POLYPUBLIE
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

| | |
|---|---|
| **Titre:** Title: | Réalisation et tests d'un système implantable dédié à la stimulation neurale sélective |
| **Auteur:** Author: | Simon Robin |
| **Date:** | 1998 |
| **Type:** | Mémoire ou thèse / Dissertation or Thesis |
| **Référence:** Citation: | Robin, S. (1998). Réalisation et tests d'un système implantable dédié à la stimulation neurale sélective [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/6847/ |

**Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

| | |
|---|---|
| **URL de PolyPublie:** PolyPublie URL: | https://publications.polymtl.ca/6847/ |
| **Directeurs de recherche:** Advisors: | Mohamad Sawan |
| **Programme:** Program: | Non spécifié |

**Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal**
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca

UNIVERSITÉ DE MONTRÉAL

# RÉALISATION ET TESTS D'UN SYSTÈME IMPLANTABLE DÉDIÉ À LA STIMULATION NEURALE SÉLECTIVE.

SIMON ROBIN
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE ET DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GENIE ÉLECTRIQUE)
JUILLET 1998

© Simon Robin, 1998.

National Library of Canada
Bibliothèque nationale du Canada

Acquisitions and Bibliographic Services
Acquisitions et services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-37461-0

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

RÉALISATION ET TESTS D'UN SYSTÈME IMPLANTABLE
DÉDIÉ À LA STIMULATION NEURALE SÉLECTIVE.

Présenté par : ROBIN Simon
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. BOIS Guy, Ph.D., président
M. SAWAN Mohamad, Ph.D., membre et directeur de recherche
M. SLAMANI Mustapha, Ph.D., membre

# DÉDICACE

*Ce mémoire est dédié à tous ceux qui m'ont encouragé dans la poursuite de ces travaux dont pricipalement mon épouse Esther ainsi que mes parents, Charles et Colette.*

## REMERCIEMENTS

Je tiens premièrement à remercier mon directeur de recherche, M. Mohamad Sawan pour son encadrement, ses conseils, la confiance qu'il eut en moi et pour son support financier. Mohamad est toujours disponible pour ses étudiants malgré son horaire très chargé.

Je tiens ensuite à remercier les membres de l'équipe de recherche en neurotechnologies (PolySTIM) qui m'ont aidé dans la conception et la réalisation de l'implant sur circuit intégré. Alexandre Beauchamp-Parent pour son aide au niveau de la logique de contrôle du circuit intégré de l'implant. Sylvain Bourret pour son aide au niveau de l'étage de sortie du circuit intégré avec sa source de courant programmable ainsi que les nombreuses heures qu'il a passées pour le "layout" de ce circuit. Abdelouahab Djemouai pour son aide dans l'électronique analogique d'entrée. Pierre Vaillancourt pour son aide au niveau de la logique de réception des données.

Toute l'équipe médicale de l'Université McGill et de l'Hôpital Royal Victoria soit Mostafa M. Elhilali, Hassan S. Shaker, Le Mai Tu, Tarek Abdel-Baky, Mahmoud Abdel-Gawad ainsi que tous les techniciens et techniciennes du Centre de Recherche sur les animaux.

Je tiens finalement à remercier Benoit Provost, Jean-François Harvey et Stéphane Boyer pour leur aide dans la réalisation du système de stimulation et tout spécialement Gaétan Décarie pour son superbe travail dans la réalisation et le montage des circuits des contrôleurs et de l'implant. Je tiens aussi à remercier la fondation canadienne des maladies du rein.

## RÉSUMÉ

L'utilisation de stimulateurs électriques implantables sert au traitement de maladies diverses. Plusieurs applications médicales ont maintenant recours aux implants électroniques dont le rétablissement des fonctions du système urinaire chez les personnes ayant subi des lésions à la colonne vertébrale. Chez ces personnes, l'implant est utilisé pour provoquer la miction de façon à remplacer l'utilisation de cathéters et de sacs collecteurs d'urine.

L'application consiste à provoquer l'écoulement urinaire par la stimulation du muscle de la vessie sans toutefois stimuler le muscle du sphincter. Toute stimulation simultanée (dyssynergie) de la vessie et du sphincter provoque une augmentation indésirable de la pression à l'intérieur de la vessie, ce qui peut mener à des problèmes sérieux notamment au niveau des reins.

Une des approches, qui fut employée dans le passé, consiste à stimuler électriquement les muscles directement. Cette méthode comporte toutefois des désavantages sérieux. Les électrodes sont soumises à des contraintes mécaniques sévères qui proviennent des muscles de la vessie ce qui provoque une détérioration prématurée. Les endroits qui sont directement en contact avec les électrodes deviennent insensibles à la stimulation avec le temps. De plus, pour provoquer une contraction uniforme du muscle de la vessie, il faut recourir à un grand nombre d'électrodes.

La meilleure approche consiste donc à stimuler directement au niveau du nerf. De cette façon, on utilise le réseau de communication (nerfs périphériques) qui est déjà en place. De plus, l'énergie requise pour provoquer les contractions d'un muscle est moindre si on stimule au niveau du nerf plutôt qu'au niveau du muscle directement. Par contre, cette approche comporte un désavantage majeur du fait que le seul nerf avec lequel on peut contrôler la vessie est aussi relié au muscle du sphincter. Par le fait même, toute stimulation électrique traditionnelle est à éviter. Nous proposons une technique de stimulation sélective qui permet de contrer ce problème. Par une combinaison de fréquences, nous arrivons à inhiber la contraction du sphincter tout en conservant la contraction de la vessie.

Le système de stimulation présenté dans ce mémoire comporte deux parties principales soit la partie externe, qui est composée du contrôleur et de son transmetteur, et le stimulateur implantable (implant) qui constitue la partie interne du système. Le contrôleur externe transmet, via l'émetteur, les paramètres de stimulation ainsi que l'énergie qui sert à alimenter l'implant.

Par la suite, la voie de la miniaturisation a été empruntée lors du développement et la réalisation d'un circuit d'implant entièrement intégré sur une seule puce. Le circuit, d'une complexité d'environ 25000 portes logiques, a été conçu et fabriqué dans un temps très court (quatre mois environs). Cette réalisation servit de point de départ pour d'autres travaux similaires dans l'équipe de recherche PolySTIM. Malheureusement, les résultats

des tests fonctionnels n'ont pas été très bons mais ont quand même permis d'avancer nos travaux de recherche sur le projet.

Les essais *in vivo* chez 11 chiens nous ont permis de valider la technique de stimulation. Les paramètres donnant les meilleurs résultats ont été établis. Les paramètres varient d'un animal à l'autre ce qui rend nécessaire l'utilisation d'un système permettant leur modification à volonté.

Les expérimentations de longue durée (phase chronique) ont permis de valider le système entier de stimulation. Quelques défaillances ont été notées mais somme toute, le système se comporte admirablement.

Les travaux faisant partie de cette maîtrise se terminent à la validation du système de stimulation. Par contre, ces essais en phase chronique se poursuivent toujours en collaboration avec le département d'Urologie de l'Hôpital Royal-Victoria et de l'Université McGill. Ils feront l'objet d'autres travaux de recherche similaires poursuivis par l'équipe de neurotechnologie PolySTIM.

## ABSTRACT

Implantable electrical stimulators are used in the treatment of many diseases. Many medical applications employ electronic implants to regain the functionality or the urinary system in spinal cord injured persons. In these people, the implant is used to provoque micturition in order to replace the use of catheters and urine collection bags.

The application consists in inducing micturition by stimulation of the bladder muscle without stimulating the sphincter muscle. Any simultaneous stimulation (dyssynergia) of the bladder and the sphincter will cause an unwanted rise of vesical pressure, which can lead to serious problems mostly in the kidneys.

One approach, used in the past, consists in stimulating the muscles directly with electrical pulses. This method contains serious disadvantages. The electrodes are submitted to severe mechanical constraints coming from the bladder muscle, which causes premature deterioration. The region that are directly in contact also become less sensitive to electrical stimulation with time. Also, a large number of electrodes are required in order to produce a uniform contraction of the bladder.

The best approach then consists in the direct stimulation of the nerve, using the already available communication network (peripheral nerves) that is already in place. Also, the energy required to induce contraction of a muscle is less if the nerve is stimulated rather than directly at the muscle. However, this method has a major disadvantage in the fact that it is the same nerve the controls both the bladder and the

sphincter. Because of this, any traditional stimulation of this nerve needs to be avoided. We propose a technique of selective electrical stimulation, which will overtake this problem. With a combination of two independent frequencies, we are able to inhibit contraction of the sphincter while creating contraction of the bladder.

The stimulation system presented in the document is composed of two major parts, which are the external part (the controller with transmitter) and the internal part (the implant). The external controller transmits, through the emitter, the parameters required for the electrical stimulation as well as the energy used by the implant.

After, the miniaturisation phase is followed in the development and the realisation of a circuit entirely integrated on a single silicon chip. The circuit, of an equivalent complexity of 25000 gates, has been fabricated in a record time of about four months. This realisation is now used as a starting point in similar ongoing projects in the PolySTIM research team. However, the results obtained with the functional tests have not been as good as expected but still allowed us to advance further in the research project.

In-vivo tests in 11 dogs permitted us to validate the selective stimulation technique. The parameters giving the best results have been established. They vary from one animal to another, which makes it necessary to have a system that permits great flexibility.

Long term experimentation (chronic phase) allowed us to validate the entire implantable stimulation system. Some minor problems have been observed but overall, the system worked well within our expectations.

All the work done in this project ends with the validation of the stimulation system. Chronic experimentation is still underway in collaboration with the Urology Department of the Royal-Victoria Hospital and the McGill University. These will be part of other similar research projects done by members of the PolySTIM team.

# TABLE DES MATIÈRES

Dédicace ..... iv

Remerciements ..... v

Résumé ..... vi

Abstract ..... ix

Table des matières ..... xii

Liste des tableaux ..... xv

Liste des figures ..... xvi

Liste des annexes ..... xviii

Liste des sigles et abréviations ..... xix

Introduction ..... 1

Chapitre 1 : Le système urinaire et la stimulation électrique ..... 5

1.1 Introduction ..... 5

1.2 Le système urinaire ..... 5

1.3 Les voies neurologiques ..... 6

1.4 La miction normale ..... 6

1.5 Les complications vésicales ..... 7

1.5.1 Mauvaise continence ..... 7

1.5.2 Mauvaise miction (vidange) ..... 7

1.5.3 Autres complications ..... 8

1.6 Le traitement des complications de la miction ..... 8

1.7 La stimulation électrique pour provoquer la miction ..... 9

1.8 La stimulation sélective ..... 11

1.9 Les limites des systèmes existants ..... 13

1.10 Conclusion ..... 16

Chapitre 2 : Proposition d'un système dédié à la stimulation sélective ..... 17
2.1 Introduction ..... 17
2.2 Les considérations de design ..... 18
2.3 Le système proposé ..... 20
2.3.1 Stimulateur pour expérimentation en phase aiguë ..... 20
2.3.2 Système de stimulation implantable ..... 21
2.4 Design des contrôleurs externes ..... 23
2.4.1 Contrôleur portatif ..... 24
2.4.2 Contrôleur avec interface pour ordinateur ..... 28
2.4.3 Transmetteur RF à couplage électromagnétique ..... 30
2.5 Design du stimulateur implantable (implant) sur circuit imprimé ..... 30
2.5.1 Architecture générale ..... 30
2.5.2 Circuit mixte de réception ..... 31
2.5.3 Circuiterie numérique de réception et de contrôle (FPGA) ..... 32
2.5.4 Circuiterie analogique de stimulation ..... 35
2.5.5 Assemblage de l'implant ..... 36
2.6 Conclusion ..... 38

Chapitre 3 : Design d'une puce dédiée à la stimulation électrique sélective ..... 39
3.1 Introduction ..... 39
3.2 Aperçu de l'implant entièrement intégré sur un dé de silicium ..... 39
3.3 Design du récepteur transcutané ..... 41
3.3.1 L'antenne ..... 42
3.3.2 Le redresseur, filtre et détecteur d'enveloppe ..... 42
3.3.3 Le décodeur Manchester ..... 43
3.3.4 Le régulateur de tension intégré ..... 43
3.4 Circuit numérique de réception, de contrôle et de génération des stimuli ..... 45
3.4.1 Méthodologie de design ..... 47
3.4.2 Module de réception des données avec détection d'erreur ..... 47
3.4.3 Module de contrôle des stimuli ..... 53

3.4.4 Étage de sortie : module de génération des stimuli ..... 56
3.5 Simulations fonctionnelles ..... 59
3.6 Réalisation du circuit intégré ..... 61
3.6.1 Synthèse et testabilité des modules numériques ..... 61
3.6.2 Dessin des masques ..... 62
3.7 Conclusion ..... 65

Chapitre 4 : Tests fonctionnels et résultats expérimentaux ..... 67
4.1 Introduction ..... 67
4.2 Résultats des simulations et tests fonctionnels ..... 67
4.2.1 Contrôleurs ..... 67
4.2.2 Implant réalisé sur circuit imprimé ..... 70
4.2.3 Implant réalisé sur circuit intégré ..... 72
4.3 Expérimentations ..... 74
4.3.1 Préparation des animaux pour expérimentation de courte durée ..... 74
4.3.2 Principaux résultats ..... 77
4.4 Conclusion ..... 81

Conclusions et recommandations ..... 82

Bibliographie ..... 85

# LISTE DES TABLEAUX

Tableau 3-1 : Description des paramètres de stimulation. ..... 42

Tableau 3-2 : Liste des plots. ..... 66

Tableau 4-1 : Efficacité de la stimulation sélective avec les paramètres optimaux..... 81

# LISTE DES FIGURES

Figure 1-1 : Schéma simplifié de la stimulation électrique urinaire. ..... 9

Figure 1.2 : Forme d'onde utilisé pour la stimulation sélective. ..... 12

Figure 1-3 : Photographie de l'implant urinaire réalisé par Sawan et al. ..... 15

Figure 2-1 : Schéma bloc du contrôleur pour expériences aiguës. ..... 20

Figure 2-2 : Schéma global du système de stimulation. ..... 22

Figure 2-3 : Schéma simplifié des deux contrôleurs externes. ..... 23

Figure 2-4 : Schéma-bloc du contrôleur portatif. ..... 25

Figure 2-5 : Étapes de développement d'un FPGA. ..... 27

Figure 2-6 : Schéma-bloc de l'interface de stimulation contrôlée par ordinateur. ..... 29

Figure 2-7 : Diagramme-bloc du stimulateur implantable. ..... 31

Figure 2-8 : Schéma-bloc de la circuiterie mixte de réception. ..... 32

Figure 2-9 : Schéma-bloc du FPGA de l'implant. ..... 32

Figure 2-10 : Représentation du contenu du registre à décalage tel que retrouvé dans un fichier de programmation. ..... 33

Figure 2-11 : Photographie du stimulateur implantable. ..... 38

Figure 3-1 : Forme d'onde générée par l'implant. ..... 40

Figure 3-2 : Délai entre les impulsions. ..... 40

Figure 3-3 : Circuit d'entrée : (a) redresseur et filtre, (b) détecteur d'enveloppe. ..... 42

Figure 3-4 : Circuit du régulateur de tension intégré. ..... 45

Figure 3-5 : Schéma bloc du stimulateur intégré. ..... 45

Figure 3-6 : Schéma général du décodeur réalisé. ..... 50

Figure 3-7 : Schéma-bloc de l'unité de contrôle. ..... 54

Figure 3-8 : Architecture de la source de courant proposée. ..... 57

Figure 3-9 : Protographie de la source de courant monolithique. ..... 63

Figure 3-10 : Dessins des masques de l'implant sur puce. ..... 64

Figure 4-1 : Premier prototype du contrôleur portatif. ..... 68

Figure 4-2 : Premier contrôleur portatif (vue extérieure du boîtier) ..... 69

Figure 4-3 : Photographie du premier protoype d'implant. ..... 70

Figure 4-4 : Photographie du circuit intégré dans son boîtier. ..... 73

Figure 4-5 : Résultats de stimulation avec et sans blocage par haute fréquence. ..... 79

## LISTE DES ANNEXES

Annexe I : Schémas et code des contrôleurs externes ..... 101

Annexe II : Schémas de l'implant sur circuit imprimé ..... 108

Annexe III : Desssin de masques, Code et résultats de synthèse de l'implant intégré..... 118

Annexe IV : Fiche technique des électrodes bipolaires..... 150

Annexe V : Articles publiés ou soumis pour publication ..... 153

## LISTE DES SIGLES ET ABRÉVIATIONS

| | |
|---|---|
| AM | *Amplitude Modulation* (Modulation d'amplitude) |
| ATPG | *Automatic Test Patters Generation* (Génération automatique de vecteurs de test) |
| CLK | *Clock* (Horloge) |
| CMC | *Canadian Microelectronics Corporation* (Société Canadienne de Micro-Électronique) |
| CMOS | *Complementary Metal-Oxyde Semiconductor* |
| CNA | Convertisseur Numérique à Analogique |
| CRC | *Cyclic Redundancy Check* |
| DAC | *Digital to Analog Converter* (Convertisseur numérique à analogique) |
| DC | *Direct Current* (Courant continu) |
| DDI | *Direct Data Input* (Mode d'entrée directe des données) |
| DFT | *Design for test* (Conception en vue de la testabilité) |
| DRC | *Design Rule Check* (Vérification des règles de design) |
| EDIF | *Electronic Data Interchange Format* |
| FIFO | *First In First Out* |
| GND | *Ground* (Mise à la masse ou à la terre) |
| LSB | *Least Significant Bit* (Bit le moins significatif) |
| POR | *Power-On Reset* |
| RF | *Radio Frequency* (Fréquence radio) |
| ROM | *Read Only Memory* |
| Vce | Tension entre le collecteur et l'émetteur d'un transistor bipolaire |
| Vds | Tension entre ke drain et la source d'un transistor CMOS |
| VHDL | *Very high speed integrated circuit Hardware Description Language* |

# INTRODUCTION

Depuis déjà plus de trois décades, plusieurs types de stimulateurs implantables ont été introduits pour le traitement des dysfonctions neurologiques ainsi que l'aide ou le remplacement d'organes éprouvant des difficultés. Les stimulateurs cardiaques ("pace-makers") sont sans doute les plus connus mais les implants sont aussi utilisés entre autres pour rétablir l'audition (implants cochléaires), pour enrayer la douleur, régulariser la respiration et aussi pour contrôler la miction [26, 35, 44, 49, 75].

Plus récemment, plusieurs tentatives ont été faites pour récupérer la fonctionnalité du système urinaire chez les personnes ayant subit des lésions au niveau de la colonne vertébrale [13-20, 28, 29, 61-73, 80-83]. La plupart de ces personnes qui souffrent de rétention urinaire doivent alors utiliser des cathéters et sacs pour procéder à l'évacuation de l'urine ce qui implique plusieurs désagréments en plus des risques d'infections qui deviennent très élevés.

Malgré que la stimulation électrique appliquée directement à la vessie pour provoquer la miction soit connue depuis très longtemps, cette technique n'a jamais donné les résultats escomptés. Les techniques plus modernes qui utilisent des systèmes électroniques implantables pour provoquer la miction par stimulation électrique au niveau des nerfs connaissent une popularité montante depuis quelques dizaines d'années mais toutes les techniques proposées jusqu'à maintenant connaissent des lacunes.

Les techniques existantes, comme la miction post-stimuli, souffrent du fait qu'elles provoquent une contraction simultanée du sphincter et de la vessie (dyssynergie). Ceci est dû au fait que c'est le même nerf qui relie ces deux muscles à la colonne vertébrale. Une solution utilisée pour remédier à ce problème consiste à procéder une neuroctomie du nerf honteux [82, 81, 83]. Cette dyssynergie provoque une haute pression vésicale qui peut à son tour causer des problèmes comme l'incontinence et des problèmes rénaux tandis que la neuroctomie est une intervention irréversible qui peut parfois causer des réductions au niveau de l'érection reflexogénique chez les mâles paraplégiques [41, 64].

La stimulation sélective [53-56, 61-62] permet de minimiser l'effet de la dyssynergie sans recourir à la neuroctomie. L'objectif de la stimulation sélective est d'atteindre le seuil d'activation des fibres somatiques avec la stimulation à haute fréquence créant ainsi le blocage en inhibant ces fibres tout en restant en dessous du seuil d'activation des fibres autonomiques qui relient la vessie. De cette façon, seules les fibres autonomiques resteront sensibles à une stimulation à plus basse fréquence créant ainsi la miction à basse pression par une contraction maximale de la vessie et une contraction minimale du sphincter.

Une fois toute la théorie explorée, il faut alors passer à la prochaine étape qui, dans une telle recherche, consiste à procéder à un certain nombre de tests expérimentaux de façon à effectuer une première validation de la technique de stimulation sélective préalablement expliquée. Normalement, cette première série de tests est effectuée in-vivo sur des animaux de laboratoire. Dans ces tests, appelés tests en phase aiguë, la technique

de stimulation est validée à la suite d'un bon nombre de stimulations sur une dizaine d'animaux. On applique des séries de stimulations électriques en modifiant les paramètres sur une gamme de valeurs de façon à déterminer les meilleurs agencements de paramètres, s'ils existent, pour atteindre l'objectif qui est la miction à basse pression. Tous les résultats de ces expériences sont sauvegardés sous différentes formes et servent à définir l'ensemble des spécifications du système implantable. On pourra alors passer à l'étape suivante qui consiste aux tests expérimentaux de longue durée appelés essais en phase chronique.

Les tests en phase chronique ont pour but de valider la technique de stimulation dans des conditions qui se rapprochent beaucoup plus de la réalité. Lors de ces essais, un stimulateur est alors implanté dans plusieurs animaux pour une durée variant de trois à six mois. Durant cette période, le stimulateur est utilisé pour provoquer la miction. Dans le cadre de ma recherche, les tests en phase chronique ont servi principalement à tester toutes le composantes du système de stimulation électrique proposé.

Les tests en phase aigüe et chronique nécessitent chacun un équipement électronique dédié à l'application. Dans le cas des essais en phase aiguë, de courte durée, un stimulateur externe d'expérimentation médicale doit être utilisé tandis que, pour les tests chroniques, un système composé d'un stimulateur implantable est requis.

Les systèmes implantables de stimulation électrique actuels n'ont pas les caractéristiques qui rendent possible la stimulation sélective. Ces caractéristiques sont

notamment la possibilité de combiner les courants de deux ondes totalement indépendantes en fréquence et en amplitude sur la même électrode. De plus, la gamme de fréquence doit s'étendre pour la haute fréquence à près de 1kHz. Le système qui était utilisé jusqu'à tout récemment est celui proposé par Sawan et ses collaborateurs [64-69]. Ce système, qui a fait ses preuves dans le passé, ne permet pas d'effectuer adéquatement la stimulation sélective. Entre autres, sa gamme de fréquence est trop limitée et la linéarité de ses sources de courant n'est pas satisfaisante.

Le besoin d'un nouveau système pour remplacer le système actuel est alors devenu une priorité. Le nouveau système doit être en mesure de couvrir toutes les spécifications requises pour la stimulation sélective tout en étant réalisable dans des délais raisonnables et bien sûr être fiable et facile à utiliser. Le remplacement proposé se ferait en deux étapes : des prototypes réalisés avec des circuits discrets et programmables en premier lieu et un circuit intégré dédié en second.

Dans le chapitre 1, le système urinaire ainsi que la stimulation électrique sélective sont exposés brièvement. Le chapitre 2 est destiné au système complet de stimulation électrique sélective qui a été réalisé dans le cadre de ces travaux. Le chapitre 3 relate la conception et la réalisation d'un circuit intégré (une puce) dédié à la stimulation sélective. Les résultats obtenus dans le cadre de ces travaux sont exposés au chapitre 4, suivi finalement des conclusions générales.

# CHAPITRE 1

# LE SYSTÈME URINAIRE ET LA STIMULATION ÉLECTRIQUE

## 1.1 Introduction

Dans ce premier chapitre, vous trouverez une description des différents aspects anatomiques et biologiques qui sont reliés au sujet de cette maîtrise. Le système urinaire, les voies neurologiques, la miction normale ainsi que les problèmes reliés à la miction sont parcourus pour donner une vision d'ensemble du problème. Différents traitements incluant la stimulation électrique sont explorés, une revue de littérature sur les systèmes existants est faite et, pour terminer le chapitre, la stimulation sélective est expliquée. Ce chapitre ne constitue qu'un aperçu de l'aspect médical de ces travaux, c'est pourquoi nous nous limiterons à l'essentiel.

## 1.2 Le système urinaire

Le système urinaire extrait les impuretés du sang et les élimine avec l'excès de liquide de l'organisme, sous forme d'urine. L'urine se forme à partir des reins. L'urine débouche des reins et s'écoule dans la vessie par l'intermédiaire de deux tuyaux appelés uretères. C'est par l'urètre que la vessie expulse l'urine vers l'extérieur. La vessie (le détrusor) est un réservoir composé de fibres musculaires qui se distend lentement au fur et à mesure que s'accumule l'urine, (similairement à la dilatation d'un ballon que l'on gonfle). À la sortie de la vessie, à la jonction de l'urètre, il existe un muscle appelé sphincter qui reste contracté et fermé entre les mictions, assurant la continence. La vessie et le sphincter sont

normalement sous contrôle volontaire, ce qui signifie que le sujet peut contrôler sa miction.

### 1.3 Les voies neurologiques

Le contrôle volontaire de la miction est effectué par le cerveau. Bien que l'on ne comprenne pas clairement le processus, on sait malgré tout qu'une zone du cerveau permet une réalisation harmonieuse de la miction en permettant la contraction vésicale et la relaxation sphinctérienne. Les influx qui cheminent du cerveau à la vessie sont véhiculés par des voies nerveuses qui descendent dans la moelle et sortent à sa partie terminale basse (racines sacrées, S1 à S5). Ces influx d'origine cérébrale provoquent la contraction volontaire de la vessie et l'ouverture du sphincter, permettant à la miction de s'effectuer. Le cerveau entretient par ailleurs une inhibition permanente de contractions réflexes et involontaires de la vessie qui peuvent se produire pendant son remplissage.

L'interruption de la communication au niveau des fibres nerveuses situées dans la partie terminale de la moelle peut entraîner des troubles de la miction, parce que la vessie privée de son innervation ne peut pas se contracter efficacement, lors de la miction volontaire.

### 1.4 La miction normale

Pour que la miction ait lieu, deux phénomènes doivent se dérouler simultanément : le muscle vésical doit se contracter pour expulser l'urine pendant que le sphincter se relâche

et s'ouvre, permettant le libre écoulement de l'urine vers l'extérieur. En somme, le déclenchement de la miction normale est sous contrôle volontaire. Quand 300 à 500 ml se sont accumulés dans la vessie, le sujet éprouve un besoin d'uriner et il décide du moment où il va vider sa vessie. La miction normale doit être complète, sans laisser de résidu post-mictionnel.

## 1.5 Les complications vésicales

### 1.5.1 Mauvaise continence

Le trouble de la continence apparaît quand la vessie est incapable de stocker l'urine qu'elle reçoit. Dans ce cas, le détrusor est hyperactif. Les contractions vésicales sont déclenchées involontairement alors que la vessie ne contient qu'une faible quantité d'urine. Le sphincter s'ouvre correctement, ce qui entraîne des mictions fréquentes, et dans les cas graves, aboutit à des pertes d'urines incontrôlées.

### 1.5.2 Mauvaise miction (vidange)

Le sphincter se resserre au lieu de se relâcher au moment où le détrusor se contracte pour expulser l'urine: c'est la dyssynergie vésico-sphinctérienne. Habituellement, une certaine quantité d'urine est évacuée, mais une quantité importante peut rester dans la vessie.

### 1.5.3 Autres complications

Les troubles de vidange vésicale provoquent une rétention d'urine chronique. Le résidu peut contribuer à provoquer une infection urinaire puisque les bactéries se multiplient librement dans l'urine qui stagne.

Des infections urinaires fréquentes dans la vessie (cystites) peuvent elles-mêmes entraîner des lésions rénales car l'urine infectée refoule de la vessie dans les reins.

## 1.6 Le traitement des complications de la miction

- Une des méthodes les plus simples pour obtenir une vidange vésicale complète est le sondage intermittent. Il permet un drainage périodique de l'urine plusieurs fois par jour et se réalise par l'introduction d'une petite sonde (cathéter) dans la vessie via l'urètre. Cette méthode requiert des manipulations fréquentes à chaque jour, cause des irritations ainsi que des infections urinaires occasionnelles.
- Le sondage permanent. Cette méthode est à éviter au maximum car la pose d'une sonde permanente accroît les risques de complications tels que l'infection urinaire.
- Le traitement médicamenteux visant à relâcher les sphincters ou à ouvrir le col de la vessie. Cette approche reste encore au stade de la recherche pour les troubles permanents.
- Le recours à la stimulation électrique pour provoquer la miction (l'objet du présent mémoire).

## 1.7 La stimulation électrique pour provoquer la miction

Les premiers stimulateurs électriques utilisés pour provoquer la miction avaient recours à la stimulation électrique au niveau des muscles directement [26, 75, 89]. Cette technique nécessite beaucoup d'énergie, requiert l'implantation d'un grand nombre d'électrodes pour provoquer une contraction uniforme de la vessie. De plus, les électrodes ainsi implantées sont soumises à des contraintes mécaniques répétées qui causent souvent leur rupture. À cause de tous ces problèmes, la stimulation au niveau du muscle fut abandonnée pour la stimulation au niveau du nerf.

Pour la stimulation électrique au niveau du nerf, une électrode de type à anneaux par exemple, est enroulée autour du nerf et branchée à un stimulateur. Le nerf qui est responsable de la miction est le nerf sacré partant de la vertèbre S2. Ce nerf est utilisé pour provoquer la contraction de la vessie.

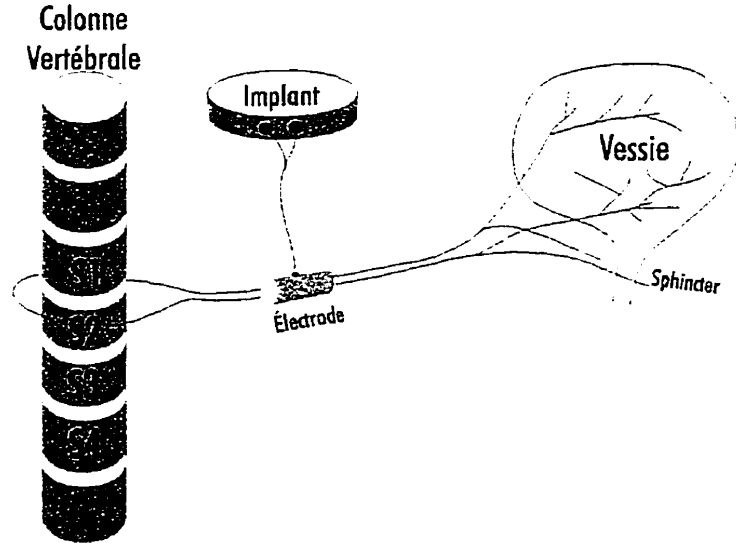



**Figure 1-1 : Schéma simplifié de la stimulation électrique urinaire.**

À première vue, il peut sembler plutôt simple de stimuler électriquement le nerf S2 pour provoquer la contraction de la vessie. Mais ce n'est pas si simple car le nerf S2 qui relie la vessie relie aussi le sphincter. Chaque stimulation électrique au niveau du nerf S2 provoque donc une contraction simultanée de la vessie et du sphincter (dyssynergie), ce qui résulte en une augmentation marquée de la pression vésicale sans toutefois provoquer une bonne miction. Cette haute pression est indésirable car elle peut causer des problèmes sérieux aux reins. Pour pallier le problème de la dyssynergie, quelques options se présentent. On peut procéder à la neuroctomie (couper) du nerf honteux reliant le sphincter ou la ryzoctomie (racines réflexes de S2). Cette technique est par contre évidemment irréversible en plus de causer la perte d'érection réflexogénique chez les mâles paraplégiques [41, 64].

Une autre technique, la miction post-stimulus [64,68], fut mise au point par Sawan et al. Elle consiste à stimuler électriquement le nerf S2 pour une certaine durée, causant ainsi une augmentation de la pression vésicale, et d'interrompre la stimulation à intervalle régulier. Puisque le muscle de la vessie réagit plus lentement que celui du sphincter, il prend donc plus de temps à se relâcher provoquant une miction au moment de l'interruption de la stimulation. Cette technique a connu un certain succès sans toutefois réaliser les résultats escomptés.

## 1.8 La stimulation sélective

Plus récemment, la stimulation sélective adaptée pour la miction a été mise au point. Cette technique a pour objectif la miction continue à basse pression sans recours à la neuroctomie. Cette technique peut, par une combinaison de deux ondes superposées sur le même nerf, réussir à stimuler le muscle de la vessie sans toutefois créer une contraction du sphincter. Autrement dit, on sélectionne les fibres qui nous intéressent (d'où son nom).

À cause de la différence physiologique entre les fibres qui relient le sphincter et les fibres qui stimulent la vessie, la stimulation sélective d'une fibre en particulier est une alternative qui est très convoitée.

Plusieurs méthodes ont été examinées, incluant le blocage anodique, le blocage par collision et le blocage par haute fréquence des fibres somatiques [53-56]. Le blocage par haute fréquence a l'avantage d'être plus pratique puisque les deux stimuli, la haute et la basse fréquence, sont appliquées à l'aide d'une électrode bipolaire unique. L'objectif consiste à atteindre le niveau d'activation des fibres somatiques (A-alpha) à l'aide des impulsions à haute fréquence, tout en restant en dessous du niveau d'activation des fibres autonomiques (A-delta) qui innervent le detrusor. Toute transmission par les fibres somatiques est alors inhibée tandis qu'une stimulation basse fréquence normale permet l'excitation du détrusor provoquant ainsi la contraction de la vessie et la miction.

L'onde qui est donc nécessaire pour réussir doit contenir deux trains d'impulsions combinés sur le même canal, en mode courant (Figure 1-2).

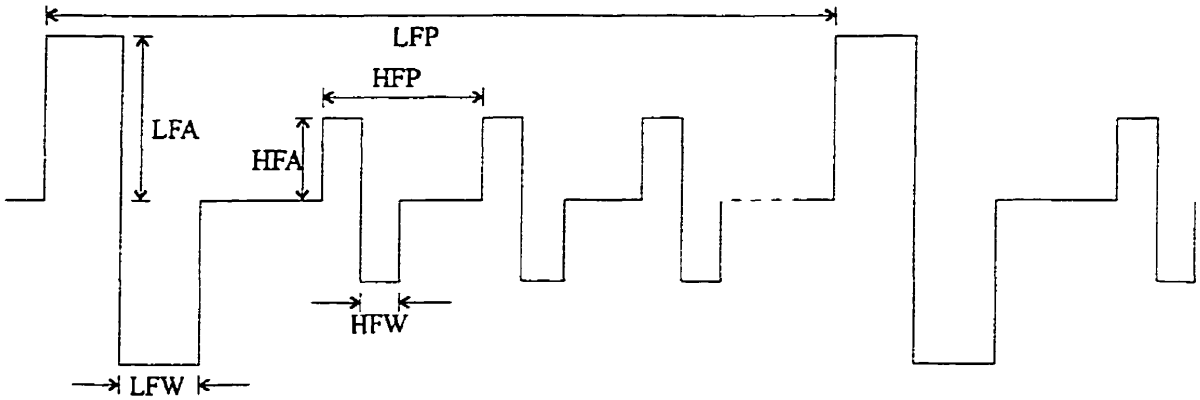



**Figure 1-2 : Forme d'onde utilisé pour la stimulation sélective.**

Le tableau suivant contient la description des différents paramètres :

| **Tableau 1-1 : Paramètres avec description** | |
|---|---|
| Abréviation | Nom |
| HFA | Amplitude des impulsions de haute fréquence |
| HFP | Période de la haute fréquence |
| HFW | Largeur des impulsions de la haute fréquence |
| HFI | Délai entre l'impulsion haute et basse de la haute fréquence |
| LFA | Amplitude des impulsions de la basse fréquence |
| LFP | Période de la basse fréquence |
| LFW | Largeur des impulsions de la basse fréquence |
| LFI | Délai entre l'impulsion haute et basse de la basse fréquence |

Pour qu'une telle technique fonctionne, le système de stimulation doit donc être en mesure de générer une onde superposant (en courant) deux signaux de formes, de fréquences et d'amplitudes indépendantes et avec une grande plage de valeurs.

## 1.9 Les limites des systèmes existants

Malgré de nombreuses découvertes dans le domaine de la microélectronique biomédicale, les progrès récents n'ont pas suffi à combler le manque dans le domaine de la restauration des fonctions vésicales. Techniquement parlant, la majorité des stimulateur disponibles commercialement ne possèdent pas les caractéristiques requises pour l'application comme :

- Une grande variété de paramètres programmables à volonté
- Efficacité dans la transmission des données et de l'énergie
- Interfaces usager simples et conviviales
- Possibilité de générer des impulsions bipolaires à haute fréquence
- Possibilité de générer deux fréquences indépendantes sur un même canal

La plupart des stimulateurs disponibles sont basés sur un démodulateur AM qui extrait les données du signal RF et qui les transmet directement aux tissus via une ou plusieurs électrodes. Les principaux modèles disponibles sont :

- Le stimulateur Avery est en mesure de générer des impulsions monopolaires en tension et à une fréquence allant jusqu'à 200 Hz [8].
- Le stimulateur Gorgis, qui est adapté d'un stimulateur cardiaque. La caractéristique la plus intéressante de cet appareil réside dans le contrôle pour l'allumer ou le fermer qui est effectué à partir d'un aimant [15].

- Le stimulateur Medtronic qui est programmable par l'entremise de deux appareils et contient un circuit de télémétrie pour confirmer l'entrée des données [46].
- L'appareil Finetech qui est composé de trois circuits de réception par couplage inductif identiques [51].
- Le stimulateur Physico-Med, est muni de 8 sorties contenant toute la même information [43].
- Le stimulateur réalisé par Sawan et al. Ce système multi-canaux est très versatile et fonctionne par l'entremise d'instructions codées sur 24 bits transmises en temps-réel [64-73].

Le système utilisé jusqu'à tout récemment par notre équipe de recherche, le plus flexible parmi la majorité des systèmes disponibles, était celui réalisé par Sawan et collaborateurs de l'Université de Sherbrooke et de l'université McGill (Figure 1-3). Ce système, qui a fait ses preuves dans le passé, ne permet pas d'effectuer adéquatement la stimulation sélective. Entre autres, sa gamme de fréquence est trop limitée et il ne permet pas une synchronisation suffisante. Par contre, plusieurs parties de ce système ainsi que l'expertise qui fut développée ont été utilisées dans le cadre de ce nouveau projet.

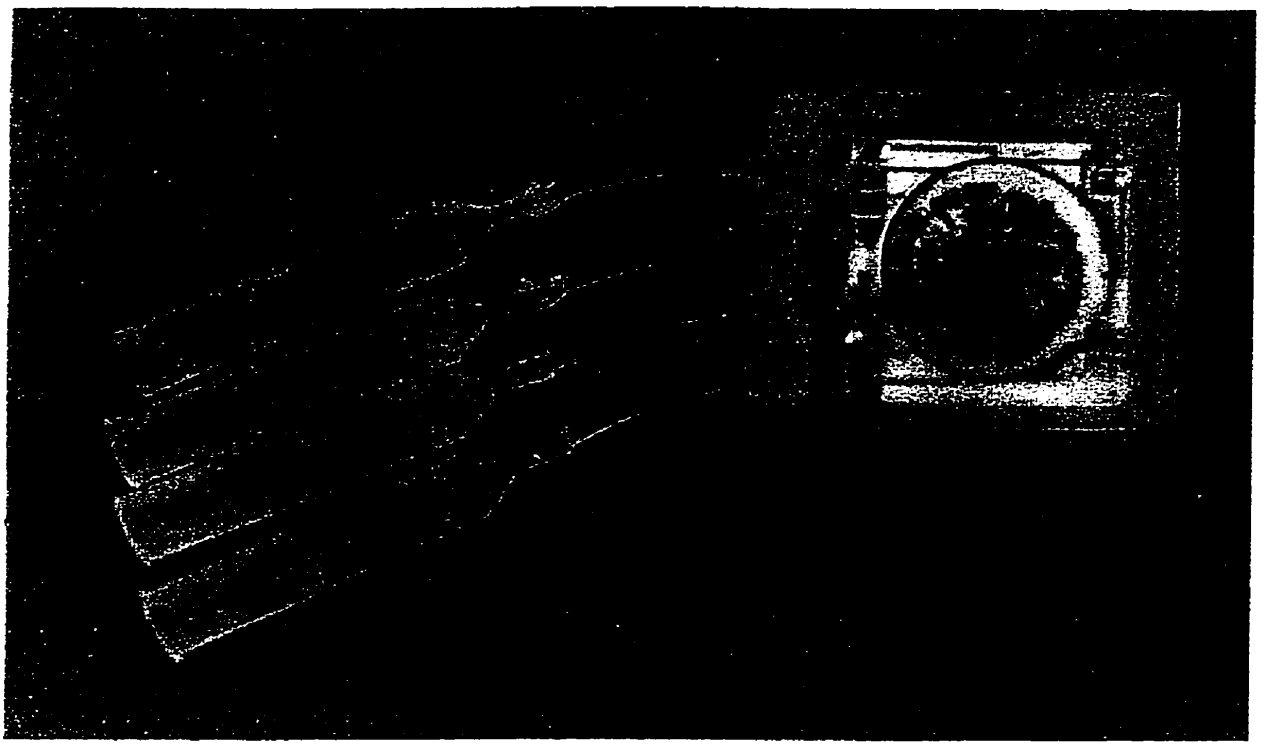


**Figure 1-3 : Photographie de l'implant urinaire réalisé par Sawan et al.**

Les principales caractéristiques du système réalisé par Sawan et al. sont :

- Le contrôleur externe : il s'agit d'un processeur miniaturisé qui est utilisé par l'usager (le médecin traitant) afin de programmer l'implant.
- L'implant : il comprend trois parties essentielles : a) une puce dédiée formant la partie centrale est fabriquée en prédiffusé CMOS 4 microns de Mitel et compte 8000 transistors environ. Elle est en mesure de commander les stimuli sur 8 canaux monopolaires ou 4 bipolaires. Il reçoit des commandes sur 24 bits en temps réel.

Deux autres stimulateurs ont été proposé dernièrement. Le premier fût développé aussi par Sawan. Il s'agit d'une version plus évoluée utilisant un circuit intégré dédié

utilisant la technologie CMOS 3 microns. Chaque stimulus nécessite la programmation de 9 paramètres ; b) une mémoire reprogrammable permettant de sauvegarder les paramètres pour des stimulations subséquentes; c) un circuit d'interface assurant la réception des commandes, de l'horloge et de l'énergie du contrôleur externe. La particularité de cet implant réside dans le fait que la génération d'une séquence complète de stimulation peut être effectuée par une simple commande provenant d'un petit transmetteur destiné au patient.

Ce circuit a une complexité de 12754 transistors. Le second, réalisé par Arabi [2], est basé sur des RAMs permettant de programmer les forme des stimuli. Cet implant permet aussi la modulation en fréquence. Ces deux derniers n'ont pas été complété car nous réalisons qu'un système plus simple et plus versatile pourra répondre aux besoins et surtout être réalisé. C'est ce que nous proposons dans ce mémoire.

## 1.10 Conclusion

Dans ce premier chapitre, le système urinaire ainsi que les problèmes qui s'y rattachent ont été explorés de façon à bien comprendre le problème qui fait l'objet de ce mémoire. Les différentes solutions disponibles actuellement ont été énumérées, dont la plus récente de Sawan et al. Un nouveau système de stimulation électrique qui s'attaque au problème par la stimulation sélective est présenté au chapitre suivant.

# CHAPITRE 2

# PROPOSITION D'UN SYSTÈME DÉDIÉ À LA STIMULATION SÉLECTIVE

## 2.1 Introduction

Après avoir validé la technique au cours d'essais *in vivo* de courte durée, le besoin pour un nouveau système implantable servant à la stimulation sélective à long terme est pressant. Un système complet, qui doit servir pour des expériences in-vivo à long terme, doit être réalisé. Un tel système est composé d'une partie interne soit le stimulateur électronique implantable (implant) et d'une partie externe qui est le contrôleur. Ce système doit être surtout versatile, posséder une bonne marge de manoeuvre étant donné l'incertitude quant aux fréquences, amplitudes et largeurs d'impulsions requises. De plus, ce système doit être réalisé dans un délai minimum pour pouvoir commencer les tests le plus rapidement possible.

L'implant doit être en mesure de réaliser la stimulation électrique au niveau du nerf de façon fiable, précise, sécuritaire et cela à partir de paramètres provenant de l'extérieur et qui peuvent être changés à volonté. Les informations sur la stimulation doivent être transmises par lien à couplage inductif à fréquence radio (RF). De plus, ce lien RF sert aussi à la transmission de l'énergie ce qui élimine le besoin d'une pile.

Deux types de contrôleurs sont alors requis. Le premier, actionné à partir d'un micro-ordinateur et d'un logiciel dédié est muni d'une interface RF, est utilisée par le personnel médical et de recherche pour déterminer les paramètres optimaux de

stimulation. Le second type de contrôleur est portatif, beaucoup plus simple et sert à réaliser la stimulation au jour le jour à partir des paramètres provenant de l'autre modèle de contrôleur.

Ce chapitre regroupe donc la description des tous les éléments qui ont été utilisés pour la conception ainsi que la réalisation de toutes les parties du système complet de stimulation électrique. Ce système est utilisé pour l'étude de la stimulation électrique sélective conjointement avec le département d'Urologie l'Université McGill de l'Hôpital Royal-Victoria de Montréal.

## 2.2 Les considérations de design

Il est important de considérer les caractéristiques qui sont requises pour le développement d'un tel système. Voici ces caractéristiques qui sont utilisées pour la conception et la réalisation d'un tel système. La plupart de ces considérations s'appliquent autant au modèle pour essais en phase aiguë qu'en phase chronique tandis que certaines de ces considérations s'appliquent plus à un ou l'autre des deux modèles.

Tout le matériel doit être facile d'utilisation car il est utilisé par plusieurs personnes qui ne sont pas nécessairement familières avec l'électronique ou la technologie en général. Ceci comprend principalement l'interface du contrôleur portatif ainsi que le logiciel qui est utilisé par le contrôleur avec interface paralèlle.

Il faut que le circuit exécute toutes les fonctions sans erreur. Un système de vérification des sorties pourra être utilisé pour s'assurer que l'implant reste toujours sécuritaire. La fiabilité touche aussi la communication entre le contrôleur et l'implant. Un bon protocole de communication comprenant la détection d'erreur de transmission est donc nécessaire.

La consommation de l'implant, incluant l'énergie requise par le circuit (énergie perdue) et l'énergie utilisée pour la stimulation au niveau du nerf (énergie utile), doit être inférieure à ce que la transmission transcutanée peut fournir. Cela s'applique aussi au contrôleur portatif qui doit fonctionner avec une pile pour une durée raisonnable.

L'implant destiné aux humains doit être de dimensions semblables ou inférieures à celles des « pacemakers » commerciaux disponibles actuellement. Par contre, pour les fins d'expérimentation dans les petits animaux, il faut que les dimensions de l'implant soient réduites en fonction de la dimension de l'animal. La solution optimale serait d'inclure tous les modules analogiques et numériques sur un même dé de silicium en éliminant tous les composants externes. Ceci demeure toutefois un grand défi, surtout pour l'intégration de l'étage d'entrée d'alimentation et de réception.

## 2.3 Le système proposé

### 2.3.1 Stimulateur pour expérimentation en phase aiguë

Ce stimulateur est en fait un appareil d'expérimentation non implantable de laboratoire à commandes analogiques qui fut conçu spécialement pour cette application. Cet appareil est utilisé pour les expériences en phase aiguë lorsque le nerf est accessible directement.

Ce stimulateur, représenté à la Figure 2-1, est composé de deux générateurs de forme d'onde indépendants. Les sorties de ces deux générateurs sont alors combinées en mode courant pour créer une seule sortie de courant afin de générer l'onde servant à la stimulation sélective. Cette sortie peut alors être branchée sur une électrode à anneaux ("cuff electrode") ou à aiguilles.

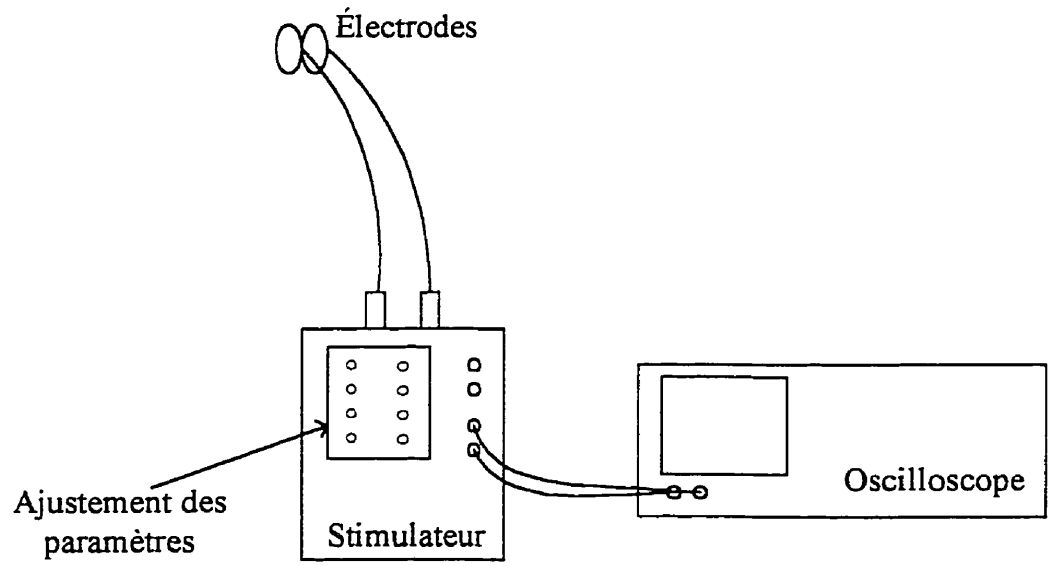



**Figure 2-1 : Schéma bloc du contrôleur pour expériences aiguës.**

Chaque générateur d'ondes possède des contrôles analogiques, réalisés à partir de potentiomètres de précision, permettant de contrôler l'amplitude, la fréquence ainsi que la largeur des impulsions hautes et basses. C'est à l'aide d'un oscilloscope que chaque paramètre est mesuré et ajusté. Chaque paramètre possède une résolution très fine puisqu'ils sont tous de type analogique. Le tableau ci-dessous contient les différents paramètres ainsi que leur plage d'opération.

**Tableau 2-1 : Paramètres du générateur d'ondes à contrôle analogique.**

| | Basse Fréquence | | | Haute Fréquence | | |
|---|---|---|---|---|---|---|
| Paramètre | Periode | Largeur imp. | Amplitude | Periode | Largeur imp. | Amplitude |
| Symbole | LFP | LFW | LFA | HFP | HFW | HFA |
| Unité | 1/Hz | Micro-sec. | mA | 1/Hz | Micro-sec. | mA |
| Plage | 1/100 - 1/10 | 0-300 | 0-2.5 | 1/1000-1/100 | 0-100 | 0-2.5 |

### 2.3.2 Système de stimulation implantable

Pour effectuer la stimulation nerveuse dans les essais chroniques, nous utilisons le système présenté à la Figure 2-2 qui comprend un contrôleur externe, un stimulateur implantable (implant) et une (ou plusieurs) électrode(s) à anneaux.

Lors d'une stimulation électrique, l'utilisateur commence par choisir les paramètres qui seront utilisés. Ensuite, la stimulation est activée, ce qui met en marche le transmetteur RF qui transmet alors les paramètres choisis à l'implant par couplage électromagnétique transcutané. Un fois que l'implant a capté les paramètres et qu'il a vérifié la validité de la transmission, il débute une séance de stimulation électrique. De plus, puisque l'implant n'est muni d'aucune source d'énergie interne, l'énergie requise pour le fonctionnement de l'implant provient aussi du couplage électro-magnétique. L'implant est muni de

connecteurs étanches spéciaux permettant de brancher l'électrode à anneaux qui est enroulé autour du nerf.

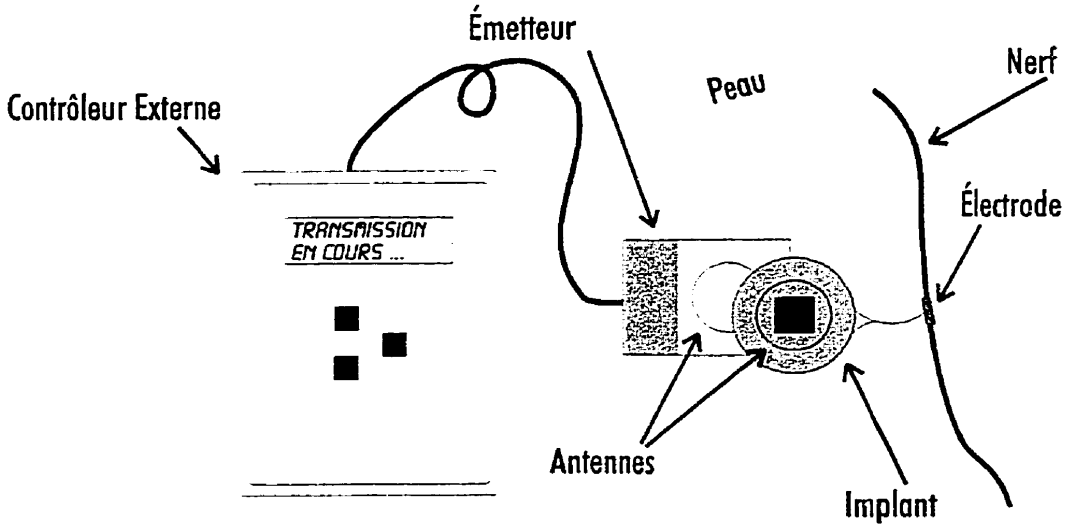



**Figure 2-2 : Schéma global du système de stimulation.**

Les paramètres qui sont transmis de façon numérique à l'implant sont les mêmes que ceux présentés au Tableau 2-1.

Voici les caractéristiques principales du système de stimulation :

- Fréquence de transmission, modulation en amplitude (AM) : 20 MHz. Il a été démontré que cette fréquence est efficace pour la transmission transcutanée [67]. De plus, l'utilisation de cette fréquence nous permet de rester compatible avec les systèmes existants dans notre équipe de recherche (PolySTIM).
- Fréquence de transmission numérique : 300 kbits/s. L'utilisation de cette fréquence nous permet aussi de rester compatible avec les systèmes existants

dans PolySTIM. Étant donné la réponse en fréquence d'un nerf, il n'est pas nécessaire d'utiliser une fréquence plus grande.

- Transmission de l'horloge par codage Manchester. C'est la façon la plus simple et la plus facile à implanter pour la reconstitution d'horloge.
- Distance maximale de transmission : 4 cm. C'est la distance maximale requise pour les types d'implants et des applications que nous addressons.

## 2.4 Design des contrôleurs externes

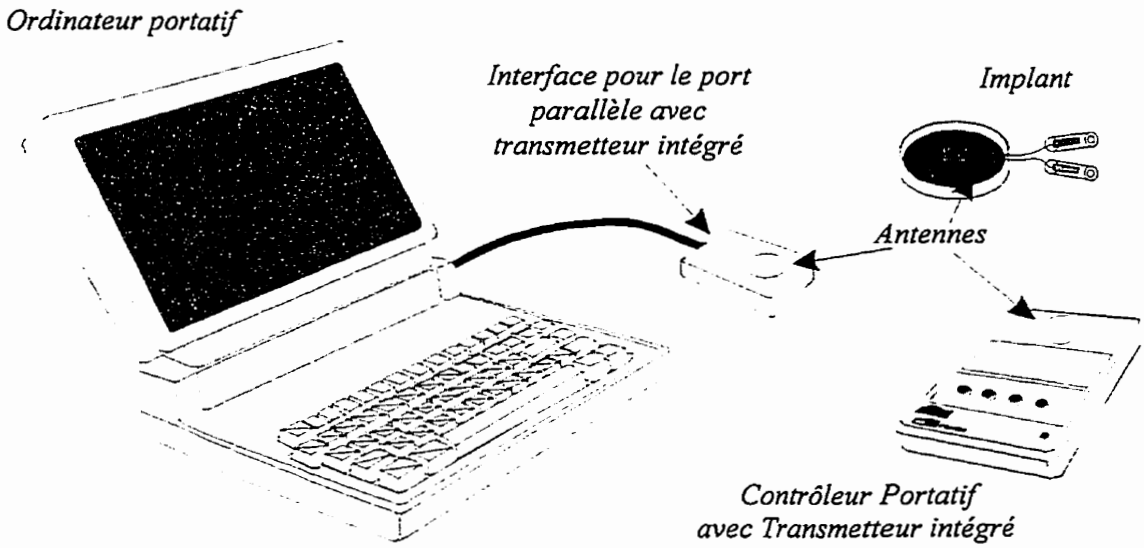



**Figure 2-3 : Schéma simplifié des deux contrôleurs externes.**

C'est le contrôleur qui constitue l'interface entre l'usager et l'implant. Cette partie du système est très importante et elle doit être fiable, efficace et simple d'utilisation. Pour nos travaux de recherche, deux types de contrôleurs ont été développés. Le premier est contrôlé à partir d'un ordinateur tandis que le second est un modèle portatif entièrement autonome. Le premier sert à déterminer précisément les paramètres qui seront utilisés

pour chaque animal tout au long des expériences en phase chronique. Une fois que les paramètres donnant les meilleurs résultats sont identifiés à l'aide du contrôleur et de l'ordinateur, ils peuvent être transférés dans la mémoire du modèle portatif pour usage quotidien. Les différents modèles de contrôleurs externes sont montrés à la Figure 2-3.

### 2.4.1 Contrôleur portatif

Le contrôleur portatif doit être très simple d'utilisation, efficace et fiable puisqu'il est utilisé quotidiennement par le personnel médical. Pour effectuer une stimulation pour un animal donné, l'utilisateur sélectionne les paramètres à l'aide des informations contenues sur l'écran à cristaux liquides et active la stimulation en appuyant sur le bouton de stimulation. Tous les paramètres dédiés à une stratégie de stimulation ainsi que l'information contenue à l'écran sont programmés dans une mémoire non volatile de type EPROM.

La Figure 2-4 représente le schéma-bloc du contrôleur portatif. Les principales parties du contrôleur sont :

- L'alimentation qui fournie 5 volts à la circuiterie numérique de contrôle ainsi que 12 volts pour le modulateur AM. L'alimentation peut provenir soit d'une pile 9 volts ou encore d'un adaptateur à courant continu branché dans l'alimentation AC 120 volts.

- L'interface usager qui est constituée de boutons et d'un écran à cristaux liquides (LCD) de deux lignes de 16 caractères chacunes.

- La mémoire de type EPROM qui contient tous les ensembles de paramètres nécessaires pour chacune des stimulations ainsi que l'information affichée à l'écran.

- Toute la logique de contrôle est entièrement intégré dans un circuit programmable FPGA (« Field Programmable Gate Array ») de type non-volatile.

- Le modulateur AM et l'antenne.

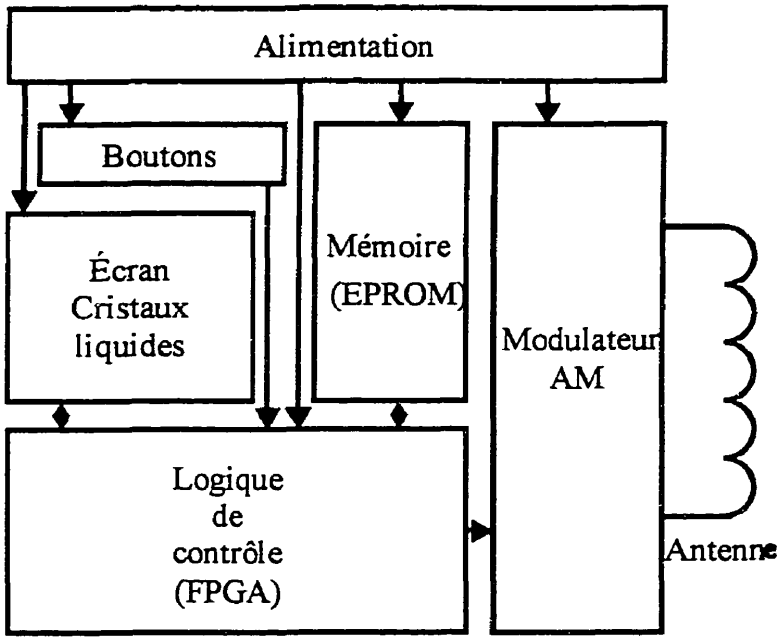



**Figure 2-4 : Schéma-bloc du contrôleur portatif.**

Le contrôleur mesure 19 cm par 10 cm par 3.3 cm et pèse 290 grammes avec la pile. Tout ce qui est requis pour contrôler la stimulation (incluant la pile) est incorporé dans un boîtier unique. Malgré que le circuit de transmission RF nécessite 12 Volts comme tension d'alimentation, il utilise une pile 9V de type D seulement. Un circuit de conversion DC à DC est utilisé pour alimenter le circuit de transmission RF à 12 volts à partir d'une pile régulière 9 volts.

Pour activer une stimulation, l'utilisateur doit choisir le bon ensemble de paramètres en appuyant sur les boutons de sélection jusqu'à ce que l'information recherchée apparaîsse à l'écran. L'information en question est sauvegardée dans la mémoire de façon à représenter un lien au patient dont les paramètres ont été choisis, comme par exemple son nom ou le nom d'un groupe ou la catégorie à laquelle le patient appartient.

Le circuit électronique est principalement composé d'un FPGA, d'une mémoire non-volatile (EPROM) et de la circuiterie analogique d'émission RF. L'EPROM contient 8 ensembles de paramètres en plus que l'information pour remplir les deux lignes de texte par stimulation. Ce nombre d'ensembles peut facilement passer à 16 ou 32 tout simplement en changeant la capacité de la mémoire. Un connecteur spécial est utilisé pour faciliter l'insertion de la mémoire.

Un FPGA de la compagnie ACTEL (modèle 1020B) est utilisé pour intégrer la quasi-totalité de la circuiterie numérique requise dans le contrôleur. Ce type de FPGA est non-volatile et ne peut être programmé qu'une seule fois à l'aide d'un programmeur commercial (« Actel Activator »).

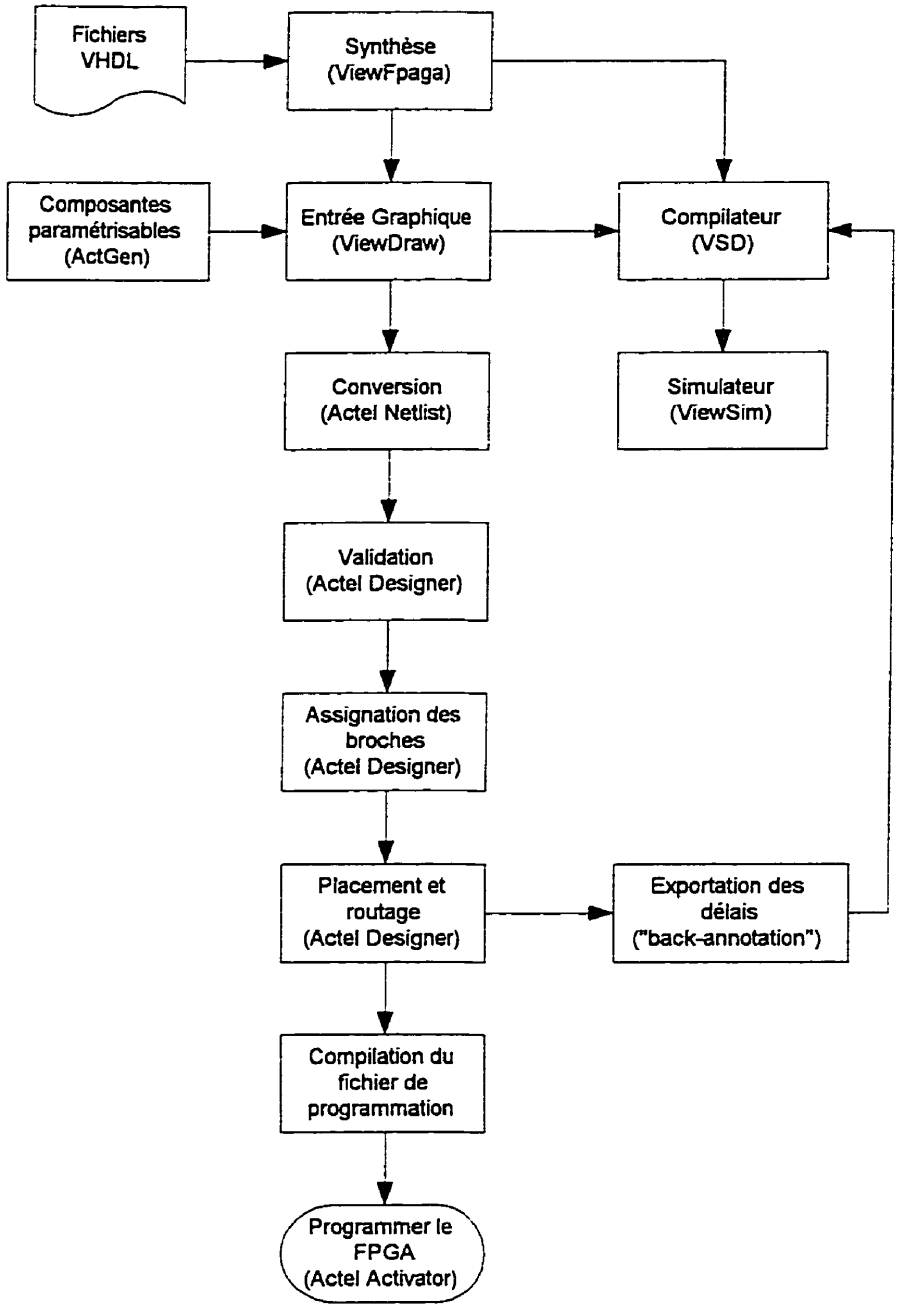



**Figure 2-5 : Étapes de développement d'un FPGA.**

Le circuit à l'intérieur du FPGA a été conçu à partir d'un mélange de synthèse VHDL (« Very high speed integrated circuit Hardware Description Language ») et d'entrée par schémas électroniques, le tout avec les outils ViewLogic et Actel. Il incorpore toute la

logique d'adressage et de contrôle de l'EPROM, la logique de commande et les signaux bi-directionnels utilisés pour le contrôle de l'écran LCD, l'encodeur Manchester et finalement l'électronique des oscillateurs à condensateur ou à quartz. Le schéma ainsi que le code utilisé pour programmer le FPGA se retrouvent à l'annexe I.

Les étapes requises pour compléter la mise au point du FPGA sont présentées à la Figure 2-5.

Une fois le développement terminé, un prototype a été monté sur une plaquette trouée pour valider le design ainsi que les composants programmables. Ensuite, un circuit imprimé dédié (PCB) a été dessiné, réalisé et monté. Le schéma électronique ainsi que le dessin du circuit imprimé se retrouvent à l'annexe I.

### 2.4.2 Contrôleur avec interface pour ordinateur

Ce contrôleur comporte une interface pour le port parallèle d'un ordinateur ainsi qu'un transmetteur AM. C'est un logiciel dédié qui constitue l'interface utilisée par l'usager pour ajuster tous les paramètres de la stimulation. Ce logiciel, programmé à partir du langage C, est très simple d'utilisation et permet de sauvegarder et de récupérer des ensembles de paramètres. Il permet en plus d'exporter les paramètres sous un format compatible avec le contrôleur portatif pour être programmés dans une mémoire de type EPROM.

Lorsque l'interface est branchée sur le port parallèle, le logiciel permet l'activation de la stimulation à l'aide du transmetteur AM interne à partir des paramètres préalablement

choisis. Cette méthode permet une grande flexibilité qui permet de déterminer les valeurs optimales des paramètres pour chacun des patients.

L'interface reçoit les paramètres via l'interface parallèle et les transmet à l'implant par modulation en amplitude avec une porteuse de 20 MHz après les avoir encodés Manchester à la fréquence de 300 kHz.

Le schéma-bloc de l'interface est présenté à la Figure 2-6. Le circuit est composé d'un FPGA qui regroupe toute la circuiterie numérique, d'un émetteur RF et d'une antenne pour transmettre l'énergie ainsi que les parmètres. Deux régulateurs de tension sont utilisés pour fournir l'alimentation du circuit numérique ainsi que celle du transmetteur.

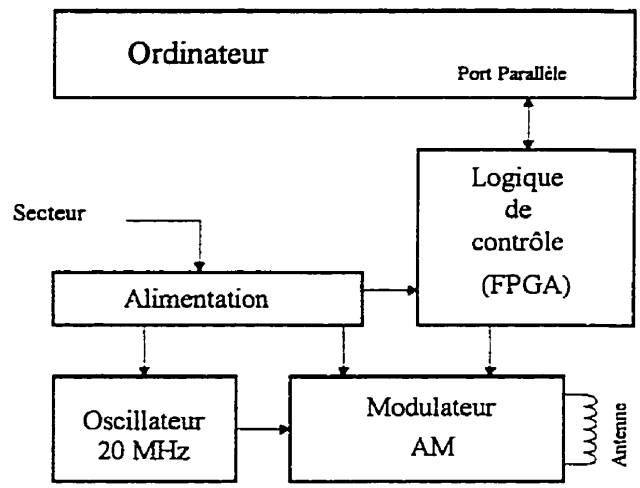



**Figure 2-6 : Schéma-bloc de l'interface de stimulation contrôlée par ordinateur.**

C'est une version modifiée du circuit imprimé destiné au contrôleur portatif qui a été utilisé pour ce contrôleur. Un nouveau FPGA a été réalisé pour cette application à l'aide du langage VHDL. Les outils View Logic et Actel ont aussi été utilisés pour compléter

les schémas, la synthèse VHDL, le placement et le routage automatique. Le plan du circuit imprimé, le contenu du FPGA ainsi que le code C se retrouvent à l'annexe I.

### 2.4.3 Transmetteur RF à couplage électromagnétique

La modulation d'amplitude (AM) d'une porteuse de 20 MHz est utilisée pour transmettre l'énergie ainsi que les données numériques à l'implant via un lien à couplage inductif. Le codage Manchester est utilisé pour transmettre les données avec la synchronisation de façon sérielle. Le modulateur AM est basé sur un oscillateur local qui transmet le signal à un amplificateur de classe D qui à son tour conduit le courant à travers une bobine de transmission [67]. C'est le même transmetteur qui est utilisé pour le contrôleur portatif ainsi que le modèle informatisé. Le plan du circuit se trouve à l'annexe I.

## 2.5 Design du stimulateur implantable (implant) sur circuit imprimé

C'est l'implant qui achemine les stimuli au nerf. L'implant présenté dans ce chapitre est celui qui fut réalisé spécifiquement pour cette recherche et qui est utilisé dans les expériences chroniques sur des chiens au Centre des Ressources Animales (« Animal Resources Center ») de l'Université McGill.

### 2.5.1 Architecture générale

L'architecture de l'implant fut proposée dans l'idée d'une réalisation rapide, simple et efficace. Pour atteindre ces objectifs, cetaines caractéristiques furent décidées dès le début. Le design doit être principalement numérique pour assurer une grande sécurité et

efficacité. Seulement des composants disponibles commercialement doivent contituer le circuit. Contrairement aux implants précédents réalisés par l'équipe PolySTIM dirigé par Sawan [64-69], un circuit imprimé sera utilisé plutôt qu'un circuit hybride. L'utilisation de composants montés en surface (SMT) et un judicieux positionnement de ceux-ci permettent de réaliser un implant de taille suffisamment petite.

Le schéma-bloc du circuit électronique présenté à la Figure 2-7 est composé principalement de trois parties soient : a) la circuiterie mixte de réception; b) la circuiterie numérique de contrôle et c) la circuiterie analogique de stimulation.

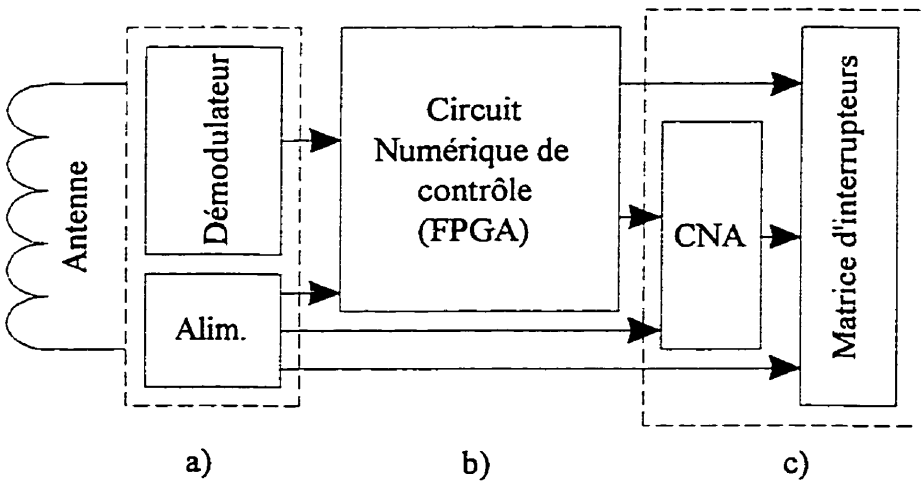



**Figure 2-7 : Diagramme-bloc du stimulateur implantable : a) la circuiterie mixte de réception; b) la circuiterie numérique de contrôle et c) la circuiterie analogique de stimulation.**

### 2.5.2 Circuit mixte de réception

La circuiterie mixte de réception (Figure 2-8) récupère l'horloge et les données sérielles qui sont codés Manchester ainsi que l'énergie nécessaire au bon fonctionnement de l'implant. Un démodulateur AM redresse le signal reçu par l'antenne qui est ensuite

transmis à un détecteur d'enveloppe. Le détecteur d'enveloppe extrait le signal qui est ensuite traité par un décodeur Manchester qui en tire les données, sous forme sérielle, et l'horloge de synchronisation. Le signal sortant du redresseur est aussi filtré et régulé pour produire l'alimentation de 5 V requise par le reste de l'implant.

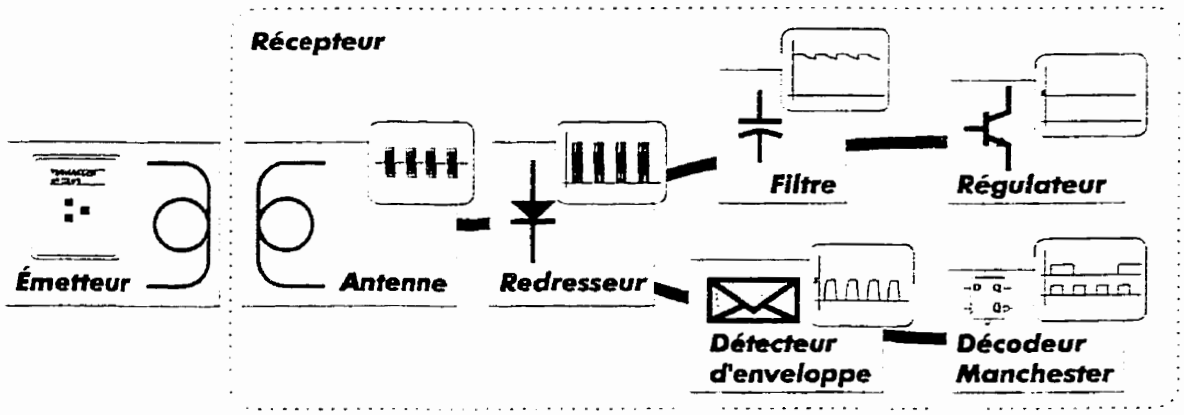



**Figure 2-8 : Schéma-bloc de la circuiterie mixte de réception.**

### 2.5.2 Circuiterie numérique de réception et de contrôle (FPGA)

La circuiterie numérique de réception des données ainsi que de contrôle des stimuli se retrouve entièrement dans le FPGA. La Figure 2-9 représente les principales parties qui constituent l'intérieur du FPGA.

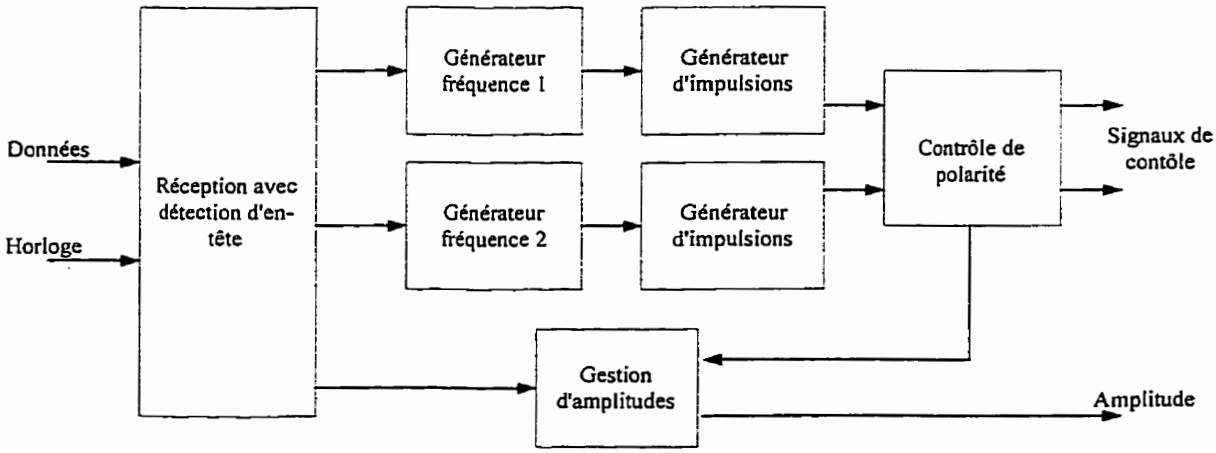



**Figure 2-9 : Schéma-bloc du FPGA de l'implant.**

Le circuit fonctionne de la manière suivante : au moment de la mise sous tension du circuit, tout le contenu du FPGA est remis à zéro sauf le premier bit du registre à décalage d'entrée qui est forcé à 1 (« Power-on reset »). Puisque les données sont transmises de façon sérielle et en continu par le contrôleur externe, un mécanisme de détection d'en-tête est utilisé pour la synchronisation des données qui entrent.

Une fois la synchronisation établie, les données sont injectées dans le registre à décalage d'entrée qui est de 56 bits. Lorsque le "1" qui était au début de la file se retrouve à la sortie du registre à décalage, le registre vient de recevoir les bonnes données, on y verouille alors le contenu jusqu'à la prochaine mise hors-tension et un signal de départ est envoyé aux autres modules. La Figure 2-10 présente l'organisation des données à l'intérieur du registre à décalage tel qu'inscrit dans le fichier de programmation.

```
Parameters Set #1
LF   f= 30Hz  PW=175us  Ampl=0.9mA
HF   f=  0Hz  PW=  0us  Ampl=  0mA
Text line 1:#1_LF_ONLY
Text line 2:
|-------header------|* *|------lfa------|
 1 0 0 1 0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 1 0 +
 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1
|------hfa------|----lfw----|----hfw----|------------lfp------------|-------hfp---------|
```

**Figure 2-10 : Représentation du contenu du registre à décalage tel que retrouvé dans un fichier de programmation.**

Lorsque les générateurs de fréquences 1 et 2 reçoivent le signal de départ, ils divisent alors la fréquence de l'horloge qu'ils reçoivent par la valeur contenue dans la section du registre à décalage qui leur est dédiée. Dans le cas du générateur 1, 14 bits lui sont destinés ce qui lui permet une plage de fréquence allant de 18.3 Hz à 300 kHz avec une

résolution de 3.33µs sur la période. Ce générateur est donc utilisé pour les ondes à basse fréquence. Le générateur 2 possède 10 bits ce qui lui donne une plage de fréquence allant de 292 Hz à 300 kHz et ce aussi avec une résolution de 3.33µs sur la période.

Les deux générateurs d'impulsions reçoivent la fréquence divisée des deux générateurs (ou diviseurs) de fréquence. À chaque transition positive de l'entrée, les générateurs d'impulsions produisent les signaux de contrôle "HAUT" et "BAS" qui sont actifs pendant toute la durée de l'impulsion haute et basse (impulsion bipolaire). Ces deux blocs sont programmés à partir de 8 bits ce qui permet de longeur d'impulsions allant de 0 à 852 µs pour les deux générateurs d'impulsions.

Le bloc de contrôle de polarité fait l'arbitrage entre les deux fréquences et sélectionne l'amplitude en conséquence en actionnant le module de gestion d'amplitude. Il y a donc deux signaux de commande (mutuellement exclusifs) qui sont transmis à la matrice de commutateurs analogiques ainsi que l'amplitude qui est transmise à un convertisseur numérique à analogique sous format 8 bits.

Le FPGA d'Actel (A1020B) avec un boîtier de 44 broches a été choisi premièrement à cause de sa dimension qui est assez réduite et aussi du fait qu'il compte un nombre de portes logiques suffisant pour notre application. Ce FPGA est aussi non-volatile, ce qui est aussi requis pour notre application et finalement, tout le matériel informatique et de programmation requis est disponible dans les laboratoires de l'École.

Pour minimiser le nombre de portes utilisées de façon à ce que toute la fonctionnalité soit contenue à l'intérieur de ce modèle particulier de FPGA, le nombre de bits utilisés pour chaque paramètre a dû être maintenu au minimum requis. De plus, chaque partie du design a été optimisée au maximum. Au début, la première version du circuit avait été réalisée avec du code VHDL. Avec cette méthode, le nombre de portes logiques utilisées dépassait grandement ce qui est disponible dans ce FPGA. Le design a donc été réalisé à nouveau principalement à partir d'entrée schématique manuelle sur l'outil ViewDraw de ViewLogic.

### 2.5.4 Circuiterie analogique de stimulation

La circuiterie qui transmet les impulsions de courant au nerf est constituée en trois blocs. Le schéma électrique de l'implant se retrouve à l'annexe II. Un convertisseur numérique à analogique, un amplificateur en mode courant et une matrice de commutateurs analogiques. Le convertisseur fourni une tension de référence relative qui est transformée en courant par l'amplificateur opérationnel. Ce courant circule alors à travers un chemin formé par deux des interrupteurs analogiques et le nerf, soit dans un sens ou dans l'autre, selon les signaux de contrôle provenant du FPGA. Il faut noter que cette technique n'est valable que si le circuit est entièrement isolé, ce qui est correspond à notre cas.

### 2.5.5 Assemblage de l'implant

Suite à l'accomplissement des différentes étapes de design du circuit allant dans le FPGA, nous poursuivons de la manière suivante :

- Programmation du FPGA : pour cette étape, le fichier de programmation "fuse file" est requis. La programmation prend environ une dizaine de minutes.

- Montage du circuit imprimé : toutes les puces et autre pièces discrètes doivent être soudées de chaque côté du circuit imprimé.

- Réalisation et installation de l'antenne : l'antenne est réalisée à partir de fil de cuivre verni de calibre 28. L'antenne doit contenir trois tours sur un diamètre de 3 centimètres. L'antenne est ensuite soudée du côté du FPGA à une distance d'environ 3 mm du circuit imprimé.

- Fabrication des connecteurs étanches : un fil d'acier inoxidable à multiples brins torsadés recouvert de deux couches de Teflon est utilisé. Un connecteur, de la compagnie AMPS, fait de cuivre plaqué or est attaché et soudé à une extrémité. Le fil est inséré à l'intérieur d'un tube de Silastic. Du Silastic est ensuite moulé autour du connecteur pour lui donner la forme finale. Le tube est soudé au reste du connecteur à ce moment.

- Nettoyage : l'implant entièrement monté est plongé dans un bain ultrasonique rempli de solvent. Cette opération vise à éliminer toute impureté qui pourrait nuire à l'efficacité de la couche de protection (l'enduit protecteur).

- Trempage dans un enduit protecteur et séchage : un enduit protecteur (« conformal coating ») est utilisé pour protéger le circuit contre l'environnement hostile que représente l'intérieur d'un corps vivant. Trois à quatre couches sont appliquées par trempage pour obtenir un enduit d'épaisseur suffisante et uniforme. Un séchage approprié doit être fait entre chaque opération de trempage.

- Moulage final : un moule, principalement constitué de deux plaques d'aluminium et d'un cylindre en cuivre, est utilisé pour cette opération. L'implant doit être positionné dans le cylindre de cuivre en laissant les fils sortir par la fente prévue à cet effet. Pour maintenir l'implant en place, des petits blocs de Silastic sont insérés entre l'implant et le cylindre. Une attache est ensuite enroulée autour du cylindre pour tenir le tout en place pendant le moulage. Quatre vis sont utilisées pour tenir les deux plaques d'aluminium ainsi que l'implant ensemble. Le Silastic est ensuite injecté, à l'aide d'une seringue, dans le trou prévu dans le bas du cylindre. Le Silastic doit être injecté très lentement pour éviter la formation de bulles d'air. Lorsque toute la cavité est remplie, le moule est mis dans un four à 100 degrés Celsius pour environ 15 à 20 minutes. Pour minimiser l'adhérance entre le silastic et les plaques d'aluminium, une mince plaque de Teflon est introduite entre les deux surfaces.

Plusieurs tests sont effectués avant, après et entre la plupart des étapes de fabrication pour assurer la bonne qualité de l'implant. L'implant entièrement terminé est montré à la Figure 2-11. Cet implant est le premier qui a été entièrement fabriqué et utilisé. On peut y voir l'antenne autour du FPGA, les deux connecteurs étanches et le Silastic qui entoure le tout. L'enduit protecteur n'est pas visible puisqu'il est transparent comme le Silastic.

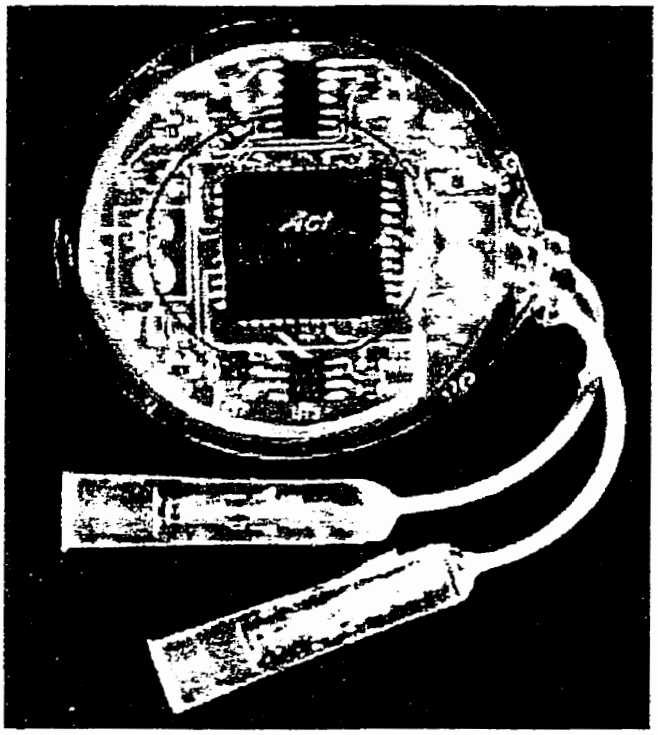



**Figure 2-11 : Photographie du stimulateur implantable.**

## 2.6 Conclusion

Nous avons maintenant fait une description complète de toutes les parties qui composent le système de stimulation électrique dédié à provoquer la miction par stimulation sélective. L'implant, qui a été expliqué dans ce chapitre, a servi d'inspiration pour l'implant sur une puce qui est décrit au chapitre suivant.

# CHAPITRE 3

# DESIGN D'UNE PUCE DÉDIÉE À LA STIMULATION ÉLECTRIQUE SÉLECTIVE

## 3.1 Introduction

Ce chapitre porte sur le design ainsi que la réalisation des parties principales d'un microstimulateur totalement intégré sur un dé de silicium. En premier lieu, la circuiterie analogique de réception et d'alimentation est explorée. Par la suite, la réalisation de la partie numérique de traitement des données reçues ainsi que de la génération des stimuli est décrite. Ces travaux sont très importants pour l'atteinte de l'objectif ultime qui consiste à la réalisation d'un microstimulateur complet ayant des dimensions beaucoup plus petites que tout ce qui est disponible actuellement dans le milieu de la recherche médicale.

## 3.2 Aperçu de l'implant entièrement intégré sur un dé de silicium

L'implant entièrement intégré sur un dé de silicium est constitué des mêmes parties principales que l'implant qui a été réalisé sur un circuit imprimé (chapitre 2). La différence réside dans le fait que, pour cet implant, toutes les pièces doivent être réalisées sur le même circuit intégré. Il faut donc refaire toutes les pièces qui, pour l'autre implant, ont été tout simplement achetées et posées sur le circuit imprimé.

Les fréquences de transmission RF et numérique ont été choisis de façon à rester compatible avec les deux modèles de contrôleurs externes décrits au chapitre 2. Les

paramètres ont été étendus de façon à donner une plus grande plage d'opération ainsi que plus de possibilités pour les expériences futures.

Les figures suivantes représentent la forme d'onde qui peut être générée par l'implant. On peut voir à la Figure 3-1 que la forme est la même à la différence qu'il est possible maintenant d'insérer un délai entre l'impulsion haute et basse de l'onde bipolaire (Figure 3-2).

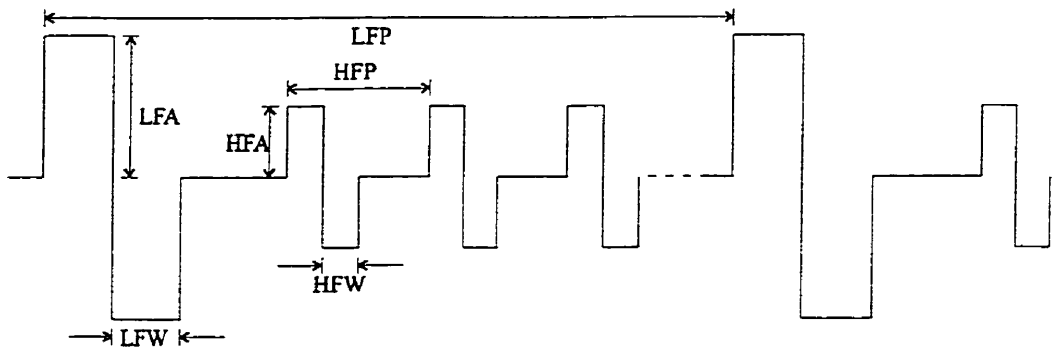



**Figure 3-1 : Forme d'onde générée par l'implant.**

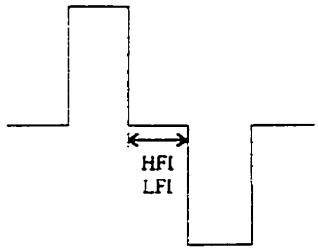



**Figure 3-2 : Délai entre les impulsions.**

Le Tableau 3-1 décrit brièvement les paramètres (Figure 3-1 et Figure 3-2) qui sont transmis de façon sérielle sur l'entrée DATA_IN. L'ordre des données et le mode de détection d'erreur seront formulés ultérieurement.

**Tableau 3-1 : Description des paramètres de stimulation.**

| Abréviation | Nom | Plage | Nb. bits |
|---|---|---|---|
| HFA | Amplitude des pulses de la haute fréquence | 0-4mA | 5 |
| HFP | Période de la haute fréquence | 0-200ms | 16 |
| HFW | Largeur des pulses de la haute fréquence | 0-850us | 8 |
| HFI | Délai entre le pulse haut et bas de la haute fréquence | 0-850us | 8 |
| LFA | Amplitude des pulses de la basse fréquence | 0-4mA | 5 |
| LFP | Période de la basse fréquence | 0-200ms | 16 |
| LFW | Largeur des pulses de la basse fréquence | 0-850us | 8 |
| LFI | Délai entre le pulse haut et bas de la basse fréquence | 0-850us | 8 |

## 3.3 Design du récepteur transcutané

Il existe plusieurs facteurs qui influencent le design d'un récepteur d'implant. Certaines considérations proviennent de l'application tandis que d'autres sont imposées par le procédé de fabrication utilisé. Les considérations de l'application sont : la surface utilisée, la consommation, la durée de vie, la fiabilité et la précision. Le procédé de fabrication comporte certaines contraintes quant à la tension maximale supportée et la précision relative et absolue des valeurs des composants qui servent pour les références.

Un récepteur transcutané typique (Figure 2-8) comporte deux blocs principaux qui sont le bloc d'alimentation et le bloc de réception et de décodage des données qui sont envoyées par le transmetteur. Le bloc d'alimentation est constitué d'un redresseur, de condensateurs de filtrage et d'un régulateur de tension. Le bloc de réception des données contient un détecteur d'enveloppe et un décodeur Manchester.

### 3.3.1 L'antenne

Pour réaliser une bobine de réception par couplage inductif, communément appelée antenne à tort, on utilise normalement quelques tours de fil de cuivre de diamètre approprié. Dans notre cas, cette option n'est pas satisfaisante. Il est donc proposé de réaliser l'antenne à l'aide de traces mises en périphérie du circuit en utilisant la (les) couche(s) métallique supérieure du procédé BiCMOS. L'antenne, ainsi que la circuiterie analogique d'entrée, n'ont pas été réalisées dans ce travail. Normann présente une approche semblable en utilisant une couche métallique mince sur un substrat mince [76].

### 3.3.2 Le redresseur, filtre et détecteur d'enveloppe

Une fois que le signal est transmis et récupéré par l'antenne, il faut en extraire l'énergie et l'information utile. C'est le redresseur qui est responsable de ramener le signal d'entrée dans la plage positive d'opération et ainsi doubler la tension. Un doubleur de tension à base de diodes, qui sont en fait des transistors NPN branchés en diodes, est utilisé.

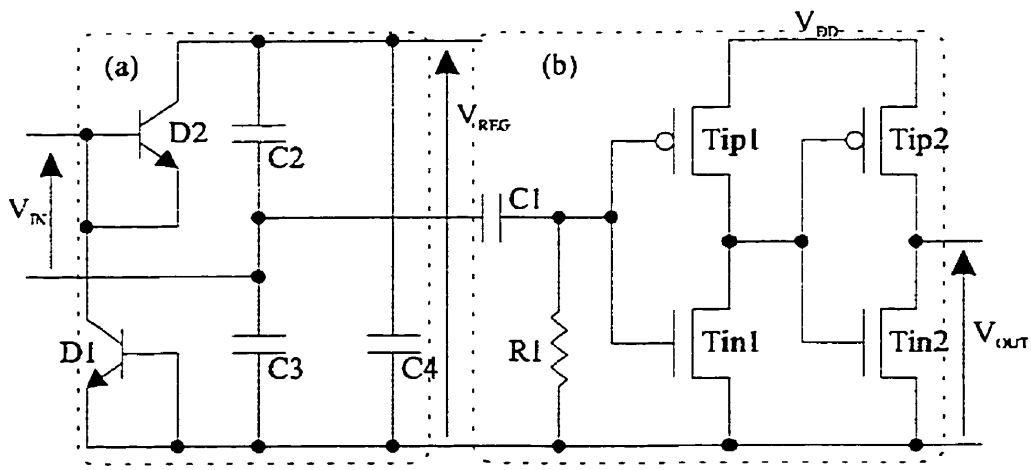



**Figure 3-3 : Circuit de réception : (a) redresseur et filtre, (b) détecteur d'enveloppe.**

L'élément responsable de l'extraction de l'information est le circuit de détection d'enveloppe du signal transmis. Ce circuit, présenté à la figure 3.3-b, est composé par les éléments suivants : le circuit RC (C1, R1) et les inverseurs (Tin1 & Tip1) et (Tin2 & Tip2). Ce circuit reçoit le signal de la sortie du milieu du redresseur. Lorsque ce signal est à zéro, le circuit RC met l'entrée de l'inverseur (Tin1 & Tip1) à un niveau logique de niveau bas. Par conséquence, le signal de sortie de l'inverseur (Tin2 & Tip2) sera aussi de bas niveau. Lorsque ce signal est différent de zéro, le circuit RC met l'entrée de l'inverseur (Tin1 & Tip1) à un niveau égal à la moitié de l'amplitude de la tension de sortie du redresseur qui est supérieure à la tension d'alimentation. Par conséquence le signal de sortie de l'inverseur (Tin2 & Tip2) sera aussi de niveau haut. La sortie de ce circuit représente exactement l'enveloppe du signal transmis.

### 3.3.3 Le décodeur Manchester

Le décodeur Manchester génère l'horloge et les données sérielles à partir de la sortie du détecteur d'enveloppe. Dans notre cas, la configuration classique utilisant des bascules D et un délai de 3T/4 est utilisée tout comme dans le cas de l'implant qui a été réalisé sur circuit imprimé (Section 2.5, chapitre 2).

### 3.3.4 Le régulateur de tension intégré

Le régulateur de tension devrait produire une tension DC de 5V bien stabilisée à partir de la tension de sortie du redresseur. Il devrait aussi fournir le courant requis par l'implant et les électrodes de stimulation. Le régulateur décrit dans cette section est de

type série (ou linéaire). Dans ce genre de régulateur, la tension de sortie est contrôlée en ajustant continuellement la tension aux bornes d'un transistor de puissance connecté entre l'entrée non-régulée et la sortie stable redressée. Comme tout le courant délivré à l'implant passe par le transistor de puissance, ce dernier devrait opérer dans sa région linéaire d'où la nomination régulateur linéaire. Le schéma de ce circuit est donné à la Figure 3-4. C'est un circuit à rétroaction, constitué essentiellement d'un élément de puissance, d'un amplificateur de tension d'erreur, d'un bloc G de génération de tension et de courant de références et d'un circuit de rétroaction. En se référant à la Figure 3-4, le fonctionnement de ce circuit est comme suit : à partir de la tension d'entrée, le bloc G produit le courant et la tension $I_{ref}$ et $V_{ref}$ qui sont indépendants des variations de la tension d'entrée. Le circuit de rétroaction retourne une fraction de la tension de sortie $V_s$ à l'entrée négative de l'amplificateur. Ce dernier compare la tension $V_{ref}$ à la tension $V_S$, et génère un signal correctif afin d'ajuster la tension aux bornes de l'élément de puissance de telle sorte que la condition $V_{ref} = V_S$ soit satisfaite. Par conséquence, la tension de sortie reste stable indépendamment des variations de la tension d'entrée.

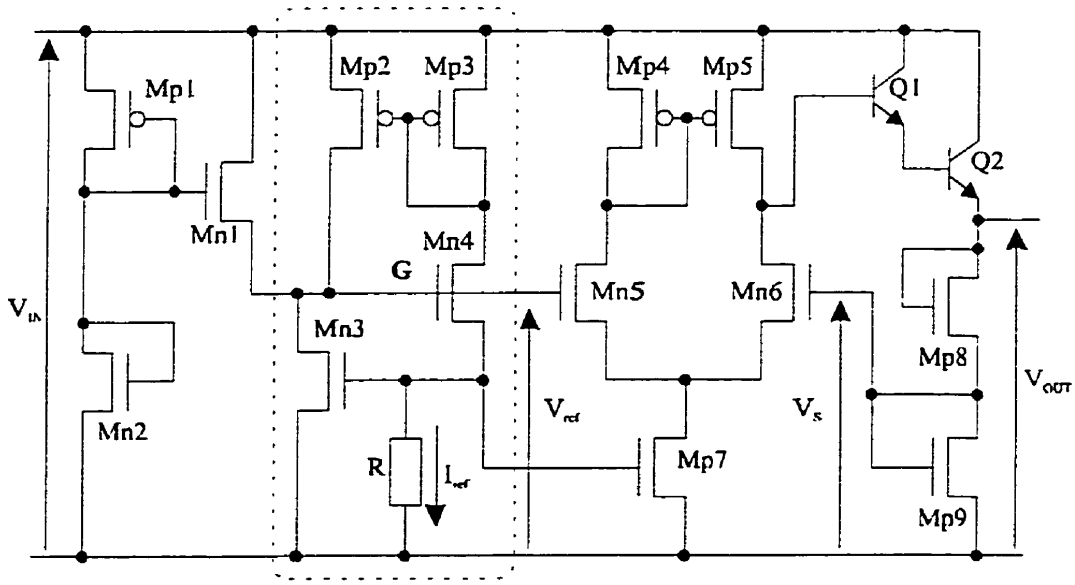



**Figure 3-4 : Circuit du régulateur de tension intégré.**

## 3.4 Circuit numérique de réception, de contrôle et de génération des stimuli

La Figure 3-5 représente les trois principaux modules qui font partie de la partie centrale de l'implant. Le premier module est responsable de la réception des données ainsi que la détection d'erreur. Une fois les données capturées, c'est le module de contrôle des stimuli qui démarre pour transmettre les commandes et les amplitudes requises par le troisième module qui génère alors les stimuli par le biais d'une paire d'électrodes au nerf.

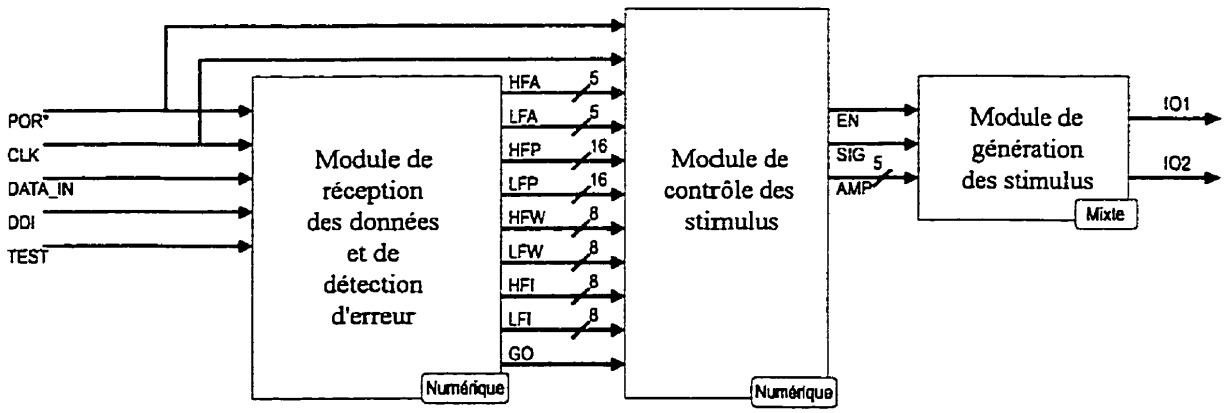



**Figure 3-5 : Schéma bloc du stimulateur intégré.**

Le module de réception des données avec détection d'erreurs est responsable du chargement des paramètres transmis par le contrôleur externe. Les données transmises sont vérifiées à l'aide d'un algorithme de détection d'erreur qui sera présenté plus loin. S'il y a détection d'erreur, la séquence est rejetée. Si elle est bonne, ce module doit charger les données de façon sérielle dans des registres à sortie parallèle. Une fois la séquence complètement reçue, il doit activer un signal qui indique au module de contrôle des stimuli que les données sont prêtes.

Le module de contrôle des stimuli est responsable d'envoyer les commandes au module de génération des stimuli à partir des données mises à sa disposition par le module de réception sous forme de registres accessibles de façon parallèle. Chacun de ces registres d'entrée correspond à un des paramètres de la stimulation. Le module de contrôle des stimuli doit convertir le contenu de ces registres en commandes qui sont transmises au module de génération des stimuli. Les paramètres de temps sont convertis en cycles d'horloges tandis que les différentes amplitudes sont transmises directement au moment approprié.

Le module de génération des stimuli doit générer des impulsions de courant bipolaires et d'amplitudes variables. Le sens du courant est déterminé par un signal de contrôle tandis que l'amplitude est codée sur 5 bits. Ce module est commandé par le module de contrôle des stimuli.

Ces modules seront décrits plus en détail dans les prochaines sections.

### 3.4.1 Méthodologie de design

Les deux premiers modules, qui sont entièrement numériques, ont été réalisés à partir de code VHDL synthétisé avec l'outil « Design Compiler » de la compagnie Synopsys. Les simulations fonctionnelles ont aussi été faites à l'aide des outils de la compagnie Synopsys. L'outil de synthèse "test compiler" a également été utilisé pour l'insertion de structures de test ainsi que pour la génération automatique de vecteurs de tests. Les résultats de synthèse ont été transférés dans l'outil de dessin de masques "Analog Artist" de Cadence pour le placement et le routage automatique avec les fichiers de la librairie "tcells" pour la technologie BiCMOS 0.8μm. Le reste du circuit, soit la source de courant et l'insertion des plots, a été réalisé directement dans l'outil "Analog Artist" de Cadence. Il est à noter que plusieurs « scripts » de la Société Canadienne de Microélectronique (SCM) ont été nécessaires à la réalisation de ce circuit.

### 3.4.2 Module de réception des données avec détection d'erreur

Le module de communication est responsable du chargement des données transmises tout en vérifiant si elles correspondent exactement à ce qui a été émis. Si une erreur est détectée dans la séquence de données reçues, le module doit la rejeter tandis que si elle est correcte, elle doit être décodée puis chargée dans les registres parallèles pour être traitée par les modules subséquents. Notons que tel que décrit précédemment, le signal reçu est déjà démodulé et échantillonné selon l'horloge insérée dans la porteuse par le codage Manchester.

*Problème de la détection des erreurs de transmission*

Puisque le lien de communication est prévu pour fournir l'alimentation de l'implant, le signal est très fort et donc le rapport de la puissance du signal sur celle du bruit est très fortement en faveur du signal. La probabilité d'erreurs de transmission est donc faible. Par contre, la conséquence directe d'une erreur de transmission non détectée est une stimulation fautive pouvant avoir des répercussions sur l'individu porteur de l'implant. La détection d'erreur doit donc fournir une fiabilité de liaison hors pair. D'un autre côté, la latence de traitement n'est pas très importante et le système dispose de plusieurs coups d'horloge pour traiter ses données.

*Algorithme de détection d'erreurs*

Plusieurs familles de codes de détection d'erreurs de transmission existent et offrent toutes des particularités intéressantes. Les contraintes guidant le choix de code de détection ne doivent pas se limiter aux simples particularités du type d'erreurs pouvant survenir. Il ne faut pas perdre de vue les contraintes de complexité et de consommation d'énergie des opérations de décodage qui seront situées dans l'implant. Le code choisi doit donc à la fois être suffisaent pour détecter des salves d'erreurs tout en ne nécessitant que peu de ressources pour le décodage. Étant donné la structure du contrôleur d'implant actuel, il est également essentiel que le code puisse être calculé à l'avance pour être mémorisé dans l'EPROM.

Le choix s'est porté vers la classe des codes linéaires, systématiques et cycliques en raison de la simplicité de leur décodage qui est réalisé par une division polynomiale. Cette

division peut être mise en oeuvre à l'aide de simples additionneurs modulo 2 et de registres. La structure obtenue est efficace au niveau de la surface requise pour l'opération et n'engendre pas une latence de traitement intolérable. Plusieurs codes font partie de cette classe, mentionnons les codes CRC (*Cyclic Redundancy Check*) et codes de Hamming.

Ces codes permettent donc de détecter et de corriger des erreurs de transmission. Cependant, dans notre application, la correction d'erreurs à partir du code n'est pas souhaitable puisque la séquence de données est transmise en boucle continue et que l'opération de correction nécessiterait du matériel supplémentaire.

*Système proposé*

Plusieurs approches peuvent résoudre les problèmes précédemment exposés. La solution proposée divise le problème en trois étapes distinctes : le module de détection d'en-têtes, le traitement des données (mise en forme et détection d'erreurs) et le post-traitement des données. Notons que si plusieurs types de séquences sont utilisés, plusieurs blocs de traitement devront être implantés tandis que le détecteur d'en-tête ne devra rechercher que plusieurs en-têtes au lieu d'une seule.

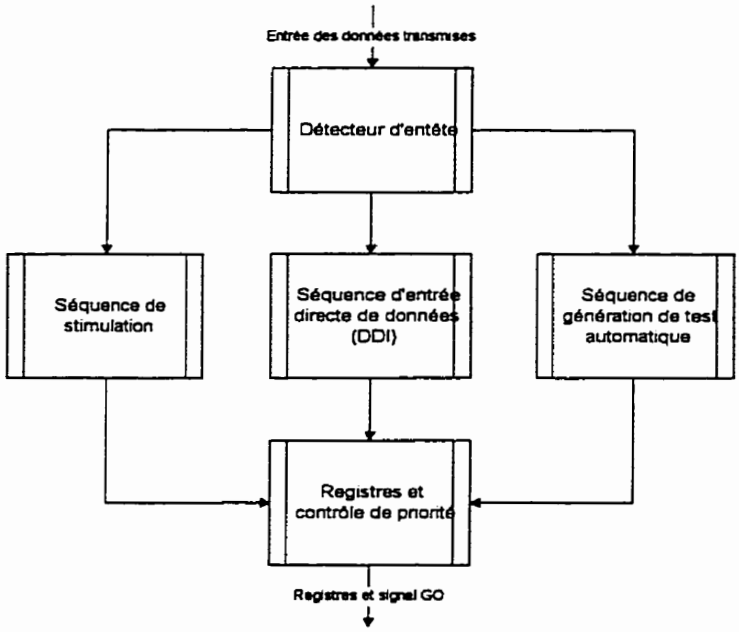



**Figure 3-6 : Schéma général du décodeur réalisé.**

*Détecteur d'en-tête*

L'insertion d'une en-tête à la séquence binaire est essentielle afin de pouvoir en détecter le début. Sans cet ajout, la synchronisation et la mise en forme de la séquence binaire serait ardue et demanderait beaucoup de matériel. Par contre, l'utilisation d'une en-tête engendre d'autres limitations. Principalement, celle-ci doit être unique, c'est-à-dire qu'elle ne doit pas se rencontrer dans la séquence de données. Il est donc essentiel de réserver une certaine chaîne de bits qui ne pourra pas être utilisée dans les mots de commande.

Le détecteur d'en-tête constitue donc le premier bloc du module de communication. Les autres signaux de ce module d'entrée sont le « Power On Reset », actif bas (/POR), l'horloge et les données sérielles. L'algorithme choisit ne fait que contrôler une pile de

type FIFO où l'entrée sérielle est insérée à chaque coup d'horloge (horloge à l'état haut) puis compare la séquence mémorisée avec l'en-tête valide. La dernière donnée sérielle mémorisée est ensuite disponible en sortie à l'état bas de l'horloge en plus de signaux indiquant si une en-tête valide a été détectée.

*Modules de traitement de la séquence binaire*

Une fois qu'une en-tête a été détectée, il faut préparer les données avant leur traitement. En effet, différents formats de séquence binaire peuvent être choisis selon des considérations de probabilités d'erreur ou d'organisation du contrôleur externe. Par exemple, pour un canal où les erreurs se produiraient par groupe, des données normalement consécutives pourraient être séparées dans la séquence pour tenter de rendre les probabilités d'erreurs indépendantes. Le module de mise en forme replace les données afin de pouvoir ensuite réaliser le traitement le plus performant et économique possible. Les entrées communes des modules de traitement sont le « /POR », l'horloge, l'entrée de données sérielles ainsi que le signal de validité de leurs en-têtes respectives. Les sorties sont les registres parallèles des paramètres de stimulation et un signal indiquant que la mise en forme a été réussie. Le design courant comprend trois types de séquences : séquence de stimulation, séquence de test automatique et séquence d'entrée directe de données.

Le décodage est réalisé au moyen de deux modules : le contrôleur de décodage et les décodeurs. Les paramètres de la séquence de stimulation (74 bits) sont codés sur 5 mots de 16 bits (80 bits) qui sont ensuite encodés par le codeur qui ajoute 15 bits par mot.

Pour faciliter la mémorisation des mots dans une ROM par le contrôleur externe, nous complétons chaque mot à 16 bits ce qui permet également d'éviter que la redondance corresponde au caractère d'échappement d'en-tête. Évidemment, la dernière étape du traitement est d'extraire les paramètres de stimulation des mots codés.

Remarquons finalement que les opérations de détection d'erreurs sont réalisées indépendamment pour chaque mot. Deux raisons motivent ce choix : la complexité des modules de détection est moindre et la latence de décodage est inférieure à la durée d'une séquence.

*Séquence de test des modules analogiques*

La séquence de tests analogiques n'a pas vraiment besoin de données. La seule en-tête suffit à indiquer à l'implant que le contrôleur demande un test des modules analogiques. Par conséquent, la mise en oeuvre actuelle effectue une recherche de l'en-tête de test et transmet un signal (« TEST ») au générateur de signal si aucun autre signal n'est valide.

Par contre, une séquence aussi courte sera plus sensible aux erreurs. Des données permettraient aussi de définir différents types de tests sans pour autant utiliser des en-têtes distinctes. Pour toutes ces raisons, il serait préférable d'implanter une en-tête de test suivie de paramètres définissant le test désirant être appliqué. Comme pour la séquence de stimulation, l'utilisation d'un code de détection d'erreurs serait préférable tout comme l'entrelaçage des mots de données et de leur code cyclique.

*Séquence d'entrée directe des données*

Le mode d'entrée directe des données a été ajouté afin de pouvoir tester l'implant en laboratoire avant son utilisation. Les signaux d'entrée sont le « /POR », l'horloge, le signal indiquant que l'en-tête est détectée, l'entrée sérielle des données ainsi qu'un signal actif haut appelé DDI (*Direct Data Input*) provenant de l'extérieur de l'implant (sonde). Pour utiliser ce mode, le signal DDI doit être activé et une séquence de données en format DDI doit être transmise à l'implant. Le format de la séquence d'entrée directe n'inclus aucun recours à un algorithme de détection d'erreurs. En effet, la seule utilité de ce mode est de pouvoir tester simplement et rapidement la fonctionnalité de l'implant. Notons que le signal DDI devra être définitivement fixé inactif avant l'encapsulation de l'implant pour éviter tout fonctionnement erroné.

### 3.4.3 Module de contrôle des stimuli

L'unité de contrôle des stimuli peut opérer dans le mode « test » ou le mode « stimulation ». Le mode "test" est utilisé pour tester la source de courant. Les commandes numériques envoyées par l'unité de commande servent à générer une rampe. Pour le mode "stimulation", l'unité de contrôle est responsable d'envoyer les commandes numériques au module de génération des stimuli (source de courant commandée) à partir des données fournies par le module de réception. Les paramètres de stimulation sont contenus dans 8 registres (LFP, LFW, LFI, LFA, HFP, HFW, HFI et HFA). Puisque l'unité de contrôle traite simultanément deux signaux (basse et haute fréquence), il y a donc 4 registres utiles pour caractériser un signal.

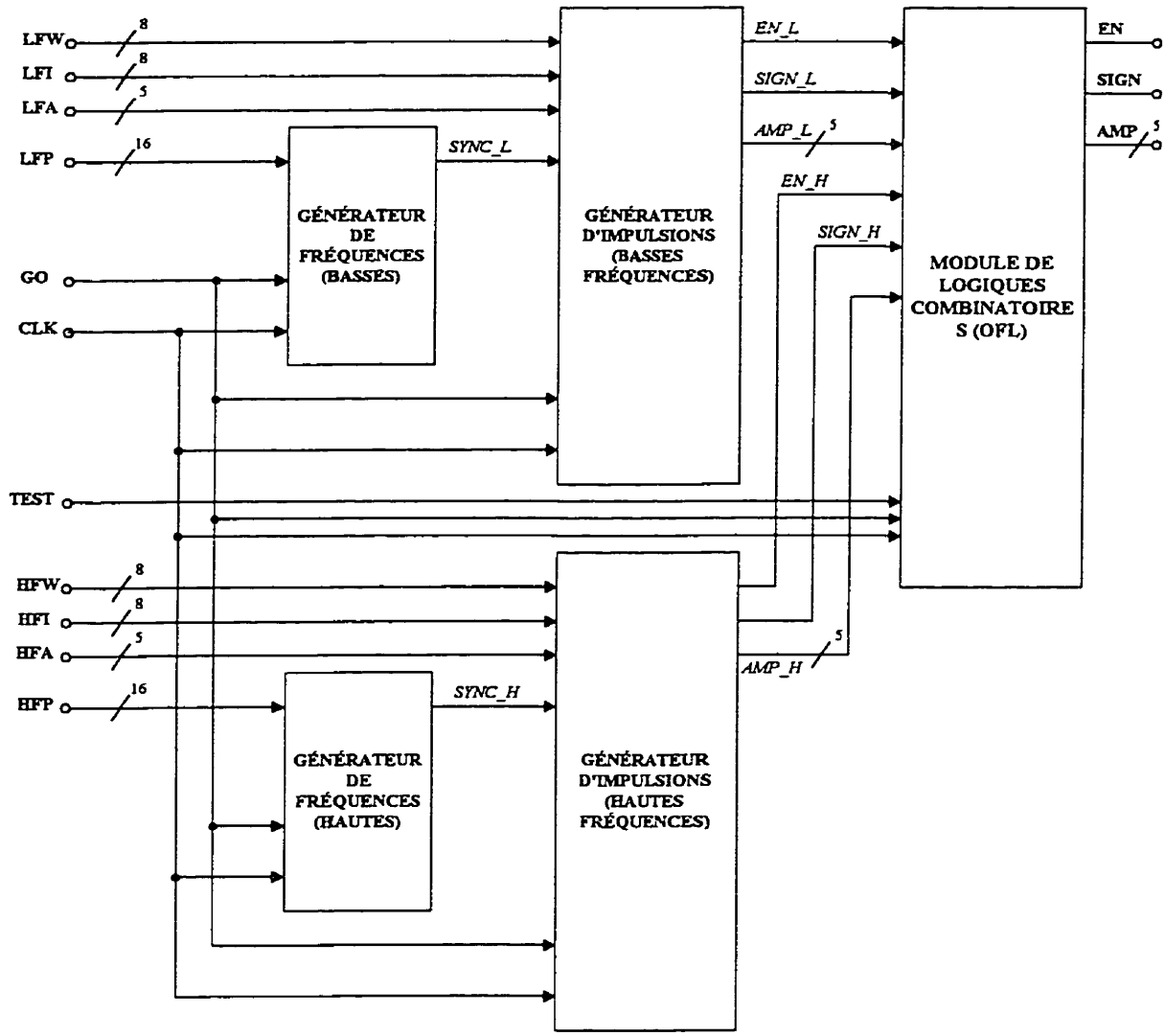



**Figure 3-7 : Schéma-bloc de l'unité de contrôle.**

Afin de faciliter le développement de l'unité de contrôle, nous traiterons les deux ondes indépendamment pour ensuite les superposer. La Figure 3-7 illustre les principaux modules formant l'unité de contrôle. Ces modules sont utilisés comme composants. Par conséquent, la description VHDL de l'unité de contrôle est structurelle. L'unité de contrôle utilise fondamentalement 3 modules ("entity") : le générateur de fréquences, le

générateur d'impulsions et le module de logiques combinatoires. La description VHDL comportementale de ces modules est de type algorithmique.

*Le générateur de fréquence*

Le générateur de fréquence est responsable de déterminer la période de l'onde à traiter. Un registre de 16 bits appelé FP, un signal GO et une horloge CLK constituent les entrées de ce module. Le but de ce module est de générer une impulsion haute sur la sortie (SYNC) tous les "FP" coups d'horloge. Lorsque le signal GO, provenant du module précédent, est à un niveau haut, cela signifie que le contenu du registre FP est valide. De plus, il est à noter que l'impulsion haute sur SYNC dure une période d'horloge afin de permettre au générateur d'impulsions de reconnaître le début de la période du signal.

*Le générateur d'impulsions*

Le générateur d'impulsions sert à générer les commandes numériques qui caractérisent l'onde analogique. Les entrées de ce module sont constituées de deux registres de 8 bits appelés respectivement FW et FI, d'un registre de 5 bits appelé FA, du signal GO, de l'horloge CLK et du signal SYNC provenant du générateur de fréquences. Pour commander numériquement la source de courant, trois sorties sont nécessaires. Tout d'abord, la sortie EN détermine si la source de courant est active ou non. La sortie SIGN contrôle la polarité de l'amplitude contenue dans le registre de sortie AMP (5 bits). Pour générer les bonnes commandes, le générateur d'impulsions utilise FW pour déterminer la largeur de l'impulsion, FI pour déterminer le délai entre l'impulsion haute et basse, et FA

pour déterminer l'amplitude. La période du signal est connue grâce au signal SYNC. En réalité, le générateur d'impulsions est initialisé lorsque le signal SYNC='1'. À ce moment, on évalue de manière séquentielle les registres FW (impulsion haute), FI (délai entre l'impulsion haute et basse) et de nouveau FW (impulsion basse) en prenant soin de modifier la polarité de SIGN lors de la deuxième évaluation de FW. De plus, afin de maximiser les performances de la source de courant, le générateur d'impulsions désactive la sortie EN lorsque l'amplitude de l'onde est nulle.

*Module de logique combinatoire*

Ce module sert à superposer les deux ondes (basse et haute fréquence) lors de la stimulation ou à générer une rampe pour tester la source de courant. C'est l'entrée TEST qui détermine si l'unité de contrôle opère dans le mode « stimulation » (TEST=0) ou dans le mode "test" (TEST=1). Pour caractériser ces ondes, les entrées du module sont les registres EN_L, SIGN_L et AMP_L, EN_H, SIGN_H et AMP_H. Évidemment, l'horloge CLK constitue aussi une entrée de ce module. Les sorties de ce module sont connectées directement à la source de courant commandée. Les fonctions des ces sorties sont identiques à celles du générateur d'impulsions et se nomment EN_SOMME, SIGN_SOMME et AMP_SOMME. Il est à noter que ce module vérifie que l'amplitude maximale de sortie ne dépasse jamais '11111' soit 4mA.

### 3.4.4 Étage de sortie : module de génération des stimuli

C'est la source de courant réalisée par Sylvain Bourret qui a été utilisée dans le circuit qui a été réalisé avec la CMC. J'en résume ici les points principaux.

L'étage de sortie de l'implant est constitué d'une source de courant commandable. Cette source doit pouvoir fournir un courant de 4 mA à une charge de 1000 ohms, soit l'impédance caractéristique d'un nerf. De plus, étant donné que la source est destinée à un circuit implantable, elle doit occuper une surface minimale et dissiper aussi peu de puissance que possible. La source possède 32 niveaux de courant de sortie (5 bits) allant de 0 à 4 mA. Il est important de rappeler que la linéarité de la source est importante afin d'assurer une stimulation adéquate. Il est aussi important de souligner que le « DC offset » de la source devra idéalement être nul afin de limiter l'accumulation de charges dans les tissus nerveux.

L'architecture utilisée pour ce projet s'inspire de la source de courant développée par Robert St-Amand lors de son projet de maîtrise [64]. On peut en voir le schéma à la Figure 3-8. La source qu'il a proposée peut fournir un courant de 2 mA à une charge de 1KΩ. Son circuit est divisé en trois grandes parties : le convertisseur numérique à analogique (CNA), la logique de commande et l'étage de sortie.

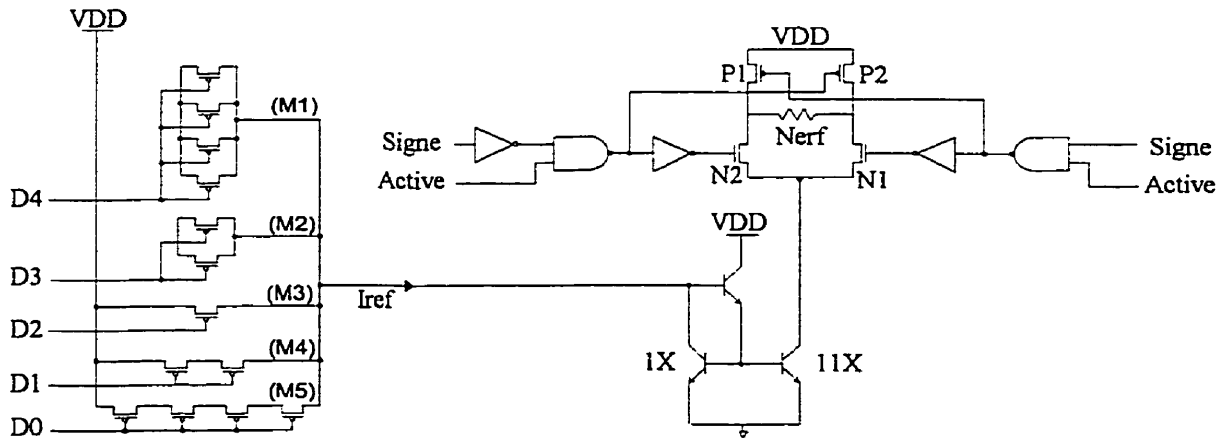



**Figure 3-8 : Architecture de la source de courant proposée.**

Le CNA utilisé par St-Amand comporte l'avantage d'occuper un faible espace tout en possédant une très bonne linéarité. Il est composé de cinq sources de courant pondérées, commandées indépendamment par 5 bits d'activation. La valeur du courant de référence produit est alors la somme des courants produits par chacune des sources actives. De plus, pour assurer une bonne linéarité, il faut s'assurer que les transistors M1 à M5 travaillent en saturation. On utilisera donc une tension de grille supérieure à 0 V pour activer les transistors. Des simulations ont montré qu'une tension de grille de 2.5 V donnait de bons résultats. Étant donné ses caractéristiques intéressantes, nous avons utilisé la même architecture que St-Amand pour le CAN.

La logique de commande du circuit de St-Amand comporte deux entrées. L'entrée SIGN, qui permet de déterminer le sens du courant à travers la charge et l'entrée IDLE qui permet d'activer ou de désactiver la source de courant. Ici aussi, le même principe que St-Amand a été conservé dans le circuit.

Étant donné que la source de courant constitue un étage analogique sensible au procédé et aux erreurs de fabrication, son design final sera fait directement en "layout" en utilisant une géométrie de type centre de masse commun (common-centroid geometry). De cette manière, nous pourrons limiter les effets de gradient dans le substrat et permettre un meilleur "matching" des transistors. Ceci sera particulièrement important pour assurer une bonne linéarité de la source (linéarité du CNA), et une amplitude de sortie adéquate (bon gain du miroir de courant en bipolaire). Le dessin de masque ("layout") final de la source de courant est présenté à l'annexe III. Les dimensions de la source sont de

150x150 microns. On peut clairement y distinguer les différents éléments qui la composent : le CNA (en bas à gauche), le miroir de courant (en bas à droite), la logique de commande (en haut à gauche) et les transistors de sortie (en haut à droite).

### 3.5 Simulations fonctionnelles

Les modules d'entrée et de contrôle ont été simulés séparément pour commencer et ont été regroupés ensuite pour simuler toute la logique ensemble. Pour simuler ces modules, des blocs de test (appelés "testbench") codés en VHDL ont été réalisés. Il est important de limiter au minimum l'intervention du simulateur proprement dit dans toutes les simulations de façon à ne pas être dépendant d'un outil de simulation plûtot qu'un autre dans le futur. En réalisant tout le code des modules de simulation en VHDL, tout ce qui a été créé reste portable sur n'importe quelle plate-forme de simulation logique, en autant qu'elle supporte le VHDL, ce qui est le cas pour la majorité les simulateurs utilisés comme Synopsys, Cadence, Mentor Graphics, Model-Tech et View-Logic pour mentionner les plus connus.

Il faut cependant dire que les modules de tests que nous avons réalisés sont quand même assez primitifs. Par exemple, tous les paramètres utilisés pour les simulations étaient directement codés dans le code au lieu de provenir de fichiers intermédiaires. Cela nécessite alors la compilation et l'élaboration à chaque fois que l'on veut modifier quelque chose dans une simulation. Une approche plus structurée aurait été de réaliser une petit module étant capable de lire un fichier de commande pour transmettre toutes les

instructions aux autres modules constituant le "testbench". De cette, façon, aucune compilation est requise pour changer les paramètres d'une stimulation, en autant que le module de contrôle n'est pas modifié.

De plus, il aurait été intéressant de réaliser un module qui contrôle les entrées du circuit, un module qui capture les sorties et un autre module qui compare les entrées avec les résultats capturés en sortie. Un tel système est beaucoup plus autonome et évite de devoir se fier seulement aux traces logiques temporelles (« waveforms ») comme façon de validation.

La source de courant a été simulée à partir de modèles SPICE générés par Cadence. Le simulateur HSPICE a été utilisé, à plusieurs reprises, pour simuler le comportement de la source de courant sous tous ces angles. La consommation, la linéarité et la plage de courant utile sont les principaux paramètres qui ont été simulés.

Les résultats de simulation pour la source de courant nous donnent une consommation de 21.7 mW lors de la production du courant maximal et ce avec une non linéarité intégrale de 0.37 LSB et une non linéarité différentielle maximale de 0.10 LSB. Il est intéressant de mentionner que la puissance dissipée par le circuit pourra être diminuée en augmentant le nombre de transistors bipolaires (donc le gain) du miroir de courant. Il est évident, cependant, que ceci nécessitera aussi une augmentation de la largeur des transistors utilisés pour former le convertisseur.

## 3.6 Réalisation du circuit intégré

### 3.6.1 Synthèse et testabilité des modules numériques

Une fois que chaque module est simulé seul, au niveau fonctionnel et que tous les modules regroupés ont aussi été simulés, nous pouvons maintenant passer à la synthèse au niveau des portes logiques. Pour cette opération, nous avions deux possibilités au niveau de la technologie BiCMOS soient les tcells et les kcells. Ce sont les deux librairies de portes logiques qui sont fournies et supportées par la CMC.

Pour effectuer la synthèse, nous avons utilisé des scripts avec l'interface texte « dc_shell » plutôt que d'utiliser l'interface graphique « dc_analyzer ». De cette façon, on automatise l'opération et minimise ainsi les risques d'erreurs tout en sauvant du temps.

Nous utilisons aussi l'outil "Test Compiler" qui est intégré à Synopsys. Cet outil permet de générer un design pour le test (DFT) et il comprend aussi un outil de génération automatique de vecteurs de test (ATPG). L'insertion de structures de test ajoute trois ports soient la broche de contrôle, d'entrée des vecteurs et de sortie des vecteurs. La broche de sortie est multiplexée avec une de nos sorties.

Une fois le "script" bien défini, la synthèse devient une tâche triviale, il suffit alors de lancer la synthèse avec une commande unique. Par contre, il ne faut pas penser que la conception des scripts soit une tâche aussi simple. Chaque commande ou option peut prendre plusieurs heures à parfaire étant donné le temps significatif d'une compilation d'un design d'une telle taille. Vous trouverez les résultats de la synthèse à l'annexe III.

Le VHDL est un langage riche qui laisse beaucoup de liberté au programmeur. Dans le cas de la synthèse de circuits numériques bien réels, la plupart des structures plus évoluées sont à proscrire et il est préférable de "coder" de façon propre et simple. Nous avons eu beaucoup de problèmes lors de la conception des modules de décodage. Les types et les structures utilisées au début n'étaient pas synthétisables et ont dû être modifiés en conséquence pour se rendre compte plus tard que le code produisait des cellules non testables à cause de l'utilisation des deux fronts de l'horloge. Par contre, les structures plus évoluées du VHDL peuvent être très utiles pour la réalisation des modules de simulation.

Pour ce qui est de la testabilité, après plusieurs modifications le pourcentage de testabilité a été poussé à 99.99% (couverture de pannes « collées-à ») ce qui est nécessaire étant donné la nature assez critique de notre application.

### 3.6.2 Dessin des masques

Pour réaliser le dessin des masques de notre circuit complet, il faut générer le masque des modules réalisés à partir du VHDL (exporté dans un fichier EDIF), y ajouter la source de courant et finalement y ajouter les plots. Le "layout" de la source de courant est représenté à la Figure 3-9 (tel que réalisé par Sylvain).

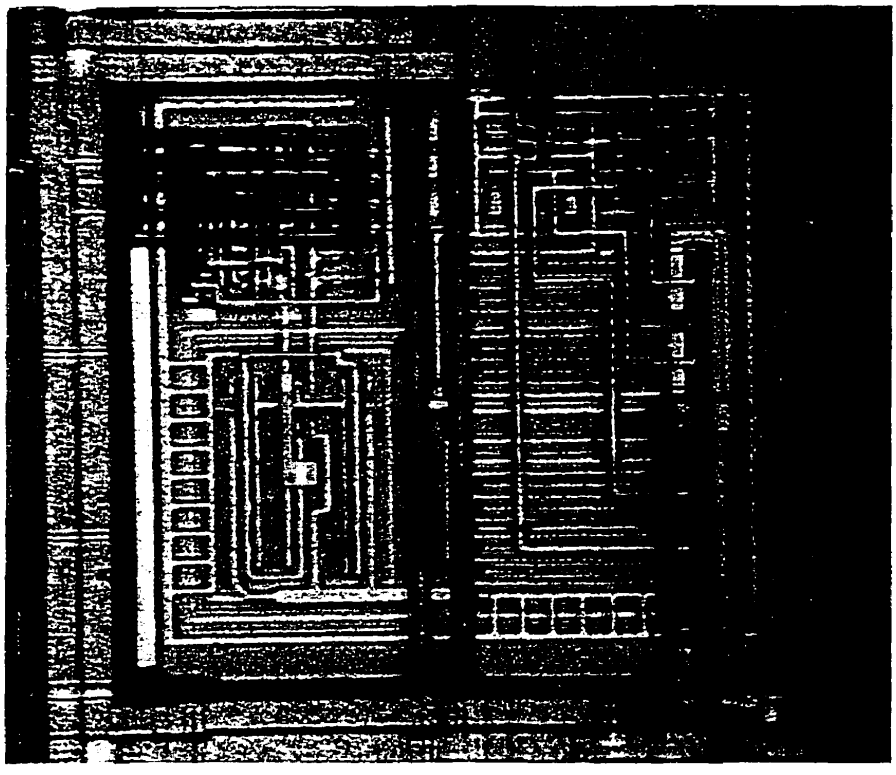


**Figure 3-9 : Protographie de la source de courant monolithique.**

Nous avons décidé d'utiliser, pour la première version, les "tcells" malgré le fait qu'elles soient plus grosses et moins rapides. L'avantage des tcells provient du fait qu'étant plus anciennes, elles sont mieux supportées par les outils (plus "débuggés"). Pour ce projet, la vitesse et la superficie sont des facteurs moins importants que la réussite et la fiabilité. Toutefois, ces dernières sont assez grosses et nous donnent un circuit assez volumineux. La surface finale obtenue avec les "tcells" est d'environ 16mm$^2$ (3620 x 4400 microns$^2$) ce qui dépasse le budget de surface alloué par la CMC. Cette surface doit donc être diminuée par l'utilisation des cellules de la librairie kcells qui sont environ trois fois plus petites que les cellules tcells. Ceci a occasionné plusieurs problèmes avec la librairie kcells au niveau des cellules de "feed-through". Une fois les dessins des masques terminés avec la technologie "kcells", la surface finale du circuit intégré est de

13.5mm² (3000 x 4500 microns²). Le résultat final du dessin des masques se retrouve à la Figure 3-10.

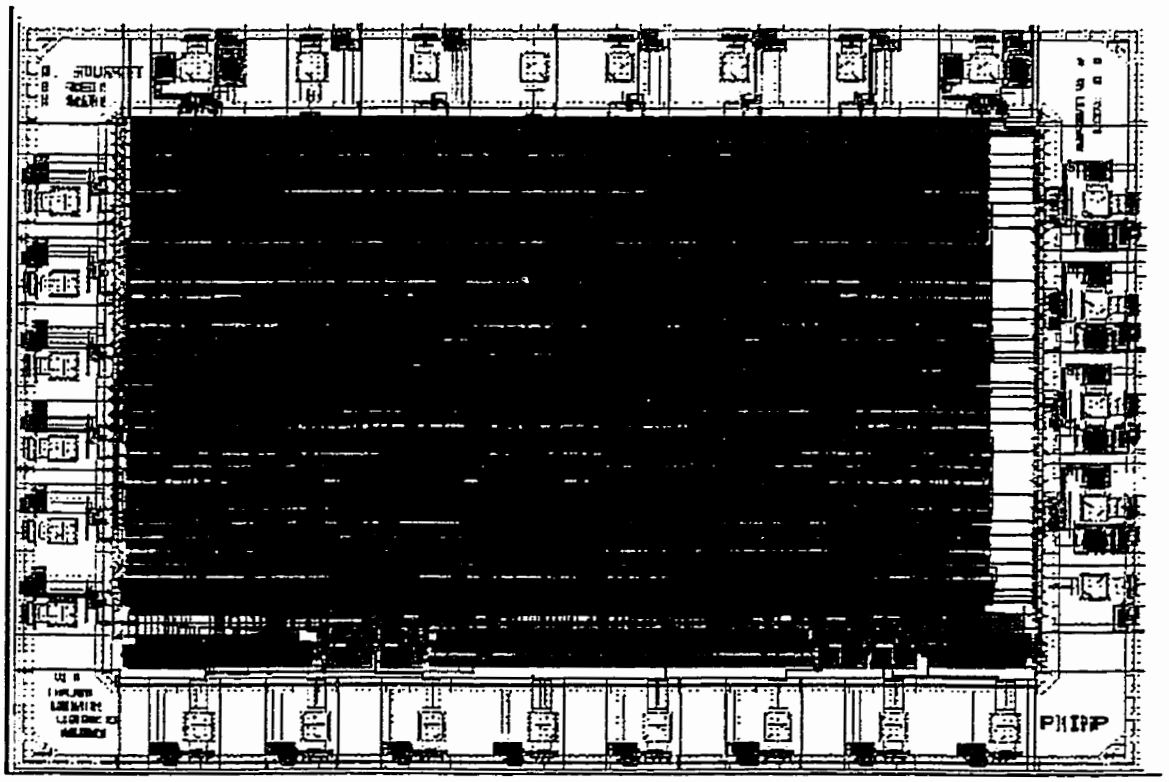


**Figure 3-10 : Dessins des masques de l'implant sur puce.**

Les plots utilisés pour notre design sont de type large ("wide"), ce choix est justifié pour deux raisons. Tout d'abord, la dimension du circuit n'est pas déterminée par les plots "IO bound". On peut donc se permettre des plots plus larges et moins longs afin de diminuer la surface occupée. La deuxième justification provient du fait que notre circuit est destiné à être implanté et alimenté par une onde à 20 MHz. Ainsi, il nous sera impossible de produire deux alimentations différentes pour les plots et le coeur du circuit.

Il est dès lors inutile d'utiliser des plots ayants deux ensembles de rails différents pour les alimentations des plots et du circuit.

Les plots du design sont listés dans le tableau suivant :

**Tableau 3-2 : Liste des plots.**

| Signal | Type | Largeur micron | Hauteur micron |
|---|---|---|---|
| VCC | Supply Vdd | 272 | 370 |
| GND | Supply Gnd | 272 | 370 |
| *POR | Num, Wide | 360 | 370 |
| CLK | Num, Wide | 360 | 370 |
| DATA_IN | Num, Wide | 360 | 370 |
| IO1 | Analog, Wide | 272 | 370 |
| IO2 | Analog, Wide | 272 | 370 |
| TEST | Num, Wide | 360 | 370 |
| DDI | Num, Wide | 360 | 370 |
| Scan Enable | Num, Wide | 360 | 370 |
| Scan Output | Num, Wide | 360 | 370 |
| Scan Input | Num, Wide | 360 | 370 |

Une fois le layout entièrement terminé et validé, les fichiers de masques ont été transmis et la fabrication a été faite par NORTEL à traver la CMC dans les délais qui avaient été prévus. Nous avons reçue 10 prototypes dans des boîtiers et 10 dés sans boîtier.

## 3.7 Conclusion

Dans ce chapitre, nous avons mis ensemble l'ingénierie biomédicale et la microélectronique pour réaliser un stimulateur électrique entièrement intégré sur un seul

dé de silicium. Pour y arriver, il faut réussir à combiner parfaitement la synthèse de haut niveau avec la conception analogique au niveau des masques.

Quand nous avons décidé d'entreprendre la conception et la réalisation de ce circuit intégré dans un délai de 4 mois, nous étions loin d'imaginer l'ampleur de la tâche qui nous attendait. La conception et la réalisation d'un tel circuit comprend une multitude d'étapes qui sont souvent peu ou pas documentées et qui finissent par consommer beaucoup de temps. Par contre, l'expérience acquise durant un tel projet vaut à elle seule les efforts fournis dans ce projet. Il faut souligner que l'équipe possède une très bonne synergie qui nous a permis de réaliser entièrement un circuit intégré de 25000 portes dans un délai très court. Il reste maintenant à tester ce circuit intégré et cette étape sera couverte dans le prochain chapitre.

# CHAPITRE 4

# TESTS FONCTIONNELS ET RÉSULTATS EXPÉRIMENTAUX

## 4.1 Introduction

Dans ce chapitre, un tour d'horizon sera fait sur les tests fonctionnels qui ont été effectués sur les différentes parties du système de stimulation et qui ont fait l'objet des chapitres précédents. Ceci regroupe en premier lieu, les différents contrôleurs et l'implant sur circuit imprimé qui ont fait l'objet du chapitre 2 et aussi l'implant sur une puce décrit au chapitre 3.

Ensuite, l'expérimentation in-vivo, qui est une phase essentielle dans l'élaboration de nouvelles techniques ainsi que le développement de nouveaux systèmes bio-médicaux, sera décrite. Finalement, les principaux résultats expérimentaux seront élaborés.

## 4.2 Résultats des simulations et tests fonctionnels

### 4.2.1 Contrôleurs

Le premier prototype (Figure 2-4) a été réalisé à partir d'un circuit imprimé pré-troué et de fils enroulés (« wire wrap ») et puis soudés. Au début, ce circuit ne devait que servir de prototype pour aider au développement du produit final sur un circuit imprimé dédié. Mais, puisque les choses devenaient pressantes et que son fonctionnement était très satisfaisant, ce « prototype » a été installé dans un boîtier et mis en service pour une période d'environ trois mois, jusqu'à ce que le développement, les tests et la réalisation du

« vrai » contrôleur soient terminés. Ce premier contrôleur a eu la vie dure dans un environnement d'utilisation très difficile et il a rendu l'âme juste après qu'il ait été remplacé. La Figure 4-1 représente le premier circuit de contrôleur portatif. On ne peut y voir toute la circuiterie numérique car elle se trouve en dessous de l'écran LCD. La Figure 4-2 représente la vue extérieure du premier contrôleur qui a été réalisé.

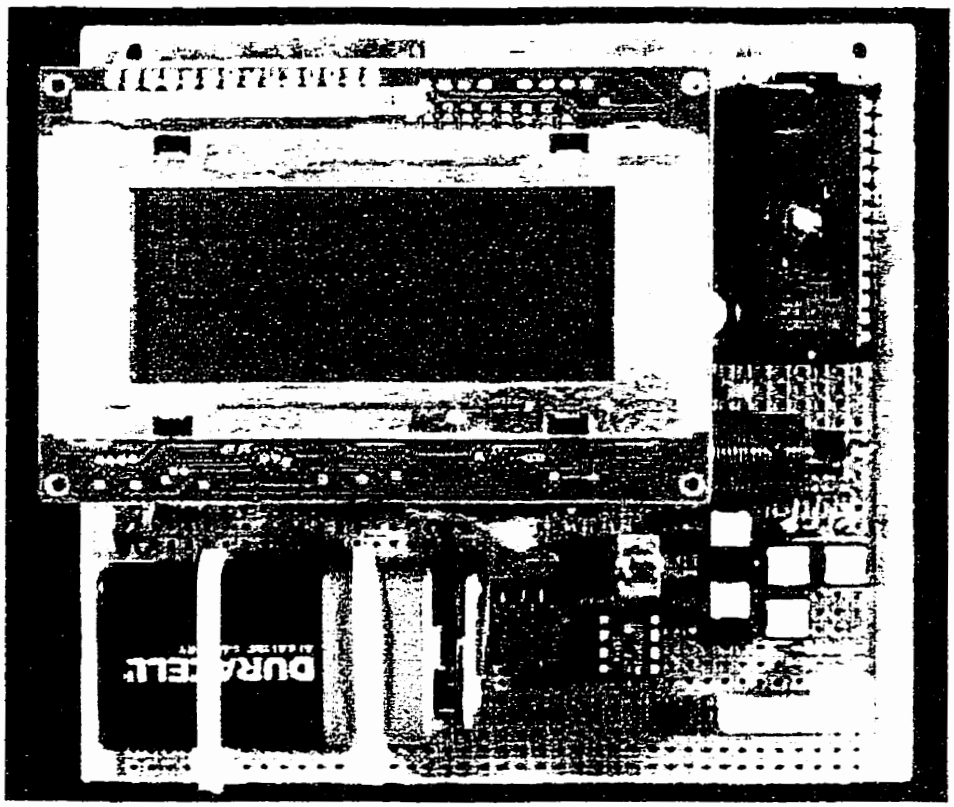


**Figure 4-1 : Premier prototype du contrôleur portatif.**

Aussitôt que les circuits imprimés ont été fabriqués, un premier contrôleur a été monté sans le transmetteur RF. Ce circuit a ensuite remplacé le prototype qui était en service à ce moment et il utilisait lui aussi l'ancien transmetteur RF. Le nouveau transmetteur RF n'a pas été utilisé au début car il a été nécessaire de faire plusieurs

ajustements avant de réussir à en tirer les performances voulues. Plusieurs types de transistors ont été utilisés et beaucoup de travail a été fait au niveau des différentes inductances (bobines) contenues dans le circuit.

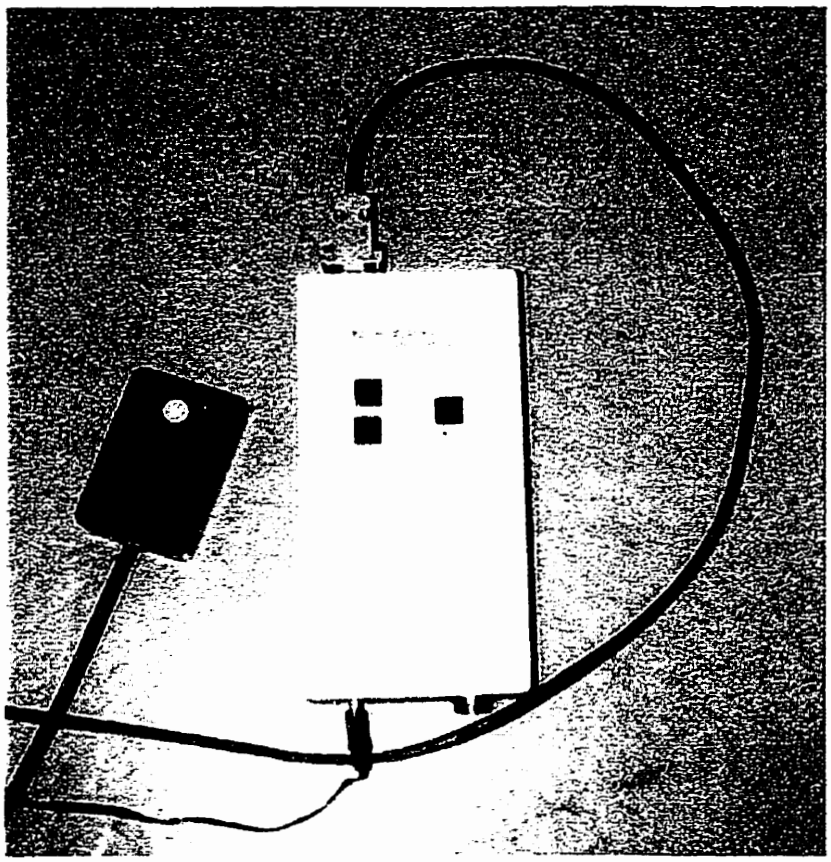


**Figure 4-2 : Premier contrôleur portatif (vue extérieure du boîtier).**

Le deuxième contrôleur portatif a été monté avec le transmetteur RF incorporé dans le même boîtier. Ce contrôleur a été installé dans un nouveau boîtier plus compact et a été configuré pour être utilisé avec une pile de 9 Volts. Ce contrôleur est encore utilisé au moment de la rédaction et ce depuis environ un an.

### 4.2.2 Implant réalisé sur circuit imprimé

Les deux premiers prototypes qui ont été réalisés (Figure 4-3) utilisaient un connecteur (« socket »). Ceci est possible à cause de la particularité des boîtiers PLCC qui permet de les souder en surface ou de les insérer dans un connecteur. De plus, une résistance variable remplace les deux résistances en diviseur de tension qui ajuste le délai de décodeur Manchester de façon à pouvoir trouver les valeurs optimales. Le premier prototype a nécessité l'utilisation d'un inverseur externe supplémentaire pour contrer une erreur qui s'était glissée à l'intérieur du FPGA. Cette erreur a été corrigée dans les FPGAs qui ont été utilisés dans les implants suivants. Une diode a été ajoutée en parallèle avec une des résistances de façon à améliorer les performances du décodeur Manchester.

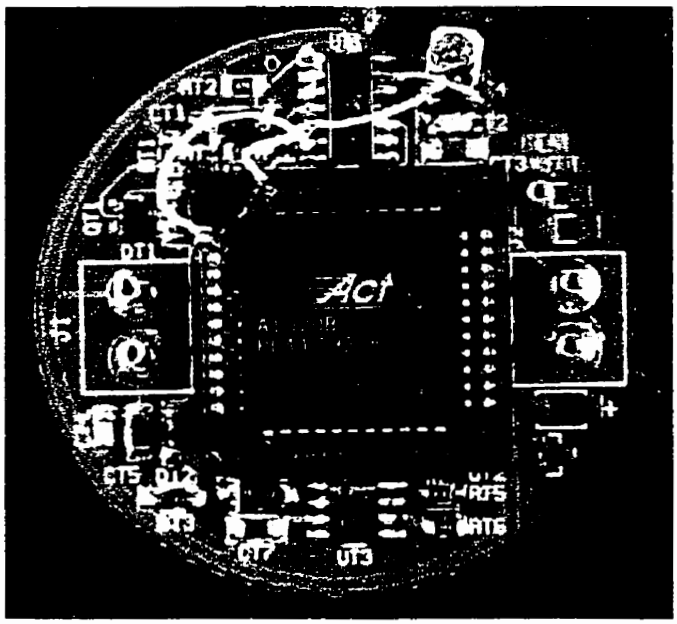


**Figure 4-3 : Photographie du premier protoype d'implant.**

Donc, en bref, deux modifications ont été apportées pour le modèle de production : la correction d'une erreur dans le FPGA et l'ajout de la diode pour augmenter les performances du décodeur Manchester.

Nous avons ensuite monté 8 implants voués à l'utilisation, en soudant le FPGA directement sur le circuit imprimé pour offrir une meilleure fiabilité. Plusieurs tests ont été faits avant et après l'encapsulation de ces implants. Ils ont tous été utilisés et sont encore utilisés à ce jour.

Voici les défaillances qui ont été enregistrées jusqu'à ce jour :

- Un implant a dû être ré-encapsulé suite à une perforation causée par un instrument pointu lors d'une opération;
- Un implant a montré des signes de défaillance assez tôt. Il a été décapsulé pour trouver l'origine du problème. Il s'agissait d'un mauvais contact au niveau de l'antenne. La soudure a été refaite et l'implant a été ré-encapsulé. Il a très bien fonctionné à partir de ce moment et jusqu'à ce jour.
- Un des fils de connecteur étanche a dû être remplacé après qu'il ait été coupé avec un scalpel lors d'une opération.

Il y a aussi eu beaucoup de problèmes qui ont été causés par des ruptures d'électrode. Les spécifications des électrodes se retrouvent à l'annexe IV. Ces électrodes sont

fabriquées au Danemark par Morten Haugland sur mesure, en fonction de nos spécifications. Les premières électrodes étaient réalisées avec du fil AS 632 de la compagnie « Cooner Cables and Wires ». Ce type de fil s'est avéré trop petit et 6 des 8 électrodes implantées ont connu des problèmes dans les 4 à 5 premiers mois d'utilisation. Nous avons remarqué que les électrodes se brisent à la jonction avec le connecteur « AMPS ».

Les électrodes suivantes ont été réalisées avec du fil plus gros (AS 634) et ils fonctionnent encore au moment de la rédaction (7 mois après l'implantation).

Au moment de la rédaction, sur 8 implants réalisés, 7 sont utilisables et 6 sont en utilisation continue. Le huitième ne fonctionne pas pour des raisons toujours inconnues.

### 4.2.3 Implant réalisé sur circuit intégré

Les résultats des simulations de tous les modules numériques ont été satisfaisants. Par contre, en rétrospective, des simulations avec délais auraient dû être réalisées même si la fréquence d'opération est très réduite pour cette technologie. Des problèmes de courses (chemins courts) entre certains signaux peuvent ainsi ne pas être détectés.

Trois types de tests ont été effectués pour cette puce. Les tests fonctionnels de la puce ont été faits en premier, nous avons ensuite utilisé les vecteurs de tests et finalement nous avons testé la partie analogique du circuit soit la source de courant programmable. Malheureusement, tous les tests effectués sur la partie numérique ont donné très peu de

bons résultats. La puce a été testée dans tous les modes possibles et les sorties se sont comportées de façon erratique et incompréhensible.

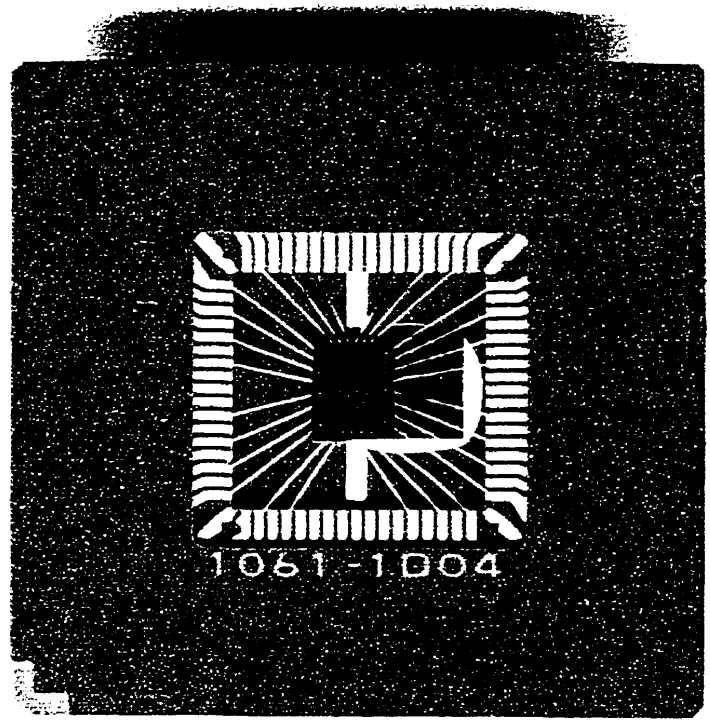



**Figure 4-4 : Photographie du circuit intégré dans son boîtier.**

Les vecteurs de tests ont été appliqués à partir du port parallèle d'un ordinateur personnel et d'un petit programme codé en langage C. Malheureusement, les vecteurs de test ont aussi échoué. Ce genre de test ne permet pas vraiment de trouver la cause du problème, il marche ou non tout simplement.

Pour la source de courant réalisée par Sylvain Bourret de l'équipe de recherche PolySTIM, les résultats initiaux ont été décevants mais un problème majeur qui s'est glissé dans le design a été découvert et a pû être corrigé avec une coupure d'un fil effectuée par impulsions laser. Après l'intervention "quasi chirurgicale", la source de courant s'est mise

à fonctionner adéquatement. Pour plus de détails, se référer au mémoire de Sylvain Bourret qui devrait paraître sous peu (1998).

## 4.3 Expérimentations

### 4.3.1 Préparation des animaux pour expérimentation de courte durée

Sans trop entrer dans les détails médicaux, cette section explique la préparation qui doit être effectuée lors d'une expérience *in vivo*.

La première étape consiste à anesthésier le chien en commençant par une injection de diazepam et de fentanyl-droperidol. Le chien est ensuite intubé et ventilé. L'anesthésie est maintenue avec un mélange d'isofluorane et d'oxygène. La température du corps est contrôlée et maintenue à l'aide de couvertures chauffantes. Le système cardio-respiratoire est vérifié de façon continue.

La pression vésicale est enregistrée via un cathéter à deux trous de dimension 8F. La plage de pression urétrale est identifiée en tirant le cathéter en maintenant un taux de perfusion constant de 2 ml par minute à travers le trou proximal. Une fois la zone optimale repérée, le cathéter est fixé de façon à prévenir les déplacements accidentels.

La vessie est vidée périodiquement et remplie avec 50 ml de solution saline stérile pour chaque série de stimulations. Les pressions vésicale et urétrale sont mesurées et enregistrées à l'aide d'un système informatisé de mesures urodynamiques (UDS 120, Laborie, Technologie médicale). Ce système permet de mesurer simultanément la pression

vésicale, urétrale ainsi que l'activité électrique (électromyogramme) du muscle du sphincter externe en utilisant des électrodes à aiguilles.

Cette technique de mesure comporte le problème suivant : il est difficile de déterminer la pression urétrale à cause de la pression vésicale qui est transmise. Pour remédier à ce problème, une autre technique a été utilisée. Il s'agit d'insérer un ballon de latex dans la vessie et d'y insérer un cathéter à deux trous, un servant à injecter ou évacuer la solution saline et l'autre à mesurer la pression. Un autre cathéter est alors placé dans la vessie pour y permettre l'évacuation de toute l'urine qui est produite pendant l'expérimentation. Avec cette nouvelle technique, on minimise la pression vésicale qui est transmise dans la lecture de la pression urétrale ce qui donne de meilleurs résultats.

La prochaine étape consiste à procéder au sectionnement de la colonne vertébrale au niveau de la vertèbre thoracique T10. Tout en gardant la même position, une laminectomie du nerf sacré est exécutée entre la vertèbre lombaire L7 et la vertèbre sacrée S1. De cette façon, les nerfs sacrées sont exposés. Il faut ensuite identifier le premier et le second nerf sacrés (S1 et S2) à l'aide de la réponse à des stimulations électriques. Un stimulateur électrique simple de modèle SD9 de la compagnie Grass est alors utilisé via une électrode à deux contacts d'une longueur de 10 mm et espacées de 4 mm. Les paramètres alors utilisés sont de 2 à 5 volts pour la tension, une fréquence de 30 Hz, 10 msec de durée d'impulsion et le tout sur une durée de 5 à 10 secondes. Le nerf donnant la meilleure contraction vésicale est retenu pour l'expérience.

Finalement, une électrode à anneaux est attachée sur le nerf qui a été retenu pour être ensuite branchée sur le stimulateur pour expérimentation en phase aiguë qui a été présenté au chapitre 2.

*L'expérimentation en phase aiguë*

On commence par déterminer les seuils de contraction du détrusor et du sphincter par stimulation unilatérale du nerf S2 avec une onde de 30 Hz. On procède en augmentant le courant graduellement. On ajuste ensuite la largeur de l'impulsion de façon à déterminer la combinaison qui donne la meilleure contraction.

La haute fréquence est alors superposée pour effectuer le blocage. Plusieurs essais sont alors effectués avec et sans la haute fréquence pour déterminer les paramètres optimaux. Normalement, les paramètres optimaux pour la haute fréquence sont : fréquence de 600 Hz, amplitude de 0.7 à 1.2 mA et une largeur d'impulsion entre 100 et 250 microsecondes. Chaque stimulation dure 10 secondes avec un repos de 60 secondes entre chacune pour prévenir la fatigue du détrusor et du sphincter.

L'efficacité du blocage par haute fréquence est calculée; soit par le pourcentage de la différence entre la pression urétrale de base et la pression maximale atteinte, soit par l'électromyogramme avec le même calcul. À la fin de chaque séance, l'animal est sacrifié avec un dose massive de nembutal par intraveineuse.

*L'expérimentation en phase chronique*

La préparation est la même que lors des expériences en phase aiguë. Toute l'opération au niveau de la colonne s'effectue de la même façon et une fois que l'électrode est en place, l'implant est branché. On fait alors quelques tests avec l'implant pour s'assurer de la fonctionnalité. On place ensuite l'implant dans une cavité en dessous de la peau en prenant garde de bien y disposer les fils. On referme ensuite avec des soutures adéquates.

Normalement, on peut commencer l'expérimentation sur l'animal après le choc opératoire, soit quelques semaines. Une fois l'expérimentation commencée, la stimulation est effectuée deux fois par jour, à tous les jours, pour chaque animal. Différents résultats sont mesurés et notés comme le volumes d'urine résultant de la stimulation ainsi que le durée de la stimulation.

### 4.3.2 Principaux résultats

Le seuil d'excitation du détrusor et du sphincter ont premièrement été évalués avec une stimulation simple sur le nerf S2. La fréquence ainsi que la largeur d'impulsion ont été fixées initialement à 30 Hz et à 150 μs. Ces paramètres permettent une bonne contraction tout en limitant la fatigue des muscles concernés. La différence entre les fibres A-delta et A-alpha résulte en une contraction du sphincter à un niveau de courant inférieur à celui du détrusor. Le seuil d'excitation du sphincter est inférieur au seuil d'excitation du détrusor de 0.075 mA à 0.3 mA et de 0.12 mA à 0.9 mA respectivement.

Les meilleurs paramètres pour la basse fréquence ont été déterminés de façon à pouvoir appliquer la haute fréquence en combinaison pour ainsi démontrer le blocage et évaluer l'activité du détrusor pendant le blocage. Avec une fréquence de 30 Hz, la contraction maximale se produit avec une largeur d'impulsion de 150 à 180 μs et avec une amplitude allant de 0.45mA à 1.8 mA. La majorité des animaux ont eu la meilleure contraction du détrusor pour une amplitude allant de 0.9 à 1.1 mA (pour 7 des 12 animaux).

Le blocage par haute fréquence est principalement dépendant de la fréquence et de l'amplitude du courant. Dans tous les essais, une fréquence de 600Hz a toujours donné les meilleurs résultats. La plage de courant ayant donné les meilleurs résultats se situe entre 1.1 mA et 1.5 mA. La majorité des animaux réagissaient bien à un courant inférieur à 1.3 mA. La largeur d'impulsion a peu d'influence, en autant qu'elle soit en haut d'un certain niveau se situant à environ 60 μs. Passé ce niveau, son augmentation n'améliore en rien, ou très peu, les résultats.

La Figure 4-5 montre un échantillon des résultats obtenus lors d'une expérience en phase aiguë. Cette figure permet rapidement de démontrer l'efficacité et la validité de la stimulation sélective. On peut y apercevoir trois relevés soient la pression vésicale en haut représentée en noir, la pression urétrale au milieu représentée en bleu et l'activité électrique du sphincter en bas tracée en rouge. On peut y apercevoir l'échelle de temps dans le haut. Dans la section montrée, trois stimulations ont été effectuées. La deuxième, celle du centre, est une stimulation normale, effectuée avec une seule fréquence tandis que

la première et la troisième sont des stimulations électriques sélectives (avec blocage par haute fréquence).

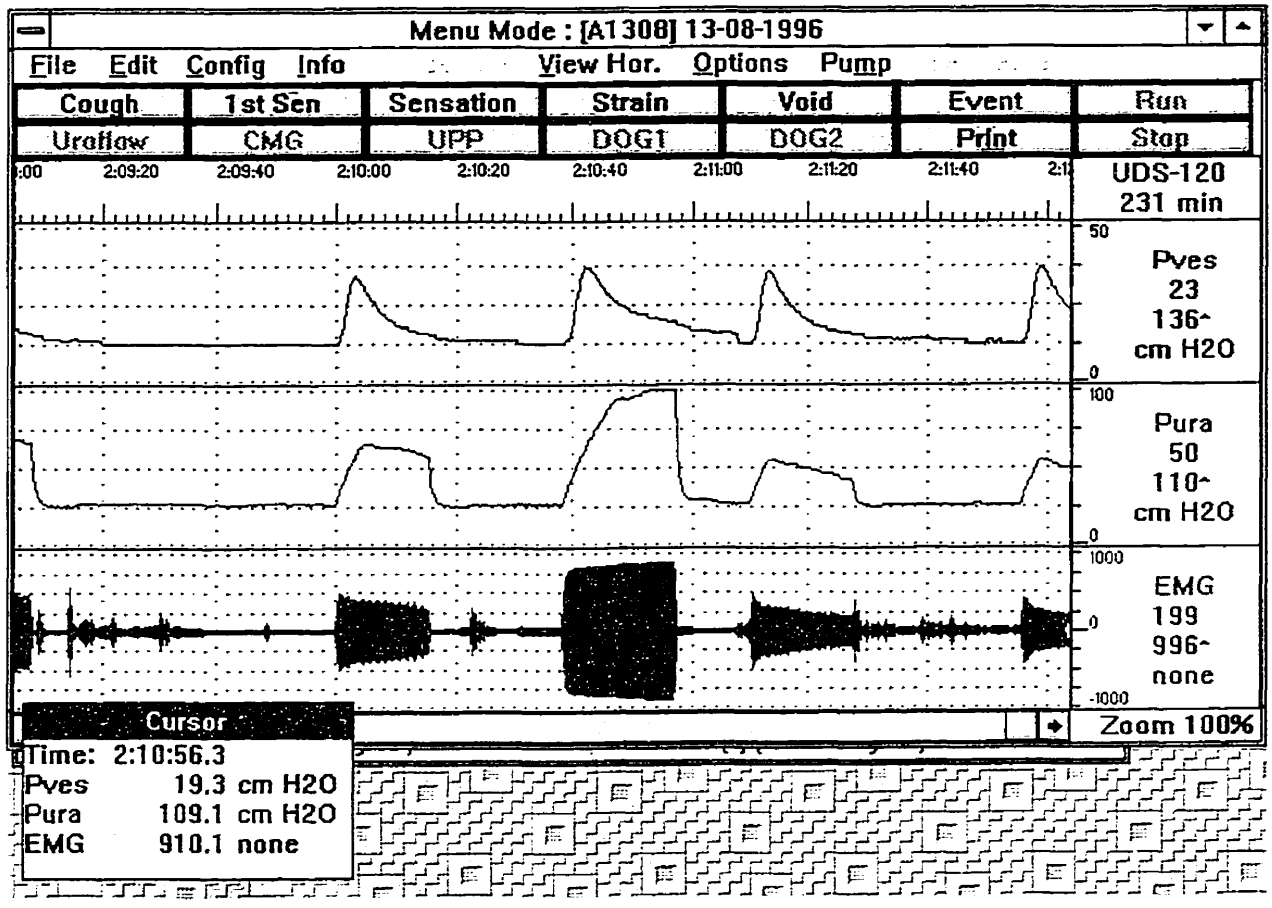



**Figure 4-5 : Résultats de stimulation avec et sans blocage par haute fréquence.**

Puisque le but de la stimulation sélective est de provoquer la miction, il faut avoir une bonne pression vésicale en minimisant la pression urétrale, ce qui est l'objectif du blocage par haute fréquence. On peut donc voir la différence à la Figure 4-5 entre la pression vésicale et l'activité électrique par rapport à la pression vésicale qui est restée quasi inchangée.

Malgré que le blocage par haute fréquence soit maintenant démontré, plusieurs points restent à examiner. Premièrement, le degré du blocage varie en fonction de chaque animal et reste incomplet. La réduction de la pression au niveau du sphincter et de l'activité électrique (EMG) se situe entre 31.1 à 75.6% et 65.7 à 90.3%. De plus, le blocage a réduit la pression vésicale mais beaucoup moins soit entre 3.5% et 36.7%. Les résultats obtenus avec les cinq derniers chiens se retrouvent dans le Tableau 4-1.

**Tableau 4-1 : Efficacité de la stimulation sélective avec les paramètres optimaux.**

| # du chien | Paramètres de la stimulation | | % de réduction | | |
|---|---|---|---|---|---|
| | Basse Fréquence Hz, µs, mA | Haute Fréquence Hz, µs, mA | Pression vésicale | Pression urétrale | Activité EMG |
| 7 | 30, 180, 1.3 | 600, 50, 1.1 | 20.0 | N/D | 90.0 |
| 8 | 30, 180, 1.3 | 600, 50, 1.3 | 36.0 | 31.1 | 65.7 |
| 9 | 30, 180, 0.9 | 600, 60, 1.3 | 20.5 | 75.6 | 90.3 |
| 10 | 30, 180, 0.9 | 600, 90, 1.1 | 12.4 | 70.4 | N/D |
| 11 | 30, 180, 0.9 | 600, 120, 1.3 | 3.5 | 65.8 | 70.4 |

Une fois que les expériences de courte durée ont été terminées et que le système implantable a été fabriqué et validé, les expériences de longues durées ont commencé. Ces expériences doivent être effectuées sur période d'environ 18 à 24 mois ce qui dépasse la durée de cette maîtrise. Malgré tout, les résultats préliminaires obtenus avec le système implantable sont très encourageants. Les résultats finaux sur les essais en phase chronique seront exposés dans des travaux subséquents de membres de l'équipe de recherche PolySTIM.

## 4.4 Conclusion

Dans ce dernier chapitre, les résultats des tests sur les différentes parties du système de stimulation ainsi que du circuit d'implant sur puce ont été décrits. L'expérimentation *in vivo* a aussi été décrite ainsi que les résultats obtenus.

# CONCLUSIONS ET RECOMMANDATIONS

Ce qui a été le plus intéressant dans ces travaux de recherche est le fait d'avoir eu l'occasion de passer de la théorie à la pratique, contrairement à plusieurs autres travaux de recherche similaires effectués depuis quelques années. La première phase d'expérimentation a permis rapidement d'obtenir des résultats qui nous ont menés à l'élaboration des spécifications du système implantable. Il n'était pas question de fabriquer un système implantable sans avoir auparavant des résultats qui validaient une partie de la théorie.

Il s'agit aussi d'un sujet multi-disciplinaire qui a été réalisé en collaboration avec l'Université McGill et le centre hospitalier Royal-Victoria, ce qui rend ces travaux très intéressants.

Bien sûr, pour réaliser un tel système en si peu de temps, il faut avoir accès à plusieurs autres travaux pour pouvoir commencer les expériences *in vivo* avec une bonne idée des stratégies et des résultats envisagés. Dans notre cas, nous avons été en mesure de s'ajuster rapidement de façon à maximiser les résultats obtenus pour chacune des expériences *in vivo* en phase aiguë. Le système utilisé a été amélioré rapidement pour satisfaire aux exigences changeantes.

Le système de stimulation développé et réalisé à l'intérieur de ces travaux de recherche est le premier, depuis celui qui avait été réalisé en prédiffusé par SAWAN et al., à être utilisé pour la recherche. Le besoin d'un nouvel implant était urgent puisque les

implants commençaient à rendre l'âme un à un à cause de leur vieillissement (10 ans environ).

Il est aussi intéressant de savoir que ce système est également utilisé pour effectuer de la recherche dans d'autres domaines comme la réaction des communications nerveuses à certains médicaments.

La réalisation d'un stimulateur entièrement intégré sur une puce rendrait possible l'expérimentation sur des animaux de beaucoup plus petite taille. C'est ce qui nous a incité à commencer des travaux dans ce sens en réalisant un implant quasi complet sur un circuit intégré. Les résultats obtenus avec ce circuit n'ont pas atteint les objectifs fixés au départ mais ces travaux ont quand même ouvert la porte pour d'autres travaux allant dans la même direction. À ce stade, il est important de noter que ce circuit est passé de l'état d'idée à une puce réelle dans un temps record.

Dans le design d'un circuit intégré, il ne faut jamais oublier de se garder des ressources pour simuler à fond le circuit avant de passer à la fabrication. C'est définitivement la partie que nous avons négligée. Un environnement de simulation plus élaboré, simulant le fonctionnement le plus réellement possible aurait été un atout fort appréciable. Dans le même sens, la simulation des vecteurs de tests aurait permis de déterminer leur validité avant de les tester sur la puce.

Les résultats obtenus lors des expérimentations en phase aiguë ainsi que notre système de stimulation électrique ont permis la rédaction de plusieurs articles qui se

retrouvent dans la bibliographie et en annexe. L'expérimentation en phase chronique devrait être terminée par Stéphane Boyer qui travaille aussi dans l'équipe de recherche PolySTIM. Des travaux d'une telle envergure ne peuvent être entièrement réalisés dans une seule Maîtrise, la collaboration entre chercheurs est une nécessité.

Les contributions apportées dans le cadre de cette maîtrise sont :

- le design et la réalisation d'un contrôleur portatif et d'un contrôleur relié à un ordinateur;
- le design et la réalisation d'un stimulateur analogique ainsi que la participation aux expérimentations en phase aiguë conjointement avec des chercheurs de l'Université McGill et de l'Hôpital Royal-Victoria;
- le design et la réalisation d'un implant, basé sur un circuit imprimé et d'une puce programmable, allant des spécifications à l'encapsulation finale;
- la participation active aux expériences en phase chronique avec les implants et les contrôleurs sur plusieurs chiens, aussi en collaboration avec les chercheurs de l'Université McGill et de l'Hôpital Royal-Victoria;
- le design et participation active à la réalisation d'une puce dédié à la stimulation électrique au sein d'une équipe de quatre personnes.

Dans le futur, les systèmes devront être encore plus versatiles, plus fiables, plus petits et incorporer plus de fonctions. L'ajout de fonctions de diagnostic couplées à un système de télémétrie semble être la voie de l'avenir. Plusieurs travaux en cours vont d'ailleurs dans cette direction.

# BIBLIOGRAPHIE


1. ANDERSON, R. U., 1983, "Urodynamic Patterns After Acute Spinal Cord Injury: Association With Bladder Trabeculation in Male Patients", Journal of Urology, Vol. 129, April, pp. 777-779.

2. ARABI, K, SAWAN, M., 1996, "Implantable Multiprogrammable Microstimulator Dedicated to Bladder Control", Medical and Biological Engineering & Computing, January, pp. 48-51.

3. ARABI, K., 1994, "Conception d'un microstimulateur neuromusculaire destiné à la récupération des dysfonctions urinaires", Mémoire de Maîtrise, École Polytechnique de Montréal, Chapitre 3, pp. 53-94.

4. ARABI, K., KAMINSKA, B., 1996, "A New Technique to Monitor the Electrode and Lead Failures in Implantable Microstimulators and Sensors", EMBS Conference 96, Article No. 486.

5. ARABI, K., SAWAN, M., 1994, "A Multiprogrammable Microstimulator Dedicated to Bladder Dysfunctions", Conférence EMBS 1994, Article No. 215.

6. ARABI, K., SAWAN, M., 1995, "A Monolithic Miniaturized Programmable Implant for Neuromuscular Stimulation", Conférence EMBS 1995, Article #816.

7. ARABI, K., SAWAN, M., 1995, "Multiprogrammable Stimulus Waveform Generator for Neuromuscular Electrical Stimulation", Conférence EMBS 1995, Article #815.

8. AVERY LABS., 1978, "The Micturition Stimulator", Avery Laboratories, Farmingdale, NY, 3pp.

9. BARADA, H., SAWAN, M., HASSOUNA, M., TU, L. M., ELHILALI, M. M., 1995, "Sphincteric Fatigue Strategy for Bladder Control : Preliminary Chronic Results in an Animal Model", Conférence EMBS 1995, No. 437.

10. BLAIVAIS, J. G., 1982, "The Neurophysiology of Micturition: A Clinical Study of 550 Patients", The Journal of Urology, Vol., pp. 958-962.

11. BOSCH, R. J. L. H., ABOSEIF, S. R., BENARD, F., STIEF, C. G., SCHMIDT, R. A., TANAGHO, E. A., 1990, "Synchronized Electrical Stimulation of the Sympathetic and Parasympathetic Innervation of the Bladder: Facilitation of the Initiation of Micturition in the Dog", The Journal of Urology, Vol. 144, November, pp. 1252-1257.

12. BOYER, A., SAWAN, M., 1995, "Stimuli Generator Dedicated for an Implantable Visual Miniaturized Stimuator", Conférence EMBS 1995, Article No. 814.

13. BRADLEY, E. W., 1969, "Micturition Reflex Amplification", The Journal of Urology, Vol. 101, March, pp. 403-407.

14. BRINDLEY, G. S., 1994, "The first 500 patients with sacral anterior root stimulation implants", Paraplegia, vol. 32, pp. 795-805.

15. BROCKLEHURST, J. C., 1982, "Noncatheter Devices for Urinary Incontinence in the Elderly.", Medical Instrumentation, Vol 16, pp. 167-168.

16. CREASEY, G. H., 1993, "Electrical Stimulation of Sacral Roots for Micturition After Spinal Cord Injury", Urologic Clinics of North America, Vol. 20, No. 3, August, pp. 505-515.

17. CREASEY, G. H., VAN KERREBROEK, P. E. V., 1996, "Neuroprostheses for Control of Micturition", Journal of Rehabilitation Research and Development, Vol. 33, No. 2, pp. 188-191.

18. D.J.EDELL, J.KUZMA ET AL., 1996, "Implantable electronic systems of tomorrow", Proc. Of IEEE, EMBS'96.

19. EDELL, D. J., KUZMA, J., PETRAITIS, "Tomorrow's Implantable Electronic Systems", Mini-Syposium on Implantable Electronic Systems of Tomorrow, pp. 1-3.

20. FALL, M., LINDSTRÖM, S., 1991, "Electrical Stimulation: A Physiologic Approach to the Treatment of Urinary Incontinence", Urologic Clinics of North America, Vol. 18, No. 2, May, pp. 393-405.

21. FANG, Z-P., MORTIMER, J.T., 1991, "Selective Activation of Small Motor Axons by Quasitrapezoidal Current Pulses", IEEE Transactions on Biomed. Eng., Vol. 38, pp. 168-174.

22. FANG, Z-P., STROTHER, R., 1996, "Reliability Evaluation on an Implanatable Multichannel Neuromuscular Stimulator", EMBS 96 Conference, Article No. 373.

23. FRANCO, I., STORRS, B., FIRLIT, C. F., ZEBOLD, K., RICHARDS, I., KAPLAN, W. E., 1992, "Selective Sacral Rhizotomy in Children with High Pressure Neurogenic Bladders: Preliminary Results", Journal of Urology, Vol. 148, August, pp. 648-650.

24. GERALD, "Electromechanical Restoration of the Micturition Reflex", IEEE Transactions on Bio-Medical Engineering, Vol. BME-18. No. 4, Juillet 1971, pp. 274-280.

25. GOODALL, E. V., DE BREIJ, J. F., HOLSHEIMER, J., 1996, "Position-Selective Activation of Peripheral Nerve Fibers with a Cuff Electrode", IEEE Transactions on Biomedical Engineering, Vol. 43, No. 8, August, pp. 851-856

26. GREATBATCH, W., HOLMES, C., 1991, "History of implantable devices", IEEE Eng. Med. Bio., pp. 38-41.

27. GRILL, W. M., MORTIMER, J. T., 1994, "Effect of Stimulus Pulsewidth on Spacial Selectivity of Neural Stimulation", Conférence EMBS 1994, Article No. 164.

28. GRILL, W. M., MORTIMER, J. T., 1996, "The Effect of Stimulus Pulse Duration on Selectivity of Neural Stimulation", IEEE Transactions on Biomedical Engineers, Vol. 43, No. 2, February, pp. 161-166.

29. GRILL, W., MORTIMER, J. T., 1993, "Selective Activation of Distant Nerve Fibers", Conférence EMBS 1993, Article No. 152, pp. 1249-1250.

30. HARB, A., SAWAN, M., ZHU, J., 1995, "A Wireless CMOS Implantable Receiver for Neuromuscular Microstimulators", WESCANEX'95, Winnipeg, Manitoba, May 1995.

31. HAUGLAND, M., 1996, "A Flexible Method for Fabrication of Nerve Cuff Electrodes ", IEEE/EMBS Proceedings, Amsterdam,.

32. HOFMANN, R., GOMEZ, R., SCHMIDT, R., TANAGO, E. A., 1993, "Effects of Nerve Stimulation on Blood Flow in the Urinary Bladder, Urethra and Pelvic Floor in the Dog", Journal of Urology, Vol 159, December, pp. 1945-1949.

33. HOHENFELLNER, M., PAICK, J-S., TRIGO-ROCHA, F., SCHMIDT, R. A., KAULA, N. F., THÜROFF, J. W., TANAGHO, E. A., 1992, "Site of Deafferenation and Electrode Placement for Bladder Stimulation: Clinical Implications", The Journal of Urology, Vol. 147, June, pp. 1665-1670.

34. HOLSHEIMER, J., STRUIJK, J. J., WESSELINK, W. A., 1996, "Effects of Electrode Configuration and Geometry on Fiber Preference in Spinal Cord Stimulation", EMBS 96 Conference, Article No. 470.

35. JONAS, U. AND TANAGHO, E. A., 1975, "Studies on the feasability of urinary bladder evacuation by direct spinal cord stimulation. Post-stimulus voiding : a way to overcome outflow resistance", Invest. Urol., No. 13, p. 151.

36. JUENEMANN, K-P., LUE, T. F., SCHMIDT, R. A., TANAGHO, E. A., 1988, "Clinical Significance of Sacral and Pudendal Nerve Anatomy", Journal of Urology, Vol. 139, January, pp. 74-80.

37. K.TAKAHASHI, M.KIKUSHI, 1995, "Externally powered implantable FES system", MHS'95 Proceedings of the 6th Int. Sym. On Micromachine and Human Science, pp. 121-126.

38. KAPLAN, S. A., CHANCELLOR, M. B., BLAIVAIS, J. G., 1991, "Bladder and Sphincter Behavior in Patients with Spinal Cord Lesions", Journal of Urology, Vol. 146, July, pp. 113-117.

39. KOLDEWIJN, E. L., RIJKHOFF, N. J. M., VAN KERREBROEK, P. E. V., DEBRUYNE, F. M. J., WIJKSTRA, H., 1994, "Selective Sacral Root Stimulation for Bladder Control: Acute Experiments in an Animal Model", Journal of Urology, Vol. 151, June, pp. 1674-1679.

40. LI, J-P., HASSOUNA, M., SAWAN, M., DUVAL, F., ELHILALI, M. M., "Long-Term Effect of Sphincteric Fatigue During Bladder Neurostimulation", Journal of Urology, Vol. 153, January, pp. 28-242.

41. LI, J-P., HASSOUNA, M., SAWAN, M., DUVAL, F., ELHILALI, M. M., 1992, "Electrical Stimulation Induced Sphincter Fatigue During Voiding", Journal of Urology, Vol. 148, September, pp. 949-951.

42. LIGHT, J. K., FAGANEL, J., BERIC, A., 1985, "Detrusor Araflexia in Suprasacral Spinal Cord Injuries", Journal of Urology, Vol. 134, pp. 295-297.

43. MAGASI, P., SIMON, Z., 1986, "Electrical Stimulation of the Bladder and Gravidity", Urology International, No. 41, pp. 241-245.

44. MALLETTE, S., SAWAN, M., FORTIER, P. A., 1995, "A New Multichannel PC-Controlled Stimulator Developped for Primary Motor Cortex Investigations", Conférence EMBS 1995, Article #813.

45. MAYR, W., BIJAK, M., GIRSCH, W., LANMÜLLER, H., RAFOLT, D., SAUERMANN, S., SCHNETZ, G., UNGER, E., 1996, "Functional electrostimulation via implants applications, limitations, perspectives",

Conference on Interdisciplinary Aspects on Computers Helping People with Special Needs, Linz, Austria, pp. 311-317.

46. MEDTRONIC, 1984, "Pulse Generator Maget", Neuro Division, Medtronic Inc., Minneapolis.

47. MELLICK, W. F., 1970, "Electrical Stimulation in Urology", The Journal of Urology, Vol., pp. 815-821.

48. MERSDORF, A., SCHMIDT, R. A., TANAGHO, E. A., 1993, "Topographic-Anatomical Basis of Sacral Neurostimulation Variations", Journal of Urology, Vol. 149, February, pp. 345-349.

49. MOUÏNE, J., WEI, M., FONTAINE, R., DUVAL, F., 1995, "A Full Custom Sixteen-Channel Cochlear Implant Microstimulator", Conférence EMBS 1995, No. 126.

50. NOVICK, A. C., 1992, "Management of Voiding Dysfunction with an Implantable Neuroprosthesis", Urologic Clinics of North America, pp. 163-170.

51. PERKINS, T. A., 1986, "Versatile Three-Channel Stimulation Controller for Restoration of Bladder Function in Paraplegia", Journal of Biomedical Engineering, Vol 8, pp. 268-271.

52. PROVOST, B., SAWAN, M., "A New Implantable Tomography Approach to Bladder Volume Monitoring", Conférence EMBS 1995, No. 812.

53. RIJKHOFF, N. J. M., *et al.*, 1994, "Acute Animal Studies on the Use of Anodal Block to Reduce Urethral Resistance in Sacral Root Stimulation", IEEE Transactions on Rehabilitation Engineering, June, Vol. 2, No. 2, pp. 92-99.

54. RIJKHOFF, N. J. M., HOLSHEIMER, J., KILDEWIJN, E. L., STRUIJK, J. J., VAN KERREBROECK, P. E. V., DEBRUYNE, F. M. J., WIJKSTRA, H., 1994, "Selective Stimulation of Sacral Nerve Roots for Bladder Control: A Study by Computer Modeling", IEEE Transactions on Biomedical Engineering, Vol. 41, No. 5, May, pp. 413-423.

55. RIJKHOFF, N. J. M., KOLDEWIJN, E. L., VAN KERREBROECK, P. E. V., DEBRUYNE, F. M. J., WIJKSTRA, H., 1994, "Canine Bladder Evacuation by Electrical Stimulation of the Ventral Sacral Nerve Roots", Conférence EMBS 1994, Article No. 633.

56. RIJKHOFF, N. J. M., WIJKSTRA, H., P. VAN KERREBROEK, E. V., DEBRUYNE, F. M. J., 1997, "Selective detrusor activation by electrical sacral nerve root stimulation in spinal cord injury", The Journal of Urology, Vol. 157, April, pp. 1504-1508.

57. ROBIN, S., SAWAN, M., 1996, "A Hand-held Controller Dedicated to Implantable Stimulators", FPD96, Totonto.

58. ROBIN, S., SAWAN, M., 1997, "Micro-Stimulateur Implantable Dédié à la Stimulation Sélective", 1er Symposium International de Biomatériaux Avancés, Octobre, Montréal.

59. ROBIN, S., SAWAN, M., ABDEL-BAKY, M., ABDEL-GAWAD, T., ELHILALI, M. M., 1997, "A New Implantable System for Neural Selective Stimulation of the Bladder, IFESS 97, Vancouver.

60. ROBIN, S., SAWAN, M., ABDEL-GAWAD, M., ABDEL-BAKY, T-M., ELHILALI' m" M., 1998, "Implantable Stimulation System Dedicated for Neural Selective Stimulation", Medical & Biological Engineering & Computing", Accepté en mai 1998 pour publication.

61. ROBIN, S., SAWAN, M., HARVEY, J-F., ABDEL-GAWAD, M., ABDEL-BAKY, T., ELHILALI, M. M., 1997, "A New Implantable Microstimulator Dedicated to Selective Stimulation of the Bladder, EMBS 97, Chicago.

62. ROBIN, S., TU, L. M., ARABI, K., SHAKER, H., SAWAN, M., ELHILALI, M. M., 1996, "A New FES System Dedicated to Selective Stimulation: Acute Studies in Dogs", IEEE/EMBS Proceedings, Amsterdam, The Netherlands.

63. RUDY, D. C., AWAD, S. A., DOWNE, J. W., "External Sphincter Dyssynergia: An Abnormal Continence Reflex", Journal of Urology, Vol. 140, pp. 105-110

64. ST-AMAND, R., SAWAN, M., SAVARIA, Y., 1996, « Design and Optimisation of a Low DC Offset Current Source Dedicated to Implantable Miniaturized Stimulators », Analog Integrated Circuits and Signal Processing , Vol. 11, pp 47-71.

65. SAWAN, M., ARABI, K., PROVOST, B., 1997, « Implantable Volume Monitor and Miniaturized Stimulator Dedicated to Bladder Control », International Society for Artificial Organs, 21(3), pp. 219-222.

66. SAWAN, M., DUVAL, F., HASSOUNA, M. M., LI, J-S., ELHILALI, M. M., 1992, "Computerized Transcutaneous Control of a Multichannel Implantable Urinary Prosthesis", IEEE Transactions on Biomedical Engineering, Vol. 39, No. 6, June, pp. 600-609.

67. SAWAN, M., DUVAL, F., HASSOUNA, M., ELHILALI, M M., 1994, "A New Transcutaneous Fully-Programmable Neural Stimulator", International Journal of Microcomputer applications, vol. 13, no. 3, 00, pp.142-147.

68. SAWAN, M., DUVAL, F., LI, J-P., HASSOUNA, M., ELHILALI, M. M., 1993, "A New Bladder Stimulator - Hand-Held Controller and Miniaturized Implant: Prelininary Results in Dogs", Biomedical Instrumentation & Technology, March/April, pp. 143-149.

69. SAWAN, M., HASSOUNA, M. M., LI, J. S., DUVAL, F., ELHILALI, M. M., 1996, "Stimulator Design and Subsequent Stimulation Parameter

Optimisation for Controlling Micturition and Reducing Urethral Resistance", IEEE Transactions on Rehabilitation Engineering, Vol. 4, No. 1, pp. 39-46.

70. SAWAN, M., HASSOUNA, M., LI, J.-S., DUVAL, F., ELHILALI, M. M., 1993, "Neural Mulichannel Electrical Stimulation for Total Bladder Evacuation", Conférence EMBS 1993, Article No. 332, pp. 1259-1260.

71. SAWAN, M., ROBIN, S., 1997, "Implantable Neural Cuff Electrodes", 1er Symposium International de Biomatériaux Avancés (SIBA), Montréal.

72. SAWAN, M., ROBIN, S., BOURRET, S., BOYER, S., 1998, "A Miniaturized Implantable Bladder Selective Electrical Stimulator", VIII Meditarranean Conference on Medical and Biomedical Engineering and Computing (Medicon'98), June, Lemesos, Cyprus.

73. SAWAN, M., ROBIN, S., PROVOST, B., EID, Y., ARABI, K., 1997, "A Wireless Implantable Electrical Stimulator Based on Two FPGAs", International Conference, Electronic Circuits and Systems, Rodos.

74. SCHNEIDAU, T., FRANCO, I., ZEBOLD, K., KAPLAN, W., 1995, "Selective Sacral Rhizotomy for the Management of Neurogenic Bladders in Spina Bifida Patients: Long-Term Followup", Journal of Urology, Vol. 154, August, pp. 766-768.

75. SELIGMAN, L. J., 1982, "Physiological Stimulators: From the Electric Fish to Programmable Implants", IEEE Transactions on Bio-Medical Engineering, vol. BME-29, no. 4, pp. 270-284.

76. SHAH, M., PHILLIPS, P. R., NORMANN, R. A., 1998, « A Study of Printed Spiral Coils for Neuroprosthetic Transcranial Telemetry Applications », IEEE Transactions on Biomedical Engineering, Vol. 45, No. 7, July, pp. 867-876.

77. SMITH, B., PECKHAM, P. H., KEITH, M. W., 1987, "An Externally Powered, Multichannel, Implantable Stimulator for Versatile Control of Paralysed Muscle", IEEE Transactions on Bio-Medical Engineering, Vol. BME-34, No. 7, July, pp. 499-508.

78. SOTIROPOULOS, A., YEAW, S., LATTIMER, J. K., 1976, "Management of Urinary Incontinence with Electronic Stimulation: Observation and Results", Journal of Urology, Vol. 116, pp. 747-750.

79. STOTTS et al, 1989, "An 8-bit Microcomputer with Analog Subsystems for Implantable Biomedical Applications", IEEE Journal of Solid-State Circuits, Vol. 24, No. 2, Avril, pp. 292-300.

80. TAI, C., JIANG, D., 1993, "A Stimulation Study of Selective Stimulation of Smaller Nerve Fibers by Biphasic Pulses", Conférence EMBS 1993, Article No. 21, pp. 1195-1195.

81. TANAGHO, E. A., 1997, "Commentary on Selective Detrusor Activation by Electrical Sacral Nerve Root Stimulation in Spinal Cord Injury", Journal of Urology, Vol. 157, p. 1196.

82. TANAGHO, E. A., SCHMIDT, R. A., 1988, "Electrical Stimulation in the Clinical Management of the Neurogenic Bladder", The Journal of Urology, Vol. 140, December, pp. 1331-1339.

83. TANAGHO, E., SCHMIDT, R. A., ORVIS, B. R., 1989, "Neural Stimulation for Control of Voiding Dysfunction: A Preliminary Report in 22 Patients with Serious Neuropathic Voiding Disorders", The Journal of Urology, Vol. 142, pp. 340-345.

84. TANG, Z., PECKHAM, H., 1994, "Multichannel Implantable Stimulation and Telemetry System for Neuromuscular Control", Conférence EMBS 1994, Article No. 310.

85. TANGHE et al, 1992, "A 16 Channel CMOS Neural Stimulator Array", IEEE Journal of Solid-State Circuits, Vol. 27, No. 12, Décembre, pp. 1819-1825.

86. TANNIR, H. F., TIMMONS W. D., 1994, "A Model of the Bladder Control System", Conférence EMBS 1994, Article No. 158.

87. THÜROFF, J., BAZEED, M. A., SCHMIDT, R. A., WIGGIN, D. M., TANAGHO, E. A., 1982, "Functional Pattern of Sacral Root Stimulation in Dogs : I. Micturition", Journal of Urology, Vol 127, May, pp. 1031-1033.

88. THÜROFF, J., BAZEED, M. A., SCHMIDT, R. A., WIGGIN, D. M., TANAGHO, E. A., 1982, "Functional Pattern of Sacral Root Stimulation in Dogs : II. Urethral Closure", Journal of Urology, Vol 127, May, pp. 1034-1036.

89. TIMM, G. W., BRADLEY, W. E., 1971, "Electromechanical Restoration of the Micturition Reflex", IEEE Transaction on Bio-Medical Engineering, Vol. BME-18, No. 4, July, pp. 274-280.

90. TROYK, P. R., SCHWAN, M. A. K., 1992, "Closed-Loop Class E Transcutaneous Power and Data Link for Microimplants", IEEE Transactions on Biomedical Engineering, vol. 39, no. 6, pp. 589-599.

91. TU, L. M., SHAKER, H.S., ROBIN, S., ARABI, K., HASSOUNA, M., SAWAN, M., ELHILALI, M. M., 1997, "Reduction of bladder outlet resistance by selective sacral root stimulation using high-frequency blockade in dogs: An acute study", Submitted to the Journal of Urology.

92. TYLER, D. J., DURAND, D. M., 1995, "Combined Modulation of Pulse Width and Pulse Amplitude to Enhance Functional Selectivity of Neural Stimulation", Conférence EMBS 1995, Article No. 821.

93. W.MAYR, M.BIJAK, 1996, "Functional electrostimulation via implants: applications, limitations, perspectives", Interdisciplinary Aspects on Computers Helping People with Special Needs, vol. 1, pp. 311-317, July.

94. ZHANG, J., KEARNEY, R. E., KIRULUTA, G., ELHILALI, M. M., HUNTER, I., 1994, "System Identification of Bladder Hydrodynamics", Conférence EMBS 1994, Article No. 582.

# ANNEXE I : SCHÉMAS ET CODE DES CONTRÔLEURS EXTERNES

## Code VHDL du FPGA dans le controleur portatif :

```
-----------------------------------------------------------------------
--                                                                   --
-- Description VHDL comportementale de la machine a etats            --
-- servant au controleur externe simplifie d'implant miniaturise.    --
-- Cette machine a etats sera programmee dans un FPGA d'Actel.       --
--                                                                   --
-- Simon Robin, 50901                                                --
-- ecole Polytechnique de Montreal                                   --
-- Section electronique                                              --
--                                                                   --
-- Version 2.0                                                       --
-- Dernieres corrections apportees le 7 juin 1996                    --
--                                                                   --
-----------------------------------------------------------------------
--                                                                   --
-- Description des signaux d'entree-sortie:                          --
--                                                                   --
--    add_eprom: adresse a l'entree de l'eprom                       --
--    out_eprom: bus de donnees a la sortie de l'eprom               --
--    DB: bus de donnees entre le FPGA et l'ecran LCD                --
--    ready: signal qui indique que le LCD est pret                  --
--    RS: selectionne entre donnees et instructions vers l'ecran     --
--    RW: selectionne entre lecture et ecriture sur DB               --
--    E: Enable (ecran)                                              --
--    right: lorsque la fleche droite est actionnee (acceptation)    --
--    up:    lorsque la fleche du haut est actionnee                 --
--    down:  lorsque la fleche du bas est actionnee                  --
--    out_mod: bit de donnee envoye au modulateur                    --
--    por: Power-on reset                                            --
--                                                                   --
-----------------------------------------------------------------------

entity vhdl_a is
   port(signal add_eprom: inout vlbit_1d(11 downto 0);
        signal out_eprom: in    vlbit_1d(7 downto 0);
        signal db:        out   vlbit_1d(7 downto 0);
        signal ready:     in    vlbit;
        signal rs:        out   vlbit;
        signal rw:        out   vlbit;
        signal E:         out   vlbit;
        signal right:     in    vlbit;
        signal up:        in    vlbit;
        signal down:      in    vlbit;
        signal out_mod:   out   vlbit;
        signal clk:       in    vlbit;
        signal por:       in    vlbit
        );
end vhdl_a;

architecture behaviour of vhdl_a is

   begin

   state_machine:process
       type state_type is(init0,
                          init1,
                          init2,
                          init3,
                          init31,
                          init4,
                          init5,
                          init6,
                          init61,
                          init7,
                          main,
                          disp1,
                          disp2,
                          disp3,
                          disp31,
                          disp4,
```

```
                              disp5,
                              disp6,
                              disp7,
                              disp8,
                              disp81,
                              disp9,
                              mod2,
                              mod3,
                              mod4,
                              mod5,
                              mod6,
                              mod7,
                              mod8,
                              mod9);

        variable state: state_type;                -- defini etat suivant
        variable add_prog: vlbit_ld(3 downto 0);   -- defini le prog. en cours

        begin

           wait until prising(clk);

           if por = '0' then -- Initialisation de la machine a etats
              RW <= '1';
              RS <= '0';
              E  <= '0';
              state := init0;
              add_prog := "0000";

----------------------------
-- etats d'initialisation --
----------------------------
           elsif ((state = init0) and (por = '1')) then
              E <= '1';
              add_eprom <= "000000000000";
              state := init1;

           elsif state = init1 then
              E <= '0';
              if ready = '0' then -- Verifie si l'ecran a termine l'init.
                 state := init2;
                 DB <= "00111000"; -- Reset de l'ecran
                 RW <= '0';
              else
                 state := init0;
              end if;

           elsif state = init2 then
              E <= '1';
              state := init3;

           elsif state = init3 then
              E <= '0';
              RW <= '1';
              state := init31;

           elsif state = init31 then
              E <= '1';
              state := init4;

           elsif state = init4 then
              E <= '0';
              if ready = '0' then
                 state := init5;
                 DB <= "00001110"; -- Met l'ecran en mode de 4 lignes
                 RW <= '0';
              else
                 state := init31;
              end if;

           elsif state = init5 then
              E <= '1';
              state := init6;

           elsif state = init6 then
              E <= '0';
              RW <= '1';
```

```
            state := init61;

         elsif state = init61 then
            E <= '1';
            state := init7;

         elsif state = init7 then
            E <= '0';
            if ready = '0' then
               RW <= '0';
               state := disp1;
            else
               state := init61;
            end if;

-----------------------------------------------
-- etats d'affichage du menu a l'ecran LCD --
-----------------------------------------------
         elsif state = disp1 then
            RS <= '0';
            RW <= '0';
            DB <= "00000001"; -- Effacer l'ecran
            state := disp2;

         elsif state = disp2 then
            E <= '1';
            state := disp3;

         elsif state = disp3 then
            E <= '0';
            RS <= '0';
            RW <= '1';
            state := disp31;

         elsif state = disp31 then
            E <= '1';
            state := disp4;

         elsif state = disp4 then
            E <= '0';
            if ready = '0' then
               state := disp5;
            else
               state := disp31;
            end if;

         elsif state = disp5 then
            RS <= '1';
            RW <= '0';
            DB <= out_eprom;
            state := disp6;

         elsif state = disp6 then
            E <= '1';
            add_eprom <= addum(add_eprom(10 downto 0),"00000000001");
            state := disp7;

         elsif state = disp7 then
            E <= '0';
            state := disp8;

         elsif state = disp8 then
            RS <= '0';
            RW <= '1';
            state := disp81;

         elsif state = disp81 then
            E <= '1';
            state := disp9;

         elsif state = disp9 then
            E <= '0';
            if ready = '0' then
               if out_eprom = X"FF" then -- Affichage termine
                  state := main;
               else
                  state := disp5;
```

```
                end if;
            else
                state := disp81;
            end if;

-----------------------------------------------------
-- etats d'ecriture de donnees vers le modulateur --
-- Ces etats lisent 3 fois 8 bits paralleles et   --
-- les envoient de fa¤on serielle                 --
-----------------------------------------------------
        elsif state = mod2 then
          if out_eprom = X"FF" then -- Fin de la sequence a envoyer
             state := main;
          else
             out_mod <= out_eprom(0);
             state := mod3;
          end if;

        elsif state = mod3 then
           out_mod <= out_eprom(1);
           state := mod4;

        elsif state = mod4 then
           out_mod <= out_eprom(2);
           state := mod5;

        elsif state = mod5 then
           out_mod <= out_eprom(3);
           state := mod6;

        elsif state = mod6 then
           out_mod <= out_eprom(4);
           state := mod7;
        elsif state = mod7 then
           out_mod <= out_eprom(5);
           state := mod8;
        elsif state = mod8 then
           out_mod <= out_eprom(6);
           state := mod9;
        elsif state = mod9 then
           out_mod <= out_eprom(7);
           add_eprom <= addum(add_eprom(10 downto 0),"00000000001");
           state := mod2;
--------------------
-- etat principal --
--------------------
        elsif state = main then
           if up = '1' then  -- changer de programme vers le haut
              add_prog := subum(add_prog(2 downto 0),"001");
              add_eprom <= "00" & add_prog(2 downto 0) & "0000000";
              state := disp1;
           elsif down = '1' then  -- changer de programme vers le bas
              add_prog := addum(add_prog(2 downto 0),"001");
              add_eprom <= "00" & add_prog(2 downto 0) & "0000000";
              state := disp1;
           elsif right = '1' then  -- actionner un programme
              add_eprom <= "01" & add_prog(2 downto 0) & "0000000";
              state := mod2;
           end if;

        else -- pour tous les etats non-definis
           state := init0;
        end if;

     end process state_machine;

  end behaviour;
-------------------------------------------------------------------------
```

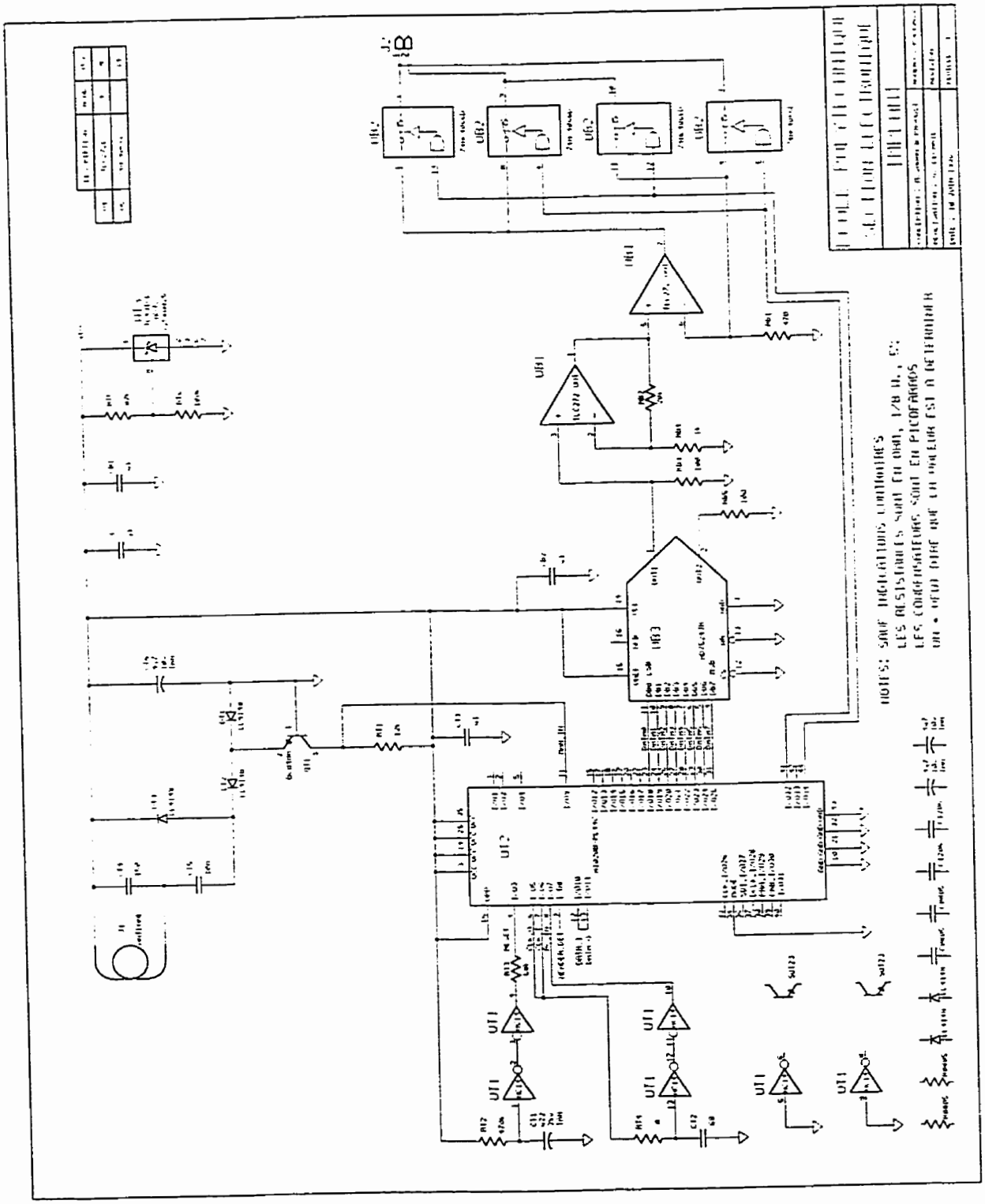


**Schéma électronique du contrôleur portatif.**

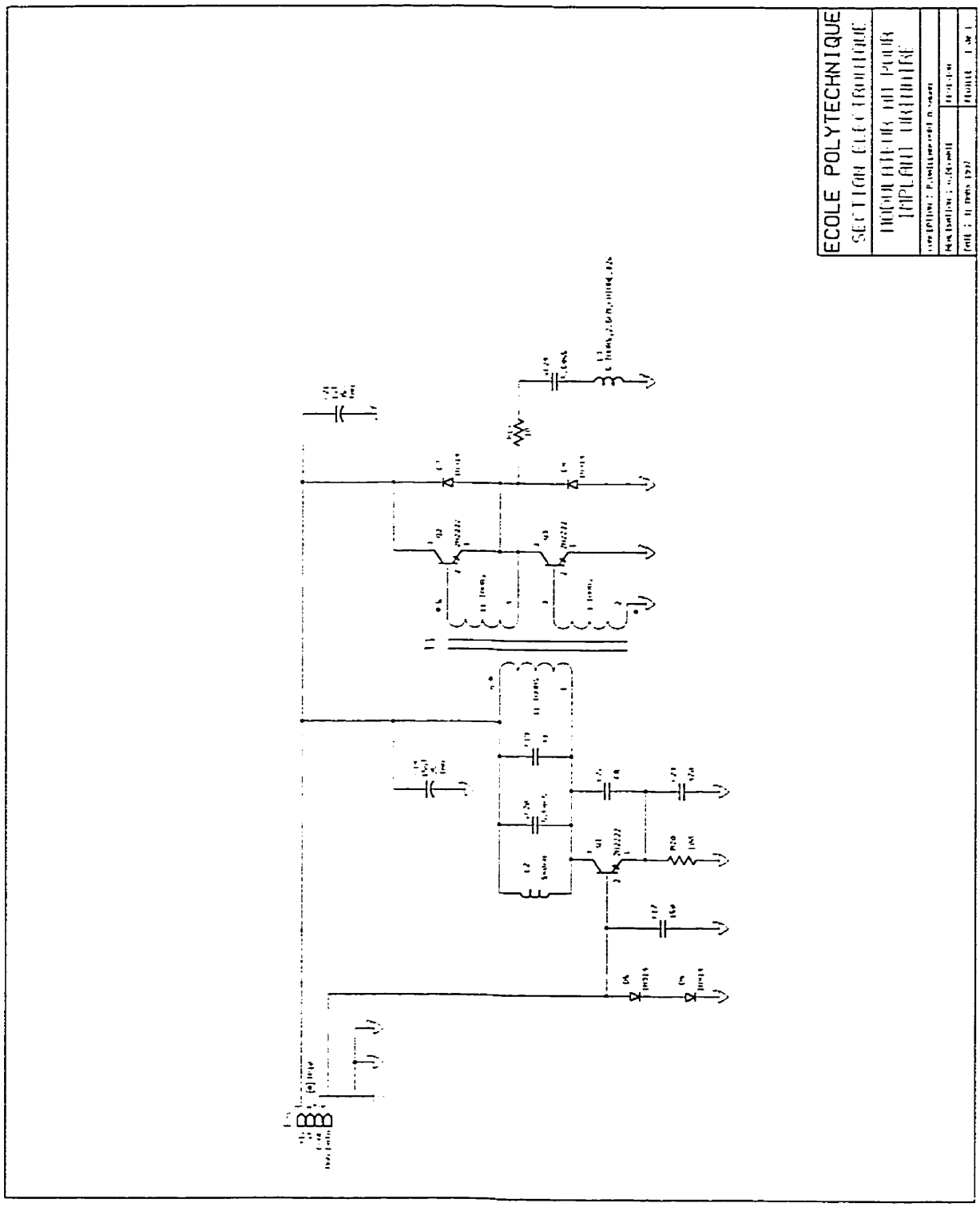
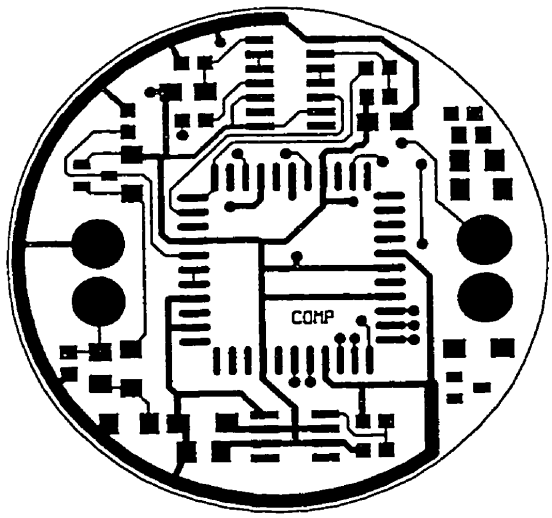
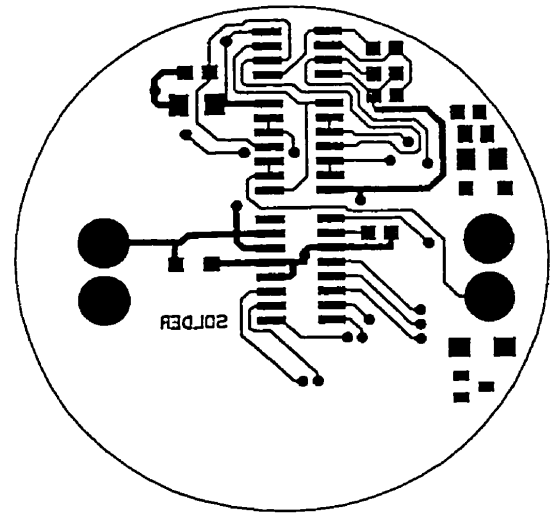


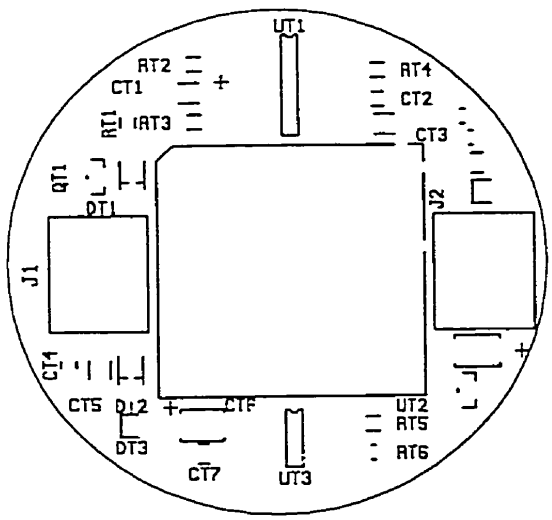

**Schéma électronique du transmetteur AM.**



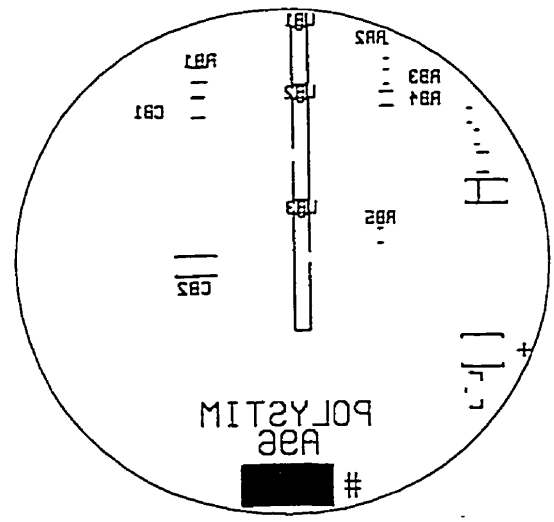

COMPONENT SIDE




SOLDER SIDE




SILKTOP




SILKBOTTOM

**Dessins des masques pour le circuit imprimé du contrôleur portatif.**

# ANNEXE II : SCHÉMAS DE L'IMPLANT SUR CIRCUIT IMPRIMÉ

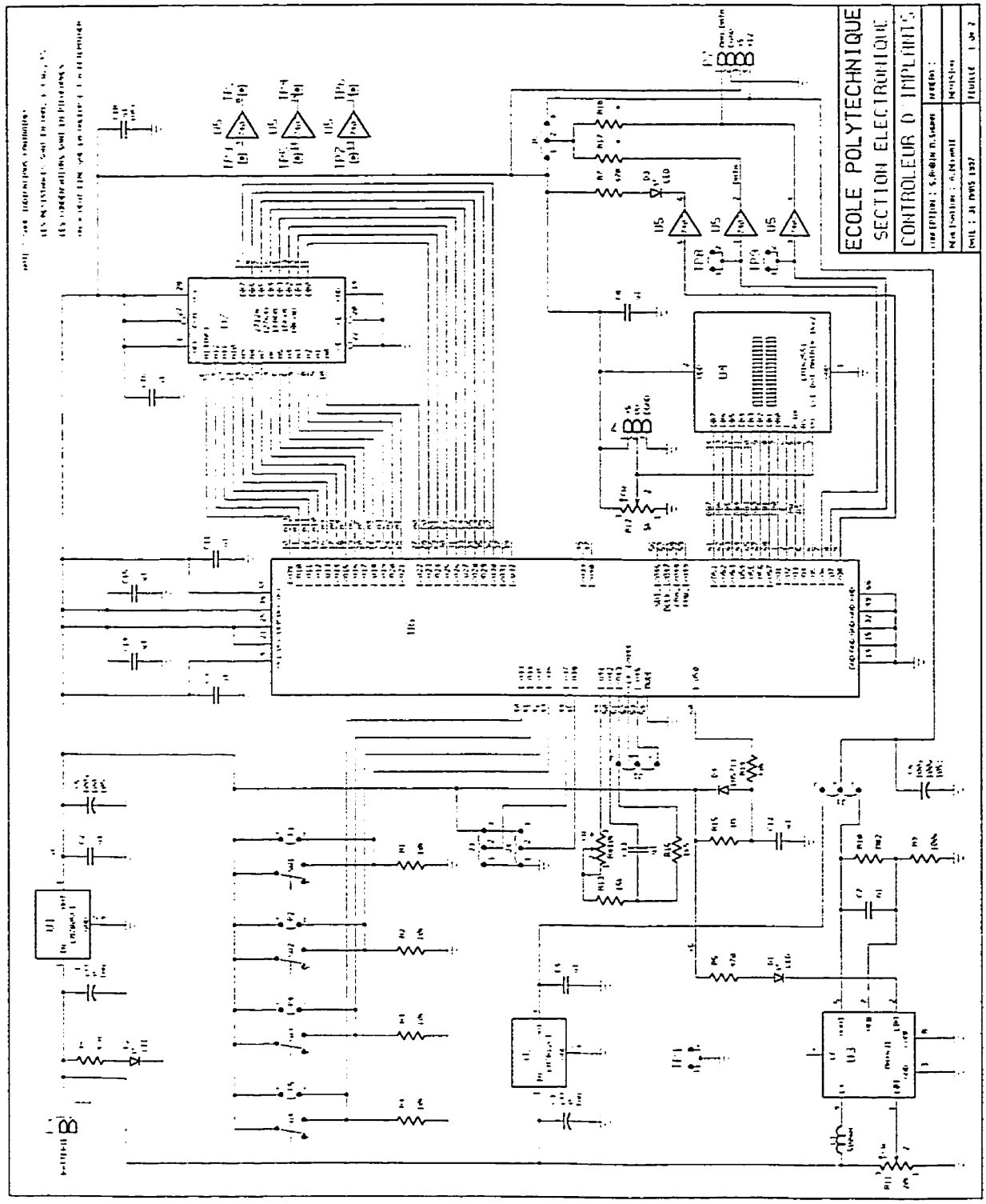
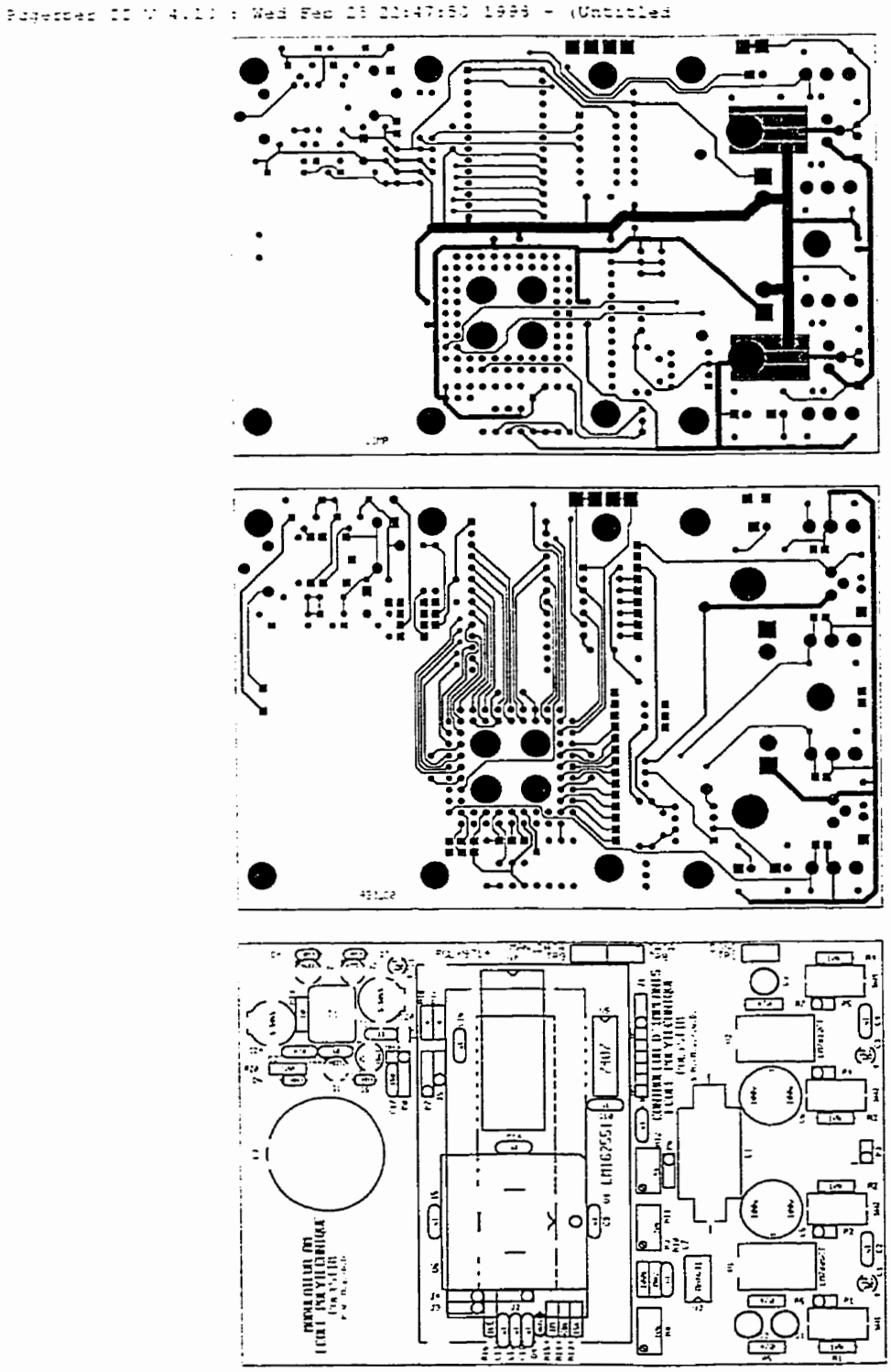



**Schéma électronique de l'implant.**



**Dessins des masques pour le circuit imprimé de l'implant.**

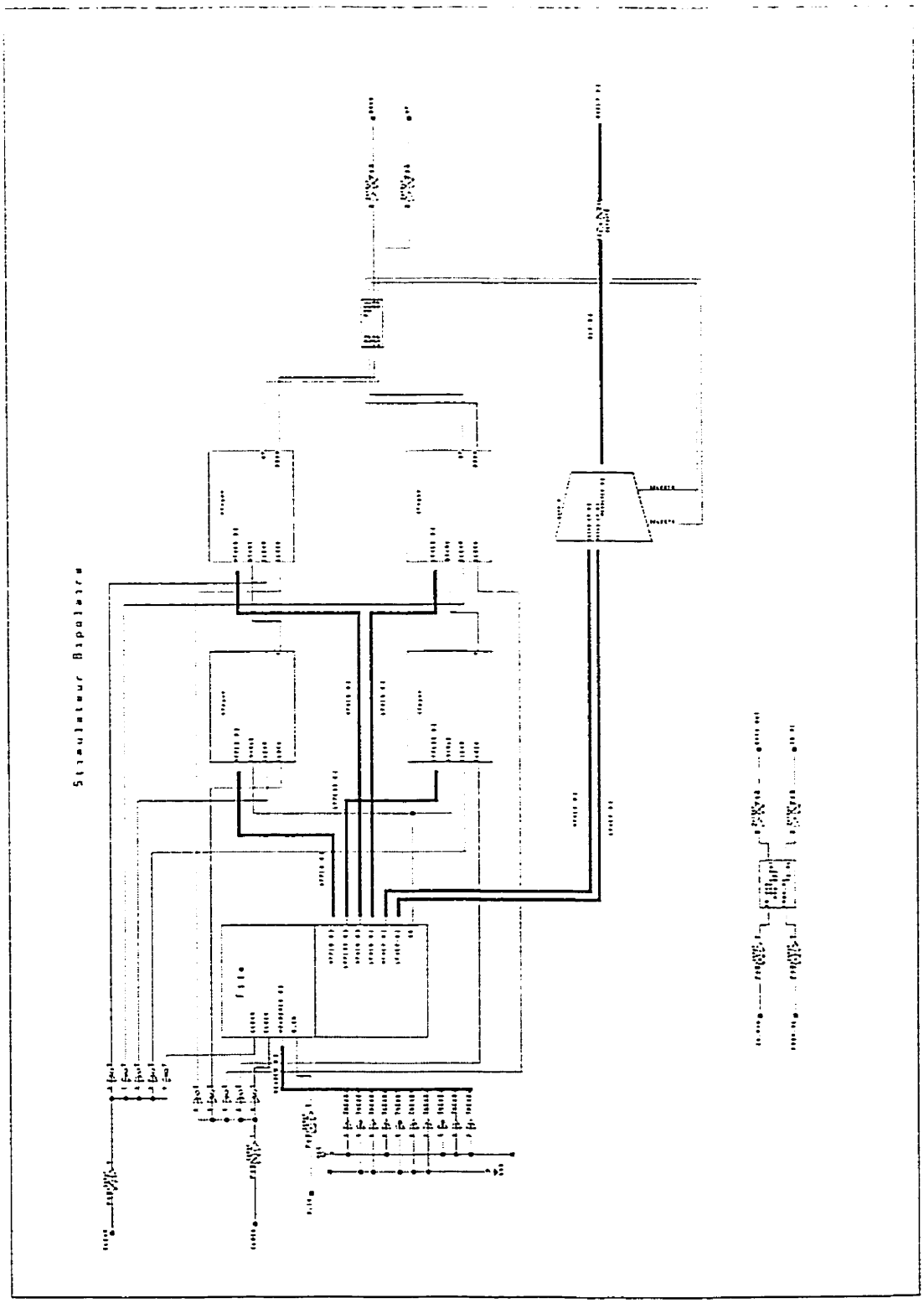



**Schéma électronique du « top-level» du FPGA de l'implant.**

Decodeur Manchester

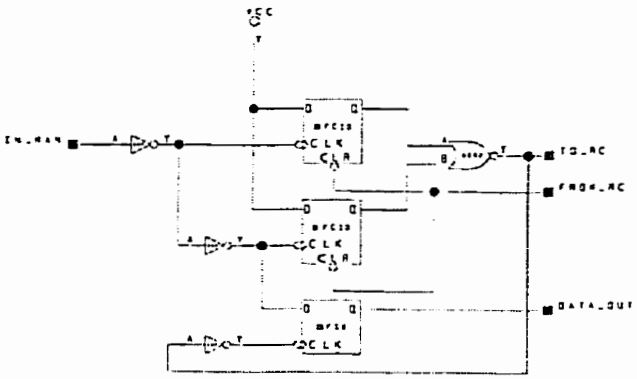



**Schéma électronique du décodeur Manchester.**

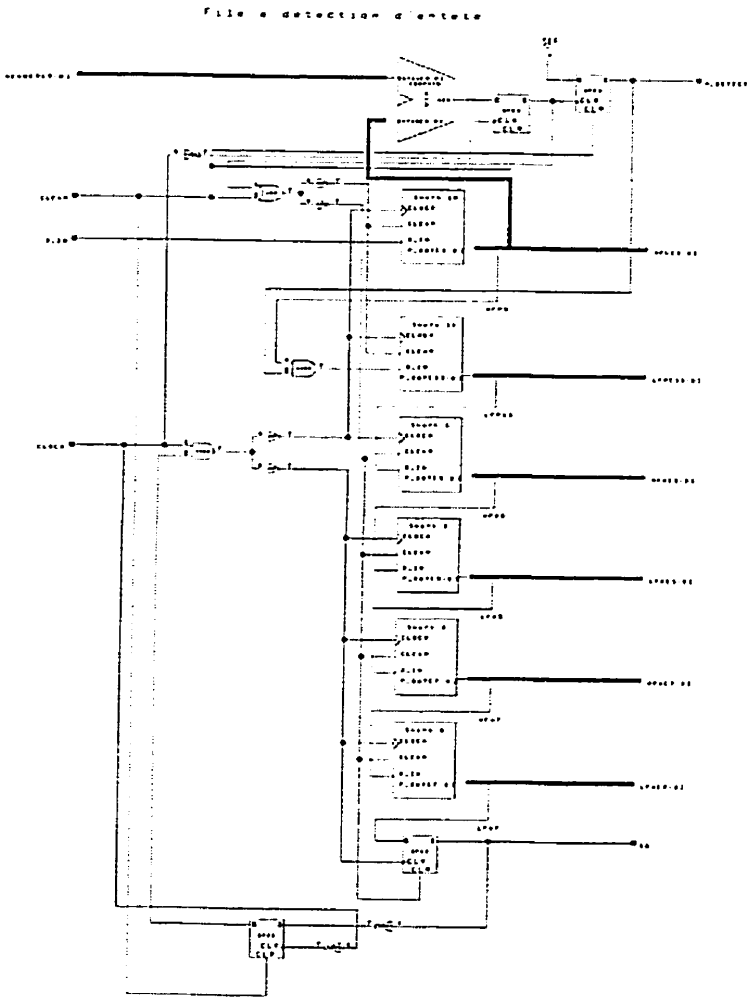


**Schéma de la file à détection d'entête.**

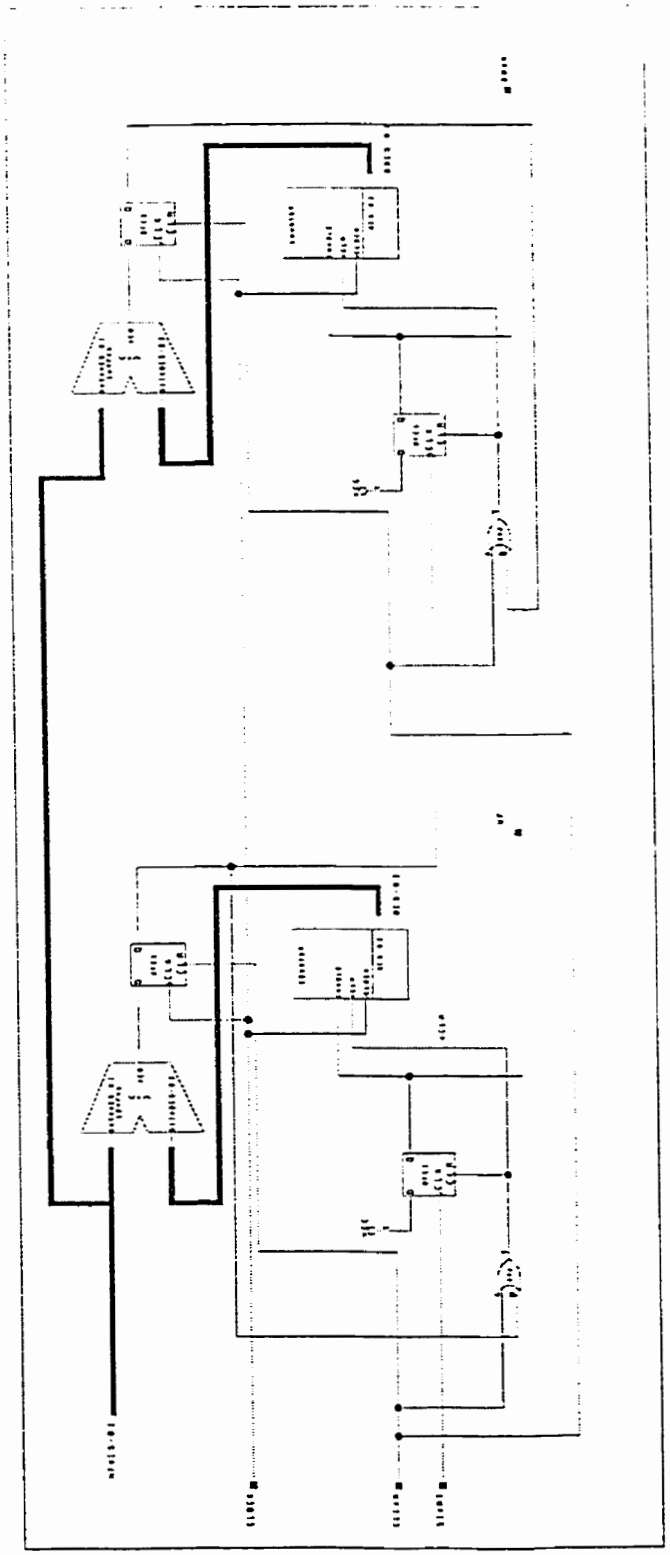


**Schéma du générateur de fréquence.**

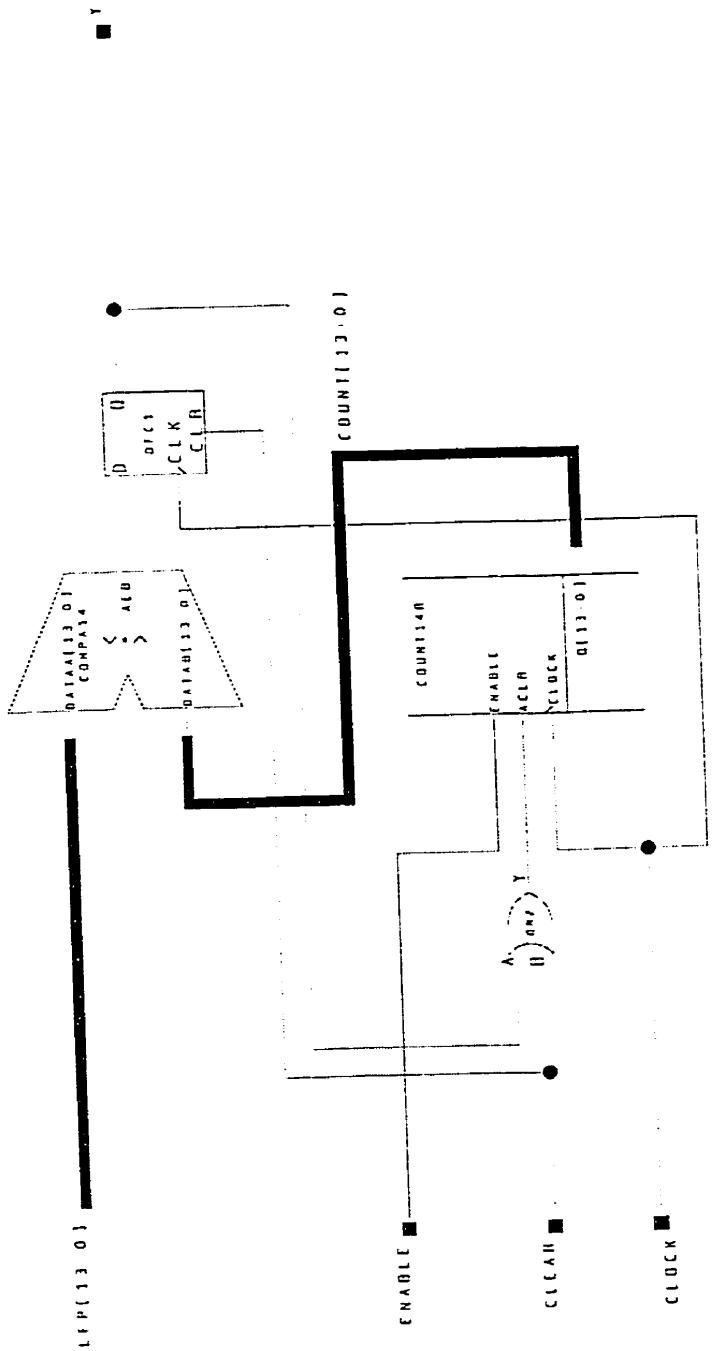



**Schéma du générateur d'impulsion.**

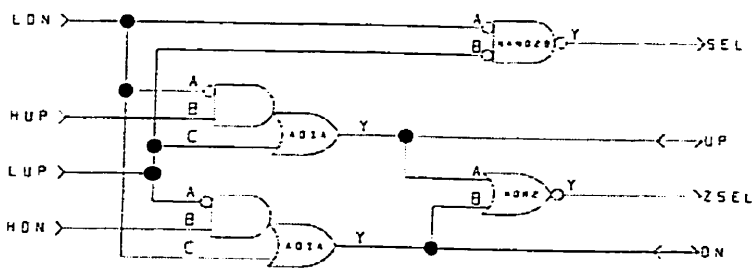



**Schéma de la logique de contrôle de la sortie.**

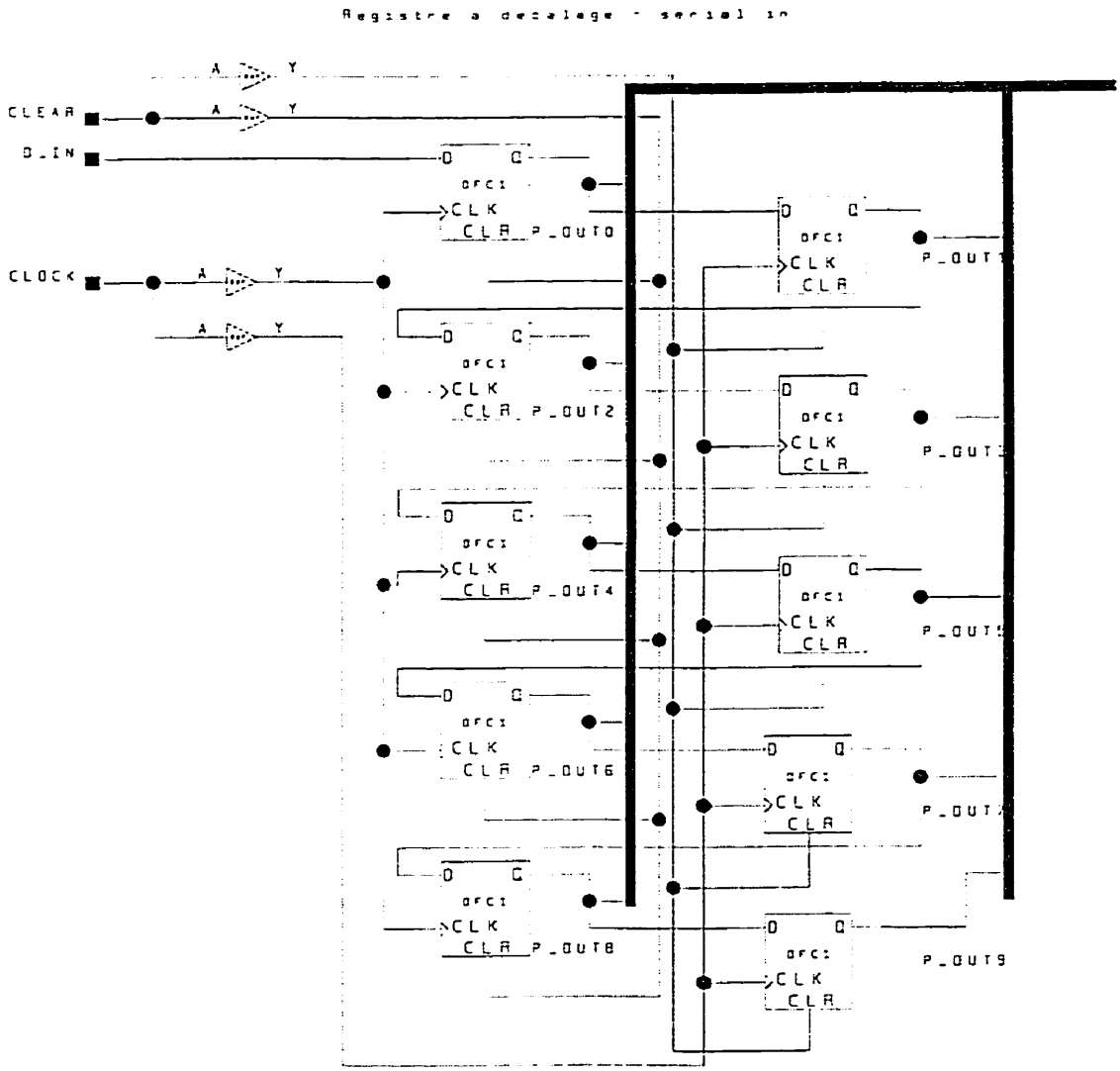
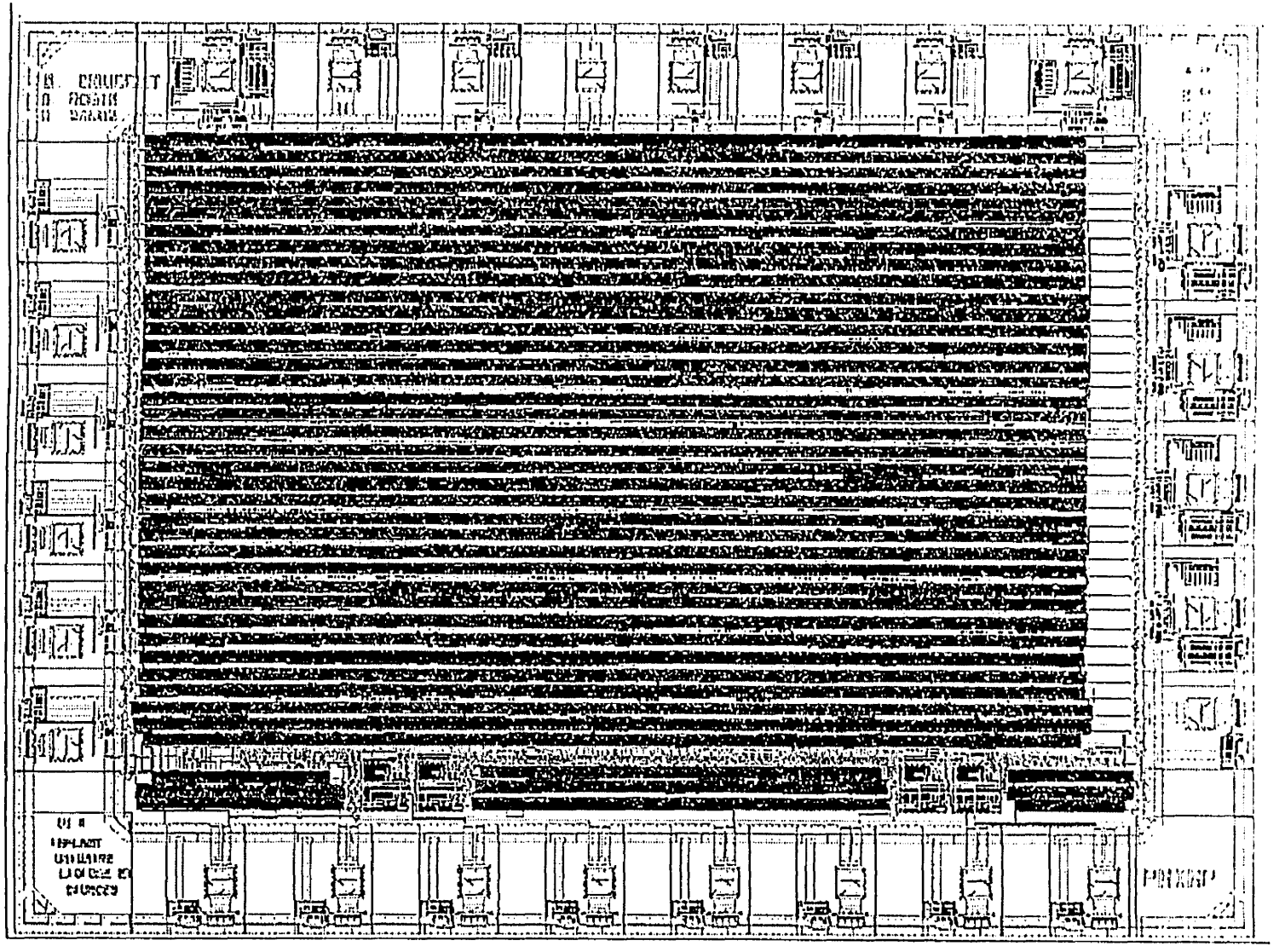



**Schéma du registre à décalage « serial-in, parallel-out ».**

# ANNEXE III : DESSIN DE MASQUES, CODE ET RÉSULTATS DE SYNTHÈSE DE L'IMPLANT INTÉGRÉ



```
-----------------------------------------------------------------------------
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- Equipe de Recherche en Neurotechnologies
--
-- ELE6305 Conception de circuit electroniques integres II
-- Realisation d'un micro-stimulateur implantable totalement integre
-- Session hiver 1997
--
-- Auteur: Simon Robin
-
-- Module de jonction de tous les modules du projet
--
-- Fichier: stim.vhdl
-----------------------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;
use ieee.std_logic_arith.all;

entity stim is
  port(clk, entree, por, ddi  : in std_logic;
       EN, SIGN : out std_logic;
       AMP : out std_logic_vector(4 downto 0));
end stim;


architecture struct of stim is

component comm
  port( clk, entree, por,ddi   : in std_logic;
        testHeaderDetect, go   : out std_logic;
        LFP, HFP               : out std_logic_vector(15 downto 0);
        LFW, LFI, HFW, HFI     : out std_logic_vector(7 downto 0);
        LFA, HFA               : out std_logic_vector(4 downto 0));
  end component;

component uc
  port(CLK, GO, TEST : in std_logic;
       LFP, HFP : in std_logic_vector(15 downto 0);
       LFW, LFI, HFW, HFI : in std_logic_vector(7 downto 0);
       LFA, HFA : in std_logic_vector(4 downto 0);
       EN, SIGN : out std_logic;
       AMP : out std_logic_vector(4 downto 0));
  end component;

  signal test, go : std_logic;
  signal LFP, HFP : std_logic_vector(15 downto 0);
  signal LFW, LFI, HFW, HFI : std_logic_vector(7 downto 0);
  signal LFA, HFA : std_logic_vector(4 downto 0);

begin
  uc_module : uc port map(clk, go, test, lfp, hfp, lfw,
                          lfi, hfw, hfi, lfa, hfa, en, sign, amp);
  comm_module : comm port map(clk, entree, por, ddi, test, go, lfp, hfp,
                              lfw, lfi, hfw, hfi, lfa, hfa);
end struct;
```

```
-----------------------------------------------------------------------------
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- Equipe de Recherche en Neurotechnologies
--
-- ELE6305 Conception de circuit electroniques integres II
-- Realisation d'un micro-stimulateur implantable totalement integre
-- Session hiver 1997
--
-- Auteurs: Pierre Vaillancourt, Simon Robin, Alexandre Beauchamp-Parent
--
-- Module de jonction de tous les modules de communication
--
-- Fichier: comm.vhdl
-----------------------------------------------------------------------------
--  Date       Modification
--
--  97-03-01   Creation sans le module de detection d'erreur BCH(31,16)
--             L'ajouter sous forme de fonction ou de component??
--  97-04-12   Modifications pour ajuster les i/o avec les modules de
--             generations des signaux de stimulation
--
-----------------------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;
use ieee.std_logic_arith.all;


entity comm is
  port( clk, entree, por,ddi   : in std_logic;

        testHeaderDetect, go   : out std_logic;
        LFP, HFP               : out std_logic_vector(15 downto 0);
        LFW, LFI, HFW, HFI     : out std_logic_vector(7 downto 0);
        LFA, HFA               : out std_logic_vector(4 downto 0)
  );
end comm;

architecture struct of comm is

        component Header
        port(  entree                  : in    std_logic;
               clk                     : in    std_logic;
               por                     : in    std_logic;
               sortie                  : out   std_logic;
               stimulationHeaderDetect         : out   std_logic;
               testHeaderDetect        : out   std_logic;
               ddiHeaderDetect                 : out   std_logic);
        end component;

        component sequence
        port(  entree : in    std_logic;
               entete : in    std_logic;
               sample : in    std_logic;
               clk    : in    std_logic;
               por    : in    std_logic;

               isValid : out  std_logic;

               block0 : out   std_logic_vector(15 downto 0);
               block1 : out   std_logic_vector(15 downto 0);
               block2 : out   std_logic_vector(15 downto 0);
               block3 : out   std_logic_vector(15 downto 0);
               block4 : out   std_logic_vector(15 downto 0);
               CRC0   : out   std_logic_vector(15 downto 0);
               CRC1   : out   std_logic_vector(15 downto 0);
               CRC2   : out   std_logic_vector(15 downto 0);
               CRC3   : out   std_logic_vector(15 downto 0);
               CRC4   : out   std_logic_vector(15 downto 0));
        end component;

        component Detect
        port(  clk    : in    std_logic;
               por    : in    std_logic;

               isValid : in   std_logic;
```

```
            block0  : in    std_logic_vector(15 downto 0);
            block1  : in    std_logic_vector(15 downto 0);
            block2  : in    std_logic_vector(15 downto 0);
            block3  : in    std_logic_vector(15 downto 0);
            block4  : in    std_logic_vector(15 downto 0);
            CRC0    : in    std_logic_vector(15 downto 0);
            CRC1    : in    std_logic_vector(15 downto 0);
            CRC2    : in    std_logic_vector(15 downto 0);
            CRC3    : in    std_logic_vector(15 downto 0);
            CRC4    : in    std_logic_vector(15 downto 0);

            sample  : out   std_logic;
            go      : out   std_logic;

            HFA     : out   std_logic_vector(4  downto 0);
            HFP     : out   std_logic_vector(15 downto 0);
            HFW     : out   std_logic_vector(7  downto 0);
            HFI     : out   std_logic_vector(7  downto 0);
            LFA     : out   std_logic_vector(4  downto 0);
            LFP     : out   std_logic_vector(15 downto 0);
            LFW     : out   std_logic_vector(7  downto 0);
            LFI     : out   std_logic_vector(7  downto 0));
    end component;

    component seq_ddi
    port(   entree  : in    std_logic;
            entete  : in    std_logic;
            ddi     : in    std_logic;
            clk     : in    std_logic;
            por     : in    std_logic;

            go      : out   std_logic;

            HFA     : out   std_logic_vector(4  downto 0);
            HFP     : out   std_logic_vector(15 downto 0);
            HFW     : out   std_logic_vector(7  downto 0);
            HFI     : out   std_logic_vector(7  downto 0);
            LFA     : out   std_logic_vector(4  downto 0);
            LFP     : out   std_logic_vector(15 downto 0);
            LFW     : out   std_logic_vector(7  downto 0);
            LFI     : out   std_logic_vector(7  downto 0));
    end component;

    signal entete, sortie, go1, go2,tmp          : std_logic;
    signal sample, isValid,ddiHeader             : std_logic;
    signal testMode, testModeRequest             : std_logic;
    signal block0, block1, block2, block3, block4       : std_logic_vector(15 downto
0);
    signal CRC0, CRC1, CRC2, CRC3, CRC4          : std_logic_vector(15 downto 0);

    signal LFA1, LFA2, HFA1, HFA2                : std_logic_vector(4  downto 0);
    signal LFP1, LFP2, HFP1, HFP2                : std_logic_vector(15 downto 0);
    signal LFW1, LFW2, HFW1, HFW2                : std_logic_vector(7  downto 0);
    signal LFI1, LFI2, HFI1, HFI2                : std_logic_vector(7  downto 0);

begin
    Header_Module  : Header port map(entree, clk, por, sortie, entete,
                                     testModeRequest, ddiHeader);

    Sequence_Module        : sequence port map(  sortie, entete, sample, clk, por,
                                     isValid,
                                     block0, block1, block2, block3, block4,
                                     CRC0, CRC1, CRC2, CRC3, CRC4);

    Detect_Module  : Detect port map(    clk, por, isValid,
                                     block0, block1, block2, block3, block4,
                                     CRC0, CRC1, CRC2, CRC3, CRC4,
                                     sample, go1,
                                     HFA1, HFP1, HFW1, HFI1,
                                     LFA1, LFP1, LFW1, LFI1);

    DDI_Module     : seq_ddi port map(   sortie, ddiHeader, ddi, clk, por, go2,
                                 HFA2, HFP2, HFW2, HFI2, LFA2, LFP2, LFW2, LFI2);
    process
    begin
        wait until clk'event and clk='1';
```

```
                if(por='0') then
                        go <= '0';
                        testHeaderDetect <= '0';
                        testMode <= '0';
                else
                        -- attention aux detections du mode test causees par des erreurs de
                        -- transmission ou dues a des mauvais formattage de la sequence
binaire
                        -- il serait preferable de faire un module de detection d'erreur
sur
                        -- les bits composant l'entete du test (modules semblables a ce qui
est
                        -- fait pour la detection d'erreur de la sequence de stimulation.

                        testHeaderDetect <= testMode;

                        if(testModeRequest = '1' and go1 = '0' and go2 = '0') then
                                testMode <= '1';
                                go <= '1';
                        end if;
                        if(go1='1' and ddi='0') then
                                HFA <= HFA1; HFP <= HFP1; HFW <= HFW1; HFI <= HFI1;
                                LFA <= LFA1; LFP <= LFP1; LFW <= LFW1; LFI <= LFI1;
                                if(testMode = '1') then
                                        go <= '0';
                                        testMode <= '0';
                                else
                                        go <= '1';
                                        testMode <= '0';
                                end if;
                        end if;
                        if(go2='1' and ddi='1') then
                                HFA <= HFA2; HFP <= HFP2; HFW <= HFW2; HFI <= HFI2;
                                LFA <= LFA2; LFP <= LFP2; LFW <= LFW2; LFI <= LFI2;
                                if(testMode = '1') then
                                        go <= '0';
                                        testMode <= '0';
                                else
                                        go <= '1';
                                        testMode <= '0';
                                end if;
                        end if;
                end if;
        end process;
end struct;
detect.vhdl
```

```
-----------------------------------------------------------------------------
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- Equipe de Recherche en Neurotechnologies
--
-- ELE6305 Conception de circuit electroniques integres II
-- Realisation d'un micro-stimulateur implantable totalement integre
-- Session hiver 1997
--
-- Auteurs: Pierre Vaillancourt, Simon Robin, Alexandre Beauchamp-Parent
--
-- Module de controle de la detection des erreurs
--     Ce module se charge d'envoyer les mots et leur CRC vers le module
--     de detection des erreurs. Il recoit le verdict de la detection
--     et enregistre les donnees valides dans les registres de sorties.
--     En cas d'erreur, il demande un nouvel echantillon.
--
-- Fichier: detect.vhdl
-----------------------------------------------------------------------------
--  Date       Modification
--
--  97-03-01   Creation sans le module de detection d'erreur BCH(31,16)
--             L'ajouter sous forme de fonction ou de component??
--  97-03-20   Modification pour le traitement parallele des blocks de
--             donnees
--  97-04-03   Modifications pour la testabilite
--
-----------------------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;
use ieee.std_logic_arith.all;


entity Detect is
        port(   clk     : in    std_logic;
                por     : in    std_logic;

                isValid : in    std_logic;
                block0  : in    std_logic_vector(15 downto 0);
                block1  : in    std_logic_vector(15 downto 0);
                block2  : in    std_logic_vector(15 downto 0);
                block3  : in    std_logic_vector(15 downto 0);
                block4  : in    std_logic_vector(15 downto 0);
                CRC0    : in    std_logic_vector(15 downto 0);
                CRC1    : in    std_logic_vector(15 downto 0);
                CRC2    : in    std_logic_vector(15 downto 0);
                CRC3    : in    std_logic_vector(15 downto 0);
                CRC4    : in    std_logic_vector(15 downto 0);

                sample  : out   std_logic;
                go      : out   std_logic;

                HFA     : out   std_logic_vector(4  downto 0);
                HFP     : out   std_logic_vector(15 downto 0);
                HFW     : out   std_logic_vector(7  downto 0);
                HFI     : out   std_logic_vector(7  downto 0);
                LFA     : out   std_logic_vector(4  downto 0);
                LFP     : out   std_logic_vector(15 downto 0);
                LFW     : out   std_logic_vector(7  downto 0);
                LFI     : out   std_logic_vector(7  downto 0)
                );
end Detect;

architecture Stimulation of Detect is
        component BCH_D
                port(   clk     : in    std_logic;
                        por     : in    std_logic;

                        detect  : in    std_logic;
                        data    : in    std_logic_vector(15 downto 0);
                        CRC     : in    std_logic_vector(15 downto 0);

                        OK      : out   std_logic;
                        finish  : out   std_logic
                );
        end component;
```

```
        type Etat is (ATTENTE, DECODE, LANCEMENT);

        signal detect          :       std_logic;
        signal fin_A, fin_B    :       std_logic;
        signal fin_C, fin_D    :       std_logic;
        signal fin_E           :       std_logic;
        signal OK_A, OK_B, OK_C :      std_logic;
        signal OK_D, OK_E      :       std_logic;

begin
        detecteurA : BCH_D port map(
                                clk     => clk,
                                por     => por,
                                detect  => detect,
                                data    =>  block0,
                                CRC     => CRC0,
                                OK      => OK_A,
                                finish  => fin_A);
        detecteurB : BCH_D port map(
                                clk     => clk,
                                por     => por,
                                detect  => detect,
                                data    =>  block1,
                                CRC     => CRC1,
                                OK      => OK_B,
                                finish  => fin_B);
        detecteurC : BCH_D port map(
                                clk     => clk,
                                por     => por,
                                detect  => detect,
                                data    =>  block2,
                                CRC     => CRC2,
                                OK      => OK_C,
                                finish  => fin_C);
        detecteurD : BCH_D port map(
                                clk     => clk,
                                por     => por,
                                detect  => detect,
                                data    =>  block3,
                                CRC     => CRC3,
                                OK      => OK_D,
                                finish  => fin_D);
        detecteurE : BCH_D port map(
                                clk     => clk,
                                por     => por,
                                detect  => detect,
                                data    =>  block4,
                                CRC     => CRC4,
                                OK      => OK_E,
                                finish  => fin_E);

        FSM : process
                variable etat  :       Etat;
        begin
                wait until clk'event and clk='1';

                if(por = '0') then
                        etat    := ATTENTE;
                        go              <= '0';
                        sample          <= '1';
                        detect          <= '0';
                else
                        case etat is
                        when ATTENTE =>
                                go      <= '0';
                                sample <= '1';
                                detect <= '0';
                                if(isValid = '1') then
                                        etat := DECODE;
                                else
                                        etat := ATTENTE;
                                end if;

                        when DECODE =>
                                go      <= '0';
                                sample <= '0';
```

```
                                detect <= '1';
                                if((fin_A and fin_B and fin_C and fin_D and fin_E)='1') then
                                        if((OK_A and OK_B and OK_C and OK_D and OK_E)='1')
then
                                                etat := LANCEMENT;
                                        else
                                                etat := ATTENTE;
                                        end if;
                                else
                                        etat := DECODE;
                                end if;

                        when LANCEMENT =>
                                go     <= '1';
                                sample <= '0';
                                detect <= '0';
                                HFA     <=      block2(15 downto 11);
                                HFP     <=      block0;
                                HFW     <=      block1(15 downto 8);
                                HFI     <=      block1(7  downto 0);
                                LFA     <=      block2(4  downto 0);
                                LFP     <=      block3;
                                LFW     <=      block4(15 downto 8);
                                LFI     <=      block4(7  downto 0);
                                etat := LANCEMENT;
                        end case;
                end if;
        end process FSM;
end Stimulation;
```

```
-----------------------------------------------------------------------------
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- Equipe de Recherche en Neurotechnologies
--
-- ELE6305 Conception de circuit electroniques integres II
-- Realisation d'un micro-stimulateur implantable totalement integre
-- Session hiver 1997
--
-- Auteurs: Alexandre Beauchamp-Parent, Simon Robin, Pierre Vaillancourt
--
-- Module de generation des signaux de commande des impulsions bipolaire
--
-- Fichier: dsp.vhdl
--
-----------------------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;
use ieee.std_logic_arith.all;


entity dsp is
  port(CLK, GO, SYNC : in std_logic;
       FW, FI : in std_logic_vector(7 downto 0);
       FA : in std_logic_vector(4 downto 0);
       EN, SIGN : out std_logic;
       AMP : out std_logic_vector(4 downto 0));
end dsp;


architecture algo of dsp is
begin
  process
  variable COMPTEUR2 : std_logic_vector(7 downto 0);
  variable COMPTEUR3 : std_logic_vector(7 downto 0);
  variable COMPTEUR4 : std_logic_vector(7 downto 0);
  variable ETAT : std_logic_vector(1 downto 0);
  begin
    wait until CLK'event and CLK='1';
    if GO='0' then
      EN <='0';
      SIGN <='-';
      AMP <="00000";
    else
      if SYNC='1' then
        COMPTEUR2 :="00000000";
        if COMPTEUR2=FW then
          ETAT :="00";
        else
          ETAT :="01";
        end if;
      end if;
      case ETAT is
        when "00" => EN <='0';
                     SIGN <='-';
                     AMP <="00000";
        when "01" => EN <='1';
                     SIGN <='1';
                     AMP <=FA;
                     COMPTEUR2 :=COMPTEUR2+'1';
                     if COMPTEUR2=FW then
                       COMPTEUR3 :="00000000";
                       COMPTEUR4 :="00000000";
                       if COMPTEUR3=FI then
                         ETAT :="11";
                       else
                         ETAT :="10";
                       end if;
                     end if;
        when "10" => EN <='0';
                     SIGN <='-';
                     AMP <="00000";
                     COMPTEUR3 :=COMPTEUR3+'1';
                     if COMPTEUR3=FI then
                       ETAT :="11";
                     end if;
```

```
        when "11" => EN <='1';
                     SIGN <='0';
                     AMP <=FA;
                     COMPTEUR4 :=COMPTEUR4+'1';
                     if COMPTEUR4=FW then
                       ETAT :="00";
                     end if;
        when others => null;
      end case;
    end if;
  end process;

end algo;
dsp_test.vhdl
```

```
-----------------------------------------------------------------------------
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- Equipe de Recherche en Neurotechnologies
--
-- ELE6305 Conception de circuit electroniques integres II
-- Realisation d'un micro-stimulateur implantable totalement integre
-- Session hiver 1997
--
-- Auteur: Pierre Vaillancourt
--
-- Module de Detection d'Entete
--     Ce module detecte une certaine sequence de bits identifiant
--     le debut du mot de commande de l'implant.  Cette sequence est
--     constante et definie pour une version particuliere de chaque
--     implant.  Le module ne faire que detecter la sequence et envoyer
--     un signal de sortie lorsqu'elle est detectee.
--
-- Fichier: header.vhdl
-----------------------------------------------------------------------------
--  Date       Modification
--
--  97-02-03   Creation
--  97-02-09   Ajout de l'entete de test
--  97-03-24    Ajout de l'entete de DID
--  97-04-03   Modifications pour la testabilite
--
-----------------------------------------------------------------------------
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;

entity Header is
       port(  entree                  : in    std_logic;
              clk                     : in    std_logic;
              por                     : in    std_logic;
              sortie                  : out   std_logic;
              stimulationHeaderDetect         : out   std_logic;
              testHeaderDetect        : out   std_logic;
              ddiHeaderDetect                 : out   std_logic);
end Header;

architecture Compare_entete of Header is
begin
       compare: process
         variable fifo   : std_logic_vector(15 downto 0);
         constant hStim  : std_logic_vector(15 downto 0) := "1111111100000001";
         constant hTest  : std_logic_vector(15 downto 0) := "1111111100000010";
         constant hDID          : std_logic_vector(15 downto 0) := "1111111100000100";
       begin
              wait until clk'event and clk='1';
              if(por='0') then
                     fifo := "0000000000000000";
              end if;

              -- envoi l'entree directement sur la sortie
              sortie <= entree;

              -- fait le shift sur le fifo lsb vers msb et concatenation
              fifo(14 downto 0) := fifo(15 downto 1);
              fifo(15) := entree;

              case fifo is
                     when hStim =>
                            stimulationHeaderDetect        <= '1';
                            testHeaderDetect       <= '0';
                            ddiHeaderDetect        <= '0';
                     when hTest =>
                            testHeaderDetect       <= '1';
                            stimulationHeaderDetect <= '0';
                            ddiHeaderDetect        <= '0';

                     when hDID =>
                            testHeaderDetect       <= '0';
                            stimulationHeaderDetect <= '0';
                            ddiHeaderDetect        <= '1';
```

```
                        when others =>
                                stimulationHeaderDetect <= '0';
                                testHeaderDetect        <= '0';
                                ddiHeaderDetect         <= '0';
                end case;
        end process compare;
end Compare_entete;
```

```
-----------------------------------------------------------------------------
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- Equipe de Recherche en Neurotechnologies
--
-- ELE6305 Conception de circuit electroniques integres II
-- Realisation d'un micro-stimulateur implantable totalement integre
-- Session hiver 1997
--
-- Auteurs: Pierre Vaillancourt, Simon Robin, Alexandre Beauchamp-Parent
--
-- Module d'entree directe de donnees a l'implant
--      Ce module doit trouver les bonnes entetes pour synchroniser la
--      sequence de commande avec les bonnes donnees au cas ou l'entete
--      se retrouverait par erreur dans le flot de donnees arrivantes.
--      Ce type d'erreur peut etre du a du bruit dans la transmission
--      et ne doit pas affecter la reception.
--
-- Fichier: seq_ddi.vhdl
-----------------------------------------------------------------------------
--  Date       Modification
--
--  97-03-24    creation pour l'ajout de test directs au uc
--  97-04-03   Modifications pour la testabilite
--  97-04-11   Standardisation du nombre de bits des registres HFA
--             et LFA de 5 bits a 8 bits (ROM duc controleur)
--
-----------------------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;
use ieee.std_logic_arith.all;

entity seq_ddi is
        port(   entree  : in    std_logic;
                entete  : in    std_logic;
                ddi     : in    std_logic;
                clk     : in    std_logic;
                por     : in    std_logic;

                go      : out   std_logic;

                HFA     : out   std_logic_vector(4  downto 0);
                HFP     : out   std_logic_vector(15 downto 0);
                HFW     : out   std_logic_vector(7  downto 0);
                HFI     : out   std_logic_vector(7  downto 0);
                LFA     : out   std_logic_vector(4  downto 0);
                LFP     : out   std_logic_vector(15 downto 0);
                LFW     : out   std_logic_vector(7  downto 0);
                LFI     : out   std_logic_vector(7  downto 0)
        );
end seq_ddi;

architecture Stimulation of seq_ddi is
        type Etat_Type is (INITIALISATION, ATTENTE, ACCUMULE, VALIDE);
begin
        FSM: process
                variable data           :   std_logic_vector(79 downto 0);
                variable etat         :     Etat_Type;
                variable index          :   integer;

        begin
                wait until clk'event and clk='1';

                if(por='0') then
                        etat := INITIALISATION;
                        go <= '0';
                else
                        case etat is
                        when INITIALISATION =>
                                etat := ATTENTE;
                                go <= '0';
                                index := 0;
                                data(31 downto 0):= "00000000000000000000000000000000";
                                data(63 downto 32):="00000000000000000000000000000000";
                                data(79 downto 64):="0000000000000000";
```

```
                        when ATTENTE =>
                                index := 0;
                                go <= '0';
                                if((ddi='1') and (entete='1')) then
                                        etat := ACCUMULE;
                                else
                                        etat := ATTENTE;
                                end if;

                        when ACCUMULE =>
                                go <= '0';

                                -- entree des donnees dans le data d'entree
                                data(78 downto 0) := data(79 downto 1);
                                data(79)          := entree;

                                if(index<79) then
                                        etat := ACCUMULE;
                                        index := index+1;
                                else
                                        if(ddi='1') then
                                                etat := VALIDE;
                                        else
                                                etat := ATTENTE;
                                        end if;
                                end if;

                        when VALIDE =>
                                go <= '1';
                                etat := ATTENTE;
                                index := 0;

                                HFA             <= data(76 downto 72);
                                HFP             <= data(71 downto 56);
                                HFW             <= data(55 downto 48);
                                HFI             <= data(47 downto 40);
                                LFA             <= data(36 downto 32);
                                LFP             <= data(31 downto 16);
                                LFW             <= data(15 downto 8);
                                LFI             <= data(7 downto 0);

                        when others =>
                                etat := INITIALISATION;
                                go        <= '0';

                        end case;
                end if;
        end process FSM;
end Stimulation;

-----------------------------------------------------------------------------
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- Equipe de Recherche en Neurotechnologies
--
-- ELE6305 Conception de circuit electroniques integres II
-- Realisation d'un micro-stimulateur implantable totalement integre
-- Session hiver 1997
--
-- Auteurs: Pierre Vaillancourt, Simon Robin, Alexandre Beauchamp-Parent
--
-- Module de synchronisation de la sequence de commande
--      Ce module doit trouver les bonnes entetes pour synchroniser la
--      sequence de commande avec les bonnes donnees au cas ou l'entete
--      se retrouverait par erreur dans le flot de donnees arrivantes.
--      Ce type d'erreur peut etre du a du bruit dans la transmission
--      et ne doit pas affecter la reception.
--
-- Fichier: sequence.vhdl
-----------------------------------------------------------------------------
--  Date        Modification
--
--  97-02-03    Creation
--  97-02-09    Modification pour le code correcteur BCH(31,16), mise en
--              forme avec les codes de detection a la fin des donnees
--  97-04-03    Modifications pour la testabilite
--
```

```
--------------------------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;
use ieee.std_logic_arith.all;


entity sequence is
        port(   entree  : in    std_logic;
                entete  : in    std_logic;
                sample  : in    std_logic;
                clk     : in    std_logic;
                por     : in    std_logic;

                isValid : out   std_logic;

                block0  : out   std_logic_vector(15 downto 0);
                block1  : out   std_logic_vector(15 downto 0);
                block2  : out   std_logic_vector(15 downto 0);
                block3  : out   std_logic_vector(15 downto 0);
                block4  : out   std_logic_vector(15 downto 0);
                CRC0    : out   std_logic_vector(15 downto 0);
                CRC1    : out   std_logic_vector(15 downto 0);
                CRC2    : out   std_logic_vector(15 downto 0);
                CRC3    : out   std_logic_vector(15 downto 0);
                CRC4    : out   std_logic_vector(15 downto 0)
        );
end sequence;

architecture Stimulation of sequence is
        type Etat_Type is (INITIALISATION, ATTENTE, ACCUMULE, VALIDE);
        type Data_Type is (blk0, blk1, blk2, blk3, blk4,
                           E_CRC0, E_CRC1, E_CRC2, E_CRC3, E_CRC4,
                           header);
begin
        FSM: process
                variable etat          :       Etat_Type;
                variable data          :       Data_Type;
                variable index           :     integer;
                variable temp            :     std_logic_vector(15 downto 0);
        begin
                wait until clk'event and clk='1';
                if(por='0') then
                        etat := ATTENTE;
                        data := blk0;
                        isValid <= '0';
                        temp := "0000000000000000";
                        index := 0;
                else
                        case etat is
                        when INITIALISATION =>
                                etat := ATTENTE;
                                isValid      <= '0';
                                temp         := "0000000000000000";
                                index := 0;
                                data := blk0;

                        when ATTENTE =>
                                data := blk0;
                                isValid <= '0';
                                if((sample='1') and (entete='1')) then
                                        etat := ACCUMULE;
                                else
                                        etat := ATTENTE;
                                end if;
                                index := 0;

                        when ACCUMULE =>
                                isValid <= '0';

                                -- entree des donnees dans le data d'entree
                                temp(14 downto 0) := temp(15 downto 1);
                                temp(15)          := entree;

                                if(index<15 ) then
                                        if(entete='0') then
                                                etat := ACCUMULE;
```

```
                index := index+1;
        else
                etat := ACCUMULE;
                data := blk0;
                index := 0;
        end if;
else
        index := 0;
        case data is
        when blk0 =>
                if(entete='1') then
                        etat := ACCUMULE;
                        data := blk0;
                else
                        etat := ACCUMULE;
                        data := blk1;
                        block0 <= temp;
                end if;

        when blk1 =>
                if(entete='1') then
                        etat := ACCUMULE;
                        data := blk0;
                else
                        etat := ACCUMULE;
                        data := blk2;
                        block1 <= temp;
                end if;

        when blk2 =>
                if(entete='1') then
                        etat := ACCUMULE;
                        data := blk0;
                else
                        etat := ACCUMULE;
                        data := blk3;
                        block2 <= temp;
                end if;

        when blk3 =>
                if(entete='1') then
                        etat := ACCUMULE;
                        data := blk0;
                else
                        etat := ACCUMULE;
                        data := blk4;
                        block3 <= temp;
                end if;

        when blk4 =>
                if(entete='1') then
                        etat := ACCUMULE;
                        data := blk0;
                else
                        etat := ACCUMULE;
                        data := E_CRC0;
                        block4 <= temp;
                end if;

        when E_CRC0 =>
                if(entete='1') then
                        etat := ACCUMULE;
                        data := blk0;
                else
                        etat := ACCUMULE;
                        data := E_CRC1;
                        CRC0 <= temp;
                end if;

        when E_CRC1 =>
                if(entete='1') then
                        etat := ACCUMULE;
                        data := blk0;
                else
                        etat := ACCUMULE;
                        data := E_CRC2;
                        CRC1 <= temp;
```

```
                                        end if;

                                when E_CRC2 =>
                                        if(entete='1') then
                                                etat := ACCUMULE;
                                                data := blk0;
                                        else
                                                etat := ACCUMULE;
                                                data := E_CRC3;
                                                CRC2 <= temp;
                                        end if;

                                when E_CRC3 =>
                                        if(entete='1') then
                                                etat := ACCUMULE;
                                                data := blk0;
                                        else
                                                etat := ACCUMULE;
                                                data := E_CRC4;
                                                CRC3 <= temp;
                                        end if;

                                when E_CRC4 =>
                                        if(entete='1') then
                                                etat := ACCUMULE;
                                                data := blk0;
                                        else
                                                etat := ACCUMULE;
                                                data := header;
                                                CRC4 <= temp;
                                        end if;

                                when header =>
                                        if(entete='1') then
                                                etat := VALIDE;
                                                data := blk0;
                                        else
                                                etat := ACCUMULE;
                                                data := blk0;
                                        end if;

                                when others =>
                                        null;
                                end case;
                        end if;

                when VALIDE =>
                        -- eviter que le isValid reste actif lorsque le sample tombe
                        isValid <= sample;
                        etat := ATTENTE;

                when others =>
                        etat := INITIALISATION;
                        isValid        <= '0';
            end case;
        end if;
        end process FSM;
end Stimulation;
```

```
-----------------------------------------------------------------------------
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- Equipe de Recherche en Neurotechnologies
--
-- ELE6305 Conception de circuit electroniques integres II
-- Realisation d'un micro-stimulateur implantable totalement integre
-- Session hiver 1997
--
-- Auteurs:Alexandre Beauchamp-Parent, Simon Robin, Pierre Vaillancourt
--
-- Module de sommation des amplitudes des deux signaux
--
-- Fichier: somme.vhdl
--
-----------------------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;
use ieee.std_logic_arith.all;

entity somme is
  port(CLK, EN_L, SIGN_L, EN_H, SIGN_H, TEST, GO : in std_logic;
       AMP_L, AMP_H : in std_logic_vector(4 downto 0);
       EN_SOMME, SIGN_SOMME : out std_logic;
       AMP_SOMME : out std_logic_vector(4 downto 0));
end somme;

architecture algo of somme is
begin
  process
  variable AMP_TOTALE : std_logic_vector(5 downto 0);
  variable ETAT_EN : std_logic_vector(1 downto 0);
  variable ETAT_SIGN : std_logic_vector(1 downto 0);
  variable SIGN_RAMPE : std_logic;
  variable AMP_RAMPE : std_logic_vector(4 downto 0);
  begin
    wait until CLK'event and CLK='1';
    if GO='0' then
      SIGN_RAMPE :='0';
      AMP_RAMPE :="00000";
    end if;
    if (TEST='1' and GO='1') then
      if AMP_RAMPE="00000" then
        EN_SOMME <='0';
        SIGN_SOMME <='-';
        AMP_SOMME <="00000";
      else
        EN_SOMME <='1';
        SIGN_SOMME <=SIGN_RAMPE;
        AMP_SOMME <=AMP_RAMPE;
      end if;
      if AMP_RAMPE="11111" then
        AMP_RAMPE :="00000";
        if SIGN_RAMPE='0' then
          SIGN_RAMPE :='1';
        else
          SIGN_RAMPE :='0';
        end if;
      else
        AMP_RAMPE :=AMP_RAMPE+'1';
      end if;
    else
      ETAT_EN :=EN_H & EN_L;
      case ETAT_EN is
        when "00" => EN_SOMME <='0';
                     SIGN_SOMME <='-';
                     AMP_SOMME <="00000";
        when "01" => EN_SOMME <='1';
                     SIGN_SOMME <=SIGN_L;
                     AMP_SOMME <=AMP_L;
        when "10" => EN_SOMME <='1';
                     SIGN_SOMME <=SIGN_H;
                     AMP_SOMME <=AMP_H;
        when "11" => ETAT_SIGN :=SIGN_H & SIGN_L;
                     case ETAT_SIGN is
                       when "00" => EN_SOMME <='1';
```

```
                                SIGN_SOMME <='0';
                                AMP_TOTALE :=('0' & AMP_L) + ('0' & AMP_H);
                                if AMP_TOTALE(5)='1' then
                                  AMP_SOMME <="11111";
                                else
                                  AMP_SOMME <=AMP_TOTALE(4 downto 0);
                                end if;
                        when "01" => if '0' & AMP_L > '0' & AMP_H then
                                  EN_SOMME <='1';
                                  SIGN_SOMME <='1';
                                  AMP_SOMME <=AMP_L - AMP_H;
                                elsif '0' & AMP_L < '0' & AMP_H then
                                  EN_SOMME <='1';
                                  SIGN_SOMME <='0';
                                  AMP_SOMME <=AMP_H - AMP_L;
                                else
                                  EN_SOMME <='0';
                                  SIGN_SOMME <='-';
                                  AMP_SOMME <="00000";
                                end if;
                        when "10" => if '0' & AMP_H > '0' & AMP_L then
                                  EN_SOMME <='1';
                                  SIGN_SOMME <='1';
                                  AMP_SOMME <=AMP_H - AMP_L;
                                elsif '0' & AMP_H < '0' & AMP_L then
                                  EN_SOMME <='1';
                                  SIGN_SOMME <='0';
                                  AMP_SOMME <=AMP_L - AMP_H;
                                else
                                  EN_SOMME <='0';
                                  SIGN_SOMME <='-';
                                  AMP_SOMME <="00000";
                                end if;
                        when "11" => EN_SOMME <='1';
                                SIGN_SOMME <='1';
                                AMP_TOTALE :=('0' & AMP_L) + ('0' & AMP_H);
                                if AMP_TOTALE(5)='1' then
                                  AMP_SOMME <="11111";
                                else
                                  AMP_SOMME <=AMP_TOTALE(4 downto 0);
                                end if;
                        when others => null;
                      end case;
        when others => null;
      end case;
    end if;
  end process;
end algo;
stim.vhdl
```

```
-----------------------------------------------------------------------------
-- Ecole Polytechnique de Montreal
-- Groupe de Recherche en Microelectronique
-- Equipe de Recherche en Neurotechnologies
--
-- ELE6305 Conception de circuit electroniques integres II
-- Realisation d'un micro-stimulateur implantable totalement integre
-- Session hiver 1997
--
-- Auteurs: Alexandre Beauchamp-Parent, Simon Robin, Pierre Vaillancourt
--
-- Module de synchronisation
--      Ce module synchronise le debut du signal a generer tous les FP coups
--      d'horloge.
--
-- Fichier: synchro.vhdl
-----------------------------------------------------------------------------
--  Date       Modification
--
--
-----------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;
use ieee.std_logic_arith.all;


entity synchro is
  port(CLK, GO : in std_logic;
       FP : in std_logic_vector(15 downto 0);
       SYNC : out std_logic);
end synchro;


architecture algo of synchro is
begin
  process
  variable COMPTEUR1 : std_logic_vector(15 downto 0);
  begin
    wait until CLK'event and CLK='1';
    if GO='0' then
      COMPTEUR1 :="0000000000000000";
      SYNC <='0';
    else
      if COMPTEUR1=FP then
        COMPTEUR1 :="0000000000000000";
        if FP="0000000000000000" then
          SYNC <='0';
        else
          SYNC <='1';
          COMPTEUR1 :="0000000000000001";
        end if;
      else
        COMPTEUR1 :=COMPTEUR1+'1';
        SYNC <='0';
      end if;
    end if;
  end process;
end algo;
```

## Résultats de synthèse du VHDL:

```
                    DC Professional (TM)
                       DC Expert (TM)
                     FPGA Compiler (TM)
                     VHDL Compiler (TM)
                      HDL Compiler (TM)
                   Library Compiler (TM)
                    Test Compiler (TM)
                 Test Compiler Plus (TM)
                       CTV-Interface
                      DesignTime (TM)

              Version v3.4b -- Apr 01, 1996
        Copyright (c) 1988-1995 by Synopsys, Inc.
                   ALL RIGHTS RESERVED

This program is proprietary and confidential information of Synopsys, Inc.
and may be used and disclosed only as authorized in a license agreement
controlling such use and disclosure.

Initializing...
analyze -format vhdl {synchro.vhdl somme.vhdl dsp.vhdl uc.vhdl comm.vhdl bch_d.vhdl
detect.vhdl header.vhdl seq_ddi.vhdl sequence.vhdl stim.vhdl}
Loading db file '/usr/local/packages/synopsys3-4b/libraries/syn/standard.sldb'
Loading db file '/usr/local/packages/synopsys3-4b/libraries/syn/gtech.db'
Loading db file '/usr/local/packages/synopsys3-4b/cmc/bicmos/syn/tcells.db'
Reading in the Synopsys vhdl primitives.
/mnt/freia/3/cours3/vlsi4/synth/synchro.vhdl:
/mnt/freia/3/cours3/vlsi4/synth/somme.vhdl:
/mnt/freia/3/cours3/vlsi4/synth/dsp.vhdl:
/mnt/freia/3/cours3/vlsi4/synth/uc.vhdl:
/mnt/freia/3/cours3/vlsi4/synth/comm.vhdl:
/mnt/freia/3/cours3/vlsi4/synth/bch_d.vhdl:
/mnt/freia/3/cours3/vlsi4/synth/detect.vhdl:
/mnt/freia/3/cours3/vlsi4/synth/header.vhdl:
/mnt/freia/3/cours3/vlsi4/synth/seq_ddi.vhdl:
/mnt/freia/3/cours3/vlsi4/synth/sequence.vhdl:
/mnt/freia/3/cours3/vlsi4/synth/stim.vhdl:
1
elaborate stim
Information: Building the design 'uc'. (HDL-193)
Information: Building the design 'comm'. (HDL-193)

Inferred memory devices in process
        in routine comm line 150 in file
          '/mnt/freia/3/cours3/vlsi4/synth/comm.vhdl'.
===============================================================================
      Register Name           Type      Width  Bus  AR  AS  SR  SS  ST
===============================================================================
          HFA_reg            Flip-flop    5     Y   N   N   N   N   N
          HFI_reg            Flip-flop    8     Y   N   N   N   N   N
          HFP_reg            Flip-flop   16     Y   N   N   N   N   N
          HFW_reg            Flip-flop    8     Y   N   N   N   N   N
          LFA_reg            Flip-flop    5     Y   N   N   N   N   N
          LFI_reg            Flip-flop    8     Y   N   N   N   N   N
          LFP_reg            Flip-flop   16     Y   N   N   N   N   N
          LFW_reg            Flip-flop    8     Y   N   N   N   N   N
           go_reg            Flip-flop    1     -   N   N   N   N   N
   testHeaderDetect_reg      Flip-flop    1     -   N   N   N   N   N
       testMode_reg          Flip-flop    1     -   N   N   N   N   N
===============================================================================

Information: Building the design 'synchro'. (HDL-193)

Inferred memory devices in process
        in routine synchro line 38 in file
          '/mnt/freia/3/cours3/vlsi4/synth/synchro.vhdl'.
===============================================================================
      Register Name           Type      Width  Bus  AR  AS  SR  SS  ST
===============================================================================
      COMPTEUR1_reg          Flip-flop   16     Y   N   N   N   N   N
         SYNC_reg            Flip-flop    1     -   N   N   N   N   N
===============================================================================
```

```
Information: Building the design 'dsp'. (HDL-193)

Inferred memory devices in process
        in routine dsp line 18 in file
          '/mnt/freia/3/cours3/vlsi4/synth/dsp.vhdl'.
===============================================================================
      Register Name          Type      Width  Bus  AR  AS  SR  SS  ST
===============================================================================
         AMP_reg           Flip-flop    5     Y   N   N   N   N   N
      COMPTEUR2_reg        Flip-flop    8     N   ?   ?   ?   ?   ?
      COMPTEUR3_reg        Flip-flop    8     Y   N   N   N   N   N
      COMPTEUR4_reg        Flip-flop    8     Y   N   N   N   N   N
          EN_reg           Flip-flop    1     -   N   N   N   N   N
         ETAT_reg          Flip-flop    2     N   ?   ?   ?   ?   ?
         SIGN_reg          Flip-flop    1     -   N   N   N   N   N
===============================================================================

Information: Building the design 'somme'. (HDL-193)

Inferred memory devices in process
        in routine somme line 15 in file
          '/mnt/freia/3/cours3/vlsi4/synth/somme.vhdl'.
===============================================================================
      Register Name          Type      Width  Bus  AR  AS  SR  SS  ST
===============================================================================
      AMP_RAMPE_reg        Flip-flop    5     Y   N   N   N   N   N
      AMP_SOMME_reg        Flip-flop    5     Y   N   N   N   N   N
       EN_SOMME_reg        Flip-flop    1     -   N   N   N   N   N
      SIGN_RAMPE_reg       Flip-flop    1     -   N   N   N   N   N
      SIGN_SOMME_reg       Flip-flop    1     -   N   N   N   N   N
===============================================================================

Information: Building the design 'Header'. (HDL-193)

Inferred memory devices in process 'compare'
        in routine Header line 45 in
          file '/mnt/freia/3/cours3/vlsi4/synth/header.vhdl'.
===============================================================================
      Register Name          Type      Width  Bus  AR  AS  SR  SS  ST
===============================================================================
   ddiHeaderDetect_reg     Flip-flop    1     -   N   N   N   N   N
         fifo_reg          Flip-flop   15     N   ?   ?   ?   ?   ?
        sortie_reg         Flip-flop    1     -   N   N   N   N   N
 stimulationHeaderDetect_reg  Flip-flop  1     -   N   N   N   N   N
   testHeaderDetect_reg    Flip-flop    1     -   N   N   N   N   N
===============================================================================

Information: Building the design 'sequence'. (HDL-193)

Inferred memory devices in process 'FSM'
        in routine sequence line 63 in file
          '/mnt/freia/3/cours3/vlsi4/synth/sequence.vhdl'.
===============================================================================
      Register Name          Type      Width  Bus  AR  AS  SR  SS  ST
===============================================================================
         CRC0_reg           Flip-flop   16     Y   N   N   N   N   N
         CRC1_reg           Flip-flop   16     Y   N   N   N   N   N
         CRC2_reg           Flip-flop   16     Y   N   N   N   N   N
         CRC3_reg           Flip-flop   16     Y   N   N   N   N   N
         CRC4_reg           Flip-flop   16     Y   N   N   N   N   N
        block0_reg          Flip-flop   16     Y   N   N   N   N   N
        block1_reg          Flip-flop   16     Y   N   N   N   N   N
        block2_reg          Flip-flop   16     Y   N   N   N   N   N
        block3_reg          Flip-flop   16     Y   N   N   N   N   N
        block4_reg          Flip-flop   16     Y   N   N   N   N   N
         data_reg           Flip-flop    4     Y   N   N   N   N   N
         etat_reg           Flip-flop    2     Y   N   N   N   N   N
        index_reg           Flip-flop   32     Y   N   N   N   N   N
       isValid_reg          Flip-flop    1     -   N   N   N   N   N
         temp_reg           Flip-flop   15     N   ?   ?   ?   ?   ?
===============================================================================

Information: Building the design 'Detect'. (HDL-193)

Inferred memory devices in process 'FSM'
        in routine Detect line 130 in file
          '/mnt/freia/3/cours3/vlsi4/synth/detect.vhdl'.
```

```
==============================================================================
      Register Name          Type      Width  Bus  AR  AS  SR  SS  ST
==============================================================================
         HFA_reg           Flip-flop    5     Y   N   N   N   N   N
         HFI_reg           Flip-flop    8     Y   N   N   N   N   N
         HFP_reg           Flip-flop   16     Y   N   N   N   N   N
         HFW_reg           Flip-flop    8     Y   N   N   N   N   N
         LFA_reg           Flip-flop    5     Y   N   N   N   N   N
         LFI_reg           Flip-flop    8     Y   N   N   N   N   N
         LFP_reg           Flip-flop   16     Y   N   N   N   N   N
         LFW_reg           Flip-flop    8     Y   N   N   N   N   N
        detect_reg         Flip-flop    1     -   N   N   N   N   N
         etat_reg          Flip-flop    2     Y   N   N   N   N   N
          go_reg           Flip-flop    1     -   N   N   N   N   N
        sample_reg         Flip-flop    1     -   N   N   N   N   N
==============================================================================

Information: Building the design 'seq_ddi'. (HDL-193)

Inferred memory devices in process 'FSM'
        in routine seq_ddi line 57 in file
          '/mnt/freia/3/cours3/vlsi4/synth/seq_ddi.vhdl'.
==============================================================================
      Register Name          Type      Width  Bus  AR  AS  SR  SS  ST
==============================================================================
         HFA_reg           Flip-flop    5     Y   N   N   N   N   N
         HFI_reg           Flip-flop    8     Y   N   N   N   N   N
         HFP_reg           Flip-flop   16     Y   N   N   N   N   N
         HFW_reg           Flip-flop    8     Y   N   N   N   N   N
         LFA_reg           Flip-flop    5     Y   N   N   N   N   N
         LFI_reg           Flip-flop    8     Y   N   N   N   N   N
         LFP_reg           Flip-flop   16     Y   N   N   N   N   N
         LFW_reg           Flip-flop    8     Y   N   N   N   N   N
         data_reg          Flip-flop   80     Y   N   N   N   N   N
         etat_reg          Flip-flop    2     Y   N   N   N   N   N
          go_reg           Flip-flop    1     -   N   N   N   N   N
        index_reg          Flip-flop   32     Y   N   N   N   N   N
==============================================================================

Information: Building the design 'BCH_D'. (HDL-193)

Inferred memory devices in process
        in routine BCH_D line 45 in file
          '/mnt/freia/3/cours3/vlsi4/synth/bch_d.vhdl'.
==============================================================================
      Register Name          Type      Width  Bus  AR  AS  SR  SS  ST
==============================================================================
        CRC_I_reg          Flip-flop   16     Y   N   N   N   N   N
          OK_reg           Flip-flop    1     -   N   N   N   N   N
         etat_reg          Flip-flop    2     Y   N   N   N   N   N
        finish_reg         Flip-flop    1     -   N   N   N   N   N
        index_reg          Flip-flop   32     Y   N   N   N   N   N
         reg_reg           Flip-flop   15     N   ?   ?   ?   ?   ?
         tmp_reg           Flip-flop   16     Y   N   N   N   N   N
==============================================================================

Current design is now 'stim'
1
set_test_methodology full_scan
1
set_scan_style multiplexed_flip_flop
1
uniquify
  Uniquifying cell 'synchro_H' in design 'uc'.  New design is 'synchro_0'.
  Uniquifying cell 'synchro_L' in design 'uc'.  New design is 'synchro_1'.
  Uniquifying cell 'dsp_H' in design 'uc'.  New design is 'dsp_0'.
  Uniquifying cell 'dsp_L' in design 'uc'.  New design is 'dsp_1'.
  Uniquifying cell 'detecteurE' in design 'Detect'.  New design is 'BCH_D_0'.
  Uniquifying cell 'detecteurD' in design 'Detect'.  New design is 'BCH_D_1'.
  Uniquifying cell 'detecteurC' in design 'Detect'.  New design is 'BCH_D_2'.
  Uniquifying cell 'detecteurB' in design 'Detect'.  New design is 'BCH_D_3'.
  Uniquifying cell 'detecteurA' in design 'Detect'.  New design is 'BCH_D_4'.
1
compile
Information: Choosing a test methodology will restrict the optimization of sequential
cells. (UIO-12)
```

```
  Loading target library 'tcells'
  Loading design 'stim'
Warning: In design 'BCH_D_4', there is 1 port not connected to any nets. (LINT-30)
Warning: In design 'BCH_D_3', there is 1 port not connected to any nets. (LINT-30)
Warning: In design 'BCH_D_2', there is 1 port not connected to any nets. (LINT-30)
Warning: In design 'BCH_D_1', there is 1 port not connected to any nets. (LINT-30)
Warning: In design 'BCH_D_0', there is 1 port not connected to any nets. (LINT-30)

Information: Use the 'check_design' command for
        more information about warnings. (LINT-99)

Information: Design 'stim' has no optimization constraints set. (OPT-108)

  Beginning CMOS optimization
  ---------------------------

  Beginning Resource Allocation  (area only)
  ------------------------------
  Allocating blocks in 'stim'
  Allocating blocks in 'uc_module'
  Allocating blocks in 'uc_module/synchro_L'
  Allocating blocks in 'uc_module/synchro_L'
  Allocating blocks in 'uc_module/synchro_H'
  Allocating blocks in 'uc_module/synchro_H'
  Allocating blocks in 'uc_module/dsp_L'
  Allocating blocks in 'uc_module/dsp_L'
  Allocating blocks in 'uc_module/dsp_H'
  Allocating blocks in 'uc_module/dsp_H'
  Allocating blocks in 'uc_module/the_somme'
  Allocating blocks in 'uc_module/the_somme'
  Allocating blocks in 'comm_module'
  Allocating blocks in 'comm_module'
  Allocating blocks in 'comm_module/Header_Module'
  Allocating blocks in 'comm_module/Header_Module'
  Allocating blocks in 'comm_module/Sequence_Module'
  Allocating blocks in 'comm_module/Sequence_Module'
  Allocating blocks in 'comm_module/Detect_Module'
  Allocating blocks in 'comm_module/Detect_Module'
  Allocating blocks in 'comm_module/Detect_Module/detecteurA'
  Allocating blocks in 'comm_module/Detect_Module/detecteurA'
  Allocating blocks in 'comm_module/Detect_Module/detecteurB'
  Allocating blocks in 'comm_module/Detect_Module/detecteurB'
  Allocating blocks in 'comm_module/Detect_Module/detecteurC'
  Allocating blocks in 'comm_module/Detect_Module/detecteurC'
  Allocating blocks in 'comm_module/Detect_Module/detecteurD'
  Allocating blocks in 'comm_module/Detect_Module/detecteurD'
  Allocating blocks in 'comm_module/Detect_Module/detecteurE'
  Allocating blocks in 'comm_module/Detect_Module/detecteurE'
  Allocating blocks in 'comm_module/DDI_Module'
  Allocating blocks in 'comm_module/DDI_Module'
  Selecting implementations in 'uc_module/synchro_L'
  Selecting implementations in 'uc_module/synchro_H'
  Selecting implementations in 'uc_module/dsp_L'
  Selecting implementations in 'uc_module/dsp_H'
  Selecting implementations in 'uc_module/the_somme'
  Selecting implementations in 'comm_module/Sequence_Module'
  Selecting implementations in 'comm_module/Detect_Module/detecteurA'
  Selecting implementations in 'comm_module/Detect_Module/detecteurB'
  Selecting implementations in 'comm_module/Detect_Module/detecteurC'
  Selecting implementations in 'comm_module/Detect_Module/detecteurD'
  Selecting implementations in 'comm_module/Detect_Module/detecteurE'
  Selecting implementations in 'comm_module/DDI_Module'

  Beginning Mapping Optimizations  (Medium effort)
  -------------------------------
  Structuring 'synchro_1_DW01_inc_16_0'
  Mapping 'synchro_1_DW01_inc_16_0'
  Structuring 'synchro_0_DW01_inc_16_0'
  Mapping 'synchro_0_DW01_inc_16_0'
  Structuring 'dsp_1_DW01_inc_8_0'
  Mapping 'dsp_1_DW01_inc_8_0'
  Structuring 'dsp_0_DW01_inc_8_0'
  Mapping 'dsp_0_DW01_inc_8_0'
  Structuring 'somme_DW01_cmp2_5_1'
  Mapping 'somme_DW01_cmp2_5_1'
  Structuring 'somme_DW01_cmp2_5_0'
  Mapping 'somme_DW01_cmp2_5_0'
```

```
Structuring 'somme_DW01_addsub_5_0'
Mapping 'somme_DW01_addsub_5_0'
Structuring 'somme_DW01_add_6_0'
Mapping 'somme_DW01_add_6_0'
Structuring 'sequence_DW01_inc_32_0'
Mapping 'sequence_DW01_inc_32_0'
Structuring 'sequence_DW01_cmp2_32_0'
Mapping 'sequence_DW01_cmp2_32_0'
Structuring 'BCH_D_4_DW01_inc_32_0'
Mapping 'BCH_D_4_DW01_inc_32_0'
Structuring 'BCH_D_4_DW01_cmp2_32_0'
Mapping 'BCH_D_4_DW01_cmp2_32_0'
Structuring 'BCH_D_3_DW01_inc_32_0'
Mapping 'BCH_D_3_DW01_inc_32_0'
Structuring 'BCH_D_3_DW01_cmp2_32_0'
Mapping 'BCH_D_3_DW01_cmp2_32_0'
Structuring 'BCH_D_2_DW01_inc_32_0'
Mapping 'BCH_D_2_DW01_inc_32_0'
Structuring 'BCH_D_2_DW01_cmp2_32_0'
Mapping 'BCH_D_2_DW01_cmp2_32_0'
Structuring 'BCH_D_1_DW01_inc_32_0'
Mapping 'BCH_D_1_DW01_inc_32_0'
Structuring 'BCH_D_1_DW01_cmp2_32_0'
Mapping 'BCH_D_1_DW01_cmp2_32_0'
Structuring 'BCH_D_0_DW01_inc_32_0'
Mapping 'BCH_D_0_DW01_inc_32_0'
Structuring 'BCH_D_0_DW01_cmp2_32_0'
Mapping 'BCH_D_0_DW01_cmp2_32_0'
Structuring 'seq_ddi_DW01_cmp2_32_0'
Mapping 'seq_ddi_DW01_cmp2_32_0'
Structuring 'seq_ddi_DW01_inc_32_0'
Mapping 'seq_ddi_DW01_inc_32_0'
Structuring 'seq_ddi'
Mapping 'seq_ddi'
Structuring 'BCH_D_0'
Mapping 'BCH_D_0'
Structuring 'BCH_D_1'
Mapping 'BCH_D_1'
Structuring 'BCH_D_2'
Mapping 'BCH_D_2'
Structuring 'BCH_D_3'
Mapping 'BCH_D_3'
Structuring 'BCH_D_4'
Mapping 'BCH_D_4'
Structuring 'Detect'
Mapping 'Detect'
Structuring 'sequence'
Mapping 'sequence'
Structuring 'Header'
Mapping 'Header'
Structuring 'comm'
Mapping 'comm'
Structuring 'somme'
Mapping 'somme'
Structuring 'dsp_0'
Mapping 'dsp_0'
Structuring 'dsp_1'
Mapping 'dsp_1'
Structuring 'synchro_0'
Mapping 'synchro_0'
Structuring 'synchro_1'
Mapping 'synchro_1'

                                                OPTIMIZATION   DESIGN RULE
          TRIALS      AREA     DELTA DELAY        COST          COST
         --------    ------    -----------       ------        ------
             124
         --------
             124

Optimization complete
---------------------
Transferring Design 'seq_ddi_DW01_cmp2_32_0' to database 'seq_ddi.db'
Transferring Design 'seq_ddi_DW01_inc_32_0' to database 'seq_ddi.db'
Transferring Design 'seq_ddi' to database 'seq_ddi.db'
Transferring Design 'BCH_D_0_DW01_inc_32_0' to database 'stim.db'
```

```
  Transferring Design 'BCH_D_0_DW01_cmp2_32_0' to database 'stim.db'
  Transferring Design 'BCH_D_0' to database 'stim.db'
  Transferring Design 'BCH_D_1_DW01_inc_32_0' to database 'stim.db'
  Transferring Design 'BCH_D_1_DW01_cmp2_32_0' to database 'stim.db'
  Transferring Design 'BCH_D_1' to database 'stim.db'
  Transferring Design 'BCH_D_2_DW01_inc_32_0' to database 'stim.db'
  Transferring Design 'BCH_D_2_DW01_cmp2_32_0' to database 'stim.db'
  Transferring Design 'BCH_D_2' to database 'stim.db'
  Transferring Design 'BCH_D_3_DW01_inc_32_0' to database 'stim.db'
  Transferring Design 'BCH_D_3_DW01_cmp2_32_0' to database 'stim.db'
  Transferring Design 'BCH_D_3' to database 'stim.db'
  Transferring Design 'BCH_D_4_DW01_inc_32_0' to database 'stim.db'
  Transferring Design 'BCH_D_4_DW01_cmp2_32_0' to database 'stim.db'
  Transferring Design 'BCH_D_4' to database 'stim.db'
  Transferring Design 'Detect' to database 'Detect.db'
  Transferring Design 'sequence_DW01_inc_32_0' to database 'sequence.db'
  Transferring Design 'sequence_DW01_cmp2_32_0' to database 'sequence.db'
  Transferring Design 'sequence' to database 'sequence.db'
  Transferring Design 'Header' to database 'Header.db'
  Transferring Design 'comm' to database 'comm.db'
  Transferring Design 'somme_DW01_cmp2_5_1' to database 'somme.db'
  Transferring Design 'somme_DW01_cmp2_5_0' to database 'somme.db'
  Transferring Design 'somme_DW01_addsub_5_0' to database 'somme.db'
  Transferring Design 'somme_DW01_add_6_0' to database 'somme.db'
  Transferring Design 'somme' to database 'somme.db'
  Transferring Design 'dsp_0_DW01_inc_8_0' to database 'stim.db'
  Transferring Design 'dsp_0' to database 'stim.db'
  Transferring Design 'dsp_1_DW01_inc_8_0' to database 'stim.db'
  Transferring Design 'dsp_1' to database 'stim.db'
  Transferring Design 'synchro_0_DW01_inc_16_0' to database 'stim.db'
  Transferring Design 'synchro_0' to database 'stim.db'
  Transferring Design 'synchro_1_DW01_inc_16_0' to database 'stim.db'
  Transferring Design 'synchro_1' to database 'stim.db'
  Transferring Design 'uc' to database 'uc.db'
  Transferring Design 'stim' to database 'stim.db'
Current design is 'stim'.
1
insert_scan
  Loading design 'stim'
Warning: In design 'somme', there are 4 submodules connected to power or ground. (LINT-30)
Warning: In design 'somme', there is 1 submodule with pins connected to the same net.
(LINT-30)
Warning: In design 'somme_DW01_cmp2_5_1', there are 2 ports not connected to any nets.
(LINT-30)
Warning: In design 'somme_DW01_cmp2_5_0', there are 2 ports not connected to any nets.
(LINT-30)
Warning: In design 'somme_DW01_addsub_5_0', there are 2 ports not connected to any nets.
(LINT-30)
Warning: In design 'somme_DW01_add_6_0', there are 4 ports not connected to any nets.
(LINT-30)
Warning: In design 'sequence', there is 1 submodule connected to power or ground. (LINT-
30)
Warning: In design 'sequence', there is 1 submodule with pins connected to the same net.
(LINT-30)
Warning: In design 'sequence_DW01_cmp2_32_0', there are 35 ports not connected to any
nets. (LINT-30)
Warning: In design 'BCH_D_0', there is 1 submodule connected to power or ground. (LINT-30)
Warning: In design 'BCH_D_0', there is 1 submodule with pins connected to the same net.
(LINT-30)
Warning: In design 'BCH_D_0_DW01_cmp2_32_0', there are 34 ports not connected to any nets.
(LINT-30)
Warning: In design 'seq_ddi', there is 1 submodule connected to power or ground. (LINT-30)
Warning: In design 'seq_ddi', there is 1 submodule with pins connected to the same net.
(LINT-30)
Warning: In design 'seq_ddi_DW01_cmp2_32_0', there are 35 ports not connected to any nets.
(LINT-30)

Information: Use the 'check_design' command for
        more information about warnings. (LINT-99)

  Checking test design rules
  Architecting Scan Chains
  Inserting Scan Cells
  Routing Scan Chains
  Routing Global Signals
  Mapping New Logic
  Transferring design 'BCH_D_0_DW01_cmp2_32_0' to database 'stim.db'
```

```
   Transferring design 'BCH_D_0_DW01_inc_32_0' to database 'stim.db'
   Transferring (new) design 'BCH_D_test_1' to database 'stim.db'
   Transferring design 'BCH_D_1_DW01_cmp2_32_0' to database 'stim.db'
   Transferring design 'BCH_D_1_DW01_inc_32_0' to database 'stim.db'
   Transferring (new) design 'BCH_D_test_2' to database 'stim.db'
   Transferring design 'BCH_D_2_DW01_cmp2_32_0' to database 'stim.db'
   Transferring design 'BCH_D_2_DW01_inc_32_0' to database 'stim.db'
   Transferring (new) design 'BCH_D_test_3' to database 'stim.db'
   Transferring design 'BCH_D_3_DW01_cmp2_32_0' to database 'stim.db'
   Transferring design 'BCH_D_3_DW01_inc_32_0' to database 'stim.db'
   Transferring (new) design 'BCH_D_test_4' to database 'stim.db'
   Transferring design 'BCH_D_4_DW01_cmp2_32_0' to database 'stim.db'
   Transferring design 'BCH_D_4_DW01_inc_32_0' to database 'stim.db'
   Transferring (new) design 'BCH_D_test_5' to database 'stim.db'
   Transferring (new) design 'Detect_test_1' to database 'Detect.db'
   Transferring (new) design 'Header_test_1' to database 'Header.db'
   Transferring design 'seq_ddi_DW01_cmp2_32_0' to database 'seq_ddi.db'
   Transferring design 'seq_ddi_DW01_inc_32_0' to database 'seq_ddi.db'
   Transferring (new) design 'seq_ddi_test_1' to database 'seq_ddi.db'
   Transferring design 'sequence_DW01_cmp2_32_0' to database 'sequence.db'
   Transferring design 'sequence_DW01_inc_32_0' to database 'sequence.db'
   Transferring (new) design 'sequence_test_1' to database 'sequence.db'
   Transferring (new) design 'comm_test_1' to database 'comm.db'
   Transferring design 'dsp_0_DW01_inc_8_0' to database 'stim.db'
   Transferring (new) design 'dsp_test_1' to database 'stim.db'
   Transferring design 'dsp_1_DW01_inc_8_0' to database 'stim.db'
   Transferring (new) design 'dsp_test_2' to database 'stim.db'
   Transferring design 'somme_DW01_add_6_0' to database 'somme.db'
   Transferring design 'somme_DW01_addsub_5_0' to database 'somme.db'
   Transferring design 'somme_DW01_cmp2_5_0' to database 'somme.db'
   Transferring design 'somme_DW01_cmp2_5_1' to database 'somme.db'
   Transferring (new) design 'somme_test_1' to database 'somme.db'
   Transferring design 'synchro_0_DW01_inc_16_0' to database 'stim.db'
   Transferring (new) design 'synchro_test_1' to database 'stim.db'
   Transferring design 'synchro_1_DW01_inc_16_0' to database 'stim.db'
   Transferring (new) design 'synchro_test_2' to database 'stim.db'
   Transferring (new) design 'uc_test_1' to database 'uc.db'
   Transferring design 'stim' to database 'stim.db'

1
write -format db -output stim.db
Writing to file /tmp_mnt/mnt/freia/3/cours3/vlsi4/synth/stim.db
1
write -format edif -hierarchy -output "stim.edif" {"stim.db:stim"}
Warning: The design 'somme_test_1'
        uses the built-in Synopsys logic 0 reference. (EDFO-46)
Warning: The design 'seq_ddi_test_1'
        uses the built-in Synopsys logic 0 reference. (EDFO-46)
Warning: The design 'seq_ddi_test_1'
        uses the built-in Synopsys logic 1 reference. (EDFO-46)
Warning: The design 'BCH_D_test_5'
        uses the built-in Synopsys logic 0 reference. (EDFO-46)
Warning: The design 'BCH_D_test_5'
        uses the built-in Synopsys logic 1 reference. (EDFO-46)
Warning: The design 'BCH_D_test_4'
        uses the built-in Synopsys logic 0 reference. (EDFO-46)
Warning: The design 'BCH_D_test_4'
        uses the built-in Synopsys logic 1 reference. (EDFO-46)
Warning: The design 'BCH_D_test_3'
        uses the built-in Synopsys logic 0 reference. (EDFO-46)
Warning: The design 'BCH_D_test_3'
        uses the built-in Synopsys logic 1 reference. (EDFO-46)
Warning: The design 'BCH_D_test_2'
        uses the built-in Synopsys logic 0 reference. (EDFO-46)
Warning: The design 'BCH_D_test_2'
        uses the built-in Synopsys logic 1 reference. (EDFO-46)
Warning: The design 'BCH_D_test_1'
        uses the built-in Synopsys logic 0 reference. (EDFO-46)
Warning: The design 'BCH_D_test_1'
        uses the built-in Synopsys logic 1 reference. (EDFO-46)
Warning: The design 'sequence_test_1'
        uses the built-in Synopsys logic 0 reference. (EDFO-46)
Warning: The design 'sequence_test_1'
        uses the built-in Synopsys logic 1 reference. (EDFO-46)
Information: a bussed port name 'A' changed to 'A<7:0>'. (EXPT-11)
Information: a bussed port name 'SUM' changed to 'SUM<7:0>'. (EXPT-11)
Information: a bussed port name 'AMP' changed to 'AMP<4:0>'. (EXPT-11)
```

```
Information: a bussed port name 'FA' changed to 'FA<4:0>'. (EXPT-11)
Information: a bussed port name 'FI' changed to 'FI<7:0>'. (EXPT-11)
Information: a bussed port name 'FW' changed to 'FW<7:0>'. (EXPT-11)
Information: Net 'n196<7>' is renamed to 'SYNC' in design 'dsp_test_1' because it's
connected to the port by that name. (EDFO-10)
Information: a bussed port name 'A' changed to 'A<7:0>'. (EXPT-11)
Information: a bussed port name 'SUM' changed to 'SUM<7:0>'. (EXPT-11)
Information: a bussed port name 'AMP' changed to 'AMP<4:0>'. (EXPT-11)
Information: a bussed port name 'FA' changed to 'FA<4:0>'. (EXPT-11)
Information: a bussed port name 'FI' changed to 'FI<7:0>'. (EXPT-11)
Information: a bussed port name 'FW' changed to 'FW<7:0>'. (EXPT-11)
Information: Net 'n196<7>' is renamed to 'SYNC' in design 'dsp_test_2' because it's
connected to the port by that name. (EDFO-10)
Information: a bussed port name 'A' changed to 'A<15:0>'. (EXPT-11)
Information: a bussed port name 'SUM' changed to 'SUM<15:0>'. (EXPT-11)
Information: Net 'net6178' is renamed to 'SUM<0>' in design 'synchro_0_DW01_inc_16_0'
because it's connected to the port by that name. (EDFO-10)
Information: Net 'net6192' is renamed to 'A<4>' in design 'synchro_0_DW01_inc_16_0'
because it's connected to the port by that name. (EDFO-10)
Information: Net 'net6193' is renamed to 'A<6>' in design 'synchro_0_DW01_inc_16_0'
because it's connected to the port by that name. (EDFO-10)
because it's connected to the port by that name. (EDFO-10)
Information: Net 'net6209' is renamed to 'A<10>' in design 'synchro_0_DW01_inc_16_0'
because it's connected to the port by that name. (EDFO-10)
Information: Net 'net6210' is renamed to 'A<11>' in design 'synchro_0_DW01_inc_16_0'
because it's connected to the port by that name. (EDFO-10)
Information: Net 'net6211' is renamed to 'SUM<12>' in design 'synchro_0_DW01_inc_16_0'
because it's connected to the port by that name. (EDFO-10)
Information: Net 'net6213' is renamed to 'A<7>' in design 'synchro_0_DW01_inc_16_0'
because it's connected to the port by that name. (EDFO-10)
Information: Net 'net6216' is renamed to 'SUM<1>' in design 'synchro_0_DW01_inc_16_0'
because it's connected to the port by that name. (EDFO-10)
Information: Net 'net6217' is renamed to 'SUM<3>' in design 'synchro_0_DW01_inc_16_0'
because it's connected to the port by that name. (EDFO-10)
Information: Net 'net6218' is renamed to 'SUM<7>' in design 'synchro_0_DW01_inc_16_0'
because it's connected to the port by that name. (EDFO-10)
Information: Net 'net6219' is renamed to 'SUM<5>' in design 'synchro_0_DW01_inc_16_0'
because it's connected to the port by that name. (EDFO-10)
Information: Net 'net6220' is renamed to 'SUM<8>' in design 'synchro_0_DW01_inc_16_0'
because it's connected to the port by that name. (EDFO-10)
Information: Net 'net6221' is renamed to 'A<3>' in design 'synchro_0_DW01_inc_16_0'
because it's connected to the port by that name. (EDFO-10)
Information: Net 'net6222' is renamed to 'A<12>' in design 'synchro_0_DW01_inc_16_0'
because it's connected to the port by that name. (EDFO-10)
Information: Net 'net6223' is renamed to 'A<14>' in design 'synchro_0_DW01_inc_16_0'
Information: Net 'LTV1<0>' is renamed to 'A<0>' in design 'sequence_DW01_cmp2_32_0'
because it's connected to the port by that name. (EDFO-10)
Information: a bussed port name 'HFP' changed to 'HFP<15:0>'. (EXPT-11)
Information: a bussed port name 'HFW' changed to 'HFW<7:0>'. (EXPT-11)
Information: a bussed port name 'LFA' changed to 'LFA<4:0>'. (EXPT-11)
Information: a bussed port name 'LFI' changed to 'LFI<7:0>'. (EXPT-11)
Information: a bussed port name 'LFP' changed to 'LFP<15:0>'. (EXPT-11)
Information: a bussed port name 'LFW' changed to 'LFW<7:0>'. (EXPT-11)
Information: Net 'testMode' is renamed to 'test_so' in design 'comm_test_1' because it's
connected to the port by that name. (EDFO-10)
Information: a bussed port name 'AMP' changed to 'AMP<4:0>'. (EXPT-11)
1
check_test -verbose
  Loading design 'stim'

Information: Starting test design rule checking for existing scan design. (TEST-220)
  ...full scan rules enabled...
  ...basic checks...
  ...basic sequential cell checks...
  ...checking combinational feedback loops...
  ...inferring test protocol...
Information: Inferred system/test clock port clk (45.0,55.0). (TEST-260)
  ...simulating parallel vector...
  ...simulating parallel vector...
  ...simulating serial scan-in...
  ...1106 bits scanned-in to 1106 cells (total scan-in 1106)...
  ...simulating parallel vector...
  ...binding scan-in state...
  ...simulating parallel vector...
  ...simulating capture clock rising edge at port clk...
  ...simulating capture clock falling edge at port clk...
  ...simulating parallel vector...
```

```
  ...creating capture clock groups...
Information: Inferred capture clock group : clk. (TEST-262)
  ...binding scan-out state...
  ...simulating serial scan-out...
  ...data scanned-out from 1105 cells (total scan-out 1106)...
  ...simulating parallel vector...
Information: Test design rule checking completed. (TEST-123)

*****************************************************
  Test Design Rule Violation Summary

  Total violations: 0
*****************************************************



*****************************************************
  Sequential Cell Summary

  0 out of 1106 sequential cells have violations
*****************************************************

SEQUENTIAL CELLS WITHOUT VIOLATIONS
   *1106 cells are valid scan cells
        uc_module/dsp_H/SIGN_reg
        uc_module/dsp_H/EN_reg
        uc_module/dsp_H/AMP_reg<0>
        uc_module/dsp_H/AMP_reg<1>
        uc_module/dsp_H/AMP_reg<2>
        uc_module/dsp_H/AMP_reg<3>
        uc_module/dsp_H/AMP_reg<

        . . .

        comm_module/Sequence_Module/index_reg<0>
        comm_module/Sequence_Module/data_reg<3>
        comm_module/Sequence_Module/data_reg<2>
        comm_module/Sequence_Module/data_reg<1>
        comm_module/Sequence_Module/data_reg<0>
        comm_module/Sequence_Module/etat_reg<1>
        comm_module/Sequence_Module/etat_reg<0>

1
create_test_patterns -output stim_patterns.vdb
  Loading design 'stim'
  Using test design rule information from previous check_test run
  Building test generation network
Information: Faults on unused cell outputs are not considered, e.g. pin qb of cell
uc_module/dsp_H/SIGN_reg (tdxp). (TEST-200)

  Combinational Test Pattern Generation starts:

    Start random pattern generation...

       84.26% faults processed ; cumulative fault coverage =  84.26%
       87.71% faults processed ; cumulative fault coverage =  87.71%
       88.73% faults processed ; cumulative fault coverage =  88.73%
       89.24% faults processed ; cumulative fault coverage =  89.24%
       89.65% faults processed ; cumulative fault coverage =  89.65%
       89.93% faults processed ; cumulative fault coverage =  89.93%
       90.09% faults processed ; cumulative fault coverage =  90.09%
       90.29% faults processed ; cumulative fault coverage =  90.29%
       90.40% faults processed ; cumulative fault coverage =  90.40%
       90.46% faults processed ; cumulative fault coverage =  90.46%
       90.50% faults processed ; cumulative fault coverage =  90.50%
       90.54% faults processed ; cumulative fault coverage =  90.54%

    ...End random pattern generation

    Start deterministic pattern generation...

       90.94% faults processed ; cumulative fault coverage =  90.94%
       91.67% faults processed ; cumulative fault coverage =  91.67%
       92.20% faults processed ; cumulative fault coverage =  92.20%
       92.61% faults processed ; cumulative fault coverage =  92.61%
       93.03% faults processed ; cumulative fault coverage =  93.03%
       93.39% faults processed ; cumulative fault coverage =  93.38%
```

```
    93.72% faults processed ; cumulative fault coverage =  93.72%
    94.15% faults processed ; cumulative fault coverage =  94.14%
    94.45% faults processed ; cumulative fault coverage =  94.44%
    94.70% faults processed ; cumulative fault coverage =  94.69%
    95.06% faults processed ; cumulative fault coverage =  95.06%
    95.49% faults processed ; cumulative fault coverage =  95.48%
    95.71% faults processed ; cumulative fault coverage =  95.70%
    95.96% faults processed ; cumulative fault coverage =  95.96%
    96.13% faults processed ; cumulative fault coverage =  96.13%
    96.37% faults processed ; cumulative fault coverage =  96.37%
    96.54% faults processed ; cumulative fault coverage =  96.54%
    96.81% faults processed ; cumulative fault coverage =  96.81%
    97.21% faults processed ; cumulative fault coverage =  97.21%
    97.39% faults processed ; cumulative fault coverage =  97.39%
    97.68% faults processed ; cumulative fault coverage =  97.68%
    98.13% faults processed ; cumulative fault coverage =  98.13%
    98.44% faults processed ; cumulative fault coverage =  98.44%
    98.68% faults processed ; cumulative fault coverage =  98.68%
    98.96% faults processed ; cumulative fault coverage =  98.96%
    99.22% faults processed ; cumulative fault coverage =  99.22%
    99.54% faults processed ; cumulative fault coverage =  99.53%
    99.74% faults processed ; cumulative fault coverage =  99.73%
    99.92% faults processed ; cumulative fault coverage =  99.91%
   100.00% faults processed ; cumulative fault coverage =  99.99%

...End deterministic pattern generation


                              Non-collapsed     Collapsed
No. of detected faults        35191             22350
No. of abandoned faults       2                 2
No. of tied faults            0                 0
No. of redundant faults       3                 2
No. of untested faults        0                 0
Total no. of faults           35196             22354
Fault coverage                99.99             99.99

No. of test patterns           180

Test Generation Time (CPU)    42.27 sec

Start compaction...

...End compaction

No. of compacted patterns      175

Compaction Time (CPU)         3.10 sec
...Writing test program stim_patterns to file
/tmp_mnt/mnt/freia/3/cours3/vlsi4/synth/stim_patterns.vdb
1
quit
1
dc_shell>
Thank you...


                        DC Professional (TM)
                           DC Expert (TM)
                         FPGA Compiler (TM)
                         VHDL Compiler (TM)
                          HDL Compiler (TM)
                       Library Compiler (TM)
                        Test Compiler (TM)
                     Test Compiler Plus (TM)
                          CTV-Interface
                         DesignTime (TM)

                  Version v3.4b -- Apr 01, 1996
            Copyright (c) 1988-1995 by Synopsys, Inc.
                       ALL RIGHTS RESERVED

This program is proprietary and confidential information of Synopsys, Inc.
and may be used and disclosed only as authorized in a license agreement
controlling such use and disclosure.

Initializing...
```

```
read -format db stim.db
Loading db file '/mnt/freia/3/cours3/vlsi4/synth/stim.db'
Current design is now '/mnt/freia/3/cours3/vlsi4/synth/stim.db:stim'
{"stim"}
report_test -port -assertions -configuration
Loading db file '/usr/local/packages/synopsys3-4b/cmc/bicmos/syn/tcells.db'
Loading db file '/usr/local/packages/synopsys3-4b/libraries/syn/gtech.db'
Loading db file '/usr/local/packages/synopsys3-4b/libraries/syn/standard.sldb'
Warning: Unable to resolve reference 'uc_test_1' in 'stim'. (LINK-5)
Warning: Unable to resolve reference 'comm_test_1' in 'stim'. (LINK-5)

****************************************
Report : test
          -assertions
          -configuration
          -port
Design : stim
Version: v3.4b
Date   : Wed Apr 16 17:52:35 1997
****************************************

Assertions:



Test Methodology: full_scan

Scan style 'multiplexed_flip_flop' has signals:
  test_scan_enable (optional)
  test_scan_enable_inverted (optional)
  test_scan_in
  test_scan_out (optional)
  test_scan_out_inverted (optional)

Design has existing scan

Number of Scan Chains         : Default [1]
Clock Mixing                  : no_mix
Rebalance Scan Chains         : False
Isolate Hierarchical Outputs  : False

Add Lockups                   : True
Dedicated Scan Ports          : False
Replace                       : True
Route                         : True

Bidirectional Mode            : Input
Disable                       : True

                                                          Test Clocks
                                                      --------------------
Port              Signal Type                      Index  Rise   Fall  Period
----------------- -------------------------------- -----  ------ ------ ------
SIGN              test_scan_out
test_se           test_scan_enable
test_si           test_scan_in

1
quit
1
dc_shell>
Thank you...
```

# ANNEXE IV : FICHE TECHNIQUE DES ÉLECTRODES BIPOLAIRES

Electrode data 24/3/97 3

Produced by: MH Produced date: 21/1/97

## Cuff wall

Cuff length: 10 mm Elastomer type: MED-1137
Inner diameter: 1.5 mm
Closing method: Sutures

## Electrodes

Number of electrodes: 2
Electrode type: Foil with flaps
Electrode material: 25um platinum foil
Width of electrodes 1 mm

## Lead wires

Lead wire form: Straight Wire type: AS 632
Lead wire length: 300 mm Supplier: Cooner Wire Co

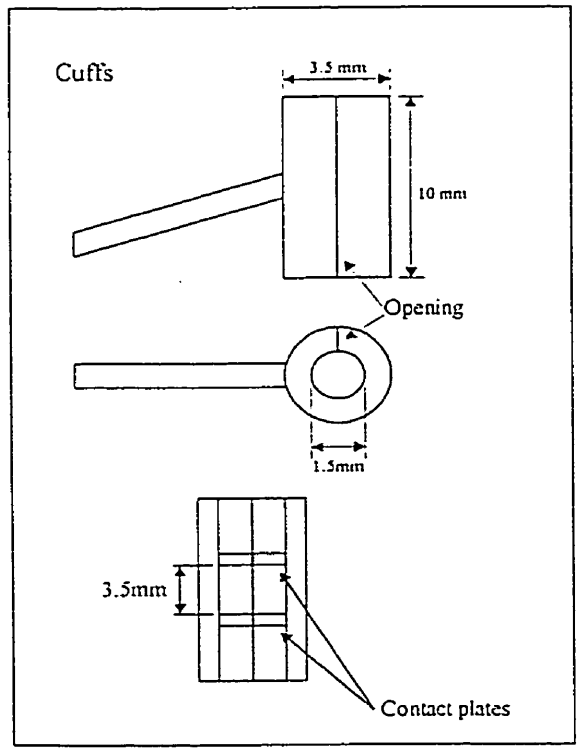
## Cleaning

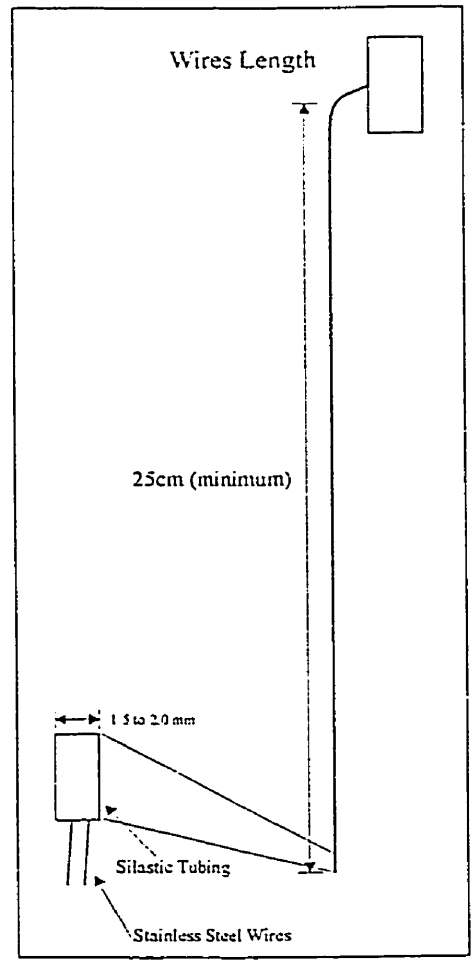
Cleaned date: Cleaned and packed by:
Sterilisation date: Sterilization method:
Sterilized by:

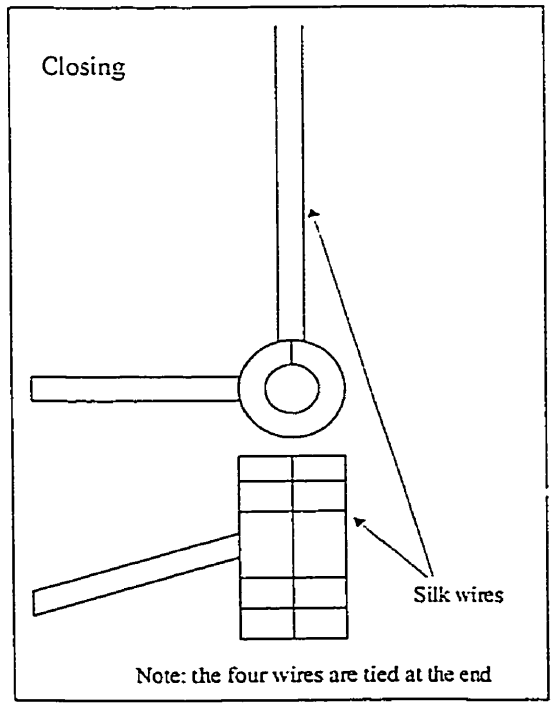
## Combined with

## Comments

Designed as specified by Simon Robin, Montreal (See faxed drawings). Interelectrode distance = 2.5mm, wires exiting at the middle of the cuff. Color coding: Brown = proximal, Red = distal.


Cuffs
3.5 mm
10 mm
Opening
1.5mm
3.5mm
Contact plates


Wires Length
25cm (minimum)
1.5 to 2.0 mm
Silastic Tubing
Stainless Steel Wires


Closing
Silk wires
Note: the four wires are tied at the end

# ANNEXE V : ARTICLES PUBLIÉS OU SOUMIS POUR PUBLICATION

# Implantable stimulation system dedicated for neural selective stimulation

S. Robin[1] M. Sawan[1] M. Abdel-Gawad[2] T. M. Abdel-Baky[2] M. M. Elhilali[2]

[1]Department of Electrical and Computer Engineering, École Polytechnique de Montréal, [2]Department of Urology, McGill University, Montréal, Canada

**Abstract**—*A functional electrical stimulation system is presented, which is dedicated for the selective neural stimulation of the bladder. The proposed system is composed of an internal stimulator (implant) and an external controller. The system is used to produce low-pressure voiding of the bladder in spinal cord injured patients. The implant is powered and operated by the external controller via radio-frequency electromagnetic coupling. All stimulation parameters are chosen externally using the controller and are sent to the implant, which produces the desired stimuli. These stimuli are applied directly to the $S_2$ nerve which is linked to the sphincter and bladder muscles. A high-frequency signal is used to inhibit the contraction of the sphincter muscle, and low-frequency pulses stimulate the bladder muscle (the detrusor). Dedicated computer software is used by the physician to select the optimal parameters for each patient and to activate the implant through a parallel port interface with built-in transmitter. The parameters are then transferred to a hand-held controller which is used by the technical staff and by the patients themselves. Acute studies have been performed to validate the selective stimulation strategy, and chronic experimentation is currently underway in dogs.*

Key words—*Selective stimulation, Implantable stimulator, Bladder, sacral nerve, Field-programmable technology.*

Med. Biol. Eng. Comput., 1998, 36, 1–3

## 1 Introduction

IN THE last 30 years, many types of implantable electrical stimulators have been introduced for treatment of neurological disorders and organ failures (GREATBATCH and HOLMES, 1991; MAYR *et al.*, 1996). More recently, many attempts have been made to recuperate the functionality of the urinary tract system for spinal cord injured (SCI) patients through sacral root stimulation (TANAGHO *et al.*, 1989; BRINDLEY, 1994). Existing techniques, such as post-stimulus voiding, unfortunately either induce simultaneous contraction of the sphincter and the bladder muscles since the nerve that controls the bladder is also connected to the sphincter (SAWAN *et al.*, 1996), or require pudendal neurectomy or rhizotomy (TANAGHO *et al.*, 1989; BRINDLEY, 1994). The dyssynergia leads to high vesical pressure, and can eventually lead to incontinence or kidney failure; the pudendal neurectomy is an irreversible process that sometimes leads to impairment of reflexogenic erection in paraplegic males.

Selective stimulation (FANG and MORTIMER, 1991; RIJKHOFF *et al.*, 1997; TU *et al.*, 1997; ROBIN *et al.*, 1996) enables dyssynergia to be minimised without neurectomy or rhizotomy. The goal is to reach the stimulation threshold of the somatic fibres with a high-frequency stimulus and thus inhibit the sphincter contraction, which means creating a blockage on the fibre conduction. These low-amplitude and high-frequency stimuli remain under the threshold of the autonomic fibres reaching the bladder. Therefore, the low-frequency and high-amplitude stimuli are unable to reach the sphincter while they activate the bladder through the autonomic fibres.

Currently available functional electrical stimulation (FES) systems like the Avery, Medtronic and Brindley-Finetech systems lack the features needed for selective stimulation such as generation of high-frequency pulses up to 1 kHz and two signals of independent parameters mixed on the same channel (SAWAN *et al.*, 1996). Our proposed system contains those features that are required for selective stimulation. Our FES is currently being used in chronic dog experimentation, and the stimulation technique has been proven in previous acute studies (TU *et al.*, 1997; ROBIN *et al.*, 1996).

## 2 System description

The proposed stimulation system is mainly composed of an implant and an external controller. Fig. 1 shows the system with the cuff electrode implanted around the nerve and connected to the implant, as well as the two models of external controllers, both with built-in RF transmitters.

*Correspondence should be addressed to Dr. M. Sawan; email: sawan@vlsi.polymtl.ca*

*st received 30 May 1997 and in final form 18 March 1998.*

© IFMBE: 1998

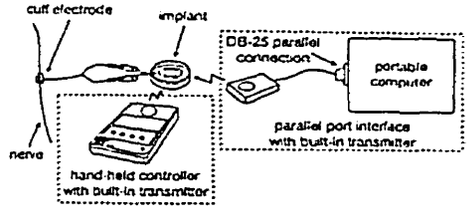



Fig. 1 *Complete stimulation system*

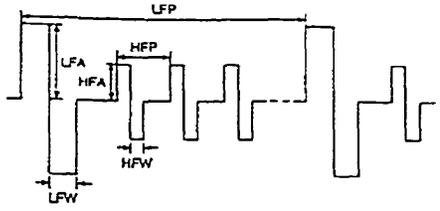



Fig. 2 *Typical waveform generated by the implant*

### 2.1 *Controller with computer interface*

Dedicated user-friendly software allows the physician to quickly view and change all stimulation parameters. The parameters can be saved and retrieved from the disk, and they can also be exported to a binary file to be used to program the read-only memory (ROM) of the hand-held controller.

The software can also be used to initiate the stimulation through the external interface, with built-in RF transmitter connected to the parallel port. This method gives full control over the parameters in order to reach the optimal settings for each patient. The interface was built using the non-volatile field programmable gate array (FPGA) technology. Voltage regulators were also added to produce the required voltage for the digital circuitry as well as the transmitter.

### 2.2 *Hand-held controller*

A very simple user interface, with only a few push-buttons and a 16 × 2 character liquid crystal display (LCD), allows the user to select the stimulation and activate it with the parameters previously selected by the physician. The display can be configured to be as simple as showing just the name of the patient. The hand-held unit is 19 cm × 10 cm × 3.3 cm and weigh 290 g.

Everything that is required is incorporated in a single box, which can either be plugged into an AC outlet or battery-operated. The electronic circuit is primarily composed of an FPGA and a ROM. The ROM is used to store the content of the display as well as the parameters sets, and the FPGA contains almost all the logic circuitry. A step-up regulator is included to provide the required 12 V for the transmitter from a standard 9 V battery.

### 2.3 *Radiofrequency-coupled interface*

Amplitude modulation (M) of a 20 mHz carrier is used to communicate energy and digital data to the implant from the controller, via an inductive coupling link. Data and a synchronisation clock are Manchester-encoded and serially transmitted. This AM modulator is based on a local oscillator that generates the signal to a class D amplifier, which drives the current through the transmission coil (SAWAN *et al.*, 1996). We use the same transmitter circuit for both models of external controller.

### 2.4 *Implantable stimulator*

The implant, connected to a bipolar cuff electrode, constitutes the internal part of our stimulation system. The implant is powered by the external controller and is inactive without its presence. Once the implant is properly powered up, it waits for a distinctive sequence of bits that are sent via the RF transmission link. The beginning of the sequence starts with a specific header and is followed by all necessary parameters for selective stimulation. The parameters and their ranges are shown in Table 1, and the typical waveform is depicted in Fig. 2.

The input stage of the implant uses discrete components to make the rectifier, regulator and the AM demodulator (SAWAN *et al.*, 1996). All the logic required to decode the signal, detect the header, store the parameters and generate the control commands used by the output stage is included in a single non-volatile FPGA, placed at the centre of the PCB. The output stage is composed of a digital/analogue convertor (DAC), a voltage/current amplifier and an analogue switch array used to change the polarity of the current. The implant is fabricated in a 4 cm circular printed circuit board, which holds the surface mounted technology components on both sides. A photograph of the implant is shown in Fig. 3.

Once the circuit is thoroughly tested, it is dipped in conformal coating to ensure its protection against humidity. Finally, the implant is moulded in Silastic for bio-compatibility.

## 3 Discussion

We have presented an implantable electrical stimulation system dedicated to produce selective stimulation. Our system is versatile by presenting many possibilities to the physician, while being very simple and safe for use by the technical staff at the McIntyre Animal Resources Centre of McGill University. The use of FPGA technology in the different parts of our system allows for a short development time, and it also enables it to be reconfigured for use in other neuromuscular stimulation applications such as incontinence or pain control.

Functional tests at all design and fabrication steps have been successfully completed. *In vitro* testing was performed, and size implant prototypes are in use *in vivo* and showing the predicted performances. Four controllers of various types are used every day with success by the physicians and medical

*Table 1 Main available parameters and range*

| | low frequency | | | high frequency | | |
|---|---|---|---|---|---|---|
| parameter | period | pulse width | amplitude | period | pulse width | amplitude |
| mnemonic | LFP | LFW | LFA | HFP | HFW | HFA |
| units | 1 $Hz^{-1}$ | µs | mA | 1 $Hz^{-1}$ | µs | mA |
| range | 1/100–1/10 | 0–300 | 0–2.0 | 1/1000–1/100 | 0–200 | 0–2.0 |

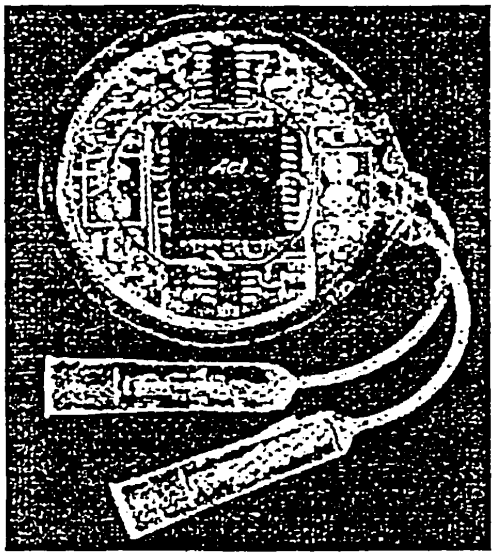


Fig. 3 *Photograph (top-side) of implantable stimulator*

staff. Although it is early in the chronic experimentation phase, present results are encouraging. We are currently in the process of tuning all parameters for each animal while collecting statistical data.

Our forthcoming research work includes a stimulator entirely integrated on a single silicon die for use in small animal experiments. Work is also underway on another model ! would include telemetry functions, enabling us to monitor ious internal events (nerve impedance and implant diagnostics).

*Acknowledgments*—The authors would like to acknowledge the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and from the Kidney Foundation of Canada (KFC); they would also like to thank P. Vaillancourt, J.-F. Harvey and K. Arabi for their input to this project.

## References

BRINDLEY, G. S. (1994): 'The first 500 patients with sacral anterior root stimulation implants', *Paraplegia*, 32, pp. 795–805

FANG, Z. P., and MORTIMER, J. T. (1991): 'Selective activation of small motor axons by quasitrapezoidal current pulses', *IEEE Trans.*, BME-38, pp. 168–174

GREATBATCH, W., and HOLMES, C. (1991): 'History of implantable devices', *J. IEEE Eng. Med. Biol. Soc.*, pp. 38–41

MAYR, W., Bijak, M., GIRSCH, W., LANMÜLLER, H., RAFOLT, D., SAUERMANN, S., SCHNETZ, G., and UNGER, E. (1996): 'Functional electrostimulation via implants applications, limitations, perspectives'. Conf. on Interdisciplinary Aspects of Computers Helping People with Special Needs, Linz, Austria, pp. 311–317

RIJKHOFF, N. J. M., WIJKSTRA, H., VAN KERREBROEK, P. E. V., and DEBRUYNE, F. M. J. (1997): 'Selective detrusor activation by electrical sacral nerve root stimulation in spinal cord injury', *J. Urol.*, 157, pp. 1504–1508

ROBIN, S., TU, L. M., ARABI, K., SHAKER, H., SAWAN, M., and ELHILALI, M. M. (1996): 'A new system dedicated to selective stimulation: acute studies in dogs'. *IEEE/EMBS Proc.*, Amsterdam, The Netherlands

SAWAN, M., HASSOUNA, M. M., LI, J. S., DUVAL, F., and ELHILALI, M. M. (1996): 'Stimulator design and subsequent parameter optimisation for controlling micturition and reducing urethral resistance', *IEEE Trans. Rehab. Eng.*, 4, pp. 39–46

TANAGHO, E., SCHMIDT, R. A., and ORVIS, B. R. (1989): 'Neural stimulation for control of voiding dysfunction: a preliminary report in 22 patients with serious neuropathic voiding disorders'. *J. Urol.*, 142, pp. 340–345

TU, L. M., SHAKER, H. S., ROBIN, S., ARABI, K., HASSDOUNA, M., SAWAN, M., and ELHILALI, M. M. (1997): 'Reduction in bladder outlet resistance by selective sacral root stimulation using high-frequency blockade in dogs: an acute study', *J. Urol.* 9.

# A New Implantable Microstimulator Dedicated to Selective Stimulation of the Bladder

S. Robin, M. Sawan, J.F. Harvey, M. Abdel-Gawad*, T. M. Abdel-Baky*, M. M. Elhilali*
Department of Electrical and Computer Engineering, École Polytechnique de Montréal
* Department of Urology, McGill University, Montréal
robin@vlsi.polymtl.ca

*Abstract* — In this paper, we present a new implantable functional electrical stimulation (FES) system specially designed for the purpose of selective stimulation of the bladder in spinal cord injured patients. This system, composed of an external controller and an implantable microstimulator (implant), is used to induce low-pressure micturition. After receiving both energy and the stimulation parameters via radio-frequency coupling from the controller, the implant starts stimulation of the $S_2$ sacral nerve. The stimulation, composed of two independent frequencies, mainly stimulates the detrusor while blocking the stimulation of the sphincter even thought both muscles are linked to the $S_2$ nerve. A computer controlled controller is used by the physician to determine the best parameters for the patient. These parameters are then stored in a portable handheld unit that is used by the medical staff or patient. Acute studies in dogs were carried out to validate our technique and chronic studies started five months ago.

## I. Introduction

Implantable neuromuscular stimulators have been used in a variety of applications over the last two decades. Electrical stimulation of the sacral roots has been used in the past in order to recuperate the functionality of the bladder in spinal cord injured patients [1,2]. It has proven itself to be a complicated problem because of the mixed somatic and autonomic fibers composing the ventral sacral root. Previous electrical stimulation techniques like post-stimulus voiding [3] cause simultaneous contraction of the sphincter and the detrusor muscle (dyssynergia) which increases the bladder pressure and can eventually cause other problems like renal failure.

In this study, we present a new implantable stimulation system based on the technique of selective stimulation of the sacral roots for low pressure voiding.

## II. General Description

The complete FES system is composed of an internal part (implant and electrode) and an external part (controller and transmitter). Figure 1 shows the complete stimulation system that we designed and are currently using. The external controller can be either the portable unit or the computerized interface.

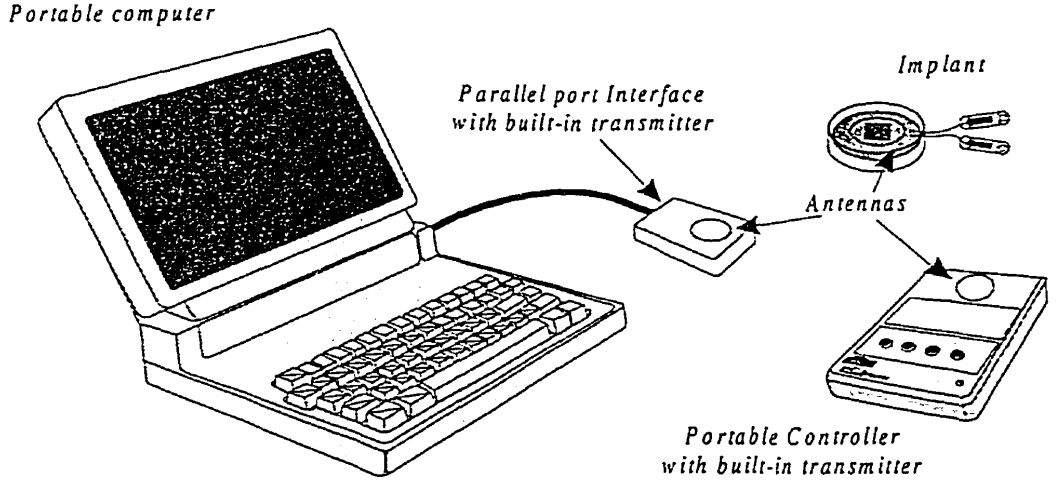



Fig. 1. The complete stimulation system.

### A. External Controllers

The external controller, whether portable or computerized, serves the purpose of transmitting energy and data to the implant in order to ensure its adequate operation.

#### Computer Interface

The computerized controller (figure 2) is in fact an interface bridging the computer to a radio-frequency transmitter via the parallel port of any PC compatible computer. The interface itself is built using non-volatile Field Programmable Gate Array technology (FPGA). A dedicated software translates the stimulation parameters into the bit stream that is required for serial transmission to the implant. The physician can adjust all the parameters with a user-friendly interface that allows saving, retrieving and also exporting of parameters sets. Once exported, the parameters sets can be used to program the read-only memory (EPROM) of the portable unit.

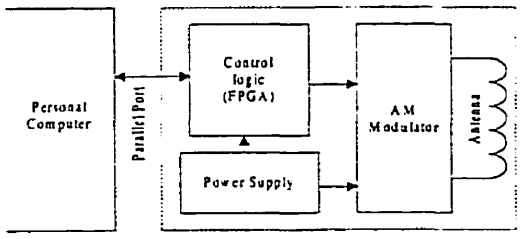



Fig. 2. Block diagram of the computerized controller.

#### Portable Unit

The portable unit (figure 3) is 19 cm x 10 cm x 3.3 cm in size and weights 290 grams. It incorporates everything inside a single box. It can be battery operated or plugged into an AC outlet.

The digital circuit is composed primarily of an EPROM and a non-volatile FPGA. The EPROM is used to store all the sets of parameters previously chosen by the physician with the computerized version of the controller. The FPGA contains almost all the logic of the entire controller.

The user interface is composed of a liquid crystal display (LCD) and four buttons are used to choose between the sets and to activate the stimulation. The content of the LCD screen is chosen by the physician and can be as simple as displaying the name of the patient.

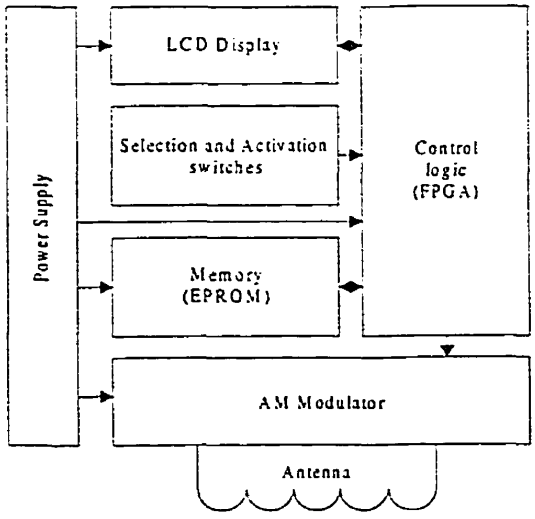



Fig. 3. Block diagram of the handheld controller.

### B. Implantable Prosthesis

The internal part (the implant) is powered by the external transmitter and is inert without its presence. After the implant is powered-up, it waits for a specific sequence of bits to start the stimulation. The data stream contains a special header and all the parameters required for the application. Figure 4 depicts a block diagram and figure 5 shows a photograph of the implant.

The implant is built on a 4 cm circular printed circuit board (PCB) that holds a FPGA and the commercially available components.

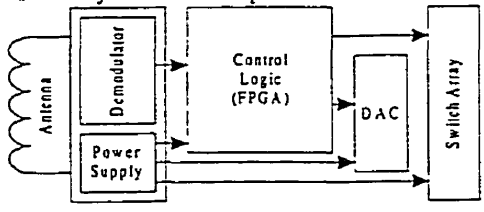



Fig. 4. Block diagram of the implantable stimulator.

The input of the stimulator is composed of an AM demodulator designed with discrete surface mounted technology (SMT) components that is used to recover

the encoded data, a rectifier bridge with shunt regulator that is used to produce a constant operating voltage and a power-on reset circuit.

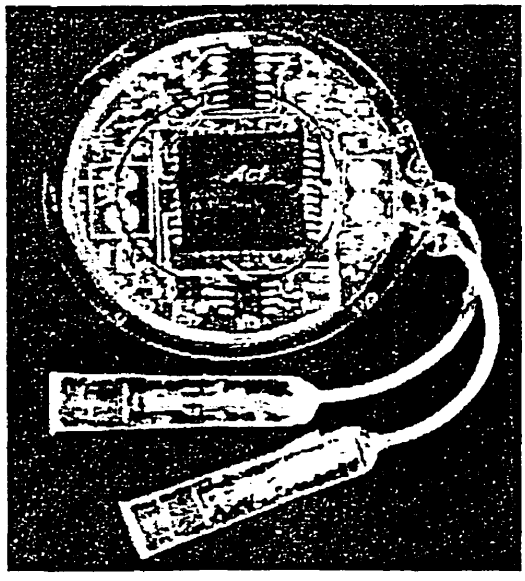


Fig. 5. Photograph of the implantable stimulator.

At the center of the implant is the FPGA which incorporates all the digital logic required for decoding and saving the parameters and also generating the proper digital control signals used by the output modules.

The output stimuli is generated from a digital to analog converter (DAC), a current amplifier and a solid-state analog switch array that is used to change the polarity of the current.

The circuit is then dipped in conformal coating or Araldite epoxy resin for protection against humidity and then molded in Silastic to ensure its bio-compatibility.

### C. *Radiofrequency-Coupled Interface*

Inductive-coupling is used to transmit the digital coded signal to the implant. It is composed of an AM modulator that produces a 20-MHz carrier which is modulated by the serial data. A class D amplifier is then used to transmit the signal through the transmission coil. Manchester encoding is used for the digital data that is transmitted at a rate of 300 kHz. The communication interface is enclosed in the latest models of handheld and computerized controllers.

## III. Materials and Methods

### A. *Experimentation on Chronic Dogs*

After spinalization at the 10$^{th}$ thoracic vertebra, the sacral roots are exposed through an extended sacral laminectomy. $S_2$ sacral roots are then identified with the response to electric stimulation of the bladder and sphincter activity. Electromyogram of the external sphincter as well as pressures from the bladder and the

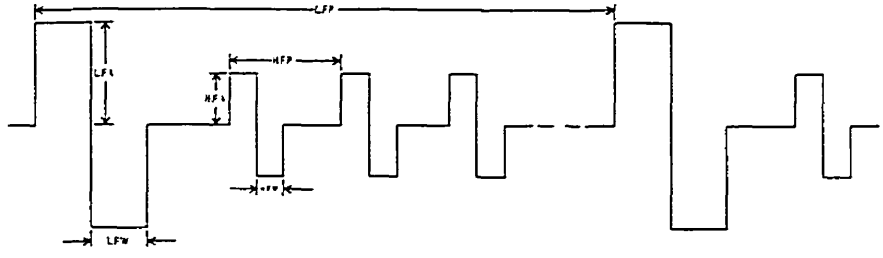



Fig. 6. Typical waveform generated by the implant.

Table 1: Main available parameters and range.

| | Low Frequency | | | High Frequency | | |
|---|---|---|---|---|---|---|
| Parameter | Period | Pulse width | Amplitude | Period | Pulse width | Amplitude |
| Mnemonic | LFP | LFW | LFA | HFP | HFW | HFA |
| Units | 1/Hz | micro-sec. | mA | 1/Hz | micro-sec. | mA |
| Range | 1/100 - 1/10 | 0-300 | 0-2.0 | 1/1000-1/100 | 0-100 | 0-2.0 |

urethra are monitored and recorded with a urodynamic recording system (UDS 120 from Laborie Inc.).

A platinum-cuff electrode [6] is attached around the nerve giving the best response to electrical stimulation. The electrode is then connected to the implant (with our dedicated connectors) which is then inserted under the skin with its antenna facing outward.

Figure 6 shows the typical waveform used and table 1 shows the parameters and their range.

## IV. Preliminary Results

Out of six implants ready for implantation, three have already been implanted. Although it is too early in the evaluation phase to completely validate our technique of selective stimulation under chronic experimentation, we did have some promising results with similar parameters than those used in previous acute studies. We did get reduced urethral pressure and EMG activity with addition of the high frequency waveform of 600 Hz frequency and 10-75 µs pulse width over the low frequency waveform of 30 Hz frequency and 150-250 µs pulse width. The system is working according to specifications since the beginning of experimentation. There was a problem with the first dog when an electrode wire broke. We then opted for stronger electrode wires for the next dogs.

## V. Discussion

This paper covers the development of a new generation of neural prosthesis dedicated to selective stimulation. This new stimulator is capable of a wide range of stimulation possibilities with accuracy. The system has been working properly with only minor problems since the beginning of experimentation.

Selective electrical stimulation of the sacral roots has been evaluated in our previous acute studies on 11 dogs [4,5]. This study demonstrated the possibility to produce low pressure micturition by stimulating the sacral nerve $S_2$ with a combination of two independent frequencies.

The use of FPGA technology in both the controllers and the implant gave us a short development time as well as reconfigurability which are two valuable aspects in a research environment.

## VI. Conclusion

We have described a new FES system dedicated to the selective stimulation of the sacral nerve in order to produce low pressure voiding. We believe that this system could be used, with no or minor modification, for other neural stimulation applications. This is due mostly to the great flexibility of the parameters and also to the use of FPGA technology that allows for reconfigurability of any of the component of our system without modification of the PCBs.

We are presently working on a similar implant that will be totally or mostly integrated on a single silicon die. This would allow its use in smaller animals like rats. We are also working on another version of the implant that would allow feedback functions (telemetry).

## Acknowledgments

Authors would like to acknowledge the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and from the Kidney Foundation of Canada (KFC).

## References

[1] M. Sawan *et al.*, "Stimulator Design and Subsequent Stimulation Parameter Oprimization for Controlling Micturition and Reducing Urethral Resistance", IEEE Transactions on Rehabilitation Engineering, Vol. 4, No. 1, March, pp. 39-46, 1996.

[2] M. Sawan, F. Duval, M. Hassouna, M.M. Elhilali, "A New Transcutaneous Fully-Programmable Neural Stimulator", International Journal of Microcomputer Applications, Vol. 13, No. 3, 1994.

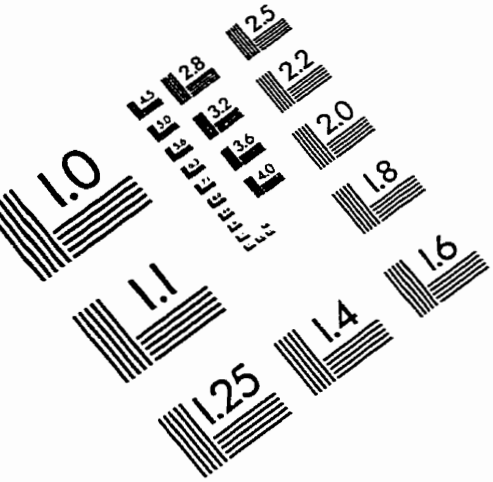
[3] U. Jonas and E. A. Tanagho, "Studies on the feasability of urinary bladder evacuation by direct spinal cord stimulation. Post-stimulus voiding: a way to overcome outflow resistance", Invest. Urol., No. 13, p. 151, 1975.

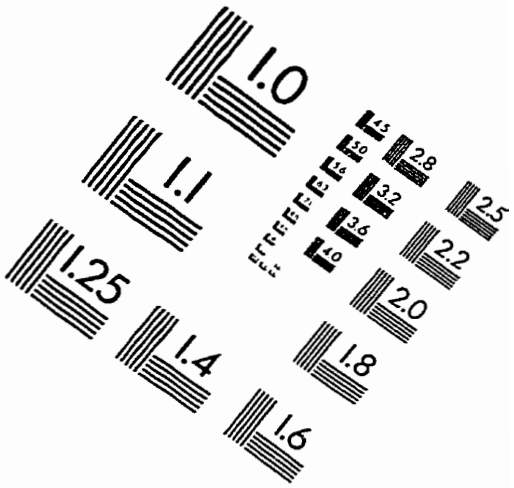
[4] S. Robin et al., "A New System Dedicated to Selctive Stimulation: Acute Studies in Dogs", IEEE/EMBS Proceedings, Amsterdam, 1996.

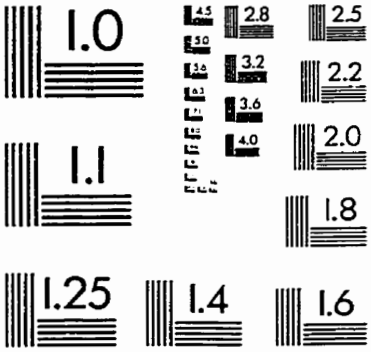
[5] L. M. Tu, H.S. Shaker, S. Robin, K. Arabi, M. Hassouna, M. Sawan, M.M. Elhilali, "Reduction of bladder outlet resistance by selective sacral root stimulation using high-frequency blockade in dogs: An acute study", Submitted to the Journal of Urology in 1997.

[6] M. Haugland, "A Flexible Method for Fabrication of Nerve Cuff Electrodes ", IEEE/EMBS Proceedings, Amsterdam, 1996.

# IMAGE EVALUATION TEST TARGET (QA-3)

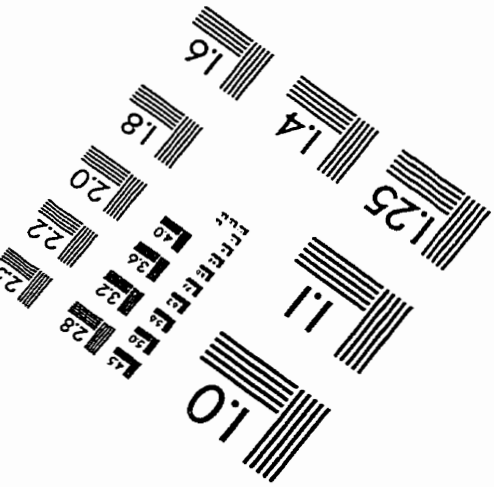


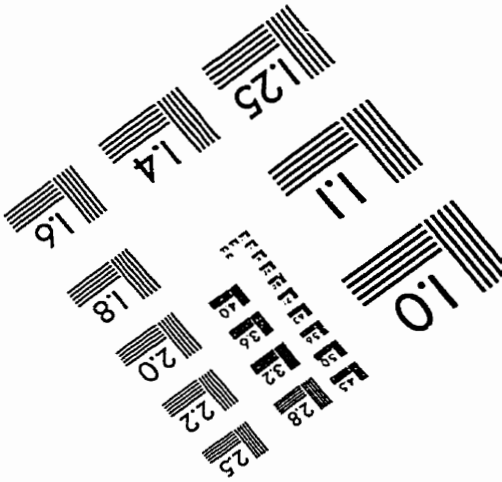




150mm

6″





APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved