ABSTRACT

Title of Dissertation:	A NEUROMORPHIC VLSI NAVIGATION SYSTEM INSPIRED BY RODENT NEUROBIOLOGY
	Shashikant Koul, Doctor of Philosophy, 2019
Dissertation directed by:	Professor Timothy K Horiuchi, Department of Electrical and Computer Engineering

Path planning is an essential capability for autonomous mobile robot navigation. Taking inspiration from long-range navigation in animals, a neuromorphic system was designed to implement waypoint path planning on *place cells* that represent the navigation space as a cognitive graph of places by embedding the place-to-place connectivity in their synaptic interconnections. Hippocampal place cells, along with other spatially modulated neurons of the mammalian brain, like grid cells, head-direction cells and boundary cells are believed to support navigation. Path planning using spike latency of place cells was demonstrated using custom-designed, multi-neuron chips on examples and applied to a robotic arm control problem to show the extension of this system to other application domains. Based on the observation that varying the synaptic current integration in place cells affects the path selection by the planning system, two models of current integration were compared. By considering the overall path execution cost increase in response to an obstruction in the planned path

execution, reduced spike latency response of a place cell to simultaneously converging spikes from multiple paths in the network was found to bias the path selection to paths offering more alternatives at various choice points. Application of the planning system to a navigation scenario was completed in software by using a place-cell based map-creation method to generate a map prior to planning and co-opting a grid-cell based path execution system that interacts with the path planning system to enable a simulated agent to do goal-directed navigation.

A NEUROMORPHIC VLSI NAVIGATION SYSTEM INSPIRED BY RODENT NEUROBIOLOGY

by

Shashikant Koul

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2019

Advisory Committee:
Professor Timothy K Horiuchi, Chair
Professor Nuno C Martins
Professor Pamela A Abshire
Professor Perinkulam S Krishnaprasad
Professor Daniel A Butts, Dean's Representative

Acknowledgements

I would like to thank my advisor, Dr. Horiuchi for his confidence in me. Any milestone in this research would not have been possible without his encouragement, support and patience. He also took an active part in my work and provided a lot of guidance. He has played a major role in shaping me as a writer and a presenter. He encouraged analyzing problems from multiple perspectives and creating project proposals. He strongly recommended taking part in workshops, conferences and communicating with the other members in the research community. I also got an opportunity to learn teaching skills from him when he taught the Analog and Digital Electronics course at the university and it was refreshing to see him bring his own knowledge and experience to the classroom.

I would like to thank Dr. Pamela Abshire and Dr. Nuno Martins, who were on my proposal committee for their feedback. I would also like to thank Dr. Krishnaprasad and Dr. Dan Butts for serving on my dissertation defense committee.

I would like to thank Dr. Tarek Massoud for the early design of the multi-neuron chip. I would like to thank Dr. Cynthia Moss for early discussions on bat behavior. I thank Justin, Jacob and Chenxi from the CSSL group here. Their energy, thoroughness and honest conversations were very motivating. I would like to thank Bathiya and Alex for the preparation and practice we did together in the early part of the program. I would like to thank Bathiya, Vikram, Venkat and Jacob who helped in proofreading. I would like to thank Vikram, Venkat, Ninad and Avinash, who listened and advised whenever I was stuck in research.

I would like to thank the Office of Naval Research for financial support. I would like to thank MOSIS that provides fabrication service for research and educational purpose.

Finally, I would like to thank my father for his patience and understanding. He always advised me to keep moving and working. I would like to thank my mother for her trust in my abilities. I would like to thank my brother for his help in difficult times. I would like to thank my wife for encouraging me, when the going got tough, for listening and creating a good atmosphere at home.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Motivation and Problem Statement

Understanding the neural mechanisms underlying animal navigation over long distances based on spatial memory has the potential to help build better autonomous mobile robots as well as provide insight into how sequential planning could operate in any task domain. Most extensively studied in rodents, the hippocampus and associated brain areas have been implicated in memory-based navigation tasks (e.g., mazes). Place cells, found in the mammalian hippocampus, are believed to be involved in the implementation of a neural map of space [1]. In this research, we present a spiking neuron model and a neuromorphic implementation for path planning inspired by place cells that use spike latency in the pathfinding process (see Fig. 1.1).

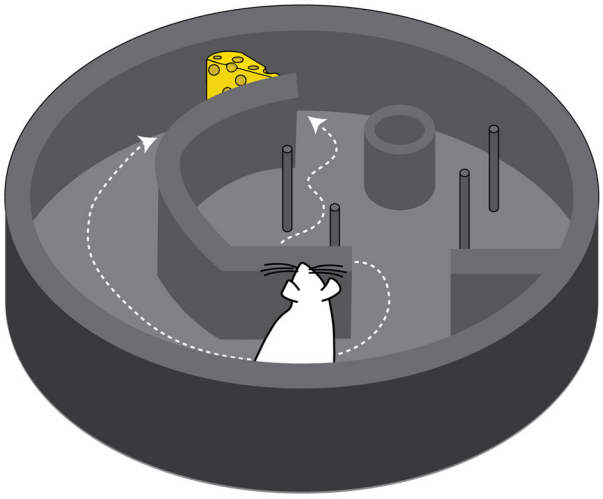


Fig. 1.1. A hypothetical experiment is described, where a rat is deciding between multiple paths to the goal in a maze. The maze is mentally represented as a collection of places, where being at a place is signaled by the activation of a place cell in rat's brain. If spreading waves of neuronal activity indicate a way to the goal, then the paths selected by the rat would depend upon the transmission of spikes between the neurons. Depending upon how neurons integrate the charges deposited by spikes, different paths may be selected for navigation. From [2].

A neural map is different from our common understanding of a map, metrically accurate, at a fixed scale, and acting as a passive repository of information [3]. The neural map is not metrically accurate everywhere, represents only locations that are known, and is embedded in the interconnections of a neural network that can perform data and context-dependent computations. This map can be visualized by taking electrical recordings from place cells in rodents that are performing tasks in a testing arena and superimposing the time-averaged spiking activity of the place cells on an image of the testing arena. The place cells' receptive fields (a.k.a. place fields) are responsive to sensory and odometric information. The shape and size of the place fields are created based on the context and are not fixed [4]. Recordings made while an animal is asleep reveal that place cells become active in sequences that corresponds to a replay of the activity recorded during wakeful experience [5]. Additionally, when the animal is awake and its motion is interrupted, the sequences of place cells activity are suggestive of planning [6]. The place cells appear to operate as a dynamic interface to the spatial memory and can be used for probing the relationship between known places and thus memory-based path planning.

Although long-range path planning based on conventional large-scale metrical maps can accomplish the task, a neuromorphic solution based on a neuromorphic spatial map would likely be a memory and energy efficient solution for doing this processing on-board in the long term [7], [8]. Place cell based planning has been tested using software simulations [8]–[11], robotic implementations [12], [13] and spiking neuron based path planning [7], [14] on neuromorphic VLSI. This research adds

decoding the spiking activity to find direction for path execution and how introduction of biologically-plausible synapse and neuron properties influence path planning.

1.2 Dissertation Overview

The dissertation is organized as follows. The second chapter provides the background for this research. Models for the representation of space, spatially-modulated neurons of the brain and models for navigation are described. The third chapter describes the navigation system model. An overview of the navigation system and simulations for mapping, planning and path execution are presented. The fourth chapter covers design, operation and testing of the neuromorphic VLSI system. The system organization and the operation of multi-neuron chips are explained and characterization tests necessary for using the chips for path planning are reported. The fifth chapter explains how the neuromorphic VLSI system is used for path planning. In this, the operation of the temporal winner-take-all, common path planning examples and controlling a robotic arm example are discussed. The sixth chapter introduces synaptic-dependent spike-latency models, presents examples that use these models and provides a method to interpret the effect of the model choices on path planning. The seventh chapter provides a short summary, contributions of this research and possible improvements. In Appendix-A, the effect of fabrication mismatch on spike latency is analyzed. In Appendix-B, the effect of transistor size on subthreshold current for transistors with the same aspect ratio is presented. In Appendix-C, programming PFET floating gate voltage by tunneling and impact-ionized hot-electron injection is discussed for some test circuits and extracted model parameters for SPICE simulation of floating gate transistors are provided.

Chapter 2: Background

2.1 Spatial Representation in Animal Navigation

Animals use many strategies for navigation. For short range movement, actions such as doing a random search, following a beacon, keeping track of self-position using dead reckoning and piloting with respect to the surrounding scene [15] are used. To carry out long-range navigation, however, an animal must possess a mental representation of different "places" and their spatial relationships. Exactly how this is represented in the brain is still a topic of debate. Possible theories (see Fig. 2.1) suggest place recognition-triggered responses, topological map or "survey map" like representations [16].

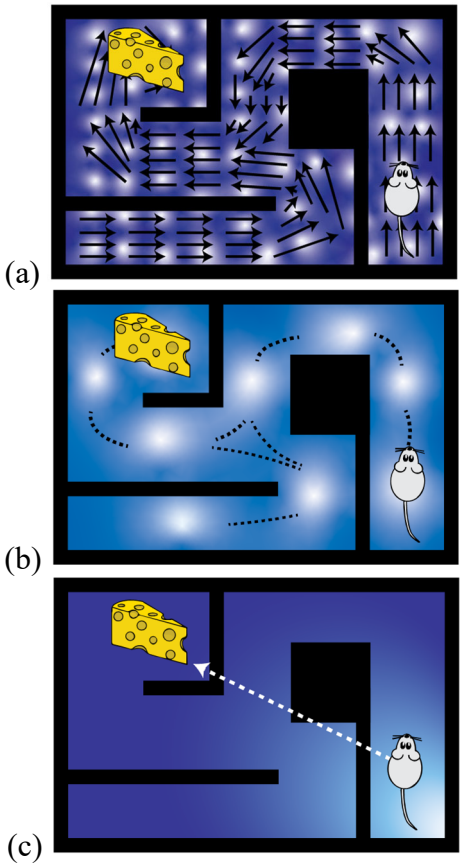


Fig. 2.1. Three different hypothetical spatial representations of a maze in a rat's head for finding the cheese, based on [16]. (a) A fine-grained, memorized, behavior policy map leading the rat from any starting location to the cheese. Once the map is in place, the rat does not need to make decisions. Returning to the starting position would require a new map. (b) A memorized spatial relationship map does not prescribe actions, but just informs the rat how to move from place to place. To navigate, the rat must explore its memory and make decisions. (c) Here, the rat infers from memory the distance and direction to the cheese and determines in real-time how to navigate the obstacles. From [2].

2.2 Spatial Cells in Rodents

Neurophysiological research on rodents in mazes suggests that place cells provide a neural substrate for a spatial map [1] (see Fig. 2.2). Experiments reveal that place fields represent recognized places in an allocentric reference frame. Interestingly, a place cell can represent multiple places in different contexts or environments. It should be noted that while much research has described how place cells respond to both

sensory cues and to odometric cues, the circuitry for producing these place fields is still poorly understood. Other types of cells (see Fig. 2.2) that are known to provide spatial information are head-direction cells, grid cells and boundary cells [17], found in the parasubiculum, entorhinal cortex and subiculum [18].

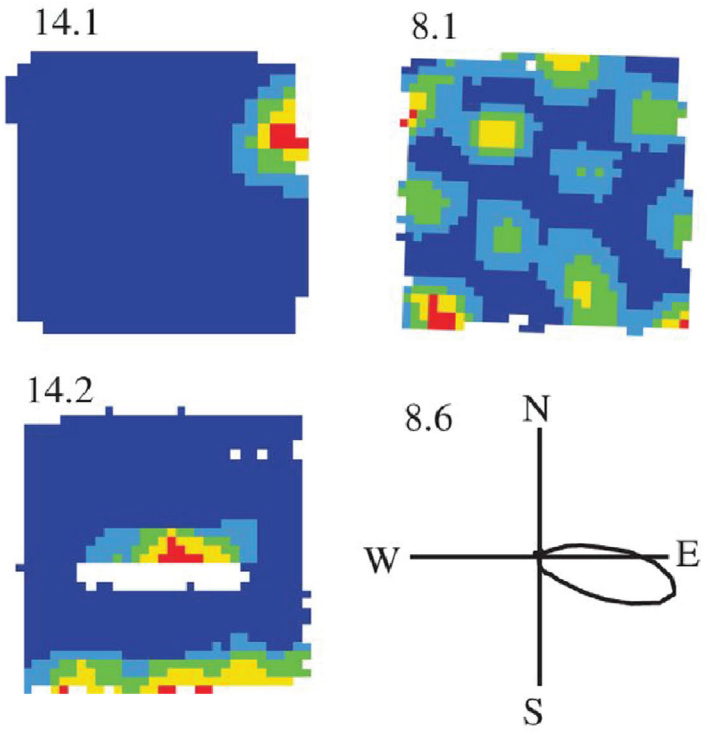


Fig. 2.2. Showing the spatially dependent activity of different cells in the hippocampal formation and surrounding regions that represent space. From the top-left going in clockwise direction: place cell, grid cell, head direction cell and boundary cell. Place cell activity represents the memory of specific locations. Grid cells have a triangular-grid activity pattern that appears to provide a metric representation of space. The head direction cell activity indicates when the animal's head is pointed in a certain direction. Boundary cell [17] activity indicates when an animal is facing a boundary in a certain direction. Modified with permission from [18].

2.3 Models for Spatial Navigation

How animals associate actions to reach goals using place cells is currently believed to be based on reward or latent learning mechanisms [19]. Synaptic weights from place cells to motor neurons could be modified using a delayed reward activity such that, during navigation, the activity of place cells could activate the motor neurons to replicate previous actions that generated the correct movement towards a goal [20]. Alternatively, the *spatial relationship* between places could be learned using the synaptic connections between their corresponding place cells. The direction to the next intermediate place of movement could be found by a phenomenon akin to mental search. There are several ways in which this process could be modeled [10], [11], [21].

2.4 Prior Work in Neuromorphic VLSI

Beyond software simulation, neuromorphic VLSI has been used to simulate the function of head direction cells, grid cells and place cells [22]–[24]. To build memory-based navigation capability on-chip, these elements are beginning to be integrated to develop a model of map creation and use [53],[54]. In this research, it is assumed that the neural map is a topological map of recognizable places and that animals navigate by mentally "hopping" sequentially between places in the known map. The interconnected network of place cells can be depicted as a graph, where the nodes of the graph correspond to place cells and the relative positions of the nodes correspond to the relative positions of the places in the physical space where the place cells become active. Thus, neighboring place cells in the graph correspond to neighboring places in the physical space. We simply refer to this network as the "map" in this dissertation. Memories of places and the spatial relationships between places are stored when

animals explore an area for the first time. Also, the spatial relationship between a place and neighboring places is stored as a "sensory view" observed at the place. By recalling the sensory view at the present location and knowing the next place to navigate to in this view, the animal is able to direct itself to the next place in the path to the goal (i.e., waypoint). In the work presented here, the next step in the path is found by: 1) spreading activation in all directions through the network starting from the place cell representing the goal location, 2) by monitoring the network neighbors of the present-location place cell for the arriving wave of activation, and 3) identifying which neighbor activates first, representing the next closest place along the shortest path (in hops) to the goal (see also [25], [26]). This model uses *time* to integrate the number of steps to the goal along multiple paths simultaneously. Long-range navigation is typically broken into temporally distinct steps such as *mapping*, *planning* and *execution* of the plan. This research focuses on the *planning* portion of the problem. We will, however, discuss how the mapping and execution behaviors interact with planning because they are intimately coupled. We note that there are other models that propose that place cells represent the transition between places that are represented in the entorhinal cortex [13].

Chapter 3: Neuro-Inspired Navigation System Model

3.1 Introduction

Navigation, the act of deciding where to go and the monitoring of this movement, is a critical survival skill. Although navigation can involve many different spatial scales and sensory systems, in this research, the focus is on path-planning based on spatial memory (i.e., internal representations of space) and occurs prior to each movement.

In this chapter, prior modeling efforts towards designing hippocampally-inspired navigation systems are discussed, followed by a description of our navigation system and computer simulations to explain the working of the entire system on an example maze. Navigation is treated as a process with three temporally distinct steps: mapping (prior experience), path planning (based on internal memory) and path execution (external motor actions). Simulations of the three processes are explained to provide a context for the neuromorphic path planning system that is presented in the next chapter.

3.2 Prior Models

Various approaches have been used to model a biologically plausible navigation system that can explain the role of place cells in navigation. Using odometry to represent inputs and using place cells as basis functions to represent space are popular choices [10], [11], [27]–[29]. There are examples of vision-based place cells representations as well [20], [30], [31]. Spatial learning examples have been demonstrated by showing generation of place fields in response to odometry and/or

visual inputs [20], [32]. Navigation learning examples have been demonstrated by creating interconnected network of place cells where the movement between the starting and ending locations is determined by the movement between the intermediate waypoints [11], [27], [30] or by associating an action with every place cell for a particular goal [29], [32], [33]. Some methods combine both approaches for navigation [9], [34]. While using an interconnected network of place cells, graph search or activity diffusion is used to determine the sequence of actions [10], [11], [21], [30]. In the other approach, also called the stimulus-response strategy, place cell activity is sufficient to drive the motor neurons activity [27], [29], [32]–[34]. In the surveyed literature, in models that use activity diffusion to explore the candidate paths, the activity is either initiated from the goal location place cell or from a neighbor of the present location place cell. When the activity is initiated from the goal location place cell, the neighboring neurons of the present location place cell are probed to select the next action [10]. When activity is initiated from a neighbor of the present location place cell, then the activity level at the goal location neuron is evaluated for different neighbor neurons or directions [21] before selection the action for movement. Path execution is generally carried out by generating a heading direction based on comparison of the stored odometry information [11], [27] or stored visual snapshots [30] or by executing actions associated with the current place [32]–[34]. Many models have been tested on mazes in simulations that were previously used in the wayfinding experiments [10], [29], [33], [35] and some have been implemented on robots as well [20], [30], [31].

Our approach follows the activity diffusion strategy, however, we make use of spike latency instead of firing rate. Instead of carrying out direction probing or using

directed activity gradients, we use the notion of top-down attention to simultaneously consider all directions possible at the current location. Other work that use a similar approach are [25], [26]. In [25], rate based neurons are used to represent place cells. This causes activity spreading from the goal to decay exponentially over the neural network limiting the maximum range between the goal and present location. In [26], spiking neurons are used, but instead of using spike latency, phase of spiking frequency is used. The place cells are stimulated externally and the goal place cell is stimulated with a stronger input. This creates a slight shift in the phase of periodically firing neurons in the network starting from the goal location place cell. Using a difference in phase of firing in the place cells neighboring to the present location place cell, the direction of movement is determined. It is noted here that the neuromorphic VLSI system uses spiking neurons, but in the simulation firing rate based neurons have been used.

3.3 System Overview

In this research, mapping, path planning, and path execution processes are carried out in sequence, i.e. they do not operate simultaneously. Mapping is carried out while the agent is exploring the maze. After exploration is complete, the agent can revisit a previously memorized location by completing multiple cycles of path planning and path execution until the goal is reached.

It is assumed that mapping and planning are two distinct steps. It has been observed in rodent electrophysiology experiments that when an animal, moving about in a maze or an arena, switches its behavior from moving to waiting or vice versa, the hippocampal place cell activities transition from one type of activity to another [18].

We assume that while the animal is moving, it creates a spatial representation of the environment and when it comes to a stop, it can plan its next movement. When the animal explores, different place cells in its hippocampus fire at various locations in the environment and the activity response of place cells to these locations are learned during the exploration. In the periods while the animal is stationary, there is large irregular activity in the hippocampus that are called sharp-wave ripples. During these events, time-compressed sequences of activities have been observed [36] and it is believed that some of these activities could represent planning [6]. Although these sequences are interpreted in the context of forward planning starting from the present location, there is evidence for "remote" replay of activities as well [37]. We note that this description also implies that switching between mapping and planning occurs from time to time, but in this work, mapping is completed first and then planning is carried out.

Place cells exhibit many other properties that have not been incorporated in this work. For example, place cell spikes exhibit phase precession [38] with respect to the theta rhythm in the local field potential. Place cells also exhibit sequential activities during sleep [5], [39]. Many of these sequences are replays of recent activity recorded during a waking experience and there is a study reporting Brownian diffusion-like activity in place cells [40] that are believed to support memory consolidation. Although these processes could interact or affect the map formation and planning processes, these are not included in this research.

3.4 Map Formation

To explore how a map is created, a MATLAB® simulation of both an agent and its environment was created. An example of a maze exploration along with agent visit frequency of different locations in the maze is provided in Fig. 3.1. The agent explores the maze using an obstacle avoidance algorithm [41] based on a simulated sonar sensor.

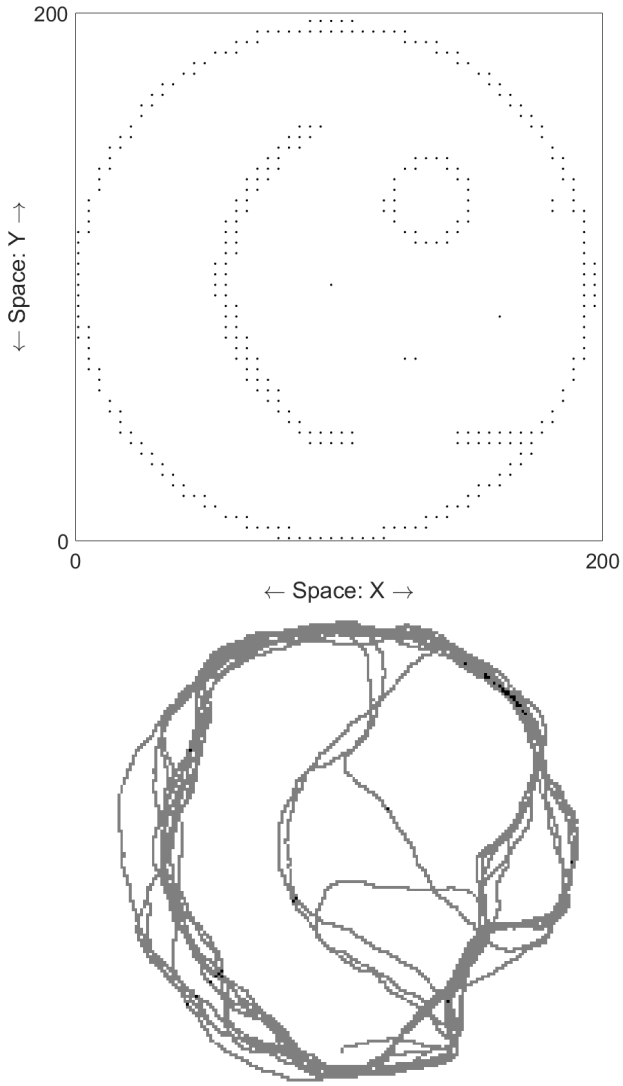


Fig. 3.1. Top: Example maze exploration using a simple obstacle avoidance mechanism [41]. Bottom: Number of visits at different locations inside the maze. Bins with visits greater than 5 are black in color. Bins with no visits are white in color.

Maps can be created on the first exposure to the environment (first foray), however, multiple opportunities to explore will enrich the map connectivity and more fully discover how places are interconnected. The agent has a population of place cells that initially do not represent any place in the maze. If an agent encounters a new place, a new place cell is selected from a reserve population to represent it. As the agent explores, if there is very low activity in the place cell population (i.e., maximum

activation is below a threshold level), then a place cell that has not been recently active is recruited from the population. Specifically, when the maximum activity falls below a certain threshold (ϑ), a new place cell that has the maximum activity is selected. Equations (3.1)-(3.3) represent the place cell "recruitment" process. When a place cell is recruited, its t_{pot_i} that is initially infinite is assigned the time of recruitment and the place cell is associated with the coordinates of the agent. Additionally, it is not used immediately to represent another place in the same environment. Using a parameter, ρ that decays with time, the recruited place cell is made less likely to be selected again. The parameter values are provided in Table 3.1.

If $$\sum_{p_i \in P} H\left(r_{p_i}(t) - \vartheta\right) = 0 \tag{3.1}$$

Where H() is the Heaviside step function and *P* represents the population of place cells.

Then $$p(t) = \underset{p_i \in P}{arg\max}\{r_{p_i}(t) - \rho H(t - t_{pot_i})\} \tag{3.2}$$

$$\left(x_{prf_{p(t)}}, y_{prf_{p(t)}}\right) = (x_b(t), y_b(t)) \tag{3.3}$$

Table 3.1. Parameter values for place cell recruitment and map formation

parameter	value	units
ϑ	0.5	Hz
ρ	2	Hz

The receptive fields of the place cells (a.k.a., "place fields") are modeled as Gaussian functions responsive to the position of the agent at the time of recruitment. Although these receptive fields could be created using sensory information (vision or sonar echo patterns) and/or a grid cell-based neural model of odometry, for the

purposes of demonstrating the path planning chip, we have chosen to model the place fields simply by using internal simulation coordinates. If using sensory and odometric inputs, the method of growing cell structures [42] could be used to create the map. The actual function used in the simulation to model the place cell activity function is a hyperbolic tangent of a Gaussian function and the mean is the location, where the place cell was recruited. Equations (3.4) and (3.5) represent the place cell activity. Their parameter values are provided in Table 3.2.

$$r_{p_i}(t) = \frac{1}{2}\left(1 + \tanh\left(\alpha_p\left(exp\left(-\frac{d_{p_i}^2(t)}{\sigma_p^2}\right) - \theta_p\right)\right)\right) \tag{3.4}$$

$$d_{p_i}^2(t) = (x_b(t) - x_{prf_i})^2 + (y_b(t) - y_{prf_i})^2 \tag{3.5}$$

p_i is the i[th] place cell and $r_{p_i}(t)$ represents the firing rate activity. ($x_b(t)$, $y_b(t)$, $\varphi_b(t)$) are the state coordinates of the agent. x_b and y_b represent its position in the maze and φ_b represents its orientation with respect to fixed axes. (x_{prf_i}, y_{prf_i}) was used to model the place cell receptive field. α_p, θ_p shape place cell activity function and σ_p controls the width of the place field. It is of interest to note that these neurons are many steps away from the sensory or motor stimuli, but in our simulations their values represent the instantaneous location of the simulated agent. Prediction using motor information could support this assumption [43].

Table 3.2. Parameter values for place cells

parameter	value
α_p	10
θ_p	0.8
σ_p	60

As the agent moves through the arena, different groups of place cells become active as the agent passes through their overlapping receptive fields. Synaptic links are created between place cells when they are co-active. Through this process, a graph of linked places is created to represent the agent's understanding of the maze (see Fig. 3.2 and 3.3).

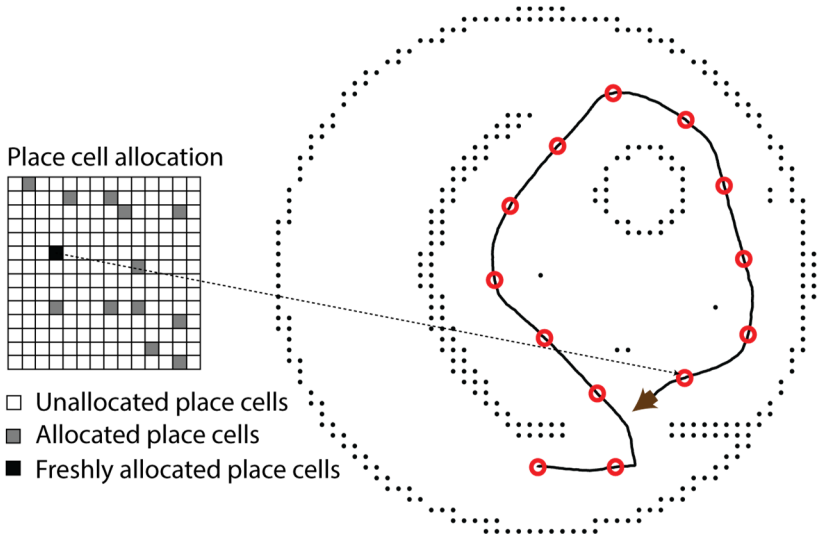


Fig. 3.2. During maze exploration, place cells are linked to create a map. The agent starts exploring the maze at the bottom. A snapshot of the place cell population is shown on the left. Square colors represent different states: white: cells not representing locations in the map, grey: cells already representing locations in the map, and black: freshly recruited place cells. The dots in the maze represent obstacles and there are two main sections in the maze separated by an arc-shaped wall. The agent is initially oriented towards the section on the right and is guided by an obstacle avoidance algorithm with a default to move straight-ahead. As the agent explores, place cells from its reserve population are recruited to represent places that are not recognized from memory. The receptive field centers of place cells receptive field are indicated on the maze by a circle, the location of the animal when the place cell was recruited. When the agent returns to the vicinity of a place it has visited before, its corresponding place cell reactivates and no new place cells are recruited. From [2].

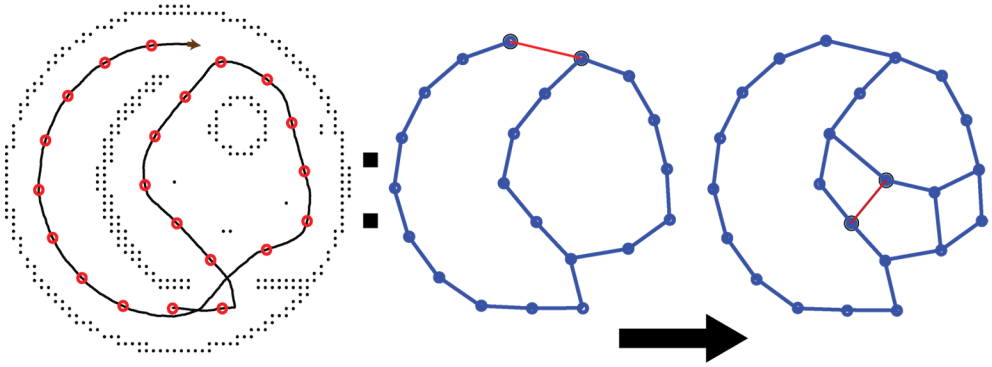


Fig. 3.3. Showing the map formation process. As the agent continues exploring the maze, more place cells are recruited to represent new places and links are formed between the place cells that represent the possibility of passage between the two places. The thick blue lines indicate previously formed links and the thin red line indicates the formation of a new link. The figure on the right indicates the final network of place cells that was created for the maze. From [2].

There are many options for the decision to create a synaptic link between place cells, however, in this model of map creation, a link is meant to represent traversability and in most cases this is bidirectional even if it was not directly experienced. As a result, co-active place cells can sometimes be inappropriately linked even when no traversable path exists (e.g., when an agent activates place cells on either side of an obstacle.) It should be noted that this bidirectional synaptic link represents the memory of expected traversability and not the specific experience that led to this memory. In addition, the link will need to contain more information for the agent to move from place to place.

An example of the map and motor actions that allow movement between places are shown in Fig. 3.4. In the map, the nodes represent places and edges represent traversable paths. The edge density for every node is determined by the environment that is mapped. It is believed that animals create and maintain separate maps for distinct environments [44]. There are computational studies [45], [46] that provide a mechanism to store multiple maps in the same neural network map. However, for simplicity, only one environment map is used in this simulation.

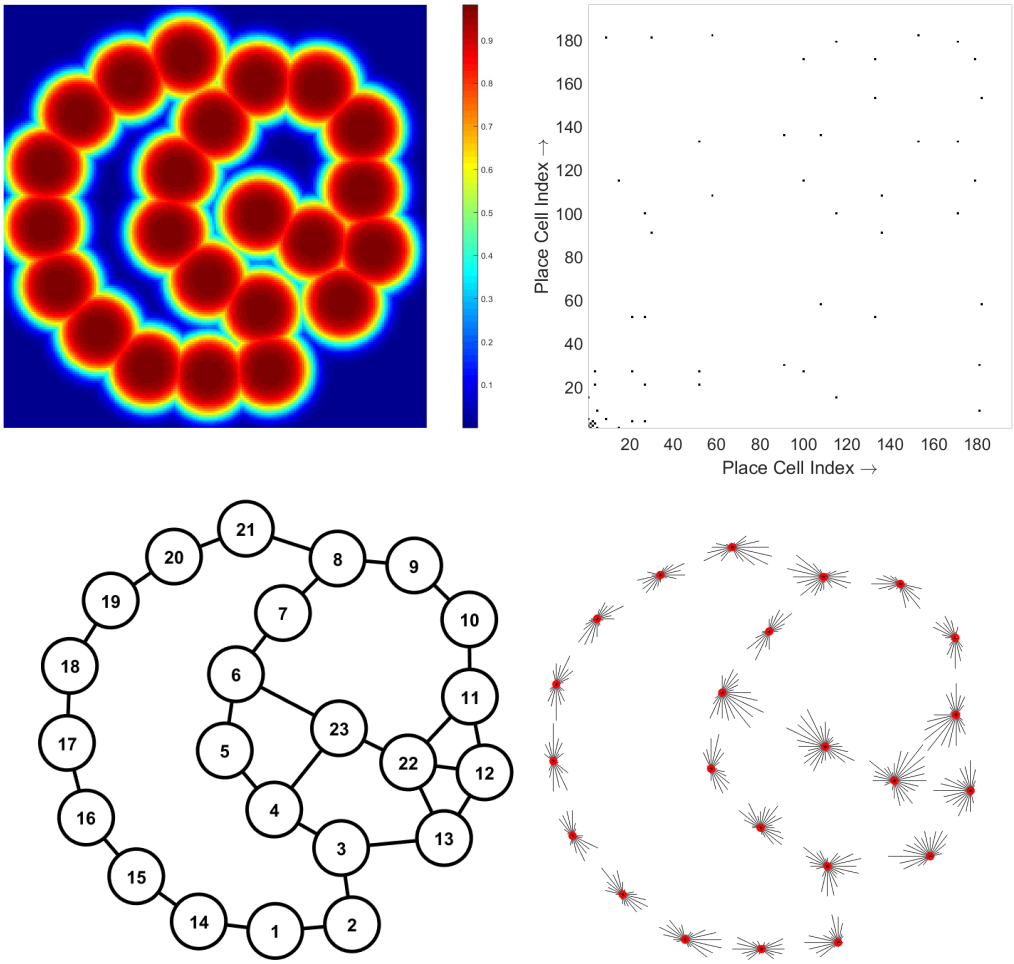


Fig. 3.4. Top-Left: Place fields of the place cells recruited during the mapping simulation. Top-Right: Place cell to place cell links are indicated by a black bin at place cell index coordinates. Bottom-Left: Map showing place cell to place cell connectivity. Bottom-Right: A whisker plot was generated by plotting the response of motor neurons to grid cell activities corresponding to connected place cells in the cognitive graph.

It is also possible that other places were visited, but their corresponding place cells were not linked, for example, when a new passage is found. As a result of this implementation, new links to previously visited place cells are made earlier than new links to new place cells (see Fig. 3.5). The place cell association with 2-D coordinates are assumed to be stable.

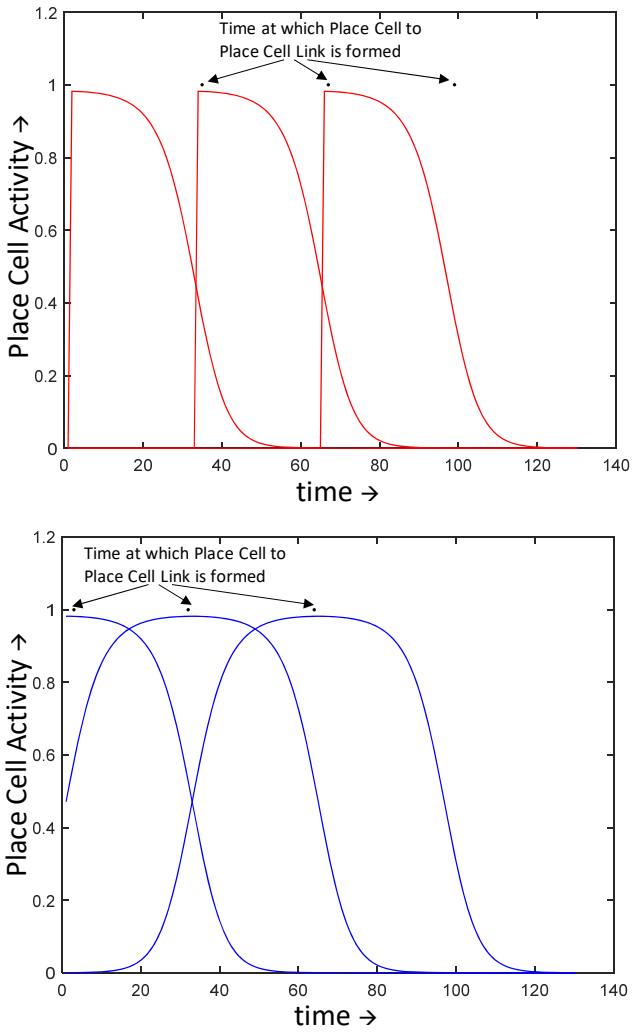


Fig. 3.5 The top panel shows place-cell-to-place-cell link formation during the first trial in a 1-D passage. The dots at the top represent the time of place-to-place link formation. The bottom panel shows place-cell-to-place-cell link formation in a subsequent trial, where the place cells have been recruited, but no links were previously created. Such a scenario can take place where walls between adjoining places were removed after place representations were learned. The dots at the top represent place-to-place link formation time. The linking time is earlier in the scenario when place fields already exist.

3.5 Path Planning

Once the map is acquired, the agent can perform a search to go to a memorized target. This process is similar to the breadth-first search algorithm and it is used to find the sequence of waypoints to reach the target. The process is called spreading activation

and this mechanism is an attractive choice for modeling the search process. An example of a typical spreading activation mechanism is shown in the Fig. 3.6.

In the neural network, a short stimulation of the target neuron, similar to [11] activates it and causes a controllable spread of activity across the network. Equations (3.6)-(3.8) are the neuron model equations used.

$$\frac{ds_{p_i}}{dt} + \frac{s_{p_i}}{\tau_p} = I_{s_i} + I_{bias} + \alpha_r tanh\left(\sum_j w_{pp_{ij}} r_{p_j}\right) - \beta_r I_{Ca_i} \tag{3.6}$$

$$r_{p_i} = \frac{1}{2}(1 + \tanh(\alpha_p(s_{p_i} - \theta_p))) \tag{3.7}$$

$$\frac{dI_{Ca_i}}{dt} + \frac{I_{Ca_i}}{\tau_{Ca}} = \gamma_r(1 - I_{Ca_i})r_{p_i} \tag{3.8}$$

I_{bias} is a lumped current injected into all neurons and models the background current. I_{s_i} is a stimulus current provided to i^{th} neuron. It is used to provide stimulation to the target place cell and inhibition to the place cell near a threat. s_{p_i} represents the synaptic charge in the neuron. The synaptic charge is also provided by recurrent connections from other neurons in the network. The neuron has an adaptation current I_{Ca_i}that acts as the refractory period and makes the neuron repolarized after a short span of activity. τ_p and τ_{Ca} are time constants of the neuron and the adaptation current. α_r controls the contribution of other place cell activities to the firing of a place cell, β_r controls the strength of Calcium adaptation current and γ_r is one of the parameters that controls the delay in adaptation. α_p, θ_p are same as in equation (3.4). In an actual neural circuit, the interneurons play an important role in ensuring that the activity is controlled, however, in this simulation, we do not model these. The parameter values used in the equations are provided in Table 3.3.

Table 3.3. Parameter values for spreading activation in the map network

parameter	value
τ_p	3.33
τ_{Ca}	2
α_r	1.5
β_r	3
γ_r	0.9
I_{bias}	0.2

For selecting an action, it is important to know where the agent is presently at and what actions are possible at the present location. Every place cell in the network has a set of direction neurons corresponding to its neighboring place cells in the map. Each direction neuron receives an input from the corresponding neighboring place cell and sends an output to a local inhibitory neuron that sends an inhibitory input back to all direction neurons associated with the place cell, forming a winner-take-all (WTA) network. When the goal-location place cell on the map is stimulated, it causes the activity to spread over the network. At the present location place cell, the direction neuron that receives the earliest activity becomes active and by activating the local inhibitory neuron inhibits the other direction neurons. We assume that there is a mechanism to attend to only the direction neurons corresponding to the present location place cell and by observing the activity of the direction neurons, the next waypoint to move to is known. This is shown in Fig. 3.7.

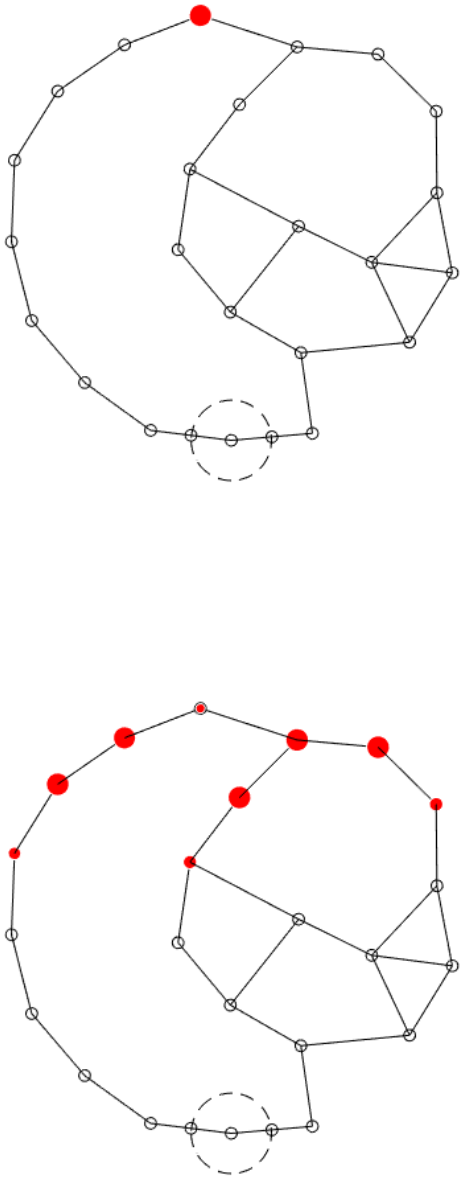


Fig. 3.6. Top panel: Time zero for spreading activation through a network, where the activity starts from the node (representing a place cell) with the red filled circle. The size of the red circle represents the activity level of the place cells. Bottom panel shows the population activity after a few time steps. The activity of the starting node drops and the activity wave spreads through the network. The dashed circle represents a winner-take-all with the direction neurons and the node inside of it represents the present location place cell.

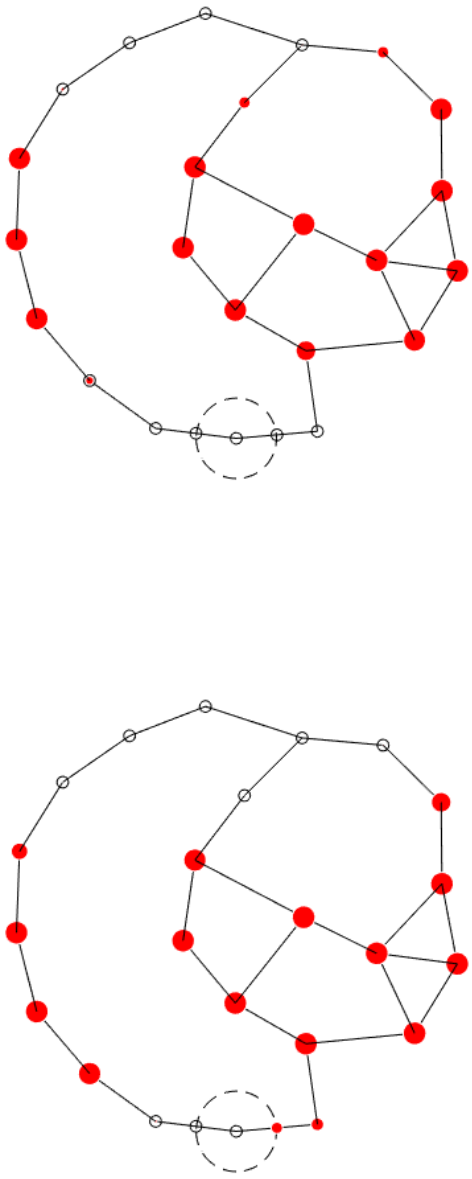


Fig. 3.7. Top panel shows spreading activation through a network. The size of the filled red circles represents the activity level of the place cells. In the bottom panel, activation of the direction neuron is shown. The dashed circle represents a winner-take-all with the direction neurons and the node inside of it represents the present location place cell. Here the direction neuron on the right-side, corresponding to the activity wave coming from the right is activated.

3.6 Path Execution

While the "winning" direction neuron indicates the next place cell along the path on the graph, the actual direction of movement from one place field to the next place field in the physical space is also required. A way to accomplish this is provided in [47] (also see [48] for variants of this approach and [49] for practical implementation) where activity patterns of two populations of grid cells are compared to move between two locations.

The activity pattern of the first grid cell population provides the present location and the second grid cell population acts as a memory to previously visited places. This model is easily incorporated in the mapping process. During the mapping step, when a place cell is recruited to represent a place, grid cell activity pattern at that moment is stored. During execution, the grid cell activity pattern that is associated with the remembered place is recalled and compared with the present grid-cell activity pattern to compute the direction for movement (see Fig. 3.8).

An example is shown in Figs. 3.9 and 3.10 to explain path execution. In this example, the waypoint location is at the origin. The agent is at some distance from the waypoint. Due to inaccuracy in landing at the center of the place field, it is assumed that the agent could be located anywhere along a horizonal line and based on the response of the motor neurons to the grid cell activities, the agent will move towards the waypoint located at the origin. The direction computed to the waypoint is indicated using the direction vectors in Fig. 3.9. The error in the direction estimates is also shown. The actual path taken by the agent starting from any location along the horizonal line to the waypoint is shown in Fig. 3.10. Also, the error distance from the center of the

place field is shown. The movement is terminated when the maximum activity of the motor neurons drops below a threshold value.

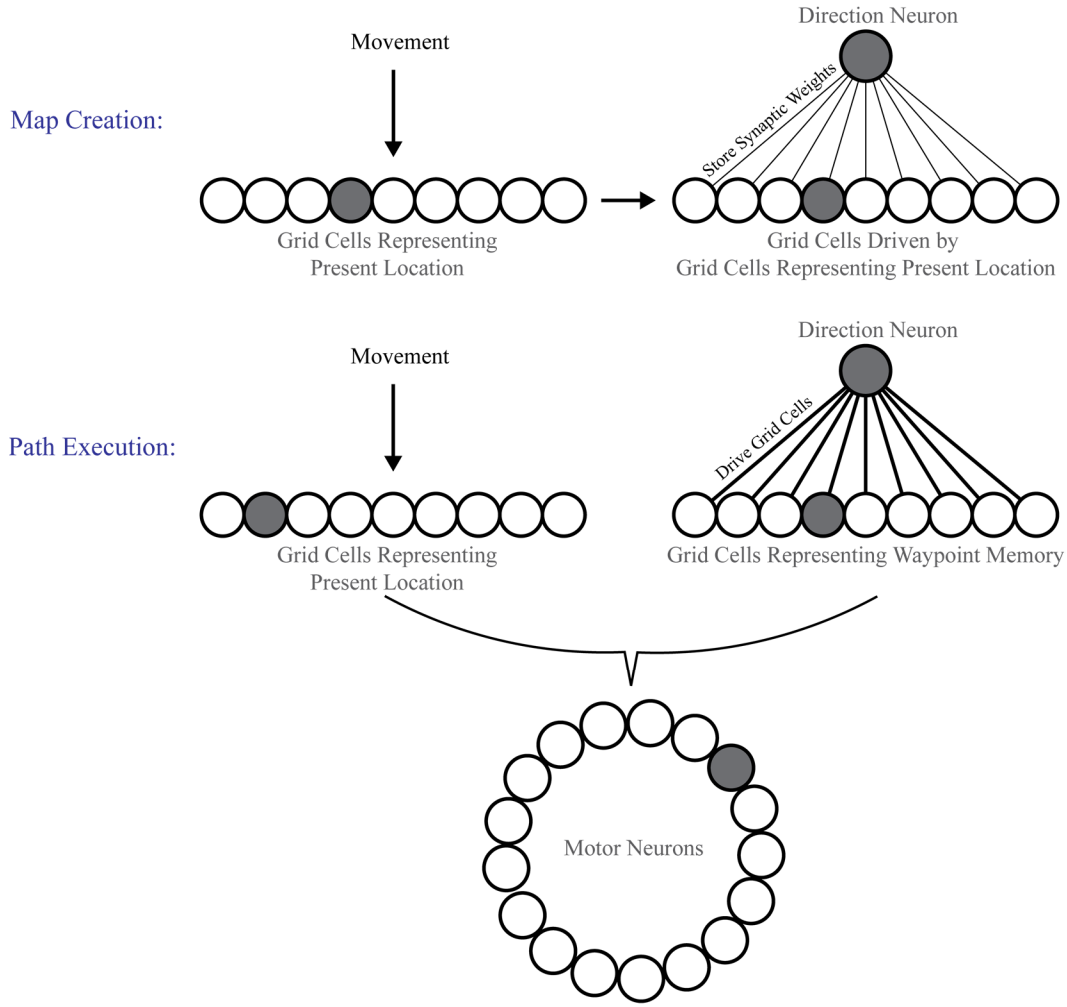


Fig. 3.8 The top figure shows the association between grid cell activity pattern and direction neurons that happens during place cell recruitment while the agent explores the environment to create a spatial map. The bottom figure shows path execution, where the waypoint memory is reactivated by the direction neuron that fires after the path planning cycle completes. By comparing the activity patterns of the two grid cell populations, it is possible to determine the movement direction to the waypoint. Based on [50]. NOTE: Solid gray indicates active neuron.

In the mapping step, the grid cell activity pattern is stored in the synaptic weights from direction neurons to the population of grid cells, such that when the direction neuron is selected later by the WTA, its activation recreates the grid cells

activity pattern. By computing the physical direction to the next place cell, the agent moves to the next waypoint in the plan. This is shown in Fig. 3.11.

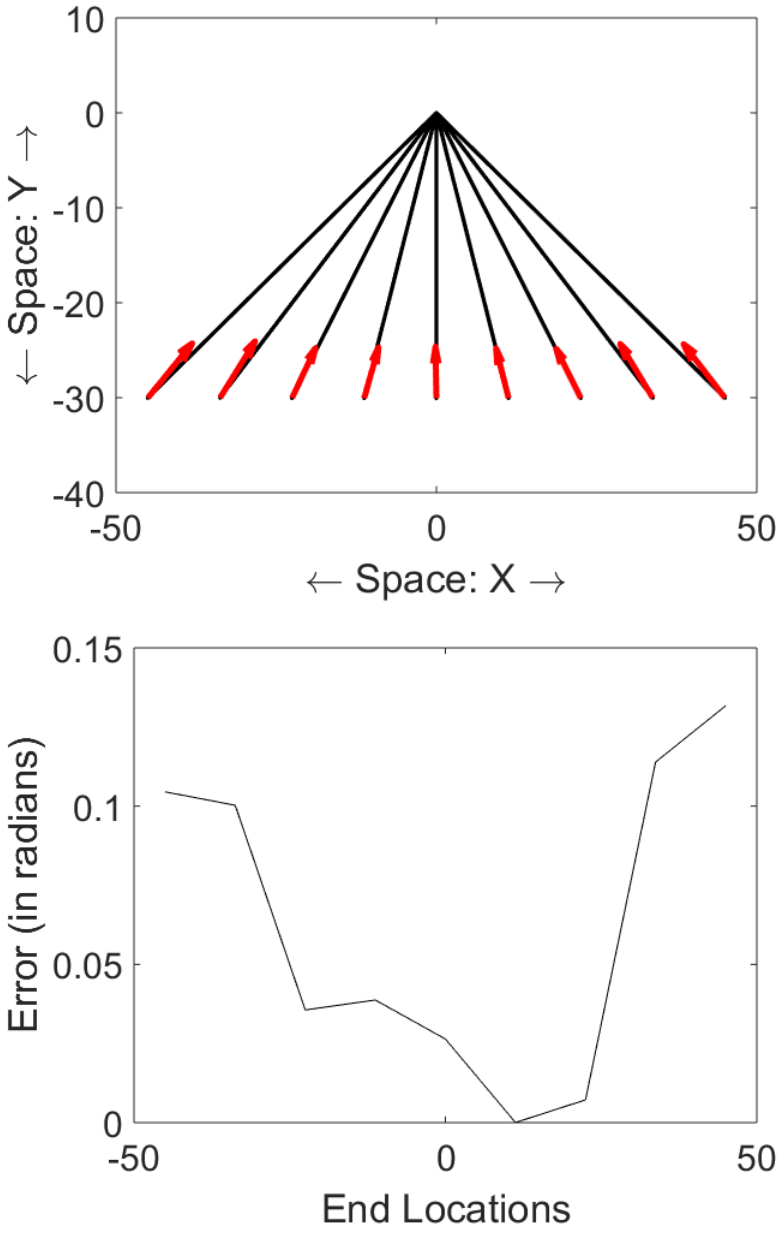


Fig. 3.9. In the top panel, finding the direction to the waypoint using grid cell population activity is depicted. The agent moves from the origin towards a location lying along a horizontal line (parallel to the x-axis). Nine different paths are shown. Once the agent reaches the location, the direction to the origin is computed using the grid cell activities at that location and the origin. In the bottom panel, the absolute decoding error is presented for different end locations of the agent.

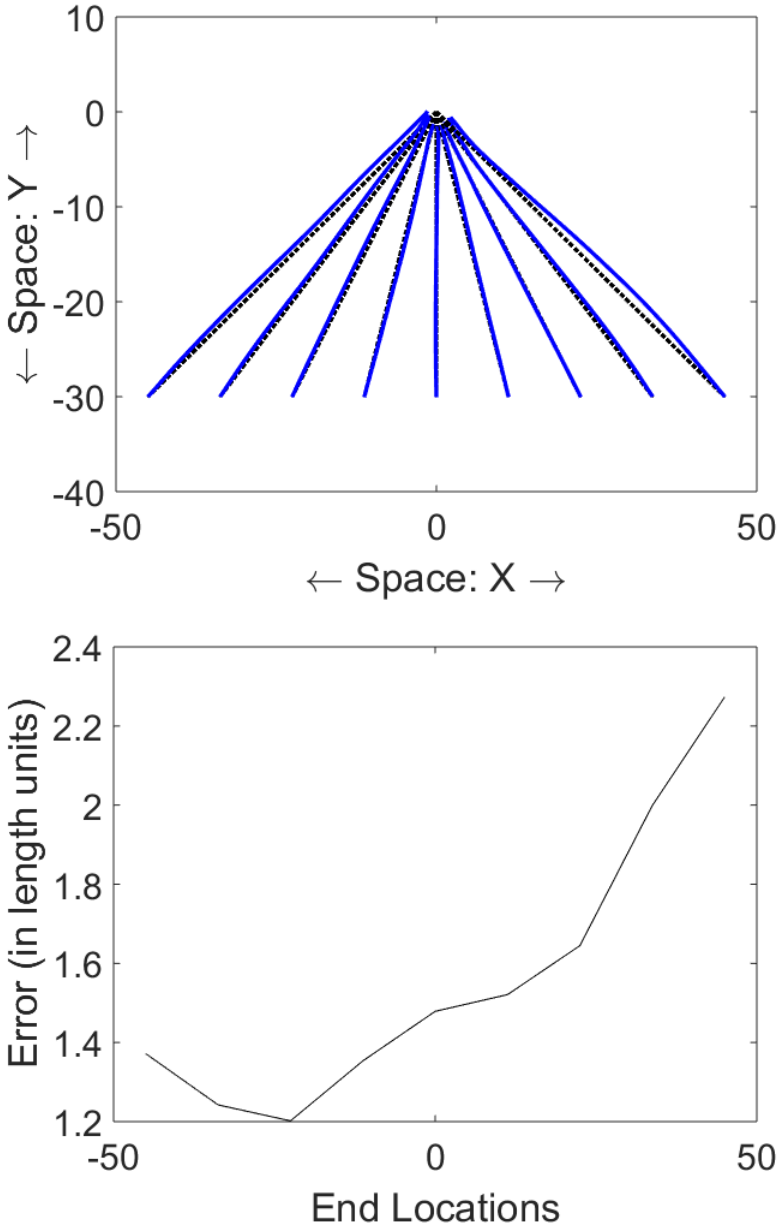


Fig. 3.10. The top panel shows paths that the agent takes back to the origin from different locations. Nine different paths are shown. The agent moves from the origin towards a location lying on a horizontal line and returns to the origin. The return path is shown in blue. The bottom panel shows the distance between the origin and the agent at the end of movement. The agent moves at a constant speed towards the origin. The movement angle is decoded based on the activities of the grid cell populations at the present location and the origin. The agent stops when the motor neuron activity falls below a threshold level.

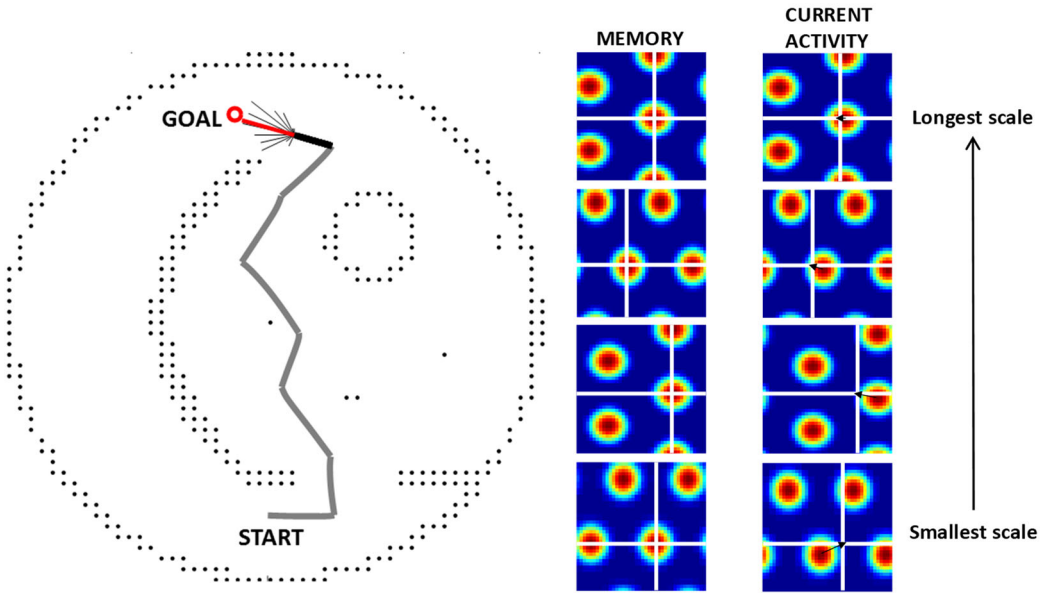


Fig. 3.11. Snapshot during the path execution process. The grey line segments represent the path just taken. The black line segment represents the path currently being executed. The whisker plot represents the possible directions the agent is using to move towards the goal. Not shown is the spreading-activation path planning process. The memorized and present-location grid-cells activity patterns that are compared to determine the direction for movement. These are based on [47]. The planning and execution cycle is repeated until the agent reaches the goal.

Although many different mechanisms could be utilized to accomplish the movement at this step (including a more sensory view approach), we have used this grid cell memory approach in this work.

3.7 Summary

In this chapter, previous modeling efforts towards hippocampally-inspired navigation systems were discussed, followed by a description of our system and a complete navigation simulation. The focus of this research was path planning, but to provide an example of the application of this model to navigation, the mapping and path execution steps were performed with a computer simulation. A spatial map was created by ensuring that every visited location had sufficient place cell activity. For

long-range navigation, it is necessary to store the spatial relationship between adjacent places and this was accomplished by synaptically linking place cells when the agent crosses the corresponding overlap of the place fields. For path execution, it is necessary to store information of how to move between two locations. This was accomplished using grid cell activity and a system that can compare the activity of two different grid cell populations and generate a direction that allows movement between the locations. Path planning was carried out by using a wave of activity spreading across the neurons. Each place cell had a WTA circuit that indicates the direction of the earliest arriving activity to the place cell. Path execution was carried out using a neural network (based on [47]) that can generate the direction of movement between the present location and a memorized waypoint by comparing their corresponding grid cell activities.

Chapter 4: Neuromorphic VLSI System

4.1 Introduction

Neuromorphic VLSI provides a low-power solution to implementing neural algorithms. Recently, there has been a growing interest in the use of spiking-neuron based systems for robotics applications [51], [52], targeting either small platforms [7] or for testing biologically-plausible neural networks [53], [54]. Currently, the popular approach is to use computers or graphics processors, however, in the long-term, implementing neuromorphic VLSI solutions to biologically-realistic, neuron-based systems would be more power and area-efficient.

In this chapter, the custom-designed, multi-neuron chip that was used in the path planning system is described and the operation of the synapse and neuron circuits is explained. The path planning system architecture with spiking neurons is presented here. It is based on the model discussed in the previous chapter, but the spiking neuron implementation of that system is presented here. Then, the system-level connection incorporating a microcontroller for interfacing with the multi-neuron chips is described. Results from characterization tests relevant to the use of these chips for path planning are also presented here.

4.2 Multi-Neuron Chip

The multi-neuron chip was fabricated using a commercially-available silicon foundry service in 0.5-μm technology with double polysilicon and triple metal layers. Each multi-neuron chip has sixteen integrate-and-fire neurons on it (see Fig. 4.1). The integrate-and-fire neuron circuit is similar to [55]. There are eleven individually-

addressable synapses per neuron (eight excitatory and three inhibitory) and one global inhibitory synapse that is used to inhibit all neurons simultaneously. There is a fully-arbitered AER encoder [56] on the chip that transmits neuron addresses for outgoing spikes. The standby power for a single chip is less than 1 nW and the operating power consumed by the 4 chips along with external biasing circuits is 7.5 mW. The newer version of the chip (also shown in Fig. 4.1) has thirty-two neurons and each neuron has fifteen individually-addressable synapses (nine excitatory and six inhibitory) and one global inhibitory synapse. The refractory period parameter on this chip has a longer range.

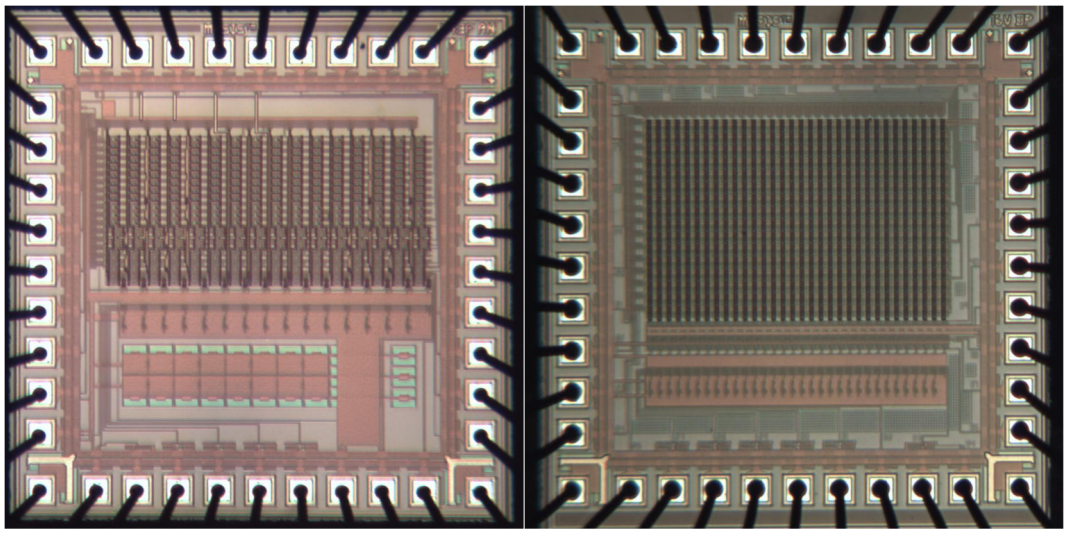


Fig. 4.1. Photomicrographs of multi-neuron chips. The first chip (left) has sixteen integrate-and-fire neurons in it. Each neuron has twelve synapses. There are eight excitatory, three inhibitory and one global inhibitory synapse. The second chip (right) has 32 integrate-and-fire neurons in it. Each neuron has sixteen synapses. There are nine excitatory, six inhibitory and one global inhibitory synapse. The area of both the chips is 2.25 mm^2. First figure taken from [2].

4.3 Synapse Circuit

The excitatory and inhibitory synapses are implemented with identical core circuits (shown in Fig. 4.2 (a)). The inhibitory synapse output is directly fed to the

neuron circuit to draw current *out of* the postsynaptic cell, whereas the excitatory synapse output is fed to a current mirror that reverses the current direction and injects current *into* the postsynaptic cell. The global inhibitory synapse circuit (shown in Fig. 4.2 (b)) is identical to the synapse circuit, but the transistor that selects the neuron to be inhibited is removed. Whenever the global inhibitory synapse is selected, it sends inhibitory currents into the entire neuron population.

The synaptic circuit is based on the Reset-and-Discharge synapse [57]. Every time the synapse receives a spike, it generates a current with a square pulse. If another spike is received after the first spike, then the spike pulse gets extended by a width equal to the delay between the input spikes. The node with the capacitor (the "pulse timer") gets reset with each incoming spike to the initial voltage. The output current of the transistor is controlled by using another transistor that is biased in the subthreshold region.

A polyI-polyII capacitor of 90 fF is used to generate a time constant of greater than 10 ms. The voltage that controls the timer length is called V_{DUR} and the voltage that controls the output current is called V_{SYNP}.

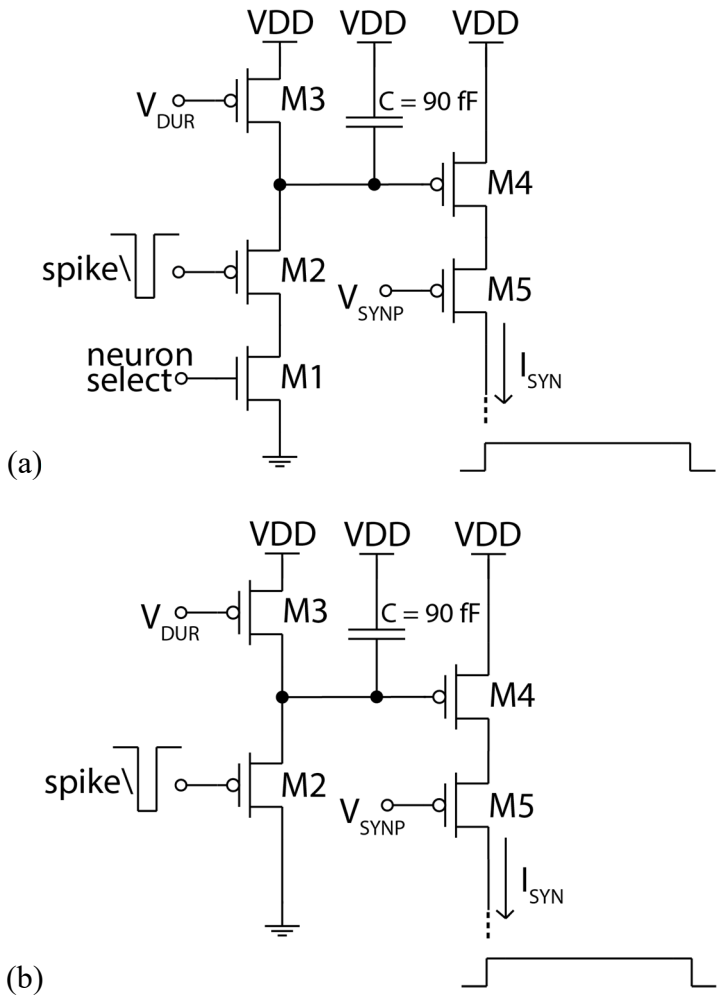


Fig. 4.2. Showing the synapse circuits. (a) Transistors M1 and M2 are used to select the neuron and synapse pair. M3, biased in the subthreshold provides current to recover the voltage on the capacitor. State of M4 controls the current width and M5 controls the current level. (b) Showing global inhibitory synapse, transistor M1 is removed so that a spike sent to global inhibitory synapse affects all neurons on the chip. Top figure taken from [2].

4.4 Neuron Circuit

The neuron circuit is shown in Fig. 4.3. A key feature of the path planning mechanism is the time delay between a spike arriving at a neuron and when the neuron fires a spike. To obtain well-controlled integration times and smaller jitter, an integrate-and-fire neuron model was used. A source-follower stage was added to the membrane

voltage to effectively increase the spiking threshold to get longer integration times without increased currents in the threshold inverter (M6 and M7). Transistors M1 and M2 collect the current from excitatory synapses and reverse the current direction. The inhibitory currents are directly fed into the node V_{MEM}. Transistors M3 provides the leak current that is set by V_{LEAK}. Transistors M4-M7 provide the threshold for the neuron. V_{SF} is set to get a longer integration time before the neuron spikes. V_{TH} is used to adjust the firing threshold. Transistors M8-M11 are a part of the axon hillock circuit (spike generator). The neuron sends out a request-to-send logic-level signal (REQ) to the address-event communications system (described in more detail below) on chip. On receiving an acknowledge signal (ACK) from the arbiter, the membrane potential is set to V_{RESET} and the transistor goes into a refractory period that can be adjusted by using V_{REFR}.

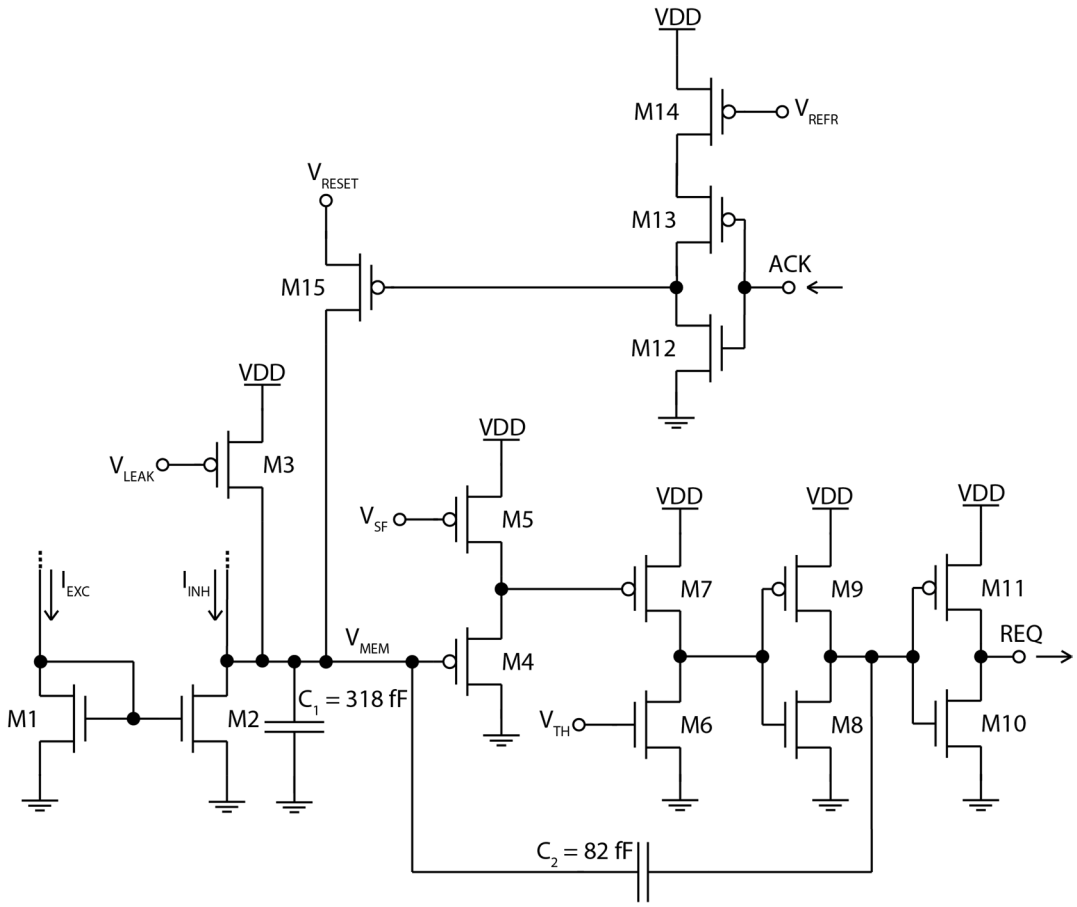


Fig. 4.3. Integrate-and-fire neuron circuit, where V_{TH}, V_{REFR} and V_{RESET} represent the firing threshold, the refractory period setting and the neuron reset potential. V_{LEAK} is used to adjust the leakage current and V_{SF} is used to adjust the current integration time. The width and length for all transistors are 3 μm. From [2].

4.5 Architecture of the Planning System

For path planning, spreading-activation is implemented as the propagation of a spike-wave emanating away from the goal-location neuron and eventually reaching the present-location neuron. The spreading-activation planning process and the neural architecture used to implement it are described using a small example in Fig. 4.4. To determine the "shortest path" through the map from the goal to the present-location place cell, the network must determine the direction (within the graph) of the first arriving spike (a.k.a., "temporal winner-take-all", see [58], [59], [41]). This will be

done by the addition of "direction neurons" associated with each link to a neighboring place cell. Each direction neuron also receives a synapse from the neighboring place cell and the direction neurons, in turn, send a synapse to a local inhibitory neuron that sends inhibitory synapses back to them. The direction neurons and the local inhibitory neuron form a temporal winner-take-all (WTA) network local to each place cell. The planning process is started by exciting the place cell that represents the goal location. When the goal location place cell fires, spikes spread from node to node throughout the whole network. As activity spreads through the network, the direction neurons and their respective local inhibitory neurons fire, implementing the WTA at each node. For each place cell, the direction neuron that fires represents the next place cell the agent should move to along the shortest path. Although the entirety of the shortest path can be found by following the winning direction cells from the present-location to the goal-location place cell, in this research, only the next movement is needed. By only attending to the direction neurons local to the present-location cell, only the next move is determined. After the agent has moved to the next place along the shortest path, attention is shifted to the direction cells at the new present-location place cell and a new path planning cycle is initiated. Repeating this process, the agent will travel from place to place along the shortest path to the goal.

Instead of creating a temporal WTA circuit at every place cell, the whole temporal WTA network could be replaced by a single temporal WTA that is dynamically configured at the present-location place cell before every planning step. By using the WTA only at the present location place cell, the number of directions neurons used overall is reduced.

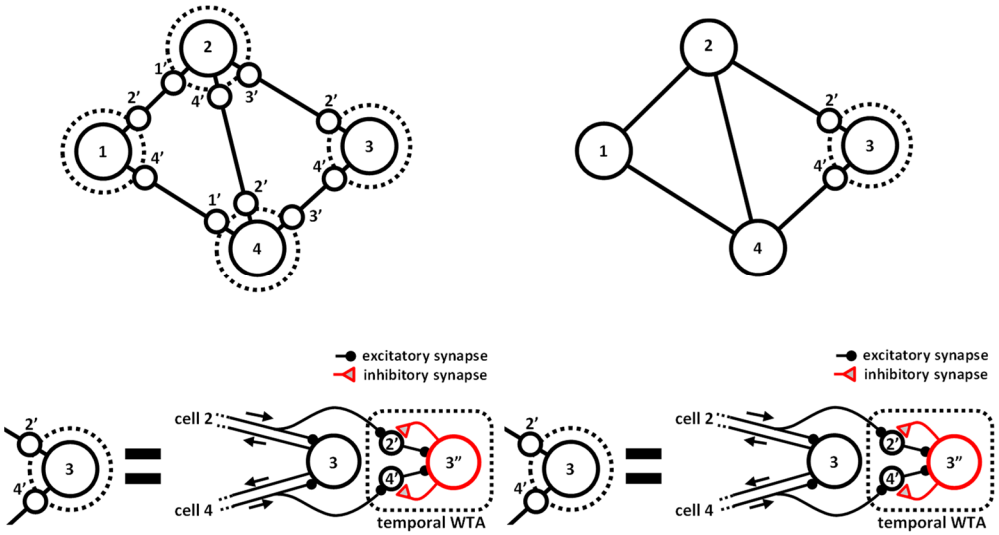
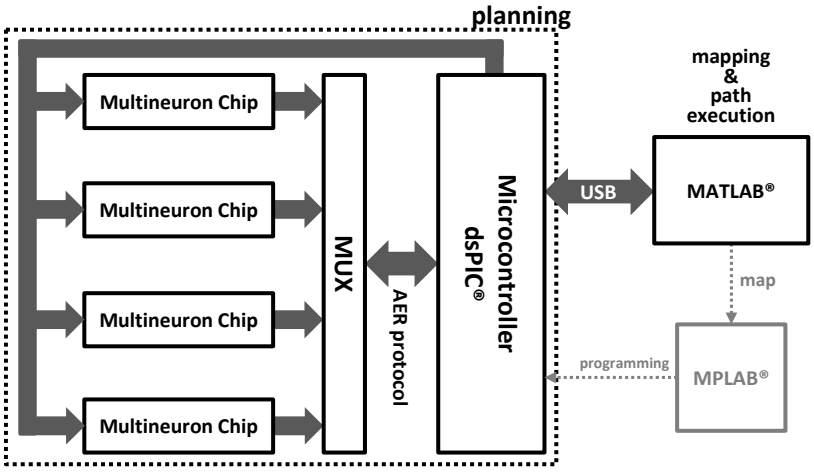


Fig. 4.4. The neural architecture is introduced using an example map with 4 locations and 5 paths. There are four place cells labeled 1-4. In the left panel, every place cell has direction neurons that correspond to inputs the place cell receives from other place cells. The connection details for place cell 3 are provided below. Place cells 2 and 4 make synaptic connections with place cell 3. They also send synaptic connections to direction neurons 2' and 4' respectively. A local inhibitory neuron, labeled 3", receives excitatory inputs from the direction neurons 2' and 4' and sends inhibitory synapses back to them forming a temporal WTA. So, if place cell 2 spikes, it would also send spikes to the direction neurons (2') of place cells 1, 3 and 4. In the temporal WTA of place cell 3, the local inhibitory neuron (3") would receive a spike from the direction neuron 2', fire a spike and inhibit the other direction neuron (4') from firing. In the right panel, the dotted circle is only around place cell 3 indicating that the agent is located at the place represented by place cell 3. As in the left panel, the inhibitory neuron receives inputs from direction neurons 2' and 4'. The WTA connections are reconfigured after the agent moves to the place corresponding to place cell 4. So, the WTA receives inputs from place cells 1, 2 and 3. From [2].

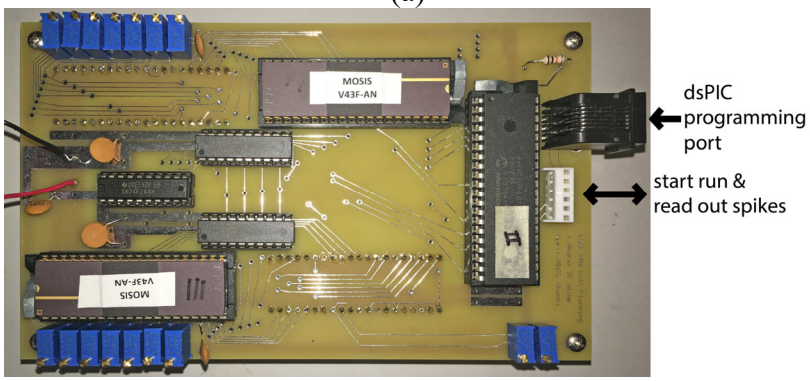
4.6 System Connections

The neuromorphic VLSI system (see Fig. 4.5) consists of four custom multi-neuron chips and one dsPIC® microcontroller (Microchip®). The multi-neuron chips communicate with the microcontroller using the Address Event Representation (AER) protocol [56] that transmits the unique address of each neuron at the time of a spike.

The communication between the microcontroller and a desktop PC uses a serial communications protocol. The map, which is generated from the MATLAB® simulation, is programmed on the dsPIC using the MPLAB® software suite provided by Microchip Technology Inc. In this test configuration, the planning is initiated from the computer and the results are stored on the microcontroller and read out after each run is complete.



(a)



(b)

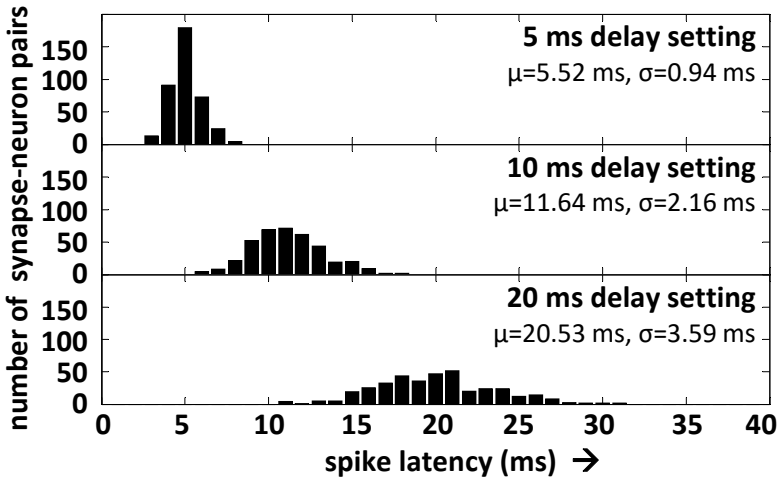
Fig. 4.5. Neuromorphic VLSI system diagram. (a) Four custom multi-neuron chips are connected to a dsPIC® microcontroller. The microcontroller handles routing between the chips using the AER protocol. The network (or map) is stored on the microcontroller. A computer is used to configure the network, start a planning run and read out neuron spikes. The spikes generated are recorded on the microcontroller and read out after the run has completed. (b) Picture of the PCB. Two neuron chips are mounted on the bottom of the board. Figure (a) from [2].

4.7 Fabrication Mismatch

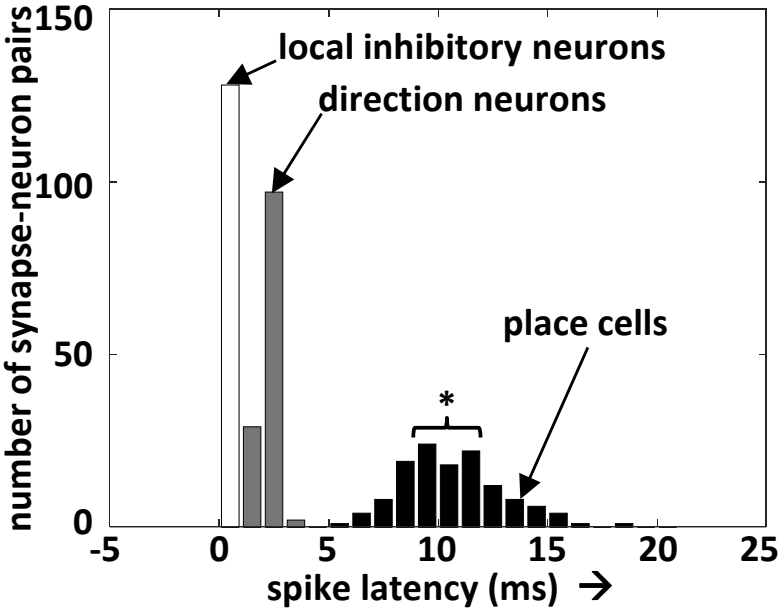
The neuron and synapse parameters are chosen such that a single incoming spike will make the postsynaptic neuron fire after a fixed delay. For fixed delay settings (using synapse-neuron pairs from three chips), we plotted a distribution of spike latencies (384 synapse-neuron pairs, see Fig. 4.6 (a)). The deviation due to fabrication mismatch is roughly 17.7 % of the mean spike latency value. To maintain consistency in the propagation delay for each link in the graph, a subset of the available synapses with more consistent propagation delays were selected for each neuron. There are different types of neurons in the network: place cells, direction neurons and a local inhibitory neuron. The spike latencies associated with the different neuron types are shown in Fig. 4.6 (b). The relationships between delay settings are important for the planning algorithm to work. Parameter values for the delay settings are provided in Table 4.1. Out of the four multi-neuron chips, one is configured to carry direction neurons. The remaining chips could be configured to carry place cells and local inhibitory neurons, because there are two sets of excitatory synapses with independent bias pins to select their parameters.

For implementing the planning graph on the chip, the map generated by the mapping process is constructed taking into consideration the spike latency data. Enough number of neurons that have the minimum variation around a given spike latency are selected for planning. After the external voltage parameters are fixed to set the mean delay for the synapse-neuron latency, a sweep is performed for different latency settings, while looking for the synapses that meet the criteria. After enough

number of synapses are found, the neurons are picked for representing the cognitive graph.



(a)



(b)

Fig. 4.6. Selecting the neurons for planning. (a) Distribution of neuron spike latencies for different delay settings, showing the relationship between mean spike latency and variation across the chips. The deviation is roughly 17.7% of the mean value and is due to device mismatch. (b) Histogram of measured spike latency setting for place cells, direction neurons and local inhibitory neurons taken from a single chip, i.e. 8x16 synapse-neuron pairs, where all neurons are configured as place cells, direction neurons or local inhibitory neurons. This is a combined plot of their latencies. Place cells selected for path planning are marked in the figure with an asterisk. From [2].

Table 4.1. Bias voltages for the multi-neuron chip. From [2].

	PINS	Voltage (V)
Shared Neuron Parameters		
	V_{TH}	0.61
	V_{RESET}	4.80
	V_{REFR}	4.41
	V_{SF}	3.50
	V_{LEAK}	4.35
Synapse Parameters for Place Cell		
Excitatory synapse	V_{SYNP}	4.26
	V_{DUR}	4.36
Synapse Parameters for Direction Neuron		
Excitatory synapse	V_{SYNP}	4.21
	V_{DUR}	4.36
Global inhibitory synapse	V_{SYNP}	4.18
	V_{DUR}†	4.36
Synapse Parameters for Local Inhibitory Neuron		
Excitatory synapse	V_{SYNP}	4.13
	V_{DUR}	4.31

VDD = 5 V

†V_{DUR} for excitatory synapse and global inhibitory synapse is a common pin.

4.8 Spike Jitter

Jitter (standard deviation of the spike latency) was estimated by repeatedly stimulating a neuron and measuring the spike latency, providing sufficient time between spikes, to ensure that the neuron was well out of its refractory period. The neurons were of different types – place cells, direction neurons and local inhibitory neurons. Data was collected from 16 neurons of each type using the first excitatory synapse for stimulation. The jitter values are based on 100 spike latency measurements. The data is shown in Fig. 4.7, where the standard deviation is plotted against the mean spike latency. It can be observed that the jitter value is proportional to the mean spike latency. Jitter divided by mean spike latency is similar among neurons. After this normalization step, the average is computed. The jitter was found to be 0.075% of the mean spike latency.

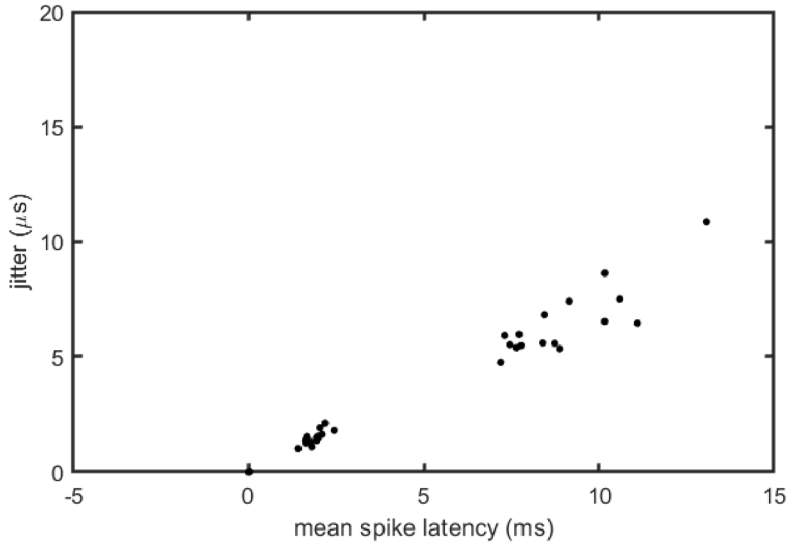


Fig. 4.7. Showing spike jitter for neurons versus their spike latencies. Jitter is proportional to the mean spike latency. So, smaller currents have more jitter. From [2].

4.9 Summary

In this chapter, the operation of the neuromorphic VLSI chip circuits was explained. The path planning system architecture and the system implementation was described. There are three types of neurons used in the neural model, but all are implemented using the same synapse and neuron circuits, but with different parameter settings. The topological graph connectivity was stored on the microcontroller and the neuron-to-neuron link latency was assumed to be constant. Given fabrication mismatch that introduces latency variability, synapses of each place cell that elicit a spike within a certain spike latency margin were selected. This step was carried out after observing the distribution of spike latencies for different delay settings. Spike jitter was also evaluated to test its impact on the operation. Since the jitter was found to be small, it was not considered for further analysis. The impact of fabrication mismatch is discussed further in Appendix-A.

Chapter 5: Path Planning by Spike Propagation

5.1 Introduction

A topological representation allows an agent to exhibit flexible behavior when faced with an obstacle, i.e. find an alternate path if it exists in its memory. Such a behavior depends on finding a path to a location, which is not in its line of sight or directly accessible. This capability can be modeled by path planning, which is the process of determining a trajectory between the present location and the goal. In this research, we use the term path planning to refer to waypoint path-planning that simplifies the problem of trajectory determination by finding the sequence of intermediate waypoints between the present location and the goal, reducing it to a symbolic problem. This simplification assumes that there is additional information associated with the waypoints, so that the direction of movement can be computed from it.

Path planning can be accomplished by using graph search algorithms if the navigation space can be represented as a graph [60]. Graph based algorithms include Dijkstra's algorithm, A*, D* [61]. Methods based on Dynamic Programming have also been used to find least cost paths and in robotics, wavefront planning and potential field-based methods are popular [62].

Neuromorphic systems provide a low-power platform for implementation of neural-network based algorithms. Neural-network models for path planning include both firing-rate (analog) [25], [63], [64] and spike-latency [11], [65] based signal representations, but recent hardware-based implementations [11], [14], [66] have

focused on using propagating wavefronts of spikes for planning. The implementation in this research is in line with the aforementioned work. In addition, we assume that the map is not a dense representation of the environment. From Chapter 3, we recall that we assume that animals represent the space in the form of a cognitive graph and the links joining the graph nodes are of equal weight. In such a system, the spike latency between any two synaptically-connected neurons is kept the same. Additionally, a temporal winner-take-all network is used to determine the direction for movement. The temporal winner-take-all consists of direction neurons that receive inputs from place cells and a local inhibitory neuron.

5.2 Operation of the Temporal WTA Circuit

The temporal WTA is dynamically associated with the present-location place cell and is configured such that the first direction neuron to fire will produce feedback inhibition to prevent any of the other direction neurons to fire. Since each direction neuron is associated with a direction of travel (to its associated neighboring place field), the direction neuron that successfully fires will determine the direction of movement to the next waypoint. The operation of the temporal WTA network is shown in Fig. 5.1 using a simple example. The graph used is shown inset. Place cells and their spikes (black dots) are numbered identically. The goal-location is associated with place cell 0 and the agent is located at the place represented by place cell 4. The local inhibitory neuron receives inputs from direction neurons 1', 2' and 3', corresponding to place cells 1, 2 and 3. The dotted circle around place cell 4 is used in the figure to indicate the implementation of the temporal WTA at the present-location place cell. The local

inhibitory neuron is labeled as 4". As the agent's position changes, a new WTA circuit is reconfigured to operate at the "present location" of the agent.

Path planning begins with the stimulation of place cell 0 (i.e., goal). After a delay, place cell 0 fires, sending spikes to place cells 1, 2 and 3. After another delay, neurons 1, 2 and 3 send spikes to place cell 4 *and* the direction neurons 1', 2' and 3' (associated with the temporal WTA at place cell 4). Without the local inhibitory neuron, all the direction neurons would fire, as shown in grey. With the local inhibitory neuron present, however, the first direction neuron to spike (1' in this case), triggers a quick, strong, feedback inhibition preventing all other direction neurons from firing. Through this mechanism, only one "action" is chosen to move from the present-location to the next place along the shortest path. Although the paths in the graph are identical in theory, the spike latencies of these three equivalent paths are slightly different due to fabrication mismatch of the transistors in the circuit, producing a random (but fixed) bias associated with each place cell-direction neuron link. The decision of exactly which neuron circuits are connected to each other occurs during the mapping process and could, in theory, be chosen with these biases in mind. In practice, however, a small random variability can be desirable to break symmetry and facilitate decision making.

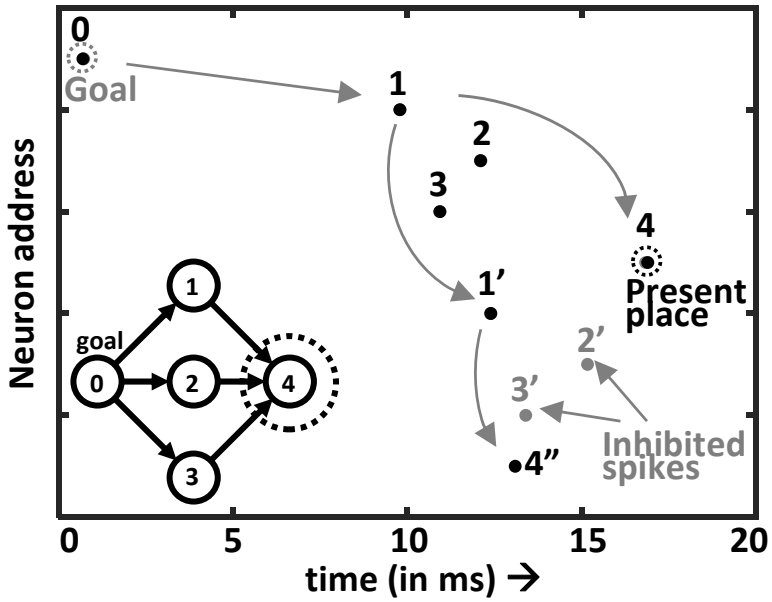


Fig. 5.1. Operation of the WTA network used for picking the next movement direction for the agent. The goal is at place cell 0 and the agent is at place cell 4, demarcated by a dotted circle in the inset graph. The decoding network consists of direction neurons and a local inhibitory neuron (not shown in the inset graph). The spikes of the direction neurons and the local inhibitory neuron are labeled 1', 2', 3' and 4". Which direction neuron fires depends upon spike latency of the direction neurons. When the local inhibitory neuron is not operating, all the direction neurons spike. When the local inhibitory neuron is operating, only the direction for the movement is picked, i.e. 1' spikes, but 2' and 3' get inhibited. Here 1', 2' and 3' provide equivalent directions. From [2].

5.3 Selecting the Shortest Path

Rats can select the shortest path from multiple paths in a maze after training [67]. This has been replicated in a computational model [35] based on a topological representation of the maze. To demonstrate this, a three-path maze was created with unequal path lengths and the place fields were uniformly distributed, so, the number of place cells that represent a path is proportional to the path length and this was tested on our neuromorphic VLSI system.

Selection of the shortest path from a selection of multiple paths of different lengths is demonstrated in Fig. 5.2. In this example, the goal is again associated with place cell 0 and the agent is at a place represented by place cell D. In the temporal WTA network, the local inhibitory neuron receives inputs from direction neurons C', 8' and B'. Planning begins with the stimulation of place cell 0. Place cell 0 spikes and stimulates place cells 1, 2 and 3. These further stimulate 4, 5 and 6. When the wave of spike activity reaches 7, 8 and 9, the direction neuron 8' is stimulated. When 8' fires, it causes the local inhibitory neuron D" to fire, preventing neurons C' and B' from firing. Because neuron 8' fired, the agent will move to the location represented by place cell 8. After a successful movement, place cell 8 becomes the new present-location place cell. A new cycle of planning is initiated, and the agent moves to the next waypoint (i.e. place 5). The planning cycles stop when the agent finally reaches the place represented by place cell 0.

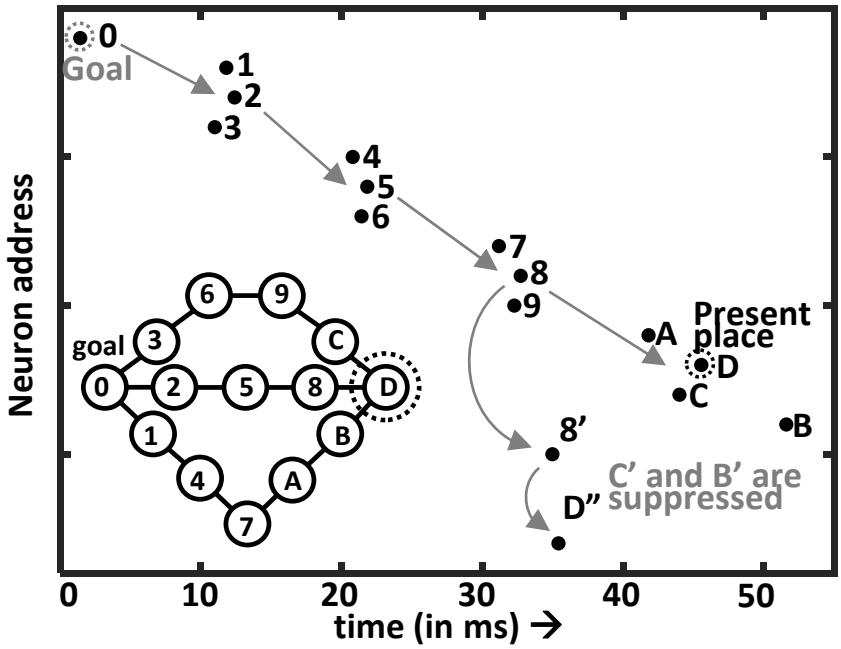


Fig. 5.2. Selection of the shortest path among multiple paths. The goal is represented by place cell 0 and the present location of the agent is represented by place cell D. The direction neurons are C', 8' and B'. The local inhibitory neuron is D". Stimulation of the goal place cell causes spikes to spread through the network, stimulating 1-3, 4-6, 7-9 place cells in succession. Activation of place cell 8 causes direction neuron 8' to spike and activation of D" causes inhibition of other direction neurons, C' and B'. This is the first step in the plan. The action is to go to place cell 8, which lies on the shortest path. From [2].

5.4 Selecting the Nearest of Multiple Goals

The same system that solves the task of selecting the shortest path to a single goal can be used to select the path to the closest of multiple goals. A system with spiking neurons can be used to solve such a problem if all the place cells that represent goal locations are stimulated at the same time (for examples, see [11], [68]). A simple example is described below, where the agent must pick the closer goal out of the two goals located on its sides.

Selection of the nearest goal, when multiple goals are present is demonstrated in Fig. 5.3. There are two goals in the example, and these are associated with place cells

1 and 7. The agent is at a location represented by place cell 5. In the temporal WTA network, the local inhibitory neuron receives inputs from direction neurons 4’ and 6’. Planning begins with the stimulation of both place cells 1 and 7. In this case, there are two “waves” propagating outward from both place cells. When the wave from the right reaches place cell 6, the direction neuron 6’ is stimulated to fire, which stimulates the WTA inhibitory cell 5" that inhibits neuron 4’. Selecting the direction represented by place cell 6 is on the path to the closer goal. The planning stops, when the agent reaches the goal represented by place cell 7. Therefore, the agent is able to select the path to the closer goal.

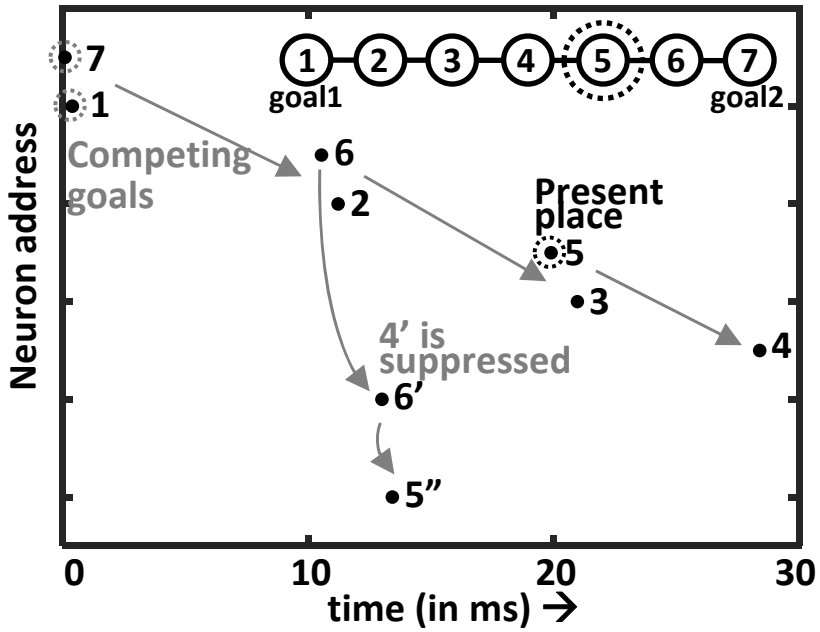


Fig. 5.3. Selection of the nearest goal among multiple goals. There are multiple goals. The goals are represented by place cells 1 and 7. The present location of the agent is represented by place cell 5. The direction neurons are 4’ and 6’. The local inhibitory neuron is 5”. Stimulation of the goal place cells causes spikes to spread through the network, stimulating 2 and 6. Activation of place cell 6 causes direction neuron 6’ to spike and activation of 5” causes inhibition of other direction neuron 4’. This is the first step in the plan. The action is to go to place cell 6, which lies on the path to the closer goal. From [2].

5.5 Demonstration with a Robotic Arm

Multi-joint robotic-arm control in presence of obstacles is a popular motion planning problem in robotics [62]. We can apply the path planning formalism to finding sequences of arm movements that lead from a starting configuration to a goal configuration. This can be done if the place cells represent a state vector of joint angles and paths represent sequences of poses. Following is an example that was used to test the planning system on such a task. In this, we use a robotic arm where the arm tip is moved to different goal locations amidst obstacles (Fig. 5.4). Four obstacles are placed radially. In this problem, the arm can take multiple paths to reach the same goal, especially when no straight unobstructed path exists between the current place and the goal.

A graph is constructed by placing waypoints to distinguish between different routes. Each waypoint represents a joint configuration (i.e. a set of four joint angles) of the arm and thus a specific point in 2-D space for the tip. Moving the arm (shown in Fig. 5.4) is carried out by repeatedly stimulating the goal node and taking action based on the node selected by the winner-take-all. For this problem, the nodes in the graph are defined by joint angles, so moving from node to node is accomplished by simply by updating the joint angle.

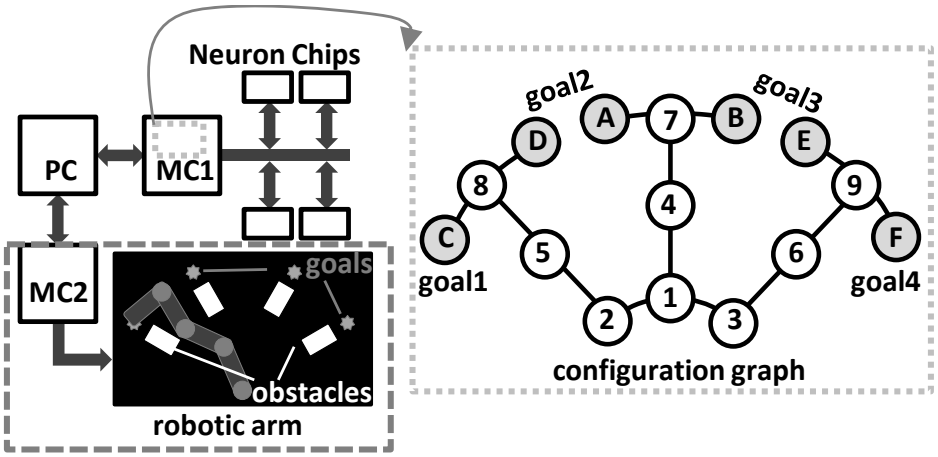


Fig. 5.4. System description and path planning on a 4-joint robot arm. The graph of allowed joint configurations is stored on the microcontroller, which also handles the communication between four neuron chips. Two chips are used to represent the map, one chip is used for the winner-take-all neurons and one chip carries the global inhibitory neuron. A path is found by propagating spikes from node C (goal) to node F (starting location). Winner-take-all neurons monitoring the current location node indicate the next movement. Following each arm movement, the planning is repeated until the goal is reached. The arm movement is carried out by using four servos that are controlled by a Micro Maestro 6-Channel USB Servo Controller from Pololu® that is connected to the PC. From [69].

The waypoints were closely spaced to control the trajectory enough to avoid colliding with any obstacles. We note that in this problem these waypoints were hand-selected. Like the navigation problem, the graph represents previously acquired knowledge and path planning only operates on this stored knowledge.

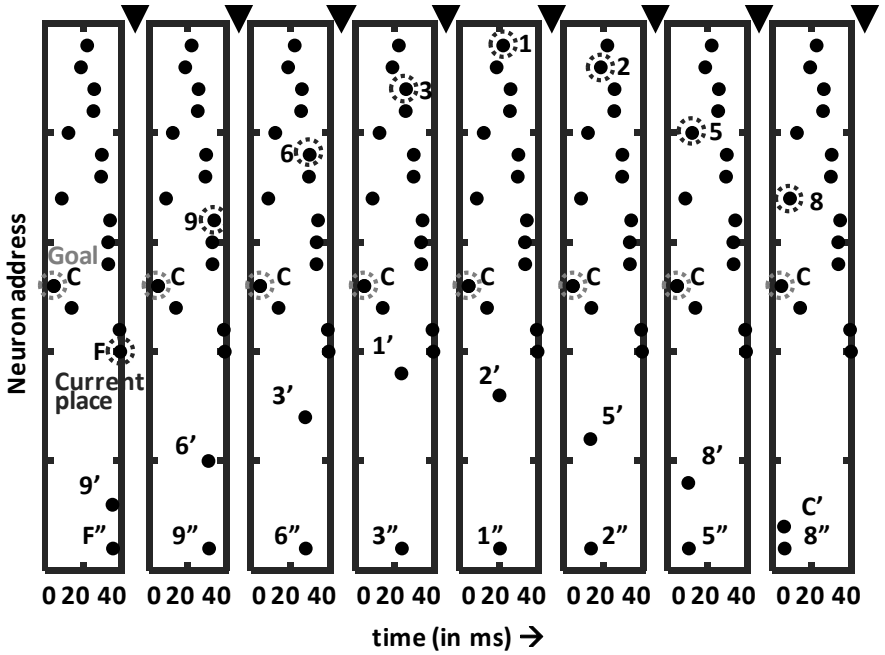


Fig. 5.5. An example of planning on a robotic arm configuration graph (shown in Fig. 5.4), showing a plan to move from state F to state C (goal 1). The direction neurons (9' to C') describe the action plan at a glance and the global inhibitory neuron labeling (F" to 8") shows the current place. The solid triangles at the top and in between the plots show when the action is carried out. Modified from [69].

5.6 Summary

In this chapter, the operation of the neuromorphic system for solving the path planning problem was explained by describing its operation on two example problems: selecting paths with the fewest number of hops and choosing the nearest goal from multiple goals. The operation of temporal winner-take-all circuit, which is a part of the system, was described to explain the selection of the next action and to highlight the issue of mismatch related difference in spike latencies on equivalent paths. To demonstrate the operation of the system on a path planning problem in another task-domain, a multi-joint robotic-arm control problem was described.

Chapter 6: Effect of Synaptic-Dependent Spike-Latency

6.1 Introduction

The response of a biological neuron to incoming spikes depends on various factors like the shape and structure of the neuron, type and distribution of ion channels on its membrane and neurotransmitters released at its synapses. Analyzing how such properties affect the computation performed at the network level is essential to understanding their role in the overall computation.

In Chapter 3, a model for the path finding process inspired by the hippocampal place cells found in the rodent brain was presented. The assumption was that a representation of the environment is captured in the rodent brain in the form of a cognitive graph and that spikes initiated at the goal location place cell spreading through the network enable parallel exploration of paths in the mental representation of the space. Spikes traveling over the network arriving at the present location place cell accumulate latency, and based on the shortest latency, the action to move towards the desired goal is selected. If the response of the neurons in this network to the spikes arriving at their synapses is varied based on a property of the neuron, such as the number of synaptic connections, then the accumulated latency could change and thus affect the action that is selected. In this chapter, the effect of synaptic-dependent spike-latency is presented.

The total time for a neuron to generate a spike (observed at its axon terminals) when the neuron is stimulated by multiple incoming spikes, depends on the synaptic response, the manner in which the membrane potential rises in response to the synaptic

current and on the axonal delay. In the model described in this chapter, the focus is on the membrane voltage response to the incoming spikes. In the previous chapter, in the place cell model used, the total synaptic current to the neuron remains fixed and thus a change in the number and timing of spikes does not alter the neuron spike latency. In this chapter, a more biologically-plausible model is adopted, in which the spike latency drops with the increase in the number of spikes.

In the previous chapter, the response of the network of place cells viewed as a path planning system can be interpreted as finding the path with the shortest number of hops in a graph. For the model in which the spike latency is dependent on the number of synaptic connections that receive a spike before the place cell fires, the response is different. To analyze this, a method is introduced, and examples are provided that help in interpreting the response. Finally, a complete path planning example is provided on a graph created from mapping simulation described in Chapter 3.

6.2 Current Saturation and Current Summation Models

In this system, the shortest path is determined based on shortest propagation time from goal-location to present-location *in the graph*. If the propagation time from node to node is constant and if linked places represented by the place cells are separated by a fixed distance, the shortest distance path can be found. It should be noted, however, that the shortest physical path may not always be the goal of path planning. Other possibilities include: shortest time, lowest "risk", shortest description length, etc. In this work, we will focus on how spike latency (or link propagation time) can be affected by properties of the graph arising out of the biologically-inspired implementation. This is referred to as synaptic-dependent spike latency. Due to the charging time of the

membrane capacitance, biological neurons respond faster with stronger excitation. In this model, when spikes propagating along different paths in the map arrive at a node at the same time, the postsynaptic latency is reduced. To highlight this important property, two neuron models (see Fig. 6.1) are introduced. Recall that the synapses in this model generate a square pulse-shaped current. The first model is called the current saturation model, where no matter how many spikes a neuron receives, its input current saturates at a fixed value. In this case, its spiking latency does not change with the number of spikes. The second model is called the current summation model, where all the synaptic currents generated by the synapses are added and the spike latency depends on the number of inputs. A larger number of spikes within a given time window produces a shorter spiking latency (shown in Fig. 6.2).

In the examples presented in the previous chapter, the spike latency was expected to be constant, making this path planning scheme similar to the wavefront algorithm [62]. In a typical biological neuron, however, converging input spikes that arrive synchronously will produce shorter latency responses. Two examples are described that highlight the effects of this fact.

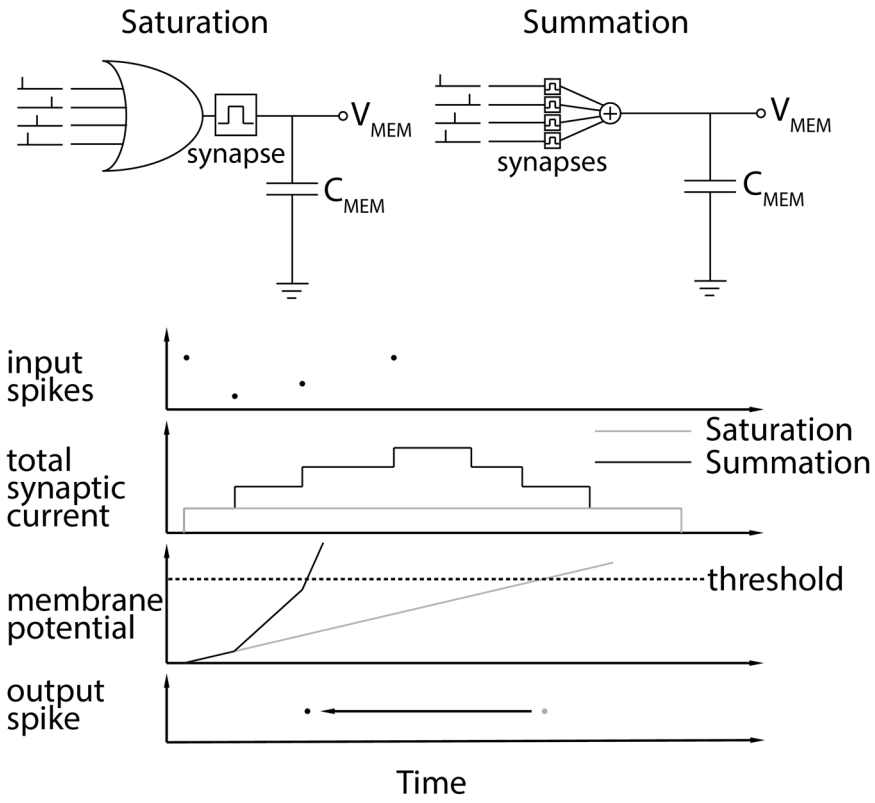


Fig. 6.1. Explains how spike latency in a biological neuron is dependent on the number of incoming spikes. The current "saturation" model limits the maximum current that flows into the neuron membrane capacitance. The current "summation" model adds the currents that flow into the neuron membrane capacitance. As a result, a neuron with current saturation model fires a spike with a fixed latency irrespective of the number of received spikes and a neuron with current summation model fires a spike with reduced latency that depends on time and number of spikes arriving in a fixed window. From [2].

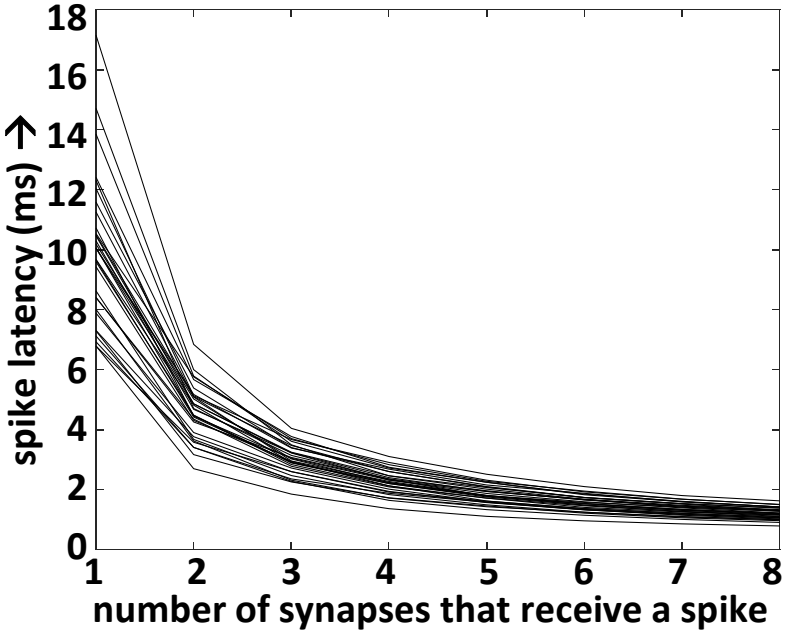


Fig. 6.2. Summation of synaptic currents results in a shorter spike latency. Showing the response from 32 neurons to multiple input spikes arriving at the same time. From [2].

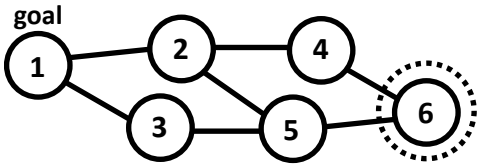
6.3 Selecting Path with More Redundancy

The first example is a network with multiple equivalent paths. Fig. 6.3 (a) depicts the graph. The goal is associated with place cell 1 and the agent is located at a place represented by place cell 6. In the temporal WTA network, the local inhibitory neuron receives inputs from direction neurons 4' and 5'. The planning begins with the stimulation of place cell 1. The spiking activity travels to place cells 2 and 3, followed by place cells 4 and 5 and then finally place cell 6. All three paths are equivalent in length, but place cell 4 only receives a spike from place cell 2, whereas place cell 5 receives spikes from place cells 2 and 3. Therefore, if the synaptic current saturation model is used, the spike latency of 4 and 5 would be similar (shown in Fig. 6.3 (b)) and the agent could choose either path (i.e., place cell 4 or 5). In the current summation

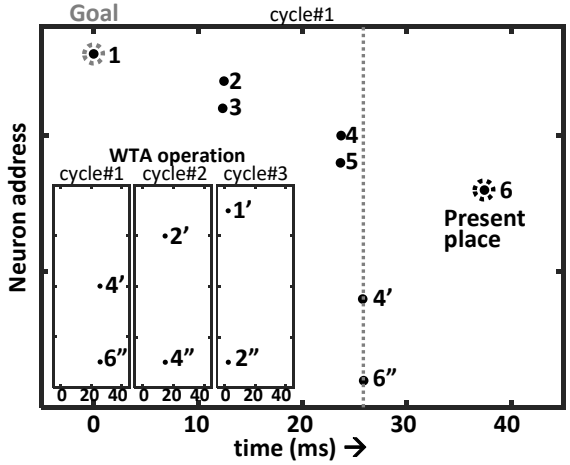
model, however, the spike latency of place cell 5 is shorter because it receives spikes from both 2 and 3 (shown in Fig 6.3 (c)) and thus direction neuron 5' fires earlier and the local inhibitory neuron 6" inhibits neuron 4'. Over the following planning steps, the agent moves from place 6 to 5; from 5 it could select 2 or 3 (3 is selected in this example) and finally reach the goal.

The advantage in selecting the route through place 5 is that it provides more alternatives to the goal than place 4. If one of the paths after place 5 were to experience a blockage, the agent could pick the other without an increase in the overall path length. Paths 6→4→2→1 and 6→5→3→1 have the same path length. The availability of more alternatives at place 5, however, has an impact on the average path length when there is an obstruction. This idea is evaluated in the following manner. Two scenarios are compared. First, all the possible paths that could be selected by the path planning process are found. In this particular example, planning using the current saturation model would yield 3 equivalent paths and the current summation model would yield only 2 paths. One path is selected by the agent for execution and the agent moves from the starting location to the goal location. An average path length is computed by averaging all the path lengths from each of the paths that could be selected by the agent. In the second scenario, one obstruction is placed on any one of the links in the selected path before the agent starts executing the plan. When the agent encounters the obstruction, it must replan, given the new map, and then complete the execution to reach the goal. This process is repeated for all possible obstructions on a path and for all the paths that each planning process could yield. The average path length is once again computed by averaging the path lengths of all the paths taken by the agent. The

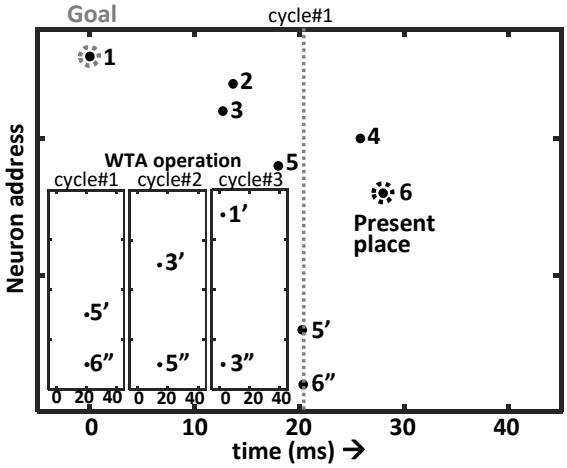
average path length for the current summation model comes out to be smaller than the current saturation model. This is depicted in Fig. 6.4. This could be interpreted as selecting a path with more redundancy.



(a)



(b)



(c)

Fig. 6.3. An example demonstrating how current summation property influences the path selection for execution in a graph with multiple equivalent solutions. In (a), the goal is represented by place cell 1 and the current location of the agent is represented by place cell 6. In first step of the plan, the direction neurons are 4' and 5' (local inhibitory neuron is not shown in the figure). (b) and (c) depict path planning in current saturation and summation models respectively. Stimulation of the goal place cell, 1, causes place cells 2 and 3 to fire, followed by place cells 4 and 5. In (c), since 5 receives spikes from 2 and 3, it fires earlier than 4. Direction neuron 5' fires and 6" inhibits 4'. The inset figures in (b) and (c) depict the operation of the temporal WTA and the selection of the path for execution. In (b), the path selected is 6→4→2→1 and in (c), the path selected is 6→5→3→1. Going to place cell 5 is better from the perspective of redundancy. From [2].

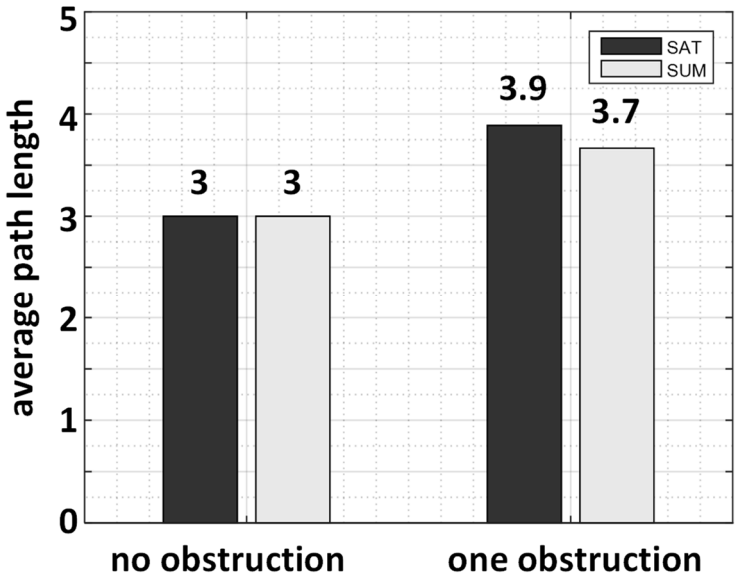


Fig. 6.4 Showing a comparison between current saturation and current summation models using average path length for the 6-node graph in Fig. 6.3. For the no obstruction run, the average path length is same for both the models. For the second run, with a single obstruction, the current summation model has a lower average path length. From [70].

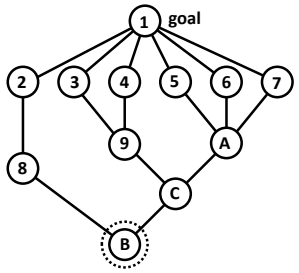
6.4 Selecting a Longer, but Risk-Averse Path

The second example considers a network with one short path and multiple longer paths, but these longer paths offer more alternatives (Fig. 6.5). The goal is associated with place cell 1 and the agent is located at a place represented by place cell B. Planning starts with the stimulation of place cell 1. For the synaptic current saturation model, the direction neuron 8' spikes earlier and hence direction neuron C' gets inhibited. The agent selects the shortest path B→8→2→1 (shown in Fig. 6.5 (b)). For the synaptic current summation model, due to multiple convergences of links at place cells 9, A and C, the direction neuron C' fires earlier and direction neuron 8' gets inhibited. In this case, the agent could pick paths B→C→A→5→1 or B→C→A→6→1 or B→C→A→7→1 (shown in Fig. 6.5 (c)). The choice of current saturation or current summation models can, therefore, result in the selection of paths of different path

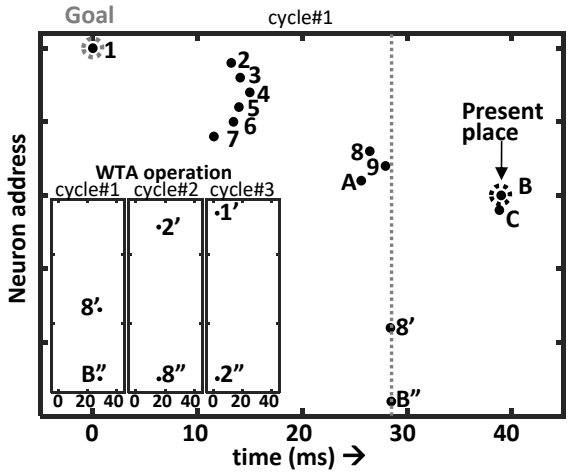
lengths. Specifically, planning using the current saturation model selects a path with shorter path length, but picking the longer path is better if there are obstructions.

Like the previous example, in the first scenario, there are no obstructions. The path selected using the current saturation model has a lower average path length than the current summation model. In the second scenario, however, after averaging over all possible paths and single obstruction failures, planning using the current summation model yields a shorter average path length. Thus, even though the path length is longer when the current summation model is used, if there is an obstruction, then on an average, choosing the longer path is better. The results are shown in Fig. 6.6.

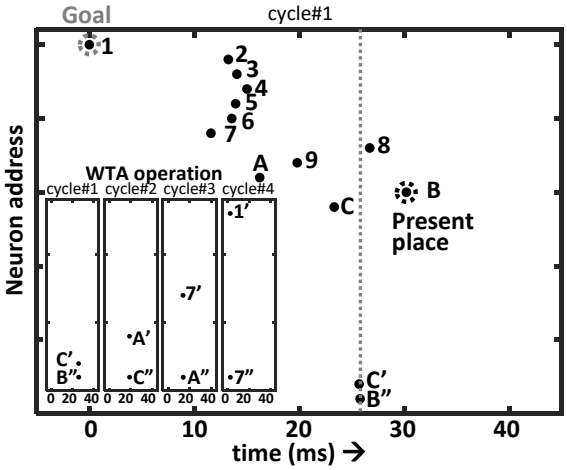
Therefore, if two paths with equal lengths are considered, then the current summation model would pick a path with places that have more alternatives. In some specific cases, where there are multiple convergences in the path, the current summation model can pick a path with a longer length.



(a)



(b)



(c)

Fig. 6.5. Effect of multiple convergences in a graph. In (a), the goal is represented by place cell 1 and the current location of the agent is represented by place cell B. In first step of the plan, the directions neurons are 8' and C' (local inhibitory neuron is not shown in the figure). (b) is current saturation model response. Stimulation of place cell 1 causes place cells 2-7 to spike, followed by 8, 9, and A. Place cell 8 causes 8' to spike and C' is inhibited. Therefore, the path B→8→2→1 is selected for execution, depicted in the inset figure showing spikes in the temporal WTA. (c) is current summation model response. Due to convergences at 9, A and C, C spikes earlier than 8 causing C' to spike and 8' is inhibited. The path B→C→A→7→1 is selected for execution. From [2].

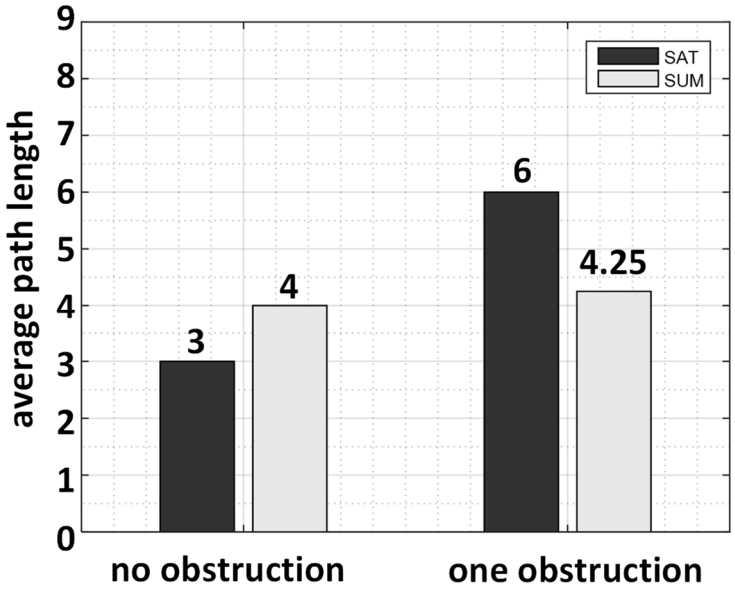


Fig. 6.6. Comparison between the current saturation and current summation models using average path length for the 12-node graph in Fig. 6.5. For the no-obstruction run, the current saturation model has a lower average path length. For the second run, with a single obstruction, the current summation model has a lower average path length. From [70].

6.5 Planning on Simulated Agent Generated Graph

To demonstrate a larger, complete, planning example, a graph was generated using the mapping algorithm described in Chapter 3. A simulated agent explores a maze using an obstacle avoidance algorithm and builds a place cell-based representation of the maze (see Fig. 3.2 for the mapping simulation). In this example, the goal is associated with place cell 21 and the agent is located at a place represented by place cell 1. Each step of planning starts with stimulation of place cell 21 and the spiking activity spreads through the network (shown in Fig. 6.7). In this example, there are three equivalent paths with the shortest length. These are path 1→14→15→ … 21, path 1→ 2→3→4→5→… 21 and path 1→2→3→4→23 … 21. Fig. 6.8 (a) depicts planning using the current saturation model. Ideally, any equivalent path could get selected. In

this example, however, circuit mismatch biases the selection towards the path 1→14→15 … 21. The activity wave from the "choice" on the left side of the maze arrives first and thus direction neuron 14' fires first and inhibits direction neuron 2'. The agent moves to the location represented by place cell 14 and in the following planning steps, it keeps moving along the left-hand path until it reaches the goal represented by place cell 21. Fig 6.8 (b) depicts planning using the current summation model. In this case, only the right-hand "choice" can be selected due to convergence at place cell 4. In this particular example, the path 1→2→3→4→23 … 21 is selected. The direction neuron 2' fires first and inhibits 14' and the agent moves to the location represented by place cell 2. When the agent is at the location represented by place cell 4, the WTA picks direction neuron 23' and thus the agent moves to the place represented by place cell 23 and continues on to the goal. In this example, between a simple path and a path rich in choices, the current summation model favors the one with more alternatives.

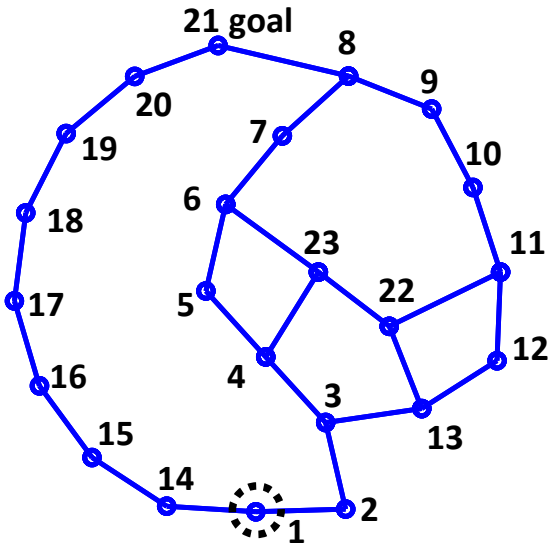
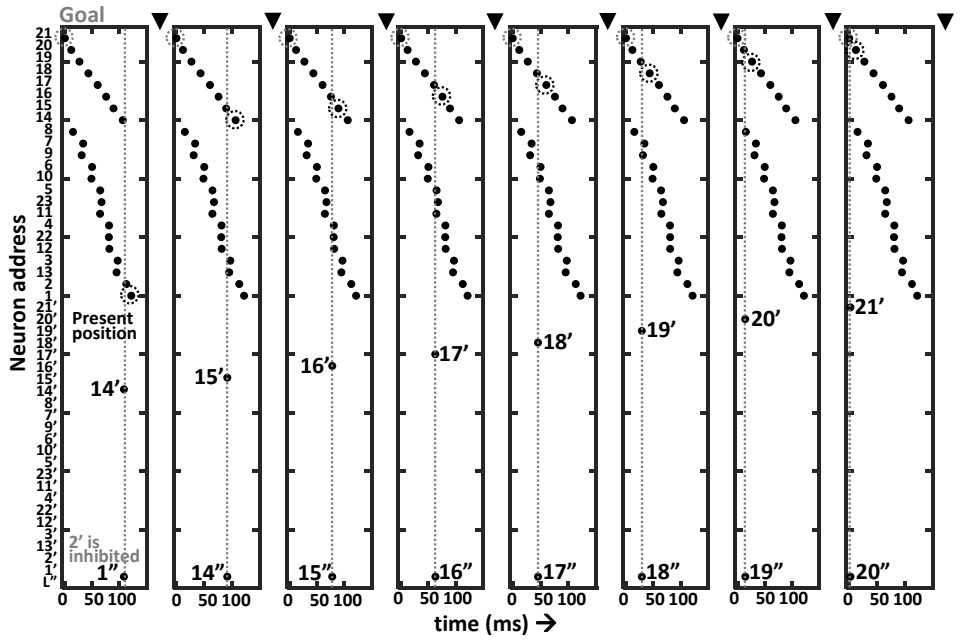
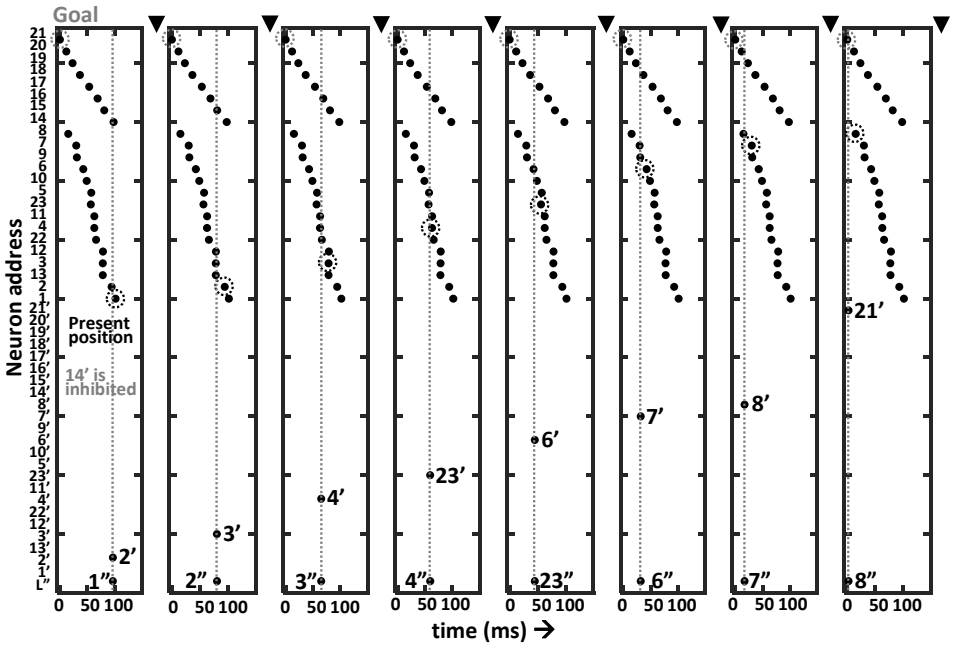


Fig. 6.7 Showing the graph that was generated from the mapping simulation (in Fig. 3.2-3.3). The goal is represented by place cell 21. There are two choices from the starting location that is represented by place cell 1. One of the choices is a straight path to the goal (on the left). The other choice (on the right) offers multiple ways to the goal. From [2].

The results of the analysis to compare the effect of the current saturation and current summation models are shown in Fig. 6.9. In the first run with no obstructions along the path, the average path length is 8. In the second run with a single obstruction anywhere along the selected path the average path length approximately increases to 12 for both the models. The planning process using the current summation model, however, has a lower average path length than the current saturation model. Like previous examples, the agent picks the path with more redundancy.



(a)



(b)

Fig. 6.8 A complete planning example on a graph generated from the mapping simulation. The neuron addresses are shown on the y-axis. The addresses for the place cells are arranged in two parts. Each part corresponds to one of the two choices in graph shown in Fig. 6.7. In each part, the addresses are arranged according to the sequence of firing. L" represents the local inhibitory neuron. (a) Planning using the current saturation model shows the selection of the path which corresponds to picking the path that leads straight to the goal and (b) Planning using the current summation model shows the selection of the path which corresponds to picking the path that has more alternatives. The inverted solid triangles between spike raster plots represent path execution. From [2].

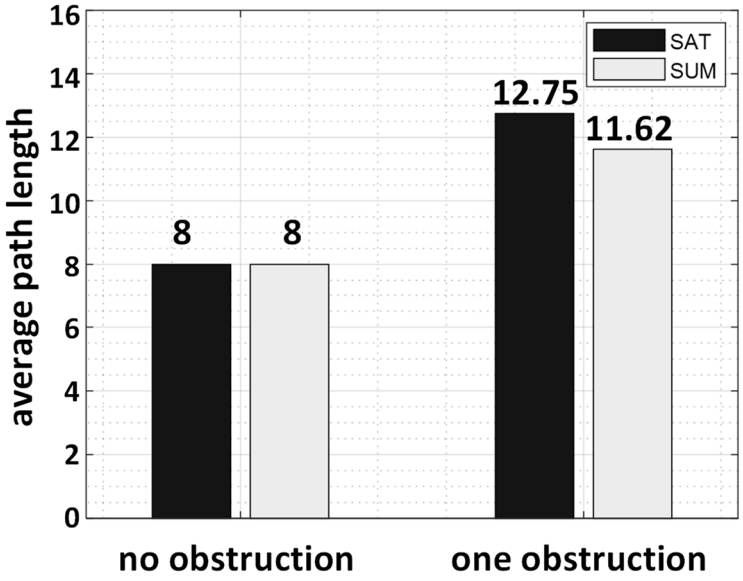


Fig. 6.9. Comparison between the current saturation and current summation models using average path length on a graph (in Fig. 6.7) generated by a simulated agent exploring a maze. For the no-obstruction run, both the models have the same average path length of 8. For the run with a single obstruction, the current summation model has a lower average path length. From [2].

6.6 Summary

In this chapter, the effect of synaptic-dependent spike-latency on path planning was analyzed. Path planning on a network of place cells using the current saturation model of the place cell was compared with the current summation model. Path planning with the current saturation model yields paths that have the fewest number of hops, whereas the results of path planning with the current summation model were interpreted as offering more redundancy in case of blockage. To make this evaluation, a method was introduced in which the total cost of travel after path execution was compared before and after the introduction of an obstruction along the planned path. In response to an obstruction, the path planner must replan and this typically increases the cost of travel. By comparing the cost of travel in the presence of an obstruction, the solutions

of the path planning systems were evaluated in a potential scenario of meeting with an obstruction. The risk-aversive choice could be made more adaptive if there is a mechanism that slows down the neurons that have fewer synaptic connections, thus biasing the selection of the paths towards those with more alternatives.

Chapter 7: Conclusion

7.1 Summary

The aim of this research was to explore how place cells can fit into a spatial memory system and how the spreading activation mechanism for path planning in memory can be implemented in such a biologically-plausible system. Key to this exploration was desire to demonstrate a neuromorphic VLSI circuit implementation of a spatial memory and long-range memory-based planning in hardware that could be used with other echolocation-based robotics research projects in the laboratory. The motivation for doing this was to build a system with high speed and flexible response. The understanding was that place cells provide a place representation system that works with the lower level sensorimotor responses and uses less memory during pathfinding to previously visited locations. This does not guarantee optimal choices, but provides a fast response. By taking the neuromorphic approach, we could build more efficient circuits in the long-term.

Among the possible mental representations of the environment that could support long-range navigation, we selected a topological representation, because animals lack precise sensors for creating large global and metrically accurate representations like those available on modern robots. Since hippocampal place cells are believed to support the place memory system, we modeled an interconnected network of place cells as a cognitive graph, where being or imagining to be in a certain place in the environment is signaled by the activity of a corresponding place cell and the links in the graph are represented by reciprocal connections between the place cells.

Typically, the problem of navigation can be broken into subproblems like mapping, planning and path execution. The emphasis in this research was laid on the planning subproblem. We designed multi-neuron chips with synapse and neuron models suitable for the implementation of a spike-based, spreading activation, path planning system. In this system, the planning process was carried out through parallel search of multiple paths using a spreading wave of activation that is initiated by stimulating the goal location place cell and by selecting the path that has the shortest spike latency using a temporal winner-take-all that observes the spiking activity arriving at the current location place cell. This system was used to demonstrate planning in a different planning task like controlling a multi-segment robotic arm.

Since neurons spike with a shorter latency, when driven with a stronger stimulus, there was reason to believe that this would affect the response of the path planning process. So, two models were introduced that could be used to discriminate this effect. These synaptic-dependent spike-latency models were current saturation and current summation models. The response of the first model is to saturate the spike latency irrespective of the number of spikes. The response of the second model is to fire with a shorter spike latency, when more spikes are received at the same time. It was shown that the response is different for both the models. Additionally, it was seen that the path selected by the current summation model could be interpreted as selecting paths with more redundancy. If there was a single path obstruction in the planned path, then this choice would lead to a shorter overall path after re-planning and this choice was interpreted as a risk-averse response.

For the sake of completion, a system level simulation was carried out that consisted of three temporally distinct steps of mapping, planning and path execution. In the mapping step, a cognitive graph representation of the environment was created and at the same time, a grid-cell based activity pattern was associated with each place for which a place cell was recruited. In the path execution step, current and memorized grid cell activity were used to generate a direction for movement between the current location and the memorized waypoint. For this, a decoder was used, which was designed by another researcher [47] and incorporated into our system.

7.2 Contribution

In this research, the problem of spatial navigation was attacked from the neuromorphic perspective. Earlier work from this laboratory emphasized the creation of a neuromorphic head direction cell system, grid cell system and grid-cell-based place cell system. This work extended prior research that was about creating the representation of the environment to navigating in the environment. More recent work in the neuromorphic based navigation systems includes [53], [54], [71], [66]. There are specific examples [7], [14] in neuromorphic VLSI that provide solutions to the path planning problem. Their focus was primarily in exploring a new analog processing chip system and a large-scale digital neuron chip system respectively for the path planning problem. Our focus was to limit the assumptions to those that were more biologically plausible. We assumed that the underlying representation of the environment is not dense and metric, although a similar assumption could be made in their system as well. Additionally, their method of selecting the path for navigation was done external to the multi-neuron chip, whereas that step was incorporated into the network in this work.

We also investigated how a neuron's response to increased stimulus would affect the path that is selected by the planning process.

7.3 Future Work

An issue with the original architecture is that for the system to work, every place for which there is a corresponding place cell, there should be multiple direction neurons that correspond to the neighbors of the place cell and there should be specific connectivity between these direction neurons and the local inhibitory neuron corresponding to the place. Additionally, this connectivity has to form while the animal is learning the spatial layout of the environment. Instead, it is possible to conceive an intracellular mechanism that causes a buildup of a chemical substance in place cells that are active while the animal is exploring the place and as soon as it wishes to go to a memorized location, the goal location place cell stimulation causes spikes to spread through the network and the neurons in which the chemical substance had built up causes a different response, for example, a bursting response, to the oncoming wave of activity. This would essentially create a temporal winner-take-all in the cells representing the neighborhood of the present location without requiring the specific connectivity between the direction neurons and local inhibitory neuron.

An issue with synaptic-dependent spike-latency based planning method is that it is not possible to control the path selection among paths with different number of alternatives to variable degree. For example, if one interprets selecting a path with more alternatives as risk-aversiveness, then it is not possible to control the degree of risk-aversiveness. An intracellular mechanism that could achieve this effect could make use of the prior planning activity to identify the neurons with multiple synaptic connections

and separate them from those with fewer connections by accumulation of a chemical substance, whenever the place cell is stimulated during planning. This substance would have a short-term effect on the spike latency by maintaining the spike latency of neurons with greater amount of the chemical substance and slow down the neurons with smaller amount of the chemical substance. If the increment in the chemical substance is controlled, then it is possible to control the degree of difference in spike latency among the paths with different alternatives. This would achieve the desired effect of controlling the overall spike latencies among paths with different number of alternatives.

Appendix-A: Effect of Fabrication Mismatch on Path Selection

Spike latency variability across neurons in the multi-neuron chip is determined by the extent of transistor mismatch. An estimate of the impact of this variability on selecting the correct path is made by comparing the probability of selecting the correct path over another path that is just one hop longer. If there are multiple paths with the same number of hops, then this variability helps in breaking the tie. Fig. A.1 (a) describes a method for making this comparison. Two paths that just differ by single hop are compared based on their total spike latency. The probability of selecting the shorter path is computed and the trend for the increasing number of hops in the correct path is plotted. It is assumed that spike latency for each neuron in the path is normally distributed (observed in Fig. 4.6) and is from the same distribution. The probability distribution of total spike latency for a chain of such neurons can be found by adding the spike latency random variables (for an N-hop path, the effective mean is N times the mean and the effective variance is N times the variance). The standard deviation in spike latency observed is 17.7% of the mean spike latency. If place cells were selected from the chips at random and the total spike latency of each path was compared, then the probability of selecting the correct path can be found by adding the probability for all cases when the total spike latency of the longer path is longer than the shorter, correct path. In Fig. A.1 (b), for a confidence level of 95%, if the spike latency variability of the population changes from 20% to 5%, the number of hops (N) that can be reliably discriminated would change from 4 up to 73. If the variability is further reduced, then the range of selecting the correct path could be extended even more. On

our chips, the variability was fixed close to 10% and for the confidence level of 95%, this translates to paths with 17 hops.

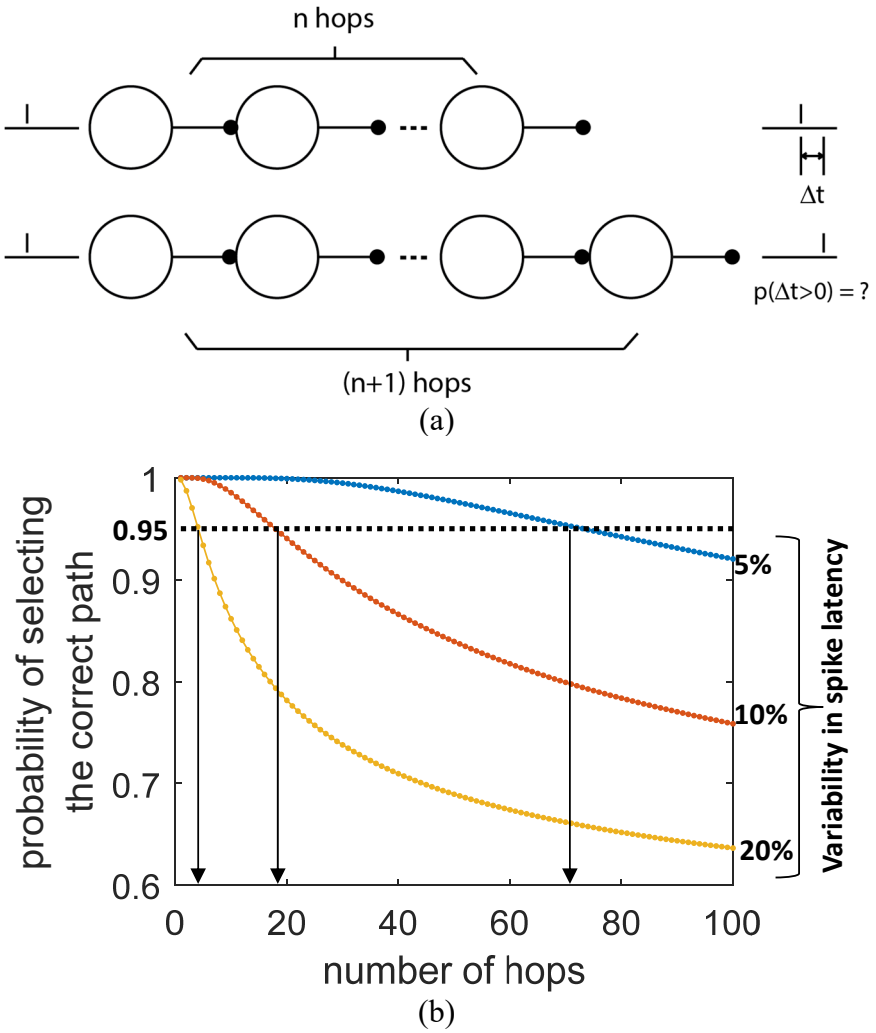


Fig. A.1. Showing the effect of variability in place cell spike latency on reliably selecting the correct path. (a) shows how variability in spike latency affects correct path selection. Here variability is defined as the ratio of the standard deviation in spike latency and the mean spike latency. (b) shows the trend in selecting the correct path as the number of hops in the path increase. With a confidence level of 95%, having 20% variability in spike latency only permits selecting paths of length 4 correctly. When the variability is reduced, the numbers of hops up to which correct selection can be made increases. From [2].

Appendix-B: Effect of Transistor Size on Subthreshold Current

In a MOS current mirror, if the input and output transistors have the same width by length ratio, the currents are expected to be the same. For different channel sizes, the currents are different due to lateral diffusion effect under the transistor gate. The difference between the currents is much greater if one of the devices is smaller than a certain limit, especially in the subthreshold region of operation. This effect can be observed in simulation of an NMOS transistor model for a commercially available 0.5 μm fabrication process. To verify this effect on silicon, we fabricated a chip with 36 NMOS transistors with 6 different channel widths and lengths. A comparison was made between the drain currents for unity width-by-length ratio as a function of channel width in the subthreshold region of operation. The difference observed in the simulation was also observed on chip. We think that this is caused by the small channel effect. We plotted current ratios using fixed channel width ratios and also using fixed channel length ratios. We observed that in the subthreshold region of operation, the current ratios using channel length ratios are closer to the expected value as compared to channel width ratios.

B.1 Background

We ran a simulation using BSIM3v3.1 NMOS transistor model (provided by MOSIS for a 0.5 μm fabrication process). We applied V_{GS} = 0.5 V and V_{DS} = 1V to the transistor and swept the channel width for unity width-by-length ratio. We plotted the current as a function of channel width (shown in Fig. B.1).

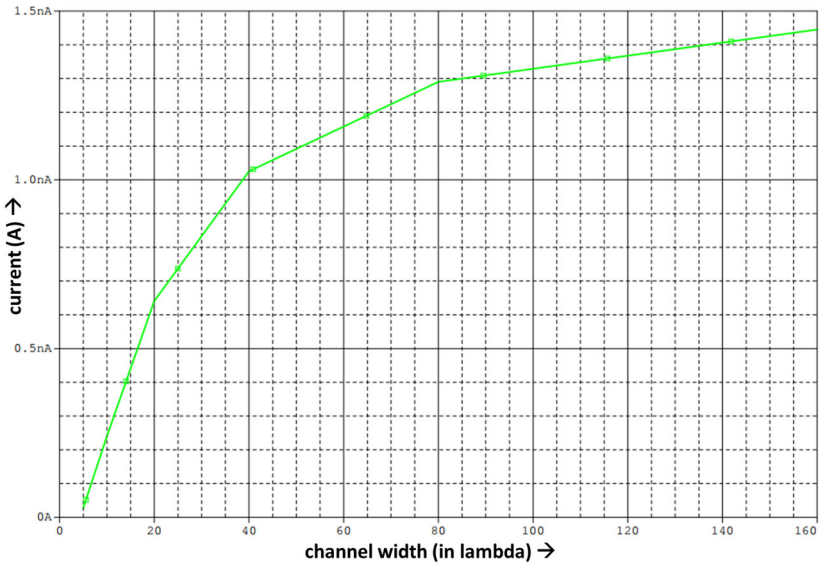
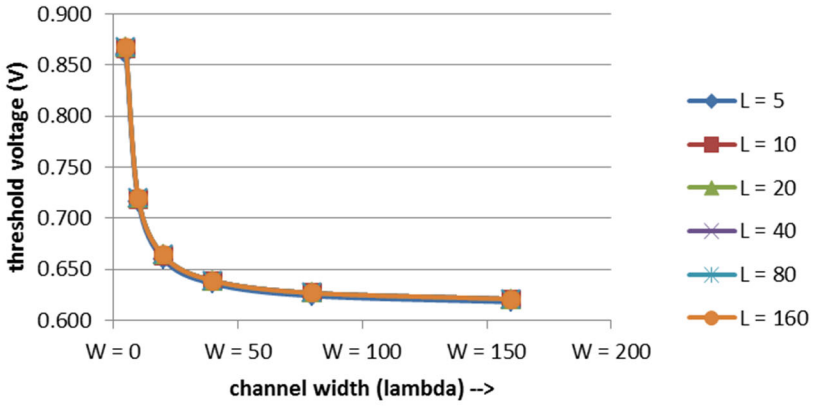
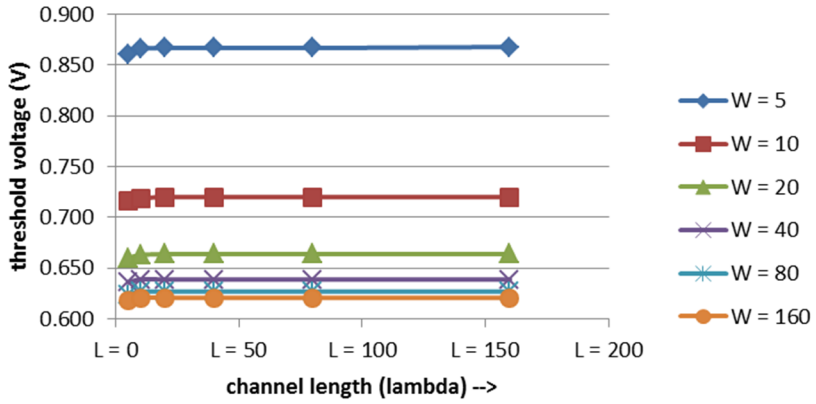


Fig. B.1. Showing NMOS transistor drain current as a function of channel width for unity width by length ratio. V_{GS} = 0.5 V and V_{DS} = 1 V. λ = 0.35 µm and it is varied in the set {5, 10, 20, 40, 80, 160}. We expect the current to be equal for the same width by length ratio, however, there seems to be two orders of difference in the magnitude.

We observed that there was around two orders of magnitude difference between the currents of the largest and the smallest size transistors. We suspected the reason to be small channel effect and therefore we plotted the threshold voltage for the transistor as a function of transistor size. This is shown in Fig. B.2, where the threshold voltage is large for narrow channels (~850mV) and it drops to ~600 mV for large channel widths. On the other hand, the threshold voltage does not change very strongly with the change in channel length.



(a)

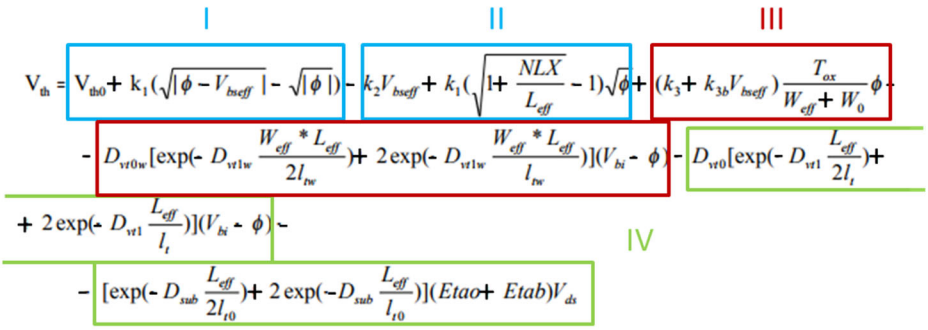


(b)

Fig. B.2. Showing the variation in threshold voltage with transistor channel width and channel length. (a) For channel width of 5 λ, the threshold voltage is ~850 mV and for channel width greater than 20 λ, the threshold voltage is under 650 mV. In the subthrehsold region of operation, this difference in threshold voltage would create a large difference in the drain current. (b) For channel lengths of 10 λ and below, the threshold voltage decreases. However, for longer lengths, the threshold voltage nearly remains the same. Therefore, the subthreshold current per unit length would not vary much with the channel length.

The threshold voltage for a MOS transistor is defined by comparing the charge carrier population density underneath the gate with the doping concentration of the semiconductor substrate. This voltage separates the operation of the device into two regions, subthreshold and above-threshold. The numerical value of this threshold voltage is computed by using a MOS capacitor with the same size and assuming an

infinite parallel plate capacitor model. For channels smaller than a certain size, this assumption is not accurate and certain correction terms are added to account for the deviations. This is shown in equation B.1 (from [72]).



$$V_{th} = V_{th0} + k_1\left(\sqrt{|\phi - V_{bseff}|} - \sqrt{|\phi|}\right) - k_2 V_{bseff} + k_1\left(\sqrt{1 + \frac{NLX}{L_{eff}}} - 1\right)\sqrt{\phi} + (k_3 + k_{3b}V_{bseff})\frac{T_{ox}}{W_{eff} + W_0}\phi - D_{vt0w}\left[\exp\left(-D_{vt1w}\frac{W_{eff} * L_{eff}}{2l_{tw}}\right) + 2\exp\left(-D_{vt1w}\frac{W_{eff} * L_{eff}}{l_{tw}}\right)\right](V_{bi} - \phi) - D_{vt0}\left[\exp\left(-D_{vt1}\frac{L_{eff}}{2l_t}\right) + 2\exp\left(-D_{vt1}\frac{L_{eff}}{l_t}\right)\right](V_{bi} - \phi) - \left[\exp\left(-D_{sub}\frac{L_{eff}}{2l_{t0}}\right) + 2\exp\left(-D_{sub}\frac{L_{eff}}{l_{t0}}\right)\right](Etao + Etab)V_{ds} \quad \text{(B.1)}$$

The terms in I represents the long channel threshold voltage. The terms in II represent the correction made to the threshold voltage to incorporate the lateral and vertical doping difference in the substrate (this doping profile is a common practice in scaling down transistors). The terms in III represent the narrow channel effect. The first term captures the threshold variation with channel width, whereas the second term is added to handle narrow channel effect in short channels. The terms in IV represent the short channel effect and DIBL effect. We found the first term in III to account for most of the threshold variation and that is discussed in detail below.

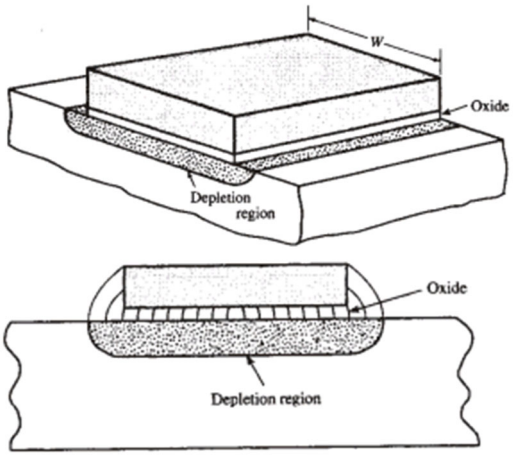


Fig. B.3. A simplified model for a MOS capacitor with fringe fields at the sides of the channel width. To reuse the formula for computing the threshold voltage, a voltage shift in the threshold voltage is introduced. This shift is calculated by dividing the additional charges by the gate oxide capacitance. The additional charges under the gate are two quarter-cylinders of charges on either side of the original charges under the gate. Taken from [73].

One of the correction terms applied to the threshold voltage calculation considers the fringe electric fields that stretch from the ends of the gate along the channel width as depicted in the Fig. B.3. The fringe electric fields that extend outwards from the gate create additional charges in the semiconductor substrate and typically a cylindrical approximation is used to model the additional charges [73]. It is assumed that the cylindrical extensions are as wide as the depletion depth. In this approach, the ideal depletion charges contribution to the threshold voltage is multiplied by a factor that is estimated by taking a ratio of the total charges for the corrected case and the ideal case. The charges underneath the gate are computed by multiplying the volume charge density with the volume $W*L*d_B$, where W and L are width and length of the transistor channel and d_B is the depletion depth. For a channel with quarter-cylindrical extensions, the new volume is $W*L*d_B+0.5*\pi*L*d_B^2$. Therefore, the ratio of the new estimate over the ideal estimate is $1+0.5*\pi*d_B/W$. The additional term $0.5*\pi*d_B/W$

multiplied by Q_B/C_{ox} gives the shift in threshold voltage, where Q_B stands for bulk charge density and C_{ox} is the oxide capacitance per unit area. Since Q_B is $q*N_A*d_B$, this expression becomes $0.5 *\pi *q*N_A*d_B^2/(C_{ox}*W)$. Here N_A is the acceptor doping density. Replacing d_B^2 by $2*e_{si}*\varphi/(q*N_A)$, where φ is the inversion surface potential and e_{si} is the permittivity of silicon, one gets $(e_{si}/e_{ox})*\pi* t_{ox} *\varphi /W$. In the first term in III in equation 1, this is given as $K_3*(t_{ox}/(W+W_0)) *\varphi$, where K_3 and W_0 are empirically derived parameters. There is an additional parameter K_{30} that is added to K_3 and it is dependent on the body potential. By replacing the values form the parameter file (K_3 = 22.8, W_0 = 0.0101 μm, t_{ox} = 14 nm and surface potential) in the equation, around 200 mV of voltage shift is observed in a 5λ-width transistor, which matches the trend observed in the threshold voltage simulation (Fig. B.2). This effect is explained in the following way. For a given charge concentration under the gate, there is more total charge overall below the oxide and therefore there would be greater charge on the gate. A greater charge on the gate for the same oxide capacitance would require a higher voltage and hence a higher threshold.

To verify this, simulations were carried out on PSPICE using the following model files - the original model file, the model file without lateral diffusion correction, i.e. $W_{eff} = W_{drawn}$ (or W) and $L_{eff} = L_{drawn}$ (or L), the model file without lateral diffusion correction and without fringe field effect, i.e. $K_3 = 0$ and $K_{3b} = 0$. For the above-threshold and the subthreshold operations, transistor currents were plotted against the change in width for unity width by length ratio in simulations. The results from the PSPICE simulation are shown in Fig. B.4 and Fig. B.5 for above-threshold and subthreshold region of operation.

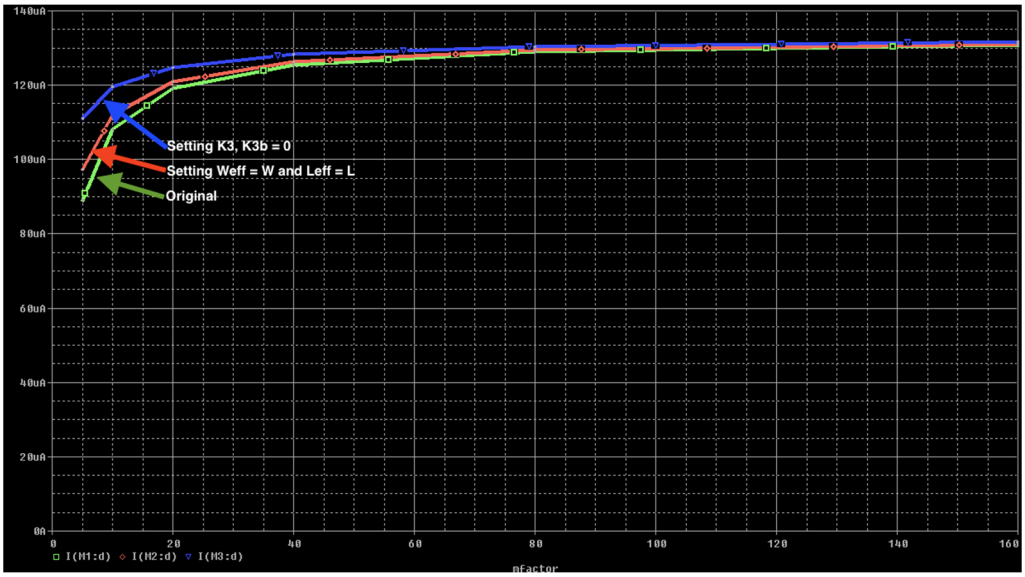


Fig. B.4. PSPICE simulation of the drain current of a MOS transistor in above-threshold region of operation, plotted against different sizes for three different models. The gate of the transistor is at 2.5V. The drain of the transistor is at 1 V. The effect of lateral diffusion is smaller than the removal of fringe fields in the model. The fringe field effect is removed by setting K_3 and $K_{3b} = 0$.

The fringe fields have a strong effect on the subthreshold region of operation for small size transistors and it is difficult to predict that only based on the reduction in the channel widths and lengths caused by lateral diffusion [74]. In the case of above-threshold operation, the effect is visible, but not as strong as the subthreshold operation.

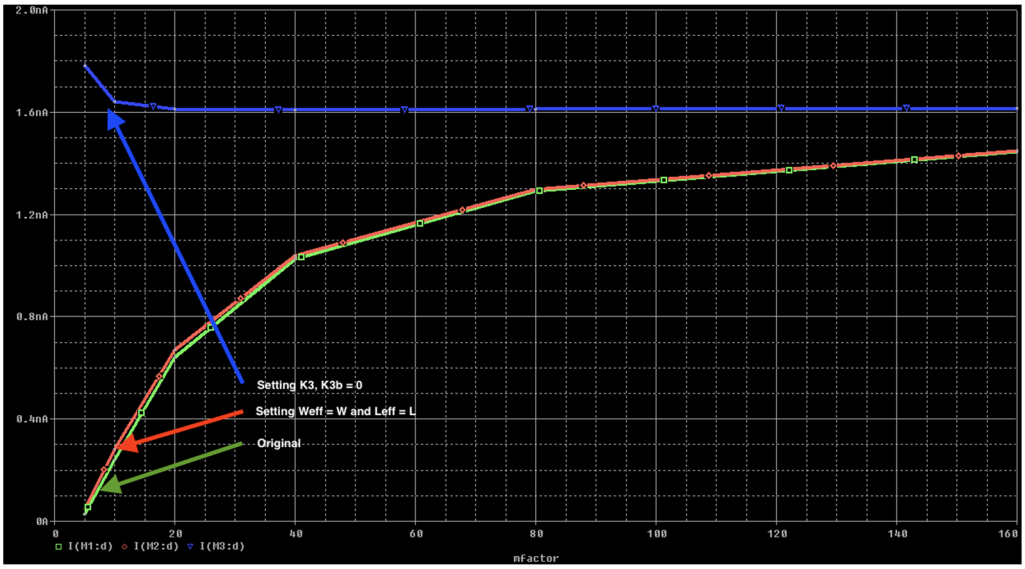


Fig. B.5. PSPICE simulation of the drain current of a MOS transistor in subthreshold region of operation, plotted against different sizes for three different models. The gate of the transistor is at 0.5V. The drain of the transistor is at 1 V. The effect of lateral diffusion is small. Removal of the fringe field effect (by setting K_3 and K_{3b} = 0) makes the curve flatter. The peak in the current at the beginning is because of short channel effect. For shorter channels, one gets a smaller threshold voltage. This short channel effect is noticeable for channel lengths under 10 λ.

B.2 Measurement Setup

The chip consists of 36 devices with different channel widths and lengths. Using this chip, drain currents for unity width by length ratios were probed and current ratios for channel length and channel width ratios were calculated. Fig. B.6 shows the image of the layout and the chip photomicrograph.

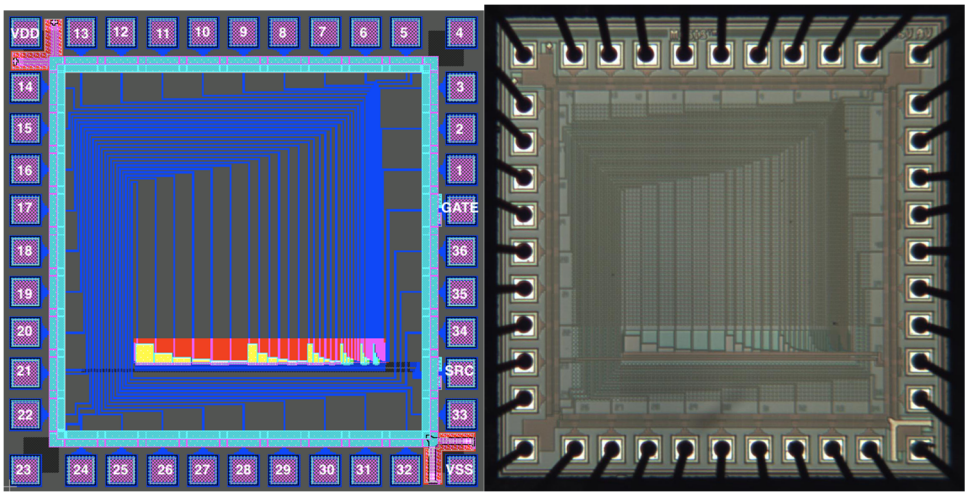


Fig. B.6. Showing the layout of the chip. The numbers on the bond pads show signal names. There are 6 sets of devices. Each set has a fixed width and the length changes from 5 λ to 160 λ (doubling). The widths across the sets vary from 5 λ to 160 λ (doubling). All the devices have a common source and bulk connection. The drains for each set are connected to pads {2-7}, {8-13}, {14,16-20}, {21-26}, {27-32} and {33-34,36,38-40}. The gate is provided on pad 1. The source is provided on pad 37. VDD and VSS are provided on pads 15 and 35. Only gate and source have ESD protection.

These chips were fabricated and packaged through MOSIS provided foundry and packaging services. A scheme of the DIP is given in Fig. B.7.

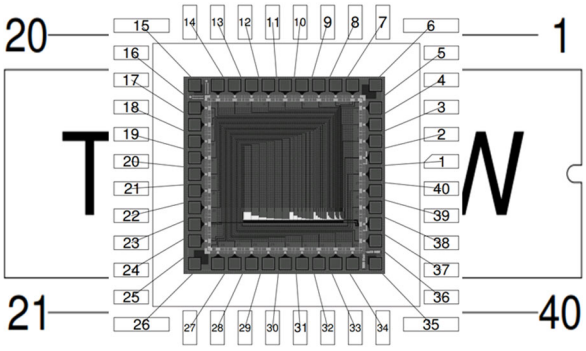


Fig. B.7. Showing orientation of the chip inside the package and package with pin number range. The first 3 sets of transistors are accessible on the left side of the package and the next 3 sets of transistors are accessible on the right side of the package.

This chip was mounted on a breadboard, which was shielded in an aluminum box. Programmable voltage sources were used to provide the voltage to the gate and

drain of the transistors and an electrometer was used to measure the current between the drain and the programmable voltage source supplying the drain voltage. The cables to the box were shielded using low noise coaxial cables. The electrometer input is a triaxial cable, with the outermost conductor connected to the earth and the box shield. The VDD, VSS were provided using an external power source set to 5 V. A 10 μF capacitor was used as a bulk capacitor for the voltage source supplying VDD. Snapshots of the connections and the measurement setup are given in figures Fig. B.8.

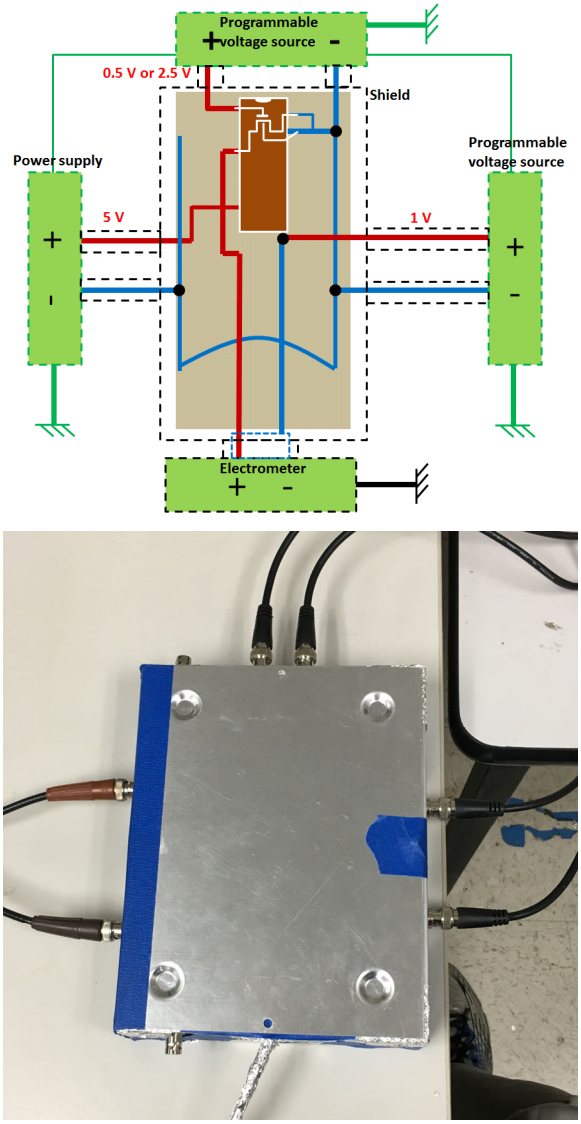


Fig. B.8. Showing the connections in the setup. The gate of the transistor is supplied with 0.5 and 2.5 V depending upon subthreshold and above-threshold operation. The drain voltage is set to 1 V. Between the drain voltage and the drain of the transistor, the electrometer leads are connected. VDD is provided by a voltage source set to 5 V. The VDD drives the guard ring and the ESD diodes on the chip. The chip is mounted on a breadboard pasted to the aluminum box that is put at the same potential as the electrometer earth. The bodies of the programmable voltage sources and the power supply are connected. VDD and VSS are provided from the left side. The gate voltage is provided from the top. The drain voltage is provided from the right side. The electrometer input is provided from the bottom. The shield for the aluminum box is separate from that of the voltage supplies.

Two measurements were made. The first measurement was made for the above-threshold region for which 2.5 V was provided at the gate and 1 V was provided at the

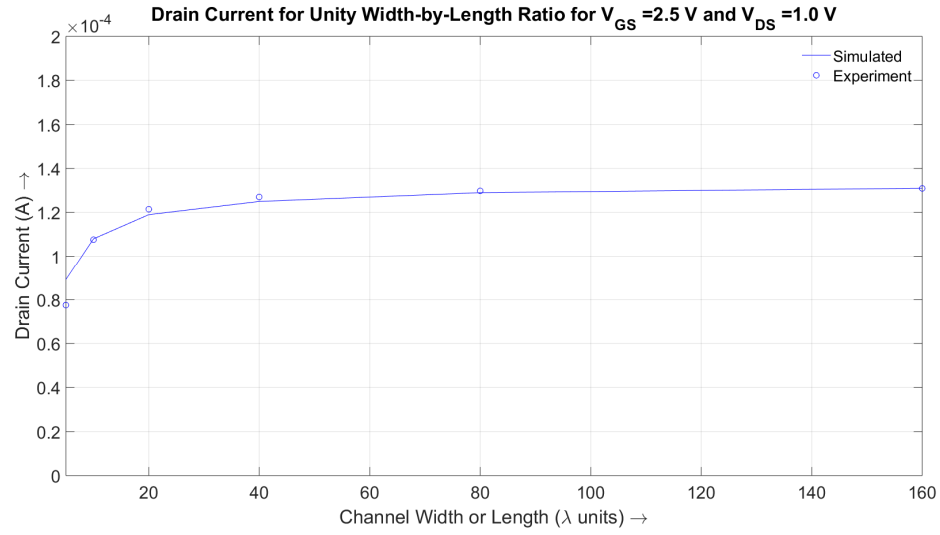
drain. The second measurement was made for the subthreshold region for which 0.5 V was provided at the gate and 1 V was provided at the drain. NOTE: In this report, data from only one chip is reported.

B.3 Results

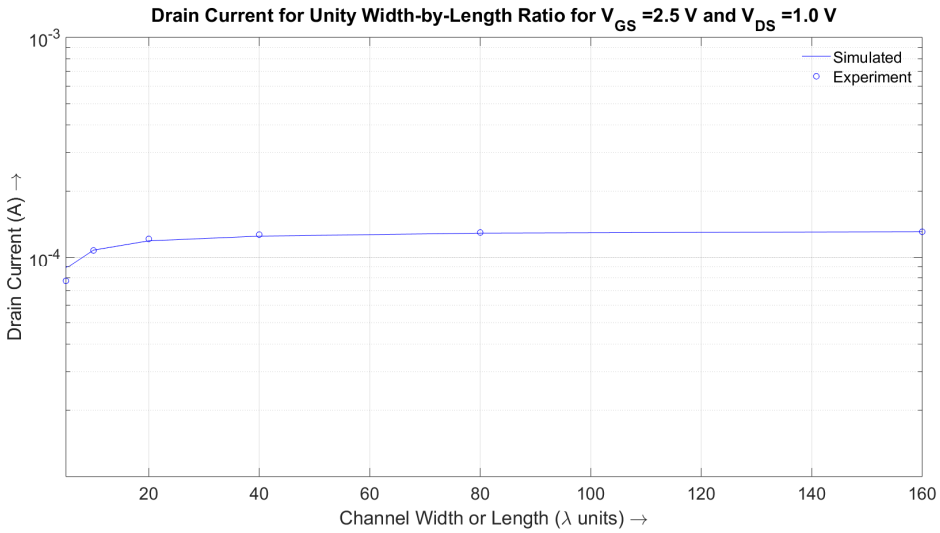
The first set of results (Fig. B.9-B.10) shows that the difference between the maximum and the minimum current for unity width-by-length ratio is much greater in the subthreshold region of operation compared to the above-threshold region of operation.

The second set of results (Fig. B.11-B.13) shows that creating current ratios using transistor channel length or width ratios are suitable for above-threshold region of operation. For subthreshold region of operation, however, it is better to use channel length ratios.

B.3.1 Drain current vs Channel Width for Unity Width-by-Length Ratio

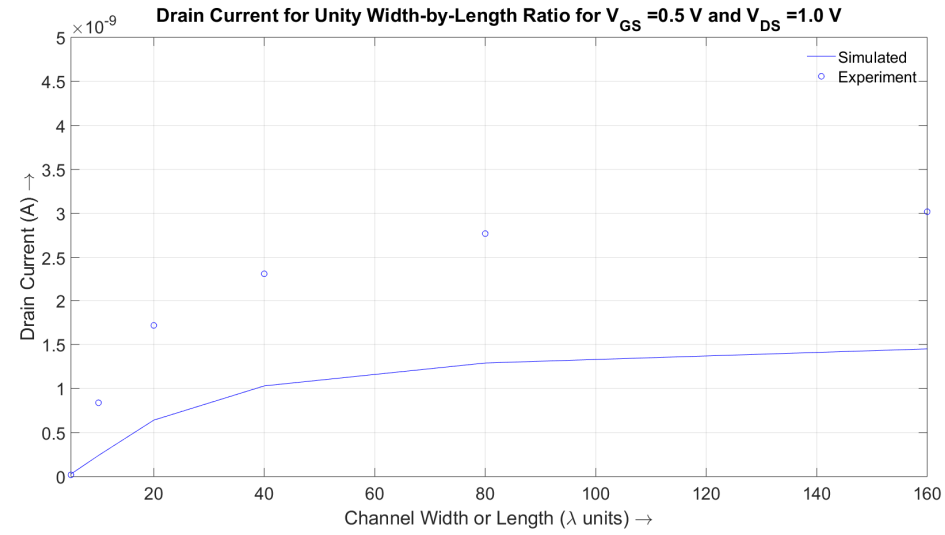


(a)

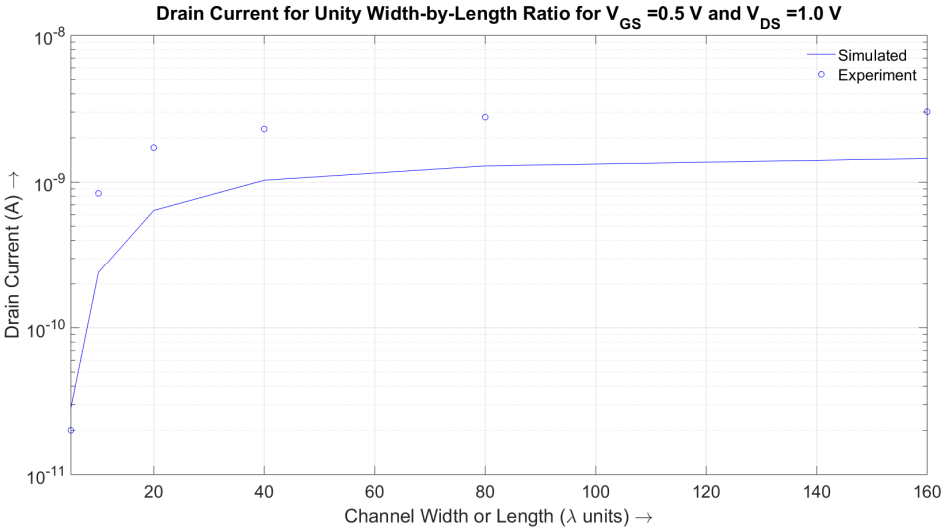


(b)

Fig. B.9. Showing drain current vs channel width or length for unity width-by-length ratio for above-threshold region of operation. (a) drain current vs channel width. (b) drain current vs channel width, semilog plot. The maximum is more than the minimum current by 62%, almost 1.6 times (from experimental values).



(a)

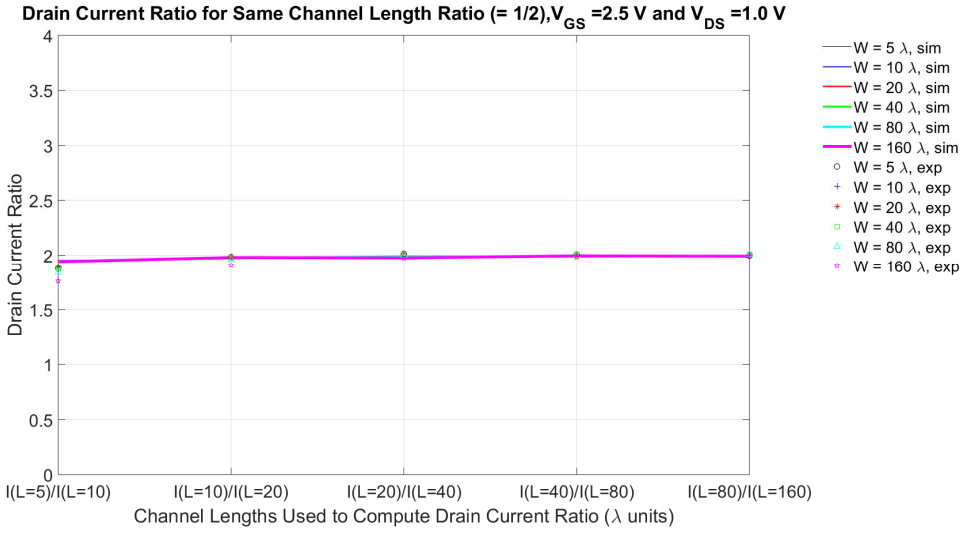


(b)

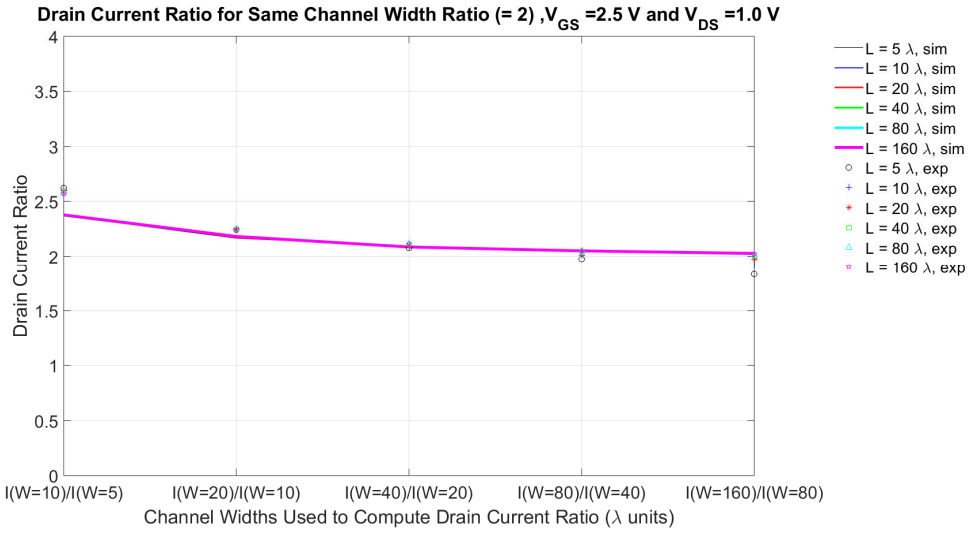
Fig. B.10. Showing drain current vs channel width (or length) for unity width-by-length ratio for subthreshold region of operation. (a) drain current vs channel width. (b) drain current vs channel width, semilog plot. The maximum current is 150 times greater than the minimum current (from experimental values)

In the above-threshold region of operation, the maximum current is more than the minimum current by 1.6 times, whereas in the subthreshold region of operation, it is 150 times. Therefore, this effect is more pronounced in the subthreshold region of operation.

B.3.2 Setting a Current Ratio of 2 Using Length or Width Sizing

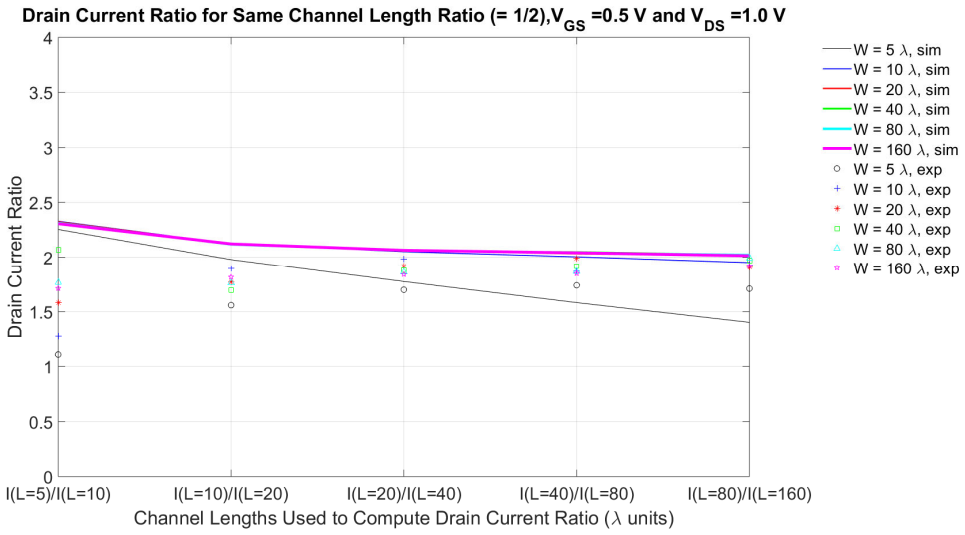


(a)

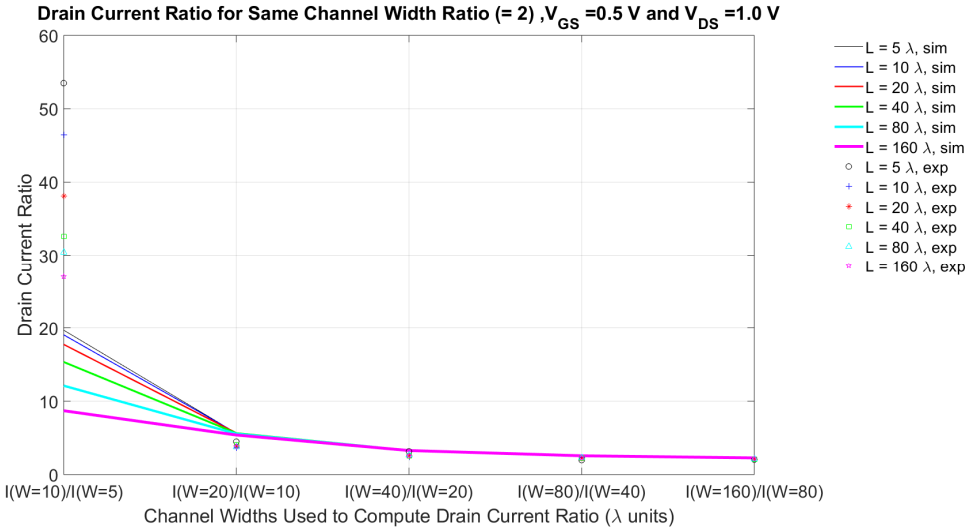


(b)

Fig. B.11. Current ratios for above-threshold operation. (a) Showing the ratio of drain currents for a fixed channel length ratio = 1/2. (b) Showing the ratio of drain currents for a fixed channel width ratio = 2.



(a)



(b)

Fig. B.12. Current ratios for subthreshold operation. (a) Showing the ratio of drain currents for a fixed channel length ratio = 1/2. (b) Showing the ratio of drain currents for a fixed channel width ratio = 2.

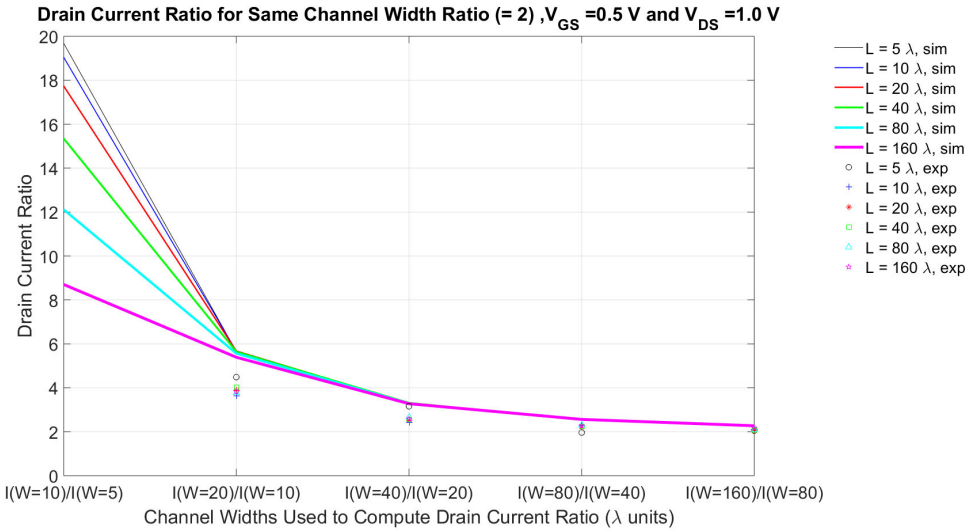


Fig. B.13. Showing the ratio of drain currents for a fixed channel width ratio for subthreshold operation (same as Fig. B.12 (b)) using expanded y-axis. NOTE: experiment data for I(W=10)/I(W=5) is not visible.

For the above-threshold region of operation, the current ratios computed from the simulation and experiment match well. Moreover, the spread of the experimental data is small. For the subthreshold region of operation, the current ratios computed from the experiment data using length ratios show a downward trend (compared to the simulation showing an upward trend) and have a large spread. For smaller channel length ratios for different widths, the spread gets larger. However, overall their values are close to the expected current ratio of 2. In comparison, the current ratios computed from the experiment data using width ratios show the same trend as the simulation and the spread is smaller, except in the case of the shortest widths ratio for different channel lengths, where the spread is much larger than the simulation data. However, overall their values are much higher than the expected current ratio of 2. So, for creating current ratios in subthreshold region of operation using the width by length ratio scaling, using channel length ratios seems to be better.

B.3.3 Drain-Source Current vs Channel Length and Width

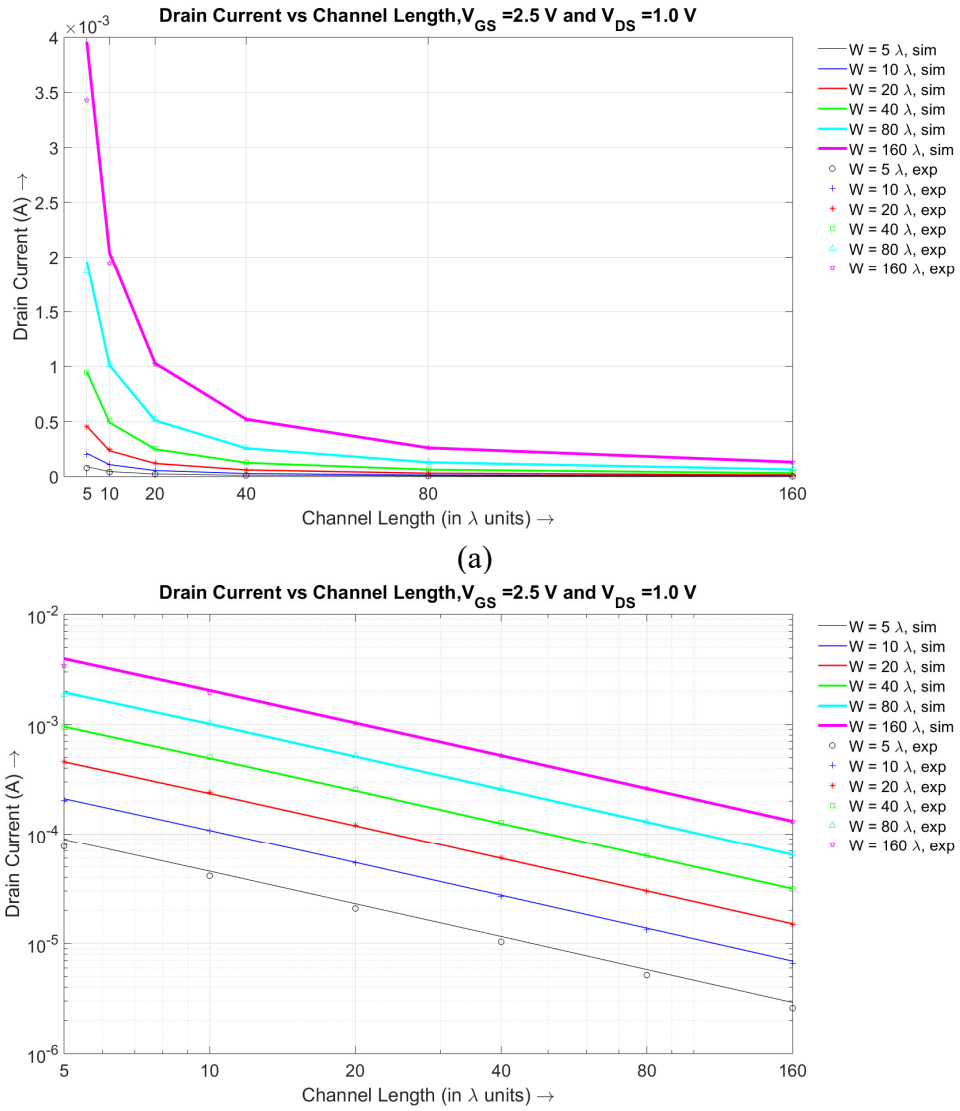
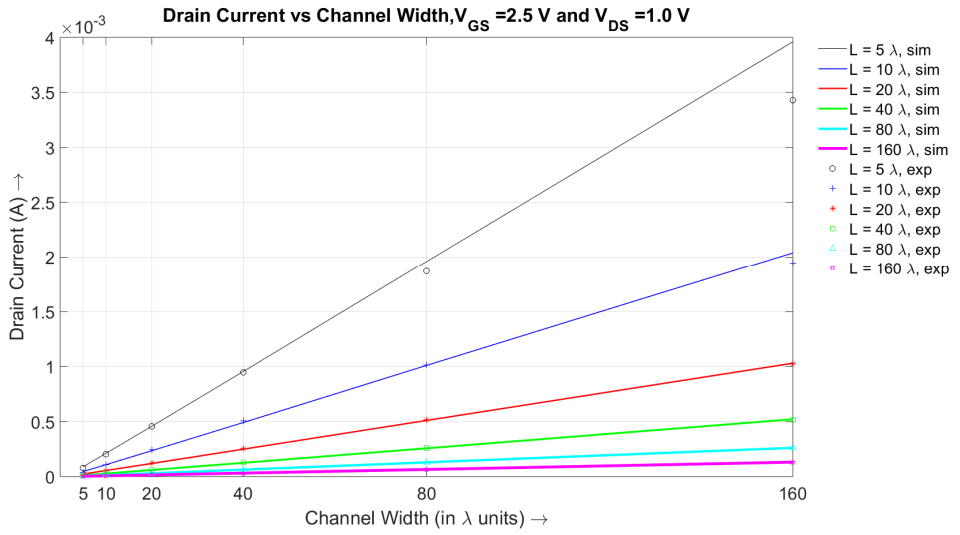
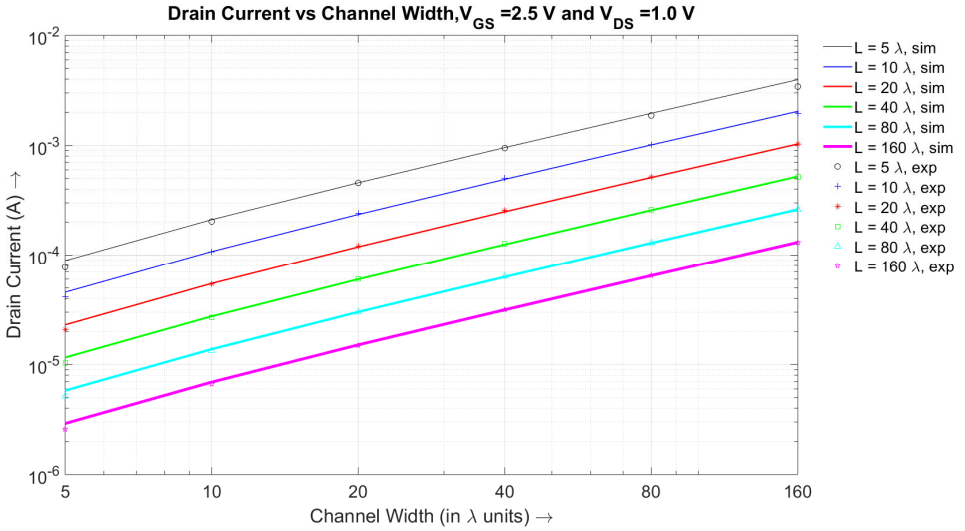


Fig. B.14. Showing drain current vs channel length plots for above-threshold operation taken from simulation and experiment data. (a) drain current vs channel length. (b) drain current vs channel length, loglog plot.

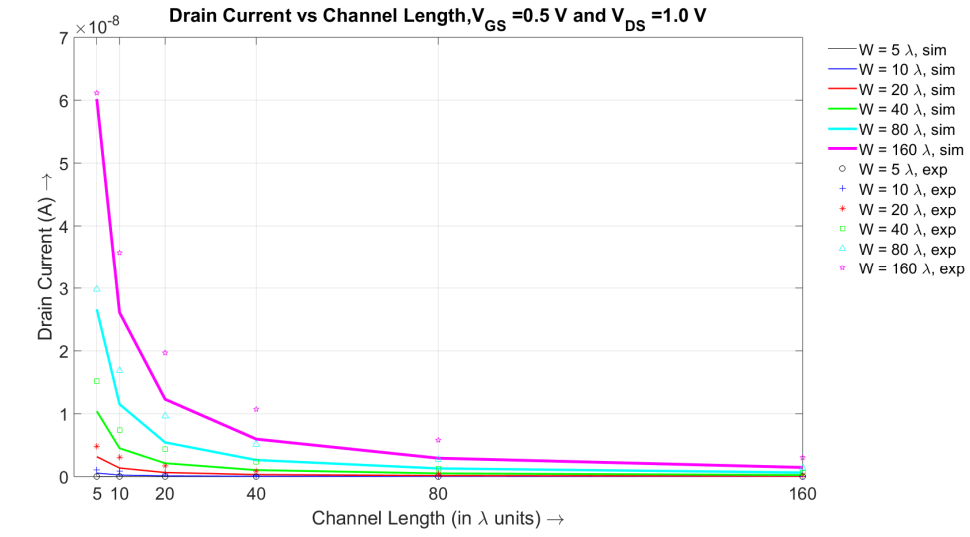


(a)

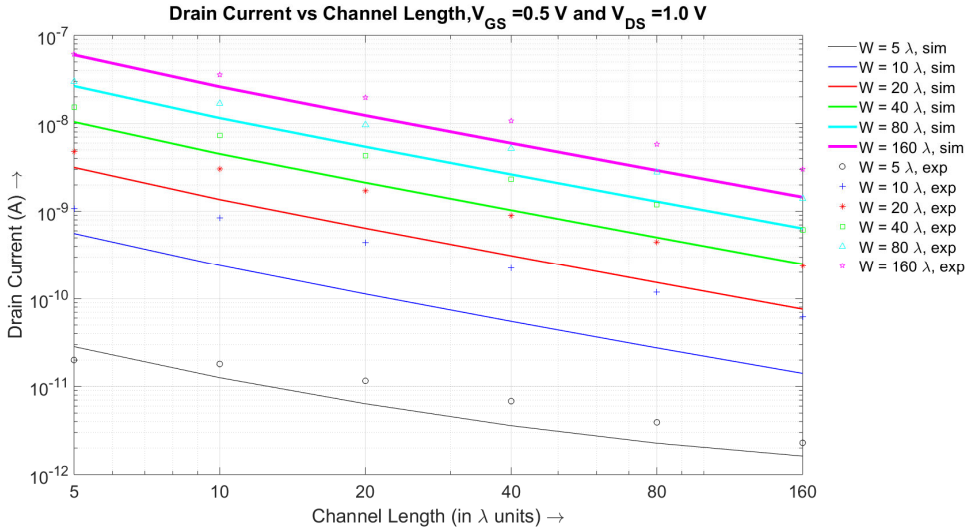


(b)

Fig. B.15. Showing drain current vs channel width plots for above-threshold operation taken from simulation and experiment data. (a) drain current vs channel width. (b) drain current vs channel width, loglog plot.

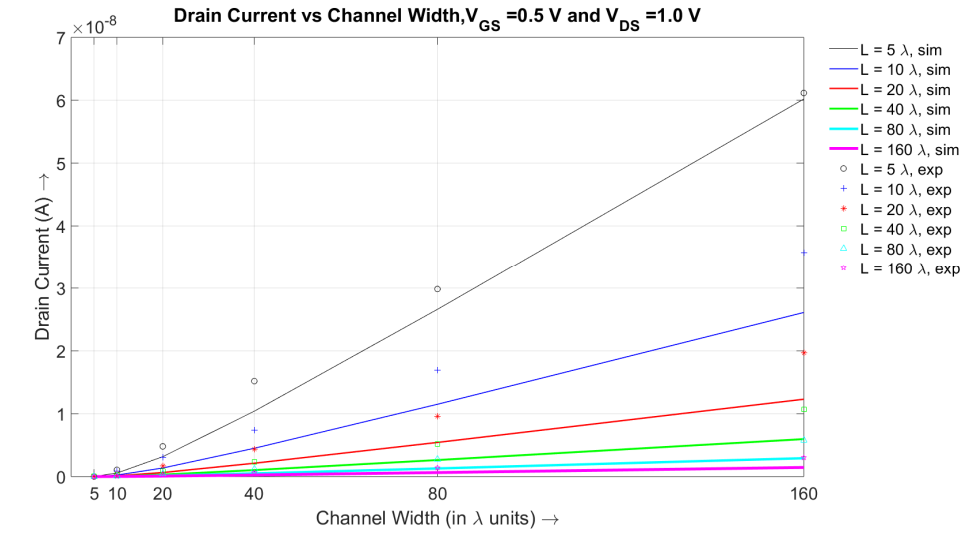


(a)

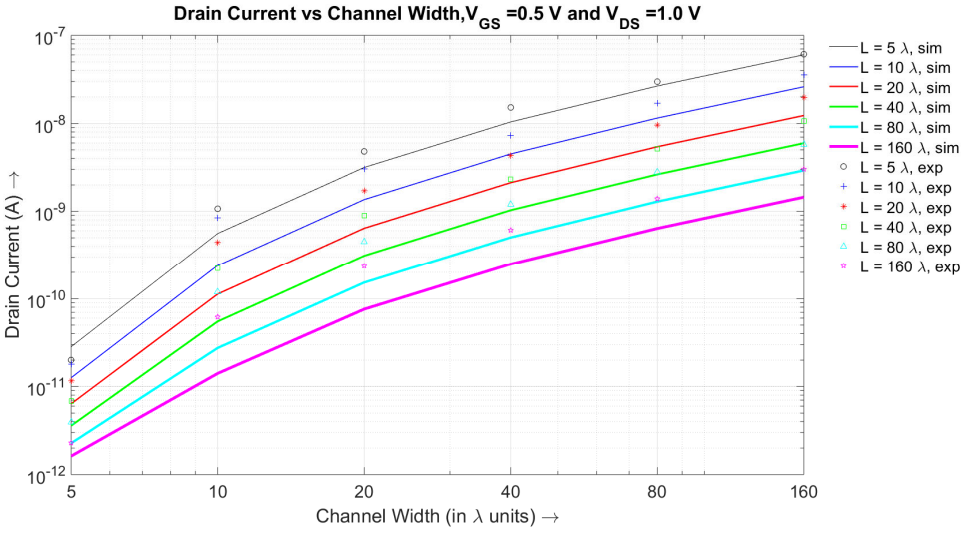


(b)

Fig. B.16. Showing drain current vs channel length plots for subthreshold operation taken from simulation and experiment data. (a) drain current vs channel length. (b) drain current vs channel length, loglog plot.



(a)



(b)

Fig. B.17. Showing drain current vs channel width plots for subthreshold operation taken from simulation and experiment data. (a) drain current vs channel width. (b) drain current vs channel width, loglog plot.

Appendix-C: Characterization of Floating Gate PMOS Transistor Injection and Tunneling Currents

C.1 Background

The aim of these experiments was to test topologies that could be used in learning circuits. Different circuits were designed to test injection and tunneling. The photomicrograph of the chip is shown in Fig. C.1. Four test structures from the chip are presented here: three topologies to test injection and one topology to test tunneling. Additionally, parameter fitting was done on the collected data to create a simulation model for SPICE based on [78].

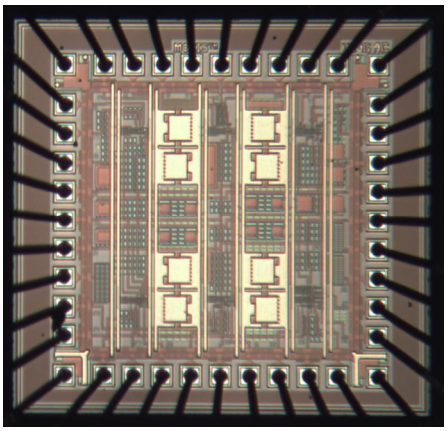


Fig. C.1. Photomicrograph of the chip with floating gate transistor test structures.

C.2 Injection Experiments

C.2.1 Using NFET at the Drain of the Floating Gate Transistor

In this technology, injection is observed at source-drain voltage levels lower than V_{DD}, so it was possible to trigger injection by connecting the drain of the floating gate transistor to ground by pulling down the drain terminal of the transistor using an

NFET. The topology is shown in Fig. C.2 and it was inspired by [75] that uses a similar test structure, but in a differential topology.

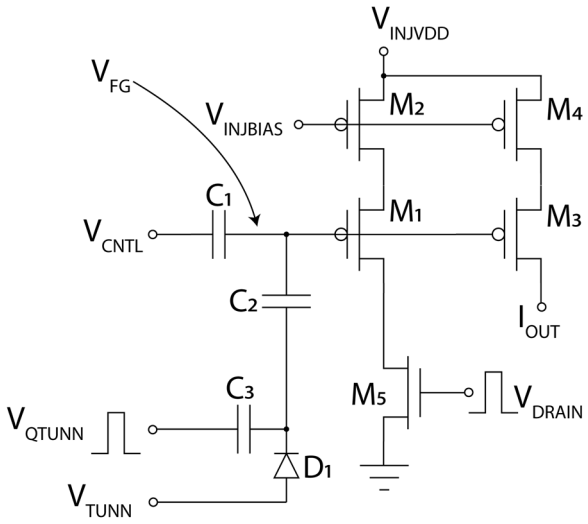


Fig. C.2. Test structure inspired by [75]. V_{CNTL} is used to set the voltage of the floating note. The floating gate transistor (M_1) current is set by fixing the M_2 bias voltage, $V_{INJBIAS}$. Injection is achieved by dropping the drain voltage of transistor M_1 low using an NFET switch.

To capture the injection trend, different $V_{INJBIAS}$ voltages were set to control the current through the floating gate transistor. In this circuit, the floating gate transistor (M_1) voltage was initialized to the same value by adjusting the V_{CNTL} voltage. These experiments were run in a succession and then V_{CNTL} was used to raise the value of the floating node. Injection was done by sending a pulse train for a fixed time on the NFET transistor gate. A pulse train of 6 ms period with a 50% duty cycle was used. Injection data was recorded after every 1 minute. For lower $V_{INJBIAS}$ levels, the source current is greater, therefore the injection rate is higher. The injection current increases exponentially as the floating gate voltage decreases. This is shown in Fig. C.3.

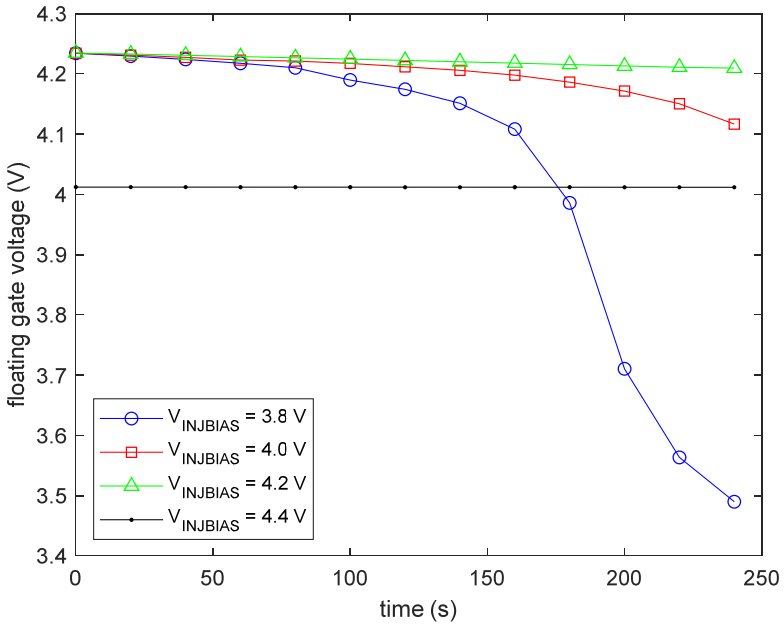


Fig. C.3. Drop in floating gate voltage for four different source current settings ($V_{INJBIAS}$ = 3.8, 4.0, 4.2 and 4.4 V). A pulse train of 9 ms with 33% duty cycle is applied for 1 minute to M_5 transistor. The injection rate increases with rising source current. NOTE: For the $V_{INJBIAS}$ = 4.4 V setting, the initial voltage on the floating node was lower.

C.2.2 Using a Charge Pump at the Drain of the Floating Gate Transistor

The benefit of using this circuit over the previous one was that the drain voltage of the transistor M1 could be made much lower than 0 V. This is based on [76] and it offers more headroom for injection and it can handle lower floating gate voltages. A drawback is that the current causes the negative going pulse to charge back up and reduces the effective time for injection.

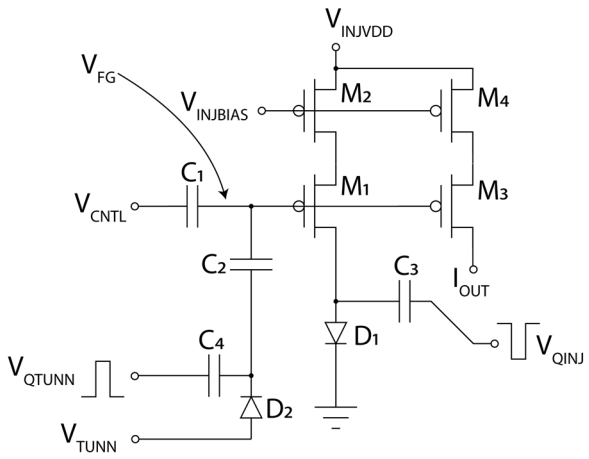


Fig. C.4. Test structure based on [76]. V_{CNTL} is used to set the voltage of the floating note. The floating gate transistor (M_1) current is set by fixing the M_2 bias voltage, $V_{INJBIAS}$. Injection is achieved by dropping the drain voltage of transistor M_1 low using a charge pump with diode D_1 and capacitor C_3. V_{QINJ} pulse is used to control injection time. C_2 is the tunneling capacitor. Capacitor C_4 and diode D_2 are used to raise the tunneling voltage.

To capture the injection trend, different $V_{INJBIAS}$ voltages were set to control the current through the floating gate transistor. In this circuit, increasing current through the transistor not only increases the injection rate, but it also reduces the effective pulse width. Up to certain limit in time for a fixed pulse-width, as the current increases, the pulse area drops linearly. Beyond that limit, the pulse area drops faster than the increase in the current. This would limit the injection charge added per pulse. This is shown in Fig. C.5.

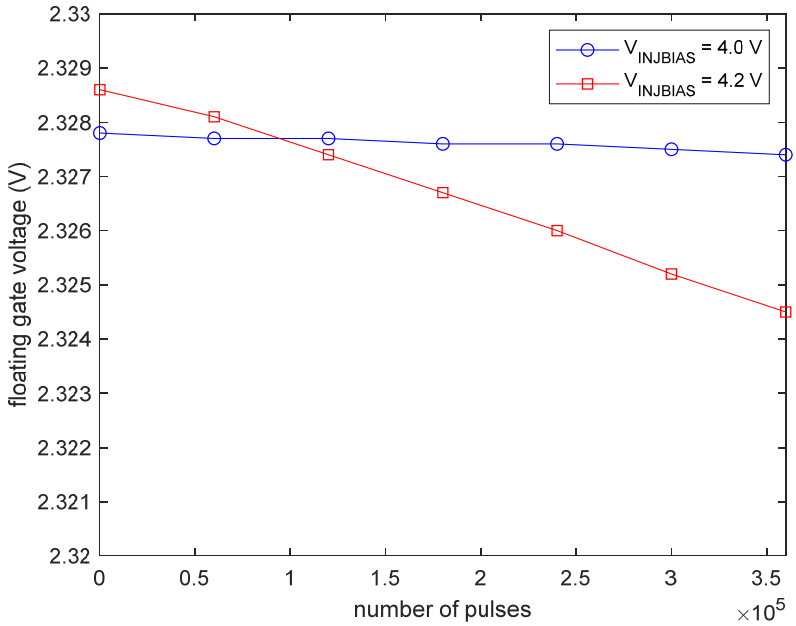


Fig. C.5. Drop in floating gate voltage for two different settings. V_{DD} = 5.0 V. An injection pulse of 18 µs is applied to V_{QINJ}. Two bias current settings are selected ($V_{INJBIAS}$ = 4.0 V and 4.2 V). The injection rate increases with rising source current, however, in this circuit, it also reduces the effective pulse width.

C.2.3 Using Feedback to Control Terminal Voltages of the Floating Gate Transistor

Injection current has a nonlinear dependence on the terminal voltages of the floating gate transistor. So, in theory, it is possible to regulate the terminal voltages to fix the injection current. This topology was based on [77], where this idea was explored.

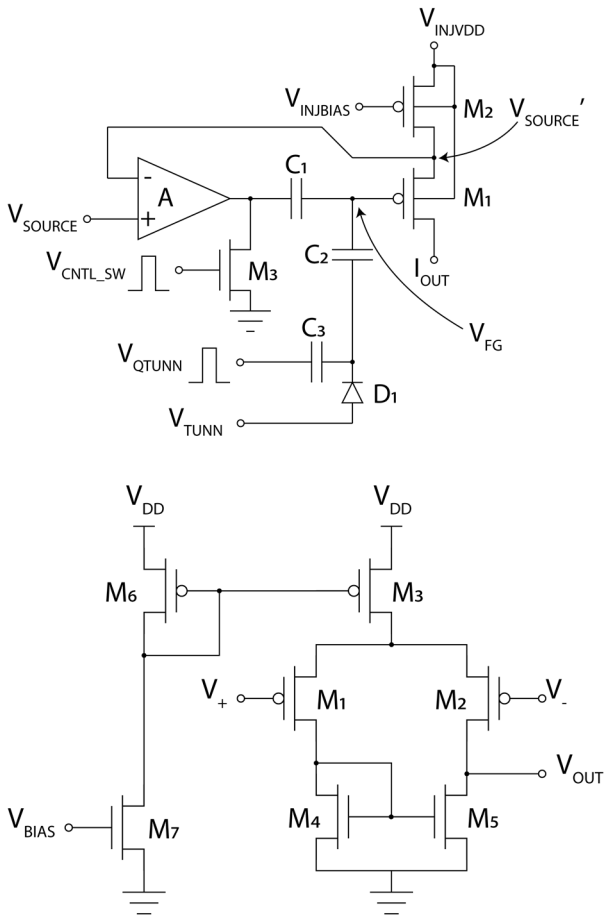


Fig. C.6. Test structure based on [77]. In the top figure, to maintain a constant rate of injection, the current through floating gate transistor M_1 is kept constant and the drain source voltage across its terminal is fixed by fixing V_{SOURCE}. NOTE: I_{OUT} is connected to ground. In the bottom figure, the feedback amplifier used to regulate the floating node voltage through the coupling capacitor C_1. C_2 is the tunneling capacitor. Capacitor C_3 and diode D_1 are used to raise the tunneling voltage.

By toggling the voltage at the gate of transistor M_3 in Fig. C.6, top figure, it is possible to do injection when the transistor M_3 is off and stop injection when the transistor M_3 is on. So, the source voltage of the floating gate transistor can be either regulated to the set voltage or it could be set by the floating node charge. Fig. C.7 is a snapshot of this experiment showing injection.

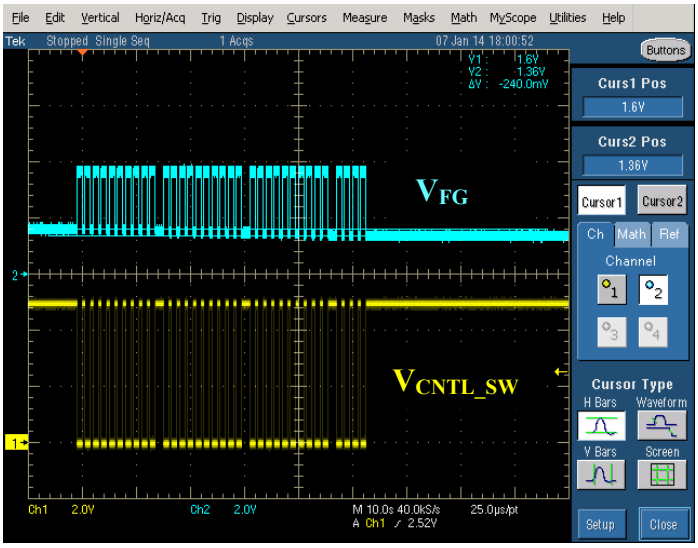


Fig. C.7. Shows the voltage difference on the floating node (read through a voltage buffer) after 34 pulses 3 ms wide are applied to the gate. V_{SD} = 5 V, V_{INJVDD} = 6 V, $V_{INJBIAS}$ = 5 V.

To determine the injection rates for difference source current settings, different $V_{INJBIAS}$ voltages were set to control the current through the floating gate transistor. The source voltage was fixed using feedback: V_{SOURCE} was set to 5 V and the drain voltage, I_{OUT} node was set to 0 V. The supply voltage V_{INJVDD} was set to 6 V. The bias currents and injection rates are reported in Table C.1 and the drop in the floating gate voltage over time is shown in Fig. C.8.

Table C.1. Injection rates for different currents

$V_{INJBIAS}$ (V)	I_{BIAS} (nA)	Injection rate (µV/ms)
5.0	351	8.4
5.1	64.6	2.7
5.2	6.14	0.4

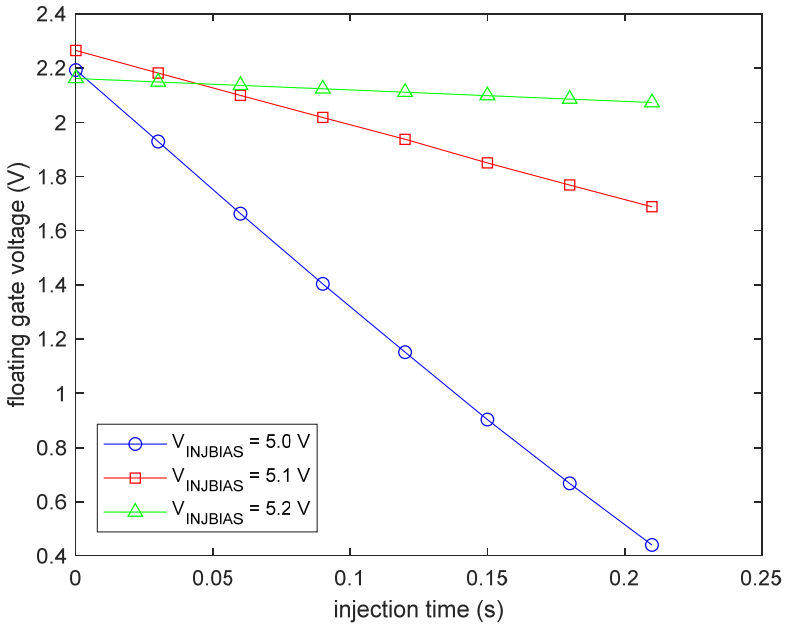


Fig. C.8. Drop in floating gate voltage for three different settings. V_{SD} = 5 V, V_{INJVDD} = 6 V and V_{INBIAS} is varied. A set of 10 pulses was applied 7 times. Each pulse has 3 ms pulse width.

C.2 Tunneling Experiment

A simple floating gate transistor was designed with a high voltage pad on the chip to use it for tunneling experiments.

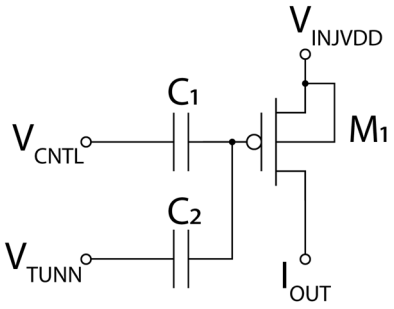


Fig. C.9. This test structure was meant to extract the model parameters for tunneling and injection experiments. V_{CNTL} is used to set the voltage of the floating note. Tunneling is achieved in this test circuit by toggling V_{TUNN} directly between 0 and tunneling voltage.

In Fig. C.10, a tunneling experiment is shown, where V_{TUNN} was toggled between 0-13 V and then 0-14 V. The initial voltage was adjusted to start with the same effective oxide voltage. For this experiment, V_{INJVDD} and I_{OUT} were both set to V_{DD} = 5 V.

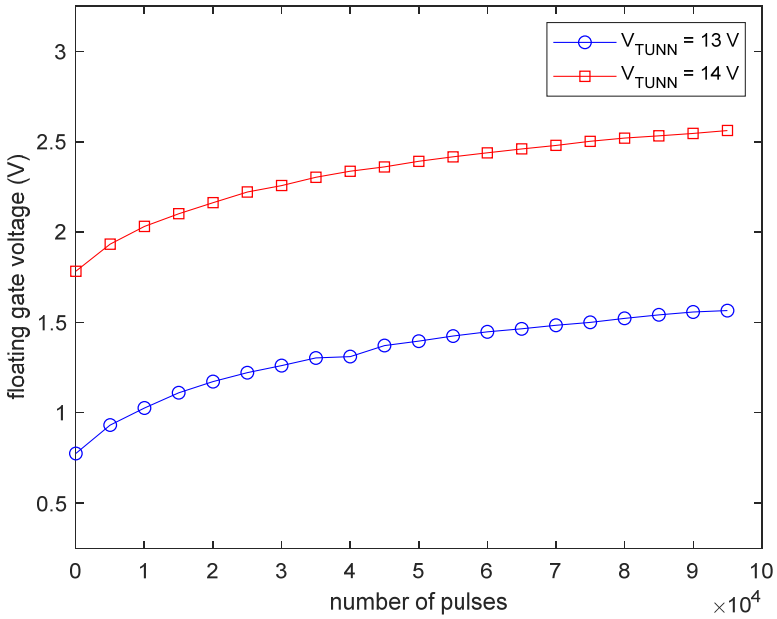


Fig. C.10. Rise in floating gate voltage recorded after every 30 seconds. Each 30 second, a set of 5,000 pulses are sent. This is a 6 ms pulse train with 50% duty cycle. The tunneling experiment is started almost at the same effective oxide voltage difference, so the curves for tunneling voltage of 13 V and 14 V look identical and shifted.

C.4 Parameter Fitting for Injection and Tunneling

The models are based on [78]. The data was collected from two different test structures. Additionally, the models were reused in the simulation to verify if the floating gate voltage matched the static tunneling and injection response. Curve fitting was done on MATLAB ® using the nlinfit() function (part of statistics and machine learning toolbox) and the figures are shown in Fig. C.11. The simulation testbench and responses are shown in Fig. C.12 and Fig. C.13. For reference, the parameter values

were compared to those published in [78]. The parameter values are given in Table C.2 and C.3.

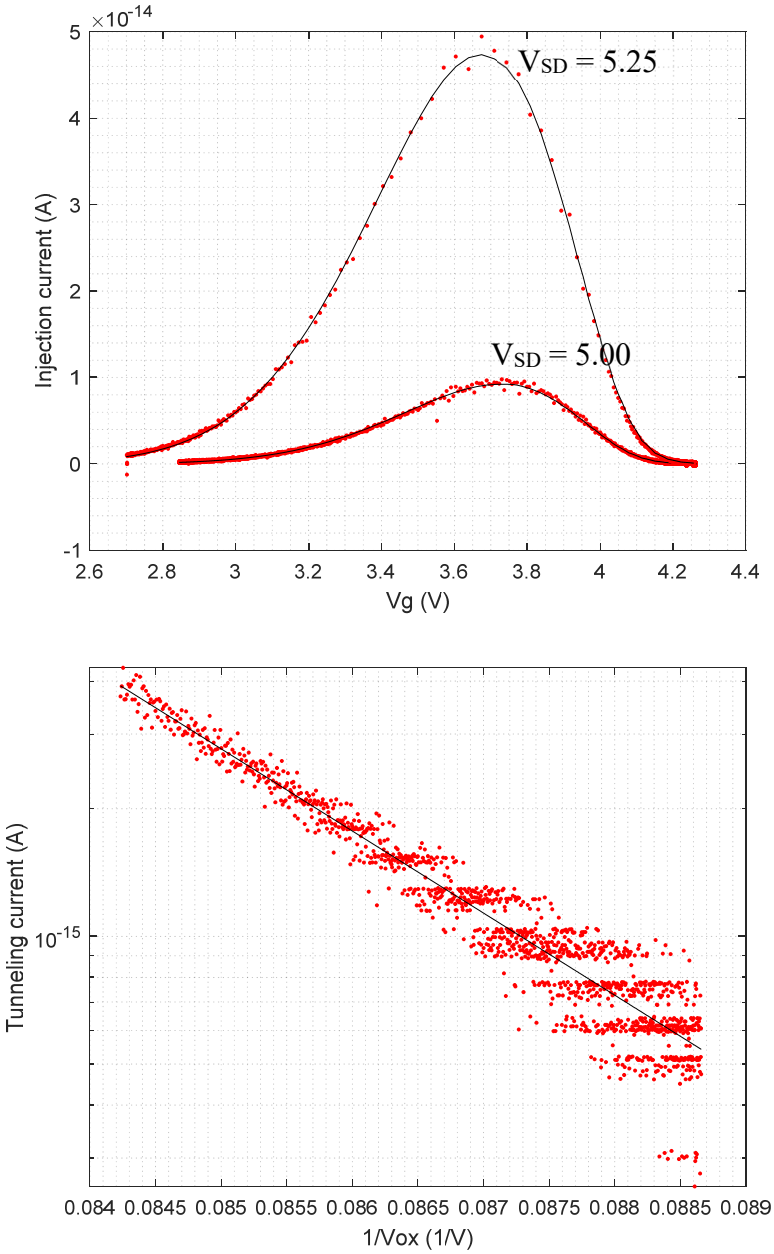


Fig. C.11. Top figure shows curve fitting on injection data, for V_{SD} = 5.00 V and 5.25 V. The injection current is plotted against the floating gate voltage. Bottom figure shows curve fitting on tunneling data for V_{TUNN} = 12. The relationship is plotted between tunneling current and oxide voltage. NOTE: The band pattern is due to noise and quantization error in the measurement close to the floating gate voltage saturation. Taking the derivative around the saturated value creates this pattern.

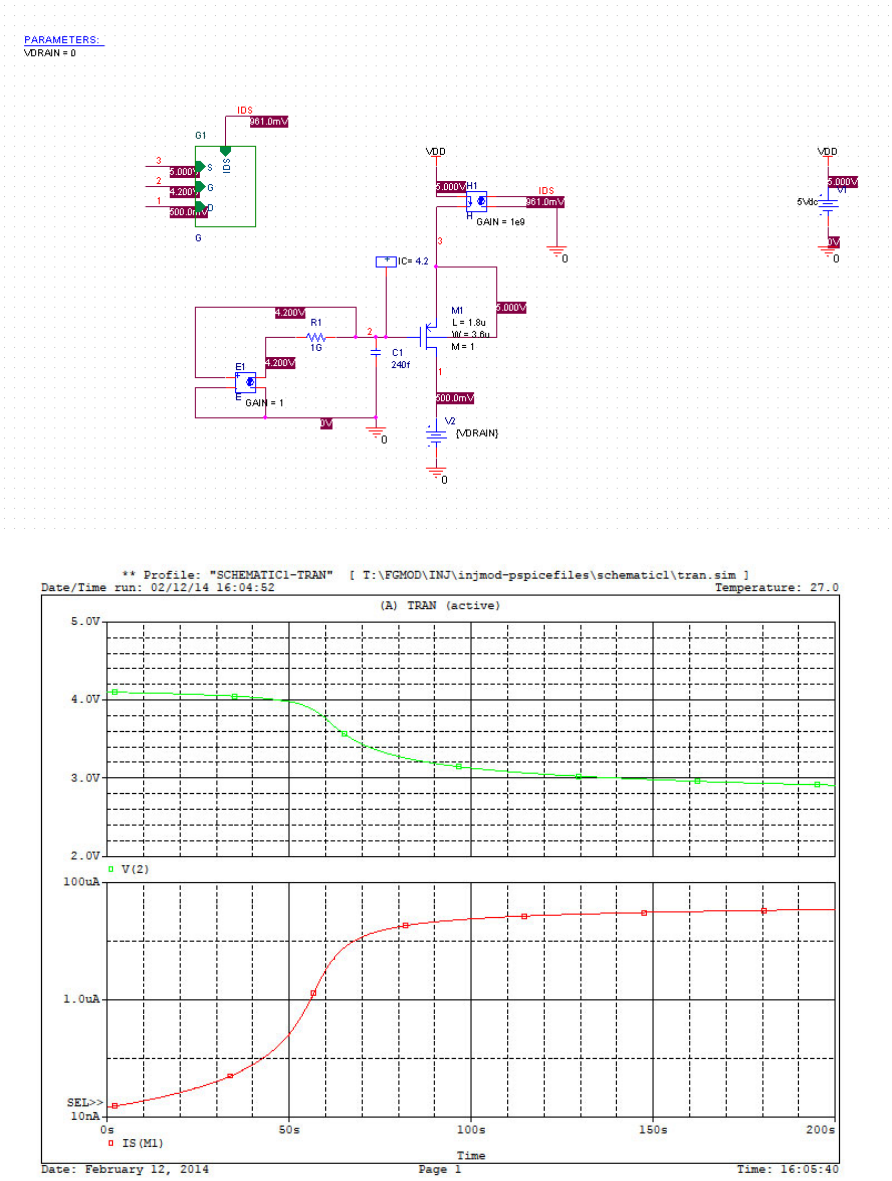


Fig. C.12. Showing injection simulation. Top figure is a testbench, where a PFET is connected between V_{DD} and 0 V. The subcircuit G_1 converts the current measured at the source and the three terminal voltages of the transistor M_1 to injection current. Bottom figure shows static injection, where the floating gate voltage and the source-drain current are shown. The floating gate node is initialized to 4.1 V.

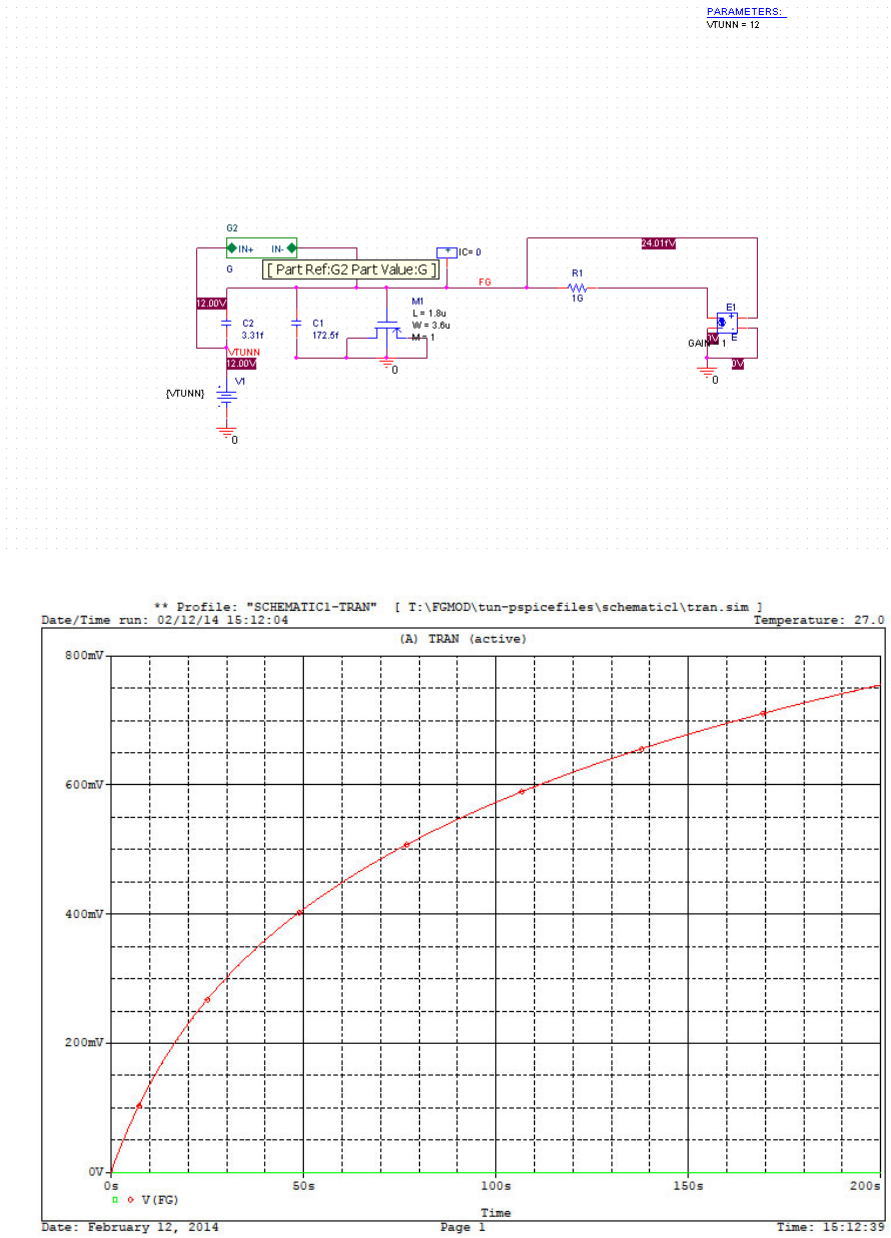


Fig. C.13. Showing tunneling simulation. Top figure is a testbench, where a PFET M_1 drain, source and bulk terminals are connected to ground. The subcircuit G_2 converts the voltage across the tunnel oxide capacitor to tunneling current that is connected in parallel to the capacitor. Figure at the bottom shows floating gate voltage increasing due to tunneling current. The floating gate voltage is initialized to 0 V.

Table C.2. Comparison of injection current parameters

	0.35 µm technology (from [78])	0.5 µm technology (this work)
Pre-exponent	1.3E-5 A	0.74E-6 A
Gate-drain factor	-155.75 V^2	-322.40 V^2
Gate-drain constant	0.702 V	1.22 V
Source-drain factor	1 V^{-1}	1.51 V^{-1}

Table C.3. Comparison of tunneling current parameters

	0.35 µm technology (from [78])	0.5 µm technology (this work)
Pre-exponent	9.35E8 A/μm^2	79.76 A/1.44 μm^2
Oxide voltage factor	368.04 V	445.88 V

C.5 Conclusion

Four circuits were designed and tested for investigating their use in learning circuits. Floating gate transistors were programmed using tunneling and injection. The data collected from the tunneling and injection experiments were used to find the parameters for SPICE models of tunneling and injection currents that could be used in the simulation of floating gate transistor-based learning circuits.

Bibliography

[1] J. O'Keefe and L. Nadel, *The Hippocampus as a Cognitive Map*. Oxford, United Kingdom: Clarendon Press, 1978.

[2] S. Koul and T. K. Horiuchi, "Waypoint Path Planning With Synaptic-Dependent Spike Latency," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 66, no. 4, pp. 1544–1557, Apr. 2019.

[3] M. Moser and E. I. Moser, "Where Am I? Where Am I Going?," *Scientific American*, vol. 314, no. 1, Springer Nature, pp. 26–33, Jan-2016.

[4] R. U. Muller and J. L. Kubie, "The effects of changes in the environment on the spatial firing of hippocampal complex-spike cells," *J. Neurosci.*, vol. 7, no. 7, pp. 1951–1968, Jul. 1987.

[5] K. Louie and M. A. Wilson, "Temporally structured replay of awake hippocampal ensemble activity during rapid eye movement sleep," *Neuron*, vol. 29, no. 1, pp. 145–156, 2001.

[6] B. E. Pfeiffer and D. J. Foster, "Hippocampal place-cell sequences depict future paths to remembered goals," *Nature*, vol. 497, pp. 74–79, May 2013.

[7] S. Koziol, S. Brink, and J. Hasler, "A Neuromorphic Approach to Path Planning Using a Reconfigurable Neuron Array IC," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 22, no. 12, pp. 2724–2737, Dec. 2014.

[8] J. L. Krichmar, "Path planning using a spiking neuron algorithm with axonal delays," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 1219–1226.

[9] L. Dollé, D. Sheynikhovich, B. Girard, R. Chavarriaga, and A. Guillot, "Path

planning versus cue responding: a bio-inspired model of switching between navigation strategies," *Biol. Cybern.*, vol. 103, no. 4, pp. 299–317, 2010.

[10] U. M. Erdem and M. E. Hasselmo, "A goal-directed spatial navigation model using forward trajectory planning based on grid cells," *Eur. J. Neurosci.*, vol. 35, no. 6, pp. 916–931, 2012.

[11] F. Ponulak and J. J. Hopfield, "Rapid, parallel path planning by propagating wavefronts of spiking neural activity.," *Front. Comput. Neurosci.*, vol. 7, no. July, p. 98, 2013.

[12] M. J. Milford, G. F. Wyeth, and D. Prasser, "RatSLAM: A Hippocampal Model for Simultaneous Localization and Mapping," in *Proceeding of the 2004 IEEE international Conference on Robotics & Automation*, 2004, pp. 403–408.

[13] N. Cuperlier, M. Quoy, and P. Gaussier, "Neurobiologically inspired mobile robot navigation and planning," *Front. Neurorobot.*, vol. 1, no. 3, pp. 1–15, Nov. 2007.

[14] K. D. Fischl, K. Fair, W. Y. Tsai, J. Sampson, and A. Andreou, "Path planning on the TrueNorth neurosynaptic system," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017, pp. 1–4.

[15] M. O. Franz and H. A. Mallot, "Biomimetic robot navigation. Robotics and autonomous Systems.," *Rob. Auton. Syst.*, vol. 30, pp. 133–153, 2000.

[16] O. Trullier, S. I. Wiener, A. Berthoz, and J.-A. Meyer, "Biologically Based Artificial Navigation Systems: Review and Prospects," *Prog. Neurobiol.*, vol. 51, no. 5, pp. 483–544, 1997.

[17] S. Stewart, A. Jeewajee, T. J. Wills, N. Burgess, and C. Lever, "Boundary coding

in the rat subiculum," *Philos. Trans. R. Soc. B Biol. Sci.*, vol. 369, no. 1635, Feb. 2014.

[18] T. Hartley, C. Lever, N. Burgess, and J. O'Keefe, "Space in the brain: how the hippocampal formation supports spatial cognition.," *Philos. Trans. R. Soc. Lond. B. Biol. Sci.*, vol. 369, no. 1635, pp. 1–18, Feb. 2014.

[19] F. Chersi and N. Burgess, "The Cognitive Architecture of Spatial Navigation: Hippocampal and Striatal Contributions," *Neuron*, vol. 88, no. 1, pp. 64–77, Oct. 2015.

[20] A. Arleo and W. Gerstner, "Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity," *Biol. Cybern.*, vol. 83, no. 3, pp. 287–299, Aug. 2000.

[21] A. V Samsonovich and G. A. Ascoli, "A simple neural network model of the hippocampus suggesting its pathfinding role in episodic memory retrieval.," *Learn. Mem.*, vol. 12, no. 2, pp. 193–208, 2005.

[22] T. M. Massoud and T. K. Horiuchi, "A neuromorphic VLSI head direction cell system," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 58, no. 1, pp. 150–163, Jan. 2011.

[23] T. M. Massoud and T. K. Horiuchi, "A neuromorphic VLSI grid cell system," in *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*, 2012, pp. 2421–2424.

[24] T. M. Massoud, "Modeling the bat spatial navigation system: A neuromorphic VLSI approach," University of Maryland College Park, 2012.

[25] R. Ivey, D. Bullock, and S. Grossberg, "A neuromorphic model of spatial

lookahead planning," *Neural Networks*, vol. 24, no. 3, pp. 257–266, Apr. 2011.

[26] A. Khajeh-Alijani, R. Urbanczik, and W. Senn, "Scale-Free Navigational Planning by Neuronal Traveling Waves," *PLoS One*, vol. 10, no. 7, pp. 1–15, Jul. 2015.

[27] W. Gerstner and L. F. Abbott, "Learning navigational maps through potentiation and modulation of hippocampal place cells," *J. Comput. Neurosci.*, vol. 4, no. 1, pp. 79–94, 1997.

[28] H. S. Wan, D. S. Touretzky, A. D. Redish, J. Elman, and A. Weigend, "Towards a Computational Theory of Rat Navigation," *Connect. Model. Summer Sch.*, pp. 11–19, 1994.

[29] M. A. Brown and P. E. Sharp, "Simulation of spatial learning in the Morris water maze by a neural network model of the hippocampal formation and nucleus accumbens," *Hippocampus*, vol. 5, no. 3, pp. 171–188, 1995.

[30] H. A. Mallot, M. O. Franz, B. Schölkopf, and H. H. Bülthoff, "The view-graph approach to visual navigation and spatial memory," in *Proceedings of the 7th International Conference on Artificial Neural Networks ICANN97, Lausanne, Switzerland*, 1997.

[31] N. Burgess, J. G. Donnett, K. J. Jeffery, and J. O'Keefe, "Robotic and neuronal simulation of the hippocampus and rat navigation," *Philos. Trans. R. Soc. London B Biol. Sci.*, vol. 352, no. 1360, pp. 1535–1543, Oct. 1997.

[32] N. Burgess, M. Recce, and J. O'Keefe, "A model of hippocampal function," *Neural Networks*, vol. 7, no. 6–7, pp. 1065–1081, 1994.

[33] D. J. Foster, R. G. Morris, and P. Dayan, "A model of hippocampally dependent

navigation, using the temporal difference learning rule.," *Hippocampus*, vol. 10, no. 1, pp. 1–16, 2000.

[34] O. Trullier and J.-A. Meyer, "Animat navigation using a cognitive graph.," *Biol. Cybern.*, vol. 83, pp. 271–285, 2000.

[35] L.-E. Martinet, D. Sheynikhovich, K. Benchenane, and A. Arleo, "Spatial Learning and Action Planning in a Prefrontal Cortical Network Model," *PLoS Comput. Biol.*, vol. 7, no. 5, pp. 1–21, 2011.

[36] D. J. Foster and M. A. Wilson, "Reverse replay of behavioural sequences in hippocampal place cells during the awake state," *Nature*, vol. 440, pp. 680–683, Mar. 2006.

[37] M. P. Karlsson and L. M. Frank, "Awake replay of remote experiences in the hippocampus," *Nat. Neurosci.*, vol. 12, no. 7, pp. 913–918, Jul. 2009.

[38] W. E. Skaggs, B. L. McNaughton, M. A. Wilson, and C. A. Barnes, "Theta Phase Precession in Hippocampal Neuronal Populations and the Compression of Temporal Sequences," *Hippocampus*, vol. 6, no. 2, pp. 149–1, 1996.

[39] A. K. Lee and M. A. Wilson, "Memory of sequential experience in the hippocampus during slow wave sleep," *Neuron*, vol. 36, no. 6, pp. 1183–1194, 2002.

[40] F. Stella, P. Baracskay, J. O'Neill, and J. Csicsvari, "Hippocampal Reactivation of Random Trajectories Resembling Brownian Diffusion," *Neuron*, vol. 102, no. 2, pp. 450–461, Apr. 2019.

[41] T. K. Horiuchi, "A spike-latency model for sonar-based navigation in obstacle

fields," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 56, no. 11, pp. 2393–2401, Nov. 2009.

[42] B. Fritzke, "Growing cell structures—A self-organizing network for unsupervised and supervised learning," *Neural Networks*, vol. 7, no. 9, pp. 1441–1460, 1994.

[43] R. U. Muller and J. L. Kubie, "The firing of hippocampal place cells predicts the future position of freely moving rats," *J. Neurosci.*, vol. 9, no. 12, pp. 4101–4110, 1989.

[44] C. B. Alme, C. Miao, K. Jezek, A. Treves, E. I. Moser, and M.-B. Moser, "Place cells in the hippocampus: eleven maps for eleven rooms.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 111, no. 52, pp. 18428–18435, 2014.

[45] J. J. Hopfield, "Neurodynamics of mental exploration.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 107, no. 4, pp. 1648–1653, 2010.

[46] A. H. Azizi, L. Wiskott, and S. Cheng, "A computational model for preplay in the hippocampus.," *Front. Comput. Neurosci.*, vol. 7, no. November, p. 161, 2013.

[47] V. Edvardsen, "A Passive Mechanism for Goal-Directed Navigation using Grid Cells," in *Proceedings of the European Conference on Artificial Life*, 2015, pp. 191–198.

[48] D. Bush, C. Barry, D. Manson, and N. Burgess, "Using Grid Cells for Navigation," *Neuron*, vol. 87, no. 3, pp. 507–520, Aug. 2015.

[49] A. Banino *et al.*, "Vector-based navigation using grid-like representations in artificial agents," *Nature*, vol. 557, no. 7705, pp. 429–433, 2018.

[50] V. Edvardsen, “Goal-directed navigation based on path integration and decoding of grid cells in an artificial neural network,” *Nat. Comput.*, vol. 18, no. 1, pp. 13–27, Mar. 2019.

[51] H. Blum, A. Dietmüller, M. Milde, J. Conradt, G. Indiveri, and Y. Sandamirskaya, “A neuromorphic controller for a robotic vehicle equipped with a dynamic vision sensor,” in *Robotics: Science and Systems XIII*, 2017.

[52] F. Galluppi *et al.*, “Event-based neural computing on an autonomous mobile platform,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2862–2867.

[53] R. Kreiser, M. Cartiglia, J. N. P. Martel, J.-A. Conradt, and Y. Sandamirskaya, “A Neuromorphic Approach to Path Integration: A Head-Direction Spiking Neural Network with Vision-driven Reset,” in *IEEE International Symposium on Circuits and Systems*, 2018, pp. 1–5.

[54] R. Kreiser, P. Pienroj, A. Renner, and Y. Sandamirskaya, “Pose Estimation and Map Formation with Spiking Neural Networks: towards Neuromorphic SLAM,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 1–8.

[55] A. Bofill-i-Petit and A. F. Murray, “Learning temporal correlations in biologically-inspired aVLSI,” in *Proceedings of the International Symposium on Circuits and Systems*, 2003, pp. 817–820.

[56] K. A. Boahen, “Point-to-point connectivity between neuromorphic chips using address events,” *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process.*, vol. 47, no. 5, pp. 416–434, May 2000.

[57] C. Bartolozzi and G. Indiveri, “Synaptic Dynamics in Analog VLSI,” *Neural Comput.*, vol. 19, no. 10, pp. 2581–2603, Oct. 2007.

[58] J. Barnden, S. Kaukanahalli, and D. Dharmavaratha, “Winner-Take-All Networks : Time-Based versus Activation-Based Mechanisms for Various Selection Goals,” in *Circuits and Systems, 1990., IEEE International Symposium on*, 1990, pp. 215–218.

[59] J. P. Abrahamsen, P. Hafliger, and T. S. Lande, “A time domain winner-take-all network of integrate-and-fire neurons,” *Int. Symp. Circuits Syst. (ISCAS) 2004*, vol. 5, no. 2, p. V--361--V--364, 2004.

[60] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, New Jersey: Prentice Hall, 2010.

[61] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[62] R. Siegwart, I. Nourbakhsh, and D. Scaramuzza, “Planning and Navigation,” in *Introduction to Autonomous Mobile Robots*, Second., Cambridge, MA, United States: The MIT Press, 2011, pp. 380–381.

[63] R. Glasius, “Trajectory Formation and Population Coding with Topographical Neural Networks,” Radboud University Nijmegen, 1997.

[64] S. X. Yang and M. Meng, “An Efficient Neural Network Approach to Dynamic Robot Motion Planning,” *Neural Networks*, vol. 13, pp. 143–148, 2000.

[65] U. Roth, M. Walker, A. Hilmann, and H. Klar, “Dynamic path planning with spiking neural networks,” in *International Work-Conference on Artificial Neural Networks*, 1997, pp. 1355–1363.

[66] T. Hwu, A. Y. Wang, N. Oros, and J. L. Krichmar, “Adaptive Robot Path

Planning Using a Spiking Neuron Algorithm With Axonal Delays," *IEEE Trans. Cogn. Dev. Syst.*, vol. 10, no. 2, pp. 126–137, 2018.

[67] E. C. Tolman and C. H. Honzig, "'Insight' in rats," *Univ. Calif. Publ. Psychol.*, vol. 4, pp. 215–232, 1930.

[68] A. Gorchetchnikov and M. E. Hasselmo, "A biophysical implementation of a bidirectional graph search algorithm to solve multiple goal navigation tasks," *Conn. Sci.*, vol. 17, no. 1–2, pp. 145–164, Mar. 2005.

[69] S. Koul and T. K. Horiuchi, "Path planning by spike propagation," in *IEEE Biomedical Circuits and Systems Conference*, 2015, pp. 1–4.

[70] S. Koul and T. K. Horiuchi, "Effect of synaptic charge convergence on path planning over a neural network," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017, pp. 1–4.

[71] F. Mirus, C. Axenie, T. C. Stewart, and J.-A. Conradt, "Neuromorphic sensorimotor adaptation for robotic mobile manipulation: From sensing to behaviour," *Cogn. Syst. Res.*, vol. 50, pp. 52–66, 2018.

[72] H. Abebe and V. Tyree, "BSIM3v3.1 Model Parameters Extraction and Optimization", October 2000, The MOSIS Service, USC-ISI.

[73] Y. Tsividis, "Small-Dimension Effects," in *Operation and Modeling of The MOS Transistor*, Second., Oxford University Press, 1999, pp. 270–274.

[74] J. Baker, H. Li, and D. Boyce, "Current Sources and Sinks," in *CMOS Circuit Design, Layout and Simulation*, First., Prentice Hall of India Private Limited, 2005.

[75] Y. L. Wong, M. H. Cohen, and P. A. Abshire, "Differential Hot Electron

Injection in an Adaptive Floating Gate Comparator," *Analog Integr. Circuits Signal Process.*, vol. 43, no. 3, pp. 281–296, Jun. 2005.

[76] Y. L. Wong, M. H. Cohen, and P. A. Abshire, "A floating-gate comparator with automatic offset adaptation for 10-bit data conversion," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 52, no. 7, pp. 1316–1326, Jul. 2005.

[77] C. Huang, P. Sarkar, and S. Chakrabartty, "Rail-to-Rail, Linear Hot-Electron Injection Programming of Floating-Gate Voltage Bias Generators at 13-Bit Resolution," *IEEE J. Solid-State Circuits*, vol. 46, no. 11, pp. 2685–2692, Nov. 2011.

[78] K. Rahimi, C. Diorio, C. Hernandez, and M. D. Brockhausen, "A simulation model for floating-gate MOS synapse transistors," in *Proc. of 2002 IEEE International Symposium on Circuits and Systems*, 2002, vol. 2, p. II-532-II-535.