Titre: Title:	Unité de contrôle et de compression de données pour la partie implantable d'un capteur d'électroneurogrammes
Auteur: Author:	Jean-François Roy
Date:	2006
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Roy, J.-F. (2006). Unité de contrôle et de compression de données pour la partie implantable d'un capteur d'électroneurogrammes [Master's thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7904/

UNIVERSITÉ DE MONTRÉAL

UNITÉ DE CONTRÔLE ET DE COMPRESSION DE DONNÉES POUR LA PARTIE IMPLANTABLE D'UN CAPTEUR D'ÉLECTRONEUROGRAMMES

JEAN-FRANÇOIS ROY

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION

DU DIPLÔME DE MAITRISE ÈS SCIENCES APPLIQUÉES

(GÉNIE ÉLECTRIQUE)

Août 2006

Your file Votre référence
ISBN: 978-0-494-19324-2
Our file Notre référence
ISBN: 978-0-494-19324-2

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

UNITÉ DE CONTRÔLE ET DE COMPRESSION DE DONNÉES POUR LA PARTIE IMPLANTABLE D'UN CAPTEUR D'ÉLECTRONEUROGRAMMES

Présenté par : JEAN-FRANÇOIS ROY

En vue de l'obtention du diplôme de : Maitre ès sciences appliquées

M. SAVARIA, Yvon, Ph. D., président

M. SAWAN, Mohamad, Ph. D., membre et directeur de recherche

M. KHOUAS, Abdelhakim Ph. D., membre et codirecteur de recherche

REMERCIEMENTS

Je voudrais d'abord remercier Monsieur Mohamad Sawan, mon directeur qui m'a laissé la chance de faire cette recherche passionnante dans un environnement aussi enviable que le laboratoire Polystim. Je remercie aussi messieurs Yvon Savaria et Abdelhakim Khouas, professeurs à l'École Polytechnique, pour avoir accepté de participer au jury de ce mémoire.

Merci à mes collègues de Polystim, avec qui j'ai eu des discussions et des partages de connaissances très intéressants. En particulier quelques membres de l'équipe Cortisens : Benoit Gosselin, pour son organisation, sa disponibilité et son dévouement, Virginie Simard, pour sa rigueur et sa justesse, Cyprien Dumortier, pour l'apport considérable de son travail à mes recherches, Pierre-Yves Robert et Amer Ayoub.

Merci au CRSNG (Conseil de Recherche en Sciences Naturelles en Génie du Canada) pour leur apport financier ainsi qu'à la CMC (Canadian Microelectronics Corporation) pour leur soutient technique et le partage de leurs ressources.

Je voudrais souligner l'aide que mes amis m'ont apportée, chacun à sa manière, par leurs idées, leurs conseils, leur motivation et leurs sourires. Pierre-Alexandre Fournier, Guillaume Germain, Olivier Duval, Ghislain Provost et Mathieu Plante, pour ne pas tous les nommer.

Je voudrais dire un merci spécial à ma famille pour être là, sans qui je ne serais pas qui je suis.

Un merci unique à Mélanie, ma fiancée, qui me stimule et m'incite toujours à continuer. On a fait beaucoup ensemble et il reste encore beaucoup plus à partager.

RÉSUMÉ

Les scientifiques s'entendent pour dire que les neurones communiquent entre eux par la transmission de signaux par impulsions ou potentiels d'action qui sont les unités de base pour l'encodage neuronal. Ce mémoire propose une analyse et un système spécialisé de contrôle et de compression de données pour la partie implantable d'un électroneurogramme.

Le signal neuronal que le système doit acquérir est masqué en partie par les artefacts rencontrés par ce système comme le bruit, la superposition d'impulsions et les salves de potentiels d'action. Une revue de la littérature montre que plusieurs équipes se sont penchées sur ce problème et ont proposé des solutions fonctionnelles mais pas nécessairement optimisées pour notre type d'application : un implant cortical pour les expérimentations chroniques.

Ce mémoire présente une nouvelle étude des méthodes et algorithmes pertinents pour la réalisation en circuits intégrés d'un système implantable. L'équipe Cortisens du laboratoire de neurotechnologies Polystim propose une solution basée sur l'utilisation de la transformée en ondelettes pour la compression du signal.

Les résultats de simulation montrent qu'il est possible d'obtenir une reconstitution quasi parfaite (spécificité de 99% et sélectivité de 100%) avec un taux de transmission aussi bas que 1.5%. Ce résultat n'est atteignable qu'avec cette méthode en ne considérant que les techniques envisageables pour une implémentation sur puce. Afin d'obtenir ces résultats prometteurs, une méthodologie par seuillage est choisie pour faire la détection de l'activité neuronale qui est la base d'une compression par événements. Cette méthode est la base puisqu'elle suppose que les potentiels d'action ne surviennent que de temps en temps et que la majorité du temps, ce n'est que du bruit qui est observé.

Différentes méthodes de détection neuronales sont donc comparées pour en conclure

que l'implémentation d'un seuil simple après une transformée en ondelettes est suffisante pour une acquisition acceptable. Les méthodes adaptatives proposées dans la littérature fournissent une bonne approche pour une configuration continue et peuvent être appliquées par un logiciel plutôt qu'en matériel. Ce qui permettra une plus grande latitude expérimentale pour le système développé par l'équipe Cortisens.

Ce mémoire met l'emphase sur la réalisation et l'optimisation du contrôleur numérique. Il se compose d'un lien entièrement bidirectionnel qui permet une configuration continue sans influencer le flot sortant de données, par l'écriture aux registres. Il comprend aussi une unité de détection de l'activité neuronale ainsi que le bloc majeur qui gère l'accès aux cellules mémoire tout en accomplissant la compression des données.

Bien que les tests préliminaires ont été effectués sur une plate-forme de prototypage rapide avec FPGA, chacun des blocs a été optimisé pour finalement obtenir une consommation logique minimisant la surface sur puce nécessaire pour une même fonctionnalité. Dans le but d'obtenir un prototype complet sur puce avec une surface restreinte, certains compromis ont dû être pris. Le bloc de transformée en ondelettes prenant manifestement plus d'espace que d'autres blocs a été retiré lors de la réalisation du prototype matériel car il utiliserait 0.92 mm^2 (56%) de l'espace, tandis que le deuxième bloc d'importance, les cellules mémoire, utilisent 0.41 mm^2 (25%) et doivent absolument faire partie du système final car la mémoire représente la fondation du système par fenêtrage basé sur l'occurrence de potentiels d'action. Ce premier prototype sur puce permettra de valider la preuve de concept en faisant l'acquisition de signaux qui corroboreront les résultats de ce mémoire.

La densité finale du système obtenu en circuits intégrés est de 92% pour une surface d'implémentation du coeur de 0.65 mm^2. Ce qui donne une surface totale sur silicium de 1.86 mm^2 incluant les tampons d'entrée/sortie et les contacts permettant la réalisation d'un premier prototype complet par l'assemblage de plusieurs puces superposées avec une matrice d'électrodes interconnectées par micro-manipulations.

ABSTRACT

Scientists agree that neurons communicate information by firing sequences of spikes which are the neuronal code unity. This master's thesis propose an analysis and a specialized system for control and data compression for an implantable electroneurogram.

The neuronal signal acquired with this system can be masked by artifacts like noise, spike superposition and spike train special behaviour. A literature review shows that many teams worked on these problems and proposed functional solutions but not optimized for our application : a cortical implant for chronic experimentations.

This master's thesis presents a new study of pertinent methods and algorithms for the realization in integrated circuits of an implantable system. The Cortisens team of the neurotechnologies Laboratory Polystim proposes a solution based on a wavelet transform for signal characterization and compression.

The simulation results show that it is possible to obtain an almost perfect reconstruction of the signal (specificity of 99% and selectivity of 100%) with a transmission rate as low as 1.5%. These results can be achieved only with this method while considering only the techniques which are conceivable for low power integrated circuits. These results are possible due to a thresholding methodology which is the basis of the compression behaviour. This method is based on the assumption that spikes only occur with low duty cycle and noise is normally observed on the signal seen on the electrodes.

Different methods for neuronal activity detection are compared before concluding that a simple threshold after a wavelet transform is enough for an acceptable acquisition. The adaptive methods proposed in literature give great insight for continuous configuration and can easily be implemented in software instead of hardware. This will let more experimental capabilities for the system still in development by the Cortisens

team.

This master's thesis emphazes on the realization and the optimization of the digital controller. It includes a full duplex link to enable continuous configuration through writable registers without interrupting the uplink data flow. It also includes a neuronal activity detection module and, finally, the most important module which is dedicated to manage the memory blocks while compressing data.

Even if preliminary tests had been done on a fast prototyping platform with a FPGA, each block had been optimized focusing on the minimization of the logic usage using the least chip area as possible for the same functionality. In order to obtain a first complete prototype with integrated circuits using a constrained area, some trade-offs were necessary. The biggest block, the wavelet transform processor, would have used 0.92 mm^2 (56%) of the space, the second biggest part, the memory blocks, use 0.41 mm^2 (25%) and must be part of the final system because it is used as the foundation for a windowing system based on spike event occurrence. This first integrated circuit prototype will validate the proof of concept by acquiring signal that will corroborate results of this master's thesis.

The final density of the realized system is 92% for an implementation area of the core of 0.65 mm^2. This ends with a total area of 1.86 mm^2 including the input/output buffers and the metal contacts forming the chip. This chip will be assembled with other chips and a microelectrode array to constitute a complete stacked dies system.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES NOTATIONS ET DES SYMBOLES

ADC = Analog Digital Converter

AFE = Analog Front End

AP = Action Potential

ASIC = Application Specific Integrated Circuit

BCI = Brain Computer Interface

CWT = Continuous Wavelet Transform

DCT = Discrete Cosinus Transform

DRC = Design Rule Check

DFT = Discrete Fourier Transform

DSP = Digital Signal Processing

DUT = Design Under Test

DWT = Discrete Wavelet Transform

ECoG = Electrocorticogramme

EEG = Electroencephalogramme

EDM = Electrical Discharge Micromachining

ENG = Electroneurogramme

ERD = Event-Related Desynchronization

ERP = Event-Related Potential

FIR = Finite Impulse Response

FFT = Fast Fourier Transform

FLI = Foreign Language Interface

GUI = Graphic User Interface

LFP = Low Field Potential

LVS = Layout Versus Schematics

MEG = Magnetoencephalogramme

MRP = Movement-Related Potential

MSE = Mean Square Error

PCA = Principal Component Analysis

SCP = Slow Cortical Potential

SNR = Signal Noise Ratio

STFT = Short Time Fourier Transform

TDM = Time Division Multiplexing

VHDL = VHSIC Hardware Description Language

VHSIC = Very-High-Speed Integrated Circuit

VLSI = Very Large Scale Integration

LISTE DES ANNEXES

INTRODUCTION

De nos jours, les systèmes d'interface humain-machine montrent de plus en plus leurs utilités en apportant une aide aux handicapés. Ces systèmes représentent l'aboutissement de plusieurs années de recherche mettant en relation plusieurs disciplines complémentaires : l'électrophysiologie avec l'étude du système nerveux, la microélectronique avec la réalisation d'implants, le traitement de signal pour l'analyse, la robotique et plusieurs autres.

Dans le cas qui nous intéresse, nous nous concentrons sur un type particulier d'interface humain-machine qui a pour but de faire l'acquisition des signaux électriques en provenance du cerveau (*Brain Computer Interface*-BCI). Parmi les percées scientifiques dans ce domaine, l'électroencéphalogramme (EEG) a été le premier type d'appareil dédié aux signaux corticaux. Grâce à lui, les scientifiques ont pu caractériser les grandes régions d'activité cérébrales comme les cortex moteur et visuel. Dans un même ordre d'idée, l'électroneurogramme (ENG) est une suite logique de l'EEG pour des besoins accrus de précision, de qualité et de sensibilité.

L'équipe Cortisens du laboratoire de recherche en neurotechnologies Polystim s'est intéressée aux techniques d'acquisition de signaux corticaux. Suivant cette direction, Polystim a développé, et développe encore, une matrice d'électrodes implantable. Chacune des électrodes permettra de suivre l'évolution localisée des stimuli corticaux. L'idée est la réalisation d'une puce implantable qui permettrait de prendre des mesures *in vivo*.

Le but du système d'acquisition consiste à faire l'échantillonnage des signaux électriques du cerveau qui se propagent par impulsions aussi appelées des potentiels d'action. Les signaux sont d'abord amplifiés et filtrés par un circuit analogique (*Analog*

Front End-AFE) puis échantillonnés par des convertisseurs analogique-à-numérique. L'unité de contrôle et de compression numérique coordonne l'échantillonage en synchronisant les différentes fréquences d'horloges pour finalement transférer les données compressées et regroupées par paquets au lien de communication sans-fil.

Ce mémoire se penche sur une partie de ce système, l'unité de contrôle et de compression de données pour la partie implantable d'un ENG. Les points d'importance y sont exposés en partant de la théorie du signal cortical pour aboutir à la réalisation matérielle d'une puce électronique.

Une mise en contexte au chapitre 1 mène à une description des signaux corticaux qui permet d'expliquer quelle information la BCI doit transmettre pour faire l'acquisition neuronale. Les données peuvent être interprétées différemment : une série d'exemples provenant de systèmes existants montrent comment il est possible d'y parvenir.

Dans le chapitre 2 sont présentés les algorithmes qui permettront de compresser le signal afin d'économiser sur la bande passante et ainsi offrir un système multi-canal avec la plus grande densité pour un même lien de communication. De plus, différentes techniques de détection de l'activité neuronale montrent les compromis rencontrés lors de la description et de l'optimisation du système. Le système en tant que tel est décrit en détails au chapitre 3 où le contrôleur numérique est situé dans l'ENG global conçu par l'équipe Cortisens.

Les décisions architecturales expliquées au chapitre 3 proviennent des résultats de simulation pour les comparaisons des algorithmes de compression et des techniques de détection d'activité neuronale décrits au chapitre 2. Tous les résultats sont concentrés dans le chapitre 4 afin de réunir dans un même contexte toutes les preuves décisionnelles.

Finalement, la méthodologie de design est présentée pour expliquer le cheminement et les étapes de développement qui ont menés à la réalisation de prototypes sur plateforme de prototypage rapide et finalement d'un prototype ASIC (*Application Specific Integrated Circuit*).

CHAPITRE 1

LES NEURONES ET LEURS SYSTÈMES D'ACQUISITION

1.1 Introduction

Depuis longtemps la population scientifique s'est intéressée à l'information que l'on peut extraire de signaux neuronaux. Lui méritant son prix Nobel en 1906 en collaboration avec Camillo Golgi sur la structure du système nerveux, le célèbre Santiago Ramón y Cajal introduit l'idée que les neurones représentent la structure fonctionnelle du cerveau, (Ramón y Cajal 1911). Depuis, les travaux ont beaucoup évolué, en particulier avec l'apport substantiel de Hebb qui amena le postulat d'apprentissage. Par contre, son influence prit du temps pour atteindre la communauté des sciences appliquées (Haykin 1998).

Plusieurs groupes se sont penchés sur le problème et ont développé le concept d'interface cerveau-ordinateur plus connu sous l'acronyme BCI (*Brain Computer Interface*). De telles recherches impliquent l'implémentation matérielle de composants dédiés pour l'observation du comportement animal, et pour l'interpréter ensuite. Ces composants se divisent en deux groupes : les implants pour usage chronique et ceux pour usage aiguë. Le transfert et le traitement ne s'appliquent pas de la même façon pour les deux cas. Dans notre contexte, le but final est de réaliser un système entièrement implantable, soit de la catégorie à usage chronique. Certaines restrictions et astuces sont donc applicables pour l'accomplissement de cette tâche. Une description des différents aspects est faite dans la section 1.2. Ces caractéristiques sont la base de la conception de prothèses neurologiques permettant de pallier certaines dysfonctions

dont souffrent les patients visés.

La caractérisation des potentiels d'action (*Action Potential*-AP) faite à la section 1.3 permet d'établir une approche plus pragmatique des tâches d'acquisition. Ce qui nous mène à la réalisation d'un système d'acquisition de signaux neuronaux actionné par l'occurrence d'évènements, soient les AP.

L'époque de la validation expérimentale a débuté depuis plusieurs années. Déjà en 1986, l'équipe Motorlab (Georgopoulos *et al.* 1986) publiait un article montrant l'application de l'encodage du mouvement provenant d'une population de neurones. Aujourd'hui, Motorlab a un bras robotisé activé par l'analyse en temps réel des signaux corticaux. Ainsi, Motorlab performe des algorithmes sur la perception et les intentions de mouvement (Schwartz *et al.* 2004). Une description des différents systèmes existants est faite dans la section 1.5.

1.2 Interface cerveau-ordinateur

Certaines applications BCI ont pour but le contrôle en temps réel, d'autres prônent plutôt l'acquisition pure et dure avec une bonne qualité. Ces deux approches se complètent dans le sens ou l'un tente d'interpréter les signaux fournis par l'autre afin d'interagir. La différence survient dans les restrictions d'application. Par exemple, un système entièrement implantable ne pourra pas offrir autant de quantité et de qualité qu'un autre système avec fils. Les contraintes varient en ce qui concerne la superficie sur le silicium, la consommation de puissance et la bande passante pour le transfert des données.

Parlant de l'intelligence artificielle, Sage décrit un système qui devrait savoir faire trois choses : emmagasiner des connaissances, appliquer les connaissances connues pour

résoudre des problèmes et, finalement, acquérir de nouvelles connaissances (Sage 1990). Haykin le cite et ajoute que les connaissances ne désignent en fait que des données, c'est la représentation ; l'application doit pouvoir se contrôler pour choisir les chemins logiques, c'est le raisonnement ; finalement, les nouvelles connaissances s'incluent dans un système par algorithme, c'est l'apprentissage (Haykin 1998). La figure 1.1 montre les différents modules essentiels de la partie implantable d'une BCI.

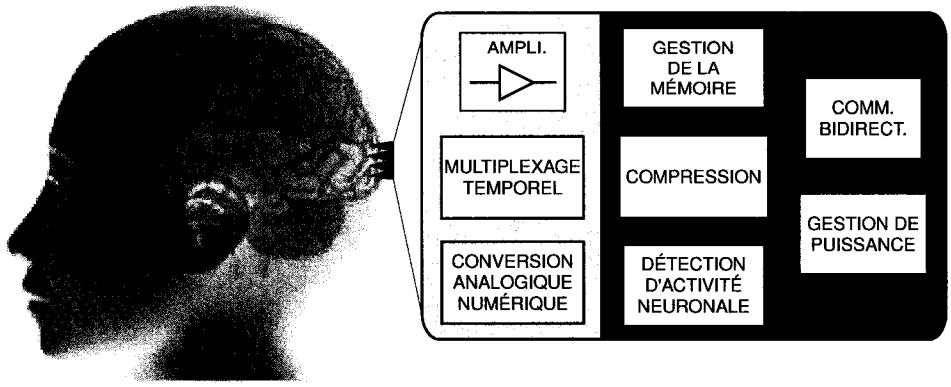


FIG. 1.1 Parties essentielles d'une interface cerveau-ordinateur

L'objectif d'une BCI est d'établir un canal de communication afin de traduire les intentions humaines, reflétées par les signaux corticaux, en un signal de contrôle pour une application logicielle ou une prothèse neurologique (Dornhege *et al.* 2003). L'important devient de se représenter les états courants possibles du cerveau en comparant les caractéristiques des signaux observés avec les paradigmes prédéfinis comme l'idée de mouvements et la vision, par exemple.

1.2.1 L'acquisition

Bien que plusieurs méthodes aient été développées pour faire l'acquisition des signaux neuronaux, trois grandes classes en sont ressorties, l'électroencéphalogramme (EEG),

l'électroneurogramme (ENG) et le magnétoencéphalogramme (MEG). Chacune offre une méthodologie applicable pour des potentiels d'action. Faisant une analogie entre l'enregistrement d'un orchestre symphonique et l'enregistrement neuronal, Buzaki explique bien les perspectives de l'acquisition d'un ensemble de neurones. Il sépare l'acquisition en trois composants critiques qui sont l'interface neurone-électrode, l'identification et la classification des AP, et les outils d'analyse et d'interprétation des salves d'impulsions. Le développement de méthodes objectives et de systèmes novateurs devient nécessaire afin de percer l'incompréhension de l'encodage neuronal, de l'intérieur du cerveau (Buzsaki 2004).

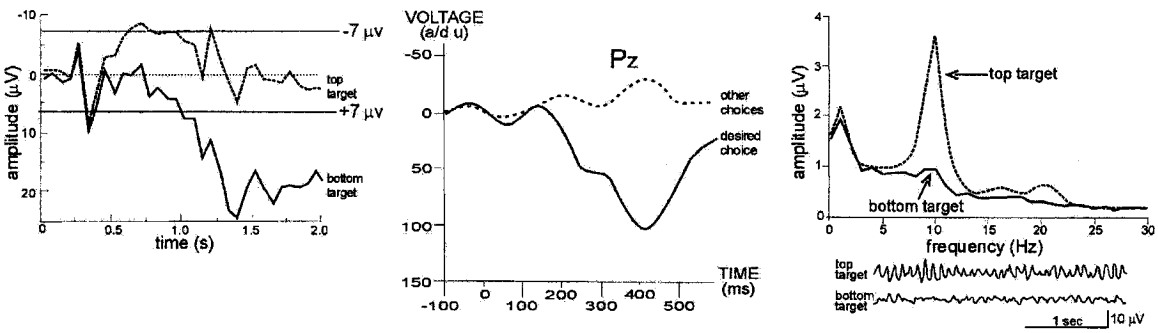
L'électroencéphalogramme

L'électroencéphalogramme (EEG) est une méthode non envahissante qui est aussi considérée comme étant une des plus simples. Elle consiste à placer des électrodes sur la surface de la tête, préférablement directement sur la peau pour plus de stabilité et pour obtenir un meilleur rapport signal sur bruit (*Signal Noise Ratio*-SNR). Par contre, la présence importante de bruit limite son utilisation à des signaux de basses fréquences. De toute façon, d'après Wolpaw, les signaux d'intérêt seraient les SCP (*Slow Cortical Potentials*), plus précisément les ondes μ ou β (Wolpaw *et al.* 2002). Les SCP sont aussi connus sous le nom de LFP (*Low Field Potentials*) et ont la forme montrée aux figures 1.2a-c.

Puisque cette technique est non envahissante, elle offre de nombreuses applications allant de l'aide aux invalides à un dispositif de jeu vidéo. Le problème majeur associé à cette technique est le mauvais SNR dû à l'atténuation des signaux au travers du crâne. Une vaste panoplie de traitements de signaux peut être appliquée à cette méthode d'acquisition, comme pour les autres méthodes d'ailleurs.

Dans son étude de cas, Meinicke résume les quatre principales approches pour ce

type d'acquisition. La première, l'approche de Wolpaw présentée précédemment et considérant seulement les SCP, aurait besoin d'un transfert de données dans les environs de 20-25 bits/min (Wolpaw *et al.* 2002). Une autre approche, basée sur les travaux de Blankertz, analyse particulièrement les ERP (*Event Related Potentials*) afin de déduire l'intention humaine, ce genre de calcul pourrait offrir une classification de signaux neuronaux avec 23 bits/min (Blankertz *et al.* 2001). La troisième approche rejoint les travaux faits par Hinterberger dans un scénario de rétroaction biologique qui nécessite beaucoup de traitements préalables pour aboutir à un transfert aussi bas que 6 bits/min dans un système pseudo temps-réel (Hinterberger *et al.* 2004). La dernière approche analysée regarde principalement les signaux P300, qui représentent une impulsion survenant environ 300 ms après une intention de mouvement. Leur modèle permet une vraie gestion temps réel avec un débit prometteur de 12 bits/min (Meinicke *et al.* 2002).



a) SCP enregistré au dessus du cortex visuel alors que l'usager tente de faire bouger un curseur avec les yeux.

b) P300 enregistré au dessus du de la région centropariétale alors que l'usager prend une décision.

c) Contenu spectral enregistré au dessus du cortex sensorimoteur alors que l'usager tente de faire bouger un curseur

FIG. 1.2 Signal typique à basse fréquence,source : (Wolpaw *et al.* 2002)

L'électroneurogramme

Cette méthode envahissante aussi appelée ECoG pour électrocorticogramme est un procédé qui consiste à la disposition de micro-électrodes directement dans le cortex

pour faire l'enregistrement électrique de l'activité. Cette méthode est particulièrement appropriée pour des implants chroniques qui permettent aux sujets de vaquer à leurs occupations normales.

La résolution spatiale de l'électroneurogramme (ENG) est nettement supérieure à celle de l'EEG puisque le crâne n'est plus là pour atténuer les signaux ; il offre donc une meilleure qualité d'acquisition en augmentant le SNR. De plus, étant donnée sa résolution, il permet une plus grande densité d'analyse qui pourrait permettre de couvrir plus exactement la région désirée. Il ouvre ainsi la voie à l'observation de région prédéfinie comme le cortex visuel ou moteur pour en extraire plus facilement les ERP qui y sont associés.

Le plus grand point négatif de cette méthode est l'obligation de recourir à la chirurgie pour insérer les électrodes jusqu'au cortex. Vu cette contrainte, cette méthode offre malheureusement une expérimentation limitée de nos jours.

Le magnétoencéphalogramme

Le magnétoencéphalogramme (MEG) est une méthode non envahissante qui consiste à lire le champ magnétique induit par les courants circulant dans les dendrites. Ces courants appelés dipôles permettent d'identifier l'activité d'un ensemble de neurones plutôt qu'un neurone en particulier en déduisant la position, la direction et la magnitude des dipôles.

On pourrait aussi inclure dans cette catégorie toutes les méthodes moyennement envahissantes comme le PET (*Positron Emission Tomography*) qui utilise un biotraceur radioactif, le fMRI (*functionnal Magnetic Resonance Imaging*) et l'imagerie par laser. Ces méthodes se rapportent principalement au flux sanguin plutôt qu'à une représentation électrique, ce qui permet tout de même de faire une corrélation

entre cette activité et l'activité neuronale. Ces techniques sont toutes freinées par une longue constante de temps et sont encore très demandantes en matériel pour un système complet et portable. Ce travail n'en fera plus allusion.

1.2.2 La stimulation électrique neuronale

La stimulation électrique neuronale est un domaine de recherche en tant que tel et ne fait pas partie de cette recherche. Il implique le développement de modèles pour l'interface tissu-électrode et pour une carte neurotopique qui permet de déterminer à quel endroit, à quel moment et de quelle façon il convient de stimuler. Sans ce modèle, on ne peut pas estimer l'impact de l'injection d'un courant ou de l'application d'une tension sur les tissus nerveux. En effet, avant de stimuler, il est important de comprendre comment les signaux transmettent et reflètent l'information pour ensuite être en mesure de reconstituer l'effet désiré.

1.2.3 La variété des tâches

Un système BCI temps-réel complet requiert grossièrement cinq tâches importantes afin d'acquérir des AP et d'agir rétroactivement.

1. Une détection d'activité neuronales permet d'identifier les données consistantes.
2. Une caractérisation des signaux permet de représenter les données en préservant le maximum d'information. Le concept d'extraction de propriétés (*feature extraction*) réfère à cette tâche.
3. Une localisation temporelle permet de reconstituer le signal échantillonné.
4. Une classification permet d'associer l'information à un comportement connu.

5. Une rétroaction permet d'agir directement par des actions spécifiques en fonction du contexte comme le cortex moteur pour activer un bras robotisé.

Les tâches dédiées à l'acquisition se résument par les trois premières tâches énumérées, tandis que les deux dernières représentent plutôt les traitements spécifiques à l'application, pour un système BCI complet.

1.3 Caractérisation des potentiels d'action

Un neurone est une cellule particulièrement adaptée à la propagation des signaux électriques générés en réponse à une réaction chimique ou d'autres stimuli. De plus, ces signaux se propagent rapidement d'une cellule à une autre. Deux approches globales peuvent être choisies pour en faire l'étude. Premièrement, l'encodage neuronal est le lien entre le stimulus et la réponse du nerf sous la forme de AP, aussi appelés impulsions. Comme le décrit Dayan, les neurones encodent et transmettent l'information sous forme de séquences de AP. Deuxièmement, à l'opposé, le décodage neuronal considère la quantité d'information contenue dans la séquence de AP et son défi est de reconstruire un stimulus en fonction de la réponse (Dayan et Abbott 2001).

L'architecture biologique simplifiée d'un neurone est dessinée à la figure 1.3. Les dendrites reçoivent des signaux d'entrée provenant d'autres neurones et les transmettent au soma. Le soma, non montré, est la cellule centrale où les échanges chimiques se produisent selon une évolution non linéaire. Si l'accumulation électrique des entrées dépasse un certain seuil, un autre signal est émis en sortie par l'axone vers d'autres cellules. L'arbre de branchement des dendrites permet à plusieurs neurones de s'interconnecter au travers d'une ou plusieurs connexions synaptiques. Un neurone simple dans le cortex d'un vertébré fait la connexion à plus de 10^4 autres neurones (Gerstner et Kistler 2002).

La figure 1.3 présente les endroits propices à l'observation neuronale. Le signal du haut représente le potentiel intracellulaire du soma, le signal du centre montre des potentiels d'action et le signal du bas représente le potentiel intracellulaire au niveau de l'axone (Dayan et Abbott 2001).

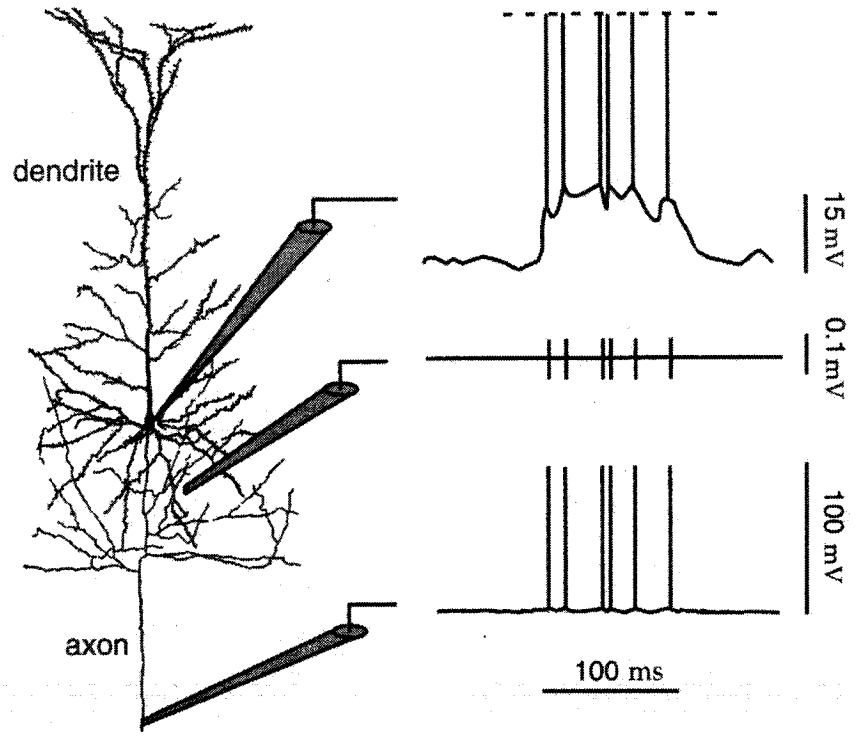


FIG. 1.3 Trois sites d'enregistrement neuronal possibles,
source : (Dayan et Abbott 2001)

Plus précisément, le signal électrique est le résultat de la différence entre les potentiels intérieurs et extérieurs de la membrane neuronale. Le signal électrique est présenté à la figure 1.4. Une pompe à ions située dans cette membrane maintient le gradient de concentration pour maintenir le potentiel (environ -70 mV). Lorsque le gradient atteint un certain niveau de seuil de 15 à 20 mV de plus que la tension de repos, le canal ionique s'ouvre et survient la phase de montée du AP, la polarisation, qui rééquilibre

ainsi les charges ioniques. La montée dure environ 1 ms, puis le canal se referme et la phase de descente s'amorce jusqu'à un certain niveau, c'est l'hyperpolarisation qui porte ce nom puisque la tension redescend à un niveau inférieur au potentiel de repos.

En dernier, vient la période réfractaire, introduit par Hodgkin et Huxley en 1952. Cette période représente le temps durant lequel les canaux ioniques sont inactifs, donc le temps durant lequel un AP ne peut avoir lieu. Dans les faits, la réactivation des canaux ioniques se fait selon une manière stochastique et le seuil d'activation peut être atteint prématurément si le potentiel d'excitation est suffisamment élevé.

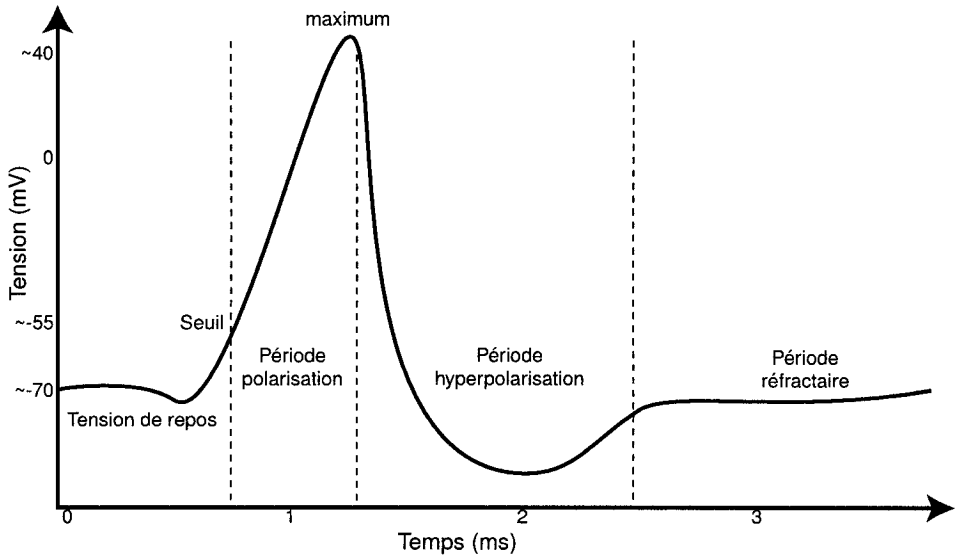


FIG. 1.4 Potentiel d'action

La période réfractaire est le temps suivant immédiatement un AP, aucune nouvelle impulsion n'est possible à ce moment car c'est le temps nécessaire à la pompe à ions pour se recharger. Sa durée s'estime entre 5 et 20 ms et un nouveau AP est fortement probable 10 ms plus tard (Pouzat *et al.* 2004). Cette définition devient particulièrement significative lorsqu'il s'agit de calculer un possible taux de compression du signal (section 2.2).

Quelques obstacles peuvent obstruer l'acquisition et nuire à l'analyse et à la reconnaissance des signaux enregistrés. Premièrement, le bruit peut masquer un AP généré par un neurone qui serait placé trop loin d'un site d'enregistrement. Deuxièmement, la superposition de plusieurs AP sur un même site qui pourrait être causée par la proximité relative de plusieurs neurones. Troisièmement, l'atténuation d'amplitude entre deux AP d'un même neurone produit lors d'un train consécutif d'impulsions. De plus, le volume d'information à considérer est trop grand pour tout conserver, surtout en considérant que plusieurs sites sont enregistrés simultanément d'où la nécessité de faire un traitement de base permettant de choisir les échantillons significatifs.

1.3.1 Le bruit

L'acquisition des signaux neuronaux est sujette à plusieurs types de bruit et parfois, il peut même devenir impossible de discerner le signal utile, on dit alors qu'il est entièrement couvert et que le SNR est pauvre ou petit. Le bruit peut provenir de toutes les sources qui ne sont pas les neurones que le système a pour but d'enregistrer.

Le bruit observé peut provenir autant de l'environnement que des outils et circuits servant à faire la lecture du signal (Oweiss et Anderson 2000). Premièrement, le bruit thermique dû aux amplificateurs composant l'étage d'entrée du circuit et faisant l'interface avec les tissus nerveux. Le bruit thermique est par définition un bruit dit blanc. Deuxièmement, le bruit de quantification qui est introduit par le convertisseur analogique-numérique (*Analog Digital Converter*-ADC). Ces deux causes peuvent être classées comme non corrélées donc indépendantes. De plus, dans cette catégorie peut s'ajouter les variations dues au déplacement du capteur par rapport au site d'enregistrement, l'impédance de l'électrode et bien d'autres sources indépendantes comme présentées dans (Musial *et al.* 2002).

Un autre type de bruit observable est directement lié à l'activité des autres neurones environnants et qui ne sont pas reconnus comme des AP. Ce type de sources contient un mélange de bruit corrélé, provenant de neurones proches mais trop éloignés pour être identifiés comme tels, ainsi que du bruit non corrélé provenant d'activité locale mais faible. Ce type de bruit est considéré comme coloré (Chandra et Optican 1997).

Bien que certains ne soient pas d'accord avec cette méthode (Kim et Kim 2000), le problème se simplifie en considérant le bruit comme étant du type blanc gaussien (Pouzat *et al.* 2002). Une bonne description[1] :
" *Un bruit blanc est un signal aléatoire dont la densité spectrale de puissance est constante quelle que soit sa fréquence. En d'autres termes, la probabilité qu'un bruit blanc possède une certaine puissance est la même pour toutes les fréquences.*"

Étant donné que le bruit est indépendant du signal, le signal échantillonné s'exprime donc comme à l'équation (1.1) où s(t) est le signal d'intérêt, tandis que b(t) est le bruit. Le signal observé pourrait ressembler au signal de la figure 1.5b.

$$s_e(t) = s(t) + b(t) \tag{1.1}$$

1.3.2 La superposition d'impulsions

Cette caractéristique est directement liée à la position spatiale de l'électrode par rapport aux neurones environnants. Si un site d'enregistrement est spatialement situé au centre de deux neurones, par exemple, des impulsions quasi simultanées pourraient être difficilement identifiables.

La figure 1.5c montre une possibilité d'occurrence d'impulsions quasi simultanées.

[1] *http ://fr.wikipedia.org/wiki/Bruit_blanc* consultée le 6 octobre 2005

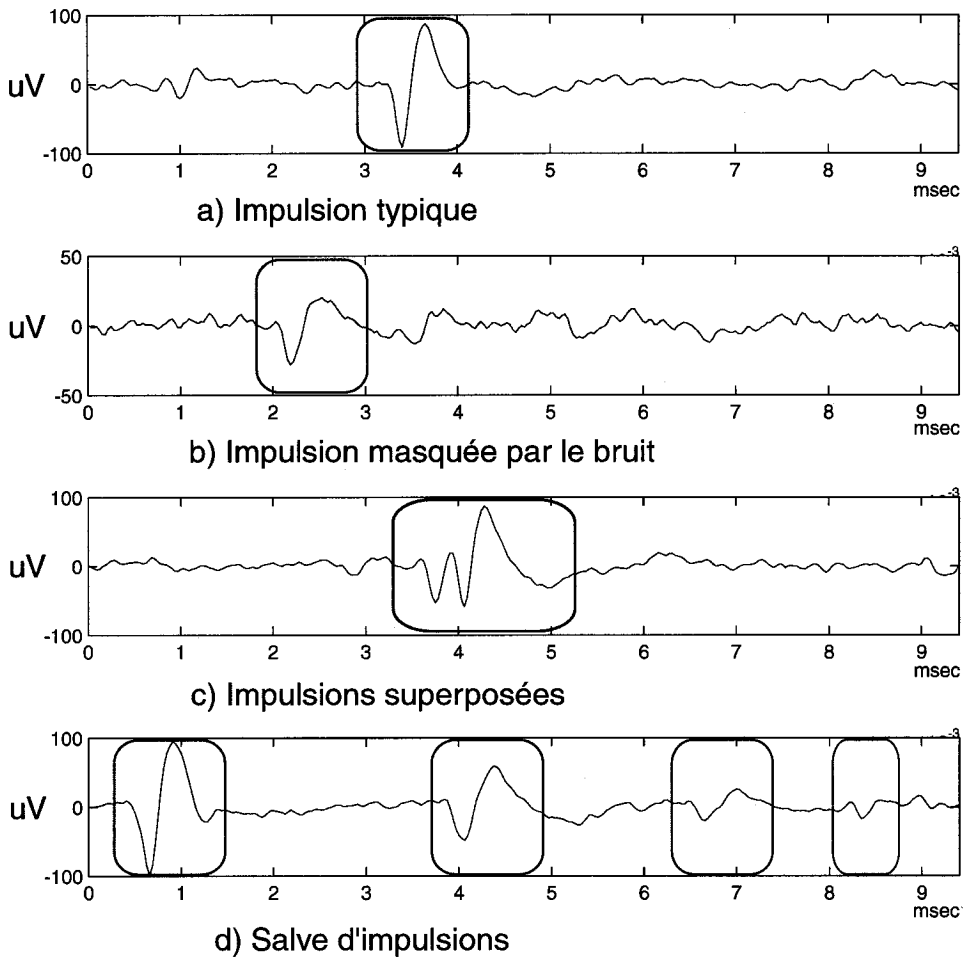


FIG. 1.5 Caractéristiques des potentiels d'action,
source du signal : université Duke (même qu'aux chapitres 2 et 4)

En d'autres occasions il serait possible que le signal résultant soit interprété comme du bruit, même par des yeux d'experts. À ce sujet, Wood présente justement les divergences d'analyses faites par différents experts (Wood *et al.* 2004).

1.3.3 Les salves d'impulsions

Lorsqu'un neurone est fortement actif, c'est-à-dire que si les dendrites alimentant un neurone permettent d'accumuler suffisamment de charges dans le soma, le neurone en question peut émettre une salve d'impulsions dont l'intervalle dépend de la période réfractaire. Selon Pouzat, suivant un AP, un nouveau AP est fortement probable 10 ms plus tard avec une atténuation significative en amplitude comme on peut l'observer sur la figure 1.5D (avec une période réfractaire plus courte) (Pouzat *et al.* 2004).

Lewicki disait en 1998 que le taux d'occurrence pouvait s'estimer entre 20 et 100 Hz, ce qui correspond à la période réfractaire. Par contre, il ajoutait que le contenu fréquentiel des AP allait jusqu'à 4 kHz (Lewicki 1998). Cette dernière information est cruciale pour le choix de la fréquence d'échantillonage. Une discussion plus approfondie est faite dans la section 2.2.

1.3.4 Les potentiels d'action en équations

Si l'on simplifie le problème à sa plus simple expression en ignorant la durée du potentiel d'action qui devrait être environ 1 ms, on obtient une série de fonctions de Dirac (δ) où le temps t_i est l'occurrence d'un AP et n le nombre de AP dans la séquence analysée.

$$s(t) = \sum_{i=1}^{n} \delta(t - t_i) \tag{1.2}$$

La quantité d'information à transférer pour une séquence de données est représentée par le taux d'activité moyen (T_{act}) pour une période d'analyse T :

$$T_{act} = \frac{n}{T} \tag{1.3}$$

Par contre, cette information ne nous informe pas suffisamment. Il serait plus pratique de connaître l'évolution temporelle de cette fonction ; surtout si le but est de l'associé à un stimulus. Prenons une moyenne de ce taux d'activité sur une fenêtre coulissante de grandeur Δt. Cette fenêtre doit être suffisamment grande pour calculer un bon estimé de la moyenne.

$$T_{act}(t) = \frac{1}{\Delta t} \int_{t}^{t+\Delta t} s(\tau) d\tau \tag{1.4}$$

où $s(\tau)$ est l'extension continue du signal de l'équation (1.2). La probabilité qu'une impulsion soit vue sur un intervalle Δt autour de t_i est simplement

$$p[t_i] = T_{act}(t) \cdot \Delta t \tag{1.5}$$

Il est important de concevoir l'acquisition de signaux neuronaux comme un procédé temporel variant et stochastique. En fait, il est impossible de prédire à quel moment et selon quelle amplitude un neurone va générer une impulsion. Pour le savoir, il faudrait connaître toutes les séquences possibles qui relient un stimulus avec les émissions d'impulsions, donc l'encodage neuronal (ce qui n'est pas encore le cas). Il serait donc plus juste de dire simplement que la probabilité qu'un neurone émette une impulsion à un moment donné est nulle si $\Delta t \to 0$.

Le $T_{act}(t)$ ne fournit malheureusement pas suffisamment d'information pour nous prédire la probabilité de l'occurrence d'une impulsion. Par exemple, la probabilité que deux neurones émettent un AP n'est pas nécessairement le produit des deux probabilités individuelles parce qu'il est possible que la présence de l'un influence l'occurrence de l'autre. Par contre, puisque la génération d'un AP dépend directement de son historique et que les intervalles successifs sont indépendants, donc les évènements eux-mêmes sont indépendants ; nous avons donc un processus de Poisson (Dayan et Abbott 2001).

Le processus de Poisson

Dans plusieurs applications comme le comptage des AP, l'intérêt est de compter le nombre d'occurrence dans un certain laps de temps ou dans l'espace. La variable aléatoire de Poisson est tout indiquée pour caractériser cette situation "imprévisible" (Leon-Garcia 1994). Sa probabilité p_k pour k occurrences est :

$$p_k = \frac{(T_{act})^k}{k!} e^{-T_{act}} \tag{1.6}$$

La moyenne, ou l'espérance E[X], et la variance VAR[X] sont équivalentes avec ce type de variables aléatoires. La variance est une mesure qui permet de voir l'étendue des données aux alentours d'une moyenne

$$E[X] = VAR[X] = T_{act} \tag{1.7}$$

Fonction de densité d'intervalle

Par définition, une fonction de densité de probabilité est une manière utile de spécifier l'information contenue dans une fonction de distribution comme l'avènement d'un AP par rapport à un précédent. Une caractéristique intéressante de la distribution de Poisson est que le nombre d'occurrences pendant un certain temps implique que l'intervalle entre chaque évènement est distribué exponentiellement (Leon-Garcia 1994). Une variable aléatoire exponentielle n'a pas la propriété de mémoire, c'est-à-dire que les évènements ne sont pas influencés par leurs précédents. Ce qui est pratiquement le cas pour un neurone si l'on ne considère pas la période réfractaire ou aucune activité n'est possible. Donc, puisque l'intervalle suit une exponentielle, il est fort probable que le temps entre deux AP soit court et peu probable qu'il soit long.

On pourrait exprimer ce taux d'activité par l'intervalle inter-impulsions, ou ISI (*Inter*

Spike Interval). Suivant une probabilité exponentielle, la moyenne donne donc :

$$E[ISI] = \frac{1}{T_{act}} \tag{1.8}$$

et la variance :

$$VAR[ISI] = \frac{1}{(T_{act})^2} \tag{1.9}$$

Cette densité représente l'activité générale du neurone plutôt que sa forme donnant seulement l'information sur l'état actuel. Cette fonction de densité semble mieux adaptée pour l'analyse de population à l'instar de l'analyse d'échantillons indépendants (Pouzat *et al.* 2004, Wood *et al.* 2004).

1.4 Les signaux d'intérêt pour l'acquisition

Plusieurs méthodes et procédés ont été mis en oeuvre pour les différents types de signaux. Que ce soient les signaux à basse fréquence ou les impulsions, chacune des caractéristiques importe ; allant de sa forme à sa fréquence d'occurrence en passant par le synchronisme des différents neurones. Dans le domaine de la computation de signaux corticaux, le terme propriétés des signaux revient souvent, surtout lorsqu'il est question de faire de l'extraction de propriété (*feature extraction*) (Dornhege *et al.* 2002). Cette tâche est effectuée par des filtres particuliers et des algorithmes dédiés.

1.4.1 Les potentiels d'action

La méthode d'acquisition d'ENG, bien qu'envahissante, est aussi beaucoup plus précise que les autres méthodes. Elle sert souvent de point de comparaison pour corroborer

les résultats obtenus avec les signaux à basse fréquence (Dornhege *et al.* 2002) car il est assez aisé de reconstituer des signaux similaires aux enregistrements EEG avec des signaux ENG.

Les types de filtres applicables pour des signaux comme les potentiels d'action couvrent une grande gamme de filtres possibles (Wilson et Emerson 2002, Lewicki 1998). Ces catégories incluent le filtrage du bruit, la détection d'événements et la compression du signal. Ces points seront couverts dans le chapitre 2.

En supposant qu'un neurone émet toujours de la même façon par son axone vers les autres neurones, la forme de l'impulsion pourrait sembler ne pas avoir d'importance pour une reconstitution fidèle. Positionner temporellement les impulsions sur une droite pourrait sembler suffisant. Par contre, ne connaissant pas la position spatiale exacte de l'électrode lors de l'acquisition, les signaux provenant de neurones distincts avoisinant peuvent être identifiés comme tels si et seulement si la forme du signal a été préalablement reconnue afin de faire la déduction. Hulata propose une méthode pour le tri des AP basée sur les paquets d'ondelettes et l'information mutuelle de Shannon (Hulata *et al.* 2002). Pouzat et Wood suggèrent l'analyse ISI (section 1.3) comme étant adéquate pour voir l'évolution de l'information (Pouzat *et al.* 2004, Wood *et al.* 2004).

1.4.2 L'information neuronale

Dans son travail de renom, Shannon propose que "*le problème fondamental est de reproduire à un point, soit l'exactitude ou l'approximation, d'un message choisi à un autre point*" (Shannon 1948). Shannon définit donc une bonne mesure du contenu de l'information (I) d'un message ainsi que l'entropie (H) qui est une moyenne de

l'information transmise par message.

$$I(x_k) = log\Big(\frac{1}{p(x_k)}\Big) = -log[p(x_k)] \tag{1.10}$$

$$H(X) = E[I(x_k)] = \sum_k p(x_k) \cdot I(x_k) = -\sum_k p(x_k) \cdot log[p(x_k)] \tag{1.11}$$

où $p(x_k)$ est la probabilité d'occurrence d'un message x_k. Cela indique que si un message survient nécessairement, aucune information supplémentaire n'est apportée en le notant $(p(x_k) = 1 \rightarrow I = 0)$; donc on peut aussi conclure que moins l'occurrence d'un événement est probable plus l'information qui découle de sa notification est grande. Cette théorie est la base pour la construction d'un système de communication qui sera efficace et fiable. (French *et al.* 2003) utilise ce théorême comme point de comparaison pour évaluer l'efficacité d'algorithmes de compression ; bien que leur méthode est un peu douteuse du point de vue qualité, l'idée reste intéressante.

De plus, on peut aussi prouver que si les variables sont indépendantes, l'entropie est additive.

$$H(X,Y) = H(X) + H(Y) \tag{1.12}$$

Cette dernière équation est d'ailleurs appliquée dans l'équation du signal bruité (1.1) où les signaux peuvent simplement être considérés comme superposés.

1.5 Système d'acquisition neuronaux existants

Plusieurs groupes de recherche ont déjà étudié la question et ont développé des prototypes. La compagnie Cyberkinetics en collaboration avec l'université de l'Utah (Branner *et al.* 2004) offre un système commercial d'enregistrement et de traitement temps-réel pour des expérimentations aiguës. Cyberkinetics a évolué à partir de tra-

vaux présentés par l'équipe de Normann qui avaient proposé un système à 100 canaux (Guillory et Normann 1999). Des chercheurs de l'Université du Michigan ont développé une électrode multi-sites à différents niveaux (Wise *et al.* 2004). Les systèmes existants se divisent grossièrement en trois groupes, soient les systèmes avec processeurs, les systèmes intégrés et les autres systèmes connexes.

1.5.1 Systèmes avec processeurs et microcontrôleurs

Certains groupes décrivent leurs systèmes de façon détaillée comme Nicolelis qui présente une méthode pour l'enregistrement simultané d'ensembles neuronaux. Ce groupe a bâti un système de 96 canaux avec des micro-fils implantés chroniquement avec une grande capacité de calcul fournie par des modules commerciaux disponibles, soient des processeurs dédiés et autres DSP (*Digital Signal Processing*) (Nicolelis 1999).

Des chercheurs du Centre de recherche de Wadsworth (Wolpaw *et al.* 2003) proposent de leur côté un système basé sur des EEG enregistrés sur la tête. Cette équipe offre un système d'enregistrement générique non envahissant et partage ouvertement son logiciel baptisé BCI-2000. Les membres de cette équipe développent actuellement des applications utiles pour les invalides du cortex moteur, comme faire bouger un curseur sur un écran d'ordinateur sans l'usage des mains.

Un projet européen appelé VSAMUEL (Folkers *et al.* 2003) présentait un système permettant l'acquisition de 128 canaux avec un échantillonnage de 50k éch/s. Le système se compose de puces commercialement disponibles : des électrodes implantables, des étages d'amplification, des ADC ainsi que des plate-formes DSP. Un module DSP est montré en exemple à la figure 1.6. Leur système permet une gestion

et un traitement en temps-réel qui utilise une DWT (*Discrete Wavelet Transform*) par banque de filtres afin de faire la compression des signaux. Ce projet est celui qui ressemble le plus à ce que notre équipe Cortisens voudrait réaliser, à l'exception que nous visons une implémentation matérielle complète sur micro-puce.

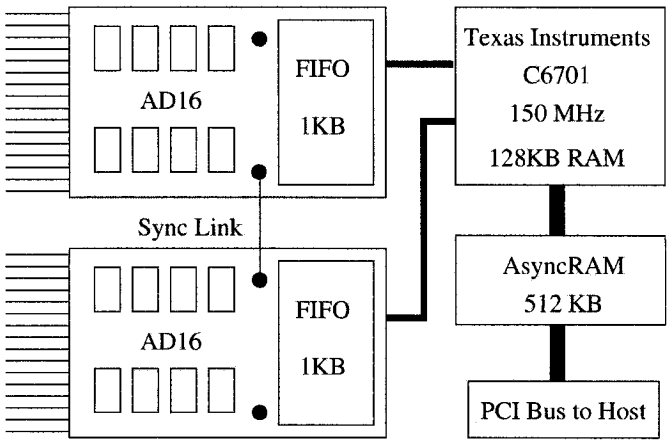


FIG. 1.6 Système exemple, module DSP du projet VSAMUEL, source : (Folkers *et al.* 2003)

1.5.2 Systèmes intégrés sur puce

Suivant une autre approche, les systèmes entièrement implantables occupent une bonne partie de l'intérêt des chercheurs car ils sont mieux adaptés pour des implantations à long terme. L'équipe de Wise propose un système hybride avec une technique avancée de micromachinage permettant une bonne densité du côté des électrodes avec un système externe non intégré (Wise *et al.* 2004).

Boahen présente dans son article une méthode de utilisant l'adressage d'événements. C'est-à-dire qu'il considère seulement si un AP est produit ou non et transmet simplement sa position spatiale plutôt que l'information intégrale. Boahen en vient à la conclusion qu'il est mieux d'utiliser un système de communication avec un lien

partagé et un arbitre plutôt qu'un lien dédié, car les informations proviennent majoritairement par accès sporadiques et un lien n'est que rarement utilisé à sa pleine capacité pour une longue période de temps. Il propose une solution originale pour la gestion de queues et de files (Boahen 2000).

Électronique analogique et mixte

L'équipe de l'université Duke (Obeid *et al.* 2004a, Obeid *et al.* 2004b) présente un système de télémétrie à seize canaux réalisant 31.5 k échantillons/s supportés par un système embarqué personnalisé comprenant un PC et une interface sans-fil 802.11b[2]. Le système est léger, alimenté avec une batterie, portable mais pas entièrement implantable.

De son côté, l'équipe de l'université de Tübingen (Hinterberger *et al.* 2004) propose aussi un système s'intéressant aux SCP mais pour en extraire la pensée. Le système utilise une boucle de rétroaction afin d'établir un lien entre les stimuli visuels et les 64 sites d'enregistrement EEG. La société Danica[3] offre un système complet avec ordinateur, écran, panneau de commande, chariot et supports.

1.5.3 Autres domaines reliés

En regardant comment des ingénieurs s'intéressant à d'autres domaines d'activité ont résolu leurs problèmes, on en tire une information précieuse. Par exemple, l'équipe de Segura-Juarez a développé un système de très grande précision et de haute densité. Les memmbres de cette équipe ont développé un système basé sur l'occurrence d'évènements pour 512 canaux échantillonnés à 100 MHz avec une résolution de 1 kHz entre chaque événement. Bien que leur but était d'analyser l'identification de parti-

[2] *http ://grouper.ieee.org/groups/802/11/* standard adopté depuis 1999
[3] *http ://www.danica.nl/neuro/neuro.htm* consultée le 27 septembre 2005

cules subatomiques, leur application présente des similitudes flagrantes avec l'acquisition de signaux neuronaux (Segura-Juarez *et al.* 2004).

Dans un autre ordre d'idées, Chueh et Hatfield présentent un système d'acquisition temps-réel pour l'analyse des odeurs, un nez électronique (H^2EN). Ils utilisent une matrice de capteurs spécialisés couplés à une application logicielle qui peut fournir une rétroaction. L'intéressant de leur article est qu'il présente le problème dans son intégralité, allant du capteur à l'utilisation graphique en passant par les algorithmes utilisés et les mathématiques s'y rattachant (Chueh et Hatfield 2002).

L'équipe Euratom (Batista *et al.* 2002) a aussi exploré les systèmes de commande basés sur l'occurrence d'événements. Ils ont particulièrement ciblé l'amélioration des performances en ce qui concerne le transport de données, le traitement de signal, la synchronisation de systèmes et l'emmagasinage de données pour de grands débits. Ils résument bien les compromis entre les différents liens de communication disponibles et compare l'utilisation de FPGA versus celle de DSP.

CHAPITRE 2

LE TRAITEMENT ET LA COMPRESSION DES SIGNAUX NEURONAUX

2.1 Introduction

Dans le chapitre 1, les signaux d'intérêts pour un système d'acquisition cortical ont été présentés. Différents systèmes et algorithmes ont été introduits et seront développés plus en détails dans ce chapitre. Afin de choisir et de valider la méthode d'application la plus pertinente, plusieurs avenues ont été évaluées. En partant des caractéristiques décrites dans le chapitre 1, et considérant qu'il est superflu de transmettre toutes les données échantillonnées, le présent travail utilise le modèle classique de la théorie de l'information de Shannon pour développer un système de traitement dédié.

Une analyse pour une compression optimale est faite dans la section 2.2. Le barème de sélection se construit en comparant les fonctions de coût utilisées dans les systèmes de contrôle intelligent associés aux paramètres décisifs tels que :

- minimiser le taux de transfert
- minimiser l'erreur quadratique moyenne (*Mean Square Error*-MSE)
- minimiser la consommation d'énergie pour une tâche spécifique, ou minimiser la complexité de l'algorithme

D'après les résultats prometteurs aperçus dans la littérature, différents traitements de données applicables pour une implémentation matérielle sont comparés dans la section 2.3. Cette section survole principalement l'application de la transformée rapide de Fourier (*Fast Fourier Transform*-FFT), de la transformée discrète en cosinus (*Dis-*

crete Cosinus Transform-DCT) et de la transformée discrète en ondelettes (*Discrete Wavelet Transform*-DWT). L'application des filtres dans un système en particulier peut varier en fonction des goulots d'étranglement rencontrés dans le flot de données désiré. C'est pourquoi une analyse des mouvements d'ensemble, des processus, ou des schèmes possibles est faite dans la section 2.4.

Par la suite, les différentes méthodes de seuillage pour la détection d'activité neuronale sont présentées à la section 2.5. Ceci permet de finalement déterminer l'architecture qui sera réalisée sous la forme d'un système numérique temps réel au chapitre 3.

2.2 Analyse pour une compression optimale

En science computationnelle et dans la théorie de l'information, la compression est l'action d'encoder l'information en moins de bits que la représentation non encodée. Dans le cas d'un système de communication, l'encodeur qui transmet doit correspondre avec le décodeur qui reçoit l'information. Cette action devient possible car les cas normaux sont redondants.

L'idéal, pour obtenir une compression maximale, est d'approcher le plus possible le taux optimal proposé par Shannon (Shannon 1948). La problématique de choisir un type de compression se pose et se résout différemment en fonction du problème. La compression d'une chaîne de caractères ne se traitera certainement pas de la même façon qu'un encodage neuronal.

Mackay analyse globalement ce problème en le contraignant à un "compresseur" de fichiers et il le divise simplement en deux (Mackay 2003) :

1. Un compresseur *avec pertes* compresse des fichiers, mais il pourrait associer

le même encodage à différentes données. Assumons que l'usager requiert une reconstitution parfaite, alors cette possible confusion mène à une erreur (dans le cas d'applications sur des images, une compression avec perte est normalement considérée comme satisfaisante). Donc pour qu'un compresseur avec perte soit pratique, il faut minimiser la probabilité que la confusion survienne.

2. Un compresseur *sans perte* associe chaque groupe de données dans un fichier à un encodage singulier. Si cette méthode réduit la taille de certains fichiers, elle cause nécessairement que d'autres pourraient être plus long.

2.2.1 Points de comparaison

Le système considéré ici ressemble plus à une application sur des images plutôt que sur une chaîne de caractères. Les compresseurs sans perte, majoritairement liés aux bases sur la minimisation de la redondance (Huffman 1952), seront donc mis de côté. Ce qui nous permet d'éliminer rapidement une grande classe de compresseurs incluant des applications logicielles populaires comme ZIP et TAR pour des fichiers, et GIF et PNG pour des images, ou même ceux produisant des séquences de référence comme l'encodage de Huffman et l'encodage arithmétique.

Puisque la quantité de données passant par le port de communication devient rapidement un goulot d'étranglement pour un système d'acquisition, l'option d'une compression avec perte devient plausible. Par contre, l'analyse des données faite ultérieurement pourrait nécessiter une reconstitution parfaite. Ce paradigme pourrait être contourné en utilisant une méthode de compression permettant une compression sans perte en même temps comme les algorithmes JPEG avec la DCT et JPEG2000 avec la DWT par exemple.

Afin de déterminer quelle méthode est la mieux adaptée pour notre système, nous devons prendre certains points de comparaison. Un des buts premiers est la maximisation de la bande passante permettant de servir un plus grand nombre de canaux pour un même lien. Un autre est que si une compression est désirée par l'usager, l'erreur de reconstruction doit être minimisée. Le choix final devra tenir compte de l'implémentation matérielle qui sera faite et la complexité du design devra être minimisée ce qui mène à un minimum de ressources et un minimum de puissance utilisée lors du traitement.

2.2.2 Débit de transfert

Le but principal de cette recherche est de minimiser la bande passante qui caractérise la quantité de données pouvant circuler sur le lien de communication. Il s'agit donc d'un contrôleur permettant une certaine configuration pouvant mener à la détermination des caractéristiques optimales. Un des buts est de planifier les développements possibles dans un avenir moyennement rapproché pour obtenir un système maximisant le nombre de canaux échantillonnés. Le débit de transfert devient donc notre point de comparaison pour l'efficacité du modèle à choisir.

Fréquence d'échantillonnage

La quantité d'information à acquérir et à transférer est phénoménale si aucun traitement ou compression n'est fait. Premièrement, le choix de la fréquence d'échantillonnage doit respecter la fréquence critique de Nyquist ($f_c = 2f_{maxsignal}$) afin d'éviter le recouvrement. Dans le cas le plus simple d'un sinus, deux points par cycle est l'échantillonnage minimum. Dans notre cas, bien que les signaux arrivent à un débit approximatif de 10 à 100 fois par seconde, le contenu fréquentiel du signal à échantillonner se trouve principalement dans une bande limitée allant jusqu'à 4 kHz. Donc, afin de

respecter le théorème d'échantillonage, la fréquence doit absolument être supérieure à 8 kHz et préférablement de beaucoup supérieure pour la détection post-acquisition des impulsions superposées.

Un échantillonage brut des données (D_{brut}) est difficilement praticable pour un système à plusieurs canaux ($N_{canal} \ggg 1$) puisque la bande passante nécessaire dépasse rapidement les capacités disponibles pour des systèmes de communication conventionnels. Par exemple, 16 canaux (N_{canal}) convertis à 8 bits par échantillon (N_{ech}) à une fréquence suréchantillonnée de 32 kHz (f_{ech}) demanderaient un débit de plus de 4 Mbits/s (équation 2.1). Ceci est plutôt loin des objectifs demandant de plus en plus de densité afin d'obtenir une bonne représentation de l'activité neuronale.

$$D_{brut} = N_{ech} N_{canal} f_{ech} \tag{2.1}$$

Compression par détection

Considérons que le signal est composé simplement de AP isolés et qu'entre chaque AP il n'y a que du bruit. En ne conservant que les impulsions, on peut réduire la quantité de données transférées. Cela donne un débit de transfert variable $D_{detect}(t)$ (équation 2.2) directement lié au taux d'activité $T_{act}(t)$ (équation 1.4) expliqué au chapitre 1.

$$D_{detect}(t) = N_{win} N_{ech} N_{canal} T_{act}(t) \tag{2.2}$$

où N_{win} est la largeur, en nombre d'échantillons, de la fonction de fenêtrage en temps discret. Les données ne sont donc transmises qu'à chacune des occurrences des AP. La largeur de la fenêtre est choisie en fonction du temps nécessaire pour enregistrer complètement un AP avec une courte période avant et après l'occurrence si une analyse plus approfondie est souhaitée ($N_{win} \propto f_{ech}$). Puisque la durée moyenne est de 1 à 2 ms, si l'on reprend l'exemple de l'échantillonnage à 32 kHz, une fenêtre d'ana-

lyse comprenant 64 points permettrait de couvrir l'entité de la durée d'un AP ; si on inclue dans le calcul l'hypothèse d'une activité de 100 pulsations par seconde, on obtient un débit d'un peu plus de 819 kbits/s. La description de différentes méthodes de détection est faite dans la section 2.5.

Déjà, une compression temporelle semble intéressante et n'influence que très peu l'exactitude du signal acquis : seule l'information reliée au bruit est mise de côté. Bossetti a d'ailleurs implémenté un système basé sur cette approche (Bossetti *et al.* 2004). Notons cependant que le débit théorique doit être pris avec un bémol ; même s'il semble beaucoup plus bas que le débit brut, il est possible que tous les canaux détectent un AP en même temps. Cela amène un nouveau problème, soit la gestion des canaux avec une mémoire tampon car l'information doit être accumulée et transmise par la suite à un débit minimal.

Compression par transformée

Si le signal a subi une transformation ou un filtrage quelconque, et qu'une sélection des coefficients seulement est transmise, on obtient une compression avec perte. L'important est que la sélection demeure représentative et suffisante pour une reconstitution acceptable. Cette méthodologie affecte directement le nombre N_{win} qui peut changer en fonction de la transformée dans l'équation (2.2). En considérant le taux de compression dû à la sélection des coefficients (T_{sel}), on obtient un nombre de bits à transmettre pour chaque paquet :

$$N_{comp} = N_{win} N_{ech} T_{sel} \tag{2.3}$$

Paquetisation

Étant donné que le transfert de données implique des fenêtres d'échantillonnage basées sur l'occurrence d'événements, il faut aussi présenter le problème de la reconstitution

chronologique du signal. Cette tâche est souvent remplie par des agglomérations de données encapsulées dans un protocole sous la forme de paquets. Les techniques de paquetisation sont multiples et représentent un sujet de recherche en tant que tel et elles sont particulièrement utilisées dans le domaine des télécommunications comme la téléphonie et internet.

Simplifions la description en considérant seulement le surplus de données transmis dans l'en-tête précédent les données utiles afin de pouvoir réassembler les données correctement. La quantité de données inclues dans cet en-tête dépend directement du protocole utilisé pour la communication. Plus le protocole sera simple, plus la proportion de données utiles sera grande ; de la même manière, plus les paquets seront gros, plus la proportion de données utile sera grande aussi. Normalement, la quantité de données associées à l'en-tête est fixe et s'additionne pour chacun des paquets transmis. Il y a donc un prix à payer pour l'envoi d'un paquet. On peut réécrire l'équation (2.2) ainsi :

$$N_{paquet} = N_{comp} + N_{head} \tag{2.4}$$

$$D_{detect}(t) = N_{paquet} N_{canal} T_{act}(t) \tag{2.5}$$

2.2.3 L'erreur quadratique moyenne

Parlant de modèle dans la théorie des décisions statistiques, Hastie résume que la méthode la plus utilisée et la plus pratique pour faire l'analyse de similitude est l'erreur quadratique moyenne (*Mean Square Error*-MSE) (Hastie *et al.* 2001). Donc, cette moyenne approche la meilleure prédiction de l'erreur possible. Étant donnée une

réponse Y en fonction d'une entrée x, la MSE se calcule :

$$MSE(Y) = \frac{1}{N}\sum_{i=1}^{N}\Big(Y_i - f(x_i)\Big)^2 = \frac{(d_{euclidienne})^2}{N} \tag{2.6}$$

ou bien en considérant le biais qui est la différence entre la moyenne de l'estimateur et la donnée à être estimée : $biais(Y) = E[Y] - f(x)$.

$$MSE(Y) = VAR[Y] + (biais(Y))^2 \tag{2.7}$$

Basée sur la distance euclidienne ($d_{euclidienne} = \|Y - f(x)\|$), l'erreur sera minimale si les points de reconstitution correspondent aux points attendus. Analytiquement, plus la variance sera grande, plus l'erreur sera grande. Elle nous donne une bonne idée de l'efficacité d'une méthode. En plus elle nous permet un référencement en fonction des autres caractéristiques désirées comme le taux de compression dans le cas de notre système. L'erreur quadratique devient donc notre point de comparaison pour la consistance du modèle à choisir.

2.2.4 Complexité d'implémentation

Cette partie concerne la différenciation qu'il devrait y avoir en matériel et en logiciel. Le strict minimum doit évidemment être conçu pour une implémentation matérielle. Le temps de calcul n'est pas un problème majeur dans une application vu que la fréquence d'échantillonage est relativement basse. Par contre, le nombre de multiplicateurs et d'additionneurs est beaucoup plus critique. Souvent, la complexité d'un système se calcule en associant une fonction de coût aux spécifications et à l'optimisation en évaluant différents détecteurs de AP (Obeid et Wolf 2004). Une telle

évaluation est faite à la section 2.5.

D'un point de vue physique, considérons la consommation de puissance et la surface de silicium qui sont les deux points principaux et qui ne sont pas indépendants l'un de l'autre. Si la surface utilisée est grande cela implique que la fonctionnalité du système utilise un plus grand nombre de portes logiques. L'utilisation logique varie en fonction du nombre de canaux désirés et en fonction de l'ampleur de la tâche à accomplir. Par exemple, un comparateur de front montant comme dans un oscilloscope est plus simple qu'un module accomplissant une DWT suivi d'un comparateur.

D'un point de vue logiciel, la communication en temps-réel avec la partie matérielle est un prérequis. De plus, plusieurs inconnues demeurent comme la connaissance du bruit qui sera observé sur les électrodes. En conservant une majeure partie du système en logiciel, il sera plus aisé de l'adapter dans le futur lorsque les algorithmes seront plus mûrs. Surtout que le système d'acquisition a pour but d'être un point de départ pour une application BCI qui n'est pas encore déterminée.

2.3 Filtres et transformées en temps réel envisageables

Chacune des méthodes présentées ici comportent globalement deux phases qui consistent premièrement à extraire des coefficients ou des paramètres importants du signal neuronal. Ce qui permet de réduire la dimensionnalité du problème tout en conservant une consistance suffisante. Deuxièmement, il faut une étape de classification qui permet de déterminer si le signal est un AP ou non. Donc les points de comparaison se limiteront à l'exactitude de détection, la capacité de compression et la qualité de reconstitution.

Pour une représentation uniforme des algorithmes, chacun utilisera le même ensemble

de données montré à la figure 2.1. Le signal a généreusement été fourni par Iyad Obeid de l'université Duke selon une expérimentation faite le 7 avril 2004 : données acquises *in vivo* sur un singe (*owl monkey*) dans le cortex prémoteur dorsal.

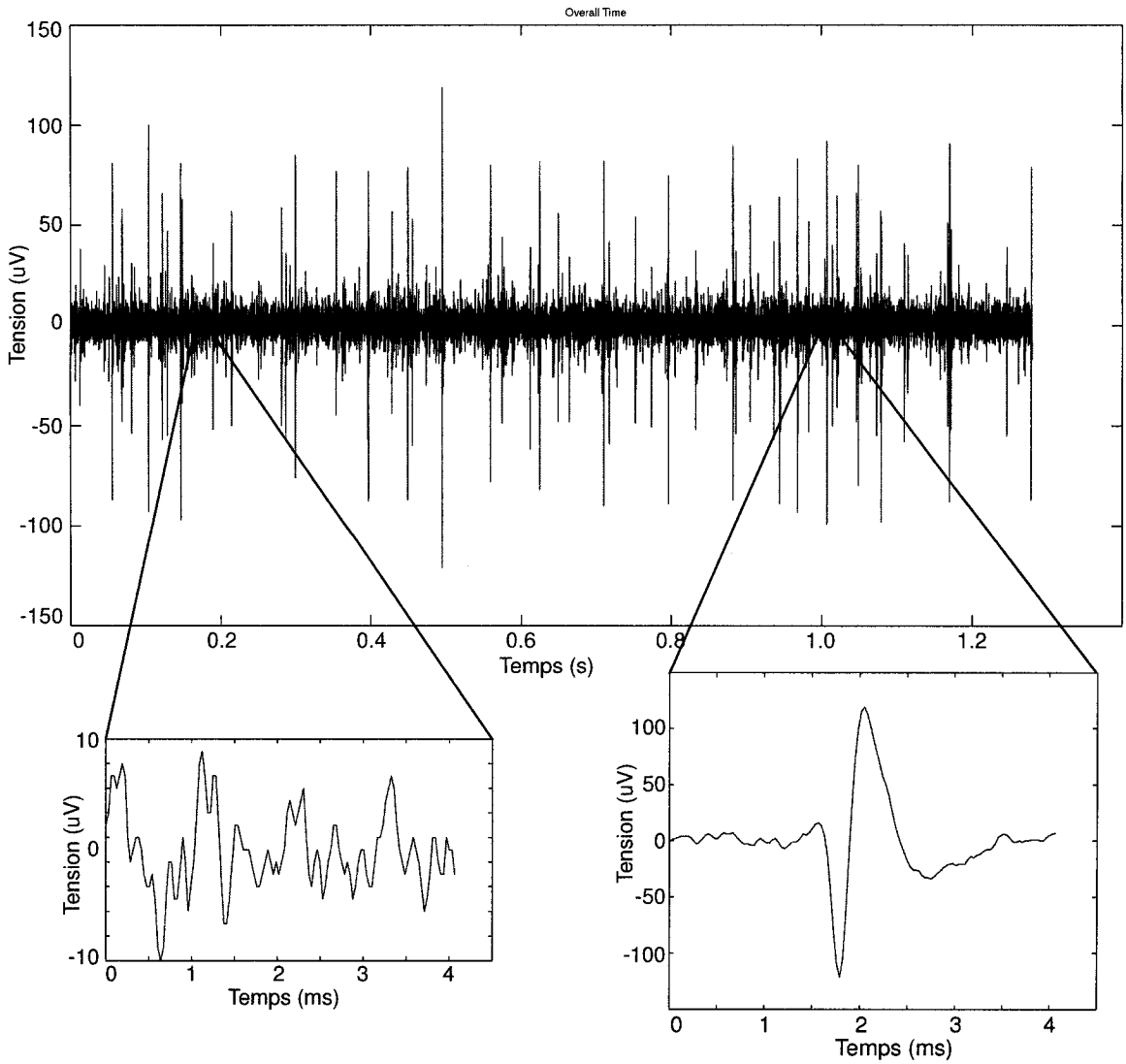


FIG. 2.1 Signal de référence avec agrandissement sur du bruit et sur un AP

2.3.1 Transformée rapide de Fourier

La transformée rapide de Fourier (*Fast Fourier Transform*-FFT) est un algorithme pour calculer la transformée discrète de Fourier (*Dicrete Fourier Transform*-DFT) avec une complexité moindre, $O(Nlog(N))$ au lieu de $O(N^2)$ où N est le nombre d'échantillon. La transformée de Fourier est une projection dans le domaine des fréquences. Elle donne une représentation différente de la même fonction selon la relation (X_k) ou sa transformée inverse (x_n) pour la reconstitution :

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} nk}, \forall k = 0, ..., N-1 \tag{2.8}$$

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} nk}, \forall n = 0, ..., N-1 \tag{2.9}$$

La figure 2.2 montre la transformée de la séquence de référence présentée à la figure 2.1. On peut y voir que les fréquences d'intérêt sont majoritairement sous les 6 kHz. Ce qui correspond à l'approximation assumée comme vraie (Lewicki 1998).

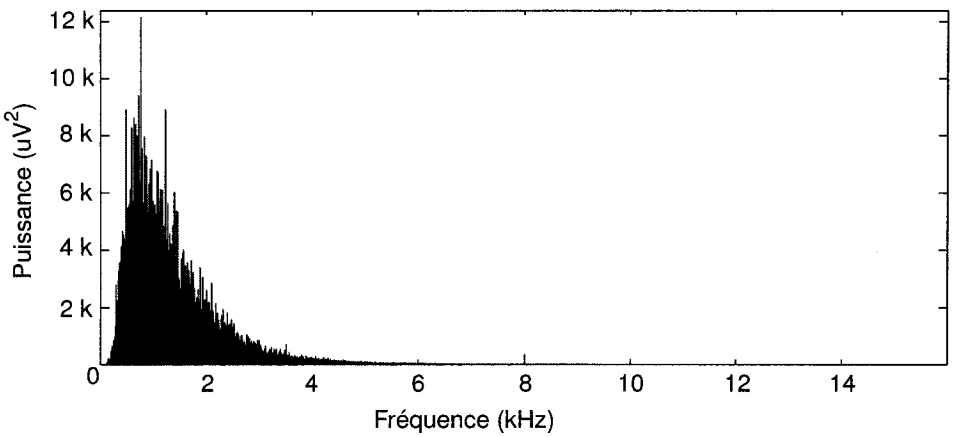


FIG. 2.2 Contenu spectral d'un signal neuronal

La méthode de compression consiste à découper le signal en petits segments et de transmettre seulement les données reliées aux fréquences d'intérêt. Bien que le signal peut sembler pouvoir se compresser énormément, il faut tenir compte du fait que le résultat de la FFT donne un nombre complexe, donc il contient aussi l'information de la phase en plus de l'amplitude. Ceci revient à dire que chacun des points contient deux fois plus d'information et nécessite aussi deux fois plus de bits pour le représenter.

L'utilisation en pratique de la FFT sur un signal continu est habituellement l'implémentation nommée STFT (*Short Time Fourier Transform*). Cet algorithme applique une fenêtre coulissante $(g(t-m))$

$$STFT = X(m,f) = \sum_{n=-\infty}^{\infty} x(n)g(n-m)e^{-2i\pi fn} \tag{2.10}$$

Le compromis majeur à faire en utilisant la STFT est le choix de la résolution qui doit demeurer fixe. Plus la fenêtre d'échantillonage est grande, plus la précision fréquentielle sera bonne ; par contre, la précision temporelle en est amoindrie. Si le but est de localiser temporellement une composante fréquentielle comme une impulsion soudaine, le choix de la résolution devient problématique.

2.3.2 Transformée discrète en cosinus

La transformée discrète en cosinus (*Discrete Cosinus Transform*-DCT) est de la même catégorie que la FFT. La complexité est la même $(O(Nlog(N)))$, par contre elle utilise seulement des nombres réels. Cette transformée linéaire peut être vue comme une rotation en fonction de l'espace d'entrée (le temps) vers une nouvelle base de projection constituée de vecteurs unitaires. Ici, le nouveau domaine de projection devient des cosinus seulement tandis que la FFT est un mélange de sinus et de cosinus.

Cette méthode est souvent utilisée dans les cas de compression avec perte car elle possède la propriété de concentrer l'information du signal dans quelques composants seulement. Cette transformée approche le cas statistiquement idéal de l'analyse par composants principaux (section 2.3.4). La DCT est utilisée dans certaines normes de compression comme JPEG pour les images et MPEG pour le vidéo. Une autre forme modifiée est aussi populaire pour d'autres applications comme les compressions audio connues Vorbis et MP3.

On voit dans la figure 2.3 que l'information est majoritairement concentrée dans les coefficients de basses fréquences.

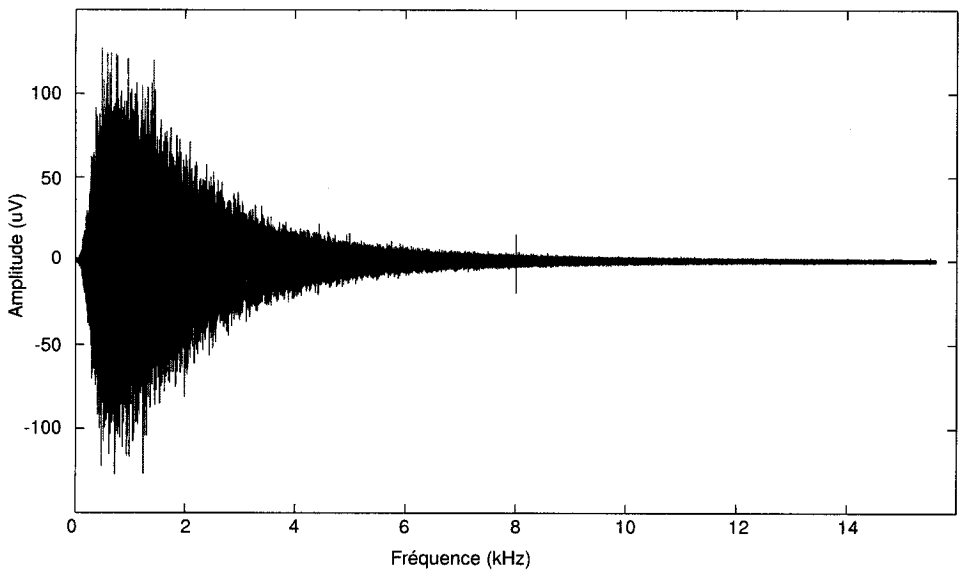


FIG. 2.3 Transformée DCT

Différents algorithmes pour l'implémentation ont été developpés, mais la forme la plus connue de la DCT est le type-II qui est un cas équivalent de la DFT avec 4N entrées où les éléments pairs sont égaux à zéro. Elle serait donc associée à une FFT optimisée qui calculerait la moitié nécessaire des coefficients pour éviter la redondance. Elle se

définit comme ceci :

$$X_k = \sum_{n=0}^{N-1} x_n cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \tag{2.11}$$

et son inverse la DCT de type-III :

$$X_k = \frac{1}{2}x_0 + \sum_{n=1}^{N-1} x_n cos\left[\frac{\pi}{N}n\left(k+\frac{1}{2}\right)\right] \tag{2.12}$$

L'avantage principal de la DCT sur la FFT est qu'elle donne une meilleure résolution fréquentielle pour une même résolution temporelle. Elle permet donc d'obtenir une meilleure localisation temporelle d'un événement fréquentiel en utilisant une plus petite fenêtre d'analyse.

Le désavantage majeur de cette technique est que le fenêtrage doit être fait préalablement car c'est le nombre d'échantillons *Nwin* qui va déterminer la résolution. De plus, la dérivée de la fonction aux bornes de la fenêtre doit être nulle, c'est-à-dire que la variation du signal aux bornes doit être minimisée, ce qui peut être fait seulement si l'AP est bien centré sur la fenêtre d'échantillonnage.

2.3.3 Transformée discrète en ondelettes

Basée sur la convolution, la transformée discrète en ondelettes (*Discrete Wavelet Transform*-DWT) est l'aboutissement de plusieurs années de recherches principalement menées par Gabor, Morlet et Grossmann dans des domaines aussi variés que la séismologie et la physique théorique. Leurs recherches ont, en fait, réactivé l'utilité d'un concept mathématique relié à l'analyse harmonique et plus spécifiquement à l'étude multi-échelle (Mallat 1999). Comme la FFT, la DWT est réversible. Dans le domaine des ondelettes, les vecteurs de projection sont plus complexes que dans le cas

de la FFT, ils sont représentés par les "fonctions mères" (ψ). La fonction d'ondelettes possède une moyenne de zéro ($\int \psi(t)dt = 0$) et elle peut être dilatée d'un facteur 'a' puis translatée par τ. Ce qui donne un vecteur orthogonal à chaque échelle :

$$\psi_{a,\tau}(t) = \frac{1}{\sqrt{a}}\psi\Big(\frac{t-\tau}{a}\Big)dt \tag{2.13}$$

C'est la propriété de multi-résolution des ondelettes, représentée à la figure 2.4.

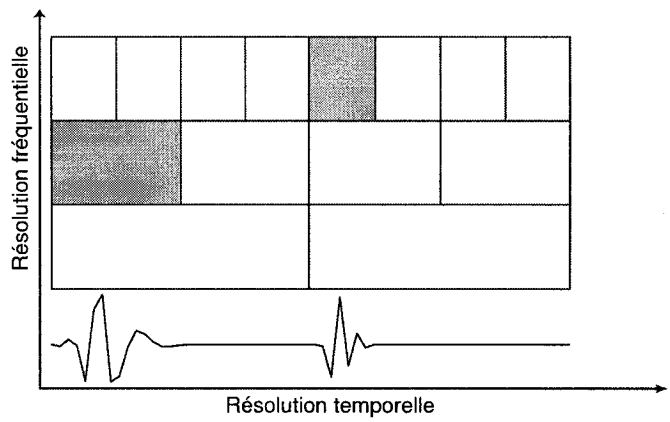


FIG. 2.4 Changement d'échelle et de translation pour l'ondelette Daubechie (db4)

Contrairement à la FFT, il existe une infinité de vecteurs de base, ou d'ondelettes mères. Grossièrement, chacune des ondelettes fait des compromis entre la compacité de la localisation dans l'espace et la fluidité du signal. La transformée en ondelettes convolue le signal d'entrée (équation 2.14) par l'ondelette mère qui peut être dilatée ou contractée pour l'analyse à différentes échelles (Kalayci *et al.* 1994).

$$Y_n = X_n * \Psi_n = \sum_{n=-\infty}^{\infty} x(n)\psi^*_{a,\tau}(n) = \frac{1}{2\pi}\sum_{w=-\infty}^{\infty} x(w)\psi^*_{a,\tau}(w) \tag{2.14}$$

Cette technique a l'avantage de résoudre autant les grandes que les petites échelles,

c'est-à-dire qu'elle permet une bonne localisation temporelle jointe à une bonne résolution fréquentielle. La détection de singularité peut être faite à la manière d'un "zoom" de résolution. La transformée peut être vue comme une mesure des variations aux alentours de la translation τ ; les coefficients élevés de la transformée indiquent la position des fronts dans le signal, les discontinuités, comme l'arrivée d'un AP.

Le choix de l'ondelette devrait suivre les propriétés intuitives suivantes (Mallat 1989) :

1. L'opérateur doit être linéaire ;
2. La fonction d'approximation à une résolution donnée est une fonction la plus similaire possible à la fonction d'entrée ;
3. L'approximation d'un signal à une résolution j+1 contient toute l'information pour calculer le même signal à une résolution moinde j ;
4. L'opération d'approximation est similaire à chaque résolution et est dérivée du facteur de dilatation dû au changement d'échelle ;
5. Le nombre d'échantillon à l'entrée est le même qu'à la sortie mais translaté ;
6. Plus la résolution diminue, plus la quantité de données que la transformée contient diminue aussi.

Les filtres passe-haut, $h(t)$, et passe-bas, $g(t)$, doivent être complémentaires, ou plutôt en mirroir quadratique (équation 2.15) qui consiste à inverser l'encodage des basses fréquences pour les hautes et vice versa.

$$g(N-1-n) = (-1)^n h(n) \tag{2.15}$$

Le filtre $h(t)$ retourne les coefficients détaillés ($d(n)$) à cette résolution tandis que le filtre $g(t)$ retourne les coefficients d'approximation ($a(n)$).

$$a(n) = \sum_{k=-\infty}^{\infty} x(k) g(2n-k) \tag{2.16}$$

$$d(n) = \sum_{k=-\infty}^{\infty} x(k)h(2n-k) \tag{2.17}$$

Connaissant la position n, et considérant que chacun des filtres possède tout leur contenu fréquentiel utile dans la moitié de leurs coefficients, Mallat prouve que la résolution fréquentielle est doublée à chaque étage de filtre. Ce qui demeure en accord avec le principe d'incertitude de Heisenberg qui stipule de l'impossibilité de connaître exactement et à la fois la position et le momentum (Mallat 1989).

L'ondelette mère doit être choisie judicieusement. En particulier pour le domaine numérique où la transformée est appliquée sur une fenêtre et non sur l'infinie ; l'utilisation de filtres à réponses impulsionnelles finies (*Finite Impulse Response*-FIR) est donc appropriée comme ceux utilisés à la figure 2.5. Afin de passer d'une échelle à l'autre, une architecture en arbre doit être utilisée et les mêmes filtres peuvent être réutilisés. À chaque étage, les coefficients pairs sont sélectionnés et les autres ignorés, un sous-échantillonnage est donc effectué ($\downarrow 2$). L'architecture montrée à la figure 2.5 est optimale, pour les AP, du point de vue qu'elle implémente seulement la moitié des filtres pour augmenter la résolution dans les bandes de fréquence d'intérêt seulement contrairement à la DWT traditionnelle.

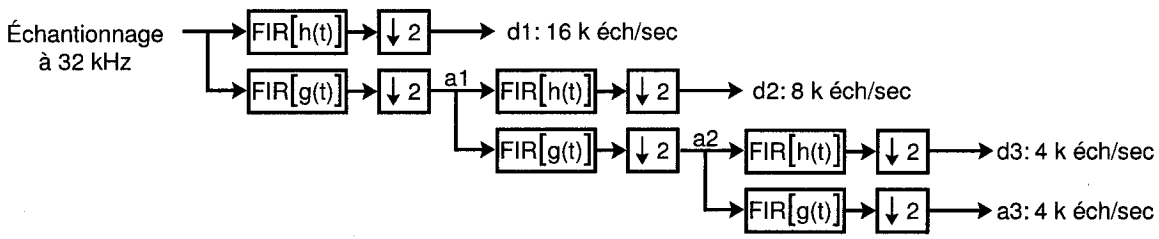


FIG. 2.5 Structure en arbre de la DWT

La figure 2.6 montre que pour chacune des échelles de mesure de l'ondelette mère, il y a différents niveaux d'activités. Ce qui indique que les coefficients associés à ces niveaux d'échelles rencontrent plus souvent des signaux y ressemblant.

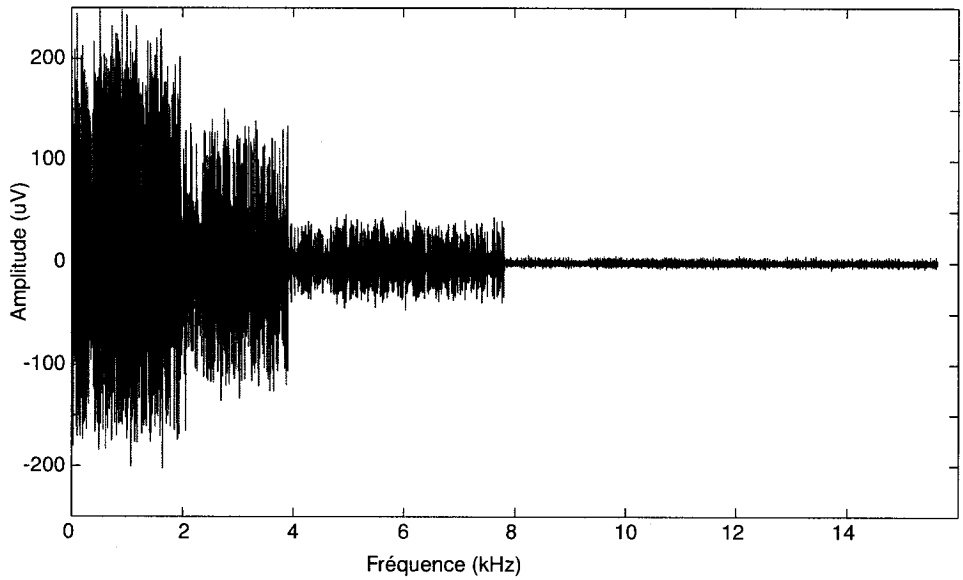


FIG. 2.6 Transformée en ondelettes

Pour faire de la compression dans le domaine des ondelettes, il faut absolument enregistrer la valeur de la donnée et la position du coefficient. Dans le cas contraire, la reconstruction risque de donner de pauvres résultats, par exemple dans le cas où seulement les premiers coefficients sont sélectionnés (les premières bandes seulement). L'application la plus connue avec les ondelettes est la norme JPEG2000 qui s'applique à la compression d'images.

La reconnaissance de formes est normalement accomplie par la convolution d'un signal de référence avec un signal de test. Le résultat de la convolution indique les similitudes entre les deux signaux. Le concept est de transformer l'espace de données d'entrée en un espace de propriétés et d'appliquer ensuite un seuil. La DWT est d'ailleurs un type de reconnaissance de forme. Une décision, le seuil, doit être prise après le filtrage pour déterminer si le signal d'entrée est considéré comme similaire.

2.3.4 Analyse par composants principaux

L'analyse par composants principaux (*Principal Component Analysis*-PCA) est utilisée en statistique pour réduire la dimensionnalité d'un ensemble de données. On l'appelle aussi KLT (*Karhunen-Loève Transform*) ou transformée de Hotelling. En fait, c'est une transformation linéaire qui consiste à déterminer une matrice de projection qui servira à obtenir un nouvel ensemble de données dont la variance sera maximale dans une dimension. Cette dimension est appelée le premier composant tandis que le second composant est la dimension de projection contenant la deuxième plus grande variance et ainsi de suite. Ceci permet de réduire la redondance du signal.

Contrairement à la FFT et la DCT, la PCA n'a pas de vecteurs de base fixes, la matrice de projection est entièrement dépendante de l'ensemble de données. La PCA sert pour l'affichage d'un ensemble de points à grande dimensionnalité ou pour un traitement ultérieur comme la classification. La compression se fait dans cette action de réduction de la dimensionnalité (Haykin 1998).

La construction d'un dictionnaire de vecteurs de projection est nécessaire pour chaque ensemble de données spécifiques. Cette méthode est un type de reconnaissance de formes. Un entraînement préalable du circuit doit être fait en utilisant des données pré-enregistrées ou en connaissant *a priori* la forme des potentiels d'action que le système échantillonnera. La figure 2.7 montre un exemple de vecteurs propres obtenus à partir de AP pré-sélectionnés. Les AP ont d'abord été alignés avec la méthode de seuillage absolu (voir section 2.5). On y voit que le cinquième composant principal commence déjà à être moins significatif et pourrait être associé au bruit ambiant.

Le principe consiste à faire un changement de base en utilisant les vecteurs propres obtenus par la décomposition en valeurs singulières ou à partir de la matrice de

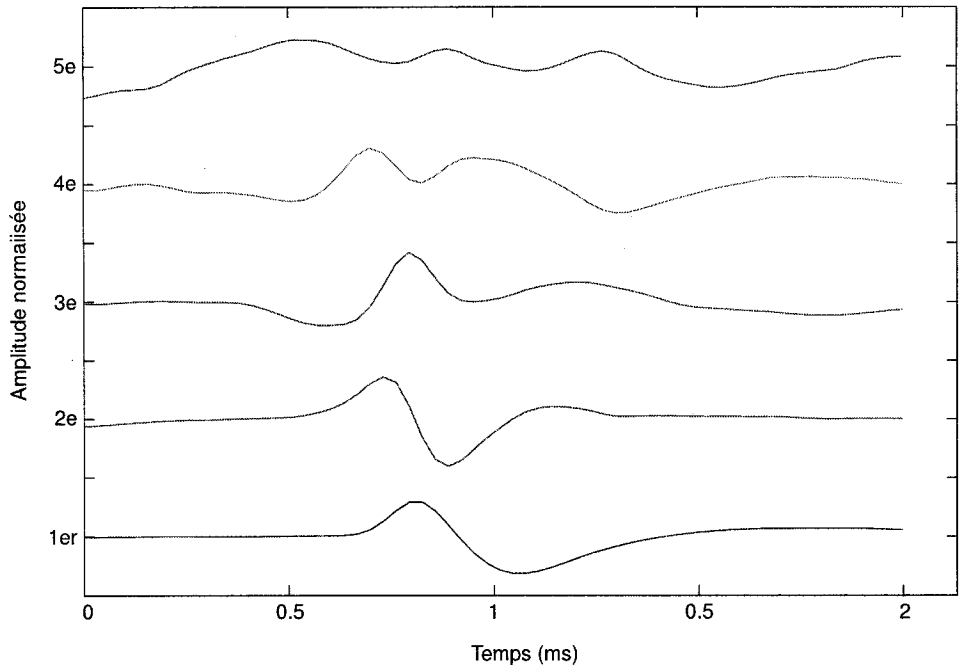


FIG. 2.7 Cinq composants principaux de AP pré-sélectionnés

covariance : la méthode classique. La covariance est une mesure en deux dimensions qui tient compte des différents échantillons pour en évaluer la variance. Nous obtenons ainsi une matrice de projection contenant les vecteurs propres, donc orthogonaux par définition. Les vecteurs propres les plus significatifs sont ceux dont la valeur propre associée est la plus grande. D'autres méthodes montrent comment le système peut s'adapter à un plus gand nombre d'échantillons pour se raffiner au fur et à mesure à l'aide d'un réseau de neurones (Costa et Fiori 2001).

Les avantages principaux de la PCA sont la précision et la versatilité de l'algorithme qui forme les vecteurs de base pour la projection en fonction des données. Il est donc statistiquement optimal. Par contre, le désavantage de cette technique dans un système temps réel (la PCA est utilisée habituellement pour la compression d'image) est que la précision est sujette à un problème de fenêtrage des échantillons.

2.3.5 Autres méthodes

Plusieurs autres méthodes ont été développées afin de traiter et de compresser des signaux neuronaux. Ces autres algorithmes utilisent, entre autres, les réseaux de neurones et des noyaux (*kernel*) spécifiques. Par exemple, Kadambe propose une méthode pour déterminer de façon adaptative l'ondelette mère la plus approprié pour un ensemble de données en particulier pour obtenir un classificateur et un compresseur "optimal" (Kadambe et Srinivasan 2005). Plusieurs méthodes semblent intéressantes, mais elles n'ont pas toutes la propriété d'une complexité limité se prêtant bien pour l'implémentation en logique matérielle.

Adjouadi utilise la transformée de Walsh qui utilise, à la manière de la DWT, la propriété de discontinuité. Plus les coefficients de la transformée de Walsh sont élevés, plus l'amplitude augmente rapidement dans le signal d'entrée (Adjouadi *et al.* 2004).

2.4 Les schèmes ou flots de données

Les algorithmes présentés dans ce chapitre ont pour but d'être placés dans un système temps réel. Cette section analyse les différentes procédures qui peuvent être utilisées pour influer sur le flot de données. L'analyse se limite à l'implémentation matérielle du système, ou jusqu'à la partie communication vers l'extérieur de l'implant. Lorsqu'il est question de traitement post-acquisition cela implique ce qui se situe en amont du lien de communication. Cela veut dire que le temps de calcul et la quantité de logique n'est plus un problème car cette action sera accomplie à l'aide de systèmes beaucoup plus puissants et essentiellement libre des contraintes d'implantation.

2.4.1 Acquisition sans traitement

Cette méthode est indubitablement la plus simple et la moins coûteuse en quantité de logique. Aucun traitement du signal n'est effectué, ou plutôt seulement du traitement post-acquisition. Comme discuté dans la section 2.2, cette méthode atteint rapidement le maximum de canaux possibles étant donné un débit maximal fixe.

Cette méthode est une des seules qui ne nécessite pas la formation de paquets de données et dont les données peuvent être acheminées dans le flot selon un processus très constant. Une unité DSP pourrait aussi être utilisée sans mémoire mais, dépendamment des algorithmes, la reconstitution chronologique pose parfois problème. Le mode par paquet devient utile dès qu'est envisagé l'usage de compression du signal.

Une variante de ce schème est l'utilisation de la compression par détection qui ajoute un comparateur pour la détection d'activité neuronale, comme à la figure 2.8a. Ce qui représente la situation de l'équation (2.2).

2.4.2 Acquisition avec traitement pré-détection

Cette méthode ajoute à l'entrée un module DSP dans le flot de données (figure 2.8b). L'intérêt de placer le module DSP au début de la chaîne est que son résultat peut servir ensuite pour faire la détection d'activité et de la compression, comme il apparaît efficace la majorité du temps (voir section 2.2).

De plus, étant donné l'emplacement du module DSP, il serait toujours possible de faire le traitement par filtrage analogique plutôt que numérique comme Simard le suggère (Simard 2005). L'avantage de faire cette étape analogiquement serait de minimiser la consommation de puissance ainsi que la surface d'implémentation.

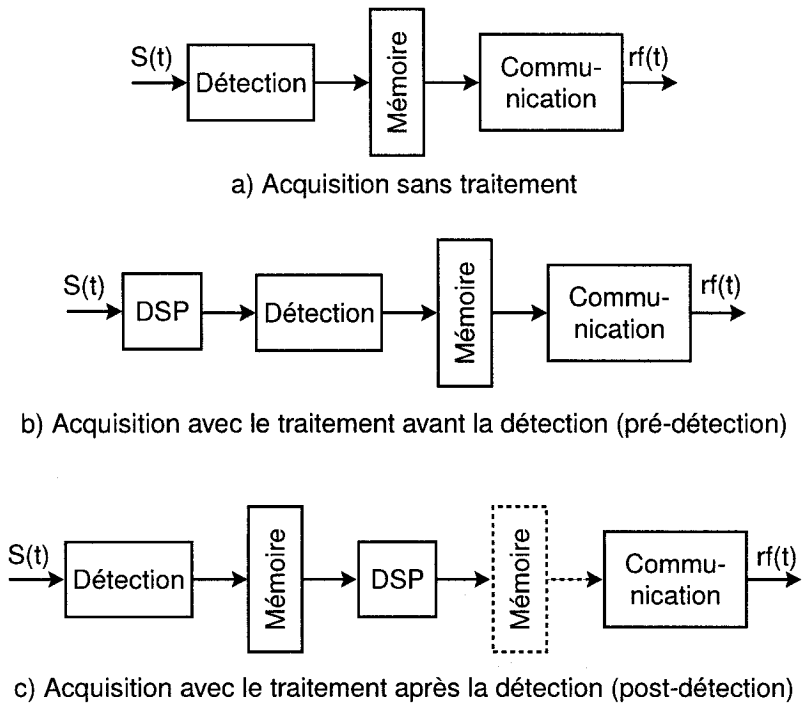


FIG. 2.8 Schèmes de flot de données

2.4.3 Acquisition avec traitement post-détection

Pour certains algorithmes, les données doivent être préalablement alignées comme pour la PCA, ainsi, la détection permet de sélectionner les échantillons valides. La compression en tant que telle est effectuée par le module DSP comme à la figure 2.8c. Le système peut-être optimisé pour limiter l'usage de mémoire, mais dans un système normalement conçu, un étage de mémoire est souvent ajouté pour faciliter l'intégration des différents modules.

Une variante du système pourrait utiliser un mélange des schèmes pré et post-détection (figures 2.8b et c). Un pré-traitement pourrait servir à la détection et le résultat pourrait être utilisé comme entrée de l'étage de compression utilisant un autre algorithme avant la transmission.

2.5 Les techniques de détection d'activité neuronale

Lorsqu'il s'agit de détection d'activité neuronale, plusieurs méthodes tentent de résoudre le problème le mieux possible. Pang énonce quelques points de comparaisons intéressants (Pang *et al.* 2003). Les abréviations dans les formules font référence au tableau 2.1.

La sensibilité (SEN) qui est une mesure le l'habileté du classificateur à détecter si un AP est survenu ou non : $SEN = \frac{VP}{VP+FN}$

La spécificité (SPE) qui est une mesure de l'habileté du classificateur à spécifier l'activité normale : $SPE = \frac{VN}{VN+FP}$

La sélectivité (SEL) qui est une mesure de l'habileté du classificateur à rejeter les mauvaises détections : $SEL = \frac{VP}{VP+FP}$

Un détecteur d'activité neuronale passe nécessairement par l'établissement d'un seuil. Ce qui différencie les détecteurs est le moyen choisi pour établir ce seuil. Si l'on considère que la détection d'un AP est un classificateur à une sortie à deux classes disant s'il estime qu'une occurrence est survenue ou non (positif ou négatif). Les possibilités de ce classificateur sont résumées par le tableau 2.1.

TAB. 2.1 Décision par seuillage

Occurrence	Décision : Potentiel d'action	
	VRAI	FAUX
VRAI	Vrai Positif (VP)	Faux Négatif (FN)
FAUX	Faux Positif (FP)	Vrai Négatif (VN)

Regardons ces possibilités du point de vue d'un système d'acquisition neuronale basé sur l'occurrence d'évènements. Un VP représente l'information qui a été justement transmise. Un FP représente l'information qui a été inutilement transmise, ou plutôt qui n'apporte pas plus information que les caractéristiques du bruit. Un FN représente

de l'information perdue car un AP a eu lieu mais le détecteur ne l'a pas vu. Finalement, un VN représente la situation normale ; sur un canal de transmission cortical, il ne devrait y avoir des AP que de temps en temps.

Faisant suite aux travaux de Obeid qui compare différentes techniques comme les seuils simple, par l'analyse d'énergie et par reconnaissance de forme. Dans tous les cas, une réserve doit être mise sur l'efficacité et la précision des algorithmes de détection (Obeid et Wolf 2004). Wood a prouvé que chacune des méthodes utilisées possède ses lacunes et que même avec une classification manuelle, les experts en neurophysiologie n'arrivent pas à s'entendre sur l'évaluation d'un ensemble de test (Wood *et al.* 2004).

2.5.1 Détection par seuil simple

Comme son nom l'indique, cette méthode est simple du fait qu'elle entre dans la catégorie des schèmes d'acquisition sans traitement de données. Il est possible de séparer les seuils simples en deux catégories, soient la détection d'un front montant ou descendant de tension appelés fronts positifs et fronts négatifs, ou les deux à la fois appelé front absolu parce que la valeur absolue du signal est utilisée comme entrée du détecteur de front montant. Si une connaissance *a priori* du SNR est disponible, le choix du front montant ou descendant peut être fait, par contre il est plus prudent d'utiliser le front absolu car il permet d'appliquer le seuil au plus grand SNR entre le positif et le négatif.

Cette technique est la plus répandue car elle répond habituellement aux demandes expérimentales (Lewicki 1998). Par contre, elle est biaisée parce qu'elle n'identifie que les AP avec les plus grandes amplitudes et pourrait ne pas être représentative de la population globale de neurones, il y a donc un compromis entre FP et FN. De plus

elle n'est pas appropriée pour la classification ultérieure et demande un ajustement minutieux pour l'isolation d'un type de neurone, sans garantie de succès. Lewicki suggère que dans la mesure du possible les données brutes devraient être enregistrées pour un traitement post-acquisition.

Le problème de cette technique est la sensibilité au bruit. D'autant plus que le SNR est souvent mauvais, le seul moyen d'augmenter la sensibilité est de diminuer le seuil pour diminuer le taux de FN, ce qui diminue la sélectivité par la même occasion en augmentant le taux de FP. La figure 2.9 montre l'effet du seuil simple sur l'ensemble de référence ; les échantillons ont été ensuite alignés en fonction du leurs maximums.

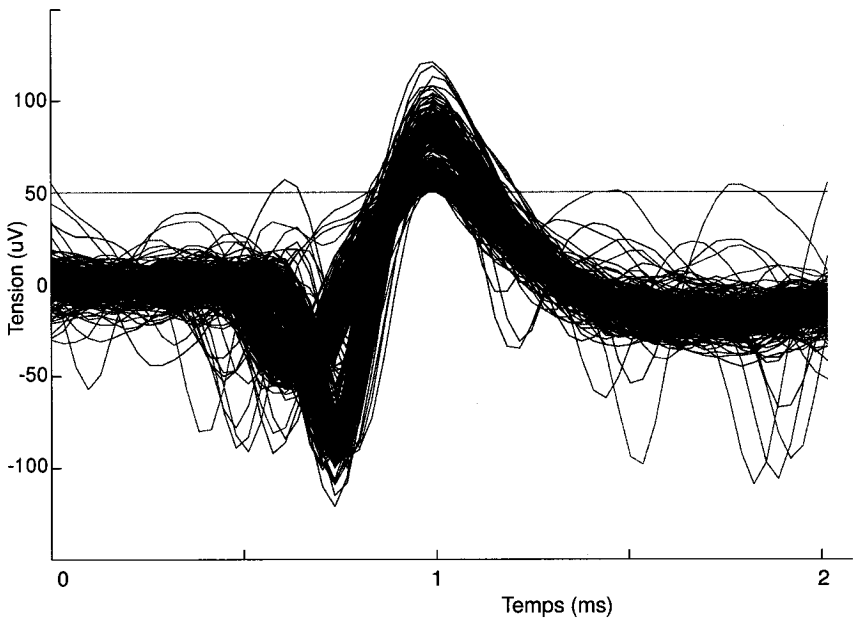


FIG. 2.9 Seuil simple avec détecteur de front montant et alignement

2.5.2 Détection par mesure d'énergie

La méthode NEO (*Nonlinear Energy Operator*) est aussi basée sur le seuil simple mais avec un calcul instantané de l'énergie au préalable (Obeid et Wolf 2004). Aussi

appelée *Teager Energy Operator*, cette méthode avait d'abord été caractérisée par Kaiser et elle donne une approximation de l'énergie avec quelques échantillons seulement et s'utilise normalement avec $1 \leq \delta \leq 4$ et se définie (Kaiser 1990) à l'équation 2.18.

$$NEO[X(n)] = x^2(n) - x(n+\delta)x(n-\delta) \tag{2.18}$$

Une démonstration de l'application des seuils est montrée à la figure 2.10.

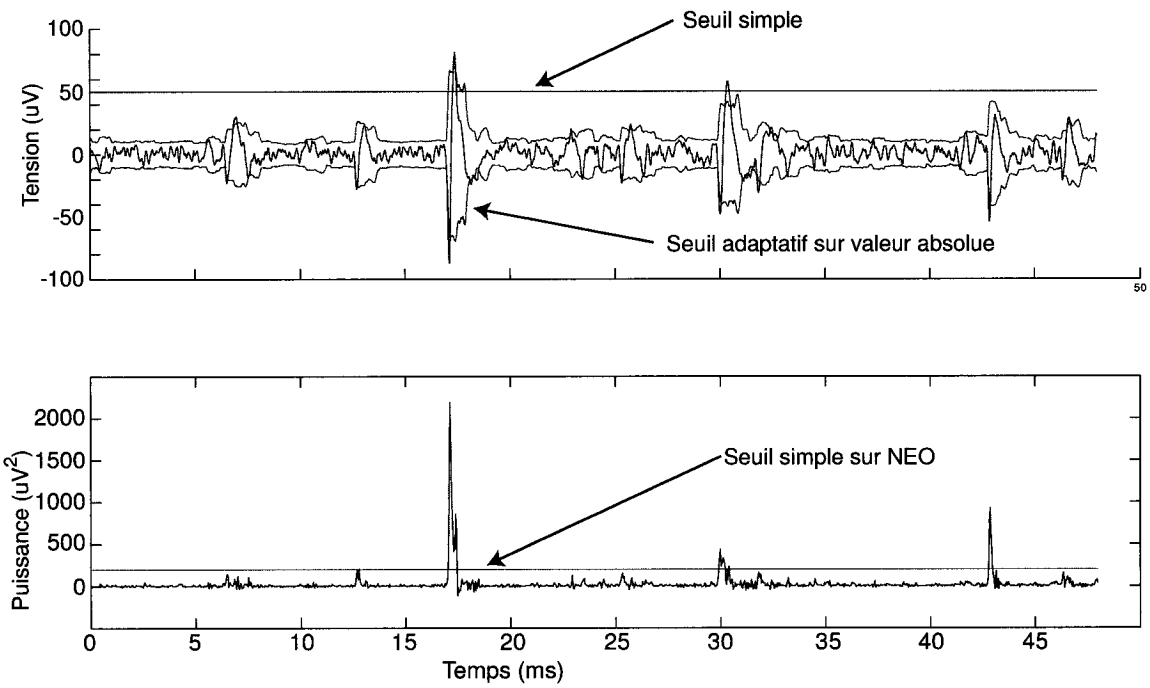


FIG. 2.10 Exemple d'application des seuils

Le peu de logique nécessaire pour faire ce calcul est intéressant pour l'implémentation matérielle ; Obeid l'a d'ailleurs réalisé dans un système BCI portable avec l'université Duke. Son analyse fait suite aux résultats prometteurs de Kim avec le NEO qui obtenait une détection quasi parfaite avec un SNR près de $0dB$ suivi d'un système

de classification par réseau de neurones (Kim et Kim 2000).

2.5.3 Détection par extraction de propriétés

Cette méthode s'occupe seulement de la forme des AP comme la hauteur versus la largeur ou l'amplitude minimum versus maximum (comparaison "peak-to-peak"). Le but est de déterminer le minimum de propriétés nécessaires pour prendre une décision acceptable. Il s'agit donc de déterminer des ensembles représentant des AP par des techniques de classification. Lewicki présente cette technique mais constate qu'elle aussi doit faire un compromis entre les FP et les FN mais cette fois-ci en deux dimensions comparativement au seuil simple (Lewicki 1998). L'utilité temps-réel de cette méthode dépend de la stabilité de l'environnement. Elle semble mieux adaptée à un traitement post-acquisition qu'à un traitment temps réel.

2.5.4 Méthodes adaptatives

Le but de cette méthode est de s'adapter au bruit ambiant pour configurer la valeur du seuil. Elle peut être intégrée avec ou sans pré-traitement des données. Si le signal sans traitement est considéré, la puissance du bruit ambiant est estimée par la variance du signal. L'estimation est valide puisque les AP ne surviennent qu'environ 1% du temps, le reste du temps ce n'est que du bruit. Le bruit peut donc s'estimer avec à l'aide de la variance du signal. Si le seuil était fixé à deux fois la déviation standard ($SD[X] = \sqrt{VAR[X]}$), 95% du signal serait rejeté selon une distribution normale du signal représentant majoritairement du bruit.

Harrison propose une méthode simple qui utilise un détecteur de "duty cycle" comme boucle de rétroaction ; si le détecteur varie trop souvent, cela veut probablement dire

qu'il détecte du bruit et non un AP. Il tient compte de la durée d'un AP afin de ne pas annoncer une détection si l'impulsion est trop courte (Harrison 2003). Ses résultats démontrent que le seuil obtenu varie entre une et deux fois la déviation standard. Pouzat utilise une méthode similaire mais il utilise jusqu'à $a = 3.5$ comme facteur multiplicatif pour la configuration du seuil (Pouzat *et al.* 2002).

$$Seuil_{adapt} = a \cdot SD[X] \tag{2.19}$$

Une variante de cette méthode consiste en un mélange du seuil simple et d'un facteur relié à la variance. Ceci permet plus de latitude expérimentale.

$$Seuil_{adapt(cst)} = SeuilSimple + a \cdot SD[X] \tag{2.20}$$

2.5.5 Transformée discrète en ondelettes

La transformée en ondelettes ressemble beaucoup à un type de reconnaissance de forme : elle se fait par convolution afin de trouver des similitudes avec le filtre d'ondelette mère. La DWT a de différent qu'elle utilise le même filtre pour plusieurs échelles de comparaison. Un AP peut être considéré comme une courte impulsion localisée. La DWT permet de cibler une échelle spécifique pour la détection d'activité neuronale.

Cette méthode permet de débruiter le signal tout en faisant une extraction des caractéristiques à chacune des résolutions permettant ainsi de classer les signaux en même temps. Si le but est seulement de détecter la présence d'un AP, cette méthode demande peut-être trop de calculs. Jumelée avec un algorithme de compression, elle devient intéressante du point de vue optimisation de la bande passante (Simard 2005, Hulata *et al.* 2000).

Plusieurs implémentations de ce type ont été tentées. Letelier utilise ce classificateur en utilisant la quantification d'énergie retrouvée dans les régions fréquentielles spécifiques (Letelier et Weber 2000). Il utilise une méthode d'acquisition avec traitement post-détection (figure 2.8c). Elle consiste à choisir le décalage qui permet de maximiser la variance sur certains coefficients associés à une forme de AP. Ils ont toutefois écarté le problème de la superposition d'impulsions en générant eux-mêmes les signaux de tests et connaissant la position exacte des AP. Ils réussissent à classifier l'occurrence de AP avec un meilleur taux que par PCA ou par réduction de dimensionnalité obtenu par la FFT. Ils tirent profit d'une fonction d'ondelette mère compacte comparativement aux sinus ou cosinus infinis.

Hall introduit l'idée de choisir un seuil par groupe de coefficients contrairement à un seuil comparé terme par terme comme dans la majorité des implémentations. Il argue qu'un estimateur par groupe est moins biaisé et réagit plus rapidement aux changements fréquentiels soudains. De plus ils auraient augmenté l'adaptabilité et réduit la MSE du fait qu'un groupe de coefficients contient aussi l'information de son voisinage, c'est-à-dire l'information d'un intervalle plutôt que spontanée (Hall *et al.* 1997). Il suggère d'utiliser les seuils terme par terme (qualifié d'universel) et par groupe de coefficients :

$$Seuil_{uni} = SD\sqrt{2\log N} \tag{2.21}$$

$$Seuil_{grp} = SD \tag{2.22}$$

où N est le nombre d'échantillons utilisés pour le calcul de l'approximation de la variance. Ce qui peut-être associé à la méthode adaptative présentée plus tôt.

CHAPITRE 3

IMPLÉMENTATIONS MATÉRIELLE ET LOGICIELLE

3.1 Introduction

Dans ce chapitre, le système numérique est décrit en détails. Ce projet étant dédié à une implémentation matérielle, les points discutés suivent une progression logique du contexte à l'architecture en passant par les problématiques rencontrées.

Premièrement, une brève description du système dans sa globalité est présentée à la section 3.2, afin de situer le contrôleur numérique dans le ENG complet proposé par l'équipe Cortisens. Deuxièmement, la section 3.3 présente les problématiques dues aux contraintes physiques et aux limites des interfaces proposées ; il y sera question de la gestion des horloges, de l'utilisation de blocs mémoire ainsi que du compromis entre l'importance du traitement versus la complexité d'implémentation. Finalement, l'architecture du contrôleur numérique est montrée à la section 3.5. Une approche par modules fonctionnels a été appliquée afin d'évoluer étape par étape dans la mise en oeuvre du prototype. Le système présenté dans cette section utilise la DWT afin d'effectuer la détection neuronale et la compression des données.

3.2 Le projet Cortisens

Le capteur implantable que l'équipe Cortisens propose est un système minimalement invasif dédié à l'acquisition intracorticale dans le but de prendre des mesures *in vivo*. Le système est constitué d'une matrice d'électrodes connectée à un système actif

composé d'un étage mixte d'amplification, de filtrage et d'échantillonnage, suivie d'un contrôleur numérique de gestion et de traitement des données, et finalement d'un lien de communication sériel sans-fil. Gosselin présente une première version du système (Gosselin *et al.* 2004a). Une seconde version composée d'un assemblage multi-puces est en voie de fabrication et devrait être testée prochainement (figure 3.1).

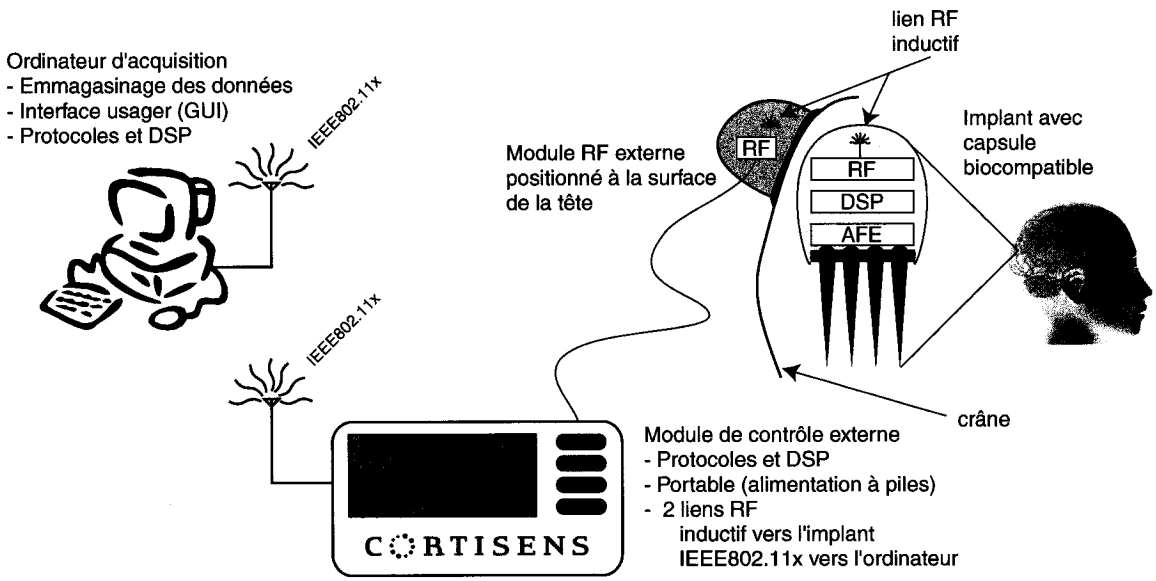


FIG. 3.1 Système conçu par l'équipe Cortisens

3.2.1 Interface Électrodes-Tissu

La matrice d'électrodes est faite de métal et est divisée en groupe de seize pointes afin de suivre une approche modulaire. L'espacement centre à centre des électrodes est d'environ $400\mu m$. La matrice est fabriquée avec la technique d'usinage par décharge électrique (EDM) et les pointes à découvert doivent avoir une dimension inférieure à $20\mu m$, afin de capter adéquatement les AP (Pigeon *et al.* 2003). Un exemple de la matrice tiré du mémoire de Pigeon est présenté à la figure 3.2 (Pigeon 2004).

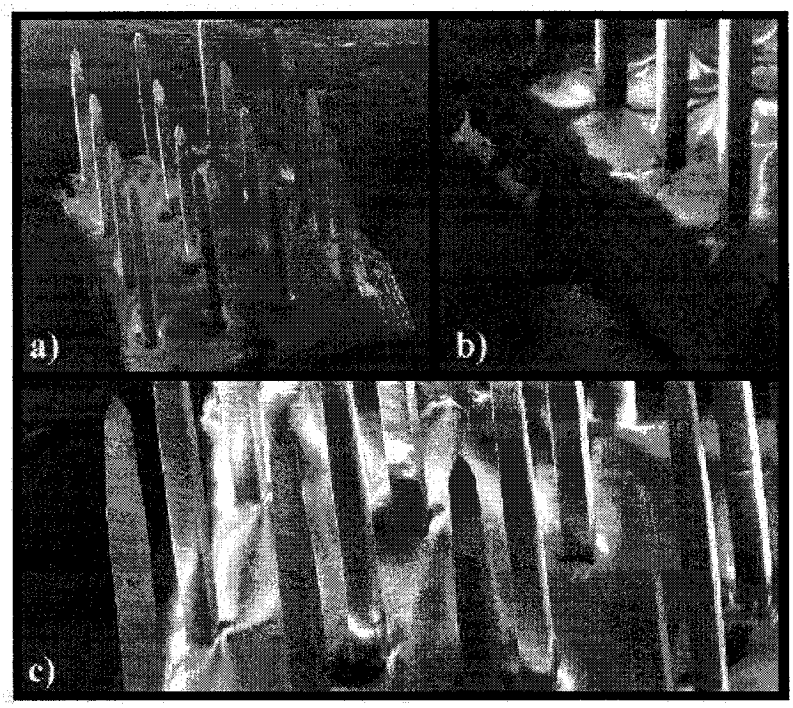


FIG. 3.2 Matrice d'électrodes fabriquée par Polystim, a) vue globale, b) découpage, c) pointes

Le but de la matrice d'électrodes est de faire le lien entre l'interface biologique, le contact avec le cortex et l'étage d'amplification analogique.

3.2.2 Amplification analogique et échantillonnage

L'étage mixte du système comprend l'électronique nécessaire au conditionnement et à l'échantillonnage des signaux : c'est l'interface d'entrée (*Analog Front End*-AFE). Une approche de traitement parallèle est adoptée avec une architecture identique pour chaque canal. Un canal est composé d'une série de filtres avec une grande impédance à l'entrée (connexion avec l'électrode) et un bon taux de rejet en mode commun ; le circuit actif ne doit pas influencer le comportement électrique aux alentours de

l'électrode. L'AFE est conçu pour consommer le moins de puissance possible tout en réduisant le bruit (Gosselin *et al.* 2004b). L'étage de préamplification utilise une technique de Chopper pour éliminer le bruit $1/f$ présent dans les signaux à très basses amplitudes ($\approx 100\mu V$).

La récupération des données suite à l'étage d'amplification se fait en deux phases. Premièrement le signal est converti et placé dans un registre à décalage un bit à la fois pour obtenir un mot binaire de 8 bits. Deuxièmement, le signal est lu en parallèle à partir du registre à décalage et les échantillons de chaque canal sont sélectionnés un à un à l'aide d'un multiplexeur activé par balayage (*Time Division Multiplexing*-TDM), comme à la figure 3.3.

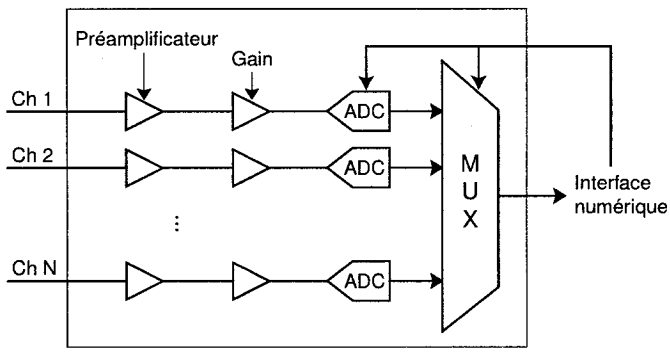


FIG. 3.3 Filtrage analogique et amplification

Le but de la section mixte du système est de faire l'interface entre les électrodes et le contrôleur numérique de l'implant. Il doit respecter les contraintes de consommation de puissance, de surface d'implémentation physique et de filtrage du bruit.

3.2.3 Contrôleur numérique

Le contrôleur permet la gestion globale du système implanté et est représenté par le bloc DSP à la figure 3.1. Il implémente un protocole simple donnant la possibilité de

configurer chacun des canaux individuellement. Il transmet les données compressées par événement, si et seulement si un AP est détecté. Une description plus ample est faite à la section 3.5.

3.2.4 Communication sans-fil

Ce module reçoit les données du contrôleur numérique et les transmet par une antenne. Il gère aussi la réception pour la configuration du système. C'est aussi ce module qui génère la fréquence de base du système, soit par recouvrement ou génération d'horloge dans un souci de compatibilité d'interface.

Le but de cette section est de faire l'interface entre le contrôleur numérique interne et le contrôleur numérique externe par un lien bidirectionnel sans-fil. À moins que le système soit alimenté par une batterie, ce qui n'est pas souhaitable, le module de communication sans fil a la responsabilité de fournir l'énergie à l'implant par un procédé inductif.

3.2.5 Module externe

Le but du système est d'être portable par le sujet afin qu'il puisse vaquer à ses occupations normales et il doit donc être autonome. Il est constitué d'un couple transmetteur-récepteur complémentaires pour la communication sans-fil avec l'implant, d'une unité de traitement pour permettre une gestion temps réel de l'implant ainsi qu'un deuxième lien de communication sans-fil, préférablement conforme à une norme répandue comme la IEEE 802.11x[1].

[1] *http ://grouper.ieee.org/groups/802/11/* standard adopté depuis 1999

3.2.6 Assemblage et prototypes

Le système conçu est composé de plusieurs puces intégrées assemblées et connectées ensemble, le but étant de ne pas dépasser la surface d'une matrice d'électrode en surface silicium pour permettre le montage de plusieurs matrices côte à côte. Une implémentation future pourrait inclure tous les circuits intégrés dans une seule puce et minimiser ainsi l'espace utilisé par les étages de protection d'entrée/sortie. La première version complète du système superposera les puces (*Die Stacking*) et les interconnectera par des fils de liaison (*Wire Bonding*).

Le sujet de ce mémoire étant exclusivement basé sur le circuit numérique, le prototypage s'est donc effectué sur une plateforme de développement rapide avec FPGA (*Field Programmable Gate Array*).

3.3 Problématiques du contrôleur numérique

Le module numérique étant situé au centre d'un système plus complexe, il est contraint de chaque côté par ses interfaces. De plus, de par sa nature, il doit permettre une grande flexibilité tout en étant robuste. Les choix de design font suite à une analyse de compromis et les différents points d'importance sont explicités dans cette section.

3.3.1 Interface avec le module RF

N'oublions pas le but premier du projet consistant à faire un système implantable donc par définition à faible consommation de puissance. Dans cet optique, deux possibilités ont été évaluées pour la réalisation d'un BUS de communication : soit le transfert sériel

ou le transfert parallèle.

L'estimation de la puissance consommée est une recherche complète en elle-même. En effet, faire une approximation sur un circuit avant sa réalisation n'est pas facile. Pourtant, certains paramètres peuvent être évalués pour en venir à la conclusion qu'une communication sérielle était plus appropriée pour notre application. Il y a le facteur de puissance statique directement liée à la quantité de logique utilisée, ainsi que la surface silicium nécessaire. Plus le circuit est gros, plus il consommera, comme un raisonnement de base le laisse supposer. Par contre, la dissipation de puissance dynamique est la plus difficile à évaluer. L'entropie du message et la dissipation de puissance son directement liées aux patrons présentés sur le BUS. Une approximation acceptable expliquée par Ferrandi considère l'alimentation (Vdd), les capacités parasites de ligne (C) et la densité de transition (D) qui est fonction de la fréquence et de l'entropie des données (Ferrandi *et al.* 1998).

$$P_{moy} = \frac{1}{2} V_{dd}^2 \sum C_i D_i \tag{3.1}$$

La densité de transition (D) est directement proportionnelle à la fréquence de transmission. Donc si le lien est parallèle, une fréquence minimale est nécessaire pour transférer l'information dans un laps de temps prédéfini en comparaison avec un lien sériel. Pour les capacités parasites de ligne (C), l'approximation devrait être faite en fonction de la technologie utilisée mais dans notre cas les deux implémentations possibles seraient destinées à la même technologie ce qui nous permet d'écarter ce facteur pour le choix du BUS.

Il ne reste donc plus qu'un seul facteur important pour l'évaluation de la capacité du système qui est la surface physiquement nécessaire pour effectuer la tâche. Si nous

considérons que pour une communication parallèle, un tampon d'entrée/sortie est nécessaire de chaque côté du système pour chacun des fils physiquement routés, il va alors de soi qu'un nombre minimal de tampon est requis pour une communication sérielle. Ce qui amène à considérer l'implémentation matérielle du système multi-puces. Chacune des entrées/sorties des puces utilise une surface non négligeable pour la gestion anti-décharges électrostatiques et les amplificateurs de puissance en sortie. Donc dans le cas d'un système multi-puces, l'approche sérielle est favorisée tandis que dans le cas d'une intégration ultérieure des différents modules, une approche parallèle serait plus appropriée.

3.3.2 Interface avec les convertisseurs analogique numérique

À ce point, les considérations les plus importantes deviennent les contraintes du circuit analogique et son interconnexion ainsi que le traitement du signal à l'étage d'entrée numérique. Est-ce qu'un CAN est utilisé par canal ou bien un seul est utilisé pour plusieurs canaux ? La question s'applique aussi au nombre d'unités de traitement prévues. Donc la nouvelle question est : doit-on pipeliner les données ?

Le pipelinage des données est une méthode qui consiste à entrer les données de plusieurs sources dans une même chaîne de traitement en série (*stream processor*) afin d'augmenter le débit de sortie. Cette technique est souvent utilisée pour des applications de traitement numérique du signal comme sur les cartes vidéo des ordinateurs. La fréquence de l'horloge qui cadence le système doit être élevée pour supporter le surplus d'information mais les unités de logique comme les additionneurs et les mutiplieurs peuvent être partagés.

Par contre, le pipelinage a l'inconvénient d'ajouter un délai de propagation au système,

une latence, en comparaison avec un autre sans pipelinage. De plus, le système nécessite souvent un design plus complexe et, parfois même, plus gros car les ressources ne peuvent pas être réutilisées ; les résultats doivent être emmagasinés sous forme de mémoire contextuelle et passés à chaque étape subséquente.

Le choix de l'interface pipeliné

Encore une fois, en considérant l'implémentation multi-puces du systèmce ENG complet, le nombre d'entrées/sorties devient très critique. Dans le design actuel, l'espace silicium permis nécessite la ségrégation des modules. Dans un système entièrement intégré les architectures pourraient être fusionnées afin d'optimiser davantage le flot de données. Les deux façons de pipeliner les données pour notre interface entre le module numérique et les CAN sont schématisées dans la figure 3.4. Comme décrit dans la section 3.2, nous avons choisi l'approche par usage d'un multiplexeur numérique avec un CAN par canal.

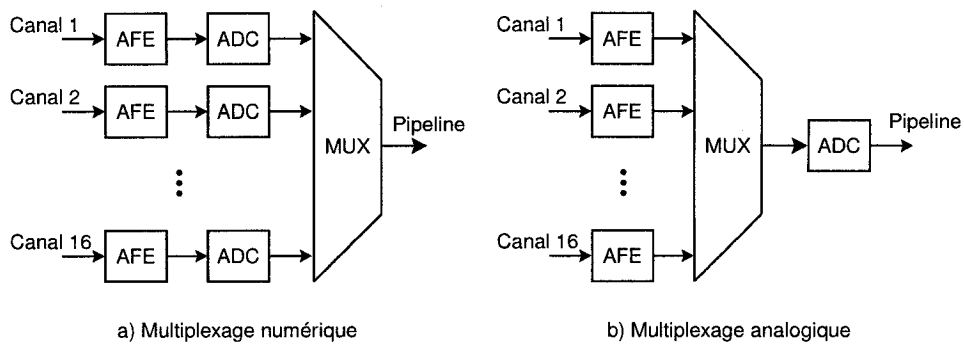


FIG. 3.4 Multiplexeur analogique vs numérique

Le choix de pipeliner entre dans la lignée des compromis de design pour une optimisation modulaire. Dans l'optique où une unité de traitement est prévu dans le schème de données et qu'une maximisation du débit de sortie est désirée, le pipelinage est tout désigné pour atteindre ces buts. L'interface ainsi déterminée facilitera la réutilisation des designs car chacun des modules peut être extrait, modifié et réintégré au système

pour la réalisation d'une nouvelle version. Le choix de l'interface pipelinée offre une plus grande latitude aux choix de design analogique et permet l'usage d'une unité de traitement.

3.3.3 Gestion des horloges

Le choix de l'horloge est une étape importante d'un design électronique. Sa génération et sa distribution dans le circuit intégré sont les deux principales fonctions à être remplies par les circuits d'horloge. Les caractéristiques influant la distribution des horloges sont : le temps d'activation avant le front (*setup time*), le temps de maintient après le front (*hold time*) et le temps de propagation. Ces effets de délais sont appliqués par les outils de design numérique afin de s'assurer que les contraintes soient respectées. Le système proposé possède plusieurs horloges à différentes fréquences en même temps pour différentes parties du circuit : l'acquisition parallèle et la communication sérielle. Ce qui implique des problèmes potentiels comme le partage d'information d'un domaine d'horloge à l'autre.

Le passage de domaines d'horloge consiste à utiliser un signal généré par une bascule activée par une horloge dans un circuit aboutissant à une bascule activée par une autre horloge. Si les horloges ne sont pas synchronisées entre elles, une métastabilité devient fort probable et le comportement logique imprévisible. Grosso modo, il existe deux façons d'assurer adéquatement le partage de signaux. Premièrement, en synchronisant les domaines d'horloges, les fronts (montants et/ou descendants) des horloges doivent arriver en même temps. Ce qui peut être obtenu en choisissant des horloges avec un commun multiple ou bien en modifiant la rationalité d'une horloge par rapport à l'autre (*duty cycle*) pour que les délais critiques ne soient jamais dépassés. La modification du *duty cycle* est une tâche fastidieuse et non garantie de fonctionne-

ment pour tous les circuits comme les cas utilisant des horloges complémentaires sans recouvrement. De plus, une horloge à très haute fréquence est habituellement utilisée pour ce genre de circuit. Deuxièmement, une communication par états asynchrones (*handshaking*) peut être utilisée. Le *handshaking* est une méthode répandue et souvent obligatoire pour des systèmes devant respecter des protocoles prédéterminés et incompatibles. Par contre, cela implique la rétention du BUS de communication lors du transfert d'un domaine d'horloge à l'autre et fonctionne plutôt selon l'analogie d'une pompe qui se charge et décharge comparativement à un processus fluide avec la synchronisation des horloges.

La synchronisation des horloges par horloges avec commun multiples est vraiment la plus simple à designer. C'est pourquoi cette méthode a été préférée aux autres. Le choix des horloges pour le présent système ENG se résume donc par le tableau 3.1.

TAB. 3.1 Horloges impliquées dans le système

Noms	Fréquences
clk_ech	32 kHz
clk_adc	8 clk_ech
clk_mux	16 clk_ech
clk_dwt	64 clk_ech
sclock	16 MHz = 2^9 clk_ech

La figure 3.5 présente une façon facile et robuste de générer des horloges avec un compteur libre et une horloge de référence. Il est bien de noter que l'horloge d'entrée ne sera pas en phase avec les horloges de sortie mais le délai demeurera constant dans le temps. Avec une fréquence de 16.6 MHz, un diviseur avec compteur 9 bits serait suffisant pour obtenir une fréquence de 32.4 kHz.

Les quatre sorties clk_mux_out, clk_adc_out, clk_ech_out et clk_dwt_out (non illustrée) sont générées à partir du compteur libre actif sur le front montant de l'horloge de

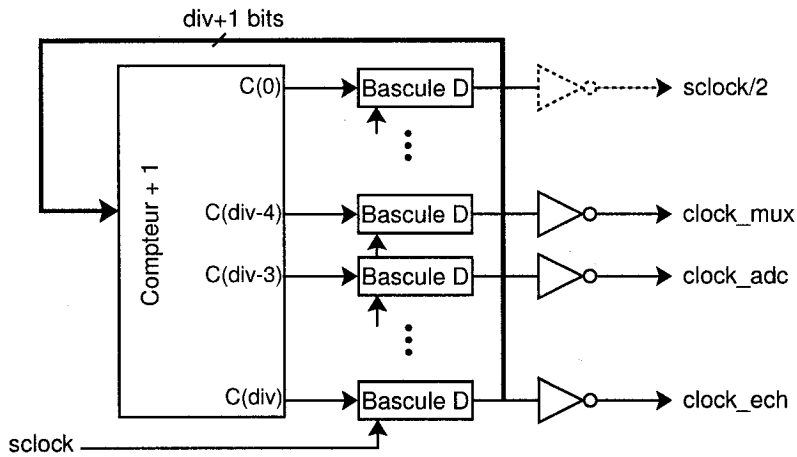


FIG. 3.5 Génération d'horloges avec un compteur libre

référence (*sclock*). Les horloges de sortie sont synchronisées par les bascules de sortie du compteur et inversées pour ajuster les fronts montants des horloges avec les compteurs globaux avec remise à zéro comme le compteur *mux_cnt* contrôler par *clk_mux* qui contrôle l'interface avec les CAN. Ainsi, tous les compteurs à différentes fréquences sont en phase les uns aux autres.

Un composant inverseur spécial doit être employé pour synchroniser les fronts montants avec le compteur global. L'inverseur symbolise à la fois l'inverse du signal pour la synchronisation et le tampon générateur de courant. Un signal d'horloge est généralement routé vers plusieurs composants, d'autres bascules en occurrences. Il doit pouvoir fournir plus de courant pour alimenter la ligne dû aux résistances et aux capacités parasites de ligne élevées (grand "fanout").

Tous les signaux d'horloge provenant du générateur sont acheminés vers l'extérieur du circuit pour faire les interfaces multi-puces prévues. En utilisant ces horloges plutôt que le signal interne à la puce, cela garantie que les délais seront les mêmes pour chacune des puces ; ce qui ne serait pas le cas autrement dû au délai introduit par les tampons d'entrées/sorties. Le délai entre le front montant de l'horloge de référence

et l'horloge générée est $T_{horloge}$ dans le tableau 3.2. Ce tableau montre aussi les délais typiques rencontrés dans le système comme présentés à la figure 3.6. Les délais d'entrée ou de propagation ($T_{e/p}$) dépendent du signal routé et peuvent varier ; normalement, le délai maximum est considéré, pareillement pour les délais de sortie (T_{sortie}) et de logique ($T_{logique}$). Dans le cas d'un signal d'horloge, l'arbre de propagation est considéré puisqu'il est conçu de façon balancée afin de synchroniser les fronts dans tout le circuit. Pour une approximation réaliste des délais un calcul est fait pour chaque domaine d'horloge. Les délais dus aux bascules D sont les mêmes pour chaque domaine puisqu'ils varient seulement en fonction de la technologie qui est le CMOS 0.18μm dans le cas du projet Cortisens.

TAB. 3.2 Délais typiques

Symboles	Causes	Description
$T_{e/p}$	f(distance) & f(cmosp18)	Délai dû aux amplificateurs d'entrée et/ou au délai de propagation
T_{hold}	f(cmosp18)	Temps nécessaire pour la stabilité des bascules
T_{su}	$T_{e/p} + T_{hold}$	Délai total d'entrée (*setup*)
$T_{logique}$	f(distance) & f(complexité)	Délai de logique combinatoire
T_v	f(cmosp18)	Temps avant que la donnée soit valide après un front montant
T_{sortie}	f(distance) & f(cmosp18)	Délai dû aux amplificateurs de sortie et/ou au délai de propagation
T_{co}	$T_v + T_{sortie}$	Délai total de sortie (*clock to out*)
$T_{horloge}$	$T_{e/p} + T_{co}$	Délai entre l'horloge de référence et l'horloge générée

Cette technique avec boucle de rétroaction externe appelée en anglais "*loopback*" offre la possibilité à l'usager du circuit intégré de prendre un générateur d'horloge externe en cas de défaut. Le système offre ainsi une plus grande flexibilité. Ainsi, les outils de

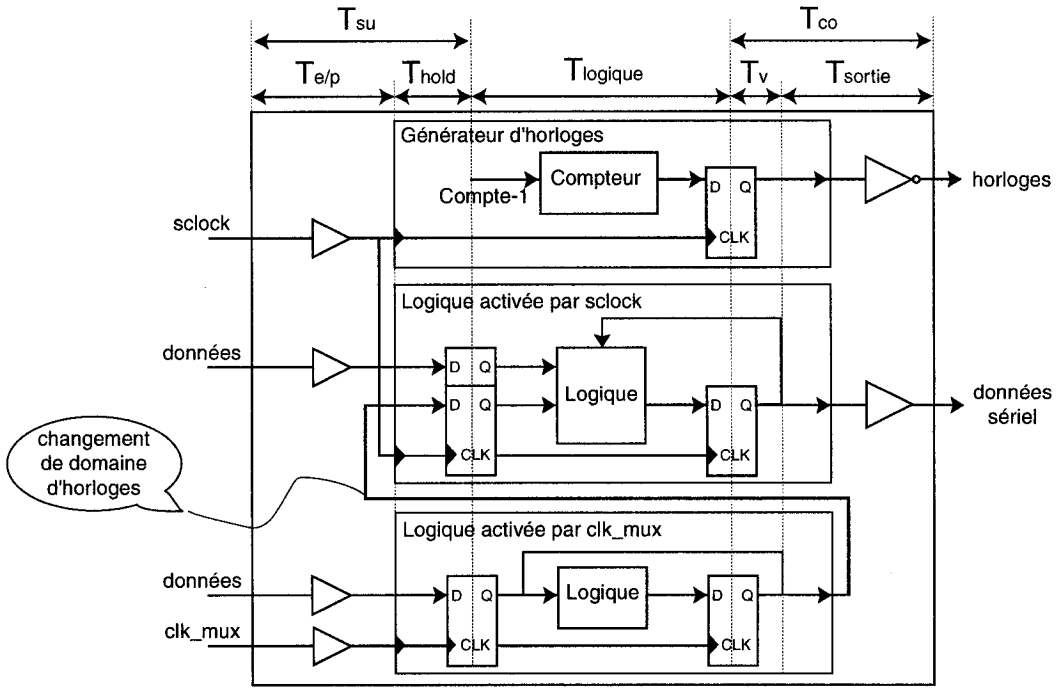


FIG. 3.6 Description des délais par rapport à la génération d'horloges

synthèse et de placement automatique sont plus aptes à s'assurer que les délais sont les mêmes pour chacun des modules car ils sont tous synchronisés sur le même signal de référence.

Puisque le système proposé a pour but de minimiser la puissance consommée, une excellente technique est le masquage de l'horloge, ou *clock gating*. Cela consiste à interrompe la propagation de l'horloge sous certaines conditions comme pour la désactivation d'un canal. Certaines parties de logiques d'un système sont mutuellement exclusives et peuvent être interrompues si le système n'en a pas besoin à cet instant. Ce qui réduit considérablement la puissance dynamique du système étant donnée que toutes les bascules branchées à ce sous-réseau de distribution d'horloge conserve le même état tant et aussi longtemps que l'horloge n'est pas rétablie. Le problème avec ce genre de système est que habituellement un délai est nécessaire pour

la réactivation du sous réseau. L'horloge ne peut donc pas être interrompue inopportunément, mais seulement dans les cas stratégiques. La version actuelle du système n'inclue pas ce genre de gestion bien qu'elle en tirerait certainement un bénéfice. Ce pourrait être une option intéressante pour un futur design.

3.3.4 Le choix des cellules mémoire

La mémoire étant d'une grande utilité et malheureusement gourmande en surface, il est important d'optimiser cette ressource. Il a été choisi d'utiliser des cellules de type SRAM (*Static Random Access Memory*) plutôt que DRAM (*Dynamic RAM*). Bien que moins compact, la SRAM consomme moins d'énergie à basses fréquences et permet des accès directs à la mémoire. Les accès directs à la mémoire permettent d'y accéder avec deux ports indépendants.

La mémoire à double port permet de lire et d'écrire à l'aide de deux contrôleurs indépendants. Ce qui est très pratique pour partager des données d'un module logique à un autre. C'est d'ailleurs la raison pour laquelle elle est utilisée pour la réalisation de ce contrôleur numérique. Le port d'écriture adresse et conduit ses données à une fréquence donnée tandis que le port de lecture adresse et reçoit ses données à sa propre fréquence. La principale précaution à prendre pour éviter les conflits est qu'une lecture et une écriture ne doivent jamais être effectuées à la même adresse au même moment, sinon la donnée lue est considérée comme indéterminée. Ce qui implique qu'un gestionnaire d'accès doit connaître les pointeurs d'accès et "protéger" les cellules mémoire pour éviter ce cas. Les pointeurs d'accès sont les adresses de lecture et d'écriture à la mémoire.

C'est dans le cas de mémoires activées par deux fréquences différentes que les problèmes

de synchronisation surviennent, c'est l'endroit où l'information passe d'un domaine d'horloge à l'autre. L'avantage d'utiliser une mémoire à double port est que seuls les signaux de contrôle sont transmis d'un domaine à l'autre et les données, elles, sont transmises par la mémoire. Les erreurs de transmission sont ainsi réduites à la gestion des pointeurs d'accès.

Quant à la génération de la mémoire, le compilateur Virage, un outil de synthèse optimisé basé sur une méthodologie automatique de déploiement de designs provenant de "propriétés intellectuelles", peut être utilisé. Il met donc en oeuvre des techniques de pointe minimisant la quantité de transistors et de routage nécessaires afin de créer des cellules mémoire pour une technologie spécifique. Il utilise une librairie de référence qui permet de générer des cellules incluant le placement et le routage du circuit. Virage est conçu de façon à optimiser la surface d'intégration, la consommation de puissance ainsi que la vitesse de fonctionnement.

3.3.5 La mémoire circulaire

La problématique de la mémoire circulaire d'entrée vient du fait que le FIFO est implémenté en SRAM. Les pointeurs de lecture et d'écriture n'ont pas le même adressage pour écrire les données en continue avec un multiplexage temporel et lire les données par canal. Puisqu'un AP peut survenir à n'importe quel moment, l'adresse de lecture doit par conséquent pouvoir débuter le transfert de fenêtres d'analyse à n'importe quel endroit dans les 16 échantillons. Il importe aussi que les données ne soient transmises qu'une seule fois.

Le FIFO sert aussi de tampon de délai. Lorsqu'une fenêtre d'analyse est complétée dans la mémoire de sortie (64 échantillons) il y a un temps de latence avant que la

mémoire ne soit lue dû à la transmission d'autres canaux ou à la machine à états du transmetteur sériel. Le dilemme suit la condition que si un AP est détecté et que la transmission précédente n'est pas terminée, une mémoire à court terme du signal de détection doit être appliqué pour attendre la fin de la lecture ou le remplissage complet du FIFO ; si tel est le cas, la nouvelle fenêtre est alors copiée même si la lecture n'est pas complétée et donc un paquet peut être perdu. Le numéro de séquence inclu dans le protocole de communication permet de les repérer (section 3.4).

3.3.6 Architectures DWT

Dans une analyse comparative avancée des architectures de processeur DWT, Dumortier conclut qu'un excellent compromis entre la consommation de puissance et la surface silicium requise est obtenu en utilisant la structure DWT polyphasée en prenant soin de bien la structurer pour un meilleur parallélisme. Il constate que le calcul rapide incluant le nombre de multiplicateurs et d'additionneurs requis ne suffit pas, il faut aussi considérer la complexité du routage, la sensibilité de quantification et la dissipation de puissance dynamique (Dumortier *et al.* 2006).

Deux architectures sont comparées, la forme basée sur la convolution (polyphasée) et celle basée sur le lissage (par factorisation). La forme polyphasée s'implémente simplement par le placement de filtres FIR (*Finite Impulse Response*) en cascade, tandis que la forme lissée permet une organisation récursive des fonctions de base. Afin d'accomplir la même tâche dans un même laps de temps, la forme lissée nécessite une plus grande fréquence mais permet la réutilisation des mêmes blocs logiques ; d'ailleurs, sa consommation de puissance est nettement supérieure. Dans l'optique d'un pipelinage des données, l'architecture polyphasée semble plus appropriée puisque la capacité (nombre de canaux) peut être augmenter aisément en maximisant l'usage

des ressources matérielles. L'architecture choisie est donc l'arbre de filtre FIR présenté à la figure 2.5.

3.4 Le protocole de communication

Un moyen d'offrir une configuration complète et indépendante de chaque canal est de lui assigner des registres. Afin d'accéder à ces registres, un format de transaction doit être défini. L'usager veut communiquer avec l'implant en envoyant soit un accès lecture ou un accès écriture. L'implant répond à un accès lecture par un retour par le lien bidirectionnel ou bien modifie le registre. Ce qui se résume aux quatre types de paquet pouvant circuler sur le lien : accès requête de lecture, accès écriture, retour de lecture et nouvelles données.

3.4.1 La signification des registres

Une largeur de mot standard de 32 bits a été choisie pour accommoder les besoins en ressources avec l'architecture standard des processeurs. Pour limiter les ressources de décodage, l'encodage des fonctions de commandes a été choisi du type activation unique (*one_hot*) ; si l'on considère 16 canaux dans l'implant pour gérer une matrice de 4x4 électrodes, 4 bits seront nécessaires pour la sélection par adresse. La répartition finale des registres est spécifiée dans le tableau 3.3. Afin d'offrir une synchronisation de tous les canaux au même moment, chacun des canaux est configuré indépendamment, mais l'activation se fait comme une écriture spéciale (bit 23).

TAB. 3.3 Trame de commande pour les accès aux registres de configuration

Assignation des bits	Description
28 à 31	Commande
24 à 27	Adresse
23	1 : Écriture spéciale activation des canaux (bit [15,0]) 0 : Écriture au registre spécifié par l'adresse
22	Réservé (toujours 0)
16 à 21	Threshold bande ca3
10 à 15	Threshold bande cd3
4 à 9	Threshold bande cd2/(sans dwt)
3	Mode Délai
2	Mode DWT
1	Mode FIFO
0	Mode Compression

3.4.2 Lien de données et trame physique

La trame physique est classée comme étant le niveau 1 (*Physical Layer*) dans le modèle de référence à sept couches OSI, alors que le lien de données est le niveau 2 (*Data Link Layer*). Le niveau 1 définit les spécifications électriques et physiques d'un système de communication comme la modulation ou le type de médium ; tandis que le niveau 2 spécifie plutôt la fonctionnalité et les procédures nécessaires au transfert comme la description de la plage mémoire. Puisque l'espace mémoire requis pour le transfert d'une commande approche 32 bits, il a été choisi que la trame de base sera aussi de 32 bits. Ainsi, un transfert de données nécessitant plus d'une trame est encapsulé dans la commande "Nouvelles données" pour se propager sur plusieurs trames. Tous les autres types d'accès sont contenus à l'intérieur d'une seule trame.

Le choix de séquence aussi est primordiale surtout pour un lien sériel. Les deux parties émetteur et récepteur doivent se synchroniser pour s'échanger les données. Cette synchronisation est effectuée par l'envoi d'une trame de référence qui est transmise

dès qu'aucune transaction n'est requise et au moins une fois entre chaque trame utile. Dans le sens extérieur-vers-implant (descendant), une trame de départ indique qu'un accès arrive, l'inverse de la trame de référence a été choisi pour simplifier ensuite la commande d'écriture/lecture est transmise comme le montre un exemple de séquences au tableau 3.4. La redondance avec la trame inversée n'est pas obligatoire mais la trame de référence oui, cela permet de synchroniser les 32 bits reçus qui sont ajoutés un à un dans un registre à décalage. La redondance assure un meilleur fonctionnement dû au fait que chaque état doit être accompli correctement pour ne pas rompre la procédure de configuration de l'implant. Le débit descendant n'est pas élevé et le permet.

TAB. 3.4 Exemple de séquences de trames

No	Lien descendant	No	Lien montant
...	Référence	...	Référence
1	Début(inv. Référence)	1	En-tête (Retour de lecture)
2	Écriture dans un registre	...	Référence
...	Référence	1	En-tête (Nouvelles données)
1	Début(inv. Référence)	2	Données échantillons(0 à 3)
2	Requête de lecture	...	...
...	Référence	17	Données échantillons(60 à 63)

Le protocole UTOPIA L2 (*Universal Test and Operations PHY Interface for ATM Level 2*) a été adopté en 1995 par un comité technique comme standard pour des systèmes permettant jusqu'à 31 esclaves pour un maître. Le protocole proposé s'inspire de ces champs de données pour finalement obtenir une bonne robustesse d'un point de vue de validation comme le numéro de séquence et l'étampe temporel.

La description de tous les champs est faite au tableau 3.5. La commande "Nouvelles données" indique un début de paquet ; l'adresse indique le canal transmis ; le numéro de séquence incrémente à chaque paquet transmis ce qui permet de valider si un

paquet a été perdu ; l'étampe de temps permet de localiser le moment où le AP a été détecté ; et la longueur indique le nombre d'échantillons composant le paquet à transmettre. Les trames subséquentes seront associées à ce paquet tant que la longueur ne correspond pas au nombre de données reçues. La longueur peut varier dans le cas d'une compression de données mais sera fixe dans le cas contraire.

TAB. 3.5 Trame d'en-tête de paquet de données

Assignation des bits	Description
28 à 31	Commande "Nouvelles données"
24 à 27	Adresse du canal
20 à 23	Numéro de séquence
8 à 19	Étampe de temps (*Timestamp*)
0 à 7	Longueur du paquet

3.4.3 Compression avec DWT

Dans le même ordre d'idée de reconstitution du paquet, si une compression a lieu, les échantillons sélectionnés doivent être identifiés. En considérant les données générées par le processeur d'ondelettes, seule les bandes a3, d3 et d2 (figure 2.5) sont nécessaires car le signal est échantillonné à 32 kHz mais seuls les coefficients des plus basses fréquences contiennent l'information pertinente. Donc, pour une fenêtre d'analyse de 64 échantillons en mode compression, un maximum de 32 coefficients sera transmis. Afin d'indiquer quel coefficient a été transmis, une trame a été ajoutée dans le protocole, un bit pour chacun des 32 bits de la trame pour indiquer à quels coefficients sont associés les données. La concordance peut être calculée pour valider que le nombre de bits activés correspond à la longueur spécifiée dans l'en-tête, sinon il y a erreur de transmission.

TAB. 3.6 Séquence de trames avec compression DWT

No Trame	Description
1	Référence
2	En-tête
3	Indicateur de coefficients
4	Données échantillons(0 à 3)
5	Données échantillons(4 à 7)
	...
Trame$_N$	Données échantillons(N_{win}-3 à N_{win})

3.5 Description du contrôleur numérique

Le module numérique conduit les données échantillonnées vers un lien sériel tout en appliquant une transformation pour compresser le signal. L'interface d'entrée du flot de données fonctionne à la même fréquence que la sortie du circuit mixte et du côté de la sortie sérielle, le flot est activé par l'horloge de référence sérielle (sclock) comme représenté à la figure 3.7. Les données entrent du côté mixte selon la cadence et le multiplexage effectué par le module de génération d'horloges (section 3.3.3).

Le module de génération des horloges et du compteur global est indépendant du reste du contrôle. De plus, afin d'assurer une meilleure tolérance aux délais d'entrée, chacun des signaux est mémorisé par une bascule D à l'entrée sans logique combinatoire. Les états des circuits sont balancés en conséquence.

3.5.1 Architecture globale du contrôleur

Les ports montrés à la figure 3.7 sont présentés en détails dans leurs sous-sections respectives. Les modules principaux en contact avec l'extérieur sont le récepteur sériel et le transmetteur sériel qui gèrent les accès aux registres ainsi que la sortie des

données. Pour l'échantillonnage, les données de l'AFE passe par le processeur DWT (ou directement au module de détection) avant de mémoriser les données dans le FIFO d'entrée. Le module de décomposition DWT transforme les échantillons un à un et les passe au module de détection qui écrit dans le FIFO d'entrée. Le module de gestion des mémoires, comme son nom l'indique, s'occupe de gérer les accès lecture à la mémoire FIFO et d'écrire les paquets dans la mémoire tampon de sortie sous la forme de paquets qui seront finalement lus par le transmetteur sériel. La description détaillée des composants, des constantes et des types utilisés dans le projet est faite à l'annexe II.

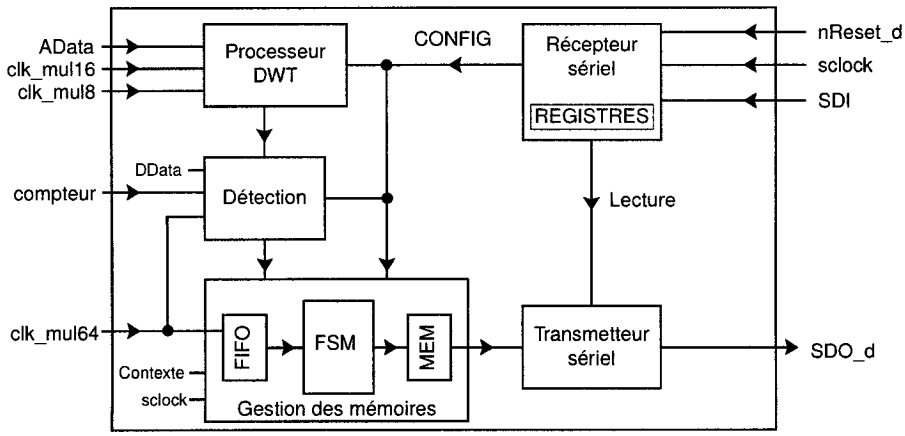


FIG. 3.7 Architecture globale du contrôleur numérique

3.5.2 Le récepteur sériel

Ce module sert de point d'entrée pour la configuration de l'implant. Il contient le décodage d'adresse pour la lecture et l'écriture aux registres et il respecte le protocole de communication présenté à la section 3.4. La configuration des canaux se fait par canal tandis que l'activation peut être simultanée. Lorsqu'une requête de lecture est décodée, la réponse est redirigée vers le transmetteur sériel par le type registres-

vers-transmetteur (reg2col_type) défini pour communiquer les requêtes. Aucune autre opération n'est acceptée tant que la terminaison n'est pas reçue. Vue la priorité des opérations, les retours de lecture sont prioritaires et le délai est court.

3.5.3 Le transmetteur sériel

Ce module est le point de sortie de l'implant. Par lui, toute information est transmise au contrôleur externe. Les requêtes de lecture sont traitées de façon prioritaire sur les données acquisitionnées mais l'étape de lecture de registre ne devrait pas se faire fréquemment car le contrôleur externe, qui écrit les données, connaît déjà leurs valeurs et ce ne devrait être utilisé que pour la validation de la configuration.

C'est dans ce module que la limite du débit de transmission se fait sentir. Les optimisations de paquetisation et de compression permettent de gérer un maximum de canaux pour le même port de sortie. Afin de maximiser le débit de sortie, la séquence de trames telle que présentée au tableau 3.4 exclue la trame inversée. Les séquences possibles se résument par :

- Retour de lecture, la plus courte avec 2 trames : référence et en-tête.
- Paquet avec compression DWT, longueur variable de 4 à 11 trames : référence, en-tête, indicateur de coefficients et données (4 à 32 octets).
- Paquet complet, sans compression (temporelle ou DWT) ou compression temporelle, la séquence la plus longue avec 18 trames : référence, en-tête et données (64 octets).

3.5.4 Le module de processeur d'ondelettes

Ce module effectue une transformation en ondelettes pipelinée de 16 canaux. Il accumule les données une à une mais les traite par couple dès que les échantillons pair et impair d'un canal ont été emmagasinés. C'est la raison pour laquelle deux horloges sont nécessaires. Les horloges exprimées par "clk_mul16" et "clk_mul8" présentées au tableau 3.1, et utilisées ici, représentent les facteurs multiplicateurs par rapport à la fréquence d'échantillonage pour un canal, soient 16 et 8 fois respectivement. Un compteur sert à maintenir le calculateur dans un état connu pour savoir quand un étage de filtre FIR possède une sortie valide pour l'entrée du filtre suivant. Les sorties de ce module sont acheminées directement au module de détection.

3.5.5 Le module de détection

Ce module contrôle les entrées du comparateur de seuil absolu et l'écriture dans la mémoire FIFO. En fonction du mode de configuration des canaux, le seuil correspondant à l'échantillon présent est sélectionné : un seuil pour les données telles quelles ou pour les bandes d'ondelettes. Chacune des données d'un canal est écrite dans le FIFO dès qu'elle est disponible. Le signal de détection activera à son tour le module de gestion des mémoires.

Il est possible que les données en sortie du module de transformée en ondelettes soient valides en même temps sur plusieurs signaux étant donnée la structure parallèle du circuit. Ce qui ne permet pas de sélectionner la sortie valide avec un simple multiplexeur cadencé à l'horloge d'entrée des échantillons, un balayage plus rapide s'impose et c'est la raison pour laquelle l'horloge du module est quatre fois supérieur à l'entrée pipelinée (64 clk_ech). Le ratio de huit coefficients par huit échantillons est conservé

selon la distribution temporelle présentée au tableau 3.7.

TAB. 3.7 Échantillonnage des coefficients DWT

d1	d2	d1	d3	d1	d2	d1	a3

3.5.6 Le module de gestion des mémoires

La gestion des mémoires est une problématique complexe du circuit. L'écriture et la lecture doivent être protégées pour abolir les conflits de simultanéité, une lecture ne doit en aucun cas être active si les deux ports d'adresse correspondent. Puisque l'accumulation ne se fait pas au même débit que la transmission, un comparateur asynchrone s'assure de respecter cette condition. La gestion de l'écriture au FIFO d'entrée est faite par le module de détection tandis que ce module s'occupe de l'écriture dans la mémoire de sortie.

Lorsque le module est mis en mode compression avec la DWT activée, il valide pour chaque échantillon si le seuil a été dépassé, si oui, il copie cette données, sinon il ne la copie pas et une compression est effectuée (par sélection de coefficients). Afin de reconstruire le signal, un vecteur de positionnement de 32 bits indique si l'échantillon a été transmis, c'est l'indicateur de coefficients. Seuls 32 bits sont nécessaires pour une fenêtre de 64 échantillons car les coefficients les moins significatifs (d1) ne sont pas transmis inutilement du fait que leur contenu fréquentiel ne correspond pas au contenu fréquentiel d'un potentiel d'action.

La mémoire tampon de sortie est séparée en portions égales pour chaque canal à une adresse fixe. L'adresse fixe permet d'accumuler les données de chaque canal indépendamment. L'horloge est la même pour les deux ports d'accès et ainsi que pour les signaux de synchronisation. Ces signaux sont les indicateurs de coefficients ,

les indicateurs qu'une fenêtre d'échantillon est prête, les numéros de séquence, les longueurs de paquets, les étampes temporelles et le retour de données de la mémoire. Tous réunis, ils forment un type qui a été défini comme le type FIFO-vers-transmetteur (fifo2col_type). Les signaux qui commandent la lecture dans la mémoire tampon sont générés par le module transmetteur sériel et comprennent entre autre les indicateurs de fin de paquet ; ils sont définis par le type transmetteur-vers-FIFO (col2fifo_type).

3.5.7 Les modes de fonctionnement

Les modes de fonctionnement permettent une configuration individuelle des canaux sur quatre bits (tableau 3.3), donc un choix entre une copie intégrale ou une transformée en ondelettes des données ; et/ou des fenêtres d'analyse basées sur la détection d'un AP suite au dépassement du seuil sur la valeur absolue du signal ; et/ou des fenêtres d'analyse décalées de seize échantillons précédents à la détection ; et/ou des fenêtres décalées de huit échantillons supplémentaires pour les données intégrales, ce qui équivaut à la latence du processeur d'ondelettes. Ce dernier revient à dire que ça permet de transmettre un paquet de données intégrales, basé sur un dépassement du seuil sur le résultat du module DWT. Le tableau 3.8 résume les différents modes de fonctionnement possibles.

Une démonstration de l'application des modes de fonctionnement est faite à la figure 3.8, pour une détection par seuillage temporel, ou à la figure 3.9, pour une détection par seuillage sur le résultat de la transformée en ondelettes.

TAB. 3.8 Modes de fonctionnement

Actif	Compr	FIFO	DWT	Délai	Description de l'action
0	X	X	X	X	Attente de configuration
1	0	X	0	X	Acquisition intégrale des données
1	1	0	0	0	Compression par fenêtrage sur seuillage simple
1	1	0	0	1	Compression par fenêtrage sur seuillage simple avec délai de 8 éch. avant le seuil
1	1	1	0	0	Compression par fenêtrage sur seuillage simple avec délai de 16 éch. avant le seuil
1	1	1	0	1	Compression par fenêtrage sur seuillage simple avec délai de 24 éch. avant le seuil
1	0	X	1	X	Acquisition intégrale des coefficients DWT
1	1	0	1	0	Compression par sélection de coefficients du processeur DWT
1	1	1	1	0	Compression par sélection de coefficients du processeur DWT et délai de 16 éch.
1	1	0	1	1	Compression par fenêtrage intégrale et seuillage sur coefficients DWT
1	1	1	1	1	Compression par fenêtrage intégrale et seuillage sur coefficients DWT et délai de 16 éch.

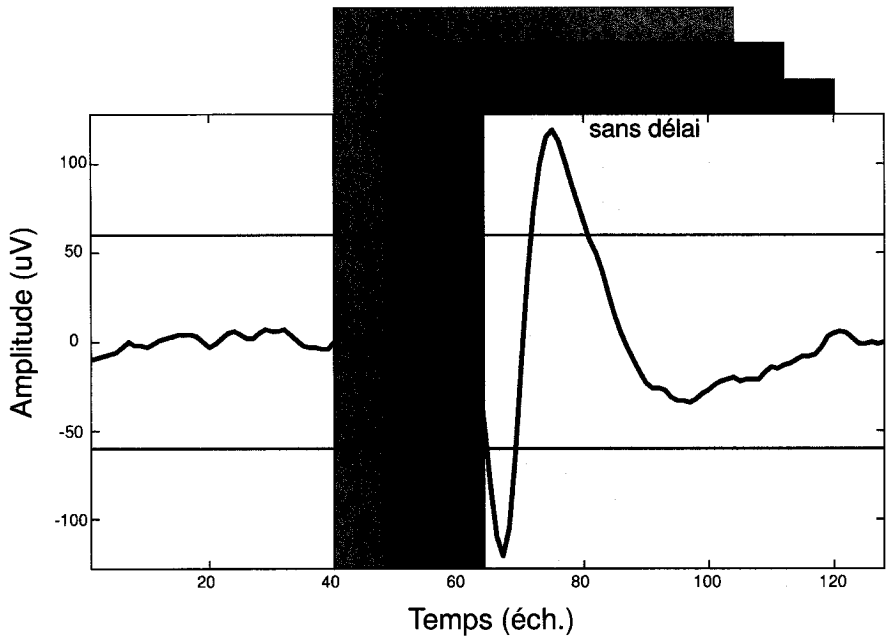


FIG. 3.8 Fonctionnement de base du système

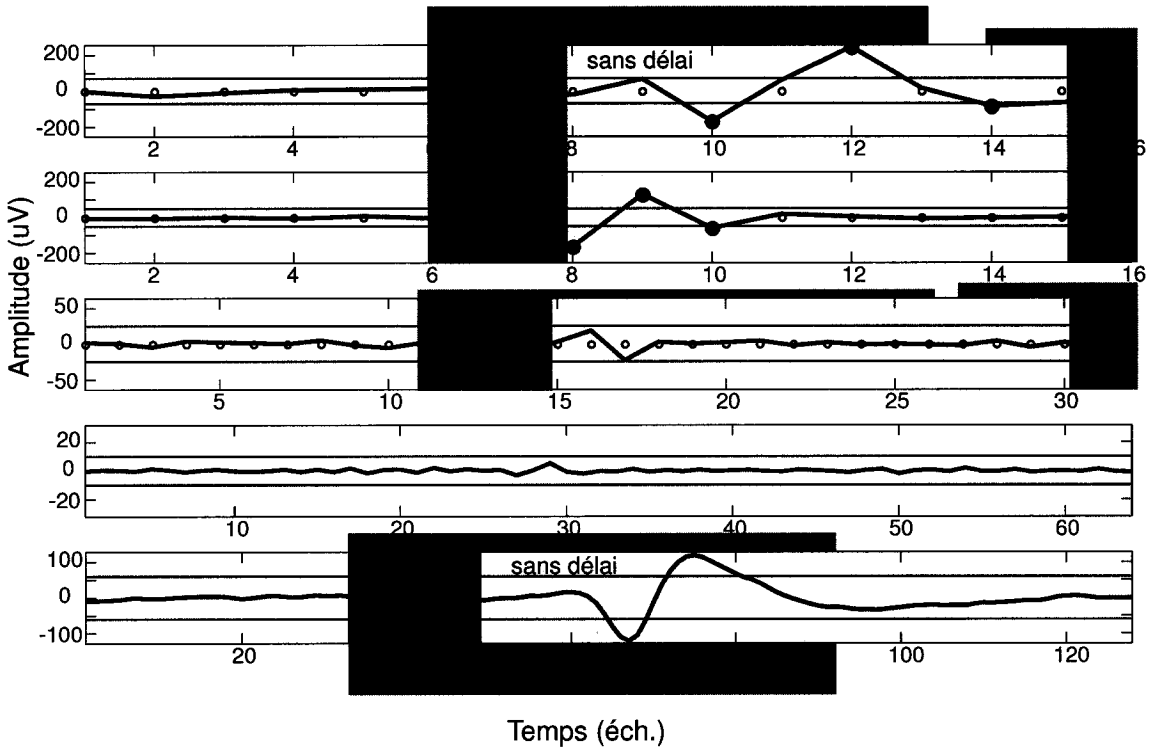


FIG. 3.9 Fonctionnement du système avec l'utilisation du processeur d'ondelettes

CHAPITRE 4

RÉSULTATS

4.1 Introduction

Dans ce chapitre sont rassemblés les résultats expliquant les choix architecturaux et algorithmiques par comparaison des différentes possibilités présentées dans les précédents chapitres. Premièrement, la comparaison des taux de compression en fonction de la qualité de reconstitution du signal prouve que l'algorithme de compression par DWT était le meilleur choix dans notre cas. Ensuite, les méthodes de détection d'activité neuronale présentées au chapitre 2 sont aussi comparées pour confirmer le choix de la DWT.

Après une description de la méthode de test et de validation pour le design VHDL, le logiciel de gestion qui doit commander le contrôleur numérique est présenté. Deux méthodes sont comparées, l'une avec des processus concurrents et des gardes pour le partage des mémoires nécessitant un système d'exploitation et une autre roulant en boucle infinie s'activant avec des indicateurs d'états.

Finalement, un survol des prototypes réalisés sur FPGA permet de montrer le processus de développement qui a mené à la réalisation d'une puce dédiée (*Application Specific Integrated Circuit*-ASIC) en procédé CMOS 0.18μm. Cette puce contient seulement une partie du circuit présenté au chapitre 3 ; une description des compromis nécessaires versus les contraintes physiques rencontrées explique quel sera le premier prototype complet de l'équipe Cortisens.

4.2 Analyse des filtres et transformées pour la compression

Avant la réalisation du système électronique, les simulations avec Matlab ont permis de valider les choix architecturaux qui ont été implémentés par la suite. Comme décrit dans le chapitre 2, certains algorithmes nécessitent une période de configuration pour l'établissement des niveaux de seuil ou pour l'entraînement du module de filtres par exemple. Pour éviter la confusion et assurer une comparaison la plus exacte possible le signal d'entrée a été divisé en deux ensembles (ENTRAÎNEMENT et ANALYSE). L'ensemble d'entraînement est utilisé si nécessaire et l'ensemble d'analyse est le point de référence pour chacun des algorithmes ou techniques comparés.

Cette section présente une analyse préliminaire de la compression pure : sans en-tête dû au protocole. Les algorithmes suivants ont été utilisés pour la comparaison : FFT (section 2.3.1), DCT (section 2.3.2), DWT (section 2.3.3) et PCA (section 2.3.4).

Les algorithmes utilisant un schème d'implémentation (présenté à la section 2.4) du type traitement post-détection sont très sensibles au problème de fenêtrage des échantillons. Par contre, les algorithmes DCT, DWT et FFT peuvent utiliser un schème avec traitement pré-détection ce qui n'est pas le cas pour l'analyse PCA. La méthode avec traitement post-détection nécessite un temps de convergence à chaque échantillon compressé, dû au remplissage nécessaire (souvent des zéros), qui diminue de beaucoup la qualité de reconstruction sur une petite fenêtre ; alors que le filtre calculé en continu se maintient en permanence dans un état actuel valide.

4.2.1 Taux de compression comparés

Comme expliqué au chapitre 2, l'erreur quadratique est le point de comparaison pour la consistance du modèle à choisir (section 2.2.3). On observe à la figure 4.1 les résultats de la compression des signaux, soit avec un traitement pré ou post détection. On voit que dans les deux cas, l'algorithme DCT donne un meilleur taux de compression que la FFT, comme le laissait supposer la théorie (expliquée dans la section 2.3) car la résolution temporelle de la DCT est supérieure pour une résolution fréquentielle choisie. Ce qui nous permet d'éliminer l'option FFT. Par contre, la relation entre la DCT et la DWT n'est pas aussi évidente. Dans le cas d'une détection sans prétraitement, la DCT minimise l'erreur tandis que la DWT donne un meilleur résultat lorsqu'il est évalué en continu. La fenêtre d'analyse pour un traitement post-détection aurait dû être agrandie pour pallier ce problème. Ce qui aurait entraîné l'usage de plus gros modules mémoire, ce qui n'est pas souhaitable.

L'algorithme DWT nécessite quelques échantillons pour stabiliser le résultat. Dans le cas d'une évaluation continue (traitement pré-détection) l'effet tend vers zéro tandis que pour le traitement d'un court échantillon, l'effet n'est plus négligeable. Cela explique aussi pourquoi l'erreur quadratique ne tend pas vers zéro mais bien vers un niveau que représente le bruit non reconstitué.

En ce qui concerne les résultats de compression avec PCA, les résultats étonnent du fait que cet algorithme devrait être optimal statistiquement. La mauvaise performance de compression s'explique par la sensibilité au fenêtrage de l'algorithme. C'est-à-dire que si les échantillons ne sont pas toujours centrés de la même manière dans la fenêtre d'analyse, la PCA ne reconnaît plus l'échantillon comme étant similaire aux vecteurs propres utilisés pour la compression. Ainsi, la projection de l'échantillon sur le vecteur propre n'a plus autant d'entropie qu'il le devrait.

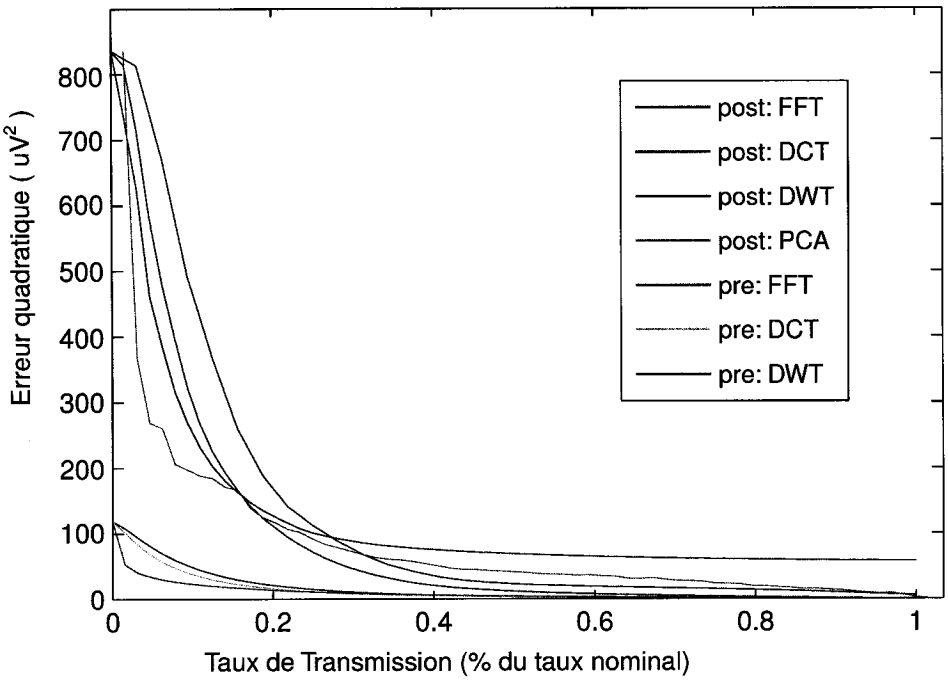


FIG. 4.1 Compression sans en-tête pour le protocole

4.3 Analyse des techniques de détection d'activité

Le taux de compression représente la spécificité et la sélectivité du système. La spécificité est augmentée lorsque le bruit est bien caractérisé (non transmis inutilement) et la sélectivité est augmentée si le taux de VP surpasse le taux de FP (le paquet transmis contient vraiment un AP). Si le système est en mesure de choisir correctement le AP émis, il compresse mieux les données. Pareillement, l'erreur quadratique s'associe à la sensibilité du système ; si tous les AP sont détectés il y aura moins d'erreur après la reconstruction. S'il y a des fausses détections, ce ne sera que de l'information supplémentaire pour caractériser le bruit. Il est évident que l'erreur quadratique est minimisée au détriment du taux de compression, dans tous les cas.

Les méthodes simple, absolue, NEO et avec seuil adaptatif transmettent les AP avec

fenêtre d'échantillonage temporelle sans compression supplémentaire. Dans le cas des méthodes DWT et DCT, un pré-traitement du signal est effectué avant une compression subséquente par sélection de coefficients.

Les techniques de détection comparées sont les suivantes :

1. Seuil simple et absolu (section 2.5.1)
2. Mesure d'énergie (NEO) (section 2.5.2)
3. Méthode avec seuil adaptatif (section 2.5.4) en fonction de l'écart type (*Standard Deviation*-SD).
4. Transformée en ondelettes (section 2.5.5) en mode continu avec une structure en arbre et seuil absolu.
5. Transformée en cosinus (section 2.5.5) en mode continu avec fenêtre coulissante et seuil absolu.

4.3.1 Taux de détection comparés

Les différentes techniques de détection d'activité neuronale sont comparées dans le tableau 4.1. Les trois niveaux de bruit sont les mêmes que ceux utilisés à la figure 4.3. On voit que les résultats obtenus par les méthodes avec compression en mode continu (DWT et DCT) donne une meilleure caractérisation du signal avec la même quantité de données. Ces deux méthodes atteignent ainsi de meilleurs taux de transmission avec une même qualité pour le signal reconstitué comparativement aux autre méthodes. On en conclue que l'entropie des données transmises est donc plus grande.

La sensibilité ($SEN = \frac{VP}{VP+FN}$) et la sélectivité ($SEL = \frac{VP}{VP+FP}$) relativisées dans le tableau 4.1 sont présentées à la section 2.5. La spécificité n'est pas montrée car elle est complémentaire aux résultats des deux autres caractéristiques. L'analyse a été

produite à partir du signal de référence dont seulement certains AP ont été localisés et le reste du signal ayant été éliminé pour connaître l'emplacement de chacun des AP existant dans le signal de test.

TAB. 4.1 Comparaison des techniques de seuillage

Taux Transmission (TT)	SNR = 10 db		SNR = 5 db		SNR = 1 db	
TT : $10 \pm 0.5\%$	SEN	SEL	SEN	SEL	SEN	SEL
Seuil Simple	100 %	91 %	100 %	99 %	98 %	100 %
Seuil Absolu	100 %	86 %	100 %	88 %	99 %	92 %
Seuil NEO	100 %	92 %	98 %	91 %	74 %	72 %
Adaptatif $(a \cdot SD)$	100 %	90 %	64 %	45 %	47 %	24 %
DWT	100 %	50 %	100 %	13 %	99 %	13 %
DCT	100 %	50 %	99 %	13 %	97 %	12 %
TT : $5 \pm 0.25\%$						
Seuil Simple	54 %	100 %	54 %	100 %	54 %	100 %
Seuil Absolu	51 %	100 %	50 %	100 %	54 %	100 %
Seuil NEO	54 %	100 %	55 %	100 %	57 %	100 %
Adaptatif $(a \cdot SD)$	54 %	99 %	28 %	55 %	11 %	21 %
DWT	100 %	51 %	100 %	14 %	91 %	21 %
DCT	100 %	50 %	97 %	23 %	81 %	20 %
TT : $2.5 \pm 0.125\%$						
DWT	100 %	80 %	97 %	45 %	76 %	36 %
DCT	100 %	62 %	91 %	46 %	65 %	28 %
TT : $1 \pm 0.05\%$						
DWT	76 %	100 %	74 %	100 %	50 %	54 %
DCT	78 %	96 %	75 %	91 %	41 %	48 %

L'analyse a été faite en utilisant des taux de compression fixes pour mettre en relation l'efficacité de la compression versus l'efficacité de la détection. Le taux de compression de départ à 10% vient de la connaissance *a priori* que le signal de référence contient un peu moins de 10% de AP. L'avantage des méthodes DWT et DCT est l'élimination des coefficients associés au bruit et un lissage de la fonction de sortie après la reconstruction du signal. Un taux de transmission de 100% signifierait que le nombre de bits transmit correspond au nombre de bits dans la séquence de test.

Le taux de transmission inclue les bits ajoutés pour les entêtes de paquet et pour les positionnements dans le cas des compressions DWT et DCT afin que le pourcentage soit représentatif de la réalité.

Les techniques simple, absolue et NEO semblent être les plus performantes dans le cas normal (à 10%). Elles montrent des sensibilités quasi complètes et des sélectivités tout aussi élevées. Il est important de faire connaître que chacun des seuils a été ajusté pour obtenir ce résultat après une période de convergence. C'est pourquoi ces méthodes surclassent le seuil adaptatif qui ajuste le seuil automatiquement. Pour obtenir des résultats expérimentaux similaires, il faudrait une boucle de rétroaction pour la reconfiguration successive des seuils (section 4.6). Dans le cas contraire, bien que moins sélectif, le seuil adaptatif reste tout indiqué car il est suffisamment sensible.

Ces techniques compressent seulement par sélection temporelle d'événements et perdent rapidement de la sensibilité dès que le seuil augmente. Dans le cas de la compression à 5%, ces techniques demeurent sélectives, mais ne transmettent que les AP de plus grandes amplitudes tandis que les méthodes DWT et DCT conservent leur bonne sensibilité bien que leur sélectivité soit plus petite. Pour ces dernières, deux autres cas de taux de transmission (2.5% et 1%) permettent de voir où la sensibilité commence à diminuer.

La sensibilité est le critère le plus important de la comparaison car il valide que la technique est consistante à reconnaître les AP ; ensuite, la sélectivité est supputée pour évaluer l'efficacité du système. On vérifie si seulement les AP sont transmis ou bien si du bruit a aussi été transmis ; ce qui indique si les paquets transmis sont spécifiques ou non. Les taux de transferts optimaux pour les techniques de détection se situent au moment où la sensibilité est de 100% pour une détection complète et que la sélectivité est maximisée.

1. DWT : Taux de transfert de 1.5% pour une sélectivité de 99%
2. DCT : Taux de transfert de 2.5% pour une sélectivité de 62%
3. Simples : Taux de transfert de 9.5% pour une sélectivité de 100%
4. Adaptatif : Taux de transfert de 18.6% pour une sélectivité de 52%

4.3.2 Taux de compression avec bruit variable

Les algorithmes DWT et DCT semblent intéressants pour une analyse plus approfondie puisqu'ils ont obtenu de bonnes performances de compression et de sensibilité (figure 4.1 et tableau 4.1). Chacun des couples de courbes sur la figure 4.2 représente le résultat obtenu avec un niveau de bruit différent. On voit que dans chacun des cas comparés, l'erreur quadratique est plus basse pour le résultat de la DWT comparativement à celui de la DCT. La DWT est donc plus spécifique que la DCT dans tous les cas : avec ou sans bruit ajouté. Plus le bruit augmente, plus les deux algorithmes tendent vers le même résultat.

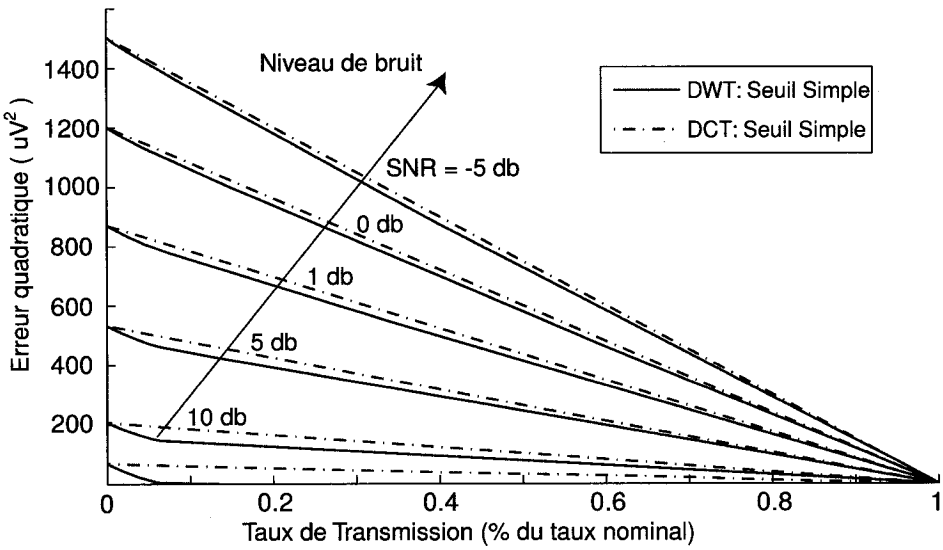


FIG. 4.2 Compressions DWT et DCT comparées

4.4 Débruitage par compression DWT

Comme il a été expliqué au chapitre 3, le signal d'intérêt se situe dans les plus basses fréquences. Les coefficients d'ondelettes qui proviennent de la transformée permettent une reconstruction sans perte du signal si tous les coefficients sont réutilisés. Par contre dans un système de compression avec perte les coefficients non significatifs peuvent être éliminés et remplacés lors de la reconstruction. Le remplacement est fait soit par une mise à zéro ou par un estimé de la valeur.

Un exemple de signal reconstitué par substitution est montré à la figure 4.3. Le même signal est présenté avec trois niveaux de bruit, le même algorithme de seuillage simple sur la DWT appliqué jusqu'à obtention d'un taux de transmission moindre que 5%. On constate que le bruit n'est pas reconstituer car les coefficients ont été réinitialisés.

Le signal sans bruit reproduit assez fidèlement le premier AP (de plus forte amplitude) tandis que le deuxième diffère un peu. Il est intéressant de voir que même si beaucoup de bruit est ajouté au signal, le système détecte correctement les AP et les reconstitue proportionnellement. Malheureusement, lorsque le bruit devient trop fort, le système détecte seulement les AP de grandes amplitudes mais en plus il ajoute des fausses détection.

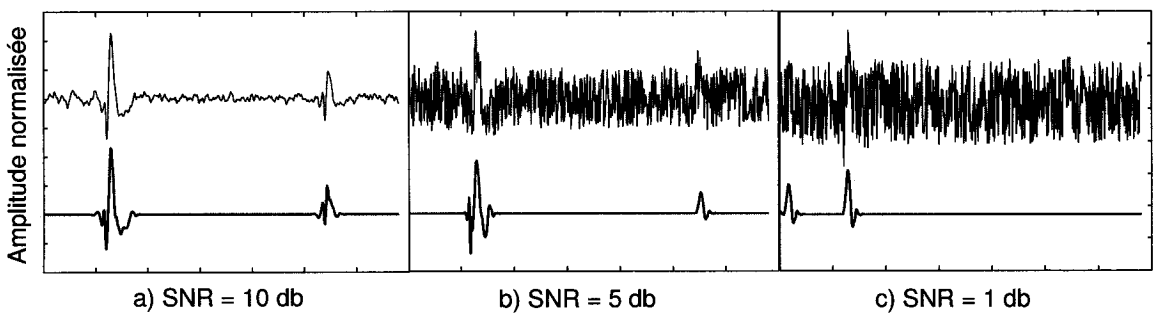


FIG. 4.3 Débruitage lors de la compression DWT à 5%

4.5 Les simulations en langage matériel

La réalisation du projet passe par sa description en langage matériel. Le langage VHDL (IEEE standard 1164) a été choisi pour la facilité d'utilisation des outils qui utilisent ce format. Certains modules Verilog sont tout de même inclus dans le projet : les fichiers générés par le compilateur Virage et les librairies de simulation fournies par les outils Synopsys et Cadence pour la réalisation du ASIC (section 4.8).

Le VHDL est un langage fortement typé : qui permet de définir des types complexes pour des systèmes facilitant ainsi la compréhension tout en assurant une intégrité du comportement du code à l'exécution. Ce qui a permis de regrouper des signaux et de faciliter le design. Le type "record", par exemple, est utilisé pour la description des ports inter-modules (voir annexe II). Les types "function" et "procedure" ont permis une description claire des bancs d'essais et ont ouvert une porte pour la méthodologie Co-Design.

4.5.1 Méthodologie Co-Design

Le projet représente une grande charge de travail et doit être développé étape par étape. L'élaboration de bancs d'essais devient primordial pour la réussite de ce projet. La figure 4.4 montre l'application globale dans son environnement de test. Le même schéma peut s'appliquer pour chacun des modules en remplaçant le système ENG par un DUT (*Design Under Test*) en utilisant les bancs d'essai appropriés. Chaque niveau peut ainsi être testé indépendamment des autres.

Le logiciel d'acquisition est écrit en langage C et sa description est faite dans la section 4.6, son but est de simplifier les commandes à des accès du type protocole de

communication (présenté à la section 3.4). Le logiciel en C commande directement un banc d'essai qui transmet et reçoit selon les trames de contrôle comme le ferait le lien sans-fil. Un autre banc d'essai simule le côté analogique tandis que d'autres modules permettent de générer des signaux globaux. Le but de cette méthode est de développer le logiciel en même temps que le matériel tout en ayant le plein contrôle de la simulation.

Le module stimpkg est un regroupement (*package*) VHDL. Il communique à une librairie C par transfert de chaîne de caractères, des messages. Lorsqu'un message est lu et interprété, le stimpkg exécute la commande qui transmet le message au bon banc d'essai. C'est aussi par lui que les données sont retransmises au logiciel de gestion si le besoin est.

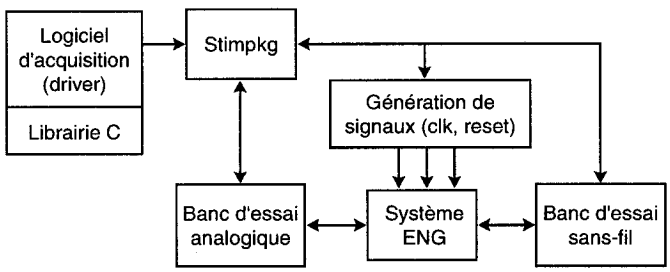


FIG. 4.4 Schéma du dispositif de test et de validation

4.5.2 Implémentation avec Modelsim

Plusieurs méthodes permettent d'accéder au VHDL par un autre langage de programmation. Le langage prévu pour la partie externe du système ENG est le C : facilement compilable vers différentes architectures de processeurs embarqués.

Le SystemC est une méthode de co-design souvent utilisée : une librairie C++ incluant plusieurs modules et fonctionnalités. Une autre méthode d'accès au VHDL par

le C est fournie avec l'outil de simulation VHDL ModelSim : la FLI (*Foreign Language Interface*). Le module de ModelSim s'active par l'ajout simple d'un attribut à l'architecture de l'entité FLI.

Puisque la méthode de test et validation est très importante, son développement a débuté très tôt dans le projet. La FLI a été préférée car la libraire SystemC n'était pas opérationnelle dans les débuts du projet pour différentes raisons administratives.

L'implémentation avec le module FLI est optimisée pour accélérer son utilisation puisqu'elle est inclue dans ModelSim. La vitesse de simulation ne peut malheureusement pas atteindre celle du temps réel mais les fonctionnalités exactes du logiciel versus le matériel peuvent et sont validées. Avec un peu de patience et un ordinateur possédant une bonne puissance de calcul, quelques secondes de simulation sont envisageables.

Une librairie dynamique en C a donc été écrite pour décrire le processus de communication par transfert de message. Les fonctionnalités logicielles sont testées une à une pour obtenir un taux de couverture du système de près de 100%.

Le taux de couverture est le pourcentage des branchements possibles que la simulation a exécuté. Ce qui permet de valider que le système a été entièrement testé. Le taux n'atteint pas 100% car plusieurs points de vérification sont inclus dans le code VHDL. La fonction "assert" permet de vérifier que le système fonctionne selon les spécifications en donnant des messages d'alerte ou d'erreur. Ces bouts de code ne sont pas compilés lors de la synthèse physique du système et peuvent être ignorés lors du test.

4.6 Logiciel de gestion de contrôle

Premièrement, afin de communiquer correctement avec le module matériel, il est important que le protocole établi soit respecté. Deuxièmement, il faut que le système respecte un certain ordonnancement. Voici la séquence suggérée :

1. Établir la communication
   (a) Choisir le mode.
   (b) Enlever la "mise à zéro".
   (c) Transmettre/attendre la trame de référence
   (processus en boucle géré par l'interface physique).
2. Configurer les seuils : bit23 = 0 (tableau 3.3)
   (a) Écriture aux registres
   (b) Validation par la lecture des registres
3. Activer l'acquisition : bit23 = 1 (tableau 3.3)
   - Si en mode **compression**, l'activation est faite en une opération d'écriture.
   - Si en mode **sans compression**, un *scheduler*[1] doit activer les canaux par groupes pour éviter que tous les canaux transmettent en même temps. La résolution est de 16 échantillons qui est la largeur du FIFO d'entrée. Il y a donc 4 niveaux de division possible pour une fenêtre de 64 échantillons.
4. Réception des paquets
   (a) Décomposition des paquets en données (protocole).
   (b) Rétroaction pour BCI ou enregistrement.
   - Si **sans adaptation**, (tâches 4 en boucle).

[1] scheduler : gestionnaire temporel de l'usage matériel qui réparti les tâches pour obtenir la fluidité désirée du flux de données

- Si **avec adaptation**, calcul des seuils et configuration des seuils (tâches 2 et 4 en boucle).

4.6.1 Le seuillage par méthode adaptative

En considérant que le changement d'environnement autour des électrodes ne se fera pas instantanément, le mode adaptatif peut être fait par logiciel sans perturber la transmission des données vers l'extérieur. Cela devient possible du fait que le lien de communication est entièrement bidirectionnel (*full-duplex*) comparativement à d'autres configurations dites partiellement bidirectionnelles (*half-duplex*).

Si le processeur du contrôleur externe peut traiter les paquets au fur et à mesure et que le processeur n'est pas à pleine capacité, il peut aussi varier les seuils de détection par des accès écriture. Je recommande une configuration logicielle pour une plus grande latitude expérimentale.

4.6.2 Implémentation du logiciel de contrôle

L'implémentation du logiciel de contrôle peut se faire de deux manières différentes : selon une approche avec un ordinateur personnel ou avec un processeur embarqué minimal (pour la portabilité).

L'approche avec ordinateur personnel permet une grande puissance de calcul et inclue tous les types de processeurs permettant de rouler un système d'exploitation relativement complexe comme Linux, Unix, Windows, etc. Un tel système permet de rouler plusieurs processus en parallèle (concurrent) et de les interconnecter par des sémaphores. Les sémaphores sont des primitives pour le mécanisme de synchronisa-

tion qui utilisent le principe d'exclusion mutuelle permettant le partage de ressources (la mémoire dans notre cas).

Une application logicielle a été réalisée en utilisant cette technique de processus parallèles (*multithreading*) et les tampons de garde (*mutex*). Cette méthode permet de transférer facilement les données d'un processus à l'autre sans danger de collision en accès mémoire. Principalement trois processus sont nécessaires :

1. Réception de données
2. Traitement des paquets pour l'enregistrement et la rétroaction.
3. Processus global choisissant les actions de contrôle global comme le choix des canaux, la configuration des seuils et la durée d'acquisition.

Différents niveaux de priorité peuvent être appliqués aux processus afin d'assurer le service des processus d'importance comme la réception de données afin d'éviter que le tampon d'entrée ne soit plein. Ultimement, le processus global serait commandé par une interface graphique (*Graphic User Interface*-GUI) que l'usager pourrait aussi utiliser pour visualiser les signaux reçus. Même dans le cas de l'approche avec processeur embarqué (prochaine section) il sera nécessaire d'utiliser un logiciel sur un ordinateur personnel pour l'emmagasinage des données et pour les traitements post-acquisition.

L'application développée en C, aurait pu être développée en C++ ou n'importe quel autre langage pouvant gérer les interfaces car le système d'exploitation est interchangeable avec un autre tant que les interfaces physiques ne changent pas.

En ce qui concerne la deuxième approche avec un processeur embarqué minimal, la majorité de ces systèmes qui pourraient servir à notre application ne permettent pas de rouler des systèmes d'exploitation complexes offrant le mode par processus parallèles. Par contre, la majorité des ces processeurs supporte les interruptions avec un processus unique et des périodes de veille. Les périodes de veilles sont des moments

où le processeur ne fait rien et se met en attente.

Un processus en boucle infini peut donc être défini avec une gestion de priorité fait par variables globales, à la manière d'une machine à états. Cette méthode est plus difficile à gérer car la gestion des fonctionnalités est faite artificiellement au lieu de l'être par le système d'exploitation dont c'est la tâche principale. D'un autre côté, la gestion des sémaphores disparaît.

Si le système le permet, un tel système fonctionnant par interruptions offre une possibilité basse puissance supérieure car il peut se mettre en état de veille tant et aussi longtemps qu'aucune transmission n'est requise. Une interruption serait activée si un paquet est transmis et le système se remettrait en état de veille après son traitement.

4.7 Réalisation sur plate-forme de prototypage rapide

Tout au long de cette recherche deux prototypes ont été développés sur des plates-formes de prototypage rapide pour évaluer les fonctionnalités de base et pour tester les cas limites plus rapidement qu'avec des simulations. Malheureusement, les outils logiciels permettant des simulations poussées sont lents à l'exécution et la simulation de plusieurs secondes est impensable. Le développement par prototypage rapide est donc une solution intéressante.

Que ce soit avec un type de FPGA ou pour un ASIC, des cellules spéciales sont utilisées pour l'implémentation des blocs mémoire. Dans le FPGA, le fournisseur donne les librairies d'utilisation parce que le circuit physique varie d'une famille de produits à une autre. Le choix du FPGA se fait donc une fois le design terminé. La structure logique est particulière à chacune des méthodes et doit être relativisée pour être comparée. Les cellules de base qui constituent les FPGA sont principalement composés

d'une table de décision (*Look-Up Table*-LUT) et d'une unité de mémoire comme une bascule. Quelques différences surviennent si l'on compare les produits des compagnie Xilinx et Altera qui sont les deux plus grands joueurs actuels dans le domaine de FPGA. Les différences majeures se distinguent dans les regroupements et les interconnexions. Chez Xilinx, les CLB (*Configurable Logic Blocks*) sont composés de quatre éléments de base, mais chez Altera, les LAB (*Logic Array Blocks*) en contiennent dix.

4.7.1 Prototype Xilinx avec lien USB

Un premier prototype a été réalisé sur une plate-forme de développement Xilinx avec un FPGA XCV2000E. L'architecture légèrement différente avait une interface USB (*Universal Serial BUS*) au lieu de l'interface sans-fil. Le but était d'avoir un point central de communication pour desservir plusieurs matrices indépendantes logiquement (Roy et Sawan 2005).

TAB. 4.2 Utilisation logique du premier prototype réalisé sur une plate-forme Xilinx

Éléments logiques		
Modules	REGs	LUTs
Communication	$\approx 0\%$	$\approx 0\%$
Compresseur (seuils)	$\approx 0\%$	$\approx 0\%$
Paquetisation	$\approx 1\%$	$\approx 1\%$
Bloc Canal (pipeline)	$\approx 36\%$	$\approx 10\%$
Total (32 canaux)	$\approx 71\%$	$\approx 23\%$

Le logiciel de gestion s'exécutait donc sur un ordinateur personnel offrant une grande puissance de calcul. Par contre, l'ordinateur est une des causes de non performance du système car le système d'exploitation (Linux) prend du temps pour réagir aux requêtes d'accès aux périphériques (USB dans notre cas). Suite à des tests exhaustifs, le projet s'est avéré non fonctionnel pour un grand nombre de canaux dû au

lien de communication. Bien que le lien USB offre un débit théorique suffisant, sa communication est faite de façon *half duplex* et chaque accès aux registres cause des dépassements de mémoire car le temps de changement de direction est trop long (5-10ms) comparativement à la quantité de mémoire dans le tampon de sortie du contrôleur numérique qui ne permet pas tant de latence.

Ce qui a amené l'équipe Cortisens à se requestionner sur le problème du lien de communication. Il a donc été décidé de développer un nouveau système avec lien sériel ne desservant qu'une matrice à la fois. Ce design est décrit au chapitre 3 en supposant que chaque matrice posséderait un lien de communication sans-fil indépendant et *full duplex.*

4.7.2 Prototype Altera avec processeur NIOS

Ce deuxième prototype, réalisé sur une plate-forme Altera, représente le système présenté au chapitre 3. C'est le dernier prototype qui a été réalisé avant le mise en oeuvre d'une réalisation ASIC (section 4.8). Le processus de ce prototype a subi une validation et une vérification conjointes avec les logiciels de synthèse Synopsys pour le ASIC et Quartus de Altera.

La surface estimée (tableau 4.3) est fournie par l'outil de synthèse Synopsys. Cette mesure est indicative et représente la surface qu'utilise chacun des éléments logiques utilisés référant à la librairie de design CMOS 0.18μm. Lors du placement et du routage du circuit, une surface supplémentaire deviendra nécessaire pour les tampons d'entrée/sortie, le routage, la propagation de l'horloge et l'alimentation. Ces données ne doivent donc pas être simplement additionner pour obtenir l'espace silicium requis. Une description exacte de l'espace requis est faite à la section 4.8 concernant la version

TAB. 4.3 Utilisation logique et surface estimée pour le système ENG

	Altera		Synopsys	
Modules	REGs	LUTs	Surface estimée (μm^2)	approx.
Récepteur sériel	457	541	50139	3%
Transmetteur sériel	54	647	17933	1%
Chaîne à délais	1024	0	70972	4%
DWT (+24 éléments DSP)	9149	1016	916233	56%
Détection	16	239	10262	1%
Gestion des mémoires	1160	2595	149848	9%
Génération des horloges	31	34	3483	1%
Blocs mémoire	0	0	407732	25%
Total (16 canaux)	**11891**	**5072**	**1626602**	

ASIC. Il est important de noter que le récepteur sériel qui occupe 3% du système inclus aussi les registres de configuration ; ce qui explique sa grande proportion face au transmetteur.

Le processeur NIOS est un processeur embarqué pouvant exécuter le code directement sur le FPGA Stratix1S40. La compagnie Altera fournit le bloc logique à la manière d'une boîte noire protégée par des "propriétés intellectuelles". Cette architecture a permis de développer une application logicielle minimaliste tout en validant la fonctionnalité matérielle. Les codes VHDL et C ont pu être intégrés en faisant abstraction du lien sans-fil qui n'est pas encore réalisé.

4.8 Réalisation d'un ASIC

Le circuit conçu et présenté au chapitre 3 a partiellement été réalisé en circuit intégré pour cette application spécifique. Une implémentation réduite est quand même utile pour la réalisation d'une preuve de concept et d'un prototype complet du système ENG.

4.8.1 Compromis architecturaux

Puisque la superficie était limitée pour la réalisation d'un premier prototype Cortisens en système intégré, certaines caractéristiques ont dû être exclues du design. Il a été convenu que les circuits assemblés pour ce prototype Cortisens devaient avoir une forme pyramidale, l'ordre d'empilement étant la matrice, le circuit analogique et mixte, le circuit numérique et finalement le circuit de communication sans-fil comme présenté au chapitre 3.

En ce qui concerne le contrôleur numérique, les modules consommant le plus d'espace sont les tampons d'entrée/sortie, le module processeur d'ondelettes et les cellules mémoires. Ces deux derniers modules augmentent en superficie en fonction du nombre de canaux gérés. Par contre, les blocs multiplicateurs et additionneurs du processeur d'ondelettes sont gourmands mais de grandeurs fixes grâce au pipelinage des données.

Bien que les tampons d'entrée/sortie prennent beaucoup d'espace, leur présence est essentielle dans un système multi-puces comme le prototype proposé. Puisque la matrice et le circuit analogique sont prévus pour 16 canaux, il est important que le circuit numérique aussi puisse assurer la gestion du même nombre de canaux. Il a donc été choisi d'éliminer le processeur d'ondelettes du prototype et d'adapter le reste du circuit pour permettre la gestion complète des canaux avec détection basée sur le seuil simple. En enlevant, le processeur DWT, la chaîne de délais n'est plus nécessaire non plus. Elle a donc été enlevée pour sauver de l'espace.

Bien que le processeur DWT ait prouvé son efficacité pour un système ENG, son utilité est complémentaire au reste du circuit : le reste doit être présent. Le module DWT a donc été enlevé dans le but d'obtenir un prototype sur puce le plus rapidement possible ; le but prioritaire de l'équipe Cortisens étant d'obtenir des résultats

in vivo pour corroborer les résultats préliminaires. La réalisation du circuit ASIC sans le processeur DWT permettra de valider plusieurs autres parties du système comme le circuit analogique, la matrice d'électrodes et la méthode d'assemblage. La réalisation du contrôleur numérique était donc essentielle même avec une section en moins. Les impacts majeurs de ce compromis sont que le taux de transmission minimal va suivre le taux d'activité neuronale, soit environ 10 % (car aucune compression supplémentaire ne sera faite) et que le module de détection sera plus sensible au bruit.

4.8.2 Module simple et sécuritaire sans mémoire RAM

Une solution minimaliste du système a aussi été prévue pour pallier une possible erreur due aux multiples variables inconnues du projet. En effet, les cellules mémoires fournies avec le compilateur Virage ne pouvaient pas être testées adéquatement avant la date limite car les dessins de niveau physique (*layout*) ne sont pas accessibles et le fournisseur (CMC Microsystems) remplace les modèles mémoire par les "vraies" cellules juste avant de soumettre le circuit final à la fonderie.

Le principe du circuit montré à la figure 4.5 est que l'écriture dans les bascules se fait en continu selon l'adresse d'entrée qui est un compteur. Il y a suffisamment de bascules pour retenir 32 échantillons, soit deux par canal. Lorsque l'une des moitiés des bascules est remplie et que l'autre moitié commence à l'être, le module sériel accède les valeurs retenues à l'aide d'un multiplexeur. Ainsi, la lecture ne s'effectue jamais en même temps que l'écriture et aucune métastabilité ne surviendra. La trame sérielle de sortie transmet un échantillon par canal d'entrée pour un total de seize échantillons. La seule contrainte de ce circuit est que la fréquence d'horloge sérielle doit être au moins dix fois plus rapide que l'horloge d'écriture. En effet, le temps requis pour l'écriture est de 16 cycles tandis que le temps de lecture et de transmission sérielle est

de $(16 * 8) + 32 = 160$ cycles étant donnés la précision de 8 bit/éch. et une trame de référence de 32 bits. Si l'horloge générée par le module d'horloge est utilisée, aucun problème n'est envisagé.

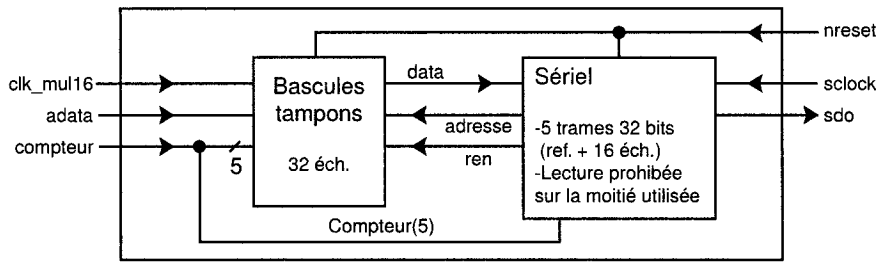


FIG. 4.5 Interface sérielle simple et sécuritaire avec changement de domaine d'horloge

4.8.3 Module avec détection et cellules mémoire

Le module est dérivé du système présenté au chapitre 3, le processeur d'ondelettes a été enlevé. Ainsi, la fréquence nécessaire devient clk_mul16 au lieu de clk_mul64 car les données à l'entrée arriveront de façon régulière et l'écriture peut donc se faire plus simplement et selon la fréquence d'entrée des données provenant du module analogique et mixte. Le système dérivé est montré à la figure 4.6.

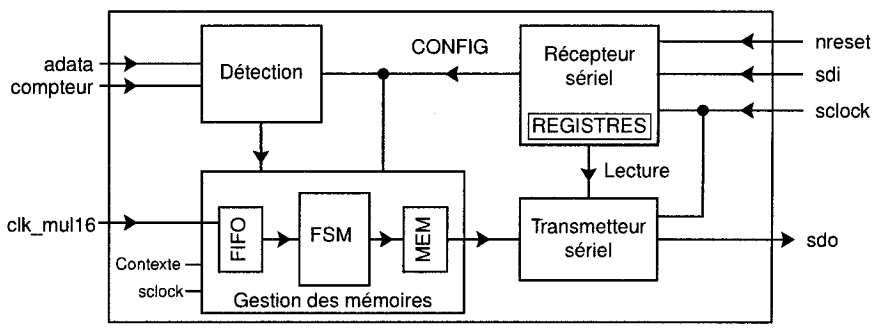


FIG. 4.6 Architecture du Système ASIC numérique

4.8.4 Les outils de développements

Une brève description des outils et étapes de développement avec les outils de synthèse et placement et routage est faite dans l'annexe I. Pour la synthèse, le logiciel Synopsys a été utilisé afin de traduire le code VHDL en un code représentatif au niveau des portes logiques de base de la technologie CMOS 0.18μm. Pour le placement et le routage le logiciel Encounter de Cadence a été utilisé autant pour la logique que pour la génération des arbres d'horloge et d'alimentation. Pour la vérification finale des règles de dessin physique (*Design Rule Check*-DRC) et pour la vérification finale du contenu logique (*Layout Versus Schematics*-LVS), l'outil Virtuoso de Cadence a été utilisé.

4.8.5 Utilisation des ressources

En joignant les deux modules indépendants pour la réalisation du prototype ASIC, on obtient le système global présenté à la figure 4.7. Le module de gestion des horloges est le même que celui présenté à la section 3.3.3.

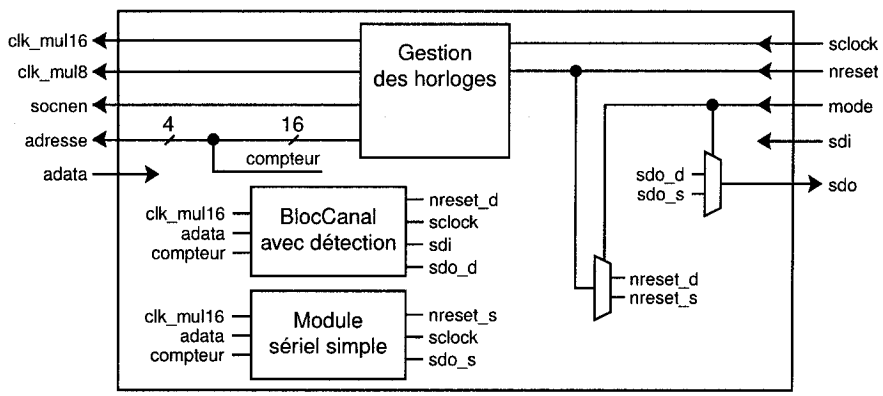


FIG. 4.7 Système ASIC conçu par l'équipe Cortisens

Les résultats du tableau 4.4 proviennent d'analyses faites par l'outil Encounter de

Cadence après le placement et le routage final de la puce. On constate que 40% de la superficie est occupée par les tampons d'entrée/sortie qui servent à fournir suffisamment de courant et protéger le circuit contre les décharges électrostatiques. En ajoutant la superficie des contacts métalliques vers l'extérieur de la puce et l'espace de garde entre les tampons et le coeurs, seulement 35% de la superficie reste pour le coeur du système ENG. Le placement du coeur a été réalisé avec une densité relativement élevée de 92% qui permet de minimiser les délais dû au routage des signaux.

La majeure partie du coeur est composée des cellules mémoire avec 69% de la superficie. Il est important de rappeler que les cellules mémoire ont été générées par le compilateur de mémoires Virage pour obtenir des blocs optimisés du point de vue surface et puissance. Le module de gestion des mémoires, qui gère aussi le contexte des paquets (numéros de séquence et l'étampe de temps) pour chacun des canaux, utilise 15% de la surface. C'est le module le plus critique du système. Les modules de communication récepteur et transmetteur occupent près de 7% du coeur tandis que les autres modules incluant l'arbre de propagation des horloges totalisent moins de 3%. La solution palliative simple et sécuritaire (Sériel Simple) utilise seulement 6% de la superficie et ne coûte pas trop cher à ajouter.

La figure 4.8 montre le système après le placement et le routage final du circuit. On voit clairement l'espace requis pour les contacts, les tampons d'entrée/sortie et le coeur. Les boîtes noires (*blackbox*) représentent les cellules fournies par le fournisseur auxquelles nous n'avions pas accès avant l'envoi de design final.

TAB. 4.4 Résumé de la surface d'implémentation sur silicium du prototype ASIC

Modules	Surface silicium (μm^2)	Utilisation
Récepteur sériel	25168	4 %
Transmetteur sériel	19509	3 %
Détection	3143	< 1%
Gestion des mémoires	90212	15 %
Blocs mémoire	407383	69 %
Génération d'horloge	3004	< 1%
Sériel Simple	36930	6 %
Arbres d'horloges (71 tampons de propagation)	5605	< 1%
autre logique	2121	< 1%
Total (coeur) : portes logiques	**593075**	densité : **92%**
Superficie du coeur	645760	35 %
Tampon d'entrée/sortie	733200	40 %
Autres (contacts + alimentations + espace de garde)	477040	25 %
Total : puce (1.6 x 1.16 mm)	**1856000**	

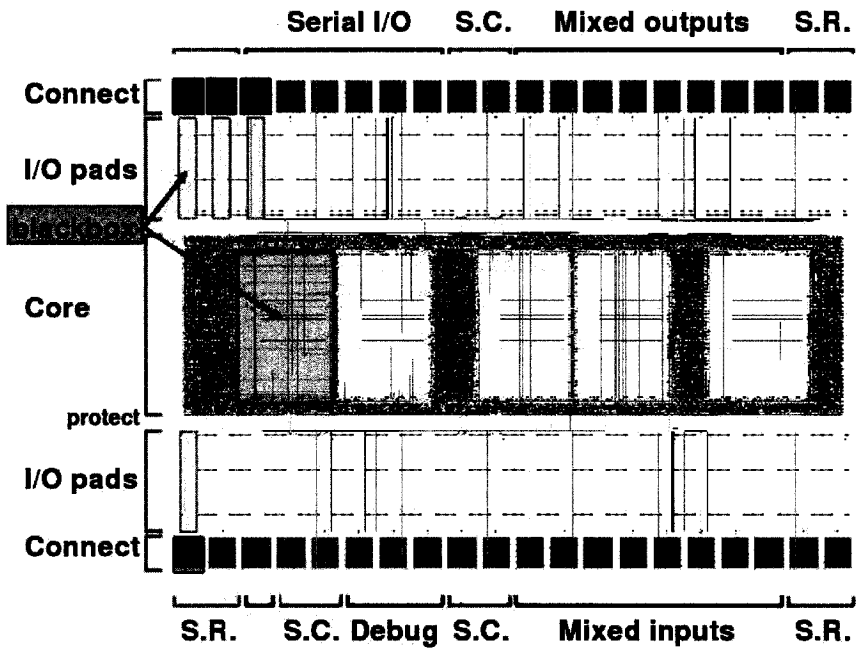


FIG. 4.8 Vue physique du système ASIC numérique

CONCLUSION

La recherche dans le domaine des interfaces cerveau-ordinateur se poursuit rigoureusement et permettra de pallier certaines dysfonctions humaines grâce aux prothèses corticales. Pour y parvenir, l'amélioration de précision, de qualité et de sensibilité des systèmes d'acquisition permettra une meilleure analyse des phénomènes électrophysiques. Les potentiels d'action résultent du transfert d'information entre les neurones conséquemment à l'activité ou à l'environnement du sujet portant la prothèse corticale. L'amélioration des techniques d'acquisition est donc tout indiquée pour un avancement dans le domaine.

Dans le chapitre 1, les différentes tâches nécessaires pour la réalisation des BCI (*Brain Computer Interface*) expliquent que l'acquisition corticale est une tâche importante du processus. La description des potentiels d'action nous a permis de décrire l'information que le système doit transmettre pour conclure que l'électroneurogramme (ENG) est une solution précise et intéressante pour les mesures expérimentales (*in vivo*).

Afin de compresser le signal, une technique simple et efficace consiste à faire une détection d'occurrence de potentiels d'action qui permet de minimiser le taux de transmission nécessaire pour desservir un maximum de canaux simultanés. Chacun des canaux représente le potentiel autour d'une électrode où plusieurs neurones peuvent être captés s'ils se trouvent à proximité. Les différents algorithmes présentés au chapitre 2 permettent, en plus d'aider à la détection de l'activité neuronale, de compresser le signal dans une même fenêtre d'échantillon pour réduire encore plus le taux de transmission sur le lien de communication.

Ce mémoire vise principalement l'évaluation et la réalisation de l'unité de contrôle et de compression de données. Le contrôleur est situé dans le système global au chapitre 3

avant d'y être décrit en détails. Les différents modules ont été optimisés individuellement en rapport avec leurs fonctionnalités afin d'aboutir à un système final logiquement minimaliste. En minimisant la logique utilisée, la surface d'implémentation ainsi que la consommation statique de puissance en seront minimisées.

L'erreur quadratique moyenne (*Mean Square Error*-MSE), le taux de compression, la sensibilité et la sélectivité sont utilisés comme point de comparaison pour finalement conclure que la méthode par transformée en ondelettes (*Discrete Wavelet Transform*-DWT) est particulièrement efficace dans un système d'acquisition neuronal. Sa qualité de bonne localisation temporelle jointe à une bonne résolution fréquentielle permet une détection fiable des potentiels d'action. L'équipe Cortisens a choisi l'ondelette mère Daubechie (db4) pour l'implémentation matérielle selon une disposition en arbre optimisée pour une résolution adéquate dans la bande fréquentielle d'intérêt sous les 4 kHz.

En plus de sa bonne sensibilité, même avec beaucoup de bruit, la méthode avec DWT offre une bonne compression tout en débruitant le signal échantillonné. Les simulations avec des signaux expérimentaux ont prouvé qu'un taux de transmission de moins de 2% est suffisant pour reconstruire le signal avec une très bonne qualité. De plus, cette méthode permet aussi de transmettre sans perte le signal si l'utilisateur le désire. La méthode avec DWT s'impose donc comme une excellente méthode d'acquisition expérimentale.

Les différentes techniques de détection de l'activité neuronale ont aussi été comparées pour conclure que la DWT se reconnaît comme pré-traitement approprié. Les méthodes adaptatives ayant fait leurs preuves selon les références consultées mais pas dans les simulations effectuées, une approche par configuration logicielle permet un compromis acceptable. Cela provient du fait que les simulations ont été effectuées

selon une approche post-acquisition qui permet un traitement global du signal avec une connaissance du niveau de bruit ; ce qui n'est pas le cas dans un système temps réel. Étant donné que le niveau de bruit ne devrait pas changer radicalement entre chaque échantillon, un seuil simple peut être utilisé avec une boucle de rétroaction joint avec un lien de communication entièrement bidirectionnel (*full duplex*). Le lien *full duplex* permet de reconfigurer les seuils pour chacun des canaux sans influencer le débit sortant de données.

Le système de contrôle et de compression numérique décrit au chapitre 3 a donc été réalisé : en premier lieu, dans sa globalité sur une plate-forme de prototypage rapide avec FPGA pour aboutir à une implémentation simplifiée en ASIC. Ce dernier, en cours de fabrication, permettra la réalisation d'un premier prototype complet pour l'équipe Cortisens par l'assemblage de plusieurs puces et de la matrice d'électrode avec une technique par superposition des puces.

Beaucoup de travail reste encore à être accompli avant d'obtenir un implant cortical d'acquisition complet. Premièrement, des signaux acquisitionnés *in vivo* devront corroborer les simulations effectuées. Deuxièmement, un traitement comme la DWT devra être ajouté au prototype ASIC par l'ajout d'un module numérique ou de filtres analogiques dédiés.

Troisièmement, le module sans-fil devra être intégré au système et une grande latitude est possible tant que le débit de données est maximisé. Un chevauchement entre les parties contrôleur numérique et lien sans-fil pourrait être bénéfique dans le cas où une correction d'erreur serait être ajoutée au protocole de communication, par exemple avec un CRC (*Cyclic Redundancy Check*).

Quatrièmement, le système global devrait être intégré sur la même puce pour éviter

l'usage inutile des tampons d'entrée/sortie dû aux échanges inter-puces.

Et finalement, une méthode algorithmique mettant en relation l'activité neuronale spatiale pourrait améliorer le taux de compression obtenu en débruitant et en évitant la redondance due aux relations inter-électrodes.

Ce mémoire propose une méthode numérique d'acquisition des signaux corticaux pour la réalisation d'une BCI. Différentes techniques et méthodes y sont analysées pour une solution minimisant le taux de transfert pour maximiser l'usage du lien de communication et ainsi permettre une plus grande densité pour un même lien de communication. Ce mémoire n'a pas la prétention d'offrir l'unique solution aux problèmes d'acquisition de signaux neuronaux, mais bien d'offrir une solution mathématiquement logique, précise et réalisable dans un circuit intégré.

RÉFÉRENCES

[Adjouadi *et al.* 2004] ADJOUADI, M., SANCHEZ, D., CABRERIZO, M., AYALA, M., JAYAKAR, P., YAYLALI, I. ET BARRETO, A. 2004. Interictal spike detection using the walsh transform. *IEEE Transaction on Biomedical Engineering*, 51, 868–872.

[Batista *et al.* 2002] BATISTA, A., COMBO, A., SOUSA, J. ET VARANDAS, C. 2002. A distributed, hardware reconfigurable and packet switched real-time control and data acquisition system. *Fusion Engineering and Design*, 60, 443–448.

[Blankertz *et al.* 2001] BLANKERTZ, B., CURIO, G. ET MULLER, K.-R. 2001. Classifying single trial eeg : Towards brain computer interfacing. *Advances in Neural Information Processing Systems.*

[Boahen 2000] BOAHEN, K. A. 2000. Point-to-point connectivity between neuromorphic chips using address events. *IEEE Transaction on Circuits and Systems-II, Analog and Digital Signal Processing*, 47, 416–434.

[Bossetti *et al.* 2004] BOSSETTI, C. A., CARMENA, J. M., NICOLELIS, M. A. L. ET WOLF, P. D. 2004. Transmission latencies in a telemetry-linked brain-machine interface. *IEEE Transaction on Biomedical Engineering*, 51, 919–924.

[Branner *et al.* 2004] BRANNER, A., STEIN, R. B., FERNANDEZ, E., AOYAGI, Y. ET NORMANN, R. A. 2004. Long-term stimulation and recording with a penetrating microelectrode array in cat sciatic nerve. *IEEE Transaction on Biomedical Engineering*, 51, 146–157.

[Buzsaki 2004] BUZSAKI, G. 2004. Large-scale recording of neuronal ensembles. *Nature Neuroscience*, 7, 446–451.

[Chandra et Optican 1997] CHANDRA, R. ET OPTICAN, L. M. 1997. Detection, classification, and superposition resolution of action potentials in multiunit single-channel recordings by an on-line real-time neural network. *IEEE Transaction on Biomedical Engineering*, 44, 403–412.

[Chueh et Hatfield 2002] CHUEH, H.-T. ET HATFIELD, J. V. 2002. A real-time data acquisition system for a hand-held electronic nose (h^2en). *Sensors and Actuators B*, 83, 262–269.

[Costa et Fiori 2001] COSTA, S. ET FIORI, S. 2001. Image compression using principal component neural networks. *Image and Vision Computing*, 19, 649–668.

[Dayan et Abbott 2001] DAYAN, P. ET ABBOTT, L. F. 2001. *Theoretical Neuroscience : Computational and Mathematical Modeling of Neural Systems.* The MIT Press, première édition.

[Dornhege *et al.* 2002] DORNHEGE, G., BLANKERTZ, B., CURIO, G. ET MULLER, K.-R. 2002. Combining features for bci. *Advances in Neural Information Processing Systems.*

[Dornhege *et al.* 2003] DORNHEGE, G., BLANKERTZ, B., CURIO, G. ET MULLER, K.-R. 2003. Increase information transfer rates in bci by csp extension to multi-class. *Advances in Neural Information Processing Systems.*

[Dumortier *et al.* 2006] DUMORTIER, C., GOSSELIN, B. ET SAWAN, M. 2006. Low-power implantable microsystem intented to multichannel cortical recording. *IEEE International Symposium on Circuits and Systems.*

[Ferrandi *et al.* 1998] FERRANDI, F., FUMMI, F., MACII, E., PONCINO, M. ET SCIUTO, D. 1998. Power estimation of behavioral descriptions. *IEEE Design Automation and Test in Europe*, 762–766.

[Folkers et Hofmann 2001] FOLKERS, A. ET HOFMANN, U. G. 2001. A multi-channel data acquisition and analysis system based on off-the-shelf dsp boards. *ECMCS*.

[Folkers *et al.* 2003] FOLKERS, A., MOSCH, F., MALINA, T. ET HOFMANN, U. G. 2003. Realtime bioelectrical data acquisition and processing from 128 channels utilizing the wavelet-transformation. *Neurocomputing*, 52, 247–254.

[French *et al.* 2003] FRENCH, A. S., HOGER, U., ICHI SEKIZAWA, S. ET TORKKELI, P. H. 2003. A context-free data compression approach to measuring information transmission by action potentials. *BioSystems*, 69, 55–61.

[Georgopoulos *et al.* 1986] GEORGOPOULOS, A. P., SCHWARTZ, A. B. ET KETTNER, R. E. 1986. Neuronal population coding of movement direction. *Science*, 233, 1416–1419.

[Gerstner et Kistler 2002] GERSTNER, W. ET KISTLER, W. M. 2002. *Spiking neuron models Single Neurons, Populations, Plasticity.* Cambridge University Press.

[Gosselin *et al.* 2004a] GOSSELIN, B., SIMARD, V., ROY, J.-F., MARROUCHE, W., DUMORTIER, C. ET SAWAN, M. 2004a. Multichannel wireless cortical recording : Circuits, system design and assembly challenges. *Biomedical Circuits and Systems, IEEE International Workshop on.*

[Gosselin *et al.* 2004b] GOSSELIN, B., SIMARD, V. ET SAWAN, M. 2004b. Low-power implantable microsystem intented to multichannel cortical recording. *IEEE International Symposium on Circuits and Systems.*

[Guillory et Normann 1999] GUILLORY, K. ET NORMANN, R. 1999. A 100-channel system for real time detection and storage of extracellular spike waveforms. *Journal of Neuroscience Methods*, 91, 21–29.

[Hall *et al.* 1997] HALL, P., PENEV, S., KERKYACHARIAN, G. ET PICARD, D. 1997. Numerical performance of block thresholded wavelet estimators. *Statistics and Computing*, 7, 115–124.

[Harrison 2003] HARRISON, R. R. 2003. A low-poer integrated circuit for adaptative detection of action potentials in noisy signals. *Proceeding of the 25th Annual International Conference of the IEEE-EMBS*, 3325–3328.

[Hastie *et al.* 2001] HASTIE, T., TIBSHIRANI, R. ET FREIDMAN, J. 2001. *The Elements of Statistical Learning : Data Mining, Inference and Prediction.* Springer, première édition.

[Haykin 1998] HAYKIN, S. 1998. *Neural Networks, A Comprehensive Foundation.* Prentice Hall, seconde édition.

[Hebb 1949] HEBB, D. 1949. *Organization of Behavior.* John Wiley and Son.

[Hinterberger *et al.* 2004] HINTERBERGER, T., SCHMIDT, S., NEUMANN, N., MELLINGER, J., BLANKERTZ, B., CURIO, G. ET BIRBAUMER, N. 2004. Brain-computer communication and slow cortical potentials. *IEEE Transaction on Biomedical Engineering*, 51, 1011–1018.

[Huffman 1952] HUFFMAN, D. A. 1952. A method for the construction of minimum-redundancy codes. *Proceeding of the I.R.E.*, 1098–1101.

[Hulata *et al.* 2002] HULATA, E., SEGEV, R. ET BEN-JACOB, E. 2002. A method for spike sorting and detection based on wavelet packets and shannon's mutual information. *Journal of Neuroscience Methods*, 117, 1–12.

[Hulata *et al.* 2000] HULATA, E., SEGEV, R., SHAPIRA, Y., BENVENISTE, M. ET BEN-JACOB, E. 2000. Detection and sorting of neural spikes using wavelet packets. *The American Physical Society*, 85, 4637–4640.

[Kadambe et Srinivasan 2005] KADAMBE, S. ET SRINIVASAN, P. 2005. Adaptive wavelets for signal classification and compression. *International Journal of Electronics and Communications*, 1.

[Kaiser 1990] KAISER, J. F. 1990. On a simple algorithm to calculate the 'energy' of a signal. *International Conference on Acoustics, Speech, and Signal Processing*, 1, 381–384.

[Kalayci *et al.* 1994] KALAYCI, T., OZDAMAR, O. ET ERDOL, N. 1994. The use of wavelet transform as a preprocessor for the neural network detection of eeg spikes. *IEEE.*

[Kim et Kim 2000] KIM, K. H. ET KIM, S. J. 2000. Neural spike sorting under nearly 0-db signal-to-noise ratio using nonlinear energy operator and artificial neural-network classifier. *IEEE Transaction on Biomedical Engineering*, 47.

[Leon-Garcia 1994] LEON-GARCIA, A., éditeur 1994. *Probability and Random Processes for Electrical Engineering.* Addison-Wesley Publishing Company, seconde édition.

[Letelier et Weber 2000] LETELIER, J. C. ET WEBER, P. P. 2000. Spike sorting based on discrete wavelet transform coefficients. *Journal of Neuroscience Methods*, 101, 93–106.

[Lewicki 1998] LEWICKI, M. S. 1998. A review of methods for spike sorting : the detection and classification of neural action potentials. *Network : Computation in Neural Systems*, 9, R52–R78.

[Mackay 2003] MACKAY, D. J. 2003. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press.

[Mallat 1989] MALLAT, S. 1989. A theory for multiresolution signal decomposition : The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11.

[Mallat 1999] MALLAT, S. 1999. *A Wavelet Tour of Signal Processing.* Academic Press, http ://www.cmap.polytechnique.fr/ mallat/book.html, seconde édition.

[Meinicke *et al.* 2002] MEINICKE, P., KAPER, M., HOPPE, F., HEUMANN, M. ET RITTER, H. 2002. Improving transfer rates in brain computer interfacing : A case study. *Advances in Neural Information Processing Systems.*

[Musial *et al.* 2002] MUSIAL, P., BAKER, S., GERSTEIN, G., KING, E. ET KEATING, J. 2002. Signal-to-noise ratio improvement in multiple electrode recording. *Journal of Neuroscience Methods*, 115, 29–43.

[Nicolelis 1999] NICOLELIS, M. A., éditeur 1999. *Methods for Neural Ensemble Recordings.* CRC Press.

[Obeid *et al.* 2004a] OBEID, I., NICOLELIS, M. A. ET WOLF, P. D. 2004a. A low power multichannel analog front end for portable neural signal recordings. *Journal of Neuroscience Methods*, 133, 27–32.

[Obeid *et al.* 2004b] OBEID, I., NICOLELIS, M. A. ET WOLF, P. D. 2004b. A multichannel telemetry system for single unit neural recordings. *Journal of Neuroscience Methods*, 133, 33–38.

[Obeid et Wolf 2004] OBEID, I. ET WOLF, P. D. 2004. Evaluation of spike-detection algorithms for a brain-machine interface application. *IEEE Transaction on Biomedical Engineering*, 51, 905–911.

[Oweiss et Anderson 2000] OWEISS, K. G. ET ANDERSON, D. J. 2000. A new approach to array denoising. *IEEE.*

[Pang *et al.* 2003] PANG, C. C. C., UPTON, A. R. M., SHINE, G. ET KAMATH, M. V. 2003. A comparison of algorithms for detection of spikes in the electroencephalogram. *IEEE Transaction on Biomedical Engineering*, 50, 521–526.

[Pigeon 2004] PIGEON, S. 2004. *Conception et fabrication d'une matrice de microélectrodes corticales implantables.* Mémoire de maîtrise, École Polytechnique de Montréal.

[Pigeon *et al.* 2003] PIGEON, S., MEUNIER, M., SAWAN, M. ET MARTEL, S. 2003. Design and fabrication of a microelectrode array dedicated for cortical electrical stimulation. *CCECE*, 813–816.

[Pouzat 2004] POUZAT, C. 2004. Technique(s) for spike-sorting. Rapport technique, Laboratoire de Physiologie Cérébrale, Université René Descartes (Paris V), http ://www.biomedicale.univ-paris5.fr/physcerv/Spike-O-Matic.html.

[Pouzat *et al.* 2004] POUZAT, C., DELESCLUSE, M., VIOT, P. ET DIEBOLT, J. 2004. Improved spike-sorting by modeling firing statistics and burst-dependent spike amplitude attenuation : A markov chain monte carlo approach. *Journal of Neurophysiology*, 91, 2910–2928.

[Pouzat *et al.* 2002] POUZAT, C., MAZOR, O. ET LAURENT, G. 2002. Using noise signature to optimize spike-sorting and to assess neuronal classification quality. *J Neurosci Methods*, 122, 43–57.

[Ramón y Cajal 1911] RAMÓN Y CAJAL, S. 1911. *Histologie du Système Nerveux de l'Homme et des Vertébrés.* Maloine.

[Roy et Sawan 2005] ROY, J.-F. ET SAWAN, M. 2005. A fully reconfigurable controller dedicated to implantable recording devices. *IEEE-NEWCAS Conference, The 3rd International.*

[Sage 1990] SAGE, A. P., éditeur 1990. *Concise Encyclopedia of Information Processing in Systems snd Organizations.* Pergamon.

[Schwartz *et al.* 2004] SCHWARTZ, A. B., D. W, M. ET REINA, G. A. 2004. Differential representation of perception and action in the frontal cortex. *Science*, 303, 380–383.

[Segura-Juarez *et al.* 2004] SEGURA-JUAREZ, J. J., CUESTA-FRAU, D., SAMBLAS-PENA, L. ET ABOY, M. 2004. A microcontroller-based portable electrocardiograph recorder. *IEEE Transaction on Biomedical Engineering*, 51, 1686–1690.

[Shannon 1948] SHANNON, C. E. 1948. The mathematical theory of communication. *Bell System Technical Journal.*

[Simard 2005] SIMARD, V. 2005. *Transformée en ondelettes pour un système d'acquisition de signaux corticaux implantable.* Mémoire de maîtrise, École Polytechnique de Montréal.

[Wilson et Emerson 2002] WILSON, S. B. ET EMERSON, R. 2002. Spike detection : a review and comparison of algorithms. *Clinical Neurophysiology*, 113, 1873–1881.

[Wise *et al.* 2004] WISE, K., J.ANDERSON, HETKE, J., , D. K. ET NAJAFI, K. 2004. Wireless implantable microsystems : High-density electronic interfaces to the nervous system. *Proceeding of the IEEE.* vol. 92, 76–97. Invited Paper.

[Wolpaw *et al.* 2002] WOLPAW, J. R., BIRBAUMER, N., MCFARLAND, D. J., PFURTSCHELLER, G. ET VAUGHAN, T. M. 2002. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113, 767–791.

[Wolpaw *et al.* 2000] WOLPAW, J. R., MCFARLAND, D. J. ET VAUGHAN, T. M. 2000. Braincomputer interface research at the wadsworth center. *IEEE Transaction on Rehabilitation Engineering*, 8, 222–226.

[Wolpaw *et al.* 2003] WOLPAW, J. R., MCFARLAND, D. J., VAUGHAN, T. M. ET SCHALK, G. 2003. The wadsworth center braincomputer interface (bci) research and development program. *IEEE Transaction on Neural Systems and Rehabilitation Engineering*, 11, 204–207.

[Wood *et al.* 2004] WOOD, F., BLACK, M. J., VARGAS-IRWIN, C., FELLOWS, M. ET DONOGHUE, J. P. 2004. On the variability of manual spike sorting. *IEEE Transaction on Biomedical Engineering*, 51, 912–918.

ANNEXE I

PROCESSUS DE PLACEMENT ET ROUTAGE DE LA PUCE ASIC

Cette annexe montre les étapes de développement avec les outils de design assistés par ordinateur.

La première étape consiste à choisir l'emplacement des tampons d'entrées/sorties du système ainsi que le mémoire. Cette tâche est réalisé en relation avec le positionnement des signaux correspondants sur la matrice et le système analogique et mixte. Une fois fait, les anneaux d'alimentation sont placés pour le coeur logique après avoir respecté une distance de sécurité évitant les possibles décharges électrostatiques. Étant donné la grosseur des cellule mémoire comparativement à la surface totale, les blocs mémoire sont placés en premier en plaçant les lignes d'alimentation tout près (figure I.1).

FIG. I.1 Placement des mémoires et des lignes d'alimentation du coeur

Les unités logiques sont ensuite automatiquement placées (figure I.2).

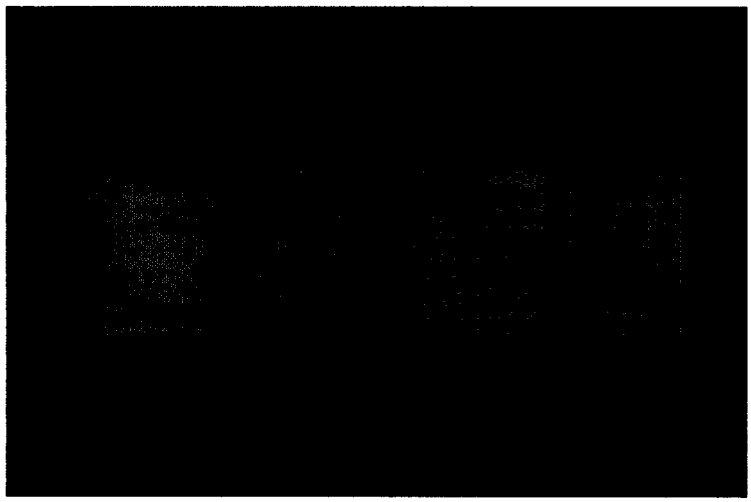


FIG. I.2 Placement des unités logiques de base

Ensuite, les tampons de synchronisation pour l'arbre d'horloge sont calculés, calibrés puis placés (figure I.3).

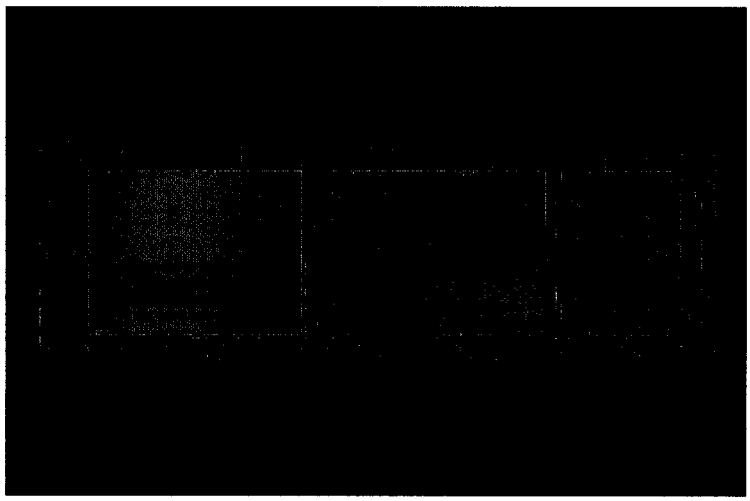


FIG. I.3 Routage de l'arbre d'horloge

L'alimentation est routée avant la logique vu que les fils nécessitent des précautions particulières (figure I.4).

FIG. I.4 Routage des alimentations

Finalement, la logique du design est routée (figure I.5)

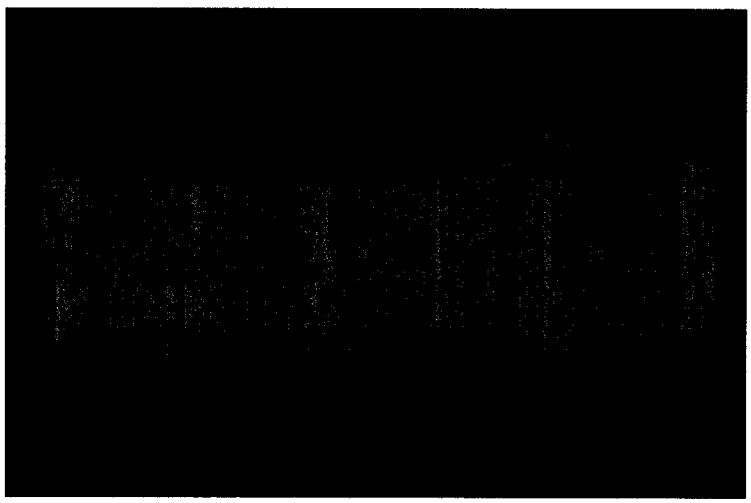


FIG. I.5 Routage final des cellules logiques

Après l'outil Cadence Virtuoso, on obtient un système final avec la réprésentation

physique (layout) de toute la puce incluant les contacts métalliques. Les règles de design (*Design Rules Check*-DRC) et de validation schématique (*Layout Versus Schematics*-LVS) sont toutes respectées. Une photo de la puce est montrée à la figure I.6.

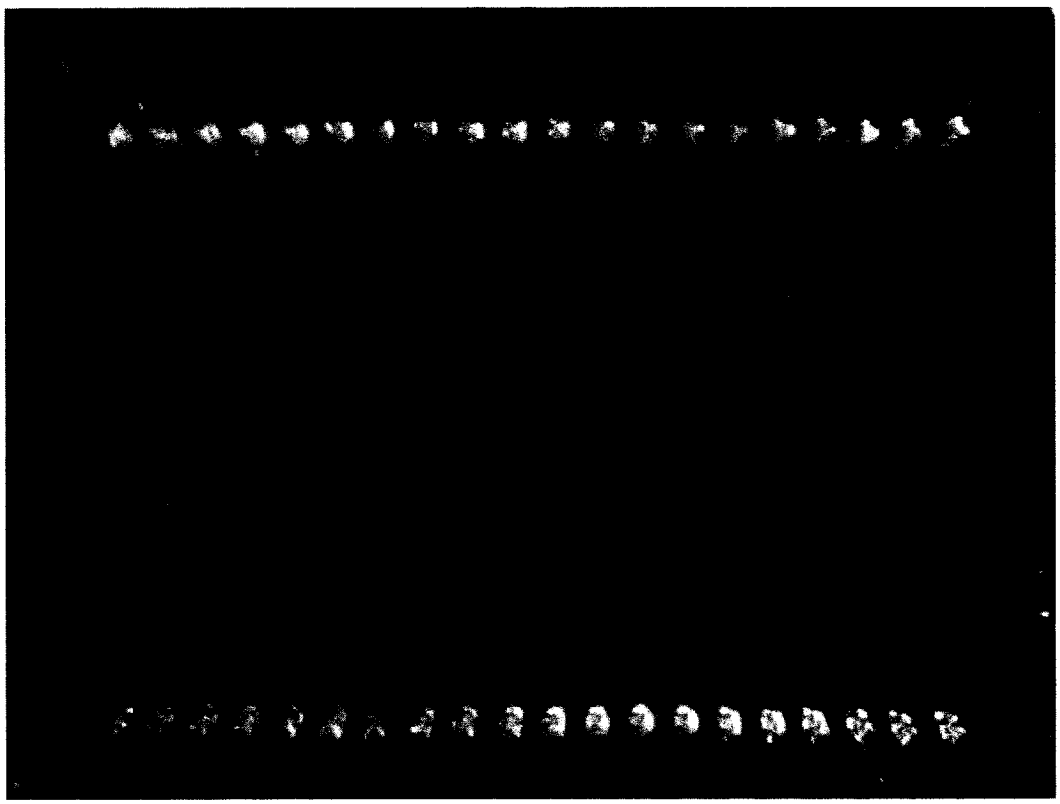


FIG. I.6 Photo de la puce une fois fabriquée

ANNEXE II

PORT VHDL DU CONTRÔLEUR NUMÉRIQUE

Afin de faciliter le design en langage matériel (VHDL) et la compréhension générale, un fichier de référence de base a été créé pour réunir en un seul point les constantes, les déclarations de types et les déclarations de composants. De cette façon chacun des composants n'avait qu'à inclure ce "*package*" pour avoir accès à ces données globales au projet. Les déclarations sont présentées dans cette annexe.

```
-- Rassemblement de matrice de canaux
constant lChanadrs: integer := 4;
constant nChannel : integer := 2**lChanadrs;

-- Bloc Canal description
type regbc_type
  is array (nChannel-1 downto 0) of std_logic_vector(23 downto 0);
constant lnwindow : integer := 6;        -- 2^6 = 64
constant nwindow : integer := 2**lnwindow;
constant lnfifo  : integer := 4;         -- 2^4 = 16
constant nfifo   : integer := 2**lnfifo;

-- Accès mémoire pour le tampon de communication
constant lmemdata : integer := 32;
constant lmemadrs : integer := lnwindow + lChanadrs - 2;

-- Analog Font End
constant lanalogdata : integer := 8;

-- Diviseur de fréquence par compteur libre pour sclock
constant lClockDivider : integer := 8;  -- from 16.6MHz to 32kHz
constant lTimeStamp    : integer := 12;
constant lGlobalCount  : integer := lChanadrs+lTimeStamp;
```

```
-- Interface fifocopy et bc2seriel module
-- dataflag : pour indiquer les coefficients DWT choisis parmis
-- les nwindow/2 échantillons possibles de la fenêtre
-- range DOIT être égal à (lmemdata-1 downto 0)
type fifo_compression_dataflag_type is
  array(nChannel-1 downto 0) of std_logic_vector((nwindow/2)-1 downto 0);
-- seqnum : pour indiquer le combientième packet à être transmis
type fifo_seqnum_type is
  array(nChannel-1 downto 0) of std_logic_vector(3 downto 0);
-- length : pour indiquer le combientième packet à être transmis
type fifo_length_type is
  array(nChannel-1 downto 0) of std_logic_vector(lnwindow downto 0);
type fifo_timestamp_type is
  array(nChannel-1 downto 0) of std_logic_vector(11 downto 0);

type fifo2col_type is
  record
    -- Lecture de la mémoire
    rdata      : std_logic_vector(lmemdata-1 downto 0);
    -- Information pour les paquets
    packetflag : std_logic_vector(nChannel-1 downto 0);
    dataflag   : fifo_compression_dataflag_type;
    seqnum     : fifo_seqnum_type;
    length     : fifo_length_type;
    timestamp  : fifo_timestamp_type;
  end record;

type col2fifo_type is
  record
    -- Contrôle pour la lecture des paquets
    ren    : std_logic;
    done   : std_logic_vector(nChannel-1 downto 0);
    radrs  : std_logic_vector(lChanadrs+lnwindow-3 downto 0);
  end record;

type reg2col_type is
  record
    data    : std_logic_vector(15 downto 0);
    adrs    : std_logic_vector(lChanadrs-1 downto 0);
    enable  : std_logic;
    request : std_logic;
  end record;
```

Le récepteur sériel

```
component seriel_slave
  port (
    -- Port Sériel d'entrée
    nreset    : in std_logic;      -- Asynchrone
    sclock    : in std_logic;      -- Horloge sérielle
    sdi       : in std_logic;      -- Donnée sérielle
    -- Retour de lecture
    reg_data  : out reg2col_type; -- Données, adresse et requête
    reg_done  : in  std_logic;    -- Requête terminée
    -- Registres
    canal_on  : out std_logic_vector(nChannel-1 downto 0);
    registres : out regbc_type);
end component;
```

Le transmetteur sériel

```
component bc2seriel
  port (
    -- Port Sériel de sortie
    nreset    : in  std_logic;      -- Asynchrone
    sclock    : in  std_logic;      -- Horloge sérielle
    sdo       : out std_logic;      -- Donnée sérielle
    -- Retour de lecture
    reg_data  : in  reg2col_type;   -- Données, adresse et requête
    reg_done  : out std_logic;      -- Requête terminée
    -- Accès mémoire, interface FIFO
    fifo2col  : in  fifo2col_type;  -- Contexte de paquet
    col2fifo  : out col2fifo_type);  -- Confirmation de transmission
end component;
```

Le module de processeur d'ondelettes

```
component DWT_3niveaux
  generic (lgt_in   : integer := 8;
           lgt_coef : integer := 8;
           lgt_dwt  : integer := 8);
  port (
    -- Port d'entrée
```

```
    nreset    : in  std_logic;  -- Asynchrone
    clk_mul16 : in std_logic;   -- Horloge d'entrée des données
    clk_mul8  : in std_logic;   -- Horloge du calculateur (pair/impair)
    -- Commande des démultiplixeurs
    ctrl1     : in std_logic;   -- compteur = f_ech/4
    ctrl2     : in std_logic;   -- compteur = f_ech/8
    data : in std_logic_vector (lgt_in-1 downto 0);
    -- Port de sortie
    CD1 : out std_logic_vector (lgt_dwt-1 downto 0);
    CD2 : out std_logic_vector (lgt_dwt-1 downto 0);
    CD3 : out std_logic_vector (lgt_dwt-1 downto 0);
    CA3 : out std_logic_vector (lgt_dwt-1 downto 0));
end component;
```

Le module de détection

```
component detect
  port (
    nreset    : in  std_logic;      -- Asynchrone
    clk_mul64 : in std_logic;       -- Horloge pour le multiplexage
    -- Données analogiques
    data      : in  std_logic_vector(lanalogdata-1 downto 0);
    ddata     : in  std_logic_vector(lanalogdata-1 downto 0);
    -- Transformée en ondelettes
    wt_cd1       : in  std_logic_vector(lanalogdata-1 downto 0);
    wt_cd2       : in  std_logic_vector(lanalogdata-1 downto 0);
    wt_cd3       : in  std_logic_vector(lanalogdata-1 downto 0);
    wt_ca3       : in  std_logic_vector(lanalogdata-1 downto 0);
    -- Configuration et multiplexage selon le mode
    regbc        : in  regbc_type;
    wtcount      : in  std_logic_vector(lChanadrs+lnfifo+1 downto 0);
    -- Accès à la mémoire FIFO
    detect_spike : out std_logic;    -- Seuil dépassé
    detect_valid : out std_logic;    -- Écriture dans le FIFO
    detect_wadrs : out std_logic_vector(lChanadrs+lnfifo-1 downto 0);
    detect_out   : out std_logic_vector(lanalogdata-1 downto 0));
end component;
```

Le module de gestion des mémoires

```
component fifocopy
  port (
    nreset     : in std_logic;       -- Asynchrone
    clk_mul64  : in std_logic;       -- Horloge pour le multiplexage
    sclock     : in std_logic;       -- Horloge sérielle
    regbc      : in regbc_type;
    -- Contexte Global
    channelnum    : in std_logic_vector(lChanadrs-1 downto 0);
    timestamp     : in std_logic_vector(11 downto 0);
    -- Écriture dans le FIFO
    detect_spike : in std_logic;
    detect_valid : in std_logic;
    detect_wadrs : in std_logic_vector(lChanadrs+lnfifo-1 downto 0);
    detect_wdata : in std_logic_vector(lanalogdata-1 downto 0);
    -- Lecture par le transmetteur sériel
    fifo2col      : out fifo2col_type;
    col2fifo      : in  col2fifo_type);
```