007.

[27] C. Mead, *Analog VLSI and Neural Systems*. Addison Wesley, 1989.

[28] M. Mahowald and R. Douglas, "A silicon neuron," *Nature*, vol. 354, pp. 515–518, 1991.

[29] C. Rasche, R. Douglas, and M. Mahowald, "Characterization of a silicon pyramidal neuron," in *Neuromorphic Systems: Engineering Silicon from Neurobiology*, L. S. Smith and A. Hamilton, Eds. World Scientific, 1998, ch. 14, pp. 169–177.

[30] A. L. Hodgkin and A. F. Huxley, "The components of membrane conductance in the giant axon of loglio," *Journal of Physiology*, vol. 116, p. 473, 1952.

[31] D. Hubel and T. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *Journal of Physiology*, vol. 160, pp. 106–154, 1962.

[32] C. M. Gray, P. König, A. K. Engel, and W. Singer, "Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which

reflects global stimulus properties," *Nature*, vol. 338, pp. 334–337, 1989.

[33] W. Bair and C. Koch, "Temporal precision of spike trains in extrastriate cortex of the behaving macaque monkey," *Neural Computation*, vol. 8, pp. 1185–1202, 1996.

[34] M. Abeles, *Corticonics, Neural Circuits of the Cerebral Cortex*. Cambridge University Press, 1991.

[35] J. O'Keefe and M. Recce, "Phase relationship between hippocampal place units and the eeg theta rhythm," *Hippocampus*, vol. 3, no. 3, pp. 317–330, July 1993.

[36] M. Mahowald, *An Analog VLSI System for Stereoscopic Vision*. Kluwer, 1994.

[37] A. Mortara and E. A. Vittoz, "A communication architecture tailored for analog VLSI artificial neural networks: intrinsic performance and limitations," *IEEE Trans. on Neural Networks*, vol. 5, pp. 459–466, 1994.

[38] P. O. Pouliquen and A. G. Andreou, "Bit-serial address-event representation," *Proceedings of Conference on Information Sciences and Systems*, March 1999.

[39] K. Boahen, "A throughput-on-demand address-event transmitter for neuromorphic chips," in *Adv. Res. in VLSI*. IEEE Comp. Soc. Press, 1999, pp. 72–86.

[40] Z. Kalayjian, J. Waskiewicz, D. Yochelson, and A. G. Andreou, "Asynchronous sampling of 2d arrays using winner-takes-all arbitration," in *ISCAS*, vol. 3, 1996, pp. 393–396.

[41] J. T. Marienborg and T. S. Lande, "Analog state transmission with digital hardware," in *NORCHIP*, 1998, pp. 249–256.

[42] FOVEON, "http://www.foveon.com."

[43] H. Ji and P. A. Abshire, Eds., *Fundamentals of Silicon-Based Phototransduction CMOS Imagers from Phototransduction to Image Processing*. Kluwer Academic Publishers.

[44] R. S. Muller and T. I. Kamins, *Device Electronics for Integrated Circuits*. John Wiley and Sons, 1986.

[45] A. Moini, *Vision Chips*. Kluwer Academic Publishers, 2000.

[46] T. Delbruck and D. Oberhoff, "Self-biasing low power adaptive photoreceptor," in *IEEE Int. Symp. Circuits Syst., vol. 4*, 2004, pp. 844–847.

[47] P. Lichtsteiner, C. Posch, and T. Delbruck, "An 128x128 120dB 15us-latency temporal contrast vision sensor," *IEEE J. Solid State Circuits*, vol. 43, no. 2, pp. 566–576, 2007.

[48] T. Delbrück and C. Mead, "An electronic photoreceptor sensitive to small changes in intensity," in *Advances in Neural Inf. Proc. Sys.*, vol. 1, 1989, pp. 720–726.

[49] K. A. Boahen, "The retinomorphic approach: Pixel-parallel adaptive amplification, filtering, and quantization," in *Neuromorphic Systems Engineering*, T. S. Lande, Ed. Kluwer, 1998, ch. 11.

[50] K. Fukushima, Y. Yamaguchi, M. Yasuda, and S. Nagata, "An electronic model of the retina," *Proceedings of the IEEE*, vol. 58, no. 12, December 1970.

[51] J. Kramer, R. Sarpeshkar, and C. Koch, "Pulse-based analog VLSI velocity sensors," *IEEE Trans. on Circ. and Sys.-II*, vol. 44, no. 2, pp. 86–100, February 1997.

[52] A. Stocker and R. Douglas, "Computation of smooth optical flow in a feedback connected analog network," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 11, 1999, pp. 706–712.

[53] S. Grossberg, E. Mingolla, and J. Williamson, "Synthetic aperture radar processing by a multiple scale neural system for boundary and surface representation," *Neural Networks*, vol. 8, no. 7/8, pp. 1005–1028, 1995.

[54] T. Serrano-Gotarredona, A. G. Andreou, and B. Linares-Barranco, "AER image filtering architecture for vision-processing systems," *IEEE Transactions on Circuits and Systems*, vol. 46, no. 9, pp. 1064–1071, 1999.

[55] J. Lazzaro and C. A. Mead, "Circuit models of sensory transduction in the cochlea," in *Analog VLSI Implementations of Neural Networks*, C. A. Mead and M. Ismail, Eds. Kluwer, 1989, pp. 85–101.

[56] R. Sarpeshkar, R. F. Lyon, and C. Mead, "A low-power wide-dynamic-range analog VLSI cochlea," in *Neuromorphic Systems Engineering*, T. S. Lande, Ed. Boston: Kluwer Academic Publishers, 1998, ch. 3, pp. 49–103.

[57] D. Rumelhart, G. Hinton, and R. Williams, *Parallel Distributed Processing*. Cambridge, MA: MIT Press, 1986, ch. 8: Learning internal representations by error propagation, pp. 319–362.

[58] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*. Addison Wesley, 1991.

[59] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, pp. 9–44, 1988.

[60] G. Tesauro, "Practical issues in temporal difference learning," *Machine learning*, vol. 8, pp. 257–277, 1992.

[61] D. O. Hebb, *The Organization of Behavior*. Wiley, New York, 1949.

[62] T. Kohonen, *Self-Organization and Associative Memory*. Berlin: Springer, 1984, p. 99.

[63] T. Masquelier, R. Guyonneau, and S. Thorpe, "Spike timing dependent plasticity finds the start of repeating patterns in continous spike trains," *PLoS ONE*, vol. 3, no. 2, p. e1377, 2008.

[64] P. Häfliger, "A spike based learning rule and its implementation in analog hardware," Ph.D. dissertation, ETH Zürich, Switzerland, 2000, http://www.ifi.uio.no/~hafliger.

[65] P. Heim and M. A. Jabri, "Long-term CMOS static storage cell performing AD/DA conversion for analogue neural network implementations," *Electronic Letters*, vol. 30, no. 25, December 1994.

[66] P. Häfliger and H. K. Riis, "A multi-level static memory cell," in *Proc. of IEEE ISCAS*, vol. 1, Bangkok, Thailand, May 2003, pp. 22–25.

[67] H. K. Riis and P. Häfliger, "Spike based learning with weak multi-level static memory," in *Proc. of IEEE ISCAS*, vol. V, Vancouver, Canada, May 2004, pp. 393–395.

[68] D. Kahng and S. M. Sze, "A floating gate and its application to memory devices," *The Bell System technical Journal*, pp. 1288–1295, July/August 1967.

[69] M. Holler, S. Tam, H. Castro, and R. Benson, "An electrically trainable artificial neural network (ETANN) with 10240 'floating gate' synapses," *Int. Joint Conf. on Neural Networks*, no. II, pp. 191–196, June 1989.

[70] C. Diorio, S. Mahajan, P. Hasler, B. Minch, and C. Mead, "A high-resolution non-volatile analog memory cell," *Proc. IEEE Intl. Symp. on Circuits and Systems*, vol. 3, pp. 2233–2236, 1995.

[71] R. Fowler and L. Nordheim, "Electron emission in intense electric fields," in *Proc. Roy. Soc. Lond.*, ser. A, vol. 119, no. 781, 1928, pp. 173–181.

[72] C. Diorio, P. Hasler, B. A. Minch, and C. Mead, "A single-transistor silicon synapse," *IEEE Trans. Electron Devices*, vol. 43, no. 11, pp. 1972–1980, 1996.

[73] ——, "A floating-gate MOS learning array with locally computed weight updates," *IEEE Transactions on Electron Devices*, vol. 44, no. 12, pp. 2281–2289, December 1997.

[74] S. Fusi, M. Annunziato, D. Badoni, A. Salamon, and D.J.Amit, "Spike-driven synaptic plasticity: theory, simulation, VLSI implementation," *Neural Computation*, vol. 12, pp. 2227–2258, 2000.