An analogue VLSI study of temporally asymmetric Hebbian learning

Adrià Bofill Petit

A thesis submitted for the degree of Doctor of Philosophy.
The University of Edinburgh.
May 2005
Abstract

The primary aim of this thesis is to examine whether temporally asymmetric Hebbian learning in analogue VLSI can support temporal correlation learning and spike-synchrony processing. Novel circuits for synapses with spike-timing-dependent plasticity (STDP) are proposed. Results from several learning experiments conducted with a chip containing a small feed-forward network of neurons with STDP synapses are presented.

The learning circuits proposed in this thesis can be used to implement weight-independent STDP and learning rules with weight-dependent potentiation. Test results show that the learning windows implemented are very similar to those found in biological neurons. The peaks of potentiation and depression, as well as the decay of both sides of the STDP learning window, can be tuned independently. Therefore, the circuits proposed can be used to explore learning rules with different characteristics.

The main challenge for on-chip learning is the long-term storage of analogue weights. Previous investigations of temporally asymmetric Hebbian learning rules have shown that weight-independent STDP creates bimodal weight distributions. This thesis investigates the suggestion that the bimodality of the learning rule may render the long-term storage of analogue values unnecessary. Several experiments have been carried out to study the weight distributions created on-chip. With both weight-independent and moderate weight-dependent learning rules the on-chip synapses develop either maximum or zero weights. The results presented show that, in agreement with theoretical analysis of STDP, the mean of the input weight vector decreases with the mean rate of the input spike trains. Some experiments reported indicate that the instability of weight-independent STDP could be used in some applications to maintain the binary weights learnt when the temporal correlations are removed from the inputs.

Test results given show that both zero-delay correlations and narrow time windows of correlation can be detected with the hardware neurons. An on-chip two-layer network has been used to detect a hierarchical pattern of temporal correlations embedded in noisy spike trains. The analysis of the activity generated by the network shows that the bimodal weight distribution emerging from STDP learning amplifies the spike synchrony of the inputs.
Declaration of originality

I hereby declare that the following thesis is based on the results of investigations conducted by myself and that the thesis is of my own composition. Work other than my own is clearly indicated in the text by reference to the relevant publications. This thesis has not, in whole or in part, been submitted for any other degree.

Adrià Bofill i Petit
I thank Prof. Alan Murray, my supervisor, for giving me the opportunity to work in his research group. I also would like to thank Dr. Martin Reekie for the circuit discussions we had in the early days of the project. I must also acknowledge the financial support I received from the University of Edinburgh.

I want to acknowledge very specially Patrice Fleury for his assistance in repeated occasions during my Ph.D. I was helped by many members of the old Integrated Systems Group and the new Institute for Integrated Micro and Nano Systems. In particular, I would like to thank Dr. Robin Woodburn, Damon Thompson, Katherine Cameron and Dr. Alasdair Sutherland.

Finally, I thank Eva for giving me love and support during all these years.
Contents

Declaration of originality ........................................ iii
Acknowledgements ...................................................... iv
Contents .................................................................. v
List of figures .......................................................... viii
Acronyms and abbreviations ......................................... x

1 Introduction................................................................ 1
  1.1 Neural Systems ..................................................... 1
  1.2 Neuromorphic hardware .......................................... 2
  1.3 Spiking neurons ..................................................... 3
  1.4 Spike-based learning .............................................. 4
  1.5 Thesis ................................................................ 4
  1.6 Thesis outline ....................................................... 5

2 Spiking neurons and temporally asymmetric Hebbian learning ............................................. 7
  2.1 Introduction ........................................................ 7
  2.2 The neural codes .................................................... 8
    2.2.1 Rate codes ..................................................... 9
    2.2.2 Spike codes ................................................... 10
  2.3 Spiking neurons ..................................................... 13
    2.3.1 Bio-physical models ......................................... 13
    2.3.2 The leaky integrate-and-fire neuron model ............. 14
  2.4 Hebbian learning ................................................... 16
  2.5 Spike-timing-dependent plasticity ............................. 17
    2.5.1 The learning window ....................................... 18
    2.5.2 Weight-dependent vs. weight-independent learning .... 20
  2.6 Learning with STDP ............................................... 22
    2.6.1 Inputs without temporal correlations ..................... 22
    2.6.2 Synchrony detection ....................................... 23
    2.6.3 Synchrony and delay lines ................................ 23
    2.6.4 Conditioning, prediction and sequence learning ....... 24
    2.6.5 Fast reaction neurons ....................................... 24
  2.7 Summary ............................................................ 25

3 Neural hardware: the analogue way ................................................. 26
  3.1 Introduction ........................................................ 26
  3.2 Neuromorphic aVLSI .............................................. 26
  3.3 Silicon wants to go fast ......................................... 27
  3.4 On-chip Learning ................................................... 28
    3.4.1 Weight Storage .............................................. 29
  3.5 Spiking hardware .................................................. 31
    3.5.1 Spike transmission .......................................... 31
### Contents

3.6 Synapses .............................................. 32  
   3.6.1 Simple synapses ................................ 32  
   3.6.2 Linear synapses .................................. 34  
   3.6.3 Dynamic synapses ................................. 35  
3.7 Silicon integrate-and-fire neurons ...................... 36  
   3.7.1 Silicon IF with capacitive feedback ............... 37  
   3.7.2 Silicon neurons with ionic currents, refractory period and spike-rate adaptation .......... 38  
   3.7.3 Switched-capacitor integrate-and-fire neuron ........ 40  
3.8 Spike-based learning in hardware ....................... 41  
3.9 Summary ............................................... 44

4 Mapping temporally asymmetric Hebbian learning into aVLSI 45  
   4.1 Introduction ....................................... 45  
   4.2 The opportunities .................................. 45  
      4.2.1 Locality ..................................... 46  
      4.2.2 Simple learning window ...................... 46  
      4.2.3 Signal averaging and summation .............. 47  
      4.2.4 Bimodal weight distributions ................. 47  
   4.3 The challenges ..................................... 48  
      4.3.1 Analogue weights and leakage ................. 48  
      4.3.2 Switching effects ............................ 49  
      4.3.3 Reduced visibility of the internal operation 49  
      4.3.4 Low number of synapses, unstable learning process and circuit mismatch 50  
   4.4 Design guidelines .................................. 50  
      4.4.1 Capacitive weight storage .................... 50  
      4.4.2 Weak inversion vs. strong inversion .......... 51  
      4.4.3 Time-scale .................................. 51  
      4.4.4 Parameter tuning, mismatch, biasing and criteria for device selection ................. 52  
      4.4.5 Neuron features ................................ 55  
      4.4.6 Weight-independent vs. weight-dependent learning ............................................. 55  
   4.5 Hardware spiking neuron with STDP synapses .......... 55  
      4.5.1 Introduction ................................... 55  
      4.5.2 Neuron architecture ......................... 56  
      4.5.3 Integrate-and-fire circuit ................... 58  
      4.5.4 Pulse sequencer ................................ 62  
      4.5.5 Learning synapse .............................. 64  
      4.5.6 Non-learning synapses ....................... 68  
      4.5.7 A-causal correlation circuit .................. 70  
      4.5.8 Causal correlation circuit .................... 71  
      4.5.9 Weight change circuit ....................... 73  
      4.5.10 Current biasing ............................. 74  
      4.5.11 Final remarks ................................ 74  
   4.6 Summary .............................................. 76

5 Neuron and learning rule characterisation .................. 78  
   5.1 Introduction ....................................... 78
# List of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Spike trains in biological neurons</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Binding features by means of spike-firing synchrony</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>Leaky integrate-and-fire model</td>
<td>15</td>
</tr>
<tr>
<td>2.4</td>
<td>STDP learning window</td>
<td>19</td>
</tr>
<tr>
<td>2.5</td>
<td>Weight-dependent vs weight-independent learning rules</td>
<td>21</td>
</tr>
<tr>
<td>3.1</td>
<td>Charge leakage in weight capacitor</td>
<td>29</td>
</tr>
<tr>
<td>3.2</td>
<td>Simple analogue VLSI synapses</td>
<td>33</td>
</tr>
<tr>
<td>3.3</td>
<td>A linear synapse for pulse-stream learning neural networks</td>
<td>35</td>
</tr>
<tr>
<td>3.4</td>
<td>A dynamic hardware synapse</td>
<td>36</td>
</tr>
<tr>
<td>3.5</td>
<td>Self-resetting silicon integrate-and-fire neuron with capacitive voltage feedback</td>
<td>38</td>
</tr>
<tr>
<td>3.6</td>
<td>Spiking neuron with Na-K currents and refractory period</td>
<td>39</td>
</tr>
<tr>
<td>3.7</td>
<td>Large resistances can be implemented with switched capacitors</td>
<td>41</td>
</tr>
<tr>
<td>3.8</td>
<td>Internal analogue variable has two long-term stable states</td>
<td>43</td>
</tr>
<tr>
<td>4.1</td>
<td>Diagram of the neuron architecture</td>
<td>57</td>
</tr>
<tr>
<td>4.2</td>
<td>Network connectivity</td>
<td>58</td>
</tr>
<tr>
<td>4.3</td>
<td>Integrate-and-fire neuron circuit</td>
<td>59</td>
</tr>
<tr>
<td>4.4</td>
<td>Components of the integrate-and-fire neuron</td>
<td>61</td>
</tr>
<tr>
<td>4.5</td>
<td>Pulse-sequencer circuit</td>
<td>63</td>
</tr>
<tr>
<td>4.6</td>
<td>Pulse-generator circuit</td>
<td>64</td>
</tr>
<tr>
<td>4.7</td>
<td>Transconductor using transistors in ohmic region</td>
<td>65</td>
</tr>
<tr>
<td>4.8</td>
<td>Schematic of the synapse circuit</td>
<td>67</td>
</tr>
<tr>
<td>4.9</td>
<td>Non-learning synapses</td>
<td>69</td>
</tr>
<tr>
<td>4.10</td>
<td>A-causal correlation circuit</td>
<td>70</td>
</tr>
<tr>
<td>4.11</td>
<td>Causal correlation circuit</td>
<td>72</td>
</tr>
<tr>
<td>4.12</td>
<td>Weight change circuit</td>
<td>73</td>
</tr>
<tr>
<td>4.13</td>
<td>Biases</td>
<td>75</td>
</tr>
<tr>
<td>5.1</td>
<td>Network architecture and chip stimulation</td>
<td>79</td>
</tr>
<tr>
<td>5.2</td>
<td>Micro-chip photograph</td>
<td>80</td>
</tr>
<tr>
<td>5.3</td>
<td>Neuron stimulation</td>
<td>81</td>
</tr>
<tr>
<td>5.4</td>
<td>Test setup</td>
<td>82</td>
</tr>
<tr>
<td>5.5</td>
<td>External biases</td>
<td>83</td>
</tr>
<tr>
<td>5.6</td>
<td>Spiking behaviour and weight update</td>
<td>84</td>
</tr>
<tr>
<td>5.7</td>
<td>Weight modification and synaptic strength</td>
<td>85</td>
</tr>
<tr>
<td>5.8</td>
<td>Synapse. Test results</td>
<td>86</td>
</tr>
<tr>
<td>5.9</td>
<td>Measurement of the learning window. Control of the peaks.</td>
<td>88</td>
</tr>
<tr>
<td>5.10</td>
<td>Measurement of the learning window. Control of the learning window time constants</td>
<td>89</td>
</tr>
<tr>
<td>5.11</td>
<td>Measurement of the learning window. Weight-dependent learning.</td>
<td>91</td>
</tr>
<tr>
<td>Figure Number</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6.1</td>
<td>Methods to build correlated spike trains</td>
<td>94</td>
</tr>
<tr>
<td>6.2</td>
<td>Weight evolution caused by uncorrelated inputs</td>
<td>97</td>
</tr>
<tr>
<td>6.3</td>
<td>Effect of input firing rate on the weight distribution</td>
<td>99</td>
</tr>
<tr>
<td>6.4</td>
<td>Learning from inputs with 0.4 correlation coefficient</td>
<td>101</td>
</tr>
<tr>
<td>6.5</td>
<td>Spike rasters of a temporal correlation experiment</td>
<td>103</td>
</tr>
<tr>
<td>6.6</td>
<td>Weight retention</td>
<td>104</td>
</tr>
<tr>
<td>6.7</td>
<td>No weight retention</td>
<td>105</td>
</tr>
<tr>
<td>6.8</td>
<td>Stimuli and final weight distribution in the network experiment</td>
<td>108</td>
</tr>
<tr>
<td>6.9</td>
<td>Weight evolution for the network learning experiment</td>
<td>110</td>
</tr>
<tr>
<td>6.10</td>
<td>On-chip synchrony amplification</td>
<td>111</td>
</tr>
</tbody>
</table>
# Acronyms and abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>Artificial Neural Network</td>
</tr>
<tr>
<td>aVLSI</td>
<td>Analogue Very-Large-Scale Integration</td>
</tr>
<tr>
<td>EPSP</td>
<td>Excitatory Post-Synaptic Potential</td>
</tr>
<tr>
<td>IF</td>
<td>Integrate and Fire</td>
</tr>
<tr>
<td>MOS</td>
<td>Metal Oxide Semiconductor</td>
</tr>
<tr>
<td>NMOS</td>
<td>N-type MOS</td>
</tr>
<tr>
<td>OpAmp</td>
<td>Operational Amplifier</td>
</tr>
<tr>
<td>PMOS</td>
<td>P-type MOS</td>
</tr>
<tr>
<td>STDP</td>
<td>Spike-Timing-Dependent Plasticity</td>
</tr>
<tr>
<td>TAH</td>
<td>Temporally Asymmetric Hebbian</td>
</tr>
<tr>
<td>VLSI</td>
<td>Very-Large-Scale Integration</td>
</tr>
</tbody>
</table>
Chapter 1
Introduction

1.1 Neural Systems

Simple animals can perform tasks that are difficult to carry out even by machines controlled by the most powerful digital computers. While computers are good at following precise instructions to perform mathematical operations on well-defined data very fast, animals are fault tolerant in regards to input data errors. Furthermore, biological neural systems do not have a strict regular architecture made up of fast and precise elements that interact among themselves and the environment in a sequential manner as governed by a central clock. Instead, nervous systems are composed of slow and imprecise units that make up a massively parallel structure that operates asynchronously.

Since the 1950s, machines and computing algorithms inspired by biological neural systems have been developed. However, it was in the 1980s with the advent of the backpropagation algorithm [1] for network training and the so-called Hopfield Network [2] that the field of Artificial Neural Networks (ANN) saw a surge of research activity both on theoretical investigations and applications to solve real-world problems. Since, the field of Artificial Neural Network has developed further and blended with other scientific disciplines such as statistics to give rise to a field of scientific research usually known as Machine Learning. Hence, although the field of ANN initially stemmed from an interest in biological neural organisation, currently draws most of its inspiration from mathematical formulations found in statistical theory and statistical mechanics.

Most researchers in ANN focus their investigations mainly on what the algorithms can do, hence abstracting the algorithms underlying the computation from their physical substrate. Other researchers, however, are also interested on how neural algorithms are implemented in biology and place as much importance to the physical systems carrying out the computation. The reasons that lead engineers and scientists working in disciplines related to ANN to study the material aspects of neural computation in biological systems are twofold.
Firstly, algorithms that have emerged in neurobiological systems have done so, not only constrained by the external environment where animals live, but also by the physical substrate where the processing of information takes place (the wetware). The algorithms and architectures that drive animal behaviour have been shaped not only by the data (the environment), but also by the physical characteristics (speed, energy consumption, mechanisms for information transmission and growth) attainable by organic materials which have evolved into very specialised cells called neurons and glia. Hence, a good understanding of the physical implementation of biological neural systems is required to decide whether some low-level features of neurons are mere technological solutions or rather key features of the algorithms underlying biological neural processing. For example, most ANN architectures take very simplistic descriptions of neurons. However, biological neurons are very feature-rich computing elements that can perform relatively complex operations at the single neuron level [3].

Secondly, many neural researchers think we should look into biology not only seeking algorithmic efficiency, but also to discover very power efficient and fault tolerant physical implementations of such algorithms which could help us develop new physical technologies for computation [4]. The style of computation carried out by biological systems is possibly very tightly coupled to their physical implementation. Thus, artificial systems performing similar tasks might benefit greatly by being implemented in a similar manner, although sometimes they will encounter different technological constraints.

1.2 Neuromorphic hardware

Among those interested in the physical constraints of neural algorithms are engineers developing neuromorphic systems. These systems do not only attempt to reproduce the high-level functions of neuronal systems, but they also aim at mimicking the low-level components of biological neurons [5]. Some neuromorphic systems work is based on idealised neurons which resemble their biological counterparts only because they generate spiking activity at the output as response to stimulations. Other neuromorphic investigations model very detailed phenomena such as calcium-dependent effects in the sensitivity of the neuron to the strength of the input current [6].

Analogue VLSI (Very Large Scale Integration) circuits are widely used for the implementation of neuromorphic systems. The non-linear characteristics of some primitive electronic devices,
and physical principles such as conservation of charge, are exploited to build compact building blocks that match those elements found in neurobiology (synapses, spiking behaviour, etc.). The rationale behind investigations on analogue VLSI neuromorphic systems is threefold:

- Some researchers goal is to build smart sensors inspired by the ability of animals to interact robustly with the environment with limited "hardware" resources.
- Analogue VLSI can also be used as a tool to investigate neuronal models embodied in a physical substrate (silicon) interacting in real time with the environment [7, 8].
- Finally, some engineers see neuromorphic systems as an opportunity to exploit some of the non-linearities of silicon devices.

In general, many neuromorphic researchers are driven by all of these goals, which are not mutually exclusive.

1.3 Spiking neurons

The spiking behaviour of neurons and the role of time are probably the most evident characteristics of biological neural processing left out by classical ANN algorithms. Thus, neurons in most ANN operate upon and communicate information represented by continuous-valued signals (e.g. MLP with backpropagation learning [1]) or binary signals (e.g. original Hopfield model [2]) at time steps set by an external clock. In contrast, other artificial neurons like the biologically inspired integrate-and-fire neuron (IF) model communicate information in an asynchronous manner according to an internal mechanism local to each neuron.

It is well known that information in biological neuronal systems is often encoded by mean firing rates of neurons [9]. However, an increasing number of evidence exists to support the idea that irregular firing patterns of neurons are more than an evolved technological solution to information transmission in "wetware". Complementary neural coding schemes like inter-neuron spike-firing synchrony, oscillations, phase coding and complex temporal firing patterns are possible if we consider that precise spike timings may signal additional information. Clearly, some of these alternative neural representations consider the temporal variability of spike firing as a feature rather than a bug [10]. Neuromorphic engineers, together with other ANN researchers interested in the implementation of neural algorithms in biology, have understandably been
attracted by the possibility of using temporal information in artificial neural systems.

1.4 Spike-based learning

In the last decade, several neurophysiological studies have shown that neural plasticity — the modification of the strength of connections between neurons — is driven by precise spike-timing differences between presynaptic and postsynaptic spikes. These new forms of plasticity are usually termed as Spike-Timing Dependent Plasticity (STDP). If the learning rule is such that the strength of the connection is potentiated when the presynaptic neuron fires immediately before the postsynaptic neuron, while depression occurs when the postsynaptic neuron fires first, then STDP implements a form of learning known as temporally asymmetric Hebbian learning (TAH). Traditionally, Hebb’s postulates [11] had been interpreted in terms of correlation detection between mean firing rates whereas now, given the new evidence, Hebb’s postulates can be reinterpreted in terms of causality relationships between pairs of spikes with precise firing times.

The advent of TAH has reinvigorated the interest in temporal information processing in the neural networks community. Many studies on theoretical aspects of TAH and its application to neural modelling have been undertaken in recent years. These have discovered, among others, an interesting property of TAH: under certain conditions TAH creates a balanced bimodal weight distribution, whereby weights of synapses saturate to either maximum or minimum values. These balanced bimodal weight distributions tend to normalise the output firing rate of neurons [12].

1.5 Thesis

This work presents an investigation on the implementation of temporally asymmetric Hebbian learning in analogue VLSI. The primary aim of this work is to examine the suggestion that temporally asymmetric Hebbian learning in mixed-model VLSI can support correlation learning and spike-synchrony processing.

The first work undertaken in this project was the analysis of TAH learning in regard to its analogue hardware mapping. Based on the findings of this analysis, novel electronic circuits for the implementation of synapses with spike-timing-dependent plasticity are proposed. The de-
dependence of the weight change on the current value of the weight (weight-dependent learning) is known to affect significantly the learning process. A specific aim of the design has been to include a weight-dependent mechanism in the learning circuit. The hypothesis put forward in this thesis is that hardware synapses with moderate weight-dependent STDP may allow learning weaker correlations than with purely weight-independent STDP.

One of the most striking features of TAH learning is that, under certain conditions, balanced bimodal weight distributions, that tend to stabilise the output firing rate, emerge from learning even when no correlations exist in the input data. In this respect, bimodal TAH learning is very different from classical rate-based Hebbian algorithms which create weight distributions aligned with the principal components of the inputs [13]. It is also an aim of this thesis to investigate the effect that these bimodal weight distributions have on spike-synchrony transmission in small networks of silicon neurons.

The storage of analogue-valued weights resulting from learning is the main obstacle to overcome in the implementation of on-chip neural learning algorithms. In this thesis it is suggested that TAH learning can help overcome the weight storage difficulty since the final weight distributions created are bimodal. Weights in a bimodal weight distribution can be considered essentially binary. Consequently, they are far easier to store on chip than analogue-valued weights. This thesis also explores whether it is possible to avoid extra circuitry for long-term weight retention by exploiting the fact that TAH learning creates an unstable learning process. It has been investigated if this instability can maintain a specific set of binary weights even if the correlations that created that weight configuration disappear from the input spike trains.

1.6 Thesis outline

This thesis is structured as follows:

Chapter 2 provides background information on spike-timing-dependent learning. The chapter starts with a discussion on relevant aspects of neural coding, including a brief presentation of the temporal vs. rate coding debate. Integrate-and-fire (IF) neuron models are presented next. Then, the chapter moves on to review biologically-plausible learning algorithms. An introduction to rate-based Hebbian algorithms leads to the presentation of temporally asymmetric Hebbian learning.
Chapter 3 presents a critical review of previous neuromorphic VLSI research relevant to the work of this thesis. First, analogue VLSI (aVLSI) is introduced as a tool for the implementation of neuromorphic systems. A discussion of on-chip learning follows. Several technologies and strategies used for long-term analogue-weight storage are discussed. The chapter follows with a review of circuits relevant to this project found in the neuromorphic literature. The circuits presented—some of them described at transistor level—will help the reader identify where the work of this thesis has drawn inspiration from, and where new contributions in circuit design have been made. Finally, work carried out by other research groups on aVLSI spike-based learning neurons is presented at the end of chapter.

Chapter 4 presents the circuits proposed to support the aVLSI implementation of temporally asymmetric Hebbian learning. The first part of the chapter explores the properties of temporally asymmetric Hebbian learning in relation to its aVLSI implementation. Next, the architecture of the neuron with STDP synapses is presented, followed by a detailed description of all circuits required for the implementation.

Chapter 5 starts with a description of the network of TAH learning neurons included in a test chip fabricated. The test setup used for the experiments carried out with this chip is also presented. Next, results from silicon show the spiking behaviour of the silicon neurons, the modification of the weight voltages, and the effect that those changes have on the strength of synapses. The last part of the chapter contains graphs that characterise in detail the STDP learning window implemented and the tunability of its weight-dependence component.

Chapter 6 presents results from learning experiments conducted with single neurons and the full on-chip network. First, the properties of the weight-distribution produced by on-chip STDP learning are compared with theoretical studies and results from software simulations reported in the literature. Then, several results show that a single neuron can learn temporal correlations. The chapter also presents experiments carried out to study the retention of binary weights when the training data is not continually presented to the synapses. The chapter ends with an investigation on the ability of the feed-forward network in the chip to detect a hierarchical structure of spike-timing synchrony. Results given illustrate the spike-timing synchrony detection and amplification properties of the network.

Chapter 7 summarises the contents of the thesis, recapitulates the conclusions of the work and suggests extensions to this investigation.
Chapter 2
Spiking neurons and temporally asymmetric Hebbian learning

2.1 Introduction

Most artificial neural networks process data represented by analogue variables. Similarly, most models of biological neuronal systems consider that meaningful information encoded by neurons can be captured by analogue-valued signals. These analogue-valued signals used in models of biological neurons are often computed by time averaging the irregular firing of action potentials that neurons actually produce. Clearly, the precise timing of a single spike does not have a role in these models and algorithms. Instead, they consider that information is only encoded in mean firing rates of neurons. In the last decade, data from several neuron recording experiments have shown that synaptic strength modifications are driven by precise timing differences between input and output spikes. These new findings have increased the interest on neural coding schemes which take into account the precise timing of spikes. This chapter will introduce spike-based computing and learning schemes upon which the circuits and methods developed in this thesis have been built.

The chapter starts with a discussion on the issue of neural coding. We contrast mean-firing-rate codes with alternative coding schemes based on precise spike timings. Then, the chapter deals with spiking neuron models. The integrate-and-fire neuron model is presented as a good alternative to complex conductance-based models. Next, a short presentation of rate-based Hebbian learning is given. The final and largest part of this chapter presents temporally asymmetric Hebbian learning rules which underlie spike-timing-dependent plasticity.

The analysis of the constraints that analogue VLSI imposes on the implementation of neural learning, and spike-based learning in particular, will be presented in chapters 3 and 4. There, we will look again at spike-timing-dependent learning from a "technological" point of view to identify benefits that temporally asymmetric Hebbian learning can bring to neural hardware systems.
2.2 The neural codes

For over a hundred years it has been known that brains of animals are made of many small computing units — neurons — connected to hundreds or thousands of other such units. Neurons communicate between them using short electrical pulses known as action potentials or spikes. All spikes generated by a neuron have almost the same shape and duration. Consequently, the information transmitted must be encoded in the timing of the spikes. As depicted in Figure 2.1, the sequence of action potentials (spike trains) generated by a neuron can be interpreted as a series of events fully described by the spike timings. For convenience, spike trains are often described as a sum of delta functions $\sum \delta(t - t_i)$, where $t_i$ are the action potentials timings.

![Figure 2.1: Spike train found in biological neurons generated with a Hodgkin-Huxley model simulator.](image)

It is widely agreed that signals used by neurons to communicate can be abstracted to sequences of spike timings. However, how neurons encode information in the neural spike trains is still unclear. The nature of the neural code is one of the most debated issues in the neuroscience literature. Broadly speaking, rate codes consider that the information encoded by spike trains can be found by applying different types of averaging on the timings of spikes, whereas spike codes consider the precise timing of every spike essential to the scheme that neurons are using to encode information. Some researchers suggest that the dichotomy between rate-based codes
and spike-based codes is neither important nor accurate. However, the following introduction to neural coding will be organised following these two categories to highlight the different approaches taken by researchers to tackle the neural code problem.

### 2.2.1 Rate codes

In their investigations, neuroscientists aim at finding neurons in the brain that encode specific features of a stimulus presented to an animal. For many years, the predominant strategy has been to identify neurons which encode a feature as those neurons which respond with a high spike-firing rate when the animal receives a stimulus which contains that feature. Several methods of spike averaging can be used to calculate the rates of firing [3, 13, 14].

As used by Adrian [9] in his early experiments, the most simple definition of rate code consist in averaging the firing activity of a neuron over a long time window to obtain a mean-firing-rate value. An analogue-valued signal representing this definition of rate is thus given by

\[ r = \frac{N_{sp}(T)}{T} \]  

(2.1)

where \( T \) is the duration of the averaging window, and \( N_{sp}(T) \) the number of spikes that occurred inside the window. Clearly, this time-window averaging interpretation of the neural code considers spike-timing variability essentially as noise.

Sometimes, to characterise the response of a neuron to a stimulus, neuroscientists use data from multiple recordings of the same neuron. Thus, the temporal axis can be binned with a much shorter time window to give a fast-varying analogue rate if we accumulate the spikes that occurred in all recordings for each small time bin of duration \( T \),

\[ r(t) = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{T} \sum_{i=1}^{n_j} \int_{t}^{t+T} \delta(t' - t_i^j) dt' \]  

(2.2)

where \( n \) is the number of recordings used to calculate the instantaneous rate, \( n_j \) the number of spikes for the \( j \)'th recording and \( t_i^j \) the timing of spike \( i \)'th for recording \( j \)'th. This form of averaging over many trials has been successfully used to correlate behaviour with neural activity. However, it is clear that neurons cannot implement this form of encoding since the
information for the stimulus must be present in the neuronal response for each independent trial. On sticky summer days, flies do not wait for several repetitions of our movements before deciding whether the shade that is suddenly covering them is a folded newspaper approaching at high speed or just a fast-moving cloud stopping the sun rays.

Rate definitions that consist of simple *time-window averaging* or *multiple-recording averaging* do not consider any possible collective encoding by a population of neurons. If we assume that all neurons in a large population of neurons have similar responses to a stimulus and project their outputs to another such large population, we can reuse equation 2.2 to compute *population-average rates* if we interpret $n$ as the number of neurons in the population and $j$ as the label identifying each neuron in the population.

### 2.2.2 Spike codes

#### 2.2.2.1 Speed of processing

One of the reasons that has led researchers to look for complementary coding schemes is that fast-reaction times of animals are incompatible with neurons computing with long time-averaging windows. Based on studies of the reaction time of monkeys after the presentation of a stimulus, it has been argued that in the visual pathway often neurons only have time to process a single spike for each input connection they receive from the previous processing stage [15]. Codes based on the time-to-first spike and the order of arrival of spikes (*rank-order coding*) have been proposed to account for single-spike processing [16]. In time-to-first spike coding, those neurons that fire first after the onset of the stimulus are considered to have strong sensitivity to the stimulus. The precise timing of the spike could then signal a gradation in the response to a particular feature [14]. Neurons which are stimulated less would respond with longer latencies.

#### 2.2.2.2 Phase coding

In some neuronal systems, large populations of neurons fire rhythmically. This collective oscillation may be used as the reference signal for the timing of individual spike generated by neurons in the population. The phase of each spike relative to the global oscillation could be used to encode additional information not present in the firing rate [17]. In the hippocampus of rats, this phase-coding scheme has been found to carry some information about the spatial
location of the rat inside an environment [18]. A phase-coding scheme has also been found in
the olfactory system [19].

2.2.2.3 Synchrony and the temporal binding hypothesis

Separate regions of the brain process different types of sensory features. For instance, in the
visual system the processing of colour, location or shape takes place along separate paths. The
question that arises then is how the feature information corresponding to each object present
in the visual scene is *bound* together, sometimes across distant parts of the brain, to create
a coherent representation of the object without mixing the features of several objects. It has
been argued that the brain needs a mechanism to solve this *binding problem*. An hypothesis
that has been surrounded with some controversy is that spike-firing synchrony would provide a
mechanism to bind together in time all features corresponding to the same object. As depicted in
Figure 2.2, those neurons which fire inside a short time window would be labelled as *belonging
together* to the same object [20]. Some authors suggest that spike binding in time mitigates the
combinatorial coding explosion that arises from the binding problem. The capacity of a code
based on precise spike timing with a precision of a few milliseconds is orders of magnitude
larger than for codes using coarse time-averaging firing rates.

![Figure 2.2: Two pairs of nearly synchronous spike trains are shown in the four top traces. Another two spike trains do not have any temporal cross-correlation. The temporal binding hypothesis would suggest that an object with feature1=A (e.g., colour=red) and feature2=X (e.g., shape=square) can be distinguished in the scene from another object which has feature1=B (e.g., colour=blue) and feature2=Y (e.g., shape=circle).](image-url)

Several research groups have found in their neuron-recording experiments clear evidence of
Spiking neurons and temporally asymmetric Hebbian learning

spike-firing synchrony between segregated regions in the brain [21, 22]. For instance, some recordings show that synchronised spike firing with no phase difference occurs between different functional columns of the visual system of cats [23]. Despite this recorded spike-firing synchrony across large distance in the brain, it remains to be proven that in effect these synchronisation phenomena are used as a binding mechanism.

Besides perfect spike-timing synchrony, more complex spatio-temporal patterns of precise spike timings may be used in the brain to encode information. A sequence of precise delays between the activation of an ordered series of neurons in the same area of the brain, or even across different regions, could represent some feature of the stimulus. Abeles has studied in detail under what conditions networks of spiking neurons can transmit this type of spike-timing synchrony patterns reliably [24].

2.2.2.4 Stimulus reconstruction

Experimental results analysed with information theory techniques show that it is possible to reconstruct the stimulus from the precise spike timings of the neuronal response. This stimulus reconstruction is based on a reverse-correlation procedure which consists on averaging over several trials the time course of the stimulus presented to the animal immediately before a spike [25]. These results have been very influential in the increased awareness about the potential significance of the precise timing of every single spike down to a resolution of 5 milliseconds or less.

2.2.2.5 Are rate-codes really different from spike codes?

The debate about the nature of the neural code is still intense. It is agreed that any plausible neural encoding should account for the fast reactions of animals. The rate population coding definition presented above is able to account for fast information transmission along a neuronal processing stream [26]. Thus, spike codes based on precise spike firing of single neurons may not be required to account for fast reaction in animals. However, in some systems the neuronal populations are too small to create reliable averaging [3]. It has been argued that some of the spike codes presented above can also be interpreted as alternative ways to look at rate codes. For instance, neurons with high firing rate have higher probability of generating a spike with a low latency from the onset of the stimulus than neurons with low-firing rate. Some authors
also argue that a good analysis of the neural code issue should start by accurately defining what is meant by rate code. With a more general definition of rate code it can be shown that the reverse-correlation method for stimulus reconstruction is just a special case of rate code where a special kernel function has been used for averaging instead of a rectangular window [14]. Regardless of the rate definition chosen, phase coding and spike-timing-synchrony coding cannot be explained in terms of time-averaging-rate codes.

2.3 Spiking neurons

Many studies of biological neuronal systems use firing-rate neuron models whose output is a continuous analogue signal representing the action potential firing rate. Firing-rate neurons are easier to study analytically than spiking neurons. Furthermore, the simulation of networks of firing-rate neurons are less computation intensive than detailed biophysical models of spiking neurons. Similar neurons without spiking activity are the norm in artificial neural networks. However, if we want to study spike-based codes like spike-firing synchrony, or spike-based learning rules, firing-rate neurons are not adequate. By using spiking neuron models no assumptions are made on the nature of the neural code. Both rate codes and spike codes can be studied with spiking neurons.

2.3.1 Bio-physical models

The inside of a neuron is separated from the extracellular liquid by a thin, charge-insulating membrane. The pass of current between the inside and the outside of neurons occurs at a multitude of ion channels and ion pumps present in the membrane. Ion pumps spend energy to actively move charged ions from one side to the other of the membrane. Charges can also cross the membrane through passive conductances called ion channels. Ion channels can be classified according to their selectivity for different ion species. The equilibrium between the diffusion of charges through the ionic channels and the movement of charges created by the ionic pumps creates a different concentration of ion species inside and outside the neuron. As a result, in equilibrium the inside of the neuron is negatively charged with respect to the extracellular medium. Hence, we say that the neuron in equilibrium is polarised.

Conductance-based models, first used by Hodgkin and Huxley on their models of action potential generation in the giant axon of the Squid, are commonly used to explain the behaviour
of biological neurons. Conductance-based models of neurons describe the neuron membrane as a capacitor in parallel with several voltage-dependent resistors that model the non-linear conductance of ion channels. The voltage-dependent conductances are controlled by a set of coupled differential equations describing the probability of finding an ion channel open [3]. When neurons do not have complex tree structures (dendritic trees), the longitudinal resistivity of segments of the dendritic tree is not taken into account in the models. Thus, the membrane can be described by a single capacitor to build a single-compartment model, or point neuron.

The communication between neurons occurs at the synapses. In most types of synapses, the arrival of a spike at the presynaptic terminal (the output of the sending neuron) causes the release of synaptic transmitter which increases the number of open ion channels in the postsynaptic terminal (the input of the receiving neuron). Thus, synapses can be modelled as extra conductances in parallel to the membrane capacitance. When stimulated, excitatory synapses experience a sudden increase of conductance that decays exponentially in a few msec. The resulting influx of current inside the membrane causes an Excitatory Post-Synaptic Potential (EPSP) which consists of a sudden membrane voltage increase which decays completely in a few msec.

The Hodgkin-Huxley model shows that the spike generation is due to the non-linear dependence of the ion-channels conductances on the voltage difference across the membrane. When the membrane voltage is driven by the synaptic inputs, or by a direct current injected inside the membrane during an experiment, to a voltage around $-50mV$, a sudden influx of $Na^+$ ions depolarises the membrane and creates the upswing of the action potential. When high $V_m$ values are reached, the sodium current stops and a delayed outflux of $K^+$ ions creates the downswing of the action potential.

### 2.3.2 The leaky integrate-and-fire neuron model

Conductance-based models can describe accurately the time course of the membrane voltage during and before the action potential. However, the Hodgkin-Huxley model is difficult to analyse. In the study of spiking-neuron systems we are not interested in the exact shape of the action potential, since we assume that all information is encoded by the timing of the action potential onsets. By abstracting the precise biophysical mechanisms it is possible to build spiking neuron models which are far easier to study and simulate. A very popular class of simplified models of spiking neuron is the integrate-and-fire (IF) neuron. It was first proposed in 1907 [27]. Fig
Figure 2.3 shows the schematic circuit of a leaky IF neuron, a common variant of IF neuron. The membrane is simply modelled with a membrane capacitor ($C_m$) and a leakage resistor ($R_m$). In some investigations, the leakage current in the membrane is implemented with a fixed current independent of the membrane voltage. This is often the case in hardware implementation of spiking neurons. The IF model disregards the biophysical mechanisms of spike generation. Instead, a hard threshold for spike generation is included. When the membrane voltages ($V_m$) reaches the firing threshold ($V_{th}$) a short pulse is generated at the output. Then, the membrane capacitor is reset and the integration of the synaptic current starts again. Neurons exhibit a period of almost total insensitivity to stimulation after the generation of the spike known as the refractory period. In IF models, the refractory period can be enforced either by increasing by a large amount the firing threshold for a short period after the spike, or with a slow decay of the conductance (idealised switch in Figure 2.3) that resets the membrane capacitor.

![Figure 2.3: Circuit schematic of a leaky integrate-and-fire neuron](image)

The simple leaky integrate-and-fire model does not account for many phenomena covered by more complex models. However, it predicts with enough precision the time course of the membrane voltage leading to spike initiation. Hence, it can be used to study spike-based codes and spike-based learning rules. Furthermore, as will be seen in the chapter 3, the IF model, with its explicit threshold mechanism, lends itself to straightforward electronic implementation using a simple voltage comparator. The leaky IF model has a reduced number of settings to adjust compared to conductance-based models. Only three parameters determine the behaviour of the neuron: the membrane time constant ($\tau_m = R_mC_m$), the threshold voltage ($V_{th}$) and the duration of the refractory period ($T_{ref}$). The value of $\tau_m$ will determine the type of neural encoding possible. If the membrane time constant is short, several coincident spikes are required to bring the membrane voltage to the firing threshold. Hence, neurons with membrane time constants
shorter than the mean interspike interval can be described mainly as detectors of synchronised input activity. On the contrary, if the leakage is small (large $\tau_m$), neurons are no able to respond to temporal patterns, and they can be considered predominantly as temporal integrators [28].

2.4 Hebbian learning

It is believed that changes in animal behaviour induced by experience and conditioning are due to physiochemical modifications that take place in neural synapses [3]. In studies of artificial neural networks, the term learning is used to describe the algorithms and mechanisms that modify the connectivity between computing units (neurons). Other developmental mechanism found in biological systems such as the creation of new synapses, growth of nerve cells, and network rearrangement due to nerve displacements, are rarely considered. Thus, learning in ANN involves changing the strength of connections. The general aim of any learning process is to create a system that will respond in the desired manner to previously unseen patterns which resemble those presented to the system during learning. Researchers working on biologically plausible models of learning and memory often favour the term synaptic plasticity to refer to synaptic strength modification.

Many studies on synaptic plasticity have been based on the postulate that Donald Hebb put forward in 1949 [11],

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.

Traditionally, Hebb's postulate has been interpreted in terms of correlations between firing rates. In so-called Hebbian learning rules, synaptic plasticity is based in the detection of coincident activity between the input and the output of the neuron. Presynaptic neurons which fire at high frequency at the same time as the postsynaptic neuron will have their connections increased. The essential idea behind this traditional interpretation of Hebb's postulate is captured by the expression

$$\frac{dw_{ij}}{dt} = \alpha r_i r_j$$

(2.3)

which tells us that the rate of change of the weight of a connection ($\frac{dw_{ij}}{dt}$) depends on the product of the presynaptic ($r_j$) and postsynaptic ($r_i$) firing rates modulated by a constant $\alpha$ [14].
Equation 2.3 defines a positive feedback mechanism between the output activity and the strength of the connection which can make all synaptic weights grow without bound. Sometimes hard-saturation limits are used to restrict weight growth. Weights can also be bounded with soft limits if the amount of weight potentiation decreases when the weight approaches the maximum value [14]. An example of this type of weight-dependent weight update is given by the equation

\[
\frac{dw_{ij}}{dt} = \alpha(w_{\text{max}} - w_{ij})r_ir_j
\]  

(2.4)

Hebb's postulate does not explicitly state any complementary mechanism to weaken the strength of connections. A simple way to have weight depression is to include weight decay in equations 2.3 and 2.4 by introducing a depression term independent of the pre and postsynaptic activity. However, simple weight decay cannot stop all weights from growing to the maximum allowed value in some situations. A synaptic weight distribution with all weights at the same value cannot perform any type of discrimination task on the input. Therefore, some sort of competition mechanism is required to force some weights to decrease when others increase.

Competition between synaptic weights is important for any learning rule used to create input selectivity. The following Hebbian learning rule proposed by Oja creates input competition using information local to each synapse [29],

\[
\frac{dw_{ij}}{dt} = \alpha(r_ir_j - w_{ij}r_i^2)
\]  

(2.5)

It can be shown that equation 2.5 creates a normalised synaptic weight array ($\sum_j w_{ij} = 1$). Weight normalisation imposes a competition between the input weights since the total strength of the synaptic array is limited. Hence, some weights have to decrease so that others can be potentiated. We will return to this idea of synaptic input competition with local learning rules in section 2.5, where the properties of spike-based weight-independent learning rules will be discussed.

### 2.5 Spike-timing-dependent plasticity

Some of the most compelling evidence to support the idea that precise spike timings play a role in the information processing carried out by biological neural systems comes from experiments
Spiking neurons and temporally asymmetric Hebbian learning

on synaptic plasticity. In the last decade, data from neuronal recordings have shown that in several systems synaptic strength modifications depend on precise spike-timing differences between presynaptic and postsynaptic spikes [30–32]. These new forms of synaptic plasticity are usually known with the term *spike-timing-dependent plasticity*, or STDP for short.

2.5.1 The learning window

Figure 2.4 shows the synaptic strength modification window found in some STDP experiments. When a presynaptic spike arrives to the synapse a few msec before the generation of the postsynaptic action potential, the synapse undergoes potentiation (i.e., the weight of the synapse is increased). An abrupt change in the direction of weight change occurs if the order of the presynaptic and postsynaptic spike is reversed. When a presynaptic spike is fired immediately after the postsynaptic spike, the synapse is depressed. Beside the abrupt transition of the weight change curve when the delay \( t_{\text{pre}} - t_{\text{post}} \) changes sign, another characteristic of STDP is the smooth decay of the learning window for both potentiation and depression when the delay between spikes \( |t_{\text{pre}} - t_{\text{post}}| \) increases. The weight remains unchanged if the delay is larger than 10 to 50 msec depending on the system [33].

Due to the spread of the data points collected in neural recording experiments, it is difficult to identify the exact shape of the STDP learning window. The shape has different characteristics depending on the system recorded [33]. Additionally, it has been shown in [34] that the same biophysical mechanism might explain the presence of both STDP learning and correlation-based Hebbian learning in different locations of a single neuron. To reduce the variety of learning window shapes found in biology, theoretical and simulation studies often describe STDP learning using a weight-change curve with exponential decay [14, 35],

\[
W(s) = \begin{cases} 
A_{\text{pot}} e^{s/\tau_{\text{pot}}} & : s \leq 0 \\
A_{\text{dep}} e^{-s/\tau_{\text{dep}}} & : s > 0,
\end{cases}
\]

and a weight update rule [36]

\[
\frac{dw_{ij}(t)}{dt} = S_j(t) \int_0^\infty W(s)S_i(t-s)ds + S_i(t) \int_0^\infty W(-s)S_j(t-s)ds
\]

where \( s = t_{\text{pre}} - t_{\text{post}} \) is the delay between the presynaptic and postsynaptic spikes.
Spiking neurons and temporally asymmetric Hebbian learning

Figure 2.4: Prototypical shape of an STDP learning window. Weight changes ($\Delta W$) are driven by spike-timing differences between the presynaptic ($t_{\text{pre}}$) and the postsynaptic spikes ($t_{\text{post}}$).

$S_j(t) = \sum_j \delta(t - t_{\text{pre}}^j)$ and $S_i(t) = \sum_i \delta(t - t_{\text{post}}^i)$ are the presynaptic spike train and postsynaptic spike train, respectively. This learning rule suggests that learning is driven by temporal correlations between presynaptic and postsynaptic spikes which may be separated by other spikes. For instance, in a sequence of firing pre-post-pre-post, the interaction between the final postsynaptic spike and the first presynaptic spike may induce potentiation in the weight. In other STDP learning rules proposed, weight changes are only driven by interactions between a postsynaptic (presynaptic) spike and its nearest preceding and posterior presynaptic (postsynaptic) spikes [37, 38].

In most so-called Hebbian learning algorithms weight changes are driven by correlations between pre- and postsynaptic firing rates. Hence, they interpret Hebb’s postulate in terms of coincidence detection; when 2 neurons are active at the same time inside a time window the
weight is increased. The relative timing of the spikes is not taken into account. In the light of STDP, Hebb’s postulate can be reinterpreted in terms of *causality*. If we consider a leaky integrate-and-fire neuron, those inputs which receive spikes immediately before the postsynaptic action potential are those which contributed the most to bring the postsynaptic membrane voltage to the firing threshold. The essential idea behind Hebb’s postulate is that the more an input contributes to the generation of the action potential, the more it should be reinforced. Thus, the learning rules underlying STDP are probably truer to Hebb’s original idea than traditional rate-based Hebbian algorithms.

Hebb’s postulate does not suggest a complementary mechanism for synaptic depression. Interestingly, the asymmetric form of STDP presented in Figure 2.4 weakens synapses which receive inputs without a relation of causality with the postsynaptic spike. In other words, depression is driven by *a-causal* interactions between pre and postsynaptic spikes. STDP learning rules with asymmetric learning windows with such a strong distinction between *causal* and *a-causal* spike interactions as shown in Figure 2.4 are sometimes referred to as *temporally asymmetric Hebbian* learning rules.

### 2.5.2 Weight-dependent vs. weight-independent learning

As with rate-based Hebbian learning, bounding the range of allowed weight values is also important in STDP to stop unlimited weight growth. Some STDP learning rules constrain the weights with *soft bounds*. When weights approach the maximum (minimum) allowed weight value, the amount of potentiation (depression) they receive is lower. A simple way to implement soft-bounds consists in modifying the learning window defined in equation 2.7 with

\[
A_{pot}(w_{ij}) = (w_{max} - w_{ij})a_{pot}
\]  

(2.8)

\[
A_{dep}(w_{ij}) = -w_{ij}a_{dep}
\]

(2.9)

where \(a_{pot}\) and \(a_{dep}\) are two constants [36]. In this way, the soft-bounds are implemented by making the learning window *weight dependent*; i.e., the amount of potentiation and depression depends on the current value of the weight.

Alternatively, the learning rule can impose *hard bounds* on the weights. With this approach, weights are simply never allowed to increase (decrease) beyond a maximum (minimum) weight
value \( w_{\text{max}} (w_{\text{min}}) \). Therefore, the weight change is said to be weight independent since the learning window is the same for the whole range of allowed weight values (see Figure 2.5A).

Data from neuronal recording experiments suggest that in some systems there is a soft bound for potentiation and a hard bound for depression [37]. The shape of such a weight-dependent learning window is given in Figure 2.5-B. Strong synapses receive less potentiation than weak synapses, whereas the learning window does not depend on the current value of the weight in the depression region.

![Figure 2.5](image)

**Figure 2.5:** (A) For weight-independent learning the weight-change is the same for the whole range of weight values. (B) In weight-dependent learning synapses with strong weights receive less potentiation than weak synapses. (C) Sketch of the weight distribution for weight-independent STDP. (D) For weight-dependent STDP a smooth unimodal weight distribution arises as seen in the diagram.

The presence of weight dependence in the learning window has a strong influence on the weight distribution that emerges from the learning process [12, 37, 39]. Provided that the area under the curve of the learning window in the potentiation region is smaller than for depression
Spiking neurons and temporally asymmetric Hebbian learning

$(A_{pot} < A_{dep}$ in Figure 2.5-A), weight-independent learning develops a bimodal weight distribution, with most weights close to either the top or bottom weight-saturation limits [12]. Figure 2.5-C shows a diagram of the typical weight distribution histogram for weight-independent STDP. It exhibits two separate peaks at the maximum and minimum weight-saturation limits. This bimodal weight distribution results from the strong competition between synapses to control the generation of postsynaptic action potentials. When a group of synapses is able to drive the membrane potential of the postsynaptic neuron above the firing threshold they will be all strengthened. In consequence, they will become more likely to generate an action potential collectively. On the other hand, synapses which have not been potentiated become less likely to cause a postsynaptic action potential. The non-potentiated synapses will have and increased probability of receiving a presynaptic spike after an action potential. Thus, due to the imbalance between the areas for the potentiation and depression regions of the learning window, the weights of these unpotentiated synapses are likely to decrease to the minimum weight value [35].

A very different weight distribution is created by weight-dependent learning rules such as that depicted in Figure 2.5-B. The weight distribution becomes smooth and unimodal. As shown in Figure 2.5-D, weight-dependent learning creates a distribution with a single peak. The reinforcement of already strong synapses in weight-independent learning creates a highly unstable learning process. In contrast, the strong competition between synapses is suppressed if weights that are already strong undergo less potentiation [37].

2.6 Learning with STDP

2.6.1 Inputs without temporal correlations

It is common to use spike trains generated from an homogeneous Poisson process (i.e., with constant mean firing rate) to study the properties of different types of STDP learning schemes. Using this type of input spike trains, it has been shown that even when no correlations are present at the input (i.e. input spike trains are purely random) the strong competition imposed by weight-independent learning creates bimodal weight distributions. The balanced bimodal weight distribution arising from weight-independent learning can normalise the output firing rate so that it becomes fairly insensitive to firing-rate changes at the input [12].

It can be shown that rate-based Hebbian learning algorithms can model the long-term effect of
Spiking neurons and temporally asymmetric Hebbian learning

STDP learning if presynaptic firing rates change in a time scale much larger than the effective STDP time window for weight modification [14, 40]. Therefore, under these conditions, rate-based Hebbian learning and temporally asymmetric Hebbian learning may be used to explain synaptic plasticity at different time scales.

2.6.2 Synchrony detection

Contrary to rate-based Hebbian models, spike-timing-dependent plasticity can be driven by precise timing correlations between presynaptic spike trains. Hence, there is interest in investigating the effect of STDP when spike trains carry information with spike codes like those discussed in section 2.2.2. If neurons act as coincidence detectors, STDP will clearly reinforce synapses which receive synchronised activity. Inputs which receive spikes within a short-time window will be able to cause a postsynaptic spike inside that window. Therefore, STDP will reinforce those neurons which show cooperation in time. In weight-independent STDP, the learning process is mainly driven by spike synchronisation and becomes insensitive to mean firing rate differences between the inputs when the presynaptic spike trains are temporally correlated [12].

2.6.3 Synchrony and delay lines

More complex timing patterns can be detected with STDP learning if the arrival of synchronised spikes to a neuron is combined with spike transmission delays between neurons. Lets take two groups of neurons which fire at instants $t_i$ and $t_j$. Both groups project their outputs to a separate neuron $k$ along transmission paths with propagation delays $\Delta_i$ and $\Delta_j$, respectively. If the propagation delays compensate for the difference in firing times (i.e., $t_i + \Delta_i = t_j + \Delta_j$) the time or arrival to neuron $k$ of both volleys of spikes is matched. Thus, if the target neurons is equipped with STDP learning it could learn to respond to a timing difference $(t_i - t_j)$ between the two groups equal to $(\Delta_j - \Delta_i)$.

Learning driven by the synchronisation of spikes that go through a bank of delay lines has been proposed to explain the development of the sound-source localisation system of the barn owl [41]. It is well known that the localisation of the sound source is carried out by an array of spike-coincidence detectors. The delay for the sound to reach the left and right ears is compensated by the internal spike propagation delays of the axons that project to the neurons in the
array. However, the disparity of axon delays broadens too much the window of spike-timing synchrony that reaches the neurons working as spike-coincidence detectors. STDP learning can prune inputs with delays that are too far from the mean delay, so that the input synchrony is enhanced. This delay-line selection technique can be extended to an array of neurons with lateral inhibition to cluster data encoded with spike-timing delays [42]. With this type of encoding, neurons with STDP synapses operate as radial-basis-function neurons (RBF) in the time domain.

2.6.4 Conditioning, prediction and sequence learning

Besides cooperation, the other important feature of temporally asymmetric Hebbian learning is the causality detection that stems from the temporal order discrimination which determines the direction of weight change. Establishing cause-effect relations between events is essential for making predictions. In behavioural psychology, the generation of predictions is investigated with conditioning experiments. (In the famous classical conditioning experiments by Pavlov, a dog is seen to salivate strongly when a bell rings if in the past the bell has been rung just before food was given to the dog.) It has been suggested that predictions on the scale of seconds, as happen in conditioning, may be explained by the causality detection implemented by STDP at the millisecond time scale. A neuron may receive synchronised spikes from neurons encoding two events which happen a few seconds apart if spikes are propagated through a long polysynaptic chain [33]. Alternatively, delayed spikes could also be provided by reverberating loops. At a much faster time scale, asymmetric STDP can also be used to learn sequences of events. Furthermore, when the spatial structure of the dendritic tree is considered, a fast temporal difference rule similar to those used to model conditioning in behavioural psychology is equivalent to temporally asymmetric Hebbian learning [43].

2.6.5 Fast reaction neurons

Accounting for fast processing in networks of spiking neurons is one of the reasons that has led to the proposal of several spike-based codes. TAH learning can explain how neurons become tuned to work with spike-timing information. Some studies have shown that TAH learning can select those inputs of a neuron which receive spikes with less latency, thus reducing the reaction time of the neuron [12]. Synapses stimulated with low latency will always receive spikes before the postsynaptic action potential. Therefore, these reinforced inputs will be able to drive the
neuron membrane above the firing threshold earlier, without requiring the contribution of the long-latency synapses, in subsequent stimulations. The same effect has been used to explain why place field neurons (neurons which encode spatial location) in the rat hippocampus become narrower and early predictors of the position of the neuron after learning [44].

2.7 Summary

Neurons communicate through sequences of short electrical pulses commonly called spike trains. Simple integrate-and-fire neuron (IF) models can be used if we abstract the detailed biophysical mechanisms involved in the generation of spikes. IF neurons are built with a leaky capacitor that integrates the synaptic input currents. When the voltage across the capacitor reaches a firing threshold an output spike is generated. Most neural computation models are based on mean firing rates of neurons. In the last two decades several complementary spike-based codes that consider the precise timing of the spikes have been proposed.

Most Hebbian learning algorithms modify the strength of synapses based on firing-rate correlations between presynaptic and postsynaptic neurons. Recent experiments have uncovered new forms of synaptic plasticity driven by precise timing differences between presynaptic and postsynaptic spikes (spike-timing-dependent plasticity, or STDP). The learning rules underlying some forms of STDP are often termed as temporally asymmetric Hebbian learning (TAH). In TAH learning, a synapse is strengthened when a presynaptic neuron fires a few msec before the postsynaptic neuron. In contrast, a synapse is weakened if the presynaptic spike arrives within a short time window after the postsynaptic event. An important feature of STDP learning which affects its computational properties is the presence of weight dependence in the learning rule (i.e. when the magnitude of the weight change depends on the current value of the weight). The advent of STDP has reinvigorated the interest in spike-based neural coding.
Chapter 3

Neural hardware: the analogue way

3.1 Introduction

This chapter presents a critical review of electronic circuits used in neuromorphic spiking systems. The discussion focuses on those circuits found in the neuromorphic literature which are most relevant to the research presented in this thesis. The aim is not to review exhaustively all contributions found in the literature, but rather to highlight the issues that have already been successfully addressed by past research, and to pinpoint the standing challenges in the design of neuromorphic electronics.

The chapter starts explaining what makes analogue VLSI (aVLSI) technology well suited for neuromorphic engineering research. Next, the chapter deals with some general issues affecting neuromorphic aVLSI design: (1) the speed of processing, (2) on-chip learning and weight storage, and (3) spike-based communication and computing. The following two sections describe in detail some synaptic and IF circuits. The chapter ends with a review of other investigations on the aVLSI implementation of spike-based learning.

3.2 Neuromorphic aVLSI

Neuromorphic systems attempt to replicate both the functionality and the structure of neurobiological systems [5]. Researchers working in neuromorphic engineering take into account the physical nature of the computational elements required by the algorithms from the early stages of the design. This approach to the design process is based on the fact that the computation schemes found in biological systems have been influenced as much by the physical properties of the computing elements, as by the performance of the algorithms.

Analogue VLSI designers build small integrated information-processing systems applying basic physical principles such as conservation of charge. In analogue VLSI the laws of physics are very evident during the whole design process. Hence, some analogue designers have naturally been attracted by neuromorphic engineering. Working on neuromorphic systems, they
can attempt to turn non-ideal characteristics of transistors and other devices (non-linearities, subthreshold currents, etc.) into resourceful features to implement functions which resemble those found in neurobiology. Furthermore, using aVLSI for neuromorphic systems is encouraged by the existence of well-documented techniques to build integrated sensors in silicon — image sensors in particular [45].

The operation of electronic circuits, when observed at its most fundamental level, consists in the controlled movement of charge between different parts of the circuit. Similarly, ionic channels—which can be considered as the rough counterparts of transistors in hardware—control the flow of charged ions across the membrane in biological neurons. Thus, some basic operations found in neurobiology can be replicated easily with simple design techniques used in analogue hardware. Notably, the addition of currents generated by the input synapses is very simple to implement in analogue electronics with a simple wire. In [5], Carver Mead highlighted the fact that the movement of charges in the channel of MOS transistors in weak inversion (subthreshold mode) is the result of essentially the same physical principles that determine the flow of charges across the membranes of neurons. However, exploiting this fact explicitly to build neuromorphic VLSI circuits has proved to be difficult. Neuromorphic circuits designed over the last two decades combine standard analogue VLSI design techniques (current mirrors, push-pull inverters, etc) with basic building blocks developed by neuromorphic engineers. This "library" of neuromorphic building blocks includes the current-mirror integrator [46] (a compact circuit widely used to introduce dynamics in silicon neurons), adaptive pixels [45] used in silicon retinas [47] and Mead's axon-hillock circuit [5].

3.3 Silicon wants to go fast

Time constants found in biological neurons are of the order of milliseconds. For instance, the specific time constant of the neural membrane, $\tau_m = R_m C_m$, in living animals is thought to be of the order of tens of milliseconds [3]. These time-constant values are perfectly suited for the processing of vision and sound signals. Analogue VLSI circuits are commonly used to implement signal processors that operate with much shorter time constants than those found in biological systems. However, most neuromorphic designs are being used in systems that process signals coming from visual or audio sensors, which resemble those that allow animals to interact with their environment.
The resistivity of the neural membrane is \(2.5 \times 10^{11} \Omega \cdot cm\) and its capacitance is approximately \(1 \mu F/cm^2\) [3]. In contrast, the CMOS process used for the circuit presented in this thesis provides a maximum capacitance of \(0.276 \mu F/cm^2\) (using the thin oxide of MOS transistors to build capacitors) and a high-resistance poly layer (RPOLYH) with \(1.2 K \Omega/\square\) [48]. To implement a 10 or 20 milliseconds membrane time constant using these two elements requires too much silicon real estate. Another reason for the low processing speeds of "wetware" is that ionic currents involved in the dynamics of biological neurons have peaks smaller than 1 nA. On the contrary, current levels in standard electronics circuits (with transistors operated in strong inversion) are in the order of μA.

A common technique employed to create slow dynamics in neuromorphic systems consists in using sub-nA currents generated by MOS transistors operated in weak inversion. A small number of neuromorphic researchers have build long time constants with switched-capacitor circuits. Despite these solutions, creating long time constants in neuromorphic aVLSI is still a difficult task. Paradoxically, whereas much effort is put by the mainstream electronic industry in designing ever faster electronic circuits, neuromorphic engineers have had to devise strategies to slow down some of their circuits.

3.4 On-chip Learning

Many ANN learning algorithms are difficult to implement in analogue hardware [49]. Analogue implementations of learning systems should target neural algorithms which do not impose severe requirements in terms of accuracy and offsets. In other words, learning to be carried out in aVLSI should be more neuromorphic; the algorithms chosen should be able to cope with imperfections and inaccuracies of the physical computing elements as in biology. Furthermore, learning should be carried out either on-chip or with chip-in-the-loop training (the forward-pass takes place in the chip but the weight-update calculations is done off-chip by a digital computer), so that circuit non-idealities are taken into account during learning.

Neural algorithms with so-called local learning rules are amenable to analogue hardware implementation. In this class of learning rules, the data needed to calculate the weight update is local to the synapse. Many ANN algorithms involve weight-update calculations that require data from across the network. In contrast, biologically inspired Hebbian-type algorithms, that only required the input to the synapse and the output of the neuron, are more suitable to
analogue hardware. Despite the differences between the planar structure of VLSI and the three-dimensional nature of brains, routing restrictions favour local learning rules both in biology and in neuromorphic aVLSI.

Most neuromorphic designs target autonomous systems that would benefit from adaptation to a changing environment. For this type of applications, where constant (on-line) learning is required, on-chip learning is sometimes the only possible option.

3.4.1 Weight Storage

The main difficulty for on-chip learning is the storage of the analogue weights that result from learning. Ideally, we would like to have a mechanism with easy weight-strength modification and permanent storage. Unfortunately, these two requirements are difficult to meet at the same time with present technologies.

In analogue VLSI, storing charge in a capacitor provides the simplest mechanism for the modification of analogue weights. Capacitors are easy to build in VLSI and have excellent charge-retention properties. The flow of charge in and out of the capacitor plates can be easily controlled with a single MOS transistor used as a switch. Unfortunately, MOS transistors have some non-ideal effects which cause the charge stored in the weight capacitor to leak. As can be seen in Figure 3.1, transistors suffer from subthreshold currents and, more importantly, leakage currents due to parasitic reverse-biased diodes formed between the drain and source diffusions and the bulk of the transistor. Another drawback of capacitive weight storage is the limited number of synapses that can be integrated on-chip due to the large area required to build a capacitor of a few $pF$.

![Figure 3.1: Charge leakage in a weight capacitor.](image-url)
Neural hardware: the analogue way

To overcome the weight degradation of capacitive weight storage, several solutions and alternatives have been proposed:

- **Floating gates.** Trapping charges in an isolated transistor gate is a technology used to implement non-volatile digital memories. Charges can be moved in and out of the isolated gate by means of hot electron injection, Fowler-Nordheim tunnelling or UV-induced conduction. Floating gates have also been applied to store analogue values on-chip. Programming floating gates with analogue values usually involves using a control loop. Some researchers have also proposed floating gates implemented with standard CMOS processes for on-chip learning [50]. However, floating gates in standard CMOS processes are still not ready for non-expert use, since it is a very temperamental technology which is difficult to implement reliably. An important challenge in the implementation of on-chip learning with floating gates is that different, unmatched physical phenomena are used for injecting (hot-electrons) and extracting (tunnelling) electrons from the floating gates.

- **Digital storage.** It should be obvious to any electronic engineer that a possible solution to the problem of analogue weight storage consists in refreshing capacitors with the output of a D/A converter whose output is multiplexed in time between several weight capacitors. This techniques is most appropriate for off-chip learning applications, whereas can be difficult to implement for on-chip learning without disturbing the learning process.

- **Local refresh.** Analogue memories with a local refresh mechanism with incremental correction have been proposed in [51]. The value of the analogue memory is quantised and compared to a set of fixed levels. Then, a small increment or decrement on the analogue memory pushes the weight slightly toward the closest fixed level. This technique has been used to implement several ANN learning algorithms in hardware. It does not require D/A converters but needs a quantising unit to compare the analogue memory to a set fixed levels. The quantising unit is often built around an A/D converter and may be too large to be included at each memory point.

The view put forward in this thesis is that neuromorphic engineers should also attempt to devise strategies at the algorithmic level that render the weight-capacitor leakage unimportant for the learning process. For instance, algorithms that combine long-term stable binary synapses with short-term analogue values. In section 3.8, important work in this direction found in the literature will be discussed with more detail [52]. An analysis of the weight-retention properties of
Neural hardware: the analogue way

3.5 Spiking hardware

Communication through short electrical pulses is one of the most prominent characteristics of biological neuronal systems. Already some of the early VLSI implementations of neural networks proposed to use streams of pulses for signal representation [53]. In analogue neural hardware, pulse-stream techniques have been used to transmit information between computing units that are made of a combination of digital and analogue circuits. The first proponents of the pulse-stream approach were clearly inspired by biological neurons. However, most research in this area has concentrated on the technological benefits that different pulse-signalling modulations can bring to the hardware implementations of neural networks [54]. Pulse representation is robust against interferences. It allows for very compact analogue multipliers for some modulation schemes (e.g., pulse-width and pulse-rate modulation). Furthermore, pulsed signals are easy to multiplex and route. Despite the implementation benefits that pulse representation can provide, most pulse-stream research did not fully embrace the spike as an essential part of the computation. Pulse-stream techniques have been mostly used to implement ANN algorithms which are based on continuous analogue-valued signals [55]. These algorithms were clearly not designed with pulses in mind.

Spikes have been used in a wide range of neuromorphic aVLSI systems since the inception of this discipline. Several circuits which emulate the behaviour of spiking neurons have been proposed (see a description of silicon integrate-and-fire neurons in section 3.7). Beside the spike trains generated internally by the hardware spiking neurons, timed events are already very evident in the stimuli of applications where neuromorphic chips are used (e.g., edge detection in silicon retinas [56]).

3.5.1 Spike transmission

As in biology, voltage pulses are an efficient way to transmit information over long distances in electronic systems. Routing signals between neurons inside an electronic chip is harder than in the brain due to the planar technology currently used to produce integrated circuits. The limitation on point-to-point communication becomes very severe when spikes have to be transmitted in and out of the chip. (A typical chip package used for prototyping has less than...
Neural hardware: the analogue way

about a hundred pins, although larger pin counts are also available.) Several communication protocols and circuits which use address-event representation (AER) to encode spike trains have been proposed to increase the input/output bandwidth of chips in spiking systems [57]. Chips send and receive addresses through an asynchronous digital bus. Emitted spikes are labelled with the address of the emitting neuron. To route spikes to neurons inside a chip, spikes need to be labelled with the internal address of the receiving neuron. Since spikes are self-timed events, no time stamp is required on the addresses as long as the bus system, which includes an address-conversion routing table, can maintain a high throughput. Too much distortion is introduced in the information encoded by the spike timings if the AER bus has low bandwidth.

3.6 Synapses

In this section, several synapse circuits are analysed. This is not an in-depth review of hardware synapses. Instead, the synapse circuits presented have been chosen to illustrate the main issues that need to be considered in the design of an aVLSI synapse.

Sometimes synapses are classified as learning or non-learning. Non-learning synapses have a constant long-term strength whereas learning synapses have their nominal long-term strength modified according to the past activity of the network. Here, we use the term synapse to refer only to the circuits which inject into the membrane a packet of charge proportional to the weight when a presynaptic spike reaches the synapse. The learning circuits needed to change the weight value will be considered in section 3.8.

3.6.1 Simple synapses

As their biological counterparts, analogue hardware synapses transform the output of one neuron (the spike) into the input (synaptic current) of another neuron. In biology, spikes received by an excitatory synapse cause a sudden increase of the synaptic conductance which returns slowly to its resting conductance value. The increase of conductance causes an inflow of current to the neuron which depolarises the membrane.

In electronics, devices that create a current output as response to an input voltage are called transconductors. Using the electronic design terminology, synapses are transconductors that
introduce a decaying peak of current inside the membrane every time they receive an action potential. Fortunately, MOS transistors are very good transconductors. An input voltage applied to the gate, creates an output current through the channel. Furthermore, the gate of a MOS transistor is very well isolated from the other terminals. Thus, implementing in analogue VLSI the essential behaviour of a synapse seems an obvious task.

Some simple synapses are shown in Figure 3.2. In Figure 3.2-A a constant current set by the weight voltage $V_w$ flows through N1-N2 when a spike reaches the synapse. A well-known problem that occurs when switching analogue circuits is the noise caused by clock-feedthrough (in this case due to the gate-drain overlap capacitance of N2). The circuit shown in Figure 3.2-B can be used to protect the membrane capacitor connected to the output node of the synapse from this problem. Transistor N4 reduces the clock feed-through considerably. However, in both A and B the charge accumulated in the parasitic capacitance at the source of N2 and the drain of N1 (N3 and N4 for the circuit in B) will create a transient error in the ideal pulse of current. This parasitic capacitance will limit the minimum change that can be introduced in the output capacitor. The same phenomena needs to be taken into account in learning circuits when injecting current into a weight capacitor. The charge accumulated in parasitic capacitances will determine the minimum weight change possible and hence the resolution of the weight updates.

A possible solution to the parasitic capacitance is shown in Figure 3.2-C. When $V_w$ sets N6 in weak inversion, the strong currents (in the order of $\mu$A) of the inverter suck the charge in the parasitic capacitance much faster than the time required by the circuit in Figure 3.2-B to recover from the transient. Note that in both B and C the on resistance of the transistor switch
Neural hardware: the analogue way

connected to the source of the current-setting transistor (N4 and N6) may cause mismatches between synapses.

The synapses shown in Figure 3.2 should be used when a compact synapse without dynamics is desired and the value of the weight is fixed. Since the V-I transfer function (the transconductance in analogue circuits parlance) of a MOS transistor in saturation is highly non-linear (exponential for weak inversion and quadratic for strong inversion) other solutions might be more appropriate for systems with learning synapses. Note, however, that some researchers see the exponential and quadratic transfer functions of MOS transistors as beneficial to the dynamic range of the weights [58].

3.6.2 Linear synapses

Linear voltage-to-current converters have been used to implement hardware synapses with variable weights. For pulse-stream implementations with frequency or pulse-width modulations, linear synapses with 2-quadrant multiplication (i.e., with weights that can take positive and negative values) have been built with the compact transconductance multiplier circuit shown in Figure 3.3 [59]. A pulse of current proportional to the weight of the synapse ($V_w$) is generated when N3 is switched on by the input voltage pulse. Bias voltages $V_{\text{loref} \cdot}, V_{\text{hiref}}$, and $V_{\text{midref}}$ keep transistor N1 and N2 in their linear region of operation (also known as triode or ohmic region). The current of a MOS transistor in ohmic region is linearly proportional to its $V_{GS}$ voltage. The output current of the synapse ($I_{\text{synapse}}$) is the difference between the drain currents of N1 and N2 set by the bias and weight voltages. The resulting current expression is given by

$$I_{\text{synapse}} = \mu C_{\text{oxx}} \frac{W_1}{L_1} (V_{GS1} - V_{GS2}) V_{DS1}$$  \hspace{1cm} (3.1)

where $V_{GS1} - V_{GS2}$ represent the weight of the synapse. Note that both N1 and N2 need to be of the same size and the biasing voltages carefully chosen for the proper operation of the circuit.

The circuit needs a fixed output voltage into which the current ($I_{\text{synapse}}$) is injected. This requires an I-V converter, built using a relatively high-gain and low-offset amplifier, to sum the contributions from all synapses in the input array. Hence, although the 3-transistor synapse is compact and simple, more complex circuitry is needed to support its operation since the
currents cannot be summed directly with a capacitor as is done in integrate-and-fire neuron circuits (see section 3.7).

Many other voltage-to-current converters can be found in any analogue integrated electronics textbook [60]. For instance, a simple differential pair could be used if we extend the input range by decreasing the transconductance of transistor in the input pair. In general, any V-I converter to be used in the implementation of a hardware synapse needs to be compact to allow for high density of synapses.

3.6.3 Dynamic synapses

Describing a biological synapse as a simple switched transconductor is a simplification. Beside the peak of increased conductance followed by a slow decay, the hardware synapses presented in Figures 3.2 and 3.3 lack the leaky integration of the presynaptic spike trains seen in biological synapses. Alternative hardware synapses with temporal dynamics similar to those found in their biological counterparts have been proposed.

The circuit of figure 3.4 illustrates the core circuit used (sometimes with additional refinements) for the implementation of many dynamic synapses [61–63]. The essential part of the circuit is the current-mirror integrator made up of P1–P2 and $C_s$ [46]. When a spike reaches the synapse, $C_s$ is discharged quickly, causing a sudden increase of synaptic current. The sources of P1 and P2 are not at the same voltage (as would be the case in the usual current-mirror configuration). Hence, when the spike ends, $C_s$ is slowly discharged through P1 causing a slow decay of the synaptic current from its peak value.
Neural hardware: the analogue way

The current-mirror integrator has been used successfully to introduce dynamics in the synapses of many neuromorphic chips. Unfortunately, the gain and time constant of the current-mirror integrator cannot be set independently of one another. Furthermore, the current-mirror integrator is more sensitive to transistor mismatch than a standard current-mirror circuit. Recently, a slightly larger alternative dynamic synapse with better parameter control has been proposed in [64].

3.7 Silicon integrate-and-fire neurons

This section presents a review of some of the circuits proposed to implement spiking behaviour in silicon neurons. Once again, this is by no means an exhaustive review. The aim is simply to present different solutions to neural activity integration and pulse generation found in the neuromorphic literature, with an emphasis on those circuits which have most influenced the circuits for the silicon neuron proposed in this thesis (cf. Chapter 4).

The spiking neuron circuits proposed by different researchers show that the summation and temporal integration of the synaptic current that takes place at the soma of biological neurons is very easy to implement using analogue circuits. Thus, early designs concentrated their attention on building robust circuits able to generate clean spikes with a repeatable shape. Some researchers have merely looked for a sound technological solution to the generation of well-shaped spikes, assuming the spiking nature of biological neurons as the level of abstraction appropriate to their investigations. Others, incorporate lower-level characteristics of biological neurons such as spike-rate adaptation and draw inspiration for their circuits from the interaction between ion channels taking part in the generation of action potentials. Another area of interest
has been the generation of temporal dynamics at the millisecond time scale. Processing the same type of signals that animals have to deal with (e.g., vision and sound signals) requires silicon neurons with long time constants. Different strategies have been devised to overcome the lack of high-value resistors in standard VLSI processes.

The spiking neuron circuits discussed here focus only on the inter-spike time intervals, which are determined by the trajectory of the membrane voltage between spikes rather than the waveform of the actual action potential. Other silicon neurons which also attempt to replicate the shape of the action potential will not be discussed [65].

The integrate-and-fire neuron model discussed in section 2.3.2 provides a good starting point for the hardware implementation of spiking neurons. For example, as will be seen shortly, the firing threshold of the IF neuron model leads naturally to an electronic voltage comparator.

### 3.7.1 Silicon IF with capacitive feedback

As well as presenting a comparison between electronic systems and neurobiology, Carver Mead’s book "Analog VLSI and Neural Systems" describes several electronic circuits for neuromorphic systems which have been used—with modifications—in many chips since its publication [5]. The integrate-and-fire neuron circuit shown in Figure 3.5 is one of those circuits. He named it the axon-hillock circuit, after the area of the neuron where the action potential is initiated and the axon is attached to the cell body.

As with most silicon spiking neurons, the input current generated by the input synapses is integrated by a capacitor ($C_{soma}$). When the voltage across the capacitor reaches the logic-threshold voltage of the first inverter, an output spike is triggered by a fast positive-feedback reaction produced with capacitive feedback around an amplifier implemented inexpensively with two push-pull inverters. The neuron self-resets through transistors N3-N4. Although the circuit does not try to model the low-level operation of biological neurons, it is obviously inspired by the principles behind the mechanism that achieves the repeatability in the shape of the spike in biological neurons; a strong and fast positive feedback initiates the upswing of the spike, which is followed by a negative feedback which resets the membrane capacitance.

The duration of the pulse generated depends on the current flowing through N4 set by $V_b$. As long as the input current coming from the synapses is much smaller than the resetting current, the duration of the spike will be well controlled.
Using a capacitive voltage divider to implement the fast positive feedback is an effective technique which has been used successfully in many neuromorphic chips. It creates a very fast upswing in the input node which makes the generation of the spike robust against noise created by the asynchronous firing of other spiking neurons in the same chip.

A drawback of the axon-hillock circuit of Figure 3.5 is its high power consumption. Most of the time, the voltage across the soma capacitor is not close to either the power or ground rails. Therefore, both P1 and N1 of the first inverter are fully on for long periods of time. The power consumption of the circuit can be improved by using current-starved inverters.

### 3.7.2 Silicon neurons with ionic currents, refractory period and spike-rate adaptation

Many other spiking neurons circuits have been proposed after Carver Mead's self-resetting silicon spiking neuron. Some spiking neurons incorporate characteristics found in their biological counterparts that were missing in Mead's neuron. For instance, some neurons proposed have refractory periods. Others also model spike-rate adaptation.

Figure 3.6 shows the silicon integrate-and-fire neurons presented by van Schaik et al. in [66],
which the authors used in a silicon model of the auditory pathway. This design explicitly draws
inspiration from the biophysical mechanism that creates action potentials in biology. A fast
influx of Na\(^+\) ions is responsible for the positive feedback which creates the upswing of the
spike. A delayed outflux of K\(^+\) ions brings the soma voltage back to its resting potential.

![Figure 3.6: Spiking neuron with Na-K currents and refractory period presented in [66]](image)

The input current is integrated by the soma capacitor. The membrane voltage across the soma
capacitor is compared to the threshold voltage \(V_{th}\) with a differential pair followed by a push-
pull inverter. Since \(V_{th}\) can be an external bias voltage, the experimenter in the lab can easily
modify the operation of the neuron. For instance, increasing \(V_{th}\) will increase the number of
simultaneous EPSP needed to generate an action potential.

The positive feedback in the IF circuit of Figure 3.6 is created by switching P2 on, which
creates an influx of "INa" current set by a PMOS transistor biased with \(V_{Na}\). This positive-
feedback mechanism occupies less area than the capacitive divider used in the Mead’s axon-
hillock circuit, since it is implemented with only two PMOS transistors. Note, however, that
an extra bias voltage is required. As for the speed of reaction, capacitive feedback creates an
un-tunable very fast upswing, whereas switching a current source allows for some control of
the strength of the positive feedback.

After a spike has been initiated, \(C_K\) is charged at a rate set by \(V_{K_{up}}\). For some voltage level
across \(C_k\) above the \(V_i\) of N2, the "IK" current, set by \(V_K\) on the gate of N3, is stronger than
the "INa" current. The IK current brings the \(C_{soma}\) back to its resting potential, causing the
falling edge of the spike. The current set by \(V_{K_{up}}\) determines the duration of the spike.
After the falling edge of the spike, \( C_k \) is discharged slowly through N9-N8 with the current set by \( V_{kdown} \). A slow decay of the voltage applied to N2 imposes a refractory period on the neuron, which lasts until the saturation current of N3 becomes weaker than the input synaptic currents \( I_{syn} \). The circuit mimics clearly the role of potassium currents in the generation of the spike and the long-lasting refractory period seen in biological neurons. Note, however, that in biological neurons, the refractory neuron is also due the inactivation of \( Na^+ \) currents [3].

Finally, \( V_{leak} \) causes a small charge leak on \( C_{soma} \) through N1. The leakage introduced in this way is a constant decay independent of the soma voltage, not an exponential decay as in biological neurons.

Another characteristic of biological neurons which is missing from the neurons of Figure 3.5 and Figure 3.6 is spike-rate adaptation. In [46], Boahen showed that, using a current-mirror integrator, spike-rate adaptation can be incorporated to silicon IF neurons with only 4 transistors.

The ideas introduced by Mead’s (capacitive feed-back), van Schaik’s (refractory period), and Boahen’s (firing-rate adaptation using a current-mirror integrator) have been used in many hardware IF neurons proposed in recent years [7,61–63,67].

### 3.7.3 Switched-capacitor integrate-and-fire neuron

The membrane time constants \( (\tau_m = R_m C_m) \) of biological neurons are of the order of tens of milliseconds [3]. Such long time constants are difficult to implement using primitive elements available in VLSI technologies (cf. section 3.3). As described above, constant currents created by a transistor operated in weak inversion, and current mirror integrators have been used in neuromorphic chips to create slow dynamics. Other researchers have turned to standard electronic design techniques for a solution to the generation of long time constants.

As shown in Figure 3.7, a resistive element can be created by regulating with two switches the pass of charge into a capacitor [60]. The switch transistors are controlled by two non-overlapping clocks \( (\theta_1 \text{ and } \theta_2) \) with the same frequency, so that the two switches are never open at the same time. The equivalent resistance of the switched capacitor is given by

\[
Req = \frac{T}{C} \quad (3.2)
\]
Neural hardware: the analogue way

where $T$ is the period of the clocks and $C$ the value of the capacitor. Very big resistances (up to hundreds of $G\Omega$) can be implemented with switched-capacitor techniques.

Elias and his collaborators successfully used switched capacitors to build complex dendritic trees with rich temporal dynamics that go beyond the simpler single-point IF neuron [68]. In [69], Glover et al. also used switched capacitors to implement the long time constants required for the IF neurons of a sound-segmentation system.

Switched-capacitor techniques allow for high tunability of resistance values (e.g., from $500K\Omega$ to $1000G\Omega$ [68]). Very large resistances can be build in a very compact manner. However, switching devices create a noisy environment which hinders the operation of other analogue circuits present in the same silicon substrate. Another drawback of switched-capacitor techniques is the extra routing required by clock lines and the additional power consumption due to the switching activity of the clock trees.

Figure 3.7: Large resistances can be implemented with switched capacitors.

3.8 Spike-based learning in hardware

So far, this chapter has introduced issues of general interest to the implementation of most spiking neural systems in analogue VLSI. The remainder of this chapter will review investigations specifically relevant to the work presented in this thesis which have been carried out by other research groups.

The implementation of learning systems in analogue hardware has been widely investigated. Most neural learning hardware research has focused on algorithms which have emerged from the artificial neural networks community in the last two decades. Most of these algorithms are not based on learning rules underlying synaptic plasticity in biological neurons.
Since the early pioneering work of Carver Mead and his collaborators in the late 1980s, neuromorphic aVLSI systems have been mainly used for information processing at the sensor level [5]. In the last decade, partly motivated by recent discoveries in synaptic plasticity in several neurobiological systems, neuromorphic engineers have started to investigate spike-based learning rules [52, 67, 70–82]. It is only natural that neuromorphic engineers are interested in this type of learning rules since for many years they have been working on chips which have spikes at the core of their designs.

There has been a debate among researchers working in STDP about the final weight distributions created by TAH learning rules [83]. Given the technological challenges of long-term weight-value retention in aVLSI, discussions on weight distribution are doubly relevant to the implementation of TAH learning in hardware. Hence, this review of spike-based learning hardware will also concentrate on the solutions proposed for long-term storage of synaptic weights.

Hafliger et al. wrote an early paper on the implementation of spike-based learning rules in analogue VLSI before spike-based learning rules were found to underlie some forms of synaptic plasticity in biological neurons [70]. They extended a rate-based learning rule to the temporal domain. The learning scheme they implemented was governed by the following weight-update rule,

$$w(t_{\text{post}}^m) = w(t_{\text{post}}^{m-1}) + \alpha c(t_{\text{post}}^m) - \beta w(t_{\text{post}}^{m-1})$$

where $w(t_{\text{post}}^k)$ is the time of the $k$'th postsynaptic spike. The novelty of their approach consists in the introduction of a correlation variable $c(t_{\text{post}}^k)$ at each synapse which makes the learning rule sensitive to spike timings. This correlation variable is increased by a fixed amount every time the synapse receives a presynaptic spike and then decays exponentially between consecutive presynaptic spike timings. The weights are updated every time a postsynaptic spike is generated. The correlation variable is reset to zero after it is used to compute a new weight update. The last term of the right-hand side of the learning rule equation ($-\beta w(t_{\text{post}}^{m-1})$) indicates that the weight modification depends on the current value of the weight (i.e., the learning rule is weight dependent).

Hafliger and his coauthors later refined the hardware implementation proposed in [70] and focused their investigations on the weight-vector normalisation properties of the aVLSI neurons using two synapses [72]. Their first two implementations used capacitive weight storage. How-
ever, this is not an optimal long-term weight storage since weight-dependent learning rules used for weight-vector normalisation tend to create analogue weights. Hence, in subsequent designs floating gate technology was used for the long-term storage of the analogue weights created by a very similar learning rule [71, 74]. It was found that learning was hindered by mismatches in the charging and discharging mechanisms of the floating gates [74].

Analogue VLSI circuits for a spike-based Hebbian-like learning rule with long-term bistable variables were presented in [52]. In the algorithm implemented, an internal analogue variable $X(t)$ drives the transition between the two states of a binary synapse. To ensure that the weights are long-term stable they use the refresh mechanism shown in Figure 3.8.

![Diagram of the internal analogue variable $X(t)$](image)

**Figure 3.8:** The internal analogue variable $X(t)$ has two long-term stable states.

The internal variable $X(t)$ is increased or decreased by a small learning circuit (not shown in the figure) at the arrival of a presynaptic spike. In conjunction with a comparator, the analogue variable stored in a capacitor sets the weight of the binary synapse. To make sure that the leakage in the capacitor does not modify the long-term value of the weight, small currents set with $V_{refP}$ and $V_{refN}$ drive $X(t)$ toward either the power supply or the ground rails depending on the value of the binary weight.

More recently, circuits for spike-timing-dependent learning rules which resemble more closely those found in biological systems have been proposed [77, 78, 80]. Following the ideas suggested in [52], Indiveri proposed an STDP synapse with two long-term stable states [77]. However, the long-term bistable analogue variable is used directly as the weight of the synapse. As shown in [77], the learning circuit proposed creates an asymmetric learning window with clearly separated potentiation and depression regions, but it does not have the long decay-
ing tails so characteristic of STDP learning rules used in most neuronal modelling investigations [14]. Also recently, Hafliger and Kolle Riis presented a new implementation of the spike-based learning rule proposed by Hafliger in his earlier papers which also includes extra circuitry to maintain long-term stable weights using capacitive storage. However, the new synapses are not bistable but have five long-term stable weight levels [81, 84].

3.9 Summary

Circuits to implement the main computational elements found in neurobiological neurons have already been proposed. Synapses with fixed weights can be easily implemented as switched transconductors. Summing and integrating the currents generated by several synapse is also straightforward in analogue electronics; most hardware neurons use a simple wire and one capacitor. The implementation of the firing mechanism of IF neurons has also been thoroughly investigated and good circuits are available.

Learning neural systems implemented with pulsed analogue hardware have been proposed. However, most of these pulse-stream designs consider pulses as a good technological solution for information communication but often implement algorithms which deal with analogue-valued signals. Unfortunately, the long-term storage of the analogue weights which result from many neural learning algorithms is a difficult task. Different technical solutions for long-term weight storage have been proposed but all have their merits and defects. Partly motivated by the recent discovery of STDP, several groups have started investigating the aVLSI implementation of spike-based learning rules.
Chapter 4
Mapping temporally asymmetric Hebbian learning into a VLSI

4.1 Introduction

Chapter 2 introduced temporally asymmetric Hebbian learning rules which have been proposed in recent years. The design techniques and the rationale for the analogue hardware implementation of neuromorphic systems and spiking neural networks have been presented in Chapter 3. In this chapter, we propose novel circuits for the implementation of temporally asymmetric Hebbian learning rules in analogue VLSI. We focus here on the design of the circuits. Next chapters will present results from functional tests and learning experiments carried out with a fabricated chip.

Beside the design presented in this chapter, other circuits for the implementation of temporally asymmetric Hebbian learning have been studied during this Ph.D. investigation. Test results of circuits proposed early into the project were presented at NIPS 2001 [76]. A reprint of the published paper has been included in Appendix C.

The chapter starts by listing the characteristics of temporally asymmetric Hebbian learning rules that make them suitable for analogue hardware implementation. Next, the challenges that the analogue VLSI implementation of this class of spike-based learning rules has to face are discussed. Next, some general design guidelines for the hardware neuron proposed are presented. The remainder of the chapter presents the architecture of the neuron with STDP synapses proposed and describes in detail the circuits of all its components.

4.2 The opportunities

As was discussed in Chapter 3, the benefits of pulse-based signal representation for the implementation of neural system in analogue hardware have been already investigated thoroughly. In the present section, the characteristics of temporally asymmetric Hebbian learning that make
it specially well suited for analogue VLSI implementation are discussed. Here, the analysis is
done from an electronic engineering point of view, hence the perspective is mostly technolog-
ical. The study of the learning properties of the hardware implementation will be presented in
the following chapters.

4.2.1 Locality

Learning algorithms requiring information stored or computed at distant sites across a network
are difficult to implement in hardware. The implementation of these algorithms may need signal
paths for the aggregation of information distributed over the network. This routing required
by the learning algorithms is in addition to the information channels used to route the output
activity of a computing element (neuron) to the inputs of next elements in the computing chain.
The extra signal paths required between separate parts of a chip can increase significantly the
complexity of the system. Furthermore, in some learning algorithms many variables need to be
summed or averaged which may add even further complexity to the system.

An important property of Hebbian learning rules is that all information required to modify the
strength of the synapse is local to the synapse. Only the input signal to the synapse and the
output activity of the postsynaptic neuron are required to compute the weight modification.
Some learning rules which add to Hebbian learning an explicit subtractive normalisation of
the weights (i.e., the weight-update equation includes a term which depends on the sum of all
weights or all inputs of the synaptic array) lose the property of locality. Temporally asymmetric
Hebbian learning maintains the locality property of Hebbian learning; weight modifications
depend exclusively on the timing of the presynaptic and postsynaptic spikes. Thus, the same
input spike that activates the hardware synapse can be used in a nearby circuit to modify the
strength of that same synapse.

4.2.2 Simple learning window

As was shown in Figures 2.4 and 2.5, the learning window of temporally asymmetric Hebbian
learning rules exhibits two peaks which decay smoothly toward zero when the delay between
the presynaptic and postsynaptic spikes increases. Different shapes for the learning window
have been used to study TAH learning. Often, an exponential decay is used to describe the
tails of the learning window (see equation 2.6). However, rigid precision in the mathematical
description is not required for the effectiveness of TAH learning. Thus, the loose requirement in the precise function implemented by the learning window gives flexibility to the design of the learning circuits.

4.2.3 Signal averaging and summation

Learning algorithms for neuromorphic systems should exploit the spike trains and spike-based circuits commonly used in these kind of systems. Clearly, rate-based learning schemes are not the best match for spiking aVLSI neurons. To implement traditional rate-based Hebbian learning algorithms with silicon spiking neurons, extra circuits need to be included to average, possibly over long time windows, the input and output activity of each neuron. Fortunately, temporally asymmetric Hebbian learning offers an alternative learning scheme where long time constant averaging is not required. In section 2.5, we saw that some TAH learning rules compute the weight change taking into account all presynaptic (postsynaptic) spikes generated prior to a postsynaptic (presynaptic) spike. This all-to-all spike interaction scheme can be formulated mathematically in a concise manner (see equation 2.6), but requires a summation of the contribution of many spikes in a sequence which may be difficult to implement in hardware. For the circuits of this thesis, it was decided to depress a synapse at the timing of every presynaptic spike by an amount which depends only on the time past since the last postsynaptic spike. Similarly, potentiation happens at each postsynaptic spike timing, and the magnitude of the weight change depends only on the time elapsed since the last presynaptic spike. This near-neighbour spike correlation approach allows for simpler learning circuitry because only the last time the presynaptic and postsynaptic neurons fired needs to be remembered, rather than a longer sequence of events.

4.2.4 Bimodal weight distributions

The main difficulty faced by the implementation of neural networks in analogue hardware is the long-term storage of analogue weights which emerge from learning. The strong competition that weight-independent TAH learning imposes between synapses creates bimodal weight distributions (see diagram in Figure 2.4). Weights can thus be considered binary, with either maximum or minimum value. This thesis suggests that the bimodality in the weight distribution makes these learning rules very good candidates to analogue VLSI implementation. Weights are analogue at the start and during learning. Therefore, the implementation can benefit from
the computation carried out by analogue circuits. At the end of the learning process, when weights have been drawn to either the maximum or the minimum saturation limit, a simple mechanism for the storage of binary weights could be triggered.

This thesis proposes that it may be possible to retain the binary weights on-chip without dedicated bistable circuitry. The learning process generated by weight-independent TAH learning rules is highly unstable so that, even when no correlations are present at the input, bimodal weight distributions emerge due to random correlations between the inputs. Hence, a specific weight distribution created by a temporal correlation pattern may be maintained after the removal of the correlation pattern. Weights would be kept at their learned binary value by the positive feedback of the learning rule. Weight retention in this manner is only possible if some random uncorrelated spike trains continue to stimulate the synapses after the learning phase.

4.3 The challenges

The hardware implementation of temporally asymmetric Hebbian learning will have to overcome several technological and design obstacles. Some of these technological challenges are common to most analogue neural hardware with on-chip learning, while other difficulties are specific to pulse-based neural chips.

4.3.1 Analogue weights and leakage

The lack of simple and robust long-term storage technology for analogue weights is probably the biggest barrier faced by on-chip learning. In section 4.2.4, it was suggested that the instability of weight-independent TAH learning may be a means of circumventing the need for long-term analogue weight storage once the binary weights have emerged. However, if capacitive storage for the weights is used, we still need to make sure that the leakage on the weight capacitors is small enough to avoid disrupting the learning process. Ultimately, the lowest speed of learning possible and the weakest spike input correlations the chip will be able to learn are partly determined by the weight-capacitor leakage. To mitigate the effect of charge leakage, the design should aim to have a voltage weight range as close as possible to the limits set by the ground and power rails (see section 4.5.5 and 4.5.9 for more details). Unfortunately, some of the circuits with transistors biased in weak inversion often used in neuromorphic synapses have a limited voltage range of operation that makes them sensitive to charge leakage in weight
4.3.2 Switching effects

Analogue circuits should ideally operate in a noise-free environment, which switching activity created by spikes does not provide. Mixed-signal electronic engineers are well aware of the disruptive effects of switching activity created by digital systems. Commonly, switching elements are kept as far away as possible from the analogue sections of the chip. Unfortunately, this separation between analogue and "switched" circuits does not exist in aVLSI implementations of networks of spiking neurons. In addition to noise injected through the substrate and line-to-line capacitive coupling, when analogue circuits are switched "on" or "off", they need some time to adapt to the new operating conditions.

A well-known effect of switching analogue circuits is the clock-feedthrough effect. That is, the large voltage transition at the gate of a transistor is capacitively coupled to its high-impedance output terminals. The interference introduced on high-impedance nodes when switching analogue circuits with pulses causes deterministic voltage errors. Thus, the effects of analogue-circuit switching are very different from the effects of random noise coupling. The "on/off" switching of analogue circuits can have destructive effects on the learning process. In contrast, analogue neural hardware may be appropriate when other types of interference exist. For instance, we could envisage neural learning schemes able to cope with an uncorrelated source of noise generated by a large section of digital logic.

4.3.3 Reduced visibility of the internal operation

Hardware implementations of spiking-neuron networks operate in real time at the pace set by the input data. Hence, they have been touted as a tool to investigate learning algorithms embedded in systems interacting in real time with the environment. However, a drawback of hardware implementations is that without additional circuitry the internal operation of the circuits is not visible to the experimenter. Thus, it is sometimes difficult to see the precise effect that design and algorithmic parameters have on the operation of the learning circuits fabricated.
4.3.4 Low number of synapses, unstable learning process and circuit mismatch

Usually, hardware spiking neurons have a small numbers of synapses — 6 learning synapses in the neurons presented later in this chapter; far fewer than the hundreds or thousands found in biological neurons. In neurons that operate as coincidence detectors, the probability that a single spike will increase the membrane voltage above the firing threshold is higher for neurons with a small number of synapses. Therefore, with a small number of synapses, random correlations between spikes will be more likely to generate a postsynaptic spike. Unfortunately, the unstable learning process created by weight-independent STDP will amplify these postsynaptic "errors" due to the small number of synapses. To make matters worse, with few synapses we lose the averaging effect that a large number of synapses would have over the fabrication defects which cause mismatches between synapses. See further discussion on this issue in section 4.4.4.

4.4 Design guidelines

Before turning to the description of the circuits, some design guidelines and general implementation decisions will be presented. Design decisions specific to each circuit (synapse, learning circuits, etc.) will be given later in section 4.5. All guidelines discussed were well defined before starting the design of the circuits presented in this chapter. Note, however, that most of them were not set in the early stages of this Ph.D. project.

4.4.1 Capacitive weight storage

The learning rules implemented will be adjusted in the learning experiments so that the final weights are close to either the maximum or minimum weight limits. Therefore, long-term storage of analogue values is not required once the weights have been learnt. Given the flexibility that capacitive weight storage adds to the design, it was decided to use capacitors to store the analogue weights during learning. We want to investigate whether, in a hardware implementation of STDP learning, a specific set of binary weights can be maintained by random correlations that occur in the input spike trains once the data (temporal correlations) are removed from the input spike trains. Hence, no additional circuits to refresh the weights or to enforce long-term bistability have been used.
4.4.2 Weak inversion vs. strong inversion

In many neuromorphic systems, MOS transistors are operated in weak-inversion (or subthreshold) mode. In this mode of operation nA-level currents are typical, and the drain current in saturation is given by the following expression

$$I_D = I_{d0} \left( \frac{W}{L} \right) e^{\left( \frac{V_{gs}}{n(V_t/q)} \right)}$$  \hspace{1cm} (4.1)

where W and L are the width and length of the transistor channel, $V_{gs}$ is the gate to source voltage, and $I_{d0}$ and n are process-dependent parameters. Unfortunately, the subthreshold region of operation is limited to a range of $V_{gs}$ values of a few hundreds of mV, since equation 4.1 is only valid for $V_{gs}$ values below the threshold voltage ($V_t$) of the transistors. Moreover, for $V_{gs}$ values well below the threshold, drain currents become often only one or two orders of magnitude higher than the parasitic junction currents. Another drawback of the subthreshold mode of operation is that mismatches in the $V_t$ of different transistors create large differences in their output currents. Current differences of up to 40% or 50% between transistors are not uncommon.

In strong inversion, μA-level currents are typical and, for transistors in saturation, the drain current has a quadratic relationship with $V_{gs}$ given by

$$I_D = \frac{\mu C_{ox}}{2} \frac{W}{L} (V_{gs} - V_t)^2$$  \hspace{1cm} (4.2)

where we have ignored the channel-length modulation effect. In strong inversion, output currents are far less sensitive to $V_t$ mismatches than in weak inversion. Furthermore, the $V_{gs}$ range is large since equation 4.2 is valid for $V_{gs} > V_t$. It was decided that our design would not be limited to using transistors only in one operation regime. The design will use transistors operated in weak inversion when generating long time constants and strong-inversion transistors when the voltage range of operation should be large.

4.4.3 Time-scale

Neuromorphic systems have been employed mainly to process the same type of signals that animals use to sense their environment (i.e., vision, sound, etc.). We envision that temporally
asymmetric Hebbian learning circuits could be used in high-level processing stages for neuromorphic systems. To this end, the circuits designed should be able to work at the same time scales as biological neural systems. Hence, the two tails of the STDP learning window implemented should extend over several $\text{msec}$ as has been found in biological systems. It is also important that the learning window is easy to tune. The aim is to have control over both peaks of maximum weight change and the width of the learning window. Furthermore, the parameters for potentiation and depression should be independent of one another. While the work presented in this thesis has focused mainly on the operation at $\text{msec}$ time scales, temporally asymmetric Hebbian learning could be used in applications with fast temporal correlations in the input data. Thus, the design should avoid limiting the range of operation of the circuits unnecessarily.

4.4.4 Parameter tuning, mismatch, biasing and criteria for device selection

Standard analogue integrated-circuit design usually targets well-understood problems and applications. Chips are not used as an experimentation tool to investigate an algorithm. The aim is to design circuits as robust as possible which, for reasons of cost, should require little post-manufacture trimming and tuning. In contrast, many neuromorphic aVLSI investigations use integrated circuits also as a vehicle to explore a problem. Some experiments carried out with the micro-chips are often devised as extensions of the experiments initially planned. In our case, the design of the TAH learning neuron should permit the tunability of most parameters of the neuron, so that on-chip learning can be studied with learning rules with different characteristics.

To provide tunability, all $nA$-level current sources and sinks will be implemented with transistors operated in weak inversion driven directly by voltage biases generated off chip. This current generation method allows for off-chip tunability but may render circuits sensitive to transistor mismatch. A current-distribution network for $nA$ currents is an impractical option, more so if we consider the spiking activity that takes place in the chip. Bias currents of the order of $\mu A$ will be generated with transistors in strong inversion. When a bias current does not affect matching between synapses of the same neuron, transistors will be biased with gate voltages generated off-chip. If a $\mu A$-level current can cause mismatches between synapses, that current should be a mirrored copy of a current generated outside the chip. This last tuning method is robust against device parameter variations in the chip. Chip tuning is further discussed in
Mapping temporally asymmetric Hebbian learning into aVLSI

section 5.2.2.

Standard analogue integrated-circuit design and neuromorphic aVLSI design also differ in the usual approach to device sizing. In standard analogue design, accurate selection of biasing and device sizes is critical for the correct operation of the circuits. For instance, analogue engineers designing amplifiers care about the value of parameters such as the transconductance \( g_m \) of transistors since it affects directly the final performance of the circuit. Also, in an operational amplifier the exact value of the compensation capacitor will determine the frequency response.

Most spiking neuron circuits consists of switched current sources and current sinks which are turned on or off for the duration of a pulse. These switched currents are used to charge capacitors which are then discharged between pulses. Thus, the sizing of devices is rarely critical for this type of designs. The common trade-offs of standard analogue circuit design between gain, stability and bandwidth are not an issue for spiking neurons. Another reason why devices sizes are non-critical in many neuromorphic chips is the post-manufacture tunability scheme discussed above. Since neurons will be adjusted in the lab for every experiment, the precise absolute value of devices becomes unimportant. The approach to sizing taken in this project consists in choosing dimensions which will allow tuning the neuron parameters by the amount desired. Thus, for nominal process parameters the dimension chosen should position the neuron parameters in the middle of the band that will be explored in the learning experiments. Device sizing will also be used to bias transistors either in strong or weak inversion. In reference to devices sizing, it is also important to note that most neuromorphic analogue circuits operate at low frequencies. Thus, transistors with non-minimal dimensions can be used.

It is clear from the discussion given above on transistor sizing and tuning that the design will not be process tolerant. However, this is not an important issue for this design because the experimenter in the lab will tune the chips to get the neuron parameters desired. On the contrary, the design should pay more attention to the effects of device mismatch (differences in the characteristics of two equally sized devices in the chip), since those cannot be eliminated by tuning.

TAH learning is most sensitive to mismatches between synapses. It is easy to see that differences in the parameters of different synapses of the same neuron can affect learning severely. If synapse A is stronger than synapse B, then A will require less temporal correlation with other synapses to be able to drive the membrane voltage above the firing threshold. Therefore,
synapse A has a higher probability to be potentiated than synapse B. In other words, differences in the initial strength of synapses make the neuron less sensitive to temporal correlations. Similarly, a synapse with a learning window with a high potentiation peak will require less temporal correlation with other synapses to reach maximum strength than a synapse with a lower peak.

In hardware implementations, the sensitivity of TAH learning to synaptic mismatch is likely to be higher than in biological neurons. The reason for this increase in sensitivity is the difference in the number of synapses per neuron between hardware neurons and biological neurons. In biological neurons the number of synapses is high. Thus, mismatches can be cancelled out on average if a neuron has several synapses stimulated by the same presynaptic neuron (or by a pool of neurons which generate similar spike-timing patterns). Unfortunately, hardware neurons usually have low number of synapses and this cancelling effect is non-existent. For all this, it is important to have matched synapses for each neuron since the learning process will not be able to compensate for the mismatches. Conversely, mismatches between circuits used in the somas (membrane leakage, refractory period, etc.) of neurons in a network may be filtered out by the learning process since weight-independent STDP is known to be fairly insensitive to firing-rate differences [12].

In order to match the learning windows of different synapses, the peaks of potentiation and depression will be set using current mirrors operated in strong inversion. In general, for circuits located at the synapse, current-biased transistors (i.e., when $V_{gs}$ is set by a current applied to the source) should be operated in weak inversion, whereas voltage-biased transistor in strong inversion. An exception to this rule will be the transistors that set the long decays (several msec) for the tails of the learning window.

In the same manner as the circuits described in section 3.6, current will be injected into weight capacitors and soma capacitors for the duration of a spike. Therefore, in order to have matched synapses, spikes should have a well controlled duration. The length of a pulse can be well controlled only if it is generated using a current source with transistors operated in strong inversion. This implies that $\mu A$-level currents will be used for the generation of the spikes. Consequently, pulse durations should be of the order of $1\mu s$ so that capacitors can be small.
4.4.5 Neuron features

Hardware spiking neurons vary in the level of detail in which they mimic features found in biological neurons. The more parameters the neuron has, the harder it is to control the effect that each parameter has on the operation of the neuron. It was decided the design would focus on the temporally asymmetric Hebbian modification of the synaptic strength. Thus, the neuron presented below does not incorporate spike-rate adaptation or short-term dynamic synapses. It has been equipped, however, with a tunable refractory period.

4.4.6 Weight-independent vs. weight-dependent learning

Having weight dependence in the learning rule has a strong impact on the computational properties of temporally asymmetric Hebbian learning rules. It was decided that the neuron designed should have learning circuits which could implement either weight-independent or weight-dependent learning rules, so that both types of learning rules could be investigated. As discussed in section 4.3.4, learning with a small number of mismatched synapse, combined with the unstable learning process created by the strong competition imposed by weight-independent learning, may hinder the detection of subtle temporal correlations between input spike trains. This thesis suggests that introducing a moderate level of weight dependence in the learning rule may stabilise the learning process, so that it becomes easier to detect weak correlations, while allowing at the same time the development of the binary weight distributions we are interested in.

4.5 Hardware spiking neuron with STDP synapses

4.5.1 Introduction

The aim of this thesis is to investigate the effect that temporal correlations between inputs have on the learning process. TAH learning forces synapses to compete for the control of the postsynaptic neuron. A synapse can only win (i.e., reach full strength) by cooperating with a group of synapses which receive temporally correlated spike trains. Three is the minimum number of synapses required to show competition and cooperation. However, the conclusions that can be drawn from temporal-correlation learning experiments with two correlated synapses competing against a single uncorrelated synapse would be quite limited. Therefore, it is rea-
sonable to say that four synapses per neuron is the minimum required to carry out interesting
temporal correlation experiments where two even groups of synapses compete for the control
of the postsynaptic potential. Emulating the rich spike-train processing that takes place in a sin-
gle biological neuron would require hundreds, or even thousands of synapses per neuron. The
implemion of such a system is beyond the scope of this Ph.D. In this project, it was decided
to have six synapses per neuron. In addition to the experiments possible with four synapses, the
hardware neurons proposed can be used in experiments consisting of two correlated synapse
competing against four uncorrelated synapses.

The circuits for the TAH learning neuron proposed were designed for a standard 0.6µm CMOS
process from Austria Micro Systems [48]. This is a mature process which had already been
used in our research group in the past with good results. It has a complete and straightforward
design kit. The circuits for the hardware neuron do not require a process optimised for either
high voltages or high-frequency operation. Therefore, this mature standard CMOS process is
appropriate for this design.

4.5.2 Neuron architecture

A diagram of the architecture of the neuron designed is shown in Figure 4.1. The synaptic
array is made up of six learning synapses, a fixed inhibitory synapse and a fixed excitatory
synapse. The soma contains the integrate-and-fire mechanism, a circuit to generate a sequence
of pulses which drive the learning circuits, and a circuit which generates a set of six non-linear
time-decaying currents which define the shape of the depression region.

Learning synapses have their strength modified according to a temporally asymmetric Hebb-
bian learning rule. As seen on the enlarged view given in Figure 4.1, learning synapses have
three main components. The causal correlation circuit is responsible for generating the time-
decaying current which defines the long-term potentiation (LTP) side of the learning window.
The function of the weight change circuit is to control the injection of the depression ($i_{LTD}$)
and potentiation ($i_{LTP}$) currents into the weight capacitor. The weight, represented by a weight
e voltage $V_w$, determines the amount of current injected into the soma when a pre pulse activates
the synapse circuit. Weights of learning synapses can only take positive values.

Two non-learning synapses have been added to the synaptic array to give more flexibility to
the experiments carried out with the chip. The inhibitory non-learning synapse allows the
Mapping temporally asymmetric Hebbian learning into a VLSI implementation of networks with global inhibition. The excitatory non-learning synapse may be used to model the effect of a strong input or some background activity. Furthermore, to test the neurons (see chapter 5) it is convenient to have a non-learning excitatory synapse to force the neuron to fire at a precise moment in time.

The core element of the soma is the integrate-and-fire (IF) circuit. The current generated by the eight input synapses ($I_{\text{synapses}}$) is integrated by the soma capacitor. When the voltage across the capacitor reaches the firing threshold, the IF circuit generates a voltage pulse at the output and resets itself. For each rising edge at the output of the IF circuit, a sequence of three consecutive pulses is generated by the pulse sequencer circuit. The longer pulse in the sequence is used in

Figure 4.1: Diagram of the neuron architecture (see text for details).
the a-causal correlation circuit to create a set of non-linear time-decaying currents $I_{LTD}[1 : 6]$. One $I_{LTD}$ current from the set is sent to the weight change circuit of each input synapse. The postsynaptic event transmitted to following neurons in the processing chain is signalled with the same long pulse (postLong) followed by a shorter one (post). These two pulses are labelled as preLong and pre when they reach the presynaptic terminal of the receiving neuron.

Figure 4.2 illustrates the transmission of pulses between the neurons proposed. The diagram shows how synapses 1 and 4 of neuron N5 in layer n+1 are stimulated by neurons N1 and N4 located in the previous layer. In the same manner, the remaining synapses of N5 are stimulated by other neurons in layer n (not shown in the figure). The communication of an event (spike) involves sending two non-overlapping pulses. These two pulses called post and postLong at the output of the sending neuron (N1 and N4 in Figure 4.2) are called pre and preLong, respectively, at the synapse of the receiving neuron (N5).

4.5.3 Integrate-and-fire circuit

Many circuits to implement in analogue VLSI the integrate-and-fire behaviour of biological neurons have already been proposed by others (see review in chapter 3). The aim in this project was to design an IF neuron with a leaky integrator and an absolute refractory period of controllable duration. As in other hardware spiking neurons, the IF neuron designed merges features
Figure 4.3: Integrate-and-fire neuron circuit. (A) Symbol. (B) Schematic.
from Mead's [5] and van Schaik's [66] neurons (cf. chapter 3). The schematic of the IF neuron circuit is shown in Figure 4.3-B. The symbol that was used in the architecture diagram is repeated for clarity in Figure 4.3-A. Next to some nodes of the circuit we have sketched the typical voltage waveform of the node around the timing of the spike. A table containing the devices sizes and current-bias ranges for all circuits presented in this chapter has been included in Appendix A.

Pulses of current arriving from the synaptic array are integrated by the soma capacitor $C_{soma}$. The charge accumulated in the soma capacitor leaks slowly through transistor N1 (other mechanisms of current leakage are available, and will be described later). If several pulses of synaptic current occur in a short time window, such that collectively they can overcome the leakage rate imposed by $V_{leak}$, the soma voltage ($V_{soma}$) increases beyond the threshold voltage ($V_{th}$) of the comparator and its output switches from low to high. The sudden voltage change in the comparator output is fed back to $V_{soma}$ through the feedback capacitor $C_{fb}$ to create the upswing of the "action potential". This method of spike generation has been chosen because it is fast and robust against interferences from other spikes generated nearby.

The falling transition of the "action potential" is controlled by the refractory period circuits. When the threshold comparator switches to a high-level output, $C_{ref}$ is reset to the power rail ($V_{dd}$) through P1. This causes the soma capacitor to be discharged to 0V (GND) through N3-N4 immediately, thus creating the falling transition of the "action potential". After the spike, the voltage across $C_{ref}$ increases slowly at the rate imposed by the external bias voltage $V_{bref}$. Transistor N3 remains fully switched on until $V_{ref}$ is lower than the logic threshold of the inverter. Therefore, $V_{soma}$ is maintained to 0V for the duration of the refractory period controlled with $V_{bref}$. To enforce the refractory period, the saturation current of transistor N4 should be set larger than the total current that can be generated by the synaptic array.

Refractoriness is the only additional feature that has been added to the spiking behaviour of the neuron. In temporally asymmetric Hebbian learning, the refractory period helps break the temporal axis into separate segments, so that only one presynaptic spike per input is "predicting" each postsynaptic spike. Furthermore, if the refractory period is longer than the timing window for the induction of weight modification, the near-neighbour presynaptic-postsynaptic spike interaction scheme for STDP learning that was chosen for the proposed hardware neurons is equivalent to an all-to-all spike interaction.
A reset signal is always convenient during characterisation and functional testing. As seen in Figure 4.3-B, the IF neuron can be reset to a known initial state. \( V_{\text{soma}} \) and \( V_{\text{ref}} \) are reset to 0V when the reset signal is set high.

The output spike of the IF neuron is not sent directly to following neurons in the processing chain. Instead, as will be shown in section 4.5.4., the spike signal is transformed by the pulse-sequencer circuit into a sequence of spikes that will drive the learning circuits of the current neuron and the synapses of the next receiving neuron.

Figure 4.4: (A) Symbol of the comparator of the integrate-and-fire neuron. (B) Schematic of the comparator. (C) Symbol of the alternative sources of soma leakage. (D) Circuit detail of the alternative soma leakage.

Figure 4.4 shows in detail some building blocks used in the IF neuron circuit. The comparator used for the firing threshold is shown in Figure 4.4-B (symbol shown in Figure 4.4-A). It is
a simple two stage comparator followed by a push-pull inverter that gives the required output polarity.

Transistor N1 in Figure 4.3-B is commonly biased in weak inversion to create a slow decay on the soma voltage. Hence, the amount of input spike synchrony required by each neuron to elevate $V_{soma}$ above the firing threshold could vary widely if transistor mismatch is poor. Thus, it was decided to include two additional leakage mechanisms in case learning was hindered by leakage mismatch (see Figure 4.4-D). Note, however, that in all test results given in chapters 5 and 6, these two additional leakage mechanisms were never used. The first alternative mechanism for soma leakage is provided with a switched capacitor implemented with switches N4 and N5, and capacitor $C_{sw}$. This switched-capacitor resistor allows the implementation of a stepped approximation to a true RC circuit for the soma as found in biological neurons. Unfortunately, the circuit requires two external clocks ($clk_{1soma}$ and $clk_{2soma}$) which may generate too much interference to the nearby circuitry. The other alternative source of soma leakage available consists of a locally-scaled down version of a μA-level current ($I_{dec.soma}$). Leakage currents will be better matched since we are dividing $I_{dec.soma}$ with highly asymmetric current mirrors (P1-P2 and N2-N3) rather than driving the gate of distant transistors with a fixed voltage, such as is done with $V_{leak}$ on the gate of N1 in Figure 4.3. These two alternative leakage mechanisms can be disabled and are only to be used if mismatches in the N1 transistors of the simpler leakage method prevent learning spike synchrony.

4.5.4 Pulse sequencer

The circuits that implement the temporally asymmetric Hebbian learning window require that presynaptic events are signalled with two consecutive pulses. The first pulse should be of the order of a few μsec, and precision in its duration is not required. The second pulse should be shorter, with a duration of about 1μsec. Differences in the duration of the short pulse between different neurons may impact the learning ability of the network. As for the postsynaptic event, the learning circuits require a short pulse followed by the longer one. A justification of these requirements is given below with the description of the a-causal correlation and causal correlation circuits, and in section 4.5.11. Figure 4.5 shows the circuit that generates the sequences of pulses required for both the presynaptic and the postsynaptic events.

With every rising edge of the output spike signal of the IF neuron (see Figure 4.3-B), three consecutive pulses are generated by the pulse-sequencer circuit. The first short pulse (post_bp)
Figure 4.5: (A) Symbol for the pulse-sequencer circuit. (B) Schematic of the pulse-sequencer circuit.

is "back-propagated" to the input synapses of the neuron and is involved in the potentiation of the input weights (see section 4.5.9). Signal Long is used in the a-causal correlation circuit located also in the "soma" of the neuron, where it is labelled as post_Long. This same Long pulse is used as the long pulse required by the causal correlation circuits of synapses that receive spikes generated by the current neuron (cf. section 4.5.8). At the receiving presynaptic terminals, the Long pulse will be labelled as pre_Long. Finally, the last short pulse (spike_out) is also sent to next presynaptic terminals, where it will be simply called pre.

The pulse-sequencer circuits contains three pulse-generator circuits (PG) connected as a chain. The pulse-generator circuit schematic is given in Figure 4.6. It generates a pulse of controlled duration with every rising edge of the input (in) pulse. Through the reset_b signal the Q output of the D-type flip-flop is initially set high. When a rising edge in the input is detected, the Q output is switched low and capacitor C_{pg} is charged through P1. When the charging current brings the voltage across C_{pg} above the threshold of the comparator (V_{thpg}), the same capacitive-feedback effect seen in the integrate-and-fire neuron creates a short pulse at the S_b input of the flip-flop. As a result, Q is set high again. Thus, the charging current flowing through P1 is disconnected and C_{pg} is discharged through N1. The duration of the out pulse equals the time it takes for the charging current to bring the voltage across C_{pg} to V_{thpg}. The only difference between the PG circuits that create long or short pulses is the strength of the
current used to charge capacitor $C_{pg}$.

Two sources of current to charge $C_{pg}$ are provided. If the experiments show that learning is sensitive to mismatches in transistor P4 of the pulse-generator circuits, a current mirror (P2-P3) can be used instead to reduce the mismatch between neurons. However, it is unlikely that mismatches in the charging currents will become an issue since the generation of short pulses requires P4 to be biased in strong inversion.

### 4.5.5 Learning synapse

No refresh circuitry for the weights has been used. Therefore, the synapse circuit requires a very large weight-voltage range to minimise the impact that charge leakage in weight capacitors has over the learning process. Having a perfectly linear voltage-to-current relationship is likely not to be important for our learning experiments. However, a simple transconductor based on a single transistor whose gate is driven by the weight voltage would make the control of the learning process difficult. Instead of the exponential or quadratic transconductance provided by a single transistor synapse, the aim was to design a synapse with some degree of linearity.
without a significant increase in the complexity of the circuit.

Circuits with transistors biased in weak inversion are often used for their low-power consumption. Unfortunately, the usable range of gate-source voltage for weak-inversion transistors is limited to a few hundreds of mV. Hence, it was decided that circuits with transistors operated in strong inversion would also be considered for the synapse. Using transistors in strong inversion creates \( \mu A \)-level currents. This has implications beyond power consumption; if the synapses generate \( \mu A \) currents, then the duration of the presynaptic spikes will have to be short (around a \( \mu \)sec or less).

It was decided that the synapse would not include any short-term dynamic plasticity. Most dynamic synapses proposed in the literature are based on the current-mirror integrator, whose drawbacks have been discussed in section 3.6.3. These type of dynamic synapses are usually designed as fixed-weight synapses.

![Figure 4.7: Transconductor using transistors in ohmic region.](image)

Transconductors —circuits that have a voltage-in/current-out transfer function— are important building blocks in mainstream analogue VLSI design. For instance, they are used to build continuous-time filters known as Gm-C filters. Many different transconductor circuits have been proposed to target different specifications and applications. The linearity of the transfer function is one of the most important specifications of a transconductor, and has received much attention from researchers working in this area [85].

A MOS transistor can be biased in the ohmic (or linear) region to obtain a drain current which
Mapping temporally asymmetric Hebbian learning into a VLSI

does linearly with the gate-to-source voltage:

\[ I_d = \mu C_{ox} \frac{W}{L} \left[ (V_{gs} - V_t)V_{ds} - \frac{V_{ds}^2}{2} \right] \] (4.3)

Equation 4.3 shows that the linear transfer function we are interested in is modulated by the voltage applied across the channel. Therefore, to use the linear region of operation to build a transconductor, the drain terminal of the transistor needs to be kept at a constant voltage. In section 3.6.2, a two-quadrant transconductor for a learning synapse (i.e. with positive and negative weight values) which exploits the linear region of a MOS transistor was presented. That circuit is essentially a two-transistor transconductor with an additional transistor switch to control the flow of output current (see Figure 3.3). It requires a good operational amplifier to maintain the drain terminals to a constant voltage. The same Op.Amp. is used to perform a current-to-voltage operation on the synaptic current. Therefore, to use the IF circuit of section 4.5.3, the voltage representing the activity at the synapses would have to be transformed again into a current signal.

Before presenting the circuit for the learning synapse, we will look briefly at the linear transconductor based on transistors in ohmic region shown in Figure 4.7 [60]. Focusing our attention first on the highlighted part of the circuit of Figure 4.7, we see that the bias voltage \( V_c \) applied to the positive input of the amplifier is effectively copied — by means of the virtual short-circuit of the Op.Amp. — to the drain of N1 \( (V_{d1}) \). Thus, the cascode effect of transistor N2 is "enhanced" using a very large amplification. Consequently, \( V_{d1} \) is kept virtually unchanged for a wide range of currents generated by the input voltage \( V_{in1} \) applied to the gate of N1. To ensure that the current flowing through N1 is linear with respect to \( V_{in1} \), \( V_c \) must be such that \( \frac{1}{\mu} < \frac{V_1 - V_{t1}}{V_{in1}} \). Commonly, the offset term \( V_{ds}^2 \) of equation 4.3 is cancelled using the differential configuration shown in the figure.

The learning synapse used in our neurons is based on the core idea behind the transconductor of Figure 4.7. However, the operational amplifier has been removed altogether. Due to area constraints, adding an operational amplifier to each synapse is not an option if we aim to eventually have tens of synapses per neuron. Furthermore, the high linearity provided by the circuit of Figure 4.7 is not required in our system.

The resulting synapse circuit is shown in Figure 4.8. Transistor P1 is the linear transconductor element and should be biased in ohmic region and in strong inversion. It has a small aspect ratio.
Mapping temporally asymmetric Hebbian learning into a VLSI

Figure 4.8: Schematic of the synapse circuit.

$W/L (W/L = 0.1)$. Its gate is driven by the weight voltage $V_w$. Since the linear transistor P1 has its source connected to the power rail ($V_{dd}$), the lower $V_w$, the larger the drain current. In other words, large weight values correspond to low weight voltages. The key role of transistor P2 is to ensure that the drain of P1 varies only by a small amount when the current increases. P2 also makes the synaptic current $I_{syn}$ independent of the voltage across the soma capacitor where it is being injected (cf. section 4.5.3). In contrast to the transconductor we saw in Figure 4.7, this circuit does not have an amplifier to boost the gain of the cascode transistor. Therefore, all voltage-to-current gain between the $V_{gs}$ voltage of P2 and its drain current must be provided by the P2 transistor itself. Transistor P2 is sized to have a large aspect ratio $W/L (W/L = 10)$ which gives the transistor a small transimpedance. Furthermore, if transistor P2 has a large aspect ratio, it will operate in weak or moderate inversion for the low-level currents generated by P1. In consequence, its transimpedance will be decreased even further given the exponential relationship between its $V_{gs}$ voltage and its drain current. The combination of a small transconductance for P1 and a small transimpedance for P2 can maintain the source of P1 at a fairly constant voltage. In other words, since P2 has large $W/L$ it needs a small voltage change at its source to conduct the current which P1 is trying to source. Through P3, the
resulting "linearised" transconductor controlled by $V_w$ will inject into the soma capacitor of the IF a pulse of current $I_{syn}$ every time a presynaptic spike reaches the synapse.

It is easy to see that the synapse circuit of Figure 4.8 has limited linearity. When the weight voltage decreases (i.e. the weight is incremented), the current $P_1$ attempts to source increases. To accommodate the extra current that $P_1$ is trying to generate, the source of $P_2$ (drain of $P_1$) increases slightly, thus reducing the transconductance of $P_1$.

The same 2-transistor transconductance described above will be used again to introduce weight dependence in the learning window (cf. section 4.5.8).

To increase inter-synapse matching and ease the tunability of the synaptic strength, the biasing circuit made up of $P_4$-$P_5$ has been used. The gate voltage of $P_2$, set by the biasing circuit, will force a synaptic current equal to $I_{w,\text{set}}$ when the weight voltage $V_w$ equals $V_{w,\text{set}}$. In most of the learning experiments we will set $V_{w,\text{set}}$ to 0V (i.e., we will fix the maximum strength allowed to the synapse).

### 4.5.6 Non-learning synapses

Non-learning synapses have a fixed long-term value which is not modified by the network activity. They are much easier to design than learning synapses which require a wide input range for the voltages and currents controlling the synaptic strength. The neuron that is being described in this chapter includes two fixed non-learning synapses: a synapse with fixed positive weight (excitatory) and a synapse with fixed negative weight (inhibitory).

#### 4.5.6.1 Excitatory synapse

Similarly to the learning synapse, the fixed excitatory synapse does not have any short-term dynamics; current is only injected into the soma capacitor for the duration of the input spike. The circuit schematic of this synapse is shown in Figure 4.9-B. When a spike reaches the synapse, a pulse of current set by $V_{exc}$ is injected into the soma. The $V_{exc}$ voltage is generated off-chip and is common to all neurons. Important aspects of this fixed synapse configuration have already been discussed in section 3.6.1. In some characterisation tests discussed in chapter 5, this fixed excitatory synapse is set very strong so that it drives the soma voltage beyond the firing threshold every time it is activated. With a $V_{exc}$ voltage close to the power rail the
Figure 4.9: Non-learning synapses. (A) Symbol of the excitatory non-learning synapse. (B) Schematic of the excitatory non-learning synapse. (C) Symbol of the inhibitory non-learning synapse. (D) Schematic of the inhibitory non-learning synapse.

A synapse becomes weaker and it could be used to introduce noise in the learning process or to model background activity.

4.5.6.2 Inhibitory synapse

The circuit for the inhibitory synapse is shown in the schematic of Figure 4.9-D. When a presynaptic spike reaches the synapse, the gate of transistor N3 is charged to voltage $V_{peak\_inh}$ and N3 starts to draw current from the soma capacitor. This inhibitory current decays toward zero with a rate set by $V_{dec\_inh}$. Hence, if N2 is biased in weak inversion, the effect of the inhibitory synapse on the neuron's soma can last much longer than the duration of the input spike. The inhibitory synapse can be set very strong so that it discharges the soma capacitors completely with each activation (shunting inhibition).
4.5.7 A-causal correlation circuit

The circuit shown in Figure 4.10 creates six non-linear time-decaying currents which determine the shape of the learning window for the depression region — the side of the learning window corresponding to a-causal interactions between a presynaptic and a postsynaptic spike. The a-causal correlation circuit is located near the integrate-and-fire circuit at the "soma" of the neuron. The non-linear time-decaying currents are sent to the weight change circuits found at each input synapse. Weight depression will be induced when a presynaptic spike reaching a synapse samples the $i_{LTD}$ current (see section 4.5.9).

The a-causal correlation circuit operates as follows. When a postsynaptic event occurs, a postLong pulse switches on transistor N2, causing capacitor $C_{LTD}$ to be charged to the gate voltage forced by the bias current $I_{bLTD}$ on the diode connected transistor N1. The postLong pulse should be long enough (a few $\mu$s) to allow $C_{LTD}$ to be fully charged. When signal postLong is set low again, the charge in the capacitor leaks slowly through N3. Hence, the voltage across the capacitor is a measure of the time elapsed since the last postsynaptic spike. This linearly decaying voltage across $C_{LTD}$ is applied to the gate of six equally sized transistors. Given the non-linear relationship between the output current and the gate-source voltage,
a set of non-linear time-decaying currents is generated.

If transistor N3 is biased in weak inversion with $V_{\text{decLTD}}$, the learning window tail for depression will be several msec wide. Since the presynaptic spikes that sample the $i_{\text{LTD}}$ currents last for about a $\mu\text{sec}$ or less (see description of the weight-change circuit in section 4.5.9), the typical peak of voltage applied to the gates of the six output transistors should be above the $V_t$ threshold. Thus, the $i_{\text{LTD}}$ current will decay in a quadratic manner with time. It must be underlined that the smooth-decaying learning window that is so characteristic of STDP is achieved in our implementation exploiting the non-linear transfer function of MOS transistors that many analogue design techniques fight against. Plots that characterise the shape of the depression side of the learning window implemented with this circuit are given in chapter 5.

As in the IF circuit, the same two additional mechanisms for capacitor leakage are provided. Here, we are not as interested in the stepped exponential decay that a switched capacitor leakage can provide. However, mismatches in the leakage currents when N3 transistors are biased in weak inversion may create different decays in the learning windows of different neurons. Notice that these extra leakage mechanisms were not used in any of the tests and experiments described in chapters 5 and 6.

### 4.5.8 Causal correlation circuit

The circuit in charge of shaping the potentiation part of the learning window is shown in Figure 4.11. If we disregard for a moment the devices involved in controlling the amount of weight dependence introduced in the learning rule, the circuit operates in a similar manner to the a-causal correlation circuit. Capacitor $C_{\text{LTP}}$ is charged by the long pulse of the presynaptic event (preLong). The rate of decay of the learning window is tuned with the external bias voltage $V_{\text{decLTP}}$ applied to the gate of N5. Similarly to the a-causal circuit, the peak of the learning window for potentiation is set by the bias current $I_{\text{bLTP}}$ injected into the diode connected transistor N3.

The causal correlation circuit proposed can generate weight-dependent potentiation with only four additional transistors (P1-P2-N1-N2). Transistors P1 and P2 implement the same linearised transconductor that was used for the synapse circuit (cf. 4.5.5). Hence, when the circuit is properly biased with the weight-dependence control voltage $V_{\text{bc}}$, an amount of current proportional to the weight of the synapse is injected into current mirror N1-N2. This weight-
Mapping temporally asymmetric Hebbian learning into a VLSI

---

**Figure 4.11:** (A) Symbol of the causal correlation circuit. (B) Schematic of the causal correlation circuit.

Dependent current is subtracted from the fixed current $I_{bLTP}$. The reduced current resulting from this subtraction creates a lower peak voltage on the gate of transistor N6. If we ignore the non-linearity of the transconductor implemented with P1-P2, the peak of the potentiation current is,

$$I_{\text{max}}^{\text{LTP}} = \frac{\beta_6}{\beta_3} \{I_{bLTP} - [\alpha(V_{dd} - V_w) + \gamma]\}$$

Equation 4.4 shows that the circuit of Figure 4.11 can introduce the desired stabilising effect on the STDP learning process: synapses with large weights (lower $V$) will undergo less potentiation than synapses with small weights ($V_w$ close to $V_{dd}$).

Weight-dependent learning can be disabled by setting $V_{bc}$ equal to the power supply voltage (5V).
4.5.9 Weight change circuit

The weight change circuit is shown in Figure 4.12-B. The typical waveform of all input signals involved in weight modification has been sketched next to the corresponding input port. Note that the waveforms are not drawn to scale; the decay of the input currents — $i_{\text{LTD}}$ and $i_{\text{LTP}}$ — can last several msec whereas the pre and post.bp pulses last for about 1μsec.

![Weight change circuit diagram](image)

**Figure 4.12:** Weight change circuit. (A) Symbol. (B) Schematic.

The weight of the synapse is represented by the voltage across the weight capacitor $C_w$. The strength of the synapse is inversely proportional to the weight voltage $V_w$. In other words, low $V_w$ voltages correspond to large weight values. The weight change circuit controls the flow of charge into the weight capacitor $C_w$. The weight is potentiated ($V_w$ decreased) when a postsynaptic spike post.bp reaches the synapse shortly after a preLong pulse has created a time-decaying current $i_{\text{LTP}}$ (see section 4.5.8). A post.bp pulse will be capable of inducing potentiation only if the postsynaptic event happens before $i_{\text{LTP}}$ has decayed completely. Similarly, the weight will be depressed ($V_w$ increased) if a presynaptic spike (pre) switches transistor
Within a few \textit{msec} of the $i_{LTD,x}$ current onset. If a presynaptic spike arrives after $i_{LTD,x}$ has decayed completely, the weight voltage remains almost unchanged. The time-decaying non-linear current $i_{LTD,x}$ is one of the six currents generated by the a-causal correlation circuit located at the soma of the neuron (see section 4.5.7).

The weight will suffer from charge leakage due to the reverse-biased junctions of the transistors connected to the weight capacitor (i.e., N3, P3 and N2). Unfortunately, it is difficult to predict the equilibrium point where the leakage currents due to the PMOS and the NMOS transistor will cancel out. The reverse-biased junction currents are not pass/fail parameters in the quality controls performed on the fabricated chips. Worse still, the reverse-biased current data provided by the foundry may not be accurate since it is not well characterised statistically. Therefore, both the leakage equilibrium point and the rate of weight degradation are unknown.

Through transistor N3, the weight can be reset to $W_{\text{init}}$. The voltage externally given to $W_{\text{init}}$ is common to all weights in the chip. Hence, it will not be possible to initialise the network with randomly distributed weights, as is often done in software simulations. The weight is reset only through an NMOS transistor. A PMOS switch is not required since we usually want to start the learning with strong synapses (low $V_w$). If the synapses are too weak initially, the synaptic array of the neuron will not be able to generate the postsynaptic activity needed to drive the learning process.

4.5.10 Current biasing

As was discussed in section 4.4.4, mismatches between synapses in a neuron may prevent learning. To obtain synapses with better matched peaks of potentiation and maximum strength, the bias blocks shown in Figure 4.13 have been included in every neuron to generate matched copies of currents $I_{bLTP}$ (see causal correlation circuit in Figure 4.11) and $I_{w.set,x}$ (see synapse circuit in Figure 4.8). The generation of the reference currents ($I_{bLTP.in}$ and $I_{w.set.in}$) for each neuron in a network is discussed in section 5.2.2.

4.5.11 Final remarks

The way in which the $i_{LTP}$ and $i_{LTD}$ currents are generated determines the form of spike interaction implemented by the temporally asymmetric Hebbian learning neurons presented. Capacitor $C_{LTP}$ ($C_{LTD}$) is recharged to its peak value at every presynaptic (postsynaptic) spike.
Figure 4.13: Current-biasing circuits of the neuron. (A) A current input $I_{bLTP_{in}}$ is used to generate matched copies of the $I_{bLTP_x}$ current used in the causal correlation circuit (see Figure 4.11). (B) A current input is used to generate matched copies of $I_{w_{set,x}}$ which are sent to the synapse circuits of Figure 4.8.

timing, and then it is discharged slowly. Therefore, $i_{LTP}$ and $i_{LTD}$ carry information only about the time past since the last presynaptic and postsynaptic spike, respectively. No information about the timing of the previous spikes is retained, since the voltages across $C_{LTD}$ and $C_{LTP}$ are reset to their peak value at every spike. As seen in the schematic of the weight-change circuit of Figure 4.12, the time-decaying current $i_{LTD}$ ($i_{LTP}$) generated by the last postsynaptic (presynaptic) spike will induce weight depression (potentiation) for every presynaptic (postsynaptic) event happening before the complete decay of the current. In other words, depression (potentiation) is driven by the delay between the last postsynaptic (presynaptic) spike and all subsequent presynaptic (postsynaptic) spikes. In practical terms, however, if a refractory period of similar duration to the effective width of each side of the learning widow is enforced, the spike interaction is limited to the presynaptic spike immediately following a postsynaptic spike, and vice versa.

The abrupt transition of the learning window is one of the most striking features of temporally asymmetric Hebbian learning. It is important that the potentiation part of the learning window includes all causal spike interactions. In particular, the synapse which receives the last presynaptic spike responsible for increasing the soma voltage above the firing threshold must be strongly potentiated. It was already discussed that the long pulses $postLong$ and $preLong$
are needed to allow the full charge of the $C_{LTD}$ and $C_{LTP}$ capacitors of the dynamic-current mirrors in the causal and a-causal correlation circuits. After the presentation of all circuits, we are in a position to explain the reasons for the sequence of non-overlapping pulses — post$_{bp}$, Long, post — used to communicate spike events. The comparator in the IF circuit responds fast when the soma voltage crosses the firing threshold. Therefore, the post$_{bp}$ pulse will often overlap with the pre spike that was responsible for that postsynaptic event. The postLong pulse is delayed with respect to the early post$_{bp}$ pulse to ensure that the last pre pulse does not inject into the weight capacitor the $i_{LTD}$ current created by the postsynaptic event. For a similar reason, the short pre pulse should come after the longer preLong pulse. In this way, we can guarantee that the post$_{pb}$ pulse samples the $i_{LTP}$ current when it has already reached its peak value.

4.6 Summary

This chapter started presenting the characteristics of temporally asymmetric Hebbian learning that make this new form of learning suitable to analogue VLSI implementation. As with other "Hebbian" learning rules, temporally asymmetric Hebbian learning requires only information local to the synapse. Also of interest is the fact that STDP learning rules do not constrain the design to the implementation of a precise mathematical function for the learning window. Moreover, learning is driven by spike-to-spike interactions, so the spike trains generated by the spiking neurons do not need to be averaged over long time windows. Finally, it is interesting for the hardware implementation that some forms of temporally asymmetric Hebbian learning generate bimodal weight distributions. Having final binary weights may be an opportunity to overcome the need for long-term storage of analogue weights.

All circuits required for the implementation of spiking neurons with STDP synapses have been presented. The learning circuits presented can create either weight-dependent or weight-independent learning rules. The weight-dependence mechanism only requires the addition of four transistors to the weight-independent circuit. The non-linear decay of the learning window has been achieved exploiting the non-linear current-to-voltage characteristic of a single MOS transistor.

This chapter has given a detailed description of the circuits for a spiking neuron with STDP synapses. Test results from a fabricated chip that characterise the operation of the neuron are
presented in the following chapter.
Chapter 5
Neuron and learning rule characterisation

5.1 Introduction

A chip with a small network of spiking neurons was fabricated to validate the circuits and methods presented in chapter 4. In this chapter, we present test results that characterise the spiking behaviour of the neurons, the synapses and the learning rules that can be implemented with the circuits proposed. The same chip has been used for the learning experiments presented in chapter 6.

The chapter starts presenting the architecture of the network. Next, the test setup used is described. The first test results given show that neurons and synapses in the chip are functional. That is, the synapses of the on-chip neurons can be stimulated from outside the chip, well-formed action potential are generated when the soma reaches the firing threshold, and the weights are modified in the right direction at the presynaptic and postsynaptic spike timings. The chapter follows with test results showing that the strength of the synapses is controlled by the weight voltage in a linear manner. The chapter ends with several graphs which characterise the tunability of the learning window implemented.

5.2 Network architecture and stimulation

The chip fabricated contains a small five-neuron feed-forward network and a separate standalone neuron. For the tests described in this and the following chapter, only the neurons in the network have been used. The architecture of the on-chip network is given in the diagram of Figure 5.1. It shows the external inputs used to stimulate the chip ($s[1 : 8]$ and $LongIn[1 : 5]$) and the output pulses ($postLong[1 : 5]$) used to monitor the spiking events generated by the on-chip neurons. The method used for biasing and tuning the neuron is discussed in section 5.2.2. The photograph of the microchip in Figure 5.2 shows the network as seen with a microscope.
Neuron and learning rule characterisation

Four neurons in the input layer (N1 to N4) can be addressed from outside the chip. As we saw earlier in chapter 4, each neuron has six STDP learning synapses, one inhibitory non-learning synapse, and one excitatory non-learning synapse. A fifth neuron (N5) is located in the second layer of the network. Four of its learning synapses receive the spike trains generated by the first-layer neurons. The two remaining learning synapses and the non-learning synapses of N5 receive externally generated spike trains.

In chapter 4 we saw that presynaptic events are signalled with two pulses, $pre_{Long}$ followed by $pre$. Synapses of neurons in the first layer and the externally addressable synapses of neuron N5 in the second layer have their presynaptic $pre$ pulses generated internally by the neuron access blocks seen in Figure 5.1. As shown in Figure 5.3, a $pre$ pulse is triggered by the falling edge of the $LongIn[x]$ pulse which is supplied externally. The circuit of the neuron access block consists simply of a pulse-generator circuit, as used in the pulse-sequencer circuit presented in section 4.5.4, and an array of AND gates to select the synapses that will receive the external $LongIn[x]$ pulse and the internally generated $pre$ pulse. The selection of synapses that receive
the presynaptic pulses is controlled with the external 8-bit signal $s[1:8]$.

Besides the chip stimulation, we need some way to monitor the activity of the network and the synaptic strength changes occurring inside the chip during learning. The postsynaptic event timings can be monitored through the $\text{postLong}$ signals of all five neurons, which are the only internally generated pulses available outside the chip. To analyse the learning process, it is useful to see the evolution of the weights of the network. To this end, six voltage buffers have been included with each neuron to monitor the voltages across the weight capacitors of the learning synapses. In addition, five voltage buffers have been used to monitor the voltage across the soma capacitor of each neuron. Seven analogue output lines can be multiplexed to observe the weights and the soma of any of the on-chip neurons. The voltage buffers use a PMOS-input differential pair. Hence, the input range saturates for weight voltages approaching the power rail. However, this will not limit the observability of the chip, since the strength of the learning synapses vanishes for voltages close to $V_{dd}$. Furthermore, the firing-threshold comparator of
the IF neuron has also a PMOS-input differential pair. Therefore, the firing threshold is well below the upper saturation limit of the voltage buffers.

5.2.1 Test setup

The arrangement used to test the chip is shown in Figure 5.4. The chip with the STDP neurons was mounted on a test board that includes a bank of potentiometers to set the current and voltage biases for the chip. In addition, the board is equipped with an array of comparators that flash a set of colour-coded LEDs when a weight voltage goes beyond two thresholds set close to the maximum and minimum saturation limits ($V_{dd}$ and GND).

The stimulation and monitoring of the on-chip neurons was done through a logic analyser system (HP16700A). A pattern generator (HP16522A) was used to drive the $s[1:8]$ and $LongIn[1:5]$ input lines. The postsynaptic spikes were recorded and time stamped by a timing-capture-and-analysis module (HP1655D). The logic analyser also contains an oscilloscope which was used to record the evolution of the weight voltages during the experiments.

The logic analyser used is network enabled. It can run an FTP server and provides an Ethernet interface. This network interface was used to load the input spike trains generated on a PC. The listings of time-stamped postsynaptic events recorded were downloaded to a PC for post-processing.
5.2.2 Bias generation and tuning

As was seen in chapter 4, neurons in the chip must be tuned with several voltage and current biases. Figure 4.13 showed that the strength of the synapses and the peak of potentiation are set by currents replicated at the neuron level from a single input current. Here, Figure 5.5 shows how the bias currents are set at the level of the network. Three external variable resistors connected to either ground or supply are used to set the reference currents for $I_{bLTD}$ (peak of depression), $I_{bLTP}$ (peak of potentiation) and $I_{w,\text{set}}$ (maximum synaptic strength). The currents entering the chip are split between the five neurons. Figure 5.5 also shows that the voltage biases used in the five neurons are directly generated outside the chip using another set of external potentiometers.

The characterisation experiments given in the following sections show how the experimenter can adjust the variable resistors (i.e., tune the voltage and current biases) to change the characteristics of the neuron, the synapses and the learning window (peaks of potentiation and depression, etc.). Clearly, this bias-setting method is not robust against process variation. Thus, the experimenter may be required to make slight adjustments of the biases to maintain exactly the same window characteristics when different chips are plugged to the test board.
5.3 Spiking behaviour and weight update

Before fully characterising the learning windows that can be implemented with the chip, we tested that the neurons were functional. The waveforms of Figure 5.6 show that a neuron in the network can be stimulated through the learning synapses and the non-learning excitatory synapse. In addition, we see that the weights are modified at the timings of the presynaptic and postsynaptic spikes in a manner consistent with temporally asymmetric Hebbian learning.

The soma of the neuron, shown in the top trace of Figure 5.6 ($V_{\text{soma}}$), is initially reset to the GND rail. The weights of the six learning synapses are shown in the bottom trace. They are preset at $1.7V$ at the start of this test. Synapses are stimulated in a sequence. The arrows on top of the figure show the presynaptic spike timings for each synapse. The first three spikes increase the soma voltage but are not able to generate an action potential. Since no previous postsynaptic event had occurred, the weight voltages of synapses $\text{syn1}$, $\text{syn2}$ and $\text{syn3}$ remain unchanged at the timing of the presynaptic spikes. Next, a very strong pulse of current is injected with
Neuron and learning rule characterisation

Figure 5.6: Test results showing the spiking behaviour of the neuron and weight updates.

the non-learning excitatory synapse (synNL), causing the neuron to fire. The neuron self-resets its soma immediately and synapses which received a spike before the postsynaptic neuron are potentiated (i.e. $V_w$ decreased). Synapse syn3, stimulated immediately before the postsynaptic event, is strongly potentiated ($V_{w3}$ decreases around 120mV). Synapses stimulated earlier undergo less potentiation. Synapse syn1 was stimulated 8msec before the postsynaptic event and, in consequence, is only slightly strengthened. Those synapses which were not stimulated before the postsynaptic event are not potentiated at all.

Let us now focus on the input activity delayed with respect of the postsynaptic spike. The refractory period of the neuron is set very short so that the soma voltage reflects the activation of the synapses which receive spikes after the postsynaptic potential. At the timing of each
presynaptic spike the weights of synapses \textit{syn4} to \textit{syn6} decrease ($V_w$ increased). The synapse stimulated immediately after the postsynaptic spike (\textit{syn4}) is depressed more than the synapse with a larger presynaptic delay (\textit{syn6}).

Similar experiments to the one just described were carried out, but with simultaneous activation of multiple neurons and multiple synapses. No interference between neurons was detected.

5.4 Synaptic-strength modification

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{synaptic_strength modificarion.png}
\caption{Test results show that the strength of the synapse is controlled by the weight voltage $V_w$.}
\end{figure}
Figure 5.8: Measurement of the synapse transduction for three different settings of $I_{w\_set}$. The peak of the EPSP in the soma increases with $(V_{dd} - V_w)$.

Figure 5.7 gives a scope capture of the neuron soma and the weight voltage of synapses 1 and 2 showing that the synaptic strength is controlled by the weight voltage $V_w$. Once again, the timing of the spikes reaching each synapse is indicated by the arrows.

Weights of synapses are initially set to 1.75V. At the start, synapses 1 and 2 are activated in isolation to show that they create a similar EPSP peak in the soma, of about 1.1V. Then, the neuron is stimulated in 5 occasions with the same input pattern; synapses 1, 3 and 4 are activated synchronously, and synapse 2 is activated 200μs later. The early synapses are able to bring the soma above the firing threshold, hence causing a postsynaptic event. Therefore, their weights are potentiated ($V_w$ decreased) at the timing of every postsynaptic event, as shown for $V_{w1}$ in the bottom trace of Figure 5.7. The spike in the soma is not shown with its full height due to the limited resolution of the scope at the time scale set for these traces. In contrast to the early synapses, synapse 2 is activated immediately after all postsynaptic events. Thus, it is strongly depressed ($V_w$ increased) as reflected by the $V_{w2}$ trace. The test ends with two additional isolated activations of synapses 1 and 2. The potentiated synapse (synapse 1) now creates a larger increase of the soma voltage, whereas the EPSP due to the depressed synapse...
(synapse 2) is smaller.

The dependence of the synaptic strength on the weight voltage is characterised by the curves of Figure 5.8. The change in the soma voltage created by a presynaptic spike was measured for different weight voltages. The weight-voltage input range of the synapse goes from the 0V ground rail to roughly 1V below the power rail ($V_{dd}$ was set to 5V). The different slopes of the three curves shown have been set with different $I_{w, set}$ bias currents. The slope depends also on the duration of the pre pulse generated by the pulse generator circuit. The maximum synaptic-strength value allowed can be tuned with the slope of the transconduction curve of the synapses. However, the strength of the synapses must always be considered with respect to the firing-threshold level. In general, it is simpler to modify the relative maximum synaptic strength by changing the firing-threshold level of the neuron.

5.5 The learning window

5.5.1 Test method

To characterise the potentiation part of the learning window implemented we repeated the following procedure several times:

1. Weights are initially reset to $W_{init}$;
2. one learning synapse is stimulated with a spike at time $t_{pre}$;
3. the non-learning excitatory synapse of the same neuron is activated with a delayed spike, increasing at each repetition the delay with respect to $t_{pre}$; and
4. finally, the voltage difference between the reset weight voltage value ($W_{init}$) and the modified weight value ($V_{w}$) of the stimulated learning synapse is measured.

The non-learning synapse was set to have a strong weight, so that the neuron fired at each stimulation. Thus, the timing of the spike exciting the non-learning synapse is equivalent to $t_{post}$, the timing of the postsynaptic event.

For the depression part of the learning curve the same procedure was used with the order of the spikes reversed; i.e., the learning synapse was activated after the non-learning excitatory synapse.
Neuron and learning rule characterisation

Figure 5.9: Measurement of the learning window implemented. The peak of the learning window for potentiation is controlled by $I_bLTP$. The peak of the learning window for depression can be tuned with $I_bLTD$. The results from 16 experiments were averaged for each point in the curves to eliminate a $\approx 2mV$ noise in the measurements.

5.5.2 Peaks and time constants

Following the procedure described, we first concentrated on the tunability of the peaks of the learning window with $I_bLTP$ and $I_bLTD$ (see schematics in Figures 4.10 and 4.11 in Chapter 4). Figure 5.9 shows four different peaks for the potentiation and depression parts of the learning window. As in the other curves shown below, we averaged results from 16 stimulations with the same pre-post delay for each point of the curve in order to eliminate a small voltage noise of about $2mV$ in the measurements. As was discussed in the presentation of the circuits in Chapter 4, a synaptic weight increase corresponds to a decrease of $V_u$. The curves look very much like the learning windows found in recordings of biological neurons [33]. When the postsynaptic spike happens a few $msec$ after the presynaptic spike ($t_{pre} - t_{post} < 0$) the
weight is increased. In contrast, if the presynaptic spike timing is delayed with respect to the postsynaptic spike \( (t_{\text{pre}} - t_{\text{post}} > 0) \) the weight is decreased.

![Graph showing the learning window](image)

**Figure 5.10:** Measurement of the learning window implemented. The decays of the learning window can be tuned with \( V_{\text{decLTD}} \) (depression) and \( V_{\text{decLTP}} \) (potentiation). The results from 16 experiments were averaged for each point in the curves to eliminate a \( \approx 2mV \) noise in the measurements.

Other important parameters of any TAR learning rule are the duration of the effective window for the induction of weight change and the decay time constants from the depression and potentiation peaks. Figure 5.10 shows that the decay of the learning windows implemented by the learning circuits proposed can be tuned for both potentiation and depression. To create msec time constants the leakage transistors of the *causal correlation* and *a-causal correlation* circuits are biased in weak inversion (transistor N3 in Figure 4.10 and N5 in Figure 4.11). The decay time constants are independent of the peaks of maximum weight change. Hence, it is easy to control the shape of the learning window.
As described in chapter 2, for weight-independent STDP the imbalance between the integral area of the potentiation part of the learning window must be smaller than the area under the depression part of the curve. Other investigations have found that, in neurons with a thousand learning synapses, an imbalance of 5% is required to achieve a balanced bimodal weight distribution able to regulate the output firing rate [12]. In our case, however, the generation of bimodal weight distributions requires the imbalance between the areas to be much larger. The reason being that, with a small number of synapses, the probability of a presynaptic spike causing a postsynaptic event is much higher than for neurons with hundreds or thousands of synapses. Therefore, the depression part of the learning curve must compensate for the increased probability of a pre-post spike interaction occurring on the potentiation (causal) part of the learning window \( t_{\text{pre}} - t_{\text{post}} < 0 \). For our neurons with only six STDP synapses, the peak for potentiation has been set in most learning experiments to be roughly half the size of the peak for depression. In addition, a slightly longer time constant for the depression (a-causal) part of the learning window also facilitates learning. The test results given in Figures 5.10 and 5.9 show that the learning windows can be tuned to achieve these requirements.

With close inspection of the learning windows in Figure 5.10, it can be seen that for large negative delays \( t_{\text{pre}} - t_{\text{post}} < 0 \) synapses are depressed very slightly. This small depression corresponds to a non-Hebbian term (i.e., non-dependent on correlations between pre- and postsynaptic spikes) in the actual learning rule implemented. It penalises all synapses which have not contributed to the generation of the spike, even if they are not activated within a few msec of the postsynaptic spike timing. Hence, this non-Hebbian term has the right sign to facilitate learning. However, this term is very weak in comparison with the size of the temporally asymmetric Hebbian learning window so, in most cases, it can be safely ignored when analysing the learning process.

5.5.3 Weight-dependent learning

An important goal in the design of the STDP synapse was to develop a learning rule which could have a well-controlled weight-dependence mechanism for potentiation. The three potentiation curves shown in Figure 5.11 were constructed using the same test method described above. Each of the three curves corresponds to a different initialisation weight voltage. Figure 5.11 shows clearly that, the stronger the synapse (lower \( V_w \)), the smaller the amount of potentiation. The difference between the potentiation peaks for \( V_w = 0.75V \) and \( V_w = 2V \) is similar to the
Figure 5.11: Measurement of the learning window with the weight-dependent mechanism enabled. The amount of potentiation depends on the current value of the weight. The results from 16 experiments were averaged for each point in the curves to eliminate a $\approx 2mV$ noise in the measurements.

difference between the peaks for $V_w = 2V$ and $V_w = 3.25V$. Therefore, equation 4.4 gives a good description of the weight-dependence mechanism introduced in the learning rule. As a consequence of being based on the same transconductor configuration, the weight-dependence mechanism works for the same weight-voltage input range as the synapse. Moreover, the linearity of the synapse seen in Figure 5.8 is consistent with the results for the weight-dependence mechanism shown in Figure 5.11. Having weight dependence does not affect the control of the peaks and decay of the learning window. The weight-dependence mechanism can be completely disabled by setting bias voltage $V_{bc}$ equal to the power supply rail voltage ($V_{dd}$) in the causal correlation circuit presented in Figure 4.11
5.6 Summary

This chapter has presented test results that characterise the spiking behaviour, the synapse and the temporally asymmetric Hebbian learning window implemented. The aims of the design defined in Chapter 4 have been achieved: the synapse has a wide voltage input range; the integrate-and-fire circuit is functional and no interferences between neurons have been detected; and, the learning circuits are able to modify the synaptic-weight voltages according to a temporally asymmetric Hebbian learning rules in response to presynaptic and postsynaptic events.

Several graphs have shown that the learning window implemented is strikingly similar to those found in biological systems. It is relatively easy to tune the peaks and time constants for both sides of the STDP learning window. Finally, we have seen that the weight-dependence mechanism works as desired for the full weight-voltage range.
Chapter 6
On-chip learning experiments

6.1 Introduction

Chapter 5 showed that the circuits proposed can be used to study on-chip temporally asymmetric Hebbian learning. It was demonstrated that the learning circuits can be tuned for different learning window shapes. In addition, test results given demonstrated that the synapse works as intended and showed neurons firing in response to input activity. Now, it remains to be proven that correlations present in the input spike trains can be captured in some way by the weight distributions developed by the neurons fabricated. To this end, several learning experiments have been carried out with the same chip used for the characterisation tests of chapter 5. As is common in theoretical studies and software simulations of STDP, input spike trains were generated from Poisson processes with varying degrees of inter-synapse correlation.

This chapter starts describing the stimuli used in the experiments. It follows with results from experiments performed to investigate the development of bimodal weight distributions in response to uncorrelated spike trains. Then, results from spike-synchrony learning experiments with a single neuron are given, followed by results from experiments that test whether bimodal weight distributions can be maintained when the input correlations are removed. The chapter ends with results from a temporal-correlation learning experiment involving all neurons of the on-chip feed-forward network.

6.2 The stimuli

The sequences of action potentials produced by neurons in the central nervous system are often modelled as spike trains with Poisson-distributed spike timings. Partly to keep the project in a neuromorphic framework, and to make comparison with theoretical and neural modelling studies easier, the stimuli used in the experiments presented in this chapter have also been generated with Poisson processes.
Figure 6.1: Methods to build correlated spike trains. (A) A generating Poisson process is thinned to produce $N$ spike trains with zero-delay correlations. (B) A Poisson process with a time-dependent rate $r(t)$ can be used to create spike-timing correlations.
A Poisson spike train with firing rate $r$ has an interspike interval $T$ with probability density distribution $f_T(t) = r \exp(-rt)$. Therefore, when a Poisson process with constant rate (homogeneous Poisson process) is used, the spike train is a sequence of random spike timings which does not have any internal correlation, nor does it have any spike-timing correlations with other inputs.

Zero-delay correlations can be introduced easily with the thinning method sketched in Figure 6.1A [86]. An homogeneous Poisson process with rate $r_i$ is used to construct a "generating" spike train. A spike in the generating train has a probability $\beta$ of being copied to one of the correlated trains. The resulting correlated spike trains are also Poisson with mean rate $\beta r_i$, and the correlation coefficient between pairs of correlated spike trains is $\beta$. Similar zero-delay correlations have been used to study temporal-correlation learning with STDP in software [87]. Temporal correlations between spike trains can also be introduced using a Poisson process with a time-dependent rate (inhomogeneous Poisson process). Some of the input spike trains for the experiments presented in this chapter have been generated using a binary rate signal for the Poisson process (see Figure 6.1B). The Poisson process used switches between periods of low spike density ($r_{g1}$) and pulses of high probability of firing ($r_{g2}$). The probability of two spikes from different spike trains coinciding inside a short time window is higher during the pulses of increased rate. A refractory period longer than the duration of the pulses of increased rate (see $T_w$ in Figure 6.1B) is enforced to ensure that the correlations are strictly between pairs of spikes. For the stimuli used in the experiments described in section 6.6, the onsets of the pulses of increased probability of firing (see timings $t_1$ to $t_5$ in Figure 6.1B) are also drawn from a Poisson process.

6.3 Bimodal weight distributions with uncorrelated inputs

Several studies have analysed the effect of temporally asymmetric Hebbian learning rules on the weight distribution [12, 39]. It has been shown that weight-independent STDP creates bimodal weight distributions, with weights having either maximum or minimum strength. As was discussed in chapter 4, the bimodality of the weight distributions is of interest for the analogue VLSI implementation.

Traditionally, Hebbian learning algorithms have been studied using mean firing rates. It is clear that a group of Poisson spike trains with slow-changing firing rates does not encode information.
using precise spike timings. Spike-based learning can provide unique insights into neural coding if we consider information encoded by spike-timing coincidences. However, spike trains with constant firing rate are also used to investigate spike-driven learning rules. By studying the learning process when correlations are not present on the inputs, some essential properties of the learning rule can be identified easier than if we focus only on the correlation learning task. In particular, the effect of temporally asymmetric Hebbian learning on the weight distribution can be studied effectively using uncorrelated input spike trains.

6.3.1 Temporal weight evolution

The first test result that illustrates the bimodality of the weight distribution created in the chip is given in Figure 6.2. It shows the temporal evolution of the six synaptic weights of a neuron stimulated with uncorrelated spike trains. Each synapse received a different spike train with mean rate of 30Hz. The spike trains lasted 40 seconds. The spike trains did not have a refractory period. The strength of the synapses — relative to the firing threshold — and the leakage in the soma were set so that a postsynaptic spike was generated when at least two synapses with maximum strength \( V_w = 0 \) were activated with a delay smaller than 6.6msec. The weight voltages of all synapses of the neuron were initially set to 1.35V. Generally, in temporally asymmetric Hebbian learning synapses must be set initially strong, so that enough postsynaptic spikes are generated [12]. If weights are weak at the start of the experiment the postsynaptic activity is low. In consequence, synapses receive little potentiation. Therefore, the amount of potentiation is not enough to compensate weight decay due to charge leakage in weight capacitors (see additional discussion on this in section 6.3.2). For all experiments discussed until section 6.6, weight-independent learning rules were used (i.e., the weight-dependent mechanism in the causal correlation circuit was disabled).

We see in Figure 6.2 that the weight of synapses 1, 3, and 5 (represented by weight voltages \( V_{w1} \), \( V_{w3} \), and \( V_{w5} \), respectively) are reinforced to values close to the maximum synaptic strength \( V_w = 0 \). The other three synapses have their weight voltages \( V_{w2} \), \( V_{w4} \) and \( V_{w6} \) increased toward the power rail \( V_{dd} = 5V \), which makes the synapses lose all their strength. Note that the voltage buffers used to monitor the weights saturate at approximately 4V. However, as we saw in the synapse transconductance curves of Figure 5.8, the strength of the synapse is effectively zero for weight voltages above 4V.

The same experiment was repeated with other instances of input spike trains with the same sta-
Figure 6.2: Weight evolution caused by uncorrelated inputs.

Statistical parameters. Every time, two groups of synapses developed, with weight voltages above 4V or inside a band of a few hundreds mVs above 0V. In every instance of the experiment, it was seen that when a synapse is weakened to the minimum weight it becomes unable to regain its strength. Similarly, a synapse never loses its strength after becoming strongly potentiated. In other words, the binary weight distributions that emerge are stable. The synapses developing strong weights changed from trial to trial. No tendency for a group of synapses to always become stronger in conjunction was apparent. Therefore, the bimodal weight distribution created at one particular trial can only be explained by the random spike timings in the first few seconds of the spike trains used in that trial.

Bimodal weight distributions developed with uncorrelated spike trains of the same mean rate are due to an unstable learning process, which can be described as follows. When a synapse is reinforced it becomes more likely to cause a postsynaptic spike, and thus more likely to be reinforced further by the learning rule. In contrast, synapses which are not reinforced have less probability of causing a postsynaptic spike. Therefore, given the negative area-integral of the STDP learning window, these synapses are more likely to be depressed. This positive
feedback created by the learning rule exists even when inputs are uncorrelated. By chance, in the first few seconds of the experiment some synapses receive spikes within a short time window, which allows them to generate an action potential. Thus, this group of synapses are all reinforced by the learning rule. Therefore, less synchrony in their activation is needed to cause another postsynaptic event in subsequent stimulations.

Weight distributions created by weight-independent temporally asymmetric Hebbian learning are very different from those created by mean-rate Hebbian learning rules, such as the Oja rule. In weight-independent temporally asymmetric Hebbian learning, the weight vector does not align with the principal component of the input rate vector. As seen in Figure 6.2, binary weights emerge even when all inputs have the same mean firing rate. In all experiments conducted with this chip using weight-independent learning and uncorrelated inputs, weights always became binary if the neuron was stimulated long enough.

6.3.2 The effect of input mean rate on the weight distribution

The probability that several uncorrelated spike trains will trigger the postsynaptic neuron depends on the mean rate of the input spike trains. When inputs fire at high rate, the probability that two or more spikes from different inputs will occur within a short time window increases. The effect of the mean firing rate of the input spike trains on the weight distribution of the neurons is illustrated by the experiment results given in Figure 6.3.

The weight distributions created by input spike trains with different mean rates were measured in several trials. The neuron was stimulated long enough to allow synapses to develop either maximum or minimum strength. At each trial, all six learning synapses of the neuron received different spike trains with the same mean rate. The number of synapses that developed maximum and minimum strength for a specific mean rate of the inputs changed from trial to trial — due to the instability of weight-independent STDP and the fact that the neurons have few synapses. For instance, for input spike trains with a mean rate of 30Hz, the neuron developed either three maximum and three minimum weights, or four maximum and two minimum. The graphs in Figure 6.3 show the result of averaging the final weight distribution of ten trials per mean-rate value. The circled-point curve in Figure 6.3 gives the average number of synapses with zero weight created for each input frequency. The squared-point curve shows the average number of synapses which developed maximum strength.
For the test results given in Figure 6.3, some preliminary testing was done to set the parameters of the neuron so that at 30Hz the neuron often developed three maximum weights and three zero weights. The point where the two lines cross depends on the average amount of current injected into the soma. For instance, the crossover point will shift to the left (i.e. less synapses with maximum strength for the same input mean rate) if the maximum allowed strength for the synapses is increased. Similarly, the crossover point will also shift to the left if the leakage in the soma is decreased. This shift is due to the increased synaptic competition when high average current is injected into the soma. Weights were initialised in all trials with the same weight voltage $W_{\text{init}} = 1.35V$. A refractory period was not imposed on the spike trains.

When the mean rate of the inputs is low, there is also a low number of coincident spike timings in the input spike trains. In response to this, the learning process creates large weights for most of the input synapses. We see in Figure 6.3 that, when all synapses receive spike trains with 15Hz mean rate, five strong synapses developed on average. When the frequency of the input spike trains increases, there are more opportunities for random coincidences of spike
On-chip learning experiments

timings. Consequently, a smaller group of synapses can drive the postsynaptic neuron on their own. At 45Hz only two synapses became maximally potentiated in the ten trials. The on-chip weight distributions shown in Figure 6.3 are in agreement with the theoretical analysis of STDP learning using the Fokker-Plank equation presented in [37] and [39], and the simulations in software reported in [12]; to compensate for the increase in the mean current coming from the synapses due to the firing rate increase, the mean input weight vector decreases.

The neuron was also stimulated in the same set of experiments using spike trains with a mean rate of 10Hz, as well as with spike trains with frequencies of 50Hz and above (not shown in the graphs). For the parameters set for the learning window — potentiation decayed to 15% of its peak value in \(5msec\), whereas depression decayed to 45%— a set of 10Hz spike trains did not produce enough postsynaptic spikes to overcome the leakage in the weight capacitors. As a result, in all trials with 10Hz inputs the six weights of the neuron decayed slowly (i.e., weight voltages increased slowly to a value close to 4V). Hence, in this experiment 15Hz is the lower operation range limit. This low mean-rate limit depends on the parameters set for the neuron and the learning window, and the charge leakage in the weight capacitors (cf. 3.4.1). The leakage is temperature dependent and was seen to vary between different weight capacitors.

With spike trains of mean rate higher than 50Hz, sometimes the neuron developed a single strong synapse. This happened because, with high-frequency spike trains without a refractory period, there is an increased probability that several EPSPs due to the same synapse can bring the postsynaptic neuron’s soma to the firing threshold without cooperation from other synapses. In some of the trials with 50Hz inputs a single strong synapse developed but then the neuron became silent. What had happened was that at the beginning of these trial a synapse had been stimulated with several pairs of consecutive spikes with short delay. This highly stimulated synapse had been able to generate a postsynaptic spike without cooperation from other synapses. As a result, this single synapse had developed a strong weight and the rest had been weakened severely. But at 50Hz, a single maximum-strength synapse was not able to generate enough average postsynaptic activity to beat the leakage in the weight capacitors. In consequence, the weight that was initially reinforced to the maximum value decayed slowly. Thus, it became more and more difficult for the synapse to generate an action potential. This undesired situation can be avoided by imposing a refractory period on the spike trains. It is also possible to reduce its likelihood if the rate of learning (set with the peaks of the learning window) is small.
6.4 Learning temporal correlation with a single neuron

We have seen that the hardware neurons proposed create bimodal weight distributions when the input spike trains are uncorrelated. In those experiments, the learning process was driven by random spike-timing coincidences between the uncorrelated inputs. The next question that will be addressed is whether temporal correlations can be learnt with a single neuron using weight-independent STDP.

The weight evolution seen in Figure 6.4 was created by the stimulation of a neuron with three correlated spike trains for synapses 1 to 3, and three uncorrelated spike trains for synapses 4 to 6. For this experiment, all inputs had a mean firing rate of 30Hz. The correlated spike trains for synapses 1 to 3 had a 0.4 correlation coefficient, and had been generated with the "thinning" technique described in section 6.2. A refractory period of 4msec was imposed on all input spike trains. All weights were initialised to 1.45V at the beginning of the experiment.

The weight of synapses 1 to 3 (represented by $V_{w1}$ to $V_{w3}$, respectively) are driven quickly to

![Figure 6.4: Learning from zero-delay correlations. Synapse 1 to 3 have a 0.4 correlation coefficient. Synapses 4 to 6 are uncorrelated.](image)
On-chip learning experiments

the maximum weight value ($V_w = 0$). They reach the maximum level after receiving around 60 spikes. The weight voltages of synapse 4 to 6 ($V_{w4}$ to $V_{w6}$) increase toward the power supply rail. As with the experiments with all inputs uncorrelated, a bimodal weight distribution emerges. This time, however, synapses with correlated activation are far more likely to cause collectively a postsynaptic event than the other synapses. Therefore, spike timings of the correlated spike trains fall with more frequency on the causal part of the learning window. In this experiment, the decay time constants of both sides of the learning window were set roughly equal (for $|t_{pre} - t_{post}| = 5 ms$ the weight change curve decayed to 28% of the peak). The depression peak was twice the size of the potentiation peak. Thus, due to the imbalance between potentiation and depression, synapses which receive uncorrelated inputs lose the competition imposed by the learning rule.

Many similar experiments have been carried out with two or three correlated inputs fighting against another group of uncorrelated inputs. The weight distributions created were always bimodal and stable. With correlation coefficients of 0.4 and higher, the correlated inputs almost invariably win the competition. However, given the instability of weight-independent STDP and the randomness of the input spike trains, there is a possibility that for a particular trial a synapse from the uncorrelated group is reinforced and a synapse with correlated input weakened. The likelihood of this situation occurring increases when the correlated inputs have a lower correlation coefficient.

Figure 6.5 shows 1 sec of the input and output spike trains for the same trial of Figure 6.4. The section of the spike trains represented starts at the 19th second after the onset of the stimulation. Thus, the activity shown occurs when a bimodal weight distribution has already developed. The top three spike rasters ($pre6$ to $pre4$) correspond to the uncorrelated inputs. Spike rasters labelled with $pre1$ to $pre3$ show the spike trains of synapses 1 to 3. The bottom raster is the output of the neuron.

As seen in the spike rasters of Figure 6.5, the output neuron fires only when there is a higher density of spikes in the rasters of $pre1$ to $pre3$. The firing threshold of the neuron was set 2.5 times larger than the peak of the EPSP caused by a synapses with maximum strength. Thus, the three strong synapses must be activated within a short temporal window to generate a postsynaptic event. Synapses 4 to 6 have lost all their strength, so their spikes are uncorrelated with the postsynaptic events.
On-chip learning experiments

Figure 6.5: Spike rasters for a temporal correlation learning experiment. The synapses which receive spike trains pre1-3 have developed strong weights. The synapses of pre4-6 have zero weight.

6.5 Weight retention

Many implementations of artificial neural networks in analogue VLSI face the challenge of storing the analogue weights learnt. The experiments carried out with the on-chip neurons show that synapses become essentially binary — having either maximum or zero strength — with both uncorrelated and correlated stimuli. If the chip with the STDP neurons is used in a system with a clear distinction between a learning phase and recall phase, it would be easy to store the weights with a latch triggered at the end of the learning phase. Here, however, the attention will be placed on the weight retention that weight-independent learning can provide at the algorithmic level. The question that will be addressed is whether a specific set of binary weights generated by input correlations can be maintained when the correlations are removed.

The results from the first experiment carried out to study weight retention are given in Figure 6.6. As in the experiment of Figure 6.4, synapses 1 to 3 of a neuron were initially stimulated by $30Hz$ spike trains with 0.4 correlation coefficient. Synapses 4 to 6 received uncorrelated spike trains with a mean rate of $30Hz$. After $15sec$, the initial correlation was removed and all synapses were stimulated with uncorrelated spike trains of the same mean rate for $30sec$. 

103
Figure 6.6: Weight retention experiment. Weights are stable after the correlation disappears. The input spike trains have an 8msec refractory period.

more. A refractory period of 8msec was enforced in all six spike trains for the whole 45sec. The learning rule used was weight independent. In all weight-retention experiments discussed here the peaks of potentiation and depression were 75mV and 117mV, respectively, and the learning windows decayed to 28% of their peak value in 5msec. The firing threshold was set 2.6 times larger than the peak of the EPSP created by a synapse with maximum strength. The leakage in the soma was set so that the peak of the EPSP decayed from its peak to 0V in 9msec.

As seen in Figure 6.6, two groups of synapses have formed by the time the temporal correlation disappears from the spike trains of synapses 1 to 3. Once again, synapses which receive spike-timing correlations win the competition imposed by the learning rule. After the correlations between synapses 1 to 3 have been removed, the weights remain stable at the value created in
the first 15 sec.

The results we saw in Figure 6.3 showed that weight distributions for uncorrelated input spike trains are frequency dependent. Thus, the weight retention achieved with the instability of weight-independent learning should also be affected by changes in the mean rate of the spike trains that stimulate the neuron after the correlations are withdrawn. It was also argued in section 6.3.2 that short refractory periods do not encourage cooperation between synapses.

Figure 6.7: Weight retention is not achieved if the mean rate of the inputs increases after the temporal correlations disappear and the spike trains have a short refractory period.

Figure 6.7 shows the weight evolution in a weight-retention experiment where the uncorrelated stimuli of the last 30 sec had a dramatic mean rate increase. As in Figure 6.6, for the first 15 sec all spike trains have a mean rate of 30 Hz. Then, for the remaining 30 sec shown, the mean rate of all six input spike trains is increased to 80 Hz. For the whole 45 sec of the experiment
a 2msec refractory period has been enforced in all input spike trains. In the first phase of the experiment, the weights of the correlated inputs \( (V_{w1} \text{ to } V_{w3}) \) reach the maximum weight level \( (V_w = 0) \), whereas the synapses with uncorrelated inputs are depressed. After the 15sec mark, the short refractory period in conjunction with the dramatic increase in mean rate destabilises the weight distribution created. At 80Hz, the mean current created with a single synapse is able to generate a high output rate. Therefore, to compensate for the increase in mean synaptic current, the mean weight vector decreases. The weight voltages of synapses 1 and 3 increase rapidly toward the power supply, and only synapse 2 remains potentiated.

Figure 6.7, shows that a weight which reaches the maximum value in the first phase of the experiment may lose its strength when the correlations are removed. Weakening of strong synapses happens when a refractory period has not been imposed in the input spike trains and the mean rate increases when the input correlations are withdrawn. This effect is a consequence of the regulation imposed by weigh-independent STDP. Thus, it is not an issue specific to the hardware implementation. It should also happen in continuous-learning systems implemented in software.

Similar weight-retention experiments have been conducted where the mean rate of the input spike trains decreased after the 15sec mark. It was seen that the lower the mean rate in the second phase of the experiment, the higher the probability that some synapses loose their strength. It is easy to see why this happens. In the second phase of the experiment, synapses with zero weight cannot contribute to the generation of the postsynaptic spike. Consequently, with a purely Hebbian learning rule there are not enough causal interactions between spikes reaching zero weight synapses and the postsynaptic spikes for the synapses to regain their strength. Thus, when the mean rate of the inputs decreases severely, the number of random coincident spikes reaching the strong synapses may not be able to compensate for the charge leakage in the weight capacitors. The low frequency limit for weight retention applies only to the hardware implementation. In software simulations, the output neuron would stop firing but the weights would not decay. The lower frequency limit for weight retention has been found to vary with changes in room temperature and across the synaptic array. For the neuron settings used in all weight-retention experiments (see above), some synapses could retain their weights even with mean rates of 5Hz and spike trains without refractory period.
Hierarchical synchrony detection and amplification

The hierarchical detection and amplification of spike-timing synchrony was investigated using the feed-forward network in the chip. The experiment described next involves all five hardware neurons included in the chip fabricated. This time, instead of using zero-delay correlations, the spike synchrony is generated with an inhomogeneous Poisson process as described in Figure 6.1B.

In several preliminary on-chip learning experiments it was found that learning rules with moderate levels of weight dependence allow learning subtler correlations than with weight-independent learning rules. Weight-dependent learning creates a less unstable learning process which allows the correction of "mistakes" made early in the learning process. For instance, a particular synapse may be reinforced at the beginning of the experiment due to random spike-timing coincidences. In moderate weight-dependent learning — with smaller weight potentiation for large weights but weight-independent depression — the increased strength of the synapse will not be self-maintained unless on average the synapse receives synchronised activity. In contrast, with weight-independent learning early "mistakes" are amplified by the positive feedback of the learning rule.

In the learning experiment with the full network presented below, the learning window was set to be weight dependent (see Figure 5.11 in chapter 5). As will be seen next, the weight-dependence circuit was adjusted so that the weight distributions created are still bimodal.

Network configuration and stimuli

The configuration of the network that was introduced in chapter 5 is shown again in Figure 6.8. Only the 6 learning synapses of each neuron are used in this experiment. The fixed inhibitory and excitatory synapse remain silent throughout. Each neuron in the input layer (N1 to N4) receives a set of six different spike trains. A single neuron in the second layer (N5) receives projections from the four neurons in the input layer and two extra spike trains supplied externally. Synapses 1 and 2 of all neurons in the first layer receive Poisson-distributed spike trains with some degree of synchrony. Pairs of spikes from synchronised inputs have a higher probability to coincide in a short time window of duration $T_w$ (window of correlation). This first level of correlation is indicated in Figure 6.8 by the double-arrowed bridges $C_1$ and $C_3$. A second level of correlation is also introduced between the correlated inputs of N1 and N2 as
On-chip learning experiments

Figure 6.8: Stimuli and final weight distribution for the hierarchical learning experiment with the on-chip network.

indicated by the arrowed bridge $C_2$. Unlike N1 and N2, inputs to neuron N3 and those of N4 are independent. Synapses 3 to 6 of neurons in the first layer, and synapses 5 and 6 of the neuron in the second layer are stimulated with completely uncorrelated Poisson-distributed spike trains.

The rate signal $r(t)$ used to create the correlated inputs had pulses of increased firing probability that lasted for $T_w=5\text{ms}$ (see diagram in Figure 6.1B). The minimum distance between consecutive windows of correlation was set to 30ms. A refractory period of 10ms (notice that $T_w < 10\text{ms}$) was enforced between consecutive spikes to ensure that each spike train had only 1 spike for every pulse of higher firing probability. Thus, this type of correlation is purely temporal (between single spike timings) rather than based on an effective firing rate increase. For correlations $C_1$ and $C_2$, the two rate values of the binary signal were 10Hz and 350Hz. For $C_3$, the two values of the binary rate signal were 10Hz and 500Hz. The rate of the Poisson process which sets the timings of the onsets of the high-rate pulses was 10Hz for all three correlations. The actual spike trains for synapses 1 and 2 of N1 and N2 were constructed merging a spike train corresponding to correlation $C1$ and a spike trains corresponding to $C2$. The uncorrelated inputs (inputs 3 to 6) of first layer neurons were Poisson distributed with a rate of 18Hz, which is the same as the mean rate of the correlated spike trains averaged over the whole duration of the experiment. The 2 direct inputs to N5 had a rate of 7Hz, which is the output rate of the
neurons in the first layer for the given inputs when they develop bimodal weight distributions through STDP learning. All input spike trains had a minimum interspike interval of 10ms. Consistently, the refractory period of the neurons in the chip was set to 10ms.

6.6.2 Final weight distribution

The weight distribution of the network resulting from learning is bimodal; with weights having either maximum or minimum strength. In Figure 6.8, the final weight distribution for all synapses has been represented graphically. Synapses which developed maximum weight strength are marked by filled black circles. These are synapses of input neurons which received synchronised activity or synapses of N5 which received inputs from N1 and N2. In contrast, weights with final minimum strength are indicated by empty circles. These correspond to synapses of first layer neurons which received uncorrelated inputs or synapses of N5 which received inputs from neurons stimulated without a secondary level of correlations (N3-N4).

6.6.3 Weight evolution

The evolution of the weights of N2 and N5 toward the saturation limits is depicted in Figure 6.9. The weights of neuron N2 are shown on top. The weights of synapses 1 and 2 evolve toward their maximum strength (which corresponds to a low $V_w$ value). Weights of remaining synapses, which receive purely uncorrelated activity, decrease (i.e. $V_w$ increase). Weights of other neurons in the input layer evolve in a similar manner. The $V_w$ traces on the bottom graph of Figure 6.9 show how N5 captures the secondary level of correlation present at the input ($C_3$). Weights of synapses receiving input from N1 and N2 are reinforced while the rest are weakened. Clearly, the second layer will only capture features from signals which have already a basic level of interesting features (primary level of correlations) detected by the first layer.

6.6.4 Synchrony amplification

The graphs in Figure 6.10 give normalised cross-correlations histograms which characterise the synchrony detection and amplification carried out by the network. Graphs A-C give cross-correlations for the input spike trains. The effect of learning on the activity of the neuron is illustrated by graphs D-H (cross-correlation after learning) and I-J (correlations with learning inhibited). Learning inhibition is achieved by resetting all weights at intervals of 500ms. The
Figure 6.9: Weight evolution of neurons N2 and N5 for the hierarchical learning experiment with the on-chip network.
Figure 6.10: On-chip correlation amplification by feed-forward network of neurons with STDP synapses. The figure shows normalised cross-correlation histograms between several spike trains of the network.
On-chip learning experiments

Histograms only take into account spikes pairs which are closer than 60ms. They have all been equally scaled to highlight synchrony amplification. For a presynaptic and the postsynaptic spike trains of a neuron, the inter-spike interval (ISI) of the cross-correlation histograms is taken as $ISI = t_{pre} - t_{post}$. Hence, causal interactions between presynaptic and postsynaptic spikes correspond to negative values of ISI in the graphs presented. Notice that the cross-correlation graphs have been labelled with the numbers of the neurons and synapses. For instance, the cross-correlation between the spike train received by synapse $I$ of neuron N2 and the spike train for synapse $I$ of neuron NI is labelled with n2sI_n1sI.

Graphs A-B-C of Figure 6.10 show the correlation of the input spike trains. Figure 6.10A shows the normalised histogram of the correlation between the input 1 and 2 of neuron N1 (n1s1_n1s2). This histogram illustrates the correlation resulting from the combination of $C_1$ and $C_2$. Figure 6.10B gives the cross-correlation between the correlated synapses of N1 and N2 (n2s1_n1s1), which corresponds to correlation $C_2$ alone. The last graph characterising the input stimulus is 6.10C. It gives the flat cross-correlation histogram that corresponds to a pair of uncorrelated input spike trains.

Graphs D, E and F characterise the effect of the input layer after STDP learning has created the bimodal weight distribution depicted in Figure 6.8. The histogram in 6.10D shows the cross-correlation of the output spike trains of neurons N1 and N2. Clearly, the synchrony between these two spike trains is higher than the synchrony of the inputs due to $C_2$ shown in 6.10B. In 6.10E, we see the strong causality relationship between the spikes at the output of N1 and one of its synchronised inputs. The same cross-correlation histogram when learning is inhibited is shown in 6.10I. The correlation between pre and postsynaptic spikes after learning is much higher than for the neuron with learning inhibited. The learning process has amplified the synchrony transmission of the neuron by pruning those synapses which receive purely random stimuli. Neurons which do not receive mutually correlated activity (N3 and N4) have a flat output activity cross-correlogram (see 6.10F).

Finally, the effect of the output neuron (N5) is shown in Figure 6.10G and 6.10H. Since the secondary level of correlation has been amplified by the input layer (compare 6.10B and 6.10D) it can now be captured by N5 more easily. The graph in 6.10G shows the synchrony between one of the synchronised inputs of N1 and the output of N5. It contrasts heavily with the same cross-correlation histogram for the experiment with learning inhibited shown in 6.10J.
6.7 Summary

It has been demonstrated with results from silicon that the neurons designed can detect spike-timing correlations. Both weight-dependent and weight-independent learning rules have been used in the experiments. Some evidence that moderate levels of weight dependence in the learning rule allow learning weak correlations has been found.

In all learning experiments the weight distributions created were bimodal. It was seen that, under certain conditions, it is possible to retain the learnt bimodal weight distributions when temporal correlations disappear from the input.

We have seen that on-chip temporally asymmetric Hebbian learning creates a learning process which behaves in agreement to theoretical analysis of STDP. Test results show that the mean of the bimodal weight distributions decreases with the mean rate of the inputs spike trains.

It has been shown that the on-chip neurons can amplify spike-synchrony correlation. This is possible thanks to the bimodal weight distributions. It was shown that a hierarchical synchrony pattern embedded in noisy spike trains can be detected by the on-chip network. The first layer amplifies a distributed synchrony, which is then integrated by the output layer. The second layer can only detect features of the inputs if they form part of a primary stream of features detected at the previous processing stage.
Chapter 7
Summary, conclusions and future work

This thesis has studied temporally asymmetric Hebbian learning rules implemented in mixed-mode VLSI. New circuits to support this new class of spike-based learning rules have been proposed and validated with test results from a chip fabricated. The response of the fabricated neurons to temporal correlations has been analysed. The effect of both correlated and uncorrelated inputs on the weight distribution has also been studied.

This final chapter starts with a complete review of every chapter in the thesis that highlights the solutions proposed to each of the issues raised. It follows with a recapitulation of the conclusions reached. The chapter ends with a section on future work with several suggestions to extend the research presented in this thesis.

7.1 Review

Chapter 1 introduced neuromorphic engineering as a discipline interested not only in neural algorithms, but also in the physical elements used to compute in both artificial and natural neural systems. The fundamental laws of physics are very evident throughout the whole design process of analogue integrated circuits. For this reason, many analogue-circuit designers have been attracted by neuromorphic engineering. The aim of neuromorphic engineers is to develop new computing technology following the same structures found in natural neural systems.

Chapter 2 highlighted the increasing prominence given to the precise timing of spikes in the last decade. Traditionally, neural models have only considered the average firing rate of neurons. New coding schemes, such as spike-timing synchrony, become possible if precise spike timings are also taken into account. Temporal synchrony has been proposed to explain how information about different features of an object in the visual scene are bound together to form a coherent representation of the object in the brain. An increasing amount of evidence suggests that spike timing and spike synchrony should be considered in models of neuronal computation. The
most compelling evidence to support this idea comes from experiments on synaptic plasticity which show that synaptic strength changes depend on precise spike-timing differences between presynaptic and postsynaptic spikes (spike-timing-dependent plasticity). In many systems, the synapse is potentiated when the postsynaptic neuron fires a few msec after the presynaptic spike timing. With the order of firing reversed, the synapse is depressed. The learning rules underlying this type of asymmetric weight change are often known as temporally asymmetric Hebbian learning. They offer an alternative to the rate-only approach of traditional Hebbian learning algorithms. Several studies found in the literature show that the presence of weight dependence in the learning rule has a dramatic impact on the computational properties of temporally asymmetric Hebbian learning. When the weight change is independent of the current synaptic weight, the final weight distribution is bimodal, with weights close to either the maximum or minimum weight value. If potentiation is smaller for strong synapses (weight dependent), the learning process can lose its instability and the weight distribution becomes smooth and unimodal (i.e., with a single peak). Chapter 2 also presented the integrate-and-fire neuron model used in many studies of spiking neuron systems.

Chapter 3 discussed the implementation of spiking neurons in analogue VLSI. Different designs for the firing mechanism were examined. Capacitive-divider feedback is a robust technique for spike generation which has been validated by many previous designs. Different strategies to produce long time constants similar to those found in biological systems have been proposed. The most common is to use transistors biased in weak inversion with drain currents of the order of nAs. Chapter 3 emphasised that the main hurdle in the implementation of on-chip learning is the long-term storage of the analogue weight values created by most neural algorithms. Unfortunately, no ready-available technology exists to store analogue values in VLSI. It is easy to modify a weight represented by the voltage across a capacitor, but the charge stored in this way leaks through the junctions of transistors controlling the flow of current into the weight capacitor. Many neuromorphic chips use transistors biased in weak inversion in their synapse circuits. This type of circuits limit the voltage range of the weights to a few hundreds of mV. This limited weight-voltage range is sufficient for synapses with fixed weights (non-learning synapses), but reduces severely the dynamic range of the weights in a learning system. Furthermore, synapses with limited voltage range make learning more sensitive to charge leakage in weight capacitors.

Chapter 4 proposed novel circuits for the analogue VLSI implementation of temporally asym-
metric Hebbian learning. The characteristics of temporally asymmetric Hebbian learning that make it amenable to analogue VLSI were identified; namely, the signals involved in the weight change are local to the synapse, the lack of complex signal averaging over long-time windows, the absence of tight requirements for the shape of the learning window, and the bimodality of the weight distributions created with weight-independent STDP. This latter characteristic was an important motivator for the implementation of this type of learning rules. During the learning phase weights are analogue and the implementation can take advantage of the analogue circuits. At the same time, since final weights are binary, it should be easy to store them on-chip. Furthermore, bimodal weight distributions are also generated with completely uncorrelated inputs. We wanted to explore whether this bimodality could be used to sustain the weights when the correlations disappear from the inputs.

Weight-independent STDP is known to create an unstable learning process. The low number of synapses in a silicon neuron renders the learning process even less stable than in neurons with many synapses, such as those used to study STDP in software. We were interested in examining whether weight-dependent potentiation can be used to reduce the instability of the learning process in on-chip temporally asymmetric Hebbian learning, thus improving the performance of the neuron in learning weak temporal-correlations. For this project, the weight dependence had to be sufficient to reduce the instability of the learning process, but not too strong as to prevent the creation of bimodal weight distributions. Therefore, having a controllable weight-dependence mechanism became another requirement for the design of the learning circuits.

It was decided that weak-inversion transistors would be used only for the long time constants of the learning window and the slow decay in the soma voltage. Another design choice made was that the transconductor in the synapse circuit would be a transistor in strong inversion to increase the input-voltage range, thus reducing the effect of charge leakage in the weight capacitor.

The latter, and largest, part of chapter 4 presents in detail all circuits for the neuron. The integrater-and-fire circuit does not present any significant novelty and is very similar to previous designs found in the literature. The main contributions are in the learning circuits. The same small linearised-transconductor circuit is used in the synapse and the weight-dependence mechanism. The non-linear decay of the learning window is a direct consequence of the non-linear gate-to-drain transconductance of the MOS transistor.
A micro-chip with five neurons was fabricated to validate the circuits and methods proposed in chapter 4. It was used for the tests described in chapter 5 and chapter 6. Test results for other circuits proposed early into the project can be found in appendix B. In chapter 5, the synapse circuit is characterised and neurons are shown to be functional (i.e., they generate action potentials in response to stimulation). No apparent interference between synapses and neurons in the chip was detected. The chapter ends with several graphs which show that the learning circuits can be tuned to produced learning windows with different time constants and different peaks of maximum weight change. The weight dependence mechanism was also tested and found to work correctly.

Chapter 6 presented results from several learning experiments. The bimodality properties of on-chip weight-independent learning were found to be consistent with theoretical analysis of STDP reported in the literature. Neurons have been stimulated with different forms of temporal correlation. Synapses which receive correlated spike trains develop maximum-value weights, whereas uncorrelated synapses loose all their strength. A feed-forward network included in the chip was used to detect a hierarchical synchrony pattern embedded into noisy spike trains. For the network learning experiment the weight-dependent mechanism for the causal part of the learning window was enabled. Thanks to the bimodal weight distribution created, the network amplifies the spike synchrony of the input. Chapter 6 also presents an analysis of the weight retention capabilities of weight-independent learning when the input spike trains lose their correlations after the emergence of bimodal weight distributions. It has been found that weight retention is possible if a refractory period is enforced on the input spike trains.

### 7.2 Conclusions

This thesis set out to examine the suggestion that temporally asymmetric Hebbian learning in mixed-mode VLSI can support temporal correlation and spike-synchrony processing. From test results with a micro-chip it has been shown that the silicon neuron with STDP synapses proposed can detect temporal correlations in the form of spike-timing synchrony. Both zero-delay correlations and narrow time windows of correlation have been learnt by the chip. It has been demonstrated that an on-chip two-layer feed-forward network can extract a hierarchical pattern of temporal correlations from noisy data.

Weight independent-learning rules are known to create a highly-unstable learning process. In
order to stabilise the learning process, it was decided to design a learning circuit with weight-dependent potentiation. There is some evidence that with moderate weight dependence neurons were able to learn weaker correlations than with purely weight-independent STDP.

This project aimed also at examining whether the bimodality of the weight distribution created by weight-independent learning rules can help bypass the challenge of providing long-term storage for analogue weights. The hardware neurons developed bimodal weight distributions with both weight-independent learning, or a combination of weight-independent depression and moderate weight-dependent potentiation. The idea of using the instability of the learning process to maintain a stable binary weight distribution when the input correlations disappear has been tested. It was found that the retention of the weights requires a delay between consecutive spikes similar to the effective window for the detection of spike synchrony set by the leakage in the soma (i.e., a refractory period should be enforced in the spike trains). Note, however, that what is meant by weight retention is not clear-cut in a continuous learning system. The original weight distribution will be disrupted if new data with a very different set of correlations is presented. Bimodal weight distributions are also of interest in systems with a clear separation between learning and recall, since binary weights can be easily stored on chip.

The results given in chapter 6 show that, even with only six synapses, on-chip STDP creates bimodal weight distributions which are in agreement with theoretical analysis based on the Fokker-Plank equation. The mean of the weight vector has been seen to decrease when the mean rate of the input spike trains increases.

Neurons with bimodal weight distributions cannot be used as principal-components analysers (PCA). However, test results given show that a network with bimodal STDP can detect and amplify weak temporal correlations. Thus, the silicon neurons proposed could be used to process temporal correlations coming from noisy sensors.

7.3 Future work

The work presented has concentrated on generating bimodal weight distributions with both weight-independent and moderate-weight-dependent learning rules. Weight-independent learning can only create binary weights. However, the same chip used for the learning experiments in chapter 6 can be used to investigate the effects of strong weight-dependent learning. It would be interesting to examine whether stable unimodal weight distributions emerge from learning
when potentiation has a strong weight dependence.

To study weight-dependent learning further, a new chip with a modified version of the learning circuits incorporating an activity-dependent scaling mechanism — as described in [87] — could be fabricated. Learning experiments could be carried out to examine if the new chip is able to extract the principal components of the inputs. The extra circuits required could be inspired by the rate-adaptation mechanism used in some analogue VLSI spiking neurons [6].

It was discussed in chapter 5 that a small non-Hebbian depression component is already present in the learning rule (see Figure 5.10). Two well-controlled non-Hebbian terms could be introduced with a simple modification of the learning circuits [36]. The effect of a small non-Hebbian depression inflicted on the weight at each postsynaptic spike timing was already discussed in chapter 5. The other non-Hebbian component would potentiate very slightly the weight of a synapse at every presynaptic spike timing. In some of our experiments the input weight vector became too small to generate any postsynaptic activity. Unfortunately with a purely Hebbian learning rule the weight cannot escape from this state. With a slight potentiation of the weights at every presynaptic spike timing, the neuron could recover from these type of undesired situations.

Another interesting variation in the learning rule would be to include a threshold for weight change. An internal variable \( x \) represented by the voltage across a capacitor would be incremented or decremented with the weight change circuits proposed in this thesis. However, the actual weight of the synapses would only be depressed by a fixed amount when the variable \( x \) crossed a threshold below the reset value of \( x \), or potentiated when the variable increased above a potentiation threshold above the reset point. After each potentiation or depression of the weight, the internal variable would be reset. This alternative learning rule could reduce the noise in the trajectory of the weight and improve learning. Some evidence exist to suggest that weight changes are indeed binary [88]. This new approach suggested has some similarities with a spike-based learning rule implemented in VLSI by Fusi et al. [52].

At the network level, it would be interesting to use on-chip temporally asymmetric Hebbian learning to discriminate between two clusters of spike synchrony. A simple experiment that could be attempted would consists of two neurons with lateral inhibition receiving the same input spike trains. The common set of inputs would contain two different groups of correlated spike trains. Thanks to fabrication mismatches, each neuron might learn to respond to a
different group of correlated inputs.
Appendix A

Tables of device sizes

The tables of this appendix give device sizes and bias values for the circuits presented in chapter 4. As discussed there, current and voltage biases are tuned in the lab to set the parameters of the neuron desired. Thus, voltages and currents biases given in the tables are only indicative.

Some capacitors have been built with the thin oxide used for the gate of MOS transistors. According to the process datasheet a nominal MOS transistor in strong inversion has a gate-bulk capacitance of $2.76 fF/\mu m^2$ [48].

<table>
<thead>
<tr>
<th></th>
<th>W</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>5μm</td>
<td>17μm</td>
</tr>
<tr>
<td>N2</td>
<td>5μm</td>
<td>17μm</td>
</tr>
<tr>
<td>N3</td>
<td>1μm</td>
<td>1μm</td>
</tr>
<tr>
<td>N4</td>
<td>8μm</td>
<td>16μm</td>
</tr>
<tr>
<td>N5</td>
<td>1μm</td>
<td>8μm</td>
</tr>
<tr>
<td>N6</td>
<td>1μm</td>
<td>8μm</td>
</tr>
<tr>
<td>P1</td>
<td>4μm</td>
<td>1μm</td>
</tr>
<tr>
<td>P2</td>
<td>5μm</td>
<td>17μm</td>
</tr>
<tr>
<td>C_{soma}</td>
<td>Cap (CPOLY)</td>
<td>826fF</td>
</tr>
<tr>
<td>C_{fb}</td>
<td>Cap (CPOLY)</td>
<td>3.8pF</td>
</tr>
<tr>
<td>C_{ref}</td>
<td>Area (MOSCAP)</td>
<td>$400 \mu m^2$</td>
</tr>
<tr>
<td>V_{bref}</td>
<td></td>
<td>740mV</td>
</tr>
<tr>
<td>V_{leak}</td>
<td></td>
<td>721mV</td>
</tr>
<tr>
<td>V_{th}</td>
<td></td>
<td>2.5V</td>
</tr>
<tr>
<td>V_{down}</td>
<td></td>
<td>1.5V</td>
</tr>
</tbody>
</table>

**Table A.1:** Devices sizes for the IF neuron circuit of Figure 4.3.
### Table A.2: Devices sizes for the comparator of Figure 4.4-B.

<table>
<thead>
<tr>
<th></th>
<th>W</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>24μm</td>
<td>1.2μm</td>
</tr>
<tr>
<td>N2</td>
<td>24μm</td>
<td>1.2μm</td>
</tr>
<tr>
<td>N3</td>
<td>10μm</td>
<td>2.4μm</td>
</tr>
<tr>
<td>N4</td>
<td>10μm</td>
<td>2.4μm</td>
</tr>
<tr>
<td>N5</td>
<td>10μm</td>
<td>2.4μm</td>
</tr>
<tr>
<td>N6</td>
<td>1μm</td>
<td>2.4μm</td>
</tr>
<tr>
<td>P1</td>
<td>24μm</td>
<td>2μm</td>
</tr>
<tr>
<td>P2</td>
<td>12μm</td>
<td>2μm</td>
</tr>
<tr>
<td>P3</td>
<td>2μm</td>
<td>2μm</td>
</tr>
<tr>
<td>Vbcomp</td>
<td>3.6V</td>
<td></td>
</tr>
</tbody>
</table>

### Table A.3: Devices sizes for the leakage circuit of Figure 4.4-D. This circuit was not used in any experiment.

<table>
<thead>
<tr>
<th></th>
<th>W</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>1μm</td>
<td>1μm</td>
</tr>
<tr>
<td>N2</td>
<td>108μm</td>
<td>2μm</td>
</tr>
<tr>
<td>N3</td>
<td>2μm</td>
<td>27μm</td>
</tr>
<tr>
<td>N4</td>
<td>0.8μm</td>
<td>1μm</td>
</tr>
<tr>
<td>N5</td>
<td>0.8μm</td>
<td>1μm</td>
</tr>
<tr>
<td>P1</td>
<td>8μm</td>
<td>9μm</td>
</tr>
<tr>
<td>P2</td>
<td>2μm</td>
<td>9μm</td>
</tr>
<tr>
<td>P3</td>
<td>3μm</td>
<td>0.6μm</td>
</tr>
<tr>
<td>Csw</td>
<td>Cap (C POLY)</td>
<td>20fF</td>
</tr>
<tr>
<td>Idec_soma</td>
<td>8μA</td>
<td></td>
</tr>
</tbody>
</table>
### Table A.4: Devices sizes for the pulse-generator circuit of Figure 4.6.
Duration was always set only with $V_{pw}$ in all experiments.

<table>
<thead>
<tr>
<th>N1</th>
<th>W</th>
<th>3μm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L</td>
<td>1μm</td>
</tr>
<tr>
<td>P1</td>
<td>W</td>
<td>2μm</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>0.6μm</td>
</tr>
<tr>
<td>P2</td>
<td>W</td>
<td>8μm</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>10μm</td>
</tr>
<tr>
<td>P3</td>
<td>W</td>
<td>4μm</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>10μm</td>
</tr>
<tr>
<td>P4</td>
<td>W</td>
<td>4μm</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>10μm</td>
</tr>
<tr>
<td>Cpg Cap (CPOLY)</td>
<td></td>
<td>220fF</td>
</tr>
<tr>
<td>Cfb Cap (CPOLY)</td>
<td></td>
<td>800fF</td>
</tr>
<tr>
<td>VthPG</td>
<td></td>
<td>2V</td>
</tr>
<tr>
<td>Vpw</td>
<td></td>
<td>3.4V</td>
</tr>
</tbody>
</table>

### Table A.5: Devices sizes for the synapse circuit of Figure 4.8.

<table>
<thead>
<tr>
<th>P1</th>
<th>W</th>
<th>3μm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L</td>
<td>30μm</td>
</tr>
<tr>
<td>P2</td>
<td>W</td>
<td>30μm</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>3μm</td>
</tr>
<tr>
<td>P3</td>
<td>W</td>
<td>4μm</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>0.6μm</td>
</tr>
<tr>
<td>P4</td>
<td>W</td>
<td>3μm</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>30μm</td>
</tr>
<tr>
<td>P5</td>
<td>W</td>
<td>30μm</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>3μm</td>
</tr>
<tr>
<td>P6</td>
<td>W</td>
<td>4μm</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>0.6μm</td>
</tr>
<tr>
<td>Vw_set</td>
<td></td>
<td>0V</td>
</tr>
<tr>
<td>Iw_set</td>
<td></td>
<td>7.8μA</td>
</tr>
</tbody>
</table>
### Table A.6: Devices sizes for the non-learning synapses of Figure 4.8. The inhibitory synapse was not used in the experiments described in this thesis.

<table>
<thead>
<tr>
<th></th>
<th>W</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>3μm</td>
<td>0.6μm</td>
</tr>
<tr>
<td>P2</td>
<td>12μm</td>
<td>8μm</td>
</tr>
<tr>
<td>N1</td>
<td>3μm</td>
<td>1μm</td>
</tr>
<tr>
<td>N2</td>
<td>10μm</td>
<td>10μm</td>
</tr>
<tr>
<td>N3</td>
<td>10μm</td>
<td>10μm</td>
</tr>
<tr>
<td>Vexc</td>
<td></td>
<td>2.8V</td>
</tr>
</tbody>
</table>

### Table A.7: Devices sizes for the a-causal correlation circuit of Figure 4.10.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>W</td>
<td>7.2μm</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>30μm</td>
</tr>
<tr>
<td>N2</td>
<td>W</td>
<td>2μm</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>0.6μm</td>
</tr>
<tr>
<td>N3</td>
<td>W</td>
<td>3μm</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>30μm</td>
</tr>
<tr>
<td>N4-9</td>
<td>W</td>
<td>3.6μm</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>30μm</td>
</tr>
<tr>
<td>CLTD</td>
<td>Area (MOSCAP)</td>
<td>600μm²</td>
</tr>
<tr>
<td>VdecLTD</td>
<td></td>
<td>700mV</td>
</tr>
<tr>
<td>IbLTD</td>
<td></td>
<td>6μA</td>
</tr>
</tbody>
</table>
### Table A.8: Devices sizes for the causal correlation circuit of Figure 4.11.

<table>
<thead>
<tr>
<th></th>
<th>W</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>1.8μm</td>
<td>15μm</td>
</tr>
<tr>
<td>N2</td>
<td>1.8μm</td>
<td>15μm</td>
</tr>
<tr>
<td>N3</td>
<td>7.2μm</td>
<td>15μm</td>
</tr>
<tr>
<td>N4</td>
<td>0.8μm</td>
<td>0.6μm</td>
</tr>
<tr>
<td>N5</td>
<td>10μm</td>
<td>10μm</td>
</tr>
<tr>
<td>N6</td>
<td>1.8μm</td>
<td>30μm</td>
</tr>
<tr>
<td>P1</td>
<td>3.2μm</td>
<td>1μm</td>
</tr>
<tr>
<td>P2</td>
<td>3.2μm</td>
<td>1μm</td>
</tr>
<tr>
<td>P3</td>
<td>2.4μm</td>
<td>6μm</td>
</tr>
<tr>
<td>P4</td>
<td>2.4μm</td>
<td>6μm</td>
</tr>
<tr>
<td>Cltp</td>
<td>Area (MOSCAP)</td>
<td>600μm²</td>
</tr>
<tr>
<td>Vbc</td>
<td>2.6V</td>
<td></td>
</tr>
<tr>
<td>IbLTD</td>
<td>6μA</td>
<td></td>
</tr>
</tbody>
</table>

### Table A.9: Devices sizes for the weight-change circuit of Figure 4.12.

<table>
<thead>
<tr>
<th></th>
<th>W</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>0.8μm</td>
<td>0.6μm</td>
</tr>
<tr>
<td>N2</td>
<td>1μm</td>
<td>4μm</td>
</tr>
<tr>
<td>N3</td>
<td>1μm</td>
<td>4μm</td>
</tr>
<tr>
<td>P1</td>
<td>3.2μm</td>
<td>1μm</td>
</tr>
<tr>
<td>P2</td>
<td>3.2μm</td>
<td>1μm</td>
</tr>
<tr>
<td>P3</td>
<td>2.4μm</td>
<td>6μm</td>
</tr>
<tr>
<td>P4</td>
<td>2.4μm</td>
<td>6μm</td>
</tr>
<tr>
<td>Cw</td>
<td>Area (MOSCAP)</td>
<td>3420μm²</td>
</tr>
</tbody>
</table>
### Table A.10: Devices sizes for the current-copying circuits of Figure 4.13.

<table>
<thead>
<tr>
<th>Device</th>
<th>W</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1-6</td>
<td>6μm</td>
<td>9μm</td>
</tr>
<tr>
<td>N7</td>
<td>12μm</td>
<td>9μm</td>
</tr>
<tr>
<td>P1-6</td>
<td>4μm</td>
<td>6μm</td>
</tr>
<tr>
<td>P7</td>
<td>8μm</td>
<td>6μm</td>
</tr>
</tbody>
</table>
Appendix B
List of publications


3. A. Bofill-i-Petit and A. F. Murray, "Synchrony Detection by Analogue VLSI Neurons with Bimodal STDP Synapses" in Advances in Neural Information Processing Systems 16 (Sebastian Thrun, Lawrence Saul and Bernhard Schölkopf, eds.), MIT Press, Cambridge (MA), 2004. (Note: This paper was presented as full oral presentation at NIPS 2003, held in Vancouver (Canada) in December 2003.)


References


References


Austria Micro Systems International AG, *0.6μm CMOS CUP process parameters*, 1998.


