# Silicon neural networks : implementation of cortical cells to improve the artificial-biological hybrid technique

Filippo Giovanni Grassia

## To cite this version:

Filippo Giovanni Grassia. Silicon neural networks : implementation of cortical cells to improve the artificial-biological hybrid technique. Other. Université Sciences et Technologies - Bordeaux I, 2013. English. NNT : 2013BOR14748 . tel-00789406

**HAL Id: tel-00789406**

**https://theses.hal.science/tel-00789406**

Submitted on 18 Feb 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : 4748

# THÈSE

PRÉSENTÉE À

# L'UNIVERSITÉ DE BORDEAUX I

ÉCOLE DOCTORALE SCIENCES PHYSIQUE ET DE L'INGÉNIEUR

**Par Filippo GRASSIA**

POUR OBTENIR LE GRADE DE

**DOCTEUR**

SPÉCIALITÉ : Électronique

# Silicon neural networks: implementation of cortical cells to improve the artificial-biological hybrid technique

Soutenue le 7 janvier 2013

Après avis de :

| | | |
|---|---|---|
| M. Frederic ALEXANDRE | Directeur de recherche | INRIA Bordeaux |
| M. Stephane BINCZAK | Professeur des universités | Université de Bourgogne |

Devant la commission d'examen formée de :

| | | | |
|---|---|---|---|
| M. Frederic ALEXANDRE | Directeur de recherche | INRIA Bordeaux | Rapporteur |
| M. Stephane BINCZAK | Professeur des universités | Université de Bourgogne | Rapporteur |
| M. Alain DESTEXHE | Directeur de recherche | CNRS Gif-sur-Yvette | Examinateur |
| M. Claude PELLET | Professeur des universités | Université Bordeaux 1 | Président |
| M. Sylvain SAIGHI | Maître de conférences (HDR) | Université Bordeaux 1 | Directeur de thèse |
| M. Timothee LEVI | Maître de conférences | Université Bordeaux 1 | Co-directeur de thèse |

-2013-

*A mamma e papà.*

# Acknowledgments

This PhD thesis was supported by funding under the Seventh Research Framework Program of the European Union (FP7 - PEOPLE - ITN - 2008) under the Grant n. 237955 (FACETS - ITN).

Je suis heureux de pouvoir ainsi exprimer ma sincère gratitude à tous ceux qui m'ont accompagné pendant mes trois années de doctorat pour mener à bien ces travaux.

Je remercie tout d'abord les Professeurs Claude Pellet et Pascal Fouillat pour m'avoir accueilli au sein du laboratoire IMS pendant la réalisation de ces travaux de thèse.

Merci également au Professeur Sylvie Renaud, pour m'avoir accordé cette opportunité d'effectuer ces travaux de thèse au sein de son groupe de recherche et de m'avoir introduit à une large communauté scientifique nationale et internationale du domaine des neurosciences. Un grand merci aussi pour la bonne ambiance qu'elle veille à diffuser entre les membres du groupe.

Je remercie vivement le Docteur Sylvain Saïghi, Maître de conférences, pour m'avoir offert ce sujet de recherche; il m'a soutenu en tant que directeur de thèse et surtout fait confiance durant ces trois années de travaux.

Je remercie tout aussi vivement le Docteur Timothée Levi, Maître de conférences, pour m'avoir aidé à clarifier mes idées dès lors que j'ai commencé à communiquer mes travaux à la communauté scientifique. Je retiendrais surtout son soutien dans toutes les situations bonnes ou mauvaises qui se sont présentées lors de cette thèse.

Je tiens à remercier particulièrement le Professeur Stéphane Binczak d'avoir accepté de rapporter sur mes travaux de thèse. Sa relecture minutieuse et la pertinence de ses remarques m'ont permis d'améliorer notamment ce manuscrit.

Je remercie aussi le Docteur Frédéric Alexandre, Directeur de recherche à l'INRIA, d'avoir accepté d'être rapporteur. Grâce à ses questions pertinentes et constructives, j'ai pu mettre davantage en valeur les contributions apportées par mon travail.

Un grand merci aussi au Docteur Alain Destexhe, directeur de recherche au CNRS, pour l'intérêt qu'il a porté à mes travaux et pour avoir accepté de faire partie de mon jury de thèse. Ses précieuses publications scientifiques ont été une référence pour cette thèse.

Je remercie vivement le Professeur Claude Pellet d'avoir accepté de présider ce jury de thèse.

I am greatly indebted to professor Takashi Kohno (Institute of Industrial Science, University of Tokyo) for his valuable scientific collaboration.

I am really grateful to professor René Schüffny (Technische Universität Dresden) and the doctor Achim Graupner, for welcoming me at the Institute of Circuits and System and ZMDI (Dresden-based semiconductor company). I have been working there in a stimulating environment.

I am truly and deeply indebted to so many people in the FACETS community that there is no way to acknowledge them all, or even any of them properly. I offer my sincerest apologies to anyone: Bjoern Kindler, Javier Baladron, Giacomo Benvenuti, Pierre Berthet, Johannes Bill, Marco Brigham, Alejandro Bujan, Love Cederstroem, Jing Fang, Diego Fasoli, Bernhard Kaplan, Mina Khoei, Ajith Padmanabhan, Radoslav Prahov, Marc Olivier Schwartz, Carlos Stein, Vasilis Thanasoulis, Tahir Uddin, Friedemann Zenke.

J'ai une pensée amicale à toutes les personnes travaillant dans les équipes Elibio et AS2N pour les diverses choses qu'elle m'a apportées et notamment les traditionnels goûters : Noelle, Yannick, Jean et Gilles. Et bien sûr, j'en ai une autre pour les doctorants, ingénieurs et stagiaires qui ont partagé mon quotidien : Adam, Adeline, Agostino, Bilel, François, Florian, Guillaume, Gwendal, Jean-Baptiste, Kais, Matthieu, Vinh, Youssef.

Enfin, j'adresse mes remerciements les plus chaleureux à ma maman et mon papa qui m'ont soutenu moralement durant ces trois ans et ma copine Florinda, avec qui je partage ma vie depuis sept ans.

# Abstract

This work has been supported by the European FACETS-ITN project. Within the framework of this project, we contribute to the simulation of cortical cell types (employing experimental electrophysiological data of these cells as references), using a specific VLSI neural circuit to simulate, at the single cell level, the models studied as references in the FACETS project. The real-time intrinsic properties of the neuromorphic circuits, which precisely compute neuron conductance-based models, will allow a systematic and detailed exploration of the models, while the physical and analog aspect of the simulations, as opposed the software simulation aspect, will provide inputs for the development of the neural hardware at the network level. The second goal of this thesis is to contribute to the design of a mixed hardware-software platform (PAX), specifically designed to simulate spiking neural networks. The tasks performed during this thesis project included: 1) the methods used to obtain the appropriate parameter sets of the cortical neuron models that can be implemented in our analog neuromimetic chip (the parameter extraction steps was validated using a bifurcation analysis that shows that the simplified HH model implemented in our silicon neuron shares the dynamics of the HH model); 2) the fully customizable fitting method, in voltage-clamp mode, to tune our neuromimetic integrated circuits using a metaheuristic algorithm; 3) the contribution to the development of the PAX system in terms of software tools and a VHDL driver interface for neuron configuration in the platform. Finally, it also addresses the issue of synaptic tuning for future SNN simulation.

**Keywords**

Computational Neurosciences, Hodgkin and Huxley model, bifurcation analysis, metaheuristic algorithms, neuromorphic systems, spiking neural networks.

# Résumé

«Réseau de neurones in silico : contribution au développement de la technique hybride pour les réseaux corticaux»

Ces travaux ont été menés dans le cadre du projet européen FACETS-ITN. Ils apportent une contribution à la simulation de cellules corticales grâce à des données expérimentales d'électrophysiologie prises comme références et d'un circuit intégré neuromorphique comme simulateur. Les propriétés intrinsèques temps réel de nos circuits neuromorphiques à base de modèles à conductance, autorisent une exploration détaillée des différents types de neurones. L'aspect analogique des circuits intégrés permet le développement d'un simulateur matériel temps réel à l'échelle du réseau.

**Objectifs**

L'objectif de ce travail de thèse est double. II consiste, dans un premier temps, à développer les modèles des cellules corticales (les plus usuelles en neurosciences computationnelles) qui puissent être implémentées dans le circuit intégré neuromorphique conçu au sein de notre équipe. Ces paramètres implémentés grâce à une technique de réglage automatique à base de métaheuristique, les susdits circuits pourront alors être configurés afin de créer un réseau de neurones déterminé. Dans un deuxième temps, une partie de ce travail de thèse a été consacrée à l'amélioration de la couche logicielle de la plateforme mixte hardware/software «PAX». Cette amélioration permet la configuration du réseau de neurone, la gestion des couplages synaptiques et l'acquisition de l'activité des neurones électroniques. Le manuscrit de thèse s'articule autour de ces deux objectifs et se compose

de quatre chapitres.

**Chapitre 1**

Après un rappel des différentes étapes ayant participé au développement de la compréhension des phénomènes biologiques s'exprimant dans les fibres nerveuses et plus particulièrement au niveau de l'axone, le premier chapitre présente les principaux modèles mathématiques permettant de reproduire les principales activités neuronales au niveau cellulaire. II met en avant une représentation ponctuelle des neurones, conséquence d'un besoin de réduction du coût computationnel pour aborder l'étude de réseaux de neurones. Le choix d'un modèle compatible avec la représentation choisie parmi d'autres est alors discuté selon la plausibilité biologique correspondante. II en ressort que le modèle d'Hodgkin et Huxley est le meilleur candidat selon le critère de plausibilité biologique mais engendre un coût computationnel conséquent, ce qui légitime l'utilisation de circuits analogiques microélectroniques pour réaliser des simulations respectant les contraintes de temps biologique. Cette problématique technologique est développée dans la deuxième partie de ce chapitre. Une revue des systèmes électroniques neuromorphiques est menée et met en avant les avantages et inconvénients propres à chaque système en termes de niveau de description, de plausibilité biologique, de taille de réseaux de neurones atteignable. Elle permet de mieux apprécier la plateforme neuromorphique développée au sein de l'équipe.

**Chapitre 2**

Le chapitre deux s'intéresse aux quatre neurones corticaux les plus courants : Fast Spiking (FS), Regular Spiking (RS), Intrinsically Bursting (IB) et Low Threshold Spiking (LTS). L'un des buts principaux de cette thèse, étant de pouvoir les émuler, se pose donc ici la question d'extraire les paramètres utiles pour une implantation in silico. Pour cela, ce chapitre revient plus en détail sur le modèle de Hodgkin et Huxley (HH) et explicite les différents courants ioniques le constituant ainsi que la possibilité de personnaliser un neurone selon ses canaux ioniques via une modification du modèle des courants ioniques correspondants. Concernant les aspects électroniques du neurone in silico, il est alors précisé la méthode déployée pour obtenir l'équivalent de tels ou tels courants ioniques biologiques. II est établi que le neurone électronique est doté de paramètres variables permettant, par combinaison de modifier son comportement selon le type d'activité désirée. Ce chapitre présente également l'implantation du modèle HH dans la puce «Galway» qui sera utilisée ici et dont la simplification principale est relative à la constante de temps des termes d'activation et d'inactivation qui est déterminée constante (avec diverses répercussions, comme sur la forme du potentiel d'action). Une analyse des bifurcations permettant d'analyser les conséquences des simplifications retenues pour les modèles sur la dynamique neuronale est ensuite présentée. Une comparaison des diagrammes de bifurcation du modèle mathématique complet et de celui simplifié est alors exposée (Figure 1).

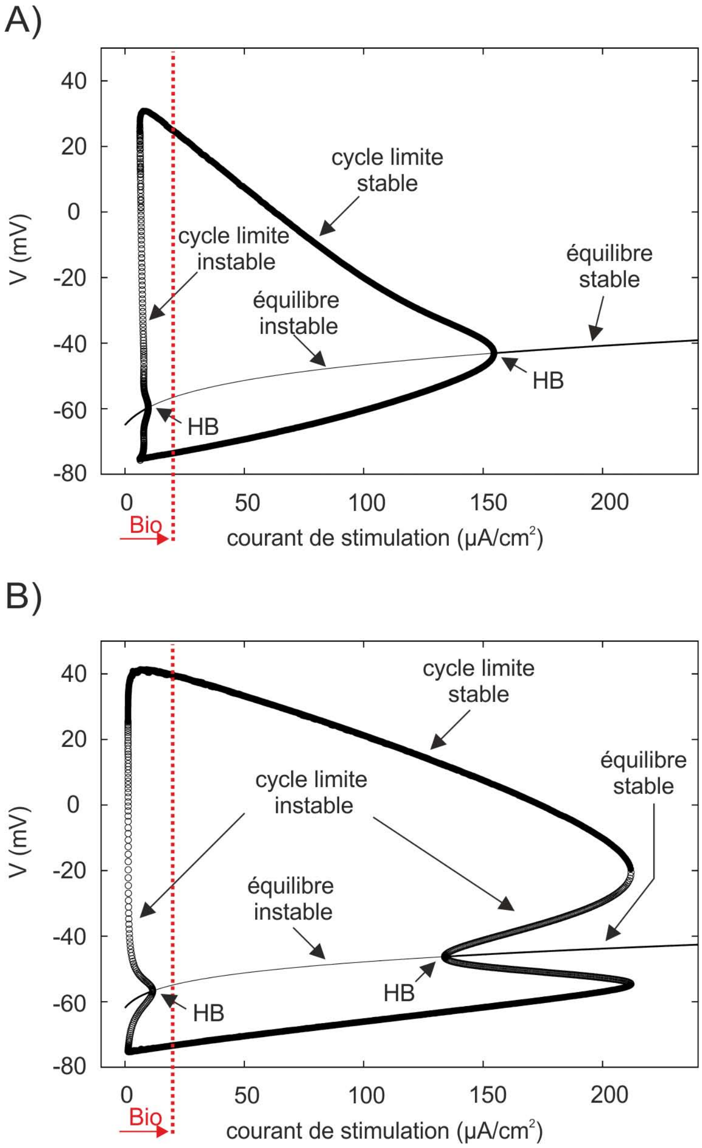


**Figure 1** – Le modèle HH (A) a la même dynamique que le modèle simplifié pour des courants de stimulations biologiques (cf. à gauche des pointillés rouges).

Ce résultat principal atteste de la similarité des dynamiques pour un même courant de stimulation biologiquement réaliste (courant de stimulation $< 20\ \mu A/cm^2$).

Ce chapitre traite aussi de l'extraction des paramètres des neurones. Il présente les valeurs des paramètres des courants ioniques à imposer pour retrouver les quatre types usuels de neurones (FS, RS, IB et LTS) présents dans les régions du néocortex et du thalamus. Des exemples d'activités temporelles sont présentés pour ces quatre classes, à partir desquels une comparaison entre modèles complet et simplifié est exposée. On retrouve des comportements similaires. Parallèlement, les valeurs des paramètres électriques à fournir pour obtenir ces comportements sur notre circuit intégré neuromimétique sont fournies sous forme de tableau pour chacun des types de neurones.

**Chapitre 3**

Le chapitre trois s'intéresse ensuite à l'aspect temps-réel de ces simulations et souligne que, en passant à la version hardware, le processus de fabrication génère des erreurs qu'il faut donc compenser en ajustant à nouveau les paramètres. Ce chapitre présente l'environnement ainsi que la méthode de paramétrage globale utilisée pour ajuster automatiquement les paramètres afin d'atteindre les dynamiques des quatre classes de neurones. Le système électronique neuromimétique est ensuite détaillé, ce qui permet de comprendre comment le modèle mathématique est transposé au sein du système. On apprend également comment est structurée la carte et quel est son rôle dans l'interfaçage (hardware et software) avec le circuit intégré neuromimétique. La méthode d'ajustement automatique utilise une métaheuristique (Figure 2) qui est l'algorithme à évolution différentielle.

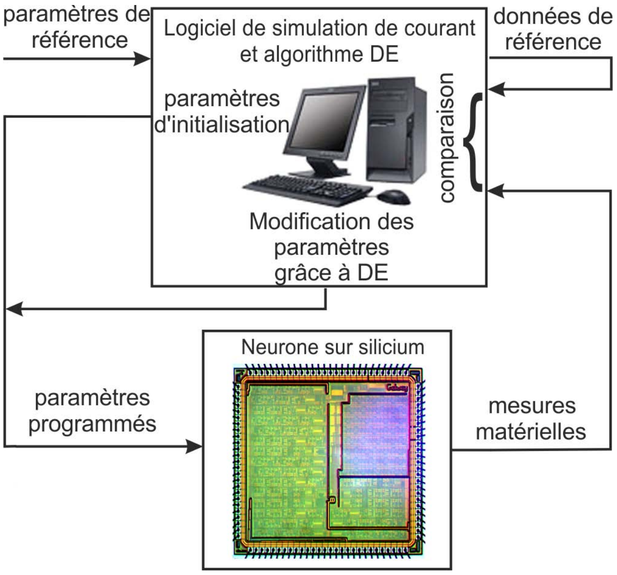

paramètres de référence
Logiciel de simulation de courant et algorithme DE
données de référence
paramètres d'initialisation
comparaison
Modification des paramètres grâce à DE
Neurone sur silicium
paramètres programmés
mesures matérielles

**Figure 2** – Diagramme du système de calibrage du neurone sur silicium.

Les comparaisons entre activités temporelles attendues et celles obtenues expérimentalement après ajustement montre que cette méthode est parfaitement adaptée (voir l'exemple pour un neurone IB Figure 3).

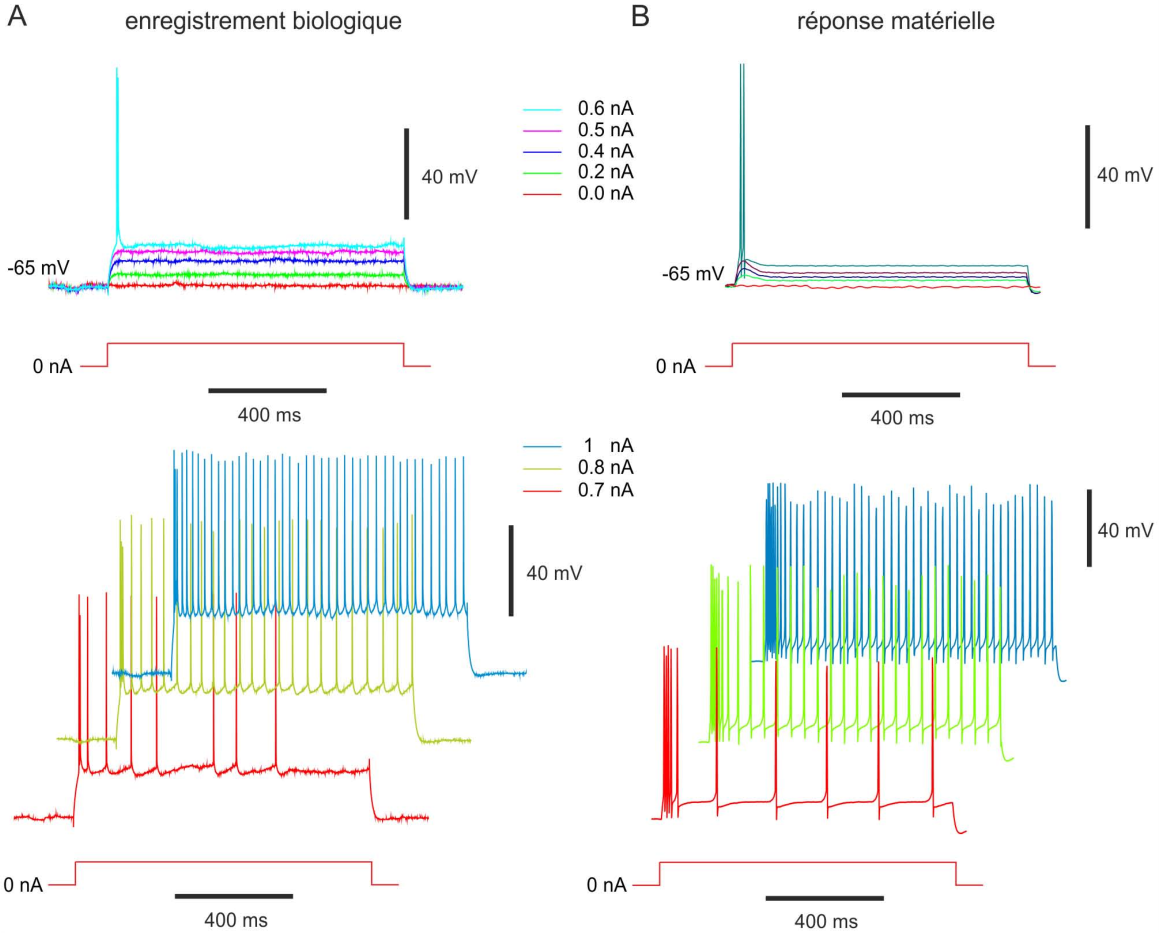


**Figure 3** – Neurone IB : Tension de membrane enregistrée sur une cellule biologique (A) et sur le circuit intégré analogique neuromorphique (B) pour différents courants de stimulation.

On peut souligner que les activités de référence proviennent d'activités biologiques et non pas de simulations basées sur un modèle mathématique. Notre implémentation dans le circuit est donc validée et elle légitime donc la plateforme réalisée.

### Chapitre 4

Le chapitre quatre s'intéresse enfin à la description de l'amélioration de la plateforme PAX. Une description des diverses couches (analogique, numérique et logicielle) est tout d'abord exposée. Le travail réalisé consiste principalement à réaliser un driver VHDL pour la configuration des neurones électroniques (Figure 4).

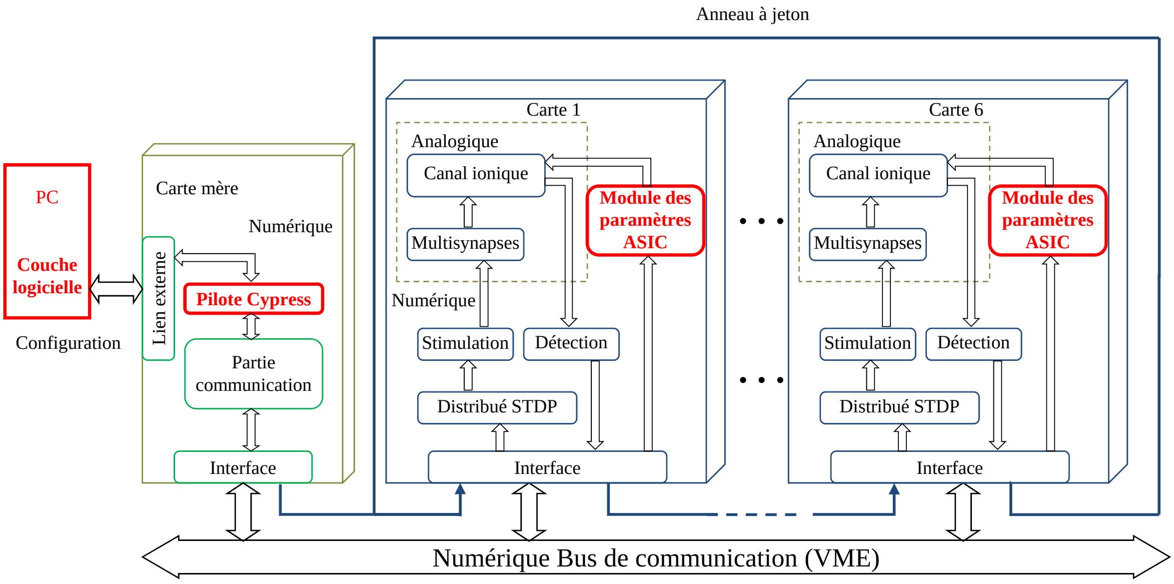


**Figure 4** – Schéma structurel du système de simulation du réseau neuronal PAX (En rouge sont représentées les fonctions développées dans ces travaux de thèse).

Des limites d'utilisation ont été identifiées durant cette étude, notamment un domaine de variation des conductances synaptiques trop réduit pour atteindre l'équivalent biologique. De même, un disfonctionnement électronique concernant le bloc synaptique a pu être décelé.

Le travail réalisé permet, dans les limites imposées par ces problèmes matériels :

- la configuration du réseau de neurone,
- la gestion des couplages synaptiques,
- l'acquisition de l'activité biomimétique des neurones électroniques.

Des exemples sont fournis afin d'illustrer les diverses étapes de simulation électronique.

### Conclusion

Nous avons abordé dans cette thèse, la validation d'un modèle simplifié de neurone biophysiquement réaliste à l'aide d'une analyse de bifurcation. Ce modèle a été implémenter grâce à une métaheuristique dans notre notre circuit intégré neuromorphique. L'activité électrique de notre circuit a été comparée avec celles enregistrées sur des cellules

biologiques. Nous affirmons ensuite que ces circuits peuvent être utilisés pour former le noyau d’une plate-forme de simulation conçue pour simuler les réseaux de neurones dans des configurations biologiquement pertinentes. Le simulateur de réseaux de neurones impulsionnels (SNN) développé par notre équipe, appelé PAX (algorithme de plasticité pour Computing System) a ensuite été détaillé. Nous avons présenté des simulations qui démontrent que la plate-forme matérielle est pleinement opérationnelle. En conclusion, nous montrons que les objectifs ont été atteints. Cependant nous avons observé quelques phénomènes qui nous ont amenés à proposer des voies d’amélioration pour l’avenir.

### Mots clés

Neurosciences computationnelles, modèle de Hodgkin Huxley, analyse de bifurcation, algorithmes métaheuristiques, systèmes neuromorphiques, réseaux de neurones.

# Contents

# Introduction

Neurosciences are one of the main research topics of this century. The human brain is said to be one of the most complex systems known to science and understanding how it works is as old a question as mankind. "The expression «computational neuroscience» reflects the possibility of generating theories of brain function in terms of the information-processing properties of structures that make up the nervous system. It implies that we ought to be able to exploit the conceptual and technical resources of computational research to help find explanations of how neural structures achieve their effects, what functions are executed by neural structures, and the nature of the states represented by the nervous system" [Churchland et al., 1993].

Over the past hundred years, biological research has accumulated an enormous amount of detailed knowledge about the structure and function of the brain. The elementary processing units in the central nervous system are neurons that are connected to each other in an intricate pattern. Neuroscientists provide biological measurements to computational neuroscientists who then propose a model for simulations, as well as for studies of the single cell or neural network dynamics. Computational neuroscientists must find compromises between two seemingly mutually exclusive requirements. The model for a single neuron must be: 1) computationally simple, yet 2) capable of producing rich firing patterns exhibited by real biological neurons. Many neuron models have been proposed. Which one to choose? The answer depends on the type of the problem. Using biophysically accurate Hodgkin-Huxley type models (HH) is computationally prohibitive, since we can only simulate a handful of neurons in real time. In contrast, using an integrate-and-fire model is computationally effective, but the model is unrealistically simple and incapable of producing the rich spiking and bursting dynamics exhibited by cortical neurons [Izhikevich, 2003]. Large scale simulations, running on conventional computing platforms, can take minutes to simulate even one second of "real" neural activity [Izhikevich and Edelman, 2008]. The use of a hardware platform is then mandatory for the biological real-time goal. In recent years, a new discipline has emerged which challenges the classical approaches to engineering and computer research: neuromorphic engineering. Neuromorphic engineering can be divided into neuromorphic modeling, reproducing neuro-physiological phenomena to increase the understanding of the nervous systems, and neuromorphic computation, which uses the neuronal properties to build neuron-like computing hardware. By combining neurophysiological principles with silicon engineering, Mahowald and Douglas produced an analog integrated circuit with the functional characteristics of real nerve cells. The analog neuromimetic integrated circuit operates in real time and consumes little power, and many "neurons" can be fabricated on a single silicon chip [Mahowald and Douglas, 1991]. The silicon neuron represents a step towards constructing artificial nervous systems that use more realistic principles of neural computation than do existing electronic neural networks.

Neuromorphic engineering proposes to fill the gap between computational neurosciences, on the one hand, and traditional engineering, on the other. Alternatives to software-based solutions [Brette et al., 2007], neuromorphic systems are often based on custom integrated circuits (IC) and systems [Misra and Saha, 2010]. A neuromorphic system can be digital, analog, or mixed.

In parallel with the development in neuroscience, microelectronic technology quickly evolves into a regime that is almost compatible with biological neural systems in terms of computational power. A hybrid neural-silicon system becomes possible and such a system will have profound implications in engineering design, from prosthetics to sensorimotor control that advance human abilities. Direct interfaces between small networks of nerve cells and synthetic devices promise to advance our understanding of neuronal function and may yield a new generation of hybrid devices that exploit the computational capacities of biological neural networks [Le Masson et al., 2002].

Hence, a hybrid neural-silicon system needs to be strongly inspired (level of detail) and linked to the actual dynamics of neuronal assemblies and be able to "talk" with their natural counterpart.

Analog neuromimetic integrated circuits and hybrid neural-silicon systems are the main research topics of the IMS BioElectronics group (Laboratoire IMS, UMR CNRS 5218, Université de Bordeaux, Talence, France).

In particular, one of the main research topics of the BioElectornics group is the development of new instrumentation tools for the exploration of the central nervous system, making use of dedicated interfaces between microelectronics and live neural networks. The technical approach defined for this research topic is to couple live large-scale neural networks and artificial neural networks embedded in analog and mixed integrated electronics and endowed with adaptive capabilities (synaptic plasticity). This hybrid coupling will use dedicated microelectrode arrays to record and electrically stimulate live neural networks, with a specific emphasis on stimulation localization. The system, which includes artificial and living neural networks, will form a closed loop with a regulated feedback. The artificial neural network will implement conductance-based neuron and synapse models, controlled by plasticity rules such as STDP (spike-timing dependent plasticity). Dedicated integrated electronics will be designed to implement the communication channels between the living and artificial networks: signal conditioning for the biological signals (from living to artificial) and adapted coding of the artificial neuron events (from artificial to living).

This research topic is expected to generate scientific advances in the field of neuromorphic engineering by the design of embedded self-organized artificial neural networks that are able to communicate in real time with entire biological networks. The current challenge is to achieve bidirectional neuro-electronic interfaces, establishing a true dynamic communication between live neural networks and electronic systems.

Our group is made up of three teams:

- AS2N (Architecture of Silicon Neural Network);
- ELIBIO (ELectronique en Interaction avec la BIOlogie);
- BIO-EM (BIO-ElectroMagnetisme).

The goal of the AS2N group is to design and use integrated circuits (IC) that contain neuromimetic (i.e. they mimic biological neural systems) components and architecture. The AS2N team has designed several specific circuits which model the behavior of neurons, in collaboration with UNIC (Unité de Neuroscience, Information et Complexité, CNRS, Gif-sur-Yvette, France). This team is currently or was recently involved in European and national projects: FACETS (FP6-2004-IST-FETPI 15879 (2005-2009)), FACETS-ITN (FP7-PEOPLE-ITN-2008-237955 (2009-2014)), PIR NeuroInf ECRéN (2009-2011) project and MHANN (ANR P2N 2011-2015).

The Elibio team works on the development of hardware-software tools for computational neurosciences and neurophysiology, including hybrid living-artificial systems, analog ASICs for biological signal conditioning and events detection, active VLSI implants for neurodegenerative diseases, and closed-loop living-artificial systems.

The main research topic of the BIO-EM team is the study of the biological effects of electromagnetic fields in a wide frequency band. The current main theme is the assessment of possible health effects related to wireless communications such as mobile telephones. In addition, the team examines some fundamental aspects related to the biological effects of electromagnetic fields.

Our team (AS2N) focuses on three main research areas:

1. The design of mixed hardware-software platforms that are specifically designed to simulate spiking neural networks [Belhadj, 2010; Daouzli, 2009];
2. Parameter estimation of neuron models [Buhry, 2010; Daouzli, 2009];
3. The design of bio-inspired hardware platforms based on new technology such as memresistors (thesis in progress).

The research subject presented in this thesis (Marie-Curie Ph.D. position funded in the European FACETS-ITN project) is entitled: "Silicon neural networks: implementation of cortical cells to improve the artificial-biological hybrid technique". This work is part of the research areas 1 and 2, in continuity with the work carried out in several theses [Belhadj, 2010; Bornat, 2006; Buhry, 2010]. This work addresses two of the fields of research of our BioElectronics group: (i) to build a hardware simulation system for computational neuroscience in order to investigate plasticity and learning phenomena in spiking neural networks; and (ii) to improve the hybrid technique, which connects silicon and biological neurons in real time to study the function of neuronal circuits.

In a previous work carried out to implement neuron models in analog circuits, our group designed several neuromimetic chips (ASICs), including the Galway chip [Bornat, 2006] that we will use in this thesis to simulate the membrane equation of neurons. These silicon neurons are based on the Hodgkin-Huxley formalism and they are optimized to reproduce a large variety of neuronal behaviors using tunable parameters. However, due to the fabrication process, there are significant mismatches on the chips. It is also necessary to adjust their parameters in order to simulate a specific neuronal activity with the neuromimetic circuits. The automated adjustment of neuron models is a key issue in computational neuroscience. As hand-tuning is very time-consuming, automated tuning of the parameters is indispensable. Hence, we propose a fully customizable fitting method, in voltage-clamp mode, to tune our neuromimetic integrated circuits [Buhry et al., 2011]; this research is part of our thesis results.

The main objectives of this research subject are:

- the simulation of the four most prominent biological cells present in the neocortex (Fast-Spiking, Regular-Spiking, Intrinsically Bursting and Low-Threshold Spiking neurons) in conductance-based analog neuromimetic integrated circuits, using electrophysiological recordings as a reference;
- to contribute to the development of a mixed hardware-software platform to simulate spiking neural networks (SNN) in real time.

To address the first objective of this thesis, we can observe through the bifurcation analysis [Grassia et al., 2012] that the simplified HH model (implemented in the Galway chip) shares the dynamics of the HH model. Therefore, the Galway chip can be used to simulate the intrinsic electrophysiological properties of four cortical neurons: "fast-spiking", "regular-spiking", "intrinsically bursting" and "low-threshold spiking" cells. For each neuron class, we propose our simplified models taking all of the complete HH type models proposed by Pospischil et al. [2008] into account. By comparing them with experimental electrophysiological data of these cells, we show that the circuits can reproduce the main firing features of cortical cell types [Grassia et al., 2011].

Regarding the second objective, in recent years, few hardware-based SNN systems have been developed (we review some of them in the first chapter). These systems are generally application-dedicated and are based on a fully digital or mixed analog/digital architecture. Applications can range from purely artificial experiments, in particular the investigation of adaptation and plasticity phenomena in networks, to experiments on hybrid biological/artificial networks.

Our SNN is based on a mixed analog/digital architecture designed to have the capacity to address biological diversity in terms of neuron types as well as plasticity rules. Our platform makes use of point-neuron conductance-based models and is controlled by spike-timing dependent plasticity [Belhadj, 2010]. In particular, our contributions to the development of this mixed hardware-software platform for the simulation of a spiking neural network in real time, include:

- the development of software tools to communicate between the host PC and the SNN;
- the development of a VHDL driver to configure the neuron in the platform.

To address these contributions, this thesis will be presented in four chapters. The first chapter will recall the basic principles of neuroscience and neuron modeling useful to understand the context of this work. Particular attention is given to the neuron models that are generally used in the neuromorphic engineering, focusing on the choices made by our team to use the Hodgkin-Huxley formalism to model the electrical activity of neurons, and then the model used at the network level. The chapter ends with a review of the hardware simulation platforms for spiking neural networks and research projects related to this work. The second chapter will address the methods used to obtain the appropriate parameter sets of the cortical neurons models that can be implemented in our analog neuromimetic chip. The validation of the parameter extraction steps is done using a bifurcation analysis [Grassia et al., 2012] that shows that the simplified HH model implemented in our silicon neuron shares the dynamics of the HH model. Finally, for the simulation of the cortical neurons models, we will propose a parameter set for the simplified models that takes the complete HH type models proposed by Pospischil et al. [2008] into account, which also takes experimental electrophysiological data into account. The third chapter will deal with the hardware simulation of the cortical neurons. Therefore, this chapter will present a description of the hardware platform and the proposed fully customizable fitting method, in voltage-clamp mode, to tune our neuromimetic integrated circuits [Buhry et al., 2011]. Then, this chapter will show that the circuits can reproduce the main firing features of the cortical cells types by comparing the hardware simulation with experimental electrophysiological data [Grassia et al., 2011]. The last chapter will describe the work done for the silicon neuron integration in the PAX system (Plasticity Algorithm for Computing System: the spiking neural network (SNN) simulator developed by our team [Belhadj, 2010]), using the cortical neuron parameter sets proposed for the analog neuromimetic chip. After a description of the features and basic functions of the PAX simulator, this chapter will show how this work contributes to the development of the PAX systems in terms of software tools and a VHDL driver interface for neuron configuration in the platform. Finally, it also addresses the issue of synaptic tuning for future SNN simulation.

The conclusion of this research subject includes a description of the results and tasks performed, and discusses the implications of this work.

# Chapter 1

# From Biology to Neuromorphic Systems

Nowadays, many software solutions are currently available for simulating neuron models. Less conventional than software-based systems, hardware-based solutions generally combine digital and analog forms of computation. As mentioned in the introduction, the main objectives of this research are: the emulation of the four most prominent biological cells (fast spiking, regular spiking, intrinsically bursting, and low-threshold spiking neurons) into an analog neuromimetic integrated circuit and the development of a mixed hardware-software platform in order to simulate real-time spiking neural networks. Therefore, the aim of this chapter is to introduce several elementary notions of neuroscience, neuron modeling and neuromorphic engineering useful to understand the context of this work. Firstly, we recall the basic principles of neuron anatomy and physiology. Secondly, we describe the most successful and widely-used models of neurons classically implemented in hardware simulators and we explain the choices made by our team to use the Hodgkin-Huxley formalism to model the electrical activity of neurons. Finally, we describe the model used at the network level, we review the hardware simulation platforms for spiking neural networks and then we present the AS2N team's projects and, in particular, the European FACETS-ITN project that has supported this research.

## 1.1 Neuroscience Principles

In biology, there are radically different types of neurons in the human brain. In brain theory, the complexities of real neurons are abstracted in many ways to help understand the different aspects of neural network development, function or learning. In neural computation (technology based on networks of "neuron-like" units), the artificial neurons are designed as variations on the abstractions of brain theory and are implemented in the software or hardware [Arbib, 1995]. Individual nerve cells, the basic units of the brain, can be considered as simple cells. The complexity of human behavior depends less on the specialization of individual nerve cells and more on the fact that a great many of these cells form precise anatomical circuits [Kandel et al., 2000]. This section[1] describes neuron anatomy and physiology and focuses on the mechanisms of action potential generation and propagation.

### 1.1.1 Neuron anatomy

A typical neuron has four morphologically defined regions: the cell body, the dendrites, the axon, and the presynaptic terminals (Figure 1.1). Each of these regions has a distinct role in the generation of signals and the communication of signals between nerve cells. Neurons are highly specialized for the processing and transmission of cellular signals. Given the diversity of functions performed by neurons in different parts of the nervous system, there is, as expected, a wide variety in the shape, size, and electrochemical properties of neurons [Cooper, 2011].

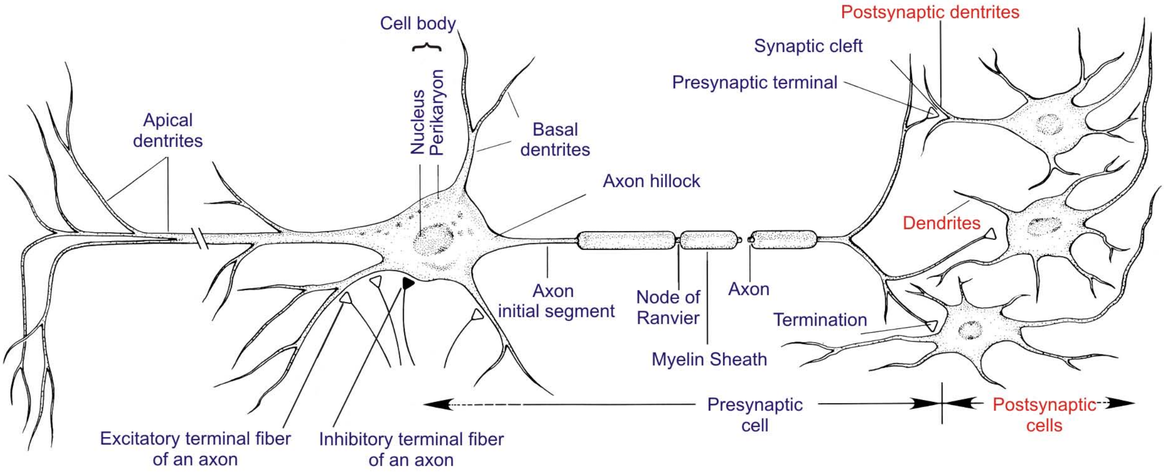


**Figure 1.1** – Structure of a neuron [Kandel et al., 2000].

#### Soma

The cell body, or soma, is the central part of the neuron. There are many different specialized types of neurons, and their sizes vary in diameter. It contains the nucleus of the cell (the storehouse of genetic information), and therefore is where most protein synthesis occurs. Roughly speaking, the soma is the "central processing unit" that performs

[1] All of the contents for the section "Neuroscience Principles" are referenced from Kandel et al. [2000], if not specially mentioned.

an important nonlinear processing step: if the total input exceeds a certain threshold, then an output signal is generated.

### Dentrites

Dendrites are the branched projections of a neuron that act to conduct the electrochemical stimulation received from other neural cells to the cell body, or soma, of the neuron from which the dendrites project, play the role of the "input device". Metaphorically, this overall shape and structure is referred to as a dendritic tree. This is where the majority of input to the neuron occurs. Electrical stimulation is transmitted onto dendrites by upstream neurons via synapses which are located at various points throughout the dendritic arbor. Dentrites and soma constitute the major part of the input surface of the neuron.

### Axon

The axon, which plays the role of the "output device", is an elongated fiber that extends from the cell body to the terminal endings and transmits the neural signal. The transmitting element of neurons can vary greatly in length; most axons in the central nervous system are very thin (between 0.2 and 20 micrometers in diameter) compared with the diameter of the cell body (50 micrometers or more). Many axons are insulated by a fatty sheath of myelin that is interrupted at regular intervals by the nodes of Ranvier. Nodes of Ranvier are constrictions in the myelin sheath that surround the axons of the nerve cells, or neurons. They occur at approximately one millimeter intervals along the length of the axon. The action potential, the cell's conducting signal, is initiated either at the axon hillock, the initial segment of the axon, or in some cases slightly farther down the axon at the first node of Ranvier [Kandel et al., 2000].

### Synapses

The junction between two neurons is called a synapse. Branches of the axon of one neuron (the presynaptic neuron) transmit signals to another neuron (the postsynaptic neuron) at a site called the synapse. The branches of a single axon may form synapses with as many as 10.000 other neurons. Whereas the axon is the output element of the neuron, the dendrites are the input elements of the neuron. Together with the cell body, they receive synaptic contacts from other neurons. Neurons can be coupled by chemical and electrical synapses. Neurons communicate by chemical and electrical synapses in a process known as synaptic transmission.

### Plasma membrane

Like all animal cells, every neuron is surrounded by a plasma membrane, a bilayer of lipid molecules with many types of protein structures embedded in it. A lipid bilayer is a powerful electrical insulator, but in neurons, many of the protein structures embedded in the membrane are electrically active. These include ion channels that permit electrically charged ions to flow across the membrane, and ion pumps that actively transport ions from one side of the membrane to the other (Figure 1.2). Inside the cell, the concentration of ions is different from that in the surrounding liquid.

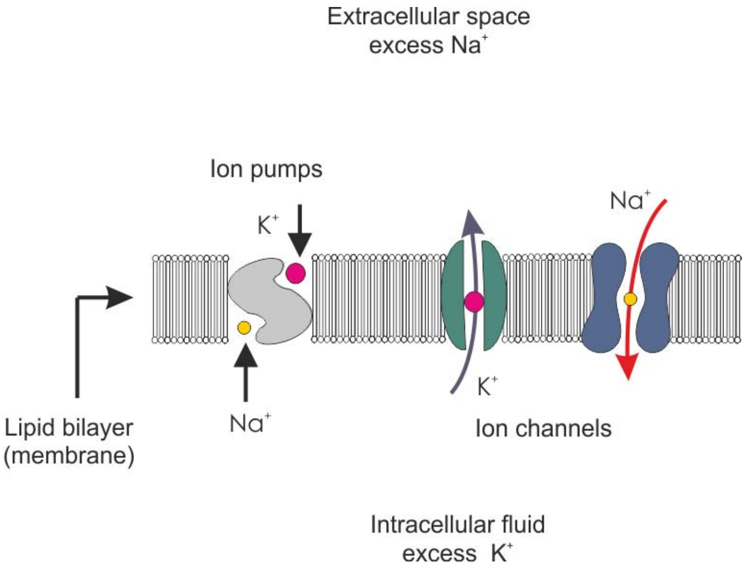


**Figure 1.2** – The neuron's membrane forms a separation between the extracellular space around the neuron and its intracellular fluid. The membrane is pierced with proteins that serve as channels for ions to "flow" though.

The charge separation gives rise to a difference of electrical potential across the membrane called the membrane potential. In a biological membrane, the reversal potential of an ion is the membrane potential at which there is no net flow of that particular ion from one side of the membrane to the other. The reversal potential is often called the "Nernst potential" as it can be calculated from the Nernst equation [Feiner and McEvoy, 1994]. In a single-ion system, the reversal potential is synonymous with equilibrium potential. Equilibrium refers to the fact that the net ion flux at a particular voltage is zero. In other words, the outward and inward rates of ion movement are the same; the ion flux is in equilibrium. Reversal refers to the fact that a change of membrane potential on either side of the equilibrium potential reverses the overall direction of ion flux [Purves et al., 2008]. The relationship between the terms "reversal potential" and "equilibrium potential" only holds true for single-ion systems. In multi-ion systems, there are areas of the cell membrane where the summed currents of the multiple ions will equal zero. While this is a reversal potential in the sense that the membrane current reverses direction, it is not an equilibrium potential because not all of the ions are in equilibrium and thus have net fluxes across the membrane.

### 1.1.2 Neuron physiology

The neuronal signal depends on rapid changes in the electrical potential difference across the nerve cell membranes. Individual sensory cells can generate changes in membrane potential in response to very small stimuli. These rapid changes in membrane potential are mediated by ion channels, a class of integral membrane proteins found in all cells of the body. Ion channels have three important properties: (1) they conduct ions, (2) they recognize and select specific ions, and (3) they open and close in response to specific electrical, mechanical, or chemical signals. Voltage-gated channels are regulated by changes in voltage, ligand-gated channels are regulated by chemical transmitters, and mechanically gated channels are regulated by pressure or stretch. An individual channel is usually most sensitive to one type of signal. In addition to the gated channels, there are non-gated channels that are normally open in the cell at resting state when nothing perturbs the cell.

#### The resisting membrane potential

Every neuron has a separation of charges across its cell membrane consisting of positive and negative ions spread over the inner and outer surfaces of the cell membrane. At resting state, a nerve cell has an excess of positive charges on the outside of the membrane and an excess of negative charges on the inside. No single ion species is distributed equally on the two sides of a nerve cell membrane. Ions are subject to two forces driving them across the membrane: (1) a chemical driving force that depends on the concentration gradient across the membrane, and (2) an electrical driving force that depends on the electrical potential difference across the membrane. Of the most abundant ions found on either side of the cell membrane, $Na^+$ is more concentrated outside the cell and $K^+$ is more concentrated inside. This separation of charge is maintained because the lipid bilayer of the membrane blocks the diffusion of ions. The membrane potential ($V_m$) is defined by

**equation** 1.1:

$$V_m = V_{in} - V_{out} \; , \qquad \textbf{(equation 1.1)}$$

where $V_{in}$ is the potential on the inside of the cell and $V_{out}$ the potential on the outside. There are two important levels of membrane potential: the resting potential, which is the value that the membrane potential maintains as long as nothing perturbs the cell, and a higher value called the threshold potential (around -55 mV).

In the nerve cell, at resting-state, the steady $Na^+$ influx is balanced by a steady $K^+$ efflux, so that the membrane potential is constant. Since, by convention, the potential outside the cell is defined as zero, the resting potential is equal to $V_{in}$. Its usual range in neurons is -60 mV to -70 mV. This balance point (usually -65 mV) is far from the $Na^+$ equilibrium potential ($E_{Na}$ = +55 mV) and is only slightly more positive than the equilibrium potential for $K^+$ ($E_K$ = -75 mV).

### The action potential

All electrical signaling involves brief changes from the resting membrane potential due to alterations in the flow of electrical current across the cell membrane resulting from the opening and closing of ion channels. A reduction of charge separation, leading to a less negative membrane potential, is called depolarization. An increase in charge separation, leading to a more negative membrane potential, is called hyperpolarization. Synaptic inputs to a neuron cause the membrane to depolarize or hyperpolarize [Kandel et al., 2000]. In physiology, an action potential is a short-lasting event (about 1 ms) in which the electrical membrane potential of a cell rapidly rises and falls, following a consistent trajectory. The absolute refractory period coincides with nearly the entire duration of the action potential. Action potentials are triggered when enough depolarization accumulates to bring the membrane potential up to the threshold. Action potentials in neurons are also known as "spikes", and the temporal sequence of action potentials generated by a neuron is called its "spike train". In neuronal systems, information is coded in the frequency or timing of action potentials. Figure 1.3 shows the genesis of an action potential. In the nerve cell, at resting potential, the membrane potential is constant. However, action potentials are triggered when enough depolarization accumulates to bring the membrane potential up to the threshold. Once the membrane potential reaches this threshold, the voltage-gated $Na^+$ channels open rapidly. The resultant increases in membrane permeability to $Na^+$ causes the $Na^+$ influx to exceed the $K^+$ efflux, creating a net influx of positive charge that causes further depolarization.

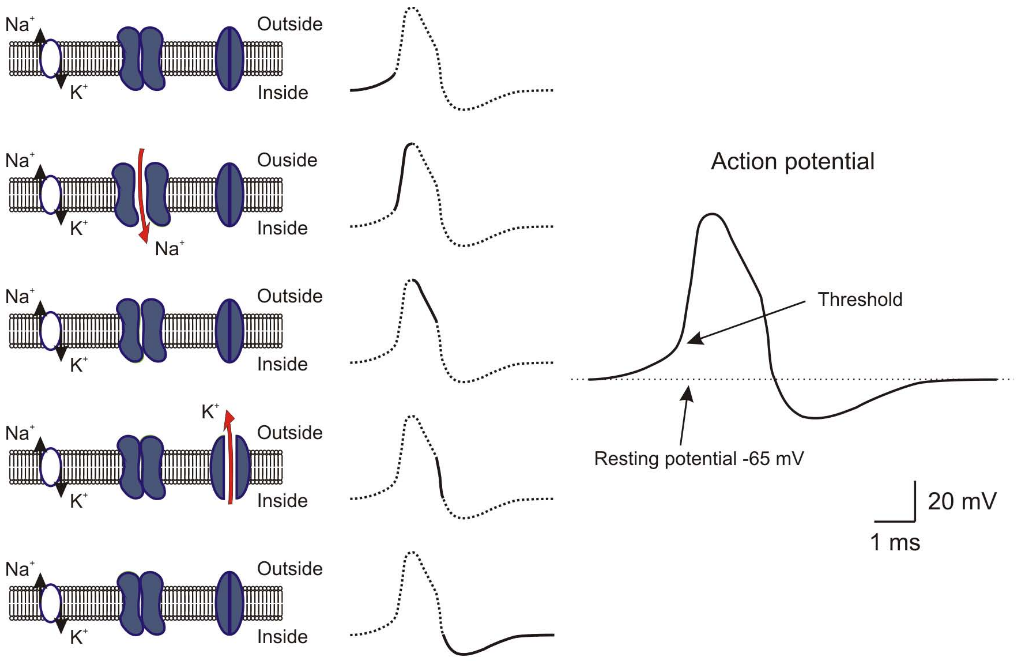


**Figure 1.3** – Ionic mechanisms of action potentials.

The increase in depolarization causes even more voltage-gated Na$^+$ channels to open, resulting in a greater influx of Na$^+$, which accelerates the depolarization even further. This regenerative, positive feedback cycle develops explosively, driving the membrane potential toward the Na$^+$ equilibrium potential of +55 mV. However, the membrane potential never quite reaches that point because K$^+$ efflux continues throughout the depolarization. Nevertheless, so many voltage-gated Na$^+$ channels open during the rising phase of the action potential that the cell's permeability to Na$^+$ is much greater than to K$^+$. Thus at the peak of the action potential, the membrane potential approaches the Na$^+$ equilibrium potential, just as at rest (when permeability to K$^+$ is predominant), the membrane potential tends to approach the K$^+$ equilibrium potential. The membrane potential would remain at this large positive value near the Na$^+$ equilibrium potential indefinitely but for two processes that repolarize the membrane, thus terminating the action potential. First, as the depolarization continues, the population of voltage-gated Na$^+$ channels gradually closes by the process of inactivation. Second, opening of the voltage-gated K$^+$ channels causes the K$^+$ efflux to gradually increase. The increase in K$^+$ permeability is slower than the increase in Na$^+$ permeability because the voltage-gated K$^+$ channels open at a slower rate. The delayed increase in K$^+$ efflux combines with a decrease in Na$^+$ influx to produce a net efflux of positive charge from the cell, which continues until the cell has repolarized to its resting membrane potential.

The action potential is also followed by a brief period of diminished excitability, or refractoriness, which can be divided into two phases. The absolute refractory period comes immediately after the action potential; during this period it is impossible to excite the cell no matter how great a stimulating current is applied. This phase is directly followed by the relative refractory period, during which it is possible to trigger an action potential but only by applying stimuli that are stronger than those normally required to reach the threshold. These periods of refractoriness are caused by changes in the state of sodium and potassium channel molecules. When closing after an action potential, sodium channels enter an "inactivated" state, in which they cannot be made to open regardless of the membrane potential; this gives rise to the absolute refractory period. Even after a sufficient number of sodium channels have transitioned back to their resting state, it frequently happens that a fraction of potassium channels remains open, making it difficult for the membrane potential to depolarize, and thereby giving rise to the relative refractory period.

**Propagation of the action potential**

The action potential generated at the axon hillock propagates as a wave along the axon. Once the membrane at any point along an axon has been depolarized beyond a threshold, an action potential is generated in that region in response to the opening of voltage-gated $Na^+$ channels. This local depolarization spreads down the axon, causing the adjacent region of the membrane to reach the threshold for generating an action potential. If sufficiently strong, this depolarization provokes a similar action potential at the neighboring membrane patches. When an action potential has occurred at a patch of membrane, the membrane patch needs time to recover before it can fire again. This absolute refractory period corresponds to the time required for the voltage-activated sodium channels to recover from inactivation. Although it limits the frequency of firing, the absolute refractory period ensures that the action potential moves in only one direction along an axon. However, only the unfired part of the axon can respond with an action potential; the part that has just fired is unresponsive until the action potential is safely out of range and cannot re-stimulate that part. The speed of action potential propagation is usually directly related to the size of the axon.

In order to enable the fast and efficient transduction of electrical signals in the nervous system, certain neuronal axons are covered with myelin sheaths. Myelin is a multilamellar membrane that enwraps the axon in segments separated by intervals known as nodes of Ranvier. It is at these uninsulated spots on the axon that the action potential becomes regenerated. Although the area of the membrane at each node is quite small, the nodal membrane is rich in voltage-gated $Na^+$ channels and thus can generate an intense depolarizing inward $Na^+$ current in response to the passive spread of depolarization down the axon. The myelin sheath reduces membrane capacitance and increases membrane resistance in the inter-node intervals, thus allowing a fast, saltatory movement of action potentials from node to node. These regularly distributed nodes thus boost the amplitude of the action potential periodically, preventing it from dying out.

**Termination**

The axon divides (near its end) into fine branches that form communication sites with other neurons. The point at which two neurons communicate is known as a synapse. The nerve cell transmitting a signal is called the presynaptic cell. The cell receiving the signal is the postsynaptic cell. The presynaptic cell transmits signals from the swollen ends of the axonal branches, called presynaptic terminals. Most presynaptic terminals end on the postsynaptic neuron's dendrites, but the terminals may also end on the cell body or, less often, at the beginning or end of the axon of the receiving cell. Synapses are either electrical or chemical and the current flows differently:

- at an electrical synapse, some of the current injected into a presynaptic cell escapes through resting ion channels in the cell membrane. However, some current also flows into the postsynaptic cell through specialized ion channels, called gap-junction channels, that connect the cytoplasm of the pre and postsynaptic cells (see Figure 1.4 A )

- at chemical synapses, the resulting depolarization of the presynaptic cell activates the release of neurotransmitter molecules packaged in synaptic vesicles into the synaptic cleft (the intracellular space between the two neurons), which then bind to receptors on the postsynaptic cell (see Figure 1.4 B ). This binding opens ion channels, thus initiating a change in membrane potential in the postsynaptic cell. The binding of the transmitter to the receptors causes the postsynaptic cell to generate a synaptic potential. Whether the synaptic potential has an excitatory or inhibitory effect will depend on the type of receptors in the postsynaptic cell, not on the particular neurotransmitter. The same transmitter can have different effects on different types of receptors. The major neurotransmitters in the brain are glutamate and $\gamma$-aminobutyric acid (GABA). The former has a excitatory effect, i.e. it induces a depolarization of the postsynaptic membrane; the latter is an inhibitory transmitter which causes an hyperpolarization of the postsynaptic neuron. GABA receptors (GABA$_A$ and GABA$_B$) are associated with a reversal potential around -75 mV. For glutamate, there are the so-called AMPA and NMDA receptors, both with a reversal potential around 0 mV. The relevant difference between them is the timescale on which they react: 5 ms and 50 ms for GABA$_A$ and GABA$_A$ receptors; 0.1 ms and 10-100 ms for AMPA and NMDA receptors.

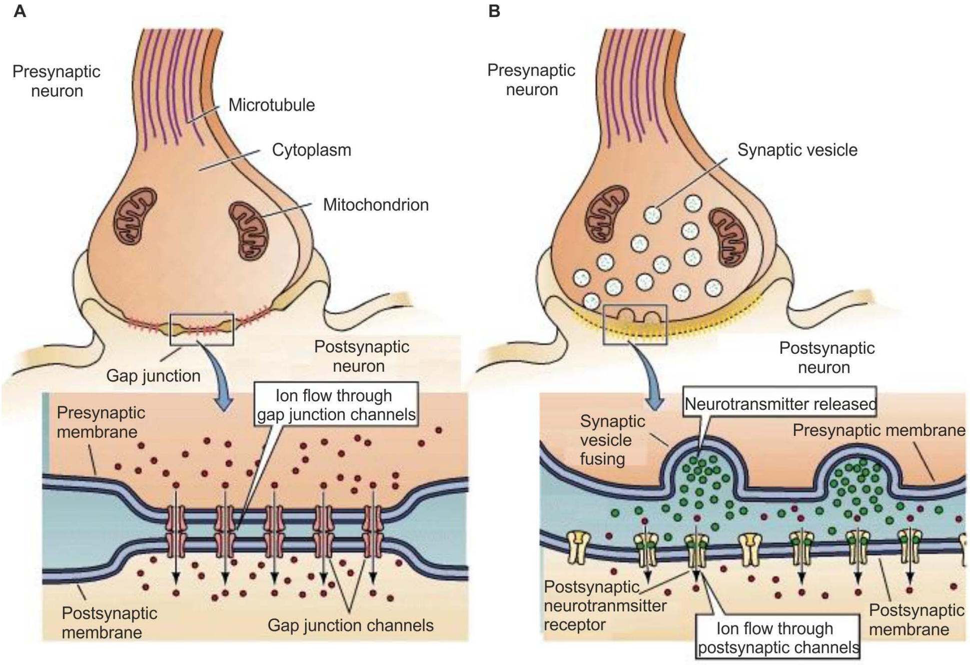


**Figure 1.4** – (A) Electrical and (B) chemical synapses [KIN450, 2009].

Electrical synapses are used primarily to send simple depolarizing signals; they do not lend themselves to producing inhibitory actions. In contrast, chemical synapses are capable of more variable signaling and thus can produce more complex behaviors. They can mediate either excitatory (with the receptor AMPA and NMDA ) or inhibitory (with the receptor GABA$_A$ and GABA$_B$) actions in postsynaptic cells and produce electrical changes in the postsynaptic cell that last from milliseconds to many minutes. Chemical synapses also serve to amplify neuronal signals, so that even a small presynaptic nerve terminal can alter the response of a large postsynaptic cell. Moreover, the strength of both forms of synaptic transmission can be enhanced or diminished by cellular activity.

In neuroscience, synaptic plasticity is the ability of the connection, or synapse, between two neurons to change in strength or efficacy in response to either use or disuse of transmission at preexisting synapses. Since memories are postulated to be represented by vastly interconnected networks of synapses in the brain, synaptic plasticity is one of the important neurochemical foundations of learning and memory.

## 1.2 Principles of Neural Modeling

A biological neuron model (also known as a spiking neuron model) is a mathematical description of the properties of nerve cells, or neurons, that is designed to accurately describe and predict biological processes. Models that describe the membrane potential of a neuron by a single variable and ignore its spatial variation are called single-compartment models. In this sub-class of models, the rich and complex dynamics of real neurons can be reproduced quite accurately by models that include aspects of ionic conductances, known as conductance-based models. To study the effects of dendritic or axonal morphologies on neuronal function, models based on the linear cable theory and multi-compartmental models have to be considered instead. In this section, we describe the most successful and widely-used single-compartment neuron models classically implemented in hardware simulators and we explain the choices made by our team to use the Hodgkin-Huxley formalism to model the electrical activity of silicon neurons. We start with a summary of the neuro-computational properties of biological spiking neurons reviewed by Izhikevich [2004]. This summary includes the kinds of behaviors for fast spiking, regular spiking, intrinsically bursting, and low-threshold spiking neurons that we want to emulate into an analog neuromimetic integrated circuit.

### 1.2.1 Neuro-computational properties of biological spiking neurons

Neurons can respond with rather different spike train patterns to identical step currents. Figure 1.5 shows the important neurocomputational features of real neurons reviewed by Izhikevich [2004]. To observe these different spike train patterns, neurophysiologists inject pulses of DC current via an electrode attached to the neuron and record its membrane potential. The input current and the neuronal response are plotted one beneath the other, as in Figure 1.5. The kind of behavior, called tonic spiking (Figure 1.5), can be observed in the three types of cortical neurons: regular spiking (RS) excitatory neurons, low-threshold spiking (LTS) and fast spiking (FS) inhibitory neurons [Connors and Gutnick, 1990; Gibson et al., 1999].

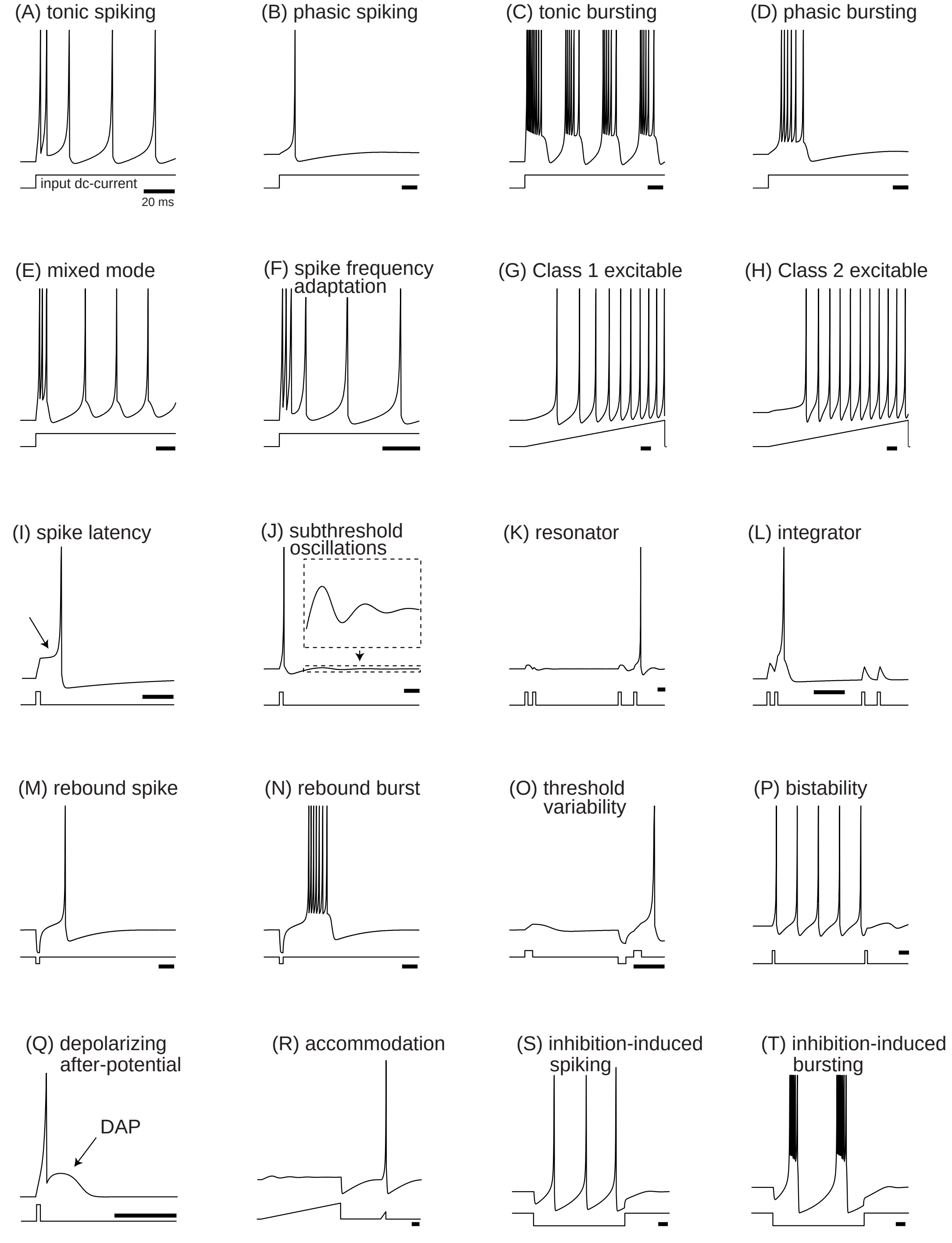


**Figure 1.5** – Summary of the neuro-computational properties of biological spiking neurons (electronic version of the figure and reproduction permissions are freely available at www.izhikevich.com).

The continuous firing of these neurons indicates that there is a persistent input. Intrinsically bursting (IB) excitatory neurons in the mammalian neocortex [Connors and Gutnick, 1990] can exhibit a mixed mode (Figure 1.5) of spiking activity: they fire a phasic burst (such neurons report the beginning of the stimulation by transmitting a burst[2] and then switch to the tonic spiking mode. The complete types of behaviors (Figure 1.5) are reviewed by Izhikevich [2004].

Which model to use for cortical spiking neurons? All models have advantages and drawbacks [Izhikevich, 2004]. In our case, we wish to reproduce the behavior of biological neurons in order to extend the hybrid technique, also called a "dynamic-clamp" [Le Masson et al., 1995], to Micro-Electrode Arrays [Bontorin et al., 2007] by making use of reconfigurable integrated circuits [Saïghi et al., 2011]. For that, we chose the most biologically plausible model: the Hodgkin-Huxley-type model. In the next sections (1.2.2 - 1.2.3 - 1.2.4), starting from the Hodgkin-Huxley formalism, we review the most used single-compartment neuron models classically implemented in hardware simulators. Afterwards, we show a comparison of most of them in terms of biological plausibility [Izhikevich, 2004].

### 1.2.2 Hodgkin-Huxley model

The first biologically relevant mathematical neuron model was proposed by Hodgkin and Huxley [1952][3]. Hodgkin and Huxley performed experiments on the giant axon of the squid and found three different types of current: sodium, potassium and leak current. It was demonstrated that the ionic permeability of the membrane can be highly dependent on the membrane potential. Hodgkin and Huxley characterized these properties of voltage-dependence and provided a mathematical model which proved that these properties were sufficient to account for the genesis of action potentials. They used the voltage-clamp technique[4] to record the ionic currents generated at different voltages. All channels may be characterized by their resistance or, equivalently, by their conductance. The schematic diagram of the Hodgkin-Huxley model is shown in Figure 1.6 where $V_M$ is the membrane

[2] Bursting is a dynamic state where a neuron repeatedly fires discrete groups of spikes. Each such burst is followed by a period of quiescence before the next burst occurs. A burst of two spikes is called a doublet, of three spikes is called a triplet, four - quadruplet, etc. [Izhikevich, 2000].

[3] The Nobel Prize in Physiology or Medicine 1963 was awarded jointly to Sir John Carew Eccles, Alan Lloyd Hodgkin and Andrew Fielding Huxley "for their discoveries concerning the ionic mechanisms involved in excitation and inhibition in the peripheral and central portions of the nerve cell membrane"

[4] The voltage clamp is used by electrophysiologists to measure the ion currents across the membrane of excitable cells, such as neurons, while holding the membrane voltage at a set level.

potential, $C_M$ is the membrane capacitance, the leakage channel is described by a voltage-independent conductance $g_L$=1/R[5] and the conductance of the other ion channels ($g_{Na}$ and $g_K$) is voltage and time dependent. $E_L$, $E_{Na}$ and $E_K$ are the reversal potentials, which are given by the Nernst equation[6].

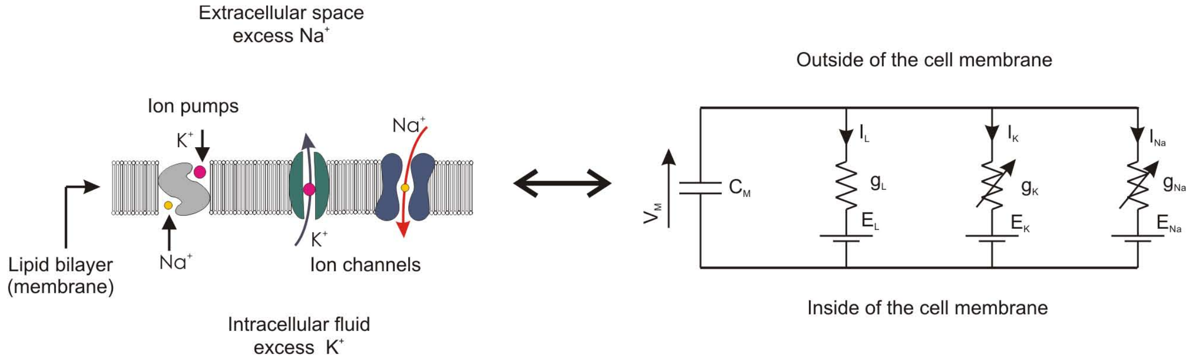


**Figure 1.6** – The schematic diagram of the Hodgkin-Huxley model: analogy between biology and electrical circuit.

Hence, from the schematic in Figure 1.6 the current flowing across the membrane is integrated on the membrane capacitance, according to expression:

$$C_{\mathrm{M}}\frac{dV_{\mathrm{M}}}{dt} = -I_{Na} - I_K - I_{\mathrm{L}} \ , \qquad \textbf{(equation } 1.2)$$

where $I_{Na}$, $I_K$ and $I_L$ are sodium, potassium and leakage currents. We will see more details of the HH model in Chapter 2.

The Hodgkin-Huxley equations are the starting point for detailed neuron models which account for numerous ion channels, different types of synapses, and the specific spatial geometry of an individual cell. Thus, other simplified models derived from the HH model have been developed since then.

[5] The passive electrical properties of the cell membrane are described by a capacity $C_M$ and a resistor R.

[6] For example, for $K^+$ ions:

$$E_K = \frac{R_g \cdot T}{Z \cdot F}\, ln\frac{[K]_o}{[K]_i} \ ,$$

where $R_g$ is the gas constant, T is the absolute temperature in degrees Kelvin, Z is the valence of the ion (Z = 1 for $K^+$ ions, Z = -1 for $Cl^-$ ions, etc), F is the Faraday constant, $[K]_o$ and $[K]_i$ are the concentration of $K^+$ ions outside and inside of the membrane, respectively.

### 1.2.3 Simplified models derived from the HH model

The behavior of high-dimensional nonlinear differential equations is difficult to visualize and even more difficult to analyze. Two-dimensional differential equations, however, can be studied by means of a phase plane analysis. A reduction of the four-dimensional HH model to a two variable neuron model is thus highly desirable. Sweeping simplifications to the Hodgkin-Huxley model were introduced by FitzHugh [1955, 1961], Nagumo et al. [1962] and Morris and Lecar [1981].

**FitzHugh-Nagumo model**

In order to reduce the Hodgkin-Huxley model to a two variable model, the general observation of FitzHugh was that the gating variables n and h have slow kinetics relative to m. Moreover, for the parameter values specified by Hodgkin and Huxley, n plus h is approximately 0.8. The FitzHugh-Nagumo model is able to reproduce many qualitative characteristics of electrical impulses along nerve and cardiac fibers, such as the existence of an excitation threshold, relative and absolute refractory periods, and the generation of pulse trains under the action of external currents. The model is described by the following equations:

$$\begin{cases} \dfrac{dV_{\mathrm{M}}}{dt} = V_{\mathrm{M}} - V_{\mathrm{M}}^3 - w + I_{\mathrm{ext}} \\ \tau\dfrac{dw}{dt} = V_{\mathrm{M}} - a - b \cdot w \, , \end{cases} \qquad \textbf{(equation } 1.3)$$

where we again have a membrane-like voltage $V_M$, input current $I_{ext}$, a slower general gate voltage w, and experimentally determined parameters a = -0.7, b = 0.8 and $\tau$ = 1/0.08.

**Morris-Lecar model**

Morris and Lecar [1981] combined Hodgkin-Huxley's and FitzHugh-Nagumo's model into a voltage-gated calcium channel model with a delayed-rectifier potassium channel. Qualitatively, the Morris-Lecar model describes the complex relationship between membrane potential and the activation of ion channels within the membrane: the potential depends on the activity of the ion channels, and the activity of the ion channels depends on the voltage. The assumptions were: 1) the equations apply to an iso-potential patch of membrane, 2) $Ca^{2+}$ carries the depolarizing current, 3) $K^+$ carries the hyperpolarizing current, 4) the activating conductance quickly relaxes to its steady state value independent of voltage, and 5) the recovery dynamics of the neuron are modeled as only a first-order differential equation [Lecar, 2007]. The model is described by the following equations:

$$\begin{cases} C_{\text{M}} \dfrac{dV_{\text{M}}}{dt} = -\sum\limits_{\text{ion}} I_{\text{ion}}(V_{\text{M}}) + I_{ext} \\ \dfrac{dw}{dt} = \dfrac{w_{\infty}(V_{\text{M}}) - w}{\tau_w(V_{\text{M}})} , \end{cases} \qquad \textbf{(equation } 1.4\textbf{)}$$

where

$$\sum_{\text{ion}} I_{\text{ion}} = g_{\bar{Ca}} \cdot m_{\infty}(V_{\text{M}} - V_{Ca}) + g_{\bar{K}} \cdot w \cdot (V_{\text{M}} - E_K) + g_{\bar{\text{fuite}}} \cdot (V_{\text{M}} - E_{\text{fuite}}) . \qquad \textbf{(equation } 1.5\textbf{)}$$

The recovery variable w describes the probability that the K$^+$ channel is conducting.

### Hindmarsh-Rose model

Building upon the FitzHugh-Nagumo model, Hindmarsh and Rose [1984] proposed a model of neuronal activity described by three coupled first-order differential equations:

$$\begin{cases} \dfrac{dx}{dt} = y + a \cdot x^2 - x^3 - z + I \\ \dfrac{dy}{dt} = 1 - b \cdot x^2 - y \\ \dfrac{dz}{dt} = r \cdot [s \cdot (x - x_R) - z] . \end{cases} \qquad \textbf{(equation } 1.6\textbf{)}$$

The relevant variable is the membrane potential, x(t), which is written in dimensionless units. There are two more variables, y(t) and z(t), which take into account the transport of ions across the membrane through the ion channels. The transport of sodium and potassium ions is made through fast ion channels and its rate is measured by y(t), which is called the spiking variable. The transport of the other ions is made through slow channels, and is taken into account through z(t), which is called the bursting variable. The model has six parameters: a, b, r, s, $x_R$ and I. It is very common to fix some of them and let the other be control parameters. Usually parameter I, which means the current that enters the neuron, is taken as a control parameter. The aim of the Hindmarsh-Rose model of neuronal activity is to study the spiking-bursting behavior of the membrane potential observed in experiments made with a single neuron.

### 1.2.4 Other widely used models of spiking neurons

Unlike the models mentioned in the previous section, which are composed of distinct ion channels with different gating variables, one dimensional model uses a threshold and reset mechanism to produce action potentials. A common one-dimensional model is the integrate-and-fire model.

#### Integrate-and-Fire (IF) model

The simplest mechanism to model neuronal spiking is the integrate-and-fire (IF) model investigated in 1907 by Louis Lapicque [Abbott, 1999]. A neuron is represented in time by:

$$I(t) = C_{\mathrm{M}} \frac{dV_{\mathrm{M}}}{dt} \qquad \textbf{(equation } 1.7)$$

in which a spike occurs when the membrane potential $V_M$ reaches the threshold voltage; a spike is produced and then $V_M$ is reset back to the resting potential. The firing frequency of the model increases linearly without bound as the input current increases. The model can be made more accurate by introducing a refractory period that limits the firing frequency of a neuron by preventing it from firing during that period. A remaining shortcoming of this model is that it implements no time-dependent memory. If the neuron receives a below-threshold signal at some time, it will keep that voltage boost forever until it fires again. In the leaky integrate-and-fire model, the memory problem is solved by adding a leak term to the membrane potential.

#### Leaky Integrate-and-Fire (LIF) model

The basic circuit of an integrate-and-fire model consists of a capacitor $C_M$ in parallel with a resistor $R_M$ driven by a current I(t) [Gerstner and Kistler, 2002]. The driving current can be split into two components. The first component is the resistive current which passes through the linear resistor $R_M$ and, the second component is the current that charges the capacitor $C_M$. Thus, the dynamic of the circuit is given by

$$I(t) - \frac{V_{\mathrm{M}}(t)}{R_{\mathrm{M}}} = C_{\mathrm{M}} \frac{dV_{\mathrm{M}}}{dt} \qquad \textbf{(equation } 1.8)$$

where $C_M$ is the membrane capacitance, $R_M$ is the membrane resistance, which is not a perfect insulator as previously assumed. The resistor in this model represents the leak currents of the neuron, thus

$$C_{\mathrm{M}} \frac{dV_{\mathrm{M}}}{dt} = I(t) - g_{\mathrm{L}} \cdot (V_{\mathrm{M}}(t) - E_{\mathrm{L}}) \, , \qquad \text{(equation 1.9)}$$

where $g_L = 1/R_M$. This dynamic holds until the threshold is crossed, after which the membrane potential is reset. For a constant current I, the threshold current for spike generation is $I_{th} = V_{th}/R_M$, and the evolution of the membrane potential through time is given by

$$V_{\mathrm{M}}(t) = R_{\mathrm{M}} \cdot I \cdot \left[1 - exp\left(\frac{t - t_s}{R_{\mathrm{M}} \cdot C_{\mathrm{M}}}\right)\right] \, , \qquad \text{(equation 1.10)}$$

where $t_s$ is the last spiking time. This simple phenomenological spiking neuron model is highly popular for studies of large-scale network dynamics where only the overall activity of the system is important [Casti et al., 2002; Koene and Hasselmo, 2005; Rangan and Cai, 2007].

### Quadratic Integrate-and-Fire model

An alternative to the LIF neuron is the quadratic IF neuron. The quadratic model describes the behavior of a certain family of neuron models called type-I models. The name refers to the shape of the firing curve (the curve describing the relation between a constant input current and the spiking rate): zero below a certain threshold, monotonically increasing after the threshold is crossed, and non-continuous when the threshold is crossed [Gerstner and Kistler, 2002]. The evolution of the membrane potential through time is given by

$$\tau_m \frac{dV_{\mathrm{M}}}{dt} = a_0 \cdot (V_{\mathrm{M}}(t) - V_{rest}) \cdot (V_{\mathrm{M}}(t) - V_{th}) + R_{\mathrm{M}} \cdot I(t) \, , \qquad \text{(equation 1.11)}$$

where $\tau_m = R_M \cdot C_M$ , $a_0 > 0$ and $V_{th} > V_{rest}$ are parameters. Note that for I = 0, if $V_M < V_{th}$ the voltage decays to the resting potential ($V_{rest}$). If $V_M > V_{th}$, the voltage increases until spike generation. Therefore $V_{th}$ is interpreted as the threshold voltage for spike generation.

### Resonate-and-Fire model

The resonate-and-fire neuron is a two-dimensional analogue of the IF neuron and may be characterized as follows [Izhikevich, 2001]:

$$\begin{cases} \dfrac{dV_{\mathrm{M}}}{dt} = I(t) - g_{\mathrm{L}} \cdot (V_{\mathrm{M}}(t) - E_{\mathrm{L}}) - W(t) \\ \tau \dfrac{dW}{dt} = \dfrac{(V_{\mathrm{M}}(t) - V_0)}{k} - W(t)\ , \end{cases} \qquad \textbf{(equation } 1.12)$$

where W is a second variable that summarizes the effect of subthreshold membrane current. The notion that underlies the resonate-and-fire neuron is that biological neuron's internal activity level is determined by the interplay of fast and slow membrane currents.

### Izhikevich's model

All of the responses in Figure 1.5 were obtained using a simple model of spiking neurons proposed by [Izhikevich, 2003]:

$$\begin{cases} \dfrac{dV_{\mathrm{M}}}{dt} = 0.04 \cdot V_{\mathrm{M}}(t)^2 + 5 \cdot V_{\mathrm{M}}(t) + 140 - u(t) + I(t) \\ \dfrac{du}{dt} = a \cdot (b \cdot V_{\mathrm{M}}(t) - u(t)) \\ \text{if } v \geq +30mV \qquad \text{then}\, v \leftarrow c,\, u \leftarrow u + d \end{cases} \qquad \textbf{(equation } 1.13)$$

where u represents a membrane recovery variable, which accounts for the activation of $K^+$ ionic currents and the inactivation of $Na^+$ ionic currents. After the spike reaches its apex (+30 mV), the membrane voltage and the recovery variable are reset. The model can exhibit firing patterns of all known types of cortical neurons with the choice of parameters a, b, c, and d [Izhikevich, 2003].

### Which model to use for cortical spiking neurons emulations?

Many single-compartment models of spiking neurons have been proposed. Which one to choose? The answer depends on the type of the problem, and therefore a comparison of the spiking neuron models in terms of biological plausibility was done by Izhikevich (Figure 1.7). As shown in Figure 1.7, if the goal is to study how the neuronal behavior depends on the measurable physiological parameters, then the Hodgkin-Huxley-type model is the most biologically plausible, but is computationally prohibitive to simulate large networks in real time. In contrast, if the goal is to simulate thousands of spiking neurons in real time, then there are plenty of models to choose from. The most efficient is the integrate-and-fire model.

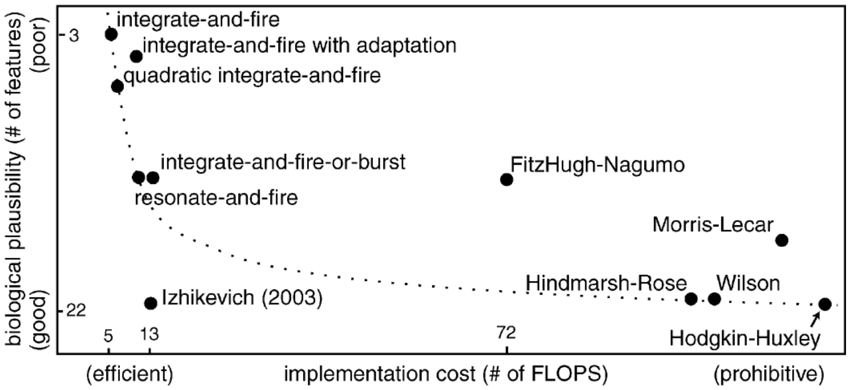


**Figure 1.7** – Biological plausibility of neuron models [Izhikevich, 2003].

In our case, in which it should be possible to use the hybrid technique to better understand the biological phenomena, the chosen model has to be the most biologically plausible. Within the family of biologically plausible point neuron models, there is a group of conductance-based models (implemented into the silicon neuron [Bornat, 2006]) in which ionic and synaptic currents charge and discharge a capacitor representing the neuron membrane. All of these models are based on the HH model, which will be described in Chapter 2.

We have no limitations with regards to flops (floating point operations per second) for our design [Bornat, 2006] thanks to the analog computation. However, due to the fabrication process, there are significant differences between the expected and actual outputs of the analog silicon neuron [Buhry et al., 2011]. It is therefore necessary to further adjust the parameters in order to reproduce a given neuronal signal. In the next section we briefly describe this key issue.

## 1.3 The automated adjustment of neuron models

The HH model is strongly dependent on nonlinear equations involving a large number of parameters: ranging from 15 parameters for the very simple monocompartmental model of a fast-spiking neuron to hundreds of parameters for neurons with more complex behaviors. It is the starting point for detailed neuron models which, in general, include more than the three types of currents considered by Hodgkin and Huxley [Pospischil et al., 2008]. The tuning of this model, in order to reproduce a given neuronal signal, is thus complex. The adjustment of neuron models is a key issue in computational neuroscience. This subject has already been investigated by several authors, considering different levels of model complexity [Geit et al., 2008]. In the literature, generally two main approaches are used to estimate the parameters of neuron models. The first makes use of membrane voltage recordings, and the second is based on ionic current recordings used in conductance-based models. In the first case, the parameters are determined simultaneously by using the membrane voltage only. Such methods are useful for different types of models, such as the leaky-integrate-and-fire, Izhikevich's, and multicompartmental models [Keren et al., 2005; Vanier and Bower, 1999]. However, one pitfall with this kind of method is that for complex models, excessive simulation times, as well as the implementation of computer clusters, are needed. To reduce the estimation time, it is sometimes possible to reduce the search space by making use of good a priori knowledge of the model's behavior [Pospischil et al., 2008]. For single-cell models such as the HH model, techniques based on ionic current recordings can be used to estimate the parameters. The best-known method is the voltage clamp introduced by Cole [1949] and later used by Hodgkin and Huxley [1952]. The voltage clamp is used by electrophysiologists to measure the ion currents across the membrane of excitable cells, such as neurons, while holding the membrane voltage at a set level. However, this method has many disadvantages due to the approximations it requires [Saïghi et al., 2008]. In particular, the separate estimation of these parameters is not really relevant, whereas their simultaneous estimation would be more efficient.

Moreover, due to the fabrication process, there are significant differences between the expected and actual outputs of the silicon neurons [Saïghi et al., 2008]. It is therefore necessary to further adjust the parameters. As hand tuning is very time consuming, due to the model's sensitivity to these parameter values and due to the large number of variables, an automated tuning of the parameters is mandatory. In Rossant et al. [2010], there is a software solution for the automatic fitting of spiking neuron models to electrophysiological recordings, but this fitting procedure could be very time consuming both in terms of computer simulations and code-writing. A systematic method for configuring VLSI network of spiking neurons was proposed by Neftci et al. [2011] that describes a parameter mapping technique that permits an automatic configuration of a VLSI neural network. For single cell models such as the HH model, techniques based on ionic current recordings can be used to estimate the parameters. We developed a new estimation method for the characterization of the HH formalism [Buhry et al., 2011]. It uses voltage clamp type recordings, but is based on the differential evolution algorithm. The parameters of an ionic channel are estimated simultaneously, such that the usual approximations of classical methods are avoided and all of the parameters of the model, including the time constant, can be correctly optimized. In a second step, this new estimation technique is applied to our neuromimetic analog integrated circuits (we will describe the estimation technique in Chapter 3) in order to simulate a specific neuronal activity with the neuromimetic circuits.

## 1.4 The network level: models used

As mentioned, a pre-synaptic neuron can change the membrane potential of the post-synaptic neuron through a synapse that can be excitatory or inhibitory. A synapse is excitatory if it causes an increase in the voltage of the postsynaptic neuron and inhibitory if it causes a decrease. Moreover, synapses can be plastic, which is the ability of the connection, or synapse, between two neurons to change in strength (or weight) in response to either use or disuse. In this section we describe the synapse model and general principles of the spike-timing-dependent plasticity rule used in the design of our spiking neural network platforms.

### 1.4.1 Chemical Synapses

To model the effect of the synapse, the equation for the voltage of the postsynaptic cell, is modified as follows:

$$C_{\mathrm{M}} \frac{dV_{\text{M-post}}}{dt} = -I_{ion} - I_{syn} \ , \qquad (\textbf{equation } 1.14)$$

where $I_{syn}$ is the current through the ion channel associated with the receptor for the neurotransmitter and $I_{ion}$ denotes the sum of the individual ionic currents of the model. $I_{syn}$ can be modeled as:

$$I_{syn} = g_{syn} \cdot (V_{post} - E_{syn}) \ , \qquad (\textbf{equation } 1.15)$$

where $E_{syn}$ is the reversal potential of the channel and gsyn is the conductance of the channel. Typically $E_{syn} = 0$ for an excitatory synapse and $E_{syn} < 0$ for an inhibitory synapse. The kinetics of the opening and closing of the ion channel is incorporated into the model for $g_{syn}$, as for the voltage gated ion channels. Following the arrival of an action potential at the presynaptic terminal, neurotransmitter molecules, T, are released into the synaptic cleft. These molecules are taken to bind to postsynaptic receptors. Letting r represent the fraction of bound receptors, these kinetics are described by the equation:

$$\frac{dr}{dt} = \alpha \cdot [T] \cdot (1 - r) - \beta \cdot r \ , \qquad (\textbf{equation } 1.16)$$

where [T] is the concentration of transmitter, $\alpha$ and $\beta$ are the forward and backward rate constants for transmitter binding [Destexhe et al., 1994]. Hence, the synaptic current is given by the equation:

$$I_{syn}(t) = \bar{g_{syn}} \cdot r(t)(V_{post}(t) - E_{syn}) \ , \qquad \textbf{(equation } 1.17)$$

where $\bar{g_{syn}}$ is the maximal conductance of the synapse. These equations provide a method for computing single synaptic currents (single synapse). Destexhe et al. [1998a] presents an algorithm, "Multiple Synapses", that can be used to simulate models with many synapses within the same compartment. This mechanism was implemented in the hardware by Bornat [2006].

### 1.4.2 Synaptic Plasticity

The dynamics of a Spiking Neural Network (SNN) and the formation of its connectivity are governed by synaptic plasticity. Plasticity rules formulate the modifications which occur in the synaptic transmission efficacy, driven by correlations in the firing activity of pre- and postsynaptic neurons. At the network level, spikes are generally processed as events, and the synaptic weight $W_{ji}$ (connection from neuron j to neuron i) varies over time, according to the learning rules.

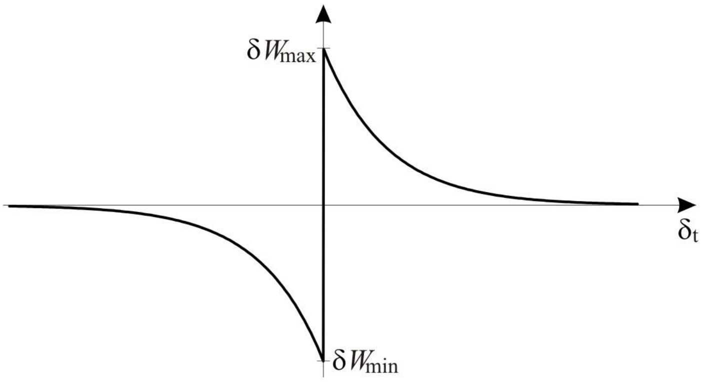


**Figure 1.8** – Variation of synaptic weight $\delta W_{ji}$ from neuron j (pre-synaptic) to neuron i (post-synaptic), as a function of the time interval $\delta$t = $t_i$ - $t_j$ ($t_i$: post-synaptic spike time, $t_j$: pre-synaptic spike time).

Figure 1.8 illustrates the principle of the Spike Timing Dependent Plasticity (STDP) algorithm[7]: when a post-synaptic spike arises after a pre-synaptic spike ($\delta$t $>$ 0), the connection is reinforced ("long-term potentiation", $\delta W_{ji} > 0$), whereas in the opposite case it is weakened ("long-term depression"). The change of the synapse plotted as a function of the relative timing of pre- and postsynaptic action potentials is called the STDP function or learning window and varies between synapse types. As with other forms of synaptic plasticity, it is widely believed that it underlies development and refinement of neuronal circuits during brain development, as well as the learning and information storage in the brain [Bi and Poo, 2001; Sjöström et al., 2008].

## 1.5 Neuromorphic Engineering: Spiking Neural Network platforms

Spiking neural networks (SNNs) fall into the third generation of neural network models, increasing the level of realism in a neural simulation. The idea is that neurons in the SNN do not fire at each propagation cycle (as happens with typical multi-layer perceptron networks), but rather, only fire when a membrane potential reaches a specific value. When a neuron fires, it generates a signal that travels to other neurons which, in turn, increase or decrease their potentials in accordance with this signal. Computational neuroscience commonly relies on software-based processing tools (NEURON, NEST, PCSIM, Brian, etc.). As mentioned in the introduction, neuromorphic engineering is a new interdisciplinary discipline that takes inspiration from biology, physics, mathematics, computer science and engineering to design analog, digital, and mixed-mode analog/digital VLSI and software systems to mimic neuro-biological architectures present in the nervous system. Some of these platforms are dedicated to the simulation of SNNs, and take into account the timing of input signals by precisely computing the neurons' asynchronous spikes. While software tools can be configured for different types of models [Brette et al., 2007; Hines and Carnevale, 1997], hardware-based SNNs are dedicated to a given type of model. In this section, we introduce the design approaches adopted for SNNs (we will focus on our SNN in Chapter 4) and then we review some hardware-based SNNs in terms of model complexity versus network size (the selection of the SNN is not exhaustive, but representative of the different design approaches). Finally, we describe the advantages and drawbacks for the different SNN hardware approaches while focusing on our choice.

[7] Details of the model used are present in the work of Belhadj [2010]

### 1.5.1 Hardware-based approach for SNN platforms

Hardware approaches have interesting properties in terms of computation time but a higher development cost and constraints on the computed models. Thus these systems are generally dedicated to an application. Applications can range from the investigation of adaptation and plasticity phenomena in networks to experiments on hybrid biological/artificial networks. For such systems, designers generally consider simplified neuron models and balance imprecision by a higher number of neurons in the network [Fieres et al., 2006; Indiveri et al., 2006]. Another approach is to reduce the number of the parameters in the system [Farquhar and Hasler, 2005], or to model neurons populations [Fieres et al., 2006; Indiveri et al., 2006; Renaud et al., 2004]. There are other approaches that limit synaptic connectivity, while others guarantee an "all-to-all" connection but with less neurons. Finally, systems differ by their properties in terms of computation time, as the aim of some systems is to simulate SNNs "as fast as possible" [Indiveri et al., 2006], and others guarantee a fixed simulation timescale [Bornat et al., 2005; Fieres et al., 2006]. In the next section[8], we describe several types of hardware-based SNNs; this review highlights the diversity of possible solutions.

### 1.5.2 A review of hardware-based SNN

In recent years, a few hardware-based SNN systems [Binczak et al., 2006; Glackin et al., 2005; Graas et al., 2004; Hasler et al., 2007; Indiveri and Fusi, 2007; Liu and Douglas, 2004; Renaud et al., 2007; Schemmel et al., 2008; Sorensen et al., 2004; Vogelstein et al., 2004], as well as pioneer platforms have been developed [Jung et al., 2001; Le Masson et al., 2002; Mahowald and Douglas, 1991]. The systems make use of multi-compartmental models [Hasler et al., 2007], point neuron conductance-based models [Binczak et al., 2006; Le Masson et al., 2002; Mahowald and Douglas, 1991; Renaud et al., 2010; Sorensen et al., 2004], or threshold type models [Glackin et al., 2005; Indiveri and Fusi, 2007; Jung et al., 2001; Liu and Douglas, 2004; Schemmel et al., 2008; Vogelstein et al., 2004], as shown in Figure 1.9.

Figure 1.9 also shows the diversity of SNN platforms in terms of design. Some platforms can be used for hybrid network experiments [Jung et al., 2001; Le Masson et al., 2002; Renaud et al., 2010; Sorensen et al., 2004]. Only three platforms are fully digital

[8] The contents are referenced by the work of [Renaud et al., 2010].

hardware systems [Glackin et al., 2005; Graas et al., 2004], using FPGAs (Field Programmable Gate Array) circuits (for large-scale LIF SNNs or small-scale HH SNNs) or ARM core [Lester and Furber, 2011].

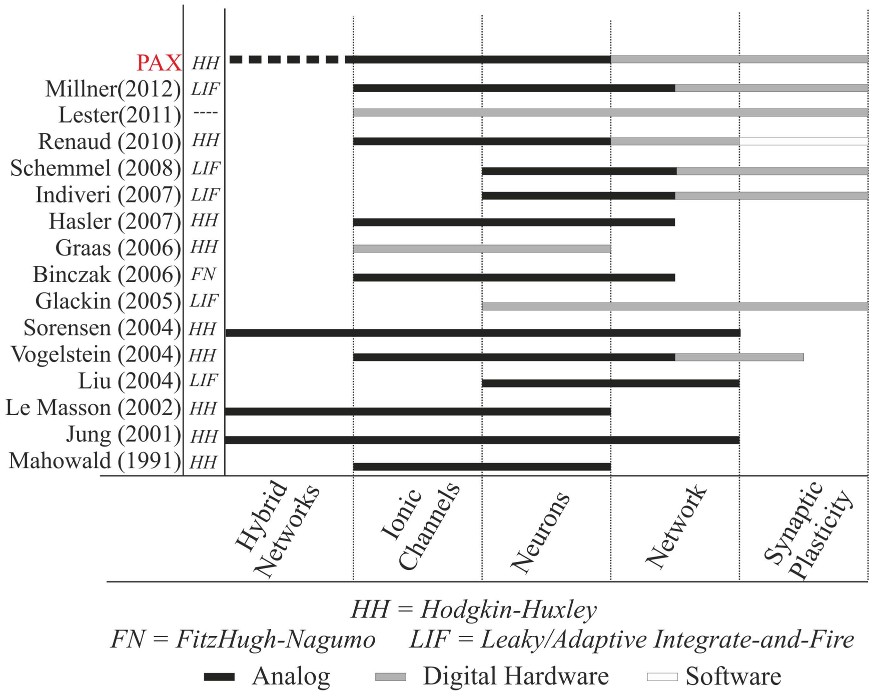


**Figure 1.9** – Analog/digital and hardware/software distribution in state-of-the-art SNN systems; Horizontally: simulation features. Figure inspired by Renaud et al. [2010].

All of the other solutions rely on analog computation using specifically designed integrated circuits at the neuron level, even for plasticity. The neurons are organized into networks of various sizes. A comparison of the SNN hardware platforms, expressed in terms of model complexity versus network size, is shown in Figure 1.10.

In the end, the architecture of a given SNN platform is the result of a compromise between computational cost and model complexity (biological plausibility), the latter of which also constrains the achievable network size (as shown in Figure 1.10). For large-scale SNNs, LIF neuron models are generally implemented (or the Adaptive-Exponential Integrate-and-Fire Neuron Model [Millner et al., 2012] in the case of the BrainScales project), using simple analog cells to simulate a point neuron.

In our case, real-time (electrical time = biological time) and biologically relevant neuron models make it possible to construct mixed living artificial networks, in which the silicon neurons are interconnected with the biological cells to form "hybrid networks"

[Le Masson et al., 2002]. Our SNN is based on the above hardware solution (see Figure 1.9, more details are provided in Chapter 4): it is a mixed hardware/software platform specifically designed to simulate spiking neural networks using conductance-based analog silicon neurons (120 neurons) and synaptic connections with STDP rules computed on FPGA [Belhadj, 2010].

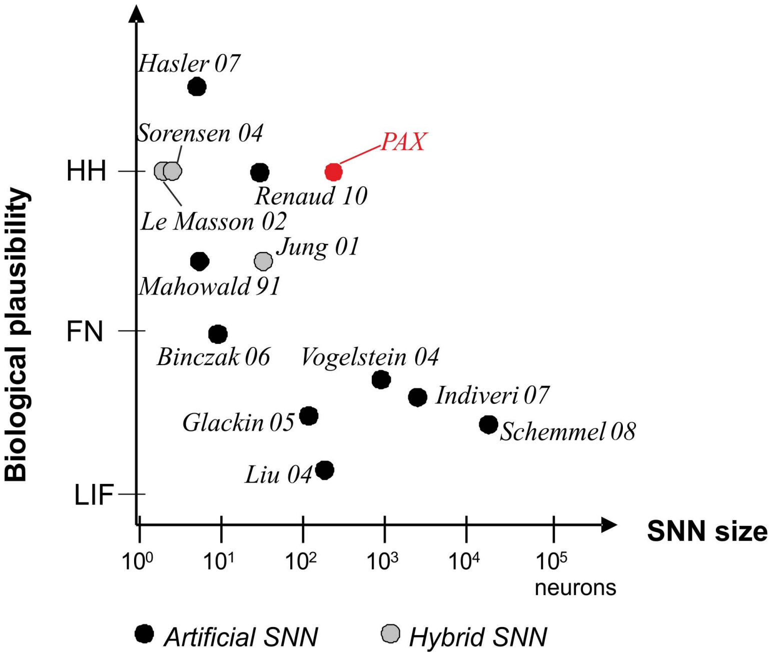


**Figure 1.10** – Comparison of SNN hardware platforms, expressed in terms of model complexity versus network size. Figure inspired by Renaud et al. [2010].

In the case of an analog mode implementation, the signals are available as continuous variables, both in time and value. As a consequence, the simulation timescale can be precisely determined, for example, in real time or accelerated time. An analog SNN has two distinct advantages: a higher integration density, since a single wire encodes one signal (instead of N wires needed to represent a digital signal with N-bits), and a higher computational power, which is obtained by exploiting the intrinsic computational features of electronic elements such as transistors. Analog approaches have interesting properties in terms of computation time but a higher development cost and constraints on the computed models.

Digital design is characterized by much lower manufacturing costs, and has the advantages of an improved time-to-market performance and straightforward re-configurability. However, when the system reaches the network level or integrates plasticity rules, digital hardware is used for connectivity computation and/or control. Event-based protocols are used to interface the neurons' activity (spike events) to the connectivity calculation element. An increasing number of neuromimetic systems present this type of mixed Analog/Digital architecture. In the end, the architecture of a given SNN platform is the result of a compromise between computational cost and model complexity.

## 1.6 AS2N team: related research projects

The goal of our research team "Architecture of Silicon Neural Networks" is to design and use integrated circuits (IC) in which the components and architecture are neuromimetic (i.e. they mimic biological neural systems). This research activity has an interdisciplinary orientation. There are both national and international collaborations within the team which includes neuroscientists as well as computer scientists and physicists. This section describes the research project related to this research.

**FACETS 2005-2009 (European project)**

The aim of the FACETS project (Fast Analog Computing with Emergent Transient States (FP6, 2004-IST-FETPI 15879)) was to address the unsolved question of how the brain computes based on the concerted work of neuroscientists, computer scientists, engineers and physicists. It combined a substantial fraction of the European groups working in the field into a consortium of 13 groups from Austria, France, Germany, Hungary, Sweden, Switzerland and the UK. About 80 scientists joined their efforts over a four year period, starting in September 2005. The goal of the FACETS project was to create a theoretical and experimental foundation for the practical realization of novel computing hardware, which exploits the concepts experimentally observed in biological nervous systems. In summary, the goals of the project can be formulated as follows:

- To provide biological input data from in-vivo and in-vitro measurements at the cell and network level, to set-up a large-scale computer data base for neural cell characterization;
- To use large-scale computer based models to test the concepts and benchmarks developed in the project, to develop a common data model for neural simulations;
- To build and use very large-scale hardware models based on the above results;
- To evaluate new computing paradigms using the FACETS benchmark for problems in vision.

It is rare for a project of this size to be carried out within the context of brain-science related work in Europe, in particular with such a strong interdisciplinary component.

**ECRéN 2009-2011 (National project)**

The aim of this project was to investigate the transfer functions of neural networks with combinatorial methods and Integrated Circuits (IC). The models and synaptic input patterns for neurons included data from in vivo and in vitro biological experiments provided by Destexhe's group (UNIC Lab). These models will be implemented in ICs that are able to generate arbitrary network patterns for a combinatorial exploration in real time. Then, the transfer function for each network will be computed. This project will produce a large quantity of data for further sorting and analysis. The analysis, in collaboration with Destexhe's group, will try establish general rules for the relationship between structure and functionality. This first part of the project should result in the foundation of a database that will be made available to the scientific community. Then, using the same protocol, this project will be extended to networks with plasticity rules such as STDP (Spike Timing Dependent Plasticity).

**FACETS-ITN 2009-2014 (European project)**

FACETS-ITN (FP7-PEOPLE-ITN-2008-237955) is a research and training network involving partners from 11 universities and research centers, three industrial companies and one semi-industrial research center from 6 European countries. It combines competencies in neurobiology, computational neuroscience, information science, physics and electrical engineering. The scientific goal is to experimentally and theoretically explore the structure and the computational principles of biological neural circuits using in-vitro and in-vivo neurobiological experiments as well as analytical approaches, model building and simulation techniques. The concepts of learning and plasticity are of particular importance. Based on the input from biology and modeling, it is expected that these principles will be used to prepare the grounds for novel hardware based computing devices. These devices are going to be built in the form of large-scale demonstrators as part of the research plan. Within the training network, 22 selected Ph.D. students (one position for this Ph.D. thesis) were integrated into an existing international research environment and receive a high level of interdisciplinary training. The training comprises an intense exchange and visiting program, specific training workshops for all of the scientific areas covered as well as in non-scientific key competencies (see training experiences in the

Appendix A.3). Community building among Ph.D. students from different disciplines includes students from other projects, in particular the current FACETS integrated project and its planned successor.

As mentioned in the introduction, the main focus of my Ph.D. thesis in the FACETS project is the emulation of cortical cell types (employing experimental electrophysiological data of these cells as references), using a specific VLSI neural circuit to simulate the models studied as references in the FACETS project at the single cell level. The real-time intrinsic properties of the neural circuits, that precisely compute neuron conductance-based models, will allow a systematic and detailed exploration of the models, whereas the physical and analog aspect of the simulations, opposed to a software simulation, will provide inputs for the development of the neural hardware at the network level. The second focus is to contribution to the design of a mixed hardware-software platform, specifically designed for the simulation of spiking neural networks, using conductance-based models that can be used for hybrid artificial neural networks.

## 1.7 Conclusion

In this first chapter, we introduced several elementary notions of neuroscience; in particular, we have seen how "real" neurons are extremely complex biophysical and biochemical entities. The Hodgkin-Huxley model describes channels and ion current flow. It is the starting point for detailed neuron models which, in general, include more than three types of currents. Thus, we described the most successful and widely-used neuron models classically implemented in hardware simulators and then we explained the choices made by our team to use the Hodgkin-Huxley formalism to model the electrical activity of neurons. Finally, after the presentation of the models used for the network level, we reviewed the hardware simulation platforms for spiking neural networks in terms of model complexity versus network size, focusing on our SNN hardware approaches, and then we presented the research activities of our team (AS2N) included the European FACETS-ITN project that has supported this research. As mentioned, our SNN platform relies on analog hardware to compute neuronal activity. Therefore in the next chapter, we propose a simplified version of the HH model and the appropriate parameter sets for the FS, RS, IB, and LTS neurons that can be implemented in our analog neuromimetic chip.

# Chapter 2

# Cortical neuron models: parameter extraction for a silicon neuron

As mentioned in the introduction, one of the objectives of this research is the emulation of the four most prominent biological cells (the FS, RS, IB, and LTS neurons) into an analog neuromimetic integrated circuit dedicated to cortical neuron simulations. Therefore, the aim of this chapter is to introduce the methods used to obtain the appropriate parameter sets for the neurons that can be implemented into our analog neuromimetic chip. In section 2.1, we introduce the HH formalism and then the simplified model that was implemented into the silicon neuron [Bornat, 2006]. Thus, we describe the steps used to extract the parameters from the HH model to the one implemented in the VLSI neurons. We use the HH model presented by Hansel et al. [1993] as a reference and we validate the parameter extraction steps through a bifurcation analysis that shows that the simplified HH model shares the dynamics of the HH model. Finally, in section 2.4, we describe the intrinsic electrophysiological properties of the four cortical neurons: the FS, RS, IB, and LTS neurons. We propose our simplified models for each neuron class, which are based on the complete HH type models proposed by Pospischil et al. [2008]. The models contain the minimal set of voltage-dependent currents to account for the experimental electrophysiological data.

## 2.1 The Hodgkin-Huxley formalism for the silicon neuron

The model introduced by Hodgkin and Huxley [1952] incorporated the results of their voltage-clamp experiments. They used the voltage-clamp technique to record the ionic currents generated at different voltages and thus infer how these ionic currents can be dynamically modulated by voltage. Many different types of voltage-dependent ion channels have been identified and are responsible for a rich repertoire of electrical behavior that is essential for neuronal function [Llinás, 1988]. The sensitivity of some ion channels to voltage is a fundamental property that constitutes the core mechanism underlying the electrical excitability of membranes, and is still today an important matter of investigation. This section[1] first describes the HH model equations and then the conductance-based models which can incorporate as many different ion channel types as are known for the particular cell being modeled. We focus on the types of ionic channels, responsible for the generation of spikes, for the four cortical neurons: the FS, RS, IB, and LTS neurons.

### 2.1.1 Hodgkin-Huxley equations

As stated in section 1.2.2, in the equivalent electrical circuit (Figure 2.1) for the Hodgkin-Huxley model of the squid giant axon, the capacitance[2] is used to model the charge storage capacity in the cell membrane, resistors are used to model the various types of ion channels embedded in the membrane, and batteries are used to represent the electrochemical potentials established by differing intra and extracellular ion concentrations.

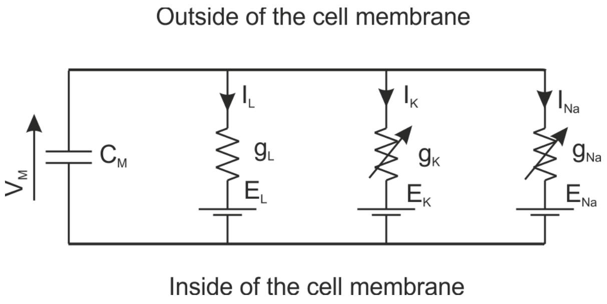


**Figure 2.1** – Equivalent circuit representation of a cell membrane.

The current flowing across the membrane is integrated in the membrane capacitance,

[1] All of the contents for the section "The Hodgkin-Huxley formalism for the silicon neuron" are referenced from Destexhe and Huguenard [2000], if not specially mentioned.

[2] The plasma membrane is made up of a molecular lipid bilayer. Inserted in this bilayer, there are membrane proteins that have the important function of transporting materials across the membrane. The lipid bilayer acts like an insulator separating two conducting media: the external medium of the axon and the internal medium or axoplasm. This geometry constitutes an electric capacitor where the two conducting plates are the ionic media and the membrane is the dielectric ($\epsilon$). The capacitance (C) of a capacitor increases with the area (A) of the plates and decreases with the separation between the plates (d) according to the equation: C=$\epsilon$·A/d. In the case of the membrane, it is more convenient to define the capacitance as being independent of the amount of area involved and call it the specific capacitance $C_{\rm M}$ which is defined as the capacitance per unit area. As the thickness is only 25 A, the specific capacitance of the membrane is very high, close to $1\mu F/cm^2$.

according to the expression:

$$C_{\mathrm{M}}\frac{dV_{\mathrm{M}}}{dt} = -I_{Na} - I_K - I_{\mathrm{L}} \,, \qquad \textbf{(equation 2.1)}$$

where I$_{Na}$, I$_K$ and I$_L$ are the sodium, potassium and leakage currents, respectively:

$$\begin{cases} I_{Na} = g_{Na}(V_{\mathrm{M}}) \cdot (V_{\mathrm{M}} - E_{\mathrm{Na}}) \\ I_K = g_K(V_{\mathrm{M}}) \cdot (V_{\mathrm{M}} - E_{\mathrm{K}}) \\ I_L = g_L \cdot (V_{\mathrm{M}} - E_{\mathrm{L}}) \,. \end{cases} \qquad \textbf{(equation 2.2)}$$

The next step is to specify how the conductances g$_{Na}$(V$_M$) and g$_K$(V$_M$) depend on the membrane potential. Hodgkin and Huxley [1952] hypothesized that ionic currents result from the assembly of several independent gating particles that must occupy a given position in the membrane to allow the flow of Na$^+$ or K$^+$ ions. Each gating particle can be in either side of the membrane and bears a net electronic charge such that the membrane potential can switch its position from the inside to the outside or vice-versa. The transition from these two states is therefore voltage-dependent, according to the diagram:

$$(outside) \underset{\beta(V_M)}{\overset{\alpha(V_M)}{\rightleftarrows}} (inside) \,, \qquad \textbf{(equation 2.3)}$$

where $\alpha$ and $\beta$ are respectively the forward and backward rate constants for the transitions from the outside to the inside position in the membrane. If x is defined as the fraction of particles in the inside position, and (1-x) as the fraction outside, one obtains the first-order kinetic equation:

$$\frac{dx}{dt} = (1-x) \cdot \alpha(V_{\mathrm{M}}) - x \cdot \beta(V_{\mathrm{M}}) \,. \qquad \textbf{(equation 2.4)}$$

If one assumes that particles must occupy the inside position to conduct ions, then the conductance must be proportional to some function of x. In the case of the squid giant axon, Hodgkin and Huxley found that the nonlinear behavior of the Na$^+$ and K$^+$ currents was best fit by assuming that:

$$\begin{cases} g_{Na}(V_{\mathrm{M}}) = \bar{g_{Na}} \cdot m^3(V_{\mathrm{M}}) \cdot h(V_{\mathrm{M}}) \\ g_K(V_{\mathrm{M}}) = \bar{g_K} \cdot n^4(V_{\mathrm{M}}) \,, \end{cases} \qquad \textbf{(equation 2.5)}$$

where $\bar{g_{Na}}$ and $\bar{g_K}$ are the maximal values of the conductances and m, h, and n represent the fraction of three different types of gating particles in the inside of the membrane. The three variables m, h, and n evolve according to the differential **equation** 2.4. The assembly of three m-type and one h-type gating particle(s) is required for Na$^+$ ions to flow through the membrane, while the assembly of four n-type gating particles is necessary for the flow of K$^+$ ions.

When it was later established that ionic currents are mediated by the opening and closing of ion channels, the gating particles were reinterpreted as gates inside the pore of the channel. Thus, the reinterpretation of Hodgkin and Huxley's hypothesis was that the pore of the channel is controlled by four gates, that these gates operate independently of each other, and that all four gates must be open in order for the channel to conduct ions. The variables m and n are called gating variables for activation, h is called a gating variable for inactivation.

Taking all of the above steps together, the following set of equations can be written:

$$\begin{cases} C_{\mathrm{M}}\dfrac{dV_{\mathrm{M}}}{dt} = -g_L\cdot(V_{\mathrm{M}}-E_{\mathrm{L}}) - \bar{g_{Na}}\cdot m^3(V_{\mathrm{M}})\cdot h(V_{\mathrm{M}})\cdot(V_{\mathrm{M}}-E_{\mathrm{Na}}) - \bar{g_K}\cdot n^4(V_{\mathrm{M}})\cdot(V_{\mathrm{M}}-E_{\mathrm{K}}) \\ \dfrac{dm}{dt} = (1-m)\cdot\alpha_m(V_{\mathrm{M}}) - m\cdot\beta_m(V_{\mathrm{M}}) \\ \dfrac{dh}{dt} = (1-h)\cdot\alpha_h(V_{\mathrm{M}}) - h\cdot\beta_h(V_{\mathrm{M}}) \\ \dfrac{dn}{dt} = (1-n)\cdot\alpha_n(V_{\mathrm{M}}) - n\cdot\beta_n(V_{\mathrm{M}})\,. \end{cases}$$

(**equation** 2.6)

The various functions $\alpha$ and $\beta$ are empirical functions of $V_M$ that have been adjusted by Hodgkin and Huxley to fit the experimental data.

The Hodgkin-Huxley model is often written in a form that is more convenient to fit the experimental data by rewriting **equation** 2.4 in the equivalent form:

$$\frac{dx}{dt} = \frac{1}{\tau_x(V_{\mathrm{M}})}\left(x_\infty(V_{\mathrm{M}}) - x\right), \qquad \textbf{(equation 2.7)}$$

where

$$\begin{cases} x_\infty(V_{\mathrm{M}}) = \dfrac{\alpha(V_{\mathrm{M}})}{\alpha(V_{\mathrm{M}})+\beta(V_{\mathrm{M}})} \\ \tau_x(V_{\mathrm{M}}) = \dfrac{1}{\alpha(V_{\mathrm{M}})+\beta(V_{\mathrm{M}})}\,. \end{cases} \qquad \textbf{(equation 2.8)}$$

For a fixed voltage $V_{\mathrm{M}}$, the variable x approaches the value $x_\infty(V_{\mathrm{M}})$ with a time constant $\tau_x(V_{\mathrm{M}})$. The Hodgkin-Huxley equations then become:

$$\begin{cases} C_{\mathrm{M}}\dfrac{dV_{\mathrm{M}}}{dt} = -g_L \cdot (V_{\mathrm{M}} - E_{\mathrm{L}}) - \bar{g_{Na}} \cdot m^3(V_{\mathrm{M}}) \cdot h(V_{\mathrm{M}}) \cdot (V_{\mathrm{M}} - E_{\mathrm{Na}}) - \bar{g_K} \cdot n^4(V_{\mathrm{M}}) \cdot (V_{\mathrm{M}} - E_{\mathrm{K}}) \\ \dfrac{dm}{dt} = \dfrac{1}{\tau_m(V_{\mathrm{M}})}\left(m_\infty(V_{\mathrm{M}}) - m\right) \\ \dfrac{dh}{dt} = \dfrac{1}{\tau_h(V_{\mathrm{M}})}\left(h_\infty(V_{\mathrm{M}}) - h\right) \\ \dfrac{dn}{dt} = \dfrac{1}{\tau_n(V_{\mathrm{M}})}\left(n_\infty(V_{\mathrm{M}}) - n\right), \end{cases} \qquad (\textbf{equation } 2.9)$$

where $m_\infty(V_{\mathrm{M}})$ is the steady-state activation and $\tau_m(V_{\mathrm{M}})$ is the activation time constant of the sodium current, and $n_\infty(V_{\mathrm{M}})$ and $\tau_n(V_{\mathrm{M}})$ represent the same terms for the potassium current. In the case of h, $h_\infty(V_{\mathrm{M}})$ and $\tau_h(V_{\mathrm{M}})$ are the steady-state inactivation and the inactivation time constant of the sodium current, respectively.

### 2.1.2 Conductance-based neuron model

Conductance-based models (also called the Hodgkin-Huxley formalism) are based on an equivalent circuit representation of a cell membrane as first put forth by Hodgkin and Huxley [1952]. Conductance-based models are the simplest possible biophysical representation of an excitable cell, such as a neuron, in which its protein molecule ion channels are represented by conductances and its lipid bilayer by a capacitor.

We already know that many different types of voltage-dependent ion channels have been identified, thus the standard formulation for a conductance-based model is given as:

$$C_{\mathrm{M}}\frac{dV_{\mathrm{M}}}{dt} = -\sum I_{ion} + I_S\ , \qquad (\textbf{equation } 2.10)$$

where $V_{\mathrm{M}}$ is the membrane potential, $C_{\mathrm{M}}$ is the membrane capacitance, and $I_S$ is a stimulation or synaptic current. $I_{ion}$ is the current for a given channel type and its associated equation is:

$$I_{ion} = g_{ion} \cdot a^p(V_{\mathrm{M}}) \cdot b^q(V_{\mathrm{M}}) \cdot (V_{\mathrm{M}} - E_{\mathrm{ion}})\ , \qquad (\textbf{equation } 2.11)$$

where $g_{ion}$ is the maximum conductance, $E_{\mathrm{ion}}$ is the ion-specific reverse potential, p and q are integers, and a and b represent the activation and inactivation terms, respectively.

They are dynamic functions which describe the permeability of membrane channels to its specific ion that are described by the equations:

$$\begin{cases} \dfrac{da}{dt} = \dfrac{1}{\tau_a(V_{\text{M}})}\,(a_\infty(V_{\text{M}}) - a) \\ \dfrac{db}{dt} = \dfrac{1}{\tau_b(V_{\text{M}})}\,(b_\infty(V_{\text{M}}) - b)\,, \end{cases} \qquad (\textbf{equation } 2.12)$$

where $\tau_a$ and $\tau_b$ are the convergence time constants which are dependent on the membrane voltage $V_{\text{M}}$. According to the first-order differential **equation** 2.12, a and b relax back towards their associated steady-state values $a_\infty$ and $b_\infty$ which are sigmoidal functions of $V_{\text{M}}$:

$$\begin{cases} a_\infty(V_{\text{M}}) = \dfrac{1}{1 + exp\left(-\frac{V_{\text{M}} - V_{offset,a}}{V_{slope,a}}\right)} \\ b_\infty(V_{\text{M}}) = \dfrac{1}{1 + exp\left(\frac{V_{\text{M}} - V_{offset,b}}{V_{slope,b}}\right)}\,. \end{cases} \qquad (\textbf{equation } 2.13)$$

In **equation** 2.13, $V_{slope}$ and $V_{offset}$ are respectively the slope and a translational offset of the sigmoid function. The values of these terms are directly observable using voltage-clamp experiments and, as it is common for HH variables [Hille, 1991], the steady-state curve is given by a sigmoidal function and the time constant curve is given by a bell-shaped function as shown in Figure 2.2.

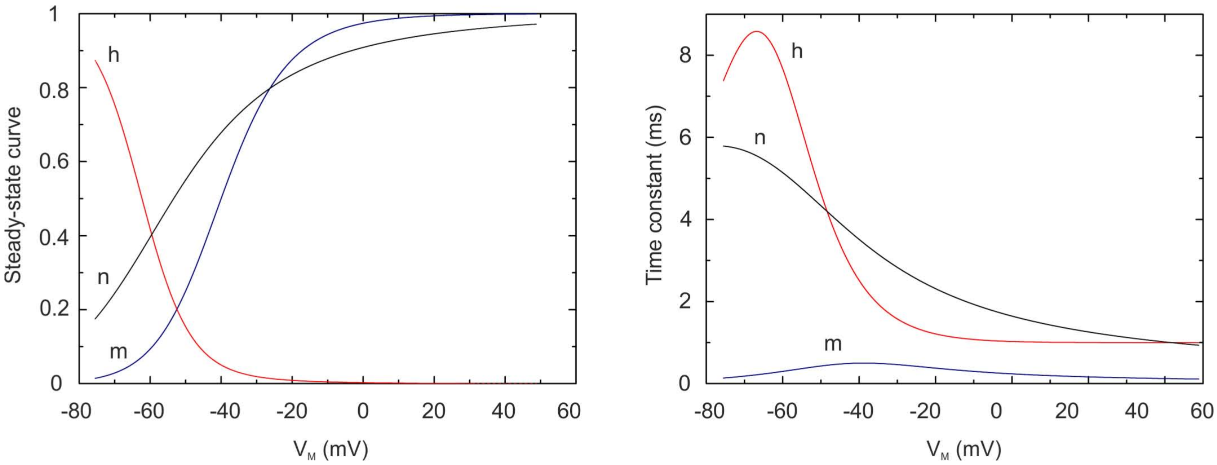


**Figure 2.2** – Steady-state and time constant curves for the three variables m, n, h in the HH model.

The data presented in Figure 2.2 are extracted from the HH model presented by

Hansel et al. [1993]. Note that the steady-state variable for an activation variable is a monotonically increasing function of voltage, while steady-state inactivation variables are a monotonically decreasing function of voltage.

Thus, conductance-based models consist of a set of ordinary differential equations (ODEs), as derived from current flow in a circuit representation following Kirchoff's laws. The number of differential equations in the set of model equations depends on the number of different ion channel types being represented with their particular activation and inactivation gating variables. This is illustrated in Figure 2.3. The voltage dependence or non-constant nature of the conductance of ion channels is captured using "activation" and "inactivation" gating variables which are described using first-order kinetics. This is represented with an arrow across the resistor in the schematic representation provided in Figure 2.3.

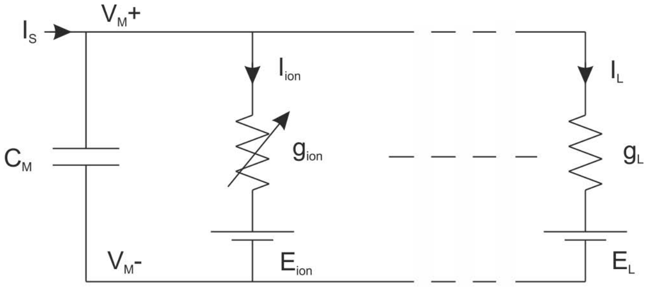


**Figure 2.3** – Conductance based model: equivalent circuit representation of a cell membrane.

The HH model describes the sodium, potassium and leakage currents, with p = 3 and q = 1; p = 4 and q = 0; p = 0 and q = 0 respectively, in **equation** 2.11. These channels are responsible for action potential generation. A model based on these three conductances reproduces well the intrinsic firing characteristics of FS cells of the ferret visual cortex in vitro [Pospischil et al., 2008]. Although the voltage-gated $Na^+$ and $K^+$ channels in the squid axon described by Hodgkin and Huxley have been found in almost every type of neuron examined, several other kinds of channels have also been identified. For example, most neurons contain voltage-gated $Ca^{2+}$ channels that open in response to membrane depolarization. A strong electrochemical gradient drives $Ca^{2+}$ into the cell and thus these channels give rise to an inward $I_{Ca}$ [Kandel et al., 2000].

Therefore, for more complex activity patterns, additional channels such as slow potassium currents ($I_M$) for spike-frequency adaptation (RS cells: $I_{Na}$, $I_K$, $I_L$ and $I_M$), L-Type calcium currents for bursting (IB cells: $I_{Na}$, $I_K$, $I_L$, $I_M$ and $I_{Ca(L)}$) and T-type calcium currents (LTS cells: $I_{Na}$, $I_K$, $I_L$, $I_M$ and $I_{Ca(T)}$) have to be taken into account [Pospischil et al., 2008]. Although these HH type models sometimes exhibit much more complex dynamics than the original HH model, they share common nonlinear characteristics and dynamics with the HH in many aspects.

### Slow potassium current for spike-frequency adaptation

Spike-frequency adaptation is a widespread neurobiological phenomenon that is exhibited by almost any type of neuron that generates action potentials and which may play an important role in neural information processing. Within the large variety of mechanisms responsible for spike-frequency adaptation, ionic currents that influence spike generation are of particular importance. A slow non-inactivating K$^+$ current was described by Yamada et al. [1989]:

$$I_{\text{M}} = \bar{g_M} \cdot p(V_{\text{M}}) \cdot (V_{\text{M}} - E_{\text{K}}), \qquad \textbf{(equation } 2.14)$$

where $\bar{g_M}$ denotes the maximum conductance and $E_{\text{K}}$ is the reversal potential. M-type currents are mainly activated during a spike. Between spikes, they deactivate slowly as determined by their time constant. The activation of M-type currents causes spike-frequency adaptation, since as potassium currents they decrease the sensitivity of the spike generator to input currents.

### L-type calcium current

The L-type calcium channel is a type of voltage-dependent calcium channel ("L" stands for long-lasting, referring to the length of activation). A type of bursting was modeled by the high-threshold Ca$^{2+}$ current, which was described by Reuveni et al. [1993]:

$$I_{Ca(L)} = g_{\bar{Ca}(L)} \cdot q^2(V_{\text{M}}) \cdot r(V_{\text{M}}) \cdot (V_{\text{M}} - E_{\text{Ca}}), \qquad \textbf{(equation } 2.15)$$

where $g_{\bar{Ca}(L)}$ is the maximum conductance of the L-type Ca$^{2+}$ calcium current, and $E_{\text{Ca}}$ is the reversal potential.

### T-type calcium current

The T-type Ca$^{2+}$ calcium current (also called a low-threshold Ca$^{2+}$ current) is responsible for the generation of bursts of action potentials in many cell types, such as thalamic neurons (Jahnsen, 1984). The T-type Ca$^{2+}$ current has activation (s) and inactivation (u) characteristics similar to the fast Na$^+$ current but is slower, and its voltage range for activation and inactivation typically occurs around resting potential. The low-threshold $Ca^{2+}$ current is given by:

$$I_{Ca(T)} = g_{\bar{Ca}(T)} \cdot s_{\infty}^{2}(V_{\mathrm{M}}) \cdot u(V_{\mathrm{M}}) \cdot (V_{\mathrm{M}} - E_{\mathrm{Ca}}), \qquad \text{(equation 2.16)}$$

where $g_{\bar{Ca}(T)}$ is the maximal conductance of the T-type $Ca^{2+}$ calcium current, and $E_{\mathrm{Ca}}$ is the reversal potential. Note that the activation variable s is considered here at steady-state because activation is fast compared to inactivation. This T-type $Ca^{2+}$ current model was also used with an independent activation variable, but produced very similar results to the model with activation at steady-state [Destexhe et al., 1998b].

## 2.2 Parameter extraction for the model implemented in the VLSI neuron

In a previous work, our team designed several neuromimetic chips, including the Galway chip [Bornat, 2006] that we used for this thesis. The Galway chip includes analog operators to compute the Hodgkin-Huxley formalism and multi-synapses to build a neural network. This section first describes the model implemented in the Galway chip and then the steps used to extract the parameter for the chip. We use the HH model presented by Hansel et al. [1993] as a reference.

### 2.2.1 The model implemented in the VLSI neuron

Our analog IC was optimized to reproduce a large variety of neuron behaviors using tunable parameters. The Galway chip includes five channel types: leakage, sodium, potassium, slow potassium, and calcium. By combining these channels, we can model a large variety of neurons.

Figure 2.4 shows the block diagram of the ionic current generator (which implements **equation** 2.11 except for the dynamic of the time constant that is fixed), where each block corresponds to a subcircuit performing the indicated operation and parameterized by tunable parameters representing the conductance model parameters. The choice of the fixed time constant in the VLSI implementation has been made to reduce the silicon area required by the neuron implementation in the chip. Consequently, the only difference between the VLSI model and the conductance-based model is the approximation used for the dynamic of the gating variable.

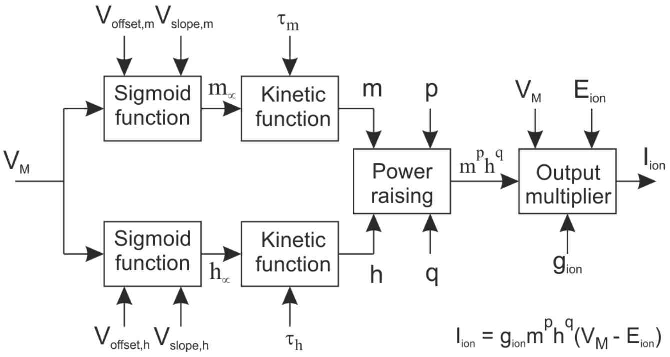


**Figure 2.4** – Block diagram of the ionic current generator [Saïghi et al., 2011].

### 2.2.2 Parameter extraction technique

As our goal is to reproduce the behavior of biological neurons, we need the parameters for the simplified model implemented in the VLSI neuron. These parameters values, which are adapted to be implemented in the VLSI neuron, are obtained through the following three steps:

1. calculating $\alpha_x(V_\mathrm{M})$ and $\beta_x(V_\mathrm{M})$ over the range $V_\mathrm{M}$ = [-100, 100] mV. The x subscript represents the activation or inactivation term,

2. identifying the $V_{offset,x}$ and $V_{slope,x}$ terms in the sigmoid function, which is equal to $x_\infty(V_\mathrm{M}) = \alpha_x(V_\mathrm{M})/(\alpha_x(V_\mathrm{M}) + \beta_x(V_\mathrm{M}))$,

3. in addition, $\tau_x$ is calculated from $\tau_x(V_\mathrm{M}) = 1/(\alpha_x(V_\mathrm{M}) + \beta_x(V_\mathrm{M}))$ at $V_\mathrm{M}$ about -70 mV (resting potential). Empirically, the choice to calculate $\tau_x$ at the resting potential is due to the fact that we focus on action potential generation. However, we see in the next section (2.3) how this approximation changes the complete dynamics of the systems through the bifurcation analysis.

The parameter values and the empirical functions $\alpha$ and $\beta$ extracted from the HH model [Hansel et al., 1993] are summarized below. The HH model contained $I_{Na}$, $I_K$ and $I_L$ with the kinetics described in **equation** 2.17 and **equation** 2.18 for the sodium and potassium current, respectively:

$$\begin{cases} \alpha_m(V_{\text{M}}) = \dfrac{0.1 \cdot (V_{\text{M}} + 40)}{1 - exp[-(V_{\text{M}} + 40)/10]} \\ \beta_m(V_{\text{M}}) = 4 \cdot exp[-(V_{\text{M}} + 65)/18] \\ \alpha_h(V_{\text{M}}) = 0.07 \cdot exp[-(V_{\text{M}} + 65)/20] \\ \beta_h(V_{\text{M}}) = \dfrac{1}{exp[-(V_{\text{M}} + 35)/10]}, \end{cases} \qquad \textbf{(equation } 2.17)$$

$$\begin{cases} \alpha_n(V_{\text{M}}) = \dfrac{0.01 \cdot (V_{\text{M}} + 55)}{1 - exp[-(V_{\text{M}} + 55)/10]} \\ \beta_n(V_{\text{M}}) = 0.125 \cdot exp[-(V_{\text{M}} + 65)/80]. \end{cases} \qquad \textbf{(equation } 2.18)$$

The model parameters are: $\bar{g_K}$ = 36 mS/$cm^2$, $\bar{g_{Na}}$ = 120 mS/$cm^2$, $\bar{g_L}$ = 0.3 mS/$cm^2$, $E_K$ = -77 mV, $E_{Na}$ = 50 mV, $E_L$ = -54.4 mV and $C_{\text{M}} = 1\mu F/cm^2$.

On the other end, the parameter values adapted (using the former approximation) to be implemented in the VLSI are summarized in Table 2.1. The specific capacitance of the membrane is $C_{\text{M}} = 1\mu F/cm^2$.

**Table 2.1** – Parameters of the simplified HH biological neuron model.

| | Sodium | Potassium | Leak |
|---|---|---|---|
| $E_{ion}\,(mV)$ | 50 | -77 | -54.4 |
| $g_{ion}\,(mS/cm^2)$ | 120 | 36 | 0.3 |
| $V_{slope,act}\,(mV)$ | -39.6 | -52.4 | |
| $V_{offset,act}\,(mV)$ | 9 | 16.2 | |
| $\tau_{act}\,(ms)$ | 0.065 | 1 | |
| $V_{slope,inact}\,(mV)$ | -62.2 | | |
| $V_{offset,inact}\,(mV)$ | 6.9 | | |
| $\tau_{inact}\,(ms)$ | 1.3 | | |

To validate our approximation, we first compare the voltage membrane simulations between the Hodgkin and Huxley model presented by Hansel et al. [1993] and the simplified HH model using the former approximation (Figure 2.5). We chose a simulation time of 200 ms and a stimulation current of 0.4 nA. In Figure 2.5, we can observe that the model implemented in the VLSI neuron has a good dynamic compared to the HH model in terms of frequency, voltage range and behavior of the neuron spike. The main difference between both lies in the waveform of the membrane voltage. Even though its dynamic is similar, apart from the action potential, the width of the spike is larger for the simplified HH model.

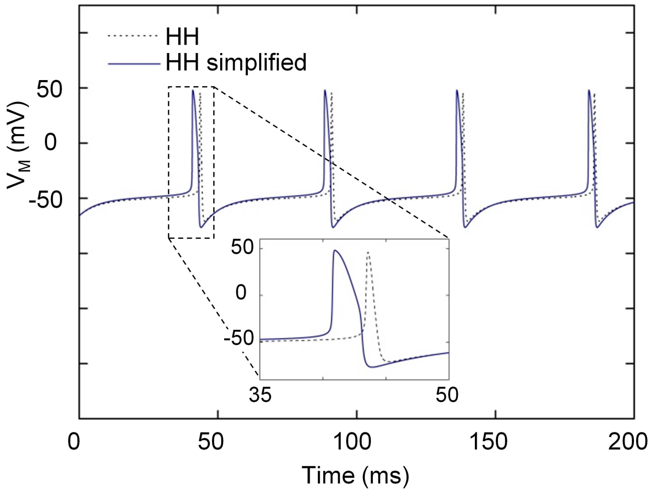


**Figure 2.5** – Membrane voltage software simulations: comparison between the HH model and the simplified HH model implemented in the VLSI neuron.

Thus, how does this approximation changes the complete dynamics of the original HH model? We can use the bifurcation analysis to provide an answer to this question. Bifurcation analysis is useful because it is believed that the computational properties of neurons are based on the bifurcations exhibited by these dynamical systems in response to some changing stimulus as shown by Izhikevich [2000] and by Rinzel and Ermentrout [1989].

These artificial systems would be useful to neuroscientists for exploring neural computation. Hence, by showing that the silicon neuron model has similar bifurcations to a certain class of biological neurons, we can claim that the silicon neuron can also perform similar computations.

## 2.3 Bifurcation analysis

All of the contents for the bifurcation analysis are referenced from Izhikevich [2000], where the bifurcation mechanisms involved in the generation of action potentials by neurons are reviewed. Hodgkin [1948] primarily classified biological neurons into two classes according to their response properties to a sustained current stimulus: the cells show either class-1 or class-2 excitability. We only consider codimension-one bifurcations: Saddle-node bifurcations and Hopf bifurcations, the bifurcations involved in the cells of class-1 and class-2 excitability (described below). The aim of this section is to introduce the notions of bifurcation analysis and then, through this analysis, to compare the complete dynamics of the original HH model and our simplified model.

We know that a neuron is quiescent if its membrane potential is at rest or it exhibits small-amplitude ("subthreshold") oscillations. In dynamical system terminology, this corresponds to the system residing at an equilibrium state[3] or a small amplitude limit cycle attractor[4], respectively.

Consider an autonomous system of ordinary differential equations (ODEs):

$$\dot{x} = f(x, \lambda), \qquad x \in \Re^n, \quad \lambda \in \Re^p \qquad (\textbf{equation } 2.19)$$

where $f$ is smooth. A bifurcation occurs at parameter $\lambda = \lambda_0$ if there are parameter $\lambda_1$ values arbitrarily close to $\lambda_0$ with dynamics that are topologically inequivalent from those at $\lambda_0$. Thus, a bifurcation of a dynamical system is a qualitative change in its dynamics produced by varying parameters. For example, the number or stability of equilibria or periodic orbits of $f$ may change with perturbations of $\lambda$ from $\lambda_0$. Each bifurcation type or singularity is given by a name; for example, an Andronov-Hopf bifurcation. No distinction has been made in the literature between "bifurcation" and "bifurcation type"; both are

[3] An equilibrium (or equilibrium point) of a dynamical system generated by an autonomous system of ordinary differential equations (ODEs) is a solution that does not change with time. More precisely, the ODE:

$$\dot{x} = f(x),$$

has an equilibrium solution $x(t) = x_e$ , if $f(x_e) = 0$. Equilibrum points are sometimes called fixed points or steady states. The stability of typical equilibria of smooth ODEs is determined by the sign of the real part of the eigenvalues (any number such that a given square matrix minus that number times the identity matrix has a zero determinant) of the Jacobian matrix. These eigenvalues are often referred to as the "eigenvalues of the equilibrium". The Jacobian matrix of a system of smooth ODEs is the matrix of the partial derivatives of the right-hand side with respect to state variables where all derivatives are evaluated at the equilibrium point $x = x_e$ . Its eigenvalues determine linear stability properties of the equilibrium. An equilibrium is asymptotically stable if all of the eigenvalues have negative real parts; it is unstable if at least one eigenvalue has a positive real part.

[4] Consider a two-dimensional (planar) system with smooth right-hand side:

$$\begin{cases} \dot{x_1} = f_1(x_1, x_2) \\ \dot{x_2} = f_2(x_1, x_2) \,. \end{cases}$$

A differential equation system has a limit cycle, if for a set of initial conditions, $x_1(t_0) = x_{10}$ and $x_2(t_0) = x_{20}$, the solution functions, $x_1(t)$ and $x_2(t)$, describe an isolated, closed trajectory ("isolated" means that neighboring trajectories are not closed). Like fixed points, limit cycles are attracting or repelling. A stable limit cycle is one which attracts all neighboring trajectories. A system with a stable limit cycle can exhibit self-sustained oscillations; most of the biological processes of interest are of this kind. Neighboring trajectories are repelled from unstable limit cycles.

called "bifurcations". The codimension of a bifurcation is the number of parameters which must be varied for the bifurcation to occur.

### 2.3.1 Codimension-one bifurcations

**Saddle-node bifurcations**

In the mathematical area of bifurcation theory, a saddle-node bifurcation or fold bifurcation is a local bifurcation in which two fixed points (or equilibria) of a dynamical system collide and annihilate each other. In systems generated by autonomous ODEs, this occurs when the critical equilibrium has one zero eigenvalue[5]. The term "saddle-node bifurcation" is most often used in reference to continuous dynamical systems. In discrete dynamical systems, the same bifurcation is often instead called a fold bifurcation.

**Hopf bifurcations**

The Hopf bifurcation theory, as shown by Hassard et al. [1981], asserts that if a parameterized system possesses an equilibrium point and two of the eigenvalues of the Jacobian matrix of the system linearized[6] around the equilibrium point are conjugate pure imaginary numbers and the others have negative real parts, one of the following bifurcations takes place as the parameter changes:

- A bifurcation from a stable equilibrium point to an unstable equilibrium point with a stable limit cycle around it; or a bifurcation with the opposite direction. We refer to this bifurcation as supercritical;
- A bifurcation from an unstable equilibrium point to a stable equilibrium point with an unstable limit cycle around it; or a bifurcation with the opposite direction. We refer to this bifurcation as subcritical.

### 2.3.2 Class 1 and 2 - Neural Excitability

A neuron is said to be excitable if a small perturbation away from a quiescent state can result in a large excursion of its potential before returning to quiescence.

[5] In the n-dimensional case with n $\geq 2$ , the Jacobian matrix at the saddle-node bifurcation has: (1) a simple zero eigenvalue $\lambda_1 = 0$, (2) $n_s$ eigenvalues with Re $\lambda_j < 0$, and (3) $n_u$ eigenvalues with Re $\lambda_j > 0$, with $n_s + n_u + 1 =$ n.

[6] In mathematics and its applications, linearization refers to finding the linear approximation to a function at a given point. In the study of dynamical systems, linearization is a method for assessing the local stability of an equilibrium point of a system of nonlinear differential equations or discrete dynamical systems.

Hodgkin [1948] primarily classified biological neurons into two classes according to their response properties to a sustained current stimulus. The cells that show Class 1 neural excitability can fire with an arbitrarily low frequency by applying a current sufficiently close-to-threshold. The frequency increases monotonically as the current increases. The cells that show Class 2 neural excitability transition from silence to firing at an arbitrary nonzero frequency. The qualitative distinction between Class 1 and Class 2 excitable neurons is that the emerging oscillations have zero frequency in the former and nonzero frequency in the latter. This reflects different underlying bifurcation mechanisms. Furthermore, when the current increases, the rest potential increases until a bifurcation occurs, resulting in a loss of stability or the disappearance of the rest potential, and the neuron activity becomes oscillatory. The bifurcation resulting in transition from a quiescent to an oscillatory state determines the class of neural excitability:

- The characteristic of Class 1 can be associated with a saddle-node bifurcation at the transition from silence to spiking;
- Class 2 excitability is observed when the resting state loses its stability via a Hopf bifurcation.

The HH model belongs to Class 2. Its dynamical properties with a constant stimulus have been studied extensively by Hassard [1978] and Hassard et al. [1981], who have shown that the Class 2 neural excitability of the HH model is generated by Hopf bifurcations.

### 2.3.3 Numerical methods for the bifurcation analysis

One of the principal uses of the bifurcation theory is to analyze the bifurcations that occur in specific families of dynamical systems. Through the use of numerical methods, many problems can be solved that would otherwise be thought to be insolvable. Several software packages (AUTO, CONTENT, MATCONT, XPPAUT, PyDSTool) can implement algorithms that perform bifurcation analyses.

To validate our approximation, we used continuation of solutions in AUTO, an open source mathematical package that can produce bifurcation curves for equilibria as well as for periodic orbits, to get the complete bifurcation diagram of the system. We compare the simulations between the HH-based model (data presented by Hansel et al. [1993]) and our simplified HH model (see Table 2.1).

Figure 2.6 shows the resulting bifurcation diagram of our simplified HH model where solid thick and solid thin curves represent stable and unstable equilibria, respectively, while solid and dashed circles denote stable and unstable limit cycles.

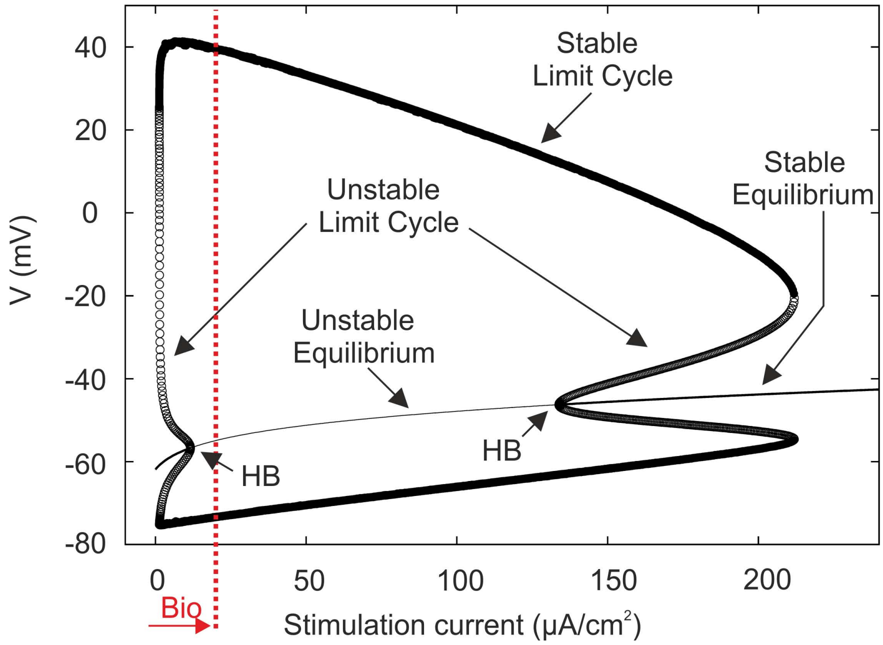


**Figure 2.6** – Bifurcation diagram for our simplified model.

It can be seen in Figure 2.6 that the limit cycles are born initially through a fold bifurcation of limit cycles. Both of the Hopf bifurcations are subcritical as they involve an unstable limit cycle. The system state jumps to the stable limit cycle when the stimulus current exceeds the Hopf bifurcation point. The amplitude of the stable limit cycle continues to reduce until it coalesces with the unstable limit cycle in another fold bifurcation. The resting state loses stability via the subcritical Hopf bifurcation at the first HB point in Figure 2.6. At the second HB point in Figure 2.6, a stable and an unstable limit cycle arise via the fold bifurcation. The bifurcation diagram in Figure 2.6 illustrates how the repetitive firing emerges when we apply a sustained stimulus current. The stable limit cycle corresponds to repetitive firing.

In order to compare this with the HH model, its bifurcation is also shown in Figure 2.7. The Hopf bifurcation for a smaller current is subcritical. The limit cycle appears by a fold bifurcation and disappears by a supercritical Hopf bifurcation. The reduction in amplitude before the subcritical Hopf bifurcation is similar to the one shown in Figure 2.6. It can be seen in Figure 2.7 that the limit cycle arises from a fold bifurcation but terminates in a supercritical Hopf bifurcation.

The qualitative nature of the plots (Figures 2.6 and 2.7) is similar as the reduction in amplitude of the limit cycle before its disappearance is present in both figures.

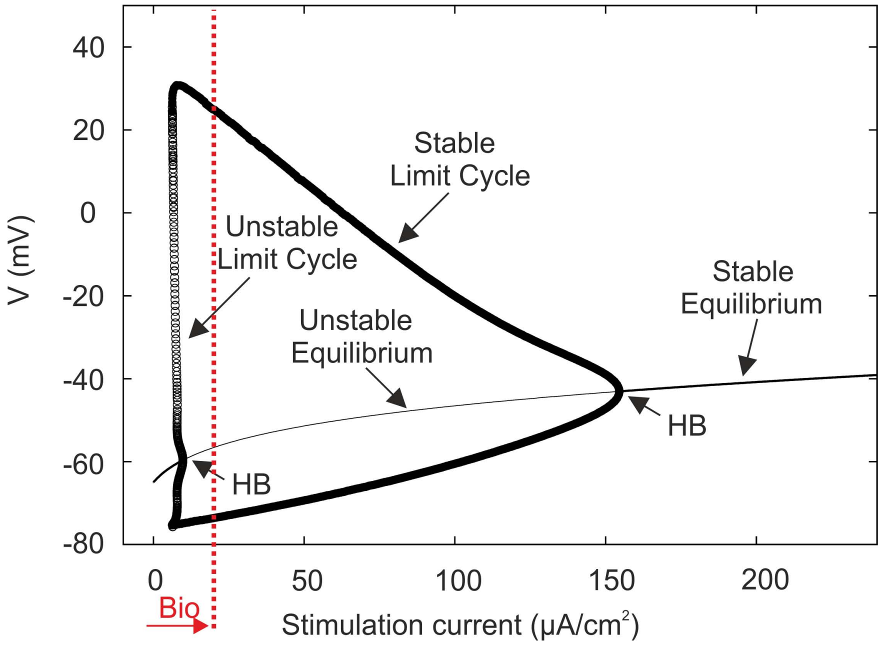


**Figure 2.7** – Bifurcation diagram for the HH model.

We see through the bifurcation analysis [Grassia et al., 2012] that the simplified HH model (implemented in the Galway chip) shares the dynamics of the HH model in a biologically meaningful range of stimulus current[7] extracted from the work presented by Pospischil et al. [2008]. Hence, we can use this approximation while keeping the nonlinear dynamical characteristics of the original model. In the next section, we use minimal Hodgkin-Huxley type models for the different classes of cortical and thalamic neurons presented by Pospischil et al. [2008] to extract the parameters for our VLSI neuron directly from biological data.

## 2.4 Minimal Hodgkin-Huxley type models for cortical neurons

In this section, we describe the intrinsic electrophysiological properties of the four cortical neurons ("fast-spiking", "regular-spiking", "intrinsically bursting" and "low-threshold spike" cells) that we want to emulate in the silicon neuron. For each class, we propose our

[7] The maximum stimulation current for cells that we want to emulate is 20 $\mu$A/$cm^2$.

simplified models taking into account the complete HH type models proposed by Pospischil et al. [2008] who reviewed[8] the development of Hodgkin-Huxley (HH) type models of cerebral cortex and thalamic neurons for network simulations. The models contain the minimal set of voltage-dependent currents to account for the experimental electrophysiological data. The intrinsic electrophysiological properties of cortical neurons were analyzed from several preparations, and these authors selected the four most prominent electrophysiological classes of neurons present in the neocortex and thalamus, inspired from Connors and Gutnick's classification (1990):

1. Fast-spiking (FS),
2. Regular-spiking (RS),
3. Intrinsically bursting (IB),
4. Low-threshold spike (LTS).

The latter class of neurons can also be used to model thalamic neurons, and the RS class is also used to model inhibitory cells with adaptation. This subdivision classifies cells based on three qualitative criteria:

1. the presence or absence of spike-frequency adaptation;
2. the presence or absence of burst discharges from depolarizing stimuli;
3. the presence or absence of burst (or any other type of) discharge following hyperpolarizing inputs (rebound response).

For each cell class, they proposed HH type models that capture the diversity of the intrinsic properties found across different cells and across different preparations. The models contain the minimal set of voltage-dependent currents to account for the data. To obtain models that are as generic as possible, they used data from different preparations in vivo and in vitro[9].

All of the models used here for the cortical neuron models are single-compartment neurons (cylinder of diameter d and length L) described by the following membrane equation:

$$C_{\mathrm{M}} \frac{dV_{\mathrm{M}}}{dt} = -I_{Na} - I_K - I_L - I_M - I_{Ca(T)} - I_{Ca(L)} \,, \qquad \textbf{(equation } 2.20)$$

[8] All of the contents for minimal Hodgkin-Huxley type models for cortical neurons are referenced from Pospischil's work, if not specially mentioned.

[9] In biology, in vivo is often used to refer to experimentation done in live isolated cells rather than in a whole organism; for example, cultured cells derived from biopsies. In this situation, the more specific term is ex vivo. Once cells are disrupted and individual parts are tested or analyzed, this is known as in vitro.

where $V_{\text{M}}$ is the membrane potential, $C_{\text{M}} = 1\ \mu$ F$/cm^2$ is the specific capacitance of the membrane, $I_{Na}$ and $I_K$ are the sodium and potassium currents responsible for action potentials, $I_L$ is the leak current[10], $I_M$ is a slow voltage-dependent potassium current responsible for spike-frequency adaptation, $I_{Ca(L)}$ is a high-threshold calcium current and $I_{Ca(T)}$ is a low-threshold calcium current.

### 2.4.1 Fast-spiking neurons

One of the major cell classes in the cerebral cortex is the "fast-spiking" (FS) neuron, which corresponds to inhibitory neurons. FS cells respond to depolarizing pulses by producing high frequency trains of action potentials without adaptation. FS cells are also the simplest kind of model as only the conductances for generating spikes are needed. This model contained only $I_{Na}$, $I_K$ and $I_L$ with the kinetics described in **equation** 2.21 and **equation** 2.22 for the sodium and potassium current, respectively ($\alpha$ and $\beta$ functions with $V_T$ = -55 mV):

$$\begin{cases} \alpha_m(V_{\text{M}}) = \dfrac{-0.32\cdot(V_{\text{M}}-V_{\text{T}}-13)}{exp[-(V_{\text{M}}-V_{\text{T}}-13)/4]-1} \\ \beta_m(V_{\text{M}}) = \dfrac{0.28\cdot(V_{\text{M}}-V_{\text{T}}-40)}{exp[(V_{\text{M}}-V_{\text{T}}-40)/5]-1} \\ \alpha_h(V_{\text{M}}) = 0.128\cdot exp[-(V_{\text{M}}-V_{\text{T}}-17)/18] \\ \beta_h(V_{\text{M}}) = \dfrac{1}{1+exp[-(V_{\text{M}}-V_{\text{T}}-40)/5]}, \end{cases} \qquad (\textbf{equation}\ 2.21)$$

$$\begin{cases} \alpha_n(V_{\text{M}}) = \dfrac{-0.032\cdot(V_{\text{M}}-V_{\text{T}}-15)}{exp[-(V_{\text{M}}-V_{\text{T}}-15)/5]-1} \\ \beta_n(V_{\text{M}}) = 0.5\cdot exp[-(V_{\text{M}}-V_{\text{T}}-10)/40]. \end{cases} \qquad (\textbf{equation}\ 2.22)$$

The model parameters are: $\bar{g_K} = 10$ mS$/cm^2$, $\bar{g_{Na}} = 50$ mS$/cm^2$, $\bar{g_L} = 0.15$ mS$/cm^2$, $E_K$ = -90 mV, $E_{Na} = 50$ mV, $E_L$ = -70 mV, Area $= 1.4 \cdot 10^{-4} cm^2$, and $C_{\text{M}} = 1\mu F/cm^2$.

Our goal is to reproduce the behavior of biological neurons; therefore, as described in section 2.2, we need the parameters for the simplified model implemented in the VLSI neuron (the VLSI model has similar dynamics to those in the HH model, as shown by the bifurcation analysis). Thus, following the steps explained in section 2.2.2, the only

[10] $I_L$=$g_L$·($V_{\text{M}}$-$E_L$), where $g_L$ is the resting (leak) membrane conductance, and $E_L$ is its reversal potential. These parameters are related to the input resistance $R_L$, which is normally measured experimentally.

difference between the VLSI FS model and the biological FS model is the approximation used for the gating variable. The parameter values that were adapted in order to be implemented in the VLSI neuron are summarized in Table 2.2.

**Table 2.2** – Parameters of the simplified biological FS neuron model.

| | Sodium | Potassium | Leak |
|---|---|---|---|
| $E_{ion}\,(mV)$ | 50 | -90 | -70 |
| $g_{ion}\,(mS/cm^2)$ | 50 | 10 | 0.15 |
| $V_{slope,act}\,(mV)$ | 6.54 | 8.05 | |
| $V_{offset,act}\,(mV)$ | -29.08 | -29.08 | |
| $\tau_{act}\,(ms)$ | 0.065 | 1.066 | |
| $V_{slope,inact}\,(mV)$ | 3.98 | | |
| $V_{offset,inact}\,(mV)$ | -33-31 | | |
| $\tau_{inact}\,(ms)$ | 1.315 | | |

The software simulations of the FS model and the simplified FS model implemented in the VLSI neuron are shown in Figure 2.8.

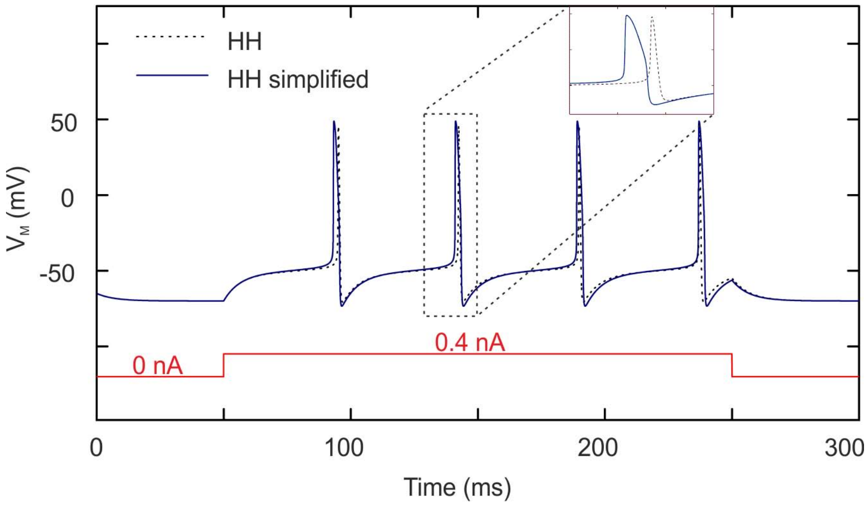


**Figure 2.8** – Membrane voltage software simulations: comparison between the HH model and the simplified HH model implemented in the VLSI neuron; response to a depolarizing current in a fast-spiking neuron model.

In Figure 2.8, we observe that the dynamics of the simplified FS model are comparable to the FS (HH) model in terms of frequency and voltage range. The main difference between the two models lies in the waveform of the membrane voltage. Even though its dynamic is similar, apart from the action potential, the width of the spike is larger for the simplified HH model. This difference appears (see the next sections) for all four cases (FS, RS, IB and LTS) because we use similar values for the sodium and potassium channels, which generate the action potential and its shape.

### 2.4.2 Regular-spiking neurons

Another common cell class in the neocortex is called the Regular-Spiking (RS) neuron, which is, in general, excitatory. The typical responses of the RS cells to the depolarizing current pulses are spike trains with adaptation. The simplest model for RS cells consists of conductances for generating spikes ($I_{Na}$, $I_K$, $I_L$; [Traub and Miles, 1991]) as well as a slow potassium current $I_M$, activated by depolarization [Yamada et al., 1989].

This model contained $I_{Na}$, $I_K$, $I_L$, and $I_M$ simulated by the Hodgkin-Huxley kinetics described in **equation** 2.21, **equation** 2.22 and **equation** 2.23 ($\alpha$ and $\beta$ functions with $V_T$=-55 mV):

$$\begin{cases} p_\infty(V_\mathrm{M}) = \dfrac{1}{1+exp[-(V_\mathrm{M}+35)/10]} \\ \tau_p(V_\mathrm{M}) = \dfrac{\tau_{max}}{3.3\cdot exp[(V_\mathrm{M}+35)/20]+exp[-(V_\mathrm{M}+35)/20]} \end{cases} . \qquad (\textbf{equation } 2.23)$$

The model parameters are: $\bar{g_K}$ = 5 mS/$cm^2$, $\bar{g_{Na}}$ = 50 mS/$cm^2$, $\bar{g_L}$ = 0.1 mS/$cm^2$, $\bar{g_M}$ = 0.07 mS/$cm^2$, $E_K$ = -90 mV, $E_{Na}$ = 50 mV, $E_L$ = -70 mV, Area = 2.9 · $10^{-4}$ $cm^2$, $\tau_{max}$ = 1000 ms and $C_\mathrm{M}$ = $1\mu F/cm^2$.

The parameter values that were adapted (using the former approximation as explained in section 2.2.2) in order to be implemented in the VLSI neuron are summarized in Table 2.3.

**Table 2.3** – Parameters of the simplified biological RS neuron model.

| | Sodium | Potassium | Leak | Slow K |
|---|---|---|---|---|
| $E_{ion}\,(mV)$ | 50 | -90 | -70 | -90 |
| $g_{ion}\,(mS/cm^2)$ | 50 | 5 | 0.1 | 0.07 |
| $V_{slope,act}\,(mV)$ | 6.54 | 8.05 | | 10 |
| $V_{offset,act}\,(mV)$ | -29.08 | -29.08 | | -35 |
| $\tau_{act}\,(ms)$ | 0.065 | 1.066 | | 100 |
| $V_{slope,inact}\,(mV)$ | 3.98 | | | |
| $V_{offset,inact}\,(mV)$ | -33.31 | | | |
| $\tau_{inact}\,(ms)$ | 1.315 | | | |

Figure 2.9 shows the software simulations of the RS model and the simplified RS model implemented in the VLSI neuron. The instantaneous frequencies at the beginning are not the same even though the frequencies are identical after the adaptation.

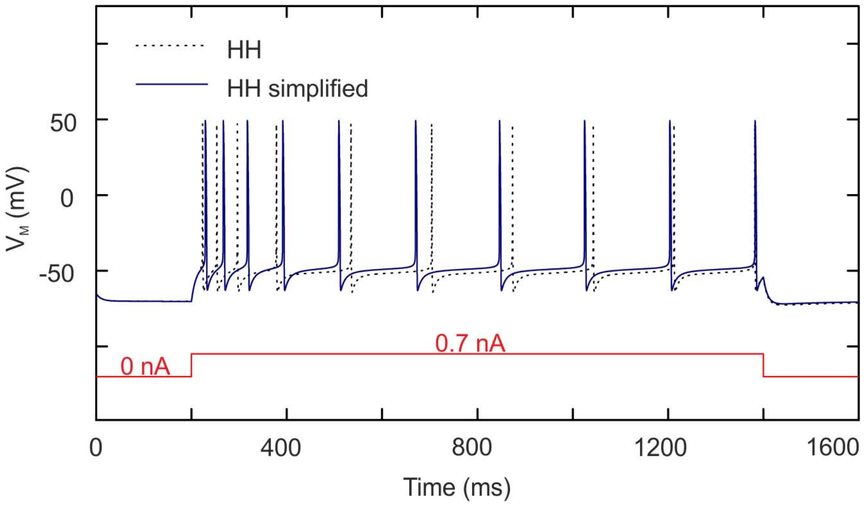


**Figure 2.9** – Membrane voltage software simulations: comparison between the HH model and the simplified HH model implemented in the VLSI neuron; response to a depolarizing current in a model of a regular-spiking neuron.

Moreover, we observe a difference for the adaptation time constant even though there is the same number of spikes. This phenomenon is clearly due to the approximation of the adaptation time constant. We have chosen to work in steady-state conditions, which is justified mainly because cortical neurons in vivo operate in states of intense and sustained firing activity [Destexhe et al., 2003], in which case, the adaptation mechanisms are expected to be at steady-state most of the time. Considering that the most important aspect is to obtain the same frequency after the adaptation period, then this approximation is relevant.

### 2.4.3 Intrinsically Bursting Neurons

Another very common cell class is the Intrinsically Bursting (IB) neuron. The IB cells represent only a few percent of the recorded cells in the primary sensory cortex, both in vivo and in vitro. This kind of neuron generates bursts of action potentials following depolarizing stimuli and then the firing rate suddenly decreases. To generate the bursting behavior, is necessary to extend the previous RS cell model by adding the L-type calcium current. This model contained $I_{Na}$, $I_K$, $I_L$, $I_M$, and $I_{Ca(L)}$ simulated by the Hodgkin-Huxley kinetics described in **equation** 2.21, **equation** 2.22, **equation** 2.23 and **equation** 2.24 ($\alpha$ and $\beta$ functions with $V_T$ = -55 mV):

$$\begin{cases} \alpha_q(V_{\text{M}}) = \dfrac{0.055 \cdot (-27 - V_{\text{M}})}{exp[(-27 - V_{\text{M}})/3.8] - 1} \\ \beta_q(V_{\text{M}}) = 0.94 \cdot exp[(-75 - V_{\text{M}})/17] \\ \alpha_r(V_{\text{M}}) = 0.000457 \cdot exp[(-13 - V_{\text{M}})/50] \\ \beta_r(V_{\text{M}}) = \dfrac{0.0065}{exp[(-15 - V_{\text{M}})/28] + 1} . \end{cases} \qquad \text{(equation 2.24)}$$

The model parameters are: $\bar{g_K}$ = 5 mS/$cm^2$, $\bar{g_{Na}}$ = 50 mS/$cm^2$, $\bar{g_L}$ = 0.01 mS/$cm^2$, $\bar{g_M}$ = 0.03 mS/$cm^2$, $\bar{g_{Ca(L)}}$ = 0.32 mS/$cm^2$, $E_K$ = -90 mV, $E_{Na}$ = 50 mV, $E_L$ = -85 mV, $E_{Ca}$ = 120 mV, Area = $2.9 \cdot 10^{-4}\, cm^2$, $\tau_{max}$ = 1000 ms and $C_{\text{M}} = 1\mu F/cm^2$.

The parameter values that were adapted (using the former approximation as explained in section 2.2.2) in order to be implemented in the VLSI neuron are summarized in Table 2.4.

**Table 2.4** – Parameters of the simplified biological IB neuron model.

| | Sodium | Potassium | Leak | Slow K | Calcium L |
|---|---|---|---|---|---|
| $E_{ion}\,(mV)$ | 50 | -90 | -85 | -90 | 120 |
| $g_{ion}\,(mS/cm^2)$ | 50 | 5 | 0.01 | 0.05 | 0.32 |
| $V_{slope,act}\,(mV)$ | 6.54 | 8.05 | | 10 | 4.20 |
| $V_{offset,act}\,(mV)$ | -29.08 | -29.08 | | -35 | -33 |
| $\tau_{act}\,(ms)$ | 0.065 | 1.066 | | 100 | 1.422 |
| $V_{slope,inact}\,(mV)$ | 3.98 | | | | 22.07 |
| $V_{offset,inact}\,(mV)$ | -33.31 | | | | -57.51 |
| $\tau_{inact}\,(ms)$ | 1.315 | | | | 448.7 |

The software simulations of the IB model and simplified IB model implemented in the VLSI neuron are shown in Figure 2.10. The IB model (solid line in Figure 2.10) reproduces the typical firing characteristics of IB cells as recorded in somatosensory cortex in vitro [Pospischil et al., 2008]. This model generated an initial burst followed by an adapting train of action potentials. In Figure 2.10, the simplified IB model is also shown (dotted line). The same observations as for the RS neuron can be made for the comparison between the IB model and the simplified IB model (Figure 2.10).

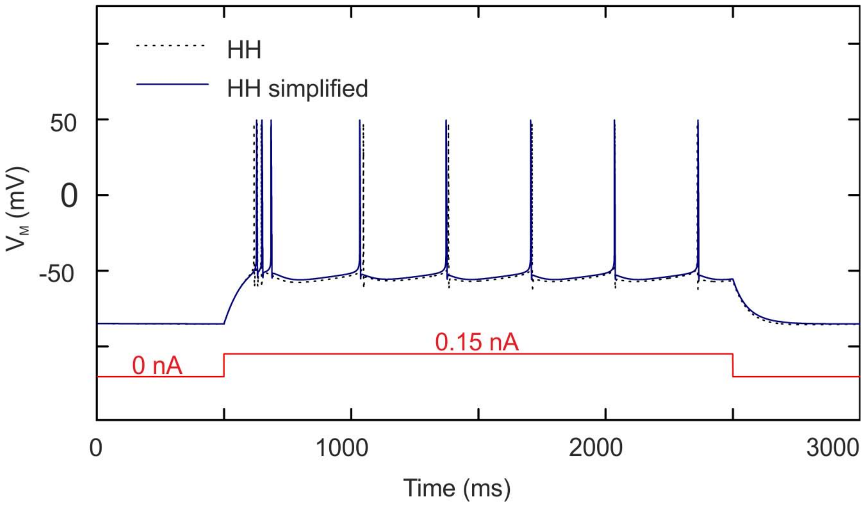


**Figure 2.10** – Membrane voltage software simulations: comparison between the HH model and simplified HH model implemented in the VLSI neuron; response to a depolarizing current (0.15 nA) in a model of an intrinsically bursting neuron.

The initial frequencies are similar, while the final frequencies are identical. For both traces, there is the same number of spikes. As with the RS neuron, our approximation is validated.

### 2.4.4 Low-Threshold Spiking Neurons

In Destexhe [2001], the activities of Low-Threshold Spiking (LTS) neurons are described in a significant fraction (about 10%) of intracellularly recorded cells in cat association cortex in vivo. These LTS neurons generate adapting trains of action potentials in response to a depolarizing current injection, similar to the classic "regular-spiking" response of cortical neurons. In addition, they generate a burst of action potentials in response to an injection of hyperpolarizing current pulses. This property was also identified in deep layers of guinea-pig cerebral cortex in vitro [De la Peña and Geijo-Barrientos, 1996] and was shown to be due to the presence of the T-type (low-threshold) calcium current. In this case, we extended the previous model of the RS cell by adding the T-type calcium current. Like for IB neurons, biological recordings obtained from the LTS neuron are rare. This model contained $I_{Na}$, $I_K$, $I_L$, $I_M$, and $I_{Ca(T)}$ simulated by the Hodgkin-Huxley kinetics described in **equation** 2.21, **equation** 2.22, **equation** 2.23 and **equation** 2.25 ($\alpha$ and $\beta$

functions with $V_T$ = -55 mV):

$$\begin{cases} s_\infty(V_{\mathrm{M}}) = \dfrac{1}{1+exp[-V_{\mathrm{M}}+V_x+57)/6.2]} \\ u_\infty(V_{\mathrm{M}}) = \dfrac{1}{1+exp[(V_{\mathrm{M}}+V_x+81)/4]} \\ \tau_u(V_{\mathrm{M}}) = \dfrac{30.8+(211.4+exp[(V_{\mathrm{M}}+V_x+113.2)/5])}{3.7\cdot(1+exp[(V_{\mathrm{M}}+V_x+84)/3.2])} \;. \end{cases} \qquad \text{(\textbf{equation} 2.25)}$$

The model parameters are: $\bar{g_K}$ = 5 mS/$cm^2$, $\bar{g_{Na}}$ = 50 mS/$cm^2$, $\bar{g_L}$ = 0.01 mS/$cm^2$, $\bar{g_M}$ = 0.03 mS/$cm^2$, $\bar{g_{Ca(T)}}$ = 1.4 mS/$cm^2$, $E_K$ = -90 mV, $E_{Na}$ = 50 mV, $E_L$ = -85 mV, $E_{Ca}$ = 120 mV, Area = $2.9 \cdot 10^{-4}$ $cm^2$, $\tau_{max}$ = 1000 ms and $C_{\mathrm{M}} = 1\mu F/cm^2$.

The parameter values that were adapted (using the former approximation as explained in section 2.2.2) in order to be implemented in the VLSI neuron are summarized in Table 2.5.

**Table 2.5** – Parameters of the simplified biological LTS neuron model.

| | Sodium | Potassium | Leak | Slow K | Calcium L |
|---|---|---|---|---|---|
| $E_{ion}\,(mV)$ | 50 | -90 | -85 | -90 | 120 |
| $g_{ion}\,(mS/cm^2)$ | 50 | 5 | 0.01 | 0.03 | 1.13 |
| $V_{slope,act}\,(mV)$ | 6.54 | 8.05 | | 10 | 6.20 |
| $V_{offset,act}\,(mV)$ | -29.08 | -29.08 | | -35 | -59 |
| $\tau_{act}\,(ms)$ | 0.065 | 1.066 | | 100 | |
| $V_{slope,inact}\,(mV)$ | 3.98 | | | | 4 |
| $V_{offset,inact}\,(mV)$ | -33.31 | | | | -83 |
| $\tau_{inact}\,(ms)$ | 1.315 | | | | 21 |

The software simulations of the LTS model and the simplified LTS model implemented in the VLSI neuron are shown in Figure 2.11. The LTS model (solid line) reproduces the typical firing characteristics of LTS cells as recorded in somatosensory cortex in vitro [Pospischil et al., 2008]. In Figure 2.11, we observe a comparison between the LTS model and the simplified LTS model of neuron response to a hyperpolarizing current. Even though the overall behaviors are similar, we note that the time when the spike occurs is different. We manually tried different values for the calcium current time constant (Table 2.5) and we chose the most acceptable. In the case of the comparison between the membrane voltage dynamics, the approximations of the time constants have an effect on the spike shape and/or on the spike frequency.

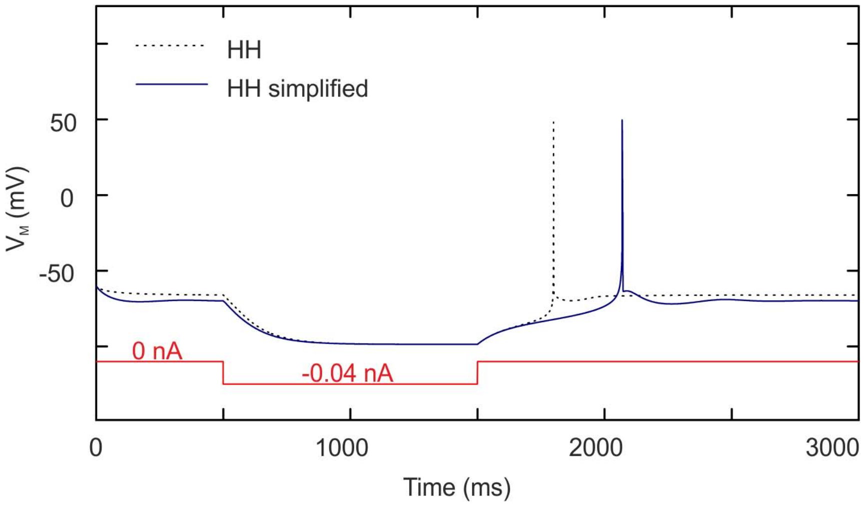


**Figure 2.11** – Membrane voltage software simulations: comparison between the HH model and the simplified HH model implemented in the VLSI neuron; response to a hyperpolarizing current (-0.04 nA) in a model of a low-threshold spiking neuron.

We expected these differences; however, as explained above, we compare the HH model and the simplified model with the software simulations to check if these approximations are suitable. On the other hand, the validation of the neuromimetic IC tuning will be done through the comparison of the IC measurements with the electrophysiological recordings because we want to emulate biological neurons with our neuromimetic chip in order to build a bio-artificial network that operates in real time.

With regards to the validation of the neuromimetic IC tuning, the tuning of conductance-based analog neuromimetic chips has already been investigated by several researchers [Rasche and Douglas, 2000; Shin and Koch, 1999; Simoni et al., 2004; Yu and Cauwenberghs, 2010]. However, none of these authors compared their results with biological data. Simoni et al. [2004] and Yu and Cauwenberghs [2010] validated the tuning primarily using the internal variables of the model, which are not usually recorded in biological cells. Rasche and Douglas [2000] and Shin and Koch [1999] focused on the control of the firing rate versus stimulation. This dependency between the frequency and the input currents is used to study the network dynamic. However, these neuromimetic designs were never compared to biological data. Moreover, the variety of the implemented cell types is limited to the fast-spiking (FS) neuron [Yu and Cauwenberghs, 2010] as well as the regular-spiking (RS) neuron [Rasche and Douglas, 2000; Shin and Koch, 1999]. Only Simoni et al. [2004] presented more complex behaviors.

## 2.5 Conclusion

The main advantage of the analog computation of neural models, compared to their numerical simulation, arises from the intrinsic local analog and parallel nature of computation. As mentioned in the introduction, we want to insert silicon neurons among biological neurons. To reach our goal, we proposed a simplified version of the HH formalism and the appropriate parameter sets for the FS, RS, IB, and LTS neurons that can be implemented in our analog neuromimetic chip. The models considered here are the simplest types of biophysical models where the intrinsic properties arise from voltage-dependent conductances which are described by differential equations (HH type models). The main motivation for this model type is the strong correspondence of their parameters with those in biology. In this chapter, we described the steps used to extract the parameters from the HH model, using biological data, to the one implemented in the VLSI neurons, validating the simplified model through a bifurcation analysis. Hence, we described the intrinsic electrophysiological properties of the four cortical neurons (FS, RS, IB, and LTS neurons) and, for each class of neuron, we proposed our simplified models taking as reference the complete HH type models proposed by Pospischil et al. [2008]. By comparing the software simulations of our simplified neuron model with the HH models for the cortical neuron cells, we show that the simplified models can reproduce the main firing features of cortical cell types. In the next chapter, after the introduction of the hardware platform, we show that the circuits can reproduce the main firing features of cortical cell types by comparing them with experimental electrophysiological data for these cells.

# Chapter 3

## Real-time cortical neuron model simulation

In a previous work focusing on the implementation of Hodgkin-Huxley type models in analog circuits, our group designed several neuromimetic chips (ASICs), including the Galway chip [Bornat, 2006], which exploit the intrinsic voltage-current of individual transistors (bipolar and MOSFET) to simulate the membrane equation of neurons. These silicon neurons are based on the Hodgkin-Huxley formalism and they are optimized to reproduce a large variety of neuron behaviors using tunable parameters. In Chapter 2, we described the steps used to extract the parameters from the HH model, using biological data, to the model implemented in the VLSI neurons. As already mentioned, we want to insert silicon neurons among biological neurons. These ASICs could be used to form the core of various simulation platforms to emulate neural networks in biologically relevant configurations. In particular, the aim is to simulate cortical networks using data and electrophysiological recordings provided by neuroscientists. Hence, we decided to qualitatively compare the dynamics of our silicon circuit to biological cells with a high level of detail (time scale, voltage range, resting potential, etc.). To fulfill our requirement, we selected the four most prominent cortical neurons (the FS, RS, IB, and LTS neurons) and then we propose a simplified version of the HH formalism and the appropriate parameter sets for the FS, RS, IB, and LTS neurons that can be implemented in our analog neuromimetic chip. Due to the fabrication process, there are significant differences between the expected and actual outputs of the chips [Saïghi et al., 2008]. It is therefore necessary to further adjust the parameters in order for the outputs of the neuromimetic circuits to be approximate with the neuronal activity. Therefore, in this chapter, we present the hardware platform and the proposed full-custom fitting method in voltage-clamp mode to tune our neuromimetic integrated circuits [Buhry et al., 2011]. Finally, we show the experimental measurements of our system which mimic the four most prominent biological cells: fast-spiking, regular-spiking, intrinsically bursting, and low-threshold spiking neurons into an analog neuromimetic integrated circuit dedicated to cortical neuron simulations. By comparing them with experimental electrophysiological data for these cells, we show that the circuits can reproduce the main firing features of cortical cell types.

## 3.1 Neuromimetic chip and dedicated board

Our goal is to build a neural simulator based on a hardware implementation that is able to reproduce the dynamics of cortical neurons. Our system is composed of our most recent chip, the Galway chip, and the dedicated board Ekerö. This chip includes analog operators to compute the HH formalism and to construct neural networks [Saïghi et al., 2011].

### 3.1.1 Neuromimetic chip

In a previous work, our library of tunable mathematical operators was validated with a preliminary prototype, the Violetta IC, followed by a second IC prototype referred to as Pamina [Saïghi, 2004], which was used to build a real-time simulator for computational neuroscience applications. From the HH formalism, it was decided to integrate a set of generic blocks, each able to compute a conductance-based model of ionic or synaptic current. The last ASIC prototype was designed in full-custom mode with a BiCMOS SiGe 0.35 $\mu$m technology process from austriamicrosystems (AMS) in the Cadence environment.

The analog computational core, built with various analog operators, represents a set of ionic current generators. Digital functions are added to manage the core topology, and analog memory cells are included in order to store the model parameters. The "Galway" chip presented here comprises:

- five neurons;
- an analog memory cell array, to store the conductance parameters;
- a matrix of switches, to control the topology of the neurons (i.e. the arrangement of the conductance and synaptic modules that form the artificial neuron);
- digital functions, to control data transfer to and from external devices;
- a multi-synapse block that gathers all of the synaptic inputs into one electronic input.

A microphotograph of the Galway chip is shown in Figure 3.1.

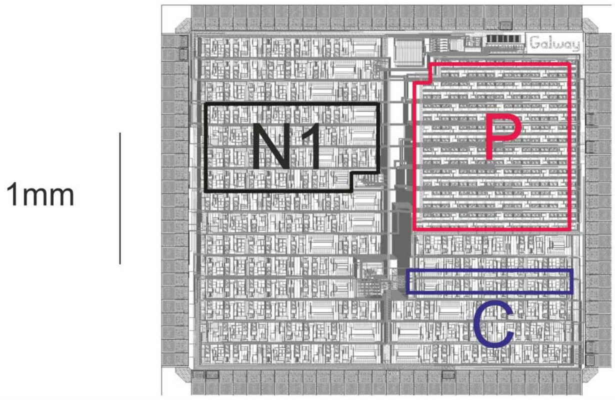


**Figure 3.1** – Microphotograph of the Galway chip (containing five neurons), where P is the analog memory cell array, C is the conductance, and N1 is neuron 1 plus a multi-synapse block.

Each neural element includes (Figure 3.2):

- a set of conductance modules, each able to generate ionic or synaptic currents following the previously presented conductance-based model;
- a spike detection module, to convert the neuron membrane voltage to a 1-bit code;
- a set of synaptic input modules: one for inhibitory input, one for excitatory input, and a third and final one for the background noise activity. These modules activate synaptic conductance modules with a digitally-controlled weight.

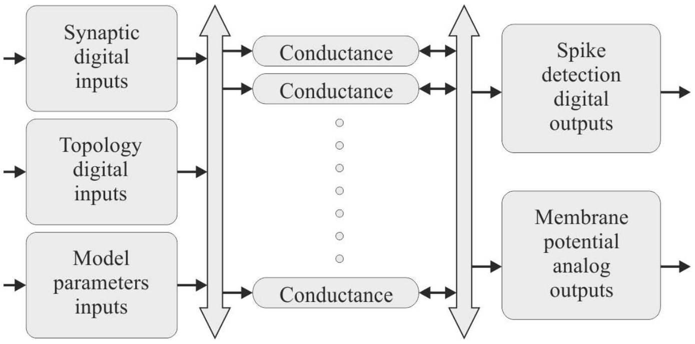


**Figure 3.2** – Block diagram of a neuron included in the Galway IC.

Due to the biological ratio of the FS, RS, IB, and LTS neurons, the five neural elements consist of:

- one FS cell including three conductances $I_{Na}$, $I_K$, and $I_L$;
- three RS cells with four conductances $I_{Na}$, $I_K$, $I_M$, and $I_L$;

- one IB or LTS cell with a fifth conductance $I_{Ca}$.

The model parameters (205 parameters must be set in order to tune all of the neurons) are stored in analog memory cells (except for the time constants which require external capacitors due to their value that ranges from 3.3 nF to 470 nF), the values of which are programmed during the configuration phase of the simulation. The analog memory cells are refreshed every 2 ms (the clock frequency is 100 kHz). During that same phase, the user will also set the topology of the neuron, i.e. define the block connectivity. A set of connected blocks will form an artificial neuron, with their respective currents summed on an external capacitance.

Taking into account the integration constraints of the microelectronic design, and in order to increase its dynamic range and noise immunity, we applied a x5 gain factor to the biological voltages:

$$V_{\text{VLSI}} = 5 \cdot V_{\text{Bio}} \, . \qquad \textbf{(equation 3.1)}$$

Let $C_{\text{VLSI}}$ and $C_{\text{Bio}}$ represent the membrane capacitances of artificial and biological neurons, respectively. The conductance mapping is proportional to the capacitance ratio, where:

$$\frac{g_{\text{VLSI}}}{g_{\text{Bio}}} = \frac{C_{\text{VLSI}}}{C_{\text{Bio}}} \, . \qquad \textbf{(equation 3.2)}$$

The current mapping then equals the product of the voltage and conductance mappings:

$$I_{\text{VLSI}} = I_{\text{Bio}} \cdot 5 \frac{C_{\text{VLSI}}}{C_{\text{Bio}}} \, . \qquad \textbf{(equation 3.3)}$$

With the Galway chip, $C_{\text{VLSI}} = 3.3$ nF and the biological neurons have:

$$C_{\text{Bio}} = C_{\text{M}} \cdot Area \, , \qquad \textbf{(equation 3.4)}$$

where $C_{\text{M}}$ is the specific membrane capacitance and Area represents the area of the cell membrane.

### 3.1.2 Dedicated board

The neuromimetic chip is embedded in a six-layer full-custom board called Ekerö (Figure 3.3).

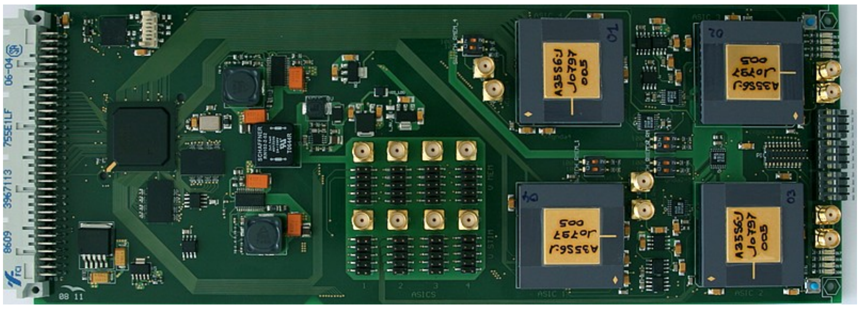


**Figure 3.3** – Photograph of the Ekerö board.

This board hosts four Galway ASICs. Each ASIC incorporates five neurons which compute in analog mode conductance-based models following the Hodgkin-Huxley formalism. Individual neurons produce, in continuous time, action potentials that express their intrinsic dynamic properties as well as their response to stimulations. The neuron type, firing rate and stimulus response magnitude can be configured in each neuron.

Synaptic digital inputs, topology digital inputs, model parameters inputs and spike detection outputs of the Galway chip are individually connected to a Xilinx Spartan3 FPGA. The FPGA is connected to a host computer with the RS232 serial protocol which allows the user to send and receive data to/from the chips. Some of the coaxial connectors are used to provide analog outputs for the observation of ionic currents and membrane voltages on an oscilloscope. The others are used for analog inputs to stimulate the silicon neuron or to impose a membrane voltage. Several Ekerö boards have already been designed for the next neural network experiments [Belhadj, 2010].

### 3.1.3 Hardware and software interfaces to the Galway chip

We describe here the interfaces (hardware and software) that allow the use of the Galway chip. We implemented the VHDL driver in the FPGA (Xilinx Spartan3 FPGA in the Ekerö board) to send the parameters to the chip. The ASIC configuration parameters consist of Hodgkin-Huxley parameter values (analog parameters) and the specifications for the neuron type and synapse behavior (digital parameters).

Only the method used to introduce the digital/analog parameters is presented. The correspondence between the parameters and the model is described in Appendix 4.5. We implemented the software graphical interface in the host PC, allowing the user to completely configure the circuit.

### Digital parameters

In order to set the topology of the silicon neuron (FS, RS, IB, and LTS), the input interface of the digital parameters is made by 14-bit words[1]. Transmission occurs via a series of three parallel signal interfaces: clock, data and validation. Validation occurs as the last bit of the word is being transmitted. A diagram of the signals is given in Figure 3.4.

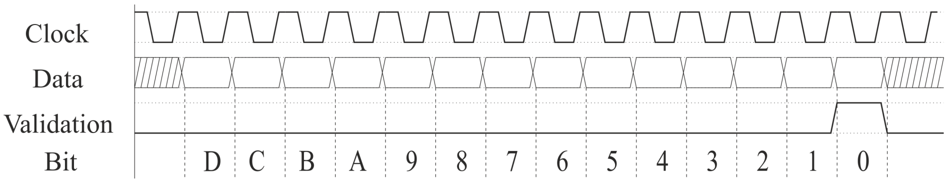


**Figure 3.4** – Timing diagram of the digital parameter input interface (the clock frequency is 100 kHz).

Signal sending is only performed during the configuration step. We designed the driver in VHDL with an operating frequency of 100 kHz. The format of the digital word implemented in VHDL is "d:n:xxxx", in which "d" indicates a digital word, "n" is the number of ASICs (from 1 to 4) and "xxxx" is the word in hexadecimal form (see Appendix 4.5).

### Analog parameters

To send the values of the Hodgkin-Huxley equation, the input interface of the analog parameters requires four signals: clock, refresh initialization (start), validation and data. The various signals are shown in the timing diagram in Figure 3.5.

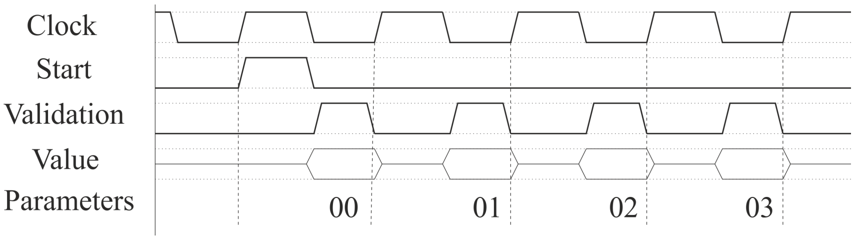


**Figure 3.5** – Timing diagram of the analog parameter input interface (the clock frequency is 100 kHz).

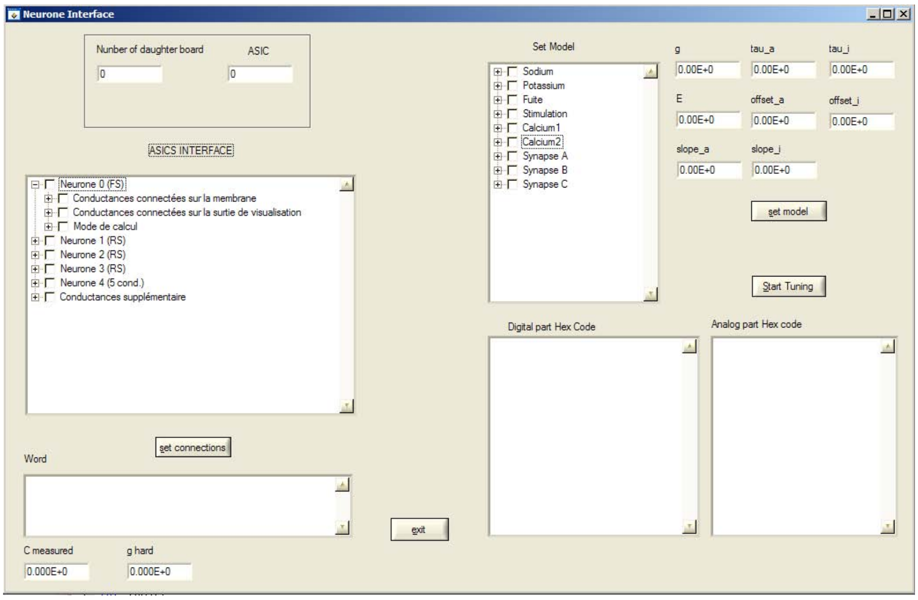
We designed the driver in VHDL taking into account that a transmission up to 205 parameters can be made with an operating frequency of 100 kHz. The format of the analog

[1] The specifications of the word in order to build the driver for the ASICs are taken from Bornat's thesis (2006), the designer of the Galway chip.

word implemented in VHDL is "a:n:xx:x.xxx", in which "a" indicates an analog word, "n" is the number of ASICs (from 1 to 4), "xx" is the type of the parameter in hexadecimal form and "x.xxx" is the value (for example 2.345 volts[2]) of the voltage parameter (see Appendix 4.5).

### Software interface

Figure 3.6 shows a picture of the software graphical interface dedicated to the Galway chip.



Neurone Interface

Nunber of daughter board: 0
ASIC: 0

ASICS INTERFACE

- ☐ Neurone 0 (FS)
  - ☐ Conductances connectées sur la membrane
  - ☐ Conductances connectées sur la surtie de visualisation
  - ☐ Mode de calcul
- ☐ Neurone 1 (RS)
- ☐ Neurone 2 (RS)
- ☐ Neurone 3 (RS)
- ☐ Neurone 4 (5 cond.)
- ☐ Conductances supplémentaire

Set Model

- ☐ Sodium
- ☐ Potassium
- ☐ Fuite
- ☐ Stimulation
- ☐ Calcium1
- ☐ Calcium2
- ☐ Synapse A
- ☐ Synapse B
- ☐ Synapse C

g: 0.00E+0
tau_a: 0.00E+0
tau_i: 0.00E+0
E: 0.00E+0
offset_a: 0.00E+0
offset_i: 0.00E+0
slope_a: 0.00E+0
slope_i: 0.00E+0

C measured: 0.000E+0
g hard: 0.000E+0

**Figure 3.6** – Software graphical interface of the Galway chip.

The FPGA in the Ekerö board is connected to a host computer with a serial protocol (RS232) that allows the user to send and receive data to/from the chips. The user chooses the model set for the ionic current and then the program successively transmits the population of parameters to the VHDL driver to emulate cortical neurons in real-time on the Galway chip. The software interface was designed in a CVI[3] integrated development environment.

[2] Note that all of the biological values of the model are converted into hardware values following the hardware conversion rules explained in section 3.1.1.

[3] LabWindows/CVI is a proven ANSI C integrated development environment that provides engineers and scientists with a comprehensive set of programming tools for creating test and control applications. LabWindows/CVI combines the longevity and reusability of ANSI C with engineering-specific functionality designed for instrument control, data acquisition, analysis, and user interface development.

## 3.2 Automated tuning system for the neuromimetic chip using metaheuristic algorithms

Our analog IC was optimized to reproduce a large variety of neuron behaviors using tunable parameters. However, due to the noise and the analog dispersion in the fabrication process, the IC does not guarantee that the parameters inside the chip are the same as those extracted from a biological cell (see Tables 2.2, 2.3, 2.4, 2.5). Because of the remaining uncertainty with regards to the stored analog parameters (fabrication and process mismatch), we need to estimate the true parameters computed by the circuit.

Considering, for example, the real-time simulation of the FS cell, 15 parameters have to be estimated (see Table 2.2). Therefore, we define three vectors containing the unknown parameters:

1. $X_{Na} = [\, g_{Na} \;\; E_{Na} \;\; \tau_m \;\; V_{slope,m} \;\; V_{offset,m} \;\; \tau_h \;\; V_{slope,h} \;\; V_{offset,h} \,]$;
2. $X_K = [\, g_K \;\; E_K \;\; \tau_n \;\; V_{slope,n} \;\; V_{offset,n} \,]$;
3. $X_L = [\, g_L \;\; E_L \,]$,

where $X_K$, $X_{Na}$ or $X_L$ denote the $X$ vector used in the next section, in the case of the FS cell. Similarly, for the other neurons:

- RS cell - $X_K$, $X_{Na}$, $X_L$ and $X_{slow-K}$;
- IB cell - $X_K$, $X_{Na}$, $X_L$, $X_{slow-K}$ and $X_{Ca(L)}$;
- LTS cell - $X_K$, $X_{Na}$, $X_L$, $X_{slow-K}$ and $X_{Ca(T)}$,

where the number of parameters to be estimated increases.

The tuning of this model, in order to reproduce a given neuronal signal (FS, RS, IB and LTS neurons), is thus difficult. As we mentioned in section 1.3, it is therefore necessary to further adjust the parameters in order for the outputs of the neuromimetic circuits to be approximate with the neuronal activity. As hand tuning is very time-consuming, an automated tuning of the parameters is mandatory. Therefore, in this section, we introduce the metaheuristic algorithms followed by the steps taken to tune the silicon neuron.

### 3.2.1 Metaheuristic algorithms

In computer science, the term metaheuristic designates a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regards to a given measure of quality. Metaheuristics are inspired by natural systems, such as metallurgy when dealing with simulated annealing (SA), evolutionary biology for genetic algorithms (GA), or ethology with ant colony algorithms or particle swarm optimization, which are not presented here.

Two metaheuristics, genetic algorithms and simulated annealing, have already been proposed to adjust the parameters of neuron models [Geit et al., 2008], but it was shown in a comparison [Buhry et al., 2008] that the differential evolution algorithm (DE) was more efficient for parameter estimation in Hodgkin-Huxley type models.

Like the genetic algorithm, the DE belongs to the class of evolutionary algorithms[4]. It uses mechanisms inspired by biological evolution: reproduction, mutation, recombination, and selection [Storn and Price, 1997]. Candidate solutions to the optimization problem play the role of individuals in a population. The DE generates a population of vectors which represents whole populations. The parameters contained in a vector $X$ are also called genes that are, in our case: $g_{ion}$, $E_{ion}$, $\tau$, $V_{slope}$, and $V_{offset}$. This population is randomly initialized with a uniform law within the boundary constraints of the model. Then, a new trial individual is built by means of three operations: Differentiation, Recombination, and Selection [Feoktistov and Janaqi, 2004].

We published a new estimation method for the characterization of the Hodgkin-Huxley formalism based on the DE algorithm [Buhry et al., 2011]. This method is an alternative technique to the estimation methods associated with voltage clamp measurements. It uses voltage clamp type recordings, but is based on the differential evolution algorithm.

As our circuits allow us to make voltage clamp-like measurements, we can use our estimation method, using current recordings, to estimate the model parameters. The parameters of one ionic channel are simultaneously estimated, and the technique is successively applied to each ionic channel to estimate the true parameters computed by the neuromimetic analog integrated circuits.

### Differential evolution algorithm

The DE is a population algorithm, similar to genetic algorithms, but differs from the latter in the recombination/mutation operation (described below), and the selection step. With regards to the mutation process in genetic algorithms, this results from small genetic alterations, whereas in the DE algorithm, it consists of a geometrical combination of vectors. With regards to the selection process in the DE algorithm, at every iteration,

[4] In evolutionary algorithms, the term population means a set of parameter vectors, the term individual denotes a parameter vector, and the term gene denotes one parameter.

each individual is compared with only one other individual; on the contrary, in genetic algorithms, one individual is often compared with the rest of the population.

In the following, we use the same notation as that proposed by Storn and Price [1997]. The DE generates a population of $NP$ individuals $X{=}X_{ion}$ containing $D$ parameters, also called "genes". The population is initialized by randomly choosing individuals within the boundary constraints of the model. Then, at each time step, new trial individuals are constructed by means of two operations: so–called "Differentiation" and "Recombination". We define $X_k^r(i)$ as the gene $i$ of the $r^{th}$ individual of the $k^{th}$ generation (or iteration).

*Differentiation:* the new parameter vector $X_{trial}$ is generated by adding the weighted difference between two other population members, $X_k^{r2}$ and $X_k^{r3}$, to a uniformly randomly-chosen population member, $X_k^{r1}$, with $r_1 \neq r_2 \neq r_3$:

$$\forall j = 1,..,NP, \quad X_{trial}^j = X_k^{r_1} + F \cdot (X_k^{r_2} - X_k^{r_3}), \tag*{(\textbf{equation} 3.5)}$$

where $F$ (factor of differentiation) is usually set to 0.5[5].

*Recombination:* the mutant individual, $X_{mut}$, inherits genes from $X_{trial}$ with a probability $CR$ (recombination constant), where $CR \in [0,1]$ is usually empirically set to 0.8 or 0.9 [Karaboga and S.Okdem, 2004]. By generating $u$ according to a uniform distribution $\mathscr{U}(0,1)$, one has:

$$\forall j = 1,..,NP, \forall i = 1,...,D, \quad X_{mut}^j(i) = \begin{cases} X_{trial}^j(i) & \text{if } u < CR \\ X_k^j(i) & \text{otherwise} \end{cases} . \tag*{(\textbf{equation} 3.6)}$$

Finally, a selection is carried out by comparing the fitness function values of $X_{mut}$ and $X_k^j$ respectively, as follows:

$$X_{k+1}^j = \begin{cases} X_{mut}^j & \text{if } F_{fit}(X_{mut}) \leq F_{fit}(X_k^j) \\ X_k^j & \text{otherwise} \end{cases} \tag*{(\textbf{equation} 3.7)}$$

where $F_{fit}$ is the fitness function[6] (not to be confused with the factor of differentiation, $F$). As for the terminal conditions, either the number of iterations, $N_{iter}$, or an upper

[5] It is recommended to choose F from the interval [0, 2] [Karaboga and S.Okdem, 2004].

[6] A fitness function is a particular type of objective function that is used to summarize, how close a given solution is to achieving the set aims.

bound for the fitness function can be set.

### 3.2.2 Automated tuning system platform

The DE algorithm requires data recorded in the voltage-clamp mode, which involves the observation of one ionic current, when the membrane voltage levels are successively applied with different steps. We use the host computer to control an oscilloscope and an arbitrary waveform generator. We add the "tuning" option to the software interface described in section 3.1.3 (Figure 3.6). The complete program has to drive the instruments and the neuromorphic system, and also computes the DE algorithm (3.7).

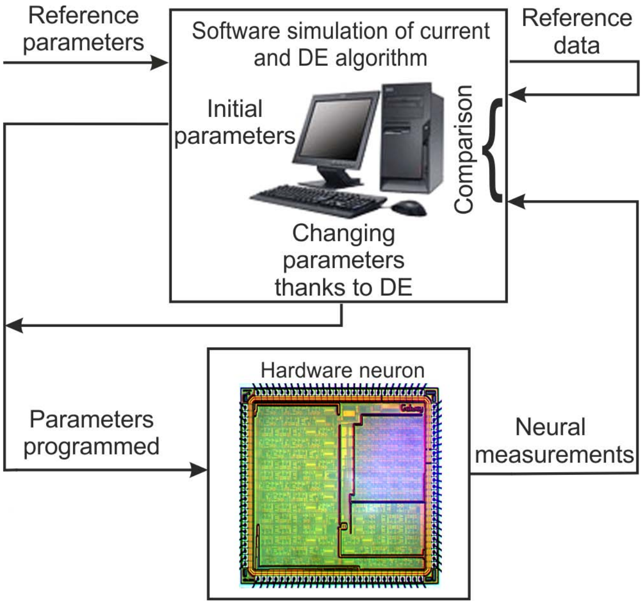


**Figure 3.7** – Schematic diagram of the automated tuning system.

Figure 3.7 shows a schematic diagram of the experimental implementation described in the following steps:

- The user chooses the model set for the ionic current;
- The program successively transmits the population of parameters and generates the different steps of the imposed voltage. Each ionic channel is optimized separately;
- The computer stores the imposed voltages and the current responses measured on the Ekerö board;
- The program calculates the theoretical current response, using the model parameters and the measured imposed voltages. Then it calculates the fitness or cost function

to be minimized, which is defined by the quadratic error between the theoretical current response of an ionic channel and the current response of the same ionic channel measured from the IC while the membrane voltage is clamped.

- The program uses the DE algorithm (Differentiation, Recombination, and Selection) to choose the new population of parameters, before sending it back to the chip.

However the chip tuning technique is not applied to the stimulation current generator ($I_{stim(hw)}$). Due to the process and component mismatch, there is an error current between the expected and actual current.
Therefore, a conversion rule on the $I_{stim(hw)}$ hardware stimulation current is applied to compensate for that error current. $I_{stim(hw)}$ obeys the rule:

$$I_{stim(hw)} = a_{comp} \cdot I_{VLSI} + b_{comp} = 5 \cdot a_{comp} \cdot I_{bio} \cdot \frac{C_{VLSI}}{C_{bio}} + b_{comp}. \qquad \textbf{(equation } 3.8)$$

We empirically defined the $a_{comp}$ and $b_{comp}$ parameters to match the biological data in the real-time simulation for each neuron. Finally, the hardware neuron model is composed of the optimized parameters from the theoretical values, $C_{VLSI}$ and the pair $a_{comp}$ and $b_{comp}$ from **equation** 3.8.

Therefore, we applied this technique to our integrated circuit in order to reproduce a biologically realistic behavior for the ionic current[7] followed by the voltage membrane in a silicon neuron. With this purpose in mind, we compare the membrane voltage of the artificial neuron with the biological neuron presented in [Pospischil et al., 2008].

## 3.3 Emulation of neocortex neurons in the VLSI hardware

Pursuant to our goal of implementing the "prototypical" types of neurons present in the neo-cortex in the VLSI hardware, we successively consider the four different cell classes (FS, RS, IB, and LTS) and show the results of our VLSI chip behavior after tuning [Grassia et al., 2011].

To compare the behaviors between the biological and the hardware neurons, we tuned the parameters shown in Tables 2.2, 2.3, 2.4 and 2.5 (see Chapter 2), in which all electronic values are converted to a biological scale as explained in section 3.1.1, and all of the biological data are reproduced from Pospischil et al. [2008].

[7] More details are given in [Buhry et al., 2011]; here, we only focus on the voltage membrane.

### 3.3.1 Fast-Spiking Neurons

Fast-spiking neurons respond to the injection of a depolarizing current pulse by producing high frequency trains of action potentials without adaptation as shown in Figure 3.8A.

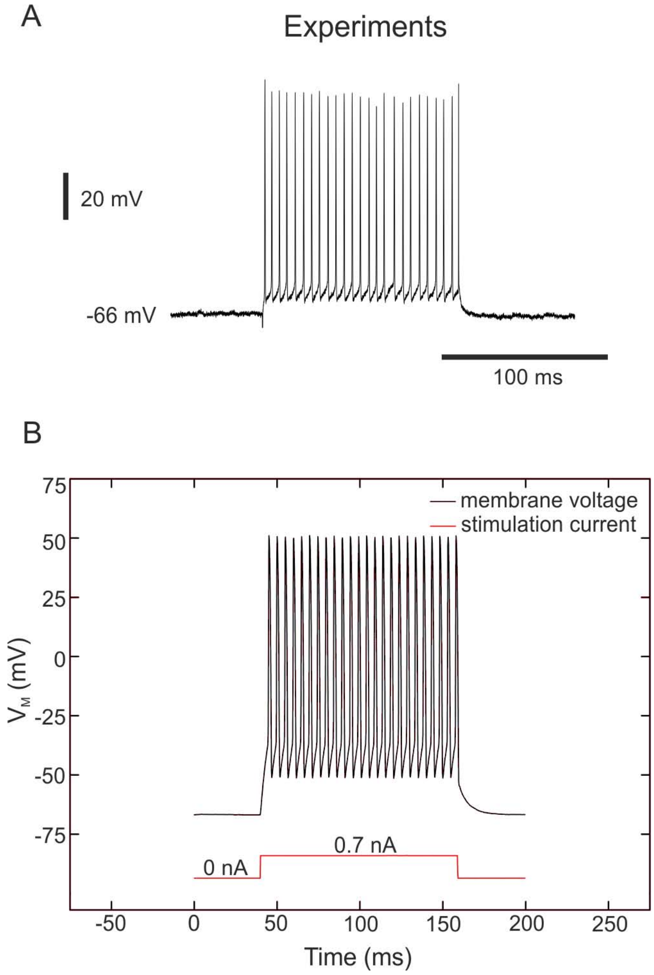


**Figure 3.8** – Membrane voltage of FS neurons. (A) Response of a FS neuron based on ferret visual cortex in vitro (Pospischil et al. [2008]; experimental data from Thierry Bal, CNRS) to the injection of a depolarizing current pulse (0.7 nA). (B) Measurements of the FS hardware neuron at a depolarizing current pulse (0.7 nA). The VLSI voltage measurements are divided by 5 in the figure in accordance with **equation** 3.1.

The FS neurons correspond to inhibitory neurons [Connors and Gutnick, 1990; Gibson et al., 1999]. Figure 3.8B shows the hardware FS neuron response with the application of a stimulation current. In both cases, the stimulation current of 0.7 nA is applied for 125 ms. We observe an identical resting potential of approximately -66 mV and a similar voltage range for the membrane voltage.

However, the comparison of the electrical behavior of the artificial FS neuron with the biological target must go further. We also apply different current pulses for plotting the frequency-current relations (firing rates) of the FS as shown in Figure 3.9. The FS neuron shows no sign of adaptation, and thus its frequency is constant during the stimulation. In Figure 3.9A, we plot the average of the firing rates for the 40 FS neurons after the tuning step as well as the variability of the data using error bars. So far, the stimulation current follows the rule presented in **equation** 3.3. We can observe the effect of the electronic leakage current on the dispersion curves. In Figure 3.9B, we apply **equation** 3.8. The plotted hardware data are the average and the standard deviation (SD) for the 40 neurons.

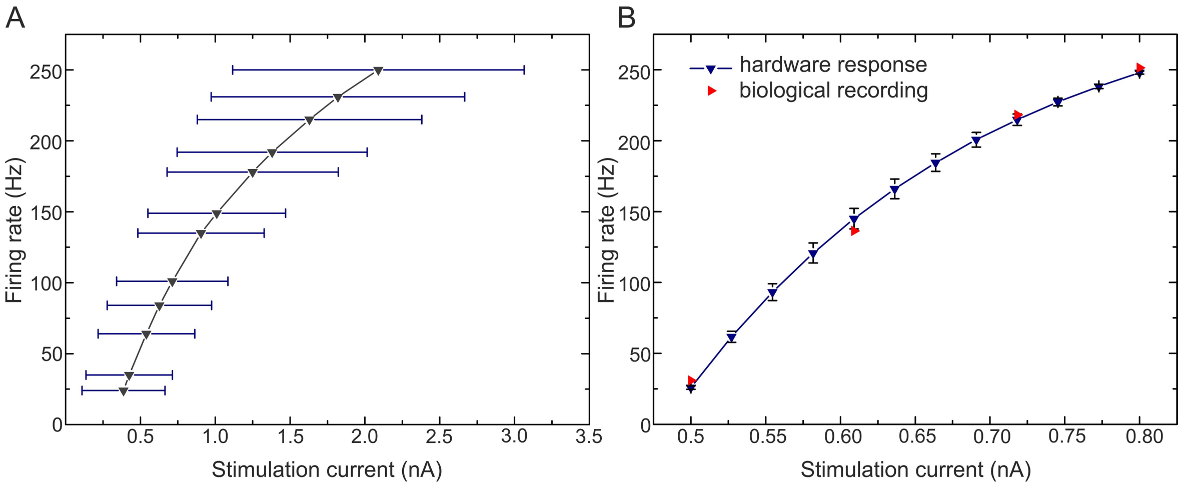


**Figure 3.9** – Frequency versus stimulation current curves of FS neurons. (A) 40 VLSI FS neurons tuned with the model parameters from Table 1 and the automated tuning technique. The stimulation current follows the **equation** 3.3. (B) Biological measurements of the FS neuron from Pospischil et al. [2008] and the 40 VLSI neurons using **equation** 3.8 for the stimulation current.

For biological reference, we take the mean frequency for each stimulation current from Figure 3B from Pospischil et al. [2008]. The biological data are only composed of four points. For each of these points, the artificial neuron frequency matches that of the biological neuron.

Table 3.1 provides the parameter values for the 40 FS neurons. We do not present the results for the time constants because the implemented values depend on external capacitors for which we did not measure the exact value. The theoretical values are provided by the model. The implemented values are computed by the optimization algorithm. In any event, we observed a large discrepancy for all of the parameters, confirming the necessity of the tuning step. The results for the RS are similar because the sodium, potassium, and leakage currents are the same, except for three parameters, and the four new parameters for the slow potassium channel present the same characteristics. Due to the scarcity of data for the LTS and IB neurons, the statistical results are not meaningful. That is why, in the next section, we will not present the same table for the RS, IB, and LTS neurons.

**Table 3.1** – Theoretical and implemented parameter values for the 40 FS neurons.

| | Theor. values | Impl. values (average) | Impl. values (SD) |
|---|---|---|---|
| $E_{Na}\,(mV)$ | 50 | 65.07 | 16.50 |
| $g_{Na}\,(mS/cm^2)$ | 50 | 34.32 | 5.17 |
| $V_{slope,m}\,(mV)$ | 6.54 | 7.56 | 2.26 |
| $V_{offset,m}\,(mV)$ | -29.08 | -33.88 | 8.59 |
| $V_{slope,h}\,(mV)$ | 3.98 | 2.99 | 1.54 |
| $V_{offset,h}\,(mV)$ | -33.31 | -38.59 | 10.92 |
| $E_K\,(mV)$ | -90 | -108.47 | 22.46 |
| $g_K\,(mS/cm^2)$ | 10 | 6.61 | 3.51 |
| $V_{slope,n}\,(mV)$ | 8.05 | 6.75 | 2.28 |
| $V_{offset,n}\,(mV)$ | -29.08 | -38.08 | 16.03 |
| $a_{comp}$ | | 0.203 | 0.050 |
| $b_{comp}$ | | 0.5 | $2.7{\cdot}10^{-11}$ |

Based on the comparison of the membrane voltage and the frequency versus the stimulation current between the biological neuron and the analog hardware neuron, we conclude that the simplification of the model and its implementation in silicon are well-suited to reproduce the behavior of the FS neuron.

### 3.3.2 Regular-Spiking Neurons

Regular-spiking neurons respond to the injection of a depolarizing current pulse by producing trains of spikes with adaptation, as shown in Figure 3.10A, for a typical RS cell from ferret visual cortex in vitro. RS neurons correspond to excitatory neurons.

After tuning our chip, we then apply a stimulation current of 0.7 nA for 200 ms, as in the biological experiment, to compare the electronic and biological behaviors (Figure 3.10B).

Like for the FS neuron, we observe that the resting potential is about -75 mV and the membrane voltage range is similar. Moreover, in both cases we observe a high frequency discharge on the first part of the response and then the frequency decreases slowly due to the adaptation phenomenon. The main difference between the two traces in Figure 3.10

is the behavior during the period between two spikes. We can also observe the same kind of difference in Figure 1 from Pospischil et al. [2008]. Thus we consider that the hardware membrane voltage reproduces the biological behavior to the utmost of its abilities.

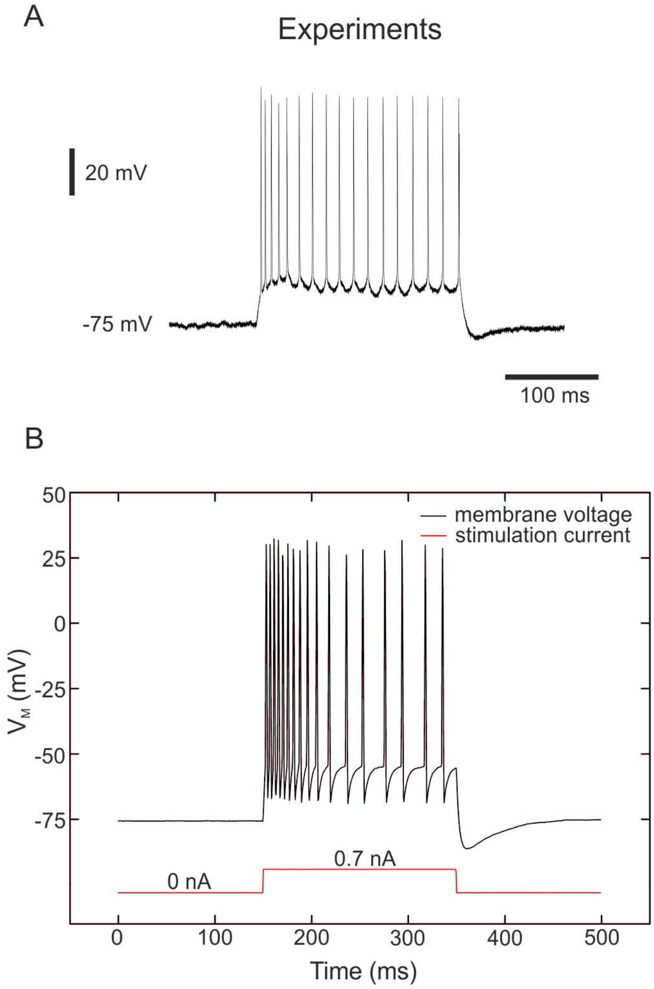


**Figure 3.10** – Membrane voltage of RS neurons. (A) Intracellular recordings of RS neurons in ferret visual cortex in vitro (Pospischil et al. [2008]; experimental data from Thierry Bal, CNRS). Responses to the injection of a depolarizing current pulse (0.7 nA). (B) Measurements of the RS hardware neuron at a depolarizing current pulse (0.7 nA) showing the typical response of a RS neuron, with spike-frequency adaptation. The VLSI voltage measurements are divided by 5 in the figure in accordance with **equation** 3.1.

The behavior of the RS neurons has been investigated further. We apply different current pulses for plotting the frequency-current relations of the RS as shown in Figure 3.11. The RS neuron has an adaptation channel, and thus we plot the instantaneous frequency (inverse of the interspike time interval) for the first and tenth spikes. Figure

3.11 shows the hardware firing rates for the first and the tenth spikes from 20 RS neurons after applying the conversion rule presented in **equation** 3.8. We plot the average and the standard deviation for the 20 hardware RS neurons.

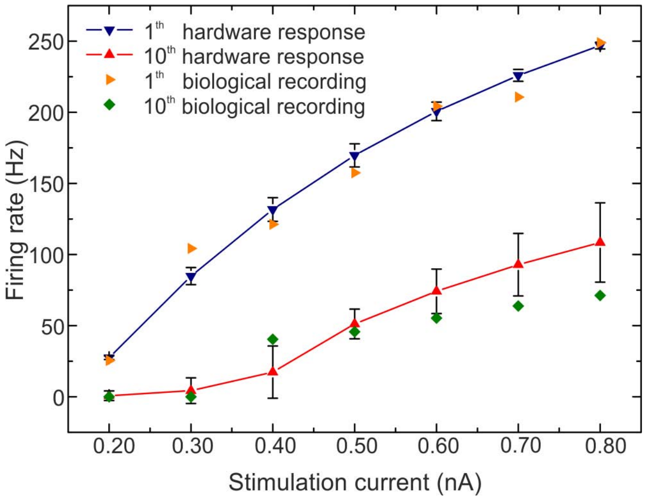


**Figure 3.11** – Frequency versus stimulation current curves of RS neurons. Biological measurements of the RS neuron from Pospischil et al. [2008] and the 20 VLSI neurons using **equation** 3.8 for the stimulation current.

For the biological data, we plot the biological recordings presented in Figure 1 from Pospischil et al. [2008]. The first spikes are similar for the hardware and the biological cells. The behavior for the tenth spike is different, especially for the high frequency. However, it is possible to observe the same difference as that in the original model in Figure 1 from Pospischil et al. [2008]. The other intermediate instantaneous frequencies do not match as well (not plotted here in order to simplify the figure as much as possible). We can observe this phenomenon in Figure 3.10 where the firing frequencies at the beginning of the activity are not identical. As explained in Chapter 2, in the section concerning the model implementation in the chip, that behavior is the result of the tradeoff between the firing rate after adaptation and the adaptation time constant to reach that adaptation. We decided to focus more on the first feature. Despite the fixed adaptation time constant, we reproduce the behavior of the RS neuron in terms of the membrane voltage, and the spike frequencies before and after adaptation.

### 3.3.3 Intrinsically Bursting Neurons

This kind of neuron responds to the injection of a depolarizing current pulse by producing bursts of action potentials and then the firing rate decreases suddenly as shown in Figure 3.12. Neuronal bursting can play important roles in communication between neurons. In particular, IB neurons (excitatory neurons in the mammalian neocortex [Connors and Gutnick, 1990]) are important for motor pattern generation and synchronization.

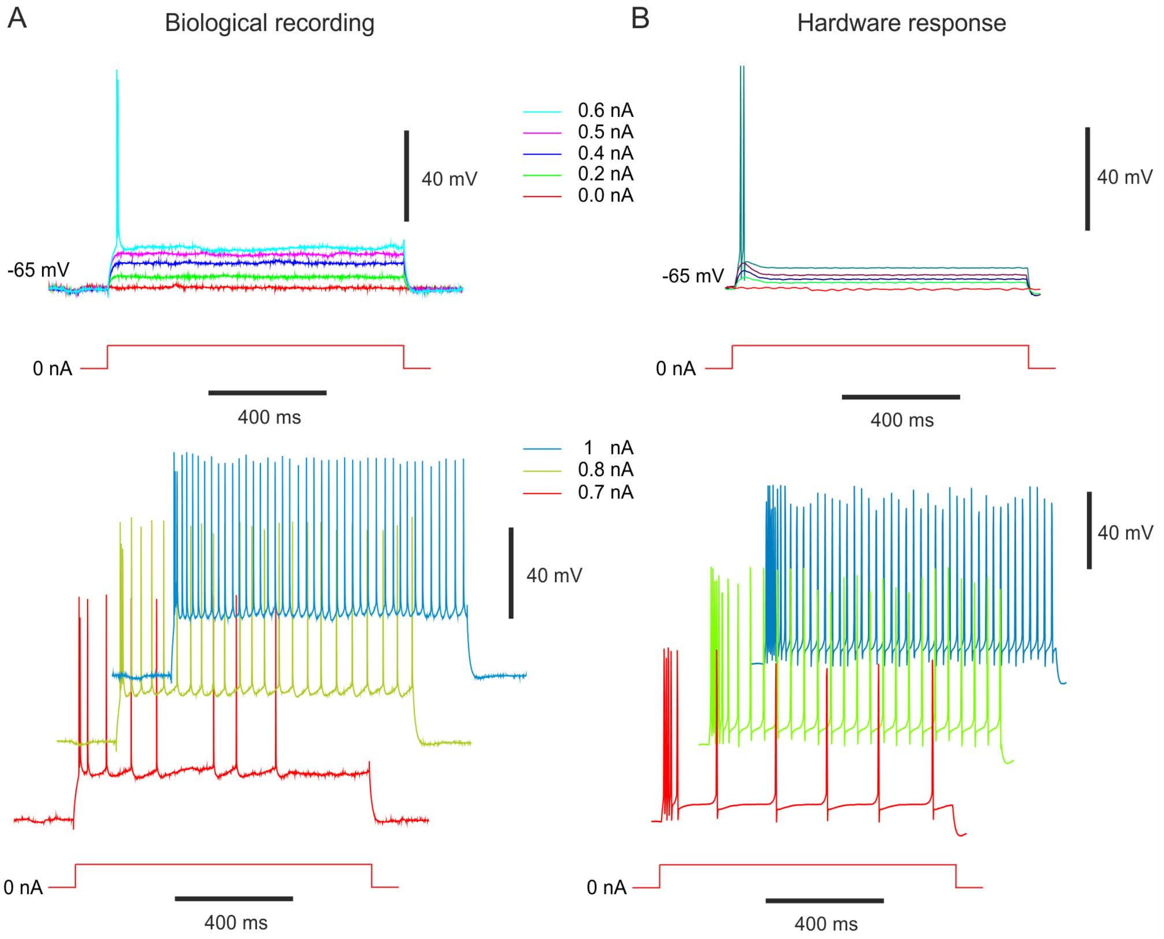


**Figure 3.12** – Membrane voltage of IB neurons. (A) Intrinsically bursting (IB) cell from guinea-pig somatosensory cortex in vitro modified from Pospischil et al. [2008]; experimental data from Cyril Monier and Yves Frégnac, CNRS). Response to different depolarizing current pulses. (B) Measurements of the IB hardware neuron at the same depolarizing current pulses. The VLSI voltage measurements are divided by 5 in the figure.

Because we have few experimental recordings for this type of cell, the frequency versus stimulation current may not be significant. For comparison, we apply the same stimulation currents to biological and hardware neurons in Figures 3.12A and 3.12B, respectively. For the weakest stimulation currents (from 0 to 0.5 nA), we observe similar membrane depolarization. For the 0.6 nA stimulation, we observe two spikes in both cases even though the frequency in the biological cell is higher. For the biggest stimulation currents (= 0.7 nA), we notice an initial fast activity (the neuron repeatedly fires discrete groups or bursts of spikes) followed by a train of action potentials. The hardware neuron is in accordance with a biological time scale.

However, we can observe two differences. The first one is the behavior of the membrane voltage between two spikes during the train of action potentials. The VLSI membrane voltage has a hyperpolarization behavior before a spike occurs. The second difference is the duration of the initial fast activity. The high frequency lasts longer for the VLSI neurons. Even though the membrane voltages are comparable for the switching frequency behavior, the effect of the L-type calcium current is to suddenly bring the neuron from one spiking frequency to another.

As in Pospischil et al. [2008], we reproduce the IB cell behavior to the detriment of the duration of the first oscillatory phase and the membrane voltage behavior during the second phase. These two differences were also observed in the original model shown in Figure 6 from Pospischil et al. [2008]. Consequently, we consider in this case that we adequately reproduce the targeted biological behavior.

### 3.3.4 Low-Threshold Spiking Neurons

Low-threshold spiking neurons respond to the injection of a depolarizing current pulse by producing adapting trains of action potentials similar to the classic "regular spiking" response of cortical neurons (Figure 3.13A). In addition, they generate a burst of action potentials in response to the injection of hyperpolarizing current pulses (Figure 3.13C). The LTS neurons correspond to inhibitory neurons [Connors and Gutnick, 1990; Gibson et al., 1999].

Like for the IB neuron, the recordings obtained from the LTS neuron are rare. The main comparison is also the behavior of the membrane voltage. The positive stimulation currents in Figures 3.13A and B show the "regular spiking" behavior of the LTS cell. The firing rate decreases while the stimulation current is applied. In both cases, the initial value of the membrane voltage is equal to -70 mV for a DC current equal to -0.11 nA. Like for the RS cell, the frequency of the hardware neuron is higher than the biological cell. This behavior is consistent due to the very similar parameters of the RS and LTS cells. The negative stimulation currents produce the rebound burst activity in Figures 3.13C and D. The initial values of the biological cell are different in Figures 3.13A and C due to a different DC stimulation current, which are equal to -0.056 and -0.11 nA for the biological neuron and the VLSI neuron, respectively.

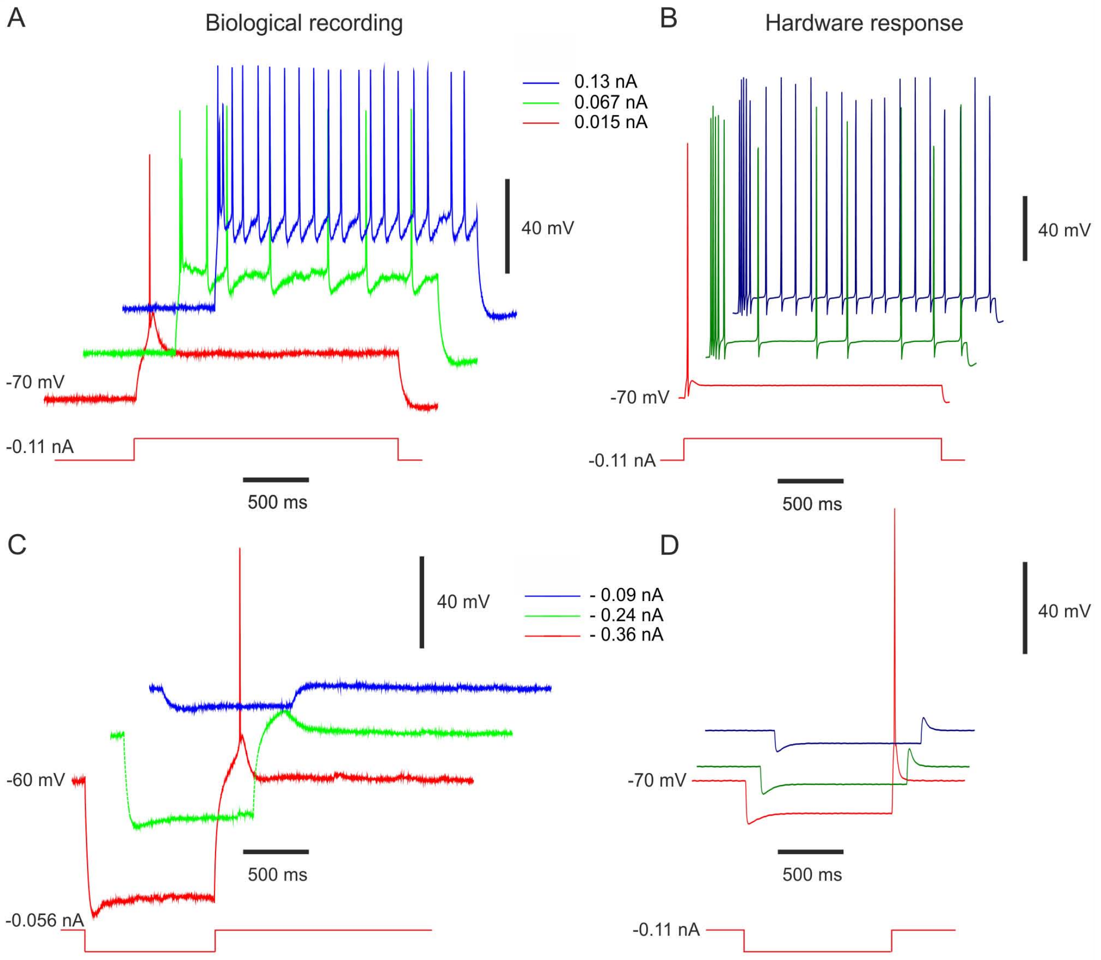


**Figure 3.13** – Measurements of LTS-Bursting behavior. Response of biological LTS cells from rat somatosensory cortex in vitro (Pospischil et al. [2008]; experimental data from Maria Toledo-Rodriguez and Henry Markram, EPFL) to (A) depolarizing and (C) hyperpolarizing stimulation pulses. Measurements of the LTS hardware neuron with the same (B) depolarizing and (D) hyperpolarizing stimulation currents applied. The VLSI voltage measurements are divided by 5 in the figure.

When the stimulation current is applied, the hardware LTS is less depolarized than the biological neuron. However, when the stimulation ends, our hardware LTS neuron reproduces the same behavior as the biological cell. We also observe a slightly higher depolarization for the hardware neuron with -0.09 and -0.24 nA current stimulation. For -0.36 nA, one spike occurs in both cases. We consider that we adequately reproduce the targeted biological behavior of the LTS cell.

## 3.4 Conclusion

As already mentioned, we want to insert silicon neurons among biological neurons. Thus, we decided to qualitatively compare the dynamics of our silicon circuit to biological cells with a level of detail that has never been seen before in a silicon neuron. To reach our goal, we propose a simplified version of the HH formalism and the appropriate parameter sets of the FS, RS, IB, and LTS neurons that can be implemented in our analog neuromimetic chip. Therefore, in this chapter, we presented the hardware platform and the proposed full-custom fitting method in voltage-clamp mode to tune our neuromimetic integrated circuits. This optimization method, based on the DE algorithm that we proposed [Buhry et al., 2011], is an alternative to the estimation methods associated with voltage-clamp measurements. In any event, we observed a large discrepancy for all of the parameters, confirming the necessity of the tuning step. As with any circular problem, we chose an arbitrary starting point to solve it. Even though the optimization phase of our tunable chip is time-consuming, we will save time in the emulation phase due to three features. First, our chip requires only one tuning of its parameters. The parameter sets are then stored in a database and the required parameters are loaded into the chip as needed to emulate any given cell type. Second, the model parameters can be modified on-line at any time. Third, the neuron type can also be modified on-line by connecting/disconnecting an ionic channel and/or modifying a few parameters. For example, the burst in the IB cell can be modulated by the value of the conductance or time constants of the calcium current. All of these on-line changes take only a few microseconds. This will enable the user to alter parameters in order to study their effect on the dynamic of the biological network. Finally, we tested the parameters obtained by comparing the behavior of the membrane voltage in the recordings of biological cells with membrane voltages simulated with our chip. This comparison is possible because of the translation rules between biological and hardware neurons based on the chip characteristics. We directly compare the behavior of our chip with biological recordings. Our results show that our system is able to reproduce the main features of four common classes of cortical cells. These ASICs can be used to form the core of various simulation platforms, designed to emulate neural networks in biologically relevant configurations. In the next chapter we will see the spiking neural network (SNN) simulator developed by our team, called PAX (Plasticity Algorithm for Computing System), that hosts the chip. We will focus on our contribution to the development of this mixed hardware-software platform for a real-time spiking neural network simulation, in terms of software tools and a VHDL driver for neuron configuration in the platform.

# Chapter 4

# Plasticity Algorithm Computing System

As already mentioned, we want to insert silicon neurons among biological neurons. Therefore, we decided to qualitatively compare the dynamics of our silicon circuit to biological cells. These ASICs can been used to form the core of a simulation platform designed to emulate neural networks in biologically relevant configurations. The spiking neural network (SNN) simulator developed by our team, called PAX (Plasticity Algorithm for Computing System), is based on a mixed analog/digital architecture designed to have the capacity to address biological diversity in terms of neuron types as well as plasticity rules. PAX makes use of point-neuron conductance-based models and is controlled by spike-timing dependent plasticity. In the previous chapters, we saw how to extract the parameter sets for our analog neuromimetic chip, and how to reproduce the main firing features of cortical cell types. Now, one important issue remains: how can we calibrate the spiking neural networks? - There are two main types of parameters that we have to tune: the plasticity configuration parameters and the ASIC configuration parameters. The first type of parameters involves sending the user-defined network topology, the STDP parameters and the initial synaptic weights (B. Belhadj addressed this as well as the design of the STDP processor in the PAX in his thesis [Belhadj, 2010] and tested it using a virtual spike generated in the VHDL driver instead of an "analog neuron"). The second type of parameters involves sending the Hodgkin-Huxley parameter values for the ASIC configuration, which can be used to specify the neuron type (see Chapter 3) and synapse behavior. Therefore, we need to tune the synapse in a biological range, as done for the neurons. Thus in this chapter, we describe the work done to integrate the silicon neuron in the PAX system, using the same parameter sets for the FS, RS, IB, and LTS neurons that we proposed for our analog neuromimetic chip. We describe the methodology adopted to tune the synapses followed by the improvements made to the PAX simulator.

Section 4.1 briefly describes the main abstractions present in the PAX at the neuron and network level while focusing on the method proposed for the synaptic tuning. In section 4.2 we describe the features and basic functions of the PAX simulator and then in section 4.3, we present our contribution to the development of the PAX in terms of software tools and the VHDL driver for the neuron configuration in the platform. Finally, we show the results of this work.

## 4.1 Abstractions in the PAX simulator and synaptic tuning

The relevance of a neuromorphic simulator is measured by its ability to reproduce the observed phenomena in biology. However, given the complexity of biological systems, we are forced to neglect certain details. As a result, we lose accuracy compared to the original structure. Abstraction is a natural approach to eliminate details deemed not useful in the reproduction of a given phenomenon. However, any abstraction must be justified by the relevance of the final results. This section briefly summarizes all of the abstractions used in the PAX simulator while focusing on the synapses. It will be useful to understand how the PAX simulator works. Therefore, we propose a methodology to calibrate the synapses.

### 4.1.1 Neuron level

As previously explained (Chapters 1 and 2), at the cellular level, the soma follow the Hodgkin-Huxley formalism. Therefore, an initial abstraction treats the membrane as a single compartment. This abstraction implicitly assumes that ion channels of the same type are grouped in the same conductance. A second abstraction (see Chapter 3) consists of an amplification of the biological electrical parameter values to manipulate the current values using integrated circuit technology and to improve the dynamics of the system. However, there is no temporal abstraction at the neuron level. Therefore, the evolution of the silicon neuron[1] is continuous and on a biological time scale similar to its biological counterparts. This choice allows interfacing with living cells [Le Masson et al., 2002].

[1] Different silicon neuron have been developed in our group, including the Galway chip, which is the ASIC used in this work.

### 4.1.2 Network level

Several abstractions are used at the network level. A first abstraction was done for the synapses.

**Synapse abstractions**

It was chosen to implement the synaptic interactions using "exponential" synapses (Figure 4.1 ) in the Galway ASIC [Bornat, 2006], where the synaptic conductance increases for a given "quantal conductance" when a presynaptic spike occurs, and then relaxes exponentially to zero.

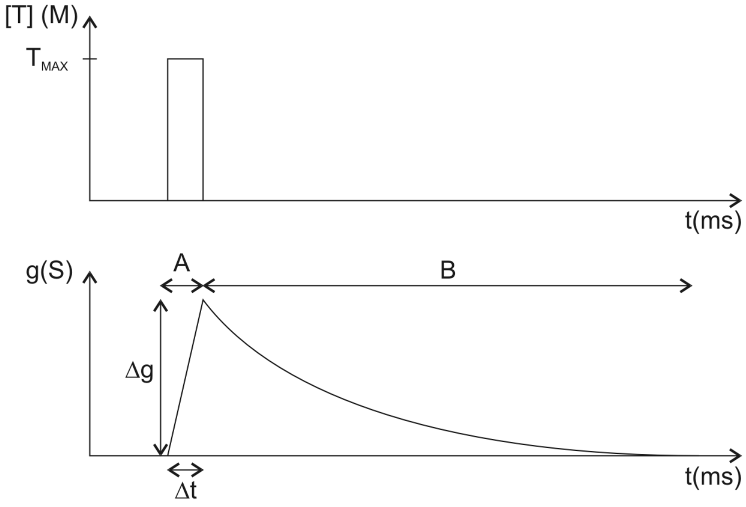


**Figure 4.1** – Exponential-decay synapse principle.

The dynamics of the synaptic conductances are captured using a two-state (open/closed) scheme, in which opening is driven by a pulse. This model can handle phenomena such as summation or saturation, and accurately describes the time course of the synaptic interactions. The associated postsynaptic current is given in **equation** 1.16 and **equation** 1.17 (see Chapter 1). Figure 4.1 illustrates the time-variation of the synaptic conductance when a transmitter concentration pulse [T] occurs, assuming that the transmitter is released when a presynaptic action potential appears. As the quantum Δg is proportional to the Δt pulse width, this latter parameter will be exploited to modulate Δg for the synaptic plasticity.

A second abstraction was implemented in the Galway ASIC for the synapse. It is a structural abstraction that avoids a high density of synaptic connections. Synaptic inputs to a neuron are grouped in only one collective synapse called a multisynapse [Destexhe et al., 1998a]. The multisynapse adds all of the stimulations caused by presynaptic neurons of the same type (inhibitory or excitatory) to stimulate the cell membrane of the postsynaptic neuron. This means that the presynaptic events are collected at a single input to generate the presynaptic pulse (Figure 4.2).

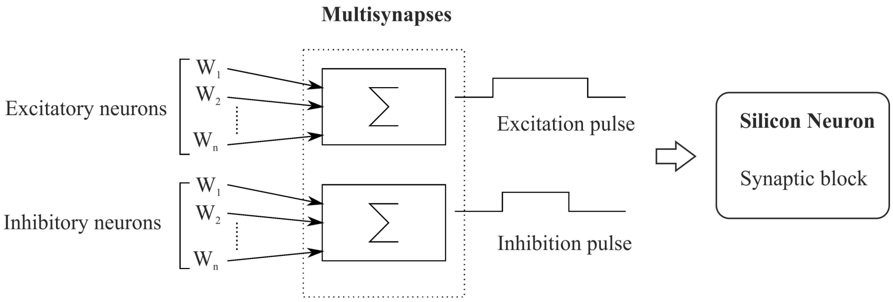


**Figure 4.2** – Multisynapses block.

The width of this pulse (60 $\mu$s is the maximum value before saturation, as measured in the Galway[2] ASIC) triggers the transition to the opening state. Together with g (maximum conductance), it encodes the strength of the multisynapse, and is dynamically updated.

### Network abstractions

Synaptic connections that link neurons are allocated in a virtual way: there are no physical connections between the cells. Each neuron is connected to a comparator. It retrieves the events of the outputs of the presynaptic neurons and then it transmits a stimuli to the inputs of the postsynaptic neurons, based on the network connectivity. The network connectivity, chosen by the user through the connectivity matrix (see Appendix A.3), determines which neuron is connected to which other neuron. This technique is known as a virtual synapse technique.

Another abstraction concerns the action potentials. We assign boolean values to indicate the presence or absence of an action potential. This assumption facilitates the processing and transmission of the events of neurons. Finally, for reasons of compatibility with the cellular level, there is no temporal abstraction made at the network level. We always maintain biological real-time operability.

[2] As a consequence, there will be a limit to the maximum number of presynaptic neurons that can be fired at the same moment with the maximum strength.

### 4.1.3 Synaptic tuning: proposed method

The Galway chip is fully reconfigurable. Therefore, it is possible to tune the synapse conductance value and the kinetics (time constant) of the exponential function (Figure 4.1) using the synaptic output current available in the Galway chip. As indicated in Bornat's thesis (2006), it is necessary to tune the conductance parameter and the time constant of the exponential decay for the synapse block. An efficient method for computing synaptic conductance was proposed by Destexhe et al. [1994] which describes the time course and the summation dynamics of the two principal types of synaptic interactions in the central nervous system: the AMPA and GABA$_A$ postsynaptic receptors for excitation and inhibition, respectively. As shown in Figure 4.3, we simulate the behavior of the AMPA and GABA$_A$ synapses.

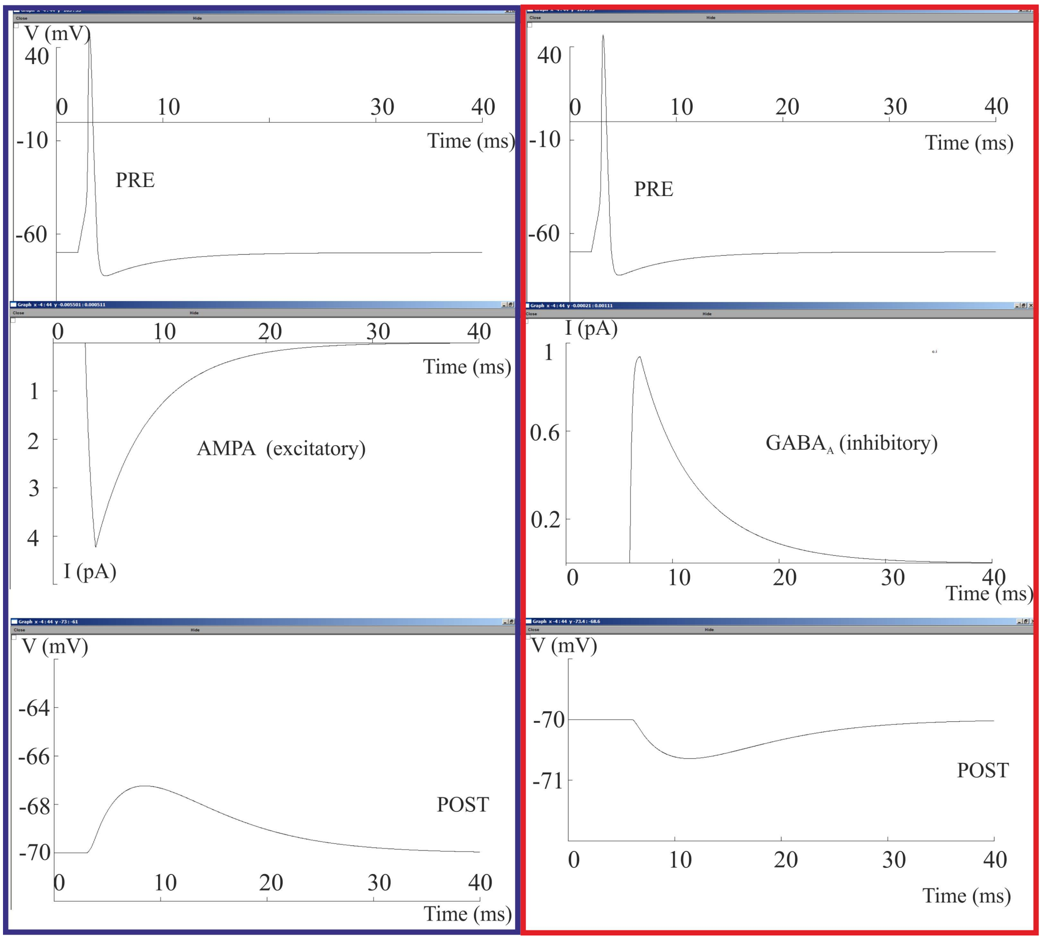


**Figure 4.3** – Simulation of postsynaptic potentials (top), a synaptic current (middle) and a presynaptic spike (middle) in simplified kinetic models for the AMPA (left) and GABA$_A$ (right) receptors.

We extracted the parameter values for **equation** 1.16 and **equation** 1.17 from the work of Destexhe et al. [1998a], which can be used to generate references for our synaptic tuning, using a conductance value of 0.1 nS and $E_{syn}$ = 0 mV, $E_{syn}$ = -80 mV for both AMPA and GABA$_A$, respectively. The membrane potentials were simulated using the NEURON software [Hines and Carnevale, 1997]. To tune the kinetics, we can consider that the time constants are roughly respectively 3 ms and 10 ms for AMPA and GABA$_A$. As shown in Figure 4.3, the biological values of the synaptic current are very low (pA). Therefore, it is not possible to tune the synapses on the Galway chip using the synaptic current output of the chip. Even if we increase the conductance value to 40 nS, we cannot measure the corresponding hardware synaptic current. This aspect, which was not considered in the design phase for the Galway chip, will affect the final results as we will see in the last section of this chapter. We are obliged to consider the contribution of the postsynaptic potential to tune the synapse.

As shown in the simulation in Figure 4.3, if we have a presynaptic spike, the contribution in terms of the postsynaptic membrane voltage is roughly +3 mV for AMPA and -1 mV for GABA$_A$, which will be converted to the hardware value as explained in Chapter 3. For every silicon neuron, it is necessary to calibrate both the excitatory and the inhibitory synapse. As suggested by the simulations (Figure 4.3), we will set the postsynaptic neuron to resting potential and then we will connect a presynaptic spike in a "virtual way", as explained in the abstractions. Thus, we need to tune the synapse directly in the PAX system[3]. We will show the results of the tuning in the last section of this chapter.

## 4.2 The PAX simulation system

Here, we describe the PAX platform which is dedicated to the simulation of cortical neural networks. It was developed by Belhadj [2010] within the framework of a collaborative project (EU project FACETS, FP6-IST-FETPI-2004- 15879), the aim of which is to lay the theoretical and experimental foundations for the design of novel computing hardware that exploits the concepts experimentally observed in the brain. Within the FACETS project, two hardware platforms have been developed to emulate SNNs. This section is dedicated to the first platform, called PAX. The second FACETS platform was developed at the University of Heidelberg [Schemmel et al., 2007] and integrates, at the wafer scale,

[3] Please note that we can also tune the synapse, not in the PAX, by sending a pulse with the FPGA directly on the synaptic block. It is equivalent to having a presynaptic spike connected to the silicon neuron.

large networks of LIF-modeled neurons in accelerated time. We start by describing the general approach used for the design and then we focus on the parts that have already been developed. In the next section (4.3), we will see the contribution that this thesis makes to the development of the platform.

### 4.2.1 The methodological approach

The PAX simulation system is organized into three layers as shown in Figure 4.4.

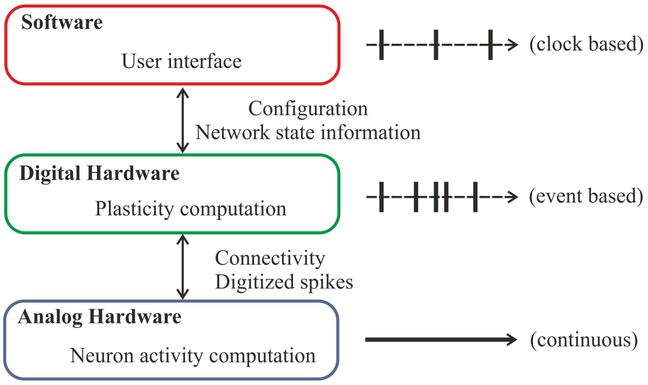


**Figure 4.4** – PAX platform processing layers and data flow.

The analog hardware layer runs continuous computations of the neuronal activity on analog ASICs. The digital hardware layer is in charge of collecting spike event information from the analog neurons and controls the synaptic connectivity, which feeds back to the analog hardware. The next layer includes the software driver and interface, which are in charge of controlling the transfer of bi-directional data to the software via a USB connection. The details for each layer are provided below.

#### Analog computing core

The core of neural computation must address the compatibility constraints with biophysical models. As already explained in Chapter 3, the main features of this analog computing core are:

- that computational models of spiking neurons are based on experimental data related to the electrophysiological behavior of neurons in different cortical areas and the electrical quantities are proportional to the biological quantities;
- time-dependent phenomena must be done in biological real-time;

- the parameterization of computational components must be flexible and easily modified to allow the configuration of several classes of neurons.

Finally, the computational cores of the ASICs are neural elements, each of which is able to process the membrane voltage. It also includes a threshold comparator used for spike detection and two synaptic conductances, respectively following the AMPA-type (excitatory) and GABA$_A$-type (inhibitory) kinetic models. Considering that each spike detector converts the membrane potential to a 1-bit code, each neural element has two output representations of the membrane potential: the continuous analog value and the asynchronous digitized spikes.

### The digital hardware layer

This layer must be able to reconfigure neural connections depending on the network topology chosen. This means that all of the possibilities of inter-neuron connection must be covered, ranging from a non-connected network to a completely connected network. It is also possible to add information about each connection: plastic or not, excitatory or inhibitory, synaptic initials weight, etc. The connections are governed by a plasticity algorithm (STDP). A reprogrammable architecture allows the rapid implementation of any changes in the plasticity model.

The analog custom ICs are controlled by the digital hardware layer. This layer is in charge of collecting spike event information from analog neurons as well as computing STDP algorithms and controlling the synaptic connectivity back to the analog hardware. In more detail, the digitized outputs of each "neural element" are directly connected to FPGA inputs. Once captured by the FPGA, these signals represent spike events, which are computed by the STDP processor. The digital hardware layer is also in charge of generating the synaptic weight-triggering signal (Figure 4.5).

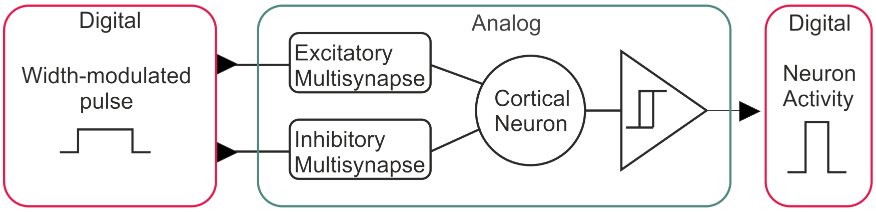


**Figure 4.5** – Structure and I/O for a neural element hardware.

This depends on the efficiency of each synaptic connection. This efficiency is calculated numerically by the plasticity algorithm and is then converted into a digital pulse (with 8-bit resolution), whose width encodes the synaptic weight value. The inhibitory or excitatory nature of the synaptic current is determined by its sign, which is either positive or negative. Using the multi-synapse scheme, the neural network can handle all-to-all connections, regardless of the number of neurons in the network.

**The software layer**

Within the software layer, a computer running a real-time operating system hosts dedicated software. It provides user interface functions to control the simulation configuration both offline and online and to collect network state information. All of the simulations are preceded by a configuration phase (see Appendix A.3) involving the definition and transmission of a set of parameters (neural network topology), and the preparation of external stimulation patterns (if needed).

### 4.2.2 A multi-board spiking neural network platform

Figure 4.6 shows the block diagram of the global system. The PAX simulation system comprises a PC, running under LINUX, and a multi-board system called a "rack".

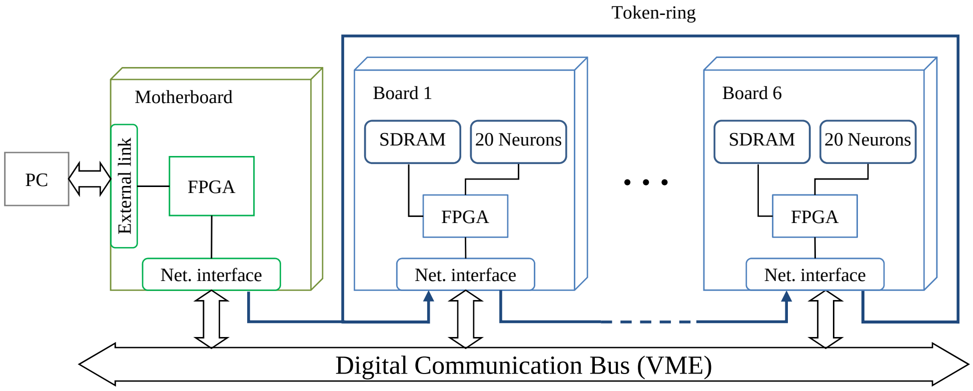


**Figure 4.6** – Structure of the neural network simulation system.

This rack comprises several (up to 20) daughter boards (Ekerö board, see Chapter 3) and one mother board, all connected via a digital communication bus. The current version of our system can host up to 120 neurons spread across six similar circuit-boards and can be extended up to 20 boards all connected to a backplane with daisy-chain facilities. Each Ekerö board hosts four analog ASICs and one Xilinx Spartan3 FPGA (denoted as FPGA_DB in Figure 4.6). Each ASIC incorporates five neurons which compute in analog mode conductance-based models following the Hodgkin-Huxley formalism. The neuron type, firing rate and response to stimulus can be configured in each neuron. When the neuron output comparator detects an action potential, a digital 1-bit event is transmitted to the FPGA_DB. In turn, the FPGA_DB transmits the address of the firing neuron across the communication channel according to a token-ring access policy defined below. In the meantime, the other FPGA_DB of the other boards scans the incoming events and selects addresses that have a connection with one or many local neurons. The input events are selected according to "virtual" connections (see Appendix A.3) stored in each FPGA_DB. Finally, the FPGA_DB computes synaptic changes following the STDP rules[4], and generates a digital pulse whose width encodes the synaptic weight. This pulse triggers the transition to the opening state of the synaptic channels in each postsynaptic neuron [Belhadj, 2010].

The network topology consists of boards connected by point-to-point links forming a circle, i.e. the token ring. A 1-bit pattern (called the token) circulates around the ring (from board i to boards i+1, i+2, etc. up until board 6, and then to boards 1, 2, etc.), granting permission to send messages (if any). Message transmission is done asynchronously over the parallel bus. Inter-board communication operates via a one-to-many broadcast. The motherboard kicks off the circulation of the token and controls the evolution of the simulation by scanning the activity over the parallel bus. Messages are then asynchronously transmitted over the bus, which signifies that there is no clock used to synchronize transmission between the boards. A photograph of the global system is shown in Figure 4.7.

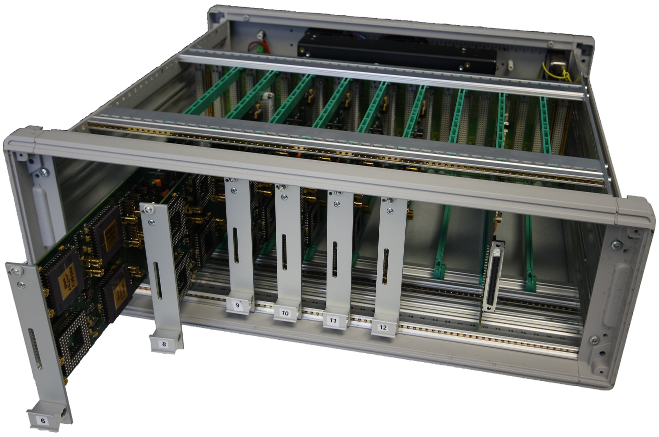


**Figure 4.7** – Photograph of the rack.

[4] This part was implemented in a previous thesis [Belhadj, 2010] using a "digital" spike instead of an analog neuron.

A commercial power supply is used; it can deliver 60 A under 5 V. The mother board is connected to the PC via a USB serial interface.

### Structure of the mother board

The mother board, called Thalamos (Figure 4.8), is a 6-layer full-custom board.

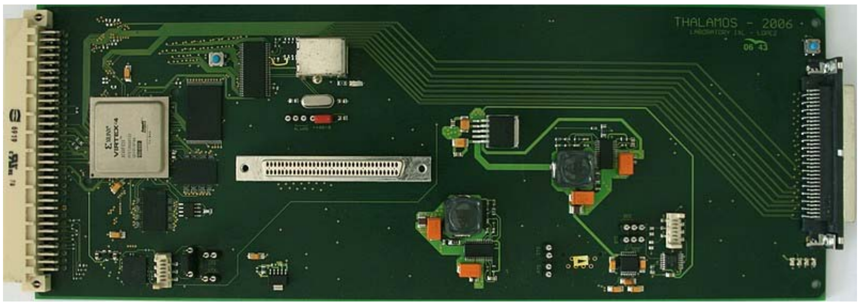


**Figure 4.8** – Photograph of the mother board Thalamos.

It hosts one FPGA (Xilinx Virtex 4 XC4VFX20, denoted as FPGA_MB in Figure 4.6). The FPGA_MB has an external clock at 100 Mhz; it is associated with two SDRAM (256 Mbits each) clocked at 50 MHz and can be programmed by JTAG and a microcontroller Cypress FX2 used for USB connection[5]. This board also comprises VME and USB connectors.

### VME backplane with daisy-chain capability

All of the boards are connected to a VME backplane via a specific connector (Figure 4.9).

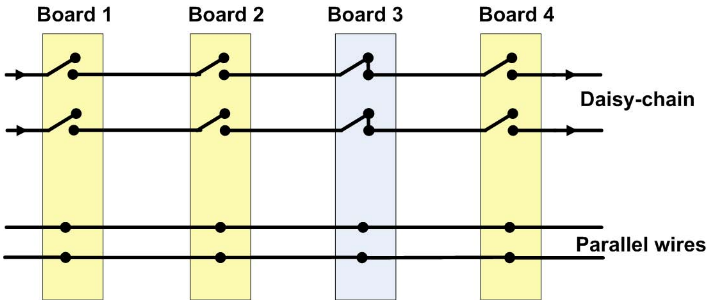


**Figure 4.9** – Synoptic of the backplane: with a daisy-chain, an empty slot (board 3) results in a short in the input/output of the daisy-chain.

[5] The maximum number of neurons that we can connect is limited by the USB connection. The amount of data was analyzed in B. Belhadj's thesis (2010); it was established that we can connect a maximum of 112 neurons.

Each connector has 96 contacts:

- six contacts for power supply (0 V, 5 V), corresponding to six backplane wires;
- 10 contacts dedicated to 5 daisy chain lines. The wire continuity is performed if the slot is empty, otherwise the FPGA_DB located on the plugged board manages the connectivity of these two contacts (Figure 4.9);
- 80 contacts for high rate data exchange (corresponding to 80 backplane wires).

Each wire supports a throughput of 25 Mbits/s. Since the FPGA (FPGA_MB and FPGA_DB) is in charge of transmitting messages over the bus, it must respect the bandwidth capacity of the metal wires. Each FPGA FPGA_MB and FPGA_DB) can transmit messages with a throughput of 100 Mbit/s which must be divided by four to match the wire backplane capacity. Thus, the amount of time required to transmit one message is 40 ns.

#### Tasks of the PAX platform

We can consider three different operations that must be performed in the layers that make up the PAX platform: operations that must be performed before, during and at the end of simulation. Before running a simulation on the PAX platform, the following tasks must be carried out:

- Task 1: the digital hardware layer has to receive the configuration parameters of the neurons (according to the chosen model) via the software layer, and send them to the corresponding ASICs;
- Task 2: the digital hardware layer has to receive the plasticity configuration parameters: this consists of a user-defined network topology, the STDP parameters and the initial synaptic weights (developed by Belhadj [2010]).

During a simulation using the PAX platform, the following tasks must be carried out:

- Task 1: the digital hardware layer has to receive spike events from the analog layer;
- Task 2: the digital hardware layer has to compute the STDP algorithms in real time and update the synaptic weights and then send the synaptic input signal to the postsynaptic neurons (developed by Belhadj [2010] and tested with the digital spike generated by the FPGA);
- Task 3: the digital hardware layer has to send information about the state of the network to the software layer.

After a simulation in the PAX system, the following tasks must be carried out:

- Task 1: the software layer has to manage the results of the simulations;
- Task 2: the software layer has to provide a final user-friendly interface to plot the data.

Table 4.1 summarizes these operations by specifying what we have to implement in the PAX platform.

**Table 4.1** – Tasks of the PAX platform performed before, during and at the end of the simulation that we must implement in this work.

| | Task 1 | Task 2 | Task 3 |
|---|---|---|---|
| before simulation | not done | done | not present |
| during simulation | not done | done | done |
| after simulation | done | not done | not present |

Figure 4.10 illustrates the different tasks of each of these layers. The digital hardware layer is a set of FPGAs.

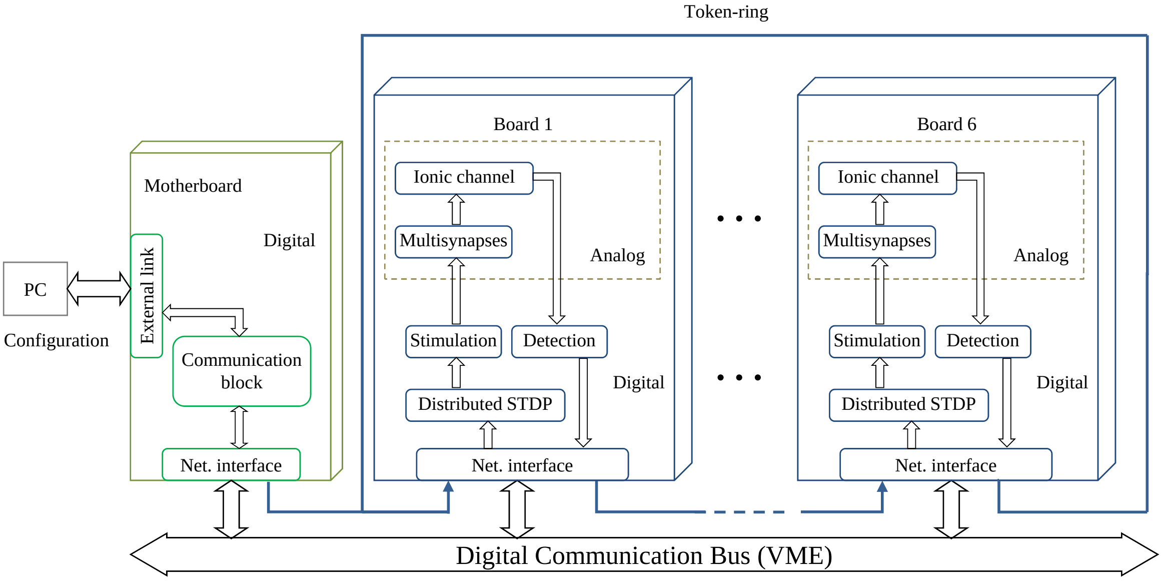


**Figure 4.10** – Structures and functions in the neural network simulation system PAX.

The digital communication between the system and the workstation is assured by the FPGA_MB of the mother board through a USB connection. The latter is in charge of conveying information to (simulation results) and from (configuration data) the workstation. On the other hand, the FPGA_DB of the daughter boards are in charge of assuring inter-board communication and the computation of the plasticity as well as the management of the analog/digital interface.

The communication between the digital and analog hardware layers in each daughter board forms a local simulation loop. Whenever a neuron generates an action potential, this will later be detected by the local FPGA_DB, in which it will be routed towards the target board according to the connectivity of the network, and then to the target ASIC. In the opposite sense, whenever an external action potential arrives to the FPGA_DB, it will set off the computation of the synaptic weights for the plastic connections and then stimulate the target neurons by sending a width-modulated pulse to the corresponding multisynapse(s). A synaptic current is then added to the silicon neuron and either helps generate a new action potential (excitatory synapse) or inhibits its generation (inhibitory synapse).

## 4.3 Tasks implemented in the PAX platform

In this section, we present the contribution of this work: the software layer provides a user interface that helps generate configuration parameters for the computational processes of the system and then the digital interface designed in VHDL interfaces the analog ASIC in the PAX platform.

### 4.3.1 Workstation software

The host computer runs on a real-time patched GNU/Linux operating system based on the Ubuntu distribution. The user defines, in the developed C code, the simulation to be performed (neuron types, network connectivity, initial synaptic weights, and communication parameters, as shown in Figure 4.11).

The developed C code converts the data that we have to send to the platform into a frame format for the digital bus (see the next section for the details of the frame format). Therefore, the workstation will send this configuration to the mother board, which in turn will distribute it to the daughter boards. This procedure defines the configuration phase of the system. The Thalamos board uses a USB serial interface to communicate with the PC. The interface relies on the dedicated Cypress FX2 micro-controller, which is connected to the FPGA_MB (Figure 4.6). In order to send the correct amount of data, we modified this driver. The VHDL code has been developed to read and send data to and from the FX2.

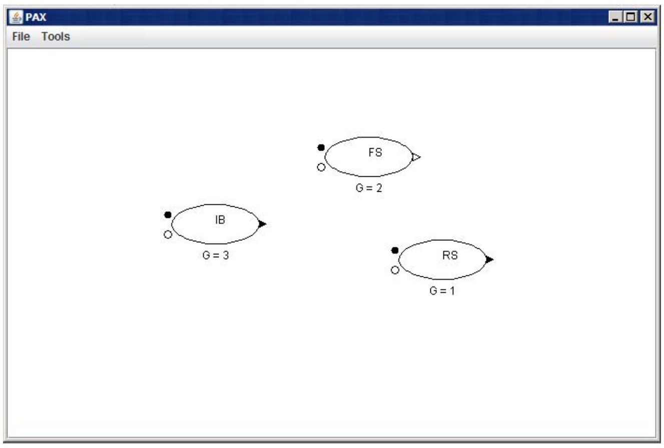


**Figure 4.11** – Software interface in the host computer for the configuration phase. As an example, three unconnected neurons (FS - RS - IB) are presented, where G indicates the neuron number.

This serial link is only used to configure the network, to send the parameters and to transfer offline data in order to process post-simulation data. The results are presented in the next section, where we validate the driver by sending parameters in the PAX platform.

### 4.3.2 VHDL ASIC module for the PAX system

The analog hardware layer comprises the ASICs of each daughter board. With four ASICs per board, each ASIC incorporates five neurons and 15 multisynapses (three multisynapses per neuron). ASICs compute, in continuous time, the electrical activity of neurons following the Hodgkin-Huxley formalism.

There are three types of parameters that the software layer has to transmit to the mother board and that the mother board then has to transmit to the ASICs:

- ASICs topology parameters for the specification of neuron type and synapse behavior (digital parameters);
- ASICs data parameters for the neurons and synapses (analog parameters);
- ASICs stimulation parameters for an external stimulation current (analog).

The serial communication (USB) between the PC and the mother board allows the FPGA to receive the digital and analog parameter values given by the user and then to write the data on the VME bus.

### Protocol for the configuration phase

To convey the data from the mother board to the daughter boards, a one-directional communication protocol for the system was designed. The frame format is given in Table 4.2.

**Table 4.2** – Configuration of the frame format.

| Board number | Module identifier | Configuration data |
|---|---|---|
| (8 bits) | (8 bits) | (32 bits) |

The frame format contains the board number to address a specific board in the PAX system, the module identifier, which selects the VHDL driver for the specific types of parameters, and then the configuration data.

The configuration data frame format that we used for the analog and digital ASIC parameters is given in Tables 4.3 and 4.4, respectively.

**Table 4.3** – Frame format of the ASIC analog parameters.

| 0 | Data | Address | ASIC | Type (A/D) |
|---|---|---|---|---|
| (6 bits) | (14 bits) | (8bits) | (3bits) | (1 bit) |

**Table 4.4** – Frame format of the ASIC digital parameters.

| 0 | Data | ASIC | Type (A/D) |
|---|---|---|---|
| (14 bits) | (14 bits) | (3 bits) | (1 bits) |

The driver for the ASIC parameters follows the protocol for analog and digital parameters, as already explained in Chapter 3.

### ASIC module task development

The configuration frames are sent by the mother board, by broadcasting, over the communication bus (Figure 4.12). The daughter boards identify whether or not these frames concern them. They test the board number field in the header of the frame to check if it matches their own number. If this is the case, the board takes the frame into account and continues on to a decapsulation procedure to identify the target internal module. The designed VHDL module (Figure 4.12) drives the refreshment procedure of the analog memory cell array that stores the parameter values of each individual neuron.

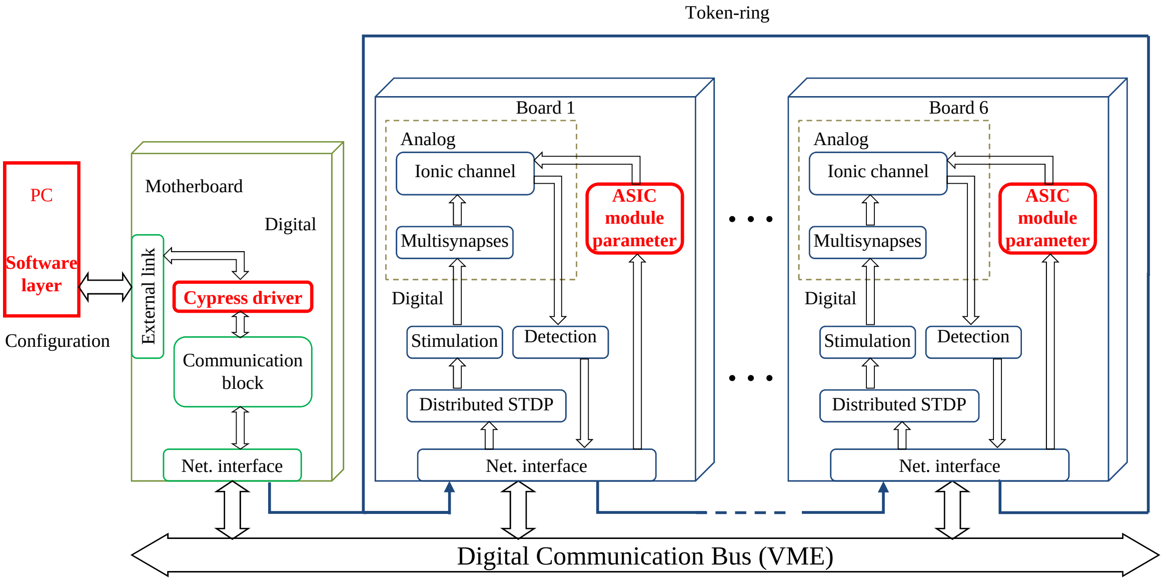


**Figure 4.12** – Structures and functions in the PAX neural network simulation system. The red parts indicate the added block, in this thesis, with respect to Figure 4.10.

The driver works on the data extracted from the digital bus making a decapsulation procedure. It decides what neuron is addressed and then it will decapsulate the data into 14 bits for the DAC (analog parameters) and for the switch inside the ASICs (topology setting). Furthermore, it generates the clock (100 kHz) for the DAC and for the digital block (as previously explained in Chapter 3).

## 4.4 Results

In this section, we present the results of the tasks implemented in the PAX system. We start with the measurements that are useful for validating the implementation of the VHDL module for the configuration of the ASICs as well as the software layer developed on the workstation. Finally, we present the synaptic tuning results.

### 4.4.1 Raster plot of neurons in the PAX system

To validate the software layer in the workstation and the VHDL module that receives the parameters, we can directly test on the ASICs if the parameters are received (an example of the parameter data for the neuron configuration is presented in Appendix 4.5). We start with the selection of five FS neurons in the software interface (Figure 4.13) with a frequency in the 13-27 Hz range, without sending the parameters.

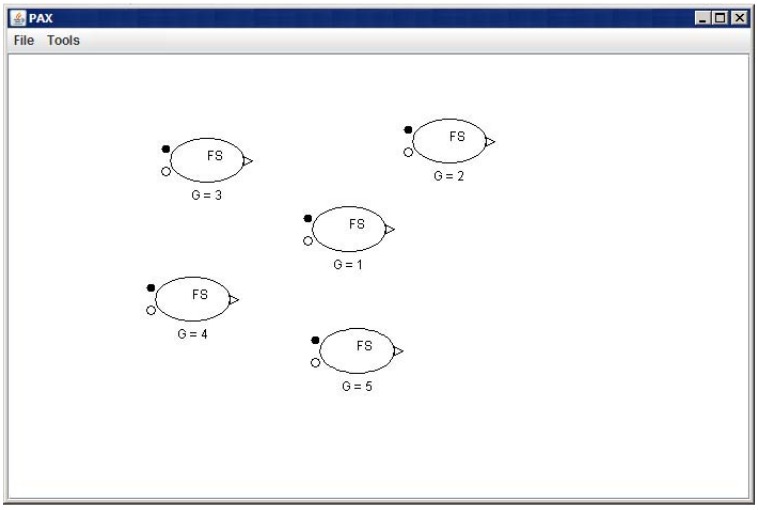


**Figure 4.13 –** Software interface in the host computer for the configuration phase of five unconnected FS neurons.

For each Galway chip, a matrix of switches has to be configured to control the topology of each neuron with a digital word, as already explained in Chapter 3. To test the module, we send one 14-bit word (for the topology) to the dedicated serial input of one Galway chip and we measure the chronograms of the ASIC inputs as shown by Figure 4.14.

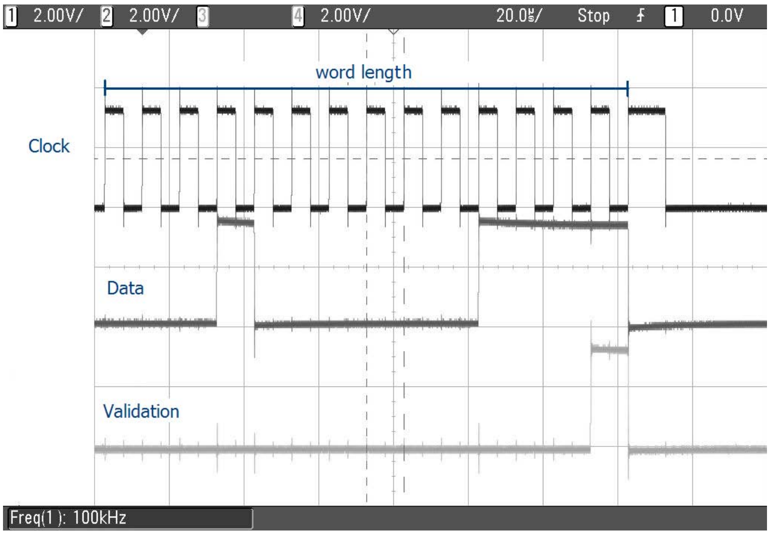


**Figure 4.14 –** Chronograms of the ASIC inputs for the topology configuration.

The data shown in Figure 4.14 are related to only one parameter value for the topology that is received by the ASICs in the correct way. In any case, in order to simulate neuronal activity, each neuron needs a set of parameters, encoded as voltage values, that are stored in the analog memory cells inside the Galway chip. The values of the original parameters are encoded as 14 bits, and are stored in a RAM synthesized into the FPGA_DB and sent via a serial DAC to the ASIC. For the five neurons, i.e. for one chip, we have to refresh 205 analog parameters every 2 ms. Thus, to test the VHDL ASIC module (and the software tools), we need to send all of the parameters for some neurons and detect the neuronal activity on the rack (see Appendix A.3 for details on the PAX configurations).

Therefore, we send the parameters for one FS neuron and then we measure the firing rate (Figure 4.15). The membrane voltage measurements are not presented because they are the same signals as those shown in Chapter 3. The difference lies in the fact that they are now measured in the PAX system, even though they were previously measured in a single board system.

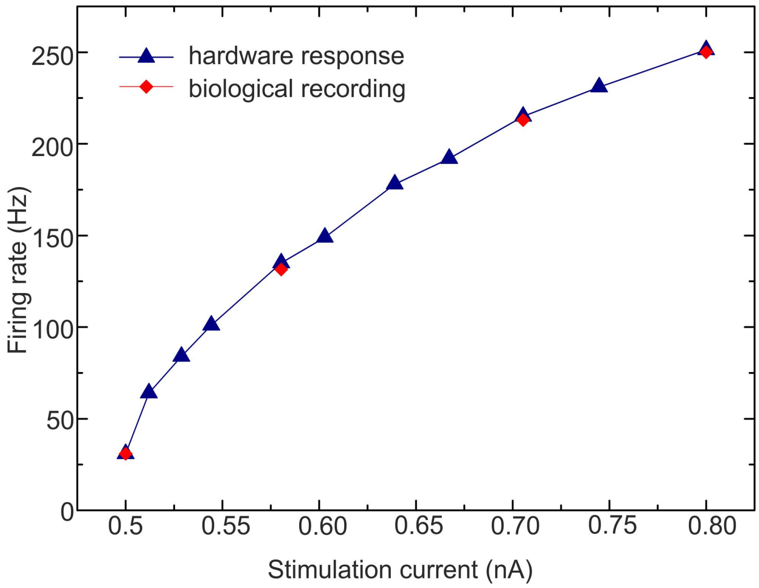


**Figure 4.15** – Firing rate of one FS neuron measured in the PAX system.

As shown in Figure 4.15, the measured firing rate provides another confirmation that the system works properly. The firing rate is within the same range as the firing rate that was already investigated in Chapter 3 for the FS cells.

Furthermore, we sent the parameters for five FS neurons (with a frequency in the 13-27 Hz range) split onto two ASICs on two different boards and then we measured the raster plot, as shown in Figure 4.16.

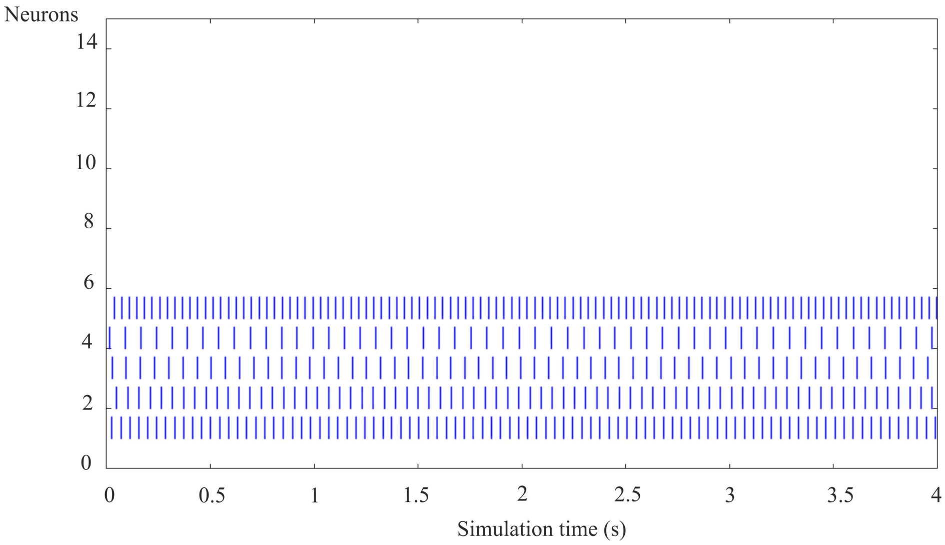


**Figure 4.16** – Raster plot of a 5-FS neuron network spread across two boards with a frequency in the 13-27 Hz range.

The system detects the neuronal activity, as we can see on the raster plot (Figure 4.16) generated offline by the software layer developed for a simulation time of four seconds.

### 4.4.2 Synaptic tuning

As previously explained, we are obliged to carry out the synaptic tuning while considering the postsynaptic membrane potential in a network of only two neurons, in which the postsynaptic neuron is at resting potential. Due to the "virtual connection", we can directly send a pulse to the postsynaptic neuron (which signifies the presence of a presynaptic spike) generated by the FPGA_DB. As indicated by the designer of the Galway chip [Bornat, 2006], we can transmit a maximum pulse of 60 $\mu$s to drive the synaptic block. This pulse and the conductance value control the strength of the synaptic connection. There are three parameters that we have to tune: the conductance value, the time constant and the reversal potential for the synapses. This procedure is done manually, while taking the potential contribution of the postsynaptic membrane into account.

Considering that the clock frequency of the FPGA_DB is 100 MHz, and the pulse of 10 ns is modulated with 8-bit resolution by the STDP processor (designed in a previous thesis [Belhadj, 2010]), we need to tune the synaptic block with a pulse from 0 up to 2.56 $\mu$s. We chose a pulse of 1.28 $\mu$s (the middle value) to tune the AMPA and GABA$_A$ synapses using biological data, as explained in section 4.1. This means that if we calibrate the synaptic conductance value sending a pulse of 1.28 $\mu$s, the modulation of this pulse (through the STDP) must change the strength of the synaptic connection. In particular, for pulse values greater than 1.28 $\mu$s, the conductance value has to increase; conversely, this value must decrease when the pulse values are less than 1.28 $\mu$s.

Due to the hardware scaling factor between the biological data and the hardware data (as explained in Chapter 3), we need to see a contribution of 15 mV for AMPA and -5 mv for GABA$_A$ in the postsynaptic neuron membrane potential. Using this value as a reference, we manually tuned the AMPA and GABA$_A$ synapses (the reversal potential values and the time constant parameter values are presented in section 4.1).

During this phase, we noticed that is not possible, as previously mentioned (section 4.1), to work with the biological constraints. This means that in the design phase for the Galway chip, a good factor scale between the biological and the hardware synaptic conductance value, which has to be in the order of one thousand, was not considered. Furthermore, it is not possible to do a simulation with plasticity (STDP) because the synaptic block does not respond to the width-modulated pulse (8-bit) generated by the STDP processor (ranging from 0 up to 2.56 $\mu$s.).

Thus, we decided to increase the pulse value and the conductance value for the AMPA and GABA$_A$ synapses. The results for the AMPA synapse are shown in Figure 4.17.

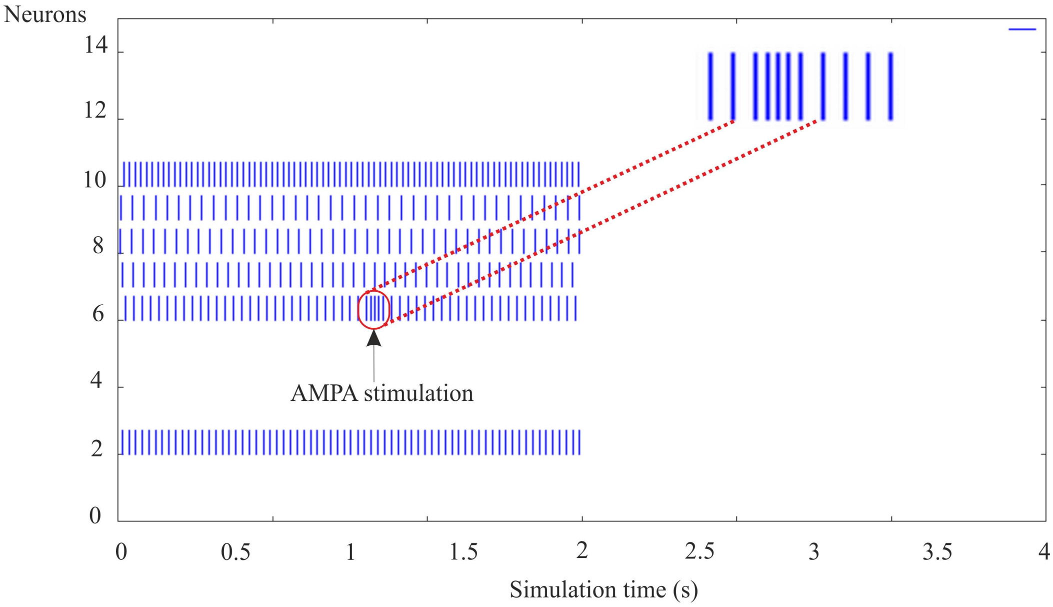


**Figure 4.17** – Raster plot of a 6-FS neuron network spread across two boards with a frequency in the 13-54 Hz range. Neuron number 6 is connected to one AMPA synapse that receives a pulse of 60 $\mu$s (presynaptic spike) after about 1 second of simulation.

We sent the parameters for the six FS neurons (with a frequency in the 13-54 Hz range) split onto two ASICs on two different boards, while taking the synapse parameters without biological meaning into account. The simulation was carried out for 2 seconds. During the simulation, after about 1 second, a pulse of 60 $\mu$s[6] was transferred to the AMPA synaptic block of FS neuron number 6 through the FPGA_DB. As shown in Figure 4.17, the system detects all of the neuron activity and the AMPA excitation is also detected on neuron 6, where we noticed an increase in the frequency (due to the excitatory synapse) after the transmission of the presynaptic spike pulse.

As for the GABA$_A$ synapse, we performed the same type of simulation as shown in Figure 4.18. As in the previous case, the system detects the activity.

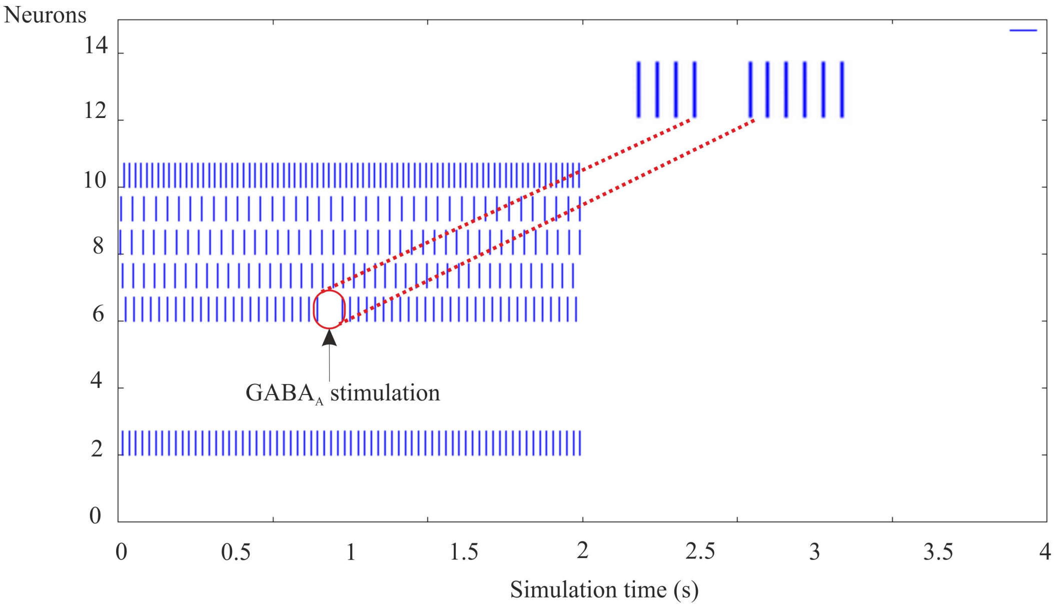


**Figure 4.18** – Raster plot of a 6-FS neuron network spread across two boards with a frequency in the 13-54 Hz range. Neuron number 6 is connected to one GABA$_A$ synapse that receives a pulse of 60 $\mu$s (presynaptic spike) before 1 second of simulation.

In the case of neuron 6, we noticed an inhibition of the activity (due to the inhibitory synapse) after the transmission of the presynaptic spike pulse. The simulations suggest that it is necessary to work with a strong synapse (increasing the conductance value) to produce some results. In this case where the synaptic current is stronger ($\mu$A) than the biological case ($\rho$A), it is possible to record the synaptic current from the ASIC; however, this is not useful for our purposes. In any case, it is not possible to conduct the plasticity

[6] This value is not the desired value for the 8-bit modulation.

(STDP) simulation, as previously explained, due to the problem on the synaptic block. As the digital platform is fully operating, it is therefore necessary to design new ASICs to complete this work. This aspect will be discussed in the final conclusions.

## 4.5 Conclusion

In this chapter, we presented our mixed hardware and software platform that is specifically designed to simulate spiking neural networks using conductance-based models of neurons. The rationale for this development is to provide a tool for computational neuroscience that ensures real-time processing and a bi-directional connection with the biological element. Thus, the main goal is to have a hardware platform that can communicate with the biological neural network. The platform, which was previously designed by Belhadj [2010], was able to perform simulations with digital spikes directly generated in the platform. The platform was missing an interface between the platform and the analog neuron, the synapses were not tuned with the biological values, and we needed a software to manage the configuration phase in the host computer.

To solve these problems, we started by introducing the main abstractions used at the neuron and network level, and then we described the features and basic functions of the PAX simulator. We focused on our contribution to the development of this mixed hardware-software platform for the real-time simulations of spiking neural networks using software tools and a VHDL driver for the neuron configuration in the platform. In particular, the system detects the neuronal activity, as can be seen on the raster plot generated by the software layer that was developed. Therefore, to address our main goal, we propose a method to calibrate the synapses in our system. However, during this phase, we noticed that is not possible to work with the biological constraints (due to a bad factor scale between the biological value and the hardware synaptic conductance value in the Galway chip). Furthermore, it is not possible to perform a pertinent simulation using the plasticity rule, because the synaptic block does not respond to the width-modulated pulse (8-bit) generated by the STDP processor (ranging from 0 up to 2.56 $\mu$s.). As a result, we also showed two raster plots of six FS neurons spread across two boards, in which neuron number 6 has been connected with an AMPA synapse (and in a second case, with only a GABA$_A$ synapse) that receives a pulse of 60 $\mu$s (presynaptic spike). The performed hardware simulations demonstrate that the hardware platform is fully operational in the digital layer. We will discuss the implications of this work in the conclusion of this thesis.

# Conclusions and implications

The understanding of neuronal circuits is a great challenge which involves a large number of researchers from different disciplinary fields. All fields, including neuromorphic engineering, contribute to this understanding. The goal of this thesis is to improve the artificial-biological hybrid technique. We simulate the four most prominent biological cells present in the neocortex in conductance-based analog neuromimetic integrated circuits using electrophysiological recordings as a reference. Our results show that our system is able to reproduce the main features of the four common classes of cortical cells. Furthermore, we contribute to the development of a mixed hardware-software platform for simulating spiking neural networks in real time. At the end of this manuscript, we will summarize the work that we carried out. In this general conclusion, we will discuss the context in which this work it is placed and then we will present the work and the results obtained in detail. We will conclude by discussing the implications of this work.

**Work context**

This work was carried out within the framework of the European project FACETS-ITN (FP7-PEOPLEITN-2008-237955; 2009-2014). This project was undertaken in a multidisciplinary environment in which neuroscientists provide biological measurements to computational neuroscientists who then propose a simulation model to study the single cell or neural network dynamics. Finally, experts in neuromorphic engineering use the neuronal properties to build neurally inspired computing hardware.

In parallel with Prof. K. Meier's group at the University of Heidelberg in Germany, our group is involved in the design, simulation and implementation of neuromorphic systems. Two design approaches are used. The Heidelberg team designs VLSI circuits to simulate large accelerated neural networks. They plan to integrate 1 million integrate-and-fire neurons. This large network is simulated at the expense of the accuracy of the models used. The simulation time scale is about $10^4$ times the biological time scale [Schemmel et al., 2007].

On the other hand, our team designs analog circuits to simulate neurons according to the Hodgkin and Huxley formalism in order to model several types of neuron complexes operating on a biological time scale. The goal is to build an artificial spiking neural network (in the order of hundreds of neurons) to achieve a high degree of realism to extend the hybrid technique, also called the "dynamic-clamp" technique [Le Masson et al., 1995], to micro-electrode arrays [Bontorin et al., 2007]. This technique consists of connecting artificial and biological neurons to create a real-time loop [Sorensen et al., 2004].

The work presented in this thesis is the continuation of work done by the group. The analog VLSI circuits that represent the artificial neurons are the result of the work of Alvado [2003], Saïghi [2004] and Bornat [2006]. Several ASICs have been designed and manufactured for this purpose. The Galway circuit is the latest version of ASICs designed by our group that integrates five configurable neurons. Neuromorphic systems were designed from these circuits [Belhadj, 2010]. The main objectives of this thesis are:

- to simulate the four most prominent biological cells present in the neocortex (Fast-Spiking, Regular-Spiking, Intrinsically Bursting and Low-Threshold Spiking neurons) in conductance-based analog neuromimetic integrated circuits using electrophysiological recordings as a reference;
- to contribute toward the development of a mixed hardware-software platform in order to simulate a spiking neural network (SNN) in real time.

### Results of this work

To address the first objective of this thesis, we performed the following work:

- we proposed the appropriate parameter sets of the FS, RS, IB, and LTS neurons for a simplified version of the HH formalism that can be implemented in our analog neuromimetic chip. By comparing the software simulations of our simplified neuron model with the HH models for cortical neuron cells, we show that the simplified models can reproduce the main firing features of cortical cell types (see Chapter 2);
- we described the steps used to extract the parameters from the HH model, using biological data, to the one implemented in the VLSI neurons, validating the simplified model through a bifurcation analysis (see Chapter 2).

In particular, using a bifurcation analysis, we showed numerical simulations of the Hopf bifurcation which is characteristic of Class 2 excitability in the HH model. We compared these results with the HH model, showing that our model shares the dynamics with the full HH model. We published this methodology to validate our simplified model by showing that our silicon neuron has nonlinear dynamical phenomenon similar to a certain class of biological neurons [Grassia et al., 2012].

Therefore, we show (see Chapter 3) that the circuits can reproduce the main firing features of cortical cell types by comparing them with experimental electrophysiological data of these cells. We performed the following work:

- we designed the interface in VHDL to send the parameters to the silicon neurons;
- due to process variations and device mismatch in the analog chips, we proposed a fully customizable fitting method, in voltage-clamp mode, to tune our neuromimetic integrated circuits. This optimization method, based on the DE algorithm that we pubblished [Buhry et al., 2011], is an alternative to the estimation methods associated with voltage-clamp measurements. In any event, we observed a large discrepancy for all of the parameters, which confirms the necessity of the tuning step;
- we designed the software interface to manage the fully customizable fitting method, in voltage-clamp mode, to tune our neuromimetic integrated circuits;
- finally, we tested the parameters obtained by comparing the behavior of the membrane voltage in the recordings of biological cells with the membrane voltages simulated with our chip. This comparison is possible due to the translation rules between biological and hardware neurons based on the chip characteristics. We directly compared the behavior of our chip with biological recordings. Our results [Grassia et al., 2011] show that our system is able to reproduce the main features of the four common classes of cortical cells.

Thus, we addressed the first objective of this thesis, which was to propose an optimization method to tune the ASIC parameters, to use a bifurcation analysis to validate the parameter extraction steps for our approximated model and then to compare the results with biological recordings. We can assert that these ASICs can be used to form the core of a simulation platform, designed to simulate neural networks in biologically relevant configurations.

Therefore, regarding the second objective of this thesis, we presented (see Chapter 4) the spiking neural networks (SNN) simulator developed by our team that is called PAX

(Plasticity Algorithm for Computing System) that hosts the chip. Hence, we described the work done for the silicon neuron integration on PAX and the synaptic tuning. We performed the following work:

- we designed the interface in VHDL for the neuron configuration in the platform that manages: 1) the ASICs topology parameters for the specification of neuron type and synapse behavior (digital parameters), 2) the ASIC data parameters for the neurons and synapses (analog parameters), 3) the ASIC stimulation parameters for an external stimulation current (analog);
- we designed the software interface to convert the data that we have to send to the platform in a frame format for the digital bus. This procedure defines the configuration phase of the system;
- we proposed a method for the synaptic tuning of the AMPA and $\text{GABA}_A$ synapses in the PAX system.

As for the results and validation of the work performed in the PAX system, we presented hardware simulations that demonstrate that the hardware platform is fully operational in the digital layer. Therefore, we have achieved the objectives assigned. Furthermore, as main goal of the work of our group, we observed a problem in the hardware multisynapse block, on the ASICs, that does not allow a pertinent simulation of biologically relevant SNNs to be performed using STDP.

**Implications**

Biological neural systems are very complex compared to models simulated with circuits. As we have seen, the abstraction of details is necessary. Our group chose to make biologically inspired neuromorphic systems. To this end, we chose silicon neuron and plasticity models to satisfy the desired degree of realism. Thus, a mixed hardware and software platform for the emulation of cortical neuron models and then SNN (composed of hundreds of silicon neurons) was proposed. As in many other fields of microelectronics, a mixed implementation offers both the advantages and disadvantages of both solutions: analog circuits have a higher integration density, and digital platforms have better programmability. In our case, taking into account the problems observed for the synaptic tuning, a newly designed analog ASIC is thus necessary to replace the one used in the PAX system. Moreover, we have to take advantage of this design to think about a standard unified communication protocol for bio-inspired and bio-mimetic information processing systems and neurophysiological devices. Many research groups have developed their own bio-inspired systems which cannot be connected to each other. These systems include communication buses to manage the neural events, i.e. action potentials, also called spikes. These solutions are not compatible with each other. We launched this work with Matthieu Ambroise, who started his thesis in October 2012.

We cannot finish this thesis without talking about the circuit calibration and commenting on the architecture chosen for the real-time simulations of small biologically inspired neural networks. Due to a mismatch and process variations in the manufacturing of the silicon neuron, it was necessary to use an automated tuning to calibrate the neurons one by one. We proposed a method based on heuristic algorithms [Buhry et al., 2011]. Optimizations are still needed to reduce the amount of time needed to use these algorithms to extract the appropriate set of parameters. Therefore, the synaptic tuning should be addressed in the same way. This means that if a new analog ASIC is designed, it will be necessary to calibrate all of the neurons and then the synapses again. This platform was designed to simulate SNNs in biological real time using conductance-based models of neurons and synaptic connections in order to develop a hybrid technique that connects silicon and biological neurons in real time. Currently, it offers the possibility of dynamically tuning the neuron model parameters (with a biological significance).

Given the complex nature of this work, the question is raised of whether or not to insist on this type of architecture or to change to a completely digital replacement of the "analog neuron" with a "digital neuron". Therefore, it becomes necessary to study the feasibility of using digital neurons in a biological time scale (with a high degree of "realism"), at the same time that a new analog silicon neuron is designed, while taking the same methodologies introduced in this thesis to validate the neuronal activity into account. In particular, the firing rate and a bifurcation analysis can be of help in the methodology used to validate the results.

What are the possible applications of a system of this type? Multiple applications from different research areas can exploit these systems: 1) a neural network simulator that operates with a high degree of realism is the right tool to perform experiments that are not feasible in biology; 2) a neuromorphic system operating in real time may be biologically interfaced with live neural networks in an open or closed loop [Bontorin et al., 2007]. The system can play the role of both stimulator synapses live and a device for acquiring cell responses in culture in-vivo . Our team is an expert in both applications. We will be able to take this work further thanks to the following projects: BRAIN-BOW, GIHON and HySNET. The BRAIN-BOW project will provide the knowledge to create a new class of neuroprosthetics that can be used to treat diseases in which a portion of the brain tissue is damaged (e.g. lesion). The ultimate goal of the project is to connect in vitro neuronal assemblies to an artificial system (a neuromorphic chip) that will attempt to restore the lost neuronal functionality, with the long-term perspective to be implanted in humans affected by invalidating brain diseases. The goal of the GIHON project is to design a universal, bio-inspired bus communication protocol. The goal of the HySNET project is to investigate the suggestion that the effects of noise on neural computation can be simulated and studied efficiently by using stochastic neuronal models in Very Large Scale Integration (VLSI) systems and by connecting the VLSI neurons to biological neurons to form hybrid networks.

# Publications

## Journal articles

Grassia, F., Lévi, T., Saïghi, S., and Kohno, T. (2012). Bifurcation analysis in a silicon neuron. *Artificial Life and Robotics*, 17:53-58.

Grassia, F., Buhry, L., Lévi, T., Tomas, J., Destexhe, A., and Saïghi, S. (2011). Tunable neuromimetic integrated system for emulating cortical neuron models. *Frontiers in Neuroscience*, 5(134).

Buhry, L., Grassia, F., Giremus, A., Grivel, E., Renaud, S., and Saïghi, S. (2011). Automated parameter estimation of the hodgkin-huxley model using the differential evolution algorithm: Application to neuromimetic analog integrated circuits. *Neural Computation*, 23(10):2599-2625.

## Conferences

Grassia, F., Lévi, T., Saïghi, S., and Kohno, T. (2012). Bifurcation analysis in a silicon neuron. *16th International Symposium on Artificial Life and Robotics, Beppu, Japan, 19-21 January 2012.*

Grassia, F., Lévi, T., Tomas, J., Renaud, S., and Saïghi, S (2011). A Neuromimetic Spiking Neural Network for Simulating Cortical Circuits. *45st Annual Conference on Information Sciences and Systems, CISS 2011, The John Hopkins University, Baltimore, MD, USA, 23-25 March 2011.*

# References

Abbott, L. F. (1999). Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Res Bull*, 50(5-6):303–304. 17

Alvado, L. (2003). *Neurones artificiels sur Silicium: une évolution vers le réseau*. PhD thesis, University Bordeaux 1 – n. 2674. 106

Arbib, M. A. (1995). *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge, MA, USA, 1st edition. 1

Belhadj, B. (2010). *Systèmes neuromorphiques temps réel : contribution à l'intégration de réseaux de neurones biologiquement réalistes avec fonctions de plasticité*. PhD thesis, University Bordeaux 1 – n. 6357. xx, xxi, xxii, 24, 27, 63, 81, 86, 90, 92, 100, 103, 106, 130

Bi, G. and Poo, M. (2001). Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual review of neuroscience*, 24(1):139–166. 24

Binczak, S., Jacquir, S., Bilbault, J.-M., Kazantsev, V. B., and Nekorkin, V. I. (2006). Experimental study of electrical fitzhugh-nagumo neurons with modified excitability. *Neural Networks*, 19(5):684–693. 25

Bontorin, G., Renaud, S., Garenne, A., Alvado, L., Le Masson, G., and Tomas, J. (2007). A real-time closed-loop setup for hybrid neural networks. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society IEEE Engineering in Medicine and Biology Society Conference*, 2007:3004–7. 13, 106, 109

Bornat, Y. (2006). *Réseaux de neurones sur silicium : une approche mixte, analogique / numérique, pour l'étude des phénomènes d'adaptation, d'apprentissage et de plasticité*. PhD thesis, University Bordeaux 1 – n. 3261. xx, 20, 23, 31, 39, 59, 64, 83, 85, 100, 106, 123

Bornat, Y., Tomas, J., Saighi, S., and Renaud, S. (2005). BiCMOS Analog Integrated Circuits for Embedded Spiking Neural Networks. In *Proeeding of XX Conference on Design of Circuits and Integrated Systems, DCIS 2005*, page 1, Lisbon, Portugal. NC. 25

Brette, R., Rudolph, M., Carnevale, T., Hines, M., Beeman, D., Bower, J., Diesmann, M., Morrison, A., Goodman, P., Harris, F., Zirpe, M., NatschlÃ¤ger, T., Pecevski, D., Ermentrout, B., Djurfeldt, M., Lansner, A., Rochel, O., Vieville, T., Muller, E., Davison, A., El Boustani, S., and Destexhe, A. (2007). Simulation of networks of spiking neurons: A review of tools and strategies. *Journal of Computational Neuroscience*, 23:349–398. xviii, 24

Buhry, L. (2010). *Estimation de paramètres de modèles de neurones biologiques sur une plate-forme de SNN (Spiking Neural Network) implantés "in silico"*. PhD thesis, University Bordeaux 1 – n. 4057. xx

Buhry, L., Grassia, F., Giremus, A., Grivel, E., Renaud, S., and Saïghi, S. (2011). Automated parameter estimation of the hodgkin-huxley model using the differential evolution algorithm: Application to neuromimetic analog integrated circuits. *Neural Computation*, 23(10):2599–2625. xx, xxii, 20, 21, 59, 67, 70, 79, 107, 109

Buhry, L., Saïghi, S., Giremus, A., Grivel, E., and Renaud, S. (2008). Parameter estimation of the Hodgkin–Huxley model using metaheuristics: application to neuromimetic analog integrated circuits. In *Proceedings of IEEE International Conference on Biomedical circuits and Systems*, pages 173–176, Baltimore USA. 67

Casti, A., Omurtag, A., Sornborger, A., Kaplan., E., Knight, B., Victor, J., and Sirovich, L. (2002). A population study of integrate-and-fire-or-burst neurons. *Neural Computation*, 14(5):957–986. 18

Churchland, P. S., Koch, C., and Sejnowski, T. J. (1993). What is computational neuroscience? In Schwartz, E. L., editor, *Computational neuroscience*, pages 46–55. MIT Press, Cambridge, MA, USA. xvii

Cole, K. S. (1949). Dynamic electrical characteristics of the squid axon membrane. *Arch. Sci. Physiol.*, 3:253–258. 21

Connors, B. W. and Gutnick, M. J. (1990). Intrinsic firing patterns of diverse neocortical neurons. *Trends in neurosciences*, 13(3):99–104. 11, 13, 48, 71, 76, 77

Cooper, D. C. (2011). *Introduction to neuroscience I*. Donald C. Cooper Ph.D. 2

Daouzli, A. (2009). *Systémes neuromorphiques: Etude et implantation de fonctions d'apprentissage et de plasticité*. PhD thesis, University Bordeaux 1 – n. 3806. xx

De la Peña, E. and Geijo-Barrientos, E. (1996). Laminar localization, morphology, and physiological properties of pyramidal neurons that have the low-threshold calcium current in the guinea-pig medial frontal cortex. *The Journal of Neuroscience*, 16(17):5301–5311. 54

Destexhe, A. (2001). Simplified models of neocortical pyramidal cells preserving somatodendritic voltage attenuation. *Neurocomputing*, 38:167–173. 54

Destexhe, A. and Huguenard, J. (2000). Which formalism to use for modeling voltage-dependent conductances? In DeSchutter, E., editor, *Computational Neuroscience: Realistic Modeling for Experimentalists*, pages 129–157. CRC Press, Boca Raton FL. 32

Destexhe, A., Mainen, Z. F., and Sejnowski, T. J. (1994). An efficient method for computing synaptic conductances based on a kinetic model of receptor binding. *Neural Computation*, 6(1):14–18. 23, 85

Destexhe, A., Mainen, Z. F., and Sejnowski, T. J. (1998a). Kinetic models of synaptic transmission. *Methods in Neuronal Modeling*, pages 1–25. 23, 83, 86

Destexhe, A., Neubig, M., Ulrich, D., and Huguenard, J. (1998b). Dendritic Low-Threshold Calcium Currents in Thalamic Relay Cells. *The Journal of Neuroscience*, 18(10):3574–3588. 39

Destexhe, A., Rudolph, M., and Pare, D. (2003). The high-conductance state of neocortical neurons in vivo. *Nature Reviews Neuroscience*, 4(9):739–751. 52

Farquhar, E. and Hasler, P. (2005). A bio-physically inspired silicon neuron. *IEEE Transactions on circuits and systems*, 52(3):477–488. 25

Feiner, A.-S. and McEvoy, A. J. (1994). The nernst equation. *Journal of Chemical Education*, 71(6):493. 4

Feoktistov, V. and Janaqi, S. (2004). Generalization of the strategies in differential evolution. In *18th International Parallel and Distributed Processing Symposium, 26-30 April 2004, Santa Fe, New Mexico, USA*. IEEE Computer Society. 67

Fieres, J., Schemmel, J., and Meier, K. (2006). Training convolutional networks of threshold neurons suited for low-power hardware implementation. In *Proceedings of the International Joint Conference on Neural Networks, part of the IEEE World Congress on Computational Intelligence, WCCI 2006, Vancouver, BC, Canada, 16-21 July 2006*, pages 21–28. IEEE. 25

FitzHugh, R. (1955). Mathematical models of threshold phenomena in the nerve membrane. *The Bulletin of Mathematical Biophysics*, 17:257–278. 15

FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1:445–466. 15

Geit, W. V., Schutter, E. D., and Achard, P. (2008). Automated neuron model optimization techniques: a review. *Biological Cybernetics*, 99:241–251. 21, 67

Gerstner, W. and Kistler, W. M. (2002). *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press. 17, 18

Gibson, J. R., Beierlein, M., and Connors, B. W. (1999). Two networks of electrically coupled inhibitory neurons in neocortex. *Nature*, 402(6757):75–79. 11, 71, 77

Glackin, B. P., McGinnity, T. M., Maguire, L. P., Wu, Q., and Belatreche, A. (2005). A novel approach for the implementation of large scale spiking neural networks on fpga hardware. In Cabestany, J., Prieto, A., and Hernàndez, F. S., editors, *Computational Intelligence and Bioinspired Systems, 8th International Work-Conference on Artificial Neural Networks*, volume 3512 of *Lecture Notes in Computer Science*, pages 552–563. Springer. 25, 26

Graas, E. L., Brown, E. A., and Lee, R. H. (2004). An fpga-based approach to high-speed simulation of conductance-based neuron models. *Neuroinformatics*, 2(4):417–435. 25, 26

Grassia, F., Buhry, L., Lévi, T., Tomas, J., Destexhe, A., and Saïghi, S. (2011). Tunable neuromimetic integrated system for emulating cortical neuron models. *Frontiers in Neuroscience*, 5(134). xxi, xxii, 70, 107

Grassia, F., Lévi, T., Saïghi, S., and Kohno, T. (2012). Bifurcation analysis in a silicon neuron. *Artificial Life and Robotics*, 17:53–58. xxi, xxii, 47, 107

Hansel, D., Mato, G., and Meunier, C. (1993). Phase dynamics for weakly coupled Hodgkin-Huxley neurons. *Europhysics Letters*, 23:367–372. 31, 37, 39, 40, 41, 45

Hasler, P. E., Koziol, S., Farquhar, E., and Basu, A. (2007). Transistor channel dendrites implementing hmm classifiers. In *International Symposium on Circuits and Systems*, pages 3359–3362. 25

Hassard, B. B. (1978). Bifurcation of periodic solutions of hodgkin-huxley model for the squid giant axon. *Journal of Theoretical Biology*, 71(3):401–420. 45

Hassard, B. D., Kazarinoff, N. D., and Wan, Y.-H. (1981). *Theory and applications of Hopf bifurcation / B.D. Hassard, N.D. Kazarinoff and Y.-H. Wan*. Cambridge University Press, Cambridge ; New York :. 44, 45

Hille, B. (1991). *Ionic Channels of Excitable Membranes*. Sinauer Associates, 2 sub edition. 36

Hindmarsh, J. L. and Rose, R. M. (1984). A model of neuronal bursting using three coupled first order differential equations. In *Proceedings of the Royal Society, B*, volume 221, pages 87–102. 16

Hines, M. L. and Carnevale, N. T. (1997). The neuron simulation environment. *Neural Computation*, 9(6):1179–1209. 24, 86

Hodgkin, A. L. (1948). The local electric changes associated with repetitive action in a non-medullated axon. *The Journal of physiology*, 107(2):165–181. 42, 44

Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500–544. 13, 21, 31, 33, 35

Indiveri, G., Chicca, E., and Douglas, R. (2006). A VLSI array of low-power spiking neurons and bistable synapses with spike–timing dependent plasticity. *IEEE Transactions on Neural Networks*, 17(1):211–221. 25

Indiveri, G. and Fusi, S. (2007). Spike-based learning in VLSI networks of integrate-and-fire neurons. In *International Symposium on Circuits and Systems*, pages 3371–3374. IEEE. 25

Izhikevich, E. M. (2000). Neural Excitability, Spiking and Bursting. *International Journal of Bifurcation and Chaos*, 10(6):1171–1266. 13, 42

Izhikevich, E. M. (2001). Resonate-and-fire neurons. *Neural Networks*, 14(6-7):883–894. 19

Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572. xvii, 19, 20

Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks*, 15(5):1063–1070. 11, 13

Izhikevich, E. M. and Edelman, G. M. (2008). Large-scale model of mammalian thalamocortical systems. *Proceedings of the National Academy of Sciences*, 105(9):3593–3598. xvii

Jung, R., Brauer, E. J., and Abbas, J. J. (2001). Real-time interaction between a neuromorphic electronic circuit and the spinal cord. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 9(3):319–326. 25

Kandel, E. R., Schwartz, J. H., and Jessell, T. M. (2000). *Principles of Neural Science*. McGraw-Hill Medical, 4th edition. 2, 3, 6, 37

Karaboga, D. and S.Okdem (2004). A simple and global optimization algorithm for engineering problems: Differential evolution algorithm. *Computer Journal of Turk Electronic Engineering*, 12(1). 68

Keren, N., Peled, N., and Korngreen, A. (2005). Constraining compartmental models using multiple voltage-recordings and genetic algorithms. *J. Neurophysiol*, 94:3730–3742. 21

KIN450 (2009). Synaptic transmission. `http://kin450-neurophysiology.wikispaces.com/Synaptic+Transmission`. 10

Koene, R. and Hasselmo, M. (2005). An Integrate-and-fire Model of Prefrontal Cortex Neuronal Activity during Performance of Goal-directed Decision Making. *Cerebral Cortex*, 15(12):1964–1981. 18

Le Masson, G., Le Masson, S., and Moulins, M. (1995). From conductances to neural network properties: analysis of simple circuits using the hybrid network method. *Progress in Biophysics and Molecular Biology*, 64(2-3):201–220. 13, 106

Le Masson, G., Renaud-Le Masson, S., Debay, D., and Bal, T. (2002). Feedback inhibition controls spike transfer in hybrid thalamic circuits. *Nature*, 417(6891):854–858. xviii, 25, 27, 82

Lecar, H. (2007). Morris-lecar model. *Scholarpedia*, 2(8):1333. 16

Lester, D. R. and Furber, S. (2011). Spinnaker: Distributed computer engineering for neuromorphics. In Apolloni, B., Bassis, S., Esposito, A., and Morabito, F. C., editors, *WIRN*, volume 234 of *Frontiers in Artificial Intelligence and Applications*, pages 324–331. IOS Press. 26

Liu, S.-C. and Douglas, R. (2004). Temporal coding in a silicon network of integrate-and-fire neurons. *Transactions on Neural Networks*, 15(5):1305–1314. 25

Llinás, R. R. (1988). The intrinsic electrophysiological properties of mammalian neurons: insights into central nervous system function. *Science*, 242(4886):1654–1664. 32

Mahowald, M. and Douglas, R. (1991). A silicon neuron. *Nature*, 354:515 – 518. xviii, 25

Millner, S., Hartel, A., Schemmel, J., and Meier, K. (2012). Towards biologically realistic multi-compartment neuron model emulation in analog vlsi. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learnin (ESANN)*, Bruges, BELGIUM. 26

Misra, J. and Saha, I. (2010). Artificial neural networks in hardware: A survey of two decades of progress. *Neurocomputing*, 74(1-3):239–255. xviii

Morris, C. and Lecar, H. (1981). Voltage oscillations in the barnacle giant muscle fiber. *Biophysical Journal*, 35:193–213. 15

Nagumo, J., Arimoto, S., and Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. In *Proceedings of Institute of Radio Engineers*, volume 50, pages 2061–2070. IEEE. 15

Neftci, E., Chicca, E., Indiveri, G., and Douglas, R. J. (2011). A systematic method for configuring vlsi networks of spiking neurons. *Neural Computation*, 23(10):2457–2497. 21

Pospischil, M., Toledo-Rodriguez, M., Monier, C., Piwkowska, Z., Bal, T., Frégnac, Y., Markram, H., and Destexhe, A. (2008). Minimal hodgkin–huxley type models for different classes of cortical and thalamic neurons. *Biological Cybernetics*, 99:427–441. xxi, xxii, 20, 21, 31, 37, 38, 47, 48, 53, 55, 57, 70, 71, 72, 74, 75, 76, 77, 78

Purves, D., Augustine, G., Fitzpatrick, D., Hall, W., and Lamantia, A. (2008). *Neuroscience, + Neurons in Action*. Sinauer Associates Incorporated. 5

Rangan, A. and Cai, D. (2007). Fast numerical methods for simulating large-scale integrate-and-fire neuronal networks. *Journal of Computational Neuroscience*, 22(1):81–100. 18

Rasche, C. and Douglas, R. (2000). An improved silicon neuron. *Analog Integrated Circuits and Signal Processing*, 23(3):227–236. 56

Renaud, S., Masson, G. L., Alvado, L., Saïghi, S., and Tomas, J. (2004). A neural simulation system based on biologically realistic electronic neurons. *Information Sciences*, 161(1-2):57–69. 25

Renaud, S., Tomas, J., Bornat, Y., Daouzli, A., and Saïghi, S. (2007). Neuromimetic ICs with analog cores: an alternative for simulating spiking neural networks. In *International Symposium on Circuits and Systems*, pages 3355–3358, New-Orleans, USA. IEEE. 25

Renaud, S., Tomas, J., Lewis, N., Bornat, Y., Daouzli, A., Rudolph, M., Destexhe, A., and Saïghi, S. (2010). Pax: A mixed hardware/software simulation platform for spiking neural networks. *Neural Networks*, 23(7):905–916. 25, 26, 27

Reuveni, I., Friedman, A., Amitai, Y., and Gutnick, M. (1993). Stepwise repolarization from $ca^{2+}$ plateaus in neocortical pyramidal cells: evidence for nonhomogeneous distribution of hva $ca^{2+}$ channels in dendrites. *The Journal of Neuroscience*, 13(11):4609–21. 38

Rinzel, J. and Ermentrout, G. B. (1989). Analysis of neural excitability and oscillations. In Koch, C. and Segev, I., editors, *Methods in neuronal modeling*. MIT Press, Cambridge, MA. 42

Rossant, C., Goodman, D. F. M., Platkiewicz, J., and Brette, R. (2010). Automatic fitting of spiking neuron models to electrophysiological recordings. *Frontiers in Neuroinformatics*, 4(2). 21

Saïghi, S. (2004). *Circuits et systèmes de modélisation analogique de réseaux de neurones biologiques: application au développement d'outils pour les neurosciences computationnelles*. PhD thesis, University Bordeaux 1 – n. 2891. 60, 106

Saïghi, S., Bornat, Y., Tomas, J., Masson, G. L., and Renaud, S. (2011). A library of analog operators based on the hodgkin-huxley formalism for the design of tunable, real-time, silicon neurons. *IEEE Transaction on Biomedical Circuits and Systems*, 5(1):3–19. 13, 40, 60

Saïghi, S., Buhry, L., Bornat, Y., Kaoua, G. N., Tomas, J., and Renaud, S. (2008). Adjusting the neurons models in neuromimetic ics using the voltage-clamp technique. In *International Symposium on Circuits and Systems, 18-21 May 2008, Seattle, Washington, USA*, pages 1564–1567. IEEE. 21, 59

Schemmel, J., Bruderle, D., Meier, K., and Ostendorf, B. (2007). Modeling synaptic plasticity within networks of highly accelerated I&F neurons. In *International Symposium on Circuits and Systems*, pages 3367–3370. IEEE. 86, 106

Schemmel, J., Fieres, J., and Meier, K. (2008). Wafer-scale integration of analog neural networks. In *Proceedings of IEEE International Conference on Neural Networks ( IJCNN)*, pages 431–438, Hong Kong, CHINA. 25

Shin, J. and Koch, C. (1999). Dynamic range and sensitivity adaptation in a silicon spiking neuron. *IEEE Transactions on Neural Networks*, 10(5):1232–1238. 56

Simoni, M. F., Cymbalyuk, G. S., Sorensen, M. E., Calabrese, R. L., and DeWeerth, S. P. (2004). A multiconductance silicon neuron with biologically matched dynamics. *IEEE Transactions on Biomedical Engineering*, 51(2):342–354. 56

Sjöström, P., Rancz, E., Roth, A., and Häusser, M. (2008). Dendritic excitability and synaptic plasticity. *Physiological Reviews*, 88:769–840. 24

Sorensen, M., DeWeerth, S., Cymbalyuk, G., and Calabrese, R. L. (2004). Using a hybrid neural system to reveal regulation of neuronal network activity by an intrinsic current. *The Journal of Neuroscience*, 24(23):5427–5438. 25, 106

Storn, R. and Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359. 67, 68

Traub, R. D. and Miles, R. (1991). *Neuronal Networks of the Hippocampus*. Cambridge University Press, New York, NY, USA. 51

Vanier, M. and Bower, J. (10 September 1999). A comparative survey of automated parameter-search methods for compartmental neural models. *Journal of Computational Neuroscience*, 7:149–171(23). 21

Vogelstein, R. J., Mallik, U., and Cauwenberghs, G. (2004). Silicon spike-based synaptic array and address-event transceiver. In *International Symposium on Circuits and Systems*, pages 385–388. 25

Yamada, W. M., Koch, C., and Adams, P. R. (1989). Methods in neuronal modeling. chapter Multiple channels and calcium dynamics, pages 97–133. MIT Press, Cambridge, MA, USA. 38, 51

Yu, T. and Cauwenberghs, G. (2010). Analog vlsi biophysical neurons and synapses with programmable membrane channel kinetics. *IEEE Transaction on Biomedical Circuits and Systems*, 4(3):139–148. 56

# Appendix A

# Configuration parameters of the Galway circuit

Part of this appendix is based on the work of Bornat [2006]; however, we reproduce it in this thesis in order to better understand the VHDL driver that we have developed to send the parameters to the ASIC. The following different tables show the parameters necessary for the configuration of the circuit [Bornat, 2006]. We start with the digital parameters, which are primarily used to select which conductances are connected to the neuron's membrane. There are five silicon neurons in the Galway chip.

## A.1 Digital parameters

The digital parameters are made up of 14-bit words that are transmitted in series. The interface was shown in section 3.1.3. The first decoding is performed on the three least significant bits. These bits determine which silicon neuron will be considered for the hardware simulation, as shown in Table A.1.

| 0 | 1 | 2 | Activated unit |
|---|---|---|---|
| 0 | 0 | 0 | Neuron 0 (FS) |
| 0 | 0 | 1 | Neuron 1 (RS) |
| 0 | 1 | 0 | Neuron 2 (RS) |
| 0 | 1 | 1 | Neuron 3 (RS) |
| 1 | 0 | 0 | Neuron 4 (5 cond.) |
| 1 | 0 | 1 | Additional conductance |

**Table A.1** – Description of the destination bit.

For the additional conductance, the bits necessary for the configuration of the register are presented in Table A.2.

| Bit | Mode | if 0 | if 1 |
|---|---|---|---|
| 5 | Neuron 3 | not connected | connected |
| 6 | Neuron 4 | not connected | connected (priority) |
| 7 | Monitoring | | |
| 8 | Power rising | $m \cdot h$ | $m^2 \cdot h$ |

**Table A.2** – Register of the additional conductance.

Bit 8 selects the temporal behavior of the additional conductance that can be connected only to neuron 3 or 4.

### A.1.1 Neuron-specific parameters

The next two bits (3 and 4) for the configuration of the neuron determine which register will be changed (except for the register for the extra conductance, as explained above). The bits necessary for the configuration of the neuron are presented in Table A.3.

| 3 | 4 | Register |
|---|---|---|
| 0 | 0 | Conductances connected to the membrane |
| 0 | 1 | Conductances connected to the visualization output |
| 1 | 0 | Simulation mode |
| 1 | 1 | *not used* |

**Table A.3** – Description of the neuron register bit.

Two registers are used to select the conductances connected to the membrane and connected to the monitoring output. One register is used to select the simulation mode of the neuron. One register is used to select an extra conductance. The table A.4 shows the channel associated with each configuration bit.

Neuron 4 is the only neuron to have all of its conductances presented in Table A.4. Otherwise, the bit is not considered. Finally, the last register to be documented is the one concerning the simulation mode for a specific neuron. Table A.5 also shows how to

| Bit | conductance |
|---|---|
| 5 | Sodium |
| 6 | Potassium |
| 7 | Leak |
| 8 | Stimulation |
| 9 | Modulation |
| A | Calcium |
| B | Synapse A |
| C | Synapse B |
| D | Synapse C |

**Table A.4** – Conductances register.

settle the slow conductances.

| Bit | Mode | if 0 | if 1 |
|---|---|---|---|
| 5 | Voltage clamp | active | simulation |
| 6 | Modulation form | $m \cdot h$ | $m^2 \cdot h$ |
| 7 | Calcium form | $m \cdot h$ | $m^2 \cdot h$ |

**Table A.5** – Simulation register.

The voltage-clamp function is used to extract the parameters of the neuron; this means that the voltage membrane $V_{\text{M}}$ is fixed by an external input.

## A.2 Analog parameters

The interface to send the analog parameters was shown in section 3.1.3. Table A.6 presents the 205 analog parameters of the circuit, which enables a direct link be-tween the parameters and their corresponding memory address. They are numbered in hexadecimal form.

| Signification | Neuron | | | | | Extra | |
|---|---|---|---|---|---|---|---|
| of the parameter | 0 | 1 | 2 | 3 | 4 | cond. | |
| $Na^+$ -act- $V_{slope}$ | 00 | 1E | 45 | 6C | 93 | | |
| $Na^+$ -act- $V_{offset}$ | 01 | 1F | 46 | 6D | 94 | | |

**Table A.6** – Correspondence between the parameter number and its order number (continued on the next page)

| Signification of the parameter | Neuron | | | | | Extra cond. | |
|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | | |
| $Na^+$ -act- $\tau$ | 02 | 20 | 47 | 6E | 95 | | |
| $Na^+$ -inact- $V_{slope}$ | 03 | 21 | 48 | 6F | 96 | | |
| $Na^+$ -inact- $V_{offset}$ | 04 | 22 | 49 | 70 | 97 | | |
| $Na^+$ -inact- $\tau$ | 05 | 23 | 4A | 71 | 98 | | |
| $Na^+$ - $g_{max}$ | 06 | 24 | 4B | 72 | 99 | | |
| $Na^+$ - $V_{equi}$ | 07 | 25 | 4C | 73 | 9A | | |
| $K^+$ - $V_{slope}$ | 08 | 26 | 4D | 74 | 9B | | |
| $K^+$ - $V_{offset}$ | 09 | 27 | 4E | 75 | 9C | | |
| $K^+$ - $\tau$ | 0A | 28 | 4F | 76 | 9D | | |
| $K^+$ - $g_{max}$ | 0B | 29 | 50 | 77 | 9E | | |
| $K^+$ - $V_{equi}$ | 0C | 2A | 51 | 78 | 9F | | |
| Leak - $g_{max}$ (1/2) | 0D | 2B | 52 | 79 | A0 | | |
| Leak - $g_{max}$ (2/2) | 0E | 2C | 53 | 7A | A1 | | |
| Leak - $V_{equi}$ | 0F | 2D | 54 | 7B | A2 | | |
| $Ca^{2+}$ -act- $V_{slope}$ | | 2E | 55 | 7C | A3 | | |
| $Ca^{2+}$ -act- $V_{offset}$ | | 2F | 56 | 7D | A4 | | |
| $Ca^{2+}$ -act- $\tau$ | | 30 | 57 | 7E | A5 | | |
| $Ca^{2+}$ -inact- $V_{slope}$ | | 31 | 58 | 7F | A6 | | |
| $Ca^{2+}$ -inact- $V_{offset}$ | | 32 | 59 | 80 | A7 | | |
| $Ca^{2+}$ -inact- $\tau$ | | 33 | 5A | 81 | A8 | | |
| $Ca^{2+}$ - $g_{max}$ (1/2) | | 34 | 5B | 82 | A9 | | |
| $Ca^{2+}$ - $g_{max}$ (2/2) | | 35 | 5C | 83 | AA | | |
| $Ca^{2+}$ - $V_{equi}$ | | 36 | 5D | 84 | AB | | |
| mod. -act- $V_{slope}$ | | | | | AC | C3 | |
| mod. -act- $V_{offset}$ | | | | | AD | C4 | |
| mod. -act- $\tau$ | | | | | AE | C5 | |
| mod. -inact- $V_{slope}$ | | | | | AF | C6 | |
| mod. -inact- $V_{offset}$ | | | | | B0 | C7 | |
| mod. -inact- $\tau$ | | | | | B1 | C8 | |
| mod. - $g_{max}$ (1/2) | | | | | B2 | C9 | |
| mod. - $g_{max}$ (2/2) | | | | | B3 | CA | |
| mod. - $V_{equi}$ | | | | | B4 | CB | |

**Table A.6** – Correspondence between the parameter number and its order number (continued on the next page)

| Signification | Neuron | | | | | Extra | |
|---|---|---|---|---|---|---|---|
| of the parameter | 0 | 1 | 2 | 3 | 4 | cond. | |
| Syn. A - $\tau$ | 10 | 37 | 5E | 85 | B5 | | |
| Syn. A - $g_{max}$ (1/2) | 11 | 38 | 5F | 86 | B6 | | |
| Syn. A - $g_{max}$ (2/2) | 12 | 39 | 60 | 87 | B7 | | |
| Syn. A - $V_{equi}$ | 13 | 3A | 61 | 88 | B8 | | |
| Syn. B - $\tau$ | 14 | 3B | 62 | 89 | B9 | | |
| Syn. B - $g_{max}$ (1/2) | 15 | 3C | 63 | 8A | BA | | |
| Syn. B - $g_{max}$ (2/2) | 16 | 3D | 64 | 8B | BB | | |
| Syn. B - $V_{equi}$ | 17 | 3E | 65 | 8C | BC | | |
| Syn. C - $\tau$ | 18 | 3F | 66 | 8D | BD | | |
| Syn. C - $g_{max}$ (1/2) | 19 | 40 | 67 | 8E | BE | | |
| Syn. C - $g_{max}$ (2/2) | 1A | 41 | 68 | 8F | BF | | |
| Syn. C - $V_{equi}$ | 1B | 42 | 69 | 90 | C0 | | |
| Stimulation (1/2) | 1C | 43 | 6A | 91 | C1 | | |
| Stimulation (2/2) | 1D | 44 | 6B | 92 | C2 | | |
| Characterization | | | | | | | CC |

**Table A.6** – Correspondence between the parameter number and its order number.

## A.3 Example of parameter configutation

An example of how the digital parameters are sent is provided below. Only the ASIC 1 sodium channel is connected to the membrane for the voltage clamp technique. Thus, the 14-bit register (for the membrane connection) has to be set to "00000100000000". The word in hexadecimal becomes "0100". We have to send the word d:1:0100 (where d:1 indicates the digital word for ASIC 1) through the PC, where it will be routed by the driver in VHDL to the ASIC. The same is also done for all of the other words.

Furthermore, we reported below an example of the parameters sending data to set the FS neuron activity on the first neuron of ASIC2 located in the Ekerö board. Thus, it is necessary to set the topology (which conductances are connected) and then the analog parameters in terms of voltage.

digital parameters

d:2:01C2

d:2:03C0

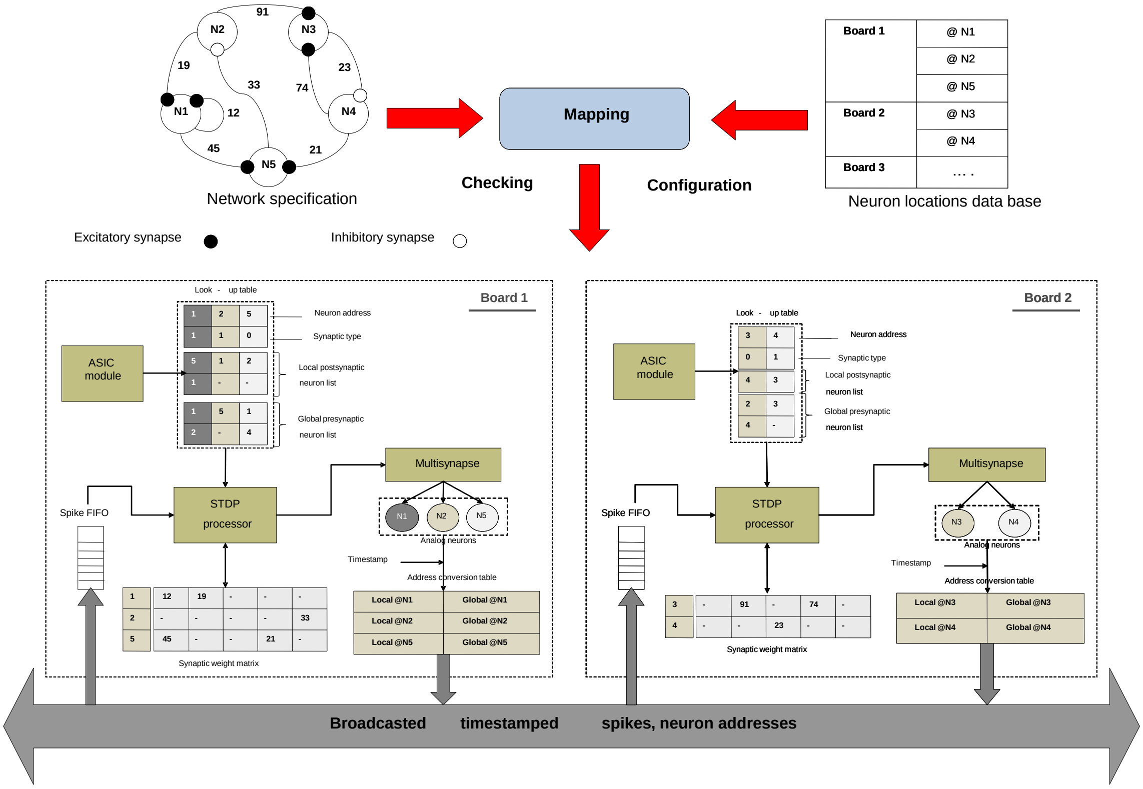
d:2:0500

```
d:2:2800
d:2:01E0
analog parameters
sodium
a:2:00:4.107
a:2:01:2.327
a:2:02:3.932
a:2:03:4.542
a:2:04:2.346
a:2:05:3.935
a:2:06:2.678
a:2:07:2.691
potassium
a:2:08:3.836
a:2:09:2.430
a:2:0A:4.137
a:2:0B:4.468
a:2:0C:2.059
leak
a:2:0D:2.567
a:2:0E:4.990
a:2:0F:2.264
external stimulation
a:2:1C:4.800
a:2:1D:4.900
```

# Appendix B

## Small network in the PAX system: an example of specification

Figure B.1 illustrates an example of a 5-neuron neural network specification.



| | |
|---|---|
| Board 1 | @ N1<br>@ N2<br>@ N5 |
| Board 2 | @ N3<br>@ N4 |
| Board 3 | … . |

Board 1

| | | |
|---|---|---|
| 1 | 2 | 5 |
| 1 | 1 | 0 |
| 5 | 1 | 2 |
| 1 | - | - |
| 1 | 5 | 1 |
| 2 | - | 4 |

| | | | | | |
|---|---|---|---|---|---|
| 1 | 12 | 19 | - | - | - |
| 2 | - | - | - | - | 33 |
| 5 | 45 | - | - | 21 | - |

| | |
|---|---|
| Local @N1 | Global @N1 |
| Local @N2 | Global @N2 |
| Local @N5 | Global @N5 |

Board 2

| | |
|---|---|
| 3 | 4 |
| 0 | 1 |
| 4 | 3 |
| 2 | 3 |
| 4 | - |

| | | | | | |
|---|---|---|---|---|---|
| 3 | - | 91 | - | 74 | - |
| 4 | - | - | 23 | - | - |

| | |
|---|---|
| Local @N3 | Global @N3 |
| Local @N4 | Global @N4 |

**Figure B.1** – Example of specification and operation of a 5-neuron network spread across two boards.

The mapping tool uses two boards to physically map the network. The neural network specification starts in the software layer, where neurons are created as well as their connections following the frame format as designed in Belhadj's thesis (2010). The configuration parameters are attributed to the network during the specification (neuron type, initial synaptic weights, synapse polarity, etc.). In addition to this specification, a database associates the board numbers with the neuron addresses. A compiling tool is in charge of mapping the neurons at their physical locations on the different boards.

There are two types of neuron addresses: local and global addresses. Local addresses are used to distinguish neurons located on the same board. They are used for local computation. Since the number of neurons does not exceed 20 neurons, the local addresses are coded on 5 bits. Global addresses are attributed for each neuron in the system, independently of their physical locations. This address represents the neurons at the network level and is highly used for communication issues. Since the global network can contain up to 400 neurons, the global addresses are coded on 9 bits.

Figure B.1 shows the implementation of the address conversion table just before the communication channel. This table is used to translate the neuron addresses from local to global addresses before initiating a communication procedure. The neuron address is then recognized by the other boards.

The general operation of the system is as follows: as already mentioned, the commu-nication between the digital and analog hardware layers in each daughter board forms a local simulation loop. Whenever a neuron generates an action potential, it will be detected by the local FPGA in which it will be routed towards the target board, based on the connectivity of the network, and then to the target ASIC. Whenever an external action potential arrives to the FPGA, it will set off the computation of the synaptic weights for the plastic connections and then stimulate the target neurons by sending a width-modulated pulse to the corresponding multisynapse(s). Hence, a synaptic voltage is then added to the membrane potential and either helps generate a new action potential (excitatory synapse) or inhibits its generation (inhibitory synapse).

# Appendix C

# Networking activity

This research subject has been funded by the European FACETS-ITN project (Marie-Curie Ph.D. position). As we have already seen in Chapter 1, FACETS-ITN (www.facets-itn.eu) is an initial training network for graduate students in the emerging field of neural computation. It offers a research opportunity to 22 graduate students working in 14 partner institutions in six European countries. Scientifically, the network spans from neurobiology and modeling to theory and novel computing architectures in hardware. As such, it covers many individual subjects that are usually well established at European universities. The joint work includes scientific research but also training in scientific subjects and in additional skills as well as extended secondments to partner laboratories. In terms of research, the network is closely linked to existing EU-funded interdisciplinary projects such as Brain-i-Nets (www.brain-i-nets.eu) and BrainScaleS (www.BrainScaleS.eu). This network emerged from the highly successful FACETS project (www.facets-project.org), which was the largest integrated project supported by the FET (Future Emerging Technologies) program of the European Commission between 2005 and 2010. Within the FACETS-ITN network, it is planned to support a visiting and exchange program for researchers. Secondments take place within the consortium and strengthen the internal scientific network beyond the project meetings. The visits include all industrial partners. Every FACETS-ITN researcher is expected to attend the training workshops offered. In addition, secondments are planned for every student, with a total duration of 3 to 5 months. This appendix briefly describes the training activities carried out within the project.

**My role in the project**

As mentionned in the introduction, the main goal of this Ph.D thesis within the FACETS project is the simulation of cortical cell types (employing the experimental electrophysiological data of these cells as references), using specific VLSI neural circuits to simulate, at the single cell level, the models studied as references in the FACETS project. The second goal is to contribute to the design of a mixed hardware-software platform, specifically designed for the simulation of spiking neural networks using conductance-based models (VLSI neural circuits) that can be used for hybrid artificial neural networks. I attended several training workshops that helped me to carry out this research project. Below are the courses that I have attended (for the detailed program, please visit the website for the FACETS project).

**FACETS-ITN secondments**

From February 2011 to April 2011:

I started my secondments on February 7th, 2011 in Dresden, Germany: 1 month at the "ZMDI company" (http://www.zmdi.com/) and 1.5 months at the "Technische Universität Dresden" (TUD). I attended a training period on highly-parallel VLSI-systems and neuromorphic circuits at the university of Dresden, Germany (Director: Prof. René Schüffny) and vector interchange format in ZMDI. During this period, I was involved in the design of a driver (VHDL) to control the temperature of a neuron chip using a Peltier element.

The driver was designed on Virtex-5 FPGA Xilinx. I developed a driver (VHDL) that reads the temperature value from the sensor, located underneath the silicon neuron, and then controls the Peltier element (placed above the chip)[1] to regulate the temperature on the chip (see Figure C.1).

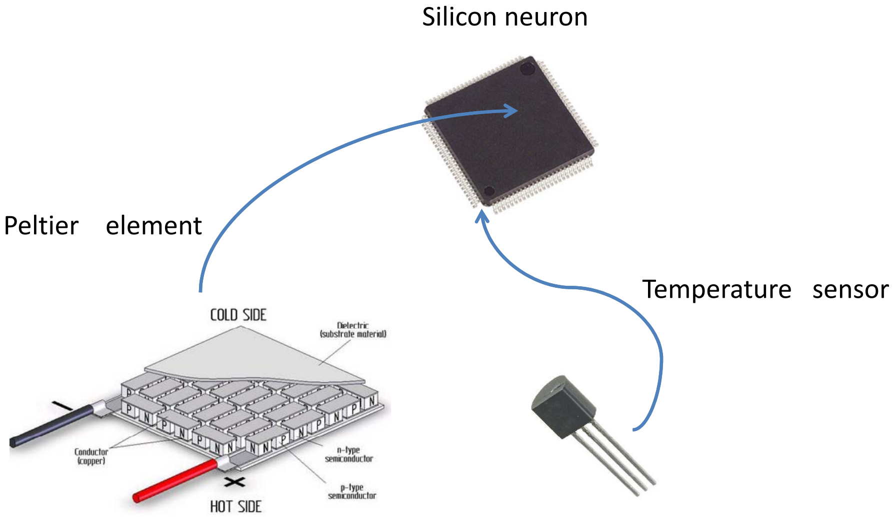


**Figure C.1** – System to control the temperature on the silicon neuron

In particular, the temperature sensor transmits the temperature value to the PC via a serial port. Therefore, according to the desired temperature value, it is possible to control the peltier element in order to cool or heat the chip through an interface (that I designed in C language).

[1] The Peltier device is a heat pump: when direct current runs through it, heat is moved from one side to the other. Therefore, it can be used either for heating or for cooling (refrigeration), although in practice the main application is cooling.

**FACETS-ITN training activity**

From March 26th to March 30th 2012: FACETS-ITN training on Course Neuro-Electronic Interfaces and additional Scientific Writing skills in Bordeaux, France.

From March 19th to March 23rd 2012: FACETS-ITN training on Experiments with Large Scale Hardware Systems and additional skills course: Oral presentation techniques in Jülich, Germany.

From March 6th to March 9th 2012: FACETS-ITN training on Theoretical Approaches to New Computing Concepts in Leysin, Switzerland.

From September 12th to September 16th 2011: FACETS-ITN training on Neuromorphic Electronic Circuits and Additional Skills Course in Heidelberg, Germany at the Kirchhoff Institut für Physik.

From June 6th to June 10th 2011: FACETS-ITN training on Theoretical Neuroscience in Lausanne, Switzerland (EPFL).

From June 3rd to June 9th 2010: FACETS-ITN training on Biology in Gif-sur-Yvette (CNRS), France.