LEARNING IN SILICON: A NEUROMORPHIC MODEL OF THE HIPPOCAMPUS

John Vernon Arthur

A DISSERTATION

in

Bioengineering

Presented to the Faculties of the University of Pennsylvania in Partial

Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2006

Kwabena Boahen,	Christopher S. Chen,
Supervisor of Dissertation	Graduate Group Chair

Acknowledgements

Many people contributed in many ways to my time at Penn. I thank them all.

Specifically, I thank my advisor, Kwabena Boahen. He has guided me through many challenges and has provided a constant example of what a researcher should be and can accomplish. Our interactions taught me a great deal and shaped my vision of research, science, and engineering for the better.

I thank my committee - Matt Dalva, Leif Finkel, and John Hopfield - who provided advice as well as new insights and different viewpoints that challenged my assumptions, improving my work.

I thank those that I had the opportunity to work with - Kareem Zaghloul, Kai Hynna, Brian Taba, Paul Merolla, Bo Wen, John Wittig, Thomas Choi, Rodrigo Alvarez, and Joe Lin - for the time we spent at work and at play. I thank them and many others for their friendship.

I thank my family, my parents for exemplifying integrity and work ethic, and my siblings for their encouragement. Finally, I thank my wife, Amy, for immeasurable contributions—her enduring support, her constant willingness to help, and her extraordinary tolerance.

Abstract

LEARNING IN SILICON: A NEUROMORPHIC MODEL OF THE HIPPOCAMPUS

John Vernon Arthur

Supervisor: Kwabena Boahen

The human brain is the most complex computing structure in the known universe; it excels at many tasks that digital computers perform poorly, such as learning input patterns and later retreiving them with only a part of the original patterns as input, realizing associative memory. Our brains perform these feats rapidly and with unmatched energy efficiency, using only about 10W, far less than a typical light bulb. To explore neurobiological processing, neuromorphic engineers use existing silicon technology to duplicate neural structure and function down to the level of ion channels, efficiently morphing brain-like computation into mixed analog and digital integrated circuits.

In this dissertation, we present a neuromorphic model of the hippocampus, a brain region critical in associative memory. We model hippocampal rhythmicity for the first time in a neuromorphic model by developing a new class of silicon neurons that synchronize by using shunting inhibition (conductance-based) with a synaptic rise-time. Synaptic rise-time promotes synchrony by delaying the effect of inhibition, providing an opportune period for neurons to spike together. Shunting inhibition, through its voltage dependence, inhibits neurons that spike out of phase more strongly (delaying the spike further), pushing them into phase (in the next cycle).

In addition, we use these neurons to implement associative memory in a recurrent network that uses binary-weighted synpases with spike timing-dependent plasticity (STDP) to

learn stimulated patterns of neuron activity and to compensate for variability in excitability. STDP preferentially potentiates (turns on) synapses that project from excitable neurons, which fire early, to lethargic neurons, which fire late. The additional excitatory synaptic current makes lethargic neurons fire earlier, thereby causing neurons that belong to the same pattern to fire in synchrony. Potentiation among neurons in the same pattern store it such that, once learned, an entire pattern can be recalled by stimulating a subset, which recruits the inactive members of the original pattern.

Contents

List of Figures

Chapter 1

Introduction

Neuromorphic engineers aim to reproduce the spike-based computation of the brain by morphing its neuroanatomy and neurophysiology into custom silicon VLSI (very large-scale integration) chips [101]. Custom hardware, using mixed analog and digital circuits, enables them to create large-scale real-time neural models. These models duplicate the mechanisms the brain uses to compute, in pursuit of not only its function but also of its efficiency, the *neuromorphic approach*. In contrast, the *bioinspired approach* uses traditional engineering techniques merely to realize the perceived function of biological elements and not their structure or efficiency.

Neuromorphic engineers have constructed large-scale real-time models of several neurobiological systems. Most implementations emulate peripheral sensory regions (such as the retina [158] and the cochlea [150]), however more recently, engineers have progressed to modeling the processes of (visual) cortical development, which includes generating orientation maps [106] and organizing axonal wiring in the retino-tectal pathway [138, 139]. Despite the importance learning from experience, attempts to model adult plasticity have

excluded system-level learning and have instead been limited to models of plastic synapses [53, 48, 19, 71].

The ability to learn from experience and to apply that knowledge to future situations is paramount to the success of every vertebrate animal. The key element of such learning is linking, *associating*, an event with its context and other temporally contiguous events (possibly deciphering its cause), which enables an animal to pursue or to avoid repeating the event, depending on whether it was beneficial or harmful, respectively. The neural correlate of such associative memory is thought to be the storage and recall of patterns of activity, pattern memory.

In this dissertation, I design and build an associative memory system capable of storing and recalling patterns, the Silicon CA3. I base the model on a part of the brain, the CA3 region of the hippocampus, which is strongly linked to learning and memorizing sequences of events, episodic memory. The system stores patterns in a manner that creates a scaffold for storing sequences of patterns by simultaneously representing multiple patterns multiplexed in time. The outline of the dissertation follows.

1.1 Dissertation Structure

In Chapter 2, I define episodic memory and review evidence that the hippocampus plays a role in its realization. Then, I review hippocampal biology relevant to episodic memory. Next, I introduce the fundamental components of associative memory networks, followed by their use in previous hippocampal models of memory. Finally, I present my model of hippocampal episodic memory and introduce a neuromorphic implementation of part of the model.

In Chapter 3, I present the silicon neuron. After describing my design strategy for realizing neuromorphic circuits, I present a conductance-based neuron that uses log-domain low-pass filters to build the neuron's soma and synapses. Next, I characterize the neuron circuit, including the soma's frequency-current response, the synapse's rise-time and decay constant, as well as the neuron's phase-response curve. Last, I discuss the neuron's role in the hippocampal model.

In Chapter 4, I present a network of silicon inhibitory interneurons that synchronize at gamma frequency (30-100Hz). First, I introduce synchrony in inhibitory networks, particularly in the hippocampus. Then, I discuss previous models of synchrony, silicon, analytical, and computational, and the insights into synchrony they provide, emphasizing the role of delay in synchrony. Next, I show with a simple analytical model that synaptic rise-time can behave as a surrogate for pure synaptic delay in controlling network frequency. Then, I switch my focus and describe the inhibitory interneuron network. I quantify the network's performance in terms of its frequency range and a measure of synchrony. Finally, I discuss the implications of the network performance at the system level.

In Chapter 5, I describe the design and characterization of a spike timing-dependent plasticity circuit. First, I summarize the observed properties of biological plasticity in the hippocampus. Then, I describe previous implementations of plasticity. Next, I present the spike timing-dependent plasticity circuit, describing the design optimizations and tradeoffs. I characterize the circuit's behavior and variability. Finally, I discuss its implications at the system level in the hippocampal model as well as the design's limitations.

In Chapter 6, I use a network of silicon neurons with plastic synapses and a theta rhythm (6-12Hz) to phase-encode the input stimulus and to compensate for variability among neurons. First, I summarize hippocampal properties related to the storage of autoassociative

patterns. Then, I describe previous VLSI implementations of abstract associative memory, contrasting them with neuromorphic associative memory. Next, I present my network of silicon neurons and its recurrent connectivity. I demonstrate the network's ability to compensate for variability while it stores and recalls patterns. I characterize this ability to compensate for intrinsic (fabrication) as well as extrinsic (input) variability. Finally, I discuss the implications of the implementation of autoassociation as well as its limitations.

In Chapter 7, I conclude with a discussion of my results and avenues for future work.

1.2 Original Contributions

In this dissertation, I present a novel model of sequence memory in the hippocampus (Chapter 2) in which connections from CA3 to the dentate gyrus (another region of the hippocampus) store the link between subsequent patterns in a series.

To realize my model, I conceived a new way to design silicon neurons (Chapter 3), based on current-mode log-domain and pulse-extender circuits. These circuits are about as dense as simpler integrator-type silicon neurons yet model conductance-based dynamics, which are more representative of biological neurons.

I use these silicon neurons to implement the first neuromorphic model of spike synchrony by inhibition (Chapter 4). The analysis I provide establishes synaptic rise-time as a surrogate for pure delay. Also, my analysis quantitatively determines the synaptic decay-constant's modulatory role in determining the frequency of synchrony. I verify these relationships with observations from the silicon network.

Further, I implement the first spike timing-dependent plasticity circuit in deep-submicron CMOS (Chapter 5). My circuit exploits the density of deep-submicron CMOS by using one bit of digital memory to store the each binary weight. Despite its implementation in deep-submicron CMOS, the circuit achieves a long integration time, which imbues it with noise immunity. And despite transistor variations, all 21,504 circuits function with the correct sign of weight change.

Finally, I implement the first neuromorphic model of associative memory that stores and recalls patterns, and completes patterns when presented with partial input (Chapter 6). Further, the network compensates for variability among its neurons, achieving precise timing. Precise timing allows multiple patterns to be active simultaneously at different phases of the theta rhythm, providing an ideal temporal structure for storing sequences. This observation led me to postulate that neuronal networks with excitatory recurrent connections achieve precise timing (synchrony) by potentiating connections from excitable neurons to lethargic ones—the plasticity enhanced phase-coding hypothesis.

Chapter 2

A Model of the Episodic Hippocampus

The hippocampus is an essential brain region in episodic memory, the storage and recall of series of events in space and time. Although the hippocampus is among the most studied regions of the brain, from its synaptic plasticity to its role in navigation, how hippocampal structure and function relate to episodic memory remains unexplained.

In this chapter, I define episodic memory and review evidence that the hippocampus plays a critical role in its realization. Then, I review the hippocampal biology relevant to episodic memory. Next, I introduce the fundamental components of associative memory networks, followed by their use in previous hippocampal models of memory. Finally, I present my model of hippocampal episodic memory.

2.1 Episodic Memory

An episode is a series of related events. Intuitively, episodic memory is the memory of experiences, storing and recalling events including time and location. In this dissertation, I adopt this definition of episodic memory[1].

Episodic memory is related to semantic memory, which recalls facts but not necessarily when and where they were learned. Episodic memory is inherently autobiographical as opposed to semantic memory—even though all semantic memories are acquired first as episodic memories. Episodic memory and semantic memory together form declarative memory, the memory of facts and events that can be recounted (in humans).

Episodic memory is inherently related to the storage and recall of sequences of events. In an example episode, an individual walks to the grocery store, picks up a loaf of bread, pays for the bread, and takes a bus home. An episodic memory system should store this sequence of events, such that it can be recalled later. Experience tells us such episodes must be stored with *one-shot learning*, as humans and animals need not experience an episode many times to remember it. This storage and recall of sequences of events is hypothesized as a role of the hippocampus.

[1] Psychologists and neuroscientists sometimes use a more restrictive definition of episodic memory in which a memory is only considered episodic if the subject relives the event through its recollection, an ability only known to exist in humans [143]. They refer to the definition I adopted as "episodic-like" memory [29, 79]. Because I am concerned with the mechanics of memory and not the sensation of a subject, I use the intuitive and less restrictive definition [38, 90].

2.2 The Episodic Hippocampus

The mammalian hippocampus is a brain region noted for its role in memory (Figure 2.1). Neuroscientists widely recognized such a role in the 1950s, when a patient, H.M., underwent an experimental surgery to treat his epilepsy, bilateral removal of the hippocampus and surrounding brain tissue. The surgery eliminated H.M.'s ability to store memories of new experiences[2] and erased many memories formed up to 11 years before the damage but left older memories intact [127]. Patient H.M. and others with similar hippocampus damage showed definitively that the hippocampus is critical in memory and suggested the types of memory in which the hippocampus plays a role; exactly what roles it plays, and how it performs these roles have been studied intensively and extensively ever since.

One early and influential hippocampal theorist, David Marr, proposed that the hippocampus was a long-term but temporary store for episodic memories [99]. He put forth that the hippocampus was a simple memory system (compared to the neocortical memory system) that rapidly stored events as they arrived as patterns. When activated, a pattern would stimulate areas of the neocortex to recall an event, acting as a sort of index into its sensory details. Further, he posited that these memories were transfered entirely to the neocortex slowly over time, providing a possible explanation as to why H.M. and others with hippocampal damage express complete retrograde amnesia a short time into the past (a few days) and reduced retrograde amnesia further back into the past (up to about a decade). By the time Marr proposed his model of hippocampal function, several neurophysiologists were studying properties of the hippocampus and its constituent elements.

To elucidate the hippocampus's role in memory, many neurophysiologists employ the

[2] H.M. lost the ability to form new memories about his experiences and facts, but he remained able to form new procedural (motor) memories/skills, although he did not recall having learned them [127].

Figure 2.1: The Human Hippocampus

The human hippocampus is a sausage-shaped brain region in the temporal lobe. Taken from the Digital Anatomist, University of Washington (www9.biostr.washington.edu/da.html).

rodent preparation. Rats (and other rodents) provided an abundant source of subjects whose hippocampi can be easily accessed and manipulated, both surgically and pharmacologically; mice enable neuroscientists to perform genetic manipulations as well [141]. These neurophysiological studies provided insight into hippocampal function.

Neurophysiologists O'Keefe and Dostrosky discovered rat pyramidal neurons that spiked selectively in specific locations within a given environment, place fields [115] (Figure 2.2). Pyramidal neurons with these properties were called *place cells* and formed the first plausible basis for a cognitive map, a global representation of an environment [116]. Later, place cells were discovered in the entorhinal cortex (the primary input to the Hippocampus), but their place fields occupied similar locations in similar but different environments, whereas hippocampal place cells' fields occupied completely different regions in each new environment, emphasizing their strong context dependence [122].

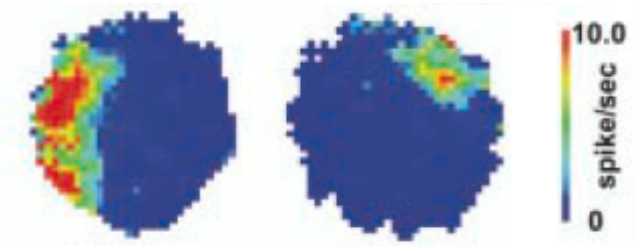

Figure 2.2: Place Cells

Place cells in the CA1 (and CA3) region of the hippocampus spike selectively in certain locations as a rat navigates the environment. Modified from [113].

Ever since the discovery of place cells, the role of the hippocampus in navigation has largely been explored in rodents. Neuroscientists typically use the Morris Water Maze, a deep circular pool with a podium just below the surface on which a rat (or mouse) can stand [110]. Naive rodents are placed in the pool and swim until they find the podium, learning its location. When returned to the pool, normal (control) rodents swim directly to the podium. Rodents with bilateral hippocampal lesions however, search for the podium again, apparently unable to remember its location.

Although they generate rich data on the rodent hippocampus, these navigational studies were difficult to relate to human studies. Many researchers were unable to reconcile the apparent differences between the human hippocampus, responsible for episodic memory, and the rodent hippocampus, responsible for navigational memory. Some neuroscientists concluded that the rodent hippocampus was only a navigational memory, simpler than the episodic memory of the human equivalent [116].

In the 1990s, evidence arose that suggested a solution for the hippocampal conundrum (although controversy continues). Neuroscientists tested the rate hippocampus's role in tasks that required more than just navigational memory and found that it encoded more

than just space. In particular, Hampson and colleagues showed that hippocampal neurons encoded place, trial phase, combinations of place and trial phase, and even trial type in a complex delayed-nonmatching-to-sample task[3] [56], variables consistent with episodic memory and far beyond a simple cognitive map (Figure 2.3). In a review article, Howard Eichenbaum made a strong case that the hippocampus is a "memory space" and not a "space memory", suggesting navigational memory is a subtype of episodic memory (Figure 2.4) [38].

Conversely, other evidence supported the notion that the human hippocampus stores navigational memories. Humans (epilepsy patients) have place cells, showing a navigational aspect to the episodic human hippocampus [39], and humans with exceptional navigational memories, London taxi drivers, have larger than average hippocampi [95].

Additional evidence suggests the (rodent) hippocampus is not only important in episodic memory but that it realizes such memory by storing and recalling sequences. Sequence memory naturally implements episodic memory, representing each event in an episode as a pattern in a sequence. Conversely, episodic memory could be implemented as pattern memory in which each episode is represented by a single pattern [36, 57]. One study showed that in rats the hippocampus is necessary to recall sequences of odors, but not to recognize individual odors [44]. Others showed that the hippocampus repeats sequences of activity during rest, both forward [132, 112] and backward [45], providing an opportunity to store the sequence in long-term memory and to link events to previous events, potentially enabling an animal to think back into the past (Figure 2.5).

[3] In the delayed-nonmatching-to-sample task, the subject (rat) traverses an enclosed room and presses an extended lever on either the left or right, depending on which lever is extended. The subject then returns to the other side of the room and waits for a delay period with its nose pressed into a hole. After the delay period, both levers are extended and the subject traverses the room again and presses the opposite lever. Hippocampal damage impairs rats in this task.

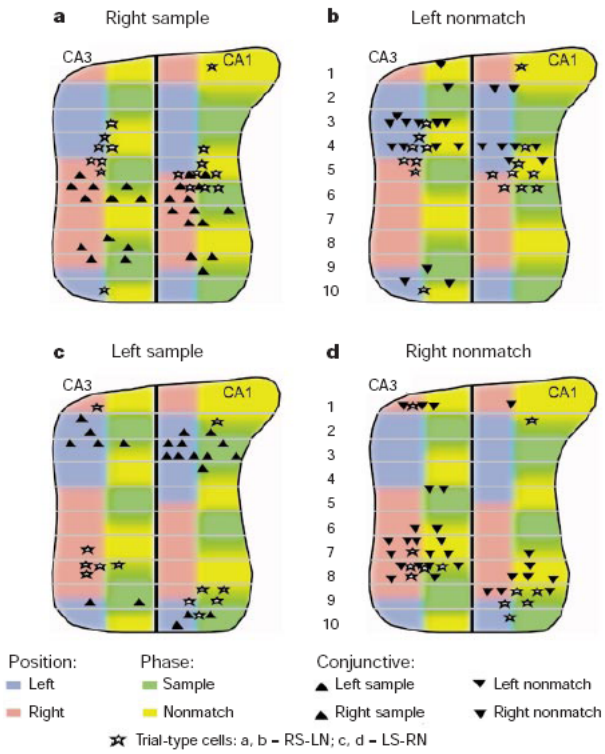

Figure 2.3: The Episodic Hippocampus

Neurons in the CA1 and CA3 regions of the hippocampus of the rat spike selectively in response to variables associated with episodic memory such as place, trial phase, combinations of place and trial phase, and even trial type. Modified from [56].

2.3 Hippocampal Biology

Ever since the hippocampus was discovered to have a crucial role in episodic memory, it has been extensively studied, yet we still do not understand it. We have an abundance of anatomical and physiological data, but we have yet to determine the mechanics of episodic memory, to connect biology with psychology and structure with function.

This section reviews the basic elements of hippocampal anatomy and physiology, fo-

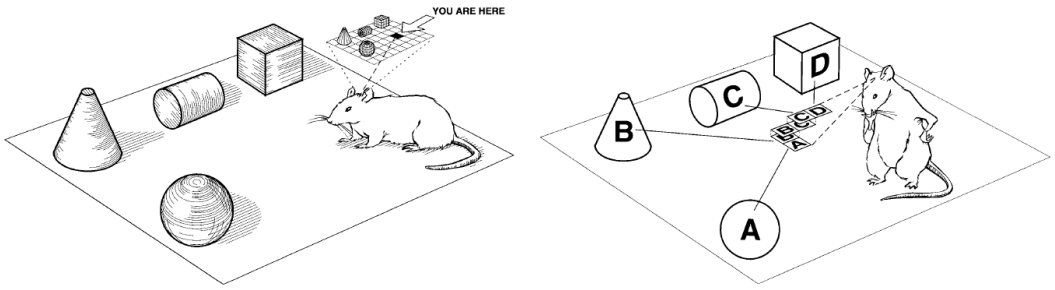

Figure 2.4: Memory Space Not Space Memory

The hippocampus is a memory space, encoding many types of information rather than just a cognitive map or space memory. Modified from [38].

cusing on elements directly related to the models of hippocampal memory we present in Sections 2.5 and 2.6 of this chapter. Specifically, I cover the hippocampus's neurons, their connectivity within and between regions, synaptic plasticity, and rhythmicity.

The hippocampus is a component of the limbic system, which is involved in emotion, motivation, and memory; humans have two hippocampi, one in each temporal lobe. *Hippocampus* comes from Greek for seahorse, because coronal slices of the hippocampus resemble the spiral of a seahorse's body and tail. The hippocampus as well as its subregions are also referred to as cornu ammonis (CA), Ammon's horn, due to the intact hippocampus's hornlike shape. The hippocampus proper consists of two major subregions: CA1 and CA3[4]. The hippocampal formation includes the hippocampus proper, the dentate gyrus (DG), and the subiculum (and related structures)[5]. In this dissertation, I am concerned with the hippocampus proper (especially CA3) and the DG, the union of which I refer to simply as the *hippocampus*.

[4]Between CA1 and CA3, resides a narrow subregion, CA2, which I and other researchers generally ignore.

[5]The subiculum, presubiculum, and parasubiculum are thought to be little more than an output relay from CA1 to various neocortical regions [30], which probably reflects our naivety more than any experiments.

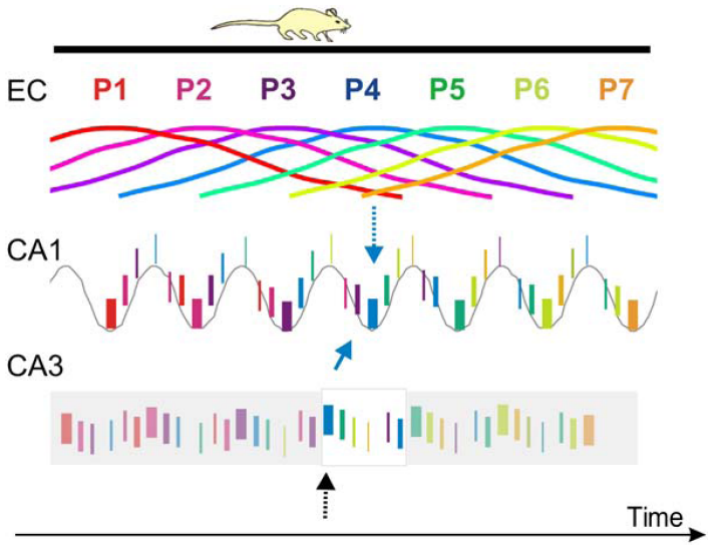

Figure 2.5: Hippocampal Sequence Activity

Neurons in the hippocampus spike in sequence, at times relative to the theta rhythm. Modified from [36].

The hippocampus is three-layer cortex, or archicortex, which is phylogenetically older and simpler than six-layer neocortex. In the CA regions one layer of pyramidal neurons cohabitate with sparse inhibitory interneurons of various types [46]. In the DG, densely packed granule neurons cohabitate with inhibitory interneurons and sparse mossy neurons (which resemble pyramidal neurons).

2.3.1 Hippocampal Neurons

In this section, I briefly describe the neurons of the DG and CA regions: granule neurons, mossy neurons, pyramidal neurons, and inhibitory interneurons (Figure 2.6)[6].

[6]In this dissertation, I use the terms *granule neuron*, *mossy neuron*, and *pyramidal neuron* to describe granule cells, mossy cells, and pyramidal cells, respectively. I use *neuron* to emphasize that these particular cells are neurons, rather than glia or nonneural cells.

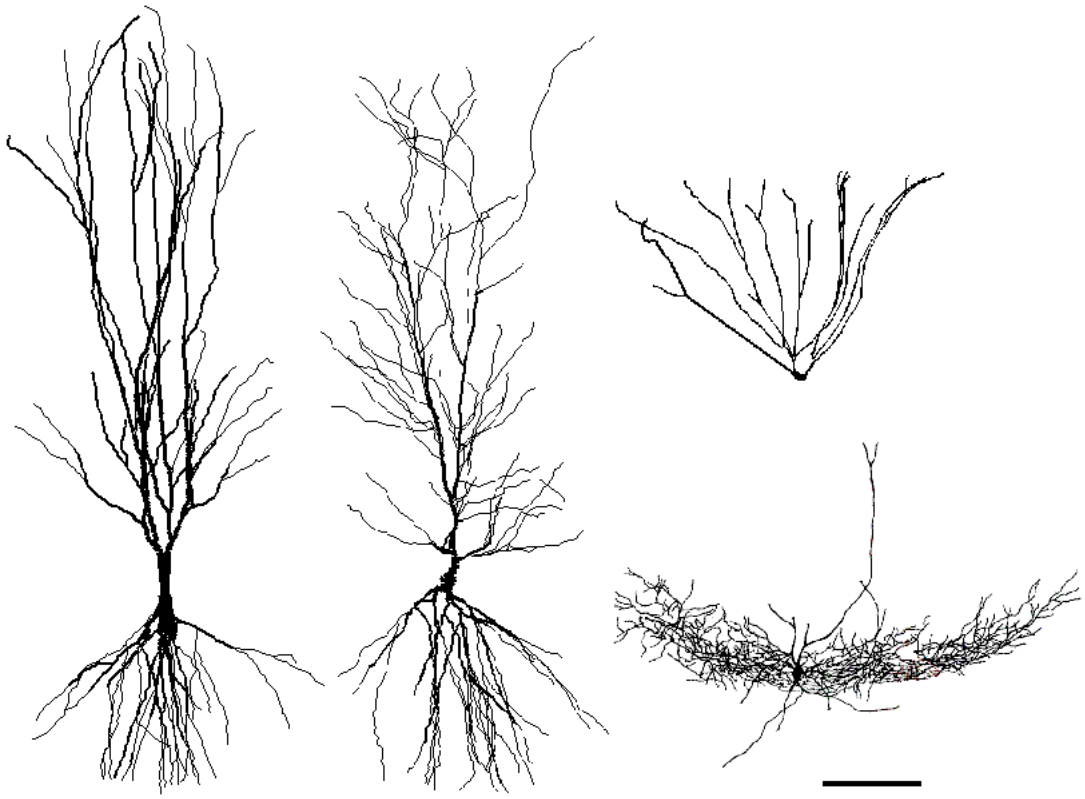

Figure 2.6: Neurons of the Hippocampus

CA3 (*left*) and CA1 (*middle*) pyramidal neurons have large somas and branching apical and basal dendrites. Dentate gyrus granule neuron (*top right*) have small somas and branching apical dendrites. Inhibitory interneurons, such as basket neurons (*bottom right*), typically have small somas, small dendritic trees (thick lines), and numerous local axonal branches (thin lines). Taken from the Cell Gallery at the Krasnow Institute for Advanced Study, George Mason University (http://krasnow.gmu.edu/cng/cell_gallery/).

2.3.1.1 Granule Neurons

The DG has two excitatory neurons types: granule neurons and mossy neurons. Granule neurons are the only principal type in the DG, meaning only granule axons project out to other parts of the hippocampus (or brain). Mossy neurons are interneurons, the only known excitatory interneuron in the hippocampus.

Granule neurons are small and densely packed in the granule layer of the DG, with their numbers estimated at about 1 million in the rat [120]. Their somas have a diameter of only about 10μm. Their apical dendrites extend into the molecular layer; they have no basal dendrites. They spike only sparsely with about one-half of one percent of the population active during active behavior in the rat [9]. Their axons, mossy fibers, project both into the hilus (layer of the DG), connecting to mossy neurons apical dendrites, and into the CA3, driving pyramidal neurons; they drive inhibitory neurons in both areas as well. Inputs to granule neurons come from the perforant path (entorhinal cortex, layer II) and from the mossy neurons in the hilus.

2.3.1.2 Mossy Neurons

Mossy neurons are larger than granule neurons with a somatic diameter several tens of microns. They are the most common neuron type in the hilus of the DG, numbering about forty thousand in the adult rat [120]. They have both apical and basal dendrites, which extend into the molecular and hilar layers, respectively. Little is known about their spike properties or activity level, but they are often assumed to be similar to pyramidal neurons in the CA regions, which they resemble. They send axons to granule neurons and inhibitory interneurons in the DG [78]. Mossy neurons receive inputs from the perforant path and

from mossy fibers.

2.3.1.3 CA3 Pyramidal Neurons

Pyramidal neurons are the principal neuron type in CA3, numbering on the order of two-hundred thousand in the adult rat. They have both apical and basal dendritic trees. They spike sparsely, with about 2.5 percent of the population spiking during active behavior. Pyramidal neurons are noted for their large active dendritic trees that can support dendritic spikes, which often result in somatic bursts [4, 148]. Their repertoire of dendritic and somatic ion channels has been well studied, resulting in detailed compartmental model neurons. Yet, little is known about how dendritic compartments and active channels influence information processing (but see [121]). CA3 pyramidal neurons receive inputs from both the perforant path and mossy fibers, as well as from other pyramidal neurons. They send their axons to CA1 (Schaffer collaterals), to other CA3 pyramidal neurons (associational/commissural collaterals), and to mossy neurons in the DG.

2.3.1.4 CA1 Pyramidal Neurons

Pyramidal neurons are the principal neuron type in CA1, numbering on the order of three-hundred thousand in the adult rat. Like CA3 pyramidal neurons, they have both apical and basal dendritic trees. CA1 pyramidal neurons support both rapid single spikes and bursts. CA1 pyramidal neurons receive inputs from entorhinal cortex (EC) layer III (not perforant path) and from CA3 pyramidal neurons via the Schaffer collateral pathway. They send their axons out of the hippocampus to EC layer III and (through the subiculum) to layers IV and V.

2.3.1.5 Inhibitory Interneurons

In contrast to only three types of excitatory neurons, the hippocampus includes numerous (at least several dozen) types of inhibitory interneurons [46]. Inhibitory interneurons are characterized by the release of GABA and local axon projection. They occupy every layer of the CA regions from oriens to molecular and every layer of the DG region from hilar to molecular. Some have large dendritic trees that cross all layers, from apical to basal, others have small trees, confined to one layer. Some have large branching trees that occupy whole regions (within a coronal slice) and others are localized to a narrow region. Some interneurons prefer to fire single spikes others bursts of spikes. We know there is a great variety of inhibitory interneurons, but we do not yet know their purpose. Clearly, they influence excitability, timing, rhythmicity, plasticity, and other properties, but we do not even know how these influence hippocampal computation.

2.3.2 Connections

The basic connectivity between regions of the hippocampus has long been known, the trisynaptic circuit.

2.3.2.1 Trisynaptic Circuit

The basic schematic of hippocampal connectivity is a simple trisynaptic circuit that describes the connectivity among excitatory neurons between hippocampal subregions (Figure 2.7) [78]. In the trisynaptic circuit, axons from the perforant path fan out, innervating DG granule neurons (and CA3 pyramidal neurons). Granule neurons project their axons

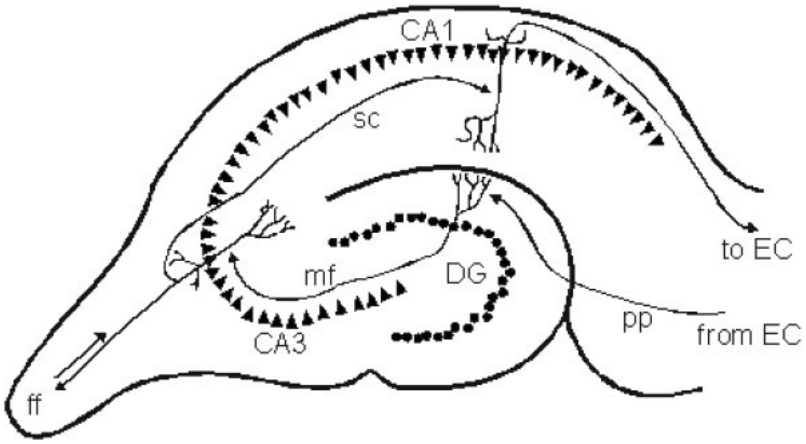

Figure 2.7: The Trisynaptic Circuit

The basic connectivity of the hippocampus is a trisynaptic circuit. Within a coronal slice, entorhinal cortex (EC) excites dentate gyrus (DG) granule neurons via the perforant path (PP). Granule neurons project their mossy fibers (MF) onto CA3 pyramidal neurons. CA3 neurons send Schaffer collateral (SC) axons to excite CA1 pyramidal neurons, the output of the hippocampus (to EC). Modified from [78].

(mossy fibers) onto synapses on CA3 pyramidal neurons. CA3 pyramidal neuron axons (Schaffer collaterals) fan out onto synapses on CA1 pyramidal neurons. Finally, CA1 pyramidal neurons project axons back to the EC. For a long period, neuroscientists assumed this simple circuit was responsible for all hippocampal processing, and although it provided a good starting point, it lacked many potentially important connections observed in the hippocampus both within each region and between regions.

2.3.2.2 Dentate Gyrus Excitatory Connectivity

The DG excitatory connectivity is the most complex in the hippocampus with two types of excitatory neurons: granule neurons and mossy neurons, which form an excitatory reciprocal network (Figure 2.8). The perforant path fans out, driving granule neurons and

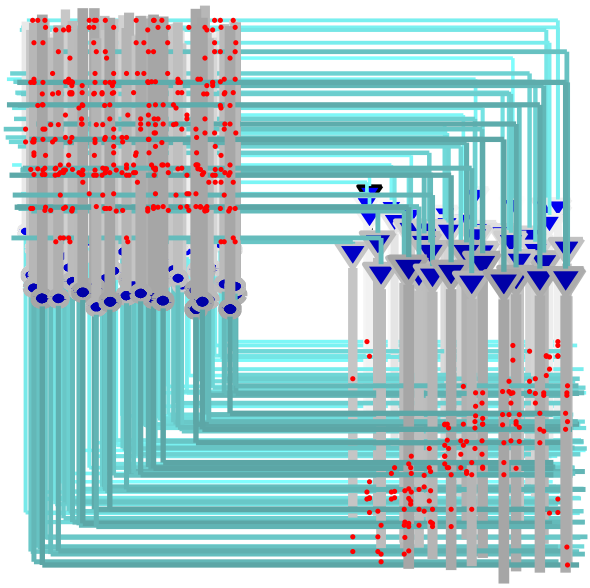

Figure 2.8: Dentate Gyrus Connectivity

Granule neurons (*left*) project mossy fibers to drive nearby mossy neurons (*right*), which reciprocally drive granule neurons across the dentate gyrus.

mossy neurons. Granule neurons excite about ten mossy neurons and about ten CA3 pyramidal neurons near their septotemporal level (within a coronal slice), with powerful mossy fiber boutons [59]. Mossy neurons reciprocally excite on the order of ten thousand granule neurons, but fan out along the septotemporal or long axis of the hippocampus.

2.3.2.3 CA3 Excitatory Connectivity

The CA3 excitatory connectivity is simpler than the connectivity in the DG, and includes another massive recurrent connection, one of the most extensive in the brain (Figure 2.9). Each pyramidal neuron receives recurrent connections from tens of thousands of its neighbors from all parts (along the septotemporal axis) of the hippocampus (though the proba-

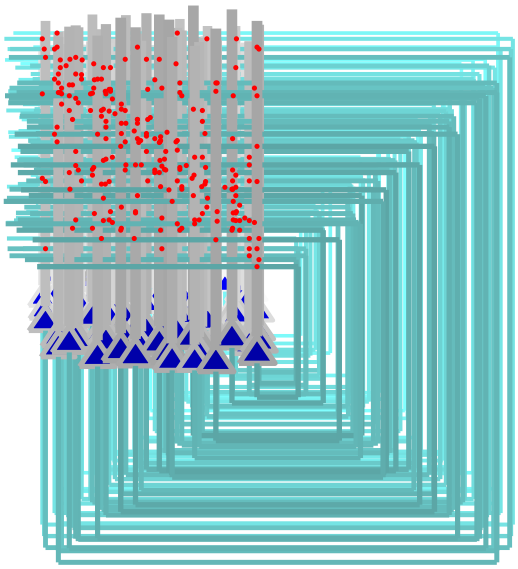

Figure 2.9: CA3 Connectivity

CA3 pyramidal neurons project collaterals to other pyramidal neurons, forming a massive recurrent network.

bility of finding a connection between any two neurons decreases with distance) and even some from the contralateral hippocampus. These recurrent collaterals form synapses on both apical and basal dendrites. The CA3 receives input from the perforant path and a few powerful inputs (about 50) from mossy fibers.

2.3.2.4 CA1 Excitatory Connectivity

The CA1 excitatory connectivity is similar to the CA3. However, the recurrent connections are fewer in number and make connections only on the basal dendrites of pyramidal neurons. The CA1 receives input from the EC and CA3.

2.3.2.5 Inhibitory Interneuron Connectivity

The numerous inhibitory interneurons receive inputs from excitatory neurons in the DG, CA1, CA3, and EC. They can form feedback or feedforward circuits with excitatory neurons and can target somas, dendrites, or both.

2.3.3 Long-Term Synaptic Plasticity

Neurobiology is renowned for its unlimited learning and adaptation, orders of magnitude more malleable than any realized by technology. Long-term synaptic plasticity is widely viewed as the mechanism responsible for learning in neurobiological systems, believed to be both necessary and sufficient, however concrete evidence of sufficiency is lacking [111]. Some other suggested mechanisms including changes in excitability and neurogenesis.

2.3.3.1 Hebb's Rule

In 1949, inspired by neurobiology, Donald Hebb published a causal correlation-based learning rule. He proposed that when one neuron repeatedly contributes to causing another neuron to fire, it should increase its effectiveness. If both neurons are active together, connections among them strengthen. This learning rule has been employed in many neural network models and is simlar to biological synaptic plasticity.

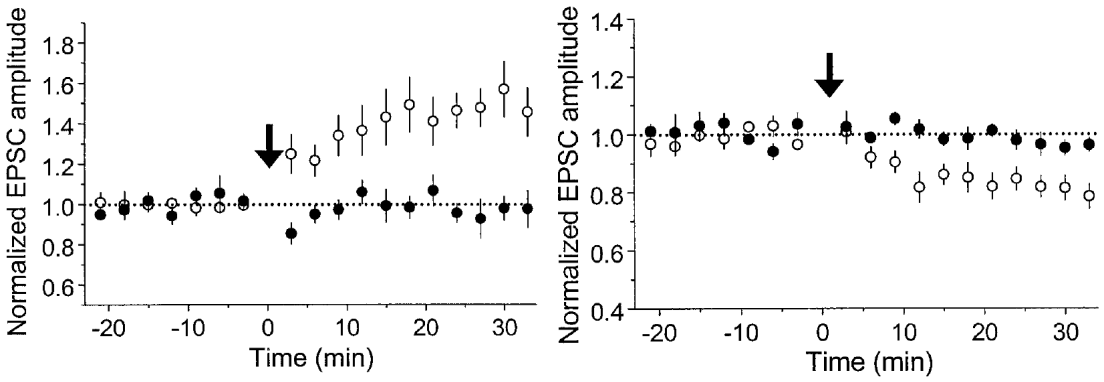

Figure 2.10: Long-Term Potentiation and Depression

Left Potentiation is a lasting increase in excitatory postsynaptic current (EPSC) amplitude. Potentiation is induced by a repetitive positively correlated postsynaptic spiking (○): Presynaptic spike proceeds postsynaptic spike by a short period (5ms) 60 times at 1Hz (arrow). *Right* Depression is a lasting decrease in EPSC amplitude. Depression is induced by a repetitive negatively correlated postsynaptic spiking (○): Presynaptic spike follows postsynaptic spike by a short period (5ms) 60 times at 1Hz (arrow). *Left and right* AP5 blocks potentiation and depression (●). Modified from [12].

2.3.3.2 Long-Term Potentiation and Depression

Long-term potentiation (LTP) and long-term depression (LTD) are long-lasting (more than 30 minutes) increases and decreases in synaptic efficacy, respectively (Figure 2.10). LTP (and later LTD) was first discovered in the hippocampus [14]. The mechanics of LTP and LTD have been extensively studied but are only partially understood. The most studied synapse is the excitatory, glutamatergic Schaffer collateral synapse from CA3 pyramidal neurons to CA1 pyramidal neurons.

Schaffer collateral (and many other types of) LTP and LTD are NMDA dependent[7].

[7]Many other types of LTP and LTD are independent of NMDA, such as one type from the mossy fibers onto CA3 pyramidal neurons.

When NMDA-receptor (NMDAR) gated synaptic channels are blocked by AP5, synapses resist both LTP and LTD [97]. NMDAR channels require both presynaptic input (glutamate) and postsynaptic depolarization to relieve a magnesium block at hyperpolarized potentials. When the block is relieved by depolarization and the channel activated by glutamate, sodium and calcium flow into a synaptic spine head.

Evidence suggests that calcium influx through NMDAR channels determines the sign of synaptic change. Low calcium influx results on average in no change, medium influx in depression, high influx in potentiation [97]. Calcium must flow through NMDAR channels; other sources are ineffective. Calcium triggers chemical cascades that add and remove (fast) AMPA-receptor (AMPAR) gated channels to and from the synapse [118]. A lengthy list of neuromodulators, neuropeptides, and intracellular messenger molecules influence LTP and LTD, through their influence on calcium or by other means. A key molecule in the synaptic spine is CaMKII.

Calcium acts on CaMKII, whose phosphorylation is necessary and possibly sufficient for LTP [118]. CaMKII is ideal for maintaining synaptic strength as it autophosphorylates. Once calcium drives phosphorylation beyond a threshold, CaMKII drives it own activation. Once the spine's calcium level decays back to baseline levels, models suggest CaMKII activation settles at one of two active levels, nearly completely phosphorylated or dephosphorylated [89]. Phosphorylated CaMKII is implicated in realizing LTP by (indirectly) inserting fast excitatory (voltage-independent) AMPAR gated synaptic channels into the spine membrane. In the hippocampus, depressed excitatory synapses include only NMDAR channels, whereas potentiated synapses include AMPAR channels as well [97].

The bistable activation of CaMKII suggests that excitatory synapses are binary; they include or exclude AMPAR channels. Recent observations support such a conclusion that

individual synapses are binary weighted, depending on whether they express AMPA-gated channels [154], suggesting that graded synaptic strengths are merely a function of numerous synaptic contacts, some fraction of which are potentiated.

On the other hand, the influenced of numerous intracellular and extracellular processes, AMPAR and NMDAR channels' number, subunit types, and probability of presynaptic release, suggest that synapses are graded. For example, even if a synapse has inserted AMPAR channels, a reduced probability of presynaptic release would weaken the average effect of the synapse on the neuron. Also, debate continues as to whether LTP and LTD are two sides of the same process, both influencing AMPAR channels, or if they manifest in entirely different ways with one affecting the postsynaptic neuron and other the presynaptic one. Even with (fixed weighted) binary AMPAR expression, synapses are not truly off or on due to their NMDAR channels.

NMDAR channels are effective, even in the absence of AMPAR channels. Although synapses are often modeled as having binary weights, the presence of NMDAR channels at depressed excitatory synapses gives them the opportunity to drive the neuron anytime its membrane is depolarized. In fact, current through NMDAR channels is the dominant source of charge at excitatory synapses: The NMDAR channel's time constant (on the order of 100ms) is much longer than that of the AMPAR channel (a few ms). So even though its peak conductance is usually lower (at potentiated synapses), its influence is significant.

Many induction protocols have been used to elicit LTP and LTD, including long high-frequency (100Hz) bursts for LTP and long low-frequency (5Hz) spikes for LTD. More recently LTP and LTD were found to be sensitive to spike timing.

2.3.3.3 Spike Timing-Dependent Plasticity

Spike timing-dependent plasticity (STDP) increases synaptic strength when presynaptic spikes arrive before postsynaptic spikes (within about 40ms) and decreases it when presynaptic spikes arrive after postsynaptic spikes (within about 40ms) (See Chapter 5). STDP was first discovered in the hippocampus and has since been studied intensively [86, 12, 13].

The mechanics behind STDP, like the intracellular mechanics of LTP and LTD, remain elusive. The way in which biology tracks timing is unclear, however, theory suggest that the backpropagation of somatic action potentials to dendritic synapses influences the calcium influx through their NMDAR channels, driving either LTP or LTD. Without understanding the details of STDP, there are challenges to understanding how it behaves outside of experimental settings, where presynaptic and postsynaptic spikes are tightly controlled. For example, it is unknown how pairings that alternate between favoring LTP and LTD interact; they may compete or remain independent until signals favoring one reaches a threshold.

Although its mechanics are unclear, at least one aspect of STDP's functional significance is obvious. It links together neurons that fire together; it is an activity-dependent causal correlation rule, precisely what Donald Hebb predicted [58]. Further, STDP suggests functional importance, where there was none before, for the rhythms that strongly influence spike timing in the brain in general, and in the hippocampus in particular (see Chapter 5).

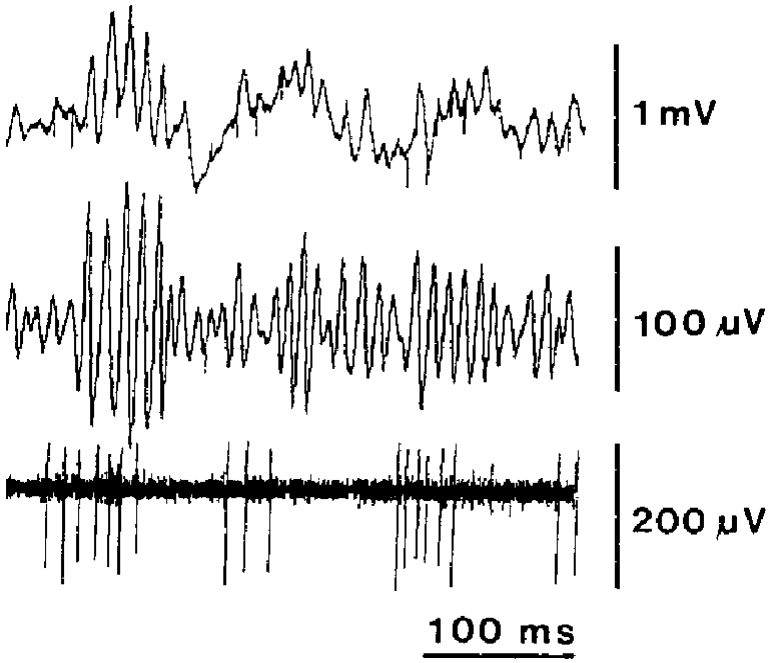

Figure 2.11: Hippocampal Theta and Gamma Activity

During navigation, microelectrodes record theta and gamma activity in rat hippocampus (in this case the dentate gyrus). *Top* In an unfiltered electrode recording, both theta and gamma are visible. *Middle* Gamma band activity is evident when the signal is bandpass filtered 40-150Hz. *Bottom* Single unit spiking activity is modulated both by theta and gamma (bandpass filtered 0.5 to 5.0kHz). From [20].

2.3.4 Rhythmicity

When observing the hippocampal local field potential in an awake, behaving animal, a salient feature is the rhythmicity, dominated by the theta (6-12Hz) and gamma (30-100Hz) rhythms (Figure 2.11) [20].

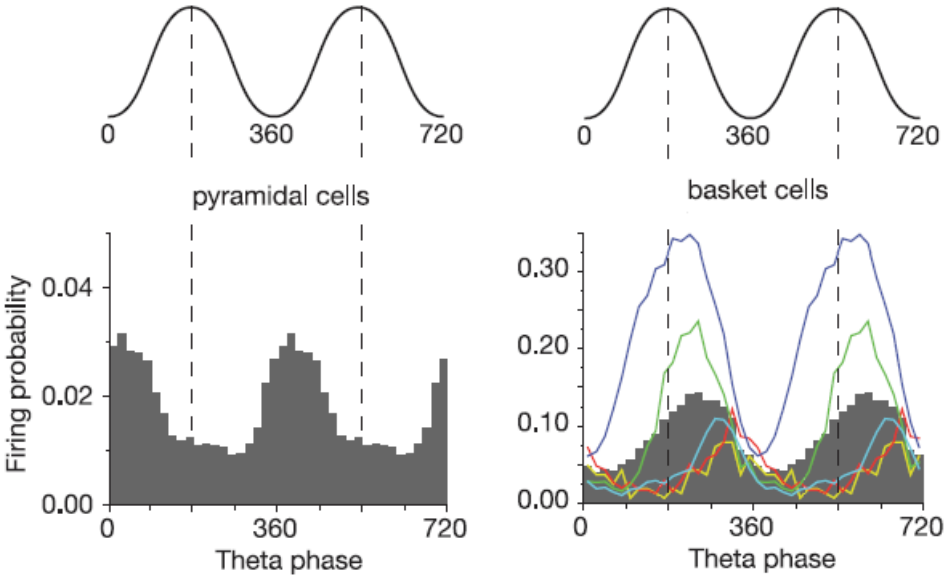

Figure 2.12: Theta Modulates Spiking

Different neuron types preferentially spike at different phases of the theta rhythm. Modified from [80].

2.3.4.1 Theta Rhythm

The theta rhythm is generated by the medial septal nucleus, which excites inhibitory interneurons across the hippocampus, rhythmically hyperpolarizing most neurons' somas, including pyramidal neurons, granule neurons, and mossy neurons. These excitatory neurons typically fire during the low phase of the theta rhythm as inhibition decreases (Figure 2.12). CA3 slices are capable of generating theta, despite the absence of the septal input, when treated pharmacologically [23].

The theta rhythm is theorized to be essential in many types of learning and memory. During the high phase of the rhythm, LTP is more easily induced, during the low phase LTD. In many models, theta provides a form of clocked reset, either starting the network over to process another input [105] or advancing it to the next item in a sequence [145].

Although theta is accepted to have a role in learning, its exact function is a matter of ongoing debate.

2.3.4.2 Gamma Rhythm

Often coactive with the theta rhythm, gamma is ubiquitous in the brain, though its usefulness has only been established in insect olfaction [136]. The gamma rhythm influences pyramidal neuron activity both shaping timing and limiting potency, possibly directing both plasticity and computation [46]. Various theories suggest it may be critical in transferring information between regions (when the gamma rhythms are coherent between them [31, 65]) as well as dividing the theta cycle into slots, where information can be stored temporarily [90]. In the hippocampus, the DG, CA3, and CA1 all demonstrate the capacity to generate gamma. The gamma rhythms between pairs of regions are phase-locked, with DG leading CA3, which leads CA1 [31].

The gamma rhythm is generated locally in each region by populations of mutually inhibitory interneurons, specifically basket neurons, one of the many types of interneurons in the hippocampus [31]. These interneurons use synapses that express $GABA_A$ receptors, a fast (on the order of 5ms) inhibitory neurotransmitter-gated channel population [10]. The degree of synchrony is enhanced by electrical gap junctions among interneurons that act as very fast excitatory connections, however, these gap junctions are unnecessary; when they (or at least some types of them) are blocked or genetically eliminated, synchrony remains but its degree is reduced [22]. For analysis of the generation and modeling of gamma, see Chapter 4.

2.4 Components of Sequence Memory

Vast work has gone into developing and characterizing abstract neural networks. In this section, I focus on the basic structure of two associative memory network and their combinations. I omit many details of abstract neural networks, instead narrowing my scope to neural network structures that relate directly to the biological associative memory.

Neural networks excel at associative memory, that is they link items, including features, objects, and events. Associations can be used to learn and recall sequences of events. There are two fundamental components of neural networks that implement sequence memory: autoassociation and heteroassociation. Autoassociative memory links components of a pattern to each other; these links can complete or correct a stored pattern when only part (or a corrupted version) of the pattern is present. Heteroassociative memory links different patterns together, such that a different pattern than the input pattern is output. Sequence memory uses heteroassociation to link patterns together.

2.4.1 Autoassociation

Autoassociative neural networks store and recall patterns, completing or correcting corrupt patterns, a form of content-addressable memory. For simplicity, I consider neural networks with binary-valued neuron inputs and outputs (neurons are active or inactive, not graded). I divide autoassociative neural networks into two groups: Willshaw networks [134, 153], which have only feedforward connections, and Hopfield networks [62], which have feedback connections as well.

The Hopfield network uses all-to-all feedback (recurrent connections) to store and recall

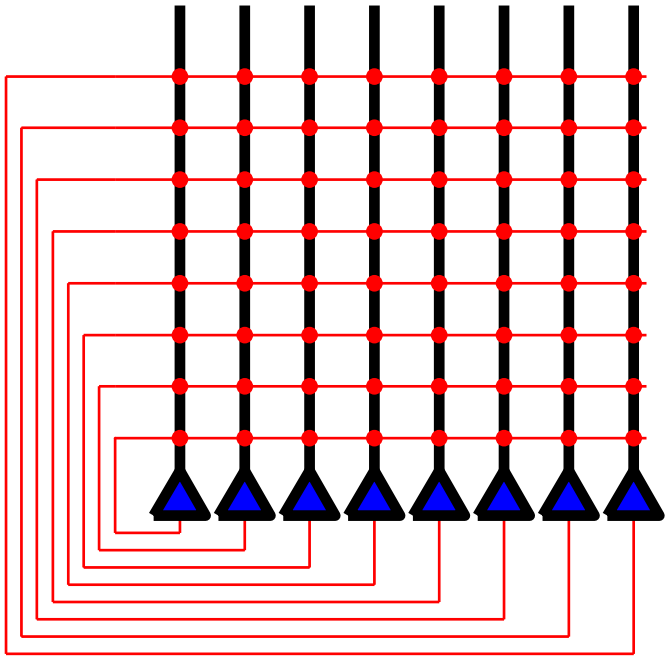

Figure 2.13: The Hopfield Network

The (eight-neuron) Hopfield network is characterized by all-to-all recurrent connectivity and employs a hebbian learning rule that enables it to store and to recall (eight-bit) binary patterns.

patterns (Figure 2.13) [62]. With its structure similar to that of the CA3 region. It stores patterns with a type of hebbian rule that increases recurrent synaptic weights between pairs of neurons that are correlated in a pattern and decreases weights between anticorrelated pairs. This hebbian pattern storage results in a symmetric weight matrix, in which neurons that are correlated or anticorrelated excite or inhibit each other, respectively. If neuron states are evaluated asynchronously and randomly, the Hopfield network is guaranteed to converge to a stable state (synchronous networks can also be used). The lack of such a guarantee had previously contributed to the limited use of recurrent networks.

One of the tremendous strengths of the Hopfield network is its ability to iteratively approach a stored pattern [62]. When a corrupt stored pattern is activated in the network,

the driven neurons that were a part of the original pattern activate the missing neurons and suppress additional neurons, which after several network iterations (asynchronous or synchronous) results in recall of the original pattern (or a close network state), the *attractor*. The recurrent architecture enables this iterative memory recall, but it also limits the number of memories the network can store to about 0.15 times the number of neurons in the network: At this level, only half of network states settle near the correct stable states (with less than 5% error).

The Willshaw network uses a feedforward architecture to store and recall autoassociative patterns (Figure 2.14). Its structure is similar to that found in the CA1 region. Each element of the input (vector) connects to every neuron. Connection weights are determined with a correlation rule similar to the one used by Hopfield [62, 153, 119]. Each neuron determines its binary value by summing and comparing its inputs to a threshold. The feedforward architecture allows the network to store more patterns than the Hopfield network, but the lack of iteration renders the network less effective at restoring corrupt patterns[8]. Combining the recurrent connections form the Hopfield network with the input fan out and neuron fanin of the Willshaw network into a single network results in a further improved autoassociative memory. The input provides a degree of error correction and the recurrent connections even more. The Willshaw network can be used to perform heteroassociation.

2.4.2 Heteroassociation

The absence of recurrent connections in the Willshaw network renders patterns unable to restore themselves actively. Without active restoration, the Willshaw network does not

[8]The number of patterns stored can exceed the number of neurons in the network and does so when information storage reaches its maximum (in terms of bits per synapse); however, in this regime recalled patterns are highly corrupt [119].

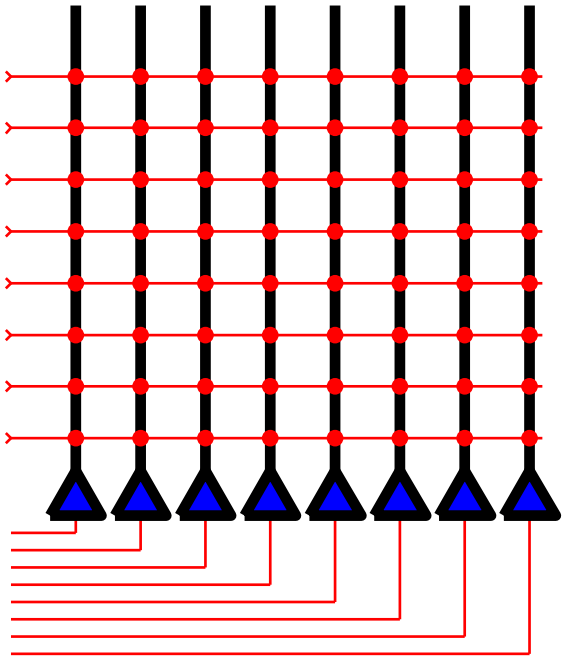

Figure 2.14: The Willshaw Network

The (eight-neuron) Willshaw network is characterized by feedforward connectivity and employs a hebbian learning rule that enables it to associate input patterns with (eight-bit) binary output patterns.

favor autoassociation over heteroassociation; it can just as easily link input pattern A to output pattern A as to output pattern B. Therefore, one can use its feedforward architecture to store and recall heteroassociative patterns. As in the autoassociative case, the network uses a correlation rule, however, each input pattern is correlated with the desired output rather than itself.

The Hopfield network can also perform heteroassociation with a minor modification, asymmetrically linking of two stored attractor patterns together. Neurons in pattern A drive each other and the neurons in pattern B, using the same correlation rule, but pattern B's neurons do not reciprocate (breaking the symmetry present in the purely autoassociative network). Using these connections the Hopfield network can recall a sequence of up to four

patterns in length, however, such sequences often fail to follow the correct trajectory [62].

2.4.3 Abstract Sequence Memory

The Hopfield network can be modified to store and recall long sequences of patterns reliably by adding a second delayed set of recurrent synapses (Figure 2.15) [81]. Like the original Hopfield network, one set of synapses stores autoassociative patterns, but the other set of synapses, activated after a long "axonal" delay, provide asymmetric links between sequential patterns, realizing heteroassociation. When the network receives an input, corresponding to a corrupted version of a stored pattern, the autoassociative (fast) synapses restore the original pattern. The network stays in this state until the delayed synapses are activated at which time these heteroassociative synapses overpower the network's state, driving its transition into the next pattern in the sequence. The process repeats, replaying a whole sequence of stored patterns. One advantage of this network is that it combines the noteworthy completion capacity of the Hopfield network with the ability to advance to the next sequence reliably.

2.5 Models of the Hippocampus

One of the first models of the hippocampus as a memory system was proposed by David Marr [99]. He hypothesized that all archicortex, including the hippocampus, was structured to perform only one function, the temporary storage of patterns. These patterns represented sensory input and could be recalled associatively (with partial input). Patterns that are determined to be unimportant would fade with time, whereas patterns determined to be

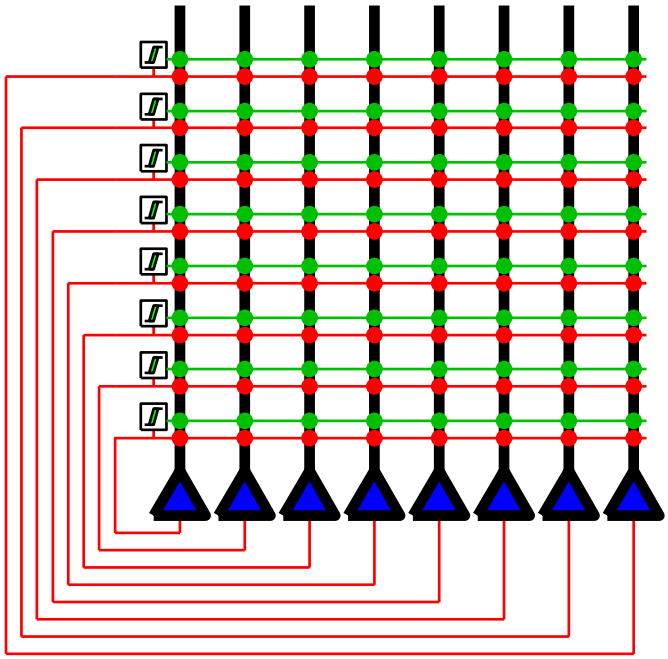

Figure 2.15: Abstract Sequence Memory Network

The (eight-neuron) Kleinfeld network employs two populations of recurrent synapses, one fast (red) and one delayed (green). The fast synapses allow the network to store and to recall patterns; the delayed synapses allow the network to link those patterns together, to store and to recall sequences of patterns.

important (episodes) could be copied or moved into neocortex. In this section, I briefly cover a few important models of autoassociative and sequence memory.

2.5.1 Models of Hippocampal Autoassociation

Modeling autoassociation in the hippocampus began with the Hopfield network [62]. Although it did not claim to model the hippocampus (or any specific brain region), the Hopfield network captured several key features of the CA3 region, including massive recurrent connectivity and hebbian learning. Other aspects were inconsistent with CA3, such as

all-to-all connectivity, a symmetric synaptic weight matrix, a lack of distinction between excitatory and inhibitory synapses (and neurons), simple (nonspiking) neurons, and an average activity level of 50 percent of the population (compared to 2.5 percent in CA3) [9]. However, all of these differences with the biological CA3 have since been addressed without compromising the concept of the attractor network.

The principles of the Hopfield network have been applied to a biologically realistic model of CA3 (Figure 2.16) [105]. This model uses reconstructed multicompartment spiking pyramidal neuron and inhibitory interneuron models. The network employs a similar hebbian learning rule to the one used in the original Hopfield network. The model employs aspects of the biology absent from the Hopfield network, notably theta and gamma rhythms (from inhibitory interneurons): Theta frequency inhibition resets the network, priming it for an input pattern, and the gamma frequency inhibition acts as a clock, providing a framework for iterative pattern recall. The network typically requires three or four gamma cycles to recall a pattern; however, with stronger recurrent synapses it may be able to recall patterns faster within one gamma cycle.

2.5.2 Models of Hippocampal Sequence Memory

There are two types of hippocampal sequence memory models (so far), distinguished by use of explicit or implicit heteroassociation. Explicit heteroassociation directly links neurons in each pattern to neurons in the subsequent pattern, storing the series. For example in sequence ABC, A links to B, which links to C. Implicit heteroassociation uses two populations of neurons: One population of *context* neurons forms (usually random) links from neurons in one pattern to the next as in explicit heteroassociation, storing a series such as 123. An arbitrary input sequence (ABC) of activity in the *input* population is stored by

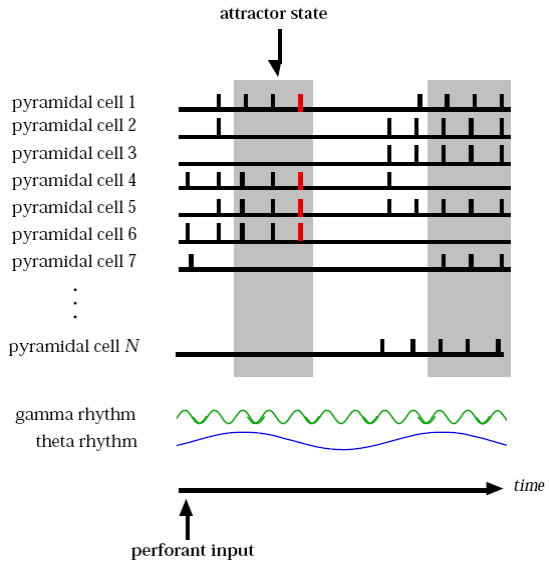

Figure 2.16: Biologically Realistic Attractor Network

The model uses detailed spiking neuron models to implement pattern storage and recall like the Hopfield network. It uses theta and gamma rhythms to reset the network and clock the neurons, from [105].

bidirectional linking each input pattern to a pattern in the series, storing the input sequence. For example, each pair, A and 1, B and 2, and C and 3, forms a bidirectional link. If we activate A, it activates 1; 1 drives B by activating 2, which drives C by activating 3, replaying ABC.

One model of hippocampal sequence memory is Levy's model (Figure 2.17) [85]. Levy and colleagues aimed to simulate the simplest possible biologically plausible model of sequence memory in CA3. They use simple (McCullock-Pitts) neurons and a simple (timing-based hebbian) learning rule [100]. The naive network has a random connectivity matrix (10 percent potentiation probability between pairs of neurons). When taught a sequence (5 trials), neurons in the sequence connect to coactive neurons. Levy refers to neurons that spike but were not driven as a part of the sequence as "context" neurons. The behavior of these context neuron changes over the training: Initially, these spike once or twice, but

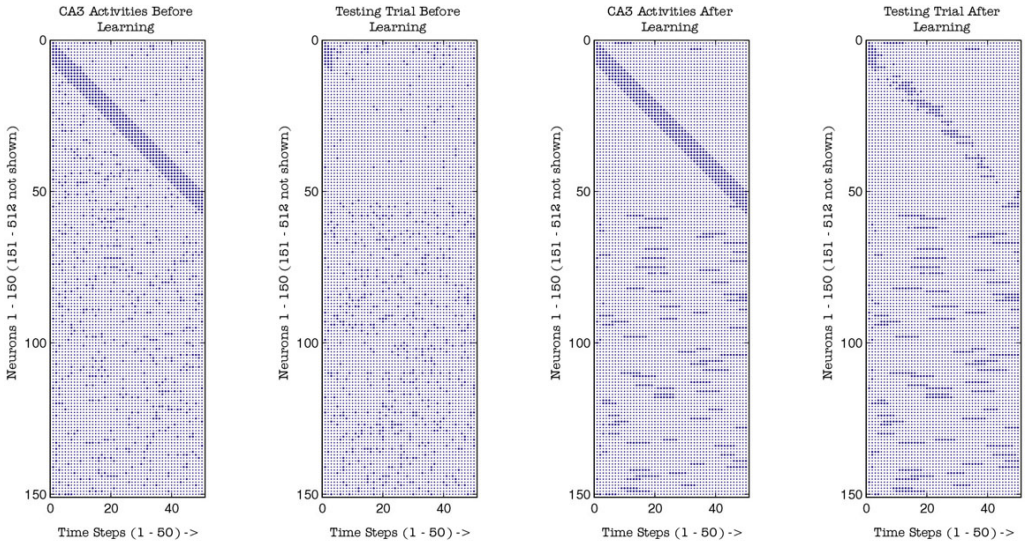

Figure 2.17: Simple Sequence Memory

Levy's network store sequences by linking background bursting neurons together and associates the input sequence with them. Modified from [85].

after learning many spike five to ten times. Expanding the period over which these neurons fire enables them to link to each other heteroassociatively and link to the training sequence autoassociatively. So the links between context neurons stores timing information and the links to and from the training sequence stores the patterns. This network takes five trials to learn a sequence. It is not clear that it could successfully store patterns with one-shot learning. Further, unlike the biological hippocampus, this model does not recall sequence faster than real-time.

Another much more biological model uses the same principles as Levy's model: context neurons link together heteroassociatively and link to the stored pattern autoassociatively (using a similar learning rule) [145]. The authors simulate simple multicompartment spiking neurons to implement a model CA3 network. In addition to storing and recalling sequences, the network recalls sequences faster than real-time and expresses phase preces-

sion, similar to the biological hippocampus (observed in CA1).

Both of these models make several assumptions about encoding information in the hippocampus. They both assume learning requires several trials, which may conflict with evidence for one-shot learning in hippocampal dependent tasks [114] (cortex could provide a buffer [76]). They assume patterns (in sequences) inherently have great overlap (75 percent in [145]), which is intuitively consistent with navigation as nearby locations tend to be similar. This may be inconsistent with other types of episodic memory in which sequential memories can be quite different (although may have similar contexts). Models that directly use heteroassociation between patterns in a sequence make a different set of assumptions.

Similar to the Hopfield network in autoassociative memory, Kleinfeld's sequence memory network was one of the first models related to sequence memory in the hippocampus [81]. Although Kleinfeld did not claim to model hippocampus (but instead to model motor sequences), his model employs features found in the hippocampus, especially the CA3, including recurrent connectivity and hebbian learning. Other aspects were inconsistent such as all-to-all connectivity and two distinct synaptic populations, one instantaneous and the other delayed. Despite these differences with the biological hippocampus, Kleinfeld's network provides insight into how the hippocampus could store sequences robustly by mixing autoassociative and heteroassociation. One group of researchers led by John Lisman used these insights in developing models of hippocampal sequence memory.

Jensen and Lisman realized that storing sequences of events that occurred much slower (seconds to hours) than neural times scales (milliseconds to hundreds of milliseconds) would benefit from a short-term memory buffer, in which the last several events (patterns) could be held [72, 73, 74, 75]. They proposed that neocortex (regions such as PFC or EC) implemented this episodic buffering of sequences of events. By buffering the last several

events in a short-term memory buffer, the sequence of patterns could be replayed (repeatedly) for the hippocampus to store at a neural time scale, also providing many opportunities for learning, facilitating one-shot learning.

Jensen and Lisman proposed that the theta and gamma rhythms provided an ideal mechanism to buffer these patterns: The theta rhythm corresponds to the repeating buffer of patterns, available to the system about every 120ms, and the gamma rhythm divides the theta period into about seven slots (during the low phase), which could hold the buffered patterns, multiplexed in time[9]. The seven-item buffer is consistent with human short-term memory experiments that showed most people can remember a list of about seven items [135]. The authors implemented short-term memory by using an afterdepolarization: A model neuron tends to spike again one theta cycle later due to a depolarizing ion-channel populations. This mechanism succeeded in demonstrating time-multiplexed short-term memory, although it proved sensitive to noise.

Jensen and Lisman modeled the hippocampus as storing these sequences of patterns in CA3 (Figure 2.18) [72, 73, 74, 75]. Similar to the Kleinfeld model, the authors used both autoassociative and heteroassociative synapse populations, composed of recurrent synapse populations in neocortex and hippocampus, respectively. Their model stored patterns and recalled them faster than real-time as well as realizing phase precession. Later, Lisman updated the model to include the DG, which acted as a local autoassociative network instead of relying on neocortex [90]. In this new model, neocortex provided short-term memory, DG autoassociation, and CA3 heteroassociation.

[9]Some models of PFC suggest that seven patterns could be held in seven different networks within PFC instead of without multiplexing in time.

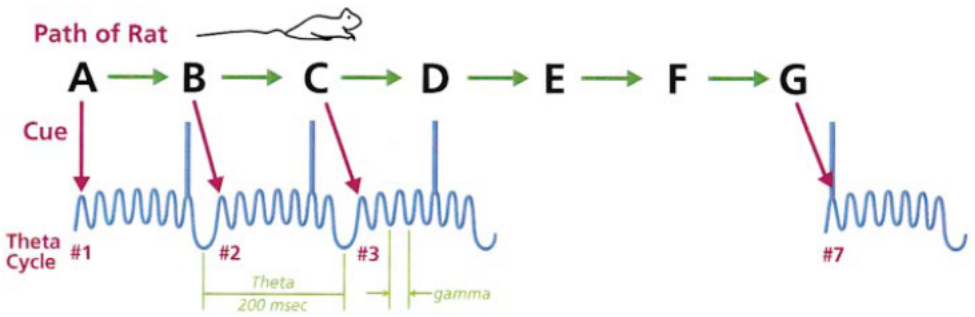

Figure 2.18: Gamma Slots Store Sequence Information

Patterns, represented by place cell spikes, are stored in the subsequent gamma cycles within a theta cycle. The first slot holds the current location. Upcoming place cells spike in earlier and earlier gamma cycles as the rat moves through the environment. Modified from [90].

2.6 Dual Autoassociation Model of Sequence Memory

My model employs features from previous models but also suggests some new features. Like previous ones, the model employs the gamma rhythm to divide the theta rhythm into slots that can hold about seven sequential patterns [73]; however, unlike others, my model explicitly focuses on the role of the gamma rhythm in coordinating timing and therefore plasticity within and between the DG and CA3, both in storing and recalling patterns and sequences of patterns.

Lisman's model provides coherent roles for DG and CA3 in sequence memory, but it has a potential problem. The model relies on the CA3 region to perform heteroassociation; however, using a single network for heteroassociation is difficult. When one pattern is activated in CA3, its neurons drive the subsequent pattern. Such drive causes the subsequent pattern to fire with a short delay (a few milliseconds), which is why Kleinfeld's network requires delayed synapses for heteroassociation. Without a long delay, sequen-

tial patterns in CA3 merge into one pattern. Of course, this problem can be solved with precisely controlled synaptic strengths and neuronal excitabilities, which could be tuned to activate the next sequential pattern one gamma period later, but this is not plausible. More likely, CA3 behaves as an autoassociative network, which does not require long synaptic delay or precisely control of network parameters[10]. Leaving two obvious possibilities for heteroassociation: the connections among the DG's two populations of neurons, which may spike out of phase, preventing patterns from merging, or the connections from CA3 to DG, which provide an effective delay because the gamma inhibition in each of these networks is out of phase (about 60 degrees) [31].

2.6.1 Autoassociative and Heteroassociative Memory Regions

We, independently, developed a model in which recurrent connections within both DG and CA3 are autoassociative and reciprocal connections from DG to CA3 and from CA3 to DG are autoassociative and heteroassociative, respectively (Figure 2.19). The autoassociative synapses transmit the pattern from DG to CA3 unchanged; the heteroassociative synapses link the pattern in the CA3 to the subsequent pattern in the DG, advancing the network to the next item in the sequence (Figure 2.20). A recent study of an abstract neural network contrasted the sequence memory performance of two network architectures [83]: one architecture is identical to my model with two autoassociative networks linked by one set of heteroassociative connections; the other architecture was similar to previous models, with one autoassociative network and one heteroassociative network reciprocally connected. My model's architecture performed better; it was less sensitive to noise and more accurate in recalling sequences.

[10]Independently, Lisman and colleagues swapped the roles of DG and CA3, with DG performing heteroassociation and CA3 autoassociation [91].

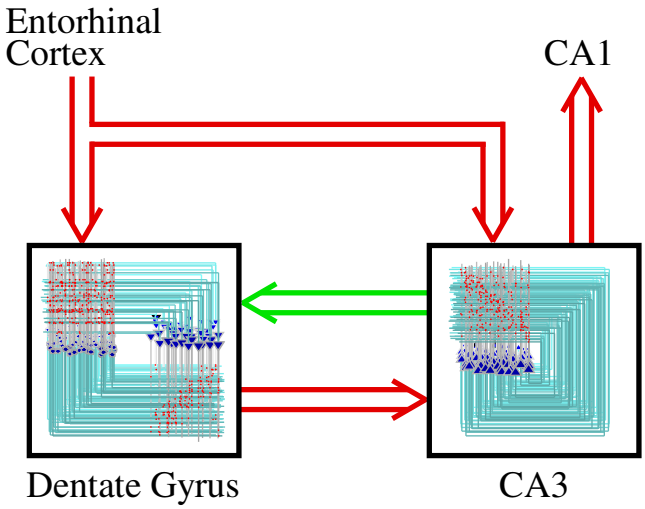

Figure 2.19: Dual Autoassociative Networks

In this model, sequence memory is realized by two autoassociative network, linked together by a single heteroassociative connection.

2.6.2 Sequence Memory Storage

My model explains how the hippocampus can store new sequences of patterns. Learning sequences relies on timing the activity of various regions to ensure sequences are not corrupted. When an animal is active, highly processed sensory information is transmitted from neocortex to hippocampus. I assume that neocortex buffers important sensory information into sequences of patterns [72]. The buffered sequence is transmitted across many theta cycles from EC to DG and CA3.

The neocortex (e.g. EC) may buffer short-term memory with multistable neurons. Rather than replaying memories in theta-gamma cycles, the EC neurons may just spike at a given constant rate. EC neurons are multistable; they can maintain a given spike rate without input [37]. Brief inputs can raise or lower the rate. Several groups of EC neurons all driving corresponding groups in the hippocampus will result in each group firing at a

different phase depending on its input rate: High input rates cause neurons to spike early in the theta phase; low rates cause neurons to spike late in the theta phase.

In DG, the recurrent granule and mossy neuron network stores the pattern autoassociatively and excites the pattern in CA3. The first pattern in the sequence from EC drives both granule and mossy neurons in the DG, causing spiking in a group of neurons. After several theta cycles, connections from mossy neurons to coactive granule neurons potentiate[11]. Granule neurons drive CA3 pyramidal neurons, communicating the pattern.

In CA3, the recurrent pyramidal neuron network stores the pattern autoassociatively and drives DG to create a heteroassociative link to the next pattern in the sequence. Each granule neuron excites about ten CA3 pyramidal neurons via powerful mossy fiber boutons, activating many of them but only after some delay. Connections among these stimulated pyramidal neurons potentiate when repeatedly activated over several theta cycles, autoassociatively storing the pattern in CA3. The active pyramidal neurons project axons back to the mossy neurons of DG, heteroassociatively storing a link in the sequence.

The activation delay occurs because gamma rhythms in DG and CA3 are out of phase. CA3 lags by about sixty degrees (4.2ms at 40Hz), and CA3 pyramidal neurons (on average) spike later in phase than (unidentified) DG neurons by several tens of degrees [31]. This lag is critical in my model: It prevents the synapses from CA3 pyramidal neurons that drive DG mossy neurons from potentiating while both DG and CA3 represent the same pattern, ensuring that the link from CA3 to DG is purely heteroassociative.

The synapses CA3 pyramidal neurons make onto DG mossy neurons satisfy the pre-before-post activation rule of STDP. When the pyramidal neurons fire, mossy and granule neurons in the DG are inactive. Several milliseconds later, the EC transmits the next

[11] Connections from granule neurons to mossy neurons may potentiate as well, but are fixed in the model.

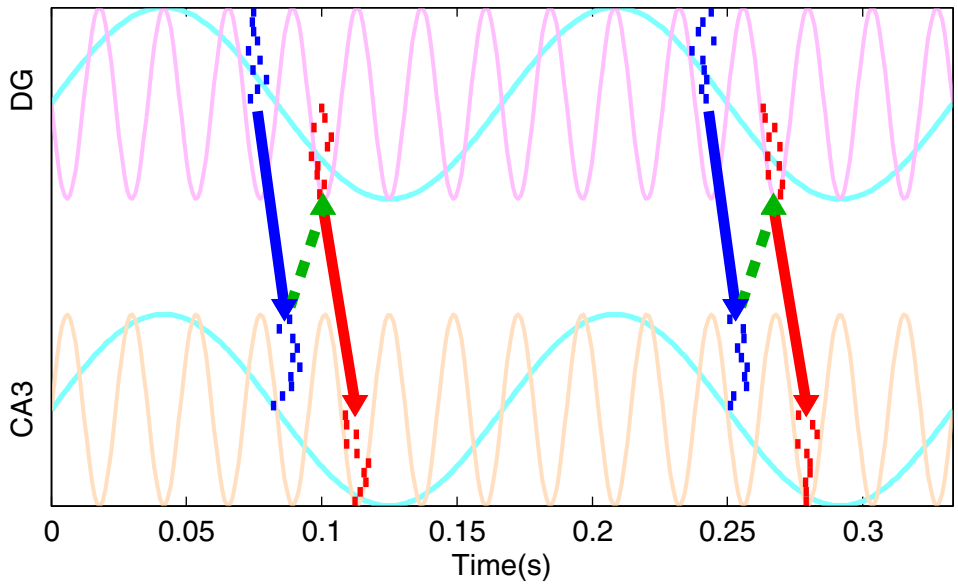

Figure 2.20: Two Network Sequence Memory

In this model, the pattern one (blue) in the dentate gyrus drives the CA3 region, whose gamma is out of phase by 180 degrees (exaggerated from 60 degrees for clarity). When the CA3 gamma decreases, CA3 pyramidal neurons, representing pattern one (blue) spike, driving the dentate gyrus just before the pattern two (red) is activated. This is the ideal protocol for STDP, which links pattern one to pattern two, storing the sequence.

(buffered) pattern in the sequence, which activates mossy and granule neurons. This cortical sequence of activation results in the potentiation of the synapses mossy neurons receive from CA3 (after many theta cycles). In my model, STDP links each pattern to the subsequent pattern in the sequence. This process can continue for long sequences, linking each pattern to the next.

2.6.3 Sequence Memory Recall

Similar to storing sequences, recalling sequences of patterns relies on timing the activity of DG and CA3. When the EC transmits the first element of a sequence to the hippocampus, the latter should correct any corruption in the pattern and recall subsequent patterns in the sequence.

In DG, the granule/mossy neuron network autoassociatively corrects the input pattern. The input pattern excites both granule and mossy neurons, causing spiking. Because the pattern has previously been stored, many connections among the active neurons are potentiated. These strong connections drive neurons in the pattern that the EC did not activate, restoring the pattern. The corrected pattern is transmitted to CA3 by granule neurons.

In CA3, the pyramidal neuron network corrects the pattern further with its potentiated recurrent synapses and activates the next pattern in DG via the heteroassociative link. Again, it is important that CA3 and DG activities are out of phase. If CA3 is synchronous with the DG, it would drive neurons in the next pattern immediately, even while the current pattern is still active. These coactive patterns' neurons would make potentiated connections, merging the two patterns into one meaningless one. Out-of-phase activity avoids this catastrophe.

2.6.4 The Role of the CA1

Although I focus my efforts on the DG and CA3 regions, no hippocampal model is complete without a role for the CA1 region. The CA1 is necessary for the DG and CA3 to function as it is their primary output to the rest of the brain. Further, animals with DG

and CA3 damage perform many navigational tasks with minimal deficit and include normal place cells [113]. Lisman assigned the CA1 region the role of detecting disagreement between the EC and CA3 [90], but clearly it has other functions. I assign the CA1 the role of mutlisensory integration. Evidence suggest it is able to merge incoming information to make place cells sensitive to context, beyond that of the entorhinal cortex [113].

2.7 Neuromorphic CA3

I have presented my model of sequence memory in the hippocampus. The next step is to build a neuromorphic chip that can store and recall sequences of patterns. In the rest of this dissertation, I present my CA3. It uses silicon neurons to generate the gamma rhythm, to encode patterns at different phases of theta rhythm, and to store and recall autoassociative patterns using synapses that express STDP.

Chapter 3

A Neuromorphic Neuron

In Chapter 2, I described my model of sequence memory in the hippocampal formation. In this chapter, I describe the design and characterization of the basic element of my hippocampal model, the silicon neuron. I use the conductance-based silicon neuron to realize pyramidal neurons and inhibitory interneurons.

In Section 3.1, I introduce my strategy for designing neuromorphic systems. In Section 3.2, I review integrator and conductance-based neuron models and introduce a log-domain low-pass filter suitable for building silicon neurons' somas and synapses. In Section 3.3, I describe my conductance-based neuron circuit, including its soma and synapses. In Section 3.4, I characterize the neuron circuit, including the soma's frequency-current response, the synapse's rise- and fall-times, and the neuron's phase-response curve. In Section 3.5, I discuss the neuron's role in the hippocampal model and other systems.

3.1 Neuromorphic Design Methodology: In the Valley of Death

Like all engineers, neuromorphic engineers encounter design tradeoffs. Neuromorphic engineers employ mixed analog and digital silicon integrated circuits as their medium. Silicon has many desirable properties, such as real-time performance and dense integration; however, it also has limitations, such as heterogeneous transistors and constrained area. The major issue specific to neuromorphic engineering is the influence of circuit complexity and transistor sizing on circuit heterogeneity and on the number of circuits that fit in a fixed silicon area, the Neuromorphic Tradeoff.

Increasing circuit complexity increases the number of transistors in a circuit, which increases the circuit size, thereby reducing the number of circuits in a fixed area. In addition, increasing circuit complexity increases circuit heterogeneity, since circuit variance is the sum of the variances of all transistors in the circuit. To compensate for heterogeneity, engineers can increase transistor sizes, but the cost is a further reduction in the number of circuits one can implement on a chip. The strategy I have adopted to manage the tradeoff between circuit complexity and transistor sizing and their influences on circuit heterogeneity and density is to simplify and to shrink.

Simplifying circuits is critical to creating functional neuromorphic systems: It reduces both circuit size and heterogeneity. Often, circuits can be simplified without reducing their functionality by eliminating structures, such as current mirrors, that do not perform computation. In addition, circuits can be simplified by sharing variables such as inhibitory currents, which are similar or identical between adjacent neurons. Eliminating unnecessary subcircuits is critical in creating functional chips. It is tempting to add numerous

biological details to every neuron, but this is a sure path to a nonfunctional system, since every biological detail increases variability among neurons. Adding more details pushes neuromorphic systems into the region of operation that has come to be called the Valley of Death (VOD).

VOD is a conceptual region where neurons have become too complex to work together, since their variance is high, but not so complex that they can adapt away their differences (Figure 3.1). VOD is characterized by complex neurons that work individually but fail as a system, because their variability is too great. One reaches this region by adding additional functionality and complexity to simple neurons, which already work, decreasing system performance. Only by further increasing complexity to the level where neurons adapt away their differences can engineers escape VOD.

Including every detail, often the strategy in computational models, is further undesirable because the added complexity makes it difficult to determine why the system did or did not function as expected. The rule I use is to keep a subcircuit (such as an additional model of a voltage-gated channel population) if and only if it is absolutely necessary for the model to work or it is an aspect whose influence on the model's performance I plan to evaluate through experiment. Therefore, I implement the least complex circuit that still achieves my goals. The disadvantages of this strategy are that one must have a detailed understanding of how the model works before constructing it, and if a critical element is overlooked, the model will not work. In addition to simplifying circuits, I manage the Neuromorphic Tradeoff by shrinking circuits.

Shrinking circuits is accomplished by reducing transistor sizing and is essential to creating functional neuromorphic systems. Shrinking circuits allows for more circuits, more neurons, to implemented in a fixed silicon area. Often, shear numbers can compensate for

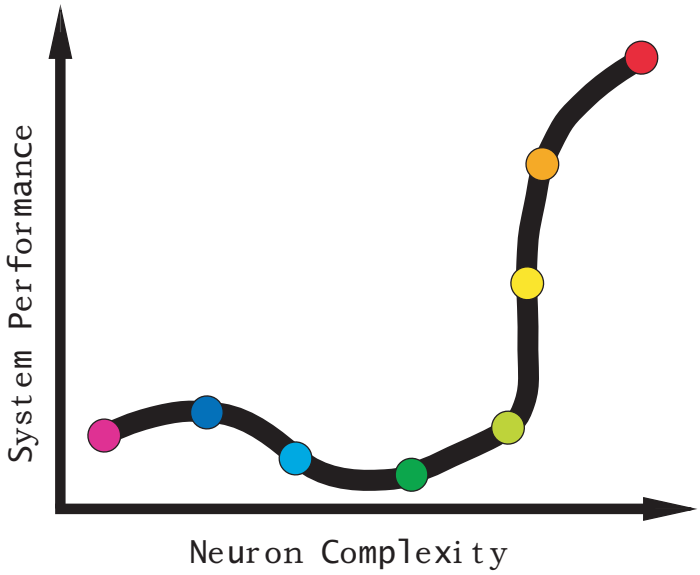

Figure 3.1: The Valley of Death

The Valley of Death is a conceptual region of circuit complexity where in adding functionality (complexity) to neurons degrades the system performance (which is assumed to be inversely related to variance). Dots represent different neuron implementations with cool colors representing simple neurons and warm colors representing biological neurons. Neurons in the middle region, the valley, neither cool nor warm, perform poorly. They are complex enough to increase variance among the populations but not complex enough to adapt.

'bad' neurons in that their behavior is averaged out. The primary disadvantage of shrinking transistors and circuits is the increase in heterogeneity. A Transistor's standard deviation in voltage offset is inversely proportional to the square root of its area (currents are log-normally distributed) [125]. Therefore, to halve a transistor's standard deviation, one must double both its length and width. I have quadrupled the area and only halved the standard deviation (in voltage offset).

Clearly, continually increasing transistor size is a loosing battle, implying that neuro-

morphic circuits must function despite significant variability. A good compromise is to make transistor dimensions several times (three to five) larger than the minimum size as this often fits well, since in silicon as well as biology wiring is typically the limiting factor in circuit size [102]. Further increasing transistor size without a specific reason is usually deleterious to system performance as the number of neurons drops. However, it is naive to assume every transistor should be the same size.

Optimally shrinking circuits requires advanced techniques to allocate area to each transistor. It is obvious that digital transistors should be smaller than analog transistors, usually near minimum size. However, sizing analog transistors is much more difficult and usually requires analysis or simulation. Many circuits are analytically tractable and the effects of transistor heterogeneity can be exactly determined (Appendix B). In these circuits, selecting which transistors are larger or smaller should be based on their effect on the overall circuit variability. Other circuits are more complex, or do not lend themselves to analysis, in which case the circuits can be simulated many times with different sizes (and corresponding variances) for each transistor to determine what distribution of areas minimizes circuit variability.

3.2 Conductance-Based Neuron Design

In the last section, I presented the primary circuit design issue facing neuromorphic engineers as well as my strategy to manage the Neuromorphic Tradeoff: reduce complexity as much as possible while still achieving my design goals. Never include a subcircuit (neural element) simply because it is there (in neurobiology). In this section, I will apply this strategy to neuron circuit design. There are two major types of neuron circuits found in neu-

romorphic implementations, unfortunately, both are referred to as "leaky integrate-and-fire neurons" (LIF).

The first LIF neuron circuit is an integrator, implemented with a capacitor (discharged by a constant leak current), with two input currents, one excitatory and one inhibitory, that add and subtract charge from the capacitor, respectively [101, 68]. When the sum of the leak and inhibitory currents exceeds the excitatory input current, the capacitor voltage decays to zero potential. When the excitatory input current exceeds the sum of the leak and inhibitory currents, the capacitor voltage charges up to the neuron threshold, at which point the neuron spikes (via an axon-hillock circuit) and the capacitor voltage is reset to zero potential. I refer to this LIF neuron as the integrator type.

The second LIF neuron circuit is conductance based, equivalent to a resistor-capacitor circuit with a variable resistor. This circuit low-pass filters an excitatory input current. Inhibitory input current influences the circuit by decreasing the effective resistance of the circuit, reducing both the effectiveness of excitatory current and the circuit time constant. If, nevertheless, the excitatory input current manages to drive the membrane potential to threshold, the neuron spikes (again via an axon-hillock circuit) and the membrane potential is reset to zero. I refer to this LIF neuron as the conductance type.

A cursory inspection of the distinctions between the two LIF neuron types reveals little difference. Both neurons spike when their excitatory drive overcomes their inhibitory drive. Both neurons increase their frequency with increased excitation. Both neurons reduce their firing rates subtractively rather than divisively as the (slow) inhibitory drive increases [61]. However, this is where the similarities end.

These two types of LIF neurons have significant differences in behavior, but rather

than list each one, I focus on one major difference that has a significant role in synchrony: the difference in phase response, which quantifies how a neuron's temporal response to transient input varies with the timing at which it is applied. Phase response captures the effect inhibitory (or excitatory) synaptic input has in extending (or contracting) the period of a neuron. I plot the increase in period versus the phase at which the neuron is inhibited to create a phase-response curve (PRC). Although the work of Holt and Koch might lead one to believe that both LIF types, the integrator type and the conductance type, respond identically to inhibition [61], that both change their firing rate similarly in response to very slow inhibition, their responses to fast inhibition, as often occurs in neurobiology (with a decay constants from 1 to 5ms for GABA) are distinct [10, 52]. I solved analytically for the PRC for both LIF types (See Appendix A) (Figure 3.2). A structured response is important for creating synchrony, whereas an unstructured response better supports asynchrony.

The integrator-type neuron's period is increased the same for inhibition at any phase, whereas the conductance-type neuron's period increases more for inhibition that arrives later. In a recurrent network, the conductance-type neuron changes its period depending on the phase differences between it and its neighboring neurons, whereas the integrator-type neuron changes its period based only on how many inhibitory inputs it receives and not based on the phase at which the inputs arrive. This neuron has no mechanism to change its behavior to match its neighbors. Therefore, since one of my goals is to model synchrony (Chapter 4) in silicon, I implement conductance-type LIF neurons instead of integrator-type LIF neurons.

The integrator-type neuron's period is decreased the same for excitation at any phase (unless the excitation is strong enough to evoke a spike immediately). On the other hand, the conductance-type neuron's period decreases more for excitation that arrives later in the period, because the effect of the excitation early in the period decays away, before the neu-

ron reaches its spike threshold. The conductance-type neuron changes its period depending on the phase at which its input arrives, whereas the integrator-type neuron changes its period based only on how many excitatory inputs it receives. Because timing is important in my model of the hippocampus, I implement conductance-type LIF neurons instead of integrator-type LIF neurons.

I implement conductance-based neurons (CBNs) in current mode, using subthreshold (MOSFET) transistors, which express an exponential (drain) current to (gate) voltage relationship (for an N-type MOSFET, NMOS):

$$I_{\mathrm{D}} = \mathrm{I}_0 e^{(\kappa V_{\mathrm{G}} - V_{\mathrm{S}})/\mathrm{U_t}} \tag{3.1}$$

where I_{D} is the subthreshold drain current in saturation ($V_{\mathrm{D}} > V_{\mathrm{S}} + 4\mathrm{U_t}$), V_{G} is the gate voltage, I_0 is the ideal current with zero source-to-gate potential, κ relates the gate voltage to the surface potential, and $\mathrm{U_t}$ is the thermal voltage.

Gilbert showed that the resulting linear relationship between transconductance and current can be exploited to implement current-mode circuits, known as translinear circuits, whose outputs are quotients and/or products of combinations of bias and input currents [50, 2]. The translinear principle has been further extended to include transient behavior by adding capacitors, becoming the dynamic translinear principle, which is commonly used to implement a class of linear first-order differential equations [1, 47]. These equations are mathematically equivalent to resistor-capacitor circuits, enabling neuromorphic engineers to use only transistors and capacitors to realize conductance-based neurons (CBNs) [5]. In addition to neurons, the dynamic translinear principle enables the design of lin-

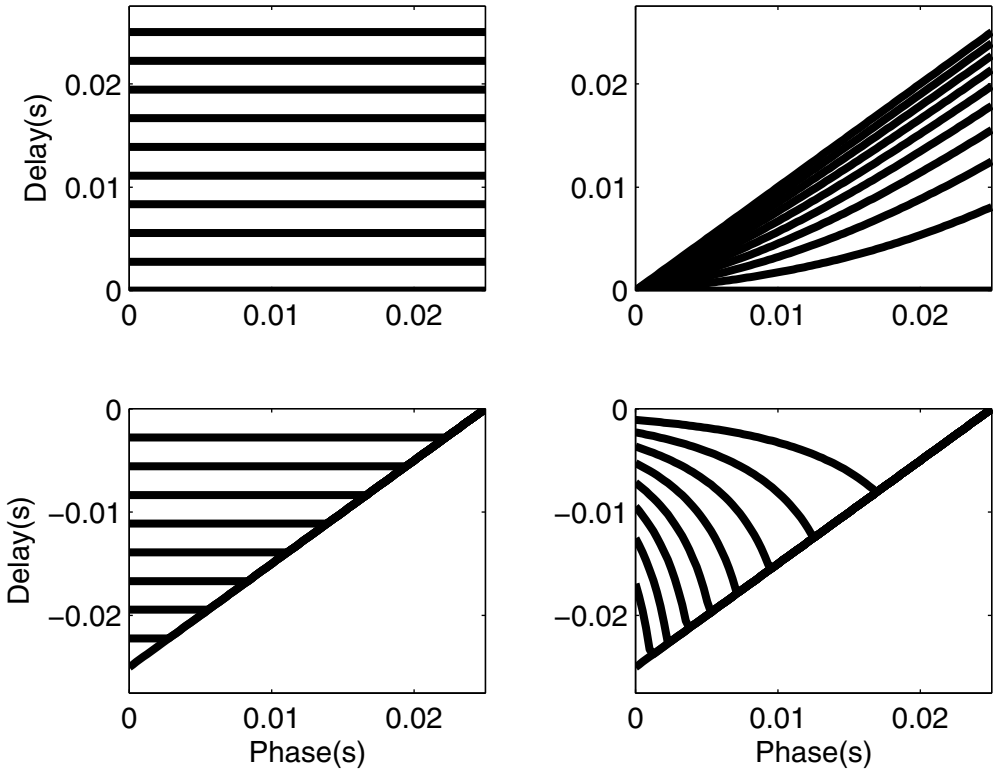

Figure 3.2: Analytically Computed Phase-Response Curves

Phase-response curves due to various strengths of inhibitory or excitatory input for integrator-type and conductance-type neurons. *Top left* The integrator-type neuron's spike is delayed the same amount of time independent of when the inhibition arrives. *Top right* The conductance-type neuron's spike is delayed more when the inhibition arrives later in the interspike interval, providing a structured response to inhibition, which is useful for synchrony. *Bottom left* The integrating-type neuron's spike is advanced the same amount of time independent on when the excitation arrives. However, strong excitation cannot cause the spike to occur before the input, resulting in a line of maximum interspike interval reduction. *Bottom right* The conductance-type neuron's spike is advanced more when the excitation arrives later in the interspike interval, since the affect of excitation has less time to decay away than if the excitation arrive early in the interval.

ear synapses, which, similar to biological synapses, decay exponentially [128, 5]. These synapses are first-order low-pass filters (LPFs). However, the CBN circuit and the LPF synapse have challenges associated with them.

The major challenge of (synapse) LPFs is that they must be interfaced to the address-event representation (AER) link. The AER link is and must be fast, communicating with a synapse for only tens of nanoseconds for each incoming spike [16, 17, 18, 108]. Such speed is necessary because a single link multiplexes inputs from all axons (often greater than 10,000) that project to a chip. To affect the LPF with such a brief duration, the current from the AER communication must be above threshold, causing the LPF to deviate from its designed behavior and introducing nonlinearities into the synapse. In an attempt to alleviate this problem, a current-mirror integrator (CMI) was introduced between the AER signal and the LPF [15][106]. This addition failed to solve the problem, since the CMI is nonlinear and is capable of sending suprathreshold currents to the LPF.

I solved the problem by replacing the CMI with a pulse-extender circuit (PE) [5]. The PE takes the brief AER signal and creates a long (few milliseconds) subthreshold pulse that can be fed into the LPF without deleteriously affecting its behavior. Thus, the dynamic translinear principle enables LPF synapses, and the PE enables address-events to stimulate them.

Another difficulty of CBNs and LPFs is that they are sensitive to transistor leak currents. This leakage is prevalent in deep-submicron technologies (such as the 0.25μm CMOS technology that I used), making the circuits difficult to design and control. The time constants are inversely proportional to the leak currents. If the leak currents are large the time constants are limited to small values, and the circuits are too fast. Some neuromorphic researchers avoid this problem by running their systems faster than real-time, in hypertime,

which is desirable in some cases [69]. However, I wish to avoid hypertime, since it is impossible to speed up physical reality to be appropriate for a hypertime chip. To combat the leak problem I use long transistors (length $>$ 10 times width) to implement the devices whose leak currents influence time constants: This tactic has successfully solved the problem for my chips. In smaller feature technologies (such as 0.18μm CMOS technology) it may be necessary to use thick-oxide transistors, whose leakage currents are significantly less than their normal, thin-oxide counterparts.

3.3 Neuron Circuit

In this section, I describe the neuron circuit, which is composed of soma and synapse circuits, both based on LPFs and PEs. The soma implements membrane dynamics and spiking; the synapse supplies excitation and inhibition. First, I describe the LPF and PE subcircuits. Then, I use them to build the soma and the synapse.

3.3.1 Low-pass Filter

The LPF satisfies a first-order ODE, mathematically identical to a resistor-capacitor (RC) circuit, which models the membrane dynamics of biological neurons. The LPF is a current mode circuit: Its input(s) and output are currents (Figure 3.3). Its output current (Equation 3.1), generated by a transistor (M_1) whose gate is connected to the capacitor (C), is analogous to the RC circuit's potential, whose dynamics are described by:

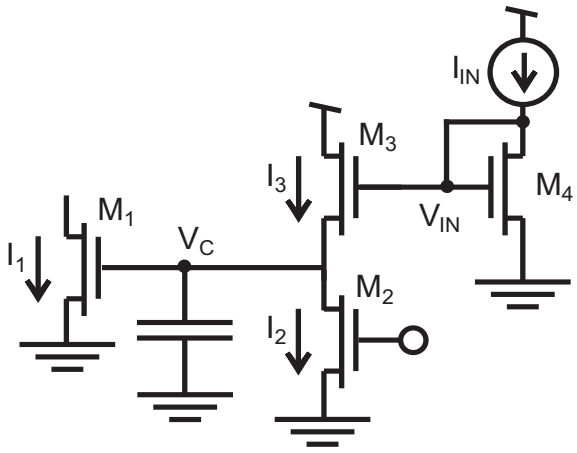

Figure 3.3: Low-Pass Filter

The current-mode low-pass filter (LPF) implements a first-order ODE that models the dynamics of a resistor-capacitor circuit.

$$\mathrm{RC}\frac{dV_{\mathrm{RC}}}{dt} + V_{\mathrm{RC}} = I_{\mathrm{IN}}\,\mathrm{R} \tag{3.2}$$

where V_{RC} is the circuit potential, I_{IN} is the input current, I_{IN} R is the steady-state potential, R is the resistance, and C is the capacitance. Like an (undriven) RC circuit initialized to a nonzero potential, M_1's current decays exponentially to zero. Transistor M_2 implements the exponential decay with a (typically) constant current sink (I_2), which linearly discharges the capacitor node (V_{C}) as described by:

$$V_{\mathrm{C}}(t) = \mathrm{V}_{\mathrm{INIT}} - \frac{\mathrm{I}_2}{\mathrm{C}}t \tag{3.3}$$

where V_{C} is initialized to $\mathrm{V}_{\mathrm{INIT}}$ and t is time. From Equation 3.1, I_1 is proportional to

an exponent to the power of V_{C}. Therefore, linearly discharging V_{C} (to ground) realizes the exponentially decaying output, due to M_1's exponential (gate) voltage-current relation, which is described by:

$$\begin{aligned} I_1(t) &= \mathrm{I}_0\, e^{\kappa V_{\mathrm{C}}(t)/\mathrm{U_t}} \\ &= \mathrm{I}_0\, e^{\kappa(\mathrm{V_{INIT}} - \mathrm{I}_2 t/\mathrm{C})/\mathrm{U_t}} \\ &= \mathrm{I_{INIT}}\, e^{-t/\tau} \end{aligned} \tag{3.4}$$

where I_1 is the output current (M_1), $\mathrm{I_{INIT}} = \mathrm{I}_0\, e^{\kappa \mathrm{V_{INIT}}/\mathrm{U_t}}$ is the initial output current, τ is the decay constant ($\frac{\mathrm{C\,U_t}}{\kappa \mathrm{I}_2}$). Thus, an RC-like exponential decay is achieved; implementing RC-like input-driven growth is less intuitive.

I must compensate for the logarithmic transistor properties to supply appropriate inputs to the LPF; because the output current is an exponential of V_{C}, one cannot drive the capacitor directly. A direct input current competes with the leak current (I_2): If the input current exceeds the leak, V_{C} increases linearly, causing I_1 to increase exponentially, drastically different from the RC dynamics I seek. In an RC circuit, the potential approaches its steady-state value with a slope proportional to the difference between the present and the steady-state potentials (Equation 3.2). So as the output current increases, I must decrease the effectiveness of the input (avoiding exponential growth). To realize such dynamics, I use a source-coupled current mirror (M_{3-4}): As V_{C} increases, the current through the input transistor (M_3) decreases exponentially, which exactly compensates for the output transistor's exponent (assuming $\kappa = 1$). The input (I_3) through the source-coupled current mirror is given by:

$$
\begin{aligned}
I_3(t) &= \mathrm{I_0}\, e^{\kappa V_{\mathrm{IN}}(t)/\mathrm{U_t} - V_{\mathrm{C}}(t)/\mathrm{U_t}} \\
&= \mathrm{I_0}\, \mathrm{I_0}\, e^{\kappa V_{\mathrm{IN}}(t)/\mathrm{U_t}} \div \mathrm{I_0}\, e^{V_{\mathrm{C}}(t)/\mathrm{U_t}} \\
&= \frac{\mathrm{I_0}\, I_{\mathrm{IN}}(t)}{I_1(t)}
\end{aligned}
\tag{3.5}
$$

where $V_{\mathrm{IN}}(t)$ is the voltage on M_4's gate due to the input current ($I_{\mathrm{IN}}(t)$).

Now, I verify that the ODE that describes the LPF is analogous to the one that describes the RC circuit. I solve for the ODE by applying Kirchhoff's Current Law at node C, which gives:

$$
\begin{aligned}
\mathrm{C}\frac{dV_{\mathrm{C}}}{dt} &= -\mathrm{I_2} + I_3 \\
&= -\mathrm{I_2} + \frac{\mathrm{I_0}\, I_{\mathrm{IN}}}{I_1}
\end{aligned}
\tag{3.6}
$$

The output is I_1; to relate its dynamics to those of V_{C}, I differentiate Equation 3.4 with respect to time:

$$
\frac{dI_1}{dt} = I_1 \frac{\kappa}{\mathrm{U_T}} \frac{dV_{\mathrm{C}}}{dt} \tag{3.7}
$$

which I solve for $\frac{dV_{\mathrm{C}}}{dt}$ and substitute into Equation 3.6. Then I multiply and divide each side of the equation by I_1 and $\mathrm{I_2}$, respectively, which results in:

$$\tau \frac{dI_1}{dt} + I_1 = \frac{I_{\text{IN}}\,\text{I}_0}{\text{I}_2} \tag{3.8}$$

which is analogous to the RC circuit (Equation 3.2). Thus far, I have assumed that $\kappa = 1$, but this is not correct ($\kappa < 1$).

I must compensate for this non-unity κ for the LPF to function properly. If I correct the assumption Equation 3.5 becomes:

$$I_3(t) = I_{\text{IN}}\left(\frac{\text{I}_0}{I_1}\right)^{1/\kappa} \tag{3.9}$$

Now if I solve Equation 3.8 for its steady state (constant I_{IN}) with nonunity κ, I calculate:

$$I_1(\infty) = \text{I}_0\left(\frac{I_{\text{IN}}}{\text{I}_2}\right)^{\kappa} \tag{3.10}$$

Since $\kappa < 1$, I_1 is suppressed, unable to increase in proportion to I_{IN} (Figure 3.4).

I compensate for κ by connecting M_3's bulk to its source (instead of to ground). The problem is the source voltage has a stronger influence on the transistor current than the gate voltage, since the effectiveness of the source is not decreased by κ (Equation 3.1). I reduce the source's influence on the current by connecting it to the bulk, which modulates the current with an effectiveness of 1-κ. The combined effectiveness of the shorted source

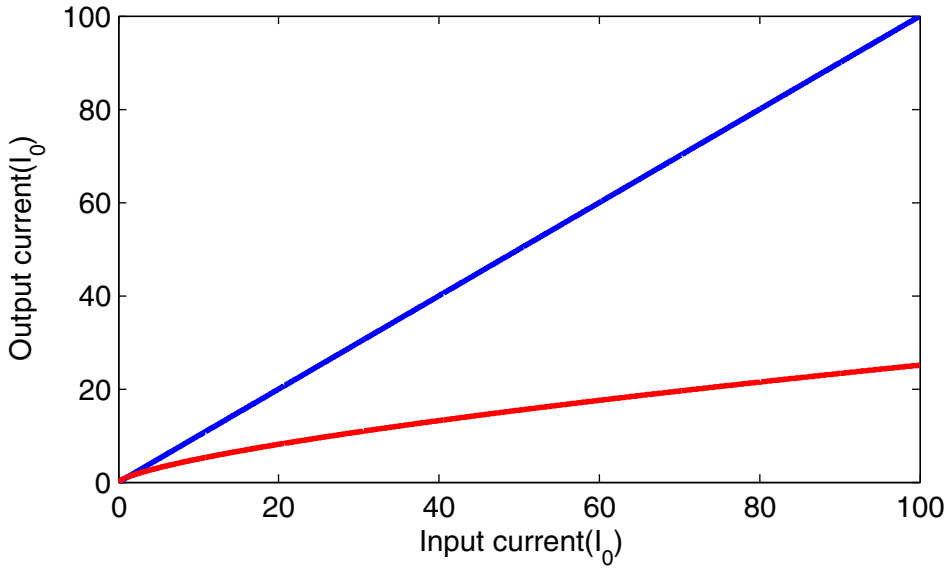

Figure 3.4: Depiction of the κ Effect

When I compensate for non-unity κ (see text), the low-pass filter's output increases linearly (blue). When I do not compensate, the low-pass filter's output increases sublinearly and is substantially suppressed (red).

and bulk is equal to the effectiveness of the gate but opposite in direction, which restores the appropriate LPF dynamics (Equation 3.8).

In most CMOS processes (including the one I use), NMOS transistors are the native devices, meaning their bulks are all shorted together through the substrate, leaving them unable to implement LPFs. Therefore, to build LPFs as described, I use PMOS transistors, which are fabricated in wells. The bulks of all transistors within a well are shorted, but each well is independent of its neighbors, allowing me to implement LPFs despite κ. The input to the LPF can be either a continuous current or pulses of current from a PE.

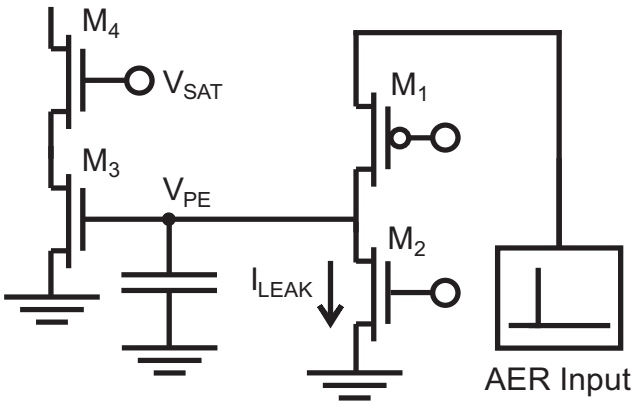

Figure 3.5: Pulse Extender

The pulse extender takes a brief (tens of nanoseconds) address-event pulse input and creates a long (up to tens of milliseconds) subthreshold current pulse.

3.3.2 Pulse Extender

In Section 3.2, I introduced the PE and its role as the interface between fast, digital AER circuits and slow, analog LPF circuits, but the PE serves other functions as well: It implements synaptic rise-time, which is important in generating synchrony, and neuronal refractory period, which helps to control excitability. The PE performs these functions by using only a handful of transistors and a capacitor to generate a subthreshold current pulse in response to a (digital) suprathreshold current pulse (Figure 3.5). During an AER communication, the receiver circuitry dumps a quanta of charge onto the PE capacitor (M_1), increasing its output voltage by $\frac{Q_{IN}}{C}$, where Q_{IN} is the quanta of charge and C is the PE capacitance. The charge decays away at a rate determined by the PE leak current (M_2).

The PE's output voltage is a triangle (Figure 3.6) described by (for an NMOS implementation):

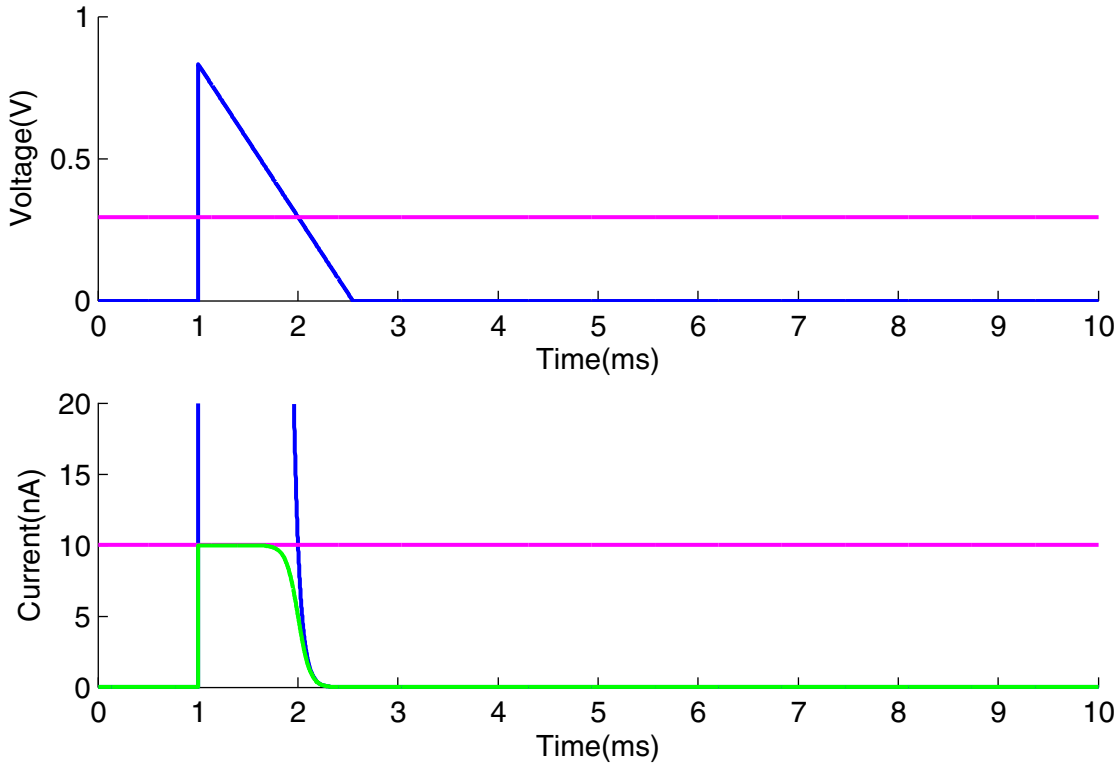

Figure 3.6: Depiction of Silicon Pulse Extender

Top The output voltage of the pulse extender (PE) decays linearly in time after a spike input, forming a triangular pulse (blue), which exceeds the constant voltage set by a bias (magenta). *Bottom* Applying these voltages to the gates of two transistors produces the currents shown (color coded). When connected in series, the two transistors produce a subthreshold current pulse (green).

$$\mathrm{C}\frac{dV_{\mathrm{PE}}}{dt} = Q_{\mathrm{IN}}\delta(t - t_i) - \mathrm{I}_{\mathrm{LEAK}} \tag{3.11}$$

where V_{PE} is the PE voltage, t_i is the arrival time of AER signals (t is time), $\mathrm{I}_{\mathrm{LEAK}}$ is the leak current, and δ is Dirac's delta function. If V_{PE} is applied to the gate of a transistor (M_3), its current will decay exponentially in time. However, typically, I apply V_{PE} to the gate of a transistor in series with another transistor (M_4), which has a constant, subthreshold gate

voltage (V_{SAT}). The current through this series pair is $\frac{I_3 I_4}{I_3+I_4}$, where I_3 and I_4 are the current M_3 and M_4 would pass on their own, respectively. While V_{PE} is larger than V_{SAT}, the pair of transistors will pass approximately I_4, until V_{PE} decays to below V_{SAT} (Figure 3.6). While V_{PE} is below V_{SAT}, the pair of transistors pass approximately I_3, resulting in an exponential tail after a current pulse of amplitude I_4.

PEs must supply LPFs with subthreshold current pulses. When two inputs arrive in short succession, the pulse amplitude cannot increase. Instead, the triangular voltage pulse becomes twice as long, producing a wider current pulse that drives the output twice as high if the LPF's decay constant is much longer than the pulse width. If V_{SAT} is much higher than ground, however, the effect of two pulses may be greater or less than twice the effect of one. For example, if the second spike arrives as the first pulse reaches V_{SAT}, the contribution of the second spike will be greater than the first because the pulse started at a higher voltage.

The PE facilitates when two AER spikes arrive in rapid succession (μs apart); the pulse becomes more than twice the width of the pulse a single AER spike produces. The second pulse is extended by the time it takes the voltage to decay from V_{SAT} to ground (this effect can be mitigated by setting V_{SAT} to just above ground and using a current amplifier to control the pulse amplitude). However, V_{PE} cannot exceed the voltage supply, therefore, if a burst of several AER spikes arrives in rapid succession, the resulting pulse cannot exceed a maximum width, $\frac{CV_{DD}}{I_{LEAK}}$. For instance, if five rapid AER spikes saturate the PE voltage, reaching the maximum pulse width, then additional spikes above five will fail to increase it. Therefore, in this regime the PE implements depression (See Section 3.4.3).

3.3.3 Soma Circuit

I construct the soma from three subcircuits: the membrane, the axon hillock (AH), and the refractory channel population (RCP) (Figure 3.7); I drive it with a simple excitatory synapse built from a PE (See Section 3.3.4). The membrane LPF realizes a leaky-integrator (RC) response to excitatory current and shunting inhibition. An input current drives the capacitor (C_L) through a source-coupled current mirror ($M_{L1\text{-}2}$). As the capacitor voltage approaches M_{L2}'s gate voltage, the current decreases, compensating for the transistors' nonlinear (logarithmic) voltage-current relation [60]. In this section, I set the input current (I_{IN}) and leak current (I_{SHUNT}) to be constant in time; I_{SHUNT} comprises the sum of an inhibitory synaptic current (which is off) and a constant current (M_{L4}) as well as a reset/refractory current (M_{L3}).

The membrane LPF's output (analogous to the potential of an RC circuit) is the soma current, I_{SOMA} (M_{A1}). Increasing I_{SHUNT} reduces the membrane's steady-state output and decreases its time constant (identical to increasing the conductance in an RC circuit). Since the soma is an LPF, it follows from Equation 3.8 that:

$$\tau \frac{dI_{SOMA}}{dt} + I_{SOMA} = \frac{I_{IN} I_0}{I_{SHUNT}} \tag{3.12}$$

where $\tau = \frac{C_L U_t}{\kappa I_{SHUNT}}$ is the soma's time constant. In this chapter, I assume I_{SHUNT} and, therefore, τ are constant.

In addition to the constant excitatory input and inhibitory leak, the membrane also receives a positive feedback current from the axon-hillock (modified from [32] by Kai

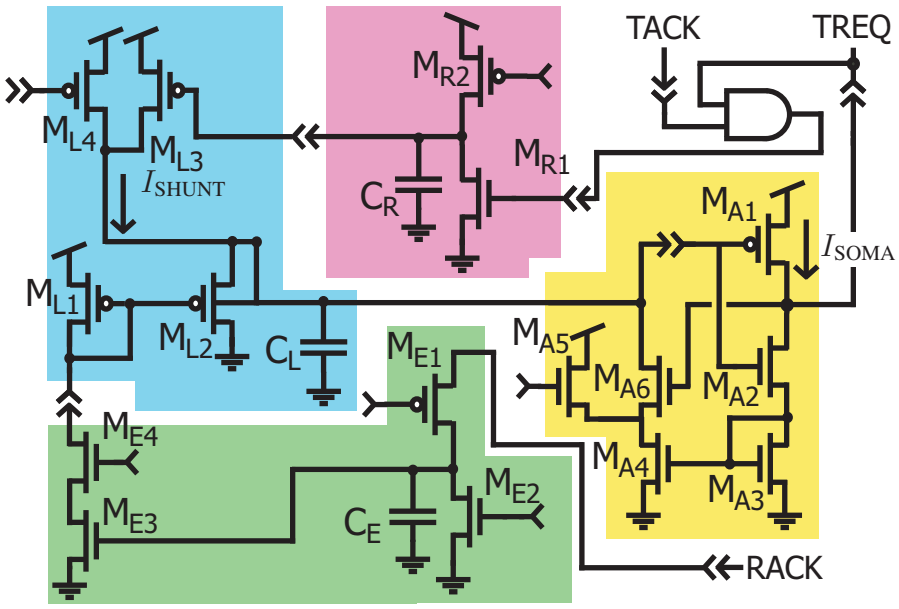

Figure 3.7: Soma Circuit

The soma circuit from which both the pyramidal neuron and interneuron are derived includes a membrane LPF ($M_{L1\text{-}4}$), an axon-hillock ($M_{A1\text{-}6}$), a refractory channel population ($M_{R1\text{-}2}$), and excitatory synapse ($M_{E1\text{-}4}$). Single arrows represent bias voltages; double arrows represent inputs and outputs. The AND gate resets the neuron's spike request to the transmitter (TREQ), gated by the acknowledge (TACK). The receiver acknowledge (RACK) activates the synapse.

Hynna). As I_{SOMA} increases, the feedback current (M_{A6}) turns on more strongly, overpowering the leak to cause a spike.

The feedback current (I_{A6}) increases in proportion to the square of I_{SOMA}. I_{SOMA} is mirrored ($\text{M}_{\text{A3-4}}$) to the AH's differential pair ($\text{M}_{\text{A5-6}}$). M_{A6}'s gate connects to the drain of M_{A2}, which behaves like a short circuit (before a spike), shorting M_{A6}'s gate to the current mirror, hence the square dependence on I_{SOMA}. The square dependence arises because increasing I_{SOMA} switches on a side of the differential pair and increases its tail current.

This relation is valid as long as M_{A5}'s gate voltage exceeds that of M_{A6}. I_{A6} is described by:

$$I_{\text{A6}} = \frac{{I_{\text{SOMA}}}^2}{\text{I}_{\text{DP}}} \tag{3.13}$$

where $\text{I}_{\text{DP}} = \text{I}_0 e^{\kappa \, \text{V}_{\text{DP}}/\text{U}_\text{t}}$, and V_{DP} is the gate voltage of M_{A5}. Because AH (through I_{A6}) drives the LPF capacitor directly, to calculate its affect (I_{AH}) on the membrane LPF, I must divide by $\frac{I_{\text{SHUNT}}}{\text{I}_0}$ and multiply by the $\frac{I_{\text{SOMA}}}{\text{I}_0}$ (opposite as in Section 3.3.1), which results in:

$$I_{\text{AH}} = \frac{{I_{\text{SOMA}}}^3}{\text{I}_{\text{DP}}\,\text{I}_{\text{SHUNT}}} \tag{3.14}$$

Augmenting Equation 3.12 with this result yields:

$$\begin{aligned} \tau \frac{dI_{\text{SOMA}}}{dt} &= -I_{\text{SOMA}} + \frac{{I_{\text{SOMA}}}^3}{\text{I}_{\text{DP}}\,\text{I}_{\text{SHUNT}}} + \frac{\text{I}_{\text{IN}}\,\text{I}_0}{\text{I}_{\text{SHUNT}}} \\ &= -I_{\text{SOMA}} + \frac{{I_{\text{SOMA}}}^3}{{\text{I}_{\text{PTH}}}^2} + I_{\text{INE}} \end{aligned} \tag{3.15}$$

where $I_{\text{INE}} = \frac{\text{I}_{\text{IN}}\,\text{I}_0}{\text{I}_{\text{SHUNT}}}$ is the effective input current and $\text{I}_{\text{PTH}} = \sqrt{\text{I}_{\text{DP}}\,\text{I}_{\text{SHUNT}}}$ is proportional to the spike threshold. If the effective input is less than some percentage of I_{PTH}, the effective input and AH feedback cannot overcome the leak, and the soma will not spike. For larger input currents, I_{SOMA} increases, which increases I_{AH}, and the resulting positive feedback causes a spike. When a spike occurs, AH initiates the process of transmitting it off chip, which activates RCP.

RCP shunts I_{SOMA} to near zero (pulls C_L to V_{DD}) for a brief period (a few milliseconds) after a spike, using a modified PE. Its capacitor (C_R) is pulled to ground during a spike (M_{R1}), which causes M_{L3} to drive C_L to V_{DD}, until the leak through M_{R2} restores C_R.

3.3.4 Synapse Circuit

Like the soma, the synapse is based on an LPF; I construct it from two subcircuits: the cleft and the receptor (Figure 3.8). The cleft ($\text{M}_{\text{C1-2}}$) models the neurotransmitter release into the gap between presynaptic and postsynaptic neurons; I implement it with a PE, which sets the synapse's rise-time. The cleft drives the receptor ($\text{M}_{\text{G1-4}}$), which models the conductance of the neurotransmitter-gated channel population. I implement the receptor with an LPF, which sets the synapse's decay constant (similar to [128]).

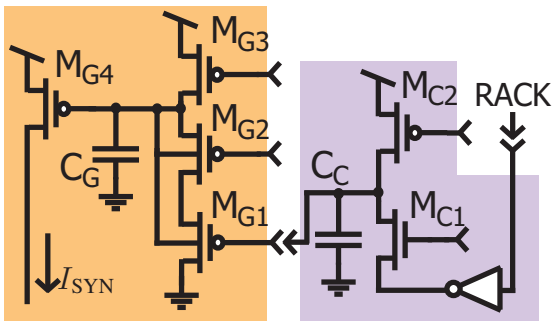

Figure 3.8: Synapse Circuit

The synapse circuit consists of a pulse extender (PE) and a low-pass filter (LPF). The PE implements a synaptic cleft ($M_{C1\text{-}2}$) and the LPF implements a synaptic conductance ($M_{G1\text{-}4}$). The receiver acknowledge (RACK) activates the synapse.

Thus, I model the synapse as a pulse of neurotransmitter filtered by an LPF, identical to simplified kinetic models [35], described by:

$$\tau \frac{dI_{\text{SYN}}}{dt} + I_{\text{SYN}} = \frac{I_{\text{PULSE}}\, \text{I}_0}{\text{I}_\tau} \tag{3.16}$$

where I_{SYN} is the synaptic output current (assuming M_{G4}'s source is connected to V_{DD}), I_{PULSE} is the input current from the PE through $M_{G1\text{-}2}$, whose maximum current is set by M_{G2}'s gate, and I_τ is the current through M_{G3}, which sets the synaptic decay constant, $\tau = \frac{C_G\, U_t}{\kappa I_\tau}$. It is important to note how the PE connects to the LPF: Rather than driving two series PMOSs with their (collective) source connected to V_{DD}, the PE drives two PMOSs with their source directly connected to the LPF capacitor. This structure saves two current mirrors, which reduces variations among circuits and reduces the space required for each one. I use this same strategy to connect synapse LPFs that excite the soma; the PMOS

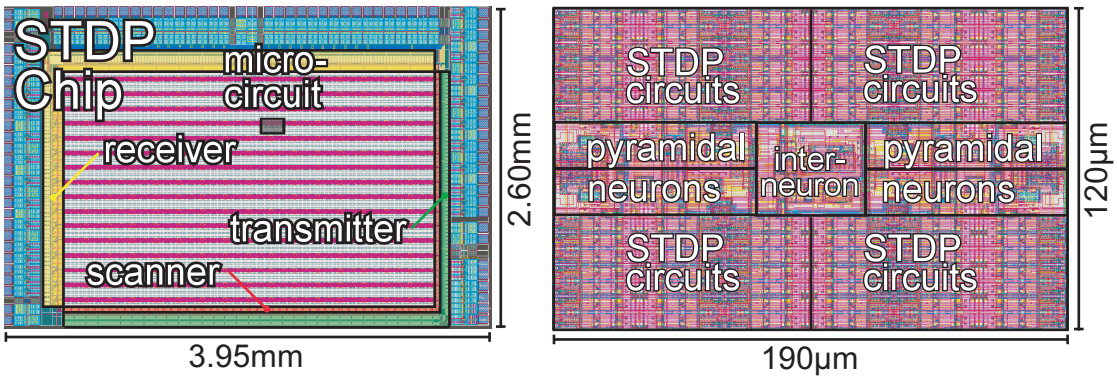

Figure 3.9: STDP Chip

Left The STDP Chip includes an address-event receiver (yellow) and transmitter (green), a mixed analog-digital scanner (red), and a 16-by-16 array of microcircuits (black box). *Right* One microcircuit includes one inhibitory interneuron and four excitatory pyramidal neurons, each with 21 STDP circuits.

output transistor directly connects its source to the soma LPF capacitor.

The basic neuron implementation consists of the soma with a simple synaptic input. The excitatory input to the soma models an AMPA synapse. It consists solely of a PE, which models both the neurotransmitter (glutamate) release into the synaptic cleft as well the brief conductance change (a few milliseconds) of AMPA channels in response to a presynaptic spike. Since the reversal potential of AMPA is much higher than the membrane potential, I approximate the conductance with a current without dramatically altering the behavior of the neuron. Therefore, when activated, the AMPA synapse directly excites the soma with a pulse of current. The AMPA synapse circuit is the simplest type of synapse, modeling the postsynaptic current as a brief current pulse.

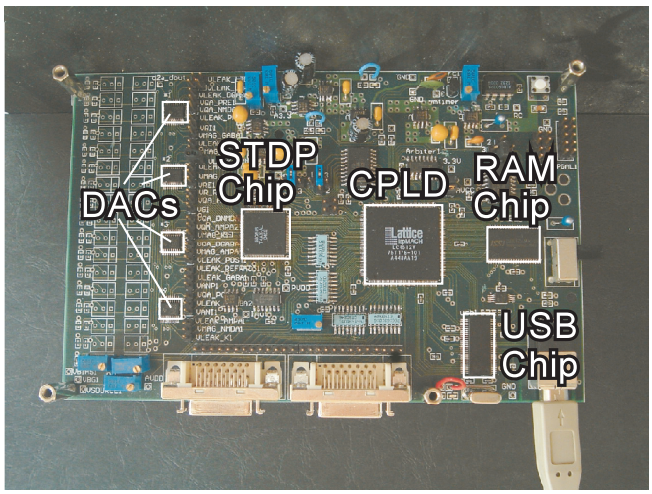

Figure 3.10: Embedded System

The STDP Chip is embedded in a circuit board including DACs, a CPLD, a RAM chip, and a USB chip, which communicates with a PC.

3.4 Characterization

I have designed, submitted, and tested a chip with an array of my silicon neurons. The STDP Chip was fabricated through MOSIS in a 1P5M 0.25μm CMOS process, with just under 750,000 transistors in just over 10mm^2 of area (Figure 3.9). It has a 16 by 16 array of microcircuits; each contains one inhibitory interneuron (28 by 36μm each) commingled with four excitatory pyramidal neurons, each with 21 STDP synapses (Chapter 5). The STDP Chip employs AER to transmit spikes off chip and to receive spike input [16, 17, 18]. In addition, it includes an analog scanner that allows me to observe the state of one neuron at a time (either its synapse or soma) [103].

To test the silicon neurons, I embedded the STDP Chip in a circuit board (Fig. 3.10). The board has four primary components: a CPLD (complex programmable logic device), the STDP Chip, a USB interface chip, and DACs (digital-to-analog converters). The central

component in the system is the CPLD; it mediates communication between the STDP Chip and the USB chip, which provides a bidirectional link with a PC. The DACs control the analog biases in the system, allowing the PC to control the system via USB.

In characterizing the neuron, I focused on three aspects: the soma's frequency-current curve (FIC), the synapse's decay constant and rise-time, and a pyramidal neuron's PRC (due to excitation), which summarizes the contributions of both the soma and synapse to spiking behavior.

3.4.1 Frequency-Current Curve

To characterize the soma's spiking behavior, I drove it with various levels of constant currents. I generated constant current by repetitively stimulating the AMPA synapse using a period (1ms) less than its pulse width (10ms), which prevented the pulse from turning off. The result was a constant current with amplitude set by the gate of M_{E4} (Figure 3.7). For small currents, I_{SOMA} reached a steady state without spiking, whereas as the current increased the soma spiked with decreasing period (Figure 3.11).

To derive the soma's FI curve, I analyzed Equation 3.15 to determine the spike frequency for constant input current. Then, I verified this result by fitting data from a silicon neuron. First, I normalized by I_{PTH}:

$$\tau \frac{dx}{dt} = -x + x^3 + r \tag{3.17}$$

where $x = \frac{I_{SOMA}}{I_{PTH}}$ is the normalized soma current and $r = \frac{I_{INE}}{I_{PTH}}$ is the normalized input

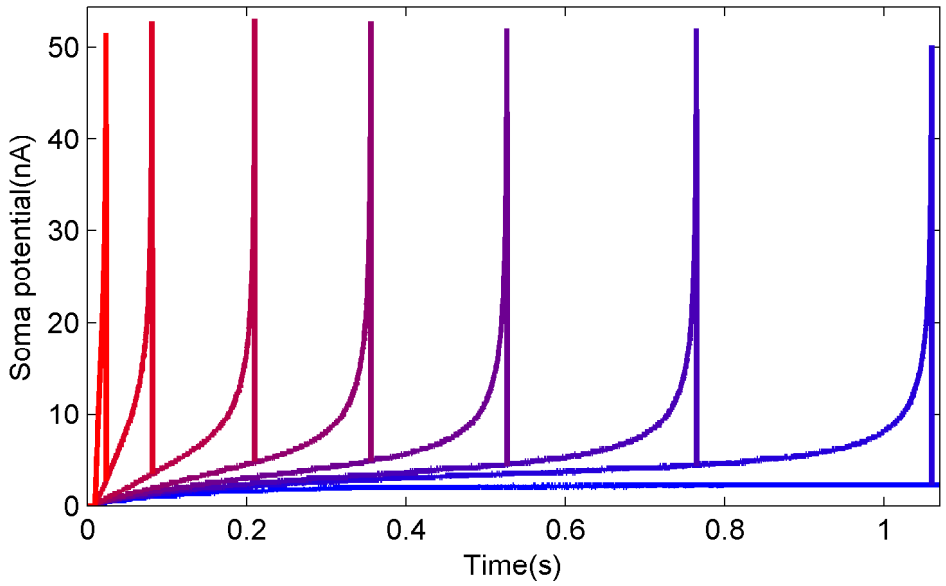

Figure 3.11: Soma Trajectory

Soma membrane (current) traces for several step-input current levels (increasing blue to red) rise like a resistor-capacitor circuit and a positive feedback spike. Below a threshold, the soma reaches a steady state at which the axon-hillock current is insufficient to overpower the leak. Above threshold, increasing input current enables the axon-hillock to overcome the leak more quickly, resulting in a shorter period.

current. Note: I_{PTH} (and therefore I_{SHUNT}) must be constant for this analysis. I found the bifurcation point, γ, of Equation 3.17 by solving it for $\frac{dx}{dt} = 0$ [137]. Qualitatively, I observed that $\frac{dx}{dt}$ has two fixed points ($x > 0$) for $r < \gamma = \frac{2}{3}\sqrt{\frac{1}{3}} \approx 0.38$ (Figure 3.12). The lower one is stable, the higher is unstable. If x is initialized below the higher fixed point, it decays to the lower; If x is initialized above the higher fixed point, it increases without bound (spikes).

At $r = \gamma$, the two fixed points merge into one. Increasing r further eliminates the equilibrium point, entering the regime where constant current causes the soma to spike: x increases from zero without bound (until it is reset). When r surpasses γ the soma's dynam-

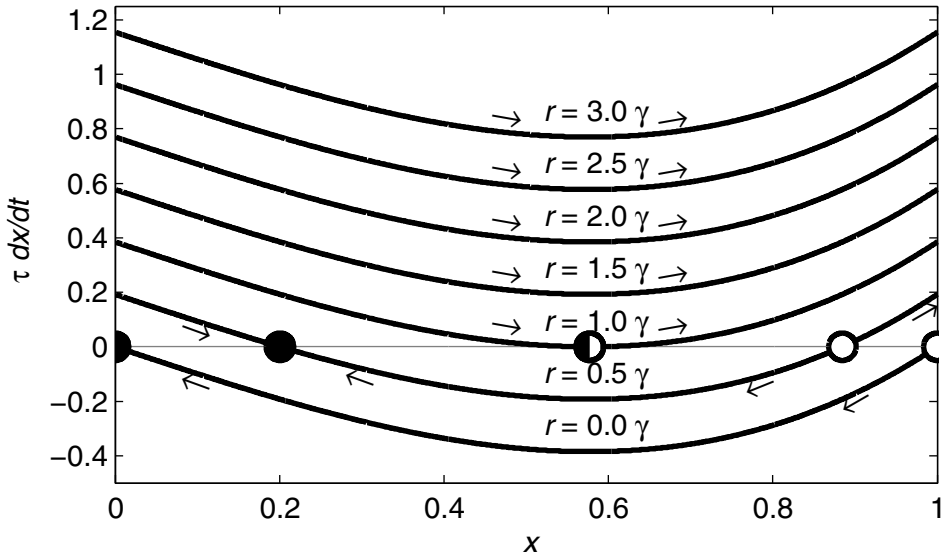

Figure 3.12: Soma Dynamics

The normalized soma current, x, has two fixed points (circles) when its input is small, $r < \gamma$, one stable (filled) and one unstable (open). As r increases beyond γ, the fixed points merge and destroy each other, entering the regime where the soma spikes with constant current input. The arrows show the direction x takes towards and away from fixed points.

ics change; the two fixed points merge, annihilating each other (saddle node bifurcation). Without any fixed points, x spikes, regardless of where it starts.

To solve Equation 3.17 for the spike period, I integrate it from $x = 0$ to x_{MAX}:

$$\int_0^{x_{\text{MAX}}} \tau \frac{dx}{-x + x^3 + r} = \int_0^T dt = T \tag{3.18}$$

where x_{MAX} ($\gg 1$) is the value at which a spike is sent off chip and x is reset to zero, and T is the spike period for a given input, r. However, this integral has no analytical solution,

so I break it into three regions, the first and last of which I can analytically solve:

$$\tau \frac{dx}{dt} = \begin{cases} -x + r, & x \ll \mathrm{r}^{1/3} \\ -x + x^3 + r, & x \text{ on the order of } r^{1/3} \\ -x + x^3, & x \gg \mathrm{r}^{1/3} \end{cases} \tag{3.19}$$

If the time x spends in the first and last regions dominates the second region, then I can analytically approximate the FI curve by ignoring the second region. The first region yields one minus an exponential rise to reach a set value, $\mathrm{t_{RC}}$:

$$\begin{aligned} \tau \frac{dx}{dt} &= -x + r, \quad x \leq \alpha r^{1/3}, \quad \alpha < 1 \\ x(t) &= \mathrm{r}(1 - e^{-t/\tau}), \quad x \leq \alpha r^{1/3} \\ x(\mathrm{t}_{RC}) &= r(1 - e^{-\mathrm{t}_{RC}/\tau}) = \alpha r^{1/3} \\ \mathrm{t}_{RC} &= \tau \log\left(\frac{1}{1 - \alpha r^{-2/3}}\right) \end{aligned} \tag{3.20}$$

As expected, if r is small, $\mathrm{t_{RC}}$ is large; if r is large, $\mathrm{t_{RC}}$ is small. Based on chip data (Figure 3.11) and simulation (not shown), this solution is a good approximation for $\alpha \leq 0.3$, where α is a scaling factor.

Now I skip to region 3, solving for the time the soma takes to complete a spike, $\mathrm{t_{SPK}}$, once the positive feedback dominates over the input:

$$
\begin{aligned}
\tau\frac{dx}{dt} &= -x + x^3, \quad x > \beta r^{1/3}, \quad \beta > 1 \qquad (3.21)\\
\mathrm{t_{SPK}} &= \int_0^{\mathrm{t_{SPK}}} dt = \int_{\beta r^{1/3}}^{x_{\mathrm{MAX}}} \tau\frac{dx}{-x + x^3}\\
\mathrm{t_{SPK}} &= \frac{\tau}{2}\log\left(\frac{1}{1 - \frac{1}{\beta^2}r^{-2/3}}\right)
\end{aligned}
$$

I could not solve analytically for $x(t)$, but I did solve for $\mathrm{t_{SPK}}$. Based on chip data (Figure 3.11) and simulation (not shown), this solution is a good approximation for $\beta \geq 1.2$, where β is a scaling factor. Note that the form of the solutions for $\mathrm{t_{RC}}$ and $\mathrm{t_{SPK}}$ are the same. Unfortunately, α and β are not nearly equal, therefore, I cannot determine the period from this approach: The first and third regions represent only a small part of the neuron's period, and the second region is intractable, so I cannot calculate the time x spends between $0.3\,r^{1/3}$ and $1.2\,r^{1/3}$.

To resolve this problem, I adopt another approach: I can integrate and solve for the period exactly if r is large, much larger than x for all x until $x^3 \gg x$ (i.e., $x^3 + r \gg x$, therefore $r \gg 1$).

$$
\int_0^{x_{\mathrm{MAX}}} \tau\frac{dx}{x^3 + r} = \tau\pi\gamma r^{-2/3} = \int_0^T dt = T \qquad (3.22)
$$

Since the frequency is the inverse of the period:

$$f = \frac{1}{T} = \frac{1}{\tau\pi\gamma} r^{2/3} = \frac{1}{\tau\pi\gamma}\left(\frac{I_{\text{INE}}}{\text{I}_{\text{PTH}}}\right)^{2/3} \tag{3.23}$$

for $r \gg 1$. From this equations, I fit τ after calculating I_{PTH}. The smallest input that reaches threshold is $r_0 = \frac{I_{\text{INE0}}}{\text{I}_{\text{PTH}}} = \gamma$. We solve for $\text{I}_{\text{PTH}} = \gamma I_{\text{INE0}}$, where I_{INE0} is the input current threshold.

The previous result (Equation 3.23) is only valid for large r, but I make a better approximation by looking at Equation 3.17. If $r < \gamma$ the frequency is zero, but Equation 3.23 does not satisfy this condition. I include this condition by assuming that Equation 3.23 is an approximation to an inverse logarthim of the same form as (the inverse of) Equations 3.20 & 3.21. Because $\left[\log\left(\frac{1}{1-y^{-1}}\right)\right]^{-1} = y$, for $y \gg 1$, but approaches zero as y approaches one; an equation of this form can satisfy Equation 3.23 when $r \gg 1$ and cease to spike below threshold ($r = \gamma$). We arrive at this equation by substituting $\left[\log\left(\frac{1}{1-(r/\gamma)^{-2/3}}\right)\right]^{-1}$ for $(r/\gamma)^{2/3}$ in Equation 3.23 yielding:

$$f = \frac{1}{\tau\pi}\frac{1}{\gamma^{1/3}}\left[\log\left(\frac{1}{1-\left(\frac{r}{\gamma}\right)^{-2/3}}\right)\right]^{-1} = \frac{1}{\tau\pi}\frac{1}{\gamma^{1/3}}\left[\log\left(\frac{1}{1-\left(\frac{I_{\text{INE}}}{\text{I}_{\text{PTH}}\,\gamma}\right)^{-2/3}}\right)\right]^{-1} \tag{3.24}$$

This approximation is valid for all values of $r > \gamma$, below which f is zero.

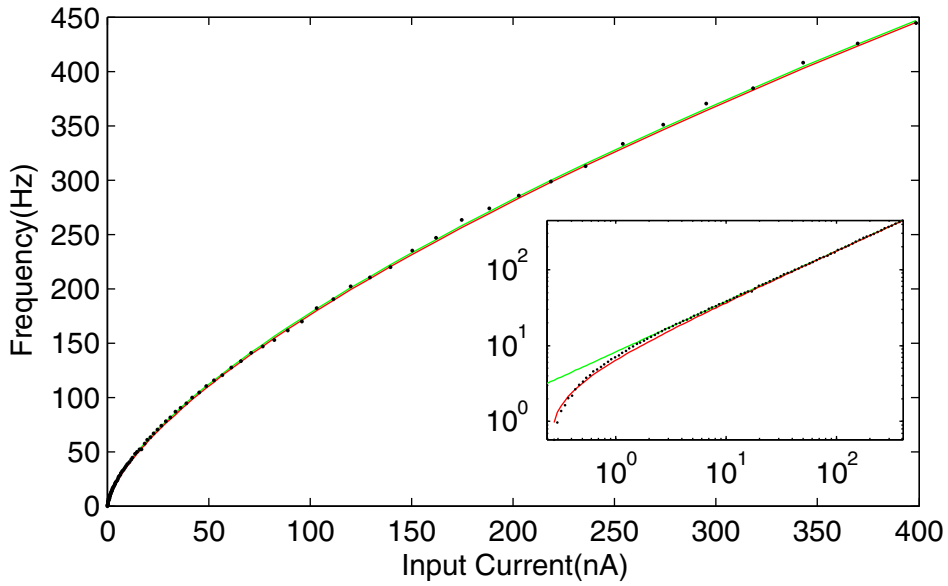

Figure 3.13: Frequency-Current Curve

The soma responds sublinearly to current above a threshold (dots). Its frequency is approximately proportional to the input current to the power of $\frac{2}{3}$ (green). However, the frequency is more complex as described in the text (red). *Inset* Both functions fit well far above threshold (0.2757nA), but the more complex function fits better near threshold (Note the log-log axes). τ was 129ms.

I verify my results, Equations 3.23 & 3.24 by comparing them to the soma's FI curve. I measure the FI curve by driving the soma with a current stepped from zero to a constant value and observing the time to spike, which is the inverse of the neuron's frequency (Figure 3.13).[1]

[1] Alternatively, I could drive the neuron with a constant current and measure its average period. However, for the purpose of analysis I wish to eliminate the affect of the refractory period, which adds a fixed number of milliseconds to the period.

3.4.2 Synaptic Rise and Decay

When stimulated with a spike, the synaptic current increased linearly, initially (far from the maximum level), and then decreased exponentially. I characterized the synaptic decay constant by varying the receptor's leak current (adjusting M_{G3}'s gate voltage; see Figure 3.8). I measured the resulting synaptic current's decay constant by fitting the logarithm of the decaying synaptic current with a line (Figure 3.14). The decay constant depended exponentially on M_{G3}'s gate voltage, because it is inversely proportional to the current through this transistor. I note that the decay is close to exponential but deviates slightly. I attribute this deviation to nonlinear behavior of the current-output pad, but other potential causes include: capacitive coupling between the gate of M_{G1} and C_G or LPF nonlinearity due to κ variations among transistors (See Appendix B).

I characterized the synaptic rise-time by varying the cleft's leak current (adjusting M_{C2}'s gate voltage; see Figure 3.8) and hence the pulse-width. I measured the resulting synaptic current's rise-time, defined as the time-to-peak (Figure 3.15). The rise-time depended exponentially on M_{C2}'s gate voltage, because the pulse-width is inversely proportional to the current through this transistor. Also, the peak current increased with the pulse width, since the receptor's current had more time to rise.

3.4.3 Synaptic Summation

I want to use my silicon synapse to model a population of synapses, using one synapse to model thousands, which both shrinks and simplifies my circuits, saving silicon area. When sharing one synapse among many inputs, I desire the synapse to sum its inputs linearly, equivalent to having many independent synapses as in biology. Equation 3.8,

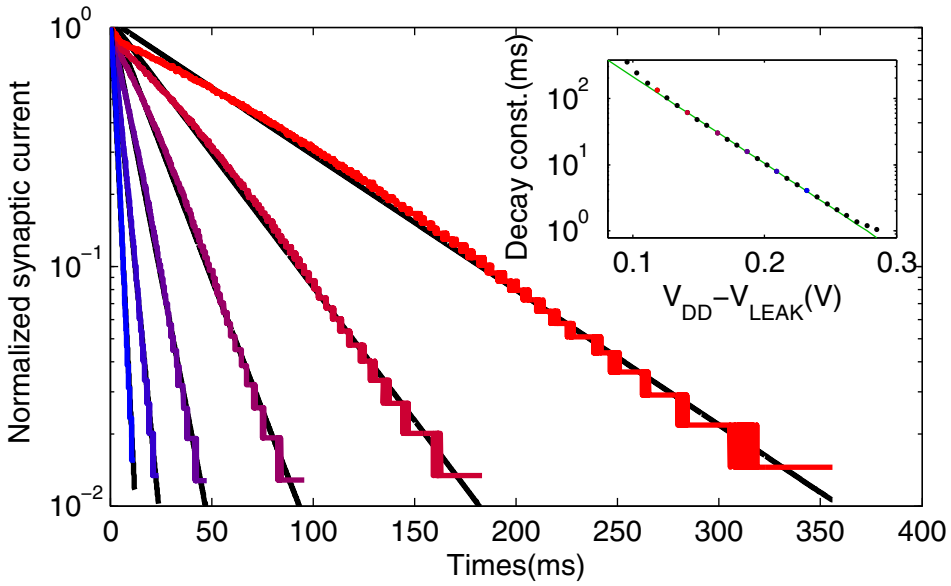

Figure 3.14: Synapse Decay Constant

The synapse responds to a spike with a low-pass filtered pulse, which decays exponentially after the pulse ends. I fit the logarithm of each decaying exponential, normalized by its peak, with a line, whose slope was the negative of the inverse of the decay constant. *Inset* Decay constant (dots) depends exponentially on the gate-source voltage of the cleft's leak transistor (M_{E2} in Figure 3.8), fit with a line.

which describes the synapse LPF, is linear. So I expect that if I stimulate the synapse with many spikes (further apart than the pulse width), they sum linear.

To evaluate the linearity of the synapse, I stimulate it with five spikes separated by 10ms (with a rise-time of 2ms). I compare the response to this burst of spikes to a single synaptic spike response, arithmetically summed five times, each copy 10ms apart (Figure 3.16). I observe that the arithmetic sum is nearly equal to the burst response, with its peak (190pA) exceeding the burst response's peak by only 9pA, a deviation of less than 5%.

To evaluate the linearity of the synapse to inputs that cause pulses to collide at the

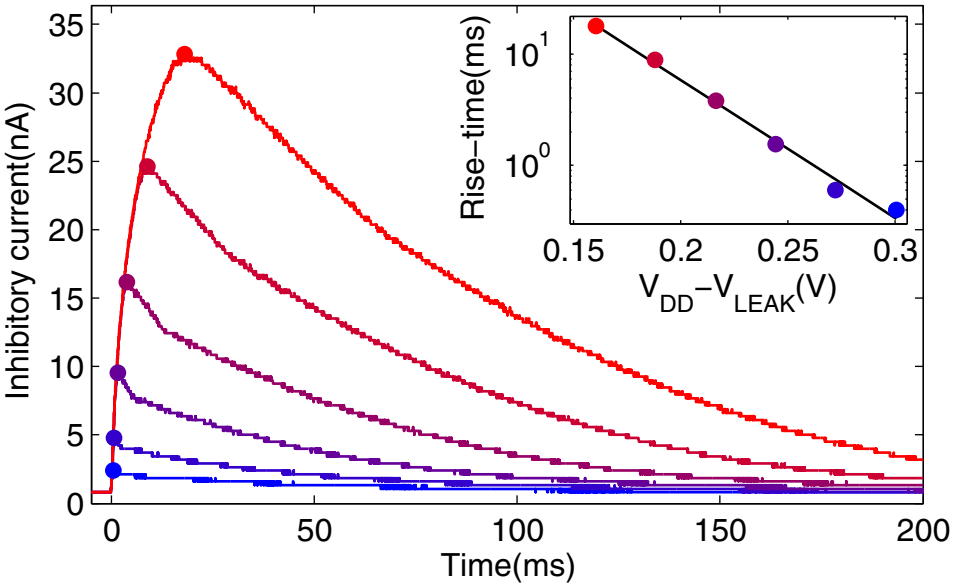

Figure 3.15: Synapse Rise-time

The synapse responds to a spike with a low-pass filtered pulse, which peaks at the end of the pulse (dots). *Inset* The time-to-peak (dots) depended exponentially on the gate-source voltage of the cleft's leak transistor (M_{E2} in Figure 3.8).

synaptic PE, I stimulate the synapse with Poisson-generated input spikes (Figure 3.17). I have verified that the synapse LPF is (nearly) linear. However, as mentioned in Section 3.3.2, its PE is nonlinear, capable of both facilitation (supralinear) and depression (sublinear). First, I stimulate the synapse with a 10Hz Poisson input: Because the input spikes are spread out in time (by at least 20ms in this instance), the pulses do not interact and the synaptic current sums linearly. Next, I stimulate the synapse with a 50Hz Poisson input: Many input interspike intervals are small (about 1ms), short enough for the pulses to interact, causing synaptic facilitation. Facilitation occurs when a spike arrives within 5ms the one before, although the rise-time (pulse width) is only 1ms. This interaction after the pulse termination is due to the pulse's tail, which is present milliseconds after the pulse (see Section 3.3.2). Then, I stimulate the synapse with a 20 spike burst at 20kHz: The input

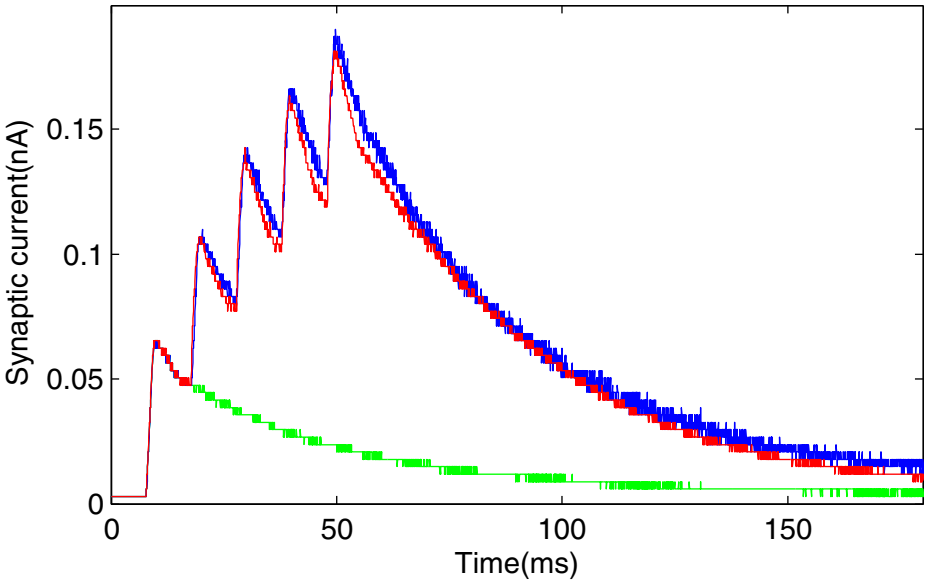

Figure 3.16: Synaptic Summation

In response to a single input, the synapse responds with a (nearly) linear rise-time and an exponential decay (green). In response to multiple inputs (10ms apart) the synapse sums the contribution of each (red). This behavior is similar to the arithmetic sum of the single spike response, shifted to match the input (blue). The decay constant and rise-time equal 45ms and 1ms, respectively.

interspike intervals are 50μs. The 20 rapid inputs saturate the PE, rendering it insensitive to further spikes and causing synaptic depression (see Section 3.3.2).

Although facilitation and depression exhibit useful properties, a silicon synapse can only represent one synapse in the regimes where they are present. Conversely, in the linear regime, one synapse can represent an entire population of independent synapses, whose currents sum to drive a soma (or dendrite).

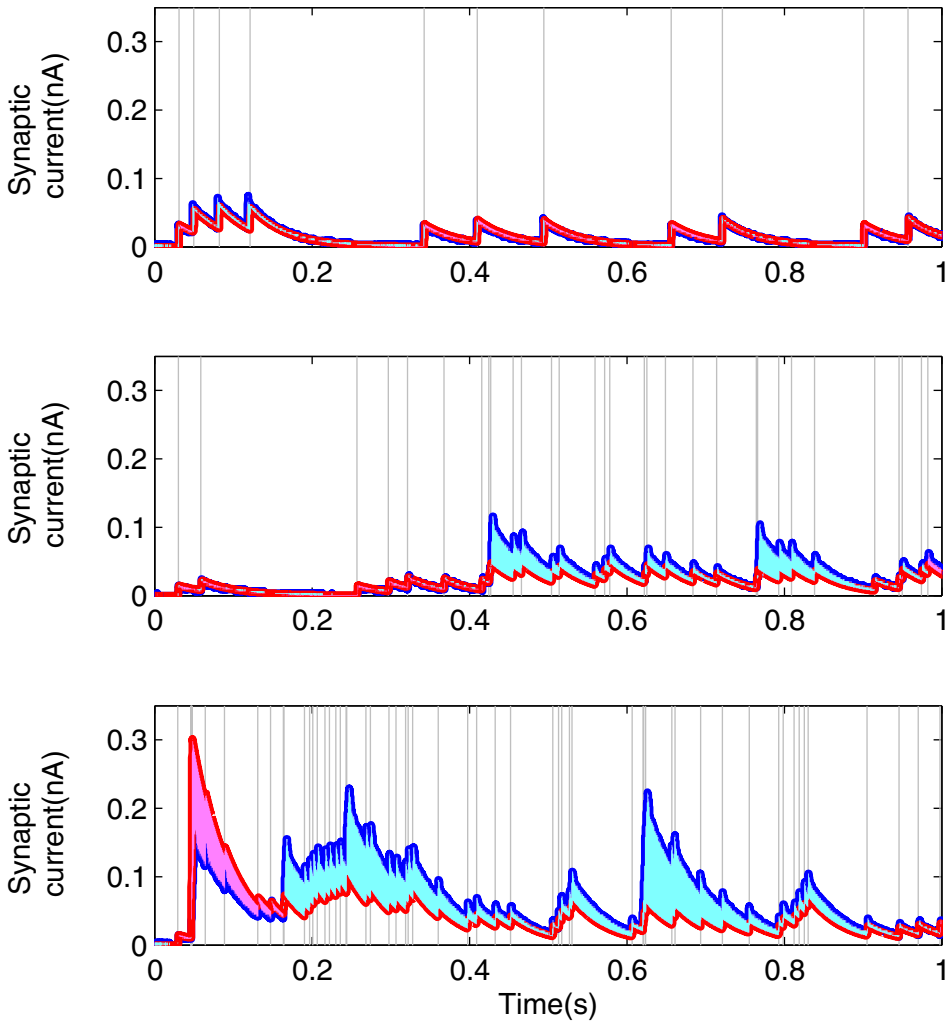

Figure 3.17: Synaptic Facilitation and Depression

Top With a Poisson-generated input at 10Hz, the synaptic current (blue) sums linearly, closely following the arithmetic sum of spike responses, generated by a single kinetic model (red), only slightly exceeding (cyan) or falling below (magenta) the model. *Middle*. With a Poisson-generated input at 50Hz, the synaptic current facilitates, exceeding the kinetic model, due to pulses overlapping. *Bottom* With a Poisson-generated input at 50Hz augmented with a 20 spike 20kHz burst (at time=0.0465s), the synaptic current initially depresses, due to the incoming inputs saturating, then facilitates. *All panels* Decay constant and rise-time equal 45ms and 1ms, respectively.

3.4.4 Phase-Response

The effect of synaptic excitation depended on the phase at which it occurred. I characterized this phase dependence in a pyramidal neuron (See Chapter 6 for schematic), which is just a soma (Figure 3.7) with an excitatory synapse (Figure 3.8). I activated the excitatory synapse at random points in the neuron's cycle, once every five cycles, observing the decrease in interspike interval (ISI). I repeated this process several hundred times and plotted the resulting PRC (Figure 3.18 *Top*). I set both the synaptic rise-time and decay constant to 1ms, resulting in a brief excitatory pulse.

The neuron was most sensitive to excitation between 10 and 20ms after it spiked (without synaptic excitation, its ISI was 24.7ms). In this sensitive region, each excitatory synaptic event subtracted 1-2ms from the neuron's ISI. During this phase of its cycle the neuron's membrane (current) was high, and the excitatory spike pushed it higher, activating the AH's positive feedback current (Figure 3.18 *Bottom*). On the other hand, excitation applied less than 10 or more than 20ms after it spiked subtracted less than 1ms from the neuron's ISI. During these phases, either the increase in soma potential was negated by the increased efficacy of the leak conductance at the higher potential or the AH's positive feedback was already active and stronger than the excitation, rendering it less effective. This is the primary deviation from the analytical solution, in which synaptic input does not compete with an AH: Instead, the neuron has a hard threshold (Figure 3.2).

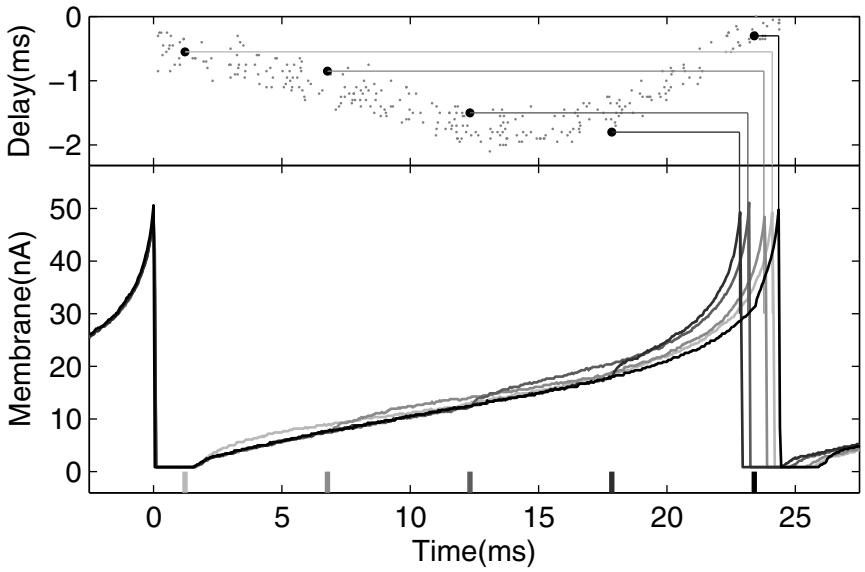

Figure 3.18: Neuron Phase-Response Curve

Bottom Membrane (current) traces of a neuron that I drove with a constant current and additionally excitation at various phases with a brief pulse; the decrease in interspike interval depended on when excitation occured (vertical bars). *Top* The phase-response curve shows that excitation is most effective between 10 and 20ms after the neuron spikes, subtracting 1-2ms to from its interspike interval (24.7ms).

3.5 Discussion

My log-domain neuron and its components are generally applicable to neuromorphic systems and I have used neurons with similar somas and synapses in another model of hippocampal associative memory in addition to the one described in this thesis [6, 5]. I have already used log-domain current-mode circuits to model multicompartment neurons with a soma coupled to many dendritic compartments [5]. In addition, other neuromorphic engineers applied similar log-domain circuits to model voltage-gated channels that influence spike timing and response reliability in a model of cochlear nucleus [155].

My circuits are similar in size and complexity to integrator-type LIFs with CMI synapses (all 1280 neurons use only about 13% of the chip's 10mm^2 area), while capable of modeling phenomena that depend on synaptic rise-time or shunting inhibition, such as synchrony by inhibition (Chapter 4). Previous conductance-based silicon neurons consumed too much area to fit more than a few copies per chip [96, 129].

The log-domain soma and synapses reproduce the behavior and the computation of biological neurons better than integrator-type LIFs and CMI synapses. The conductance-based soma's FI curve increases sublinearly with increasing current, similar to biological neurons, whereas the integrator-type neuron's FI curve increases linearly with increasing current. The current-mode synapse realizes first-order linear dynamics (exponential decay) with tunable rise-time, whereas the CMI synapse rises nearly instantaneously, decays inversely with time, and sums nonlinearly. Lastly, the conductance-based soma's PRC closely matches that of neocortical neurons, verifying the similarity between their temporal dynamics [124].

Chapter 4

Silicon Gamma Synchrony

In Chapter 2, I described my model of sequence memory in the hippocampal formation, alluding to the importance of the coherent gamma rhythm; and in Chapter 3, I presented my silicon neuron. This chapter describes the design and characterization of a network of silicon inhibitory interneurons that generates a gamma rhythm (30-100Hz).

Section 4.1 introduces synchrony in inhibitory networks, particularly in the hippocampus, which is known to actively generate gamma synchrony. Then in Section 4.2, I discuss previous models of synchrony, ranging from silicon to analytical to computational, and the insights into synchrony they provide, emphasizing two recent studies that combine analytical and computational aspects to demonstrate the delay model of synchrony (DMS). DMS posits that, in controlling inhibitory rhythmicity, the critical parameter is synaptic delay. Next, in Section 4.3, I show with a simple analytical model that synaptic rise-time can behave as a surrogate for pure synaptic delay in controlling network frequency. Then in Section 4.4, I switch focus and describe neuromorphic implementation of an inhibitory interneuron. In Section 4.5, I quantify the performance of the interneuron network in terms

of frequency and a measure of synchrony, showing that the chip's behavior is consistent with DMS. Finally in Section 4.7, I discuss the implications of the network performance at the system level as well as the limitations and nonidealities of the network and how they can be overcome in future chips.

4.1 Gamma Synchrony in the Hippocampal Formation

When observing the hippocampal local field potential in an awake, behaving animal, a salient feature is the rhythmicity, dominated by the theta and gamma rhythms. In this section, I focus on the hippocampal gamma rhythm (Figure 2.11), a well studied and characterized phenomenon, which has several features that may contribute to computation. The gamma rhythm is coherent across whole hippocampal regions, providing similar synchronizing signals to all pyramidal neurons within a region [20]. It is coherent between regions: always between CA3 and CA1 and transiently between the CA regions and the dentate gyrus, possibly gating information transfer between regions [31, 65]. Further, gamma may gate information transfer between the hippocampus and other forebrain structures, as during behavior gamma is observed simultaneously in multiple brain regions [46].

The gamma rhythm shapes pyramidal neuron activity in time, potentially influencing processing or plasticity [46]. According to one theory, it divides the theta oscillation into slots, where memories are temporarily stored [90]. As presented in Chapter 2 in my model of hippocampal function, the gamma rhythm enforces a phase shift between the dentate gyrus and CA3 regions to shape synaptic plasticity to form sequence memory. In addition to the hypothetical function of the gamma rhythm, many observations provide insights into how it is generated.

Evidence suggests that the hippocampus generates the gamma rhythm internally in the CA regions, mediated by $GABA_A$ synapses from interneurons. The internal generation of the gamma rhythm is established by lesion studies, in which subcortical inputs are removed, and the gamma rhythm remains [20]. Further, in the CA regions, severing the cortical input leaves the gamma rhythm intact [20]. The role of interneuron $GABA_A$ is verified by modulating extracellular chloride, which modulates the gamma rhythm; by observing current source densities, which show gabaergic inhibition of interneurons' (and pyramidal neurons') somas; and by recording interneurons that fire phase locked to the gamma rhythm [46]. Although the gamma rhythm is generated by interneurons via $GABA_A$, the mechanisms that these interneurons and their $GABA_A$ synapses employ to control the frequency—and manage variability—are not well understood. Therefore, to elucidate these mechanisms, many researchers have undertaken silicon, analytical, and computational studies.

4.2 Previous Models

In attempting to understand the generation of gamma synchrony by inhibitory interneurons it is useful to review related prior work. Ranging from silicon to analytical to computational, numerous studies have contributed to the understanding of how networks of inhibitory interneurons generate synchrony and rhythmicity (for a review see [41]). In the next section, I mention models of synchrony in silicon. In the following section, I review primarily analytical studies, which employ simple models to provide intuition into synchrony. Then, I review computational studies, which often employ elaborate models to study the behavior of these networks.

4.2.1 Silicon Models

Few studies of silicon synchrony have been published, reflecting the lack of synchrony in most neuromorphic systems. One study employs two nonspiking neurons to generate both membrane potential synchrony and antisynchrony depending on coupling strength [33]. Another study uses many simple integrate-and-fire (integrator-type) neurons, coupled with fast synapses, but the network only synchronizes weakly, requiring excitatory coupling among neurons to rescue it [92]. These silicon studies of synchrony have primarily shown that generation of synchrony is not trivial. We gain insight by looking at analytical and computational models.

4.2.2 Analytical Models

Theoretical studies focus mainly on idealized and simplified models to analytically track some aspect of synchrony and provide intuition into how synchronous models function. Simplified models are attractive because complete (and complex) models are intractable to analysis directly. Synchronous systems can be simplified in many ways to make them approachable, which include using: simplified interactions among interneurons, weak coupling, mild heterogeneity, simplified interneuron dynamics, and reduced system size.

These simplifying techniques have contributed greatly to understanding of synchronous systems. One of the first researchers to study synchronization in coupled oscillators was Kuramoto, who assumed an infinite number of oscillators and used simplified interactions among his heterogeneous population of oscillators to exactly solve for the degree of synchrony [82]. This work showed that increased coupling strength can compensate for heterogeneity, with appropriate oscillator interactions, demonstrating the importance of coupling

strength and oscillator phase response. Assuming weak coupling and a system of only two interneurons, Chow showed that inhibitory interneurons synchronize even with weak heterogeneity [28]. In addition, this work showed at what point weak heterogeneity overcomes weak coupling to break synchrony. With identical assumptions, Ermentrout and Gutkin showed that inhibition could still synchronize interneurons with various forms of spike frequency adaptation, implying self inhibition does not deleteriously affect synchrony [40]. Also, with identical assumptions to Chow, Lewis and Rinzel showed that inhibition, gap junctions, or both could generate synchrony [87].

4.2.3 Computational models

Unlike theoretical studies of synchrony, which tend to use simple models, computational studies usually use more elaborate models that better approximate known neurobiology. These modeling studies build large, semirealistic networks and use brute force to simulate the models in many parameter regimes. Often, such studies produce copious amounts of data and compelling demonstrations, but their conclusions are often difficult to interpret, especially when they contradict intuition. Unless all combinations of parameters are explored, which is clearly impossible, it is difficult to attribute behavior to one parameter or another. Even though computational studies include many known neurobiological elements, many other aspects of neurobiology are not known and are often arbitrarily specified. Further, some computational studies (for convenience and comparison) make the same assumptions as the analytical studies, such as weak coupling and mild heterogeneity. Nevertheless, these studies have greatly contributed to understanding of interneuron synchrony.

Computational studies have supported theoretical results and have made predictions

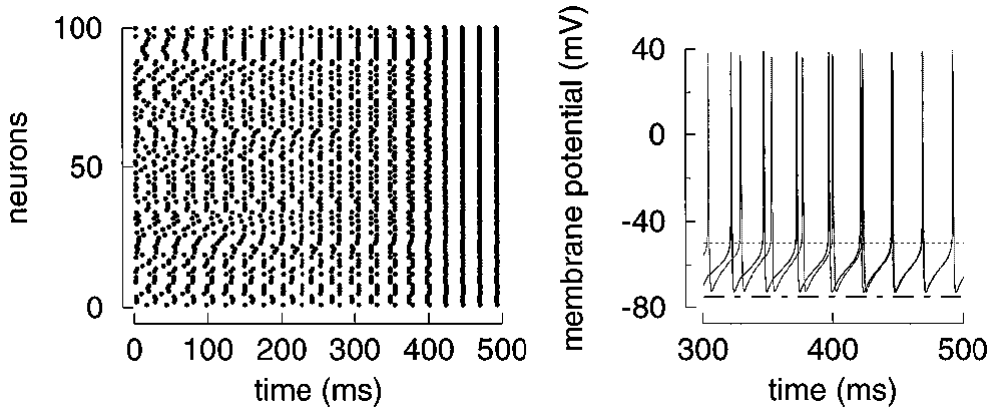

Figure 4.1: Synchrony in Simulated Interneurons

A population of simulated interneurons with weak inhibitory coupling and constant input current synchronize without excitatory connections. From [146].

about neurobiological function. Wang and Buzsaki simulated one of the first semirealistic inhibitory interneuron networks, demonstrating that a network composed solely of interneurons could synchronize (Figure 4.1) [146]. They showed robust synchrony across (global) parameter variation but sensitivity to heterogeneity. White et al. simulated small inhibitory interneuron networks (between 2 to 10 interneurons), showing the behavior of these systems for various parameter sets and concluding that synchrony is fragile in heterogeneous networks [151]. Tiesinga and Jose simulated large networks (up to 1000 interneurons) with significant heterogeneity and synaptic noise [140]. They showed that strong synchrony (where every interneuron fires every cycle) is inherently unstable and that stochastic weak synchrony (where every interneuron fires probabilistically in every cycle) is largely stable.

In subjecting their model to experimental verification, neurobiological observations revealed potential problems with many analytical and computational studies' assumptions and questioned their dogma. Supported by many studies, the dogma maintained that net-

work frequency was determined by the gabaergic decay constant, which had to be greater than 5ms to achieve synchrony [146, 151, 28]. However, observations put the gabaergic decay constant in the hippocampal formation between 1 and 5ms, too fast for the simulations to synchronize within the gamma regime [10, 52]. The dogma was wrong. Two groups independently recognized the problem and sought to solve it.

Recently, the two groups published their computational studies, which showed that the key parameter in determining the frequency of an inhibitory interneuron network's rhythmicity is the combined axonal and synaptic delay, the effective delay. To provide intuition into the numerical results, both studies augmented simulation with simple analysis. Wang and Brunel performed the first study, which explicitly assumed that the network activity and network inhibition must be 180 degrees out of phase (Figure 4.2) [21]. Using this assumption, they showed that the effective delay is the dominant contributor to the 180 degree phase difference. Further, they showed that in many regimes the network frequency varies inversely with the effective delay, consistent with their simulations.

Maex and De Schutter performed the second study, which employed a simple delay differential equation to describe the network activity and solve for its periodic solution [94]. They demonstrated that the analytical solution was consistent with their simulations, which showed the network period was close to four times the effective delay and only weakly dependent on the gabaergic decay constant (Figure 4.3). In addition, they demonstrated that the network is robust to the presence of mild noise and heterogeneity. These two studies corrected numerous previous studies that assumed delay was unimportant to synchrony. They establish unequivocally that delay is the critical parameter influencing the frequency of synchrony—the delay model of synchrony (DMS).

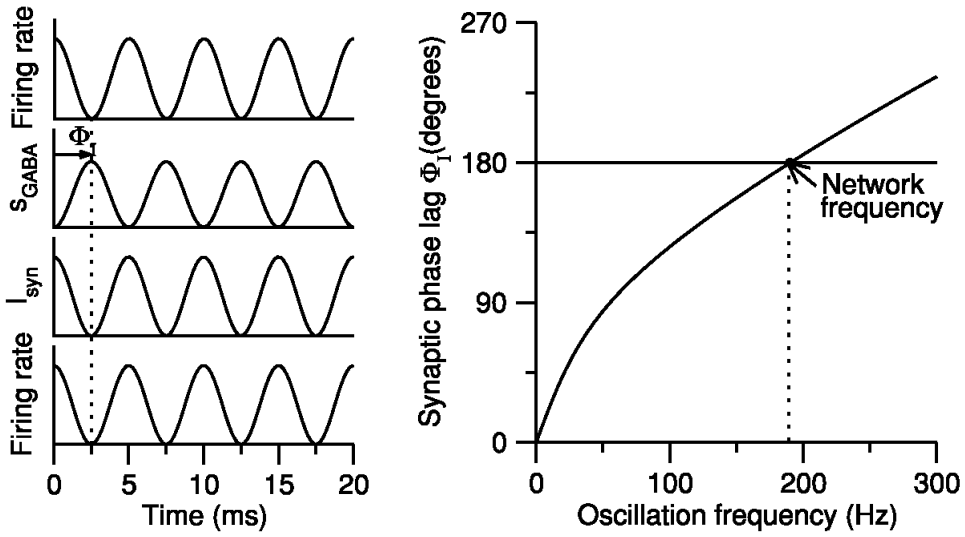

Figure 4.2: Phase Relationship Between Interneuron Activity and Inhibition

For synchrony to occur the oscillatory component of the $GABA_A$ activation must be 180 degrees out of phase with the spiking activity. *Left* Synchrony is a self consistency problem: The firing rate determines the $GABA_A$ activations, which influences the inhibitory synaptic current, determining the firing rate. *Right* The phase lag of the inhibition relative to the network firing rate depends on frequency. When the phase lag equals 180 degrees, synchrony is stable. From [21].

4.3 Including Synaptic Rise-Time

The delay model of synchrony (DMS) qualitatively and quantitatively describes the synchronous behavior of populations of interneurons with axonal and synaptic delay, pure delay. In this section, I extend DMS to include synaptic rise-time in the effective delay. Specifically, I show that the effective delay of a synapse whose impulse response (i.e., spike response) is a low-pass filtered pulse is equal to half the pulse width. Following the argument of Maex and De Schutter [94], I show that the network period resides between two and four times the effective delay, which is between one and two times the pulse width.

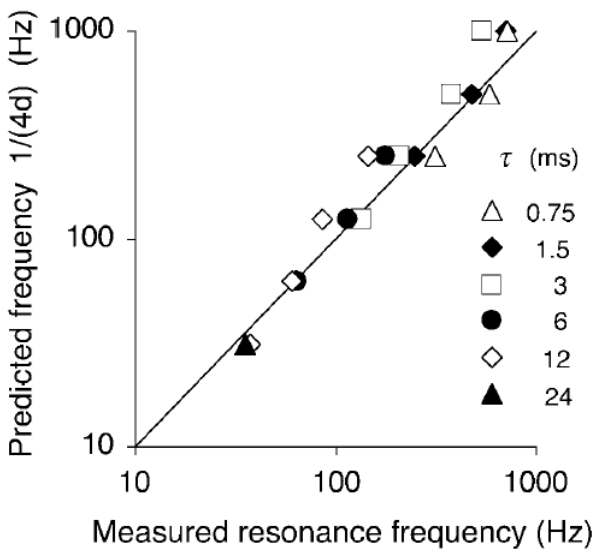

Figure 4.3: Predicted Versus Simulated Frequency

The Delay Model of Synchrony predicts the network period is approximately four times the effective synaptic delay. Simulations show that for different $GABA_A$ time constants the network maintains this relationship. From [94].

Consider a population of inhibitory interneurons whose activity, $A(t)$, is modeled by the following delay differential equation:

$$\lambda \dot{A(t)} = -A(t) + F(I - k(t) * A(t)) \tag{4.1}$$

where I is the constant input current, $k(t)$ is the synaptic impulse response, F is the neuronal transfer function, λ is the membrane time constant, and $*$ represents convolution in time. λ can be instantaneous when the feedback inhibition (or any input) is fast [49], therefore, I set it to zero, which is equivalent to assuming the network activity is 180 degrees out

of phase with the inhibition [21]. I wish to determine if a synchronous solution exists. To this end, I linearize the neuronal transfer function about the asynchronous state, $A(t) = C$, in which neuronal firing is considered constant, which gives:

$$0 = -y(t) - ak(t) * A(t) \tag{4.2}$$

where $y(t) = A(t) - C$ is the difference of network activity from the asynchronous state, and a is the linearized neuronal transfer function at the asynchronous state.

Next, I assume $y(t) = e^{i\omega t}$ ($\omega = 2\pi f$, where f is the oscillation frequency) and transform 4.2 into the frequency domain.

$$0 = -Y(\omega) - aY(\omega)K(\omega) \tag{4.3}$$

I assume the synaptic impulse response is a convolution of a pulse of width T_p and an exponential [35], which in the frequency domain is represented by

$$K(\omega) = 2dB\text{sinc}(\omega d)e^{-i\omega d} \times \frac{G}{(\frac{1}{\tau} + i\omega)} \tag{4.4}$$

where τ is the synaptic decay constant, $d = \frac{T_p}{2}$ is the effective synaptic delay, B is the pulse amplitude, and G is the exponential amplitude. The pulse provides effective delay (whereas the convolution of two exponentials would not).

I substitute this result into 4.3 and separate it into two equations, one for the real part and the other for the imaginary part, using the identity $e^{ix} = \cos(x) + i\sin(x)$, which gives:

$$\begin{aligned} \sin(\omega d) &= \frac{\omega\tau}{2ad\tau BG\text{sinc}(\omega d)} \\ \cos(\omega d) &= \frac{-1}{2ad\tau BG\text{sinc}(\omega d)} \end{aligned} \tag{4.5}$$

Assuming a solution exists (that is the magnitude of the denominators is greater than or equal to the magnitude of its respective numerator), the two equations can be combined to obtain:

$$\tan(\omega d) = -\omega\tau \tag{4.6}$$

This is identical to the relation for a synapse with a (purely) delayed exponential impulse response (similar to [94]). Since $\tan(x)$ is negative for $\frac{\pi}{2} < x < \pi$, if $\tau \gg d$, $\omega d \Rightarrow \frac{\pi}{2}$, Equation 4.6 yields $T = 4d$, where T is the network period. Conversely, if $\tau \ll d$, $\omega d \Rightarrow \pi$, yielding $T = 2d$. Therefore, T depends strongly on d ($2d < T < 4d$), which provides its absolute minimum and maximum, and weakly on τ, which determines the value it takes within that range. Relate the network period back to the pulse width, $T_p < T < 2T_p$. Thus, I have extended the model put forth by [94] to include the synaptic rise-time (using a pulse), which describes the behavior of my a silicon inhibitory interneuron network that uses nonzero synaptic rise-time instead of pure synaptic delay to synchronize.

4.4 Interneuron Circuit

Similar to the neuron circuit in Chapter 3, I construct the interneuron from two circuit modules based on log-domain LPFs [47]: the soma and the synapse (Figure 4.4). This neuron is different in that it excludes the PE-LPF excitatory synapse and includes a PE excitatory synapse and a PE-LPF inhibitory synapse configured to inhibit itself and its neighbors through a diffusor, which spreads the synaptic current to neighboring silicon interneurons [15]. To test the silicon interneurons, I use the same setup that I used to test the silicon neuron (Figure 3.10).

4.4.1 Phase-Response Curve

In characterizing the interneuron, I focus on the inhibitory PRC; the excitatory effect of a similar synapse circuit on a similar soma circuit was explored in Chapter 3. The effect of synaptic inhibition depended on the phase at which it occured. I characterize this phase dependence by inhibiting the interneuron at a random point in its cycle, once every five cycles, observing the increase in interspike interval (ISI). I repeated this process several hundred times and plotted the resulting PRC (Figure 4.5 *Top*). The rise-time was set to 1.5ms and the synaptic decay constant was 5ms, as found in biology [10, 52].

The interneuron was most sensitive to inhibition between 15 and 30ms after it spiked (its uninhibited ISI was 38ms). In this sensitive region, each inhibitory spike added more than 8ms to the interneuron's ISI. During this phase of its spiking cycle the interneuron's membrane (current) was high, resulting in more effective shunting inhibition (Figure 4.5 *Bottom*). On the other hand, inhibition applied less than 5 or more than 32ms after it spiked added less than 4ms to the interneuron's ISI. During these phases, either its membrane

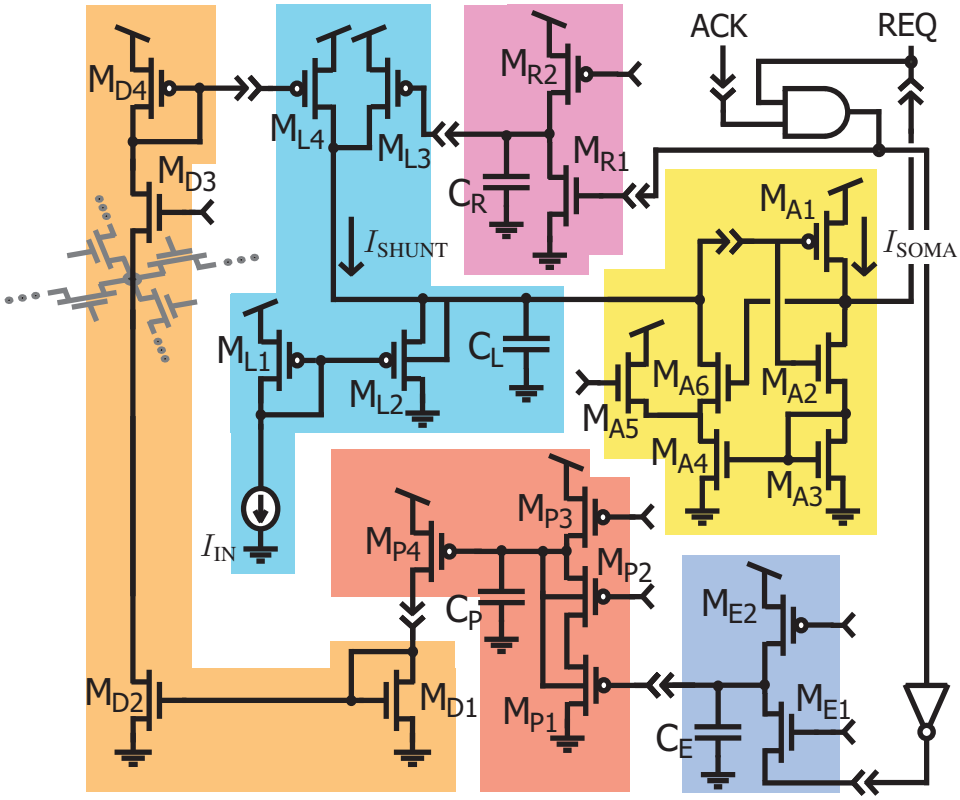

Figure 4.4: Interneuron Circuit

The interneuron circuit comprises two modules, both based on log-domain low-pass filters: the soma and the synapse. The soma includes the membrane ($M_{L1\text{-}4}$), the axon-hillock ($M_{A1\text{-}6}$), and the refractory channel population ($M_{R1\text{-}2}$). The synapse includes the cleft ($M_{E1\text{-}2}$) and the receptor ($M_{P1\text{-}4}$). A diffusor with current mirrors spreads inhibition to neighboring interneurons ($M_{D1\text{-}4}$). Single arrows represent bias voltages; double arrows represent inputs and outputs. The AND gate resets the interneuron's spike (TREQ), gated by TACK.

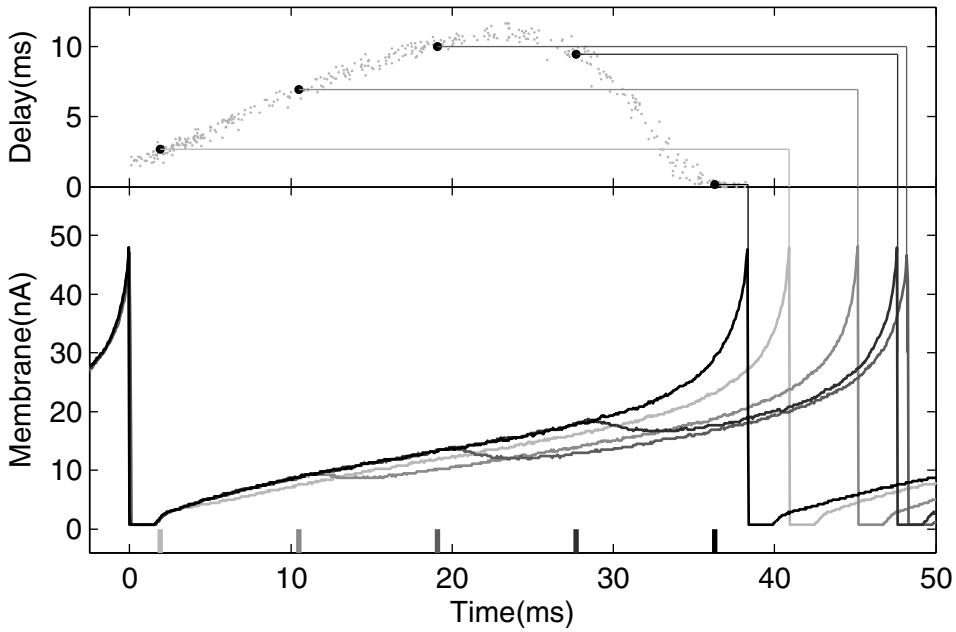

Figure 4.5: Interneuron Phase Response Curve

Bottom Membrane (current) traces of an interneuron driven with a constant current and inhibited at various phases; the increase in interspike interval depended on when inhibition occured (vertical bars). *Top* The phase-response curve shows that inhibition is most effective between 15 and 30ms after the interneuron spikes, adding about 8ms to its interspike interval (38ms).

potential was low, so shunting inhibition was less effective, or the inhibition did not have time to rise to its peak effectiveness. And near the cycle's end, the positive feedback from the axon-hillock (AH) turned on, overpowering the inhibition.

4.5 Interneuron Network

In the previous section, I described the interneuron circuit, which, in this section, I use in the construction of an inhibitory interneuron network. The purpose of the interneuron net-

work is to synchronize in the gamma frequency range with the ultimate goal of entraining pyramidal neurons. Further, I demonstrate the role of delay (implemented in the form of gabaergic rise-time) in enabling synchrony and in determining network frequency.

4.5.1 Experimental Setup

The inhibitory interneuron network resides on the STDP Chip and is composed of 256 interneuron circuits (arranged in a 16 by 16 grid), all of which are used in each experiment. Each interneuron inhibits its neighbors through a diffusor on a square grid. The diffusor can be tuned to no spread, such that an active interneuron receives most of the inhibition it puts into the diffusor and the rest of the interneurons receive negligible inhibition, or it can be tuned to a global spread, such that every interneuron receives approximately the same inhibition. In all experiments (except when noted) the inhibition was tuned to be global[1]. Each interneuron received a constant excitatory input current (except where noted). This current was realized by using the ampaergic synapses outside of their intended operating regime. I tuned the pulse-width to be long (tens of milliseconds) and stimulate the ampaergic synapse at a shorter interval than this (100s of Hz), extending the pulse indefinitely and creating a constant excitatory current.

4.5.2 Quantifying Synchrony

Since the purpose of the network is robust synchronization, to evaluate its performance quantitatively I need a measure of synchrony. Several measures have been used to quantify synchrony [146, 147, 51]. In numerical studies, heterogeneity is usually minimal, therefore,

[1]To ensure global spread of inhibition, I set the lateral (NMOS) diffusor bias 500mV higher than the horizontal bias (M_{D3}).

all interneurons fire at about the same rate and suppression of interneurons is rare. In these systems there is qualitatively little difference between the prevalent synchrony measures. However, in silicon, where heterogeneity is abundant and suppression common, different synchrony measures can give quantitatively and qualitatively distinct results. I considered three measures to quantify the synchrony of the interneuron network: synchrony index (SI), coherence index (CI), and vector strength (VS). For the silicon interneuron network every measure has advantages and disadvantages.

SI is the normalized average cross correlation of all pairs of interneurons in the network [146]. The normalized cross correlation between interneurons x and y is defined as:

$$\kappa_{xy} = \frac{\sum_{l=1}^{K} X(l)Y(l)}{\sqrt{\sum_{l=1}^{K} X(l) \sum_{l=1}^{K} Y(l)}} \tag{4.7}$$

where $X(l)$ and $Y(l)$ are the binned spike trains of the pair of interneurons. SI is the mean of all κ_{xy}. SI has some desirable properties. It is naturally comparable to observed cross correlations from pairs of recorded interneurons in vivo or in vitro. Since it is normalized, its values range between zero and one. For values near zero, interneurons are asynchronous, whereas for values near one, interneurons are synchronous. SI also has undesirable properties. Its value is strongly dependent on bin size, which is usually chosen to be one-tenth of the network period. Its value is reduced by interneurons that do not fire or fire slower than the network frequency, a common occurrence in silicon. Its value is increased by interneurons that fire faster than the network frequency, also common in silicon, where some interneurons fire several times faster than the average. Because of these undesirable properties SI is not a good measure for synchrony in a silicon (or any highly heterogeneous) system.

CI is the standard deviation divided by the mean rate [147]: The mean rate is computed from the time histogram of all spikes in the network. First, I measure the network activity by calculating the instantaneous population firing rate, $R(t_i)$:

$$R(t_i) = \frac{\text{total number of spikes in } (t_i, t_i + \Delta t)}{N\Delta t} \tag{4.8}$$

where N is the number of interneurons and time is divided into small bins (Δt =1 ms). Then we compute the mean and standard deviation of $R(t_i)$. CI has some desirable properties. It is comparable to the coherence of local field potentials, observed in biology. It is independent of the number of interneurons that fire, whether they fire faster or slower than the network frequency. CI also has undesirable properties. Its value is not restricted to be between zero and one. Like SI, its value is strongly dependent on bin size. Its independence on the number of interneurons that fire gives it preference to one single interneuron firing, which is perfectly synchronous with itself. Therefore, CI rewards suppression of interneurons. To fairly portray the network state, CI must be augmented by the number of interneurons in the network that fire.

VS is the normalized vector sum of the average number of spikes at each phase of the (gamma) cycle [51]. Including all interneurons in the network, I create a period histogram of N=50 bins at the measured network period (see next section). Using the period histogram, I compute the normalized vector strength as:

$$VS = \frac{\sqrt{(\sum_{i=1}^{\mathrm{N}} X_i)^2 + (\sum_{i=1}^{\mathrm{N}} Y_i)^2}}{\sum_{i=1}^{\mathrm{N}} T_i} \tag{4.9}$$

where T_i is the period histogram's ith bin, with $X_i = T_i \cos(\frac{2\pi(i-1)}{\mathrm{N}})$ and $Y_i = T_i \sin(\frac{2\pi(i-1)}{\mathrm{N}})$. VS has desirable properties. Like CI, it is comparable to the coherence of the local field potential; it is independent of the number of interneurons that fire and their rates; it is only weakly dependent on bin size. VS also has undesirable properties. Like CI, it rewards suppressing interneurons, and therefore, must be augmented by the number of active interneurons, and it punishes nonconstant network frequency. If the network frequency shifts, or if the network experiences a phase shift, VS decreases.

Each synchrony measure has advantages and disadvantages. For the interneuron network the best choice is VS as it is only weakly dependent on bin size and its values ranges between zero and one. Further, its disadvantages can be controlled. Considering the number of active interneurons compensates for its tendency to reward interneuron suppression, and using periods of data where the frequency is nearly constant alleviates the punishment of frequency shifts.

4.5.3 Computing Network Frequency and Period

I define the network frequency as the frequency component with the greatest power. To determine the frequency component with the maximum power, I bin all spikes recorded over a period of 250ms (1ms bins) to obtain $R(t_i)$ as before in Equation 4.8. I filtered the binned activity with a gaussian low-pass filter (corner frequency of 250Hz) to suppress harmonics and subtract the mean to remove the DC component. I converted the filtered activity from the time domain to frequency domain with a Discrete Fourier Transform (16,384 points), employing a Hamming Window (128 points) to reduce spectral leakage between frequency components [3]. The network period is the inverse of the strongest

network frequency.

4.6 Results

The inhibitory interneuron network required mutual inhibition to synchronize (Figure 4.6). When the interneurons were uncoupled from their neighbors, such that all of the inhibition is self inhibition (like a potassium channel population), they spike asynchronously. Each interneuron spiked at is own rate (Figure 4.6 *Left*). On the other hand, when the interneurons were coupled globally, so all inhibition was (approximately) evenly distributed to each interneuron in the network, they synchronized at a network frequency of 38Hz (Figure 4.6 *Right*) with an input current of 31nA[2].

In addition to synchrony, there were several differences between the uncoupled and coupled networks' behaviors. In the uncoupled case all interneurons spiked, since no other interneurons could inhibit them, whereas in the coupled case more than half of the interneurons (136) were suppressed by the more excitable ones. In the uncoupled network the average firing rate was 42Hz versus 38Hz for the coupled network (excluding suppressed interneurons). This result is contrary to intuition: The coupled network consisted of more excitable interneurons on average and received less inhibition (inhibition from 120 interneurons is shared among 256 versus one to one in the uncoupled cased). However, interneurons that inhibited themselves received inhibition at the least effective region of the PRC (see Figure 4.5), immediately after they spiked, whereas interneurons that inhib-

[2]To estimate the input current to an interneuron, I measured its amplified soma current ($I_{\mathrm{SOMA}} = \frac{I_{\mathrm{IN}}\,\mathrm{I}_0}{I_{\mathrm{SHUNT}}}$) through a current-out pad, which yielded I_{IN} when I_{SHUNT}=I_0. Hence, I disabled the inhibitory synapse (by lowering its synapse strength); I also disabled the axon-hillock (by raising its spike threshold). I estimated the input current as the pad current divided by the pad amplifier's gain (3564). Because the gain decreased for currents above 34nA, I fit lower values with an exponential to extrapolate the input current.

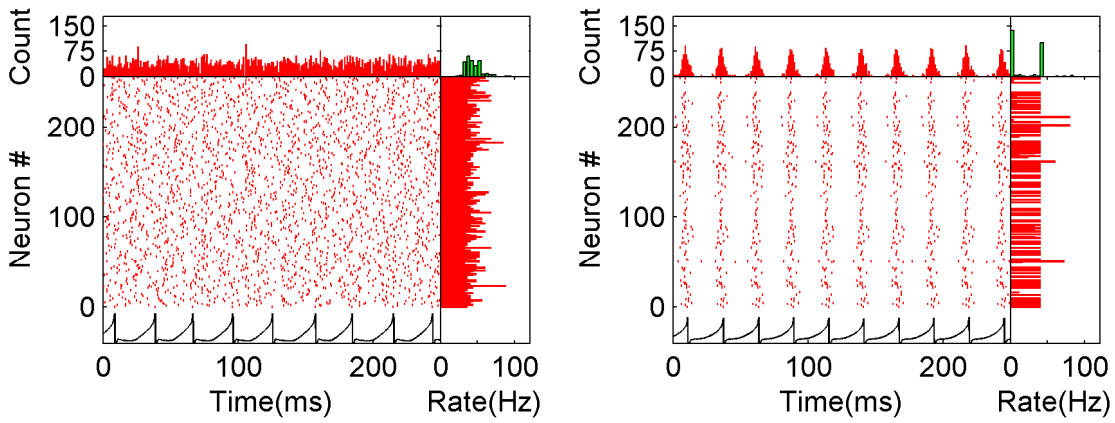

Figure 4.6: Synchrony Depends on Mutual Inhibition.

The lower left quadrant of both panels is a rastergram of the network activity and membrane potential (current) of one representative interneuron; each line represents a spike. The upper left is the histogram in time of all spikes in the network. The lower right is a histogram of firing rates across the interneuron population. The upper right is a histogram of the rates of all interneurons in the population. *Left* The network of inhibitory interneurons fails to synchronize when each neuron inhibits only itself (VS$= 0.06$). *Right* The network synchronizes when each interneuron inhibits itself and every other interneuron (VS$= 0.83$).

ited each other received inhibition at several points of the PRC (unless the synchrony was perfect), resulting in more effective inhibition.

Clearly, mutual inhibition can effectively generate synchrony in a silicon interneuron network. However, this synchrony must break down at some point. Based on numerical and analytical studies, I expect that the critical parameter in the generation and pacing of synchrony is the gabaergic rise-time, the effective delay as shown mathematically in Section 4.3.

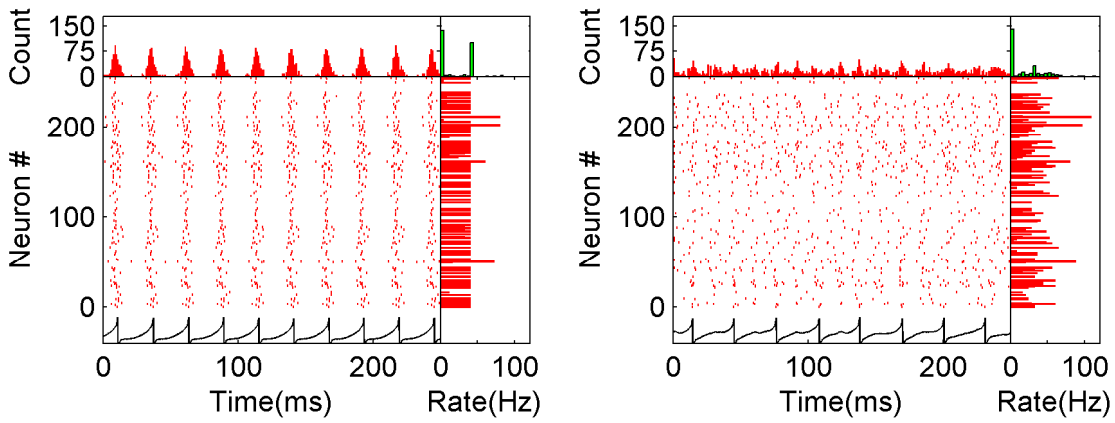

Figure 4.7: Synchrony Depends on Rise-time

Left The network of inhibitory interneurons synchronizes when the $GABA_A$ synapse has a several (11.7) ms rise-time (VS$= 0.83$). *Right* Conversely, the network fails to synchronize when the gabaergic rise-time is a fraction of a ms (VS$= 0.18$). The conventions are the same as Figure 4.6

4.6.1 Gabaergic Rise-time

The inhibitory interneuron network required a finite gabaergic rise-time to synchronize (Figure 4.7). When the interneuron network had gabaergic synapses with a finite rise-time (11.7ms) it synchronized (Figure 4.7 *Left*). Conversely, when the gabaergic synapses had an infinitesimal rise-time (0.1ms), the interneuron network failed to synchronize (Figure 4.7 *Right*), even when the inhibition was increased to the same level. That is, such that interneurons spiked in approximately the same numbers and at the same average rate in both cases: 120 neurons fired at an average of 38Hz with a finite rise-time; 114 neurons fired at average rate of 36Hz with an infinitesimal rise-time. Clearly, for the silicon interneuron network to synchronize, two requirements must be met: mutual gabaergic inhibition and nonzero gabaergic rise-time.

To confirm the gabaergic rise-time's pivotal role in synchrony, I varied it and measured

the network period (the inverse of the strongest frequency). The network period was one to two times the rise-time, depending on the fall-time (i.e., gabaergic decay constant), plus an offset, caused by the AH's positive feedback overpowering inhibition shortly before a spike. With a rise-time of 11.7ms, and a gabaergic decay constant of 5ms (same as Figure 4.7), the network period (24.2ms), minus an offset (7.3ms), was 1.44 times the rise-time (Figure. 4.8), which corresponds to an network period of 2.88 times the effective delay from Section 4.3 (which equals half of the rise-time). This result is consistent with the study of Maex and De Schutter, who predicted that the network period should be between two and four times the synaptic delay [94]. In silicon, this same proportionality constant yielded a good fit for a wide range of rise-times ranging from 7 to 100ms, even though the network was only synchronous (VS > 0.5) for rise-times between 7 and 58ms. On the low frequency end, the relationship broke down as the rise-time increased beyond 60ms; inhibition grew too strong to maintain even weak synchrony, forming a winner-take-all network. On the high frequency end, the relationship broke down due to two factors: Interneurons were unable to maintain the high network frequency, due to insufficient input, and as the rise-time dropped to levels comparable with the decay constant, the linear relation ceases to be valid as described by Equation 4.6.

The AH's offset had the effect of increasing the rise-time by 5.1ms (7.3ms $\div$ 1.44), which corresponds to an AH delay of about 2.6ms, since the rise-time is twice the effective delay. This is expected since the AH takes several milliseconds to fully depolarize the soma (and send the spike off chip) once positive-feedback takes over. This effect is visible in the PRC; at the end of its period, the interneuron was resistant to inhibition (Figure 4.5). The positive feedback's speed depends on how far above threshold the input current drives the soma and therefore, on the magnitude of the input current (Chapter 3).

The interneuron network synchronized strongly for an intermediate range of rise-times.

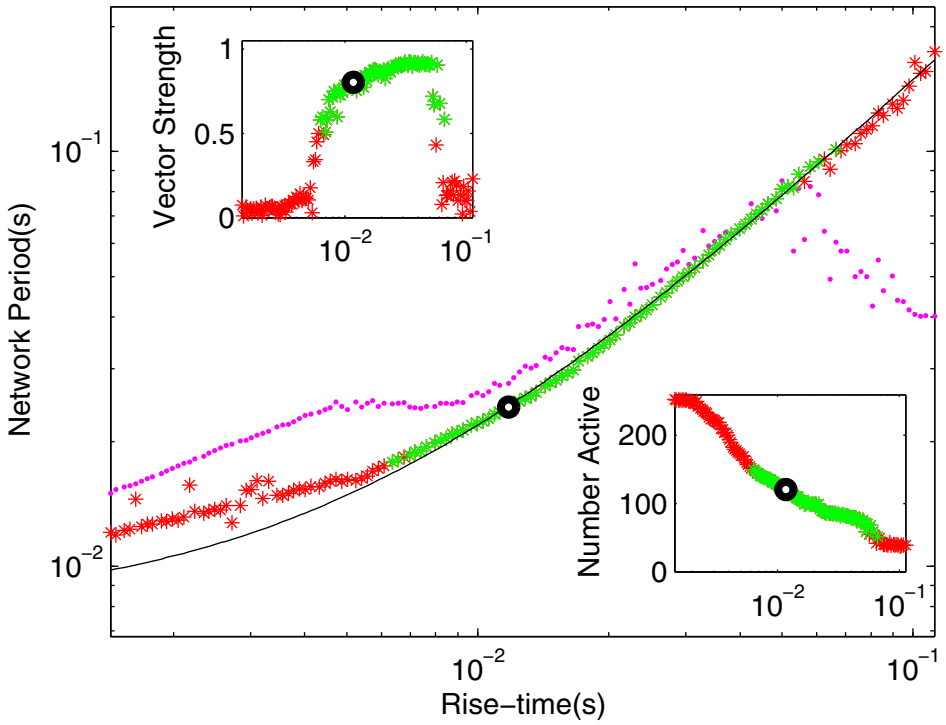

Figure 4.8: Network Period Versus Rise-time

The network period (red asterisks) increases linearly with the rise-time (black line; fit from 7 to 58ms). The network period ceases to be linear and saturates when the rise-time is small (below 7ms). The magenta dots show the mean interneuron period (the inverse of the number of spikes divided by NAI divided by the measurement duration, 250ms). *Left inset* Vector strength peaks for a rise-time of about 25ms. *Right inset* Number of active interneurons decreases as rise-time increases. For all panels, green asterisks signify highly synchronous network states with vectors strength above 0.5 and greater than 50 active interneurons. The black circle is the baseline setting; it has the same parameter values across figures.

VS peaked for a rise-time of about 25ms at this parameter setting (Figure 4.8 *Left inset*). The network was strongly synchronous (VS above 0.5) for rise-times between 10 and 40ms; also, in this regime the mean of active interneurons' spike frequencies (all spikes divided by NAI divided by the measurement duration, 250ms) was closest to the network frequency. The network was asynchronous for very fast rise-times (Figure 4.7 *Right*), and additionally in this case the fast rise-time resulted in reduced inhibition, contributing to the asynchrony. Also, the network was asynchronous for slow rise-times. This agrees with intuition, since the typical active interneuron period ranges from 30 to 40ms (33 to 25Hz). Therefore, if the rise-time (pulse width) exceeds 50ms, the interneurons perceive constant inhibition, which just reduces their firing rates rather than motivating synchrony.

Extended and stronger inhibition due to longer rise-times suppressed more interneurons. NAI decreased as the rise-time increased (Figure 4.8 *Right inset*). NAI reached its lower limit at about 50; it did not continue to decrease since the rise-time was much longer than the decay constant, enabling inhibition to reach its maximum (set by the gabaergic strength). Therefore, further increasing the rise-time did not increase the maximum level of inhibition to the network.

4.6.2 Other Gabaergic Parameters

I have verified that the gabaergic rise-time directly affected the network period. But because changing the rise-time changes both the inhibitory delay and amplitude (Figure 4.7), it is unclear which affect influenced the period. If the change in network period was caused by the change in inhibitory delay, I expect that changing the gabaergic strength (with a fixed rise-time) would not change the network period.

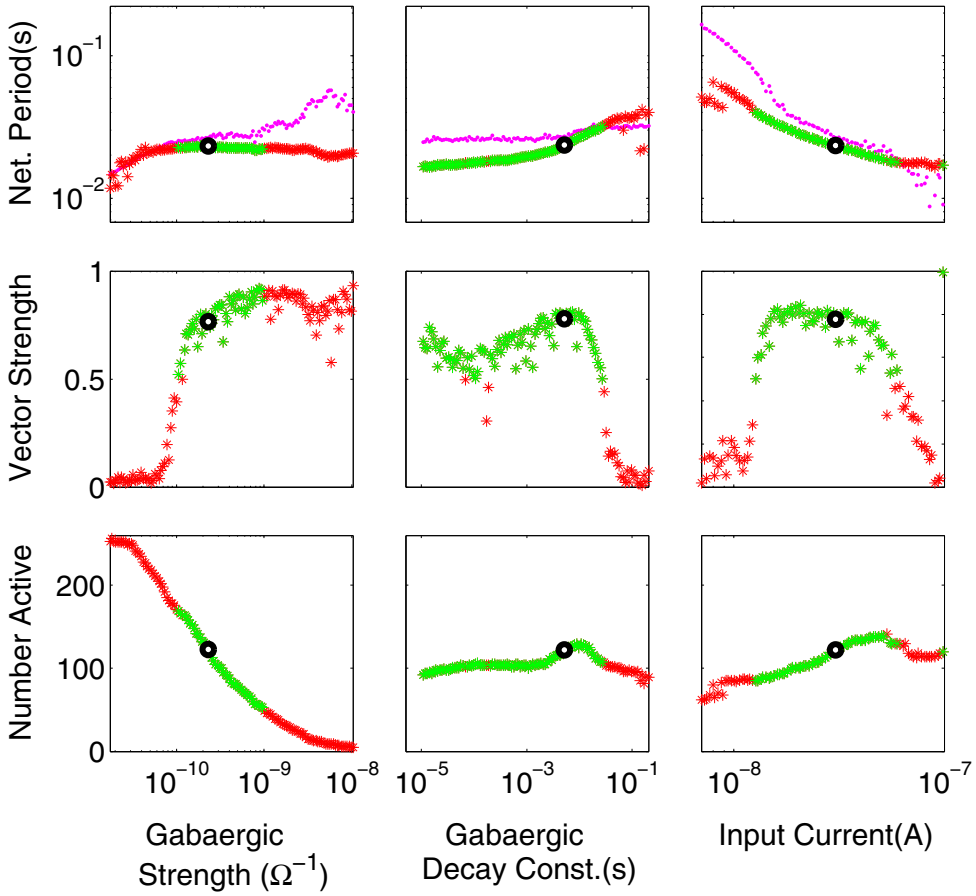

Figure 4.9: Delay Model of Synchrony Controls

Gabaergic strength, gabaergic decay constant and input current influence network period as well as the vector strength and number of active interneurons, but none affect the network period as strongly as the gabaergic rise-time. Conventions are the same as in Figure 4.8.

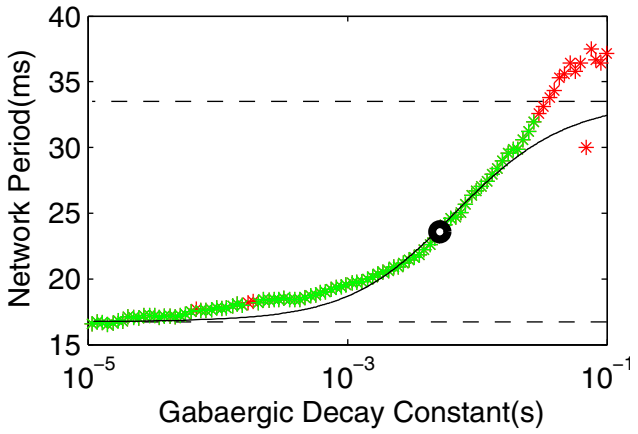

Figure 4.10: The Gabaergic Decay Constant

The gabaergic decay constant modulates network period (asterisks) in the range of two to four times the effective delay (dashed lines). For synchronous network states (light asterisks), the network period is similar to the predicted value (solid line) with an effective delay of 8.4ms (fit from Figure 4.8).

To determine the gabaergic strength's influence on synchrony, I varied it and measured the network period.[3] The network period was nearly constant, within 2ms (7%), even though I varied the gabaergic strength nearly an order of magnitude (Figure 4.9 *Left column*). Therefore, I conclude that rise-time affects the network period by changing the inhibitory delay, supporting DMS.

The gabaergic strength did not change the network period but it had a strong effect on VS and number of active interneurons (NAI). A small gabaergic strength (less than $0.11\text{G}\Omega^{-1}$) resulted in asynchronous spiking, because interneurons only weakly interact and were, therefore, unable to entrain each other. As the gabaergic strength increased, VS increased, until it reaches a critical point (about $1.0\text{G}\Omega^{-1}$) where NAI became low, and the

[3]To estimate the gabaergic strength, I measured the synapse's maximum current (achieved by stimulating the synapse fast enough to keep M_{P1} on) through a current-out pad, divided by the pad amplifier's gain (3564), which I fit to an exponential for various gate voltages on M_{P2}, obtaining I_{MAX} and κ values, which I used to calculate the equivalent conductance, $\frac{\kappa\, I_{\text{MAX}}}{\text{U}_\text{T}}$, where $\text{U}_\text{T} = 25.6\text{mV}$.

network exhibited winner-take-all behavior.

I also investigated the gabaergic decay-constant's role by varying it and measuring the network period (Figure 4.10). My analysis (Section 4.3) confirmed that the effective delay strongly influences the network period; it predicts that the network period is between two and four times the effective delay with the proportionality constant's exact value determined by the gabaergic decay constant, which has a modulatory role. As predicted, the network period increased (16.2ms to 29.8ms) with increasing gabaergic decay constant (3.2μs to 20.6ms) for synchronous network states. Thus the network period ranged from 1.0 to 1.8 times the gabaergic rise-time (plus offset) consistent with my extension of DMS (Equation 4.6).

4.6.3 Input Current

To evaluate the influence of input current on synchrony, I varied it, observing the change in network period. The input current had a modulatory effect on the network period, which decreased from 38.2 to 19.6ms as the input current increased from 13 to 46nA (Figure 4.9 *Right column*). The input current changed the network period indirectly by influencing the time it takes the AH to spike after the soma reaches threshold, which changed the delay offset. At medium input current (31nA), the AH delay was 2.6ms, resulting in a network period of 24.2ms. For a smaller input current (13nA), a 7.4ms-delay accounted for the extended network period (38.2ms); for a larger input current (46nA), a 1.0ms-delay accounted for the truncated network period (19.6ms).

In addition to the AH delay, input current influenced VS and NAI. At small input current levels (below 10nA) interneurons spiked at low rates, resulting in a deficiency of inhibition

in the network, which was insufficient to synchronize the population. As input current increased, spike rates and inhibition increased, enabling synchrony, and increasing NAI. Large input currents (above 50nA) drove many interneurons hard enough to make them insensitive to inhibition, resulting in an asynchronous state. These most excitable interneurons suppressed the other interneurons that were still sensitive to inhibition, reducing NAI.

4.6.4 Poisson Input

My results show that silicon interneurons synchronize in the gamma frequency range. However, most of the neurons spike at the network rhythm, which is inconsistent with biological observations: Individual interneuron's membranes phase lock with gamma, but they do not spike each cycle; instead, they randomly skip cycles due to suppression from other interneurons in the population [46]. A numerical study showed that neuronal variability and noise in excitatory synapses can account for this biological behavior [140].

To test the network in a noisy environment, I replaced the constant current input with Poisson generated pulses (1ms) at an average rate of 600Hz, independently generated for each neuron (Figure 4.11). With all parameters the same as without input noise (Figure 4.7 *Left*), interneurons synchronized with a network period (25ms) almost identical to the one without noise (26ms).

As expected, interneurons did not spike in every cycle, skipping cycles when they received less input, which rendered them susceptible to suppression by other interneurons (Figure 4.11). The random spiking provided an opportunity for less excitable interneurons to participate (NAI = 148 compared to 120 with constant input), because more excitable interneurons, which inhibited them, did not spike every cycle. It also reduced the network

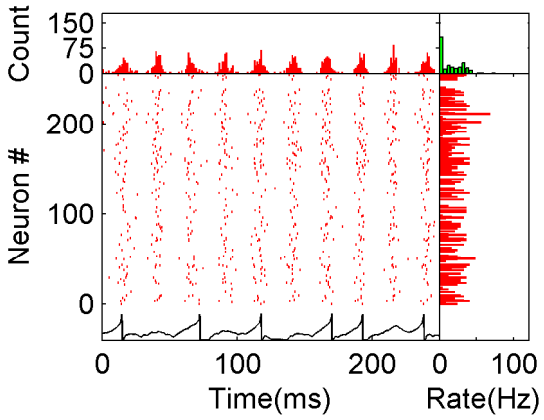

Figure 4.11: Synchrony with Noisy Input

The interneuron network synchronizes robustly even with noisy input. The noisy input degrades VS (0.54), but fails to abolish synchrony. The conventions are the same as Figure 4.6.

coherence (VS = 0.75 compare to 0.54) by jittering interneurons' spiking phases. Reducing the input rate increased the noise and reduced excitatory drive, causing VS and NAI to decrease, but the network period remained constant (not shown).

4.6.5 Pyramidal Neuron Entrainment

To confirm that interneurons entrain pyramidal neurons, I drove all 1,024 pyramidal neurons with a constant current input. When the pyramidal neurons spiked, they drove the interneurons to spike with some delay (6ms average); the interneurons in turn inhibited the pyramidal neurons (Figure 4.12). The current drive to the interneurons then ceased, as did their spiking. Once the inhibition decayed sufficiently, the pyramidal neurons spiked again, repeating the process. The network behavior was similar to the interneuron network alone but with added delay—the time the pyramidal neurons spiked to the time the interneurons spiked. Adding twice the average time difference to the effective delay I found earlier (fit

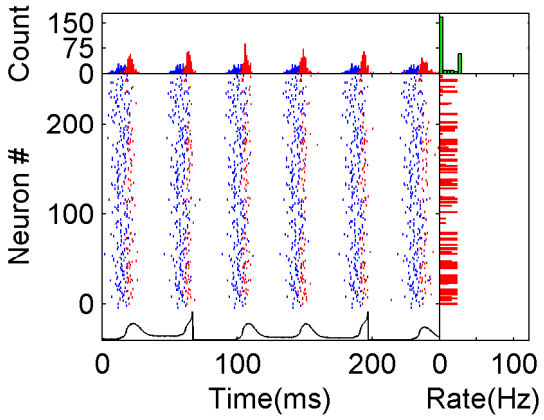

Figure 4.12: Synchrony in Pyramidal Neuron-Interneuron Network

The inhibitory interneuron network becomes more synchronous (VS$= 0.93$) with excitatory pyramidal neuron excitation. The pyramidal neurons (blue) drive the interneurons (red), which then inhibit the pyramidal neurons, as can be seen by the lag in phase of the interneurons. The conventions are the same as Figure 4.6.

in Figure 4.8) yielded a network period of 41.4ms; I measured 42.9ms.

The interaction between the interneuron and pyramidal neuron populations provided a structured input to the interneurons, increasing VS to 0.95 from 0.83 with the constant current input (Figure 4.7 *Left*). The interaction decreased NAI to 89 from 120, a result of the added variability in potency and timing from the pyramidal neurons' excitatory drive to the interneurons. Unlike Poisson input, excitation from pyramidal neurons was repetitive, providing approximately the same excitation each period; however the drive received varied among interneurons. On one end of the spectrum, interneurons near excitable pyramidal neurons that consistently spiked early and drove strong synapses, received potent excitation, causing many of them to spike. On the other hand, interneurons near lethargic pyramidal neurons that consistently spiked late and drove weak synapses received feeble excitation, causing many of them to be silenced by inhibition.

4.7 Discussion

In this chapter, I presented my silicon model of synchrony generated by mutually inhibitory interneurons. Based on the dynamic translinear principle, I designed the interneuron circuit using a current-mode conductance-based approach. This approach reproduced the important characteristics of the phase sensitivity like PRCs of biological neurons and enabled the network to create structure and synchrony with only structureless (excitatory) input [124].

This work demonstrates that, for the generation and control of robust synchrony, rise-time can act as an effective delay. This extends Maex and De Schutter's conjecture that the network period is four times the sum of the axonal and synaptic delay to include the effective delay contributed by the gabaergic rise-time [94]. In addition, the interneuron network synchrony verifies their work in a physical system.

This work is the first biologically plausible model of synchronous inhibitory interneurons implemented in silicon. It demonstrates that silicon is an appropriate medium to build dense networks of conductance-based neurons. Further, it demonstrates that results from silicon networks can support neurobiological theories as well as contribute to them (such as the effective delay due to gabaergic rise-time). Also, this work shows that synchrony can be achieved in highly heterogeneous networks (CV=0.24 for the uncoupled network's firing rate), far beyond what has been shown in computational studies (CV=0.06 for background input current [140]).

In addition, my colleagues and I have used these same interneurons and pyramidal neurons elseware to show how local inhibitory interactions, which generate synchrony in a spatially distributed manner, can mediate binding, which is not possible with a global clock [8]. When these interactions are far-reaching (mediated by a diffusive grid), coherence was

weakly dependent on distance; however, when the extent of the interactions was only a few nodes, coherence depended strongly on distance: Patches of neurons were incoherent until they fused into a single patch, realizing binding.

The interneuron network still has several areas of potential improvement. The gabaergic synapses include a rise-time but lack pure axonal delay, which would, intuitively, improve the degree of network synchrony. In future chips, I propose to build pure delay into the synapses on chip as shown in Figure 4.13, where the AER reset signal also triggers a delay element. The delay element could be as simple as a PE with a circuit sensitive to falling transitions at its output. Another issue is that the gabaergic synapses are shared between both the interneuron and pyramidal neuron populations, whereas in biology the gabaergic synapses have different properties, especially decay constants, for the different populations [10, 52]. Different decay constants would potentially improve the degree of synchrony in the network.

The range over which the silicon network is synchronous (VS> 0.50) is limited, ranging from about 7 to 58Hz. Faster synchrony may be desirable, in for example, modeling sharp waves [157]. If I decrease the rise-time and increase the input current, the silicon network can synchronize faster up to at least 200Hz but fewer interneurons fire (not shown) and I encounter a bug[4] in the AH at these higher rates that, occasionally (once every several seconds), artificial synchronizes all interneurons. In the data shown the bug occurs only rarely, if at all, because either the rise-time is too long or the input current is too low to trigger it. The pure synaptic delay will help in this respect.

[4]The AH bug occurs when a neuron raises REQ but immediately lowers it before the transmitter acknowledges by raising ACK. The bug is caused by an increase in inhibition (from another neuron) just as REQ is activated by AH, pulling the soma voltage up (towards rest), which turns off AH. This brief request violates the address-event protocol, locking up the transmitter, until this neuron or another in the same row reaches threshold and activates REQ. During this period, all requests from other rows must wait for the violation to clear, which releases the transmitter. Once released the transmitter services all waiting neurons as fast as it can, artificially synchronizing them at the microsecond scale.

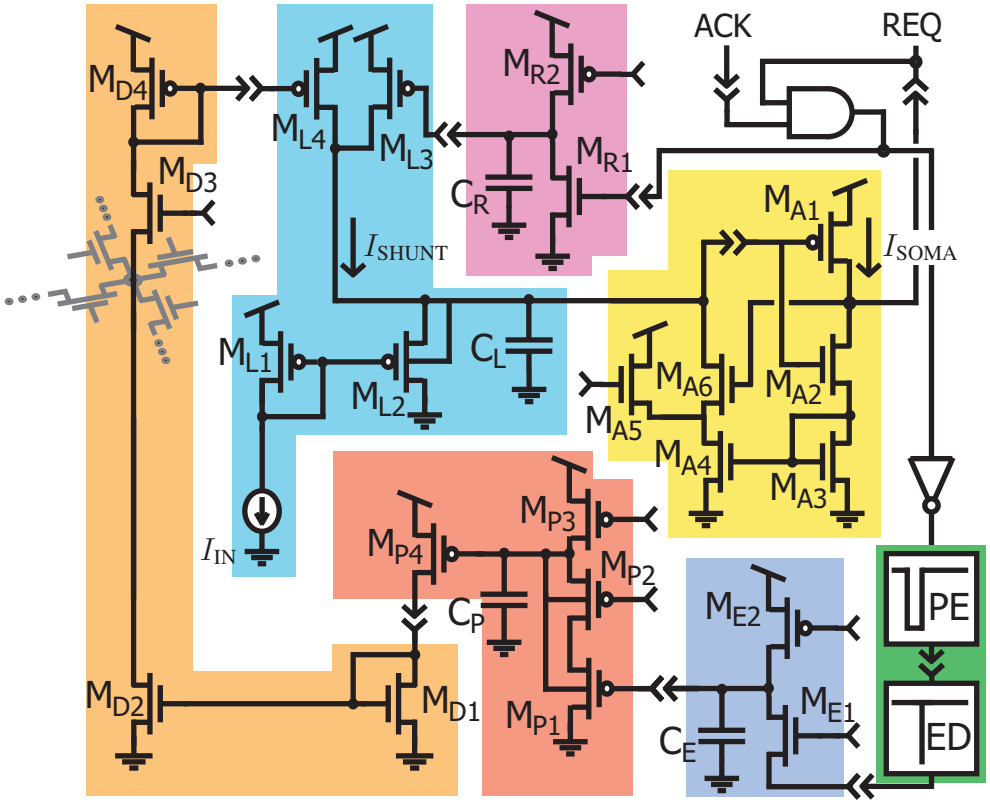

Figure 4.13: Interneuron Circuit with Axonal Delay

This interneuron circuit is identical to one in Figure 4.4 but with two added subcircuits (green): One is a pulse extender (PE), the axonal delay, that delays synaptic activation. The second subcircuit, the rising edge detector (ED), receives input from the added pulse circuit and when the delay pulse circuit turns off sends an (active low) output impulse to the gabaergic synapse's cleft ($M_{E1\text{-}2}$).

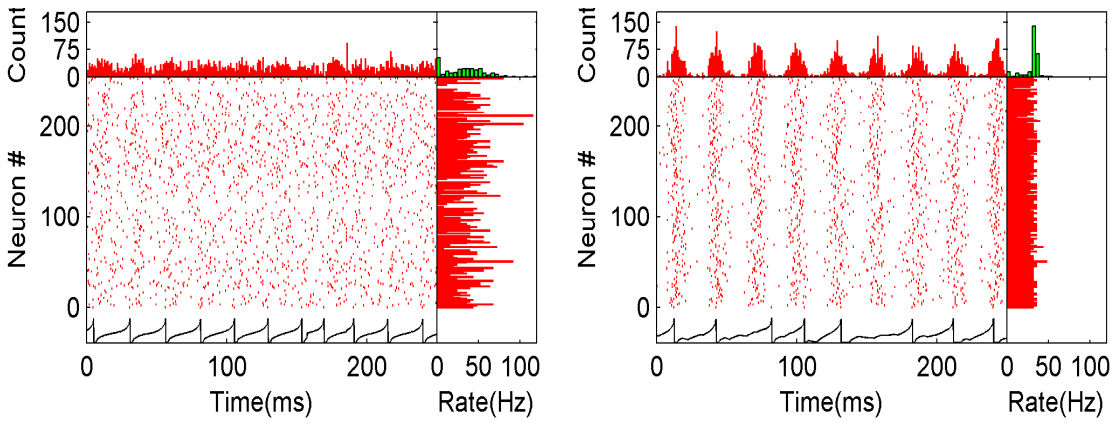

Figure 4.14: Homeostasis Compensates for Variability

Interneurons synchronize with reduced suppression when their input rates are adapted based on their output rates. *L*eft Before adaptation, using weak inhibition the interneurons (number active = 204) spiked asynchronously (vector strength $= 0.05$). *R*ight After several minutes of adaptation, the interneurons (number active = 243) spiked synchronously (vector strength $= 0.66$). Conventions are the same as in Figure 4.6.

Another limitation of the silicon interneuron network is interneuron suppression. It is not known whether a large fraction of interneurons are suppressed in biology, however, if too many do not fire, it is an inefficient use of neural resources. Some data shows that interneurons are stochastically suppressed [46]. I have shown that noisy input current can reduce suppression, but only from 53% down to 42%. Other biological mechanisms may reduce the suppression and/or increase synchrony. As mentioned before, the addition of pure synaptic delay to silicon synapses may decrease the number of interneurons that are suppressed. Also, as suggested by biological observations, adding gap junctions between pairs of silicon interneurons may help recruit less excitable interneurons [22]. Another tactic employed by biology that may be used to reduce suppression among interneurons is homeostasis of excitability.

Homeostatic mechanisms could reduce variability among neurons, decreasing suppres-

sion (Figure 4.14) [123]. For this small population of (256) interneurons, I can implement such a mechanism in real-time with a PC. The PC stimulates each neuron with an independent Poisson spike train, generated in real-time. Using the algorithm from [123], I can adapt the average rate of Poisson input spikes (rather than the maximum input current) to each neuron to increase or decrease their spike rate towards a reference rate (40Hz). This is similar to decreasing the probability of synaptic release at excitatory synapses from pyramidal neurons. After several minutes, adaptation reduced the interneurons' firing rate CV from 0.50 to 0.22, reducing the number of suppressed neurons to 12 (5%). In this experiment, I reduce the gabaergic strength, which makes the adaptation more stable and results in less suppression. Similar homeostasis of excitability could be implemented on chip, by scaling input current in each neuron's circuit rather than probability of release, realizing a scalable solution to suppression. For example, each neuron could compare a low-pass filtered version of its spike rate with a common reference current. If the filtered rate is less than the reference, a circuit would increase the synaptic strength (stored on a floating gate); if the filtered rate is greater than the reference, the circuit would decrease the synaptic strength. This comparison would express variability, which would result in variability among interneurons, but the one comparison would be less variable than the circuit's intrinsic variability from multiple sources.

In summary, my silicon interneuron network generates robust gamma frequency synchrony with rise-time as the critical parameter. It shows the usefulness of silicon in modeling conductance-based neural systems, reproducing and verifying theoretical results. Also, the silicon implementation suggests several improvements for future models, including adding pure delay and gap junctions to the synapses and interneurons, respectively.

Chapter 5

Silicon Synaptic Spike Timing-Dependent Plasticity

In Chapter 2, I described my model of sequence memory in the hippocampal formation, describing biological spike timing-dependent plasticity (STDP) and alluding to its importance in learning patterns and sequences of patterns. This chapter describes the design and test of the STDP circuit (in the STDP synapse), beginning with an brief review of STDP in the hippocampus.

Section 5.1 summarizes the observed properties of biological STDP, primarily in the hippocampal formation. Then in Section 5.2, I describe previous implementations of STDP, as well as membrane voltage-dependent plasticity (MVDP), and the benefits and disadvantages of each implementation. Next, in Section 5.3, I present my STDP circuit, describing the design optimizations and tradeoffs. Section 5.4 characterizes the STDP circuit's behavior and variability. Finally, in Section 5.5, I discuss the implications of the STDP circuit at the system level in the hippocampal model as well as its limitations and nonidealities and

how they can be overcome in future chips.

5.1 STDP in the Hippocampus

Neurobiology expresses amazing adaptation and learning: Its primary means of change is believed to be synaptic plasticity. Experiments show that a population of synapses increase (LTP) and decrease (LTD) their (summed) efficacy depending on their presynaptic input's correlation with their postsynaptic target neurons' activity [14]. This activity-dependent learning rule correlates pre and postsynaptic activity in the synapse by using NMDA receptor-gated channels, which require both presynaptic input (glutamate release) and postsynaptic activity (membrane potential above the NMDA threshold) [34]. The mechanism that introduces timing into synaptic plasticity is thought to be backpropagation of postsynaptic spikes to dendrites, signaling the synapse that it was effective in driving the soma (or at least causally correlated).

STDP is a particular synaptic learning rule in which (usually) a presynaptic spike repeatedly followed by a postsynaptic spike results in LTP and a postsynaptic spike repeatedly followed by a presynaptic spike results in LTD (Figure 5.1). Studies recording single synapses show that they are sensitive to timing and express a binary weight, either fully potentiated or fully depressed, but there may be several synapses between pairs of neurons leading to analog behavior (See Section 2.3.3). [154].

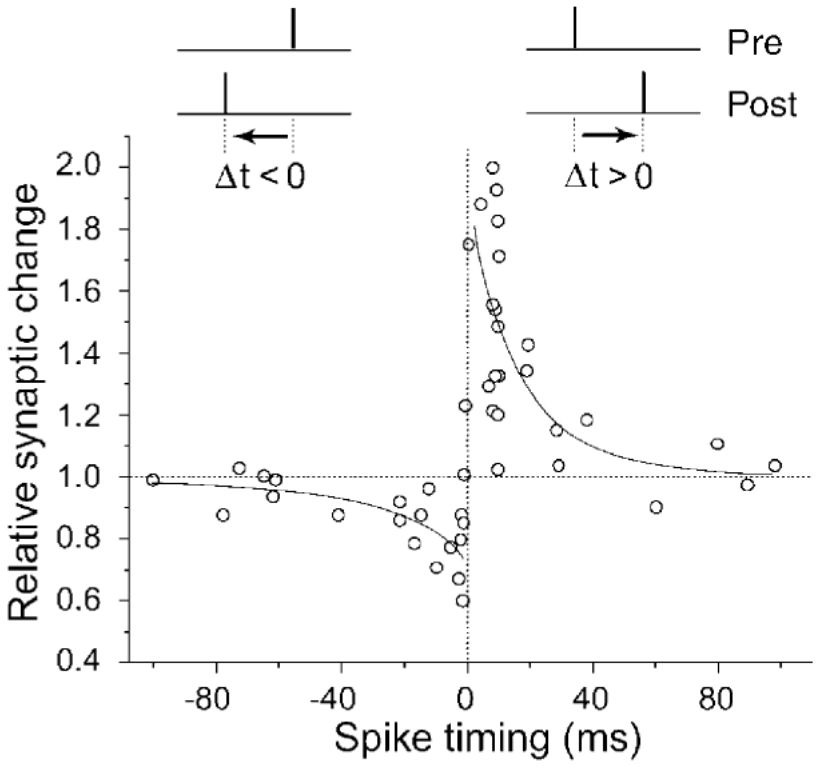

Figure 5.1: Hippocampal Spike Timing-Dependent Plasticity

Synaptic potentiation and depression each occur within a window of presynaptic and postsynaptic spike timing differences (Adapted from [12, 13]).

5.2 Plastic Silicon Synapses

Because synaptic plasticity is central to the brain's ability to learn, several neuromorphic implementations have been constructed. Two types of circuits emerged: One type implements MVDP, with various degrees of biological detail. The other type implements STDP by using spike timing directly as the plasticity signal, ignoring voltage entirely. MVDP circuits are inherently sensitive to the variance of their constituent elements, which often results in weight changes of the wrong sign. On the other hand, STDP circuits are less sensitive to their component variance when determining the sign of synaptic change, and

are, therefore, more robust than MVDP circuits.

5.2.1 Membrane Voltage-Dependent Plasticity

The simplest implementation of MVDP increased or decreased the synaptic weight when a synaptic input arrived while the neuron membrane was above or below a threshold, respectively [48]. This synaptic weight took graded values over short time scales, but drifted slowly up or down when the synaptic strength was above or below a threshold, realizing synaptic bistability. Shortly before the neuron spiked, its membrane potential was likely to be high; shortly after the neuron spiked, its membrane potential was likely to be low, realizing STDP on average. However, when the neuron was hyperpolarized and did not spike the synaptic weight decreased whenever it received a synaptic input, inconsistent with STDP. Improving on this design, I previously developed an MVDP circuit free from the inappropriate weight decrease found in other designs.

My improved implementation of MVDP increased the synaptic weight (all-or-none potentiation), when the current through a model NMDA receptor-gated channel population exceeded a threshold [5]. The NMDA current depended both on synaptic input and membrane depolarization. In this circuit, depression was designed to be implicit: when a weak (depressed) synapse potentiates it sent an AER signal off chip, which informed the system that it should swap the signaling synapse's location in memory with the oldest potentiated synapse (on the same dendrite). Thus, old strong synapses would be depressed at random, but would have the opportunity to be potentiated again.

This system had the advantage that if a neuron (or dendrite) was hyperpolarized, the synaptic weights remained fixed, while still implementing STDP. On the other hand, the

synapses were complex; they required several transistors and relied on voltage comparisons (NMDA threshold), which were prone to variations among transistors. The variations among MVDP circuits degraded the performance of the system: Many synapses potentiated every time they received an input and others never potentiated. STDP implementations that use timing eliminate voltage comparisons, reducing the effects of variations among synapses.

5.2.2 Spike Timing-Dependent Plasticity

The first STDP implementation was based on the *Modified Riccati Rule*, a spike-based rule in which pre-before-post activity increased the synaptic weight, whereas postsynaptic activation without a closely preceding presynaptic spike decreased synaptic strength [53]. This implementation differed from biological STDP: synapses were weakened even without presynaptic spikes. To rectify this difference, another STDP synapses was constructed.

The subsequent STDP implementation used timing both to potentiate and depress synapses: pre-before-post and post-before-pre activation increased and decreased synaptic weights, respectively [19]. This synapse, like its predecessor, employed graded synaptic weights without bistability. Therefore, over long time periods synapses decayed to zero, forgetting what they learned. This was rectified in a similar circuit by adding bistability [71] in the same way as [48], maintaining one bit of weight information indefinitely.

One-bit synaptic weights are not only easier to implement, but they are also computationally more powerful and better represent biological synapses than graded synaptic weights. Synapses that retain or even limit weight information to one bit are more efficient to implement as they require less space than multibit circuits, which have either

multiple stable levels [55] or a continuum of levels, typically requiring the use of floating gates [43, 54]. Using STDP, multibit circuits converge to a bimodal distribution [133] similar to a one-bit weight (but see [19]). Analysis of one-bit versus multibit synapses in neural associative memory systems showed that memory capacity surprisingly increased with the one-bit synapse; it prevented neurons from relying heavily on one or a few inputs (or neighbors) and from discounting others [119]. In addition, single biological synapses express a binary weight [154]; theory suggest such weights are more robust to variations among synapses and more stable against ongoing protein recycling [89]. For these reasons, I migrated to an STDP design, presented next.

5.3 The Silicon STDP Circuit

I present the first mixed analog and digital STDP circuit in deep-submicron (DSM) CMOS. DSM enables the fabrication of larger numbers of devices and circuits, but presents many challenges as well, such as high transistor leakage, variability, and noise. My circuit mitigates the disadvantages of DSM for analog circuits with design innovations: Separate potentiation and depression integrators increase immunity to leakage, mismatch and noise, enabling an integration time an order of magnitude larger than possible with nondifferential designs. Further, my circuit capitalizes on the advantages of DSM for digital circuits with dense designs: compact SRAM cells store binary synaptic weights. I arrived at this design after considering several requirements.

5.3.1 Specifications

The STDP circuit must explicitly realize timing dependence, which promotes correct behavior by eliminating the comparison between the neural membrane voltage and a reference in MVDP. The only previous MVDP-based implementation shown to reproduce STDP suffered from poor yield in DSM CMOS due to mismatch (worse than larger feature-size CMOS technology) [5]. Also, the voltage-gated channels necessary for MVDP are more complex to implement than an explicit timing dependence.

The STDP circuit weight must be binary and stable (over time) to increase the network memory capacity and to retain memories, respectively. A (digital) binary weight also simplifies read and write operations to and from the synapse, respectively; enabling rapid and efficient characterization of the STDP circuit as well as the network in which it is embedded. The binary weighted STDP circuit must integrate its inputs over time to reduce sensitivity to noise.

To achieve a long integration time in DSM, the STDP circuit use two independent capacitors: one for LTP and one for LTD. The integrators do not permanently hold the synaptic state; they only integrate input pairings (pre-before-post or post-before-pre). Using two capacitors eliminates the need for positive feedback, required to maintain the weight over time when a single capacitor integrates both potentiating and depressing pairings. Without positive feedback, the input currents only have to compete with leakage, therefore, they can be much smaller, realizing prolonged integration while maintaining a small capacitor size.

Using two capacitors implicitly assumes that LTP and LTD signals are integrated independently. Most STDP models assume that LTP and LTD signals compete, using a single integrator, with one signal driving the integrator up and other down. Neither this compe-

tition nor independent integration have been explored experimentally, and therefore, the question is still up in the air (See Section 2.3.3.3). In many cases, these two modes of operation yield similar behaviors; for example, repetitive input patterns drive a single integrator in one direction and only one integrator of a pair, yielding LTP or LTD alone. In other circumstances, the two models differ: With a single integrator a synapse can only repeatedly alternate between LTP and LTD if the inputs driving each are well matched; however, with dual integrators synaptic strength can bounce between LTP and LTD if both integrators receive strong input, even when it is unbalanced.

5.3.2 Circuit Design

The STDP circuit, for which the STDP Chip is named, is the most abundant, with 21,504 copies on the chip. This circuit is built from three subcircuits: decay, integrator, and SRAM (Figure 5.2). The decay and integrator are used to implement potentiation and depression (in a symmetric fashion). The SRAM holds the current binary state of the synapse, which is either potentiated or depressed.

The STDP circuit gates received address events. The potentiation signal, $\sim LTD$, connects to the gate of an NMOS transistor in series with the transistor that meters charge onto the PE of an excitatory synapse (Figure 3.8). So if the circuit is depressed, $\sim LTD$ is low, blocking any charge from reaching the PE. Conversely, if potentiated, $\sim LTD$ is high, allowing the charge to pass through and drive the PE. Although all 21 STDP Circuit on a pyramidal neuron have independent addresses, they all drive the same PE-LPF synapse 5.3).

For potentiation, the decay subcircuit remembers the last presynaptic spike (and only

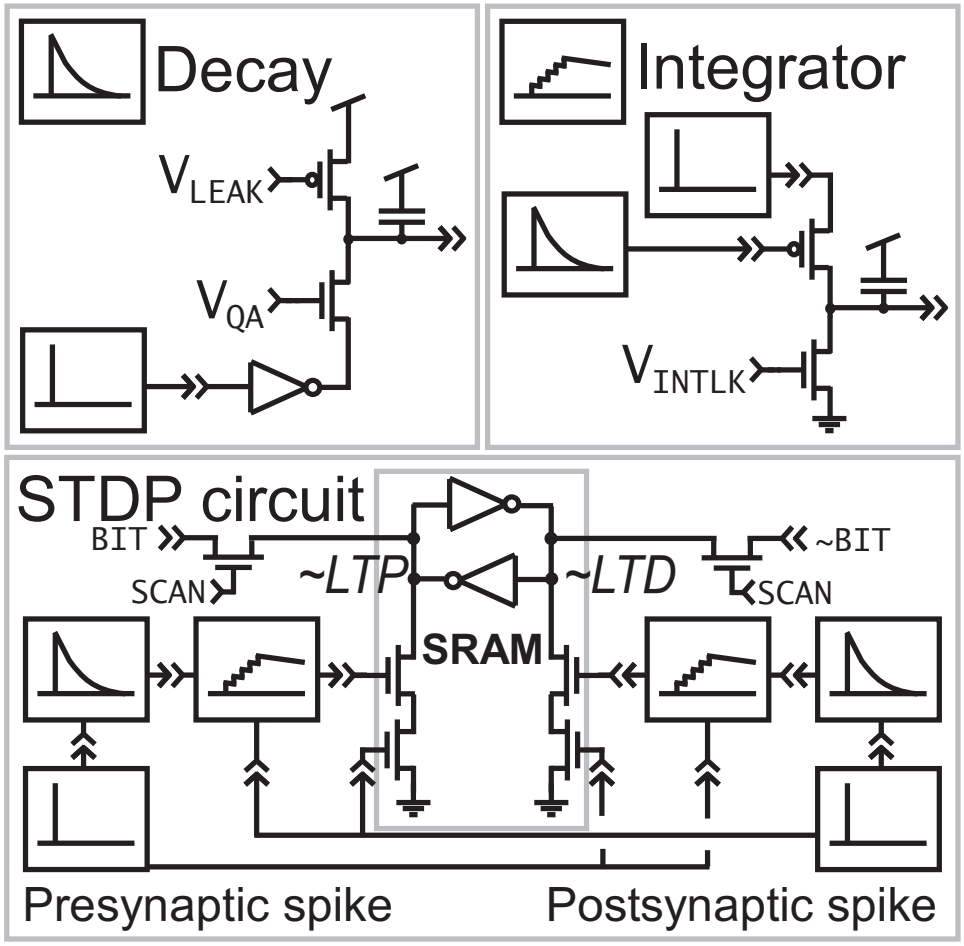

Figure 5.2: STDP Circuit Design

The circuit is composed of three subcircuits: decay, integrator, and SRAM.

the last spike) for a brief period. Its capacitor is charged when that spike occurs and discharges linearly thereafter. A postsynaptic spike samples the charge remaining on the capacitor, passes it through a quadratic function, and dumps the resultant charge into the integrator subcircuit. This charge decays linearly thereafter. At the time of the postsynaptic spike, the SRAM, a cross-coupled inverter pair, reads the voltage on the integrator's capacitor. If it exceeds a threshold, the SRAM switches state from depressed to potentiated ($\sim$*LTD* goes high and $\sim$*LTP* goes low). To reduce power consumption, the potentia-

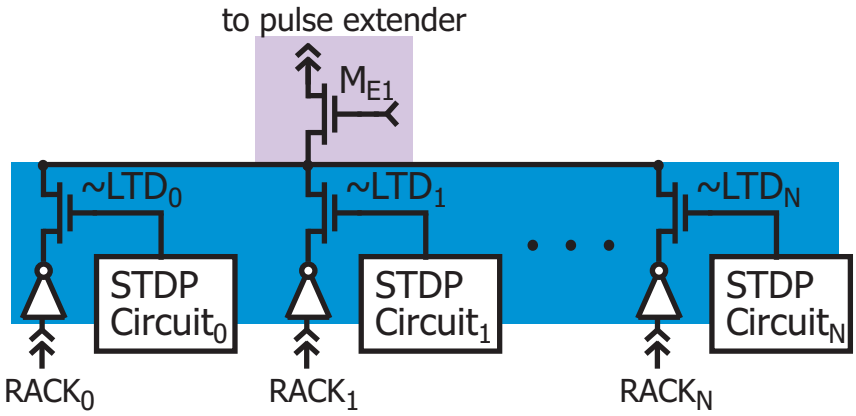

Figure 5.3: STDP Circuits Gate Synaptic Input

All 21 STDP circuits on a pyramidal neuron receive independent address-event input ($RACK_{0...N}$) but share the same synapse. (M_{E1} is the same as in the fixed synapse in Figure 3.8).

tion integrator only drives the SRAM during postsynaptic spikes. The depression side of the STDP circuit is exactly symmetric, except that it responds to postsynaptic activation followed by presynaptic activation and switches the SRAM's state from potentiated to depressed ($\sim$*LTP* goes high and $\sim$*LTD* goes low). This digital state can be read or written from off chip, using a scanner [103].

The scanner has two functions: to read and to write each SRAM's state. It is composed of two linear shift-registers: one for vertical and one for horizontal. The user selects a synapse using two digital-input pads, one for a vertical clock and one for a horizontal clock: Each time a clock is pulsed, the shift-register advances one row or column (wrapping around to the beginning if it reaches the end). When SCAN is high, selected by the vertical shift-register, a row is selected (see Figure 5.2): Every SRAM in the row drives either its $\sim$BIT or BIT line low, depending on if it is depressed or potentiated respectively [156]. The horizontal scanner selects one column's $\sim$BIT and BIT lines, which drive a digital-output pad that can be read by a user. Two additional input pads, $\sim$WRITEHI and $\sim$WRITELO,

allow the user to overwrite the state of an SRAM: If ∼WRITEHI or ∼WRITELO are pulled low a circuit is activated that removes the tonic high precharge and pulls the BIT or ∼BIT line of the currently selected SRAM low, flipping its state to potentiated or depressed, respectively, if necessary.

The scanner also selects one analog current at a time to drive a current-out pad, which I measure with an analog-to-digital converter. To test an STDP circuit, I select the analog current from the synapse that it drives. In this way, I can observe the excitatory postsynaptic current (EPSC)[1]. This gives us a way to measure how many pre-before-post or post-before-pre pairings at a given time difference are required to potentiate or depress, respectively.

5.3.3 Analysis

Here, I analyze the behavior of the STDP circuit, including the number of pairings at each pre-before-post and post-before-pre interval that is required for a state transition. The LTP integrator's potential, V_{LTP}, is:

$$V_{\text{LTP}} = \frac{n\, I_{\text{decay}}(t_{\text{diff}})\, \Delta t_{\text{pre}} - (n-1) I_{\text{INTLK}}\, \Delta t_{\text{period}}}{C_{\text{LTP}}} \tag{5.1}$$

where n is the number of pre-before-post pairings at time difference of t_{diff} and a period of Δt_{period}. $I_{\text{decay}}(t_{\text{diff}})$ is the current from the decay subcircuit, sampled for a duration of Δt_{pre}, the presynaptic pulse width, that drives the integrator's capacitor, C_{LTP}; I_{INTLK} is the

[1]The analog EPSC is a more convenient measure of synaptic strength than the digital SRAM state: If the SRAM under test is selected by the scanner its capacitance increases several orders of magnitude (due to the bit lines), preventing the integrators from switching the SRAM's state. Therefore, I would have to repeatedly scan through the array after every pairing to measure the synaptic state.

integrator's constant leak. The leak term is multiplied by $n-1$, because no charge can leak off the capacitor until the first pair has arrived, driving V_{LTP} above ground.

When V_{LTP} exceeds the SRAM's threshold, V_{THR}, the SRAM transitions from depressed to potentiated. The number of pre-before-post pairings until the STDP circuit potentiates is given by $n = \text{ceil}(n_{\text{C}})$, where $\text{ceil}(x)$ rounds x up to the nearest integer and:

$$n_{\text{C}} = \left(\frac{\text{C}_{\text{LTP}}\text{V}_{\text{THR}} - \text{I}_{\text{INTLK}}\,\Delta\text{t}_{\text{period}}}{I_{\text{decay}}(t_{\text{diff}})\,\Delta\text{t}_{\text{pre}} - \text{I}_{\text{INTLK}}\,\Delta\text{t}_{\text{period}}} \right) \tag{5.2}$$

obtained by setting $V_{\text{LTP}} = \text{V}_{\text{THR}}$ in Equation 5.1. I approximate n, which is a positive integer, with n_{C}, which is a nonzero positive continuous value that is on average $\frac{1}{2}$ less than n, due to the ceil function. Notice that all terms on the right side of the equation are constant except for $I_{\text{decay}}(t_{\text{diff}})$, which is approximately a quadratic[2], $I_{\text{decay}}(0)(1-\frac{t_{\text{diff}}}{\text{t}_{\text{E}}})^2$. Further, combining constants, yields:

$$n_{\text{C}} = \frac{1}{\alpha\,(1-\frac{t_{\text{diff}}}{t_{\text{E}}})^2 - \beta} \tag{5.3}$$

where t_{E} gives the value of t_{diff} where n_{C} grows to infinity, assuming β is negligible; otherwise it occurs at $\text{t}_{\text{E}}(1-\sqrt{\frac{\beta}{\alpha}})$;[3] $\alpha = \frac{I_{\text{decay}}(0)\,\Delta\text{t}_{\text{pre}}}{\text{C}_{\text{LTP}}\text{V}_{\text{THR}} - \text{I}_{\text{INTLK}}\,\Delta\text{t}_{\text{period}}}$ and $\beta = \frac{\text{I}_{\text{INTLK}}\,\Delta\text{t}_{\text{period}}}{\text{C}_{\text{LTP}}\text{V}_{\text{THR}} - \text{I}_{\text{INTLK}}\,\Delta\text{t}_{\text{period}}}$. The STDP circuit's efficacy is the inverse of $n_{\text{C}}(t_{\text{diff}})$.

[2] $I_{\text{decay}}(t_{\text{diff}})$ is quadratic because the decay subcircuit drives the PMOS transistor in the integrator subcircuit above threshold.

[3] For values of $t_{\text{diff}} > \text{t}_{\text{E}}(1-\sqrt{\frac{\beta}{\alpha}})$, $n_{\text{C}} \equiv \infty$; that is no number of pairings at that t_{diff} will ever potentiate the synapse.

5.4 Characterization

In this section, I characterize the STDP circuit's behavior as well as the variability among STDP circuits. I characterize the behavior by measuring efficacy for LTP (pre-before-post) and LTD (post-before-pre) given spike timing difference. Next, I measure a STDP efficacy curve (analogous to Figure 5.1), which quantifies the relationship between efficacy and spike timing. After characterizing a single curve, I study the influence of various parameters on the curve. Next, I study the variability among the many STDP curves and verify that all 21,504 STDP circuits both potentiate and depress. Finally, I measure the STDP circuit's response to varied Poisson presynaptic and postsynaptic rates.

I measured a single STDP circuit's efficacy, defined as the number of pairings required for a synaptic transition, for a given spike timing. I activated a single plastic synapse (weak) and a fixed synapse (strong)—which elicited a presynaptic and a postsynaptic spike at different relative times, the spike-timing difference of interest. I repeated this pairing at 10Hz, counting the number of pairings required to potentiate (or depress) the synapse. For example, if twenty pairings were required to potentiate the synapse, the efficacy was one twentieth.

During each pairing, I measured the synaptic strength by directly observing the EPSC, selected using the scanner (Figure 5.4). Although the SRAM is a digital circuit it does not switch abruptly due to the weak PMOS transistors that pull one side or the other to V_{DD}. Instead, each pairing pulls some charge off of the SRAM that the PMOS slowly restores; the affect is visible when the synapse depressed. After several pairings, the synaptic weight decreases a little during each pairing until it finally transitions to zero (Figure 5.4 *Left*). This affect is only visible when depressing because $\sim$*LTD* gates the synapse; if *LTP* gated the weight, the affect would only be visible when potentiating.

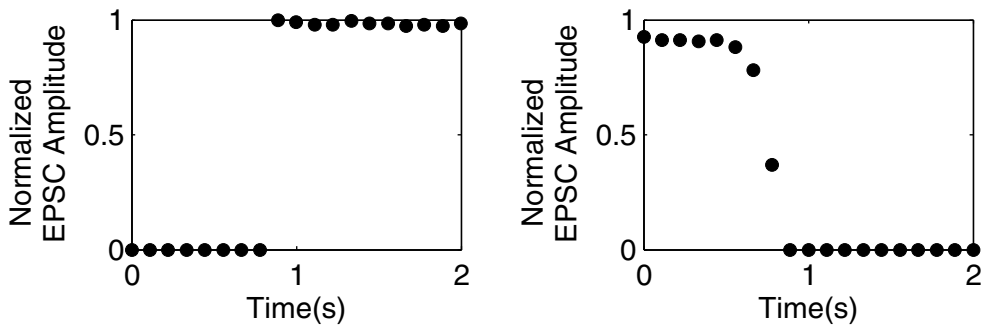

Figure 5.4: STDP Circuit Transition

Left The excitatory postsynaptic current increases (potentiates) after the presynaptic spike repeatedly precedes the postsynaptic spike by 9ms . *Right* It decreases (depresses) after the postsynaptic spike repeatedly precedes the presynaptic spike by 9ms. Time zero is the first spike pairing.

5.4.1 Spike Timing-Dependent Plasticity Results

I characterized a single STDP circuit by measuring the efficacy of various pre-before-post and post-before-pre spike-timing differences (Figure 5.5). Fitting the efficacies with Equation 5.3, the decay-time (t_{diff} at which $n_{\text{C}} = \infty$) for potentiation and depression are 21.8ms and 48.1ms, respectively. This behavior is similar to that observed in the hippocampus: potentiation has a shorter time window and higher maximum efficacy than depression [12].

5.4.2 Curve Parameters

It is important that the STDP curve can be tuned to increase or decrease the maximum efficacy and decay time until it is in the appropriate region of operation. If synapses learn slowly, switching states only after hundreds of pairings, the peak efficacy should be increased to enable learning on a shorter, neural time scale. On the other hand, if they learn fast, after a single pairing, the peak efficacy should be reduced to improve immunity to

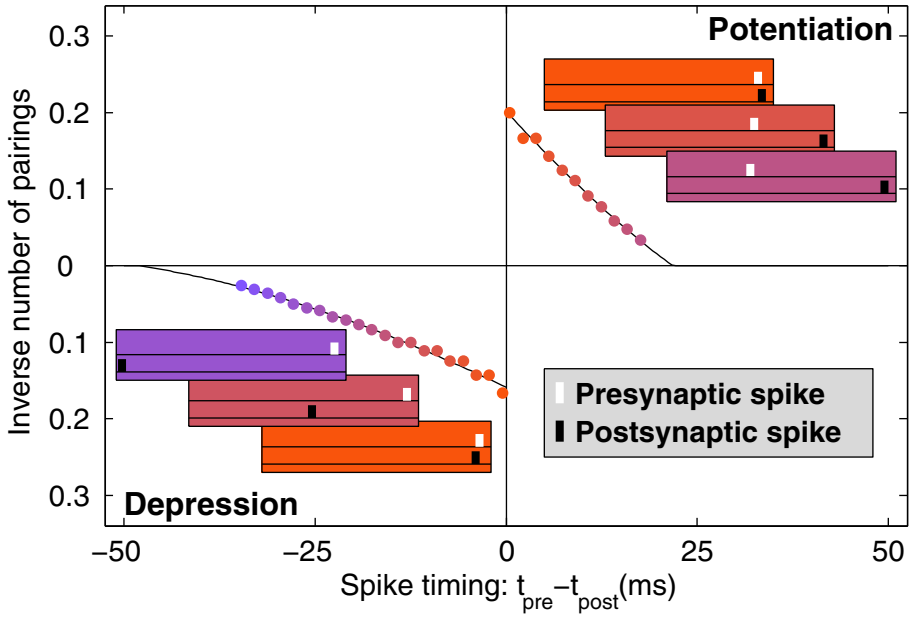

Figure 5.5: STDP Circuit Efficacy Curve

The circuit potentiates when the presynaptic spike precedes the postsynaptic spike, repeatedly, and depresses when the postsynaptic spike precedes the presynaptic spike, repeatedly. Lines are fits from Equation 5.3.

spurious spike pairings. Further, the decay time must be appropriate to the timing in the system: A long decay time links unrelated neurons, whereas a short decay time fails to make appropriate connections. Two biases control these parameters in the circuit: V_{QA} and V_{LEAK} (Figure 5.2). V_{QA} scales the circuit's efficacy, raising and lowering the entire curve (Figure 5.6). In this regime of operation V_{QA} sets the decay subcircuit's saturation level; however, if V_{QA} was much lower (near 1.0V) it would determine how much charge is metered in every spike. V_{LEAK} changes the circuit's decay-time with only a minor influence on the peak efficacy (Figure 5.7).

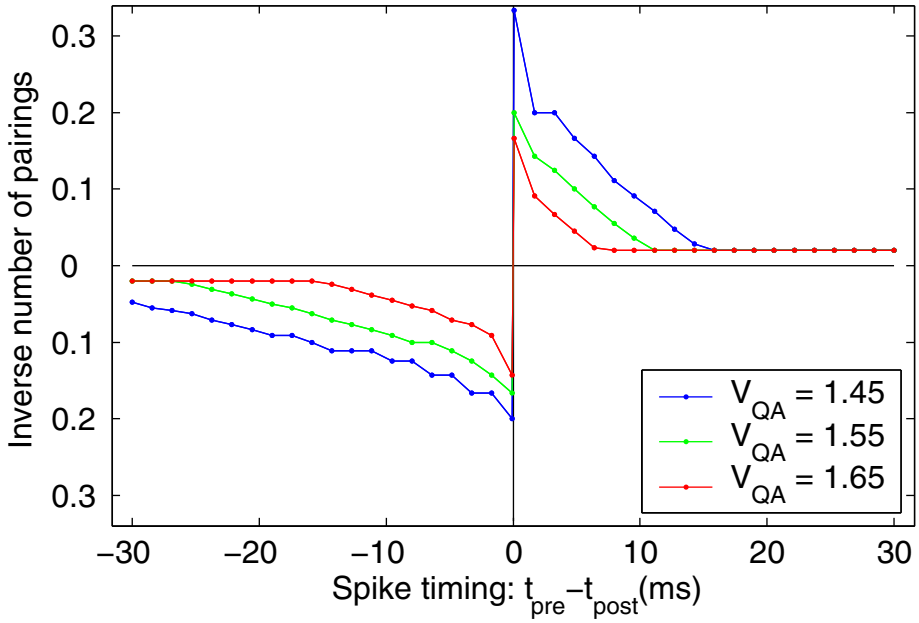

Figure 5.6: Tuning Peak Efficacy

As V_{QA} decreases, both potentiation and depression efficacy curves increase by setting their saturation levels. Lines are a guide for the eye.

5.4.3 Variability

Neurobiological synapses and neurons express variability, yet perform many tasks better than engineered systems with much less variability among elements. Therefore, when implementing neurobiological algorithms using neurobiological structures, we expect such systems to function despite variability. However, in the past, neuromorphic chip performance has suffered due to variability: For example, in an associative memory implementation using MVDP, the system was unable to learn patterns due to the extreme variability among MVDP circuits [5].

I characterize the variability among synapses by measuring the efficacy curves and

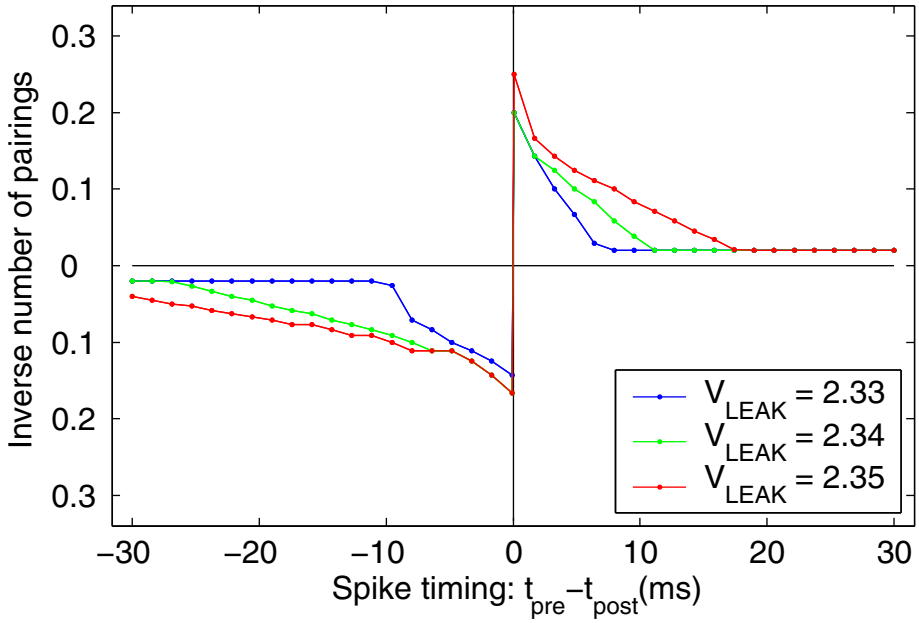

Figure 5.7: Tuning Decay Time

As V_{LEAK} increases, the slopes of both potentiation and depression efficacy curves increases (less negative). Lines are a guide for the eye.

comparing the standard deviation for each spiking-timing difference. I tested 121 (0.6%) synapses spread out across the chip[4] (Figure 5.8). Efficacies expressed a CV below 0.25 for absolute spike-timing differences below 10ms. Further, every synapse (of 121 tested) depressed with a spike-timing difference of -0.5ms; all but one synapse potentiated with a spike-timing difference of 0.5ms (in less than 50 pairings).

I only characterized variability among 121 STDP circuits because characterizing all 21,504 circuits is impractical. Through the scanner, I can only observe the influence of one STDP circuit at a time on a single synapse. Each synapse STDP curve takes over 4

[4]I started with the neuron in the third column of the third row and selected the neuron at every third column in that row, up to column 32. I proceeded to select neurons at the same column locations in every third row up to row 32. For every neuron, I selected the second (of 21) synapses. I selected the same synapses on every neuron because I observed no systematic difference depending on synapse number.

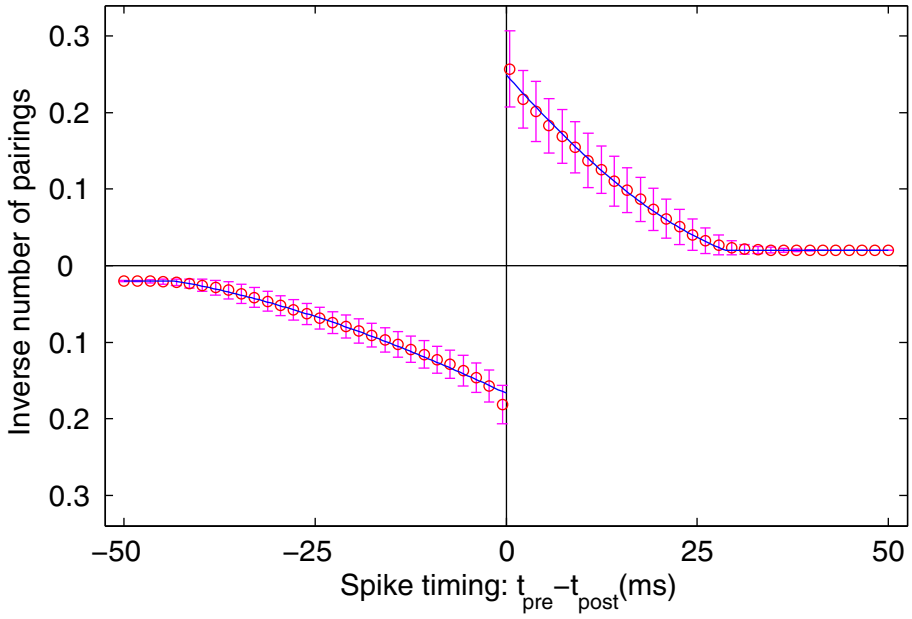

Figure 5.8: STDP Curve Variability

Variability among 121 STDP circuits is low as measured by the standard deviation at each spike timing. Lines are fits from Equation 5.3.

minutes; all 21,504 would take over two solid months of experiments. Over such a long period, bias values would drift considerably (due to changes in temperature), interfering with the estimation of variability. Instead, I verified that all the synapses were capable of both potentiating and depressing at one pre-before-post and one post-before-pre pairing, respectively. In addition, I stimulated and recorded from all STDP circuits simultaneously with Poisson inputs.

5.4.4 Transition Verification

I verified that all synapses both potentiate and depress by driving every presynaptic STDP circuit as well as every postsynaptic neuron with one pre-before-post and one post-before-pre pairing. To accelerate the experiment (down to just over twenty-four hours), I increased the frequency of pairings to 100Hz, compensating for the rapid pairings by increasing the leak current of the potentiation and depression decay subcircuits (by decreasing V_{LEAK}). This increase effectively scales time, running the experiment in *hypertime* as in [69]. I choose a time difference of one percent of the time between pairings or 100μs.

I found that every STDP circuit both potentiated and depressed with pre-before-post and post-before-pre pairings, respectively, which confirmed that all 21,504 circuits switched with the correct sign, functioning appropriately (Figure 5.9). Although they functioned appropriately, the STDP circuits expressed variability in the number of pairings that they required to switch states. For potentiation, 98.8% of circuits required between one and five pairings, 0.1% required between six and one-hundred fifty pairings, and 1.0% was excluded[5]. For depression, 91.1% of circuits required between one and five pairings, 7.4% required between six and fifty, 0.4% required between fifty-one and three-hundred fifty-eight pairings, and 1.0% was excluded. Whereas nearly all STDP circuits readily potentiated, the STDP circuits on four neurons were reluctant to depress with most of them requiring more than one-hundred pairings.

[5]Due to a bug in the CPLD code, I could not use the analog scanner to observe the number of pairings that 1.0% of STDP circuits required to switch, however, I was able to verify that they all switched the appropriate direction by scanning out their digital state.

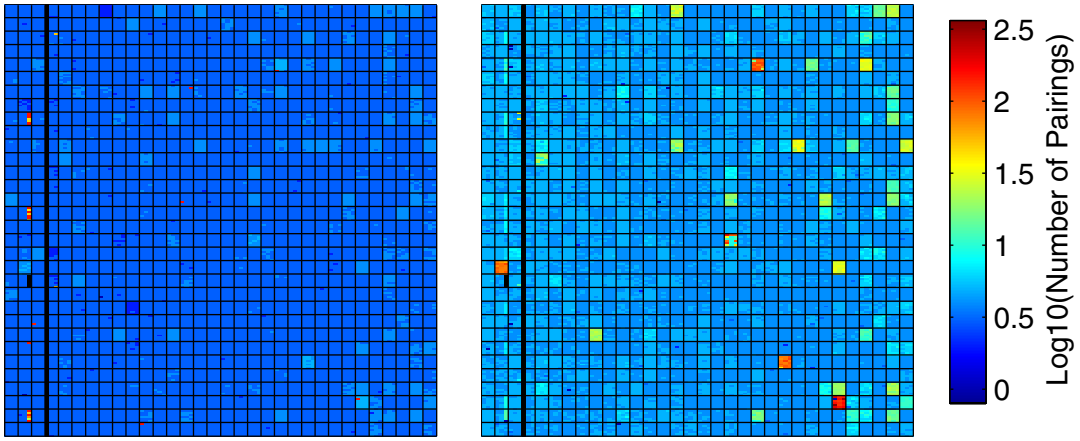

Figure 5.9: Potentiation and Depression Verification

All STDP circuits both potentiate and depress with Poisson input. The STDP chip includes a 32 by 32 array of pyramidal neurons (boxes), each with a 3 by 7 grid of synapses (dots). The number of pairings required to potentiate (*left*) or depress (*right*) at 100Hz with 100μs pre-before-post or post-before-pre pairings varies among neurons. Black dots represent STDP circuits that could not be measured directly.

5.4.5 Poisson Stimulus

I characterized STDP circuits' responses to Poisson generated spike trains at various rates by simultaneously driving every postsynaptic neuron and every presynaptic synapse with independently generated Poisson spike trains. Each postsynaptic neuron was driven exclusively through its fixed synapse, which was tuned such that for every input the postsynaptic neuron spiked once, occurring within tens of μs, unless it just spiked and was within its refractory period (about 1ms) in which case the input was ignored. Potentiated synapses were tuned to be ineffective at driving neurons. There were no explicit pairings, just those that randomly occurred due to the independent inputs.

I varied both the presynaptic and postsynaptic rates independently from 1 to 71Hz[6] in 1Hz increments across experiments. For each pair of presynaptic and postsynaptic rates, I performed two experiments, one in which all synapses began potentiated and one all depressed. At the end of each 12 second experiment, I recorded the state of all synapses, noting the mean synaptic weight and in how many experiments each synapse switched (in each direction). For each pair presynaptic and postsynaptic rates, I averaged the mean synaptic weights from the initially potentiated and depressed experiments, computing the fraction potentiated.

The different presynaptic and postsynaptic rates influenced the STDP circuits' probabilities of potentiation and depression (Figure 5.10). When both rates were comparable and above 20Hz, potentiation and depression were about equally likely (mean synapse weight of about 0.5). When both rates were below 10Hz, neither potentiation or depression occurred (also resulting in an mean weight of about 0.5), STDP circuits maintained their previous states. For high postsynaptic and low presynaptic rates, potentiation dominated (mean synaptic weight near 1), whereas for low postsynaptic and high presynaptic rates, depression dominated (mean synaptic weight near 0). Over all experiments (5041 starting in each state), I verified that every STDP circuit potentiated and depressed many times, additionally verifying the functionality of the circuits.

These results show that low presynaptic rates (about 3Hz) coupled with high postsynaptic rates (above 20Hz) produce strong synaptic potentiation. One presynaptic spike primes the presynaptic decay subcircuit. Because the decays circuit saturates with each presynaptic spike, infrequent inputs are as effective as frequent ones. When many postsynaptic spikes follow a presynaptic spike they drive the potentiation integrator repeatedly, quickly

[6]71Hz (1.6 million input events per second) was the maximum rate that could be achieved while maintaining Poisson statistics.

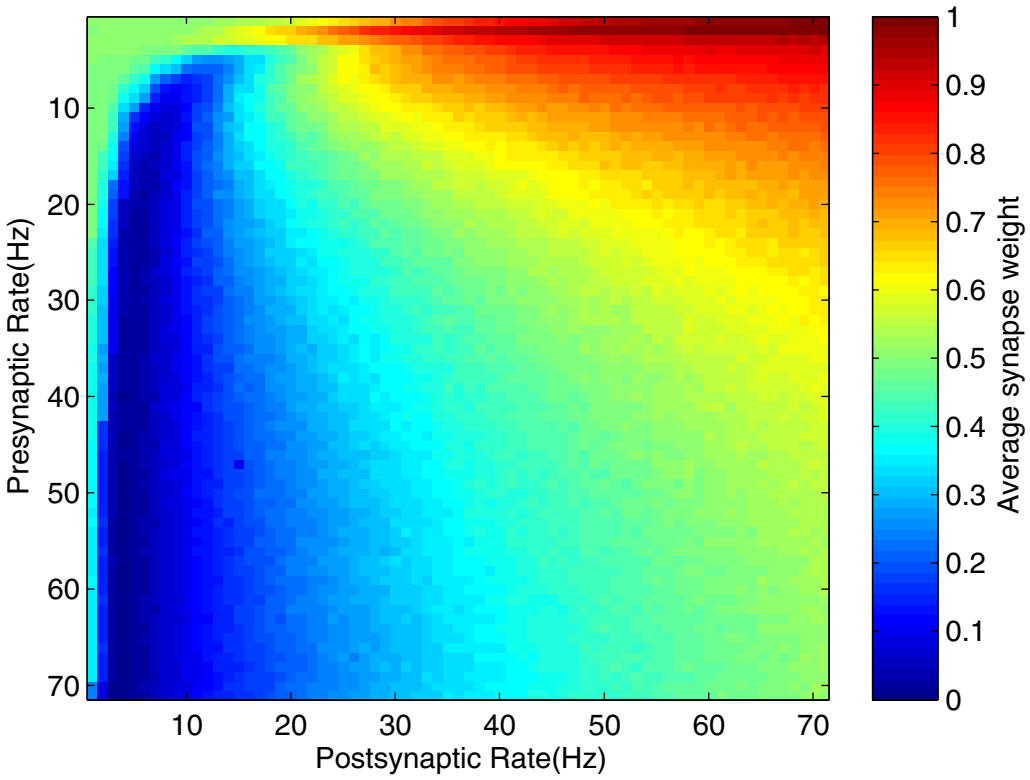

Figure 5.10: Poisson Spike Input

Poisson presynaptic and postsynaptic rates influence the expected synaptic state. Each point represents the mean of all synapses after 12 seconds of input.

reaching threshold and potentiating the synapse. The contrary phenomenon occurs favoring depression when postsynaptic rates are low (about 3Hz) and presynaptic rates are high (above 10Hz). When many presynaptic spikes follow a postsynaptic one they drive the depression integrator repeatedly, quickly reaching threshold and depressing the synapse.

When both presynaptic and postsynaptic rates are above 10Hz both integrators spend significant time above threshold; therefore, the most recent spike, presynaptic or postsynaptic, determines the state of an STDP circuit. The probability the circuit is potentiated

(P_{LTP}) is approximately the probability that the last arriving spike was postsynaptic, given by $\nu_{\text{POST}}/(\nu_{\text{PRE}}+\nu_{\text{POST}})$, where ν_{PRE} and ν_{POST} are the Poisson presynaptic and postsynaptic rates, respectively.

5.5 Discussion

Although the STDP circuits are themselves variable, with different efficacies and decay constants, timing ensures the sign of the weight-change is correct for every circuit measured (Figure 5.8). For this reason, I chose STDP over other more physiological implementations of plasticity, such as MVDP, which has the capability to learn with graded voltage signals, such as those found in active dendrites, providing more computational power [121].

Previously, I investigated a MVDP circuit, which modeled a voltage-dependent NMDA-receptor-gated synapse [5]. It potentiated when the calcium current analog exceeded a threshold, which was designed to occur only during a dendritic action potential. This circuit produced behavior similar to STDP, implying it could be used in PEP. However, it was sensitive to variability in the NMDA and potentiation thresholds, causing a fraction of the population to potentiate anytime the synapse received an input and another fraction to never potentiate, rendering both subpopulations useless. Therefore, the simpler, less biophysical STDP circuit won out over the MVDP circuit.

One reason STDP is powerful both in silicon and biology is its inherent noise immunity. A synapses switches states quickly when the signal-to-noise ratio (SNR) is high, which occurs when presynaptic spikes repeatedly occur immediately before postsynaptic spikes (for potentiation). Conversely, a synapses switches states slowly when SNR is low, which occurs when presynaptic spikes repeatedly occur long before postsynaptic spikes. When

the information available to a synapse is clear, it decides quickly, however, when less clear, the synapse integrates over time to ensure that it makes the correct decision. For example, when the pre-before-post time difference was 0.5ms and 17.6ms the STDP circuit took 5 and 30 pairings to potentiate, respectively (Figure 5.5). The tactic of acting quickly with clear information and slowly with unclear information is a general strategy found in biological elements, such as voltage-gated ionic channels [70].

To achieve noise immunity, the STDP circuit must integrate its inputs over time. Previous STDP designs that used a single capacitor with bistability to realize temporal integration and to store the synaptic weight are impractical in DSM [48, 71]. To maintain the synaptic state, the bistable feedback current onto a single capacitor must be larger than the worst-case transistor leakage currents, which are larger in DSM than in larger feature-size CMOS technology. These increased feedback currents compete with inputs that change the synaptic weight, which must be increased to overcome them. Increasing the input currents reduces the number of input pairings required to drive the weight on the capacitor above or below the bistable threshold, reducing the integration time. By using independent capacitors for LTP and LTD, I eliminate the need for positive feedback. Without positive feedback, the input currents only have to compete with leakage, therefore, they can be much smaller, realizing a long integration time.

Due to its design innovations, the STDP circuit functions properly despite its implementation in DSM CMOS. The circuit's structure, dual integrators with SRAM, successfully realizes biological plasticity with a level of variability such that all synapses function. However, a higher level of variability may be optimal. In general, the variability that a system can tolerate increases with the number of elements, since behavior can be averaged across more elements. Because variability is inversely related to circuit size (see Chapter 3), I could build more STDP circuits of smaller size while maintaining system performance,

but with an additional benefit: As the number of STDP circuits increases, the number of memories that the system can store increases. However, in the current implementation wire density limits the minimum circuit layout size.

In future implementations, I plan to use even smaller feature-size CMOS technologies, such as 0.18μm or 0.13μm which include more metal layers, 6 and 8, respectively, compared to only 5 in 0.25μm. These deeper submicron technologies suffer from increased leakage currents, which limit integration time, however this issue can be mitigated by using thick oxide devices, which ameliorate the problem of large leakage currents. Future implementations will support many more synapses and therefore much greater memory capacity than the current implementation, which nonetheless exceeds what has been achieved to date.

Metaplasticity can mitigate the low memory capacity of the current implementation, which has only 21 STDP circuits per pyramidal neuron. Metaplasticity moves the learning calculation on long time scales from hardware to software. The software can occasionally read the synaptic states from the network, looking for synapses whose states are consistent. Those consistently potentiated or depressed are moved to the fixed synapse on the same neuron or deleted entirely, respectively. When a synapse is constantly potentiated, it is moved from a plastic synapse to a fixed synapse, where it is permanently potentiated (until it is moved back), rather than being continually evaluated. Conversely, when a synapse is constantly depressed, it is deleted and replaced by a new synapse or a synapse that has been on the fixed synapse for a long period. In this manner, 21 STDP synapses and one fixed synapse can function as a larger population. Further, in future implementations the number of synapses can be increased by spreading them out across multiple chips.

Distributing STDP synapses across chips can realize a great increase in the number of

synapses available to each neuron. Because my synapses use STDP, where spikes and not voltage are the learning signal, there is no reason STDP circuits cannot be located apart from the neuron they drive. Such a system would allow arbitrary increase in the number of STDP synapses, up to the limit set by the system bandwidth.

Chapter 6

Timing in Silicon Autoassociation

In Chapter 2, I described my model of sequence memory in the hippocampal formation, describing how two recurrent networks, one autoassociative and one heteroassociative, can store and recall sequences of patterns. Then in Chapters 3 and 5, I presented the pyramidal neuron and STDP circuits, respectively. This chapter employs a network of pyramidal neurons with STDP synapses to realize an autoassociative network that stores and recalls patterns and compensates for variations among the neuron population.

Section 6.1 summarizes hippocampal properties related to the storage of autoassociative patterns in the CA3 region. Then, in Section 6.2, I describe previous VLSI implementations of abstract associative memory, contrasting them with neuromorphic associative memory. Next, in Section 6.3, I present my network of silicon neurons and its recurrent connectivity. In Section 6.4, I show the network's ability to compensate for variability in excitability and to store and recall patterns. In Section 6.5, I characterize the network's ability to compensate for variability from both intrinsic (fabrication) and extrinsic (input) sources. Finally, in Section 6.6, I discuss the implications of my implementation of autoassociation

as well as its limitations.

6.1 Hippocampal Associative Memory

The CA3 region of the hippocampus is believed to be its autoassociative memory, storing and recalling patterns [99, 111, 152]. It realizes associative memory using massive recurrent synapses that express STDP. These synapses potentiate when neurons are coactive in a pattern, so that when an incomplete pattern is presented, they recruit the missing neurons, recalling the pattern. Animal studies suggest that patterns stored and retrieved by CA3 (often) relate to navigation, with locations represented by pyramidal neurons that are known as place cells [84].

Place cells consistently encode the animal's location despite changes to the environment. In mice, removal of three of four visual cues in an environment did not alter place cells' preferred locations, suggesting pattern completion [113].[1] Place cells encode an animal's position in both their firing rate and spike timing.

Evidence suggests precise spike timing is important in place cell coding. Place cells use timing in their spike activity (in addition to rate) to encode location in space [115]. They employ a phase code: the timing at which a neuron spikes relative to the phase of a global inhibitory theta rhythm (5-12Hz) conveys information [117]. As an animal approaches a place cell's preferred location (its place field), the place cell not only increases its spike rate, but also spikes at earlier phases in the theta cycle (Figure 6.1). One theoretical utility of the phase code is that it enables several patterns to be active simultaneously for storage or recall [90]: Neurons with overlapping place fields can be active at different phases of the

[1] The authors measured CA1 place cells, but those in CA3 have similar properties.

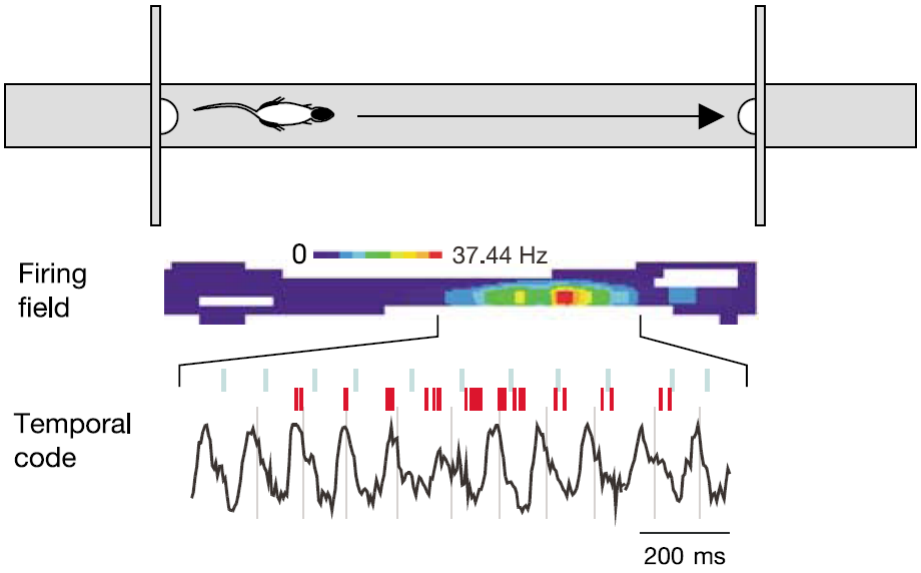

Figure 6.1: Place Cell Encoding

Place cells encode location information in both their spike rate (*middle*) and their spike timing (*bottom*). As the animal traverses the place field of a given neuron, the phases of its spikes relative to the theta rhythm (black) decrease, from [67].

theta rhythm, thus remaining independent and distinguishable.

To implement a phase code, the theta rhythm is thought to prevent spiking until the input synaptic current exceeds the sum of the neuron threshold and the decreasing inhibition on the downward phase of the cycle [104]. However, even with identical inputs and common theta inhibition, neurons do not spike in synchrony. Variability in excitability spreads the activity in phase. Lethargic neurons (such as those with high thresholds) spike late in the theta cycle, since their input exceeds the sum of the neuron threshold and theta inhibition only after the theta inhibition has had time to decrease. Conversely, excitable neurons (such as those with low thresholds) spike early in the theta cycle. Consequently, variability in excitability translates into variability in timing.

I hypothesize that the hippocampus achieves its precise spike timing (about 10ms) through *plasticity enhanced phase-coding* (PEP) (Figure 6.2). The source of hippocampal timing precision in the presence of variability (and noise) remains unexplained. Synaptic plasticity can compensate for variability in excitability if it increases excitatory synaptic input to neurons in inverse proportion to their excitabilities. Recasting this in a phase-coding framework, I desire a learning rule that increases excitatory synaptic input to neurons directly related to their phases. Neurons that lag require additional synaptic input, whereas neurons that lead require none. STDP observed in the hippocampus satisfies these criteria, as it requires repeated pre-before-post spike pairings (within a time window) to potentiate and repeated post-before-pre pairings to depress a synapse [13].

In the PEP model, STDP at recurrent synapses not only compensates for variability among neurons, (in theory) it also stores stimulus activity patterns. When neurons in a pattern are coactive, STDP potentiates many of the synapses among them. This is similar to programming a pattern into a Hopfield network by increasing the recurrent synaptic weights among neurons in the pattern [62]. These strengthened recurrent connections store the pattern: When a subset of the pattern is activated, the recurrent connections from the active to inactive neurons are excited, causing the remaining neurons in the original pattern to spike, which realizes pattern recall.

6.2 Previous Silicon Autoassociation

Associative memory is central to neural computation, but general purpose computers are poorly suited to such processing [102], motivating engineers to build associative memories in hardware. Although their implementations have had little to do with the biological

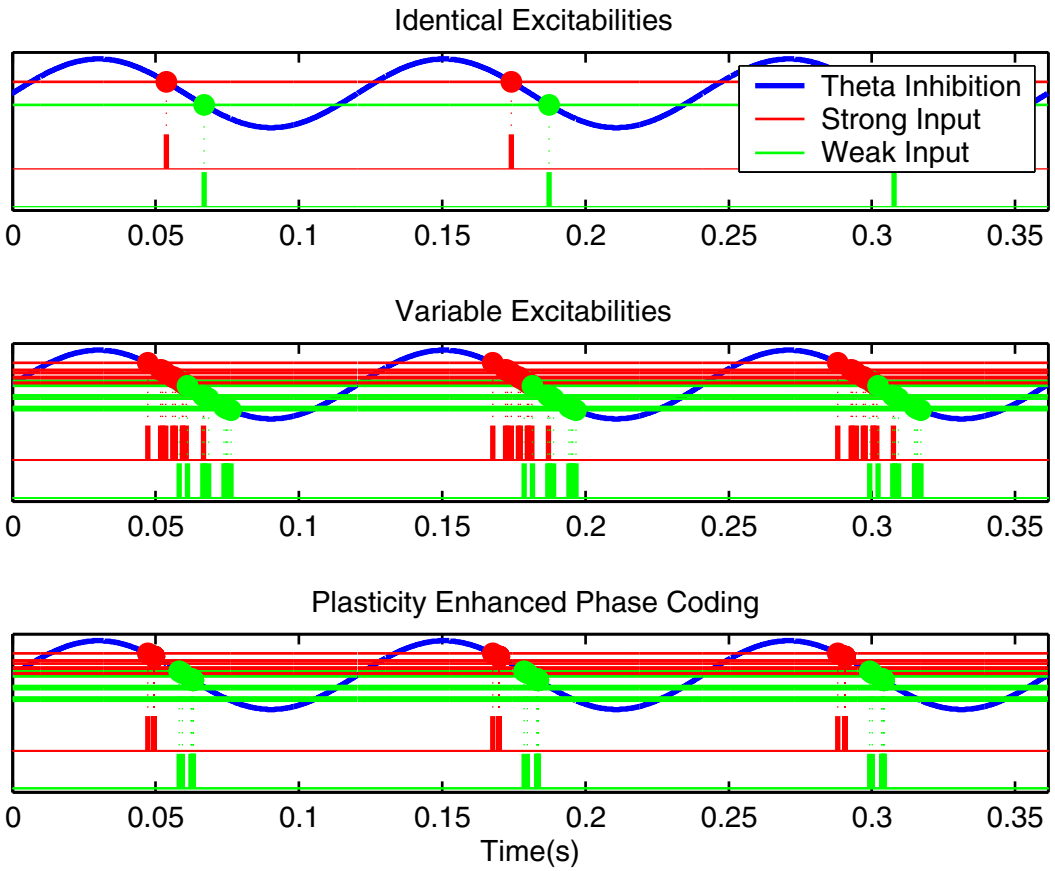

Figure 6.2: Plasticity Enhanced Phase-coding

Top Neurons with strong input spike early in phase (red) and those with weak input spike late (green), encoding input strength with phase, because the theta rhythm prevents spiking until the input synaptic current exceeds the decreasing inhibition on the downward phase of the cycle [104]. *Middle* Variability in excitability degrades spike-timing precision; excitable neurons (with low thresholds) spike earlier, whereas lethargic neurons (with high thresholds) spike later. *Bottom* Lethargic neurons spike with excitable neurons, because plasticity enhanced phase-coding preferentially potentiates synapses from excitable to lethargic neurons, providing additional synaptic drive.

CA3, they face similar tradeoffs and therefore can inform design decisions, such as the appropriate level of abstraction for building the silicon CA3. Engineers have built VLSI autoassociative neural networks of both analog, digital, and mixed circuit types, using either nonspiking or spiking neurons. I focus on networks with recurrent architectures, inspired by the Hopfield network [62, 63].

I categorize these recurrent networks into two classes: abstract and neuromorphic. These classes can be distinguished by the type of neurons that they use: Abstract networks use nonspiking neurons, whereas neuromorphic networks use spiking neurons. Nonspiking neurons encode information instantaneously in their (analog or digital) activation level; spiking neurons can encode information by their activation level (spike rate), by whether or not they are active (over a time interval), or by the time at which they spike. As I will show, the differences extend deeper than merely their neurons and their encoding.

The differences between abstract and neuromorphic networks arose due to the distinct goals associated with each type. Abstract associative networks developed out of engineers' desire to compete with general purpose processors in tasks that could benefit from pattern recognition and optimization, computations at which associative networks excel [64]. These objectives required that abstract networks be fast but put minimal demand on their power requirements. Neuromorphic networks (not necessarily associative) developed out of engineers' desire to exploit the computational efficiency of neurobiological networks, especially in portable processing that involves adapting to and processing information about the environment [102]. These objectives required that neuromorphic networks be power efficient but only required them to be fast enough to interact with the environment, identical to the demands on neurobiological systems. Further, neuromorphic associative networks encode information in the same manner believed to be used by biological networks, including encoding several patterns simultaneously (Section 6.1).

Abstract networks' emphasis on speed, rather than power efficiency, enabled engineers to implement them in various design modes to optimize performance. The networks were implemented with custom analog [130, 66, 26, 144] or digital [77] circuits, and more recently on field-programmable gate arrays (FPGAs) [109, 93][2]. Although they were constructed using different design methodologies, they shared several properties, such as numerous high-speed synapses that were not only power hungry but consumed silicon area. Traditionally, associative networks were fully connected, requiring nearly N^2 connections for N neurons (usually with binary synapses). Intuitively, full connectivity preserved the general purpose nature of the abstract network and maximized its memory capacity for a given number of neurons [119]. One typical fully-connected (14-neuron) analog network converged to a stored pattern in 150ns, enabling to it recognize up to 6.7 million patterns per second [144]. Unfortunately, it used 50μW per synapse, which would result in over 10W, the power consumption of the entire brain, in a fully connected network of only 450 neurons.[3] Space constraints have limited the size of abstract networks to well under a hundred neurons in any implementation. Ultimately, abstract networks failed to find a niche before general purpose processors improved enough to take over the functions for which they were intended.

Neuromorphic networks' emphasis on biological structure and function require engineers to use an efficient medium, such as application specific integrated circuits (ASICs), which can be designed to emulate dense neuron arrays in real-time by using mixed ana-

[2]Contrary to intuition, analog abstract network implementations were the fastest: Neurons summed inputs from the rest of the population by summing currents from their synapses, which requires little area. In contrast, digital (including FPGA) implementations were slower because they shared hardware. Parallel computations would require nearly N^2 multiply–add circuits for N neurons, which would be extremely large. Instead, a single neuron (or set of neurons) would be multiplexed to calculate the outputs of all neurons of the present time step sequentially, given their states at the last time step and the synaptic states. Hence, the digital implementations were slower than the analog implementations but required less area.

[3]To be as efficient as the brain's 10^{15} synapses, each one can only use 10fW (10^{-14}), a factor of 5 billion less than this silicon model [144].

log and digital circuits with on- or off-chip recurrent connectivity. Abstract associative networks have benefited from using ASICs as well. Neuromorphic associative networks require each neuron to connect only sparsely to the rest of the population (on the order of 10% in the biological CA3 [78]), enabling chips to include more neurons (than a fully connected implementation) without increasing the silicon area or power budgets. ASICs have similar interchip connectivity properties to FPGAs[4], but can realize more neurons and use less power, enabling fewer chips to achieve the same amount of processing. Further, ASICs are necessary to efficiently morph neurobiological structure and function, performing computation with analog circuits and communicating with digital circuits. Neuromorphic chips have realized as many as 9,216 spiking neurons on a single chip [106] (compared to 500 neurons with similar complexity and no connectivity on an FPGA [149]) and 35,648 in a multichip system [27]. Further, significant work is being done on realizing programmable and expandable neuromorphic systems [108][88].

Two fundamental differences between abstract and neuromorphic implementations of autoassociation are learning and variability. In both abstract and neuromorphic networks, learning is realized by increasing the connectivity strength among neurons in the same pattern. In abstract networks, learning is separate from recall and is usually performed offline, often in the form of training on a dataset, consisting of all patterns that the network will need to recall. During recall, synapses are fixed so no learning takes place. In neuromorphic networks, learning is continuous: Synapses can change state when a previously

[4]Researchers have used programmable logic to simulate detailed spiking neurons in real-time [149]: FPGAs seem like a natural choice for neural modeling hundreds of neurons because they are dense and reconfigurable. However, FPGAs are fine grained, meaning that they have many small logic blocks that only connect to other nearby blocks. Such an architecture can simulate many neurons but cannot implement massive interconnectivity (for associative memory) without spending most of its resources on connections (which increases further for plastic connections). Clearly, course grained complex programmable logic devices (CPLDs), where each block connects globally or nearly globally, are superior for modeling massive connectivity, but they lack the density of FPGAs and are capable of modeling only a handful of neuron circuits. Model neurons can be instantiated across many FPGAs but then one of their greatest assets, high internal bandwidth is lost.

learned pattern is being recalled or when a new pattern is presented. Both abstract and neuromorphic networks must manage intrinsic variability among elements (as well as extrinsic variability in inputs). Abstract networks use traditional engineering methods to manage intrinsic variability by building a few expensive (in both cost and power) and precise elements. Conversely, inspired by biology, neuromorphic systems limit intrinsic variability by building many inexpensive and imprecise elements and actively compensating for intrinsic (and extrinsic) variability as needed.

My interest lies in building a large scale neuromorphic associative memory—one that uses biological principles and structure both to store and to recall patterns. Specifically, I desire a sparsely connected network of spiking neurons with plastic synapses, similar to CA3. No large-scale neuromorphic associative memory networks have previously been built, but related neuromorphic systems bear mention.

Although not a recurrent associative network, a neuromorphic large-scale learning system has previously been built [138]. Although functionally and structurally distinct from the CA3, it merits mention. This system learned topography in feedforward connections by identifying correlated inputs and grouping them together: Neurons that fired together wired together. It grouped inputs by moving their axons to nearby locations, similar to biological axon remodeling during development. This architecture provided two inherent mechanisms to compensate for variations among elements: Diffusive signals spatially filtered the output of strongly and weakly active neurons, and axons that moved inappropriately had many opportunities to sense their error and return. This system lacked feedback and did not recall patterns but it learned and stored information based on correlations, using a number of connections proportional to N rather than N^2.

A small scale, neuromorphec chip has been built that used a connectivity pattern similar

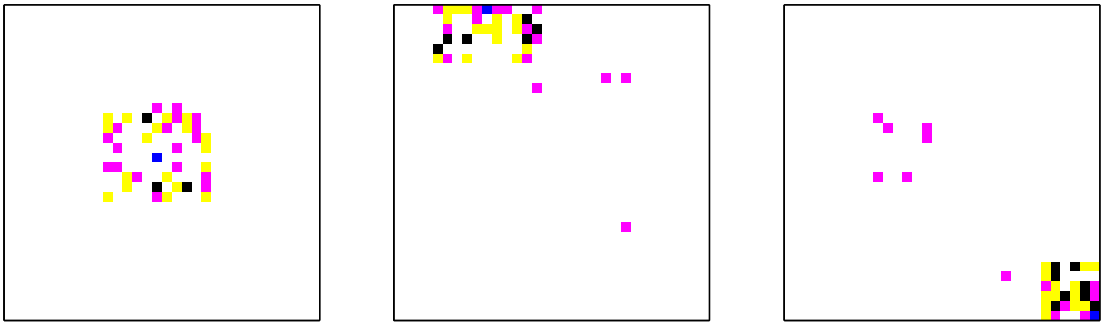

Figure 6.3: Pyramidal Neuron Connectivity

Each pyramidal neuron (blue) sends (magenta) and receives (yellow) randomly chosen STDP synapses to and from neurons up to 5 nodes away (black indicates both). Neurons near chip boundaries (*middle and right*) may send some connections farther away, when all local synapses are occupied.

to the CA3 to store a pattern [25]. This system used 14 neurons with 56 plastic recurrent synapses (29% N^2). The authors used this network to store a pattern; they verified its storage by observing changes in synaptic states, however, they did not attempt to recall it. This chip attempted to implement associative memory but it did not realize precise timing, an important aspect of the CA3 memory system. The chip lacked any mechanism to compensate for the considerable variations among its neurons, because its plastic synapses lacked timing dependence, a necessity in compensating for variability to achieve precise timing (in the PEP model).

6.3 Neuromorphic CA3

The STDP Chip models the associative memory of the CA3 region of the brain. The autoassociative network consists of 1,024 silicon pyramidal neurons recurrently connected

by 21,504 binary synapses that express STDP, enabling the network to learn stimulated patterns of activity and to compensate for variability in excitability. Specifically, STDP preferentially potentiates synapses that project from excitable neurons, which spike early, to lethargic neurons, which spike late. Because the potentiated synapses connect coactive neurons, they store the stimulated pattern; once learned, the entire pattern can be recalled by stimulating a subset. In addition to storing the pattern, the additional excitatory synaptic current makes lethargic neurons spike earlier, thereby causing neurons that belong to the same pattern to spike in synchrony, which realizes a temporal code.

I configured the STDP Chip as a recurrent network (Section 3.4). Briefly, a CPLD implements the recurrent connections by accessing a lookup table, stored in a RAM chip. The CPLD also connects to a USB interface chip, which provides a bidirectional link with a PC, allowing the PC to vary the leak current (via a digital to analog converter) in real-time to create the global inhibitory theta rhythm.

Each neuron received recurrent connections from 21 randomly selected neurons within an 11 by 11 neighborhood centered on itself (see Figure 6.3). It made recurrent connections to randomly chosen neurons within the same neighborhood. These connections were mediated by STDP circuits, initialized to the depressed state. In the coming section, I present the pyramidal neuron and a network of these neurons.

6.3.1 Pyramidal Neuron

The pyramidal neuron (Chapter 3) consists of a refractory period and calcium-dependent potassium circuit (RCK), a synapse circuit, and a soma circuit. RCK inhibits the neuron, analogous to a fast potassium channel population (refractory) and a slower calcium de-

pendent population (adaptation). The synapse circuit excites the neuron; its current rises (nearly) linearly and falls exponentially. The soma circuit is a leaky integrator; it receives excitation from the synapse circuit and shunting inhibition from RCK as well as a constant leak shunt. The soma circuit is connected to an axon hillock circuit, the locus of spike generation, which consists of a model voltage-dependent sodium channel population. It initiates the AER signaling process required to send a spike off chip.

6.3.2 Pyramidal Network

To characterize the pyramidal network's variability, I excited 81 neurons with Poisson-like 58Hz spike trains (Figure 6.4 *Top left*). I made these spike trains Poisson-like by starting with a regular 200Hz spike train and dropping spikes randomly, with probability of 0.71. Thus spikes were delivered to neurons that won the coin toss in synchrony every 5ms. However, neurons did not lock onto the input synchrony due to filtering by the synaptic time constant (see Figure 6.4 *Top left*). They also received a common inhibitory input at the theta frequency (8.3Hz), via their leak current[5]. Each neuron was prevented from firing more than one spike in a theta cycle by its model calcium-dependent potassium channel population.

The pyramidal neurons' spike times were variable. To quantify the spike variability, I used timing precision, defined as twice the standard deviation of spike times accumulated from five theta cycles. With an input rate of 58Hz the timing precision was 34ms.

[5]In this experiment, I used a sinusoidal voltage to drive the theta inhibition. The resulting current was an exponential of this voltage, which provided theta inhibition similar to a triangular wave: All neurons are strongly inhibited, then released from inhibition

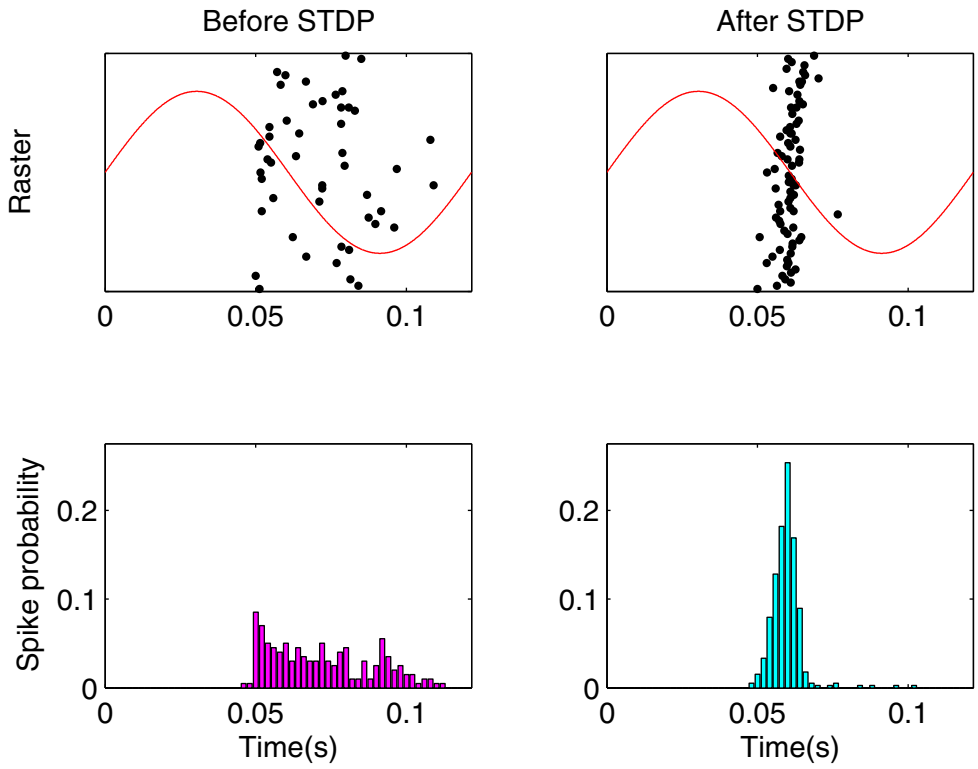

Figure 6.4: STDP Variability Compensation

Spike rasters and histogram of 81 neurons (9 by 9 cluster) with common theta inhibition (red) display timing variability due to neuronal variability and noise (58Hz input) before STDP; after five seconds of STDP, spikes are synchronous. Neurons receive a common theta input (8Hz) and are limited to one spike per theta cycle by the model calcium-dependent potassium channel population.

6.4 Neuromorphic Autoassociation

As described in Section 6.2, managing variability is critical in implementing autoassociation. In this section, I describe the silicon CA3's ability to compensate for variability while learning patterns, which can later be recalled.

6.4.1 Variability Compensation

I carried out an experiment designed to test the STDP circuit's ability to compensate for variability in spike timing through PEP. I chose a 9 by 9 cluster of neurons and delivered spikes at a mean rate of 50 to 100Hz to each one (dropping spikes with a probability of 0.75 to 0.5 from a regular 200Hz train) and provided common theta inhibition as before.

I compared the variability in spike timing after five seconds of learning with the initial distribution. Phase coding was enhanced after STDP (Figure 6.4). Before STDP, spike timing among neurons was highly variable, except for the very highest input rate. After STDP, variability was virtually eliminated, except for the very lowest input rate (Figure 6.5). Initially, the variability, characterized by timing precision, was inversely related to the input rate, ranging from 34 for lowest input rate down to 13ms for the highest input rate. After five seconds of STDP, variability decreased and was largely independent of input rate, falling below 11ms (Figure 6.6).

To confirm that PEP enables the Neuromorphic CA3 to encode multiple simultaneous patterns at different phases of the theta rhythm, I stimulated two patterns with different mean input rates (generated as before): I stimulated one pattern with a regular 200Hz spike train (100 neurons shaped like a **U**) and the other at a Poisson-like 67Hz spike train (94

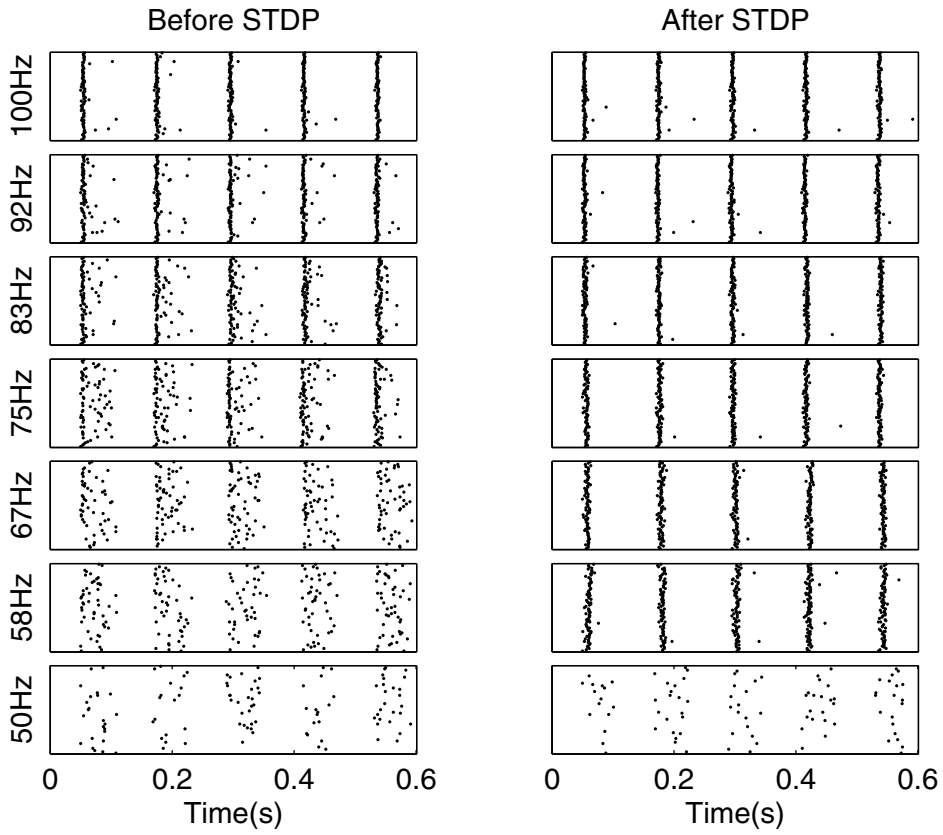

Figure 6.5: Variability Compensation at Different Rates

Before STDP, variability in spikes times (top) depends on input rate; after STDP (five seconds), variability is independent of input rate (for input rates above 50Hz). Timing precision, defined as twice the standard deviation of spike times from five theta cycles, decreases to below 11ms.

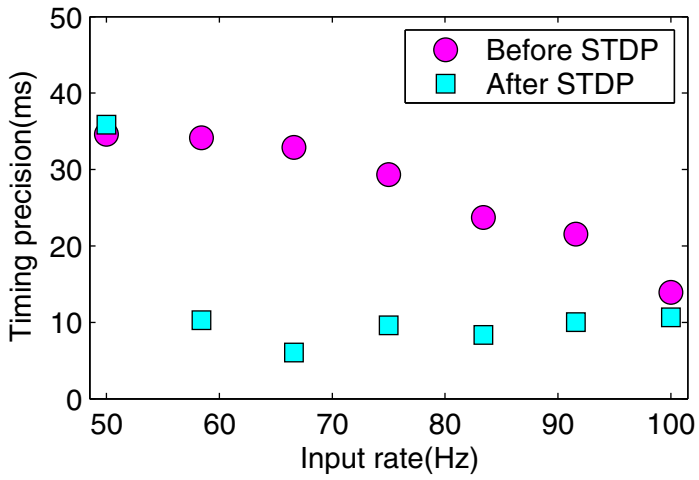

Figure 6.6: STDP Improves Timing Precision

Before STDP, variability in spikes times depends on input rate; after STDP (five seconds), variability is independent of input rate (for input rates above 50Hz). Timing precision, defined as twice the standard deviation of spike times from five theta cycles, decreases to below 11ms.

neurons shaped like a **P**). I placed the patterns at diametric corners to eliminate interactions between them. For the high input-rate pattern, the timing precision remained nearly constant; initially it was 1.9ms and after 10 seconds of STDP it was 1.8ms. For the low input rate pattern, the timing precision improved from 18ms to 7.8ms. After learning, the patterns fired at different phases of the theta rhythm with the high input pattern leading by 8ms (Figure 6.7). When the pyramidal neurons drive globally inhibitory interneurons (Chapter 4), the patterns separation can be increased to tens of milliseconds, depending on the strength and time course of the inhibition (data not shown). By ensuring the patterns are separated by tens of milliseconds (one gamma period), I provide appropriate timing information to link sequential patterns together (Chapter 2).

Comparing the number of potentiated synapses each neuron made or received with its

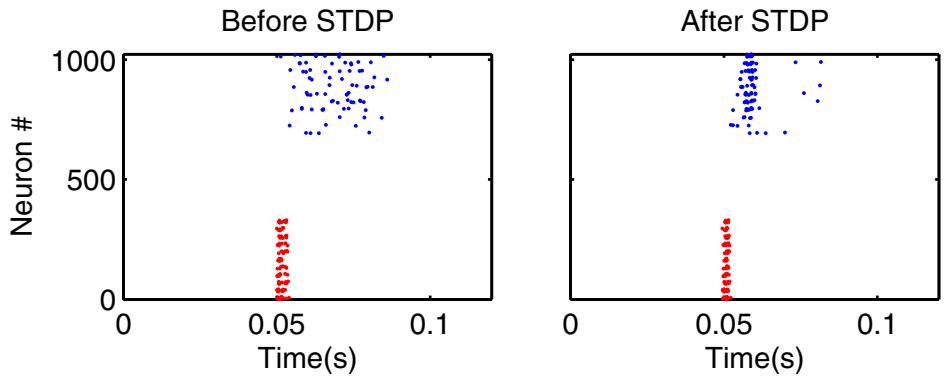

Figure 6.7: Coactive Patterns

Plasticity enhanced phase-coding enables two independent patterns to be active simultaneously, firing at different phases of the theta rhythm. *Left* Before STDP, neurons in the pattern with the higher input rate (red) spike early in phase (timing precision = 1.9ms), whereas neurons in the pattern with the lower input rate (blue) spike later and more dispersed in phase (timing precision = 18ms). *Right* After STDP the phase relationship is maintained but the timing precision of the lower input rate pattern improves (7.8ms).

excitability confirmed the PEP hypothesis (i.e., leading neurons provide additional synaptic current to lagging neurons via potentiated recurrent synapses). In this experiment, to eliminate variability due to noise (as opposed to excitability), I provided a 17 by 17 cluster of neurons with a regular 200Hz excitatory input. Theta inhibition was present as before and all synapses were initialized to the depressed state. After 10 seconds of STDP, a large fraction of the synapses were potentiated (Figure 6.8). When the number of potentiated synapses each neuron made or received was plotted versus its rank in spiking order (Figure 6.9), a clear correlation emerged (r = -0.71 or 0.76, respectively). As expected, neurons that spiked early made more and received fewer potentiated synapses. In contrast, neurons that spiked late made fewer and received more potentiated synapses.

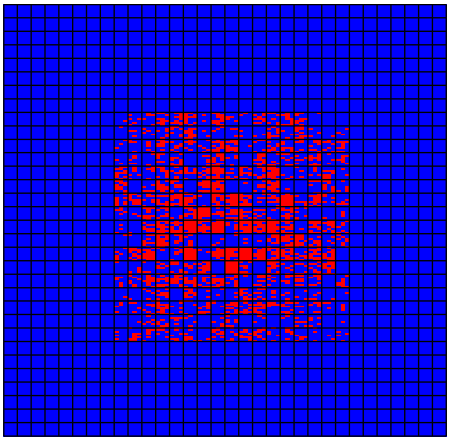

Figure 6.8: Synapse Potentiate During Stimulation

Although initialized to the depressed state (blue), numerous synapses transitioned to the potentiated state (red) after STDP (10 seconds); they were driven with a regular 200Hz input train and common theta inhibition.

6.4.2 Pattern Storage and Recall

After STDP, the network could recall an entire pattern given a subset, thus the same mechanisms that compensated for intrinsic variability and noise could also compensate for lack of information. I chose a 9 by 9 cluster of neurons as the input pattern and delivered a Poisson-like spike train with mean rate of 67Hz to each neuron as in the first experiment. Theta inhibition was present as before and all synapses were initialized to the depressed state. Before STDP, I stimulated a subset of the pattern and only neurons in that subset spiked (Figure 6.10 *Left*). After five seconds of STDP, I stimulated the same subset again. This time they recruited spikes from other neurons in the pattern, completing it (Figure 6.10 *Right*).

Upon varying the fraction of the pattern presented, I found that the fraction recalled increased faster than the fraction presented. I selected subsets of the original pattern ran-

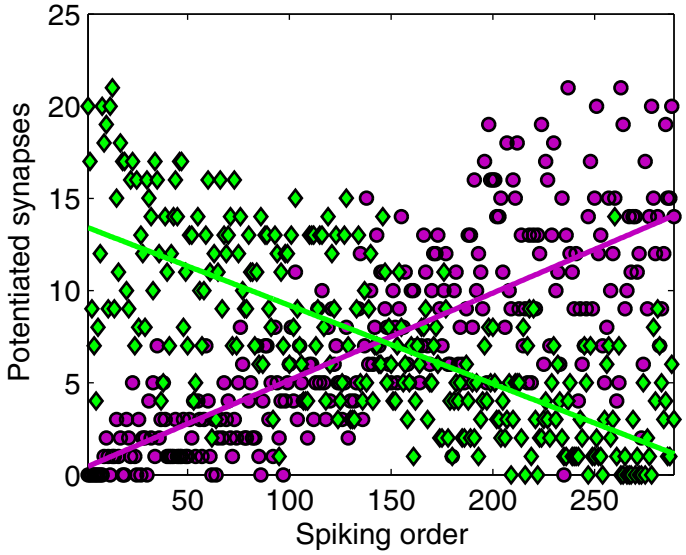

Figure 6.9: STDP Realizes PEP

The number of potentiated synapses that neurons make (green) and receive (purple) is negatively (r = -0.71) and positively (r = 0.76) correlated with their rank in the firing order, respectively. All neurons (17 by 17 cluster) received a regular 200Hz input train and common theta inhibition.

domly, varying the fraction of neurons chosen from 0.1 to 1.0 (ten trials for each). I classified neurons as active if they spiked in the two second period over which I recorded. Thus, I characterized PEP's pattern-recall performance as a function of the probability that the pattern in question's neurons are activated (Figure 6.11). At a fraction of 0.50 presented, nearly all of the neurons in the pattern are consistently activated (0.91 ± 0.06), showing robust pattern completion. I fit the recall performance with a sigmoid that reached 0.50 recall fraction with an input fraction of 0.30; the slope was 2.66. No spurious neurons were activated during any trials.

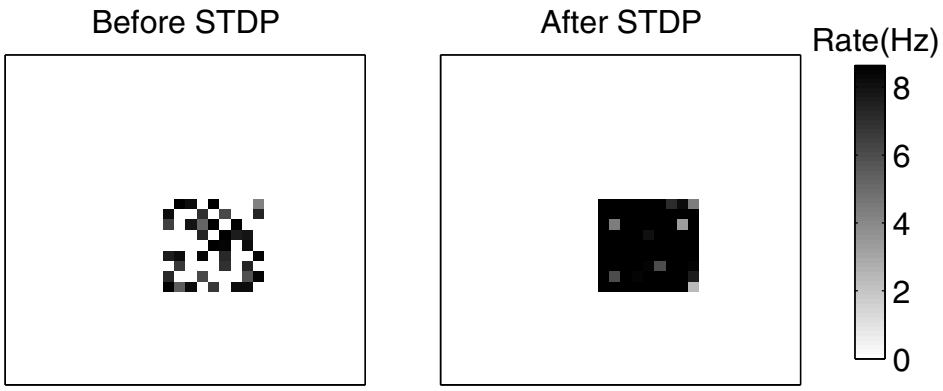

Figure 6.10: Pattern Completion

Before STDP (left), half (40) of the neurons in a pattern (9 by 9 cluster) are stimulated; only they are activated. After inducing STDP (right) while stimulating the whole pattern, almost the whole pattern is activated when half of the neurons are stimulated.

6.5 Sources of Spike-Timing Variation

In Section 6.4, I showed that the silicon CA3 uses STDP to learn patterns while simultaneously improving spike-timing precision. Because timing precision is important to both neuromorphic and biological systems, I characterized the properties of spike-timing precision, focusing on STDP's role in compensating for the two sources of variability: intrinsic fabrication variability (fixed mismatch) and extrinsic input variability (temporal jitter). Mismatch varies the time at which different neurons spike, although when driven with constant (or regular) input those times are consistent from one theta cycle to another. Input noise jitters the time each neuron spikes across theta cycles. I explore timing precision with a modest recurrent synaptic strength (when potentiated) and modest calcium-dependent potassium channel population strength, tuned to prevent neurons from spiking more than once per theta cycle.

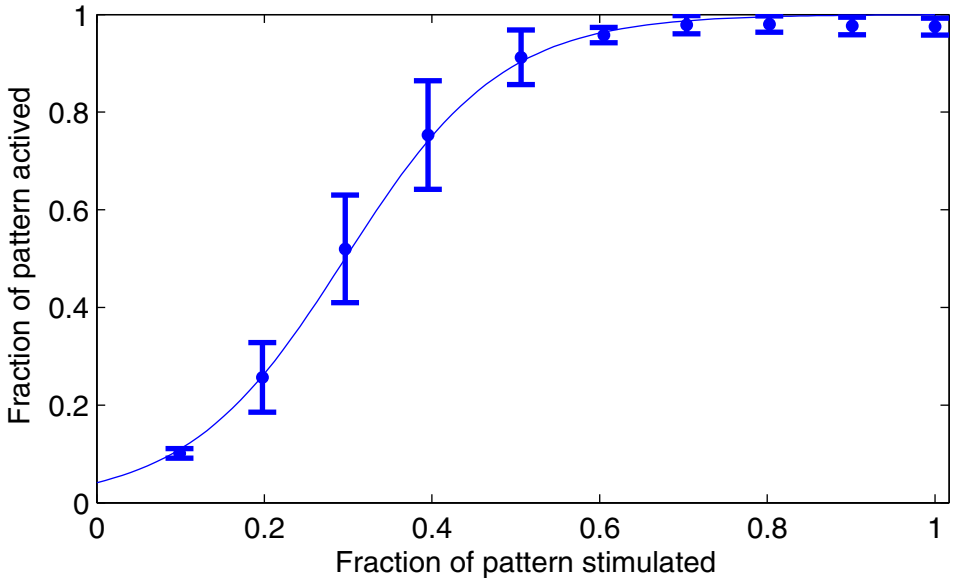

Figure 6.11: Pattern Completion Depends on Fraction of a Pattern Activated

The fraction of the pattern activated increases faster than the fraction stimulated. Potentiated recurrent synapses recruit the remaining neurons in the original pattern. Stimulating half of the neurons in the pattern activates 91±5.6 percent of neurons in the original pattern, showing robust pattern completion (Error bars are standard deviation).

Qualitatively, timing precision improves with increasing synaptic strength[6], whereas autoassociative networks have the highest (information) storage capacity with (binary) recurrent synapses of intermediate strength [119]. When recurrent synapses are too strong, a pattern can only have limited overlapping members with other patterns without strong interference. When recurrent synapses are too weak, most members of a pattern must be active before others will be recalled. Here, I adopt a synaptic strength approximately equal

[6]I have explored the absolute limit of timing precision in the network, achieving precision below 1ms (data not shown). But this result requires unreasonably strong recurrent synapses and calcium-dependent potassium (KCA). A strong recurrent synapse causes a spike in its target in less than 1ms, and a strong KCA prevents the target from spiking twice in one theta cycle. Further increase in both synaptic strength and KCA results in some excitable neurons spiking twice per cycle (because the synaptic input from the most excitable neuron overpowers KCA in some neurons) and lethargic ones spiking only intermittently.

to the one used previously, which recalls most of a pattern when half of its members are stimulated (Figure 6.11). Further, I use a recurrent synaptic decay constant of approximately 50ms, in between measured values for AMPA and NMDA components of synapses in the hippocampus [34] and a somatic time constant of 10ms (at the theta inhibition's minimum). It is important to consider the effect of NMDAR-gated channels, because in biology they contribute most charge at excitatory synapses (See Section 2.3.3.2).

To characterize spike-timing precision, I stimulated a patch of 100 neurons (10 by 10 box), observing the influence of STDP on both timing mismatch and jitter. I drove all neurons with a global theta inhibitory current (conductance) at 8.75Hz[7]. I observe spike-timing precision and its components across 25 theta cycles before STDP (turned off) and across 25 cycles after 15 seconds[8] of STDP.

I calculated measures of spike-timing precision, timing mismatch and timing jitter, given by σ_{t}, σ_{m}, and σ_{j}, respectively. To calculate spike-timing precision, I considered the standard deviation of all spike times in each of 25 theta cycles. The mean of these standard deviations is σ_{t} (one-half of timing precision). To calculate timing mismatch, I considered each neuron's mean spike time across 25 theta cycles: The standard deviation of these mean spike times across the population is σ_{m}. To calculate timing jitter, I considered each neuron's standard deviation in spike times across 25 theta cycles. The mean of these standard deviations across the population is σ_{j}.

[7] In this experiment, I used a log-sin voltage to drive the theta inhibition. The resulting inhibition was a sinusoid, which I verified by observing a neuron's membrane potential, driven with a constant excitatory current with the axon hillock turned off (not shown).

[8] Without noise, the system ceases to lower its spike-timing variations after one second, however when the noise is increased, it can take up to 10 seconds. 15 seconds is long enough to give the system time to integrate information across many cycles even in the noisiest case.

6.5.1 Components of Variation

Both mismatch and jitter degrade spike-timing precision. Mismatch is fixed (for a given set of biases), whereas I impose jitter by stimulating the neurons with (independently generated) noisy input currents, keeping each current's mean constant but varying its standard deviation. I generated a noisy input current by stimulating the neurons' AMPA synapses, driving them with input spike trains generated by dropping spikes from a regular 4kHz input, keeping each spike with probability λdt, where dt $= \frac{1}{4\text{kHz}}$, and λ is the desired input rate[9]. The resultant distribution is a Bernolli distribution (binomial distribution with one sample per trial). To maintain the mean input current, I scale the AMPA synapse strength, $I_{\text{A}} = \frac{I_\mu}{(\lambda\text{dt})^\beta}\frac{\text{dt}}{\text{w}}$, in inverse proportion to the average input frequency, where I_μ is the desired mean current, β is a near unity error factor[10], and w is the input pulse width. The resulting current standard deviation, I_σ is given by:

$$\begin{aligned} I_\sigma &= I_{\text{A}}\frac{\text{w}}{\text{dt}}\sqrt{\lambda\text{dt}(1-\lambda\text{dt})} \\ &= I_\mu\sqrt{(\lambda\text{dt})^{1-2\beta} - (\lambda\text{dt})^{2-2\beta}} \end{aligned} \tag{6.1}$$

Because the theta rhythm decreases approximately linearly in the interval during which neurons spike, I assume that σ_{j} is proportional to I_σ, given by:

[9]The generated excitatory postsynaptic current is in the shape of a pulse with height I_A and width w, which is less than dt (see Chapter 3).

[10]I decrease λ exponentially and increase the AMPA strength bias voltage, V_{W} linearly, resulting in an exponential increase in W: I attempt to match the two exponentials as well as possible by fitting the AMPA strength, I_{A} versus its bias voltage for one neuron (data not shown). The differences in the exponential rates is the error factor, β.

$$\sigma_{\mathrm{j}} = \alpha\sqrt{(\lambda \mathrm{dt})^{1-2\beta} - (\lambda \mathrm{dt})^{2-2\beta}} \tag{6.2}$$

where α is a proportionality constant.

To observe the mismatch and jitter components of variation both before and after STDP, I stimulated the patch of neurons using input spike trains with average frequencies, λ, ranging from 4Hz to 4kHz, scaling the AMPA strength as described above. For λ above 74Hz, most neurons fire one spike per theta cycle, whereas for lower λ, lethargic do not spike in every cycle; I_{A} is large enough to overcome RCK, enabling excitable neurons to spike twice in a theta cycle.

I fit σ_{m} by taking its mean value across all frequencies from 74Hz to 4kHz (Figure 6.12). Before STDP, $\langle\sigma_{\mathrm{m}}\rangle = 4.8$ms (where $\langle x\rangle$ is the mean of x); after STDP it decreased to 2.1ms. As expected, σ_{m} was nearly constant before STDP with a maximum deviation from its mean of 0.55ms. Similarly, after STDP it expressed a maximum deviation of 0.55ms. Across this range of frequencies and jitters, STDP compensated for mismatch equally well.

I fit σ_{j} to Equation 6.2 for λ ranging from 74Hz[11] to 3.4kHz (Figure 6.12). Before STDP, the proportionality factor, α was equal to 1.54ms; after STDP it decreased to 0.99ms. The error factor, β, was equal to 0.93 and 0.96 before and after STDP, respectively. Since β was nearly constant and close to unity before and after STDP, I see from Equation 6.2 that the before/after ratio of timing jitter, $\frac{\sigma_{\mathrm{j\,PRE}}}{\sigma_{\mathrm{j\,POST}}}$, is approximately equal to the ratio of proportionality constants, $\frac{\alpha_{\mathrm{PRE}}}{\alpha_{\mathrm{POST}}}$, which was equal to 1.57. Therefore, over this range of frequencies

[11] Below 10Hz, neurons receive infrequent input, potent enough to overcome RCK. Each neuron's spikes reflect its Poisson input, which is distributed uniformly in time. σ_{j} saturates near $\frac{1}{8.75\mathrm{Hz}\sqrt{12}}$, which is the standard deviation of a uniform distribution over an 8.75Hz theta cycle.

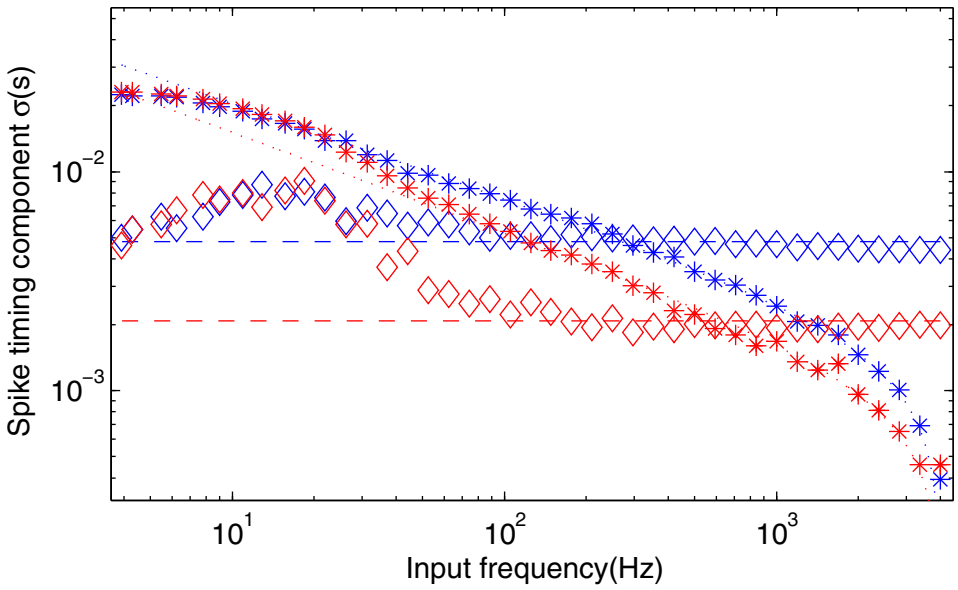

Figure 6.12: Components of Spike-Timing Variation

Before STDP, timing mismatch (blue ⋄) is approximately constant for frequencies above 74Hz (blue dashed line). After STDP, its value is reduced (red ⋄) and constant above 100Hz (red dashed line). Before STDP, timing jitter (blue *) depends on input frequency, fit between 74Hz and 3.4kHz (blue dotted line). After STDP, it is reduced (red *); fit between 74Hz and 3.4kHz (red dotted line).

and corresponding jitters, STDP compensates in proportion to the level of jitter. Intuitively, potentiated synapses narrows the jitter distribution by providing additional current to target neurons, which reduces the noisy input necessary to drive the membrane above threshold, enabling neurons to spike earlier.

6.5.2 Timing Variation

I measured and fit the mismatch and jitter components of variation. Here, I use those measurements and fits to characterize σ_t, (one-half) the spike-timing precision.

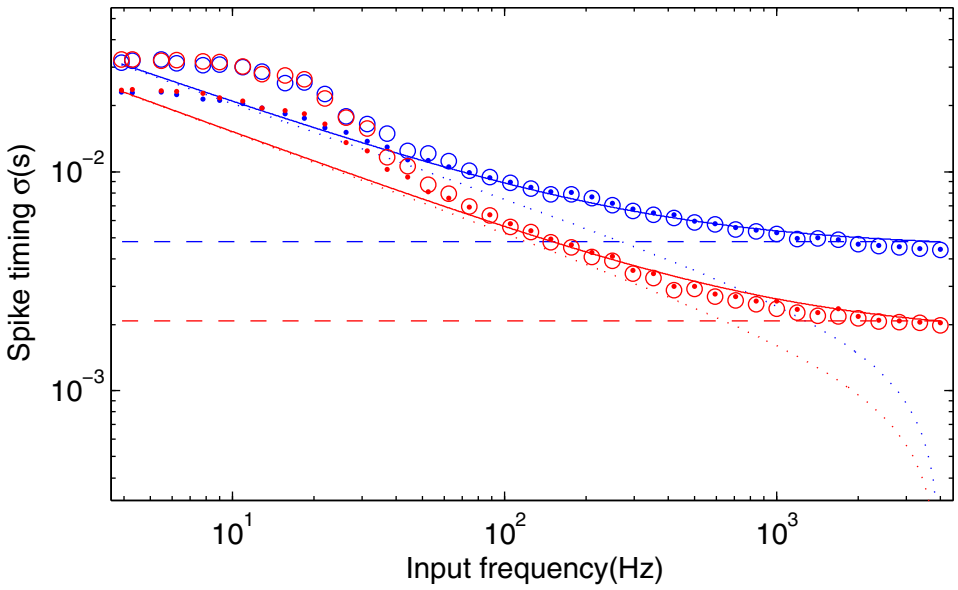

Figure 6.13: Spike-Timing Variation

Both before (blue) and after (red) STDP, the spike-timing standard deviation (○) is approximately equal to the square root of the sum of the mismatch and jitter variances (•) and is well fit (solid lines) by combining the fits (dashed and dotted lines, respectively) for these components from Figure 6.12.

The contributions to variance from mismatch and jitter are additive (Figure 6.13). Mismatch varies the time at which different neurons spike, although when driven with constant (or regular) input each neuron spikes consistently from one theta cycle to another. Jitters varies the time at which each neuron spikes across theta cycles. The distribution of spike times due to mismatch is jittered by the temporal noise, convolving the mismatch distribution with the jitter distribution (both approximately gaussian), which results in a new (gaussian) distribution whose variance is the sum of the mismatch and jitter variances. Therefore, its standard deviation is given by:

$$\begin{aligned} \sigma_{\mathrm{t}} &= \sqrt{\sigma_{\mathrm{m}}^2 + \sigma_{\mathrm{j}}^2} \\ &= \sqrt{\sigma_{\mathrm{m}}^2 + \alpha^2[(\lambda \mathrm{dt})^{1-2\beta} - (\lambda \mathrm{dt})^{2-2\beta}]} \end{aligned} \tag{6.3}$$

I combined σ_{m} and σ_{j} as described in Equation 6.3 to estimate σ_{t}; I found a good match for λ ranging from 74Hz to 4kHz. Before STDP, the estimate is within 4 percent (maximum deviation of 230μs). After STDP, the estimate is within 8.5 percent (maximum deviation of 200μs). These results demonstrate that not only does STDP compensate for imprecise timing due to mismatch; it also compensates for imprecise timing due to noise in the system.

6.6 Discussion

These results demonstrate that PEP successfully compensates for graded variations in my silicon recurrent network using binary weighted synapses (in contrast with [24], where weights are graded). While the chip results are encouraging, variability was not eliminated in every case. In the case of the lowest input (50Hz), I see virtually no change (Figure 6.5). I suspect the timing remains imprecise because, with such low input, neurons do not spike every theta cycle and, consequently, provide fewer opportunities for the STDP synapses to potentiate. This shortfall illustrates the system's limits; it can only compensate for variability within certain bounds, and only for activity appropriate to the PEP model.

As expected, STDP is the mechanism responsible for PEP. STDP potentiated recurrent synapses from leading neurons to lagging neurons, reducing the disparity among the diverse population of neurons. Even though the STDP circuits are themselves variable, with different efficacies and time constants, when using timing the sign of the weight-change is always correct (Chapter 5). For this reason, I chose STDP over other more physiological implementations of plasticity, such as membrane-voltage-dependent plasticity (MVDP), which has the capability to learn with graded voltage signals [48], which is useful in active dendrites, providing more computational power [121].

My results demonstrate that STDP compensates for input noise in addition to intrinsic variability. The effect of input noise on spike-timing precision was reduced across a broad range of noise levels, loosing effectiveness only when noise levels increased enough to cause excitable neurons to spike multiple times per theta cycle and lethargic neurons to skip cycles. Again, this range highlights the system's limitations, only compensating for variability within a bounded region.

Counterintuitively, these results suggest that mismatch among neurons may improve STDP's ability to compensate for input noise. Mismatch provides the seed that influences the (average) time at which each neuron spikes, determining which connections are potentiated. In the absence of mismatch, the signal to potentiate (or depress) synapses would be much weaker, on average zero. I expect fewer potentiated connections would be made (and would just as easily depress after they were made). The smaller number of potentiated synapses would likely render such a system less effective at compensating for temporal noise. Further, such a system would form potentiated synapses much more slowly and therefore, any compensation for noise would take more time to develop than in a system that expresses (some) mismatch among its neurons.

Associative storage and recall naturally emerge in the PEP network when synapses between neurons coactivated by a pattern are potentiated. These synapses allow neurons to recruit their peers when a subset of the pattern is presented, thereby completing the pattern. However, this form of pattern storage and completion differs from Hopfield's attractor model [62]. Rather than forming symmetric, recurrent neuronal circuits, my recurrent network forms asymmetric circuits in which neurons make connections primarily to less excitable neurons in the pattern. In both the Poisson-like and regular cases, only about six percent of potentiated connections were reciprocated, as expected by chance.

I propose PEP as a candidate neural mechanism for information coding and storage in the hippocampal system. Observations from the CA1 region of the hippocampus suggest that basal dendrites (which primarily receive excitation from recurrent connections) support submillisecond timing precision, consistent with PEP [4]. I have shown, in a silicon model, PEP's ability to exploit such fast recurrent connections to sharpen timing precision as well as to associatively store and recall patterns.

Chapter 7

Conclusions

In this dissertation, I presented the STDP Chip, which realizes a neuromorphic model based on the CA3 region of the hippocampus. In the model, inhibitory interneurons synchronize, using mutual inhibition to generate the gamma rhythm. Further, excitatory neurons use STDP at recurrent synapses to store patterns, which can later be recalled with only a partial pattern. These same STDP synapses simultaneously compensate for variance among neurons in a pattern, resulting in precise timing (about 10ms).

I conclude this dissertation with a discussion of what I have learned about designing neuromorphic systems. First, in Section 7.1, I discuss the appropriate level of abstraction for neuromorphic implementations, emphasizing its critical role in producing functional systems. Next, in Section 7.2, I highlight the negative and positive roles mismatch plays in silicon and biological systems. Then, in Section 7.3, I present avenues of future work, which include extending and improving my neuromorphic system. In Section 7.4, I present insights into biology based on my neuromorphic model. Finally, in Section 7.5, I conclude that neuromorphic systems can be used to study neural processing.

7.1 Appropriate Level of Abstraction

In hindsight, the critical factor in building a functional system was not strict adherence to biology, but instead it was designing elements with the *appropriate level of abstraction* for the task at hand [15]. For neuromorphic engineers, the appropriate level of abstraction is the one at which I include the minimal complement of neuron, synapse, and ion channel types for the desired system as instructed by neurobiology, far beyond bioinspiration but short of complete mimicry. When I remove elements form this minimal complement, I cripple the systems, rendering it unable to reproduce the critical functions of its biological counterpart. When I add elements to the minimal complement, I also cripple the system, taking away precious silicon area from critical elements and increasing the sources of variability. In the balanced regime, neuromorphic systems can reproduce the behavior of neurobiology while evading the Valley of Death (see Chapter 3), in which systems fail due to excessive mismatch among elements.

Although one should not simplify systems just to facilitate analysis, employing the appropriate level of abstraction can enable us to tease apart the mechanisms responsible for a systems behavior, as in the inhibitory interneuron network (see Chapter 4). In that system, the key to realizing synchrony by mutual inhibition was designing neurons with the appropriate level of abstraction.

7.1.1 Silicon Gamma Synchrony

I constructed a network that generated synchrony among interneurons using delayed shunting inhibition. I used simple conductance-based neurons and synapses with a rise-time (in the form of a pulse filtered by an exponential), which provided effective delay. This was

the appropriate level of abstraction for this system because rise-time and shunting inhibition are both critical elements of synchrony by inhibition; excluding either of them would eliminate synchrony. Adding additional properties such as other types of ionic channels would only reduce the number of neurons and increase their mismatch, resulting in increased suppression. Further, my implementation's simplicity allowed me to perform a simple analysis to determine that rise-time behaved as an effective delay and to determine quantitatively how the inhibitory decay constant would modulate the network frequency, which the silicon interneuron network confirmed.

7.1.2 Silicon Synaptic Plasticity

STDP was the appropriate level of abstraction for synaptic plasticity in my system. STDP lacks the complex biophysical mechanics of synaptic LTP and LTD in the hippocampus; instead it is a simplification of empirical observations. Although I have previously implemented biophysical plasticity in the form of MVDP (See Chapter 5), I selected STDP for this system. It proved more resistant to device variability, always distinguishing correctly between pre-before-post and post-before-pre, ensuring the sign of synaptic weight change was correct (Chapter 5).

7.2 Mismatch Matters

Mismatch among elements of the same type is expressed by all physical systems[1], biological and silicon; intuitively, it degrades their system performance. Many neural systems suffer from this variability, which renders some neurons excitable and others lethargic. Counterintuitively, in some systems mismatch enhances performance, analogous to the role of noise in stochastic resonance, in which the information encoded about a weak input signal increases as external noise in added (up to a point), despite decreasing the signal-to-noise ratio [11].

7.2.1 Bane of Mismatch

Mismatch is the bane of neuromorphic systems as they tend to employ many small devices and small currents, which increases the mismatch among transistors and therefore circuits. Many implementations report degradation at the hand of mismatch, including retinas [158], cochleae [150], and associative memories [5], among others.

Neuromorphic systems have employed many techniques to manage mismatch, used in biology. Some systems have tried to minimize the affects of mismatch by learning transistor threshold variations and compensating for it with floating gates, programming mismatch equal and opposite to the intrinsic fabrication mismatch [43]. In a similar manner, I compensated for interneuron excitability mismatch by changing the probability of release at their (PC driven) excitatory synapses (Chapter 4). These two methods are analogous to

[1]Many silicon systems reduce mismatch by carefully engineering each transistor, which increases cost (silicon area). Systems also reduce mismatch by using outputs of transistors in an abstract manner as ones and zeros, instead of continuous values, which increases the energy required to perform computation, trading mismatch for higher power consumption [102].

homeostasis of excitability in neurobiology [98]. Other strategies require less drastic and costly measures, such as pooling multiple outputs [150], a common biological tactic. Other systems employ robust algorithms that are inherently insensitive to mismatch, such as axon guidance [138] and synchrony by inhibition (Chapter 4).

7.2.2 Boon of Mismatch

Neurobiology has evolved the ability to compensate for variability among neurons, limited by mismatch in the compensation mechanism [123]. Yet, measurements from neurons are notoriously variable. One explanation for this mismatch among neurons is that biology capitalizes on it, using it to perform computation.

Mismatch has the potential to be a boon for neuromorphic systems, enhancing computation. Mismatch among simulated neurons and among silicon neurons has successfully been used to generate orientation maps [142, 106, 107]. In this thesis, mismatch is a seed for learning (Chapter 6). In compensating for intrinsic (fixed) mismatch, the network also becomes robust against external (temporal) noise, demonstrating one way biology could combat variance among its elements.

7.3 Future Work

My silicon CA3 suggests several avenues of future work, building upon the current system and creating future systems.

7.3.1 Sequence Memory

In this dissertation, I presented a neuromorphic system that performs the role of the CA3 region in my model of sequence memory (Chapter 2), storing and recalling patterns. These patterns were active with precise timing such that multiple patterns could be multiplexed within one theta cycle, providing an ideal substrate to link a sequence together. The next step is to realize a dentate gyrus network to complete the sequence memory network, realizing the sequence memory model.

As a first implementation, I plan to use only one excitatory neuron type for the DG, mirroring the structure of the CA3, although the DG includes two types, granule neurons and mossy neurons. With the same structure as the CA3, the DG will be able to store and recall patterns. I will connect the DG and the CA3 as suggested by my model: The DG will drive the CA3, such that the CA3 will repeat its activity patterns but with some delay due to out-of-phase gamma rhythms. Then the CA3 will activate synapses in the DG just before the next pattern in a sequence is activated, providing an opportunity to link the pattern in the CA3 with the subsequent pattern in the DG, storing the sequence.

This two-chip network will give us an opportunity to study sequence memory in a physical system in the presence of noise and mismatch. I hope to determine the critical aspects of implementing sequence memory. Further, this system may give us insight into the complex DG architecture's reciprocally connected populations. This system promises to be the first neuromorphic model of sequence learning, but it also highlights potential improvements that could be made in future implementations.

7.3.2 Neural Scale

In future implementations of my system, the chips would benefit from increased the numbers of neurons and synapses. One of the largest strengths of neuromorphic engineering is the ability to build real-time model systems that approach the scale (often within a few orders of magnitude) of the neurobiological systems that they model as in [158, 155].

The current chip, with only 1024 pyramidal neurons each with 21 plastic synapses, is limited to small-scale exploration. I wish to explore many properties of the system, including properties that depend on scale such as network capacity, both in terms of pattern storage and sequence storage. With the current implementation, it is unlikely each neuron could be active in more than a handful of patterns.

In recent years, my colleagues and I have developed a programmable multichip architecture that can link together many chips to create very large scale systems [108, 88]. The largest system built to date consisted of five chips and a total of over 35,000 spiking neurons [27]. My colleagues and I will apply this multichip architecture and more advanced silicon technology to increase the size of our silicon hippocampus drastically, both in terms of number of neurons and number of plastic synapses per neuron.

As engineers better understand their simple systems, it makes sense to include neuron properties that were previously eliminated to reach a deeper understanding of the biological system, bearing in mind that one needs to include mechanisms that ensure robust operation despite the added variability, such as homeostasis. In the hippocampus, one neuron property is theorized to enhance the computational power of the network, active dendrites.

7.3.3 Silicon Dendrites

With the fabrication of this chip, I took a step back from a previous design in which neurons included active dendritic compartments. I sought to simplify the design and avoid the difficulties associated with multicompartment silicon neurons. However, future versions would benefit from including active dendrites. They add significant computational processing to neurons, effectively endowing each neuron with a additional layer of processing [121].

Although the technical aspects of including dendrites are daunting, requiring more advanced learning rules than STDP and various silicon ionic channel populations [5], they are essential to decipher the processing of the brain.

7.4 Insights into Neurobiology

A large part of neuromorphic engineering is reproducing processing found in neurobiology, implementing the insights of neuroscientists. However, neuromorphic engineers also have the opportunity to contribute to knowledge about the brain. Several results in this dissertation suggest ways that biology functions.

My study of gamma synchrony yielded insight into how biology may generate synchrony. The simple analysis of the role of synaptic rise-time (identical to delay) revealed the subsidiary role of the inhibitory decay constant: It modulated the network's synchronous period, which I verified with the silicon interneuron network. This suggests biology evolved synchrony in the hippocampus (and other parts of the brain) that would be robust against the inhibitory decay constant, which varies between neurons in different parts of the hippocampus [10]. Further, the network synchronizes despite suppression of a significant fraction of

the population.

The silicon CA3 demonstrates the power of spike timing-dependent plasticity. In silicon, my STDP circuits (all 21,504 of them) always express the correct dependence on timing and therefore never change synaptic weight with the wrong sign. Although this seems to be more of a technical issue than providing insight into to biology, it suggests that in physical systems timing is a robust signal.

The most provocative insight generated by the silicon CA3 is that biology may employ plasticity enhanced phase-coding, which uses recurrent STDP to enhance spike timing of coactive neurons in a pattern. If the CA3 region is an attractor network where patterns are stored, PEP naturally compensates for variability (and noise) while patterns are stored, making other mechanisms (such as homeostasis) superfluous. PEP may underlie timing precision in other plastic recurrent networks as well, such as the premotor nucleus RA, which is critical in songbird sequence learning [42].

7.5 Conclusion

The ability of the silicon CA3 to synchronize using inhibition as well as to store and to recall patterns, while simultaneously compensating for mismatch shows the utility of neuromorphic systems for exploring neural processing. The system highlights that neuromorphic models are uniquely able to study the brain in their own right; because they are based on similar physics, they encounter similar limitations. Neuromorphic models of brain regions inform us on biological tradeoffs: whether it is beneficial to pay the price to compensate for mismatch, to manage it with robust algorithms, or to exploit it for computation. With this in mind, to unlock the mysteries of neural computation, engineers must push deeper

into the brain, building ever larger and more complex systems, of ever increasing the detail, increasing the appropriate level of abstraction as the previous one is understood ensuring along the way that conclusions are tested with biological studies.

Appendix A

Phase Response

In this appendix, I analytically solve for the excitatory and inhibitory phase-response curves of an integrator- and a conductance-type neuron.

A.1 Instantaneous Inhibition

Inhibition affects the two neuron types differently. Inhibition removes a fixed charge from the integrator type, whereas it removes a variable amount from the conductance type, pulling its potential to a fixed fraction of its value before inhibition.

A.1.1 Integrator-Type Neuron

Consider an integrate-and-fire neuron that receives instantaneous inhibition at time t_i described by:

$$\mathrm{C}\dot{V} = \mathrm{I_{IN}} - \mathrm{I_{LEAK}} - \mathrm{Q}\delta(t - t_i) \tag{A.1}$$

where V is the neuron potential, $\mathrm{I_{IN}}$ and $\mathrm{I_{LEAK}}$ are the input and leak currents, respectively ($\mathrm{I_{IN}} > \mathrm{I_{LEAK}}$), and Q is the charge that inhibition removes from the neuron's capacitance, C.

When the neuron potential reaches threshold, $\mathrm{V_{TH}}$, its spikes. Without inhibition, The neuron spikes with period:

$$\mathrm{T}_0 = \frac{\mathrm{C\,V_{TH}}}{\mathrm{I_{IN}} - \mathrm{I_{LEAK}}} \tag{A.2}$$

When inhibition arrives between $t = 0$ and $t = \mathrm{T}_0$, the inhibition delays the neuron from reaching threshold and spiking. I solve for the effect of inhibition by finding the membrane trajectory before and after the inhibitory event. Before inhibition the neuron potential follows a linear trajectory, given by:

$$V(t) = \frac{\mathrm{I_{IN}} - \mathrm{I_{LEAK}}}{\mathrm{C}}\, t, \quad t < t_i \tag{A.3}$$

After inhibition the neuron potential continues on a linear trajectory, but shifted by the inhibition:

$$V(t) = \frac{\mathrm{I_{IN}} - \mathrm{I_{LEAK}}}{\mathrm{C}}\, t - \frac{\mathrm{Q}}{\mathrm{C}}, \quad t \geq t_i \tag{A.4}$$

I set Equation A.4 equal to $\mathrm{V_{TH}}$ and substitute t for the period, T, which yeilds:

$$\begin{aligned} T &= \frac{\mathrm{C\,V_{TH}}}{\mathrm{I_{IN}} - \mathrm{I_{LEAK}}} + \frac{\mathrm{Q}}{\mathrm{I_{IN}} - \mathrm{I_{LEAK}}} \\ &= \mathrm{T_0} + \frac{\mathrm{Q}}{\mathrm{I_{IN}} - \mathrm{I_{LEAK}}} \end{aligned} \tag{A.5}$$

The spike delay, or change in period, $\Delta T = T - \mathrm{T_0}$, is constant:

$$\Delta T = \frac{\mathrm{Q}}{\mathrm{I_{IN}} - \mathrm{I_{LEAK}}} \tag{A.6}$$

Thus, the integrator-type neuron is insensitive to the phase at which an inhibitory event arrives. It cannot distinguish the time of occurrence, only whether one arrived at all (Figure 3.2 *Top left*).

A.1.2 Conductance-Type Neuron

Consider an conductance-based neuron that receives instantaneous inhibition at time t_i described by:

$$\mathrm{C}\dot{V} = -V\mathrm{G}_{\mathrm{LEAK}} + \mathrm{I}_{\mathrm{IN}} - \mathrm{C}V\mathrm{f}\delta(t - t_i) \tag{A.7}$$

where I_{IN} is the input current, $\mathrm{G}_{\mathrm{LEAK}}$ is the leak conductance (to zero potential), and $\mathrm{Cf}\delta(t - t_i)$ is a conductance (to zero potential). f ($0 < \mathrm{f} < 1$) is the fraction of charge that inhibition removes from the neuron's capacitance, C. Since $\delta(t - t_i)$ has an area of one, we can integrate it to determine the charge that it removes from the neuron's membrane, which is equal to $\mathrm{C}V\mathrm{f}$, reducing the membrane potential from V to V(1-f).

As with the integrator type, when the neuron potential reaches threshold, V_{TH}, its spikes. Without inhibition, The neuron spikes with period T_0, given by:

$$\mathrm{T}_0 = \tau \log\left(\frac{1}{1 - \frac{\mathrm{V}_{\mathrm{TH}}}{\mathrm{V}_{\mathrm{IN}}}}\right) \tag{A.8}$$

where $\mathrm{V}_{\mathrm{IN}} = \frac{\mathrm{I}_{\mathrm{IN}}}{\mathrm{G}_{\mathrm{LEAK}}}$ and $\tau = \frac{\mathrm{C}}{\mathrm{G}_{\mathrm{LEAK}}}$. When inhibition arrives between $t = 0$ and $t = \mathrm{T}_0$, the inhibition delays the neuron from reaching threshold and spiking. As before, I solve for the effect of inhibition by finding the membrane trajectory before and after the inhibitory event. Before inhibition the neuron potential approaches V_{IN} with a decaying exponential trajectory, given by:

$$V(t) = \mathrm{V}_{\mathrm{IN}}(1 - e^{-t/\tau}), \quad t < t_i \tag{A.9}$$

After the shunting inhibition is applied at $t = t_i$ the new trajectory is:

$$V(t) = \mathrm{V_{IN}}(1-\mathrm{f})(1-e^{-t_i/\tau})+\mathrm{V_{IN}}[1-(1-\mathrm{f})(1-e^{-t_i/\tau})](1-e^{-(t-t_i)/\tau}), \quad t \geq t_i \quad \text{(A.10)}$$

I set Equation A.10 equal to $\mathrm{V_{TH}}$ and substitute t for the period, T, which yields:

$$\begin{aligned} T &= t_i + \tau \log\left(\frac{1-(1-\mathrm{f})(1-e^{-t_i/\tau})}{1-\frac{\mathrm{V_{TH}}}{\mathrm{V_{IN}}}}\right) \qquad \text{(A.11)} \\ &= \tau\log(e^{t_i/\tau}) + \tau\log\left(\frac{1}{1-\frac{\mathrm{V_{TH}}}{\mathrm{V_{IN}}}}\right) + \tau\log[1-(1-\mathrm{f})(1-e^{-t_i/\tau})] \\ &= \mathrm{T_0} + \tau\log\{e^{t_i/\tau}[1-(1-\mathrm{f})(1-e^{-t_i/\tau})]\} \\ \Delta T &= \tau\log[\mathrm{f}\,e^{t_i/\tau}+1-\mathrm{f}] \end{aligned}$$

This equation shows that the spike delay or change in period, ΔT, depends on the phase at which inhibition arrives (Figure 3.2 *Top right*).

A.2 Instantaneous Excitation

I perform analogous calculations for integrator-type and conductance-type neuron responses to excitation.

A.2.1 Integrator-Type Neuron

Consider an integrate-and-fire neuron that receives instantaneous excitation at time t_i described by:

$$\mathrm{C}\dot{V} = \mathrm{I_{IN}} - \mathrm{I_{LEAK}} + \mathrm{Q}\delta(t - t_i) \tag{A.12}$$

where all parameters are the same as with inhibition and Q is the charge that excitation adds to the neuron's capacitance.

When the neuron potential reaches threshold, $\mathrm{V_{TH}}$, it spikes. Without excitation, The neuron spikes with period T_0 given by Equation A.2. When excitation arrives between $t = 0$ and $t = \mathrm{T}_0$, it pushes the neuron to reach threshold and spike earlier. As before, I solve for the effect of excitation by finding the membrane trajectory before and after the excitatory event. Before excitation, the neuron potential follows a linear trajectory, given by Equation A.3. After excitation, the neuron potential continues on a linear trajectory, but shifted by the excitation:

$$V(t) = \frac{\mathrm{I_{IN}} - \mathrm{I_{LEAK}}}{\mathrm{C}}\, t + \frac{\mathrm{Q}}{\mathrm{C}}, \quad t \geq t_i \tag{A.13}$$

I set Equation A.4 equal to $\mathrm{V_{TH}}$ and substitute t for the period, T, which yields:

$$
\begin{aligned}
T &= \frac{\mathrm{C\,V_{TH}}}{\mathrm{I_{IN} - I_{LEAK}}} - \frac{\mathrm{Q}}{\mathrm{I_{IN} - I_{LEAK}}} \\
&= \mathrm{T_0} - \frac{\mathrm{Q}}{\mathrm{I_{IN} - I_{LEAK}}}
\end{aligned}
\tag{A.14}
$$

However, because excitation cannot push the neuron to spike before t_i, the period becomes:

$$
T = \max(\mathrm{T_0} - \frac{\mathrm{Q}}{\mathrm{I_{IN} - I_{LEAK}}}, t_i)
\tag{A.15}
$$

The spike delay, or change in period, $\Delta T = T - \mathrm{T_0}$, is constant early in the cycle, independent of t_i, but causes the neuron to spike immediately at t_i, later in the cycle:

$$
\Delta T = \max(-\frac{\mathrm{Q}}{\mathrm{I_{IN} - I_{LEAK}}}, t_i - \mathrm{T_0})
\tag{A.16}
$$

Thus, the integrator-type neuron is insensitive to the phase at which an excitatory event arrives, unless it is close to threshold (Figure 3.2 *Bottom left*).

A.2.2 Conductance-Type Neuron

Consider an conductance-based neuron that receives instantaneous excitation at time t_i described by:

$$\mathrm{C}\dot{V} = -V\mathrm{G}_{\mathrm{LEAK}} + \mathrm{V}_{\mathrm{IN}}\mathrm{G}_{\mathrm{LEAK}} + \mathrm{Q}\delta(t - t_i) \tag{A.17}$$

where all parameters as above and Q is the charge that excitation adds to the neuron's capacitance, increasing the neuron potential by ΔV. Although the excitation is not conductance based, the conductance-based leak alters the neuron's response compared to the integrator type.

As with the integrator type, when the neuron potential reaches threshold, V_{TH}, its spikes. Without excitation, The neuron spikes with period T_0 given by Equation A.8. When excitation arrives between $t = 0$ and $t = \mathrm{T}_0$, the excitation pushes the neuron to reach threshold and spike sooner. As before, I solve for the effect of excitation by finding the membrane trajectory before and after the excitatory event. Before excitation, the neuron potential approaches V_{IN} with a decaying exponential trajectory given by Equation A.9. After excitation, the neuron potential continues on the trajectory trajectory:

$$V(t) = \mathrm{V}_{\mathrm{IN}} - [\mathrm{V}_{\mathrm{IN}} - \mathrm{V}_{\mathrm{IN}}(1 - e^{-t_i/\tau}) - \Delta\mathrm{V}]e^{-(t-t_i)/\tau} \tag{A.18}$$

I set Equation A.18 equal to V_{TH} and substitute t for the period, T, which yields:

$$T = \max(\mathrm{T}_0 - \tau \log\left(\frac{1}{1 - \frac{\Delta\mathrm{V}}{\mathrm{V}_{\mathrm{IN}}}e^{t_i/\tau}}\right), t_i) \tag{A.19}$$

The spike delay, or change in period, ΔT, depends on the phase at which excitation arrives

(Figure 3.2 *Bottom right*), with

$$\Delta T = \max(-\tau \log\left(\frac{1}{1 - \frac{\Delta \mathrm{V}}{\mathrm{V_{IN}}} e^{t_i/\tau}}\right), t_i - \mathrm{T}_0) \tag{A.20}$$

The conductance-type neuron is sensitive to the phase at which even nonconductance-based excitatory or conductance-based inhibitory input arrives.

Appendix B

Low-Pass Filter

In this appendix, I derive the governing equation of the basic element of my neuron and synapse circuits, the current-mode log-domain low-pass filter (LPF) (Figure B.1). I analyze an NMOS LPF for simplicity (all potentials referenced to ground), but all of my circuits are implemented with PMOS transistors, because it is difficult to connect NMOS bulks to potentials different from the ground (substrate).

The equations I derive aid us in describing circuit behavior and estimating variability (mismatch) among repeated circuits. Here, I focus on the mismatch of two properties of the LPF: the equilibrium (steady-state) current and the decay constant. Assuming linearity (not necessarily true), these two properties are sufficient to describe the behavior of the LPF completely. In addition to linearity, the analysis assumes all transistors are operating in saturation ($V_{DS} > 0.1V$) in the subthreshold ($V_{GS} < V_{THR} \approx 0.4V$ in 0.25μm CMOS) regime.

The LPF consists of four transistors and one capacitor. Its input is a current, I_{IN}, and

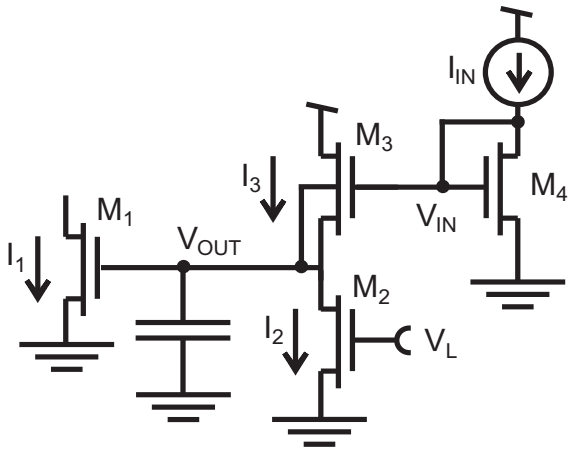

Figure B.1: Low-Pass Filter

The current-mode low-pass filter implements a first-order ODE that models the dynamics of a resistor-capacitor circuit.

its output is also a current, I_1 (through M_1). With a constant I_{IN}, the steady state currents through each NMOS transistor are given by:

$$I_1 = I_{01}\, e^{\kappa_1 V_{OUT}/U_t} \tag{B.1}$$

$$I_2 = I_{02}\, e^{\kappa_2 V_L/U_t} \tag{B.2}$$

$$I_3 = I_{03}\, e^{\kappa_3 (V_{IN}-V_{OUT})/U_t} \tag{B.3}$$

$$I_4 = I_{04}\, e^{\kappa_4 V_{IN}/U_t} \tag{B.4}$$

where I_i is the drain-source current through M_i and I_{0i} is its drain-source current at $V_{GS} = 0$; κ_i is the subthreshold slope coefficient for M_i and U_t is the thermal voltage, 25.6mV at room temperature. These equations assume that M_3's bulk is connected to V_{OUT}.

To find I_1, I first solve for V_{IN} by using $I_{IN} = I_4$.

$$V_{IN} = \frac{U_t}{\kappa_4} \log\left(\frac{I_{IN}}{I_{04}}\right) \tag{B.5}$$

Now, I substitute this result into Equation B.3 to find I_3.

$$I_3 = I_{03}\left(\frac{I_{IN}}{I_{04}}\right)^{\kappa_3/\kappa_4} e^{-\kappa_3 V_{OUT}/U_t} \tag{B.6}$$

Setting $I_2 = I_3$, I can solve for V_{OUT}'s steady-state value.

$$V_{OUT} = \frac{U_t}{\kappa_3} \log\left[\left(\frac{I_{IN}}{I_{04}}\right)^{\kappa_3/\kappa_4} \frac{I_{03}}{I_{02}} e^{\kappa_2 V_L/U_t}\right] \tag{B.7}$$

Using this result, I find the steady-state output current:

$$I_1 = I_{01}\left(\frac{I_{IN}}{I_{04}}\right)^{\kappa_1/\kappa_4}\left(\frac{I_{03}}{I_{02}} e^{\kappa_2 V_L/U_t}\right)^{\kappa_1/\kappa_3} \tag{B.8}$$

Notice that I_1 is nonlinear, if $\kappa_1 \neq \kappa_4$. Now I simplify Equation B.8 by assuming $\kappa_{1-4} = \kappa$[1], which yields:

[1] κ varies with source-bulk voltage and for different transistor geometries, especially for thin-gate-oxide devices in 0.25μm CMOS and smaller technologies; however, threshold voltage variations are usually dominant.

$$I_1 = I_{IN} \frac{I_{01} I_{03}}{I_{04} I_{02}} e^{\kappa V_L / U_t} \tag{B.9}$$

Now, I substitute I_{0i} with $I_0 e^{\Delta V_{THi}}$.

$$I_1 = I_{IN}\ e^{(\kappa V_L + \Delta V_{TH1} + \Delta V_{TH3} - \Delta V_{TH2} - \Delta V_{TH4})/U_t} \tag{B.10}$$

This equations shows that the mismatch among the LPF steady-state output currents depends on the variances of the four transistors. It is important to note that this mismatch would be much more complex if I considered kappa variation also; each ΔV_{THi} term would be multiplied by a ratio of κ terms. The variation of the sum of ΔV_{THi} is a random variable, whose variance equals the sum of the individual variances. If all transistors were the same size (and had the same variance), the circuit's variance would be four times that of each individual transistor.

Now that I have the LPF's steady-state output current, I calculate its decay constant, which is given by:

$$\begin{aligned} \tau &= \frac{\mathrm{C_L\,U_t}}{\kappa_1 \mathrm{I_2}} \\ &= \frac{\mathrm{C_L\,U_t}}{\kappa_1 \mathrm{I_{02}}\, e^{\kappa_2 \mathrm{V_L}/\mathrm{U_t}}} \\ &= \frac{\mathrm{C_L\,U_t}}{\kappa_1 \mathrm{I_0}\, e^{\Delta \mathrm{V_{TH2}} + \kappa_2 \mathrm{V_L}/\mathrm{U_t}}} \end{aligned} \tag{B.11}$$

where $\mathrm{C_L}$ is the capacitance on node $\mathrm{V_{OUT}}$. The mismatch among decay constants depends on the variations of κ_{1-2} and $\Delta \mathrm{V_{TH2}}$. Simplifying Equation B.11 by assuming $\kappa_{1-2} = \kappa$, yeilds:

$$\tau = \frac{\mathrm{C_L\,U_t}}{\kappa \mathrm{I_0}\, e^{\Delta \mathrm{V_{TH2}} + \kappa \mathrm{V_L}/\mathrm{U_t}}} \tag{B.12}$$

This equations shows that the mismatch among the LPF decay constants depends (primarily) on the variance of the one transistor, $\mathrm{M_2}$. Thus, is pays to make $\mathrm{M_2}$ a large transistor: It contributes to both the steady-state output current variability and the decay-constant variability. Also, increasing its length reduces its leakage current, which determines the LPF's maximum possible decay constant.

Appendix C

SRAM Scanner

In this appendix, I describe the scanner circuits that read and write the state of the STDP circuits' SRAM cells (Chapter 5) as well as select an analog current for observation.

C.1 Architecture

The scanner consists of two linear shift-registers: V Scanner (224 segments) for vertical and H Scanner (112 segments) for horizontal (Figure C.1). The user selects a subpixel using two digital-input pads, VCLOCK for vertical and HCLOCK for horizontal: Each time a clock is pulsed, the shift-register advances one row or column (wrapping around to the beginning if it reaches the end). When V Scanner selects a row by raising SCAN (also lowering ∼ASCAN) every SRAM in the row drives either its ∼BIT or BIT line low, signaling its state. H Scanner selects one column's ∼BIT and BIT lines, which drive a digital-output pad, SRAM_{OUT}, that can be read by the user.

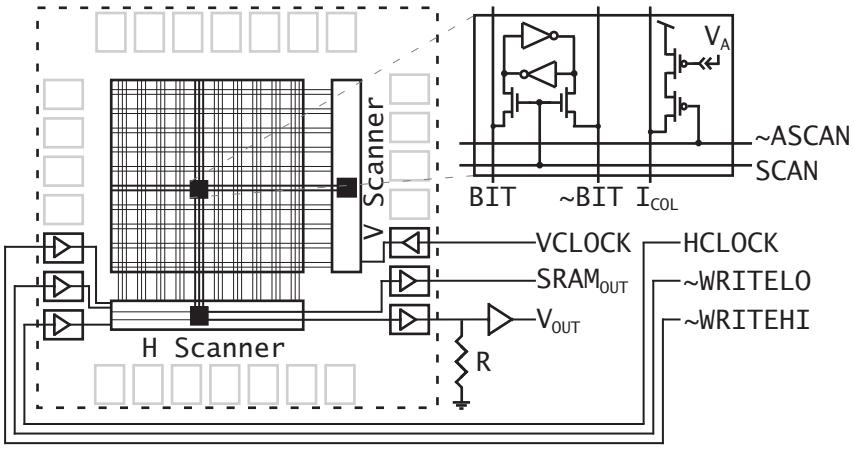

Figure C.1: Scanner Architecture

The scanner consists of two linear shift-registers: V Scanner for vertical and H Scanner for horizontal. The user selects a subpixel using two digital-input pads, VCLOCK for vertical and HCLOCK for horizontal. The selected pixel's SRAM and current values are sent off chip. The user can write the selected pixels SRAM by driving ∼WRITEHI or ∼WRITELO low.

Two additional input pads, ∼WRITEHI and ∼WRITELO, allow the user to overwrite the state of an SRAM: If ∼WRITEHI or ∼WRITELO are pulled low a circuit is activated that removes the tonic high precharge and pulls the BIT or ∼BIT line of the currently selected SRAM low, flipping its state to potentiated or depressed, respectively, if necessary.

Analog currents are sent off chip at the same time as digital values. Only 32 of 224 rows include an ∼ASCAN line as there are only two rows of neurons per metapixel, whereas their are 16 rows of SRAM. When ∼ASCAN is low it allows V_A to drive its current as the column current, I_{COL}. All 112 columns include an I_{COL} line, whereas only 96 include ∼BIT or BIT lines, since their are 7 currents per metapixel row and only 6 SRAM cells. H Scanner selects one column current to drive a current out pad, which drives an off-chip resistor, R, generating (buffered) voltage V_{OUT}.

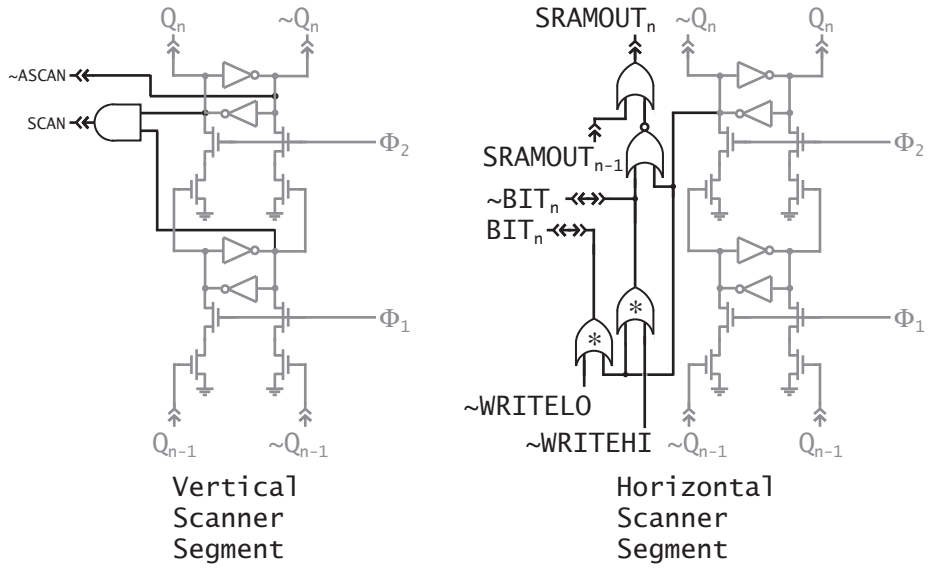

Figure C.2: Scanner Segments

The vertical (left) and horizontal (right) segments contain a shift-register (gray) and additional circuitry (black). The vertical segment contains circuitry to activate the SCAN line only if both shift-registers have the same state, which occurs when Φ_2 is high. The horizontal scanner contains circuitry to send the selected segment's SRAM state off chip and to overwrite it. * indicates a weak pull up.

C.2 Segments

The vertical and horizontal segments are based on a shift register, consisting of a pair of cross-coupled inverters (Figure C.2). Each pair receives two nonoverlapping phases from a clock. Input clocks, VCLOCK and HCLOCK, each drive their own clock-generator circuit, which creates the nonoverlapping phases, Φ_1 and Φ_2. The first pair of inverters receives active high and active low input from the previous scanner segment (or from the last one, if it is the first), Q_{n-1} and $\sim Q_{n-1}$. When Φ_1 goes high, the first inverter pair copies Q_{n-1}'s state, and latches it when Φ_1 goes low. When Φ_2 goes high, the second inverter pair copies the state of the first, completing the transfer of Q_{n-1}'s state to Q_n.

The vertical segment includes circuitry to prevent any SCAN row signal from being active. While SCAN is high, SRAM cells in the selected row are shorted to their ∼BIT or BIT lines, which act as large capacitors on the order of picofarads. Such large capacitance prevents the internally generated signals from changing the SRAMs' states. To solve this problem, I AND the states of both inverter pairs. Therefore, the SCAN line is only high when Φ_2 is high. The user need only keep VCLOCK high to ensure Φ_2 is low and no row is selected, except when scanning out the SRAM array.

The horizontal segment includes circuitry to send the selected SRAM's state off chip. To send the selected SRAM's state off chip, I NOR every SRAM's $\sim BIT_n$ line with $\sim Q_n$. If both are low, the NOR gate is high, signaling SRAM n is selected and its state is low. I OR the output of all NOR gates together: A gate in each pixel ORs its own NOR-gate output with that of the previous segment. The signal sent off chip, SRAMOUT, is the inverted value of the selected SRAM's state.

The horizontal segment also includes circuitry to overwrite the selected SRAM. The ∼WRITEHI and ∼WRITELO signal the circuit to write the selected SRAM segment high and low, respectively. I OR each one with $\sim Q_n$. If the write line and $\sim Q_n$ are low, the OR gate's output is low. The OR gate's NMOS output transistors are strong and pulling the bit line low, which overcomes the weak PMOS transistors of the SRAM and overwrites its state, depending on which write line was low. On the other hand, if the write line or $\sim Q_n$ are high, the OR gate's output is high; however the OR gate's output PMOSs are very weak, too weak to overcome the strong NMOSs in the SRAM.

The OR's PMOSs do serve an important purpose: They drive the ∼BIT or BIT lines high when VCLOCK transitions to high, pulling SCAN low. If the any lines were left low, SCAN's transition to high would short the line capacitance to the SRAMs, which would

overcome the SRAMs' weak PMOSs, overwriting the bit. With the lines high, SCAN's transition still shorts the line capacitance to the SRAMs, but the SRAMs' NMOSs are strong, able to pull the (appropriate) bit lines low, preserving their states.

Bibliography

[1] R. Adams. Filtering the log domain. In *Proceedings of the 63rd Convention of the Audio Engineering Society*, New York, 1979.

[2] A. Andreou and K. Boahen. Translinear circuits in subthreshold MOS. *Analog Integrated Circuits and Signal Processing*, 9:141-66, 1996.

[3] G. Andrea, M. Savino, and A. Trotta. Windows and interpolation algorithms to improve electrical measurement accuracy. *Instrumentation and Measurement, IEEE Transactions on*, 38(4):856-63, 1989.

[4] G. Ariav, A. Polsky, and J. Schiller. Submillisecond precision of the input-output transformation function mediated by fast sodium dendritic spikes in basal dendrites of CA1 pyramidal neurons. *J Neurosci*, 23(21):7750-8, 2003.

[5] J. V. Arthur and K. Boahen. Recurrently connected silicon neurons with active dendrites for one-shot learning. In *Neural Networks, 2004, Proceedings, 2004 IEEE International Joint Conference on*, volume 3, 2004.

[6] J. V. Arthur and K. Boahen. Learning in silicon: timing is everything. In Y. Weiss, B. Scholkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, Vancouver, 2005. MIT Press.

[7] J. V. Arthur and K. Boahen. Silicon neurons that inhibit to synchronize. In *Circuits and Systems, 2006, ISCAS '06. Proceedings of the 2006 IEEE International Symposium on*, volume 8, Kos, 2006.

[8] J. V. Arthur and K. Boahen. Synchrony in Silicon: The Gamma Rhythm. *IEEE Trans Neural Netw*, submitted.

[9] C. Barnes. Effects of aging on the dynamics of information processing and synaptic weight changes in the mammalian hippocampus. *Prog Brain Res*, 8:89-104, 1990.

[10] M. Bartos, I. Vida, M. Frotscher, J. R. Geiger, and P. Jonas. Rapid signaling at inhibitory synapses in a dentate gyrus interneuron network. *J Neurosci*, 21(8):2687-98, 2001.

[11] R. Benzi, A. Sutera, and A. Vulpiani. The mechanism of stochastic resonance. *J Phys A*, 14:L453-7, 1981.

[12] G. Q. Bi and M. M. Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J Neurosci*, 18(24):10464-72, 1998.

[13] G. Q. Bi and H. X. Wang. Temporal asymmetry in spike timing-dependent synaptic plasticity. *Physiol Behav*, 77(4-5):551-5, 2002.

[14] T. V. Bliss and T. Lomo. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *J Physiol*, 232(2):331-56, 1973.

[15] K. Boahen. Retinomorphic Vision Systems. In *5th International Conference on Microelectronics for Neural Networks and Fuzzy Systems (MicroNeuro '96)*, 1996. IEEE Computer Society.

[16] K. Boahen. A burst-mode word-serial address-event link-I: transmitter design. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 51(7):1269-80, 2004.

[17] K. Boahen. A burst-mode word-serial address-event link-II: receiver design. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 51(7):1281-91, 2004.

[18] K. Boahen. A burst-mode word-serial address-event link-III: analysis and test results. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 51(7):1292-1300, 2004.

[19] A. Bofill-i-Petit and A. Murray. Synchrony detection by analogue VLSI neurons with bimodal STDP synapses. In S. Thrun, L. Saul, and B. Scholkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, Vancouver, 2003. MIT Press.

[20] A. Bragin, G. Jando, Z. Nadasdy, J. Hetke, K. Wise, and G. Buzsaki. Gamma (40-100 Hz) oscillation in the hippocampus of the behaving rat. *J Neurosci*, 15(1 Pt 1):47-60, 1995.

[21] N. Brunel and X. J. Wang. What determines the frequency of fast network oscillations with irregular neural discharges? I. Synaptic dynamics and excitation-inhibition balance. *J Neurophysiol*, 90(1):415-30, 2003.

[22] D. L. Buhl, K. D. Harris, S. G. Hormuzdi, H. Monyer, and G. Buzsaki. Selective impairment of hippocampal gamma oscillations in connexin-36 knock-out mouse in vivo. *J Neurosci*, 23(3):1013-8, 2003.

[23] G. Buzsaki. Theta oscillations in the hippocampus. *Neuron*, 33(3):325-40, 2002.

[24] K. Cameron, V. Boonsobhak, A. Murray, and D. Renshaw. Spike timing dependent plasticity (STDP) can ameliorate process variations in neuromorphic VLSI. *IEEE Trans Neural Netw*, 16(6):1626-37, 2005.

[25] E. Chicca, D. Badoni, V. Dante, M. D'Andreagiovanni, G. Salina, L. Carota, S. Fusi, and P. Del Giudice. A VLSI recurrent network of integrate-and-fire neurons connected by plastic synapses with long-term memory. *IEEE Trans Neural Netw*, 14(5):1297-1307, 2003.

[26] T.-D. Chiueh and R. Goodman. VLSI implementation of a high-capactiy neural network associative memory. In D. Touretzky, editors, *Advances in neural information processing systems*, volume 2, Denver, 1990. Morgan Kaufmann Publishers Inc.

[27] T. Choi, P. Merolla, J. Arthur, K. Boahen, and B. Shi. Neuromorphic implementation of orientation hypercolumns. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 52(6):1049-60, 2005.

[28] C. Chow. Phase-locking in weakly heterogeneous neuronal networks. *Physica D*, 118(3):343-70, 1998.

[29] N. S. Clayton and A. Dickinson. Episodic-like memory during cache recovery by scrub jays. *Nature*, 395(6699):272-4, 1998.

[30] D. C. Cooper, S. Chung, and N. Spruston. Output-mode transitions are controlled by prolonged inactivation of sodium channels in pyramidal neurons of subiculum. *PLoS Biol*, 3(6):e175, 2005.

[31] J. Csicsvari, B. Jamieson, K. D. Wise, and G. Buzsaki. Mechanisms of gamma oscillations in the hippocampus of the behaving rat. *Neuron*, 37(2):311-22, 2003.

[32] E. Culurciello, R. Etienne-Cummings, and K. Boahen. A biomorphic digital image sensor. *Solid-State Circuits, IEEE Journal of*, 38(2):281-94, 2003.

[33] G. S. Cymbalyuk, G. N. Patel, R. L. Calabrese, S. P. DeWeerth, and A. H. Cohen. Modeling alternation to synchrony with inhibitory coupling: a neuromorphic VLSI approach. *Neural Comput*, 12(10):2259-78, 2000.

[34] D. Debanne, N. C. Guerineau, B. H. Gahwiler, and S. M. Thompson. Physiology and pharmacology of unitary synaptic connections between pairs of cells in areas CA3 and CA1 of rat hippocampal slice cultures. *J Neurophysiol*, 73(3):1282-94, 1995.

[35] A. Destexhe, Z. Mainen, and T. Sejnowski. An efficient method for computing synaptic conductances based on a kinetic model of receptor binding. *Neural Computation*, 6(1):14-8, 1994.

[36] G. Dragoi and G. Buzsaki. Temporal encoding of place sequences by hippocampal cell assemblies. *Neuron*, 50(1):145-57, 2006.

[37] A. V. Egorov, B. N. Hamam, E. Fransen, M. E. Hasselmo, and A. A. Alonso. Graded persistent activity in entorhinal cortex neurons. *Nature*, 420(6912):173-8, 2002.

[38] H. Eichenbaum, P. Dudchenko, E. Wood, M. Shapiro, and H. Tanila. The hippocampus, memory, and place cells: is it spatial memory or a memory space? *Neuron*, 23(2):209-26, 1999.

[39] A. D. Ekstrom, M. J. Kahana, J. B. Caplan, T. A. Fields, E. A. Isham, E. L. Newman, and I. Fried. Cellular networks underlying human spatial navigation. *Nature*, 425(6954):184-8, 2003.

[40] B. Ermentrout, M. Pascal, and B. Gutkin. The effects of spike frequency adaptation and negative feedback on the synchronization of neural oscillators. *Neural Comput*, 13(6):1285-310, 2001.

[41] G. B. Ermentrout and C. C. Chow. Modeling neural oscillations. *Physiol Behav*, 77(4-5):629-33, 2002.

[42] M. S. Fee, A. A. Kozhevnikov, and R. H. Hahnloser. Neural mechanisms of vocal sequence generation in the songbird. *Ann N Y Acad Sci*, 1016:153-70, 2004.

[43] M. Figueroa, S. Bridges, and C. Diorio. On-chip compensation of device-mismatch effects in analog VLSI neural networks. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17, Vancouver, 2004. MIT Press.

[44] N. J. Fortin, K. L. Agster, and H. B. Eichenbaum. Critical role of the hippocampus in memory for sequences of events. *Nat Neurosci*, 5(5):458-62, 2002.

[45] D. J. Foster and M. A. Wilson. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 440(7084):680-3, 2006.

[46] T. F. Freund and G. Buzsaki. Interneurons of the hippocampus. *Hippocampus*, 6(4):347-470, 1996.

[47] D. Frey. Log-domain filtering: an approach to current-mode filtering. *IEE Proceedings G. Circuits, Devices and Systems*, 140(6):406-16, 1993.

[48] S. Fusi, M. Annunziato, D. Badoni, A. Salamon, and D. J. Amit. Spike-driven synaptic plasticity: theory, simulation, VLSI implementation. *Neural Comput*, 12(10):2227-58, 2000.

[49] W. Gerstner. Population dynamics of spiking neurons: fast transients, asynchronous states, and locking. *Neural Comput*, 12(1):43-89, 2000.

[50] B. Gilbert. Translinear circuits: a proposed classification. *Electron Lett*, 11(1):14-6, 1975.

[51] J. M. Goldberg and P. B. Brown. Response of binaural neurons of dog superior olivary complex to dichotic tonal stimuli: some physiological mechanisms of sound localization. *J Neurophysiol*, 32(4):613-36, 1969.

[52] A. Gupta, Y. Wang, and H. Markram. Organizing principles for a diversity of GABAergic interneurons and synapses in the neocortex. *Science*, 287(5451):273-8, 2000.

[53] P. Hafliger and M. Mahowald. Spike based normalizing Hebbian learning in an analog VLSI artificial neuron. *Analog Integrated Circuits and Signal Processing*, 18(2-3):133-9, 1999.

[54] P. Hafliger and C. Rasche. Floating gate analog memory for parameter and variable storage in a learning silicon neuron. In *Circuits and Systems, 1999, ISCAS '99. Proceedings of the 1999 IEEE International Symposium on*, volume 2, Orlando, 1999.

[55] P. Hafliger and H. Riis. A multi-level static memory cell. In *Circuits and Systems, 2003, ISCAS '03. Proceedings of the 2003 IEEE International Symposium on*, volume 5, Bangkok, 2003.

[56] R. E. Hampson, J. D. Simeral, and S. A. Deadwyler. Distribution of spatial and nonspatial information in dorsal hippocampus. *Nature*, 402(6762):610-4, 1999.

[57] K. D. Harris, J. Csicsvari, H. Hirase, G. Dragoi, and G. Buzsaki. Organization of cell assemblies in the hippocampus. *Nature*, 424(6948):552-6, 2003.

[58] D. Hebb. *The organization of behavior*, Wiley, New York, 1949.

[59] D. A. Henze, L. Wittner, and G. Buzsaki. Single granule cells reliably discharge targets in the hippocampal CA3 network in vivo. *Nat Neurosci*, 5(8):790-5, 2002.

[60] W. Himmelbauer and A. Andreou. Log-domain circuits in subthreshold MOS. In *Circuits and Systems, 1997. Proceedings of the 40th Midwest Symposium on*, volume 1, Sacramento, 1997.

[61] G. R. Holt and C. Koch. Shunting inhibition does not have a divisive effect on firing rates. *Neural Comput*, 9(5):1001-13, 1997.

[62] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci U S A*, 79(8):2554-8, 1982.

[63] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc Natl Acad Sci U S A*, 81(10):3088-92, 1984.

[64] J. J. Hopfield. Pattern recognition computation using action potential timing for stimulus representation. *Nature*, 376(6535):33-6, 1995.

[65] F. Hoppensteadt and E. Izhikevich. Pattern recognition via synchronization in phase-locked loop neural networks. *IEEE Trans Neural Netw*, 11(3):734-8, 2000.

[66] R. Howard, D. Schwartz, J. Denker, R. Epworth, H. Graf, W. Hubbard, L. Jackel, B. Straughn, and D. Tennant. An associative memory based on an electronic neural network. *Electron Devices, IEEE Transactions on*, 34(7):1553-6, 1987.

[67] J. Huxter, N. Burgess, and J. O'Keefe. Independent rate and temporal coding in hippocampal pyramidal cells. *Nature*, 425(6960):828-32, 2003.

[68] K. Hynna and K. Boahen. Space-rate coding in an adaptive silicon neuron. *Neural Netw*, 14(6-7):645-56, 2001.

[69] K. Hynna and K. Boahen. A silicon implementation of the thalamic low threshold calcium current. In *Engineering in Medicine and Biology Society, 2003, Proceedings of the 25th Annual International Conference of the IEEE*, volume 3, 2003.

[70] K. Hynna and K. Boahen. Neuronal ion-channel dynamics in silicon. In *Circuits and Systems, 2006, ISCAS '06. Proceedings of the 2006 IEEE International Symposium on*, volume 8, Kos, 2006.

[71] G. Indiveri. Neuromorphic bistable VLSI synapses with spike-timing-dependent plasticity. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, Vancouver, 2003. MIT Press.

[72] O. Jensen, M. A. Idiart, and J. E. Lisman. Physiologically realistic formation of autoassociative memory in networks with theta/gamma oscillations: role of fast NMDA channels. *Learn Mem*, 3(2-3):243-56, 1996.

[73] O. Jensen and J. E. Lisman. Novel lists of 7 +/- 2 known items can be reliably stored in an oscillatory short-term memory network: interaction with long-term memory. *Learn Mem*, 3(2-3):257-63, 1996.

[74] O. Jensen and J. E. Lisman. Theta/gamma networks with slow NMDA channels learn sequences and encode episodic memory: role of NMDA channels in recall. *Learn Mem*, 3(2-3):264-78, 1996.

[75] O. Jensen and J. E. Lisman. Hippocampal CA3 region predicts memory sequences: accounting for the phase precession of place cells. *Learn Mem*, 3(2-3):279-87, 1996.

[76] O. Jensen and J. E. Lisman. An oscillatory short-term memory buffer model can account for data on the Sternberg task. *J Neurosci*, 18(24):10688-99, 1998.

[77] A. Johannet, L. Personnaz, G. Dreyfus, J.-D. Gascuel, and M. Weinfeld. Specification and implementation of a digital Hopfield-type associative memory with on-chip training. *IEEE Trans Neural Netw*, 3(4):529-39, 1992.

[78] D. Johnston and D. Amaral, "Hippocampus," in *The synaptic organization of the brain*, G. Shepherd, Ed. Oxford: Oxford University Press, 2004.

[79] E. Kart-Teke, M. A. De Souza Silva, J. P. Huston, and E. Dere. Wistar rats show episodic-like memory for unique experiences. *Neurobiol Learn Mem*, 85(2):173-82, 2006.

[80] T. Klausberger, P. J. Magill, L. F. Marton, J. D. B. Roberts, P. M. Cobden, G. Buzsaki, and P. Somogyi. Brain-state- and cell-type-specific firing of hippocampal interneurons in vivo. *Nature*, 421(6925):844-8, 2003.

[81] D. Kleinfeld. Sequential state generation by model neural networks. *Proc Natl Acad Sci U S A*, 83(24):9469-73, 1986.

[82] Y. Kuramoto. *Chemical cscillations, waves, and turbulence*, Springer-Verlag, Berlin; New York, 1984.

[83] M. Lawrence, T. Trappenberg, and A. Fine. Rapid learning and robust recall of long sequences in modular associator networks. *Neurocomputing*, 69:634-41, 2006.

[84] I. Lee, G. Rao, and J. J. Knierim. A double dissociation between hippocampal subfields: differential time course of CA3 and CA1 place cells for processing changed environments. *Neuron*, 42(5):803-15, 2004.

[85] W. B. Levy. A sequence predicting CA3 is a flexible associator that learns and uses context to solve hippocampal-like tasks. *Hippocampus*, 6(6):579-90, 1996.

[86] W. B. Levy and O. Steward. Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus. *Neuroscience*, 8(4):791-7, 1983.

[87] T. J. Lewis and J. Rinzel. Dynamics of spiking neurons connected by both inhibitory and electrical coupling. *J Comput Neurosci*, 14(3):283-309, 2003.

[88] J. Lin, P. Merolla, J. Arthur, and K. Boahen. Programmable connections in neuromorphic grids. In *Circuits and Systems, 2006. Proceedings of the 49th Midwest Symposium on*, San Juan, 2006.

[89] J. Lisman. A mechanism for the Hebb and the anti-Hebb processes underlying learning and memory. *Proc Natl Acad Sci U S A*, 86(23):9574-8, 1989.

[90] J. E. Lisman. Relating hippocampal circuitry to function: recall of memory sequences by reciprocal dentate-CA3 interactions. *Neuron*, 22(2):233-42, 1999.

[91] J. E. Lisman, L. M. Talamini, and A. Raffone. Recall of memory sequences by interaction of the dentate and CA3: a revised model of the phase precession. *Neural Netw*, 18(9):1191-201, 2005.

[92] S. C. Liu and R. Douglas. Temporal coding in a silicon network of integrate-and-fire neurons. *IEEE Trans Neural Netw*, 15(5):1305-14, 2004.

[93] Y. Maeda and M. Wakamura. Simultaneous perturbation learning rule for recurrent neural networks and its FPGA implementation. *IEEE Trans Neural Netw*, 16(6):1664-72, 2005.

[94] R. Maex and E. De Schutter. Resonant synchronization in heterogeneous networks of inhibitory neurons. *J Neurosci*, 23(33):10503-14, 2003.

[95] E. A. Maguire, D. G. Gadian, I. S. Johnsrude, C. D. Good, J. Ashburner, R. S. Frackowiak, and C. D. Frith. Navigation-related structural change in the hippocampi of taxi drivers. *Proc Natl Acad Sci U S A*, 97(8):4398-403, 2000.

[96] M. Mahowald and R. Douglas. A silicon neuron. *Nature*, 354(6354):515-8, 1991.

[97] R. C. Malenka and R. A. Nicoll. Long-term potentiation–a decade of progress? *Science*, 285(5435):1870-4, 1999.

[98] E. Marder and J. M. Goaillard. Variability, compensation and homeostasis in neuron and network function. *Nat Rev Neurosci*, 7(7):563-74, 2006.

[99] D. Marr. Simple memory: a theory for archicortex. *Philos Trans R Soc Lond B Biol Sci*, 262(841):23-81, 1971.

[100] W. S. McCulloch and W. H. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* , 5:115-133. 1943.

[101] C. Mead. *Analog VLSI and neural systems*, Addison-Wesley, Reading, MA, 1989.

[102] C. Mead. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10):1629-36, 1990.

[103] C. Mead and T. Delbruck. Scanners for visualizing activity of analog VLSI circuitry. *Analog Integrated Circuits and Signal Processing*, 1(2):93-106, 1991.

[104] M. R. Mehta, A. K. Lee, and M. A. Wilson. Role of experience and oscillations in transforming a rate code into a temporal code. *Nature*, 417(6890):741-6, 2002.

[105] E. D. Menschik and L. H. Finkel. Neuromodulatory control of hippocampal function: towards a model of Alzheimer's disease. *Artif Intell Med*, 13(1-2):99-121, 1998.

[106] P. Merolla and K. Boahen. A recurrent model of orientation maps with simple and complex cells. In S. Thrun, L. Saul, and B. Scholkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, Vancouver, 2003. MIT Press.

[107] P. Merolla and K. Boahen. Dynamic computation in a recurrent network of heterogeneous silicon neurons. In *Circuits and Systems, 2006, ISCAS '06. Proceedings of the 2006 IEEE International Symposium on*, volume 8, Kos, 2006.

[108] P. A. Merolla; J. V. Arthur, B. E. Shi, and K. A. Boahen. Expandable Networks for Neuromorphic Chips. *IEEE Transactions on Circuits and Systems I*, in press.

[109] I. Mihu, R. Brad, and M. Breazu. Specifications and FPGA implementation of a systolic Hopfield-type associative memory. In *IJCNN '01 International Joint Conference on Neural Networks*, volume 1, Washington, DC, 2001.

[110] R. Morris. Developments of a water-maze procedure for studying spatial learning in the rat. *J Neurosci Methods*, 11(1):47-60, 1984.

[111] R. G. Morris, E. I. Moser, G. Riedel, S. J. Martin, J. Sandin, M. Day, and C. O'Carroll. Elements of a neurobiological theory of the hippocampus: the role of activity-dependent synaptic plasticity in memory. *Philos Trans R Soc Lond B Biol Sci*, 358(1432):773-86, 2003.

[112] Z. Nadasdy, H. Hirase, A. Czurko, J. Csicsvari, and G. Buzsaki. Replay and time compression of recurring spike sequences in the hippocampus. *J Neurosci*, 19(21):9497-507, 1999.

[113] K. Nakazawa, M. C. Quirk, R. A. Chitwood, M. Watanabe, M. F. Yeckel, L. D. Sun, A. Kato, C. A. Carr, D. Johnston, M. A. Wilson, and S. Tonegawa. Requirement for hippocampal CA3 NMDA receptors in associative memory recall. *Science*, 297(5579):211-8, 2002.

[114] K. Nakazawa, L. D. Sun, M. C. Quirk, L. Rondi-Reig, M. A. Wilson, and S. Tonegawa. Hippocampal CA3 NMDA receptors are crucial for memory acquisition of one-time experience. *Neuron*, 38(2):305-15, 2003.

[115] J. O'Keefe and J. Dostrovsky. The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34:171-5, 1971.

[116] J. O'Keefe and L. Nadel. *The hippocampus as a cognitive map*, Clarendon Press, Oxford, 1978.

[117] J. O'Keefe and M. L. Recce. Phase relationship between hippocampal place units and the EEG theta rhythm. *Hippocampus*, 3(3):317-30, 1993.

[118] N. Otmakhov, J. H. Tao-Cheng, S. Carpenter, B. Asrican, A. Dosemeci, T. S. Reese, and J. Lisman. Persistent accumulation of calcium/calmodulin-dependent protein kinase II in dendritic spines after induction of NMDA receptor-dependent chemical long-term potentiation. *J Neurosci*, 24(42):9324-31, 2004.

[119] G. Palm, F. Schwenker, F. Sommer, and A. Strey. Neural associative memories. In A. Krikelis and C. Weems, editors, *Associative Processing and Processors*, Los Alamitos, 1997. IEEE CS Press.

[120] P. E. Patton and B. McNaughton. Connection matrix of the hippocampal formation: I. The dentate gyrus. *Hippocampus*, 5(4):245-86, 1995.

[121] P. Poirazi and B. W. Mel. Impact of active dendrites and structural plasticity on the memory capacity of neural tissue. *Neuron*, 29(3):779-96, 2001.

[122] G. J. Quirk, R. U. Muller, J. L. Kubie, and J. B. Ranck, Jr. The positional firing properties of medial entorhinal neurons: description and comparison with hippocampal place cells. *J Neurosci*, 12(5):1945-63, 1992.

[123] A. Renart, P. Song, and X. J. Wang. Robust Spatial Working Memory through Homeostatic Synaptic Scaling in Heterogeneous Cortical Networks. *Neuron*, 38:473-485, 2003.

[124] A. D. Reyes and E. E. Fetz. Two modes of interspike interval shortening by brief transient depolarizations in cat neocortical neurons. *J Neurophysiol*, 69(5):1661-72, 1993.

[125] A. Rodriguez-Vazquez, G. Linan, S. Espejo, and R. Dominguez-Castro. Mismatch-induced trade-offs and scalability of analog preprocessing visual microprocessor chips. *Analog Integrated Circuits and Signal Processing*, 37:73-83, 2003.

[126] V. Santhakumar, R. Bender, M. Frotscher, S. T. Ross, G. S. Hollrigel, Z. Toth, and I. Soltesz. Granule cell hyperexcitability in the early post-traumatic rat dentate gyrus: the 'irritable mossy cell' hypothesis. *J Physiol*, 524 Pt 1:117-34, 2000.

[127] W. B. Scoville and B. Milner. Loss of recent memory after bilateral hippocampal lesions. *J Neurol Neurosurg Psychiatry*, 20(1):11-21, 1957.

[128] R. Shi and T. Horiuchi. A summating, exponentially-decaying CMOS synapse for spiking neural systems. In S. Thrun, L. Saul, and B. Scholkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, Vancouver, 2003. MIT Press.

[129] M. F. Simoni, G. S. Cymbalyuk, M. E. Sorensen, R. L. Calabrese, and S. P. DeWeerth. A multiconductance silicon neuron with biologically matched dynamics. *IEEE Trans Biomed Eng*, 51(2):342-54, 2004.

[130] M. Sivilotti, M. Emerling, and C. Mead, "A novel associative memory implemented using collective computation," in *Artificial neural networks: electronic implementations*. Piscataway, NJ: IEEE Press, 1990, pp. 11-21.

[131] P. J. Sjostrom, G. G. Turrigiano, and S.B. Nelson. Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron*, 32(6):1149-64, 2001.

[132] W. E. Skaggs and B. L. McNaughton. Replay of neuronal firing sequences in rat hippocampus during sleep following spatial experience. *Science*, 271(5257):1870-3, 1996.

[133] S. Song, K. D. Miller, and L. F. Abbott. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nat Neurosci*, 3(9):919-26, 2000.

[134] K. Steinbuch. Die Lernmatrix. *Kybernetik*, 1:36-45, 1961.

[135] S. Sternberg. High-speed scanning in human memory. *Science*, 153(736):652-4, 1966.

[136] M. Stopfer, S. Bhagavan, B. H. Smith, and G. Laurant. Impaired odor discrimination on desynchronization of odour-encoding neural assemblies. *Nature*, 390(6655):70-4, 1997.

[137] S. H. Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry and engineering* , Addison–Wesley, Reading, 1994.

[138] B. Taba and K. Boahen. Topographic map formation by silicone growth cones. In S. Thrun, L. Saul, and B. Scholkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, Vancouver, 2002. MIT Press.

[139] B. Taba and K. Boahen. Silicon growth cones map silicone retina. In Y. Weiss, B. Scholkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, Vancouver, 2005. MIT Press.

[140] P. H. E. Tiesinga and J. V. Jose. Robust gamma oscillations in networks of inhibitory hippocampal interneurons. *Network: Computation in Neural Systems*, 11:1-23, 2000.

[141] Y. P. Tang, E. Shimizu, G. R. Dube, C. Rampon, G. A. Kerchner, M. Zhuo, G. Liu, and J. Z. Tsien. Genetic enhancement of learning and memory in mice. *Nature*, 401(6748):63-9, 1999.

[142] U. A. Ernst, K. R. Pawelzik, C. Sahar-Pikielny, M. V. Tsodyks. Intracortical origin of visual maps. *Nature Neuroscience*, 4(4):431-6, 2000.

[143] E. Tulving. Episodic memory: from mind to brain. *Annu Rev Psychol*, 53:1-25, 2002.

[144] M. Verleysen, B. Sirletti, A. Vandemeulebroecke, and P. Jespers. Neural networks for high-storage content-addressable memory: VLSI circuit and learning algorithm. *Solid-State Circuits, IEEE Journal of*, 24(3):562-9, 1989.

[145] G. V. Wallenstein and M. E. Hasselmo. GABAergic modulation of hippocampal population activity: sequence learning, place field development, and the phase precession effect. *J Neurophysiol*, 78(1):393-408, 1997.

[146] X. J. Wang and G. Buzsaki. Gamma oscillations by synaptic inhibition in a hippocampal interneuron network. *Journal of Neuroscience*, 16:6402-6413, 1996.

[147] X. J. Wang. Pacemaker Neurons for the Theta Rhythm and Their Synchronization in the Septohippocampal Reciprocal Loop. *J Neurophysiol*, 87:889-900, 2002.

[148] D. S. Wei, Y. A. Mei, A. Bagal, J. P. Kao, S. M. Thompson, and C. M. Tang. Compartmentalized and binary behavior of terminal dendrites in hippocampal pyramidal neurons. *Science*, 293(5538):2272-5, 2001.

[149] R. K. Weinstein and R. H. Lee. Architectures for high-performance FPGA implementations of neural models. *J Neural Eng*, 3(1):21-34, 2006.

[150] B. Wen and K. Boahen. Active bidirectional coupling in a cochlear chip. In Y. Weiss, B. Scholkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, Vancouver, 2005. MIT Press.

[151] J. A. White, C. C. Chow, J. Ritt, C. Soto-Trevino, and N. Kopell. Synchronization and oscillatory dynamics in heterogeneous, mutually inhibited neurons. *J Comput Neurosci*, 5(1):5-16, 1998.

[152] T. J. Wills, C. Lever, F. Cacucci, N. Burgess, and J. O'Keefe. Attractor dynamics in the hippocampal representation of the local environment. *Science*, 308(5723):873-6, 2005.

[153] D. J. Willshaw, O. P. Buneman, and H. C. Longuet-Higgins. Non-holographic associative memory. *Nature*, 222(197):960-2, 1969.

[154] G. M. Wittenberg and S. S. Wang. Malleability of spike-timing-dependent plasticity at the CA3-CA1 synapse. *J Neurosci*, 26(24):6610-7, 2006.

[155] J. Wittig, Jr and K. Boahen. Silicon neurons that phase-lock. In *Circuits and Systems, 2006, ISCAS '06. Proceedings of the 2006 IEEE International Symposium on*, volume 8, Kos, 2006.

[156] S. Yamamoto, N. Tanimura, K. Nagasawa, S. Meguro, T. Yasui, O. Minato, and T. Masuhara. A 256K CMOS SRAM with variable impedance data-line loads. *Solid-State Circuits, IEEE Journal of*, 20(5):924-8, 1985.

[157] A. Ylinen, A. Bragin, Z. Nadasdy, G. Jando, I. Szabo, A. Sik, and G. Buzsaki. Sharp wave-associated high-frequency oscillation (200 Hz) in the intact hippocampus: network and intracellular mechanisms. *J Neurosci*, 15(1 Pt 1):30-46, 1995.

[158] K. A. Zaghloul and K. Boahen. Optic nerve signals in a neuromorphic chip II: Testing and results. *IEEE Trans Biomed Eng*, 51(4):667-75, 2004.